EECS 151/251A Discussion 11 Flip Flops

Daniel Grubb, 11/9, 11/10, 11/15/21

Administrivia

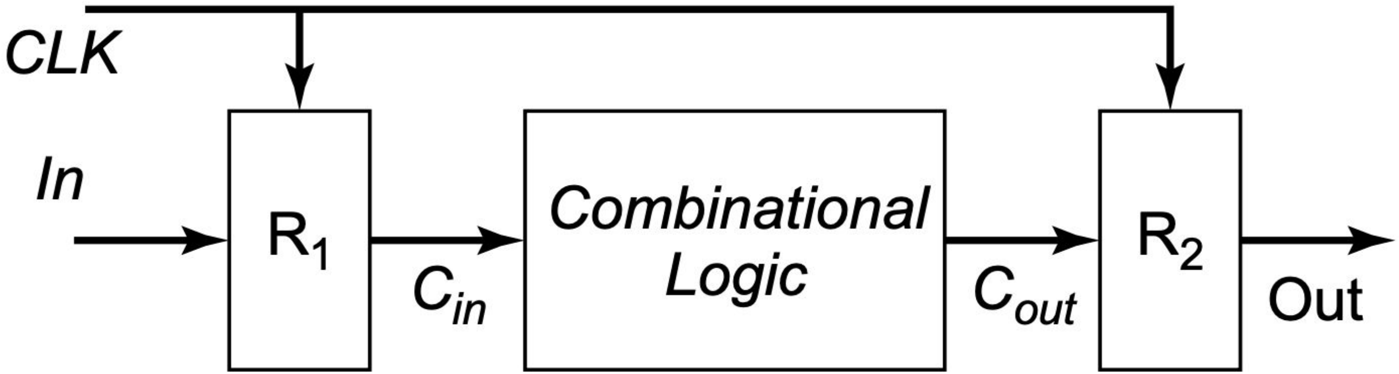
- Midterm 2 scores released
    - Regrades ASAP
- Homework 9 due Monday
- Project checkpoint 2 this week

Outline

- Latches
- Flip flops
- Timing
- Non-idealities

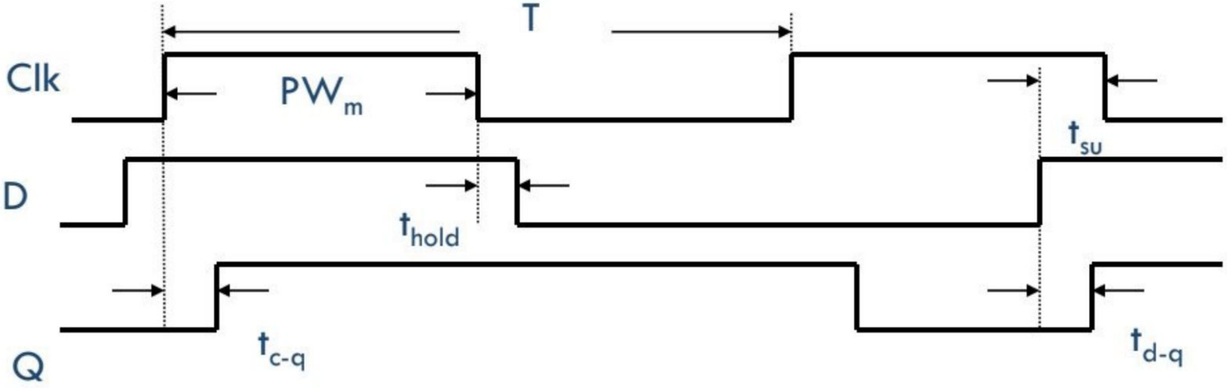
Sequential Logic and Timing

- How do we design synchronous systems?
- How do we store information? Static vs dynamic? Examples?

