cadence®

IEEE 1801 User Guide

Product Version 20.10
March 2020

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

1

Preface

This preface provides information on:

- Scope
- Requirements
- Cadence Products Supporting IEEE 1801
- Typographical Conventions
- Customer Support

1.1 Scope

This manual describes how to capture the power intent for your design using IEEE 1801, a standardized format for specifying power-saving techniques early in the design process, to deliver an end-to-end low-power design solution to IC engineers. It also describes how to build an IEEE 1801 file using different methodologies and gives coding guidelines for how to use IEEE 1801 commands in different Cadence environments.

It contains the following chapters:

- IEEE 1801 Supported Commands and Options
- Creating an IEEE 1801 File
- Coding Guidelines for IEEE 1801 Files

⚠ For the sake of simplicity, the flow described in this document assumes that you are starting with RTL code that does not contain any instantiations of low power logic.

1.2 Requirements

To use this manual, you should be familiar with IC power consumption concepts and low-power simulation.

Additionally, review the following revisions of the IEEE 1801 standard:

- *IEEE Standard 1801-2009*
- *IEEE Standard 1801-2013*
- *IEEE Standard 1801-2015*
- *IEEE Standard 1801-2018*

To download the IEEE 1801-2018 specification, navigate to the following website:
http://standards.ieee.org/findstds/standard/1801-2018.html

1.3 Cadence Products Supporting IEEE 1801

- Cadence® Encounter® Conformal Low Power
- Cadence® Encounter® DFT Architect
- Cadence® Encounter® True Time ATPG
- Cadence® Genus™ Synthesis Solution
- Cadence® Innovus™ Implementation System
- Cadence® Xcelium™ Simulator
- Cadence® Palladium
- Cadence® Tempus® Timing Signoff Solution
- Cadence® Voltus® Power Integrity Solution
- Cadence® JasperGold® Apps

For information on any product option, feature, or package that supports IEEE 1801, contact your local sales or AE contact.

Refer to the product documentation above for information on:

- How IEEE 1801 is used in the product.
- Which IEEE 1801 commands are supported for each product.

- When to read the IEEE 1801 file in the product flow.
- What general command(s) are related to IEEE 1801.

1.4 Typographical Conventions

This manual uses a consistent formatting style to help you locate and interpret information easily.

Table 1.1: Document Conventions

Typeface	Description
`courier font`	Indicates code, commands, option names, or keywords that you must type literally.
courier italic	Indicates user-defined arguments or variables for which you must substitute a name or value.
`|`	The pipe character (OR-bar) is a separator for alternative syntax or parameters.
`[ ]`	Square brackets indicate optional parameters.
`{ }`	Braces indicate that a choice is required from the list of arguments. For example, you must choose one of the following: `{argument1 | argument2 | argument3}`
`...`	Three dots (`...`) indicate that you can repeat the previous argument. • If the three dots are used with brackets (that is, `[argument]...`), you can specify zero or more arguments. • If three dots are used without brackets (`argument...`), you must specify at least one argument, but can specify more.
`#`	The pound sign precedes comments.
`%`	Denotes the UNIX prompt.
click here	Blue underlined text denotes a hyperlink.

1.5 Customer Support

If you have a problem using this documentation, you can submit a customer case to Cadence Support. When doing so, please provide enough information about the problem so that it can be investigated efficiently. Describe the problem in full, give the version of the software you are using, and state the exact circumstances in which the problem occurs.

This section contains

1.5.1 Cases

Cases are your way of giving feedback, asking questions, getting solutions, and reporting problems for a given Cadence product. Unless told otherwise, Cadence Support staff will respond to your case. If Cadence Support cannot answer your question, Cadence research and development personnel will get involved.

It is important to specify the severity level of the service request as accurately as possible. There are three levels of severity:

- **Critical** – You cannot proceed without a solution to the issue.
- **Important** – You can proceed, but you need a solution to the issue.
- **Minor** – You prefer to have a solution, but you can wait for it.

⚠ You can request Support to increase the severity level of an issue. Therefore, do not use **Critical** unless resolution of an issue is absolutely necessary and urgently required.

1.5.2 Using Cadence Online Support

Cadence encourages you to submit cases using Cadence Online Support. With Cadence Online Support you can also track your open cases.

To use Cadence Online Support to submit a case:

1. If you do not yet have a Cadence Online Support account, go to http://support.cadence.com/ and click *Register Now* under the *New User* heading.

   You must provide a valid *HostID* for any Cadence product (Conformal, JasperGold, vManager,

Xcelium, or other). The *HostID* is contained in the SERVER line of your Cadence product license file.

> ⚠ If you already have a Cadence Online Support account, then you only need to update your Cadence Online Support preferences to include a valid HostID for a Cadence product.

2. Log in to Cadence Online Support, and on the upper left side of the page click *Submit Case* under the *Cases* heading.

   A form is presented for submission of your service request. Select *Xcelium* in the *Product* list box and click *Continue*. Follow the online instructions to complete the Service Request.

Creating Group Privileges in Cadence Online Support

Sometimes it is beneficial to view the cases of others on your project.
To create group privileges in Cadence Online Support:

1. Open a Cadence Online Support service request by clicking *Submit Case* under the *Cases* heading.
2. Select a Cadence product in the *Product* list box (for example, Xcelium) and click *Continue*.
3. Fill in the required fields in the form presented.
   Explain in the Stated Problem text box that you want to create a group of users.
4. In the *People to notify upon Case creation* field, include the email addresses of the users you want to have group privileges.

   > ⚠ Each person receiving group privileges must have a Cadence Online Support account.

5. Click *Submit Case* to complete the case.

Visit http://www.cadence.com/support/Pages/default.aspx to learn more about Cadence Global Customer Support and the Support Offerings we provide. For more details about our support process, visit http://www.cadence.com/support/Pages/support_process.aspx.

2

IEEE 1801 Supported Commands and Options

This chapter includes a Cadence product support summary of IEEE 1801 commands and options.

See the sections below for more details:

For more details and the very latest support information, please refer to the low power documentation which corresponds to your product.

2.1 Product Legend

Abbreviation	Definition
AMSD	Cadence® AMSDesigner ®
CLP	Cadence® Encounter ® Conformal Low Power
ET	Cadence® Encounter® DFT Architect
Genus	Cadence® Genus™ Synthesis Solution
Innovus	Cadence® Innovus™ Implementation System
XLM	Cadence® Xcelium™ Simulator
JG	Cadence® JasperGold® Apps
PZ1	Cadence® Palladium® Z1 Verification Computing Platform
PXP	Cadence® Palladium® XP Verification Computing Platform
Tempus	Cadence® Tempus® Timing Signoff Solution
Voltus	Cadence® Voltus® Power Integrity Solution

2.2 Support Legend

Abbreviation	Definition	Notes
*	Beta Quality	Not used for production. For the 14.1 release, CLP LP-EC is beta quality.
C	Cadence Extension	
LA	Limited Access	Production quality. For the 14.1 release, CLP LP Verify support is limited access.
NA	Not Applicable	
NS	Not Supported	May be supported in the future.
PS	Partially Supported	
S	Supported	
V	UPF or IEEE 1801 Version	

2.3 Power Intent Commands

This section documents Cadence support for the following IEEE 1801 power intent commands:

- add_domain_elements (deprecated)
- add_port_state (legacy)
- add_power_state
- add_pst_state (legacy)
- apply_power_model
- associate_supply_set
- begin_power_model (legacy)
- bind_checker
- connect_logic_net
- connect_supply_net
- connect_supply_set
- create_composite_domain
- create_hdl2upf_vct
- create_logic_net
- create_logic_port
- create_power_domain
- create_power_state_group
- create_power_switch
- create_pst (legacy)
- create_supply_net
- create_supply_port
- create_supply_set
- create_upf2hdl_vct
- define_power_model
- describe_state_transition (deprecated)
- end_power_model (legacy)
- find_objects
- load_simstate_behavior
- load_upf
- load_upf_protected (deprecated)
- map_isolation_cell (deprecated)
- map_level_shifter_cell

- map_power_switch
- map_retention_cell
- merge_power_domains (deprecated)
- name_format
- save_upf
- set_design_attributes
- set_design_top
- set_domain_supply_net (legacy)
- set_equivalent
- set_isolation
- set_isolation_control (deprecated)
- set_level_shifter
- set_partial_on_translation
- set_pin_related_supply (deprecated)
- set_port_attributes
- set_power_switch (deprecated)
- set_repeater
- set_retention
- set_retention_control (deprecated)
- set_retention_elements
- set_scope
- set_simstate_behavior
- set_variation
- sim_assertion_control
- sim_corruption_control
- sim_replay_control
- upf_version
- use_interface_cell

2.3.1 add_domain_elements (deprecated)

> This command was deprecated in IEEE 1801-2013.

Description: Add design elements to a power domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NA	S	NA	NS	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-elements`												
1.0	NS	S	S	NA	S	NA	NS	NS	S	S	NA	NA

2.3.2 add_port_state (legacy)

> This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Specify a state for a UPF supply port.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-state`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.3 add_power_state

Description: Specify legal state and voltage values for power domains and supply sets.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	S	S	S	PS[1]	S	S	S	S

1. JasperGold supports providing a primary supply set of a given power domain as a reference for the `add_power_state` command.

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-state`												
2.0	S	S	S	NS	S	S	S	PS	S	S	S	S
`-supply_expr`												
2.0	S	S	S	NS	PS	S	S	PS[1]	S	S	S	S
`-logic_expr`												
2.0	S	S	S	NS	S[2]	S[2]	S	PS	S	S	S	S
`-simstate`												
2.0	S	S	S	NS	NA	NA	S	PS	S	S	NA	NA
`-legal | -illegal`												
2.0	S	NS	NS	NS	NA	NA	S	S	S	S	NA	NA
`-update`												
2.0	S	S	S	NS	S	S	S	NS	S	S	S	S
`-complete`												
2.1	NS	NA	NA	NS	NS	NS	NS	PS	NS	NS	NS	NS
`-supply`												
2.1		S	S				S	PS	S	S		
`-domain`												
2.1		S	S				S	PS	S	S		
`-group`												
3.0							S					

1. For JasperGold, `-supply_expr` supports only pure Verilog expressions. The supply expression can have notations, but JasperGold ignores the voltage description.
2. `-logic_expr` with logic signals is only applicable in tools dealing with dynamic events like simulation/emulation. It not applicable for static tools, because a logic signal state is dynamic. Hence, it is only applicable for the tools, which can detect when the state has changed and does not impact the static tools. The static tools (such as Genus and Innovus) will not do anything with `-logic_expr` on logic nets. The corresponding `-supply_expr` (if any) will be honored appropriately for the state.

2.3.4 add_pst_state (legacy)

This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Specify a power state for each supply net defined in the power state table (PST).

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-pst`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
`-state`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.5 apply_power_model

Description: Connect a power model to a design instance.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	S	S	S	NS	NS	NS	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-elements`												
2.1	NS	S	S	NS	S	S	S	NS	NS	NS	S	S
`-supply_map`												
2.1	NS	S	S	NS	NS	S	S	NS	NS	NS	S	S

2.3.6 associate_supply_set

Description: Associate two or more supply sets.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	S	S	S	PS	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-handle`												
2.0	S	S	S	NS	PS	S	S	PS[1]	S	S	S	S

1. JasperGold does not support level shifter handles.

2.3.7 begin_power_model (legacy)

> This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Define a power model. Superseded by `define_power_model`.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	S	S	S	NS	NS	NS	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-for`												
2.1	NS	S	S	NS	NS	S	S	NS	NS	NS	S	S

2.3.8 bind_checker

⚠ This command and its options apply only to the 1801 Linter.

Description: Insert SystemVerilog checker modules and bind them to design elements.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	NA	NA	NA	NA	NA	NS	NS	NA	NA	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-module												
1.0	NS	NA	NA	NA	NA	NA	NS	NS	NA	NA	NA	NA
-elements												
1.0	NS	NA	NA	NA	NA	NA	NS	NS	NA	NA	NA	NA
-ports												
1.0	NA	NA	NA	NA	NA	NA	NS	NS	NA	NA	NA	NA
-bind_to												
2.0	NA	NA	NA	NA	NA	NA	NS	NS	NA	NA	NA	NA

2.3.9 connect_logic_net

Description: Connect logic nets to one or more ports.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-ports												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-reconnect												
2.1		S	S				S	S	NS	NS		

2.3.10 connect_supply_net

Description: Connect a supply net to one or more ports.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-ports												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-pg_type												
1.0	NS	NS	NS	NS	NS	S	S[1]	NS	S	S	S	S
-vct												
1.0	S	NA	NA	NA	NS	NA	S	NA	S	S	NA	NA
-pins (deprecated in IEEE 1801-2013)												
1.0	NS	NS	NS	NS	NS	S	NS	NS	NS	NS	S	S
-cells												
1.0	NS	NS	NS	NS	NS	NA	NS	NS	NS	NS	NA	NA
-domain												
1.0	NS	NS	NS	NS	NS	S	NS	NS	S	S	S	S
-rail_connection (deprecated in IEEE 1801-2013)												
1.0	NS	NA	NA	NA	NA	NA	NS	NS	NS	NS	NA	NA

1. For Xcelium, the `-pg_type` option is parsed, but has no effect on simulation.

2.3.11 connect_supply_set

Description: Connect a supply set to one or more design elements.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	NS	NS	NS	NS	S	NS	NS	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-connect												
2.0	NS	NS	NS	NS	NS	S	NS	NS	S	S	S	S
-elements												
2.0	S	NS	NS	NS	NS	S	NS	NS	S	S	S	S
-exclude_elements												
2.0	NS	NS	NS	NS	NS	S	NS	NS	S	S	S	S
-transitive												
2.0	NS	NS	NS	NS	NS	S	NS	NS	S	S	S	S

2.3.12 create_composite_domain

Description: Define a composite domain that includes one or more subdomains.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-subdomains												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-supply												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-update												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.3.13 create_hdl2upf_vct

Description: Define a value conversion table for converting HDL logic values to UPF supply net values.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	NA	NA	NS	NS	NA	S	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-hdl_type												
1.0	NS	NA	NA	NS	NS	NA	S	NS	S	S	NA	NA
-table												
1.0	NS	NA	NA	NS	NS	NA	S	NS	S	S	NA	NA

2.3.14 create_logic_net

Description: Define a logic net in the active scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.15 create_logic_port

Description: Define a logic port in the active scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-direction`												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.16 create_power_domain

Description: Define a collection of design elements that share the same primary power supply.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-simulation_only (deprecated in IEEE 1801-2015)												
1.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
-elements												
2.0	NS	S	S	NS	S	S	S	S	S	S	S	S
-elements {.}												
2.1		S	S				S	S	S	S		
-exclude_elements												
2.0	S	S	S	NS	S	S	S	NS	S	S	S	S
-include_scope (deprecated in IEEE 1801-2013)												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-supply												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-supply {extra_supplies_n}												
	NS	S	S	NS	NS	S	NS	NS	NS	NS	S	S
-scope (deprecated in IEEE 1801-2013)												
1.0	S	NA	NA	NA	S	NA	S	S	S	S	NA	NA
-define_func_type												
2.0	NS	NA	NA	NA	NS	NA	NS	NA	S	S	NA	NA
-update												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-available_supplies												
2.1	NS	S	S	NS	NS	S	NS	NS	NS	S	S	S
-power_up_states												
C	NA	NA	NA	NA	NA	NA	NA	NS	S	S	NA	NA
-power_down_states												
C	NA	NA	NA	NA	NA	NA	NA	NS	S	S	NA	NA
-atomic												
2.1		NA	NA				NS	NS	NS	NS		

2.3.17 create_power_state_group

Description: Define a simple name, to be used in the current scope, for a group of related power states.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.0							S					

2.3.18 create_power_switch

Description: Define a power switch.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-output_supply_port												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S
-input_supply_port												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S
-control_port												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S
-on_state												
1.0	S	S	S	NS	NA	NA	S	S	S	S	NA	NA
-off_state												
1.0	S	S	S	NS	NA	NA	S	S	S	S	NA	NA
-supply_set												
2.0	S	S	S	NS	NA	S	S	NS	S	S	S	S
-on_partial_state												
1.0	S	NA	NA	NA	NA	NA	S	S	S	S	NA	NA
-ack_port												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S
-ack_delay												
1.0	S	NA	NA	NA	NA	NA	S	NA	S	S	NA	NA
-error_state												
1.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
-domain												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S
-update												
2.0		S	S				NS	NS	S	S		
-instance												
2.0		S	S				NS	NS	NS	NS		

2.3.19 create_pst (legacy)

This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Define a name for the power state table (PST).

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-supplies`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.20 create_supply_net

Description: Create a supply net in the active scope or in the scope of the specified domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-domain`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
`-reuse`												
1.0	S	S	S	NS	NA	NA	S	S	S	S	NA	NA
`-resolve`												
1.0	S	NA	NA	NA	NA	NA	PS1	PS1	S	S	NA	NA

1. Xcelium and JasperGold do not support `-resolve parallel_one_hot`.
   JasperGold supports `-resolve strong` if you run `set_lpv_enable_upf_extensions on`.

2.3.21 create_supply_port

Description: Create a supply port on an instance in the active scope or in the scope of the specified domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-domain`												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
`-direction`												
1.0	S	S	S	NS	NA	S	S	S	S	S	S	S

2.3.22 create_supply_set

Description: Create a supply set in the active scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-function`												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
`-reference_gnd (deprecated in IEEE 1801-2015)`												
2.0	NS	NA	NA	NS	NS	NA	S	PS[1]	S	S	NA	NA
`-update`												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

1. JasperGold only uses this option as information in GUI and query commands.

2.3.23 create_upf2hdl_vct

Description: Define a value conversion table for converting UPF supply net values to HDL logic values.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0		NA	NA				S	NS	S	S		

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-hdl_type`												
1.0		NA	NA				S	NS	S	S		
`-table`												
1.0		NA	NA				S	NS	S	S		

2.3.24 define_power_model

Description: Define a power model. Replaces `begin_power_model` and `end_power_model`.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.1							S					

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-for`												
3.1							S					

2.3.25 describe_state_transition (deprecated)

> This command was deprecated in IEEE 1801-2015.

