Horowitz & Hill Chap. 8 Digital Electronics

"States"

HIGH ↔ TRUE ↔ 1

LOW ↔ FALSE ↔ 0

A state can be indicated by a symbol, e.g. Q

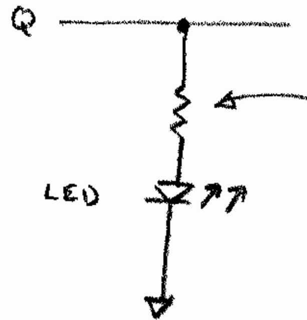
negation:

Q	$\bar{Q}$
0	1
1	0

sometimes written as Q'

this diagram is called a "truth table"

LEDs are often used to indicate state



always use a current-limiting resistor

typ. 150 - 330 Ω

Families of digital logic

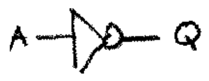
	TTL (74LSxx)	CMOS (74Cxx)	High-Speed CMOS (74HCxx)
Speed	Fastest	Slowest	Almost as fast as TTL
Power Consumed	Most	Least (Best for battery-powered circuits)	
True Voltage	$T \approx +3.4\text{ V}$	$T \approx +4.9\text{ V}$	
False Voltage	$F \approx +0.2\text{ V}$	$F \approx +0.1\text{ V}$	
Power Supply	$V_{cc} = +5\text{ V}$	$+5V < V_{DD} < +15$ (9V battery is okay)	$2V < V_{DD} < 6V$

For all 3 families, the most common power-supply voltage is +5V & GND

Logic Gates

Name | Symbol | Boolean | Truth Table

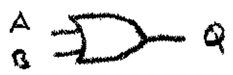
Inverter



$Q = \bar{A}$

A	Q
0	1
1	0

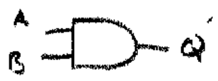
OR



$Q = A + B$

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

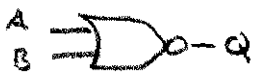
AND



$Q = A \cdot B$

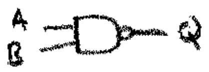
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

NOR

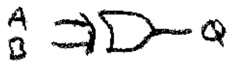


$Q = \overline{A + B}$

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

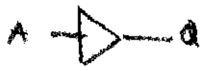
NAND  $Q = \overline{A \cdot B}$

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

XOR  $Q = A \oplus B$

↑ "exclusive or"

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Buffer  $Q = A$

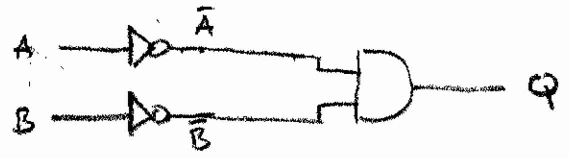
A	Q
0	0
1	1

Boolean Math

you can work problems by making a big truth table:

inputs → intermediate steps → output

ex $Q = \bar{A} \cdot \bar{B}$



inputs				output
A	B	$\bar{A}$	$\bar{B}$	Q
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

↑ result: same as NOR ↑

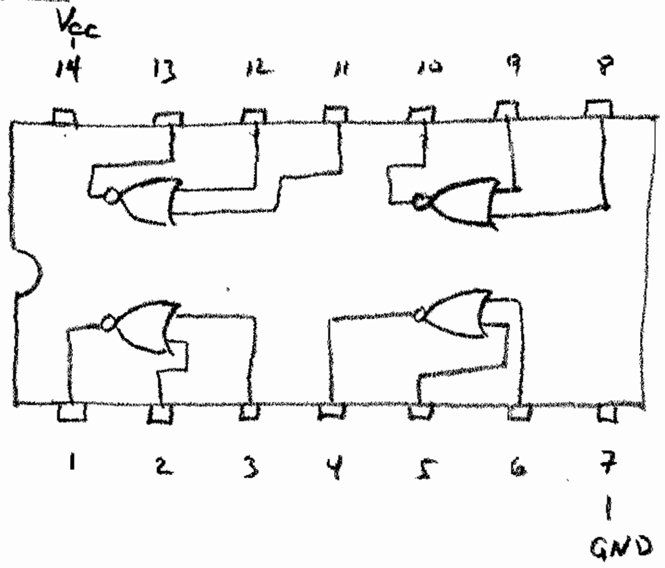
$$\Rightarrow \bar{A} \cdot \bar{B} = \overline{A + B}$$

Digital Chips

Part no.

