Addressing the Challenges of Generically Specifying Power Intent with Multi-Rail Macros

Brandon S. Skaggs
Principal CAD Engineer
Cypress Semiconductor
201 E. Main Street, Lexington, KY 40507

***Abstract* -- The Unified Power Format (UPF) language is well suited for describing power distribution networks for standard cells such that connections can be made automatically; however, cell macros with multiple power rails often require explicit, manual connections that are often technology-dependent--reducing the portability of the design. Methods for abstracting these connections are discussed, including recommended enhancements to the *Liberty* Library specification and IEEE 1801 UPF standards to better support such cells.**

I. Introduction

The IEEE1801 Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems—commonly known as the Unified Power Format (UPF) standard—provides a powerful mechanism for providing intended power supply connectivity and behavior for soft IP—which is often described in abstract hardware description language (HDL) without power intent. By providing this power intent via UPF; power domains, their associated supplies, and expected operational states can be defined such that functional operation can be verified (via power-aware simulation) and implementation can be properly directed.

However, today there are limited ways that supply connections to multi-rail macros (macros having multiple primary or bias supply sets) can be described in an abstract, generic manner—leading to the verbose and tedious practice of explicitly specifying supply set connections within UPF files. Rather than simply including a sub-component macro within the appropriate power domain, supply net connections must be made directly to the macro supply ports. The current problem is that these technology- and device-specific supply port connections within the UPF reduce design portability.

This paper proposes enhancements to the UPF standard and extensions to the Liberty model format that would allow multi-rail modules to benefit from the same automatic supply set connections based on power domain membership that connect standard cells today.

II. Definition of Terms

Unified Power Format	IEEE standard format for describing intended design power intent abstractly
Liberty model	*Synopsys* proprietary library format for describing technology cells
Retention	Preserving and restoring the state of an element when primary power is corrupted
Isolation	Preserving valid logic signaling on interfaces when driving logic is powered down
Power Domain	A portion of a circuit with common power requirements and resources
Supply Net	A power or ground connection path
Supply Set	A collection of supply nets related to a domain, grouped for convenience

III. Background

A. Liberty Attributes

Synopsys' *Liberty* model specification defines cell attributes such that EDA tools can map cells and associated pins to the correct signals during implementation. Included in the specification are attributes that allow identification of power-related functionality—as well as a fixed set[1] of attributes for the definition of power/ground pin types.

[1] Contemporary versions (2017.06 and newer) defined an additional *user_pg_type* attribute that accepts an arbitrary string; however, EDA industry support for this attribute is not well documented.

Table I identifies a select set of Liberty attributes used for mapping library cells to power intent function and describes how they are used by implementation tools[I].

TABLE I
LIBERTY ATTRIBUTES – CELL TYPE & FUNCTION

FUNCTION	ATTRIBUTE	VALUE	PURPOSE
Retention	retention_cell	*retention_cell_style* (string)	Identifies a cell as having retention capability. Cell 'style' allows use of more than one type of retention element.
	retention_pin	save/restore/ save_restore	Maps a cell pin to the save, restore, or both save & restore actions
	save_action/restore_action	L(low)/H(high)/ R(rise)/F(fall)	Indicates the trigger events associated with save/restore
Level-Shifting	is_level_shifter	true/false	Identifies a cell as a level shifter
	level_shifter_type	LH/LH/HL_LH	Specifies cell shifts low-high, high_low, or both (default)
	input_voltage_range	*floating value*	Allowed range of input pin
	output_voltage_range	*floating value*	Allowed range of output pin
Isolation	is_isolation_cell	true/false	Identifies a cell as an isolation cell
	isolation_cell_data_pin	true/false	Input data pin
	isolation_cell_enable_pin	true/false	Isolation control
Macro	is_macro_cell	true/false	Defines macro cells
	is_isolated	true/false	Specifies that a macro pin does not require external isolation
	isolation_enable_condition	*Boolean expr*	Enable isolation of macro pin
	is_analog	true/false	Specifies an analog pin

Table II lists a subset of power-specific pin attributes that can appear on pins of the types listed in Table I; of specific interest to this multi-rail macro discussion are the allowed values for *pg_type* and their expected meanings.

