EE-421: Digital System Design

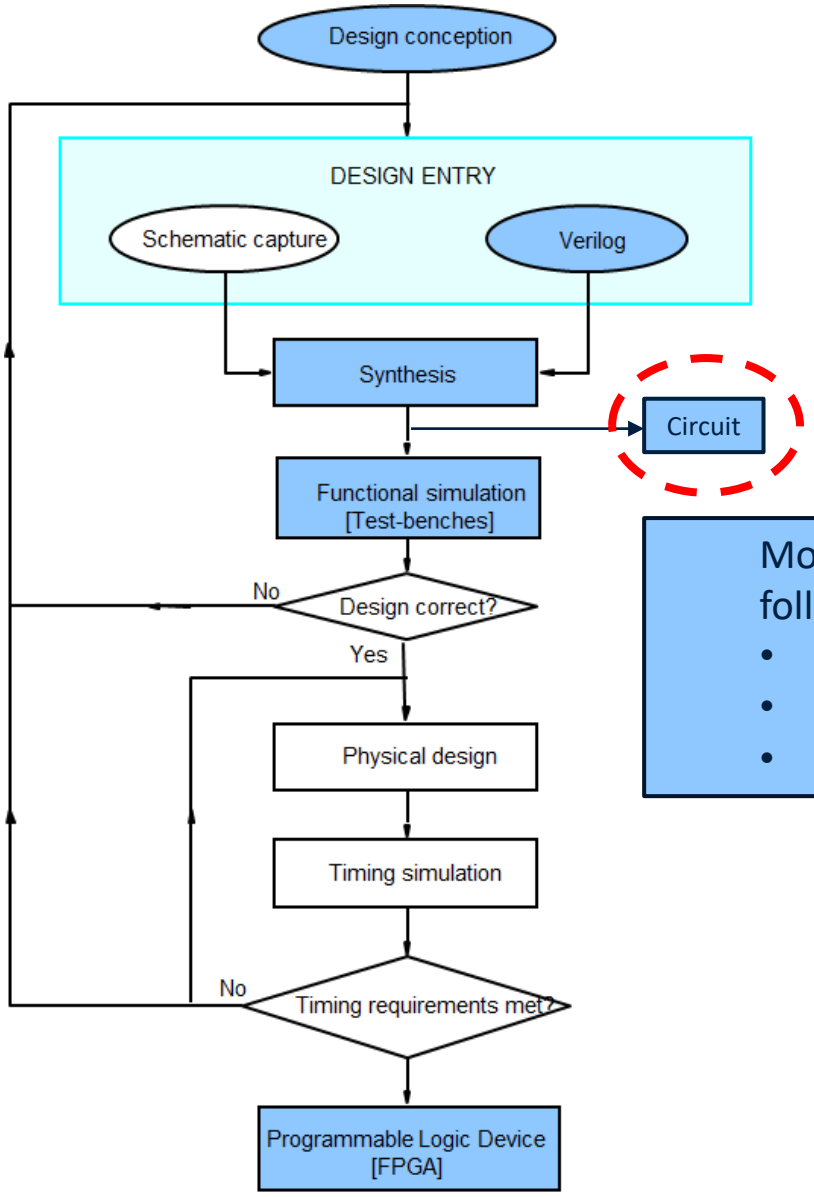
Circuit Timing

Instructor: Dr. Rehan Ahmed [rehan.ahmed@seecs.edu.pk]

NUST

SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

Where are we Heading?



More specifically, we'll focus on the following circuit attributes:

- Combinational Timing Constraints
- Circuit Speed
- Flip-flop Timing Constrains

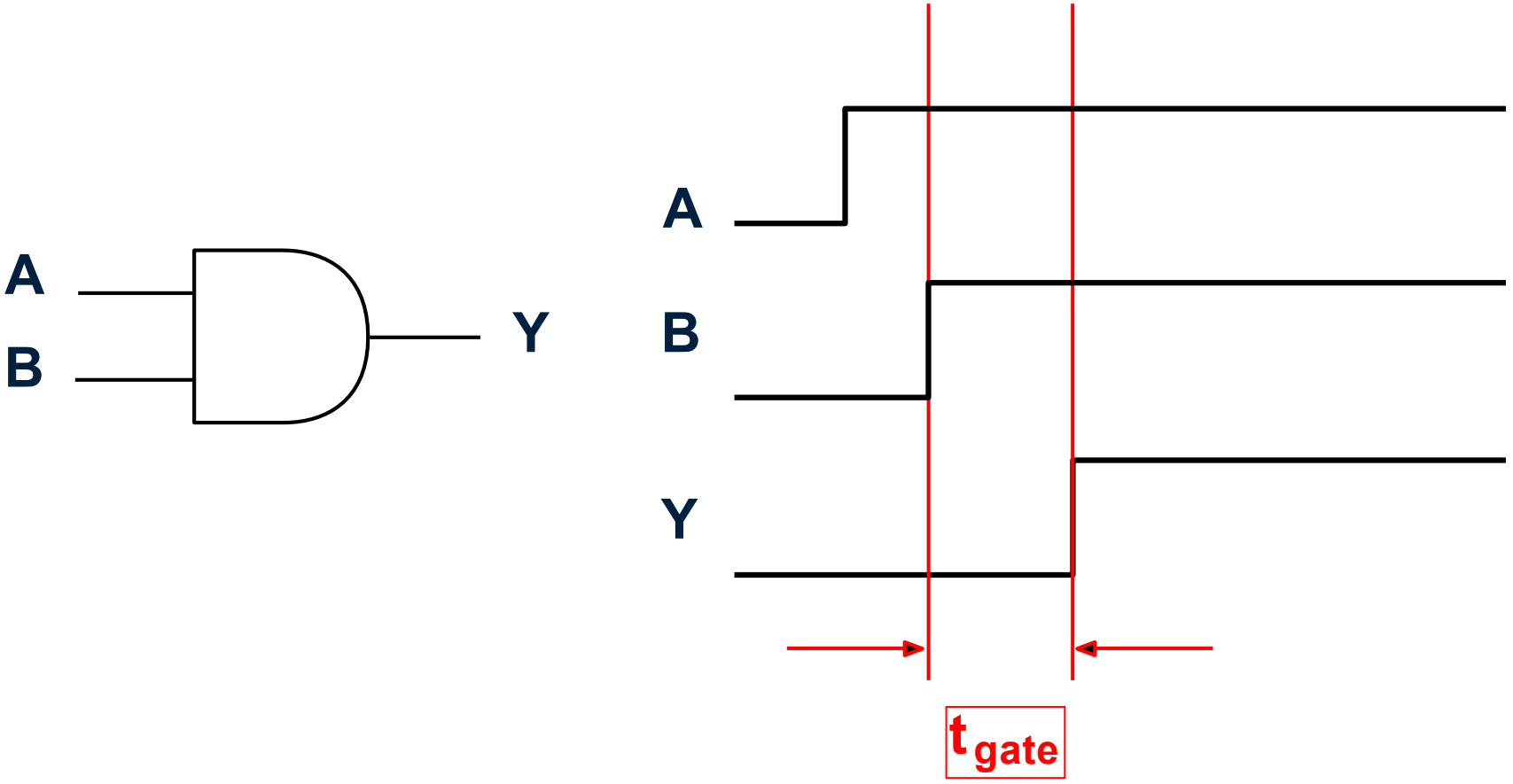
This Lesson: Timing Concepts and Terminology

- Combinational Timing Constraints:
  - Gate Propagation Delay
  - Critical Path Delay
- Circuit Speed:
  - Minimum Clock Period
  - Maximum Clock Frequency
- Flip-Flop Timing Constraints:
  - Clock-to-Q Delay
  - Setup Time
  - Hold Time

Propagation Delay in Gates

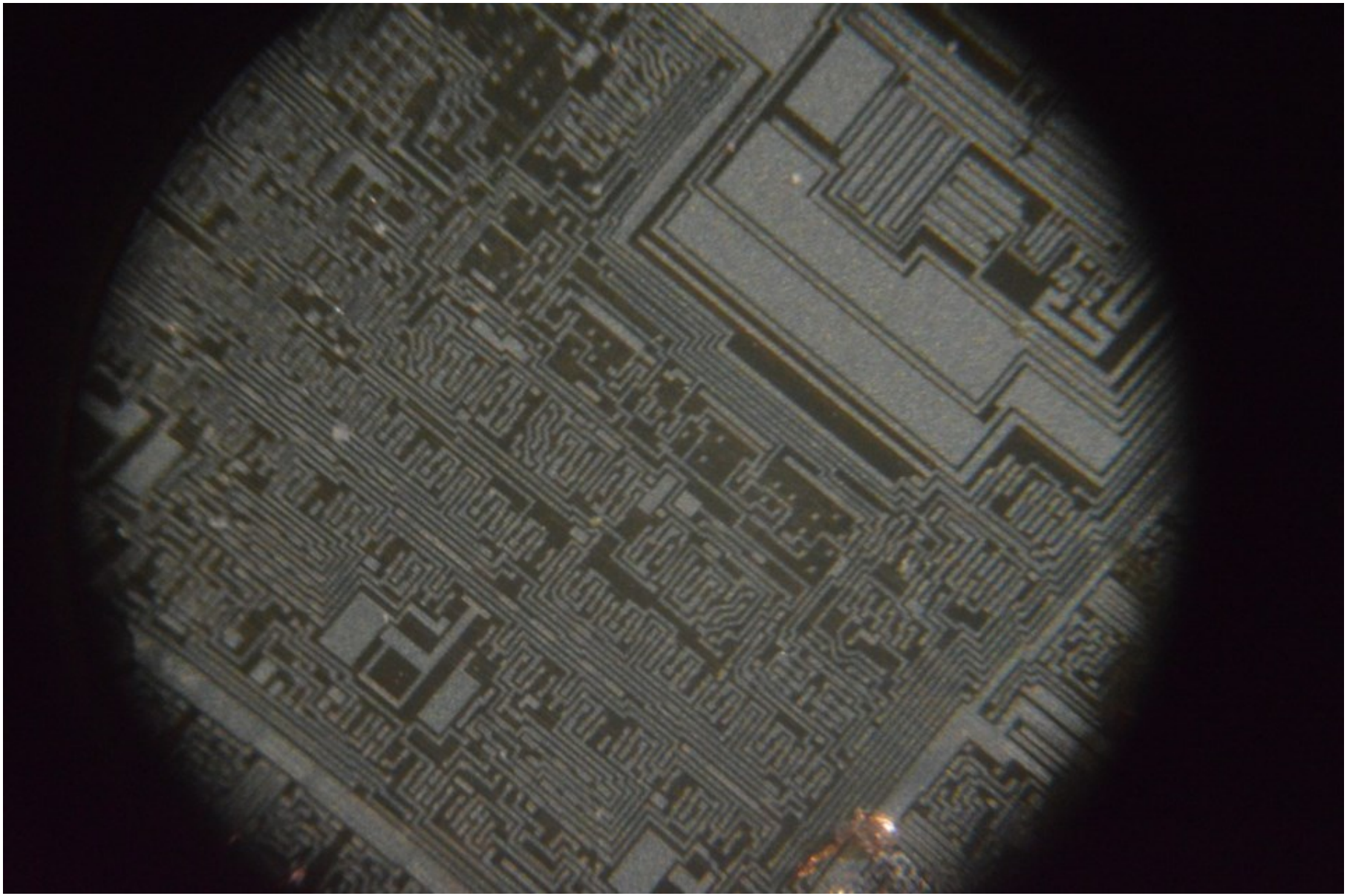
Gate Delay (Propagation Delay)

