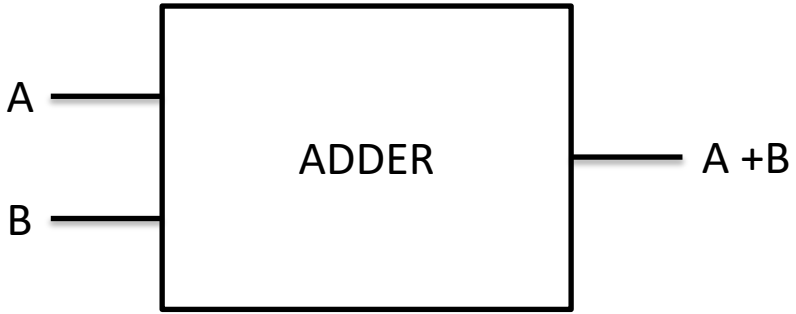
COMPUTER SYSTEMS AND ORGANIZATION

Adders

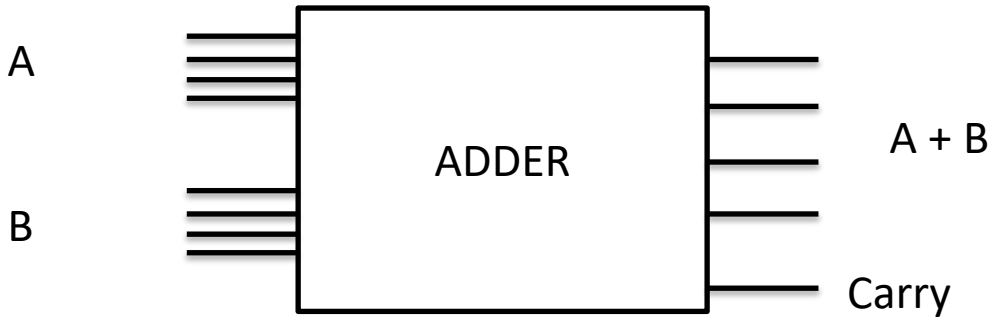
Daniel Graham

REVIEW

THE IDEA



4-BIT ADDER



Great now let's build it with gates.

ADDING

```
1 1 1 1   ← Carries
  0 1 1 1
+ 1 0 1 1
  0 0 1 0
```

Let start by building a half adder something that just adds two bits.

Let's build a truth table.

A	B	A + B	C.out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

We can implement
A + B with an XOR gate
And the C.out (Carry out)
With an AND gate

ADDING

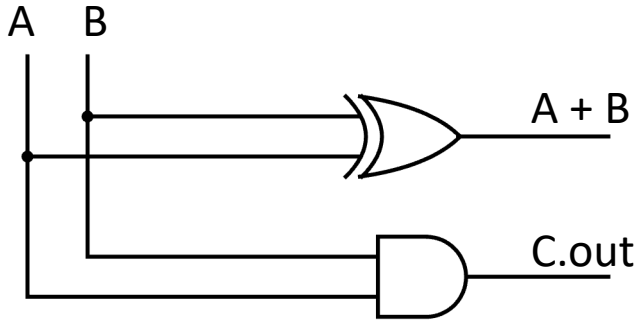
```
1 1 1 1   ← Carries
  0 1 1 1
+ 1 0 1 1
  0 0 1 0
```

Let start by building a half adder something that just adds two bits.

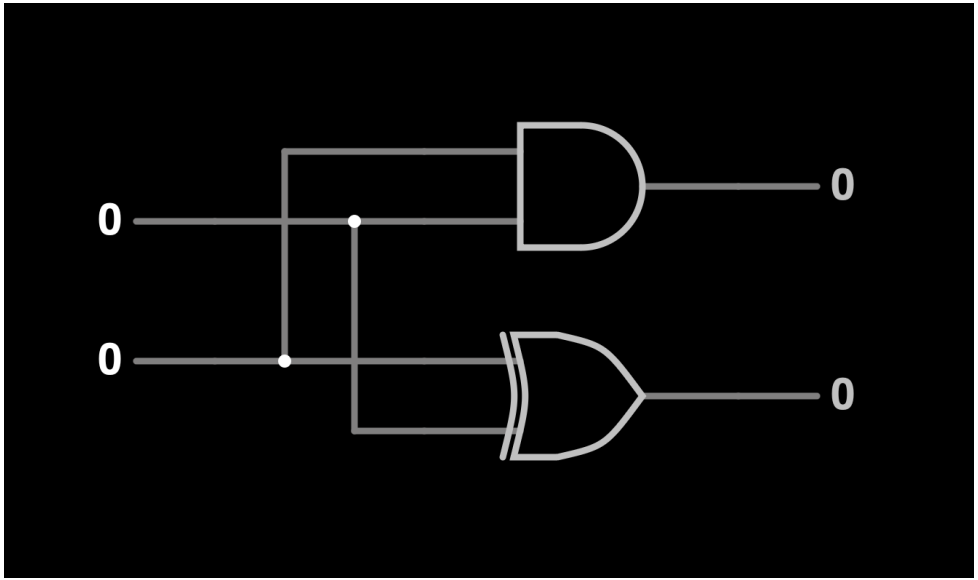
Let's build a truth table.

A	B	A + B	C.out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

We can implement
A + B with an XOR gate
And the C.out (Carry out)
With an AND gate



HALF ADDER DEMO



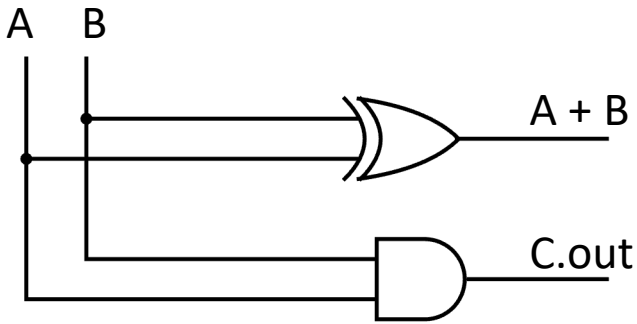
https://tinyurl.com/ygpea8v4

http://www.falstad.com/circuit/circuitjs.html?ctz=CQAgjCAMB0l3BWc0FwCwCY0HYEA4cEMElURTJyBTAWjDACgwE0QMs21KBmANj06VKGKOSZl2rMGlZ8B01sNEIGAGXAZ5vSnkphtbUQDMAhgBsAzlXJQ1GgZJC62HEZVOXrSSAwDu9lykDRx9-fWEOcIDQ8AMwTUDov1iI1kcQ5PitPQBOESiYsDyU8GLiXlswsoQK9JrK0vzgyIMfAFkQOXAZEDR9brS2FAYOrqxKPtquQwxhoA

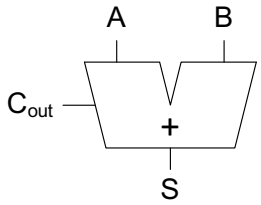
ADDING

```
1 1 1 1   ← Carries
  0 1 1 1
+ 1 0 1 1
  0 0 1 0
```

We can implement
A + B with an XOR gate
And the C.out (Carry out)
With an AND gate