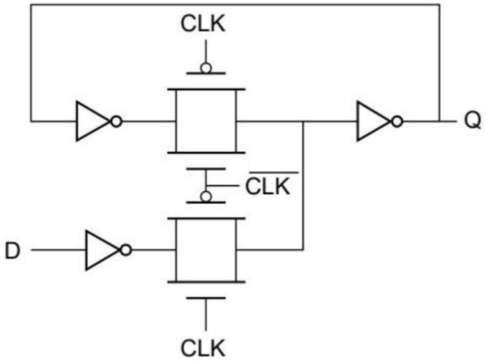


Latch Timing

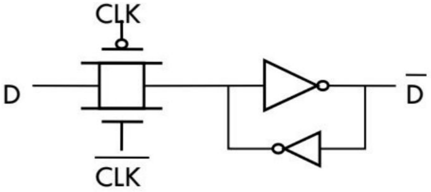
- A latch is a level sensitive storage element
  - Can be **positive** or **negative**
- Positive latch is transparent when clock is high, opaque when clock is low
  - Timing relative to the **falling edge** of the clock
  - Transparent: q=d, sampled value held when latch is opaque
  - $T_{clk\text{-}q}$: time from rising edge of clock to output change
  - $T_{d\text{-}q}$: time from input to output when latch is transparent



