LOGIC GATES

COMP370

Introduction to Computer Architecture

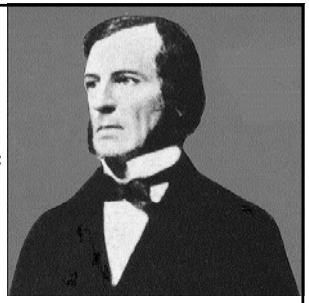
Introduction

- Basic Logic Gates
- Truth Tables
- Logical Functions
  - Truth Tables
  - Logical Expression
  - Graphical Form

Most Difficult Reading Topics

- Logic gates and figuring out how to read them
- Logical Circuit Equivalence
- NAND NOR and XOR truth tables
- Using the rules to create and read the logic gates using 0's and 1's
- Transistor implementation
- Difference between positive logic and negative logic

Logic

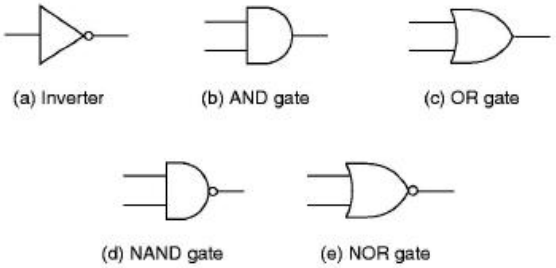


- Formal logic is a branch of mathematics that deals with true and false values instead of numbers.
- In the mid-19th century, George Bool developed many Logic ideas.
- Boolean logic deals with equations where the operators are “AND” or “OR” instead of “add” and “multiply”.

Electric Logic

- Logical values can easily be expressed by an electrical circuit.
- "True" or "1" can be defined as voltage on a wire while "False" or "0" can be defined as no voltage. We will use positive logic.
- Analog values can be anything while digital only has discrete values, 0 or 1
- Electrical devices called "gates" can implement the logical

Logic Gates & Symbols



Note that gates can have more than 2 inputs.

AND gate

- The AND gate is an electronic circuit that gives a **true** output (1) only if **all** its inputs are true. A dot (·) is used to show the AND operation i.e. A·B. Note that the dot is sometimes omitted i.e. AB

2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Inputs and Outputs

- The inputs can be considered as logical (true or false) values.
- You can also think of the inputs as voltage (true) or ground (false).
- The output of the gate will also be voltage (true) or ground (false).

inputs 

output

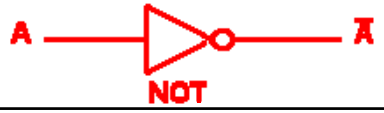
OR gate

- The OR gate is an electronic circuit that gives a true output (1) if **one or more** of its inputs are true. A plus (+) is used to show the OR operation.

2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate

- The **NOT** gate is an electronic circuit that produces an inverted version of the input at its output.
- It is also known as an *inverter*.
- If the input variable is A, the inverted output is known as NOT A.
- This is also shown as A', or **Ā** with a bar over the top



A	A’
0	1
1	0

NAND gate

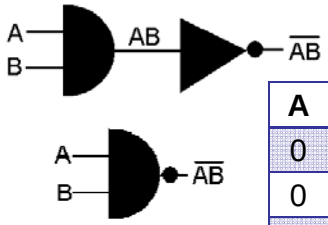
- This is a **NOT-AND** gate which is equal to an **AND** gate followed by a **NOT** gate.
- The outputs of all **NAND** gates are true if **any** of the inputs are false.
- The symbol is an **AND** gate with a small circle on the output. The small circle represents inversion.

NAND gate Cont.

2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Logical Equivalence

- A NAND gate gives the same results as an AND gate feeding into a NOT gate.
- Circuits are logically equivalent if they produce the same truth table output.

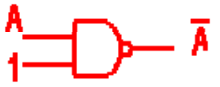


A	B	AB	NOT AB	NAND AB
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

NAND as a NOT

- Below are two ways that a **NAND** gate can be configured to produce a **NOT** gate. They are logically equivalent.
- It can also be done using NOR logic gates in the same way.

A	A	NAND AA
0	0	1
1	1	0



A	1	NAND A'
0	1	1
1	1	0

NOR gate

- This is a **NOT-OR** gate which is equal to an **OR** gate followed by a **NOT** gate.
- The outputs of all **NOR** gates are **false** if **any** of the inputs are **true**.
- The symbol is an **OR** gate with a small circle on the output. The small circle represents inversion.