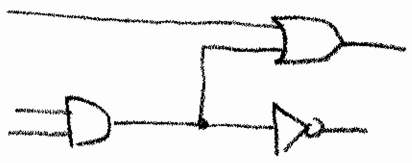
$$\underbrace{74}_{\text{it's a digital chip}}\;\underbrace{LS}_{\text{TTL}}\;\underbrace{02}_{\text{"Quadruple 2-input NOR gate"}}$$

Pin Diagram



Design reminder

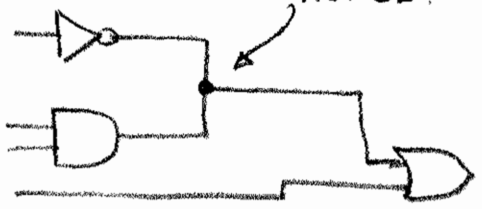
It is OK to connect one output to multiple inputs



OK: two inputs connected to one output

It is NOT ok to connect multiple outputs together

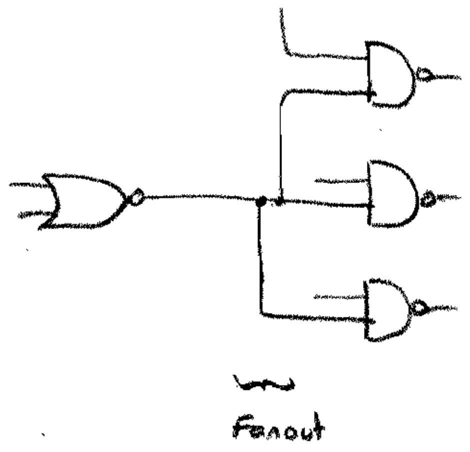
NOT OK: two outputs are connected



"Fanout"

When one circuit's output drives multiple inputs, that's called "fanout"

ex. "Fanout of 3"



A digital gate can typically source/sink enough current for a fanout of 10 or more

