HSBI

Hochschule Bielefeld

University of Applied Sciences and Arts



Image by Qwen-Image 2.1

Logic operations

DET-02 · Digital Electronics

GUIDING QUESTIONS

- A car has four doors and a boot: which logic switches on the interior light if any of them is opened?
- Why does a single warning lamp light up as soon as any one of a dozen faults occurs?
- With just two signals, the energy supplier can make a heat pump pause, run normally or step up its heating – how does that work?



Image by Qwen-Image 2.1

INTENDED LEARNING OUTCOMES

- You analyse given circuits by means of their logic expression, truth table and timing behaviour.
- You transform logic expressions with Boolean algebra.
- You recognize the VHDL code of a logic operation and sketch the corresponding circuit.



Image by Qwen-Image 2.1

AGENDA

1. From the transistor to the CMOS circuit
2. Forms of representation
3. Basic gates
4. Derived operations
5. Boolean algebra
6. Analysis of given circuits
7. What have you learnt?



Image by Qwen-Image 2.1

REVIEW OF DET-01

- Analogue signals map information continuously, digital signals encode it with 0 and 1
- Bit: the smallest unit of information; positive logic: LOW = 0; HIGH = 1
- Positional number systems: decimal, binary, octal, hexadecimal
- Binary arithmetic and two's complement
- Today: the circuits that process the bits

HSBI

Hochschule Bielefeld

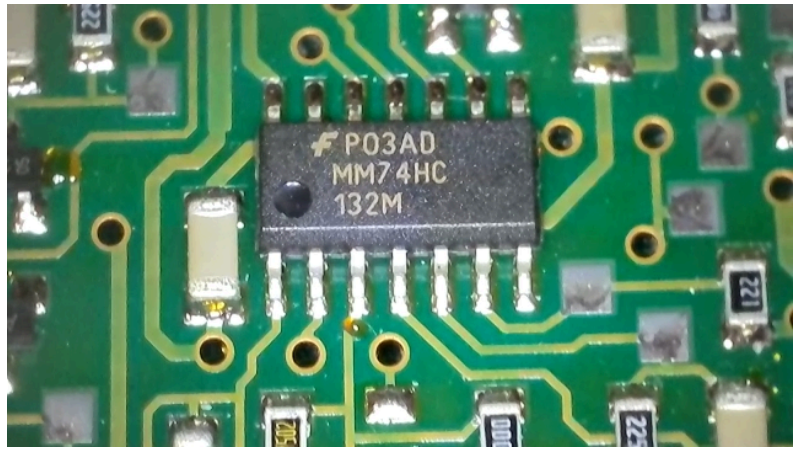
University of Applied Sciences and Arts

1 From the transistor to the CMOS circuit

- Processors, gates and transistors
- Inside the MOSFET
- The MOSFET as a switch

PROCESSORS, GATES AND TRANSISTORS

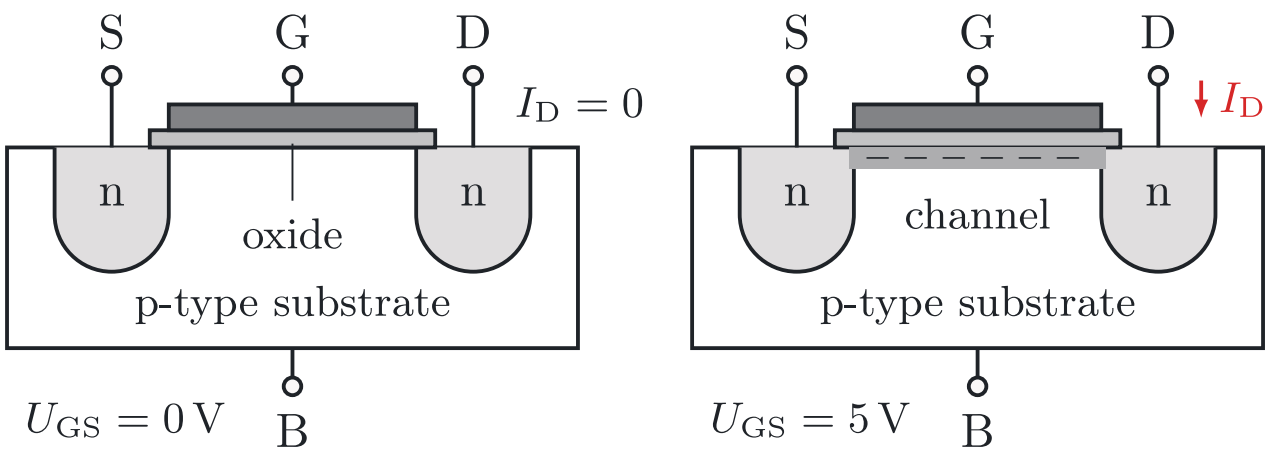
- A processor consists of logic gates, a logic gate of transistors
- Integrated circuit (IC): billions of transistors on one chip
- Bipolar junction transistor (BJT): controlled by a current
- Metal oxide semiconductor field-effect transistor (MOSFET): controlled by a voltage
- Digital electronics uses almost only MOSFETs: control without power, no current at rest in complementary metal oxide semiconductor (CMOS) technology, small size
- Limit: resistance and gate charge – every logic gate needs time