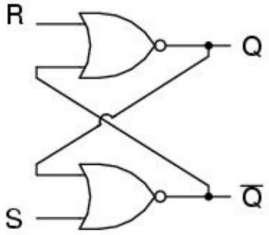
Latch Types

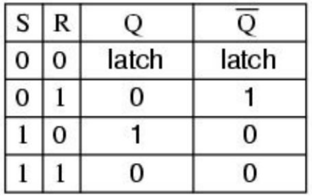


Feedback-breaking latch



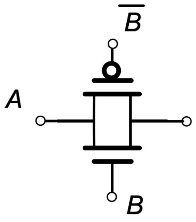
State-forcing latch



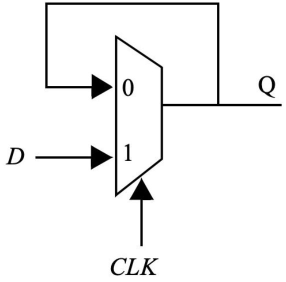


S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

SR latch



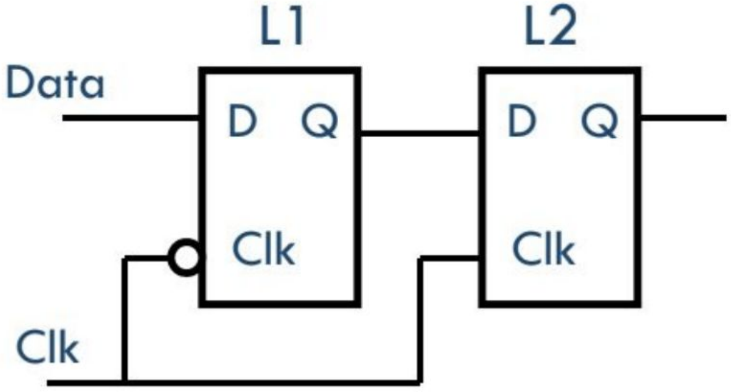
Transmission gate



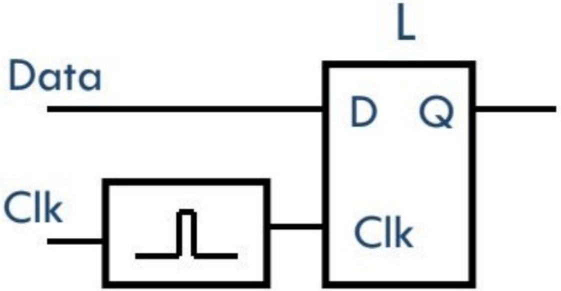
Positive Latch

Flip Flops

- Single bit registers

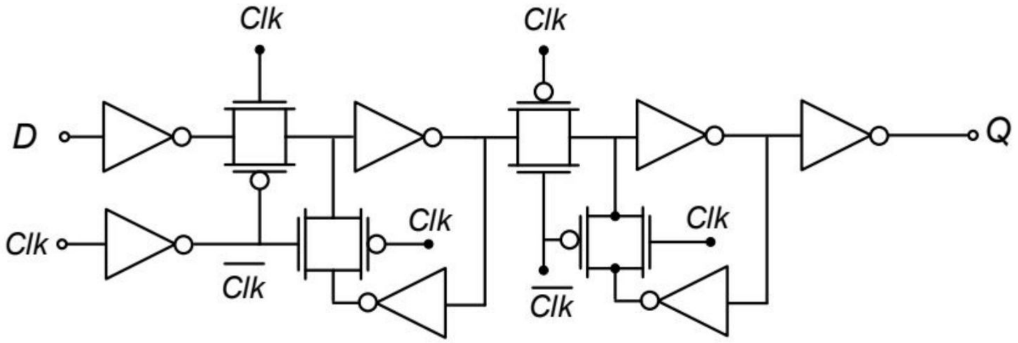


- Two latches
  - Commonly used
  - L2 holds output stable when clock is high
  - Negative hold time



- Pulsed-clock latch
  - Latch becomes transparent for the pulse duration, then holds data
  - Not common anymore
  - Positive hold time
  - How to generate the pulse?

Hold, Setup, Clk-q Times

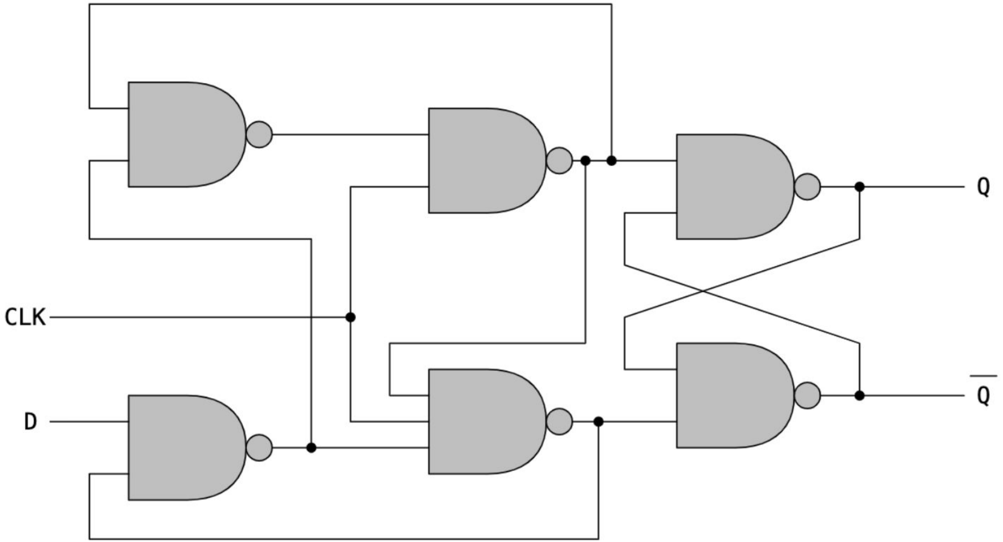


How is this FF implemented?
Positive or negative FF?

- Hold time: data must be stable **after the clock edge**
  - Violations are mostly fatal!
  - Can be negative

- Setup time: data must be stable **before the clock edge**
  - Violations can be worked around
- Clk-q time: delay from a clock edge to q=d
  - Essentially the delay of one latch

