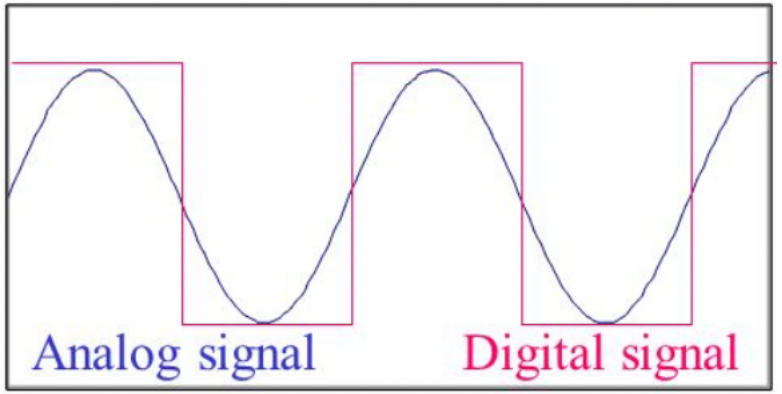
Synchronous Digital Systems

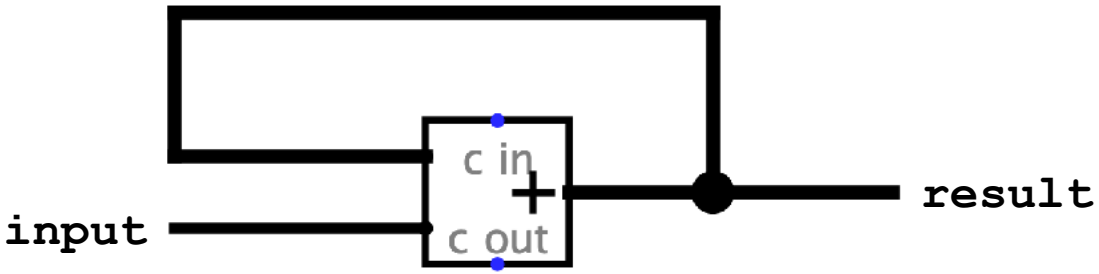
Synchronous Digital Systems

- Synchronous
  - All operations are coordinated by something called a clock
- Digital
  - All values are discrete
  - A value can be on (1) or off (0)



Accumulator

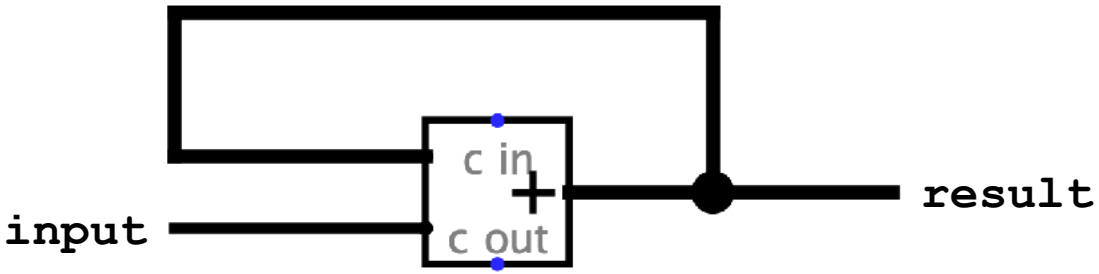
- Assume the result starts at 0
- It takes 5ns for the adder to compute the result
- A new value comes in every 10 ns
  - At t = 0, input = 5
  - At t = 10, input = 7



time	input	result
0	5	
5	5	
10	7	
15	7	

Accumulator

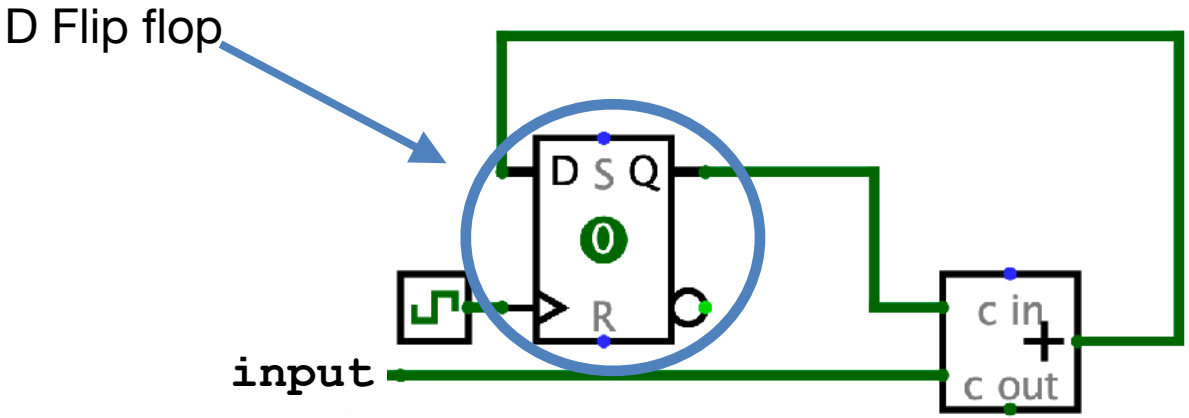
- Assume the result starts at 0
- It takes 5ns for the adder to compute the result
- A new value comes in every 10 ns
  - At t = 0, input = 5
  - At t = 10, input = 7



time	input	result
0	5	0
5	5	5
10	7	10
15	7	17

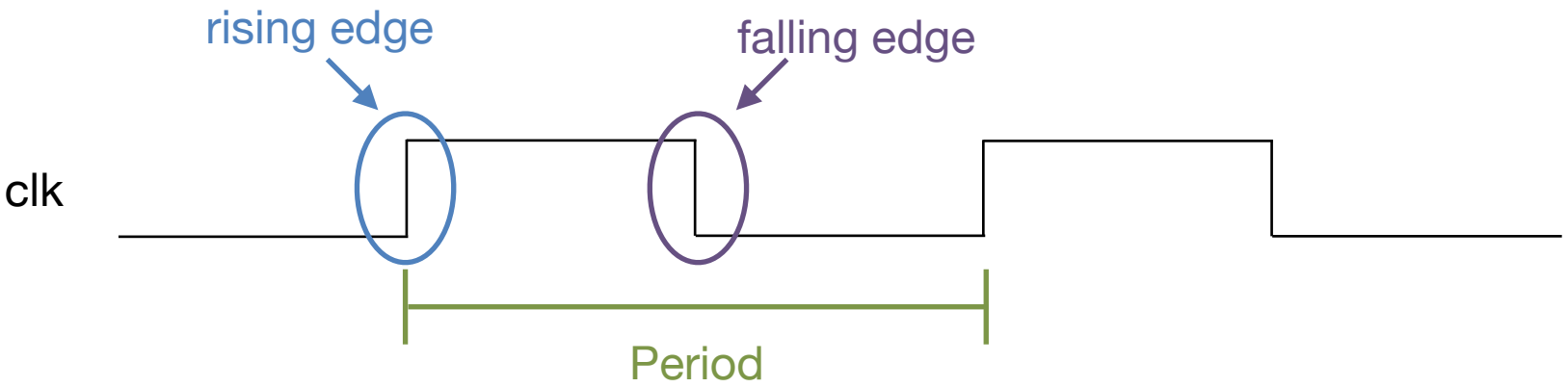
Accumulator

- We need to add a barrier so that the output of the adder does not go to the input of the adder until the next input comes



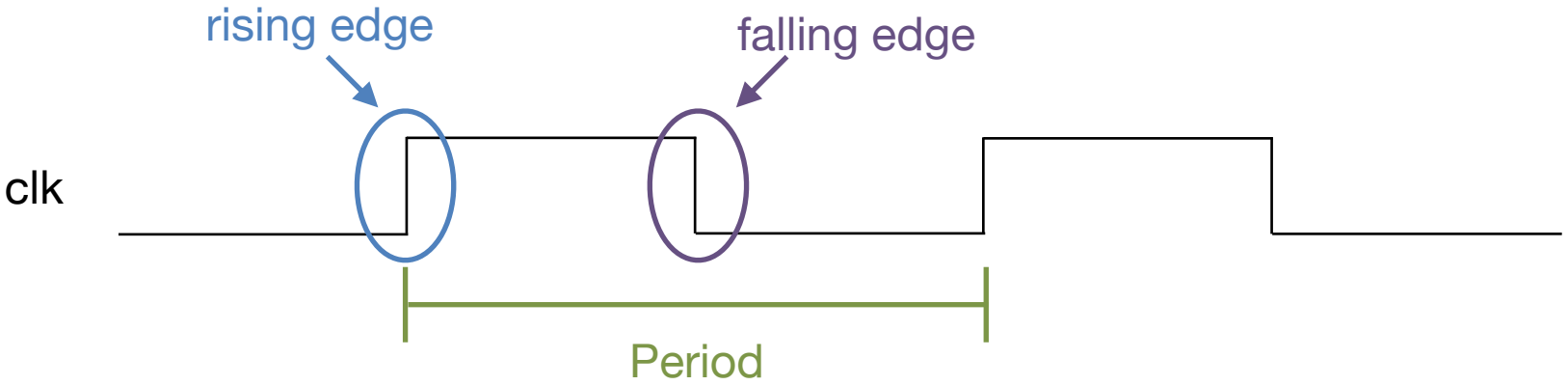
Clock Signal

- Oscillates between a high and low state
- Period = time between one rising edge to the next rising edge
- Frequency = 1/Period



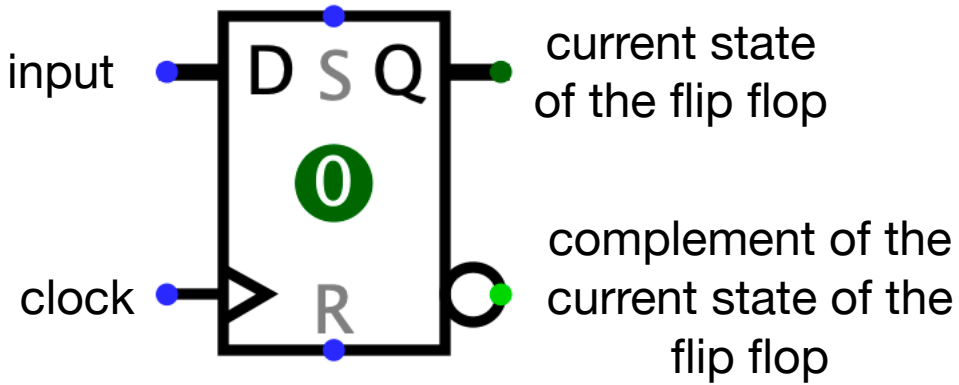
Clock

- Unit for frequency is Hertz (Hz)
- Common clock frequency is 4 GHz
  - the clock goes through 4 billion cycles every second
  - Period = 1/frequency
  - Period = 1/4 GHz = 0.25 ns



D Flip-Flops

When S = 1, set the state of the flip flop to 1
Does not depend on the clock



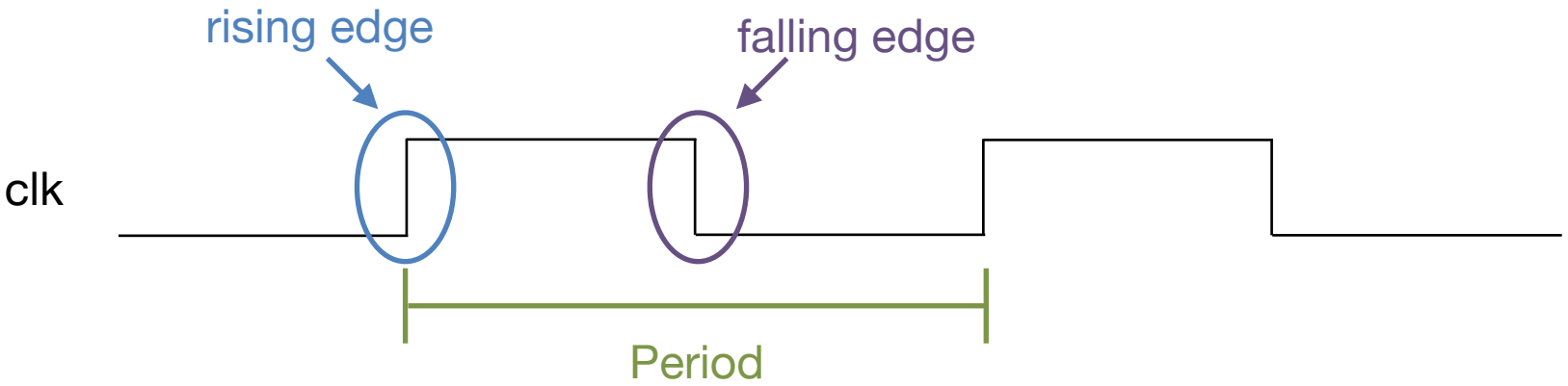
When R = 1, set the state of the flip flop to 0
Does not depend on the clock

