Digital Systems 1 (DS1)

Core introductory course DS1:

02 40 6152 00 - D1, TCS, sem. 3
02 42 6170 00 - D1, CS, sem. 3

Advanced compulsory course DS2:

02 40 6153 00 – D2, TCS, sem. 5

DS1 & DS2 course manager:

dr inż. Piotr Dębiec,

Institute of Electronics
Medical Electronics Division
email: pdebiec@p.lodz.pl
http://eletel.p.lodz.pl/pdebiec/ds1/ → user name: student_eit
password: logika

Office hours 2016/17:

Monday, 16-17, WEEIA Library, 18/22 Stefanowskiego Street
(Please, announce your coming one day in advance, e.g. by email)

DS1 goals …

1. Acquainting students with architecture and principles of operation of fundamental digital electronics elements, and with the theory used in digital circuit design.

2. Presentation of basic methods, techniques, and tools used for design and analysis of combinational and sequential digital systems.

3. Presentation of selected, simple examples of heuristic, creative methods to design non-standard digital circuits.

DS1 learning outcomes …

After crediting the course student is able to:

1. **Analyze** simple combinational and sequential circuits.

2. **Design** simple, non-standard combinational circuit.

3. **Design** simple, synchronous finite state machine on the basis of the state transition diagram of the circuit.

4. **Find** formal **models** of simple digital circuits on the basis the description of its operation.

5. **Implement** and **test** of the target circuits at the laboratory workstation.

6. **Present** the **results** of the experiments in the form of a written report.

More information: **http://programy.p.lodz.pl**

DS1 resources …

1. Mano M. M., Ciletti M., **Digital Design**, 5th edition, Prentice Hall Inc., 2013.

2. Wilkinson B.: **Digital System Design**, Prentice Hall, 2003.

3. Burger P.: **Digital Design. A Practical Course**, John Willey & Sons, Inc., New York, 1988.

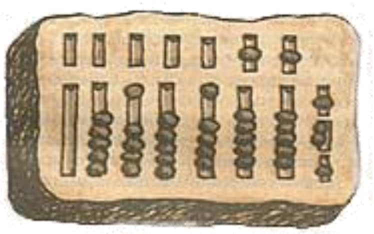
4. Leszczyński Zygmunt : **Teoria układów logicznych**. Politechnika Łódzka, Skrypty dla Szkół Wyższych, Łódź 1990 (in Polish).

5. Lecture and laboratory materials published on the **DS1 web page**.

6. Tyszer J., Mrugalski G., Pogiel A., Czysz D., "**Technika cyfrowa. Zbiór zadań z rozwiązaniami**.", Wydawnictwo BTC, 2010 (in Polish).

…and many other resources , e.g. video lectures in the Internet : MIT, UC Berkeley, YouTube.

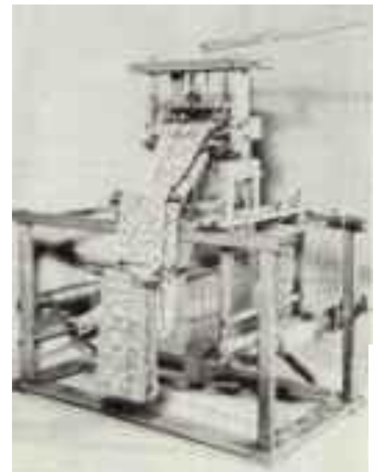
Brief history …



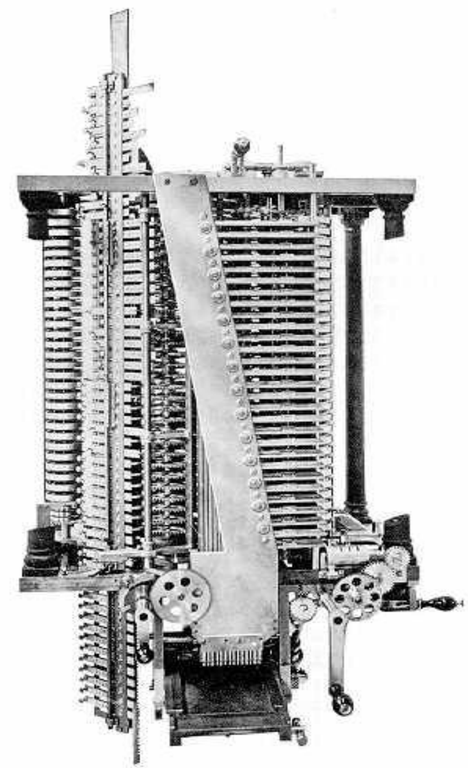
ABACUS – **5th century B.C.** (Rome, Greece, China)
- first calculating tool
- addition, subtraction, multiplication, division, and even square-rooting
- still used by clerks in China, Japan, India
- binary (base 2) – quinary (base 5) decimal calculations

Pascaline – **1642,** first mechanical calculator (addition, subtraction, multiplication, division) – Blaise Pascal
- calculations in decimal system
- crown-type gears connected in series (similarly to odometer in cars)
- metal wheels to dial the operands

Brief history …



Jacquard Loom, Joseph-Marie Jacquard
- **1st half of 19th century**
- the first flexible steam loom
- punched cards for coding pattern of fabric
- about 24 thousand cards used for Jacquard’s portrait



Analytical Engine - **1812** Charles Babbage
- **design** of the first general-purpose **c**omputer
- decimal system
- 50.000 mechanical elements
- powered by steam engine
- punched cards as a memory
- logarithms calculations, solution of differential equations
- 3 main parts: „store”, „mill” and „sequential mechanism”
- precursor of modern computers

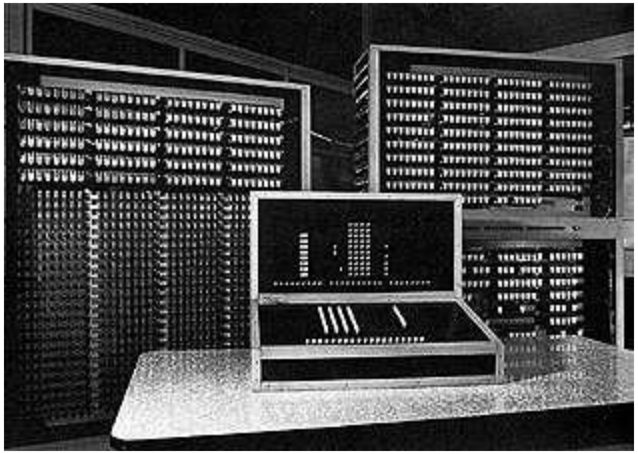
Brief history …

„Laws of Thought”: George Boole (England) – **1854**

- beginnings of Boolean algebra and calculus of logic

„Turing Machine”: Alan Turing (England) – **1936**

- abstract model of a computer
- beginnings of sequential systems theory



Z3 Computer - Konrad Zuse (Germany) – **1941**

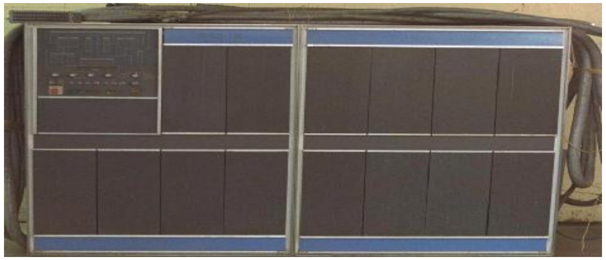
- first electronic computer,
- **binary system** (22-bits), 1 addition/sec.
- 2000 **relays**, 4kW, 1 ton , f_{clk} = 5-10 Hz
- design of aircrafts and missiles

ENIAC – **E**lectrical **N**umerical **I**ntegrator **A**nd **C**alculator (USA) – **1944**

- 160kW, 30 tons, 1800 m^2
- **logic gates AND, OR NOT**
- 18.000 **electron (vacuum) tubes**
- 5000 additions (357 multiplications) per second
- **decimal system**
- **programmed manually**

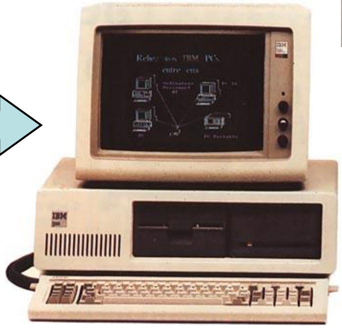
Brief history …

1956 – 1964 - 2nd generation computers, e.g. **IBM1401**
- transistors
- **multiplexers**
- beginning of operating systems