Description: Specify the legality of a state transition for a particular object.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-object`												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA
`-from -to`												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA
`-paired`												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA
`-legal | -illegal`												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA

2.3.26 end_power_model (legacy)

> This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Close a power model definition block. Superseded by `define_power_model`.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	S	S	S	NS	NS	NS	S	S

2.3.27 find_objects

Description: Search for and return 1801 design objects that match the criteria in the specified scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	PS	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
scope												
2.0		S	S	NS	S		S	PS "." scope	S	S		
-pattern												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-object_type												
2.0	PS	S	S	NS	PS inst & pin	S	PS[1]	S	S	S	S	S
-direction												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-transitive												
2.0	S	S	S	NS	NS	S	S	S	S	S	S	S
-regexp | -exact												
2.0	S	S	S	NS	NS	NA	S	S	S	S	NA	NA
-ignore_case												
2.0	S	S	S	NS	NS	NA	S	NS	S	S	NA	NA
-non_leaf | -leaf_only												
2.0	S	NA	NA	NS	NS	NS	S	S	S	S	NS	NS

1. Xcelium does not support -object_type *process*.

2.3.28 load_simstate_behavior

Description: Load an 1801 file containing default simstate behavior for a library.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NA	NA	NA	NA	NA	NS	NS	NS	NS	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-file`												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	NS	NS	NA	NA

2.3.29 load_upf

Description: Execute the commands in the specified 1801 file.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-scope												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-version (deprecated in IEEE 1801-2015)												
1.0	PS	S	S	NS	NS	NS	PS	NS	S	S	NS	NS
-hide_globals												
3.0							NS					
-parameters												
3.0							NS					

2.3.30 load_upf_protected (deprecated)

This command was deprecated in IEEE 1801-2018.

Description: Execute the 1801 file in a protected environment which prevents corruption of existing variables.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	PS	PS	NS	NS	NS	PS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-hide_globals`												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
`-scope`												
2.0	NS	S	S	NS	NS	NS	S	NS	NS	NS	NS	NS
`-version`												
2.0	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
`-params`												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.3.31 map_isolation_cell (deprecated)

> This command was deprecated in IEEE 1801-2013.

Description: Specify a certain isolation strategy for one or more library cells.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	S	S	S	NS	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	NS	S	S	NS	S	S	S	NS	S	S	S	S
-elements												
1.0	NS	S	S	NA	NS	NS	NS	NS	NS	NS	NS	NS
-lib_cells												
1.0	NS	S	S	NS	S	S	S	NS	NS	NS	S	S
-lib_cell_type												
1.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-lib_model												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NS	NS	NS
-port												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.3.32 map_level_shifter_cell

Description: Specify a particular level-shifter strategy for a simulation or implementation model.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	S	S	NS	NS	NA	NA	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	NS	S	S	NS	S	S	NS	NS	NA	NA	S	S
-lib_cells												
1.0	NS	S	S	NS	S	S	NS	NS	NA	NA	S	S
-elements												
1.0	NS	S	S	NS	NA	NS	NS	NS	NA	NA	NS	NS

2.3.33 map_power_switch

Description: Specify a power switch model to use for the implementation of the corresponding switch instance.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	NA	S	NS	NS	NA	NA	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	NS	S	S	NS	NA	S	NS	NS	NA	NA	S	S
-lib_cells												
1.0	NS	S	S	NS	NA	NA	NS	NS	NA	NA	NA	NA
-port_map												
2.0	NS	S	S	NS	NA	NS	NS	NS	NA	NA	NS	NS

2.3.34 map_retention_cell

Description: Specify a certain retention strategy for library cells. This command can constrain implementation choices and/or define the functional behavior for verification.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	S	S	S	NS	NA	NA	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	NS	S	S	NS	S	S	S	NS	NA	NA	S	S
-elements												
1.0	NS	S	S	NS	NS	NS	S	NS	NA	NA	NS	NS
-exclude_elements												
1.0	NS	S	S	NS	NS	NS	NS	NS	NA	NA	NS	NS
-lib_cells												
1.0	NS	S	S	NS	NS	S	S	NS	NA	NA	S	S
-lib_cell_type												
1.0	NS	S	S	NS	NS	S	NS	NS	NA	NA	S	S
-lib_model												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NA	NA	NS	NS
-port (name changed to -port_map in IEEE 1801-2013)												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NA	NA	NS	NS
-port_map												
2.1							NS					

2.3.35 merge_power_domains (deprecated)

This command was deprecated in IEEE 1801-2013.

Description: Merge two or more existing power domains into a single domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	NS	NS	NA	NS	NA	NS	NS	NA	NA	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-power_domains`												
1.0	NS	NS	NS	NA	NS	NA	NS	NS	NA	NA	NA	NA
`-scope`												
1.0	NS	NS	NS	NA	NS	NA	NS	NS	NA	NA	NA	NA
`-all_equivalent`												
1.0	NS	NS	NS	NA	NS	NA	NS	NS	NA	NA	NA	NA

2.3.36 name_format

Description: Define the name format of implicitly created objects.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	NS	PS	NS	PS	NS	NS	PS	PS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-isolation_prefix												
1.0	NS	S	S	NS	NS	S	NS	S	NS	NS	S	S
-isolation_suffix												
1.0	NS	S	S	NS	NS	NA	NS	S	NS	NS	NA	NA
-level_shift_prefix												
1.0	NS	S	S	NS	NS	S	NS	PS	NS	NS	S	S
-level_shift_suffix												
1.0	NS	S	S	NS	NS	NA	NS	PS	NS	NS	NA	NA
-implicit_support_suffix												
2.0	NS	NA	NA	NS	NS	NA	NS	PS	NS	NS	NA	NA
-implicit_logic_prefix												
2.0	NS	S	S	NS	NS	NA	NS	PS	NS	NS	NA	NA
-implicit_logic_suffix												
2.0	NS	S	S	NS	NS	NA	NS	PS	NS	NS	NA	NA

2.3.37 save_upf

Description: Create an 1801 file that contains power intent for the specified scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NA	NA	NA	NS	NS	NS	NA	NS	NA	NA	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-scope`												
1.0	NA	NA	NA	NS	NS	NS	NA	NS	NA	NA	NS	NS
`-version`												
1.0	NA	NA	NA	NS	NS	NS	NA	NS	NA	NA	NS	NS

2.3.38 set_design_attributes

Description: Control attributes for IEEE 1801 design elements.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	S	S	NS	S (14.2)	NS	S	NS	S	S	NS	NS

Options Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-elements`												
2.0	NS	S	S	NS	NS	NS	S	NS	S	S	NS	NS
`-exclude_elements`												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NS	NS	NS
`-attribute`												
2.0	NS	S	S	NS	(14.2)	NS	S	NS	S	S	NS	NS
`-models`												
2.0							S	NS	S			

The following sections cover supported Cadence extensions to the `-attribute` option:

- Global Attributes

- NOR Isolation Attributes
- Non-Standard Retention Attributes

Global Attributes

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-attribute add_ps <2.0 | 2.1 | ignore_v>												
C							S	NS				
-attribute default_applies_to <1.0 | 2.0 | 2.1>												
C							S	NS				
-attribute default_power_vct												
C							NS	NS	S			
-attribute default_ground_vct												
C							NS	NS	S			
-attribute domain_interface_def <1.0 | 2.0 | 2.1>												
C							S	NS				
-attribute logic_expr_drives_supply_expr												
C							S					
-attribute override_lib_corruption <TRUE | FALSE>												
C							NS	NS	S			
-attribute support_wildcard <0 | 1>												
C							S	NS				
-attribute top_ports_have_anon_supply <0 | 1>												
C							S	NS				
-attribute terminal_boundary <TRUE | FALSE>												
C							S	NS				
-attribute UPF_dont_touch <TRUE | FALSE>												
C							S	NS	S			

NOR Isolation Attributes

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-attribute {iso_nor TRUE}`												
C	S	NS	NS	NS	NS	NS	S	NS	S	S	NS	NS
`-elements {`*`domain_name.isolation_strategy`*`}`												
C	S	NS	NS	NS	NS	NS	S	NS	S	S	NS	NS

Non-Standard Retention Attributes

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-attribute {ret_cell_type arm_retn_cell_1}`												
C		NS	NS				S	NS	S	S		
`-attribute {ret_cell_type arm_retn_cell_1p}`												
C		NS	NS				S	NS				
`-attribute {ret_cell_type arm_retn_cell_2}`												
C		NS	NS				S	NS				
`-elements {`*`retention_strategy_list`*`}`												
C		NS	NS				S	NS	S	S		

2.3.39 set_design_top

Description: Specify the design top module or use `-testbench` to specify the top-level of the testbench scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NA	NS	NA	S	S	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-testbench												
C	S	NA	NA	NS	NS	NA	S	NS	NS	NS	NA	NA

2.3.40 set_domain_supply_net (legacy)

This command is legacy in the latest IEEE 1801 standard, and is included for backward compatibility.

Description: Specify the primary power and ground net for a power domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-primary_power_net												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-primary_ground_net												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.41 set_equivalent

Description: Specify two or more power nets or supply sets as electrically or functionally equivalent.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	NS	NS	S	NS	S	S	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-functions_only												
2.1	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-nets												
2.1	NS	S	S	NS	NS	NS	S	NS	S	S	NS	NS
-sets												
2.1	NS	S	S	NS	NS	NS	NS	NS	S	S	NS	NS

2.3.42 set_isolation

Description: Define an isolation strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-elements												

1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-exclude_elements												
2.1	NS	S	S				S	S	S	S		
-source												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-sink												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-applies_to												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-applies_to_boundary												
3.0							S					
-applies_to_clamp												
2.0	NS	NA	NA	NS	NS	S	NS	NS	S	S	S	S
-applies_to_sink_off_clamp												
2.0	NS	NA	NA	NS	NS	S	NS	NS	S	S	S	S
-applies_to_source_off_clamp												
2.0	NS	NA	NA	NS	NS	S	NS	NS	S	S	S	S
-isolation_power_net (legacy)												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-isolation_ground_net (legacy)												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-no_isolation												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-isolation_supply_set												
2.0	S	S	S	NS	NS	S	S	S	S	S	S	S
-isolation_signal												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-isolation_sense												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-name_prefix												

2.0	NA	S	S	NS	S	S	NS	S	NS	NS	S	S
-name_suffix												
2.0	NA	S	S	NA	S	NA	NS	S	NS	NS	NA	NA
-clamp_value												
2.0	S	S	S	NS	PS	S	S	S	S	S	S	S
-sink_off_clamp (deprecated in IEEE 1801-2013)												
2.0	NS	NA	NA	NS	S	S	NS	NS	S	S	S	S
-source_off_clamp (deprecated in IEEE 1801-2013)												
2.0	NS	NA	NA	NS	S	S	NS	NS	S	S	S	S
-location												
2.0	NS	S	S	NS	S self parent other	S	S self parent other	PS self	S	S	S	S
-force_isolation												
2.0	NS	S	S	NS	NS	NS	NS	NS	S	S	NS	NS
-instance												
2.0	NS	NA	NA	NS	NS	NA	NS	NS	NS	NS	NA	NA
-diff_supply_only												
2.0	S	S	S	NS	S	13.2	S	S	S	S	13.2	13.2
-target												
C							S	NS	S	NS		
-transitive (deprecated in IEEE 1801-2013)												
2.0	NA	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-update												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-use_equivalence (deprecated in IEEE 1801-2018)												
2.1		NS	NS				S	NS	S	S		

2.3.43 set_isolation_control (deprecated)

This command was deprecated in IEEE 1801-2013.

Description: Specify the control signals for a previously defined isolation strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	PS	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-isolation_signal												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-isolation_sense												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-location												
1.0	NS	S	S	NS	S self parent	S	NS	PS	S	S	S	S

2.3.44 set_level_shifter

Description: Define a level-shifter strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-elements												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-no_shift												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-threshold												
1.0	NS	S	S	NS	S	NA	NS	NS	NS	NA	NA	NA
-force_shift												
2.0	NS	S	S	NS	NS	NS	NS	NS	NS	NA	NS	NS
-source												
2.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-sink												
2.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-applies_to												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-rule												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-location												
1.0	NS	S	S	NS	S self parent other	S	NS	NS	NS	NA	S	S
-name_prefix												
2.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-name_suffix												
2.0	NS	S	S	NS	S	NA	NS	NS	NS	NA	NA	NA
-input_supply_set												
2.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S

-output_supply_set												
2.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S
-internal_supply_set												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NA	NS	NS
-instance												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NA	NS	NS
-transitive												
1.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NA	NS	NS
-update												
1.0	NS	S	S	NS	S	S	NS	NS	NS	NA	S	S

2.3.45 set_partial_on_translation

Description: Define the translation of PARTIAL_ON to FULL_ON or OFF for the purpose of evaluating the power state of supply sets and power domains.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`[OFF | FULL_ON]`												
2.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-full_on_tools`												
2.0	NA	NA	NA	NA	NA	NA	NA	NS	NS	NS	NA	NA
`-off_tools`												
2.0		NA	NA				NA	NS	NS	NS		
`-isolation_sense`												
2.0	NA	NA	NA	NA	NA	NA	NA	NS	NA	NS	NA	NA

2.3.46 set_pin_related_supply (deprecated)

This command was deprecated in IEEE 1801-2013.

Description: Define the related power and ground pair for a library cell.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NS	S	S	NA	S	S	NS	NS	NA	NA	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-pins												
1.0	NS	S	S	NA	S	S	NS	NS	NA	NA	S	S
-related_power_pin												
1.0	NS	S	S	NA	S	S	NS	NS	NA	NA	S	S
-related_ground_pin												
1.0	NS	S	S	NA	S	S	NS	NS	NA	NA	S	S

2.3.47 set_port_attributes

Description: Specify characteristics (attributes) to describe ports on a power domain interface.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	PS	S	S	PS	S	S	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
[-ports | -exclude_ports]												
2.0	S	S	S	NS	S	S	S	PS ports	S	S	S	S
-applies_to												
2.0	S	S	S	NS	S	NS	S	NS	S	S	NS	NS
-domains (deprecated in IEEE 1801-2013)												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-exclude_domains (deprecated in IEEE 1801-2013)												

2.0	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-elements												
2.0	S	S	S	NS	S	NS	S	S	S	S	NS	NS
-exclude_elements												
2.0	S	S	S	NS	NS	NS	NS	NS	S	S	NS	NS
-model												
2.0	PS	S	S	NS	NS	NS	PS	NS	S	S	NS	NS
-attribute												
2.0	NS	S	S	NS	NS	NS	NS	NS	S	S	NS	NS
-clamp_value												
2.0	NS	S	S	NS	NS	NS	S	S	S	S	NS	NS
-sink_off_clamp												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-source_off_clamp												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-receiver_supply												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-driver_supply												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-related_power_port												
2.0	S	S	S	NS	NS	NS	S	NS	S	S	NS	NS
-related_ground_port												
2.0	S	S	S	NS	NS	NS	S	NS	S	S	NS	NS
-related_bias_port												
2.0	NS	S	S	NS	NS	NS	NA	NS	NS	NS	NS	NS
-repeater_supply (deprecated in IEEE 1801-2013)												
2.0	NS	S	S	NS	NS	NS	S	NS	NS	NS	NS	NS
-pg_type												
2.0	NS	S	S	NS	NS	NS	NS	NS	S	S	NS	NS
-transitive (deprecated in IEEE 1801-2013)												

2.0	NS	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-feedthrough												
2.1		S	S		NS		NS	NS	NS	NS		
-unconnected												
2.1		S	S		NS		NS	NS	NS	NS		

s

2.3.48 set_power_switch (deprecated)

> This command was deprecated in IEEE 1801-2013.

Description: Extend a switch by adding input supply ports, output supply ports, and states.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-output_supply_port												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-input_supply_port												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-control_port												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-on_state												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-supply_set												
2.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-on_partial_state												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-off_state												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS
-error_state												
1.0	NA	NS	NS	NS	NA	NS	NA	NS	NS	NS	NS	NS

2.3.49 set_repeater

Description: Define a repeater strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1		S	S		S		S	NS	S	S		

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
2.1							S					
-elements												
2.1		S	S		S		S	NS	S	S		
-exclude_elements												
2.1		S	S		S		S	NS	S	S		
-source												
2.1		S	S		S		S	NS	S	S		
-sink												
2.1		S	S		S		S	NS	S	S		
-use_equivalence												
2.1		NA	NA				NS	NS	S	S		
-applies_to												
2.1		S	S		S		S	NS	S	S		
-applies_to_boundary												
3.0							S					
-repeater_supply_set (changed to -repeater_supply in UPF 3.0)												
2.1		S	S		S		S	NS	S	S		
-repeater_supply												
3.0							S					
-instance												
2.1		NA	NA				NS	NS	NS	NS		
-update												
2.1		S	S				S	NS	S	S		
-name_prefix												
2.1		S	S				NS	NS	NS	NS		
-name_suffix												
2.1		S	S				NS	NS	NS	NS		

2.3.50 set_retention

Description: Define a retention strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-elements												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-exclude_elements												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-retention_power_net (legacy)												
1.0	S	S	S	NS	S	S	S	NS	S	S	S	S
-retention_ground_net (legacy)												
1.0	S	S	S	NS	S	S	S	NS	S	S	S	S
-retention_supply_set												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-no_retention												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-save_signal												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S
-restore_signal												
2.0	S	S	S	NS	S	S	S	S	S	S	S	S

-save_condition												
2.0	S	NA	NA	NA	NS	NA	S	S	S	S	NA	NA
-restore_condition												
2.0	S	NA	NA	NA	NS	NA	S	S	S	S	NA	NA
-retention_condition												
2.0	S	S	S	NA	NS	NA	S	S	S	S	NA	NA
-use_retention_as_primary												
2.0	NS	S	S	NA	NS	NA	NS	NS	S	S	NA	NA
-parameters												
2.0	S	NA	NA	NA	NS	NA	S	NS	NS	NS	NA	NA
-instance												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	NS	NS	NS	NS
-transitive												
2.0	S	NA	NA	NS	NS	NS	NS	NS	S	S	NS	NS
-update												
2.0	S	S	S	NS	S	S	S	NS	S	S	S	S

2.3.51 set_retention_control (deprecated)

This command was deprecated in IEEE 1801-2013.

