An Approach to Create Scalable Power Management Verification Environment

Pranav Mopkar

Intel Corporation, Santa Clara, CA, USA

accellera SYSTEMS INITIATIVE

2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SANTA CLARA, CA, USA MARCH 2 - 5, 2026

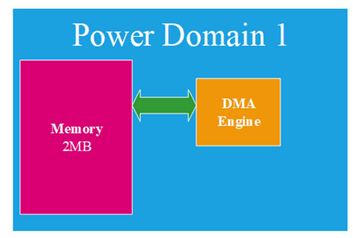
What are we trying to solve?

Problem: Verifying power management for complex designs with multiple, reconfigurable power domains (PDs) is a significant challenge.

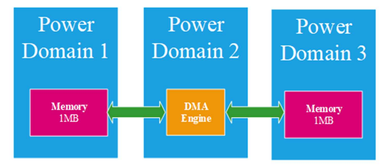
Solution Proposed: A new framework that decouples Power Domains (PDs) from Functional Blocks (FBLKs) to create a scalable and reconfigurable verification environment.

Limitations of Existing Methods

- ❖ Architectural Fluidity: Architects frequently reorganize logic by moving functional blocks between different power domains to optimize power consumption.
  - Example: A DMA engine and memory, initially in one PD, might be re-architected so the memory is split, and the DMA is moved to its own independent PD.
- ❖ Rigid Verification Environments: Traditional Verification Environments(VEs) often hardcode the association between functionality and PDs, viewing each PD as a single monolithic block.
- ❖ Major Drawback: When the design architecture changes, the tightly coupled VE requires a significant overhaul to move functionality-dependent code, resulting in major reconfiguration efforts.



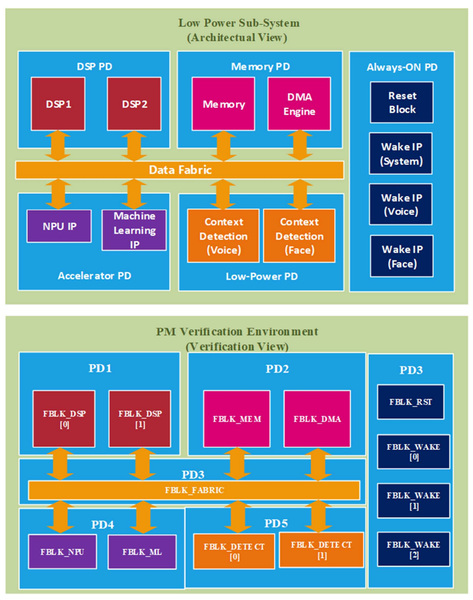
Configuration A



Configuration B

Framework Overview

- ❖ Core Principle: Treat PDs and FBLKs as separate, independent entities in the verification environment.
- ❖ Generic Templates:
  - ➢ PD Template: Models standardized PD behaviors like requests to the Power Management Unit (PMU) for power and clocks. Captures common signals like FET enables, clock requests/acks, and reset signals.
  - ➢ FBLK Template: Models common FBLK characteristics like control registers, functional clocks/resets, idle signals, and interrupts.
- ❖ Configuration Objects: Each template instance uses a configuration object to manage differences between the instances. This allows a generic template to be customized for a specific PD (e.g., its wake events) or FBLK (e.g., whether it has an interrupt) without changing the template itself.



