Technical Content

VLSI Domain Skills (Overview of your skillset) (Must have in all resumes at the start)

HDL:	**Verilog**
HVL:	**SystemVerilog**
Verification Methodologies:	Constraint Random Coverage Driven Verification Assertion Based Verification - SVA
TB Methodology:	**UVM**
Protocols:	**AXI, AHB, UART, I2C, SPI**
EDA Tool:	Mentor Graphics - Questasim and Xlinix - ISE, Xilinx - Vivado
Domain:	ASIC/FPGA front-end Design and Verification
Programming Languages:	C [Datatype | Array | Pointers | Memory Allocation | List | Queues and stacks | Data structure, Functions] C++ (Good knowledge of OOPs concept, Class, Inheritance, Polymorphism)
Operating System	Linux
Scripting Languages:	Perl Scripting
Core Skills:	RTL Coding using Synthesizable constructs of Verilog, FSM based design, Simulation, CMOS Fundamentals, Code Coverage, Functional Coverage, Synthesis, Static Timing Analysis, Assertion Based Verification using SystemVerilog Assertions.

College skills

Tools used in Engineering

MATLAB - Digital Signal Processing

Keil MDK or Arduino IDE - Microcontroller Programming

MASM or TASM - Assembly Language Programming (8085 or 8086) etc

LTSpice (Spice Simulator) - Circuit analysis

NS2 - Network Simulator for Wireless communication.

Cadence Virtuoso - Custom IC design and package/PCB design/analysis

Subject Expertise

Digital electronics, Analog electronic, Network analysis, Analog & Digital Communications, Signals & System, Signal Processing, Microprocessors/ Microcontrollers, Computer Organization, Wireless communication

VLSI Design

Digital Electronics :

Combinational & Sequential circuits, FSM, Memories, CMOS implementation, Stick diagram,

STA :

STA Basics, Comparison with DTA, Timing Path and Constraints, Different types of clocks

Clock domain and Variations, Clock Distribution Networks, Fixing timing failure

Verilog Programming :

Data types, Operators, Processes, BA & NBA, Delays in Verilog, begin - end & fork join blocks, looping & branching construct, System tasks & Functions, compiler directives, FSM coding, Synthesis issues, Races in simulation, pipelining RTL & TB Coding,

Advanced Verilog & Code Coverage:

Generate block, Continuous Procedural Assignments, Self-checking testbench, Automatic Tasks

Named Events and Stratified Event Queue, Code Coverage: Statement and branch coverage,

Condition & Expression Coverage, Toggle & FSM Coverage

FPGA :

1) Knowledge in FPGA Architecture i.e. CLBs, SWITCH MATRIX, I/O Blocks.
2) Knowledge in FPGA Design flow in Xilinx- ISE i.e. from RTL to .bit file generation.
3) Knowledge in FPGA Internal resources i.e.Memories, Dedicated MUX, SRL16 Shift register, DCM blocks.
4) Have worked on understanding Technology Schematics for RTL codes for SPARTAN 3 series.
5) Have basic ideas on other families like Kintex, Artix, Virtex.

FPGA Design Tool Skill-sets

1) Have worked on Xilinx-ISE 14.7 and have sound knowledge on converting RTL code to .bit file and download the same to the FPGA kit.
2) Have worked on developing Testbench using Xilinx ISE.

DFT Skills

Experience with multiple aspects of the following

ATPG, Test Coverage
JTAG, BSDL, IJTAG
Memory and Logic BIST
Synthesis scan stitching
Memory BIST implementation
Scan/Jtag/boundary-scan insertion and ATPG pattern generation
Test coverage and fault coverage analysis
Knowledge/experience with ATPG / DFT tools: Tessent, System Verilog, RTL

Verification

System Verilog HVL (Can be used if, no verification project is completed)

Memories - Dynamic array, Queue, Associative array, Task & Function - Pass by reference

Interface - Modport and clocking block

Basic and advanced object-oriented programming - Handle assignments, Copying the object contents, Inheritance, polymorphism, static properties and methods, virtual classes and parameterized classes.