TABLE II
LIBERTY ATTRIBUTES – POWER-RELATED PIN ATTRIBUTES

ATTRIBUTE	VALUE	PURPOSE
related_power_pin	*pg_pin_name*	Assigned to logic pins to associate with related supplies
related_ground_pin	*pg_pin_name*	Assigned to logic pins to associate with related supplies
pg_type	primary_power	Specifies that pg_pin is a primary power source (default)
pg_type	primary_ground	Specifies that pg_pin is a primary ground source
pg_type	backup_power	Specifies that pg_pin is a backup (secondary) power source (for retention register, always-on logic, etc.)
pg_type	backup_ground	Specifies that pg_pin is a backup (secondary) ground source (for retention register, always-on logic, etc.)
pg_type	internal_power	Specifies that pg_pin is an internal power source for switch cells
pg_type	internal_ground	Specifies that pg_pin is an internal ground source for switch cells
pg_type	nwell	Specifies regular n-wells for substrate-bias modeling
pg_type	pwell	Specifies regular p-wells for substrate-bias modeling
pg_type	deepnwell	Specifies isolation n-wells for substrate-bias modeling
pg_type	deeppwell	Specifies isolation p-wells for substrate-bias modeling
std_cell_main_rail	true	Set on primary power pin to identify main cell power

B. UPF Syntax and Low-Power-Cell Connections

Within UPF descriptions, supply nets are often grouped in *supply sets*; each supply set can contain supply nets representing each of the main types of supply connection listed in Table II—*power*, *ground*, *nwell*, *pwell*, *deepnwell*, and *deeppwell*. Additionally, *power domains* are created with predefined handles for *primary*, *default_retention*, and *default_isolation* supply sets.

Strategies can be defined for the use of retention, isolation, and level-shifting cells using the *set_retention*, *set_isolation*, and *set_level_shifter* commands. These commands have options for specifying the design elements within the hierarchy to apply the strategy to, the control signals that enable the strategy (if needed), and the secondary supply set to be used (i.e., retention supply, isolation supply, or sink/source supply sets).

By leveraging these power-related attributes within Liberty models of technology-specific low-power cells, synthesis tools can select and insert retention, level-shifting, and isolation cells as described in the desired power intent. Place-and-route tools can then automatically connect the appropriate power and bias connections using the same domain and supply information provided within the power-intent: the primary supply sets functions of *power*, *ground*, *nwell*, and *pwell* are connected to the cell's supply pins having Liberty *pg_type* attributes *primary_power*,

primary_ground, *nwell*, and *pwell*, respectively. Cells that require secondary or backup supplies to perform their power management function (retention state elements and isolation cells, for example) have these secondary supplies (*default_retention*, *default_isolation*) automatically connected to pins having Liberty attributes *backup_power* and *backup_ground*.

C. Multi-Rail Macros

However, more complex cells—cells containing multiple primary or bias connections—require special attention; since multiple *primary_power* or *primary_ground* connections can exist, *pg_type* attribute information is not enough in itself for tools to conclusively make appropriate connections.

Furthermore, the connection of secondary or backup supplies is often predicated by the cell model containing one of the attributes from Table I identifying it as a retention, level-shifter, or isolation cell—as well as the definition of a retention, level-shifter, isolation strategy associated with the cell within the UPF file. Such attribute assignments and strategies may be inappropriate for macros that model these strategies internally, and as such they cannot be a requirement for establishing supply connections.

The inability to make definitive, attribute-based assignment of power pins leads to the verbose and tedious practice of explicitly specifying supply connections directly with UPF syntax (Figure I). These connections are technology-dependent—since the number and names of supply pins can vary across technologies or vendor implementations.