- Time that it takes for combinational gate output to change after inputs change



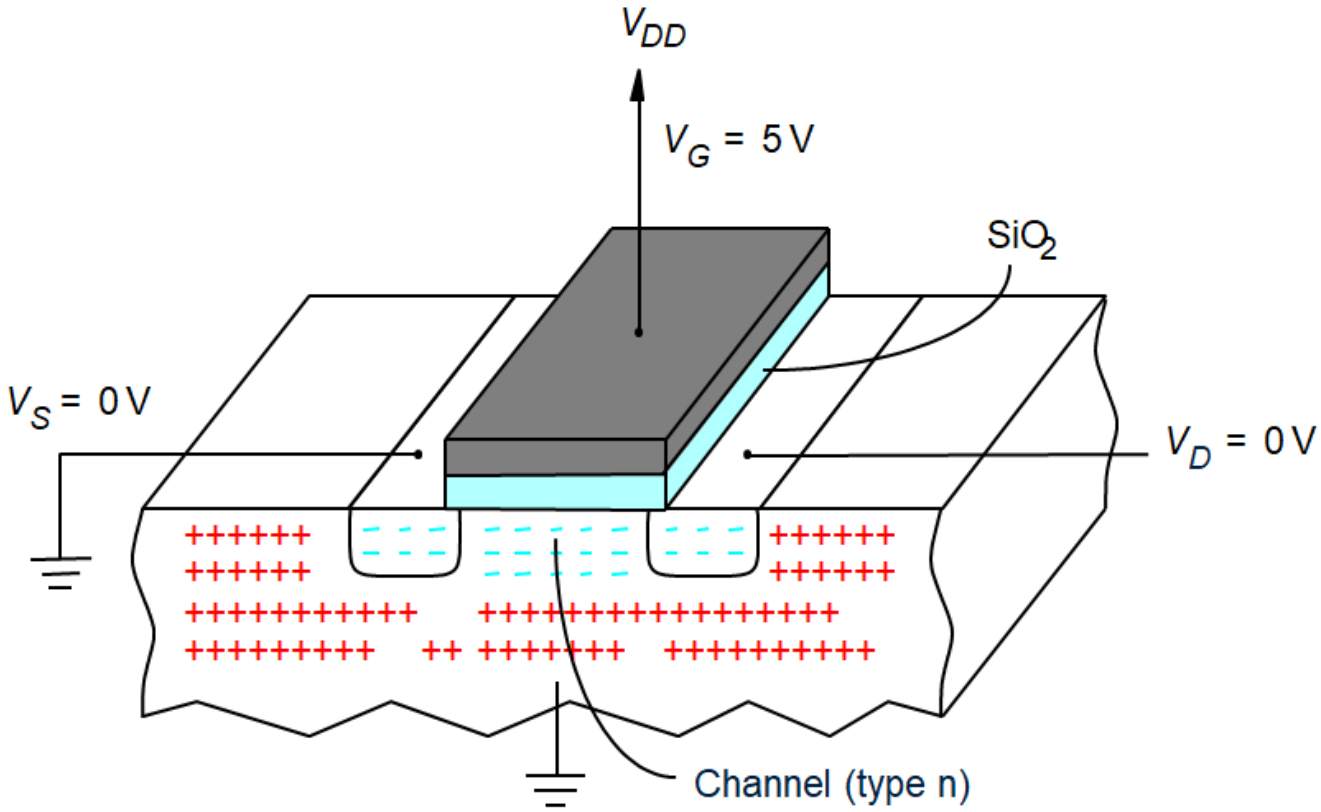
Propagation Delay in Gates: But WHY?

Inside a Chip: Microscopic View



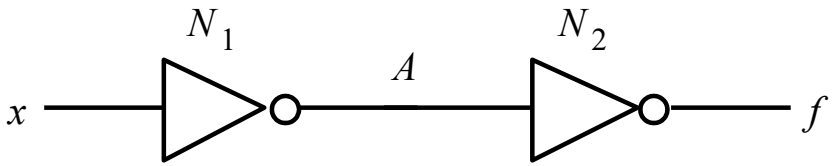
Courtesy of Shaheer Sajid (BEE-9)

Physical Structure of an NMOS Transistor

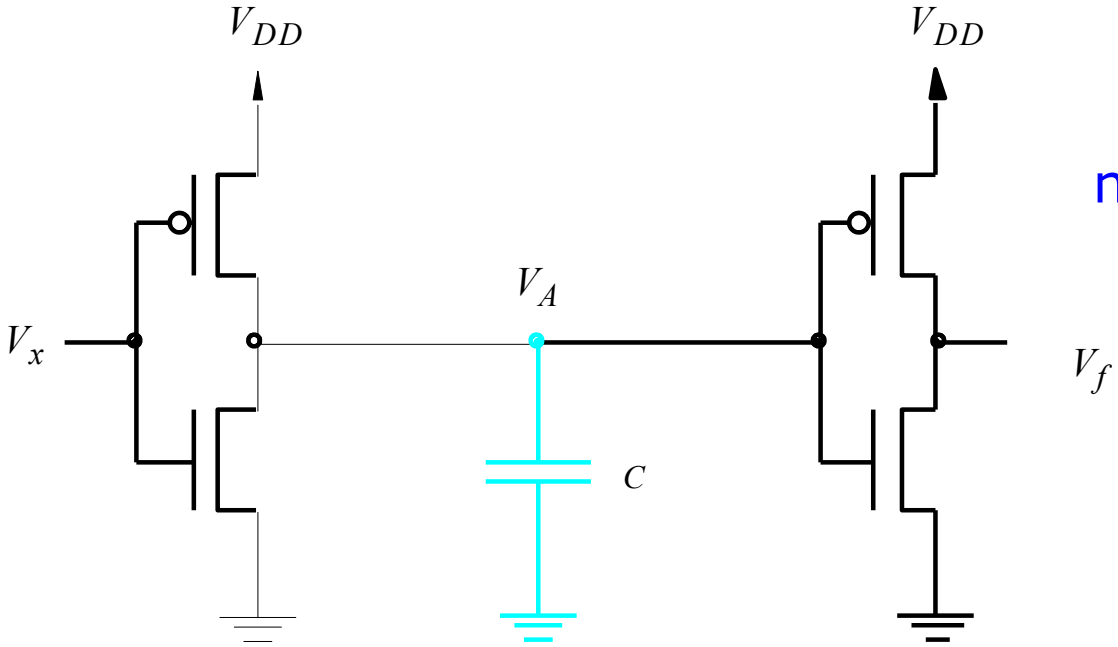


- Wherever two types of materials meet or overlap inside the transistor, a *capacitor* is effectively created:
  - aka, *parasitic* or *stray* capacitance
    - Results as an undesired side effect of a transistor fabrication

Stray Capacitance in Logic Gates



(a) A NOT gate driving another NOT gate

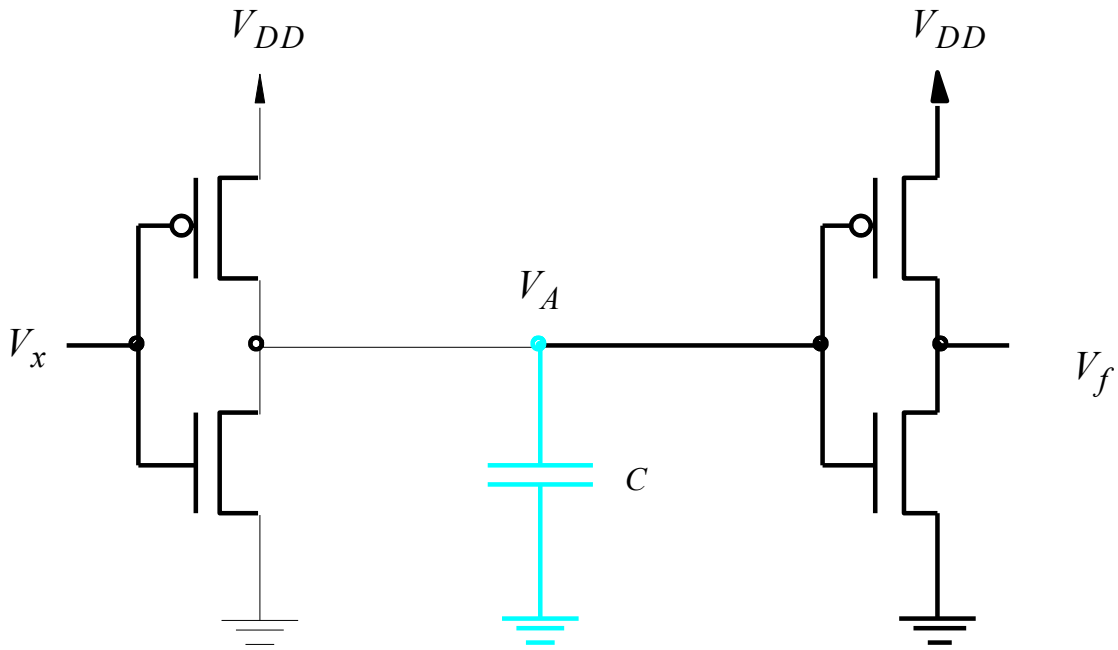


(b) The capacitive load at node A

A number of parasitic capacitors are attached to node A, some caused by N1 and others caused by N2.