NOR gate cont.

2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

EXOR gate

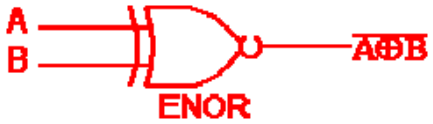
- The '**Exclusive-OR**' gate is a circuit which will give a true output if **either, but not both**, of its two inputs are true.
- An encircled plus sign ( ) is used to show the EOR operation.

2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

EXNOR gate

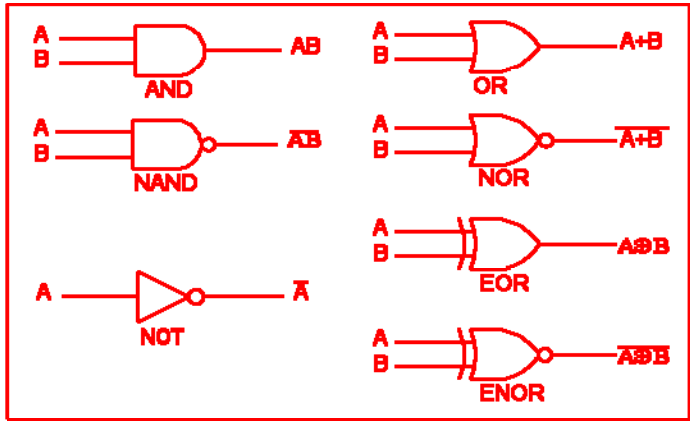
- The '**Exclusive-NOR**' gate circuit does the opposite to the **EOR** gate.
- It will give a false output if **either, but not both**, of its two inputs are true.
- The symbol is an **EXOR** gate with a small circle on the output.
- The small circle represents inversion.

EXNOR gate cont.



2 Input EXNOR gate		
A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Logic gate symbols



Truth Table

- Logic gates representation using the Truth table

NOT gate	
A	$\overline{A}$
0	1
1	0

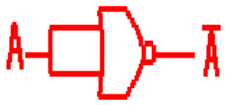
INPUTS		OUTPUTS					
A	B	AND	NAND	OR	NOR	EXOR	EXNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

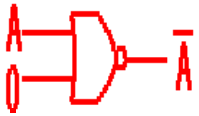
Try it

- Draw the circuit diagrams like the ones below to show how a **NOR** gate can be made into a **NOT** gate.

NOR Inverter

- Wiring the **NOR** gate to become an inverter





Three Input AND Gate

- Here is an example of a three input **AND** gate.
- Notice that the truth table for the three input gate is similar to the truth table for the two input gate.
- It works on the same principle, this time all three inputs need to be true (1) to get a true output.

Three Input AND Gate Cont.

3 Input AND gate			
A	B	C	A.B.C
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

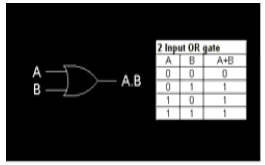
Four Input AND Gate

- Here is an example of a four input **AND** gate.
- It also works on the same principle, all four inputs need to be true (1) to get a true output.
- The same principles apply to 5, 6,…, n input gates.

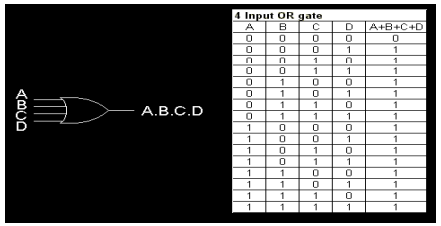
Four Input AND Gate cont.

4 Input AND gate				
A	B	C	D	A.B.C.D
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

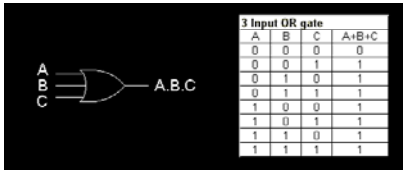
Two, Three, and Four OR gates



2 Input OR gate			
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



4 Input OR gate				
A	B	C	D	A+B+C+D
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



3 Input OR gate			
A	B	C	A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Connecting Gates

- The output of one logic gate can be used as the input to another logic gate.
- The output of one gate can be used as the inputs to several other gates.