Low Power Extension In UVM Power Management

Priyanka Gharat, Shikhadevi Katheriya, Avnita Pal

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Problem Statement/Introduction

Incorporating Power Architecture either is in conjunction with or in sequence to functional verification using different languages, many times with different team members using different tools, and divergent approaches leading to potential errors, almost a fourth dimension to our strategy leveraging the test bench architecture.

- Industry seems to be following a parallel path with respect to:
- Methodologies based test bench
- Power Architecture, including Unified Power Formats (UPF
- Both are fundamental requirements to IP and ASIC verification especially in the power saving mobile world.
- It would be more efficient to do Methodologies based Functional Verification and Coverage interleaved with Low Power Implementation.

We have noted previous works in power libraries for VMM and have corrected shortfalls and failings and have modified suitably to work within UVM, which is a much enhanced Methodology.

Proposed Methodology/Advantages

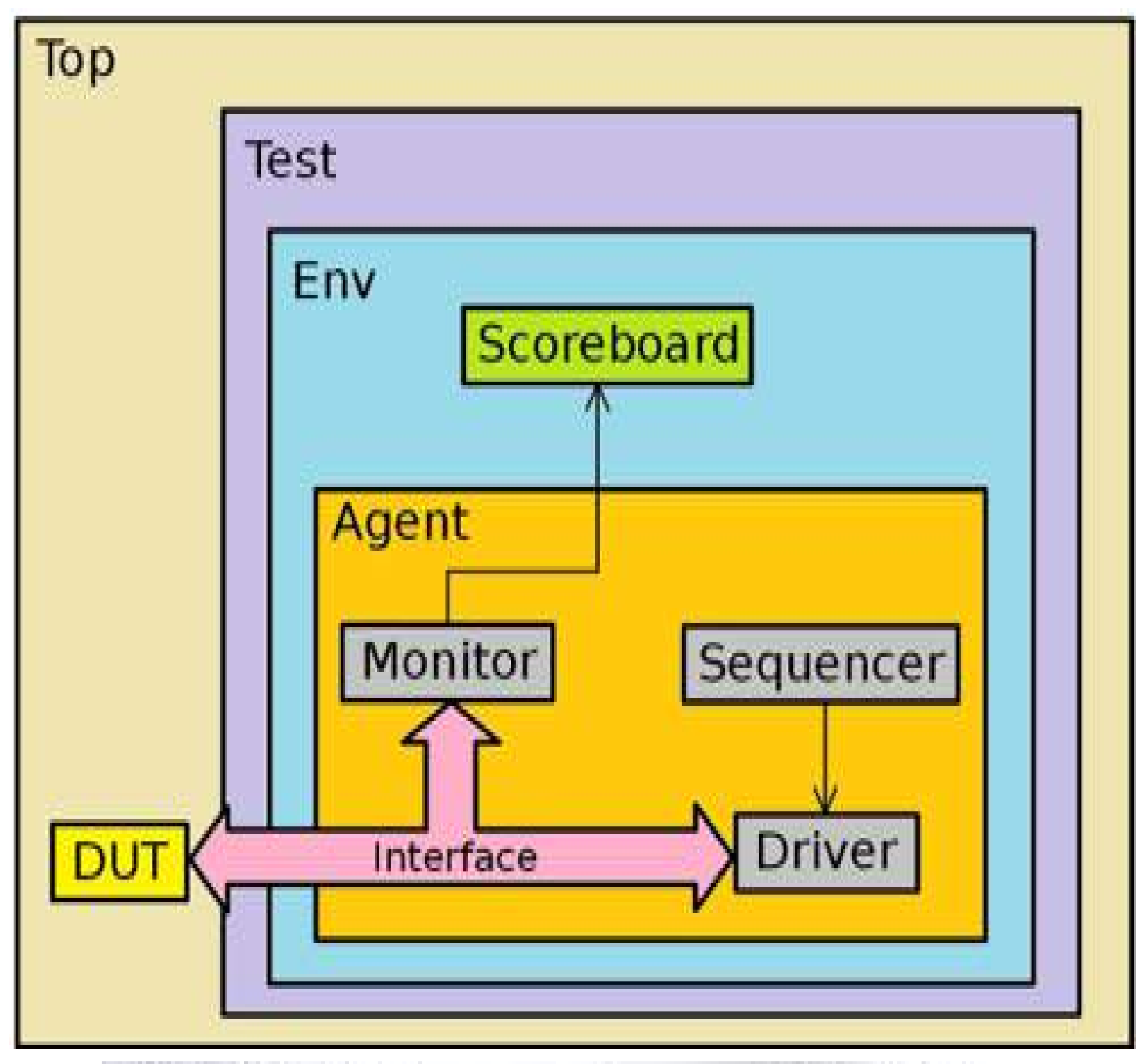
This Poster demonstrates Power Libraries classes built in SystemVerilog (UVM_Power) expanding UVM Package Library with Power Domains, Supply Sets, Switches, States and Low Power Strategies as Base Class which may be used within UVM Environment. These Power Base Classes are further built for multi-Cores, Bus Interface, Memory, Etc.