All capacitances are approximated by a single equivalent capacitance.

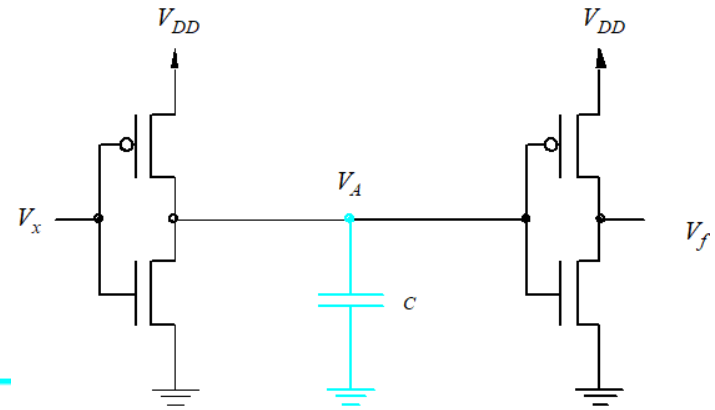
Impact of Stray Capacitance on Speed of Operation (1)

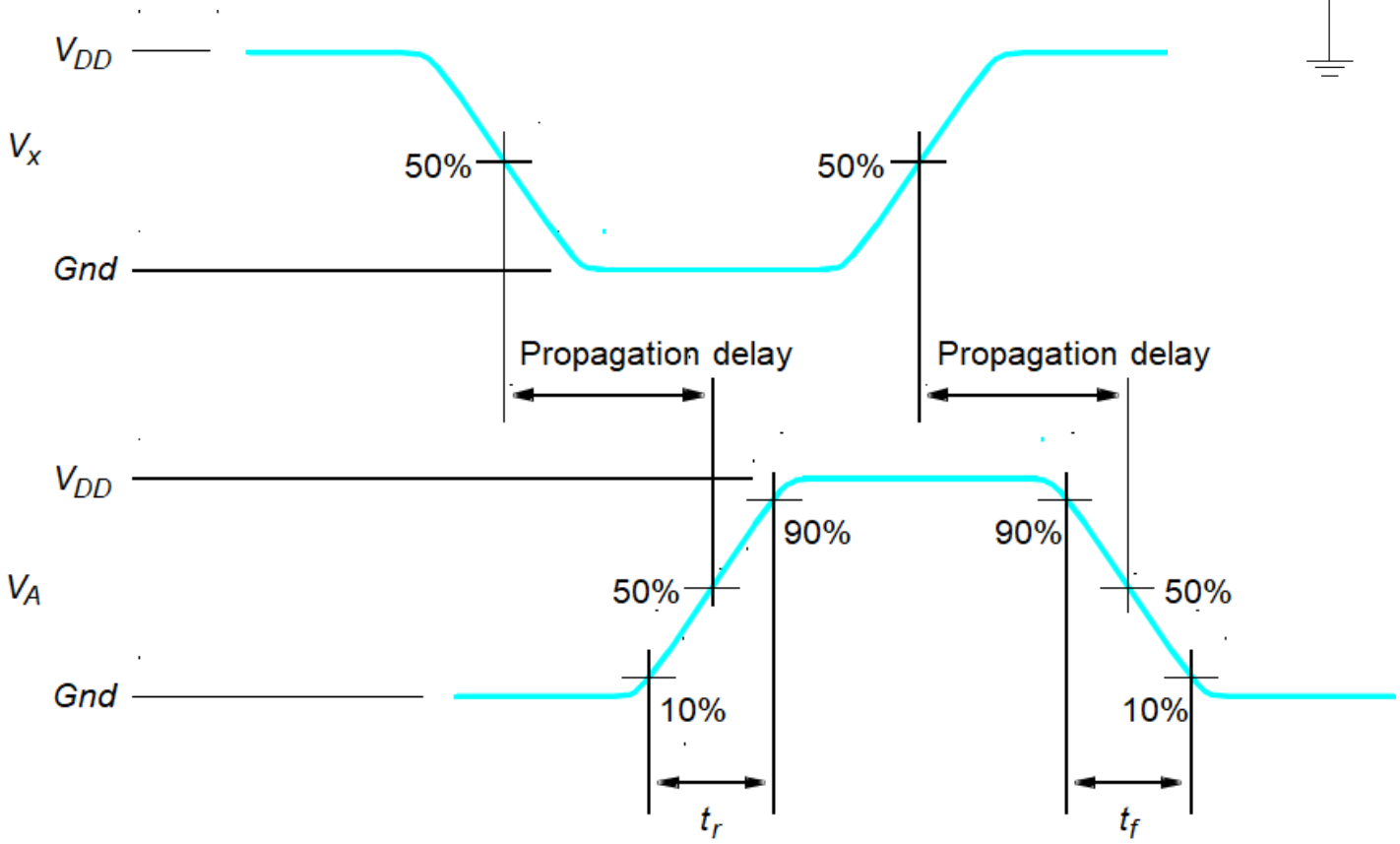


- The existence of stray capacitance has a negative effect on the speed of operation of logic circuits.

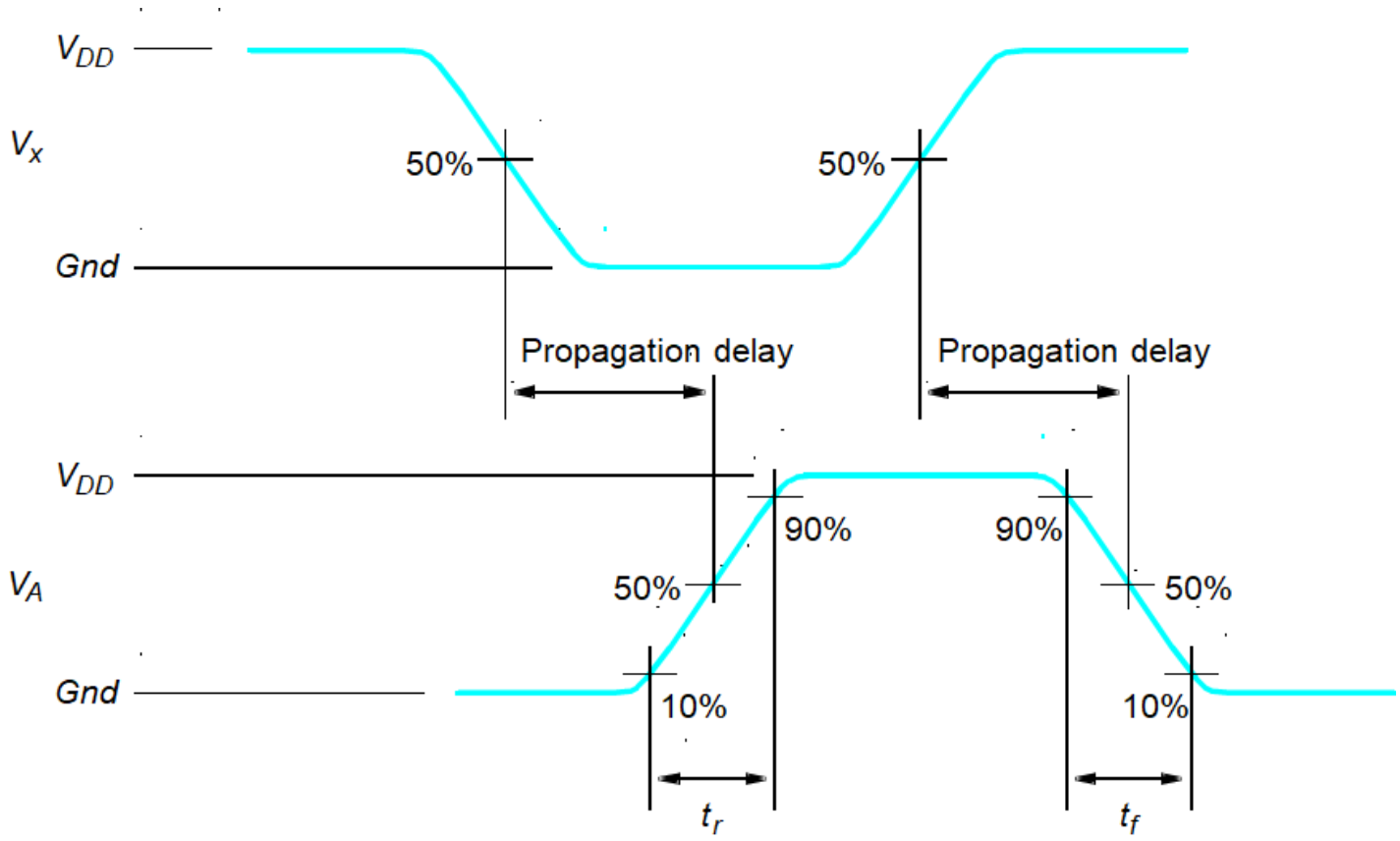
- Voltage across a capacitor cannot change instantaneously,
  - The time needed to charge or discharge a capacitor depends on the size of the capacitance C and on the amount of current through the capacitor.

