2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION JAPAN

Optimizing UPF Integration Efficiency through Enabling Automation with UPVM for Unified Power Verification

Gopi Srinivas Deepala, Engineer – VLSI Design, Silicon Interfces, Mumbai, India
(*gopi@siliconinterfaces.com*)

Lakshya Miglaini, Engineer – VLSI Design, Silicon Interfaces, Mumbai, India
(lakshya@siliconinterfaces.com)

Sastry Puranapanda, Seniour Engineer – VLSI Design, Silicon Interfaces, Mumbai, India,
(sastry@siliconinterfaces.com)

***Abstract*- This paper presents an innovative approach to enhancing UPF integration efficiency by leveraging automated enablement with UPVM for unified power verification (UPV). We introduce a Python-based automation tool designed to seamlessly integrate UPF power strategies with UPVM class libraries. UPVM, a burgeoning open-source standard for Unified Power Verification Methodology™, serves as the vital link between UVM and UPF. The automation tool streamlines the process from RTL to GLS and GDSII stages for Design under Test (DUTs), emphasizing low-power functionalities. It simplifies complex tasks such as power domain management, hierarchical organization, supply port creation, interconnection, state transition handling, and comprehensive power strategy implementation. By bridging the gap between UPF and UVM methodologies, the tool effectively addresses challenges related to power management during verification. Furthermore, it ensures strict compliance with low-power criteria by facilitating the seamless translation of power objectives from RTL to GLS/GDSII. Through its holistic and integrated approach, this tool aims to revolutionize low-power semiconductor design practices, leading to heightened power efficiency, error reduction, and smoother transitions across diverse design development stages**

Keywords- Unified Power Verification Methodology (UPVM), Unified Power Format (UPF), Universal Verification Methodology (UVM), Automation, Power Management, Power Verification.

I. INTRODUCTION

Researchers are actively integrating power verification methodologies into semiconductor design processes. For instance, the merging of UPF and UVM methodologies addresses power management challenges during verification. A recent development is the creation of a script called aUPVM.py, a significant advancement in implementing UPF strategies in semiconductor design. This script simplifies UPF file generation by guiding developers through steps such as initializing the UPF file with the appropriate version and selecting the top module from the design hierarchy.

The script's notable feature is its ability to create power domains based on the design hierarchy, ensuring compatibility with existing UPF files before allowing developers to add the UPF file. This enhances the precision and efficiency of power domain creation, critical for effective power management. Additionally, the script facilitates the creation of ports, nets, and supply sets, along with associated supply sets and logic ports. It establishes connections between logic nets and generates power switches and states, contributing to comprehensive low-power strategy implementation in UPF files.

Integration with UPVM libraries further enhances the script's capabilities. Utilizing System Verilog classes and packages, the script implements UPF files based on low-power strategies defined in lpdut files, aligning with industry standards. After generating the low-power UPF file, power verification occurs during UVM verification phases. The script interfaces with scoreboards to compare testbench power strategies with RTL DUT power scenarios, ensuring accurate and reliable semiconductor designs.

