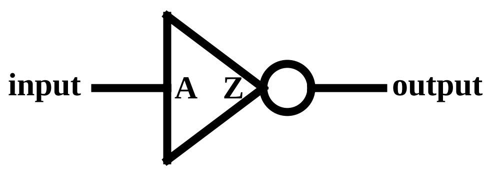
# The fundamental logic gate family

The seven fundamental logic gates are:

1. NOT (Inverter)
2. AND
3. NAND
4. OR
5. NOR
6. XOR
7. XNOR

**NOT**:  We have looked at the inverter before. Its symbol and truth table are repeated here. Note that the inverter has only one input.

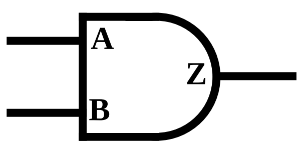



**Inverter or NOT gate**

| input | output |
|---|---|
| 0 | 1 |
| 1 | 0 |

$z = \overline{a}$

**AND**:  The AND gate may have any number of inputs and one output. For the output of a AND gate to be logic one, all inputs must be at logic one. If any input is logic zero, the output is logic zero. The symbol and truth table for a 2 input AND gate are given below:




**AND Gate**

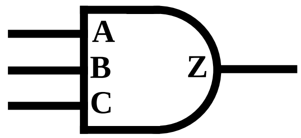
| inputs | | output |
|---|---|---|
| A | B | Z |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$z = ab$

**“all ones makes a one”**

In equation form, two variables beside each other mean AND. “A and B” is written AB. You can think as AND as the binary version of multiplication.

If the AND gate was a three input type it would have the following schematic symbol and truth table.

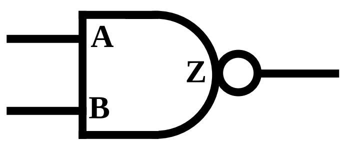



**"all ones makes a one"**

| inputs A | B | C | output Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**z = abc**

**NAND**: The NAND gate may have any number of inputs and one output. For the output of a NAND gate to be logic zero, all inputs must be at logic one. If any input is logic zero, the output is logic one. The symbol and truth table for a 2 input nand gate are given below:



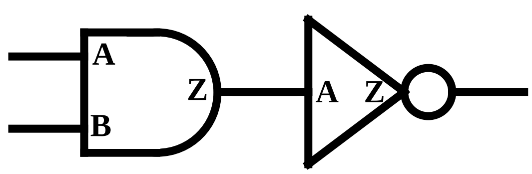

**NAND Gate**

| inputs A | B | output Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**z = $\overline{ab}$**

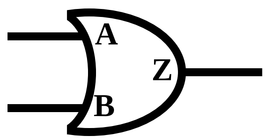
**"any zero makes a one"**

The overline above the ab indicates an inversion of the expression below it. Another way to implement this gate would be to follow the AND gate with a NOT gate. The resultant output would be the same. This is shown below.




**Another way to build a NAND gate**

**OR**:  The OR gate may have any number of inputs and one output. For the output of a OR gate to be logic one, at least one input must be at logic one. If all inputs are logic zero, the output is logic zero. The symbol and truth table for a 2 input OR gate are given below:



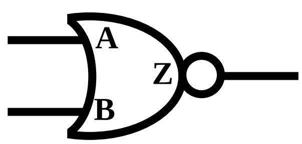

**OR Gate**

| inputs | | output |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**"any one makes a one"**

$z = a + b$

**NOR**:  The NOR gate may have any number of inputs and one output. For the output of a NOR gate to be logic zero, at least one input must be at logic one. If all inputs are logic zero, the output is logic one. The symbol and truth table for a 2 input NOR gate are given below:



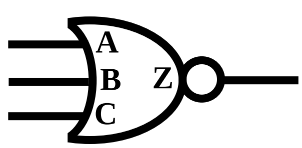

**NOR Gate**

| inputs | | output |
|---|---|---|
| **A** | **B** | **Z** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**"any one makes a zero"**

$z = \overline{a + b}$

Like the AND, NAND, and OR gates, the NOR gate may have as many as 5 inputs depending upon the technology used for implementation. The schematic symbol and truth table for a three input NOR gate is shown below.

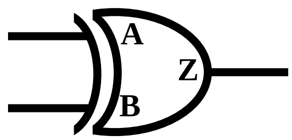



**NOR Gate**

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

| inputs | | | output |
|---|---|---|---|
| **A** | **B** | **C** | **Z** |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**XOR**:  The XOR gate may have any number of inputs and one output. Common implementations have only two or sometimes three inputs. For the output of a XOR gate to be logic one, there must be an odd number of inputs at logic one. Otherwise, the output is logic zero. The symbol and truth table for a 2 input XOR gate are given below:



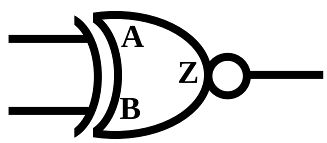

**XOR Gate**

| inputs | | output |
|---|---|---|
| **A** | **B** | **Z** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

$z = a \oplus b$

**“either or but not both makes a one”**

**XNOR**:  The “xnor” gate may have any number of inputs and one output. Common implementations have only two or sometimes three inputs. For the output of a “nxor” gate to be logic zero, there must be an odd number of inputs at logic one. Otherwise, the output is logic one. The symbol and truth table for a 2 input XNOR gate are given below:



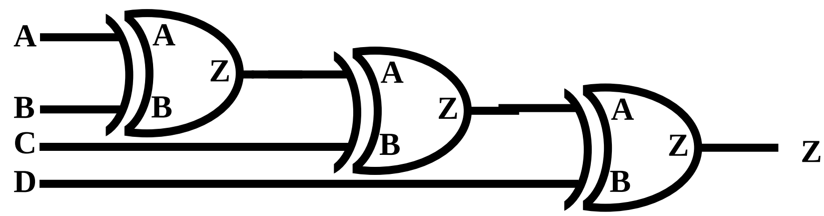

**XNOR Gate**

| inputs | | output |
|---|---|---|
| **A** | **B** | **Z** |
| **0** | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |
| **1** | **1** | **1** |

$z = \overline{a \oplus b}$

**“either or but not both makes a zero”**

XOR or XNOR gates of more than three inputs are rarely seen. They can be implemented however by simply cascading multiple levels of XOR or XNOR gates together.  For example, a four input XOR gate could be implemented as follows:




**Four input XOR Gate**

The XOR and XNOR gates are very useful. With them we can implement adders and parity detector.  For example the four input XOR gate above outputs a “1” whenever an odd number of “1’s” are input to the gate.