VTC: Voltage Transfer Characteristics



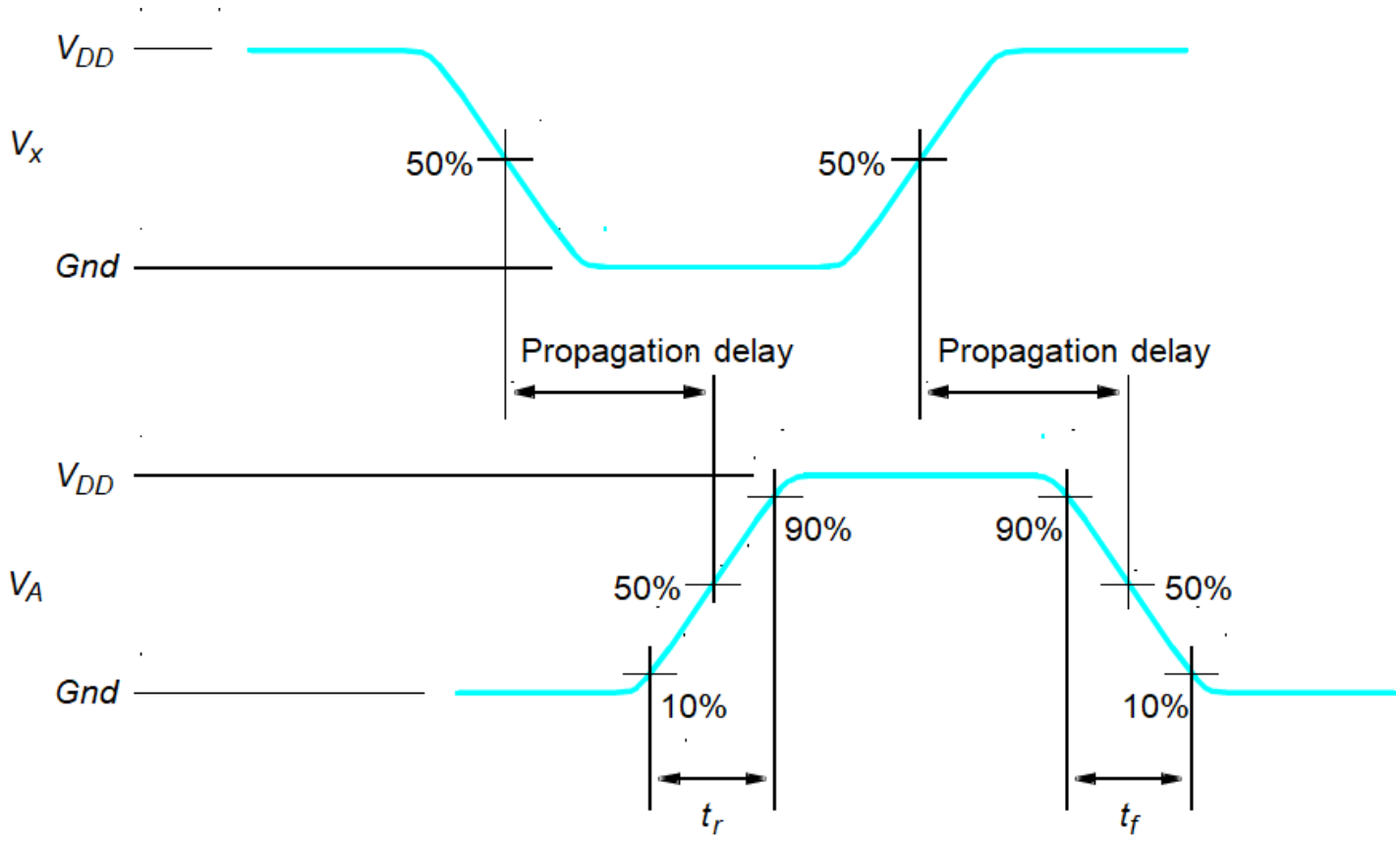


Rise Time



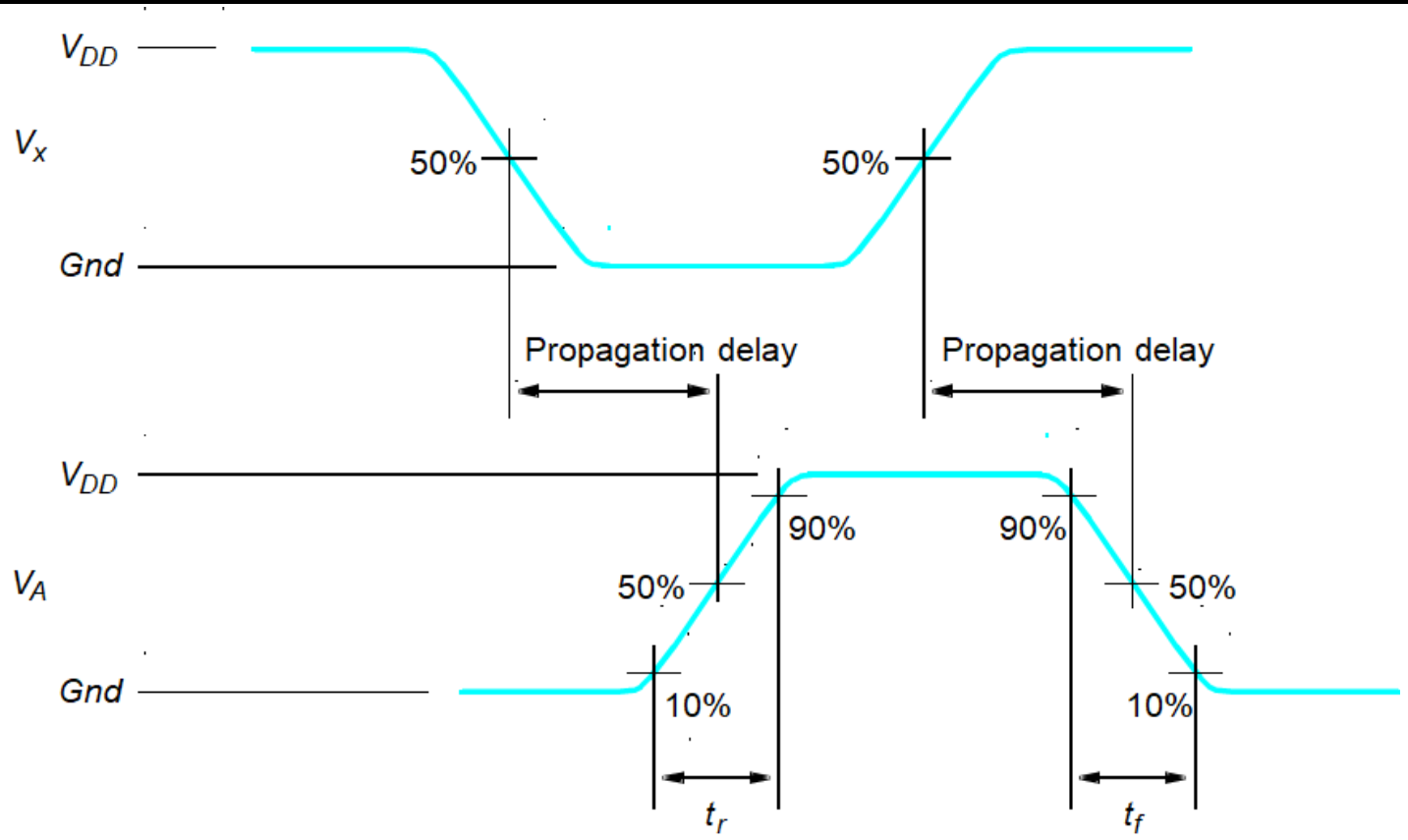
- The time needed for V_A to change from *low to high* is called the *rise time*
  - i.e time elapsed from when V_A is at 10% of V_{DD} until it reaches 90% of V_{DD}

Fall Time



- The time needed for V_A to change from *high to low* is called the *fall time*
  - i.e time elapsed from when V_A is at 90% of V_{DD} until it reaches 10% of V_{DD}

Propagation Delay – (1)



- The total amount of time needed for the change at Vx to cause a change in VA:
  - This interval is known as the *propagation delay* (*tp*)
  - i.e the time from when Vx reaches 50% of VDD until the time VA reaches the same level

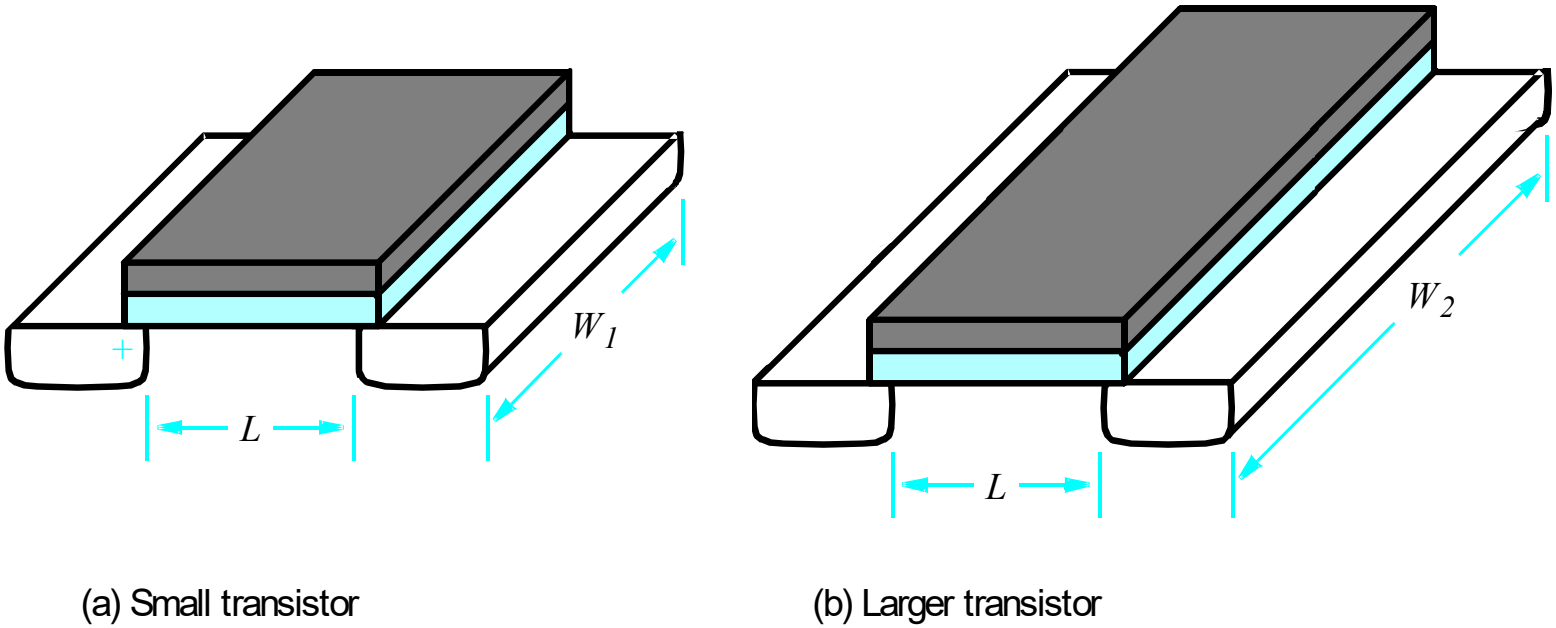
Propagation Delay – (2)

- Propagation delay is given as:

$$t_p \cong \frac{1.7\,C}{k_n' \frac{W}{L} V_{DD}}$$

- This expression specifies that speed of the circuit depends on:
  - C
  - W/L (dimension of the transistor or transistor size)

Transistor Sizes



(a) Small transistor

(b) Larger transistor

- In logic circuits,
  - *L* is usually set to the minimum value that is permitted according to the specifications of the fabrication technology used (technology node i.e 180nm, 65nm … 16nm, 7nm)
  - The value of *W* is chosen depending on the amount of current flow, hence propagation delay, that is desired

FPGA Speed Grade

- FPGA speed grade indicates the device speed:
  - Check the vendor documentation on speed grade
  - For example: Intel MAX and Classic devices use the speed grade to indicate the delay in nanoseconds (ns) through a macrocell in the device.
    - For example, a MAX device with a -10 speed grade has a delay of 10 ns through a macrocell.
  - Devices with low speed grade numbers run faster than devices with high speed grade numbers.
    - While true in general, again check the vendor documentation.

