Lecture 13

Digital electronics

We are concerned with signals only in two states: yes/no, on/off, high/low, 0/1.

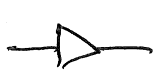
When we are working with a real circuit it is convenient to talk about voltage states. Then we would say: high / low

On paper we would represent it as 1 or 0

Truth tables spell out output state(s) for all possible combinations of input(s)

Some common elements of digital electronics, also known as gates

Symbol — Expression



Buffer

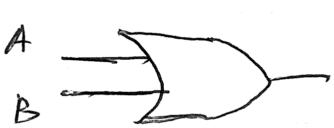
in	out
1	1
0	0

Out = A

inverter

in	out
1	0
0	1

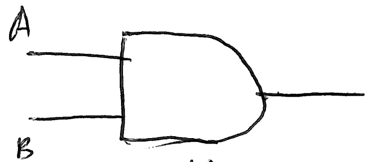
$Out = \overline{A}$



OR

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

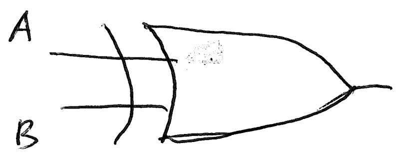
Out = A + B



AND

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

$out = AB$



exclusive OR,
XOR

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

$out = A \oplus B$

P3

Switch on a buzzer when a car engine is working and a driver is present but no earlier that 1st timer (t_1) is triggered, switch the buzzer off after time t_2 but do not do it if special switch overrides t_2 timer.

$$e \cdot d \cdot t_1 \cdot (\overline{t_2} + s)$$

e	d	t_1	t_2	s	b
0	X	X	X	X	0
X	0	X	X	X	0
X	X	0	X	X	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1

note X stands for do not care

Adding two numbers (actually bits)

We would like to do arithmetic sum

A + B in decimal world it is easy

Both A and B can be 0 or 1

$$\underset{\substack{\uparrow \\ \text{arithmetic}}}{A + B} = \left\{ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} \text{ in decimal} \right\} = \left\{ \begin{matrix} 00 \\ 01 \\ 10 \end{matrix} \text{ in binary} \right\}$$

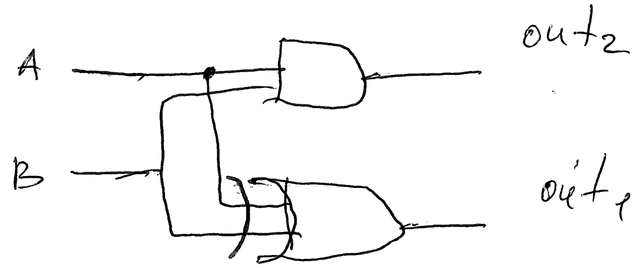
Truth table

A	B	out_2	out_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

↑ carry on bit (under the out_2 column)

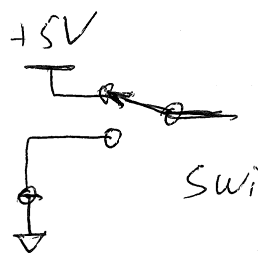
$out_2 = A \cdot B$

$out_1 = A \oplus B$ (↑ logical)



