WP01082 METASTABILITY

What is Metastability?

Metastability occurs when a signal transition deviates from the setup (tSU) or hold (tH) time requirements of a register. Metastability is prevented and the temporal constraints in synchronous systems are met. However, since timing violations can occur randomly, metastability is often observed when signals move between asynchronous clock domains.

In a metastability condition, a register oscillates between '0' and '1' and then settles, temporarily holding an undefined logic level. Device architecture, operational conditions, and process technology all affect the resolution time. If the metastability output settles before the next register can catch up, the system is operating properly. Otherwise, mismatched logic levels can lead to system failure.

Metastability and Design Failures

Especially in asynchronous field transitions, when metastability results in one register producing an unstable signal, the system may experience logic errors. If the output of the metastability register is collected by another register before stabilization, system instability may result from different registers sampling different values.

A popular solution is synchronization registers, which are extra registers that delay signal capture so that metastability can be resolved before the signal is propagated. Several registers timed by the same or phase-correlated clocks form a synchronization register chain. To optimize metastability robustness, synchronization typically uses two or three registers.

Calculating Metastability MTBF

MTBF (Mean Time Between Failures) provides an estimate of how frequently metastability-induced failures occur in a design. The formula for calculating MTBF is

$$MTBF = \frac{e^{t_{MET}/C_2}}{C_1 \cdot f_{CLK} \cdot f_{DATA}}$$

- C1, C2: Device-specific constants determined by process technology and operating conditions.
- f_CLK: Clock frequency of the receiving domain.
- f_DATA: Toggle rate of the asynchronous data signal.
- t_MET: Available metastability settling time (timing slack beyond t_CO).

System-Level MTBF

The total MTBF for a design depends on individual synchronizer chains:

$$failure\_rate_{design} = \frac{1}{MTBF_{design}} = \sum_{i=1}^{number\ of\ chains} \frac{1}{MTBF_i}$$

The worst-performing synchronizer chain dominates the overall metastability robustness, making it critical to optimize all clock domain crossings.

Characterizing Metastability Constants

FPGA vendors determine the C1 and C2 constants by testing FPGA circuits under controlled conditions. Since real-world FPGA metastability failures are rare (due to high MTBF values), test circuits are designed to force frequent metastability events.

A common test setup involves two unrelated clocks, a fast toggling data input, and a synchronizer feeding multiple destination registers. If metastability occurs, discrepancies between register outputs indicate unresolved metastability, enabling statistical analysis of metastability behavior.

Improving Metastability MTBF

Constants C1 and C2 are the factory values of the product. So the designer need to increase t_MET parameter in the formula to higher MTBF values.

Since MTBF depends exponentially on *t_MET*, increasing available timing slack dramatically improves robustness. Design strategies include:

- Adding synchronization registers (e.g., using three registers instead of two).
- Optimizing placement to reduce delays in synchronization chains.
- Using FIFO buffers for complex clock domain crossings, ensuring data integrity.