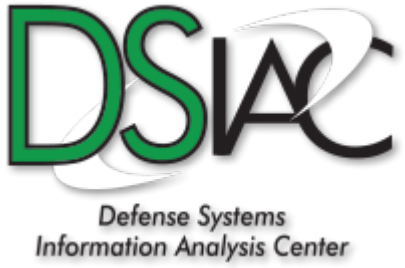
Common Problems and Considerations for High-Speed Field-Programmable Gate Array (FPGA) Designs



Levi Smolin
EngeniusMicro, LLC.
Director of Advanced Sensors
Levi.Smolin@engeniusmicro.com
256-334-9622

DSIAC is a DoD Information Analysis Center (IAC) sponsored by the Defense Technical Information Center (DTIC), with policy oversight provided by the Office of the Under Secretary of Defense (OUSD) for Research and Engineering (R&E). DSIAC is operated by the SURVICE Engineering Company.

Background

High-Speed Considerations

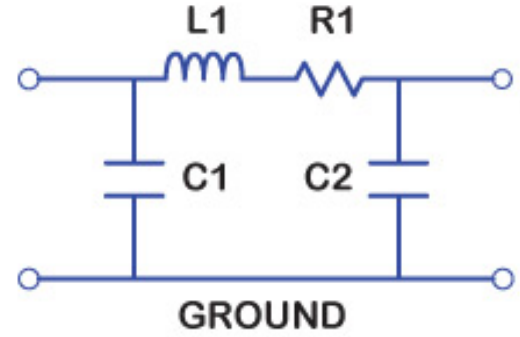
- **Definition of “high speed”**
  - Integrity of signals is affected by characteristics of your design.
  - Often 500 MHz and above
- **More common than ever**
  - DDR3 memory
  - Analog-to-digital converters
  - Digital-to-analog converters
  - Bluetooth, Wi-Fi, Ethernet, etc.

High-speed signals are often length matched.

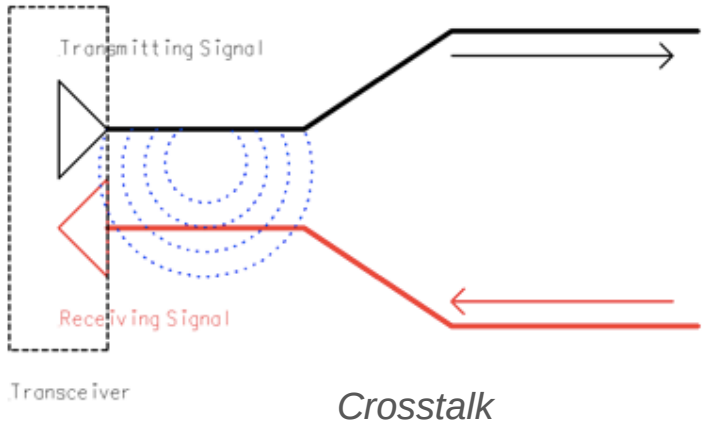
High-Speed Considerations

Propagation

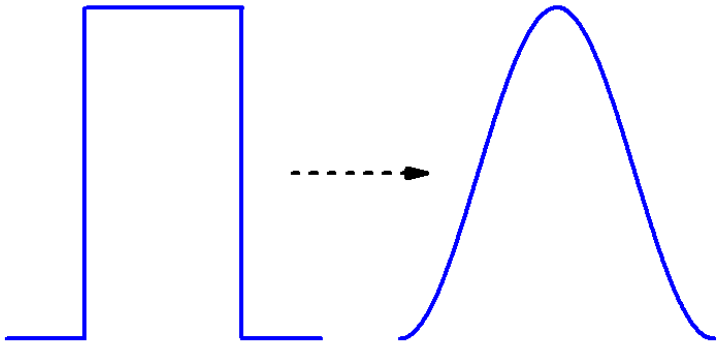
- Pulse smearing
- Crosstalk
- Noise
- Attenuation



Model of transmission line



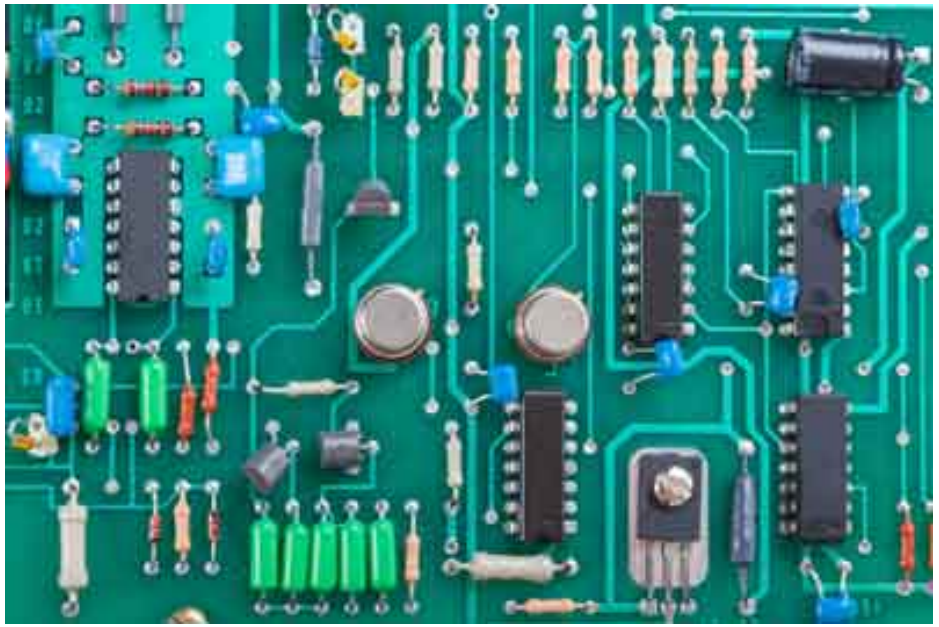
Crosstalk



Pulse smearing

Printed Circuit Board (PCB)

