Lecture 16

- Timing
  - Terminology
  - Timing issues
  - Asynchronous inputs

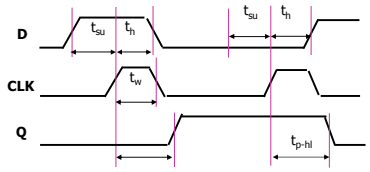
Terminology

- Setup time t_{su}: Amount of time the input must be stable before the clock transitions high (or low for negative-edge triggered FF)
- Hold time t_h: Amount of time the input must be stable after the clock transitions high (or low for negative-edge triggered FF)

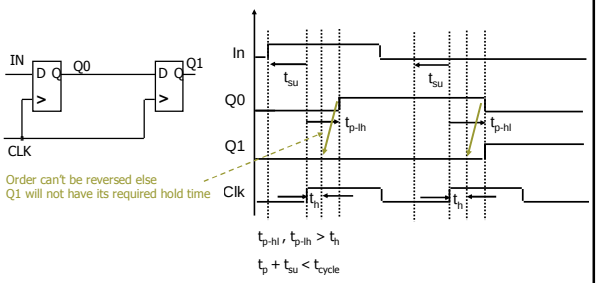
Terminology

- Clock width t_w : Minimum clock width that must be met in order for FF to work properly
- Propagation delays $t_{p\text{-}lh}$ and $t_{p\text{-}hl}$: Delay between clocking event and change in output (high to low, low to high)
  - Longer than hold time

Terminology



Cascading flip-flops



Cascading flip-flops

- Flip-flop propagation delays exceed hold times
  - Second stage commits its input before Q0 changes

System considerations

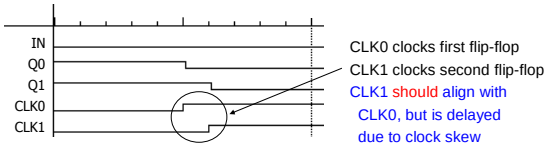
- Basic rules for correct timing
  - Clock flip-flops synchronously (all at the same time)
    - No flip-flop changes state more than once per clock cycle
    - FF propagation delay > hold time
  - Avoid mixing positive-edge triggered and negative-edge triggered flip-flops in the same circuit

System considerations

- Use edge-triggered flip-flops wherever possible
  - Avoid latches
  - Most common: Master-slave D

Clock skew

- Goal: Clock all flip-flops at the same time



Original state: IN = 0, Q0 = 1, Q1 = 1
Next state: Q0 = 0, Q1 = 0 (should be Q1 = 1)

Clock skew

- Difficult to achieve in high-speed systems
  - Clock delays (wire, buffers) are comparable to logic delays
- $T_p > T_h \rightarrow T_p > T_{skew} + T_h$
- If $T_{skew} < 0$, $T_{period} + T_{skew} > T_p + T_{su}$

Asynchronous versus synchronous

- Asynchronous
  - State changes occur when state inputs change
  - Feedback elements may be wires or delays
- Synchronous
  - State changes occur synchronously
  - Feedback elements are clocked

Asynchronous inputs

- Clocked circuits are synchronous
  - Circuit changes state only at clock edges
  - Signals (voltages) settle in-between clock edges
- Unclocked circuits or signals are asynchronous
  - No master clock
  - Real-world inputs (e.g. a keypress) are asynchronous

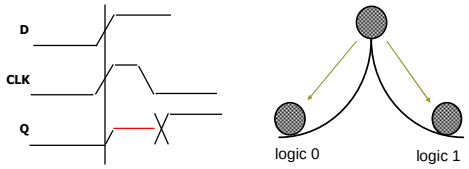
Asynchronous inputs

- Synchronous circuits have asynchronous inputs
  - Reset signal, memory wait, user input, etc.
  - Inputs can change at any time
    - We must synchronize the input to our clock
    - Inputs will violate flip-flop setup/hold times

Synchronizer failure

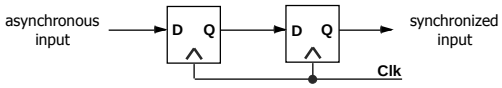
- Occurs when FF input changes near clock edge
  - Input is neither 1 or 0 when clock goes high
  - Output may be neither 0 or 1
    - May stay undefined for a long time
  - Undefined state is called metastability

Synchronizer failure



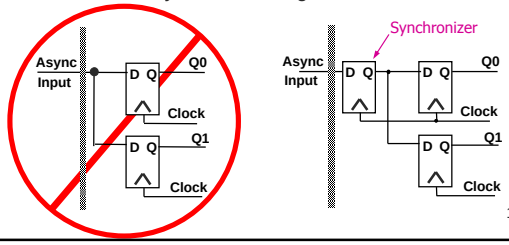
Minimizing synchronizer failure

- Failure probability can never be zero
  - Cascade two (or more) flip-flops
    - Effectively synchronizes twice
    - Both would have to fail for system to fail



Handling asynchronous inputs

- Never fan-out asynchronous inputs
  - Synchronize at circuit boundary
  - Fan-out synchronized signal



Summary

- For sequential logic circuits, timing issues have to be considered.
- Inputs are often asynchronous and can cause problems.
- Different amount of delay at different part of the circuit can cause problems also.
- Solutions:
  - Cascade flip flops in series
  - Design to keep timing alignment in mind (length of wires, etc)