1965 – 1980 - 3rd generation computers
- integrated circuits
- **minicomputers (e.g. PDP-8)**
- multitasking

1980 – now 4th generation computers
- microprocessors (1980 - computer **IBM PC**)
- beginnings of LANs



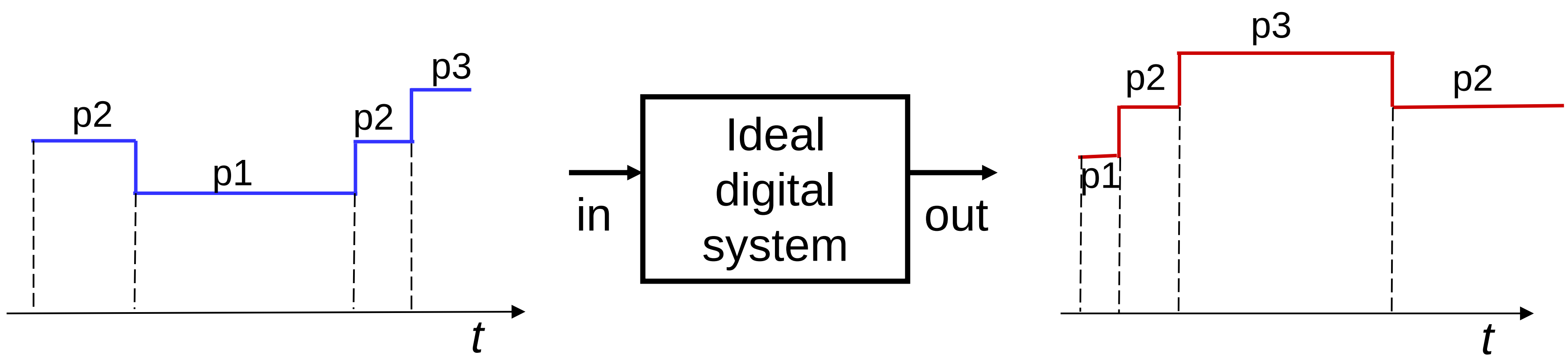
5th generation:
- neural network computing
- artificial inteligence
- fuzzy logic
- nanotechnology – quantum computing

?

Introduction

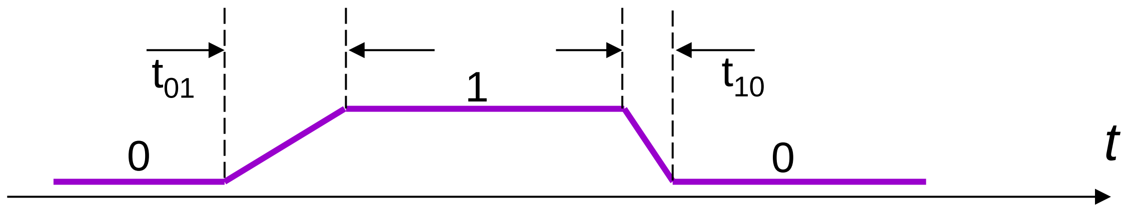
Two main features of ideal digital systems:

- Input and output signals take only discrete values
- Signals change their values at discrete time-points only



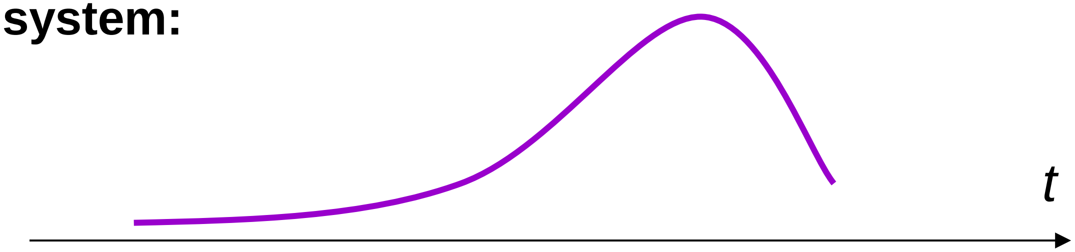
Introduction

Real digital system:



Real digital signal (binary)

Analog system:

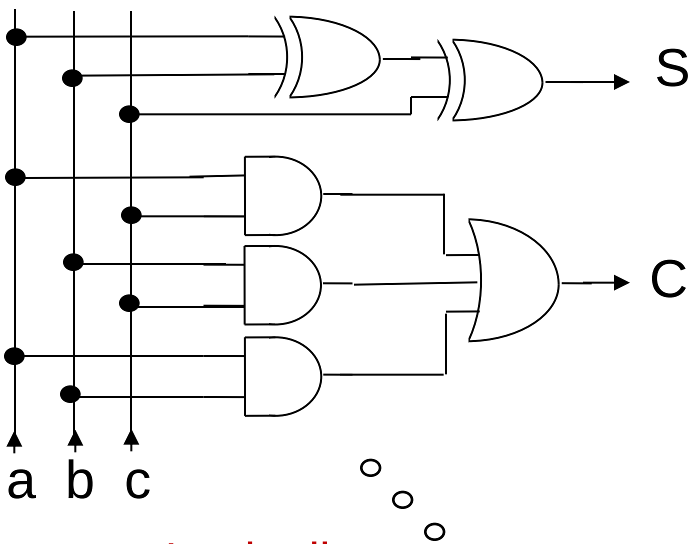


Analog signal

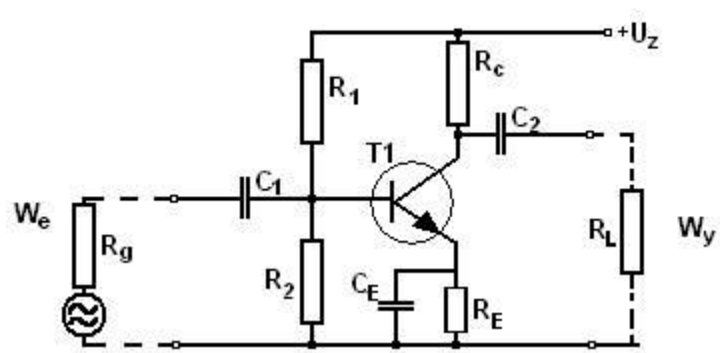
Digital vs analog system

Block diagrams:

Detailed diagrams:



Logic diagram



http://home.agh.edu.pl/~maziarz/LabPE/rc.html

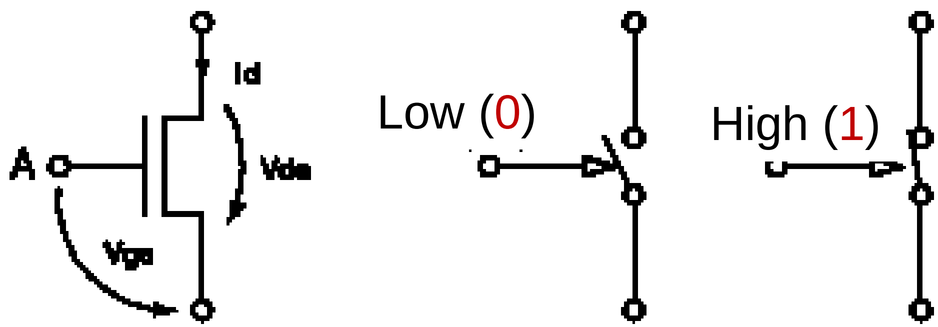
Schematic diagram

Basic digital elements - gates

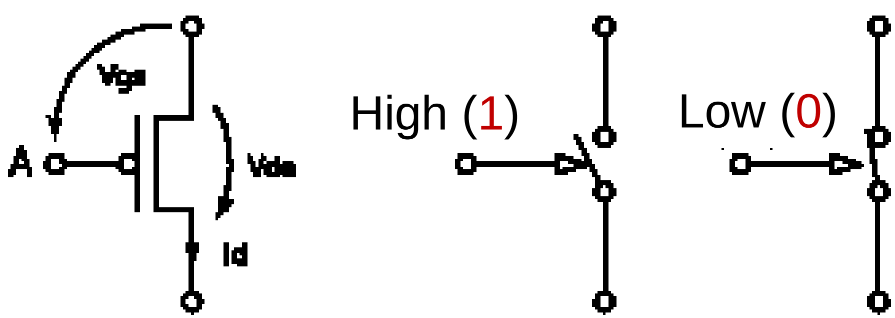
- Manufactured mostly in CMOS technology
- Have one or more inputs and one output
- Built of 1, 2 or 3 pairs of transistors working as switches