Driving LEDs

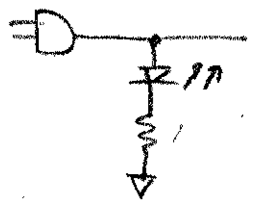
many student projects use LEDs

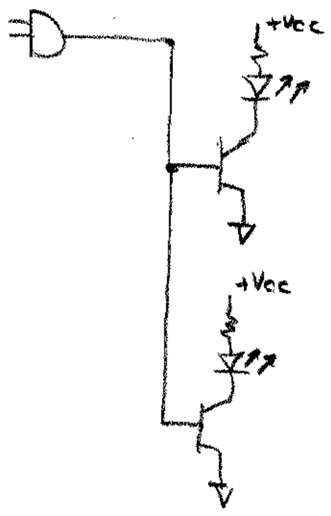
it's easy to use too many LEDs

because an LED is a big power consumer

a gate can drive one LED

if you need to drive more, use a transistor switch

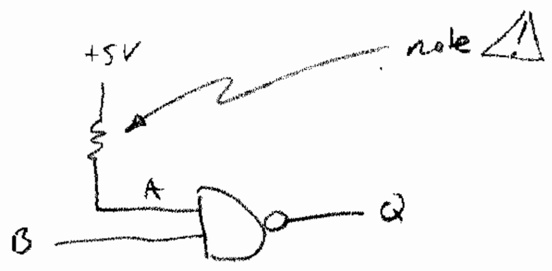




How to use another gate as an inverter

recall NAND truth table

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



always use a resistor when connecting a digital input directly to +5V

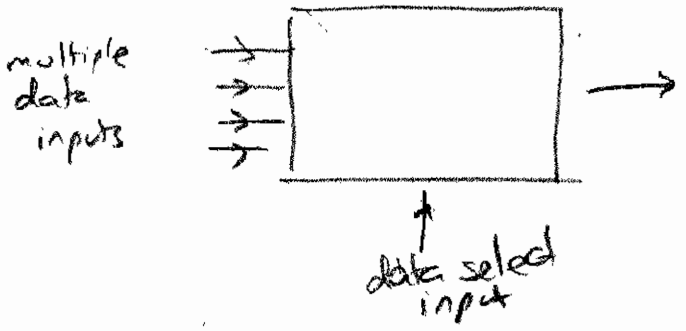
input	output
B	Q
0	1
1	0

↖ inverter

---

multiplexer concept

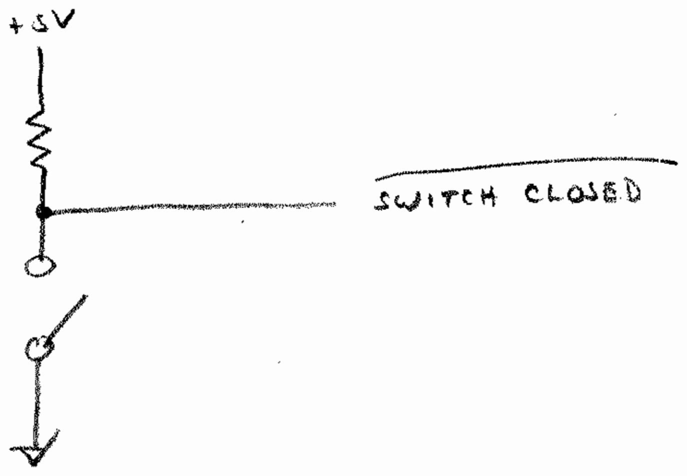
you'll use in Lab 8 - it's like a switch to choose an input



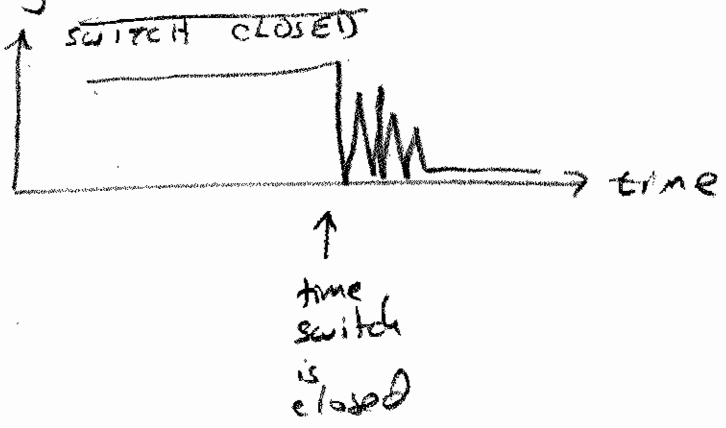
one output, the same as the input that is specified by the "data select input"

Mechanical Switches

a user interface to enter data



the only trouble with this is "bounce"



cure for bounce later & LAB 9.

Binary - Coded Decimal

LAB 9

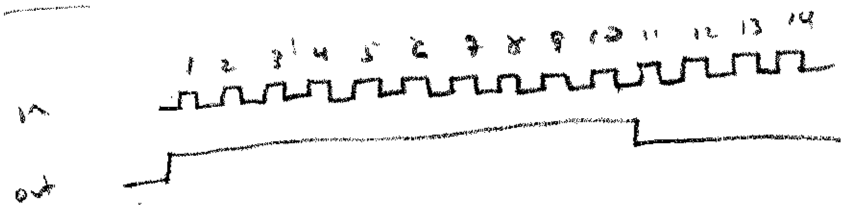
Four binary bits:

- called A, B, C, D
- represent numbers from 0 to 9

* count	D	C	D	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Divide-by-10 Counter (LAB 9)