- **Summary**
  - Combines multiple components
- **Features**
  - Low nonrecurring expenses (NREs)
  - Somewhat low unit costs
  - Fast design time
  - No reusability
  - Bad at high speed
  - Large size
  - High-power use

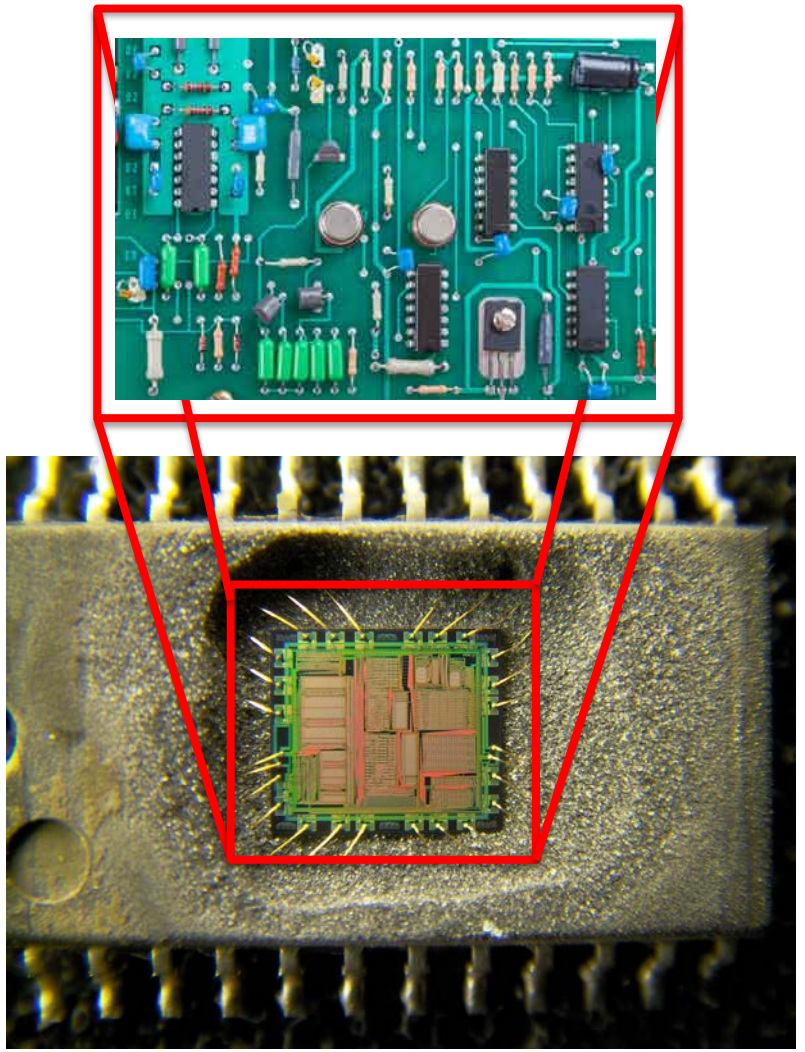


PCB with discrete components

Application-Specific Integrated Circuit (ASIC)

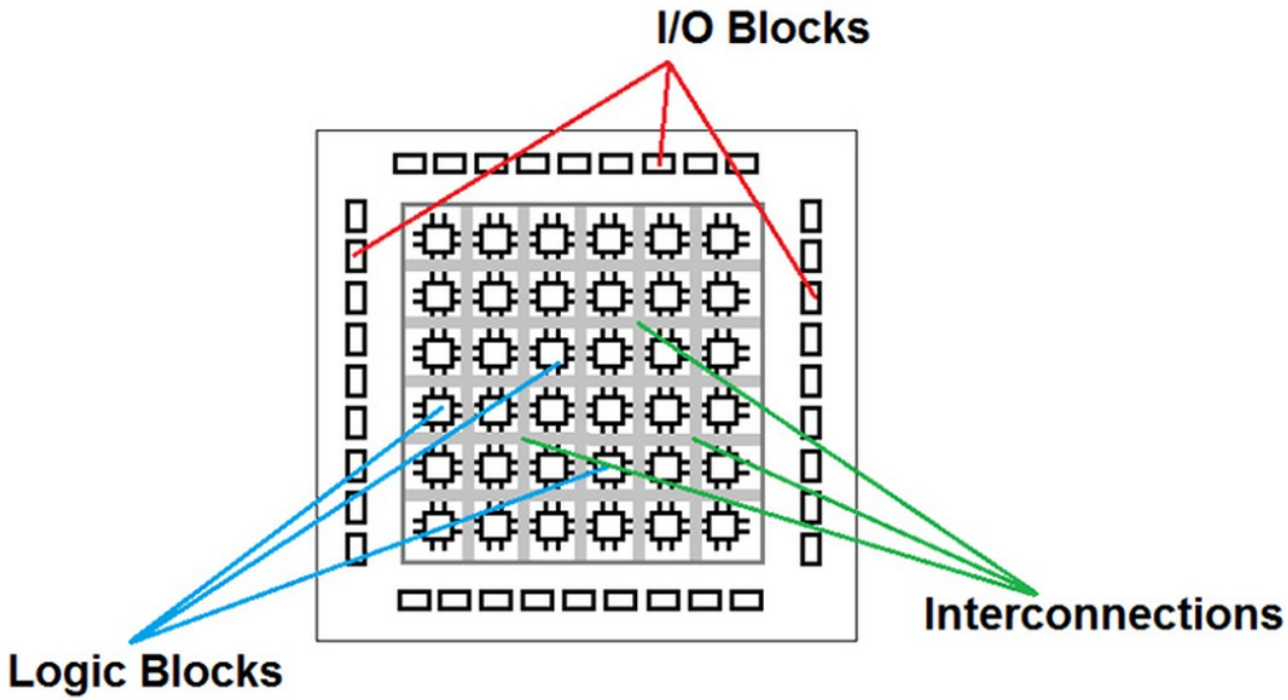
- **Summary**
  - Custom design
  - Single purpose
  - Integrates many component parts into one chip solution