Half Adder

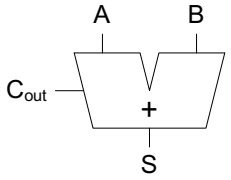


A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = A \oplus B$

$C_{out} = AB$

Half Adder

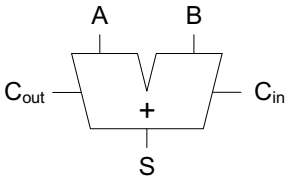


A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = A \oplus B$

$C_{out} = AB$

Full Adder



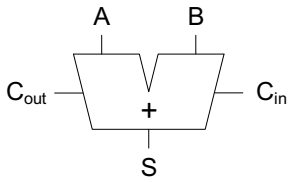
C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$S = A \oplus B \oplus C_{in}$

$C_{out} = AB + AC_{in} + BC_{in}$

```
1 1 1 1   ← Carries
  0 1 1 1
+ 1 0 1 1
-------
  0 0 1 0
```

Full Adder

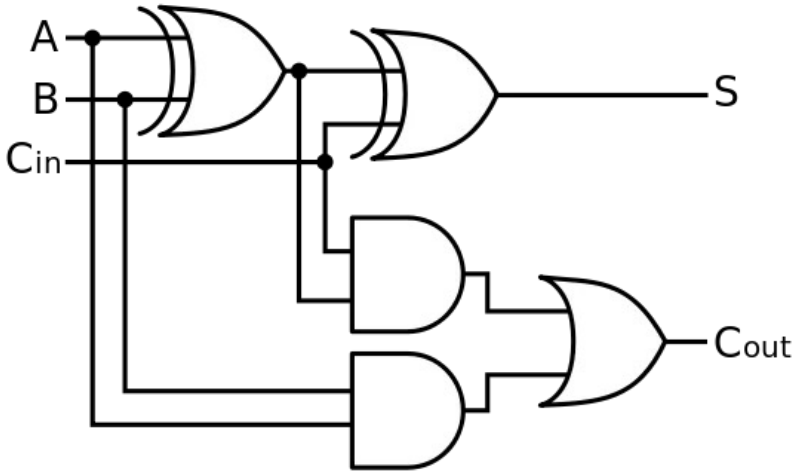


C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$S = A \oplus B \oplus C_{in}$

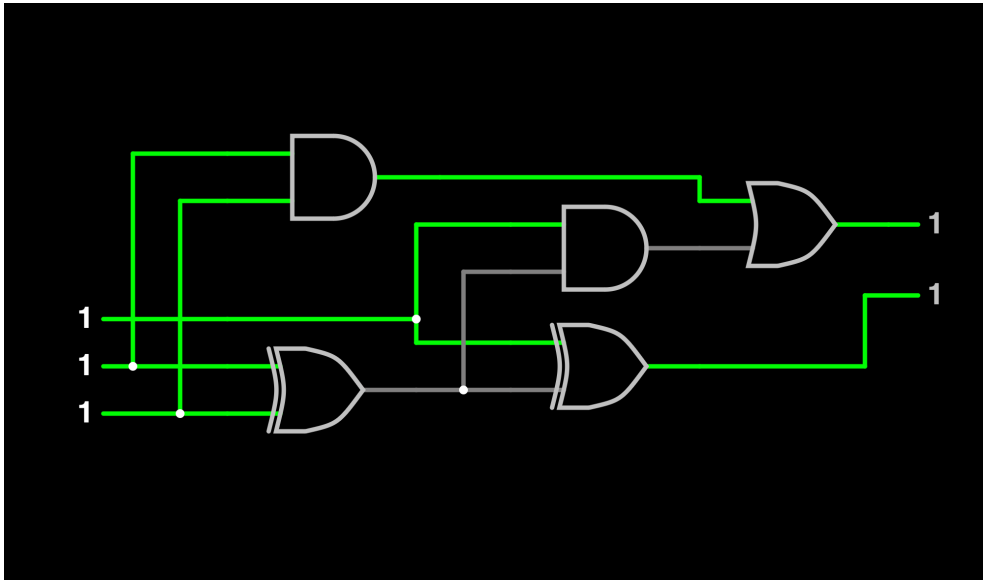
$C_{out} = AB + AC_{in} + BC_{in}$

```
1 1 1 1   ← Carries
  0 1 1 1
+ 1 0 1 1
  0 0 1 1 0
```



C.out has been rewritten to reduce the number of gates needed.