N-MOS and P-MOS switches:

N-MOS

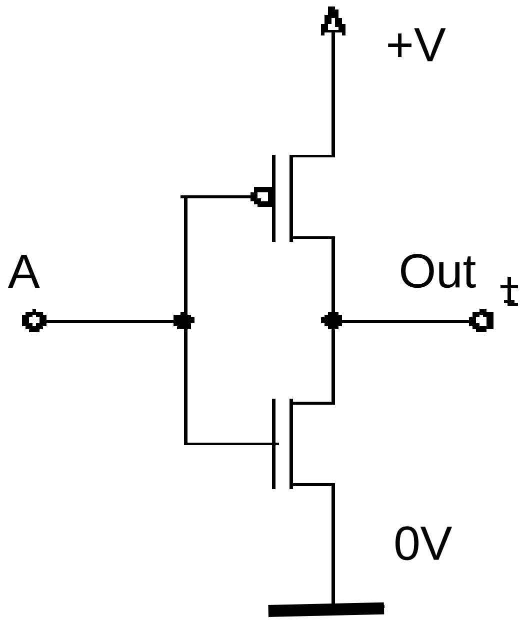


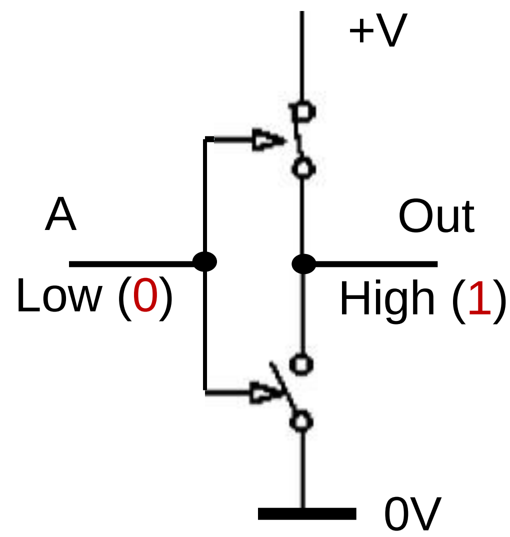
P-MOS

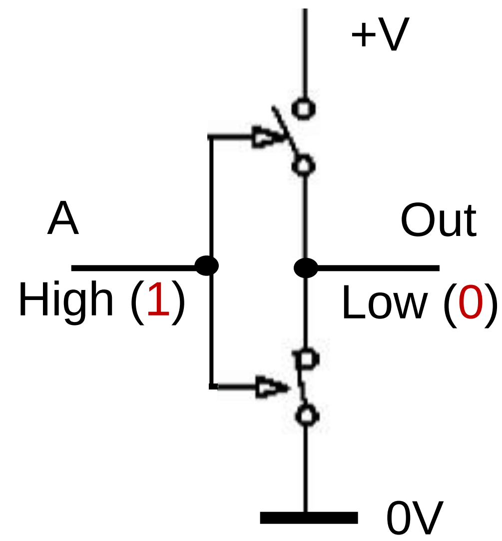


http://www.iue.tuwien.ac.at

NOT gate

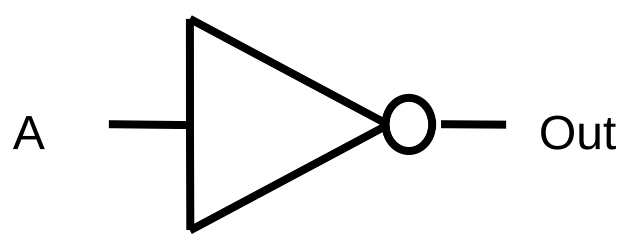






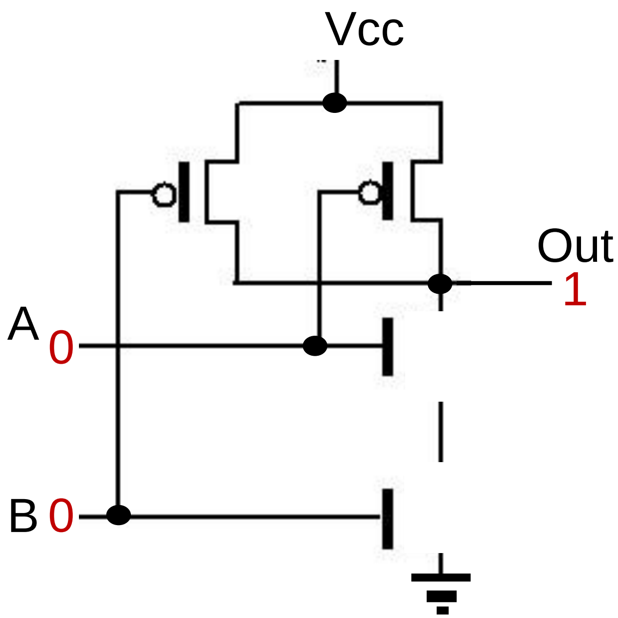
A	Out
Low	High
High	Low

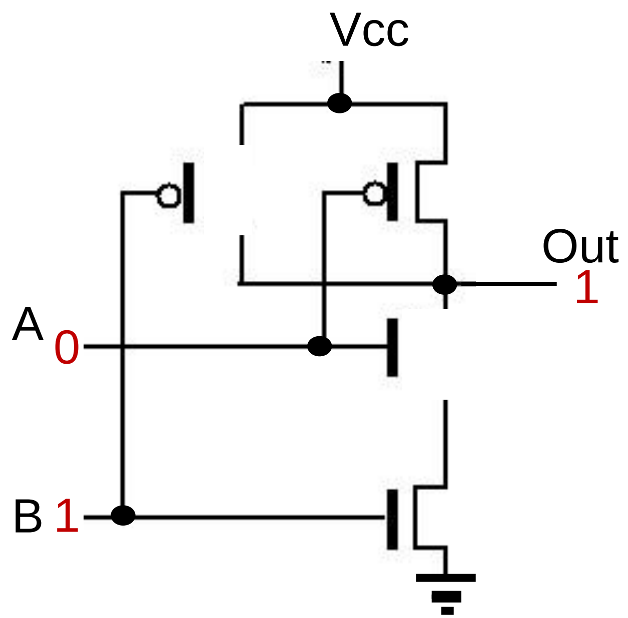
A	Out
0	1
1	0



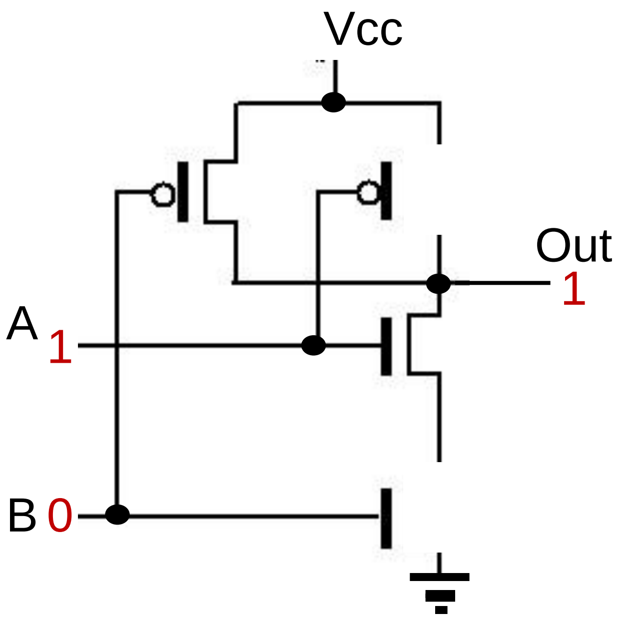
NOT $\bar{A}$

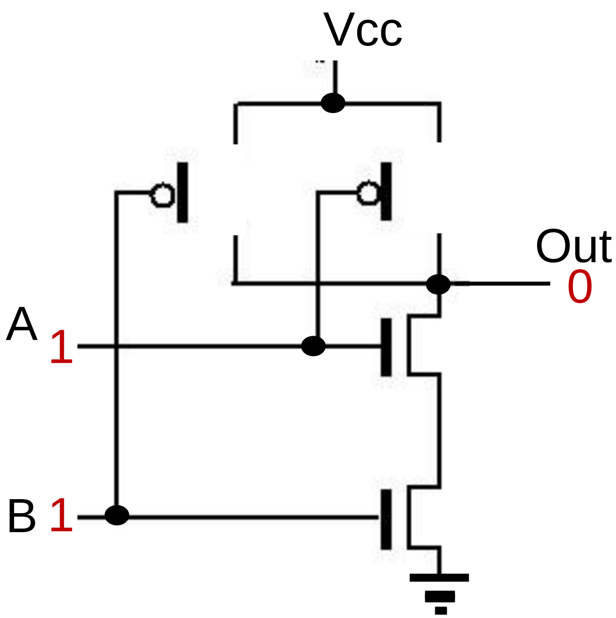
NAND gate



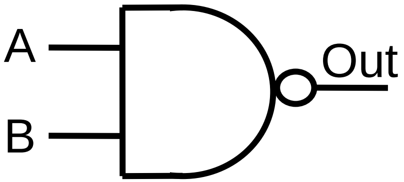


NAND gate



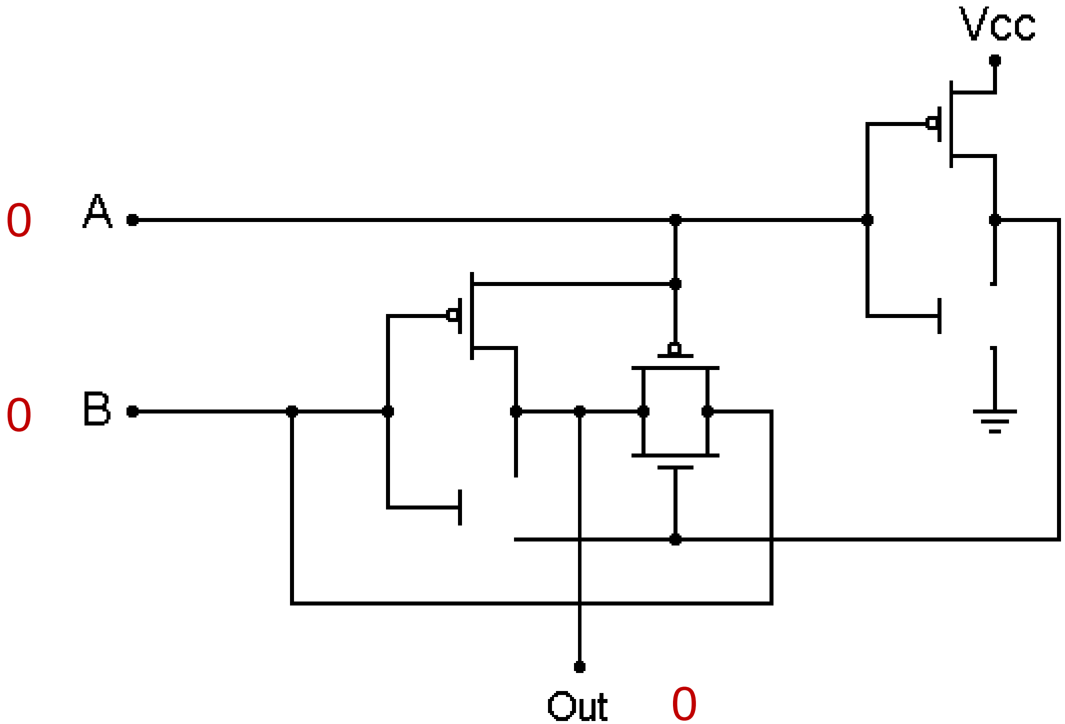