Integrated circuit 74HC132 with four gates

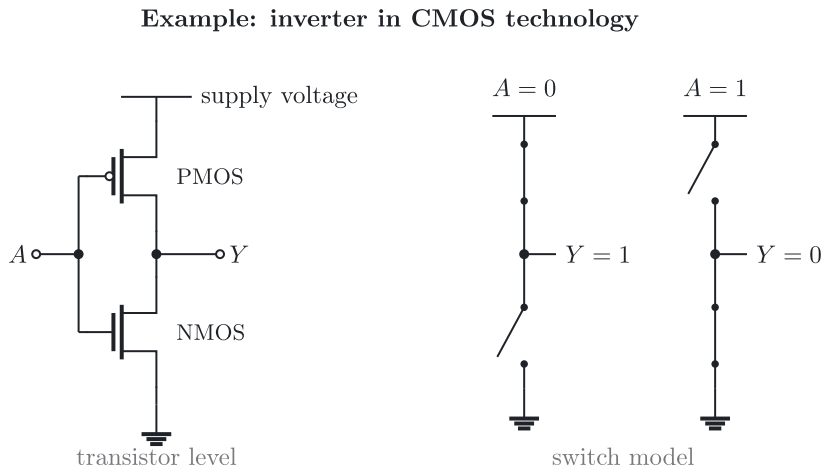
INSIDE THE MOSFET

- Gate – as in a fence: its field opens and closes the channel
- Metal, oxide, semiconductor: the layers under the gate
- n-channel MOSFET (NMOS): n regions in a p substrate
- p-channel MOSFET (PMOS): built the other way round



THE MOSFET AS A SWITCH

- The MOSFET as an electrically controlled switch
- NMOS closes at 1, PMOS at 0
- *Complementary*: one switches on when the other switches off
- CMOS inverter: always one transistor off, current only while switching



2 Forms of representation

- Seven ways of describing a gate
- Truth table
- Logic expression
- Schematic symbol
- Contact logic
- Karnaugh map
- Timing diagram
- VHDL – circuits described as text

SEVEN WAYS OF DESCRIBING A GATE

- One function, seven forms of representation – each for its own purpose

Form of representation	Type	Main use
Schematic symbol	graphical	design and documentation of circuits
Contact logic	schematic of contacts	intuitive approach; relay and contactor controls
Truth table	table	complete definition of the function
Logic expression	mathematical	design and transformation with Boolean algebra
VHDL code	text	description, simulation and synthesis for programmable devices
Karnaugh map	graphical	minimization, from DET-03
Timing diagram	graphical, over time	behaviour over time, delays

- Example AND gate: the display is off if a call is active (A) and the sensor is covered (B)

TRUTH TABLE

- Every combination of the inputs with the value of the output
- Ascending like a binary count; the leftmost input has the highest weight
- n inputs: 2^n rows

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

- A: call active, B: proximity sensor covered, Y: display off

LOGIC EXPRESSION

- The output as an equation of the inputs

$$Y = A \cdot B$$

- Read: ‘Y equals A and B’
- Dot for AND, plus for OR, bar for NOT

SCHEMATIC SYMBOL

- International Electrotechnical Commission (IEC): a rectangle with a qualifying symbol
- In DET, IEC 60617 only; the American symbol grey



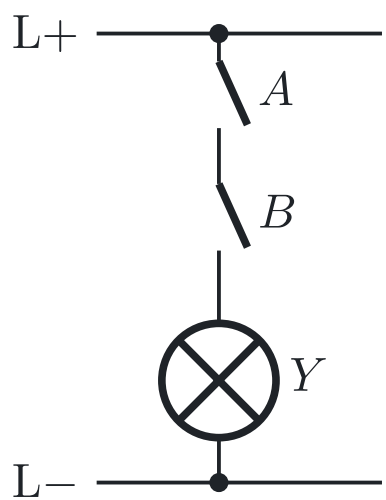
IEC 60617



IEEE 91 / ANSI Y32.14

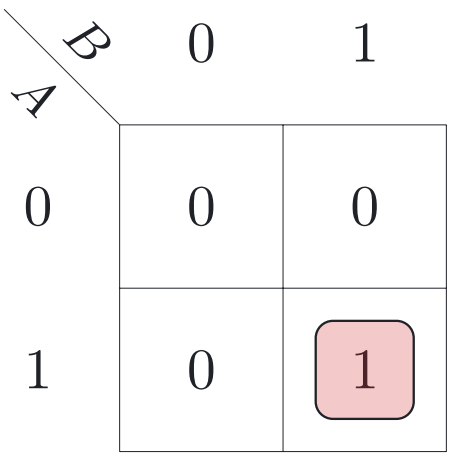
CONTACT LOGIC

- Inputs: switches or relay contacts; output: a lamp between L+ and L−
- Normally open contact = variable, normally closed contact = inverted variable
- AND: two normally open contacts in series