DEMO FULL ADDER

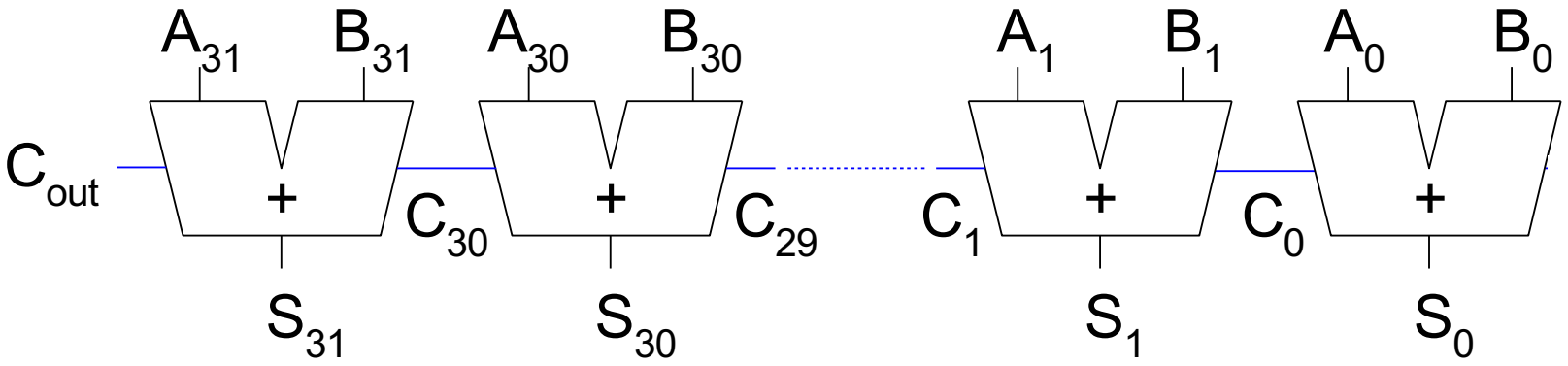


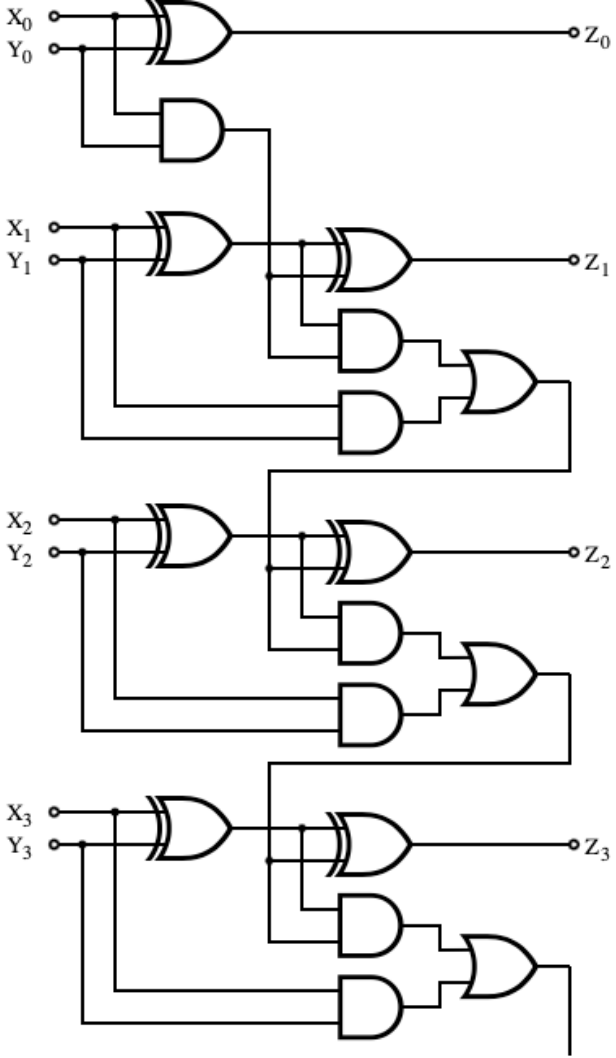
https://tinyurl.com/2cfbbshs

http://www.falstad.com/circuit/circuitjs.html?ctz=CQAgjCAMB0l3BWc0FwCwCY0HYEA4cEMElURTJyBTAWjDACgwE1w1WNsMRPveoeAhExYgAzGIBsPBNLSSOsgd1IMA7jy6buYyBy2R14vdvFTThjROlhJlfrcqWz0rPa1uoR-p59YvGn6sYOw8-s4OdqYY4d54eOAYCfLBSV4AsuQAnNyOIJL+eXwoRvR88WwcFc4hwWDceJRlASCN4PWt9krObcQ2oX1eADKdMnIJg7kgAGYAhgA2AM5U5MPt5ckTFVNzSytIhiO1YaxoE-47C8urNaHHzcc1HfcdSQnOKYnJCpUt1uA5YzBQHOXTAnQmfgfH6DBDYVzdIxwhHSZE8SDvBiZNEYDH5C544rCZhNO5pN5fZSrYlkf5geEgT706T2IQibhMqIIQF5VmqIA