Flip Flops

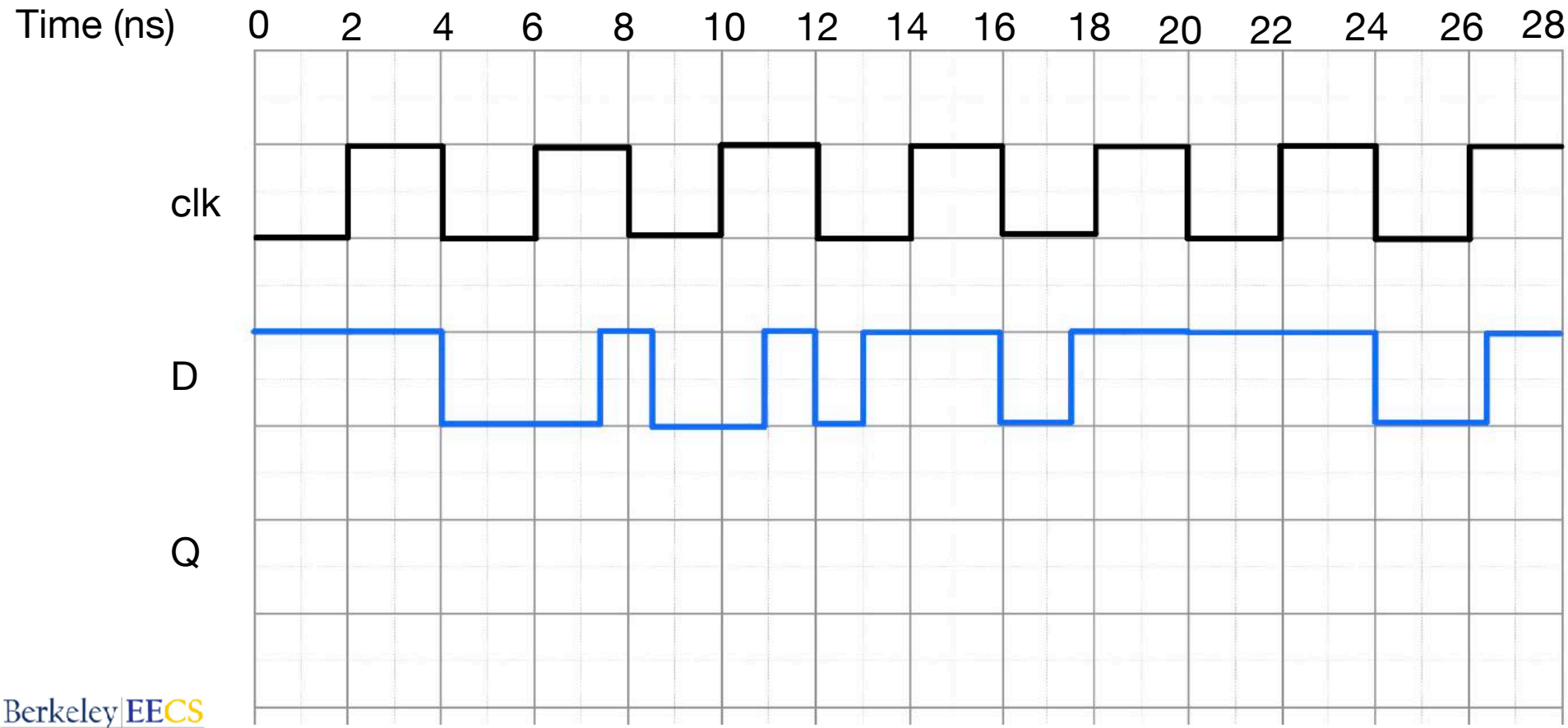
- Synchronous: dependent on the clock
- Asynchronous: independent of the clock
- A Flip-flop is a state element
- State Element: A circuit component that can hold a value

D Flip-Flop

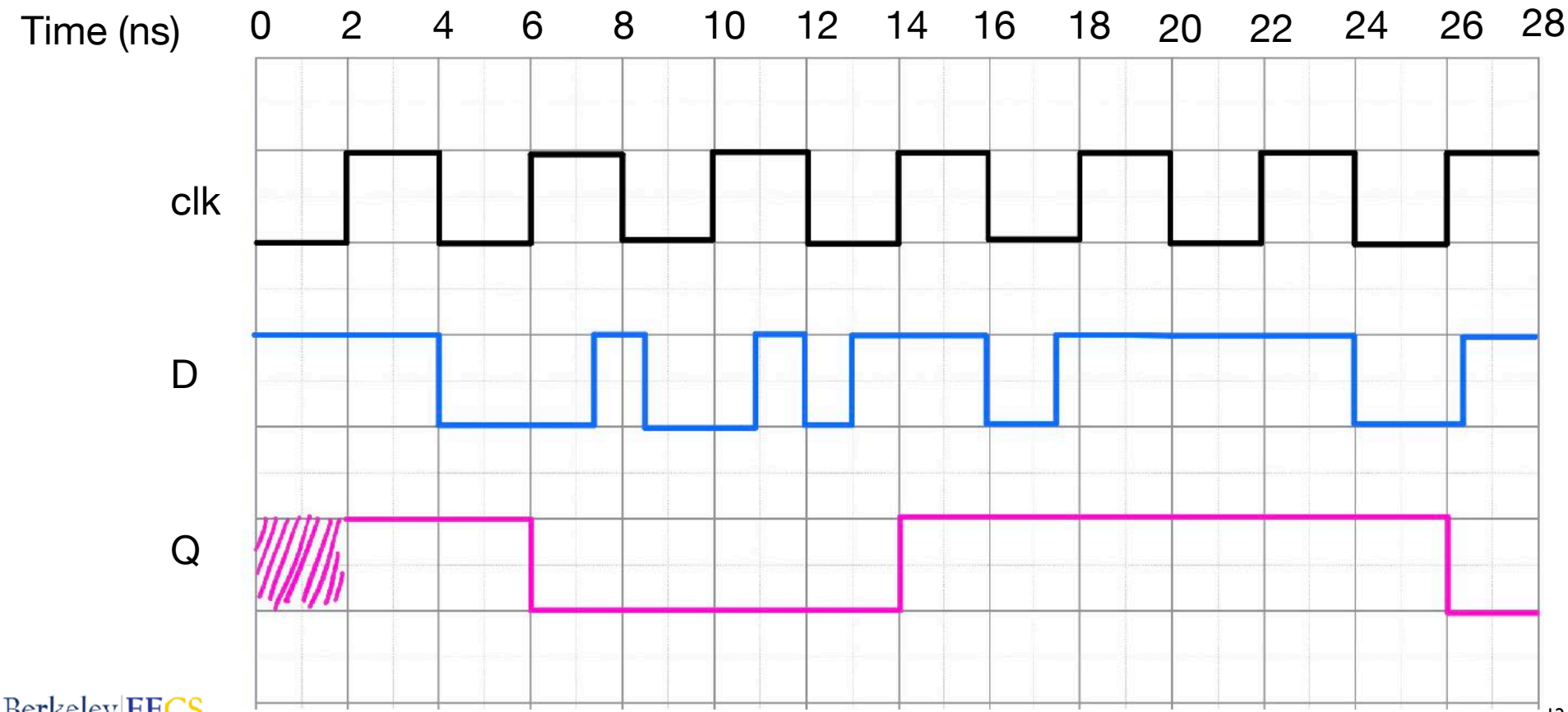
- Rising edge triggered
  - Stores D to Q the instant the clock goes from 0 to 1
- Falling edge triggered
  - Stores D to Q the instant the clock goes from 1 to 0



Positive Edge Triggered D Flip-flop Timing Diagram Example

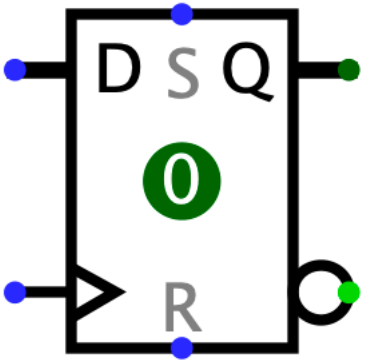


D Flip-flop Timing Diagram Example



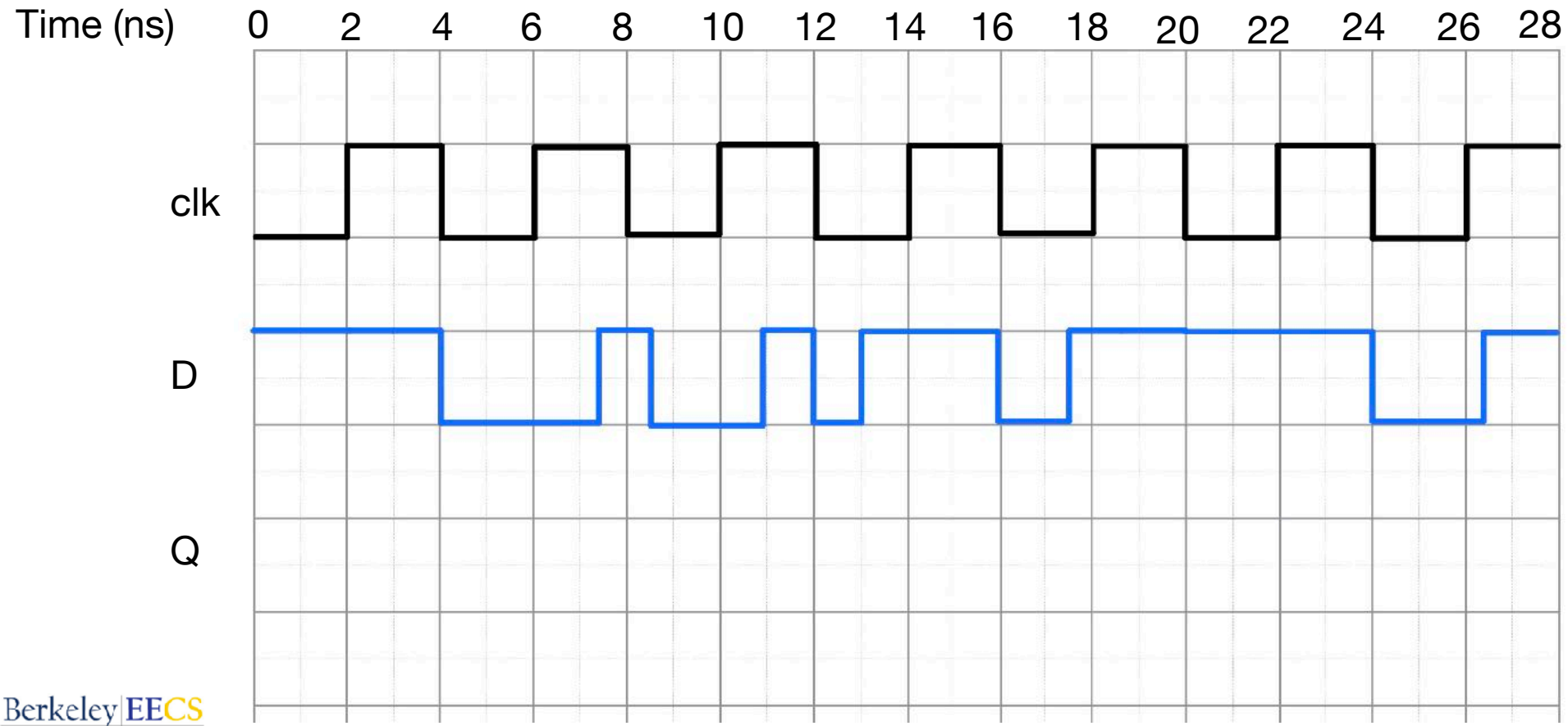
Clock-to-Q delay

- The amount of time that it take for the input to propagate to the output after the clock trigger



