WP272

The main topic of this article is why global reset should not be used in the rtl designs and recommends local reset usage instead of global reset.

In most cases, global resets are not timing critical. The reason for this, often global resets are much slower frequecny than clock frequecny. This gives enough time for all flip flop to reset in one clock period.

Howerer, at higher clock frequecny, global reset signal might not able to reach all flip flop, and this causes timing problems. Additionally, some flip flop can be even metastable. And timing issues can occur when the design uses feedback.

Costs:

Routing Resource Usage:

The global reset signal uses a large routing resource to reach all flip-flops.

This consumes resources that could be used for routing other signals.

High fan-out networks can cause delays.

Logic Resource Consumption:

Most flip-flops in FPGAs have reset inputs. However, when global reset is used, additional reset logic is required. This can lead to unnecessary gate consumption, increasing design space.

Performance Loss:

Using extra routing and logic can slow down the system clock speed.

Unoptimized routing due to global reset can cause critical paths in the FPGA to become longer.

Inability to Take Advantage of Special FPGA Features:

Features such as SRL16E used in Xilinx FPGAs cannot operate at full efficiency when a global reset signal is used.

SRL16E efficiently creates a shift register by storing 16 flip-flops in a LUT. However, since these flip-flops do not support reset, the reset requirement limits the use of SRL16E.

What should do instead ?

A more efficient solution is to ensure that only critical components are reset in the design.

Reset Only Necessary Flip-Flops:

If a module needs a specific initial state, only the flip-flops in that module should be reset.

Use Synchronous Reset:

Synchronizing the reset signal to the clock edge prevents metastability.

Use Local Reset Networks:

Reduce delays by creating a small reset network.

Avoid high fan-out (reaching too many components) reset signals.

Controlled Reset Release:

When the reset is released, the flip-flops should be released according to a specific clock pulse.

The reset output can be synchronized by passing it through a register.

WP275

The most important thing is to know flip-flop architecture in FPGA. In flip-flop architecture, Reset signal is highest priority, then Set signal is next highest priority, Clock Enable signal is lowest priority. When designing, the code must follow this control priorities for less area occupying.

Use Single-Level Logic:
4-input LUTs operate at a single level, saving resources and latency.

Avoid Two-Level Logic:
Operations requiring more than 4 inputs use additional LUTs and increase latency.

Avoid Global Reset:
FPGAs start in a known state at startup; unnecessary global resets degrade performance.

Set Control Priorities Correctly:
Incorrectly sequenced enable, reset, and set signals can overwhelm the design by using additional LUTs.

Use Synchronous Resets:
Asynchronous resets can degrade performance by adding unnecessary logic levels.