DVCon 2012

Design & Verification Conference & Exhibition

February 28 – March 1, 2012

Is Power State Table(PST) Golden?

By

Ankush Bagotra, Neha Bajaj, Harsha Vardhan

R&D Engineer, CAE, CAE

Synopsys Inc.

SYNOPSYS®

Overview

- Low Power Design Today
- Unified Power Format (UPF)
- Low Power Design Flows
- Power State Table (PST)
- PST Complexities
- High Level Voltage Relationship Constraints (HLVRC)
- Case Study
- Applications of HLVRC
- PST Management(Some best practices)
- Conclusion
- Limitations

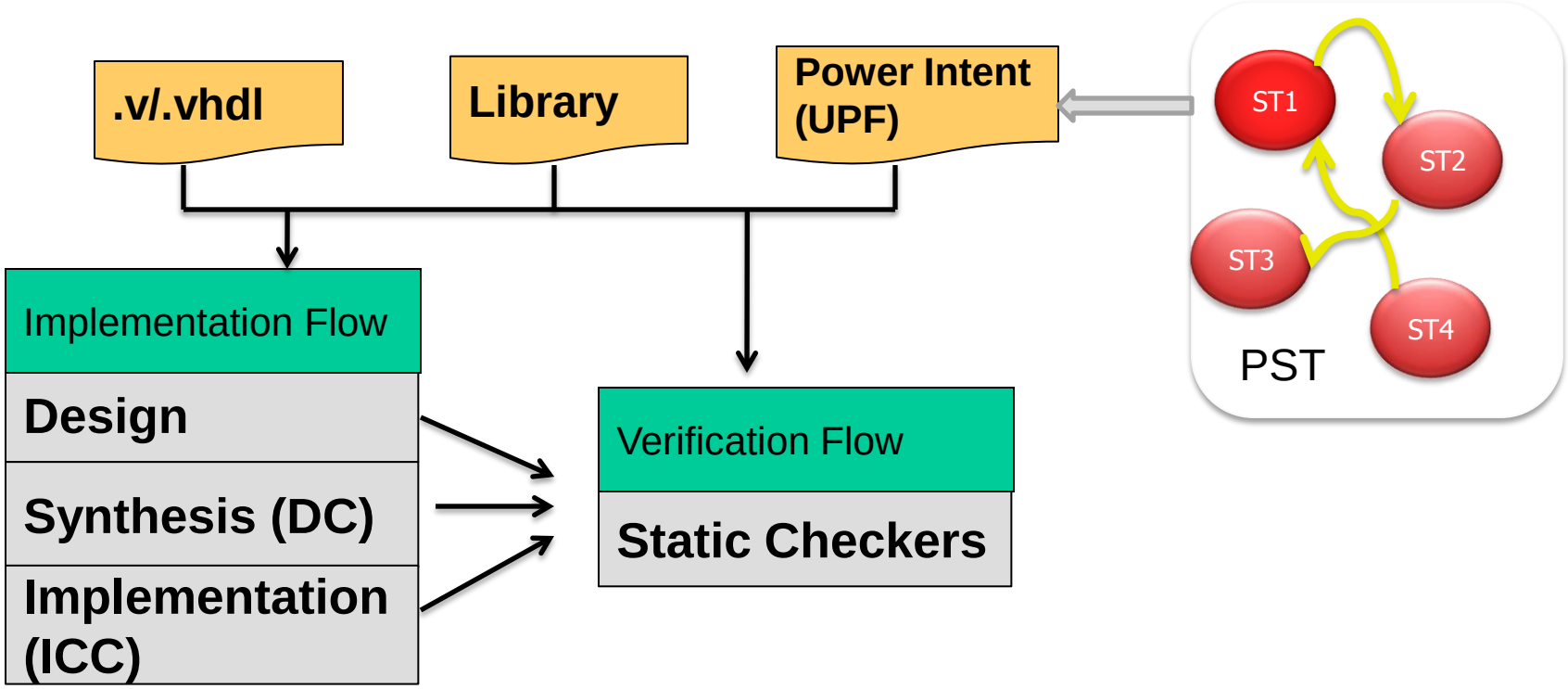
Low Power Design Today

- With Chips becoming complex :
  - number of power domains are increasing
  - hierarchical power domain distribution methodologies are becoming common.
- Power formats like UPF provides a consistent format to specify power-aware design intent and semantics
- Power State Table (PST) defined in UPF is used as a golden reference by implementation tools and static verification checkers.
- Extensive and thorough simulation ensures whether the PST coverage is complete or not