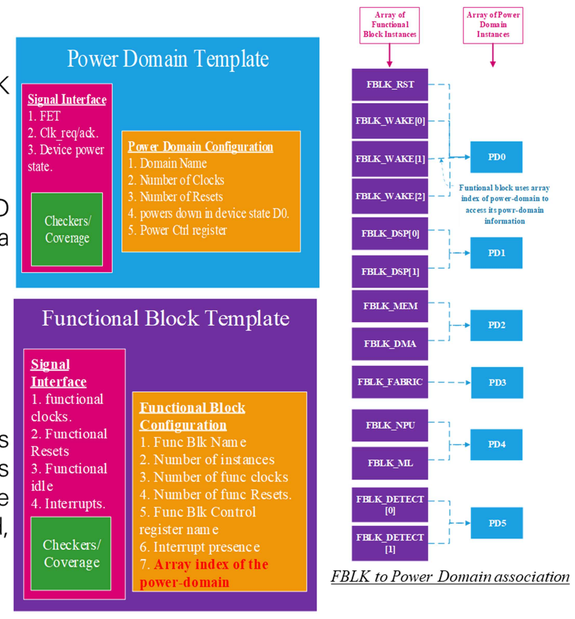
Environment Assembly and Reconfiguration

Building a Flexible VE

- ❖ Instantiation: Create arrays of PD and FBLK template instances in the environment.

  pwr_domain_template pwr_domains [NUM_PWR_DOMAINS];
  func_blk_template func_blks [NUM_FUNC_BLKS];
- ❖ Association: Link an FBLK to its containing PD by assigning the PD's array index to a configuration field in the FBLK instance.
- ❖ Example:

  func_blks[DSP_CORE].INST[0].cfg.power_domain = "PD1";
- ❖ Simple Reconfiguration: When an FBLK moves to a new PD, only the array index in this association needs to be updated. The core verification logic remains unchanged, significantly simplifying the process.



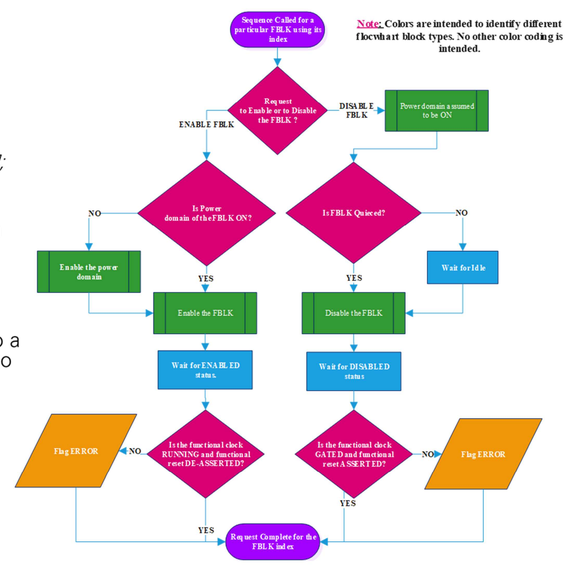
FBLK to Power Domain association

Scalable Stimulus, Checkers, and Coverage

Building a Flexible VE

- ❖ Instantiation: Create arrays of PD and FBLK template instances in the environment.

  pwr_domain_template pwr_domains [NUM_PWR_DOMAINS];
  func_blk_template func_blks [NUM_FUNC_BLKS];
- ❖ Association: Link an FBLK to its containing PD by assigning the PD's array index to a configuration field in the FBLK instance.
- ❖ Example:
  func_blks[DSP_CORE].INST[0].cfg.power_domain = "PD1";Simple Reconfiguration: When an FBLK moves to a new PD, only the array index in this association needs to be updated. The core verification logic remains unchanged, significantly simplifying the process.



Flowchart for Function Block enable/disable sequence

Results and Conclusion

Demonstrated Impact and Applicability

- ❖ Proven Results on a sub-system IP:
  - Applied to a highly configurable sub-system IP with up to 15 PDs and ~30 FBLKs.
  - VE reconfiguration time reduced from 1.5 weeks to a couple of days.
  - Total verification effort was reduced by two-thirds for every new SoC generation.
  - Successfully deployed across four generations of the IP, with other teams now adopting the framework.
- ❖ Conclusion:
  - The framework provides the best scalability returns for complex designs (4+ PDs, 6+ FBLKs).
  - The approach is applicable at both the subsystem and the full SoC level.
  - It serves as a foundational base upon which more advanced power management verification techniques can be built.

intel