Lecture 11 - Digital

Announcements

Homework 4 - Due Wednesday March 6 - Computational HW
Progress Report 3 - Due Monday March 4
Progress Report 4 - Due Monday March 11
Homework 5 - Due Wednesday March 20

Reading Now:

Practical Electronics:
Chapter 12 - Digital Electronics - 12.1.1 + 1.3 +
12.2 - Logic Gates
12.6 - Flip-Flops - 12.6.1, 12.6.2, 12.6.3, 12.6.5

If you need more on implementing circuits on breadboards, read:
Exploring Raspberry Pi, Chapter 4, pages 121, 122.

Exploring Raspberry Pi:
Chapter 4 - pages 133 - 138 - Transistors as switches.
pages 141 - 150 - Logic Gates.
Chapter 5 - pages 173 - 176 - Programming RPi with Python
Chapter 6 - pages 219 - 227 - Interfacing RPi.

Week of Mar 6:

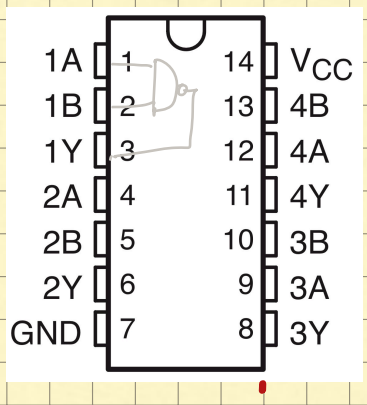
Practical Electronics:
Chapter 8 - Operational Amplifiers - 8.2, **8.3**, 8.4, 8.5, 8.8, **8.17,** 8.18
Exploring Raspberry Pi:
Chapter 4 - pages 152 - 155 - Opamps

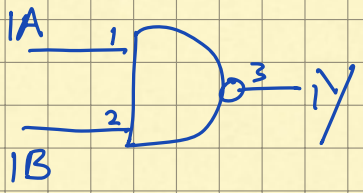
Week of Mar 20:

Practical Electronics:
12.7 - Counters - 12.7.1, 12.7.2
12.8 - Shifters - 12.8.1, 12.8.2, 12.8.3
Exploring Raspberry Pi:
Chapter 8 - pages 330 - 336 - SPI, 74HC595

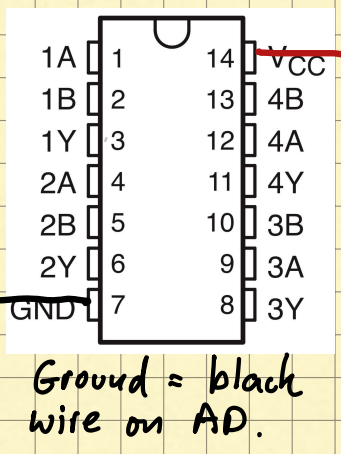
Logic Chips

From 74HC00N Data sheet:





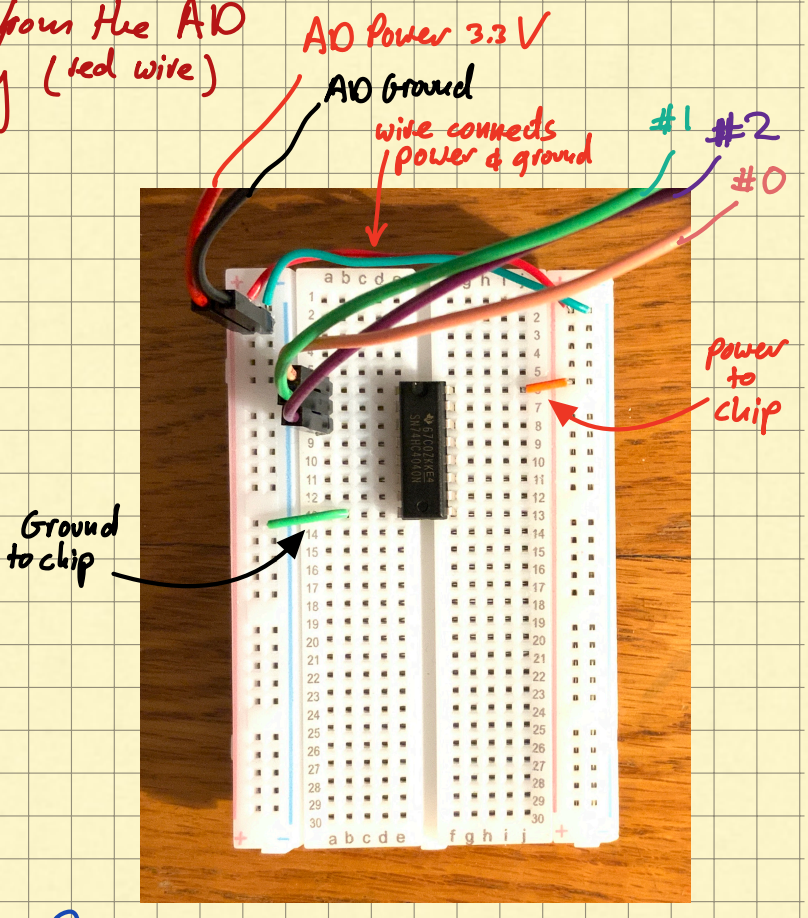
Every device (i.e. logic chip) needs power and ground!!