Description: Specify the control signals and assertions for a previously defined retention strategy.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-save_signal												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-restore_signal												
1.0	S	S	S	NS	S	S	S	S	S	S	S	S
-assert_r_mutex												
1.0	NS	NA	NA	NS	NS	NA	NS	NS	NS	NS	NA	NA
-assert_s_mutex												
1.0	NS	NA	NA	NS	NS	NA	NS	NS	NS	NS	NA	NA
-assert_rs_mutex												
1.0	NS	NA	NA	NS	NS	NA	NS	NS	NS	NS	NA	NA

2.3.52 set_retention_elements

Description: Specify a list of elements whose collective state shall be maintained if retention is applied.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NA	S	NS	NS	NS	NS	NA	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-elements												
2.0	NS	NA	S	NS	NS	NS	NS	NA	NS	NS	NS	NS
-applies_to												
2.0	NS	NA	NS	NS	NS	NS	NS	NA	NS	NS	NS	NS
-exclude_elements												
2.0	NS	NA	S	NS	NS	NS	NS	NA	NS	NS	NS	NS
-retention_purpose												
2.0	NS	NA	NS	NS	NS	NS	NS	NA	NS	NS	NS	NS
-transitive												
2.0	NS	NA	NA	NS	NS	NS	NS	NA	NS	NS	NS	NS
-expand (deprecated in IEEE 1801-2013)												
2.0	NS	NA	NA	NS	NS	NS	NS	NA	NS	NS	NS	NS

2.3.53 set_scope

Description: Specify the current scope.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
1.0	S	S	S	NS	S	S	S	S	S	S	S	S

2.3.54 set_simstate_behavior

Description: Specify the simulation simstate behavior for a model or library.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-lib												
2.0	NS	NA	NA	NA	NA	NA	NS	NS	NS	NS	NA	NA
-model												
2.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
-elements												
2.0	S	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
-exclude_elements												
2.0	S	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA

2.3.55 set_variation

Description: Specify how much a supply source may vary below and above its nominal voltage.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.0							NS					

set_simstate_behavior options

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-supply												
3.0							NS					
-range												
3.0							NS					

2.3.56 sim_assertion_control

Description: Control the behavior of assertions during low-power verification.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.1							NS					

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-control_expr												
3.1							NS					
-controlling_domain												
3.1							NS					
-domain												
3.1							NS					
-elements												
3.1							NS					
-exclude_elements												
3.1							NS					
-model												
3.1							NS					
-type												
3.1							NS					
-transitive												
3.1							NS					

2.3.57 sim_corruption_control

Description: Disable corruption semantics for a specific set or type of design elements.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.1							NS					

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-domain												
3.1							NS					
-elements												
3.1							NS					
-exclude_elements												
3.1							NS					
-model												
3.1							NS					
-type												
3.1							NS					
-transitive												
3.1							NS					

2.3.58 sim_replay_control

Description: Specify the initial blocks to replay when a domain powers up.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
3.1							NS					

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-controlling_domain												
3.1							NS					
-domain												
3.1							NS					
-elements												
3.1							NS					
-exclude_elements												
3.1							NS					
-model												
3.1							NS					
-transitive												
3.1							NS					

2.3.59 upf_version

Description: Specifies the UPF version used to interpret 1801 commands and command options.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	S	S	S	NS	PS	S	S	S	S	S	S	S

Mapping the version value to an LRM Standard:

- 1.0 = UPF 1.0
- 2.0 = IEEE 1801-2009
- 2.1 = IEEE 1801-2013
- 3.0 = IEEE 1801-2015
- 3.1 = IEEE 1801-2018

2.3.60 use_interface_cell

Description: Specify a functional model and a list of implementation targets for isolation and level-shifing.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	S	S	NS	S	NS	NS	NS	NA	NA	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-strategy												
2.0	NS	S	S	NS	S	NS	NS	NS	NA	NA	NS	NS
-domain												
2.0	NS	S	S	NS	S	NS	NS	NS	NA	NA	NS	NS
-lib_cells												
2.0	NS	S	S	NS	S	NS	NS	NS	NA	NA	NS	NS
-map												
2.0	NS	S	S	NS	NS	NS	NS	NS	NA	NA	NS	NS
-elements												
2.0	NS	S	S	NS	NS	NS	NS	NS	NA	NA	NS	NS
-exclude_elements												
2.0	NS	S	S	NS	NS	NS	NS	NS	NA	NA	NS	NS
-applies_to_clamp												
2.0	NS	S	S	NS	NS	NS	NS	NS	NA	NA	NS	NS
-update_any (deprecated in IEEE 1801-2018)												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	NA	NA	NS	NS
-force_function												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NA	NA	NS	NS
-inverter_supply_set												
2.0	NS	NA	NA	NS	NS	NS	NS	NS	NA	NA	NS	NS

2.4 Power Management Cell Commands

This section documents Cadence support for the following IEEE 1801 power-management cell commands:

- define_always_on_cell
- define_diode_clamp
- define_isolation_cell
- define_level_shifter_cell
- define_power_switch_cell
- define_retention_cell

2.4.1 define_always_on_cell

Description: Identify always on cells.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-isolated_pins												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-enable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.4.2 define_diode_clamp

Description: Identify diode cells or cell pins wth diode protection.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	PS	PS	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-data_pins												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-type												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.4.3 define_isolation_cell

Description: Identify isolation cells.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-enable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-always_on_pins												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-non_dedicated												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.4.4 define_level_shifter_cell

Description: Identify level-shifter cells.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-input_voltage_range												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-output_voltage_range												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_input_voltage_range												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_output_voltage_range												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-direction												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-input_power_pin												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-output_power_pin												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-input_ground_pin												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-output_ground_pin												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-enable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-valid_location												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-bypass_enable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

-multi_stage												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.4.5 define_power_switch_cell

Description: Identify a power- or ground-switch cell.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-type												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-stage_1_enable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power_switchable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_switchable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-stage_2_enable												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-always_on_pins												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-gate_bias_pin												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.4.6 define_retention_cell

Description: Identify state retention cells.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
-cells												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-cell_type												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-always_on_pins												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-restore_function												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-save_function												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-restore_check												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-save_check												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

-retention_check												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-hold_check												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-always_on_components												
2.1	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
-power_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS
-ground_switchable												
2.1	NS	S	S	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.5 Non-Standard Commands

This section documents support for non-standard 1801 commands, including Cadence command extensions.

- create_assertion_control
- create_supply_resolution_function
- set_related_supply_net
- set_sim_control

2.5.1 create_assertion_control

Description: Specify the condition for disabling 1801 assertions.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-name`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-assertions`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-domains`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-exclude`												
C	NS	NA	NA	NA	NA	NA	NS	NS	S	S	NA	NA
`-assertion_control`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-type`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-supply_set`												
C	NS	NA	NA	NA	NA	NA	S	NS	S	S	NA	NA
`-lib_model`												
C							S[1]	NS				

1. For Xcelium, when specifying the option `-lib_model`, you cannot use any option other than `-name` with the `create_assertion_control` command.

2.5.2 create_supply_resolution_function

Description: Define a custom supply net resolution function.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
C		NS	NS				S	NS				

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-voltage`												
C		NS	NS				S	NS				
`-state`												
C		NS	NS				S	NS				
`-ignore_off_voltage`												
C		NS	NS				S	NS				

2.5.3 set_related_supply_net

Description: Specify related power and ground supplies for 1801 design objects.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
	S	S	S	NS	S	S	S	S[1]	S	S	S	S

1. JasperGold requires `set_lpv_enable_upf_extensions on`.

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 16.5)	PXP (UXE 17.5)	Tempus	Voltus
-object_list												
	S	S	S	NS	S	S	S	S	S	S	S	S
-power												
	S	S	S	NS	S	S	S	S	S	S	S	S
-ground												
	S	S	S	NS	S	S	S	S	S	S	S	S
-reset												
	NA	NA	NA	NS	S	NA	NA	S	NS	NS	NA	NA

2.5.4 set_sim_control

Description: Specify an action to be taken on selected targets during a simulation run when power is switched off or restored.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-targets`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-action power_up_replay`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-action disable_corruption`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-action {disable_isolation | disable_retention}`												
C	NS	NA	NA	NA	NA	NA	NS	NA	NA	NA	NA	NA
`-controlling_domain`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-domains`												
C	NS	NA	NA	NA	NA	NA	NS	NA	NA	NA	NA	NA
`-instances`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-modules`												
C	S	NA	NA	NA	NA	NA	S	NA	NA	NA	NA	NA
`-libcells`												
C	NS	NA	NA	NA	NA	NA	NS	NA	NA	NA	NA	NA
`-disable_timing_warnings`												
C	NS	NA	NA	NA	NA	NA	NS	NA	NA	NA	NA	NA

2.6 IEEE 1801 Queries

This section documents Cadence support for the following IEEE 1801 query commands:

- query_design_attributes
- query_isolation
- query_port_attributes
- query_port_state
- query_power_domain
- query_power_domain_element
- query_power_state
- query_power_switch
- query_pst
- query_pst_state
- query_retention
- query_supply_net
- query_supply_set
- query_upf

2.6.1 query_design_attributes

Description: Query attributes for a design element or model.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.6.2 query_isolation

Description: Query information for one or more isolation strategies.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0		NA	NA				S	S	NS	NS		

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-domain`												
2.0		NA	NA				S	S	NS	NS		
`-detailed`												
2.0		NA	NA				S	S	NS	NS		

2.6.3 query_port_attributes

Description: Query the attributes for a specified port.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.1							NS	S				

2.6.4 query_port_state

Description: Return the state information for a specified port.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	S	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-state`												
2.0							NS	S				
`-detailed`												
2.0							NS	S				

2.6.5 query_power_domain

Description: Query one or more power domains.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	S	S	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-non_leaf | -all`												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
`-no_elements`												
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS
`-detailed`												
2.0	NS	NS	NS	NS	NS	NS	S	S	NS	NS	NS	NS

2.6.6 query_power_domain_element

Description: Return domain membership information for an instance.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
		NS	NS				NS	S				

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-power`												
		NS	NS				NS	S1				
-non_leaf												
							NS	S				
-all												
							NS	S				
-no_elements												
							NS	S				

1. JasperGold requires `set_lpv_enable_upf_extensions on`.

2.6.7 query_power_state

Description: Return state information for a power domain or supply set.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

2.6.8 query_power_switch

Description: Query information for a UPF power switch.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	S	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-detailed`												
2.0	NS	NS	NS	NS	NS	NS	NS	S	NS	NS	NS	NS

2.6.9 query_pst

Description: Query a power state table (PST).

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	S	NS	NA	NS	NS

2.6.10 query_pst_state

Description: Return state information for the specified power state table (PST).

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	NS	S	NS	NA	NS	NS

2.6.11 query_retention

Description: Query the retention strategies for the specified domain.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0	NS	NS	NS	NS	NS	NS	S	S	NS	NS	NS	NS

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-domain`												
2.0	NS	NS	NS	NS	NS	NS	S	S	NS	NS	NS	NS
`-detailed`												
2.0	NS	NS	NS	NS	NS	NS	S	S	NS	NS	NS	NS

2.6.12 query_supply_net

Description: Query a supply net.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0							NS	PS[1]				

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-domain`												
2.0							NS	S				
`is_supply | -detailed`												
2.0							NS	PS				

1. When using `query_supply_net`, JasperGold does not support the `-detailed` option.

2.6.13 query_supply_set

Description: Query a supply set.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0							NS	PS[1]				

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-detailed`												
2.0							NS	S				
`-transitive`												

2.0							NS	NS				

1. When using `query_supply_set`, JasperGold does not support the `-transitive` option.

2.6.14 query_upf

Description: Find objects in the logical hierarchy, including UPF created or inferred objects.

Command Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
2.0							NS	PS[1]				

Option Support

V	AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
`-direction`												
2.0							NS	S				
`-ignore_case`												
2.0							NS	S				
`-inst_type`												
2.0							NS	S				
`-non_leaf	-leaf_only`											
2.0							NS	NS				
`-object_type`												
2.0							NS	NS				
`-pattern`												
2.0							NS	S				
`-regexp	-exact`											
2.0							NS	S				
`-transitive`												
2.0							NS	S				

1. When using `query_upf`, JasperGold does not support `-object_type`, `-non_leaf`, or `-leaf_only`.

2.7 Wildcard Support

Cadence supports using the asterisk (*) as a wildcard when specifying options to the following 1801 commands:

- connect_supply_net
- set_isolation
- set_level_shifter
- set_repeater
- set_retention
- set_port_attributes

This section provides details on this support by product.

2.7.1 connect_supply_net

The products below support using the * wildcard character in the option:

- `-ports`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
S	S	S	NS	NS	NS	S	S	S	S	NS	NS

2.7.2 set_isolation

The products below support using the * wildcard character in the options:

- `-elements`
- `-exclude_elements`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
S	S	S	NS	NS	NS	PS	S	S	S	NS	NS

2.7.3 set_level_shifter

The products below support using the * wildcard character in the options:

- `-elements`
- `-exclude_elements`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
NS	S	S	NS	NS	NS	NS	NA	NS	NA	NS	NS

2.7.4 set_repeater

The products below support using the * wildcard character in the options:

- `-elements`
- `-exclude_elements`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
NS	S	S	NS	NS	NS	NS	NA	S	NS	NS	NS

2.7.5 set_retention

The products below support using the * wildcard character in the options:

- `-elements`
- `-exclude_elements`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
S	S	S	NS	NS	NS	S	S	S	S	NS	NS

2.7.6 set_port_attributes

The products below support using the * wildcard character in the option:

- `-ports`

AMSD	CLP LP-EC	CLP LP-Verify	ET	Genus	Innovus	XLM	JG	PZ1 (VXE 19.10)	PXP (UXE 17.5)	Tempus	Voltus
S	S	S	NS	NS	NS	S	NS	S	S	NS	NS

3

Creating an IEEE 1801 File

This chapter shows how to create a single, complete IEEE 1801 file for the following low power techniques:

- Creating an IEEE 1801 File for an MSV Design
  - Example: A Complete IEEE 1801 File for MSV
  - Steps to Create the IEEE 1801 File for MSV Design
- Creating an IEEE 1801 File for a PSO Design
  - Example: A Complete IEEE 1801 File for PSO
  - Steps to Create the IEEE 1801 File for Designs Using PSO
- Creating an IEEE 1801 File for a DVFS Design
  - Complete IEEE 1801 File for DVFS Example
  - Steps to Create the IEEE 1801 File for DVFS Design

Each section is self-contained. Click the hypertext links above to navigate through each low power technique. Do note that if you intend to learn more than one technique, you might find some repetition.

Note: Other chapters will show how to use multiple IEEE 1801 files.

> The content of the IEEE 1801 file can change through the design process. The tools in the design process need different information. Therefore you can start the design with an incomplete IEEE 1801 file.

3.1 Creating an IEEE 1801 File for an MSV Design

A Multiple Supply Voltage (**MSV**) design uses multiple supply voltages for the core logic, as shown in Figure 3-1, An Example of MSV Design. The top design and instance `inst_A` operate at voltage `VDD1`, while instance `inst_B` operates on voltage `VDD2`, and instance `inst_C` operates at voltage `VDD3`.

A portion of the design that operates at the same operating voltage (that is, uses the same main power supply belongs to the **power domain** that corresponds to that operating voltage.

A steady state of the design is called a **power state**. Pure MSV designs have only one power state because the operating voltage of the power domains is assumed not to change.

To pass signals between portions of the design that operate at different voltages, **level shifters** are needed. A level shifter is a special cell that has two separate voltage supplies and shifts the input voltage level to the output voltage level. The set_level_shifter command defines the level shifter requirement; this includes which supply sets need to be connected to the level shifters input and output supply pin.

The tools reading 1801 semantics can derive the input and output supplies for level shifters automatically. The input supply set will be the supply set of the driving logic, and the output supply set will be the supply of the receiving logic (if all receivers are on the same supply). For more complex situations, you must explicitly specify the input and output supplies using the set_level_shifter command.

