Libero SoC Tcl Command Reference Guide

Libero SoC v2025.1

Introduction (Ask a Question)

The Microchip Libero® System-on-Chip (SoC) design suite offers high productivity with its comprehensive, easy-to-learn, easy-to-adopt development tools for designing with Microchip's power efficient Flash Field-Programmable Gate Arrays (FPGAs), SoC FPGAs, and Rad-Tolerant FPGAs. The suite integrates industry-standard Synopsys SynplifyPro® synthesis and Siemens EDA QuestaSim® simulation with best-in-class constraints management, debug capabilities, and secure production programming support.

Supported Device Family

The following table lists the family of devices that Libero SoC supports. This guide covers all these device families. However, some information in this guide might apply to certain device families only. In this case, such information is clearly identified.

Table 1. Device Family Supported by Libero SoC

Device Family	Description
PolarFire®	PolarFire FPGAs deliver the industry's lowest power at mid-range densities with exceptional security and reliability.
PolarFire SoC	PolarFire SoC is the first SoC FPGA with a deterministic, coherent RISC-V CPU cluster, and a deterministic L2 memory subsystem enabling Linux® and real-time applications.
SmartFusion® 2	SmartFusion 2 addresses fundamental requirements for advanced security, high reliability, and low power in critical industrial, military, aviation, communications, and medical applications.
IGLOO® 2	IGLOO 2 is a low-power mixed-signal programmable solution.
RTG4™	RTG4 is Microchip's family of radiation-tolerant FPGAs.

Important: This document is updated frequently. The latest version of this document is available at this location: Libero SoC Design Suite Documentation.

Documentation Scope and Audience

This document is intended for automation engineers with good knowledge of system-level scripting.

© 2025 Microchip Technology Inc. and its subsidiaries

Table of Contents

Introduction....1

1. Tcl Scripting Overview....13
   - 1.1. Tcl Commands and Supported Families....13
   - 1.2. Tcl Command Documentation Conventions....13
   - 1.3. Basic Syntax....14
   - 1.4. Types of Tcl Commands....16
   - 1.5. Variables....16
   - 1.6. Command Substitution....17
   - 1.7. Quotes and Braces....17
   - 1.8. Filenames....18
   - 1.9. Lists and Arrays....18
   - 1.10. Control Structures....19
   - 1.11. Print Statement and Return Values....20
   - 1.12. Running Tcl Scripts from the Command Line....21
   - 1.13. Exporting Tcl Scripts....22
   - 1.14. extended_run_lib....22
   - 1.15. Sample Tcl Script - Project Manager....24
   - 1.16. How to Derive Required Part Information from A "Part Number"....25

2. Building a Libero Design Using Tcl....26
   - 2.1. HDL to Programming Flow....26
   - 2.2. Running the Tcl....31
   - 2.3. Updating a Project....31

3. Project Manager Tcl Commands....32
   - 3.1. add_file_to_library....32
   - 3.2. add_library....32
   - 3.3. add_modelsim_path....33
   - 3.4. add_profile....34
   - 3.5. associate_stimulus....36
   - 3.6. build_design_hierarchy....37
   - 3.7. change_link....37
   - 3.8. change_all_links....38
   - 3.9. change_vault_location....39
   - 3.10. check_fdc_constraints....40
   - 3.11. check_hdl....41
   - 3.12. check_ndc_constraints....41
   - 3.13. check_pdc_constraints....42
   - 3.14. check_sdc_constraints....43
   - 3.15. cleanall_tool....44
   - 3.16. close_design....45
   - 3.17. close_project....45
   - 3.18. configure_core....46
   - 3.19. create_and_configure_core....47
   - 3.20. configure_tool....49
   - 3.21. create_links....50

© 2025 Microchip Technology Inc. and its subsidiaries

3.22. create_set....52
3.23. create_smartdesign....53
3.24. delete_component....54
3.25. delete_files....55
3.26. download_core....55
3.27. download_latest_cores....56
3.28. edit_profile....57
3.29. edit_post_layout_design....59
3.30. export_as_link....60
3.31. export_ba_files....61
3.32. export_bitstream_file....62
3.33. export_bsdl_file....69
3.34. export_component_to_tcl....70
3.35. export_design_summary....71
3.36. export_firmware....72
3.37. export_fp_pdc....73
3.38. export_ibis_file....74
3.39. export_interrupt_map....74
3.40. export_io_pdc....75
3.41. export_job_data....76
3.42. export_netlist_file....78
3.43. export_parameter_report....78
3.44. export_pin_reports....79
3.45. export_profiles....81
3.46. export_prog_job....82
3.47. export_script....85
3.48. export_sdc_file....85
3.49. export_smart_debug_data....86
3.50. force_update_design_flow....89
3.51. generate_component....90
3.52. generate_sdc_constraint_coverage....91
3.53. get_libero_release....92
3.54. get_libero_version....92
3.55. get_tool_options....93
3.56. get_tool_state....96
3.57. import_component....97
3.58. import_component_data....98
3.59. import_files....100
3.60. import_mss_component....102
3.61. is_synthesis_enabled....103
3.62. new_project....103
3.63. one_way_passcode....108
3.64. open_project....109
3.65. open_smartdesign....110
3.66. organize_constraints....111
3.67. organize_sources....112
3.68. organize_tool_files....113
3.69. project_general_settings....114
3.70. project_settings....116

© 2025 Microchip Technology Inc. and its subsidiaries

3.71. publish_block....119
3.72. refresh....121
3.73. remove_core....121
3.74. remove_library....122
3.75. remove_profile....123
3.76. rename_file....123
3.77. rename_library....124
3.78. run_tool....125
3.79. save_log....128
3.80. save_project....128
3.81. save_project_as....129
3.82. save_smartdesign....130
3.83. select_profile....131
3.84. set_actel_lib_options....131
3.85. set_active_testbench....132
3.86. set_as_target....133
3.87. set_device....134
3.88. set_global_include_file....136
3.89. set_global_include_path_order....137
3.90. set_modelsim_options....138
3.91. set_option....141
3.92. set_root....142
3.93. set_user_lib_options....142
3.94. smartdesign....143
3.95. unlink_copy_locally....144
3.96. unlink_files....145
3.97. unset_as_target....146
3.98. unset_global_include_file....147
3.99. update_component_version....147
3.100. use_source_file....148
3.101. update_and_run_tool....149

4. SmartDesign Tcl Commands....153
4.1. create_smartdesign_testbench....153
4.2. sd_add_pins_to_group....153
4.3. sd_apply_presentation....154
4.4. sd_clear_pin_attributes....155
4.5. sd_configure_core_instance....156
4.6. sd_connect_instance_pins_to_ports....157
4.7. sd_connect_net_to_pins....158
4.8. sd_connect_pins_to_constant....159
4.9. sd_connect_pins....160
4.10. sd_connect_pin_to_port....161
4.11. sd_copy_paste....162
4.12. sd_create_bif_net....164
4.13. sd_create_bif_port....165
4.14. sd_create_bus_net....167
4.15. sd_create_bus_port....168
4.16. sd_create_pin_group....169

© 2025 Microchip Technology Inc. and its subsidiaries

4.17. sd_create_pin_slices 170
4.18. sd_create_scalar_net 171
4.19. sd_create_scalar_port 173
4.20. sd_cut_paste 173
4.21. sd_delete_bif_pin 175
4.22. sd_delete_instances 176
4.23. sd_delete_nets 177
4.24. sd_delete_pin_group 178
4.25. sd_delete_pin_slices 179
4.26. sd_delete_ports 180
4.27. sd_disconnect_instance 181
4.28. sd_disconnect_pins 182
4.29. sd_duplicate_instance 183
4.30. sd_hide_bif_pins 184
4.31. sd_instantiate_component 186
4.32. sd_instantiate_core 187
4.33. sd_instantiate_hdl_core 188
4.34. sd_instantiate_hdl_module 189
4.35. sd_instantiate_macro 190
4.36. sd_invert_pins 191
4.37. sd_mark_pins_unused 192
4.38. sd_remove_pins_from_group 193
4.39. sd_rename_instance 194
4.40. sd_rename_net 195
4.41. sd_rename_pin_group 196
4.42. sd_rename_port 197
4.43. sd_replace_component 198
4.44. sd_reset_layout 200
4.45. sd_set_port_synth_attr 201
4.46. sd_set_net_synth_attr 202
4.47. sd_set_inst_synth_attr 203
4.48. sd_set_comp_synth_attr 205
4.49. sd_remove_port_synth_attr 206
4.50. sd_remove_net_synth_attr 207
4.51. sd_remove_inst_synth_attr 209
4.52. sd_remove_comp_synth_attr 210
4.53. sd_save_core_instance_config 211
4.54. sd_show_bif_pins 212
4.55. sd_update_instance 213

5. HDL Tcl Commands 215
5.1. create_hdl_core 215
5.2. remove_hdl_core 216
5.3. hdl_core_add_bif 216
5.4. hdl_core_assign_bif_signal 217
5.5. hdl_core_delete_parameters 219
5.6. hdl_core_extract_ports_and_parameters 220
5.7. hdl_core_remove_bif 220
5.8. hdl_core_rename_bif 221

© 2025 Microchip Technology Inc. and its subsidiaries

5.9. hdl_core_unassign_bif_signal 222

6. Command Tools 224
6.1. CONFIGURE_ACTIONS_PROCEDURES 224
6.2. CONFIGURE_CHAIN 225
6.3. CONFIGURE_PROG_OPTIONS 226
6.4. EXPORTNETLIST 228
6.5. EXPORTSDF 229
6.6. FLASH_FREEZE 230
6.7. GENERATEDEBUGDATA 231
6.8. GENERATEPROGRAMMINGDATA 231
6.9. GENERATEPROGRAMMINGFILE 232
6.10. INIT_LOCK 234
6.11. IO_PROGRAMMING_STATE 236
6.12. PLACEROUTE 237
6.13. PROGRAMDEVICE 241
6.14. PROGRAMMER_INFO 243
6.15. PROGRAM_SPI_FLASH_IMAGE 246
6.16. SIM_PRESYNTH 247
6.17. SIM_POSTSYNTH 248
6.18. SIM_POSTLAYOUT 249
6.19. SPM 249
6.20. SPM_OTP 254
6.21. SYNTHESIZE 255
6.22. UPDATE_ENVM 258
6.23. USER_PROG_DATA 259
6.24. VERIFYPOWER 260
6.25. VERIFYTIMING 261

7. MSS Tcl Commands 265
7.1. mss_configure_envm 265
7.2. mss_configure_instance 265
7.3. mss_disable_instance 265
7.4. mss_enable_instance 266
7.5. mss_save_instance_config 266

8. SmartTime Tcl Commands 267
8.1. all_inputs 267
8.2. all_outputs 267
8.3. all_registers 268
8.4. check_constraints 269
8.5. clone_scenario 270
8.6. create_clock 270
8.7. create_generated_clock 272
8.8. create_scenario 276
8.9. create_set 277
8.10. expand_path 278
8.11. get_cells 280
8.12. get_clocks 281

© 2025 Microchip Technology Inc. and its subsidiaries

8.13. get_current_scenario....281
8.14. get_nets....282
8.15. get_pins....283
8.16. get_ports....284
8.17. list_clock_groups....285
8.18. list_clock_latencies....286
8.19. list_clock_uncertainties....286
8.20. list_clocks....287
8.21. list_disable_timings....288
8.22. list_false_paths....288
8.23. list_generated_clocks....289
8.24. list_input_delays....290
8.25. list_max_delays....290
8.26. list_min_delays....291
8.27. list_multicycle_paths....292
8.28. list_objects....292
8.29. list_output_delays....293
8.30. list_paths....294
8.31. list_scenario....295
8.32. read_sdc....296
8.33. remove_all_constraints....297
8.34. remove_clock....297
8.35. remove_clock_groups....298
8.36. remove_clock_latency....299
8.37. remove_clock_uncertainty....300
8.38. remove_disable_timing....302
8.39. remove_false_path....303
8.40. remove_generated_clock....304
8.41. remove_input_delay....305
8.42. remove_max_delay....306
8.43. remove_min_delay....307
8.44. remove_multicycle_path....308
8.45. remove_output_delay....309
8.46. remove_scenario....310
8.47. remove_set....311
8.48. rename_scenario....311
8.49. report....312
8.50. save....315
8.51. set_clock_groups....316
8.52. set_clock_latency....318
8.53. set_clock_to_output....320
8.54. set_clock_uncertainty....321
8.55. set_current_scenario....324
8.56. set_disable_timing....324
8.57. set_external_check....325
8.58. set_external_delay....326
8.59. set_false_path....327
8.60. set_input_delay....329
8.61. set_input_jitter....331

© 2025 Microchip Technology Inc. and its subsidiaries

8.62. set_max_delay....332
8.63. set_min_delay....334
8.64. set_multicycle_path....336
8.65. set_options....338
8.66. set_output_delay....340
8.67. set_system_jitter....343
8.68. write_sdc....343

9. SmartPower Tcl Commands....345
9.1. smartpower_add_new_custom_mode....345
9.2. smartpower_add_new_scenario....346
9.3. smartpower_add_pin_in_domain....347
9.4. smartpower_battery_settings....348
9.5. smartpower_change_clock_statistics....349
9.6. smartpower_change_setofpin_statistics....351
9.7. smartpower_commit....352
9.8. smartpower_compute_vectorless....353
9.9. smartpower_create_domain....353
9.10. smartpower_edit_custom_mode....354
9.11. smartpower_edit_scenario....355
9.12. smartpower_export_mpe_report....356
9.13. smartpower_get_temperature....357
9.14. smartpower_get_tetaja....358
9.15. smartpower_get_thermalmode....359
9.16. smartpower_import_vcd....359
9.17. smartpower_init_do....361
9.18. smartpower_initialize_clock_with_constraints....363
9.19. smartpower_init_set_clocks_options....364
9.20. smartpower_init_set_combinational_options....366
9.21. smartpower_init_setofpins_values....367
9.22. smartpower_init_set_othersets_options....368
9.23. smartpower_init_set_primaryinputs_options....369
9.24. smartpower_init_set_registers_options....370
9.25. smartpower_init_set_set_reset_options....371
9.26. smartpower_remove_all_annotations....372
9.27. smartpower_remove_custom_mode....373
9.28. smartpower_remove_domain....374
9.29. smartpower_remove_file....375
9.30. smartpower_remove_pin_frequency....376
9.31. smartpower_remove_pin_of_domain....376
9.32. smartpower_remove_pin_probability....378
9.33. smartpower_remove_scenario....378
9.34. smartpower_report_power....379
9.35. smartpower_report_power_activity_map....385
9.36. smartpower_report_power_history....388
9.37. smartpower_report_power_peak_analyzer....389
9.38. smartpower_report_power_sequencer....392
9.39. smartpower_restore....395
9.40. smartpower_set_cooling....395

© 2025 Microchip Technology Inc. and its subsidiaries

9.41. smartpower_set_default_io_enable_rate....396
9.42. smartpower_set_mode_for_analysis....397
9.43. smartpower_set_mode_for_pdpr....398
9.44. smartpower_set_operating_conditions....399
9.45. smartpower_set_operating_conditions....400
9.46. smartpower_set_pin_frequency....402
9.47. smartpower_set_pin_probability....403
9.48. smartpower_set_preference....404
9.49. smartpower_set_process....405
9.50. smartpower_set_scenario_for_analysis....406
9.51. smartpower_set_tambient....406
9.52. smartpower_set_temperature_opcond....407
9.53. smartpower_set_thermalmode....408
9.54. smartpower_set_voltage_opcond....409
9.55. smartpower_temperature_opcond_set_design_wide....410
9.56. smartpower_temperature_opcond_set_mode_specific....412
9.57. smartpower_voltage_opcond_set_design_wide....413
9.58. smartpower_voltage_opcond_set_mode_specific....415

10. Programming and Configuration Tcl Commands....418
10.1. configure_design_initialization_data....418
10.2. configure_ram....420
10.3. configure_envm....421
10.4. configure_snvm....422
10.5. configure_spiflash....422
10.6. configure_uprom....423
10.7. export_spiflash_image....424
10.8. generate_design_initialization_data....425
10.9. generate_initialization_mem_files....426
10.10. modified_client....427
10.11. nvm_update_serialization_client....428
10.12. nvm_update_storage_client....429
10.13. remove_permanent_locks....430
10.14. select_programmer....431
10.15. set_auto_update_mode....432
10.16. set_manufacturer....432
10.17. set_programming_interface....433
10.18. set_cipher_text_auth_client....434
10.19. set_plain_text_auth_client....435
10.20. set_plain_text_client....436
10.21. set_usk_client....437
10.22. set_client....438
10.23. set_data_storage_client....440
10.24. update_storage_client....441

11. FlashPro Express Tcl Commands....443
11.1. close_project....443
11.2. configure_flashpro3_prg....443
11.3. configure_flashpro4_prg....444

© 2025 Microchip Technology Inc. and its subsidiaries

11.4. configure_flashpro5_prg....446
11.5. configure_flashpro6_prg....446
11.6. create_job_project....448
11.7. auto_construct_job_project....448
11.8. get_connected_programmers....450
11.9. enable_prg....450
11.10. enable_prg_type....451
11.11. export_script....452
11.12. open_project....453
11.13. ping_prg....453
11.14. refresh_prg_list....454
11.15. remove_prg....455
11.16. run_selected_actions....455
11.17. save_log....457
11.18. save_project....457
11.19. scan_chain_prg....458
11.20. self_test_prg....458
11.21. set_fpexpress_mode....459
11.22. set_hsm_params....460
11.23. set_prg_name....461
11.24. set_programming_action....462
11.25. set_programming_file....463
11.26. set_programming_interface....464
11.27. set_spi_flash_action....464
11.28. set_spi_flash_file....465
11.29. update_fp6_programmers....466

12. Configure JTAG Chain Tcl Commands....467
12.1. add_actel_device....467
12.2. select_libero_design_device....468
12.3. copy_device....468
12.4. cut_device....469
12.5. enable_device....470
12.6. remove_device....471
12.7. set_device_name....472
12.8. add_non_actel_device....472
12.9. add_non_actel_device_to_database....473
12.10. construct_chain_automatically....474
12.11. paste_device....475
12.12. remove_non_actel_device_from_database....476
12.13. set_bsdl_file....477
12.14. set_device_ir....478
12.15. set_device_order....479
12.16. set_device_tck....479
12.17. set_device_type....480

13. SmartDebug Tcl Commands....482
13.1. Smart Debug Tcl Commands....482

14. System Builder Tcl Commands....566

© 2025 Microchip Technology Inc. and its subsidiaries

14.1. sb_configure_page....566
14.2. sb_add_core....566
14.3. sb_configure_core....567
14.4. sb_configure_core_count....567
14.5. sb_move_core....568
14.6. sb_enable_core....569
14.7. sb_disable_core....569
14.8. sb_enable_peripheral....570
14.9. sb_disable_peripheral....570
14.10. sb_configure_peripheral....570
14.11. sb_set_fic_direct_mode....571
14.12. sb_configure_envm....571
14.13. open_sb_component....572
14.14. generate_sb_component....572
14.15. close_sb_component....572

15. Simultaneous Switching Noise Analyzer (SSNA) Tcl Commands....573
15.1. ssn_analyzer_noise_report....573
15.2. ssn_analyzer_rerun_analysis....574
15.3. ssn_analyzer_set_dontcare....575
15.4. ssn_analyzer_set_pulse_width....577
15.5. ssn_analyzer_set_static....579
15.6. ssn_analyzer_summary_report....580

16. HSM Tcl Commands....583
16.1. list_all_hsm_tickets....583
16.2. complete_prog_job....583
16.3. get_job_status....584
16.4. process_job_request....585
16.5. remove_hsm_tickets....586
16.6. set_hsm_params....588

17. Derive Constraints Tcl Commands....590
17.1. Working with the derive_constraints Utility....590
17.2. set_device....591
17.3. read_verilog....592
17.4. read_vhdl....593
17.5. set_top_level....594
17.6. read_sdc....594
17.7. read_ndc....595
17.8. derive_constraints....596
17.9. write_sdc....597
17.10. write_pdc....597
17.11. write_ndc....598
17.12. add_include_path....599

18. Revision History....600

Microchip FPGA Support....603

Microchip Information....603

© 2025 Microchip Technology Inc. and its subsidiaries

Trademarks....603
Legal Notice....603
Microchip Devices Code Protection Feature....604

© 2025 Microchip Technology Inc. and its subsidiaries

1. Tcl Scripting Overview (Ask a Question)

Tcl, the Tool Command Language, pronounced `tickle`, is an easy-to-learn scripting language that is compatible with Libero® SoC software. You can run scripts from either the Windows® or Linux command-line or store and run a series of commands in a `*.tcl` batch file.

Libero SoC provides additional capabilities and built-in Tcl Commands:

- Running Tcl Scripts from the Command Line
- Exporting Tcl Scripts
- extended_run_lib
- Tcl Commands, as specified in this document

For complete information on Tcl scripting, refer to one of the books available on this subject. You can also find information about Tcl at web sites such as http://www.tcl.tk.

1.1. Tcl Commands and Supported Families (Ask a Question)

When we specify a family name, we refer to the device family and all its derivatives, unless otherwise specified. See Supported Families in the Tcl command help topics for the families supported for a specific Tcl command.

1.2. Tcl Command Documentation Conventions (Ask a Question)

The following table lists the typographical conventions used for the Tcl command syntax.

Syntax Notation	Description
`command -argument`	Commands and arguments appear in `Courier New` typeface.
`variable`	Variables appear in `Courier New` typeface. You must substitute an appropriate value for the variable.
[-argument value] [variable]+	Optional arguments begin and end with a square bracket with one exception: if the square bracket is followed by a plus sign (+), then users must specify at least one argument. The plus sign (+) indicates that items within the square brackets can be repeated. Do not enter the plus sign character.
#	Comments can be written using a hash (#) sign in the beginning of a text that you want the interpreter to ignore. Comments can be used to provide helping text in your Tcl program.

Important: All Tcl commands are case sensitive. However, their arguments are not.

1.2.1. Examples (Ask a Question)

Syntax for the `get_clocks` command followed by a sample command:

```
get_clocks variable
```

For example, `get_clocks clk1`.

Syntax for the `backannotate` command followed by a sample command:

```
backannotate -name file_name -format format_type -language language -dir directory_name [-
netlist] [-pin]
```

For example, `backannotate -dir {..\design} -name "fanouttest_ba.sdf" -format "SDF" -language "VERILOG" -netlist`.

© 2025 Microchip Technology Inc. and its subsidiaries

1.2.2. Wildcard Characters (Ask a Question)

You can use the following wildcard characters in names used in Tcl commands:

Wildcard	What it Does
\	Interprets the next character literally.
?	Matches any single character.
*	Matches any string.
[]	Matches any single character among those listed between brackets (that is, [A-Z] matches any single character in the A-to-Z range).

Important: The matching function requires that you add a slash (\) before each slash in the port, instance, or net name when using wildcards in a PDC command. For example, if you have an instance named `A/B12` in the netlist, and you enter that name as `A\\/B*` in a PDC command, you will not be able to find it. In this case, you must specify the name as `A\\\\/B*`.

1.2.3. Special Characters [ ], { }, and \ (Ask a Question)

Sometimes square brackets ([ ]) are part of the command syntax. In these cases, you must either enclose the open and closed square brackets characters with curly brackets ({ }) or precede the open and closed square brackets ([ ]) characters with a backslash (\). If you do not, you will get an error message.

Important: Tcl commands are case sensitive. However, their arguments are not.

1.2.4. Entering Arguments on Separate Lines (Ask a Question)

To enter an argument on a separate line, you must enter a backslash (\) character at the end of the preceding line of the command as shown in the following example:

```
backannotate -dir \
{..\design} -name "fanouttest_ba.sdf" -format "SDF" -language "VERILOG" \
-netlist
```

See Also

- Tcl Scripting Overview
- Basic Syntax

1.3. Basic Syntax (Ask a Question)

Tcl scripts contain one or more commands separated by either new lines or semicolons. A Tcl command consists of the name of the command followed by one or more arguments. The format of a Tcl command is:

```
command arg1 ... argN
```

The command in the following example computes the sum of 2 plus 2 and returns the result, 4.

```
expr 2 + 2
```

The `expr` command handles its arguments as an arithmetic expression, computing and returning the result as a string. All Tcl commands return results. If a command has no result to return, it returns an empty string.

© 2025 Microchip Technology Inc. and its subsidiaries

To continue a command on another line, enter a backslash (\) character at the end of the line. For example, the following Tcl command appears on two lines:

```
import -format "edif" -netlist_naming "Generic" -edif_flavor "GENERIC" {prepi.edn}
```

Comments must be preceded by a hash character (#). The comment delimiter (#) must be the first character on a line or the first character following a semicolon, which also indicates the start of a new line. To create a multi-line comment, you must put a hash character (#) at the beginning of each line.

Important: Be sure that the previous line does not end with a continuation character (\). Otherwise, the comment line following it will be ignored.

1.3.1. Special Characters (Ask a Question)

Square brackets ([ ]) are special characters in Tcl. To use square brackets in names such as port names, you must either enclose the entire port name in curly braces, or lead the square brackets with a slash (\) character.

1.3.2. Sample Tcl Script (Ask a Question)

```
#Create a new project and set up a new design
new_project -location {D:/2Work/my_pf_proj} -name {my_pf_proj} -project_description {}\
-block_mode 0 -standalone_peripheral_initialization 0 -use_enhanced_constraint_flow 1\
-hdl {VERILOG} -family {PolarFire} -die {MPF300TS_ES} -package {FCG1152} -speed {-1} \
-die_voltage {1.0} -part_range {EXT} -adv_options {IO_DEFT_STD:LVCMOS 1.8V}\
-adv_options {RESTRICTPROBEPINS:1} -adv_options {RESTRICTSPIPINS:0}\
-adv_options {SYSTEM_CONTROLLER_SUSPEND_MODE:1} -adv_options {TEMPR:EXT} \
-adv_options {VCCI_1.2_VOLTR:EXT} -adv_options {VCCI_1.5_VOLTR:EXT} \
-adv_options {VCCI_1.8_VOLTR:EXT} -adv_options {VCCI_2.5_VOLTR:EXT} \
-adv_options {VCCI_3.3_VOLTR:EXT} -adv_options {VOLTR:EXT}
#Import HDL source file
import_files -convert_EDN_to_HDL 0 -hdl_source {C:/test/prep1.v}
#Import HDL stimulus file
import_files -convert_EDN_to_HDL 0 -stimulus {C:/test/prep1tb.v}
#set the top level design name
set_root -module {prep1::work}
#Associate SDC constraint file to Place and Route tool
organize_tool_files -tool {PLACEROUTE} -file {D:/2Work/my_pf_proj/constraint/user.sdc} \
-module {prep1::work} -input_type {constraint}
#Associate SDC constraint file to Verify Timing tool
organize_tool_files -tool {VERIFYTIMING} -file {D:/2Work/my_pf_proj/constraint/user.sdc}\
-module {prep1::work} -input_type {constraint}
#Run synthesize
run_tool -name {SYNTHESIZE}
#Configure Place and Route tool
configure_tool -name {PLACEROUTE} -params {DELAY_ANALYSIS:MAX} -params {EFFORT_LEVEL:false}\
-params {INCRPLACEANDROUTE:false} -params {MULTI_PASS_CRITERIA:VIOLATIONS}\
-params {MULTI_PASS_LAYOUT:false} -params {NUM_MULTI_PASSES:5} -params {PDPR:false}\
-params {RANDOM_SEED:0} -params {REPAIR_MIN_DELAY:false} -params {SLACK_CRITERIA:WORST_SLACK}
\
-params {SPECIFIC_CLOCK:} -params {START_SEED_INDEX:1} -params {STOP_ON_FIRST_PASS:false}\
-params {TDPR:true}
#Run Place and Route
run_tool -name {PLACEROUTE}
#Configure Timing Report Generation
configure_tool -name {VERIFYTIMING} -run_tool -name {PLACEROUTE}params
{CONSTRAINTS_COVERAGE:1}\
-params {FORMAT:XML} -params {MAX_TIMING_FAST_HV_LT:0} -params {MAX_TIMING_SLOW_LV_HT:1}\
-params {MAX_TIMING_SLOW_LV_LT:0} -params {MAX_TIMING_VIOLATIONS_FAST_HV_LT:0} \
-params {MAX_TIMING_VIOLATIONS_SLOW_LV_HT:1} -params {MAX_TIMING_VIOLATIONS_SLOW_LV_LT:0}\
-params {MIN_TIMING_FAST_HV_LT:1} -params {MIN_TIMING_SLOW_LV_HT:0} -params
{MIN_TIMING_SLOW_LV_LT:0} -params {MIN_TIMING_VIOLATIONS_FAST_HV_LT:1} -params
{MIN_TIMING_VIOLATIONS_SLOW_LV_HT:0} \
-params {MIN_TIMING_VIOLATIONS_SLOW_LV_LT:0}
#Run Verify Timing tool
run_tool -name {VERIFYTIMING}
#Run Power Verification tool
run_tool -name {VERIFYPOWER} #Export bitstream
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
export_bitstream_file -file_name {prep1} \
-export_dir {D:\2Work\my_pf_proj\designer\prep1\export} -format {STP} -master_file 0 \
-master_file_components {} -encrypted_uek1_file 0 -encrypted_uek1_file_components {} \
-encrypted_uek2_file 0 -encrypted_uek2_file_components {} \
-trusted_facility_file 1 -trusted_facility_file_components {FABRIC}
```

1.4. Types of Tcl Commands (Ask a Question)

This section describes the following types of Tcl commands:

- Built-in Commands
- Procedures Created with the proc Command

1.4.1. Built-in Commands (Ask a Question)

Built-in commands are provided by the Tcl interpreter. They are available in all Tcl applications.

- Tcl provides several commands for manipulating file names, reading and writing file attributes, copying files, deleting files, creating directories, and so on.
- You can execute an external program using `exec`. Upon execution, the return value is the output (on stdout) from the external program, for example:

```
set tmp [ exec myprog ] puts stdout $tmp
```

- You can easily create collections of values (lists) and manipulate them in a variety of ways.
- You can create arrays - structured values consisting of name-value pairs with arbitrary string values for the names and values.
- You can manipulate the time and date variables.
- You can write scripts that can wait for certain events to occur, such as an elapsed time or the availability of input data on a network socket.

1.4.2. Procedures Created with the proc Command (Ask a Question)

You use the `proc` command to declare a procedure. You can then use the name of the procedure as a Tcl command.

The following sample script consists of a single command named `proc`. The `proc` command takes three arguments:

- The name of a procedure (`myproc`)
- A list of argument names (`arg1 arg2`)
- The body of the procedure, which is a Tcl script

```
proc myproc { arg1 arg2 } {
# procedure body
}
myproc a b
```

1.5. Variables (Ask a Question)

With Tcl scripting, you can store a value in a variable for later use. You use the set command to assign variables. For example, the following set command creates a variable named x and sets its initial value to 10.

```
set x 10
```

A variable can be a letter, a digit, an underscore, or any combination of letters, digits, and underscore characters. All variable values are stored as strings.

© 2025 Microchip Technology Inc. and its subsidiaries

In the Tcl language, you do not declare variables or their types. Any variable can hold any value. Use the dollar sign ($) to obtain the value of a variable, for example:

```
set a 1
set b $a
set cmd expr
set x 11
$cmd $x*$x
```

The dollar sign $ tells Tcl to handle the letters and digits following it as a variable name and to substitute the variable name with its value.

Global Variables

Variables can be declared global in scope using the Tcl global command. All procedures, including the declaration can access and modify global variables, for example:

```
global myvar
```

1.6. Command Substitution (Ask a Question)

By using square brackets ([]), you can substitute the result of one command as an argument to a subsequent command, as shown in the following example:

```
set a 12
set b [expr $a*4]
```

Tcl handles everything between square brackets as a nested Tcl command. Tcl evaluates the nested command and substitutes its result in place of the bracketed text. In the example above, the argument that appears in square brackets in the second set command is equal to 48 (that is, 12* 4 = 48).

Conceptually,

```
set b [expr $a * 4]
```

expands to

```
set b [expr 12 * 4 ]
```

and then to

```
set b 48
```

1.7. Quotes and Braces (Ask a Question)

The distinction between braces ({ }) and quotes (" ") is significant when the list contains references to variables. When references are enclosed in quotes, they are substituted with values. However, when references are enclosed in braces, they are not substituted with values.

The following table lists an example code.

With Braces	With Double Quotes
set b 2	set b 2
set t { 1 $b 3 }	set t " 1 $b 3 "
set s { [expr $b + $b] }	set s " [expr $b + $b] "
puts stdout $t	puts stdout $t
puts stdout $s	puts stdout $s

© 2025 Microchip Technology Inc. and its subsidiaries

The above example code will generate the following output:

```
1 $b 3 vs. 1 2 3
[ expr $b + $b ] 4
```

1.8. Filenames (Ask a Question)

In Tcl syntax, filenames should be enclosed in braces { } to avoid backslash substitution and white space separation. Backslashes are used to separate folder names in Windows-based filenames. The problem is that sequences of “\n” or “\t” are interpreted specially. Using the braces disables this special interpretation and specifies that the Tcl interpreter handle the enclosed string literally. Alternatively, double-backslash “\\n” and “\\t” would work as well as forward slash directory separators “/n” and “/t”. For example, to specify a file on your Windows PC at `c:\newfiles\thisfile.adb`, use one of the following:

```
{C:\newfiles\thisfile.adb}
C:\\newfiles\\thisfile.adb
"C:\\newfiles\\thisfile.adb"
C:/newfiles/thisfile.adb
"C:/newfiles/thisfile.adb"
```

If there is white space in the filename path, you must use either the braces or double-quotes. For example:

```
C:\program data\thisfile.adb
```

should be referenced in Tcl script as

```
{C:\program data\thisfile.adb} or "C:\\program data\\thisfile.adb"
```

If you are using variables, you cannot use braces { } because, by default, the braces turn off all special interpretation, including the dollar sign character. Instead, use either double-backslashes or forward slashes with double quotes. For example:

```
"$design_name.adb"
```

Important: To use a name with special characters such as square brackets [ ], you must put the entire name between curly braces { } or put a slash character \ immediately before each square bracket.

1.9. Lists and Arrays (Ask a Question)

A list is a way to group data and handle the group as a single entity. To define a list, use curly braces { } and double quotes “ ”. For example, the following set command {1 2 3 }, when followed by the list command, creates a list stored in the variable "a". This list will contain the items "1," "2," and "3".

```
set a { 1 2 3 }
```

Here is another example:

```
set e 2
set f 3
set a [ list b c d [ expr $e + $f ] ] puts $a
```

displays (or outputs):

```
b c d 5
```


© 2025 Microchip Technology Inc. and its subsidiaries

Tcl supports many other list-related commands such as `lindex`, `linsert`, `llength`, `lrange`, and `lappend`.

1.9.1. Arrays (Ask a Question)

An array is a method to group data. Arrays are collections of items stored in variables. Each item has a unique address that you use to access it. You do not need to declare them or specify their size.

Array elements are managed similarly to other Tcl variables. You create them with the `set` command, and you can use the dollar sign ($) to access their values.

```
set myarray(0) "Zero"
set myarray(1) "One"
set myarray(2) "Two"

for {set i 0} {$i < 3} {incr i 1} {\
    puts stdout $myarray($i)}
```

Output:

```
Zero One Two
```

In the preceding example, an array called `myarray` is created using the `set` statement, which assigns a value to its first element. The for-loop statement prints out the value stored in each element of the array.

1.9.2. Special Arguments (Command-Line Parameters) (Ask a Question)

You can determine the name of the Tcl script file while executing the Tcl script by referring to the `$argv0` variable.

```
puts “Executing file $argv0”
```

To access other arguments from the command line, you can use the `lindex` command and the `argv` variable: To read the Tcl file name:

```
lindex $argv 0
```

To read the first passed argument:

```
lindex $argv 1
```

For example:

```
puts "Script name is $argv0" ; # accessing the scriptname puts "first argument is [lindex
$argv 0]"
puts "second argument is [lindex $argv 1]" puts "third argument is [lindex $argv 2]" puts
"number of argument is [llength $argv]" set des_name [lindex $argv 0]
puts "Design name is $des_name"
```

1.10. Control Structures (Ask a Question)

Tcl control structures are commands that change the flow of execution through a script. These control structures include commands for conditional execution (if-then-elseif-else) and looping (while, for, catch).

An "if" statement only executes the body of the statement (enclosed between curly braces) if the Boolean condition is found to be true.

1.10.1. if/else Statements (Ask a Question)

```
if { “$name” == “paul” } then {
…
# body if name is paul
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
} elseif { $code == 0 } then {
…
# body if name is not paul and if value of variable code is zero
} else {
…
# body if above conditions is not true
}
```

1.10.2. for Loop Statement (Ask a Question)

A "for" statement will repeatedly execute the body of the code as long as the index is within a specified limit.

```
for { set i 0 } { $i < 5 } { incr i } {
…
# body here
}
```

1.10.3. while Loop Statement (Ask a Question)

A "while" statement will repeatedly execute the body of the code (enclosed between the curly braces) as long as the Boolean condition is found to be true.

```
while { $p > 0 } {
…
}
```

1.10.4. catch Statement (Ask a Question)

A "catch" statement suspends normal error handling on the enclosed Tcl command. If a variable name is also used, then the return value of the enclosed Tcl command is stored in the variable.

```
catch { open "$inputFile" r } myresult
```

1.11. Print Statement and Return Values (Ask a Question)

1.11.1. Print Statement (Ask a Question)

Use the puts command to write a string to an output channel. Predefined output channels are "stdout" and "stderr." If you do not specify a channel, then puts display text to the stdout channel.

Note: The STDIN Tcl command is not supported by Microchip SoC tools.

For example:

```
set a [ myprog arg1 arg2 ]
puts "the answer from myprog was $a (this text is on stdout)"
puts stdout "this text also is on stdout"
```

1.11.2. Return Values (Ask a Question)

The return code of a Tcl command is a string. You can use a return value as an argument to another function by enclosing the command with square brackets [ ]. For example:

```
set a [ prog arg1 arg2 ] exec $a
```

The Tcl command "exec" will run an external program. The return value of "exec" is the output (on stdout) from the program. For example:

```
set tmp [ exec myprog ] puts stdout $tmp
```


© 2025 Microchip Technology Inc. and its subsidiaries

1.12. Running Tcl Scripts from the Command Line (Ask a Question)

You can run Tcl scripts from your Windows or Linux command line as well as pass arguments to scripts from the command line.

Note: Tcl commands in this section are not case-sensitive.

To execute a Tcl script file in the Libero SoC Project Manager software from a shell command line:

At the prompt, type the path to the Microchip SoC software installation location followed by the word "SCRIPT" and a colon, and then the name of the script file as follows:

- For Linux: `<location of Libero SoC software installation>/bin/libero script:<filename>.tcl`. For example, to run the Tcl script file "myscript.tcl", type:

```
C:\libero\bin\libero script:myscript.tcl
```

- For Windows: `<location of Libero SoC software installation>\Designer\bin\libero.exe script:<filename>.tcl`. Where `<location of Microchip SoC software>` is the root directory in which you installed the Microchip SoC software, and `<filename>` is the name, including a relative or absolute path, of the Tcl script file to execute. For example, to run the Tcl script file "myscript.tcl", type:

```
C:/Microchip/Libero/Designer/bin/libero.exe script:myscript.tcl
```

To pass arguments from the command line to your Tcl script file:

At the prompt, type the path to the Microchip SoC software installation location followed by the SCRIPT argument:

- For Linux: <location of Microchip SoC software>\bin\libero “SCRIPT_ARGS:<filename arg1 arg2 ...>”. For example:

```
C:\libero\bin\libero SCRIPT:myscript.tcl “SCRIPT_ARGS:one two three”
```

- For Windows: <location of Microchip SoC software>/Designer/bin/libero.exe “SCRIPT_ARGS:<filename arg1 arg2 ...>”. Where <location of Microchip SoC software> is the root directory in which you installed the Microchip SoC software, and “SCRIPT_ARGS:<filename arg1 arg2 ...>” is the name, including a relative or absolute path, of the Tcl script file and arguments you are passing to the script file. For example:

```
C:/Microchip/Libero/Designer/bin/libero.exe SCRIPT:myscript.tcl “SCRIPT_ARGS:one two three”
```

To obtain the output from the log file:

At the prompt, type the path to the Microchip SoC software installation location followed by the SCRIPT, SCRIPT_ARGS and LOGFILE arguments.

```
<location of Microchip SoC software> SCRIPT:<filename>.tcl “SCRIPT_ARGS:a b c”
LOGFILE:<output.log>
```

Where:

- `location of Microchip SoC software` is the root directory in which you installed the Microchip SoC software.
- `SCRIPT:<filename>.tcl` is the name, including a relative or absolute path, of the Tcl script file.
- `SCRIPT_ARGS` are the arguments you are passing to the script file.
- `output.log` is the name of the log file.

© 2025 Microchip Technology Inc. and its subsidiaries

For example:

```
C:\libero\designer\bin\libero SCRIPT:testTclparam.tcl "SCRIPT_ARGS:a b c"
LOGFILE:testTclparam.log
```

1.13. Exporting Tcl Scripts (Ask a Question)

You can write out a Tcl script file that contains the commands executed in the current session. You can then use this exported Tcl script to re-execute the same commands interactively or in batch. You can also use this exported script to become more familiar with Tcl syntax.

You can export Tcl scripts from the Project Manager.

To export a Tcl session script from the Project Manager:

1. From the **File** menu, choose **Export Script File**. The **Export Script** dialog box appears.
2. Click **OK**. The **Script Export Options** dialog box appears:
3. Check the **Include Commands from Current Design [Project] Only** checkbox. This option applies only if you opened more than one design or project in your current session. If so, and you do not check this box, Project Manager exports all commands from your current session.
4. Select the radio button for the appropriate filename formatting. To export filenames relative to the current working directory, select **Relative filenames (default)** formatting. To export filenames that include a fully specified path, select **Qualified filenames (full path; including directory name)** formatting.
5. Choose **Relative filenames** if you do not intend to move the Tcl script from the saved location, or **Qualified filenames** if you plan to move the Tcl script to another directory or machine.
6. Click **OK**. Project Manager saves the Tcl script with the specified filename.

   Notes:
   - When exporting Tcl scripts, Project Manager always encloses filenames in curly braces to ensure portability.
   - Libero SoC software does not write out any Tcl variables or flow-control statements to the exported Tcl file, even if you had executed the design commands using your own Tcl script. The exported Tcl file only contains the tool commands and their accompanying arguments.

1.14. extended_run_lib (Ask a Question)

Note: This is not a Tcl command; it is a shell script that can be run from the command-line.

The `extended_run_lib` Tcl script enables you to run the multiple pass layout in batch mode from a command-line.

```
$ACTEL_SW_DIR/bin/libero script:$ACTEL_SW_DIR/scripts/extended_run_lib.tcl
logfile:extended_run.log "script_args:-root path/designer/module_name [-n numPasses] [-
starting_seed_index numIndex] [-compare_criteria value] [-c clockName] [-analysis value] [-
slack_criteria value] [-stop_on_success] [-timing_driven|-standard] [-power_driven value] [-
placer_high_effort value]"
```

Note: There is no option to save the design files from all the passes. Only the (Timing or Power) result reports from all the passes are saved.

Arguments

Parameter	Value	Description
-root	path/designer/module_name	The path to the root module located under the designer directory of the Libero project.
[-n]	numPasses	Sets the number of passes to run. The default number of passes is 5.

© 2025 Microchip Technology Inc. and its subsidiaries

extended_run_lib (continued)

Parameter	Value	Description
[-starting_seed_index]	numIndex	Indicates the specific index into the array of random seeds which is to be the starting point for the passes. Value may range from 1 to 100. If not specified, the default behavior is to continue from the last seed index that was used.
[-compare_criteria]	value	**Value** / **Description** frequency: Use clock frequency as criteria for comparing the results between passes. This option can be used in conjunction with the -c option (described below). violations: Use timing violations as criteria for comparing the results between passes. This option can be used in conjunction with the -analysis, -slack_criteria and - stop_on_success options (described below). power: Use total power as criteria for comparing the results between passes, where lowest total power is the goal.
[-c]	clockName	Applies only when the clock frequency comparison criteria is used. Specifies the particular clock that is to be examined. If no clock is specified, then the slowest clock frequency in the design in a given pass is used. The clock name should match with one of the Clock Domains in the Summary section of the Timing report.
[-analysis]	value	Applies only when the timing violations comparison criteria is used. Specifies the type of timing violations (the slack) to examine. The following table shows the acceptable values for this argument: **Value** / **Description** max: Examines timing violations (slack) obtained from maximum delay analysis. This is the default. min: Examines timing violations (slack) obtained from minimum delay analysis.
[-slack_criteria]	value	Applies only when the timing violations comparison criteria is used. Specifies how to evaluate the timing violations (slack). The type of timing violations (slack) is determined by the -analysis option. The following table shows the acceptable values for this argument: **Value** / **Description** worst: Sets the timing violations criteria to Worst slack. For each pass obtains the most amount of negative slack (or least amount of positive slack if all constraints are met) from the timing violations report. The largest value out of all passes will determine the best pass. This is the default. tns: Sets the timing violations criteria to Total Negative Slack (tns). For each pass it obtains the sum of negative slack values from the first 100 paths from the timing violations report. The largest value out of all passes determines the best pass. If no negative slacks exist for a pass, then the worst slack is used to evaluate that pass.
[-stop_on_success]		Applies only when the timing violations comparison criteria is used. The type of timing violations (slack) is determined by the `-analysis` option. Stops running the remaining passes if all timing constraints have been met (when there are no negative slacks reported in the timing violations report).
[-timing_driven|-standard]		Sets layout mode to timing driven or standard (non-timing driven). The default is `-timing_driven` or the mode used in the previous layout command.

© 2025 Microchip Technology Inc. and its subsidiaries

extended_run_lib (continued)

Parameter	Value	Description
[-power_driven]	value	Enables or disables power-driven layout. The default is off or the mode used in the previous layout command. The following table shows the acceptable values for this argument: Value: off — Does not run power-driven layout. Value: on — Enables power-driven layout.
[-placer_high_effort]	value	Sets placer effort level. The default is off or the mode used in the previous layout command. The following table shows the acceptable values for this argument: Value: off — Runs layout in regular effort. Value: on — Activates high effort layout mode.

Returns

A nonzero value will be returned on error.

1.15. Sample Tcl Script - Project Manager (Ask a Question)

The following Tcl commands create a new project and set your project options.

```
new_project -location {D:/2Work/my_pf_proj} -name {my_pf_proj} -project_description {}\
-block_mode 0 -standalone_peripheral_initialization 0 -use_enhanced_constraint_flow 1\
-hdl {VERILOG} -family {PolarFire} -die {MPF300TS_ES} -package {FCG1152} -speed {-1} \
-die_voltage {1.0} -part_range {EXT} -adv_options {IO_DEFT_STD:LVCMOS 1.8V}\
-adv_options {RESTRICTPROBEPINS:1} -adv_options {RESTRICTSPIPINS:0}\
-adv_options {SYSTEM_CONTROLLER_SUSPEND_MODE:1} -adv_options {TEMPR:EXT} \
-adv_options {VCCI_1.2_VOLTR:EXT} -adv_options {VCCI_1.5_VOLTR:EXT} \
-adv_options {VCCI_1.8_VOLTR:EXT} -adv_options {VCCI_2.5_VOLTR:EXT} \
-adv_options {VCCI_3.3_VOLTR:EXT} -adv_options {VOLTR:EXT}
#Import HDL source file
import_files -convert_EDN_to_HDL 0 -hdl_source {C:/test/prep1.v}
#Import HDL stimulus file
import_files -convert_EDN_to_HDL 0 -stimulus {C:/test/prep1tb.v}
#set the top level design name
set_root -module {prep1::work}
#Associate SDC constraint file to Place and Route tool
organize_tool_files -tool {PLACEROUTE} -file {D:/2Work/my_pf_proj/constraint/user.sdc} \
-module {prep1::work} -input_type {constraint}
#Associate SDC constraint file to Verify Timing tool
organize_tool_files -tool {VERIFYTIMING} -file {D:/2Work/my_pf_proj/constraint/user.sdc}\
-module {prep1::work} -input_type {constraint}
#Run synthesize
run_tool -name {SYNTHESIZE}
#Configure Place and Route tool
configure_tool -name {PLACEROUTE} -params {DELAY_ANALYSIS:MAX} -params {EFFORT_LEVEL:false}\
-params {INCRPLACEANDROUTE:false} -params {MULTI_PASS_CRITERIA:VIOLATIONS}\
-params {MULTI_PASS_LAYOUT:false} -params {NUM_MULTI_PASSES:5} -params {PDPR:false}\
-params {RANDOM_SEED:0} -params {REPAIR_MIN_DELAY:false} -params {SLACK_CRITERIA:WORST_SLACK}
\
-params {SPECIFIC_CLOCK:} -params {START_SEED_INDEX:1} -params {STOP_ON_FIRST_PASS:false}\
-params {TDPR:true}
#Run Place and Route
run_tool -name {PLACEROUTE}
#Configure Timing Report Generation
configure_tool -name {VERIFYTIMING} -run_tool -name {PLACEROUTE}params
{CONSTRAINTS_COVERAGE:1}\
-params {FORMAT:XML} -params {MAX_TIMING_FAST_HV_LT:0} -params {MAX_TIMING_SLOW_LV_HT:1} \
-params {MAX_TIMING_SLOW_LV_LT:0} -params {MAX_TIMING_VIOLATIONS_FAST_HV_LT:0} \
-params {MAX_TIMING_VIOLATIONS_SLOW_LV_HT:1} -params {MAX_TIMING_VIOLATIONS_SLOW_LV_LT:0}\
-params {MIN_TIMING_FAST_HV_LT:1} -params {MIN_TIMING_SLOW_LV_HT:0} -params
{MIN_TIMING_SLOW_LV_LT:0} -params {MIN_TIMING_VIOLATIONS_FAST_HV_LT:1} -params
{MIN_TIMING_VIOLATIONS_SLOW_LV_HT:0} \
-params {MIN_TIMING_VIOLATIONS_SLOW_LV_LT:0}
#Run Verify Timing tool
run_tool -name {VERIFYTIMING}
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
#Run Power Verification tool
run_tool -name {VERIFYPOWER}
#Export bitstream
export_bitstream_file -file_name {prep1} \
-export_dir {D:\2Work\my_pf_proj\designer\prep1\export} -format {STP} -master_file 0 \
-master_file_components {} -encrypted_uek1_file 0 -encrypted_uek1_file_components {} \
-encrypted_uek2_file 0 -encrypted_uek2_file_components {} \
-trusted_facility_file 1 -trusted_facility_file_components {FABRIC}
```

1.16. How to Derive Required Part Information from A "Part Number" (Ask a Question)

In order to use Tcl Commands such as `set_device` or a new design; certain part information items must be specified. Many of these items can be derived from the "Part Number" you have chosen. For example, suppose the Part Number is: **MPF300XT-1FCG784I**

Table 1-1. Part Information for MPF300XT-1FCG784I

Part Information	Description	Example
`-family <family name>`	The `<family name>` usually known	`-family {PolarFire}`
`-die <die name>`	Derive this information from the Part Number, the characters before the "-"	**MPF300XT**-1FCG784I `-die {MPF300XT}`
`-speed <speed grade>`	If there is a digit immediately after the "-", -<digit> will be the <speed grade> value (preceded by a "-"). **Note:** If there is no digit, the default speed grade is STD.	MPF300XT-**1**FCG784 `-speed {-1}`
`-package <package name>`	The next sequence of letters, followed by a sequence of digits will constitute the package type and "size". **Note:** If there is a trailing letter after the <digits>; this letter is **not** part of the <package name>; but is rather part of the <part range>	For PolarFire®, this combination will simply constitute the <package name>. MPF300XT- 1**FCG784**I `-package {FCG784}`
`-part_range <part range>`	The last letter (if any) will indicate the `<part_range>`. • I: IND • E: EXT • M: MIL • <none>: COM	MPF300XT- 1FCG784**I** `-part_range {IND}`

© 2025 Microchip Technology Inc. and its subsidiaries

2. Building a Libero Design Using Tcl (Ask a Question)

Microchip's Libero SoC Design Suite enables the rapid development of FPGA designs by offering a comprehensive set of industry-standard third-party tools. It also offers a rich IP library for various applications such as Communications, Networking, Imaging and Video, Industrial, Defense, and Aerospace.

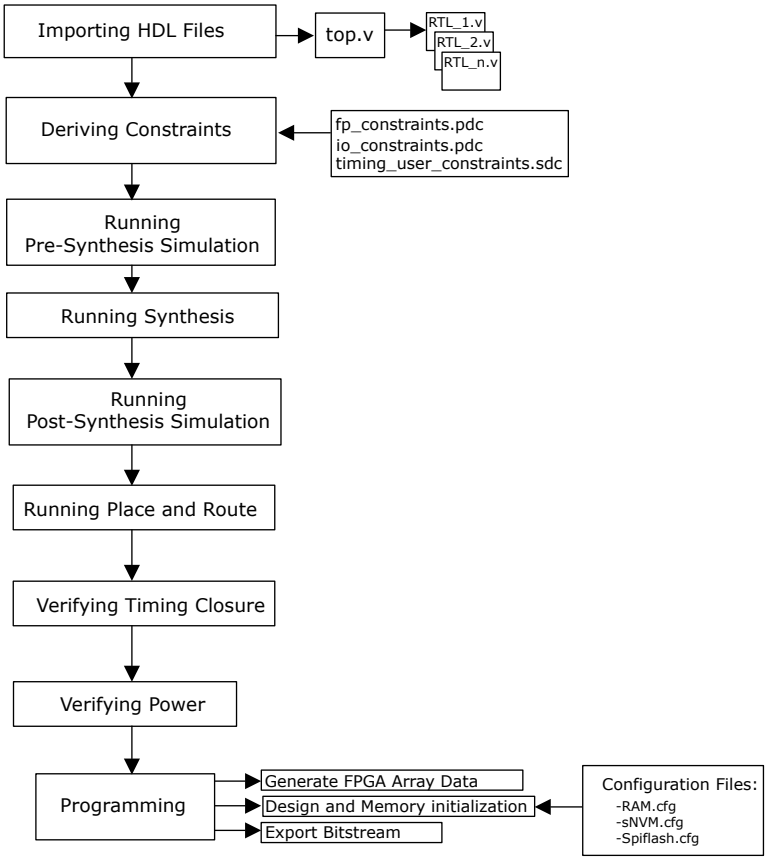
Tcl commands help facilitate the end-to-end design flow automation. Libero SoC supports Tcl scripting, which facilitates the automation of end-to-end design flow from HDL to programming. This chapter describes how the HDL to programming flow is executed in the Libero SoC Design Suite using Tcl, how to run the Tcl, and how to update a project with the updated HDL and constraints files.

2.1. HDL to Programming Flow (Ask a Question)

This section describes how the HDL to programming flow is executed in the Libero SoC Design Suite using Tcl. The following figure shows the Libero SoC HDL to programming flow.

As a best practice, the following resources are recommended in the top-level Tcl folder for building a Libero SoC project.

Figure 2-1. HDL to Programming Flow



- `parameters.tcl`: This Tcl file contains project variables as shown in the following snippet.

```
set prj_family      "PolarFire";
set prj_die         "MPF100T";
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
set prj_package    "FCG484";
set prj_speed      "-1";
set prj_root       "top";
```

Tip: Libero SoC tools profile can also be configured in the `parameters.tcl` file.

- `src`: This folder contains all the required HDL, constraints, memory configuration, and stimulus source files required for a project.
- `design.tcl`: This Tcl file imports all the HDL, constraints, memory configuration, and stimulus sources files. The following snippet shows a sample `design.tcl` file.

```
import_files -hdl_source {./src/rtl_1.v} -library {work};
import_files -hdl_source {./src/rtl2.v} -library {work};
.
.
import_files -hdl_source {./src/top.v} -library {work};
import_files -sdc {./src/constraint/timing_user_constraints.sdc};
import_files -pdc {./src/constraint/io_constraints.pdc};
import_files -pdc {./src/constraint/fp_constraints.pdc};
import_files -stimulus {./src/stimulus/testbench_presynth.v} -library {stimulus};
import_files -stimulus {./src/stimulus/testbench_postsynth.v} -library {stimulus};
organize_tool_files -tool {SYNTHESIZE} \
 -file ./src/constraint/timing_user_constraints.sdc \
 -module {$prj_root} \
 -input_type {constraint}
organize_tool_files -tool {PLACEROUTE} \
 -file ./src/constraint/timing_user_constraints.sdc \
 -file ./src/constraint/io_constraints.pdc \
 -file ./src/constraint/fp_constraints.pdc \
 -module {$prj_root} \
 -input_type {constraint}
organize_tool_files -tool {VERIFYTIMING} \
 -file ./src/constraint/timing_user_constraints.sdc \
 -module {$prj_root} \
 -input_type {constraint}
organize_tool_files \
-tool {SIM_PRESYNTH} \
-file {./src/stimulus/testbench_presynth.v}

organize_tool_files \
-tool {SIM_POSTSYNTH} \
-file {./src/stimulus/testbench_postsynth.v}
configure_ram -cfg_file {./src/src_cfg/RAM.cfg}
configure_tool \
         -name {GENERATEPROGRAMMINGFILE} \
         -params {program_fabric:true} \
         -params {program_security:false} \
     -params {sanitize_snvm:false}
associate_stimulus -file testbench.v -mode new -module stimulus
```

- `libero.tcl`: This Tcl file executes the HDL to programming flow in Libero SoC.

Tip: In the `design.tcl` file, parameters for all the Libero SoC tools can be configured using the configure_tool command.

2.1.1. Importing HDL Files (Ask a Question)

In this step, the following tasks are being executed:

1. Creating a Libero SoC project.
2. Importing all the HDL, constraints, and stimulus source files that are required for the project.
3. Building the design hierarchy by defining the top module.

© 2025 Microchip Technology Inc. and its subsidiaries

In `libero.tcl`, use the following commands to execute these tasks.

```
source ./parameters.tcl;

set libero_cmd "new_project \
                -location {./exprj} -name {exprj} \
                -family {$prj_family} -die {$prj_die} -package {$prj_package} \
                -speed {$prj_speed} \
                -hdl {VERILOG}";

eval file delete -force ./exprj;

eval $libero_cmd;

source ./design.tcl;

build_design_hierarchy;

set_root $prj_root;
```

For more information about HDL commands, see the HDL Tcl Commands.

2.1.2. Deriving Constraints (Ask a Question)

In this step, timing constraints are derived. In `libero.tcl`, use the following command to derive timing constraints.

```
derive_constraints_sdc
```

For more information about applying constraints, see the Derive Constraints Tcl Commands.

2.1.3. Running Pre-Synthesis Simulation (Ask a Question)

In this step, the functionality of the HDL is verified before Synthesis using a test bench. In `libero.tcl`, use the following command to run pre-synthesis simulation.

```
run_tool -name {SIM_PRESYNTH}
```

As a best practice, use this command with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {SIM_PRESYNTH}  }] } {
      puts "SIM_PRESYNTH FAILED \n"
   } else {
      puts "SIM_PRESYNTH PASSED \n"
   }
```

For more information about adding pre-synthesis simulation parameters, see the run_tool command.

2.1.4. Running Synthesis (Ask a Question)

In this step, HDL is synthesized according to constraint files. In `libero.tcl`, use the following command to run Synthesis.

```
run_tool -name {SYNTHESIZE}
```

Use this command with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {SYNTHESIZE}  }] } {
      puts "SYNTHESIZE FAILED \n"
   } else {
      puts "SYNTHESIZE PASSED \n"
   }
```

For more information about adding Synthesis parameters, see the run_tool command.

© 2025 Microchip Technology Inc. and its subsidiaries

2.1.5. Running Post-Synthesis Simulation (Ask a Question)

In this step, the functionality of the HDL is verified after Synthesis using a test bench. In `libero.tcl`, use the following command to run post-synthesis simulation.

```
run_tool -name {SIM_POSTSYNTH}
```

Use this command with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {SYNTHESIZE}  }] } {
      puts "SIM_POSTSYNTH FAILED \n"
   } else {
      puts "SIM_POSTSYNTH PASSED \n"
   }
```

For more information about adding Synthesis parameters, see the run_tool command.

2.1.6. Running Place and Route (Ask a Question)

In this step, HDL is placed and routed according to constraint files. In `libero.tcl`, use the following command to run Place and Route.

```
run_tool -name {PLACEROUTE}
```

Use this command with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {PLACEROUTE}  }] } {
      puts "PLACEROUTE FAILED \n"
   } else {
      puts "PLACEROUTE PASSED \n"
   }
```

For more information about adding Place and Route parameters, see the run_tool command.

2.1.7. Verifying Timing Closure (Ask a Question)

In this step, HDL is verified for timing closure. In `libero.tcl`, use the following command to run Timing Verification.

```
run_tool -name {VERIFYTIMING}
```

Use this command with the catch statement as shown in the following snippet.

```
if {[catch {run_tool -name {VERIFYTIMING}  }] } {
      puts "VERIFYTIMING FAILED \n"
   } else {
      puts "VERIFYTIMING PASSED \n"
   }
```

For more information about adding Verify Timing parameters, see the run_tool command.

2.1.8. Verifying Power (Ask a Question)

In this step, power verification is executed before generating the programming bit stream. In `libero.tcl`, use the following command for verifying the power consumption.

```
run_tool -name {VERIFYPOWER}
```

Use this command with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {VERIFYPOWER}  }] } {
      puts "VERIFYPOWER FAILED \n"
   } else {
      puts "VERIFYPOWER PASSED \n"
   }
```


© 2025 Microchip Technology Inc. and its subsidiaries

For more information about adding Verify Power parameters, see the run_tool command.

2.1.9. Programming (Ask a Question)

In this step, the following tasks are executed:

1. Generating the FPGA Array Data
2. Configuring memories
3. Generating initialization clients for configured memories
4. Generating design initialization data
5. Generating and exporting the programming bit stream (`.stp` and `.job`)

In `libero.tcl`, use the following commands to execute these tasks.

```
run_tool -name {GENERATEPROGRAMMINGDATA}
generate_design_initialization_data
run_tool -name {GENERATEPROGRAMMINGFILE}

export_bitstream_file \
         -file_name {test} \
         -format {STP DAT PPD } \
         -limit_SVF_file_size 0 \
         -limit_SVF_file_by_max_filesize_or_vectors {SIZE} \
         -svf_max_filesize {1024} \
         -svf_max_vectors {1000} \
         -trusted_facility_file 1 \
         -trusted_facility_file_components {FABRIC SNVM} \
         -zeroization_likenew_action 0 \
         -zeroization_unrecoverable_action 0

export_prog_job \
         -job_file_name {test} \
         -bitstream_file_type {TRUSTED_FACILITY} \
         -bitstream_file_components {FABRIC SNVM} \
         -zeroization_likenew_action 0 \
         -zeroization_unrecoverable_action 0 \
         -program_design 1 \
         -program_spi_flash 0 \
         -design_bitstream_format {PPD}

close_project -save 1
```

Use these commands with the `catch` statement as shown in the following snippet.

```
if {[catch {run_tool -name {GENERATEPROGRAMMINGDATA}  }] } {
      puts "GENERATEPROGRAMMINGDATA FAILED \n"
   } else {
      puts "GENERATEPROGRAMMINGDATA PASSED \n"
   }

if {[catch {generate_design_initialization_data}  }] } {
      puts "DESIGN INITIALIZATION FAILED \n"
   } else {
      puts "DESIGN INITIALIZATION PASSED \n"
   }

if {[catch {run_tool -name {GENERATEPROGRAMMINGFILE}  }] } {
      puts "GENERATEPROGRAMMINGFILE FAILED \n"
   } else {
      puts "GENERATEPROGRAMMINGFILE PASSED \n"
   }
```

For more information about adding programming parameters, see the following.

- export_bitstream_file
- export_prog_job
- run_tool
- generate_design_initialization_data

© 2025 Microchip Technology Inc. and its subsidiaries

- close_project

2.2. Running the Tcl (Ask a Question)

From Windows command-line, the following command runs the `libero.tcl` and creates the Libero SoC project.

```
C:/Microchip/Libero/Designer/bin/libero.exe SCRIPT:<$Tcl_source_directory>/libero.tcl
```

After successful execution of Tcl script, the Libero project is created within the Tcl source directory.

For more examples on basic and advanced Libero SoC Tcl, see Libero SoC Design Suite Tcl Examples.

2.3. Updating a Project (Ask a Question)

You must update a Libero SoC project when the HDL and constraint files of a project change. To update the Libero SoC project, import the latest source files and re-run `libero.tcl`.

© 2025 Microchip Technology Inc. and its subsidiaries

3. Project Manager Tcl Commands (Ask a Question)

3.1. add_file_to_library (Ask a Question)

Description

This Tcl command adds a file to a library in your project.

```
add_file_to_library -library "library name" [-file "filename"]
```

Arguments

Parameter	Type	Description
library	string	Name of the library where you wish to add your file.
file	string	Specifies the new name of the file you wish to add (must be a full pathname). This parameter is optional

Error Codes

Error Code	Description
None	Parameter 'file' is missing or has invalid value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_file_to_library -library "library name" [-file "file"]+ '.
None	File is not in the project

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Add a file named `foo.vhd` from the `./project/hdl` directory to the library `my_lib`.

```
add_file_to_library -library "my_lib" -file "./project/hdl/foo.vhd"
```

See Also

- add_library
- remove_library
- rename_library

3.2. add_library (Ask a Question)

Description

This Tcl command adds a VHDL library to your project. To add a library, right-click the design module name in the Design Hierarchy select and Add VHDL Library from the context menu.

```
add_library -library "library name"
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
library	string	Specifies the name of your new library.

Error Codes

Error Code	Description
None	Required parameter 'library' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_library -library "library name."

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Create a new library with 'my_lib' name in your project.

```
add_library -library my_lib
```

See Also

- remove_library
- rename_library

3.3. add_modelsim_path (Ask a Question)

Description

This Tcl command adds a ModelSim simulation library to your project.

```
add_modelsim_path -lib library_name  [-path library_path] \
                [-remove "library_name"]
```

Arguments

Parameter	Type	Description
lib	string	Name of the library you want to add.
path	string	Path to library that you want to add. Enables you to change the mapping for your simulation library (both Verilog and VHDL). Type the pathname or click the **Browse** button to navigate to your library directory.
remove	string	Name of library you want to remove (if any).

Error Codes

Error Code	Description
None	Required parameter 'lib' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_modelsim_path -lib "library name" [-path "library path"] [-remove "TRUE | FALSE"]'

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Add the ModelSim library "msim_update2" located in the "c:\modelsim\libraries" directory and remove the library "msim_update":

```
add_modelsim_path -lib "msim_update2" \
                  -path "c:\modelsim\libraries" \
                  -remove "msim_update"
```

3.4. add_profile (Ask a Question)

Description

This Tcl command enables you to add new profile in your project and specified a Software IDE tool in your Tools profile. This command sets the same values as the Add or Edit Profile dialog box. The newly added profile becomes the active tool profile for the specified type of tool. You must provide a unique name.

```
add_profile -name profilename \
            -type value \
            -tool profiletool \
            -location tool_location \
            [-args tool_parameters ] \
            [-batch value ] \
            [-license license_information ] \
            [-32bit value ]
```

Arguments

Parameter	Type	Description
name	string	Specifies the unique name of your new profile.
type	string	Specifies your profile type, where value is one of the following: • synthesis - new profile for a synthesis tool. The default synthesis tool included with Libero SoC is Synplify Pro ME. • simulation - new profile for a simulation tool. The default simulator tool included with Libero SoC is QuestaSim Pro ME. • stimulus - new profile for a stimulus tool. Default not specified. Stimulus tool included with Libero SoC is WFL. • identifydebugger - new identify debugger tool profile. The default identify debugger tool included with Libero SoC is Identify Debugger. • coreconfig • sd_ide
tool	string	Name of the tool you are adding to the profile. It is mandatory.
location	string	Full pathname to the location of the tool you are adding to the profile. It is mandatory.
args	list of strings	Tool parameters (if any).

© 2025 Microchip Technology Inc. and its subsidiaries

add_profile (continued)		
Parameter	**Type**	**Description**
batch	boolean	Runs the tool in batch mode (if TRUE). Possible values are: • TRUE | 1 - runs the profile in batch mode. • FALE | 0 - Does not run the profile in batch mode.
license	string	License information.
32bit	boolean	Valid values are: 1, 0.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Unable to add profile: user Profile with same name already exists.
None	Required parameter 'tool' is missing.
None	Required parameter 'location' is missing.
None	Parameter 'location' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	type: Invalid argument value: 'value' (expecting synthesis, simulation, stimulus, coreconfig, identifydebugger or sw_ide).
None	Required parameter 'type' is missing.
None	Parameter 'tool' has illegal value.
None	Required parameter 'tool' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_profile -name "profile name" -type "synthesis | simulation | stimulus | coreconfig | identifydebugger | sw_ide" -tool "profile tool" -location "profile tool location" [-args "profile tool parameters"] [-batch "TRUE | FALSE"] [-license "profile License parameters"] [-32bit "TRUE | FALSE"]

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Create a new FlashPro tool profile called 'myflashpro' linked to a FlashPro installation in my "c:\programs\bin" directory:

```
add_profile -name "myflashpro" -type "synthesis" \
            -tool flashpro.exe -location {c:\programs\bin\flashpro.exe} \
            -batch FALSE
```


© 2025 Microchip Technology Inc. and its subsidiaries

Create a new Synthesis tool profile called 'synpol' linked to a Synplify Pro ME installation in my / sqatest/bin directory:

```
add_profile -name "synpol" -type synthesis -tool "Synplify Pro ME" \
            -location "/sqatest9/bin/synplify_pro" -batch FALSE
```

See Also

- edit_profile
- remove_profile
- select_profile
- export_profiles

3.5. associate_stimulus (Ask a Question)

Description

This Tcl command associates a stimulus file in your project. Before running simulation, you must associate a test bench. If you attempt to run simulation without an associated test bench, the Libero SoC Project Manager asks you to associate a test bench or open the simulator without a test bench.

```
associate_stimulus [-file {absolute path and name of the file}] \
                   [-mode "new | add | remove" ] -module "module"
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute path and name of the file to which you want to associate your stimulus files.
mode	string	Specifies whether you are creating a new stimulus association, adding or removing. Possible values are: • new - creates a new stimulus file association. • add - adds a stimulus file to an existing association. • remove - removes a stimulus file association.
module	string	Sets the module, where value is the name of the module. This is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	'file' is not in the project.
None	Required parameter 'module' is missing.
None	Parameter 'module' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'associate_stimulus [-file "file"]* [-mode "new | add | remove"] -module "module" '
None	mode: Invalid argument value: 'value' (expecting new, add or remove).

Supported Families

Supported Families
PolarFire®
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

associate_stimulus (continued)
Supported Families
SmartFusion® 2
IGLOO® 2

Example

The following example associates a new stimulus file 'stim.vhd' for stimulus.

```
associate_stimulus -file stim.vhd -mode new -module stimulus
```

See Also

- organize_sources

3.6. build_design_hierarchy (Ask a Question)

Description

This Tcl command rebuilds the Design/Stimulus Hierarchy. Any change to the design sources/ stimuli invalidates the design hierarchy/stimulus hierarchy. You can build the Design Hierarchy and Simulation Hierarchy by clicking the Build Hierarchy button.

```
build_design_hierarchy
```

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command rebuilds the Design/Stimulus Hierarchy.

```
build_design_hierarchy
```

3.7. change_link (Ask a Question)

Description

This Tcl command changes the source of a linked file in your project.

```
change_link -file filename -path new_source_path
```

Arguments

Parameter	Type	Description
file	string	Specifies absolute or relative path and name of the linked file you want to change.
path	string	Location of the file (absolute or relative path) you want to link to.

Error Codes

Error Code	Description
None	Required parameter 'path' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

change_link (continued)	
Error Code	**Description**
None	Required parameter 'file' is missing.
None	Parameter 'file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'change_link -file "file" -path "new source path"'

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

1. Change the link to a file 'sim1.vhd' in your project and link it to the `c:\Microchip\link_source\simulation_test.vhd` file.

```
change_link -file sim1.vhd \
            -path {c:\Microchip\link_source\simulation_test.vhd}
```

2. Change link for the file 'test.v' from 'E:\Share\test.v' to 'E:\Share\test\srcs\test.v' using environment variable 'MSCC_ROOT_1' that has the root directory path 'E:\Share'.

```
change_link -file {${MSCC_ROOT_1}/test.v} \
            -path {${MSCC_ROOT_1}/test/srcs/test.v}
```

See Also

- create_links
- unlink_files

3.8. change_all_links (Ask a Question)

Description

This Tcl command changes the source of all the linked files in your project.

```
change_all_links -current_path {current source path} \
                 -new_path {new source path}
```

Arguments

Parameter	Type	Description
current_path	string	Current source path of linked files. It is mandatory.
new_path	string	New source path of linked files. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	No files are linked in the project from the specified current path. Please check the path.
None	Param 'new_path' should not be empty.

© 2025 Microchip Technology Inc. and its subsidiaries

change_all_links (continued)	
Error Code	**Description**
None	Parameter 'new_path' has illegal value.
None	Required parameter 'new_path' is missing.
None	Parameter 'current_path' has illegal value.
None	Required parameter 'new_path' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'change_all_links -current_path "current_path" -new_path "new_path"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Change the source of all the link files in project from "./src/link_source_old" to "./src/link_source_new"

```
change_all_links -current_path {./src/link_source_old} \
                 -new_path {./src/link_source_new }
```

3.9. change_vault_location (Ask a Question)

Description

This Tcl command changes the location of the vault. Equivalent to clicking the Project menu, and choosing Vault/Repositories Setting and selecting new vault location, by default is "/usr/local/Microchip/common"/. The vault location is common to all Microchip software:

- Project Manager
- Firmware Catalog

Changing your vault location here updates the vault location for all the Microchip tools you use on your machine.

Important: This command overrides the vault location for all projects.

```
change_vault_location -location location
```

Arguments

Parameter	Type	Description
location	string	Specifies the new vault location. Value must be a file path. It is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'location' is missing.
None	Parameter 'location' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'change_vault_location -location "location" ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command changes the old location of vault into new location.

```
change_vault_location -location {new_vault_path/vault}
```

3.10. check_fdc_constraints (Ask a Question)

Description

This Tcl command checks FDC (Synplify Netlist Constraint) constraints files associated with the Synthesis tool. FDC constraints are used to optimize the HDL design using Synopsys SynplifyPro Synthesis engine and have the *.fdc extension.

```
check_fdc_constraints -tool {synthesis}
```

Arguments

Parameter	Type	Description
tool	string	The valid value is synthesis only.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'check_fdc_constraints -tool "synthesis | designer | physynth | timing | compilenetlist | pnr"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following command checks FDC constraints files associated with the Synthesis tool.

```
check_fdc_constraints -tool {synthesis}
```

See Also

- check_ndc_constraints

3.11. check_hdl (Ask a Question)

Description

This Tcl command checks the HDL in the specified file for validity.

```
check_hdl -file { absolute path and name of the HDL file }
```

Arguments

Parameter	Type	Description
file	string	Specified absolute path and name of the HDL file you want to check. It is mandatory.

Error Codes

Error Code	Description
None	HDL file is not in the project.
None	Required parameter 'file' is missing.
None	Parameter 'file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'check_hdl -file "file"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command checks HDL on the file hdl1.vhd.

```
check_hdl -file {/project/hdl/hdl1.vhd}
```

3.12. check_ndc_constraints (Ask a Question)

Description

This Tcl command checks NDC (Compile Netlist Constraint) constraints files associated with the Synthesis tool. NDC constraints are used to optimize the post-synthesis netlist with the Libero SoC Compile engine and have the *.ndc file extension.

```
check_ndc_constraints -tool {synthesis}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
tool	string	The valid value is synthesis.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'check_ndc_constraints -tool "synthesis | designer | physynth | timing | compilenetlist | pnr"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following Tcl command checks NDC constraints files associated with the Synthesis tool.

```
check_ndc_constraints -tool {synthesis}
```

See Also

- check_fdc_constraints

3.13. check_pdc_constraints (Ask a Question)

Description

This Tcl command checks Physical Design Constraints (PDC) constraint files associated with the Libero Place and Route tool. PDC Tcl is divided between I/O attribute and pin information from all floorplanning and timing constraints.

- The I/O Attributes tab allows you to manage I/O attributes/constraints for your design's Inputs, Outputs and Inouts. All I/O constraint files (PDC) have the *.pdc file extension and are placed in the `<project_location>/constraint/io` folder.
- The Floor Planner tab allows you to manage floorplanning constraints. Floorplanning constraints files have the *.pdc file extension and are placed in the `<project_location>\constraint\fp` folder.

```
check_pdc_constraints -tool {designer}
```

Arguments

Parameter	Type	Description
tool	string	The valid value is designer.

Error Codes

Error Code	Description
None	Please run 'Compile Netlist' before checking the validity of the associated constraint files.

© 2025 Microchip Technology Inc. and its subsidiaries

check_pdc_constraints (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'check_ndc_constraints -tool "synthesis | designer | physynth | timing | compilenetlist | pnr"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command checks PDC constraints files associated with the Libero Place and Route (designer) tool.

```
check_pdc_constraints -tool {designer}
```

3.14. check_sdc_constraints (Ask a Question)

Description

This Tcl command checks Synopsys Design Constraints (SDC) constraint files associated with the Libero tools: designer, synthesis or timing.

Note: This command cannot be run until Compile has been run.

```
check_sdc_constraints -tool {tool_name}
```

Arguments

Parameter	Type	Description
tool	string	Valid values are: synthesis, designer and timing.

Error Codes

Error Code	Description
None	type: Invalid argument value: '' (expecting synthesis, designer, physynth, timing or compilenetlist).
None	Parameter 'param_name' is not defined. Valid command formatting is 'check_pdc_constraints -tool "synthesis | designer | physynth | timing | compilenetlist | pnr"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This command checks the SDC constraint files associated with Timing Verification.

```
check_sdc_constraints -tool {timing}
```

This command checks the SDC constraint files associated with Place and Route.

```
check_sdc_constraints -tool {designer}
```

This command checks the SDC constraint files associated with Synthesis.

```
check_sdc_constraints -tool {synthesis}
```

3.15. cleanall_tool (Ask a Question)

Description

This Tcl command cleans all the tools till the tool name passed as the argument to the command. All the remaining child tools if any that are in Pass state will be converted to Out of Date state.

```
cleanall_tool -name {tool_name}
```

Arguments

Parameter	Type	Description
name	string	Tool_name is the tool passed as the argument till which all the tools are cleaned. All the remaining child tools after tool_name if any that are in Pass state will be converted to Out of Date state.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'cleanall_tool -name "tool name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

If design flow is ran till Place and Route, below is the behavior of the cleanall_tool command.

```
cleanall_tool -name {SYNTHESIZE}
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.16. close_design (Ask a Question)

Description

This Tcl command closes the current design and brings Designer to a fresh state to work on a new design. This is equivalent to selecting the **Close** command from the **File** menu.

```
close_design -component "smartdesign component name"
```

Arguments

Parameter	Type	Description
component	string	Specifies the name of the SmartDesign component to be closed. It is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'component' is missing.
None	Parameter 'component' has illegal value.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example checks is the "Top" design component successfully closed or no.

```
if {[ catch {close_design -component {Top}} ]} {
    puts “Failed to close design”
# Handle Failure
} else {
    puts “Design closed successfully”
# Proceed with processing a new design
}
```

See Also

- create_smartdesign
- open_smartdesign
- save_smartdesign

3.17. close_project (Ask a Question)

Description

This Tcl command closes the current project in Libero SoC. This command is equivalent to selecting **Close Project** from the **File** menu.

```
close_project [-save value]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
-save	boolean	Saves the current project in Libero SoC before closing project. • TRUE, true or 1 - saves your project in Libero SoC before closing project. Default is true. • FALSE, false or 0 - closes your project without saving.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
close_project
```

See Also

- open_project

3.18. configure_core (Ask a Question)

Description

This Tcl command modifies the configuration of an existing core component in the SmartDesign. This command works for core components created for different types of cores namely, Sg cores, System Builder cores and Direct cores. In the Libero SoC, choose `View > Windows > Catalog`. The Catalog displays a list of available cores, busses and macros. Double-click a core to open the core generator and configure it and add it to your design.

Limitations: The command does not work for SmartFusion 2 and IGLOO 2 System Builder components, SmartFusion 2 MSS component, RTG4 PCIE_SERDES_IF_INIT (RTG4 High Speed Serial Interface 1 - EPCS and XAUI - with Initialization), NPSS_SERDES_IF_INIT (RTG4 High Speed Serial Interface 2 - EPCS and XAUI - with Initialization), and RTG4FDDRC_INIT (RTG4 DDR Memory Controller with initialization) core components.

```
configure_core -component_name component_name -params core_parameters
```

Arguments

Parameter	Type	Description
component_name	string	Specifies the name of the component to be configured. It is mandatory.
params	string	Specifies the parameters needed to configure the core component. It is mandatory. It can either take single parameter or multiple parameters at a time. This command will fail if none of the core parameters are specified.

Error Codes

Error Code	Description
None	Required parameter 'component_name' is missing.
None	Unable to create core. A Component with that name already exists.

© 2025 Microchip Technology Inc. and its subsidiaries

configure_core (continued)	
Error Code	**Description**
None	Parameter 'p' is not defined. Valid command formatting is 'configure_core -component_name "component_name" [-params "[params]+"].
None	Cannot find Spirit core configuration file for vendor:Actel library:Simulation name:<core_name> version:1.0.1.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following commands modifies the configuration of "Core_UART" and "PF_CCC_C0" core components - sets cores parameters values in the SmartDesign.

```
configure_core -component_name {PF_CCC_C0} \
               -params "GL1_0_IS_USED:false" \
                       "GL0_0_IS_USED:true” “GL0_0_OUT_FREQ:200”}
```

```
configure_core -component_name {Core_UART} \
               -params {"BAUD_VAL_FRCTN_EN:false" \
                        "RX_FIFO:0" "RX_LEGACY_MODE:0" \
                        "TX_FIFO:1" "USE_SOFT_FIFO:1"}
```

See Also

- create_and_configure_core
- remove_core
- download_core
- download_latest_cores

3.19. create_and_configure_core (Ask a Question)

Description

This Tcl command creates a configured core component for a core selected from the Libero Catalog. This command works for core components created for different types of cores namely, Sg cores, System Builder cores and Direct cores. In the Libero SoC, choose **View** > **Windows** > **Catalog**. The Catalog displays a list of available cores, bus and macros. From the Catalog, you can create a component from the list of available cores, add a processor or peripheral, add a bus interface to your SmartDesign component, instantiate simulation cores or add a macro (Arithmetic, Basic Block, etc.) to your SmartDesign component.

Limitations: The command does not work for SmartFusion 2 and IGLOO 2 System Builder components, SmartFusion 2 MSS component, and RTG4 PCIE_SERDES_IF_INIT (RTG4 High Speed Serial Interface 1 - EPCS and XAUI - with Initialization), NPSS_SERDES_IF_INIT (RTG4 High Speed Serial Interface 2 - EPCS and XAUI - with Initialization) and RTG4FDDRC_INIT (RTG4 DDR Memory Controller with initialization) core components.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: For Direct Core and Solutions cores, refer to the core handbook or the core user guide for a list of valid parameters and values.

```
create_and_configure_core -core_vlnv {Vendor:Library:Name:Version} \
                          -component_name "component_name" \
                          -params "core_parameters"
```

Arguments

Parameter	Type	Description
core_vlnv	string	Specifies the version identifier of the core being configured. It is mandatory.
component_name	string	Specifies the name of the configured core component. It is mandatory.
params	string	Specifies the parameters needed to configure the core component. It is mandatory. It can either take single parameter or multiple parameters at a time. This command will fail if none of the core parameters are specified.

Error Codes

Error Code	Description
None	Required parameter 'core_vlnv' is missing.
None	Parameter 'core_vlnv' of command 'create_and_configure_core' cannot be empty.
None	Required parameter 'component_name' is missing.
None	Unable to create core. A Component with that name already exists.
None	Cannot find Spirit core configuration file for vendor:Actel library:Simulation name:<core_name> version:1.0.1.
None	Parameter 'param_name' is not defined. Valid command formatting is 'create_and_configure_core -core_vlnv "vlnv for the core" -component_name "name of the created component" [-params "[params]+"] '

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command configured SgCore(PF_CCC_C3) core - sets core parameters values, specifies the version identifier of the SgCore(PF_CCC_C3) core.

```
create_and_configure_core -core_vlnv {Actel:SgCore:PF_CCC:1.0.115} \
-component_name {PF_CCC_C3} \
-params {"PLL_IN_FREQ_0:25" \
"GL0_0_IS_USED:true" \
"GL0_0_OUT_FREQ:150" \
"GL0_1_IS_USED:true" \
"GL0_1_OUT_FREQ:50"}
```

See Also

- configure_core
- remove_core
- download_core

© 2025 Microchip Technology Inc. and its subsidiaries

- download_latest_cores

3.20. configure_tool (Ask a Question)

Description

This Tcl command is a general purpose command that is used to set the parameters for any tool called by Libero for the families. The command requires the name of the tool and one or more parameters in the `tool_parameter:value` format. These parameters are separated and passed to the tool to setup its run.

Important: You can repeat <params> argument for multiple parameters.

```
configure_tool -name {tool_name} -params {<parameter>:<value>}
```

Arguments

Parameter	Type	Description
tool_name	string	Specifies the name of tool for which you wish to configure tool options. It is mandatory. Each tool_name has its own set of parameters. • COMPILE • CONFIGURE_ACTION_PROCEDURES • CONFIGURE_PROG_OPTIONS • CONFIGURE_PROG_OPTIONS_RTG4 • SYNTHESIZE • PLACEROUTE • VERIFYTIMING • VERIFYPOWER • GENERATEPROGRAMMINGDATA • GENERATEPROGRAMMINGFILE • PROGRAMDEVICE • PROGRAM_OPTIONS • PROGRAMMER_INFO • SPM • FLASH_FREEZE • PROGRAM_RECOVERY • USER_PROG_DATA • INIT_LOCK
params	string	Specifies the tool options/parameters for the value you want to configure. List of parameters and values. There may be multiple `-params` arguments (see example below). This is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'params' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_tool -name "tool name" [-params "params"]+ '.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the COMPILE command options, `DISPLAY_FANOUT_LIMIT` to 10 and `MERGE_SDC` to true:

```
configure_tool -name {COMPILE} \
               -params {DISPLAY_FANOUT_LIMIT:10} \
               -params {MERGE_SDC:true}
```

See Also

- get_tool_options

3.21. create_links (Ask a Question)

Description

This Tcl command creates a link (or links) to a file/files in your project. Specify absolute or relative path and name of the file you want to link. The same file you cannot link to different libraries.

```
create_links [-convert_EDN_to_HDL "TRUE | FALSE"] \
[-hdl_source_folder "Source folder"]* \
[-library "library"] \
[-hdl_source "file"]* \
[-stimulus "file"]* \
[-edif "file"]* \
[-sdc "file"]* \
[-ndc "file"]* \
[-fp_pdc "file"]* \
[-io_pdc "file"]* \
[-net_fdc "file"]* \
[-verilog_netlist "file"]* \
[-vcd "file"]* \
[-dcf "file"]* \
[-pin "file"]* \
[-crt "file"]* \
[-gcf "file"]*
```

Arguments

Parameter	Type	Description
convert_EDN_to_HDL	boolean	Use the `-convert_EDN_to_HDL` parameter to convert the EDIF file to HDL and then import the converted HDL file. Valid values: TRUE, 1, true, FALSE, 0 or false. If the `-edif` option is not specified or the `-convert_EDN_to_HDL` is used with another option, EDIF to HDL conversion will fail.
hdl_source_folder	string	Name of the HDL folder you want to link. For unlink folder you must unlink files form folder one by one.
library	string	Specifies the name of the library where you want to link file. The same file you cannot link different libraries.
hdl_source	string	Name of the HDL file you want to link.
stimulus	string	Name of the stimulus file you want to link.
edif	string	Name of the EDIF Netlist file you want to link. It used with `convert_EDN_to_HDL` option.

© 2025 Microchip Technology Inc. and its subsidiaries

create_links (continued)

Parameter	Type	Description
sdc	string	Name of the SDC file you want to link.
ndc	string	Name of the NDC (Compile Netlist Constraint) file you want to link.
fp_pdc	string	Name of the Floor Planner PDC file you want to link.
io_pdc	string	Name of the IO PDC file you want to link.
net_fdc	string	Name of the FDC (Synplify Netlist Constraint) file you want to link.
vcd	string	Name of the VCD file you want to link.
verilog_netlist	string	Name of the VM (Synthesized Verilog Netlist) file you want to link.
pin	string	Name of the PIN file you want to link. (Not supported for PolarFire, SmartFusion 2, IGLOO 2, and RTG4 families).
dcf	string	Name of the DCF(Timing Constraint Files) file you want to link. (Not supported for PolarFire, SmartFusion 2, IGLOO 2, and RTG4 families).
gcf	string	Name of the GCF (ProASIC® Constraint Files) file you want to link. (Not supported for PolarFire, SmartFusion 2, IGLOO 2, and RTG4 families).
crt	string	Name of the CRT (Criticality Files) file you want to link. (Not supported for PolarFire, SmartFusion 2, IGLOO 2, and RTG4 families).

Error Codes

Error Code	Description
None	Error: Parameter 'vhd' is not defined. Valid command formatting is 'create_links [-convert_EDN_to_HDL "TRUE | FALSE"] \ [-hdl_source_folder "Source folder"]* [-library "library"] \ [-ndc "file"]* [-crt "file"]* \ [-fp_pdc "file"]* [-hdl_source "file"]* \ [-stimulus "file"]* [-io_pdc "file"]* \ [-pin "file"]* [-gcf "file"]* \ [-sdc "file"]* [-net_fdc "file"]* \ [-verilog_netlist "file"]* \ [-dcf "file"]* [-vcd "file"]*
None	Cannot import the 'Timing Constraint Files'; your selected family does not support DCF constraints.
None	Cannot import the 'ProASIC Constraint Files'; your selected family does not support GCF constraints.
None	Cannot import the 'Criticality Files'; your selected family does not support CRT constraints.
None	Cannot import the 'Pin Files'; your selected family does not support PIN constraints.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

1. Create a link to the file hdl1.v.

```
create_links -hdl_source hdl1.v
```

2. Link files to the project located at "E:\Share\abc.edn" and "E:\Share\test.v" using Environment variable "MSCC_ROOT_1" that has the root directory path "E:\Share".

```
create_links \
-convert_EDN_to_HDL 0 \
-library {work} \
-edif {${MSCC_ROOT_1}/abc.edn} \
-hdl_source {${MSCC_ROOT_1}/test.v}
```

See Also

- change_link

© 2025 Microchip Technology Inc. and its subsidiaries

- unlink_files

3.22. create_set (Ask a Question)

Description

This Tcl command creates a set of paths to be analyzed. Use the arguments to specify which paths to include. To create a set that is a subset of a clock domain, specify it with the `-clock` and `-type` arguments. To create a set that is a subset of an inter-clock domain set, specify it with the `-source_clock` and `-sink_clock` arguments. To create a set that is a subset (filter) of an existing named set, specify the set to be filtered with the `-parent_set` argument.

```
create_set -name "set name" -parent_set "parent set name" \
           -type <set_type> -clock <clock name> \
           -source_clock <clock name> -sink_clock <clock name> \
           -in_to_out -source <port/pin pattern> -sink <port/pin pattern>
```

Arguments

Parameter	Type	Description
name	string	Specifies a unique name for the newly created path set.
parent_set	string	Specifies the name of the set to filter from.
clock	string	Specifies that the set is to be a subset of the given clock domain. This argument is valid only if you also specify the `-type` argument.
type	string	Specifies the predefined set type on which to base the new path set. You can only use this argument with the `-clock` argument, not by itself. • reg_to_reg - paths between registers in the design. • async_to_reg - paths from asynchronous pins to registers. • reg_to_async - paths from registers to asynchronous pins. • external_recovery - the set of paths from inputs to asynchronous pins. • external_removal - the set of paths from inputs to asynchronous pins. • external_setup - paths from input ports to registers. • external_hold - paths from input ports to registers. • clock_to_out - paths from registers to output ports.
in_to_out	None	Specifies that the set is based on the "Input to Output" set, which includes paths that start at input ports and end at output ports.
source_clock	string	Specifies that the set will be a subset of an inter-clock domain set with the given source clock. You can only use this option with the `-sink_clock` argument.
sink_clock	string	Specifies that the set will be a subset of an inter-clock domain set with the given sink clock. You can only use this option with the `-source_clock` argument.
source	string	Specifies a filter on the source pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.
sink	string	Specifies a filter on the sink pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.

Supported Families

Supported Families
PolarFire®
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

create_set (continued)
Supported Families
SmartFusion® 2
IGLOO® 2

Example

The following example creates set with "my_user_set" name. Filters all C* ports and D* pins in the current design.

```
create_set -name {my_user_set} -source {C*} -sink {D*}
```

The following example creates set with "my_other_user_set" name that is a subset (filter) of an existing "my_user_set" set.

```
create_set -name {my_other_user_set} -parent_set {my_user_set} \
           -source {CL*}
```

The following example creates set with "another_set" name which is the subset of an inter-clock domain set with the given source clock.

```
create_set -name {another_set} -source_clock {EXTERN_CLOCK} \
           -sink_clock {MY_GEN_CLOCK}
```

3.23. create_smartdesign (Ask a Question)

Description

This Tcl command creates a SmartDesign to your project. To create design, click the Create SmartDesign tool name from Design Flow.

```
create_smartdesign -sd_name "smartdesign_component_name"
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component to be created. It is mandatory.

Error Codes

Error Code	Description
None	Cannot create SmartDesign ' smartdesign_component_name' because it violates HDL naming rules. Please specify a different name.
None	Required parameter 'sd_name' is missing.
None	Parameter 'component' is missing or has invalid value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'create_smartdesign -sd_name "sd_name"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

create_smartdesign (continued)
Supported Families
IGLOO® 2

Example

Creates new design component with "top" name.

```
create_smartdesign -sd_name {top}
```

3.24. delete_component (Ask a Question)

Description

This Tcl command deletes a component from the Design Hierarchy.

```
delete_component -component_name "component_name"
```

Arguments

Parameter	Type	Description
component_name	string	Specifies the name of the component to be deleted. It is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'component_name' is missing.
None	Parameter 'component_name' of command 'delete_component' cannot be empty.
None	Parameter 'param_name' is not defined. Valid command formatting is 'delete_component -component_name "name of the component to delete" '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command deletes a component with component name from the Design Hierarchy.

```
delete_component -component_name {component}
```

The following command deletes a component with shifter name from the Design Hierarchy.

```
delete_component -component_name {shifter}
```

See Also

- generate_component
- import_component

© 2025 Microchip Technology Inc. and its subsidiaries

3.25. delete_files (Ask a Question)

Description

This Tcl command deletes files in your Libero SoC project.

```
delete_files -file {value} [-from_disk {TRUE | FALSE}]
```

Arguments

Parameter	Type	Description
file	string	Specifies the file you wish to delete from the project. This parameter is required for this Tcl command. It does not delete the file from the disk.
from_disk	boolean	Specifies if the file will be deleted from disk or not. The default value is 'False'. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'delete_files [-file "file"]+ [-from_disk "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Delete the files file1.vhd and file2.vhd from the project.

```
delete_files -file {./project/hdl/file1.vhd} -from_disk {TRUE}
```

3.26. download_core (Ask a Question)

Description

This Tcl command downloads a core and adds it to your repository. The Catalog enables you to download cores from a web repository into a Vault. A Vault is a local directory (either local to your machine or on the local network) that contains cores downloaded from one or more repositories. A repository is a location on the web that contains cores that can be included in your design. The Catalog displays all the cores in your Vault.

You may want to import a core from a file when:

- You do not have access to the internet and cannot download the core.

© 2025 Microchip Technology Inc. and its subsidiaries

- A core is not complete and has not been posted to the web (you have an evaluation core).

```
download_core -vlnv "Vendor:Library:Name:Version" [-location "location"]
```

Arguments

Parameter	Type	Description
vlnv	string	Vendor, library, name and version of the core you want to download. It is mandatory. You can repeat this argument for multiple VLNVs.
location	string	Location of the repository where you wish to add the core. It is optional.

Error Codes

Error Code	Description
None	Required parameter 'vlnv' is missing.
None	Parameter 'vlnv' is missing or has invalid value.
None	Parameter 'location' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'download_core [-vlnv "vlnv"]+ [-location "location"] ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example downloads the System Builder (PF_DDR4) core to the repository www.microsemi.com/repositories/SgCore.

```
download_core -vlnv {Actel:SystemBuilder:PF_DDR4:1.0.102} \
        -location {www.microsemi.com/repositories/SgCore}
```

See Also

- configure_core
- create_and_configure_core
- download_latest_cores

3.27. download_latest_cores (Ask a Question)

Description

This Tcl command is used to download the latest cores into the vault. A project does not need to be open to run this command. This command takes no arguments. The **Catalog Options** dialog box enables you to customize your Catalog. **Display only the latest version of a core** is checked by default. This option, if checked, shows the latest versions of cores that are not in the Vault, and also filters out any duplicate cores that have the same Vendor, Library, and Name, with an earlier version number.

© 2025 Microchip Technology Inc. and its subsidiaries

Important:

- If there are no cores to be downloaded, you will see the following message:

```
Info:All the latest cores are present in the vault.
```

- If there is no internet access, you will see the following message:

```
Cannot download cores. Please make sure you're connected to the internet
and internet access is not disabled from the Preferences.
```

```
download_latest_cores
```

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command downloads the latest cores into the vault.

```
download_latest_cores
```

See Also

- configure_core
- download_core
- remove_core

3.28. edit_profile (Ask a Question)

Description

This Tcl command enables you to edit profile in your project. You can edit profile from **Project** > **Tool Profiles** or right-click tool name and select **Edit Profile**.

```
edit_profile -name profilename \
             [-type value] \
             [-tool {profile tool} ] \
             [-location {profile location} ] \
             [-args "profile tool parameters" ] \
             [-batch "TRUE|FALSE" ] \
             [-new_name "new profile name" ]
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of your new profile.
type	string	Specifies your profile type, where value is one of the following: • synthesis - new profile for a synthesis tool. • simulation - new profile for a simulation tool. • stimulus - new profile for a stimulus tool. • identifydebugger - new identify debugger tool profile.
tool	string	Name of the tool you are adding to the profile.

© 2025 Microchip Technology Inc. and its subsidiaries

edit_profile (continued)		
Parameter	**Type**	**Description**
location	string	Full pathname to the location of the tool you are adding to the profile.
args	list of strings	Profile tool parameters (if any).
batch	string	Runs the tool in batch mode (if TRUE). Possible values are: • TRUE - runs the profile in batch mode. • FALSE - does not run the profile in batch mode.
new_name	string	Specifies new name of profile.

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	Parameter 'tool' has illegal value.
None	Parameter 'location' has illegal value.
None	Parameter 'new_name' has illegal value.
None	type: Invalid argument value: '' (expecting synthesis, simulation, stimulus, coreconfig, identifydebugger or sw_ide).
None	batch: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'edit_profile -name "profile name" [-type "synthesis | simulation | stimulus | coreconfig | identifydebugger | sw_ide"] [-tool "profile tool"] [-location "profile tool location"] [-args "profile tool parameters"] [-batch "TRUE | FALSE"] [-license "profile License parameters"] [-32bit "TRUE | FALSE"] [-new_name "new profile name"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

To edit a FlashPro tool profile called 'mySynplify' linked to a new SynplifyPro installation in my c:\programs\actel\SynplifyPro\bin directory, change the name to updated_synplifypro.

```
edit_profile -name {mySynplify} -type synthesis \
             -tool {Synplify.exe} \
             -location "c:\programs\actel\SynplifyPro\bin\synplify_pro" \
             -batch FALSE -new_name "updated_synplifypro"
```

See Also

- add_profile
- remove_profile
- export_profiles

© 2025 Microchip Technology Inc. and its subsidiaries

3.29. edit_post_layout_design (Ask a Question)

Description

This is the Design Flow tool that allows you to tune I/O signal integrity parameters and external timing without executing Place and Route again. Input is provided using a PDC file.

The PDC file contains one or more invocations of two PDC commands:

- `edit_io`
- `edit_instance_delay`

The `edit_post_layout_design` command fails when any commands in the input PDC file fail. The PDC commands fail, if the syntax is incorrect, the referenced instances do not exist, or the values are out of legal ranges. If the batch command fails, the layout state of the design does not change. If the batch command succeeds:

- Layout state changes to reflect the values in the PDC commands.
- Pin report and delay instance report files are regenerated to reflect the latest values.
- Downstream tools Verify Timing, Verify Power, Generate FPGA Array Data, and Generate Back Annotated Files are invalidated.

Notes:

- The file is not managed by the project (no import/link into the project).
- The file does not show up in the Libero Constraint Manager window. It is not a source to the layout tool.
- The file is not linked. Therefore, there is no clean/un-link option for it.

The `edit_post_layout_design` command generates the `<project>/designer/<root>/top_editpostlayout_log.log` file to provide information about the run, and the PDC file that is used in the run.

```
edit_post_layout_design \
            -file {absolute or relative path and name of the pdc file}
```

Arguments

Parameter	Type	Description
file	string	Mandatory. Specifies absolute or relative path and name of the `.pdc` file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Specify a pdc file as an input.
None	File `/prj/user.pdc` does not exist.
19138170	Parameter 'param_name' is not defined. [edit_io -param_name][/prj/example.pdc]
19138170	Required parameter 'port_name' is missing. [edit_io][/prj/example.pdc]
19138170	Parameter 'port_name' is missing or has invalid value. [edit_io][/prj/example.pdc]
19138170	Parameter 'port_name' has illegal value. [edit_io][/prj/example.pdc]
19138170	Required parameter 'inst_name' is missing. [edit_instance_delay][/prj/example.pdc]
19138170	Parameter 'param_name' is not defined. [edit_instance_delay -param_name][edit_instance_delay][/prj/example.pdc]
19138170	Required parameter 'properties' is missing. [edit_instance_delay][/prj/example.pdc]

© 2025 Microchip Technology Inc. and its subsidiaries

edit_post_layout_design (continued)	
Error Code	**Description**
19138170	Parameter 'properties' has illegal value. [edit_instance_delay][/prj/example.pdc]
None	Post layout tuning cannot be done because Place and Route has not been run. Run Place and Route and then try again.
19137989	Port name does not exist in the netlist or is not connected to an IoCell macro.[edit_io -port_name "port_name"] [[/prj/example.pdc]
None	Parameter 'param_name' is not defined. Valid command formatting is 'edit_post_layout_design -file "file"'.

Supported Families

PolarFire®
PolarFire SoC

Example

The following example edits the post-layout design database using 'edit_io' invocation of PDC commands with 'port_name' option:

```
edit_post_layout_design -file {./user.pdc}
```

The contains the 'edit_io' PDC command:

```
edit_io -port_name "D"
```

3.30. export_as_link (Ask a Question)

Description

This Tcl command exports a file to another directory and links to the file.

```
export_as_link -file {absoulte or relative path and name of the file} \
               -path link_path
```

Arguments

Parameter	Type	Description
file	string	Specifies absolute or relative path and name of the file you want to export as a link.
path	string	Path of the link.

Error Codes

Error Code	Description
None	'c.v' must not be part of a component and must be local to the project.
None	Required parameter 'file' is missing.
None	Parameter 'file' has illegal value.
None	Required parameter 'path' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_as_link -file "file" -path "link path" '.

Supported Families

Supported Families
PolarFire®
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

export_as_link (continued)
Supported Families
SmartFusion® 2
IGLOO® 2

Example

Export the file hdl1.vhd as a link to `c:\Microchip\link_source`.

```
export_as_link -file hdl1.vhd -path {c:\Microchip\link_source}
```

3.31. export_ba_files (Ask a Question)

Description

Tcl command to export the backannotated files. The backannotated files are:

- <design_name>_fast_hv_lt_ba.v
- <design_name>_slow_lv_ht_ba.v
- <design_name>_slow_lv_lt_ba.v (Verilog backannotated netlist) or <design_name>_fast_hv_lt_ba.vhd
- <design_name>_slow_lv_ht_ba.vhd
- <design_name>_slow_lv_lt_ba.vhd (VHDL backannotated netlist)
- <design_name>_fast_hv_lt_ba.sdf
- <design_name>_slow_lv_ht_ba.sdf
- <design_name>_slow_lv_lt_ba.sdf (Standard Delay Format) timing file.

These files are passed to the default simulator for postlayout simulation. Before exporting, you need to run 'Place and Route'.

```
export_ba_files -export_dir {absolute path to folder location} \
                -export_file_name {name of file} \
                -vhdl {value} \
                -min_delay {value}
```

Arguments

Parameter	Type	Description
export_dir	string	Specifies the path where you wish to export the backannotated files.
export_file_name	string	File name to generate the files. If not specified, it takes <design_name> as the default. If specified it takes <design_name_file_name>.
vhdl	integer	Generates the <design_name>_ba.v and <design_name>_ba.sdf when set to 0 and <design_name>_ba.vhd and <design_name>_ba.sdf when set to 1. Default is 0.
min_delay	integer	Set to 1 to export enhanced min delays to include your best-case timing results in your Back Annotated file. Default is 0.

Error Codes

Error Code	Description
None	Required parameter 'export_dir' is missing.
None	vhdl: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	min_delay: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).

© 2025 Microchip Technology Inc. and its subsidiaries

export_ba_files (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_ba_files -export_dir "Export Dir Name" \ [-export_file_name "Export File Name"] \ [-vhdl "TRUE | FALSE"] \ [-min_delay "TRUE | FALSE"] ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example creates the "`E:\designer\export\sd1`" directory where the following backannotated files are generated and exported.

- test_fast_hv_lt_ba.sdf
- test_fast_hv_lt_ba.v
- test_slow_lv_ht_ba.sdf
- test_slow_lv_ht_ba.v
- test_slow_lv_lt_ba.sdf
- test_slow_lv_lt_ba.v

```
export_ba_files -export_dir {E:\designs\export\sd1} \
                -export_file_name {test} \
                -vhdl 0 \
                -min_delay 1
```

3.32. export_bitstream_file (Ask a Question)

Description

This Tcl command configures the parameters for the bitstream to be exported from Libero.

Important: RTG4 and PolarFire devices do not support the security, SPI directory, or serialization options that SmartFusion 2 and IGLOO 2 devices support.

The following is the syntax for the Export Bitstream File Tcl command for SmartFusion 2, IGLOO 2, and RTG4.

Important: The Tcl script file exported from Libero will include all command options. You can modify or remove the options.

```
export_bitstream_file [-file_name "file_name"] \
[-export_dir "export_dir"] \
[-format "PPD | DAT | HEX | STP | CHAIN_STP | SPI | SVF" ] \
[-for_ihp "TRUE | FALSE" ] \
[-master_file "TRUE | FALSE" ] \
[-master_file_components "SECURITY | FABRIC | ENVM" ] \
[-encrypted_uek1_file "TRUE | FALSE" ] \
[-encrypted_uek1_file_components "FABRIC | ENVM" ] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
[-encrypted_uek2_file "TRUE | FALSE" ] \
[-encrypted_uek2_file_components "FABRIC | ENVM" ] \
[-encrypted_uek3_file "TRUE | FALSE" ] \
[-encrypted_uek3_file_components "FABRIC | ENVM" ] \
[-trusted_facility_file "TRUE | FALSE" ] \
[-trusted_facility_file_components 'FABRIC | ENVM" ] \
[-add_golden_image "TRUE | FALSE" ] \
[-golden_image_address "golden_image_address" ] \
[-golden_image_design_version "golden_image_design_version" ] \
[-add_update_image "TRUE | FALSE" ] \
[-update_image_address "update_image_address" ] \
[-update_image_design_version "update_image_design_version" ] \
[-serialization_stapl_type "serialization_stapl_type" ] \
[-serialization_target_solution "serialization_target_solution" ] \
[-script "script" ] \
[-force_rtg4_otp "TRUE | FALSE" ] \
[-master_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek1_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek2_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek3_include_plaintext_passkey "TRUE | FALSE" ]
```

The following is the syntax for the Export Bitstream File Tcl command for PolarFire.

```
export_bitstream_file [-file_name "file_name"] \
[-export_dir "export_dir"] \
[-format "PPD | DAT | HEX | STP | SPI" ] \
[-for_ihp "TRUE | FALSE" ] \
[-limit_SVF_file_size "TRUE | FALSE" ] \
[-limit_SVF_file_by_max_filesize_or_vectors "limit_SVF_file_by_max_filesize_or_vectors" ] \
[-svf_max_filesize "svf_max_filesize" ] \
[-svf_max_vectors "svf_max_vectors" ] \
[-master_file "TRUE | FALSE" ] \
[-master_file_components "SECURITY | FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-encrypted_uek1_file "TRUE | FALSE" ] \
[-encrypted_uek1_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-encrypted_uek2_file "TRUE | FALSE"] \
[-encrypted_uek2_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-trusted_facility_file "TRUE | FALSE"] \
[-trusted_facility_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-trusted_facility_keep_fabric_operational "TRUE | FALSE"] \
[-trusted_facility_skip_startup_seq "TRUE | FALSE"] \
[-zeroization_likenew_action "TRUE | FALSE" ] \
[-zeroization_unrecoverable_action "TRUE | FALSE" ] \
[-master_backlevel_bypass "TRUE | FALSE" ] \
[-uek1_backlevel_bypass "TRUE | FALSE" ] \
[-uek1_keep_fabric_operational "TRUE | FALSE"] \
[-uek1_skip_startup_seq "TRUE | FALSE"] \
[-uek2_backlevel_bypass "TRUE | FALSE" ] \
[-uek2_keep_fabric_operational "TRUE | FALSE"] \
[-uek2_skip_startup_seq "TRUE | FALSE"] \
[-master_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek1_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek2_include_plaintext_passkey "TRUE | FALSE" ] \
[-script "script" ]
```

The following is the syntax for the Export Bitstream File Tcl command for PolarFire SoC.

Important: High water mark values are needed when One Way Passcode (OWP) is enabled and SPI file is being exported.

