Positive and Negative Logic (B)

Gate Review and Positive and Negative Logic

The following is considered a review of

- **logic gates**,
- **truth tables**, and
- **positive logic**.

It is assumed that the student has been well grounded in the basic logic gate structures and uses, as well as the truth tables which describe their operations.

What is not normally known is that every **Positive Logic** gate has a **Negative Logic** twin.

- The Negative Logic gates operation is the same as that of Positive Logic gates, but the actual gate symbol looks different.
- You can't go to an electronics parts supplier and ask for a **Negative logic** gate. It doesn't exist as a part.
- The concept of **Negative Logic** is a state of mind or a way of looking at the gate which is different from the way you are used to looking at it.

Over the next several pages we will be discussing both the familiar **Positive Logic** gate and the associated **Negative Logic** gate. We will also discuss why this new way of looking at gates is desired.

All of the gates will be first shown using **Positive Logic**, i.e., their **inputs** will all be **active high**. It is quite often nice to be able to have **active low** inputs. Thus, we also can represent the same functions using **NEGATIVE LOGIC**.

We can easily convert **Positive** Logic symbols into **Negative** logic symbols by: **Invert each input and output to its opposite ACTIVE state**.

Note that in order to make the leap from **positive** to **negative** logic, all we do is apply **DeMorgan's Theorem.** This theorem is reviewed later but the basic equations that it is based on are:

$$f(A,B) = \overline{A \bullet B} = \overline{A} + \overline{B} \qquad \text{and} \qquad \overline{A + B} = \overline{A} \bullet \overline{B}$$

A	B	Pos Logic AND (AB)	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$	Neg Logic AND $\left(\overline{\overline{A} + \overline{B}}\right)$
0	0	0	1	1	1	0
0	1	0	1	0	1	0
1	0	0	0	1	1	0
1	1	1	0	0	0	1

$\Uparrow$ _________**Proven**_______ $\Uparrow$

What these equations suggest is that in order to get a gate which is equivalent to the **Positive Logic** gate but is based on **Negative Logic** instead, all you have to do is:

- **Inner Shape** of Gate
  - If the internal symbol was an OR gate, change it to an AND gate,
  - if it was an AND gate, change it to an OR gate.
  - (Leave the NOT shape alone.)
- **Outputs** of Gate
  - If the output was **Active-High**, change it to **Active-Low**
  - If the output was **Active-Low**, change it to **Active-High**
- **Inputs** of Gate
  - If the inputs are **Active-High**, change them to **Active-Low**
  - If the inputs are **Active-Low**, change them to **Active-High**

The Gates

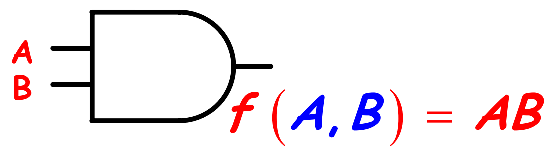
Positive Logic AND gate

The **Positive Logic AND** gate's output achieves its **ACTIVE** state **(HIGH)** when **"all of its inputs"** achieve their **ACTIVE** state **(HIGH).**

The paragraph above will be the standard statement for ALL of the gates. It will differ only in the sections in quotes and parenthesis. The **internal** AND **symbol** means "**all of its inputs**" while the **internal** OR **symbol** means "**one or more of its inputs**."

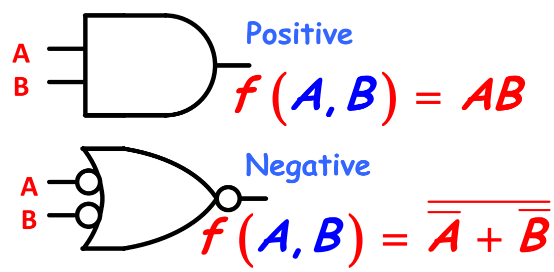
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

$$f(A,B) = AB$$
$$= \sum m(3)$$



Negative Logic AND gate

The **Negative Logic** symbol for the **AND gate** consists of the **OR internal shape** and **ACTIVE LOW** inputs and outputs.



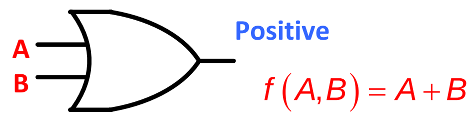
The **Negative Logic AND** gate's output achieves its **ACTIVE** state **(LOW)** when "**one or more of its inputs**" achieve their **ACTIVE** state **(LOW).**

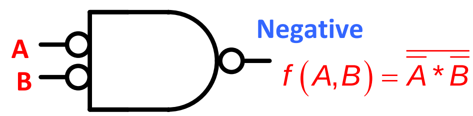
Positive Logic OR gate

The Positive Logic OR gates output achieves its ACTIVE state, (HIGH), when "one or more of its inputs" achieve their ACTIVE state, (HIGH).