Figure 3.1: An Example of MSV Design

Operating Voltage (OV)	Instances Operating on OV	Libraries Characterized for OV	Power Domains for OV
VDD1: 0.8	top, inst_A	lib1, lib 2	PD1
VDD2: 1.0	inst_B	lib3	PD2
VDD3: 1.2	inst_C	lib4	PD3

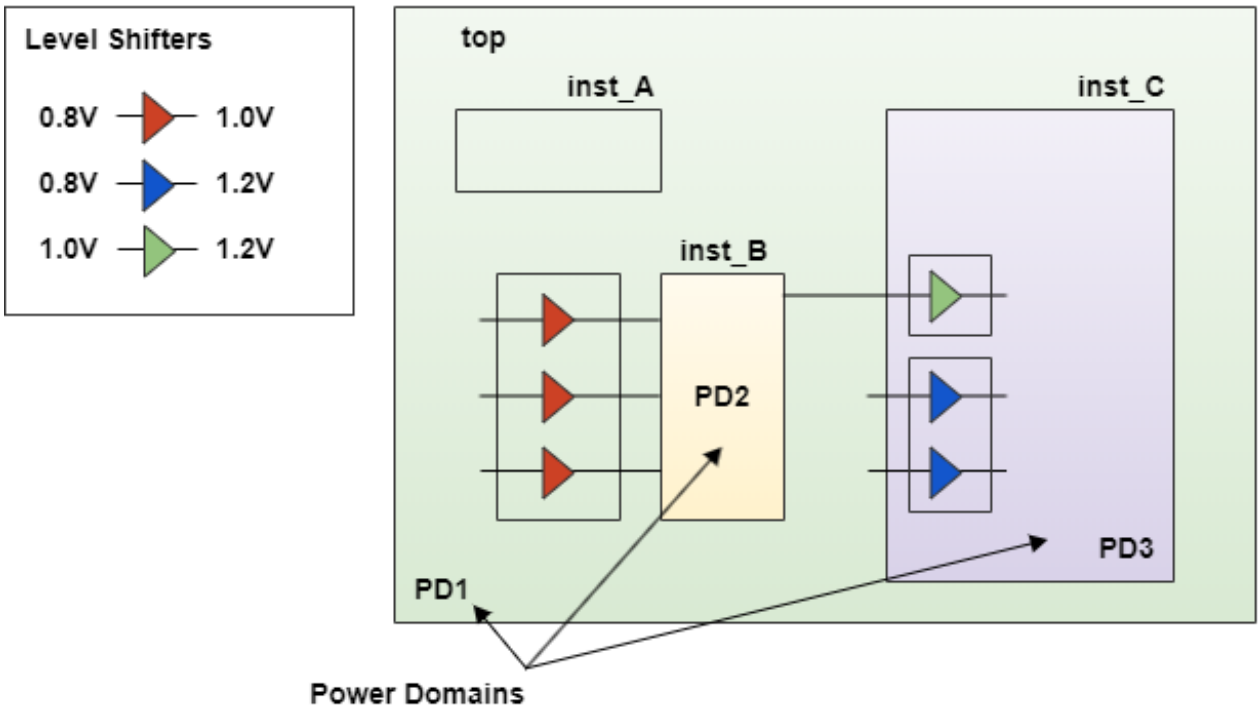


Figure 3-1, An Example of MSV Design, shows the typical operating voltage for each power domain.

3.1.1 Example: A Complete IEEE 1801 File for MSV

```
upf_version 2.0

#Define supply ports and nets
create_supply_port VDD
create_supply_net VDD
connect_supply_net VDD -ports VDD

create_supply_port VDD2
create_supply_net VDD2
connect_supply_net VDD2 -ports VDD2

create_supply_port VDD3
create_supply_net VDD3
connect_supply_net VDD3 -ports VDD3

create_supply_port VSS
create_supply_net VSS
connect_supply_net VSS -ports VSS

#Connect supply nets to power domain supply sets
create_supply_set ss_PD1 -function {power VDD} -function {ground VSS}
create_supply_set ss_PD2 -function {power VDD2} -function {ground VSS}
create_supply_set ss_PD3 -function {power VDD3} -function {ground VSS}

# Create Power Domains
create_power_domain PD1 -include_scope -supply {primary ss_PD1}
create_power_domain PD2 -elements {inst_A inst_B} -supply {primary ss_PD2}
create_power_domain PD3 -elements { inst_C} -supply {primary ss_PD3}

# Create level shifter rules for LD1 -
# which is at a lower voltage than the other domains
set_level_shifter LS_PD1_PD2 -domain PD2 -applies_to inputs \
   -location self \
   -source PD1.primary \
   -input_supply_set PD1.primary -output_supply_set PD2.primary

set_level_shifter LS_PD1_PD3 -domain PD3 -applies_to inputs \
   -location self \
   -source PD1.primary \
   -input_supply_set PD1.primary -output_supply_set PD3.primary

set_level_shifter LS_PD2_PD3 -domain PD3 -applies_to inputs \
   -location self \
   -source PD2.primary \
   -input_supply_set PD2.primary -output_supply_set PD3.primary
```

```
#Define power states for each domain
add_power_state PD1.primary \
   -state ON { -supply_expr {power == `{FULL_ON, 0.8} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL}

add_power_state PD2.primary \
   -state ON { -supply_expr {power == `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL}

add_power_state PD3.primary \
   -state ON { -supply_expr {power == `{FULL_ON, 1.2} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL}

add_power_state PD1 -state PM1 {-logic_expr { PD1.primary == ON && PD2.primary == ON &&
PD3.primary == ON }}
```

3.1.2 Steps to Create the IEEE 1801 File for MSV Design

This section describes the information to include in an IEEE 1801 file for an MSV design. The example shown in Figure 3-1, An Example of MSV Design, is used throughout this section.

The following design-related information captures the power intent and constraints:

- Specifying the Power Domains
- Specifying the Power and Ground Nets and Ports
- Specifying the Supply Set
- Associating the Supply Set to the Power Domain
- Specifying the Voltage States Allowed for Each Domain
- Defining System Level Power States
- Specifying the Rules to Create Level Shifter Logic

This last step is needed for the physical implementation:

- Specifying the Global Connections

3.1.2.1 Specifying the Power Domains

To identify portions of the design that operate on the same voltage, use the create_power_domain command to specify a power domain:

```
create_power_domain power_domain
    [-elements elements_list]
```

For Figure 3-1, An Example of MSV Design, specify three domains:

```
create_power_domain PD1 -include_scope -supply {primary ss_PD1}
create_power_domain PD2 -elements {inst_A inst_B} -supply {primary ss_PD2}
create_power_domain PD3 -elements { inst_C} -supply {primary ss_PD3}
```

The top module belongs to the top level domain. You can specify the top level domain with either the `-include_scope` option or `-elements {.}`.

3.1.2.2 Specifying the Power and Ground Nets and Ports

For IEEE 1801, you can define the supply ports and nets (power and ground) used in a design. The supply ports are external ports that provide a voltage supply to the device. The supply nets are the internal nets that will be used to connect domains and power pins of individual components in the design. Once specified, these power or ground supplies can be associated to power domains, and as connections to special cells.

Define the power and ground supplies using the following syntax:

```
create_supply_port supply_port_name
create_supply_net supply_net_name
```

Then, connect the supply port to the supply net as shown:

```
connect_supply_net supply_net_name -port supply_port_name
```

For a complete description of these commands, see the IEEE 1801-2013 specification.

3.1.2.3 Specifying the Supply Set

For IEEE 1801, a basic building block of the low power intent is the supply set. Because a supply set is a bundle of supply nets, you can use the supply set to simplify the connection of supply nets to power domains and special cells. That is, rather than specifying each supply net individually, you can use a supply set instead.

To define a supply set for each power domain, use the following syntax:

```
create_supply_set ss_name [-function {supply_function supply_net}]*
```

The supply set defines a set of functions such as primary power and primary ground, and then

defines which supply nets provide those functions. For example, the following command specifies that the primary power for `ss_PD1` is provided by the supply net `VDD1`, and the ground function is provided by `VSS`:

```
create_supply_set ss_PD1 -function {power VDD1} -function {ground VSS}
```

3.1.2.4 Associating the Supply Set to the Power Domain

To associate supply sets to power domains, use the `associate_supply_set` command with the following syntax:

```
associate_supply_set ss_name -handle power_domain.handle
```

For example:

```
associate_supply_set ss_PD1 -handle PD1.primary
```

The associated supply set defines that all primary power and ground pins for any cell in `PD1` will be connected to the nets defined in the `ss_PD1` supply set. In this case, it means that all primary power pins in the `PD1` domain will be connected to the `supply_net VDD1`.

3.1.2.5 Specifying the Voltage States Allowed for Each Domain

For IEEE 1801, Cadence recommends using an `add_power_state` command for each power domain. Use this command to define the allowed operating voltage for each specified domain. Although it is not required, it does provide a very clear definition of power intent for more complex designs.

To specify the operating voltages used in the design, use the add_power_state command. For example:

```
add_power_state PD1.primary \
     -state ON {-supply_expr {power == \
     `{FULL_ON, 0.8} && ground == `{FULL_ON, 0.0}} \
     -simstate NORMAL}

add_power_state PD2.primary \
     -state ON {-supply_expr {power == \
     `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0}} \
     -simstate NORMAL}

add_power_state PD3.primary \
     -state ON {-supply_expr {power == \
```

```
    `{FULL_ON, 1.2} && ground == `{FULL_ON, 0.0}} \
    -simstate NORMAL}
```

If your designs have a `pmos` or `nmos` bias, you can also specify these using the `-supply_expr` option.

3.1.2.6 Defining System Level Power States

The `add_power_state` command can also define system level power states. A system level power state defines the interactions between domains rather than the allowed states per domain.

In general, power states are used as follows:

- For verification with IEEE 1801.
- For coverage and to create assertions with simulation.
- To verify all level shifting and isolation requirements in the power intent when running low power design checks.

A pure MSV design is considered to have only one power mode, and each power domain should be associated with a single voltage. A pure MSV design is a design that uses multiple supply voltages but that has no other low power techniques, such as the PSO or DVFS methodologies.

To define the power state, use the `add_power_state` command:

```
add_power_state PD1 \
    -state PM1 {-logic_expr \
    { PD1.primary == ON && PD2.primary == ON && PD3.primary == ON }
```

The system level `add_power_state` command should only specify power domain states with the `-logic_expr` option, not individual power nets. In an hierarchical IEEE 1801 flow, `-logic_expr` could also refer to other lower level system states. In this way, the top-level does not need to specify all the lower level domains.

3.1.2.7 Specifying the Rules to Create Level Shifter Logic

Depending on your technology, you may need level shifters when passing signals...

- From a power domain with a lower voltage to a power domain with a higher voltage.
- From a power domain with a higher voltage to a power domain with a lower voltage.

In both cases, to create a level shifter strategy to use between power domains or a set of pins, use the set_level_shifter command, the most common options and values are:

```
set_level_shifter strategy_name
    -domain domain_name \
    -applies_to <inputs|outputs|both> \
    -location <parent|self|other> \
    -input_supply_set supply_set_name \
    -output_supply_set supply_set_name \
    -threshold <value|list>
```

The level shifter strategy typically applies to the inputs or outputs of a specific domain as specified by the -applies_to option. The `-location` option provides guidance as to where the implementation tools should place the isolation cell: either on the inside of the domain specified (`self`) or outside of the hierarchy (`other`).

The threshold defines when to insert level shifting, this option is often used to avoid level shifting when the voltage difference is small.

For Figure 3-1, An Example of MSV Design, specify the following:

```
set_level_shifter LS_PD2_inputs -domain PD2 \
    -applies_to inputs \
    -location parent \
    -input_supply_set PD1.primary -output_supply_set PD2.primary

set_level_shifter _outputs LS_PD1_PD3 -domain PD3 \
    -applies_to inputs -pins {B} \
    -location self \
    -input_supply_set PD1.primary -output_supply_set PD3.primary

set_level_shifter _outputs LS_PD2_PD3 -domain PD3 \
    -applies_to inputs -pins {a} \
    -location self \
    -input_supply_set PD2.primary
    -output_supply_set PD3.primary
```

3.1.2.8 Specifying the Global Connections

If the design has hard macros with non-default or multiple power and ground connections, then the supply net must be used to specify how to connect global nets, such as power and ground nets. In the example used in this section, there are no special requirements on the design, so this command is not needed. But it is frequently used for memories and other macros in practice.

```
connect_supply_net net_name
    -ports pin_list \
    -domain domain
```

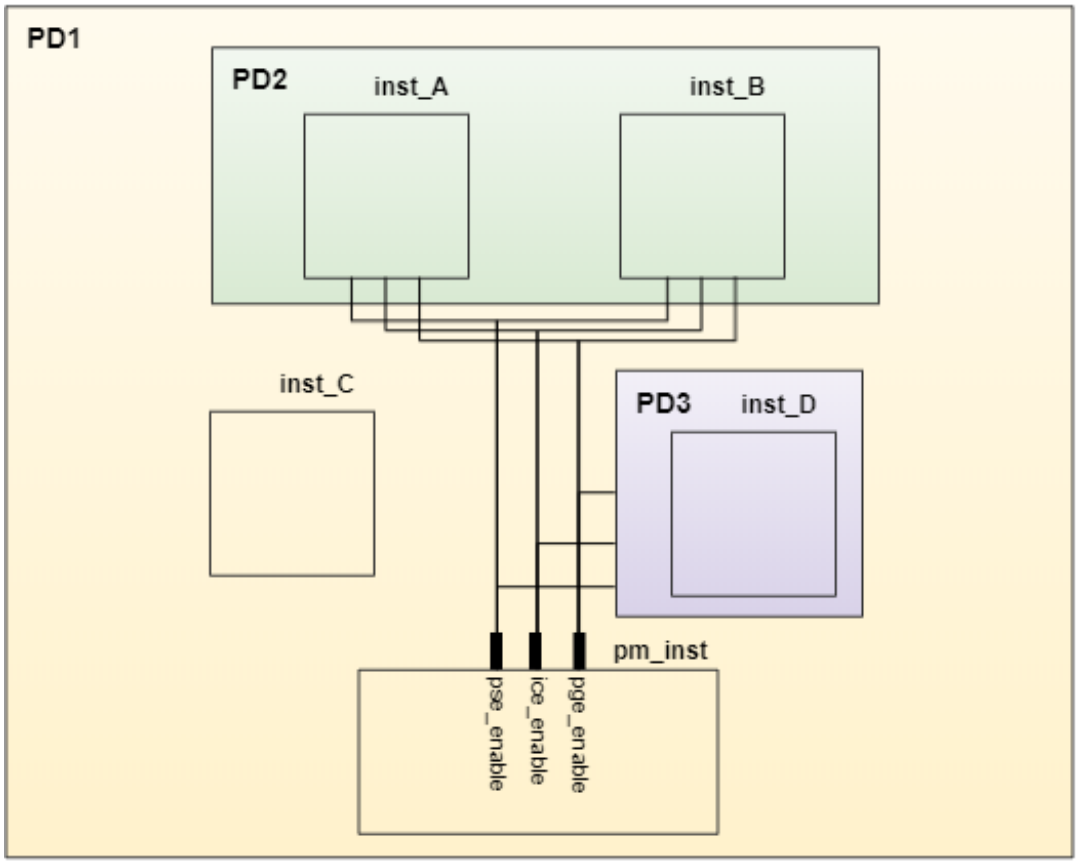
3.2 Creating an IEEE 1801 File for a PSO Design

A design using power shut off (**PSO**) is a design in which some portions can be switched on and off as needed to save leakage and dynamic power.

Logic blocks (hierarchical instances), leaf instances, and pins that use the same *main* power supply and that can be simultaneously switched on or off are said to belong to the same **power domain**. Figure 3-2, An Example of a Design with PSO, has three power domains:

- The top-level of the design, top, and hierarchical instances, inst_C and pm_inst, are always on: they belong to domain PD1.
- Hierarchical instances inst_A and inst_B are always switched on and off simultaneously: they belong to power domain PD2.
- Hierarchical instance inst_D can be switched on and off independently from hierarchical instances inst_A and inst_B: it belongs to power domain PD3.

Figure 3.2: An Example of a Design with PSO



Power domain `PD1` is never powered down. It is called an **unswitched** domain.

Power domains `PD2` and `PD3` can be powered down. They are referred to as **switchable domains**.

A steady state of the design in which some power domains are switched on and some power domains are switched off is called a power state. In a power state, each power domain operates on a specific voltage. Table 3-1, Power States, shows the three power states of the example design.

Table 3.1: Power States

Power Mode	Power Domain		
	PD1	PD2	PD3
PM1	1.1V	1.1V	1.1V
PM2	1.1V	0.0V	1.1V
PM3	1.1V	0.0V	0.0V

⚠ A voltage of 0.0V indicates that the power domain is off.

In order to prevent unknown values in the power domains that are powered down from propagating to the domains that remain powered on, **isolation cells** are needed. These isolation cells are typically placed at the boundaries of the power domains that are powered down or the input boundary of domains receiving a powered down signal. You can, however, also insert isolation cells at the intermediate domain boundaries as required.

To help facilitate powered down blocks in resuming normal operation, **state retention cells** can be used for some sequential cells to keep their previous state prior to power down. Since these cells retain their state, they do not need to be reset or reconfigured after a power cycle. This allows the device to more quickly resume normal operation and, in some cases, continue where it left off in its processing.

For switchable domains, you need to indicate how the power supply is connected and disconnected from the gates.

- For internal switchable domains, you must add **power switch logic.**
- For external switchable domains, the power switch logic is not part of the chip, so a control signal may not be available. In IEEE 1801, the add_power_states command will define states to indicate whether or not domains can power off. In simulation, the testbench will drive the voltage of the switchable pins.

For this example, we are assuming that power domains PD2 and PD3 are internal switchable domains.

Special control signals are used to shut down a power domain, enable state retention, and control the working of the power switch logic. Table 3-2, Signals Controlling the Power Domains, shows the signals used in this example.

Table 3.2: Signals Controlling the Power Domains

Power Domain	Control Signals		
	power switch	**isolation cell**	**state retention cell**
PD1	no control signal	no control signal	no control signal
PD2	ps_enable[0]	ice_enable[0]	pge_enable[0]
PD3	ps_enable[1]	ice_enable[1]	pge_enable[1]

When a domain is switchable, it derives its power from another power domain through either internal or external power switch logic.