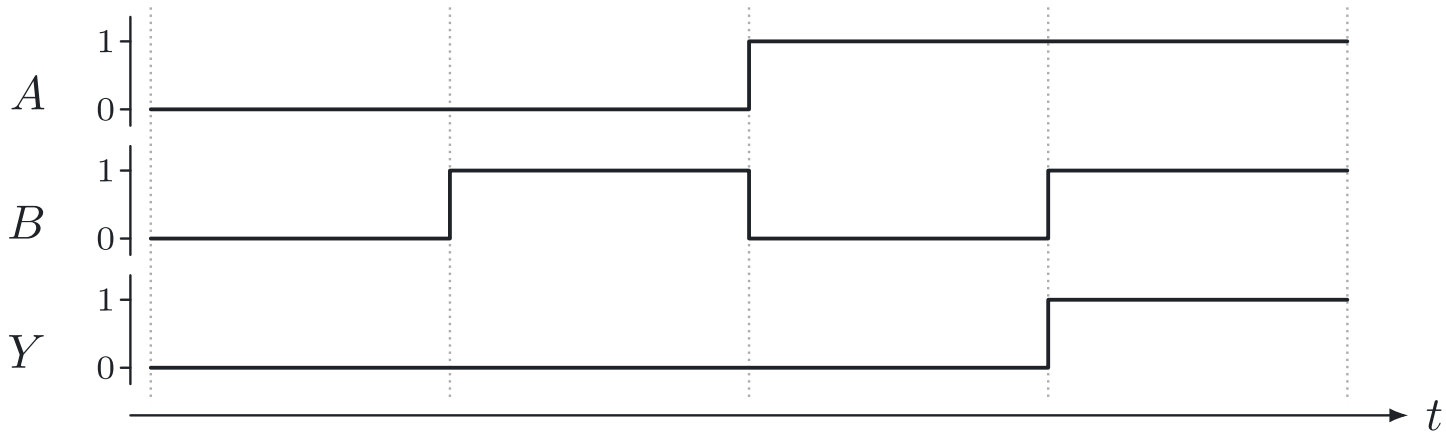
KARNAUGH MAP

- The truth table as a grid: one cell for each combination of the inputs
- With two or more inputs; for minimization in DET-03



TIMING DIAGRAM

- The signals over time, each either 0 or 1
- The inputs run through all combinations in the order of the truth table
- Here ideal, without delay



VHDL – CIRCUITS DESCRIBED AS TEXT

- Very high speed integrated circuit hardware description language (VHDL): describes hardware as text
- `entity`: inputs and outputs, `architecture`: behaviour
- `<=` assigns; all assignments work at the same time
- One operator for each gate: `not`, `and`, `or`, `nand`, `nor`, `xor`, `xnor`
- Goal: recognize the code, sketch the circuit

```vhdl
-- A: call active
-- B: proximity sensor covered
-- Y: display off
entity display_off is
  port (A, B : in  bit;
        Y    : out bit);
end entity;

architecture behaviour of display_off is
begin
  Y <= A and B;
end architecture;
```

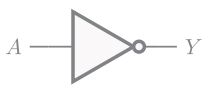
3 Basic gates

- The NOT gate
- The AND gate
- The OR gate
- Gates with more than two inputs
- Propagation delay