One use: reduce frequency by 10



Another use:

use 4 outputs D C B A

which advance (as in * above) at each clock input

when it gets to 9, it starts again at 0,

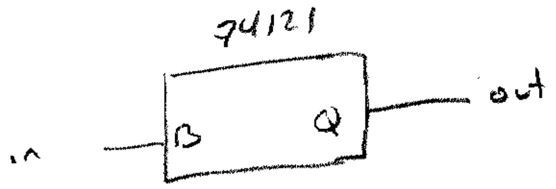
(like a digit in car odometer)

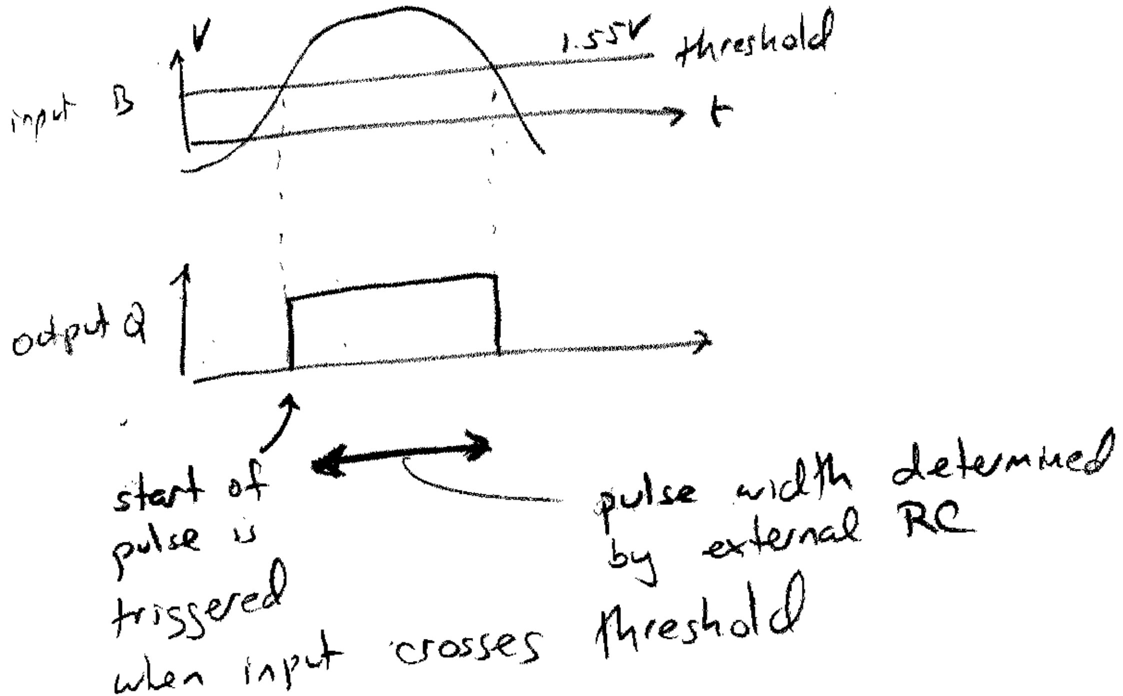
<u>One-Shot</u> LAB 10

↑ aka "monostable multivibrator"

a device widely used to:

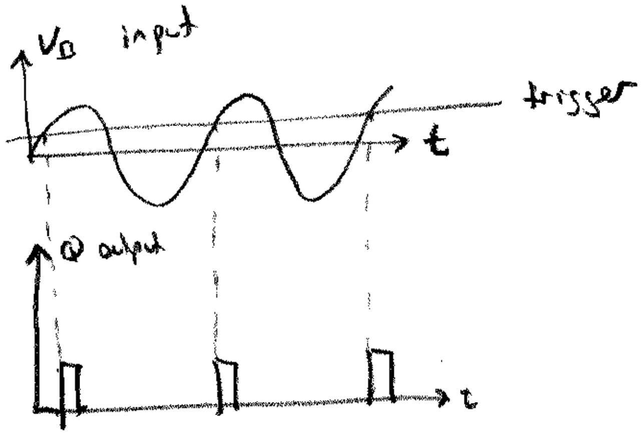
- produce a trigger pulse
- produce a pulse of a desired duration
- analog-to-digital conversion




