Digital technology

Information can be presented in various forms. For computer analysis of data, a coding that uses sequences of 0s and 1s is used. The numeral system that uses only the numbers 0 and 1 is called the binary system. It is possible to express any information in this system. Computer systems use different types of coding. The most famous is the ASCII code (American Standard Code of Information Interchange). Using seven bits it can accommodate 128 symbol combinations. It has 32 control symbols (e.g. end of row, new row, etc), 10 number symbols (0 to 9), lower and upper case (English) alphabet and numerous other symbols.

Binary information in digital systems (hardware) is processed using logical circuits. Logical functions can be written in Boolean algebra. Boolean algebra expresses the logical state of a given situation that may occur.

Example: The school bell *B* will ring (*B* will be true) if and only if classes *D* are in session (*D* is true) and the time allotted *T* to the class has passed (*T* is true). This logical function can be expressed as: *B* = *D* **and** *T*.

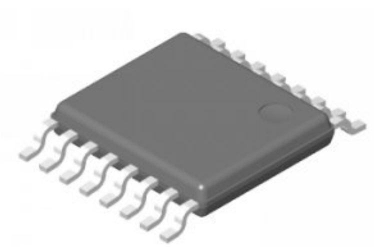
The bell will ring if both previously mentioned conditions are met, however in the case of emergency *M*, the bell can be controlled manually (*M* is true). To express the manual override, the logical function is modified as follows: *B* = *D* **and** *T* **or** *M*.

B *is* TRUE *if* (D **and** T) *are* TRUE **or** M *is* TRUE

Boolean algebra is a system using a set of numbers {0, 1} and a group of operations that consist of three basic operations (logical operations): AND, OR, NOT. By combining these three basic operations we can create various types of functions. The most used function is NAND (AND-NOT combination) and all other functions can be expressed by using this function.

Logical function can be described by an equation, a truth table, a diagram or a graph (timing relationship).

Digital circuits work with discreet (discontinuous) values of 0 and 1.



Logical circuit has *n* inputs and *m* outputs depending on the logical function. Inputs and outputs assume the logical values of 0 and 1.

Each state can be expressed in positive logic as follows:

Logic 1 – level H (high) – +5 V – TRUE – (ON) enabled state

Logic 0 – level L (low) – 0 V – FALSE – (OFF) off mode

Note: Negative logic has the high and low values reversed.

Types of logic circuits:

1. **Combinational** - output depends on the combination of input values
2. **Sequential** – output depends on a combination of input values and the time sequence of previous states.
   - **a. Synchronous** – the value appears at the output only at a certain moment
   - **b. Asynchronous** – the output changes immediately with the input

Logical conjunction

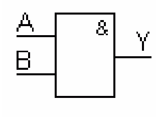
The logical conjunction can be defined for multiple input values. The AND-gate produces binary output of 1 when all of its inputs terminals are at the binary voltage level of 1 (high). Essentially this gate is the equivalent of switches connected in series.

Function: Y = A • B
Abbreviation: AND
Operator: • (dot)
Notation: A • B, AB, A∩B

Truth table:

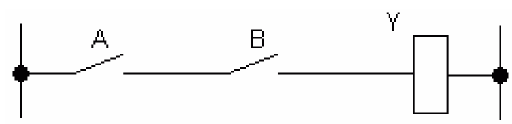
A	B	Y=A • B
0	0	0
0	1	0
1	0	0
1	1	1

Logical symbol:

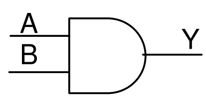


Logical AND circuits:

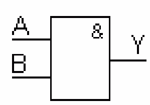
1. using switches



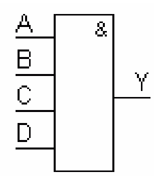
2. AND-gate (two and more inputs)



USA/old symbol



EU symbol



Logical disjunction

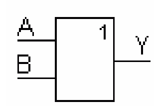
The logical disjunction can be defined for multiple input values. The OR-gate produces binary output of 1 when at least one of its input terminals is at the binary voltage level of 1 (high). Essentially this gate is the equivalent of parallel-connected switches.

Function: Y = A + B
Abbreviation: OR
Operator: +
Notation: A + B, A ∪ B

Truth table:

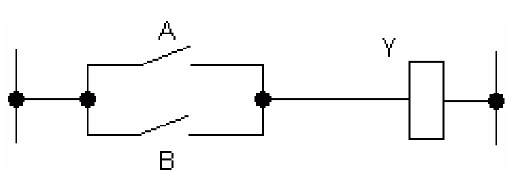
A	B	Y=A + B
0	0	0
0	1	1
1	0	1
1	1	1

Logical symbol:



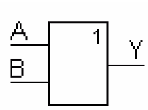
Logical OR circuits:

1. Using switches

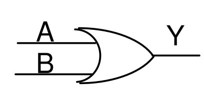


2. OR-gate

EU symbol



USA/old symbol



Logical negation

The inverter circuit also known as a NOT-circuit, produces a binary 1 output only when its input is at the binary 0 voltage level. This circuit is essentially a phase-reversing one-stage amplifier. Because of this inversion characteristic, the output voltage level is high when the input is low, and vice versa.

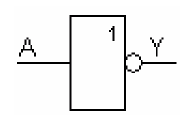
Function: $Y = \overline{A}$
Abbreviation: **N**OT

Notation $\overline{A}$ (read as *not A*)

Truth table:

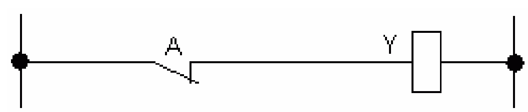
A	$Y = \overline{A}$
0	0
0	1

Logical symbol:



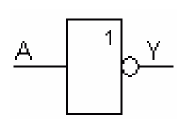
Logical NOT circuit:

1. Using switches

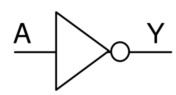


2. An inverter

EU symbol



USA/old symbol



VOCABULARY

AND-gate – obvod logického súčinu (hradlo)
OR-gate – obvod logického súčtu (hradlo)
NOT-gate – invertor
essentially – v podstate
recognize – rozpoznávať, rozlišovať
and vice versa – a naopak
in the case of emergency – v prípade núdze
override – potlačiť, nahradiť
sequential – postupný; tu: sekvenčný
equivalent – rovnocenný
phase-reversing one-stage amplifier – jedno-stupňový obracač fáze
truth table – pravdivostná tabuľka
accommodate –
logical state – logický stav
previously mentioned condition – vyššie uvedenej podmienky
depend – v závislosti od
abbreviation – skratka
notation – zápis, matematická symbolika