$V_{cc} = 3.3V$

This comes from the AD power supply (red wire)

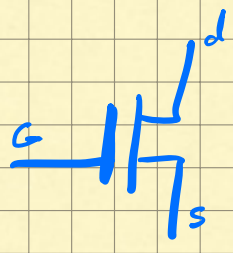
Ground = black wire on AD.



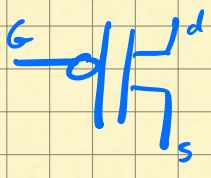
Physical setup to test chip.

CMOS Logic Gates

CMOS logic gates often use simplified schematic diagrams, leaving the details to the chip designer.



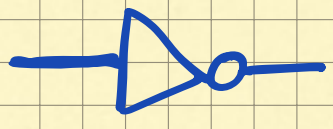
N-Channel conducts when gate is "high"



P-channel conducts when gate is "low"

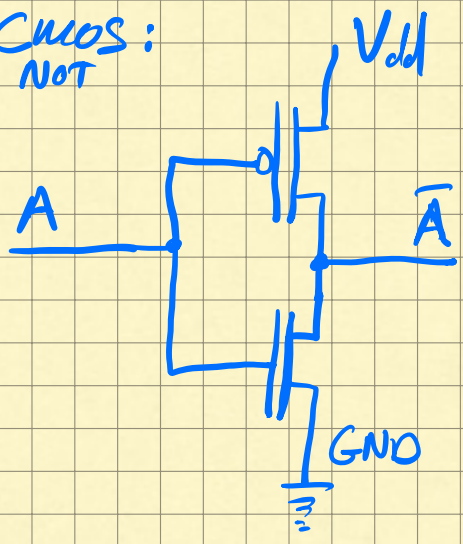
NOT or Inverter:

$A \rightarrow \bar{A}$



A	$\bar{A}$
0	1
1	0

CMOS NOT:



AND Gate

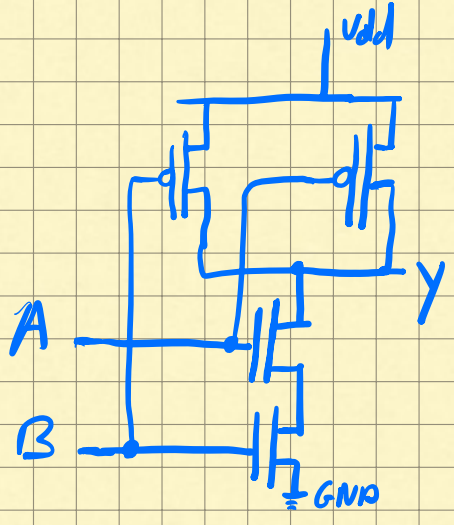
$Y = A \cdot B$

NAND

$Y = \overline{A \cdot B}$

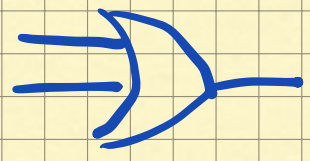
A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