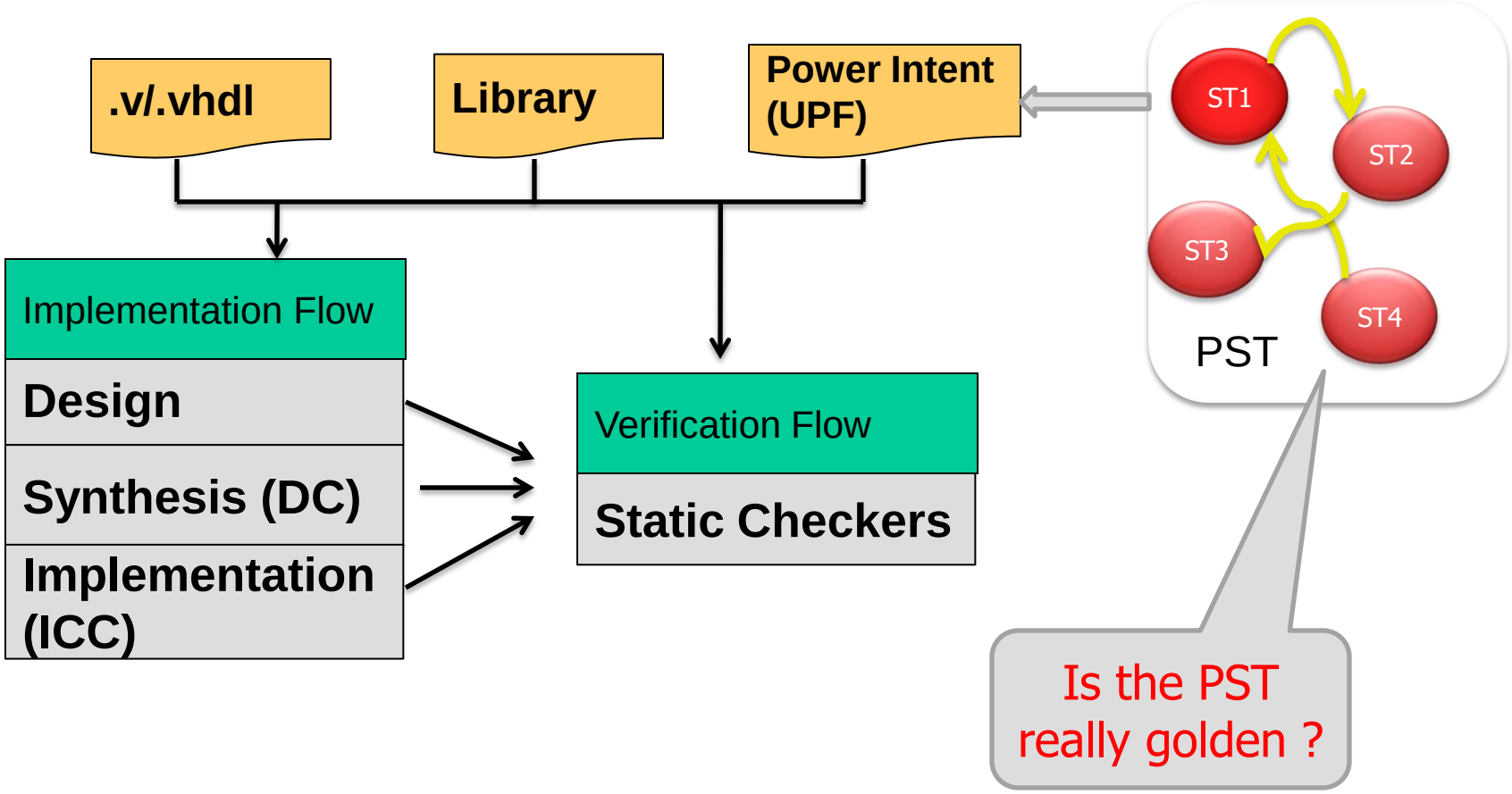
Unified Power Format (UPF)

