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

Timing
**Synchronous Timing**

![Synchronous Timing Diagram](image)

**Latch Parameters**

![Latch Parameters Diagram](image)

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

Delays can be different for rising and falling data transitions.

Timing Constraints

Cycle time (max): $T_{Clk} > t_{clk-q} + t_{logic} + t_{setup}$

Race margin (min): $t_{hold} < t_{clk-q,min} + t_{logic,min}$
Clock Nonidealities

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

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

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

Clock Uncertainties

Sources of clock uncertainty

- Power Supply
- Interconnect
- Capacitive Load
- Temperature
- Coupling to Adjacent Lines
- Clock Generation
- Device Variation
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

Minimum cycle time:
\[ T_{\text{clk}} - \delta = t_{\text{clock-q}} + t_{\text{setup}} + t_{\text{logic}} \]

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

Hold time constraint:
\[ t_{(\text{clock-q, min})} + t_{(\text{logic, min})} > t_{\text{hold}} + \delta \]

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

$\bar{Clk}$

$t_{clk-q,min}$

$t_{logic,min}$

Nominal clock edge

Data must not arrive before this time

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

<table>
<thead>
<tr>
<th>Clock Period</th>
<th>Adder</th>
<th>Absolute Value</th>
<th>Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a₁ + b₁</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>a₂ + b₂</td>
<td>a₁ + b₁</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a₃ + b₃</td>
<td>a₂ + b₂</td>
<td>log(a₁ + b₁)</td>
</tr>
<tr>
<td>4</td>
<td>a₄ + b₄</td>
<td>a₃ + b₃</td>
<td>log(a₂ + b₂)</td>
</tr>
<tr>
<td>5</td>
<td>a₅ + b₅</td>
<td>a₄ + b₄</td>
<td>log(a₃ + b₃)</td>
</tr>
</tbody>
</table>

### Latch-Based Clocking

(Domino logic almost always uses latch-based clocking)
Latch vs. Flip-flop

- In a flip-flop based system:
  - Data launches on one rising edge
    - And must arrive before next rising edge
  - If data arrives late, system fails
    - If it arrives early, wasting time
  - 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

Time Borrowing Example

![Time Borrowing Diagram]
**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 un-certainty
  - More in EE241

**Next Lecture**

- Clock and power distribution