Lecture/Laboratory #8

- Review Voltage Comparator and **Schmitt Trigger** from Lecture #7.
- Discussion laboratory work

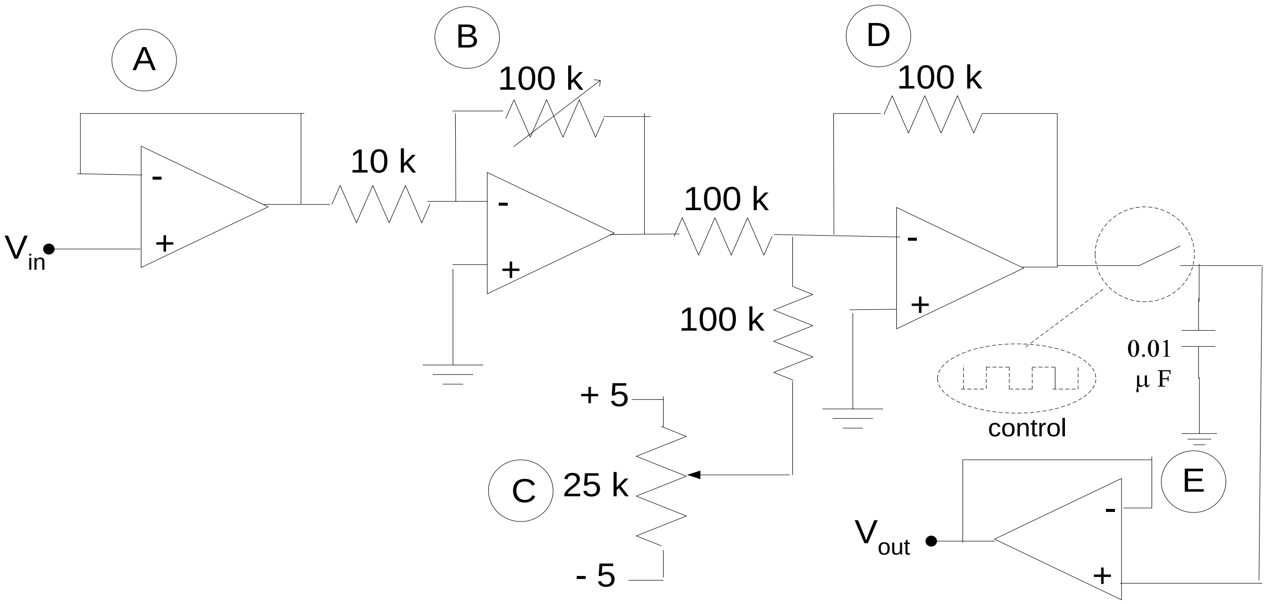
- **"Front End" for Digital Sampling Oscilloscope**
- Introduction to Digital Logic
- Logic Devices
  - Combinational vs. Sequential Logic
  - Gates and Invertors

Front End (for Digital Sampling Oscill.)

Now you have knowledge of sufficient number of electronic components to build "front end" of digital sampling oscilloscope (more precisely → one line of front end; we sill have to develop method of accommodation of several input lines via multiplexing).

Hence, we need "front end" which can take in an input signal, manipulate it and prepare it for the AID converters. Front end has to have the ability to amplify input input signal, provide DC offset, and run it through a sample-and-hold circuit. It should have a high input impedance and relatively low output impedance. Later we will be able to include control logic circuits, which will control logic circuits, which will control signal flow from several input devices.

Front End (cont.)



