NAND and NOR

As a universal gates

This presentation is about NAND Logic in the sense of building other logic gates using just NAND and NOR gates.

What are a Universal Gate And why NAND and NOR are known as universal gates?

- A gate which can be use to create any Logic gate is called Universal Gate

- NAND and NOR are called Universal Gates because all the other gates can be created by using these gates

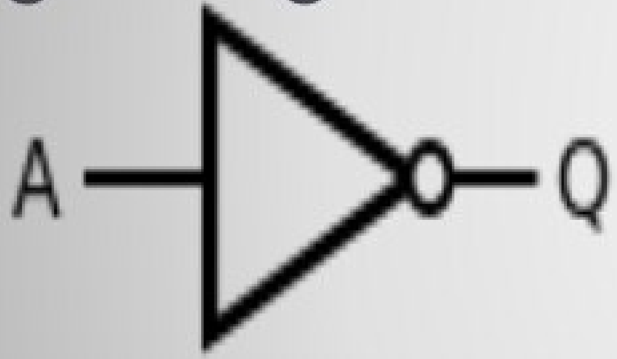
Proof for NAND gates

- Any Boolean function
  can be implemented using AND, OR and NOT
  gates

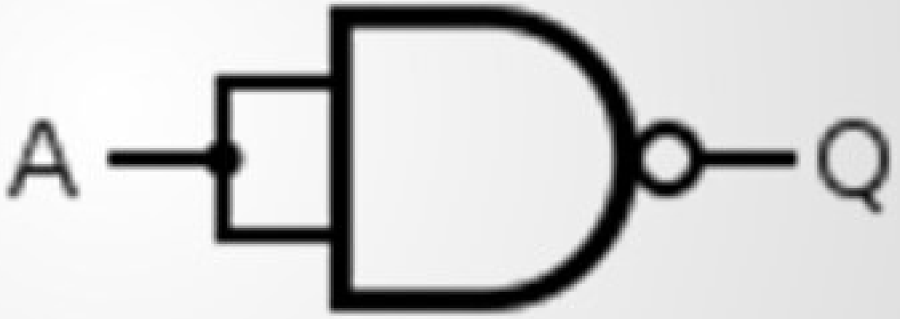
- In the same way AND, OR and NOT gates can be implemented using NAND gates only,

Implementation of NOT using NAND

A NOT gate is made by joining the inputs of a NAND gate together.



Desired NOT Gate

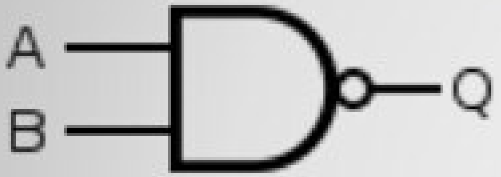


NAND Construction

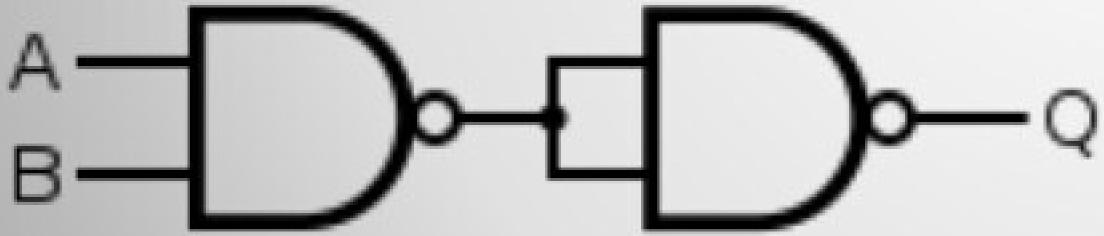
Input	Output
0	1
1	0

Implementation of AND using NAND

- A NAND gate is an inverted AND gate.