CMOS NAND Gate



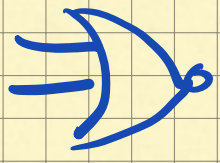
OR GATE

$Y = A + B$



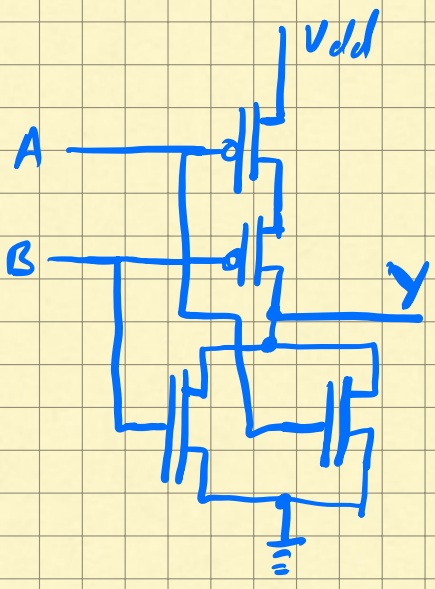
NOR Gate

$Y = \overline{A + B}$



A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

CMOS NOR Gate



Flip-Flops

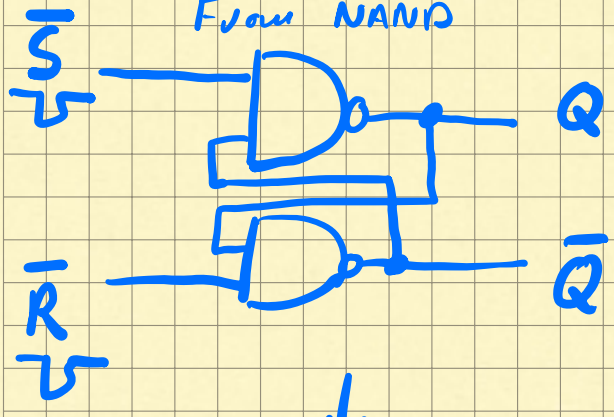
"Memory" devices

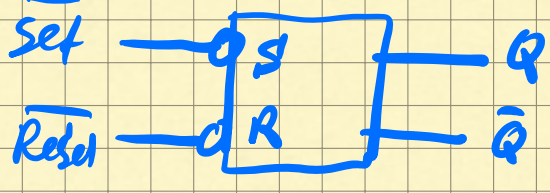
"S" is "Set" — Q = output, 1 when "set", "0" when reset.

"R" is "Reset" — $\bar{Q}$ = inverse output, 0 when "set"

R-S Flip Flop = Reset-Set Flip Flop

From NAND





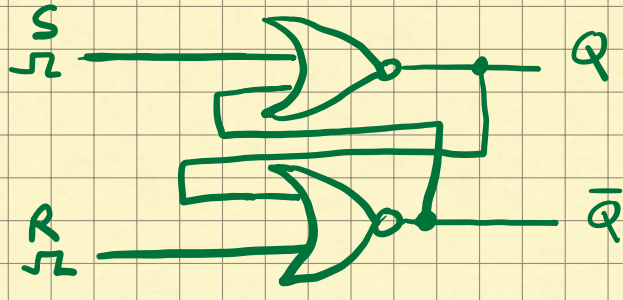
This kind, "hold" is for $\bar{S}$ and $\bar{R}$ at 1

0 on $\bar{S}$ sets Q → 1

0 on $\bar{R}$ sets Q → 0

(Both at 0 is undefined i.e. "Bad")

From NOR



This kind, "hold" is for both S and R at 0

1 on S sets Q → 1

1 on R sets Q → 0

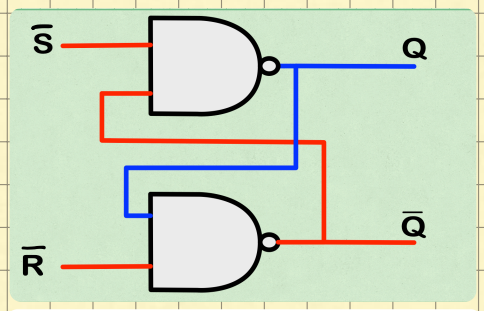
(Both at 1 is undefined i.e. "Bad")