This proposal is to interleave Functional Verification Methodology and Power Architecture in a single existing and widely deployed methodologies based platform, like UVM.

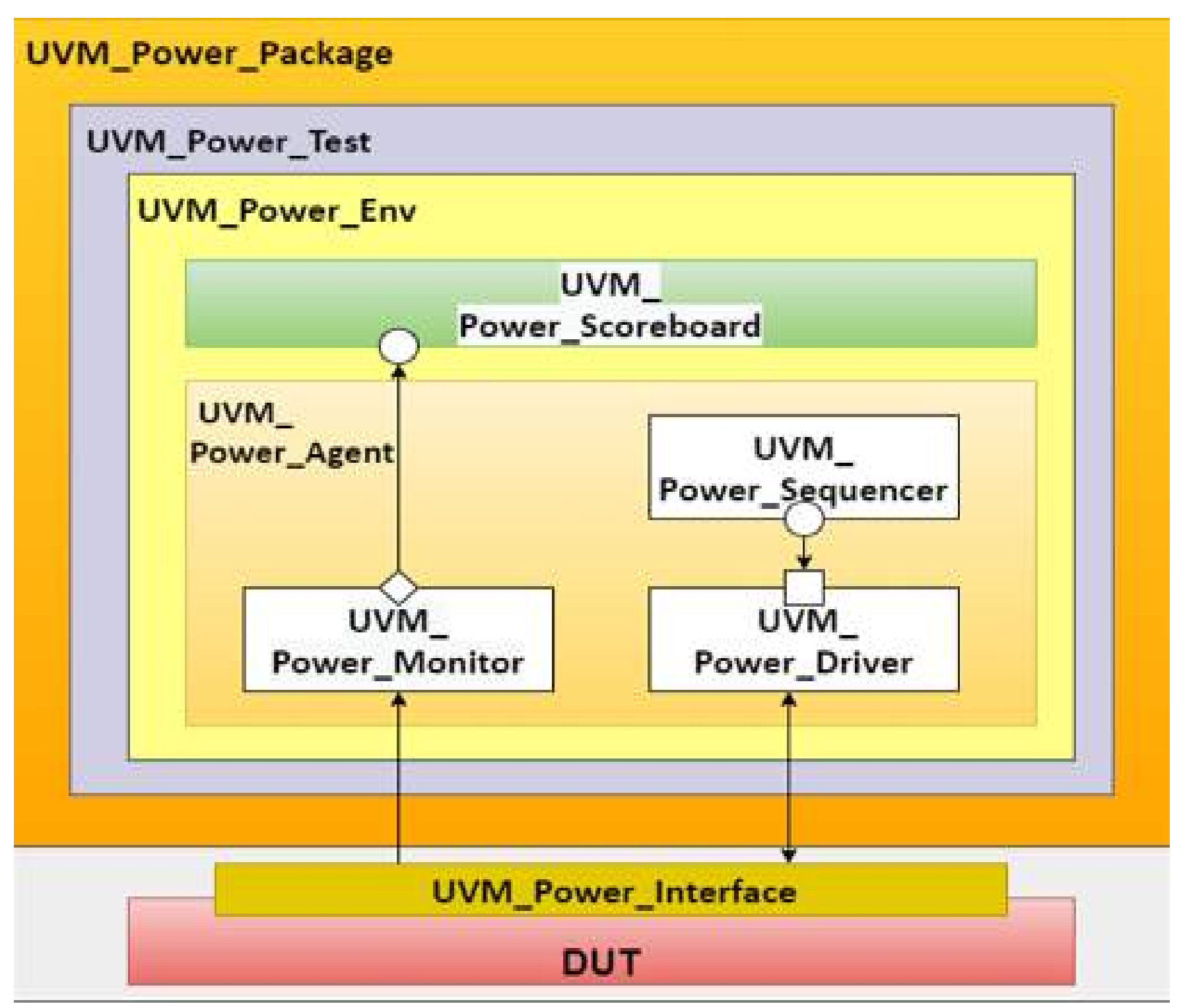
- With low power strategies, based on UPF and multi-core extensions, a low power or power aware designer or verification engineers would now be able to have a strategy/plan whilst the design/verification is being undertaken.
- As the needs for smaller and Low Power Aware designs needs increase doing the Power Architecture Strategy, especially the Verification as an afterthought post Functional Verification may lead to unwanted re-spins detrimental to costs as well as time to market guidelines.
- Bringing in Power Verification at an earlier stage will bring down the total time for incorporating power strategies resulting in far shorter design cycles.

