Topic: Design For Testability (DFT).

Aimed for: Verification *Author: Hassan TaqiEddin*

Objective

To integrate Design for Testability techniques into high-level design, enabling efficient validation, debugging, and showcasing robustness, coverage and ease of testing and verification.

Overview of Workflow

1. **Preparation and Analysis**
    - Understand new design requirements and identify critical states for verification.
    - Locate potential bugs regions.
2. **Design for testability strategies**
    - Add monitors to enhance observability of sequential elements.
    - Implement Built-In Self-Test (BIST).
    - Insert test points for easier debug access.
3. **Test Plan Development**
    - Create detailed test plans covering functional tests, fault coverage, and corner-case scenarios.
    - Automate the generation of test patterns.
4. **Validation and Debugging**
    - Run simulation to validate DFT logic and measure coverage.
    - Debug issues using reports from monitors, BIST modules, and test points.
5. **Documentation**
    - Document DFT strategies, metrics, and results.

Note: You can use System Verilog for functional coverage, randomization, and automated pattern generation to enhance verification efforts.

Implementation Example

ALU BIST (Built-in Self-Test)

- **Implementation**:

    1. Add pseudo-random pattern generators in your design.
    2. Include error detection mechanism for output compression (Assertions or conditions).
    3. Validate BIST logic through simulation.

Memory Access Path Testing

- **Implementation**:
    1. Identify crucial signals in memory access paths.
    2. Add monitors at critical places.
    3. Validate memory access path testability through dedicated simulation scenarios.

Proposed Output

- Processor designs with:
  - Integrated monitors for sequential element observability.
  - BIST modules for automated testing.
  - Test points for critical signals.
- Validation reports including fault coverage statistics, test results, and debugging metrics.

Benefits

- ✓ Detect faults early during the design cycle.
- ✓ Simplify debugging by leveraging enhanced observability.
- ✓ Reduce design errors and minimize rework iterations.
- ✓ Improve design reliability and scalability.
- ✓ Achieve higher-quality designs through robust testing methodologies.

Conclusion

Integrating DFT in processor design ensures quality, reliability, and testability; by embedding testability in architectural and RTL stages, you can efficiently address challenges and enhance your competitive edge in your design.

!! Important note: Most of DFT techniques affects the design performance, use them only for verification and early implementations, some of them are non-synthesizable which can't be validated on FPGAs.