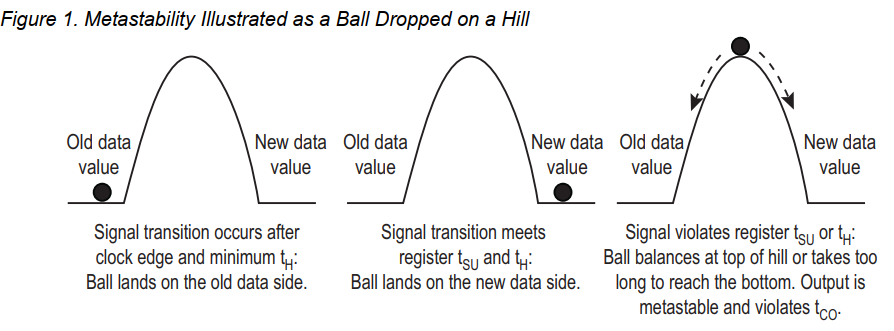
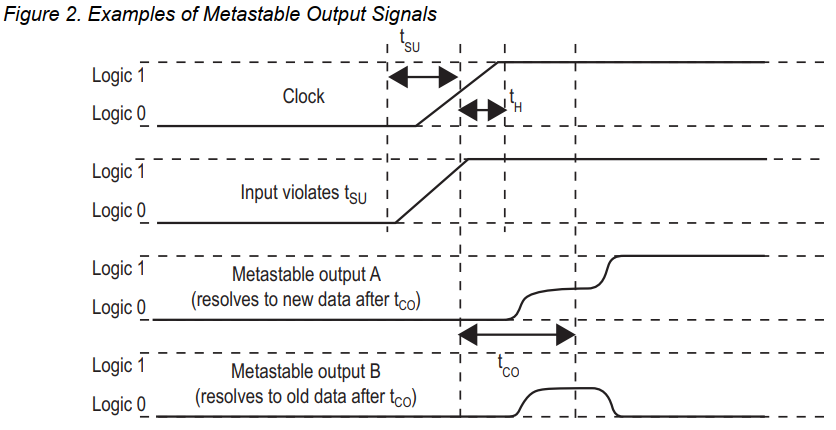
**Understanding Metastability Summary**

What Is Metastability?

All registers in digital devices such as FPGAs have defined signal timing requirements that allow each register to correctly capture data at its inputs and produce an output signal. To ensure reliable operation, the input to a register must be stable for a minimum time before the clock edge (register setup time or tSU) and for a minimum time after the clock edge (register hold time or tH). The register output then is available after a specified clock-to-output delay (tCO). If a data signal transition violates a register’s tSU or tH requirements, the output of the register may go into a metastable state. In a metastable state, the register output hovers at a value between the high and low states for some period of time, which means the output transition to a defined high or low state is delayed beyond the specified tCO.

In synchronous systems, the input signals must always meet the register timing requirements, so metastability does not occur. Metastability problems commonly occur when a signal is transferred between circuitry in unrelated or asynchronous clock domains. The designer cannot guarantee that the signal will meet tSU and tH requirements in this case, because the signal can arrive at any time relative to the destination clock. However, not every signal transition that violates a register’s tSU or tH results in a metastable output. The likelihood that a register enters a metastable state and the time required to return to a stable state vary depending on the process technology used to manufacture the device and on the operating conditions. In most cases, registers will quickly return to a stable defined state.





When a register’s data input violates the tSU or tH, it is analogous to the ball being dropped on the hill. If the ball lands near the top of the hill, the ball takes too long to reach the bottom, which increases the delay from the clock transition to a stable output beyond the defined tCO.  
in Figure2 for A and B, In both cases, the output transition to a defined 1 or 0 state is delayed beyond the register’s specified tCO.

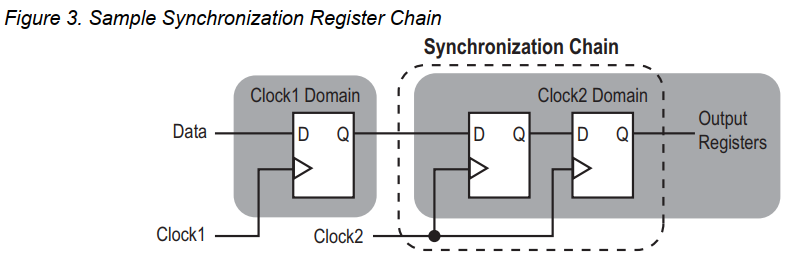
When Does Metastability Cause Design Failures?

If the data output signal resolves to a valid state before the next register captures the data, then the metastable signal does not negatively impact the system operation. But if the metastable signal does not resolve to a low or high state before it reaches the next design register, it can cause the system to fail.

Synchronization Registers

When a signal transfers between circuitry in unrelated or asynchronous clock domains, it is necessary to synchronize this signal to the new clock domain before it can be used. The first register in the new clock domain acts as a synchronization register.

In a full synchronizer circuit, the signal-crossing clock domains should pass from the originating flip-flop in the original clock domain to the first flipflop of the synchronizer without passing through any combinational logic between the originating flip-flop and the first flip-flop of the synchronizer. **NO COMBINATONAL LOGIC HERE!**



The timing slack available in the synchronizer register-to-register paths is the time available for a metastable signal to settle, and is known as the available metastability settling time.