V
1.55V threshold
input B
t
output Q
start of pulse is triggered when input crosses threshold
pulse width determined by external RC

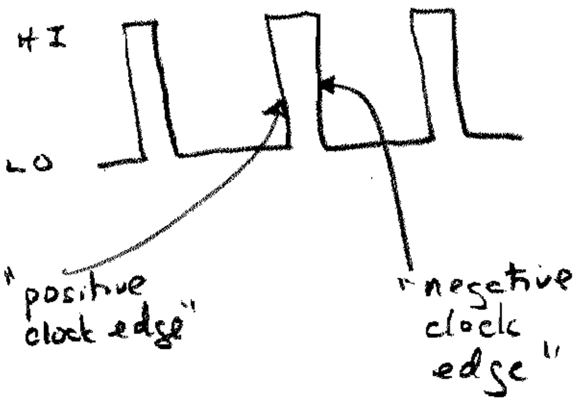
one-shot (cont.)

In LAB 10, you will convert an analog input (sine wave) to a digital output (pulse train) using a one-shot



Digital Clocks (CLK)

- a pulse train
- used by digital circuits that involve:
  - timing
  - transferring data

terminology: 

Flip Flops

Flip flops are digital devices that:

- have memory
- have two stable states

which state the flip-flop has depends on previous history — this is related to the concept of "memory"

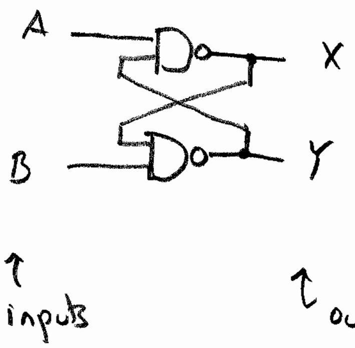
Types of flip-flops:

- SR = set reset, the simplest flip-flop
- Clocked Flip Flops:
  - D = data flip flop, these are often combined to make a "shift register"
  - JK has two data inputs; otherwise much like a D flip flop

Here, we will focus on how an S-R flip-flop functions. In the lab you will also use D and JK flip flops.

SR (Set-Reset) Flip Flop

Has Two data inputs, no clock input



recall NAND truth table:

inputs		output
0	0	1
0	1	1
1	0	1
1	1	0

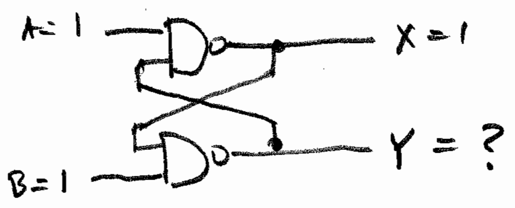
don't erase from blackboard

How it is used:

- inputs A & B are usually HIGH (and the flip-flop outputs will be in one of two possible stable states)
- one input is momentarily made LOW ⊔ to SET or RESET the outputs.

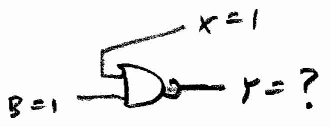
first stable state

suppose inputs A & B are HIGH & output X is HIGH
what is output Y ?



examine the NAND truth table (previous page)

for the lower NAND gate



both inputs are HIGH, i.e. 1

⇒ output must be LOW

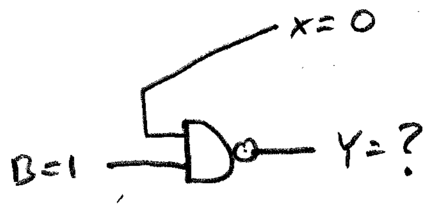
⇒ Y = 0

so, if A & B are HIGH and X is HIGH
then Y must be LOW

second stable state

suppose inputs A & B are HIGH & output X is LOW
what is output Y ?

again, examine NAND truth table
for the lower NAND gate



NAND truth table ⇒ Y=1

so, if A & B are HIGH & output X is LOW

then Y must be HIGH

summary of the two stable states:

- when both inputs A, B are HIGH, there are two stable states of the output X, Y

  first: X=1, Y=0
  second: X=0, Y=1

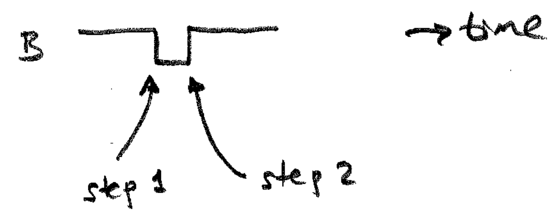
- Q: which of these two states will it be in?

