

# Common Power Format: Architecture & Support Capabilities

**Koorosh Nazifi**  
**September 28, 2006**



# CPF in the making

- Original concept formulated in 2005 –
  - Need to consolidate various specification formats for power within Cadence implementation tool flow
  - Quickly evolved to recognition of the need for tool and methodology automation to address design and verification challenges of advanced LP techniques



## Fundamental limitations to overcome

- Current infrastructure lacks support for LP specification
- Verilog (HDL) no longer capable of bridging the gap between design and verification for LP design
- Implementation centric techniques ignore the verification needs and don't provide flow support for exchange of LP design intent between various tools



# Architectural tradeoffs: Explicit vs. Implicit

- Lengthy debate among architects representing varying disciplines
- Both viable and technically feasible
- Customer requirements became principal guiding factor



# Key customer requirements

- Ease of adoption – Plug and play within current flow
- IP reuse and portability
- Consolidated view for ALL power related specification throughout design and verification flows



# Why CPF

- Eases adoption
  - Language neutral
  - No need to change golden RTL
- Supports IP reuse and portability
  - Eases migration of non power-aware RTL to power-aware
  - Supports multiple instantiation of a module with varying power architecture
- Single view for all power spec
  - Common specification for both design and verification
  - Eliminates duplication of power related information
- Eases architectural exploration & tradeoff



# Information supported in CPF

- Design intent and constraints
  - Power domain
    - Logical
    - Physical
    - Analysis view
  - Power mode
  - Power Logic
    - Level Shifter Logic
    - Isolation Logic
    - State-Retention logic
    - Switch Logic & Control Signals
- Technology information



# What is CPF

- ASCII specification file
  - Easily extendable to address new requirements
  - Similar to RTL, interpretive file, not command file
- Language structure supports multiple levels of abstractions for ease of use
  - Specification, implementation, analysis
- Full-feature language constructs to support
  - MSMV, PSO, SRPG, DVFS, ....
- Support IP reuse methodology
  - Ease of reuse for soft IP
  - Instantiation of modules with differing power structure

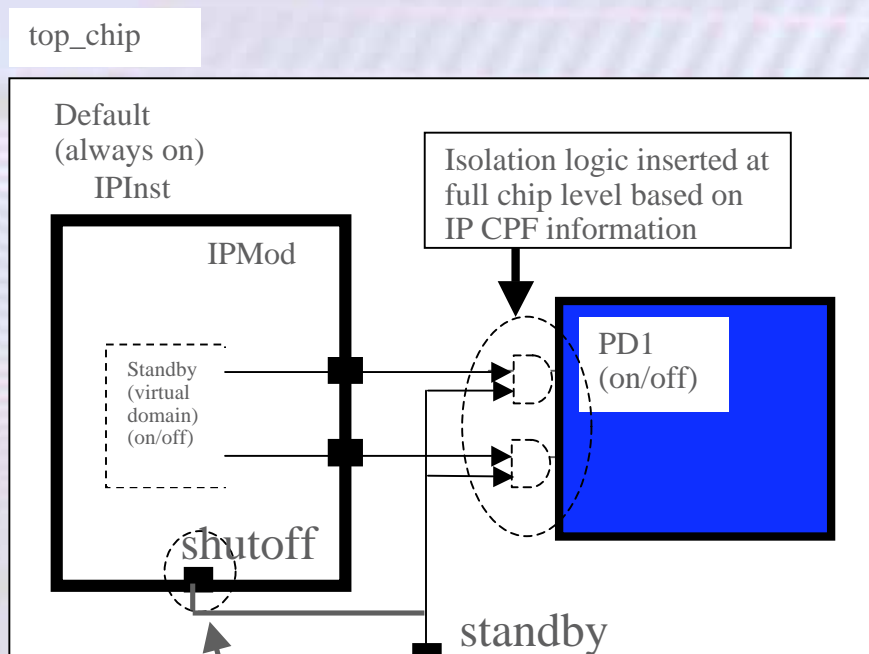


# Language structure

- Create – functional specification
- Update – constraint specification
- Define – technology related specification



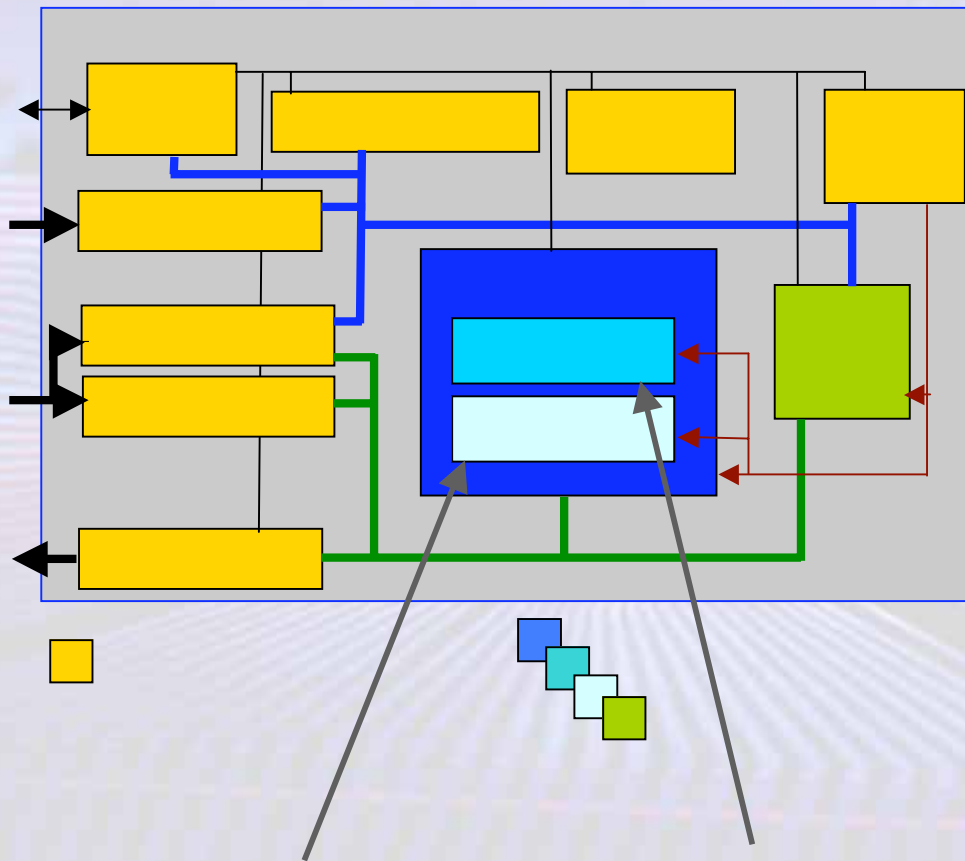
# Hierarchical support



there is no power down control port 'shutoff' at RTL



# Multi instance module support



both are instantiated from module 'alu'



## CPF – Solid base to build upon

- Representative of over 200 mm of technical innovation from key architects representing various disciplines in design, verification, and physical implementation tool flow
- Includes over 400 inputs from key industry leaders representing semiconductor, system, foundry and IP companies
- Fully addresses all of the key customer requirements
- Holistic approach to solving the LP design and verification challenges of our industry



# CPF Spec Release Plan

Products	2006				2007						
	1Q	2Q	3Q		4Q		1Q	2Q	3Q	4Q	
CPF Spec		v0.5, r1.7.1, 5/22	Advisory review delivery v0.8, 9/8	Advisory review feedback mtg, 9/14	Advisory input deadline, 10/15	Cadence formal response, 11/2	Release to Advisory and IEEE WG v1.0, 1/31				