```
export_bitstream_file [-file_name "file_name"] \
[-export_dir "export_dir"] \
[-format "PPD | DAT | HEX | STP | SPI" ] \
[-for_ihp "TRUE | FALSE" ] \
[-limit_SVF_file_size "TRUE | FALSE" ] \
[-limit_SVF_file_by_max_filesize_or_vectors "limit_SVF_file_by_max_filesize_or_vectors" ] \
[-svf_max_filesize "svf_max_filesize" ] \
[-svf_max_vectors "svf_max_vectors" ] \
[-master_file "TRUE | FALSE" ] \
[-master_file_components "SECURITY | FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-encrypted_uek1_file "TRUE | FALSE" ] \
[-encrypted_uek1_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-encrypted_uek2_file "TRUE | FALSE"] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
[-encrypted_uek2_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-trusted_facility_file "TRUE | FALSE"] \
[-trusted_facility_file_components "FABRIC | SNVM | ENVM | FABRIC_SNVM" ] \
[-zeroization_likenew_action "TRUE | FALSE" ] \
[-zeroization_unrecoverable_action "TRUE | FALSE" ] \
[-master_backlevel_bypass "TRUE | FALSE" ] \
[-uek1_backlevel_bypass "TRUE | FALSE" ] \
[-uek2_backlevel_bypass "TRUE | FALSE" ] \
[-master_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek1_include_plaintext_passkey "TRUE | FALSE" ] \
[-uek2_include_plaintext_passkey "TRUE | FALSE" ] \
[-sanitize_snvm "TRUE | FALSE" ] \
[-sanitize_envm "TRUE | FALSE" ] \
[-trusted_facility_keep_fabric_operational "TRUE | FALSE" ] \
[-trusted_facility_skip_startup_seq "TRUE | FALSE" ] \
[-uek1_keep_fabric_operational "TRUE | FALSE" ] \
[-uek1_skip_startup_seq "TRUE | FALSE" ] \
[-uek1_high_water_mark {1234} ] \
[-uek2_keep_fabric_operational "TRUE | FALSE" ] \
[-uek2_skip_startup_seq "TRUE | FALSE" ] \
[-uek2_high_water_mark {1234} ] \
```

Arguments

Parameter	Type	Description
file	string	Specifies the name of the file that will be exported. The default file name is the design name. **Important:** File name must start with the design name.
export_dir	string	Specifies the directory location for the export. By default, the file is saved in the `<designer>/<design_name>/export` directory.
format	string	Specifies the bitstream file formats to be exported. Space is used as a delimiter. The value can be any one of PPD, STP, CHAIN_STP, DAT, SPI, HEX, and SVF. If omitted, PPD and DAT files will be exported. **Notes:** • Exporting CHAIN_STP, SVF, and SPI files is not supported in RTG4. • Exporting CHAIN_STP and SVF files is not supported in PolarFire.
for_ihp	boolean	Specifies the export of bitstream files for Microchip In-House Programming (IHP). Valid values are: TRUE, true, 1, FALSE, false, 0. The default value is 0.
limit_SVF_file_size	boolean	Specifies limit on the SVF file size. Valid values are: TRUE, true, 1, FALSE, false, 0.
limit_SVF_file_by_max_filesize_or_vectors	boolean	Specifies limit on the SVF file size or vectors number. Valid values are: TRUE, true, 1, FALSE, false, 0.
svf_max_filesize	integer	Specifies svf file maximum size. It is equal to or greater than 0 KB.
svf_max_vectors	integer	Specifies maximum number of vectors in file. It must be equal to or greater than 0.
script	string	Absolute path of script file. This is an optional parameter.
force_rtg4_otp	boolean	Enforces the use of One-Time Programming (OTP). It is optional. Valid values are: TRUE, true, 1, FALSE, false, 0. Default is 0.

© 2025 Microchip Technology Inc. and its subsidiaries

export_bitstream_file (continued)		
Parameter	**Type**	**Description**
uek2_high_water_mark	integer	High Water Mark (HWM) is required for SPI files only if atleast one locked component (not permanently locked) is selected and OWP is enabled. If OWP is disabled, HWM is disabled too. An error is generated if the HWM is not specified when required. You must keep a track of the HWM and ensure the value is increasing. **Important:** HWM value for UEK1 file is not required. HWM for UEK1 and HWM for UEK2 cannot be equal. The HWM value can be upto 32 HEX characters. 0 is an illegal value.

Security-Related Options

The following table lists the Security-related options.

Important: One of the `trusted_facility file`, `master_file`, `encrypted_uek1_file`, and `encrypted_uek2_file`, or `encrypted_uek3_file` must be set to 1. 1 indicates that this particular file type will be exported; 0 indicates that it will not be exported. For example, if `trusted_facility_file` is set to 1, all other file types must be set to 0.

If `trusted_facility_file` is set to 0, a combination of `master_file` and `uek1_file`, `uek2_file`, and `uek3_file` can be set to 1. In this case, master_file must be set to 1.

Export the bitstream file as you may require the design components saved in the exported bitstream file.

Parameter	Type	Description
trusted_facility_file	boolean	Specifies the bitstream file to be exported. • 1 - Indicates that this particular file type will be exported. • 0 - Indicates that it will not be exported.
trusted_facility_file_components	string	Specifies the components of the design that will be saved to the bitstream file. The default is FABRIC. The value can be: • PolarFire® - one or any combination of FABRIC, SNVM, ENVM, or FABRIC_SNVM. • SmartFusion® 2, IGLOO® 2, and RTG4™ - one or any combination of FABRIC, ENVM.

Zeroization Options:

The following table lists the Zeroization options.

Parameter	Type	Description
zeroization_likenew_action	boolean	Specifies that all data will be erased, and the device can be reprogrammed immediately.
zeroization_unrecoverable_action	boolean	Specifies that all data will be erased, and the device cannot be reprogrammed and must be scrapped.

Custom Security Options

The following table lists the Custom security options.

© 2025 Microchip Technology Inc. and its subsidiaries

Parameter	Type	Description
master_file	boolean	Specifies the bitstream files to be exported. Depends on the selected security. **Important:** If `-master_file` is 1, SECURITY must be selected.
master_file_components	string	Specifies the components in the design that will be saved to the bitstream file. • PolarFire® - SECURITY, FABRIC, SNVM, FABRIC_SNVM, ENVM • SmartFusion® 2, IGLOO® 2, and RTG4™ - SECURITY, FABRIC, ENVM **Important:** 1. The SECURITY option is available in `-bitstream_file_components` only when file type is INITIATOR in `-bitstream_file_type`. 2. SNVM must be programmed with FABRIC. 3. Security-only programming must be performed only on erased or new devices. If performed on device with fabric programmed, the fabric will be disabled after performing security-only programming. You must reprogram the fabric to re-enable it.
encrypted_uek1_file	boolean	Specifies the bitstream file to be exported. Default is 0. Valid values are: • 1 - indicates that this particular file type will be exported. • 0 - indicates that it will not be exported.
encrypted_uek1_file_components	string	Specifies the components of the design that will be saved to uek1 bitstream. The value can be any one or both of FABRIC and ENVM. • PolarFire - FABRIC, ENVM, FABRIC_SNVM, SNVM • SmartFusion 2, IGLOO 2, and RTG4 - FABRIC, ENVM **Important:** sNVM should be programmed with FABRIC.
encrypted_uek2_file	boolean	Specifies the bitstream file to be exported. Default is 0. Valid values are: • 1 - indicates that this particular file type will be exported. • 0 - indicates that it will not be exported.
encrypted_uek2_file_components	string	Specifies the components of the design that will be saved to uek2 bitstream. • PolarFire - FABRIC, ENVM, FABRIC_SNVM, SNVM. • SmartFusion 2, IGLOO 2 and RTG4 - FABRIC, ENVM. **Important:** sNVM should be programmed with FABRIC.

© 2025 Microchip Technology Inc. and its subsidiaries

export_bitstream_file (continued)		
Parameter	**Type**	**Description**
encrypted_uek3_file	boolean	Specifies the bitstream file to be exported. Valid values are: • 1 - indicates that this particular file type will be exported. • 0 - indicates that it will not be exported.
encrypted_uek3_file_components	string	Specifies the components of the design that will be saved to uek3 bitstream. The value can be any one or both of FABRIC and ENVM.
master_include_plaintext_passkey	boolean	Specifies that the Initiator file includes plain text passkey. This argument is optional.
uek1_include_plaintext_passkey	boolean	Specifies that uek1 includes plain text passkey. This argument is optional.
uek2_include_plaintext_passkey	boolean	Specifies that uek2 includes plain text passkey. This argument is optional.
uek3_include_plaintext_passkey	boolean	Specifies that uek3 includes plain text passkey. This argument is optional.

Bypass Back Level Protection Options

The following table lists the Bypass Back Level Protection options. These options are only supported by the SPI bitstream files. Export the bitstream file as you may require the design components saved in the exported bitstream file.

Parameter	Type	Description
master_backlevel_bypass	boolean	Specifies the Bypass Back Level protection for Golden/Recovery bitstream if back level protection is enabled in _master file.
uek1_backlevel_bypass	boolean	Specifies the Bypass Back Level Protection for Golden/Recovery bitstream if back level protection is enabled in _uek1 file.
uek2_backlevel_bypass	boolean	Specifies the Bypass Back Level Protection for Golden/Recovery bitstream if back level protection is enabled in _uek2 file.

SPI-Related Options

The following table lists the SPI-related options. These are optional.

Parameter	Type	Description
add_golden_image	boolean	• 1 - To enable golden SPI image in SPI direct. • 0 - To disable golden SPI image in SPI direct.
golden_image_address	string	32-bit hexadecimal address for golden image.
golden_image_design_version	string	Decimal value for golden image design version.
add_update_image	boolean	• 1 - To enable golden update SPI image. • 0 - To disable golden update SPI image.
update_image_address	string	Hexadecimal value for update image address.
update_image_design_version	string	Decimal value for update image design version.

Serialization Options

The following table lists the serialization options. These are optional.

Parameter	Type	Description
serialization_stapl_type	string	Serialization stapl file type either SINGLE or MULTIPLE. Default is SINGLE.
serialization_target_solution	string	Target programming hardware – Flashpro_3_4_5 or generic_STAPL_player. Default is Flashpro_3_4_5.

Advanced Options

© 2025 Microchip Technology Inc. and its subsidiaries

The following table lists the advanced options. These options are available for PolarFire SoC device family only.

Parameter	Type	Description
trusted_facility_keep_fabric_operational	boolean	Specifies to keep fabric in operational state during programming if file is programmed at a trusted facility. Valid values are: TRUE, true, 1, FALSE, false, 0.
trusted_facility_skip_startup_seq	boolean	Specifies to skip device start-up sequence after programming if file is programmed at a trusted facility and `trusted_facility_keep_fabric_operational` is TRUE. Valid values are: TRUE, true, 1, FALSE, false, 0.
uek1_keep_fabric_operational	boolean	Specifies to keep fabric in operational state during programming if file is encrypted using UEK1 custom security. Valid values are: TRUE, true, 1, FALSE, false, 0.
uek1_skip_startup_seq	boolean	Specifies to skip device start-up sequence after programming if file is programmed using UEK1 custom security and `uek1_keep_fabric_operational` is TRUE. Valid values are: TRUE, true, 1, FALSE, false, 0.
uek2_ keep_fabric_operational	boolean	Specifies to keep fabric in operational state during programming if file is programmed using UEK2 custom security. Valid values are: TRUE, true, 1, FALSE, false, 0.
uek2_ skip_startup_seq	boolean	Specifies to skip device start-up sequence after programming if file is programmed using UEK2 custom security and `uek2_keep_fabric_operational` is TRUE. Valid values are: TRUE, true, 1, FALSE, false, 0.

Error Codes

Error Code	Description
None	Warning: SPI Directory options are not supported for RTG4™ devices and will be ignored.
None	Incorrect Update SPI Image address format. Address must be 32-bit HEX number.
None	Golden SPI Image parameters are required to export SPI directory.
None	Export SPI files is not supported. Export SVF files is not supported.
None	You have not configured custom security options. You can only export bitstream files to program at trusted facility.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Export bitstream file for design with default security:

```
export_bitstream_file \
-trusted_facility_file 1 \
-trusted_facility_file_components {FABRIC}
```

Export bitstream file for design with custom security options.

© 2025 Microchip Technology Inc. and its subsidiaries

Export bitstreams to Initiator, uek1 and uek2 encrypted files. Initiator file to include security, fabric components and Export Pass Key in Plaintext, uek1 and uek2 encrypted files to include FABRIC with **Like new Zeroization** option enabled.

```
export_bitstream_file\
-file_name {fftousram_new} \
-export_dir
{X:\10_docs_review\pf2.2_sp1\Programming_sars\99412\clkint_fftousram_ac_latch_launch\des
igner\fftousram_new\export} \
-format {PPD DAT STP HEX} \
-for_ihp 1 \
-master_file 1 \
-master_file_components {SECURITY FABRIC} \
-encrypted_uek1_file 1 \
-encrypted_uek1_file_components {FABRIC} \
-encrypted_uek2_file 1 \
-encrypted_uek2_file_components {FABRIC} \
-trusted_facility_file 0 \
-trusted_facility_file_components {} \
-zeroization_likenew_action 1 \
-zeroization_unrecoverable_action 0 \
-master_backlevel_bypass 0 \
-uek1_backlevel_bypass 0 -uek2_backlevel_bypass 0 \
-master_include_plaintext_passkey 1 \
-uek1_include_plaintext_passkey 0 \
-uek2_include_plaintext_passkey 0
```

The following example, intented for SmartFusion 2 and IGLOO 2 families, exports SPI directory for programming recovery:

```
export_bitstream_file \
-add_golden_image 1 \
-golden_image_address {1111} \
-golden_image_design_version {1} \
-add_update_image 1 \
-update_image_address {1211} \
-update_image_design_version {1}
```

The following example exports bitstream file for design with MSS/serialization clients. This example failed in case of PolarFire family.

```
export_bitstream_file \
-file_name {mss1} \
-format {STP} \
-trusted_facility_file 1 \
-trusted_facility_file_components {FABRIC} \
-serialization_stapl_type {SINGLE} \
-serialization_target_solution {FLASHPRO_3_4_5}
```

3.33. export_bsdl_file (Ask a Question)

Description

This Tcl command exports the BSDL to a specified file. The BSDL file provides a standard file format for electronics testing using JTAG. It describes the boundary scan device package, pin description and boundary scan cell of the input and output pins. BSDL models are available as downloads for many Microchip SoC devices. The exported file has a *.bsd file name extension.

```
export_bsdl_file -file {absolute or relative path and name of BSDL file}
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name of the *.bsd file. If the specified file path is missing, the file is created in the <project_name>/designer/<design_name> directory.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the BSDL to a sd1.bsd file.

```
export_bsdl_file -file {E:/designs/export/sd1.bsd}
```

3.34. export_component_to_tcl (Ask a Question)

Description

This Tcl command exports the Tcl command for the selected component. The components can be SmartDesign components, configured cores and HDL+ cores.

```
export_component_to_tcl -component component_name \
                        [-library "library_name" ] \
                        [-package "package_name" ] \
                        [-recursive "TRUE | FALSE"] \
                        [-folder "folder"] \
                        [-file "file_path"]
```

Arguments

Parameter	Type	Description
component	string	Specifies the name of the component for which the Tcl command is exported. It is mandatory.
library	string	Specifies the name of the library the component belongs to. It is optional.
package	string	Specifies the name of the package the HDL+core belongs to. It is optional.
recursive	boolean	Specifies if a SmartDesign component needs to be exported recursively. It is optional.
file	string	Specifies the path where you wish to export the Tcl file. It is mandatory.

Error Codes

Error Code	Description
None	Please specify a file path for the 'file' parameter.
None	Required parameter 'component' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_component_to_tcl -component "component name" [-library "library name"] [-package "package name"] [-file "file"] [-recursive "TRUE | FALSE"] [-folder "folder"] '.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the Tcl command for "work" library "pattern_gen_checker" component.

```
export_component_to_tcl -component {pattern_gen_checker} \
                        -library {work} -package {} folder {} \
                        -file {./pattern_gen_checker.tcl}
```

3.35. export_design_summary (Ask a Question)

Description

This Tcl command exports an HTML file containing information about your root SmartDesign in your project. The HTML report provides information on:

- Generated Files
- I/Os
- Hardware Instances
- Firmware
- Memory Map

```
export_design_summary -file {absolute or relative path and name of HTML file}
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name of file where you wish to export the HTML file. It is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example exports an HTML to specified file which contains information about your root SmartDesign.

```
export_design_summary -file {D:/Designs/test/sd1.html}
```

3.36. export_firmware (Ask a Question)

Description

This Tcl command exports design firmware configuration data, which consists of:

- Component configuration for MSS/HPMS, FDDR and SERDES blocks instantiating your design.
- Compatible firmware drivers for your peripherals

It also creates a workspace and project specific to the IDE tool of your choice (SoftConsole, Keil or IAR).

To open your exported firmware projects, you must invoke the third-party development tool (SoftConsole, Keil or IAR) outside Libero SoC.

If you make any changes to your design, you must re-export firmware.

```
export_firmware -export_dir {absolute or relative path} \
                [-create_project {0|1}] \
                [-software_ide {SoftConsole | Keil | IAR EWARM}]
```

Arguments

Parameter	Type	Description
export_dir	string	Specifies absolute or relative path and name of folder for the exported firmware. Default exported firmware created<project_name>/firmware direcotry.
create_project	boolean	Generates the workspace and project for the specified IDE tool. Default is 0. Valid values are: TRUE, 1, true, FALSE, 0 or false.
software_ide	string	Specifies one of three IDE tool name: SoftConsole | IAR EWARM | Keil. If you use `-create_project` parameter and `-software_ide` paramter at the same time, Libero exports the workspace and project for that Software IDE tool to the `export_path {SoftConsole|Keil|IAR}` folder. Default is IAR EWARM.

Error Codes

Error Code	Description
None	Required parameter 'export_dir' is missing.
None	Parameter 'export_dir' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_firmware -export_dir "Export directory" [-create_project "TRUE | FALSE"] [-software_ide "Software IDE"] ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following command export design firmware configuration data, generates the workspace and project for the SoftConsole IDE tool.

```
export_firmware \
-export_dir {D:\Designs\software_drivers} \
-create_project {1} \
-software_ide {SoftConsole}
```

3.37. export_fp_pdc (Ask a Question)

Description

This Tcl command exports the Floorplanning Physical Design Constraint (*.pdc) File. You can export the Floorplan PDC file from Constraint Manager > I/O Attributes or Constraint Manager Floor Planner or from File menu. Constraints can be exported to PDC file for reference, but must be manually added to an existing PDC or imported via the Constraints Editor for the changes to affect the final paced and routed design. The exported file has *.pdc file name extension. Before exporting, you need to run 'Place and Route'.

```
export_fp_pdc \
         -file { absolute or relative path and name of *.pdc file } \
         [-mode { PDC_PLACE | PDC_FULL_PLACEMENT }]
```

Arguments

Parameter	Type	Description
file	string	Specifies absolute or relative path and name of the *.pdc file. It is mandatory.
mode	string	Choose the type of information that you want to export. Use `PDC_PLACE` to export user's floorplanning constraints, for example, fixed logic and regions. Use `PDC_FULL_PLACEMENT` to export information about all of the physical design constraints (I/O constraints, I/O Banks, routing constraints, region constraints, global and local clocks). This is an optional parameter. Default is `PDC_PLACE`.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_fp_pdc -file "Export File Name" [-mode "Export Mode"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example exports information about all of the physical designs constraints (I/O constraints, I/O Banks, routing constraints, region constraints, global and local clocks).

```
export_fp_pdc -file {E:/designs/export/sd1_fp.pdc} \
              -mode {PDC_FULL_PLACEMENT}
```

See Also

- export_io_pdc

3.38. export_ibis_file (Ask a Question)

Description

This Tcl command exports the IBIS (Input/Output Buffer Information Specification) model report. The IBIS model report provides an industry-standard file format for recording parameters like driver output impedance, rise/fall time, and input loading, which may then be used by software applications such as Signal Integrity tools or IBIS simulators. The exported file has a *.ibs(name <root>.ibs) file name extension.

```
export_ibis_file -file {absolute or relative path and name of *.ibs file}
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name of IBIS file to export. If the file path is not specified, the file is created in `<project_name>/designer/<design_name>` directory.

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the IBIS (Input/Output Buffer Information Specification) model report.

```
export_ibis_file -file {E:/designs/export/sd1.ibs}
```

3.39. export_interrupt_map (Ask a Question)

This Tcl command exports the interrupt connectivity map of a chosen SmartDesign to a file. As input, it takes the file path to export the `.json` file in a particular location and the SmartDesign name for which you want to export the interrupt report.

© 2025 Microchip Technology Inc. and its subsidiaries

Tip: This command can be executed for all families without receiving any error messages. However, the exported data might only be of assistance for the PolarFire SoC family.

```
export_interrupt_map -file {C:\Users\name\Desktop\tmp.json} -sd_name topSD
        -file : exported file path/name
        -sd_name : SmartDesign name, for which the connectivity map will be constructed
```

Arguments

Parameter	Type	Description
file	String	Specifies the exported file path/name.
sd_name	String	Specifies the SmartDesign name, for which the connectivity map will be constructed.

Return Type	Description
None	None

Error Codes

Error Code	Description
Error: SmartDesign 'DesignName' doesn't exist. Please specify a valid SmartDesign component name.	When specifying wrong SmartDesign name in the `sd_name` parameter.
Error: Invalid extension specified for the report. Please specify a valid extension. Valid extension is \'json\'.	When the specified file extension is not correct.
Error: The report cannot be exported. Please check the specified file path and write permissions to it.	When the file is not possible to open (file path is wrong, we do not have write permission).
Error: Unable to load SmartDesign 'DesignName' model, please provide a valid SmartDesign.	When the specified SmartDesign is impossible to load
Error: 'DesignName' is not a SmartDesign component. Please specify a valid SmartDesign component name.	When the specified SmartDesign component type is not valid (is test bench, IP core, etc)
Info: Successfully exported Interrupt Map Report file to: \'FileName\'\n".	When the JSON file is successfully exported.
Please specify a valid file path and name.	If the "File Name" label is empty in pop-up dialog.
Please specify a valid extension (`.json`) for the report.	If the "File Name" label doesn't have correct extension in pop-up dialog.
Please specify a valid SmartDesign component name.	If the "SmartDesign Name" label is empty in pop-up dialog.

Example

This example exports the interrupt connectivity map of the chosen SmartDesign to a file.

```
export_interrupt_map -file {C:\Users\name\Desktop\tmp.json} -sd_name topSD
```

3.40. export_io_pdc (Ask a Question)

Description

This Tcl command exports the I/O Physical Design Constraint (*.pdc) File. You can export the I/O PDC file from Constraint Manager > I/O Attributes or Constraint Manager Floor Planner or from File menu. Constraints can be exported to PDC file for reference, but must be manually added to an existing PDC or imported via the Constraints Editor for the changes to affect the final paced and routed design. The exported file has *.pdc file name extension. Before exporting, you need to run 'Place and Route'.

```
export_io_pdc -file { absolute or relative path and name of *.pdc file }
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
file	string	Specifies absolute or relative path and name of the *.pdc file. It is mandatory. There may be multiple -file arguments (see example below).

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_io_pdc -file "Export IO PDC File Name" '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example exports information about all of the physical design constraints (I/O constraints, I/O Banks, routing constraints, region constraints, global and local clocks). Created both "sd_io1.pdc" and "sd_io2.pdc" files.

```
export_io_pdc -file {./sd_io1.pdc} -file {sd_io2.pdc}
```

See Also

- export_fp_pdc

3.41. export_job_data (Ask a Question)

Description

This Tcl command configures the parameters for the Job Manager Data Container (JDC) file to be exported from Libero and used by Job Manager. The exported file has a `*.jdc` file name extension. All parameters are optional. Default values are used if parameters are omitted.

```
export_job_data \
         [-file_name "filename"] \
         [-export_dir {absolute or relative path of the exported file}] \
         [-components "SECURITY | FABRIC | ENVM"] \
         [-include_spi_flash "value"]
```

Arguments

Parameter	Type	Description
file_name	string	Specifies the name of the file that will be exported. The default file name is the design name.
export_dir	string	Specifies the directory location for the export. By default, the file is saved in the `Libero projects/designer/<design_name>/export` directory with `*.jdc` file extension.

© 2025 Microchip Technology Inc. and its subsidiaries

export_job_data (continued)		
Parameter	**Type**	**Description**
components	string	Specifies the components of the design that will be saved to the file. The value can be any one or a combination of `SECURITY`, `FABRIC`, `SNVM`, and `ENVM` if they are available in the design. If the parameter is omitted, all available components of the design will be saved. **Important:** The SECURITY component must be selected if user security is initialized for the current Libero® design.
include_spi_flash	boolean	Use the Configure Design Initialization Data and Memories tool to configure this option. Valid value is 0 and 1. Default is 0.

Error Codes

Error Code	Description
None	PolarFire®: No bitstream components or SPI Flash data are selected to include in exported programming data file.
None	PolarFire: SPI Flash Memory is not configured. Use the Configure Design Initialization Data and Memories tool to configure it.
None	IGLOO® 2, PolarFire: The eNVM component is not available in the current design.
None	SmartFusion® 2: The eNVM component is not supported in the current design.
None	PolarFire: Invalid argument value. Expecting SECURITY, FABRIC, SNVM, ENVM or FABRIC_SNVM.
None	SmartFusion components: Invalid argument value. 'FABRIC_SNVM' (expecting SECURITY, FABRIC or ENVM)
None	SmartFusion 2, IGLOO 2: Parameter 'include_spi_flash' is not defined. Valid command formatting is 'export_job_data [-file_name "file_name"] \ [-export_dir "export_dir"] \ [-components "[SECURITY | FABRIC | ENVM]+"]
None	SmartFusion 2, IGLOO 2: There are no bitstream components to include in exported programming data file.
None	IGLOO 2: Invalid argument value: 'FM' (expecting SECURITY, FABRIC or ENVM).
None	include_spi_flash: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_job_data [-file_name "file_name"] \ [-export_dir "export_dir"] \ [-components "[SECURITY | FABRIC | SNVM | ENVM | FABRIC_SNVM]+"] \ [-include_spi_flash "TRUE | FALSE"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example exports Job Manager Data Container file (sd1.jdc) with FABRIC component.

```
export_job_data \
-file_name {sd1} \
-export_dir {D:\sd_prj\test3T\designer\sd1\export} \
-components {FABRIC}
```

3.42. export_netlist_file (Ask a Question)

Description

This Tcl command exports the netlist after the compile state has completed. The netlist can be either Verilog or VHDL. Microchip recommends exporting the netlist after the compile state has successfully completed.

```
export_netlist_file \
         -file { absolute or relative path and filename for netlist } \
         [-vhdl { value }]
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name of netlist file. If specified relative path of netlist file created in <project_name>/designer/<design_name> directory.
vhdl	boolean	Generates the netlist in VHDL (when set to "TRUE" or 1) or Verilog (when set to "FALSE" or 0). Default is 0 (Verilog netlist).

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	vhdl: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_netlist_file -file "Export Netlist" \ [-vhdl "TRUE | FALSE"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the netlist (Verilog) after the compile state has completed.

```
export_netlist_files -file {E:/designs/export/sd1/sd1.v} -vhdl 0
```

3.43. export_parameter_report (Ask a Question)

Description

This Tcl command generates a report in respective format (XML) selected with the list of components in the design along with their Tcl parameters that were used to configure. Also

© 2025 Microchip Technology Inc. and its subsidiaries

the report shows the available latest versions of the cores used in the design in either vault or repositories selected. This command supports generating parameter report for only that active top module which is not instantiated in any other module.

```
export_parameter_report [-file filename_with_extension] \
                        [-top top_module_name]
```

Arguments

Parameter	Type	Description
file	string	Name of the output generated report in xml format. If filename with extension is not provided, the report will be generated as "top_module_name_parameter_report.xml". This argument is optional.
top	string	Name of the top module which is not instantiated in any other module. This overrides the active top module.If the top module name is not provided, then the report for the active root module will be generated. This argument is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter/Generic report cannot be exported for a module that is instantiated in any other module. It can only be exported for top level modules. Please set a top level module as the root or specify a top level module in the Tcl command export_parameter_report to be able to export the report.
None	Parameter 'top' has illegal value.
None	Parameter 'file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_parameter_report [-file "file name"] [-top "top name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example reports into {./src/sd1_parameter_report.xml} file.

```
export_parameter_report -file {./src/sd1_parameter_report.xml} -top top
```

3.44. export_pin_reports (Ask a Question)

Description

This Tcl command configures and exports a pin report file to a specified folder/directory location. The pin report lists the pins in your device sorted according to your preference: sort by Port Name or Sorted by Package Pin Name. The pin report generates two files:

- <design>_pinrpt_name.rpt - pin report sorted by name.

© 2025 Microchip Technology Inc. and its subsidiaries

- <design>_pinrpt_number.rpt - pin report sorted by pin number.

Export Pin Report generates a Bank Report by default; the filename is `<design>-bankrpt.rpt`. Export Pin Report also generates an I/O Register Combining Report listing the I/Os which have been combined into a Register for getting timing performance. You must select at least one report.

```
export_pin_reports -export_dir {absolute path to folder location} \
                   [-pin_report_by_name {value}] \
                   [-pin_report_by_pkg_pin {value}] \
                   [-bank_report {1|0}] \
                   [-io_report {1|0}]
```

Arguments

Parameter	Type	Description
export_dir	string	Specifies the folder, disk location where you want to save pin report. It is mandatory.
pin_report_by_name	integer	Set to 1 to have the pin report sorted by pin name. By default, this box is checked.
pin_report_by_pkg_pin	integer	Set to 1 to have pin report sorted by package pin number, 0 to not sort by package pin number. By default, this box is checked.
bank_report	boolean	Set to 1 to generate the I/O bank report, 0 to not generate the report. By default, this box is checked.
io_report	boolean	Set to 1 to generate the I/O report, 0 to not generate the report. By default, this box is checked.

Error Codes

Error Code	Description
None	Required parameter 'export_dir' is missing.
None	pin_report_by_name: Invalid argument value: 'pin_name' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_pin_reports -export_dir "Export Dir Name" \ [-pin_report_by_name "TRUE | FALSE"] \ [-pin_report_by_pkg_pin "TRUE | FALSE"] \ [-bank_report "TRUE | FALSE"] \ [-io_report "TRUE | FALSE"] \ [-defvar "user def variables"]* '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports pin report sorted by pin name and not sorted by package pin number, generated I/O and I/O bank reports.

```
export_pin_reports \
-export_dir {E:/designs/export} \
-pin_report_by_name {1} \
-pin_report_by_pkg_pin {0} \
-bank_report {1} \
-io_report {1}
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.45. export_profiles (Ask a Question)

Description

This Tcl command exports your tool profiles. Performs the same action as the **Export Profiles** dialog box.

```
export_profiles -file { absolute path and name of exported file } \
                [-export "value" ]
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute path and name to the exported profile file.
export	string	Specifies your profile export options. The following table shows the acceptable values for this argument: • predefined - exports only predefined profiles • user - exports only user profiles • all - exports all profiles

Error Codes

Error Code	Description
None	Parameter 'file' : the file '/prj/profiles_command/exp.tcl' has got an invalid extension. Valid extension is 'ini'.
None	Required parameter 'file' is missing.
None	Parameter 'file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_profiles -file "file name" [-export "predefined | user | all"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports all profiles to the file './profile_commands/all_profiles'.

```
export_profiles -file "./profile_commands/all_profiles" -export all
```

See Also

- add_profile
- edit_profile
- remove_profile
- select_profile

© 2025 Microchip Technology Inc. and its subsidiaries

3.46. export_prog_job (Ask a Question)

Description

This Tcl command configures the parameters for the FlashPro Express programming job to be exported.

Note: The Programming Mode (JTAG/SPI-Slave) setting from the Programming Connectivity and Interface tool will be exported in the job file.
RTG4 devices do not support the security options supported by SmartFusion 2 and IGLOO 2 devices.

The syntax for the export programming job Tcl command for SmartFusion 2 and IGLOO 2 is shown below:

```
export_prog_job \
-job_file_name {file} \
-export_dir {absolute or relative path} \
-bitstream_file_type {TRUSTED_FACILITY | MASTER | UEK1 | UEK2} \
-bitstream_file_components {SECURITY | FABRIC | ENVM} \
-include_plaintext_passkey {0 | 1} \
-design_bitstream_format {PPD | STP} \
-prog_optional_procedures \
{action1 | procedure1 | procedure2 ; action2 | procedure1 | procedure2 | procedure3;}
```

The syntax for the export programming job Tcl command for RTG4 is below:

```
export_prog_job \
-job_file_name {file} \
-export_dir {dir} \
-force_rtg4_otp {0 | 1} \
-design_bitstream_format {PPD | STP}
```

The syntax for the export programming job Tcl command for PolarFire is below:

```
export_prog_job \
-job_file_name {file} \
-export_dir {dir} \
-bitstream_file_type {TRUSTED_FACILITY | MASTER | UEK1 | UEK2} \
-bitstream_file_components {SECURITY | ENVM | FABRIC | SNVM | FABRIC_SNVM} \
-zeroization_likenew_action {0 | 1} \
-zeroization_unrecoverable_action {0 | 1} \
-program_design {0 | 1} \
-program_spi_flash {0 | 1} \
-include_plaintext_passkey {0 | 1} \
-design_bitstream_format {PPD | STP} \
-prog_optional_procedures \
{action1|procedure1|procedure2;action2|procedure1|procedure2|procedure3;} \
-skip_recommended_procedures \
{action1 | procedure1 | procedure2 ; action2 | procedure1 | procedure2 | procedure3;}
```

Arguments

Parameter	Type	Description
job_file_name	string	The name of the exported file. Name must start with design name. If omitted, design name will be used.
export_dir	string	Location where the job file will be saved; any folder can be specified (absolute or release path). The default folder is the Libero designer/<design_name>/export folder.
force_rtg4_otp	boolean	Enforces the use of one-time programming (OTP). This argument is optional. Valid values are: TRUE, 1, true, FALSE, 0 or false. The default value is 0.
bitstream_file_type	string	Bitstream file to be included in the programming job. Only one bitstream file can be included in a programming job. Possible values are: TRUSTED_FACILITY, MASTER, UEK1, UEK2 or UEK3. If omitted, type will be TRUSTED_FACILITY.

© 2025 Microchip Technology Inc. and its subsidiaries

export_prog_job (continued)

Parameter	Type	Description
bitstream_file_components	list of components	The list of components to be included in the programming job. Components should be delimited by space. `bitstream_file_components` can be any one of SECURITY, ENVM, FABRIC SNVM (only PolarFire) or FABRIC_SNVM (only PolarFire) any combination of them. **Notes:** 1. The SECURITY option is available in `-bitstream_file_components` only when file type is MASTER in `-bitstream_file_type`. 2. SNVM must always be programmed with FABRIC. 3. Security-only programming must be performed only on erased or new devices. If performed on a device with fabric programmed, the fabric will be disabled after performing security-only programming. You must reprogram the fabric to re-enable it.
include_plaintext_passkey	boolean	Includes plaintext passkey. Valid values are: TRUE, 1, true, FALSE, 0 or false. This argument is optional. Default is 0.
design_bitstream_format	string	Specifies the bitstream file formats to be exported. Space is used as a delimiter. The value can be any one of PPD, STP. If omitted, the bitstream file will be in PPD format. • PPD will allow for improved programming times with FlashPro6 programmer. • Use STAPL for static algorithm and data from release to release.
prog_optional_procedures	string	Specifies optional procedures to program. Format: action1 | procedure1 | procedure2; action2 | procedure1 | procedure2 | procedure3; Available actions and procedures depend on the selected bitstream components. Mandatory procedures are always included. See the section Configure Actions and Procedures for supported actions and procedures.
zeroization_likenew_action	boolean	Specifies that all data will be erased and the device can be reprogrammed immediately. Valid values are TRUE, 1, true, FALSE, 0 or false. Default is 0.
zeroization_unrecoverable_action	boolean	Specifies that all data will be erased. The device cannot be reprogrammed and it must be scrapped. Valid values are TRUE, 1, true, FALSE, 0 or false. Default is 0.
program_design	boolean	Specifies to program the design. This argument is optional. Valid values are TRUE, 1, true, FALSE, 0 or false. Default is 1.
program_spi_flash	boolean	Specifies to program SPI Flash. Configure it before using. Use the Configure Design Initialization Data and Memories tool to configure it. This argument is optional. Valid values are TRUE, 1, true, FALSE, 0 or false. Default is 0.
skip_recommended_procedures	string	Specifies recommended procedures to skip. Format: action1 | procedure1 | procedure2; action2 | procedure1 | procedure2 | procedure3; See the section Configure Actions and Procedures for supported actions and procedures.

Return Type	Description
integer	Returns 0 on success, 1 on failure.

Error Codes

Error Code	Description
None	bitstream_file_type: Invalid argument value: 'A' (expecting MASTER, UEK1, UEK2, UEK3 or TRUSTED_FACILITY).
None	bitstream_file_components: Invalid argument value: 'A' (expecting SECURITY, FABRIC, SNVM, ENVM or FABRIC_SNVM).
None	Custom security component is available for Master bitstream file only.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

SmartFusion 2/IGLOO 2

The following example exported FlashPro Express programming job in X:\12.0_Release\g4_fftousram\designer\top\export\top file. Included MASTER Bitstream file with PPD format, SECURITY and FABRIC components, plaintext passkey and specified PROGRAM, DO_VERIFY; optional procedures to program:

```
export_prog_job \
-job_file_name {top} \
-export_dir {X:\12.0_Release\g4_fftousram\designer\top\export} \
-bitstream_file_type {MASTER} \
-bitstream_file_components {SECURITY FABRIC} \
-include_plaintext_passkey 1 \
-design_bitstream_format {PPD} \
-prog_optional_procedures {PROGRAM | DO_VERIFY;}
```

RTG4

The following example exported FlashPro Express programming job in X:\12.0_Release\rtg4_ff_usram\designer\top\export\top file with PPD format and also enforces the use of one-time programming (OTP):

```
export_prog_job \
-job_file_name {top} \
-export_dir {X:\12.0_Release\rtg4_ff_usram\designer\top\export} \
-force_rtg4_otp 1 \
-design_bitstream_format {PPD}
```

PolarFire

The following example exported FlashPro Express programming job in X:\12.0_Release\pf_fftousram_ac_latch_launch\designer\fftousra\fftousram file. Included MASTER Bitstream file with PPD format, SECURITY, FABRIC and SNVM components, plaintext passkey and specified PROGRAM, DO_VERIFY; optional procedures to program, programs the design and device is not reprogrammed:

```
export_prog_job \
-job_file_name {fftousram} \
-export_dir {X:\12.0_Release\pf_fftousram_ac_latch_launch\designer\fftousram\export} \
-bitstream_file_type {MASTER} \
-bitstream_file_components {SECURITY FABRIC SNVM} \
-zeroization_likenew_action 0 \
-zeroization_unrecoverable_action 0 \
-program_design 1 \
-program_spi_flash 0 \
-include_plaintext_passkey 0 \
-design_bitstream_format {PPD} \
-prog_optional_procedures {PROGRAM | DO_VERIFY;} \
-skip_recommended_procedures {VERIFY_DIGEST | DO_ENABLE_FABRIC;}
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.47. export_script (Ask a Question)

Description

This Tcl command explicitly exports the Tcl command equivalents of the current Libero session. With this command you can re-execute the same commands interactively or in batch.

You must supply a file name with the `-file` parameter and the `-relative_path` parameter to specify whether an absolute or relative path is used in the exported script file.

```
export_script \
-file { absolute or relative path to constraint file } \
-relative_path <value>
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name to the exported script file.
relative_path	boolean	Sets your option to use a relative or absolute path in the exported script; use 1 for relative path, 0 for absolute.

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'relative_path' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the Tcl command equivalents of the current Libero session.

```
export_script -file {./exported.tcl} -relative_path 1
```

3.48. export_sdc_file (Ask a Question)

Description

This Tcl command exports the file for timing constraints. The exported file has a `*.sdc` file name extension.

```
export_sdc_file -file { absolute path and name of *.sdc file }
```

Arguments

Parameter	Type	Description
file	string	Specifies the SDC file to export.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_sdc_file -file "Export SDC File Name"' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports the SDC file for timing constraints to sd1.sdc.

```
export_sdc_file -file {E:/designs/export/sd1.sdc}
```

3.49. export_smart_debug_data (Ask a Question)

Description

This Tcl command exports debug data for the SmartDebug application and creates *.ddc that contains data based on selected options. The command corresponds to the Export SmartDebug Data tool in Libero. This file is used by SmartDebug (standalone application) to create a new SmartDebug project, or it can be imported into a device in SmartDebug (standalone application).

Important:

- If you do not specify any design components, all components available in the design will be included by default except the bitstream components.
- The generate_bitstream parameter is required if you want to generate bitstream file and include it in the exported file.
- You must specify the bitstream components you want to include in the generated bitstream file or all available components will be included.
- If you choose to include bitstream, and the design has custom security, the custom security bitstream component must be included.

```
export_smart_debug_data [-file_name "file_name"] \ [-export_dir "export_dir"] \ [-
probes "TRUE | FALSE"] \ [-package_pins "TRUE | FALSE"] \ [-memory_blocks "TRUE
| FALSE"] \ [-envm_data "TRUE | FALSE"] \ [-security_data "TRUE | FALSE"]
\ [-display_security_in_smartdebug "TRUE | FALSE"]\ [-chain "TRUE | FALSE"] \ [-
programmer_settings "TRUE | FALSE"] \ [-ios_states "TRUE | FALSE"] \ [-generate_bitstream
"TRUE | FALSE"] \ [-bitstream_format "PPD | STP"] \ [-bitstream_security "TRUE | FALSE"] \ [-
bitstream_fabric "TRUE | FALSE"] \ [-bitstream_envm "TRUE | FALSE"] \ [-bitstream_snvm "TRUE
| FALSE"] \ [-snvm_data "TRUE | FALSE"] \ [-sanitize_snvm "TRUE | FALSE" ] \ [-sanitize_envm
"TRUE | FALSE" ] \ [-master_include_plaintext_passkey "TRUE|FALSE"]\ [-uprom_data "TRUE |
FALSE"] \ [dpk_security "TRUE | FALSE"] \ [upk1_security "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
file_name	string	Name of the exported file with extension `.ddc`.

© 2025 Microchip Technology Inc. and its subsidiaries

export_smart_debug_data (continued)

Parameter	Type	Description
export_dir	string	Location where the DDC file will be exported. If omitted, the design export folder will be used.
probes	boolean	Specified probes related data to be included in the DDC file, data used for Active/Live
package_pins	boolean	Specifes probe insertion related data to be included in the DDC file.
memory_blocks	boolean	Specifes memory blocks such as LSRAM and USRAM related data to be included in DDC file.
security_data	boolean	Specifies security keys related data.
display_security_in_smartdebug	boolean	If security_data is set to 1, then this option is included to hide/show passkeys on SmartDebug main window **option provided in 12.4 release**.
chain	boolean	This option is set if user wants to connect the devices in chain.
programmer_settings	boolean	Programmer related settings from Libero to be included in DDC file.
io_states	boolean	I/O states information preserved during programming to be included in DDC file.
snvm_data	boolean	Set to 1 if snvm debug related information needs to be included in the DDC file. **Note:** This parameter is only applicable to PolarFire, PolarFire SoC and RT PolarFire devices.
envm_data	boolean	Set to 1 if envm debug related information needs to be included in DDC file. Applicable to PF SOC device only. Option provided from 12.5 release onwards.
generate_bitstream	boolean	Bitstream content to be included in the DDC file.
bitstream_format	string	Bitstream generated format option. The valid values are: PPD, STAPL.
bitstream_security	boolean	Bitstream security pass keys information needed in bitstream to be included along with bitstream.
master_include_plaintext_passkey	boolean	This option indicates whether plaintext passkey needs to be included for REPROGRAM/ERASE action in SASD.
bitstream_fabric	boolean	Bitstream fabric component included in bitstream.
bitstream_snvm	boolean	If snvm component needs to be included in the bitstream. Applicable to PF, PFSOC, RTPF family only.
bitstream_envm	boolean	If envm component needs to be included in the bitstream. Applicable to SF2, PFSOC family only.
sanitize_snvm	boolean	If snvm has to be sanitized during ERASE action. Applicable to PF, PFSOC only, providede in 2021.2 release.
sanitize_envm	boolean	If envm has to be sanitized during ERASE action. Applicable to PFSOC only, provided in 2021.2 release.
dpk_security	boolean	Specifies to include the debug pass key security information for debug.
upk1_security	boolean	Specifies to include the User Passcode Key 1 security information for debug.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	probes: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).

© 2025 Microchip Technology Inc. and its subsidiaries

export_smart_debug_data (continued)	
Error Code	**Description**
None	package_pins: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	memory_blocks: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	security_data: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	chain: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	programmer_settings: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	ios_states: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	generate_bitstream: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	bitstream_security: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	bitstream_fabric: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	bitstream_snvm: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, or false).
None	Parameter 'param_names' is not defined. Valid command formatting is'export_smart_debug_data [-file_name "file_name"] \[-export_dir "export_dir"] \ [-probes "TRUE | FALSE"] \[-package_pins "TRUE | FALSE"] \[-memory_blocks "TRUE | FALSE"] \[-envm_data "TRUE | FALSE"] \[-security_data "TRUE | FALSE"] \[-display_security_in_smartdebug "TRUE | FALSE"] \[-chain "TRUE | FALSE"] \[-programmer_settings "TRUE | FALSE"] \[-ios_states "TRUE | FALSE"] \[-generate_bitstream "TRUE | FALSE"] \[-bitstream_format "PPD | STP"] \ [-bitstream_security "TRUE | FALSE"] \[-bitstream_fabric "TRUE | FALSE"] \ [-bitstream_envm "TRUE | FALSE"] \[-bitstream_snvm "TRUE | FALSE"] \[-master_include_plaintext_passkey "TRUE | FALSE"] \[-uprom_data "TRUE | FALSE"] \[-snvm_data "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following examples show the export_smart_debug_data command with all parameters.

SmartFusion 2, IGLOO 2, RTG4 example:

```
export_smart_debug_data \ -file_name {sd1} \ -export_dir
{d:\sd_prj\test3T\designer\sd1\export} \ -force_rtg4_otp 1 \ -probes 1 \ -package_pins 0 \
-memory_blocks 1 \ -envm_data 0 \ -security_data 1 \ -chain 1 \ -programmer_settings 1
\ -ios_states 1 \ -generate_bitstream 0 \ -bitstream_security 0 \ -bitstream_fabric 0 \
-bitstream_envm 0
```


© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire examples:

```
export_smart_debug_data \ -file_name "top" \ -export_dir."/" \ -probes 1 \ -package_pins 0
\ -memory_blocks 1 \ -security_data 1 \ -chain 1 \ -programmer_settings 1 \ -ios_states 1 \
-generate_bitstream 1 \ -bitstream_security 0 \ -bitstream_fabric 1 \ -bitstream_snvm 1
```

```
export_smart_debug_data \ -file_name {smartdebug_export.ddc} \
-export_dir{H:\Designs\mss\designer\sd\export} \ -probes 1 \ -package_pins 0 \ -memory_blocks
1 \ -envm_data 1 \ -security_data 1 \ -display_security_in_smartdebug 0 \ -chain 1 \
-programmer_settings 1 \ -ios_states 1 \ -generate_bitstream 1 \ -bitstream_format {PPD}
\ -bitstream_security 0 \ -bitstream_fabric 1 \ -bitstream_envm 1 \ -sanitize_envm 1 \
-bitstream_snvm 1 \ -sanitize_snvm 1 \ -master_include_plaintext_passkey 0 \ -snvm_data 1
```

3.50. force_update_design_flow (Ask a Question)

Description

This Tcl command is for validating the invalidated tool states of Synthesize/Compile and Place and Route Design Flow steps to Pass state from OOD (Out of Date) state

```
force_update_design_flow
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'force_update_design_flow'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example validating the invalidated tool states of Synthesize/Compile and Place and Route Design Flow steps.

```
force_update_design_flow
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.51. generate_component (Ask a Question)

Description

This Tcl command generates a SmartDesign or a core component VHDL code. After generating component, the VHDL file is placed in the <project_folder>/component/work/<component_name> folder.

```
generate_component -component_name "component_name" \
                   [-recursive 0|1 ]
```

Arguments

Parameter	Type	Description
component_name	string	Specifies the name of the SmartDesign component or the core component is generated. It is mandatory.
recursive	integer	Specifies if a SmartDesign component must be generated recursively. It is optional. It is '0' by default and generates only the specified component. If set to '1', all the dependent components, which are in ungenerated state will be generated along with the SmartDesign component. It is recommended to generate all components individually.

Error Codes

Error Code	Description
None	Parameter 'component_name' cannot be empty.
None	recursive: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'generate_component [-component_name "component_name"] [-name "name"] [-recursive "TRUE | FALSE"]' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command generates SmartDesign "sd2" only.

```
generate_component -component_name {sd2}
```

The following command generates SmartDesign "TOP" and all its dependent components, which are in ungenerated state.

```
generate_component -component_name {TOP} -recursive 1
```

See Also

- delete_component
- import_component
- export_component_to_tcl

© 2025 Microchip Technology Inc. and its subsidiaries

3.52. generate_sdc_constraint_coverage (Ask a Question)

Description

This Tcl command generates the constraint coverage report. The constraint coverage report contains information about the coverage of the paths from associated SDC constraints in the design. Two constraint coverage reports can be generated: one for Place and Route, and other for Timing Verification.

To run this command, there is no need to run Place-and-Route first, but the design must be in the post-synthesis state. The generated constraint coverage reports (*.xml) are listed in the Reports tab and are physically located in the `<prj_folder>/designer/<module>/*constraints_coverage.xml` file.

Note: This command cannot be run until Compile has been run. Constraint Coverage Reports can be generated only after synthesis. A warning message appears, if the design is not in the post-synthesis state when this button is clicked.

```
generate_sdc_constraint_coverage -tool {PLACEROUTE | VERIFYTIMING}
```

Arguments

Parameter	Type	Description
tool	string	Specifies whether the constraint coverage report is based on the SDC constraint file associated with Place and Route or associated with Timing Verification. Use this dialog box to configure the 'Verify Timing' tool to generate a timing constraint coverage report and detailed static timing analysis and violation reports based on different combinations of process speed, operating voltage, and temperature. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'tool' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'generate_sdc_constraint_coverage -tool "tool name"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This command generates the SDC Constraint Coverage report for the SDC file associated with Place and Route.

```
generate_sdc_constraint_coverage -tool {PLACEROUTE}
```

This command generates the SDC Constraint Coverage report for the SDC file associated with Timing Verification.

```
generate_sdc_constraint_coverage -tool {VERIFYTIMING}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- Understanding Constraints Coverage Reports

3.53. get_libero_release (Ask a Question)

Description

This Tcl command returns the release number of the Libero SoC release. The value that is returned is the same as the release number that is displayed in the **Help** > **About Libero Window**.

```
get_libero_release
```

Arguments

Return Type	Description
string	Return the release number of the Libero SoC release.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example displays Libero release number with var1 variable.

```
get_libero_release
#save into a variable
set var1 [get_libero_release]
#display the variable
puts "Libero Release is $var1"
```

You will see output similar to the following.

```
Libero Version is 12.6.0
```

See Also

- get_libero_version

3.54. get_libero_version (Ask a Question)

Description

This Tcl command returns the version number of the Libero SoC version. The value that is returned is the same as the version number that is displayed in the **Help** > **About Libero window**.

```
get_libero_version
```

Arguments

Return Type	Description
string	Returns the version number of the Libero SoC version.

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

get_libero_version (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example displays Libero version number with var2 variable.

```
get_libero_version
#save into a variable
set var2 [get_libero_version]
#display variable
puts "Libero Version is $var2"
```

You will see output similar to the following.

```
Libero Version is 12.6.0
```

See Also

- get_libero_release

3.55. get_tool_options (Ask a Question)

Description

This Tcl command retrieves the configured options or parameters of a tool in the Libero Design Flow. You can use it to obtain the value of a single tool option or multiple tool options.

Important: You can repeat the -params argument for multiple parameters.

```
get_tool_options -name {tool_name} -params {parameter_names}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specifies the name of tool for which you want to know the configured tool options. It is mandatory. • SYNTHESIZE • COMPILE • PLACEROUTE • VERIFYTIMING • VERIFYPOWER • EXPORTNETLIST • CONFIGURE_PROG_OPTIONS • SPM • PROGRAMDEVICE • GENERATEPROGRAMMINGFILE • CONFIGURE_ACTION_PROCEDURES • PROGRAM_SPI_FLASH_IMAGE • SPM_OTP • FLASH_FREEZE • INIT_LOCK • IO_PROGRAMMING_STATE • UPDATE_ENVM • EXPORTSDF
params	string	Specifies the tool options/parameters for which you want to know the configured value. It is optional. It can either take single parameter or multiple parameters at a time. If omitted, all configured options/ parameters are returned.

Return Type	Description
options/parameters	Returns the configured options/parameters of a tool in the Libero Design Flow.

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter value does not exist.
None	Parameter 'param_name' does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'get_tool_options -name "tool name" [-params "[params]+"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

The following table lists all the tools for which this command is applicable.

© 2025 Microchip Technology Inc. and its subsidiaries

Tool Name (Tcl)	Tool Display Name	Supported Families
SYNTHESIZE	Synthesize	SmartFusion 2, IGLOO 2, RTG4, PolarFire
COMPILE	Compile	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PLACEROUTE	Place and Route	SmartFusion 2, IGLOO 2, RTG4, PolarFire
VERIFYTIMING	Verify Timing	SmartFusion 2, IGLOO 2, RTG4, PolarFire
VERIFYPOWER	Verify Power	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTNETLIST	File > Export > Netlist	SmartFusion 2, IGLOO 2, RTG4, PolarFire
CONFIGURE_PROG_OPTIONS	Configure Programming Options	SmartFusion 2, IGLOO 2, RTG4, PolarFire
SPM	Configure Security	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PROGRAMDEVICE	Run PROGRAM Action	SmartFusion 2, IGLOO 2, RTG4, PolarFire
GENERATEPROGRAMMINGFILE	Generate Bitstream	SmartFusion 2, IGLOO 2, RTG4, PolarFire
CONFIGURE_ACTION_PROCEDURES	Configure Actions and Procedures	SmartFusion 2, IGLOO 2, RTG4, PolarFire
FLASH_FREEZE	Configure Flash*Freeze	SmartFusion 2, IGLOO 2
INIT_LOCK	Configure Register Lock Bits	SmartFusion 2, IGLOO 2, RTG4, PolarFire
IO_PROGRAMMING_STATE	Configure I/O States During JTAG Programming	SmartFusion 2, IGLOO 2, RTG4, PolarFire
UPDATE_ENVM	Update eNVM Memory Content	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTSDF	Generate Back Annotated Files	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PROGRAM_SPI_FLASH_IMAGE	Run PROGRAM SPI_IMAGE Action	PolarFire
SPM_OTP	Configure Permanent Locks for Production	PolarFire

Example

1. The following command retrieves the value of the `RETIMING` parameter for the `SYNTHESIZE` tool.

```
puts [get_tool_options -name {SYNTHESIZE} -params {RETIMING}]
Output: true
```

2. The following example gets the values of multiple parameters.

```
set p [get_tool_options \
            -name {PLACEROUTE} \
            -params {REPAIR_MIN_DELAYEFFORT_LEVEL IOREG_COMBINING}]
puts "$p"
Output:
REPAIR_MIN_DELAY true EFFORT_LEVEL true IOREG_COMBINING false
```

3. When no parameters are given, in this case the following command returns all configured parameters of a VERIFYTIMING Libero tool.

```
puts [get_tool_options -name {VERIFYTIMING}]
Output:
CONSTRAINTS_COVERAGE true FORMAT XML MAX_TIMING_FAST_HV_LT false
MAX_TIMING_MULTI_CORNER true MAX_TIMING_SLOW_LV_HT false MAX_TIMING_SLOW_LV_LT
false MAX_TIMING_VIOLATIONS_FAST_HV_LT false MAX_TIMING_VIOLATIONS_MULTI_CORNER
true MAX_TIMING_VIOLATIONS_SLOW_LV_HT false MAX_TIMING_VIOLATIONS_SLOW_LV_LT
false MIN_TIMING_FAST_HV_LT false MIN_TIMING_MULTI_CORNER true
MIN_TIMING_SLOW_LV_HT false MIN_TIMING_SLOW_LV_LT false
MIN_TIMING_VIOLATIONS_FAST_HV_LT false MIN_TIMING_VIOLATIONS_MULTI_CORNER true
MIN_TIMING_VIOLATIONS_SLOW_LV_HT false MI
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.56. get_tool_state (Ask a Question)

Description

This Tcl command is used to get the state of a tool in the Libero Design Flow. It can be run on all tools which have a tool state in the UI such as green check mark/error/out of date/has not run yet/tool run with warnings, and so on.

```
get_tool_state -name {tool_name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of tool for which you wish to get the tool state. It is mandatory. • SYNTHESIZE • COMPILE • PLACEROUTE • VERIFYTIMING • VERIFYPOWER • PROGRAMDEVICE • GENERATE_MEMORY_MAP • GENERATEPROGRAMMINGFILE • EXPORTPROGRAMMINGFILE • EXPORTPROGRAMMINGJOB • EXPORTJOBDATA • EXPORTNETLIST • EXPORTSMARTDEBUGDATA • PUBLISHBLOCK • EXPORTSDF • GENERATEPROGRAMMINGDATA

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'get_tool_state -name "tool name"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

The following table shows the possible tool states/tool tips.

© 2025 Microchip Technology Inc. and its subsidiaries

Important: The tool states described may not be applicable to all tools.

Tool State/Tool tip	Description
Tool successfully run	When the execution is successful.
Device/Project settings have changed	When the Device/Project settings that affect the tool are modified.
Tool configuration has changed	When a tool's configuration options are changed.
Tool failed	When a tool execution fails.
Tool has not run yet	When a tool has not run yet.
Tool inputs are out of date	When a parent tool state is cleaned, or when a design source is modified, or something in another tool is modified that the current tool is dependent on.
Tool executed	When a tool has run successfully.
Tool has timing violations/Timing constraints have not been met	When the Verify Timing tool has run successfully but the design has timing violations.
Tool successfully run/Timing constraints have been met	When the Verify Timing tool has run successfully and there are no timing violations for the design.

The following table shows all the tools for which this command is applicable.

Tool Name (Tcl)	Tool Display Name	Supported Families
SYNTHESIZE	Synthesize	SmartFusion® 2, IGLOO® 2, RTG4™, PolarFire®
COMPILE	Compile	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PLACEROUTE	Place and Route	SmartFusion 2, IGLOO 2, RTG4, PolarFire
VERIFYTIMING	Verify Timing	SmartFusion 2, IGLOO 2, RTG4, PolarFire
VERIFYPOWER	Verify Power	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PROGRAMDEVICE	Run PROGRAM Action	SmartFusion 2, IGLOO 2, RTG4, PolarFire
GENERATE_MEMORY_MAP	Generate Memory Map	SmartFusion 2, IGLOO 2, RTG4, PolarFire
GENERATEPROGRAMMINGFILE	Generate Bitstream	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTPROGRAMMINGFILE	Export Bitstream	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTPROGRAMMINGJOB	Export FlashPro Express Job	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTJOBDATA	Export Job Manager Data	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTNETLIST	File > Export > Netlist...	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTSMARTDEBUGDATA	Export SmartDebug Data	SmartFusion 2, IGLOO 2, RTG4, PolarFire
PUBLISHBLOCK	Publish Block	SmartFusion 2, IGLOO 2, RTG4, PolarFire
EXPORTSDF	Generate Back Annotated Files	SmartFusion 2, IGLOO 2, RTG4, PolarFire
GENERATEPROGRAMMINGDATA	Generate FPGA Array Data	SmartFusion 2, IGLOO 2, RTG4, PolarFire

Example

The following example gets the state of synthesize tool in the Libero Design flow.

```
set state [get_tool_state -name {SYNTHESIZE}]
puts "$state"
Output: Tool successfully run
```

3.57. import_component (Ask a Question)

Description

This Tcl command imports a component `*.cxf` file into the Libero project. After import, the `*.cxf` file is placed in the `<project_folder>/component/work/<component_name>` folder.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Only the `*.cxf` file format is supported for component import.

```
import_component -file <path_to_component.cxf>
```

Arguments

Parameter	Type	Description
file	string	The `-file` argument specifies the location of the component `*.cxf` file to import. Both absolute and relative paths are supported.

Error Codes

Error Code	Description
None	Could not find the specified core file.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example imports a component `my_mult.cxf` file into the Libero project.

```
import_component -file {D:/test/my_design/my_mult.cxf}
```

See Also

- import_component_data
- delete_component
- generate_component

3.58. import_component_data (Ask a Question)

Description

A Libero SoC general purpose Tcl command to import component data into an existing Libero project. Component refers to MDDR, FDDR, and SerDes peripherals in SmartFusion 2 devices. Component data refers the initialization or configuration register values (`*init_reg` or `*init.mem` files) of those peripherals. Use this command if and when:

1. The synthesized netlist or HDL files in the existing Libero SoC project contain no component (MDDR, FDDR, or SerDes) information.
2. You want to add components (MDDR, FDDR, or SerDes) into the existing design.

Note: The eNVM config file can have any name. Either `*_init.reg` (register configuration file) or `*.mem` files (memory files) can be used. Both cannot be used together in the same `import_component_data` command.

```
import_component_data \
    -module root # name of the top_level (root) \
    -fddr file_path_and_name # has to be FDDR_init.reg or .mem \
    -mddr file_path_and_name # has to be MDDR_init.reg or .mem \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
    -serdes0 file_path_and_name # has to be SERDESIF_0_init.reg or .mem \
    -serdes1 file_path_and_name # has to be SERDESIF_1_init.reg or .mem \
    -serdes2 file_path_and_name # has to be SERDESIF_2_init.reg or .mem \
    -serdes3 file_path_and_name # has to be SERDESIF_3_init.reg or .mem \
    -envm_cfg file_path_and_name # SmartFusion 2, IGLOO 2 only \
    -uprom_cfg file_path_and_name # RTG4 only
```

Arguments

Parameter	Type	Description
module	string	Specifies the top level design name of the Libero project into which component data is being imported.
fddr	string	Specifies the file path and file name to the `FDDR_init.reg` or `.mem`.
mddr	string	Specifies the file path and file name to the `MDDR_init.reg` or `.mem`.
serdes0	string	Specifies the file path and file name to the `SERDESIF_0_init.reg` or `.mem`.
serdes1	string	Specifies the file path and file name to the `SERDESIF_1_init.reg` or `.mem`.
serdes2	string	Specifies the file path and file name to the `SERDESIF_2_init.reg` or `.mem`.
serdes3	string	Specifies the file path and file name to the `SERDESIF_3_init.reg` or `.mem`.
envm_cfg	string	Specifies the file path and file name. **Note:** This argument is only supported for the SmartFusion® 2 and IGLOO® 2 families of devices.
uprom_cfg	string	Specifies the file path and file name. **Note:** This argument is only supported for the RTG4™ family of devices.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'import_component_data -module "root module name" [-mddr "mddr register file name"] [-fddr "fddr register file name"] [-serdes0 "serdes0 register file name"] [-serdes1 "serdes1 register file name"] [-serdes2 "serdes2 register file name"] [-serdes3 "serdes3 register file name"] [-envm_cfg "user envm cfg file name"] [-uprom_cfg "user uprom cfg file name"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The component name for IGLOO 2 devices might have a different file extension (`*.mem` or `*.reg`), depending on the Libero SoC release version used to generate the components.

© 2025 Microchip Technology Inc. and its subsidiaries

The following is an example of importing design components created with a Libero SoC pre-v11.4 release into an IGLOO 2 project.

```
import_component_data -module <root> \
                      -fddr <file_path>/FDDR_init.mem \
                      -mddr <file_path>/MDDR_init.mem \
                      -serdes0 <file_path>/SERDESIF_0_init.mem \
                      -serdes1 <file_path>/SERDESIF_1_init.mem \
                      -serdes2 <file_path>/SERDESIF_2_init.mem \
                      -serdes3 <file_path>/SERDESIF_3_init.mem \
                      -envm_cfg <user_cfg_file_path>
```

The following is an example of importing design components created with Libero SoC v11.4 or subsequent releases into an IGLOO 2 project.

Note: The `*.reg` file extension is required.

```
import_component_data -module <root> \
                      -fddr <file_path>/FDDR_init.reg \
                      -mddr <file_path>/MDDR_init.reg \
                      -serdes0 <file_path>/SERDESIF_0_init.reg \
                      -serdes1 <file_path>/SERDESIF_1_init.reg \
                      -serdes2 <file_path>/SERDESIF_2_init.reg \
                      -serdes3 <file_path>/SERDESIF_3_init.reg \
                      -envm_cfg <user_cfg_file_path>
```

The following is an example of importing design components created with a Libero SoC pre-v11.4 release into a SmartFusion 2 project.

```
import_component_data -module <root> \
                      -fddr <file_path>/FDDR_init.reg \
                      -mddr <file_path>/MDDR_init.reg \
                      -serdes0 <file_path>/SERDESIF_0_init.reg \
                      -serdes1 <file_path>/SERDESIF_1_init.reg \
                      -serdes2 <file_path>/SERDESIF_2_init.reg \
                      -serdes3 <file_path>/SERDESIF_3_init.reg \
                      -envm_cfg <user_cfg_file_path>
```

The following is an example of importing design components created with Libero SoC v11.4 or a subsequent release into a SmartFusion 2 project.

```
import_component_data -module <root> -envm_cfg <user_cfg_file_path>
```

3.59. import_files (Ask a Question)

Description

This Tcl command enables you to import design source files and constraint files.

For importing constraint files, import_files has retired the `-pdc` parameter for SmartFusion 2, IGLOO 2, PolarFire and RTG4. It has been replaced with two new parameters to match the new design flow. Physical Design Constraints (PDC) Tcl must now be divided between I/O attribute and pin information from all floorplanning and timing constraints. These commands must now reside in and be imported as separate files. The new parameters specify the type of *.pdc file being imported. The path to the file can be absolute or relative but must be enclosed in curly braces { }.

Use the `-convert_EDN_to_HDL` parameter to convert the EDIF file to HDL and then import the converted HDL file.

Note: The EDIF File is not imported.

```
import_files \
-schematic {file} \
-symbol {file} \
-smartgen_core {file} \
-ccp {file} \
-stimulus {file} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
-hdl_source {file} \
-io_pdc {file} \
-fp_pdc {file} \
-edif {file} \
-sdc {file} \
-crt {file} \
-dcf {file} \
-vcd {file} \
-saif {file} \
-simulation {file} \
-profiles {file} \
-cxf {file} \
-templates {file} \
-ccz {file} \
-modelsim_ini {file} \
-library {file} \
-convert_EDN_to_HDL {true | false}
```

Arguments

Parameter	Type	Description
schematic	string	Specifies the schematics you wish to import into your IDE project. Type parameter must be repeated for each file.
symbol	string	Specifies the symbols you wish to import into your IDE project. Type parameter must be repeated for each file.
smartgen_core	string	Specifies the SmartGen Cores you wish to import into your project. Type parameter must be repeated for each file.
ccp	string	Specifies the Arm® or Cortex®-M1 cores you wish to import into your project. Type parameter must be repeated for each file.
stimulus	string	Specifies HDL stimulus files you wish to import into your project. Type parameter must be repeated for each file.
hdl_source_folder	string	Name of the HDL folder you want to import into your project.
hdl_source	string	Specifies the HDL source files you wish to import into your project. Type parameter must be repeated for each file.
io_pdc	string	Specifies the PDC file that contains the I/O attribute and pin information.
fp_pdc	string	Specifies the PDC file that contains the timing and placement information.
edif	string	Specifies the EDIF files you wish to import into your project. Type parameter must be repeated for each file. This is a mandatory option if you want to convert EDIF to HDL with the `-convert_EDN_to_HDL` option. This option is not supported in PolarFire.
sdc	string	Specifies the SDC constraint files you wish to import into your project. Type parameter must be repeated for each file.
crt	string	Specifies the CRT constraint files you wish to import into your project. Type parameter must be repeated for each file. This option is not supported for PolarFire®, SmartFusion® 2, IGLOO® 2, and RTG4™ families.
dcf	string	Specifies the DCF constraint files you wish to import into your project. Type parameter must be repeated for each file. **Note:** Not supported for PolarFire, SmartFusion 2, IGLOO 2, and RTG4 families.
vcd	string	Specifies the VCD constraint files you wish to import into your project. Type parameter must be repeated for each file.
saif	string	Specifies the SAIF constraint files you wish to import into your project. Type parameter must be repeated for each file.
simulation	string	Specifies the simulation files you wish to import into your Libero SoC project. Type parameter must be repeated for each file.
profiles	string	Specifies the profile files you wish to import into your Libero SoC project. Type parameter must be repeated for each file.

© 2025 Microchip Technology Inc. and its subsidiaries

import_files (continued)		
Parameter	**Type**	**Description**
cxf	string	Specifies the CXF (Component) file you wish to import into your Libero SoC project. Type parameter must be repeated for each file.
templates	string	Specifies the template file you wish to import into your project.
ccz	string	Specifies the IP Control core file you wish to import into your project.
modelsim_ini	string	Specifies the ModelSIM INI file that you wish to import into your project.
library	string	Specifies the library file that you wish to import into your project. If a library file is not available it will be created and added to the library.
convert_EDN_to_HDL	boolean	The `-edif` option is mandatory. If the `-edif` option is not specified or the `-convert_EDN_to_HDL` is used with another option, EDIF to HDL conversion will fail.

Error Codes

Error Code	Description
None	Unable to find the file 'a.v'.
None	Cannot import the 'ProASIC Constraint Files'; your selected family does not support GCF constraints.
None	Cannot import the 'Criticality Files'; your selected family does not support CRT constraints.
None	Cannot import the 'Timing Constraint Files'; your selected family does not support DCF constraints.
None	Cannot import the 'Pin Files'; your selected family does not support PIN constraints.
None	Error: Parameter is not defined. Valid command formatting is 'import_files [-convert_EDN_to_HDL "TRUE | FALSE"] \ [-hdl_source_folder "Source folder"]* \ [-library "library"] \ [-cxz "file"]* \ [-cxf "file"]* \ [-ccp "file"]* \ [-crt "file"]* \ [-hdl_source "file"]* \ [-stimulus "file"]* \ [-templates "file"]* \ [-modelsim_ini "file"]* \ [-fp_pdc "file"]* \ [-io_pdc "file"]* \ [-sdc "file"]* \ [-ndc "file"]* \ [-net_fdc "file"]* \ [-icf "file"]* \ [-ccz "file"]* \ [-cpz "file"]* \ [-pin "file"]* \ [-gcf "file"]* \ [-saif "file"]* \ [-schematic "file"]* \ [-simulation "file"]* \ [-smartgen_core "file"]* \ [-symbol "file"]* \ [-verilog_netlist "file"]* \ [-dcf "file"]* \ [-profiles "file"]* \ [-vcd "file"]*

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The command below imports the HDL source files file1.vhd and file2.vhd.

```
import_files -hdl_source file1.vhd -hdl_source file2.vhd
```

3.60. import_mss_component (Ask a Question)

Description

This Tcl command imports the MSS component to your project.

```
import_mss_component [-file {path to the *.cxz file}]
```

Arguments

Parameter	Type	Description
file	string	Path of the *.cxz files to be imported.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'import_mss_component -file "file"'.

Supported Families

PolarFire® SoC

Example

Importing a mss component file located at ‘./src/file_name.cxz’ to your project.

```
import_mss_component -file {./src/file_name.cxz}
```

3.61. is_synthesis_enabled (Ask a Question)

Description

Tcl query to determine if a project can be synthesized. The source is an `.edn` file.

```
is_synthesis_enabled
```

Arguments

Return Type	Description
1	Synthesis is enabled.
0	Synthesis is disabled.

Example

```
set p [is_synthesis_enabled]
puts “$p”
```

3.62. new_project (Ask a Question)

Description

This Tcl command creates a new project in Libero SoC. If you do not specify a location, Libero SoC saves the new project in your current working directory.

```
new_project -name project_name \
-location project_location \
-family family_name \
-project_description "brief text description of project" \
-die device_die \
-package package_name \
-hdl HDL_type \
-speed speed_grade \
-die_voltage value \
-part_range value \
-block_mode {1 | 0} \
-ondemand_build_dh {1 | 0} \
-adv_options value \
-use_relative_path {1 | 0} \
-linked_files_root_dir_env root_dir_env \
-standalone_peripheral_initialization {1 | 0} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
-instantiate_in_smartdesign {1 | 0} \
-use_enhanced_constraint_flow {1 | 0}
```

Arguments

Parameter	Type	Description
name	string	The name of the project. This is used as the base name for most of the files generated from Libero SoC.
location	string	The location of the project. Must not be an existing directory.
project_description	string	A brief text description of the design in your project.
family	string	The Microchip SoC device family for your targeted design.
die	string	Sets device die for your targeted design.
package	string	Sets device package for your targeted design.
hdl	string	Sets the HDL type for your new project. Valid values are: • VHDL—sets your new projects HDL type to VHDL. • VERILOG—sets your new projects to Verilog.
speed	string	Sets the speed grade for your project. Possible values depend on your device, die, and package. See your device datasheet for details.
die_voltage	floating point	Sets the die voltage for your project. Possible values depend on your device. See your device datasheet for details.
part_range	string	Sets your default temperature range for your project • PolarFire: EXT (Extended), IND (Industrial), MIL (Military) • SmartFusion 2: COM (Commercial), IND, TGrade2 (Automotive), MIL • IGLOO 2: COM, IND, TGrade1, TGrade2, MIL • RTG4: MIL
ondemand_build_dh	boolean	Enter "1" to enable or "0" (default) to disable On Demand Build Design Hierarchy.
block_mode	boolean	Enter "1" to enable or "0" (default) to disable design block creation.
instantiate_in_smartdesign	boolean	Enter "1" to enable or "0" (default) to disable Instantiate SystemBuilder/MSS components in a SmartDesign. When set to "1", a System Builder or MSS component is auto-instantiated in a SmartDesign component upon creation. The default is 1.
use_relative_path	boolean	Enter "1" to use relative path or "0" (default) to use absolute path setting for the linked files in the project.
linked_files_root_dir_env	boolean	The System Environment variable that has valid root directory path. All the linked files in the project will be referenced relative to the path set in the Environment variable. The value in this argument is used only if the relative path is set in -use_relative_path argument.
adv_options	string	Sets your advanced options, such as temperature and voltage settings. For more information, see the following table.

The following are advanced options for temperature and voltage settings.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
IO_DEFT_STD:LVTTL	Sets your I/O default value to LVTTL. This value defines the default I/O technology to be used for any I/Os that you need not explicitly set a technology for in the I/O Editor. It could be any of: • LVTTL • LVCMOS 3.3V • LVCMOS 2.5V • LVCMOS 1.8V • LVCMOS 1.5V • LVCMOS 1.2V
DSW_VCCA_VOLTAGE_RAMP_RATE	(SmartFusion 2 and IGLOO 2 only) This value defines the Maximum VDD and VPP power supply ramp rate. Power-up management circuitry is designed into every SmartFusion 2 and IGLOO 2 SoC FPGA. These circuits ensure easy transition from the powered-off state to powered-up state of the device. The SmartFusion 2 or IGLOO 2 system controller is responsible for systematic power-on reset whenever the device is powered on or reset. All the I/Os are held in a high-impedance state by the system controller until all power supplies are at their required levels and the system controller has completed the reset sequence. The power-on reset circuitry in SmartFusion 2 and IGLOO 2 devices requires the VDD and VPP supplies to ramp monotonically from 0 V to the minimum recommended operating voltage within a predefined time. There is no sequencing requirement on VDD and VPP. Four ramp rate options are available during design generation: • 50 µs • 1 ms • 10 ms • 100 ms Each selection represents the maximum ramp rate to apply to VDD and VPP.
PLL_SUPPLY	(SmartFusion 2, IGLOO 2 only) This value sets the voltage for the power supply you plan to connect to all the PLLs in your design, such as MDDR, FDDR, SERDES, and FCCC. Two values are available: • 2.5 • 3.3
RESTRICTPROBEPINS	This value reserves your pins for probing, if you intend to debug using SmartDebug. Two values are available: • 1 (Probe pins are reserved) • 0 (No probe pins are reserved)
RESTRICTSPIPINS:1	(RTG4 only) Sets to 1 to reserve pins for SPI functionality in Programming. This reserved SPI pin option is displayed in the Compile Report when the compile process completes.
SYSTEM_CONTROLLER_SUSPEND_MODE	(SmartFusion 2, IGLOO 2 only) Enables SmartFusion 2 and IGLOO 2 designers to suspend operation of the System Controller. Enabling this bit instructs the System Controller to place itself in a reset state once the device is powered up. This effectively suspends all system services from being performed. For a list of system services, refer to the SmartFusion 2 or IGLOO 2 System Controller User Guide for your device on the Microchip website. Two values are available: • 1 (System Controller Suspend Mode is enabled) • 0 (System Controller Suspend Mode is disabled

The following are options for Analysis Operating Conditions so that Timing and Power analysis can be performed at different operating conditions.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
TEMPR	Sets your default temperature range for operating condition analysis. • COM (Commercial) • MIL (Military) • IND (Industrial)
VCCI_1.2_VOLTR	Sets the Default I/O Voltage Range for 1.2V, which could be • COM (Commercial) • MIL (Military) • IND (Industrial) • Custom These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis.
VCCI_1.5_VOLTR	Sets the Default I/O Voltage Range for 1.5V, which could be • COM (Commercial) • MIL (Military) • IND (Industrial) • Custom These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis.
VCCI_1.8_VOLTR	Sets the Default I/O Voltage Range for 1.8V, which could be • COM (Commercial) • MIL (Military) • IND (Industrial) • Custom These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis.
VCCI_2.5_VOLTR	Sets the Default I/O Voltage Range for 2.5V, which could be • COM (Commercial) • MIL (Military) • IND (Industrial) • Custom These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis.
VCCI_3.3_VOLTR	Sets the Default I/O Voltage Range for 3.3V, which could be • COM (Commercial) • MIL (Military) • IND (Industrial) • Custom These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis.
VOLTR	Sets the core voltage range for operating condition analysis. These settings are propagated to Verify Timing, Verify Power and Backannotated Netlist to perform Timing/Power Analysis. Can be one of the following: • COM (Commercial) • MIL (Military) • IND (Industrial)

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	auto_update_modelsim_ini: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	auto_update_viewdraw_ini: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	block_mode: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	auto_generate_synth_hdl: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	You do not have write access to /prj_path/exprj/viewdraw/vf/project.lst. ViewDraw A.E. cannot open.
None	enable_set_mitigation: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	auto_file_detection: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	hdl: Invalid argument value: '' (expecting VHDL or VERILOG).
None	Parameter 'param_name' is not defined. Valid command formatting is 'project_settings [-hdl "VHDL | VERILOG"] [-verilog_mode "SYSTEM_VERILOG | VERILOG_2K"] [-vhdl_mode "VHDL_2008 | VHDL_93"] [-system_verilog_mfcu "TRUE | FALSE"] [-auto_update_modelsim_ini "TRUE | FALSE"] [-auto_update_viewdraw_ini "TRUE | FALSE"] [-enable_viewdraw "TRUE | FALSE"] [-standalone_peripheral_initialization "TRUE | FALSE"] [-instantiate_in_smartdesign "TRUE | FALSE"] [-ondemand_build_dh "TRUE | FALSE"] [-auto_generate_synth_hdl "TRUE | FALSE"] [-auto_generate_physynth_hdl "TRUE | FALSE"] [-auto_run_drc "TRUE | FALSE"] [-auto_generate_viewdraw_hdl "TRUE | FALSE"] [-auto_file_detection "TRUE | FALSE"] [-sim_flow_mode "TRUE | FALSE"] [-vm_netlist_flow "TRUE | FALSE"] [-enable_set_mitigation "TRUE | FALSE"] [-display_fanout_limit "display_fanout_limit"] [-abort_flow_on_sdc_errors "TRUE | FALSE"] [-abort_flow_on_pdc_errors "TRUE | FALSE"] [-block_mode "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

1. Creates a new project in the `./designs/mydesign` directory, with the HDL type Verilog for the SmartFusion 2 family.

```
new_project -location {./designs/mydesign} \
-name {mydesign} \
-use_enhanced_constraint_flow 1 \
-use_relative_path 1 -linked_files_root_dir_env {MSCC_ROOT_1} \
-standalone_peripheral_initialization 1 -hdl {VERILOG} -family {SmartFusion 2} \
-die {M2S150TS} -package {FCS536} -speed {-1} -die_voltage {1.2} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
-part_range {COM} -adv_options {DSW_VCCA_VOLTAGE_RAMP_RATE:100_MS} \
-adv_options {IO_DEFT_STD:LVCMOS 2.5V} \
-adv_options {PLL_SUPPLY:PLL_SUPPLY_25} \
-adv_options {RESTRICTPROBEPINS:1} \
-adv_options {SYSTEM_CONTROLLER_SUSPEND_MODE:0} \
-adv_options {TEMPR:IND} -adv_options {VCCI_1.2_VOLTR:IND} \
-adv_options {VCCI_1.5_VOLTR:IND} -adv_options {VCCI_1.8_VOLTR:IND} \
-adv_options {VCCI_2.5_VOLTR:IND} -adv_options {VCCI_3.3_VOLTR:IND} \
-adv_options {VOLTR:IND}
```

2. Creates a new project in the `D:/2Work/my_pf_proj` directory, with the HDL type Verilog for PolarFire. Sets up a new design and runs Libero tools.

```
new_project -location {D:/2Work/my_pf_proj} -name {my_pf_proj} \
-project_description {} - block_mode 0 -standalone_peripheral_initialization 0 \
-use_enhanced_constraint_flow 1 -use_relative_path 1 \
-linked_files_root_dir_env {MSCC_ROOT_1} -hdl {VERILOG} -family {PolarFire} \
-die {MPF300TS_ES} -package {FCG1152} -speed {-1} - die_voltage {1.0} \
-part_range {EXT} -adv_options {IO_DEFT_STD:LVCMOS 1.8V} -adv_options
{RESTRICTPROBEPINS:1} \
-adv_options {RESTRICTSPIPINS:0} -adv_options {SYSTEM_CONTROLLER_SUSPEND_MODE:1} \
-adv_options {TEMPR:EXT} -adv_options {VCCI_1.2_VOLTR:EXT} -adv_options
{VCCI_1.5_VOLTR:EXT} \
-adv_options {VCCI_1.8_VOLTR:EXT} -adv_options {VCCI_2.5_VOLTR:EXT} \
-adv_options {VCCI_3.3_VOLTR:EXT} -adv_options {VOLTR:EXT}
#Import HDL source file
import_files -convert_EDN_to_HDL 0 -hdl_source {C:/test/prep1.v}

#Import HDL stimulus file
import_files -convert_EDN_to_HDL 0 -stimulus {C:/test/prep1tb.v}
#set the top level design name
set_root -module {prep1::work}

#Associate SDC constraint file to Place and Route tool
organize_tool_files -tool {PLACEROUTE} -file {D:/2Work/my_pf_proj/constraint/user.sdc} \
-module {prep1::work} -input_type {constraint}

#Associate SDC constraint file to Verify Timing tool
organize_tool_files -tool {VERIFYTIMING} -file {D:/2Work/my_pf_proj/constraint/user.sdc} \
-module {prep1::work} -input_type {constraint}
#Run synthesize
run_tool -name {SYNTHESIZE}
#Configure Place and Route tool
configure_tool -name {PLACEROUTE} -params {DELAY_ANALYSIS:MAX} -params
{EFFORT_LEVEL:false} \
-params {INCRPLACEANDROUTE:false} -params {MULTI_PASS_CRITERIA:VIOLATIONS} \
-params {MULTI_PASS_LAYOUT:false} -params {NUM_MULTI_PASSES:5} -params {PDPR:false} \
-params {RANDOM_SEED:0} -params {REPAIR_MIN_DELAY:false} -params
{SLACK_CRITERIA:WORST_SLACK} \
-params {SPECIFIC_CLOCK:} -params {START_SEED_INDEX:1} -params {STOP_ON_FIRST_PASS:false} \
-params {TDPR:true}
```

3.63. one_way_passcode (Ask a Question)

Description

This Tcl command enables/disables One Way Passcode (OWP).

Important: This Tcl command is applicable for Libero SoC Design Suite v2022.3 and later.

```
one_way_passcode -enable true/false (or 0/1)
```

Arguments

Parameter	Type	Description
enable	boolean	Enter 1 to enable or 0 (default) to disable **One-Way Passcode** (OWP). Enter true to enable or false (default) to disable **One-Way Passcode** (OWP).

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	NA

Supported Families

PolarFire® SoC

Example

The following example enables One Way Passcode (OWP).

```
one_way_passcode -enable 1
one_way_passcode -enable true
```

3.64. open_project (Ask a Question)

Description

This Tcl command opens an existing Libero SoC project. You can create backup of your original project before opening.

```
open_project -file "project file" \
             [-do_backup_on_convert "TRUE | FALSE"] \
             [-backup_file "backup archive file name"]
```

Arguments

Parameter	Type	Description
file	string	Must include the complete path to the PRJ file. If you do not provide the full path, Libero SoC infers that you want to open the project from your current working directory.
do_backup_on_convert	boolean	Sets the option to backup your files if you open a project created in a previous version of Libero SoC. • TRUE - creates a backup of your original project before opening. • FALSE - opens your project without creating a backup.
backup_file	string	Sets the name of your backup file (if you choose to do_backup_on_convert).

Error Codes

Error Code	Description
None	Unable to find project file: /exprj/exprj.prjx.
None	Required parameter 'file' is missing.
None	Parameter 'backup_file' has illegal value.
None	Parameter 'backup_file' : the file '/prj_path/archive' has got an invalid extension. Valid extension is 'zip'.
None	do_backup_on_convert: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'open_project -file "project file" [-do_backup_on_convert "TRUE | FALSE"] [-backup_file "backup archive file name"] ' .

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command opens project.prjx project from test directory without creating a backup file.

```
open_project -file {c:/netlists/test/project.prjx}
```

See Also

- close_project
- new_project
- save_project

3.65. open_smartdesign (Ask a Question)

Description

This Tcl command opens a SmartDesign. You must either open or create a SmartDesign before using any of the SmartDesign specific commands "sd_*".

Important: This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
open_smartdesign [-sd_name "smartdesign_component_name"]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of an existing SmartDesign component. It is mandatory. You can specify the name of SmartDesign as follow: {<design_name>} or {<design_name>::<module_name>}.

Error Codes

Error Code	Description
None	Parameter 'sd_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'open_smartdesign [-sd_name "sd_name"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

Open "top" SmartDesign from your project.

```
open_smartdesign -sd_name {top}
```

See Also

- create_smartdesign

3.66. organize_constraints (Ask a Question)

Description

This Tcl command organizes the constraint files in your project. The Organize Constraint Files dialog box enables you to set the constraint file and order in the Libero SoC.

```
organize_constraints [-file "filename" ] \
                     [-mode "new | add | remove" ] \
                     [-designer_view "designer view" ] \
                     [-module "module" ] \
                     -tool "value"
```

Arguments

Parameter	Type	Description
file	string	Specifies the name of the file to which you want to associate your stimulus files.
mode	string	Specifies whether you are creating a new stimulus association, adding or removing; possible values are: • new - creates a new stimulus file association. • add - adds a stimulus file to an existing association. • remove - removes a stimulus file association.
designer_view	string	Sets the name of the Designer View in which you wish to add the constraint file, where name is the name of the view (such as impl1).
module	string	Sets the module, where value is the name of the module.
tool	string	Identifies the intended use for the file, possible values are: • synthesis - file to be used for synthesis. • designer - file to be used in Designer. • phsynth - file to be used in physical synthesis.

Error Codes

Error Code	Description
None	Required parameter 'tool' is missing.
None	Parameter 'module' has illegal value.
None	Parameter 'designer_view' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'organize_constraints [-file "file"]* [-mode "new | add | remove"] [-designer_view "designer view"] [-module "module"] -tool "synthesis | designer | physynth | timing | compilenetlist | pnr"'.

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

organize_constraints (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The example adds the constraint file delta.vhd in the Designer View impl2 for the Designer tool.

```
organize_constraints -file "delta.vhd" -mode "new" \
                     -designer_view "impl2" \
                     -module "constraint_tool" \
                     -tool "designer"
```

See Also

- organize_tool_files

3.67. organize_sources (Ask a Question)

Description

This Tcl command enables you to set the source file order in the Libero SoC.

To specify the file order:

1. In the **Design Flow** window, right-click **Synthesize or Simulation** tool and choose **Organize Input Files** > **Organize Source Files**. The **Organize Source Files** dialog box appears.
2. Click the Use list of files organized by User radio button to Add/Remove source files for the selected tool.
3. Use the Up and Down arrows to change the order of the Associated Source files.

```
organize_sources [-file filename ] [-mode "new | add | remove" ] \
                 -module value -tool "synthesis | simulation" \
                 [-use_default value ]
```

Arguments

Parameter	Type	Description
file	string	Specifies the name of the file to which you want to associate your source files. It is optional. Default is empty.
mode	string	Specifies whether you are creating a new source files association, adding, or removing for the selected tool; possible values are: • new - creates a new source file association. • add - adds a source file to an existing association. • remove - removes a source file association.
module	string	Sets the module, where value is the name of the module. You can specify as {<module::work>} or {<module>} <module>. This is mandatory.
tool	string	Identifies the intended use for the file, possible values are: • synthesis - file to be used for synthesis. • simulation - file to be used in simulation.

© 2025 Microchip Technology Inc. and its subsidiaries

organize_sources (continued)		
Parameter	**Type**	**Description**
use_default	string	Uses the default values for synthesis or simulation; possible values are: • TRUE, true, 1 - uses default values for synthesis or simulation. This is the default value. • FALSE, false, 0 - uses user-defined values for synthesis or simulation.

Error Codes

Error Code	Description
None	'file' is not in the project.
None	Required parameter 'module' is missing.
None	mode: Invalid argument value: 'mode_value' (expecting new, add or remove).
None	tool: Invalid argument value: 'designer' (expecting synthesis or simulation).
None	Parameter 'param_name' is not defined. Valid command formatting is 'organize_sources [-file "file"]* [-mode "new | add | remove"] -module "module" -tool "synthesis | simulation" [-use_default "TRUE | FALSE"] ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The example organizes a new stimulus file 'stim.vhd' using default settings.

```
organize_sources -file "stim.vhd" -mode "new" -module "stimulus" \
                 -tool "synthesis" -use_default "TRUE"
```

See Also

- organize_tool_files

3.68. organize_tool_files (Ask a Question)

Description

This Tcl command is used to specify specific files to be passed to and used by a Libero tool. If you do not want to pass file by Libero tools then specify `-file` option value as empty: `-file {}`.

```
organize_tool_files \
         -tool {tool_name} \
         [-file {absolute path to specific file} ] \
         [-module {<design_name>::work} ] \
         -input_type {value}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
tool	string	Specifies the name of the tool files you want to organize. Valid values are:`SYNTHESIZE | COMPILE | PLACEROUTE | SIM_PRESYNTH | SIM_POSTSYNTH | SIM_POSTLAYOUT | VERIFYTIMING`.
file	string	Specifies the absolute path to the specific file; there may be multiple -file arguments (see example below). It is mandatory. You can specify file as: `-file {filename}` or `-file "filename"`. You can repeat this argument for specifing multiple files.
module	string	Module definition, format is `<design_name>::work` or `<design_name>`. It is optional. Default is `<root_design_name>::work`.
input_type	string	Specifies type of input file. Possible values are:`constraint | source | simulation | stimulus | unknown`. It is mandatory.

Error Codes

Error Code	Description
None	'user.sdc' is not in the project.
None	Required parameter 'file' is missing.
None	Required parameter 'tool' is missing.
None	Required parameter 'input_type' is missing.
None	input_type: Invalid argument value: 'value' (expecting source, constraint, simulation, stimulus or unknown).
None	Parameter 'param_name' is not defined. Valid command formatting is 'organize_tool_files -tool "tool name" [-file "file"] + [-module "module"] -input_type "source | constraint | simulation | stimulus | unknown" '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command organizes the test_derived.sdc and user.sdc files of SDC (Synopsys Design Constraint) file type for the tool VERIFYTIMING for the sd1::work design.

```
organize_tool_files \
-tool {VERIFYTIMING} \
-file {D:/Designs/my_proj/constraints/test_derived.sdc} \
-file {D:/Designs/my_proj/constraints/user.sdc} \
-module {sd1::work} \
-input_type {constraint}
```

3.69. project_general_settings (Ask a Question)

Description

This Tcl command modifies the settings for linked files storage in your Libero SoC project. Once a user changes the settings from Absolute mode to relative mode, the user will not be able to switch

© 2025 Microchip Technology Inc. and its subsidiaries

back the setting to absolute mode. The value in this argument is used only if the relative path is set in -use_relative_path argument.

```
project_general_settings \
             [-use_relative_path { 1 | 0 }] \
             [-linked_files_root_dir_env {linked files root directory}]
```

Arguments

Parameter	Type	Description
use_relative_path	boolean	Enter "1" to use relative path or "0" (default) to use absolute path setting for the linked files in the project.
linked_files_root_dir_env	string	The System Environment variable that has valid root directory path. All the linked files in the project will be referenced relative to the path set in the Environment variable.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Please specify an Environment variable for the root directory path for linked files in the project. [OK]
None	The Environment variable '' specified for the root directory path for linked files is not defined/set in your environment. Please make sure to set the appropriate root directory path in this Environment variable before you open Libero. You will need to exit Libero and open a new session for any changes you made to the directory path set in the Environment variable to reflect here.
None	use_relative_path: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'linked_files_root_dir_env' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'project_general_settings [-linked_files_root_dir_env "linked files root directory"] [-use_relative_path "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example modifies the project general settings.

```
project_general_settings \
             -use_relative_path 1 \
             -linked_files_root_dir_env {MSCC_ROOT_2}
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.70. project_settings (Ask a Question)

Description

This Tcl command modifies project flow settings for your Libero SoC project. The **Project Settings** dialog box enables you to modify your Device, HDL, and **Design Flow** settings and your Simulation Options. In Libero SoC, from the **Project** menu, click **Project Settings**.

```
project_settings \
[-hdl "VHDL | VERILOG"] \
[-verilog_mode "SYSTEM_VERILOG | VERILOG_2K"] \
[-vhdl_mode "VHDL_2008 | VHDL_93"] \
[-auto_update_modelsim_ini "TRUE | FALSE"] \
[-auto_update_viewdraw_ini "TRUE | FALSE"] \
[-standalone_peripheral_initialization "TRUE | FALSE"] \
[-ondemand_build_dh "TRUE | FALSE"] \
[-auto_generate_synth_hdl "TRUE | FALSE"] \
[-auto_run_drc "TRUE | FALSE"] \
[-auto_generate_viewdraw_hdl "TRUE | FALSE"] \
[-auto_file_detection "TRUE | FALSE"] \
[-enable_set_mitigation "TRUE | FALSE"] \
[-enable_design_separation "TRUE" | "FALSE"] \
[-display_fanout_limit "display_fanout_limit"] \
[-block_mode "TRUE | FALSE"] \
[-abort_flow_on_sdc_errors "TRUE | FALSE"] \
[-abort_flow_on_pdc_errors "TRUE | FALSE"] \
[-sim_flow_mode "TRUE | FALSE"] \
[-auto_generate_physynth_hdl "TRUE | FALSE"] \
[-instantiate_in_smartdesign "TRUE | FALSE"] \
[-enable_viewdraw "TRUE | FALSE"] \
[-vm_netlist_flow "TRUE | FALSE"]
[-system_verilog_mfcu "TRUE | FALSE"]
[-abort_flow_on_3.3V_IO "1 | 0"]
```

Arguments

Parameter	Type	Description
hdl	string	Sets your project HDL type. Valid values are: VHDL or VERILOG.
verilog_mode	string	Sets the Verilog standard to Verilog-2001 or System Verilog. Valid values are: VERILOG_2K or SYSTEM_VERILOG. The default value is SYSTEM_VERILOG.
vhdl_mode	string	Sets the VHDL standard. Valid values are: VHDL-2008 or VHDL-1993. The default value is VHDL-2008.
auto_update_modelsim_ini	boolean	Sets the auto-update `modelsim.ini` file option. Valid values are True, true, 1, FALSE, false or 0: • TRUE, true, 1 - updates the `modelsim.ini` file automatically. The default value is 1. • FALSE, false, 0 - modelsim.ini file does not update automatically.
auto_update_viewdraw_ini	boolean	Sets the auto-update `viewdraw.ini` file option. Valid values are True, true, 1, FALSE, false or 0: • TRUE, true, 1 - updates the `viewdraw.ini` file automatically. The default value is 1. • FALSE, false, 0 - `viewdraw.ini` file does not update automatically.
block_mode	boolean	Enables you to create and publish design blocks (`*.cxz` files) in Libero® SoC. Puts the Project Manager in Block mode. Design blocks are low-level components that may have completed the place-and-route step and met the timing and power requirements. These low-level design blocks can then be imported into a Libero® SoC project and re-used as components in a higher level design. By default, this box is unchecked. • TRUE, true, 1 - Enable to create and publish design blocks file. • FALSE, false, 0 - Disables to create and publish design blocks file.

© 2025 Microchip Technology Inc. and its subsidiaries

project_settings (continued)		
Parameter	**Type**	**Description**
auto_generate_synth_hdl	boolean	Auto-generates your HDL file after synthesis. Valid values are: • TRUE, true, 1 - Enables auto-generation of your HDL file after synthesis. • FALSE, false, 0 - Disables auto-generation of your HDL file after synthesis. The default value is 0.
auto_run_drc	boolean	Automatically runs the design rule check immediately after synthesis. Valid values are: • TRUE, true, 1 - Enables and auto-runs the design rule check immediately after synthesis. • FALSE, false, 0 - Disables and auto-runs the design rule check immediately after synthesis. the default value is 0.
auto_generate_viewdraw_hdl	boolean	Automatically generates your HDL netlist after Save and Check in ViewDraw. Valid values are: • TRUE, true, 1 - Enables auto-generation of your HDL netlist. The default value is 1. • FALSE, false, 0 - Disables auto-generation of your HDL netlist.
auto_file_detection	boolean	Automatically detects when new files have been added to the Libero SoC project folder. Valid values are: • TRUE, true, 1 - Automatically detects new added files in Libero SoC project. The default value is 1. • FALSE, false, 0 - Automatically does not detect newly added files in Libero SoC project.
standalone_peripheral_initialization	boolean	Use Standalone Initialization for MDDR/FDDR/SERDES Peripherals – Check this box if you want to create your own peripheral initialization logic in SmartDesign, for each of your design peripherals (MDDR/FDDR/SERDES). When checked, System Builder does not build the peripherals initialization logic for you. Standalone initialization is useful if you want to make the initialization logic of each peripheral separate from and independent of each other. By default, this box is checked. Valid values are: TRUE, true, 1 , FALSE, false or 0.
ondemand_build_dh	integer	Enable/disable On Demand Build Design Hierarchy. Valid values are: TRUE, true, 1, FALSE, false or 0. The default value is 1.
enable_set_mitigation	boolean	Enable Single Event Transient mitigation - controls the mitigation of Single Event Transient (SET) in the FPGA fabric. When this box is checked, SET filters are turned on globally (including URAM, LSRAM, MACC, I/O FF, Regular FF, DDR_IN, DDR_OUT) to help mitigate radiation-induced transients. By default, this box is unchecked. Valid values are: TRUE, true, 1, FALSE, false or 0.
enable_design_separation	integer	Set it to “1” if your design is for security and safety critical applications and you want to make your design’s individual subsystems (design blocks) separate and independent (in terms of physical layout and programming) to meet your design separation requirements. When set to “1”, Libero generates a parameter file (`MSVT.param`) that details the design blocks present in the design and the number of signals entering and leaving a design block. Microchip provides a separate tool, known as Microchip Separation Verification Tool (MSVT), which checks the final design place-and-route result against the `MSVT.param` file and determines whether the design separation meets your requirements. The default value is 1. This option available for SmartFusion® 2 and IGLOO® 2.
display_fanout_limit	integer	Use this option to set the limit of high fanout nets to be displayed; the default value is 10. This means the top 10 nets with the highest fanout will appear in the `root_compile_netlist.log` file.

© 2025 Microchip Technology Inc. and its subsidiaries

project_settings (continued)		
Parameter	**Type**	**Description**
abort_flow_on_pdc_errors	boolean	Abort flow if errors are found in Physical Design Constraints(PDC). Valid values are: TRUE, true, 1, FALSE, false or 0. By default, this box is checked.
abort_flow_on_sdc_errors	boolean	Abort flow if errors are found in Timing Constraints (SDC). Valid values are: TRUE, true, 1, FALSE, false or 0. By default, this box is checked.
auto_generate_physynth_hdl	boolean	Auto-generates your HDL file after physical synthesis. Valid values are: TRUE, true, FALSE, false or 0. The default value is 1.
instantiate_in_smartdesign	boolean	Instantiate System Builder/MSS component in a SmartDesign on creation. Uncheck this box if you are using this project to create System Builder or MSS components and do not plan on using them in a SmartDesign-based design. This is especially useful for design flows where the System Builder or MSS components are stitched in a design using HDL. Valid values are: TRUE, true, 1, FALSE, false or 0.
enable_viewdraw	boolean	Enable/disable to create a schematic source file in Libero SoC. Valid values are: TRUE, true, 1, FALSE, false or 0.
vm_netlist_flow	boolean	Sets Synthesis gate level netlist format. By default, this box is checked. Valid values are: TRUE, true, 1, FALSE, false or 0. • TRUE, true, 1 - sets Verilog netlist format. • FALSE, false, 0 - sets EDIF netlist format.
sim_flow_mode	boolean	Instantiate simulation cores in your SmartDesign Testbench. Simulation cores are basic cores that are useful for stimulus, such as driving clocks, resets, and pulses. Valid values are: TRUE, true, 1, FALSE, false or 0.
system_verilog_mfcu	boolean	Enable/disable the System Verilog Multi-File Compilation Unit (MFCU). Valid values are: TRUE, true, 1, FALSE, false or 0. **Note:** This option is only applicable when -verilog_mode is "SYSTEM_VERILOG".
abort_flow_on_3.3V_IO	boolean	The following options are supported for this parameter. • 1 - The Place-and-Route tool is aborted and an error message is generated in the message log, if 3.3V I/Os are found in the design. • 0 - The Place-and-Route tool generates the same error message as a warning without interrupting the design flow. **Note:** This option is only applicable for RT PolarFire® family of devices.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

SmartFusion 2, IGLOO 2:

The following example sets your project to VHDL, disables the auto-update of the ModelSim INI or ViewDraw INI files, enables the automatic generation of HDL after synthesis, enables auto-detection

© 2025 Microchip Technology Inc. and its subsidiaries

for files, sets the display of high fanout nets to the top 12 high fanout nets, enables SET filters to mitigate radiation-induced transients, and enables design separation methodology for the design.

```
project_settings -hdl "VHDL" \
-auto_update_modelsim_ini "FALSE" \
-auto_update_viewdraw_ini "FALSE"\
-block_mode "FALSE" -auto_generate_synth_hdl "TRUE" \
-auto_file_detection "TRUE" \
-display_fanout_limit {12} \
-enable_set_mitigation {1}
```

3.71. publish_block (Ask a Question)

Description

Tcl command publishes a block with the conditions related to place and route. This is the project_setting command. To enable Block Creation for a new project:

1. Select **New Project** from the **Libero SoC Project** menu.
2. Check the **Enable Block Creation** checkbox.
3. Select the **Enhanced Constraint Flow** for the new project. In an existing project, from the **Project** menu, select **Project Settings** > **Design Flow** and check the **Enable Block Creation** checkbox.
4. After Block Creation is enabled, **Publish Block** appears in the **Design Flow** window.
5. Expand **Publish Design**, right-click **Publish Block** and select **Export** from the context-menu that appears. By default, this option is not unchecked.

```
publish_block -file {absolute or relative path} \
              [-publish_placement value] \
              [-publish_routing value] \
              [-publish_region value] \
              [-vhdl value ]
```

Arguments

Parameter	Type	Description
file	string	Specifies the location (absolute or relative) to publish the block. Default is /designer/<designer_name>/<designer_name>.cxz.
publish_placement	boolean	Valid values are: TRUE, 1, true, FALSE, 0 or false • FALSE, false, 0 - no placement or routing will be published and preserved. Only the netlist is preserved. • TRUE, true, 1 - publishes placement if all the macros in your design are placed or assigned to a region. Default this box is checked.
publish_routing	boolean	Valid values are: TRUE, 1, true, FALSE, 0 or false • FALSE, false, 0 - routing will not be published and added to the block. This block will be completely rerouted completely in the top design. • TRUE, true, 1 - publish routing to be part of the block. `publish_placement` must be 1 for this option to take effect. All the macros should be placed or assigned to a region. Default this box is checked.

© 2025 Microchip Technology Inc. and its subsidiaries

publish_block (continued)		
Parameter	**Type**	**Description**
publish_region	boolean	Valid values are: TRUE, 1, true, FALSE, 0 or false • FALSE, false, 0 - region constraints are not added to the block published. • TRUE, true, 1 - Region constraints will be published and preserved. This is not recommended and should be done only if the user wants to keep the regions in the top design. Example: the user wants to see an empty region in the top design. In general, the regions used to control placement should not be part of the block. Default this box is checked.
vhdl	boolean	Valid values are: TRUE, 1, true, FALSE, 0 or false • FALSE, false, 0 - generates a Verilog netlist to be used for synthesis and simulation. Default value. • TRUE, true, 1 - generates a VHDL file format.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Required parameter 'publish_placement' is missing.
None	Required parameter 'publish_routing' is missing.
None	Required parameter 'publish_region' is missing.
None	publish_placement: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	publish_routing: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	publish_region: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	vhdl: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'publish_block -file "Publish Block file Name" \ -publish_placement "TRUE | FALSE" \ -publish_routing "TRUE | FALSE" \ -publish_region "TRUE | FALSE" \ [-vhdl "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example exports Publish Block file in the `\test_block\designer\top\top.cxz` file, generates a Verilog netlist. Publishes placement and region if all the macros in your design are placed or assigned to a region:

```
publish_block -file {D:\test_block\designer\top\top.cxz} \
              -publish_placement 1 -publish_routing 1 -publish_region 1 \
              -vhdl 0
```


© 2025 Microchip Technology Inc. and its subsidiaries

3.72. refresh (Ask a Question)

Description

This Tcl command refreshes your project, rebuilds the Design Hierarchy, updates the view and checks for updated links and files. This command is equivalent to selecting **Refresh Design Hierarchy (F5)** from the **View** menu.

```
refresh
```

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command refreshes project, rebuilds the Design Hierarchy, updates the view and checks for updated links and files.

```
refresh
```

3.73. remove_core (Ask a Question)

Description

This Tcl command removes a core from your project.

```
remove_core [-vlnv "Vendor:Library:Name:Version" ]
```

Arguments

Parameter	Type	Description
vlnv	string	Specifies the version identifier of the core being configured. It is mandatory.

Error Codes

Error Code	Description
None	The core is not in the vault.
None	The supplied name is not a valid VNLV core id: 'value'.
None	Parameter 'vlnv' is missing or has invalid value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_core [-vlnv "vlnv"]+ '

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example removes Simulation RESET_GENR from project.

```
remove_core -vlnv {Actel:Simulation:RESET_GEN:1.0.1}
```

See Also

- configure_core
- download_core
- create_and_configure_core

3.74. remove_library (Ask a Question)

Description

This Tcl command removes a VHDL library from your project. To remove library, right-click the design module name in the **Design Hierarchy** and select **Remove VHDL Library** from the context menu.

```
remove_library -library name
```

Arguments

Parameter	Type	Description
library	string	Specifies the name of the library you wish to remove.

Error Codes

Error Code	Description
None	Required parameter 'library' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_library -library "library name"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Remove a library with the "my_lib" name from your project.

```
remove_library -library "my_lib"
```

See Also

- add_library
- rename_library

© 2025 Microchip Technology Inc. and its subsidiaries

3.75. remove_profile (Ask a Question)

Description

This Tcl command deletes a tool profile.

```
remove_profile -name profilename
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the profile you wish to delete.

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	Profile 'profilename' does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_profile -name "profile name"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command deletes the profile 'custom1'.

```
remove_profile -name "custom1"
```

See Also

- add_profile
- select_profile
- edit_profile
- export_profiles

3.76. rename_file (Ask a Question)

Description

This Tcl command copies/renames a file specified by the `-file` parameter to a different name specified by the `-target` parameter. Creates a new file in the <project_name> directory.

```
rename_file -file { absolute path and the name of file } -target { new_filename }
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute path and original name of the file.

© 2025 Microchip Technology Inc. and its subsidiaries

rename_file (continued)		
Parameter	**Type**	**Description**
target	string	Specifies the just new name of the file. If specified constraint file, then created new file in constraint directory, otherwise new file created in the <project_name> directory, not renamed.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'file' has illegal value.
None	Required parameter 'target' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command copies the constraint a.sdc file specified by the `-file` parameter to a b.sdc specified by `-target` parameter in the "c:/user/" directory.

```
rename_file -file {c:/user/a.sdc} -target {b.sdc}
```

The following command copies the a.v verilog file specified by the `-file` parameter to a b.v specified by `-target` parameter in the project directory not in hdl.

```
rename_file -file { /libero_prj/hdl/a.v } -target {b.v}
```

3.77. rename_library (Ask a Question)

Description

This Tcl command renames a VHDL library from your project. To renames library, right-click the design module name in the Design Hierarchy select Rename VHDL Library from the context menu.

```
rename_library -library "library name" -name "library new name"
```

Arguments

Parameter	Type	Description
library	string	Identifies the current name of the library that you wish to rename.
name	string	Specifies the new name of the library.

Error Codes

Error Code	Description
None	Required parameter 'library' is missing.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'rename_library -library "old library name" -name "new library name"' .

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example renames a library from 'my_lib' to 'my_new_lib'.

```
rename_library -library "my_lib' -name "my_new_lib"
```

See Also

- add_library
- remove_library

3.78. run_tool (Ask a Question)

Description

This Tcl command starts the specified tool. For tools that support command files, an optional command file can be supplied through the `-script` parameter. Select a root before running this tool.

Note: Where possible, the value of `tool_name` corresponds to the name of the tool in Libero SoC. Invoking some tools will cause Libero SoC to automatically run some upstream tools in the design flow. For example, invoking Place and Route will invoke Synthesis (if not already run) before it runs Place and Route.

```
run_tool -name {tool_name} \
         [-script {absolute or relative path to script file} ]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specified tool name. The following is a list of supported tool names. • SYNTHESIZE • COMPILE • SIM_PRESYNTH • SIM_POSTSYNTH • SIM_POSTLAYOUT • PLACEROUTE • VERIFYTIMING • VERIFYPOWER • GENERATEPROGRAMMINGFILE • GENERATE_MEMORY_MAP • PROGRAMDEVICE • CONFIGURE_CHAIN • SMARTDEBUG • SSNANALYZER • UPDATE_ENVM • UPDATE_UPROM • EXPORTNETLIST • EXPORTSDF • GENERATEPROGRAMMINGDATA • GENERATEDEBUGDATA • GENERATE_SPI_FLASH_IMAGE • PROGRAM_SPI_FLASH_IMAGE
script	string	Specify absolute or relative path of the script file. This is an optional parameter.

Supported tool_names

The following table lists the supported tool names for `run_tool -name {tool_name}`.

tool_name	Parameter	Description
SYNTHESIZE	N/A	Runs synthesis on your design.
COMPILE	N/A	Runs Compile with default or configured settings.
SIM_PRESYNTH	N/A	Runs pre-synthesis simulation with your default simulation tool
SIM_POSTSYNTH	N/A	Runs post-synthesis simulation with your default simulation tool.
SIM_POSTLAYOUT	N/A	Runs the post layout simulation on the simulation tool.
PLACEROUTE	N/A	Runs Place and Route tool with default or configured settings.
VERIFYTIMING	-script {`script_file`}	Runs timing analysis with default settings/configured settings in `script_file`.
VERIFYPOWER	-script {`script_file`}	Runs power analysis with default settings/configured settings in `script_file`.
GENERATEPROGRAMMINGFILE	N/A	Generates the bitstream used for programming within Libero. This is similar to executing "Generate FPGA Array Data" in the Libero design flow.

