2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION JAPAN

Optimizing UPF Integration Efficiency through Enabling Automation with UPVM for Unified Power Verification
UPVMによる統合電力検証の自動化を実現し、UPF統合効率を最適化

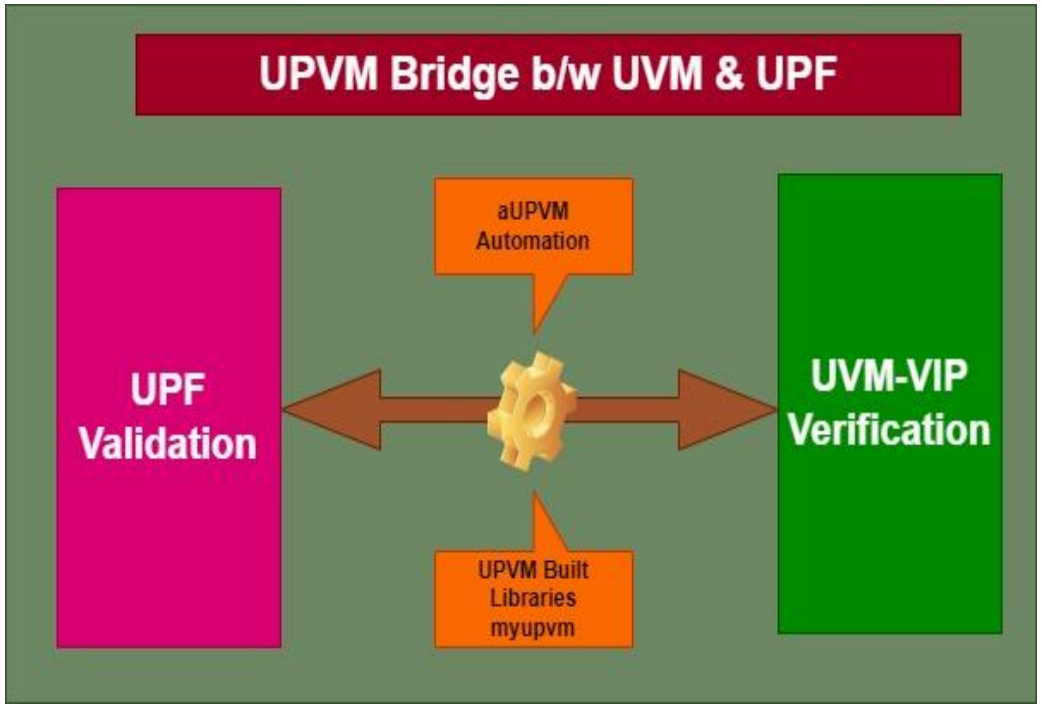
Gopi Srinivas Deepala, Lakshya Miglani, and Sastry Puranapanda

Silicon Interfaces®
a software and vlsi design center

accellera SYSTEMS INITIATIVE

Kon'nichiwa and Introduction

- **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.**
- **Incorporating Power Architecture seems to be** more like an afterthought **post Functional Verification**, almost a fourth dimension **to our strategy leveraging the test bench architecture.**
- **Leading EDA tools providers tend to have tools with Functional Simulation with UVM switches and** *a separate* **Low Power and Power Aware strategies.**
- **It would be** *more efficient* **to do Methodologies based Functional Verification and Coverage along with Low Power Implementation.**



Our Primary Implementations (so far)

- **DAC 2022** – “Low Power Classes as extension to UVM Package Library”
- **DVCon INDIA 2022** - “Low Power Extension in UVM Power Management”
- **DVCon Japan 2023** - “Integrating L1&L2 Cache for multi-Core UVM based extended Low Power Library Package” (DVCon Japan 2023 -「マルチコアUVMベースの拡張低消費電力ライブラリパッケージ向けL1&L2キャッシュの統合」)
- **DVCon Europe 2023** – “Unified Architecture of L1 L2 Cache with Low Power Extensions for MultiCore UVM-based Library Package”
- **DVCon Taiwan 2023 –** “UVM-based extended Low Power Library package with Low Power Multi-Core Architectures”
- Our past work’s are shows Implementation of the UPVM Library Packages for the Power Validation.
- Integrating the power strategies with in a UVM verification using the UPVM Library Packages.