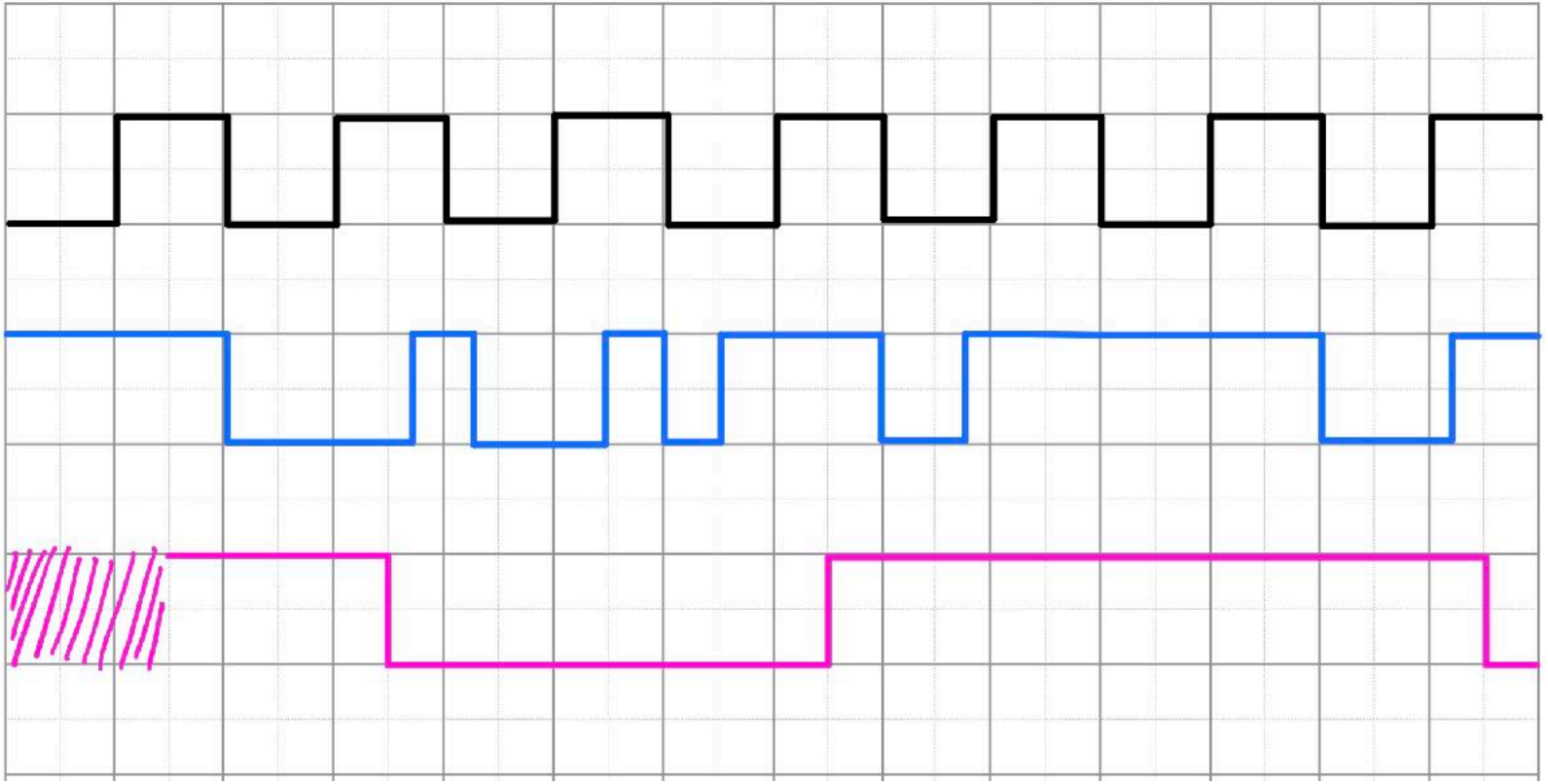
New Timing Diagram with clk-to-q Delay

clk-to-q = 1ns



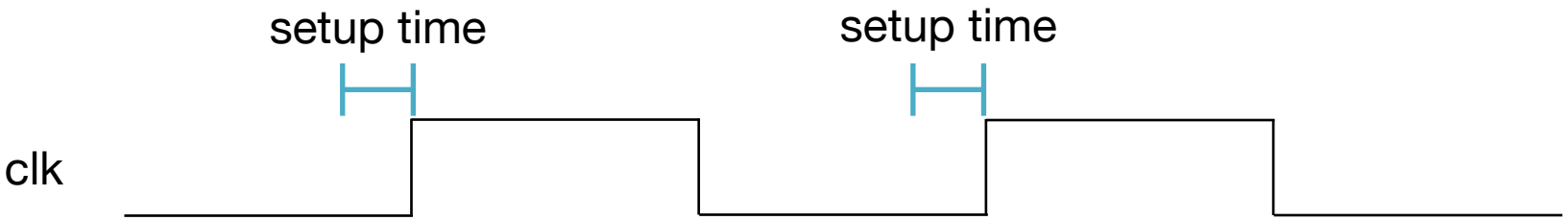
New Timing Diagram with clk-to-q Delay

clk-to-q = 1ns



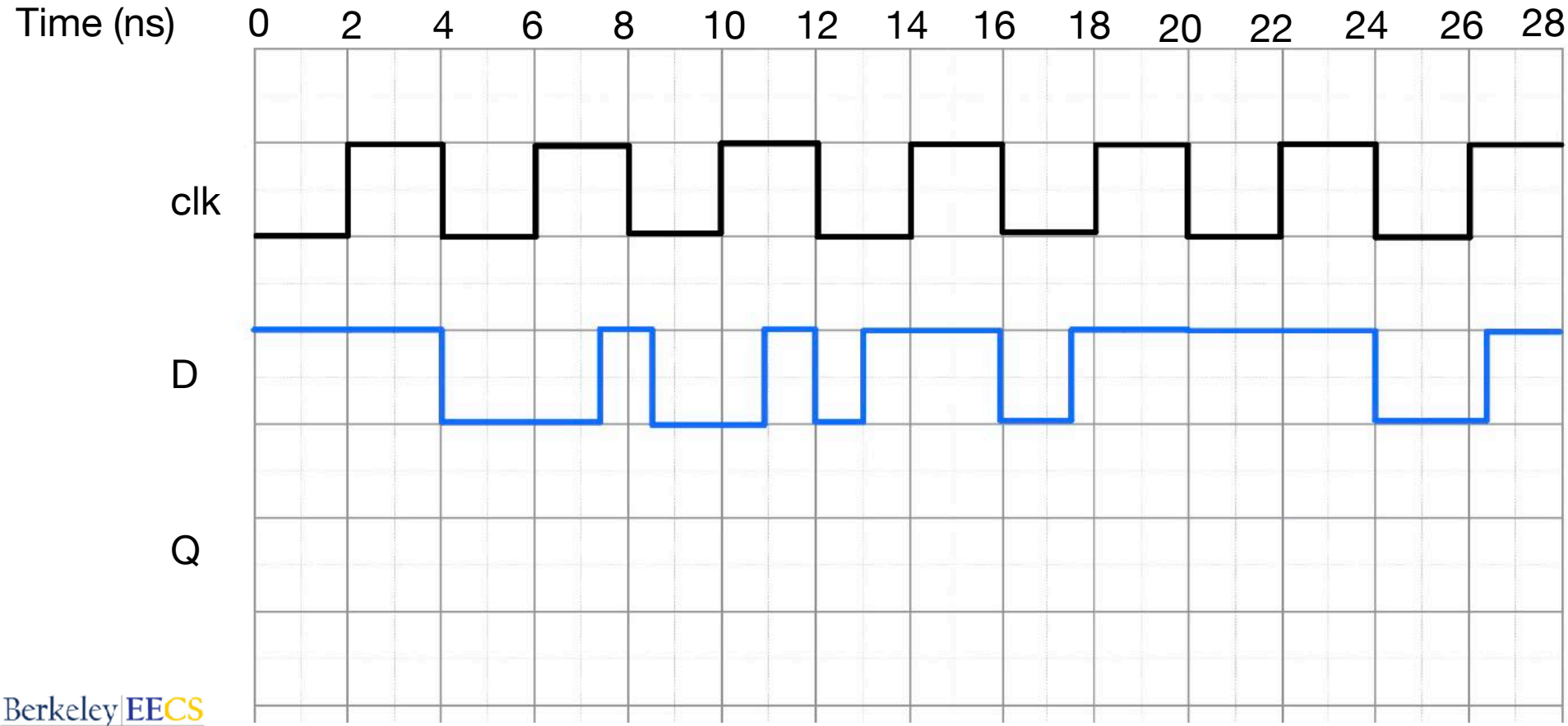
Set-up Time

- The amount of time that the input needs to be stable BEFORE the clock trigger
- This example is a positive edge triggered flip-flop



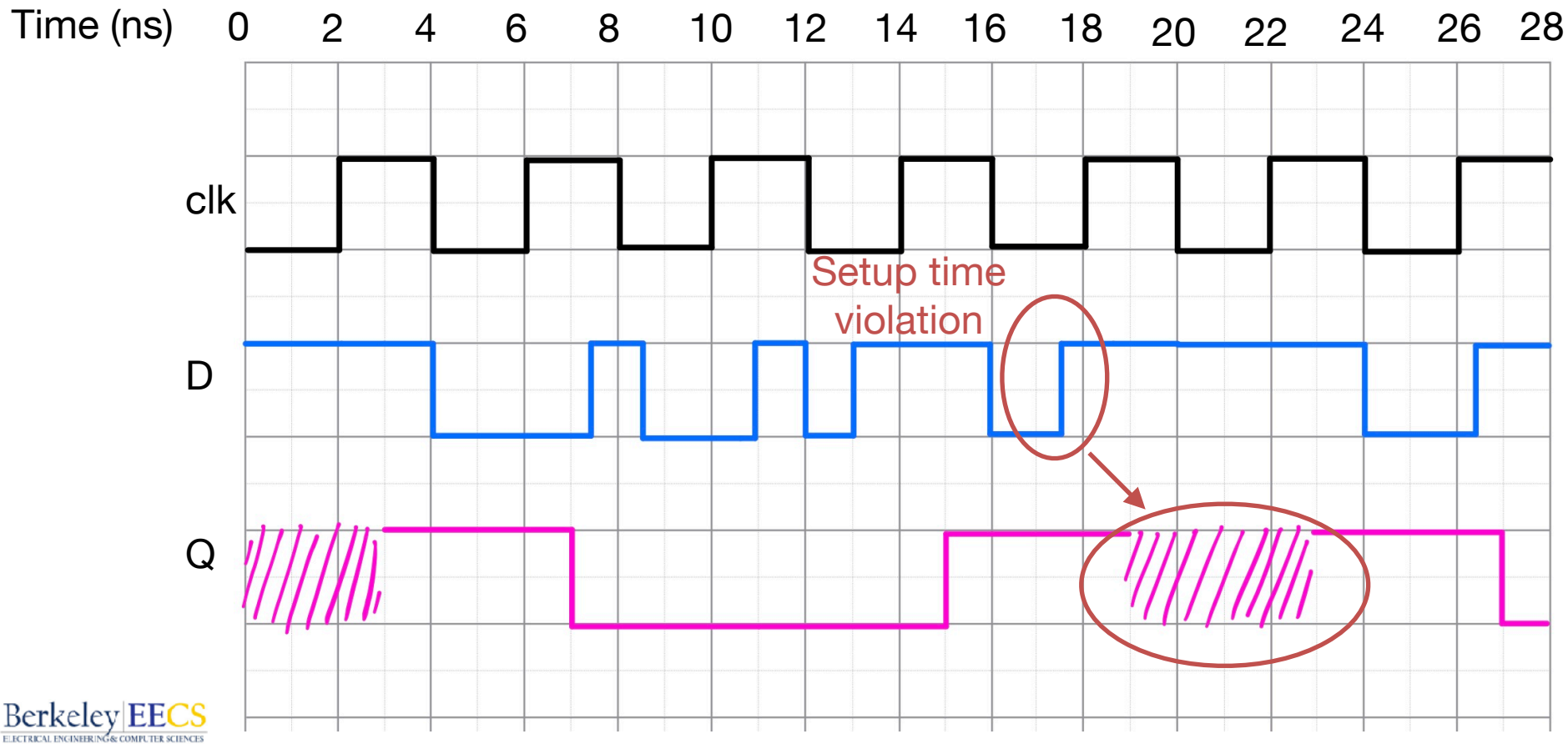
Timing Diagram with clk-to-q and Setup Time

clk-to-q = 1ns
setup time = 1ns



Timing Diagram with clk-to-q and Setup Time

clk-to-q = 1ns
setup time = 1ns

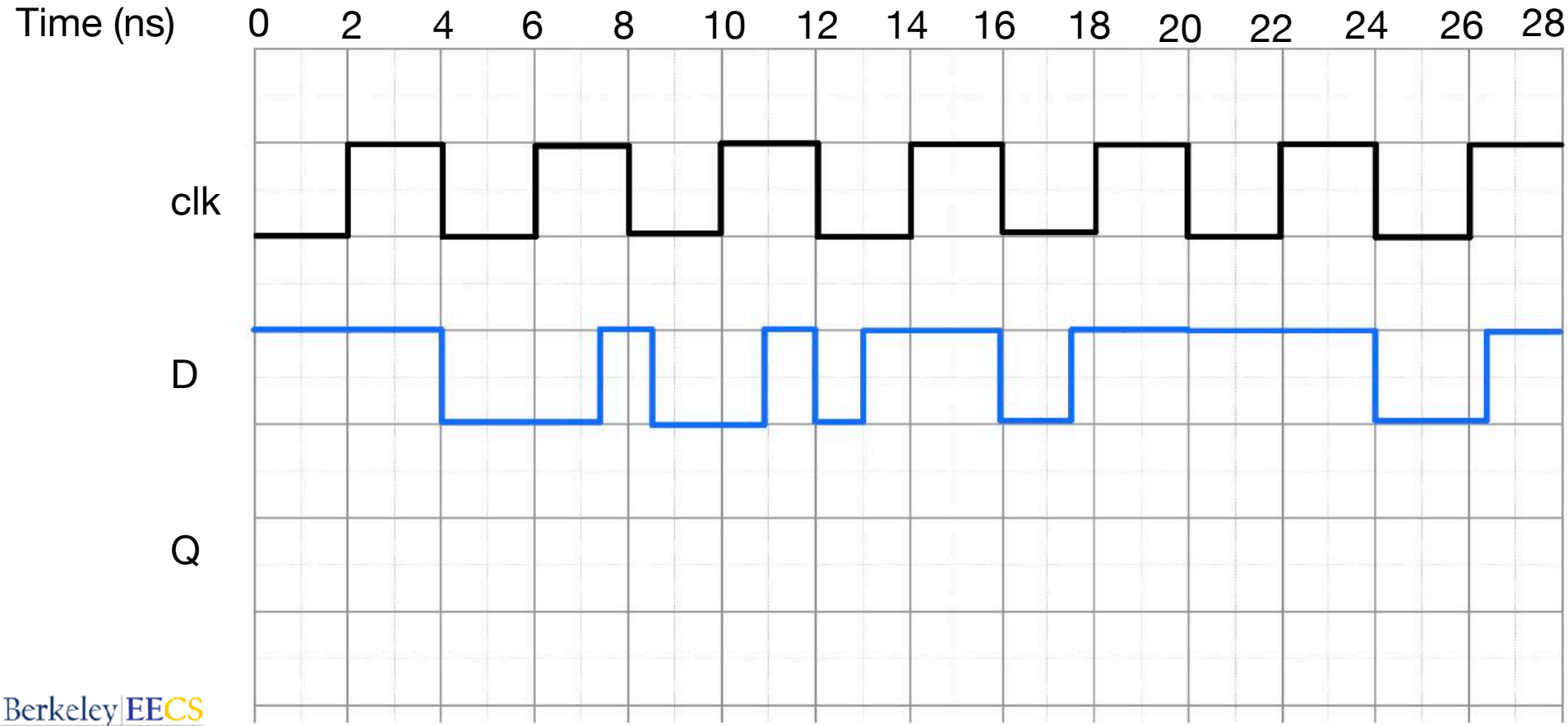


Hold Time

- The amount of time that the input needs to be stable AFTER the clock trigger
- This example is a positive edge triggered flip-flop