A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

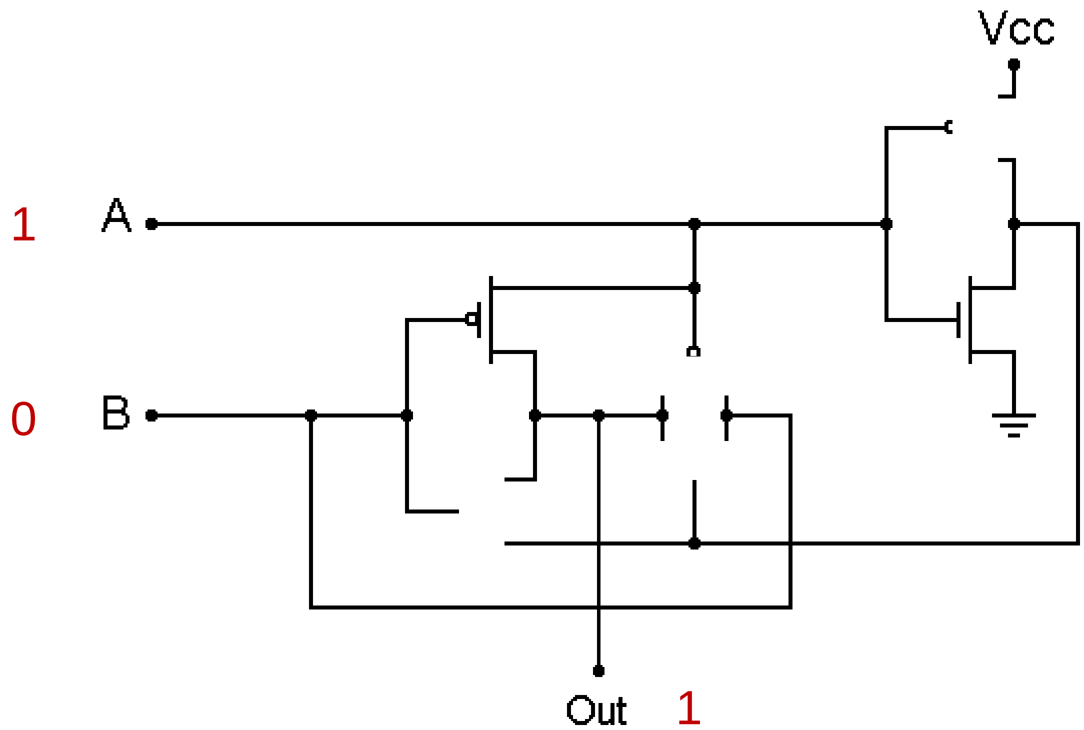


NAND $\overline{A*B}$

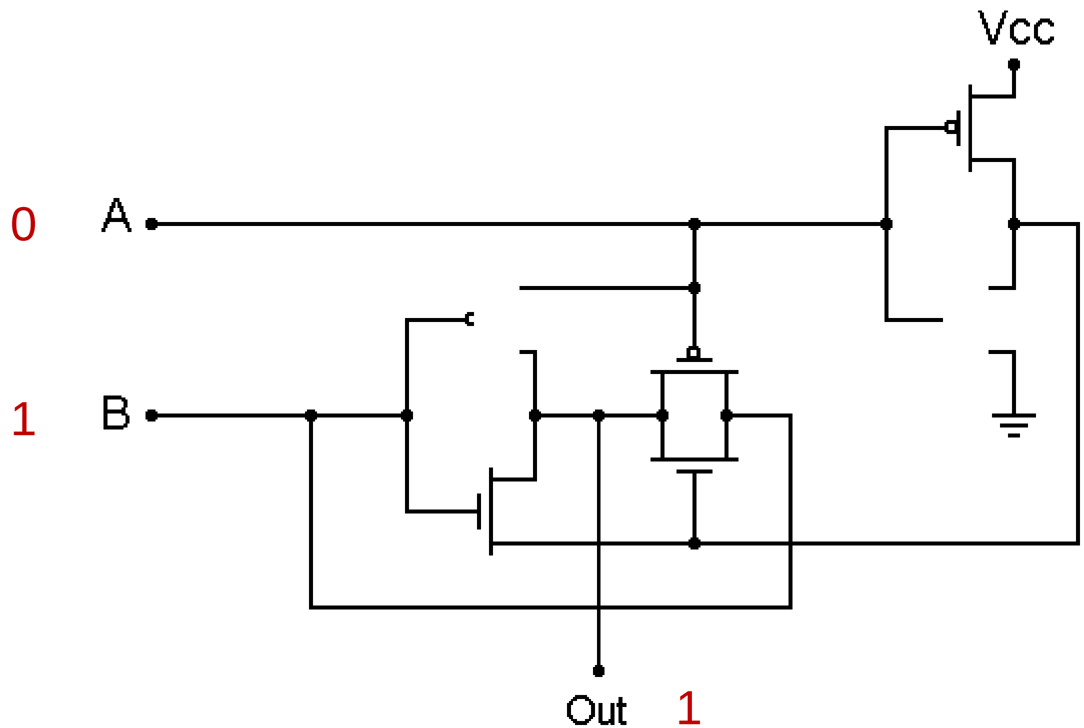
Basic digital elements - gates



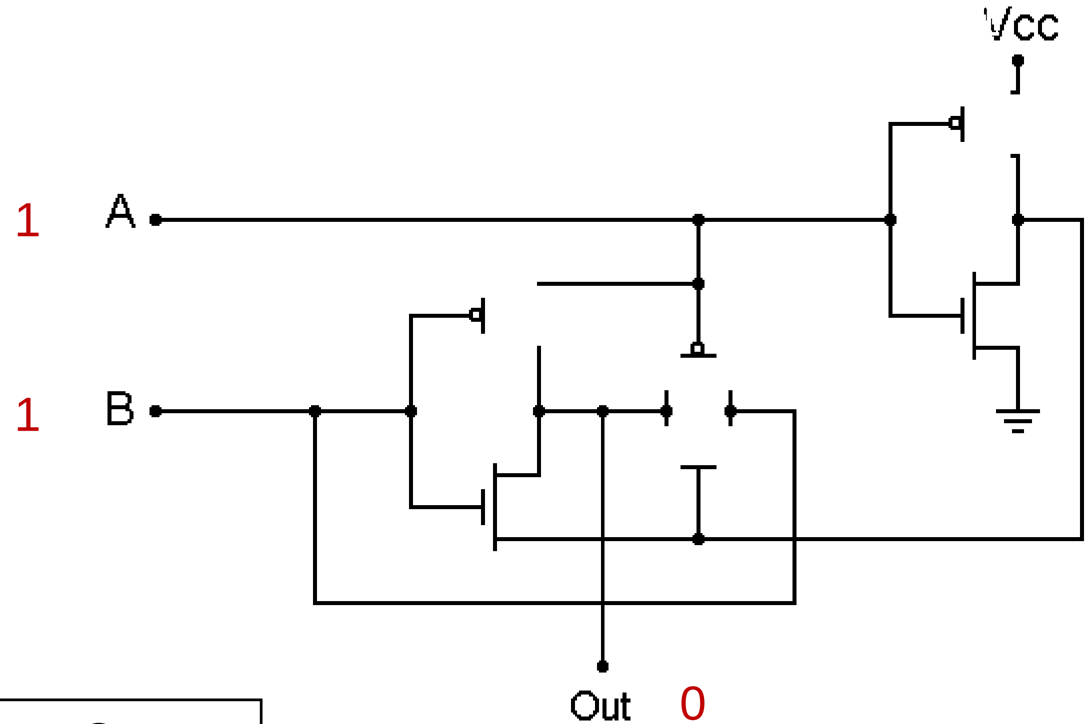
Basic digital elements - gates



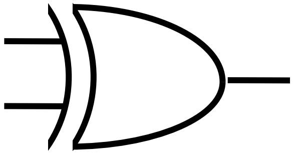
Basic digital elements - gates



Basic digital elements - gates



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

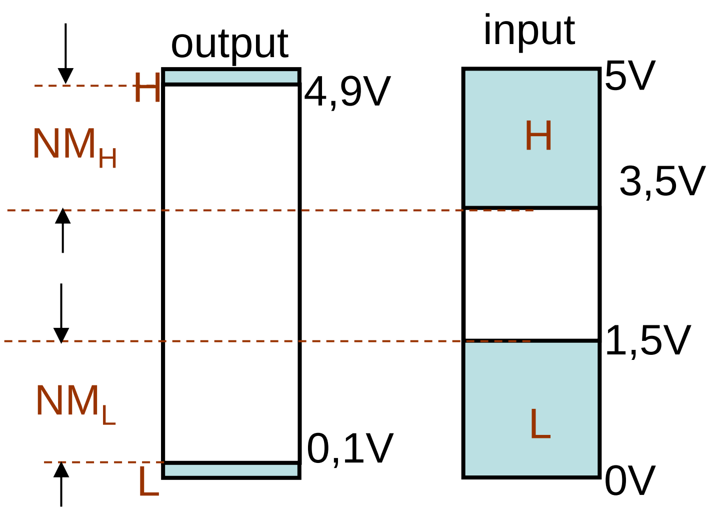


XOR

$A \oplus B$

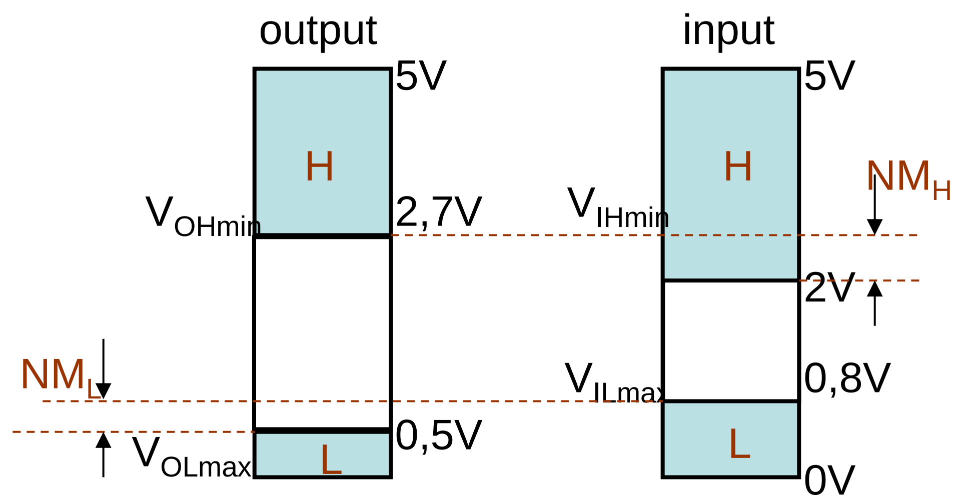
Noise margins