Main Idea (本旨)

Open Source Standard for UPVM™ (UPVM™ のオープンソース標準)

UPVM is *an emerging standard* **library** for **unified Power Verification™** within the **SV/UVM environment** for the purposes of *early-stage* integration of power management and strategies (UPVMは、SV/UVM環境内での統合型Power Verification™のための新しい標準ライブラリであり、電力管理と戦略の早期統合を目的としています。) (along with DUT Design/Verification) and has

a) a standard **library package (標準ライブラリパッケージ)** consisting of SV Classes for power management/strategies

b) a **myUPVM templates** wherein the Low Power Designer **extends the Library Package Classes** for the **DUT under consideration (低電力デザイナーが検討中のDUTのライブラリパッケージクラスを拡張するmyUPVMテンプレート)**

c) **a structured query based automation tool (構造化クエリベースの自動化ツール)** which generates an intermediate file for the myUPVM templates to be populated configuring the power strategies for the DUT

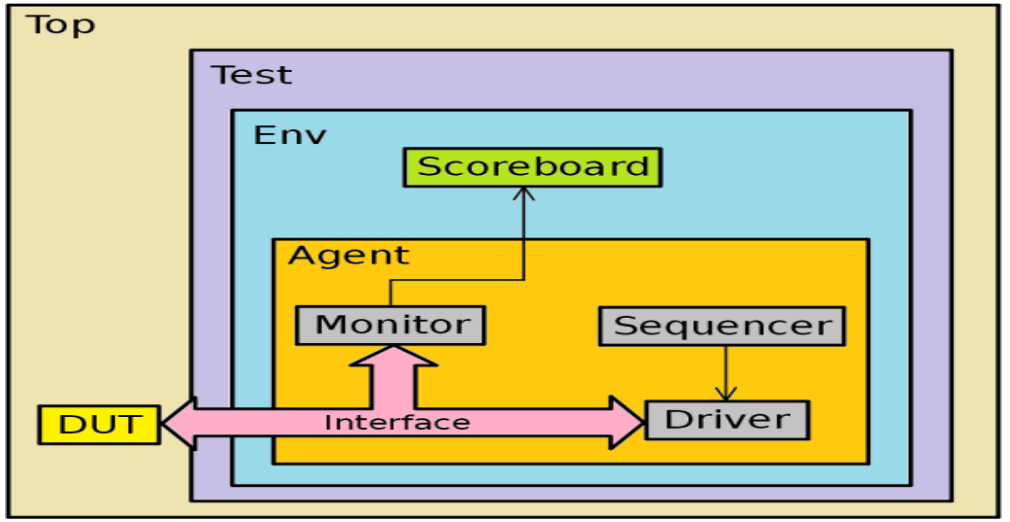
d) **Scoreboard** (スコアボード) to validate the power strategies

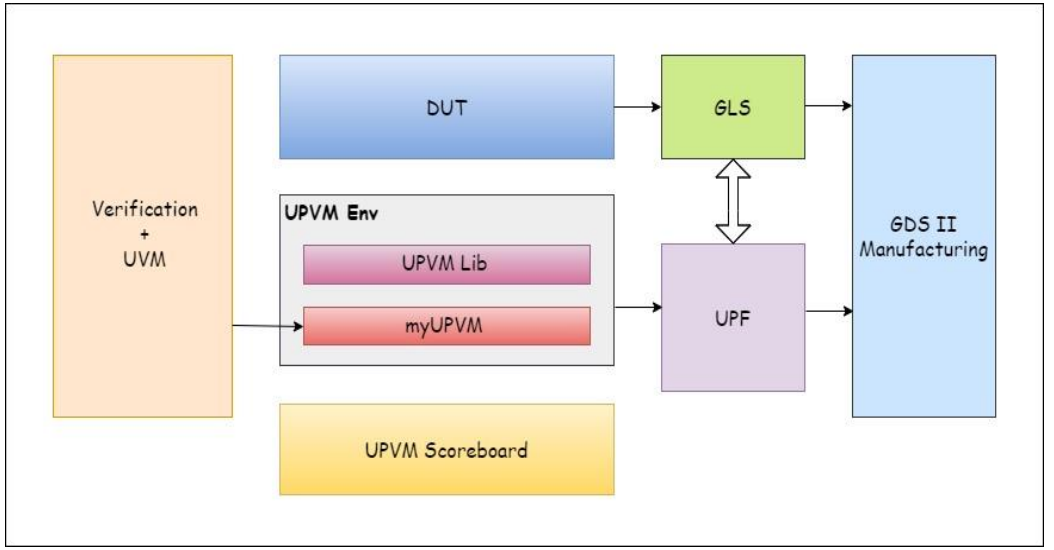
e) **generate a UPF file on compilation (コンパイル時にUPFファイルを生成する)** and simulation of the full SV Code.