  A, For flip-flops in general, this will depend on the history of previous inputs applied to the flip flop. For the SR flip flop, it depends on whether an input has been "RESET"

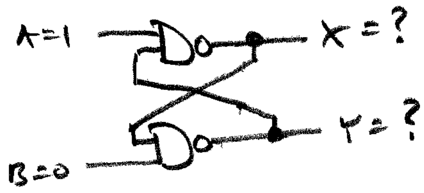
reset for SR flip-flops

Suppose the SR flip flop is initially in the stable state: inputs A = B = 1
outputs X = 1, Y = 0

Next, let's bring B to LOW momentarily

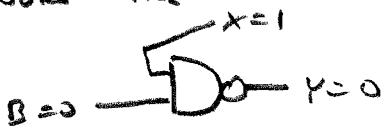


After step 1, the inputs look like this:



To find outputs X & Y, let's start by asking "can X be HIGH?"

If X is HIGH, then the lower gate looks like



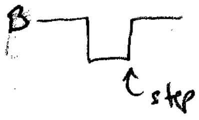
which is not allowed (see NAND truth table)

⇒ X cannot be HIGH
⇒ X must be LOW

Thus, after step 1, the inputs are $A=1, B=0$
outputs are $X=0, Y=1$ *

note: output state is now opposite of what it was before step 1

<u>After step 2</u>



bring B back to high

- The outputs after step 1 (* above) were $X=0, Y=1$
  Now we also have $A=1, B=1$
- This combination $A=1, B=1$ inputs $X=0, Y=1$ outputs is one of the two stable states.
- So the outputs do not change in step 2

Summary for SR flip flop "reset":

Usually you keep both inputs high.
If you momentarily bring input B LOW, then back to HI, that will "reset" the flip-flop to the state $X=0, Y=1$. (If it was already in that state, then nothing changes.)

Timing Diagram (general)

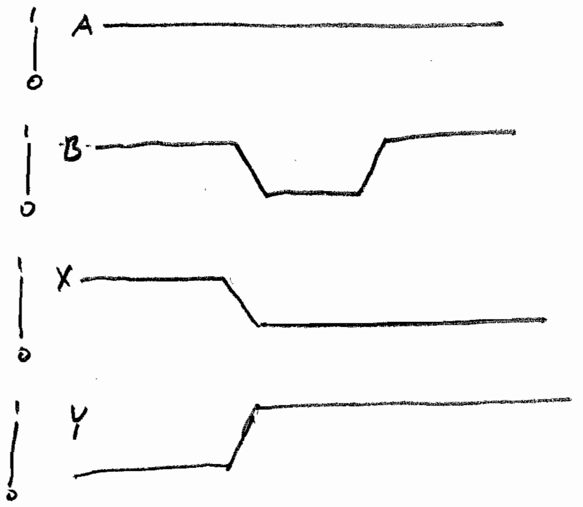
In digital electronics, we need "truth tables" for gates, and for flip-flops (& other devices that involve timing or memory) we also need "timing diagrams"

discuss: [for special project: timing diagrams, unlike schematic diagrams, are not graded, but they might help you in designing your project & debugging it

Timing Diagram
SR flip flop (showing it "resetting" when input B is brought Low).

inputs:

outputs:



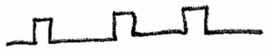
The other possibility for resetting the SR flip flop is to bring input A Low, (instead of input B). This will reset the flipflop to state $X=1$, $Y=0$