Example

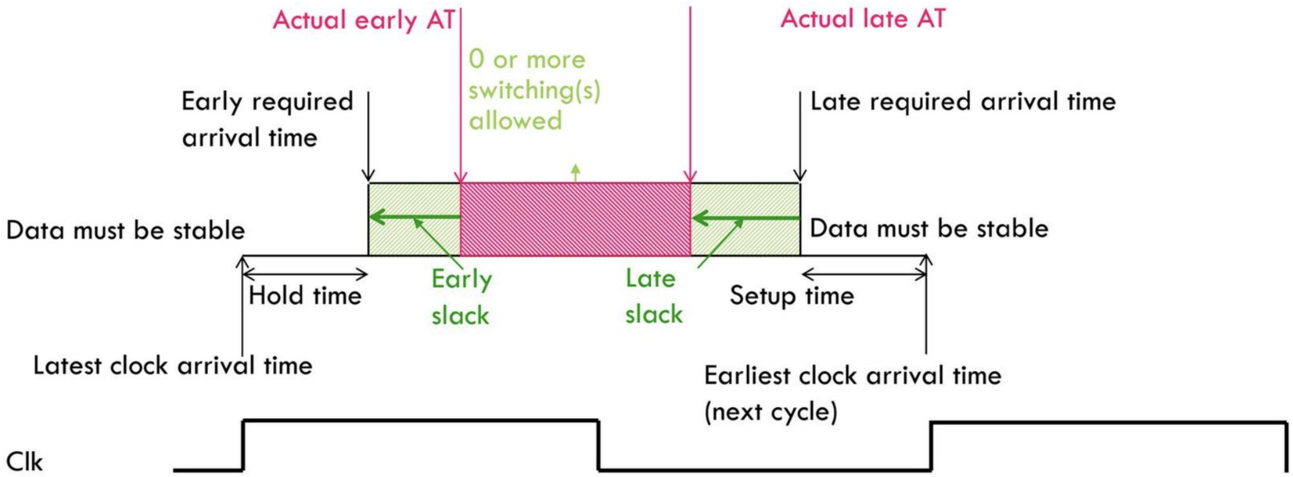


a) What type of circuit is this (flip flop, latch, combinational, other)?

b) If the circuit is a flip flop, is it a positive or negative edge triggered flip flop? If the circuit is a latch, is it a transparent-high or transparent-low latch?

