**(Question 1) Create and document your exhaustive verification plan.**

**Simplified Verification Plan for Simple Interrupt Controller**

**Objective:**

Ensure the Simple Interrupt Controller functions as expected using a Verilog testbench with various test scenarios.

**Scope:**

• **Interrupt Generation:** Validate correct interrupt generation.

• **Interrupt Acknowledgment:** Ensure proper acknowledgment of interrupts.

• **Reset Functionality:** Verify reset and initialization.

• **Multiple Interrupts:** Handle multiple interrupts correctly.

• **Bus Interface:** Test WISHBONE read/write operations.

**Testbench Components:**

• **DUT (Device Under Test):** Simple Interrupt Controller (simple\_pic).

• **Clock and Reset Generator:** Simulates clock and reset signals.

• **Stimulus Generator:** Provides input test vectors.

• **Monitor and Checker:** Observes outputs and checks correctness.

**Test Scenarios:**

• **Reset Test:** Apply reset; ensure all outputs are in default state.

• **Single Interrupt:** Trigger one interrupt and check acknowledgment.

• **Multiple Interrupts:** Test handling of multiple interrupts.

• **No Interrupts:** Ensure no interrupts without triggers.

• **Spurious Acknowledge:** Test invalid acknowledgment handling.

• **WISHBONE Read/Write:** Validate correct data read and write operations.

• **Edge Case:** Simulate continuous interrupts; verify stability.

• **Interrupt Masking:** Test interrupt masking and prioritization.

• **Interrupt Clear:** Ensure clearing and resetting interrupts work as expected.

• **Overrun Test:** Handle multiple interrupts before acknowledgment.

**Coverage Metrics:**

• **Functional Coverage:** Ensure all test scenarios are covered.

• **Code Coverage:** Verify line, branch, and FSM coverage.

• **Assertions Coverage:** Ensure all assertions are executed.

**Pass/Fail Criteria:**

• All test cases should pass without failures.

• No crashes or unexpected behavior.

**Tools:**

• **Simulator:** Aldec Riviera-PRO(EDA Playground)

• **Language:** Verilog.

**Waveform:**