RIPPLE CARRY ADDER

Next let's build a full adder





RIPPLE CARRY ADDER

1 1 1 1 ← Carries

0 1 1 1

+ 1 0 1 1

1 1 1 0

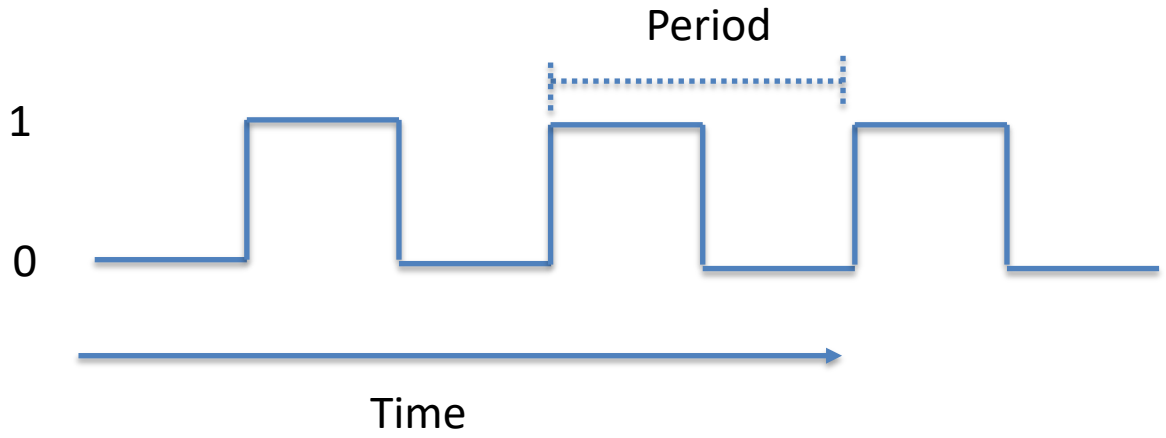
1. Clocks
2. Latches
3. Flip Flops

CLOCKS

A clock is something that produces a periodic signal

Period is length of time for one clock cycle

Example

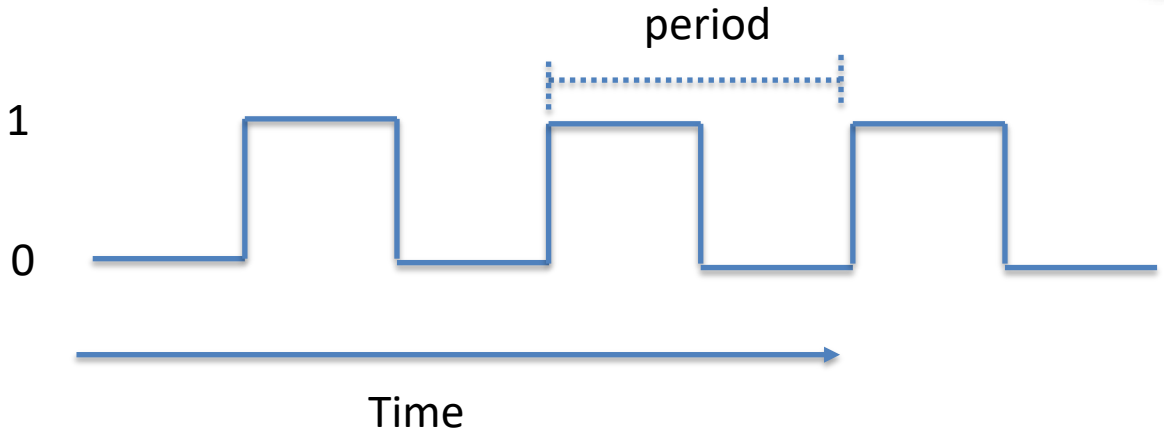


CLOCKS

Clock frequency Intel Core i-9 3.0GHz

Frequency = 1/ period

Period = 1/frequency

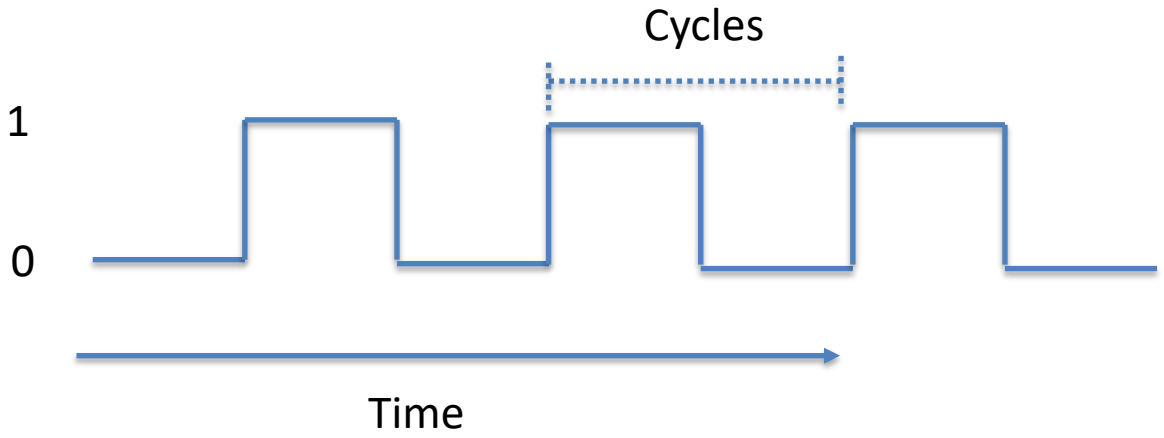


CLOCKS

Clock frequency Intel Core i-9

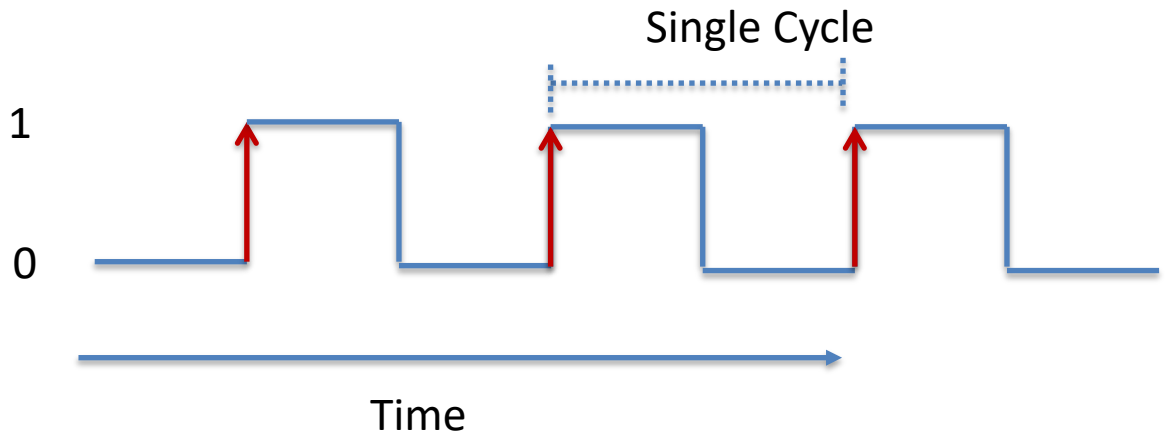
3.0GHz. = $3*10^9$ = 3,000,000,000 cycles per second

Thank is fast.



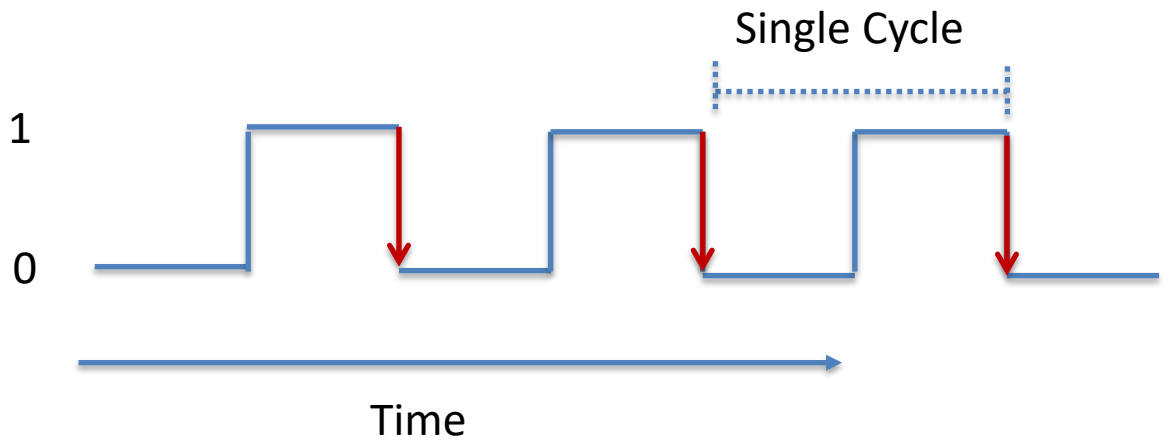
CLOCKS EDGES

Rising Edge (Also called positive edge)



CLOCKS EDGES

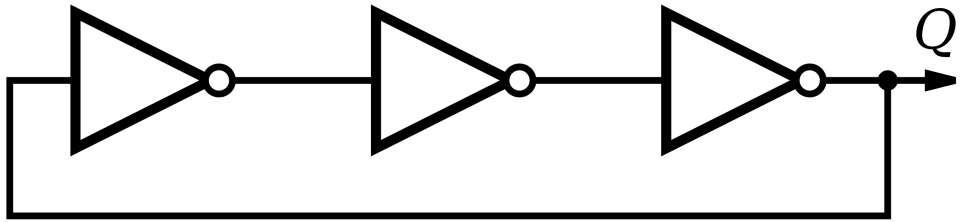
Falling Edge.



We will build a single cycle machine it will complete all the computation in a single cycle

USING RING OSCILLATORS TO GENERATE CLOCKS

A clock is something that produces a periodic signal



Frequency = 1/(2*t*n)

Where t is time delay of an inverter and n is number of inverters

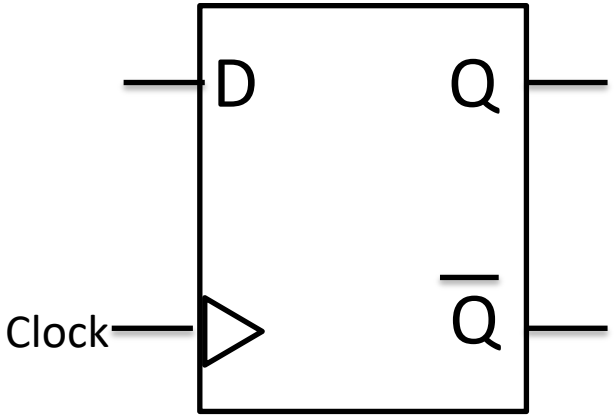
Let's walk through an example and assume that Q starts off as 0. Draw wave form that results.