A synchronization register chain, or synchronizer, is defined as a sequence of registers that meets the following requirements:

■ The registers in the chain are all clocked by the same or phase-related clocks

■ The first register in the chain is driven from an unrelated clock domain, or asynchronously

■ Each register fans out to only one register, except the last register in the chain

Note that any asynchronous input signals, or signals that transfer between unrelated clock domains, can transition atany point relative to the clock edge of the capturing register. Therefore the designer cannot predict the sequence of asignal’s transitions or the number of destination clock edges until the data transitions. For example, if a bus ofasynchronous signals is transferred between clock domains and synchronized, the data signals could transition ondifferent clock edges. As a result, the received values of the bus data could be incorrect.

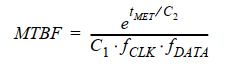
The designer must accommodate this behavior with circuitry such as dual-clock FIFO (DCFIFO) logic to store thesignal values, or hand-shaking logic. FIFO logic uses synchronizers to transmit control signals between the two clockdomains, then data is written and read with dual-port memory.

if an asynchronous signal acts as part of hand-shaking logic between two clock domains, control signals indicate when data can be transferred between clock domains. In this case, synchronization registers are used to ensure that metastability will not interfere with the reception of control signals and that the data has enough settling time for any metastable conditions to resolve before the data is used. In a properly-designed system, the design functions correctly as long as each signal resolves to a stable value before it is used.

Calculating Metastability MTBF

The mean time between failures, or MTBF, due to metastability provides an estimate of the average time between instances when metastability could cause a design failure. A higher MTBF (such as hundreds or thousands of years between metastability failures) indicates a more robust design. The required MTBF depends on the system application. For example, a life-critical medical device requires a higher MTBF than a consumer video-display device. Increasing the metastability MTBF reduces the chance that signal transfers will cause any metastability problems on the device.

The MTBF of a synchronizer chain is calculated with the following formula and parameters:

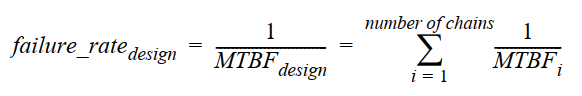


The C1 and C2 constants depend on the device process and operating conditions.

The fCLK and fDATA parameters depend on the design specifications: fCLK is the clock frequency of the clock domain receiving the asynchronous signal and fDATA is the toggling frequency of the asynchronous input data signal. Faster clock frequencies and faster-toggling data reduce (or worsen) the MTBF.

The tMET parameter is the available metastability settling time, or the timing slack available beyond the register’s tCO, for a potentially metastable signal to resolve to a known value. The tMET for a synchronization chain is the sum of the output timing slacks for each register in the chain.

The overall design MTBF can be determined by the MTBF of each synchronizer chain in the design. The failure rate for a synchronizer is 1/MTBF, and the failure rate for the entire design is calculated by adding the failure rates for each synchronizer chain, as follows:



Improving Metastability MTBF

Due to the exponential factor in the MTBF equation, the tMET/C2 term has the largest effect on the MTBF calculation. Therefore metastability can be improved by optimizing the device’s C2 constant with architecture enhancements, or optimizing the design to increase the tMET in the synchronization registers.

FPGA Architecture Enhancements

**The metastability time constant C2 in the MTBF equation depends on various factors related to the process technology used to manufacture the device, including the transistor speed and the supply voltage.** Faster process technologies and faster transistors allow metastable signals to resolve more quickly. As FPGAs have migrated from 180-nm process geometries to 90 nm, the increase in transistor speed usually improves metastability MTBF. Therefore, metastability has not been a major concern for FPGA designers.

However, as the supply voltage reduces with reduced process geometries, the threshold voltage for the circuit does not decrease proportionally. When a register goes metastable, its voltage is approximately one-half of the supply voltage. **With a reduced power supply voltage, the metastable voltage level is closer to the threshold voltage in the circuit. When these voltages get closer together, the gain of the circuit is reduced and the registers take longer to transition out of metastability**. As FPGAs enter the 65-nm process geometry and lower, with power supplies at 0.9V and lower, the threshold voltage consideration is becoming more important than the increase in transistor speed. Therefore, metastability MTBFs generally get worse unless the vendor designs the FPGA circuitry to improve metastability robustness.

Design Optimizations

The exponential factor in the MTBF equation means that an increase in the design-dependent tMET value increases a synchronizer MTBF exponentially. For example, if the C2 constant for a given device and set of operating conditions is 50 ps, then an increase of just 200 ps in the tMET makes the exponent 200/50 and increases the MTBF by factor e4, or more than 50 times, while an increase of 400 ps multiplies the MTBF by e8, or almost 3000 times.

**To improve metastability MTBF, designers can increase** tMET **by adding extra register stages to synchronization register chains. The timing slack on each additional register-to-register connection is added to the** tMET **value. Designers commonly use two registers to synchronize a signal, but Altera recommends using a standard of three registers for better metastability protection. However, adding a register adds an additional latency stage to the synchronization logic, so designers must evaluate whether that is acceptable.** **You also need to place the flipflops close to each other to ensure the  
smallest possible clock skew between them.**

Conclusion

Metastability can occur when signals are transferred between circuitry in unrelated or asynchronous clock domains. The mean time between metastability failures is related to the device process technology, design specifications, and timing slack in the synchronization logic. FPGA designers can improve system reliability and increase metastability MTBF by increasing the tMET with design techniques that add timing slack in synchronization registers.