The effort is to interleave Functional Verification Methodology and Power Architecture in SystemVerilog, like a single existing and widely deployed methodologies-based platform, like UVM.

UPVM Library Package (日本語は以下になります。)

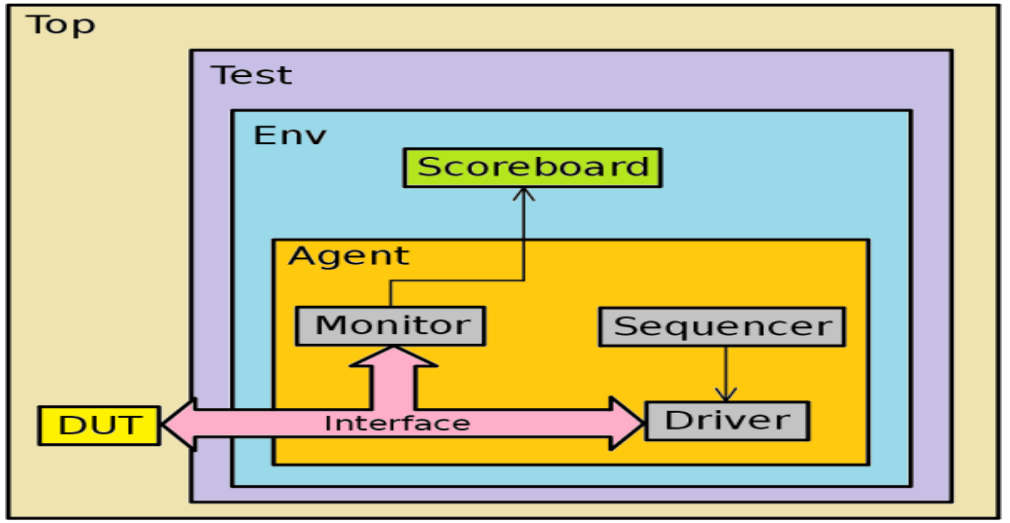
- The Low Power library is provided as in-built Power Domain classes within a package (UPVM Package) as Extension to UVM Package Library to be used by DUT designer as part of the testbench.
- Our implementation for Power Libraries are for Power Domains, Ports, Supply Nets/Sets, Switches, States and Low Power Strategies.
- These Power Libraries are offered as Base Class which is used within UPVM Scoreboard for Power Strategy Validation.
- UVM Power Classes for Device, Memory, Bus Interface for signals (AMBA AXI, ACE, CHI, PCIe, Wishbone) is now extended to Core, multi-Core, (ARM_Core, Intel_core, Open_Source_core), Etc