NOT GATE

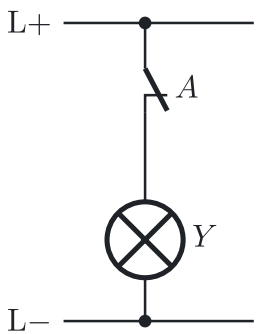


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



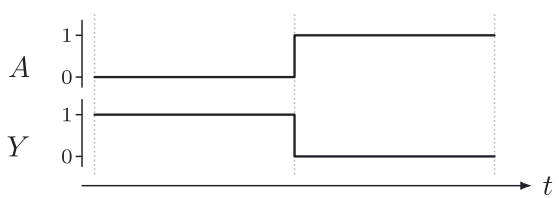
Contact logic

A	Y
0	1
1	0

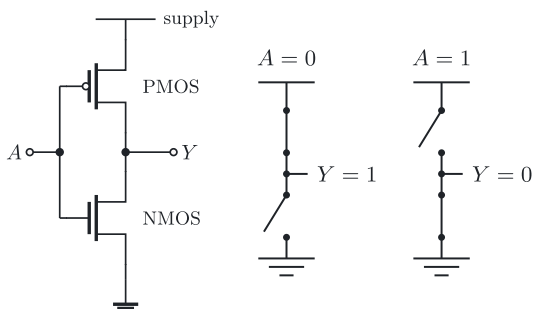
Truth table

$$Y = \overline{A}$$

Expression



Timing diagram – Open in the simulator



CMOS circuit

AND GATE

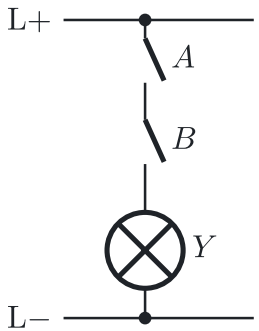


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



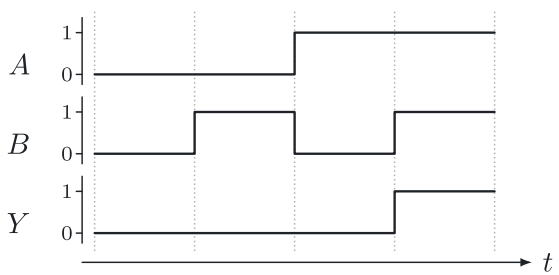
Contact logic

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

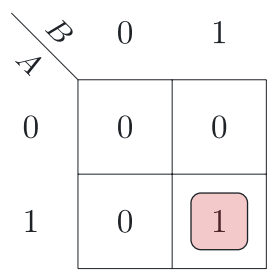
Truth table

$$Y = A \cdot B$$

Expression



Timing diagram – Open in the simulator



Karnaugh map

OR GATE

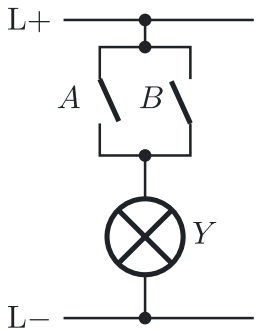


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



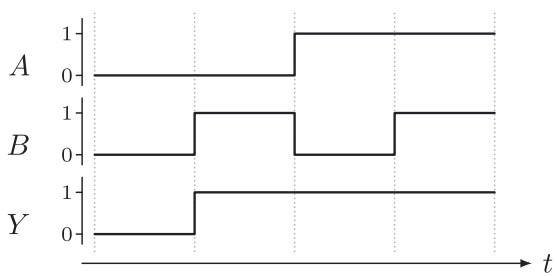
Contact logic

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

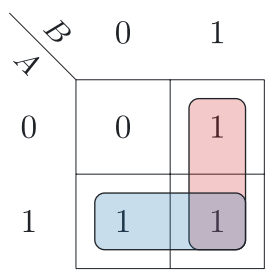
Truth table

$$Y = A + B$$

Expression



Timing diagram – Open in the simulator



Karnaugh map

MORE THAN TWO INPUTS

- AND: 1 only if all inputs are 1; OR: 1 if at least one input is 1
- Interior light: an OR of four doors and the boot

$$Y = A + B + C + D + E$$

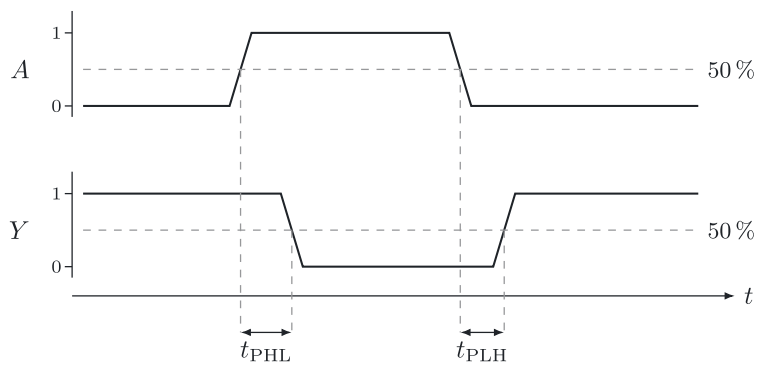
- Y: interior light on, inputs A to D: doors 1 to 4 open, E: boot open
- Warning lamp: the OR of all fault bits – the collective fault
- Fan-in: a gate has a limited number of inputs – combine smaller gates



Image by Qwen-Image 2.1

PROPAGATION DELAY

- A real gate needs time: from 50 % of the input to 50 % of the output



- 74HC00: typically 9 ns at 4.5 V

- Origin: the transistor charges or discharges the capacitance of the following inputs and lines

$$t_{\mathrm{s}} \approx R_{\mathrm{on}} \cdot C_{\mathrm{L}}$$

- t_{s}: switching time, R_{on}: resistance of the switched-on transistor, C_{L}: capacitance at the output, the load
- Faster switching costs more power
- Delays add up; false pulses: hazards in DET-04

4 Derived operations

- The NAND gate
- The NOR gate
- The XOR gate
- The XNOR gate
- IEC and American symbols side by side