- **Features**
  - High NREs
  - Low unit costs
  - Slow design time
  - No reusability
  - Good at high speeds
  - Small size
  - Low power



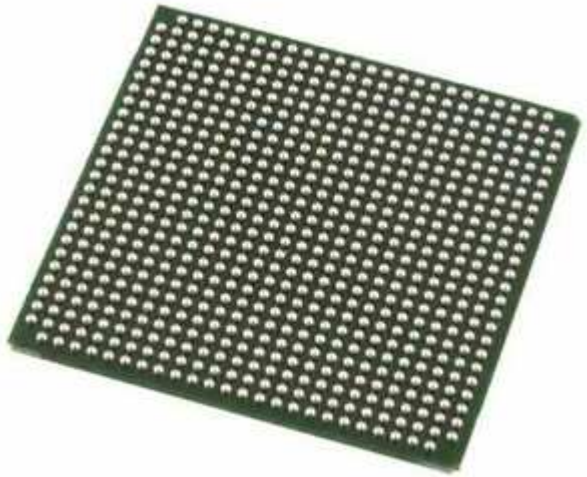
Field Programmable Gate Arrays

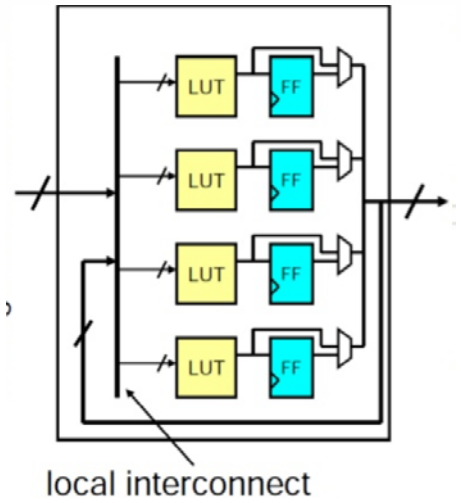
- **Benefits**
  - Generic structure
  - Alternative to ASIC
  - Xilinx, Altera (Intel), Lattice
- **Very "middle of the road"**
  - Good for low-volume, application-specific uses
  - Fast development time
  - Reprogrammable
    - Great for test beds, rapidly evolving tech, etc.
- **Present their own considerations**

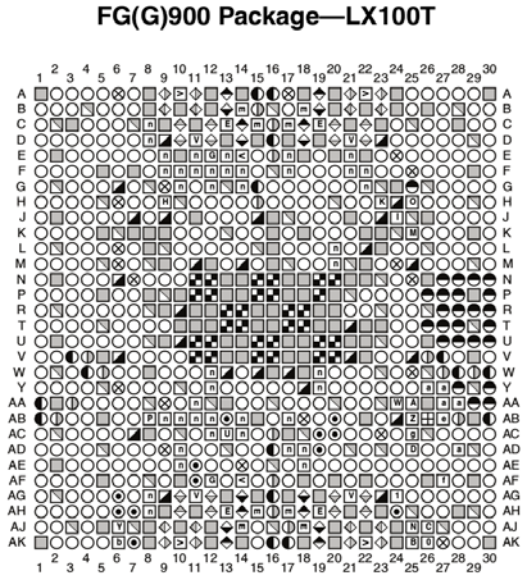


Field Programmable Gate Arrays

- **I/O**
  - Buffers
- **Configurable logic blocks**
  - Look-up tables
  - Flip-flops
  - Logic gates
- **Dedicated hardware**
  - DSPs
  - Memory
- **Interconnects**

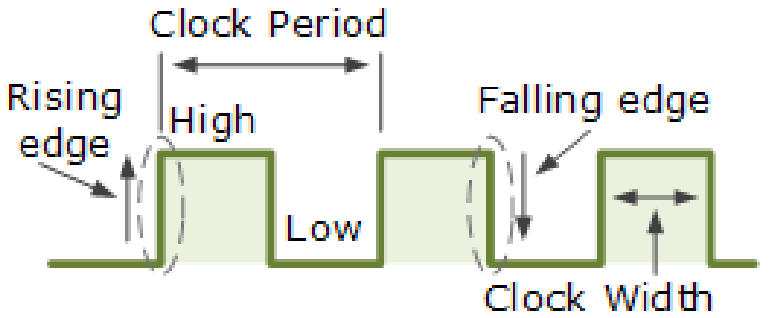




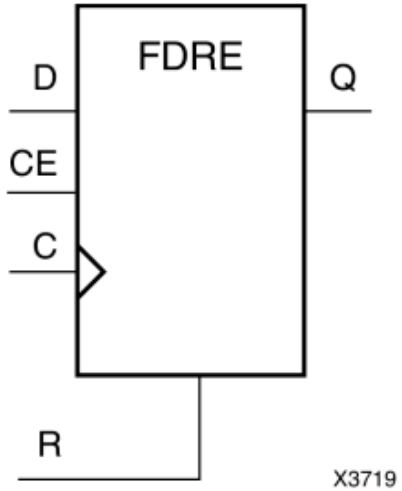


Digital Design Considerations

- **Clocks**
  - Used as timekeepers
  - **Good clock is essential.**
    - **Low jitter, phase noise**
- **Signal types**
  - Synchronous
  - Asynchronous
- **Flip-flop**
  - Holds data
  - Adds one clock delay



