2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION TAIWAN

UPF centric agentic tool for UPVM frameworks seamlessly integrated to low power ASICs

Chandini Prudvi, Himanshu Vishwakarma, Sastry Puranapanda
Silicon Interfaces, Mumbai, India

Silicon Interfaces®

accellera SYSTEMS INITIATIVE

Agenda

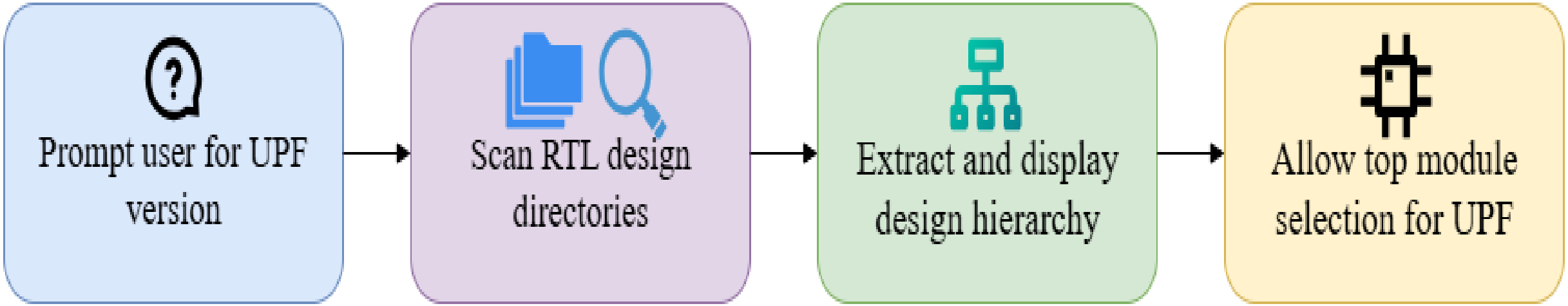
- Abstract
- Introduction
- Automation with **conversational AI-enabled Agentic Tool** aUPVM
- Power Verification automated Workflow ensuring Functional Correctness
- aUPVM – Implementation
- Incorporating UPVM to Verification Environment
- Evidence
- Applications
- Conclusion

Abstract

- aUPVM enables **conversational AI-enabled** Machine Learning of low-power management and strategies from designers with minimal intervention, integrating seamlessly with SystemVerilog-based UPVM™ for early-stage ASIC power validation.
- It bridges design and verification, enabling early power intent validation and efficient low-power signoff.
- The intelligent aUPVM™ tool is built on regenerative use of the low power repository to assist the designer to accelerate power strategies for DUT.

Introduction

- Prior work integrated UPF and UVM within SystemVerilog to address low-power design challenges in semiconductor workflows.
- aUPVM.py builds on this by automating UPF Low Power management/strategies using a Python-based approach.
- The script simplifies UPF creation by:

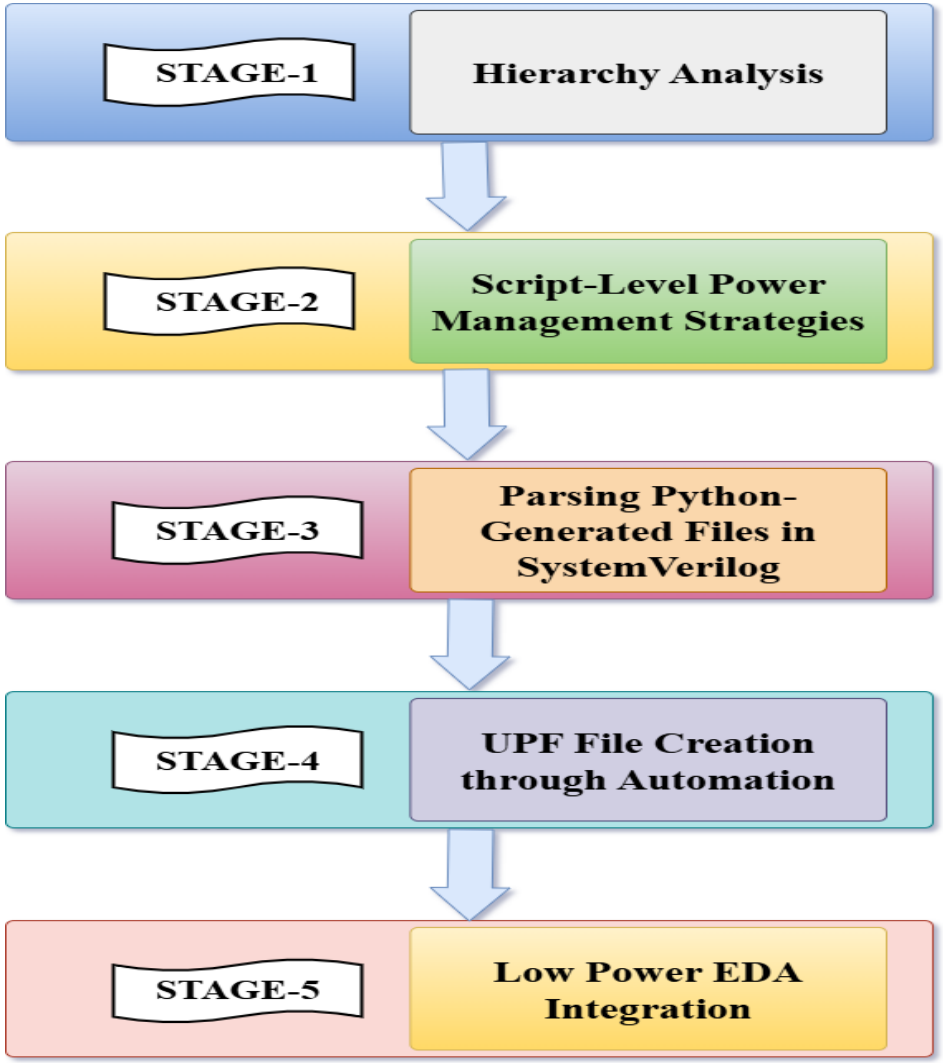


Automation with conversational AI-enabled Agentic Tool aUPVM

- Interactive, conversational tool that collaborates with the Low Power Designer to identify UPF power management strategies.
- Automatically analyzes hierarchy and ML by scanning DUT/Testbench directories for modules and instances.
- Smart guidance system ensures:
  - Elements are within the scope
  - Power domains are assigned for Nets, Switches, Retention, Isolation and Level Shifters.
- Generates an intermediate format file compatible with myUPVM, which auto-generates UPF file.

Power Verification Workflow Ensuring Functional Correctness

1. The tool uses set_directory to analyze the design hierarchy
2. Automates power domains, ports, supply sets, and low-power strategies like retention, isolation, and level shifters.
3. myUPVM modules use the UPVM SV library to read Python data, customize templates, and use a Scoreboard to detect power contradictions.
4. Generate a custom UPF file based on the extracted data.
5. Final step integrates the generated UPF with low-power EDA tools for post-synthesis and layout.



Incorporating UPVM to Verification Environment

- The UPVM Library Package provides a reusable and extensible base for modelling power architecture within the SystemVerilog/UVM environment defines a comprehensive set of SystemVerilog class libraries that model key components of power intent, structured around the power elements.

```
module myUPVM;
import upvm_pkg::*;

  class my_upf_version extends upf_version;
        function new(string filename,string upf_version[$]);
                this.version.push_back("2.0");
                super.version_fun(filename,upf_version);
        endfunction
  endclass
  class my_set_design_top extends set_design_top;
    function new(string filename,string top_module);
        this.design_top="pcie_top";
    super.set_design_top_fun(filename,top_module);
    endfunction

  endclass

  class my_load_upf_c extends load_upf_c;
        function new(string filename);
                super.load_upf_fun(filename);
        endfunction
  endclass

  class my_set_dir_c extends set_dir_c;
        function new(string filename);
                super.set_dir(filename);
        endfunction

  endclass
  class my_hirearchy_c extends hirearchy_c;
        function new();
                super.hi_fun();
        endfunction
  endclass

  class my_associate_supply_set_c extends associate_supply_set_c;
        function new(string filename, string label_create[$],handle_create[$]);
                super.associate_supply_set_fun(filename,label_create,handle_create);
        endfunction
  endclass
  class my_set_scope extends set_scope;
   string array;
    function new(string filename,string module_list[$]);
        this.module_list.push_back("pcie_phy");
      super.set_scope_fun(filename,module_list);
    endfunction
```

Continue…..