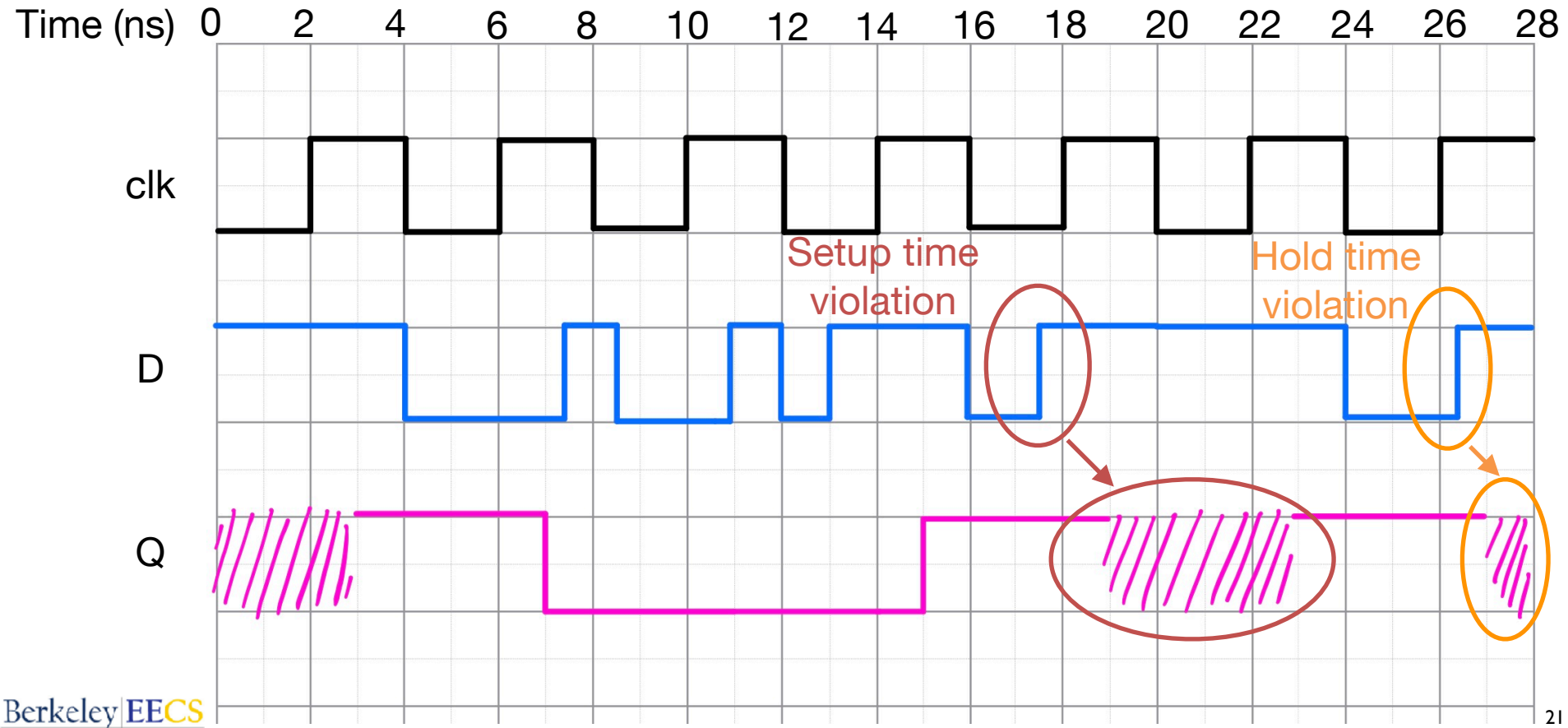
Timing Diagram with clk-to-q, Setup, and Hold Time

clk-to-q = 1ns
setup time = 1ns
hold time = 1ns



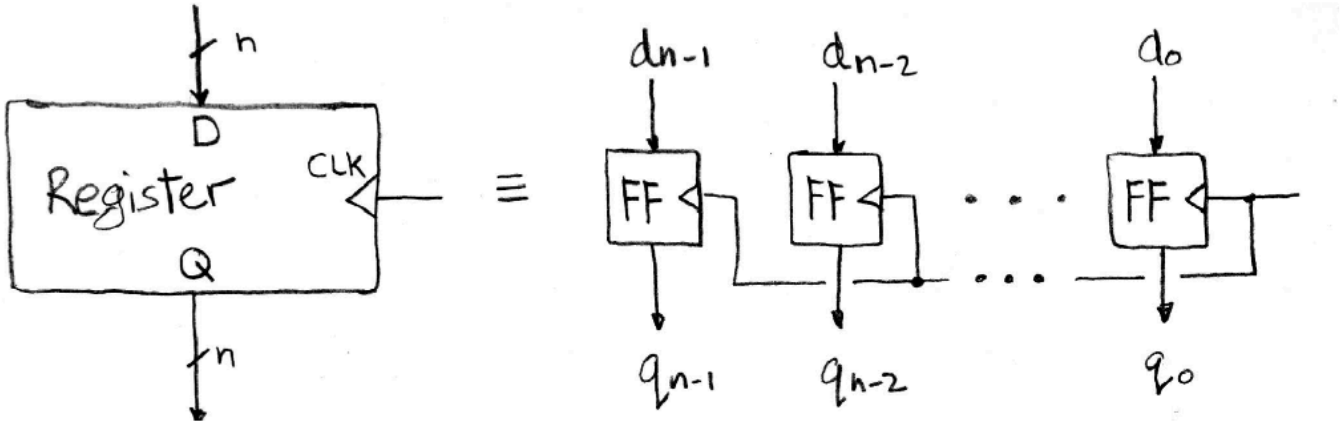
Timing Diagram with clk-to-q, Setup, and Hold Time

clk-to-q = 1ns
setup time = 1ns
hold time = 1ns

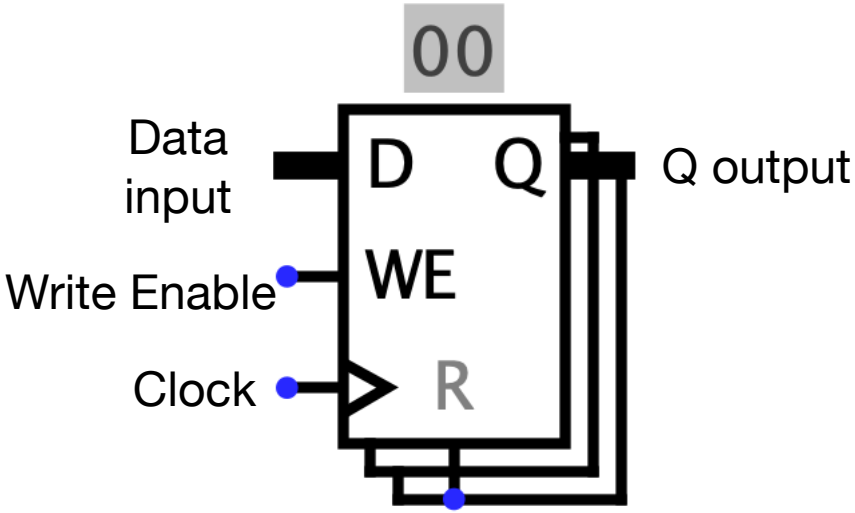


How to store a 32 bit number?

- Put 32 flip flops together
- This is called a register
- A Register is a state element



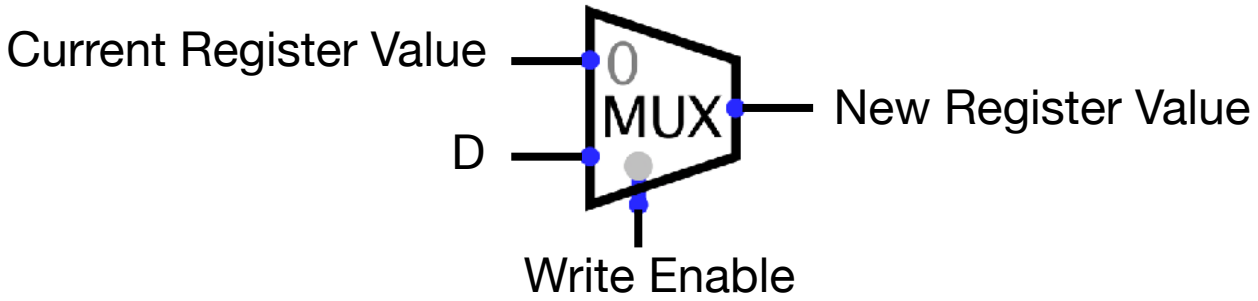
Registers



When 1, set register contents to 0

Write Enable

- When 0, the contents of the register stay the same on the clock trigger
- When 1, the contents of the register are updated on the clock trigger



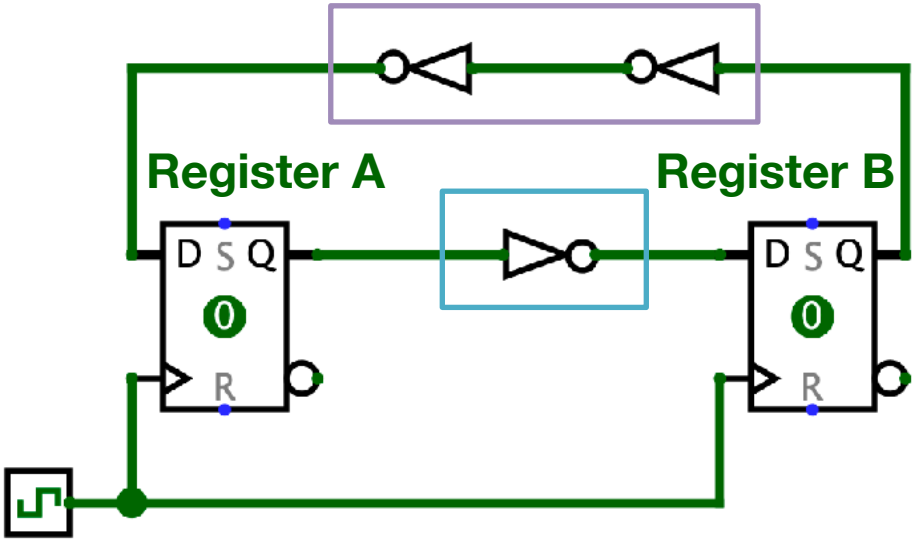
Combinational logic delay

- The amount of time that it takes for a value to propagate through the combinational logic

Inverter Delay = 2ns

Combinational logic delay from register A to register B = 2ns

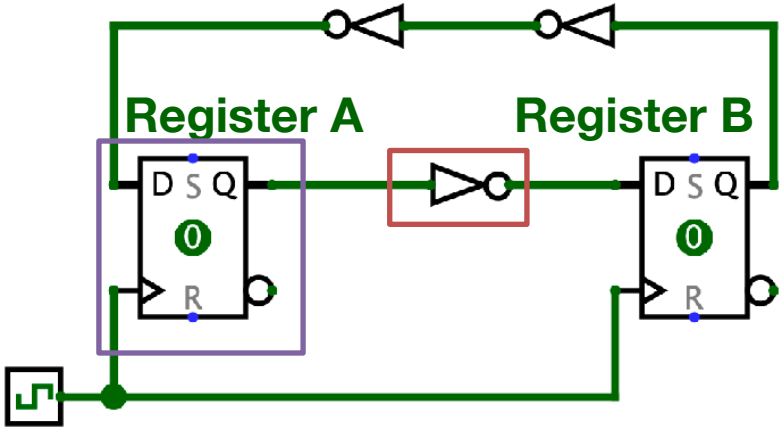
Combinational logic delay from register B to register A = 4ns



What is the maximum allowable hold time for register B?