THE D FLIP FLOP

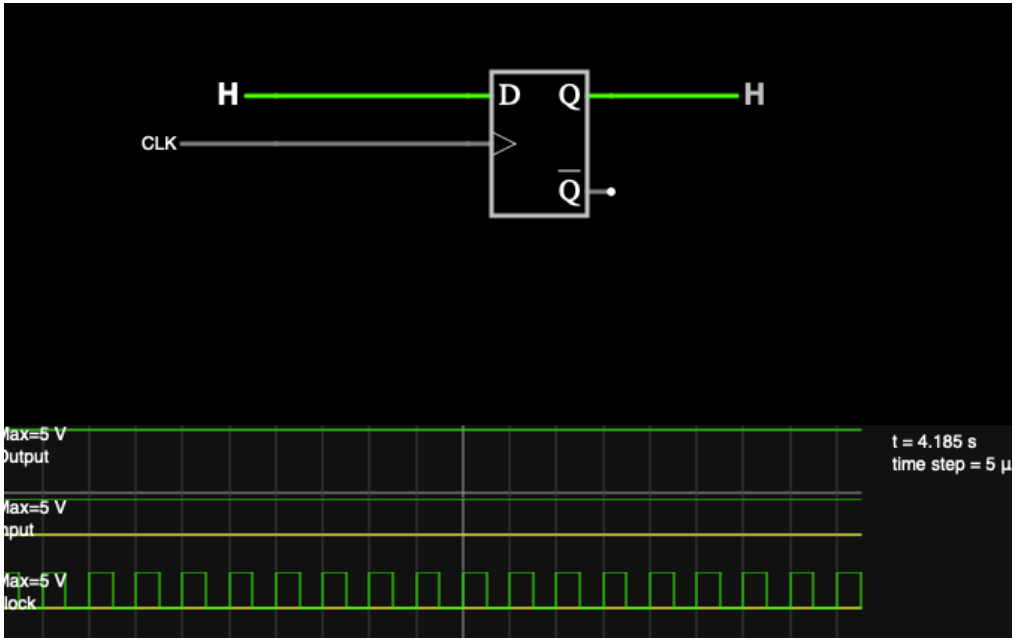
STORING SINGLE

Goal

1. Understand the behavior of a positive edge-triggered D flip-flop.
    - How do we store a bit
    - What happens when the clock changes
    - What does it mean to be a positive edge triggered flip flop
    - What is Q and $\overline{Q}$



BUILT SIMULATOR VERSION



Use this link to experiment with the flipflop during lecture. Try different things as see how it works

https://tinyurl.com/2dhk5kvg

http://www.falstad.com/circuit/circuitjs.html?ctz=CQAgjCAMB0l3BWcMBMcUHYMGZIA4UA2ATmIxAUgoqoQFMBaMMAKDASUPxABZsUQGPD178oFFgHcQXPKIE88VPgMhSZ3HoRGLl2qCwAyvJfN6KzVCADMAhgBsAznWpqASib064vfRAFgPijQSMFIVDAILACygpA6YkrKYlRhLAD21PrKkKSu0BBWIADyAK4ALgAOFRngMiI5eeGw8GSECIQo4SABINggAJYAdtXltQLZvLnE+fC5GO2d3QIC-QDG9ulrANYsQA

D FLIP FLOP

Two inputs and Two outputs

Inputs

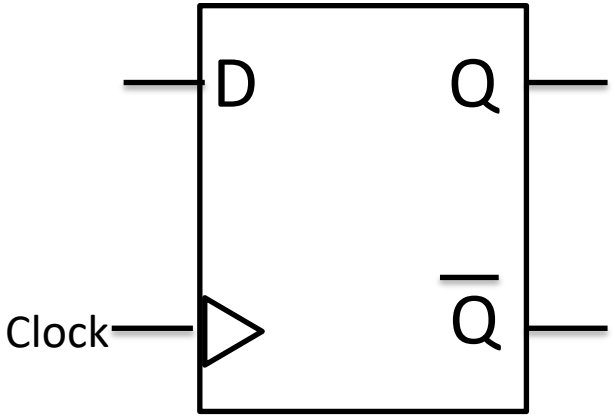
D – Data in (The single bit that we want the flip flop to store)

Clock – the output wave from the clocks we discussed earlier

Output

Q – The value of the internal state. 1 if the internal state is one and zero if the internal state is 0

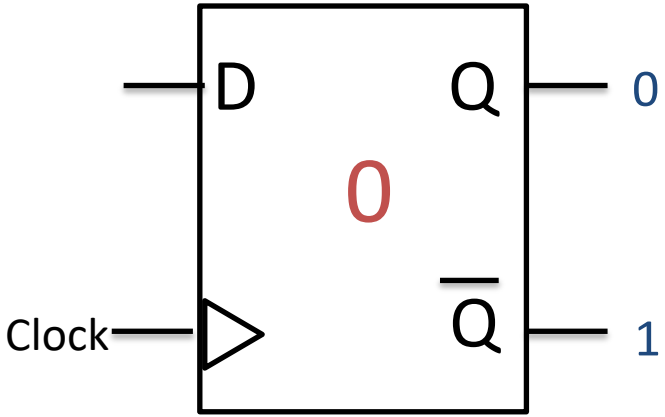
Q (bar) – Is the inverse of Q.



STATE OF A D FLIP-FLOP

Let assume that the D flip state is where its internal state is zero

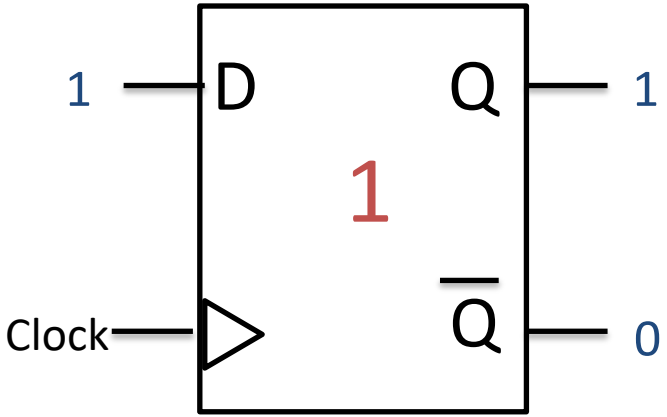
What is the value of Q and Q bar?



SETTING A D FLIP-FLOP TO 1

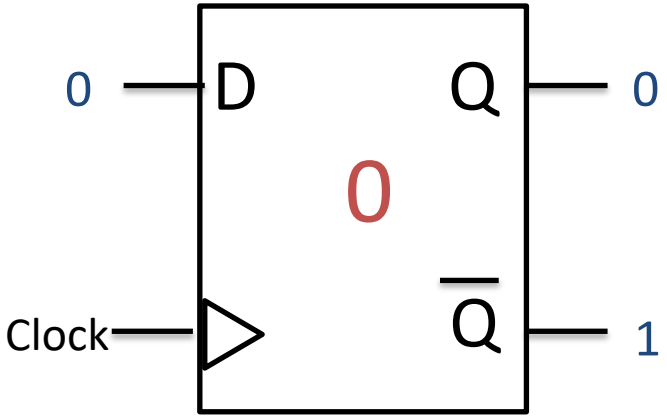
Now let's set the internal state of the Flip-Flop to 1 by setting D to 1

Notice that the outputs Q and Q (bar) also change.



SETTING A D FLIP-FLOP TO 0

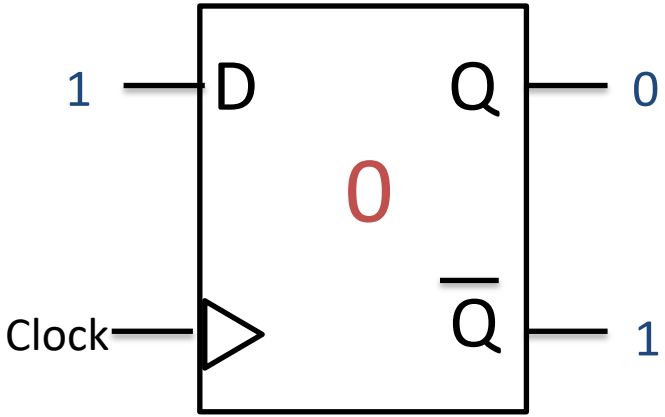
We can set the flip back to zero by setting D to **0.**
Notice that Q is now 0 and Q (bar) is now 1.



