WP272

1. **Global Reset Isn't Timing-Critical** – Global resets are slow, unsynchronized, and can introduce timing issues.
2. **Does It Really Matter?** – While often harmless, global resets can cause failures in feedback-driven circuits.
3. **Automatic Coverage of the 99.99% of Cases** – Xilinx FPGAs automatically initialize registers, reducing the need for global resets.
4. **Strategy for the 0.01% of Cases** – Use local, synchronous resets only where precise control is necessary.
5. **Reset Costs More Than You Think!** – Global resets increase resource usage, degrade performance, and limit FPGA optimization.

WP275

1. **Single-Level Logic** – Keeping logic within a single LUT ensures minimal resource usage and maximum performance.
2. **Two-Level Logic** – Functions exceeding four inputs require multiple LUTs, increasing size and delay.
3. **Adding a Reset** – Avoid unnecessary global resets since FPGAs start in a known state after configuration.
4. **Adding More Controls** – Use dedicated flip-flop control inputs efficiently to minimize extra logic.
5. **Do Not Mix Asynchronous and Synchronous Resets** – Mixing reset types forces inefficient synthesis and larger designs.
6. **Synchronous Design** – Using synchronous resets improves performance and avoids timing issues.
7. **Get Your Priorities Right** – Align control signal priorities with FPGA flip-flop behavior for optimal synthesis.
8. **Writing Sympathetic Code** – Code should naturally match FPGA architecture to reduce logic overhead.
9. **The Challenge** – Review and refine HDL code to avoid unnecessary resets and misprioritized controls.