Physical implementation

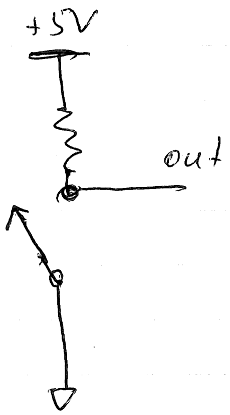
How to get High / low



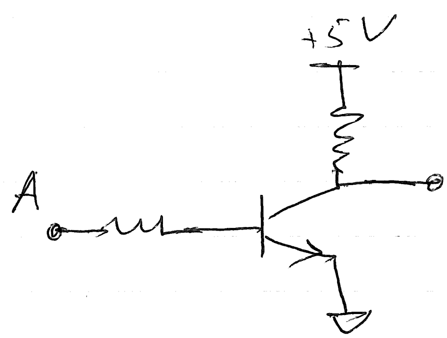
switch

but this requires a double throw switch

How about



someone has to physically close this switch

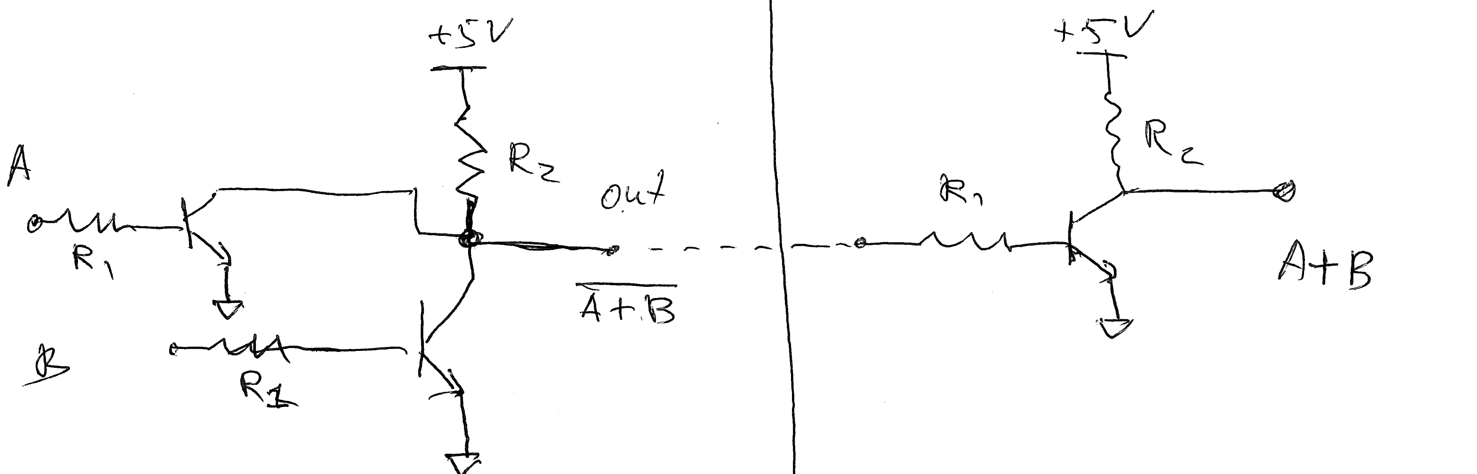


out $= \overline{A}$

this is actually inverter

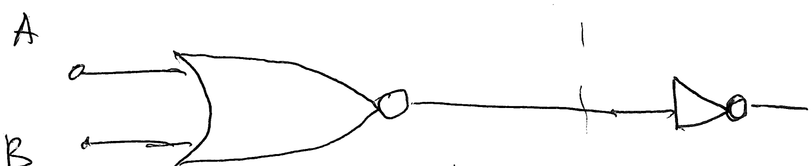
NOR gate $\overline{A+B}$

OR gate $\overline{\overline{A+B}}$
just adding an inverter



transistors must be in saturation
i.e. $\beta \cdot I_B \gg \frac{5V}{R}$

$R_1 = 680\,\Omega$, $R_2 = 470\,\Omega$



NOR

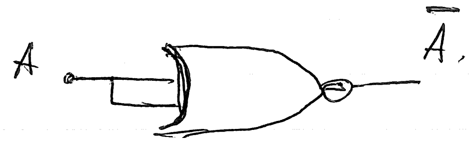
AB	out
00	1
01	0
10	0
11	0

Above is an example of Resistor-Transistor Logic (RTL)

Current implementations are quite different.
See TTL (transistor transistor logic) and CMOS series

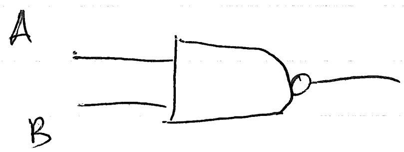
One gate type can be built from others.

For example



inverter out of NOR gate

There is also NAND gate which is very useful too



out $= \overline{(A \cdot B)}$

It can be shown that you can construct any gate (inverter, OR, AND, XOR) from a combination of either NOR or NAND, or obviously mix of NOR and NAND

HW 1

Using only a NOR gate show schematics for OR, AND, and XOR. You can use inverter since we know how to build it with NOR

HW 2 ~~Ba~~ Construct 3 bit adder : A + B + C (arithmetic) with gates of your choice.

Note 1+1+1 = 11 in binary