- Hold Time
  - The amount of time that the input to the register must remain stable after the rising edge of the clock
- Max hold time
  - The amount of time that it takes for the input to B to change after the trigger
  - clk-to-q delay of register A + Combinational logic delay

max hold time = 3ns + 2ns
= 5ns

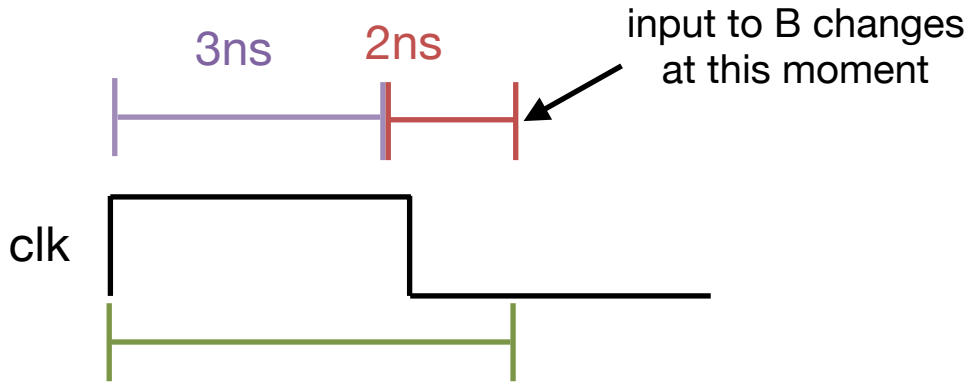


Inverter Delay = 2ns

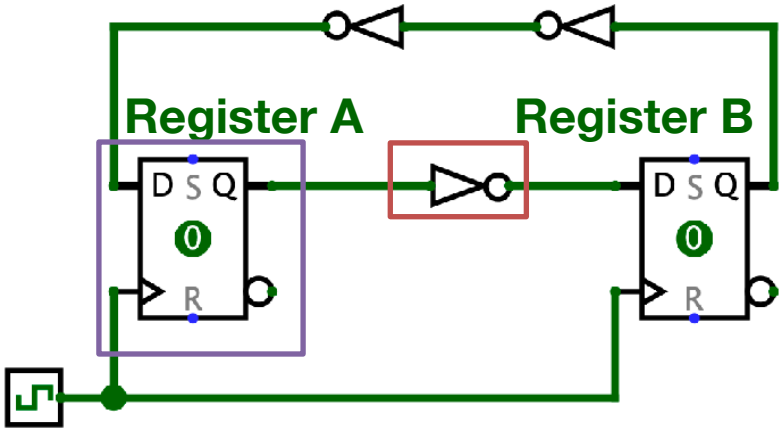
Clk-to-q delay of registers = 3 ns

Set up time of registers = 2ns

What is the maximum allowable hold time for register B?



max hold time = 5ns



Clk-to-q delay of register A = 3ns

Combinational Delay = 2ns

max hold time = 3ns + 2ns

= 5ns

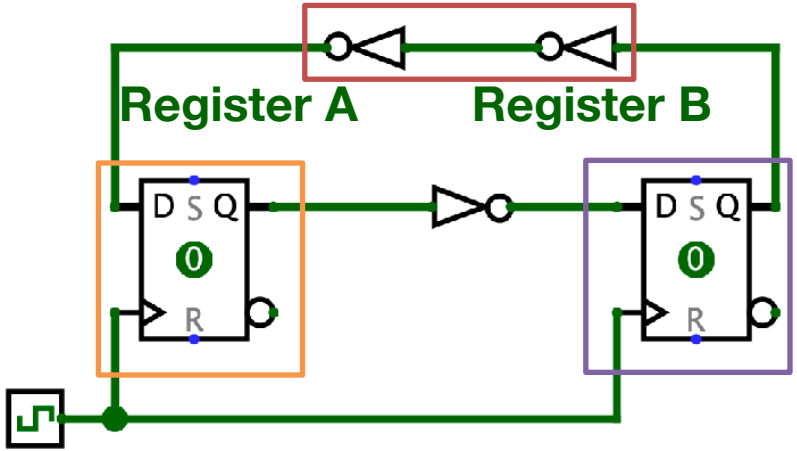
Inverter Delay = 2ns

Clk-to-q delay of registers = 3 ns

Set up time of registers = 2ns

What is the minimum clock cycle time (aka critical path)?

- Clock Cycle
  - Min clock cycle = The time it takes for the input of one state element to reach the input of the next state element
  - clk-to-q delay + longest combinational delay + setup time
  - In this circuit, the critical path occurs between Register B and Register A



Clk-to-q delay of register B = 3ns

Longest Combinational Delay = 4ns

Setup time of register A = 2ns

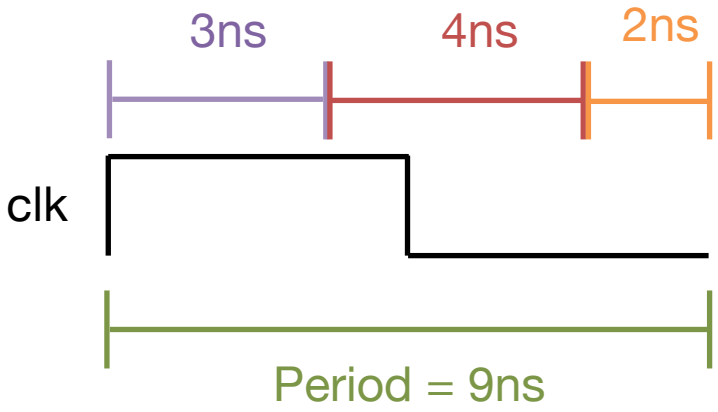
Min clock cycle time = 9ns

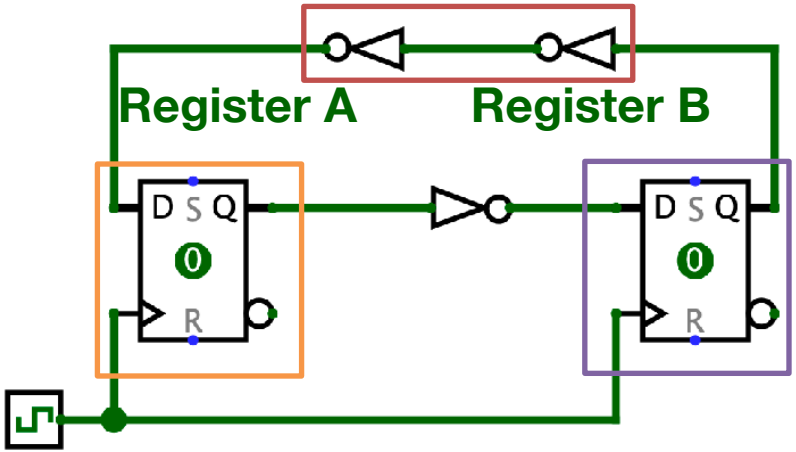
Inverter Delay = 2ns

Clk-to-q delay of registers = 3 ns

Set up time of registers = 2ns

Clock Cycle Time





Clk-to-q delay of register B = 3ns

Longest Combinational Delay = 4ns

Setup time of register A = 2ns

Min clock cycle time = 9ns

Inverter Delay = 2ns

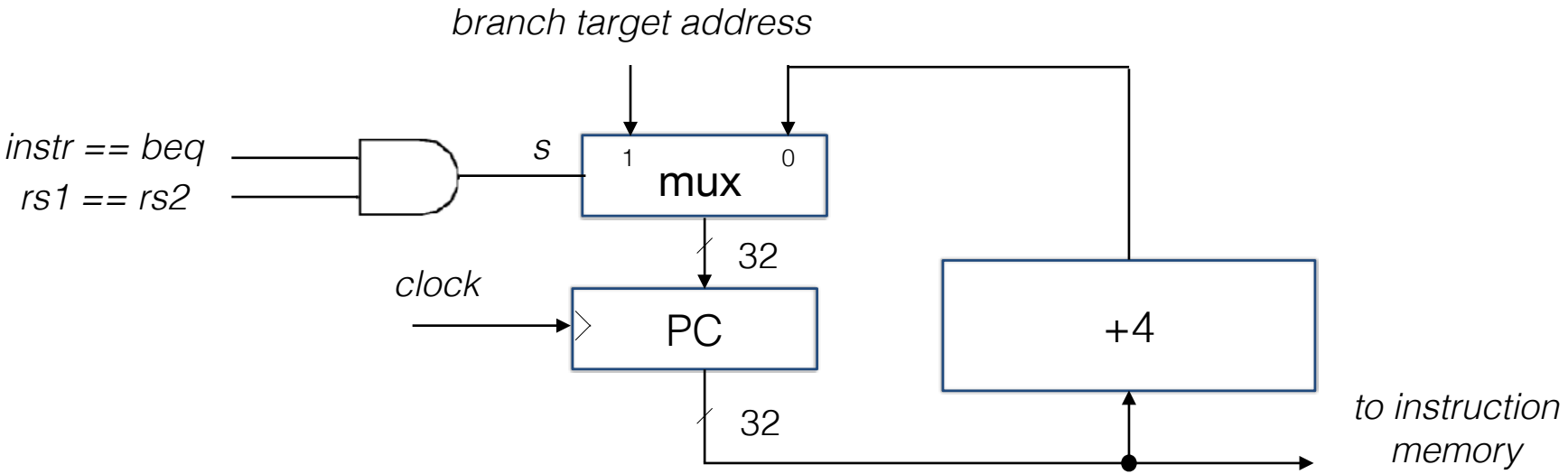
Clk-to-q delay of registers = 3 ns

Set up time of registers = 2ns

Circuit Timing Analysis

- Due to limited time in lecture, we don’t have time to go over more complex examples
- Go to discussion or review the discussion worksheets for more complex examples

PC register

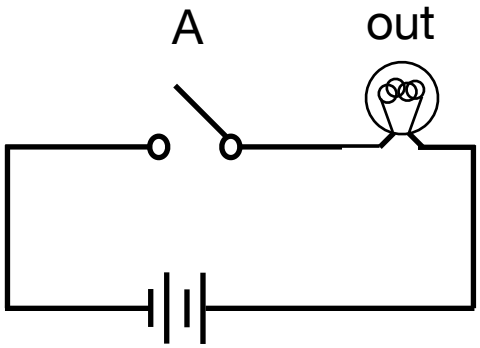

branch target address
instr == beq
rs1 == rs2
s
1
0
mux
32
clock
PC
+4
32
to instruction memory

Combinational vs Sequential Logic

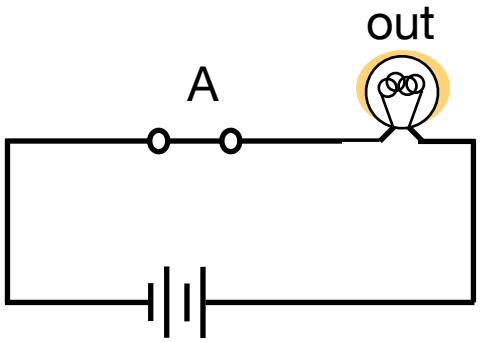
- Combinational
  - As soon as the inputs are available, the output starts being computed
  - Output depends only on the current input
- Sequential
  - Synchronized with a clock signal
  - Output depends on a combination of inputs and previous state

Transistors

Switches



When switch is open, light bulb is off



When switch is closed, light bulb is on

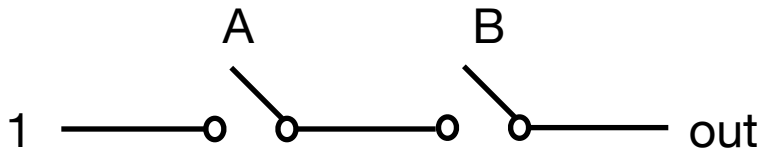
open switch = 0
closed switch = 1

light off = 0
light on = 1

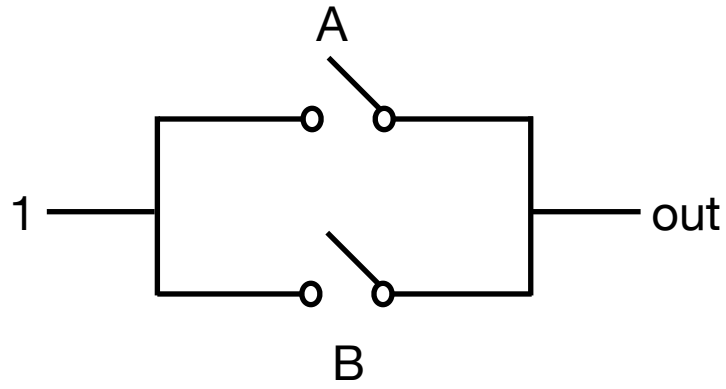
out = A

Boolean Expressions from Switches

open switch = 0
closed switch = 1



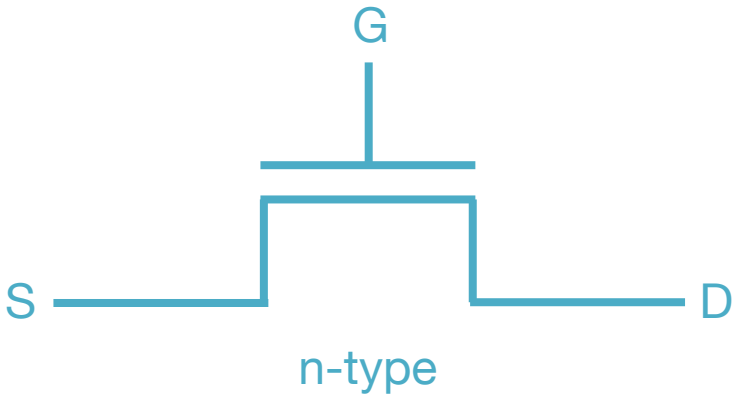
out = AB



out = A + B

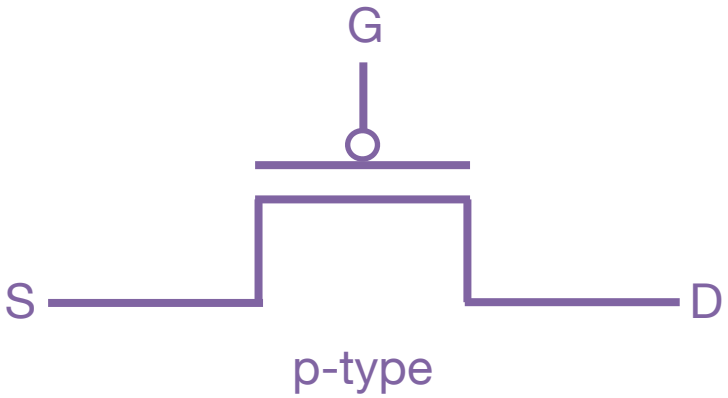
Metal-Oxide Semiconductor Field Effect Transistor