In this example, power domains `PD2` and `PD3` derive their power from power domain supply net `VDD`, so `VDD` will become the input voltage for both power switches.

The majority of instances in a power domain are driven by the same power supply. For switchable domains, it is the primary power and ground nets of the (primary) power domain. This supply is automatically attached to the power and ground pins (follow-pins) of all the instances of that domain. In IEEE 1801, the power nets and connections are defined using **supply sets**.

On the other hand, isolation cells and state retention cells are driven by multiple power supplies. These special low power instances can have two sets of power and ground pins, and supply sets for each of these can be explicitly defined in the 1801 file. When using the `create_power_domain` command, the **isolation supply set** defines the default additional supply for isolation cells and the **retention supply set** defines the additional supply for retention cells. These can be overwritten by explicit supply sets defined on the `set_isolation` and `set_retention` commands.

If the design can operate in different power modes, you need to check if the design functions correctly in each of these modes not only at the typical conditions but also when slightly different operating conditions apply. IEEE 1801 files will only define the nominal operating voltage and do not specify any of the related timing information. For a multi-mode, multi-corner timing analysis, additional views and constraints need to be specified in tool-specific design files.

3.2.1 Example: A Complete IEEE 1801 File for PSO

```
upf_version 2.0

#Define supply ports and nets
create_supply_port VDD
create_supply_net VDD
```

```
connect_supply_net VDD -ports VDD

create_supply_port VSS create_supply_net VSS connect_supply_net VSS -ports VSS

create_supply_net VDD2 create_supply_net VDD3

#Connect supply nets to power domain supply sets
create_supply_set ss_PD1 -function {power VDD} -function {ground VSS}
create_supply_set ss_PD2 -function {power VDD2} -function {ground VSS}
create_supply_set ss_PD3 -function {power VDD3} -function {ground VSS}

# Create Power Domains create_power_domain PD1 -include_scope -supply {primary
ss_PD1} create_power_domain PD2 -elements {inst_A inst_B} -supply {primary
ss_PD2} create_power_domain PD3 -elements { inst_C} -supply {primary
ss_PD3} set_port_attributes -ports [find_objects . -pattern * -object_type port] \
   -driver_supply PD1.primary -receiver_supply PD1.primary

#Specify Isolation Constraints on outputs of switchable domains
set_isolation iso2 -domain PD2 -applies_to outputs \
   -isolation_supply_set PD1.primary\
   -location parent \
   -isolation_signal {pm_inst/ice_enable[0]} \
   -clamp_value 0

set_isolation iso3 -domain PD3 -applies_to outputs \
   -isolation_supply_set PD1.primary \
   -location parent \
   -isolation_signal {pm_inst/ice_enable[1]} \
   -clamp_value 0

#Specify retention for switchable domains
set_retention st2 -domain PD2 \
   -save_signal {pm_inst/pge_enable[0] high} \
   -restore_signal {pm_inst/pge_enable[0] low} \
   -retention_supply_set PD1.primary

set_retention st3 -domain PD3 \
   -save_signal {pm_inst/pge_enable[1] high} \
   -restore_signal {pm_inst/pge_enable[1] low} \
   -retention_supply_set PD1.primary

#Define power states for each domain
add_power_state PD1.primary \
   -state ON { -supply_expr {power == `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL}

add_power_state PD2.primary \
```

```
   -state ON { -supply_expr {power == `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL} \
   -state OFF { -supply_expr {power == `{OFF} && ground == `{FULL_ON, 0.0}} -simstate
CORRUPT}

add_power_state PD3.primary \
   -state ON { -supply_expr {power == `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0}} -
simstate NORMAL} \
   -state OFF { -supply_expr {power == `{OFF} && ground == `{FULL_ON, 0.0}} -simstate
CORRUPT}

# Add System Level States
add_power_state PD1 -state PM1 {-logic_expr { PD1.primary == ON && PD2.primary == ON &&
PD3.primary == ON }}
add_power_state PD1 -state PM2 {-logic_expr { PD1.primary == ON && PD2.primary == OFF
&& PD3.primary == ON }} -update
add_power_state PD1 -state PM3 {-logic_expr { PD1.primary == ON && PD2.primary == OFF
&& PD3.primary == OFF }} -update
add_power_state PD1 -state PM4 {-logic_expr { PD1.primary == ON && PD2.primary == OFF
&& PD3.primary == OFF }} -update

#Create power switches
create_power_switch PD2_sw -domain PD2 \
   -output_supply_port {vddo PD2.primary.power} \
   -input_supply_port {vddi PD1.primary.power} \
   -control_port {EN pm_inst/pse_enable[0]} \
   -on_state {full_on vddi {!EN}} \
   -off_state {full_off {EN}}

create_power_switch PD3_sw -domain PD3 \
   -output_supply_port {vddo PD3.primary.power} \
   -input_supply_port {vddi PD1.primary.power} \
   -control_port {EN pm_inst/pse_enable[1]} \
   -on_state {full_on vddi {!EN}} \
   -off_state {full_off {EN}}
```

3.2.2 Steps to Create the IEEE 1801 File for Designs Using PSO

This section describes the information to include in an IEEE 1801 file for a design using the PSO methodology. Since these cells retain their state, they do not need to be reset or reconfigured after a power cycle. This allows the device to more quickly resume normal operation and, in some cases, continue where it left off in its processing. Figure 3-2, An Example of a Design with PSO, is used throughout this section.

- Specifying the Power Domains
- Specifying the Power and Ground Nets and Ports
- Specifying Supply Sets
- Associate the Supply Set to the Power Domains
- Specifying the Voltage States Allowed for Each Domain
- Defining System Level Power States
- Specifying the Rules to Create Isolation Logic
- Specifying the Rules to Create State Retention Logic
- Specifying the Power Switch Logic
- Updating the Rules with Information for Implementation
- Specifying Low Power Cell Attributes

3.2.2.1 Specifying the Power Domains

To identify portions of the design that operate on the same voltage and that can be simultaneously switched on or off, use the `create_power_domain` command with the following syntax to specify a power domain and its associated instances:

```
create_power_domain -name power_domain
    [-elements instance_list]
    [-exclude_elements instance_list]
    [-include_scope]
    [-supply {supply_set_handle [supply_set_ref]]
```

For Figure 3-2, An Example of a Design with PSO, specify three power domains:

```
create_power_domain PD1 -include_scope
```

```
create_power_domain PD2 -elements {inst_A inst_B}
create_power_domain PD3 -elements {inst_C}
```

3.2.2.2 Specifying the Power and Ground Nets and Ports

For IEEE 1801, you can define the supply ports and nets (power and ground) used in a design. The supply ports are external ports that provide a voltage supply to the device. The supply nets are the internal nets that will be used to connect domains and power pins of individual components in the design. Once specified, these power or ground supplies can be associated to power domains, and as connections to special cells.

Define the power and ground supplies using the following syntax:

```
create_supply_port supply_port_name
create_supply_net supply_net_name
```

Then, connect the supply port to the supply net as shown:

```
connect_supply_net supply_net_name -port supply_port_name
```

For a complete description of these commands, see the IEEE 1801-2013 specification.

3.2.2.3 Specifying Supply Sets

For IEEE 1801, a basic building block of the low power intent is the supply set. Because a supply set is a bundle of supply nets, you can use the supply set to simplify the connection of supply nets to power domains and special cells. That is, rather than specifying each supply net individually, you can use a supply set instead.

To define a supply set for each power domain, use the following syntax:

```
create_supply_set ss_name [-function {supply_function supply_net}]*
```

The supply set defines a set of functions such as primary power and primary ground, and then defines which supply nets provide those functions. For example, to specify that the primary power for `ss_PD1` is provided by the supply net `VDD` and the ground is provided by `VSS`, use the following `create_supply_set` command:

```
create_supply_set ss_PD1 -function {power VDD} -function {ground VSS}
create_supply_set ss_PD2 -function {power VDD2} -function {ground VSS}
create_supply_set ss_PD3 -function {power VDD3} -function {ground VSS}
```

⚠ `PD2` and `PD3` will receive their power function from `VDD2` and `VDD3` respectively.

3.2.2.4 Associate the Supply Set to the Power Domains

To associate supply sets to power domains, use the `associate_supply_set` command with the following syntax:

```
associate_supply_set ss_name -handle power_domain.handle
```

For example:

```
associate_supply_set ss_PD1 -handle PD1.primary
```

This associated supply set defines that all primary power and ground pins for any cell in `PD1` will be connected to the nets defined in the `ss_PD1` supply set. In this case, it means that all primary power pins in the `PD1` domain will be connected to the `supply_net VDD`.

3.2.2.5 Specifying the Voltage States Allowed for Each Domain

For IEEE 1801, Cadence recommends using an `add_power_state` command for each power domain. Use this command to define the allowed operating voltage for each specifed domain. Although it is not required, it does provide a very clear definition of power intent for more complex designs.

To specify the operating voltages used in the design, use the `add_power_state` command. For example:

```
add_power_state PD1.primary \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 0.8} && ground == `{FULL_ON, 0.0}} \
    -simstate NORMAL}

add_power_state PD2.primary \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 1.0} && ground == `{FULL_ON, 0.0} \
    -simstate NORMAL} \

-state OFF { -supply_expr {power == \
    `{OFF} && ground == `{FULL_ON, 0.0}} \
    -simstate CORRUPT}

add_power_state PD3.primary \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 1.2} && ground == `{FULL_ON, 0.0}} \
    -simstate NORMAL} \
```

```
-state OFF { -supply_expr {power == \
    `{OFF} && ground == `{FULL_ON, 0.0}} \
    -simstate CORRUPT}
```

The `add_power_state` for domain `PD2` has two defined states:

- `ON` is the powered up state, and is defined by specifying that the power and ground net are in the `FULL_ON` state.
- `OFF` specifies that the power is shutoff and the primary power net is in the `OFF` state.

Both `FULL_ON` and `OFF` are keywords for IEEE 1801 that define the functional state of the supply.

Note: If your designs have a pmos or nmos bias, you can also specify these using the `-supply_expr` option.

3.2.2.6 Defining System Level Power States

The `add_power_state` command can also define system level power states. A system level power state defines the interactions between domains rather than the allowed states per domain.

The IEEE 1801 specification allows for both legal and illegal power states. Legal states are used extensively by checking and verification tools to ensure that proper isolation and level shifting has been inserted into a design. Typically, Illegal states are used by simulation tools to detect errors in power management logic.

For a PSO design, the `add_power_state` command must define all the valid power state combinations (on and off). Table 3-1, Power States, illustrates a design with three valid states, despite having only two power shutoff domains: If `PD3` is off when `PD2` is always off, this requirement should be captured by the states.

Example:

```
add_power_state PD1 \
    -state PM1 {-logic_expr { PD1.primary == ON \
    && PD2.primary == ON && PD3.primary == ON }}

add_power_state PD1 \
    -state PM2 {-logic_expr { PD1.primary == ON \
    && PD2.primary == OFF && PD3.primary == ON }} -update

add_power_state PD1 \
    -state PM3 {-logic_expr { PD1.primary == ON \
    && PD2.primary == OFF && PD3.primary == OFF }} -update
```

```
add_power_state PD1 \
    -state PM4 {-logic_expr { PD1.primary == ON \
    && PD2.primary == ON && PD3.primary == OFF }} -illegal -update
```

⚠ The `add_power_state` commands above refer to the power domain states only. This is the recommended style for system level power states. If this were an hierarchical IEEE 1801 flow, the `-logic_expr` option could also refer to other lower level system states. In this way, the top-level states can utilize a higher level of abstraction and do not need to know all the low level details of each domain or subsystem.

3.2.2.7 Specifying the Rules to Create Isolation Logic

Isolation is used to prevent unknown states from propagating from a domain that is powered down to powered on logic in the design. In IEEE 1801, the power intent for inserting isolation cells is called an **isolation strategy**. An isolation strategy typically defines the value of a pin when the driver of that pin powers off. To define when isolation cells must be added or to specify which pins must be isolated, use the `set_isolation` command. The syntax below lists common options and values:

```
set_isolation strategy_name
    -domain domain_name
    -isolation_signal expression
    [-applies_to {inputs|outputs|both}]
    [-elements pin_list]
    [-clamp_value { 0 | 1 | latch | Z }
    [-isolation_supply_set supply_set_list]
    [-location {self other parent fanout}]
    [-diff_supply_only]
    [-source <source_domain_name | source_supply_ref>]
    [-sink <sink_domain_name | sink_supply_ref>]
```

The isolation strategy is always related to a power domain. By default, it applies to all the inputs and outputs of the domain specified in the strategy, but you can filter it based on a number of parameters.

A specific list of pins or instances:

- `-diff_supply_only` — Only isolate if the driver and receiver of the pin are from different power supplies. This option prevents unnecessary isolation insertion.
- `-source`/`-sink` — Specify constraints based on the driver or receiver of the pin. These options

give more flexibility and control to the isolation, but are uncommon in most designs.

The functionality of the isolation is defined by the `-clamp_value` and `-isolation_signal` options.

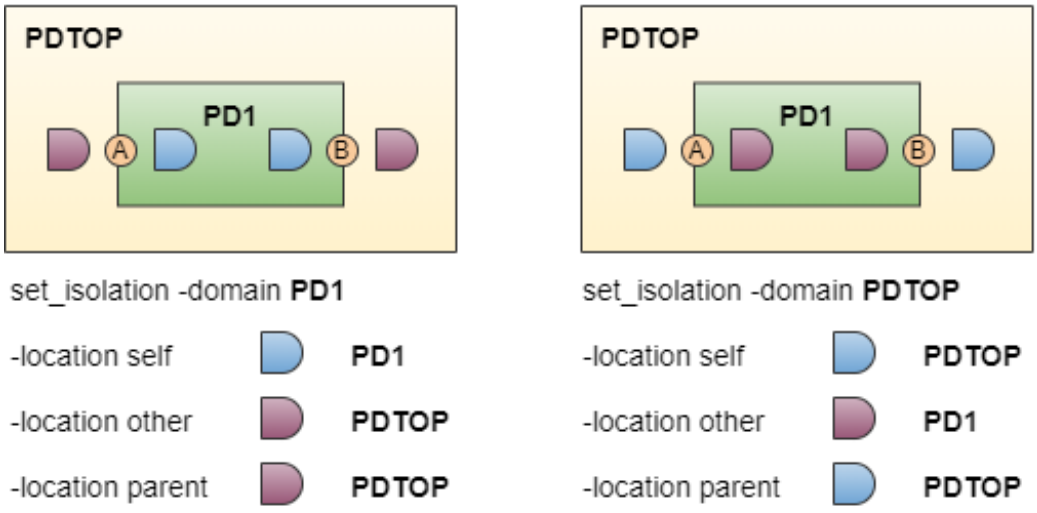
- `-clamp_value` defines the logic value that the pin should take when isolated. The most common arguments are `0`, `1`, and `latch`.
- `-isolation_signal` defines the control for the isolation When the isolation signal is active the specified value will be driven on the pin.

`-location` specifies where in the hierarchy the isolation should be inserted. Typically, this will be either inside of the specified domain (`self`) or on the pin just outside the specified domain (`other`). See Figure 3-3, Using the Location Option, for an illustrated explanation.

Isolation always applies to a pin on the interface between two domains. Pins `A` and `B` are on the interface between `PDTOP` and `PD1`. The `-location` option defines what side of that pin the isolation should be placed

- *Self*: Place the cell in the same domain as that specified by the `-domain` option
- *Other*: Place the cell in the other domain (the one not specified by the `-domain` option)
- *Parent*: Place the isolation cell at a higher level of the hierarchy

Figure 3.3: Using the Location Option



`-isolation_supply_set` defines the power supply that will be connected to the secondary power pins, if any, for the inserted isolation cell.

Typically, isolation logic is needed to prevent unknown signals going from a power domain being switched off to a power domain that remains on. In some cases, the input of a power shutoff domain requires a stable input value. In this case, additional isolation is required on that input pin. This is not the typical case, but is sometimes required for certain types of memories.

Referring to Figure 3-2, An Example of a Design with PSO, isolation logic will be needed in power modes 2 and 3 for any nets going from power domain `PD2` to `PD1` and `PD3`, and for any nets going from power domains `PD3` and `PD2` to `PD1`. For example:

```
set_isolation iso2 -domain PD2 -applies_to outputs \
    -isolation_supply_set PD1.primary \
    -location parent \
    -isolation_signal {pm_inst/ice_enable[0]} \
    -clamp_value 0

set_isolation iso3 -domain PD3 -applies_to outputs \
    -isolation_supply_set PD1.primary \
    -location parent \
    -isolation_signal {pm_inst/ice_enable[1]} \
    -clamp_value 0
```

3.2.2.8 Specifying the Rules to Create State Retention Logic

State retention is a design methodology that allows a set of registers to be preserved through a power on/power off sequence. There are two common applications for state retention:

- Preserve Configuration Registers — Preserve a specific set of configuration registers through the power sequence. This allows the domain to start up with the same configuration and speeds up the power on cycle. Without retention, the device would need to be reconfigured and in many cases this adds overhead to the system software.
- Full restore of the domain — Allow the device to start up in the exact state that it was powered off. That is, any operations in the domain essentially pause during power shutoff and then resume immediately on power up.

  In many cases, you can save ALL registers in the design. This ensures that a full *known* state of the device is restored. However, IEEE 1801 does not require that all registers are maintained, and you may attempt to optimize a partial set of registers. This partial set provides a smaller area for the design but runs some risk of missing a register and having incomplete or incorrect restoration.