Constraint Randomization - constraint overriding and inheritance, Distribution and conditional constraints, Soft, static and inline constraints.

Thread synchronization techniques - events, semaphores and Mailbox - built-in methods

Functional coverage - Cover groups, bins and cross-coverage, CRCDV and regression testing

System Verilog HVL

Interface and clocking block, Inheritance and Polymorphism, Constraint randomization - Inline, distribution, conditional, soft and static constraints. Mailbox and semaphores, Functional coverage, CRCDV and regression testing.

System Verilog Assertions

Types of assertions, assertion building blocks, sequences with edge definitions and logical relationship. Sequences with different timing relationships. clock definitions, implication and repetition operators, different sequence compositions, inline and binding assertions, advanced SVA Features and assertion Coverage

UVM

UVM Objects & Components
UVM Factory & overriding methods
Stimulus Modelling
UVM Phases
UVM Configuration
TLM
UVM Sequence, virtual sequence & sequencer
Introduction to RAL

VLSI Projects

[1] AHB2APB Bridge IP Core Design

HDL: Verilog

EDA Tool: ISE-Xilinx

Description: The AHB to APB bridge is designed as an AHB slave which converts AHB transactions to APB transactions by implementing pipelining at the AHB slave interface. Thus, the bridge supports AHB burst transfers.

Responsibilities:

- Architected the block level structure for the bridge.
- Developed Verilog RTL for each block.
- Verified each block with different transfers like single READ, WRITE & Burst READ, WRITE.
- Synthesized the design.
- Generated code coverage report for RTL Design signoff.

[2] AHB2APB Bridge IP Core Verification

HVL: System Verilog

TB Methodology: UVM

EDA Tool: Riviera Pro – Aldec

Description: The AHB to APB bridge is an AHB slave which works as an interface between the high speed AHB and the low performance APB buses.

Responsibilities:

- Architected the class based verification environment in UVM.
- Defined Verification Plan
- Verified the RTL module with UVM Test Bench with different test scenarios like single READ, WRITE & Burst READ, WRITE with different burst lengths.

- Generated functional and code coverage for the RTL verification sign-off.

[3] AXI Master UVC - AMBA AXI4 Protocol Verification

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The AMBA AXI protocol is targeted at high-performance, high-frequency system and includes a number of features that make it suitable for a high-speed submicron interconnects.

AXI Master UVC is a configurable UVM based verification IP. It verifies the AXI protocol and generates the required functional coverage

Responsibilities:

- Architected the class based verification environment in UVM.
- Defined Verification Plan
- Developed an environment with single master agent which can be configured later.
- Verified the UVC by connecting our mater agent with a slave agent developed by different team.
- Verified the Master UVC's functionality with different types of transfers and bursts.
- Generated functional coverage for verification sign-off.

[4] AXI Slave UVC - AMBA AXI4 Protocol Verification

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The AMBA AXI protocol is targeted at high-performance, high-frequency system and includes a number of features that make it suitable for a high-speed submicron interconnects.

AXI Master UVC is a configurable UVM based verification IP. It verifies the AXI protocol and generates the required functional coverage

Responsibilities:

- Architected the class based verification environment in UVM.
- Defined Verification Plan
- Developed an environment with single slave agent which can be configured later.
- Verified the UVC by connecting our slave agent with a master agent developed by different team.
- Verified the Slave UVC's functionality with different types of transfers and bursts.
- Generated functional coverage for verification sign-off.

[5] AHB Master UVC – AMBA2 AHB Protocol Verification

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The AMBA2 AHB protocol addresses the requirement of high-performance system back bone bus, which supports burst operation, split transaction.

AHB Master UVC is a configurable UVM based verification IP. It verifies the AHB protocol and generates the required functional coverage

Responsibilities:

- Architected the class-based verification environment in UVM.
- Defined Verification Plan
- Developed an environment with single master agent which can be configured later.
- Verified the UVC by connecting our mater agent with a slave agent developed by different team.
- Verified the Master UVC's functionality with different types of transfers and bursts.
- Generated functional coverage for verification sign-off.