Digital clock signal

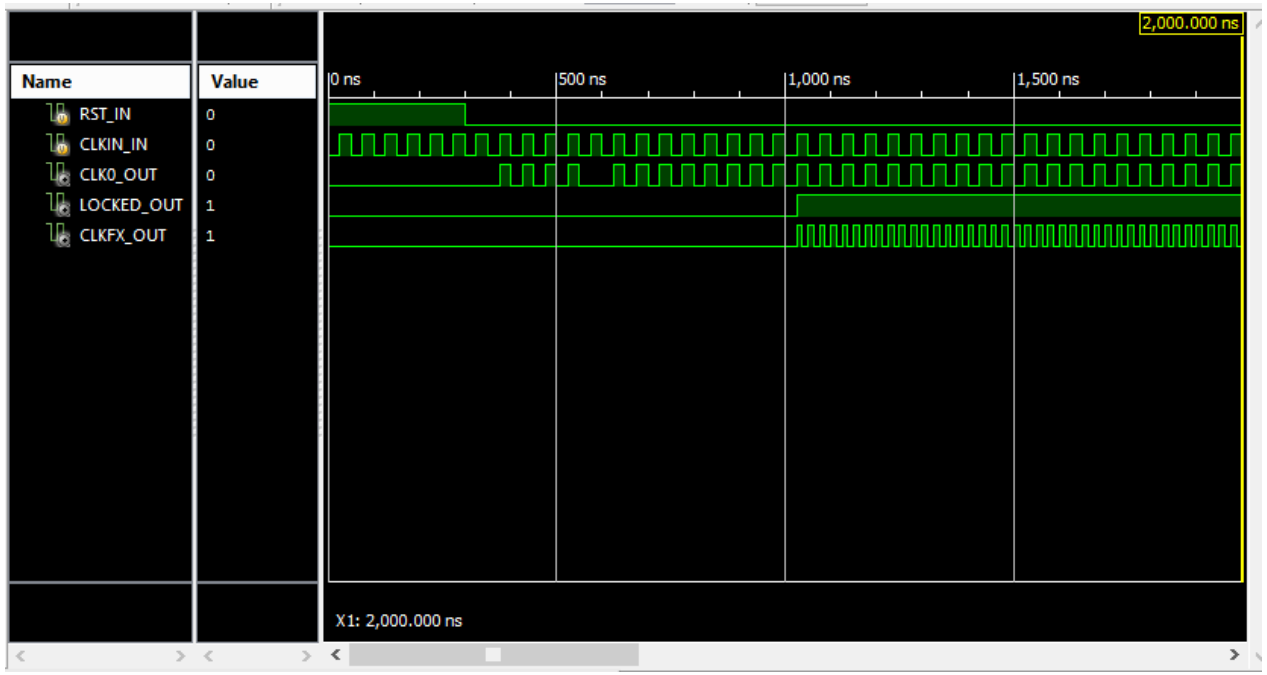


Flip-flop

Design Considerations

Typical FPGA Design Flow

- Start from design specs
- Code for functionality
- Run functional simulation
  - Does not account for hardware



*Tools like Modelsim run **functional** simulations only.*

- Synthesize/implement design
- Run timing analysis
  - Design fails timing requirements.
- Recompile design

Code to Hardware

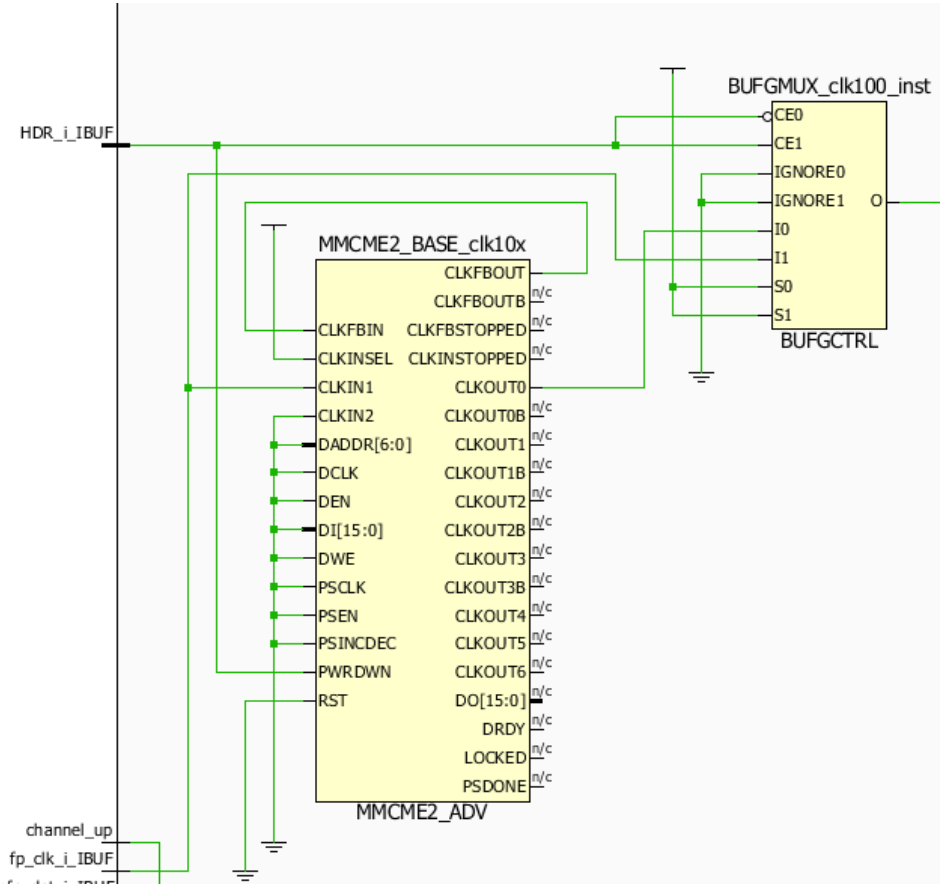
- **Hardware Description Languages (HDLs)**
  - Specialized language to describe hardware components
- **VHDL, Verilog**
  - Slight differences
- **Synthesis**
  - The process of turning HDL into a design implemented in logic gates and other hardware
  - Pin diagrams
- **Implementation**
  - The process of placing and routing the synthesized design onto an FPGA
  - Physical placement is very important for design functionality.

```
led1Proc : process(user_clk)
begin
  if rising_edge(user_clk) then
    if unsigned(count1) > LED_COUNT1 then
      count1 <= (others => '0');
      led1   <= not led1;
    else
      count1 <= std_logic_vector(unsigned(count1) + 1);
    end if;
  end if;
end process;
```