NAND GATE

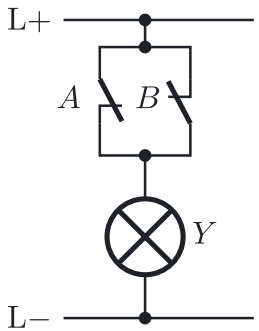


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



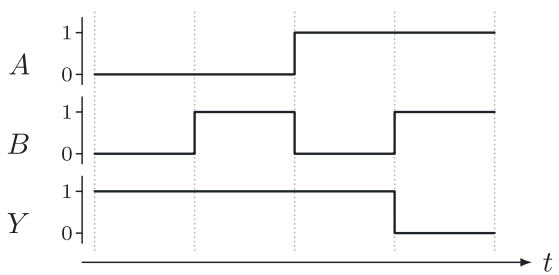
Contact logic

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

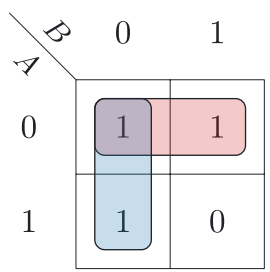
Truth table

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

Expression



Timing diagram – Open in the simulator



Karnaugh map

NOR GATE

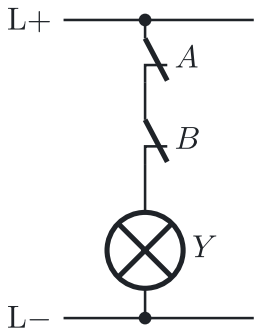


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



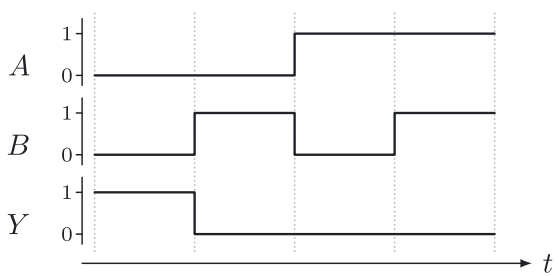
Contact logic

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

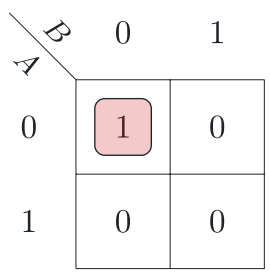
Truth table

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

Expression

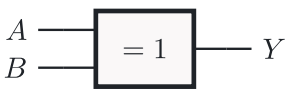


Timing diagram – Open in the simulator



Karnaugh map

XOR GATE

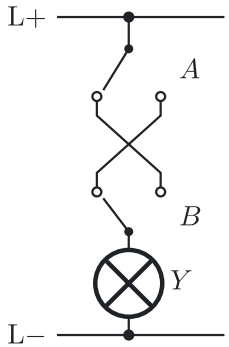


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



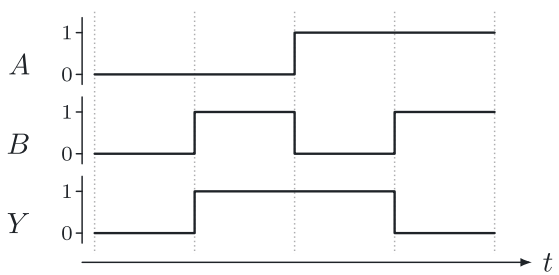
Contact logic

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

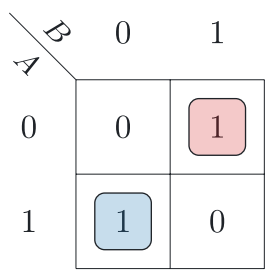
Truth table

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

Expression

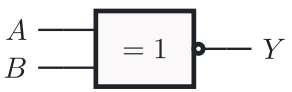


Timing diagram – Open in the simulator



Karnaugh map

XNOR GATE

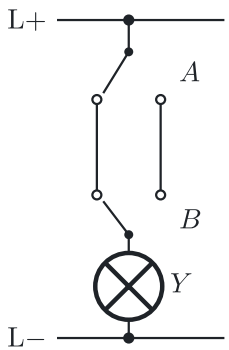


IEC 60617



IEEE 91 / ANSI Y32.14

Schematic symbol



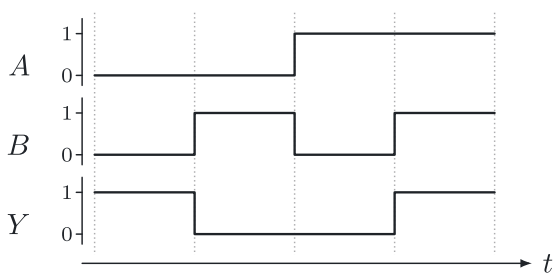
Contact logic

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

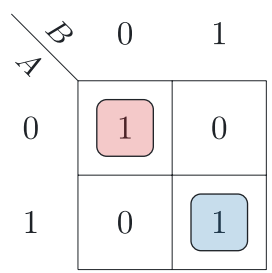
Truth table

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

Expression



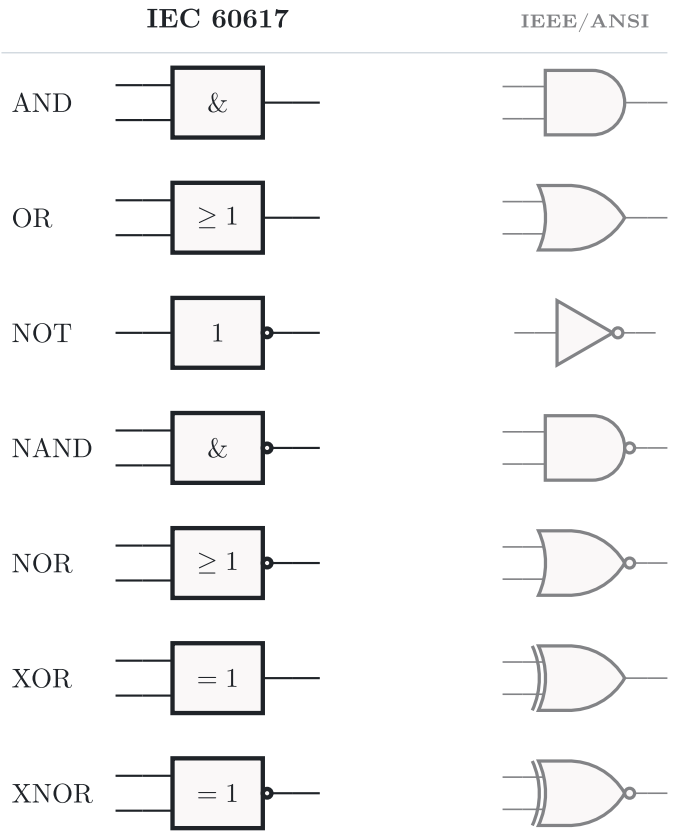
Timing diagram – Open in the simulator



Karnaugh map

IEC AND AMERICAN SYMBOLS

- The qualifying symbol states the rule: & all inputs 1, ≥1 at least one, =1 exactly one
- A circle at the output negates it
- American symbols: a shape of their own for each operation
- In DET, all circuits according to IEC 60617



5 Boolean algebra

- Notations of Boolean algebra