[6] AHB slave UVC – AMBA2 AHB Protocol Verification

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The AMBA2 AHB protocol addresses the requirement of high-performance system back bone bus, which supports burst operation, split transaction.

AHB slave UVC is a configurable UVM based verification IP. It verifies the AHB protocol and generates the required functional coverage

Responsibilities:

- Architected the class-based verification environment in UVM.
- Defined Verification Plan
- Developed an environment with single slave agent which can be configured later.
- Verified the UVC by connecting our slave agent with a master agent developed by different team.
- Verified the Master UVC's functionality with different types of transfers and bursts.
- Generated functional coverage for verification sign-off.

[7] SPI Controller Core - Verification

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The SPI IP core provides serial communication capabilities with external device of variable length of transfer word. This core can be configured to connect with 32 slaves.

Responsibilities:

- Architected the class based verification environment in UVM

- Defined Verification Plan
- Verified the RTL module using SystemVerilog
- Generated functional and code coverage for the RTL verification sign-off

[8] UART- IP Core – Verification

HVL : SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: The UART IP core provides serial communication capabilities, which allow communication with modem or other external devices. UART will operate in three different modes – Simplex mode, Full Duplex mode and loopback mode.

Responsibilities:

- Architected the class based verification environment in UVM
- Defined Verification Plan
- Verified the RTL module using SystemVerilog
- Generated functional and code coverage for the RTL verification sign-off

[9] GPIO – Verification

HVL : SystemVerilog

TB Methodology: UVM

EDA Tools: Riviera Pro - Aldec

Description: General purpose I/Os used in SoC

Responsibilities:

- Architected the class based verification environment in UVM
- Verified the RTL module using SystemVerilog

- Generated functional and code coverage for the RTL verification sign-off

Curriculum Project

Router 1x3 – RTL design and Verification

HDL: Verilog

HVL: SystemVerilog

TB Methodology: UVM

EDA Tools: Questasim and ISE

Description: The router accepts data packets on a single 8-bit port and routes them to one of the three output channels, channel0, channel1 and channel2.

Responsibilities:

- Architected the block level structure for the design
- Implemented RTL using Verilog HDL.
- Architected the class based verification environment using SystemVerilog
- Verified the RTL model using SystemVerilog.
- Generated functional and code coverage for the RTL verification sign-off
- Synthesized the design.

Alarm clock project - RTL design

HDL: Verilog-2001

EDA Tools: Modelsim, Quartus-prime

Description: The digital alarm clock displays time in LCD format in a 24hr format.

Responsibilities:

- Developed the RTL code for each of the sub-blocks used in the block level architecture of the Alarm clock.
- Verified each sub-block using task-based Verilog Testbench.
- Finally verified the top-level RTL using Verilog Testbench.
- Generated code coverage for the RTL design sign-off.

RISC - V Design:

The RV32I Processor is designed to support all RV32I Base Integer Instructions (Total -39). It's a three stage pipelined processor which executes 32 bit instructions in program order.

Pipelined Stage I - The instructions are fetched from memory.
Pipelined Stage II - The instructions are decoded and the control signals for all units are generated. Branches, jumps and stores are executed in advance in this stage
Pipelined Stage III - Executes complete instruction and writes back the results in the register file

Responsibilities for Design:

- Developed RTL codes for all modules of RISC V processor.
- Verified individual modules with Linear TB code
- The top module is synthesised and simulated

RISC-V Verification

The RV32I Processor is designed to support all RV32I Base Integer Instructions (Total -39). It's a three stage pipelined processor which executes 32 bit instructions in program order.
Pipelined Stage I - The instructions are fetched from memory.
Pipelined Stage II - The instructions are decoded and the control signals for all units are generated. Branches, jumps and stores are executed in advance in this stage
Pipelined Stage III - Executes complete instruction and writes back the results in the register file

Responsibilities for Verification:

- Developed a class based TB using UVM methodology.
- Verified all the different types of instructions by developing multiple testcases.

- Developed the coverage models and signed-off the verification by achieving 100% coverage.