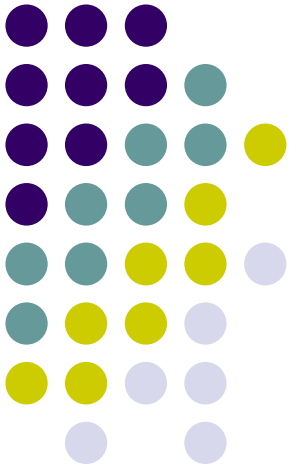
Frantz Fanon University

Break chains, visible or invisible -
build up from ruins, personal or social.

Digital Logic Design



FRANTZ FANON UNIVERSITY

COLLAGE OF COMPUTING AND IT

DEPARTMENT OF ICT

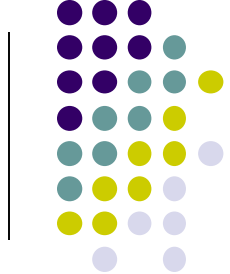
Suleiman Gargaare

MSc in Computer Science, MSc in Project Management & BSc in Information Technology.

Lecturer

suleiman.gargaare@gmail.com

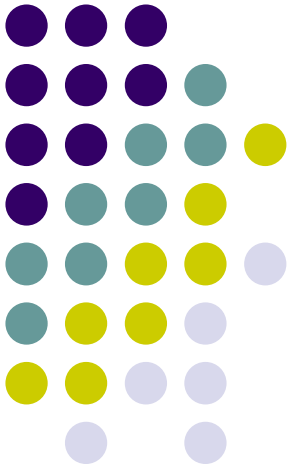
Course Contents



- Chapter 1 - Number Systems
- Chapter 2 - Binary Arithmetic
- **Chapter 3 - Logic Gates**
- Chapter 4 - Boolean Algebra
- Chapter 5 - K-Map

Digital Logic Design

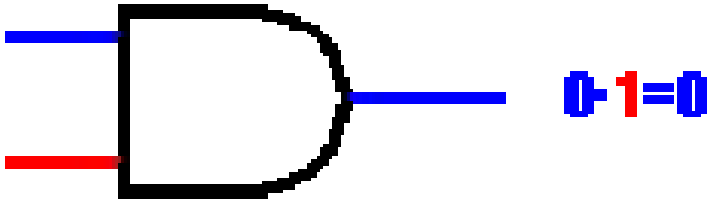
Chapter 3 – Logic Gates



Logic Gates & Functions

- When we deal with logical circuits (as in computers), we not only need to deal with logical functions; we also need some special symbols to denote these functions in a logical diagram.
- There are three fundamental logical operations, from which all other functions, no matter how complex, can be derived. These functions are named **AND**, **OR**, and **NOT**. Each of these has a specific symbol and a clearly-defined behavior, as follows: -

The AND Gate



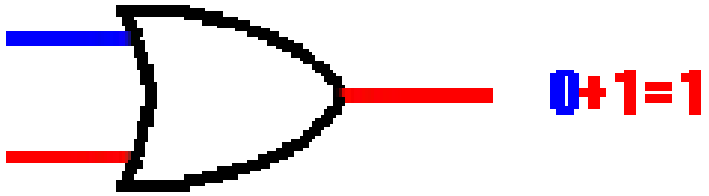
- The AND gate implements the **AND** function. With the gate shown above, **both inputs must have logic 1** signals applied to them in order for **the output to be logic 1**. With either input at logic 0, the output will be held to logic 0. There is no limit to the number of inputs that may be applied to an AND function.

Cont..

- The truth table for a two-input AND gate looks like: -

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

The OR Gate



- The OR gate is sort of the **reverse of the AND gate**. The OR function, like its verbal counterpart, allows **the output to be true (logic 1)** if any one or more of its inputs are true. Verbally, we might say, "If it is raining OR if I turn on the sprinkler, the grass will be wet." Note that the grass will still be wet if the sprinkler is on and it is also raining.

Cont..

- The truth table for a two-input OR gate looks like: -

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

The NOT Gate or Inverter

- The inverter is a little different from AND and OR gates in that it always has exactly one input as well as one output. Whatever logical state is applied to the input, the opposite state will appear at the output. The NOT function is necessary in many applications and highly useful in others. A practical verbal application might be: The door is NOT locked = you may enter

Cont..

