

PRIYANSHU  
[priyanshud141@gmail.com](mailto:priyanshud141@gmail.com)  
<https://priyanshumishra77.github.io/priyanshu.github.io/>  
<https://www.linkedin.com/in/priyanshu-mishra-68837511b/>  
contact: +91-9039391582

## PROFESSIONAL SUMMARY

- **3+ years** of experience in **DV(CPU Verification)** and **Embedded Hardware domain**.
- Functional verification of **AMBA Protocol: APB, AHB, and AXI protocol**.
- Functional verification of **Serial Protocol: I2C, SPI & USB**
- Good understanding of **Processor architecture, micro-architecture, and digital design, RISC-V ISA**
- Good understanding of **C, Python, RISC Assembly, Verilog & System Verilog**.
- Knowledge of various simulation tools for **Computer Architecture and VLSI**.
- Experience in building a **Verification Environment** from scratch using **System Verilog** and methodologies like **UVM**.
- Proficient in writing **test cases, simulation, and debugging**.
- Hands-on expertise of working on **Test Plan** and test **bench development** in **UVM environment**.
- Good understanding of **ASIC Design Flow/SoC** and **Design Verification Techniques**.
- Basic Knowledge of **GPU programming(CUDA, OpenMP)**, **Hardware accelerators**, and **Neuromorphic computing**.

## PROFESSIONAL EXPERIENCE

**Company:** **SISOC SEMICONDUCTOR**

**Verification Engineer**

**Feb 2024 – Present**

### PROJECT DETAILS:

**Project:** **RISC-V Processor Design and Verification; & exploring x86**

**Description:** Building & leading a team of **10 engineers**. Interviewing, **Mentoring** and assisting them in designing a **high-performance RISC-V CPU** from scratch.

### Roles & Responsibilities:

- Taught(**Graduate Computer Architecture**)- **RISC-V ISA & Micro-architecture** design for RV32I/64I.
- Illustrated them on tools and techniques for **optimization and performance analysis** for a fully functional CPU(examples include:- **Branch prediction techniques, BHSR, Cache design, mapping techniques MSHR, Hardware & software prefetching, neural network and reinforcement learning, loop unrolling, Tomasulo and Amdahl's rule for performance analysis, dynamic scheduling**).
- Various **Open-source tools** and **toolchain** discussion, **LLVM**. Taught **C/RISC-V assembly code**.
- Understood and guided the team to design various **RISC-V Core(DLX, Mor1kx, & CLARVI)**.
- Gave an overview of **UVM for RISC-V CPU Verification**.
- Teaching myself and doing various **projects** on **IA32/IA64(x86 Architecture)** □ **Y86-64 processor design** and **x86 assembly practice** on **NASM tool**.
- Exploring **RISC-V(Packed SIMD/SIMD Vector extension)/ Bit manipulation/Vortex(GPU)**.

**Tools & Languages:** **C/RISC & x86 Assembly, Verilog, and System Verilog, UVM, NASM.**

**PROJECT DETAILS:****Project1: RISC-V Processor Verification**

**Description:** The project involves creating a **RISC-V CPU** from the ground up. It comprises all previously supported and **recently added extensions**, and **privileged and unprivileged specs**.

**Roles & Responsibilities:**

- Understood and wrote tests for RISC-V Organisation and Architecture, RV32bit/64bit, **extensions: (IMAFDC/ZICCLSM/ZA64RS/ZIFENCE/ZICSR)**.
- Understood the specification (**unprivileged and privileged**).
- Wrote the **verification strategy/plan** for the above variants of RISC-V.
- **Encoding** of pseudo-instructions and writing the risc-v instructions **test cases**, debugging and **handling the traps** for hint-instructions, **misaligned addresses/load/store/faults/illegal trap exceptions**.
- Wrote **macros** for different instructions.
- Implementation of the **switching modes (M, S, and U)** of risc-v/various features/enabling the **virtualization (sv32/39/48/57)** and **memory subsystems (cache, MMU, TLB, VM/PM)**.
- Implementation of **CSR listing**.
- Understanding the **hypervisor mode**.
- Understanding of the **risc-v environment, linker script, and toolchain makefile and makefrag**.