- An AND gate is made by following a NAND gate with a NOT gate

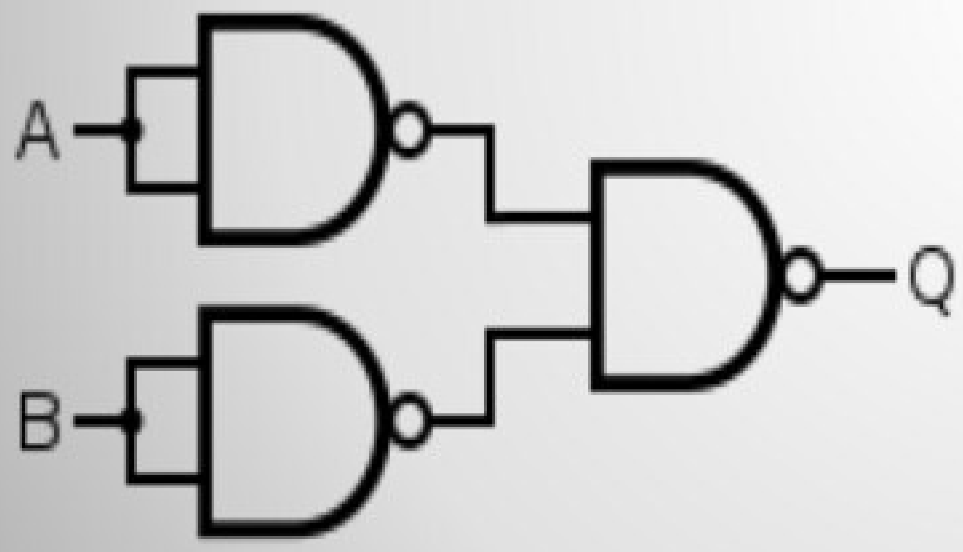


NAND Construction

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

Implementation of OR gate using NAND

- If the truth table for a NAND gate is examined or by applying De Morgan's Laws, it can be seen that if any of the inputs are 0, then the output will be 1. To be an OR gate,

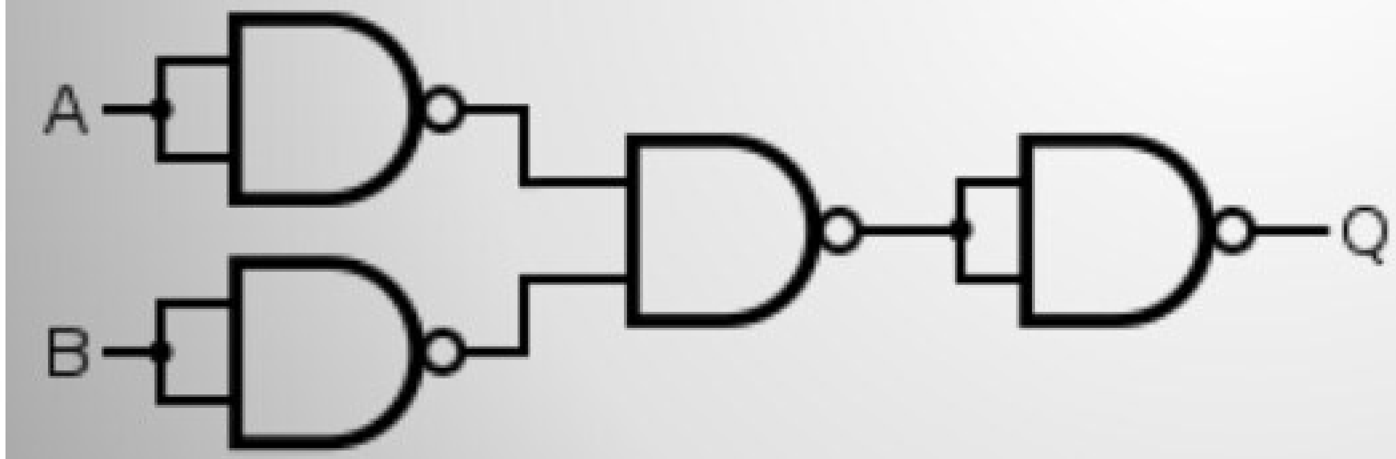


NAND Construction

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

Implementation of NOR gate using NAND

- A NOR gate is simply an inverted OR gate. Output is high when neither input A nor input B is high:

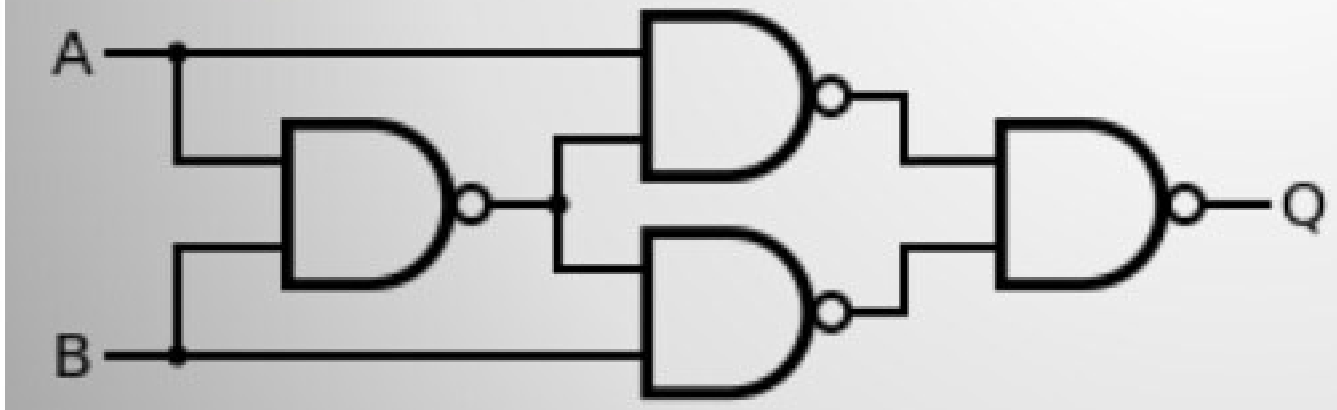


NAND Construction

Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	0

Implementation of XOR gate using NAND

- An XOR gate is constructed similarly to an OR gate, except with an additional NAND gate inserted such that if both inputs are high, the inputs to the final NAND gate will also be high,

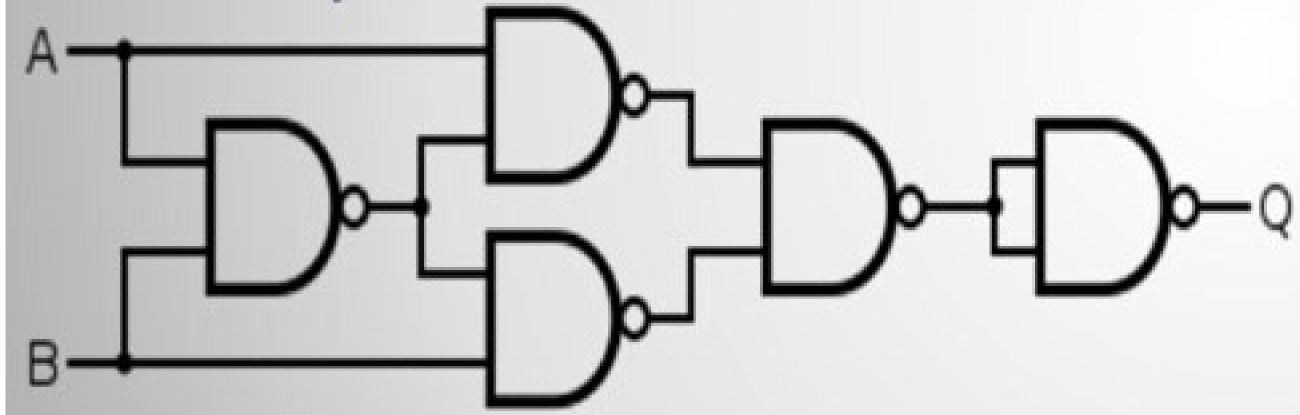


NAND Construction

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

Implementation of XNOR gate using NAND

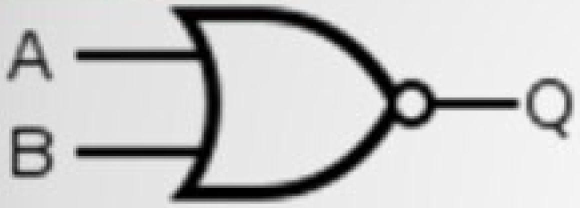
- An XNOR gate is simply an XOR gate with an inverted output:



NAND Construction

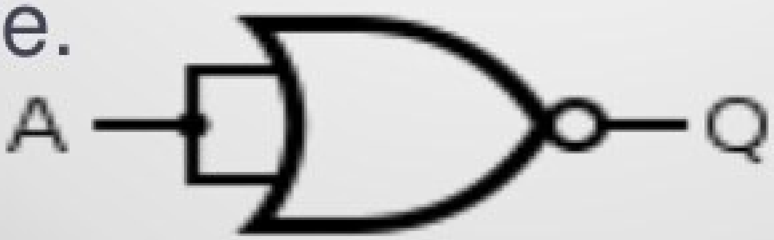
Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	1

Proof for NOR gates

- Like NAND gates, NOR gates are so-called "universal gates" that can be combined to form any other kind of logic gate. A NOR gate is logically an inverted OR gate 