- By leveraging System Verilog classes and packages provided within the myUPVM library, the script is able to interpret power management directives from lpdut (low-power Design Under Test) files and convert them into structured, standards-compliant UPF constructs

```
package upvm_pkg;
class upf_version;
  int fd;
  string line;
  string st;
  string substring;
class set_design_top;
  string design_top;
  int fd;
  string line;
class set_scope;
  string module_list[$];
  int fd;
  string line;
class create_power_domain;
  string pd_nam[];
  int fd;
  string line;
class create_supply_port;
  string port_dec[$];
  int fd,count,fd_a;
  string line,tp1;
  string array_port[$];
class create_supply_net;
      string net_dec[$];
      string line,array_net_substring,net_substring;
      string array_net[$],array_net_pd[$];
      int fd,fd_a,net_end,k;
      function string create_supply_net_fun(string filename,output string net_dec[$]);
```

Evidence

- aUPVM.py was validated on a low-power PCIe design, automating UPF generation via UPVM library integration.
- The resulting UPF files are fully compatible with industry-standard EDA tools for GLS and GDSII validation, ensuring consistency and compliance throughout the flow.

```
#-------------------------------------------------------------------------------
#This sets the upf version
#-------------------------------------------------------------------------------
upf_version 2.0

#-------------------------------------------------------------------------------
#This sets the design top level
#-------------------------------------------------------------------------------
set_design_top pcie_tb

set_scope .

#-------------------------------------------------------------------------------
# Loading UPF file
#-------------------------------------------------------------------------------
load_upf /home/chandini/Workspace/UPVM_latest/UVM/pcie1.upf

#-------------------------------------------------------------------------------
# Create power domain
#-------------------------------------------------------------------------------
create_power_domain PD_PCIE_TOP
create_power_domain PD_PCIE_RX -elements { pcie_rx } \
        -supply  {primary} \
        -supply  {backup}
create_power_domain PD_PCIE_TX -elements { pcie_tx } \
        -supply  {primary} \
        -supply  {backup}
create_power_domain PD_PCIE_RX_S2P -elements { pcie_rx/phy_rx/cdr/s2p }
create_power_domain PD_PCIE_RX_PLL -elements { pcie_rx/phy_rx/cdr/pll }
create_power_domain PD_PCIE_TX_P2S -elements { pcie_tx/phy_tx/p2s }
```

```
699 report_power_domain
700  Power Domain         : PD_PCIE_RX
701  Full Name            : PD_PCIE_RX
702  Current Scope        : pcie_top
703  Elements             : pcie_rx
704  Available Supply Nets : GndRx PwrRx
705  Available Supply Sets : PD_PCIE_RX.backup PD_PCIE_RX.primary
706                          PD_PCIE_RX_PLL.primary PD_PCIE_RX_S2P.primary
707                          PD_PCIE_TOP.primary PD_PCIE_TX.backup
708                          PD_PCIE_TX.primary PD_PCIE_TX_P2S.primary
709  Level Shifter Strategies :
710  Repeater Strategies  :
711  Isolation Strategies :
712      Strategy    : RX_ISO
713      Supply Nets : PwrRx, GndRx
714  Retention Strategies :
715  Power Switch Strategies :
716  Connections          : -- Power --                    -- Ground --
717  Primary Supply Set   : PD_PCIE_RX.primary.power       PD_PCIE_RX.primary.ground
718 ------------------------------------------------------------------------
719  Power Domain         : PD_PCIE_RX_PLL
720  Full Name            : PD_PCIE_RX_PLL
721  Current Scope        : pcie_top
722  Elements             : pcie_rx/phy_rx/cdr/pll
723  Available Supply Nets : GndRxPll PwrRxPll sw_out2
724  Available Supply Sets : PD_PCIE_RX.backup PD_PCIE_RX.primary
725                          PD_PCIE_RX_PLL.primary PD_PCIE_RX_S2P.primary
726                          PD_PCIE_TOP.primary PD_PCIE_TX.backup
727                          PD_PCIE_TX.primary PD_PCIE_TX_P2S.primary
728  Level Shifter Strategies :
729      Strategy    : RX_PLL_LS
730      Supply Nets :
731  Repeater Strategies  :
732  Isolation Strategies :
733  Retention Strategies :
734  Power Switch Strategies :
735      Strategy    : PS_RX_PLL
736      Supply Nets : PwrRxPll, sw_out2
737  Connections          : -- Power --                    -- Ground --
738  Primary Supply Set   : PD_PCIE_RX_PLL.primary.power   PD_PCIE_RX_PLL.primary.ground
739 ------------------------------------------------------------------------
740  Power Domain         : PD_PCIE_RX_S2P
741  Full Name            : PD_PCIE_RX_S2P
```