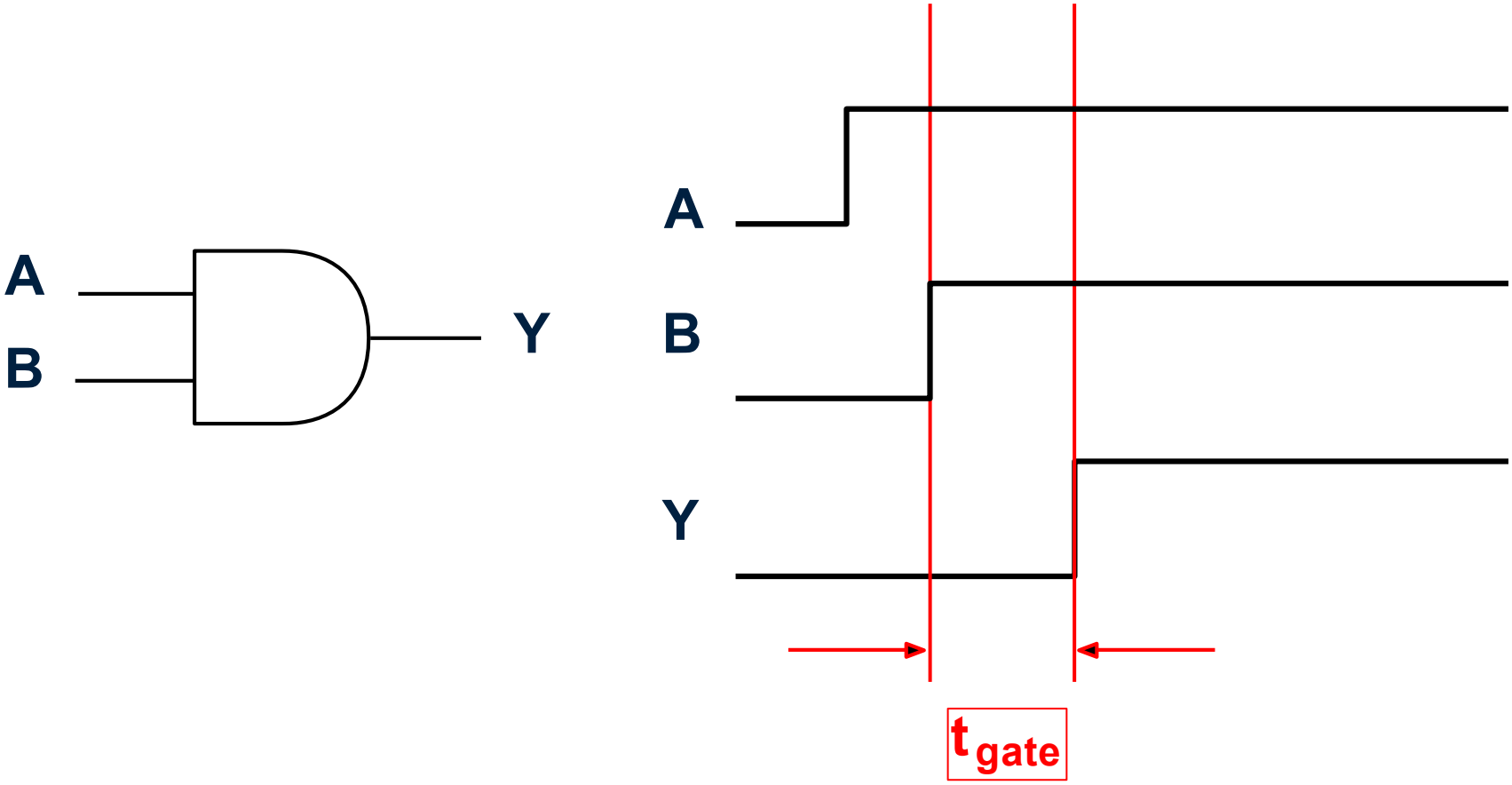
Sample Code: EPF10K130EQC240-1X

EPF 10K130E	Q	C	240	-1	X
FLEX 10K130E	Quad flat pack	Commercial range	240 pins	-1 speed grade	Has phase-locked loops (PLLs)

Path Delay

Recall: Gate Delay (Propagation Delay)

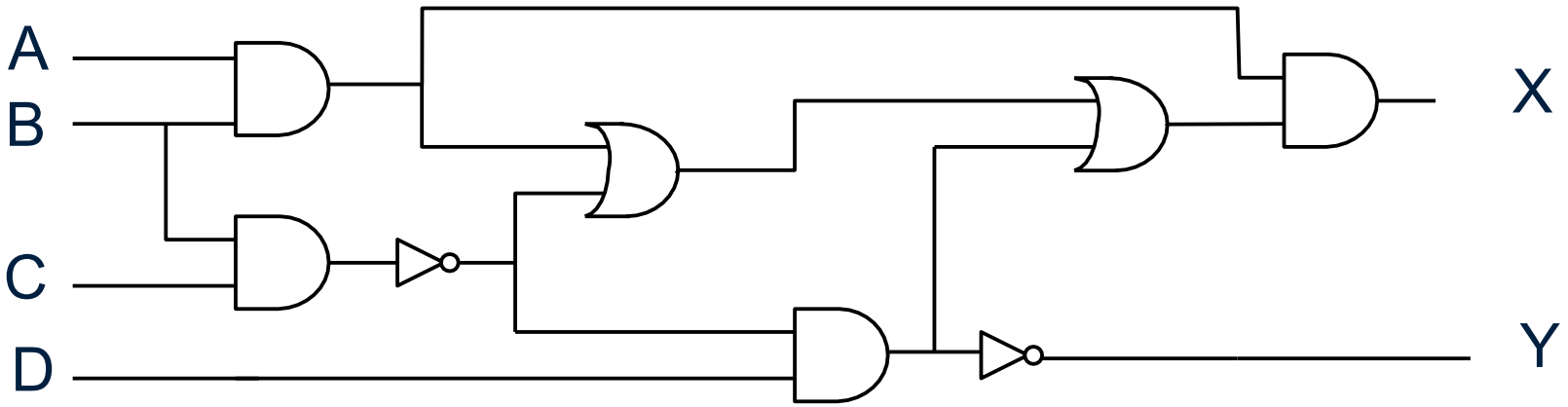
- Time that it takes for combinational gate output to change after inputs change



Path Delay

- Delay through a series of combinational gates:
  - Specifically, the time it takes for the output of the series of gates to change after the inputs to the path change.

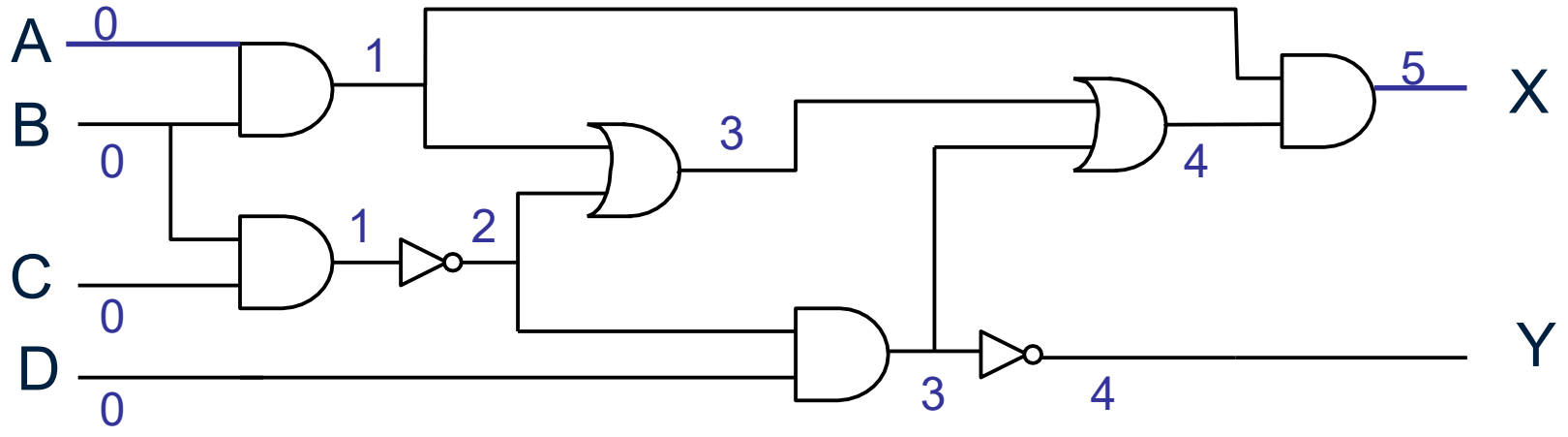
- Example: What is the path delay in the following cct?
  - Assume propagation delay for each gate is the same (1ns in this example)