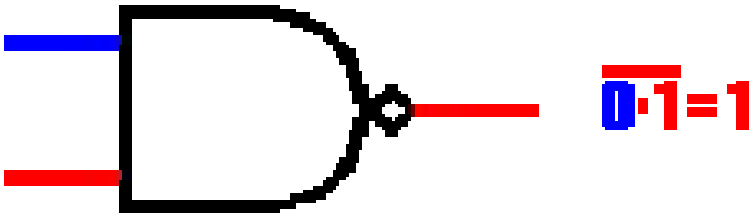
- The truth table for the NOT gate is shown below: -

A	$\bar{A}$
0	1
1	0

Derived Logic Functions and Gates

- The Some combinations of basic functions have been given names and logic symbols of their own. The first is called **NAND**, and consists of an **AND function followed by a NOT function**. The second, as you might expect, is called **NOR**. This is an **OR function followed by NOT**. The third is a **variation of the OR function**, called the **Exclusive-OR**, or **XOR** function. Each of these derived functions has a specific logic symbol and behavior, which we can summarize as follows:

The NAND Gate



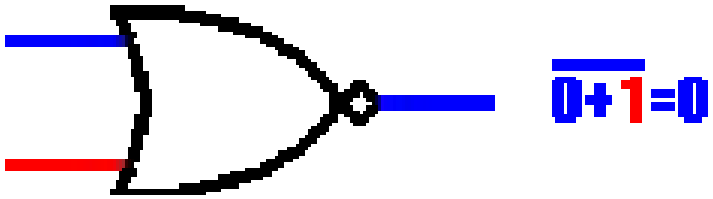
- The **NAND** gate implements the NAND function, which is exactly **inverted from the AND function**. *Both inputs must have logic 1 signals applied to them in order for the output to be logic 0*. With either input at logic 0, the output will be held to logic 1. The circle at the output of the NAND gate denotes the logical inversion, just as it did at the output of the inverter.

Cont..

- The truth table for a two-input NAND gate looks like: -

A	B	A.B
0	0	1
0	1	1
1	0	1
1	1	0

The NOR Gate



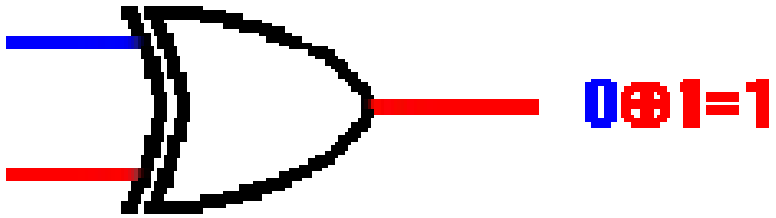
- The **NOR** gate is an **OR gate with the output inverted**. Where the OR gate allows the output to be true (logic 1) *if any one or more of its inputs are true, the NOR gate inverts this and forces the output to logic 0 when any input is true*.
- In symbols, the NOR function is designated with a plus sign (+), with an over-bar over the entire expression to indicate the inversion. This is an OR gate with a circle to designate the inversion.

Cont..

- The truth table for a two input NOR gate looks like: -

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

The Exclusive-OR, or XOR Gate



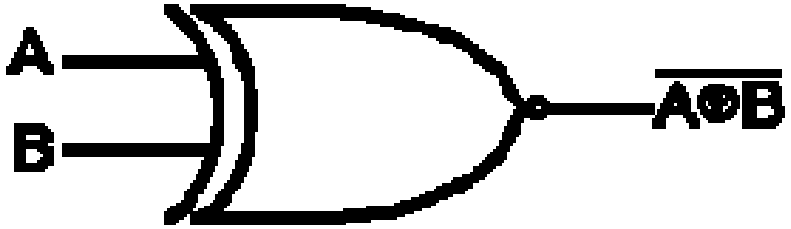
- The **Exclusive-OR**, can be stated as, "**Either A or B, but not both**." The XOR gate **produces logic 1 output only if its two inputs are different**. *If the inputs are the same, the output is logic 0*. The XOR symbol is a variation on the standard OR symbol. It consists of a plus (+) sign with a circle around it.

Cont..

- The truth table for a two-input XOR gate looks like: -

A	B	AOB
0	0	0
0	1	1
1	0	1
1	1	0

The eXclusive-NOR or XNOR Gate

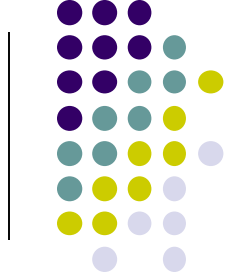


- The **exclusive-NOR** or XNOR gate has two or more inputs. **The output is equivalent to inverting the output from the exclusive-OR gate described previous slide**. Therefore an equivalent circuit would comprise an XOR gate, the output of which feeds into the input of a NOT gate. In general, an XNOR gate gives an output value of 1 when there are an even number of 1's on the inputs to the gate.

Cont..

- The output from the XNOR gate is written as AOB which reads "A XNOR B".
- The truth table for a two-input XNOR gate looks like: -

A	B	AOB
0	0	1
0	1	0
1	0	0
1	1	1



Thank You