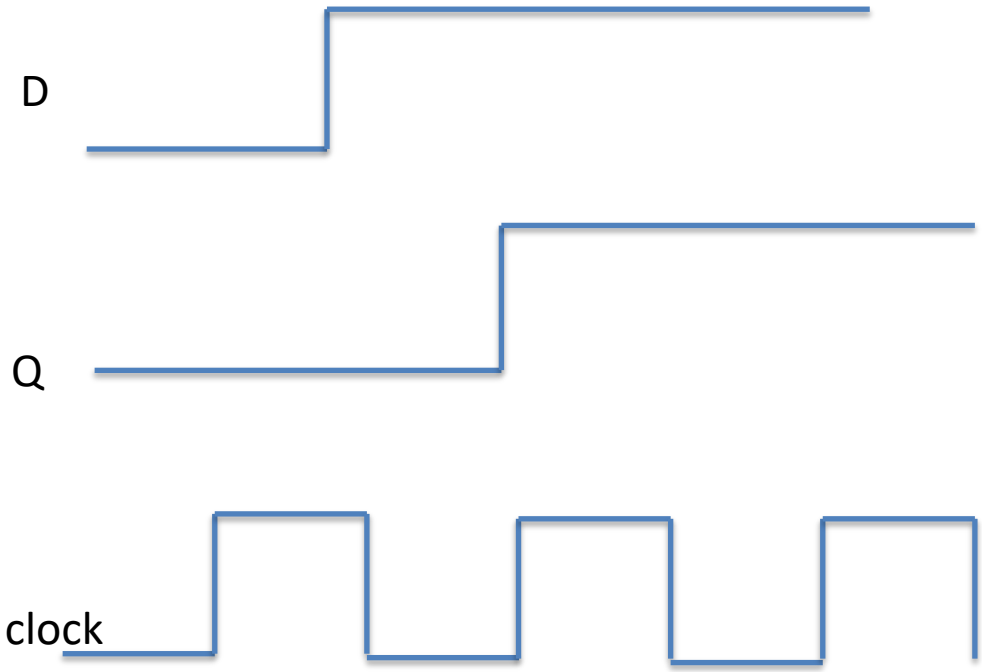
WHAT ABOUT THE CLOCK

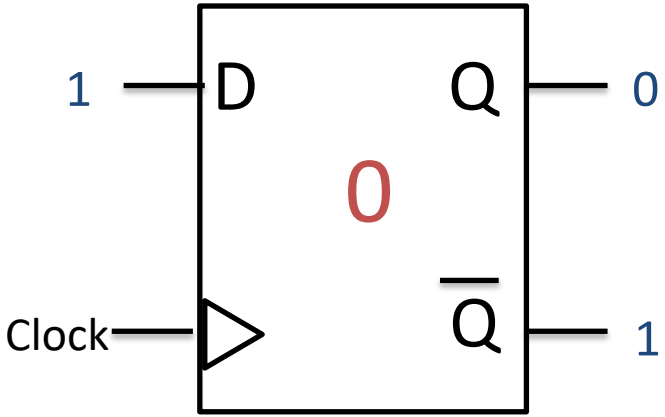
The Clock determines timing. Specifically

