Introduction to FPGA Design

Getting Started with Xilinx FPGAs
Version 2.1i

Outline

- Hierarchical Design
- **Synchronous Design for Xilinx FPGAs**
- Summary

Synchronous Design

- Why Synchronous Design?
- Xilinx FPGA Design Tips

Why Synchronous Design?

- **Synchronous circuits are more reliable**
  - *Events are triggered by clock edges which occur at well-defined intervals*
  - *Outputs from one logic stage have a full clock cycle to propagate to the next stage*
    - Skew between data arrival times is tolerated within the same clock period
- **Asynchronous circuits are less reliable**
  - *A delay may need to be a specific amount (e.g. 12ns)*
  - *Multiple delays may need to hold a specific relationship (e.g. DATA arrives 5ns before SELECT)*

Case Studies

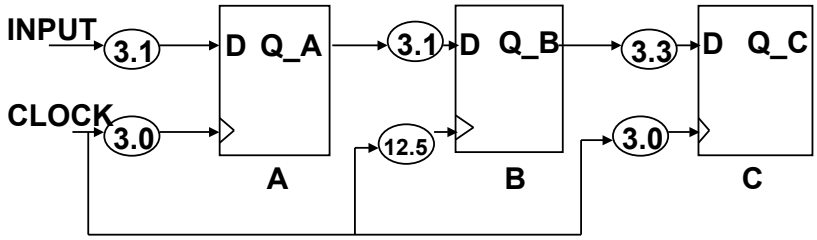
- **The design I did two years ago no longer works. What did Xilinx change in their FPGAs?**
  - *SRAM process improvements and geometry shrinks increase speed*
  - *Normal variations between wafer lots*
- **My design was working, but I re-routed my FPGA and now my design fails. What is happening?**
  - *Logic placement has changed, which affects internal routing delays*
- **My design passes a back-annotated timing simulation but fails in circuit. Is the timing simulation accurate?**
  - *Yes, the simulation is accurate*
  - *Timing simulation uses worst-case delays*
  - *Actual board-level conditions are usually better*

Design Tips

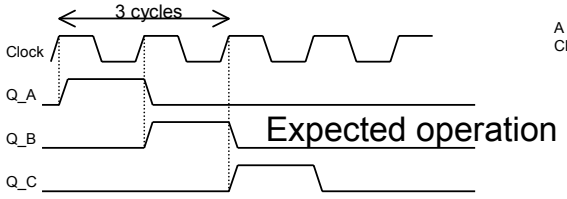
Xilinx FPGA Design

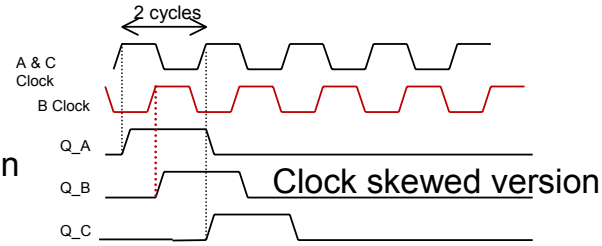
- **Reduce clock skew**
- **Clock dividers**
- **Avoid glitches on clocks and asynchronous set/reset signals**
- **The Global Set/Reset network**
- **Select a state machine encoding scheme**
- **Access carry logic**
- **Build efficient counters**

