Lecture 16
Timing and Clock Issues

Overview

- Sequential system timing requirements
- Impact of clock skew on timing
- Impact of clock jitter on timing
- Clock distribution
Clocked Synchronous State Machine

Flip-Flop Timing Parameters
Latch Timing Parameters

State Machine Timing

Timing margin equation: \( t_{\text{clk}} \geq t_{\text{flip}} + t_{\text{comb}} + t_{\text{setup}} \)

Setup time margin = \( t_{\text{clk}} - t_{\text{flip}} - t_{\text{comb}} - t_{\text{setup}} \)
Satisfying Timing Requirements

- The period must be long enough for the data to propagate through the registers and logic and to be set up at the destination register before the next rising edge of the clock. Satisfied by making $T$ long enough. **Cycle time:**
  - $T_{CLK} > t_{c-q} + t_{logic} + t_{su}$

- The hold time at the destination register must be shorter than the minimum propagation delay through the logic network. This requirement is independent of system clock; manufacturer’s minimum delay specifications are needed. Guarantee that minimum combinational logic delay is larger than hold time. **Race margin:**
  - $t_{hold} < t_{c-q,cd} + t_{logic,cd}$

Clock Uncertainties

1. Clock Generation
2. Devices
3. Interconnect
4. Power Supply
5. Temperature
6. Capacitive Load
7. Coupling to Adjacent Lines
Clock Nonidealities

- Clock Skew
  - Spatial variations in equivalent clock edges
  - Mostly deterministic

- Clock Jitter
  - Temporal variations in consecutive clock edges
  - Mostly random

- Pulse Width Variation

Clock Skew and Clock Jitter

- Clock skew and jitter can affect the cycle times
- Clock skew can cause race conditions
Overview

- Sequential system timing requirements
- Impact of clock skew on timing
- Impact of clock jitter on timing
- Clock distribution

Clock Skew

All flip-flops in clocked synchronous state machine should be clocked at the “same” time. Violating this rule may result in hold time violations.

**Bad design**

“Same” time means that difference between active edges should be small compared to hold time.

Clock rise and fall times should be short, in case flip-flops respond to different voltage levels. (Use similar flip-flops when possible.)
Clock Skew

Assume the following timing parameters are available:

- **Contamination or minimum delay** ($t_{c-q,cd}$) and maximum propagation delay ($t_{c-q}$) of the register
- Setup ($t_{su}$) and Hold ($t_{hold}$) times for registers
- Contamination delay ($t_{logic,cd}$) and maximum delay ($t_{logic}$) of the combinational logic
- The positions of the rising edges of clocks CLK1 and CLK2 ($t_{CLK1}$ and $t_{CLK2}$) relative to a *global reference*. Ideally $t_{CLK1} = t_{CLK2}$. 
Positive Clock Skew

- Launching edge arrives before the receiving edge
- Minimum clock cycle:
  \[ T + \delta \geq t_{c-q} + t_{\text{logic}} + t_{su} \]

Negative Clock Skew

- Receiving edge arrives before the launching edge
- Minimum clock cycle:
  \[ T + \delta \geq t_{c-q} + t_{\text{logic}} + t_{su} \]
Positive and Negative Clock Skew

(a) Positive skew

(b) Negative skew

Impact of Clock Skew on Timing:
Cycle Time (Long Path)

\[ t_{c-q} + t_{logic} + t_{su} < T_{Clk} + \delta \]
\[ T_{Clk} > t_{c-q} + t_{logic} + t_{su} - \delta \]
Impact of Clock Skew on Timing:
Race Margin (Short Path)

Overview
- Sequential system timing requirements
- Impact of clock skew on timing
- Impact of clock jitter on timing
- Clock distribution
Clock Jitter

Impact of Clock Jitter on Timing: Cycle Time (Late-Early Problem)
Impact of Clock Jitter on Timing

Negative impact on cycle time

\[ T_{\text{Clk}} > t_{c-q} + t_{\text{logic}} + t_{su} + 2 \ t_{jitter} \]

No direct effect on race immunity (same Clk edge)