1. When the values change
2. How long the values remain. Let's look at an example. Where goes from 0 to 1



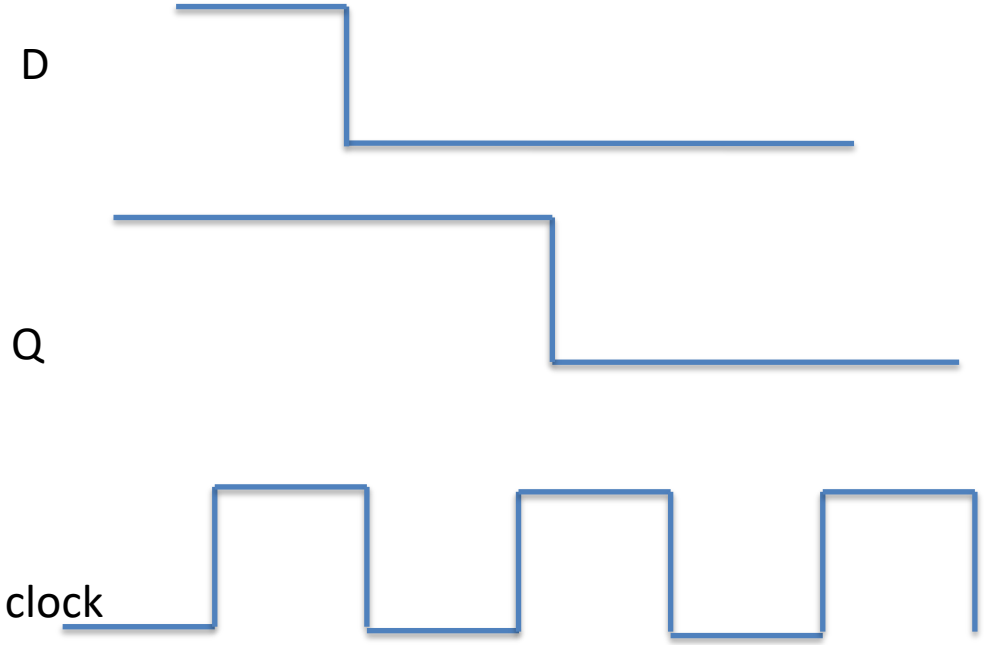
WHAT ABOUT THE CLOCK

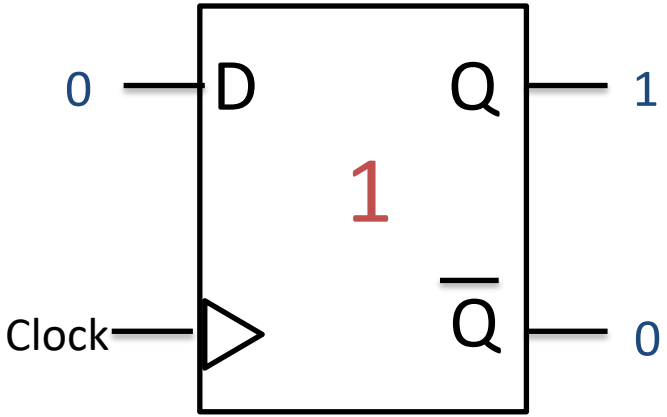




Notice that Q doesn't change until the rising edge

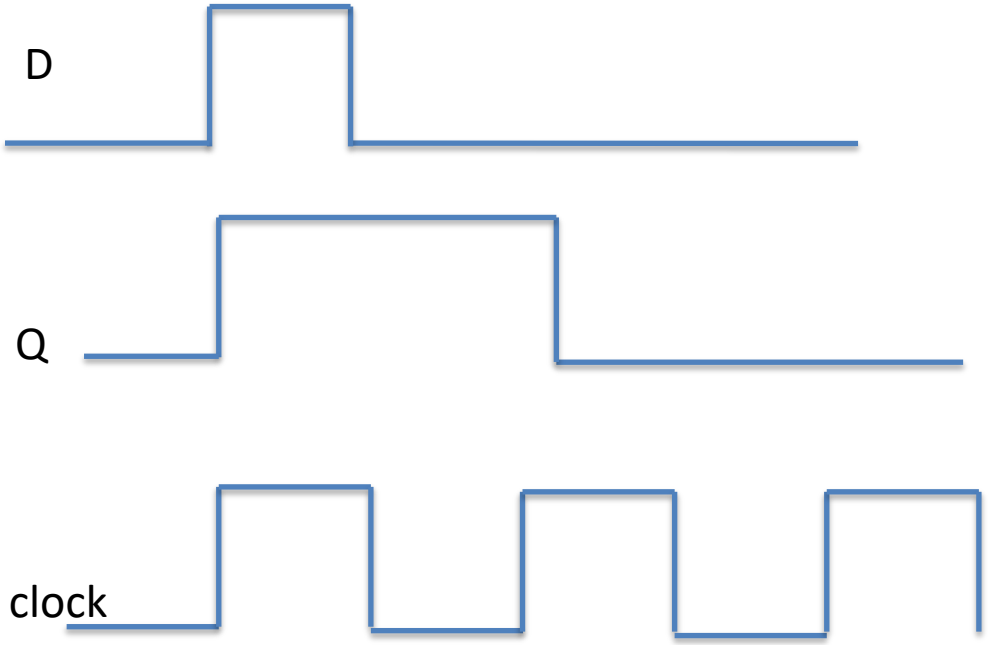
WHAT ABOUT THE CLOCK

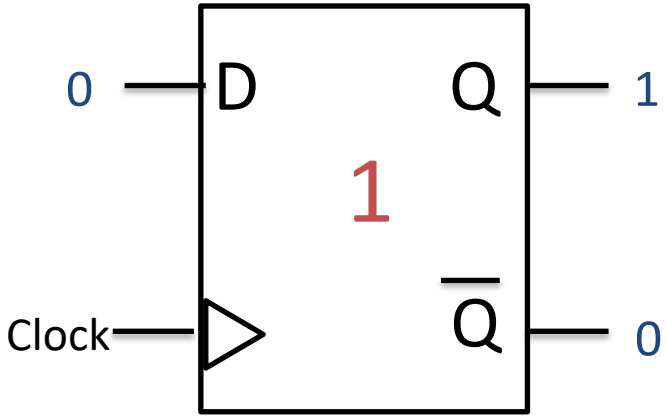




Notice that Q doesn't change until the rising edge

THE FLIP FLOP HOLD HOLDS THE VALUE FOR A CLOCK CYCLE





BUILDING A REGISTER FROM FLIP FLOPS

BUILDING A REGISTER FROM FLIP FLOPS

What is a register?

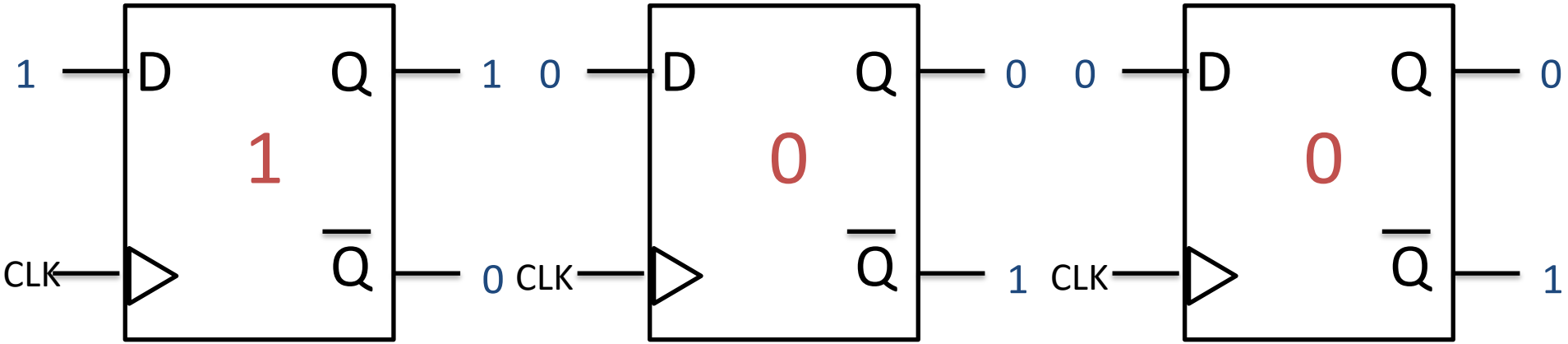
Is a memory unit that stores bit for 1 or more or more clock cycles

Examples of things we can store in a register

The number 5 in binary 101 (This would need a three-bit register)

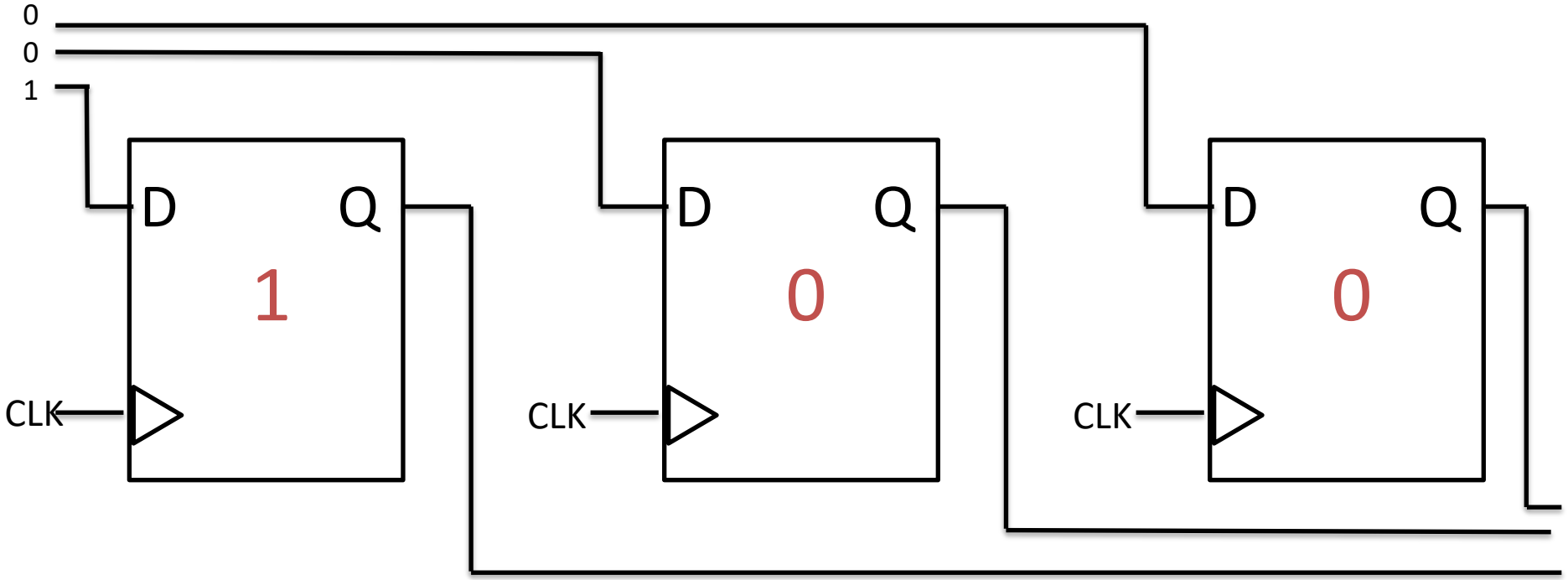
Who could we build a 3-bit register?

