Leveraging UVM-based Low Power Package Library to SOC Designs

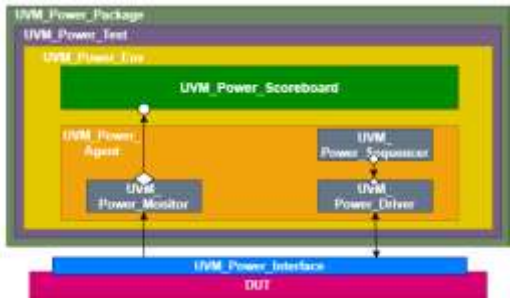
Shikhadevi Katheriya, Avnita Pal, Puranapanda Sastry

accellera SYSTEMS INITIATIVE

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

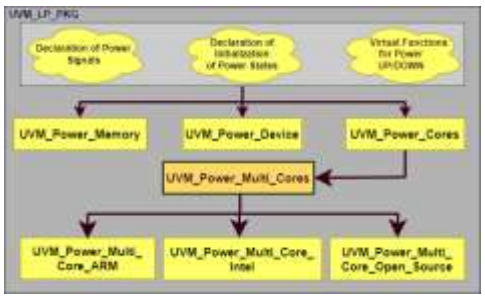
INTRODUCTION (or REQUIREMENTS)

- **Consolidating Power Management architecture within UVM methodologies alleviates the divide for functional verification and the power management engineers.** The library components with low power strategies, including UPF interleaves Functional Verification Methodology and Low Power Architecture in a single existing and widely deployed methodolo-gies-based platform, like UVM, and a low power or power aware designer or verification engineers would now be able to have a strategy/plan.
- This poster is proposing Low Power extension for multiCores, for example ARM multicores using SV interfacing to C using DPI) which may be extended to the SOC Design under Test.



OBJECTIVES

- The aim of the poster is to use the proposed in-built Power Domain classes in SystemVerilog (UVM_Power) as Extension to UVM Low Power Package Library Bus Interface for signals, Memory and Devices. However, for Cores, multi-Cores, (ARM, Intel, Open Source, Etc.), the LP Designs are in-built to the developer environment and hence, the UVM structure proposed in this poster incorporates the C Library within SV Packages.
- These Power Libraries are offered as Base Class which may extended by UVM. The idea is to have Low Power objects available to the Designer of an SOC which may be inherited and further extended.



RESULTS

- Here in the following source code for 1.
- C routines for power DOWN/UP source file are “arm_cortex _a53_assembly_code”, which contains all the C functions with ASM code for cortexA53 processor/cluster with 64-bit ISA.
- This Assembly Code is being enveloped using DPI calls into a SV Package by declaring source file for SystemVerilog Package “uvm_lp_core_pd_pkg”, which contains SV functions which is calling the C functions with the help of DPI import.

```
`include "arm_cortex_a53_assembly_code.c"

package uvm_lp_core_pd_pkg;

      ----------

      uvm_lp_core_power_standby_methods_e wf;
      uvm_lp_core_cmo_type_e cmo_type;
      uvm_lp_core_barrier_type_e barrier;

      import "DPI-C" function void disable_cache_func();
      import "DPI-C" function void clean_invalidate_dcache_func(cmo_type);
      import "DPI-C" function void cpu_extended_contrl_reg_func();
      import "DPI-C" function void barrier_func(barrier);
      import "DPI-C" function void transition_func(wf);
      import "DPI-C" function void debug_sig_func(bit DBGPWRDUP);
      import "DPI-C" function void activate_output_clamp_func(bit CLAMPCOREOUT);
      import "DPI-C" function void cpu_processor_power_func(bit nCPUPORESET);
      import "DPI-C" function void power_domain_cpu_func(bit PDCPU);
endpackage
```

RESULTS

- The full implementation needs to be done in an ARM environment in close collaboration and cooperation from ARM in the ARM Cortex Development environment.
- So, that would permit us to actually PowerUp and PowerDown cores. In this poster, we have observed the outputs using $display and C printf (using DPI) check the results.
- We are not showing those statements since they are only for validation purposes.
- Further, In the Assembler code which is essential for testing will run on ARM Development Environment

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include "svdpi.h"

//#include<ARMv6T2.h>

typedef enum {   CLEAN_BY_SETWAY,
                 INVALIDATE_BY_SETWAY,
                 CLEAN_INVALIDATE_BY_SETWAY,
                 .......
      }cmo_type_e;
```

```
//--------Routines for Power Down----------

void disable_cache_func() {
        asm volatile (
             "mrs x0, SCTLR_EL3\n\t"
             "bic x0, x0, #(1 << 12)\n\t"
             "bic x0, x0, #(1 << 2)\n\t"
             "msr SCTLR_EL3, x0\n\t"
             "isb"
             );
}
```

“arm_cortex _a53_assembly_code”

CONCLUSIONS

Proposed in-built Power Domain Classes as extension to UVM Low Power Package as Library for Cores, multiCores, (ARM, Intel, Open-Source Cores, Etc.), using Power management architecture shall bring in Power Verification at an earlier stage will bring down the total time for incorporating power strategies resulting in far shorter design cycles for SOC Designs.

The poster is considering ARM multi-core as a case study, but the same concepts may be applied to Intel, ARC or any Open-Source Cores).

Further, routines for Bus Interface for signals (AMBA AXI, CHI, PCIe, and Wishbone), Memory and Device needs to be written (like UPF type Power Domain architecture as previously proposed in [5] and [6] as given in Reference).

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/time to market guidelines.

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] Arm Cortex-A53 MPCore Processor Technical Reference Manual r0p4.

[4] Verification Methodology Manual for Low Power https://www.synopsys.com/company/resources/synopsys press/vmm-low-power.html.

[5] UVM and UPF: an application of UPF Information Model by Amit Srivastava, Harsh Chilwal, Srivatsa Vasudevan in DVCON 2019.

[6] Low Power Classes as extension to UVM Package Library by Shikhadevi Katheriya, Avnita Pal, et al, 59th Design Automation Conference, San Francisco, United States.

Add acknowledgements, organization involved, logo’s contact infomation here

shikha@siliconinterfaces.com
avnita@siliconinterfaces.com
sastry@siliconinterfaces.com

Silicon Interfaces®
a software and vlsi design center

Silicon Interfaces
info@siliconinterfaces.com
www.siliconinterfaces.com