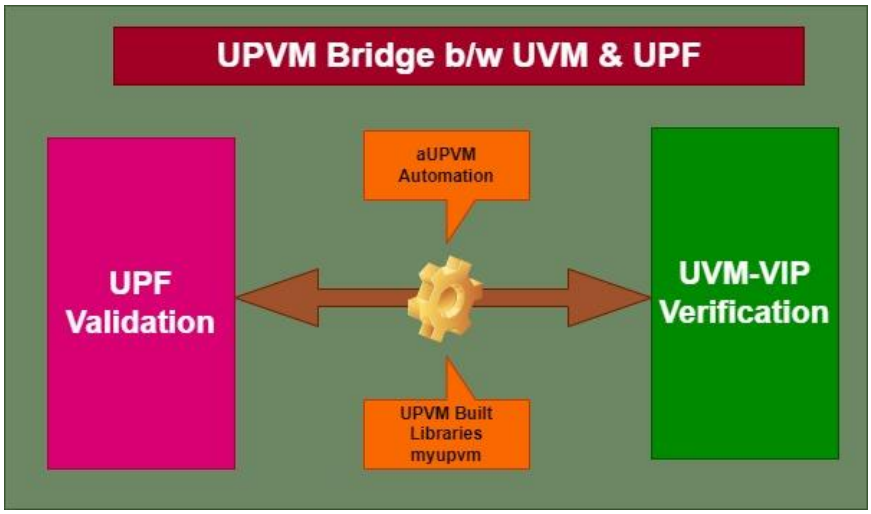

Figure: Unified Power Verification Methodology (UPVM) - Bridge

The aUPVM script represents a significant advancement in automating the creation of UPF (Unified Power Format) strategies for semiconductor design. Its primary objective is to streamline the generation of UPF files by guiding developers through an intuitive interface.

A standout feature of the script is its interactive nature, which allows it to gather user preferences regarding specific UPF strategies. Through a series of prompts and options, the script adeptly navigates users through the steps necessary to craft UPF strategies that align with their design objectives.

```
*********************************************
**** aUPVM Provided by Silicon Interfaces ****
*********************************************

Provide the UPF version : 2.0
```

Figure: aUPVM automation Script

Additionally, aUPVM assists users in initializing the UPF file with the correct version and promptly identifies the directory containing the design files. It then presents the design hierarchy, enabling users to easily identify the top module and ensuring accuracy in UPF file generation.

```
/home/gopisrinivas/Workspace/PA/PCIe_latest
Is This Directory Contains the Your Design Files (yes/no): yes
System Verilog Files Found in this Directory
pcie_tb
    pcie_top
        pcie_receiver
            pcie_tlp
            pcie_phy
                pcie_byte_unstrip
                pcie_phy_cdr
                    pcie_block_align
                    pcie_pll
                    pcie_elastic_buffer
                    pcie_s2p
                pcie_decoder_10x8
                pcie_packet_filtering
            pcie_dll
        pcie_transmitter
            pcie_phy_tx
                pcie_phy_packet_framing
                pcie_phy_byte_striping
                pcie_phy_p2s
            pcie_tlp_tx
            pcie_dll_tx
```

Figure: Design Hierarchy

Throughout power domain creation, the aUPVM script prompts users to designate elements within the hierarchy that necessitate power domains. Following this, it displays the module hierarchy for each module, simplifying the selection process.

```
Available modules:
pcie_rx
/pcie_rx/tsl
/pcie_rx/phy_rx
/pcie_rx/phy_rx/us
/pcie_rx/phy_rx/cdr
/pcie_rx/phy_rx/cdr/ba
/pcie_rx/phy_rx/cdr/pll
/pcie_rx/phy_rx/cdr/eb
/pcie_rx/phy_rx/cdr/s2p
/pcie_rx/phy_rx/d0
/pcie_rx/phy_rx/pf
/pcie_rx/dll
pcie_tx
/pcie_tx/phy_tx
/pcie_tx/phy_tx/pf_tx
/pcie_tx/phy_tx/strip
/pcie_tx/phy_tx/p2s
/pcie_tx/tlp
/pcie_tx/dlp
```

Figure: Module Hierarchy for selecting the Power Domain elements

```
Do want to set this path - Enter yes:
Do you to find Directory with in path - Enter goto:
Do you want to go back one Directory - Enter goback: goto
Insert the Directory from the list: PA
/home/gopisrinivas/Workspace/PA
Avalible Directories in this path:
/home/gopisrinivas/Workspace/PA
        .visualizer
        cluster
        tool_example
        PCIe
        PCIe_latest
```

Figure: Directory Selection Procedure

To align with UPF 2.0 standards for directory selection, please navigate through the directories and select the specific directory where you wish to generate the UPF file. Utilize the provided commands to move up, down, or across directories. Once you have reached the desired directory, confirm your selection to begin UPF file generation.