HCMOS

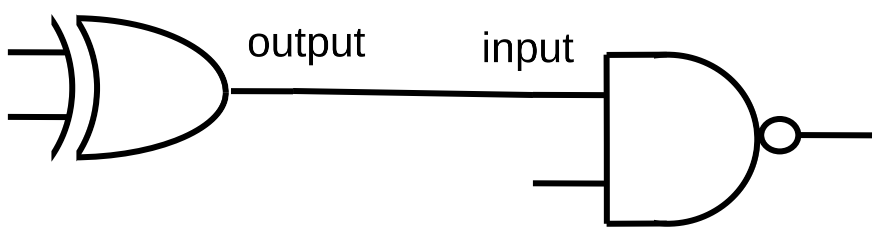


$NM_H = 1,4V$ $NM_L = 1,4V$

TTL



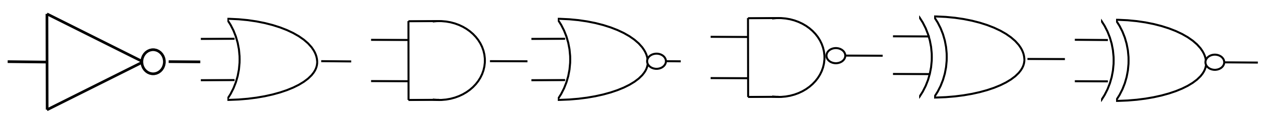
$NM_H = 0,7V$ $NM_L = 0,3V$



Basic digital elements - gates

a	b	$\overline{a}$	$a+b$	$a*b$	$\overline{a+b}$	$\overline{a*b}$	$a\oplus b$	$a\overline{\oplus}b$
0	0	1	0	0	1	1	0	1
0	1	1	1	0	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	0	1	1	0	0	0	1

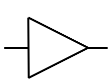
NOT OR AND NOR NAND XOR XNOR



only 2-input

Tri-state versions

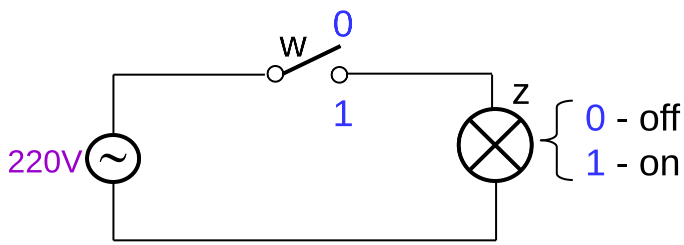
buffer



Introduction

Example 1.