(A) High Z_{input} Voltage follower:

Z_{in} =1- 2 MΩ , Z_{out} = 75 Ω

(B) Variable gain amplifier: Z_{in} = 10 k, Gain = 0 – 10 x

(C) DC offset adjustment : I_{in} = ± 2 mA (adding voltage follower → better circuit)

(D) Summing amplifier: Output here is DC adjusted

(E) Sample – and – hold: C = 0.01μF was selected in order to provide sampling rate at 40 μsec with precision 1:256

$\Rightarrow$ 40 μsec x 256 = 10^{-2} sec

$\tau = RC = 0.01\ \mu F \times 1\ M\Omega = 10^{-2}$ sec

Introduction to Digital Logic

Basic Logic Concepts

- Digital electronics : circuits that deal with data made of 1's and 0's.
- Especially convenient for input signals, which are naturally discrete (e.g. pulses from a particle detector, "bits" of data from computer).
- So far we have dealt with RC circuits, amplifiers, op-amps, etc, for which input and output signals were varied over a range of values (e.g. continuously varying voltages from measuring instruments)
- Conversion of analog (continuous) signal to digital (discrete) from ⇒ A/D converters
- Importance of digital system ⇒ transmission of information without pick up of the noise.

"Digital electronics" – circuits with only two states at any point:
"High" and "Low" voltages.

In Boolean logic "1" and "0" corresponds to "True" and "False", respectively.

In electronic industry the states are connected with Boolean logic ↔
High → 1 Low → 0

Logic Devices

Combinational vs. sequential (clocked) logic

Digital electronics ⇒ generation of digital outputs from digital inputs (e.g. adding two 16 – bit numbers as input and generation 16 – bit output as a sum.

Output is predetermined function of the input ⇒ combinational task (logic) combinational tasks can be performed with “gates” ⇒ Boolean algebra applied to binary systems.

“Sequential” logic → requires also knowledge of the past inputs, and therefore cannot be done by combinational function of the inputs alone. It requires some type of digital memory and use sequential procedures.

Examples of sequential tasks:

(a) Counting 1’s in a sequence;

(b) Converting a string of bits into a parallel set of bits

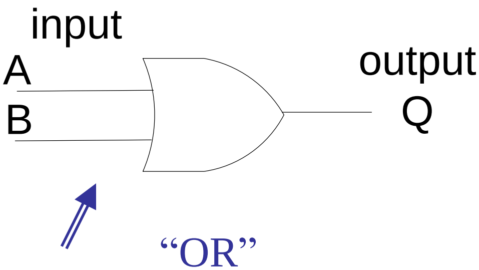
(c) Recognizing a certain pattern in a sequence

(d) giving one output pulse for each four input pulses; etc.

Gates

OR gate:

The output of OR gate is HIGH if either input (or both) is HIGH:



two inputs
in OR gate

“OR”

Truth Table

Inputs		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

The Boolean symbol for OR is “+”

“A OR B” ⇒ A + B

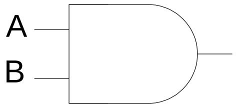
⇓

$Q = A + B$

In general, gates can have any number of inputs, but standard packages usually contain four 2 – input gates, three 3 – input gates, or two 4 - input gates.

Gates (cont.)

AND gate:



$Q = A \bullet B$

(AB)

“AND”

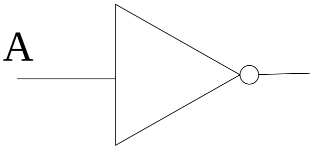
“A AND B” ⇒ $A \bullet B$

Truth Table

Inputs		Output
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

INVERTER (’NOT” function)

Inverter is “gate” with only one input



$Q = \bar{A}$

“INV”

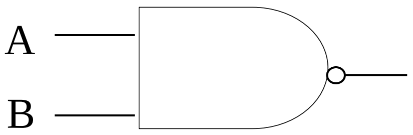
Boolean symbol:

“NOT A” ⇒ $\bar{A}$

A	Q
0	1
1	0

Gates (cont.)