When integrating a UPF file in accordance with UPF 2.0 standards, the automation process first reads the power domains within the UPF file. Subsequently, it compares these domains with the current design hierarchy. The addition of the UPF file will only proceed if there is a match between the power domains in the UPF file and the elements in the existing design hierarchy. Any mismatch will trigger an error, indicating the failure of UPF file addition. Ensuring alignment between the power domains specified in the UPF file and the elements present in the current design hierarchy is crucial before proceeding further.

```
Do want to set this path - Enter yes:
Do you to find Directory with in path - Enter goto:
Do you want to go back one Directory - Enter goback: yes
/home/gopisrinivas/Workspace/PA/PCIe

        pcie_upf1.upf

        pcie.upf

        pcie_latest2.upf

        pcie_latest1.upf

        pcie_latest.upf

        pcie_final.upf

        pcie_latest3.upf

No more UPF files are Available
```

Figure: Listing the UPF Files with in the Path

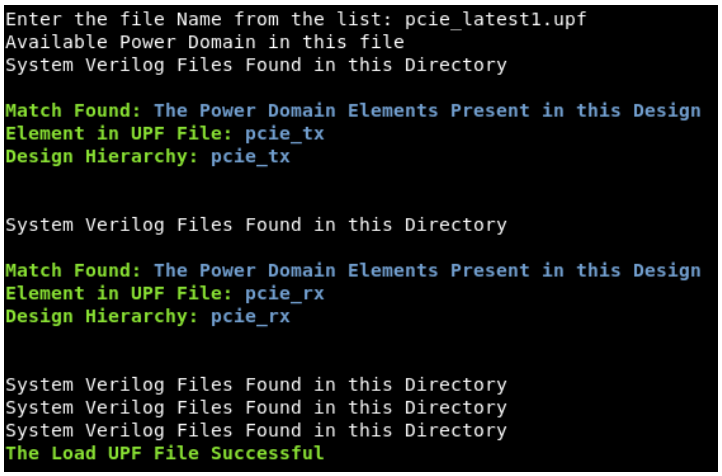

Figure: Listing the UPF Files with in the Path

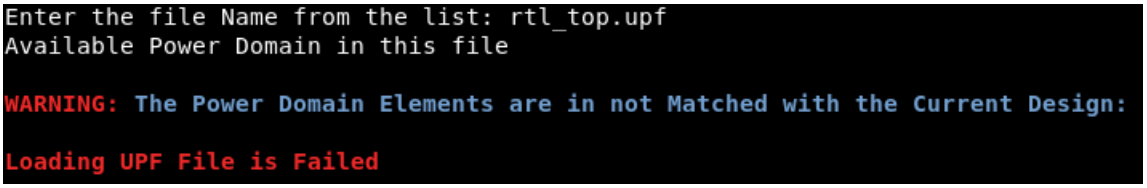

Figure: Match Fail Elements Comparison

The automation simplifies users' selection of preferred strategies for UPF file creation by offering a streamlined approach. Users can choose from a comprehensive range of options, spanning tasks like creating nets, logic nets, and ports, and seamlessly connecting these elements. Additionally, users have the flexibility to establish supply sets and associative supply sets, configure power switches and states, and incorporate state and logic expressions into their UPF file. Furthermore, the automation facilitates the integration of level shifters, isolation cells, and retention cells into the power management scheme. Users maintain full control over these strategies, enabling them to tailor the UPF file generation process to their specific design requirements. After users have made their selections, the automation consolidates all output into an lpdut file. This lpdut file serves as the foundation for generating the UPF file, leveraging the capabilities of UPVM libraries from myupvm.sv to ensure the effective implementation of the specified power management strategies.

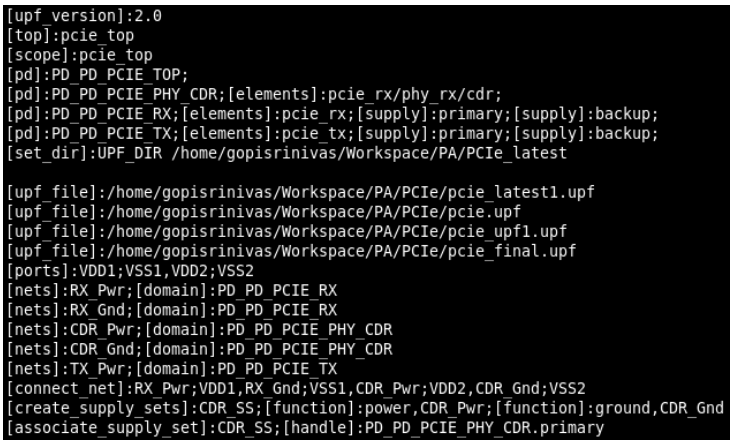

Figure: lpdut File outcome of Automation

Following the generation of the lpdut file through Python automation, the subsequent step involves leveraging UPVM libraries to construct UPF files based on this output. These libraries, developed using System Verilog