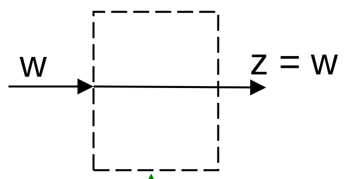
Find a logical model of a simple electrical circuit.



w	z
0	0
1	1

Truth table

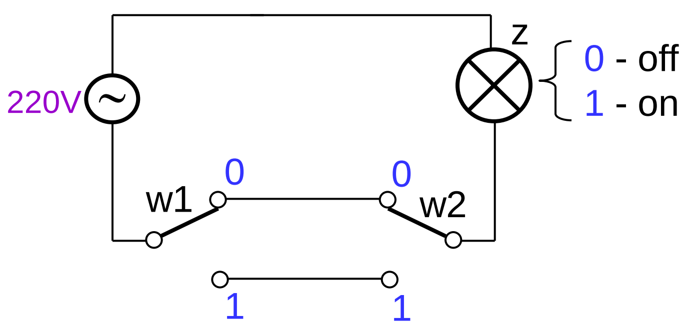
Model logiczny



no logic elements

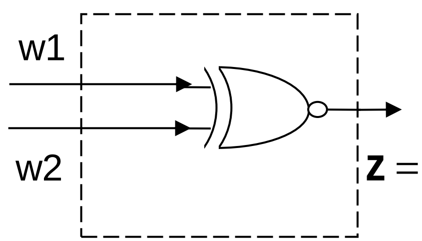
Example 2.

Find a logical model of 2-point hotel switch (corridor switch).



w1	w2	z
0	0	1
0	1	0
1	0	0
1	1	1

Logical model



$\mathbf{z} = \overline{\mathbf{w1} \oplus \mathbf{w2}}$

Logic element – XNOR gate

Introduction

Example 3.

Find a logical model of 3-point hotel switch.

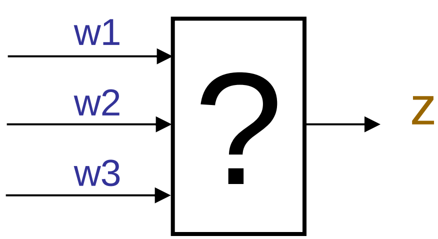
Electrical specification:

- For each state of any two of the switches, changing the state of the third switch should result in changing the bulb state (on/off).

Electrical diagram: try to draw by yourselves !

Logical specification:

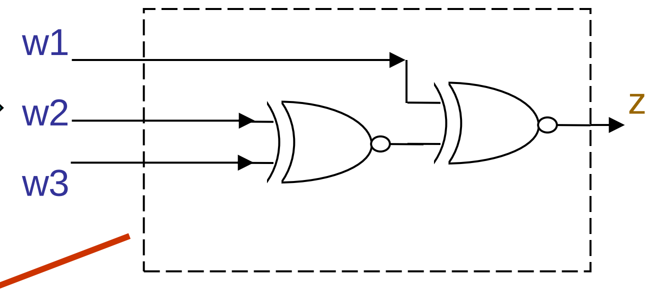
- For each logic state of any two of the inputs, changing logic state on the third input should result in changing logic state on the output.



w1	w2	w3	z	
0	0	0	0	on
0	0	1	1	off (w3)
0	1	1	0	on (w2)
0	1	0	1	off (w3)
1	1	0	0	on (w1)
1	1	1	1	off (w3)
1	0	1	0	on (w2)
1	0	0	1	off (w3)

Gray code

Logical model (logical diagram)



Two XNOR gates

How to find logical diagrams for the given specification ?

A little theory: Boolean Algebra definition (dual)

Boolean algebra : $A = \{B, +, *, \overline{\ }\}$

B – a set of at least 2 elements

B – closed under all the operators

B - contains 2 special elements **„0”** i **„1”**

+ , ***** - logic **addition** and logic **multiplication** (binary operators)

$\overline{\ }$ - **complement** operator (unary operator)

Four axioms are satisfied:

A1. Commutativity

$a+b = b+a \quad , \quad a*b = b*a$

A2. Distributivity

$a+(b*c) = (a+b)*(a+c) \quad , \quad a*(b+c) = a*b+a*c$

A3. Identity elements

$a + \text{"0"} = a \quad , \quad a * \text{"1"} = a$

A4. Complement

$a + \overline{a} = \text{"1"} \quad , \quad a * \overline{a} = \text{"0"}$

Theorems of Boolean Algebra

$\mathbf{a}+\mathbf{a}=\mathbf{a} \longrightarrow \mathbf{a}*\mathbf{a}=\mathbf{a}$

$\mathbf{a}+1=1 \longrightarrow \mathbf{a}*0=0$

$\overline{\overline{\mathbf{a}}}=\mathbf{a} \longrightarrow \overline{\overline{\mathbf{a}}}=\mathbf{a}$

$\mathbf{a}+(\mathbf{a}*\mathbf{b})=\mathbf{a} \longrightarrow \mathbf{a}*(\mathbf{a}+\mathbf{b})=\mathbf{a}$

$\mathbf{a}+(\overline{\mathbf{a}}*\mathbf{b})=\mathbf{a}+\mathbf{b} \longrightarrow \mathbf{a}*(\overline{\mathbf{a}}+\mathbf{b})=\mathbf{a}*\mathbf{b}$

Associativity:

$\mathbf{a}+(\mathbf{b}+\mathbf{c})=(\mathbf{a}+\mathbf{b})+\mathbf{c} \longrightarrow \mathbf{a}*(\mathbf{b}*\mathbf{c})=(\mathbf{a}*\mathbf{b})*\mathbf{c}$

de Morgan’s Laws:

$\overline{(\mathbf{a}+\mathbf{b})}=\overline{\mathbf{a}}*\overline{\mathbf{b}} \longrightarrow \overline{(\mathbf{a}*\mathbf{b})}=\overline{\mathbf{a}}+\overline{\mathbf{b}}$

Two-element Boolean Algebra

B = {0, 1}

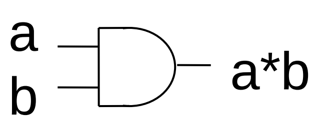
0, 1 identity elements

*, +, $\overline{}$

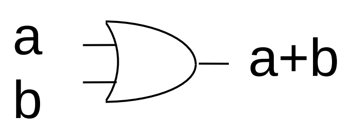
a	b	a * b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a + b
0	0	0
0	1	1
1	0	1
1	1	1

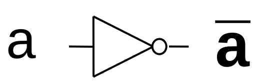
a	$\overline{a}$
0	1
1	0



AND



OR



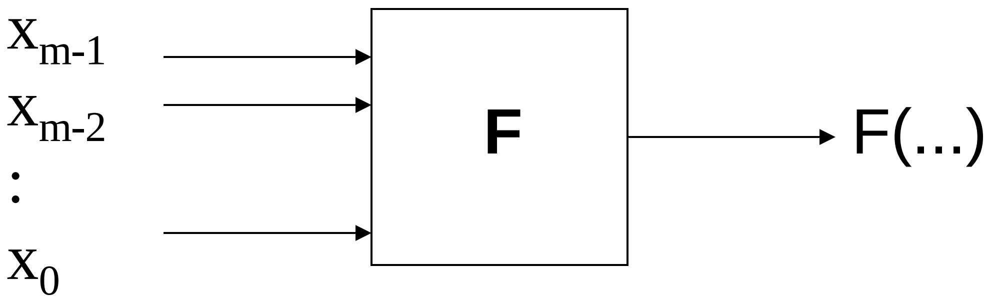
NOT

Are the axioms A1 - A4 satisfied?

Boolean function and Boolean formula

Boolean function of **m** variables is a map of the set $\{0,1\}^m$ into the set $\{0,1\}$.

$$F(x_{m-1}, x_{m-2}, \ldots, x_0) \quad : \quad \{0,1\}^m \rightarrow \{0,1\}$$



Boolean formula is an expression built of variables’ literals and symbols 0 , 1 , + , * , ¯ .