BUILDING A REGISTER FROM FLIP FLOPS



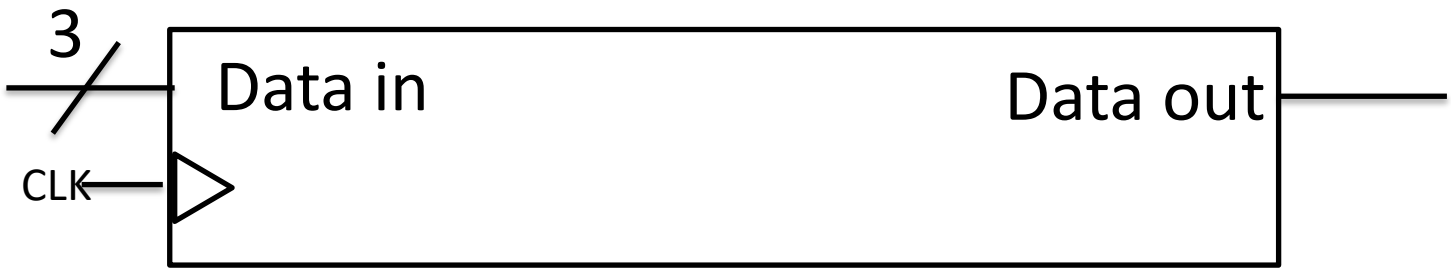
Who could we build a 3-bit register? We can build a 3 bit register with three flip-flops

BUILDING A REGISTER FROM FLIP FLOPS



Removed Q (bar) for reability

REGISTER SYMBOLS



3-BIT COUNTER

Let's put it all together and build a 3-bit counter

Circuit that counts from

000,

001,

010,

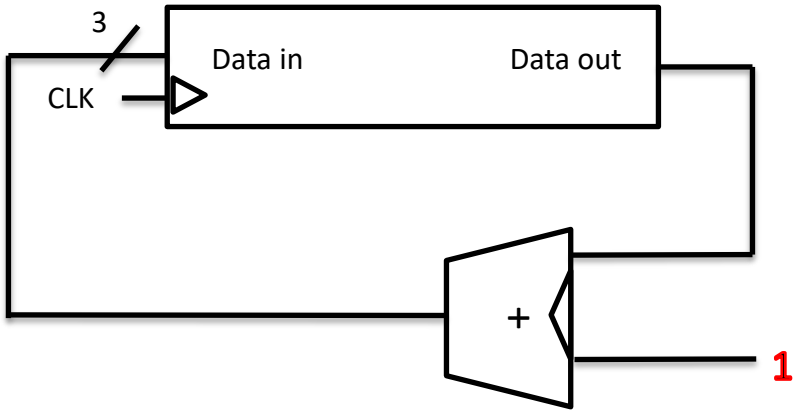
011,

100,

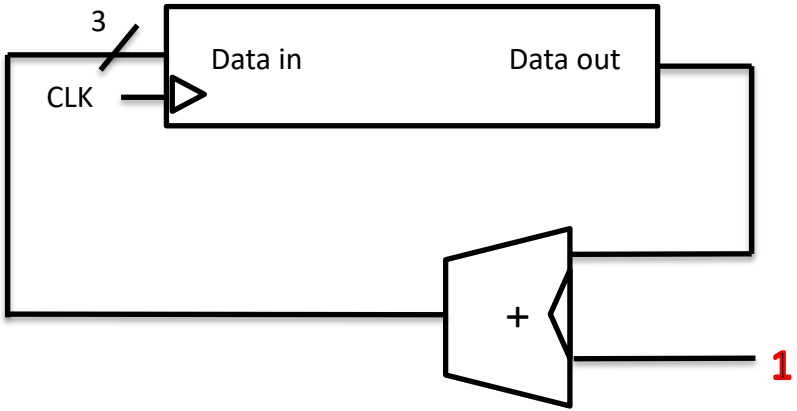
101,

110,

111



DRAWING 3-BIT COUNTER OVER SERVERAL CYCLES



Draw the Timing Diagram

EXAM QUESTIONS SPRING 2023

11. [4 points] The register that we discussed in class (the positive-edge-triggered D flip-flop) has inputs `D` and `clock` and output `Q`. If `Q` is 1 and the `clock` is transitioning from 1 to 0, which of the following is true? *Fill in the circle completely for all that apply.*

    ◯ `D` must also be 1 (not 0)

    ◯ `D` may be 0 or 1

    ◯ `Q` will transition from 1 to 0 with the clock

    ◯ `Q` will transition to the value of `D` when the clock transitions to 0

12. [4 points] To build a 4-bit counter circuit, we could directly connect the outputs of the circuit back to the inputs without the need of a register.

    ◯ True

    ◯ False

11. [4 points] The register that we discussed in class (the positive-edge-triggered D flip-flop) has inputs `D` and `clock` and output `Q`. If `Q` is 1 and the `clock` is transitioning from 1 to 0, which of the following is true? *Fill in the circle completely for all that apply.*

- ◯ `D` must also be 1 (not 0)
- ◯ `D` may be 0 or 1
- ◯ `Q` will transition from 1 to 0 with the clock
- ◯ `Q` will transition to the value of `D` when the clock transitions to 0

B

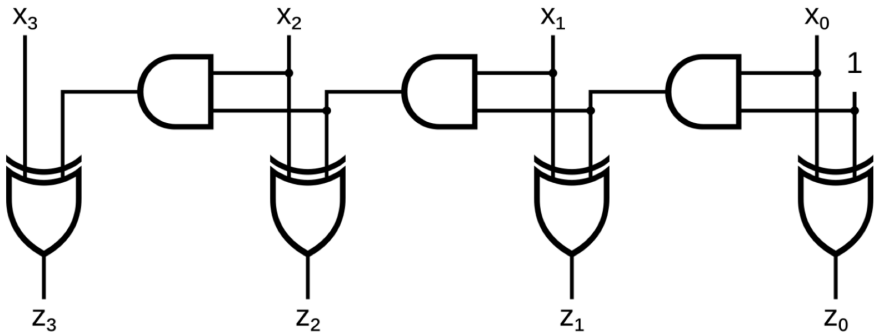
12. [4 points] To build a 4-bit counter circuit, we could directly connect the outputs of the circuit back to the inputs without the need of a register.

- ◯ True
- ◯ False

False

Page 5: Circuits

10. [16 points] In class, we discussed a 4-bit increment circuit below that added 1 to the input.



How can we change this circuit to instead increment by 2, i.e., `x += 2`? Draw the new circuit below. *Note: you should not use more gates than the original circuit.*