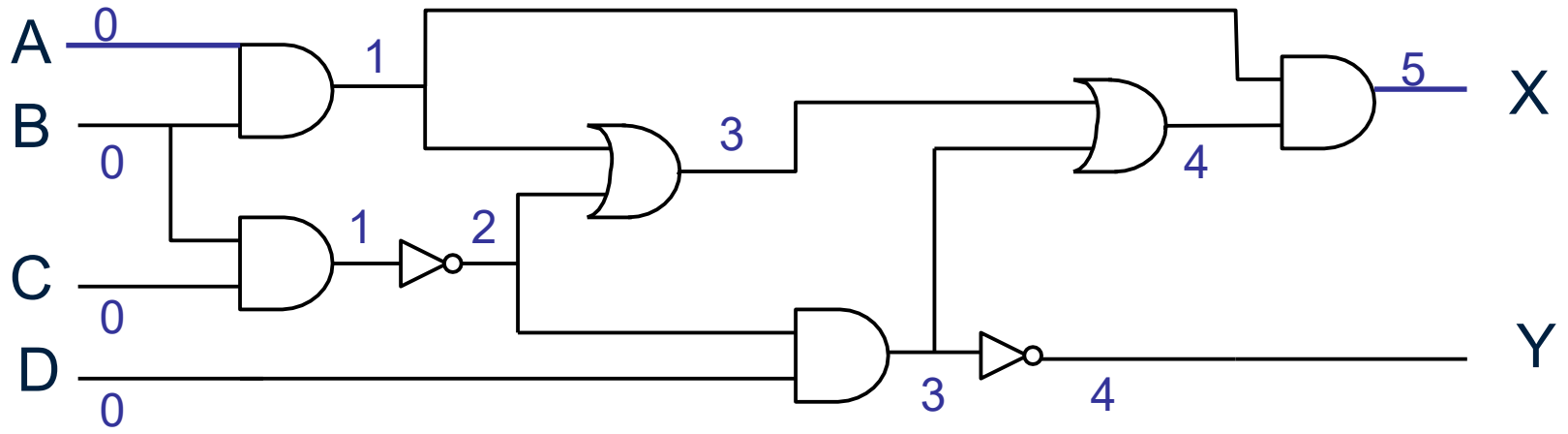
Path Delay

- Delay through a series of combinational gates:
  - Specifically, the time it takes for the output of the series of gates to change after the inputs to the path change.

- Example: What is the path delay in the following cct?
  - Assume propagation delay for each gate is the same (1ns in this example)



Comments on the Cct

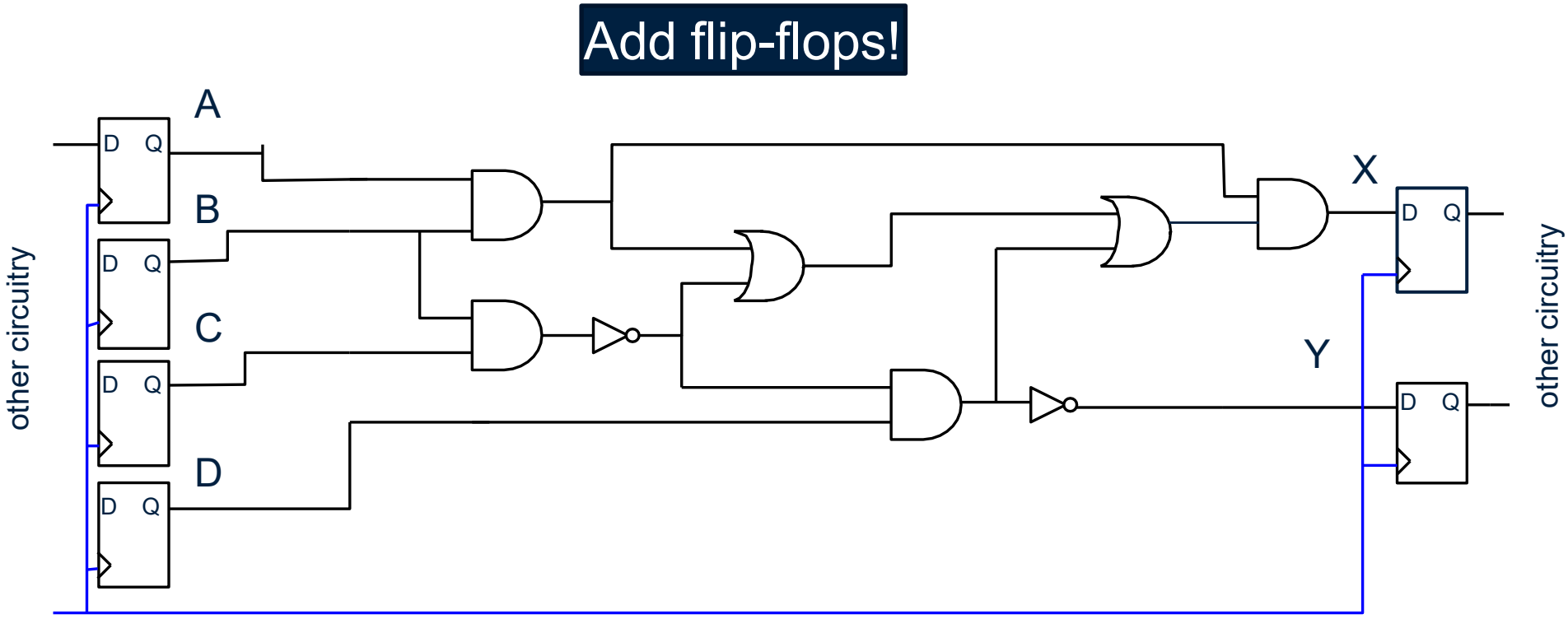


- In the circuit above, if we apply all inputs at time 0, then all the outputs have settled to their final values after 5ns.

- Do you see any problem with this?
  - Some outputs might settle earlier
  - Some outputs may switch back and forth a few times before settling to a final value
  - What about input synchronization for the downstream logic?

How to Synchronize Inputs?

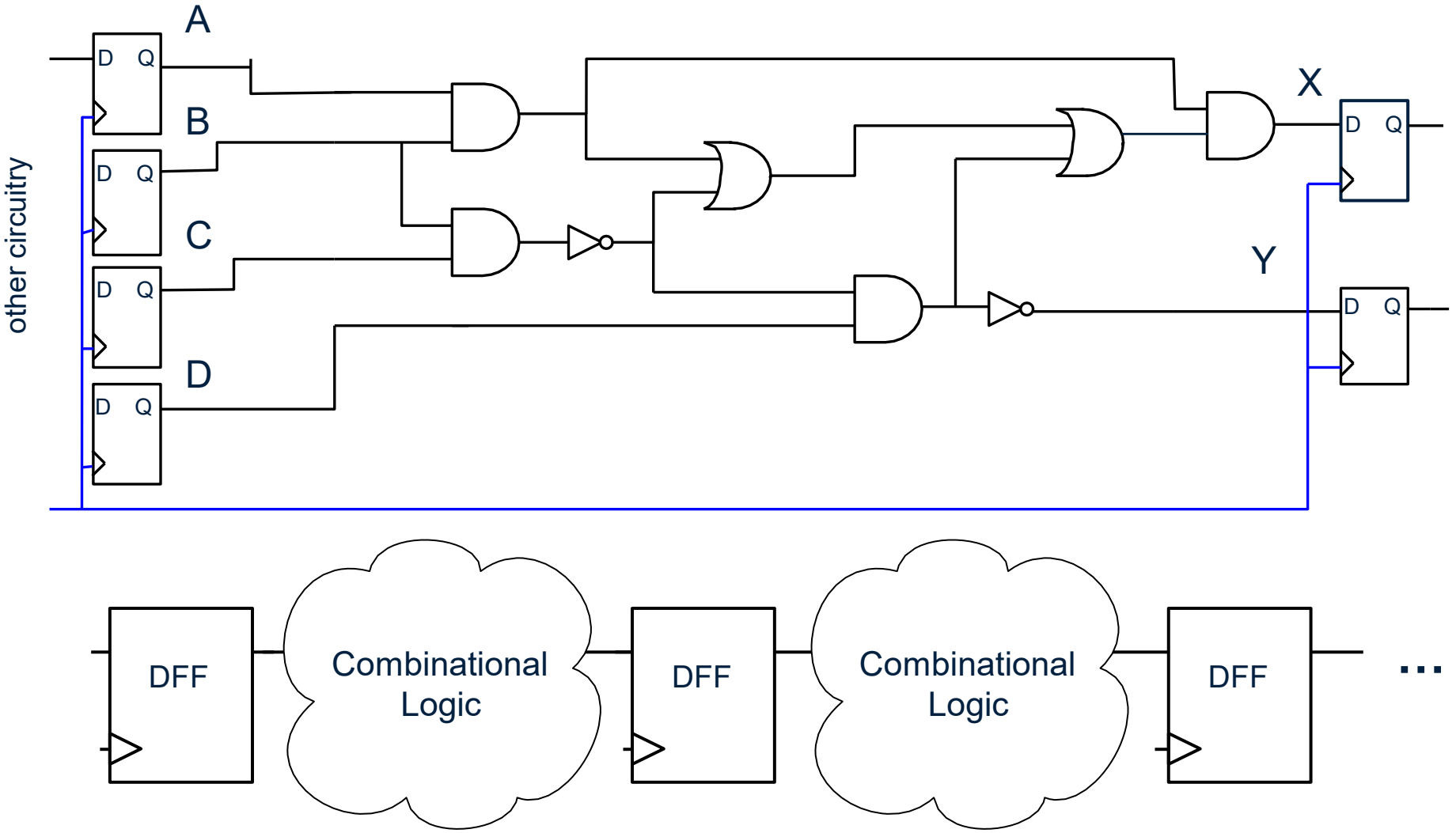
How to Synchronize Inputs?