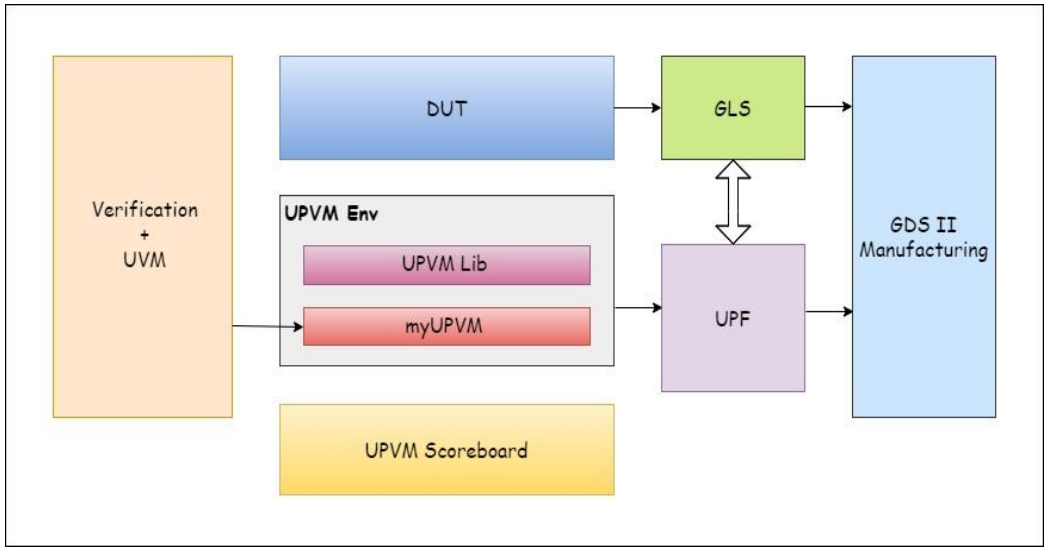




UPVM ライブラリ パッケージ

- 低電力ライブラリは、DUT 設計者がテストベンチの一部として使用する UVM パッケージ ライブラリの拡張機能として、パッケージ (UPVM パッケージ) 内の組み込み電力ドメイン クラスとして提供されます。
- 当社の電源ライブラリの実装は、電源ドメイン、ポート、電源ネット/セット、スイッチ、状態、低電力戦略を対象としています。
- これらの電源ライブラリは、UPVMスコアボード内で電源戦略の検証に使用される基本クラスとして提供されます。
- デバイス、メモリ、信号用バス インターフェイス (AMBA AXI、ACE、CHI、PCIe、Wishbone) の UVM 電力クラスが、コア、マルチコア (ARM_Core、Intel_core、Open_Source_core) などに拡張されました。

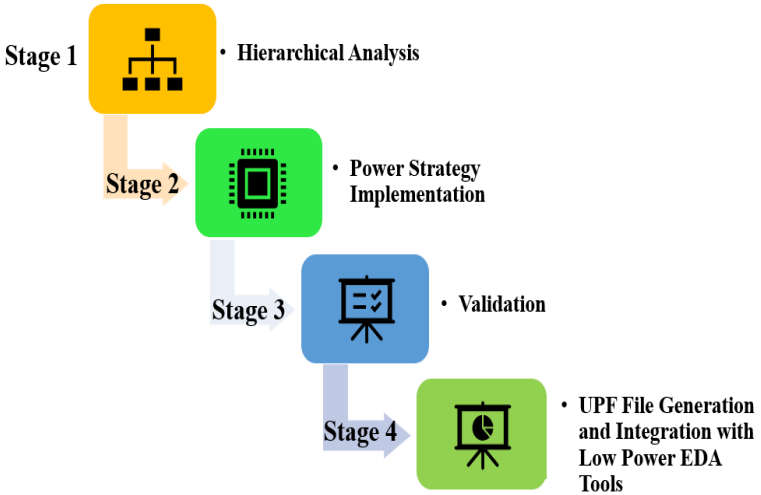




Automation WITH aUPVM™

- Python-based tools
- Interactive query-based tool which works with the Power Designer to identify UPF Power Management and Strategies
- Hierarchy Chart generator from DUT/TB directory based modules and instance of modules
- Smart guides to ensure
  - Elements are within scope
  - Power Domains listed for Nets, Switches, Retention, Isolation, Level Shifters
- Generates an internal format file for myUVPM automatically, which in turn generates UPF

aUPVM – Implementation (（日本語訳は次です。）)



Step-by-Step Automation for Power Verification with Functional Correctness:

Stage 1: Hierarchy Analysis with Python Script

- Initiate the process of employing Python-based tool to analyze the design hierarchy with and set_directory.
- Extracted Hierarchy guides the selection of set_design_top and set_scope, influencing power domain decisions.
- The Code Management constructs for load_upf and save_upf is incorporated.