Implementation Details/Diagram

An overall UPF structure is created using UVM classes which include different task as creating power domain, different scopes then supply nodes for each of the domain which are created. These classes are used as library and can be extended for creating structure based on DUT/SOC architecture.



Fig[1]. Hierarchy of UVM



Fig[2]. Hierarchy of UVM Power Domain

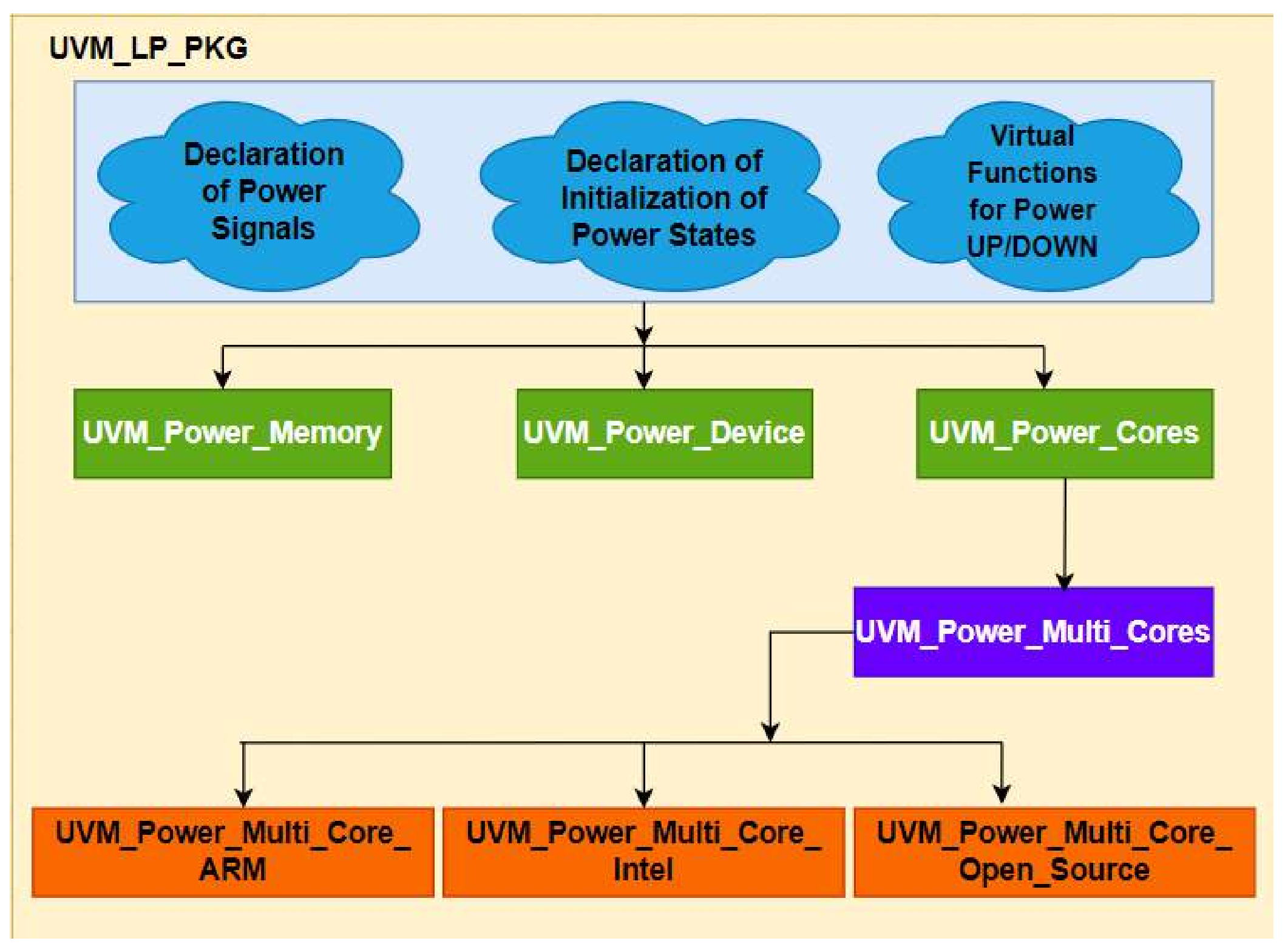
Implementation Details/Flow Chart

UVM design includes Top level base class of UVM_power and various extended packages for UVM_power_device, UVM_power_memory, UVM_power_core and further class extend for UVM_power_multicore. UVM_power_pkg which is top level package has predefined base classes for data based on power signal need for transaction, Initialization of different power state which can be used by the scope defined different level, Supply net used to connect to different Power domains.

This top level package has predefined base classes including Low power UPF strategies such as Level Shifter, Retention and Isolation based on the user defined design.

Based on the Power state of different Power domain various virtual functions, task and subroutine are being called.

This top level UVM_power_pkg is incorporated in test bench and further extended based on specification defined in UVM design.


UVM_LP_PKG
Declaration of Power Signals
Declaration of Initialization of Power States
Virtual Functions for Power UP/DOWN
UVM_Power_Memory
UVM_Power_Device
UVM_Power_Cores
UVM_Power_Multi_Cores
UVM_Power_Multi_Core_ ARM
UVM_Power_Multi_Core_ Intel
UVM_Power_Multi_Core_ Open_Source

Results Table

Sample code for Low Power Package Library

```
package uvm_power_pkg;
   import uvm_pkg::*;
   class uvm_low_power;
   //`uvm_component_utils(uvm_low_power); - Factory
Registration
   //function new (string name, uvm_component
uvm_low_power);
   //endfunction : new - Construction
class low_power;

function string create_power_domain(input string
domain_name,input string states, input int index);
  string power_domain[];
  power_domain=new[index];
  for(int i=0;i<index;i++)
     begin