- Rising edge on clock at time 0:
  - Assuming no delay in the flip-flops, the outputs of the source (left four) flip-flops change at time 0.
- Some time later (one clock cycle), the clock goes high again, and the destination flip-flops read in X and Y

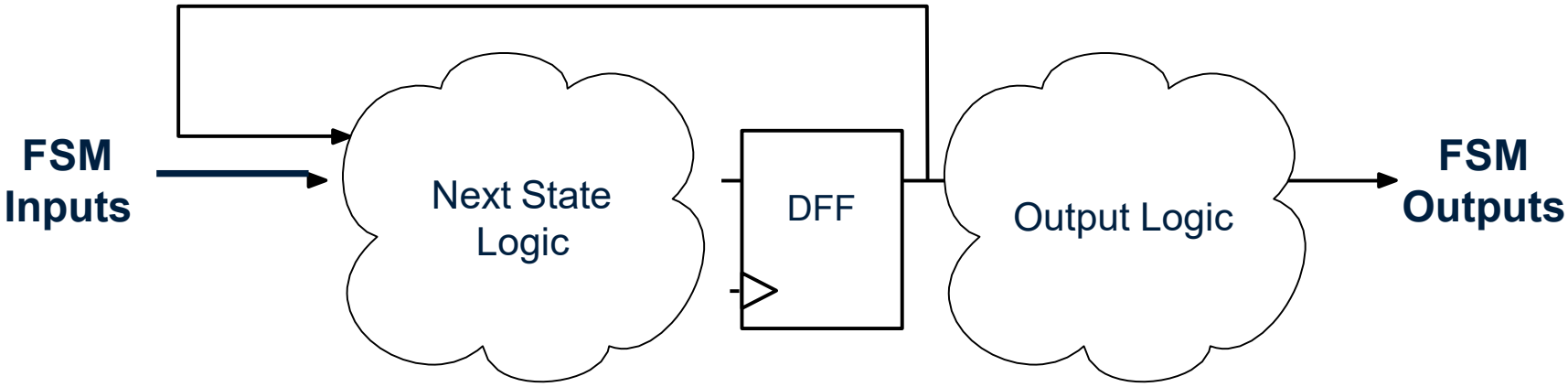
General Structure of a Digital System

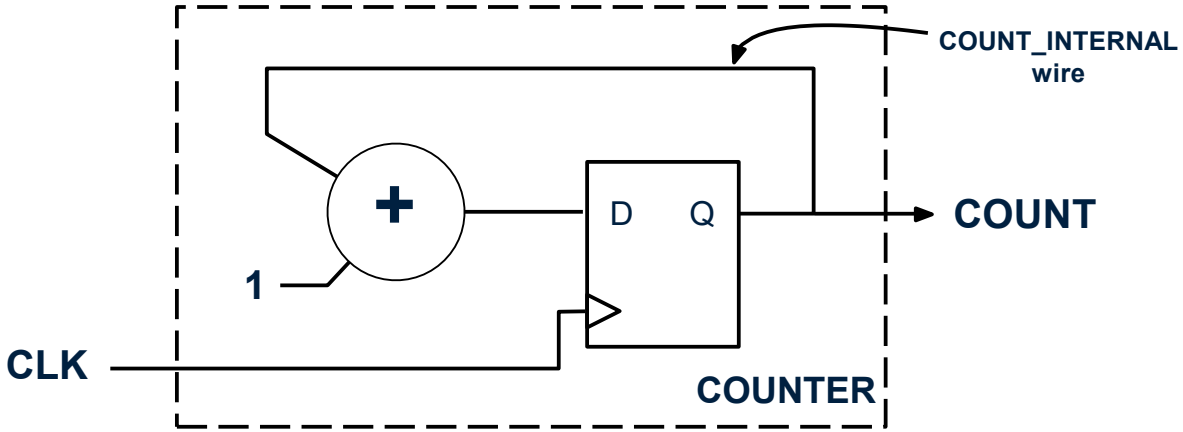
- Digital systems are made up of many stages of flip-flops and combinational logic.



You've Seen this Before

- Finite State Machines, shift registers, counters, etc…

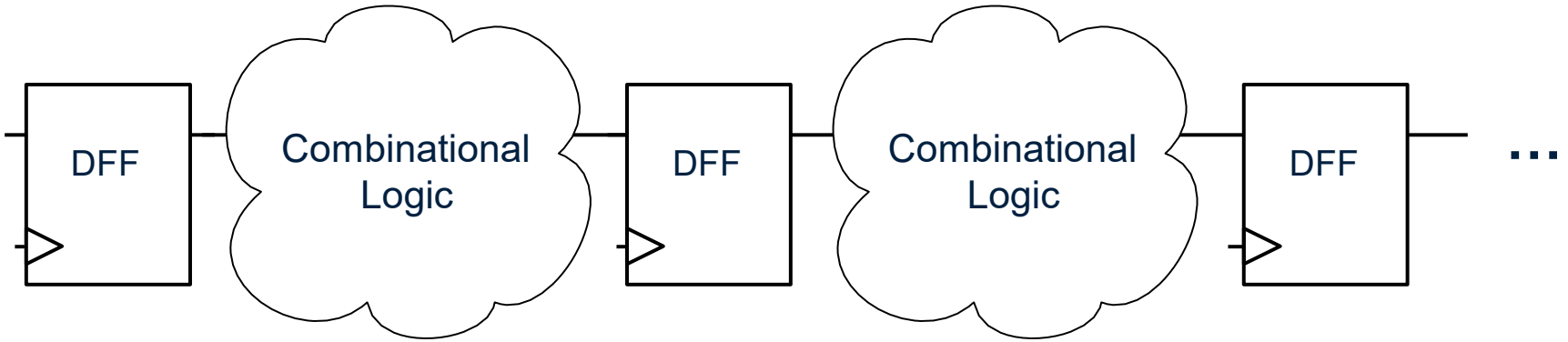




How fast can we run the clock?

How fast can we run the clock?

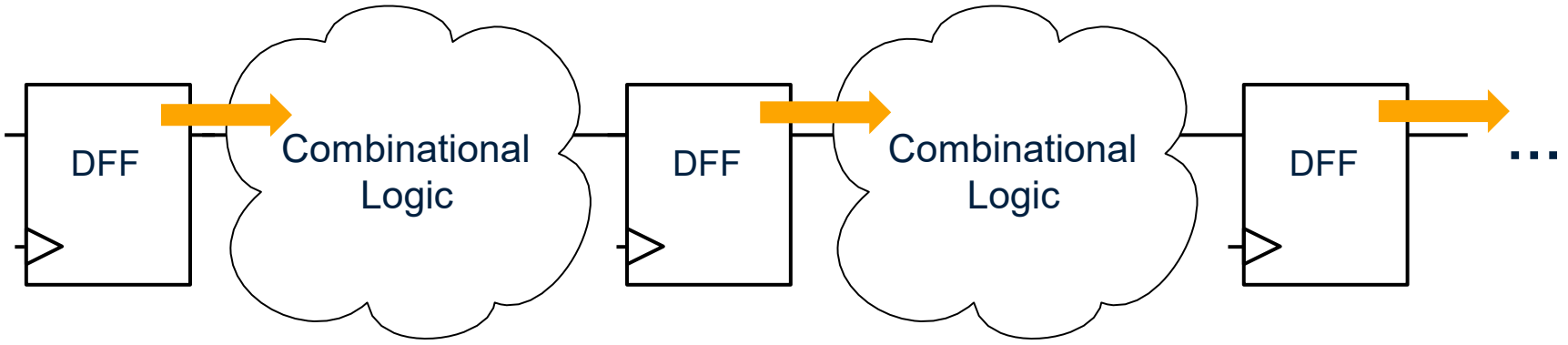
Clock's purpose is to tell flip-flops when to read in data



How fast can we run the clock?

Key Idea

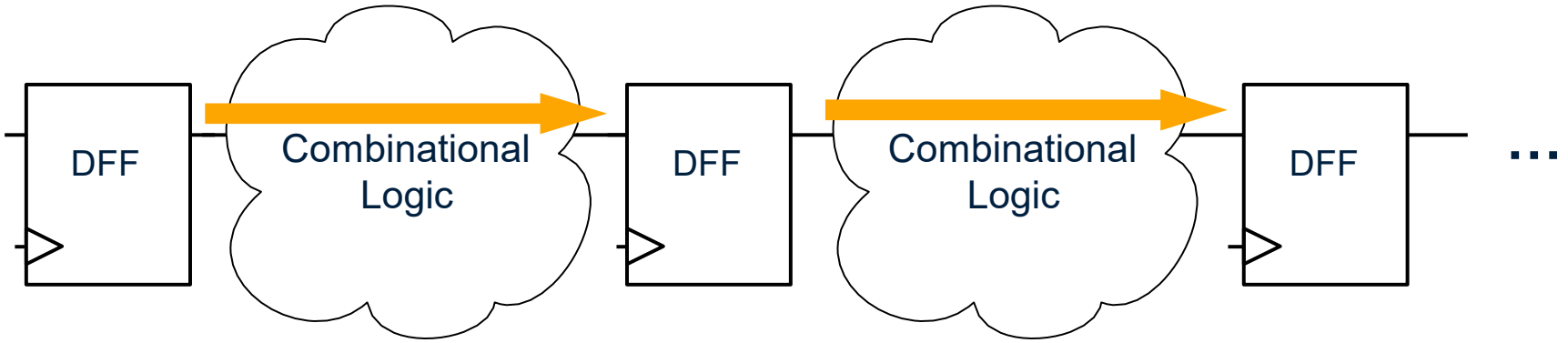
1. On clock edge, new inputs sent into combinational logic



How fast can we run the clock?

Key Idea

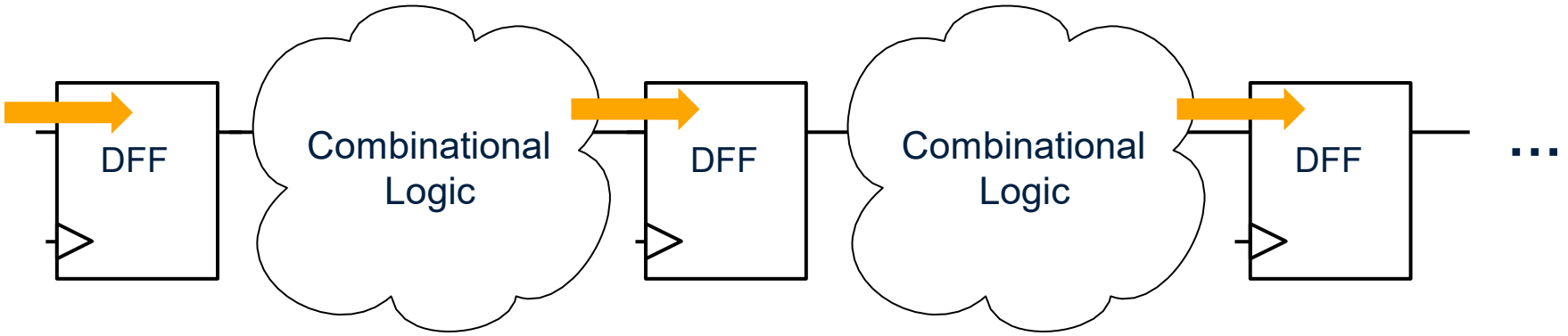
1. On clock edge, new inputs sent into combinational logic
2. Takes some time for outputs to settle



How fast can we run the clock?