Stage 2: Implementation of Power-Saving Strategies within the Python Script

- Implementing power domains, ports, supply sets, and power-saving strategies like retention, isolation, and level shifters.
- Automation scripts remain pivotal, ensuring optimal power configuration throughout the design.

Stage 3: Reading the Python intermediate file outputs in SystemVerilog

- In this stage the SystemVerilog myUPVM modules interfaces with UPVM SV library, reads the python generated internal format output file and collects the data to customize myUPVM template modules.
- Scoreboard will compare the library data and analyse to see if any power contradictions are present.

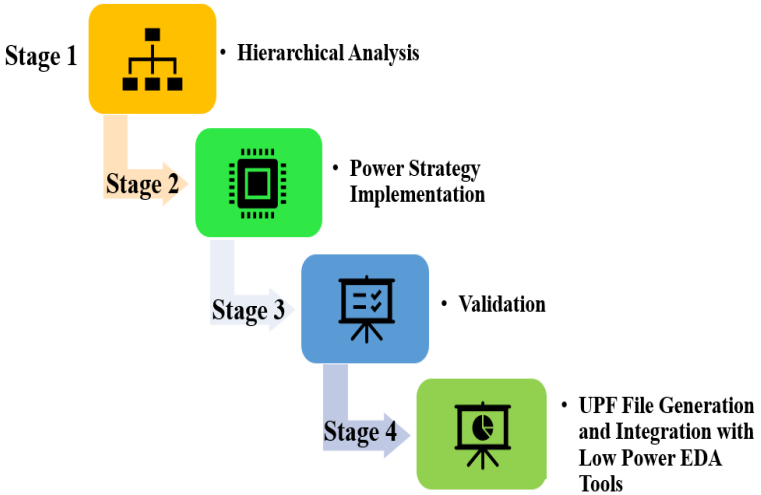
Stage 4: Automation Resumes for UPF File Generation

- Generate a custom UPF file based on the extracted data.

Stage 5: Integration with Low Power EDA Tools

- Culminate the Process in Stage 5 by integrating the SV generated UPF file with Low Power EDA tools where then continues the journey for UPF post synthesis and backend layouts.

aUPVM - 実装



機能の正確性を備えた電力検証のステップバイステップの自動化:

Stage 1: Python スクリプトによる階層分析

- Pythonベースのツールを採用して設計階層を分析するプロセスを開始し、set_directoryを使用します。
- 抽出された階層は、set_design_top と set_scope の選択をガイドし、電源ドメインの決定に影響を与えます。
- load_upf および save_upf のコード管理構造が組み込まれています。

Stage 2: Python スクリプト内での省電力戦略の実装

- 電源ドメイン、ポート、電源セット、および保持、分離、レベル シフターなどの省電力戦略を実装します。
- 自動化スクリプトは依然として重要であり、設計全体にわたって最適な電源構成を保証します。

Stage 3: SystemVerilog で Python 中間ファイル出力を読み取る

- この段階では、SystemVerilog myUPVM モジュールは UPVM SV ライブラリとインターフェイスし、Python で生成された内部形式の出力ファイルを読み取り、データを収集して myUPVM テンプレート モジュールをカスタマイズします。.
- スコアボードはライブラリ データを比較し、電力の矛盾が存在するかどうかを分析します。

Stage 4: UPF ファイル生成の自動化再開

- 抽出されたデータに基づいてカスタム UPF ファイルを生成します。

Stage 5: 低消費電力EDAツールとの統合

- ステージ 5 では、SV で生成された UPF ファイルを低電力 EDA ツールと統合してプロセスを完了し、その後 UPF ポスト合成とバックエンド レイアウトのプロセスを継続します。

Evidence: aUPVM Query Based Tool

Python Automation Code

```
import os
import os.path
import re
import sys
import traceback
from collections import OrderedDict

def set_path_upf_directory(file_name):
    try:
        with open(file_name,'a') as f:
            path = compatible_input(os.getcwd()+"\n"+" Do you want proceed with this path (yes/no)? ")
            if path == "yes":
                path = os.getcwd()
                f.write("[set_dir]:UPF_DIR "+path)
            if path == "no":
                pa = "/home"
                pa1 =  []
                directory = ""
                print("Enter the Full Path where do you want:"+"\n")
                while True:
                    print("Available Directories in this path: "+"\n"+pa)
```