VHDL code example

Code to Hardware

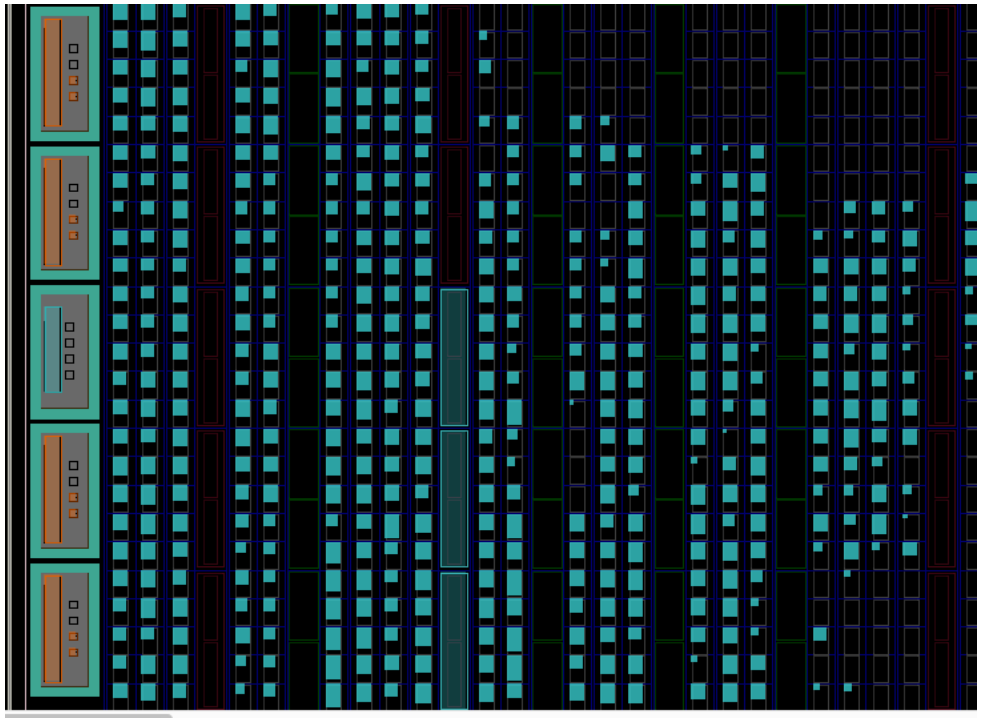
- **Synthesis**
  - The process of turning HDL into a design implemented in logic gates and other hardware
  - Pin diagrams



A synthesized design

Code to Hardware

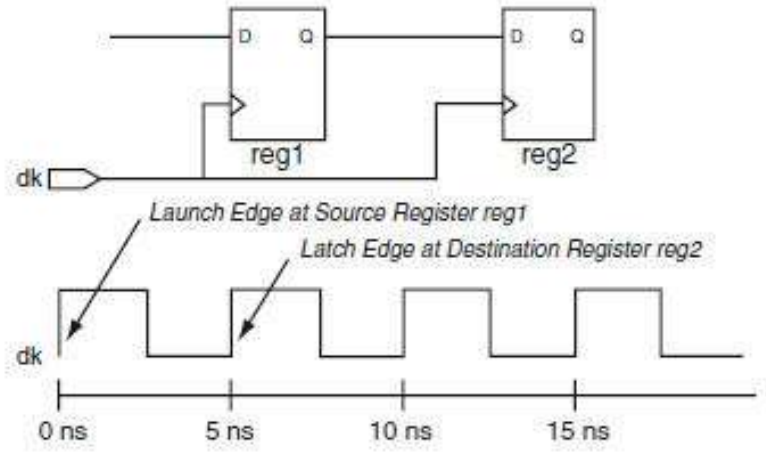
- **Implementation**
  - The process of placing and routing the synthesized design onto an FPGA
  - Physical placement is very important for design functionality.



