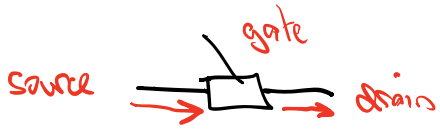
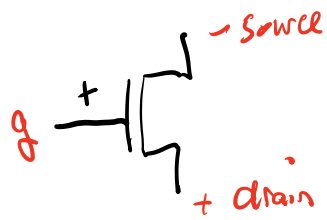
Transistors:



CMOS transistors – complementary metal oxide semiconductor

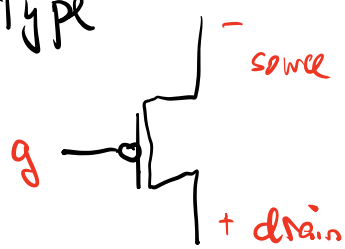
sometimes it conducts, sometimes not.

N-type:



When voltage is applied at the gate, current flows from the source to the drain.

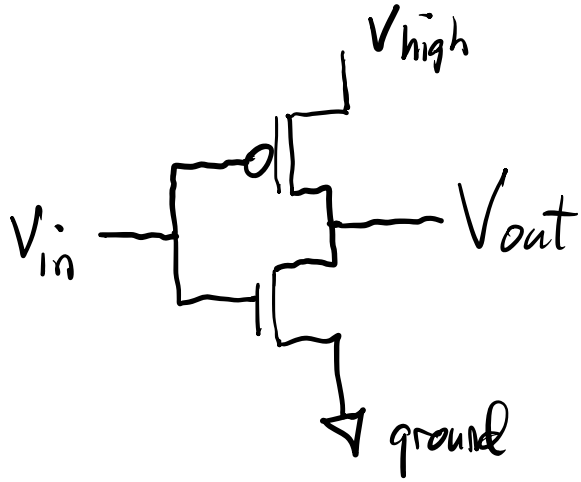
P-type



When there is low voltage at g, current will flow from the source to the drain.

- When voltage is applied at g, no current will flow from source to drain.

Building an <u>inverter</u> from transistors



V_{in} is high: V_{out} will be low.

- high voltage represents 1, low voltage represents 0
  - so $V_{in} = 1$, $V_{out} = 0$

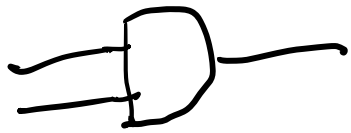
V_{in} is low: V_{out} is high

- $V_{in} = 0$, $V_{out} = 1$

This is the boolean "NOT" operation.

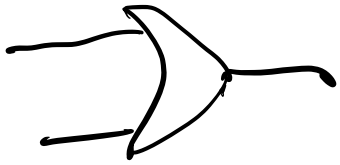
 NOT GATE

Other gates:



AND Gate

- output is 1 when both inputs are 1.



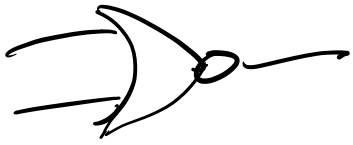
OR GATE

- output is 1 when at least one input is 1.

You can build a circuit for any boolean formula with NOT, AND, and OR gates.

Other commonly used gates:

NAND
(NOT AND)



NOR
(NOT OR)