FIGURE I
UPF SYNTAX FOR SPECIFYING SUPPLY NET CONNECTIONS

```
connect_supply_net vcc_arr_out -ports {sram/VDDCE}
connect_supply_net vcc_per_out -ports {sram/VDDPE}
connect_supply_net vssd        -ports {sram/VSSE}
connect_supply_net vccdpslp    -ports {sram/BIASCNW}
connect_supply_net vcc_per_out -ports {sram/BIASNW}
connect_supply_net vssd        -ports {sram/BIASPW}
```

IV. MULTI-RAIL MACRO CONNECTION ISSUES

The underlying issues of attribute-based supply connections can be best seen using examples of commercially-available multi-rail macros currently in use. The supply pins and their associated *pg_type* attributes will be examined to show the current limitations of attribute-based supply net connections.

A. SRAM Examples

One place where the generic assignment of cell power supplies could be helpful is in the connection of supply nets associated with SRAM instances. However, Table III shows the difficulty of making such connections in an abstract, attribute-based way.

The ‘Vendor A’ type SRAMs specify both nwell bias and periphery power supplies as *primary_power*—making the connection of these supplies non-deterministic; any association of *primary_power pg_type* to a supply net would map to both the nwell bias and periphery power. There would be no way to specify a separate supply for one or the other. (Note that in this case, had the nwell bias been defined as pg_type *nwell* all connections could have been unique.)

The ‘Vendor B’ type SRAMs also have conflicts that make deterministic connection of supplies impossible: both BIASCNW and BIASNW pins are specified as *backup_power* when they are clearly meant for separate supply connections—one is switched while the other is always on.

TABLE III
SRAM TYPES WITH SUPPLY PINS, *PG_TYPES*

TYPE	SUPPLY PIN	PIN PG_TYPE	FUNCTION
Vendor A	VDDRET	backup_power	Retention power
	VDDB	primary_power	nwell bias (always on)
	VDD	primary_power	Periphery power (switchable)
	VSS	primary_ground	Common ground
Vendor B	VDDCE	backup_power	Core power (always on)
	VDDPE	primary_power	I/O power (switched)
	VSSE	primary_ground	Common ground
	BIASCNW	backup_power	Channel nwell bias (always on)
	BIASNW	backup_power	nwell bias / switched off
	BIASPW	backup_ground	pwell bias

B. *Analog Sensor Cell Example*

The example from Table IV describing the supply pins of a sensor cell is even more dramatic; fourteen (14) separate power and ground connections—to both digital and analog logic—share only two unique *pg_type* values: *primary_power* and *primary_ground*.

TABLE IV
ANALOG SENSOR CELL WITH SUPPLY PINS, *PG_TYPES*

TYPE	SUPPLY PIN	PIN PG_TYPE	FUNCTION
SensorCell	vssd	primary_ground	Digital ground
	vccdpslp	primary_power	Digital hibernate supply
	vccact	primary_power	Digital supply
	vsub_vic_csd	primary_ground	Substrate ground for isolation guard ring
	vssa_cref	primary_ground	Quiet analog ground for reference caps
	vssa	primary_ground	Quiet analog ground for IDAC components
	vssa_idac_term	primary_ground	Noisy IDAC termination ground
	vssa_idac_a	primary_ground	IDACA ground
	vssa_idac_b	primary_ground	IDACB ground
	vssa_sc	primary_power	Quiet analog ground for cap switching
	vdda_sc	primary_power	Analog supply for shield switches
	vdda_idac_term	primary_power	Analog supply for IDAC termination
	vdda_idac_ab	primary_power	Analog supply for IDAC A and B
	vdda	primary_power	Analog supply

The number of possible supplies to a macro like this demonstrate the potential complexity in describing connections to multi-rail macros in general.

Because macros can have such complex power requirements, many EDA tools limit automatic supply set connections to cells that explicitly identify as *retention*, *level-shifting*, or *isolation* cells.

V. POSSIBLE SOLUTIONS

Four methods for allowing for automatic supply port connections of multi-rail macros are presented in this section—with varying degrees of scalability and portability to new technologies. The first three methods presented are achievable today; the fourth and best solution requires action by industry governing boards and subsequent vendor adoption.