© 2025 Microchip Technology Inc. and its subsidiaries

run_tool (continued)

tool_name	Parameter	Description
GENERATE_MEMORY_MAP (SmartFusion 2, IGLOO 2 and RTG4 only)	N/A	Exports an XML file in <prj_folder> component/work/ <design> /<design>_DataSheet.xml. The file contains information about your root SmartDesign in your project.
PROGRAMDEVICE	N/A	Programs your device with configured parameters.
CONFIGURE_CHAIN	-script {script_file}	Takes a script that contains FlashPro-specific Tcl commands and passes them to FlashPro Express for execution.
SMARTDEBUG	-script {script_file}	Takes a script that contains SmartDebug-specific Tcl commands and passes them to SmartDebug for execution.
SSNANALYZER	-script {script_file}	Takes a script that contains Simultaneous Switching Noise (SSN)-specific Tcl commands and passes them to the SSN tool for execution. Simultaneous Switching Noise (SSN) is a Libero SoC tool that analyzes and generates a Nosie Margin report for I/Os after layout. Depends on Die ane Package.
UPDATE_ENVM (SmartFusion 2 and IGLOO 2 only)	-script {update_config_file}	Takes a script file that updates the client(s) in the ENVM. In the script file, the client(s) to be updated may be a serialization client or a data storage client or a mix of serialization clients and data storage clients.
UPDATE_UPROM (RTG4 only)	-script {update_config_file}	Takes a script file that updates the data storage client(s) in RTG4 UPROMS.
EXPORTNETLIST	N/A	This command exports a .v/.vhd netlist file to the active synthesis implementation folder.
EXPORTSDF	N/A	This command exports the back annotated files to the designer/impl1 folder.
GENERATEPROGRAMMINGDATA	N/A	Generates the files needed by generating programming bitstream files.
GENERATEDEBUGDATA (PolarFire only)	N/A	Generates the files needed by SmartDebug during device debug.
GENERATE_SPI_FLASH_IMAGE (PolarFire only)	N/A	Generates SPI Flash Image file used for programming SPI FLASH Image within Libero.
PROGRAM_SPI_FLASH_IMAGE (PolarFire only)	N/A	Programs SPI Flash Image with configured parameters.

Error Codes

Error Code	Description
None	You must specify the Tcl script to run with the CONFIGURE_CHAIN tool.
None	You must specify the Tcl script to run with the SMARTDEBUG tool.
None	PROGRAMDEVICE: No programmer is connected.
None	SPI Flash Memory is not configured. Use the Configure Design Initialization Data and Memories tool to configure it.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

```
run_tool \
-name {COMPILE}
run_tool \
-name {SYNTHESIZE}
# control _synopsys.tcl contains the synthesis-specific Tcl commands
run_tool \
-name {VERIFYTIMING} \
-script {./SmartTime.tcl}
# Script file contains SmartTime-specific Tcl commands
run_tool \
-name {VERIFYPOWER} \
-script {./SmartPower.tcl}
# Script file contains SmartPower-specific Tcl commands
run_tool \
-name {SMARTDEBUG}
-script {./sd_test.tcl}
# Script file contains SmartDebug-specific Tcl commands
```

3.79. save_log (Ask a Question)

Description

This Tcl command saves your Libero SoC log file.

```
save_log -file { absolute or relative path of log file }
```

Arguments

Parameter	Type	Description
file	string	Specifies the log file name. It must be the absolute or relative path.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example saves the log of Libero SoC with the name 'my_logfile.log'.

```
save_log -file {my_logfile.log}
```

3.80. save_project (Ask a Question)

Description

This Tcl command saves the current project in Libero SoC. Equivalent to clicking the File menu, and choosing Save Project.

```
save_project
```


© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Saves the project in your current working directory.

```
save_project
```

See Also

- new_project
- close_project

3.81. save_project_as (Ask a Question)

Description

This Tcl command saves the current project in Libero SoC with a different name and in a specified directory. You must specify a location using the `-location` parameter.

```
save_project_as -name "project name" \
                -location "project location" \
                [-files "all | project | sources" ] \
                [-replace_links "TRUE | FALSE" ]
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of your new project.
location	string	Must include the complete path of the PRJ file. If you do not provide the full path, Libero SoC infers that you want to save the project to your current working directory. This is a required parameter.
files	string	Specifies the files you want to copy into your new project. • all - copies all your files into your new project. • project - copies only your Libero SoC project files into your new project. • sources - copies only the source files into your new project.
replace_links	boolean	Specifies whether or not you want to update your file links in your new project. • true | 0 - replaces (updates) the file links in your project during your save. • false | 1 - saves your project without updating the file links.

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'location' is missing.
None	replace_links: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).

© 2025 Microchip Technology Inc. and its subsidiaries

save_project_as (continued)	
Error Code	**Description**
None	files: Invalid argument value: 'value' (expecting all, project or sources).
None	Parameter 'param_name' is not defined. Valid command formatting is 'save_project_as -location "project location" -name "project name" [-replace_links "TRUE | FALSE"] [-archive_project "TRUE | FALSE"] [-files "all | project | sources"] [-designer_views "all | current | none"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Saves your current Libero SoC project as mydesign.prjx in the c:/netlists/testprj/mydesign directory.

```
save_project_as -location {c:/netlists/testprj/mydesign} -name mydesign.prjx
```

See Also

- new_project
- open_project
- save_project

3.82. save_smartdesign (Ask a Question)

Description

This Tcl command saves all the changes made in a SmartDesign component.

```
save_smartdesign -sd_name "smartdesign_component_name"
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component to be saved. It is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'save_smartdesign -sd_name "sd_name" '.

Supported Families

Supported Families
PolarFire®
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

save_smartdesign (continued)
Supported Families
SmartFusion® 2
IGLOO® 2

Example

The following example saves "top" SmartDesign.

```
save_smartdesign -sd_name {top}
```

See Also

- open_smartdesign

3.83. select_profile (Ask a Question)

Description

This Tcl command selects an existing profile to use in your project.

```
select_profile -name profilename
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the profile you wish to use.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command selects an existing profile 'custom1'.

```
select_profile -name custom1
```

See Also

- add_profile
- edit_profile
- remove_profile
- export_profiles

3.84. set_actel_lib_options (Ask a Question)

Description

This Tcl command enables you to choose the default library for your device, or to specify your own library. Enter the full pathname of your own library to use it for simulation.

```
set_actel_lib_options -use_default_sim_path {0|1} [-sim_path "path" ]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
use_default_sim_path	boolean	Enables/Disables you to choose the default library for your device. This is mandatory. Valid values are: • TRUE, true, 1 - Uses the default simulation library. • FALSE, false, 0 - Disables the default simulation library; enables you to specify a different simulation library with the `-sim_path {full pathname}` option.
sim_path	string	Specifies the full pathname to your simulation library. It is optional.

Error Codes

Error Code	Description
None	Required parameter 'use_default_sim_path' is missing.
None	use_default_sim_path: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_actel_lib_options -use_default_sim_path "TRUE | FALSE" [-sim_path "path"] '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example specifies "c:\sim_lib\test." as simulation library.

```
set_actel_lib_options -use_default_sim_path FALSE -sim_path {c:\sim_lib\test}
```

See Also

- set_modelsim_options

3.85. set_active_testbench (Ask a Question)

Description

This Tcl command sets the active testbench module to be executed during the simulation process.

Important: To specify a custom list of stimulus files needed for the simulation, use the `associate_stimulus` command.

```
set_active_testbench "<testbench_name>::<library_name>,<file_path>"
```

Error Codes

Error Code	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Example 1: This example instructs the tool to use the `tb1` module from the `work` library and specifies the location of the source file (`stimulus/tb1.v`).

```
set_active_testbench "tb1::work,stimulus/tb1.v"
```

Example 2: This example instructs the tool to use the `sdtb1` SmartDesign testbench module from the `work` library and specifies the location of the source file (`component/work/sdtb1/sdtb1.v`).

```
set_active_testbench "sdtb1::work,component/work/sdtb1/sdtb1.v"
```

3.86. set_as_target (Ask a Question)

Description

This Tcl command sets a SDC, PDC file as the target file to receive and store new constraints. To set *.sdc *.pdc files as target, click the Manage Constraints tool name from Design Flow.

```
set_as_target [-type { constraint_file_type to associate } ] \
              -file { file to set as target }
```

Arguments

Parameter	Type	Description
type	string	Specifies the file type: sdc, io_pdc or fp_pdc.
file	string	Specifies the file which shoukd be set as target.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	-file option can't be empty.Please specify a valid path
None	Invalid argument passed to -type option, the valid values are io_pdc/fp_pdc/sdc
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_as_target [-type "type to associate"] -file "file to set as target"' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This command sets the SDC file project_folder /constraints/user.sdc as the target to receive and store new SDC commands.

```
set_as_target -type {sdc} -file {./constraint/user.sdc}
```

This command sets the PDC file project_folder /constraints/user.pdc as the target to receive and store new PDC commands.

```
set_as_target -type {pdc} -file {./constraint/user.pdc}
```

See Also

- unset_as_target

3.87. set_device (Ask a Question)

Description

This Tcl command sets your device family, die, and package in the Project Manager. This command generates set_device.log file that contains information of device details. The file is saved in the `<project_name>/tooldata` folder.

Important: The changes you made may invalidate your components and/or design flow; you may have to regenerate your components and rerun your design flow.

```
set_device [-family "family" ] [-die "die" ] [-die_voltage "value" ] \
           [-part_range "part_range" ] [-package "package" ] \
           [-speed "speed_grade" ] [-adv_options "value" ]
```

Arguments

Parameter	Type	Description
family	string	Sets device family. If you have already set device name, you cannot change the device family name for the current project, you can change the rest of the arguments.
die	string	Sets device die. It is optional.
die_voltage	floating point	Sets the die voltage for your project. Possible values depend on your device. See your device datasheet for details.
part_range	string	Sets your default temperature range for your project • PolarFire: EXT, IND, MIL • SmartFusion 2: COM, IND, TGrade 2, MIL • IGLOO 2: COM, IND, TGrade 1, TGrade 2, MIL • RTG4: MIL
package	string	Sets device package. It is optional.
speed	string	Sets device speed grade. Valid value is: -1 and STD. It is optional.

© 2025 Microchip Technology Inc. and its subsidiaries

set_device (continued)		
Parameter	**Type**	**Description**
adv_options	string	Sets your advanced options, such as temperature and voltage settings. • IO_DEFT_STD:LVTTL - Sets your I/O default value to LVTTL. • TEMPR:COM - Sets your default temperature range; can be COM (Commercial), MIL (Military) or IND (industrial). • VCCI_1.5_VOLTR:COM - Sets VCCI to 1.5 and voltage range to Commercial. • VCCI_1.8_VOLTR:COM - Sets VCCI to 1.8 and voltage range to Commercial. • VCCI_2.5_VOLTR:COM - Sets VCCI to 2.5 and voltage range to Commercial. • VCCI_3.3_VOLTR:COM - Sets VCCI to 3.3 and voltage range to Commercial. • VOLTR:COM - Sets your voltage range; can be COM (Commercial), MIL (Military) or IND (industrial). • RESTRICTPROBEPINS:1 - Sets to 1 to reserve your pins for probing if you intend to debug using SmartDebug.

You can pass custom specific temperature ranges depending on the selected part range.

The value parameter needs minimum, typical and maximum temperatures values to be added. The following table denotes the minimum, typical and maximum temperatures that are preset by the tool. You can set any value for the temperatures present within the range.

Part Range	Minimum	Typical	Maximum
EXT	0	25	100
COM (Available only for SmartFusion 2 and IGLOO 2 devices)	0	25	85
IND	- 40	25	100
MIL	- 55	25	125

Error Codes

Error Code	Description
None	Unable to select speed 'ALL'. The speed value must be one of the following: '-1 STD'.
None	Unable to select family 'value'. The family value must be one of the following: 'IGLOO 2, SmartFusion 2, RTG4, PolarFire, PolarFireSoC' .
None	Unable to select die 'MPF0T'. The die value must be one of the following: 'MPF300T_ES MPF300TS_ES MPF050T MPF050TS MPF100T MPF100TS MPF200T MPF200TS MPF300T MPF300TS MPF500T MPF500TS MPF300XT MPF050TL MPF100TL MPF200TL MPF300TL MPF500TL MPF050TLS MPF100TLS MPF200TLS MPF300TLS MPF500TLS RTPF500T RTPF500TL RTPF500TS RTPF500TLS' .
None	Invalid value "TGrade2" for PART_RANGE variable; expected one of COM IND.
None	Invalid value "MIL" for VCCI_1.2_VOLTR variable; expected one of COM IND.
None	No family was specified. Select a family before proceeding.

© 2025 Microchip Technology Inc. and its subsidiaries

set_device (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device [-family "family"] [-die "die"] [-package "package"] [-speed "speed"] [-die_voltage "die_voltage"] [-part_range "part_range"] [-adv_options "adv_options"]*.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command sets SmartFusion 2 family with M2S005 die, 484 FBGA package, {-1} speed, IND part range, temperature and voltage settings to your device.

```
set_device -family {SmartFusion 2} \
-die {M2S005} \
-package {484 FBGA} \
-speed {-1} \
-die_voltage {1.2} \
-part_range {IND} \
-adv_options {IO_DEFT_STD:LVCMOS 2.5V} \
-adv_options {RESERVEMIGRATIONPINS:1} \
-adv_options {VCCI_2.5_VOLTR:COM} \
-adv_options {VCCI_3.3_VOLTR:COM} \
-adv_options {VOLTR:IND}
```

The following commands sets RTG4 family with RTG4150_ES die, 1657_CG package, {STD} speed, MIL part range, and custom temperature range set at a minimum = 0, typical = 25, and maximum = 70.

```
set_device -family {RTG4} -die {RTG4150_ES} -package {1657_CG} -speed {STD} -die_voltage
{1.2} -part_range {MIL} -adv_options  {TEMPR:0 25 70}
```

The following commands sets PolarFire family with MPF200T die, FCG784 package, {-1} speed, EXT part range, and custom temperature range set at a minimum = 0, typical = 25, and maximum = 70.

```
set_device -family {PolarFire} -die {MPF200T} -package {FCG784} -speed {-1} -part_range {EXT}
-adv_options  {TEMPR:0 25 70}
```

3.88. set_global_include_file (Ask a Question)

Description

This Tcl command for setting a file as global include file.

```
set_global_include_file -file {filename}
```

Arguments

Parameter	Type	Description
file	string	The user provides the file path to set as global include file. The file path can be absolute path, linked files path or path with Environment variable from project settings.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Invalid Global Include File 'value'.
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_global_include_file -file "set global include file"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example sets global include file with absolute file path.

```
set_global_include_file -file {E:\prj\global_include_files\hdl\add.h}
```

ENV from Project settings

```
set_global_include_file -file {${ENV_2}/xyz.h}
```

Linked Files TCL variable

```
set_global_include_file -file "$proj/hdl/abcd.v"
```

3.89. set_global_include_path_order (Ask a Question)

Description

This Tcl command is for providing global include paths information.

```
set_global_include_path_order [-paths {“path_1” “path_2” “path_3”}]
```

Arguments

Parameter	Type	Description
paths	string	The user provides the list of paths to be considered as global include paths.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	One or more Global Include Paths set are invalid or do not exist.
None	Invalid Global Include Path - 'value'.

© 2025 Microchip Technology Inc. and its subsidiaries

set_global_include_path_order (continued)	
Error Code	**Description**
None	Parameter 'paths' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_global_include_path_order [-paths "[paths]+"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example considers as global include the following paths.

```
set_global_include_path_order -paths { “./src/GLOBAL_INCLUDE_PATH_1” “./src/
GLOBAL_INCLUDE_PATH_2” “${ENV_VAR}/GLOBAL_INCLUDE_PATH_1”}
```

3.90. set_modelsim_options (Ask a Question)

Description

This Tcl command sets your ModelSim simulation options. You can set change how Libero SoC handles Do files in simulation, import your own Do files, set simulation run time, and change the DUT name used in your simulation. You can sets options from the **Project Settings** > **Simulation** menu. Default values are used if parameters are omitted.

```
set_modelsim_options \
[-use_automatic_do_file "TRUE | FALSE"] \
[-user_do_file {path}] \
[-sim_runtime {value}] \
[-tb_module_name {value}] \
[-tb_top_level_name {value}] \
[-include_do_file "TRUE | FALSE" \
[-included_do_file {value}] \
[-type {value}] \
[-resolution {value}] \
[-add_vsim_options {value}] \
[-display_dut_wave "TRUE | FALSE"] \
[-log_all_signals "TRUE | FALSE"] \
[-do_file_args {value}] \
[-dump_vcd "TRUE | FALSE"] \
[-vcd_file "VCD file name"] \
[-sdf_corner "sdf_corner"] \
[-verilog {value}] \
[-VHDL {value}] \
[-disable_pulse_filtering "TRUE | FALSE"] \
[-timeunit {value}] \
[-timeunit_base {value}] \
[-precision {value}] \
[-precision_base {value}]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
use_automatic_do_file	boolean	Automatically create a DO file that enables you to simulate your design. Following are the valid values: • TRUE, true, 1 - uses the default `automatic.do` file in your project. This box is checked by default. • FALSE, false, 0 - uses a different `*.do` file; use the other simulation options to specify it.
user_do_file	string	Specifies the location of your user-defined `*.do` file.
sim_runtime	number and unit of time	Sets your simulation runtime. It is optional. Value is the number and unit of time, such as {1000 ns}.
tb_module_name	string	Specifies your test bench module name, where value is the name. Default value is "test bench".
tb_top_level_name	string	Sets the top-level instance name in the test bench, where value is the name. Default is <top>_0.
include_do_file	boolean	Enables you to include DO file. Valid values are: • TRUE, true, 1 - Includes the `*.do` file. • FALSE, false, 0 - Does not include the `*.do` file.
included_do_file	string	Specifies the path of the included `*.do` file, where the value is the name of the file. Including a DO file enables you to customize the set of signal waveforms that will be displayed in ModelSim. Specify this argument with `-include_do_file` argument. Default is `work.do`.
type	string	Resolution type; possible values are: • min - minimum. This is the default value. • typ - typical. • max - maximum.
resolution	unit of time	Sets your resolution value. Value is the number and unit of time, such as {1ps}. The default is family-specific, but you can customize it to fit your needs.
add_vsim_options	string	Adds more Vsim options, where value specifies the option(s).
display_dut_wave	boolean	Enables ModelSim to display signals for the tested design. Following are the possible values: • FALSE, false, 0 - displays the signal for the top_level_test bench. • TRUE, true, 1 - enables ModelSim to display the signals for the tested design.
log_all_signals	boolean	Saves and logs all signals during simulation. • TRUE, true, 1 - logs all signals. • FALSE, false, 0 - does not log all signals. Default this box checked.
do_file_args	list of strings	Specifies `*.do` file command parameters. Default is empty.
dump_vcd	boolean	Dumps the VCD file when simulation is complete. Following are the possible values: • TRUE, true, 1 - dumps the VCD file • FALSE, false, 0 - does not dump the VCD file. Default this box checked.
vcd_file	string	Specifies the name of the dumped VCD file, where value is the name of the file. Default is "`power.vcd`".

© 2025 Microchip Technology Inc. and its subsidiaries

set_modelsim_options (continued)		
Parameter	**Type**	**Description**
sdf_corner {parameter}	string	Sets the corner on which the post layout simulation will be done. • slow_lv_ht - slow process, low voltage and high temperature operating conditions. Default value. • slow_lv_lt - slow process, low voltage and low temperature operating conditions. • fast_hv_lt - fast process, high voltage and low temperature operating conditions.
verilog	integer	HDL Testbench file type can be either Verilog or VHDL, possible values are: 1 or 0. Default value is 0.
VHDL	integer	HDL Testbench file type can be either Verilog or VHDL, possible values are: 1 or 0. Default value is 0.
disable_pulse_filtering	boolean	Specifies to enable/disable pulse filtering during SDF based simulations. • TRUE, true, 1 - enable pulse filtering. • FALSE, false, 0 - disable pulse filtering. The default value is false.
timeunit	integer	TimeScale time unit value. Default value is 1.
timeunit_base	unit of time	TimeScale precision base value. The default setting is ns, possible values are: • s - second • ms - milisecond • us - microsecond • ns - nanosecond • ps - picosecond • fs - femtosecond
precision	integer	TimeScale precision value. Default value is 100.
precision_base	unit of time	TimeScale precision base value. The default setting is ps; possible values are: • s - second • ms - milisecond • us - microsecond • ns - nanosecond • ps - picosecond • fs - femtosecond

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Sets ModelSim options to use the automatic `*.do` file, sets simulation runtime to 1000 ns, sets the testbench module name to "testbench", sets the testbench top level to top_0, sets simulation type

© 2025 Microchip Technology Inc. and its subsidiaries

to "max", resolution to 1 ps, adds no vsim options, does not log signals, adds no additional DO file arguments, dumps the VCD file with a name `power.vcd`.

```
set_modelsim_options -use_automatic_do_file 1 -sim_runtime {1000ns} \
-tb_module_name {testbench} -tb_top_level_name {top_0} -include_do_file 0 \
-type {max} -resolution {1ps} -add_vsim_options {} -display_dut_wave 0 \
-log_all_signals 0 -do_file_args {} - dump_vcd 0 -vcd_file {power.vcd}
```

See Also

- set_actel_lib_options

3.91. set_option (Ask a Question)

Description

This Tcl command sets your synthesis and FPGA Hardware Breakpoint Auto Instantiation options on a module. Default values are used if parameters are omitted.

```
set_option [-synth "TRUE | FALSE" ] [-fhb "TRUE | FALSE" ] \
          [-module "module_name"]
```

Arguments

Parameter	Type	Description
synth	boolean	Enables/Disables Libero synthesis tool for root design in your project. Default is 1. Possible values are: • TRUE | true | 1 - Enables synthesis tool for your design. Default value. • FALSE | false | 0 - Disables synthesis tool for your design.
fhb	boolean	• TRUE | true | 1 - Enables FPGA Hardware Breakpoint Auto Instantiation. Default value. • FALSE | false | 0 - Disable FPGA Hardware Breakpoint Auto Instantiation.
module	string	Identifies the module on which you will run synthesis. Default is {<root_design_name>::work}.

Error Codes

Error Code	Description
None	Project Manager does not support PLACE, you cannot activate it in the flow.
None	Cannot find module named "top".
None	Parameter 'module' has illegal value.
None	synth: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	fhb:Invalid argument value: ' ' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'module' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_option [-synth "TRUE | FALSE"] [-fhb "TRUE | FALSE"] [-module "module"]

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

set_option (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example
The following command sets synthesis option on the module top.vhd.

```
set_option -synth "TRUE" -module "top"
```

3.92. set_root (Ask a Question)

Description
This Tcl command sets the module you specify as the root. Project has one top module that is the root of the design hierarchy for the implementation.

```
set_root -module "module name"
```

Arguments

Parameter	Type	Description
set_root	string	Specifies the name the module you want to set as root.

Error Codes

Error Code	Description
None	Cannot find module named "module_name".
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_root -module "module" '.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example
Set the module mux8 as root of your design hierarchy.

```
set_root mux8
```

3.93. set_user_lib_options (Ask a Question)

Description
This Tcl command sets your user library options during simulation. If you do not use a custom library these options are not available.

```
set_user_lib_options -name {Library name} \
                     -path {Library path} \
                     -option {value}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Sets the name of your user library. This is mandatory.
path	string	Sets the path name of your user library. This is mandatory.
option	string	Sets your default compile options on your user library. This is mandatory; possible values are: • do_not_compile - user library is not compiled. • refresh - user library is refreshed. • compile - user library is compiled. • recompile - user library is recompiled. • refresh_and_compile - user library is refreshed and compiled.

Error Codes

Error Code	Description
None	Required parameter 'path' is missing.
None	Required parameter 'name' is missing.
None	Required parameter 'option' is missing.
None	option: Invalid argument value: 'value' (expecting do_not_compile, refresh, compile, recompile or refresh_and_compile).
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_user_lib_options -name "Library name" -path "Library path" -option "do_not_compile | refresh | compile | recompile | refresh_and_compile" ' .

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The example below sets the name for the user library to "test1", the path to `c:/msemi_des_files/libraries/test1`, and the compile option to "do not compile".

```
set_user_lib_options -name {test1} -path {c:/msemi_des_files/libraries/test1} \
-option {do_not_compile}
```

3.94. smartdesign (Ask a Question)

Description

This Tcl command changes severity of Smart Design Data/ID Width mismatch, Memory Map DRC from error to warning vice versa.

```
smartdesign -memory_map_drc_change_error_to_warning {true|false} \
            -bus_interface_data_width_drc_change_error_to_warning {true|false} \
            -bus_interface_id_width_drc_change_error_to_warning {true|false}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
memory_map_drc_change_error_to_warning	Boolean	Specify true or 1 to change memory map error DRC to warning.
bus_interface_data_width_drc_change_error_to_warning	Boolean	Specify true or 1 to change bus interface data width mismatch error DRC to warning.
bus_interface_id_width_drc_change_error_to_warning	Boolean	Specify true or 1 to change bus interface ID width mismatch error DRC to warning.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartdesign [-memory_map_drc_change_error_to_warning true | false] [-bus_interface_data_width_drc_change_error_to_warning true | false] [-bus_interface_id_width_drc_change_error_to_warning true | false]

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example change severity of SmartDesign Data, ID Width mismatch, Memory Map Error to Warning DRC:

```
smartdesign -memory_map_drc_change_error_to_warning true \
            -bus_interface_data_width_drc_change_error_to_warning true \
            -bus_interface_id_width_drc_change_error_to_warning true
```

3.95. unlink_copy_locally (Ask a Question)

Description

This Tcl command removes a link to a file and copies the file locally to the project.

```
unlink_copy_locally -file {filename}
```

Arguments

Parameter	Type	Description
file	string	Name of the linked (remote) file you want to unlink and copy locally to the project.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Unable to find the file '*/src'.
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'unlink_copy_locally -file "file"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Unlink the file file1.vhd from my project and copy locally to the project.

```
unlink_copy_locally -file file1.vhd
```

3.96. unlink_files (Ask a Question)

Description

This Tcl command removes a link to a file in your project and it is not removed link to a folder. You can unlink a file form a folder, and to unlink the whole folder you have to unlink all files one by one from the link folder.

```
unlink_files \
       [-file {absolute or related path and name of the linked file}] \
       [-from_disk {0|1}]
```

Arguments

Parameter	Type	Description
file	string	Absolute or relative path and name of the linked (remote) file you want to unlink. There may be multiple -file arguments (see example below). You can repeat this argument to unlink multiple files.
from_disk	boolean	Removes file from disk. Valid values are: TRUE or 1, FALSE or 0.

Error Codes

Error Code	Description
None	Parameter 'file' is missing or has invalid value.
None	Required parameter 'file' is missing.
None	Unable to find the file.
None	Parameter 'param_name' is not defined. Valid command formatting is 'unlink_files [-file "file"]+ [-from_disk "TRUE | FALSE"] ' .

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

unlink_files (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

1. Unlink the file file1.vhd from my project.

```
unlink_files -file {E:\Share\file1.vhd}
```

2. Unlink files “E:\Share\abc.v” and “E:\Share\abc.sdc" from the project using Environment variable “MSCC_ROOT_1” that has the root directory path “E:\Share”.

```
unlink_files -file {${MSCC_ROOT_1}/abc.v} -file {${MSCC_ROOT_1}/abc.sdc}
```

See Also

- create_links
- change_link

3.97. unset_as_target (Ask a Question)

Description

This Tcl command unsets a target file in the Constraints view to not to store constraints. To unset *.sdc *.pdc files as target, click the Manage Constraints tool name from Design Flow.

```
unset_as_target -file { absolute path to file }
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute path to file to be unset as a target.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'unset_as_target -file "file to unset as target"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This command unsets the SDC file <project_folder>/constraint/user.sdc as target to not store SDC command.

```
unset_as_target -file {/prj_path/constraint/user.sdc}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- set_as_target

3.98. unset_global_include_file (Ask a Question)

Description

This Tcl command is intended for unsetting a global include file.

```
unset_global_include_file -file {filename}
```

Arguments

Parameter	Type	Description
file	string	The user provides the file path to unset as global include file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Invalid Global Include File 'value'.
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'unset_global_include_file -file "unset global include file"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example unsets {./src/define.h} global include file.

```
unset_global_include_file -file {./prj/hdl/define.h}
```

3.99. update_component_version (Ask a Question)

Description

This Tcl command allows to update a component of an instance with another version. Connections of the instance will be kept.

```
update_component_version -component_name {"comp_name"} -new_version {"version"} [-
download_core]
```

Arguments

Parameter	Type	Description
component_name	string	Name of the component for which version needs to be replaced.

© 2025 Microchip Technology Inc. and its subsidiaries

update_component_version (continued)		
Parameter	**Type**	**Description**
new_version	string	New version of a component that will be used as a replacement. This argument supports regexp syntax. If the specified version is 2.1.*, latest version that starts with 2.1 will be used as a replacement.
download_core	flag	This argument specifies whether core can be searched on the remote repositories or not. This is an optional argument. If it is set and appropriate version of the core is not found in the local vault, it will be searched and downloaded (if any) from the remote repository.

Error Codes

Error Code	Type	Description
None	Error	Update version failed. The specified core version \'%sCore\' v\%sVersion\ is neither available in your vault, nor available remotely for download in any of the repositories.
None	Error	Component \'%sComponent\' doesn't exist in the project.
None	Info	Component \'%sComponent\' was successfully updated to \'%sCore\' core version \'%sVersion\'.
None	Info	Version of the \'%sCore\' core of the \'%sComponent\' component is already \'%sVersion\'.
None	Info	Component \'%sComponent\' is already using core \'%sCore\' v\%sVersion\, which is the latest accessible version of the core that matches '\%sWildCard\' expression.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
update_component_version -component_name {COREJTAGDEBUG_C1} -new_version {3.1.100}
```

```
update_component_version -component_name {COREJTAGDEBUG_C0} -new_version {4.*} -download_core
```

3.100. use_source_file (Ask a Question)

Description

This Tcl command defines a module for your project.

```
use_source_file \
-file {full pathname} \
-module value
```

Arguments

Parameter	Type	Description
file	string	Specifies the Verilog or VHDL file. Value is the name of the file you wish use (including the full pathname). This is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

use_source_file (continued)		
Parameter	**Type**	**Description**
module	string	Specifies the module in which you want to use the file. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Required parameter 'module' is missing.
None	'file1.vhd' does not define module 'top'.
None	'/prj/hdl/file1.v' is not in the project.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Specify file1.vhd in the `./project/hdl` directory, in the module named top.

```
use_source_file -file "./project/hdl/file1.vhd" -module "top"
```

3.101. update_and_run_tool (Ask a Question)

Description

The Tcl command updates and runs the specified tool if there are any modifications with respect to the source files required by the tool. For tools that support command files, an optional command file can be supplied through the `-script` parameter.

```
update_and_run_tool -name {<tool_name >} \
-script {<absolute or relative path to script file>}
```

Tip: `-script` is an optional parameter.

© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specified tool name. The following is a list of supported tool names: • SYNTHESIZE • COMPILE • SIM_PRESYNTH • SIM_POSTSYNTH • SIM_POSTLAYOUT • PLACEROUTE • VERIFYTIMING • VERIFYPOWER • GENERATEPROGRAMMINGFILE • GENERATE_MEMORY_MAP • PROGRAMDEVICE • CONFIGURE_CHAIN • SSNANALYZER • EXPORTNETLIST • EXPORTSDF • GENERATEPROGRAMMINGDATA • GENERATEDEBUGDATA • GENERATE_SPI_FLASH_IMAGE • PROGRAM_SPI_FLASH_IMAGE
script	string	Specifies absolute or relative path of the script file. This is an optional parameter.

Supported tool_names

The following table lists the supported tool names for `update_and_run_tool -name {tool_name}`.

tool_name	Parameter	Description
SYNTHESIZE	-script {script_file}	Runs synthesis on your design.
COMPILE	N/A	Runs Compile with default or configured settings.
SIM_PRESYNTH	N/A	Runs pre-synthesis simulation with your default simulation tool.
SIM_POSTSYNTH	N/A	Runs post-synthesis simulation with your default simulation tool.
SIM_POSTLAYOUT	N/A	Runs the post layout simulation on the simulation tool.
PLACEROUTE	N/A	Runs Place and Route tool with default or configured settings.
VERIFYTIMING	-script {script_file}	Runs timing analysis with default settings/configured settings in script_file.
VERIFYPOWER	-script {script_file}	Runs power analysis with default settings/configured settings in script_file.
GENERATEPROGRAMMINGFILE	N/A	Generates the bit stream used for programming within Libero. This is similar to executing "Generate FPGA Array Data" in the Libero design flow.
GENERATE_MEMORY_MAP (SmartFusion 2, IGLOO 2 and RTG4 only)	N/A	Exports an XML file in <prj_folder> component/work/<design> / <design>_DataSheet.xml. The file contains information about the root SmartDesign in your project.
PROGRAMDEVICE	N/A	Programs your device with configured parameters.

© 2025 Microchip Technology Inc. and its subsidiaries

update_and_run_tool (continued)

tool_name	Parameter	Description
CONFIGURE_CHAIN	-script {script_file}	Takes a script that contains FlashPro-specific Tcl commands and passes them to FlashPro Express for execution.
SSNANALYZER	-script {script_file}	Takes a script that contains the Simultaneous Switching Noise (SSN)-specific Tcl commands and passes them to the SSN tool for execution. The SSN is a Libero SoC tool that analyzes and generates a Noise Margin report for I/Os after layout depending on Die ane Package.
EXPORTNETLIST	N/A	This command exports a .v/.vhd netlist file to the active synthesis implementation folder.
EXPORTSDF	N/A	This command exports the back annotated files to the designer/impl1 folder.
GENERATEPROGRAMMINGDATA	N/A	Generates the files needed by generating programming bit stream files.
GENERATEDEBUGDATA (PolarFire only)	N/A	Generates the files needed by SmartDebug during device debug.
GENERATE_SPI_FLASH_IMAGE (PolarFire only)	N/A	Generates SPI Flash Image file used for programming SPI FLASH Image within Libero.
PROGRAM_SPI_FLASH_IMAGE (PolarFire only)	N/A	Programs SPI Flash Image with configured parameters.

Error Codes

Error Code	Description
None	You must specify the Tcl script to run with the CONFIGURE_CHAIN tool.
None	You must specify the Tcl script to run with the SMARTDEBUG tool.
None	PROGRAMDEVICE: No programmer is connected.
None	SPI Flash Memory is not configured. Use the Configure Design Initialization Data and Memories tool to configure it.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
update_and_run_tool \
-name {COMPILE}
update_and_run_tool \
-name {VERIFYTIMING} \
-script {./SmartTime.tcl}
# Script file contains SmartTime-specific Tcl commands
update_and_run_tool \
-name {VERIFYPOWER} \
-script {./SmartPower.tcl}
# Script file contains SmartPower-specific Tcl commands
update_and_run_tool \
-name {SSNANALYZER}
-script {<full_path>/ssn.tcl}
# Script file contains the SSN-specific Tcl commands
```


© 2025 Microchip Technology Inc. and its subsidiaries

Important: Where possible, the value of `tool_name` corresponds to the name of the tool in Libero SoC.
Invoking some tools will cause Libero SoC to automatically run some upstream tools in the design flow. For example, invoking Place and Route will invoke Synthesis (if not already run) before it runs Place and Route.

© 2025 Microchip Technology Inc. and its subsidiaries

4. SmartDesign Tcl Commands (Ask a Question)

The SmartDesign Tcl commands can be used to create a design in the SmartDesign. You must either create or open a SmartDesign before you can use any of the SmartDesign commands - sd_* .

All SmartDesign Tcl commands are supported by the PolarFire family.

4.1. create_smartdesign_testbench (Ask a Question)

Description

This command is used to create an empty SmartDesign testbench.

```
create_smartdesign_testbench
-sd_name{smartdesign_testbench_component_name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign testbench component. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

4.2. sd_add_pins_to_group (Ask a Question)

Description

This tcl command adds one or more pins to a pin group on an instance in a SmartDesign component.

```
sd_add_pins_to_group -sd_name {smartdesign component name} \
                     -instance_name {instance name} \
                     -group_name {group name} \
                     -pin_names {pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance on which the pin group is present. It is mandatory.
group_name	string	Specifies the name of the group to add the pins to. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_add_pins_to_group (continued)		
Parameter	**Type**	**Description**
pin_names	string	Specifies the list of instance pins to be added to the pin group. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_add_pins_to_group -sd_name "sd_name" -instance_name "instance_name" [-group_name "group_name"] [-pin_names "[pin_names]+"]'.
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example adds "ARESETN" and "CLK" pins to the Group of "COREAX_C0_0" instance in the "top" design:

```
sd_add_pins_to_group -sd_name {top} \
                     -instance_name {COREAX_C0_0} \
                     -group_name {Group} \
                     -pin_names {ARESETN ACLK}
```

See Also

- sd_create_pin_group
- sd_remove_pins_from_group

4.3. sd_apply_presentation (Ask a Question)

Description

This tcl command allows you to change the presentation view. It takes the SmartDesign name and file path as a parameter containing all the presentation data.

```
-sd_name {SmartDesign_component_name}
-pm_model_file {Presentation_data_path}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign on which the presentation will be changed. It is mandatory.
pm_model_file	string	Specifies the file path which contains the presentation data. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Error: {Path} file does not exist.
None	Error: {Path} file does not have read access
None	Required parameter 'sd_name' is missing.
None	The component 'SmartDesign_component_name' does not exist.
None	Required parameter 'pm_model_file' is missing.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example changes the presentation view of a SmartDesign.

```
sd_apply_presentation -sd_name {sd} -pm_model_file {/home/user/sd.pm}
```

4.4. sd_clear_pin_attributes (Ask a Question)

Description

This tcl command clears all attributes on one or more pins/ports in a SmartDesign. Pin attributes include pin inversion, mark as unused and constant value settings.

Important: This command will not work on multiple pins in release v2021.1. Support for multiple pins will be provided in the next Libero release.

```
sd_clear_pin_attributes -sd_name {smartdesign component name} \ -pin_names {port or pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_names	string	Specifies the name of the port/pin for which all the attributes must be cleared. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'sd_name' has illegal value.
None	Parameter 'pin_names' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_clear_pin_attributes (continued)	
Error Code	**Description**
None	Required parameter 'pin_names' is missing.
None	Required parameter 'sd_name' is missing.
None	The component 'design_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_clear_pin_attributes -sd_name "sd_name" -pin_names "pin_names"'.
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example clears all attributes of the pin {RAM1K18_0:A_DOUT_CLK} in the {sd1} SmartDesign component.

```
sd_clear_pin_attributes -sd_name {sd1} -pin_names {RAM1K18_0:A_DOUT_CLK}
```

This example clears all attributes of the pin {CARRY_OUT} in the {top} SmartDesign component.

```
sd_clear_pin_attributes -sd_name {top} -pin_names {CARRY_OUT}
```

See Also

- sd_mark_pins_unused

4.5. sd_configure_core_instance (Ask a Question)

Description

This tcl command configures the parameters of a core instance (Direct Instantiation) in a SmartDesign component. This command can configure multiple core parameters at a time.

```
sd_configure_core_instance -sd_name {smartdesign component name} \
                           -instance_name {core instance name} \
                           -params {core parameters}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the core instance in the SmartDesign which needs to be configured. It is mandatory.
params	string	Specifies the parameters that need to be configured for the core instance. It is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'instance_name' is missing.
None	Parameter 'instance_name' has illegal value.
None	The component 'comonent_name' doesn't exist.
None	Parameter 'params' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_configure_core_instance -sd_name "sd_name" -instance_name "instance_name" [-params "[params]+"] .

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example configures {COREFIFO_0} instance in the {SD1} SmartDesign component with the following parameters: "SYNC:0", "param2:value2" and "param3:value3".

```
sd_configure_core_instance \
              -sd_name {SD1} \
              -instance_name {COREFIFO_0} \
              -params {"SYNC:0" "param2:value2" "param3:value3"}
```

4.6. sd_connect_instance_pins_to_ports (Ask a Question)

Description

This tcl command connects all pins of an instance to new SmartDesign top level ports.

```
sd_connect_instance_pins_to_ports -sd_name {smartdesign component name} \
                                  -instance_name {instance name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the instance name for which all the pins must be connected to top level ports. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_connect_instance_pins_to_ports (continued)	
Error Code	**Description**
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_instance_pins_to_ports -sd_name "sd_name" -instance_name "instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example connects pins to ports of the instance {CORESPI_C0_0}.

```
sd_connect_instance_pins_to_ports -sd_name {top} \
                                  -instance_name {CORESPI_C0_0}
```

4.7. sd_connect_net_to_pins (Ask a Question)

Description

This tcl command connects a list of SmartDesign top level ports and/or instance pins to a net.

```
sd_connect_net_to_pins -sd_name {smartdesign component name} \
                       -net_name {net name} \
                       -pin_names {port or pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
net_name	string	Specifies the name of the net to be connected to pins/ports in the SmartDesign component. It is mandatory.
pin_names	list of string	Specifies the name of the ports/pins to be connected to the net in the SmartDesign. It is mandatory. The command will fail if: • The ports/pins do not exist. • The ports/pins and the net being connected are of different range/ size. • There is more than one port/pin driving the net.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'sd_name' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_connect_net_to_pins (continued)	
Error Code	**Description**
None	Required parameter 'net_name' is missing.
None	Parameter 'net_name' has illegal value.
None	Required parameter 'pin_names' is missing.
SDCTRL05	Pin 'pin_name' does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_net_to_pins -sd_name "sd_name" -net_name "net_name" -pin_names "[pin_names]+" '.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example connects "clk_net" net to "CLK RAM64x12_0:R_CLK" and "RAM64x12_0:W_CLK" pins:

```
sd_connect_net_to_pins \
             -sd_name {top} \
             -net_name {clk_net} \
             -pin_names {"CLK RAM64x12_0:R_CLK"  "RAM64x12_0:W_CLK"}
```

4.8. sd_connect_pins_to_constant (Ask a Question)

Description

This tcl command connects SmartDesign top level output ports or input instance pins to constant values.

Important: This command will not work on multiple pins in release v2021.1. Support for multiple pins will be provided in the next Libero release.

```
sd_connect_pins_to_constant -sd_name {smartdesign component name} \ -pin_names {port or pin names} \ -value {constant value}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_names	string	Specifies the names of the top level output ports or the instance level input pins to be tied to constant values. It is mandatory.
value	string	Specifies the constant value to be assigned to the port/pin. It is mandatory. The acceptable values to this argument are GND/VCC/ hexadecimal numbers.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'value' has illegal value.
None	Required parameter 'value' is missing.
None	Parameter 'pin_names' has illegal value.
None	Required parameter 'pin_names' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	The component 'design_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_pins_to_constant -sd_name "sd_name" -pin_names "pin_names" -value "value"'.
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example connects "bypass" pin to "GND".

```
sd_connect_pins_to_constant -sd_name {top} \ -pin_names {bypass} \ -value {GND}
```

4.9. sd_connect_pins (Ask a Question)

Description

This Tcl command connects a list of SmartDesign top level ports and/or instance pins together.

```
sd_connect_pins -sd_name {smartdesign component name} \
                -pin_names {port or pin or slice names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_names	string	Specifies the port names, pin names and/or slice names to be connected together. It is mandatory. This command will fail if the ports, pins or slices do not exist. This command will also fail if the ports, pins and/or slices are not of the same size/range.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Cannot connect top-level input port 'port_name' to top-level input port 'port_name' since they are both drivers.
None	Required parameter 'pin_names' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_connect_pins (continued)	
Error Code	**Description**
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_pins -sd_name "sd_name" -pin_names "[pin_names]+"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

- The following example connects "CLK" port to "CLK" pins of "MACC_PA_0:CLK" and "DFN1_0:CLK" instances:

```
sd_connect_pins -sd_name {top} \
                -pin_names {CLK MACC_PA_0:CLK DFN1_0:CLK}
```

- The following example connects "MACC_PA_0:A" pin to "RAM1K20_0:A_DIN[17:0]":

```
sd_connect_pins -sd_name {top} \
                -pin_names {MACC_PA_0:A RAM1K20_0:A_DIN[17:0]}
```

See Also

- sd_create_scalar_port
- sd_connect_pin_to_port

4.10. sd_connect_pin_to_port (Ask a Question)

Description

This Tcl command connects a SmartDesign instance pin to a new top level port. This command is equivalent to the 'Promote to Top Level' GUI action on an instance pin.

Note: This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_connect_pin_to_port -sd_name {smartdesign component name} \
                       -pin_name {pin name} \
                       [-port_name {port name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_name	string	Specifies the name of the instance level pin that needs to be connected to a top level port. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_connect_pin_to_port (continued)		
Parameter	**Type**	**Description**
port_name	string	Specifies the name of the new top level port that the instance pin will be connected to. It is optional. If the port name is not specified, the new port takes the name of the instance pin. If the port name as defined by these rules already exists, the tool automatically creates a new port with name <port_name>_<index> (index is an automatically generated integer starting at 0 such that the port name is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Pin 'instance_name:pin_name' does not exist.
None	Required parameter 'sd_name' is missing.
None	You must specify a pin name.
None	Cannot promote pin 'pin_name' on 'instance_name' to top because it is already connected to a top level port.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_pin_to_port -sd_name "sd_name" [-pin_name "pin_name"] [-port_name "port_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

- The following command connects "D" input pin of "DFN1_0" instance to "D" input port which was created with the name of port_name and automatically generated index(unique):

```
sd_connect_pin_to_port -sd_name {top} -pin_name {DFN1_0:D}
```

- The following command connects "DFN1_0" instance "Q" output pin to "Q_OUT" port:

```
sd_connect_pin_to_port -sd_name {top} \
                       -pin_name {DFN1_0:Q} \
                       -port_name {Q_OUT}
```

See Also

- sd_connect_pin

4.11. sd_copy_paste (Ask a Question)

Description

This Tcl command used to copies and pastes ports, nets and instances within or cross SmartDesign.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: The "net" can be copied and pasted if at least two connections are specified. This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the 'Export Component Description(Tcl)' but will present in Libero Project 'Export Script File'. You must specify at least one valid instance or port.

```
sd_copy_paste -sd_name {SmartDesign component name} \
              -source_sd_name {source SmartDesign name} \
              [-instance_names {instance names}] \
              [-port_names {port names}] \
              [-net_names {net names}]
```

Arguments

Parameter	Type	Description
sd_name	string	Name of the SmartDesign where the copied data need to be pasted. This parameter is mandatory.
source_sd_name	string	Name of the SmartDesign from where object needs to be copied. This parameter is mandatory.
instance_names	list of strings	Names of the instances from "source_SmartDesign_name" SmartDesign that need to be copied to "SmartDesign_component_name". This parameter is optional.
port_names	list of strings	Names of the ports from "source_SmartDesign_name" SmartDesign that need to be copied to "SmartDesign_component_name". This parameter is optional.
net_names	list of strings	Names of the nets from "source_SmartDesign_name" SmartDesign that need to be copied to "SmartDesign_component_name". This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'source_sd_name' is missing.
None	The paste command failed for net 'net_name' in SmartDesign 'sd'. The net could not be found in the source SmartDesign.
None	Parameter 'net_names' has illegal value.
None	The paste command failed for port 'port_name' in SmartDesign 'sd'. The port could not be found in the source SmartDesign.
None	The paste command failed for instance 'instance_name' in SmartDesign 'sd'. The instance could not be found in the source SmartDesign.
None	Parameter 'instance_names' has illegal value.
None	The paste command failed. You must specify at least one valid instance or port.
None	Parameter 'source_sd_name' has illegal value.
None	Parameter 'pram_name' is not defined. Valid command formatting is 'sd_copy_paste -sd_name "sd_name" -source_sd_name "source_sd_name" [-instance_names "[instance_names]+"] [-port_names "[port_names]+"] [-net_names "[net_names]+"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

sd_copy_paste (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example copies "AND2_0" instance, "A", "B", "Y" ports with their appropriate nets from "sd1" and pastes into "sd2":

```
sd_copy_paste -sd_name {sd2} -source_sd_name {sd1} -instance_names {"AND2_0"} \
         -port_names {"A" "B" "Y"} -net_names {"A" "B" "Y"}
```

The following example copies only "NAND2_0" instance from "sd1" and pastes in "sd2":

```
sd_copy_paste -sd_name {sd2} -source_sd_name {sd1} -instance_names {"NAND2_0"}
```

The following example copies only "A", "B", "Y" ports from "sd1" and pastes in "sd2":

```
sd_copy_paste -sd_name {sd2} -source_sd_name {sd1} -port_names {"A" "B" "Y"}
```

See Also

- sd_cut_paste

4.12. sd_create_bif_net (Ask a Question)

Description

This tcl command creates a bus interface (BIF) net in a SmartDesign component. Any net created must be connected to two or more ports/pins using the command “sd_connect_net_to_pins”.

Note: This command is not required to build a SmartDesign component. It is not exported when you select Libero Project - `Export Script File’ or ‘Export Component Description(Tcl)’ on a SmartDesign component. This command is used to manually create a Tcl script and specify a new name to the net that connects two or more ports/pins.

```
sd_create_bif_net -sd_name {smartdesign component name} \
                  -net_name {net name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
net_name	string	Specifies the name of the net to be added in the SmartDesign component. It is mandatory. The command will fail if there is an already existing net with the same name.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'net_name' has illegal value.
None	Required parameter 'net_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_create_bif_net (continued)	
Error Code	**Description**
None	Cannot add net "bus_net" because a net with the same name already exists.
None	Cannot connect the two bus interface pins 'BIF_1' and 'instance:BIF_1' because they are not compatible.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_bif_net -sd_name "sd_name" -net_name "net_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This command creates BIF net named "bifnet1" in the "TOP" SmartDesign component.

```
sd_create_bif_net -sd_name {TOP} -net_name {bifnet1}
```

See Also

- sd_connect_net_to_pins

4.13. sd_create_bif_port (Ask a Question)

Description

This Tcl command creates a SmartDesign Bus Interface port of a given type. This command is used to create top level Bus Interface ports in a SmartDesign component to connect to the instance level Bus Interface ports of the same type.

```
sd_create_bif_port -sd_name {smartdesign component name} \
                   -port_name {port name} \
                   -port_bif_vlnv {vendor:library:name:version} \
                   -port_bif_role {port bif role} \
                   -port_bif_mapping {[bif port name:port name]+}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
port_name	string	Specifies the name of the Bus Interface port to be added in the SmartDesign. It is mandatory.
port_bif_vlnv	string	Specifies the version identifier of the Bus Interface port to be added in the SmartDesign. It is mandatory.
port_bif_role	string	Specifies the role of the Bus Interface port to be added in the SmartDesign.
port_bif_mapping	string	Specifies the mapping between the bus interface formal names and the SmartDesign ports mapped onto that bus interface port. It is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	The component 'design_name' does not exist.
None	Required parameter 'port_name' is missing.
None	Parameter 'port_name' has illegal value.
None	Required parameter 'port_bif_vlnv' is missing.
None	Parameter 'port_bif_vlnv' has illegal value.
None	Parameter 'port_bif_role' has illegal value.
None	Required parameter 'port_bif_role' is missing.
None	Parameter 'port_bif_mapping' has illegal value.
None	Required parameter 'port_bif_mapping' is missing.
SDCTRLO2	Port name 'bif_name' already exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_bif_port -sd_name "sd_name" -port_name "port_name" -port_bif_vlnv "port_bif_vlnv" -port_bif_role "port_bif_role" -port_bif_mapping "[port_bif_mapping]+"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example creates bif port with "BIF_1" name, "AMBA:AMBA2:APB:r0p0" vendor:library:name:version, {"PADDR:PADDR" \"PENABLE:PENABLE" \ "PWRITE:PWRITE" \ "PRDATA:PRDATA" \ "PWDATA:PWDATA" \ "PREADY:PREADY" } bif mapping and "slave" role:

```
sd_create_bif_port -sd_name {sd1} \
                   -port_name {BIF_1} \
                   -port_bif_vlnv {AMBA:AMBA2:APB:r0p0} \
                   -port_bif_role {slave} \
                   -port_bif_mapping {"PADDR:PADDR"
\
                                      "PENABLE:PENABLE" \
                                      "PWRITE:PWRITE" \
                                      "PRDATA:PRDATA" \
                                      "PWDATA:PWDATA" \
                                      "PREADY:PREADY" }
```

See Also

- hdl_core_add_bif
- hdl_core_assign_bif_signal
- sd_create_bif_net

© 2025 Microchip Technology Inc. and its subsidiaries

4.14. sd_create_bus_net (Ask a Question)

Description

This tcl command creates a bus net of a given range in a SmartDesign component. Any net created must be connected to two or more ports/pins using the command “sd_connect_net_to_pins”.

Note: This command is not required to build a SmartDesign component. It is not exported when you select Libero Project - 'Export Script File' or 'Export Component Description(Tcl)' on a SmartDesign component. This command is used to manually create a Tcl script and specify a new name to the net that connects two or more ports/pins.

```
sd_create_bus_net -sd_name {smartdesign component name} \
                  -net_name {net name} \
                  -net_range [left index range:right index range]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
net_name	string	Specifies the name of the net to be added in the SmartDesign component. It is mandatory.
net_range	string	Specifies the range of the net added to the SmartDesign component. The range is defined by its left and right rangeindices. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'net_range' has illegal value.
None	Required parameter 'net_range' is missing.
None	Parameter 'net_name' has illegal value.
None	Required parameter 'net_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_bus_net -sd_name "sd_name" -net_name "net_name" -net_range "net_range"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example creates bus net in range [5:0] named "ab1" and connects it to the "RAM64x12_0:R_ADDR" and "a" pins.

© 2025 Microchip Technology Inc. and its subsidiaries

This new net is visible in the UI only when it is connected to two or more ports/pins using the command“sd_connect_net_to_pins” as shown below.

```
sd_create_bus_net -sd_name {top} \
                  -net_name {ab1} \
                  -net_range {[5:0]]}
sd_connect_net_to_pins -sd_name {top} \
                       -net_name {ab1} \
                       -pin_names {a RAM64x12_0:R_ADDR}
```

See Also

- sd_connect_net_to_pins
- sd_create_bus_port

4.15. sd_create_bus_port (Ask a Question)

Description

This tcl command creates a bus port of a given range in a SmartDesign component.

```
sd_create_bus_port -sd_name {smartdesign component name} \
                   -port_name {port name} \
                   -port_direction {IN|OUT|INOUT} \
                   -port_range {[left range index:right range index]}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
port_name	string	Specifies the name of the bus port added to be SmartDesign component. It is mandatory.
port_direction	string	Specifies the direction of the bus port added to the SmartDesign component. It is mandatory.
port_range	string	Specifies the range of the bus port added to the SmartDesign component. The range is defined by the left and rightindices. It is mandatory. The range must be specified inside the square brackets.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'port_range' has illegal value.
None	Required parameter 'port_range' is missing.
None	Parameter 'port_direction' has illegal value.
None	Required parameter 'port_direction' is missing.
None	Parameter 'port_name' has illegal value.
None	Required parameter 'port_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_bus_port -sd_name "sd_name" -port_name "port_name" -port_direction "IN | OUT | INOUT" -port_range "port_range"'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example creates bus port with name "test_port13", direction "OUT" and in [9:36] range.

```
sd_create_bus_port -sd_name {top} \
                   -port_name {test_port13} \
                   -port_direction {OUT} \
                   -port_range {[9:36]}
```

```
sd_create_bus_port -sd_name {top} \
                   -port_name {test_port4} \
                   -port_direction {IN} \
                   -port_range {[31:0]}
```

See Also

- sd_create_bus_net

4.16. sd_create_pin_group (Ask a Question)

Description

This tcl command creates a group of pins in a SmartDesign component. A pin group is only used to manage the complexity of the SmartDesign canvas. There is no actual netlist functionality related to pin group commands. Pin groups cannot be created for top level ports.

```
sd_create_pin_group -sd_name {smartdesign component name} \
                    -instance_name {instance name} \
            [-group_name {group name}] \
            [-pin_names {pin to be added to the group}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance on which the pin group is added. It is mandatory.
group_name	string	Specifies the name of the pin group. It is optional. If the group name is not specified, the default name will be 'Group'. If the name 'Group' is already taken, then the group name will be 'Group_<index>' (index is auto-incremented).
pin_names	list of strings	Specifies the list of instance pins to be added to the pin group(example below). It is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Unconnected input pin "instace_name":"pin_name".

© 2025 Microchip Technology Inc. and its subsidiaries

sd_create_pin_group (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_pin_group -sd_name "sd_name" [-group_name "group_name"] -instance_name "instance_name" [-pin_names "[pin_names]+"]'.
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example creates a pin group named "MyGroup" on the "COREAXINTERCONNECT_C0_0" instance in the "TOP" design:

```
sd_create_pin_group -sd_name {TOP} \
                    -instance_name {COREAXINTERCONNECT_C0_0} \
                    -group_name {MyGroup} \
                    -pin_names {ACLK ARESETN}
```

The following example creates a group of "FORCE_DISP", "DISP_SEL" and "WA_RSTn" pins in a "TOP" design named "MyGroup" on the "CorePCS_C0_0" instance:

```
sd_create_pin_group -sd_name {TOP} \
                    -group_name {MyGroup} \
                    -instance_name {CorePCS_C0_0} \
                    -pin_names {"FORCE_DISP" "DISP_SEL" "WA_RSTn"}
```

See Also

- sd_add_pins_to_group
- sd_delete_pin_group
- sd_rename_pin_group
- sd_remove_pins_from_group

4.17. sd_create_pin_slices (Ask a Question)

Description

This tcl command creates slices for a SmartDesign top level bus port or an instance level bus pin.

```
sd_create_pin_slices -sd_name {smartdesign component name} \
                     -pin_name {port or pin _name} \
                     -pin_slices {port or pin slices}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_name	string	Specifies the name of the bus port or bus pin to be sliced. It is mandatory. This command will fail if the port/pin isscalar or if the bus port/pin does not exist.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_create_pin_slices (continued)		
Parameter	**Type**	**Description**
pin_slices	string	Specifies the port/pin slices as a list of bus ranges which must be contained within the port/pin bus range. It ismandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'pin_slices' has illegal value.
None	Required parameter 'pin_slices' is missing.
None	Parameter 'pin_name' has illegal value.
None	Required parameter 'pin_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_pin_slices -sd_name "sd_name" -pin_name "pin_name" -pin_slices "[Ranges of pin slices]+"'.
SDCTRL06	Cannot connect output pin 'pin1' of 'instance' to output pin 'pin2' of 'sdesign' since they are both drivers.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example creates slices for "Rdata" pin.

```
sd_create_pin_slices -sd_name {sub} \
                     -pin_name {Rdata} \
                     -pin_slices {[4:3] [2:0]}
```

4.18. sd_create_scalar_net (Ask a Question)

Description

This Tcl command creates a scalar net in a SmartDesign component. Any net created must be connected to two or more ports/pins using the command "`sd_connect_net_to_pins`".

Note: This command is not required to build a SmartDesign component. It is not exported when user selects 'Export Component Description(Tcl)' on a SmartDesign component. This command is used to manually create a Tcl script and specify a new name to the net that connects two or more ports/pins.

```
sd_create_scalar_net -sd_name {smartdesign component name} \
                     -net_name {net name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
net_name	string	Specifies the name of the net added to the SmartDesign component. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'net_name' is missing.
None	Required parameter 'pin_names' is missing.
None	Required parameter 'sd_name' is missing.
None	Unconnected input pin instance_name:pin_name
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_connect_net_to_pins -sd_name "sd_name" -net_name "net_name" -pin_names "[pin_names]+"'.
None	Parameter 'net_name' has illegal value.
None	Net 'net_name' does not exist.
None	The component '' doesn't exist.
None	The component 'component_name' doesn't exist.
SDCTRL05	Pin 'pin_or_port_name' does not exist.
None	Cannot promote pin 'pin_name' on 'instance_name' to top because it is already connected to a top level port.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example creates a scalar net with "clk_net" name in a "Top" component and connects `CLKRAM64x12_0:R_CLK, RAM64x12_0:W_CLK` ports via `clk_net` net:

Note: This new net is visible in the UI only when it is connected to two or more ports/pins using the command sd_connect_net_to_pins shown below.

```
sd_create_scalar_net -sd_name {Top} -net_name {clk_net}
sd_connect_net_to_pins -sd_name {Top} -net_name {clk_net} \
                       -pin_names {CLKRAM64x12_0:R_CLK RAM64x12_0:W_CLK}
```

See Also

- sd_create_scalar_port

© 2025 Microchip Technology Inc. and its subsidiaries

4.19. sd_create_scalar_port (Ask a Question)

Description

This Tcl command creates a scalar port in a SmartDesign component.

```
sd_create_scalar_port -sd_name {smartdesign component name} \
                      -port_name {port name} \
                      -port_direction {IN|OUT|INOUT}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
port_name	string	Specifies the name of the port added to the SmartDesign component. It is mandatory.
port_direction	string	Specifies the direction of the port added to the SmartDesign component. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Invalid argument value: '' (expecting IN, OUT or INOUT).
None	Parameter 'port_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_create_scalar_port -sd_name "sd_name" -port_name "port_name" -port_direction "IN | OUT | INOUT"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following tcl command creates "P1" scalar input port in the "main" design:

```
sd_create_scalar_port -sd_name {main} \
                      -port_name {P1} \
                      -port_direction {IN}
```

See Also

- sd_create_scalar_net

4.20. sd_cut_paste (Ask a Question)

Description

This Tcl cuts and pastes ports and instances from one SmartDesign to another one.

© 2025 Microchip Technology Inc. and its subsidiaries

Important: This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the 'Export Component Description(Tcl)' but will present in Libero Project 'Export Script File'. You must specify at least one valid instance or port.

```
sd_cut_paste -sd_name {SmartDesign component name} \
             -source_sd_name {source SmartDesign name} \
             [-instance_names {instance names}] \
             [-port_names {port names}]
```

Arguments

Parameter	Type	Description
sd_name	string	Name of the SmartDesign where the copied data need to be pasted. This parameter is mandatory.
source_sd_name	list of strings	Name of the SmartDesign from where object needs to be cut. This parameter is mandatory.
instance_names	list of strings	Names of the instances from "source_SmartDesign_name" SmartDesign that need to be cut to "SmartDesign_component_name".
port_names	list of strings	Names of the ports from "source_SmartDesign_name" SmartDesign that need to be cut to "SmartDesign_component_name". This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'source_sd_name' is missing.
None	The paste command failed for port 'port_name' in SmartDesign 'sd`'. The port could not be found in the source SmartDesign.
None	Parameter 'port_names' has illegal value.
None	The paste command failed for instance 'instance_name' in SmartDesign 'sd'. The instance could not be found in the source SmartDesign.
None	Parameter 'instance_names' has illegal value.
None	The paste command failed. You must specify at least one valid instance or port.
None	Parameter 'source_sd_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_cut_paste -sd_name "sd_name" -source_sd_name "source_sd_name" [-instance_names "[instance_names]+"] [-port_names "[port_names]+"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example cuts "AND2_0" instance, "A", "B", "Y" ports from "sd1" and pastes into "sd2":

```
sd_cut_paste -sd_name {sd2} -source_sd_name {sd1} -instance_names {"AND2_0"} -port_names {"A"
"B" "Y"}
```

The following example cuts "NAND2_0" instance from "sd1" and pastes in "sd2":

```
sd_cut_paste -sd_name {sd2} -source_sd_name {sd1} -instance_names {"NAND2_0"}
```

The following example cuts only "A", "B", "Y" ports from "sd1" and pastes in "sd2":

```
sd_cut_paste -sd_name {sd2} -source_sd_name {sd1} -port_names {"A" "B" "Y"}
```

See Also

- sd_copy_paste

4.21. sd_delete_bif_pin (Ask a Question)

Description

This tcl command deletes one or more BIF pins from the SmartDesign component.

Important: This command will not delete multiple bifs in this release. Support for deleting multiple pin a will be provided in the next Libero release. It is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_delete_bif_pin -sd_name {SmartDesign name} \
                  [-library "library"] \
                  -pin_name {pin name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory. It is mandatory.
library	string	Specifies the name of the library the component belongs to. It is optional.
pin_name	string	Specifies the name of the BIF port to be deleted. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	The component 'design_name' doesn't exist.
None	Required parameter 'pin_name' is missing.
None	Parameter 'pin_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_delete_bif_pin -sd_name "sd_name" -pin_name "pin_name"'.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_delete_bif_pin (continued)	
Error Code	**Description**
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families	Supported Versions
PolarFire®	v12.4+
SmartFusion® 2	v12.4+
RTG4™	v12.4+
IGLOO® 2	v12.4+
PolarFire SoC	v12.6+

Example

This examples deleted already create "BIF_1" port from the 'Top' design:

```
sd_delete_bif_pin -sd_name {sd1} -pin_name {BIF_1}
```

See Also

- hdl_core_add_bif
- hdl_core_assign_bif_signal
- sd_create_bif_port

4.22. sd_delete_instances (Ask a Question)

Description

This tcl command deletes one or more instances from a SmartDesign component.

Note: This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_delete_instances -sd_name {smartdesign component name} \
                    -instance_names {instance names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_names	string	Specifies the instance names to be deleted. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_names' has illegal value.
None	Required parameter 'instance_names' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_delete_instances (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_delete_instances -sd_name "sd_name" -instance_names "[instance_names]+"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example deletes "RAM64X12_0" instance.

```
sd_delete_instances -sd_name {top} \
                    -instance_names {RAM64X12_0}
```

See Also

- sd_disconnect_instance
- sd_duplicate_instance

4.23. sd_delete_nets (Ask a Question)

Description

This tcl command deletes one or more nets from the SmartDesign component.

Important: This command will not delete multiple nets in this release. Support for deleting multiple nets will be provided in the next Libero release. This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_delete_nets -sd_name {smartdesign component name} \
               -net_names {net names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
net_names	string	Specifies the net names to be deleted. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	The component 'design_name' doesn't exist.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_delete_nets (continued)	
Error Code	**Description**
None	Net 'net_name' does not exist.
None	Parameter 'net_names' has illegal value.
None	Required parameter 'net_names' is missing.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example deletes "B_REN_0" net from "top" design.

```
sd_delete_nets -sd_name {top} -net_names {B_REN_0}
```

See Also

- sd_create_scalar_net
- sd_create_bif_net

4.24. sd_delete_pin_group (Ask a Question)

Description

This tcl command deletes a pin group from an instance in a SmartDesign component.

```
sd_delete_pin_group -sd_name {smartdesign component name} \
                    -instance_name {instance name} \
                    -group_name {group name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance from which the group pin needs to be deleted. It is mandatory.
group_name	string	Specifies the name of the pin group to be deleted. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.
None	Required parameter 'group_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_delete_pin_group -sd_name "sd_name" -group_name "group_name" -instance_name "instance_name"'.
SDCTRL05	Pin 'group_name' does not exist.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example deletes the "MyGroup" group on the "COREAXINTERCONNECT_C0_0" instance in the "TOP" design:

```
sd_delete_pin_group -sd_name {TOP} \
                    -instance_name {COREAXINTERCONNECT_C0_0} \
                    -group_name {MyGroup}
```

See Also

- sd_create_pin_group
- sd_add_pins_to_group
- sd_rename_pin_group
- sd_remove_pins_from_group

4.25. sd_delete_pin_slices (Ask a Question)

Description

This tcl command deletes SmartDesign top level port slices or instance pin slices.

Note: This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_delete_pin_slices -sd_name {smartdesign component name} \
                     -pin_name {port or pin name} \
                     -pin_slices {port or pin slices}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_name	string	Specifies the name of the bus port or bus pin for which the slices must be deleted. It is mandatory.
pin_slices	string	Specifies the ranges of the port and/or pin slices to be deleted. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'pin_slices' has illegal value.
None	Required parameter 'pin_slices' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_delete_pin_slices (continued)	
Error Code	**Description**
None	Parameter 'pin_name' has illegal value.
None	Required parameter 'pin_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_delete_pin_slices -sd_name "sd_name" -pin_name "pin_name" -pin_slices "[Ranges of pin slices]+"'.
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example deletes {[17:16] [15:1] [0]} slices of "A" pin.

```
sd_delete_pin_slices -sd_name {top} \
                     -pin_name {A} \
                     -pin_slices {[17:16] [15:1] [0]}
```

See Also

- sd_create_pin_slices

4.26. sd_delete_ports (Ask a Question)

Description

This Tcl command deletes one or more ports from the SmartDesign component.

Note: This command will not work on multiple ports. if `-port_names` argument specified multiple times, then it takes the value of the last argument. This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_delete_ports -sd_name {smartdesign component name} \
                -port_names {port names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
port_names	string	Specifies the name of the port to be deleted. It is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Port 'port_name' doesn't exist.
None	Required parameter 'port_names' is missing.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_delete_ports -sd_name "sd_name" -port_names "port_names"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command deletes "REF_CLK_0" port from Top design:

```
sd_delete_ports -sd_name {Top} -port_names {REF_CLK_0}
```

4.27. sd_disconnect_instance (Ask a Question)

Description

This tcl command clears all the connections on an instance in a SmartDesign component.

Note: This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_disconnect_instance -sd_name {smartdesign component name} \
                       -instance_name {instance name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance for which all the connections must be cleared. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_disconnect_instance (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_disconnect_instance -sd_name "sd_name" -instance_name "[instance_name] +"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example disconnects {RAM1K18_1" instance.

```
sd_disconnect_instance -sd_name {sd1} -instance_name {RAM1K18_1}
```

See Also

- sd_rename_instance
- sd_update_instance
- sd_delete_instances
- sd_duplicate_instance

4.28. sd_disconnect_pins (Ask a Question)

Description

This Tcl command disconnects a list of SmartDesign top level ports and/or instance pins from the net they are connected to.

Note: This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported file when user selects Libero Project -'Export Component Description(Tcl)' on a SmartDesign component.

```
sd_disconnect_pins -sd_name {smartdesign component name} \
                   -pin_names {port or pin or slice names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_names	string	Specifies the port, pin and/or slice names to be disconnected as follows: {"instance_name:pin_name" "port_name"}. This command will fail if the ports, pins and/or slices do not exist. There may be multiple `-pin_names` arguments (see example below). This is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'pin_names' is missing or has invalid value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_disconnect_pins -sd_name "sd_name" -pin_names "[pin_names]+"'.
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

- The following command disconnects "B_ren" and "B_ADRR[12:12]" slice from the "SD1" design:

```
sd_disconnect_pins -sd_name {SD1} \
                   -pin_names {B_ren RAM1K20_0:B_ADRR[12:12]}
```

- The following command disconnects "AND2_0:B", "AND3_0:APF_XCVR_ERM_C0_0:LANE0_RX_READY" pin/ports from the "SD2" SmartDesign:

```
sd_disconnect_pins \
      -sd_name {SD2} \
      -pin_names {AND2_0:B AND3_0:APF_XCVR_ERM_C0_0:LANE0_RX_READY}
```

See Also

- sd_connect_pins

4.29. sd_duplicate_instance (Ask a Question)

Description

This tcl command creates a new instance in a SmartDesign with the same module/component as the original instance.

Note: This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_duplicate_instance
            -sd_name {smartdesign component name} \
            -instance_name {instance name} \
            [-duplicate_instance_name {duplicate instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance to be duplicated. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_duplicate_instance (continued)		
Parameter	**Type**	**Description**
duplicate_instance_name	string	Specifies the name of the duplicate instance. It is optional. If value is not specified the default name is provided.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_duplicate_instance -sd_name "sd_name" -instance_name "instance_name" [-duplicate_instance_name "duplicate_instance_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example duplicates "PF_CCC_C0_0" instance.

Note: Will create instance named PF_CCC_C0_1.

```
sd_duplicate_instance -sd_name {top} -instance_name {PF_CCC_C0_0}
```

This example dupicates {PF_CCC_C0_0} instance with {abc_1} name.

```
sd_duplicate_instance -sd_name {top} \
                      -instance_name {PF_CCC_C0_0} \
                      -duplicate_instance_name {abc_1}
```

See Also

- sd_disconnect_instance

4.30. sd_hide_bif_pins (Ask a Question)

Description

This tcl command hides one or more already exposed internal scalar or bus pins/ports of a Bus Interface pin/port.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: This command will not hide multiple pins/ports in this release. Support to hide multiple pins/ports will be provided in the next Libero release. This command is not required to build a SmartDesign component. This command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_hide_bif_pins -sd_name {smartdesign component name} \
                 -bif_pin_name {name of the bif pin or port} \
                 -pin_names {pins or ports to be exposed}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
bif_pin_name	string	Specifies the name of the Bus Interface pin for which the internal pins must be hidden. It is mandatory.
pin_names	string	Specifies the bus interface internal pin/port names to be hidden. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'pin_names' has illegal value.
None	Required parameter 'pin_names' is missing.
None	Parameter 'bif_pin_name' has illegal value.
None	Required parameter 'bif_pin_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_hide_bif_pins -sd_name "sd_name" -bif_pin_name "bif_pin_name" -pin_names "pin_names"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example hides "TX_PLL_LOCK_0" pin of "CLKS_FROM_TXPLL_0" bif pin.

```
sd_hide_bif_pins -sd_name {SD1} \
                 -bif_pin_name {CLKS_FROM_TXPLL_0} \
                 -pin_names{TX_PLL_LOCK_0}
```

See Also

- sd_show_bif_pins

© 2025 Microchip Technology Inc. and its subsidiaries

- sd_delete_bif_pin

4.31. sd_instantiate_component (Ask a Question)

Description

This Tcl command instantiates a Libero SmartDesign component or a core component into another SmartDesign component.

```
sd_instantiate_component -sd_name {smartdesign component name} \
                         -component_name {component module name} \
                         [-instance_name {instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component in which other components will be instantiated. It is mandatory.
component_name	string	Specifies the name of the component being instantiated in the SmartDesign component. It is mandatory. The components include SmartDesign components, core components created for different types of cores from the catalog and blocks.
instance_name	string	Specifies the instance name of the Libero component being instantiated in the SmartDesign component. It is optional. By default, the instance name is <component_module_name>_<index> (index is an automatically generated integer starting at 0 such that the instance name is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'component_name' has illegal value.
None	Parameter 'component_to_instantiate' is missing or has invalid value.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'sd_instantiate_component -sd_name "sd_name" [-component_name "component_name"] [-instance_name "instance_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example instantiates "sd1" component into "sd1_0."

```
sd_instantiate_component -sd_name {sub} \
                         -component_name {sd1} \
                         -instance_name {sd1_0}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Ths example instantiates "PF_CCC_C0" component.

```
sd_instantiate_component -sd_name {top} -component_name {PF_CCC_C0}
```

4.32. sd_instantiate_core (Ask a Question)

Description

This tcl command instantiates a core from the catalog directly into a SmartDesign component (Direct Instantiation)without first having to create a component for the core. The file-set related to the core is generated only when theSmartDesign in which the core is instantiated is generated. The GUI equivalent of this command is not currentlysupported in Libero. To instantiate a core in a SmartDesign component in the GUI, you have to first create acomponent for the core.

```
sd_instantiate_core -sd_name {smartdesign component name} \
                    -core_vlnv {vendor:library:name:verison} \
                    [-instance_name {instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
core_vlnv	string	Specifies the version identifier of the core being instantiated in the SmartDesign component. It is mandatory.
instance_name	string	Specifies the instance name of the core being instantiated in the SmartDesign. It is optional. By default, the instancename is <core_name>_<index> (index is an automatically generated integer starting at 0 such that the instancename is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_name' has illegal value.
None	Parameter 'core_vlnv' has illegal value.
None	Required parameter 'core_vlnv' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'sd_instantiate_core -sd_name "sd_name" -core_vlnv "core_vlnv" [-instance_name "instance_name"] [-promote_all "TRUE | FALSE"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example instantiates "COREAXI4INTERCONNECT_C0_0" instance with {Actel:DirectCore:COREAXI4INTERCONNECT:2.5.100} vendor, library, name and version.

```
sd_instantiate_core \
        -sd_name {top} \
        -core_vlnv {Actel:DirectCore:COREAXI4INTERCONNECT:2.5.100} \
        -instance_name {COREAXI4INTERCONNECT_C0_0}
```

4.33. sd_instantiate_hdl_core (Ask a Question)

Description

This Tcl command instantiates a HDL+ core in a SmartDesign component. HDL+ core definition must be created on a HDL module before using this command.

```
sd_instantiate_hdl_core -sd_name {smartdesign component name} \
                        -hdl_core_name {hdl core module name} \
                        [-instance_name {instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
hdl_core_name	string	Specifies the name of the HDL+ core module being instantiated in the SmartDesign component. It is mandatory.
instance_name	string	Specifies the instance name of the HDL+ core being instantiated in the SmartDesign. It is optional. By default, the instance name is <hdl_core_module_name>_<index>; (index is an automatically generated integer starting at 0 such that the instance name is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'hdl_core_name' is missing.
None	'1' violates HDL naming rules, please specify a different name
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_instantiate_hdl_core -sd_name "sd_name" -hdl_core_name "hdl_core_name" [-instance_name "instance_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example instantiates a HDL+ "top" core in a "mydesign" SmartDesign component as "top_0" (by default):

```
sd_instantiate_hdl_core -sd_name {mydesign} \
                        -hdl_core_name {top} \
                        -instance_name {}
```

See Also

- sd_instantiate_hdl_macro
- sd_instantiate_core

4.34. sd_instantiate_hdl_module (Ask a Question)

Description

This Tcl command instantiates a HDL module in a SmartDesign component. The HDL file in which the HDL module is defined must be imported/linked before running this command.

```
sd_instantiate_hdl_module -sd_name {smartdesign component name} \
                          -hdl_module_name {hdl module name} \
                          -hdl_file {hdl file} \
                          [-instance_name {instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
hdl_module_name	string	Specifies the name of the HDL module being instantiated in the SmartDesign component. It is mandatory.
hdl_file	string	Specifies the path of the HDL file in which the HDL module is defined. The HDL file path can be relative to project folder for imported files but the path has to be complete for linked files. It is mandatory.
instance_name	string	Specifies the instance name of the HDL module. It is optional. By default, the instance name is <hdl_module_name>_<index> (index is an automatically generated integer starting at 0 such that the instance name is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'hdl_file' is missing.
None	You cannot instantiate a sub-module 'module_name' of HDL module.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_instantiate_hdl_module -sd_name "sd_name" [-hdl_module_name "hdl_module_name"] -hdl_file "hdl_file" [-instance_name "instance_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

sd_instantiate_hdl_module (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example instantiates a 'top' module in a SmartDesign 'mydesign' component "top_0" instance name (by default):

```
sd_instantiate_hdl_module -sd_name {mydesign} \
                          -hdl_module_name {top} \
                          -hdl_file {hdl/top.v} \
                          -instance_name {}
```

See Also

- sd_instantiate_hdl_core
- sd_instantiate_core

4.35. sd_instantiate_macro (Ask a Question)

Description

This tcl command instantiates a Microsemi primitive macro in a SmartDesign component.

```
sd_instantiate_macro -sd_name {smartdesign component name} \
                     -macro_name {macro module name} \
                     [-instance_name {instance name}]
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
macro_name	string	Specifies the name of the macro being instantiated in the SmartDesign component. It is mandatory.
instance_name	string	Specifies the instance name of the macro. It is optional. By default, the instance name is <macroname>_<index> (index is an automatically generated integer starting at 0 such that the instance name is unique in the SmartDesign).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'macro_name' is missing.
None	Required parameter 'sd_name' is missing.
None	Unable to find macro 'some_macro_name' in the ADLIB database.
None	The component 'component_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_instantiate_macro -sd_name "sd_name" -macro_name "macro_name" [-instance_name "instance_name"]'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

- The following Tcl command shows how to use instance of "MX2" multiplexer macro in TOP design:

```
sd_instantiate_macro -sd_name {TOP} -macro_name {MX2} -instance_name {MX2_0}
```

- The following Tcl command shows how to use instance of "MX2" multiplexer macro in TOP design, in this case creates instance with macro name and index(unique number):

```
sd_instantiate_macro -sd_name {TOP} -macro_name {MACC_PA}
```

4.36. sd_invert_pins (Ask a Question)

Description

This Tcl command inverts one or more top level ports or instance level pins in a SmartDesign.

Note: This command will not work on multiple pins in release v2021.1. Support for multiple pins will be provided in the next Libero release.

```
sd_invert_pins -sd_name {smartdesign component name} \ -pin_names {port or pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Name of the SmartDesign component. It is mandatory.
pin_names	string	Specified the names of the ports or pins to be inverted. It is mandatory. This parameter can take multiple values(example below). The command will fail if the port/pin does not exist.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	The component 'design_name' doesn't exist.
None	Parameter 'd' is not defined. Valid command formatting is 'sd_invert_pins -sd_name "sd_name" -pin_names "[pin_names]+"'.
SDCTRL05	Pin 'pin_name' does not exist.

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

sd_invert_pins (continued)
Supported Families
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example inverts "din_d" top level port in a "mydesign" SmartDesign:

```
sd_invert_pins -sd_name {mydesign} -pin_names {"din_d"}
```

The following example inverts "top_0:clk" instance pin in a "mydesign" SmartDesign:

```
sd_invert_pins -sd_name {mydesign} -pin_names {"top_0:clk"}
```

4.37. sd_mark_pins_unused (Ask a Question)

Description

This tcl command marks one or more SmartDesign instance level output pins as unused. When an output pin is marked as unused, no Design Rule Check (DRC) warning will be printed for floating output pins while generating the SmartDesign.

Note: This command will not work on multiple pins in release v2021.1. Support for multiple pins will be provided in the next Libero release.

```
sd_mark_pins_unused -sd_name {smartdesign component name} \
                    -pin_names {port or pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
pin_names	string	Specifies the names of the instance pins to be marked as unused. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'pin_names' has illegal value.
None	Required parameter 'pin_names' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'sd_mark_pins_unused -sd_name "sd_name" -pin_names "pin_names"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

sd_mark_pins_unused (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example marks unused "PF_CCC_C0_0:PLL_LOCK_0" pin.

```
sd_mark_pins_unused -sd_name {top} -pin_names {PF_CCC_C0_0:PLL_LOCK_0}
```

See Also

- sd_clear_pin_attribute

4.38. sd_remove_pins_from_group (Ask a Question)

Description

This tcl command removes one or more pins from a pin group on an instance in a SmartDesign component.

```
sd_remove_pins_from_group -sd_name {smartdesign component name} \
                          -instance_name {instance name} \
                          -group_name {group name} \
                          -pin_names {pin names}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance on which the pin group is present. It is mandatory.
group_name	string	Specifies the name of the pin group from which pins need to be removed. It is mandatory.
pin_names	string	Specifies the list of pin names to be removed from the pin group(example below). It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_remove_pins_from_group -sd_name "sd_name" -instance_name "instance_name" [-group_name "group_name"] [-pin_names "[pin_names]+"]'.
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.
None	The component 'component_name' doesn't exist.
SDCTRL05	Pin 'group_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

sd_remove_pins_from_group (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example removes "ARESETN ACLK" pin from "MyGroup", that is on "COREAXINTERCONNECT_C0_0" instance, in the "TOP" design:

```
sd_remove_pins_from_group -sd_name {TOP} \
                          -instance_name {COREAXINTERCONNECT_C0_0} \
                          -group_name {Group} \
                          -pin_names {ARESETN ACLK}
```

The following example removes "WA_RSTn", "DISP_SEL" pins from "MyGroup", that is on "CorePCS_C0_0" instance, in the "TOP" design:

```
sd_remove_pins_from_group -sd_name {TOP} \
              -instance_name {CorePCS_C0_0} \
                          -group_name {MyGroup} \
                          -pin_names {"WA_RSTn" "DISP_SEL"}
```

See Also

- sd_create_pin_group
- sd_add_pins_to_group

4.39. sd_rename_instance (Ask a Question)

Description

This Tcl command renames an instance name in a SmartDesign component. This command can be used to rename any type of instances (instances of other SmartDesigns components, core components, HDL modules, HDL+ cores and Microchip macros) in a SmartDesign.

Note: This command is not required to build a SmartDesign component. It command maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_rename_instance -sd_name {smartdesign component name} \
                   -current_instance_name {instance name} \
                   -new_instance_name {new instance name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component in which the instance name has to be renamed. It is mandatory.
current_instance_name	string	Specifies the name of the instance to be renamed. It is mandatory.
new_instance_name	string	Specifies the new instance name. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'new_instance_name' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_rename_instance (continued)	
Error Code	**Description**
None	Parameter 'current_instance_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	'Instance name' has been duplicated, please specify a different name.
None	Instance 'instance_name' does not exist.
None	The first character of '*' must be a letter.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_rename_instance -sd_name "sd_name" -current_instance_name "current_instance_name" -new_instance_name "new_instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command renames "DFF_0" instance name with a new "DFF_1" name in the 'top' SmartDesign

```
sd_rename_instance -sd_name {top} \
                   -current_instance_name {DFF_0} \
                   -new_instance_name {DFF_1}
```

See Also

- sd_instantiate_macro

4.40. sd_rename_net (Ask a Question)

Description

This Tcl command renames a net in a SmartDesign component.

```
sd_rename_net -sd_name {smartdesign component name} \
              -current_net_name {current net name} \
              -new_net_name {new net name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
current_net_name	string	Specifies the name of the net to be renamed in the SmartDesign component. It is mandatory.
new_net_name	string	Specifies the new name of the net in the SmartDesign. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_rename_net (continued)	
Error Code	**Description**
None	Required parameter 'current_net_name' is missing.
None	Required parameter 'new_net_name' is missing.
None	Parameter 'new_net_name' has illegal value.
None	Parameter 'current_net_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_rename_net -sd_name "sd_name" -current_net_name "current_net_name" -new_net_name "new_net_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This command renames "clk_net" net name to "clk_rclk_wclk" in the "top" design:

```
sd_rename_net -sd_name {top} \
              -current_net_name {clk_net} \
              -new_net_name {clk_rclk_wclk}
```

This command renames "USER_RESETN" to "reset_input" in the "PCIe_EP_Demo" design:

```
sd_rename_net -sd_name {PCIe_EP_Demo} \
              -current_net_name {USER_RESETN} \
              -new_net_name {reset_input}
```

See Also

- sd_create_scalar_net
- sd_delete_nets

4.41. sd_rename_pin_group (Ask a Question)

Description

This Tcl command renames a pin group on an instance in a SmartDesign component.

```
sd_rename_pin_group -sd_name {smartdesign component name} \
                    -instance_name {instance name} \
                    -current_group_name {current pin group name} \
                    -new_group_name {new pin group name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance on which the pin group is present. It is mandatory.
current_group_name	string	Specifies the name of the pin group to be renamed. It is mandatory.
new_group_name	string	Specifies the new name of the pin group. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.
None	Required parameter 'current_group_name' is missing.
None	Required parameter 'new_group_name' is missing.
None	Parameter 'param' is not defined. Valid command formatting is 'sd_rename_pin_group -sd_name "sd_name" -current_group_name "current_group_name" -new_group_name "new_group_name" -instance_name "instance_name"'.
SDCTRL05	Pin 'MyGroup' does not exist.
None	Parameter 'current_group_name' has illegal value.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This command renames "Group" to "MyNewGroup" on the "COREAXINTERCONNECT_C0_0" instance in the "TOP" design:

```
sd_rename_pin_group -sd_name {TOP} \
                    -instance_name {COREAXINTERCONNECT_C0_0} \
                    -current_group_name {Group} \
                    -new_group_name {MyNewGroup}
```

See Also

- sd_create_pin_group
- sd_add_pins_to_group

4.42. sd_rename_port (Ask a Question)

Description

This Tcl command renames a SmartDesign port name.

Important: This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the exported SmartDesign component Tcl description.

```
sd_rename_port -sd_name {smartdesign component name} \
               -current_port_name {port name} \
               -new_port_name {new port name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
current_port_name	string	Specifies the name of the port to be renamed in the SmartDesign component. It is mandatory. **Note:** Only port names can be renamed, and not port types (scalar ports cannot be renamed as bus ports and vice versa).
new_port_name	string	Specifies the new name of the specified port. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	'P*' is containing the invalid character '*'.
None	Parameter 'new_port_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'current_port_name' has illegal value.
None	Port 'port_name' doesn't exist.
None	Required parameter 'current_port_name' is missing.
None	Required parameter 'new_port_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_rename_port -sd_name "sd_name" -current_port_name "current_port_name" -new_port_name "new_port_name"'.
SDCTRL02	Port name already exists.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command renames c1 port name with c2 in "top" design:

```
sd_rename_port -sd_name {top} -current_port_name {c1} -new_port_name {c2}
```

See Also

- sd_create_scalar_port

4.43. sd_replace_component (Ask a Question)

Description

This Tcl command replaces a SmartDesign instance component definition with a new component definition. This command is useful when the interface (port-list) of the component/module instantiated in a SmartDesign has changed. It can be used to replace any type of instance such

© 2025 Microchip Technology Inc. and its subsidiaries

as instances of other SmartDesign components, core components, HDL modules and HDL+ cores in a SmartDesign.

Important: This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the 'Export Component Description(Tcl)' but will present in Libero Project 'Export Script File'.

```
sd_replace_component -sd_name {smartdesign component name} \
                     -instance_name {instance name} \
                     -new_component_name {component name to be replaced}
                     -replace_all_instances {1|0}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance to be replaced with the new component definition. It is mandatory. The component can be a Core component/SmartDesign/HDL/HDL+.
new_component_name	string	This argument is used to specify the new component name and it's corresponding file path. This is a mandatory argument. For the HDL modules we need you need to specify HDL module name. The component can be a Core/SmartDesign/HDL/HDL+ etc and the file path is the (CXF/HDL) file corresponding to the component. User must be specified as follows: {component_name%file_path%library%package}.
replace_all_instances	boolean	Specifiy true or 1 to replace all the instances in the SmartDesign with the same component. Specify false or 0 to replace only mentioned instance. Default value is 1. This is an optional argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance' is missing.
None	Required parameter 'new_component_name' is missing.
None	Instance 'instance_name' does not exist.
None	The component 'component_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_replace_component -sd_name "sd_name" -instance "instance" -new_component_name "new_component"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

sd_replace_component (continued)
Supported Families
IGLOO® 2

Example

The following example replaces "sd_0" instance with "sd_1" module:

```
sd_replace_component -sd_name {mydesign} \
                     -instance {sd_0} \
                     -new_component_name {top%/exprj/hdl/sd_1.v}
                     -replace_all_instances 0
```

4.44. sd_reset_layout (Ask a Question)

Description

This tcl command resets the canvas layout of the given SmartDesign.

```
sd_reset_layout \
-sd_name {SmartDesign_component_name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign on which the layout will be auto arranged. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	The component 'SmartDesign_component_name' does not exist.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example resets the canvas layout of the given SmartDesign.

```
sd_reset_layout -sd_name {sd}
```


© 2025 Microchip Technology Inc. and its subsidiaries

4.45. sd_set_port_synth_attr (Ask a Question)

Description

This Tcl command sets synthesis attribute to the specified port of the specified SmartDesign. The attribute is added in the HDL file at the component generation.

```
sd_set_port_synth_attr \
        -sd_name {SmartDesign component name} \
        -port_name {port name} \
        -attr_name {synthesis attribute name} \
        -attr_value {synthesis attribute value}
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component in which the port synthesis attribute is set. This is a mandatory argument.
port_name	string	Name of the port in SmartDesign that synthesis attribute is added to. This is a mandatory argument.
attr_name	string	Synthesis attribute to be added. This is a mandatory argument.
attr_value	string	Value of the attribute. Some attributes do not require values. This is a mandatory argument.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
SDCTRL05	Pin 'port_name' does not exist.
None	Parameter 'port_name' has illegal value.
None	Required parameter 'port_name' is missing.
None	Parameter 'attr_value' has illegal value.
None	Required parameter 'attr_value' is missing.
None	Parameter 'attr_name' has illegal value.
None	Required parameter 'attr_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_set_port_synth_attr -sd_name "sd_name" -attr_name "attr_name" -attr_value "attr_value" -port_name "port_name"'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example sets "syn_keep" attribute with "false" value on the "PRESETN" port in the "top" design.

```
sd_set_port_synth_attr -sd_name {top} -attr_name {syn_keep} \
                       -attr_value {false} -port_name {PRESETN}
```

See Also

- sd_remove_port_synth_attr

4.46. sd_set_net_synth_attr (Ask a Question)

Description

Set synthesis attribute to the specified net of the specified SmartDesign. The attribute is added in the HDL file at the component generation.

```
sd_set_net_synth_attr \
        -sd_name {SmartDesign component name} \
        -net_name {net name} \
        -attr_name {synthesis attribute name} \
        -attr_value {synthesis attribute value}
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component in which the port synthesis attribute is set. This is a mandatory argument.
net_name	string	Name of the net in SmartDesign that synthesis attribute is added to. This is a mandatory argument.
attr_name	string	Synthesis attribute. This is a mandatory argument.
attr_value	string	Value of the attribute. Some attributes do not require values.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Net 'net_name' does not exist.
None	Parameter 'net_name' has illegal value.
None	Required parameter 'net_name' is missing.
None	Parameter 'attr_value' has illegal value.
None	Required parameter 'attr_value' is missing.
None	Parameter 'attr_name' has illegal value.
None	Required parameter 'attr_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_set_net_synth_attr -sd_name "sd_name" -attr_name "attr_name" -attr_value "attr_value" -net_name "net_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example sets "syn_keep" attribute with "false" value on the "CLK" net in the "top" design.

```
sd_set_net_synth_attr -sd_name {top} -attr_name {syn_keep} \
                      -attr_value {false} -net_name {CLK}
```

See Also

- sd_remove_net_synth_attr

4.47. sd_set_inst_synth_attr (Ask a Question)

Description

This Tcl command sets synthesis attribute to the specified instance of the specified SmartDesign. The attribute is added in the HDL file at the component generation.

```
sd_set_inst_synth_attr \
        -sd_name {SmartDesign component name} \
        -instance_name {instance name} \
        -attr_name {synthesis attribute name} \
        -attr_value {synthesis attribute value}
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component in which the port synthesis attribute is set. This is a mandatory argument.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_set_inst_synth_attr (continued)		
Parameter	**Type**	**Description**
instance_name	string	Name of the instance in SmartDesign that synthesis attribute is added to. This is a mandatory argument.
attr_name	string	Synthesis attribute. This is a mandatory argument.
attr_value	string	Value of the attribute. Some attributes do not require values.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Instance 'instance_name' does not exist.
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.
None	Parameter 'attr_value' has illegal value.
None	Required parameter 'attr_value' is missing.
None	Parameter 'attr_name' has illegal value.
None	Required parameter 'attr_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_set_inst_synth_attr -sd_name "sd_name" -attr_name "attr_name" -attr_value "attr_value" -instance_name "instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example sets "syn_insert_buffer" attribute with "none" value on the "COREABC_C0_0" instance in the "top" design.

```
sd_set_inst_synth_attr -sd_name {top} -attr_name {syn_insert_buffer} \
                       -attr_value {none} -instance_name {COREABC_C0_0}
```

See Also

- sd_remove_inst_synth_attr

4.48. sd_set_comp_synth_attr (Ask a Question)

Description

This Tcl command sets synthesis attribute to the specified SmartDesign module. The attribute is added in the HDL file at the component generation.

```
sd_set_comp_synth_attr \
        -sd_name {SmartDesign component name} \
        -attr_name {synthesis attribute name} \
        -attr_value {synthesis attribute value}
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign module in which the synthesis attribute is set. This is a mandatory argument.
attr_name	string	This argument specifies the synthesis attribute. This is a mandatory argument.
attr_value	string	Value of the attribute. Some attributes do not require values.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Error: Failed to set synthesis attribute 'tlm_psioooc_eynn_pin' to component 'top'. This attribute cannot be set to components.
None	Parameter 'attr_value' has illegal value.
None	Required parameter 'attr_value' is missing.
None	Parameter 'attr_name' has illegal value.
None	Required parameter 'attr_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_set_comp_synth_attr (continued)	
Error Code	**Description**
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_set_comp_synth_attr -sd_name "sd_name" -attr_name "attr_name" -attr_value "attr_value"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example sets "syn_no_compile_point" attribute with "false" value on the "top" component.

```
sd_set_comp_synth_attr -sd_name {top} \
                       -attr_name {syn_no_compile_point} \
                       -attr_value {false}
```

See Also

- sd_remove_comp_synth_attr

4.49. sd_remove_port_synth_attr (Ask a Question)

Description

This Tcl command removes synthesis attribute from the specified port of the specified SmartDesign.

```
sd_remove_port_synth_attr \
        -sd_name {SmartDesign component name} \
        -port_name {port name} \
        [-attr_name {synthesis attribute name}]
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component from which the port synthesis attribute is removed. This is a mandatory argument.
port_name	string	Name of the port in SmartDesign from which synthesis attribute is removed. This is a mandatory argument.
attr_name	string	Synthesis attribute to be deleted. If the argument is not specified, all the attributes are deleted.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive

© 2025 Microchip Technology Inc. and its subsidiaries

sd_remove_port_synth_attr (continued)		
Name	**Object**	**Attribute/Directive**
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Failed to remove synthesis attribute 'syn_preserve' from the port 'port_name'. This attribute does not exist.
SDCTRL05	Pin 'port_name' does not exist.
None	Parameter 'port_name' has illegal value.
None	Required parameter 'port_name' is missing.
None	Parameter 'attr_name' has illegal value.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_remove_port_synth_attr -sd_name "sd_name" [-attr_name "attr_name"] -port_name "port_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example removes "syn_preserve" attribute of "PRESETN" port in the "top" design.

```
sd_remove_port_synth_attr -sd_name {top} -attr_name {syn_preserve} \
                          -port_name {PRESETN}
```

See Also

- sd_set_port_synth_attr

4.50. sd_remove_net_synth_attr (Ask a Question)

Description

This Tcl command removes synthesis attribute from the specified net of the specified SmartDesign.

```
sd_remove_net_synth_attr \
        -sd_name {SmartDesign component name} \
        -net_name {net name} \
        [-attr_name {synthesis attribute name}]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component from which the net synthesis attribute is removed. This is a mandatory argument.
net_name	string	Name of the net in SmartDesign from which synthesis attribute is removed. This is a mandatory argument.
attr_name	string	Synthesis attribute to be deleted. If the argument is not specified all the attributes are deleted.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Failed to remove synthesis attribute 'syn_keep' from the net 'net_name'. This attribute does not exist.
None	Net 'net_name' does not exist.
None	Parameter 'net_name' has illegal value.
None	Required parameter 'net_name' is missing.
None	Parameter 'attr_name' has illegal value.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_remove_net_synth_attr -sd_name "sd_name" [-attr_name "attr_name"] -net_name "net_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example removes "syn_keep" attribute of "CLK" net in the "top" design.

```
sd_remove_net_synth_attr -sd_name {top} \
                         -attr_name {syn_keep} \
                         -net_name {CLK}
```

See Also

- sd_set_net_synth_attr

4.51. sd_remove_inst_synth_attr (Ask a Question)

Description

This Tcl command removes the synthesis attribute from the specified instance of the specified SmartDesign.

```
sd_remove_inst_synth_attr \
        -sd_name {SmartDesign component name} \
        -instance_name {instance name} \
        [-attr_name {synthesis attribute name}]
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign component from which the instance synthesis attribute is removed. This is a mandatory argument.
inst_name	string	Name of the instance in SmartDesign from which synthesis attribute is removed. This is a mandatory argument.
attr_name	string	Synthesis attribute to be deleted. If the argument is not specified, all the attributes are deleted.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Failed to remove the synthesis attribute 'syn_insert_buffer' from the instance 'top'. This attribute does not exist.
None	Instance 'instance_name' does not exist.
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_remove_inst_synth_attr (continued)	
Error Code	**Description**
None	Parameter 'attr_name' has illegal value.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_remove_inst_synth_attr -sd_name "sd_name" [-attr_name "attr_name"] -instance_name "instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example removes the "syn_insert_buffer" attribute of "COREABC_C0_0" instance in the "top" design.

```
sd_remove_inst_synth_attr -sd_name {top} -attr_name {syn_insert_buffer} \
                          -instance_name {COREABC_C0_0}
```

See Also

- sd_set_inst_synth_attr

4.52. sd_remove_comp_synth_attr (Ask a Question)

Description

This Tcl command removes the synthesis attribute from the specified SmartDesign module.

```
sd_remove_comp_synth_attr -sd_name {SmartDesign component name} \
                          [-attr_name {synthesis attribute name}]
```

Arguments

Parameter	Type	Description
sd_name	string	This argument is used to specify the name of the SmartDesign module from which the synthesis attribute is removed. This is a mandatory argument.
attr_name	string	Synthesis attribute to be deleted. If the argument is not specified, all the attributes are deleted.

Synthesis Attributes

The following is the list of attributes and directives available in the Synopsys® FPGA synthesis tool that are supported by SmartDesign.

Name	Object	Attribute/Directive
syn_insert_buffer	port, instance	Attribute
syn_keep	net	Directive
syn_maxfan	port, net, instance, register	Attribute
syn_no_compile_point	module/architecture	Attribute

© 2025 Microchip Technology Inc. and its subsidiaries

sd_remove_comp_synth_attr (continued)		
Name	**Object**	**Attribute/Directive**
syn_noclockbuf	port, net, module/architecture	Attribute
syn_noprune	instance, module/architecture	Directive
syn_preserve	register, port, module/architecture	Directive

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Failed to remove the synthesis attribute 'syn_no_compile_point' from the component 'top'. This attribute does not exist.
None	Parameter 'attr_name' has illegal value.
None	The component 'sd_name' does not exist.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_remove_comp_synth_attr -sd_name "sd_name" [-attr_name "attr_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example removes the "syn_no_compile_point" attribute of the "top" component.

```
sd_remove_comp_synth_attr -sd_name {top} \
                          -attr_name {syn_no_compile_point}
```

See Also

- sd_set_comp_synth_attr

4.53. sd_save_core_instance_config (Ask a Question)

Description

This Tcl command is used to save the core instance configuration specified using one or more 'sd_configure_core_instance' commands. This command is typically used after configuring a core instance in a SmartDesign, to save that core instance's configuration.

```
sd_save_core_instance_config -sd_name {smartdesign component name} \
                             -instance_name {core instance name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_save_core_instance_config (continued)		
Parameter	**Type**	**Description**
instance_name	string	Specifies the name of the core instance in the SmartDesign for which the configuration must be saved. It ismandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'instance_name' has illegal value.
None	Required parameter 'instance_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'sd_save_core_instance_config -sd_name "sd_name" -instance_name "instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example saves "COREFIFO_0" core instance configuration.

```
sd_save_core_instance_config -sd_name {SD1} -instance_name {COREFIFO_0}
```

4.54. sd_show_bif_pins (Ask a Question)

Description

This tcl command exposes one or more internal scalar or bus pins/ports of a Bus Interface pin/port. A Bus Interfacepin/port is usually a group of normal scalar or bus pins/ports grouped together and used to connect instances thathave similar interfaces.

Note:
This command will not expose multiple pins/ports in release v2021.1. Support to expose multiple scalar or bus pins/portswill be provided in the next Libero release.

```
sd_show_bif_pins -sd_name {smartdesign component name} \
                 -bif_pin_name {name of the bif pin or port} \
                 -pin_names {pins or ports to be exposed}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
bif_pin_name	string	Specifies the name of the Bus Interface pin/port for which the internal pins/ports need to be exposed. It is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

sd_show_bif_pins (continued)		
Parameter	**Type**	**Description**
pin_names	string	Specifies the names of the Bus Interface internal pins/ports to be exposed. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'pin_names' has illegal value.
None	Required parameter 'pin_names' is missing.
None	Parameter 'bif_pin_name' has illegal value.
None	Required parameter 'bif_pin_name' is missing.
None	Parameter 'sd_name' has illegal value.
None	Required parameter 'sd_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_show_bif_pins -sd_name "sd_name" -bif_pin_name "bif_pin_name" -pin_names "pin_names"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example exposes "COREAXI4INTERCONNECT_C0_0:MASTER0_AWADDR" pin for "COREAXI4INTERCONNECT_C0_0:AXI4mmaster0" bif pin.

```
sd_show_bif_pins -sd_name {top} \
                 -bif_pin_name {COREAXI4INTERCONNECT_C0_0:AXI4mmaster0} \
                 -pin_names {COREAXI4INTERCONNECT_C0_0:MASTER0_AWADDR}
```

See Also

- sd_delete_bif_pin

4.55. sd_update_instance (Ask a Question)

Description

This Tcl command updates an instance in a SmartDesign with its latest definition. This command is useful when the interface (port-list) of the component/module instantiated in a SmartDesign has changed. This command can be used to update any type of instance such as instances of other SmartDesign components, core components, HDL modules and HDL+ cores in a SmartDesign.

© 2025 Microchip Technology Inc. and its subsidiaries

Note:
This command is not required to build a SmartDesign component. It maps to an interactive user action in the SmartDesign Canvas and will not be present in the 'Export Component Description(Tcl)' but will present in Libero Project 'Export Script File'.

```
sd_update_instance -sd_name {smartdesign component name} \
                   -instance_name {instance name}
```

Arguments

Parameter	Type	Description
sd_name	string	Specifies the name of the SmartDesign component. It is mandatory.
instance_name	string	Specifies the name of the instance to be updated. It is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'sd_name' is missing.
None	Required parameter 'instance_name' is missing.
None	The component 'comp_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'sd_update_instance -sd_name "sd_name" -instance_name "instance_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example updates 'dff' instance in a 'top' SmartDesign with its latest definition:

```
sd_update_instance -sd_name {top} -instance_name {dff}
```

See Also

- sd_instantiate_component

© 2025 Microchip Technology Inc. and its subsidiaries

5. HDL Tcl Commands (Ask a Question)

5.1. create_hdl_core (Ask a Question)

Description

This Tcl command is used to create a core component from an HDL core. If an incorrect module name is specified, the command fails.

Important: If you are running this command standalone, you need to add the `build_design_hierarchy` command prior to this command.

```
create_hdl_core -file "file name" -module "module name" \
[-library "library name"] [-package "package name"]
```

Arguments

Parameter	Type	Description
file	string	Specify the absolute file path of the module from which you want to create a core component. This is a mandatory argument.
module	string	Specify the module name for which you want to create a core component. This is a mandatory argument.
library	string	Specify the library name from which you want to create an HDL core. This is an optional argument.
package	string	Specify the package name from which you want to create a core component. This is an optional argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'create_hdl_core -file "file name" -module "module name" [-library "library"] [-package "package"]'.
None	Required parameter 'module' is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example creates core component for "test_hdl_core" module from the "hdl_core.v" HDL:

```
create_hdl_core -file {./HDL_CORE_TEST/hdl/hdl_core.v} \
-module {test_hdl_core}
```

5.2. remove_hdl_core (Ask a Question)

Description

This Tcl command removes and HDL core component from the current project. The command will fail if the module name is not specified or is incorrect.

```
remove_hdl_core -hdl_core_name { hdl_core_name }
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the module name from which you want to delete a core component. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is ' remove_hdl_core -hdl_core_name "hdl core name"'.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example removes 'test_hdl_core' HDL core component:

```
remove_hdl_core -hdl_core_name {test_hdl_core}
```

5.3. hdl_core_add_bif (Ask a Question)

Description

This Tcl command adds a bus interface to an HDL core.

The command will fail if the module name or Bus Interface Definition are not specified or are incorrect.

```
hdl_core_add_bif \
-hdl_core_name { hdl_core_name } \
-bif_definition { Name:Vendor:Library:Role } \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
-bif_name { bus_interface_name } \
[-signal_map { signal_map }]
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name to which the bus interface needs to be added. This is a mandatory argument.
bif_definition	string	Specify the Bus Interface Definition Name, Vendor, Library and Bus Role of the core in the format {N:V:L:R}. This is a mandatory argument.
bif_name	string	Specify the bus interface port name being added to the HDL core. This is a mandatory argument.
signal_map	list of strings	This argument is used to specify the signal map of the bus interface. This is an optional argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	The bus interface 'BIF_name' has already been defined.
None	Parameter 'signal_map' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_add_bif -hdl_core_name "hdl_core_name" -bif_definition "BIF definition" -bif_name "BIF name" [-signal_map "[signal map]+"]'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command adds 'BIF_1' bus interface to 'test_hdl_core' HDL core with the spceified bus interface definition.

```
hdl_core_add_bif -hdl_core_name {test_hdl_core} \
-bif_definition {AHB:AMBA:AMBA2:master} -bif_name {BIF_1}
```

See Also

- hdl_core_remove_bif
- hdl_core_rename_bif

5.4. hdl_core_assign_bif_signal (Ask a Question)

Description

This Tcl command maps a bus interface signal definition name to an HDL core module port name.

© 2025 Microchip Technology Inc. and its subsidiaries

This Tcl command maps bus interface signal definition name to an HDL core module port name. The command will fail if the HDL core name is not specified or is incorrect.

```
hdl_core_assign_bif_signal
-hdl_core_name { hdl_core_name } \
-bif_name { bus_interface_name } \
-bif_signal_name { bif_signal_name } \
-core_signal_name { core_signal_name }
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name to which the bus interface signal needs to be added. This is a mandatory argument.
bif_name	string	Specify the bus interface name for which you want to map a core signal. This is a mandatory argument.
bif_signal_name	string	Specify the bus interface signal name that you want to map with the core signal name. This is a mandatory argument.
core_signal_name	string	Specify the core signal name for which you want to map the bus interface signal name. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Required parameter 'bif_name' is missing.
None	Parameter 'bus_interface' is missing or has invalid value.
None	Required parameter 'bif_signal_name' is missing.
None	Required parameter 'core_signal_name' is missing.
None	The bus interface 'BIF_name' has not been defined.
None	The bus interface signal 'SIGNAL NAME' has not been defined in the bus definition of the bus interface 'BIF name'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_assign_bif_signal -hdl_core_name "hdl_core_name" -bif_name "BIF name" -bif_signal_name "BIf signal name" -core_signal_name "core signal name"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command adds 'HWRITE' bus interface signal to 'test_hdl_core' HDL core, maps with the 'myHRESULT' core signal name:

```
hdl_core_assign_bif_signal -hdl_core_name {test_hdl_core} \
-bif_name {BIF_1} -bif_signal_name {HWRITE} -core_signal_name {myHRESULT}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- hdl_core_unassign_bif_signal

5.5. hdl_core_delete_parameters (Ask a Question)

Description

This Tcl command deletes parameters from a HDL core definition. After this command usage It will be impossible to configure parameters of HDL core.

The command will fail if the module name or parameter list are not specified or are incorrect.

```
hdl_core_delete_parameters -hdl_core_name { module_name } \
-parameters { parameter_list }
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name from which you want to delete parameters. This is a mandatory argument.
parameters	list of strings	Specify the list of parameters from a HDL core. This is typically done to remove parameters from the list of parameters that was automatically extracted using the "hdl_core_extract_ports_and_params" command. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Required parameter 'parameters' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_delete_parameters -hdl_core_name "hdl core name" -parameters "[parameter list]+"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example deletes "WIDTH" parameters from a "test_hdl_core" HDL core definition:

```
hdl_core_delete_parameters -hdl_core_name {test_hdl_core} \
-parameters {WIDTH}
```

See Also

- hdl_core_extract_ports_and_parameter

© 2025 Microchip Technology Inc. and its subsidiaries

5.6. hdl_core_extract_ports_and_parameters (Ask a Question)

Description

This Tcl command automatically extracts ports and generic parameters from an HDL core module description.