Flip-Flops

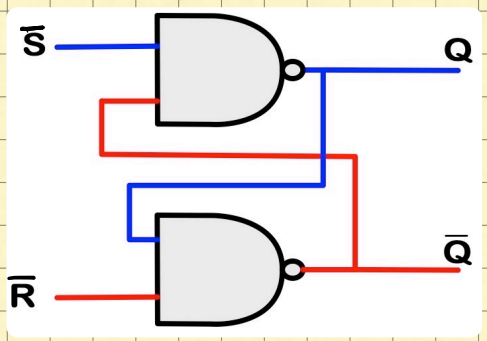
"Memory devices"

Reset-Set flip-flop.

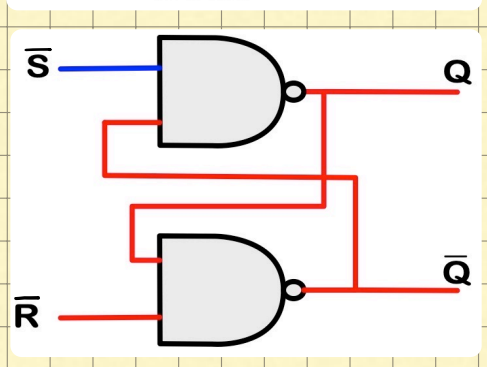
For this initial "hold" state of the flip-flop, both inputs are at 1 and either Q or Q-bar is at 1.



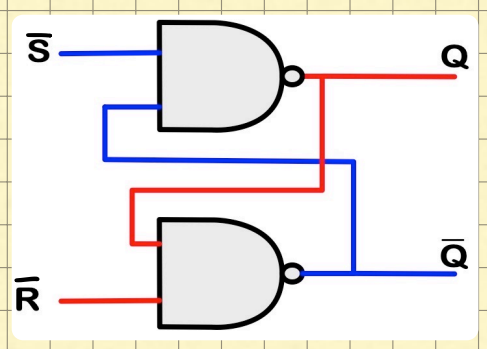
If Q was at 0, then a "set" would set it to 1. The input "S" is thus brought to 0. The output of the top NAND does not change instantly. Propagation time is about 1 ns.



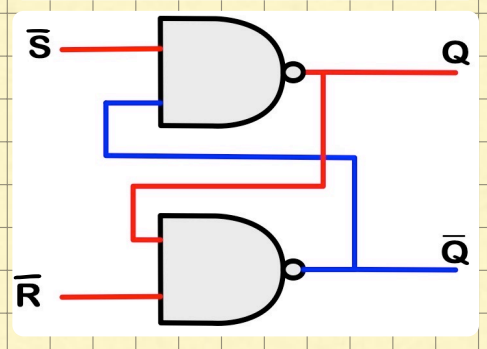
The top NAND has 0 and 1 for input, so after ~ 1 ns the output flips to 1. The bottom NAND now sees 1 and 1 on the inputs.



After again a propagation time of about 1 ns the bottom NAND changes the output to 0.

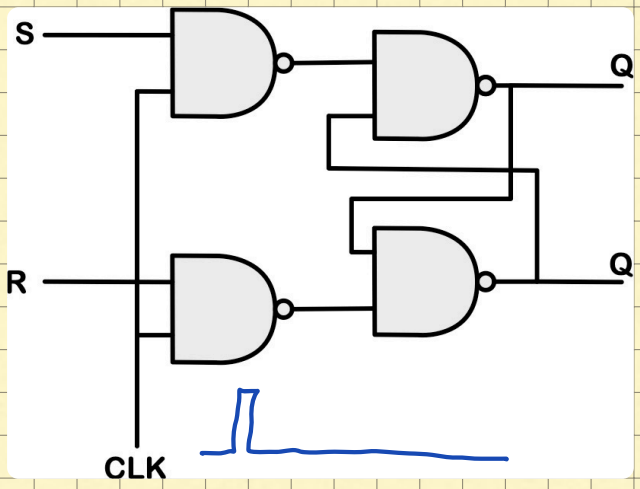


The input at "S" can now go back to 1, and the flip-flop will retain the new state.
To reset the flip-flop, apply a 0 on "R" and follow this same story (but upside down).



R-S Flip-Flop with Clock.