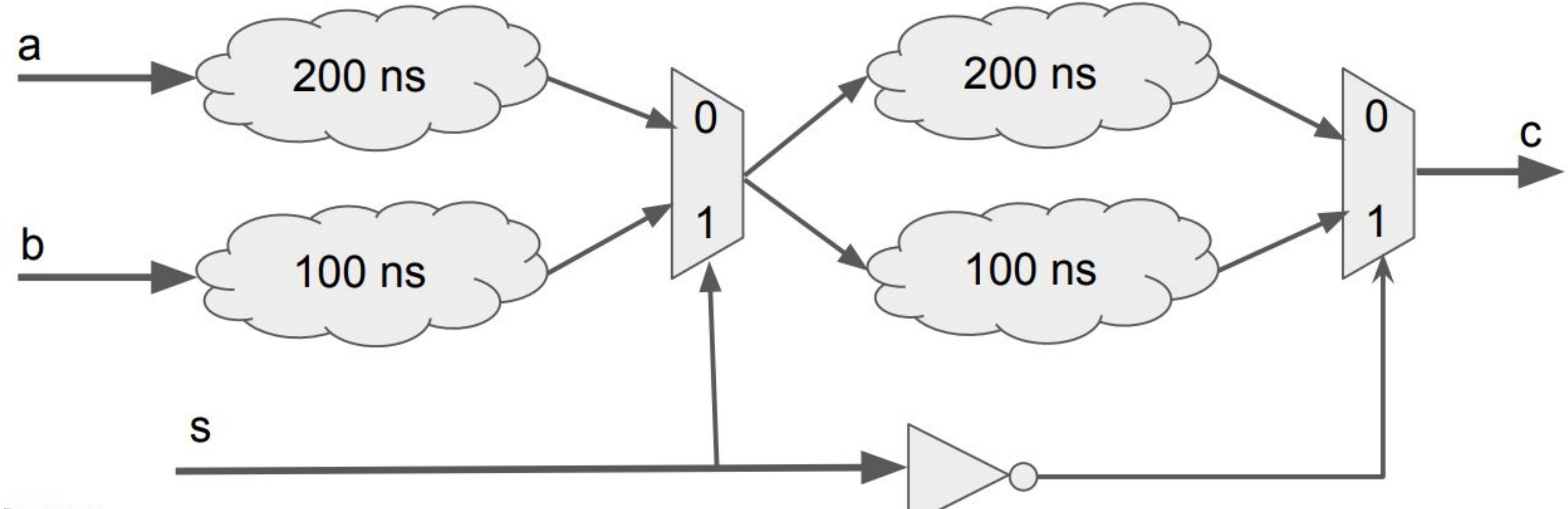
Timing Constraints

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


Actual early AT
Actual late AT
0 or more switching(s) allowed
Early required arrival time
Late required arrival time
Data must be stable
Data must be stable
Hold time
Early slack
Late slack
Setup time
Latest clock arrival time
Earliest clock arrival time (next cycle)
Clk

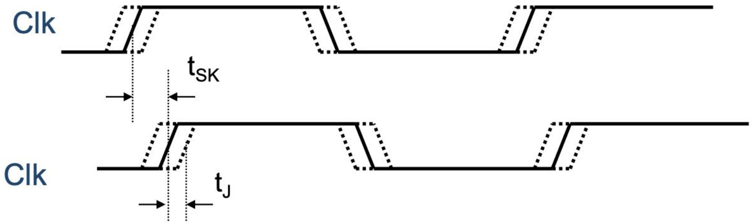
False Paths

- Be careful about finding the critical path by statically adding up delays
- Some paths may not be exercised depending on the logic structure
- What is the critical path in this circuit?



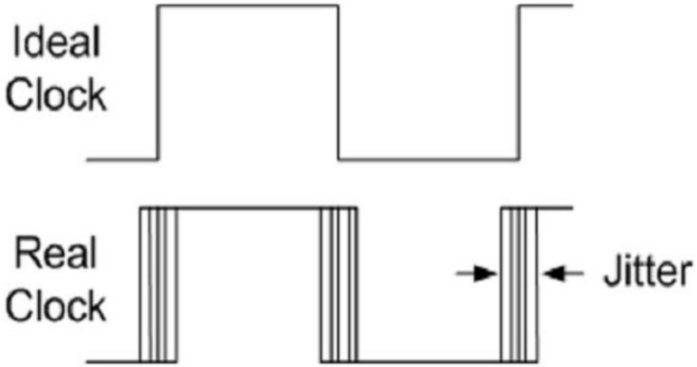
Clock Skew

- Skew: the **deterministic** clock arrival time difference between two flops
  - Launching flop -> receiving flop
  - Positive skew implies receiving clock arrives later than to the launching
    - That path has more time
  - Negative skew implies receiving clock arrives earlier than to the launching
    - That path has less time
- Updated equations
  - Setup: $T_{clk} > t_{clk\text{-}q} + t_{logic,max} + t_{setup} - t_{skew}$
    - Positive skew -> more time to propagate
  - Hold: $t_{hold} + t_{skew} < t_{clk\text{-}q} + t_{logic,min}$
    - Skew effect is opposite to the effect on setup



Clock Jitter

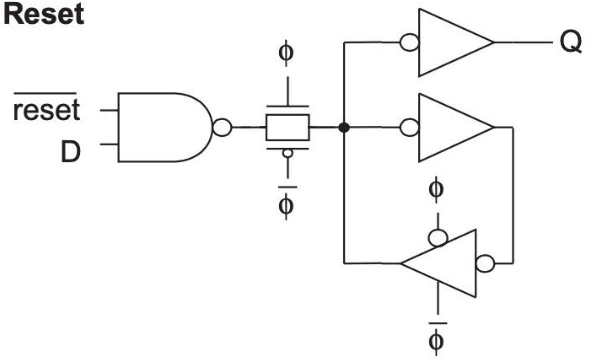
- Jitter: the **non-deterministic** difference in clock arrival times
  - Types: period & cycle-to-cycle
  - Treated similarly to skew in timing calculations
    - Assume worst case jitter in the unfavorable direction for timing calculation
  - Lump jitter of both the launching and receiving FFs into an equivalent skew
  - New equation?