To define a rule for replacing selected registers or all registers in the specified power domain with

state retention registers, use the `set_retention` command. Common options are:

```
set_retention
    strategy_name
    -domain power_domain
    [-elements element_list]
    [-exclude_elements element_list]
    [-save_signal {logic_net <high | low | posedge | negedge>}
    [-restore_signal {logic_net <high | low | posedge | negedge>}]
    [-restore_condition expr]
    [-save_condition expr]
    [-retention_condition expr]
    [-retention_supply supply_set_ref]
```

The retention strategy is defined per domain and specifies which elements should be retained and the control over that retention. The `-save_signal` and `-restore_signal` options define the control signal that triggers the save and restore operations respectively. The `-save_condition`, `-restore_condition`, and `-retention_condition` options are used by simulation and static checking as additional checks on the operation of the retention. For instance, to ensure the clock is held low at the time of the save, specify `-save_condition {!clk}`.

The retention cells have a separate retention supply that is kept active when the domain is off. This retention supply is specified with the `-retention_supply_set` option.

For Figure 3-2, An Example of a Design with PSO:

```
set_retention st2 -domain PD2 \
    -save_signal {pm_inst/pge_enable[0] high} \
    -restore_signal {pm_inst/pge_enable[0] low} \
    -retention_supply_set PD1.primary

set_retention st3 -domain PD3 \
    -save_signal {pm_inst/pge_enable[1] high} \
    -restore_signal {pm_inst/pge_enable[1] low} \
    -retention_supply_set PD1.primary
```

3.2.2.9 Specifying the Power Switch Logic

In a PSO design, one must define the power switch that controls the shutoff behavior of the domain. This includes the input and output power nets, the control ports, and each allowed state for the power switch.

To specify the power switch, use the `create_power_switch` command with the following syntax:

```
create_power_switch
    -domain power_domain
    -output_supply_port {port supply_net_name}
    {-input_supply_port {port supply_net_name}}*
    {-control_port {port_name [net_name]}*
    {-on_state {state_name input_supply_port {boolean_expression}}}*
    [{-off_state {state_name {boolean_expression}}]*
    [-ack_port {port_name net_name [logic_value]}]*
    [-ack_delay {port_name delay}]*
```

The typical power switch will have a single control signal that defines when the input supply will be connected to the output supply. Use the `-on_state` option to define the control logic. With advanced syntax, the `create_power_switch` command allows for the definition of much more complicated switches, such as multiple input voltages and/or multiple control signals.

The `-ack_port` option defines an acknowledge signal that generates a value when the output power has been fully restored. This is often used as an input to other power switches or as an input to an FSM waiting for the domain to be fully restored before proceeding to the next state.

For Figure 3-2, An Example of a Design with PSO:

```
create_power_switch PD2_sw -domain PD2 \
    -output_supply_port {vddo PD2.primary.power} \
    -input_supply_port {vddi PD1.primary.power} \
    -control_port {EN pm_inst/pse_enable[0]} \
    -on_state {full_on vddi {!EN}} \
    -off_state {full_off {EN}}

create_power_switch PD3_sw -domain PD3 \
    -output_supply_port {vddo PD3.primary.power} \
    -input_supply_port {vddi PD1.primary.power} \
    -control_port {EN pm_inst/pse_enable[1]} \
    -on_state {full_on vddi {!EN}} \
    -off_state {full_off {EN}}
```

3.2.2.10 Updating the Rules with Information for Implementation

The user can optionally specify the exact set of cells to map to for isolation, retention, and for power switch insertion. To do this use the `map_*` commands in IEEE 1801:

```
map_retention_cell st2 -domain PD2 -lib_cells DRFF
map_retention_cell st3 -domain PD3 -lib_cells DRFF
use_interface_cell iso1 -strategy iso2 -domain PD2 -lib_cells ISOLNX2M
```

```
use_interface_cell iso2 -strategy iso3 -domain PD3 -lib_cells ISOLNX2M
map_power_switch PD2_sw -domain PD2 -lib_cells {HEAD32M}
map_power_switch PD3_sw -domain PD3 -lib_cells {HEAD32M}
```

3.2.2.11 Specifying Low Power Cell Attributes

The 1801-2013 specification allows for the definition of library attributes in two different manners:

- Native 1801 commands
- Attributes in the Liberty definition of the technology

The IEEE-2013 recommendation is to use native 1801 commands because that format is entirely within the 1801 specification. While the Liberty format is the de facto standard for library information, it is not a format controlled by the IEEE Standards Association or any other open standards body.

For PSO designs, the common low power cells are:

```
define_always_on_cell

define_isolation_cell

define_retention_cell
```

Example of 1801 low power cells:

```
define_always_on_cell -cells LP_AON -power VDD -ground VSS

define_isolation_cell -cells ISO_ON \
    -power VDD -ground VSS \
    -enable EN \
    -valid_location on

define_isolation_cell cells ISO_OFF \
    -power VDD -ground VSS \
    -power_switchable VDD_SW \
    -enable EN \
    -valid_location off

define_retention_cell -cells RET1 \
    -power VDD -ground VSS \
    -power_switchable VDD_SW \
    -clock_pin CK \
    -always_on_pin Q \
    -save_function {SV posedge}
```

3.3 Creating an IEEE 1801 File for a DVFS Design

Dynamic voltage frequency scaling (DVFS) reduces the power in the chip by scaling down the voltage and frequency when peak performance is not required.

A design using DVFS can be seen as a special case of an MSV design operating in multiple design modes.

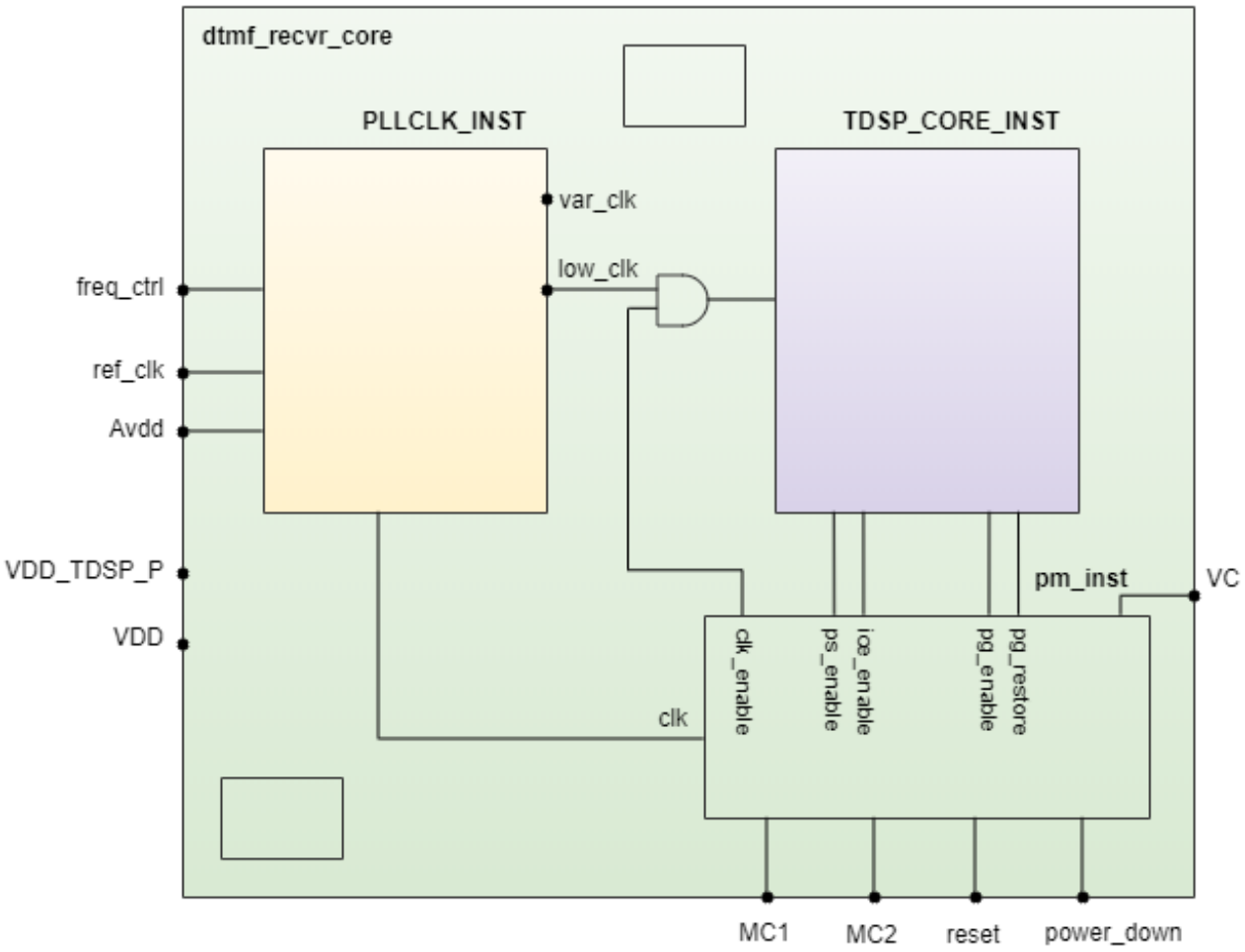
- In a pure MSV design different portions of the design operate on different voltages and these portions remain operating at their respective operating voltage.
- In a DVFS design, some portions can dynamically change to other voltages depending on the design mode or can even be switched off.
  Consequently, a DVFS design must satisfy different constraints in different design modes.

Requirements for DVFS Designs

DVFS designs require variable power supply(ies) that can generate the required voltage levels with minimal transition energy losses and a quick voltage transient response. When scaling the voltage, the frequency must be scaled accordingly to meet signal propagation delay requirements. A power scheduler can intelligently compute the appropriate frequency and voltage levels needed to execute the various applications.

Figure 3.4: An Example of a DVFS Design

Instances Operating on Same Operating Voltage	Corresponding Power Domain	Libraries Used for all Models
dtmf_recvr_core, pm_inst	AO	ao_bc_0v99, ao_wc_0v99, ao_bc_0v792, ao_wc_0v792
TDSP_CORE_INST	TDSPCORE	tdsp_bc_0v792, tdsp_wc_0v792
PLLCLK_INST	PLL	ao_bc_0v99, ao_wc_0v99



The dtmf_recvr_core design shown in Figure 3-4, An Example of a DVFS Design, has several other blocks which for the sake of simplicity are not shown here. However, they all operate at the same voltage as the top-level of the design.

The voltage of the top-level design is scaled depending on the requested function of the design. If the processing speed is critical, a higher voltage is used; if the processing speed is not critical, the voltage is dynamically scaled down together with the clock frequency to save power. For this design, we assume that the voltage supply is dynamically controlled external to the chip. The input power signal for the top-level and the blocks that operate at the same voltage is `VDD`.

⚠ The design used here uses both the DVFS and PSO methodologies.

The `dtmf_recvr_core` design further contains:

- The `TDSP_CORE_INST` block

  This digital signal processing block operates at a lower voltage because its processing speed is not critical. When the block does not need to be operational, it is shut down.

  The power input for this block is `Vdd_TDSP_R`. The `clk_enable` signal disables the clock when the block is shut down.

- The `PLLCLK_INST` block

  This block is used to generate the clocks needed by all the blocks in the design. It has a reference clock, `ref_clk`, that is used to generate all other clocks.

  Because the design uses two operating voltages, two clock signals are created:

- The `low_clk` clock signal which feeds the `TDSP_CORE_INST` block has a constant lower frequency.
- The `var_clk` clock signal feeds the top-level design and other blocks and can vary in clock frequency depending on the operating voltage.

  The `freq_ctrl` signal ensures that the frequency of the `var_clk` signal used for the top-level design is scaled proportional to the voltage.

  Note: Because this block is an analog block, it needs to operate at a constant voltage to ensure correct functionality. Therefore a dedicated power input has been specified, `Avdd`.

- The `pm_inst` block

  This block generates all power control signals for the chip, and operates at the same voltage

as the top-level of the design.

In DVFS designs, a collection of logic blocks (hierarchical instances) and leaf instances that use the same main power supply and whose voltage and frequency can simultaneously change or be switched off belong to the same power domain.

The example design in Figure 3-4, An Example of a DVFS Design, has the following power domains:

- The `PLLCLK_INST` block is the only block in the design that operates at constant voltage 0.99V. This block belongs to power domain `PLL`.
- The `TDSP_CORE_INST` block operates at voltage 0.792V and it is the only block that is shut down at certain times. This block belongs to power domain `TDSPCORE`.
- The `pm_inst` block, the top-level design and the remaining blocks are always powered on but their operating voltage and frequency can change. They belong to power domain `AO`.

Power domains `PLL` and `AO` are never powered down. They are referred to as unswitched domains. Power domain `TDSPCORE` can be powered down and is called a switchable domain.

A steady state of a design in which some power domains are switched on and some power domains are switched off is called a power state. In a power state, each power domain operates on a specific voltage and an on or off state. Table 3-3, Power States, shows the operating voltages for each of the power domains in the three power states of the `dtmf_recvr_core` design. The voltages shown in this table correspond to the worst case voltages.

Table 3.3: Power States

Power Mode	Corresponding Power Domain		
	AO	**PLL**	**TDSPCore**
full	0.99	0.99	0.792
slow	0.99	0.99	0.0
sleep	0.792	0.99	0.0

To pass signals between portions of the design that operate on different voltages, level shifters are needed.

In order to prevent unknown values in the power domains that are powered down from propagation to the domains that remain powered on, isolation cells are needed. These isolation cells are typically placed at the boundaries of the power domains that are powered down or the input boundary of domains receiving a powered down signal. You can, however, also insert isolation

cells at the intermediate domain boundaries as required.

To help facilitate powered down blocks in resuming normal operation, state retention cells can be used for some sequential cells to keep their previous state prior to power down. Since these cells retain their state, they do not need to be reset or reconfigured after a power cycle. This allows the device to more quickly resume normal operation and, in some cases, continue where it left off in its process.

- For switchable domains you need to indicate how the power supply is connected and disconnected from the gates.
- For internal switchable domains, you must add power switch logic.
- For external switchable domains, the power switch logic is not part of the chip, so a control signal may not be available. In IEEE 1801, the `add_power_state` commands for the design will define states to indicate which domains can power off. In simulation, the testbench will drive the voltage of the switchable pins.

⚠ For this example, we are assuming that power domain `TDSPCore` is an internal switchable domain.

Special control signals are used to control the supply voltage, shut down a power domain, enable state retention, restore the state of the registers when powering up a power domain, and control the working of the power switch logic. Table 3-4, Signals Controlling the Power Domains, shows the signals used in this design example.

Table 3.4: Signals Controlling the Power Domains

Power Domain	Control Signals			
	voltage control	**power switch**	**isolation cell**	**state retention cell**
AO	VC	no control signal	no control signal	no control signal
PLL	no control signal	no control signal	no control signal	no control signal
TDSPCore	no control signal	ps_enable	iso_enable	pg_enable and pg_restore

When a domain is switchable, it derives its power from another power domain through either internal or external power switch logic.

In this example, power domain `TDSPCore` derives its power from power domain `AO`.

On the one hand, the majority of instances in a power domain are driven by the same power supply. For switchable domains, it is the primary power and ground nets of the (primary) power domain to which the instances belong that provide the power supply to the power and ground pins (follow-pins) of the cell.

On the other hand, isolation cells and state retention cells are driven by multiple power supplies. These special low power instances can have two sets of power and ground pins, and supply sets for each of these can be explicitly defined in the IEEE 1801 file. After using the `create_power_domain` command to specify a power domain, the `-isolation_supply_set` option defines the default additional supply for isolation cells and the `-retention_supply_set` option defines the additional supply for retention cells. These can be overridden by explicit supply sets defined using the `set_isolation` and `set_retention` commands.