**Tools & Languages:** RISC-V GNU toolchain, gdb, SPIKE, Assembly, Gitlab, Jira, AsciiDoc.

**PROJECT DETAILS****Project2: RISC-V Processor Verification**

**Description:** The project involves the **understanding risc-v processor**(instructions/R-I type, S-type/golden reference model etc.) and **micro-architectural** aspects of a **computer architecture**

**Roles & Responsibilities:**

- **RISC-V architecture** expertise (RV32/64IMAFDC)
- Study of **Pipeline, Cache and computer architecture**
- Studied about **Pipelining and Hazards, Cache, and Cache coherence**
- Working on writing **RISC-V assembly level tests** targeting various **architecture/microarchitecture** features
- Working on **test plan, test bench development, and coverage analysis** for processor verification
- Wrote **testcases** for **ALU and load, store instructions**
- Using of **random generators and reference models** for **verification**
- Wrote **testcases** for various **Register-Immediate Integer type Instruction's** like, **ADDI, SLTI, SLTIU, XORI, ORI, ANDI, SLLI, SRLI, SRAI**.

**Tools & Language:** Spike, GNU toolchain, Git, assembly, SystemVerilog

**Project3: RISCV core verification using Google Random Instruction Generator**

**Description:** This project involves the understanding of **Google/RISCV-dv SV/UVM** based **Random Instruction Generator**.

**Roles & Responsibilities:**

- Understanding the existing **SV/UVM generator flow**
- Studying **RISC-V** and **ISA specification**
- Generating and running different **assembly tests**

**Tools & Languages:** **System Verilog**, **Git** and **YAML**, **GCC compiler**, **SPIKE** and **OVPSim**

---

**Project4: CVA6 (Ariane core) Verification**

**Description:** This project involves the study and **verification** of **micro-architectural** details of a **RISCV** processor (**CVA6 core**).

**Roles & Responsibilities:**

- Understanding the **CVA6 architecture** and **microarchitecture**.
- Compiling and running assembly tests on **CVA6** using **Synopsys VCS**.
- Used **Google RISCV-DV random instruction generator** to generate **random assembly tests** and **Spike** for the **simulation** of these **random tests**.

**Tools & Languages:** **System Verilog** and **Synopsys VCS**

---

**Client:** **Intel**

**ASIC Verification Engineer**

**Jun 2022 - Oct 2022**

---

**PROJECT DETAILS:**

---

**Project5: RPLS regression and GNR FE DV**

**Description:** **RPLS regression** and **GNRIO DFD FE DV**.

**Roles & Responsibilities:**

- **Failure monitoring** and creating sheets for error **signatures**, **triage**, and **bucketization**.
- **GNRIO** running **testcases** and compare between different versions.
- Explored different **hierarchy blocks** and **environment**.
- **Assertion failure debug**.
- Study of **IP's**

**Tools & Languages:** **Venus Tool**, **NB Flow Manager** and **Simregress Command**, **Verdi**

---

**Company:** **VGIT**

**Verification Trainee Engineer**

**Jan 2021-Aug 2021**

---

**PROJECT DETAILS**

---

**Project6: Verification of AMBA AXI.**

**Description:**

The **AMBA AXI protocol** is targeted at **high-performance**, **high-frequency system design**. It has **separate address/control and data phases** and **supports for unaligned data transfers** using **byte strobes**, **burst-based transactions** with only **start address issued**, and **support for burst lengths up to 256 beats**

**Roles & Responsibilities:**

- Understanding the **specifications of the block**.
- Developed **UVM Test bench** for **verification**.

- Listing down **features, scenarios**.
- Worked on **code and functional coverages**.
- Verified the **testcases on AWSIZE, AWBURST, AWLEN, WSTRB, WLAST**.