A. *Querying for Supply Pin Availability*

UPF 2.0 and newer UPF specifications provide a *find_objects* command; this allows for the querying of design objects (signals, instance names, ports, etc.) at a specified level of hierarchy matching an arbitrary regular expression. Conditional connection of supply pins can be achieved based on their existence:

FIGURE II
UPF SYNTAX FOR CONDITIONALLY MAKING SUPPLY NET CONNECTIONS BY QUERY

```
set vddce_port [find_objects my_ip/sram -object_type port -pattern VDDC*]
if {[llength $vddce_port] ne 0} {
   connect_supply_net   vcc_arr_out -ports " $vddce_port "
}
```

By providing rules for the connection of all possible supply nets within the set of technologies supported, any specific set of connections for the selected technology would be guaranteed.

While providing a degree of flexibility that would work in a limited set of cases, the drawbacks of this approach are clear:

- It's not scalable; new IP with new supply pin names would likely not "just work".
- It's not guaranteed to be deterministic; an arbitrary set of IP could have overlapping supply pin names that could potentially need to be connected differently.

B. *Sourcing Tech-specific Connections via Hierarchical Flow*

Another option would be the direct sourcing of an implementation-specific UPF by a reference UPF; rather than making the technology-specific connections directly within a UPF file, the technology-specific connections would be referenced from a technology-specific UPF—that would be selected based on environment/configuration settings:

FIGURE III
UPF SYNTAX FOR INCLUDING SUBMODULE UPF WITHIN A HIERARCHICAL FLOW

```
set_scope my_ip
load_upf ${DESIGN}/ip/rtl/ip_top/${TECH}/ip_top.upf
set_scope .
```

Note that the selection of the technology-specific file is still required; there must be knowledge somewhere in the system of what the implementation is. However, as shown above, a solution where the technology-specific file path is determined from configuration variables can be envisioned.

C. IP-based UPF Generators

One alternative to providing a library of static UPF files for supported technologies is to dynamically generate technology-specific UPF files with automation scripts capable of distinguishing among the supported technologies and outputting power-intent in line with the selected implementation. While functionally no different than the approach of sourcing technology-specific UPF files within a hierarchical flow (described above in Section B), maintenance can be more manageable, as the power-intent code that is common across technologies need only be updated in one 'generator' script.

D. Language/Standard Improvements

A longer-term solution could include Liberty or UPF specification enhancements to allow multi-rail macros more flexibility in describing related supply pins or supply pin types. Vendor support of 'user-defined' *pg_type* extentions[II] could allow the definition of additional attributes to help clarify cases like those discussed in Section III where *pg_type* alone does not provide a deterministic set of connections.

Also, the most recent UPF revision does not allow the mapping of arbitrary supply set functions to *pg_types*; only a power domain's *primary* function can be mapped; enhancing the *create_power_domain* command to allow association of a domain's *default_retention* supply set to the *backup_power pg_type* would allow more flexibility when including multi-rail macros within the domain.

VI. PROPOSED LANGUAGE EXTENSIONS

A. Liberty Model Attributes Enhancements

The difficulty in mapping the small number of *pg_type* possibilities listed in Table II to multi-rail macro power connections can be reduced simply by providing more options for *pg_type* to take on—as well as a well-defined method for mapping these to new or existing UPF concepts.

Table V presents a series of new *pg_type* values and how each could be defined to be connected to a UPF-described supply set.

TABLE V
PROPOSED LIBERTY *PG_TYPES* ATTRIBUTE EXTENSIONS

ATTRIBUTE VALUE	FUNCTION	UPF AUTOMATIC ASSIGNMENT
macro_io_primary_power	I/O padring power	PD.primary.power
macro_io_backup_power	I/O padring power (dual rail cell)	PD.default_retention.power
macro_io_ground	I/O padring ground	PD.primary.ground
macro_core_primary_power	Core macro power	PD.primary.power
macro_core_backup_power	Core macro power (dual rail cell)	PD.default_retention.power
macro_core_ground	Core macro ground	PD.primary.ground
macro_isolation_primary_power	Isolation supply	PD.primary.power
macro_isolation_backup_power	Isolation supply (dual rail cell)	PD.default_isolation.power
macro_isolation_ground	Isolation ground	PD.default_isolation.ground
macro_retention_primary_power	Retention supply	PD.primary.power
macro_retention_backup_power	Retention supply (dual rail cell)	PD.default_retention.power
macro_retention_ground	Retention ground	PD.default_retention.ground
macro_reference_primary_power	Analog reference supply	PD.primary.power
macro_reference_backup_power	Analog reference supply (dual rail cell)	PD.default_isolation.power
macro_reference_ground	Analog reference ground	PD.default_isolation.ground