Implemented design

Definitions

- **Slack**
  - Margin by which timing requirement is met (or not)
  - Negative slack is bad.
- **Worst negative slack**
  - Signal which comes closest to missing (or misses worst) timing requirements
- **Launch edge**
  - Clock edge that sends data from one element to the next
- **Latch edge**
  - Clock edge that captures data on an element



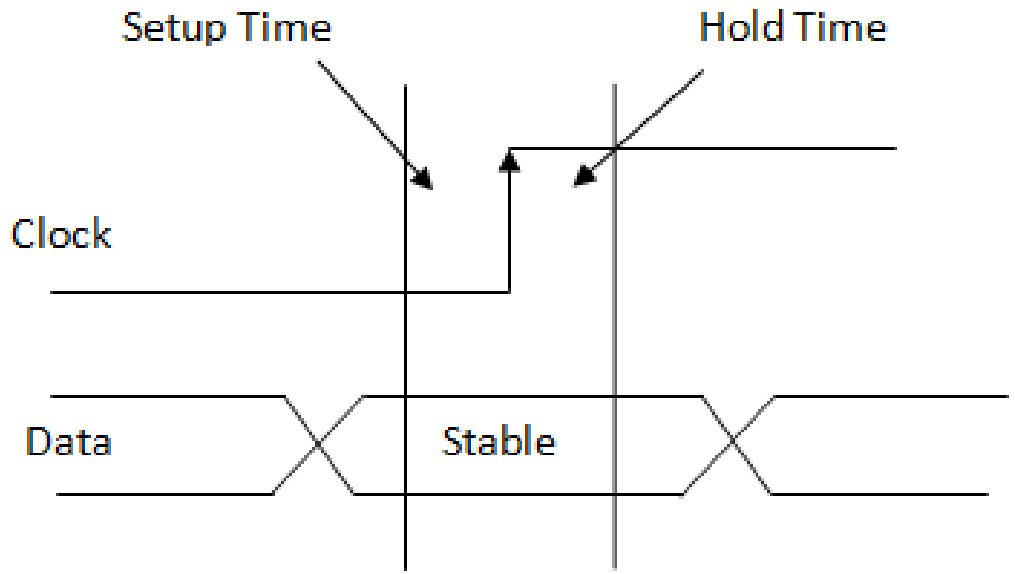
Launch and latch edges

Timing Closure

- **Definition**
  - Process by which a design is made to meet timing requirements
- **Vendor tools help find common problems.**
- **Common problems**
  - Setup and hold
  - High fan-out
  - Multiple clocks

Setup and Hold Violation

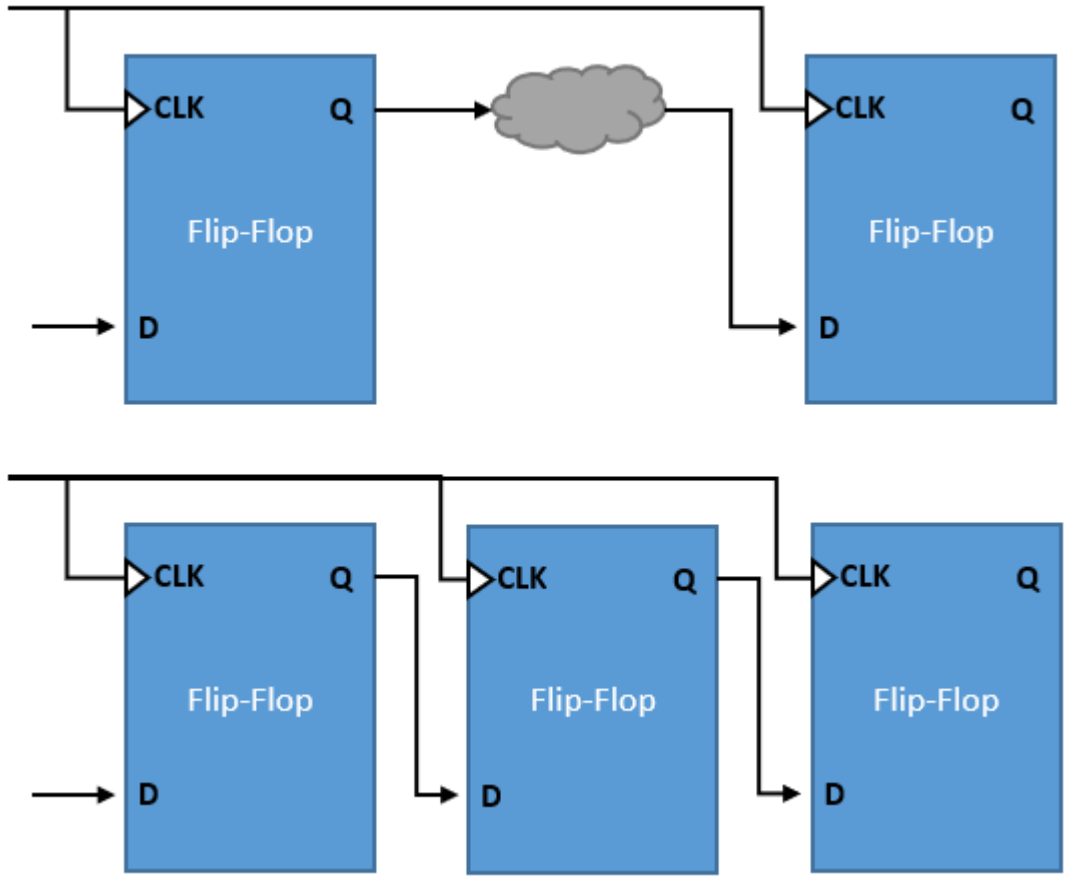
- **Setup time**
  - Length of time data must be stable **before** the clock signal changes.
- **Hold time**
  - Length of time data must be stable after the clock signal changes.
- **Violations occur often because a signal cannot propagate in time.**