The command will fail if the module name is not specified or is incorrect.

```
hdl_core_extract_ports_and_parameters \
-hdl_core_name { hdl_core_name }
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specifies the HDL core name from which you want to extract signal names and generic parameters. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_extract_ports_and_parameters -hdl_core_name "hdl_core_name"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example automatically extracts ports and parameters from an "test_hdl_core" HDL core module description:

```
hdl_core_extract_ports_and_params -hdl_core_name {test_hdl_core}
```

See Also

- hdl_core_delete_parameters

5.7. hdl_core_remove_bif (Ask a Question)

Description

This Tcl command removes existing bus interface from an HDL core. The command will fail if the module name is not specified or is incorrect.

```
hdl_core_remove_bif \
-hdl_core_name { hdl_core_name } \
-bif_name { bus_interface_name }
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name from which the bus interface needs to be removed. This is a mandatory argument.
bif_name	string	Specify the bus interface name that needs to be removed from the HDL core. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Parameter 'bus_interface' is missing or has invalid value.
None	The bus interface 'BIF_name' has not been defined.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_remove_bif -hdl_core_name "hdl_core_name" -bif_name "BIF name"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command removes "BIF_1" bus interface from the "mod1" HDL core with the spceified bus interface name:

```
hdl_core_remove_bif -hdl_core_name {mod1} -bif_name {BIF_1}
```

See Also

- hdl_core_add_bif:q
- hdl_core_rename_bif

5.8. hdl_core_rename_bif (Ask a Question)

Description

This Tcl command renames existing bus interface port of a HDL core. The command will fail if the module name is not specified or is incorrect.

```
hdl_core_rename_bif
-hdl_core_name { hdl_core_name } \
-current_bif_name { current_bus_interface_name } \
-new_bif_name { new_bus_interface_name }
```

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name for which the bus interface needs to be renamed. This is a mandatory argument.

© 2025 Microchip Technology Inc. and its subsidiaries

hdl_core_rename_bif (continued)		
Parameter	**Type**	**Description**
current_bif_name	string	Specify the bus old bus interface name that needs to be renamed for the HDL core. This is a mandatory argument.
new_bif_name	string	Specify the new bus interface name that needs to be updated for the HDL core. This is a mandatory argument.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Parameter 'current_bus_interface' is missing or has invalid value.
None	Required parameter 'new_bif_name' is missing
None	The bus interface 'BIF_name' has not been defined.
None	You must specify at least one parameter among: 'name, connection_required, interface_rendering, description, export'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_rename_bif -hdl_core_name "hdl core name" -current_bif_name "Current BusInterface name" -new_bif_name "New BusInterface name"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command renames the existing 'BIF_1' bus interface port name from the 'dffp' HDL core with the spceified new 'BIF_2' bus interface name:

```
hdl_core_rename_bif -hdl_core_name {test_hdl_plus} \
-current_bif_name {BIF_1} -new_bif_name {BIF_2}
```

See Also

- hdl_core_add_bif
- hdl_core_remove_bif

5.9. hdl_core_unassign_bif_signal (Ask a Question)

Description

This Tcl command unmaps an existing bus interface signal from a bus interface. The command will fail if the HDL core name is not specified or is incorrect.

```
hdl_core_unassign_bif_signal
-hdl_core_name { hdl_core_name } \
-bif_name { bus_interface_name } \
-bif_signal_name { bif_signal_name }
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
hdl_core_name	string	Specify the HDL core name from which the bus interface signal needs to be deleted. This is a mandatory argument.
bif_name	string	Specify the bus interface name for which you want to unassign a core signal. This is a mandatory argument.
bif_signal_name	string	Specify the bus interface signal name for which you want to unassign a core signal. This argument is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hdl_core_name' is missing.
None	Required parameter 'bif_name' is missing.
None	The bus interface 'BIF_name' has not been defined.
None	Required parameter 'bif_signal_name' is missing.
None	Parameter 'bus_interface' is missing or has invalid value.
None	The bus interface signal 'SIGNAL NAME' has not been defined in the bus interface 'BIF name'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'hdl_core_unassign_bif_signal-hdl_core_name "hdl_core_name" -bif_name "BIF name" -bif_signal_name "BIf signal name"'

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command unmaps/unassigns an existing 'PENABLE' bus interface signal from a 'test_hdl_plus' HDL core for 'BIF_2' bus interface:

```
hdl_core_unassign_bif_signal -hdl_core_name {test_hdl_plus} \
-bif_name {BIF_2} -bif_signal_name {PENABLE}
```

See Also

- hdl_core_assign_bif_signal

© 2025 Microchip Technology Inc. and its subsidiaries

6. Command Tools (Ask a Question)

6.1. CONFIGURE_ACTIONS_PROCEDURES (Ask a Question)

Description

CONFIGURE_ACTIONS_PROCEDURES is a command tool used in configure_tool. It configures action with optional or recommended procedures for a Libero target device.

Tip:

- Available actions and their procedures depend on current bit stream components selected in the Generate Bitstream and Configure Options tools.
- Changing procedures for the action selected to run invalidates the Run Action tool state. Changing any other action does not affect the Run Action tool state.

```
configure_tool -name {CONFIGURE_ACTIONS_PROCEDURES} \
-params \
{-prog_optional_procedures:act1|proc1|proc2;act2|proc1|proc2|proc3;} \
-params \
{-skip_recommended_procedures:act1|proc1|proc2;act2|proc1|proc2|proc3;}
```

For more information about programming actions and supported procedures, see Libero SoC Design Flow User Guide.

Arguments

Parameter	Type	Description
prog_optional_procedures	string	Specifies optional procedures and actions to configue action with optional procedures.
skip_recommended_procedures	string	Specifies recommended procedures and actions to configure action with recommended procedures.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example configures "DO_VERIFY" action with the optional "PROGRAM" procedure, programs all selected family features: FPGA Array, targeted eNVM clients, and security settings:

```
configure_tool \
    -name {CONFIGURE_ACTION_PROCEDURES} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
    -params {prog_optional_procedures:PROGRAM|DO_VERIFY;} \
    -params {skip_recommended_procedures:}
```

The following example configures "DO_ENABLE_FABRIC" action with the recommended "VERIFY_DIGEST" procedure, calculates the digests for the components:

```
configure_tool \
    -name {CONFIGURE_ACTION_PROCEDURES} \
    -params {prog_optional_procedures:} \
    -params {skip_recommended_procedures:VERIFY_DIGEST|DO_ENABLE_FABRIC;}
```

6.2. CONFIGURE_CHAIN (Ask a Question)

Description

This Tcl command tool used in "run_tool." It takes a script file that contains specific Tcl commands and passes them to FlashPro Express for execution.

Important: For a new Libero project without a JTAG chain, executing this command causes Libero to first add the existing design device to the JTAG chain and then execute the commands from the script. If, for example, the script fpro_cmds.tcl contains commands to add four devices, executing the command run_tool -name. {CONFIGURE_CHAIN} -script {fpro_cmds.tcl} will create a JTAG chain of the Libero design device and the four devices. For existing Libero projects that already have a JTAG chain, the command is executed on the existing JTAG chain.

```
run_tool -name {CONFIGURE_CHAIN} -script {fpro_cmds.tcl}
```

Arguments

Parameter	Type	Description
name	string	Specify tool name.
script	string	Specify absolute or relative path of the script file. This is an optional parameter. fpro_cmds.tcl is a Tcl script that contains specific Tcl commands to configure JTAG chain. For details on JTAG chain programming Tcl commands, refer to the Tcl commands section in the Libero SoC Online Help. Do not include any project-management commands such as open_project, save_project, or close_project in this fpro_cmds.tcl script file. The run_tool -name {CONFIGURE_CHAIN} command generates these projectmanagement commands for you.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

CONFIGURE_CHAIN (continued)
Supported Families
IGLOO® 2

Example

The following example runs "CONFIGURE_CHAIN" tool with the script:

```
run_tool -name {CONFIGURE_CHAIN} -script {d:/fpro_cmds.tcl}
#Example fpro_cmds.tcl command file for the -script parameter
add_actel_device -file {./prj/designer/impl/sd.stp} -name {dev}
enable_device -name {M2S050TS_5} \
          -enable 0 add_non_actel_device \
              -ir 2 \
              -tck 1.00 \
              -name {Non-Microchip Device}
add_non_actel_device -ir 2 \
                     -tck 1.00 \
             -name {Non-Microchip Device (2)}
remove_device -name {Non-Microchip Device}
set_device_to_highz -name {M2S050TS_5} -highz 1
add_actel_device -device {M2S050TS} -name {M2S050TS (3)}
select_libero_design_device -name {M2S050TS (3)}
```

6.3. CONFIGURE_PROG_OPTIONS (Ask a Question)

Description

"CONFIGURE_PROG_OPTIONS" is a command tool used in configure_tool. It sets the programming options for PolarFire, PolarFire SoC, RTG4, SmartFusion 2 and IGLOO 2 devices.

The command usage for the PolarFire is the following:

```
configure_tool -name {CONFIGURE_PROG_OPTIONS} \
-params {design_version:<value>} \
-params {back_level_version:<value>} \
-params {silicon_signature:<value>}
```

The command usage for the SmartFusion 2 and IGLOO 2 is the following:

```
configure_tool -name {CONFIGURE_PROG_OPTIONS} \
-params {design_version:<value>} \
-params {enable_auto_update:true | false} \
-params {enable_prog_recovery:true | false} \
-params {silicon_signature:<value>} \
-params {spi_clk_freq:<value>} \
-params {spi_data_transfer_mode:<value>}
```

The command usage for the RTG4 is the following:

```
configure_tool -name {CONFIGURE_PROG_OPTIONS} \
-params {design_version:<value>} \
-params {disable_digest_check:<value>} \
-params {disable_fabric_erase_write_verify:<value>} \
-params {disable_jtag:<value>} \
-params {disable_probe_read_write:<value>} \
-params {disable_spi:<value>} \
-params {one_time_programmable:<value>} \
-params {silicon_signature:<value>} \
-params {system_controller_suspend_mode:<value>}
```

Arguments

The following table list the "CONFIGURE_PROG_OPTIONS" arguments for PolarFire.

Parameter	Type	Description
design_version	Integer {0 through 65535}	Sets the design version. It must be greater than the Back Level version in SPM Update Policy.

© 2025 Microchip Technology Inc. and its subsidiaries

CONFIGURE_PROG_OPTIONS (continued)		
Parameter	**Type**	**Description**
back_level_version	Integer {0 through 65535}	Sets the back level version.
silicon_signature	Hex {<max length 8 Hex characters>}	32-bit (8 hex characters) silicon signature to be programmed into the device. This field can be read from the device using the JTAG USERCODE instruction.

The following table list the "CONFIGURE_PROG_OPTIONS" arguments for SmartFusion 2/IGLOO 2.

Parameter	Type	Description
design_version	Integer {0 through 65535}	Sets the design version. It must be greater than the Back Level version in SPM Update Policy.
silicon_signature	Hex {<max length 8 Hex characters>}	32-bit (8 hex characters) silicon signature to be programmed into the device. This field can be read from the device using the JTAG USERCODE instruction.
enable_auto_update	boolean	Sepcify "TRUE" or 1 to enable auto update, spceify "FALSE" or 0 to disable auto update. This parameter is available for SmartFusion 2.
enable_prog_recovery	boolean	Sepcify "TRUE" or 1 to enable programming recovery, spceify "FALSE" or 0 to disable programming recovery. This parameter is available for SmartFusion 2.
spi_clk_freq	double	Sets SPI clock frequency from a list of possible values {1.00 | 2.08 | 3.13 | 4.16 | 5.00 | 6.25 | 8.30 | 12.50 | 25.00 }. This parameter is available for SmartFusion 2.
spi_data_transfer_mode	binary	SPI data transfer mode sets the values for SPS, SPO and SPH in the UI. SPS has a fixed value of 1 and cannot be changed. The user can change the value of only SPO and SPH to 0 or 1. The acceptable value are the following: {100 | 101 | 111 | 110}. This parameter is available for SmartFusion 2.

The following table list the "CONFIGURE_PROG_OPTIONS" arguments for RTG4.

Parameter	Type	Description
design_version	Integer {0 through 65535}	Sets the design version. It must be greater than the Back Level version in SPM Update Policy.
silicon_signature	Hex {<max length 8 Hex characters>}	32-bit (8 hex characters) silicon signature to be programmed into the device. This field can be read from the device using the JTAG USERCODE instruction.
disable_digest_check	boolean	Enable or disable digest check.
disable_fabric_erase_write_verify	boolean	Enable or disable Fabric Erase/Write/Verify.
disable_jtag	boolean	Enable or disable JTAG interface.
disable_probe_read_write	boolean	Enable or disable Probe Read/Write.
disable_spi	boolean	Enable or disable SPI interface.
one_time_programmable	boolean	Set "true" or 1 to make the device one-time programmable. **Note:** After programming the device you will NOT be able to erase or program the device. You will be able to run programming actions VERIFY and VERIFY_DIGEST as well as use SmartDebug to debug with probes and read the digest of the Fabric.
system_controller_suspend_mode	boolean	Enable or disable System Controller Suspend mode.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Incorrect Back Level version value format.
None	Back Level version must be less than or equal to Design version.
None	parameter disable_spi does not exist (PolarFire).
None	parameter one_time_programmable does not exist (PolarFire).
None	Incorrect Design version value format.
None	Design version value must be between 0 and 65535.
None	Silicon signature must be max 8 HEX chars.
None	Illegal SPI clock frequency '1.11 MHz'.
None	Programming Recovery must be enabled to allow Auto Update.
None	Silicon signature must be max 8 HEX chars.
None	Unknown SPI data transfer mode: SPS(0) SPO(0) SPH(0).

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example configures programming parameters for the PolarFire device:

```
configure_tool -name {CONFIGURE_PROG_OPTIONS} \
               -params {design_version:255} \
               -params {back_level_version:200} \
               -params {silicon_signature:abcdef}
```

The following example configures programming parameters for the SmartFusion 2 device:

```
configure_tool -name {CONFIGURE_PROG_OPTIONS}\
               -params {design_version:255}
               -params {enable_auto_update:true}
               -params {enable_prog_recovery:true}
               -params {silicon_signature:abcdef}
               -params {spi_clk_freq:25.00}
               -params {spi_data_transfer_mode:100}
```

See Also

- CONFIGURE_ACTIONS_PROCEDURES
- set_programming_action

6.4. EXPORTNETLIST (Ask a Question)

Description

"EXPORTNETLIST" is a command tool used in the run_tool command. This command exports a `*.v` or `*.vhd` netlist file to the active synthesis implementation folder.

```
run_tool -name {EXPORTNETLIST}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command exports netlist file:

```
run_tool -name {EXPORTNETLIST}
```

6.5. EXPORTSDF (Ask a Question)

Description

"EXPORTSDF" is a command tool used in the configure_tool and run_tool command. This command exports the back annotated <root_name>_ba.v and <root_name>_ba.sdf files to the designer/<root_name> folder.

```
configure_tool -name {EXPORTSDF} -params {DELAY_TYPE: true | false}
run_tool -name {EXPORTSDF}
```

Arguments

Parameter	Type	Description
DELAY_TYPE	boolean	Exports your enhanced min delays to include your best-case timing results in your Back Annotated file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

EXPORTSDF (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example exports Back Annotated file:

```
configure_tool -name {EXPORTSDF} -params {DELAY_TYPE:true}
```

6.6. FLASH_FREEZE (Ask a Question)

Description

"FLASH_FREEZE" is a command tool used in configure_tool. Use the "configure_tool -name {FLASH_FREEZE}" command to specify:

- The state of the uRAM and LSRAM when the FPGA fabric is in the Flash Freeze state.

```
configure_tool -name {FLASH_FREEZE} \
               -params {name:value} \
               -params {name:value}
```

- The MSS clock source when the FPGA fabric is in the Flash Freeze state.

```
configure_tool -name {FLASH_FREEZE} -params {parameter:value}
```

Arguments

Parameter	Type	Description
FF_RAM_STATE	string	Specifies the uRAM and LSRAM state during Flash Freeze. The possible for this argument are: • SUSPEND - Sets to Suspend; LSRAM and uSRAM contents are retained. By default is SUSPEND. • SLEEP - Sets to Sleep; LSRAM and uSRAM contents are not retained.
FF_MSS_CLOCK	string	Specifies the MSS Clock Source during Flash Freeze. The possible value for this argument are: • RCOSC_1MHZ - On-Chip 1 MHz RC Oscillator. By default is RCOSC_1MHZ. • RCOSC_50MHZ - On-Chip 50 MHz RC Oscillator.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.

Supported Families

Supported Families
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example sets uRAM and LSRAM state as "SUSPEND" and MSS Clock On-Chip 50 MHz RC Oscillator:

```
configure_tool -name {FLASH_FREEZE} \
               -params {FF_MSS_CLOCK:RCOSC_50MHZ} \
               -params {FF_RAM_STATE:SLEEP}
```

6.7. GENERATEDEBUGDATA (Ask a Question)

Description

"GENERATEDEBUGDATA" a command tool used in the run_tool command. This Tcl command generates the files needed by SmartDebug during device debug.

```
run_tool -name {GENERATEDEBUGDATA}
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Generate Debug Data:

```
run_tool -name {GENERATEDEBUGDATA}
```

See Also

- export_smart_debug_data

6.8. GENERATEPROGRAMMINGDATA (Ask a Question)

Description

"GENERATEPROGRAMMINGDATA" is the name of a command tool used in the run_tool command. The "run_tool -name {GENERATEPROGRAMMINGDATA}" Tcl command generates the files needed for generating programming bitstream files.

```
run_tool -name {GENERATEPROGRAMMINGDATA}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following command generats programming bitstream file:

```
run_tool -name {GENERATEPROGRAMMINGDATA}
```

6.9. GENERATEPROGRAMMINGFILE (Ask a Question)

Description

"GENERATEPROGRAMMINGFILE" is a command tool used in the configure_tool and run_tool commands. The `configure_tool -name {GENERATEPROGRAMMINGFILE}` Tcl command configures tool options. The `run_tool -name {GENERATEPROGRAMMINGFILE}` Tcl command runs the specified tool with the options specified in configure_tool.

Important: The tools have the new Tcl parameters sanitize_snvm (PolarFire, RT PolarFire, and PolarFire SoC) and sanitize_envm (SmartFusion 2, IGLOO 2, and PolarFire SoC) available from Libero v12.6.

The command usage for the PolarFire is the following:

```
configure_tool -name {GENERATEPROGRAMMINGFILE} \
               -params {program_fabric: true | false } \
               -params {program_security: true | false } \
               -params {program_snvm: true | false} \
               -params {sanitize_snvm: true | false}
run_tool -name {GENERATEPROGRAMMINGFILE}
```

The command usage for the RTG4 is the following:

```
configure_tool -name {GENERATEPROGRAMMINGFILE} \
               -params {program_fabric: true | false } \
               -params {program_security: true | false } \
               -params {program_mode:selected_features} \
               -params {program_envm: true | false} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                -params {sanitize_envm: true | false}
run_tool -name {GENERATEPROGRAMMINGFILE}
```

The command usage for the SmartFusion 2, IGLOO 2 is the following:

```
configure_tool -name {GENERATEPROGRAMMINGFILE} \
                -params {program_fabric: true | false } \
                -params {program_security: true | false } \
                -params {program_snvm: true | false} \
                -params {sanitize_snvm: true | false}
run_tool -name {GENERATEPROGRAMMINGFILE}
```

The command usage for the PolarFire SoC is the following:

```
configure_tool -name {GENERATEPROGRAMMINGFILE} \
                -params {program_fabric_snvm: true | false} \
                -params {program_security: true | false} \
                -params {program_envm: true | false} \
                -params {sanitize_envm: true | false} \
                -params {sanitize_snvm: true | false}
run_tool -name {GENERATEPROGRAMMINGFILE}
```

Arguments

The following table list the "GENERATEPROGRAMMINGFILE" arguments for PolarFire.

Parameter	Type	Description
program_fabric	boolean	Include fabric component in the programming bitstream. The acceptable values are: true, false.
program_security	boolean	Include custom security component in the programming bitstream ("true" only if custom security was defined). The acceptable values are: true, false.
program_snvm	boolean	Include sNVM component in the programming bitstream ("true" only if sNVM available in the design). The acceptable values are: true, false.
sanitize_snvm	boolean	Enable eNVM sanitization.

The following table list the "GENERATEPROGRAMMINGFILE" arguments for RTG4.

Parameter	Type	Description
program_fabric	boolean	Include fabric component in the programming bitstream. The acceptable values are: true, false.
program_mode	string	The possible value is "selected_features."
program_security	boolean	Include custom security component in the programming bitstream ("true" only if custom security was defined). The acceptable values are: true, false.
program_envm	boolean	Include eNVM component in the programming bitstream ("true" only if eNVM available in the design). The acceptable values are: true, false.
sanitize_envm	boolean	Enable eNVM sanitization.

The following table list the "GENERATEPROGRAMMINGFILE" arguments for SmartFusion 2, IGLOO 2.

Parameter	Type	Description
program_fabric	boolean	Include fabric component in the programming bitstream. The acceptable values are: true, false.
program_security	boolean	Include custom security component in the programming bitstream ("true" only if custom security was defined). The acceptable values are: true, false.
program_envm	boolean	Include eNVM component in the programming bitstream ("true" only if eNVM available in the design). The acceptable values are: true, false.
sanitize_snvm	boolean	Enable sNVM sanitization.

© 2025 Microchip Technology Inc. and its subsidiaries

The following table list the "GENERATEPROGRAMMINGFILE" arguments for PolarFire SoC.

Parameter	Type	Description
program_fabric_snvm	boolean	Include Fabric/sNVM component in the programming bitstream. The acceptable values are: true, false.
program_security	boolean	Include custom security component in the programming bitstream ("true" only if custom security was defined). The acceptable values are: true, false.
program_envm	boolean	Include eNVM component in the programming bitstream ("true" only if eNVM available in the design). The acceptable values are: true, false.
sanitize_envm	boolean	Enable eNVM sanitization.
sanitize_snvm	boolean	Enable sNVM sanitization.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	You must select at least one component to program.
None	Parameter program_snvm does not exist.
None	Parameter program_envm does not exist.
None	Fabric/sNVM is not selected for programming; sNVM sanitization is not available.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Configure "GENERATEPROGRAMMINGFILE" for PolarFire device:

```
configure_tool -name {GENERATEPROGRAMMINGFILE} \
               -params {program_fabric:false} \
               -params {program_security:true} \
               -params {program_snvm:true}
```

See Also

- GENERATEPROGRAMMINGDATA

6.10. INIT_LOCK (Ask a Question)

Description

"INIT_LOCK" is a TCL equivalent command name for the Configure Register Lock Bits tool. This command imports a Lock Bit Configuration File (*.txt) and configures the Register Lock Bits of FDDR, MSS, and SERDES blocks for SmartFusion 2 and IGLOO 2 devices so that they cannot be overwritten by Fabric or MSS Masters that have write access to these registers. This command takes only one parameter, INIT_LOCK_FILE.

```
configure_tool \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
    -name {INIT_LOCK} \
    -params {INIT_LOCK_FILE: <path to the config lock bit file>}
```

Arguments

Parameter	Type	Description
INIT_LOCK_FILE	string	Specifies full or relative path to the config lock bit file that contains the Register Lock Bit settings.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Register Lock Bit Text File Template

An initial Configuration Lock Bit file can be generated from the Design Flow window (Design Flow window > Generate FPGA Array Data) or via "run_tool -name {GENERATEPROGRAMMINGDATA}" TCL command. The file is named <proj_location>/designer/<root>/<root>_init_config_lock_bits.txt. This is the initial and the default Lock Bit Configuration File. Use this file as a template to make changes. Modify it to ensure that the lock bits are set to "0" for all register bits you want to lock. Save the file as a *.txt file with a different name and import the file into the project using the Register Lock Bit Settings dialog box (Design Flow window > Configure Register Lock Bits).

Register Lock Bit Text File Syntax

A valid entry in the Lock Bit Configuration file is defined as <lock_parameters> <lock bit value> pair.

- If the lock bit is for a register the parameter name is defined as:
  - <Physical block name>_<register name>_LOCK
- If the lock bit is for a field the parameter name is defined as:
  - <Physical block name>_<register name>_<field name>_LOCK
- The physical block name (which may vary with device family and die) is defined as:
  - MSS
  - FDDR
  - SERDES_IF_x (where x is 0,1,2,3 to indicate the physical SERDES location) for SmartFusion 2, and IGLOO 2 (010/025/050/150) devices
  - SERDES_IF2 (060/090) devices (only one SERDES block per device for SmartFusion 2 and IGLOO 2 devices)

Set the lock bit value to ‘1’ to indicate that the register can be written, “0” to indicate that the register cannot be written (locked). Lines starting with “#” or “;” are comments and empty lines are allowed in the Lock Bit Configuration file(example below).

```
# Register Lock Bits Configuration File for MSS, SERDES(s) and Fabric DDR
# Microchip Corporation - Microchip Libero Software Release v12.4 (Version 12.900.0.16)
# Date: Tue Jun 15 09:33:35 2021

# Lock Value = 0, disables modification of the Register field.


# HPMS/MDDR ( Unused )
 MSS_ESRAM_CONFIG_LOCK                              1
 MSS_ESRAM_MAX_LAT_LOCK                             1
 MSS_DDR_CONFIG_LOCK                                1
 MSS_ENVM_CONFIG_LOCK                               1
 MSS_ENVM_REMAP_BASE_LOCK                           1
 MSS_ENVM_FAB_REMAP_LOCK                            1
 MSS_CC_CONFIG_LOCK                                 1
 MSS_CC_CACHEREGION_LOCK                            1
 MSS_CC_LOCKBASEADDR_LOCK                           1
 MSS_CC_FLUSHINDX_LOCK                              1
```


© 2025 Microchip Technology Inc. and its subsidiaries

During Map File generation (Design Flow window > Generate FPGA Array Data), Libero SoC validates the Register Lock Bit Configuration file and displays error message when invalid parameter or parameter values are found.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.

Supported Families

SmartFusion® 2

IGLOO® 2

Example

Configure Register Lock Bits

```
# Full path
configure_tool -name {INIT_LOCK} \
               -params {INIT_LOCK_FILE:D:/designs/g4_fclk_mddr_clk/
sb_init_config_lock_bits_src.txt}
# Relative path from project folder
configure_tool -name {INIT_LOCK} \
               -params {INIT_LOCK_FILE:../sb_init_config_lock_bits_src.txt}
```

6.11. IO_PROGRAMMING_STATE (Ask a Question)

Description

"IO_PROGRAMMING_STATE" is a command tool used in the configure_tool Tcl command. The `configure_tool -name {IO_PROGRAMMING_STATE}` Tcl command loads the I/O State information from a file during programming. The file used for loading the I/O State information during programming is specified in a parameter to the command.

Note: I/O States During programming will be used during programming or when exporting the bitstream.

```
configure_tool \
    -name {IO_PROGRAMMING_STATE} \
    -params {ios_file:absolute_path_to_i/o_state_information_file(*.ios)}
```

Arguments

Parameter	Type	Description
ios_file	string	Specifies the path to the configuration file(*.ios) of the I/O States. It is mandatory.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.
None	Unable locate file '*.ios'

Supported Families

Supported Families
PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

IO_PROGRAMMING_STATE (continued)
Supported Families
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example configures IO States, loads States from prj_path/designer/top/top.ios configure file.

```
configure_tool -name {IO_PROGRAMMING_STATE} \
               -params {ios_file:/prj_path/designer/top/top.ios}
```

6.12. PLACEROUTE (Ask a Question)

Description

To place and route a design in Libero SoC, you must first configure the "PLACEROUTE" tool with the configure_tool command and then execute the "PLACEROUTE" tool with the run_tool command.

Note: At least one "parameter:value" must be specified. You can repeat `-params` argument for multiple parameters.

```
configure_tool -name {PLACEROUTE} -params {parameter:value}
run_tool -name {PLACEROUTE}
```

Arguments

The following table list the "PLACEROUTE" arguments for PolarFire.

Parameter	Type	Description
TDPR	boolean	Set to true or 1 to enable Timing-Driven Place and Route. Default is true or 1.
PDPR	boolean	Set to true or 1 to enable Power-Driven Place and Route. Default is false or 0.
IOREG_COMBINING	boolean	Set to true or 1 to enable I/O Register Combining. Default is false or 0.
GB_DEMOTION	boolean	Set to true or 1 to enable Global Pins Demotion. Default is true or 1.
REPLICATION	boolean	Set to true or 1 to enable Driver Replication. Default is false or 0.
EFFORT_LEVEL	boolean	Set to true or 1 to enable High Effort Layout to optimize design performance. Default is false or 0.
INCRPLACEANDROUTE	boolean	Set to true or 1 to use previous placement data as the initial placement for the next run. Default is false or 0. This parameter gets reset to false or 0 after each run of Place and Route to prevent inadvertent loss of performance across large design changes. Incremental Place and Route should be used in conjunction with Compile points synthesis.
REPAIR_MIN_DELAY	boolean	Set to True or 1 to enable Repair Minimum Delay violations. Default is false or 0.
MULTI_PASS_LAYOUT	boolean	Set to true or 1 to enable Multi-Pass Layout Mode for Place and Route. Default is false or 0.
NUM_MULTI_PASSES	integer	Specifies the positive number of passes to run. The default is 5.
START_SEED_INDEX	integer	Indicates the random seed index(positive integer) which is the starting point for the Multi-pass Layout. If not specified, the default behavior is to continue after the last seed index which was used.

© 2025 Microchip Technology Inc. and its subsidiaries

PLACEROUTE (continued)		
Parameter	**Type**	**Description**
MULTI_PASS_CRITERIA	string	Specifies the criteria used to run multi-pass layout. The acceptable values are the following: • SLOWEST_ClOCK: Use the slowest clock Fmax in the design in a given pass as the performance reference for the layout pass. • SPECIFIC_CLOCK: Use a specific clock performance as the reference for all layout passes. • VIOLATIONS: Apply SLACK_CRITERIA from the Timing violations report as the performance reference for all passes. This is the default criterion. • TOTAL_POWER: Specifies the best pass to be the one that has the lowest total power (static + dynamic) out of all layout passes.
SPECIFIC_CLOCK	string	Applies only when MULTI_PASS_CRITERIA is set to SPECIFIC_CLOCK. It specifies the name of the clock in the design used for performance measurement.
DELAY_ANALYSIS	string	Applies only when MULTI_PASS_CRITERIA is set to VIOLATIONS. Specifies the type of timing violations (slacks) to be examined. The acceptable values are the following: • max: Use timing violations (slacks) obtained from maximum delay analysis. This is the default. • min: Use timing violations (slacks) obtained from minimum delay analysis.
STOP_ON_FIRST_PASS	boolean	Applies only when MULTI_PASS_CRITERIA is set to VIOLATIONS. It stops performing remaining passes if all timing constraints are met (when there are no negative slacks reported in the timing violations report). **Note:** The type of timing violations (slacks) used is determined by the DELAY_ANALYSIS parameter.
SLACK_CRITERIA	string	Applies only when MULTI_PASS_CRITERIA is set to VIOLATIONS. Specifies how to evaluate the timing violations (slacks). The acceptable values are the following: • WORST_SLACK: The largest amount of negative slack (or least amount of positive slack if all constraints are met) for each pass is identified and then the largest value out of all passes will determine the best pass. This is the default value. • TOTAL_NEGATIVE_SLACK: The sum of negative slacks from the first 100 paths for each pass in the Timing Violation report is identified. The largest value out of all passes will determine the best pass. If no negative slacks exist for a pass, then use the worst slack to evaluate that pass. **Note:** The type of timing violations (slacks) used is determined by the DELAY_ANALYSIS parameter.
RGB_COUNT	integer	Allows an entity to override the placer's RGB/RCLK bandwidth constraint. This option is useful for Block Creation. The possible values for this parameter are 1-18.
RANDOM_SEED	integer	Sets the actual seed non-negative integer value for next Place and Route run. When MULTI_PASS_LAYOUT is True or 1, the START_SEED_INDEX takes precedence over RANDOM_SEED. It overrides RANDOM_SEED for the first run to $START_SEED_INDEX-1.

The following table list the "PLACEROUTE" arguments for SmartFusion 2, IGLOO 2 and RTG4.

Parameter	**Type**	**Description**
TDPR	boolean	Set to true or 1 to enable Timing-Driven Place and Route. Default is true or 1.

© 2025 Microchip Technology Inc. and its subsidiaries

PLACEROUTE (continued)		
Parameter	**Type**	**Description**
PDPR	boolean	Set to true or 1 to enable Power-Driven Place and Route. Default is false or 0.
IOREG_COMBINING	boolean	Set to true or 1 to enable I/O Register Combining. Default is false or 0.
EFFORT_LEVEL	boolean	Set to true or 1 to enable High Effort Layout to optimize design performance. Default is false or 0.
INCRPLACEANDROUTE	boolean	Set to true or 1 to use previous placement data as the initial placement for the next run. Default is false or 0.
REPAIR_MIN_DELAY	boolean	Set to True or 1 to enable Repair Minimum Delay violations for the routher when TDPR option is set to true or 1. Default is false or 0.
NUM_MULTI_PASSES	integer	Specifies the positive number of passes to run. The default is 5. Maximum is 25.
START_SEED_INDEX	integer	Indicates the random seed index(positive integer) which is the starting point for the passes. Its value should range from 1 to 100. If not specified, the default behavior is to continue from the last seed index which was used.
MULTI_PASS_LAYOUT	boolean	Set to true or 1 to enable Multi-Pass Layout Mode for Place and Route. Default is false or 0.
MULTI_PASS_CRITERIA	string	Specifies the criteria used to run multi-pass layout. The acceptable values are the following: • SLOWEST_ClOCK: Use the slowest clock frequency in the design in a given pass as the performance reference for the layout pass. • SPECIFIC_CLOCK: Use a specific clock frequency as the performance reference for all layout passes. • VIOLATIONS: Use the pass that best meets the slack or timing-violations constraints. This is the default. • TOTAL_POWER: Specifies the best pass to be the one that has the lowest total power (static + dynamic) out of all layout passes.
SPECIFIC_CLOCK	string	Applies only when MULTI_PASS_CRITERIA is set to SPECIFIC_CLOCK. It specifies the name of the clock in the design used for Timing Violation Measurement.
DELAY_ANALYSIS	string	Used only when MULTI_PASS_CRITERIA is set to “VIOLATIONS”. Specifies the type of timing violations (slacks) to be examined. The default is "max." • max: Use timing violations (slacks) obtained from maximum delay analysis. • min: Use timing violations (slacks) obtained from minimum delay analysis.
STOP_ON_FIRST_PASS	boolean	Applies only when "MULTI_PASS_CRITERIA" is set to "VIOLATIONS." It stops performing remaining passes if all timing constraints are met (when there are no negative slacks reported in the timing violations report). **Note:** The type of timing violations (slacks) used is determined by the "DELAY_ANALYSIS" parameter.

© 2025 Microchip Technology Inc. and its subsidiaries

PLACEROUTE (continued)		
Parameter	**Type**	**Description**
SLACK_CRITERIA	string	Applies only when "MULTI_PASS_CRITERIA" is set to "VIOLATIONS." Specifies how to evaluate the timing violations (slacks). The acceptable values are the following: • WORST_SLACK: The largest amount of negative slack (or least amount of positive slack if all constraints are met) for each pass is identified and then the largest value out of all passes will determine the best pass. This is the default value. • TOTAL_NEGATIVE_SLACK: The sum of negative slacks from the first 100 paths for each pass in the Timing Violation report is identified. The largest value out of all passes will determine the best pass. If no negative slacks exist for a pass, then use the worst slack to evaluate that pass. **Note:** The type of timing violations (slacks) used is determined by the "DELAY_ANALYSIS" parameter.
RGB_COUNT	integer	Allows an entity to override the placer's RGB/RCLK bandwidth constraint. This option is useful for Block Creation. The possible values for this parameter are 1-18.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	EFFORT_LEVEL is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	INCRPLACEANDROUTE is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MULTI_PASS_CRITERIA is set to an illegal value 'value'. Legal values are SLOWEST_CLOCK, SPECIFIC_CLOCK, VIOLATIONS and TOTAL_POWER.
None	IOREG_COMBINING is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	PDPR is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	REPAIR_MIN_DELAY is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	SLACK_CRITERIA is set to an illegal value 'value'. Legal values are WORST_SLACK and TOTAL_NEGATIVE_SLACK.
None	START_SEED_INDEX is set to an illegal value 'value'. Legal values must be between 1 and 101.
None	STOP_ON_FIRST_PASS is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	TDPR is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	GB_DEMOTION is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	REPLICATION is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MULTI_PASS_LAYOUT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	DELAY_ANALYSIS is set to an illegal value 'value'. Legal values are MAX and MIN.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

PLACEROUTE (continued)
Supported Families
SmartFusion® 2
IGLOO® 2

Example

The following example configures "PLACEROUTE" for PolarFire:

```
configure_tool -name {PLACEROUTE} \
               -params {TDPR:true} \
               -params {EFFORT_LEVEL:true} \
               -params {GB_DEMOTION:true} \
               -params {INCRPLACEANDROUTE:false} \
               -params {IOREG_COMBINING:false} \
               -params {PDPR:false} \
               -params {REPAIR_MIN_DELAY:true} \
               -params {REPLICATION:false} \
               -params {MULTI_PASS_LAYOUT:true} \
               -params {MULTI_PASS_CRITERIA:VIOLATIONS} \
               -params {NUM_MULTI_PASSES:5} \
               -params {SLACK_CRITERIA:WORST_SLACK} \
               -params {START_SEED_INDEX:1} \
               -params {STOP_ON_FIRST_PASS:false}
run_tool -name {PLACEROUTE}
```

The following example configures "PLACEROUTE" for SmartFusion 2:

```
configure_tool -name {PLACEROUTE} \
               -params {EFFORT_LEVEL:true} \
               -params {INCRPLACEANDROUTE:false} \
               -params {IOREG_COMBINING:false} \
               -params {MULTI_PASS_CRITERIA:VIOLATIONS} \
               -params {MULTI_PASS_LAYOUT:false} \
               -params {NUM_MULTI_PASSES:5} \
               -params {PDPR:false} \
               -params {REPAIR_MIN_DELAY:true} \
               -params {SLACK_CRITERIA:WORST_SLACK} \
               -params {SPECIFIC_CLOCK:} \
               -params {START_SEED_INDEX:1} \
               -params {STOP_ON_FIRST_PASS:false} \
               -params {TDPR:true}
```

6.13. PROGRAMDEVICE (Ask a Question)

Description

"PROGRAMDEVICE" is a command tool used in configure_tool and run_tool. Configure_tool allows you to configure the tool's parameters and values prior to executing the tool. "run_tool" executes the tool with the configured parameters. To program the design in Libero SoC, you must first configure the "PROGRAMDEVICE" tool with configure_tool command and then execute the "PROGRAMDEVICE" command with the run_tool command. Use the commands to configure the programming action and the programming procedures associated with the program action.

```
configure_tool -name {PROGRAMDEVICE} \
               -params {prog_action: params_value}
run_tool -name {PROGRAMDEVICE}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
prog_action	string	The acceptable values for this argument are the following: • PROGRAM - Programs all selected family features: FPGA Array, targeted eNVM clients and security settings. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_PROGRAM_ACTIONTYPE, VERIFY_IDCODE, PROC_ENABLE, DO_PROGRAM, DO_VERIFY (optional) and DO_EXIT. • VERIFY - Verifies all selected family features: FPGA Array, targeted eNVM clients and security settings. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_VERIFY_ACTIONTYPE, VERIFY_IDCODE, PROC_ENABLE, DO_VERIFY and DO_EXIT. • ERASE - Erases the selected family features: FPGA Array and security settings. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_ERASE_ACTIONTYPE, VERIFY_IDCODE, PROC_ENABLE, DO ERASE and DO_EXIT. • DEVICE_INFO - Displays the IDCODE, the design name, the checksum, and device security settings and programming environment information programmed into the device. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_DEVICE_INFO_ACTIONTYPE, VERIFY_IDCODE, DO_DEVICE_INFO and DO_EXIT. • READ_IDCODE - Reads the device ID code from the device. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_READ_IDCODE, VERIFY_IDCODE, PRINT_IDCODE and DO_EXIT. • ENC_DATA_AUTHENTICATION - Encrypted bitstream authentication data. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, SET_AUTHORIZATION_ACTIONTYPE, VERIFY_IDCODE, DO_AUTHENTICATION and DO_EXIT. • VERIFY_DIGEST - Calculates the digests for the components included in the bitstream and compares them against the programmed values. Mandatory procedures are: INIT_VARIABLES_FOR_ACTION, VERIFY_IDCODE, PROC_ENABLE, DO_ENABLE_FABRIC (recommended), DO_ENABLE_SECURITY (recommended), DO_VERIFY_DIGEST (recommended) and DO_EXIT. • READ_DEVICE_CERTIFICATE - Reads the device certificate from the device.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example configures "PROGRAMDEVICE" tool to display the IDCODE, design name, device security settings and calculates the digests:

```
configure_tool -name {PROGRAMDEVICE} \
               -params {prog_action:VERIFY_DIGEST} \
               -params {prog_action:DEVICE_INFO}
run_tool -name {PROGRAMDEVICE} #run_tool takes no parameters
```

6.14. PROGRAMMER_INFO (Ask a Question)

Description

"PROGRAMMER_INFO" is a command tool used in configure_tool. The `configure_tool -name {PROGRAMMER_INFO}` Tcl command sets the programmer settings, similar to the way FlashPro commands set the programmer settings. For the JTAG interface, you can set specific voltage and force TCK frequency values for your programmer in this dialog. For the SPI Target interface, you can set specific voltage and force SCK frequency values for your programmer in this dialog.

Note: This command supports the FlashPro3, FlashPro4, FlashPro5, and FlashPro6 programmers.

```
configure_tool -name {PROGRAMMER_INFO} -params {name:value}
```

Arguments

The following table list the FlashPro3 arguments.

Attention: When running tcl command with parameters `{flashpro3_clk_mode:discrete_clk}` and/or `{flashpro4_clk_mode:discrete_clk}`, a warning message is added to the log which states that using the Discrete Clocking TCK mode will increase the programming time significantly.

Parameter	Type	Description
flashpro3_clk_mode	string	Specifies free-running or discrete TCK. The possible value for this argument are: free_running_clk or discrete_clocking. Default TCK mode setting is Free running clock.
flashpro3_force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. The possible value for this argument are: ON or OFF. Default is OFF.
flashpro3_ freq	integer	Specifies the TCK frequency in MHz. It can be between 1MHz to 6MHz.
flashpro3_vpump	string	The Vpump option is checked to instruct the FlashPro3 programmer to supply Vpump to the device. The possible value for this argument are: ON or OFF. Default is ON.

The following table list the FlashPro4 arguments.

Parameter	Type	Description
flashpro4_clk_mode	string	Specifies free-running or discrete TCK. The possible value for this argument are: free_running_clk or discrete_clocking. Default TCK mode setting is Free running clock.
flashpro4_force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. The possible value for this argument are: ON or OFF. Default is OFF.
flashpro4_ freq	integer	Specifies the TCK frequency in MHz. It can be between 1MHz to 6MHz.

© 2025 Microchip Technology Inc. and its subsidiaries

PROGRAMMER_INFO (continued)		
Parameter	**Type**	**Description**
flashpro4_vpump	string	The Vpump option is checked to instruct the FlashPro4 programmer to supply Vpump to the device. The possible value for this argument are: ON or OFF. Default is ON.

The following table list the FlashPro5 arguments.

Parameter	Type	Description
flashpro5_clk_mode	string	Specifies free-running or discrete TCK. The possible value for this argument are: free_running_clk or discrete_clocking. Default TCK mode setting is Free running clock.
flashpro5_force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. The possible value for this argument are: ON or OFF. Default is OFF.
flashpro5_ freq	integer	Specifies the TCK frequency in MHz. It can be between 1MHz to 6MHz or 10, 16, 30 MHz.
flashpro5_vpump	string	The Vpump option is checked to instruct the FlashPro5 programmer to supply Vpump to the device. The possible value for this argument are: ON or OFF. Default is ON.

The following table list the FlashPro6 arguments.

Parameter	Type	Description
flashpro6_force_sck_freq	string	Forces the FlashPro software to use the SCK frequency. Valid values are ON, OFF(default).
flashpro6_force_tck_freq	string	Forces the FlashPro software to use the TCK frequency. Valid values are ON, OFF(default).
flashpro6_sck_freq	integer	Specifies the SCK frequency in MHz. SCK is used with a maximum frequency of 40 MHz, and the default frequency is 20 MHz. Limitation of the SCK frequency for the selected programmer: 1.00, 2.00, 2.50, 3.33, 4.00, 5.00, 6.67, 8.00, 10.00, 13.33, 20.00(default), 40.00 MHz.
flashpro6_tck_freq	integer	Specifies the TCK frequency in MHz. It can be between 1MHz to 6MHz or 10, 16, 30 MHz.
flashpro6_vpump	string	The Vpump option is checked to instruct the FlashPro6 programmer to supply Vpump to the device. The possible value for this argument are: ON or OFF. Default is ON.

The following table list the FlashPro programmer arguments.

Parameter	Type	Description
flashpro_drive_trst	string	Valid values are ON, OFF(default). This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_force_freq	string	Valid values are ON, OFF(default). This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_force_vddp	string	The possible value for this argument are ON(default), OFF. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_freq	integer	Specifies the TCK frequency in MHz. It can be between 1MHz to 6MHz or 10, 16, 30 MHz. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_vddl	string	The possible value for this argument are: ON or OFF. Default is ON. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.

© 2025 Microchip Technology Inc. and its subsidiaries

PROGRAMMER_INFO (continued)		
Parameter	**Type**	**Description**
flashpro_vddp	integer	The possible value for this argument are: 2.5V or 3.3V. Default is 2.5V. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_vpn	string	The possible value for this argument are: ON or OFF. Default is ON. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.
flashpro_vpp	string	The possible value for this argument are: ON or OFF. Default is ON. This parameter supported by FlashPro Programmer for PolarFire, SmartFusion 2, IGLOO 2 and RTG4 devices.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Illegal Vddp value V for FlashPro. Expected: 2.5V or 3.3V.
None	Invalid Clock Mode for FlashPro3. Expected: 'free_running_clk' or 'discrete_clk'.
None	Invalid Clock Mode for FlashPro4. Expected: 'free_running_clk' or 'discrete_clk'.
None	Invalid Clock Mode for FlashPro5. FlashPro5 programmer supports 'free_running_clk' clock mode only.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

For FlashPro3 programmer:

```
configure_tool -name {PROGRAMMER_INFO} \
               -params {flashpro3_clk_mode:free_running_clk} \
               -params {flashpro3_force_freq:OFF} \
               -params {flashpro3_freq:400000} \
               -params {flashpro3_vpump:ON}
```

For FlashPro4 programmer:

```
configure_tool -name {PROGRAMMER_INFO} \
               -params {flashpro4_clk_mode:free_running_clk} \
               -params {flashpro4_force_freq:OFF} \
               -params {flashpro4_freq:400000} \
               -params {flashpro4_vpump:ON}
```

For FlashPro4 programmer:

```
configure_tool -name {PROGRAMMER_INFO} \
               -params {flashpro5_clk_mode:free_running_clk} \
               -params {flashpro5_force_freq:OFF} \
               -params {flashpro5_freq:400000} \
               -params {flashpro5_vpump:ON}
```


© 2025 Microchip Technology Inc. and its subsidiaries

For FlashPro6 programmer:

```
configure_tool -name {PROGRAMMER_INFO} \
               -params {flashpro6_force_freq:OFF} \
            -params {flashpro6_freq:400000}
```

See Also

- configure_flashpro3_prg
- configure_flashpro4_prg
- configure_flashpro5_prg
- configure_flashpro6_prg

6.15. PROGRAM_SPI_FLASH_IMAGE (Ask a Question)

Description

"PROGRAM_SPI_FLASH_IMAGE" is a command tool used in configure_tool and run_tool to program SPI Flash Image with configured parameters.

Important: If the Device ID does not match when running any action, the action will fail.

```
configure_tool -name {PROGRAM_SPI_FLASH_IMAGE} \
               -params {spi_flash_prog_action:procedure}
run_tool -name {PROGRAM_SPI_FLASH_IMAGE}
```

Arguments

Parameter	Type	Description
spi_flash_prog_action	string	The acceptable values for this argument are the following: • ERASE_SPI_FLASH - This action erases the entire SPI Flash. Mandatory procedures are: VERIFY_DEVICE_ID and ERASE_DIE. **Note:** Partial programming of the SPI flash is supported when using PolarFire / PolarFire SoC FPGAs and the FlashPro6 programmer. In that case, the ERASE_SPI_FLASH action allows you to erase only selected clients. • PROGRAM_SPI_IMAGE - This action will erase the entire SPI flash then program the SPI image. Mandatory procedures are: VERIFY_DEVICE_ID, ERASE_DIE and PROGRAM_IMAGE. • READ_SPI_IMAGE - This action reads the SPI Image from the SPI Flash. Mandatory procedures are: VERIFY_DEVICE_ID and READ_IMAGE. • VERIFY_SPI_IMAGE - This action verifies the SPI Image on the SPI Flash. Mandatory procedures are: VERIFY_DEVICE_ID and VERIFY_IMAGE.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Illegal SPI Flash programming action: action_name.

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC

Example

The following example configures "PROGRAM_SPI_FLASH_IMAGE" tool and runs:

```
configure_tool -name {PROGRAM_SPI_FLASH_IMAGE} \
               -params {spi_flash_prog_action: PROGRAM_SPI_FLASH}
run_tool -name {PROGRAM_SPI_FLASH_IMAGE}
```

6.16. SIM_PRESYNTH (Ask a Question)

Description

"SIM_PRESYNTH" is a command tool used in the run_tool command. The "run_tool -name {SIM_PRESYNTH}" Tcl command runs the simulator for pre-synthesis simulation on the simulation tool.

Important: Before running simulation, you must associate a testbench. If you attempt to run simulation without an associated testbench, the Libero SoC Project Manager asks you to associate a testbench or open the simulator without a testbench.

```
run_tool -name {SIM_PRESYNTH}
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example runs the pre-synthesis simulation tool:

```
run_tool -name {SIM_PRESYNTH}
```

See Also

- SIM_POSTSYNTH
- SIM_POSTLAYOUT

© 2025 Microchip Technology Inc. and its subsidiaries

- associate_stimulus
- organize_tool_files

6.17. SIM_POSTSYNTH (Ask a Question)

Description

"SIM_POSTSYNTH" is a command tool used in the run_tool command. The "run_tool -name {SIM_POSTSYNTH}" Tcl command runs the post-synthesis simulation on the simulation tool.

Important: Before running simulation, you must associate a testbench. If you attempt to run simulation without an associated testbench, the Libero SoC Project Manager asks you to associate a testbench or open the simulator without a testbench and then run "SYNTHESIZE" tool.

```
run_tool -name {SIM_POSTSYNTH}
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example runs "SYNTHESIS" tool then the post-synthesis simulation tool:

```
run_tool -name {SYNTHESIZE}
run_tool -name {SIM_POSTSYNTH}
```

See Also

- SIM_PRESYNTH
- SIM_POSTLAYOUT
- associate_stimulus
- organize_tool_files

© 2025 Microchip Technology Inc. and its subsidiaries

6.18. SIM_POSTLAYOUT (Ask a Question)

Description

"SIM_POSTLAYOUT" is a command tool used in the run_tool command. The "run_tool -name {SIM_POSTLAYOUT}" Tcl command runs the post layout simulation on the simulation tool.

Important: Before running simulation, you must associate a testbench. If you attempt to run simulation without an associated testbench, the Libero SoC Project Manager asks you to associate a testbench or open the simulator without a testbench.

```
run_tool -name {SIM_POSTLAYOUT}
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example runs the post layout simulation tool:

```
run_tool -name {SIM_POSTLAYOUT}
```

See Also

- SIM_PRESYNTH
- SIM_PRESYNTH
- associate_stimulus
- organize_tool_files

6.19. SPM (Ask a Question)

Description

"SPM" is a command tool used in configure_tool. To configure security using Tcl, you must use the configure_tool Tcl command to pass the SPM configuration parameters.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: At least one "parameter:value" must be specified. You can repeat `-params` argument for multiple parameters.

```
configure_tool -name {SPM} -params {parameter:value}
```

Arguments

The following table list the "SPM" arguments for PolarFire.

Parameter	Type	Description
back_level_protection	boolean	The possible value for this argument are: true, 1 , false or 0. Specify true or 1 to set back level protection; Update Policy.
debug_passkey	hexadecimal	Specify value of DPK, value must be 64 hex characters; Debug Policy.
disable_authenticate_action	boolean	Disables Authenticate action. The possible value for this argument are: true, 1, false or 0.
disable_autoprog_iap_services	boolean	Disables Auto Programming and IAP Services. The possible value for this argument are: true, 1, false or 0.
disable_debug_jtag_boundary_scan	boolean	Disables debug JTAG Boundary Scan. The possible value for this argument are: true, 1, false or 0.
disable_debug_read_temp_volt	boolean	Disables reading temperature and voltage sensor (JTAG/SPI Slave). The possible value for this argument are: true, 1, false or 0.
disable_debug_ujtag	boolean	Disables debug; UJTAG. The possible value for this argument are: true, 1, false or 0.
disable_ext_zeroization	boolean	Disables external zeroization through JTAG/SPI Slave. The possible value for this argument are: true, 1, false or 0.
disable_external_digest_check	boolean	Disables external Fabric/sNVM digest requests through JTAG/SPI Slave. The possible value for this argument are: true, 1, false or 0.
disable_jtag	boolean	Disables JTAG. The possible value for this argument are: true, 1, false or 0.
disable_program_action	boolean	Disables Program action. The possible value for this argument are: true, 1, false or 0.
disable_puf_emulation	boolean	Disables external access to PUF emulation through JTAG/SPI Slave The possible value for this argument are: true, 1, false or 0.
disable_smartdebug_debug	boolean	Disables user debug access and active probes. The possible value for this argument are: true, 1, false or 0.
disable_smartdebug_live_probe	boolean	Disables SmartDebug Live Probe. The possible value for this argument are: true, 1, false or 0.
disable_smartdebug_snvm	boolean	Disables SmartDebug sNVM. The possible value for this argument are: true, 1, false or 0.
disable_spi_slave	boolean	Disables SPI Slave interface. The possible value for this argument are: true, 1, false or 0.
disable_user_encryption_key_1	boolean	Disables UEK1; Key Mode Policy. The possible value for this argument are: true, 1, false or 0.
disable_user_encryption_key_2	boolean	Disables UEK2; Key Mode Policy. The possible value for this argument are: true, 1, false or 0.
disable_verify_action	boolean	Disables Verify action. The possible value for this argument are: true, 1, false or 0.
fabric_update_protection	string	Fabric update protection. The possible values for this argument are the following: • open - updates allowed using user defined encryption keys. • disabled - disables Erase/Write operations.
security_factory_access	string	Microchip factory test mode access. The possible values for this argument are the following: • open - factory test mode access allowed. • disabled - disables factory test mode access.

© 2025 Microchip Technology Inc. and its subsidiaries

SPM (continued)		
Parameter	**Type**	**Description**
security_key_mode	string	Key mode access. The possible values for this argument are the following: • custom - custom security options. • default - bit stream encryption with default key.
snvm_update_protection	string	sNVM update protection. The possible values for this argument are the following: • open - updates allowed using user defined encryption keys. • disable - disables Write operations.
user_encryption_key_1	hexadecimal	Specify value of UEK1, value must be 64 hex characters.
user_encryption_key_2	hexadecimal	Specify value of UEK2, value must be 64 hex characters.
user_passkey_1	hexadecimal	Specify value of Flashlock/UPK1, value must be 64 hex characters.
user_passkey_2	hexadecimal	Specify value of Flashlock/UPK2, value must be 64 hex characters.

The following table list the "SPM" arguments for SmartFusion 2 and IGLOO 2.

Parameter	Type	Description
back_level_bypass	boolean	The possible values for this argument are: true, 1 , false or 0. Specify true or 1 to bypass the back level protection; Update Policy.
back_level_protection	boolean	The possible values for this argument are: true, 1 , false or 0. Specify true or 1 to set back level protection; Update Policy.
back_level_update_version	integer	Specify back level version value between 0 and 65535; Update Policy.
debug_cortex_m3	boolean	The possible values for this argument are: true, 1 , false or 0. Specify true or 1 to disable Cortex M3 debug. This lock bit is protected by DPK; Debug Policy; SmartFusion 2 only.
debug_digest_request	boolean	The possible value for this argument are: true, 1 , false or 0. Specify true or 1 to disable design digest check request via JTAG and SPI. Use FlashLock/UPK1 to allow digest check; Debug Policy.
debug_disable_jtag	boolean	The possible values for this argument are: true, 1 , false or 0. Specify true or 1 to disable JTAG (1149.1) test instructions (HIGHZ, EXTEST, INTEST, CLAMP, SAMPLE, and PRELOAD). I/Os will be tri-stated when in JTAG programming mode. Use FlashLock/UPK1 to unlock; Debug Policy.
debug_passkey	hexadecimal	Specify value of DPK, value must be 64 hex characters; Debug Policy.
debug_ujtag_access	boolean	The possible values for this argument are: true, 1 , false or 0. Specify true or 1 to disable access to UJTAG. Use DPK to unlock; Debug Policy.
disable_user_encryption_key_1	boolean	The possible value for this argument are: true, 1 , false or 0. Specify true or 1 to disable UEK1; Key Mode Policy.
disable_user_encryption_key_2	boolean	The possible value for this argument are: true, 1 , false or 0. Specify true or 1 to disable UEK2; Key Mode Policy.
disable_user_encryption_key_3	boolean	Disables UEK3; Key Mode Policy. The possible values for this argument are: true, 1, false or 0. **Note:** UEK3 is only available for M2S060, M2GL060, M2S090, M2GL090, M2S150, and M2GL150 devices. All other devices will set this to false by default.
factory_access	string	Sets Microchip factory test mode access level. The possible values for this argument are the following: • Open - All Microchip factory test mode access without FlashLock/UPK1. • FlashLock(default) - Microchip factory test mode is disabled. FlashLock/UPK1 is required to unlock. • Permanent - Permanently disable Microchip factory test mode access

© 2025 Microchip Technology Inc. and its subsidiaries

SPM (continued)		
Parameter	**Type**	**Description**
iap_isp_services	boolean	The possible value for this argument are: true, 1, false or 0. Specify true or 1 to disable access to IAP/ISP services; Update Policy.
security_key_mode	string	Key mode access. The possible values for this argument are the following: • Custom - Custom security settings. Allows user encryption keys, security policy settings, and Microchip factory test mode access level. • Default - Bitstream encryption with default key. No security lock bits are set.
smartdebug_access	string	Debug Policy. The possible values for this argument are the following: • Full - SmartDebug has full access to debug features. • None - Disable read/write access to SmartDebug architecture. DPK is required for read/write access.
update_auto_prog_lock	boolean	Disables Auto Programming; Update Policy. The possible value for this argument are: true, 1, false or 0.
update_envm_protection	string	Update Policy. The possible values for this argument are the following: • Passkey - eNVM updates are disabled. Use FlashLock/UPK1 to unlock Write/Verify/Read operations. • Open - Updates to eNVM are allowed using UEK1 or UEK2; FlashLock/UPK1 is NOT required for updates.
update_fabric_protection	string	Update Policy. The possible values for this argument are the following: • Passkey - Fabric updates are disabled. Use FlashLock/UPK1 to unlock Erase/Write/Verify/ operations. • Open - Updates to Fabric are allowed using UEK1 or UEK2; FlashLock/UPK1 is NOT required for updates.
update_jtag_lock	boolean	Disables access to JTAG programming. Use FLashLock/UPK1 to unlock; Update Policy. The possible value for this argument are: true, 1, false or 0.
update_spi_slave_lock	boolean	Disables access to SPI Slave. Use FLashLock/UPK1 to unlock; Update Policy. The possible value for this argument are: true, 1, false or 0.
use_debug_policy	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to used Debug Policy.
use_key_mode_policy	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to used Key Mode Policy.
use_update_policy	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to used Update Policy.
use_user_key_set_1	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to enable User Key Set 1.
use_user_key_set_2	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to enable User Key Set 2.
use_user_key_set_3	boolean	The possible values for this argument are: true, 1, false or 0. Specify true or 1 to enable User Key Set 3. **Note:** User Key Set 3 is only available for M2S060, M2GL060, M2S090, M2GL090, M2S150, and M2GL150 devices.
user_encryption_key_1	hexadecimal	Specify value of UEK1, value must be 64 hex characters.
user_encryption_key_2	hexadecimal	Specify value of UEK2, value must be 64 hex characters.
user_encryption_key_3	hexadecimal	Specify value of UEK3, value must be 64 hex characters. **Note:** UEK3 is only available for M2S060, M2GL060, M2S090, M2GL090, M2S150, and M2GL150 devices. All other devices will set this to false by default.
user_passkey_1	hexadecimal	Specify value of Flashlock/UPK1, value must be 64 hex characters.

© 2025 Microchip Technology Inc. and its subsidiaries

SPM (continued)		
Parameter	**Type**	**Description**
user_passkey_2	hexadecimal	Specify value of UPK2, value must be 64 hex characters.
user_security_policy_protection	string	The possible values for this argument are the following: • FlashLock - User keys and Security policies will be protected from erase/write by FlashLock/UPK1. • Permanent - Permanently protect UEK1, UEK2, Security Policies, and Microchip factory test mode access level. **Note:** Once programmed, these settings cannot be changed.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.
None	Key size is incorrect.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2

Example

The following example configures SPM for PolarFire:

```
configure_tool \
         -name {SPM} \
         -params {back_level_protection:false} \
         -params {disable_smartdebug_live_probe:false} \
         -params {disable_smartdebug_snvm:false} \
         -params {disable_user_encryption_key_1:false} \
         -params {disable_user_encryption_key_2:false}
```

The following example configures SPM for SmartFusion 2:

```
configure_tool \
         -name {SPM} \
         -params {back_level_bypass:false} \
         -params {back_level_protection:false} \
         -params {back_level_update_version: 32} \
         -params {debug_cortex_m3:false} \
         -params {debug_digest_request:false} \
         -params {debug_disable_jtag:false} \
         -params
{debug_passkey:8A1081239567235A7453336CFBBC45668754SADDCAFA7010FA209F7396F3EA17} \
         -params {debug_ujtag_access:false} \
         -params {disable_user_encryption_key_1:false} \
         -params {disable_user_encryption_key_2:false} \
         -params {disable_user_encryption_key_3:false} \
         -params {factory_access:flashlock} \
         -params {iap_isp_services:true} \
         -params {security_key_mode:custom} \
         -params {smartdebug_access:full} \
         -params {update_auto_prog_lock:true} \
         -params {update_envm_protection:passkey} \
         -params {update_fabric_protection:passkey} \
         -params {update_jtag_lock:false} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
          -params {update_spi_slave_lock:false} \
          -params {use_debug_policy:false} \
          -params {use_key_mode_policy:false} \
          -params {use_update_policy:false} \
          -params {use_user_key_set_1:true} \
          -params {use_user_key_set_2:false} \
          -params {use_user_key_set_3:false} \
          -params
{user_encryption_key_1:9E108123949848EC7453336DFBBC0CAE60C8541C2AFA7010FA209F7396F3EA17} \
          -params
{user_encryption_key_2:4D5656BA56541156C54E54563D2114BC45C854B456563010FA265F7396F3EA17} \
          -params
{user_encryption_key_3:CA5665B39498DAEC745355BDFB89535BA4A45DFC2AFA7010FA209F7396F3EA17} \
          -params
{user_passkey_1:252BED2AB1C4C5BAE13C4791CEDF7A069D940A6935629A0A9CE5B24E21C13D39} \
          -params
{user_passkey_2:B59EAD2356B66DAAE1654981BEA57A045653231CA5547A0A99AD254E234BCA39} \
          -params {user_security_policy_protection:flashlock}
```

6.20. SPM_OTP (Ask a Question)

Description

SPM_OTP is a command tool used in configure_tool to pass the SPM configuration parameters.

Important: At least one "parameter:value" must be specified. You can repeat <params> argument for multiple parameters.

```
configure_tool -name {SPM_OTP} -params {parameter:value}
```

Arguments

Parameter	Type	Description
permanently_disable_debugging	boolean	Specifies that the SmartDebug access control, and reading temperature, and voltage sensor settings is either permanently enabled or disabled. A value of true or 1 will permanently disable debugging. The default value is false.
permanently_disable_dpk	boolean	Specifies that the Debug Pass Key is either permanently enabled or disabled. A value of true or 1 will permanently disable FlashLock DPK unlocking. The default value is false.
permanently_disable_factory_access	boolean	Specifies that the access policy for Microchip factory test mode is either permanently enabled or disabled. A value of true or 1 will permanently disable Microchip factory test mode. The default value is false.
permanently_disable_prog_interfaces	boolean	Specifies that the Programming interfaces such as Auto Programming, JTAG, SPI Target are either permanently enabled or disabled. A value of true or 1 will permanently disable all of the programming interfaces. The default value is false.
permanently_disable_upk1	boolean	Specifies that the User Key UPK1 is either permanently enabled or disabled. A value of true or 1 will permanently disable FlashLock UPK1 unlocking. The default value is false.
permanently_disable_upk2	boolean	Specifies that the User Key UPK2 is either permanently enabled or disabled. A value of true or 1 will permanently disable FlashLock UPK2 unlocking. The default value is false.
permanently_write_protect_fabric	boolean	Specifies that the write protection for fabric is either permanently enabled or disabled. A value of the true or 1 will make the fabric one-time programmable. The default value is false.
one_way_passcode	boolean	Specifies 1 to enable or 0 (default) to disable **One-Way Passcode** (OWP). Specifies true to enable or false (default) to disable **One-Way Passcode** (OWP).

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.
None	Permanent locks cannot be configured for bit stream encryption with default key.
None	The Configure Permanent Locks for Production tool is not available for PolarFire SoC.

Supported Families

PolarFire®

Example

The following example specifies that `SPM_OTP` tool is configured to permanently disable user keys UPK1 and UPK2:

```
configure_tool -name {SPM_OTP} \
               -params {permanently_disable_debugging:false} \
               -params {permanently_disable_dpk:false} \
               -params {permanently_disable_factory_access:false} \
               -params {permanently_disable_prog_interfaces:false} \
               -params {permanently_disable_upk1:true} \
               -params {permanently_disable_upk2:true} \
               -params {permanently_write_protect_fabric:false}
```

The following example specifies that `SPM_OTP` tool is configured to permanently disable programming interfaces:

```
configure_tool -name {SPM_OTP} \
               -params {permanently_disable_debugging:false} \
           -params {permanently_disable_dpk:false} \
               -params {permanently_disable_factory_access:false} \
               -params {permanently_disable_prog_interfaces:true} \
               -params {permanently_disable_upk1:false} \
               -params {permanently_disable_upk2:false} \
               -params {permanently_write_protect_fabric:false}
```

The following example specifies that `SPM_OTP` tool is configured to enable `one_way_passcode`:

```
configure_tool -name {SPM_OTP} \
 -params {one_way_passcode:true} \
```

See Also

- SPM

6.21. SYNTHESIZE (Ask a Question)

Description

"SYNTHESIZE" is a command tool used in configure_tool and run_tool. Configure_tool is a general purpose Tcl command that allows you to configure a tool's parameters and values prior to executing the tool. Configure tool then executes the specified tool with the configured parameters with run_tool command.

```
configure_tool -name {SYNTHESIZE} [-params {name:value}]
run_tool -name {SYNTHESIZE}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
CLOCK_ASYNC	integer	Specifies the threshold value for asynchronous pin promotion to a global net. The default is 12.
CLOCK_GLOBAL	integer	Specifies the threshold value for Clock pin promotion. The default is 2.
CLOCK_DATA	integer	Specifies the threshold value for data pin promotion.The default is 5000. The possible values are integer values between 1000 and 200,000.
RAM_OPTIMIZED_FOR_POWER	boolean	Set to true or 1 to optimize RAM for Low Power; RAMS are inferred and configured to ensure the lowest power consumption. Set to false or 0 to optimize RAM for High Speed at the expense of more FPGA resources. The possible values are boolean {true | false | 1 | 0}.
RETIMING	boolean	Set to true or 1 to enable Retiming during synthesis. Set to false or 0 to disable Retiming during synthesis. The possible values are boolean {true | false | 1 | 0}.
SYNPLIFY_OPTIONS	string	Specifies additional synthesis-specific options. Options specified by this parameter override the same options specified in the user Tcl file if there is a conflict.
SYNPLIFY_TCL_FILE	string	Specifies the absolute or relative path name to the user Tcl file containing synthesis-specific options.
BLOCK_MODE	boolean	Set to true or 1 when you have blocks in your design and you want to enable the Block mode. Set it to false or 0 if you don't have blocks in your design. Default is false or 0. The possible values are boolean {true | false | 1 | 0}.
BLOCK_PLACEMENT_CONFLICTS	string	Instructs the COMPILE engine what to do when the software encounters a placement conflict. When set to: ERROR - Compile errors out if any instance from a Designer block becomes unplaced. This is the default.KEEP - If some instances get unplaced for any reason, the non-conflicting elements remaining are preserved but not locked. Therefore, the placer can move them into another location if necessary. LOCK- If some instances get unplaced for any reason, the non-conflicting elements remaining are preserved and locked.DISCARD – Discards any placement from the block,even if there are no conflicts. The possible values are {ERROR | KEEP | LOCK | DISCAR D}.
BLOCK_ROUTING_CONFLICTS	string	Instructs the COMPILE engine what to do when the software encounters a routing conflict. When set to: ERROR - Compile errors out if any route in any preserved net from a Designer block is deleted. This is the default. KEEP – If a route is removed from a net for any reason, the routing for the non-conflicting nets is kept unlocked. The router can re-route these nets. LOCK – If routing is removed from a net for any reason, the routing for the non- conflicting nets is kept as locked, and the router will not change them. DISCARD - Discards any routing from the block,even if there are no conflicts. The possible values are {ERROR | KEEP | LOCK | DISCAR D}.
PA4_GB_COUNT	integer	The number of available global nets is reported. Minimum for all dies is "0". Default and Maximum values are die-dependent: 005/010 die: Default =Max = 8 025/050/060/090/150 die: Default=Max=16 RT4G075/RT4G150: Default=24, Max=48.Note: For RTG4, default is 48.
PA4_GB_MAX_RCLKINT_INSERTION	integer	Specifies the maximum number of global nets that could be demoted to row-globals. Default is 16, Minis 0 and Max is 50.
PA4_GB_MIN_GB_FANOUT_TO_USE_RCLKINT	integer	Specifies the Minimum fanout of global nets that could be demoted to row-globals. Default is 300.Min is 25 and Max is 5000.
LANGUAGE_SYSTEM_VLOG	boolean	Set to true if the Verilog files contain System Verilog constructs. The possible values are boolean {true | false}.
LANGUAGE_VERILOG_2001	boolean	Set to true if Verilog files contain Verilog 2001 constructs. The possible values are boolean {true | false}.
CDC_MIN_NUM_SYNC_REGS	integer	Minimum number of synchronizer registers. Range: 2 - 9. Default: 2

© 2025 Microchip Technology Inc. and its subsidiaries

SYNTHESIZE (continued)		
Parameter	**Type**	**Description**
CDC_REPORT	boolean	The generated CDC report will not contain any synchronizer circuits formed with macros instantiated from the catalog. The generated report, with the name `<root_name>_cdc.csv`, will be visible in the respective Synthesis node of the report view. The report contains all CDC inferred from the RTL design and explains the reason(s) why a synchronizer is considered unsafe. The possible values are boolean {true | false}.

Return Type	Description
Integer	Both for "configure_tool -name {SYNTHESIZE}" and "run_tool -name {SYNTHESIZE}" returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Parameter 'params' has illegal value.
None	Required parameter 'params' is missing.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_tool -name "tool name" [-params "params"]+'.
None	RETIMING is set to an illegal value value. Legal value are true, false, 0 or 1.
None	RETIMING has no value. Legal values are true, false, 0 or 1.
None	RAM_OPTIMIZED_FOR_POWER is set to an illegal value value. Legal value are true, false, 0 or 1.
None	RAM_OPTIMIZED_FOR_POWER has no value. Legal values are true, false, 0 or 1.
None	PA4_GB_MIN_GB_FANOUT_TO_USE_RCLKINT is set to an illegal value value. Legal value must be between 25 and 5000.
None	PA4_GB_MIN_GB_FANOUT_TO_USE_RCLKINT has no value. Legal value must be between 25 and 5000.
None	PA4_GB_MAX_RCLKINT_INSERTION has no value. Legal value must be between 0 and 50.
None	CLOCK_GLOBAL is set to an illegal value value. Legal value must be between 2 and 200000.
None	CLOCK_GLOBAL has no value. Legal value must be between 2 and 200000.
None	CLOCK_DATA is set to an illegal value value. Legal value must be between 1000 and 200000.
None	CLOCK_DATA has no value. Legal value must be between 1000 and 200000.
None	CLOCK_ASYNC is set to an illegal value value. Legal value must be between 12 and 200000.
None	CLOCK_ASYNC has no value. Legal value must be between 12 and 200000.
None	BLOCK_ROUTING_CONFLICTS is set to an illegal value value. Legal values for BLOCK_ROUTING_CONFLICTS are ERROR, LOCK, KEEP and DISCARD.
None	BLOCK_ROUTING_CONFLICTS has no value. Legal values for BLOCK_ROUTING_CONFLICTS are ERROR, LOCK, KEEP and DISCARD.
None	BLOCK_PLACEMENT_CONFLICTS is set to an illegal value value. Legal values for BLOCK_PLACEMENT_CONFLICTS are ERROR, LOCK, KEEP and DISCARD.
None	BLOCK_PLACEMENT_CONFLICTS has no value. Legal values for BLOCK_PLACEMENT_CONFLICTS are ERROR, LOCK, KEEP and DISCARD.
None	BLOCK_MODE is set to an illegal value value. Legal value are true, false, 0 or 1.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

Configure "SYNTHESIZE" tool and run.

```
configure_tool -name {SYNTHESIZE} \ -params {BLOCK_MODE:false} \
               -params {BLOCK_PLACEMENT_CONFLICTS:ERROR} \
               -params {BLOCK_ROUTING_CONFLICTS:ERROR} \
               -params {CLOCK_ASYNC:12} -params {CLOCK_DATA:5010} \
               -params {CLOCK_GLOBAL:2} \
               -params {PA4_GB_MAX_RCLKINT_INSERTION:16} \
               -params {PA4_GB_MIN_GB_FANOUT_TO_USE_RCLKINT:299} \
               -params {RAM_OPTIMIZED_FOR_POWER:false} \
               -params {RETIMING:false} \
               -params {SYNPLIFY_TCL_FILE:./test.tcl} \
               -params {SYNPLIFY_OPTIONS:set_option \
               -run_prop_extract 1;
               set_option -maxfan 10000;
               set_option -clock_globalthreshold 2;
               set_option -async_globalthreshold 12;
               set_option -globalthreshold 5000;
               set_option -low_power_ram_decomp 0;}
```

```
run_tool -name {SYNTHESIZE}
```

To configure Synthesis tool for adding the IDC file:

```
configure_tool -name {SYNTHESIZE} -params {ACTIVE_IMPLEMENTATION:synthesis}
-params {CREATE_IMPLEMENTATION_IDENTIFY:Synthesis_Jan27_N} -params
{SYNPLIFY_OPTIONS:} -params {SYNPLIFY_TCL_FILE:D:/Syn_27.tcl}
```

Where `Syn_27.tcl` script contains the following given command for adding IDC file:

```
set_option -ident_constraint  D:/MK.idc
```

6.22. UPDATE_ENVM (Ask a Question)

Description

"UPDATE_ENVM" is a command tool used in the run_tool command. The "run_tool -name {UPDATE_ENVM} -script {update_config_file}" Tcl command updates eNVM Memory Content without having to rerun Synthesize and Place and Route. It is useful if you have reserved space in the eNVM configurator within the MSS for firmware development.

Note: Before running this tool, you must configure MSS which internally uses eNVM, create client.

```
run_tool -name {UPDATE_ENVM} -script {update_config_file.cfg}
```

Arguments

Parameter	Type	Description
script	string	Specifies the path to the configuration file of the eNVM client. This parameter is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Cannot open eNVM configuration file.
None	TCL script 'update_config_file.cfg' not found.

Supported Families

SmartFusion® 2

IGLOO® 2

Example

This example runs and configures the eNVM clients with {./eNVM.cfg}:

```
run_tool -name {UPDATE_ENVM} -script "eNVM.cfg"
```

See Also

- nvm_update_serialization_client
- nvm_update_storage_client

6.23. USER_PROG_DATA (Ask a Question)

Description

"USER_PROG_DATA" is a command tool used in configure_tool. Configure_tool -name {USER_PROG_DATA} sets the Design Version and Silicon Signature in your device.

Note: The Configure User Programming Data tool is obsolete. You must now use the Configure Programming Options tool to configure design version and silicon signature.

```
configure_tool -name {USER_PROG_DATA} \
                           -params {name:value} \
                           -params {name:value}
```

Arguments

Parameter	Type	Description
design_version	integer	Sets the design version. It must be greater than the Back level version in SPM Update Policy. The possible values are integers 0 through 65535.
silicon_signature	hexadeciml	32-bit (8 hex characters) silicon signature to be programmed into the device. This field can be read from the device using the JTAG USERCODE instruction. The possible values are 8 Hex characters (max length 8 Hex characters)

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_tool -name "tool name" [-params "params"]+'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

SmartFusion® 2
IGLOO® 2
RTG4™

Example

This example configures the {USER_PROG_DATA} tool with the following parameters.

```
configure_tool -name {USER_PROG_DATA} \
               -params {design_version:255} \
               -params {silicon_signature:abcdffff}
```

6.24. VERIFYPOWER (Ask a Question)

Description

"VERIFYPOWER" is a command tool used in run_tool. The command run_tool passes an absolute path of the script file that contains power-specific Tcl commands to the "VERIFYPOWER" command and executes it.

```
run_tool -name {VERIFYPOWER} -script {power_analysis.tcl}
```

Arguments

Parameter	Type	Description
script	string	Specify absolute path of the script file. This is an optional parameter. Script contains power-specific Tcl commands. **Note:** You can include power-specific Tcl commands to generate power reports.

Return Type	Description
Integer	Returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	None

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example changes SmartPower operating condition settings from the default to 40 °C junction temperature and 1.25V VDD.

```
# Change from pre-defined temperature and voltage mode (COM,IND,MIL) to SmartPower custom
smartpower_set_temperature_opcond -use "design"
smartpower_set_voltage_opcond -voltage "VDD" -use "design"

# Set the custom temperature to 40C ambient temperature.
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
smartpower_temperature_opcond_set_design_wide -typical 40 \
                                              -best 40 \
                                              -worst 40
# Set the custom voltage to 1.25V
smartpower_voltage_opcond_set_design_wide -voltage "VDD" \
                                          -typical 0.970 \
                                  -best 1.25 -worst 1.25
```

6.25. VERIFYTIMING (Ask a Question)

Description

"VERIFYTIMING" is a command tool used in configure_tool and run_tool. Use "configure_tool -name {VERIFYTIMING}" to configure "VERIFYTIMING" tool, run_tool passes a script file that contains timing specific Tcl commands to the VERIFYTIMING command and executes it.

Note: At least one "parameter:value" must be specified. You can repeat `-params` argument for multiple parameters.

```
configure_tool -name {VERIFYTIMING} -params {parameter:value}
run_tool -name {VERIFYTIMING} -script {timing.tcl}
```

Where, `timing.tcl` is a script that contains the SmartTime-specific Tcl commands. You can include SmartTime-specific Tcl commands to create user path, to set and generate timing reports.

Arguments

The following table list the "VERIFYTIMING" arguments for PolarFire, SmartFusion 2, IGLOO 2, RTG4 and PolarFire SoC.

Parameter	Type	Description
MAX_TIMING_MULTI_CORNER	boolean	Max Delay Static Timing Analysis report based on multi-corners. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MAX_TIMING_VIOLATIONS_MULTI_CORNER	boolean	Max Delay Static Timing Analysis violation report based on multi-corners. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MIN_TIMING_MULTI_CORNER	boolean	Min Delay Static Timing Analysis report based on multi-corners. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MIN_TIMING_VIOLATIONS_MULTI_CORNER	boolean	Min Delay Static Timing Analysis violation report based on multi-corners. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MAX_TIMING_SLOW_LV_HT	boolean	Max Delay Static Timing Analysis report based on Slow process, Low Voltage, and High Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MIN_TIMING_FAST_HV_LT	boolean	Min Delay Static Timing Analysis report based on Fast process, High Voltage, and Low Temperature operating conditions.
MAX_TIMING_FAST_HV_LT	boolean	Max Delay Static Timing Analysis report based on Fast process, High Voltage, and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MIN_TIMING_SLOW_LV_HT	boolean	Min Delay Static Timing Analysis report based on Slow process, Low Voltage and High Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MAX_TIMING_VIOLATIONS_SLOW_LV_HT	boolean	Max Delay Static Timing Analysis violation report based on Slow process, Low Voltage and High Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.

© 2025 Microchip Technology Inc. and its subsidiaries

VERIFYTIMING (continued)		
Parameter	**Type**	**Description**
MIN_TIMING_VIOLATIONS_FAST_HV_LT	boolean	Min Delay Static Timing Analysis violation report based on Fast process, High Voltage and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MAX_TIMING_VIOLATIONS_FAST_HV_LT	boolean	Max Delay Static Timing Analysis violation report based on Fast process, High Voltage and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MIN_TIMING_VIOLATIONS_SLOW_LV_HT	boolean	Min Delay Static Timing Analysis violation report based on Slow process, Low Voltage and High Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MAX_EXPANDED_PATHS_TIMING	integer	Sets the number of expanded paths under each section. The number of paths set can be anywhere from 1 to 20000. Default is 5.
MAX_EXPANDED_PATHS_VIOLATION	integer	Sets the number of expanded paths under each section. The number of paths set can be anywhere from 0-20000. Default is 20.
MAX_PARALLEL_PATHS_TIMING	integer	Sets the number of parallel paths for each expanded path.
MAX_PARALLEL_PATHS_VIOLATION	integer	Sets the number of parallel paths for each expanded path. The number of paths set can be anywhere from 1-20000. Default is 1.
MAX_PATHS_TIMING	integer	Sets the number of reported paths under each section. The number of paths set can be anywhere from 1- 20000. Default is 10.
MAX_PATHS_VIOLATION	integer	Sets the number of reported paths under each section. The number of paths set can be anywhere from 1- 20000. Default is 20.
SLACK_THRESHOLD_VIOLATION	real	Sets a maximum slack threshold value in nanoseconds. Paths will be filtered based on the slack threshold value only in Timing Violation reports. Value entered is a real number.
CONSTRAINTS_COVERAGE	boolean	Constraint Timing Coverage report.
FORMAT	string	Selects the format for the Timing Report: Text(txt), XML(xml), CSV(csv) or HTML(html) format.
SMART_INTERACTIVE	boolean	When SMART_INTERACTIVE parameter is set to 1. The verify timing tool opens timing report explorer interactively and reports timing paths. Available from Libero v12.6+.
MAX_PATHS_INTERACTIVE_REPORT	integer	Specifies maximum number of path in the max and min analysis. The default value is set to 1000. The Range is 1-10000. Available from Libero v12.6+.

The following table list the "VERIFYTIMING" arguments for PolarFire and PolarFire SoC.

Parameter	**Type**	**Description**
MAX_TIMING_SLOW_LV_LT	boolean	Max Delay Static Timing Analysis report based on Slow process, Low Voltage, and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 1.
MIN_TIMING_SLOW_LV_LT	boolean	Min Delay Static Timing Analysis report based on Slow process, Low Voltage and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MAX_TIMING_VIOLATIONS_SLOW_LV_LT	boolean	Max Delay Static Timing Analysis violation report based on Slow process, Low Voltage and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.
MIN_TIMING_VIOLATIONS_SLOW_LV_LT	boolean	Min Delay Static Timing Analysis violation report based on Slow process, Low Voltage and Low Temperature operating conditions. Specify 0 or "False" to turn report generation OFF and 1 or "True" to turn it ON. Default is 0.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
Integer	"configure_tool -name {VERIFYTIMING}" returns 0 on success and 1 on failure. "run_tool -name {VERIFYTIMING} -script {<timing.tcl>}" returns 0 on success and 1 on failure.

Error Codes

Error Code	Description
None	Required parameter 'params' is missing.
None	FORMAT is set to an illegal value 'forma_value'. Formats for timing reports generation are XML, TEXT, HTML and CSV.
None	MAX_EXPANDED_PATHS_TIMING is set to an illegal value 'value'. Legal values must be between 1 and 200000.
None	MAX_PARALLEL_PATHS_VIOLATION is set to an illegal value 'values'. Legal values must be between 1 and 200000.
None	MAX_PARALLEL_PATHS_TIMING is set to an illegal value 'value'. Legal values must be between 1 and 200000.
None	MAX_PATHS_VIOLATION is set to an illegal value 'value'. Legal values must be between 1 and 200000.
None	MAX_PATHS_TIMING is set to an illegal value 'value'. Legal values must be between 1 and 200000.
None	MAX_TIMING_FAST_HV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_MULTI_CORNER is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_SLOW_LV_HT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_SLOW_LV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_VIOLATIONS_SLOW_LV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_VIOLATIONS_SLOW_LV_HT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_VIOLATIONS_MULTI_CORNER is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MAX_TIMING_VIOLATIONS_FAST_HV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_SLOW_LV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_SLOW_LV_HT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_MULTI_CORNER is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_FAST_HV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_VIOLATIONS_SLOW_LV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_VIOLATIONS_SLOW_LV_HT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_VIOLATIONS_MULTI_CORNER is set to an illegal value 'value'. Legal value are true, false, 0 or 1.
None	MIN_TIMING_VIOLATIONS_FAST_HV_LT is set to an illegal value 'value'. Legal value are true, false, 0 or 1.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example specifies text as the Report format and turns on report generation for Constraint_coverage report, the Max_Delay_Analysis Timing Report based on Slow process, Low Voltage, and High Temperature and the Max_Delay_Analysis Timing Violation Report based on Slow Process, Low Voltage, and High Temperature operating conditions. The script turns off Min_Delay_Analysis Timing Violation Report generation based on Slow process, Low Voltage, and High Temperature operating conditions. It also sets the maximum number of timing paths to 25 and maximum slack threshold of 0.04:

```
configure_tool -name {VERIFYTIMING} \
               -params {FORMAT:txt} \
               -params {CONSTRAINT_COVERAGE:1} \
               -params {MAX_TIMING_SLOW_LV_HT:1} \
               -params {MAX_TIMING_VIOLATIONS_SLOW_LV_HT:1} \
               -params {MIN_TIMING_VIOLATIONS_SLOW_LV_HT:0}
               -params {MAX_PATHS_TIMING:25} \
               -params {SLACK_THRESHOLD_VIOLATION:0.04}

run_tool -name {VERIFYTIMING} -script {timing.tcl}
```

Sample SmartTime Tcl Script <timing.tcl>

```
# Create user path set -from B_reg
create_set -name from_B_reg \
    -source {B_reg*[*]:CLK} -sink {*}

# Create user set -from A, B, C
create_set -name from_in_ports \
    -source {A B C} \
    -sink {*}

# Generate Timing Reports
report \
    -type timing \
    -analysis min \
    -format text \
    -max_paths 10 \
    -print_paths yes \
    -max_expanded_paths 10 \
    -include_user_sets yes \
    min_timing.rpt

# Export SDC
write_sdc -scenario {Primary} exported.sdc

#save the changes
save
```


© 2025 Microchip Technology Inc. and its subsidiaries

7. MSS Tcl Commands (Ask a Question)

7.1. mss_configure_envm (Ask a Question)

This command is used to specify a `.cfg`file with all clients info in the ENVM core instance of the MSS component.

```
mss_configure_envm \
-component_name {component_name} \
-cfg_file {file_path}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the MSS component.
-cfg_file {file_path}	string	Mandatory. Path of the configuration file(.cfg) used to configure the envm.

Example

```
mss_configure_envm -component_name {test_sb_MSS} -cfg_file{./ENVM.cfg}
```

7.2. mss_configure_instance (Ask a Question)

Description

This command is used to configure the parameters of a core instance inside the MSS component.

```
mss_configure_instance \
-component_name {component_name} \
-instance_name {instance_name} \
-params{param:value list}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the MSS component.
-instance_name {instance_name}	string	Mandatory. Name of the core instance to be configured inside the MSS component.
-params{param:value list}	string	Mandatory. List of parameters and values to be configured for the selected core instance inside the MSS component.

Example

```
mss_configure_instance -component_name {test_sb_MSS} -instance_name {CC} -params {\
"CACHE_ENABLED:false" \
"CC_CACHE_REGION:128MB_0001" }
```

7.3. mss_disable_instance (Ask a Question)

Description

This command is used to disable a core instance inside the MSS component.

```
mss_disable_instance \
-component_name {component_name} \
-instance_name {instance_name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the MSS component.
-instance_name {instance_name}	string	Mandatory. Name of the core instance to be disabled inside the MSS component.

Example

```
mss_disable_instance -component_name {test_sb_MSS} -instance_name {I2C_1}
```

7.4. mss_enable_instance (Ask a Question)

Description

This command is used to enable a core instance inside the MSS component.

```
mss_enable_instance \
-component_name {component_name} \
-instance_name {instance_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the MSS component.
-instance_name {instance_name}	string	Mandatory. Name of the core instance to be enabled inside the MSS component.

Example

```
mss_enable_instance -component_name {test_sb_MSS} -instance_name {MMUART_0}
```

7.5. mss_save_instance_config (Ask a Question)

Description

This command is used to save the configuration of a core instance inside the MSS component specified using the 'mss_configure_instance' command.

```
mss_save_instance_config \
-component_name {component_name} \
-instance_name {instance_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the MSS component.
-instance_name {instance_name}	string	Mandatory. Name of the core instance inside the MSS component.

Example

```
mss_save_instance_config -component_name {test_sb_MSS} -instance_name {CAN}
```


© 2025 Microchip Technology Inc. and its subsidiaries

8. SmartTime Tcl Commands (Ask a Question)

8.1. all_inputs (Ask a Question)

Description

Returns an object representing all input and inout pins in the current design. This command is usually used with a command which puts the same attributes on input ports. If you want only certain ports, use the `get_ports` command.

```
all_inputs
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
object	Returns an object representing all input and inout pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You can only use this command as part of a `-from`, `-to` argument in the following Tcl commands: `set_min_delay`, `set_max_delay`, `set_multicycle_path`, and `set_false_path`. It cannot be used with `-through` option.

Example

The following example sets a maximum delay by constraining all paths from all_inputs to ck1 clock with a delay less then 2 ns.

```
set_max_delay 2 -from [all_inputs] -to [get_clocks ck1]
```

Related Examples on GitHub

- all_inputs

8.2. all_outputs (Ask a Question)

Description

Returns an object representing all output and inout pins in the current design. This command is usually used with a command which puts the same attributes on output ports. If you want only certain ports, use `get_ports` command.

```
all_outputs
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
object	Returns an object representing all output and inout pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You can only use this command as part of a `-from`, `-to` argument in the following Tcl commands: `set_min_delay`, `set_max_delay`, `set_multicycle_path`, and `set_false_path`. It cannot be used with `-through` option.

Example

The following example sets a maximum delay by constraining all paths from `all_inputs` to `all_outputs` with a delay less then 2 ns.

```
set_max_delay 2 -from [all_inputs] -to [all_outputs]
```

Related Examples on GitHub

- all_outputs

8.3. all_registers (Ask a Question)

Description

Returns an object representing register pins or register cells (default) in the current scenario based on the given parameters. If you do not specify an option, this command returns an object representing registers cells.

```
all_registers [-clock clock_name ] [-async_pins] \
[-output_pins] [-data_pins] [-clock_pins]
```

Arguments

Parameter	Type	Description
`clock`	string	Specifies the name of the clock domain to which the registers belong. If no clock is specified, all registers in the design will be targeted.
`async_pins`	None	Lists all register pins that are async pins for the specified clock (or all registers asynchronous pins in the design).
`output_pins`	None	Lists all register pins that are output pins for the specified clock (or all registers output pins in the design).
`data_pins`	None	Lists all register pins that are data pins for the specified clock (or all registers data pins in the design).
`clock_pins`	None	Lists all register pins that are clock pins for the specified clock (or all registers clock pins in the design).

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
object	Returns an object representing register pins or cells in the current scenario based on the given parameters.

Error Codes

Error Code	Description
Error: SDC0021	Invalid max delay constraint: the -from value is incorrect.
Error: SDC0023	Invalid max delay constraint: the -to value is incorrect.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You can only use this command as part of a `-from`, `-to` argument in the following Tcl commands: `set_min_delay`, `set_max_delay`, `set_multicycle_path`, and `set_false_path`.

Example

The following example sets a maximum delay by constraining all paths from `ff_m:CLK` or `ff_s2:CLK` to `ff_m:Q pin` with a delay less than 2.000 ns.

```
set_max_delay 2.000 -from { ff_m:CLK ff_s2:CLK } \
-to [all_registers -clock_pins -clock {ff_m:Q}]
```

Related Examples on GitHub

- all_registers

8.4. check_constraints (Ask a Question)

Description

Checks all timing constraints in the current scenario for validity. This command performs the same checks as when the constraint is entered through SDC or Tcl.

When a constraint file is checked, the Constraint Checker does the following:

- Checks the syntax.
- Compares the design objects (pins, cells, nets, ports) in the constraint file versus the design objects in the netlist (RTL or post-layout ADL netlist). Any discrepancy (for example, constraints on a design object which does not exist in the netlist) are flagged as errors and reported in the `*_sdc.log` file.

```
check_constraints
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

RTG4™

Example
The following example checks timing constraints in the current scenario.

```
check_constraints
```

8.5. clone_scenario (Ask a Question)

Description
Creates a timing scenario with the new_scenario_name, which includes a copy of all constraints in the original scenario. The new scenario is then added to the list of scenarios. You must provide a unique name (that is, it cannot already be used by another timing scenario).
Note: It is recommended to use the `organize_tool_files` command instead of `clone_scenario`.

```
clone_scenario original new_scenario_name
```

Arguments

Parameter	Type	Description
`original`	string	Specifies the name of the source timing scenario to clone (copy). The source must be a valid, existing timing scenario.
`new_scenario_name`	string	Specifies the name of the new scenario to be created.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example
The following example creates a new timing scenario with the name `my_new_scenario` by duplicating an existing one (primary).

```
clone_scenario primary my_new_scenario
```

See Also
- create_scenario
- remove_scenario
- rename_scenario

Related Examples on GitHub
- clone-scenario

8.6. create_clock (Ask a Question)

Description
Creates a clock constraint on the specified sources in the current design, or a virtual clock if no source other than a name is specified. It also defines its period and waveform. The static timing

© 2025 Microchip Technology Inc. and its subsidiaries

analysis tool uses this information to propagate the waveform across the clock network to the clock pins of all sequential elements driven by this clock source.

The clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

```
create_clock [ -name clock_name ] [-add] -period period_value \
[ -waveform edge_list ][ source_objects ]
```

Table 8-1. Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the clock constraint. You must specify either a clock name or a source. If the `-name` option is not used, the clock name is specified as source name. The clock name refer to the clock in other commands. You can specify name as: `-name {clk}` or `-name clk`.
`add`	None	Specifies that a new clock constraint is created at the same source port as the existing clock without overriding the existing constraint. The name of the new clock constraint with the `-add` option must be different than the existing clock constraint. Otherwise, it will override the existing constraint, even with the `-add` option. The `-name` option must be specified with the `-add` option.
`period`	real	Specifies the clock period in nanoseconds. The value you specify is the minimum time over which the clock waveform repeats. The `period_value` must be greater than zero.
`waveform`	real	Specifies the rise and fall times of the clock waveform in ns over a complete clock period. There must be exactly two transitions in the list, a rising transition followed by a falling transition. So in the edge list, the falling edge value must be greater than the rising edge value. For example, a clock waveform of period 19 that has a rising edge at 3 ns and a falling edge at 8 ns will have the waveform defined as [3 8].
`source_objects`	list of string	Specifies the source of the clock constraint. The source can be ports, pins, or nets in the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing one. Specify either a source or a clock name.

Return Type	Description
`integer`	Returns the ID of the clock constraint.

Table 8-2. Error Codes

Error Code	Description
Error: SDC0001	Invalid clock constraint: clock source is incorrect.
Error: SDC0006	Invalid clock constraint: clock period is incorrect for the specified clock.
Error: SDC0007	Invalid clock constraint: waveform is incorrect.
Error: SDC0061	Invalid clock constraint: Missing or Illegal parameter/value.
Error: SDC0069	Invalid clock constraint: Need to specify clock name with `-add` option.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example creates two clocks, one on port CK1 with a period of 6, and the other on port CK2 with a period of 6, a rising edge at 0, and a falling edge at 3.

```
create_clock -name {my_user_clock} -period 6 CK1
```

```
create_clock -name {my_other_user_clock} -period 6 -waveform {0 3} {CK2}
```

The following example creates a clock on port CK3 with a period of 7, a rising edge at 2, and a falling edge at 4.

```
create_clock -period 7 -waveform {2 4} [get_ports {CK3}]
```

The following example creates a new clock constraint `clk2`, in addition to `clk1`, on the same source port clk1 without overriding it.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports clk1]
```

```
create_clock -name clk2 -add -period 20 -waveform {0 10} [get_ports clk1]
```

The following example does not add a new clock constraint, even with the `-add` option, but overrides the existing clock constraint because of the same clock names.
Note: To add a new clock constraint in addition to the existing clock constraint on the same source port, the clock names must be different.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports clk1]
```

```
create_clock -name clk1 -add -period 50 -waveform {0 25} [get_ports clk1]
```

The following example shows the SDC constraint that must be added for 050 devices, with max accuracy of 4% and 52 MHz (clock period 19.230 ns).

```
create_clock -name {OSC_0/I_RCOSC_25_50MHZ/CLKOUT} -period 19.230 [ get_pins {OSC_0/
I_RCOSC_25_50MHZ/CLKOUT}]
```

Related Examples on GitHub

- create_clock

8.7. create_generated_clock (Ask a Question)

Description

Creates a generated clock in the current design, at a declared source, by defining its frequency with respect to the frequency at the reference pin. The static timing analysis tool uses this information to compute and propagate its waveform across the clock network to the clock pins of all sequential elements driven by this source.

The generated clock information is also used to compute the slacks in the specified clock domain that drive optimization tools such as place-and-route.

```
create_generated_clock [-name clock_name ] [-add] \
                       [-master_clock clock_name ] \
                       -source reference_pin \
                       [- divide_by divide_factor ] \
                       [-multiply_by multiply_factor ] \
                       [-invert] source [-edges values ] \
                       [-edge_shift values ]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specifies the name of the clock constraint. If the -name option is not used, the generated clock receives the same name specified in the source. The clock name refers to the clock in other commands. You can specify -name {my_gen_clk} or -name my_gen_clk.
add	None	Specifies that the generated clock constraint is a new clock constraint in addition to the existing one at the same source. Use this option to capture the case where multiple generated clocks must be specified on the same source, because multiple clocks fan into the master pin. If you specify this option, you must also use the -name option. The name of the clock constraint should be different from the existing clock constraint. With this option, the -master_clock and -name options must be specified.
master_clock	string	Specifies the master clock used for the generated clock when multiple clocks fan into the master pin. This option must be used in conjunction with -add option of the generated clock. **Notes:** • The master_clock option is used only with the -add option for the generated clocks. • If there are multiple master clocks fanning into the same reference pin, the first generated clock specified uses the first master clock as its source clock. • The clocks generated subsequently, specified with the -add option, can choose any of the master clocks as their source clock (including the first master clock specified).
source	string	Specifies the reference pin in the design from which the clock waveform is to be derived. You must specify the -source reference pin.
divide_by	integer	Specifies the frequency division factor. This option cannot be used with -egde list. If -egde is specified, the divide_by value defaults to one. For example, if the divide_factor is equal to 2, the generated clock period is twice the reference clock period. If you set the divide_by value to 1.2 or 4/2 or 8a2, it is being truncated as 1 or 4 or 8, and no warning is reported.
multiply_by	integer	Specifies the frequency multiplication factor. This option cannot be used with -egde list. If egde is specified, the multiply_by and divide_by values default to 1. For example, if the multiply_factor is equal to 2, the generated clock period is half the reference clock period. If you set the multiply_by value to 1.2 or 4/2 or 8a2, it is being truncated as 1 or 4 or 8, and no warning is reported.
invert	None	Specifies that the generated clock waveform is inverted with respect to the reference clock.
source	list of strings	Specifies the source of the clock constraint on internal pins of the design. If you specify a clock constraint on a pin that already has a clock, the new clock replaces the existing clock. Only one source is accepted. Wildcards are accepted as long as the resolution shows one pin. You must specify a source.
edges	list of integers	Specifies a list of positive integers which represent the edges from the source clock that are to form the edges of the generated clock. To generate the clock, three values must be specified. If you specify less than three values, a tool tip indicates an error. This option cannot be used with the -divide_by/-multiply_by factor.
edge_shift	list of floating point numbers	Specifies a list of three floating point numbers which represent the amount of shift (in nanoseconds) that the specified edges are to undergo to yield the final generated clock waveform. These floating point values can be positive or negative. Positive value indicates a shift later in time, while a negative value indicates a shift earlier in time. With this option, the -edges option must be specified.

© 2025 Microchip Technology Inc. and its subsidiaries

Important:
In the SmartTime tool, the phase shift specified with the `-phase` argument is treated as a delay within the clock network rather than a waveform translation. As a result, it is not considered when determining the expected clock edges for setup and hold analysis. Consequently, the first active edge of the launch clock serves as the starting point, and the second active edge of the capture clock remains the endpoint, just as it would without a phase shift.

Return Type	Description
`integer`	Returns the ID of the generated clock constraint.

Error Codes

Error Code	Description
Error: SDC0004	Invalid generated clock constraint: Name does not match any clock name or source.
Error: SDC0015	Invalid generated clock constraint: Port list is incorrect.
Error: SDC0016	Invalid generated clock constraint: Port list is empty.
Error: SDC0061	Invalid generated clock constraint: The `-edges` argument is empty invoked from within command.
Error: SDC0062	Invalid generated clock constraint: The `-edges` list size must be three.
Error: SDC0063	Invalid generated clock constraint: The `-edges` list elements are not in increasing order.
Error: SDC0065	Invalid generated clock constraint: The `-edges` cannot be used with the `-multiply_by` or `-divide_by` option.
Error: SDC0066	Invalid generated clock constraint: The `-edge_shift` does not have accompanying `-edges`.
Error: SDC0069	Invalid clock constraint: Need to specify a clock name with the `-add` option.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a generated clock on pin `U1/reg1:Q` with a period twice as long as the period at the reference port CLK.

```
create_generated_clock -name {my_user_clock} -divide_by 2 \
                       -source [get_ports {CLK}] U1/reg1:Q
```

The following example creates a generated clock at the primary output of myPLL with a period that is three fourth of the period at the reference pin clk.

```
create_generated_clock -divide_by 3 -multiply_by 4 \
                       -source clk [get_pins {myPLL:CLK1}]
```

The following example creates a new generated clock gen2, in addition to gen1, derived from same master clock as the existing generated clock, and the new constraint is added to pin `r1/CLK`.

```
create_generated_clock -name gen1 -multiply_by 1 \
                       -source [get_ports clk1] [get_pins r1/CLK]
create_generated_clock -name gen2 -add -master_clock clk1 \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                        -source [get_ports clk1] \
                        -multiply_by 2 [get_pins r1/CLK]
```

The following example does not create a new generated clock constraint in addition to the existing clock, but overrides even with the `-add` option enabled because the same names are used.

```
create_generated_clock -name gen2 -source [get_ports clk1] \
                        -multiply_by 3 [get_pins r1/CLK]
create_generated_clock -name gen2 -source [get_ports clk1] \
                        -multiply_by 4 -master_clock clk1 \
                        -add [get_pins r1/CLK]
```

The following example shows an SDC constraint for a generated clock of 50 MHz reference clock and 100 MHz output clock with a 90° phase shift.

```
create_generated_clock \
          -name {FCCC_0/GL0} -multiply_by 4 -divide_by 2
\
          -source [get_pins {FCCC_0/CCC_INST/RCOSC_25_50MHZ}] \
          -phase 0 [get_pins {FCCC_0/CCC_INST/GL0}]
```

```
create_generated_clock \
          -name {FCCC_0/GL1} -multiply_by 4 -divide_by 2 \
          -source [get_pins {FCCC_0/CCC_INST/RCOSC_25_50MHZ}] \
          -phase 90 [get_pins {FCCC_0/CCC_INST/GL1}]
```

The following example uses the `-pll_feedback` and the `-pll_output` switches with the `create_generated_clock` constraint when the CCC/PLL is used in the external feedback mode.

Important: The following example is supported by PolarFire and RTG4 devices.

```
create_clock -period 25.000
-waveform {0.000 12.500}
-name {sys_clk} [get_ports \{clk}]

create_generated_clock
-name {PLL_200MHZ}
-multiply_by 5
-source [ get_pins { phase_locked_loop0/pll_pf_40in_40_100_200out_0/pll_inst_0/REF_CLK_0 } ]
-pll_output [ get_pins { phase_locked_loop0/pll_pf_40in_40_100_200out_0/pll_inst_0/OUT0 } ]
-pll_feedback [ get_pins { phase_locked_loop0/pll_pf_40in_40_100_200out_0/pll_inst_0/
FB_CLK } ]
-phase 0 [ get_pins { phase_locked_loop0/pll_pf_40in_40_100_200out_0/pll_inst_0/OUT0 } ]
```

The command in the following example implies that the generated clock will have:

- Rising edge at 2 ns after the first edge of the source clock. This means that the first edge of the source is shifted by 2 ns.
- Falling edge on the third edge of the source clock. This means that the third edge of the source is shifted by 0 ns.
- Next rising edge at 2 ns after the third edge of the source clock. This means that the third edge of the source is shifted by 2 ns.

```
create_generated_clock
-name PULSE
-source [get_ports clk]
-edges {1 3 3}
-edge_shift {2 0 2}  [get_pins NAND1/Y]
```

Note: The three parameters in the `-edges` and `-edge_shift` switches represent, in order, the rising edge, falling edge, and the subsequent rising edge.

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- create_clock
- remove_generated_clock

Related Examples on GitHub

- create_generated_clock

8.8. create_scenario (Ask a Question)

Description

Creates a new timing scenario with the specified name. You must provide a unique name (that is, it cannot already be used by another timing scenario).

A timing scenario is a set of timing constraints used with a design. Scenarios enable you to easily refine the set of timing constraints used for Timing-Driven Place-and-Route, so as to achieve timing closure more rapidly.

This command creates an empty timing scenario with the specified name and adds it to the list of scenarios.

Note: It is recommended to use the `organize_tool_files` command instead of this command.

```
create_scenario name
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the new timing scenario. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a new timing scenario with the "scenario_A" name.

```
create_scenario scenario_A
```

Related Examples on GitHub

- create_scenario

See Also

- clone_scenario
- list_scenario
- remove_scenario
- rename_scenario

© 2025 Microchip Technology Inc. and its subsidiaries

8.9. create_set (Ask a Question)

Description

Creates a set of paths to be analyzed. Use the arguments to specify which paths to include. To create a set that is a subset of a clock domain, specify it with the `-clock` and `-type` arguments. To create a set that is a subset of an inter-clock domain set, specify it with the `-source_clock` and `-sink_clock` arguments. To create a set that is a subset (filter) of an existing named set, specify the set to be filtered with the `-parent_set` argument.

```
create_set\ -name <name>\ -parent_set <name>\ -type <set_type>\ -clock <clock name> \ -
source_clock <clock name>\ -sink_clock <clock name>\ -in_to_out\ -source <port/pin pattern> \
-sink <port/pin pattern>
```

Arguments

Parameter	Type	Description
`name`	string	Specifies a unique name for the newly created path set.
`parent_set`	string	Specifies the name of the set to filter from.
`clock`	string	Specifies that the set is to be a subset of the given clock domain. This argument is valid only if you also specify the `-type` argument.
`type`	string	Specifies the predefined set type on which to base the new path set. You can only use this argument with the `-clock` argument, not by itself. • reg_to_reg - paths between registers in the design. • async_to_reg - paths from asynchronous pins to registers. • reg_to_async - paths from registers to asynchronous pins. • external_recovery - the set of paths from inputs to asynchronous pins. • external_removal - the set of paths from inputs to asynchronous pins. • external_setup - paths from input ports to registers. • external_hold - paths from input ports to registers. • clock_to_out - paths from registers to output ports.
`in_to_out`	None	Specifies that the set is based on the "Input to Output" set, which includes paths that start at input ports and end at output ports.
`source_clock`	string	Specifies that the set will be a subset of an inter-clock domain set with the given source clock. You can only use this option with the `-sink_clock` argument.
`sink_clock`	string	Specifies that the set will be a subset of an inter-clock domain set with the given sink clock. You can only use this option with the `-source_clock` argument.
`source`	string	Specifies a filter on the source pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.
`sink`	string	Specifies a filter on the sink pins of the parent set. If you do not specify a parent set, this option filters all pins in the current design.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example creates set with "my_user_set" name. Filters all C* ports and D* pins in the current design.

```
create_set -name { my_user_set } -source { C* } -sink { D* }
```

The following example creates set with `my_other_user_set` name that is a subset (filter) of an existing "my_user_set" set.

```
create_set -name { my_other_user_set } -parent_set { my_user_set } -source { CL* }
```

The following example creates set with `another_set` name which is the subset of an inter-clock domain set with the given source clock.

```
create_set -name { another_set } -source_clock { EXTERN_CLOCK } \
-sink_clock { MY_GEN_CLOCK }
```

Related Examples on GitHub

- create_set

See Also

- remove_set

8.10. expand_path (Ask a Question)

Description

Displays expanded path information (path details) for paths. The paths to be expanded are identified by the parameters required to display these paths with list_paths. For example, to expand the first path listed with list_paths `-clock {MYCLOCK} -type {register_to_register}`, use the command expand_path `-clock {MYCLOCK} -type {register_to_register}`. Path details contain the pin name, type, net name, cell name, operation, delay, total delay, and edge as well as the arrival time, required time, and slack. These details are the same as details available in the SmartTime Expanded Path window.

```
expand_path \
-index value \
-set name \
-clock clock_name \
-type set_type \
-analysis {max| min} \
-format {csv | text} \
-from_clock clock_name \
-to_clock clock_name
```

Arguments

Parameter	Type	Description
`index value`	list of integers	Specify the index of the path to be expanded in the list of paths and display them. The index starts at 1, and defaults to 1. If index value is less than 1, then it is considered as 1. List of specified indexes can be not sequential. Only the paths with indices lower than the max_paths option value will be expanded.
`analysis {min|max}`	string	Specify whether the timing analysis is done via max-delay (setup check) or min-delay (hold check). Valid values are min/max or mindelay/maxdelay.
`format {csv | text}`	string	Specify the file format of the output. It can be either text—ASCII text format (default) or csv (comma separated values).
`set`	string	Displays a list of paths from the named set. You can either use the `-set` option to specify a user set by its name or use both `-clock` and `-type` to specify a set.

© 2025 Microchip Technology Inc. and its subsidiaries

expand_path (continued)

Parameter	Type	Description
`clock`	string	Displays the set of paths belonging to the specified clock domain. You can either use this option along with `-type` to specify a set or use the `-set` option to specify the name of the set to display.
`type`	string	Specifies the type of paths in the clock domain to display in a list. You can only use this option with the `-clock` option. You can either use this option along with `-clock` to specify a set or use the `-set` option t specify a set name. • reg_to_reg—paths between registers in the design. • async_to_reg—path from asynchronous pins to registers. • reg_to_async—path from registers to asynchronous pins. • external_recovery—set of paths from input ports to asynchronous pins. • external_removal—set of paths from input ports to asynchronous pins. • external_setup—path from input ports to registers. • external_hold—path from input ports to registers. • clock_to_out—path from registers to output ports.
`from_clock`	string	Displays a list of timing paths for an inter-clock domain set belonging to the source clock specified. You can only use this option with the `-to_clock` option, not by itself.
`to_clock`	string	Displays a list of timing paths for an inter-clock domain set belonging to the sink clock specified. You can only use this option with the `-from_clock` option, not by itself.

Return Type	Description
`string`	Displays expanded path information (path details) for paths.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example displays first expanded path information (path details) for paths between registers in the design.

