Announcements

- Homework #8 due next Tuesday
- Project Phase 3 part 1 due this Sun.

Class Material

- Last lecture
  - Latches and flip-flops
- Today’s lecture
  - Timing
- Reading
  - Chapter 7, 10

Synchronous Timing

Latch Parameters

Delays can be different for rising and falling data transitions
Register Parameters

Delays can be different for rising and falling data transitions.

<table>
<thead>
<tr>
<th>D</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>t_{setup}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{hold}</td>
<td></td>
</tr>
</tbody>
</table>

| C | t_{clk-q} |

Clock Uncertainties

Sources of clock uncertainty

Timing Constraints

Cycle time (max): \( T_{\text{CLK}} > t_{\text{clk-q} + t_{\text{logic}} + t_{\text{setup}}} \)

Race margin (min): \( t_{\text{hold}} < t_{\text{clk-q,min} + t_{\text{logic,min}}} \)

Clock Nonidealities

- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, \( t_{\text{SK}} \)

- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) \( t_{\text{JS}} \)
  - Long term \( t_{\text{JL}} \)

- **Variation of the pulse width**
  - Important for level sensitive clocking

Clock Skew and Jitter

- Both skew and jitter affect the effective cycle time
- Only skew affects the race margin (usually)

Positive and Negative Skew

(a) Positive skew

(b) Negative skew
**Positive Skew**

Launching edge arrives before the receiving edge

**Negative Skew**

Receiving edge arrives before the launching edge

**Timing Constraints**

$$t_{clk-q} + t_{logic} + t_{setup} + \delta < t_{CLK} - t_{JS,1} - t_{JS,2} - \delta$$

Minimum cycle time:

$$T_{clk} - \delta = t_{clk-q} + t_{setup} + t_{logic}$$

Worst case is when receiving edge arrives early (positive $$\delta$$)

**Hold time constraint:**

$$t_{clk-q, min} + t_{logic, min} > t_{hold} + \delta$$

Worst case is when receiving edge arrives late

Race between data and clock

**Longest Logic Path in Edge-Triggered Systems**

Latest point of launching

Earliest arrival of next cycle

**Clock Constraints in Edge-Triggered Systems**

If launching edge is late and receiving edge is early, the data will not be too late if:

$$t_{clk-q} + t_{logic} + t_{setup} < T_{CLK} - t_{JS,1} - t_{JS,2} - \delta$$

Minimum cycle time is determined by the maximum delays through the logic

$$t_{clk-q} + t_{logic} + t_{setup} + \delta + 2t_{JS} < T_{CLK}$$

Skew can be either positive or negative
**Shortest Path**

Earliest point of launching

Clk

Clk

Nominal clock edge

Data must not arrive before this time

---

**Latch-Based Clocking**

| Domino logic almost always uses latch-based clocking |

- In a flip-flop based system:
  - Data launches on one rising edge
  - Must arrive before next rising edge
  - If data arrives late, system fails
  - Flip-flops have hard edges
- In a latch-based system:
  - Data can pass through latch while it is transparent
  - Long cycle of logic can borrow time into next cycle
  - As long as each loop finished in one cycle

---

**Clock Constraints in Edge-Triggered Systems**

If launching edge is early and receiving edge is late:

\[
t_{clk-q,min} + t_{logic,min} - t_{JS,1} > t_{hold} + t_{JS,2} + \delta
\]

Minimum logic delay

\[
t_{clk-q,min} + t_{logic,min} > t_{hold} + 2t_{JS} + \delta
\]

(This assumes jitter at launching and receiving clocks are independent – which usually is not true)

---

**Pipelining**

Reference

<table>
<thead>
<tr>
<th>Clock Period</th>
<th>Adder Absolute Value</th>
<th>Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a_1 + b_1)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(a_2 + b_2)</td>
<td>(\log(a_1 + b_1))</td>
</tr>
<tr>
<td>3</td>
<td>(a_3 + b_3)</td>
<td>(\log(a_2 + b_2))</td>
</tr>
<tr>
<td>4</td>
<td>(a_4 + b_4)</td>
<td>(\log(a_3 + b_3))</td>
</tr>
<tr>
<td>5</td>
<td>(a_5 + b_5)</td>
<td>(\log(a_4 + b_4))</td>
</tr>
</tbody>
</table>

| Pipelined |

---

**Time Borrowing Example**
Latch vs. Flip-flop Summary

- Flip-flops generally easier to use
  - Most digital ASICs designed with register-based timing
- But, latches (both pulsed and level-sensitive) allow more flexibility
  - And hence can potentially achieve higher performance
  - Latches can also be made more tolerant of clock uncertainty
  - More in EE241

Next Lecture

- Clock and power distribution