Note that each new attribute has the *macro* prefix—distinguishing them as part of the extended set of macro-specific *pg_type*s. This *macro* prefix could be used to enable automatic connections without the need for a guiding strategies (as is the requirement for retention, isolation, and level-shifting); (1) If the cell has the attribute *is_macro*, and (2) it has power/ground pins of *pg_type* defined with the *macro_* prefix, then the proposed connections to the related power domain, supply set, and function can be made automatically.

The proposed values in Table V are meant to cover the set of common power/ground connection schemes for macros—separate pad/core power supplies, dedicated retention and/or isolation supplies, and analog/reference supply connections. While they wouldn't cover the full set of possibilities, they would provide a clear mapping between the macro power connections and the three common supply set handles provided for a power domain within UPF—*primary*, *default_isolation*, and *default_retention*.

B. UPF Language Enhancements

There are two paths for improving multi-rail macro connections via IEEE 1801 / UPF language enhancements: improving the likelihood of automatic connections based on *Liberty pg_type* attributes and providing more flexible methods for mapping supply sets to *Liberty pg_types*.

Create a 'default_backup' Supply Set Handle

The *create_power_domain* command already provides a method for mapping a supply set to an arbitrary *supply set handle*, and earlier revisions pre-defined handles named *default_isolation* and *default_retention* for providing supply sets for isolation- and retention-related cells within a domain. However, these supply sets were only mapped to cells defined to be isolation or retention cells that were inserted as part of an isolation or retention strategy.

A more flexible option would be to provide a default supply set handle name *default_backup*; this supply set would be automatic connected to the *pg_pin*s of cell macros within the domain whose *pg_type* are *backup_power* and *backup_ground*—without the requirement that the cell be of a specific retention or isolation cell type. While this simple mapping would not be sufficient for many multi-rail macros, the capability could allow simple macros to have their power mapping performed automatically.

Extend the 'create_power_domain -define_func_type' Argument

The *create_power_domain* also provides a method for mapping a domain's *primary* supply set handle's functions (i.e., *power, ground, nwell, etc.*) to a specific *pg_pin_type* attribute (*primary_power, primary_ground, etc.*). However, this option is only available for the *primary* supply set for a domain.

What is needed is the capability to map any existing supply set handle's functions to any arbitrary *pg_pin_type* attribute—including the *user_pg_type* attribute that can also be applied to a *pg_pin*. In fact, support for *user_pg_type* in this way could allow companies to assign attributes based on internal naming conventions—standardizing the interface to vendor-supplied IP in such a way as to allow their UPF power descriptions to then be vendor agnostic.

VII. CONCLUSION

Multi-rail macros can have complex power sourcing requirements that are well beyond the requirements of standard retention, isolation, and level-shifting cells; consequently, automatic, attribute-based supply net connection of these cells is often not possible.

Even for cases where a macro cell does contain a unique set of attributes that *could* allow automatic connections to be made, many EDA tools will not attempt these connections because they are focused on connections for cells that identify as retention, isolation, or level-shifting cells.

The underlying issue (abstracting technology-specific connections) can be accomplished by adopting a hierarchical flow—a common industry practice; however, extending the *Liberty* and UPF language specifications to allow more flexibility for macro-specific or user-defined attributes to play a role in associating supply sets with Liberty attributes could reduce the need for hierarchical abstraction or technology-specific UPF syntax in some simple cases.

VIII. REFERENCES

[I] Synopsys Logic Library Reference Manual, P-2019.03

[II] Ibid., page 234