Latches
another name for SR flipflop

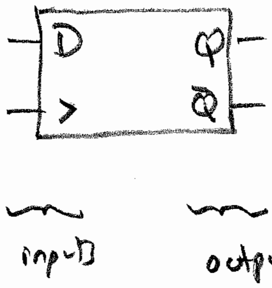
Application of SR flipflops
switch debouncing
(text p. 506 & lab 9)

Data D-flip flop (as in lab 9)

- is a clocked flip flop
  - clock input is a train of pulses

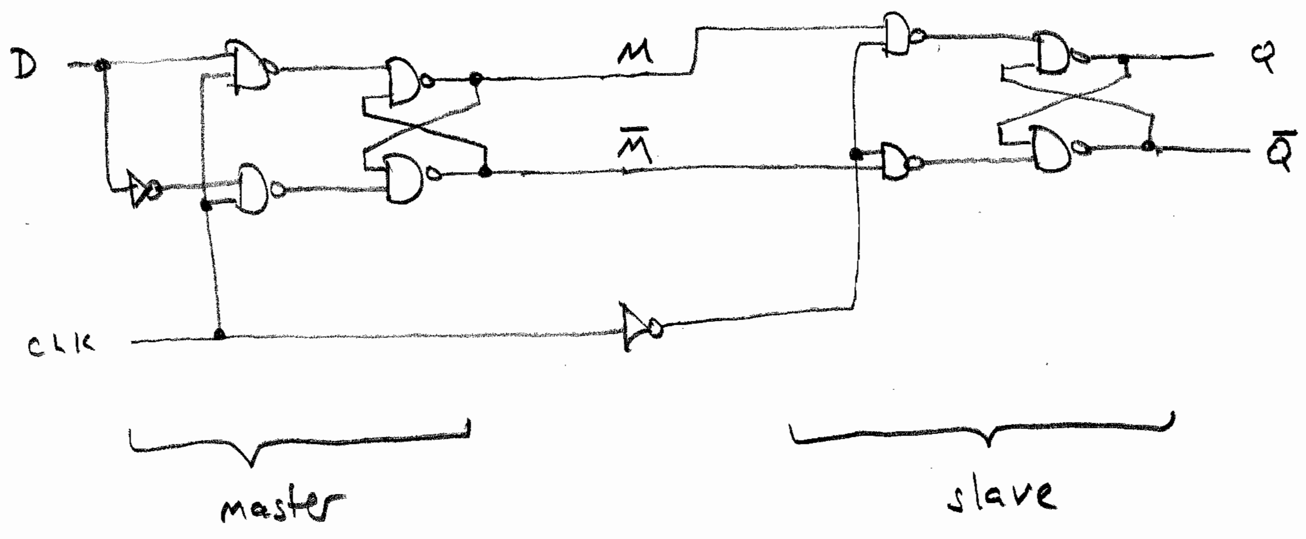


  - indicated on chip with an arrow >

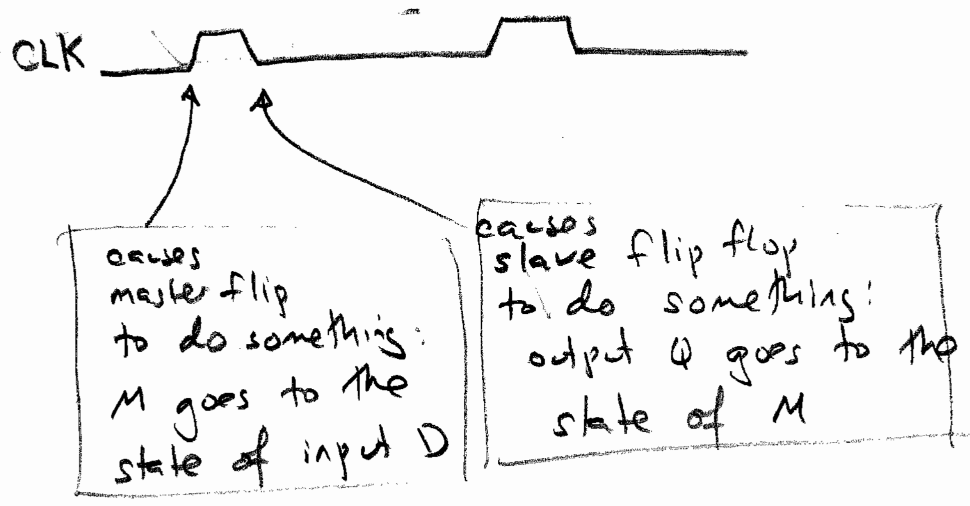


  - also has CLR input to "CLEAR" state of flip flop (CLR = 1 forces Q→1)
  - requires +5V, GND power supply

what's inside a D-flip flop



timings:

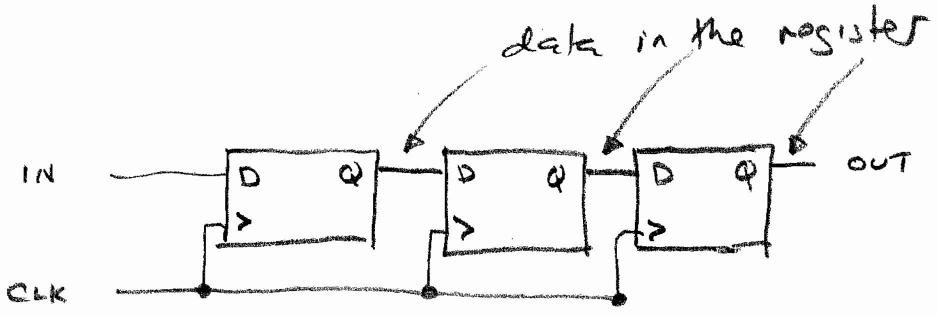

CLK
causes master flip to do something: M goes to the state of input D
causes slave flip flop to do something: output Q goes to the state of M

so after one clk pulse

the data (0 or 1) that was on input D is transferred to output Q

Applications of D flip-flop:

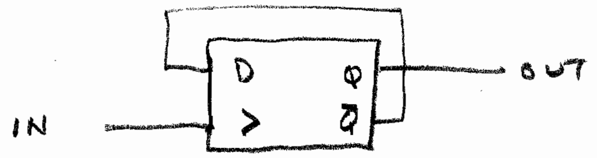
shift Register (Lab 9)

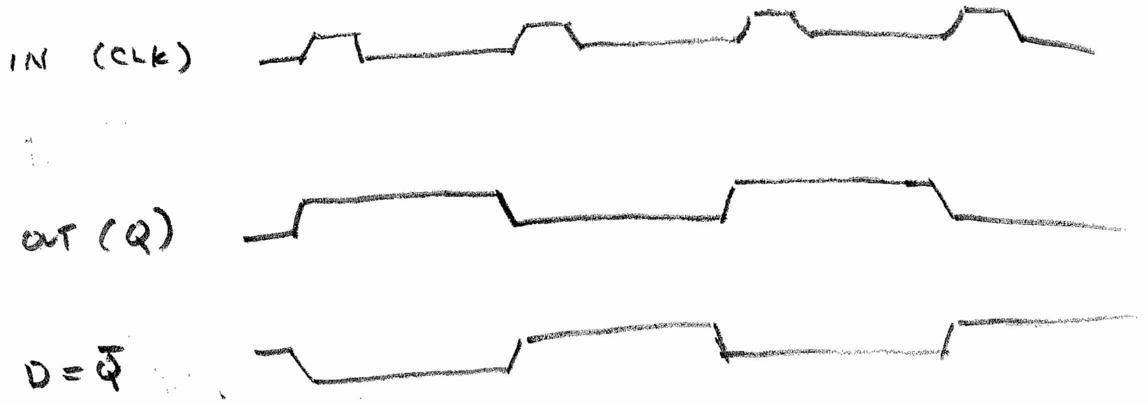


at each CLK pulse
the data in the register (a sequence of 1's & 0's) shifts to the right.

useful for serial memory

Divide-by-2 Counter


IN
D
Q
$\bar{Q}$
OUT


IN (CLK)
OUT (Q)
$D = \bar{Q}$