- Industry standard extension of logic specification for low power intent
- Consistent semantics for verification and implementation
- UPF components :
  - Supply distribution network and switching
  - Power Domain and Power State Specification
  - Isolation, level shifting, retention rules and policies
  - Simulation semantics to accurately model power states
- Typical low power verification flows
  - Static verification
  - Dynamic simulation
  - Equivalence checking

Low Power Design Flows

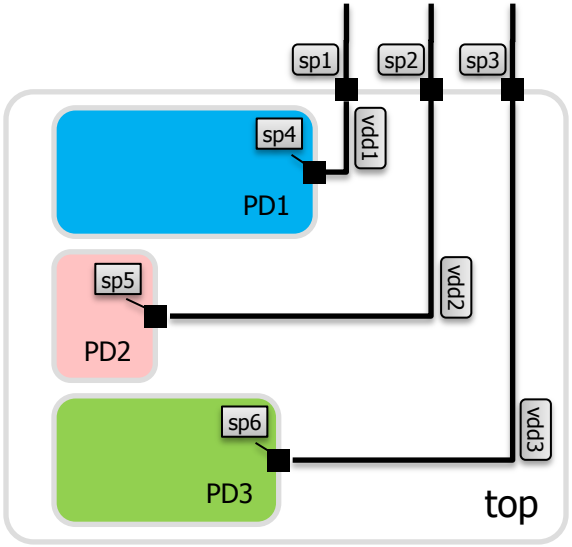


Low Power Design Flows



Power State Table (PST)

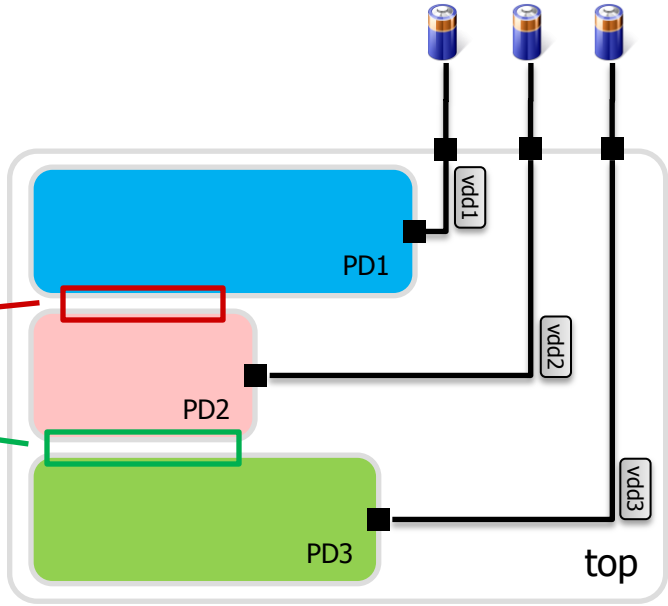
- Defines legal low power state space
  - Defines values for each of the supply net/port in design
  - Establishes the relationship among supply nets/ports
- Is defined on a design or at block levels
- Golden constraint for static verification and implementation tools



final_pst	vdd1	vdd2	vdd3
ALL_OFF	VDD1_OFF	VDD2_OFF	VDD3_OFF
ALL_ON	VDD1_ON	VDD2_ON	VDD3_ON
MODE1	VDD1_OFF	VDD2_ON	VDD3_OFF
MODE2	VDD1_OFF	VDD2_OFF	VDD3_ON

How PST is used

States	vdd1	vdd2	vdd3
S0	ON	OFF	OFF
S1	ON	ON	ON
S2	OFF	OFF	OFF



- vdd1 can be ON while vdd2 is OFF
  - Isolation policy is required between PD1 & PD2
- vdd2 & vdd3 cannot be switched separately
  - signals between those power domains do not need to be isolated

PST Complexities –Is PST Golden?

- Exponential state space for large designs
  - Theoretical vs. Practical
- State Reach ability
  - Legal vs. Illegal states
  - Dynamic verification can only prove whether a PST state is reachable or not
- Hierarchical Flows- PST merging
  - Under vs. Over constrained PST

High Level Voltage Relationship Constraints (HLVRC)

- High level low power architectural intent of design.
  - hierarchical rail order relationships
  - power network dependencies
- Significance
  - automatic derivation of elaborated constraints (PST)
  - automatic comparison and consistency checks on user supplied constraints (PST) before they are golden constraints for implementation and static verification

HLVRC Semantics

define_rail _name <rail_name> -value <voltage_value>

set_rail_order –order <number> -rail <rail_name> -rail <rail_name> …..