packages, are specifically crafted to utilize the utilities and functionalities required for UPF file construction. The myupvm.sv file consolidates these UPVM built-in libraries, streamlining the process of UPF file creation.

Users may be prompted to execute the myupvm.sv file, which utilizes the UPVM libraries to interpret the lpdut file and generate the corresponding UPF file. This ensures the seamless translation of low-power strategies outlined in the lpdut file into UPF files, aligning with industry standards and regulations.

Moreover, UPVM libraries serve a critical function in accessing UPVM strategies within the UVM testbench and Scoreboard. These libraries are seamlessly integrated into the UVM testbench codebase, enabling access to UPVM strategies during UPF file creation and verification. Additionally, they are incorporated into the Scoreboard to facilitate comparison between testbench power strategies and RTL DUT power scenarios during verification. This comprehensive validation process ensures the effectiveness of the power management strategies implemented in the UPF file within the UVM environment.

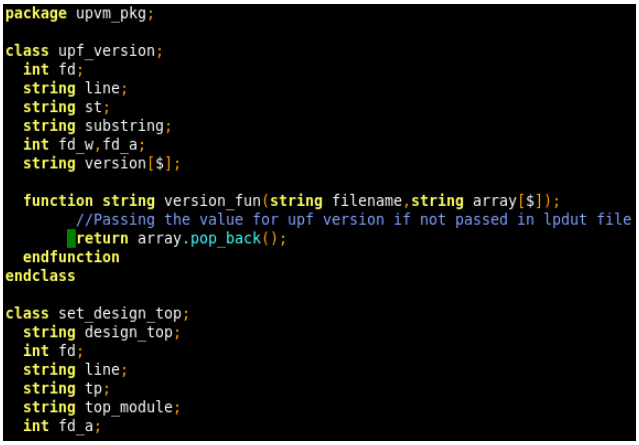

Figure: UPVM Libraries Package

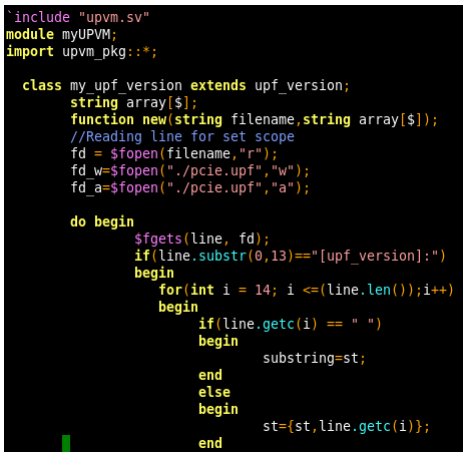

Figure: UPVM Libraries Accessing in UVM VIP

The primary function of the aUPVM tool is to facilitate UPVM generation, offering several valuable applications to the semiconductor industry:

1. **Efficient UPF File Generation:** aUPVM streamlines the process of generating UPF files, enhancing efficiency in semiconductor design.
2. **Standardized Power Domain Management:** UPVM ensures consistency and standardization in power domain management practices.
3. **Low Power Design Implementation:** The tool enables the implementation of low-power design strategies, promoting energy efficiency in semiconductor devices.
4. **Streamlined Design Workflows:** aUPVM contributes to smoother design workflows, reducing complexity and optimizing resource allocation.
5. **Enhanced Power Verification:** UPVM-based verification methods enhance the accuracy and effectiveness of power verification processes.