Building circuits for boolean formulas

"$+$" : OR

"$\cdot$" : and

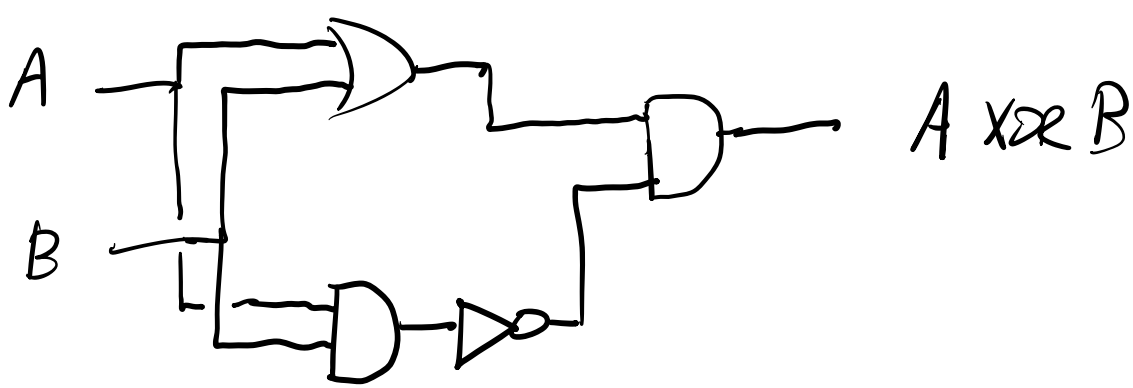
"$\overline{A}$" : not A

notation w/in boolean formulas

XOR :

$$A \text{ XOR } B = (A+B)\cdot\overline{(A\cdot B)}$$

Circuit:



Another way to represent boolean functions is through <u>truth tables</u>

<u>XOR</u>

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Write the inputs in increasing order in binary

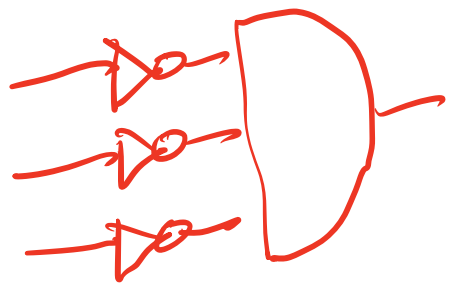
Multi-output truth tables

A	B	C	O1	O2
0	0	0	1	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Algorithm for converting a truth table to a circuit

- Repeat for each output:
  - for each row in which the output is 1, send all inputs to an AND gate, but negate those inputs that are Ø first.

(e.g. first row above:



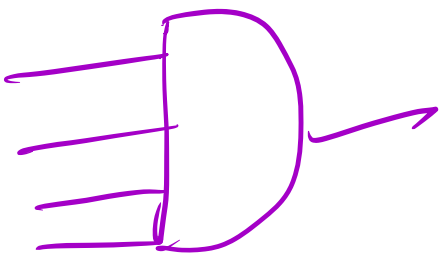
)

- take the outputs of the above AND gates for all the rows where the output is 1 and send them to an OR gate.
- the output of the OR gate is the desired output.

Two things to note:

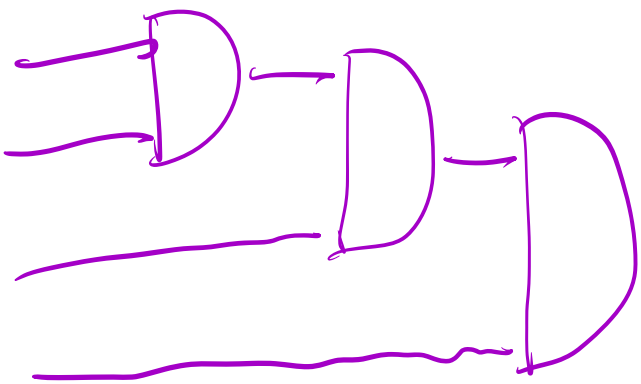
- you can use a circle to represent negation of an input or output of a gate.

- A multi-input AND or OR gate is built from a sequence of AND or OR gates:

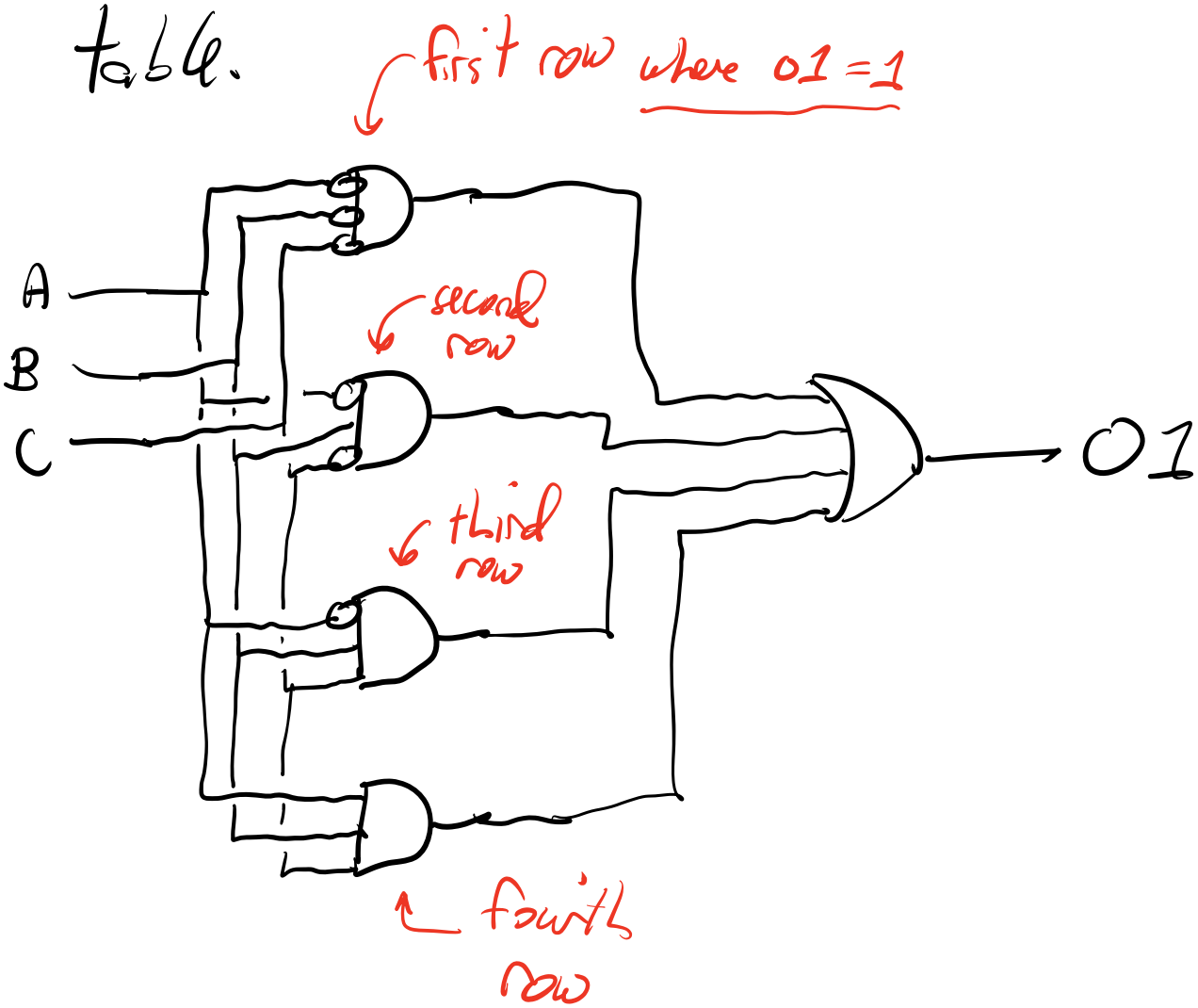


is implemented

as



Implementing the circuit for O1 (output 1) in the above truth table.



Combinational Circuit

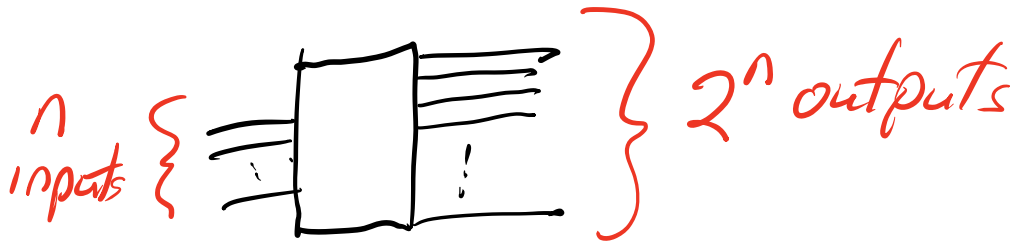
- the output is determined by the current values of the inputs

Sequential Circuit

- outputs may depend on the values of the inputs at some previous time.
  - retains state
  - needed for storing data (registers, memory, cache)

Useful combinational building blocks:

"Decoder"



If the n inputs, when treated as as an n-input binary, carries the value i, then the ith output will be asserted (set to 1) and all other outputs will be 0.

1-bit decoder (1 input, 2 outputs)