vc_static_shell>

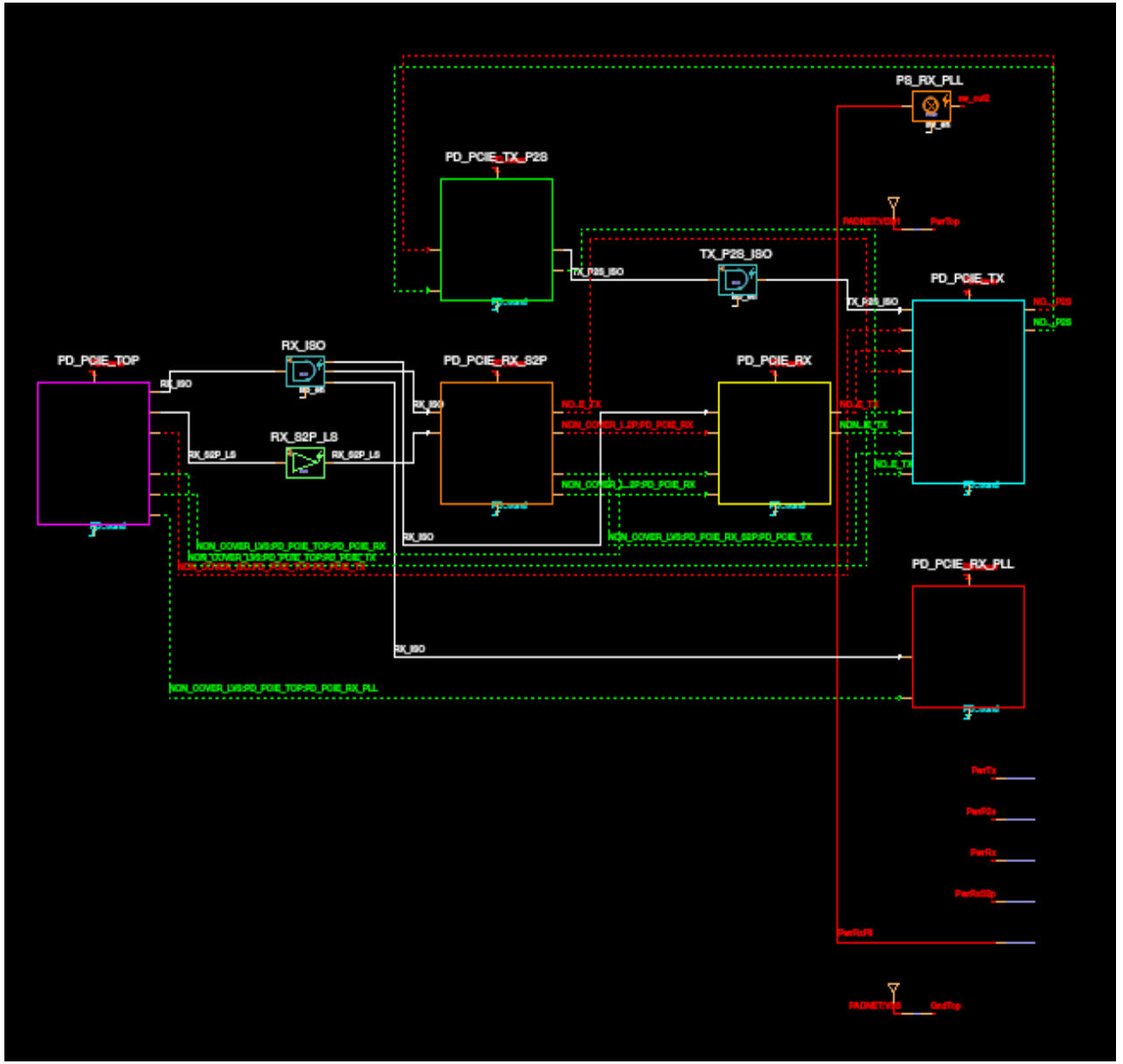
```
930 ------------------------------------------------------------------------
931 report_isolation_strategy
932  Isolation Strategy   : RX_ISO
933  Full Name            : PD_PCIE_RX/RX_ISO
934  Power Domain         : PD_PCIE_RX
935  Force Isolation      :
936  Power Supply Net     : PwrRx
937  Ground Supply Net    : GndRx
938  Elements             :
939  Exclude Elements     :
940  Source Supply Set    :
941  Sink Supply Set      :
942  Clamp Value          : 1
943  Applies To           : inputs
944  Diff Supply Only     : 0
945  Isolation Signal     : iso_en
946  Location             : self
947  Isolation Sense      : high
948  Isolation Lib Cells  :
949 ------------------------------------------------------------------------
950  Isolation Strategy   : TX_P2S_ISO
951  Full Name            : PD_PCIE_TX_P2S/TX_P2S_ISO
952  Power Domain         : PD_PCIE_TX_P2S
953  Force Isolation      :
954  Power Supply Net     : PwrP2s
955  Ground Supply Net    : GndP2s
956  Elements             :
957  Exclude Elements     :
958  Source Supply Set    :
959  Sink Supply Set      :
960  Clamp Value          : 1
961  Applies To           : outputs
962  Diff Supply Only     : 0
963  Isolation Signal     : iso_en
964  Location             : self
965  Isolation Sense      : high
966  Isolation Lib Cells  :
967 ------------------------------------------------------------------------
```