UPVM-based Low Power Verification offers significant benefits, including cost reduction and accelerated time-to market by integrating low-power verification early in the design process alongside functional verification. UPVM-based designs are poised to play critical roles in various industries:

1. **Mobile Devices:** UPVM-based devices can expedite the release of smartphones and tablets, potentially advancing them ahead of current technology by half a generation.
2. **Automotive Electronics:** UPVM-based processors can minimize the launch time gap for advanced vehicle technologies reliant on semiconductor chips.

3. **Medical Devices:** Time-efficient semiconductor components enabled by UPVM can expedite the development of medical devices, potentially improving their timely deployment and impact on saving lives.
4. **Data Centers:** UPVM-based devices used in data centers can be delivered ahead of schedule, enhancing data processing capabilities and efficiency in managing vast amounts of data.

II. RESULT

Testing the aUPVM.py tool on a low-power PCIe design, where automation was employed to generate UPF constructs using UPVM libraries, showcased promising outcomes. Notably, a significant reduction in the complexity of writing UPF code was observed, indicating notable simplification factors. Moreover, the seamless integration with UPVM libraries facilitated early-stage power validation, enhancing efficiency in the verification process.

```
class scoreboard;

my_upf_version upf_ver;
my_hirearchy_c my_hy;
my_set_design_top msdp11;
my_set_scope ssp;
my_create_power_domain my_crt_pd;
my_create_supply_port my_crt_sp;
my_create_supply_net my_crt_sn;
my_connect_supply_net my_cnt_spyn;
my_set_domain_supply_net my_sdsn;
my_create_logic_net_c my_crt_lg_nt;
my_add_port_state my_aps;
my_create_pst my_crt_pst;
my_add_pst_st my_apst;
my_set_retention_c my_srnt;
my_set_isolation_c my_sit;
my_set_level_shifter_c my_sls;
my_power_switch_c my_pwr_stch;
my_supply_sets my_ss;
my_load_upf_c my_ldupf;
my_associate_supply_set_c my_asociat_ss;
my_add_power_state_c my_apwrs;
my_set_dir_c my_sd;
```

Figure: UVM Scoreboard instantiation for Strategies Verification

```
create_power_domain PD_PCIE_TOP
create_power_domain PD_TX -elements { pcie_tx } \
        -supply  {primary} \
        -supply  {backup} \
create_power_domain PD_RX -elements { pcie_rx } \
        -supply  {primary} \
        -supply  {backup} \
create_power_domain PD_TXPLL -elements { pcie_tx/phy_tx/pll }
create_power_domain PD_PHY -elements { pcie_rx/phy }
create_power_domain PD_CDR -elements { pcie_rx/phy/cdr }
create_supply_net Pwr -domain PD_TX
create_supply_net Gnd -domain PD_TX
create_supply_net PwrTxpll -domain PD_TXPLL
create_supply_net GndTxpll -domain PD_TXPLL
create_supply_net Pwr -domain PD_PHY
create_supply_net Gnd -domain PD_PHY
create_supply_net Pwr -domain PD_CDR
create_supply_net Gnd -domain PD_CDR
create_supply_net sw_out -domain PD_PHY
connect_supply_net Pwr -ports {VDD1}
connect_supply_net Gnd -ports {VSS1}
connect_supply_net PwrTxpll -ports {VDD2}
connect_supply_net GndTxpll -ports {VSS2}
add_port_state PST1 \
        -state {NORMAL 1.0} \
        -state {SHUTDOWN 0.8} \
        -state {OFF 0.0}
```

Figure: UPF File Generated by UPVM

III. COUNCLUSION

The utilization of UPVM libraries is integral to both the creation and verification of UPF files for power management in semiconductor designs. These libraries play a crucial role in streamlining the generation of UPF files, ensuring alignment with design requirements and industry standards. By integrating UPVM strategies into the UVM testbench and Scoreboard, a comprehensive verification and validation of power management strategies are achieved, facilitating early-stage Unified Power Verification. This integration not only enhances the reliability and efficiency of semiconductor designs but also simplifies the UPF file creation process. Additionally, it enables thorough examination and debugging of low-power features, thereby contributing significantly to the overall success of semiconductor design projects.

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