```
puts [expand_path -clock { myclock } -type { reg_to_reg }]
```

The following example displays expanded paths details with 1, 2, and 3 indexes from list of paths.

```
puts [expand_path -clock { myclock } -type { reg_to_reg } -index { 1 2 3 } -format text]
```

Related Examples on GitHub

- expand_path

See Also

- list_paths

© 2025 Microchip Technology Inc. and its subsidiaries

8.11. get_cells (Ask a Question)

Description

Returns a collection of instance (cell) objects in the current design that match a specified search pattern. You can use this command only as part of a `-from`, `-to` argument in the following Tcl commands: `set_max delay`, `set_multicycle_path`, and `set_false_path`. Wildcards can be used to select multiple cells at once. If no objects match the criteria, the empty string is returned.

```
get_cells pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	Specifies the pattern to match the instances to return. For example, `get_cells U18*` returns all instances starting with the characters `U18`, where `*` is a wild card character that represents any character string. This is mandatory.

Return Type	Description
`object`	Returns an object representing the cells (instances) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets maximum delay constraining all paths from reg* cells to out ports with a delay less than 2 ns.

```
set_max_delay 2 -from [get_cells {reg*}] -to [get_ports {out}]
```

Related Examples on GitHub

- get_cells

See Also

- get_clocks
- get_nets
- get_pins
- get_ports

© 2025 Microchip Technology Inc. and its subsidiaries

8.12. get_clocks (Ask a Question)

Description

Returns an object representing the clock(s) that match those specified in the current timing scenario. Wildcards can be used to select multiple clocks at once. If no objects match the criteria, the empty string is returned.

```
get_clocks pattern
```

Arguments

Parameter	Type	Description
pattern	string	Mandatory. Specifies the pattern to match to the SmartTime on which a clock constraint has been set.

Return Type	Description
object	Returns an object representing the clock(s) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets maximum delay constraining all paths from datal port to ck1 clock with a delay less then 2 ns.

```
set_max_delay -from [get_ports datal] -to [get_clocks ck1]
```

Related Examples on GitHub

- get_clocks

See Also

- create_clock
- create_generated_clock

8.13. get_current_scenario (Ask a Question)

Description

Returns the name of the current timing scenario.

```
get_current_scenario
```

Arguments

Parameter	Type	Description
None	None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
string	Name of the current timing scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the name of the current timing scenario.

```
get_current_scenario
```

Related Examples on GitHub

- get_current_scenario

See Also

- create_scenario
- set_current_scenario
- remove_scenario
- rename_scenario

8.14. get_nets (Ask a Question)

Description

Returns a collection of nets matching the pattern you specify. You can only use this command as source objects in create clock (`create_clock`) or create generated clock (`create_generated_clock`) constraints and as `-through` arguments in the set false path, set minimum delay, set maximum delay, and set multicycle path constraints. Wildcards can be used to select multiple nets at once. If no objects match the criteria, the empty string is returned.

```
get_nets pattern
```

Arguments

Parameter	Type	Description
pattern	string	Specifies the pattern to match the names of the nets to return. For example, `get_nets N_255*` returns all nets starting with the characters `N_255`, where `*` is a wildcard that represents any character string. This is mandatory.

Return Type	Description
object	Returns an object representing the nets that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets maximum delay constraining all paths from `RDATA1` port passing `-through net_chkp1 net_chkqi` nets.

```
set_max_delay 2 -from [get_ports RDATA1] -through [get_nets {net_chkp1 net_chkqi}]
```

The following example specifies all paths through the nets Tblk/rm/n* to be false.

```
set_false_path –through [get_nets {Tblk/rm/n*}]
```

The following example creates a clock on cknet net with a period of 2.5 ns.

```
create_clock -name  mainCLK -period 2.5 [get_nets {cknet}]
```

Related Examples on GitHub

- get_nets

See Also

- create_clock
- create_generated_clock
- set_false_path
- set_min_delay
- set_max_delay
- set_multicycle_path

8.15. get_pins (Ask a Question)

Description

Returns an object representing the pin(s) that match those specified in the pattern argument. Wildcards can be used to select multiple pins at once. If no objects match the criteria, the empty string is returned.

```
get_pins pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	Specifies the pattern to match the pins to return. For example, `get_pins clock_gen*` returns all pins starting with the characters `clock_gen`, where `*` is a wildcard that represents any character string. This is mandatory.

Return Type	Description
`object`	Returns an object representing the pin(s) that match those specified in the pattern argument.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a clock on pin `clock_gen/reg2:Q` with a period of 10 ns.

```
create_clock -period 10 [get_pins clock_gen/reg2:Q]
```

Related Examples on GitHub

- get_pins

See Also

- create_clock
- create_generated_clock
- set_false_path
- set_min_delay
- set_max_delay
- set_multicycle_path

8.16. get_ports (Ask a Question)

Description

Returns an object representing the port(s) that match those specified in the pattern argument. Wildcards can be used to select multiple ports at once. If no objects match the criteria, the empty string is returned.

```
get_ports pattern
```

Arguments

Parameter	Type	Description
`pattern`	string	Specifies the pattern to match the ports.

Return Type	Description
`object`	Returns an object representing the port(s) that match those specified in the pattern argument.

Error Codes

Error Code	Description
None	Required parameter `_AtclParam0_` is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example creates a clock on port `CK1` with a period of 10 ns.

```
create_clock -period 10 [get_ports CK1]
```

Related Examples on GitHub

- get_ports

See Also

- create_clock
- create_generated_clock
- set_input_delay
- set_output_delay
- set_false_path
- set_min_delay
- set_max_delay
- set_multicycle_path

8.17. list_clock_groups (Ask a Question)

Description

Returns the details for all the existing clock groups in the current timing constraint scenario.

```
list_clock_groups
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Details about all of the clock groups constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Example

With this command we get the details about all of the existing clock groups in the current timing constraint scenario.

```
puts [list_clock_groups]
```

Related Examples on GitHub

- list_clock_groups

See Also

- set_clock_groups
- remove_clock_groups

8.18. list_clock_latencies (Ask a Question)

Description

Returns details about all of the clock latencies in the current timing constraint scenario.

```
list_clock_latencies
```

Arguments

Return Type	Description
string	Returns details about all of the clock latencies in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the clock latencies in the current timing constraint scenario.

```
puts [list_clock_latencies]
```

Related Examples on GitHub

- list_clock_latencies

See Also

- set_clock_latency
- remove_clock_latency

8.19. list_clock_uncertainties (Ask a Question)

Description

Returns details about all of the clock uncertainties in the current timing constraint scenario.

```
list_clock_uncertainties
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Returns details about all of the clock uncertainties.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the clock uncertainties in the current timing constraint scenario.

```
puts [list_clock_uncertainties]
```

Related Examples on GitHub

- list_clock_uncertainties

See Also

- set_clock_uncertainty
- remove_clock_uncertainty

8.20. list_clocks (Ask a Question)

Description

Returns details about all of the clock constraints in the current timing constraint scenario.

```
list_clocks
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
string	Returns details about all of the clock constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example displays the details about all of the clock constraints in the current timing constraint scenario.

```
puts [list_clocks]
```

Related Examples on GitHub

- list_clocks

See Also

- create_clock
- remove_clock

8.21. list_disable_timings (Ask a Question)

Description

Returns the list of disable timing constraints for the current scenario.

```
list_disable_timings
```

Arguments

Return Type	Description
string	Returns list of disable timing constraints for the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the disable timing constraints in the current timing constraint scenario.

```
puts [list_disable_timings]
```

Related Examples on GitHub

- list_disable_timings

See Also

- set_disable_timing
- remove_disable_timing

8.22. list_false_paths (Ask a Question)

Description

Returns details about all of the false paths in the current timing constraint scenario.

```
list_false_paths
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Return Type	Description
string	Returns details about all of the false paths in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the false paths in the current timing constraint scenario.

```
puts [list_false_paths]
```

Related Examples on GitHub

- list_false_paths

See Also

- set_false_path
- remove_false_path

8.23. list_generated_clocks (Ask a Question)

Description

Returns details about all of the generated clock constraints in the current timing constraint scenario.

```
list_generated_clocks
```

Arguments

Return Type	Description
string	Returns details about all of the generated clock constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example displays the details about all of the generated clock constraints in the current timing constraint scenario.

```
puts [list_generated_clocks]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Related Examples on GitHub

- list_generated_clocks

See Also

- create_generated_clock
- remove_generated_clock

8.24. list_input_delays (Ask a Question)

Description

Returns details about all of the input delay constraints in the current timing constraint scenario.

```
list_input_delays
```

Arguments

Return Type	Description
string	Details about all of the input delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the input delay constraints in the current timing constraint scenario.

```
puts [list_input_delays]
```

Related Examples on GitHub

- list_input_delays

See Also

- remove_input_delay
- set_input_delay

8.25. list_max_delays (Ask a Question)

Description

Returns details about all of the maximum delay constraints in the current timing constraint scenario.

```
list_max_delays
```

Arguments

Return Type	Description
string	Details about all of the max delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the maximum delay constraints in the current timing constraint scenario.

```
puts [list_max_delays]
```

Related Examples on GitHub

- list_max_delays

See Also

- remove_max_delay
- set_max_delay

8.26. list_min_delays (Ask a Question)

Description

Returns details about all of the minimum delay constraints in the current timing constraint scenario.

```
list_min_delays
```

Arguments

Return Type	Description
`string`	Details about all of the min delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the minimum delay constraints in the current timing constraint scenario.

```
puts [list_min_delays]
```

Related Examples on GitHub

- list_min_delays

See Also

- remove_min_delay
- set_min_delay

© 2025 Microchip Technology Inc. and its subsidiaries

8.27. list_multicycle_paths (Ask a Question)

Description

Returns details about all of the multicycle paths in the current timing constraint scenario.

```
list_multicycle_paths
```

Arguments

Return Type	Description
string	Returns details about all of the multicycle paths in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the multicycle paths constraints in the current timing constraint scenario.

```
puts [list_multicycle_paths]
```

Related Examples on GitHub

- list_multicycle_paths

See Also

- remove_multicycle_path
- set_multicycle_path

8.28. list_objects (Ask a Question)

Description

Returns a list of object matching the parameter. Objects can be nets, pins, ports, clocks, or instances.

```
list_objects <object>
```

Arguments

Parameter	Type	Description
objects	string	Any timing constraint parameter (object can be nets, pins, ports, clocks, or instances). This is mandatory.

Return Type	Description
list of objects	Returns a list of nets, pins, ports, clocks, or instances.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example lists all the inputs in your design.

```
list_objects [all_inputs]
```

You can also use wildcards to filter your list, as in the following command.

```
list_objects [get_ports a*]
```

Related Examples on GitHub

- list_objects

8.29. list_output_delays (Ask a Question)

Description

Returns details about all of the output delay constraints in the current timing constraint scenario.

```
list_output_delays
```

Arguments

Return Type	Description
`string`	Details about all of the output delay constraints in the current timing constraint scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the details about all of the output delay constraints in the current timing constraint scenario.

```
puts [list_output_delays]
```

Related Examples on GitHub

- list_output_delays

See Also

- remove_output_delay
- set_output_delay

© 2025 Microchip Technology Inc. and its subsidiaries

8.30. list_paths (Ask a Question)

Description

Returns a list of the n worst paths matching the arguments. The number of paths returned can be changed using the `set_options -limit_max_paths` <value> command.

```
list_paths \
-analysis <max | min> \
-format <csv | text> \
-set <name> \
-clock <clock name> \
-type <set_type> \
-from_clock <clock name> \
-to_clock <clock name> \
-in_to_out \
-from <port/pin pattern> \
-to <port/pin pattern>
```

Arguments

Parameter	Type	Description
`analysis`	string	Specifies whether the timing analysis is done for `max-delay` (setup check) or `min-delay` (hold check). Valid values are: max or min.
`format`	string	Specifies the list format. It can be either text (default) or csv (comma separated values). Text format is better for display and csv format is better for parsing.
`set`	string	Returns a list of paths from the named set. You can either use the `-set` option to specify a user set by its name or use both `-clock` and `-type` to specify a set.
`clock`	string	Returns a list of paths from the specified clock domain. This option requires the `-type` option. You cannot use wildcards when specifying a clock name.
`type`	string	Specifies the type of paths to be included. It can only be used along with `-clock`. Valid values are: • `reg_to_reg` -paths between registers in the design. • `async_to_reg` -paths from asynchronous pins to registers. • `reg_to_async` -paths from registers to asynchronous pins of registers. • `external_recovery` -paths from input ports to asynchronous pins of registers. • `external_removal` -paths from input ports to asynchronous pins of registers. • `external_setup` -paths from input ports to data pins of registers. • `external_hold` -paths from input ports to data pins of registers. • `clock_to_out` -paths from registers to output ports.
`from_clock`	string	Used along with `-to_clock` to get the list of paths of the inter-clock domain between the two clocks.
`to_clock`	string	Used along with `-from_clock` to get the list of paths of the inter-clock domain between the two clocks.
`in_to_out`	None	Used to get the list of path between input and output ports.
`from`	string	Filter the list of paths to those starting from ports or pins matching the pattern.
`to`	string	Filter the list of paths to those ending at ports or pins matching the pattern.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
list of strings	Returns a list of the n worst paths matching the arguments.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command displays the list of register to register paths of clock domain clk1.

```
puts [ list_paths -clock clk1 -type reg_to_reg ]
```

Related Examples on GitHub

- list_paths

8.31. list_scenario (Ask a Question)

Description

Returns a list of names of all of the available timing scenarios.

```
list_scenario name
```

Arguments

Return Type	Description
list of strings	List of all names of the available timing scenarios.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

With this command we get the list of available timing scenario names.

```
list_scenario
```

Related Examples on GitHub

- list_scenario

See Also

- clone_scenario
- set_current_scenario
- get_current_scenario

© 2025 Microchip Technology Inc. and its subsidiaries

- remove_scenario
- rename_scenario

8.32. read_sdc (Ask a Question)

Description

Evaluates an SDC file, adding all constraints to the specified scenario (or the current/default one if none is specified). Existing constraints are removed if `-add` is not specified.

```
read_sdc \
-add \
-scenario scenario_name \
-netlist ( user | optimized ) \
-pin_separator ( : | / ) \
-ignore_errors file_name
```

Table 8-3. Arguments

Parameter	Type	Description
`add`	None	Specifies that the constraints from the SDC file is added on top of the existing ones, overriding them in case of a conflict. If not used, the existing constraints are removed before the SDC file is read.
`scenario`	string	Specifies the scenario to add the constraints to. The scenario is created if none exists with this name.
`netlist`	string	Specifies whether the SDC file contains object defined at the post-synthesis netlist (user) level or physical (optimized) netlist (used for timing analysis).
`pin_separator`	char	Specify the pin separator used in the SDC file. It can be either ':' or '/'.
`ignore_errors`	None	Optional. Specifies whether to avoid reporting errors for derived constraints targeting the logic that becomes invalid due to logic optimization. It is an optional argument. Some IPs may have extra logic present depending on other IPs used in the design but the synthesis tool will remove this logic if fewer IPs were used. In such cases, the implementation flow will halt without -ignore_errors flag. **Note:** Do not use this flag outside similar use cases.
`file`	string	Specify the SDC file name.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command removes all constraints from the current/default scenario and adds all constraints from `design.sdc` file to it.

```
read_sdc design.sdc
```


© 2025 Microchip Technology Inc. and its subsidiaries

8.33. remove_all_constraints (Ask a Question)

Description

Removes all timing constraints from analysis.

```
remove_all_constraints
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example removes all timing constraints from analysis.

```
remove_all_constraints
```

Related Examples on GitHub

- remove_all_constraints

See Also

- check_constraints

8.34. remove_clock (Ask a Question)

Description

Removes the specified clock constraint from the current timing scenario. If the specified name does not match a clock constraint in the current scenario, or if the specified ID does not refer to a clock constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

```
remove_clock -name clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the clock constraint to remove from the current scenario. Specify either a clock name or an ID. **Note:** Specify clock name as {CLK}, not [get_clocks {CLK}].
id	integer	Specifies the ID of the clock constraint to remove from the current scenario. Specify either an ID or a clock name that exists in the current scenario.

Error Codes

Error Code	Description
None	Invalid clock name argument.
None	Only one argument is needed.

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a clock names.

Example
The following example removes the clock constraint named `my_user_clock`.

```
remove_clock -name my_user_clock
```

The following example removes the clock constraint using its ID.

```
set clockId [create_clock -name my_user_clock -period 2]
```

```
remove_clock -id $clockId
```

Related Examples on GitHub
- remove_clock

See Also
- create_clock
- create_generated_clock

8.35. remove_clock_groups (Ask a Question)

Description
Removes a clock group by specifying its name or its group ID. If the arguments do not match, or if the ID does not refer to a clock group, the command fails.
Note: The exclusive flag is not needed when removing a clock group by ID. These flags are mutually exclusive. Only one can be specified.

```
remove_clock_groups [-id constraint_ID | -name groupname ] \
[-physically_exclusive | -logically_exclusive | -asynchronous]
```

Arguments

Parameter	Type	Description
id	integer	Specifies the clock group by the ID. You must specify either a clock group ID or a clock group name that exists in the current scenario.
name	string	Specifies the clock group by name (to be always followed by the exclusive flag).
physically_exclusive	None	Specifies that the clock groups are physically exclusive with respect to each other.
logically_exclusive	None	Specifies that the clocks groups are logically exclusive with respect to each other.
asynchronous	None	Specifies that the clock groups are asynchronous with respect to each other.

Error Codes

Error Code	Description
None	Only one argument is needed.

© 2025 Microchip Technology Inc. and its subsidiaries

remove_clock_groups (continued)	
Error Code	**Description**
None	Invalid clock Groups name argument.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock groups name.

Example

The following commands removes clock groups with the `mygroup3` names and the clock groups with id 12.

```
remove_clock_groups -name mygroup3 -physically_exclusive
```

```
remove_clock_groups id 12
```

Related Examples on GitHub

- remove_clock_groups

See Also

- set_clock_groups
- list_clock_groups

8.36. remove_clock_latency (Ask a Question)

Description

Removes a clock source latency from the specified clock and from all edges of the clock. If the specified name does not match a generated clock constraint in the current scenario, or if the specified ID does not refer to a generated clock constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

```
remove_clock_latency -source clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
`source`	`string`	Specifies either the clock name or source name of the clock constraint from which to remove the clock source latency. You must specify either a clock name or an ID.
`id`	`integer`	Specifies the ID of the clock constraint to remove the clock source latency from the current scenario. You must specify either an ID or a clock name that exists in the current scenario.

Error Codes

Error Code	Description
None	Only one argument is needed.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying the clock names.

Example

The following example removes the clock source latency from the specified clock.

```
remove_clock_latency -source [get_clocks {my_clock} ]
```

Related Examples on GitHub

- remove_clock_latency

See Also

- set_clock_latency

8.37. remove_clock_uncertainty (Ask a Question)

Description

Removes a clock-to-clock uncertainty from the current timing scenario by specifying either its exact arguments or its ID.

If the specified arguments do not match clocks with an uncertainty constraint in the current scenario, or if the specified ID does not refer to a clock-to-clock uncertainty constraint, this command fails. Do not specify both the exact arguments and the ID.

```
remove_clock_uncertainty -from | -rise_from | -fall_from from_clock_list -to | -rise_to | \
-fall_to to_clock_list -setup {value} -hold {value} | -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. Only one of the -from, -rise_from, or -fall_from arguments can be specified for the constraint to be valid.
rise_from	list of strings	Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. Only one of the -from, -rise_from, or -fall_from arguments can be specified for the constraint to be valid.
fall_from	list of strings	Specifies that the clock-to-clock uncertainty applies only to falling edges of the source clock list. Only one of the -from, -rise_from, or -fall_from arguments can be specified for the constraint to be valid.
from_clock_list	list of strings	Specifies the list of clock names as the uncertainty source.
to	list of strings	Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. Only one of the -to, -rise_to, or -fall_to arguments can be specified for the constraint to be valid.

© 2025 Microchip Technology Inc. and its subsidiaries

remove_clock_uncertainty (continued)		
Parameter	**Type**	**Description**
rise_to	list of strings	Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. Only one of the -to, -rise_to, or -fall_to arguments can be specified for the constraint to be valid.
fall_to	list of strings	Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. Only one of the -to, -rise_to, or -fall_to arguments can be specified for the constraint to be valid.
to_clock_list	list of strings	Specifies the list of clock names as the uncertainty destination.
setup	None	Specifies that the uncertainty applies only to setup checks. If none or both -setup and -hold are present, the uncertainty applies to both setup and hold checks.
hold	None	Specifies that the uncertainty applies only to hold checks. If none or both -setup and -hold are present, the uncertainty applies to both setup and hold checks.
id	integer	Specifies the ID of the clock constraint to remove the clock source uncertainty from the current scenario. You must specify either the exact parameters to set the constraint or its constraint ID.

Error Codes

Error Code	Description
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example removes uncertainties from Clk1 clock to Clk2 clock domains.

```
remove_clock_uncertainty -from [ get-clock {Clk1} ] -to [ get_clock {Clk2} ]
```

The following example removes uncertainties between Clk1, Clk2, Clk3, and Clk4 clock domains with specific edges.

```
remove_clock_uncertainty -from [ get_clocks {Clk1} ]  -fall_to [ get_clocks {Clk2 Clk3} ]
-setup
```

```
remove_clock_uncertainty 4.3 -fall_from [ get_clocks {Clk1 Clk2} ] -rise_to [ get_clocks {*} ]
```

```
remove_clock_uncertainty 0.1 -rise_from [ get_clocks {Clk1 Clk2} ] \
-fall_to [ get_clocks {Clk3 Clk4} ] -setup
```

```
remove_clock_uncertainty 5 -rise_from [ get_clocks {Clk1} ] -to [ get_clocks {*} ]
```

```
remove_clock_uncertainty -id $clockId
```

Related Examples on GitHub

- remove_clock_uncertainty

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- list_clock_uncertainties
- set_clock_uncertainty

8.38. remove_disable_timing (Ask a Question)

Description

Removes a disable timing constraint by specifying its arguments, or its ID. If the arguments do not match a disable timing constraint, or if the ID does not refer to a disable timing constraint, the command fails.

```
remove_disable_timing -from value -to value name -id constraint_ID
```

Arguments

Parameter	Type	Description
from	string	Specifies the starting port. The -from and -to arguments must either both be present or both omitted for the constraint to be valid.
name	string	Specifies the cell(instance) name where the disable timing constraint will be removed. It is an error to supply both a cell name and a constraint ID, as they are mutually exclusive. No wildcards are allowed when specifying a clock name, either alone or in an accessor command.
to	string	Specifies the ending port. The -from arguments must either both be present or both omitted for the constraint to be valid.
id	string	Specifies the constraint name where the disable timing constraint will be removed. It is an error to supply both a cell name and a constraint ID, as they are mutually exclusive. No wildcards are allowed when specifying a clock name, either alone or in an accessor command.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as get_pins or get_ports.

Example

The following example removes disable timing constraint between A and Y ports.

```
remove_disable_timing -from A -to Y -id new_constraint
```

Related Examples on GitHub

- remove_disable_timing

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- set_disable_timing
- list_disable_timings

8.39. remove_false_path (Ask a Question)

Description

Removes a false path constraint from the current timing scenario by specifying either its exact arguments or its ID. If the arguments do not match a false path constraint in the current scenario, or if the specified ID does not refer to a false path constraint, this command fails.
Note: Do not specify both false path arguments and the constraint ID.

```
remove_false_path [-from from_list] [-to to_list] [-through through_list] | -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
through	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the disabled paths must pass.
to	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
id	integer	Specifies the ID of the false path constraint to remove from the current scenario. You must specify either the exact false path arguments to remove or the constraint ID that refers to the false path constraint to remove.

Error Codes

Error Code	Description
None	Invalid arguments -from/-to/-through.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as get_pins or get_ports.

Example

The following example specifies all false path to remove.

```
remove_false_path -through U0/U1:Y
```


© 2025 Microchip Technology Inc. and its subsidiaries

The following example removes the false path constraint using its id.

```
set fpId [set_false_path -from [get_clocks c*] -through {topx/reg/*} \
-to [get_ports out15] ]
```

```
remove_false_path -id $fpId
```

Related Examples on GitHub

- remove_false_path

See Also

- set_false_path

8.40. remove_generated_clock (Ask a Question)

Description

Removes the specified generated clock constraint from the current scenario. If the specified name does not match a generated clock constraint in the current scenario, or if the specified ID does not refer to a generated clock constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

```
remove_generated_clock -name clock_name | -id constraint_ID
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the generated clock constraint to remove from the current scenario. You must specify either a clock name or an ID.
id	integer	Specifies the ID of the generated clock constraint to remove from the current scenario. You must specify either an ID or a clock name that exists in the current scenario.

Error Codes

Error Code	Description
None	Invalid clock name argument.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock name.

Example

The following example removes the generated clock constraint named my_user_clock

```
remove_generated_clock -name my_user_clock
```


© 2025 Microchip Technology Inc. and its subsidiaries

Related Examples on GitHub

- remove_generated_clock

See Also

- create_generated_clock

8.41. remove_input_delay (Ask a Question)

Description

Removes an input delay by specifying both the clocks and port names or the ID of the input delay constraint to remove. If the clocks and port names do not match an input delay constraint in the current scenario, or if the specified ID does not refer to an input delay constraint, this command fails.

Do not specify both the clock and port names and the constraint ID.

```
remove_input_delay -clock clock_name port_pin_list | -id constraint_ID
```

Arguments

Parameter	Type	Description
clock	string	Specifies the clock name to which the specified input delay value is assigned. **Note:** You must specify clock name as {CLK}, not [get_clocks {CLK}].
port_pin_list	list of strings	Specifies the port names to which the specified input delay value is assigned.
id	integer	Specifies the ID of the clock with the input_delay value to remove from the current scenario. You must specify either both a clock name and list of port names or the input_delay constraint ID.

Error Codes

Error Code	Description
None	Parameter -clock has illegal value.
None	Invalid clock/port arguments.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock or port names, either alone or in an accessor command.

Example

The following example removes the input delay from CLK1 on port data1.

```
remove_input_delay -clock [get_clocks CLK1] [get_ports data1]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Related Examples on GitHub

- remove_input_delay

See Also

- set_input_delay

8.42. remove_max_delay (Ask a Question)

Description

Removes a maximum delay constraint from the current timing scenario by specifying either its exact arguments or its ID. If the arguments do not match a maximum delay constraint in the current scenario, or if the specified ID does not refer to a maximum delay constraint, this command fails.

Do not specify both maximum delay arguments and the constraint ID.

```
remove_max_delay [-from from_list] [-to to_list] [-through through_list]
```

```
remove_max_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
through	list of strings	Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.
to	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
id	integer	Specifies the ID of the maximum delay constraint to remove from the current scenario. You must specify either the exact maximum delay arguments to remove or the constraint ID that refers to the maximum delay constraint to remove.

Error Codes

Error Code	Description
None	Invalid arguments -from/-to/-through.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a clock or port name, either alone or in an accessor command.

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example specifies a range of maximum delay constraints to remove.

```
remove_max_delay -through U0/U1:Y
```

Related Examples on GitHub

- remove_max_delay

See Also

- set_max_delay
- set_min_delay
- remove_min_delay

8.43. remove_min_delay (Ask a Question)

Description

Removes a minimum delay constraint from the current timing scenario by specifying either its exact arguments or its ID. If the arguments do not match a minimum delay constraint in the current scenario, or if the specified ID does not refer to a minimum delay constraint, this command fails.

Do not specify both minimum delay arguments and the constraint ID.

```
remove_min_delay [-from from_list] [-to to_list] [-through through_list]
```

```
remove_min_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
through	list of strings	Specifies a list of pins, ports, cells, or nets through, which the disabled paths must pass.
to	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
id	integer	Specifies the ID of the minimum delay constraint to remove from the current scenario. You must specify either the exact minimum delay arguments to remove or the constraint ID that refers to the minimum delay constraint to remove.

Error Codes

Error Code	Description
None	Invalid arguments -from/-to/-through.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Exceptions

You cannot use wildcards when specifying a clock or port name, either alone or in an accessor command.

Example

The following example specifies a range of minimum delay constraints to remove.

```
remove_min_delay -through U0/U1:Y
```

Related Examples on GitHub

- remove_min_delay

See Also

- set_min_delay

8.44. remove_multicycle_path (Ask a Question)

Description

Removes a multicycle path constraint from the current timing scenario by specifying either its exact arguments or its ID. If the arguments do not match a multicycle path constraint in the current scenario, or if the specified ID does not refer to a multicycle path constraint, this command fails.
Note: Do not specify both multicycle path arguments and the constraint ID.

```
remove_multicycle_path [-from from_list] [-to to_list] [-through through_list]
```

```
remove_multicycle_path -id constraint_ID
```

Arguments

Parameter	Type	Description
from	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
through	list of strings	Specifies a list of pins, ports, cells, or nets through which the disabled paths must pass.
to	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
id	string	Specifies the ID of the multicycle path constraint to remove from the current scenario. Specify either the exact multicycle path arguments to remove or the constraint ID that refers to the multicycle path constraint to remove.

Error Codes

Error Code	Description
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Exceptions

You cannot use wildcards when specifying a clock name, either alone or in an accessor command such as `get_pins` or `get_ports`.

Example

The following example specifies all the multicycle paths between reg1 and reg2.

```
remove_multicycle_path -from [get_pins {reg1}] -to [get_pins {reg2}]
```

Related Examples on GitHub

- remove_multicycle_path

See Also

- set_multicycle_path

8.45. remove_output_delay (Ask a Question)

Description

Removes an output delay by specifying both the clocks and port names or the ID of the output_delay constraint to remove. If the clocks and port names do not match an output delay constraint in the current scenario, or if the specified ID does not refer to an output delay constraint, this command fails.

Note: Do not specify both the clock and port names and the constraint ID.

```
remove_output_delay -clock clock_name port_pin_list
```

```
remove_output_delay -id constraint_ID
```

Arguments

Parameter	Type	Description
`clock`	string	Specifies the clock name to which the specified output delay value is assigned. **Note:** You must specify clock name as `{CLK}`, not `[get_clocks {CLK}]`.
`port_pin_list`	list of strings	Specifies the port names to which the specified output delay value is assigned.
`id`	integer	Specifies the ID of the clock with the output_delay value to remove from the current scenario. You must specify either both a clock name and list of port names or the output_delay constraint ID.

Error Codes

Error Code	Description
None	Parameter -clock has illegal value.
None	Invalid clock/port arguments.
None	Only one argument is needed.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

Exceptions
You cannot use wildcards when specifying a clock or port names, either alone or in an accessor command.

Example
The following example removes the output delay from `CLK1` on port `out1`.

```
remove_output_delay -clock {CLK1} [get_ports out1]
```

Related Examples on GitHub
- remove_output_delay

See Also
- set_output_delay

8.46. remove_scenario (Ask a Question)

Description
Removes a scenario from the constraint database and removes it to the list of scenarios.

```
remove_scenario name
```

Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the scenario to delete. This is mandatory.

Error Codes

Error Code	Description
None	Required parameter `_AtclParam0_` is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions
You cannot use wildcards when specifying a set name to remove.

Example
The following command removes the scenario named `my_scenario`.

```
remove_scenario my_scenario
```

Related Examples on GitHub
- remove_scenario

See Also
- create_scenario
- clone_scenario

© 2025 Microchip Technology Inc. and its subsidiaries

- rename_scenario

8.47. remove_set (Ask a Question)

Description

Removes a set of paths from analysis. Only user-created sets can be deleted.

```
remove_set -name name
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the set of paths to delete.

Error Codes

Error Code	Description
None	Required parameter -name is missing.
None	Unable to find set.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

You cannot use wildcards when specifying a set name.

Example

The following command removes the set named my_set.

```
remove_set -name my_set
```

Related Examples on GitHub

- remove_set

See Also

- create_set

8.48. rename_scenario (Ask a Question)

Description

Renames an existing timing scenario to a new name. The new name you provide must be unique and cannot be used by another timing scenario.

Note: It is recommended to use the organize_tool_files command instead of this command.

```
rename_scenario old_name new_name
```

Arguments

Parameter	Type	Description
old_name	string	Specifies the name of the existing timing scenario to be renamed.

© 2025 Microchip Technology Inc. and its subsidiaries

rename_scenario (continued)		
Parameter	**Type**	**Description**
new_name	string	Specifies the new name for the new scenario.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command renames the my_old_scenarioscenario name into a my_new_scenario new name.

```
rename_scenario my_old_scenario my_new_scenario
```

Related Examples on GitHub

- rename_scenario

See Also

- create_scenario
- clone_scenario
- remove_scenario

8.49. report (Ask a Question)

Description

Specifies the type of reports to be generated and the type of Analysis (Max Delay or Min Delay) performed to generate the reports. Using this command, you can generate the following types of reports:

- Timer report—This report displays the timing information organized by clock domain.
- Timing Violations report—This flat slack report provides information about constraint violations.
- Bottleneck report—This report displays the points in the design that contribute to the most timing violations.
- Datasheet report—This report describes the characteristics of the pins, I/O technologies, and timing properties in the design.
- Constraints Coverage report—This report displays the overall coverage of the timing constraints set on the current design.
- Combinational Loop report—This report displays loops found during initialization.
- Clock Domain Crossing report—This report analyzes timing paths that cross from one clock domain (the source clock) to another clock domain (the destination clock).

If the specified parameter/value is not correct, this command fails.

```
report -type (timing | violations | datasheet | bottleneck | constraints_coverage |
combinational_loops | cdc) \ -analysis <max|min> \ -format (csv|text) \ <filename> \
timing options \ -max_parallel_paths <number> \ -max_paths <number> \ -print summary (yes|
no) \ -use_slack_threshold (yes|no) \ -slack threshold <double> \ -print_paths (yes|no) \
-max_expanded_paths <number> \ -include_user_sets (yes|no) \ -include_clock_domains (yes|no)
\ -select_clock_domains <clock name list> \ -limit_max_paths (yes|no) \ -include_pin_to_pin
(yes|no) \ bottleneck options \ -cost_type (path_count|path_cost) \ -max_instances <number>
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
\ -from <port/pin pattern> \ -to <port/pin pattern> \ -set_type <set_type> \ -set_name <set
name> \ -clock <clock name> \ -from_clock <clock name> \ -to_clock <clock name> \ -in_to_out \
```

Arguments

Parameter	Type	Description
type	string	Specifies the type of the report to be generated. It is mandatory. • timing—Timing Report • violations—Timing Violation Report • datasheet—Datasheet Report • bottleneck—Bottleneck Report • constraints_coverage—Constraints Coverage Report • combinational_loops—Combinational Loops Report
analysis	string	Specifies the type of Analysis (Max Delay or Min Delay) Performed to generate the reports. It is optional. **Note:** This argument should not be used to generate datasheet reports. The command may fail if this argument generate datasheet report. • max—Timing report considers maximum analysis (default). • min—Timing report considers minimum analysis.
format	string	Specifies the format in which the report is generated. It is optional. • text—Generates a text report (default). • csv—Generates the report in a comma-separated value format which you can import into a spreadsheet. **Note:** CDC type generates report in CSV format only.
filename	string	Specifies the file name of the generated report. It is mandatory.

Table 8-4. Timing Options and Values

Parameter/Value	Description
max_parallel_paths <number>	Specifies the max number of parallel paths. Parallel paths are timing paths with the same start and end points.
max_paths <number>	Specifies the max number of paths to display for each set. This value is a positive integer value greater than zero. Default is 100.
print_summary (yes|no)	Yes to include and No to exclude the summary section in the timing report.
use_slack_threshold (yes|no)	Yes to include slack threshold and no to exclude threshold in the timing report. The default is to exclude slack threshold.
slack_threshold <double>	Specifies the threshold value to consider when reporting path slacks. This value is in nanoseconds (ns). By default, there is no threshold (all slacks reported).
print_paths <yes|no>	Specifies whether the path section (clock domains and in-to-out paths) will be printed in the timing report. Yes to include path sections (default) and no to exclude path sections from the timing report.
max_expanded_paths <number>	Specifies the max number of paths to expand per set. This value is a positive integer value greater than zero. Default is 100.
include_user_sets (yes|no)	If yes, the user set is included in the timing report. If no, the user set is excluded in the timing report.
include_clock_domains (yes|no)	Yes to include and no to exclude clock domains in the timing report.

© 2025 Microchip Technology Inc. and its subsidiaries

Table 8-4. Timing Options and Values (continued)

Parameter/Value	Description

`select_clock_domains <clock_name_list>`
or
`Select_clock_domains -yes -clock_domain <clock_name_list>`
	Defines the clock domain to be considered in the clock domain section. The domain list is a series of strings with domain names separated by spaces. Both the summary and the path sections in the timing report display only the listed clock domains in the `clock_name_list`.

`limit_max_paths (yes|no)`	Yes to limit the number of paths to report. No to specify that there is no limit to the number of paths to report (default).
`include_pin_to_pin (yes|no)`	Yes to include and no to exclude pin-to-pin paths in the timing report.

Table 8-5. Bottleneck Options and Values

`cost_type <path_count|path_cost>`	Specifies the cost_type as either path_count or path_cost. For path_count, instances with the greatest number of path violations will have the highest bottleneck cost. For path_cost, instances with the largest combined timing violations will have the highest bottleneck cost.
`max_instances <number>`	Specifies the maximum number of instances to be reported. Default is 10.
`from <port/pin pattern>`	Reports only instances that lie on violating paths that start at locations specified by this option.
`to <port/pin pattern>`	Reports only instances that lie on violating paths that end at locations specified by this option.
`clock <clock name>`	This option allows pruning based on a given clock domain. Only instances that lie on these violating paths are reported.
`set_name <set name>`	Displays the bottleneck information for the named set. You can either use this option or use both -clock and -type. This option allows pruning based on a given set. Only paths that lie within the named set will be considered towards bottleneck.
`set_type <set_type>`	This option can only be used in combination with the -clock option, and not by itself. The options allows you to filter which type of paths should be considered towards the bottleneck: • reg_to_reg—Paths between registers in the design • async_to_reg—Paths from asynchronous pins to registers • reg_to_async—Paths from registers to asynchronous pins • external_recovery—The set of paths from inputs to asynchronous pins • external_removal—The set of paths from inputs to asynchronous pins • external_setup—Paths from input ports to registers • external_hold—Paths from input ports to registers • clock_to_out—Paths from registers to output ports
`from_clock <clock name>`	Reports only bottleneck instances that lie on violating timing paths of the inter-clock domain that starts at the source clock specified by this option. This option can only be used in combination with to_clock.
`to_clock <clock name>`	Reports only instances that lie on violating paths that end at locations specified by this option.
`in_to_out`	Reports only instances that lie on violating paths that begin at input ports and end at output ports.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
file	Generates SmartTime report file with the specified format.

Error Codes

Error Code	Description
None	Parameter -type has illegal value
None	Required parameter -type is missing
None	Required parameter _AtclParam0_ is missing

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example generates a timing violation report named `timing_viol.txt`. The report considers an analysis using maximum delays and does not filter paths based on slack threshold. It reports two paths per section and one expanded path per section:

```
report \ -type violations \ -analysis max \ -use_slack_threshold no \ -limit_max_paths yes \
-max_paths 2 \ -max_expanded_paths 1 \ timing_viol.txt
```

The following example generates a datasheet report named `datasheet.csv` in CSV format:

```
report -type datasheet -format csv datasheet.csv
```

You can use any one of the following examples to report inter-clock domain timing paths:

```
report -type timing -select_clock_domains yes -clock_domain { clk1 clk2 } report1.txt
```

or

```
report -type timing -select_clock_domains { clk1 clk2 } report2.txt
```

Related Examples on GitHub

- report

8.50. save (Ask a Question)

Description

Saves all changes made prior to this command. This includes changes made on constraints, options, and sets.

```
save
```

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

IGLOO® 2
RTG4™

Example

The following script sets the maximum number of paths reported by list_paths to 10, reads an SDC file, and save both the option and the constraints into the design project.

```
set_options -limit_max_paths 10
read_sdc somefile.sdc
save
```

Related Examples on GitHub

- save

8.51. set_clock_groups (Ask a Question)

Description

Disables timing analysis between the specified clock groups. No paths are reported between the clock groups in both directions. Paths between clocks in the same group continue to be reported.

Important: If you use the same name and the same exclusive flag of a previously defined clock group to create a new clock group, the previous clock group is removed and a new one is created in its place. The exclusive flags for the arguments above are all mutually exclusive. Only one can be specified.

```
set_clock_groups [-name name ] \
[-physically exclusive | -logically exclusive | -asynchronous] \
[-comment comment_string ] \
-group clock_list
```

Arguments

Parameter	Type	Description
name	string	Name given to the clock group.
physically_exclusive	None	Specifies that the clock groups are physically exclusive with respect to each other. Examples are multiple clocks feeding a register clock pin. **Important:** The exclusive flags are all mutually exclusive. Only one can be specified.
logically_exclusive	None	Specifies that the clocks groups are logically exclusive with respect to each other. Examples are clocks passing through a MUX.
asynchronous	None	Specifies that the clock groups are asynchronous with respect to each other, as there is no phase relationship between them. **Important:** The exclusive flags are all mutually exclusive. Only one can be specified.
group	list of strings	Specifies a list of clocks. There can any number of groups specified in the set_clock_groups command.

Return Type	Description
integer	Returns the ID of the clock group.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

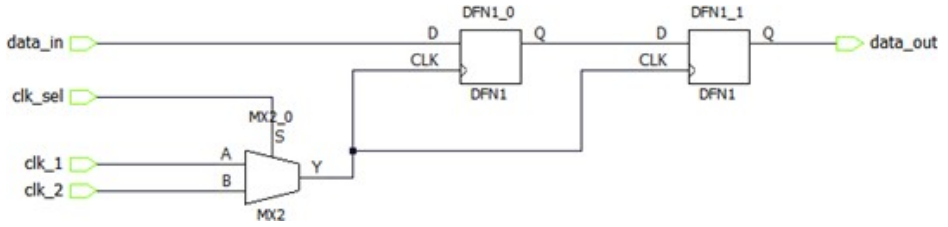
Error Code	Description
None	Invalid set_clock_groups constraint - only one of -physically_exclusive, -logically_exclusive, or -asynchronous should be used.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

1. The following figure shows how to use the `set_clock_groups` constraint for multiplexed clocks..

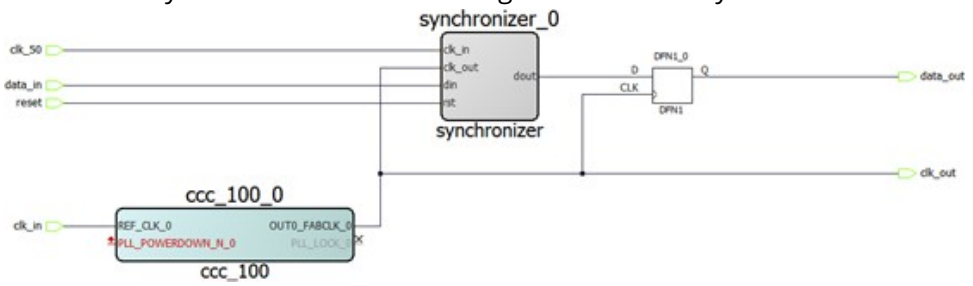


SDC:

```
create_clock -name clk_1 -period 5 [ get_ports clk_1 ]
create_clock -name clk_2 -period 10 [ get_ports clk_2 ]
set_clock_groups -logically_exclusive -group clk_1 -group clk_2
```

This command implies that `clk_1` is asynchronous to `clk_2`.

2. Here, there are three synchronous clocks receiving data from an asynchronous clock.



SDC:

```
create_clock -name clk_in -period 10 [ get_ports clk_in ]
create_clock -name clk_50 -period 20 [ get_ports clk_50 ]
create_generated_clock -name ccc_100 -divide_by 2 \
-source [ get_pins ccc_100_0/ccc_100_0/pll_inst_0/REF_CLK_0 ] \
[ get_pins ccc_100_0/ccc_100_0/pll_inst_0/OUT0 ] \
create_generated_clock -name clk_out -divide_by 1 \
-source [ get_pins { ccc_100_0/ccc_100_0/pll_inst_0/OUT0 } ] \
[ get_ports clk_out ]
set_clock_groups -asynchronous -group { clk_in ccc_100 clk_out } -group clk_50
```

This command implies the following:

- `clk_in` is asynchronous to `clk_50`.

© 2025 Microchip Technology Inc. and its subsidiaries

- ccc_100 is asynchronous to clk_50.
- clk_out is asynchronous to clk_50.
- The place and route engine and the timing tool treat the paths between clk_in, ccc_100, and clk_out as synchronous.

Related Examples on GitHub

- set_clock_groups

See Also

- list_clock_groups
- remove_clock_groups

8.52. set_clock_latency (Ask a Question)

Description

Defines the delay between an external clock source and the definition pin of a clock within SmartTime.

Clock source latency defines the delay between an external clock source and the definition pin of a clock within SmartTime. You can specify both an "early" delay and a "late" delay for this latency, providing an uncertainty which SmartTime propagates through its calculations. Rising and falling edges of the same clock can have different latencies. If only one value is provided for the clock source latency, it is taken as the exact latency value, for both rising and falling edges.

Clock latency is of two types:

- External latency (source) - From an external source to on-chip clock definition point using the -source argument.
- Internal latency (network) - From a clock generator to an end point (FF) due to clock tree synthesis (CTS).

In FPGA, internal clock latency is not used because the clock tree is already inserted and Libero SoC tool is already aware of the delay. Hence, set_clock_latency is primarily used to model external latency in FPGA.

When external clock latency is modeled using the -source arguments, hold calculations are never impacted because the hold checks occur for the same clock edge. Setup times are only impacted when the -early and -late arguments are used because otherwise the clock is uniformly delayed to all endpoints in the design. The following table summarizes the behavior:

-source used	-early used	-late used	Result
Yes	No	No	No setup or hold impact
Yes	No	Yes	No setup or hold impact
Yes	Yes	No	No setup or hold impact
Yes	Yes	Yes	Setup check only
No	N/A	N/A	Not recommended for FPGA designs.

Setup time is calculated for a scenario where the launch edge is delayed and the capture edge is early:

- The -early value is added from the required time.
- The -late value is added to the arrival time

```
set_clock_latency -source [-rise] [-fall] [-early] [-late] delay clock
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
source	None	Specifies the source latency on a clock pin, potentially only on certain edges of the clock.
rise	None	Specifies the edge for which this constraint will apply. If neither or both rise and fall are passed, the same latency is applied to both edges.
fall	None	Specifies the edge for which this constraint will apply. If neither or both fall and rise are passed, the same latency is applied to both edges.
late	None	Optional. Specifies that the latency is late bound on the latency. The appropriate bound provide the most pessimistic timing scenario. However, if the value of -late is less than the value of -early, optimistic timing takes place which could result in incorrect analysis. If neither or both -early and -late are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.
early	None	Optional. Specifies that the latency is early bound on the latency. The appropriate bound provide the most pessimistic timing scenario. However, if the value of -late is less than the value of -early, optimistic timing takes place which could result in incorrect analysis. If neither or both -early and -late are provided, the same latency is used for both bounds, which results in the latency having no effect for single clock domain setup and hold checks.
delay	floating point	Specifies the latency value for the constraint.
clock	string	Specifies the clock to which the constraint is applied. This clock must be constrained.

Error Codes

Error Code	Description
Error: SDC0061	Invalid clock latency constraint: Parameter has illegal value invoked from within command.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets an early clock source latency of 0.4 on the rising edge of main_clock. It also sets a clock source latency of 1.2, for both the early and late values of the falling edge of main_clock. The late value for the clock source latency for the falling edge of main_clock remains undefined.

```
set_clock_latency -source -rise -early 0.4 { main_clock }
```

```
set_clock_latency -source -fall 1.2 { main_clock }
```

Related Examples on GitHub

- set_clock_latency

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- create_clock
- create_generated_clock

8.53. set_clock_to_output (Ask a Question)

Description

Defines the timing budget available inside the FPGA for an output relative to a clock.

This non-standard sdc command is intended to constrain an output flip-flop clock-to-output time. This command is only honored by Libero SoC Timing analysis engine. It is not honored by Libero SoC Place and Route, Synplify Synthesis, or any other sdc-compliant synthesis tool. Timing analysis is performed from the initial clock source. If sdc-compliant commands are desired, see set_max_delay, set_min_delay, and set_output_delay.

The clock argument for this command must be set to the actual clock driving the output flip-flop.

```
set_clock_to_output delay_value -clock clock_ref [-max] [-min] output_list
```

Arguments

Parameter	Type	Description
delay_value	integer	Specifies the clock to output delay in nanoseconds. This time represents the amount of time available inside the FPGA between the launch clock edge and the data change at the output port.
clock	string	Specifies the clock to which the specified clock to output is related. This is a mandatory argument. **Note:** The clock parameter needs to be set to the clock driving the FF. If it is a generated clock, STA is performed from the initial clock source.
max	None	Specifies that delay_value refers to the maximum clock to output at the specified output. If you do not specify -max or -min options, the tool assumes maximum and minimum clock to output constraint values to be equal.
min	None	Specifies that delay_value refers to the minimum clock to output at the specified output. If you do not specify -max or -min options, the tool assumes maximum and minimum clock to output constraint values to be equal.
output_list	list of strings	Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Error Codes

Error Code	Description
None	Required parameter -clock is missing

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

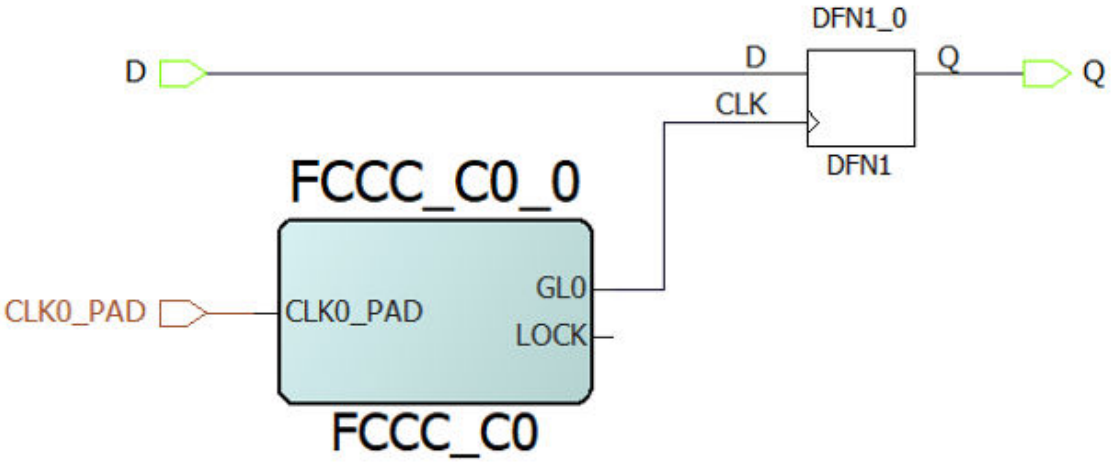
Examples

The following example sets an output delay of 0.3 ns for port Q relative to the clk clock.

```
set_clock_to_output -max 0.3 -clock { clk } [ get_ports { Q } ]
```

The following example sets an output delay of 12 ns for port Q relative to the FCCC_C0_0/FCCC_C0_0/GL0 clock. Timing analysis will be performed from the CLK0_PAD clock source, assuming that there is a “create_clock” on CLK0_PAD and a “create_generated_clock” on FCCC_C0_0/FCCC_C0_0/GL0 whose source is defined as the CCC reference clock CLK0_PAD. It is recommended to use the supplied SDC constraints generated with the configured CCC core. The supplied SDC constraints can be added to the top-level design using the Derive Constraints action in the Libero SoC Constraints Manager or using the Derive Constraints utility described in the Custom Flow User Guide

```
set_clock_to_output -max 12 -clock { FCCC_C0_0/FCCC_C0_0/GL0 } [ get_ports { Q } ]
```



Related Examples on GitHub

- set_clock_to_output

8.54. set_clock_uncertainty (Ask a Question)

Description

Specifies simple clock uncertainty for single clock and clock-to-clock uncertainty between two clocks (from and to).

The `set_clock_uncertainty` command sets the timing uncertainty of clock networks. It can be used to model clock jitter or add guard band in timing analysis.

SmartTime computes uncertainty values very similar to the `set_clock_latency` command (when used with the `-early` and `-late` arguments), but with one difference in the timing report. Uncertainty number is always applied to the required time calculation whereas latency numbers are applied to both arrival and required time. Uncertainty is added for both setup and hold checks.

To ensure that clock jitter is modeled accurately using the `set_clock_uncertainty` command, you must explicitly use `-setup` argument. If the `-setup` argument is not specified, SmartTime will apply the uncertainty value to both setup and hold calculations leading to incorrect jitter modeling.

`-setup` used	`-hold` used	Result
No	No	Setup and hold checks.

© 2025 Microchip Technology Inc. and its subsidiaries

set_clock_uncertainty (continued)		
-setup used	**-hold used**	**Result**
No	Yes	Hold checks only.
Yes	No	Setup checks only. For jitter modeling.
Yes	Yes	Setup and hold checks.

Either simple clock uncertainty or clock-to-clock uncertainty can be specified. Simple clock uncertainty can be set on a clock or on any pin in the clock network. It will then apply to any path with the capturing register in the forward cone of the uncertainty. If multiple simple uncertainty applies to a register, the last one (in the propagation order from the clock source to the register) is used. Clock-to-clock uncertainty applies to inter-clock paths. Both `from` clock and `to` clock must be specified. Clock-to-clock uncertainty has higher priority than simple uncertainty. If both are set (a clock-to-clock uncertainty and a simple clock uncertainty on the `to` clock), the simple clock uncertainty will be ignored for inter-clock paths, only the clock-to-clock uncertainty will be used.

```
set_clock_uncertainty [-setup] [-hold] uncertainty [object_list -from from_clock |
-rise_from rise_from_clock | -fall_from fall_from_clock -to to_clock | -rise_to rise_to_clock
|
-fall_to fall_to_clock ]
```

Arguments

Parameter	Type	Description
uncertainty	floating point	Specifies the time in nanoseconds that represents the amount of variation between two clock edges.
object_list	list of strings	Specifies a list of clocks, ports, or pins for simple uncertainty; the uncertainty is applied either to destination flops clocked by one of the clocks in the object list option, or destination flops whose clock pins are in the fanout of a port or a pin specified in the object_list option.
from	list of strings	Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.
rise_from	list of strings	Specifies that the clock-to-clock uncertainty applies only to rising edges of the source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.
fall_from	list of strings	Specifies that the clock-to-clock uncertainty applies only to falling edges of source clock list. Only one of the `-from`, `-rise_from`, or `-fall_from` arguments can be specified for the constraint to be valid.
from_clock/rise_from_clock/ fall_from_clock	list of strings	Specifies the list of clock names as the uncertainty source.
to	list of strings	Specifies that the clock-to-clock uncertainty applies to both rising and falling edges of the destination clock list. Only one of the `-to`, `-rise_to`, or `-fall_to` arguments can be specified for the constraint to be valid.
rise_to	list of strings	Specifies that the clock-to-clock uncertainty applies only to rising edges of the destination clock list. Only one of the `-to`, `-rise_to`, or `-fall_to` arguments can be specified for the constraint to be valid.
fall_to	list of strings	Specifies that the clock-to-clock uncertainty applies only to falling edges of the destination clock list. Only one of the `-to`, `-rise_to`, or `-fall_to` arguments can be specified for the constraint to be valid.
to_clock/rise_to_clock/fall_to_clock	list of strings	Specifies the list of clock names as the uncertainty destination.
setup	None	Specifies that the uncertainty applies only to setup checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

© 2025 Microchip Technology Inc. and its subsidiaries

set_clock_uncertainty (continued)		
Parameter	**Type**	**Description**
hold	None	Specifies that the uncertainty applies only to hold checks. If none or both `-setup` and `-hold` are present, the uncertainty applies to both setup and hold checks.

Return Type	Description
integer	Returns the ID of the clock uncertainty constraint.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

Simple Clock Uncertainty constraint examples.

The following example specifies uncertainty of 2 ns.

```
set_clock_uncertainty 2 [get_clocks clk]
```

The following example specifies setup uncertainty of 2 ns.

```
set_clock_uncertainty 2 -setup [get_clocks clk]
```

Clock to Clock Uncertainty constraint examples:

The following example specifies uncertainties of 10ns between Clk1 and Clk2 clock domains.

```
set_clock_uncertainty 10 -from [get_clocks { Clk1 }] -to [get_clocks { Clk2 }]
```

The following example specifies setup uncertainties between Clk1 and {Clk2 Clk3} clock domains with specific edges.

```
set_clock_uncertainty 0 -from [get_clocks { Clk1 }] -fall_to [get_clocks { Clk2 Clk3 }] -setup
```

```
set_clock_uncertainty 4.3 -fall_from [get_clocks { Clk1 Clk2 }] -rise_to *
```

```
set_clock_uncertainty 0.1 -rise_from [ get_clocks { Clk1 Clk2 }] \
-fall_to [get_clocks { Clk3 Clk4 }] -setup
```

```
set_clock_uncertainty 5 -rise_from [get_clocks {Clk1}]  -to [ get_clocks {*} ]
```

Related Examples on GitHub

- set_clock_uncertainty

See Also

- list_clock_uncertainties
- remove_clock_uncertainty

© 2025 Microchip Technology Inc. and its subsidiaries

8.55. set_current_scenario (Ask a Question)

Description

Specifies the timing scenario for the Timing Analyzer to use. All commands that follow this command will apply to the specified timing scenario. A timing scenario is a set of timing constraints used with a design. If the specified scenario is already the current one, this command has no effect.

After setting the current scenario, constraints can be listed, added, or removed, the checker can be invoked on the set of constraints, and so on.

This command uses the specified timing scenario to compute timing analysis.

Note: It is recommended to use the `organize_tool_files` command instead of this command.

```
set_current_scenario name
```

Arguments

Parameter	Type	Description
`name`	string	Specifies the name of the timing scenario to which to apply all commands from this point on.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following command sets `scenario_A` as current scenario of the timing scenario.

```
set_current_scenario scenario_A
```

Related Examples on GitHub

- set_current_scenario

See Also

- create_scenario
- get_current_scenario
- remove_scenario
- rename_scenario

8.56. set_disable_timing (Ask a Question)

Description

Disables timing arcs within a cell (instance) and returns the ID of the created constraint if the command succeeded. To specify a Disable Timing constraint, open the Set Constraint to Disable Timing Arcs dialog box in the following way: From the **Constraints** menu, click **Disable Timing**.
Note: This constraint is for the Place and Route tool and the Verify Timing tool. It is ignored by the Synthesis tool.

```
set_disable_timing -from value -to value name
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
from	string	Specifies the starting point for the timing arc. The -from and -to arguments must either both be present or both omitted for the constraint to be valid.
to	string	Specifies the ending point for the timing arc. The -from and -to arguments must either both be present or both omitted for the constraint to be valid.
name	string	Specifies the instance(cell) name for which the disable timing arc constraint will be created.

Return Type	Description
integer	Returns the ID of created constraint.

Error Codes

Error Code	Description
None	Required parameter _AtclParam0_ is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example disables timing arcs within A to Y.

```
set_disable_timing -from A -to Y
```

Related Examples on GitHub

- set_disable_timing

See Also

- remove_disable_timing

8.57. set_external_check (Ask a Question)

Description

Defines the external setup and hold delays for an input relative to a clock.

```
set_external_check delay_value -clock clock_ref [-setup] [-hold] input_list
```

Arguments

Parameter	Type	Description
delay_value	integer	Specifies the delay at the input port/pin inside the FPGA.
clock	string	Specifies the reference clock to which the specified external check is related. This is a mandatory argument.
setup or hold	None	Specifies that delay_value refers to the setup/hold check at the specified input. This is a mandatory argument if -hold is not used. You must specify either the -setup or -hold option.

© 2025 Microchip Technology Inc. and its subsidiaries

set_external_check (continued)		
Parameter	**Type**	**Description**
input_list	list of strings	Provides a list of input ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Return Type	Description
integer	Returns the ID of the external setup and hold constraint.

Error Codes

Error Code	Description
None	Required parameter -clock is missing.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets an external setup delay with 0.2 ns for A input port.

```
set_external_check -setup 0.2 -clock { clk } [ get_ports { A } ]
```

Related Examples on GitHub

- set_external_check

See Also

- set_input_delay

8.58. set_external_delay (Ask a Question)

Description

Specifies the external delay between -from and -to ports (outside of chip). The delay is considered during Timing Analysis for PLL external feedback delay calculation when the PLL output goes outside of the chip through the -from pin, and re-enters the chip through the -to pin, which then connects to the PLL feedback clock input pin.

Important: This constraint is not supported by the Synplify Pro Synthesis software. In Libero flow, this constraint is skipped for Synplify Pro Synthesis software.

```
set_external_delay -from value -to value [-min] [-max] <delay_value>
```

Arguments

Parameter	Type	Description
from	string	Specifies the output port that is connected to PLL output. This argument is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

set_external_delay (continued)		
Parameter	**Type**	**Description**
to	string	Specifies the input port that is connected to PLL feedback. This argument is mandatory.
min	flag	Specifies the external feedback delay for minimum analysis.
max	flag	Specifies the external feedback delay for maximum analysis.
delay_value	real	Specifies the external delay value between -from to -to ports in nanoseconds. This argument is mandatory. If neither the -min nor -max parameter value is specified, the same delay_value is used for both minimum and maximum analysis.

Return Type	Description
void	No return type

Error Codes

Error Code	Description
SDC0015	Invalid external delay constraint: port list <specified_port> is incorrect.
SDC0078	Invalid external delay constraint: shared -from or -to ports
SDC0061	Error in command set_external_delay: Parameter -from has illegal value
SDC0061	Error in command set_external_delay: Parameter -to has illegal value

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets the PLL external feedback off-chip delay between ports GL0_CLK_OUT and FB_CLK_IN to 3 ns.

```
set_external_delay -from [ get_ports { GL0_CLK_OUT} ] -to [ get_ports { FB_CLK_IN } ] 3.0
```

8.59. set_false_path (Ask a Question)

Description

Identifies paths that are considered false and excluded from the timing analysis in the current timing scenario. The set_false_path command identifies specific timing paths as being false. The false timing paths are paths that do not propagate logic level changes. This constraint removes timing requirements on these false paths so that they are not considered during the timing analysis. The path starting points are the clocks, the input ports or register clock pins, and the path ending points are the clocks, the register data pins or output ports. This constraint disables setup and hold checking for the specified paths.

The false path information always takes precedence over multiple cycle path information and overrides maximum delay constraints. If more than one object is specified within one -through option, the path can pass through any objects.

© 2025 Microchip Technology Inc. and its subsidiaries

You must specify at least one of the -from, -to, or -through arguments for this constraint to be valid.

```
set_false_path [-ignore_errors] [-from from_list ] [-through through_list ] [-to to_list ]
```

Arguments

Parameter	Type	Description
ignore_errors	None	Specifies to avoid reporting errors for derived constraints targeting the logic that becomes invalid due to logic optimization. It is an optional argument. Some IPs may have extra logic present depending on other IPs used in the design but the synthesis tool will remove this logic if fewer IPs were used. In such cases, the implementation flow will halt without -ignore_errors flag. **Note:** It is not recommended to use this flag outside similar use cases.
from	list of strings	Specifies a list of timing paths starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
through	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the disabled paths must pass.
to	list of strings	Specifies a list of timing paths ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.

Return Type	Description
None	None

Error Codes

Error Code	Description
Error: SDC0021	Invalid false path constraint: the -from value is incorrect.
Error: SDC0022	Invalid false path constraint: the -from is empty.
Error: SDC0024	Invalid false path constraint: the -to is empty.
Error: SDC0026	Invalid false path constraint: the -through is empty.
Warning:	cell (get_cells) is incorrect type; -through objects must be of type net (get_nets), or pin (get_pins). **Note:** Constraint will be disabled.
Warning:	port (get_ports) is incorrect type; -through objects must be of type net (get_nets), or pin (get_pins). **Note:** Constraint will be disabled.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example specifies all paths from clk1 clock to data pins of a specific register reg_2 as false paths.

```
set_false_path -from [get_clocks {clk1}] -to reg_2:D
```


© 2025 Microchip Technology Inc. and its subsidiaries

The following example specifies all paths through the pin U0/U1:Y to be false.

```
set_false_path -through U0/U1:Y
```

The following example specifies a derived false path constraint through the `PCIe_Demo_0/SYSRESET_POR/POWER_ON_RESET_N` pin.

```
set_false_path -ignore_errors -through [ get_pins {PCIe_Demo_0/SYSRESET_POR/
POWER_ON_RESET_N } ]
```

Related Examples on GitHub

- set_false_path

See Also

- remove_false_path

8.60. set_input_delay (Ask a Question)

Description

Creates an input delay on a port list by defining the arrival time of an input relative to a clock in the current scenario.

The `set_input_delay` command sets input path delays on input ports relative to a clock edge. This usually represents a combinational path delay from the clock pin of a register external to the current design. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds input delay to path delay for paths starting at primary inputs.

A clock is a singleton that represents the name of a defined clock constraint. This can be:

- a single port name used as source for a clock constraint.
- a single pin name used as source for a clock constraint; for instance `reg1:CLK`. This name can be hierarchical (for instance, `toplevel/block1/reg2:CLK`).
- an object accessor that will refer to one clock: `[get_clocks {clk}]`.

Notes:

- The behavior of the -add_delay option is identical to that of PrimeTime(TM).
- If, using the `-add_delay` mechanism, multiple constraints are otherwise identical, except they specify different `-max` or `-min` values
  - The surviving `-max` constraint will be the maximum of the `-max` values.
  - The surviving `-min` constraint will be the minimum of the `-min` values.

```
set_input_delay delay_value -clock clock_ref [-max] [-min] [-clock_fall] [-rise] [-fall] [-
add_delay] \
input_list
```

Arguments

Parameter	Type	Description
`delay_value`	real	Specifies the arrival time in nanoseconds that represents the amount of time for which the signal is available at the specified input after a clock edge.
`clock`	string	Specifies the clock reference to which the specified input delay is related. This is a mandatory argument.
`max`	None	Specifies that the `delay_value` refers to the longest path arriving at the specified input. If you do not specify `-max` or `-min` options, the tool assumes maximum and minimum input delays to be equal.

© 2025 Microchip Technology Inc. and its subsidiaries

set_input_delay (continued)

Parameter	Type	Description
min	None	Specifies that the delay_value refers to the shortest path arriving at the specified input. If you do not specify -max or -min options, the tool assumes maximum and minimum input delays to be equal.
clock_fall	None	Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.
rise	None	Specifies that the delay is relative to a rising transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
fall	None	Specifies that the delay is relative to a falling transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
add_delay	None	Specifies that this input delay constraint should be added to an existing constraint on the same port(s). The -add_delay option capture information on multiple paths with different clocks or clock edges leading to the same input port(s).
input_list	list of string	Provides a list of input ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Return Type	Description
integer	Returns the ID of the clock input delay constraint.

Error Codes

Error Code	Description
Error: SDC0004	clk does not match any clock name or source.
Error: SDC0015	port list [get_ports {CLK_0_D}] is incorrect.
Error: SDC0054	Invalid IO delay constraint: the min delay is greater than max delay.
Error: SDC0061	Parameter _AtclParam0_ has illegal value.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets an input delay of 1.2 ns for port data1 relative to the rising edge of CLK1.

```
set_input_delay 1.2 -clock [get_clocks CLK1] [get_ports data1]
```

The following example sets a different maximum and minimum input delay for port IN1 relative to the falling edge of CLK2.

```
set_input_delay 1.0 -clock_fall -clock CLK2 -min {IN1}
```

```
set_input_delay 1.4 -clock_fall -clock CLK2 -max {IN1}
```


© 2025 Microchip Technology Inc. and its subsidiaries

The following example demonstrates an override condition of two constraints. The first constraint is overridden because the second constraint specifies a different clock for the same input.

```
set_input_delay 1.0 -clock CLK1 -max {IN1}
```

```
set_input_delay 1.4 -clock CLK2 -max {IN1}
```

The next example is almost the same as the previous one, however, in this case, the user has specified `-add_delay`, so both constraints will be honored.

```
set_input_delay 1.0 -clock CLK1 -max {IN1}
```

```
set_input_delay 1.4 -add_delay -clock CLK2 -max {IN1}
```

The following example is more complex:

- All constraints are for an input to port PAD1 relative to a rising edge clock `CLK2`. Each combination of `{-rise, -fall} x {-max, -min}` generates an independent constraint. But the max rise delay of 5 and the max rise delay of 7 interfere with each other.
- For a -max option, the maximum value overrides all lower values. Thus the first constraint will be overridden and the max rise delay of 7 will survive.

```
set_input_delay 5 -max -rise -add_delay [get_clocks CLK2] [get_ports PAD1] # will be
overridden
```

```
set_input_delay 3 -min -fall -add_delay [get_clocks CLK2] [get_ports PAD1]
```

```
set_input_delay 3 -max -fall -add_delay [get_clocks CLK2] [get_ports PAD1]
```

```
set_input_delay 7 -max -rise -add_delay [get_clocks CLK2] [get_ports PAD1]
```

Related Examples on GitHub

- set_input_delay

See Also

- set_output_delay
- remove_input_delay
- remove_output_delay

8.61. set_input_jitter (Ask a Question)

Description

Sets the input jitter for a given clock.

```
set_input_jitter <input_jitter> <clock>
```

Important:

- This constraint is not supported as a Tcl command. Enter it as part of a timing constraint (`.sdc`) file.
- The SynplifyPro synthesis software ignores the `set_input_jitter` SDC constraint.

© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
<input_jitter>	floating point	Specifies the input jitter value in nanoseconds. This value must be greater than zero. The input_jitter value allows the user to specify the clock jitter on the external clock. For RTG4™ FPGAs, the user must include the larger of the external clock jitter or the RTG4 Input Buffer jitter data sheet specifications using this constraint. Furthermore for RTG4 FPGAs, if the input port is a Strobe input to the RTG4 CCC SpaceWire clock and data recovery circuit, use this constraint to account for the CCC SpaceWire effective recovered clock jitter data sheet specification during STA.
<input clock>	string	Specifies the clock on which to apply the input clock jitter value. The user must ensure there is also a create_clock SDC constraint applied to the same clock input port to define that input as a clock for Static Timing Analysis (STA).

Example

The following example sets the input jitter to 2.1 nanosecond.

Important: The clock used to set the jitter is the clock name that is defined in the `create_clock` command. CLK is the clock port name.

```
create_clock -name { clockName }-period 20 -waveform {0 10 } [ get_ports { CLK } ]
set_input_jitter 2.1 [ get_clocks { clockName } ]
```

8.62. set_max_delay (Ask a Question)

Description

Specifies the required maximum delay for timing paths in the current design. The path length for any startpoint in `from_list` to any endpoint in `to_list` must be less than the `delay_value`. The timing engine automatically derives the individual maximum delay targets from clock waveforms and port input or output delays. For more information, refer to the `create_clock`, `set_input_delay`, and `set_output_delay` commands. The maximum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multi-cycle path constraint.

You must specify at least one of the `-from` , `-to`, or `-through` arguments for this constraint to be valid.

```
set_max_delay delay_value [-ignore_clock_latency] [-from from_list ] [-to to_list ] [-through
through_list ]
```

Arguments

Parameter	Type	Description
delay_value	floating point	Specifies a floating point number in nanoseconds that represents the required maximum delay value for specified paths. • If the path starting point is on a sequential device, the tool includes clock skew in the computed delay. • If the path starting point has an input delay specified, the tool adds that delay value to the path delay. • If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay. • If the ending point has an output delay specified, the tool adds that delay to the path delay.

© 2025 Microchip Technology Inc. and its subsidiaries

set_max_delay (continued)

Parameter	Type	Description
`from`	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
`to`	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
`through`	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the timing paths must pass.
`ignore_clock_latency`	flag	Specifies that the calculation of slack for the specified path excludes clock skew and jitter, and only the data path is considered. **Notes:** • The flag is useful for analyzing the paths between the sequential elements driven by asynchronous clocks. • This argument is not supported by the Synplify Pro Synthesis software. In Libero Design flow, this option is skipped for the Synplify Pro Synthesis software.

Return Type	Description
`integer`	Returns the ID of the clock maximum delay constraint.

Error Codes

Error Code	Description
Error: SDC0021	Invalid max delay constraint: the `-from` value is incorrect.
Error: SDC0022	Invalid max delay constraint: the `-from` is empty.
Error: SDC0023	Invalid max delay constraint: the `-to` value is incorrect.
Error: SDC0024	Invalid max delay constraint: the `-to` is empty.
Error: SDC0026	Invalid max delay constraint: the `-through` is empty
Error: SDC0061	Invalid max delay constraint: Missing or Illegal parameter/value.
Warning	cell (get_cells) is incorrect type;"`-through`" objects must be of type net (`get_nets`), or pin (`get_pins`). **Note:** Constraint will be disabled.
Warning	port (get_ports) is incorrect type;"`-through`" objects must be of type net (`get_nets`), or pin (`get_pins`). **Note:** Constraint will be disabled.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets a maximum delay by constraining all paths from `ff1a:CLK or ff1b:CLK` to `ff2e:D` with a delay less than 5 ns.

```
set_max_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```


© 2025 Microchip Technology Inc. and its subsidiaries

The following example sets a maximum delay by constraining all paths to output ports whose names start by "out" with a delay less than 3.8 ns.

```
set_max_delay 3.8 -to [get_ports out*]
```

Related Examples on GitHub

- set_max_delay

See Also

- set_max_delay
- remove_max_delay

8.63. set_min_delay (Ask a Question)

Description

Specifies the required minimum delay for timing paths in the current design. The path length should be such that the delay along the specified path should be more than the value mentioned for the `delay_value` switch. The timing engine automatically derives the individual minimum delay targets from clock waveforms and port input or output delays. For more information, refer to the `create_clock`, `set_input_delay`, and `set_output_delay`commands. The minimum delay constraint is a timing exception. This constraint overrides the default single cycle timing relationship for one or more timing paths. This constraint also overrides a multi-cycle path constraint.

You must specify at least one of the `-from`, `-to`, or `-through` arguments for this constraint to be valid.

```
set_min_delay delay_value [-ignore_clock_latency][-from from_list ] [-to to_list ] [-through through_list ]
```

Arguments

Parameter	Type	Description
`delay_value`	floating point	Specifies a floating point number in nanoseconds that represents the required minimum delay value for specified paths. • If the path starting point is on a sequential device, the tool includes clock skew in the computed delay. • If the path starting point has an input delay specified, the tool adds that delay value to the path delay. • If the path ending point is on a sequential device, the tool includes clock skew and library setup time in the computed delay. • If the ending point has an output delay specified, the tool adds that delay to the path delay.
`from`	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
`to`	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
`through`	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the timing paths must pass.

© 2025 Microchip Technology Inc. and its subsidiaries

set_min_delay (continued)		
Parameter	**Type**	**Description**
ignore_clock_latency	flag	Specifies that the calculation of slack for the specified path excludes clock skew and jitter, and only the data path is considered. **Notes:** • The flag is useful for analyzing the paths between the sequential elements driven by asynchronous clocks. • This option is not supported by the Synplify Pro Synthesis software. In the Libero design flow, this option is skipped for the Synplify Pro Synthesis software.

Return Type	Description
integer	Returns the ID of the clock minimum delay constraint.

Error Codes

Error Code	Description
Error: SDC0021	Invalid min delay constraint: the -from value is incorrect.
Error: SDC0022	Invalid min delay constraint: the -from is empty.
Error: SDC0023	Invalid min delay constraint: the -to value is incorrect.
Error: SDC0024	Invalid min delay constraint: the -to is empty.
Error: SDC0026	Invalid min delay constraint: the -through is empty.
Error: SDC0061	Invalid min delay constraint: Missing or Illegal parameter/value.
Warning	port (get_ports) is incorrect type;"-through" objects must be of type net (get_nets), or pin (get_pins).

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets a minimum delay by constraining all paths from ff1a:CLK or ff1b:CLK to ff2e:D with a delay more than 5 ns.

```
set_min_delay 5 -from {ff1a:CLK ff1b:CLK} -to {ff2e:D}
```

The following example sets a minimum delay by constraining all paths to output ports whose names start by "out" with a delay more than 3.8 ns.

```
set_min_delay 3.8 -to [get_ports out*]
```

Related Examples on GitHub

- set_min_delay

See Also

- set_min_delay
- remove_max_delay

© 2025 Microchip Technology Inc. and its subsidiaries

8.64. set_multicycle_path (Ask a Question)

Description

Defines a path that takes multiple clock cycles in the current scenario. Setting multiple cycle paths constraint overrides the single cycle timing relationships between sequential elements by specifying the number of cycles that the data path must have for setup or hold checks. If you change the multiplier, it affects both the setup and hold checks.

False path information always takes precedence over multiple cycle path information. A specific maximum delay constraint overrides a general multiple cycle path constraint. If you specify more than one object within one `-through` option, the path passes through any of the objects.

You must specify at least one of the `-from` , `-to`, or `-through` arguments for this constraint to be valid.

```
set_multicycle_path ncycles [-setup] [-hold] [-setup_only] [-from from_list] \
[-through through_list ] [-to to_list] [ -start ] [ -end ]
```

Arguments

Parameter	Type	Description
`ncycles`	integer	Specifies an integer value that represents a number of cycles the data path must have for setup or hold check. The value is relative to the starting point or ending point clock, before data is required at the ending point. Number of cycles must be greater than `1`. If you set ncycles as `2.2` or `4/2` or `"8a"` then it is being truncated as `2` or `4` or `8`, and no warning is reported.
`setup`	None	Optional. Applies the cycle value for the setup check only. The default hold check will be applied unless you have specified another set_multicycle_path command for the hold value.
`hold`	None	Optional. Applies the cycle value for the hold check only. This option does not affect the setup check. **Note:** If you do not specify `-setup` or `-hold`, the cycle value is applied to the setup check and the default hold check is 0 not ncycles -1 .
`setup_only`	None	Optional. Specifies that the path multiplier is applied to setup paths only.
`from`	list of strings	Specifies a list of timing path starting points. A valid timing starting point is a clock, a primary input, an inout port, or a clock pin of a sequential cell.
`through`	list of strings	Specifies a list of pins, ports, nets, or instances (cells) through which the multiple cycle paths must pass.
`to`	list of strings	Specifies a list of timing path ending points. A valid timing ending point is a clock, a primary output, an inout port, or a data pin of a sequential cell.
`start`	None	Changes the clock edges used to launch and capture the data. By default, setup multicycle holds the multicycle shifts launching edge backwards (that is-start is the default for hold). -start allows you to change those defaults and hold multicycle by shifting the launching clock forward.
`end`	None	Changes the clock edges used to launch and capture the data. By default, setup multicycle shifts the capturing edge forward (that is-end is the default for setup). -end allows you to change the defaults and specify a setup multicycle by shifting the capturing clock backward.

Error Codes

Error Code	Description
Error: SDC0004	clk does not match any clock name or source.

© 2025 Microchip Technology Inc. and its subsidiaries

set_multicycle_path (continued)	
Error Code	**Description**
Error: SDC0015	port list [get_ports { CLK_0_D }] is incorrect.
Error: SDC0054	Invalid IO delay constraint: the min delay is greater than max delay.
Error: SDC0061	Parameter _AtclParam0_ has illegal value.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Exceptions

Multiple priority management is not supported in Microchip SoC designs. All multiple cycle path constraints are handled with the same priority.

Example

The following example sets all paths between reg1 and reg2 to 3 cycles for setup check. Hold check is measured at the previous edge of the clock at reg2.

```
set_multicycle_path 3 -from [get_pins {reg1}] -to [get_pins {reg2}]
```

The following example specifies that four cycles are needed for setup check on all paths starting at the registers in the clock domain ck1. Hold check is further specified with two cycles instead of the three cycles that would have been applied otherwise.

```
set_multicycle_path 4 -setup -from [get_clocks {ck1}]
```

```
set_multicycle_path 2 -hold -from [get_clocks {ck1}]
```

The following example specifies that four cycles are needed for setup only check on all paths starting at the registers in the clock domain `REF_CLK_0`.

```
set_multicycle_path -setup_only 4 -from [ get_clocks { REF_CLK_0 } ]
```

The following are few more examples of the `set_multicycle_path -setup_only` command.

```
set_multicycle_path -setup_only 4 -from [ all_registers ]
set_multicycle_path -setup_only 4 -through [ get_cells { comb_0/XOR2_0 } ]
set_multicycle_path -setup_only 4 -from [ get_clocks { PF_CCC_C0_0/PF_CCC_C0_0/pll_inst_0/
OUT0 } ] -to [ get_clocks { PF_CCC_C0_0/PF_CCC_C0_0/pll_inst_0/OUT1 } ]
```

The following example shifts lauching edge 1 clk1 cycle backward

```
set_multicycle_path -setup -start 2 -from [ get_clocks clk1 ] -to [ get_clocks clk2 ]
```

The following example shifts capturing edge 1 clk2 cycle forward

```
set_multicycle_path -setup -end 2 -from [ get_clocks clk1 ] -to [ get_clocks clk2 ]
```

Related Examples on GitHub

- set_multicycle_path

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- remove_multicycle_path

8.65. set_options (Ask a Question)

Description

Sets options for timing analysis which can be changed in the **SmartTime Options** dialog box in the SmartTime GUI. All of the options from SmartTime are passed on to place-and-route tool, and some affect timing-driven place-and-route.

```
set_options \
[-max_opcond value ] \
[-min_opcond value ] \
[-interclockdomain_analysis value ] \
[-use_bibuf_loopbacks value ] \
[-enable_recovery_removal_checks value ] \
[-break_at_async value ] \
[-filter_when_slack_below value ] \
[-filter_when_slack_above value ] \
[-remove_slack_filters] \
[-limit_max_paths value ] \
[-expand_clock_network value ] \
[-expand_parallel_paths value ] \
[-analysis_scenario value ] \
[-tdpr_scenario value ] \
[-reset]
```

Arguments

Parameter	Type	Description
max_opcond	string	Sets the operating condition to use for Maximum Delay Analysis. The acceptable values for max_opcond for PolarFire can be the following: • slow_lv_ht - use slow_lv_ht conditions for maximum delay analysis • slow_lv_lt - use slow_lv_lt conditions for maximum delay analysis • fast_hv_lt - use fast_hv_lt conditions for maximum delay analysis Default is slow_lv_lt. max_opcond for SmartFusion® 2, IGLOO® 2, and RTG4™ can be as following: • worst - use worst case conditions for maximum delay analysis • typical - use typical conditions for maximum delay analysis • best - use best case conditions for maximum delay analysis Default is worst.

© 2025 Microchip Technology Inc. and its subsidiaries

set_options (continued)

Parameter	Type	Description
min_opcond	string	Sets the operating condition to use for Minimum Delay Analysis. The acceptable values for min_opcond for PolarFire can be the following: • slow_lv_ht - use slow_lv_ht conditions for minimum delay analysis • slow_lv_lt - use slow_lv_lt conditions for minimum delay analysis • fast_hv_lt - use fast_hv_lt conditions for minimum delay analysis Default is fast_hv_lt. min_opcond for SmartFusion® 2, IGLOO® 2, and RTG4™ can be as following: • worst - use worst case conditions for minimum delay analysis • typical - use typical conditions for minimum delay analysis • best - use best case conditions for minimum delay analysis Default is best.
interclockdomain_analysis	string	Enables or disables inter-clock domain analysis. Value can be the following: • yes - enables inter-clock domain analysis • no - disables inter-clock domain analysis Default is yes. Timing-driven place-and-route is affected by this option.
use_bibuf_loopbacks	string	Instructs the timing analysis whether to consider loopback path in bidirectional buffers (D->Y, E->Y) as false-path {no}. Default is no; that is loopbacks are false paths. Values can be the following: • yes - enables loopback in bibufs • no - disables loopback in bibufs
enable_recovery_removal_checks	string	Enables recovery checks to be included in max-delay analysis and removal checks in min-delay analysis. Default is no. Values can be the following: • yes - enables recovery an removal checks • no - disables recovery and removal checks
break_at_async	string	Specifies whether or not timing analysis is allowed to cross asynchronous pins (clear, reset of sequential elements). Default is yes. Values can be the following: • yes - enables breaking paths at asynchronous ports • no - disables breaking paths at asynchronous ports. Timing-driven place-and-route is affected by this option.
filter_when_slack_below	floating point	Specifies a minimum slack value for paths reported by list_paths. Not set by default.
filter_when_slack_above	floating point	Specifies a maximum slack value for paths reported by list_paths. Not set by default.
remove_slack_filters	None	Removes the slack minimum and maximum set using -filter_when_slack_below and -filter_when_slack_above.
limit_max_paths	integer	Specifies the maximum number of paths reported by list_paths. Default is 20. Number must be greater than 0.

© 2025 Microchip Technology Inc. and its subsidiaries

set_options (continued)		
Parameter	**Type**	**Description**
expand_clock_network	string	Specify whether or not clock network details are reported in expand_path. Default is yes. Values can be the following: • yes - enables expanded clock network information in paths • no - disables expanded clock network information in paths.
expand_parallel_paths	integer	Specify the number of parallel paths {paths with the same ends} to include in expand_path. Default is 1. Number must be greater than 0.
analysis_scenario	string	Specify the constraint scenario to be used for timing analysis. Default scenario is Primary.
tdpr_scenario	string	Specify the constraint scenario to be used for timing-driven place-and-route. Default scenario is Primary. Timing-driven place-and-route is affected by this option.
reset	None	Reset all options to the default values, except those for analysis and TDPR scenarios, which remain unchanged.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following script commands the timing engine to use best operating conditions for both max-delay analysis and min-delay analysis:

```
set_options -max_opcond {best} -min_opcond {best}
```

```
set_options -max_opcond {fast_hv_lt} -min_opcond {fast_hv_lt}
```

The following script changes the scenario used by timing-driven place-and-route and saves the change in the Libero project for place-and-route tools to see the change.

```
set_options -tdpr_scenario {My_TDPR_Scenario}
```

Related Examples on GitHub

- set_options

8.66. set_output_delay (Ask a Question)

Description

Defines the output delay of an output relative to a clock in the current scenario.

The `set_output_delay` command sets output path delays on output ports relative to a clock edge. Output ports have no output delay unless you specify it. For in/out (bidirectional) ports, you can specify the path delays for both input and output modes. The tool adds output delay to path delay for paths ending at primary outputs.

© 2025 Microchip Technology Inc. and its subsidiaries

Notes:

- The behavior of the -add_delay option is identical to that of PrimeTime(TM).
- If, using the -add_delay mechanism, multiple constraints are otherwise identical, except they specify different -max or -min values.
  - the surviving -max constraint will be the maximum of the -max values.
  - the surviving -min constraint will be the minimum of the -min values.

```
set_output_delay [-max] [-min] delay_value -clock clock_ref [-clock_fall] [-rise] [-fall] \
[-add_delay] output_list
```

Arguments

Parameter	Type	Description
delay_value	float	Specifies the amount of time before a clock edge for which the signal is required. This represents a combinational path delay to a register outside the current design plus the library setup time (for maximum output delay) or hold time (for minimum output delay).
clock	string	Specifies the clock reference to which the specified output delay is related. This is a mandatory argument.
max	None	Specifies that delay_value refers to the longest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.
min	None	Specifies that delay_value refers to the shortest path from the specified output. If you do not specify -max or -min options, the tool assumes the maximum and minimum output delays to be equal.
clock_fall	None	Specifies that the delay is relative to the falling edge of the clock reference. The default is the rising edge.
rise	None	Specifies that the delay is relative to a rising transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
fall	None	Specifies that the delay is relative to a falling transition on the specified port(s). If -rise or -fall is not specified, then rising and falling delays are assumed to be equal.
add_delay	None	Specifies that this output delay constraint should be added to an existing constraint on the same port(s). The -add_delay option capture information on multiple paths with different clocks or clock edges leading to the same output port(s). **Notes:** • The behavior of the -add_delay option is identical to that of PrimeTime(TM). • If, using the -add_delay mechanism, multiple commands are otherwise identical, except they specify different -max or -min values. • the surviving -max constraint will be the maximum of the -max values. • the surviving -min constraint will be the minimum of the -min values.
output_list	list of string	Provides a list of output ports in the current design to which delay_value is assigned. If you need to specify more than one object, enclose the objects in braces ({}).

Return Type	Description
integer	Returns the ID of the clock output delay constraint.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
Error: SDC0004	Invalid output delay constraint: clk does not match any clock name or source.
Error: SDC0015	Invalid output delay constraint: port list is incorrect.
Error: SDC0054	Invalid I/O delay constraint: the min delay is greater than max delay.
Error: SDC0061	Invalid output delay constraint: Missing or Illegal parameter/value.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following example sets an output delay of 1.2 ns for port OUT1 relative to the rising edge of CLK1.

```
set_output_delay 1.2 -clock [get_clocks CLK1] [get_ports OUT1]
```

The following example sets a different maximum and minimum output delay for port OUT1 relative to the falling edge of CLK2.

```
set_output_delay -min {OUT1} 1.0 -clock_fall -clock CLK2
set_output_delay -max {OUT1} 1.4 -clock_fall -clock CLK2
```

The following example demonstrates an override condition of two constraints. The first constraint is overridden because the second constraint specifies a different clock for the same output.

```
set_output_delay 1.0 {OUT1} -clock CLK1 -max
set_output_delay 1.4 {OUT1} -clock CLK2 -max
```

The next example is almost the same as the previous one, however, in this case, the user has specified `-add_delay`, so both constraints will be honored.

```
set_output_delay 1.0 {OUT1} -clock CLK1 -max
set_output_delay 1.4 {OUT1} -add_delay -clock CLK2 -max
```

The following example is more complex:

- All constraints are for an output to port PAD1 relative to a rising edge clock CLK2. Each combination of {-rise, -fall} x {-max, -min} generates an independent constraint. But the max rise delay of 5 and the max rise delay of 7 interfere with each other.
- For a `-max` option, the maximum value overrides all lower values. Thus the first constraint will be overridden and the max rise delay of 7 will survive.

```
set_output_delay 5 [get_clocks CLK2] [get_ports PAD1] -max -rise -add_delay # will be
overridden
set_output_delay 3 [get_clocks CLK2] [get_ports PAD1] -min -fall -add_delay
set_output_delay 3 [get_clocks CLK2] [get_ports PAD1] -max -fall -add_delay
set_output_delay 7 [get_clocks CLK2] [get_ports PAD1] -max -rise -add_delay
```

Related Examples on GitHub

- set_output_delay

© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- set_input_delay
- remove_input_delay

8.67. set_system_jitter (Ask a Question)

Description

Sets the system jitter and overrides the automatically computed value.

```
set_system_jitter <system_jitter>
```

Important: This constraint is not supported as a Tcl command. Enter it as part of a timing constraint (`.sdc`) file.

Arguments

Parameter	Type	Description
system_jitter	floating point	Specifies the system jitter value in nanoseconds. This value must be greater than zero.

8.68. write_sdc (Ask a Question)

Description

Writes timing constraints into an SDC file. If multiple constraint scenarios are defined, `-scenario` allows the user to specify which scenario to write. By default, the current scenario is written.

```
write_sdc \
-scenario scenario_name \
-pin_separator ( : | / ) \
file name
```

Arguments

Parameter	Type	Description
scenario	string	Specifies the scenario to write. By default the current scenario is used.
pin_separator	char	Specify the pin separator used in the SDC file. It can be either ':' or '/'.
file name	string	Specify the SDC file name.

Supported Families

PolarFire®
PolarFire SoC
SmartFusion® 2
IGLOO® 2
RTG4™

Example

The following script merges two SDC files and writes the result into a third SDC file.

```
read_sdc first.sdc
read_sdc -add second.sdc
write_sdc margin.sdc
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- read_sdc

© 2025 Microchip Technology Inc. and its subsidiaries

9. SmartPower Tcl Commands (Ask a Question)

9.1. smartpower_add_new_custom_mode (Ask a Question)

Description

This Tcl command creates a new custom mode.

```
smartpower_add_new_custom_mode -name {mode name} \
                               -base_mode {base mode} \
                               [-description {mode description}]
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the new custom mode. This parameter is mandatory.
base_mode	string	Specifies the name of the base mode used to create the new custom mode. It must be one of the following: Active, Standby or Flash*Freeze (RTG4, SmartFusion 2, and IGLOO 2). This parameter is mandatory.
description	string	Specifies the description of the new custom mode. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	base_mode: Invalid argument value: 'mode_name' (expecting Active, Static, new_mode_name1, new_mode_name2 or new_mode_name4).
None	Parameter 'base_mode' has illegal value.
None	Required parameter 'base_mode' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_add_new_custom_mode -name "name" [-description "description"] -base_mode "Active | Static | new_mode_name1 | new_mode_name2 | new_mode_name4"'.

Supported Families

Supported Families	Supported Versions
PolarFire®	v12.4+
SmartFusion® 2	v12.4+
RTG4™	v12.4+
IGLOO® 2	v12.4+
PolarFire SoC	v12.6+

Example

This example creates a new custom mode "NewCustomMode" based on the Active mode:

```
smartpower_add_new_custom_mode -name {NewCustomMode} \
                               -base_mode {Active} \
                               -description {frequency 10 MHz}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- smartpower_remove_custom_mode

9.2. smartpower_add_new_scenario (Ask a Question)

Description

This Tcl command creates a new custom scenario entering the scenario name, duration (total duration for the sequence must equal to 100%), and selecting previously defined operating modes for this sequence.

```
smartpower_add_new_scenario -name {New Scenario name} \
                            [-description {description of scenario}] \
                            -mode {Mode_name:duration}
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the new scenario.
description	string	Specifies the description of the new scenario. This parameter is optional.
mode	string	Specifies the previously defined mode(s) and duration(s) for the specified scenario. This parameter is mandatory. There may be multiple -mode arguments (see example below).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'mode' is missing.
None	mode: Invalid argument value: 'mode_name' (expecting Active, Static or Flash*Freeze).
None	The sum of the duration must be 100%. Current sum: duration_value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_add_new_scenario -name "scenario name" [-description "description"] [-mode "":""]+'

Supported Families

Supported Families	Supported Versions
PolarFire®	v12.4+
SmartFusion® 2	v12.4+
RTG4™	v12.4+
IGLOO® 2	v12.4+
PolarFire SoC	v12.6+

Example

This example creates a new scenario called "MyScenario" with the specified modes and durations:

```
smartpower_add_new_scenario -name "MyScenario" -mode "Custom_1:50.00" \
                          -mode "Custom_2:25.00" -mode "Active:25.00"
```

See Also

- smartpower_edit_scenario

© 2025 Microchip Technology Inc. and its subsidiaries

- smartpower_remove_scenario

9.3. smartpower_add_pin_in_domain (Ask a Question)

Description

This tcl command adds a pin into a clock or set domain.

```
smartpower_add_pin_in_domain -pin_name {pin name} \
                             -pin_type {value} \
                             -domain_name {domain name} \
                             -domain_type {value}
```

Arguments

Parameter	Type	Description
pin_name	string	Specifies the name of the pin to add to the domain.
pin_type	string	Specifies the type of the pin to add. The acceptable values for this argument are the following: • clock - The pin to add is a clock pin. • data - The pin to add is a data pin.
domain_name	string	Specifies the name of the domain in which to add the specified pin.
domain_type	string	Specifies the type of domain in which to add the specified pin. The acceptable values for this argument are the following: • clock - The domain is a clock domain. • set - The domain is a set domain.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_name' is missing.
None	Required parameter 'pin_type' is missing.
None	Required parameter 'domain_type' is missing.
None	Required parameter 'domain_name' is missing.
None	domain_type: Invalid argument value: 'value' (expecting clock or set).
None	Parameter 'pin_name' has illegal value.
None	Failed to add pin "pin_name" to "domain".
None	pin_type: Invalid argument value: 'value' (expecting clock or data).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_add_pin_in_domain [-pin_name "name of pin"]+ -domain_type "clock | set" -domain_name "name of domain" -pin_type "clock | data"'.

Supported Families

Supported Families	Supported Versions
PolarFire®	v12.4+
SmartFusion® 2	v12.4+
RTG4™	v12.4+
IGLOO® 2	v12.4+
PolarFire SoC	v12.6+

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example adds a "XCMP3/U0/U1:Y" clock pin to an existing Clock "clk" domain:

```
smartpower_add_pin_in_domain -pin_name {XCMP3/U0/U1:Y} \
                             -pin_type {clock} \
                             -domain_name {clk} \
                             -domain_type {clock}
```

The following example adds a "XCMP3/U0/U1:Y" data pin to an existing Set "myset" domain:

```
smartpower_add_pin_in_domain -pin_name {XCMP3/U0/U1:Y} \
                             -pin_type {data} \
                             -domain_name {myset} \
                             -domain_type {set}
```

See Also

- smartpower_create_domain

9.4. smartpower_battery_settings (Ask a Question)

Description

This SmartPower Tcl command sets the battery capacity in SmartPower. The battery capacity is used to compute the battery life of your design.

```
smartpower_battery_settings [-capacity {decimal value}]
```

Arguments

Parameter	Type	Description
capacity	decimal	Specify the battery capacity in mA*Hours. Value must be a positive decimal. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	capacity: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'capacity' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_battery_settings [-capacity "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the battery capacity to 1800 mA * Hours.

```
smartpower_battery_settings -capacity {1800}
```


© 2025 Microchip Technology Inc. and its subsidiaries

9.5. smartpower_change_clock_statistics (Ask a Question)

Description

This Tcl command changes the default frequencies and probabilities for a specific domain.

Note: This command is associated with the functionality of Initialize frequencies and probabilities dialog box.

```
smartpower_change_clock_statistics -domain_name {value} \
                                   -clocks_freq {value} \
                                   -clocks_proba {value} \
                                   -registers_freq {value} \
                                   -registers_proba {value} \
                                   -set_reset_freq {value} \
                                   -set_reset_proba {value} \
                                   -primaryinputs_freq {value} \
                                   -primaryinputs_proba {value} \
                                   -combinational_freq {value} \
                                   -combinational_proba {value}
```

Arguments

Parameter	Type	Description
domain_name	string	Specifies the domain name in which to initialize frequencies and probabilities.
clocks_freq	string	Specifies the user input frequency in Hz, KHz or MHz for all clocks. Must be a positive decimal value.
clocks_proba	decimal	Specifies the user input probability in percentage(%) for all clocks. Must be a positive decimal value and less than or equal to 100.000.
registers_freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate in percentage(%). Must be a positive decimal value and less than or equal to 100.000. If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
registers_proba	decimal	Specifies the user input probability in percentage(%) for all registers. Must be a positive decimal value and less than or equal to 100.000.
set_reset_freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate in percentage(%). Must be a positive decimal value. If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
set_reset_proba	decimal	Specifies the user input probability in percentage(%) for all set/reset nets. Must be a positive decimal value and less than or equal to 100.000.
primaryinputs_freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate in percentage(%). Must be a positive decimal value. If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
primaryinputs_proba	decimal	Specifies the user input probability in percentage(%) for all primary inputs. Must be a positive decimal value and less than or equal to 100.000.
combinational_freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate percentage(%). Must be a positive decimal value. If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
combinational_proba	decimal	Specifies the user input probability in percentage(%) for all combinational combinational output. Must be a positive decimal value and less than or equal to 100.000.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'domain_name' is missing.
None	Parameter 'domain_name' has illegal value.
None	Parameter 'combinational_proba' has illegal value.
None	combinational_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'combinational_proba' must be a positive decimal value.
None	Parameter 'combinational_proba' must be less than or equal to 100.000.
None	Parameter 'combinational_freq' has illegal value.
None	combinational_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'primaryinputs_proba' has illegal value.
None	primaryinputs_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'primaryinputs_proba' must be a positive decimal value.
None	Parameter 'primaryinputs_proba' must be less than or equal to 100.000.
None	Parameter 'primaryinputs_freq' has illegal value.
None	primaryinputs_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'set_reset_proba' has illegal value.
None	set_reset_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'set_reset_proba' must be a positive decimal value.
None	Parameter 'set_reset_proba' must be less than or equal to 100.000.
None	Parameter 'set_reset_freq' has illegal value.
None	set_reset_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'registers_proba' has illegal value.
None	registers_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'registers_proba' must be a positive decimal value.
None	Parameter 'registers_proba' must be less than or equal to 100.000.
None	Parameter 'registers_freq' has illegal value.
None	registers_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'clocks_proba' has illegal value.
None	clocks_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'clocks_proba' must be a positive decimal value.
None	Parameter 'clocks_proba' must be less than or equal to 100.000.
None	Parameter 'clocks_freq' has illegal value.
None	clocks_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_change_clock_statistics -domain_name "domain name" \ [-clocks_freq "decimal value [unit { Hz | KHz | MHz }]"] \ [-clocks_proba "decimal value"] \ [-registers_freq "decimal value [unit { Hz | KHz | MHz | % }]"] \ [-registers_proba "decimal value"] \ [-set_reset_freq "decimal value [unit { Hz | KHz | MHz | % }]"] \ [-set_reset_proba "decimal value"] \ [-primaryinputs_freq "decimal value [unit { Hz | KHz | MHz | % }]"] \ [-primaryinputs_proba "decimal value"] \ [-combinational_freq "decimal value [unit { Hz | KHz | MHz | % }]"] \ [-combinational_proba "decimal value"] '.

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all clocks with:

```
smartpower_change_clock_statistics -domain_name {my_domain} \
                                   -clocks_freq {10 MHz} \
                                   -clocks_proba {20} \
                                   -registers_freq {10 MHz} \
                                   -registers_proba {20} \
                                   -set_reset_freq {10MHz} \
                                   -set_reset_proba {20} \
                                   -primaryinputs_freq {10 MHz} \
                                   -primaryinputs_proba {20} \
                                   -combinational_freq {10 MHz} \
                                   -combinational_proba {20}
```

See Also

- smartpower_init_set_combinational_options
- smartpower_init_set_primaryinputs_options

9.6. smartpower_change_setofpin_statistics (Ask a Question)

Description

This tcl command changes the default frequencies and probabilities for a specific set.

Note: This command is associated with the functionality of Initialize frequencies and probabilities dialog box.

```
smartpower_change_setofpin_statistics \
            -domain_name "domain name" \
            [-data_freq "decimal value [unit { Hz | KHz | MHz }]"] \
            [-data_proba "decimal value"]
```

Arguments

Parameter	Type	Description
domain_name	string	Specifies the domain name in which to initialize data frequencies and probabilities.
data_freq	string	Specifies the user input data frequency in Hz, KHz, or MHz for all sets of pins.
data_proba	decimal	Specifies the user input data probability in % for all sets of pins.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'domain_name' is missing.
None	Parameter 'domain_name' has illegal value.
None	Parameter 'data_proba' has illegal value.
None	data_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'data_freq' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_change_setofpin_statistics (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_change_setofpin_statistics -domain_name "domain name" \ [-data_freq "decimal value [unit { Hz | KHz | MHz }]"] \ [-data_proba "decimal value"]'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

The following example initializes all clocks withs:

```
smartpower_change_setofpin_statistics -domain_name {my_domain} \
                                      -data_freq {10 MHz} \
                                      -data_proba {20}
```

9.7. smartpower_commit (Ask a Question)

Description

This Tcl command saves the changes to the design file.

```
smartpower_commit [-file "SmartPower settings file (.zip)"]
```

Arguments

Parameter	Type	Description
file	string	Path to the *.zip where power analysis details will be saved. You can specify a relative or absolute path. This parameter is optional. If the argument is not specified "smartpower.swp" file is created under <project path>/designer/<component name> directory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_commit [-file "SmartPower settings file (.zip)"]'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example saves changes, power analysis details in the "/prj/designer/top/sp_details.zip":

```
smartpower_commit -file "./sp_details.zip"
```

9.8. smartpower_compute_vectorless (Ask a Question)

Description

This Tcl command executes a vectorless analysis of the current operating mode.

```
smartpower_compute_vectorless
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_compute_vectorless'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

```
smartpower_compute_vectorless
```

9.9. smartpower_create_domain (Ask a Question)

Description

This Tcl command creates a new clock or set domain.

Note: The domain name cannot be the name of an existing domain. The domain type must be either clock or set.

```
smartpower_create_domain -domain_type {domain type} -domain_name {domain name}
```

Arguments

Parameter	Type	Description
domain_type	string	Specifies the type of domain to create. The acceptable values for this argument are: • clock - The domain is a clock domain. • set - The domain is a set domain.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_create_domain (continued)		
Parameter	**Type**	**Description**
domain_name	string	Specifies the name of the new domain.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'domain_name' has illegal value.
None	Required parameter 'domain_name' is missing.
None	domain_type: invalid argument value: 'type_name'(expecting set or clock).
None	A domain with name "domain_name" already exists.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_create_domain -domain_name "domain name" -domain_type "set | clock"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example creates a new set domain named "myset":

```
smartpower_create_domain -domain_type {set} -domain_name {myset}
```

See Also

- smartpower_init_do
- smartpower_remove_domain

9.10. smartpower_edit_custom_mode (Ask a Question)

Description

This Tcl command edits a custom mode. You should specify at least one of the following optional parameters: new name and description.

```
smartpower_edit_custom_mode -name {old mode name} \
                            [-new_name {new mode name}] \
                            [-description {mode description}]
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the custom mode you want to edit. This parameter is mandatory.
new_name	string	Specifies the new name of the custom mode. This parameter is optional.
description	string	Specifies the description of the new custom mode. This parameter is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	Parameter 'new_name' has illegal value.
None	Parameter 'description' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_edit_custom_mode -name "name" [-description "description"] [-new_name "new mode name"]'.
None	A custom mode with name "mode_name" does not exist.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example edits custom mode "CustomMode" and renames it "CustomMode2":

```
smartpower_edit_custom_mode -name {CustomMode} \
                            -new_name {CustomMode2} \
                            -description {frequency 10 MHz}
```

See Also

- smartpower_add_new_custom_mode
- smartpower_remove_custom_mode

9.11. smartpower_edit_scenario (Ask a Question)

Description

This Tcl command edits a custom scenario: scenario name, previously defined mode(s) and duration(s).

```
smartpower_edit_scenario -name {custom scenario name} \
                         [-description {description of scenario}] \
                         -mode {mode_name:duration} \
                         -new_name {New Scenario name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the scenario.
description	string	Specifies the description of the scenario.
mode	string	Specifies the mode(s) and duration(s) for the specified scenario. Possible values are {<operating mode>:<duration>}
new_name	string	Specifies the new name for the scenario.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'new_name' has illegal value.
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_edit_scenario -name "scenario name" [-description "description"] [-mode "":""]* [-new_name "new mode name"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example edits the name of "MyScenario" to "NewScenario":

```
smartpower_edit_scenario -name {MyScenario} -new_name {NewScenario} -mode "Active:100.00"
```

See Also

- smartpower_add_new_scenario
- smartpower_remove_scenario

9.12. smartpower_export_mpe_report (Ask a Question)

Description

This Tcl command exports the Microchip Power Estimation(MPE) report in XML format. The generated `.xml` report contains the following information:

- Device Settings
- Thermal Settings
- Voltage Source
- Clocks
- Logic Breakdown
- LSRAM Breakdown with Advanced Settings
- uSRAM Breakdown with Advanced Settings
- Math Breakdown with Advanced Settings
- PLL and DLL
- I/Os
- Crypto
- Transceivers

In addition, the following information is available for PolarFire SoC devices:

- MSS RISC-V (Quad U54)
- AXI MSS/Fabric Interfaces
- MDDR
- MSS I/O Interfaces

© 2025 Microchip Technology Inc. and its subsidiaries

- User Crypto

Note: This command supported only for G5 families.

```
smartpower_export_mpe_report -filename {file_name.xml}
```

Arguments

Parameter	Type	Description
file_name	string	Name of the XML file to be exported. This argument is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'filename' has illegal value.
None	Required parameter 'filename' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_export_mpe_report -filename "filename"'.

Supported Families

PolarFire®
PolarFire SoC

Example

The following command exports the Microchip Power Estimation (MPE) report in mpe_report.xml file:

```
smartpower_export_mpe_report -filename {mpe_report.xml}
```

9.13. smartpower_get_temperature (Ask a Question)

Description

Enter description here

```
smartpower_get_temperature [-what "TRUE | FALSE"] [-opcond "Best | Typical | Worst"]
```

Arguments

Parameter	Type	Description
what	boolean	
opcond	string	Specifies the operating condition. The acceptable values for this argument are the following: • Worst - The operating condition is set to worst case. • Typical - The operating condition is set to typical case. • Best - The operating condition is set to best case.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined.
None	what: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0, false, ambient or opcond).
None	opcond: Invalid argument value: 'value' (expecting Best, Typical or Worst).

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2

Example

Enter example description here

```
smartpower_get_temperature  -what "TRUE" -opcond "Best"
```

9.14. smartpower_get_tetaja (Ask a Question)

Description

Enter description here

```
smartpower_get_tetaja [-style "case_cooling | still_air | 1.0_mps | 2.5_mps | custom"]
```

Arguments

Parameter	Type	Description
style	string	

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined.
None	style: Invalid argument value: 'value' (expecting case_cooling, still_air, 1.0_mps, 2.5_mps or custom).

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2

Example

Enter example description here

```
smartpower_get_tetaja -style "custom"
```


© 2025 Microchip Technology Inc. and its subsidiaries

9.15. smartpower_get_thermalmode (Ask a Question)

Description

Enter description here

```
smartpower_get_thermalmode
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

Enter example description here

```
smartpower_get_thermalmode
```

9.16. smartpower_import_vcd (Ask a Question)

Description

This Tcl command imports into SmartPower a VCD file generated by a simulation tool. SmartPower extracts the frequency and probability information from the VCD.

Note: SmartPower stops importing VCD in Static mode as now VCD basically contains signal transitions and Static mode means that the device is off. VCD file generation refer to the simulation related help in libero user guide.