- Three terminals
  - Source = input
  - Gate = controls whether the switch is open or closed
  - Drain = output



n-type

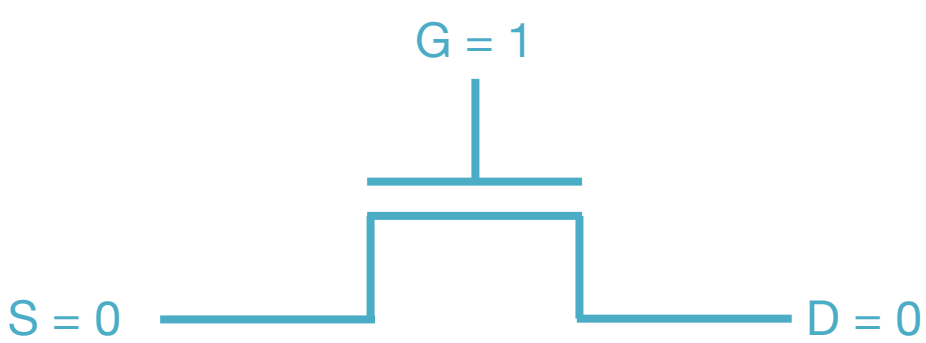
Gate = 1, switch is closed
Gate = 0, switch is open

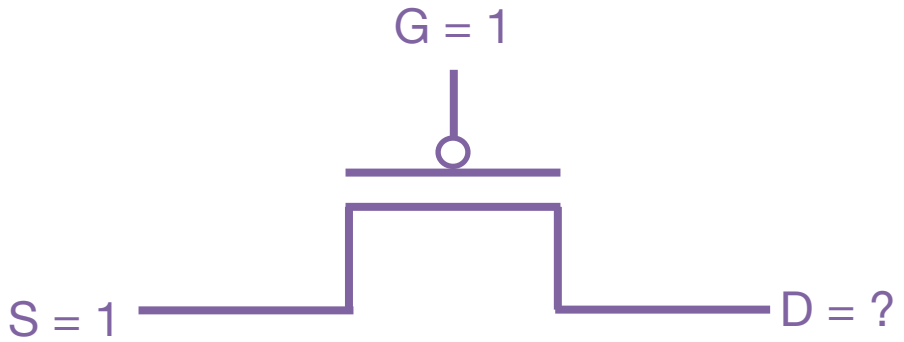


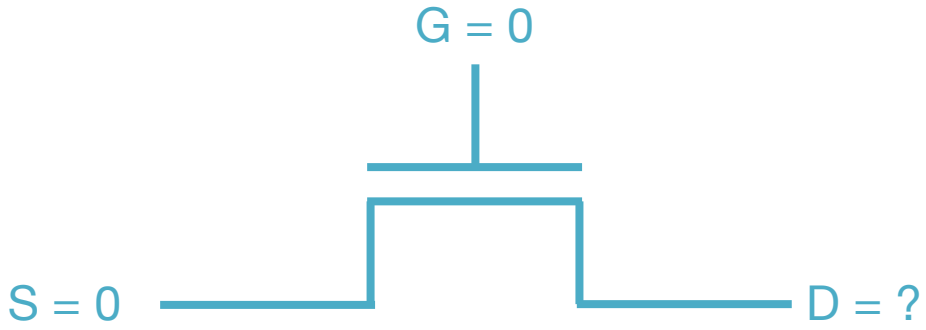
p-type

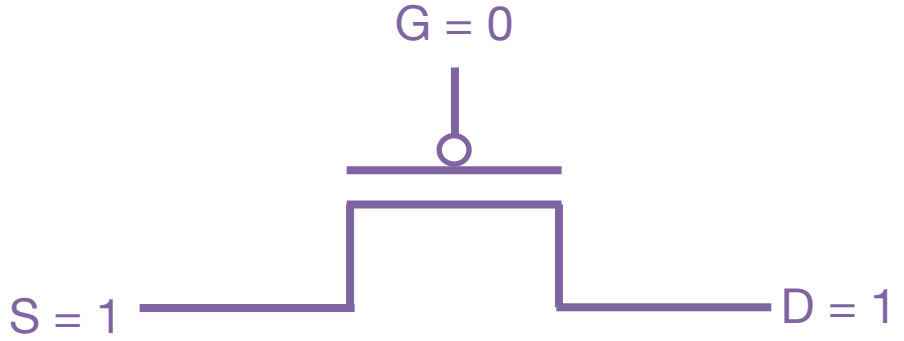
Gate = 0, switch is closed
Gate = 1, switch is open

nFET vs pFET



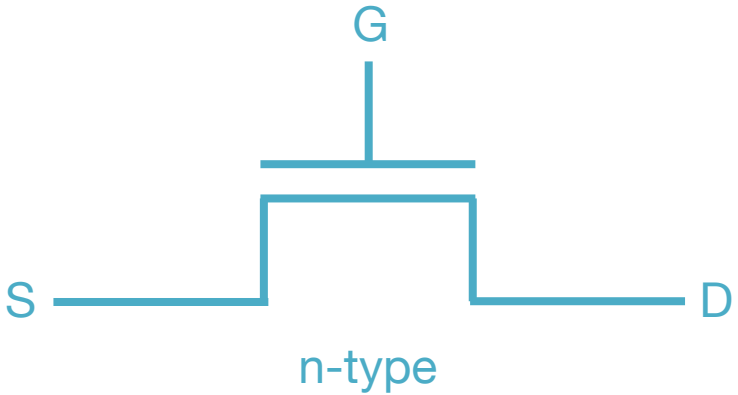






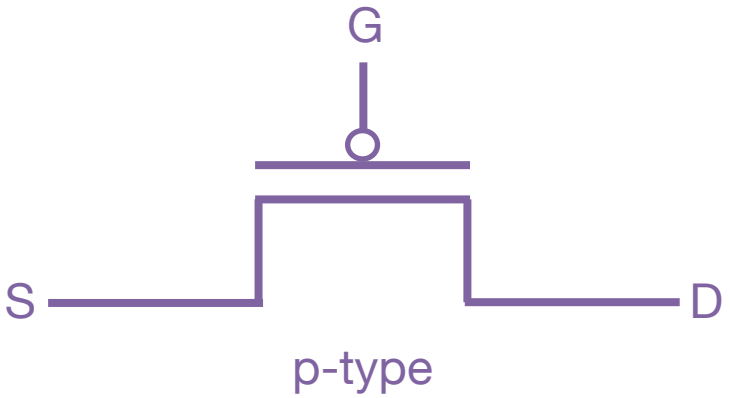
nFET vs pFET

- nFETs are **not good** at passing 1s, so we usually hook up the source of an nFET to a 0
- pFETs are **not good** at passing 0s, so we usually hook up the source of a pFET to 1



n-type

Gate = 1, switch is closed
Gate = 0, switch is open



p-type

Gate = 0, switch is closed
Gate = 1, switch is open

Vdd and Ground

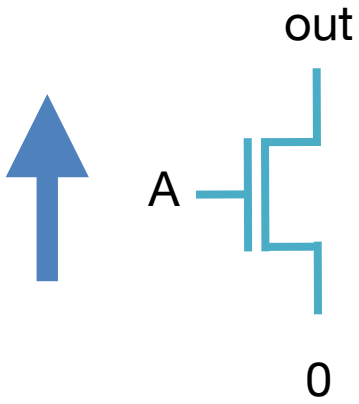
- Vdd = supply voltage pin (Logic 1)
- Ground = Logic 0

Vdd

Ground

Building an Inverter with Transistors

A	out
0	1
1	0

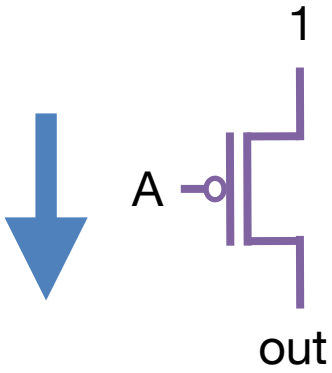


The output is 0 when A is 1

Gate = 1, switch is closed
Gate = 0, switch is open

Building an Inverter with Transistors

A	out
0	1
1	0

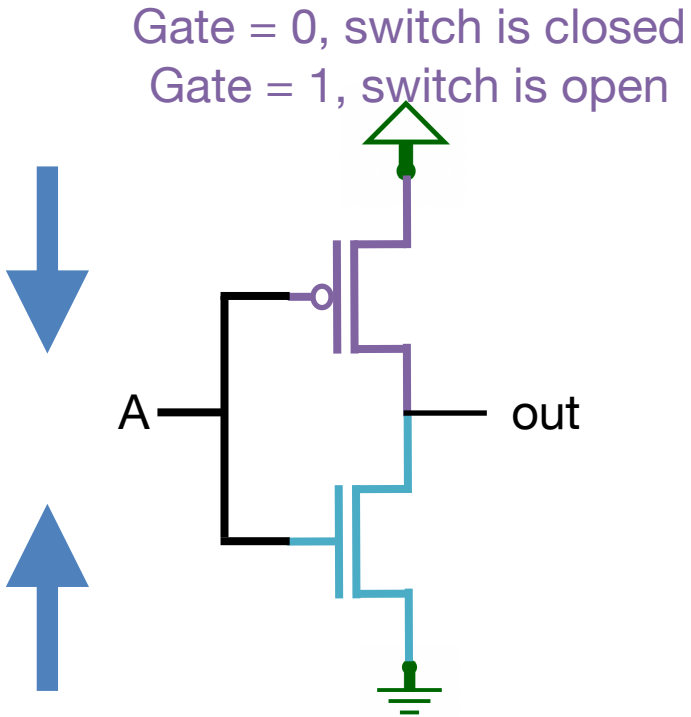


The output is 1 when A is 0

Gate = 0, switch is closed
Gate = 1, switch is open

Building an Inverter with Transistors

A	out
0	1
1	0



Building an Inverter with Transistors

CMOS Inverter

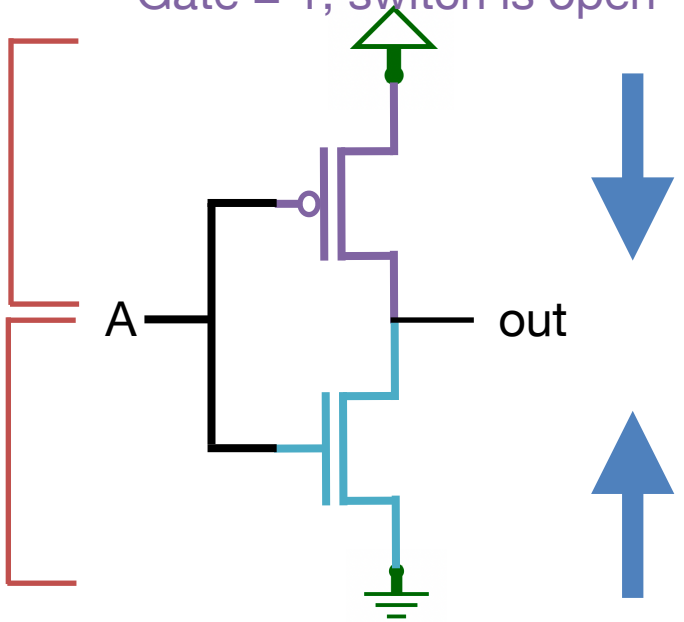
Complementary

A	out
0	1
1	0

Gate = 0, switch is closed
Gate = 1, switch is open

Pull-up network

Pull-down network



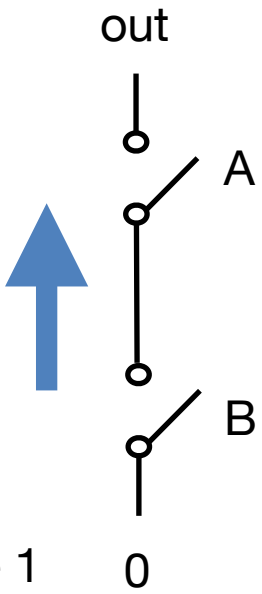
Gate = 1, switch is closed
Gate = 0, switch is open

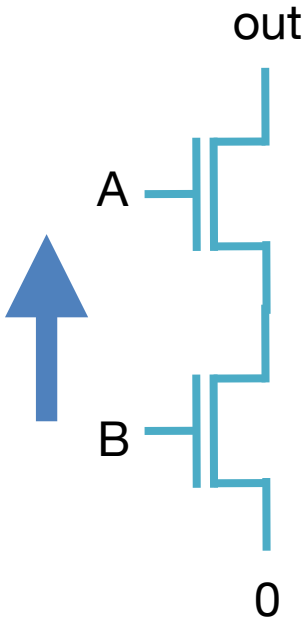
CMOS (Complementary MOS)