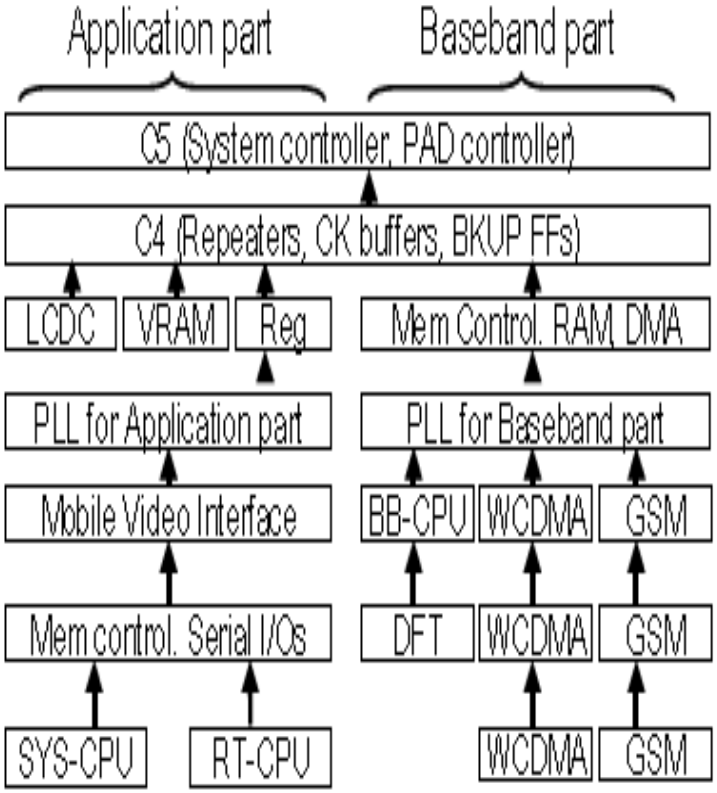
set_rail_constraint -main_rail <rail_name > -dependent_rail <rail_name> ….

define_rail _name defines the rails present in the design and their respective voltage values as per high level design intent.

set_rail_order is used to indicate the order of the rails. '0' order number indicates the rail is more 'on' than all other rails. The increasing order number indicates the rails are more relative off.

set_rail_constraint is used to define the dependency among rails of different order. There can be multiple rails dependent on a signal main rail.

HLVRC Significance - Ease of Representation



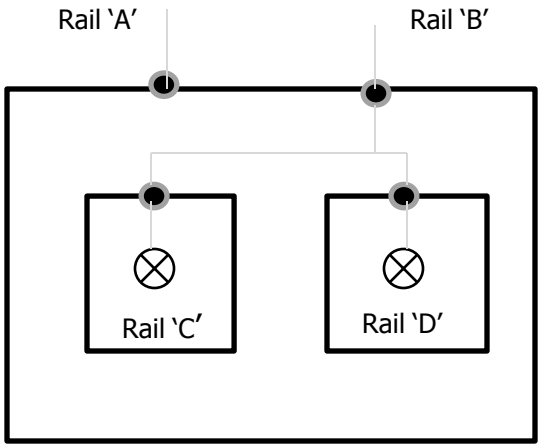
Ref : Hierarchical Power Distribution and Power Management Scheme for a Single Chip Mobile Processor. DAC, 2006

```
define_rail –name C5 –value {1.0} –value {OFF}
define_rail –name C4 –value {1.0} –value {OFF}
define_rail –name LCDC –value {1.0} –value {OFF}
define_rail –name VRAM –value {1.0} –value {OFF}
define_rail –name REG –value {1.0} –value {OFF}
define_rail –name PLL_app –value {1.0} –value {OFF}
……..
set_rail_order –order 0 –rail C5
 set_rail_order –order 1–rail C4
set_rail_order –order 2–rail LCDC –rail VRAM –rail REG –rail MEM_ctrl
set_rail_order –order 3–rail PLL_app –rail PLL_base
set_rail_order –order 4–rail Mobile_V –rail BB_CPU –rail WCDMA_1 – rail GSM_1
set_rail_order –order 5–rail MEM_serial –rail DFT –rail WCDMA_2 –rail GSM_2
set_rail_order –order 6–rail SYS_CPU –rail RT_CPU –rail WCDMA_3 –rail GSM_3

set_rail_constraint –main_rail C5 –dependent_rail C4
set_rail_constraint –main_rail C4 –dependent_rail  MEM_ctrl –dependent_rail VRAM
        –dependent_rail REG –dependent_rail LCDC
set_rail_constraint –main_rail MEM_ctrl –dependent_rail PLL_base
set_rail_constraint –main_rail PLL_base –dependent_rail BB_CPU –dependent_rail
        WCDMA_1 -dependent_rail  GSM_1
set_rail_constraint –main_rail BB_CPU –dependent_rail DFT
set_rail_constraint –main_rail WCDMA_1 –dependent_rail WCDMA_2
set_rail_constraint –main_rail GSM_1-dependent_rail  GSM_2
set_rail_constraint –main_rail WCDMA_2 –dependent_rail WCDMA_3
set_rail_constraint –main_rail GSM_2-dependent_rail  GSM_3
……
……
……
```