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

$$f(A,B) = A + B$$
$$= \sum m(1,2,3)$$





Negative Logic OR gate

The Negative Logic OR gates output achieves its ACTIVE state, (LOW), when "all of its inputs" achieve their ACTIVE state, (LOW).

A	B	Pos Logic OR $(A+B)$	$\overline{A}$	$\overline{B}$	$\overline{A} \bullet \overline{B}$	Neg Logic OR $\left(\overline{\overline{A} \bullet \overline{B}}\right)$
0	0	0	1	1	1	0
0	1	1	1	0	0	1
1	0	1	0	1	0	1
1	1	1	0	0	0	1

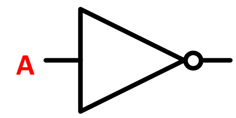
$\Uparrow$ _______Proven _______ $\Uparrow$

Positive Logic INVERTER (NOT) gate

The Positive Logic NOT gates output achieves its ACTIVE state (LOW) when "its input" achieves its ACTIVE state (HIGH).

A	$f(A)$
0	1
1	0

$$f(A) = \overline{A} = \sum m(0)$$

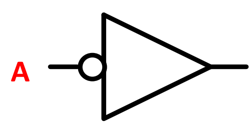


$$f(A) = \overline{A}$$

Negative Logic INVERTER (NOT) gate

In the case of the INVERTER, the basic shape doesn't change between Positive and Negative logic. The truth table would be obvious.

The Negative Logic NOT gates output achieves its ACTIVE state (HIGH) when "its input" achieves its ACTIVE state (LOW).



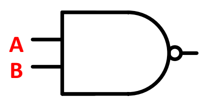
$$f(A) = \overline{A}$$

Positive Logic NAND gate

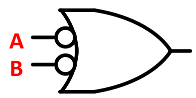
The Positive Logic NAND gates output achieves its ACTIVE state (LOW)
"ONLY when BOTH of its inputs" achieve their ACTIVE states, (HIGH).

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

$$f(A,B) = \overline{AB} = \sum m(0,1,2)$$



Positive $f(A,B) = \overline{AB}$



Negative $f(A,B) = \overline{A} + \overline{B}$

Negative Logic NAND gate

The Negative Logic NAND gates output achieves its ACTIVE state (HIGH)
"when at least one of its inputs" achieve their ACTIVE states, (LOW).

A	B	Pos Logic NAND $\overline{A * B}$	$\overline{A}$	$\overline{B}$	Neg Logic NAND $\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

⇑ __Proven___ ⇑

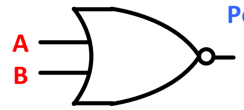
Positive Logic NOR gate

The Positive Logic NOR gates output achieves its ACTIVE state (LOW) "only when AT LEAST ONE of its inputs" achieve their ACTIVE state (HIGH).

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

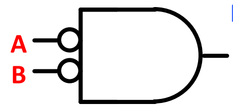
$$f(A,B) = \overline{A+B}$$
$$= \sum m(0)$$

Positive



$$f(A,B) = \overline{A+B}$$

Negative



$$f(A,B) = \overline{A}\ \overline{B}$$

Negative Logic NOR gate

The Negative Logic NOR gates output achieves its ACTIVE state (HIGH) "only when ALL of its inputs" achieve their ACTIVE state (LOW).

A	B	Pos Logic NOR $(\overline{A+B})$	$\overline{A}$	$\overline{B}$	Neg Logic NOR $(\overline{A}*\overline{B})$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

⇑ ______Proven _____ ⇑

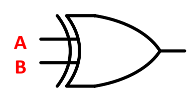
The XOR gate

The XOR gates output achieves its ACTIVE state (HIGH) "only when an ODD number" of its inputs are in their ACTIVE states (HIGH).

There isn't a Negative Logic version of this gate.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$$f(A,B) = A \oplus B = \sum m(1,2)$$



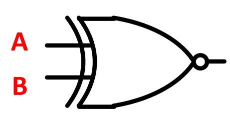
$f\left(A,B\right) = A \oplus B$

The XNOR gate

The XNOR gates output achieves its ACTIVE state (LOW) "ONLY when an ODD number" of its inputs are in their ACTIVE states (HIGH).

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

$$f(A,B) = \overline{A \oplus B} = \sum m(0,3)$$



$f\left(A,B\right) = A \odot B$

There isn't a Negative Logic version of this gate.