- Axioms and precedence
- Theorems with one variable
- Theorems with several variables
- De Morgan's laws
- Transforming expressions step by step

NOTATIONS OF BOOLEAN ALGEBRA

- Three disciplines, three notations for the logic variables A and B – in DET, the notation of electrical engineering

Operation	Mathematics and logic	Computer science	Electrical engineering
AND	$A \wedge B$	`A && B` or `A & B`	$A \cdot B$ or AB
OR	$A \vee B$	`A || B` or `A | B`	$A + B$
NOT	$\neg A$	`!A` or `~A`	$\overline{A}$ or A'
NAND	$\neg(A \wedge B)$	`!(A && B)`	$\overline{A \cdot B}$
NOR	$\neg(A \vee B)$	`!(A || B)`	$\overline{A + B}$
XOR	$A \oplus B$	`A ^ B`	$A \oplus B$
XNOR	$A \Leftrightarrow B$	`A == B`	$\overline{A \oplus B}$

AXIOMS AND PRECEDENCE

- Axioms: basic statements without proof – the values 0 and 1, NOT, AND, OR
- Precedence: NOT before AND, AND before OR

$$Y = \overline{A} \cdot B + C = \left(\left(\overline{A}\right) \cdot B\right) + C$$

- Output Y, inputs A, B and C, each 0 or 1
- VHDL: only `not` takes precedence – mixed operators need parentheses

THEOREMS WITH ONE VARIABLE

- Proof by exhaustion: try every combination
- Theorems in pairs: one for OR, one for AND

Theorem	With OR	With AND
Identity	$A + 0 = A$	$A \cdot 1 = A$
Null element	$A + 1 = 1$	$A \cdot 0 = 0$
Idempotence	$A + A = A$	$A \cdot A = A$
Complements	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
Involution, $\overline{\overline{A}} = A$	–	–

THEOREMS WITH SEVERAL VARIABLES

- Rules for transforming expressions – the basis for minimization

Theorem	With OR	With AND
Commutative property	$A + B = B + A$	$A \cdot B = B \cdot A$
Associative property	$(A + B) + C = A + (B + C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Distributive property	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
Absorption	$A + A \cdot B = A$	$A \cdot (A + B) = A$
Uniting	$A \cdot B + A \cdot \overline{B} = A$	$(A + B) \cdot (A + \overline{B}) = A$

- Logic variables A, B and C, each 0 or 1
- Careful: NAND and NOR are not associative