Jitter reduces performance

Impact of Clock Skew and Jitter: Cycle Time (Late-Early Problem)

\[ t_{c-q} + t_{\text{logic}} + t_{su} < T_{\text{Clk}} - t_{jitter} - t_{jitter} + \delta \]

\[ T_{\text{Clk}} > t_{c-q} + t_{\text{logic}} + t_{su} - \delta + 2 \ t_{jitter} \]
Impact of Clock Skew and Jitter: Race Margin (Early-Late Problem)

\[ T_{CLK} > t_{c,q} + t_{\text{logic}} + t_{su} + \delta + 2t_{\text{jitter}} \]

- Positive skew improves performance
- Negative skew reduces performance
- Jitter reduces performance

Minimum clock cycle (cycle time)

Minimum logic delay (race)

Notes:
- Absolute delay through a clock distribution path is not important
- What matters is the relative arrival time at the register points at the end of each path

Combined Impact of Clock Skew and Jitter
Overview

- Sequential system timing requirements
- Impact of clock skew on timing
- Impact of clock jitter on timing
- Clock distribution

Dealing with Clock Skew and Jitter

- Balance clock paths (tree distribution)
- Don’t use gated clocks
- Use negative skew to eliminate race conditions (at the cost of performance):
  - Add up the components that result in the time budget - the period must be greater than this value
  \[ T_{CLK} > t_{c-q} + t_{\text{logic}} + t_{su} - \delta \quad (\delta < 0) \]
Clock Distribution

In real systems (such as Xilinx XCS200) clock distribution is an important consideration. Goal: simultaneous clocking of all flip-flops.

Master clock must be buffered through a tree so that slave clocks have same delay. (Done for you in Xilinx FPGAs.)

Clock Distribution

- Distribute clock in a tree fashion
- H-Tree
More Realistic H-Trees

Example: EV6 (Alpha 21264) Clocking
600 MHz – 0.35 micron CMOS

- 2 Phase, with multiple conditional buffered clocks
  - 2.8 nF clock load
  - 40 cm final driver width
- Local clocks can be gated “off” to save power
- Reduced load/skew
- Reduced thermal issues
- Multiple clocks complicate race checking

Global clock waveform

- $t_{rise} = 0.35$ ns
- $t_{cycle} = 1.67$ ns
- $t_{skew} = 50$ ps
Spartan-6 FPGA

- CLB
- I/O
- CMT
- BUFG
- BUFIO
- Block RAM
- DSP48

Spartan-6 FPGA Global Clock Network

- 16 global clock buffers in the Spartan-6 FPGA allow clocks to be distributed to potentially every clocked element on the die
- 16 HCLK lines connect clock signals resources in each row
- HCLK lines can be driven by
  - Global clock buffers
  - DCM outputs
  - PLL outputs
Spartan-6 FPGA I/O Clock Network

- Special clock network dedicated to I/O logical resources
  - Independent of global clock resources
  - Speeds up to 1 GHz
- Multiple sources for clocking I/O logic
  - BUFIO2: for high-speed dedicated I/O clock signals
  - BUFPLL: for clocks driven by the PLL in the CMT

Spartan-6 FPGA Clock Management Tile (CMT)
Digital Clock Manager (DCM)

- Multiply or Divide an Incoming Clock Frequency or synthesize a completely new frequency by a mixture of clock multiplication and division.
- Condition a Clock, ensuring a clean output clock with a 50% duty cycle.
- Phase Shift a clock signal, either by a fixed fraction of a clock period or by precise increments.
- Eliminate Clock Skew, either within the device or to external components, to improve overall system performance and to eliminate clock distribution delays.
- Mirror, Forward, or Rebuffer a Clock Signal, often to deskew and convert the incoming clock signal to a different I/O standard—for example, forwarding and converting an incoming LVTTL clock to LVDS.

Eliminating Clock Skew

[Diagram showing clock signals and FPGA configurations]
Eliminating Clock Skew

Quadrant Phase Shifting
Fine Phase Shifting

Summary

- Clock skew and clock jitter – increasingly important issues with technology downscaling
- CAD tools (e.g., ISE WebPack) take care of many issues automatically
Appendix A: Asynchronous Inputs

Very simple synchronizer circuit:

![Diagram](attachment:image.png)

This works most of the time (failure rate proportional to system clock).
Asynchronous Inputs: Multiple Synchronizers

Sample an asynchronous signal at one place in your circuit. Otherwise, system might see inconsistent values of input.