Implementation of NOT gate using NAND

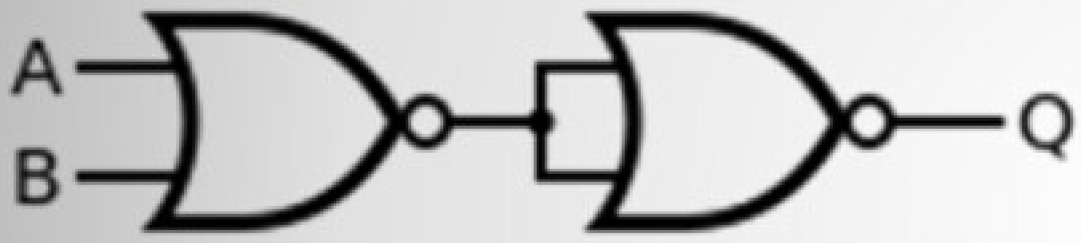
- NOT made by joining the inputs
- of a NOR gate.

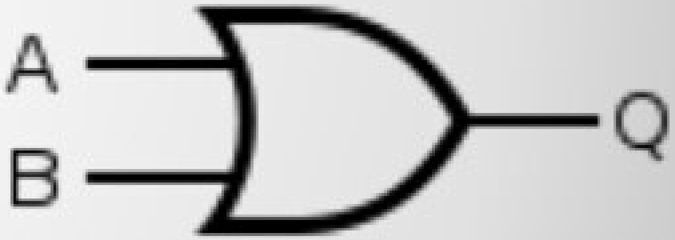


Input	Output
0	1
1	0

Implementation of OR gate using NOR

- The OR gate is simply a NOR gate followed by another NOR gate



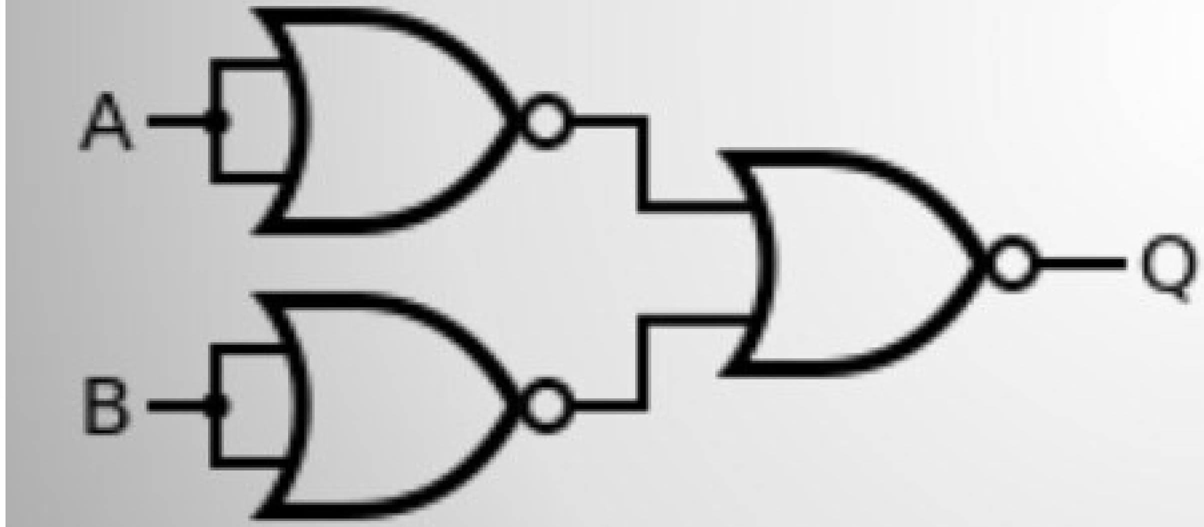


Desired Gate

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

Implementation of AND gate using NOR

- an AND gate is made by inverting the inputs to a NOR gate.