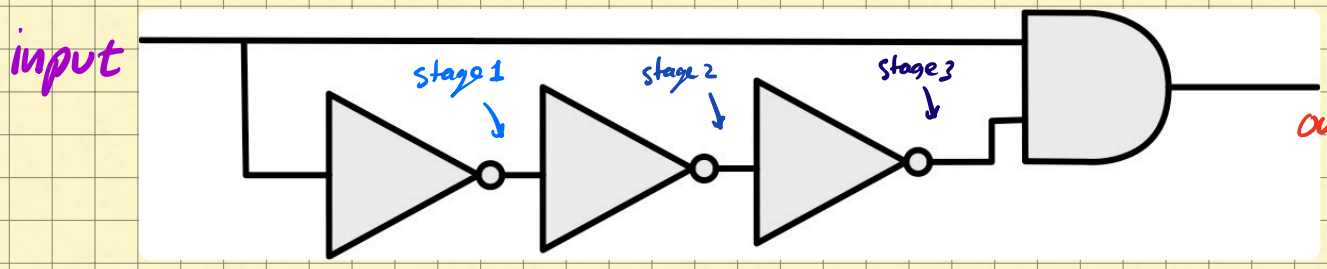
To have the R-S Flip-Flop change state (output) only at well defined moments, we add a clock input, which guarantees the transition can only happen when the clock is 1.

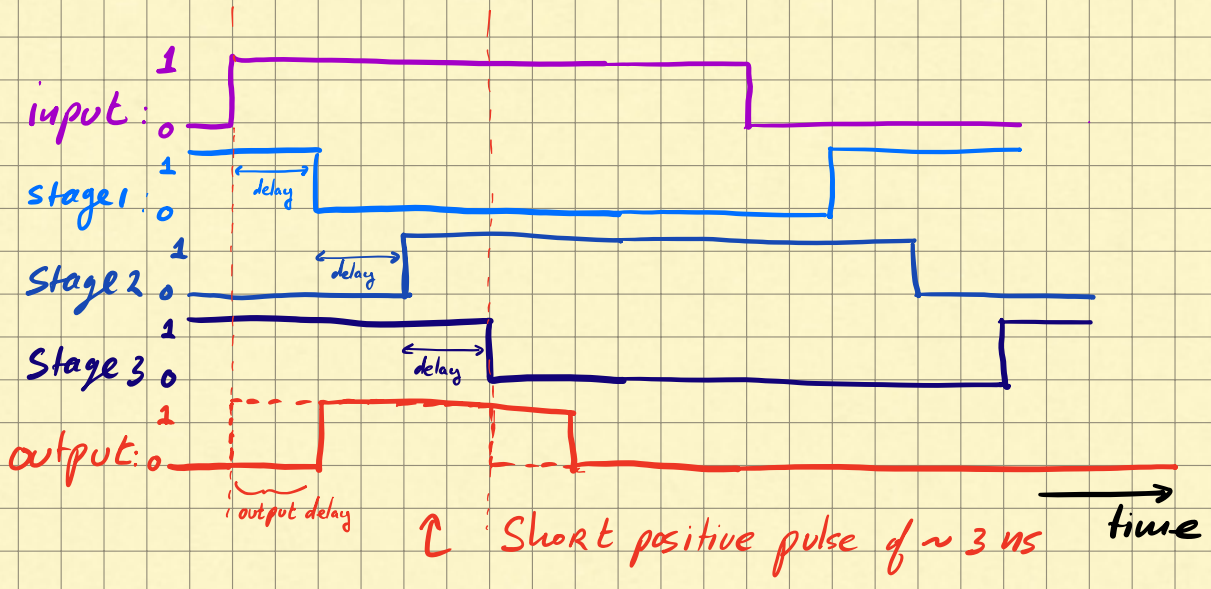


↑ Clock input. To have well defined transition time this clock must be narrow.

A clock signal is usually a square wave. To ensure that the transition happens during a really well defined time, we want a circuit that sets the CLK input of our flip-flop to 1 for only a well determined, short, amount of time, only during the 0 to 1 transition.
Such a circuit can be created with an AND gate and an inverting delay. The delay can be created by 3 inverters back to back. Each inverter has a delayed response of about 1 ns, so 3 inverters delay the signal by 3 ns.
The signal time-diagram is shown below the circuit.