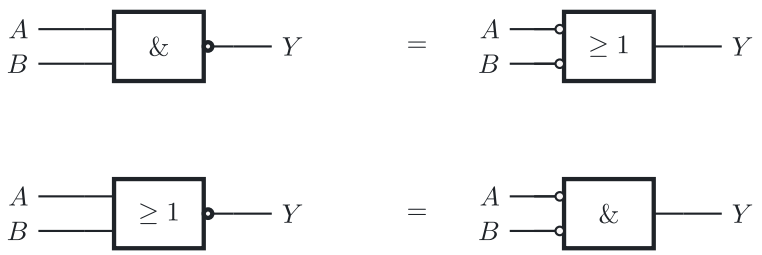
DE MORGAN'S LAWS

- NAND = OR with inverted inputs,
  NOR = AND with inverted inputs

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

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

- Break the bar, change the operator



- Bridge from AND and OR to the NAND and NOR gates of CMOS

TRANSFORMING STEP BY STEP

- The theorems turn an expression into a shorter one with the same function

$$
\begin{aligned}
Y &= \overline{A} \cdot B + A \cdot \overline{B} + A \cdot B \\
&= \overline{A} \cdot B + A \cdot B + A \cdot \overline{B} + A \cdot B && \text{idempotence, commutative property} \\
&= B \cdot \left(\overline{A} + A\right) + A \cdot \left(\overline{B} + B\right) && \text{commutative and distributive properties} \\
&= B \cdot 1 + A \cdot 1 && \text{complements} \\
&= A + B && \text{identity, commutative property}
\end{aligned}
$$

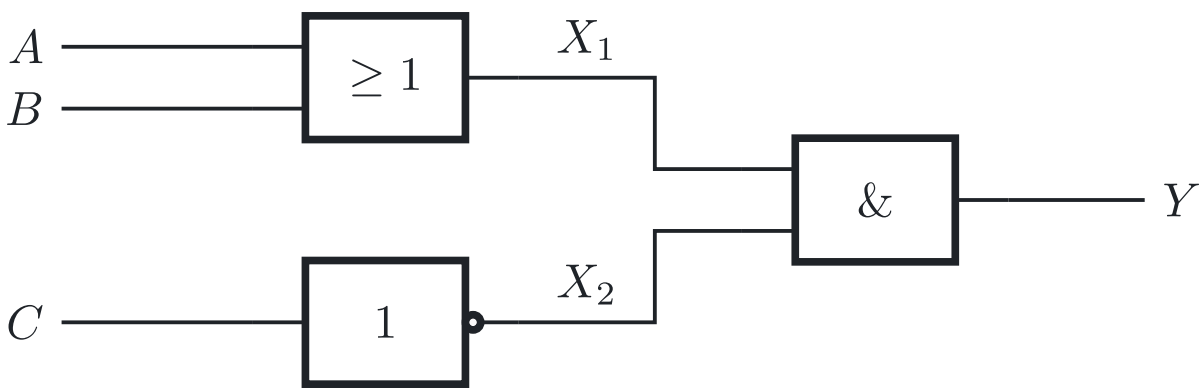
- A: power button pressed, B: phone lifted, Y: screen on
- Functionally complete: every function can be built – also from NAND alone

6 Analysis of given circuits

- From the circuit to the expression
- From the circuit to the truth table
- Timing behaviour of a circuit
- From VHDL code to the circuit
- The same circuit in a programmable logic controller
- Two bits, four operating states

FROM THE CIRCUIT TO THE EXPRESSION

- Label the intermediate nodes, then work from the inputs to the output
  - A: power button pressed, B: phone lifted, C: proximity sensor covered during a call, Y: display on



$$X_1 = A + B$$
$$X_2 = \overline{C}$$
$$Y = X_1 \cdot X_2 = (A + B) \cdot \overline{C}$$

FROM THE CIRCUIT TO THE TRUTH TABLE

- Evaluate every node for every combination, from left to right

A	B	C	$X_1 = A + B$	$X_2 = \overline{C}$	$Y = X_1 \cdot X_2$
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

