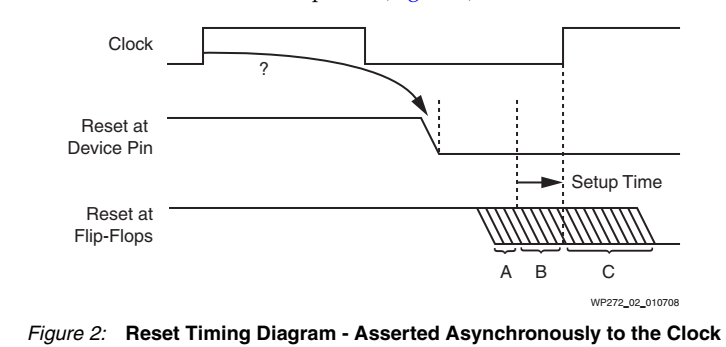
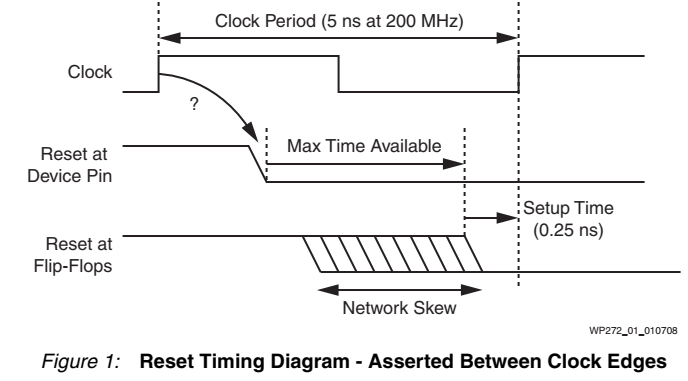
Summary of WP272

The typical drivers of a global reset signal:  
• Press switch: Definitely slow and very undefined timing.  
• Power supply status output: Active for a long period until supply stable.  
• Microprocessor: Pulse tends to be long.

While such global resets are perceived as non-critical due to slow timing, their release timing becomes critical as clock rates increase. Asynchronous or skewed reset de-assertion can cause flip-flops to activate on different clock edges, risking metastability or functional failures.

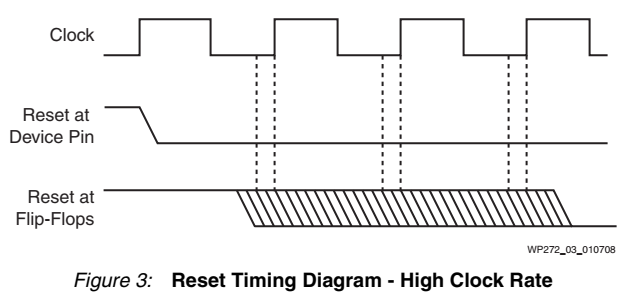
For example:



In Figure 1, the reset signal is de-asserted between clock edges and then spreads to various flip-flops. Each flip-flop needs the reset signal to be de-asserted before the active clock edge, with enough setup time. As the clock speed increases, the time available to distribute the reset signal decreases. Since the reset signal has a high fan-out, meeting timing requirements is challenging.

In Figure 2, If the reset is released asynchronously (which is common), all flip-flops may not be released on the same clock edge, even if the distribution time is shorter than a clock period.

Flip-flops receiving the reset release at point A will activate on the first clock edge, while those at point C will activate on the next clock edge. Flip-flops at point B may behave unpredictably and could even cause metastability.

In Figure 3, With higher clock speeds and distribution delays in large devices, it becomes difficult to ensure all flip-flops are released in sync with the same clock edge.

In High Fan-Out Networks, Reset signals in large FPGAs face distribution skew, making it difficult to meet setup times, especially in high-speed designs.

99.99% Cases: In pipelined/dataflow circuits (e.g., FIR filters), reset timing is less critical as invalid data flushes out naturally.

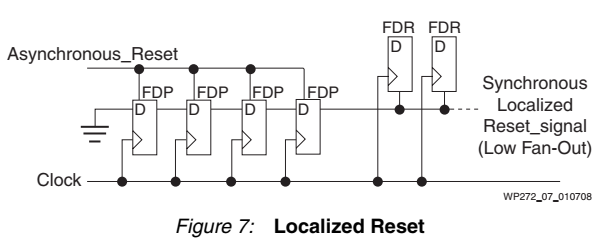
0.01% Critical Cases: Circuits with feedback paths (e.g., state machines, IIR filters) or synchronized operations (e.g., one-hot state machines) require precise reset release. A single flip-flop releasing early can corrupt states or cause instability. If you have ever had one of the circuits that doesn’t work the first time, then maybe you have encountered one of the 0.01% cases and have been unlucky enough to have released the reset at the wrong time.

Global Reset Cost:

While implementing a design, the costs of a global reset in HDL code can be overlooked. However, the cost can be significant:

* Routing resources of the device are used.
* Reduces freedom for other connections.
* May lower system performance potentially requiring a higher device speed grade.
* Increased routing time.
* Logic resources of the device are used.
* Use of dedicated reset on flip-flops.
* Operational resets result in additional gate before D input or dedicated reset input.
* Almost certainly will impact size of design.
* Additional logic level almost certainly will impact system performance.
* Increased place and route time.
* Prevents use of highly efficient features, such as SRL16E.
* The SRL16E implements up to 16 flip-flops in each LUT.
* These virtual flip-flops do not support reset, and this prevents synthesis tools from
* delivering the full advantage offered by this feature when the HDL specifies a reset.
* Up to 16 times increase in size and product costs.
* Additional size potentially reduces system performance.
* Increased place and route times.

Therefore we can conclude some takeaways:

* Use Localized Synchronous Resets. Use shift-register-based circuits to synchronize reset release with clock domains. This ensures de-assertion aligns with clock edges, minimizing skew.  
  
* Apply resets only to critical components (e.g., feedback loops) using synchronous set/reset flip-flops (FDS/FDR).
* Avoid Global Resets: Leverage FPGA initialization post-configuration for most cases.
* There is no need for global reset designing with Xilinx FPGAs, because they acting as a comprehensive "power-on reset." by initializing all flip-flops and RAM cells during configuration, This eliminates the need for global resets in cost cases, simplifying simulation and avoiding undefined states.
* Targeted Resets: Identify and isolate critical circuits requiring synchronous resets.
* Design Reviews: Assess reset necessity for each component to optimize resource use and reliability.
* The critical parts of a system that must truly be reset should be identified and the release of those resets on start up or during operation must be controlled as carefully as any other signal within a synchronous circuit.
* When creating each section of a design, simply ask, “Does this bit need to be reset”?