EDGE-Clock Circuit:

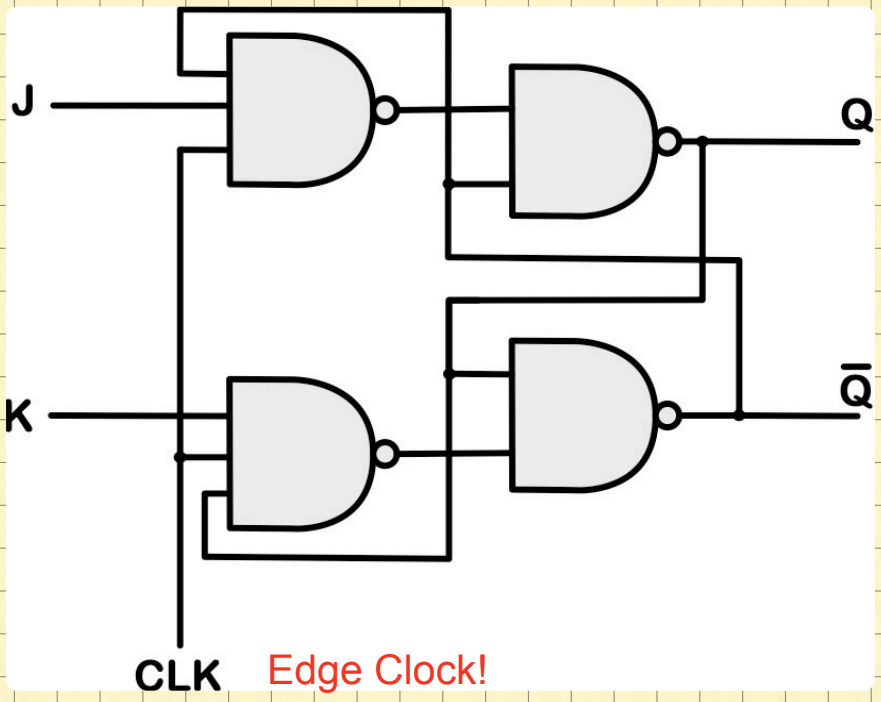




J-K Flip-Flop

Toggle, or divider circuit.

With the R-S Flip-Flop with clock, it is not permitted to have both inputs at 1 at the same time (undefined behavior), and the clock does not help with this. If, however, we use 3 input NANDS for the clock, and feed the outputs back to the 3rd input (one more level of feedback), then the new configuration does allow for both inputs 1. Now, when both inputs are 1 the circuit will toggle the output at each clock.

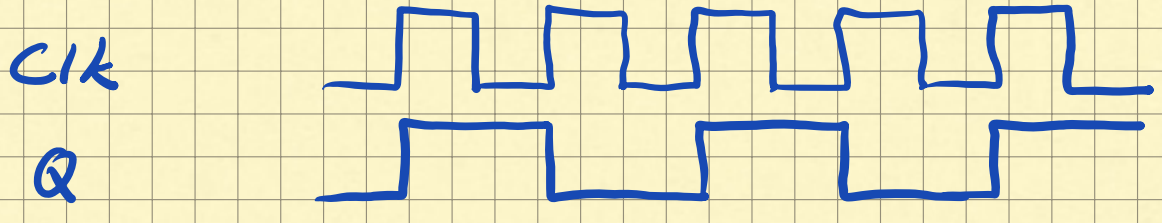


If both J = 1 and K = 1, then $Q \rightarrow \bar{Q}$

So:
- J = 0, K = 0 = "hold"
- J = 1, K = 0 = Set $Q \rightarrow 1$
- J = 0, K = 1 = Reset $Q \rightarrow 0$
- J = 1, K = 1 = "toggle" $Q \rightarrow \bar{Q}$