Key Idea

1. On clock edge, new inputs sent into combinational logic
2. Takes some time for outputs to settle
3. Don't want to read in new outputs before they are ready

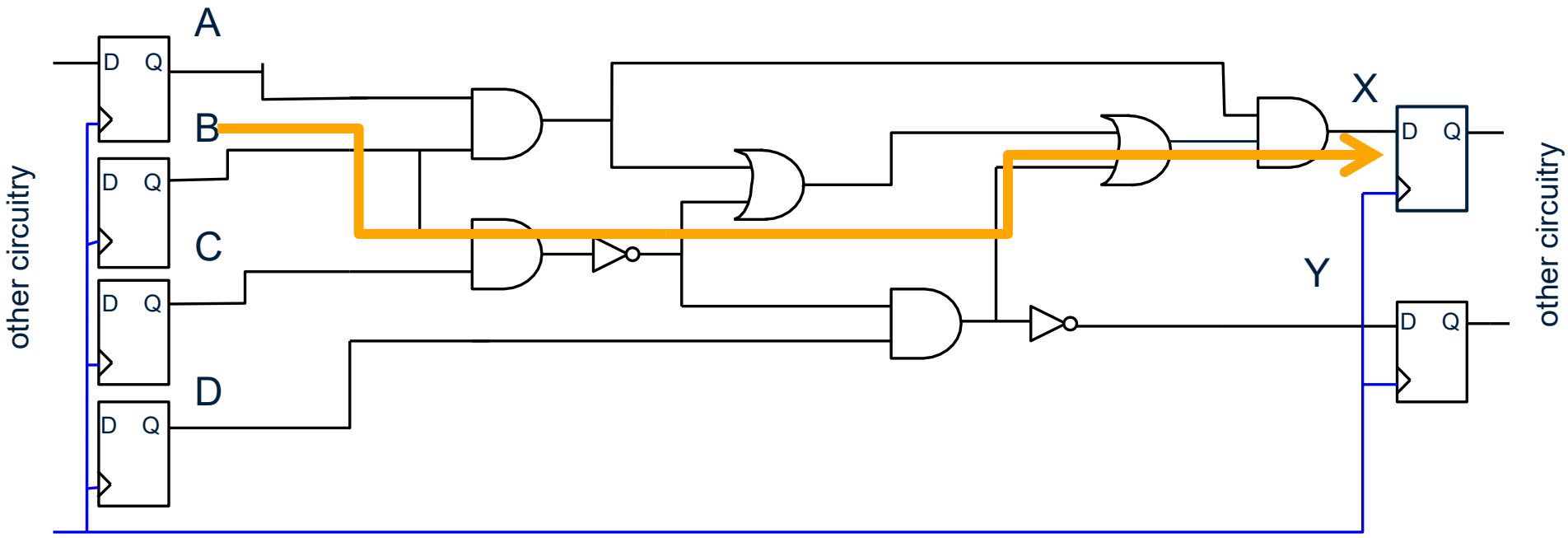


Critical Path and its Delay

- Critical Path:
  - The path between the *source FF output to destination FF input* with the longest delay

- Critical Path Delay:
  - The delay of the critical path

Critical Path Delay: Example



- Critical Path:
  - B ↠ AND ↠ INV ↠ OR ↠ OR ↠ AND ↠ X
- Critical Path Delay:
  - 5ns

Clock period cannot be smaller than 5ns or else X register will read in wrong data

Impact of Critical Path Delay on Operating Frequency

- Critical path delay limits the clock period (and hence the maximum operating clock frequency)

$$t_{ClockMin} \geq t_{CritcalPath}$$

- From the previous example:

$$t_{CritcalPath} = 5ns$$

$$\rightarrow \quad t_{ClockMin} = 5ns$$

$$\rightarrow \quad f_{ClockMax} = \frac{1}{5ns} = 200MHz = f_{Max}$$

What happens if you run the clock SLOWER than f_{Max}?

$$f_{Clock} < f_{Max} \quad \leftrightarrow \quad t_{Clock} > t_{CriticalPath}$$

- No problem!
- Combinational logic has more than enough time to settle

What happens if you run the clock FASTER than f_{Max}?

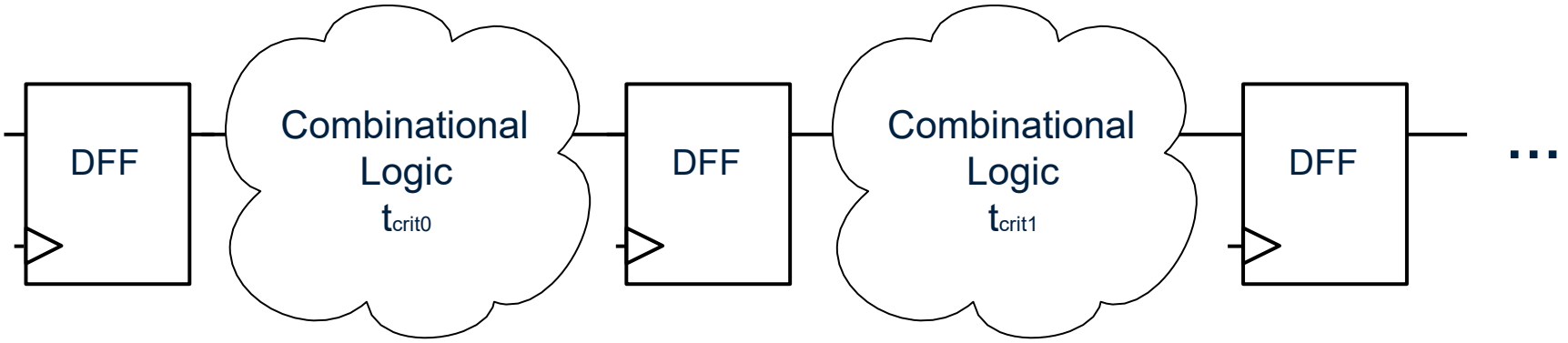
$$f_{Clock} > f_{ClockMax} \leftrightarrow t_{Clock} < t_{CriticalPath}$$

- **WRONG DATA READ INTO OUTPUT FLIP-FLOPS**

- Combinational logic does NOT have enough time to settle to correct value
  - Note: This only applies to your critical path(s).
  - Your non-critical path FFs might still be ok, but you need to design for the worst-case.

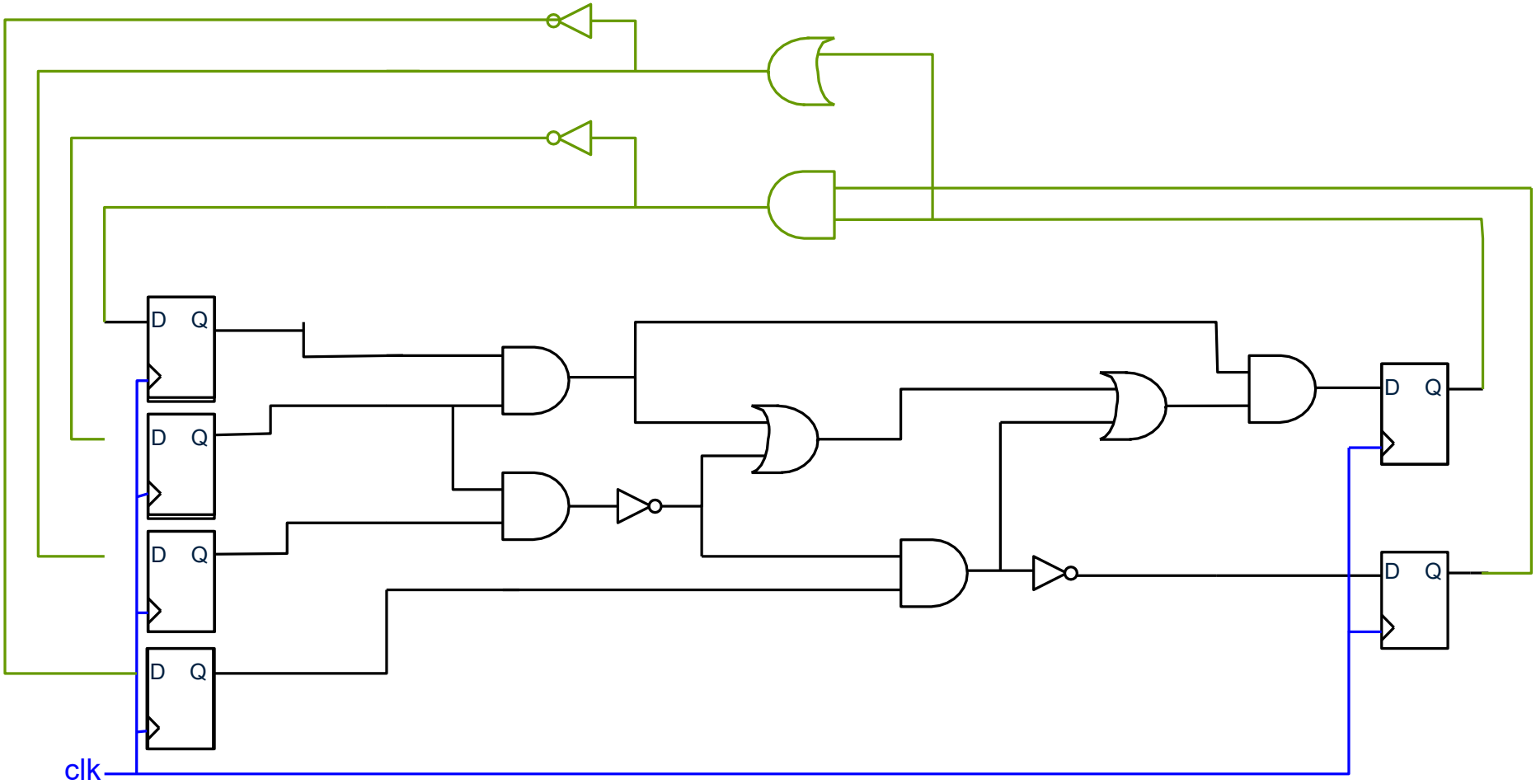
Critical Path in the Entire Design

- Since a single clock controls all flip-flops in all stages, you need to look for the critical path amongst all stages



$$t_{CriticalPath} = \max_{foreach\,path i} \left(t_{crit\_i} \right)$$

Class Activity: Find the Critical Path



THANK YOU