Executing Query Based Automation

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

Provide the UPF version : 2.0
```

```
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
```

Walking through Directories

```
/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
    pcie_s2p
```

Analyzing SV files & Printing the Design Hierarchy

```
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
```

Listing out hierarchical instance for Power Domain Elements

aUPVM UPF Strategies integration

```
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
```

List out the UPF Files

```
Enter the file Name from the list: pcie_latest1.upf
Available Power Domain in this file
System Verilog Files Found in this Directory

Match Found: The Power Domain Elements Present in this Design
Element in UPF File: pcie_tx
Design Hierarchy: pcie_tx


System Verilog Files Found in this Directory

Match Found: The Power Domain Elements Present in this Design
Element in UPF File: pcie_rx
Design Hierarchy: pcie_rx


System Verilog Files Found in this Directory
System Verilog Files Found in this Directory
System Verilog Files Found in this Directory
The Load UPF File Successful
```

Matching Power Domains with Current Design Hierarchy

```
Enter the file Name from the list: rtl_top.upf
Available Power Domain in this file

WARNING: The Power Domain Elements are in not Matched with the Current Design:

Loading UPF File is Failed
```

Matching Power Domains Failed

aUPVM output file Ipdut

```
[upf_version]:2.0
[top]:pcie_top
[scope]:pcie_top
[pd]:PD_PD_PCIE_TOP;
[pd]:PD_PD_PCIE_PHY_CDR;[elements]:pcie_rx/phy_rx/cdr;
[pd]:PD_PD_PCIE_RX;[elements]:pcie_rx;[supply]:primary;[supply]:backup;
[pd]:PD_PD_PCIE_TX;[elements]:pcie_tx;[supply]:primary;[supply]:backup;
[set_dir]:UPF_DIR /home/gopisrinivas/Workspace/PA/PCIe_latest

[upf_file]:/home/gopisrinivas/Workspace/PA/PCIe/pcie_latest1.upf
[upf_file]:/home/gopisrinivas/Workspace/PA/PCIe/pcie.upf
[upf_file]:/home/gopisrinivas/Workspace/PA/PCIe/pcie_upf1.upf
[upf_file]:/home/gopisrinivas/Workspace/PA/PCIe/pcie_final.upf
[ports]:VDD1;VSS1,VDD2;VSS2
[nets]:RX_Pwr;[domain]:PD_PD_PCIE_RX
[nets]:RX_Gnd;[domain]:PD_PD_PCIE_RX
[nets]:CDR_Pwr;[domain]:PD_PD_PCIE_PHY_CDR
[nets]:CDR_Gnd;[domain]:PD_PD_PCIE_PHY_CDR
[nets]:TX_Pwr;[domain]:PD_PD_PCIE_TX
[connect_net]:RX_Pwr;VDD1,RX_Gnd;VSS1,CDR_Pwr;VDD2,CDR_Gnd;VSS2
[create_supply_sets]:CDR_SS;[function]:power,CDR_Pwr;[function]:ground,CDR_Gnd
[associate_supply_set]:CDR_SS;[handle]:PD_PD_PCIE_PHY_CDR.primary
```

UPVM Package Libraries & Leveraging in myUPVM

UPVM Package Library

```
package upvm_pkg;

class upf_version;
  int fd;
  string line;
  string st;
  string substring;
  int fd_w,fd_a;
  string version[$];

  function string version_fun(string filename,string array[$]);
        //Passing the value for upf version if not passed in lpdut file
        return array.pop_back();
  endfunction
endclass

class set_design_top;
  string design_top;
  int fd;
  string line;
  string tp;
  string top_module;
  int fd_a;
```

Extending UPVM Libraries in myUPVM