} On clk = 1

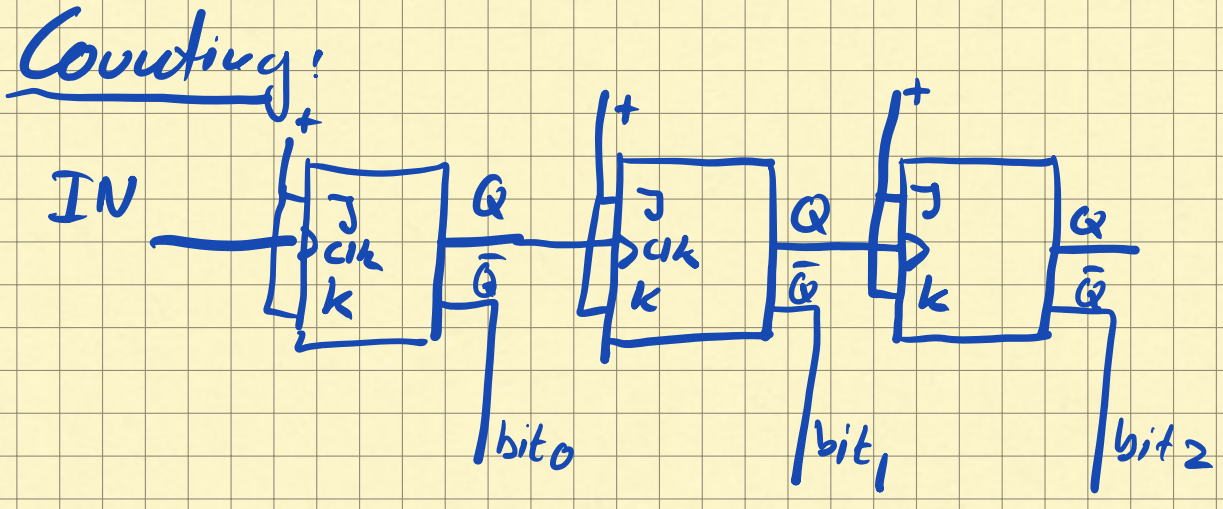
↳

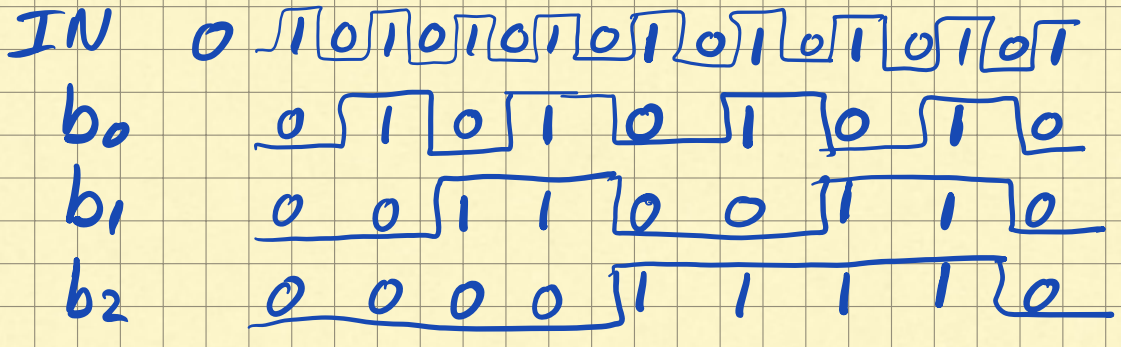


↳ So Q is ½ the frequency of clk.

Counter Circuit

Counting:

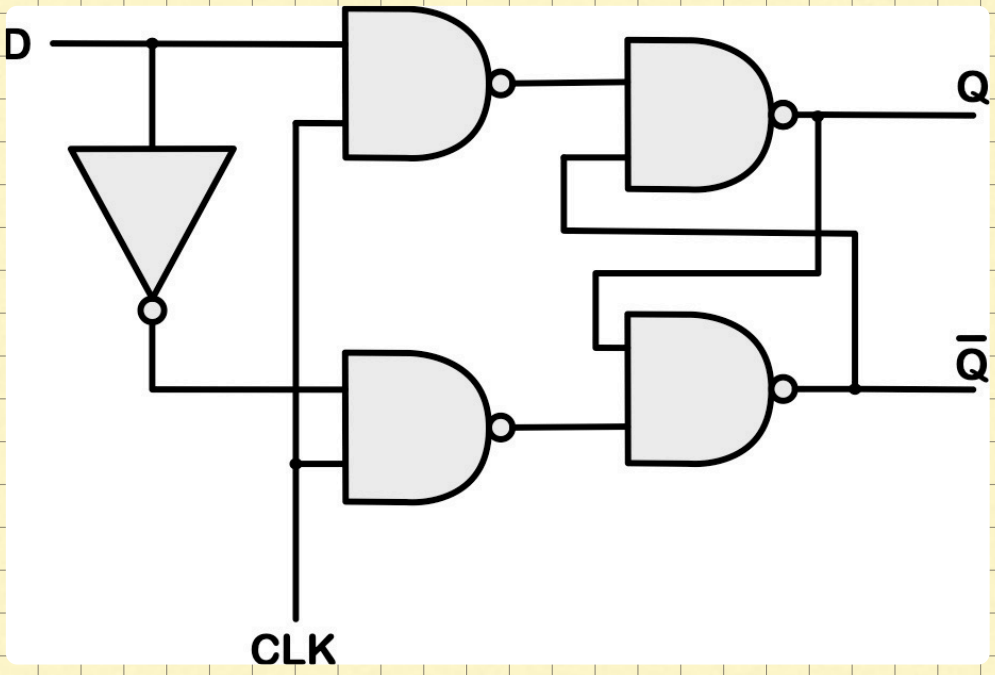




D-Flip Flop

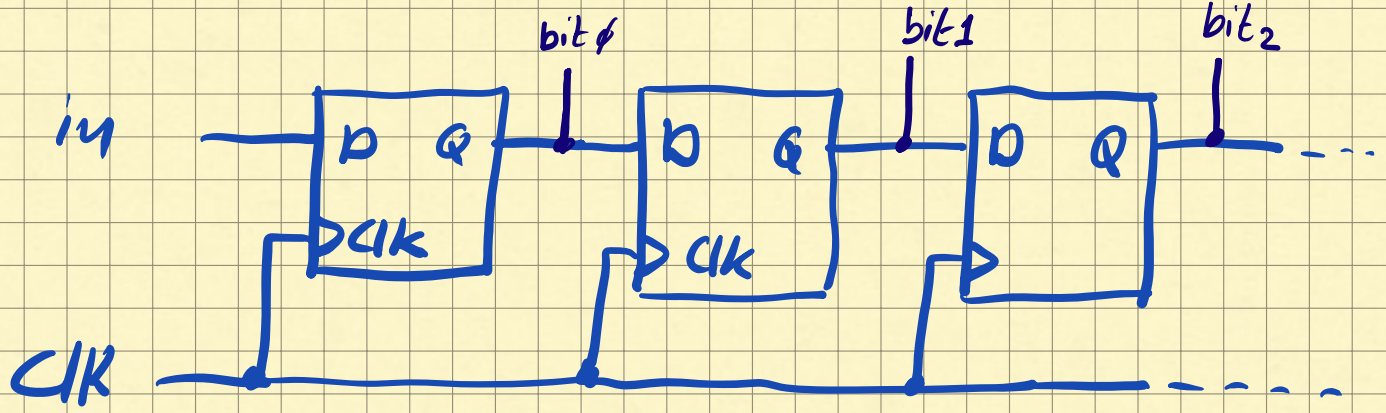
Store a data "bit"

Another way of solving the issue with double 1 for input to the R-S flip-flop, is to eliminate one of the inputs. Instead, we invert the one input to the S and connect it the R input.
We now have a D-flip-flop, which will store what ever the value is on the D input during a clock high, until the next clock high.
It should be clear that this type of flip-flop will be combined with the



On clk → 1
the bit on
D is stored
until the next
clk → 1

Shifter Circuit



Shift register (3 stages shown)

On each clock a bit is shifted in on the left, and the bit is passed along the chain

NOTICE: This is what your book says, but is it true?

NO, if 1 on in, then on the clock all D & Q go to 1 instantly. So why does everyone simplify it this way!

To make a shift register you need more complicated D-Flip Flops. Look up the actual circuit of the SN74HC74 D flipflop. There is lots of extra circuitry to make sure the flip flop only changes state on the 5 rising edge, but the DATA must be set before the rising edge occurs. So, if you chain these, you get the desired effect.