Clock Skew



- **This shift register will not work because of clock skew!**



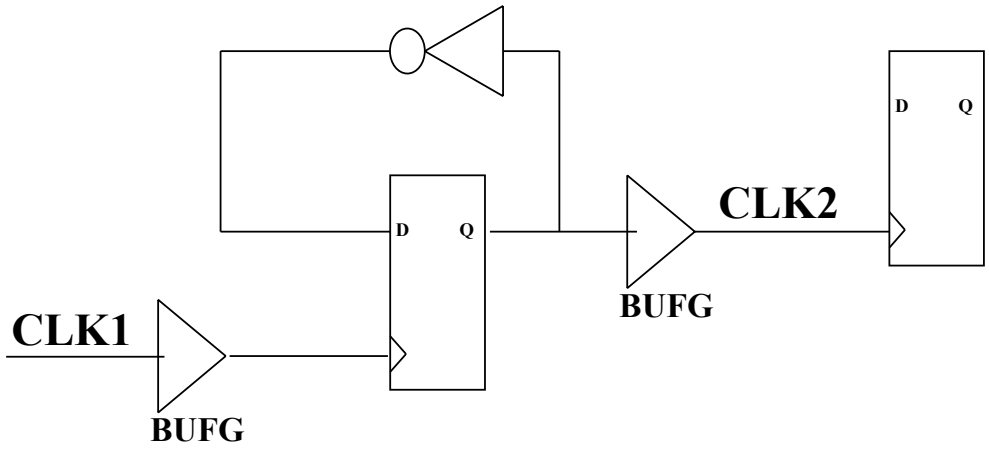


Use Global Buffers to Reduce Clock Skew

- **Global buffers are connected to dedicated routing**
  - *This routing network is balanced to minimize skew*
- **All Xilinx FPGAs have global buffers**
- **Different types of global buffers**
  - *XC4000E/L and Spartan have 4 BUFGPs and 4 BUFGSs*
  - *XC4000EX/XL/XV have 8 BUFGLSs*
  - *Virtex has 4 BUFGs or BUFGDLLs*
- **You can always use a BUFG symbol and the software will choose an appropriate buffer type**
  - *Most synthesis tools can infer global buffers onto clock signals*

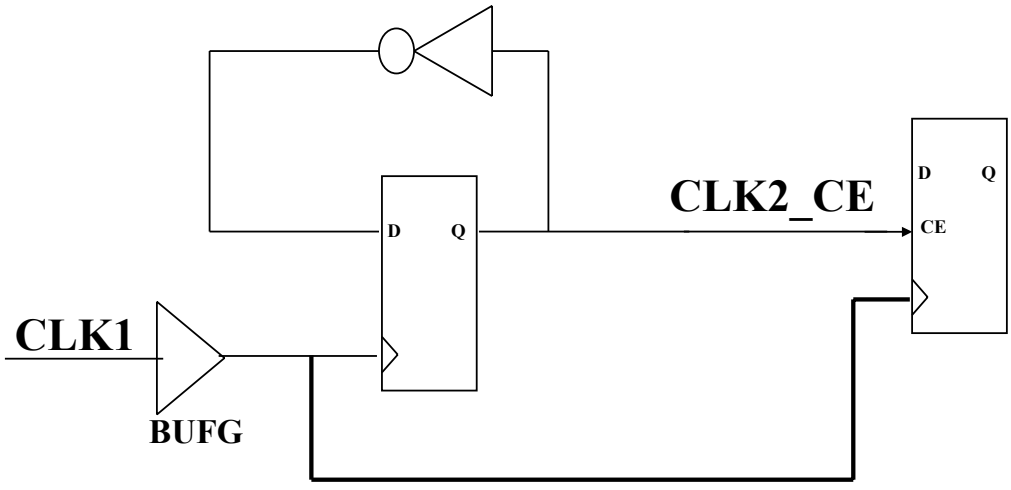
Traditional Clock Divider

- Introduces clock skew between CLK1 and CLK2
- Uses an extra BUFG to reduce skew on CLK2