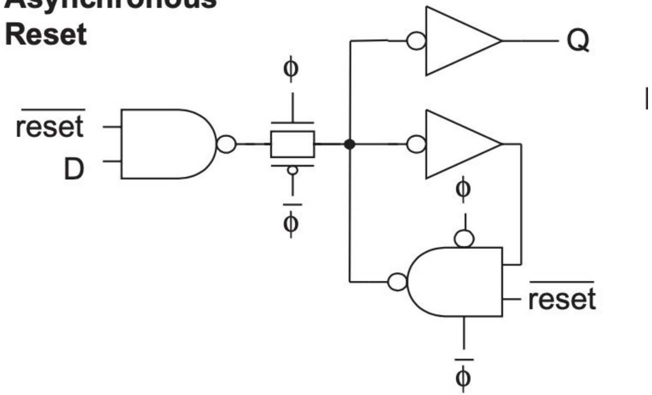
Reset

- Why do we care about reset?
- Two main reset strategies
  - Synchronous: reset can only occur when the clock is toggling
  - Asynchronous: reset can occur without the clock toggling
    - Reset should still be deasserted synchronously
- FF overhead? Distributing reset?

Synchronous Reset

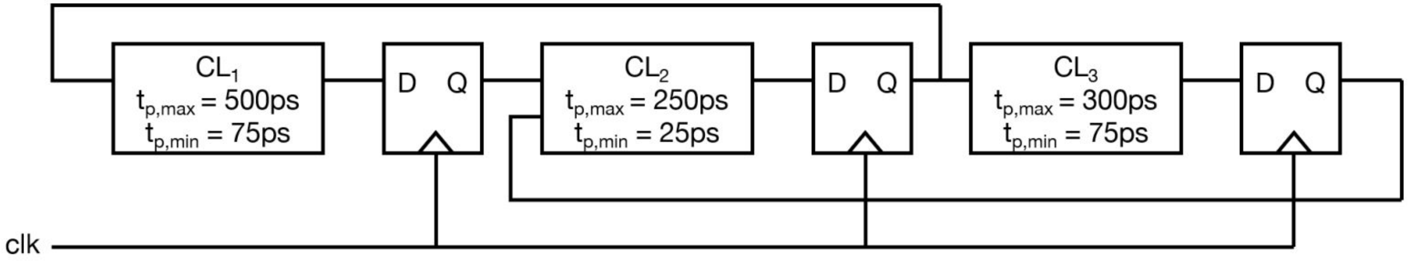


Asynchronous Reset



Timing Example I

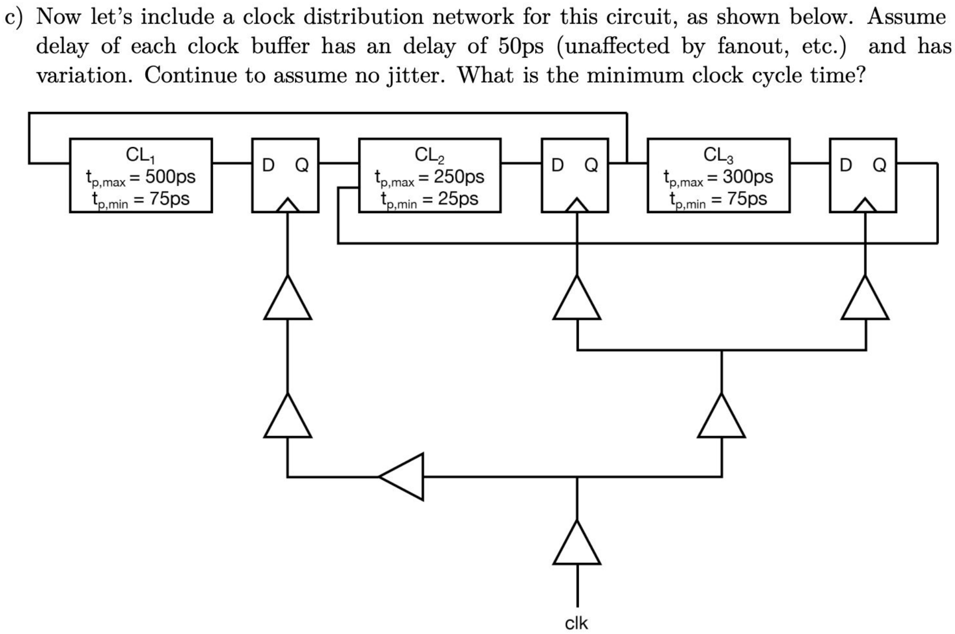
Consider the following sequential circuit. The minimum and maximum logic delays are annotated