```
import_vcd -file {VCD file} \
           [-opmode {mode name}] \
           [-with_vectorless {TRUE | FALSE}] \
           [-partial_parse {TRUE | FALSE}] \
           [-start_time {decimal value}] \
           [-end_time {decimalvalue}] \
           [-auto_detect_top_level_name {TRUE | FALSE}] \
           [-top_level_name {top level name}] \
           [-glitch_filtering {false | auto | true}] \
           [-glitch_threshold {integer value}] \
           [-stop_time {decimal value}]
```

Arguments

Parameter	Type	Description
file	string	Absolute path to a VCD file. Value must be a file path. This parameter is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_import_vcd (continued)		
Parameter	**Type**	**Description**
opmode	string	Operating mode in which the VCD will be imported. Operating mode name "Active" by default. If the mode doesn't exist, it will be created. Value must be a string. This parameter is optional.
with_vectorless	boolean	Specify the method to set the frequency and probability information for signals not annotated by the VCD TRUE(default): use the vectorless analysis, FALSE: use average value computed from the VCD. Value must be a boolean. This parameter is optional.
partial_parse	boolean	Enable partial parsing of the VCD. Specify the Smart time and End time to partially parse the VCD file. Start time and End time need to be specified when TRUE. Value must be one of TRUE, 1, true, FALSE, 0 or false. By default is FALSE. This parameter is optional.
start_time	decimal	Specify the starting timestamp of the VCD extraction in ns. It must be lower than the specified end_time. It must be lower than the last timestamp in the VCD file. Value must be a positive decimal nanoseconds(ns). This parameter is optional.
end_time	decimal	Specify the end timestamp of the VCD extraction in ns. It must be higher than the specified start_time. Value must be a positive decimal nanoseconds(ns). This parameter is optional.
auto_detect_top_level_name	boolean	Enable the auto detection of the top level name in the VCD file. Top_level_name needs to be specified when FALSE. Value must be a boolean. By default is TRUE. This parameter is optional.
top_level_name	string	Specify the full hierarchical name of the instance of the design in the VCD file. Value must be a string. This parameter is optional.
glitch_filtering	string	Enable to filter pulses of short duration by specifying automatic glitch filtering or by specifying vlaue to the filtering threshold. This parameter is optional. Value must be on of the following: • AUTO - Enable glitch filtering with predefined threshold based on the family. • TRUE - Enable glitch filtering, glitch_threshold must be specified. • FALSE - Disable glitch filtering.
glitch_threshold	integer	Specify the threshold in ps below which glitches are filtered out. Value must be a positive integer. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Mode "Static" is not active, please specify an active mode.
None	opmode: Invalid argument value: 'mode_name' (expecting Active or Static).
None	partial_parse: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false)
None	start_time: Invalid argument value: 'time_value' (expecting decimal value).
None	Parameter 'start_time' must be a positive decimal value.
None	end_time: Invalid argument value: 'time_value' (expecting decimal value).
None	Parameter 'end_time' must be a positive decimal value.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_import_vcd (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_import_vcd [-format "file format"] \ -file "VCD file" \ [-opmode "mode name"] \ [-with_vectorless "TRUE | FALSE"] \ [-partial_parse "TRUE | FALSE"] \ [-start_time "decimal value"] \ [-end_time "decimal value"] \ [-auto_detect_top_level_name "TRUE | FALSE"] \ [-top_level_name "top level name"] \ [-glitch_filtering "false | auto | true"] \ [-glitch_threshold "integer value"] \ [-stop_time {decimal value}]'

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The Tcl command imports the power.vcd file generated by the simulator into SmartPower:

```
smartpower_import_vcd -file {/home/example/simulation/power.vcd}
```

The Tcl command extracts information between 1ms and 2ms in the simulation, and stores the information into a custom mode:

```
smartpower_import_vcd -file {/home/example/simulation/power.vcd} -partial_parse {TRUE}
-start_time {1000000} -end_time {2000000} -opmode {power_1ms_to_2ms}
```

9.17. smartpower_init_do (Ask a Question)

Description

This Tcl command initializes the frequencies and probabilities for clocks, registers, set/reset nets, primary inputs, combinational outputs, enables and other sets of pins, and selects a mode for initialization.

```
smartpower_init_do [-opmode "Active"] \
           [-with "vectorless | default"] \
           [-clocks "TRUE | FALSE"] \
           [-registers "TRUE | FALSE"] \
           [-set_reset "TRUE | FALSE"] \
           [-primaryinputs "TRUE | FALSE"] \
           [-combinational "TRUE | FALSE"] \
           [-enables "TRUE | FALSE"] \
           [-othersets "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
opmode	string	This parameter is optional and specifies the mode in which to initialize frequencies and probabilities. The value must be Active or Flash*Freeze (RTG4, SmartFusion 2 and IGLOO 2).
with	string	This sets the option of initializing frequencies and probabilities with vectorless analysis or with fixed values. The acceptable values for this argument are the following: • vectorless - Initializes frequencies and probabilities with vectorless analysis. • fixed - Initializes frequencies and probabilities with fixed values.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_init_do (continued)		
Parameter	**Type**	**Description**
clocks	boolean	This sets the option of initializing frequencies and probabilities for all clocks. The following table shows theacceptable values for this argument: • TRUE, true or 1 - Initializes frequencies and probabilities for all clocks. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all clocks.
registers	boolean	This sets the option of initializing frequencies and probabilities for all registers. The following table shows theacceptable values for this argument: • TRUE, true or 1 - Initializes frequencies and probabilities for all registers. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all registers.
set_reset	boolean	This sets the option of initializing frequencies and probabilities for all set/reset nets. The following table shows theacceptable values for this argument: • TRUE, true or 1 - Initializes frequencies and probabilities for all set/reset nets. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all set/reset nets.
primaryinputs	boolean	This sets the option of initializing frequencies and probabilities for all primary inputs. The acceptable values for this argument are the following: • TRUE, true or 1 - Initializes frequencies and probabilities for all primary inputs. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all primary inputs.
combinational	boolean	This sets the option of initializing frequencies and probabilities for all combinational outputs. The acceptable values for this argument are the following: • TRUE, true or 1 - Initializes frequencies and probabilities for all combinational outputs. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all combinational outputs.
enables	boolean	This sets the option of initializing frequencies and probabilities for all enable sets of pins. The acceptable values for this argument are the following: • TRUE, true or 1 - Initializes frequencies and probabilities for all enable sets of pins. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all enable sets of pins.
othersets	boolran	This sets the option of initializing frequencies and probabilities for all other sets of pins. The acceptable values for this argument are the following: • TRUE, true or 1 - Initializes frequencies and probabilities for all other sets of pins. • FALSE, false or 0 - Does not initialize frequencies and probabilities for all other sets of pins

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'othersets' has illegal value.
None	Parameter 'enables' has illegal value.
None	enables: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'combinational' has illegal value.
None	combinational: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'primaryinputs' has illegal value.
None	primaryinputs: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'set_reset' has illegal value.
None	set_reset: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'registers' has illegal value.
None	registers: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'clocks' has illegal value.
None	clocks: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'opmode' has illegal value.
None	opmode: Invalid argument value: 'Static' (expecting Active or Flash*Freeze).
None	Parameter 'with' has illegal value.
None	with: Invalid argument value: 'value' (expecting vectorless, default or fixed).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_do [-opmode "Active"] \ [-with "vectorless | default"] \ [-clocks "TRUE | FALSE"] \ [-registers "TRUE | FALSE"] \ [-set_reset "TRUE | FALSE"] \ [-primaryinputs "TRUE | FALSE"] \ [-combinational "TRUE | FALSE"] \ [-enables "TRUE | FALSE"] \ [-othersets "TRUE | FALSE"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all clocks with:

```
smartpower_init_do -with {vectorless} \
                   -opmode {my_mode} \
                   -clocks {true} \
                   -registers{true} \
                   -asynchronous {true} \
                   -primaryinputs {true} \
                   -combinational {true} \
                   -enables {true} \
                   -othersets {true}
```

9.18. smartpower_initialize_clock_with_constraints (Ask a Question)

Description

This Tcl command initializes the clock frequency and the data frequency of a single clock domain with a specified clock name and the initialization options.

© 2025 Microchip Technology Inc. and its subsidiaries

Notes:

- This command is associated with the functionality of Initialize frequencies and probabilities dialog box.
- This command is associated with the right click menu Synchronize Domain with SmartTime on a single clock domain in the Domains tab.

```
smartpower_initialize_clock_with_constraints -clock_name {value}
```

Arguments

Parameter	Type	Description
clock_name	string	Specifies the name of the clock that will be initialized. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'clock_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_initialize_clock_with_constraints [-clock_name "name of clock"]+'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2

Example

The following example initializes "my_clock" with clock constraints from SmartTime:

```
smartpower_initialize_clock_with_constraints -clock_name {my_clock}
```

See Also

- smartpower_create_domain

9.19. smartpower_init_set_clocks_options (Ask a Question)

Description

This Tcl command initializes the clock frequency options of all clock domains.

Note: This command is associated with the functionality of Initialize frequencies and probabilities dialog box.

```
smartpower_init_set_clocks_options [-with_clock_constraints {value}] \
                                   [-with_default_values {value}] \
                                   [-freq {value}] \
                                   [-duty_cycle {value}]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
with_clock_constraints	boolean	This sets the option of initializing the clock frequencies with frequency constraints from SmartTime. The acceptable values for this argument are the following: • true - Sets initialize clock frequencies with clock constraints ON. • false - Sets initialize clock frequencies with clock constraints OFF.
with_default_values	boolean	This sets the option of initializing the clock frequencies with a user input default value. The acceptable values for this argument are the following: • true - Sets initialize clock frequencies with default values ON. • false - Sets initialize clock frequencies with default values OFF.
freq	string	Specifies the user input frequency in Hz, KHz or MHz.
duty_cycle	decimal	Specifies the user input duty cycles in percentage(%).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'duty_cycle' has illegal value.
None	Parameter 'duty_cycle' must be less than or equal to 100.000.
None	Parameter 'duty_cycle' must be a positive decimal value.
None	duty_cycle: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'freq' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'with_default_values' has illegal value.
None	with_default_values: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'with_clock_constraints' has illegal value.
None	with_clock_constraints: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_clocks_options [-with_clock_constraints "TRUE | FALSE"] [-with_default_values "TRUE | FALSE"] [-freq "decimal value [unit { Hz | KHz | MHz }]"] [-duty_cycle "decimal value"]'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

The following example initializes all clocks after executing "smartpower_init_do":

```
smartpower_init_do with -clocks {true}
smartpower_init_set_clocks_options -with_clock_constraints {true} \
                                   -with_default_values {true} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                                        -freq {10 MHz} \
                                        -duty_cycle {20}
```

9.20. smartpower_init_set_combinational_options (Ask a Question)

Description

This Tcl commands initializes the frequency and probability of all combinational outputs.

```
smartpower_init_set_combinational_options [-freq {value}] \
                                          [-proba {value}]
```

Arguments

Parameter	Type	Description
freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate in percentage(%). If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
proba	decimal	Specifies the user input probability in percentage(%). Must be a positive decimal value and less than or equal to 100.000.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'proba' has illegal value.
None	Parameter 'proba' must be a positive decimal value.
None	Parameter 'proba' must be less than or equal to 100.000.
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'freq' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_combinational_options [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] [-proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all combinational signals after executing smartpower_init_do with -combinational {true}:

```
smartpower_init_set_combinational_options -freq {10 MHz} -proba {20}
```

See Also

- smartpower_init_set_primaryinputs_options

© 2025 Microchip Technology Inc. and its subsidiaries

9.21. smartpower_init_setofpins_values (Ask a Question)

Description

This tcl command initializes the frequency and probability of all sets of pins.

```
smartpower_init_setofpins_values -domain_name "IOsEnableSet | MemoriesEnableSet" \
                         [-freq "decimal value [unit { Hz | KHz | MHz }]"] \
                         [-proba "decimal value"]
```

Arguments

Parameter	Type	Description
domain_name	string	Specifies the set of pins that will be initialized. The acceptable values for this argument are the following: • IOsEnableSet - Specifies that the IOsEnableSet set of pins will be initialized. • MemoriesEnableSet - Specifies that the MemoriesEnableSet set of pins will be initialized.
freq	string	Specifies the user input frequency in Hz, MHz, or KHz.
proba	decimal	Specifies the user input probability in percentage(%).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'domain_name' is missing.
None	Parameter 'domain_name' has illegal value.
None	domain_name: Invalid argument value: 'value' (expecting IOsEnableSet or MemoriesEnableSet).
None	Parameter 'proba' has illegal value.
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'freq' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_setofpins_values -domain_name "IOsEnableSet | MemoriesEnableSet" [-freq "decimal value [unit { Hz | KHz | MHz }]"] [-proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all primary inputs after executing "smartpower_init_do with":

```
smartpower_init_do with -othersets {true}
smartpower_init_setofpins_values -domain_name {IOsEnableSet} \
                                 -freq {10 MHz} \
                                 -proba{20}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- smartpower_init_do

9.22. smartpower_init_set_othersets_options (Ask a Question)

Description

This Tcl command initializes the frequency and probability of all other sets.

Note: This command is associated with the functionality of Initialize Frequencies and Probabilities dialog box.

```
smartpower_init_set_othersets_options \
                [-freq "decimal value [unit { Hz | KHz | MHz }]"] \
                [-proba "decimal value"] \
                [-with "fixed | default"] \
                [-input_freq "decimal value [unit { Hz | KHz | MHz }]"] \
                [-input_proba "decimal value"]
```

Arguments

Parameter	Type	Description
freq	string	Specifies the default frequency and units(possible values are Hz, KHz or MHz). This parameter is optional.
proba	decimal	Specifies the default probability. Must be positive decimal value, less than or equal to 100.000. This parameter is optional.
with	string	Specifies vectorless or default analysis. This parameter is optional.
input_freq	decimal	Specifies the input frequency(decimal possible value) and units(possible values are Hz, KHz or MHz). This parameter is optional.
input_proba	decimal	Specifies the input probability. Must be positive decimal value, less than or equal to 100.000. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	input_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'input_proba' has illegal value.
None	Parameter 'input_proba' must be less than or equal to 100.000.
None	Parameter 'input_proba' must be a positive decimal value.
None	input_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'input_freq' has illegal value.
None	with: Invalid argument value: 'value' (expecting default or fixed).
None	Parameter 'with' has illegal value.
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'proba' must be less than or equal to 100.000.
None	Parameter 'proba' must be a positive decimal value.
None	Parameter 'proba' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'freq' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_init_set_othersets_options (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_othersets_options [-freq "decimal value [unit { Hz | KHz | MHz }]"] [-proba "decimal value"] [-input_freq "decimal value [unit { Hz | KHz | MHz }]"] [-input_proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all other sets after executing "smartpower_init_do" with "-othersets {true}":
Note: Warning: -with flag is obsolete; use -with argument of the command smartpower_init_do instead.

```
smartpower_init_set_othersets_options \
                        -freq "10 MHz" \
                        -proba "10" \
             [-with "fixed"] \
                        -input_freq "10 MHz" \
                        -input_proba "10"
```

See Also

- smartpower_init_do

9.23. smartpower_init_set_primaryinputs_options (Ask a Question)

Description

This Tcl command initializes the frequency and probability of all primary inputs.

```
smartpower_init_set_primaryinputs_options \
              [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] \
              [-proba "decimal value"]
```

Arguments

Parameter	Type	Description
freq	string	Specifies the user input frequency (in Hz, KHz or MHz) or the toggle rate in percentage(%). If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency.
proba	decimal	Specifies the user input probability in percentage(%). Must be a positive decimal value and less than or equal to 100.000.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'proba' has illegal value.
None	proba: Invalid argument value: 'value' (expecting decimal value).

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_init_set_primaryinputs_options (continued)	
Error Code	**Description**
None	Parameter 'proba' must be a positive decimal value.
None	Parameter 'proba' must be less than or equal to 100.000.
None	Parameter 'freq' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_primaryinputs_options [-freq "decimal value [unit {Hz | KHz | MHz | %}]"] [-proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all primary inputs after executing smartpower_init_do with -primaryinputs {true}:

```
smartpower_init_set_primaryinputs_options -freq {10 MHz} -proba {20}
```

See Also

- smartpower_init_set_combinational_options
- smartpower_change_clock_statistics

9.24. smartpower_init_set_registers_options (Ask a Question)

Description

This Tcl command initializes the frequency and probability of all register outputs.

```
smartpower_init_set_registers_options \
        [-with "default"] \
            [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] \
        [-proba "decimal value"]
```

Arguments

Parameter	Type	Description
freq	string	Specifies the user input frequency (in Hz, KHz, or MHz) or the toggle rate in percentage(%). If the unit is not provided and togglerate is active, the value is handled as a toggle rate; if toggle rate is not active, the value is handled as a frequency.
proba	decimal	Specifies the user input probability in percentage(%). Must be less than or equal to 100.000.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'proba' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_init_set_registers_options (continued)	
Error Code	**Description**
None	Parameter 'proba' must be less than or equal to 100.000.
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'freq' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_registers_options [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] [-proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all register outputs after executing "smartpower_init_do with -registers {true}":

```
smartpower_init_set_registers_options -freq {10 MHz} \
                                      -proba {20}
```

9.25. smartpower_init_set_set_reset_options (Ask a Question)

Description

This Tcl command initializes the frequency and probability of all set and reset nets.

```
smartpower_init_set_set_reset_options \
         [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] \
         [-proba "decimal value"]
```

Arguments

Parameter	Type	Description
freq	decimal	Specifies the user input frequency (in Hz, KHz, or MHz) or the toggle rate (in %). If the unit is not provided and toggle rate is active, the value is handled as a toggle rate, if toggle rate is not active, the value is handled as a frequency. This parameter is optional.
proba	decimal	Specifies the user input probability in persentage(%). This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'proba' must be a positive decimal value.
None	Parameter 'proba' must be less than or equal to 100.000.
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'proba' has illegal value.
None	freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz, MHz or %)).

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_init_set_set_reset_options (continued)	
Error Code	**Description**
None	Parameter 'freq' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_init_set_set_reset_options [-freq "decimal value [unit { Hz | KHz | MHz | % }]"] [-proba "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example initializes all set/reset nets after executing "smartpower_init_do" with "-set_reset {true}":

```
smartpower_init_do -set_reset {true}
smartpower_init_set_set_reset_options -freq {10 MHz} -proba {20}
```

See Also

- smartpower_init_do

9.26. smartpower_remove_all_annotations (Ask a Question)

Description

This Tcl command removes all initialization annotations for the specified mode.

Note: This command is associated with the functionality of Initialize frequencies and probabilities dialog box.

```
smartpower_remove_all_annotations [-opmode {value}]
```

Arguments

Parameter	Type	Description
opmode	string	Removes all initialization annotations for the specified mode, where value must be Active or Flash*Freeze (RTG4, SmartFusion 2 and IGLOO 2).

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'opmode' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_all_annotations [-opmode "Active"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

IGLOO® 2
PolarFire SoC

Example

The following example initializes all clocks with opmode "Acitve":

```
smartpower_remove_all_annotations -opmode {Active}
```

9.27. smartpower_remove_custom_mode (Ask a Question)

Description

This Tcl command removes a custom mode from the current design.

```
smartpower_remove_custom_mode -name {deleted mode name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the custom mode to be removed. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	A custom mode with name "mode_name" does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_custom_mode -name "mode to deleted"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example removes a "MyCustomMode" custom mode from the current design:

```
smartpower_remove_custom_mode -name {MyCustomMode}
```

See Also

- smartpower_add_new_custom_mode
- smartpower_edit_custom_mode

© 2025 Microchip Technology Inc. and its subsidiaries

9.28. smartpower_remove_domain (Ask a Question)

Description

This Tcl command removes an existing clock or set domain.

```
smartpower_remove_domain -domain_name {domain name} -domain_type "set | clock"
```

Arguments

Parameter	Type	Description
domain_type	string	Specifies the type of domain to remove. The acceptable values for this argument are: • clock - The domain is a clock domain. • set - The domain is a set domain.
domain_name	string	Specifies the name of the domain to remove.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'domain_name' is missing.
None	Parameter 'domain_name' has illegal value.
None	domain_type: Invalid argument value: 'value' (expecting set or clock).
None	Could not delete domain 'domain name'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_domain -domain_name "domain name" -domain_type "set | clock"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example removes the clock domain names "myclk":

```
smartpower_remove_domain -domain_type {clock} -domain_name {myclk}
```

This example removes the set domain names "myset":

```
smartpower_remove_domain -domain_type {set} -domain_name {myset}
```

See Also

- smartpower_create_domain

© 2025 Microchip Technology Inc. and its subsidiaries

9.29. smartpower_remove_file (Ask a Question)

Description

This Tcl command removes the VCD file from the specified mode or all operating mode. Frequency and probability information of signals annotated by the VCD are set back to the default value.

```
remove file -file {the name of the VCD file} \
            -format {the type of the removed file} \
            -opmode {mode name}
```

Arguments

Parameter	Type	Description
file	string	Specifies the VCD file to be removed. This is mandatory.
format	string	Specifies that the type to be removed is a VCD file. This is mandatory.
opmode	string	Specifies the operating mode. This is optional. Below shows the acceptable values for this argument: • Active(PolarFire®, SmartFusion® 2, IGLOO® 2, RTG4™, PolarFire SoC) - the operating mode is set to Active. • Flash*Freeze(SmartFusion 2, IGLOO 2) - the operating mode is set to Flash*Freeze. • Custom mode name - the operating mode is custom cerated mode.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'format' has illegal value.
None	Required parameter 'format' is missing.
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	opmode: Invalid argument value: 'mode_name' (expecting Active, Static, Flash*Freeze, Shutdown, Sleep or CustomMode).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_file -file "file name" -format "vcd" [-opmode "Active | Static"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example removes the file test.vcd from the Active mode:

```
smartpower_remove_file -file "test.vcd" -format VCD -opmode "Active"
```

This example removes the VCD file power1.vcd from all operating modes:

```
smartpower_remove_file -file “power1.vcd” -format VCD
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- smartpower_import_vcd

9.30. smartpower_remove_pin_frequency (Ask a Question)

Description

This Tcl command removes the frequency associated with a specific pin. This pin will have a default frequency based on its domain.

Note: The pin_name must be the name of a pin that already exists in the design and already belongs to a domain. Execute the "smartpower_commit" Tcl command for preserving any changes in SmartPower.

```
smartpower_remove_pin_frequency -pin_name {pin_name}
```

Arguments

Parameter	Type	Description
pin_name	string	Specifies the name of the pin for which the frequency will be removed.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_name' is missing.
None	Parameter 'pin_name' is missing or has invalid value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_pin_frequency [-pin_name "name of pin"]+'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example removes the frequency from the pin named "d_in":

```
smartpower_remove_pin_frequency -pin_name {d_in}
```

See Also

- smartpower_set_pin_frequency

9.31. smartpower_remove_pin_of_domain (Ask a Question)

Description

This Tcl command removes a clock pin or a data pin from a clock or set domain, respectively.

Note: The domain name must be the name of an existing domain. The pin name must be the name of an existing pin. Execute the "smartpower_commit" Tcl command for preserving any changes in SmartPower.

```
smartpower_remove_pin_of_domain [-pin_name {name of pin}] \
                                -pin_type {clock | data} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                                -domain_name {name of domain} \
                                -domain_type {clock | set}
```

Arguments

Parameter	Type	Description
pin_name	string	Specifies the name of the pin to remove from the domain.
pin_type	string	Specifies the type of the pin to remove. The acceptable values for this argument are the following: • clock - The pin to remove is a clock pin. • set - The pin to remove is a data pin.
domain_name	string	Specifies the name of the domain from which to remove the pin.
domain_type	string	Specifies the type of domain from which the pin is being removed. The acceptable values for this argument are the following: • clock - The domain is a clock domain. • set - The domain is a set domain.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'domain_type' is missing.
None	domain_type: Invalid argument value: 'type name' (expecting clock or set).
None	Required parameter 'domain_name' is missing.
None	Required parameter 'pin_name' is missing.
None	Required parameter 'pin_type' is missing.
None	pin_type: Invalid argument value: 'type_name' (expecting clock or data).
None	Parameter 'pin_name' is missing or has invalid value.
None	Parameter 'domain_name' is missing or has invalid value.
None	Failed to remove pin "in" from domain 'domain_value'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_pin_of_domain [-pin_name "name of pin"]+ -domain_type "clock | set" -domain_name "name of domain" -pin_type "clock | data"'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

The following example removes the clock pin named "XCMP3/UO/U1:Y" from the clock domain named "myclk":

```
smartpower_remove_pin_of_domain -pin_name {XCMP3/U0/U1:Y} \
                                -pin_type {clock} \
                                -domain_name {myclk} \
                                -domain_type {clock}
```


© 2025 Microchip Technology Inc. and its subsidiaries

The following example removes the data pin named "count" from the set domain named "InputSet":

```
smartpower_remove_pin_of_domain -pin_name {count} \
                                -pin_type {data} \
                                -domain_name {InputSet} \
                                -domain_type {set}
```

9.32. smartpower_remove_pin_probability (Ask a Question)

Description

This tcl command removes the probability value associated with a specific pin. This pin will have a default probability based on the domain set it belongs to.

```
smartpower_remove_pin_probability -pin_name "name of pin"
```

Arguments

Parameter	Type	Description
pin_name	string	Specifies the name of the pin with the probability to remove. This pin must be the direct driver of an enable pin. There may be multiple `-pin_name` arguments. This is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_name' is missing.
None	Parameter 'pin_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_pin_probability [-pin_name "name of pin"]+'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example removes the probability of the pin driving the enable pin of a bidirectional I/O:

```
smartpower_remove_pin_probability -pin_name "mybibuf/U0/U1:EOUT"
```

See Also

- smartpower_set_pin_probability

9.33. smartpower_remove_scenario (Ask a Question)

Description

This Tcl command removes a custom scenario from the current design.

```
smartpower_remove_scenario -name {Scenario name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specifies the name of the custom scenario to be removed. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_remove_scenario -name "scenario name"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example removes a "MyScenario" custom scenario from the current design:

```
smartpower_remove_scenario -name {MyScenario}
```

See Also

- smartpower_add_new_scenario
- smartpower_edit_scenario

9.34. smartpower_report_power (Ask a Question)

Description

This Tcl command creates a power report, which enables you to determine if you have any power consumption problems in your design. It includes information about the global device and SmartPower preferences selection and hierarchical details (including gates, blocks, and nets), with a block-by-block, gate-by-gate, and net-by-net power summary SmartPower results.

Notes:

- Flash*Freeze is available only for certain families and devices(RTG4, SmartFusion 2 and IGLOO 2).
- Worst and Best are available only for certain families and devices.

```
smartpower_report_power \
              [-powerunit "W | mW | uW"] \
              [-frequnit "Hz | KHz | MHz"] \
              [-opcond "Best | Typical | Worst"] \
              [-opmode "Active | Static"] \
              [-toggle "TRUE | FALSE"] \
              [-power_summary "TRUE | FALSE"] \
              [-rail_breakdown "TRUE | FALSE"] \
              [-type_breakdown "TRUE | FALSE"] \
              [-clock_breakdown "TRUE | FALSE"] \
              [-thermal_summary "TRUE | FALSE"] \
              [-battery_life "TRUE | FALSE"] \
              [-clock_summary "TRUE | FALSE"] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
[-opcond_summary "TRUE | FALSE"] \
[-annotation_coverage "TRUE | FALSE"] \
[-style "Text | CSV"] \
[-sortorder "Ascending | Descending"] \
[-sortby "Alphabetical | Power Values"] \
[-instance_breakdown "TRUE | FALSE"] \
[-power_threshold "TRUE | FALSE"] \
[-filter_instance "TRUE | FALSE"] \
[-min_power "decimal value [unit { W | mW | uW }]"] \
[-max_instance "integer value"] \
[-activity_sortorder "Ascending | Descending"] \
[-activity_sortby "Pin Name | Net Name | Domain | Frequency | Source"] \
[-activity_summary "TRUE | FALSE"] \
[-frequency_threshold "TRUE | FALSE"] \
[-filter_pin "TRUE | FALSE"] \
[-min_frequency "decimal value [unit { Hz | KHz | MHz }]"] \
[-max_pin "integer value"] \
[-enablerates_summary "TRUE | FALSE"] \
[-enablerates_sortorder "Ascending | Descending"] \
[-enablerates_sortby "Pin Name | Net Name | Type | Polarity | Rate | Source"] \
{filename}
```

Arguments

Parameter	Type	Description
powerunit	string	Specifies the unit in which power is set. The acceptable values for this argument are the following: • W - The power unit is set to watts. • mW - The power unit is set to milliwatts. • uW - The power unit is set to microwatts.
frequnit	string	Specifies the unit in which frequency is set. The acceptable values for this argument are the following: • Hz - The frequency unit is set to hertz. • kHz - The frequency unit is set to kilohertz. • MHz - The frequency unit is set to megahertz.
opcond	string	Specifies the operating condition. The following acceptable values for this argument are the following: • worst - The operating condition is set to worst case. • typical - The operating condition is set to typical case. • best - The operating condition is set to best case.
opmode	string	Specifies the operating mode. The following acceptable values for this argument are the following: • Active - The operating mode is set to Active. • Static - The operating mode is set to Static. • Flash*Freeze - The operating mode is set to Flash*Freeze.
toggle	boolean	Specifies the toggle. The acceptable values for this argument are thr following: • TRUE, true or 1 - The toggle is set to true. • FALSE, false or 1 - The toggle is set to false.
power_summary	boolean	Specifies whether to include the power summary, which shows the static and dynamic values in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the power summary in the report. • FALSE, false or 0 - Does not include the power summary in the report.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power (continued)

Parameter	Type	Description
rail_breakdown	boolean	Specifies whether to include the breakdown by rail summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the breakdown by rail summary in the report. • FALSE, false or 0 - Does not include the breakdown by rail summary in the report.
type_breakdown	boolean	Specifies whether to include the breakdown by type summary in the report. The acceptable values for this argument are thr following: • TRUE, true or 1 - Includes the breakdown by type summary in the report. • FALSE, false or 0 - Does not include the breakdown by type summary in the report.
clock_breakdown	boolean	Specifies whether to include the breakdown by clock domain in the report. The acceptable values for this argument are thr following: • TRUE, true or 1 - Includes the breakdown by clock domain summary in the report. • FALSE, false or 0 - Does not include the breakdown by clock domain summary in the report.
thermal_summary	boolean	Specifies whether to include the thermal summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the thermal summary in the report. • FALSE, false or 0 - Does not include the thermal summary in the report.
battery_life	boolean	Specifies whether to include the battery life summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the battery life summary in the report. • FALSE, false or 0 - Does not include the battery life summary in the report.
opcond_summary	boolean	Specifies whether to include the operating conditions summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the operating conditions summary in the report. • FALSE, false or 0 - Does not include the operating conditions summary in the report.
clock_summary	boolean	Specifies whether to include the clock domains summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the clock summary in the report. • FALSE, false or 0 - Does not include the clock summary in the report.
style	string	Specifies the format in which the report will be exported. The acceptable values for this argument are the following: • Text - The report will be exported as Text file. • CSV(by default) - The report will be exported as CSV file.
sortby	string	Specifies how to sort the values in the report. The acceptable values for this argument are the following: • power values - Sorts based on the power values. • alphabetical - Sorts in an alphabetical order.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power (continued)

Parameter	Type	Description
sortorder	string	Specifies the sort order of the values in the report. The acceptable values for this argument are the following: • ascending - Sorts the values in ascending order. • descending - Sorts the values in descending order.
instance_breakdown	boolean	Specifies whether to include the breakdown by instance in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the breakdown by instance in the report. • FALSE, false or 0 - Does not include the breakdown by instance in the report.
power_threshold	boolean	This specifies whether to include only the instances that consume power above a certain minimum value. Whenthis command is set to true, the -min_power argument must also be used to specify that only the instances thatconsume power above this minimum power value are the ones that are included in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the power threshold in the report. • FALSE, false or 0 - Does not include the power threshold in the report.
filter_instance	boolean	This specifies whether to have a limit on the number of instances to include in the Power report. When this commandis set to true, the `-max_instance` argument must also be used to specify the maximum number of instances to beincluded into the Power report. The acceptable values for this argument are the following: • TRUE, true or 1 - Indicates that you want to have a limit on the number of instances to include in the Powerreport. • FALSE, false or 0 - Indicates that you do not want to have a limit on the number of instances to include in thePower report.
min_power	decimal	Specifies which block to expand based on the minimum power value of a block.
max_instance	integer	Sets the maximum number of instances to a specified integer greater than or equal to 0 (zero). This will limit the maximum number of instances to be included in the Power report.
activity_sortorder	string	Specifies the sort order for the activity summary. The acceptable values for this argument are the following: • ascending - Sorts the values in ascending order. • descending - Sorts the values in descending order.
activity_sortby	string	Specifies how to sort the values for the activity summary. The acceptable values for this argument are the following: • pin name - Sorts based on the pin name. • net name - Sorts based on thepnet name. • domain - Sorts based on the clock domain. • frequency - Sorts based on the clock frequency. • source - Sorts based on the clock frequency source.
activity_summary	boolean	Specifies whether to include the activity summary in the report. The following acceptable values for this argument are the following: • TRUE, true or 1 - Includes the activity summary in the report. • FALSE, false or 0- Does not include the activity summary in the report.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power (continued)		
Parameter	**Type**	**Description**
frequency_threshold	boolean	Specifies whether to add a frequency threshold. The acceptable values for this argument are the following: • TRUE, true or 1 - Adds a frequency threshold. • FALSE, false or 0 - Does not add a frequency threshold.
filter_pin	boolean	Specifies whether to filter by maximum number of pins. The acceptable values for this argument are the following: • TRUE, true or 1 - Filters by maximum number of pins. • FALSE, false or 0 - Des not filter by maximum number of pins.
min_frequency	decimal	Sets the minimum frequency to {decimal value [unit { Hz | KHz | MHz}]}.
max_pin	integer	Sets the maximum number of pins.
enablerates_sortorder	string	Specifies the sort order for the probabilities summary. The acceptable values for this argument are the following: • ascending - Sorts the values in ascending order. • descending - Sorts the values in descending order.
enablerates_sortby	string	Specifies how to sort the values for the probabilities summary. The acceptable values for this argument are the following: • pin name - Sorts based on the pin name. • net name - Sorts based on the net name. • domain - Sorts based on the clock domain. • frequency - Sorts based on the clock frequency. • source - Sorts based on the clock frequency source.
enablerates_summary	boolean	Specifies whether to include the probabilities summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the activity summary in the report. • FALSE, false or 0 - Does not include the activity summary in the report.
with_annotation_coverage	boolean	Specifies whether to include the annotation coverage summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the annotation coverage summary in the report. • FALSE, false or 0 - Does not include the annotation coverage summary in the report.
filename	string	Specifies the name or path of the file to be exported. This argument is mandatory. Default created under designer/<root_name> dircetory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	opcond: Invalid argument value: 'value' (expecting Best, Typical or Worst).
None	powerunit: Invalid argument value: 'value' (expecting W, mW or uW).
None	opmode: Invalid argument value: 'value' (expecting Active, Static or Flash*Freeze).
None	frequnit: Invalid argument value: 'value' (expecting Hz, KHz or MHz).
None	toggle: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power (continued)	
Error Code	**Description**
None	power_summary: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	rail_breakdown: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	type_breakdown: Invalid argument value: ' value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	clock_breakdown: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	thermal_summary: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	battery_life: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	power_threshold: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	min_power: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are W, mW or uW)).
None	filter_instance: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	max_instance: Invalid argument value: 'integer' (expecting integer value).
None	Parameter 'max_instance' must be a positive integer value.
None	activity_sortorder: Invalid argument value: 'value' (expecting Ascending or Descending).
None	activity_summary: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	frequency_threshold: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	filter_pin: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	max_pin: Invalid argument value: 'value' (expecting integer value).
None	Parameter 'max_pin' must be a positive integer value.
None	enablerates_sortorder: Invalid argument value: 'value' (expecting Ascending or Descending).
None	enablerates_sortby: Invalid argument value: 'value' (expecting Pin Name, Net Name, Type, Polarity, Rate or Source).
None	enablerates_summary: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example generates a Power report named power_report.xml:

```
smartpower_report_power -frequnit "MHz" \
                        -opcond "Typical" \
                        -opmode "Active" \
                        -toggle "TRUE" \
                        -battery_life "TRUE" \
                        -style "TEXT" \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                        -power_summary "TRUE" \
                        -activity_sortby "Source" \
                        power_report.txt
```

9.35. smartpower_report_power_activity_map (Ask a Question)

Description

This Tcl command the activity and hazards report reads a VCD file and reports transitions and hazards for each clock cycle of the VCD file.

```
smartpower_report_power_activity_map \
              -vcd_file "VCD file" \
              [-style "Text | CSV"] \
              [-partial_parse "TRUE | FALSE"] \
              [-start_time "decimal value"] \
              [-end_time "decimal value"] \
              [-auto_detect_top_level_name "TRUE | FALSE"] \
              [-top_level_name "top level name"] \
              -report_query "Report by Cycle - summary | Report by Net - summary | Report by
Cycle - detailed | Report by Net - detailed"\
              [-report_type "activity and power | activity | power"] \
              [-sortby "functional transitions | total power | spurious transitions |
functional power | spurious power"] \
              [-sortorder "ascending | descending"] \
              [-max_cycle "integer value"] \
              [-max_net "integer value"] \
              [-clock_settings "":""] \
              [-glitch_filtering "false | auto | true"] \
              [-glitch_threshold "integer value"] \
              [-auto_construct_clock_domain "TRUE | FALSE"] \
              [-clock_period "decimal value"] \
              [-clock_offset "decimal value"] \
              [-opmode "Active | Static"] \
              [-file "file"]
```

Arguments

Parameter	Type	Description
vcd_file	string	Specifies the path to the *.vcd file that you want to import.
style	string	Specifies the format in which the report will be exported. The acceptable values for this argument are the following: • Text - The report will be exported as Text file. • CSV(by default) - The report will be exported as CSV file.
partial_parse	boolean	Specifies whether to partially parse the *.vcd file. The acceptable values for this argument are the following: • TRUE, true or 1 - Partially parses the *.vcd file. • FALSE, false or 0 - Does not partially parse the *.vcd file.
start_time	decimal	This option is available only if `-partially_parse` is set to true. Specifies the start time (in ns) to partially parse the *.vcd file.
end_time	decimal	This option is available only if `-partially_parse` is set to true. Specifies the start time (in ns) to partially parse the *.vcd file.
auto_detect_top_level_name	boolean	Specifies whether to automatically detect the top-level name. The acceptable values for this argument are the following: • TRUE, true or 1 - Automatically detects the top-level name. • FALSE, false or 0 - Does not automatically detect the top-level name.
top_level_name	string	Specifies the top-level name.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power_activity_map (continued)

Parameter	Type	Description
report_type	string	Specifies the report query type. The acceptable values for this argument are the following: • activity - Includes activity information for each net. • power - Includes power information for each net. • activity and power - Includes activity and power information for each net.
report_query	string	Specifies the report type. The acceptable values for this argument are the following: • Report by Net - summary - Provides a summary report for each net. • Report by Net - detailed - Provides a detailed report for each net. • Report by Cycle - summary - Provides a summary report for each cycle. • Report by Cycle - detailed - Provides a detailed report for each cycle.
sortby	string	Specifies how to sort the values in the report. The acceptable values for this argument are the following: • total power - Sorts based on the power values. • spurious power - Sorts based on the spurious power. • functional power - Sorts based on the functional values. • spurious transitions - Sorts based on the spurious transitions. • functional transitions - Sorts based on the functional transitions.
sortorder	string	Specifies the sort order of the values in the report. This could be descending or ascending.
max_net	integer	Specifies the maximum number of nets to report. In a net summary or net details report, this argument limits the total number of entries. In a cycle details report, this argument limits the number of nets reported for each cycle.
max_cycle	integer	Specifies the maximum number of cycles to report. In a cycle summary or cycle details report, this argument limits the total number of entries. In a net details report, this argument limits the number of cycles reported for each net.
clock_settings	string	Specifies the settings for the clock. The format is ">clock name<:>active edge {value}<". The acceptable values for this argument are the following: • rising - Sets the clock to a rising active edge. • falling - Sets the clock to a falling active edge. • both - Sets the clock to both rising and falling active edge. • not_active - Does not use the signal as a clock.
glitch_filtering	string	Specifies whether to use glitch filtering. The acceptable values for this argument are the following: • TRUE, true or 1 - Glitch filtering is on. • FALSE, false or 0 - Enables automatic glitch filtering. This option will ignore any value specified in -glitch_threshold. • auto - Glitch filtering is off.
glitch_threshold	integer	This option is only available when `-glitch_filtering` is set to true. Specifies the glitch filtering value(in ps).

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power_activity_map (continued)		
Parameter	**Type**	**Description**
auto_construct_clock_domain	boolean	Specifies whether to automatically construct the clock domain. The acceptable values for this argument are the following: • TRUE, true or 1 - Automatically constructs the clock domain. • FALSE, false or 0 - Does not automatically construct the clock domain.
clock_period	decimal	Use this option to specify a virtual clock period (in ps). This should be used if `-auto_construct_clock_domain` is set to false.
clock_offset	decimal	Use this option to specify the time of the first active edge of the virtual clock (in ps). This should be used if `-auto_construct_clock_domain` is set to false.
opmode	string	Use this option to specify the mode from which the operating conditions are extracted to generate the report. • Active - The operating mode is set to Active. • Flash*Freeze - The operating mode is set to Flash*Freeze.
filename	string	Specifies the name or path of the file to be exported. This argument is mandatory. Default created under designer/<root_name> dircetory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'vcd_file' is missing.
None	Required parameter 'report_query' is missing.
None	VCD: Incorrect VCD file. "enddefinitions" not found.
None	VCD: File ./power.vcd doesn't exist: Failed to generate power report.
None	style: Invalid argument value: 'value' (expecting Text or CSV).
None	partial_parse: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	start_time: Invalid argument value: 'value' (expecting decimal value).
None	end_time: Invalid argument value: 'value' (expecting decimal value).
None	auto_detect_top_level_name: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	report_type: Invalid argument value: 'value' (expecting activity and power, activity or power).
None	report_query: Invalid argument value: 'value' (expecting Report by Cycle - summary, Report by Net - summary, Report by Cycle - detailed or Report by Net - detailed).
None	sortby: Invalid argument value: 'value' (expecting functional transitions, total power, spurious transitions, functional power or spurious power).
None	sortorder: Invalid argument value: 'value' (expecting ascending or descending).
None	max_net: Invalid argument value: 'value' (expecting integer value).
None	max_cycle: Invalid argument value: 'value' (expecting integer value).
None	glitch_filtering: Invalid argument value: 'value' (expecting false, auto or true).
None	glitch_threshold: Invalid argument value: 'value' (expecting integer value).
None	auto_construct_clock_domain: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	clock_period: Invalid argument value: 'value' (expecting decimal value).
None	clock_offset: Invalid argument value: 'value' (expecting decimal value).

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

This example generates an activity and hazards power report named "report_power_activity_map.txt":

```
smartpower_report_power_activity_map \
                -partial_parse "TRUE" \
                -start_time "1.0" \
                -end_time "3.0" \
                -auto_detect_top_level_name "TRUE" \
                -top_level_name "mytopmodule" \
                -glitch_filtering "auto" \
                -style "Text" \
                -opmode "Active" \
                -vcd_file power.vcd \
                -report_query "Report by Net - summary" \
                -clock_period "2000.0" \
                -clock_offset "10.0" \
                "report_power_activity_map_partial.txt"
```

See Also

- smartpower_report_power
- smartpower_report_power_peak_analyzer

9.36. smartpower_report_power_history (Ask a Question)

Description

Enter description here

```
smartpower_report_power_history
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

Enter example description here

```
smartpower_report_power_history
```

9.37. smartpower_report_power_peak_analyzer (Ask a Question)

Description

This Tcl command creates a cycle accurate power report, which reports a power waveform with one power value per clock period or half-period instead of an average power for the whole simulation.

```
smartpower_report_power_peak_analyzer \
                    -vcd_file "VCD file" \
                    [-style "Text | CSV"] \
                    [-partial_parse "TRUE | FALSE"] \
                    [-start_time "decimal value"] \
                    [-end_time "decimal value"] \
                    [-auto_detect_top_level_name "TRUE | FALSE"] \
                    [-top_level_name "top level name"] \
                    [-glitch_filtering "false | auto | true"] \
                    [-glitch_threshold "integer value"] \
                    [-opmode "Active | Static"] \
                    [-auto_detect_sampling_period "TRUE | FALSE"] \
                    [-sampling_clock "clk"] \
                    [-sampling_rate_per_period "TRUE | FALSE"] \
                    [-sampling_offset "decimal value"] \
                    [-sampling_period "decimal value"] \
                    [-use_only_local_extrema "TRUE | FALSE"] \
                    [-use_power_threshold "TRUE | FALSE"] \
                    [-power_threshold "decimal value"] \
                    [-file "file"]
```

Arguments

Parameter	Type	Description
vcd_file	string	Specifies the path to the *.vcd file that you want to import.
style	string	Specifies the format in which the report will be exported. The acceptable values for this argument are the following: • Text - The report will be exported as Text file. • CSV(by default) - The report will be exported as CSV file.
partial_parse	boolean	Specifies whether to partially parse the *.vcd file. The acceptable values for this argument are the following: • TRUE, true or 1 - Partially parses the *.vcd file. • FALSE, false or 0 - Does not partially parse the *.vcd file.
start_time	decimal	This option is available only if `-partially_parse` is set to true. Specifies the start time (in ns) to partially parse the *.vcd file.
end_time	decimal	This option is available only if `-partially_parse` is set to true. Specifies the start time (in ns) to partially parse the *.vcd file.
auto_detect_top_level_name	boolean	Specifies whether to automatically detect the top-level name. The acceptable values for this argument are the following: • TRUE, true or 1 - Automatically detects the top-level name. • FALSE, false or 0 - Does not automatically detect the top-level name.
top_level_name	string	Specifies the top-level name.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power_peak_analyzer (continued)

Parameter	Type	Description
glitch_filtering	string	Specifies whether to use glitch filtering. The acceptable values for this argument are the following: • TRUE, true or 1 - Glitch filtering is on. • FALSE, false or 0 - Enables automatic glitch filtering. This option will ignore any value specified in -glitch_threshold. • auto - Glitch filtering is off.
glitch_threshold	integer	This option is only available when `-glitch_filtering` is set to true. Specifies the glitch filtering value(in ps).
power_summary	boolean	Specifies whether to include the power summary, which shows the static and dynamic values in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the power summary in the report. • FALSE, false or 0 - Does not include the power summary in the report.
auto_detect_sampling_period	boolean	Specifies whether to automatically detect the sampling period. The acceptable values for this argument are the following: • TRUE, true or 1 - Automatically detects the sampling period. • FALSE, false or 0 - Does not automatically detect the sampling period.
sampling_clock	string	Specifies the sampling clock.
sampling_rate_per_period	boolean	Specifies whether to set the sampling rate per period. The acceptable values for this argument are the following: • TRUE, true or 1 - Specifies the sampling rate per period. • FALSE, false or 0 - Specifies the sampling rate per half period.
sampling_offset	decimal	Specifies the offset used to calculate the sampling offset (in ps).
sampling_period	decimal	Specifies the offset used to calculate the sampling period (in ps).
use_only_local_extrema	boolean	Specifies whether to limit the history size by keeping only local extrema. The acceptable values for this argument are the following: • TRUE, true or 1 - Limits the history size by keeping only local extrema. • FALSE, false or 0 - Does not limit the history size by keeping only local extrema.
use_power_threshold	boolean	Specifies whether to limit the history size by setting a power threshold. The acceptable values for this argument are the following: • TRUE, true or 1 - Limits the history size by setting a power threshold. • FALSE, false or 0 - Does not limit the history size by setting a power threshold.
power_threshold	decimal	Sets the power threshold value.
opmode	string	Use this option to specify the mode from which the operating conditions are extracted to generate the report. • Active - The operating mode is set to Active. • Flash*Freeze - The operating mode is set to Flash*Freeze.
filename	string	Specifies the name or path of the file to be exported. This argument is mandatory. Default created under designer/<root_name> dircetory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'vcd_file' is missing.
None	VCD: Incorrect VCD file. "enddefinitions" not found.
None	VCD: File ./power.vcd doesn't exist: Failed to generate power report.
None	style: Invalid argument value: 'value' (expecting Text or CSV).
None	partial_parse: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	start_time: Invalid argument value: 'value' (expecting decimal value).
None	end_time: Invalid argument value: 'value' (expecting decimal value).
None	auto_detect_top_level_name: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	glitch_filtering: Invalid argument value: 'value' (expecting false, auto or true).
None	glitch_threshold: Invalid argument value: 'value' (expecting integer value).
None	auto_detect_sampling_period: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	sampling_clock: Invalid argument value: 'value' (expecting "sampling clock pin name").
None	sampling_rate_per_period: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	sampling_offset: Invalid argument value: 'value' (expecting decimal value).
None	sampling_period: Invalid argument value: 'value' (expecting decimal value).
None	use_only_local_extrema: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	use_power_threshold: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	power_threshold: Invalid argument value: 'value' (expecting decimal value).
None	opmode: Invalid argument value: 'value' (expecting Active, Static or Flash*Freeze).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_report_power_peak_analyzer [-partial_parse "TRUE | FALSE"] \ [-start_time "decimal value"] \ [-end_time "decimal value"] \ [-auto_detect_top_level_name "TRUE | FALSE"] \ [-top_level_name "top level name"] \ [-glitch_filtering "false | auto | true"] \ [-glitch_threshold "integer value"] \ -vcd_file "VCD file" \ [-style "Text | CSV"] \ [-opmode "Active | Static"] \ [-auto_detect_sampling_period "TRUE | FALSE"] \ [-sampling_clock "clk"] \ [-sampling_rate_per_period "TRUE | FALSE"] \ [-sampling_offset "decimal value"] \ [-sampling_period "decimal value"] \ [-use_only_local_extrema "TRUE | FALSE"] \ [-use_power_threshold "TRUE | FALSE"] \ [-power_threshold "decimal value"] \ [-file "file"]'

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example generates a cycle accurate power report named "report_power_cycle_based.txt":

```
smartpower_report_power_peak_analyzer \
            -partial_parse "TRUE" \
                    -vcd_file "power.vcd" \
                    -style "Text" \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                    -start_time "0.05" \
                    -end_time "1.00" \
                    -auto_detect_top_level_name "TRUE" \
                    -glitch_filtering "FALSE" \
                    -glitch_threshold "100" \
                    -auto_detect_sampling_period "TRUE" \
                    -sampling_clock "clk" \
                    -sampling_rate_per_period "TRUE" \
                    -sampling_offset "0.00" \
                    -sampling_period "10000.00" \
                    -use_only_local_extrema "TRUE" \
                    -use_power_threshold "TRUE" \
                    -power_threshold "0.00" \
                    -opmode "Active" \
                    "report_power_cycle_based.txt"
```

See Also

- smartpower_report_power
- smartpower_report_power_activity_map

9.38. smartpower_report_power_sequencer (Ask a Question)

Description

This Tcl command creates a scenario power report for a previously defined scenario. It includes information about the global device and SmartPower preferences selection, and the average power consumption and the excepted battery life for this sequence.

Notes:

- Flash*Freeze is available only for certain families and devices(RTG4, SmartFusion 2, and IGLOO 2).
- Worst and Best are available only for certain families and devices.

```
smartpower_report_power_sequencer \
                  [-powerunit "W | mW | uW"] \
                  [-frequnit "Hz | KHz | MHz"] \
                  [-opcond "Best | Typical | Worst"] \
                  [-toggle "TRUE | FALSE"] \
                  [-scenario ""] \
                  [-style "Text | CSV | XML"] \
                  [-battery_life "TRUE | FALSE"] \
                  [-battery_capacity "decimal value"] \
                  [-rail_breakdown "TRUE | FALSE"] \
                  [-type_breakdown "TRUE | FALSE"] \
                  [-mode_breakdown "TRUE | FALSE"] \
                  [-opcond_summary "TRUE | FALSE"] \
                  {filename}
```

Arguments

Parameter	Type	Description
powerunit	string	Specifies the unit in which power is set. The acceptable values for this argument are the following: • W - The power unit is set to watts. • mW - The power unit is set to milliwatts. • uW - The power unit is set to microwatts.
frequnit	string	Specifies the unit in which frequency is set. The acceptable values for this argument are the following: • Hz - The frequency unit is set to hertz. • kHz - The frequency unit is set to kilohertz. • MHz - The frequency unit is set to megahertz.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_report_power_sequencer (continued)		
Parameter	**Type**	**Description**
opcond	string	Specifies the operating condition. The following acceptable values for this argument are the following: • worst - The operating condition is set to worst case. • typical - The operating condition is set to typical case. • best - The operating condition is set to best case.
toggle	boolean	Specifies the toggle. The acceptable values for this argument are thr following: • TRUE, true or 1 - The toggle is set to true. • FALSE, false or 1 - The toggle is set to false.
scenario	string	Specifies a scenario that the report is generated from.
style	string	Specifies the format in which the report will be exported. The acceptable values for this argument are the following: • Text - The report will be exported as Text file. • CSV(by default) - The report will be exported as CSV file. • XML - The report will be exported as XML file.
battery_life	boolean	Specifies whether to include the battery life summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the battery life summary in the report. • FALSE, false or 0 - Does not include the battery life summary in the report.
battery_capacity	decimal value	Specifies the battery capacity(decimal positive value) in A*H.
rail_breakdown	boolean	Specifies whether to include the breakdown by rail summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the breakdown by rail summary in the report. • FALSE, false or 0 - Does not include the breakdown by rail summary in the report. This is default value.
type_breakdown	boolean	Specifies whether to include the breakdown by type summary in the report. The acceptable values for this argument are thr following: • TRUE, true or 1- Includes the breakdown by type summary in the report. • FALSE, false or 0 - Does not include the breakdown by type summary in the report. This is the default value.
mode_breakdown	boolean	Specifies whether to include the breakdown by mode in the report. The acceptable values for this argument are thr following: • TRUE, true or 1 - Includes the breakdown by mode in the report. • FALSE, false or 0 - Does not include the breakdown by mode in the report. This is the default value.
opcond_summary	boolean	Specifies whether to include the operating conditions summary in the report. The acceptable values for this argument are the following: • TRUE, true or 1 - Includes the operating conditions summary in the report. • FALSE, false or 0 - Does not include the operating conditions summary in the report.
filename	string	Specifies the name or path of the file to be exported. This argument is mandatory. Default created under designer/<root_name> dircetory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	There are no scenarios in this design, create a scenario in SmartPower, and then generate the scenario report.
None	powerunit: Invalid argument value: 'value' (expecting W, mW or uW).
None	frequnit: Invalid argument value: 'value' (expecting Hz, KHz or MHz).
None	opcond: Invalid argument value: 'value' (expecting Best, Typical or Worst).
None	toggle: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	style: Invalid argument value: 'value' (expecting Text, CSV or XML).
None	battery_life: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	battery_capacity: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'battery_capacity' must be a positive decimal value.
None	rail_breakdown: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	type_breakdown: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	mode_breakdown: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	opcond_summary: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	—
None	—

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example generates a scenario power report named report.txt for "my_scenario":

```
smartpower_report_power_sequencer -scenario my_scenario \
                                  -rail_breakdown true \
                                  -type_breakdown true \
                                  -mode_breakdown true \
                                  -style text \
                                  -battery_capacity 10 \
                                  report.txt
```

See Also

- smartpower_report_power

© 2025 Microchip Technology Inc. and its subsidiaries

9.39. smartpower_restore (Ask a Question)

Description

This Tcl command restores all power information previously committed in SmartPower.

```
smartpower_restore
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_restore'.

Supported Families

Supported Families	Supported Versions
PolarFire®	v12.4+
SmartFusion® 2	v12.4+
RTG4™	v12.4+
IGLOO® 2	v12.4+
PolarFire SoC	v12.6+

Example

This example restores all power information previously committed in SmartPower.

```
smartpower_restore
```

See Also

- smartpower_commit

9.40. smartpower_set_cooling (Ask a Question)

Description

This Tcl command sets the cooling style to one of the predefined types or custom value.

Note: To compute the junction temperature, set the following three commands:"smartpower_set_thermalmode, smartpower_set_tambient and smartpower_set_cooling. The junction temperature will be updated when an output command is executed, such as report(Power). Parameter 'teta' is ignored when style is set to custom.

```
smartpower_set_cooling -style "case_cooling | still_air | 1.0_mps | 2.5_mps | custom" [-teta
"positive decimal value"]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
style	string	Specifies the cooling style to custom value or to one of the predefined types with a default thermal resistance value. The acceptable values for this argument are the following: • 1.0_mps - Predefined cooling style. • 2.5_mps - Predefined cooling style. • case_cooling - Predefined cooling style. • still_air - Predefined cooling style. • custom - Cooling style defined by user input.
teta	decimal	Specifies the thermal resistance in "C/W". This argument is only available when style is set to Custom.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'style' is missing.
None	Parameter 'style' has illegal value.
None	style: Invalid argument value: 'value' (expecting case_cooling, still_air, 1.0_mps, 2.5_mps or custom).
None	teta: Invalid argument value: 'decimal value' (expecting decimal value).
None	Parameter 'teta' must be a positive decimal value.
None	Parameter 'teta' has illegal value.
None	Parameter 'teta' is required when style is set to custom.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_cooling -style "case_cooling | still_air | 1.0_mps | 2.5_mps | custom" [-teta "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the cooling style to still air:

```
smartpower_set_cooling -style {still_air}
```

See Also

- smartpower_set_thermalmode
- smartpower_set_tambient

9.41. smartpower_set_default_io_enable_rate (Ask a Question)

Description

Enter description here

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Command 'smartpower_set_default_io_enable_rate' is deprecated, but it is still executable. Use "smartpower_change_setofpin_statistics" instead.

```
smartpower_set_default_io_enable_rate -pin_enable_rate "decimal value"
```

Arguments

Parameter	Type	Description
pin_enable_rate	decimal	

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_enable_rate' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_default_io_enable_rate -pin_enable_rate "decimal value"'.
None	Parameter 'pin_enable_rate' must be less than or equal to 100.000.
None	Parameter 'pin_enable_rate' must be a positive decimal value.
None	pin_enable_rate: Invalid argument value: 'value' (expecting decimal value).

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

Example

Enter example description here

```
smartpower_set_default_io_enable_rate -pin_enable_rate "4"
```

9.42. smartpower_set_mode_for_analysis (Ask a Question)

Description

This Tcl command sets the mode for cycle-accurate power analysis.

```
smartpower_set_mode_for_analysis -mode {value}
```

Arguments

Parameter	Type	Description
mode	string	Specifies the mode for cycle-accurate power analysis. This parameter is mandatory. The acceptable values for this argument are the following: • Active - The operating mode is set to Active. • Static - The operating mode is set to Static. • Flash*Freeze - The operating mode is set to Flash*Freeze(RTG4™, SmartFusion® 2, IGLOO® 2).

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'mode' is missing.
None	Parameter 'mode' has illegal value.
None	mode: Invalid argument value: 'mode_name' (expecting Active, Static or Flash*Freeze).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_mode_for_analysis -mode "Active | Static | Flash*Freeze"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example sets the mode for analysis to active:

```
smartpower_set_mode_for_analysis -mode {active}
```

9.43. smartpower_set_mode_for_pdpr (Ask a Question)

Description

This Tcl command sets the operating mode used by the Power Driven Place and Route (PDPR) tool during power optimization.

```
smartpower_set_mode_for_pdpr -opmode {operating mode name}
```

Arguments

Parameter	Type	Description
opmode	string	Sets the operating mode for your power driven place and route. Value must be a valid operating mode. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'opmode' is missing.
None	opmode: Invalid argument value: 'mode_name' (expecting Active, Flash*Freeze or <CustomCreatedModeName>).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_mode_for_pdpr -opmode "Active | Flash*Freeze".
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_mode_for_pdpr -opmode "Active".

Supported Families

PolarFire®
SmartFusion® 2
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

IGLOO® 2

Example

This example creates custom mode "MyCustomMode" based on "Active" mode and sets the "MyCustomMode" mode as the operating mode for Power Driven Place and Route(PDPR):

```
smartpower_add_new_custom_mode \
            -name "MyCustomMode" \
                    -description "for PDPR" \
            -base_mode "Active"
smartpower_set_mode_for_pdpr -opmode "MyCustomMode"
```

See Also

- smartpower_add_new_custom_mode
- smartpower_export_mpe_report

9.44. smartpower_set_operating_conditions (Ask a Question)

Description

This Tcl command sets the operating conditions used in SmartPower.

```
smartpower_set_operating_conditions -airflow "still_air "1.0_mps | 2.5_mps | custom" \
                                    -heatsink "None | custom | 10mm_Low_Profile |
15mm_Medium_Profile | 20mm_High_Profile" \
                                    -boardmodel "None_Conservative | JEDEC_2s2p" \
                                    [-teta_ja {decimal value}] \
                                    [-teta_sa {decimal value}]
```

Arguments

Parameter	Type	Description
airflow	string	Specifies the value for the still air operating condition. The acceptable values for this argument are the following: • 1.0_mps - Sets the operating conditions to best. • 2.5_mps - Sets the operating conditions to typical. • custom - Sets the operating conditions to worst. • still_air.
heatsink	string	Specifies the value of the operating condition. The acceptable values for this argument are the following: • none - No heat sink. • custom - Sets a custom heat sink size. • 10mm_Low_Profile - 10 mm heat sink. • 15mm_Low_Profile - 15 mm heat sink. • 20mm_High_Profile - 20 mm heat sink.
boardmodel	string	Specifies your board model. The acceptable values for this argument are the following: • None_Conservative - No board model, conservative routing. • JEDEC_2s2p - JEDEC 2s2p board model.
teta_ja	decimal	This is an optional parameter, that sets your teta ja value. It must be a positive decimal.
teta_sa	decimal	This is an optional parameter, that sets your teta sa value. It must be a positive decimal.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'boardmodel' has illegal value.
None	Required parameter 'boardmodel' is missing.
None	Parameter 'heatsink' has illegal value.
None	Required parameter 'heatsink' is missing.
None	Parameter 'airflow' has illegal value.
None	Required parameter 'airflow' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_operating_conditions -airflow "still_air | 1.0_mps | 2.5_mps | custom" -heatsink "None | custom | 10mm_Low_Profile | 15mm_Medium_Profile | 20mm_High_Profile" -boardmodel "None_Conservative | JEDEC_2s2p" [-teta_ja "decimal value"] [-teta_sa "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the operating conditions to best:

```
smartpower_set_operating_conditions -airflow "1.0_mps" \
                        -heatsink "20mm_High_Profile" \
                        -boardmodel "JEDEC_2s2p"
```

9.45. smartpower_set_operating_conditions (Ask a Question)

Description

This Tcl command sets the operating conditions used in SmartPower.

```
smartpower_set_operating_conditions -airflow "still_air | 1.0_mps | 2.5_mps | custom" \
                                    -heatsink {None | custom | 10mm_Low_Profile |
15mm_Medium_Profile | 20mm_High_Profile} \
                                    -boardmodel {None_Conservative | JEDEC_2s2p} \
                                    [-teta_ja {decimal value}] \
                                    [-teta_sa {decimal value}]
```

Arguments

Parameter	Type	Description
still_air	string	Specifies the value for the still air operating condition. The acceptable values for this argument are the following: • 1.0_mps - Sets the operating conditions to best. • 2.5_mps - Sets the operating conditions to typical. • custom - Sets the operating conditions to worst.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_set_operating_conditions (continued)

Parameter	Type	Description
heatsink	string	Specifies the value of the operating condition. The following table shows the acceptable values for this argument: • none - No heat sink. • custom - Sets a custom heat sink size. • 10mm_Low_Profile - 10 mm heat sink. • 15mm_Low_Profile - 15 mm heat sink. • 20mm_High_Profile - 20 mm heat sink.
boardmodel	string	Specifies your board model. The following table shows the acceptable values for this argument: • None_Conservative - No board model, conservative routing. • JEDEC_2s2p - JEDEC 2s2p board model.
teta_ja	decimal	This is an optional parameter, that sets your teta ja value. It must be a positive decimal.
teta_sa	decimal	This is an optional parameter, that sets your teta sa value. It must be a positive decimal.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'airflow' is missing.
None	Parameter 'airflow' has illegal value.
None	airflow: Invalid argument value: 'value' (expecting still_air, 1.0_mps, 2.5_mps or custom).
None	Parameter 'heatsink' has illegal value.
None	heatsink: Invalid argument value: 'value' (expecting None, custom, 10mm_Low_Profile, 15mm_Medium_Profile or 20mm_High_Profile).
None	Required parameter 'heatsink' is missing.
None	Parameter 'boardmodel' has illegal value.
None	boardmodel: Invalid argument value: 'value' (expecting None_Conservative or JEDEC_2s2p).
None	Required parameter 'boardmodel' is missing.
None	teta_ja: Invalid argument value: 'value' (expecting decimal value).
None	teta_sa: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_operating_conditions -airflow "still_air | 1.0_mps | 2.5_mps | custom" -heatsink "None | custom | 10mm_Low_Profile | 15mm_Medium_Profile | 20mm_High_Profile" -boardmodel "None_Conservative | JEDEC_2s2p" [-teta_ja "decimal value"] [-teta_sa "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example sets the operating conditions to best:

```
smartpower_set_operating_conditions -airflow "1.0_mps" \
                        -heatsink "20mm_High_Profile" \
                        -boardmodel "JEDEC_2s2p"
```

9.46. smartpower_set_pin_frequency (Ask a Question)

Description

This Tcl command sets the frequency of a pin in megahertz (MHz). If you do not use this command, each pin will have default frequency based on its domain.

Note: The pin_name must be the name of a pin that already exists in the design and already belongs to a domain. When specifying the unit, a space must be between the frequency value and the unit. Execute the "smartpower_commit" Tcl command for preserving any changes in SmartPower.

```
smartpower_set_pin_frequency -pin_name {pin_name} -pin_freq {value}
```

Arguments

Parameter	Type	Description
pin_freq	string	Specifies the value of the frequency in Hz, KHz and MHz, which can be any positive decimal number.
pin_name	string	Specifies the name of the pin for which the frequency will be set.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_name' is missing.
None	Parameter 'pin_name' is missing or has invalid value.
None	Required parameter 'pin_freq' is missing.
None	Failed to set "pin name" pin frequency to 2.60 Hz.
None	pin_freq: Parameter format incorrect (expecting a positive decimal value, optionally followed by a unit (possible values are Hz, KHz or MHz)).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_pin_frequency [-pin_name "name of pin"]+ -pin_freq "decimal value [unit { Hz | KHz | MHz }]"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the frequency of the pin named "d_in" to 100 MHz:

```
smartpower_set_pin_frequency -pin_name {d_in} -pin_freq {100}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- smartpower_remove_pin_frequency

9.47. smartpower_set_pin_probability (Ask a Question)

Description

This tcl command sets the probability value of a pin driving an enable pin. For I/Os, if you do not use this command, the probability of the IOEnableSet is used. For memories, if you do not use this command, the probability of the MemoriesEnableSet is used.

```
smartpower_set_pin_probability -pin_name "name of pin" -pin_proba "decimal value"
```

Arguments

Parameter	Type	Description
pin_name	string	Specifies the name of a pin for which the probability will be set. This pin must be the direct driver of an enable pin. There may be multiple `-pin_name` arguments. This is mandatory.
pin_proba	decimal value	Specifies the value of the pin probability as a percentage(%), which can be any positive decimal between 0 and 100 inclusive.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'pin_name' is missing.
None	Failed to set "value" pin frequency to 11.00.
None	pin_proba: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'pin_proba' must be less than or equal to 100.000.
None	Parameter 'pin_proba' must be a positive decimal value.
None	ERROR in setSystemDomainIdOfPin. The pin indoes not exist.
None	'smartpower_set_pin_probability [-pin_name "name of pin"]+ -pin_proba "decimal value"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example sets the probability of the pin driving the enable pin of a bidirectional I/O:

```
smartpower_set_pin_probability -pin_name {mybibuf/U0/U1:EOUT} -pin_proba {50.4}
```

The following example sets the probability of the "in" and "out" pins:

```
smartpower_set_pin_probability -pin_name {in out} -pin_proba {50.4}
```

See Also

- smartpower_remove_pin_probability

© 2025 Microchip Technology Inc. and its subsidiaries

9.48. smartpower_set_preference (Ask a Question)

Description

This Tcl command sets the following preferences: power unit, frequency unit, operating mode, operating conditions, and toggle. These preferences can also be set from the preferences dialog box.

Note: Running the script with invalid argument it will trigger a warning message: Ignoring invalid argument 'argname'.

```
smartpower_set_preference [-powerunit "W | mW | uW"] \
                          [-frequnit "Hz | KHz | MHz"] \
                          [-opcond "Best | Typical | Worst"] \
                          [-opmode "Active | Static"] \
                          [-toggle "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
powerunit	string	Specifies the unit in which power is set. The acceptable values for this argument are the following: • W - The power unit is set to watts. • mW - The power unit is set to milliwatts. • uW - The power unit is set to microwatts.
frequnit	string	Specifies the unit in which frequency is set. The acceptable values for this argument are the following: • H - The frequency unit is set to hertz. • kHz - The frequency unit is set to kilohertz. • MHz - The frequency unit is set to megahertz.
opmode	string	Specifies the operating mode. The acceptable values for this argument are the following: • active - The operating mode is set to active. • static - The operating mode is set to static. • Flash*Freeze - The operating mode is set to Flash*Freeze (RTG4, SmartFusion 2 and IGLOO 2).
opcond	string	Specifies the operating condition. The acceptable values for this argument are the following: • worst - The operating condition is set to worst-case. • typical - The operating condition is set to typical case. • best - The operating condition is set to best case.
toggle	boolean	Specifies the toggle. The acceptable values for this argument are the following: • TRUE, true or 1 - The toggle is set to true. • FALSE, false or 0 - The toggle is set to false.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'powerunit' has illegal value.
None	powerunit: Invalid argument value: 'value' (expecting W, mW or uW).
None	frequnit: Invalid argument value: 'value' (expecting Hz, KHz or MHz).

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_set_preference (continued)	
Error Code	**Description**
None	opmode: Invalid argument value: 'value' (expecting Active, Static or Flash*Freeze).
None	opcond: Invalid argument value: 'value' (expecting Best, Typical or Worst).
None	toggle: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2

Example

This example sets the frequency of the power unit to "watts", the frequency unit to "Hz", the operating mode to "active", the operating condition to "typical", and the toggle to "true":

```
smartpower_set_preference -powerunit {w} \
                          -frequnit {hz} \
                          -opmode {active} \
                          -opcond{typical} \
                          -toggle {true}
```

9.49. smartpower_set_process (Ask a Question)

Description

This Tcl command sets the process used in SmartPower to one of the pre-defined types.

```
smartpower_set_process -process {value}
```

Arguments

Parameter	Type	Description
process	string	Specifies the value of the operating condition. The acceptable values for this argument are the following: • Typical - Sets the process for SmartPower to typical. • Maximum - Sets the process for SmartPower to maximum.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'process' is missing.
None	Parameter 'process' has illegal value.
None	process: Invalid argument value:""(expecting Typical or Maximum).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_process -process "Typical | Maximum"'

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC

Example
This example sets the operating conditions to typical:

```
smartpower_set_process -process {Typical}
```

9.50. smartpower_set_scenario_for_analysis (Ask a Question)

Description
This Tcl command sets the scenario for cycle-accurate power analysis.

```
smartpower_set_scenario_for_analysis -scenario {value}
```

Arguments

Parameter	Type	Description
scenario	string	Specifies the mode for cycle-accurate power analysis.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'scenario' is missing.
None	scenario: Invalid argument value: 'ScenarioName' (expecting existing_scenario_name).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_scenario_for_analysis -scenario "MyScenario."

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example
The following example sets the scenario for analysis to "my_scenario" custom scenario:

```
smartpower_set_scenario_for_analysis -scenario {my_scenario}
```

See Also
- smartpower_add_new_scenario
- smartpower_remove_scenario

9.51. smartpower_set_tambient (Ask a Question)

Description
Enter description here

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Command 'smartpower_set_tambient' is deprecated, but it is still executable. Use "smartpower_temperature_opcond_set_design_wide" instead.

```
smartpower_set_tambient -temp "decimal value"
```

Arguments

Parameter	Type	Description
temp	decimal	—

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'temp' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_tambient -temp "decimal value"'.
None	Parameter 'temp' must be less than or equal to 125.000.
None	temp: Invalid argument value: 'value' (expecting decimal value).

Supported Families

PolarFire®

SmartFusion®2

RTG4™

IGLOO® 2

Example

Enter example description here

```
smartpower_set_tambient -temp "5"
```

9.52. smartpower_set_temperature_opcond (Ask a Question)

Description

This tcl command sets the temperature in the operating conditions to one of the predefined types.

```
smartpower_set_temperature_opcond -use {value}
```

Arguments

Parameter	Type	Description
use	string	Specifies the temperature in the operating conditions. The acceptable values for this argument are the following: • oprange - Sets the temperature in the operating conditions as specified in your Project Settings. • design - Sets the temperature in the operating conditions as specified in the SmartPower design-wide operating range. Applies to SmartPower only. • mode - Sets the temperature in the operating conditions as specified in the SmartPower mode-specific operating range. Applies to SmartPower only.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'use' has illegal value.
None	Required parameter 'use' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_temperature_opcond -use "oprange | design | mode"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the temperature in the operating conditions as specified in the custom mode-settings:

```
smartpower_set_temperature_opcond -use {mode}
```

See Also

- smartpower_temperature_opcond_set_design_wide
- smartpower_temperature_opcond_set_mode_specific

9.53. smartpower_set_thermalmode (Ask a Question)

Description

This Tcl command sets the mode of computing junction temperature.

Note: To compute the junction temperature, set the smartpower_set_thermalmode, smartpower_set_tambientand smartpower_set_cooling commands. The junction temperature will be updated when an output command is executed, such as report(Power).

```
smartpower_set_thermalmode [-mode {value}]
```

Arguments

Parameter	Type	Description
mode	string	Specifies the mode in which the junction temperature is computed. The acceptable values for this argument are the following: • ambient - The junction temperature will be iteratively computed with total static power. • opcond - The junction temperature will be given as one of the operating condition range valuesspecified in the device selection.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'mode' is missing.
None	Parameter 'mode' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_thermalmode -mode "opcond | ambient"'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

The following example sets the computing of the junction temperature to ambient mode:

```
smartpower_set_thermalmode -mode {ambient}
```

9.54. smartpower_set_voltage_opcond (Ask a Question)

Description

This Tcl command sets the voltage in the operating conditions.

```
smartpower_set_voltage_opcond -voltage {value} -use {value}
```

Arguments

Parameter	Type	Description
voltage	string	Specifies the voltage supply in the operating conditions. The acceptable values for this argument are the following: • VDD - Sets the voltage operating conditions for VDD. • VDD18 - Sets the voltage operating conditions for VDD18. • VDDAUX - Sets the voltage operating conditions for VDDAUX. • VDDI 1.1 - Sets the voltage operating conditions for VDD 1.1. • VDDI 1.2 - Sets the voltage operating conditions for VDDI 1.2. • VDDI 1.35 - Sets the voltage operating conditions for VDDI 1.35. • VDDI 1.5 - Sets the voltage operating conditions for VDDI 1.5. • VDDI 1.8 - Sets the voltage operating conditions for VDDI 1.8. • VDDI 2.5 - Sets the voltage operating conditions for VDDI 2.5. • VDDI 3.3 - Sets the voltage operating conditions for VDDI 3.3. • VDD25 - Sets the voltage operating conditions for VDD25. • VDDA - Sets the voltage operating conditions for VDDA. • VDDA25 - Sets the voltage operating conditions for VDDA25. • VPP - Sets the voltage operating conditions for VPP.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_set_voltage_opcond (continued)		
Parameter	**Type**	**Description**
use	string	Specifies the voltage in the operating conditions for each voltage supply. The acceptable values for this argument are the following: • oprange - Sets the voltage in the operating conditions as specified in your Project Settings. • design - Sets the voltage in the operating conditions as specified in the SmartPower design-wideoperating range. Applies to SmartPower only. • mode - Sets the voltage in the operating conditions as specified in the SmartPower mode-specificoperating range. Applies to SmartPower only.

Return Type	Description
None	None

Error Codes

Error Code	Description
	Parameter 'use' has illegal value.
	Required parameter 'use' is missing.
	Parameter 'voltage' has illegal value.
	Required parameter 'voltage' is missing.
	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_set_voltage_opcond -voltage "VDD | VDD18 | VDDI 1.8 | VDD25" -use "oprange | design | mode"'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

This example sets the VCCA as specified in the SmartPower mode-specific settings:

```
smartpower_set_voltage_opcond -voltage {VDD} -use {mode}
```

See Also

- smartpower_voltage_opcond_set_mode_specific
- smartpower_voltage_opcond_set_design_wide

9.55. smartpower_temperature_opcond_set_design_wide (Ask a Question)

Description

This Tcl command sets the temperature for SmartPower design-wide operating conditions.

```
smartpower_temperature_opcond_set_design_wide \
                        [-best {value}] \
                                [-typical {value}] \
                                [-worst {value}] \
                                [-thermal_mode {value}]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
best	decimal	Specifies the best temperature (in degrees Celsius) used for design-wide operating conditions.
typical	decimal	Specifies the typical temperature (in degrees Celsius) used for design-wide operating conditions.
worst	decimal	Specifies the worst temperature (in degrees Celsius) used for design-wide operating conditions.
thermal_mode	string	Specifies the mode in which the junction temperature is computed. The acceptable values for this argument are the following: • ambient - The junction temperature will be iteratively computed with total static power. • opcond - The junction temperature will be given as one of the operating condition rangevalues specified in the device selection.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'thermal_mode' has illegal value.
None	Parameter 'worst' has illegal value.
None	Parameter 'typical' has illegal value.
None	Parameter 'best' has illegal value.
None	Invalid best voltage value (00) for VDD. VDD best voltage must be between 0.970 and 1.030.
None	Invalid typical voltage value (00) for VDD. VDD typical voltage must be between 0.970 and 1.030.
None	Invalid argument value: 'voltage_value' (expecting VDD, VDDAUX, VDDI 1.8, VDD25 or VDD18).
None	Invalid voltage sequence. The value for best cannot exceed typical and typical cannot exceed worst.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_temperature_opcond_set_design_wide [-thermal_mode "opcond | ambient"] [-best "decimal value"] [-typical "decimal value"] [-worst "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the temperature for design-wide operating conditions to "Best 20", "Typical 30", and "Worst 60":

```
smartpower_temperature_opcond_set_design_wide -best {20} -typical {30} -worst {60}
```

See Also

- smartpower_temperature_opcond_set_mode_specific

© 2025 Microchip Technology Inc. and its subsidiaries

- smartpower_set_temperature_opcond

9.56. smartpower_temperature_opcond_set_mode_specific (Ask a Question)

Description

This Tcl command sets the temperature for SmartPower mode-specific operating conditions.

```
smartpower_temperature_opcond_set_mode_specific \
                    -opmode "Active | Static" \
                    [-thermal_mode "opcond | ambient"] \
                    [-best "decimal value"] \
                    [-typical "decimal value"] \
                    [-worst "decimal value"]
```

Arguments

Parameter	Type	Description
opmode	string	Specifies the operating mode. The acceptable values for this argument are the following: • Active - The operating mode is set to Active. • Static - The operating mode is set to Static. • Flash*Freeze - The operating mode is set to Flash*Freeze(RTG4™, SmartFusion® 2 and IGLOO® 2).
thermal_mode	string	Specifies the mode in which the junction temperature is computed. This parameter is optional. The acceptable values for this argument are the following: • ambient - The junction temperature will be iteratively computed with total static power. • opcond - The junction temperature will be given as one of the operating condition rangevalues specified in the device selection.
best	decimal	Specifies the best temperature (in degrees Celsius) for the selected mode. This parameter is optional.
typical	decimal	Specifies the typical temperature (in degrees Celsius) for the selected mode. This parameter is optional.
worst	decimal	Specifies the worst temperature (in degrees Celsius) for the selected mode. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'worst' has illegal value.
None	Parameter 'typical' has illegal value.
None	typical: Invalid argument value: 'value' (expecting decimal value).
None	Parameter 'best' has illegal value.
None	best: Invalid argument value: 'best_value' (expecting decimal value).
None	Parameter 'thermal_mode' has illegal value.
None	thermal_mode: Invalid argument value: 'value' (expecting opcond or ambient).
None	Required parameter 'opmode' is missing.
None	Parameter 'opmode' has illegal value.
None	opmode: Invalid argument value: 'value' (expecting Active or Static).
None	Invalid temperature sequence for 'Active' mode. The value for best cannot exceed typical and typical cannot exceed worst.

© 2025 Microchip Technology Inc. and its subsidiaries

smartpower_temperature_opcond_set_mode_specific (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_temperature_opcond_set_mode_specific -opmode "Active | Static" [-thermal_mode "opcond | ambient"] [-best "decimal value"] [-typical "decimal value"] [-worst "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the temperature for mode-specific operating conditions for mode1:

```
smartpower_temperature_opcond_set_mode_specific \
                              -opmode {Active} \
                    -thermal_mode {ambient} \
                              -best {20} \
                              -typical {30} \
                              -worst {60}
```

See Also

- smartpower_temperature_opcond_set_design_wide
- smartpower_set_temperature_opcond

9.57. smartpower_voltage_opcond_set_design_wide (Ask a Question)

Description

This tcl command sets the voltage settings for SmartPower design-wide operating conditions.

```
smartpower_voltage_opcond_set_design_wide \
          -voltage "VDD | VDD18 | VDDI 1.8 | VDD25" \
          [-best "decimal value"] \
          [-typical "decimal value"] \
          [-worst "decimal value"]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
voltage	string	Specifies the voltage supply in the operating conditions. The acceptable values for this argument are the following: • VDD - Sets the voltage operating conditions for VDD. • VDDAUX - Sets the voltage operating conditions for VDDAUX. • VDD25 - Sets the voltage operating conditions for VDD25. • VDDI 2.5 - Sets the voltage operating conditions for VDDI 2.5. • VDD18 - Sets the voltage operating conditions for VDD18. • VPP - Sets the voltage operating conditions for VPP. • VCCA - Sets the voltage operating conditions for VCCA. • VCCI 3.3 - Sets the voltage operating conditions for VCCI 3.3. • VCCI 2.5 - Sets the voltage operating conditions for VCCI 2.5. • VCCI 1.8 - Sets the voltage operating conditions for VCCI 1.8. • VCCI 1.5 - Sets the voltage operating conditions for VCCI 1.5. • VCC33A - Sets the voltage operating conditions for VCC33A. • VCCDA - Sets the voltage operating conditions for VCCDA. • VPP - Sets the voltage operating conditions for VPP.
best	decimal	Specifies the best voltage used for design-wide operating conditions.
typical	decimal	Specifies the typical voltage used for design-wide operating conditions.
worst	decimal	Specifies the worst voltage used for design-wide operating conditions.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'worst' has illegal value.
None	Parameter 'typical' has illegal value.
None	Parameter 'best' has illegal value.
None	Invalid best voltage value (0.000) for VDDI 1.8. VDDI 1.8 best voltage must be between 1.710 and 1.890.
None	Parameter 'voltage' has illegal value.
None	Required parameter 'voltage' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_voltage_opcond_set_design_wide -voltage "VDD | VDD18 | VDDI 1.8 | VDD25" [-best "decimal value"] [-typical "decimal value"] [-worst "decimal value"]'

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example sets VDDI 1.8 for design-wide to best 1.72, typical 1.85 and worst 1.88:

```
smartpower_voltage_opcond_set_design_wide \
                              -voltage {VDDI 1.8} \
                              -best {1.72} \
                              -typical {1.85} \
                              -worst {1.88}
```

See Also

- smartpower_set_voltage_opcond
- smartpower_voltage_opcond_set_mode_specific

9.58. smartpower_voltage_opcond_set_mode_specific (Ask a Question)

Description

This tcl command sets the voltage settings for SmartPower mode-specific use operating conditions.

```
smartpower_voltage_opcond_set_mode_specific \
                  -opmode "Active | Static" \
                  -voltage "VDD | VDD18 | VDDI 1.8 | VDD25" \
                  [-best "decimal value"] \
                  [-typical "decimal value"] \
                  [-worst "decimal value"]
```

Arguments

Parameter	Type	Description
opmode	string	Specifies the mode from which the operating conditions are extracted to generate the report. • Active - The operating mode is set to Active. • Static - The operating mode is set to Static. • Flash*Freeze - The operating mode is set to Flash*Freeze(RTG4™, SmartFusion® 2 and IGLOO® 2).
voltage	string	Specifies the voltage in the operating conditions. The acceptable values for this argument are the following: • VDD - Sets the voltage operating conditions for VDD. • VDD18 - Sets the voltage operating conditions for VDD18. • VDDAUX - Sets the voltage operating conditions for VDDAUX. • VDDI 1.1 - Sets the voltage operating conditions for VDD 1.1. • VDDI 1.2 - Sets the voltage operating conditions for VDDI 1.2. • VDDI 1.35 - Sets the voltage operating conditions for VDDI 1.35. • VDDI 1.5 - Sets the voltage operating conditions for VDDI 1.5. • VDDI 1.8 - Sets the voltage operating conditions for VDDI 1.8. • VDDI 2.5 - Sets the voltage operating conditions for VDDI 2.5. • VDDI 3.3 - Sets the voltage operating conditions for VDDI 3.3. • VDD25 - Sets the voltage operating conditions for VDD25. • VDDA - Sets the voltage operating conditions for VDDA. • VDDA25 - Sets the voltage operating conditions for VDDA25.
best	decimal	Specifies the best voltage used for mode-specific operating conditions.
typical	decimal	Specifies the typical voltage used for mode-specific operating conditions.
worst	decimal	Specifies the worst voltage used for mode-specific operating conditions.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'worst' has illegal value.
None	Parameter 'typical' has illegal value.
None	Parameter 'best' has illegal value.
None	Parameter 'voltage' has illegal value.
None	Parameter 'opmode' has illegal value.
None	Required parameter 'opmode' is missing.
None	Required parameter 'voltage' is missing.
None	Invalid best voltage value (1.200) for VDD18. VDD18 best voltage must be between 1.710 and 1.890.
None	Invalid best voltage value (1.200) for VDDI 1.8. VDDI 1.8 best voltage must be between 1.710 and 1.890.
None	Invalid best voltage value (1.200) for VDDAUX. VDDAUX best voltage must be between 2.375 and 2.625.
None	Invalid best voltage value (1.200) for VDD25. VDD25 best voltage must be between 2.375 and 2.625.
None	Invalid best voltage value (1.111) for VDDI 2.5. VDDI 2.5 best voltage must be between 2.375 and 2.625.
None	Invalid best voltage value (1.200) for VDD. VDD best voltage must be between 0.970 and 1.030.
None	Invalid best voltage value (1.111) for VDD. VDD best voltage must be between 0.970 and 1.030.
None	Invalid best voltage value (1.111) for VDD18. VDD18 best voltage must be between 1.710 and 1.890.
None	Invalid worst voltage value (1.111) for VDD. VDD worst voltage must be between 0.970 and 1.030.
None	voltage: Invalid argument value: 'voltage_value' (expecting VDD, VDDI 2.5 or VPP).
None	voltage: Invalid argument value: 'voltage_value' (expecting VDD, VDDAUX, VDDI 1.8, VDD25 or VDD18).
None	voltage: Invalid argument value: 'voltage_value' (expecting VDD, VDD18, VDDI 1.8, VDDAUX or VDD25).
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartpower_voltage_opcond_set_mode_specific -opmode "Active | Static" -voltage "VDD | VDD18 | VDDI 1.8 | VDD25" [-best "decimal value"] [-typical "decimal value"] [-worst "decimal value"]'.

Supported Families

PolarFire®
SmartFusion® 2
RTG4™
IGLOO® 2
PolarFire SoC

Example

This example sets the voltage for the static mode and sets "best to 0.98", "typical to 1" and "worst to 1.01":

```
smartpower_voltage_opcond_set_mode_specific \
                        -opmode {active} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                    -voltage {VDD} \
                    -best {0.98} \
                    -typical {1} \
                    -worst {1.01}
```

See Also

- smartpower_set_voltage_opcond
- smartpower_voltage_opcond_set_design_wide

© 2025 Microchip Technology Inc. and its subsidiaries

10. Programming and Configuration Tcl Commands (Ask a Question)

10.1. configure_design_initialization_data (Ask a Question)

Description

This Tcl command sets the parameter values needed for generating initialization data.

```
configure_design_initialization_data \
    [-second_stage_start_address {sNVM address for the second initialization client}] \
    [-third_stage_uprom_start_address {uPROM address for the third initialization stage
client}] \
    [-third_stage_spi_start_address {SPI address for the third initialization stage client}] \
    [-third_stage_snvm_start_address {sNVM address for the third initialization stage
client}] \
    [-third_stage_spi_type {SPIFLASH_NO_BINDING_PLAINTEXT | SPIFLASH_BINDING_DEFAULT |
SPIFLASH_BINDING_UEK1 | SPIFLASH_BINDING_UEK2}] \
    [-third_stage_spi_clock_divider {1 | 2 | 4 | 6}] \
    [-init_timeout {int between 1 and 128 seconds}] \
    [-auto_calib_timeout {Auto Calibration timeout value in milliseconds}] \
    [-broadcast_RAMs {0 | 1}] \
    [-custom_cfg_file {Initialization file for custom configuration}]
```

Arguments

Parameter	Type	Description
second_stage_start_address	string	String parameter for the start address of the second stage initialization client. Specified as a 32-bit hexadecimal string. The first stage client is always placed in sNVM, so it must be a valid sNVM address aligned on a page boundary. There are 221 sNVM pages and each page is 256 bytes long, so the address will be between 0 and DC00. **Note:** Although the actual size of each page is 256 bytes, only 252 bytes are available to the user. The first stage initialization client is always added to SNVM at 0xDC00 (page 220). So the valid addresses for the second stage initialization client are 0x0 (page 0) to 0xDB00 (page 219).
third_stage_uprom_start_address	string	String parameter for the uPROM start address of the third stage initialization client. This parameter is optional. Specified as a 32-bit hexadecimal string and must be valid uPROM address aligned on a block boundary.
third_stage_spi_start_address	string	String parameter for the sNVM start address of the third stage initialization client. This parameter is optional. Specified as a 32-bit hexadecimal string and must be valid sNVM address.
third_stage_snvm_start_address	string	String parameter for the SPIFLASH start address of the third stage initialization client. This parameter is optional. Specified as a32-bit hexadecimal string and must be valid SPIFLASH address.
third_stage_spi_type	string	The value must be one of SPIFLASH_NO_BINDING_PLAINTEXT or SPIFLASH_BINDING_DEFAULT or SPIFLASH_BINDING_UEK1 or SPIFLASH_BINDING_UEK2. This parameter determines the valid value for parameter 'third_stage_start_address'.
third_stage_spi_clock_divider	integer	Specifies the clock frequency appropriate for the SPIFLASH memory on board. The value can be 1, 2, 4, or 6. The default value is 1 which is 80 MHz. The other values are 2-40 MHz, 4-20 MHz and 6-13.33 Mhz.
init_timeout	integer	Timeout value in seconds. Initialization is aborted if it does not complete before timeout expires. The value can be between 1 and 128. The default value is 128.
auto_calib_timeout	integer	The Auto Calibration Time Out value specifies the time out before which the IO Calibration instructions must be completed. The default value is 3000 milliseconds. This time out value is applicable only for MPF100T, MPF200T, MPF300T and MPF500T devices.

© 2025 Microchip Technology Inc. and its subsidiaries

configure_design_initialization_data (continued)		
Parameter	**Type**	**Description**
broadcast_RAMs	boolean	Specifies broadcast instructions to initialize RAM's to zero's. Value can be either 0 or 1. This parameter is optional.
custom_cfg_file	string	Specifies the initialization file for custom configuration. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'broadcast_RAMs' has illegal value.
None	Invalid value for parameter 'Auto Calibration Timeout'.
None	Parameter 'auto_calib_timeout' has illegal value.
None	init_timeout: Invalid argument value: '85,5' (expecting integer value).
None	Parameter 'init_timeout' must be greater than or equal to 1.
None	Parameter 'init_timeout' must be less than or equal to 128.
None	Parameter 'init_timeout' has illegal value.
None	third_stage_spi_clock_divider: Invalid argument value: 'value' (expecting 1, 2, 4 or 6).
None	Parameter 'third_stage_spi_clock_divider' has illegal value.
None	third_stage_spi_type: Invalid argument value: 'value' (expecting SPIFLASH_NO_BINDING_PLAINTEXT, SPIFLASH_BINDING_DEFAULT, SPIFLASH_BINDING_UEK1 or SPIFLASH_BINDING_UEK2).
None	Parameter 'third_stage_spi_type' has illegal value.
None	Invalid value for parameter 'SPI_third_stage_start_address'. Address must be a valid 32-bit hexadecimal string.
None	Parameter 'third_stage_spi_start_address' has illegal value.
None	Invalid value for parameter 'sNVM_third_stage_start_address'. Address must be a valid 32-bit hexadecimal string.
None	Parameter 'third_stage_snvm_start_address' has illegal value.
None	Invalid value for parameter 'uPROM_third_stage_start_address'. Address must be a valid 32-bit hexadecimal string.
None	Parameter 'third_stage_uprom_start_address' has illegal value.
None	Invalid value for parameter 'second_stage_start_address'. Address must be a valid 32-bit hexadecimal string.
None	Parameter 'second_stage_start_address' has illegal value.
None	Invalid custom configuration file: value.
None	Parameter 'custom_cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_design_initialization_data \ [-second_stage_start_address "sNVM address for the second initialization stage client"] \ [-third_stage_uprom_start_address "uPROM address for the third initialization stage client"] \ [-third_stage_snvm_start_address "sNVM address for the third initialization stage client"] \ [-third_stage_spi_start_address "SPI address for the third initialization stage client"] \ [-third_stage_spi_type "SPIFLASH_NO_BINDING_PLAINTEXT | SPIFLASH_BINDING_DEFAULT | SPIFLASH_BINDING_UEK1 | SPIFLASH_BINDING_UEK2"] \ [-third_stage_spi_clock_divider "1 | 2 | 4 | 6"] \ [-init_timeout "integer value"] \ [-auto_calib_timeout "Auto Calibration timeout value in seconds"] \ [-custom_cfg_file "Initialization file for custom configuration"] \ [-broadcast_RAMs "broadcast option to initialize RAMs to zeros"]'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC

Example

This example initializes data with sNVM client.

```
configure_design_initialization_data \
            -second_stage_start_address {0x0000aa00} \
            -third_stage_uprom_start_address {0x00000000} \
            -third_stage_snvm_start_address {0x0000aa00} \
            -third_stage_spi_start_address {0x00000400} \
            -third_stage_spi_type {SPIFLASH_NO_BINDING_PLAINTEXT} \
            -third_stage_spi_clock_divider {4} \
            -init_timeout 85 \
            -auto_calib_timeout {1400} \
            -broadcast_RAMs {0}
```

This example initializes data with uPROM client.

```
configure_design_initialization_data \
            -second_stage_start_address {0x00000000} \
            -third_stage_uprom_start_address {0xffffffee2} \
            -third_stage_snvm_start_address {0x00000000} \
            -third_stage_spi_start_address {0x00000400} \
            -third_stage_spi_type {SPIFLASH_NO_BINDING_PLAINTEXT} \
            -third_stage_spi_clock_divider {4} \
            -init_timeout 45 \
            -auto_calib_timeout {2000} \
            -broadcast_RAMs {0}
```

This example initializes data with SPI-FLASH client.

```
configure_design_initialization_data \
            -second_stage_start_address {0x00000000} \
            -third_stage_uprom_start_address {0x00000000} \
            -third_stage_snvm_start_address {0x00000000} \
            -third_stage_spi_start_address {0x000AC120}
            -third_stage_spi_type {SPIFLASH_BINDING_UEK2} \
            -third_stage_spi_clock_divider {2} \
            -init_timeout 20 \
            -auto_calib_timeout {500} \
            -broadcast_RAMs {1}
```

10.2. configure_ram (Ask a Question)

Description

This Tcl command configures the Fabric RAM clients in the Fabric RAMs tab of the Design and Memory Initialization tool. The target storage type for the third stage initialization can be specified for each Fabric RAM client in the cfg file specified here.

Note: You must run Generate Design Initialization Data (generate_design_initialization_data) after configuring the Fabric RAMs (configure_ram) and/or Design Initialization (configure_design_initialization_data)

```
configure_ram -cfg_file {path_to_configuration_file.cfg}
```

Arguments

Parameter	Type	Description
cfg_file	string	Specifies the path to the configuration file of the Fabric RAM client. This parameter is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'cfg_file' is missing.
None	Unable to locate or read RAM configuration file '*RAM.cfg'.
None	You have to specify RAM configuration file for 'cfg_file' parameter.
None	Parameter 'cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_ram -cfg_file "cfg_file"'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example configures the Fabric RAM clients in the {`./src/RAM.cfg`} file.

```
configure_ram -cfg_file {./src/RAM.cfg}
```

10.3. configure_envm (Ask a Question)

Description

This Tcl command configures the eNVM clients in the eNVM tab of the Design and Memory Initialization tool. This command also specifies the user eNVM clients.

Important: You must run Generate Design Initialization Data (generate_design_initialization_data) after configuring eNVM (configure_envm) and/or Design Initialization (configure_design_initialization_data).

```
configure_envm -cfg_file {path_to_configuration_file.cfg}
```

Arguments

Parameter	Type	Description
cfg_file	string	Specifies the path to the configuration file of the eNVM client. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'cfg_file' is missing.
None	Unable to locate or read ENVM configuration file '*ENVM.cfg'.
None	You have to specify ENVM configuration file for 'cfg_file' parameter.
None	Parameter 'cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_envm -cfg_file "cfg_file"'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC

Example

This example configures the eNVM clients with {./src/ENVM.cfg} file.

```
configure_envm -cfg_file {./src/ENVM.cfg}
```

10.4. configure_snvm (Ask a Question)

Description

This Tcl command configures the sNVM clients in the sNVM tab of the Design and Memory Initialization tool. Can specify user sNVM clients using this command.

Note: You must run Generate Design Initialization Data (generate_design_initialization_data) after configuring sNVM (configure_snvm) and/or Design Initialization (configure_design_initialization_data).

```
configure_snvm -cfg_file {path_to_configuration_file.cfg}
```

Arguments

Parameter	Type	Description
cfg_file	string	Specifies the path to the configuration file of the sNVM client. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'cfg_file' is missing.
None	Unable to locate or read SNVM configuration file '*SNVM.cfg'.
None	You have to specify SNVM configuration file for 'cfg_file' parameter.
None	Parameter 'cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_snvm -cfg_file "cfg_file"'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example configures the sNVM clients with {./src/SNVM.cfg} file.

```
configure_snvm -cfg_file {./src/SNVM.cfg}
```

10.5. configure_spiflash (Ask a Question)

Description

This Tcl command configures the SPI Flash clients in the SPI Flash tab of the Design and Memory Initialization tool. Can specify user SPI FLASH clients using this command.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: You must run Generate Design Initialization Data (generate_design_initialization_data) after configuring SPI Flash (configure_spiflash) and/or Design Initialization (configure_design_initialization_data).

```
configure_spiflash -cfg_file {path_to_configuration_file.cfg}
```

Arguments

Parameter	Type	Description
cfg_file	string	Specifies the path to the configuration file of the SPI FLASH client. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'cfg_file' is missing.
None	Unable to locate or read SPI Flash configuration file '*SNVM.cfg'.
None	You have to specify SPI Flash configuration file for 'cfg_file' parameter.
None	Parameter 'cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_spiflash -cfg_file "cfg_file"'.

Supported Families

PolarFire®

PolarFire SoC

Example

This example configures the SPI Flash clients with {`./src/spiflash.cfg`} file.

```
configure_spiflash -cfg_file {./src/spiflash.cfg}
```

10.6. configure_uprom (Ask a Question)

Description

This Tcl command configures the uPROM clients in the uPROM tab of the Design and Memory Initialization tool. Can specify user uPROM clients using this command.

Note: You must run Generate Design Initialization Data (generate_design_initialization_data) after configuring uPROM (configure_uprom) and/or Design Initialization (configure_design_initialization_data).

```
configure_uprom -cfg_file {path_to_configuration_file.cfg}
```

Arguments

Parameter	Type	Description
cfg_file	string	Specifies the path to the configuration file of the uPROM client. This parameter is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'cfg_file' is missing.
None	You have to specify UPROM configuration file for 'cfg_file' parameter.
None	Parameter 'cfg_file' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'configure_uprom -cfg_file "cfg_file"'.
None	Unable to locate or read UPROM configuration file '*UPROM.cfg'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example configures the uPROM clients with {`./src/UPROM.cfg`} file.

```
configure_uprom -cfg_file {./src/UPROM.cfg}
```

10.7. export_spiflash_image (Ask a Question)

Description

This Tcl command exports an SPI Flash image file to a specified directory.

```
export_spiflash_image \
            [-file_name {name of file}] \
            [-export_dir {absolute path to folder location}]
            [-format {SPI Flash Image format}]
            [-spiflash_device {SPI Flash device type}]
```

Arguments

Parameter	Type	Description
file_name	string	Specifies the name of the image file that will be exported. This parameter is optional. The default file name is the root component's name.
export_dir	string	Specifies the directory location for the export. This parameter is optional. The default export location is an export folder in the design hierarchy.
format	string	Specifies the SPI Flash image format. This parameter is optional. The following options are supported. • STP • BIN (default)
spiflash_device	string	Specifies the type of SPI Flash memory device. This parameter is optional. The following options are supported. • MT25QL01GB (default) • MT25QU01GB • MT25QL02GCBB • MT25QU02GCBB **Important:** This parameter is only applicable if `format` is set to STP.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'export_dir' has illegal value.
None	Parameter 'file_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'export_spiflash_image [-file_name "file_name"] [-export_dir "export_dir"]'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example exports SPI Flash image in {`.src/top.bin`} file.

```
export_spiflash_image -file_name {top} -export_dir {./src} -format {BIN}
```

10.8. generate_design_initialization_data (Ask a Question)

Description

This Tcl command creates the memory files on disk, adds the initialization clients to the target memories, and writes the configuration files to disk. This command also runs validation on the saved configuration files and writes out errors (if any) in the log. This command causes the UI of the Configure Design Initialization Data and Memories tool to refresh and show the latest configuration and validation errors (if any) in the tables. This command takes no parameters.

```
generate_design_initialization_data
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'generate_design_initialization_data'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example creates the memory files on disk, adds the initialization clients to the target memories, and writes the configuration files to disk.

```
generate_design_initialization_data
```


© 2025 Microchip Technology Inc. and its subsidiaries

10.9. generate_initialization_mem_files (Ask a Question)

Description

This Tcl command sets the parameter values needed for generating memory files to be used with design initialization clients.

```
generate_initialization_mem_files \
            [-second_stage_start_address {valid snvm address}] \
            [-third_stage_start_address {valid address for third stage memory type}] \
            [-third_stage_memory_type {UPROM | SNVM | SPIFLASH_NONAUTH}] \
            [-third_stage_spi_clock_divider {1 | 2 | 4 | 6}] \
            [-init_timeout {int between 1 and 128 seconds}] \
            [-custom_cfg_file {valid user specified configuration file}]
```

Arguments

Parameter	Type	Description
second_stage_start_address	string	String parameter for the start address of the second stage sNVM initialization client. Specified as a 32-bit hexadecimal string.The second stage client is always placed in sNVM, so it must be a valid sNVM address aligned on a page boundary. This address will be between 0 and DB00. There are 221 sNVM pages and each page is 256 bytes long. The last two pages are reserved for the first stage initialization client so they are not available for the second stage initialization client.
third_stage_start_address	string	The memory where the third stage initialization client will be placed. The value can be UPROM, SNVM, or SPIFLASH_NONAUTH. The default is sNVM. This parameter determines the valid value for parameter 'third_stage_start_address'.
third_stage_memory_type	string	String parameter for the start address of the third stage initialization client. Specified as a 32-bit hexadecimal string, and must be one of the following: • valid sNVM address aligned on a page boundary. • valid UPROM address aligned on a block boundary. • valid SPIFLASH address.
third_stage_spi_clock_divider	integer	The value can be 1, 2, 4, or 6. The default value is 1.
init_timeout	integer	Timeout value in seconds. Initialization is aborted if it does not complete before timeout expires. The value can be between 1 and 128. The default value is 128.
custom_cfg_file	string	Specifies the user_specified configuration file to be loaded in.

Return Type	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

Example

```
generate_initialization_mem_files \
            -second_stage_start_address 200 \
            -third_stage_memory_type UPROM \
            -third_stage_start_address 400 \
            -third_stage_spi_clock_divider 6 \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
            -init_timeout 120 \
            -custom_cfg_file {./src/my.txt}
```

10.10. modified_client (Ask a Question)

Description

This Tcl command appears in the fabric RAM configuration file when the Fabric RAM is modified to initialize memory.

```
modified_client \
   -logical_instance_name  { RAM user defined instance name }  \
   -storage_type { Initialization client storage type }  \
   -content_type { NO_CONTENT | MEMORY_FILE }  \
   -memory_file_format { Intel-Hex | Motorola-S | Simple-Hex | Microsemi-Binary }  \
   -memory_file { path }
```

Arguments

Parameter	Type	Description
logical_instance_name	string	Specifies the name of the user defined memory instance.
storage_type	string	Specifies storage type to initialize RAM: • SNVM • µPROM • SPI
content_type	string	Specifies the type of memory content: • MEMORY_FILE - content_file parameter must be specified. • NO_CONTENT - no content memory file.
memory_file_format	string	Specifies the memory file format: • Intel-Hex • Motorola-S • Simple-Hex • Microchip-Binary
memory_file	string	Specifies the absolute or relative path of the memory file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

PolarFire SoC

Example

The following example show the `modified_client` Tcl command.

```
modified_client \
   -logical_instance_name  {PF_TPSRAM_C0_0/PF_TPSRAM_C0_0}  \
   -storage_type {SNVM}  \
   -content_type {MEMORY_FILE}  \
   -memory_file_format {Intel-Hex}  \
   -memory_file {Y:/TP_mem/TPRAM_1024_40lp/mem_files/hex .hex}
```


© 2025 Microchip Technology Inc. and its subsidiaries

10.11. nvm_update_serialization_client (Ask a Question)

Description

This Tcl command updates an existing serialization client in the SmartFusion 2 and IGLOO 2 eNVM.

Note: You can repeat <params> argument for specifying multiple parameters.

```
nvm_update_serialization_client -params {parameter:value}
```

This command is usually put in a configuration `*.cfg` file and passed as an argument to the script parameter of the run_tool command.

```
run_tool -name {UPDATE_ENVM} -script "update.cfg"
```

Arguments

Parameter	Type	Description
client_name	string	Specifies the name of the eNVM serialization client to update.
number_of_words	decimal	Specifies the number of words, `number_of_words` available to users = Number of user pages * BYTES_PER_PAGE *8 / word_size.
use_for_simulation	boolean	Specifies whether or not the serialization client is used for simulation. The possible value are: true, 1, false or 0.
base_address	hexadecimal	Specifies the client base address. eNVM address range and available number of words are device dependent. See the eNVM Configuration User Guide for details.
maximum_devices_to_program	integer	Specify maximum devices to program.
reprogram	boolean	Specifies whether reprogram or not.
content_from_file	boolean	Specify the content from file.
number_of_pages	decimal	Specify the number of pages, `number_of_words` available to users = Number of user pages * BYTES_PER_PAGE *8 / word_size.
content_file	string	Specify absolute or relative path to content file.
content_file_format	string	Specifies the content file format: Decimal or Hexadecimal.
start_value	integer	Specifies the start value.
step_value	integer	Specifies the step value.
maximum_value	integer	Specifies the maximum value. See eNVM Configuration User Guide for details.
use_as_rom	boolean	Specifies whether or not the serialization client is to be used as ROM.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

SmartFusion® 2

IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example configures eNVM client with the `update_envm.cfg`: sets name, maximum devices to program and maximum value:

```
nvm_update_serialization_client \
                    -client_name {client1} \
                    -maximum_devices_to_program {6}
                    -maximum_value {12}
```

See Also

- UPDATE_ENVM
- nvm_update_storage_client

10.12. nvm_update_storage_client (Ask a Question)

Description

This Tcl command updates an existing storage client in the SmartFusion 2 and IGLOO 2 eNVM.

Note: You can repeat <params> argument for specifying multiple parameters.

```
nvm_update_storage_client -params {parameter:value}
```

This command is usually put in a configuration `*.cfg` file and passed as an argument to the script parameter of the run_tool command.

```
run_tool -name {UPDATE_ENVM} -script "update.cfg"
```

Arguments

Parameter	Type	Description
client_name	string	Specifies the name of the eNVM storage client to update.
number_of_words	decimal	Specifies the number of words, `number_of_words` available to users = Number of user pages * BYTES_PER_PAGE *8 / word_size.
use_for_simulation	boolean	Specifies whether or not the storage client is used for simulation. The possible value are: true, 1, false or 0.
base_address	hexadecimal	Specifies the client base address. eNVM address range and available number of words are device dependent. See the eNVM Configuration User Guide for details.
retrieve_address	boolean	Specifies whether or not the address is retrieved from a file.
reprogram	boolean	Specifies whether reprogram or not the data storage client is re-programmed.
memory_file_format	string	Specifies the memory file format: INTELHEX | MOTOROLAS | SIMPLEHEX | BINARY.
memory_file	string	Specifies the absolute or relative path of the memory file.
content_type	string or integer	Specifies the content type.
lock_address	boolean	If set to 1, the start address of the client(s) is locked and cannot be changed during optimization..
static_fill_pattern	string	Specifies the static fill pattern: 0 or 1.
use_as_rom	boolean	Specifies whether or not the data storage client is to be used as ROM.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	None

Supported Families

SmartFusion® 2

IGLOO® 2

Example

The following example configures eNVM storage client with "update_envm.cfg": sets name, maximum devices to program and maximum value:

```
nvm_update_storage_client \
            -client_name {client1} \
            -word_size 32 \
                    -number_of_words {512}
                    -maximum_value {12}
```

See Also

- UPDATE_ENVM
- nvm_update_serialization_client

10.13. remove_permanent_locks (Ask a Question)

Description

Removes all the locks configured in SPM_OTP. This command can only be used when at least one lock is disabled using SPM_OTP.

```
remove_permanent_locks
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_permanent_locks'.