Example:

$$\mathbf{x} + \mathbf{y} * \overline{\mathbf{z}} + 1$$

Boolean function representations

- Truth table

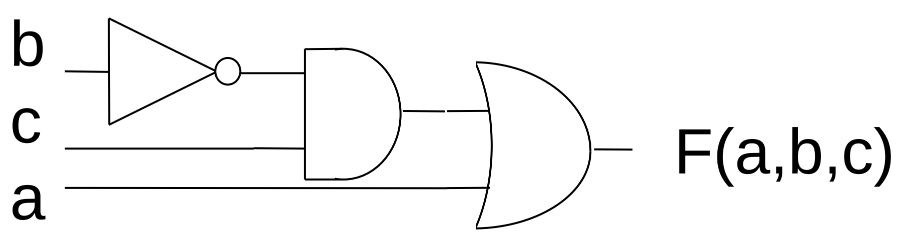
a	b	c	F(a,b,c)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

- Boolean formula (analytical)

$$\mathbf{F(a,b,c) = (a+b+c)(a+\overline{b})}$$

Note the lack of * symbol

- Logic diagram (logic gates)



Analytical forms for Boolean functions

Normal: $f = a(\overline{b} + \overline{c})$

Other: $f = ac + \overline{a}(\overline{b} + \overline{c})$

Conjunctive (PoS):

$f = a(\overline{b} + \overline{c})$

Disjunctive (SoP):

$f = a\overline{b} + a\overline{c}$

Canonical

Non-canonical:

$f = a(\overline{b} + \overline{c})$

Canonical

Non-canonical:

$f = a\overline{b} + a\overline{c}$

$$f(a,b,c) = a\overline{b}c + a\overline{b}\,\overline{c} + ab\overline{c} = \sum(4,5,6)$$

$$f(a,b,c) = (a+b+c)(a+b+\overline{c})(a+\overline{b}+c)(a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+\overline{c}) = \prod(0,1,2,3,7)$$

Minterms and maxterms

Minterm is an ordered logical product consisting of all the variables; Each variable appears only once (in affirmative or negative form).

Maxterm is an ordered logical sum consisting of all the variables; Each variable appears only once (in affirmative or negative form).

Example

m=3; (x_2,x_1,x_0)

Index		Minterm	Maxterm	f
0	000	$\bar{x}_2\bar{x}_1\bar{x}_0 = m_0$	$x_2 + x_1 + x_0 = M_0$	0
1	001	$\bar{x}_2\bar{x}_1 x_0 = m_1$	$x_2 + x_1 + \bar{x}_0 = M_1$	1
2	010	$\bar{x}_2 x_1\bar{x}_0 = m_2$	$x_2 + \bar{x}_1 + x_0 = M_2$	0
3	011	$\bar{x}_2 x_1 x_0 = m_3$	$x_2 + \bar{x}_1 + \bar{x}_0 = M_3$	1
4	100	$x_2\bar{x}_1\bar{x}_0 = m_4$	$\bar{x}_2 + x_1 + x_0 = M_4$	0
5	101	$x_2\bar{x}_1 x_0 = m_5$	$\bar{x}_2 + x_1 + \bar{x}_0 = M_5$	0
6	110	$x_2 x_1\bar{x}_0 = m_6$	$\bar{x}_2 + \bar{x}_1 + x_0 = M_6$	1
7	111	$x_2 x_1 x_0 = m_7$	$\bar{x}_2 + \bar{x}_1 + \bar{x}_0 = M_7$	1

Minterms and maxterms *cont..*

Every Boolean function can be expressed as a logical sum of minterms:

$$\mathbf{f} = \sum_{\mathbf{i}} \mathbf{m_i}$$

where decimal index „i” refers to all the minterms for which the function is equal to logic-1.

Every Boolean function can be expressed as a logical product of maxterms:

$$\mathbf{f} = \prod_{\mathbf{i}} \mathbf{M_i}$$

where decimal index „i” refers to all the maxterms for which the function is equal to logic-0.

Canonical forms

Example.

Find the canonical disjunctive form of **f**: $\mathbf{f(x,y,z) = xy + \bar{x}z}$

1. Analytical method:

$$f(x, y, z) = xy(z + \bar{z}) + \bar{x}z(y + \bar{y}) = xyz + xy\bar{z} + \bar{x}yz + \bar{x}\bar{y}z$$

2. Using the truth table:

	x y z	f
0	0 0 0	0
1	0 0 1	1
2	0 1 0	0
3	0 1 1	1
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

$$\mathbf{f(x,y,z)} = \sum(1,3,6,7) = \mathbf{m}_1 + \mathbf{m}_3 + \mathbf{m}_6 + \mathbf{m}_7 =$$
$$= \mathbf{\bar{x}\bar{y}z + \bar{x}yz + xy\bar{z} + xyz}$$

$$\mathbf{f(x,y,z)} = \prod(0,2,4,5) = \mathbf{M}_0\mathbf{M}_2\mathbf{M}_4\mathbf{M}_5 =$$
$$= \mathbf{(x+y+z)(x+\bar{y}+z)(\bar{x}+y+z)(\bar{x}+y+\bar{z})}$$

Negation of functions in normal forms

de Morgan's Laws – generalization:

$$\overline{(x_{n-1} \ldots x_1 x_0)} = \overline{x}_{n-1} + \ldots + \overline{x}_1 + \overline{x}_0 \xrightarrow{\text{duality}} \ldots\ldots\ldots$$

Example:

$$f(a,b,c) = (a+b)(\overline{a}+\overline{c})$$

$$\overline{f(a,b,c)} = \overline{(a+b)} + \overline{(\overline{a}+\overline{c})} = \overline{a}\,\overline{b} + ac$$

For canonical form:

$$f(a,b,c) = \sum(0,1,5,7) = m_0 + m_1 + m_5 + m_7$$

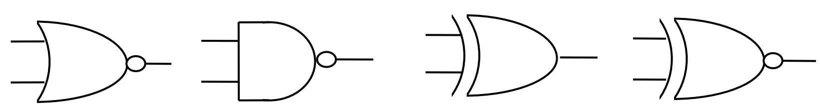
$$\overline{f(a,b,c)} = \overline{m}_0\overline{m}_1\overline{m}_5\overline{m}_7 = M_0 M_1 M_5 M_7 = \prod(0,1,5,7)$$

On the other hand, it is known (???) that:

$$\overline{f(a,b,c)} = m_2 + m_3 + m_4 + m_6 = \sum(2,3,4,6)$$

Elementary logic functions of two variables

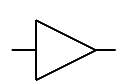
a	b	$\overline{a+b}$	$\overline{ab}$	$a \oplus b$	$a \overline{\oplus} b$
0	0	1	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	0	0	0	1
		NOR	NAND	XOR	XNOR



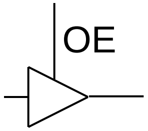
only 2-input

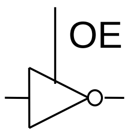
Additionally (from definition):
- logical addition OR
- logical multiplication AND
- complement (negation) NOT

buffer



Tri-state gates

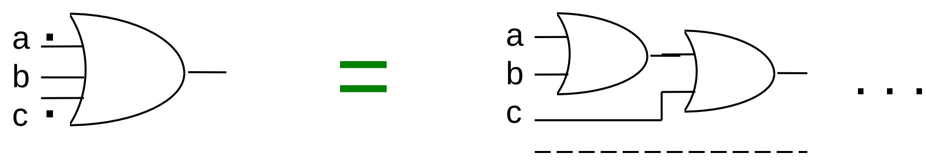




Elementary logic functions of multiple variables

OR:

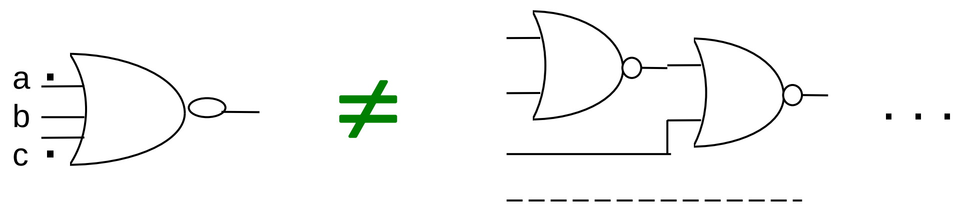
$$\mathbf{f} = \mathbf{a} + \mathbf{b} + \mathbf{c} + \ldots = ((\mathbf{a} + \mathbf{b}) + \mathbf{c}) + \ldots.$$



associative

NOR:

$$\mathbf{f} = \overline{\mathbf{a} + \mathbf{b} + \mathbf{c} + \ldots.} \neq \overline{(\overline{(\mathbf{a} + \mathbf{b})} + \mathbf{c})} + \ldots.$$



not associative

AND,XOR,XNOR: associative

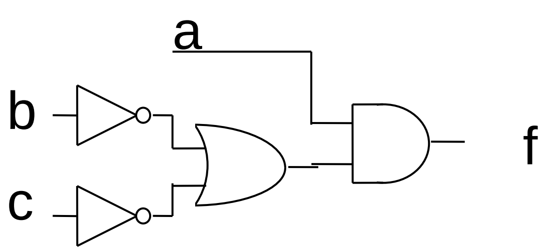
NAND: not associative

Simple cost factor of logic circuit design

K = < number of gates’ inputs, number of gates >

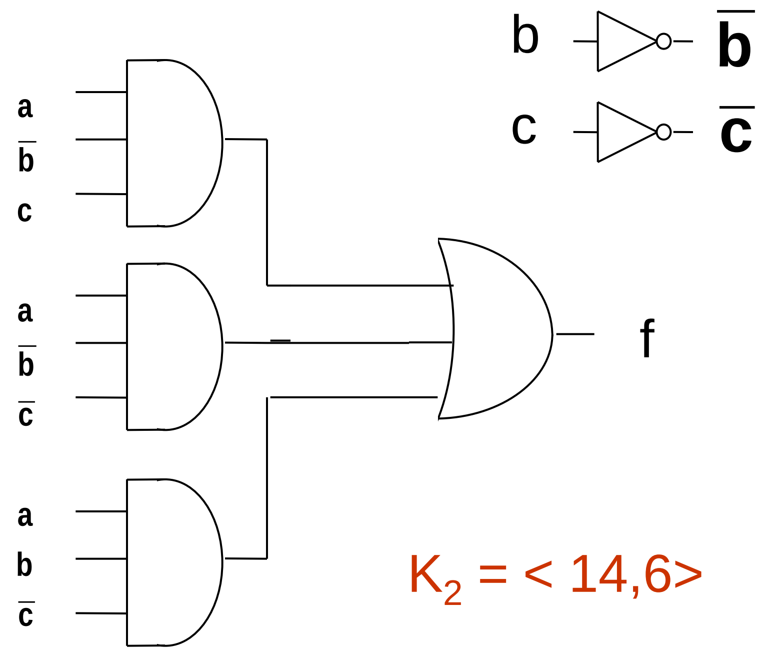
Example

$$\mathbf{f} = \mathbf{a}(\overline{\mathbf{b}} + \overline{\mathbf{c}})$$



K_1 = < 6,4>

$$\mathbf{f} = \mathbf{a}\overline{\mathbf{b}}\mathbf{c} + \mathbf{a}\overline{\mathbf{b}}\overline{\mathbf{c}} + \mathbf{a}\mathbf{b}\overline{\mathbf{c}}$$



K_2 = < 14,6>

Question: How to design the cheapest circuits?

Minimization of Boolean functions

Rules:

$$\boxed{\begin{array}{l} Ax + A\bar{x} = A \\ (A + x)(A + \bar{x}) = A \end{array}}$$

Classic minimization methods:

- algebraic
- Karnough maps
- Quine – McCluskey algorithm)

Algebraic method: heuristic algebraic transformations

Example

$$(a+b+c)\boxed{(a+\bar{b}+c)(a+\bar{b}+\bar{c})=}$$

$$(a+b+c)(a+\bar{b}+c\bar{c}) = (a+b+c)\boxed{(a+\bar{b})}=$$

$$a+(b+c)\bar{b}\ \boxed{=a+\bar{b}c}$$

Minimal form

Note. There can exist many different minimal forms of the same logic function

Minimization of Boolean functions (*cont.)*

Karnough Maps (up to 5 variables)

Gray code

a \ bc	00	01	11	10
0	0	0	1	1
1	1	1	0	1

$$(\bar{a}b)c + (\bar{a}b)\bar{c} = \bar{a}b$$

$$(a\bar{b})c + (a\bar{b})\bar{c} = a\bar{b}$$

$$(a\bar{c})\bar{b} + (a\bar{c})b = a\bar{c}$$

Neighbouring cells are:

- cells with common side
- extreme, opposite cells

Minimal disjunctive normal form

$$\boxed{f(\min) = a\bar{b} + \bar{a}b + a\bar{c}}$$

K = < 12,7 >

Minimization of Boolean functions (*cont.)*

Karnough Maps (cont.)

a \ bc	00	01	11	10
0	0	0	1	1
1	1	1	0	1

$$[(\mathbf{a}+\mathbf{b})+\mathbf{c}][(\mathbf{a}+\mathbf{b})+\overline{\mathbf{c}}]=\mathbf{a}+\mathbf{b}$$

$$\overline{\mathbf{a}}+\overline{\mathbf{b}}+\overline{\mathbf{c}}$$

$$\boxed{\mathbf{f}=(\mathbf{a}+\mathbf{b})(\overline{\mathbf{a}}+\overline{\mathbf{b}}+\overline{\mathbf{c}})}$$

Minimal conjunctive normal form

K = < 10,6 >

- merging of minterm or maksterms we will call glueing
- lack of efficient, strict algorithm
- you can glue 2,4,8,....,2^n „ones” (or ”zeros”)
- you should glue the most numerous groups first

Minimization of Boolean functions (*cont.)*

Karnough Maps (cont.)

ab \ cd	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	0	0	1
10	1	0	0	1

$$\mathbf{f} = \overline{\mathbf{d}} + \overline{\mathbf{a}}\,\overline{\mathbf{c}} + \overline{\mathbf{a}}\,\overline{\mathbf{b}}$$

ab \ cd	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	0	0	1
10	1	0	0	1

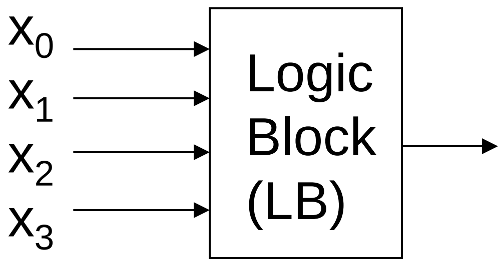
$$\mathbf{f} = (\overline{\mathbf{a}} + \overline{\mathbf{d}})(\overline{\mathbf{b}} + \overline{\mathbf{c}} + \overline{\mathbf{d}})$$

Students are to learn minimization of 5-variable functions by themselves!

Functions with don’t care output states

- functions which output is not specified for at least one input variable values

Example



$$f = x_3x_2x_1x_0 + \bar{x}_3x_2x_1x_0$$

Thermometric code:

Temp.	$x_3x_2x_1x_0$	f	
0	0 0 0 0	0	**Off-set**
1	0 0 0 1	0	
2	0 0 1 1	0	
3	0 1 1 1	1	**On-set**
4	1 1 1 1	1	
...	....	-	**Dc-set**

x_3x_2 \ x_1x_0	00	01	11	10
00	0	0	0	-
01	-	-	1	-
11	-	-	1	-
10	-	-	-	-

$f = x_2$

LB is not needed !

Functions with don’t care output states (cont.)

These functions appear in two situations:

- some binary sequences never occur on input **(as in example)**
- for some input sequences actual output value is not important

Minimization using Karnough maps with don’t care states:

- some don’t care states („–”) in Karnough map are treated as „ones”, and others as „zeros” so as to obtain the most numerous sets of glued cells
- don’t glue the cells in which there are only don’t care states: „–”

1

2

3

ab \ cd	00	01	11	10
00	1	-	-	1
01	-	-	0	1
11	1	0	0	0
10	1	0	0	-

ab \ cd	00	01	11	10
00	1	-	-	1
01	-	-	0	1
11	1	0	0	0
10	1	0	0	-

?

ab \ cd	00	01	11	10
00	1	-	-	1
01	-	-	0	1
11	1	0	0	0
10	1	0	0	-

?