- Uses complementary and symmetrical pairs of p-type and n-type MOSFETs to build logical functions
- Consists of a pull-up and pull down-network
- The gates that we see in the remaining slides are all CMOS gates

Building a NAND Gate with Transistors

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0



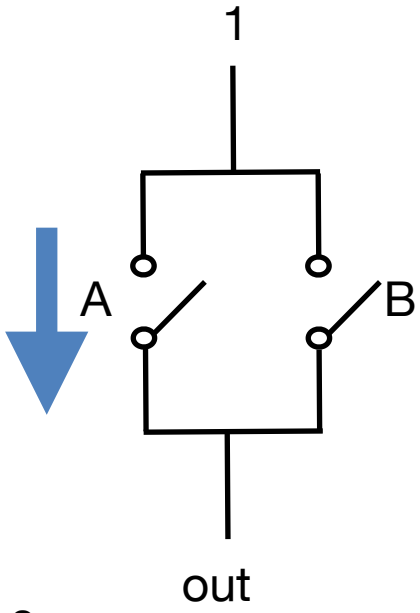


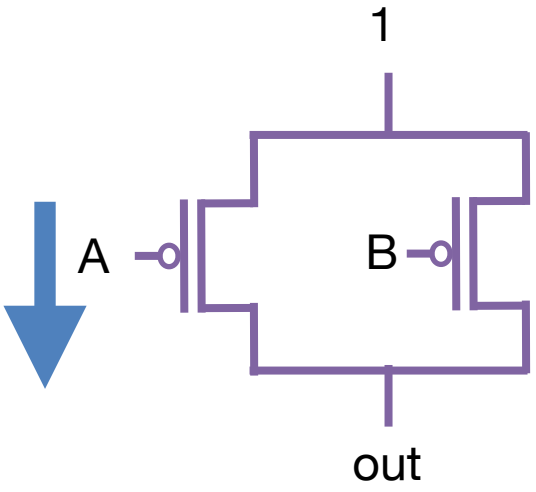
The output is 0 when A and B are 1

Gate = 1, switch is closed
Gate = 0, switch is open

Building a NAND Gate with Transistors

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0



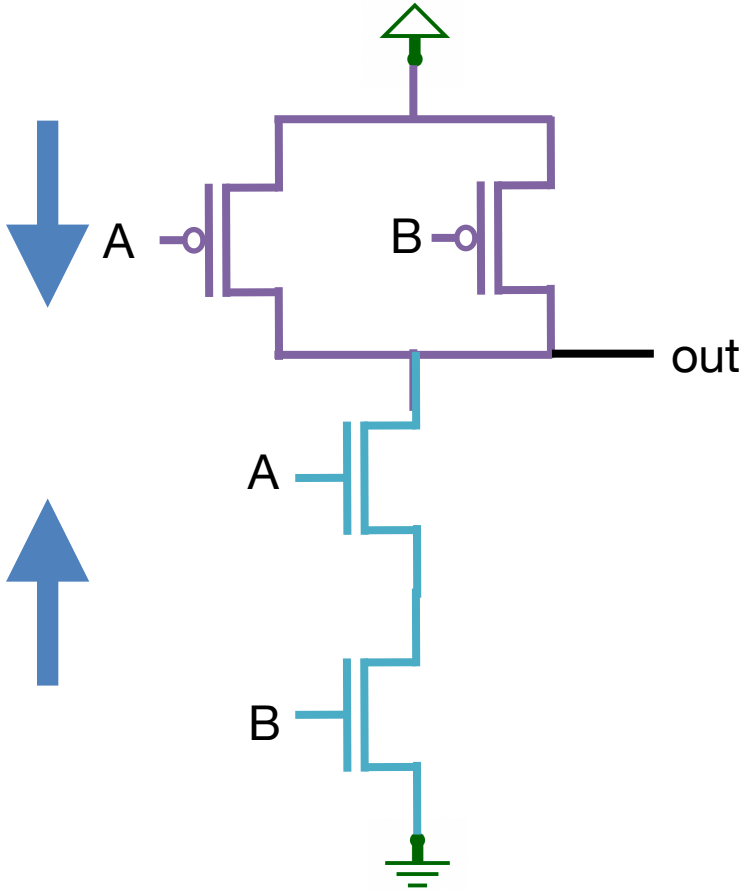


The output is 1 when A or B is 0

Gate = 0, switch is closed
Gate = 1, switch is open

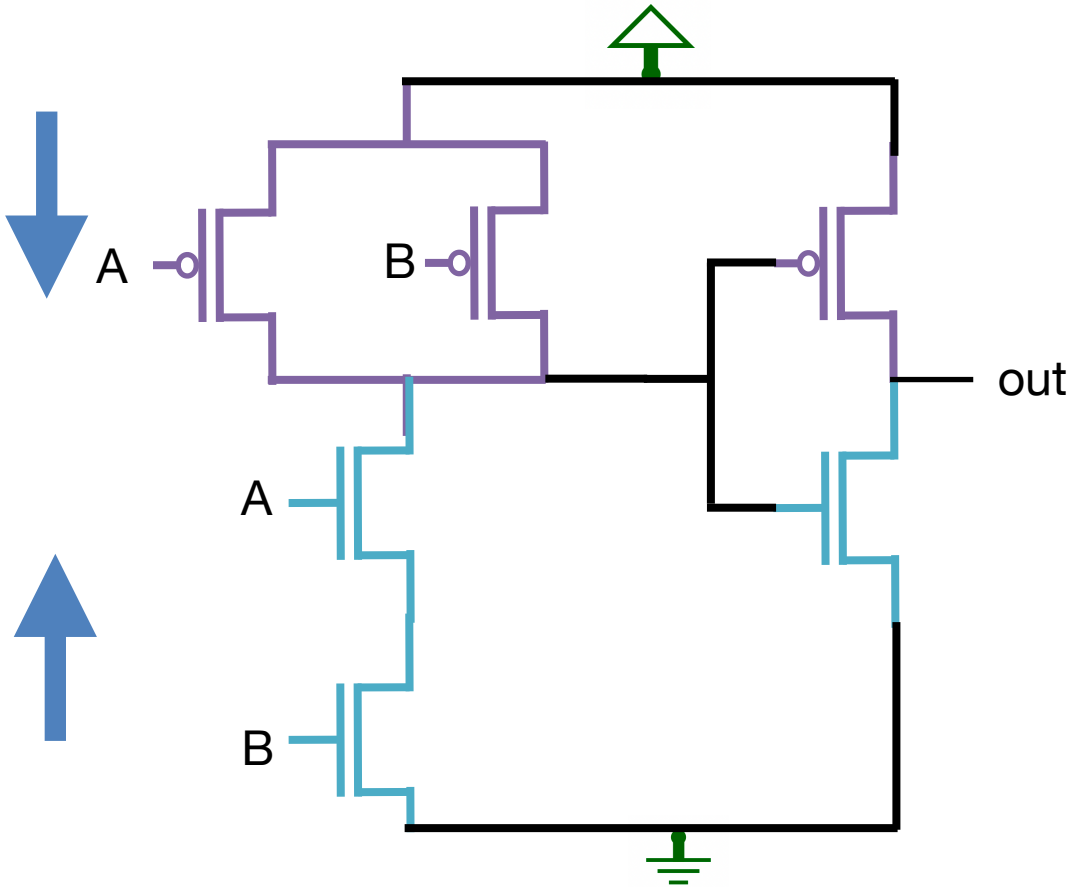
Building a NAND Gate with Transistors

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0



Building an AND Gate with Transistors

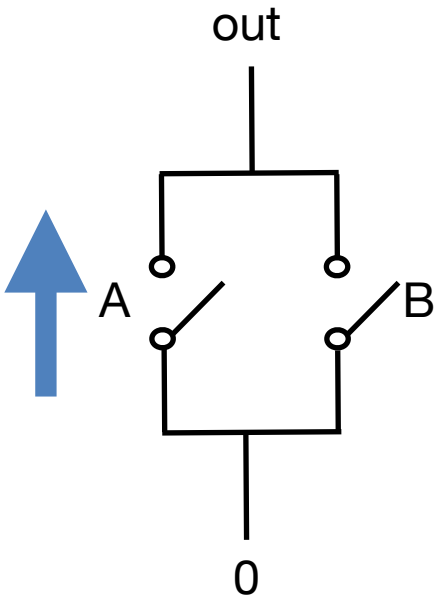
A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

