# WP272 SUMMARY

* Using global reset in FPGA designs is not a very good idea and should be avoided.
* The assertion of the reset signal is not timing critical. The duration of the reset pulse is generally long compared to the clock period and even if flip-flops are reset at slightly different times, it does not matter because all operations are effectively halted. All top-level behavior is frozen, so minor variations in when flip-flops are reset do not cause issues.
* The deassertion of the reset signal is timing critical.
* Let’s say a reset signal is de-asserted at some time between clock edges. The signal then propagates through various flip-flops. At each flip-flop, the signal should be de-asserted a “set-up period” before the active clock edge. Since the reset signal is a very high fan-out network, meeting the de-assertion timing requirement is very hard.
* The timing of the de-assertion of reset does not matter in a pipelined process. After a few cycles, the entire pipeline will be operational, and any incorrect data will be flushed out of the system. Only the first few outputs may be invalid. Actually, there is no point in a reset.

A diagram of a function

AI-generated content may be incorrect.

* One-Hot state machine is an example where the timing of the reset de-assertion matter. If the second flip-flop comes out of reset at a clock edge different from the first flip-flop, the value ‘1’ will disappear and the state machine will be stuck at all zeros state forever. The probability of this goes lower if the flip-flops are close physically (low skew on localized reset network). A diagram of a machine

  AI-generated content may be incorrect.
* If recovery time is not guaranteed, it could still happen. If the de-assertion of the reset occurs too close to a rising edge, some flip-flops may enter **metastability** (undefined behavior).
* Ultimately, the circuits with feedback loops require reset timing analysis. Circuits without feedback generally do not need a reset.
* When a Xilinx FPGA is configured or reconfigured, every cell is initialized. This can be used as the ultimate reset.
* If asynchronous reset needs to be used, a localized reset circuit can be inserted to control flip-flops that require a reset. Here, active high asynchronous resets are used. When reset is asserted, all flip-flops in the chain are preset to ‘1’. Following the release of the asynchronous reset, the chain begins to fill with ‘0’s each clock cycle. Eventually, the last flip-flop makes the transition from High to Low, and the localized reset is released synchronously with the clock.

A diagram of a computer program

AI-generated content may be incorrect.

* **When creating each section of a design, simply ask, “Does this bit need to be reset”?**

# WP275 SUMMARY

* The design size can be reduced by optimizing logic implementation to make better use of FPGA resources.
* Try to avoid global resets if possible.
* The number of back-to-back connected LUTs can be reduced by utilizing the control signals in flip-flops.
* Reset has the highest priority; Set has the second highest priority and Clock Enable has the lowest priority.
* Priority levels must be taken into account when writing the code so that the dedicated control inputs of the flip-flops can be utilized instead of LUT resources.
* For example, if an asynchronous reset is used, set and clock enable should also be asynchronous (if they exist) so that they can be handled using the dedicated ports in the flip-flop. The same logic applies for the synchronous case.

A diagram of a diagram

AI-generated content may be incorrect.