Supported Families

PolarFire®

PolarFire SoC

Example

This example removes all the locks configured in SPM_OTP.

```
remove_permanent_locks
```


© 2025 Microchip Technology Inc. and its subsidiaries

10.14. select_programmer (Ask a Question)

Description

This Tcl command enables the specified programmer and disables all other connected programmers. This command is useful when multiple programmers are connected.

```
select_programmer -programmer_id {programmer id} \
                  [-host_name {host name}] \
                  [-host_port {host port}]
```

Arguments

Parameter	Type	Description
programmer_id	string	The programmer to be enabled. This parameter is mandatory.
host_name	string	The host name or IP address. This argument is required for a remote programmer and optional for a local programmer. For local programmer, if specified it must be “localhost”. This parameter is optional.
host_port	string	This argument is required for a remote programmer and optional for a local programmer. If omitted, the default port is used (currently, the default is 80). For a local host, both “localhost” and its port should be specified or omitted. This parameter is optional. **Note:** The def variable “LOCAL_PROGRAM_DEBUG_SERVER_PORT” is used to set a different default local hostport.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'programmer_id' is missing.
None	Parameter 'host_port' has illegal value.
None	Parameter 'host_name' has illegal value.
None	Unable to select programmer 'value'.
None	Parameter 'programmer_id' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'select_programmer -programmer_id "programmer_id" [-host_name "host_name"] [-host_port "host_port"]'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

This example selects the programmer with {00557} id, {localhost} host name and {80} host port.

```
select_programmer -programmer_id {00557} \
                  -host_name {localhost} \
                  -host_port {80}
```


© 2025 Microchip Technology Inc. and its subsidiaries

This example selects the programmer with {00557} id.

```
select_programmer -programmer_id {00557}
```

10.15. set_auto_update_mode (Ask a Question)

Description

This Tcl command enables or disables auto update. This command is added to the spiflash.cfg file that is given as the parameter to the configure_spiflash command.

```
set_auto_update_mode {0|1}
```

Arguments

Parameter	Type	Description
None	boolean	If set_auto_update_mode is 0, auto update is disabled. If `set_auto_update_mode` is 1, auto update is enabled.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

PolarFire SoC

Example

This example sets auto update disabled.

```
set_auto_update_mode {0}
```

10.16. set_manufacturer (Ask a Question)

Description

This command specifies the manufacturer for the SPI Flash device.

```
set_manufacturer {MICRON | SPANSION | Macronix | Winbond }
```

Arguments

Parameter	Type	Description
None	string	The value for the set_manufacturer command must be one of the following: • MICRON • SPANSION • Macronix • Winbond

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

PolarFire SoC

Example

This example sets the manufacturer {MICRON}.

```
set_manufacturer {MICRON}
```

10.17. set_programming_interface (Ask a Question)

Description

This Tcl command sets the programming interface.

```
set_programming_interface [-interface {JTAG | SPI_SLAVE}]
```

Arguments

Parameter	Type	Description
interface	string	Specify the programming interface as JTAG or SPI_SLAVE. The default is JTAG.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	interface: Invalid argument value: 'value' (expecting JTAG or SPI_SLAVE).
None	Parameter 'interface' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_programming_interface [-interface "JTAG | SPI_SLAVE"]'.

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

RTG4™

PolarFire SoC

Example

This example sets programming

```
set_programming_interface -interface {SPI_SLAVE}
```


© 2025 Microchip Technology Inc. and its subsidiaries

10.18. set_cipher_text_auth_client (Ask a Question)

Description

This Tcl command is added to the `sNVM.cfg` file that is given as the parameter to the configure_snvm command. Cipher-text Authenticated clients have 236 bytes available for user data in each page of sNVM.

```
set_cipher_text_auth_client \
              -client_name {name} \
              -number_of_bytes {number} \
              -content_type {MEMORY_FILE | STATIC_FILL} \
              -content_file_format {Microchip-Binary 8/16/32 bit} \
              -content_file {path} \
              -start_page {number} \
              -use_for_simulation {0} \
              -reprogram {0 | 1} \
              -use_as_rom {0 | 1}
```

Arguments

Parameter	Type	Description
client_name	string	The name of the client. Needs to start with an alphabetic letter. Underscores and numerals are allowed at all positions other than the first.
number_of_bytes	integer	The size of the client specified in bytes.
content_type	string	Source of data for the client. This can either be a memory file or all zeros. Allowed values are MEMORY_FILE or STATIC_FILL.
content_file_format	string	Only 'Microchip-Binary 8/16/32 bit' is supported at this time.
content_file	string	Path of the memory file. This can be absolute or relative to the project.
start_page	integer	The page number in sNVM where data for this client will be placed.
use_for_simulation	integer	Only value 0 is allowed.
reprogram	boolean	Specifies whether the client will be programmed into the final design or not. Possible values are 0 or 1.
use_as_rom	boolean	Specifies whether the client will allow only reads or both read and writes. Possible values are 0 or 1.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®
PolarFire SoC

Example

The following example adds CipherText Authenticated client:

```
set_cipher_text_auth_client \
        -client_name {client1}  \
        -number_of_bytes 5 \
        -content_type {STATIC_FILL}  \
        -content_file_format {Microchip-Binary 8/16/32 bit}  \
        -content_file {}  \
        -mem_file_base_address {0x00000000}  \
        -start_page 100 \
        -use_for_simulation 0 \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
        -reprogram 0 \
        -use_as_rom 1
```

See Also

- set_plain_text_client
- set_plain_text_auth_client
- set_usk_client

10.19. set_plain_text_auth_client (Ask a Question)

Description

This Tcl command is added to the `sNVM.cfg` file that is given as the parameter to the configure_snvm command. Plain-text Authenticated clients have 236 bytes available for user data in each page of sNVM.

```
set_plain_text_auth_client \
        -client_name {name} \
            -number_of_bytes {number} \
            -content_type {MEMORY_FILE | STATIC_FILL} \
            -content_file_format {Microchip-Binary 8/16/32 bit} \
            -content_file {path} \
            -start_page {number} \
            -use_for_simulation 0 \
            -reprogram {0 | 1} \
            -use_as_rom {0 | 1}
```

Arguments

Parameter	Type	Description
client_name	string	Specifies the name of the client. Needs to start with an alphabetic letter. Underscores and numerals are allowed at all positions other than the first.
number_of_bytes	integer	Specifies the size of the client specified in bytes.
content_type	string	Specifies the source of data for the client. This can either be a memory file, or all zeros. Allowed values are MEMORY_FILE or STATIC_FILL.
content_file_format	string	Only 'Microchip-Binary 8/16/32 bit' is supported at this time.
content_file	string	Path of the memory file. This can be absolute, or relative to the project.
start_page	integer	The page number in sNVM where data for this client will be placed.
use_for_simulation	boolean	Only value 0 is allowed.
reprogram	boolean	Specifies whether the client will be programmed into the final design or not. Possible values are 0 or 1.
use_as_rom	boolean	Specifies whether the client will allow only reads, or both read and writes. Possible values are 0 or 1.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example sets the Plain-text Authenticated client with the following parameter values.

```
set_plain_text_auth_client \
            -client_name {b} \
            -number_of_bytes {12} \
            -content_type {MEMORY_FILE} \
            -content_file_format {Microchip-Binary 8/16/32 bit} \
            -content_file {} \
            -start_page 2 \
            -use_for_simulation 0 \
            -reprogram 1 \
            -use_as_rom 0
```

See Also

- set_plain_text_client
- set_cipher_text_auth_client
- set_usk_client

10.20. set_plain_text_client (Ask a Question)

Description

This Tcl command is added to the `sNVM.cfg` file that is given as the parameter to the configure_snvm command.

```
set_plain_text_client -client_name {name} \
                      -number_of_bytes {number} \
                      -content_type {MEMORY_FILE | STATIC_FILL} \
                      -content_file_format {Microchip-Binary 8/16/32 bit} \
                      -content_file {path} \
                      -start_page {number} \
                      -use_for_simulation {0} \
                      -reprogram {0 | 1} \
                      -use_as_rom {0 | 1}
```

Arguments

Parameter	Type	Description
client_name	string	Specifies the name of the client. Needs to start with an alphabetic letter. Underscores and numerals are allowed at all positions other than the first.
number_of_bytes	integer	The size of the client specified in bytes.
content_type	string	Source of data for the client. This can either be a memory file, or all zeros. Allowed values are MEMORY_FILE or STATIC_FILL
content_file_format	string	Only 'Microchip-Binary 8/16/32 bit' is supported at this time.
content_file	string	Path of the memory file. This can be absolute, or relative to the project.
start_page	integer	The page number in sNVM where data for this client will be placed.
use_for_simulation	boolean	Only value 0 is allowed.
reprogram	boolean	Specifies whether the client will be programmed into the final design or not. Possible values are 0 or 1.
use_as_rom	boolean	Specifies whether the client will allow only reads, or both read and writes. Possible values are 0 or 1.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

PolarFire SoC

Example

This example sets plain_tets client with the following parameter values.

```
set_plain_text_client -client_name {a} \
                      -number_of_bytes 12 \
                      -content_type {MEMORY_FILE} \
                      -content_file_format {Microchip-Binary 8/16/32 bit} \
                      -content_file {} \
                      -start_page 1 \
                      -use_for_simulation 0 \
                      -reprogram 1 \
                      -use_as_rom 0
```

See Also

- set_cipher_text_auth_client
- set_usk_client

10.21. set_usk_client (Ask a Question)

Description

This Tcl command is added to the `sNVM .cfg` file that is given as the parameter to the configure_snvm command. The USK client is required if sNVM has one or more clients of type 'Authenticated'.

```
set_usk_client -start_page {number} \
               -key {Hexadecimal string of size 24} \
               -use_for_simulation {0 | 1} \
               -reprogram {0 | 1}
```

Arguments

Parameter	Type	Description
start_page	integer	The page number in sNVM where data for this client will be placed.
key	hexadecimal	A string of 24 hexadecimal characters.
use_for_simulation	boolean	Specifies whether the client will be used for simulation or not. Possible values are 0 or 1.
reprogram	boolean	Specifies whether the client will be programmed into the final design or not. Possible values are 0 or 1.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
PolarFire SoC

Example

This example sets the USK client with the following parameter values.

```
set_usk_client -start_page 4 \
               -key {D8C8831F3A2F72EDC569503F} \
               -use_for_simulation 0 \
               -reprogram 1
```

See Also

- set_plain_text_client
- set_plain_text_auth_client
- set_cipher_text_auth_client

10.22. set_client (Ask a Question)

Description

Use the following Tcl command to specify the client that will be added to the SPI Flash memory configuration file along with the `configure_spiflash` command.

```
set_client -client_name {client name} \
           -client_type {FILE_SPI | FILE_SPI_GOLDEN | FILE_SPI_UPDATE |
FILE_DATA_STORAGE_INTELHEX} \
           -content_type {MEMORY_FILE | STATIC_FILL} \
           -content_file {} \
           -start_address {} \
           -client_size {} \
           -program {0|1}
```

Use the following Tcl command to specify the client that will be added to the fabric RAM configuration file along with the `configure_ram` command.

```
set_client \
   -logical_instance_name  {RAM user defined instance name}  \
   -storage_type { Initialization client storage type}  \
   -content_type { NO_CONTENT | MEMORY_FILE } \
   -memory_file_format { Intel-Hex | Motorola-S | Simple-Hex | Microsemi-Binary}  \
   -memory_file {path}
```

Arguments

Parameter	Type	Description
client_name	string	The name of the client. Maximum of 32 characters, letters or numbers or “-” or “_”.
client_type	string	The -client_type can be FILE_SPI, FILE_SPI_GOLDEN, FILE_SPI_UPDATE orFILE_DATA_STORAGE_INTELHEX. • FILE_SPI - SPI Bitstream. • FILE_SPI_GOLDEN - Recovery/Golden SPI Bitstream. • FILE_SPI_UPDATE - Auto Update SPI Bitstream, available only if Auto Update is enabled. See set_auto_update_mode. • FILE_DATA_STORAGE_INTELHEX - Data Storage client.

© 2025 Microchip Technology Inc. and its subsidiaries

set_client (continued)		
Parameter	**Type**	**Description**
content_type	string	Specifies the type of memory content: • MEMORY_FILE - content_file parameter must be specified. • STATIC_FILL - client memory will be filled with 1s, no content memory file. • NO_CONTENT - no content memory file.
content_file	string	Absolute or relative path to the content memory file.
start_address	integer	The client start address. Note that some space is reserved for the SPI Flash Memory directory. This is a decimal value of bytes.
client_size	integer	Client's size in bytes. If a content file is specified, the size must be equal to or larger than the file size. This is a decimal value.
program	boolean	Only program | 1 is supported in this release.
logical_instance_name	string	Specifies the name for the user defined memory instance.
storage_type	string	Specifies storage type to initialize RAM: • SNVM • UPROM • SPI
memory_file_format	string	Specifies the memory file format: • Intel-Hex • Motorola-S • Simple-Hex • Microchip-Binary
memory_file	string	Specifies the absolute or relative path of the memory file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®
PolarFire SoC

Example

The following examples show the `set_client` Tcl command for SPI Flash.

```
set_client -client_name {golden} \
           -client_type {FILE_SPI_GOLDEN} \
           -content_type {MEMORY_FILE} \
           -content_file {E:\top_design_ver_1.spi} \
           -start_address {1024} \
           -client_size {9508587} \
           -program {1}
```

```
set_client -client_name {cl1} \
           -client_type {FILE_DATA_STORAGE_INTELHEX} \
           -content_type {STATIC_FILL} \
           -content_file {} \
           -start_address {1024} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
            -client_size {100} \
            -program {1}
```

The following examples show the `set_client` Tcl command for the fabric RAM configuration file.

```
set_client \
   -logical_instance_name  {PF_TPSRAM_C0_0/PF_TPSRAM_C0_0}  \
   -storage_type {SNVM}  \
   -content_type {MEMORY_FILE}  \
   -memory_file_format {}  \
   -memory_file {Y:/DP_mem/DPRAM_1k_20/hexmod.hex}
```

10.23. set_data_storage_client (Ask a Question)

Description

This Tcl command is added to the `uprom.cfg` file, which will then be given as the parameter to the configure_uprom command.

```
set_data_storage_client -client_name {name} \
                        -number_of_words {number} \
                        -content_type {MEMORY_FILE | STATIC_FILL} \
                        -memory_file_format {Microsemi-Binary} \
                        -memory_file {path} \
                        -base_address {hexadecimal_string} \
                        -use_for_simulation {0}
```

Arguments

Parameter	Type	Description
client_name	string	The name of the client. Must start with an alphabetic letter. Underscores and numerals are allowed at all positions other than the first.
number_of_words	integer	The size of the client specified in number of words.
content_type	string	Source of data for the client. This can either be a memory file, or all zeros. Allowed values are MEMORY_FILE or STATIC_FILL. • MEMORY_FILE - content memory file must be specified. • STATIC_FILL - client memory will be filled with 1s, no content memory file.
memory_file_format	string	Only 'Microchip-Binary' is supported at this time.
memory_file	string	Path of the memory file. This can be absolute, or relative to the project.
base_address	hexadecimal	Hexadecimal address where the first byte of user data will be placed.
use_for_simulation	boolean	Only value 0 is allowed.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example sets data storage client with the following parameter values.

```
set_data_storage_client -client_name {client1} \
                        -number_of_words {57} \
                        -use_for_simulation {0} \
                        -content_type {MEMORY_FILE} \
                        -memory_file_format {Microsemi-Binary} \
                        -memory_file {D:/local_z_folder/work/memory_files/
sar_86586_uprom.mem} \
                        -base_address {0}
```

10.24. update_storage_client (Ask a Question)

Description

This command updates an existing uPROM storage client for the RTG4 uPROM.

Note: You can repeat <params> argument for specifying multiple parameters.

```
update_storage_client -params {parameter:value}
```

This command is usually put in a configuration "*.cfg" file and passed as an argument to the script parameter of the run_tool command.

```
run_tool -name {UPDATE_UPROM} -script "update.cfg"
```

Arguments

Parameter	Type	Description
client_name	string	Specifies the name of the uPROM storage client to update.
number_of_words	decimal	Specifies the number of words, decimal 1 to 10, 400. `number_of_words` available to users = Number of user pages * BYTES_PER_PAGE *8 / word_size.
use_for_simulation	boolean	Specifies whether or not the storage client is used for simulation. The possible value are: true, 1, false or 0.
base_address	hexadecimal	Specifies the client base address: hexadecimal(0-0x289F). eNVM address range and available number of words are device dependent. See the eNVM Configuration User Guide for details.
retrieve_address	boolean	Specifies whether or not the address is retrieved from a file.
memory_file_format	string	Specifies the memory file format: {Microchip Binary}.
memory_file	string	Specifies the absolute or relative path of the memory file.
content_type	string or integer	Specifies the content type: MEMORY_FILE, STATI_FILL or NO_CONTENT.
lock_address	boolean	If set to 1, the start address of the client(s) is locked and cannot be changed during optimization..
static_fill_pattern	string	Specifies the static fill pattern: 0 or 1.
use_as_rom	boolean	Specifies whether or not the data storage client is to be used as ROM.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

RTG4™

Example

The following example configures eNVM storage client with "update_envm.cfg": sets name, maximum devices to program and maximum value:

```
update_storage_client \
      -client_name {inc_dat} \
      -number_of_words {10400} \
      -use_for_simulation {0} \
      -content_type {MEMORY_FILE} \
      -memory_file_format {Microchip-Binary} \
      -memory_file "E:/no-IDE/rtg4_uprom_example/uprom1.mem" \
      -base_address {0}
```

See Also

- nvm_update_storage_client

© 2025 Microchip Technology Inc. and its subsidiaries

11. FlashPro Express Tcl Commands (Ask a Question)

11.1. close_project (Ask a Question)

Description

This Tcl command closes the FlashPro or FlashPro Express project. Equivalent to clicking the Project menu, and choosing Close Job Project.

```
close_project
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

Example

This command closes the FlashPro/FlashPro Express project:

```
close_project
```

See Also

- open_project

11.2. configure_flashpro3_prg (Ask a Question)

Description

This Tcl command changes FlashPro3 programmer settings. You can configure FlashPro programmer in Libero SoC Software or via this command in FlashPro Express software. You will be able to set VPUMP voltage for the programmer and force TCK drop down list of frequencies you want to use for the programming. Similarly other programmer such as FlashPro4/5/6 can also be selected and related options can be set.

```
configure_flashpro3_prg [-vpump {ON | OFF}] \
[-clk_mode {discrete_clk | free_running_clk}] \
[-force_freq {ON | OFF}] \
[-freq {freq}]
```

Arguments

Parameter	Type	Description
vpump	string	Enables FlashPro programmer to drive VPUMP. Set to ON to drive VPUMP. Valid values are ON or OFF.

© 2025 Microchip Technology Inc. and its subsidiaries

configure_flashpro3_prg (continued)		
Parameter	**Type**	**Description**
clk_mode	string	Specifies free running or discrete TCK. Valid value is "discrete_clk" or "free_running_cl".
force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. Valid values are ON or OFF.
freq	integer	Specifies the TCK frequency in MHz. It can be between 1 MHz to 6 MHz.

Return Type	Description
None	None

Error Codes

Error Code	Description

None	Parameter 'parameter_name' is not defined. Valid command formatting is
	`configure_flashpro3_prg [-vpump "ON | OFF"] \`
	`[-clk_mode "free_running_clk | discrete_clk"] \`
	`[-force_freq "ON | OFF"] \`
	`[-freq "freq"]`

None	Invalid 'freq' argument value: (expecting 1000000, 2000000, 3000000, 4000000 or 6000000).
None	Invalid 'clk_mode' argument value: '' (expecting free_running_clk or discrete_clk).

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example sets the VPUMP option to ON, TCK to free running, and uses the TCK frequency specified in the programmer file (force_freq is set to OFF):

```
configure_flashpro3_prg -vpump {ON} \
-clk_mode {free_running_clk} \
-force_freq {OFF} -freq {4}
```

The following example sets VPUMP to ON, TCK to discrete, forces the FlashPro software to use the TCK frequency specified in the software (-force_freq is set to ON) at a frequency of 2 MHz:

```
configure_flashpro3_prg -vpump {ON} \
-clk_mode {discrete_clk} \
-force_freq {ON} -freq {2}
```

See Also

- configure_flashpro4_prg
- configure_flashpro5_prg
- configure_flashpro6_prg

11.3. configure_flashpro4_prg (Ask a Question)

Description

This Tcl command changes FlashPro4 programmer settings. You can configure FlashPro programmer in Libero SoC Software or wia this command in FlashPro Express software. You will be able to set VPUMP voltage for the programmer and force TCK drop down list of frequencies you

© 2025 Microchip Technology Inc. and its subsidiaries

want to use for the programming. Similarly other programmer such as FlashPro3/5/6 can also be selected and related options can be set

```
configure_flashpro4_prg [-vpump {ON|OFF}] \
[-clk_mode {discrete_clk | free_running_clk}] \
[-force_freq {ON|OFF}] \
[-freq {freq}]
```

Arguments

Parameter	Type	Description
vpump	string	Enables FlashPro4 programmer to drive VPUMP. Set to ON to drive VPUMP. Valid values are ON or OFF.
clk_mode	string	Specifies free running or discrete TCK. Valid value is "discrete_clk" or "free_running_cl".
force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. Valid values are ON or OFF.
freq	integer	Specifies the TCK frequency in MHz. It can be between 1 MHz to 6 MHz.

Return Type	Description
None	None

Error Codes

Error Code	Description

None	Parameter 'parameter_name' is not defined. Valid command formatting is
	`configure_flashpro4_prg [-vpump "ON | OFF"] \`
	`[-clk_mode "free_running_clk | discrete_clk"] \`
	`[-force_freq "ON | OFF"] \`
	`[-freq "freq"]`

None	Invalid argument value: (expecting 1000000, 2000000, 3000000, 4000000, 5000000 or 6000000).
None	Invalid 'clk_mode' argument value: '' (expecting free_running_clk or discrete_clk).

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example sets the VPUMP option to ON and uses a free running TCK at a frequency of 4 MHz (force_freq is set to OFF):

```
configure_flashpro4_prg -vpump {ON} \
-clk_mode {free_running_clk} \
-force_freq {OFF} \
-freq {4}
```

The following example sets the VPUMP option to ON, uses a discrete TCK and sets force_freq to ON at 2 MHz:

```
configure_flashpro4_prg -vpump {ON} \
-clk_mode {discrete_clk} \
-force_freq {ON} \
-freq {2}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- configure_flashpro3_prg
- configure_flashpro5_prg
- configure_flashpro6_prg

11.4. configure_flashpro5_prg (Ask a Question)

Description

This Tcl command changes FlashPro5 programmer settings. You can configure FlashPro programmer in Libero SoC Software or wia this command in FlashPro Express software. You will be able to set TCK/SCK frequency from the drop down list of frequencies you want to use for the programming.

```
configure_flashpro5_prg \
         -force_freq {ON} \
         -freq {freq}
```

Arguments

Parameter	Type	Description
force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. Valid values are ON, OFF (default).
freq	integer	Specifies the TCK frequency in MHz. The default frequency is 4 MHz. It can be between 1 MHz to 6 MHz or it can be 10,15 or 30 MHz.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Error: freq: Invalid argument value: '50000000' (expecting 1000000, 2000000, 3000000, 4000000, 5000000, 6000000, 10000000, 15000000 or 30000000).
None	Error: force_freq: Invalid argument value: '' (expecting ON or OFF).

Supported Families

PolarFire®

PolarFire SoC

SmartFusion® 2

11.5. configure_flashpro6_prg (Ask a Question)

Description

This Tcl command changes FlashPro6 programmer settings. You can configure FlashPro6 programmer in Libero SoC Software or via this command in FlashPro Express software. You will be able to set VPUMP voltage for the programmer and force JTAG (Joint Test Action Group) or SPI (Serial Peripheral Interface bus) clock from the TCK (JTAG clock) or SCK (SPI clock) drop down list of frequencies you want to use for the programming. Similarly other programmer such as FlashPro5/4/3 can also selected and related options can be set.

```
configure_flashpro6_prg [-vpump {ON | OFF} ] \
[-force_freq {ON | OFF}] \
[-freq "freq"] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
[-force_sck_freq {ON | OFF} \
[-sck_freq "sck_freq"]
```

Arguments

Parameter	Type	Description
vpump	string	Enables FlashPro5 programmer to drive VPUMP. Set to ON to drive VPUMP. Valid value is ON (default) or OFF.
force_freq	string	Forces the FlashPro software to use the TCK frequency specified by the software rather than the TCK frequency specified in the programmer file. Valid values are ON, OFF(default).
freq	integer	Specifies the TCK frequency in MHz. TCK is used with a maximum frequency of 20 MHz, and the default frequency is 4 MHz. It can takes the value between 1 MHz to 6 MHz or it can be 10,15 or 30 MHz.
force_sck_freq	string	Forces the FlashPro software to use the SCK frequency. Valid values are ON, OFF(default).
sck_freq	floating	Specifies the SCK frequency in MHz. SCK is used with a maximum frequency of 40 MHz, and the default frequency is 20 MHz. Limitation of the SCK frequency for the selected programmer: 1.00, 2.00, 2.50, 3.33, 4.00, 5.00, 6.67, 8.00, 10.00, 13.33, 20.00 MHz

Return Type	Description
None	None

Error Codes

Error Code	Description

None	Parameter 'parameter_name' is not defined. Valid command formatting is
	`configure_flashpro6_prg [-vpump "ON | OFF"] \`
	`[-force_freq "ON | OFF"] \`
	`[-freq "freq"] \`
	`[-force_sck_freq "ON | OFF"] \`
	`[-sck_freq "sck_freq"]'`

None	Invalid 'freq' argument value: '4' (expecting 1000000, 2000000, 3000000, 4000000, 5000000, 6000000, 7000000, 8000000, 9000000, 10000000, 11000000, 12000000, 13000000, 14000000, 15000000, 16000000, 17000000, 18000000, 19000000 or 20000000).
None	Invalid 'sck_freq' argument value: '2' (expecting 400000, 500000, 600000, 700000, 800000, 1000000, 2000000, 2500000, 3330000, 4000000, 5000000, 6670000, 8000000, 10000000, 13330000, 20000000 or 40000000)

Supported Families

PolarFire®

PolarFire SoC

SmartFusion® 2

Example

The following example sets TCK at a frequency of 4 MHz and sets force_freq to OFF:

```
configure_flashpro6_prg -force_freq {OFF} -freq {4}
```

The following example sets SCK at a frequency of 2 MHz and sets force_sck_freq to ON:

```
configure_flashpro6_prg -force_sck_freq {ON} -sck_freq {2}
```

See Also

- configure_flashpro3_prg

© 2025 Microchip Technology Inc. and its subsidiaries

- configure_flashpro4_prg
- configure_flashpro5_prg

11.6. create_job_project (Ask a Question)

Description

This Tcl command creates FlashPro Express job using the programming job exported from Libero.

```
create_job_project -job_project_location {job project location} \
-job_file {path and name of job file} \
-overwrite value
```

Arguments

Parameter	Type	Description
job_project_location	string	Specifies the location for your FlashPro Express job (*.job) project. Must include the complete path to the *.pro file. If you do not provide the full path, FlashPro infers that you want to open the project from your current working directory.
job_file	string	Path to the Libero job file (*.job) that is used as input to create the Flashpro Express job project.
overwrite	boolean	Set value to TRUE, true or 1 to overwrite your existing job project. Valid values are: TRUE, true, 1, FALSE. false or 0.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example creates a job project named test.job in the \fpexpress directory. It does not overwrite the existing job project:

```
create_job_project \
-job_project_location {D:\fpexpress} \
-job_file {D:\test\designer\test\export\test.job} -overwrite 0
```

See Also

- open_project

11.7. auto_construct_job_project (Ask a Question)

Description

This Tcl command is available in the developer mode only. It creates a new job project by running auto construct that has a JTAG chain with all the devices disabled. You can save and re-open projects with all devices disabled.

```
auto_construct_job_project \
        -job_project_location {./} \
        -job_name {KR_rotate_1} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
          -programmer {S201Z7NKMY} \
         -overwrite 0
```

System Behaviour

The programmer parameter is processed based on the availability of connected programmers. The following outlines how the tool behaves in different scenarios:

- If the programmer parameter is missing and there is one connected programmer, it is used for auto construct.
- If multiple programmers are connected, the command fails.
- If the programmer parameter is set but the specified programmer is not connected while another programmer is connected, the command runs auto construct on the connected programmer.

Arguments

Parameter	Type	Description
-job_project_location location	string	Specifies the location for your FlashPro Express job project
-job_name	string	Specify the name of the Flashpro Express job project.
-programmer	string	Specify the name of the connected programmer. **Note:** This is an optional parameter.
-overwrite	boolean	Set value to 0 to overwrite your existing job project. **Note:** This is an optional parameter.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example creates a job project named `test.job` in the `\fpexpress` directory. It does not overwrite the existing job project.

```
auto_construct_job_project \
-job_project_location {D:\fpexpress} \
-job_name {test} \
-programmer {138015F} \
-overwrite 0\
```

See Also

- create_job_project
- open_project

© 2025 Microchip Technology Inc. and its subsidiaries

11.8. get_connected_programmers (Ask a Question)

Description

This Tcl command returns a list of currently connected programmers which can be parsed and used to enable or disable programmers.

```
get_connected_programmers
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

PolarFire SoC

PolarFire 2X

Example

```
{1380217 FP6} {S200XTYQWQ FP5}
```

See Also

None

11.9. enable_prg (Ask a Question)

Description

This Tcl command enables or disables the programmer specified. It will give error if the programmer specified for the -name option is not connected to that machine.

```
enable_prg \
-name {programmer_id} \
-enable {value}
```

Arguments

Parameter	Type	Description
name	string	Specify the programmer ID.
enable	boolean	Specify 1 or TRUE to enable the programmer, specify 0 or FALSE to disable the programmer. This is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Programmer mentioned in the -name parameter is not connected to the machine

© 2025 Microchip Technology Inc. and its subsidiaries

enable_prg (continued)	
Error Code	**Description**
None	The programmer with name 'programmer ID' does not exist.
None	Required parameter 'enable' is missing.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is enable_prg [-name "name"]+ -enable "TRUE | FALSE".

Supported Families

PolarFire®

SmartFusion® 2

Example

The following command enables the programmer with programmer ID 13802A15:

```
enable_prg -name {13802A15} -enable 1
```

11.10. enable_prg_type (Ask a Question)

Description

This Tcl command enables or disables the programmer of the type specified in the option -prg_type. If there are multipe programmers of same type connected to the machine, then use the enable_prg TCL command.

```
enable_prg_type \
-prg_type {programmer_type} \
-enable <value>
```

Arguments

Parameter	Type	Description
prg_type	string	Specify one of the following programmer type: FP | FP3 | FP4 | FP5 | FP6 | FP6Lite | PP. This is mandatory.
enable	boolean	Specifies 1 or TRUE to enable programmer, specifies 0 or FALSE to disables programmer. This is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'prg_type' is missing.
None	Required parameter 'enable' is missing.
None	'type' is an invalid programming type. Please specify one of the following valid programming types: (FP|FP3|FP4|FPLite|PP)
None	Parameter 'param_name' is not defined. Valid command formatting is enable_prg_type -prg_type "prg_type" -enable "TRUE | FALSE".

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion® 2

Example

The following command enables the programmer with programmer type FP6:

```
enable_prg_type -prg_type FP6 -enable TRUE
```

See Also

- enable_prg
- ping_prg

11.11. export_script (Ask a Question)

Description

This Tcl command explicitly exports the Tcl command equivalents of the current FlashPro Express session. With this command you can re-execute the same commands interactively or in batch.

You must supply a file name with the -file parameter and the -relative_path parameter to specify whether an absolute or relative path is used in the exported script file.

```
export_script \
-file {absolute or relative path to exported file} \
-relative_path <value>
```

Arguments

Parameter	Type	Description
file	string	Specifies the absolute or relative path and name for the exported TCL script.
relative_path	boolean	Sets your option to use a relative or absolute path in the exported script; use 1 for relative path, 0 for absolute path.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'relative_path' is missing.

Supported Families

PolarFire®

SmartFusion® 2

Example

The following command exports the Tcl command equivalents of the current FlashPro Express session in exported.tcl:

```
export_script -file {./exported.tcl} -relative_path 1
```


© 2025 Microchip Technology Inc. and its subsidiaries

11.12. open_project (Ask a Question)

Description

This Tcl command opens FlashPro Express project (*.pro) that was created in FPExpress.

```
open_project -project {path and name of the project file} \
[-connect_programmers value]
```

Arguments

Parameter	Type	Description
project	string	Specify the path of the FPExpress project with extension *.pro. Project path should include the complete path to the *.pro file. If you do not provide the full path, FlashPro Express infers that you want to open the project from your current working directory.
connect_programmers	boolean	Valid values are: TRUE, true, 1, FALSE. false or 0. This is optional. Default is 1.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'project' is missing.

Supported Families

PolarFire®

SmartFusion® 2

Example

This command opens the 'FPPrj1.pro' project from the FPProject1 directory:

```
open_project -project {./FPProject1/FPPrj1.pro} -connect_programmers 1
```

See Also

- close_project

11.13. ping_prg (Ask a Question)

Description

This is the FlashPro-specific tcl command that pings one or more programmers. Right-click a programmer and choose Ping.

Note: You can click the Refresh/Rescan for Programmers button to quickly ping new programmers.

```
ping_prg -name {name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the programmer to be pinged. Repeat this argument for multiple programmers. See example below. This is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' is missing or has invalid value.
None	The programmer with name does not exist.

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example pings the programmers 'FP300085' and 'FP30086':

```
ping_prg -name {FP300085} -name {FP300086}
```

See Also

- ping_prg

11.14. refresh_prg_list (Ask a Question)

Description

This Tcl command refreshes the programmer list. This is most often used to have FlashPro or FlashPro Express detect a programmer that you have just connected.

```
refresh_prg_list
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

Example

This command refreshes programmers list:

```
refresh_prg_list
```


© 2025 Microchip Technology Inc. and its subsidiaries

11.15. remove_prg (Ask a Question)

Description

This Tcl command removes the programmer from the programmer list. Right-click a programmer to Ping, Self-Test, Scan, Check Chain or Remove it from the list.

```
remove_prg -name { name }
```

Arguments

Parameter	Type	Description
name	string	Specifies the programmer to be removed. You can repeat this argument for multiple programmers.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The programmer with name 'prg_name' does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_prg [-name "name"]+'

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example removes the programmer '03178' from the programmer list:

```
set prg_name "03178"
remove_prg -name "$prg_name"
```

11.16. run_selected_actions (Ask a Question)

Description

FlashPro-specific Tcl command. Runs the selected action on the specified programmer. If no programmer name is specified, the action is run on all connected programmers. A programming file must be loaded.

```
run_selected_actions [-name "name"] \
                     [-force_rtg4_erase "TRUE | FALSE"] \
                     [-prog_spi_flash "TRUE | FALSE"] \
                     [-disable_prog_design "TRUE | FALSE"] \
                     [-spi_flash_image "spi_flash_image"] \
                     [-spi_flash_action "spi_flash_action"]
```

Arguments

Parameter	Type	Description
name	string	Optional argument that specifies the programmer name. You can repeat this argument for multiple programmers(example below).
force_rtg4_erase	boolean	This action is for RT4G device only. When the value of this option is set to "TRUE" the erase action will be run on RT4G device when erase action is executed in batch mode using TCL script.

© 2025 Microchip Technology Inc. and its subsidiaries

run_selected_actions (continued)		
Parameter	**Type**	**Description**
disable_prog_design	boolean	Specify 1 or "TRUE" to disable device programming, specify 0 or "FALSE" to enable device programming.
prog_spi_flash	boolean	Specify 1 or "TRUE" to enable spi flash programming, sepcify 0 or "FALSE" to disable spi flash programming.
spi_flash_image	string	Provide the path of spi flash image (bin) file.
spi_flash_action	string	The value can be one of these: PROGRAM_SPI_IMAGE, VERIFY_SPI_IMAGE, READ_SPI_IMAGE, ERASE_SPI_FLASH.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	The programmer with name doesn't exist.
None	Parameter 'name' has illegal value.
None	Parameter 'argument_name' is not defined. Valid command formatting is 'run_selected_actions [-name "name"]* [-force_rtg4_erase "TRUE | FALSE"] \ [-prog_spi_flash "TRUE | FALSE"] [-disable_prog_design "TRUE | FALSE"] \ [-spi_flash_image "spi_flash_image"] [-spi_flash_action "spi_flash_action"].

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

RTG4™

Example

The following example runs the selected actions on the programmers 'FP30085' and 'FP30086':

```
run_selected_actions -name {FP300085} -name {FP300086}
```

The following example catches pass/fail:

```
if {[catch {run_selected_actions -name {FP300085}]} {
    puts "Error running Action"
} else {
    puts "Action passed"}
```

The following example returning exit code to the command line (returns exit 99 on script failure, otherwise returns 0):

```
if {[catch {run_selected_actions}]}{
    exit 99
} else {
    exit 0
}
```

See Also

- set_programming_action

© 2025 Microchip Technology Inc. and its subsidiaries

11.17. save_log (Ask a Question)

Description

This Tcl command saves your FlashPro or FlashPro Express log file. Equivalent to clicking the Project menu, and choosing Export Log File.

```
save_log -file {absolute or relative path of log file}
```

Arguments

Parameter	Type	Description
file	string	Specifies absolute or relative path and the name of the log file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example saves the log file with the name 'my_logfile.log':

```
save_log -file {my_logfile.log}
```

11.18. save_project (Ask a Question)

Description

This Tcl command saves the FlashPro or FlashPro Express project. Equivalent to clicking the Project menu, and choosing Save Job Project.

```
save_project
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire®

SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

Saves the FlashPro/FlashPro Express job in your current working directory:

```
save_project
```

11.19. scan_chain_prg (Ask a Question)

Description

This Tcl command shows how the devices are ordered in the chain in the Log window. The scan chain operation scans and analyzes the JTAG chain connected to programmer(s) you have selected and checks that chain scanned matches the chain configured in FlashPro Express. To scan a chain: Right-click the programmer you want to scan and choose Scan and check chain. i.e. Device 1: 2A54CF1 Mfr: Microsemi Part: M2AA090T or Device 2: Unknown. In single mode, this command runs scan chain on a programmer. In chain mode, this command runs scan and check chain on a programmer if devices have been added in the grid.

```
scan_chain_prg -name { programmer_name }
```

Arguments

Parameter	Type	Description
name	string	Specifies the programmer name. This is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'scan_chain_prg [-name "name"]
None	The programmer with name 'prg_name' does not exist

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example runs scan chain on a single programmer (single mode) named 'E21428R':

```
set prg_name {E21428R}
scan_chain_prg -name "$prg_name"
```

11.20. self_test_prg (Ask a Question)

Description

This Tcl command Runs Self-Test on a programmer. Right-click a programmer to Ping, Self-Test, Scan, Check Chain or Remove it from the list.

Note: You must connect the programmer to the self-test board that comes with your programmer before performing a self-test. Self-test is not supported with FlashPro5/4 programmers. These programmers are rigorously tested at the factory during production.

```
self_test_prg -name { name }
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
name	string	Specifies the programmer name to run Self-Test. You can repeat this argument for multiple programmers.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The programmer with name 'prg_name' does not exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'self_test_prg [-name "name"]+'
None	programmer 'prg_name' : Self Test FAILED. Make sure the programmer is connected to the Loopback Test Board.

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example runs Self-Test on a '03A178' programmer from the programmer list:

```
set prg_name "03A178"
self_test_prg -name "$prg_name"
```

11.21. set_fpexpress_mode (Ask a Question)

Description

This TCL command allows the user to select the FlashPro express mode: Operator mode or Developer mode. Operator mode is the current default mode. Developer mode allows to update jobs before running programming.

Note: This TCL command will error out if there is an open job. The user has to close open jobs (or have close_project TCL command in the TCL script) before running this command.

```
set_fpexpress_mode [-mode {operator|developer}]
```

Arguments

Parameter	Type	Description
mode	string	Specify FlashPro express mode. The possible value for this argument are: • operator_mode - FlashPro Express mode is set to operator mode. • developer_mode - FlashPro Express mode is set to developer mode.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'mode' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_fpexpress_mode -mode "operator_mode | developer_mode"'.
None	mode: Invalid argument value: ''(expecting operator_mode or developer_mode).

Supported Families

PolarFire®
PolarFire SoC

Example

This example sets FlashPro express mode as "developer_mode":

```
set_fpexpress_mode -mode {developer_mode}
```

11.22. set_hsm_params (Ask a Question)

Description

This Tcl command saves the HSM parameters for the FlashPro Express application. These parameters remain in effect until overridden by another invocation of this command.

Notes:

- The HSM parameters are persistent between multiple FlashPro Express sessions on the same computer.
- HSM parameters only need to be set for HSM flow jobs.

```
set_hsm_params -hsm_server_name hsm_server \
               -hsm_type_u {TRUE|FALSE} \
               -m_hsm_uuid m_uuid \
               -ftp_username {ftp_username} \
               -ftp_password {ftp_password}
```

Arguments

Parameter	Type	Description
hsm_server	string	Name or IP address of HSM server computer.
hsm_type_u	boolean	The possible value for this argument are: • TRUE or 1 - FlashPro Express will use the Manufacturer features of the User HSM. • FALSE or 0 - FlashPro Express will use a Manufacturer HSM.
m_hsm_uuid	string	Specifies UUID of HSM to be used for FlashPro Express tasks.
ftp_username	string	Specifies the user name to access the HSM files via FTP server.
ftp_password	string	Specifies the password to access the HSM files via FTP server.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hsm_server_name' is missing.
None	Could not get server information. Please check server address and connection and try again.

© 2025 Microchip Technology Inc. and its subsidiaries

set_hsm_params (continued)	
Error Code	**Description**
None	HSM server name cannot be empty.
None	hsm_type_u: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Warning:Deprecated 'hsm_type_u' parameter is used.
None	FTP login password must be specified along with the user name.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_hsm_params -hsm_server_name "HSM server machine name or IP address" \ [-m_hsm_uuid "Manufacturer HSM UUID *DEPRECATED PARAMETER*"] \ [-hsm_type_u "TRUE | FALSE"] \ [-ftp_username "FTP login user name"] \ [-ftp_password "FTP login password"]

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets M-HSM parameters:

```
set_hsm_params -hsm_server_name {11.22.33.44} \
               -hsm_type_u {0} \
               -m_hsm_uuid {00000000000000000000000000000000000000000002} \
               -ftp_username {hsm} \
               -ftp_password {hsm}
```

11.23. set_prg_name (Ask a Question)

Description

This Tcl command changes the user name of a programmer. Enter the new programmer name in the Programmer window to rename the programmer. By default, the programmer name is the same as the programmer ID.

```
set_prg_name -name { name } -new_name { new_name }
```

Arguments

Parameter	Type	Description
name	string	Identifies the old programmer name.
new_name	string	Specifies the new programmer name.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is set_prg_name -name "name" -new_name "new_name"
None	Required parameter 'new_name' is missing.
None	The programmer with name 'some_device_name' does not exist.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example changes the name of the programmer 'FP300086' to 'FP3Prg2':

```
set prj_name "FP300086"
set_prg_name -name "$prj_name" -new_name {FP3Prg2}
```

11.24. set_programming_action (Ask a Question)

Description

This is the FlashPro-specific tcl command. Selects the action for a device. The device name parameter must be specified only in chain programming mode. A programming file must be loaded. The device must be a Microchip device.

```
set_programming_action [ -name {name} ] -action { action }
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. It is optional.
action	string	Specifies the action. This is mandatory. Valid values are: • PROGRAM - Programs all selected family features: FPGA Array, targeted eNVM clients and security settings. • ENC_DATA_AUTHENTICATION - Encrypted bitstream authentication data. This action is only visible if every device in the chain contains encrypted bitstream files. Selecting this action causes each bitstream file to be checked for authentication. • ERASE - Erases the selected family features: FPGA Array and security. • DEVICE_INFO - Displays the IDCODE, the design name, the checksum, and device security settings and programming environment information programmed into the device. • READ_IDCODE - Reads the device ID code from the device. • VERIFY - Verifies all selected family features: FPGA Array, targeted eNVM clients and security settings. • VERIFY_DIGEST - Calculates the digests for the components included in the bitstream and compares them against the programmed values.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'action' is missing.
None	You must specify the device name parameter for the command 'set_programming_action' in chain programming mode.
None	The device with name does not exist.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire®
SmartFusion® 2

Example

The following example sets the programming action in single programming mode::

```
set_programming_action -action {PROGRAM}
```

And in chain programming mode:

```
set_programming_action -name {MyDevice1} -action {ERASE}
```

See Also

- set_programming_action

11.25. set_programming_file (Ask a Question)

Description

This Tcl command Sets the programming file for a device. Either the -file or the -no_file flag must be specified. A programming file must be loaded(exported from Libero). The device must be a Microchip device.

```
set_programming_file [-name {name}] \
-file {absolute or relative path and the name of file} \
[-no_file]
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. This argument must be specified only in chain programming mode. It is optional.
file	stirng	Specifies the absolute or relative path and the name of programming file either stp or ppd file. stp and ppd are exported from libero - export bitstream dialog.
no_file	none	Specifies to unload the current programming file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	You must specify the device name parameter for the command 'set_programming_file' in chain programming mode.
None	You must either specify the 'file' or the 'no_file' parameter.
None	Parameter 'file' has illegal value.
None	The device with name 'device_name' does not exist.

Supported Families

PolarFire®
SmartFusion® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Exceptions
Must be a Microchip device.

Example
The following command sets "e:/test/test.ppd" programming file in single programming mode:

```
set_programming_file -file {e:/test/test.ppd}
```

The following command sets "e:/test/test.stp" programming file in chain programming mode:

```
set_programming_file -name {MyDevice1} -file {e:/test/test.stp}
set_programming_file -name {MyDevice1} -no_file
```

11.26. set_programming_interface (Ask a Question)

Description
This Tcl command is used to select JTAG or SPI_SLAVE interface for programming.

```
set_programming_interface -interface {interface}
```

Arguments

Parameter	Type	Description
interface	string	Specifies JTAG or SPI_SLAVE programming.

Return Type	Description
None	None

Error Codes

Error Code	Description

Supported Families

PolarFire®

SmartFusion® 2

Example
The following example selects SPI_SLAVE programming:

```
set_programming_interface -interface {spi_slave}
```

11.27. set_spi_flash_action (Ask a Question)

Description
This Tcl command specifies SPI Flash programming action. You can program, verify or erase SPI Flash using this command.

```
set_spi_flash_action [-name {name}] -spi_flash_action {action}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. It is optional.

© 2025 Microchip Technology Inc. and its subsidiaries

set_spi_flash_action (continued)

Parameter	Type	Description
spi_flash_action	string	Specifies one of the following actions: PROGRAM_SPI_IMAGE, VERIFY_SPI_IMAGE, READ_SPI_IMAGE, ERASE_SPI_FLASH. • PROGRAM_SPI_IMAGE: This action will erase the entire SPI flash then program the SPI image. • VERIFY_SPI_IMAGE: This action verifies the SPI Image on the SPI Flash. • READ_SPI_IMAGE: This action reads the SPI Image from the SPI Flash. • ERASE_SPI_FLASH: This action erases the entire SPI Flash.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'spi_flash_action' is missing.
None	You must specify the device name parameter for the command 'set_spi_flash_action' in chain programming mode.
None	The action 'prg_action' is not supported. You must select the programming action from this list: 'PROGRAM_SPI_IMAGE, VERIFY_SPI_IMAGE, READ_SPI_IMAGE, ERASE_SPI_FLASH'.
None	The device with 'device_name' name does not exist.

Supported Families

PolarFire®

SmartFusion® 2

Example

The following example sets the "VERIFY_SPI_IMAGE" SPI Flash programming action in single programming mode, verifies the SPI Image on the SPI Flash:

```
set_spi_flash_action -name {MPFS250T_ES} \
                     -spi_flash_action {VERIFY_SPI_IMAGE}
```

11.28. set_spi_flash_file (Ask a Question)

Description

This Tcl command specifies SPI Flash programming file(.bin) to be associated with the device.

```
set_spi_flash_file [-name {name}] \
-file {path and the name of the programming file} \
-no_file
```

Arguments

Parameter	Type	Description
name	string	This argument must be specified only in chain programming mode. It is optional.
file	string	Specifies the SPI Flash programming file *.bin.
no_file	none	Specifies to unload/unspecify the SPI Flash programming file.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'action' is missing.
None	The device with name does not exist.

Supported Families

PolarFire®
SmartFusion® 2

Example

The following example sets the "VERIFY_SPI_IMAGE" SPI Flash programming action in single programming mode:

```
set_spi_flash_file -name {MPFS250T_ES} \
                   -spi_flash_action {VERIFY_SPI_IMAGE}
```

11.29. update_fp6_programmers (Ask a Question)

Description

This Tcl command updates all the FlashPro6 programmers that require update. This command takes no parameters. To execute the Tcl Command in Libero SoC, add the command before `run_tool`. To execute the Tcl command in FlashPro Express tool or SmartDebug tool, add the Tcl command before `run_selected_actions`.

```
update_fp6_programmers
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'update_fp6_programmers'.

Supported Families

PolarFire®
PolarFire SoC

Example

This example updates all the FlashPro6 programmers that require update.

```
update_fp6_programmers
```


© 2025 Microchip Technology Inc. and its subsidiaries

12. Configure JTAG Chain Tcl Commands (Ask a Question)

These commands take a script that contains JTAG chain configuration-specific Tcl commands and passes them to FlashPro Express for execution.

Note that these commands cannot be executed directly from Libero.

12.1. add_actel_device (Ask a Question)

Description

This Tcl command adds an Actel device to the chain. Either the file or device parameter must be specified. Chain programming mode must have been set.

```
add_actel_device -name {name} \
                 [-file {filename}] \
                 [-device {device}]
```

Arguments

Parameter	Type	Description
name	string	Specifies the device user name. This parameter is mandatory.
file	string	Specifies a programming filename. This is an optional parameter.
device	string	Specifies the device family (such as MPF300). This is an optional parameter.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'name' has illegal value.
None	You must either specify the 'file' or the 'device' parameter.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_actel_device [-file "file"] [-device "device"] [-name "name"]'.
None	Parameter name is missing or has invalid value.
None	Parameter 'device' has illegal value.
None	Parameter 'file' has illegal value.
None	Empty file 'stp_file.stp'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example adds "MPF300T" device with name "dev_name":

```
add_actel_device -device {MPF300T} -name {dev_name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

12.2. select_libero_design_device (Ask a Question)

Description

This Tcl command selects the Libero design device for the Programming Connectivity and Interface tool within Libero. This command is needed when the tool cannot automatically resolve the Libero design device when there are two or more identical devices that match the Libero design device in the configured JTAG chain.

```
select_libero_design_device -name {device name}
```

Arguments

Parameter	Type	Description
name	string	Specifies a user-assigned unique device name in the JTAG chain. This paramater is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The device '' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'select_libero_design_device -name "name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example selects {M3E050TS} device for the Programming Connectivity and Interface tool within Libero.

```
select_libero_design_device -name {M3E050TS}
```

12.3. copy_device (Ask a Question)

Description

This Tcl command copies a device in the chain to the clipboard. Chain programming mode must be set. See the paste_device command for more information.

```
copy_device -name {name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. You can repeat this argument to copy multiple devices. This paramater is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The device with name 'test_name' doesn't exist.
None	Parameter 'param_name' is not defined. Valid command formatting is 'copy_device [-name "name"]+'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The example copies the device 'MyDevice1' to the same location with a new name 'MyDevice2'.

```
copy_device -name {MyDevice1} -name {MyDevice2}
```

See Also

- paste_device

12.4. cut_device (Ask a Question)

Description

This Tcl command removes one or more devices from the chain. It places the removed device in the clipboard. Chain programming mode must be set to use this command. See the paste_device command for more information.

```
cut_device -name {name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. You can repeat this argument for multiple devices.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	The device with name 'device_name' doesn't exist.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'cut_device [-name "name"]+'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example removes the devices 'MyDevice1' and 'MyDevice2' from the chain.

```
cut_device -name {MyDevice1} -name {MyDevice2}
```

See Also

- paste_device

12.5. enable_device (Ask a Question)

Description

This Tcl command enables or disables a device in the chain (if the device is disabled, it is bypassed). Chain programming mode must be set. The device must be a Microchip device.

```
enable_device -name {device name} -enable {TRUE | FALSE}
```

Arguments

Parameter	Type	Description
name	string	Specifies your device name. This parameter is mandatory.
enable	boolean	Specifies whether the device is to be enabled or disabled. If you specify multiple devices, this argument applies to all specified devices. Specify 1 or "TRUE" to enable device, specify 0 or "FALSE" to enable device.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'enable' is missing.
None	The device with name 'MyDevice1' doesn't exist
None	enable: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'enable' has illegal value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'enable_device -name "name" -enable "TRUE | FALSE"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

enable_device (continued)
Supported Families
RTG4™
SmartFusion® 2
IGLOO® 2

Example
The following example disables the device 'MyDevice' in the chain.

```
enable_device -name {MyDevice} -enable {FALSE}
```

12.6. remove_device (Ask a Question)

Description
This Tcl command removes the device from the chain. Chain programming mode must be set.

```
remove_device -name {name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. You can repeat this argument for multiple devices.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The device with name 'test_name' doesn't exist.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_device [-name "name"]+'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example
The following example removes a device 'M2L050TA' from the chain:

```
remove_device -name {M2L050TA}
```


© 2025 Microchip Technology Inc. and its subsidiaries

12.7. set_device_name (Ask a Question)

Description

This Tcl command changes the user name of a device in the chain. Chain programming mode must be set.

```
set_device_name -name {name} -new_name {new_name}
```

Arguments

Parameter	Type	Description
name	string	Identifies the old device name. This parameter is mandatory.
new_name	string	Specifies the new device name. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'new_name' is missing.
None	The device with name 'device_name' doesn't exist.
None	Parameter 'name' has illegal value.
None	Parameter 'new_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device_name -name "name" -new_name "new_name"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example changes the user name of the device from 'MyDevice1' to 'MyDevice2':

```
set_device_name -name {MyDevice1} -new_name {MyDevice2}
```

12.8. add_non_actel_device (Ask a Question)

Description

This Tcl command adds a non-Actel device in the chain. Either the file or (-tck and -ir) parameters must be specified. The Chain programming mode must have been set.

```
add_non_actel_device [-file "file"] \
                     [-ir "integer value"] \
                     [-tck "decimal value"] \
                     [-name "name"]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
file	string	Specifies a BSDL file.
ir	integer	Specifies the IR length.
tck	integer	Specifies the maximum TCK frequency (in MHz).
name	string	Specifies the device user name.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	You must either specify the 'file' or the 'ir' and 'tck' parameters.
None	Parameter 'file' has illegal value.
None	Parameter 'ir' has illegal value.
None	Parameter 'tck' has illegal value.
None	Parameter 'name' has illegal value.
None	tck: Invalid argument value: '' (expecting decimal value).
None	Parameter 'tck' must be greater than or equal to 1.000
None	Parameter 'tck' must be less than or equal to 100.000.
None	ir: Invalid argument value: '' (expecting integer value).
None	Parameter 'ir' must be greater than or equal to 2.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_non_actel_device [-file "file"] [-ir "integer value"] [-tck "decimal value"] [-name "name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This example adds {MyDevice} non actel device in the chain:

```
add_non_actel_device -ir {8} -tck {5} -name {MyDevice}
```

12.9. add_non_actel_device_to_database (Ask a Question)

Description

This Tcl command imports settings via a BSDL file that adds non-Actel or non-Microchip devices to the device database so that they are recognized during scan chain and auto-construction operations.

```
add_non_actel_device_to_database -file {file path to the *.bsdl}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
file	string	Specifies the path to the BSDL file and the BSDL filename add to the database. You can repeat this argument for specifing multiple BSDL files.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'file' is missing or has invalid value.
None	Parameter 'file' has illegal value.
None	Failed to import '*.bsd'.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_non_actel_device_to_database [-file "file"]+'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example uses a BSDL file to add a non-Microchip device to the device database:

```
add_non_actel_device_to_database -file {./src/top.bsd}
```

See Also

- remove_non_actel_device_from_database

12.10. construct_chain_automatically (Ask a Question)

Description

This Tcl command automatically starts chain construction for the specified programmer.

```
construct_chain_automatically -name {name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the programmer(s) name(s). You can repeat this argument to specify multiple programmers names.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' is missing or has invalid value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'construct_chain_automatically [-name "name"]+'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

For a single programmer:

```
construct_chain_automatically -name {E2004NSXB1}
```

See Also

- scan_chain_prg
- enable_device
- set_debug_programmer
- set_device_name
- set_programming_file
- set_programming_action
- run_selected_actions

12.11. paste_device (Ask a Question)

Description

This Tcl command pastes the devices that are on the clipboard in the chain, immediately above the <position_name> device, if this parameter is specified. Otherwise it places the devices at the end of the chain. The chain programming mode must be enabled.

```
paste_device [-position_name {position name}]
```

Arguments

Parameter	Type	Description
position_name	string	Optional argument that specifies the name of a device in the chain.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'position_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'paste_device [-position_name "position_name"]'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example pastes the devices on the clipboard immediately above the device 'MyDevice' in the chain.

```
paste_device -position_name {MyDevice}
```

See Also

- copy_device
- cut_device

12.12. remove_non_actel_device_from_database (Ask a Question)

Description

This Tcl command removes settings for non-Microchip or non-Actel device from the device database.

```
remove_non_actel_device_from_database [-name {device name}]
```

Arguments

Parameter	Type	Description
name	string	Specifies the non-Actel or non-Microchip device name to be removed from the database. You can repeat this argument for multiple devices.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	The device 'device_name' does not exist in the database.
None	Parameter 'name' is missing or has invalid value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_non_actel_device_from_database [-name "name"]+'.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example removes the "F1502AS" device from the database:

```
remove_non_actel_device_from_database -name {F1502AS}
```

See Also

- add_non_actel_device_to_database

12.13. set_bsdl_file (Ask a Question)

Description

This Tcl command sets a BSDL file to a non-Microchip device in the chain. Chain programming mode must have been set. The device must be a non-Microchip device.

```
set_bsdl_file -name {device name} -file {path to the BSDL file}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name.
file	string	Specifies the BSDL file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_bsdl_file -name "name" -file "file"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example sets the BSDL file `./src/top.bsd` to the device 'MyDevice':

```
set_bsdl_file -name {MyDevice} -file {./src/top.bsd}
```

12.14. set_device_ir (Ask a Question)

Description

This Tcl command sets the IR length of a non-Microchip device in the chain. Chain programming mode must be set. The device must be a non-Microchip device.

```
set_device_ir -name {device name} -ir {integer value}
```

Arguments

Parameter	Type	Description
name	string	Specifies the non-Microchip device name.
ir	integer	Specifies the IR length greater than or equal to 2.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'ir' is missing.
None	ir: Invalid argument value: 'value' (expecting integer value).
None	Parameter 'ir' must be greater than or equal to 2.
None	Parameter 'name' has illegal value.
None	The device with name 'device_name' doesn't exist.
None	The device 'device_name' is an Microchip device. The command 'set_device_ir' only applies to non-Microchip devices.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device_ir -name "name" -ir "integer value"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the IR length to '2' for the non-Microchip device 'MyDevice':

```
set_device_ir -name {MyDevice} -ir {2}
```


© 2025 Microchip Technology Inc. and its subsidiaries

12.15. set_device_order (Ask a Question)

Description

This Tcl command sets the order of the devices in the chain to the order specified. Chain programming mode must have been set. Unspecified devices will be at the end of the chain.

```
set_device_order -name {device name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the device name. To specify a new order you must repeat this argument and specify each device name in the order desired. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'name' is missing or has invalid value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device_order [-name "name"]+'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the device order for 'MyDevice1', 'MyDevice2', 'MyDevice3', and 'MyDevice4'.'MyDevice2' is unspecified so it moves to the end of the chain.

```
set_device_order -name  {MyDevice2} -name {MyDevice3} -name {MyDevice1} -name {MyDevice4}
```

12.16. set_device_tck (Ask a Question)

Description

This Tcl command sets the maximum TCK frequency of a non-Microchip device in the chain. Chain programming mode must be set. The device must be a non-Microchip device.

```
set_device_tck -name {name} -tck {tck}
```

Arguments

Parameter	Type	Description
name	string	Specifies the non-Microchip device name.

© 2025 Microchip Technology Inc. and its subsidiaries

set_device_tck (continued)		
Parameter	**Type**	**Description**
tck	decimal	Specifies the maximum TCK frequency (in MHz). TCK frequency be greater than or equal to 1.000 and less than or equal to 100.000.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Required parameter 'tck' is missing.
None	Parameter 'tck' must be greater than or equal to 1.000.
None	Parameter 'tck' must be less than or equal to 100.000.
None	Parameter 'tck' has illegal value.
None	tck: Invalid argument value: '' (expecting decimal value).
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device_tck -name "name" -tck "decimal value"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the maximum TCK frequency of the non-Microchip device 'MyDevice':

```
set_device_tck -name {MyDevice} -tck {2.25}
```

12.17. set_device_type (Ask a Question)

Description

This Tcl command changes the family of a Microchip device in the chain. The device must be a Microchip device.

```
set_device_type [-name {device name}] -type {type}
```

Arguments

Parameter	Type	Description
name	string	Identifies the name of the device you want to change. This parameter is optional.
type	string	Specifies the device family. This parameter is mandatory.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'type' is missing.
None	Parameter 'type' has illegal value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_device_type [-name "name"] -type "type"'.

Supported Families

Supported Families
PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the device 'MyDevice' to the type MPF324.

```
set_device_type -name {MyDevice} -type {MPF324}
```


© 2025 Microchip Technology Inc. and its subsidiaries

13. SmartDebug Tcl Commands (Ask a Question)

13.1. Smart Debug Tcl Commands (Ask a Question)

13.1.1. add_probe_insertion_point (Ask a Question)

Description

This Tcl command adds a probe point to be connected to user-specified I/Os for probe insertion flow. This command will fail if any of the parameters are missing.

Note:
Probe Insertion feature disabled in the SmartDebug Demo and Standalone modes.

```
add_probe_insertion_point -net {net_name} \
                          -driver {driver_name} \
                          -pin {pin_name} \
                          -port {port_name}
```

Arguments

Parameter	Type	Description
net	string	Specify name of the existing net which is added in probe insertion list. This parameter is mandatory.
driver	string	Specify driver name of the net. This parameter is mandatory.
pin	string	Specify Package pin name(that is, I/O to which the net will be routed during probe insertion). This parameter is mandatory.
port	string	Specify user-specified name for the probe insertion. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Port name is already used.
None	Not a valid pin or already used pin.
None	Parameter 'param_name' is not defined. Valid command formatting is 'add_probe_insertion_point [-net "net_name"] \[-driver "driver_name"] \ [-pin "pin_name"] \ [-port "port_name"]'.
None	Probe insertion operations are not supported in Standalone SmartDebug.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example adds a probe point to the probe insertion list:

```
add_probe_insertion_point -net {sw} \
                          -driver {sw_buf/IN:Y} \
                          -pin {Unassigned} \
                          -port {Probe_Insert0}
```

See Also

- remove_probe_insertion_point
- program_probe_insertion

13.1.2. add_to_probe_group (Ask a Question)

Description

This Tcl command adds the specified probe points to the specified probe group.

This command will fail if any of the optins are incorrect.

```
add_to_probe_group -name {Probe name} -group {Group name}
```

Arguments

Parameter	Type	Description
name	string	Specifies one or more probes to add.
group	string	Specifies name of the probe group.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'group' has illegal value.
None	Required parameter 'group' is missing.
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'add_to_probe_group [-name "name"]+ \-group "group name"'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example adds {DFN1_0_Q:DFN1_0/U0:Q} instance to the {probe_group}.

```
add_to_probe_group -name {DFN1_0_Q:DFN1_0/U0:Q} -group {probe_group}
```

See Also

- create_probe_group

© 2025 Microchip Technology Inc. and its subsidiaries

- remove_from_probe_group
- move_to_probe_group

13.1.3. check_flash_memory (Ask a Question)

Description

The command performs diagnostics of the page status and data information as follows: • Page Status – includes ECC2 check of the page status information, write count • Page Data - ECC2 check

```
check_flash_memory [-name { device_name }] \
                   [-startpage { integer_value }] \
                   [-endpage { integer_value }] \
                   [-access { all | status | data }] \
                   [-show { summary | pages }] \
                   [-file { filename }]
```

Arguments

Parameter	Type	Description
name	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
startpage	integer	Startpage value must be an integer. You must specify a –endpage along with this argument.
endpage	integer	Endpage value must be an integer. You must specify a –startpage along with this argument.
access	string	You must set -startpage and -endpage before use. Specifies what NVM information to check: page status, data or both. **All**: Shows the number of pages with corruption status, data corruption and out-of-range write count (default). **Status:** Shows the number of pages with corruption status and the number of pages with out-of-range write count. **Data:** Shows only the number of pages with data corruption.
show	string	This is an optional argument. You must set -startpage and -endpage before use. Specifies output level, as explained in the table below. **Summary:** Displays the summary for all checked pages (default). **Pages**: Displays the check results for each checked page.
file	string	This is an optional argument. You must set -startpage and -endpage before use. Specifies name of output file for memory check.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'show' has illegal value.
None	Parameter 'file' has illegal value.
None	Parameter 'endpage' has illegal value.
None	Missing '-endpage' argument for page range. Specify a page range with both a -startpage and an -endpage argument.
None	Parameter 'startpage' has illegal value.
None	Missing '-startpage' argument for page range. Specify a page range with both a -startpage and an -endpage argument.

© 2025 Microchip Technology Inc. and its subsidiaries

check_flash_memory (continued)	
Error Code	**Description**
None	Parameter 'param_name' is not defined. Valid command formatting is'check_flash_memory [-deviceName "device name"] [-block "integer value"] [-client "client name"] [-startpage "integer value"] [-endpage "integer value"] [-access "all | status | data"] [-show "summary | pages"] [-file "filename"]'.
None	Invalid value for -show: 'show_value'. Value should be 'summary' or 'pages'.
None	endpage: Invalid argument value: 'endpage_value' (expecting integer value).
None	startpage: Invalid argument value: 'startpage_value' (expecting integer value).
None	Invalid value for -access: 'access_value'. Value should be 'all' or 'status' or 'data'.
None	Missing specification for Flash Memory area. Use one of:-client [-block]or-startpage -endpage -block.

Supported Families

SmartFusion 2
IGLOO 2*
RTG4*

Example

This example checks flash memory form pages 0 to 1 and saves their pages to check_flash_memory.txt file:

```
check_flash_memory -startpage 0 -endpage 1 \
                   -file {check_flash_memory.txt} \
                   -show {pages}
```

See Also

- read_flash_memory

13.1.4. close_project (Ask a Question)

Description

This Tcl command closes a SmartDebug project.

```
close_project
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2

© 2025 Microchip Technology Inc. and its subsidiaries

IGLOO 2
RTG4

Example

This command closes the SmartDebug project:

```
close_project
```

See Also

- new_project
- open_project

13.1.5. complete_debug_job (Ask a Question)

Description

Completes the current open job and generates a job status container including cryptographically signed Job Ticket end certifiers and Certificates of Conformance (if enabled) of the programmed devices. It archives ticket data from the HSM database. The resultant Job Status container can be imported into Job Manager and validated using U-HSM. If the job status file is not specified, the information is printed in the log window and no Job Status container is created for subsequent verification.

The HSM Job can be completed only if the number of devices in each HSM ticket has been exhausted. If devices remain, the job can only be terminated by using the `-terminate` option.

Note: This command fails if there are devices left in any HSM ticket and the terminate option is not used.

```
complete_debug_job [-job_status_file "job status file"] [-terminate]
```

Arguments

Parameter	Type	Description
job_status_file	string	Full path to the output Job Status container, which contains End-Job Certifier and CofCs. If not specified, information is printed in the log window.
terminate	flag	Terminates the HSM job even if there are devices left in any HSM ticket. This parameter is optional if the number of devices in all tickets has been exhausted.

Error Codes

Error Code	Description
None	Fpeng error: Chain manager is not set.
None	Parameter 'param_name' is not defined. Valid command formatting is 'complete_debug_job [-job_status_file "job status file"] \ [-terminate]'.

Supported Families

PolarFire
SmartFusion 2
IGLOO 2

Example

This example terminates an HSM job.

```
complete_debug_job -terminate -job_status_file {./MyJobStatusEnd}
```


© 2025 Microchip Technology Inc. and its subsidiaries

13.1.6. create_probe_group (Ask a Question)

Description

This Tcl command creates a new probe group.

```
create_probe_group -name {Group name}
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the new probe group.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'create_probe_group -name "group name"'.
None	Parameter 'name' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example creates new probe group named "my_new_grp" :

```
create_probe_group -name my_new_grp
```

13.1.7. debug_ddr (Ask a Question)

Description

This Tcl command retrieves/gets the training data from the Training IP and displays the status of different stages of training along with the eye width chart.

```
debug_ddr [-ddr_type type_of_ddr] \
          [-data_width width] \
          [-slot memory_slot] \
          [-inst_path ddr_instance_path_from_top] \
          [-frequency frequency]

debug_ddr [-deviceName "device name"] \
          -ddr_type "DDR Type" \
          -data_width "integer value" \
          -slot "DDR Slot" -inst_path \
          "Instace Path from Top" \
          -frequency "decimal value"
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
ddr_type	string	Specifies DDR type. Supported DDR types are: DDR4/DDR3/LPDDR4.
data_width	integer	Specify data width. Supported data widths are 16, 32, and 64.
slot	string	Slot that is used for the memory (for example, NORTH_NE/ NORTH_NW).
inst_path	string	The instance path is from top module.
frequency	double	Specifies frequency in MHz.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'ddr_type' is missing.
None	Required parameter 'data_width' is missing.
None	data_width: Invalid argument value: 'data' (expecting integer value).
None	frequency: Invalid argument value: expecting decimal value.
None	DDR Debug: Valid values for "-ddr_type" parameter are DDR3, DDR4, LPDDR3. Provided value is a Error: .
None	DDR Debug: Valid values for "-ddr_width" parameter for DDR3 are 16, 32 and 64.
None	Parameter 'param_name' is not defined. Valid command formatting is 'debug_ddr [-deviceName "device name"] -ddr_type "DDR Type" -data_width "integer value" -slot "DDR Slot" -inst_path "Instace Path from Top" -frequency "decimal value" '.

Supported Families

PolarFire
PolarFire SoC

Example

This example gets the training data from the Training IP and displays the status of different stages of training along with the eye width chart:

```
debug_ddr -ddr_type {DDR4} -data_width 32 -slot {NORTH_NE} \
          -inst_path {PF_DDR4_C0_0} -frequency 800.00
```

See Also

- ddr_read
- ddr_write

13.1.8. debug_mss_ddr (Ask a Question)

Description

This command retrieves the training data from the memory subsystem, which executes the initialization and training sequence, and displays the status of various training stages.

```
debug_mss_ddr [-deviceName "device name"]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Specifies the device name.

Return Type	Description
None	None

Supported Families

PolarFire SoC

Example

This example gets the training data from the memory subsystem and displays the status of different stages of training:

```
debug_mss_ddr
```

13.1.9. debug_iod (Ask a Question)

Description

This Tcl command gets the training data from the CORERXIODBITALIGN IP and displays Eye Width and Sampling Edge.

```
debug_iod [-deviceName "device name"] \
          -iod_type {RX_DDRX_B_G_DYN/RX_DDRX_B_R_DYN/RX_DDRX_B_G_FDYN} \
          -inst_path {PF_IOD_GENERIC_RX instance path from Top}
```

Important: CORERXIODBITALIGN IP must have the output debug pins either connected or promoted to the top for SmartDebug to detect and identify the debug signals. The following pins must be configured for CORERXIODBITALIGN for IOD training to succeed and to be able to perform tasks such as reading delay taps, left taps, right taps, and bit align errors:

- RX_BIT_ALIGN_LEFT_WIN
- RX_BIT_ALIGN_RGHT_WIN
- BIT_ALGN_ERR
- DEM_BIT_ALGN_TAPDLY
- BIT_ALGN_DONE
- BIT_ALGN_START

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
iod_type	string	Specifie iod type. Valid types are: RX_DDRX_B_G_DYN, RX_DDRX_B_R_DYN and RX_DDRX_B_G_FDYN.
inst_path	string	PF_IOD_GENERIC_RX instance path from Top.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'iod_type' is missing.
None	Parameter 'iod_type' has illegal value.
None	Required parameter 'inst_path' is missing.
None	Parameter 'inst_path' has illegal value.
None	IOD Debug: Provide the path of IOD instace for top level module in the design valid for "-inst_path" parameter.

Supported Families

PolarFire
PolarFire SoC

Example

Get training data from {PF_IOD_GENERIC_RX_C1_0} instance.

```
debug_iod -iod_type {RX_DDRX_B_G_DYN} -inst_path {PF_IOD_GENERIC_RX_C1_0}
```

See Also

- debug_ddr

13.1.10. ddr_read (Ask a Question)

Description

This tcl command reads the value of specified configuration registers pertaining to the DDR memory controller (MDDR/FDDR).

```
ddr_read -deviceName "deive name" \
         -block {DDR name} \
         -name {register name}
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
block	string	Specify block name: fddr | mddr | east_fddr | west_fddr. • Specifies which DDR configurator is used in the Libero design. • SmartFusion 2 and IGLOO 2 - fddr and mddr 56. • RTG4 - east_fddr and west_fddr.
name	string	• Specifies which configuration registers need to be read. • A complete list of registers is available in the DDR Interfaces User Guides for the respective families.

Return Type	Description
Returns 16-bit hexadecimal value.	The result of the command in the example below will be: Register Name: DDRC_DYN_REFRESH_1_CR Value: 0x1234 “ddr_read” command succeeded.

Error Codes

Error Code	Description
None	Required parameter 'block' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

ddr_read (continued)	
Error Code	**Description**
None	Parameter 'block' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'ddr_read [-deviceName "device name"] -block "DDR Block Name" -name "DDR Resgister Name"'.

Supported Families

SmartFusion 2
IGLOO 2
RTG4

Example

Read DDR Controller register DDRC_DYN_REFRESH_1_CR for a configured FDDR block on a SmartFusion 2 or IGLOO 2 device:

```
ddr_read -block fddr -name DDRC_DYN_REFRESH_1_CR
```

See Also

- ddr_write

13.1.11. ddr_write (Ask a Question)

Description

This tcl command writes the value of specified configuration registers pertaining to the DDR memory controller (MDDR/FDDR).

```
ddr_write [-deviceName "device name"] \
          -block {ddr name} \
          -name {register name} \
          -value {hexadecimal value}
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
block	string	Specify blcok name: fddr | mddr | east_fddr | west_fddr. • Specifies which DDR configurator is used in the Libero design. • SmartFusion 2 and IGLOO 2 - fddr and mddr • RTG4 - east_fddr and west_fddr.
name	string	• Specifies which configuration registers need to be read. • A complete list of registers is available in the DDR Interfaces User Guides for the respective families.
value	hexadecimal	• Specifies the value to be written into the specified register of a given block. • Hex_value in the form of “0x12FA”.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'value' has illegal value.
None	Required parameter 'value' is missing.
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'block' has illegal value.
None	Required parameter 'block' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'ddr_write [-deviceName "device name"] -block "DDR Block Name" -name "DDR Resgister Name" -value "DDR register value"'.

Supported Families

SmartFusion 2
IGLOO 2
RTG4*

Example

Write a 16-bit value DDR Controller register DDRC_DYN_REFRESH_1_CR for a configured FDDR block on a SmartFusion 2 or IGLOO 2 device:

```
ddr_write -block fddr -name DDRC_DYN_REFRESH_1_CR -value 0x123f
```

See Also

- ddr_read

13.1.12. delete_active_probe (Ask a Question)

Description

This Tcl command deletes either all or the selected active probes.

Note:
You cannot delete an individual probe from the Probe Bus.

```
delete_active_probe -deviceName "device name" -all | -name {probe name}
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
all	None	Deletes all active probe names.
name	string	Deletes the selected probe names.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'name' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

1. This example deletes all active probe names.

```
delete_active_probe -all
```

2. This example deletes the selected "out[5]:out[5]:Q" and "my_grp1.out[1]:out[1]:Q" active probe names;

```
delete_active_probe -name out[5]:out[5]:Q \
                    -name my_grp1.out[1]:out[1]:Q
```

3. This example deletes the group, bus and their members.

```
delete_active_probe -name my_grp1 \
                    -name my_busT
```

See Also

- select_active_probe
- create_probe_group

13.1.13. load_live_probe_list* (Ask a Question)

Description

This Tcl command loads the list of live probes from the file(*.txt).

```
load_live_probe_list [-deviceName "device name"] -file "filename"
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
file	string	Specify path and the name of input file(*.txt). This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'load_live_probe_list [-deviceName "device name"] \ -file "filename"'.

Supported Families

PolarFire
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion 2
IGLOO 2
RTG4

Example

The following example loads M3T device live probes list from live_probe_list.txt file. Text file which has the probes list saved from previous SmartDebug save action.

```
save_live_probe_list -file {./live_probe_list.txt} load_live_probe_list
                -deviceName {M3T} -file {./live_probe_list.txt}
```

See Also

- save_live_probe_list

13.1.14. eye_monitor_power (Ask a Question)

Description

This tcl command switches on and off power eye monitor.

```
eye_monitor_power [-deviceName "device name"] \
                  -switch {on | off} \
                  -lane {physical lane name}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
switch	string	This argument specifies if the eye monitor is on or off.
lane	string	Specify the physical lane instance name.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Eye Monitor Power On/Off: Lane name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Parameter 'lane' has illegal value.
None	Parameter 'switch' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'eye_monitor_power [-deviceName "device name"] [-switch "Eye Monitor Power"] [-lane "Physical Lane Name"]'.

Supported Families

PolarFire
PolarFire SoC

Example

This example turns on the eye monitor.

```
eye_monitor_power -switch {on} -lane {Q0_LANE0}
```


© 2025 Microchip Technology Inc. and its subsidiaries

This example turns off the eye monitor.

```
eye_monitor_power -switch {off} -lane {Q0_LANE0}
```

13.1.15. event_counter (Ask a Question)

Description

This Tcl command runs on signals that are assigned to Channel A through the Live Probe feature and displays the total events.

It is run after setting the live probe signal to channel A. The user specifies the duration to run the event_counter command.

```
event_counter -run | -stop -after {duration in seconds}
```

Arguments

Parameter	Type	Description
run	none	Run event counter.
stop	none	Stop event counter. This parameter must be specified with the -after parameter.
after	integer	Specify duration in seconds to stop event_counter. This argument is required when –stop argument is specified.

Return Type	Description
string	Displays the total events with value-property format.

Error Codes

Error Code	Description
None	Missing argument. Must specify '-run' or '-stop'.
None	Must specify time by using the argument '-after'.
None	after: Invalid argument value: 'value' (expecting integer value).
None	No signal assigned to channel A.
None	Parameter 'param_name' is not defined. Valid command formatting is'event_counter [-deviceName "device name"] [-run "TRUE | FALSE"] [-stop "TRUE | FALSE"] [-after "integer value"]'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

The following example assigns 'Q_c:DFN1_0:Q' signal to Channel A, runts event counter with the 5 delay seconds to stop:

```
set_live_probe -probeA {Q_c:DFN1_0:Q}
event_counter -run
event_counter -stop -after 5
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- fhb_control
- run_frequency_monitor
- set_live_probe

13.1.16. execute_dfe_calibration (Ask a Question)

Description

This Tcl command executes calibration. There are two types of calibration DFE (decision feedback equalizer) and CTLE (continuous time linear equalizer).

```
execute_dfe_calibration [-deviceName "device name"] \
                        -lane {physical location name} \
                        -full_calibration {0 | 1}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
lane	string	Specify the physical lane instance name.
full_calibration	boolean	This parameter specifies what kind of calibration you want to execute. • 1 - execute full calibration both DFE and CTLE Calibrations. • 0 - execute DFE Calibration.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Execute DFE Calibration: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name
None	Parameter 'lane' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'execute_dfe_calibration [-deviceName "device name"] [-lane "Physical Lane Name"] [-full_calibration "TRUE | FALSE"]'.

Supported Families

PolarFire
PolarFire SoC

Example

This example executes the dfe calibration for lane "Q0_Lane0".

```
execute_dfe_calibration -lane {Q0_Lane0} -full_calibration 1
```


© 2025 Microchip Technology Inc. and its subsidiaries

13.1.17. export_ddr_training_data (Ask a Question)

Description

This Tcl command exports the training data that it reads from the device into a simple text file, which helps users to compare data between multiple runs by exporting the training data.

```
export_ddr_training_data [-file "file name"]
```

Arguments

Parameter	Type	Description
file	string	Specifies the file name with file path to save the exported data.

Supported Families

PolarFire

PolarFire SoC

Example

This example exports the DDR training data to `D:\exportedData.txt`.

```
export_ddr_training_data -file D:\exportedData.txt
```

13.1.18. export_mss_ddr_training_data (Ask a Question)

Description

This Tcl command exports the training data read from device to the specified file.

```
export_mss_ddr_training_data [-file "file name"]
```

Arguments

Parameter	Type	Description
file	string	Specifies the file name with file path to export the MSS DDR I/O margin training results.

Supported Families

PolarFire SoC

Example

This example exports the MSS DDR training data to `D:\exportedData.txt`.

```
export_mss_ddr_training_data -file D:/exportedData.txt
```

13.1.19. fhb_control (Ask a Question)

Description

This Tcl command provides FPGA Hardware Breakpoint (FHB) capability for SmartDebug.

```
fhb_control [-deviceName "device name"] -halt \
            -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] -run \
            -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] -step {number of steps} \
            -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] -reset \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
            -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] -arm_trigger \
-trigger_signal {probe point signal to trigger the HALT operation} \
-trigger_edge_select {rising} -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] \
            [-capture_waveform "integer value"] \
            [-vcd_file "Waveform File"]

fhb_control [-deviceName "device name"] \
            [-clock_domain_status] \
            -clock_domain {clock domain name(s)/all}

fhb_control [-deviceName "device name"] \
            [-capture_waveform "integer value"] \
            [-vcd_file "Waveform File"]
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
clock_domain	string	Specifies clock domain names to halt|run|step|reset|disarm|. Can be single or multiple clock domains, halted in order specified by user.
halt	none	Specifies to halt the clock.
run	none	Specifies to run the clock.
step	integer	Specifies to step the clock "number of steps" times. Minimum value is 1.
reset	none	Specifies to reset FHB configuration for the specified clock domain.
arm	none	Specifies to arm FHB configuration for the specified clock domain.
trigger_signal	string	probe point signal to trigger the HALT operation} Set the trigger signal to arm the FHBs.
trigger_edge_select	string	Specifies the trigger signal edge to arm the FHBs. FHBs will be armed on rising edge of trigger signal.
delay	integer	Sepcifies the value between 0 to 255 of delay cycles before halt.
clock_domain_status	none	Specifies to read and display status of specified clock domain(s). Can be single or multiple clock domains.
disarm	none	Specifies to disarm FHB configuration for the specified clock domain.
capture_waveform	integer	Specifies to capture waveform of all the added signals to active probes in the specified clock domain for "number of steps".
vcd_file	string	Target file to save the data and see the waveform.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Fhb control: One of the following parameters should be set:-halt, -run, -step, -arm, -disarm, -clock_domain_status, -reset and -capture_waveform
None	Fhb control: Clock Domain all not found in the design.
None	step: Invalid argument value: 'step_value' (expecting integer value).
None	Fhb control: Minimum value of -step should be 1.
None	Fhb control: -trigger_edge_select parameter value can only be rising.

© 2025 Microchip Technology Inc. and its subsidiaries

fhb_control (continued)	
Error Code	**Description**
None	capture_waveform: Invalid argument value: 'capture_waveform value' (expecting integer value).
None	Fhb control: Minimum value of -capture_waveform should be 1.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

```
fhb_control -halt -clock_domain {“FCCC_0/GL0_INST “ “FCCC_0/GL1_INST” }
fhb_control -run -clock_domain {“FCCC_0/GL0_INST “ “FCCC_0/GL1_INST” }
fhb_control -step -clock_domain {“FCCC_0/GL0_INST “ “FCCC_0/GL1_INST” }
fhb_control -reset -clock_domain {“FCCC_0/GL0_INST “ “FCCC_0/GL1_INST” }
fhb_control -arm_trigger -trigger_signal {q_0_c[14]:count_1_q[14]:Q} \
            -trigger_edge_select {rising} -delay 0 \
            -clock_domain {"FCCC_0/GL0_INST"}
fhb_control -disarm_trigger -trigger_signal {q_0_c[14]:count_1_q[14]:Q} \
            -trigger_edge_select {rising} -delay 0 \
            -clock_domain {"FCCC_0/GL0_INST"}
fhb_control -capture_waveform {10} \
            -vcd_file {D:/wvf_location/waveform.vcd}
fhb_control -clock_domain_status \
-clock_domain { "FCCC_0/GL0_INST" "FCCC_0/GL1_INST" "FCCC_0/GL2_INST" }
```

See Also

- event_counter
- run_frequency_monitor

13.1.20. get_programmer_info (Ask a Question)

Description

This Tcl command lists the IDs of all FlashPro programmers connected to the computer. Command will fail if programmers are not connected.

```
get_programmer_info
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
List of IDs	Returns the list of IDs of all FlashPro programmers connected to the computer.

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion 2
IGLOO 2
RTG4

Example

Get the list of all connected programmers IDs:

```
set IDs [get_programmer_info];
puts "IDs of connected programmers: $IDs";
```

See Also

- read_device_status
- read_id_code

13.1.21. get_user_clock_frequencies (Ask a Question)

Description

This Tcl command calculates the user clock frequencies.

Note:
Before this commands usage user has to enable FHB(FPGA Hardware Breakpoint) controller from Libero project settings.

```
get_user_clock_frequencies [-deviceName "device name"]
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.

Return Type	Description
String	Displays user clock frequency in megahertz (MHz).

Error Codes

Error Code	Description
None	Fhb control: The design does not have FHB enabled.
None	Parameter 'param_name' is not defined. Valid command formatting is. 'get_user_clock_frequencies [-deviceName "device name"] '.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

Get 'M4' device user clock frequensy:

```
get_user_clock_frequencies -deviceName "M4"
```

Output:

© 2025 Microchip Technology Inc. and its subsidiaries

User clock frequency - clocking_0\/PF_out0 value = 100.07 MHz

User clock frequency - clocking_0\/PF_out1 value = 132.02 MHz

See Also

- event_counter
- run_frequency_monitor

13.1.22. import_ddc_file (Ask a Question)

Description

This is a Standalone SmartDebug command. Enables you to import DDC file (created through Export SmartDebug Data in Libero) into the debug project.

```
import_ddc_file -import_ddc "DDC file" -device_name "device name"
```

Arguments

Parameter	Type	Description
import_ddc	string	Specify path to the DDC file. This parameter is mandatory.
device_name	string	Specify the device name. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'import_ddc' is missing.
None	Required parameter 'device_name' is missing.
None	Failed to import DDC file '*.ddc'. There is no device 'device_name' in the current JTAG chain.
None	Parameter 'param_name' is not defined. Valid command formatting is 'import_ddc_file -import_ddc "DDC file" -device_name "device name"'

Supported Families

PolarFire
SmartFusion 2
IGLOO 2
RTG4

Example

This example importes DDC file to the debug project:

```
import_ddc_file -import_ddc {./src/top.ddc} -device_name {MPF250T_ES}
```

See Also

- new_project

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.23. load_active_probe_list (Ask a Question)

Description

This Tcl command loads the list of probes from the file.

```
load_active_probe_list [-deviceName "device name"] \
                       -file "path to the file"
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
file	string	The input file location.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'load_active_probe_list [-deviceName "device name"] -file "filename"'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example loads the active probe list from “./my_probes.txt” file.

```
load_active_probe_list -file “./my_probes.txt”
```

See Also

- delete_active_probe
- read_active_probe
- save_active_probe_list
- select_active_probe
- write_active_probe

13.1.24. load_SI_design_defaults (Ask a Question)

Description

This Tcl command loads the Signal Integrity parameter options for the selected lane instance.

```
load_SI_design_defaults [-deviceName "device name"] \
                        [-lane "Lane Instance Name"] \
                        [-all_lanes "TRUE | FALSE"]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
lane	string	Specify the physical lane instance name.
all_lanes	boolean	If you want to load design defaults for all lanes, then give "TRUE" to the argument, else "FALSE".

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Signal Integrity: Must not specify both '-lane' and '-all_lanes' command arguments.
None	Signal Integrity: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Parameter 'lane' has illegal value.
None	Signal Integrity: Must specify one of '-lane' or '-all_lanes' command arguments.
None	Parameter 'param_name' is not defined. Valid command formatting is'load_SI_design_defaults [-deviceName "device name"] [-lane "Lane Instance Name"] [-all_lanes "TRUE | FALSE"]'.

Supported Families

PolarFire
PoarFire SoC

Example

This example loads design defaults for lane "Q0_LANE0"

```
load_SI_design_defaults -lane {Q0_LANE0}
```

See Also

- signal_integrity_write
- signal_integrity_import
- signal_integrity_export

13.1.25. loopback_mode (Ask a Question)

Description

This Tcl command applies loopback mode to a specified lane.

```
loopback_mode -lane {Physical Lane name} -apply -type {loopback_type}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
lane	string	Specify the physical location of the lane.

© 2025 Microchip Technology Inc. and its subsidiaries

loopback_mode (continued)		
Parameter	**Type**	**Description**
apply	none	Apply specified loopback to specified lane.
type	string	Specify the loopback type to apply. Type valid values are: EQ-NearEnd, EQ-FarEnd, CDRFarEnd and NoLpbk.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Loopback Mode: Must specify '-apply' argument.
None	Loopback Mode: Transceiver physical Lane Name must be specified.
None	Parameter 'type' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'loopback_mode [-deviceName "device name"] [-lane "Physical Lane Name"] [-apply "TRUE | FALSE"] [-type "Loopback type"]'.

Supported Families

PolarFire
PolarFire SoC

Example

This examples applies EQ-FarEnd|EQ-NearEnd|CDRFarEnd|NoLpbk loopback mode to a "Q0_LANE0" lane.

```
loopback_mode -lane {Q0_LANE0} -apply -type {EQ-FarEnd}
loopback_mode -lane {Q0_LANE0} -apply -type {EQ-NearEnd}
loopback_mode -lane {Q0_LANE0} -apply -type {CDRFarEnd}
loopback_mode -lane {Q0_LANE0} -apply -type {NoLpbk}
```

See Also

- loopback_test
- smartbert_test
- prbs_test

13.1.26. loopback_test (Ask a Question)

Description

This Tcl command used to start and stop the loopback tests. Loopback data stream patterns are generated and checked by the internal SerDes block. These are used to self-test signal integrity of the device. You can switch the device through predefined tests.

Note:
loopback_test is renamed as loopback_mode in G5.

```
loopback_test [-deviceName "device name"] [-start] \
              -serdes "integer value" -lane "integer value" \
              -type "Loopback Type"
loopback_test [-deviceName "device name"] [-stop] \
              -serdes "integer value" -lane "integer value"
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Specifies device name. This parameter is optional if only one device is available in the current configuration or set for debug.
start	none	Starts the loopback test.
stop	none	Stops the loopback test.
SerDes	integer	Specifies SerDes block number. Must be between 0 and 4 and varies between dies.
lane	integer	Specifies SerDes lane number. Must be between 0 and 3.
type	string	Specifies the loopback test type. Loopback test types are: Must be meso (PCS Far End PMA RX to TX Loopback), plesio and parallel.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'loopback_test [-deviceName "device name"] [-start "TRUE | FALSE"] [-stop "TRUE | FALSE"] -serdes "integer value" -lane "integer value" [-type "Loopback type"]'.
None	Required parameter 'serdes' is missing.
None	Required parameter 'lane' is missing.
None	serdes: Invalid argument value: 'serdes_value' (expecting integer value).
None	lane: Invalid argument value: 'lane_value' (expecting integer value).
None	Loopback test: IDCode verify failed.
None	Loopback test: Invalid loopback type specified.

Supported Families

IGLOO 2
RTG4

Example

Start and stop loopback tests.

```
loopback_test -start -serdes 1 -lane 1 -type meso
loopback_test -start -serdes 0 -lane 0 -type plesio
loopback_test -start -serdes 1 -lane 2 -type parallel
loopback_test -stop -serdes 1 -lane 2
```

See Also

- loopback_mode
- prbs_test
- smartbert_test

13.1.27. move_to_probe_group (Ask a Question)

Description

This Tcl command moves the specified probe points to the specified probe group.

© 2025 Microchip Technology Inc. and its subsidiaries

Note:
Probe points related to a bus cannot be moved to another group.

```
move_to_probe_group -name {probe name} -group {group name}
```

Arguments

Parameter	Type	Description
name	string	Specifies one or more probes to move.
group	string	Specifies name of the probe group.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'group' has illegal value.
None	Required parameter 'group' is missing.
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'move_to_probe_group [-name "name"]+ \-group "group name"'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example moves {out[5]:out[5]:Q} and {grp1.out[3]:out[3]:Q} probes to the {my_grp2}:

```
move_to_probe_group -name {out[5]:out[5]:Q} \
                    -name {grp1.out[3]:out[3]:Q} \
                    -group {my_grp2}
```

See Also

- create_probe_group
- add_to_probe_group

13.1.28. mss_add_register (Ask a Question)

Description

This tcl command records whenever registers are selected in the tool.

```
mss_add_register -reg_name {register name} -reset {0|1}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
reg_name	string	Specifies name of the register.
reset	boolean	When set to 1, all the previously selected registers will be cleared from the list and the new ones will be added. If 0, then adds the register to the old list.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Enter register names and reset value. Params are "reg_name" and "reset".
None	Parameter 'reg_name' has illegal value.
None	register is not found in the valid list provided in pfsoc_regmap.htm file.
None	Parameter 'reset' has illegal value.
None	reset: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'param_name' is not defined. Valid command formatting is'mss_add_register [-deviceName "device name"] \[-reg_name "Add Register Names"]* \[-reset "TRUE | FALSE"]'.

Supported Families

PolarFire SoC

Example

This example adds the following registers into mss register list.

```
mss_add_register \
-reg_name {ATHENA:CSRMAIN} \
-reg_name {ATHENA:CSRMERRS} \
-reg_name {ATHENA:CSRMERRT0} \
-reg_name {ATHENA:CSRMERRT1} \
-reg_name {ATHENA:CSRMERRV} \
-reset 0
```

See Also

- mss_read_register
- mss_write_register
- mss_export_register

13.1.29. mss_read_register (Ask a Question)

Description

This tcl command reads all selected registers by their register names and displays the values. The values are in hexadecimal format.

```
mss_read_register [-deviceName "device name"]  \
                  [-reg_name {register name}"] \
                  [-axiQos "interger value"] \
                  [-axiProt "integer value"] \
                  [-axiCache "integer value"] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                    [-axiLock "integer value"] \
                    [-silent "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
reg_name	string	This is an optional argument, that specifies the name of the register according to the hierarchy seen in the UI separated by colon.
axiQos	integer	This is an optional parameter that specifies the value of the attribute QoS on Axi interface.
axiCache	integer	This is an optional parameter that specifies the value of the attribute Cache on Axi interface.
axiProt	integer	This is an optional parameter that specifies the value of the attribute Protocol on Axi interface.
axiLock	integer	This is an optional parameter that specifies the value of the attribute Lock on Axi interface.

Return Type	Description
Hexadecimal	Reads all selected registers by their register names and displays the values.

Error Codes

Error Code	Description
None	Register is not found in the valid list provided in pfsoc_regmap.htm file.
None	Parameter 'reg_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'mss_read_register [-deviceName "device name"] \[-reg_name "Register Name"]* \[-axiQos "integer value"] \[-axiProt "integer value"] \[-axiCache "integer value"] \[-axiLock "integer value"] \[-silent "TRUE | FALSE"]'.

Supported Families

PolarFire SoC

Example

Read register with parameters. Read with {ATHENA:CSRMAIN} and {ATHENA:CSRMERRS} registers:

```
mss_read_register \
    -reg_name {ATHENA:CSRMAIN} \
    -reg_name {ATHENA:CSRMERRS}
```

Register read without parameters (reads all the selected registers that are added using mss_add_register command):

```
mss_read_register
```

See Also

- mss_write_register
- mss_export_register
- mss_add_register

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.30. mss_write_register (Ask a Question)

Description

This tcl command writes value to the selected registers.

If there is a conflict in the values where full register write values and also field values of the same register, then a full register write is executed and field write will be ignored.

```
mss_write_register [-deviceName "device name"] \[-reg_name "Register Name"]* \[-value
"Register write value"]* \[-axiQos "integer value"] \[-axiProt "integer value"] \[-axiCache
"integer value"] \[-axiLock "integer value"] \[-silent "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
reg_name	string	Name of the register according to the hierarchy seen in the UI separated by colon. It can also contain register field separated by colon.
value	hexadecimal	Hexadecimal value - ranges from 1-bit to 64-bits.
axiQos	integer	This is an optional parameter that specifies the value of the attribute QoS on Axi interface.
axiCache	integer	This is an optional parameter that specifies the value of the attribute Cache on Axi interface.
axiProt	integer	This is an optional parameter that specifies the value of the attribute Protocol on Axi interface.
axiLock	integer	This is an optional parameter that specifies the value of the attribute Lock on Axi interface.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Can not write into a read-only register or a field.
None	Invalid register name specified.
None	Parameter 'reg_name' has illegal value.
None	register is not found in the valid list provided in pfsoc_regmap.htm file.
None	Invalid register value specified.
None	Parameter 'value' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'mss_write_register [-deviceName "device name"] \[-reg_name "Register Name"]* \[-value "Register write value"]* \[-axiQos "integer value"] \[-axiProt "integer value"] \[-axiCache "integer value"] \[-axiLock "integer value"] \[-silent "TRUE | FALSE"]'

Supported Families

PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This example writes the values into the registers.

```
mss_write_register \
    -reg_name {MMUART0_LO:RBR} -value {0x123} \
    -reg_name {MMUART0_LO:IER} -value {0xFFFFFFFF} \
    -reg_name {MMUART0_LO:IIR:IIR} -value {0x3}
```

See Also

- mss_read_register
- mss_export_register
- mss_add_register

13.1.31. mss_import_register (Ask a Question)

Description

This Tcl command imports the register list from a `*.csv` file generated by register export operation.

```
mss_import_register \
           -file_name {absolute or relative path to the *.csv file} \
           [-deviceName "device name"]
```

Arguments

Parameter	Type	Description
file_name	string	Specifies the absolute and relative path to the `*.csv` file.
deviceName	string	Specify the device name. This parameter is optional, if only one device is available in the current configuration.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Register Access: Must specify '-file_name'.
None	Required parameter 'file_name' is missing.
None	Parameter 'file_name' has illegal value.
None	Register Access: file specified for import must have `.csv` extension.
None	Unable to write to the file: `/prj_path/imported_file.csv`
None	Parameter 'param_name' is not defined. Valid command formatting is 'mss_import_register [-deviceName "device name"] -file_name "filename" '.

Supported Families

PolarFire SoC

Example

This example imports the register list from the `{./MssRegisters_SmartDebug.csv}` file.

```
mss_import_register -file_name {./MssRegisters_SmartDebug.csv}
```

See Also

- mss_add_register

© 2025 Microchip Technology Inc. and its subsidiaries

- mss_read_register
- mss_write_register

13.1.32. mss_export_register (Ask a Question)

Description

This command reads the selected registers by mss_add_register command and save the content to a `*.csv` file.

```
mss_export_register -file_name {path the file} [-all]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
file_name	string	Specifies the path to the file, where the command saves the exported data.
all	none	This is an optional parameter that is used to export all registers shown in the hierarchy.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Register Access: Must specify '-file_name.
None	Parameter 'file_name' has illegal value.
None	Register Access: file specified for Export must have .csv extension.
None	Cannot export register information to the file: List of selected registers is empty.
None	Parameter 'param_name' is not defined. Valid command formatting is 'mss_export_register [-deviceName "device name"] [-file_name "File Name"] [-all "TRUE | FALSE"]'.

Supported Families

PolarFire SoC

Example

This example exports all mss registers into {./mss_exp.csv} file.

```
mss_export_register -file_name {./mss_exp.csv} -all
```

See Also

- mss_add_register
- mss_read_register
- mss_write_register

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.33. new_project (Ask a Question)

Description

This Tcl command creates a SmartDebug project that enables the user to debug the design. Either DDC can be used to create a project or construct automatically with DDC.

```
new_project [-location {project location}] \
        [-name {name of the new SmarDebug project}] \
        -import_ddc {path to the DDC file} \
        [-auto_construct {"TRUE"|"FALSE"}] \
        [-set_programmer {set debug programmer}]
```

Arguments

Parameter	Type	Description
location	string	Specifiy the location of the project where user wants to create the project. Must not be an existing directory. This parameter is optional. If -location option is omitted, the tool creates a new project in the current directory.
name	string	Specifiy name of the new project. This parameter is optional. This parameter is optional. If this option is omitted, the tool creates a new project with 'untitled_project' name.
import_ddc	string	Specifiy the path to the DDC(Design Debug Data Container) file exported from Libero to be imported. Set empty parameter value if -auto_construct is 1.
auto_construct	boolean	Valid values are:TRUE or 1, FALSE or 0(default). Specify 1 or TRUE if you want to create new project importing DDC file otherwise specify 0 or FALSE. This parameter is optional.
set_programmer	string	Set ID code of the programmer. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'new_project [-location "project folder"] [-name "name"] [-import_ddc "DDC file"] [-auto_construct "TRUE | FALSE"] [-set_programmer "set debug programmer"].

.

Supported Families

PolarFire
SmartFusion 2
IGLOO 2
RTG4

Example

Create new project using Standalone SmartDebug:

```
new_project -location {/exprj} \
        -name {exprj} \
        -import_ddc {./src/top.ddc} \
        -auto_construct 0 \
        -set_programmer {AF01QVEAF}
```


© 2025 Microchip Technology Inc. and its subsidiaries

See Also

- import_ddc_file

13.1.34. open_project (Ask a Question)

Description

This Tcl command opens an existing SmartDebug project (*.dprj).

```
open_project -project {relative or absolute path and name of the project file}
```

Arguments

Parameter	Type	Description
project	string	Specify relative or absolute path to the *.dprj file. This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'project' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'open_project -project "file"'

Supported Families

PolarFire
SmartFusion 2
IGLOO 2
RTG4

Example

This command opens the 'SDPrj.dprj' project from the SDProject directory:

```
open_project -project {./SDProject/SDPrj.dprj}
```

See Also

- new_project

13.1.35. optimize_dfe (Ask a Question)

Description

This Tcl command supports the Optimize DFE (decision feedback equalizer) feature in SmartDebug.

```
optimize_dfe [-deviceName "device name"] \
             -dfe_algorithm {type of dfe algorithm} \
             -lane {lane(s) configured in the design}
```

Arguments

Parameter	Type	Description
deviceName	script	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.

© 2025 Microchip Technology Inc. and its subsidiaries

optimize_dfe (continued)		
Parameter	**Type**	**Description**
dfe_algorithm	script	This command executes Dfe Algorithm with type of dfe algorithm and lanes as input. Algorithm selection has two options: software_based -executes DfeSs.tcl script xcvr_based -executes internal Dfe Auto Calibration. This argument is mandatory.
lane	script	List of lane(s) configured in the design. This argument is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'optimize_dfe [-deviceName "device name"] -lane "[Physical Lane Name]+" -dfe_algorithm "Dfe Algorithm Selection"'.
None	Parameter 'deviceName' has illegal value.
None	Parameter 'lane' has illegal value.
None	Required parameter 'lane' is missing.
None	Required parameter 'dfe_algorithm' is missing.
None	Parameter 'dfe_algorithm' has illegal value.
None	Optimize DFE: dfe_algorithm has invalid option. Possible options: software_based, xcvr_based.
None	Execute DFE Calibration: Execute DFE calibration falied.
None	Optimize DFE: Transceiver Physical Lanes Q1_LANE0 are configured in CDR Mode.XCVR_BASED Dfe Algorithm is valid for DFE configured lanes only.

Supported Families

PolarFire
PolarFire SoC*

Example

This example oftimizes dfe for lane "Q2_LANE0" using software_based algorithm.

```
optimize_dfe -lane {"Q2_LANE0"} -dfe_algorithm {software_based}
```

This example oftimizes dfe for lane "Q2_LANE0" using xcvr_based algorithm.

```
optimize_dfe -lane {"Q2_LANE0"} -dfe_algorithm {xcvr_based}
```

This example oftimizes dfe for lane "Q2_LANE0" and “Q0_LANE0” using xcvr_based algorithm.

```
optimize_dfe -lane {"Q2_LANE0" “Q0_LANE0”} -dfe_algorithm {xcvr_based}
```

13.1.36. optimize_receiver (Ask a Question)

Description

This Tcl command allows you to optimize the DFE (decision feedback equalizer) coefficients and/or CTLE (continuous time linear equalizer) settings for the selected lanes, depending on the receiver mode. For CDR mode receivers, the CTLE settings and/or DFE coefficients can be optimized. For DFE mode receivers, the CTLE settings and DFE coefficients can be optimized.

© 2025 Microchip Technology Inc. and its subsidiaries

Important: This feature is available for MPF300T, MPF100T, MPF200T, MPF500T, MPFS250T, MPFS025T, and MPFS095T devices.

```
optimize_receiver [-deviceName "device name"] -lane {physical lane name} [-
force_dfe_calibration {1/0/TRUE/FALSE}]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration or a device has already been selected using the set_debug_device command.
lane	string	Specifies the physical lane instance name.
force_dfe_calibration	boolean	Optional parameter to run DFE calibration on CDR mode receivers. When set to false, it disables running DFE calibration and DFE coefficients are reset.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Optimize RECEIVER: The DFE Calibration process on lane Q1_LANE0 failed and the Calibration process timed out.
None	Optimize RECEIVER: Transceiver Physical Lane LANE_NAME not found in the design
None	Required parameter 'lane' is missing.
None	Parameter 'lane' is missing or has an invalid value.
None	Parameter 'lane' has an illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'optimize_receiver [-deviceName "device name"] -lane "[Physical Lane Name]+"'.

Supported Families

PolarFire

PolarFire SoC

Example

This example optimizes receiver for the "Q0_LANE0" lane:

```
optimize_receiver -lane {Q0_LANE0} - force_dfe_calibration 1
```

See Also

- optimize_dfe

13.1.37. pcie_config_space (Ask a Question)

Description

This Tcl command displays the value of the entered parameter in the SmartDebug log window and returns the register:field value to the Tcl.

```
pcie_config_space -pcie_logical_name {Pcie Logical Name} \
                  -paramNamelist {Pcie Parameter Name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
pcie_logical_name	string	Complete logical hierarchy of the PCIe block whose status is to be read from the device. This parameter is mandatory.
paramNamelist	string	Parameter name to read from the device. This parameter is optional.

Return Type	Description
string	Returns register:field value.

Error Codes

Error Code	Description
None	Parameter 'arg_name' is not defined. Valid command formatting is 'pcie_config_space [-deviceName "device name"] -pcie_logical_name "Pcie Logical Name" [-paramNameList "[Pcie Parameter Name]+"] [-allparams "TRUE | FALSE"]'.
None	Required parameter 'pcie_logical_name' is missing.
None	Parameter 'pcie_block_name' is not defined. Valid command formatting is'pcie_config_space [-deviceName "device name"] -pcie_logical_name "Pcie Logical Name" [-paramNameList "[Pcie Parameter Name]+"] [-allparams "TRUE | FALSE"]'.
None	Parameter 'param_name' is not defined. Valid command formatting is'pcie_config_space [-deviceName "device name"] -pcie_logical_name "Pcie Logical Name" [-paramNameList "[Pcie Parameter Name]+"] [-allparams "TRUE | FALSE"]'.

Supported Families

PolarFire
PolarFire SoC

Example

Output Display in SmartDebug window: 512 bytes

Return value to the tcl script: 0x2

```
pcie_config_space -pcie_block_name {sb_0/CM1_Subsystem/my_pcie_0} \
                  -param_name {neg_max_payload}
```

See Also

- pcie_ltssm_status

13.1.38. pcie_ltssm_status (Ask a Question)

Description

This Tcl command displays the current LTSSM state from the PLDA core in the SmartDebug log window and returns the register:field value to the Tcl.

```
pcie_ltssm_status -pcie_logical_name {PCIe logical name}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.

© 2025 Microchip Technology Inc. and its subsidiaries

pcie_ltssm_status (continued)		
Parameter	**Type**	**Description**
pcie_logical_name	string	Complete logical hierarchy of the PCIE block whose status is to be read from the device. This parameter is mandatory.

Return Type	Description
string	Returns register:field value.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'pcie_ltssm_status [-deviceName "device name"] -pcie_logical_name "Pcie Logical Name"'.
None	Required parameter 'pcie_logical_name' is missing.
None	Parameter 'pcie_block_name' is not defined. Valid command formatting is'pcie_ltssm_status [-deviceName "device name"] -pcie_logical_name "Pcie Logical Name"'.

Supported Families

PolarFire
PolarFire SoC

Example

Output Display in SmartDebug window: Configuration.Linkwidth.start Return value to the tcl script:

```
pcie_ltssm_status -pcie_block_name {sb_0/CM1_Subsystem/my_pcie_0}
```

See Also

- pcie_config_space

13.1.39. plot_eye (Ask a Question)

Description

This Tcl command is used to plot eye and export eye plots.

```
plot_eye [-deviceName "device name"] \
         -lane "Physical Lane Name" \
         [-file "filename"]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
lane	string	Specify the lane instance name.
file	string	Specify the path to the location where the file is to be exported.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'lane' is missing.
None	Parameter 'file' has illegal value.
None	Plot Eye: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Parameter 'lane' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'plot_eye [-deviceName "device name"] -lane "Physical Lane Name" [-file "filename"] .

Supported Families

PolarFire
PolarFire SoC

Example

This example plots eye for lane {Q2_LANE0} and saves it into {./export.txt} file.

```
plot_eye -lane {Q2_LANE0} -file {./export.txt}
```

13.1.40. prbs_test (Ask a Question)

Description

This Tcl command used in PRBS test to start, stop, reset the error counter and read the error counter value. PRBS data stream patterns are generated and checked by the internal SERDES block. These are used to self-test signal integrity of the device. You can switch the device through several predefined patterns.

Note:
prbs_test is renamed as smartbert_test in G5.

```
prbs_test [-deviceName device_name ] -start -serdes "integer value" \
          -lane "integer value" [-near] -pattern "PatternType"
prbs_test [-deviceName device_name ] -stop -serdes "integer value" \
          -lane "integer value"
prbs_test [-deviceName device_name ] -reset_counter \
          -serdes "integer value" -lane "integer value"
prbs_test [-deviceName device_name ] -read_counter \
          -serdes "integer value" -lane "integer value"
```

Arguments

Parameter	Type	Description
deviceName	string	Specifies device name. This parameter is optional if only one device is available in the current configuration or set for debug.
start	none	Starts the prbs test.
stop	none	Stops the prbs test.
reset_counter	none	Resets the prbs error count value to 0.
read_counter	none	Reads and prints the error count value.
SerDes	integer	SerDes block number. Must be between 0 and 4 and varies between dies.
lane	integer	SerDes lane number. Must be between 0 and 4.
near	none	Corresponds to near-end (on-die) option for prbs test. Not specifying implies off-die.

© 2025 Microchip Technology Inc. and its subsidiaries

prbs_test (continued)		
Parameter	**Type**	**Description**
pattern	string	The pattern sequence to use for PRBS test. It can be one of the following: prbs7, prbs11, prbs23, or prbs31.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'prbs_test [-deviceName "device name"] [-start "TRUE | FALSE"] [-stop "TRUE | FALSE"] [-reset_counter "TRUE | FALSE"] [-read_counter "TRUE | FALSE"] [-pattern "Pattern type"] -serdes "integer value" -lane "integer value" [-near "TRUE | FALSE"] '
None	Required parameter 'serdes' is missing.
None	serdes: Invalid argument value: 'serdes_value' (expecting integer value).
None	Required parameter 'lane' is missing.