on the figure. The flip-flops have the following properties: $t_{clk-q} = 50ps$, $t_{setup} = 50ps$, and $t_{hold} = 25ps$.



- T_{min}? Meets hold?

Timing Example I

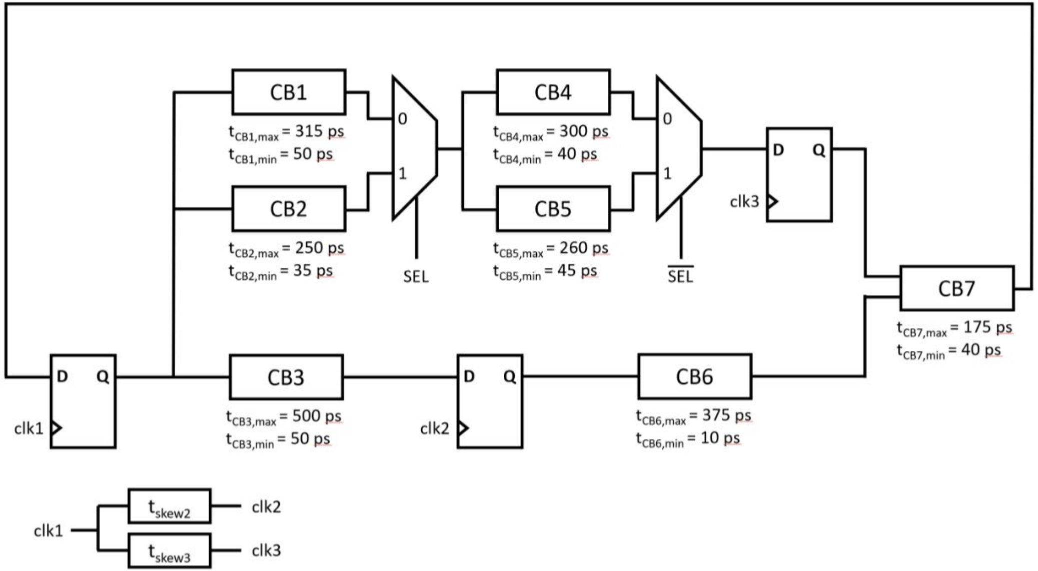
- Each buffer: 50ps
- T_{min}? Meets hold?
- What if we add 50ps of jitter?



c) Now let's include a clock distribution network for this circuit, as shown below. Assume the delay of each clock buffer has an delay of 50ps (unaffected by fanout, etc.) and has no variation. Continue to assume no jitter. What is the minimum clock cycle time?