vc_static_shell>

```
760 ------------------------------------------------------------------------
761  Power Domain         : PD_PCIE_TOP
762  Full Name            : PD_PCIE_TOP
763  Current Scope        : pcie_top
764  Elements             : pcie_top
765  Available Supply Nets : GndTop PwrTop
766  Available Supply Sets : PD_PCIE_RX.backup PD_PCIE_RX.primary
767                          PD_PCIE_RX_PLL.primary PD_PCIE_RX_S2P.primary
768                          PD_PCIE_TOP.primary PD_PCIE_TX.backup
769                          PD_PCIE_TX.primary PD_PCIE_TX_P2S.primary
770  Level Shifter Strategies :
771  Repeater Strategies  :
772  Isolation Strategies :
773  Retention Strategies :
774  Power Switch Strategies :
775  Connections          : -- Power --                    -- Ground --
776  Primary Supply Set   : PD_PCIE_TOP.primary.power      PD_PCIE_TOP.primary.ground
777 ------------------------------------------------------------------------
778  Power Domain         : PD_PCIE_TX
779  Full Name            : PD_PCIE_TX
780  Current Scope        : pcie_top
781  Elements             : pcie_tx
782  Available Supply Nets : GndTx PwrTx
783  Available Supply Sets : PD_PCIE_RX.backup PD_PCIE_RX.primary
784                          PD_PCIE_RX_PLL.primary PD_PCIE_RX_S2P.primary
785                          PD_PCIE_TOP.primary PD_PCIE_TX.backup
786                          PD_PCIE_TX.primary PD_PCIE_TX_P2S.primary
787  Level Shifter Strategies :
788  Repeater Strategies  :
789  Isolation Strategies :
790  Retention Strategies :
791  Power Switch Strategies :
792  Connections          : -- Power --                    -- Ground --
793  Primary Supply Set   : PD_PCIE_TX.primary.power       PD_PCIE_TX.primary.ground
794 ------------------------------------------------------------------------
795  Power Domain         : PD_PCIE_TX_P2S
796  Full Name            : PD_PCIE_TX_P2S
797  Current Scope        : pcie_top
798  Elements             : pcie_tx/phy_tx/p2s
```