Setup and hold requirements

Setup and Hold

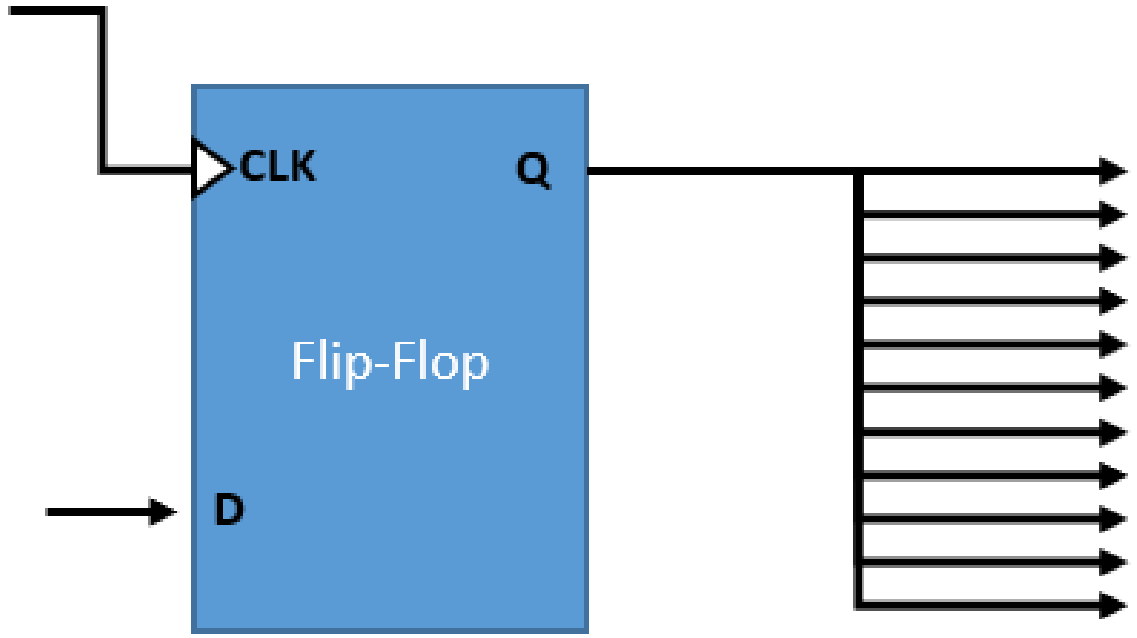
- **Add additional delay to signal path**
  - Must be accounted for in overall design
- **Lower clock frequency**
  - Can be hard to achieve
- **Change physical location of registers on FPGA**
  - Can be done with constraints



Adding another flip-flop gives additional time to propagate a signal.

High Fan-Out

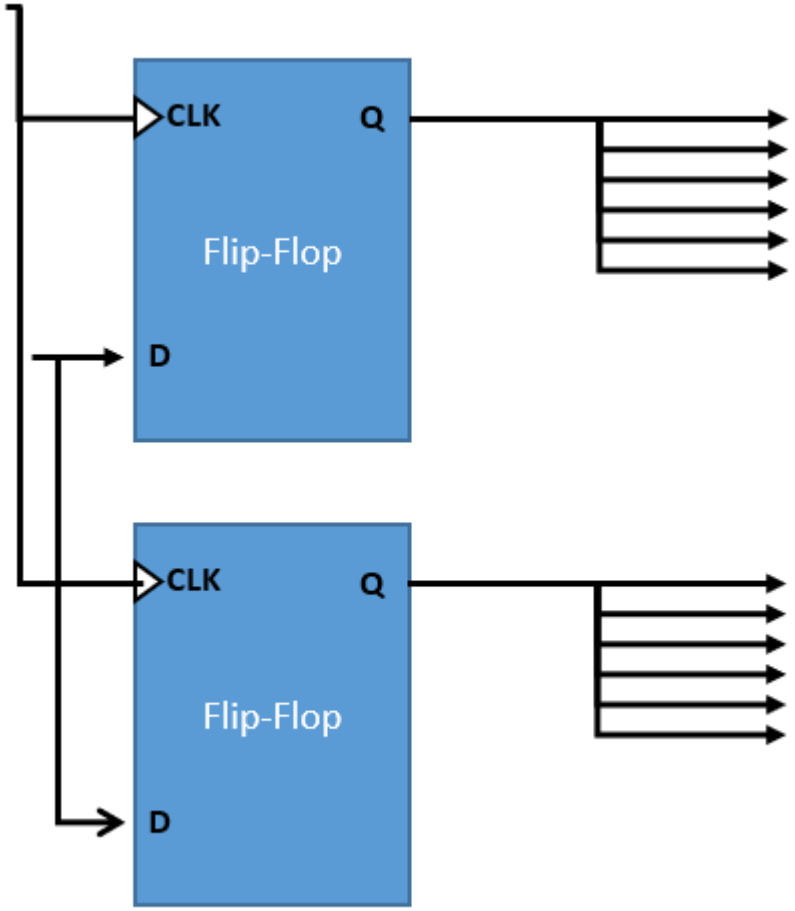
- **Fan-outs**
  - The number of endpoints the output of a register is routed to
- **Problems**
  - Long delay for routing
  - Hard to optimize placement of launching flip-flop relative to latching flip-flops



