**Assignment 1**

**Task: Boundary Scan Test Implementation**

**Objective:** Apply boundary scan test on hardware.

**Answer:**

**Steps:**

**Create Test Vector for Boundary Scan:**

    -Identify the pins and the sequence of operations to be tested.

    -Develop the test vector that sets and reads these pins.

     Test Vector:

*yaml*

 TDI: 10101010

TMS: 1100

TCK: 1111

Expected Output: 01010101

**Apply Vector and Check Output Pins:**

       Use a boundary scan tool or JTAG tester to apply the test vector.

       Capture the output from the boundary scan cells.

*sh*

openocd -f interface/jtag.cfg -f target/target.cfg

jtag scan\_chain

jtag shift ir 10101010

jtag shift dr 1100

**Summarize Test Outcome and Identify Errors:**

          Compare the captured output with the expected output.

           Document any discrepancies or errors.

*yaml*

Test Vector Applied: TDI=10101010, TMS=1100, TCK=1111

Expected Output: 01010101

Actual Output: 01000101

Errors Identified: Mismatch at bit positions 2 and 6.

1. Understand Boundary Scan Testing

Boundary scan testing (also known as JTAG testing) is a technique used to test interconnects (connections between integrated circuits) on a PCB (Printed Circuit Board). It allows you to test chips even if they are not physically accessible by using dedicated test logic built into integrated circuits.

2. Preparation

Before starting, ensure you have the following:

* Hardware with JTAG/Boundary Scan Support: Verify that the hardware you are testing supports boundary scan testing via JTAG (Joint Test Action Group).
* Boundary Scan Test Tool: Use a boundary scan test tool or software that supports creating test vectors and analyzing results. Common tools include XJTAG, JTAG Technologies, or vendor-specific tools.
* Hardware Documentation: Refer to the hardware datasheets, schematics, and any specific JTAG instructions provided by the hardware manufacturer.

**3. Steps to Implement Boundary Scan Testing**

**Step 1: Create Test Vector for Boundary Scan**

* **Identify Boundary Scan Cells:** These are dedicated registers on each pin of the JTAG-compatible components.
* **Develop Test Patterns:** Create a test vector that will set and read values on these boundary scan cells. This involves defining the sequence of values to be applied and the expected results.

**Step 2: Apply Test Vector**

* **Connect Boundary Scan Tool:** Connect your boundary scan test tool to the JTAG interface of the hardware.
* **Load Test Vector:** Upload the test vector (test patterns) into the boundary scan tool.

**Execute Test:** Apply the test vector to the hardware. This involves shifting the test patterns through the boundary scan chain and observing the responses.

**Step 3: Check Output Pins**

* **Monitor Output Responses:** Observe the responses from the output pins during the application of the test vector.
* **Compare Expected vs. Actual Results:** Compare the observed outputs with the expected results defined in your test vector.

**Step 4: Summarize Test Outcome**

* **Record Results:** Document the results of the boundary scan test. Note any discrepancies or errors observed during testing.
* **Identify Errors:** If discrepancies or errors are detected, investigate further to identify the root cause (e.g., incorrect connections, faulty components).

**4. Test Outcome Analysis**

* **Review Logs and Reports:** Use the logs and reports generated by the boundary scan tool to analyze the test outcomes in detail.
* **Troubleshoot Issues:** Address any identified errors or discrepancies. This may involve re-checking connections, reviewing hardware design, or validating software configurations.

**5. Documentation**

* **Document Test Results:** Record the test results, including any issues encountered, actions taken, and resolutions applied.

**Update Test Procedures:** Update test procedures and documentation based on the findings to improve future boundary scan testing processes.

**Additional Tips**

* **Repeat Testing:** Perform boundary scan testing multiple times to ensure consistency and reliability of results.
* **Collaborate:** Work closely with hardware engineers and developers to understand the hardware design and facilitate troubleshooting.