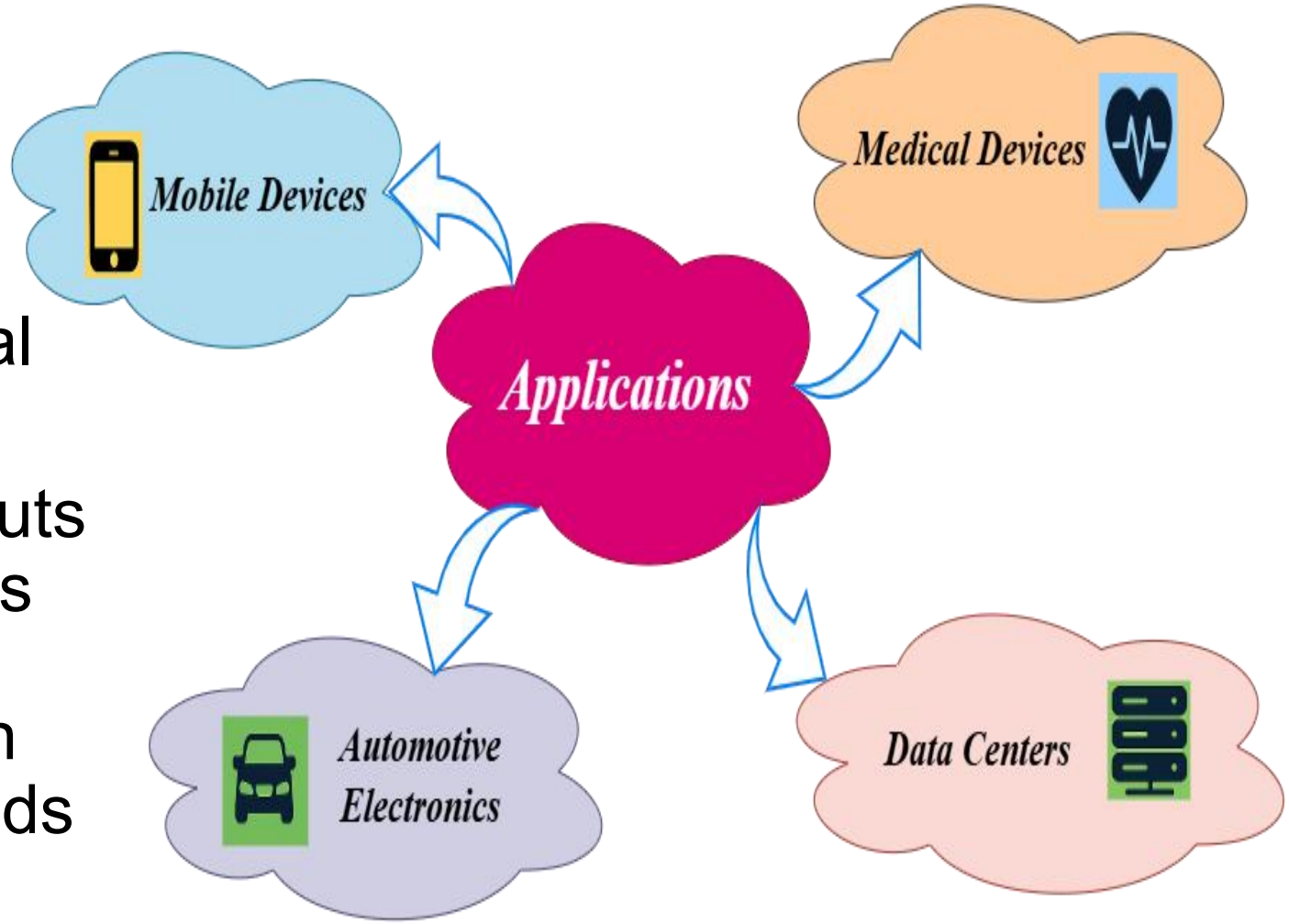
VC Static Console

vc static shell>



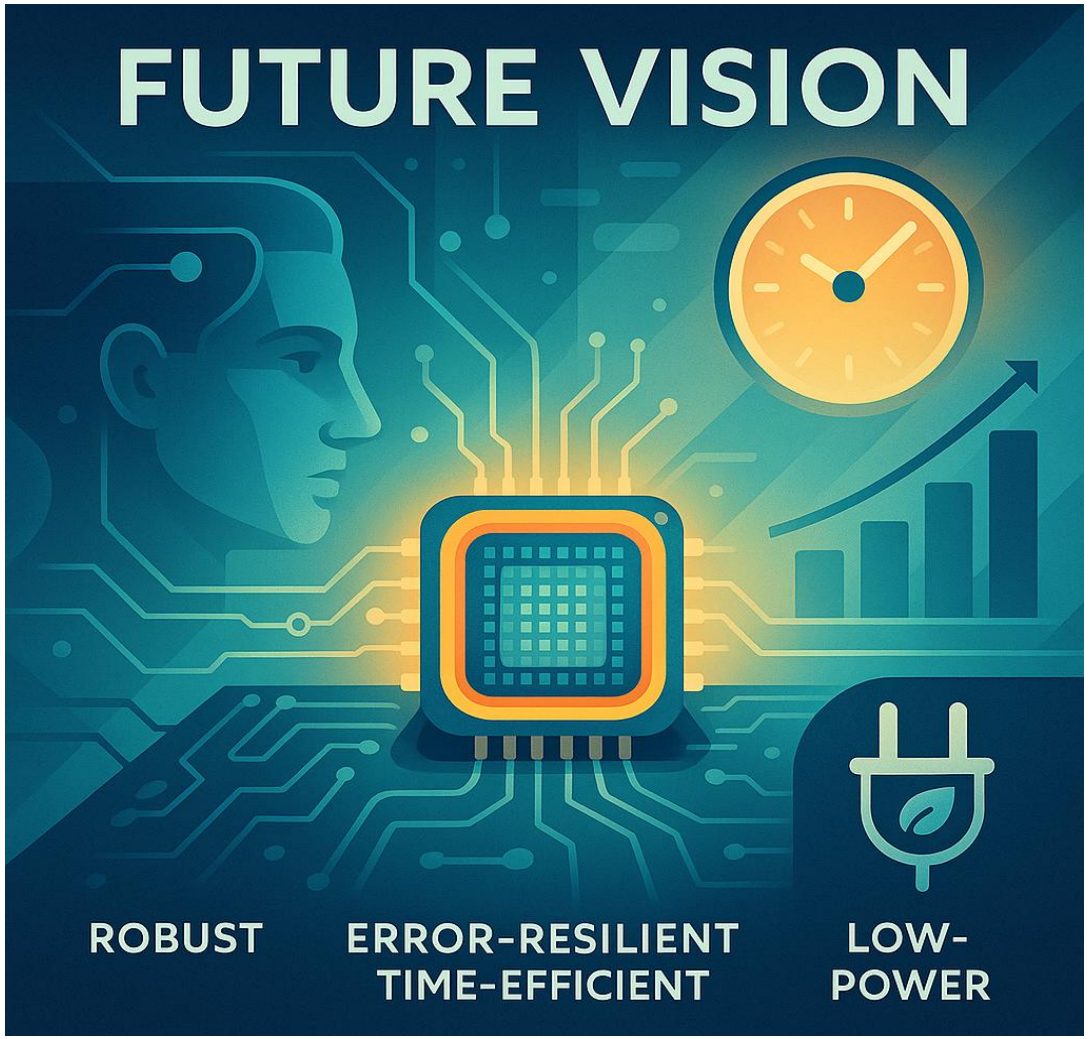
Applications

- UPVM-Based Low-Power Verification offers a strategic advantage by enabling early integration of power intent verification alongside functional verification.
- Early-stage power validation cuts development cost and shortens time-to-market. Embedding power-aware checks early with UPVM reduces risks and speeds up product readiness.



Conclusion

- **Key Achievements**
  - Automates UPF Generation
  - Reduces Manual Effort & Errors
  - Ensures Standards Compliance
- **Impact**
  - Faster Validation
  - Workflow Efficiency



Question & Answer

- Contact Details
  - Support, info@siliconinterfaces.com
  - Heena Bhatia, heena@siliconinterfaces.com
  - http://www.siliconinterfaces.com Agent Betsie™
  - http://www.youtube.com/@SiliconInterfaces
  - http://www.linkedin.com/company/silicon-interfaces
  - http://x.com/silicongo