Timing Example II

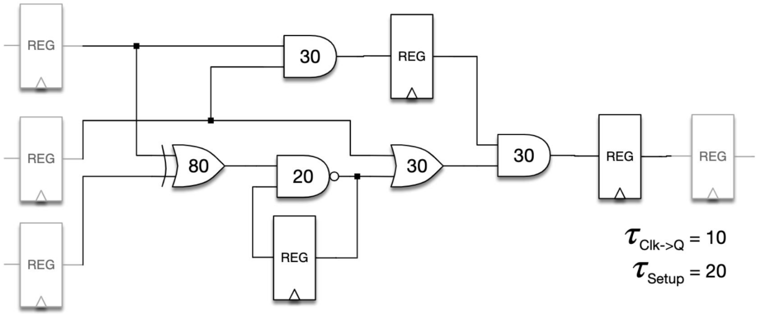
In this problem, you are asked to perform setup and hold timing analyses of the given circuit. Each flip flop is identical with a clock-to-Q delay of $t_{c-q} = 50ps$, setup time of $t_{setup} = 75ps$ and a hold time of $t_{hold} = 85ps$. The signal SEL (and $\overline{SEL}$) is an input to the circuit, and while its logic value is unknown to you, assume that it is constant during normal operation.



(a) Given that there is no skew between the clock signals, what is the minimum clock period this circuit can operate with?

Timing Example III

(a) Without modifying the circuit, derive the maximum clock frequency.

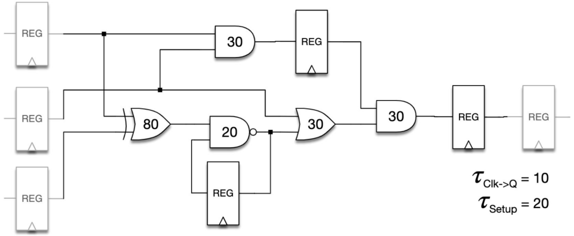


$\tau_{\text{Clk->Q}} = 10$

$\tau_{\text{Setup}} = 20$

Timing Example III

(b) Now retime the circuit (without changing the latency between any input and the output) to maximize the clock frequency. Assume that you cannot move the gray registers on the border of the circuit. Draw the new circuit and derive the new maximum clock frequency.



$\tau_{\text{Clk->Q}} = 10$

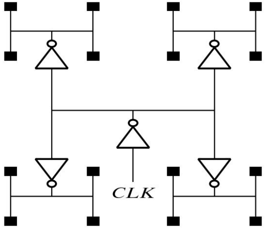
$\tau_{\text{Setup}} = 20$

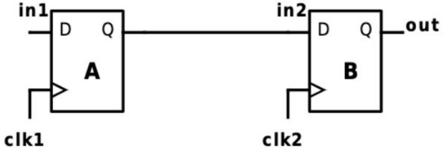
Latch Timing

- FF timing
  - Data launches on rising edge and must arrive before next rising edge
  - The system fails if it does not arrive on time
- Latch timing
  - Data can pass through a latch while it is transparent
    - More flexibility
  - Short paths can “pass slack” and long paths can “borrow time”
  - More in EE241B
- FFs are easier to use, but latches can achieve better performance and have more flexibility

Practical Clock Considerations

- The clock is distributed across the chip using a clock tree + related circuits
  - Clock tree fans out to all clocked element and uses buffers
  - Tools try to balance paths from the source to leaf nodes
  - Different strategies including ad-hoc, meshes, h-trees, etc. depending on needs
  - Clock tree affects skew+jitter, costs power and area
- PLLs/DLLs can be used to sync internal clock
  - Help regulate clock in presence of noise, temperature change, etc.
- Designers give tools SDCs (ASIC) to constrain timing
  - Defining clocks, setting false paths, delays, uncertainty, etc.
- Clock gating to reduce dynamic power (how?)
- Multiple clock domains
  - Need to synchronize between domains, synchronize resets
  - Single bit synchronizers, async queues for multi-bit domain crossings
- Lots of practical effects in a real tapeout!
  - Common interview questions too!




in1
D
Q
A
clk1
in2
D
Q
B
out
clk2