What Is GLS?

- GLS simulates my digital design after it's converted into a netlist of gates and connections.
- It works on the actual gate implementation, not just high-level code.
- Timing delays from gates and wires are included for realism.

Why I Use GLS

- It helps me find issues that higher-level simulations miss, like glitches and race conditions.
- I verify if the synthesized logic functions correctly.
- It checks design-for-test features and potential integration problems.

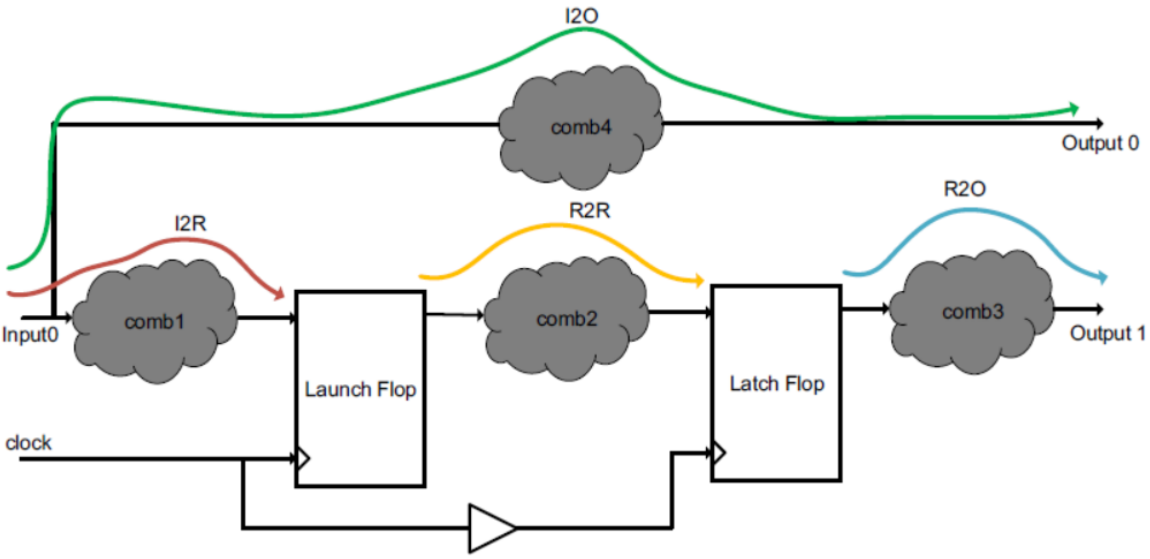
How I Perform GLS

- I load the gate-level netlist into a simulator.
- I apply test inputs and observe outputs to ensure correct behavior.
- I include timing delay files so signal timings are realistic.
- I watch signal waveforms to identify any unexpected behavior.

Considerations for GLS

- GLS takes more time and memory than RTL simulation.
- It’s often the last simulation step before tapeout.
- It helps catch timing-dependent bugs before fabrication.

What Is STA?

- STA analyzes the timing of digital circuits without needing input test patterns.
- It checks every possible signal path for timing violations using mathematical models.
- The goal is to confirm signals arrive and settle within timing constraints.
- 

Setup and Hold Checks

- Setup time: data must be stable before the clock edge for correct capture.
- Hold time: data must remain stable after the clock edge to avoid errors.
- STA verifies these conditions on all sequential elements.

Understanding Slack

- Slack is the margin between required and actual signal arrival times.
- Positive slack means timing is met with safety margin.
- Negative slack indicates timing violations that require fixing.

Clock Definitions

- I define clocks with frequency, duty cycles, and uncertainty for accurate analysis.
- Clock skew and jitter are important, especially for multiple clock domains.
- Proper clock setup directs the entire timing verification.

Path-Based Timing Analysis

- STA inspects all paths from one register to another through combinational logic.
- The critical path is the slowest path, limiting the maximum clock speed.
- STA reports identify these paths for optimization focus.

Performing STA with OpenSTA

- I import netlists, libraries, and timing constraints into the tool.
- I run setup and hold analysis checking every timing path.
- The tool generates slack tables and violation reports.
- Timing closure is achieved when all paths have positive slack.

Why STA Is Essential

- STA is faster than dynamic simulation and covers all paths comprehensively.
- It scales well for very large and complex designs.
- It ensures reliable operation at the target clock frequency.

Summary of What I Learned

- GLS simulates real gate behavior with timing delays to validate logic correctness after synthesis.
- STA statically checks timing for every path, ensuring setup and hold constraints are met without running simulations.
- Together, GLS and STA give me deep confidence that my design works correctly and meets timing requirements before fabrication.
- Understanding and applying these is crucial for successful chip development and avoiding costly errors late in the process.

Thank you for VSD for this program I learned a lot Thanks to VSD Team , and IIT Gandhinagar for the RSIC V Program…… Dr Raheem from MJCET Hyderabad.