Case Study

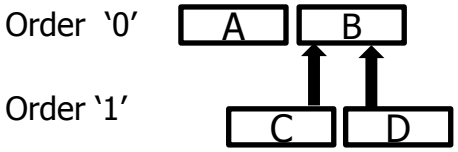
Topology



HLVRC

```
define_rail –name A –value {1.2} –value {OFF}
define_rail –name B –value {1.2} –value {OFF}
define_rail –name C –value {1.2} –value {OFF}
define_rail –name D –value {1.2} –value {OFF}
set_rail_order –order 0 –rail A –rail B
set_rail_order –order 1 –rail C –rail D
set_rail_constraint –main_rail B–dependent_rail C –
      dependent_rail D
```

Case Study Ordering



Golden PST Inferred

	A	B	C	D
State1	ON	ON	*	*
State2	ON	OFF	OFF	OFF
State3	OFF	ON	*	*
State4	OFF	OFF	OFF	OFF

•In the PST the '*' indicates don't care
•The maximum possible number of states for this topology is 16 but with the HLVRC inference, the states were reduced to 10.

Application of HLVRC

- Syntax Checks For Rails
  - A rail not specified in the PST defined in UPF but present in HLVRC

User Defined PST

	A	B	C
State1	ON	ON	*
State2	ON	OFF	OFF
State3	OFF	ON	*
State4	OFF	OFF	OFF

Golden PST

	A	B	C	D
State1	ON	ON	*	*
State2	ON	OFF	OFF	OFF
State3	OFF	ON	*	*
State4	OFF	OFF	OFF	OFF

Application of HLVRC

- Over Constraint/ Under Constraint PST specification in UPF
  - States not possible or extra validated against original architectural low power intent

Redundant PST State

	A	B	C	D
State7	ON	OFF	ON	OFF

User Defined PST

Missing PST State

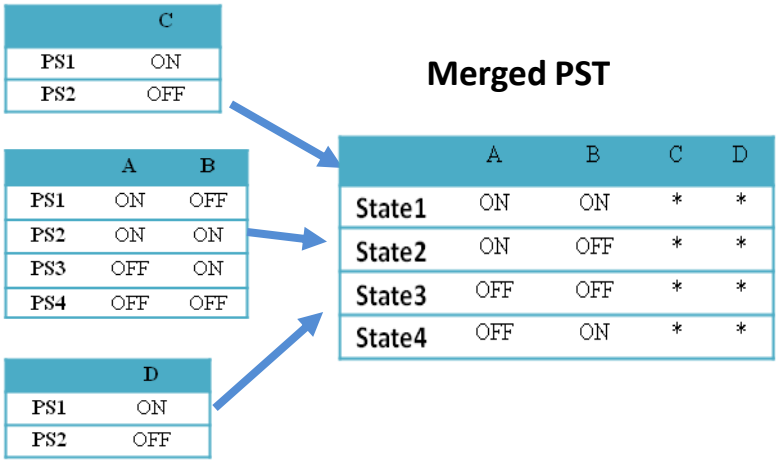
	A	B	C	D
State6	ON	ON	ON	OFF

Golden PST

	A	B	C	D
State1	ON	ON	*	*
State2	ON	OFF	OFF	OFF
State3	OFF	ON	*	*
State4	OFF	OFF	OFF	OFF

