Agenda

- Latches
- Flip-Flops
- SRAMs
Latches
Latch Timing

- A positive latch is **transparent** \((q = d)\) when the clock is **high** and **opaque** \((q = d, \text{sampled at negedge clock})\) when the clock is **low**
- \(t_{d\rightarrow q}\) : delay from \(d\) to \(q\) when the latch is transparent
- \(t_{\text{clk}\rightarrow q}\) : delay from the rising clock edge to \(d\) propagating to \(q\)

![Latch Timing Diagram]

Note: The diagram shows the timing relationships for a latch, including the high and low clock phases, propagation delays, and the interaction between the clock, input \(D\), and output \(Q\).
Latch Circuits

‘Feedback-breaking’ latch
Transparent high

‘State-forcing’ latch
Transparent low

SR latch
Common interview question
Building a Flip-Flop from Latches

- **Clock pulsed latch**
  - Latch becomes transparent for the pulse duration only, then holds data
  - Not common anymore, sometimes used in high performance circuits
  - Positive hold time

Pair of latches – edge triggered (posedge clk!!)
- Commonly used technique
- L2 holds output data stable when clock is high.
- Negative hold time
Flip-Flops
Hold, Setup, clk->q Time

- This is a negative edge flip-flop as drawn
- We'll consider the positive edge case

- Setup time: data must be stable before the clock edge
  - Violations can be avoided (how?)

- Clk-to-q time: delay from a clock edge to q = d
  - Essentially the delay of 1 latch
Path Timing Constraints

- **Setup constraint:** \( T_{\text{clk}} > t_{\text{clk->q}} + t_{\text{logic,max}} + t_{\text{setup}} \)
  - The clock period must be greater than the delay of the critical path
- **Hold constraint:** \( t_{\text{hold}} < t_{\text{clk->q}} + t_{\text{logic,min}} \)
  - The minimum logic delay must be greater than the hold time
False Paths

- Be careful about finding the critical path by statically adding up delays
- Some paths may not be exercised based on logic expressions
- Here, the critical path is not 400ns. What is it?
Clock Skew

- Skew: the **deterministic clock arrival time difference** between 2 flops
  - 2 flops referred to as launching & receiving
  - Positive = clock to receiving arrives **later** than to launching
  - Negative = clock to receiving arrives **earlier** than to launching
- New timing equations:
  - Setup: $t_{\text{setup}} < t_{\text{clk}} - t_{\text{clk-q}} - t_{\text{logic, max}} + t_{\text{setup}} - t_{\text{skew}}$
    - Positive skew can improve clock frequency
    - Negative skew hurts setup margin
  - Hold: $t_{\text{hold}} + t_{\text{skew}} < t_{\text{clk-q}} + t_{\text{logic, min}}$
    - Skew effect is opposite from setup
Clock Jitter

- Jitter is the **non-deterministic** difference in clock arrival times
  - Types: period & cycle-to-cycle
  - Can be treated like skew in timing calculations
  - Assume worst case jitter in the unfavorable direction for timing calculation
  - Lump jitter of both the launching and receiving FFs into an equivalent skew

"Dual-dirac" model (source)
SRAMs
SRAM Structure:

Static

SD-RAM

Dynamic

Sync

Bit Line(s)

SRAM Cell

Word Line
SRAM Structure:

- **Row Decoder**
  - Inputs: \(A[K], A[K-1], \ldots, A[0]\)
  - Outputs: \(2^{N-K}\) Bit Line(s)

- **Column Decoder**
  - Inputs: \(A[K], A[K-1], \ldots, A[0]\)
  - Outputs: \(M \times 2^K\)

- **Sense Amplifiers/Drivers**
  - Input: \(2^{N-K}\) Bit Line(s)
  - Output: \(M \times 2^K\)

- **SRAM Cell**
  - Connected to Bit Line(s) and Word Lines (\(A[K], A[K-1], \ldots, A[0]\))
SRAM: Basic Static Memory Component:

\[ D \quad \text{pos feedback} \quad \overline{D} \]
SRAM: How to Write?

- **Challenge**: How do you overpower the feedback loop?
- **Challenge**: Writing 0 vs writing 1?
SRAM: How to Write?

- **Challenge**: How do you overpower the feedback loop?
- **Challenge**: Writing 0 vs writing 1?
6T SRAM Cell

enable -
6T SRAM Operation

Write

Hold

Read
6T SRAM Cell Sizing

- Read Sizing:
  \[ M_{6,5} < M_{3,1} \]

- Write Sizing:
  \[ M_{6,5} > M_{2,4} \]
Dual Port SRAM Modifications: 1 Read 1 Write

- What additional logic is needed?
Questions?