NAND gate:

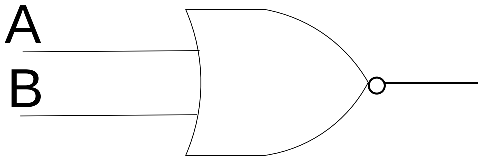


$Q = \overline{A \cdot B}$

"NAND"

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

NOR gate:



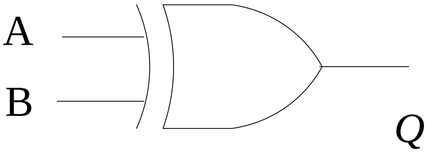
$Q = \overline{A + B}$

"NOR"

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

Gates (cont.)

XOR (Exclusive – OR) gate:



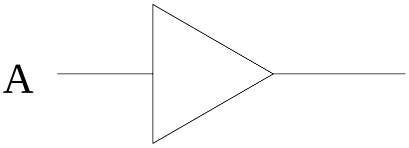
$Q = A\overline{B} + \overline{A}B$

"XOR"

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

Output is HIGH if inputs are different

BUFFER

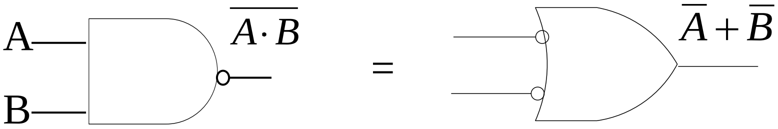


Q = A

A	Q
0	0
1	1

Gate interchangeability

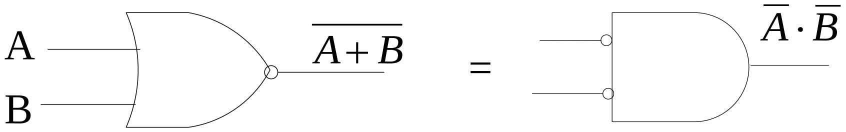
Example 1: NAND ⇒



Gates (cont.)

Gate interchangeability

Example 2: NOR ⇒



Logic Identities

The last two examples illustrated how gates NAND and NOR can be constructed in different ways. These possibilities are result of logic identities:

$$\left.\begin{aligned}\overline{A\cdot B}&=\overline{A}+\overline{B}\\ \overline{A+B}&=\overline{A}\cdot\overline{B}\end{aligned}\right\}$$

Related to DeMorgan’s theorem

→ important for circuit design.

Similarly we can write other logic identities:

Logic Identities (cont.)

$ABC = (AB)C = A(BC)$

$AB = BA$

$AA = A$

$A1 = A$

$A0 = 0$

$A(B+C) = AB + AC$

$A + AB = A$

Proof by Truth Tables:

A	B	C	ABC	AB	C	(AB)C
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	0	0
1	1	1	1	1	1	1

A	A	AA	
0	0	0	A
1	1	1	

A	B	AB	A + AB	
0	0	0	0	A
1	0	0	1	
0	1	0	0	
1	1	1	1	

Logic Identities (cont.)

Some Truth Tables:

$$A + B + C = (A + B) + C = A + (B + C)$$

$$A + BC = (A + B)(A + C)$$

A	B	C	BC	A +BC	A+B	A+C	(A+B)(A+C)
0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1
0	0	1	0	0	0	1	0

$$A + B = B + A$$

$$A + A = A$$

A	A	A + A	
0	0	0	A
0	1	1	

$$A + 1 = 1$$

A	1	A + 1	
0	0	1	1
0	1	1	

Logic Identities (cont.)

$$A + 0 = A$$

$$\overline{1} = 0 \;\; ; \qquad \overline{0} = 1$$

$$A + \overline{A} = 1$$

$$A\overline{A} = 0$$

$$\overline{(\overline{A})} = A$$

$$A + \overline{A}B = A + B \Longrightarrow$$

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

$$\overline{A}B + A\overline{B} = A \oplus B$$

$$A \oplus B = \overline{A}B + \underset{\substack{\| \\ 0}}{A\overline{A}} + A\overline{B} + \underset{\substack{\| \\ 0}}{B\overline{B}} = A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) = (A + B)(\overline{AB})$$