Application of HLVRC

- Merged PST
  - Validation for redundant or missing states during merging



	C
PS1	ON
PS2	OFF

	A	B
PS1	ON	OFF
PS2	ON	ON
PS3	OFF	ON
PS4	OFF	OFF

	D
PS1	ON
PS2	OFF

Merged PST

	A	B	C	D
State1	ON	ON	*	*
State2	ON	OFF	*	*
State3	OFF	OFF	*	*
State4	OFF	ON	*	*

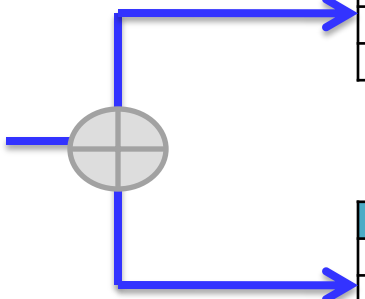
Golden PST

	A	B	C	D
State1	ON	ON	*	*
State2	ON	OFF	OFF	OFF
State3	OFF	ON	*	*
State4	OFF	OFF	OFF	OFF

PST Management (Some best practices)

- Multiple PSTs per scope

	SP1	SP2	SP3	SP4	SP5	SP6
State1	S1	S1	S1	S4	S4	S4
State2	S2	S2	S2	S4	S4	S4
State3	S3	S3	S3	S4	S4	S4
State4	S1	S1	S1	S5	S5	S5
State5	S2	S2	S2	S5	S5	S5
State6	S3	S3	S3	S5	S5	S5
State7	S1	S1	S1	S6	S6	S6
State8	S2	S2	S2	S6	S6	S6
State9	S3	S3	S3	S6	S6	S6



	SP1	SP2	SP3
State1	S1	S1	S1
State2	S2	S2	S2
State3	S3	S3	S3

	SP4	SP5	SP6
State1	S4	S4	S4
State2	S5	S5	S5
State3	S6	S6	S6

PS : Assumption all implementation & verification tools will have this consistent merging principle :

A "block" PST cannot make a legal state which is illegal according to a "top" PST. Neither can a "top" PST make a legal state that is illegal according to a "block" PST. Any state that is illegal according to any PST must be illegal. The final set of legal states is those that are not ruled out by any other PST.

PST Management (Some best practices)

- Use of don't cares or wild cards for similar rails in a PST state will make PST more concise and more readable

	SP1	SP2	SP3	SP4	SP5	SP6
State1	S1	S1	S1	S4	S4	S4
State2	S2	S2	S2	S4	S4	S4
State3	S3	S3	S3	S4	S4	S4
State4	S1	S1	S1	S5	S5	S5
State5	S2	S2	S2	S5	S5	S5
State6	S3	S3	S3	S5	S5	S5
State7	S1	S1	S1	S6	S6	S6
State8	S2	S2	S2	S6	S6	S6
State9	S3	S3	S3	S6	S6	S6

	SP1	SP2	SP3	SP4	SP5	SP6
State1	*	*	*	S4	S4	S4
State2	*	*	*	S5	S5	S5
State3	*	*	*	S6	S6	S6

Or

	SP1	SP2	SP3	SP4	SP5	SP6
State1	S1	S1	S1	*	*	*
State2	S2	S2	S2	*	*	*
State3	S3	S3	S3	*	*	*

Specification using wild cards reduced 9 states to 6 states

PST Management (Some best practices)

- Establish PST relationships using direct references
- Restrict supply-net availability to have optimal number of supply nets in PST.

Conclusion

- Acknowledged the problem of considering the PST defined in UPF as golden in view of complex low power SoCs with hierarchical PST, each having a large no of states.

- Addressed the problem by presenting HLVRC to generate a golden PST by capturing the architectural low power intent or validating a "so called golden PST from UPF" against the intent captured by HLVRC

Limitations

Framework does not honor the multiple voltage states for a supply net/port.

Thank You

Questions?