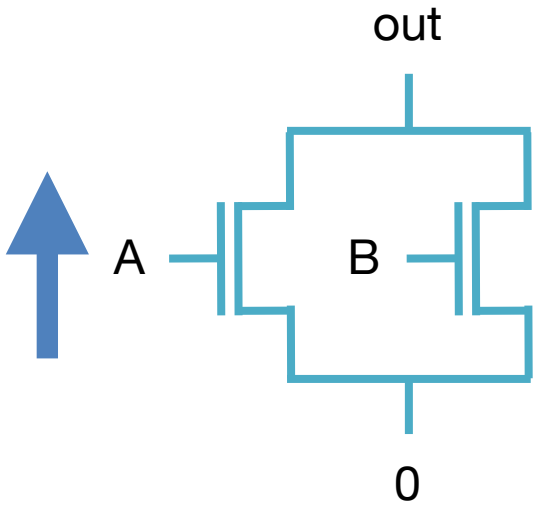


Building a NOR Gate with Transistors

A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

The output is 0 when A or B are 1

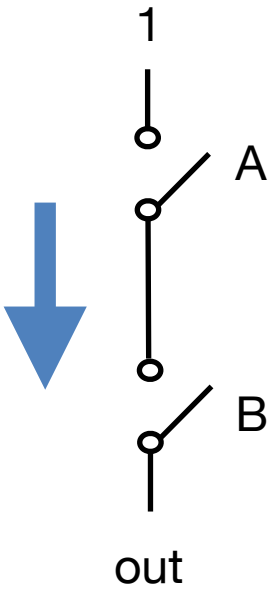


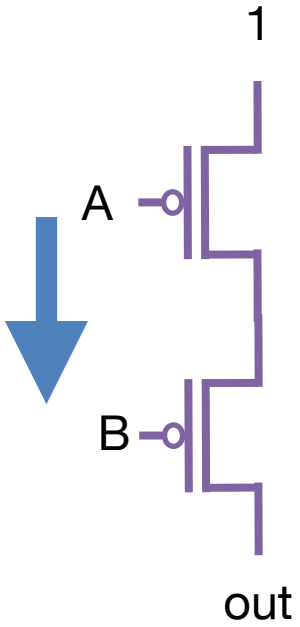


Gate = 1, switch is closed
Gate = 0, switch is open

Building a NOR Gate with Transistors

A	B	out
0	0	1
0	1	0
1	0	0
1	1	0



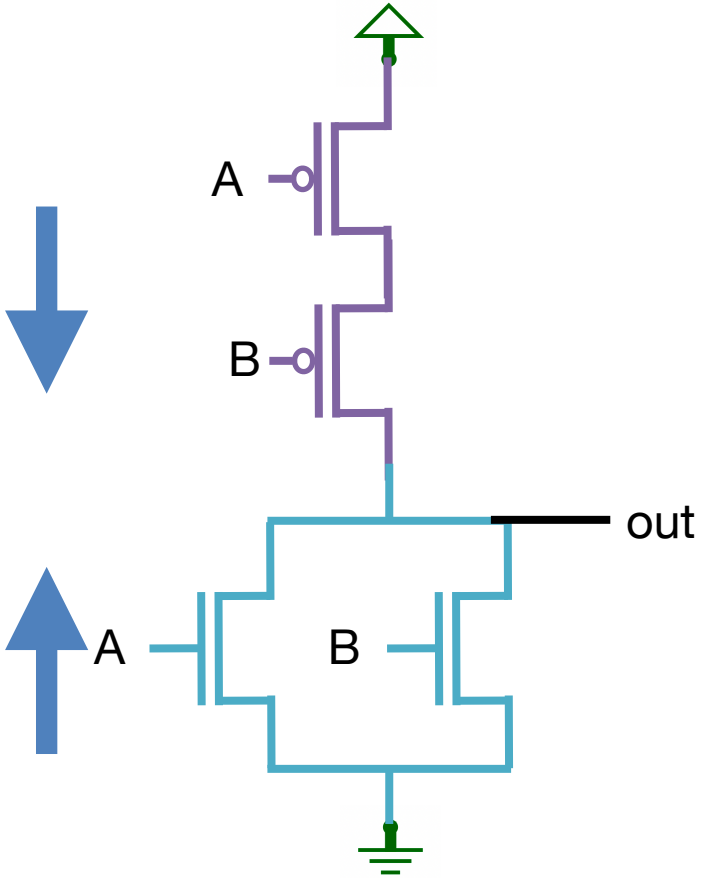


The output is 1 when A and B are 0

Gate = 0, switch is closed
Gate = 1, switch is open

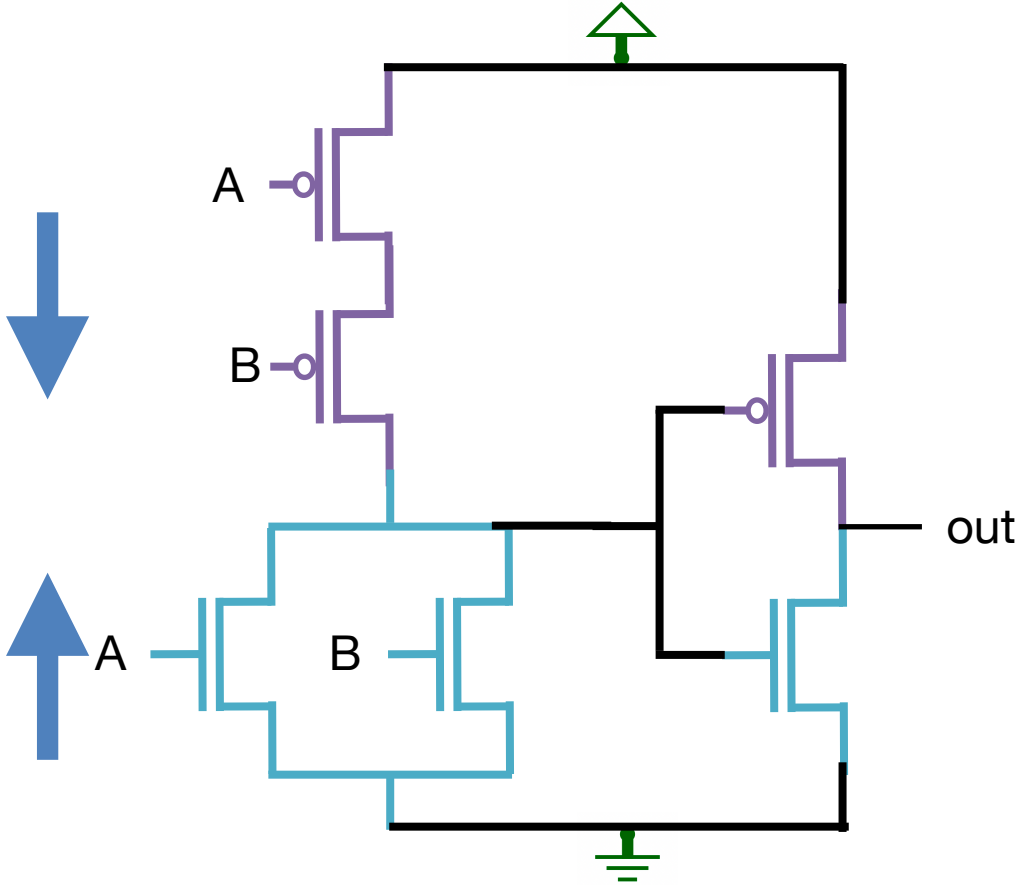
Building a NOR Gate with Transistors

A	B	out
0	0	1
0	1	0
1	0	0
1	1	0



Building an OR Gate with Transistors

A	B	out
0	0	1
0	1	0
1	0	0
1	1	0



Transistors

- Inverter
  - 2 transistors
- NAND gate
  - 4 transistors
- NOR gate
  - 4 transistors
- AND gate
  - 6 transistors
- OR gate
  - 6 transistors

DeMorgan's Law

$$\overline{(A + B)} = \overline{A}\,\overline{B}$$

$$(A + B) = \overline{\overline{A}\,\overline{B}}$$

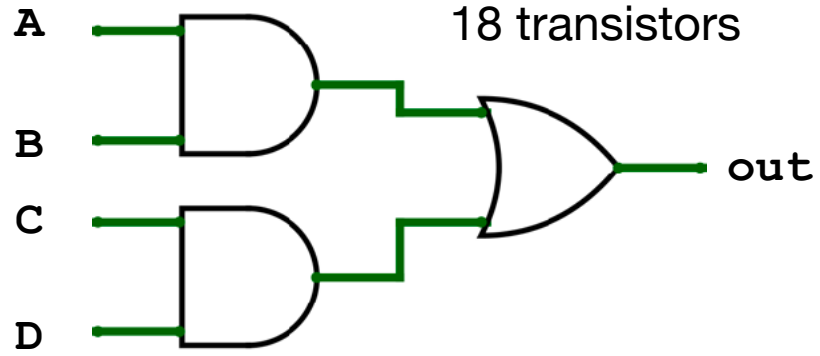
$$AB + CD = \overline{(\overline{AB})\,(\overline{CD})}$$

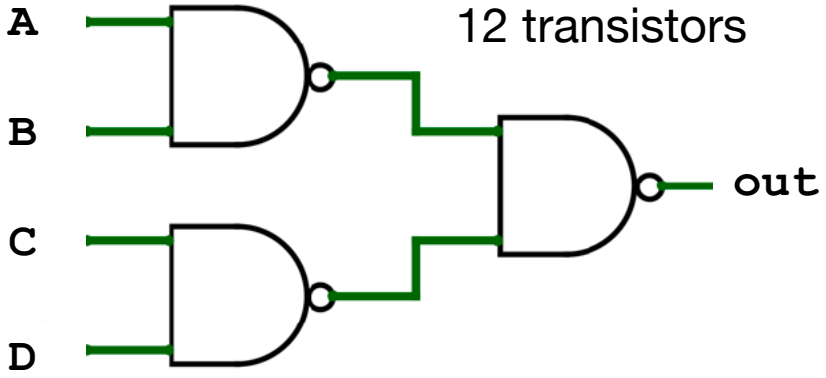
Converting ANDs and ORs to NANDs and NORs

$$out = AB + CD$$

DeMorgan's Law

$$out = \overline{\overline{(AB)}\,\overline{(CD)}}$$

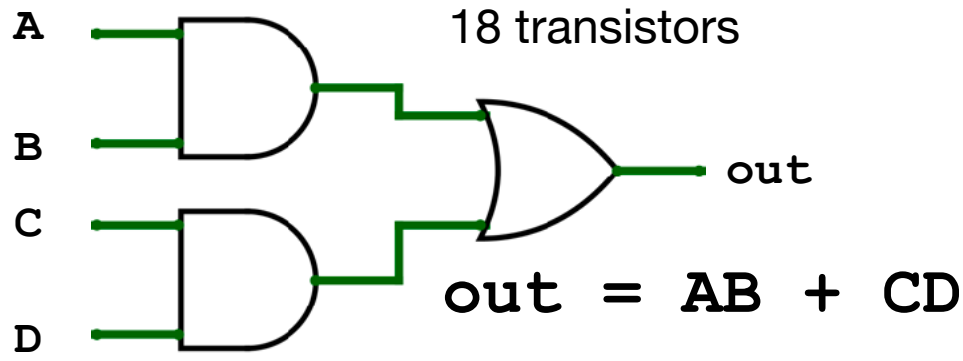


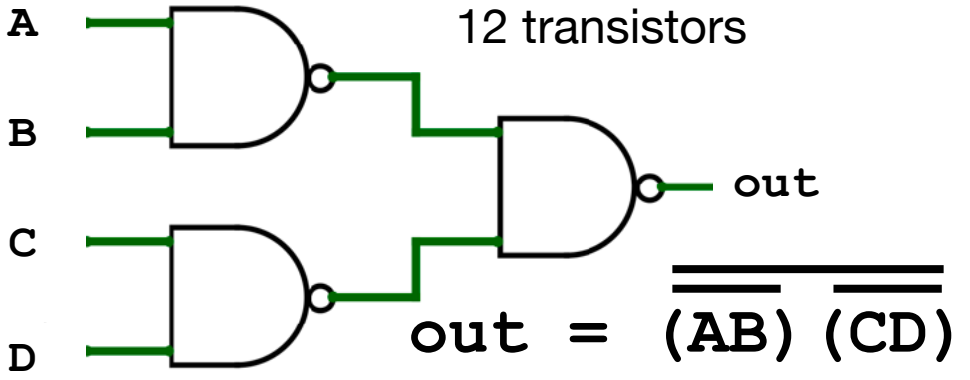


Converting ANDs and ORs to NANDs and NORs

A	B	out
0	0	
0	1	
1	0	
1	1	

A	B	out
0	0	
0	1	
1	0	
1	1	

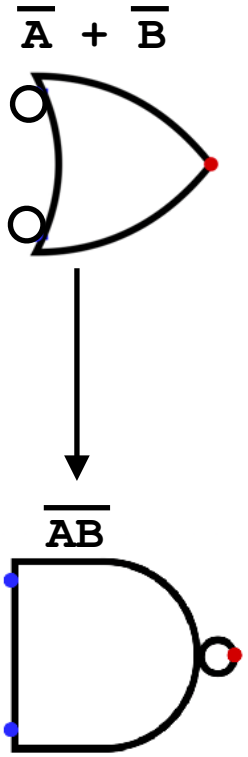


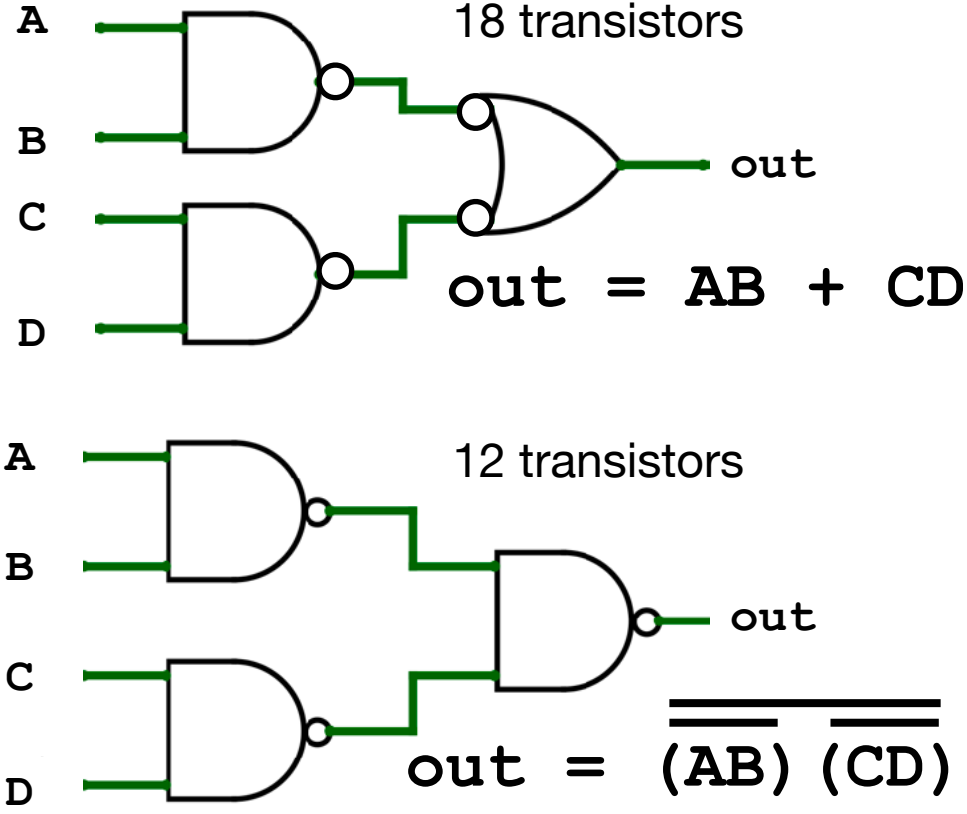


Converting ANDs and ORs to NANDs and NORs

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

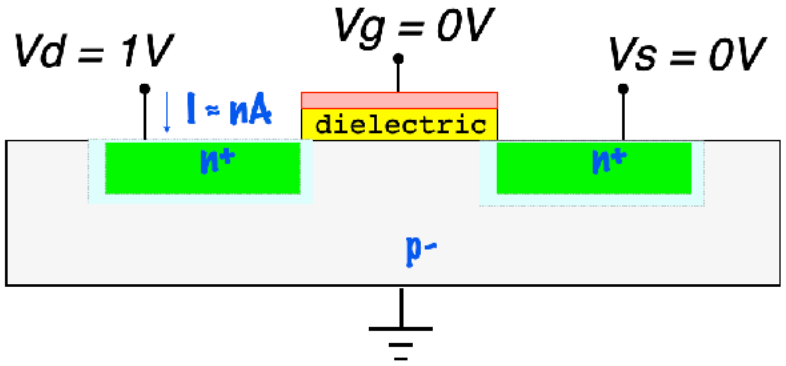




Cross-section of n-type MOS transistor: nFET

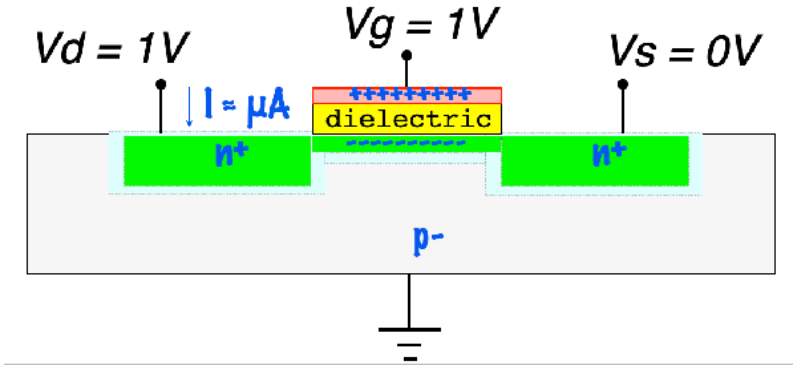
- Three electrical terminals: gate, source, drain
- Vg, Vd, Vs indicates voltages on each node

nFET is off



(I is “leakage”)

nFet is on



Vg = 1V, small region near the surface turns from p-type to n-type.

Simplified Explanation of Transistors

- https://www.youtube.com/watch?v=IcrBqCFLHIY

Summary

- Two types of circuits
  - Combinational
  - Sequential
- Transistors
- CMOS Gates