3.3.1 Complete IEEE 1801 File for DVFS Example

```
upf_version 2.0

### Create supply ports and nets
create_supply_port VDD
create_supply_port VDDL
create_supply_port avdd
create_supply_port VSS

create_supply_net VDD_TDSP
create_supply_net VDD
create_supply_net VDDL
create_supply_net avdd
create_supply_net VSS

### Create supply sets
create_supply_set ss_AO    -function {power VDD}      -function {ground VSS}
create_supply_set ss_TDSP  -function {power VDD_TDSP} -function {ground VSS}
create_supply_set ss_PLL   -function {power avdd}     -function {ground VSS}
create_supply_set ss_VDDL  -function {power VDDL}     -function {ground VSS}

### Create power domains
create_power_domain AO -include_scope -supply {primary ss_AO}
create_power_domain TDSPCore  -supply {primary ss_TDSP}  \
    -elements {TDSP_CORE_INST0}

create_power_domain PLL  -supply {primary ss_PLL}  \
    -elements {PLLCLK_INST}
```

```
### Create isolation and level shifter strategies
set_isolation ISORULE1 \
    -domain TDSPCore \
    -isolation_supply_set AO.primary \
    -applies_to outputs \
    -isolation_signal {PM_INST/isolation_enable} \
    -isolation_sense low \
    -clamp_value 1 \
    -location parent

set_level_shifter LSRULE_L2H_CORE\
   -domain TDSPCore \
   -applies_to outputs \
   -input_supply_set ss_VDDL\
   -output_supply_set AO.primary \
   -location parent

set_level_shifter LSRULE_H2L \
    -domain TDSPCore \
    -applies_to inputs \
    -location parent \
    -input_supply_set AO.primary \
    -output_supply_set TDSPCore.primary

set_level_shifter LSRULE_H2L_PLL \
    -domain PLL \
    -applies_to outputs \
    -location parent \
    -input_supply_set PLL.primary \
    -output_supply_set AO.primary

set_level_shifter LSRULE_L2H_PLL \
    -domain PLL \
    -applies_to inputs \
    -location parent \
    -input_supply_set AO.primary \
    -output_supply_set PLL.primary

### Create retention strategies
set_retention SRPG_TDSP \
    -domain TDSPCore \
    -restore_signal {PM_INST/state_retention_restore low} \
    -save_signal {PM_INST/state_retention_save high} \
    -retention_supply_set ss_VDDL
```

```
### Create domain level states
add_power_state PLL.primary \
    -state HIGH {-supply_expr {power == `{FULL_ON, 0.99} && ground == \
    `{FULL_ON, 0.0}} -simstate NORMAL}

add_power_state ss_VDDL \
    -state LOW {-supply_expr {power == `{FULL_ON, 0.792} && ground == \
    `{FULL_ON, 0.0}} -simstate NORMAL}

add_power_state AO.primary \
    -state HIGH {-supply_expr {power == `{FULL_ON, 0.99} && ground == \
    `{FULL_ON, 0.0}} -simstate NORMAL} \
    -state LOW {-supply_expr {power == `{FULL_ON, 0.792} && ground == \
    `{FULL_ON, 0.0}} -simstate NORMAL}

add_power_state TDSPCore.primary \
    -state LOW {-supply_expr {power == `{FULL_ON, 0.792} && ground == \
    `{FULL_ON, 0.0}} -simstate NORMAL} \
    -state OFF {-supply_expr {power == `{OFF} && ground == \
    `{FULL_ON, 0.0}} -simstate CORRUPT}

### Create system-level modes
add_power_state AO -state FULL {-logic_expr {AO.primary == HIGH && \
                TDSPCore.primary == LOW && PLL.primary == HIGH && ss_VDDL == LOW}}

add_power_state AO -state SLOW {-logic_expr {AO.primary == LOW && \
                TDSPCore.primary == LOW && PLL.primary == HIGH && ss_VDDL == LOW}} -
update

add_power_state AO -state SLEEP {-logic_expr {AO.primary == LOW && \
                TDSPCore.primary == OFF && PLL.primary == HIGH && ss_VDDL == LOW}} -
update

### Create power switch, level shifter, isolation cell
create_power_switch TDSP_sw -domain TDSPCore \
    -output_supply_port {vddo TDSPCore.primary.power} \
    -input_supply_port {vddi VDDL} \
    -control_port {EN PM_INST/power_switch_enable} \
    -on_state {full_on vddi EN} \
    -off_state {off !EN}
```

3.3.2 Steps to Create the IEEE 1801 File for DVFS Design

This section describes the information to include in an IEEE 1801 file for a design using the DVFS methodology. The example shown in Figure 3-4, An Example of a DVFS Design, is used throughout this section.

The following information is needed for both design creation and logic verification:

- Specifying the Power Domains
- Specifying the Power and Ground Nets and Ports
- Specifying the Supply Set
- Associate the Supply Set to the Power Domains
- Specifying the Voltage States Allowed for Each Domain
- Defining System Level Power States
- Specifying the Rules to Create Level Shifter Logic
- Specifying Rules to Create Isolation Logic
- Specifying the Rules to Create State Retention Logic
- Specifying the Power Switch Logic
- Specifying Low Power Library Cell Attributes

3.3.2.1 Specifying the Power Domains

To identify portions of the design that operate on the same voltage and that can be simultaneously switched on or off, use the `create_power_domain` command with the following syntax to specify a power domain and its associated instances:

```
create_power_domain power_domain
    [-elements instance_list]
    [-exclude_elements instance_list]
    [-include_scope]
    [-supply {supply_set_handle [supply_set_ref]}]
```

For Figure 3-4, An Example of a DVFS Design:

```
create_power_domain AO -include_scope
create_power_domain TDSPCore -elements {TDSP_CORE_INST}
create_power_domain PLL -elements {PLLCLK_INST}
```

3.3.2.2 Specifying the Power and Ground Nets and Ports

For IEEE 1801, you can define the supply ports and nets (power and ground) used in a design. The supply ports are external ports that provide a voltage supply to the device. The supply nets are the internal nets that will be used to connect domains and power pins of individual components in the design. Once specified, these power or ground supplies can be associated to power domains, and as connections to special cells.

Define the power and ground supplies using the following syntax:

```
create_supply_port supply_port_name

create_supply_net supply_net_name
```

Then, connect the supply port to the supply net as shown:

```
connect_supply_net supply_net_name -port supply_port_name
```

For a complete description of these commands, see the IEEE 1801-2013 specification.

3.3.2.3 Specifying the Supply Set

For IEEE 1801, a basic building block of the low power intent is the supply set. Because a supply set is a bundle of supply nets, you can use the supply set to simplify the connection of supply nets to power domains and special cells. That is, rather than specifying each supply net individually, you can use a supply set instead.

To define a supply set for each power domain, use the following syntax:

```
create_supply_set ss_name [-function {supply_function supply_net}]*
```

The supply set defines a set of functions like primary power and primary ground, and then defines which supply nets provide those functions. For example, to specify that the primary power for `ss_AON` is provided by the supply net `VDD`, and the ground function is provided by `VSS` use the following `create_supply_set` command:

```
create_supply_set ss_AO -function {power VDD} -function {ground VSS}
create_supply_set ss_TDSP -function {power VDD_TDSP_CORE} -function {ground VSS}
create_supply_set ss_PLL -function {power Avdd} -function {ground Avss}
```

3.3.2.4 Associate the Supply Set to the Power Domains

To associate supply sets to power domains, use the `associate_supply_set` command with the following syntax:

```
associate_supply_set ss_name -handle power_domain.handle
```

For example:

```
asscoiate_supply_set ss_AO -handle AO.primary
```

The associate supply set defines that all primary power and ground pins for any cell in `AO` will be connected to the nets defined in the `ss_PD1` supply set. In this case, it means that all primary power pins in the `PD1` domain will be connected to the `supply_net VDD`.

3.3.2.5 Specifying the Voltage States Allowed for Each Domain

For IEEE 1801, Cadence recommends using an `add_power_state` command for each power domain. Use this command to define the allowed operating voltage for each specifed domain. Although it is not required, it does provide a very clear definition of power intent for more complex designs.

To specify the operating voltages used in the design, use the `add_power_state` command. For example:

```
add_power_state AO.primary \
    -state HIGH { -supply_expr {power == \
    `{FULL_ON, 0.99} && ground == `{FULL_ON, 0.0}} -simstate NORMAL} \
    -state LOW { -supply_expr {power == \
    `{FULL_ON, 0.792} && ground == `{FULL_ON, 0.0}} -simstate NORMAL}

add_power_state TDSPCore.primary \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 0.792} && ground == `{FULL_ON, 0.0} -simstate NORMAL} \
    -state OFF { -supply_expr {power == \
    `{OFF} && ground == `{FULL_ON, 0.0}} -simstate CORRUPT}

add_power_state PLL.primary \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 0.99} && ground == `{FULL_ON, 0.0}} -simstate NORMAL}

add_power_state ss_TDSPCore_ref \
    -state ON { -supply_expr {power == \
    `{FULL_ON, 0.99} && ground == `{FULL_ON, 0.0}} -simstate NORMAL}
```

The `add_power_state` for domain `AON` has two defined states, `HIGH` and `LOW` that reflect the high and low voltage cases for DVFS. For `TSDPCore`, the `ON` and OFF states reflect the power shutoff capabilities of the domain.

If your designs have a `pmos` or `nmos` bias, you can also specify these using the `-supply_expr` option.

3.3.2.6 Defining System Level Power States

The `add_power_state` command can also define system level power states. A system level power state defines the interactions between domains rather than the allowed states per domain.

For IEEE 1801, power states are primarily used for verification. For simulation, power states are used for coverage and creating assertions. For low power design checks, power states can be used to verify that all required level shifting and isolation is specified in the power intent.

The `VC` pin is a voltage control pin that is used to define if the `AO` domain is in the high or low voltage state. You can use this signal in an `add_power_state` command to differentiate between these modes. For example:

```
add_power_state PD1 \
    -state full {-logic_expr { VC== 1 && AO.primary == HIGH \
    && ss_TDSPCore_ref == ON && TDSPCore.primary == ON && PLL.primary == ON }}

add_power_state PD1 \
    -state slow {-logic_expr { VC == 0 && AO.primary == LOW \
    && ss_TDSPCore_ref == ON && TDSPCore.primary == ON && PLL.primary == ON }}

add_power_state PD1 \
    -state sleep {-logic_expr {VC == 0 && AO.primary == LOW \
    && ss_TDSPCore_ref == ON && TDSPCore.primary == ON && PLL.primary == OFF }}
```

To associate the nominal conditions with power domains, use the `create_power_mode` command.

The add_power_state commands above refer to the power domain states only. This is the recommended style for system level power states. If this were an IEEE 1801 hierarchical flow, the `-logic_expr` option could also refer to other lower level system states. In this way, the top-level states can utilize a higher level of abstraction and do not need to know all the low level details of each domain or subsystem.

3.3.2.7 Specifying the Rules to Create Level Shifter Logic

Depending on your technology, you may need level shifters when passing any signals...

- From a power domain with a lower voltage to a power domain with a higher voltage.
- From a power domain with a higher voltage to a power domain with a lower voltage.

  In both cases, to create a rule to use between power domains or a set of pins, the `set_level_shifter` command is required, the most common options and values are:

```
set_level_shifter strategy_name
    -domain domain_name \
    -applies_to <inputs|outputs|both> \
    -location <parent|self|other> \
    -input_supply_set supply_set_name \
    -output_supply_set supply_set_name \
    -threshold <value|list>
```

The level shifter strategy typically applies to the inputs or outputs of a specific domain as specified by the `-applies_to` option. The `-location` option provides guidance as to where the implementation tools should place the isolation cell: either on the inside of the domain specified (`self`) or outside of the hierarchy (`other`). The `-threshold` option defines when to insert level shifting. You can also use it to avoid level shifting altogether when the voltage difference is small.

For Figure 3-4, An Example of a DVFS Design:

```
set_level_shifter LS_AON_L2H -domain AO -applies_to inputs \
    -location parent \
    -input_supply_set TDSPCore.primary -output_supply_set AO.primary

set_level_shifter _outputs LS_AON_H2L -domain AO -applies_to outputs \
    -location self \
    -input_supply_set AO.primary -output_supply_set ss_TDSPCore.primary

set_level_shifter _outputs LS_PLL_H2L -domain PLL -applies_to outputs \
    -location self \
    -input_supply_set PLL.primary -output_supply_set TDSPCore.primary
```

3.3.2.8 Specifying Rules to Create Isolation Logic

Isolation is used to prevent unknown states from propagating from a domain that is powered down to powered on logic in the design. In IEEE 1801, the power intent for inserting isolation cells is called an **isolation strategy**. An isolation strategy typically defines the value of a pin when the driver of that pin powers off. To define when isolation cells must be added or to specify which pins must be isolated, use the `set_isolation` command. The syntax below lists common options and values:

```
set_isolation strategy_name
    -domain domain_name
    -isolation_signal expression
    [-applies_to {inputs|outputs|both}]
    {-elements pin_list | \
     -from power_domain_list | \
     -to power_domain_list}...
    [-clamp_value { 0 | 1 | latch | Z}]
    [-isolation_supply_set supply_set_list]
    [-location {self other parent fanout}]
    [-diff_supply_only]
    [-source <source_domain_name | source_supply_ref>]
    [-sink <sink_domain_name | sink_supply_ref>]
```

The isolation strategy is always related to a power domain. By default, it applies to all the inputs and outputs of the domain specified in the strategy, but you can filter it based on a number of parameters.

A specific list of pins or instances:

- `-diff_supply_only` — Only isolate if the driver and receiver of the pin are from different power supplies. This option prevents unnecessary isolation insertion.
- `-source`/`-sink` — Specify constraints based on the driver or receiver of the pin. These options give more flexibility and control to the isolation, but are uncommon in most designs.

The functionality of the isolation is defined by the `-clamp_value` and `-isolation_signal` options.

- `-clamp_value` defines the logic value that the pin should take when isolated. The most common arguments are `0`, `1`, and `latch`.
- `-isolation_signal` defines the control for the isolation When the isolation signal is active the specified value will be driven on the pin.

`-location` specifies where in the hierarchy the isolation should be inserted. Typically, this will be either inside of the specified domain (`self`) or on the pin just outside the specified domain (`other`).

See Figure 3-3, Using the Location Option, for an illustrated explanation.

`-isolation_supply_set` defines the power supply that will be connected to the secondary power pins, if any, for the inserted isolation cell.

Typically, isolation logic is needed to isolate signals going from a power domain being switched down to a power domain that remains on. If an input of a powered down domain requires a stable signal for electrical reasons, isolation is required even if the signal goes from a powered on domain to a powered down domain.

In Figure 3-4, An Example of a DVFS Design, isolation logic will be needed in power mode `sleep` for any nets going from power domain `TDSPcore`. For example:

```
set_isolation ISORULE -domain TDSPCore -applies_to outputs \
    -isolation_supply_set ss_TDSPCore_ref \
    -location parent \
    -isolation_signal {pm_inst/ice_enable[0]} \
    -clamp_value 0
```

3.3.2.9 Specifying the Rules to Create State Retention Logic

State retention is a design methodology that allows a set of registers to be preserved through a power on or power off sequence. There are two common applications for state retention:

- Preserve Configuration Registers — Preserve a specific set of configuration registers through the power sequence. This allows the domain to start up with the same configuration and speeds up the power on cycle. Without retention, the device would need to be reconfigured and in many cases this adds overhead to the system software.
- Full restore of the domain — Allow the device to start up in the exact state that it was powered off. That is, any operations in the domain essentially pause during power shutoff and then resume immediately on power up.

  In many cases, you can save ALL registers in the design. This ensures that a full *known* state of the device is restored. However, IEEE 1801 does not require that all registers are maintained, and you may attempt to optimize a partial set of registers. This partial set provides a smaller area for the design but runs some risk of missing a register and having incomplete or incorrect restoration.

To define a rule for replacing selected registers or all registers in the specified power domain with state retention registers, use the `set_retention` command. Common options are:

```
set_retention strategy_name
```

```
    -domain power_domain
    [-elements element_list]
    [-exclude_elements element_list]
    [-save_signal {logic_net <high | low | posedge | negedge>}
    -restore_signal {logic_net <high | low | posedge | negedge>}]
    [-restore_condition expr]
    [-save_condition expr]
    [-retention_condition expr]
    [-retention_supply_set supply_set_ref]
```

The retention strategy is defined per domain and specifies which elements should be retained and the control over that retention. The `-save_signal` and `-restore_signal` options define the control signal that triggers the save and restore operations respectively. The `-save_condition`, `-restore_condition`, and `-retention_condition` options are used by simulation and static checking as additional checks on the operation of the retention. For instance, to ensure the clock is held low at the time of the save, specify `-save_condition {!clk}`.

The retention cells have a separate retention supply that is kept active when the domain is off. This retention supply is specified with the `-retention_supply_set` option.

For Figure 3-4, An Example of a DVFS Design, use the following command:

```
set_retention SRPG_TDSP -domain TDSPCore \
    -save_signal {PM_INST/pg_restore[0] high} \
    -restore_signal {PM_INST/pg_enable[0] low} \
    -retention_supply_set SS_TDSPCore_Ref
```

3.3.2.10 Specifying the Power Switch Logic

In a DVFS design, one must define the power switch that controls the shutoff behavior of the domain. This includes the input and output power nets, the control ports and each allowed state for the power switch.

To specify the power switch, use the `create_power_switch` command.

```
create_power_switch
    -domain power_domain
    -output_supply_port {port supply_net_name}
    {-input_supply_port {port supply_net_name}}*
    {-control_port {port_name [net_name]}*
    {-on_state {state_name input_supply_port {boolean_expression}}}*
    [{-off_state {state_name {boolean_expression}}]*
    [-ack_port {port_name net_name [logic_value]}]*
    [-ack_delay {port_name delay}]*
```

The typical power switch will have a single control signal that defines when the input supply will be connected to the output supply. The `-on_state` and `-off_state` options are used to define that control logic. Advanced syntax allows for the definition of much more complicated switches, such as multiple input voltages and multiple control signals.

The `-ack_port` option defines an acknowledge signal that generates a value when the output power has been fully restored. You might use this as an input to other power switches or as an input to an FSM waiting for the domain to be fully restored before proceeding to the next state.

For Figure 3-4, An Example of a DVFS Design, use the following command:

```
create_power_switch PD2_sw -domain PD2 \
    -output_supply_port {vddo TDSPCore.primary.power} \
    -input_supply_port {vddi ss_TDSPCore_ref.power} \
    -control_port {EN PM_INST/ps_enable } \
    -on_state {full_on vddi {!EN}} \
    -off_state {full_off {EN}}
```

3.3.2.11 Specifying Low Power Library Cell Attributes

The 1801-2013 specification allows for the definition of library attributes in two different manners:

- Native 1801 commands
- Attributes in the Liberty definition of the technology

The IEEE-2013 recommendation is to use native 1801 commands because that format is entirely within the 1801 specification. While the Liberty format is the defacto standard for library information, it is not a format controlled by the IEEE Standards Association or any other open standards body.

For DVFS designs with PSO, the common low power cells are:

```
    define_always_on_cell

    define_level_shifter_cell

    define_isolation_cell

    define_retention_cell
```

Example of 1801 low power cells:

```
define_always_on_cell -cells LP_AON -power VDD -ground VSS

define_level_shifter_cell -cells LS_HTL \
     -input_voltage_range {{1.0 1.1}} -output_volatage range {{0.8 0.9}} \
     -input_power_pin VDD -ground VSS -output_power_pin VDDL \
     -direction high_to_low -valid_location source
```

```
define_level_shifter_cell -cells LS_LTH \
     -input_voltage_range {{0.8 0.9}} -output_voltage_range {{1.0 1.1}} \
     -input_power_pin VDD -ground VSS -output_power_pin VDDH \
     -direction low_to_high -valid_location source

define_isolation_cell -cells ISO_ON \
     -power VDD -ground VSS \
     -enable EN \
     -valid_location on

define_isolation_cell -cells ISO_OFF \
     -power VDD -ground VSS \
     -power_switchable VDD_SW \
     -enable EN \
     -valid_location off

define_retention_cell -cells RET1 \
     -power VDD -ground VSS \
     -power_switchable VDD_SW \
     -clock_pin CK \
     -always_on_pin Q \
     -save_function {SV posedge}
```

4

Coding Guidelines for IEEE 1801 Files

This chapter describes how to develop IEEE 1801 code that is Cadence-compatible.