        power_domain[i] = {"PD_",domain_name}; // domain
created here
        return power_domain[i];
     end
endfunction

function string create_supply_port (input string in_port, input
string type_of_signal);
    if(type_of_signal=="power_high")
      return {in_port,"_VDDH"};   ...
   else if(type_of_signal=="power_low")
      return {in_port,"_VDDL"};
   else if(type_of_signal=="ground")
      return {in_port,"_VSS"};
    else
      return "NULL";
  endfunction

function string create_supply_net(input string in_signal, input string
type_of_signal);
    if(type_of_signal=="power")
      return {in_signal,"_Pwr"};
    else if(type_of_signal=="ground")
      return {in_signal,"_Gnd"};
    else
      return {in_signal,"_net"};
  endfunction

function string connect_supply_net(input string in_port, input string
in_net);
    return {in_port,"_",in_net};
endfunction

function bit isolation_cell(input in_signal);
      isolation_cell = in_signal;
endfunction

function bit retention_cell(input in_signal,restore);
  reg memory;
      memory = in_signal;
      if(restore == 1)
       retention_cell = memory;
endfunction

function bit power_switch(input in_signal,switch_control);
      if(switch_control == 1)
        power_switch = in_signal;
      else  power_switch = 1'bx;
endfunction
endclass
endpackage
```

Conclusion

- Incorporating Power Management architecture within UVM methodologies alleviates challenges of functional verification engineer and power management divide
- Proposed in-built Power Domain Classes as extension to UVM Package as Library may be extended to Devices, multi-Cores, Memories, Bus Interface, Etc giving one package for implementation ease
- Consolidation of Functional Verification and Power Management will lead to reduced verification time and better chance to meet the time to market deadlines.

REFERENCES

1. UVM Community (accellera.org) https://accellera.org/community/uvm
2. Guide to changes in IEEE 1801-2013 (UPF 2.1) (techdesignforums.com)
3. Verification Methodology Manual for Low Power https://www.synopsys.com/company/resources/synopsys-press/vmm-low-power.html

Silicon Interfaces®
a software and vlsi design center