Supported Families

SmartFusion 2
IGLOO 2
RTG4

Example

The following example starts PRBS test with the "prbs11" pattern:

```
prbs_test -start -serdes 1 -lane 0 -near -pattern "prbs11"
```

See Also

- smartbert_test
- loopback_mode
- loopback_test

13.1.41. process_job_request (Ask a Question)

Description

Processes a job request received from Job Manager. It is part of the Job Ticket generation process.
Note: The HSM server name must be specified using the HSM_SERVER_NAME DEF variable.

```
process_job_request -request_file "job request file"  -reply_file "job reply file" [-
overwrite_reply "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
request_file	string	Specifies the full file name of the job request file. This argument is mandatory.
reply_file	string	Specifies the full job name of the job reply file.

© 2025 Microchip Technology Inc. and its subsidiaries

process_job_request (continued)		
Parameter	**Type**	**Description**
overwrite_reply	string	Optional parameter. • 1: overwrites any pre-existing reply_file. • 0: prevents overwriting of any pre-existing reply_file.

Error Codes

Error Code	Description
None	Required parameter "request_file" is missing.
None	None Required parameter "reply_file" is missing.
None	Parameter "request_file" : the file "/request_file_name" does not exist.
None	overwrite_reply: Invalid argument value: "value" (expecting TRUE, 1, true, FALSE, 0 or false).
None	Cannot get HSM.
None	Parameter "param_name" is not defined. Valid command formatting is "process_job_request -request_file "job request file" \ -reply_file "job reply file" \ [-overwrite_reply "TRUE | FALSE"]".

Supported Families

PolarFire
SmartFusion 2
IGLOO 2

Example

This example processes a job request.

```
process_job_request \
                -request_file {D:/flashpro_files/jobmgr_project/cm_request.req} \
                -reply_file {D:/flashpro_files/jobmgr_project/cm_reply.rep} \
                -overwrite_reply {TRUE}
```

13.1.42. program_probe_insertion (Ask a Question)

Description

This Tcl command runs the probe insertion flow on the selected nets. This triggers Place and Route in incremental mode, and the selected probe nets are routed to the selected package pin. After incremental Place and Route, Libero automatically reprograms the device with the added probes. The log window shows the status of the Probe Insertion run.

Note:
Probe Insertion feature disabled in the SmartDebug Demo and Standalone modes.

```
program_probe_insertion
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Probe insertion operations are not supported in Standalone SmartDebug.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example runs the probe insertion flow:

```
program_probe_insertion
```

See Also

- add_probe_insertion_point
- remove_probe_insertion_point

13.1.43. read_active_probe (Ask a Question)

Description

This Tcl command reads active probe values from the device. The target probe points are selected by the select_active_probe command.

Note:
When the user tries to read at least one signal from the bus/group, the complete bus or group is read. The user is presented with the latest value for all the signals in the bus/group.

```
read_active_probe [-deviceName device_name ] \
                  [-name probe_name ] \
                  [-group_name bus_name | group_name ] \
                  [-value_type b|h] \
                  [-file file_path ]
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
name	string	Instead of all probes, read only the probes specified. The probe name should be prefixed with bus or group name if the probe is in the bus or group.
group_name	string	Instead of all probes, reads only the specified buses or groups specified here.
value_type	string	Optional parameter, used when the read value is stored into a variable as a string. b = binary h = hex
file	string	Optional. If specified, redirects output with probe point values read from the device to the specified file.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Parameter 'value_type' has illegal value.
None	Parameter 'name' has illegal value.
None	Parameter 'group_name' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example reads active probes of {group1}.

```
read_active_probe -group_name {group1}
```

See Also

- select_active_probe
- write_active_probe
- delete_active_probe

13.1.44. read_envm_memory (Ask a Question)

Description

This is a PolarFire SoC specific tcl command to read the ENVM memory from the device. It reads from the client configured in Libero or a page range can be given as inputs. The output will be in a matrix form displayed byte-wise and several rows with page number information.

Client name is optional in the command. However, if client name is specified, then it is validated against its start page and end page from the design.

```
read_envm_memory [-deviceName "device name"] \
                 [-client "client name"] \
                 -startpage "integer value" \
                 -endpage "integer value" \
                 [-fileName "envm data file name"]
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
client	string	Specifies the client name.
startpage	string	Specifies the start page that is integer value.
endpage	string	Specifies the end page that is integer value.
fileName	string	Specifies the file name path where the data will be saved.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
String	The output will be in a matrix form displayed byte-wise and several rows with page number information.

Error Codes

Error Code	Description
None	Required parameter 'startpage' is missing.
None	Required parameter 'endpage' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'read_envm_memory [-deviceName "device name"] [-client "client name"] -startpage "integer value" -endpage "integer value" [-fileName "envm data file name"]'

Supported Families

PolarFire SoC*

Example

This example reads eNVM memory from 0 to 205 pages.

```
read_envm_memory -startpage "0" -endpage "205"
```

13.1.45. read_id_code (Ask a Question)

Description

This Tcl command reads the ID code of a device. Each device has a unique ID, thereby executing this command returns a hexadecimal value.

Note:
Being able to read the IDCODE is an indication that the JTAG interface is working correctly.

```
read_id_code [-deviceName "device name"]
```

Arguments

Parameter	Type	Description
deviceName	none	Specify device name. This parameter is optional if only one device is available in the current configuration.

Return Type	Description
Hexadecimal number	Returns the hexadecimal ID code of the DUT/product.

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following command reads the IDCODE from the current configuration:

```
read_id_code
```

See Also

- set_debug_device
- read_device_status

13.1.46. read_device_status (Ask a Question)

Description

This Tcl command displays a summary of the device. Device status like ID code, design information, digest information, security and programmer information can be know using this command. Returns a log that can be saved to a file or printed.

```
read_device_status [-deviceName "device name"] [-file "filename"]
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
file	string	Specify path and the name of file where device status will be saved. This parameter is optional.

Return Type	Description
String	Displays the device information report with the value-property format.

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'read_device_status [-deviceName "device name"] [-file "filename"]'
None	Unable to read device information for the selected device: IDCode verify failed..

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

The following example saves the details of the device in log_file:

```
read_device_status -file {./log_file}
```

See Also

- read_id_code

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.47. read_lsram (Ask a Question)

Description

This tcl command reads a specified block of large SRAM from the device.

```
read_lsram [-deviceName "device name"] \
           [-name "LSRAM block name"] \
           [-logicalBlockName "USRAM user defined block name"] \
           [-port "LSRAM port name"] \
           [-fileName "Data file name"] \
           [-file "Data file name"]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
name	string	Specifies the name for the target block.
logicalBlockName	string	Specifies the name for the user defined memory block.
port	string	Specifies the port for the memory block selected. Can be either Port A or Port B.
fileName	string	Optional; specifies the output file name for the data read from the device.
file	string	Optional; specifies the output file name for the data read from the device.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Parameter 'fileName' has illegal value.
None	Port port_name is an invalid Port name.
None	Parameter 'port' has illegal value.
None	RAM port name must be specified.
None	LSRAM block cannot be read. Use phyical block option to read.
None	Parameter 'logicalBlockName' has illegal value.
None	Missing argument. Must specify '-name' or '-logicalBlockName'.
None	Parameter 'deviceName' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'read_lsram [-deviceName "device name"] [-name "LSRAM block name"] [-logicalBlockName "USRAM user defined block name"] [-port "LSRAM port name"] [-fileName "Data file name"] [-file "Data file name"]'.
None	LSRAM block name failed to read: Target block not found in debug file.
None	Parameter 'name' has illegal value.

Supported Families

PolarFire
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion 2
IGLOO 2
RTG4

Example

Reads the LSRAM Block Fabric_Logic_0/U2/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM1K20_IP from the PolarFire device and writes it to the file output.txt.

```
read_lsram \
-name {Fabric_Logic_0/U2/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM1K20_IP} \
-fileName {output.txt}
```

This example reads the uSRAM logical Block {Fabric_Logic_0/U3/F_0_F0_U1} from {Port A}.

```
read_lsram -logicalBlockName {Fabric_Logic_0/U2/F_0_F0_U1} -port {Port A}
```

See Also

- write_lsram

13.1.48. read_usram (Ask a Question)

Description

This tcl command reads a uSRAM block from the device.

```
read_usram [-deviceName "device name"] \
           [-name "USRAM block name"] \
           [-logicalBlockName "USRAM user defined block name"] \
           [-port "USRAM port name"] \
           [-file "Data file name"] \
           [-fileName "Data file name"]
```

Phisical block

```
read_usram -name {RAMS_LSRAM_URAM_0/PF_DPSRAM_C0_0/PF_DPSRAM_C0_0/
PF_DPSRAM_C0_PF_DPSRAM_C0_0_PF_DPSRAM_R0C0/INST_RAM1K20_IP}
```

Logical block

```
read_usram -logicalBlockName {RAMS_LSRAM_URAM_0/PF_URAM_C0_0/PF_URAM_C0_0} -port {Port A}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
name	string	Specifies the name for the target block.
logicalBlockName	string	Specifies the name of the user defined memory block.
port	string	Specifies the port of the memory block selected. Can be either Port A or Port B.
file	string	This parametr is ptional. Specifies the output file name for the data read from the device.
fileName	string	This parametr is ptional. Specifies the output file name for the data read from the device.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Missing argument. Must specify '-name' or '-logicalBlockName'.
None	Error: Parameter 'param_name' is not defined. Valid command formatting is'read_usram [-deviceName "device name"] [-name "USRAM block name"] [-logicalBlockName "USRAM user defined block name"] [-port "USRAM port name"] [-file "Data file name"] [-fileName "Data file name"]'.
None	Parameter 'name' has illegal value.
None	Error reading USRAM block value from the device: Target block not found in debug file.
None	Parameter 'file' has illegal value.
None	Port_name is an invalid Port name.
None	Parameter 'port' has illegal value.
None	RAM port name must be specified.
None	LSRAM block cannot be read. Use phyical block option to read.
None	Parameter 'logicalBlockName' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

Reads the uSRAM Block Fabric_Logic_0/U3/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM64x12_IP from the PolarFire device and writes it to the file sram_block_output.txt.

```
read_usram \
-name {Fabric_Logic_0/U3/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM64x12_IP} \
-fileName {output.txt}
```

This example reads the uSRAM logical Block {Fabric_Logic_0/U3/F_0_F0_U1} from {Port A}.

```
read_usram -logicalBlockName {Fabric_Logic_0/U3/F_0_F0_U1} -port {Port A}
```

See Also

- write_usram

13.1.49. read_snvm_memory (Ask a Question)

Description

This Tcl command reads client or page(s) in the sNVM (Secure Non volatile memory) memory from the device and returns a plain text (status and data of page).

Note:
If you have more than one client configured in Libero and use a client name with inappropriate -startpage, -endpage or -uskKey option values, then the command will fail.

```
read_snvm_memory [-deviceName "device name"] \
                 [-client "client name"] \
                 -startpage "integer value" \
                 -endpage "integer value" \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                [-fileName "snvm data file name"] \
                -uskKey "usk key"
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
client	string	Name of the client configured in Libero.
startpage	integer	Start page number configured in Libero.
endpage	integer	End page number.
fileName	string	Name of output file for memory read.
uskKey	hexadecimal	User Secret Key security key configured for the client in hexadecimal format

Return Type	Description
String	Displays page status and data of sNVM memory with plain text format.

Error Codes

Error Code	Description
None	Number of usk keys entered do not match with page range mentioned.
None	The start page value is incorrect for the client entered.
None	The end page value is incorrect for the client entered.
None	Invalid usk key specified. Input should be either '0' or 24 hexadecimal characters.
None	Parameter 'param_name' is not defined. Valid command formatting is 'read_snvm_memory [-deviceName "device name"] [-client "client name"] -startpage "integer value" -endpage "integer value" [-fileName "snvm data file name"] -uskKey "usk key"'.
None	Parameter 'uskKey' has illegal value.
None	Required parameter 'uskKey' is missing.
None	Parameter 'fileName' has illegal value.
None	Parameter 'endpage' has illegal value.
None	Parameter 'startpage' has illegal value.
None	Parameter 'client' has illegal value.
None	Client name was not found.
None	Parameter 'deviceName' has illegal value.
None	Parameter 'startpage' must be a positive integer value.
None	startpage: Invalid argument value: 'value' (expecting integer value).
None	endpage: Invalid argument value: 'value' (expecting integer value).
None	Parameter 'endpage' must be a positive integer value.

Supported Families

PolarFire
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

Example

Read "0" startpages and "2" endpages from sNMV memeory from the device and save into svnm.txt file:

```
read_snvm_memory -startpage "0" -endpage "2" \
                 -fileName "snvm.txt" -uskKey {0:0:0}
```

13.1.50. read_uprom_memory (Ask a Question)

Description

This Tcl command reads a uPROM memory block from the device.

Note:
If you have more than one clients configured in Libero and If you use client name with inappropriate -startaddress and -words then the command will fail.

```
read_uprom_memory [-deviceName "device name"] \
                  [-client "client name"] \
                  -startaddress "start address" \
                  -words "integer value" \
                  [-fileName "Data file name"]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
client	string	Specifie the name of client for memory read. Clients are configured in Libero tool under "device and memory initialisation". Those are a bunch of pages which can be used by the design to load data. Each client can have a different purpose.
startaddress	hexadecimal	Specifies the start address of the uPROM memory block.
words	integer	Specifies the number of 9-bit words.
fileName	string	Name of output file for memory read.

Return Type	Description
String	Displays page status and data of uPROM memory with plain text format..

Error Codes

Error Code	Description
None	Parameter 'deviceName' has illegal value.
None	Parameter 'client' has illegal value.
None	Parameter 'startaddress' has illegal value.
None	Required parameter 'startaddress' is missing.
None	Parameter 'words' has illegal value.
None	Required parameter 'words' is missing.
None	Parameter 'fileName' has illegal value.
None	Invalid argument value: ' -word {1.0} ' (expecting integer value).
None	Parameter 'param_name' is not defined. Valid command formatting is 'read_uprom_memory [-deviceName "device name"] [-client "client name"] -startaddress "start address" -words "integer value" [-fileName "Data file name"]'.
None	The startaddress value is incorrect for the client entered.

© 2025 Microchip Technology Inc. and its subsidiaries

read_uprom_memory (continued)	
Error Code	**Description**
None	The word number is incorrect for the client entered.
None	Parameter 'words' must be a positive integer value.

Supported Families

PolarFire
PolarFire SoC

Example

1. This example reads 10 9-bit words from uPROM memory '0xA' address :

```
read_uprom_memory -startaddress {0xA} -words {10}
```

2. This example reads 9-bit words from uPROM memory '0xA' address :

```
read_uprom_memory -client {client1} \
                  -startaddress "2" \
                  -words "2" \
                  -fileName "./data.txt"
```

13.1.51. read_flash_memory (Ask a Question)

Description

The command reads information from the NVM(Non volatile memory) modules. There are two types of information that can be read:

• Page Status – includes ECC1, ECC2, status, write count, access protection.

• Page Data

```
read_flash_memory [-deviceName { device_name }] \
                  [-startpage { integer_value }] \
                  [-endpage { integer_value }] \
                  [-access { all | status | data }] \
                  [-file { filename }]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
startpage	integer	Startpage is for page range. The value must be an integer. You must specify a -endpage and - block along with this argument.
endpage	integer	Endpage is for page range. The value must be an integer. You must specify a -startpage and -block along with this argument.
access	string	Specifies what eNVM information to check: page status, data or both. By default "all".
file	string	Name of output file for memory read.

Return Type	Description
String	Displays page status and data of Flash Memnory Contet with plain text format.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Parameter 'access' has illegal value.
None	Parameter 'endpage' has illegal value.
None	Parameter 'startpage' has illegal value.
None	Parameter 'deviceName' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'read_flash_memory [-deviceName "device name"] [-block "integer value"] [-client "client name"] [-startpage "integer value"] [-endpage "integer value"] [-access "all | status | data | raw"] [-file "filename"]'

Supported Families

SmartFusion 2
IGLOO 2*
RTG4*

Example

This example checks eNVM data information from 0 to 2 pages.

```
read_flash_memory -startpage 0 -endpage 2 -access {data} \
                  -file {flash_memory}
```

See Also

- check_flash_memory

13.1.52. record_actions (Ask a Question)

Description

This sequence can be used to access registers from an external processor to perform the same actions done in SmartDebug, to provide the register sequence for each of the actions performed in the XCVR Debug Window.

Note:

This command is valid only when the XCVR block is presented in Libero Design.

```
record_actions -start_recording | -stop_recording -file {file name}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
start_recording	none	Specifies the moment of start recording.
stop_recording	none	Specifies the moment of stop recording.
file	string	Specify path and the name of output *.txt file. This parameter is mandatory when `stop_recording` is specified.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Start Record parameter already set. Recording is in progress.
None	Record action parameters are absent. Either -start_recording or -stop_recording should ba passed as a parameter.
None	Both parameters cannot be passed. Either -start_recording or -stop_recording should ba passed as a parameter.
None	Error: Parameter 'param_name' is not defined. Valid command formatting is'record_actions [-deviceName "device name"] [-start_recording "TRUE | FALSE"] [-stop_recording "TRUE | FALSE"] [-file "file name"]'.

Supported Families

PolarFire
PolarFire SoC

Example

This example starts recording, then stops it and saves the recorded data in the {./actions} file.

```
record_actions -start_recording
record_actions -stop_recording -file {./actions.txt}
```

13.1.53. remove_from_probe_group (Ask a Question)

Description

This Tcl command removes the specified probe points from the group. That is, the removed probe points won't be associated with any probe group.

This command will fail if the specified value of the parameter is incorrect.

Note:
Probes cannot be removed from the bus.

```
remove_from_probe_group -name {group_name.probe_point_name}
```

Arguments

Parameter	Type	Description
name	string	Specifies one or more probe points to remove from the probe group.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'name' has illegal value.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'remove_from_probe_group [-name "name"]+'.

Supported Families

PolarFire
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion 2
IGLOO 2
RTG4

Example

This example removes DFN1_0_Q:DFN1_0/U0:Q instace from new_group.

```
remove_from_probe_group -name new_group.DFN1_0_Q:DFN1_0/U0:Q
```

See Also

- create_probe_group
- add_to_probe_group
- move_to_probe_group

13.1.54. remove_probe_insertion_point (Ask a Question)

Description

This Tcl command removes probe point from probe insertion list. The command will fail if the net name or driver are not specified or are incorrect.

Notes:

- Deleting probes from the probes list without clicking 'Run' does not automatically remove the probes from the design.
- Probe Insertion feature disabled in the SmartDebug Demo and Standalone modes.

```
remove_probe_insertion_point -net {net_name} -driver {driver_name}
```

Arguments

Parameter	Type	Description
net	string	Specify name of the existing net which is added in probe insertion list. This parameter is mandatory.
driver	string	Specify driver name. This parameter is mandatory.

Return Type	Description
None	Probe insertion operations are not supported in Standalone SmartDebug.

Error Codes

Error Code	Description
None	No probe point with net: "net_name" and driver: "driver_name" is added to be removed.
None	Parameter 'param_name' is not defined. Valid command formatting is 'remove_probe_insertion_point [-net "net_name"] [-driver "driver_name"] '.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example removes probe from the probe insertion list:

```
remove_probe_insertion_point -net {count_c[0]} -driver {Counter_out[0]:Q}
```

See Also

- add_probe_insertion_point
- program_probe_insertion

13.1.55. rescan_programmer (Ask a Question)

Description

This Tcl command rescans for programmer connected to the host via the USB port.

Notes:

- This command does not have any arguments.
- In the demo mode, this command returns "simulation".

```
rescan_programmer [-deviceName "device name"]
```

Arguments

Return Type	Description
String	Displays the programmer ID.

Error Codes

Error Code	Description
None	No programmers found. Check the programmer connection to the computer and ensure the drivers are properly installed.
None	Parameter 'param_name' is not defined. Valid command formatting is 'rescan_programmer [-deviceName "device name"] '.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example rescans the programmer connection.

```
rescan_programmer
```

This example rescans and retrieves the programmer name.

```
set programmer_id [rescan_programmer]
puts $programmer_id
```


© 2025 Microchip Technology Inc. and its subsidiaries

13.1.56. run_frequency_monitor (Ask a Question)

Description

This tcl command calculates the frequency of any signal in the design that can be assigned to Live Probe channel A and displays the Frequency. The Frequency unit of measurement is in Megahertz (MHz).

It is run after setting the live probe signal to channel A.

```
run_frequency_monitor [-deviceName "device name"] \
                      -signal "signal name" \
                      -time "time in seconds to delay before calculate"
```

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
signal	string	Specifies the signal name assigned to Live Probe channel A. This parameter must be specified with the -time parameter.
time	integer or double	Specifies the duration in seconds to run frequency monitor. The value can be 0.1, 1, 5, 8, or 10.

Return Type	Description
String	Displays the Frequency with value-property format.

Error Codes

Error Code	Description
None	No recognized device 'device_name' is available for debugging.
None	Parameters are missing.
None	Parameter '-signal' is missing.
None	Parameter '-time' is missing.
None	Invalid monitor time specified. The values can be either 0.1, 1, 5, 8 or 10.
None	Parameter 'param_name' is not defined. Valid command formatting is'run_frequency_monitor [-deviceName "device name"] [-signal "signal"] [-time "time"]'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

```
set_live_probe -probeA {Q_c:DFN1_0:Q} -probeB {}
run_frequency_monitor -signal {Q_c:DFN1_0:Q} -time {0.1}
```

See Also

- fhb_control
- set_live_probe
- event_counter

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.57. save_active_probe_list (Ask a Question)

Description

This Tcl command saves the list of active probes to a file.

```
save_active_probe_list [-deviceName "device name"] \
                       -file "path to the file"
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
file	string	The output file location.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'file' has illegal value.
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is'save_active_probe_list [-deviceName "device name"] -file "filename"'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example saves the active probe list in “./my_probes.txt” file.

```
save_active_probe_list -file “./my_probes.txt”
```

See Also

- delete_active_probe
- load_active_probe_list
- read_active_probe
- select_active_probe
- write_active_probe

13.1.58. save_live_probe_list (Ask a Question)

Description

This Tcl command saves the list of live probes to a file(*.txt).

```
save_live_probe_list [-deviceName "device name" -file "filename"
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Specify device name. This parameter is optional if only one device is available in the current configuration.
file	string	Specify path and the name of output file(*.txt). This parameter is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'save_live_probe_list [-deviceName "device name"] \ -file "filename"'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

The following example saves live probes list to live_probe_list.txt file:

```
save_live_probe_list -file {./live_probe_list.txt}
```

13.1.59. scan_ecc_memories (Ask a Question)

Description

This Tcl command scans and reports any errors detected in the PolarFire, PolarFire SoC, RTG4, RT PolarFire, and RT PolarFire SoC TPSRAM blocks that have ECC enabled.

```
scan_ecc_memories [-deviceName "device name"] \
                  [-fileName "Output file name"]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
fileName	string	Specify the name of the file where output is redirected. This argument is mandatory.

Return Type	Description
string	Reports memory blocks where data corruption has been detected.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'scan_ecc_memories [-deviceName "device name"] [-fileName "Output file name"]'.
None	Parameter 'fileName' has illegal value.
None	Unable to write to the file.

Supported Families

PolarFire
PolarFire SoC
RTG4

Example

This example scans TPSRAM blocks and saves the report into the {./output.txt} file. Log is provided with names of logical and physical blocks if corruption in data is detected. If no corruption is detected, then an appropriate message is provided.

```
scan_ecc_memories -filename {./output.txt}
```

13.1.60. select_active_probe (Ask a Question)

Description

This Tcl command manages the current selection of active probe points to be used by active probe READ operations. This command extends or replaces your current selection with the probe points found using the search pattern.

```
select_active_probe [-name probe_name_pattern ] \
                    [-reset true|false ]
```

Arguments

Parameter	Type	Description
name	string	Specifies the name of the probe. Optionally, search pattern string can specify one or multiple probe points. The pattern search characters “*” and “?” also can be specified to filter out the probe names.
reset	boolean	This optional parameter resets all previously selected probe points. If name is not specified, empties out current selection.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'reset' has illegal value.
None	Parameter 'name' has illegal value.
None	Cannot select active probe: Specified probe point(s) not found.
None	Parameter 'param_name' is not defined. Valid command formatting is'select_active_probe [-name "name"]* \[-reset "TRUE | FALSE"]'.

Supported Families

PolarFire

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

```
Select_active_probe -name out[5]:out[5]:Q
Select_active_probe -name out.out[1]:out[1]:Q \
                    -name out.out[3]:out[3]:Q \
                    -name out.out[5]:out[5]:Q
```

See Also

- write_active_probe
- read_active_probe

13.1.61. serdes_lane_reset (Ask a Question)

Description

This Tcl command resets lane in EPCS and PCI Lane modes. The result is shown in the log window/ console.

```
serdes_lane_reset -serdes "integer value" -lane "integer value"
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
serdes	integer	Specifies SerDes block number. It must be between 0 and 4 varies between dies. It must be one of the SerDes blocks used in the design.
lane	integer	Specifies SerDes lane number. It must be between 0 and 3. It must be one of the lanes enabled for the block in the design.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'serdes' is missing.
None	Parameter 'serdes' has illegal value.
None	Required parameter 'lane' is missing.
None	Parameter 'lane' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'serdes_lane_reset [-deviceName "device name"] -serdes "integer value" -lane "integer value"'.

Supported Families

SmartFusion 2
IGLOO 2
RTG4

© 2025 Microchip Technology Inc. and its subsidiaries

Example

This exmaple resets Lane 0, for specified SerDes block:

```
serdes_lane_reset -serdes 0 -lane 0
```

See Also

- serdes_read_regster
- serdes_write_regster

13.1.62. serdes_read_register (Ask a Question)

Description

This tcl command reads the SerDes register value and displays the result in the log window/console.

```
serdes_read_register -serdes "integer value" \
                     [-lane "integer value"] \
                     -name "Serdes Register Name"
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
serdes	integer	Specify SerDes block number. Must be between 0 and and varies between dies.
lane	integer	Specify SerDes lane number. Must be between 0 and 3. The lane number must be specified when the lane register is used. Otherwise, the command will fail. When the lane number is specified along with the SYSTEM or PCIe register, the command will fail with an error message, as the lane is not applicable to them.
name	string	Specify name of the SerDes register.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	SerDes block number must be specified and must be one of the following.
None	Parameter 'serdes' has illegal value.
None	serdes: Invalid argument value: '' (expecting integer value).
None	Reg_name is either an invalid or unsupported SerDes register.
None	Parameter 'lane' has illegal value.
None	lane: Invalid argument value: '' (expecting integer value).
None	Parameter 'name' has illegal value.
None	0 is either an invalid or unsupported SerDes register.
None	Parameter 'param_name' is not defined. Valid command formatting is 'serdes_read_register [-deviceName "device name"] [-serdes "integer value"] [-lane "integer value"] [-name "Serdes Register Name"] '.

© 2025 Microchip Technology Inc. and its subsidiaries

Supported Families

SmartFusion 2
IGLOO 2
RTG4

Example

This example reads {SYSTEM_SER_PLL_CONFIG_HIGH} register value of the SerDes 0.

```
serdes_read_register -serdes 0 -name {SYSTEM_SER_PLL_CONFIG_HIGH}
```

See Also

- serdes_lane_reset
- serdes_write_register

13.1.63. serdes_write_register (Ask a Question)

Description

This tcl command writes the value to the SerDes register. Displays the result in the log window/ console.

```
serdes_write_register [-serdes "integer value"] \
                      [-lane "integer value"] \
                      -name "Serdes register name" \
                      -value "Serdes register value"
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
serdes	integer	SerDes block number. Must be between 0 and 5 and varies between dies.
lane	integer	SerDes lane number. Must be between 0 and 3. The lane number should be specified when the lane register is used. Otherwise, the command will fail. When the lane number is specified along with the SYSTEM or PCIe register, the command will fail with an error message, as the lane is not applicable to them.
name	string	Name of the SerDes register.
value	hexadecimal	Specify the value in hexadecimal format.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'name' is missing.
None	Parameter 'serdes' has illegal value.
None	Parameter 'lane' has illegal value.
None	Parameter 'name' has illegal value.
None	Required parameter 'value' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

serdes_write_register (continued)	
Error Code	**Description**
None	Parameter 'value' has illegal value.
None	'Reg_name' is either an invalid or unsupported SerDes register.
None	SerDes lane number should not be specified for system register.
None	Parameter 'parm_name' is not defined. Valid command formatting is 'serdes_write_register [-deviceName "device name"] [-serdes "integer value"] [-lane "integer value"] -name "Serdes register name" -value "Serdes register value" '

Supported Families

SmartFusion 2
IGLOO 2
RTG4

Example

This example writes {0x5533} value to the {SYSTEM_SER_PLL_CONFIG_HIGH} SerDes register:

```
serdes_write_register -serdes 0 \
                      -name {SYSTEM_SER_PLL_CONFIG_HIGH} \
                      -value {0x5533}
```

See Also

- serdes_lane_reset
- serdes_read_register

13.1.64. set_debug_device (Ask a Question)

Description

Sets the device name in the tool.

```
set_debug_device -name “device name”
```

Arguments

Parameter	Type	Description
name	string	Specify the name of the device. This argument is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_debug_device -name "device name"
None	Required parameter 'name' is missing.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example sets MPF250T_ES as the device name:

```
set_debug_device -name “MPF250T_ES”
```

13.1.65. set_debug_programmer (Ask a Question)

Description

Identifies the programmer you want to use for debugging (if you have more than one).

```
set_debug_programmer -name { programmer_name }
```

Arguments

Parameter	Type	Description
name	string	Specify the programmer. This argument is mandatory.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Unable to select programmer 'prog_name' for debug.
None	Required parameter 'name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_debug_programmer -name "programmer name"'
None	Parameter 'name' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

The following example selects the programmer 10841:

```
set_debug_programmer -name {10841}
```

13.1.66. set_hsm_params (Ask a Question)

Description

Saves HSM parameters for the SmartDebug application. These parameters remain in effect until overridden by another invocation of this command.

```
set_hsm_params -hsm_server_name "hsm_server"
```

Arguments

Parameter	Type	Description
-hsm_server_name	string	Specifies the name or IP address of the HSM server computer.

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Required parameter 'hsm_server_name' is missing.
None	Could not get server information. Check server address and connection and try again.
None	HSM server name cannot be empty.
None	hsm_type_u: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Warning:Deprecated 'hsm_type_u' parameter is used.
None	FTP login password must be specified along with the user name.

Supported Families

PolarFire
SmartFusion 2
IGLOO 2

Example

This example sets the IP address of the server name computer.

```
set_hsm_params - hsm_server_name {10.241.140.224}
```

13.1.67. set_live_probe (Ask a Question)

Description

This Tcl command assigns channels A and/or B to the specified probe point(s). At least one probe point must be specified. Only exact probe name is allowed (that is, no search pattern that may return multiple points).

Note:
For RTG4, only one probe channel (Probe Read Data Pin) is available: A

```
set_live_probe [-deviceName "device_name" ] \
               [-probeA "probe point A" ] \
               [-probeB "probe point B" ]
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration or set for debug. This parameter is optional.
probeA	string	Specifies target probe point for the probe channel A. This parameter is optional.
probeB	string	Specifies target probe point for the probe channel B. This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_live_probe [-deviceName "device name"] [-probeA "probe point A"] [-probeB "probe point B"] '.
None	Parameter 'probeA' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

set_live_probe (continued)	
Error Code	**Description**
None	Cannot set live probes: Probe A is not found.
None	Cannot set live probes: IDCode verify failed.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Exception

- The array must be programmed and active.
- Active probe read or write operation will affect current settings of Live probe since they use the same probe circuitry inside the device.
- Setting only one Live probe channel affects the other one, so if both channels need to be set, they must be set from the same call to set_live_probe.
- Security locks may disable this function.
- To be available for Live probe, ProbeA and ProbeB I/Os must be reserved for Live probe respectively.

Example

The following example sets Live probe channel A to the probe point A12 on device sf2.

```
set_live_probe [-deviceName sf2] [-probeA A12]
```

See Also

- unset_live_probe

13.1.68. signal_integrity_export (Ask a Question)

Description

This Tcl command exports the current selected parameter options and other physical information for the selected lane/all lanes instance to an external PDC file.

The exported content will be in the form of two "set_io" commands, one for the TXP port and one for the RXP port of the selected lane instance.

```
signal_integrity_export -lane {phisical lane name} \
                        -pdc_file_name {path to the *.pdc file}
signal_integrity_export -pdc_file_name {path to the *.pdc file} \
                        -all_lanes
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
lane	string	Specifies the physical location of the lane. You must specify either 'lane' or 'all_lanes' parameter.
pdc_file_name	string	The path of the pdc file to be saved

© 2025 Microchip Technology Inc. and its subsidiaries

signal_integrity_export (continued)		
Parameter	**Type**	**Description**
all_lanes	none	Specifies all physical location of the lanes. You must specify either 'lane' or 'all_lanes' parameter.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_names' is not defined. Valid command formatting is 'signal_integrity_export [-deviceName "device name"] [-lane "Lane Instance Name"] [-pdc_file_name "PDC File Name"] [-all_lanes "TRUE | FALSE"]'.
None	Signal Integrity: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Parameter 'pdc_file_name' has illegal value.
None	Signal Integrity: Must not specify both '-lane' and '-all_lanes' command arguments.

Supported Families

PolarFire
PolarFire SoC

Example

The following example exports the current selected parameter options and other physical information for the "Q0_LANE0" lane instance to an ./SI_Q0LANE0.pdc:

```
signal_integrity_export -lane {Q0_LANE0} \
                        -pdc_file_name {./SI_Q0LANE0.pdc}
```

See Also

- signal_integrity_import
- signal_integrity_write
- load_SI_design_defaults

13.1.69. signal_integrity_import (Ask a Question)

Description

This Tcl command imports Signal Integrity parameter options and other physical information for the selected lane/all lanes from an external PDC file.

```
signal_integrity_import -lane {phisical lane name} \
                        -pdc_file_name {path to the *.pdc file}
signal_integrity_import -all_lanes \
                        -pdc_file_name {path to the *.pdc file}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
lane	string	Specifies the physical location of the lane. Must specify either '-lane' or '-all_lanes' command arguments.
pdc_file_name	string	The path of the pdc file to be saved.

© 2025 Microchip Technology Inc. and its subsidiaries

signal_integrity_import (continued)		
Parameter	**Type**	**Description**
all_lanes	none	Specifies all physical location of the lanes. Must specify either '-lane' or '-all_lanes' command arguments.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'signal_integrity_import [-deviceName "device name"] [-lane "Lane Instance Name"] [-all_lanes "TRUE | FALSE"] [-pdc_file_name "PDC File Name"]'.
None	Signal Integrity: Must specify '-pdc_file_name.
None	Signal Integrity: Import from *.pdc failed. Signal Integrity Constraints of lane not available in the file.
None	Signal Integrity: Unable to Import from *.pdc file.
None	Signal Integrity: Must specify one of '-lane'or '-all_lanes' command arguments.
None	Signal Integrity: Must not specify both '-lane' and '-all_lanes' command arguments.
None	Signal Integrity: Lane Name not found in the list of assigned physical lanes in Libero. Provide the correct lane name.

Supported Families

PolarFire
PolarFire SoC

Example

The following example imports Signal Integrity parameter options and other physical information for the "Q0_LANE0" lane from ./SI_Q0LANE0.pdc:

```
signal_integrity_import -lane {Q0_LANE0} \
                        -pdc_file_name {./SI_Q0LANE0.pdc}
```

See Also

- signal_integrity_export
- signal_integrity_write
- load_SI_design_defaults

13.1.70. signal_integrity_write (Ask a Question)

Description

This Tcl command writes parameter to a specified lane.

```
signal_integrity_write \
[-deviceName "device name"] \
[-lane "Lane Instance Name"] \
[-TX_EMPHASIS_AMPLITUDE "TX Transmit Emphasis and Amplitude"] \
[-TX_IMPEDANCE "TX Impedance"] \
[-TX_POLARITY "TX Impedance"] \
[-TX_TRANSMIT_COMMON_MODE_ADJUSTMENT "TX Transmit Common Mode Adjust"] \
[-RX_TERMINATION "RX Termination"] \
[-RX_LOSS_OF_SIGNAL_DETECTOR_LOW "RX Loss of Signal Detector Low "] \
[-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH "RX Loss of Signal Detector High "] \
[-RX_PN_BOARD_CONNECTION "RX Board Connection "] \
[-RX_POLARITY "Polarity RX "] \
[-RX_CTLE "RX CTLE"] \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
[-RX_INSERTION_LOSS "RX Insertion Loss"] \
[-RX_CDR_GAIN "RX CDR Gain"]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration.
lane	string	Specifies the physical location of the lane.
TX_EMPHASIS_AMPLITUDE	string	Specifies TX Emphasis Amplitude.
TX_IMPEDANCE	integer	Specifies TX Impedance(ohms) value. Possible values are: 100, 150, 85 180.
TX_TRANSMIT_COMMON_MODE_ADJUSTMENT	integer	Specifies TX Transmit Common Mode Adjustment (% of VDDA). Possible values are: 50, 60, 70, and 80.
RX_INSERTION_LOSS	string	Specifies RX Insertion Loss. Possible values are: 6.5dB, 17.0bdB and 25.0dB.
RX_CTLE	string	RX CTLE value.
RX_CDR_GAIN	string	Specifies CDR Gain value. It can be "Low" or "High".
RX_TERMINATION	integer	Specifies RX Termination(ohms). Possible values are: 85, 100 and150.
RX_PN_BOARD_CONNECTION	string	Specifies RX P/N Board Connection. Possible values are "AC_COUPLED_WITH_EXT_CAP" or "DC_COUPLED".
RX_LOSS_OF_SIGNAL_DETECTOR_LOW	string/ integer	Specifies RX Loss Signal Detector value. Possible values are: Off, PCIE, SATA, BMR and 1, 2, 3, 4, 5, 6 and 7.
RX_LOSS_OF_SIGNAL_DETECTOR_HIGH	string/ integer	Specifies RX Loss Signal Detector value. Possible values are: Off, PCIE, SATA, BMR and 1, 2, 3, 4, 5, 6 and 7.
RX_POLARITY	string	Specifies Polarity (P/N reversal) value, it can be "Normal" or "Inverted".

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'signal_integrity_write [-deviceName "device name"] [-lane "Lane Instance Name"] [-TX_EMPHASIS_AMPLITUDE "TX Transmit Emphasis and Amplitude"] [-TX_IMPEDANCE "TX Impedance"] [-TX_POLARITY "TX Impedance"] [-TX_TRANSMIT_COMMON_MODE_ADJUSTMENT "TX Transmit Common Mode Adjust"] [-RX_TERMINATION "RX Termination"] [-RX_LOSS_OF_SIGNAL_DETECTOR_LOW "RX Loss of Signal Detector Low "] [-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH "RX Loss of Signal Detector High "] [-RX_PN_BOARD_CONNECTION "RX Board Connection "] [-RX_POLARITY "Polarity RX "] [-RX_CTLE "RX CTLE"] [-RX_INSERTION_LOSS "RX Insertion Loss"] [-RX_CDR_GAIN "RX CDR Gain"]'.
None	Parameter 'RX_CDR_GAIN' has illegal value.
None	Parameter 'RX_INSERTION_LOSS' has illegal value.
None	Parameter 'RX_CTLE' has illegal value.
None	Signal Integrity: RX_INSERTION value is invalid. It must be 6.5dB, 17.0dB or 25.0dB.
None	Parameter 'RX_POLARITY' has illegal value.
None	Parameter 'RX_PN_BOARD_CONNECTION' has illegal value.
None	Parameter 'RX_LOSS_OF_SIGNAL_DETECTOR_HIGH' has illegal value.

© 2025 Microchip Technology Inc. and its subsidiaries

signal_integrity_write (continued)	
Error Code	**Description**
None	Parameter 'RX_LOSS_OF_SIGNAL_DETECTOR_LOW' has illegal value.
None	Parameter 'RX_TERMINATION' has illegal value.
None	Parameter 'TX_TRANSMIT_COMMON_MODE_ADJUSTMENT' has illegal value.
None	Parameter 'TX_IMPEDANCE' has illegal value.
None	Parameter 'TX_EMPHASIS_AMPLITUDE' has illegal value.
None	Signal Integrity: Must specify one of '-TX_EMPHASIS_AMPLITUDE', '-TX_IMPEDANCE','-TX_POLARITY','-TX_TRANSMIT_COMMON_MODE-ADJUSTMENT','-RX_TERMINATION','-RX_LOSS_OF_SIGNAL_DETECTOR_LOW', '-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH', 'RX_PN_BOARD_CONNECTION', '-RX_POLARITY', '-RX_CTLE', '-RX_INSERTION_LOSS' or '-RX_CDR_GAIN' arguments.
None	Parameter 'lane' has illegal value.

Supported Families

PolarFire
PolarFire SoC

Example

Write signal integrity on "Q2_LANE0" lane with the possible values of parameters:

```
signal_integrity_write \
-lane {Q2_LANE0} \
-TX_EMPHASIS_AMPLITUDE {400mV_with_-3.5dB} \
-TX_IMPEDANCE {100} \
-TX_TRANSMIT_COMMON_MODE_ADJUSTMENT {50} \
-RX_TERMINATION {100} \
-RX_LOSS_OF_SIGNAL_DETECTOR_LOW {1} \
-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH {3} \
-RX_PN_BOARD_CONNECTION {AC_COUPLED_WITH_EXT_CAP} \
-RX_POLARITY {Normal} \
-RX_CTLE {No_Peak_+2.8dB} \
-RX_INSERTION_LOSS {6.5dB} \
-RX_CDR_GAIN {High}
```

See Also

- signal_integrity_import
- signal_integrity_export
- load_SI_design_defaults

13.1.71. smartbert_test (Ask a Question)

Description

This Tcl command is used for the following:

- Start a Smart BERT test - Start a test with a specified PRBS patterns on a specified SmartBERT lane.
- Stop a Smart BERT test - Stop SmartBERT/PRBS test on a specified lane.
- Reset error count - Reset counter of a lane during selected pattern test.
- Inject error - Inject error into a SmartBERT IP lane.

```
smartbert_test -start -pattern {pattern name} \
               -lane {Physical Location} \
               [-smartbert_ip {TRUE | FALSE}] \
               [-EQ-NearEndLoopback]
smartbert_test -reset_counter -lane {Physical Location}
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
smartbert_test -lane {Physical Location} [-inject_error {TRUE | FALSE}]
smartbert_test -stop -lane {Physical Location}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
lane	string	Specify the physical location of the lane.
start	none	Start the Smart BERT test.
pattern	string	Specify the pattern type of the Smart BERT test. Valid values of pattern type are: PRBS7, PRBS9, PRBS15, PRBS23 and PRBS31.
smartbert_ip	boolean	This parameter applicable to the lane configured through SmartBERT IP.
EQ-NearEndLoopback	none	Enable EQ-Near End Loopback on specified lane.
reset_counter	none	Reset lane error counter on hardware and cumulative error count on the UI.
inject_error	boolean	Specifies to inject error into a SmartBERT IP. Valid values are: TRUE or FALSE.
stop	none	Stop the smart BERT test.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	SmartBert test: Must specify one of '-start', '-stop', '-reset_counter' or '-read_counter' arguments.
None	SmartBert test: Lane Name not found in the list of assigned physical lanes in Libero. Provide the correct lane name.
None	PRBS test: Invalid pattern type specified.
None	SmartBert test: Is not a G5 Device.
None	Parameter 'param_name' is not defined. Valid command formatting is 'smartbert_test [-deviceName "device name"] [-smartbert_ip "TRUE | FALSE"] [-start "TRUE | FALSE"] [-stop "TRUE | FALSE"] [-reset_counter "TRUE | FALSE"] [-read_counter "TRUE | FALSE"] [-pattern "Pattern type"] [-lane "Physical Lane Name"] [-EQ-NearEndLoopback "TRUE | FALSE"] [-inject_error "TRUE | FALSE"] '.

Supported Families

PolarFire
PolarFire SoC

Example

The following example starts Smart BERT test with a "prbs7" PRBS patterns on a "Q0_LANE0" SmartBERT lane:

```
# Transceiver lane without SmartBert IP without EQ-NearEndLoopback
smartbert_test -start -pattern {prbs7} -lane {Q0_LANE0}

# Transceiver SmartBERT IP lane
smartbert_test -start -smartbert_ip "TRUE" \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                -pattern {prbs7} -lane {Q0_LANE0} \
                -EQ-NearEndLoopback "TRUE"
```

The following example resets counter of a "Q0_LANE0" lane during selected pattern test.

```
smartbert_test -reset_counter -lane {Q0_LANE0}
```

The following stops Smart BERT/PRBS test on a "Q0_LANE0" SmartBERT lane:

```
smartbert_test -stop -lane {Q0_LANE0}
```

The following example injects error into a "Q0_LANE0" SmartBERT IP lane:

```
smartbert_test -lane {Q0_LANE0} -inject_error {TRUE}
```

See Also

- prbs_test
- loopback_mode
- loopback_test

13.1.72. static_pattern_transmit (Ask a Question)

Description

This Tcl command starts and stops a Static Pattern Transmit on selected lanes.

```
static_pattern_transmit -start -lane {Transceiver Physical Lane Name} \
                        -pattern {pattern_type} \
                        -value {user_pattern_value}
static_pattern_transmit -stop -lane {Transceiver Physical Lane Name} \
                        -pattern {empty} -value {empty}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
lane	string	Specify Transceiver physical Lane Name.
start	none	Start the Static Pattern Transmit.
stop	none	Stop the Static Pattern Transmit.
pattern	string	Specify "pattern_type" of Static Pattern Transmit. "pattern_type" valid values are: • fixed - Fixed Pattern is a 10101010... pattern. Length is equal to the data width of the Tx Lane. • maxrunlength - Max Run Length Pattern is a 1111000... pattern. Length is equal to the data width of the Tx Lane, with half 1s and half 0s. • custom - User Pattern is a user defined pattern in the value column. Length is equal to the data width.
value	hexadecimal	Specify user_pattern_value in hex if pattern_type selected is custom. Takes the input pattern to transmit from the Lane Tx of selected lanes. Internal validators dynamically check the pattern and indicate when an incorrect pattern is given as input.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'static_pattern_transmit [-deviceName "device name"] [-start "TRUE | FALSE"] [-stop "TRUE | FALSE"] [-lane "Physical Lane Name"] [-pattern "Pattern type"] [-value "User pattern Value"]'.
None	Static Pattern Transmit: Must specify one of '-start', '-stop' arguments.
None	Static Pattern Transmit: Transceiver physical Lane Name must be specified.
None	Static Pattern Transmit: Must specify pattern type argument.
None	Static Pattern Transmit: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Static Pattern Transmit: Invalid static pattern type specified.
None	Static Pattern Transmit: Pattern Length exceeds the expected size.

Supported Families

PolarFire
PolarFire SoC

Example

The following examples starts/stops fixed/maxrunlength Static Pattern transmit on "Q0_LANE0" /"Q0_LANE1" lane:

```
static_pattern_transmit -start -lane {Q0_LANE0} \
                        -pattern {fixed} -value {}
static_pattern_transmit -stop -lane {Q0_LANE0}

static_pattern_transmit -start -lane {Q0_LANE1}
                        -pattern {maxrunlength} -value {}
static_pattern_transmit -stop -lane {Q0_LANE1}
```

The following examples starts/stops fcustom Static Pattern transmit on "Q2_LANE2" lane with "1010111" user pattern value:

```
static_pattern_transmit -start -lane {Q2_LANE2} \
                        -pattern {custom} -value {1010111}
static_pattern_transmit -stop -lane {Q2_LANE1}
```

13.1.73. transceiver_lane_reset (Ask a Question)

Description

This tcl command resets the transceiver lane.

```
transceiver_lane_reset [-deviceName "device name"] \
                       -lane {physical location}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
lane	string	Specify the physical lane instance name.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Phy Reset: Lane Name not found in the list of assigned physical lanes in Libero.Provide the correct lane name.
None	Parameter 'lane' has illegal value.
None	Phy Reset: Transceiver physical Lane Name must be specified.
None	Parameter 'param_name' is not defined. Valid command formatting is'transceiver_lane_reset [-deviceName "device name"] [-lane "Physical Lane Name"]'.
None	Parameter 'deviceName' has illegal value.

Supported Families

PolarFire
PolarFire SoC

Example

This tcl example resets the lane {Q0_LANE0}.

```
transceiver_lane_reset -lane {Q0_LANE0}
```

13.1.74. tvs_monitor (Ask a Question)

Description

This Tcl command reads the Temperature and Voltage Sensor (TVS) values from device and saves the values in a `.csv` user specified extension file.

Important: This command is valid only when the TVS IP is presented in Libero Design.

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
start	none	Starts the TVS monitor.
stop	none	Stops the TVS monitor.
interval	integer	Specifies the duration in Seconds. The allowed limit is 1 sec to 60 sec. This parameter is optional. The default interval is 1 sec.
file	string	Specify the name of the file where output is redirected. This argument is mandatory.

Error Codes

Error Code	Description
None	TVS Monitor feature is not available for this design.
None	Failed to read TVS Channels values.
None	Parameters are absent. Either -start or -stop should be passed as a parameter.

Supported Families

PolarFire

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC

Example

This tcl example reads the TVS.

```
tvs_monitor -start -file {./temp.csv}
tvs_monitor -start -interval 10 -file {./temp.csv}
tvs_monitor -stop
```

13.1.75. ungroup (Ask a Question)

Description

This Tcl command disassociates the probes as a group.

```
ungroup -name {group name}
```

Arguments

Parameter	Type	Description
name	string	Specifies name of the group.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	None

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

Ungroup 'my_group' probe groups:

```
ungroup -name "my_group"
```

See Also

- create_probe_group
- add_to_probe_group

13.1.76. unset_live_probe (Ask a Question)

Description

This Tcl command discontinues the debug function and clears both live probe channels (Channel A and Channel B). An all zeros value is shown for both channels in the oscilloscope.

© 2025 Microchip Technology Inc. and its subsidiaries

Note:
For RTG4, only one probe channel (Probe Read Data Pin) is available.

```
unset_live_probe [-deviceName "device name"] \
                 [-probeA "TRUE | FALSE"] \
                 [-probeB "TRUE | FALSE"]'
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
probeA	boolean	Specify 1 or TRUE for unset live probe on Channel A, otherwise specify 0 or FALSE.
probeB	boolean	Specify 1 or TRUE for unset live probe on Channel B, otherwise specify 0 or FALSE.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Cannot unset live probes: Mention the name of the channel to unset.
None	Parameter 'param_name' is not defined. Valid command formatting is 'unset_live_probe [-deviceName "device name"] [-probeA "TRUE | FALSE"] [-probeB "TRUE | FALSE"]'.
None	probeA: Invalid argument value: (expecting TRUE, 1, true, FALSE, 0 or false).
None	probeB: Invalid argument value: (expecting TRUE, 1, true, FALSE, 0 or false).
None	Parameter 'deviceName' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

The following example unsets both live probe channels (Channel A and Channel B) from the device sf2:

```
unset_live_probes -probeA 1 -probeB 1 [-deviceName {sf2}]
```

See Also

- set_live_probe

13.1.77. update_fp6_programmers (Ask a Question)

Description

This Tcl command updates all the FlashPro6 programmers that require update. This command takes no parameters. To execute the Tcl Command in Libero SoC, add the command before `run_tool`. To

© 2025 Microchip Technology Inc. and its subsidiaries

execute the Tcl command in FlashPro Express tool or SmartDebug tool, add the Tcl command before `run_selected_actions`.

```
update_fp6_programmers
```

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'update_fp6_programmers'.

Supported Families

PolarFire®

PolarFire SoC

Example

This example updates all the FlashPro6 programmers that require update.

```
update_fp6_programmers
```

13.1.78. write_active_probe (Ask a Question)

Description

This Tcl command sets the target probe point on the device to the specified value. The target probe point name must be specified.

```
write_active_probe [-deviceName device_name ] \
                    -name probe_name \
                    -value true|false \
                    -group_name group_bus_name \
                    -group_value "hex-value" | "binary-value"
```

Arguments

Parameter	Type	Description
deviceName	string	Parameter is optional if only one device is available in the current configuration.
name	string	Specifies the name for the target probe point. Cannot be a search pattern.
value	boolean	Specifies values to be written. True = High, False = Low.
group_name	string	Specify the group or bus name to write to complete group or bus.
group_value	string	Specify the value for the complete group or bus. Hex-value format : "<size>'h<value>" Binary-value format: "<size>'b<value>"

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'group_value' has illegal value.
None	Active probe value must be specified.
None	Parameter 'group_name' has illegal value.
None	Parameter 'value' has illegal value.
None	Parameter 'name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'write_active_probe [-deviceName "device name"] \[-name "Probe point name"]* \[-value "TRUE | FALSE"]* \[-group_name "Group or Bus Name"]* \[-group_value "Group or Bus value"]* \[-silent "TRUE | FALSE"]'.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example writes to a single probe.

```
write_active_probe -name out[5]:out[5]:Q -value true
```

This example writes to a probe in the group:

```
write_active_probe -name grp1.out[3]:out[3]:Q -value "low"
```

This example writes the value to complete group:

```
write_active_probe -group_name grp1 -group_value “8’hF0”
```

This example writes multiple probes at the same time:

```
write_active_probe -group_name out \
                   -group_value “8’b11110000” \
                   -name out[2]:out[2]:Q \
                   -value true
```

See Also

- select_active_probe
- read_active_probe
- save_active_probe_list
- load_active_probe_list

13.1.79. write_lsram (Ask a Question)

Description

This tcl command writes a word into the specified large TPSRAM location.

TPSRAM block has aspect ratio of 512x40 (ECC disabled) and 512x33 (ECC enabled). SmartDebug enhanced the physical block view to read and write as 40-bit and 33-bit data. The write value is more than the size of integer and hence provided a new parameter -tpsramValue to accommodate the changes

© 2025 Microchip Technology Inc. and its subsidiaries

Write onto TPSRAM physical block → 40-bit wide or 33-bit wide for PF and 18-bit wide or 36-bit wide for RTG4

Physical block

```
write_lsram [-deviceName "device name"] \
            -name {physical block name} \
            -offset {offset value} \
            -value {integer value} \
            [-tpsramValue "TPSRAM physical block word value"]
```

Logical block

```
write_lsram [-deviceName "device name"] \
            -logicalBlockName {block name} \
            -port {port name} \
            -offset {offset value} \
            -logicalValue {hexadecimal value}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
name	string	Specifies the name for the target block.
logicalBlockName	string	Specifies the name of the user defined memory block.
port	string	Specifies the port of the memory block selected. Can be either Port A or Port B.
offset	integer	Offset (address) of the target word within the memory block.
logicalValue	hexadecimal	Specifies the hexadecimal value to be written to the memory block. Size of the value is equal to the width of the output port selected.
value	integer	Word to be written to the target location. Depending on the configuration of memory blocks, the width can be 1, 2, 5, 10, or 20 bits. This is an integer, which minimum value is 0 and may go up to depending on the size of each location}
tpsramValue	integer	integer value, minimum value is 0 to (2^N - 1) where N is number of bits configured. PolarFire , PolarFire Soc and RTG4 only.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'write_lsram [-deviceName "device name"] [-name "LSRAM block name"] [-logicalBlockName "USRAM user defined block name"] [-port "LSRAM port name"] [-offset "integer value"] [-logicalValue "LSRAM block word value"] [-value "integer value"] [-tpsramValue "TPSRAM physical block word value"]'.
None	Parameter 'name' has illegal value
None	Missing argument. Must specify '-name' or '-logicalBlockName'.
None	Parameter 'logicalValue' has illegal value.
None	Error write LSRAM block PF_DPSRAM_C0_0/PF_DPSRAM_C0_0: Target memory block should first be read before write..
None	Parameter 'logicalBlockName' has illegal value.
None	LSRAM block cannot be read. Use phyical block option to read.

© 2025 Microchip Technology Inc. and its subsidiaries

write_lsram (continued)	
Error Code	**Description**
None	RAM port name must be specified.
None	Parameter 'port' has illegal value.
None	Port port_name is an invalid Port name.
None	Parameter 'file' has illegal value Parameter 'tpsramValue' has illegal value.
None	Parameter 'value' has illegal value.
None	value: Invalid argument value: 'value' (expecting integer value).
None	Parameter 'offset' has illegal value.
None	offset: Invalid argument value: 'value' (expecting integer value).
None	Active probe value must be specified.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

This example writes a value of 69905 to the physical block of device PolarFire in the "PF_DPSRAM_C0_0/INST_RAM1K20_IP" with an offset of 3:

```
write_lsram -name {PF_DPSRAM_C0_0/INST_RAM1K20_IP} \
            -offset 3 -value 69905
```

```
write_lsram -logicalBlockName {PF_DPSRAM_C0_0/PF_DPSRAM_C0_0} \
            -port {Port B} -offset {1} -logicalValue {0xA} \
            -tpsramValue 300
```

See Also

- read_lsram

13.1.80. write_usram (Ask a Question)

Description

This tcl command writes a 12-bit word into the specified uSRAM location.

```
write_usram [-deviceName "device name"] \
            [-name "USRAM block name"] \
            [-logicalBlockName "USRAM user defined block name"] \
            [-port "USRAM port name"] \
            [-offset "integer value"] \
            [-logicalValue "USRAM block word value"] \
            [-value "integer value"]
```

Physical block

```
write_usram -name block_name \
            -offset offset_value \
            -value integer_value
```

Logical block

```
write_usram -logicalBlockName block_name \
            -port port_name \
            -offset offset_value \
            -logicalValue hexadecimal_value
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
name	string	Specifies the name for the target block.
logicalBlockName	string	Specifies the name of the user defined memory block.
port	string	Specifies the port of the memory block selected. Can be either Port A or Port B.
offset	integer	Offset (address) of the target word within the memory block.
logicalValue	integer	Specifies the hexadecimal value to be written to the memory block. Size of the value is equal to the width of the output port selected.
value	integer	12- bit value to be written.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'logicalValue' has illegal value.
None	offset: Invalid argument value: 'offset_value' (expecting integer value).
None	Parameter 'offset' has illegal value.
None	Active probe value must be specified.
None	Port_name is an invalid Port name.
None	Parameter 'port' has illegal value.
None	LSRAM port name must be specified.
None	LSRAM block word cannot be written. Use phyical block word to write.
None	Missing argument. Must specify '-name' or '-logicalBlockName'.
None	Parameter 'value' has illegal value.
None	Parameter 'name' has illegal value.

Supported Families

PolarFire
PolarFire SoC
SmartFusion 2
IGLOO 2
RTG4

Example

Writes a value of 0x291 to the device PolarFire in the Fabric_Logic_0/U3/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM64x12_IP with an offset of 0.

```
write_lsram \
-name {Fabric_Logic_0/U3/F_0_F0_U1/ramtmp_ramtmp_0_0/INST_RAM64x12_IP} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
-offset 0 \
-value 291
```

```
write_usram -logicalBlockName {Fabric_Logic_0/U3/F_0_F0_U1} -port {Port A} -offset 1
-logicalValue {00FFF}
```

See Also

- read_usram

13.1.81. xcvr_add_register (Ask a Question)

Description

This Tcl command adds transceiver registers details to the SmartDebug register access interface.

```
xcvr_add_register [-deviceName {device name} \
                  [-reg_name {Add Register Names}]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required, if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
reg_name	string	Name of the register.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	reg_name register is not added to Register access list.
None	Parameter 'reg_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is 'xcvr_add_register [-deviceName "device name"] [-reg_name "Add Register Names"]'.
None	Register Access: Must specify '-reg_name

Supported Families

PolarFire
PolarFire SoC

Example

This example adds all the registers under the {SERDES1} component.

```
xcvr_add_register -reg_name {SERDES1}
```

See Also

- xcvr_export_register
- xcvr_read_register
- xcvr_write_register

© 2025 Microchip Technology Inc. and its subsidiaries

13.1.82. xcvr_export_register (Ask a Question)

Description

This Tcl command exports previously added transceiver registers details to a *.csv file.

```
xcvr_export_register [-deviceName {device name}] \
                     [-file_name {file Name}] [-all]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
file_name	string	Path of the export file.
all	none	Specify to export all transceiver registers details to *.csv file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is'xcvr_export_register [-deviceName "device name"] [-file_name "File Name"] [-all "TRUE | FALSE"]'.
None	Parameter 'file_name' has illegal value.
None	Register Access: file specified for Export must have .csv extension.
None	Register Access: Must specify '-file_name.

Supported Families

PolarFire
PolarFire SoC

Example

Export previously added transceiver registers details to a .csv file:

```
xcvr_export_register -file_name {register_export.csv}
```

See Also

- xcvr_add_register
- xcvr_read_register
- xcvr_write_register

13.1.83. xcvr_import_register (Ask a Question)

Description

This Tcl command imports exported transceiver registers details from a `*.csv` file.

```
xcvr_import_register \
            -file_name {absolute or relative path to the *.csv file name}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
file_name	string	Path of the exported transceiver file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'file_name' is missing.
None	Parameter 'file_name' has illegal value.
None	Register Access: file specified for Import must have .csv extension.
None	Register Access: Must specify '-file_name.
None	Parameter 'param_name' is not defined. Valid command formatting is'xcvr_import_register -file_name "File Name"'.

Supported Families

PolarFire
PolarFire SoC

Example

Import previously exported transceiver registers details from a `.csv` file:

```
xcvr_import_register -file_name {register_list.csv}
```

13.1.84. xcvr_read_register (Ask a Question)

Description

This Tcl command reads SCB registers and their field values. Read value is in hex format. This command is used in SmartDebug Signal Integrity.

```
xcvr_read_register [-deviceName {device name}] \
                   -inst_name {instanse name} \
                   -reg_name [<reg_name> | <reg_name:field_name>]
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the set_debug_device command.
inst_name	string	Specify the lane instance name used in the design.
reg_name	string	Specify the <reg_name> for register name or <reg_name>:<field_name> for the register's field.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'reg_name' has illegal value.
None	Required parameter 'reg_name' is missing.
None	Parameter 'inst_name' has illegal value.
None	Required parameter 'inst_name' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'xcvr_read_register [-deviceName "device name"] -inst_name "Instance Name" -reg_name "Transceiver Register Name" '.

Supported Families

PolarFire
PolarFire SoC

Example

Reading pcslane's 32-bit register LNTV_R0:

```
xcvr_read_register -inst_name {CM1_PCIe_SS_0/PF_PCIE_0/LANE1} \
                   -reg_name {LNTV_R0}
```

See Also

- xcvr_add_register
- xcvr_export_register
- xcvr_write_register

13.1.85. xcvr_write_register (Ask a Question)

Description

This Tcl command writes SCB registers and their field values. Write value is in hex format. This command is used in SmartDebug Signal Integrity.

```
xcvr_write_register [-deviceName {device name}] \
                    [-inst_name {Instance name}] \
                    -reg_name {Transceiver register name} \
                    -value {Transceiver register value}
```

Arguments

Parameter	Type	Description
deviceName	string	Optional user-defined device name. The device name is not required if there is only one device in the current configuration, or a device has already been selected using the `set_debug_device` command.
inst_name	string	Specify the lane instance name used in the design.
reg_name	string	Specify the <reg_name> for register name or <reg_name>:<field_name> for the register's field.
value	integer	Specify the value in hex format.

Return Type	Description
None	None

© 2025 Microchip Technology Inc. and its subsidiaries

Error Codes

Error Code	Description
None	Parameter 'value' has illegal value.
None	Required parameter 'value' is missing.
None	Parameter 'reg_name' has illegal value.
None	Required parameter 'reg_name' is missing.
None	Parameter 'inst_name' has illegal value.
None	Parameter 'param_name' is not defined. Valid command formatting is'xcvr_write_register [-deviceName "device name"] [-inst_name "Instance name"] [-broadcast "TRUE | FALSE"] -reg_name "Transceiver register name" -value "Transceiver register value"'.
None	Must specify either '-inst_name' or '-broadCast' parameter.

Supported Families

PolarFire
PolarFire SoC

Example

Writing pcscmn's 32-bit register GSSCLK_CTRL

```
xcvr_write_register -inst_name {CM1_PCIe_SS_0/PF_PCIE_0/LANE1} \
                    -reg_name {GSSCLK_CTRL} \
                    -value 0xffffffff
```

See Also

- xcvr_add_register
- xcvr_export_register
- xcvr_read_register

© 2025 Microchip Technology Inc. and its subsidiaries

14. System Builder Tcl Commands (Ask a Question)

14.1. sb_configure_page (Ask a Question)

Description

This Tcl command is used to configure the parameters of a System Builder component page.

```
sb_configure_page \
-component_name {component_name} \
-page_name {page_name} \
[-params {param_list}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-page_name {page_name}	string	Mandatory. Name of the system builder page being configured.
-params {param_list}	string	Mandatory. List of parameters being configured within given page.

Example

```
sb_configure_page -component_name {sb} -page_name{DEVICEFEATURES} -params {USE_ENVM:1}
```

14.2. sb_add_core (Ask a Question)

Description

This Tcl command is used to add cores to Master/Slave subsystems in the **Peripherals** page of the System Builder component.

Notes: There are two types of cores that can be added using the `sb_add_core` command.

1. One type is of actual cores like CoreI2C, CoreGPIO with fixed core versions available in the **Peripherals** page. If, for example, CoreI2C is added to a subsystem with core_name specified as i2c_peripheral and a core count of 4, then the actual instance names of CoreI2C added will be i2c_peripheral_0, i2c_peripheral_1, i2c_peripheral_2 and i2c_peripheral_3 in the generated design.
2. Second type of cores are Fabric AMBA Slave and Fabric AMBA Master. Adding them to a subsystem will configure the bus core of the subsystem to enable and expose master/slave interfaces of the bus core on the generated System Builder component to be connected to actual masters/slaves in the fabric. If, for example, Fabric AMBA Slave configured as AHBLite(`AMBA_INTERFACE_TYPE:AHBLITE`) is added to a subsystem with core_name specified as ahb_slave and a core count of 4, then the CoreAHBLite bus core of that subsystem will be configured to enable and expose 4 AHBLite slave bus interfaces with names ahb_slave, ahb_slave_1, ahb_slave_2 and ahb_slave_3 on the System Builder component to be connected to actual slave peripherals in the fabric.

```
sb_add_core \
-component_name {component_name} \
-core_vlnv {vendor:library:name:version} \
[-core_name {core_name}] \
-subsystem_name {subsystem_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.

© 2025 Microchip Technology Inc. and its subsidiaries

sb_add_core (continued)		
Parameter	**Type**	**Description**
-core_vlnv {vendor:library:name:version}	string	Mandatory. Version identifier of the core being instantiated in the SmartDesign.
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component. If no details are provided, the instance name will be automatically defined as `<vlnv_core_name>_n` (where `n` starts at 0).
-subsystem_name {subsystem_name}	string	Mandatory. Name of the subsystem the core is being added to.

Example

```
sb_add_core -component_name {sb} -core_vlnv {Actel:SystemBuilder:AMBA_SLAVE:0.0.102}
-core_name {AMBA_SLAVE_0} -subsystem_name {FIC0_Master_Subsystem}
```

14.3. sb_configure_core (Ask a Question)

Description

This Tcl command is used to configure cores that are already added to Master/Slave subsystem in the **Peripherals** page of the System Builder component.

```
sb_configure_core \
-component_name {component_name} \
[-core_name {core_name}] \
[-params {param_list}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component.
-params {param_list}	string	Mandatory. List of the configuration parameters for the selected core.

Example

```
sb_configure_core -component_name {sb} -core_name {AMBA_SLAVE_0} -params
{"AMBA_INTERFACE_TYPE:AHBLITE" "NUM_OF_INTERRUPTS:0"}
```

14.4. sb_configure_core_count (Ask a Question)

Description

This command is used to specify the number of instances of a core already added to Master/Slave subsystem in the **Peripherals** page of the System Builder component. All instances will have the same configuration.

Notes: There are two types of cores that can be added using the `sb_add_core` command.

1. One type is of actual cores like CoreI2C, CoreGPIO with fixed core versions available in the **Peripherals** page. If, for example, CoreI2C is added to a subsystem with core_name specified as i2c_peripheral and a core count of 4, then the actual instance names of CoreI2C added will be i2c_peripheral_0, i2c_peripheral_1, i2c_peripheral_2 and i2c_peripheral_3 in the generated design.
2. Second type of cores are Fabric AMBA Slave and Fabric AMBA Master. Adding them to a subsystem will configure the bus core of the subsystem to enable and expose master/slave interfaces of the bus core on the generated System Builder component to be connected to actual masters/slaves in the fabric. If, for example, Fabric AMBA Slave configured as AHBLite(`AMBA_INTERFACE_TYPE:AHBLITE`) is added to a subsystem with core_name specified

© 2025 Microchip Technology Inc. and its subsidiaries

as ahb_slave and a core count of 4, then the CoreAHBLite bus core of that subsystem will be configured to enable and expose 4 AHBLite slave bus interfaces with names ahb_slave, ahb_slave_1, ahb_slave_2 and ahb_slave_3 on the System Builder component to be connected to actual slave peripherals in the fabric.

```
sb_configure_core_count \
-component_name {component_name} \
-core_vlnv {vendor:library:name:version} \
[-core_name {core_name}] \
-subsystem_name {subsystem_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-core_vlnv {vendor:library:name:version}		Mandatory. Version identifier of the core being instantiated in the SmartDesign.
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component. If no details are provided, the instance name will be automatically defined as `<vlnv_core_name>_n` (where `n` starts at 0).
-subsystem_name {subsystem_name}	string	Mandatory. Name of the subsystem the core is being added to.

Example

```
sb_configure_core_count -component_name {sb} -core_name {AMBA_SLAVE_0} -count {2}
```

14.5. sb_move_core (Ask a Question)

Description

This Tcl command is used to move cores from one Subsystem to another in the **Peripherals** page of System Builder. In the exported Tcl description file of a System Builder component, the only scenario where this command will be seen is when the Fabric DDR is used and is moved from its default 'Fabric DDR Subsystem' to a different Subsystem in the **Peripherals** page.
Note: In a System Builder design, if the Fabric External DDR Memory (FDDR) is enabled on the **Device Features** page, then the Fabric DDR Subsystem is automatically enabled with the core Fabric_DDR_RAM added to it in the **Peripherals** page.
A Fabric AMBA Master configured as AXI or AHBLite can be added to the Fabric DDR Subsystem to enable a fabric master to access external DDR memory using FDDR. Alternatively, the Fabric DDR RAM can also be moved (drag and drop) to other Subsystems in the Peripherals page so that the master(s) in that Subsystem will also be able to access external DDR memory using FDDR.

For example, the Fabric_DDR_RAM can be moved from its default 'Fabric DDR Subsystem' to 'MSS FIC_0 - MSS Master Subsystem' (FIC0_Master_Subsystem) which will enable Cortex-M3 in the MSS to access external DDR memory using FDDR via FIC_0 Master address space. The Tcl command `sb_move_core` will be used to capture the action of moving the Fabric_DDR_RAM to a different Subsystem in the exported Tcl description for a System Builder component.

```
sb_move_core \
-component_name {component_name} \
[-core_name {core_name}] \
-subsystem_name {subsystem_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.

© 2025 Microchip Technology Inc. and its subsidiaries

sb_move_core (continued)		
Parameter	**Type**	**Description**
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component.
-subsystem_name {subsystem_name}	string	Mandatory. Name of the subsystem the core is being added to.

Example

```
sb_move_core -component_name {sb} -core_name {Fabric_DDR_RAM} -subsystem_name
{FIC0_Master_Subsystem}
```

14.6. sb_enable_core (Ask a Question)

Description

This Tcl command is used to enable the cores/bus interfaces in various subsystems of the **Peripherals** page of the System Builder component (excluding the MSS Peripherals).

```
sb_enable_core \
-component_name {component_name} \
[-core_name {core_name}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component.

Example

```
sb_enable_core -component_name {sb} -core_name {FIC_0_AMBA_MASTER}
```

14.7. sb_disable_core (Ask a Question)

Description

This Tcl command is used to disable the cores/bus interfaces in various subsystems of the **Peripherals** page of the System Builder component (excluding the MSS Peripherals).

```
sb_disable_core \
-component_name {component_name} \
[-core_name {core_name}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-core_name {core_name}	string	Optional. Name of the instance of the core in the System Builder component.

Example

```
sb_disable_core -component_name {sb} -core_name {FIC_0_AMBA_MASTER}
```


© 2025 Microchip Technology Inc. and its subsidiaries

14.8. sb_enable_peripheral (Ask a Question)

Description

This Tcl command is used to enable the MSS peripherals in various subsystems of the **Peripherals** page of the System Builder component.
Note: By default, MSS_GIO, MSS_USB, MSS_MAC and MSS_CAN are disabled in all devices. Use the `sb_enable_peripheral` command to enable the required peripheral.

```
sb_enable_peripheral \
-component_name {component_name} \
[-peripheral_name {peripheral_name}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}		Mandatory. Name of the system builder component.
-peripheral_name {peripheral_name}		Mandatory. Name of the MSS peripheral instance to be enabled.

Example

```
sb_enable_peripheral -component_name {sb} -peripheral_name {MSS_SPI_0}
```

14.9. sb_disable_peripheral (Ask a Question)

Description

This Tcl command is used to disable the MSS peripherals in various subsystems of the **Peripherals** page of the System Builder component.
Note: By default, MSS_MMUART_0/1, MSS_I2C_0/1, MSS_SPI_0/1 are enabled in all devices. Use the `sb_disable_peripheral` command to disable the peripherals that are not needed.

```
sb_disable_peripheral \
-component_name {component_name} \
[-peripheral_name {peripheral_name}]
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-peripheral_name {peripheral_name}	string	Mandatory. Name of the MSS peripheral instance to be disabled.

Example

```
sb_disable_peripheral -component_name {sb} -peripheral_name {MSS_SPI_0}
```

14.10. sb_configure_peripheral (Ask a Question)

Description

This Tcl command is used to configure the MSS peripherals in various subsystems of the **Peripherals** page of the System Builder component.

```
sb_configure_peripheral \
-component_name {component_name} \
[-peripheral_name {peripheral_name}]
[-params {param_list}]
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-peripheral_name {peripheral_name}	string	Mandatory. Name of the MSS peripheral instance to be enabled.
-params {param_list}	string	Mandatory. List of the configuration parameters for the selected MSS peripheral.

Example

```
sb_configure_peripheral -component_name {sb} -peripheral_name {MM_UART_0} -params
{"MODE:MODE_ASYNC" "MODE_DUPLEX:MODE_FULL" "MODEM_MUX:MUX_IO" "TX_RX_MUX:MUX_IO"
"USE_MODEM:false" }
```

14.11. sb_set_fic_direct_mode (Ask a Question)

Description

This Tcl command is used to set/unset the Fabric Interface Controller (FIC) direct mode in the **Peripherals** page of the System Builder component.

```
sb_set_fic_direct_mode \
-component_name {component_name} \
-mode {true|false}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-mode {true|false}		Mandatory. Set the mode to 'true' to enable the direct mode, else 'false'.

Example

```
sb_set_fic_direct_mode -component_name {sb} -mode {true}
```

14.12. sb_configure_envm (Ask a Question)

Description

This Tcl command is used to specify a `.cfg` file with all clients information in the **ENVM** tab of the **Memories** page in the System Builder component.

```
sb_configure_envm \
-component_name {component_name} \
-cfg_file {file_path}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.
-cfg_file <file_path>	string	Mandatory. Path of the configuration file (.cfg) used to configure the ENVM.

Example

```
sb_configure_envm -component_name {sb} -cfg_file{./ENVM.cfg}
```


© 2025 Microchip Technology Inc. and its subsidiaries

14.13. open_sb_component (Ask a Question)

Description

This Tcl command opens a System Builder component. You must open a System Builder component before you can configure any of its pages using the `sb_*` commands.

Note: To create a System Builder design, the `create_and_configure_core` command must be used, but with empty params. This creates an ungenerated System Builder component with default configuration. Then before configuring any of the System Builder pages, the System Builder component needs to be opened using the `open_sb_component` command.

```
open_sb_component \
-component_name {component_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.

Example

```
open_sb_component -component_name
```

14.14. generate_sb_component (Ask a Question)

Description

This Tcl command is used to generate a System Builder component.

```
generate_sb_component \
-component_name {component_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.

Example

```
generate_sb_component -component_name {sb}
```

14.15. close_sb_component (Ask a Question)

Description

Close a System Builder component. You must close the System Builder component after you are done with configuring all its pages and generating it.

```
close_sb_component \
-component_name {component_name}
```

Arguments

Parameter	Type	Description
-component_name {component_name}	string	Mandatory. Name of the system builder component.

Example

```
close_sb_component -component_name {sb}
```


© 2025 Microchip Technology Inc. and its subsidiaries

15. Simultaneous Switching Noise Analyzer (SSNA) Tcl Commands (Ask a Question)

15.1. ssn_analyzer_noise_report (Ask a Question)

Description

This Tcl command specific to the Simultaneous Switching Noise (SSN) Analyzer. It instructs the SSN Analyzer to generate a noise report of all the used I/Os in the design.

```
ssn_analyzer_noise_report -style {file format} \
                          -filename {full path to the filename}
```

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
style	string	Specifies the file format for the report. Valid values are Text, CSV, and XML.
filename	string	Specifies the full path to filename for the report.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'filename' has illegal value.
None	Required parameter 'filename' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'noise_report [-Style "Text | CSV | XML"] -filename "filename"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example generates a noise report in text format and saves it in `{./report}`.

```
ssn_analyzer_noise_report -style {Text} -filename {./report}
```

15.2. ssn_analyzer_rerun_analysis (Ask a Question)

Description

This Tcl command is specific to the Simultaneous Switching Noise (SSN) Analyzer. It instructs the SSN Analyzer to run the SSN analysis and compute the noise margin numbers.

```
ssn_analyzer_rerun_analysis
```

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

© 2025 Microchip Technology Inc. and its subsidiaries

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
None	None	None

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'param_name' is not defined. Valid command formatting is 'rerun_analysis'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example reruns the SSN Analyzer and computes the Noise Margin number:

```
ssn_analyzer_rerun_analysis
```

15.3. ssn_analyzer_set_dontcare (Ask a Question)

Description

This Tcl command is specific to the Simultaneous Switching Noise (SSN) Analyzer. It sets specific I/Os to the dont_care state or resets dont_care I/Os to non-dont_care. A dont_care I/O is considered as an aggressor only and not as a victim.

```
ssn_analyzer_set_dontcare -io {ioName} \
                          -iobank {ioBankName} \
                          -value {integer value}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
io	string	Specifies the I/O to be dont_care (the don't_care I/O is not be considered as a victim, whereas it is considered as an aggressor for the SSN analysis) or resets dont_care I/Os to non-dont_care.
iobank	string	Specifies the I/O bank name the specified I/O belongs to.
value	integer	Specifies an integer of 0 or 1 where • 1 is used to set an I/O to be dont_care. • 0 is used to reset an I/O to be non-dont_care.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Parameter 'ioBank' is not defined. Valid command formatting is 'set_dontcare -io "io" -iobank "iobank" -value "integer value"'.
None	Required parameter 'iobank' is missing.

© 2025 Microchip Technology Inc. and its subsidiaries

ssn_analyzer_set_dontcare (continued)	
Error Code	**Description**
None	Parameter 'io' has illegal value.
None	Required parameter 'io' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_dontcare -io "io" -iobank "iobank" -value "integer value"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the I/O named "DATA2" in I/O bank "Bank2" to dont_care:

```
ssn_analyzer_set_dontcare -io {DATA2}
                          -ioBank {Bank2}
                          -value {1}
```

The following example sets the dont_care I/O named "DATA1" in I/O bank "Bank3" to non-dont_care:

```
ssn_analyzer_set_dontcare -io {DATA1} \
                          -ioBank {Bank3} \
                          -value {0}
```

15.4. ssn_analyzer_set_pulse_width (Ask a Question)

Description

This Tcl command is specific to the Simultaneous Switching Noise (SSN) Analyzer. It sets the pulse width for SSN calculation.

```
ssn_analyzer_set_pulse_width -pulseWidth {value}
```

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

© 2025 Microchip Technology Inc. and its subsidiaries

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
pulseWidth	decimal	Specifies the threshold value for pulse width. The signal bounce pulse width must reach this value before the signal bounce can be recognized for SSN Analysis. Valid values are 0 ns or 1 ns. A value of 0ns means any signal bounce with pulse width over 0 ns is recognized for SSN analysis. A value of 1 ns means only signal bounces with pulse width at or above 1 ns are recognized for SSN analysis.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	pulseWidth: Invalid argument value: 'value' (expecting integer value).
None	Parameter 'pulseWidth' must be a positive integer value.
None	Parameter 'pulseWidth' has illegal value.
None	Required parameter 'pulseWidth' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_pulse_width -pulseWidth "integer value"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

This Tcl command sets the pulse width threshold value to be 1.0 ns.

```
ssn_analyzer_set_pulse_width -pulseWidth 1.0
```


© 2025 Microchip Technology Inc. and its subsidiaries

15.5. ssn_analyzer_set_static (Ask a Question)

Description

This Tcl command is specific to the Simultaneous Switching Noise (SSN) Analyzer. It sets specific I/Os to the static state or resets static I/Os to be non-static. When the -value is "1", it sets a specific I/O as static to the SSN Analyzer. A static I/O is considered neither as a victim nor as an aggressor. When the -value is "0", this commands resets a static I/O to be non-static.

```
ssn_analyzer_set_static -io {ioName} \
                        -iobank {ioBankName} \
                        -value {integer value}
```

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
io	string	Specifies the I/O name to be marked as static (neither to be considered as a Victim nor as an Aggressor for the SSN analysis) or a static I/O to be non-static.
iobank	string	Specifies the I/O bank name the specified I/O belongs to.

© 2025 Microchip Technology Inc. and its subsidiaries

ssn_analyzer_set_static (continued)		
Parameter	**Type**	**Description**
value	integer	Specifies an integer value of either 0 or 1 where: • 1 is used to set a particular I/O to be static. • 0 is used to reset a static I/O to be non-static.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'iobank' is missing.
None	Parameter 'io' has illegal value.
None	Required parameter 'io' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'set_static -io "io" -iobank "iobank" -value "integer value"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

Example

The following example sets the I/O named "DATA1" in I/O bank "Bank3" to static:

```
ssn_analyzer_set_static -io {DATA1} \
                        -ioBank {Bank3} \
                        -value {1}
```

15.6. ssn_analyzer_summary_report (Ask a Question)

Description

This Tcl command is specific to the Simultaneous Switching Noise (SSN) Analyzer. It instructs the SSN Analyzer to generate a SSN Analyzer summary report of all the used I/Os in the design.

```
ssn_analyzer_summary_report -style {file format} \
                            -file {filename}
```

Supported Die/Package

Family	Die	Package
SmartFusion® 2	M2S150|T|TS	1152 FC
	M2S090|T|TS	676 FBGA
	M2S060|T|TS	676 FBGA
	M2S050|T|TS|T_ES	896 FBGA
	M2S025|T|TS	484 FBGA
	M2S025|T|TS	400 VF
	M2S010|T|TS	484 FBGA

© 2025 Microchip Technology Inc. and its subsidiaries

ssn_analyzer_summary_report (continued)		
Family	**Die**	**Package**
IGLOO® 2	M2GL150|T|TS	1152 FC
	M2GL090|T|TS	676 FBGA
	M2GL060|T|TS	676 FBGA
	M2GL050|T|TS	896 FBGA
	M2GL025|T|TS	484 FBGA
	M2GL025|T|TS	400 VF
	M2GL010|T|TS	484 FBGA

Not Supported Die/Package

Family	Die	Package
RTG4™	RT4G150|L	1657 FCG
PolarFire®	MPF200TS	FCS325
	MPF300TS	FC484/FCS536/FCV484
	MPF300XT	FCG484
	MPF500TS	FC1152/FC784
	RTPF500T|TL|TS|TLS	CG1509

Important: 1 ns pulse width is only supported for MPF300XT/FCG1152.

Arguments

Parameter	Type	Description
style	string	Specifies the file format for the report. Valid values are text, csv, and xml.
file	string	Specifies the filename for the report.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'filename' is missing.
None	Parameter 'param_name' is not defined. Valid command formatting is 'summary_report [-Style "Text | CSV | XML"] -filename "filename"'.

Supported Families

Supported Families
PolarFire®
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

The following example generates a summary report in XML format and saves it in the `summary_report.xml` file in the current directory:

```
ssn_analyzer_summary_report -style {xml} \
                            -filename {./summary_report}
```


© 2025 Microchip Technology Inc. and its subsidiaries

16. HSM Tcl Commands (Ask a Question)

16.1. list_all_hsm_tickets (Ask a Question)

Description

This Tcl command returns the list of HSM tickets.

Note: HSM server name must be specified via HSM_SERVER_NAME DEF variable.

```
list_all_hsm_tickets [-output_file "Output file"] \
                     [-hsm_byte_order "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
output_file	string	Specifies the output file, which contains a list of ticket IDs inside HSM. This parameter is optional.
hsm_byte_order	boolean	-

Return Type	Description
List of strings	Returns a list of HSM tickets.

Error Codes

Error Code	Description
None	Fpeng error: Unable to create/get HSM client.
None	Parameter 'param_name' is not defined. Valid command formatting is 'list_all_hsm_tickets [-output_file "Output file which contains list of ticket IDs inside HSM"] \ [-hsm_byte_order "TRUE | FALSE"]'.

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

PolarFire SoC

Example

The following example displays the list of the tickets in HSM server:

```
puts [list_all_hsm_tickets]
```

16.2. complete_prog_job (Ask a Question)

Description

This Tcl command completes the current open job and generates a Job Status container including cryptographically signed Job Ticket end certifiers and Certificates of Conformance (if enabled) of the programmed devices. It archives ticket data from the HSM database. The resultant Job Status container can be imported into Job Manager and validated using U-HSM. If the job status file is not specified, the information is printed in the log window, and no Job Status container is created for subsequent verification. The HSM Job can only be completed, if the number of devices in each HSM ticket has been exhausted. If devices remain, the job can only be terminated by using the "-terminate" option.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: This command fails, if there are devices left in any HSM ticket, and the terminate option is not used.

```
complete_prog_job [-job_status_file "job status file"] [-terminate "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
job_status_file	string	Specifies full path to the output Job Status container, which contains End-Job Certifier and CofCs. If not specified, information is printed in the log window. This parameter is optional.
terminate	boolean	This option terminates the HSM job even if there are devices left in any HSM ticket. This parameter is optional, if the number of devices in all tickets are exhausted.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Fpeng error: Chain manager not set
None	Parameter 'param_name' is not defined. Valid command formatting is 'complete_prog_job [-job_status_file "job status file"] \ [-terminate "TRUE | FALSE"]'.

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

PolarFire SoC

Example

This example terminates HSM job:

```
complete_prog_job -terminate
```

See Also

- get_job_status

16.3. get_job_status (Ask a Question)

Description

This Tcl command exports status of current open job. The job status contains a number of devices left for each HSM ticket. If job status file is not specified, the information is printed in the log window.

```
get_job_status [-job_status_file "job status file"] [-archive "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
job_status_file	string	Path to the output FlashPro Express job status container. The job status file can be sent to the Job Manager application and Certificates of Conformance (if available) validated using the U-HSM. This parameter is optional.

© 2025 Microchip Technology Inc. and its subsidiaries

get_job_status (continued)		
Parameter	**Type**	**Description**
archive	boolean	Moves the HSM ticket log files from the HSM ticket database to the HSM ticket archive. The archive folder is specified during HSM installation and setup. This parameter is optional. **Note:** If no `job_status_file` is specified, the archive option prints the Certificates of Conformance in the log window without exporting them.

Return Type	Description
Integer	Returns a number of devices.

Error Codes

Error Code	Description
None	Fpeng error: Chain manager not set
None	Parameter 'param_name' is not defined. Valid command formatting is 'get_job_status [-job_status_file "job status file"] \ [-archive "TRUE | FALSE"]'.

Supported Families

PolarFire®

SmartFusion® 2

RTG4™

IGLOO® 2

PolarFire SoC

Example

The following example displays the status of current open job status:

```
puts [get_job_status]
```

See Also

- create_job_project

16.4. process_job_request (Ask a Question)

Description

This Tcl command processes a job request received from Job Manager. It is part of the Job Ticket generation process.

Important: HSM server name must be specified via HSM_SERVER_NAME DEF variable.

```
process_job_request -request_file "job request file" \
                    -reply_file "job reply file" \
            [-overwrite_reply "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
request_file	string	Specifies full file name of job request file. This parameter is mandatory.
reply_file	string	Specifies full file name of job reply file. This parameter is mandatory.

© 2025 Microchip Technology Inc. and its subsidiaries

process_job_request (continued)		
Parameter	**Type**	**Description**
overwrite_reply	boolean	This parameter is optional. The possible value of this parameter are the following: • TRUE or 1: Allows overwriting of any pre-existing reply_file. • FALSE or 0: Does not allow overwriting of any pre-existing reply_file.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'request_file' is missing.
None	Required parameter 'reply_file' is missing.
None	Parameter 'request_file' : the file '/request_file_name' does not exist.
None	overwrite_reply: Invalid argument value: 'value' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Cannot get HSM.
None	Parameter 'param_name' is not defined. Valid command formatting is 'process_job_request -request_file "job request file" \ -reply_file "job reply file" \ [-overwrite_reply "TRUE | FALSE"]'.

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

PolarFire SoC

Example

This example processes a job request:

```
process_job_request \
        -request_file {D:/flashpro_files/jobmgr_project/cm_request.req} \
        -reply_file {D:/flashpro_files/jobmgr_project/cm_reply.rep} \
        -overwrite_reply {TRUE}
```

See Also

- set_hsm_params
- create_job_project

16.5. remove_hsm_tickets (Ask a Question)

Description

This Tcl command removes HSM tickets from the HSM using one of the following methods:

- By specifying the job reply file in which case all tickets that are in the reply file is deleted.
- By specifying each of the ticket IDs value in hexadecimal string.

© 2025 Microchip Technology Inc. and its subsidiaries

Important:

- This command must be used very carefully as it removes HSM tickets, rendering any FlashPro Express jobs based on those tickets to be unusable.
- This command does not require a FlashPro Express project to be created or opened.

```
remove_hsm_tickets [-ticket_ids "Ticket IDs to remove"] \
                   [-reply_file "Reply file containing ticket IDs to remove"] \
           [-force "TRUE | FALSE"]
```

Arguments

Parameter	Type	Description
ticket_ids	hexadecimal	Specifies Hex value of each ticket ID to be removed. This parameter is optional. This parameter can specify several values separated by spaces.
reply_file	string	Specifies full file name of job request file. This parameter is optional.
force	boolean	This parameter is optional.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Specify one of the required arguments: 'ticket_ids' or 'reply_file'.
None	'value' is not a valid hex string. Specify a valid hex string for ticket ID.
None	Parameter 'param_name' is not defined. Valid command formatting is remove_hsm_tickets [-ticket_ids "[Ticket IDs to remove]+"] \ [-reply_file "Reply file containing ticket IDs to remove"] \ [-force "TRUE | FALSE"]'.

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

PolarFire SoC

Example

This example removes the HSM tickets from the HSM by specifying Job reply file:

```
remove_hsm_tickets \
            -reply_file {D:flashpro_filesjobmgr_project12cm_reply.rep}
```

This example removes the HSM tickets from the HSM by specifying each of the ticket IDs value in hexadecimal string.

```
remove_hsm_tickets \
         -ticket_ids {00000000000000000000899f252d9fb55442aa7e
000000000000000000000b6f385c6a9eeca69705c 00000000000000000000ed5702d0b767ba686b82}
```


© 2025 Microchip Technology Inc. and its subsidiaries

16.6. set_hsm_params (Ask a Question)

Description

This Tcl command saves the HSM parameters for the Job Manager application. These parameters remain in effect until overridden by another invocation of this command.

```
set_hsm_params -hsm_server_name hsm_server \
           -u_hsm_server {uuid} \
           -u_master_hsm_uuid {u_master_uuid} \
               -hsm_key_set_dir {keyset_dir} \
               -m_hsm_uuid {m_uuid}
```

Arguments

Parameter	Type	Description
hsm_server_name	string	Name or IP address of HSM server machine.
u_hsm_uuid	string	Specifies User HSM UUID.
u_master_hsm_uuid	string	Specifies User HSM Master UUID.
hsm_keyset_dir	string	Specifies "Keyset" repository location: a directory in which the keyset files are created or used.
m_hsm_uuid	string	Specifies manufacturer HSM UUID. This command saves the HSM parameters for the Job Manager application. This remains in effect until its overridden using this same command.

Return Type	Description
None	None

Error Codes

Error Code	Description
None	Required parameter 'hsm_server_name' is missing.
None	Could not get server information. Check server address and connection and try again.
None	HSM server name cannot be empty.
None	hsm_type_u: Invalid argument value: '' (expecting TRUE, 1, true, FALSE, 0 or false).
None	Warning:Deprecated 'hsm_type_u' parameter is used.
None	FTP login password must be specified along with the user name.
None	Parameter 'hsm_type_u' is not defined. Valid command formatting is 'set_hsm_params -hsm_server_name "HSM server machine name or IP address" \ -u_hsm_uuid "User HSM UUID" \ -u_master_hsm_uuid "User Master HSM UUID" \ -hsm_keyset_dir "path" \ -m_hsm_uuid "Manufacturer HSM UUID"'

Supported Families

PolarFire®

SmartFusion® 2

IGLOO® 2

PolarFire SoC

Example

The following example sets M-HSM parameters:

```
set_hsm_params -hsm_server_name {11.22.33.44} \
               -hsm_type_u {0} \
               -m_hsm_uuid {00000000000000000000000000000000000000000002} \
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
                -ftp_username {hsm} \
                -ftp_password {hsm}
```


© 2025 Microchip Technology Inc. and its subsidiaries

17. Derive Constraints Tcl Commands (Ask a Question)

The derive_constraints utility helps you derive constraints from the RTL or the configurator outside the Libero SoC design environment. To generate constraints for your design, you need the User HDL, Component HDL, and Component Constraints files. The SDC/NDC component constraints files are available under <project>/component/work/<component_name>/<instance_name>/ directory after component configuration and generation.

Each component constraint file consists of the set_component tcl command (specifies the component name) and the list of constraints generated after configuration. The constraints are generated based on the configuration and are specific to each component.

Example 17-1. Component Constraint File for the PF_CCC Core

Here is an example of a component constraint file for the PF_CCC core:

```
set_component PF_CCC_C0_PF_CCC_C0_0_PF_CCC
# Microchip Corp.
# Date: 2021-Oct-26 04:36:00
# Base clock for PLL #0
create_clock -period 10 [ get_pins { pll_inst_0/REF_CLK_0 } ]
create_generated_clock -divide_by 1 -source [ get_pins { pll_inst_0/
REF_CLK_0 } ]
-phase 0 [ get_pins { pll_inst_0/OUT0 } ]
```

Here, create_clock and create_generated_clock are reference and output clock constraints respectively, which are generated based on the configuration.

17.1. Working with the derive_constraints Utility (Ask a Question)

Derive constraints traverse through the design and allocate new constraints for each instance of component based on previously provided component SDC/NDC/PDC files. For the CCC reference clocks, it propagates back through the design to find the source of the reference clock. If the source is an I/O, the reference clock constraint will be set on the I/O. If it is a CCC output or another clock source (for example, SerDes, oscillator), it uses the clock from the other component and reports a warning if the intervals do not match. Derive constraints will also allocate constraints for some macros like on-chip oscillators if you have them in your RTL.

To execute the derive_constraints utility, you must supply a .tcl file command-line argument with the following information in the specified order.

1. Specify device information using the information in section set_device.
2. Specify path to the RTL files using the information in section read_verilog or read_vhdl.
3. Set top level module using the information in section set_top_level.
4. Specify path to the component SDC/NDC files using the information in section read_sdc or read_ndc.
5. Execute the files using the information in section derive_constraints.
6. Specify path to the SDC/PDC/NDC derived constraints file using the information in section write_sdc or write_pdc or write_ndc.

Example 17-2. Execution and Contents of the derive.tcl File

The following is an example command-line argument to execute the derive_constraints utility.

```
$ <libero_installation_path>/bin{64}/derive_constraints derive.tcl
```


© 2025 Microchip Technology Inc. and its subsidiaries

The contents of the `derive.tcl` file:

```
# Device information set_device -family PolarFire -die MPF100T
-speed -1 # RTL files read_verilog -mode system_verilog
project/component/work/txpll0/txpll0_txpll0_0_PF_TX_PLL.v read_verilog -mode
system_verilog {project/component/work/txpll0/txpll0.v} read_verilog -mode
system_verilog {project/component/work/xcvr0/I_XCVR/xcvr0_I_XCVR_PF_XCVR.v}
read_verilog -mode system_verilog {project/component/work/xcvr0/xcvr0.v}
read_vhdl -mode vhdl_2008 {project/hdl/xcvr1.vhd} #Component SDC
files set_top_level {xcvr1} read_sdc -component {project/component/work/
txpll0/txpll0_0/txpll0_txpll0_0_PF_TX_PLL.sdc} read_sdc -component {project/
component/work/xcvr0/I_XCVR/xcvr0_I_XCVR_PF_XCVR.sdc} #Use derive_constraint
command derive_constraints #SDC/PDC/NDC result files write_sdc {project/
constraint/xcvr1_derived_constraints.sdc} write_pdc {project/constraint/fp/
xcvr1_derived_constraints.pdc}
```

17.2. set_device (Ask a Question)

Description

Specify family name, die name, and speed grade.

```
set_device -family <family_name> -die <die_name> -speed <speed>
```

Arguments

Parameter	Type	Description
-family <family_name>	String	Specify the family name. Possible values are PolarFire, PolarFire SoC, IGLOO 2, SmartFusion 2, and RTG4.
-die <die_name>	String	Specify the die name.
-speed <speed>	String	Specify the device speed grade. Possible values are STD or -1.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Required parameter—die is missing	The die option is mandatory and must be specified.
ERR0005	Unknown die 'MPF30'	The value of -die option is not correct. See the possible list of values in option's description.
ERR0023	Parameter—die is missing value	The die option is specified without value.
ERR0023	Required parameter—family is missing	The family option is mandatory and must be specified.
ERR0004	Unknown family 'PolarFire®'	The family option is not correct. See the possible list of values in option's description.
ERR0023	Parameter—family is missing value	The family option is specified without value.
ERR0023	Required parameter—speed is missing	The speed option is mandatory and must be specified.
ERR0007	Unknown speed '<speed>'	The speed option is not correct. See the possible list of values in option's description.
ERR0023	Parameter—speed is missing value	The speed option is specified without value.

Supported Families

PolarFire®
PolarFire SoC

© 2025 Microchip Technology Inc. and its subsidiaries

RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
set_device -family {PolarFire} -die {MPF300T_ES} -speed -1
```

```
set_device -family SmartFusion 2 -die M2S090T -speed -1
```

17.3. read_verilog (Ask a Question)

Description

Read a Verilog file using Verific.

```
read_verilog [-lib <libname>] [-mode <mode>] <filename>
```

Arguments

Parameter	Type	Description
-lib <libname>	String	Specify the library that contains the modules to be added into the library.
-mode <mode>	String	Specify the Verilog standard. Possible values are `verilog_95`, `verilog_2k`, `system_verilog_2005`, `system_verilog_2009`, `system_verilog`, `verilog_ams`, `verilog_psl`, `system_verilog_mfcu`. Values are case insensitive. Default is `verilog_2k`.
filename	String	Verilog file name.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Parameter—lib is missing value	The lib option is specified without value.
ERR0023	Parameter—mode is missing value	The mode option is specified without value.
ERR0015	Unknown mode '<mode>'	The specified verilog mode is unknown. See the list of possible verilog mode in—mode option description.
ERR0023	Required parameter file name is missing	No verilog file path is provided.
ERR0016	Failed due to Verific's parser	Syntax error in verilog file. Verific's parser can be observed in the console above the error message.
ERR0012	set_device is not called	The device information is not specified. Use `set_device` command to describe the device.

Supported Families

PolarFire®
PolarFire SoC
RTG4™

© 2025 Microchip Technology Inc. and its subsidiaries

SmartFusion® 2
IGLOO® 2

Example

```
read_verilog -mode system_verilog {component/work/top/top.v}
```

```
read_verilog -mode system_verilog_mfcu design.v
```

17.4. read_vhdl (Ask a Question)

Description

Add a VHDL file into the list of VHDL files.

```
read_vhdl [-lib <libname>] [-mode <mode>] <filename>
```

Arguments

Parameter	Type	Description
-lib <libname>	—	Specify the library in which the content needs to be added.
-mode <mode>	—	Specifies the VHDL standard. Default is VHDL_93. Possible values are vhdl_93, vhdl_87, vhdl_2k, vhdl_2008, vhdl_psl. Values are case insensitive.
filename	—	VHDL file name.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Parameter—lib is missing value	The lib option is specified without value.
ERR0023	Parameter—mode is missing value	The mode option is specified without value.
ERR0018	Unknown mode '<mode>'	The specified VHDL mode is unknown. See the list of possible VHDL mode in—mode option description.
ERR0023	Required parameter file name is missing	No VHDL file path is provided.
ERR0019	Unable to register `invalid_path.v` file	The specified VHDL file does not exist or does not have read permissions.
ERR0012	`set_device` is not called	The device information is not specified. Use `set_device` command to describe the device.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

```
read_vhdl -mode vhdl_2008 osc2dfn.vhd
```

```
read_vhdl {hdl/top.vhd}
```

17.5. set_top_level (Ask a Question)

Description

Specify the name of the top-level module in RTL.

```
set_top_level [-lib <libname>] <name>
```

Arguments

Parameter	Type	Description
-lib <libname>	String	The library to search for the top-level module or entity (Optional).
name	String	The top-level module or entity name.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Required parameter top level is missing	The top level option is mandatory and must be specified.
ERR0023	Parameter—lib is missing value	The lib option is specified without values.
ERR0014	Unable to find top level <top> in library <lib>	The specified top-level module is not defined in the provided library. To fix this error, the top module or library name must be corrected.
ERR0017	Elaborate failed	Error in the RTL elaboration process. The error message can be observed from console.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
set_top_level {top}
```

```
set_top_level -lib hdl top
```

17.6. read_sdc (Ask a Question)

Description

Read a SDC file into the component database.

```
read_sdc -component <filename>
```


© 2025 Microchip Technology Inc. and its subsidiaries

Arguments

Parameter	Type	Description
-component	—	This is a mandatory flag for `read_sdc` command when we derive constraints.
filename	String	Path to the SDC file.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Required parameter file name is missing.	The mandatory option file name is not specified.
ERR0000	SDC file <file_path> is not readable.	The specified SDC file does not have read permissions.
ERR0001	Unable to open <file_path> file.	The SDC file does not exist. The path must be corrected.
ERR0008	Missing `set_component` command in <file_path> file	The specified component of SDC file does not specify the component.
ERR0009	<List of errors from sdc file>	The SDC file contains incorrect sdc commands. Example when there is an error in `set_multicycle_path` constraint: Error while executing command `read_sdc: in <sdc_file_path> file`: Error in command `set_multicycle_path: Unknown parameter [get_cells {reg_a}]`.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
read_sdc -component {./component/work/ccc0/ccc0_0/ccc0_ccc0_0_PF_CCC.sdc}
```

17.7. read_ndc (Ask a Question)

Description

Read a NDC file into the component database. The command can be used for RTG4 designs using RTG4FCCCECALIB core.

```
read_ndc -component <filename>
```

Arguments

Parameter	Type	Description
-component	—	This is a mandatory flag for `read_ndc` command when we derive constraints.
filename	String	Path to the NDC file.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0001	Unable to open <file_path> file	The NDC file does not exist. The path must be corrected.
ERR0023	Required parameter—AtclParamO_ is missing.	The mandatory option filename is not specified.
ERR0023	Required parameter—component is missing	Component option is mandatory and must be specified.
ERR0000	NDC file '<file_path>' is not readable.	The specified NDC file does not have read permissions.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
read_ndc -component {component/work/ccc1/ccc1_0/ccc1_ccc1_0_RTG4FCCCECALIB.ndc}
```

17.8. derive_constraints (Ask a Question)

Description

Instantiate component SDC/PDC/NDC files into the design-level database.

```
derive_constraints
```

Arguments

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0013	Top-level is not defined	This means that the top-level module or entity is not specified. To fix this call, issue the `set_top_level` command before the `derive_constraints` command.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

© 2025 Microchip Technology Inc. and its subsidiaries

Example

```
derive_constraints
```

17.9. write_sdc (Ask a Question)

Description

Writes a constraint file in SDC format.

```
write_sdc <filename>
```

Arguments

Parameter	Type	Description
<filename>	String	Path to the SDC file will be generated. This is a mandatory option. If the file exists, it will be overwritten.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0003	Unable to open <file path> file.	File path is not correct. Check whether the parent directories exist.
ERR0002	SDC file '<file path>' is not writable.	The specified SDC file does not have write permission.
ERR0023	Required parameter file name is missing.	The SDC file path is a mandatory option and needs to be specified.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
write_sdc "derived.sdc"
```

17.10. write_pdc (Ask a Question)

Description

Writes physical constraints (Derive Constraints only).

```
write_pdc <filename>
```

Arguments

Parameter	Type	Description
<filename>	String	Path to the PDC file will be generated. This is a mandatory option. If the file path exists, it will be overwritten.

© 2025 Microchip Technology Inc. and its subsidiaries

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Messages	Description
ERR0003	Unable to open <file path> file	The file path is not correct. Check whether the parent directories exist.
ERR0002	PDC file '<file path>' is not writeable.	The specified PDC file does not have write permission.
ERR0023	Required parameter file name is missing	The PDC file path is a mandatory option and needs to be specified.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
write_pdc "derived.pdc"
```

17.11. write_ndc (Ask a Question)

Description

Writes NDC constraints into a file. The command can be used for RTG4 designs using RTG4FCCCECALIB core.

```
write_ndc <filename>
```

Arguments

Parameter	Type	Description
filename	String	Path to the NDC file will be generated. This is a mandatory option. If the file exists, it will be overwritten.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Messages	Description
ERR0003	Unable to open <file_path> file.	File path is not correct. The parent directories do not exist.
ERR0002	NDC file '<file_path>' is not writable.	The specified NDC file does not have write permission.
ERR0023	Required parameter _AtclParamO_ is missing.	The NDC file path is a mandatory option and needs to be specified.

Supported Families

PolarFire®

© 2025 Microchip Technology Inc. and its subsidiaries

PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
write_ndc "derived.ndc"
```

17.12. add_include_path (Ask a Question)

Description

Specifies a path to search include files when reading RTL files.

```
add_include_path <directory>
```

Arguments

Parameter	Type	Description
directory	String	Specifies a path to search include files when reading RTL files. This option is mandatory.

Return Type	Description
0	Command succeeded.
1	Command failed. There is an error. You can observe the error message in the console.

List of Errors

Error Code	Error Message	Description
ERR0023	Required parameter include path is missing.	The directory option is mandatory and must be provided.

Note: If the directory path is not correct then add_include_path will be passed without an error. However, `read_verilog/read_vhd` commands will be fail due to Verific's parser.

Supported Families

PolarFire®
PolarFire SoC
RTG4™
SmartFusion® 2
IGLOO® 2

Example

```
add_include_path component/work/COREABC0/COREABC0_0/rtl/vlog/core
```


© 2025 Microchip Technology Inc. and its subsidiaries

18. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
T	08/2025	Updated the get_connected_programmers command.
S	07/2025	The following are the changes made in this revision: • Updated the Arrays section. • Added the set_active_testbench command.
R	05/2025	The following are the changes made in this revision: • Updated the download_latest_cores command. • Updated the export_bitstream_file command. • Updated the export_job_data command. • Updated the get_tool_options command. • Updated the get_tool_state command. • Updated the project_settings command. • Updated the create_generated_clock command. • Updated the get_clocks command. • Updated the set_clock_groups command. • Updated the set_false_path command. • Updated the export_spiflash_image command. • Added the auto_construct_job_project command. • Added the get_connected_programmers command. • Added the debug_mss_ddr command. • Added the export_mss_ddr_training_data command.
Q	09/2024	The following are the changes made in this revision: • Updated the Introduction section. • Updated the add_modelsim_path command. • Updated the set_modelsim_options command. • Updated the set_max_delay command. • Updated the set_min_delay command.

© 2025 Microchip Technology Inc. and its subsidiaries

Revision History (continued)

Revision	Date	Description
P	08/2024	The following are the changes made in this revision: • Updated the Introduction section. • Updated the add_modelsim_path command. • Updated the add_profile command. • Updated the associate_stimulus command. • Updated the export_smart_debug_data command. • Updated the set_modelsim_options command. • Updated the create_hdl_core command. • Updated the PROGRAM_SPI_FLASH_IMAGE command. • Updated the SIM_PRESYNTH command. • Updated the SIM_POSTSYNTH command. • Updated the SIM_POSTLAYOUT command. • Updated the set_input_jitter command. • Updated the set_max_delay command. • Updated the set_min_delay command. • Updated the set_options command. • Added the configure_envm command.
N	02/2024	The following are the changes made in this revision: • Updated the export_smart_debug_data command. • Added the create_smartdesign_testbench command. • Updated the create_generated_clock command.
M	09/2023	The following are the changes made in this revision: • Moved the is_synthesis_enabled command from System Builder to Project Manager Tcl commands. • Updated the set_clock_to_output command.
L	08/2023	The following are the changes made in this revision: • Updated the import_component_data command.
K	05/2023	The following are the changes made in this revision: • Updated the run_tool command.
J	04/2023	The following are the changes made in this revision: • Added the set_external_delay command
H	12/2022	The following are the changes made in this revision: • Updated the export_bitstream_file command. • Added the export_interrupt_map command. • Added the one_way_passcode command. • Updated the SPM_OTP command. • Updated the set_clock_to_output command. • Updated the set_input_jitter command. • Added the complete_debug_job command. • Added the process_job_request command. • Added the set_hsm_params command. • Added the tvs_monitor command.

© 2025 Microchip Technology Inc. and its subsidiaries

Revision History (continued)

Revision	Date	Description
G	08/2022	The following are the changes made in this revision: • Added the Building a Libero Design Using Tcl chapter. • Added the update_and_run_tool command. • Added the sd_apply_presentation command. • Added the sd_reset_layout command. • Updated the PROGRAMMER_INFO command. • Updated the optimize_receiver command.
F	04/2022	The following are the changes made in this revision: • Removed the `smartpower_init_set_enables_options` section. • Added modified_client command. • Updated for bug fixes and other minor enhancements.
E	12/2021	The following are the changes made in this revision: • Added Derive Constraints Tcl Commands. • Added update_fp6_programmers. • Added smartdesign. • Added update_component_version. • Updated for bug fixes and other minor enhancements.
D	08/2021	The document was updated for bug fixes and small enhancements.
C	08/2021	The following changes were made in this revision: • Added set_debug_device. • Added mss_import_register, rescan_programmer, and xcvr_add_register. • Added edit_post_layout_design. • Added sd_set_port_synth_attr, sd_set_net_synth_attr, sd_set_inst_synth_attr, sd_set_comp_synth_attr, sd_remove_port_synth_attr, sd_remove_net_synth_attr, sd_remove_inst_synth_attr, and sd_remove_comp_synth_attr. • Added Simultaneous Switching Noise Analyzer (SSNA) Tcl Commands. • Added HSM Tcl Commands. • Unified Tcl commands for all families into one single document.
B	4/2021	The following changes were made in this revision: • Updated SmartTime Tcl Commands with references to the Libero® SoC Design Suite Tcl Examples GitHub repository. • Updated Project Manager Tcl Commands • Updated HDL Tcl Commands • Updated Command Tools • Updated FlashPro Express Tcl Commands • Added the mss_add_register, mss_read_register, mss_write_register, and mss_export_register SmartDebug Tcl commands.
A	11/2020	Initial Revision

© 2025 Microchip Technology Inc. and its subsidiaries

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-1760-7

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

© 2025 Microchip Technology Inc. and its subsidiaries

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

© 2025 Microchip Technology Inc. and its subsidiaries