4.1 Background

By following these guidelines, you can develop IEEE 1801 code that is easily handled by Cadence tools, with the extra benefit of making the power intent more portable. Do note that this document assumes that you are familiar with the basic syntax and semantics of IEEE 1801; it is not a tutorial or guide for the IEEE 1801 language.

There are three industrial documents that are related to the IEEE 1801 standard. For convenience, throughout this chapter they will be referred to as LRM1, LRM2, and LRM2.1 (LRM stands for Language Reference Manual).

- LRM1 (*Accellera Unified Power Format Standard*, Version 1.0, 2007, also known simply as UPF 1.0): This language is not part of the IEEE 1801 standard, but is the basis of the IEEE

1801 standard.

- LRM2 (*IEEE 1801-2009 Standard for Design and Verification of Low Power Integrated Circuits*, also known as UPF 2.0): This is the initial low power standard developed and implemented by the IEEE Standards Association in 2009.
- LRM2.1 (*IEEE 1801-2013 Standard for Design and Verification of Low Power Integrated Circuits*, also known as UPF 2.1): This is the 2013 update to the IEEE standard for low power.

In general, the commands and options evolve from LRM1 to LRM2 and to LRM2.1. You can specify which version you want to use by issuing the *upf_version* command, with values 1.0, 2.0, and 2.1 respectively corresponding to LRM1, LRM2, and LRM2.1.

Strictly speaking, despite the evolution, history, and similarity of each standard, the three LRMs define three *different* languages — backward compatibility has not been maintained and certain commands and options have been deprecated. In theory, Cadence tools are built upon the 1801-2013 standard; however, this is not always the case. If the 1801-2013 standard is supported, you should also be able to use the majority of its commands and options successfully with any of the previous languages. In summary, Cadence's strategy can be described both syntactically and semantically:

- **Syntactically**: The majority of Cadence tools support all three languages. Note that this statement is true in concept only. In practice, there are some commands and options in all the languages that are not supported, and even overall LRM support may be limited with certain tools. Use the *upf_version* command to verify the version and syntax when authoring a design. Otherwise, consult the product manuals for those tools supporting IEEE 1801 to get more information.
- **Semantically**: If a Cadence tool follows LRM2.1, it might ignore the *upf_version* command given certain rules.

  If a command available in LRM1 and LRM2 has the same semantics as that in LRM2.1, the command should work as expected. This scenario represents most cases.

  If a command available in LRM1 and LRM2 has different semantics than that in LRM2.1, you should understand the difference and either control, avoid, or recognize whatever has changed.

1. Control: When the difference is an important one to keep, Cadence provides a proprietary attribute that you can directly control.

   Example (see the "Definition of Domain Interface" Section for details):

   The definition of *domain interface* is different in all three languages. To control which definition to use, you may issue the following command:

```
set_design_attributes -elements . -attribute {domain_interface_def xyz}
```

For this attribute, *xyz* can be either "1.0", "2.0", or "2.1".

2. Avoid: When there is a way to mask the difference and hence completely avoid the difference, you should always do it.

   Example (see "When absent" in the "Issues with -applies_to" Section for details):

   The `-applies_to` option in the `set_level_shifter` command is not mandatory. When it is absent, the default value is *both* in LRM1, *outputs* in LRM2, and *none* in LRM2.1. Cadence recommends to always have it specified in all `set_level_shifter` commands, thereby always avoiding and masking this difference out.

3. Recognize: When certain semantics are simply not reasonable, you should recognize them and ignore them.

   Example (see "When used with -source/-sink" in the "Issues with -applies_to" Section for details):

   In LRM2, the following limitation exists for `set_isolation`:

   "It shall be an error if `-applies_to` is specified along with `-source` and/or `-sink`."

   This limitation is without merit, and is removed in LRM2.1. Cadence tools ignore this and simply follow LRM2.1.

   For the rest of this document, each section discusses an aspect of the semantic difference among the three languages. At the beginning of each section, succinct short guidelines are given as a quick reference.

Product command and option support

Please refer to "IEEE 1801 Command Support Table" Section of this document for individual product command and option support. BNF commands show all LRM options.

4.2 Definition of Domain Interface

Guidelines: (If needed) Use `set_design_attributes -elements -attribute {domain_interface_def <1.0/2.0/2.1>}` to control the definition of a domain interface.

The semantics of many commands in IEEE 1801 are built upon the concept of the domain interface. The definition of a domain interface differs in all the three LRMs. For details, consult the LRMs. As a short summary:

1. In LRM1: There is no explicit definition. However, the common practice in the industry is to only include the "lowConn" side of a boundary port.
2. In LRM2 (p.89): Include both "lowConn" and "highConn" sides of a boundary port.
3. In LRM2.1 (p.6): Include both "lowConn" and "highConn" sides of a boundary port, plus the ports on a macro instance that have a different associated supply set than the connected supply set.

As an example, the following command:

```
set_isolation iso1 -domain PD1 -applies_to outputs ...
```

Would potentially select three different sets of target ports under the three interpretations in the LRMs. In order to set a desirable choice, issue the `set_design_attributes` command below:

```
set_design_attributes -elements . -attribute {domain_interface_def xyz}
```

Where `xyz` can be:

1. String "1.0", which corresponds to the LRM1 bullet above.
2. String "2.0", which corresponds to the LRM2 bullet above.
3. String "2.1", which corresponds to the LRM2.1 bullet above (This value may not be supported by all the Cadence tool yet).

The `set_design_attributes` command associates a Cadence propriety attribute, *`domain_interface_def`*, to the current scope. The command can appear anywhere in the 1801 files, and the setting is effective throughout the entire run. In other words, this command is not order dependent and can be issued in any scope. If multiple settings on this attribute exist, the attribute's value must all be the same.

If you do not set this attribute, the default value depends on the `upf_version` command specified in the input 1801 files:

- If the lowest `upf_version` value throughout all the 1801 input files is "1.0", this attribute's value is "UPF"

- If the lowest `upf_version` value throughout all the 1801 input files is "2.0", this attribute's value is "1801-2009".
- If the lowest `upf_version` value throughout all the files is "2.1", this is not currently supported, but when the support is ready, this attribute's value will be "1801-2013".
- If the input 1801 file does not contain the `upf_version` command, the attribute defaults to "1801-2009"

4.3 Issues with -applies_to

Guidelines:

1. Always explicitly specify `-applies_to` in `set_isolation` and `set_level_shifter`.
2. Always view `-applies_to` as a filter, with no exceptions.

Commands `set_isolation` and `set_level_shifter` both have the option `-applies_to`, which serves as a filter on the candidate ports. The next three subsections discuss issues with `-applies_to`.

4.3.1 When absent

When the option `-applies_to` is absent in a `set_isolation` or `set_level_shifter` command, the LRMs have different default values:

1. LRM1: "*outputs*" for `set_isolation`; "*both*" for `set_level_shifter`
2. LRM2: "*outputs*" for `set_isolation` and `set_level_shifter`
3. LRM2.1: no defaults for `set_isolation` and `set_level_shifter`

As an example, in the following command:

```
set_level_shifter ls1 -domain PD1
```

The *`-applies_to`* option is absent. Under the three different defaults in the three LRMs, this command assumes different `-applies_to` values. That means this same command will have different behaviors under different interpretations. In order to make your 1801 code more portable and to remove potential mismatched interpretations, Cadence recommends that you always explicitly specify the `-applies_to` option.

4.3.2 When used with -source/-sink

In LRM2 (p.93), command `set_isolation` has the following limitation:

"It shall be an error if `-applies_to` is specified along with `-source` and/or `-sink`"

This limitation is without merit, as both the `-source and -sink` options are filters that work on a different aspect than `-applies_to`. Cadence tools ignore this limitation.

4.3.3 When used with -elements that has a port listed

In LRM1 (p.49), `set_isolation` has the following limitation between `-elements` and `-applies_to`:

"*If `-elements` directly specifies a port by name (not implicitly, by specifying the port's instance or an ancestor of that instance), then the isolation strategy shall apply to that port regardless of whether that port's mode matches the one specified by the `-applies_to` option.*"

This basically means that if `-elements` contains a port name, then `-applies_to` has no effect on that port. In LRM2, there is different terminology but similar semantics. In LRM2.1 there is no such limitation.

Cadence tools follows LRM2.1's semantics. That is, `-applies_to` is always honored, regardless of how a candidate port is specified in `-elements`. This is a better because `-applies_to` can be a true filter with no exceptions (Although one potentially unexpected exception is when the `-elements` list returns the value of a `find_objects` command).

4.4 PST versus add_power_state

Guidelines: Use the PST flow to specify legal modes.

When specifying the legal power modes of a design, there are two flows: the PST flow and the `add_power_state` flow.

1. LRM1 describes the PST (power state table) flow, which basically consists of the following commands: `add_port_state`, `create_pst`, and `add_pst_state`.
2. LRM2 introduces the `add_power_state` flow, which should be able to replace the PST flow, although the PST flow is still part of LRM2.
3. LRM2.1 revises the syntax for the `add_power_state` command, and also further marks the PSTs as deprecated.

At the time of this document, not all Cadence tools support the `add_power_state` command . Therefore, Cadence currently recommends using the PSTs. However, in the long run, Cadence will recommend the use of the `add_power_state` command, with the syntax in LRM2.1. After all, using `add_power_state` properly elevates the user's abstraction level, especially alleviating an RTL designer from having to deal with physical nets (as is the case with PSTs).

> Xcelium
>
> There is no SimVision or Tcl support for the `add_power_state` command.

4.5 set_port_attributes

Guidelines: For the `set_port_attributes` command,

1. Use only the supported options listed below.
2. Avoid the non-standard command `set_related_supply_net`, unless you are using UPF 1.0.

The `set_port_attributes` command has a large number of options. At the time of this document, Cadence recommends that you use only the following options:

- `-ports` (and `-exclude_ports` if needed)
- `-elements`
- `-applies_to` (the LRM2.1 syntax)
- `-driver_supply` or `-receiver_supply`

With these options, this command can be used to associate the driver or receiver supply sets of top-level ports, which is one of the most important applications of this command. For example:

```
set_port_attributes -ports {in1 in2} -driver_supply ss1
```

Use this command to set the external driver supply for top-level ports `in1` and `in2` to be `ss1`. Cadence will support other options for future releases, but only with the syntax described in LRM2.1. This is because:

1. LRM1 does not have the command.
2. The syntax described in LRM2 is somewhat cumbersome and confusing, especially the different usage of `-applies_to`. Cadence does not plan to support these, not even syntactically.

Also note that, as this command does not exist in LRM1, if a user is authoring a UPF 1.0 file, there is no official way to associate the supplies for top-level ports. To patch this deficiency, an industry practice using a non-standard command `set_related_supply_net` is often used. Cadence discourages the use of such non-standard commands. However, to support legacy IP written in UPF 1.0, Cadence tools do support this command.

4.6 Anonymous Supply for Top-level Ports

Guidelines: Always associate supplies for top-level ports.

The previous section (on `set_port_attributes`) describes how a supply set can be associated with a top-level port. What happens if a user simply does not specify such an association on a top-level port?

1. In LRM1 and LRM2, this specific situation is not addressed so the semantics are not clear.
2. In LRM2.1, an anonymous supply that is not equivalent to any user-defined supply is implicitly associated with the top-level port.

Cadence recommends to always explicitly associate a supply set with each top-level port (see the previous section for details). Without such an association, the power intent is usually incomplete, and the software is likely to generate an error. Other tools in the industry assume all top-level ports are implicitly associated with the primary supply of the top module's power domain. This behavior has no LRM basis and is error prone. Cadence does not recommend this behavior; however, with that said and to increase compatibility for legacy IPs, Cadence provides a proprietary attribute that you can use to control this behavior. You can set this attribute with the following command:

```
set_design_attributes -elements . -attribute {top_ports_have_anon_supply <0/1>}
```

The Cadence proprietary attribute, `top_ports_have_anon_supply`, takes a value of either 0 or 1 where 1 is usually the default. For designs that use multiple 1801 files, this attribute's value depends on the lowest `upf_version` specified. For example:

- If the lowest `upf_version` is 1.0 or 2.0 across multiple 1801 files, the value is 0.
- If the lowest `upf_version` is 2.1 across multiple 1801 files, the value is 1.

The user can always issue an explicit command to override the default value; however, do note that the guideline is to always associate top-level ports' to supplies. Not associating or relying on the proprietary attribute will create 1801 code that is not portable.

4.7 Precedence Rules for set_isolation, set_level_shifter, and set_repeater

Guidelines:

1. Avoid relying on `-source` and `-sink` having higher precedence, as stated in LRM2
2. Avoid relying on bit-blasted ports having higher precedence, as stated in LRM2.1

In commands `set_isolation`, `set_level_shifter`, or `set_repeater`, a target port may be applicable to more than one such command. It is then up to the precedence rules to determine which strategy this target port should take action on. The LRMs' precedence rules have:

1. In LRM1, no wording.
2. In LRM2 (p.21):
    a. *Direct UPF commands. The power intent is applied through an explicit UPF command reference to a design object.*
    b. *Power intent applied to a parent is inherited by each child and transitively applied to descendents, except when a direct UPF command applies.*
    c. *Strategies specified with both the* **`-source`** *and* **`-sink`** *options.*
    d. *Strategies specified with a* **`-source`** *or* **`-sink`***.*
3. In LRM2.1 (p.44):
    a. *Command that applies to part of a multi-bit port specified explicitly by name*
    b. *Command that applies to a whole port specified explicitly by name*
    c. *Command that applies to all ports of an instance specified explicitly by name*
    d. *Command that applies to all ports of a specified power domain with a given direction*
    e. *Command that applies to all ports of a specified power domain*

Some observations after comparing the two different sets of precedence rules include:

Observation A: The general philosophy of "more specific rules wins" is the same (2a, 2b versus 3c, 3d, and 3e).

Observation B: LRM2 gives `-source` and `-sink` higher precedence (2c and 2d)

Observation C: LRM2.1 gives bit-blasted port higher precedence (3a and 3b)

For maximum portability in terms of precedence rules, Cadence recommends that the user only rely on the common philosophy shared between the two LRMs (i.e., Observation A above). Conversely speaking, a user should:

- Avoid relying on `-source` and `-sink`being higher precedence (Observation B above)
    - Avoid this simply because this is not in LRM2.1.
- Avoid relying on bit-blasted port being higher precedence (Observation C above)
    - Avoid this not only because this is not in LRM2, but also because find_objects may or may not return bit-blasted port names.

Note that precedence rules are there only to increase convenience, as there are enough constructs in these commands that can theoretically make all strategies have disjointed target ports, thereby completely avoiding the application of these rules. With a tradeoff on convenience, you can increase portability.

For example, say port *p* is applicable to both strategies below:

```
set_isolation iso1 -domain PD1 -sink SS2 ...
set_isolation iso2 -domain PD1 ...
```

Under LRM2's precedence rules, strategy `iso1` has higher precedence (because `-sink` is present). However under LRM2.1's precedence rules, both strategies have the same precedence. To increase portability, you could either add `-elements` to `iso1`:

```
set_isolation iso1 -domain PD1 -sink SS2 -elements p ...
```

Or add another `-sink` to `iso2`:

```
set_isolation iso2 -domain PD1 -sink SS3 ...
```

As another example, say bit 2 of bus q is applicable to both strategies below:

```
set_isolation iso3 -domain PD1 -elements {q[2]} ...
set_isolation iso4 -domain PD1 -elements {q} ...
```

Under LRM2's precedence rules, these two strategies have the same precedence. However under LRM2.1's precedence rules, *iso3* has higher precedence (because bit 2 is a specific bit of bus *q*). To increase portability, you could change `iso4` to either:

```
set_isolation iso4 -domain PD1 -elements {q[0] q[1] q[3] })
```

Or:

```
set_isolation iso4 -domain PD1 -elements {q} -exclude_elements {q[2]}
```

Either way would make `iso3` and `iso4` have disjointed target ports, thereby avoiding the ambiguous

interpretations from different LRMs.

4.8 Avoid deprecated/legacy command/option if possible

Guidelines: Avoid deprecated or legacy commands and options, if possible.

LRM2.1 marks many commands and options as either deprecated or legacy. While it is not practically possible to completely avoid using them at the current time, Cadence recommends avoiding them when you can.

An example of one option that has deprecated values that should be avoided is `-location` when used with `set_isolation` and `set_level_shifter`:

- In LRM2, the possible values of `-location` are `self`, `other`, `parent`, `automatic`, `fanout`, `fanin`, `faninout`, and `sibling`.
- In LRM2.1, the last 3 values above are marked for deprecation. Users should simply avoid using these deprecated values.

An example of commands that still cannot be avoided at the current time are those used to define PSTs (power state tables). See "PST versus add_power_state" Section for more details.

> Xcelium
>
> For legacy reasons, IES supports many of the above UPF 1.0 commands and options. Please refer to the Xcelium Low-Power Simulation IEEE 1801 documentation for details.

4.9 Miscellaneous Guidelines

1. Always specify `-location` in `set_isolation` and `set_level_shifter`. Avoid using the default, which are different in the LRMs.