A high-fan-out situation

High Fan-Out

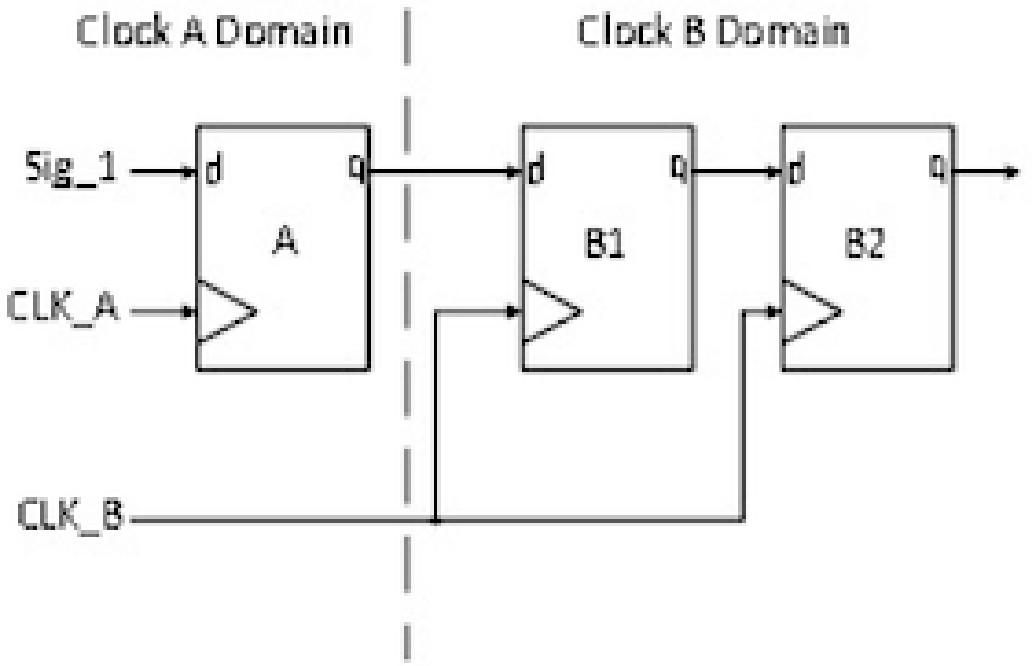
- **Duplicate launch flip-flop**
  - New flip-flop can be placed to optimize routing for distance destinations.
- **Be careful of routing of original data line**
  - Add delays as necessary



Duplicated flip-flop can be placed optimally.

Multiple Clocks

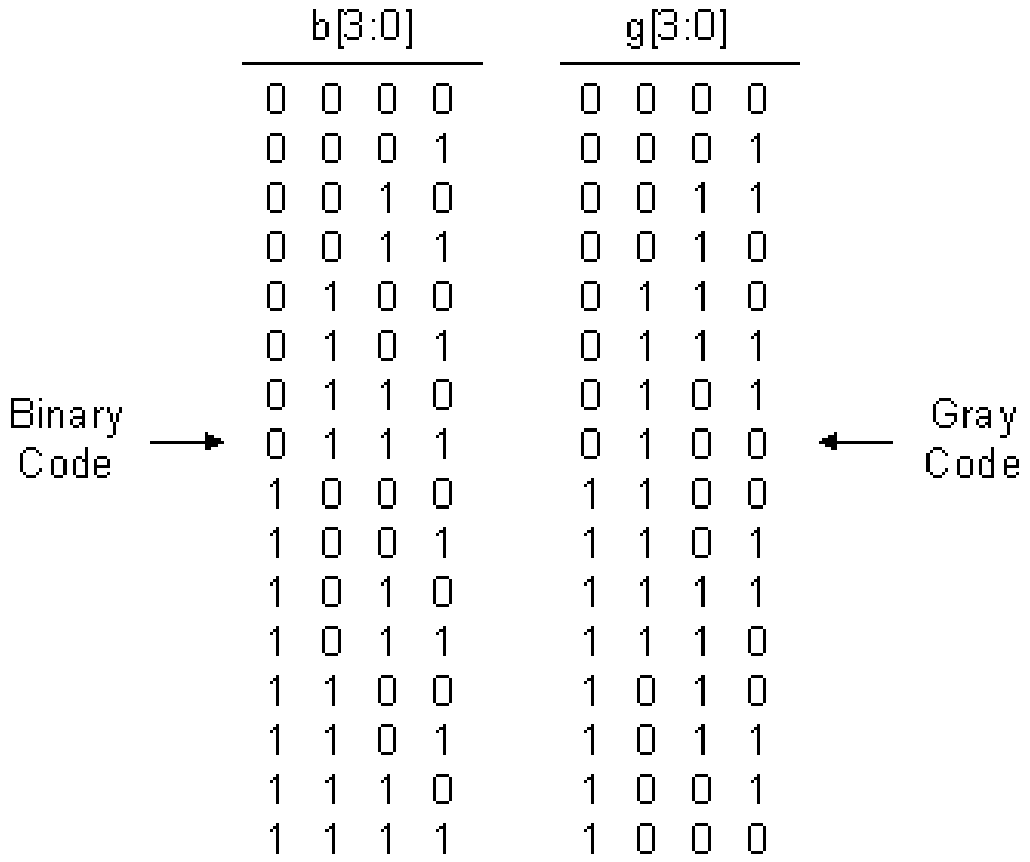
- **Tricky to transition correctly**
  - Can violate setup/hold
  - Can give nondeterminism
- **Design must accommodate for potential drift and offset between non-related clocks**
- **Design must correctly constrain:**
  - False path
  - Multicycle delay



Signal crossing between two clock domains

Multiple Clocks

- **Single bit**
  - Can be solved by chaining registers
  - Removes indeterminism.
  - Three are usually sufficient.
- **Multibit**
  - Must ensure that all bits are synchronized.
  - Try to use vendor asynchronous first-in first-outs.
- **Gray code**
  - Binary sequence where only one bit changes per state.



Signal crossing between two clock domains

Summary

- Simulation not equal to synthesis
- Look to vendor references and guidelines

Questions?