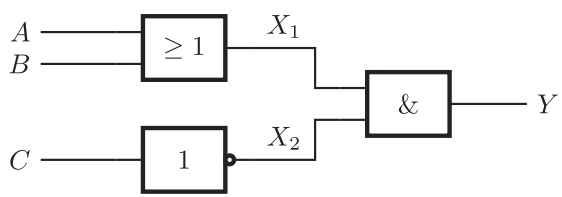
Open in the simulator

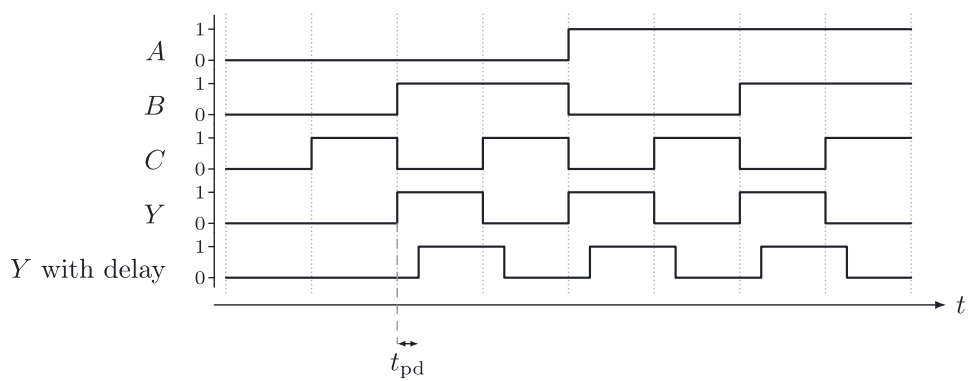
TIMING BEHAVIOUR OF A CIRCUIT

- The delay of a circuit is the delay of its longest path
- Stage by stage: the slowest input of a gate plus its own delay

$$t_{\mathrm{pd,circuit}} = \max\left(t_{\mathrm{pd,OR}},\ t_{\mathrm{pd,NOT}}\right) + t_{\mathrm{pd,AND}}$$

$$t_{\mathrm{pd,circuit}} = \max\left(9\,\mathrm{ns},\ 9\,\mathrm{ns}\right) + 9\,\mathrm{ns} = 18\,\mathrm{ns}$$





FROM VHDL CODE TO THE CIRCUIT

- Every operator becomes a gate, every signal an intermediate node; the order of the lines does not matter

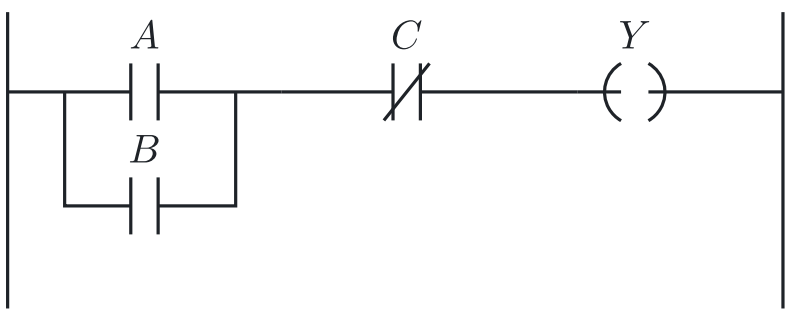
```
entity display_on is
  port (A, B, C : in  bit;   -- A: power button, B: phone lifted, C: covered in a call
        Y       : out bit);  -- Y: display on
end entity;

architecture behaviour of display_on is
  signal X1, X2 : bit;       -- intermediate nodes
begin
  X1 <= A or B;
  X2 <= not C;
  Y  <= X1 and X2;
end architecture;
```

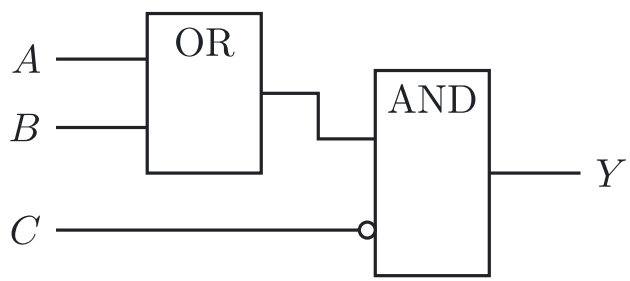
THE SAME CIRCUIT IN A CONTROLLER

- Programmable logic controller (PLC): programming languages according to IEC 61131-3
- Ladder diagram (LD): contact logic turned by 90 degrees
- Function block diagram (FBD): function blocks like schematic symbols

Ladder diagram



Function block diagram



- Instruction list (IL): text, one instruction per line – no longer in the edition of 2025

TWO BITS, FOUR OPERATING STATES

- Smart Grid Ready (SG Ready): two switching signals A and B, four operating states – one gate per state

A	B	State	Meaning
0	0	2	normal operation
0	1	3	boost: a recommendation to switch on, with increased set points for heating and hot water
1	0	1	block: the heat pump must not start, for at most two hours
1	1	4	start command

$$Y_1 = A \cdot \overline{B} \quad \text{block}$$
$$Y_2 = \overline{A + B} \quad \text{normal operation}$$
$$Y_3 = \overline{A} \cdot B \quad \text{boost}$$
$$Y_4 = A \cdot B \quad \text{start command}$$

Open in the simulator

7 What have you learnt?

- The building blocks of knowledge
- Answers to the guiding questions

WHAT YOU HAVE LEARNT

- Basic gates and derived logic operations
- Schematic symbols according to IEC 60617 and the American symbols
- Seven forms of representation
- Analysis of given circuits
- Rules of Boolean algebra and its notations
- Propagation delay

ANSWERS TO THE GUIDING QUESTIONS

- Interior light: an OR of the five door contacts
- Warning lamp: the OR of all fault bits – the collective fault
- Heat pump: two bits give four combinations, one gate per operating state



Image by Qwen-Image 2.1

Your turn: first on your own, then together.