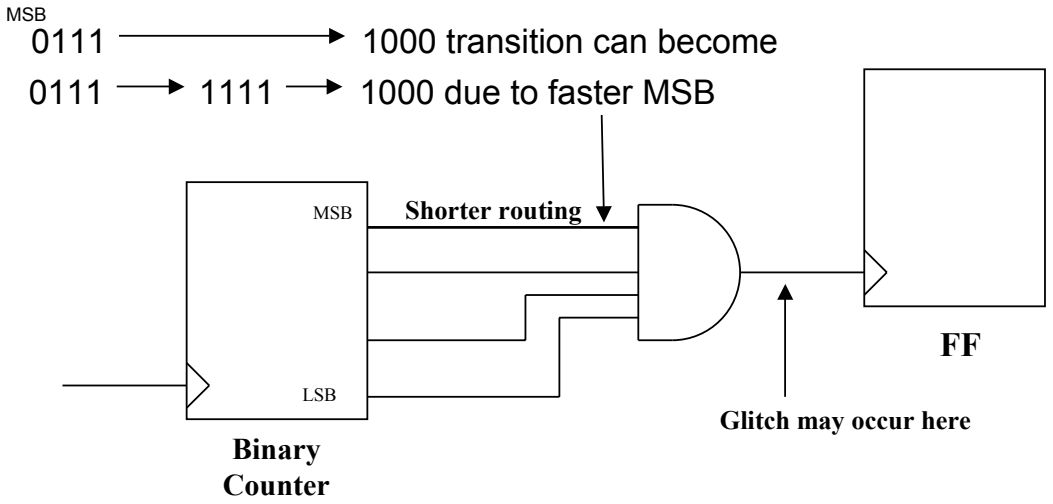
Recommended Clock Divider

- No clock skew between flip-flops



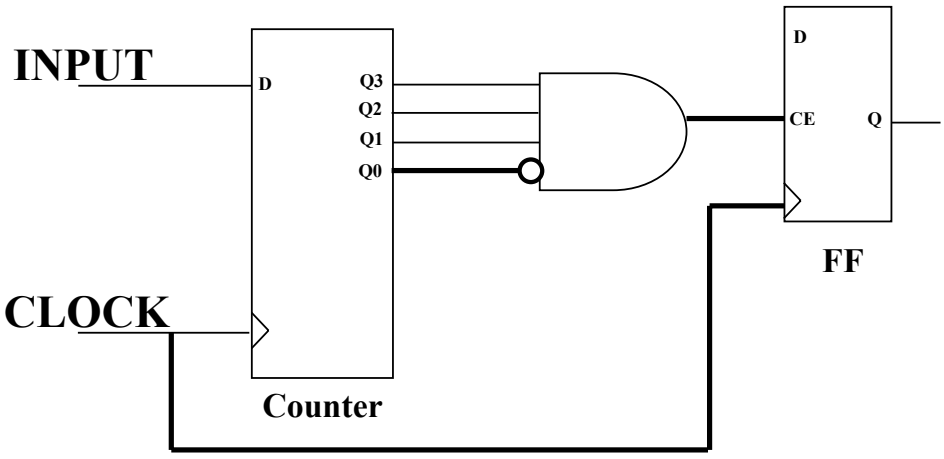
Avoid Clock Glitches

- **Because flip-flops in today's FPGAs are very fast, they can respond to very narrow clock pulses**
- **Never source a clock signal from combinatorial logic**
  - *Also known as "gating the clock"*



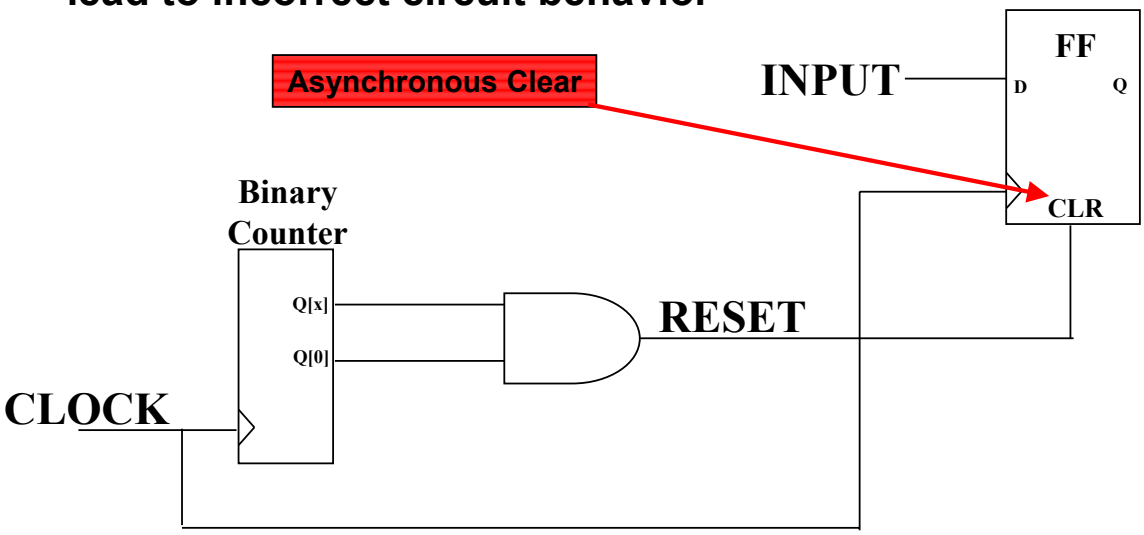
Avoid Clock Glitches: Answer

- **Complete in the circuit to create the same function, but without glitches on the clock**



Avoid Set/Reset Glitches

- Glitches on asynchronous clear or preset inputs can lead to incorrect circuit behavior



Avoid Set/Reset Glitches

- Convert to synchronous set or reset when possible