```
`include "upvm.sv"
module myUPVM;
import upvm_pkg::*;

  class my_upf_version extends upf_version;
        string array[$];
        function new(string filename,string array[$]);
        //Reading line for set scope
        fd = $fopen(filename,"r");
        fd_w=$fopen("./pcie.upf","w");
        fd_a=$fopen("./pcie.upf","a");

        do begin
                $fgets(line, fd);
                if(line.substr(0,13)=="[upf_version]:")
                begin
                   for(int i = 14; i <=(line.len());i++)
                   begin
                        if(line.getc(i) == " ")
                        begin
                                substring=st;
                        end
                        else
                        begin
                                st={st,line.getc(i)};
                        end
```

Power Strategies Validation & UPF creation

```
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;
```

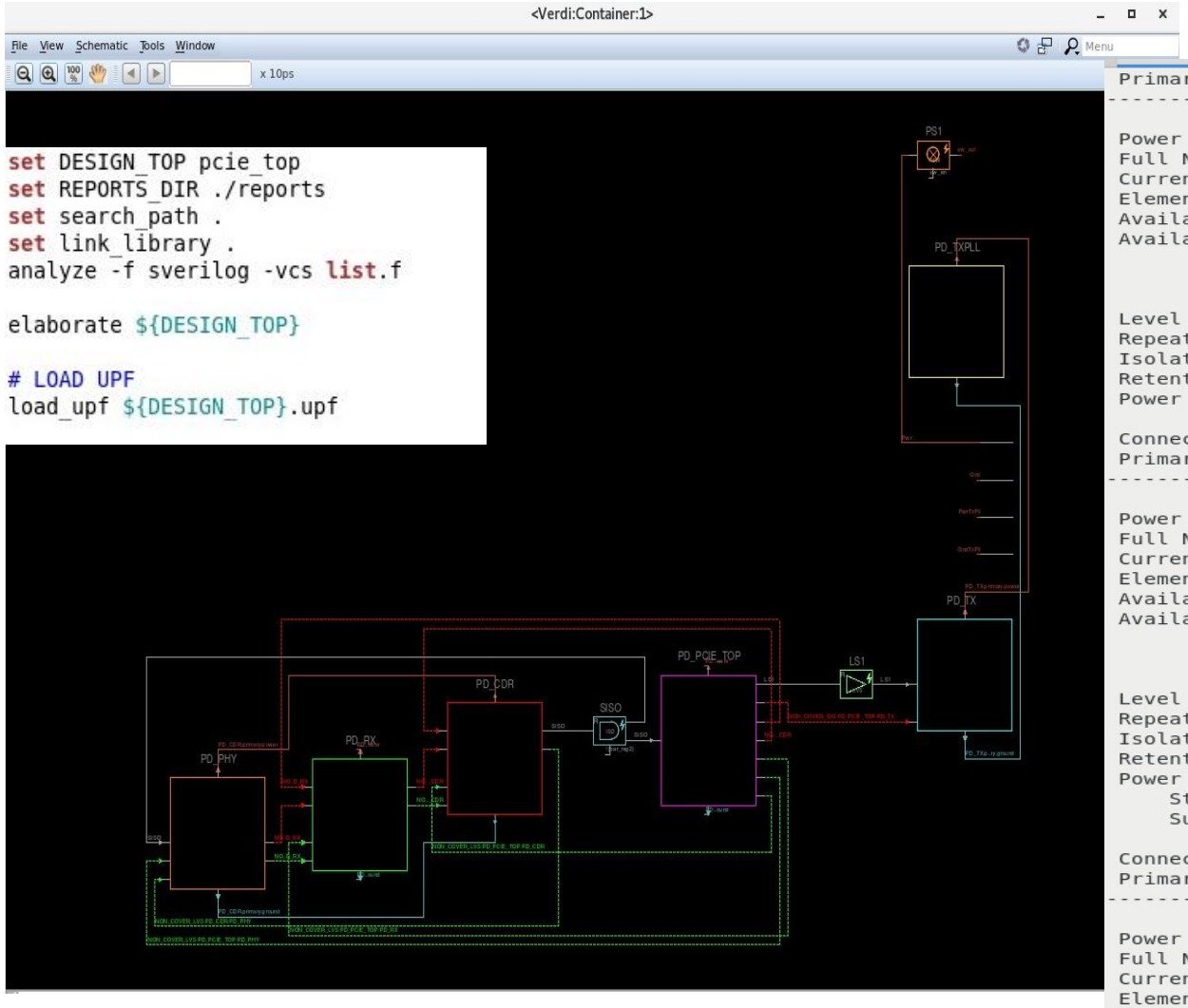
Scoreboard for Power Strategies Validation & UPF file Creation