**Tools & Languages:** System Verilog, UVM, Questasim

---

### Project7: Verification of AMBA AHB

---

#### Description:

AHB implements the features required for **high-performance, high-clock frequency systems** including **burst transfers, single-clock edge operation, non-tristate implementation, wide data bus configurations**

#### Roles & Responsibilities:

- Understanding the **specifications of the block**
- **Test plan development**
- Developed **UVM components for AHB Master agent**.
- Listing down features, scenarios.
- Checked all the **Signals and their functionalities**

**Tools & Languages:** System Verilog, UVM, Questasim.

---

### Project8: IP Level Verification of I2C

---

#### Description:

The I2C bus uses two wires: **serial data (SDA) and serial clock (SCL)**. All I2C master and slave devices are connected with only those two wires. Each device can be a transmitter, a receiver, or both. **Some devices are masters – they generate bus clocks and initiate communication on the bus, other devices are slaves and respond to the commands on the bus. To communicate with a specific device, each slave device must have an address that is unique on the bus.** I2C master devices (usually microcontrollers) **don't need an address since no other (slave) device sends commands to the master.**

#### Roles & Responsibilities:

- **Test plan development**
- Analyzing **waveforms**
- Coding **Test bench components including reference models**
- **Verification closure using Functional coverage and code coverage as closing criteria.**
- **Debugged and test issues reported.**

**Tools & Languages:** System Verilog, UVM, Questasim.

---

### Project9: Design and Verification of SPI Controller using System Verilog

---

**Description:** SPI Controller is a **design block** that acts as an **interface between processor and SPI slave**. SPI architecture is based on **one master and multiple slaves**. This controller has 2 interfaces, one is **APB interface** use for configuring the **SPI registers, address, and data**, other is **SPI interface** used for connecting with **SPI slaves**. SPI uses **SCLK, MOSI, MISO, and CS** to connect **master to slave**.

#### Roles & Responsibilities:

- Developing **SPI Controller RTL code** and verify the same using **System Verilog**
- Listing down **design features**
- Setting up **testbench** and **testbench component coding**

- Test plan development and testcase coding

**Tools & Languages:** Mentor QuestaSIM, System Verilog

**Company:** Sakthi CADD Startup  
2020

**Embedded Hardware Design Engineer**

**Jan 2020 – Dec**

#### PROJECT DETAILS

**Project10: Development of a Warehouse logistics of an autonomous vehicle**  
2020

**Jan 2020 – Dec**

**Description:** This vehicle replaces the traditional fork/lift trucks to handle pallets in rack areas

#### **Roles & Responsibilities:**

- Read the **datasheet**
- Designed and developed **BMS IC**
- **Develop battery pack**
- Prepared **schematics, layout and BOM.**

**Tools & Languages:** OrCAD Cadence CIS, Allegro, PI Expert

#### TECHNICAL SKILLS

<b>Protocol Knowledge:</b>	Serial(I2C, SPI, USB), AMBA(APB, AHB & AXI) protocol, basic of TileLink
<b>HDVL:</b>	Verilog, System Verilog
<b>Programming Languages:</b>	C, Python, assembly
<b>Methodology:</b>	UVM
<b>Tools:</b>	Synopsys VCS, Mentor's Questa Sim, Aldec- Riviera, GVIM, VS code, git, Jira, Vivado
<b>Operating System:</b>	Linux, Windows

#### ACHIEVEMENTS & AWARDS

- Certification in **Computer Architecture** by Saylor Academy
- Workshop on **CPU Design using Verilog** by NIT TRICHY
- Course on **Hardware Modeling using Verilog** by NPTEL
- Course on **GPU programming** by Vishal Vinod(IIT Madras), **Hardware for Deep learning** by Adi Teman(Bar-Ilan university, Israel), and **Neuromorphic computing** by Shubham Sahay(IIT Kanpur).
- Workshop on **Machine Learning on Hardware** by NIT TRICHY.
- Workshop in **Embedded Systems and VLSI Design** by IIT Kanpur
- Certification in **FPGA Computing System** by Coursera
- Attended **2nd Workshop on RISC-V and OpenPOWER in HPC** at the ICS International Conference on Supercomputing, 2021
- Under top 10 performers in **SoC Design using OpenPOWER** workshop held by VIT and IBM.
- Attended the **35th international conference on VLSI Design** by VLSI Society India

#### EDUCATION

**B.E. in Electronics and Communication**, passed out in 2015-2019 session with 7.19 CGPA from IES College of Technology, Bhopal, India.