```
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}
```

Created UPF File

VCLP Result

Integration with Low Power EDA Tool




```
set DESIGN_TOP pcie_top
set REPORTS_DIR ./reports
set search_path .
set link_library .
analyze -f sverilog -vcs list.f

elaborate ${DESIGN_TOP}

# LOAD UPF
load_upf ${DESIGN_TOP}.upf
```

```
Primary Supply Set   : PD_CDR.primary.power          PD_CDR.primary.ground
-------------------------------------------------------------------------

Power Domain         : PD_PCIE_TOP
Full Name            : PD_PCIE_TOP
Current Scope        : pcie_top
Elements             : pcie_top
Available Supply Nets : Gnd1 Pwr1
Available Supply Sets : PD_CDR.primary PD_PCIE_TOP.primary
                        PD_PHY.primary PD_RX.primary PD_TX.primary
                        PD_TXPLL.primary SS_NET1

Level Shifter Strategies :
Repeater Strategies  :
Isolation Strategies :
Retention Strategies :
Power Switch Strategies :

Connections          : -- Power --                   -- Ground --
Primary Supply Set   : PD_PCIE_TOP.primary.power     PD_PCIE_TOP.primary.ground
-------------------------------------------------------------------------

Power Domain         : PD_PHY
Full Name            : PD_PHY
Current Scope        : pcie_top
Elements             : pcie_rx/phy
Available Supply Nets : Gnd Gnd1 Pwr Pwr1 sw_out
Available Supply Sets : PD_CDR.primary PD_PCIE_TOP.primary
                        PD_PHY.primary PD_RX.primary PD_TX.primary
                        PD_TXPLL.primary SS_NET1

Level Shifter Strategies :
Repeater Strategies  :
Isolation Strategies :
Retention Strategies :
Power Switch Strategies :
    Strategy     : PS1
    Supply Nets : Pwr, sw_out

Connections          : -- Power --                   -- Ground --
Primary Supply Set   : PD_CDR.primary.power          PD_CDR.primary.ground
-------------------------------------------------------------------------

Power Domain         : PD_RX
Full Name            : PD_RX
Current Scope        : pcie_top
Elements             : pcie_rx
```

Conclusion (結論)

- **UVM + UPF**

- Industry seems to be following a parallel path with respect to:
- Methodologies based test bench
- Power Architecture, including Unified Power Formats (UPF)
- Incorporating Power Architecture seems to be more like an afterthought post Functional Verification, almost a fourth dimension to our strategy leveraging the test bench architecture.
- Learn UPF
- Validating and Debugging Low Power is truly complex.

- **Advantages of UPVM™**

- OOPS-based (OOPSベース)
- Low Power, now within SV/UVM environment which Power Verification Engineers already know (低消費電力、電力検証エンジニアがすでに知っているSV/UVM環境内)
- Power Verification is UPF-based.
- No need to learn UPF Constructs. (UPF構造を学ぶ必要はありません)
- Verification/Validation at ***early-stage*** whilst verifying the UVM. (UVMを検証しながら初期段階で検証/検証)
- Lower learning curves to learn Low Power tools, since the Library has python scripts to automate **myUPVM**
- UPF generation and integrate with tools for verification (UPFの生成と検証ツールとの統合) at RTL, GLS and GDSII as per IEEE standards.

SUMMARY

- 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 respin's 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. (早い段階で電力検証を導入すると、電力戦略を組み込むための合計時間が短縮され、設計サイクルが大幅に短縮されます。)
- Proposed in-built Power Domain Classes as UPVM Package as Library for Devices, Memories, Bus Interface cores using Power Management Architecture & UPF Constraints may be implemented to include Power Domain in UPVM.

Arigatō Gozaimasu!

- *Silicon Interfaces*™
- info@siliconinterfaces.com
- www.siliconinterfaces.com