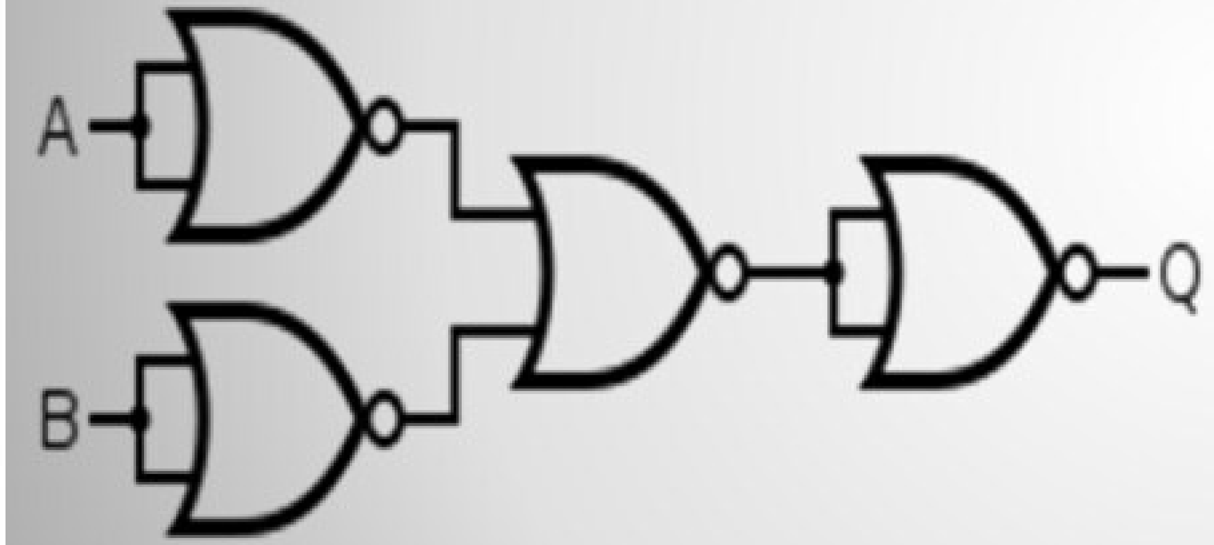


NOR Construction

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

Implementation of NAND gate using NOR

- A NAND gate is made using an AND gate in series with a NOT gate:

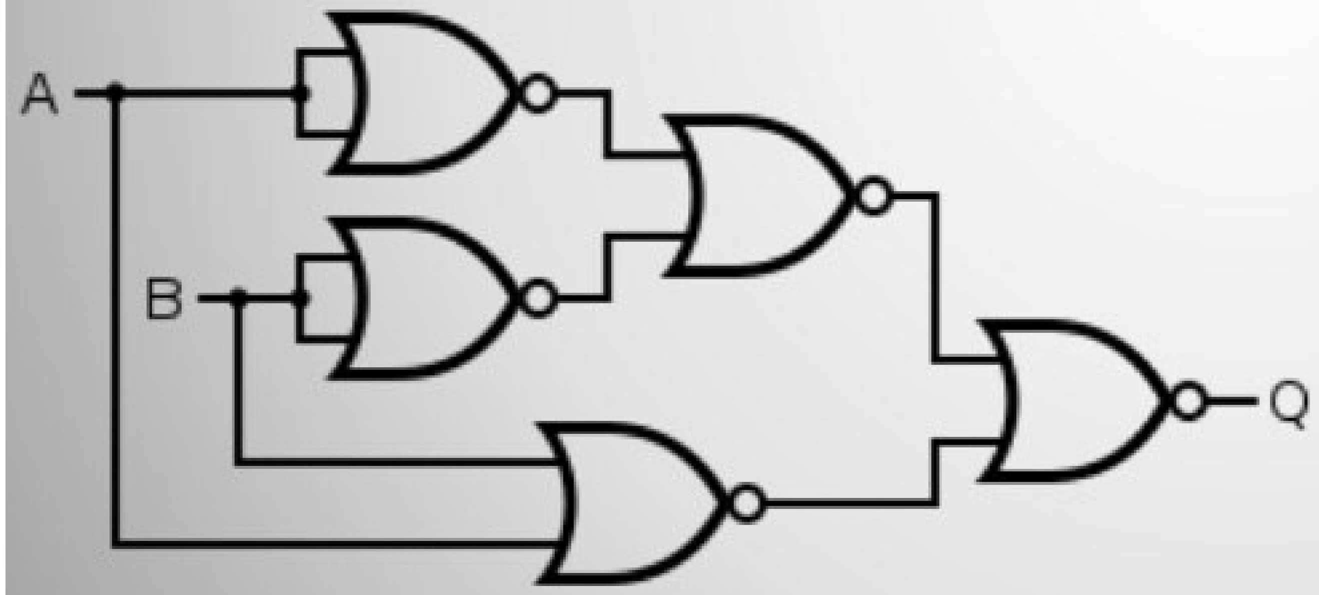


NOR Construction

Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

Implementation of XOR gate using NOR

- An XOR gate is made by connecting the output of 3 NOR gates (connected as an AND gate) and the output of a NOR gate to the respective inputs of a NOR gate.

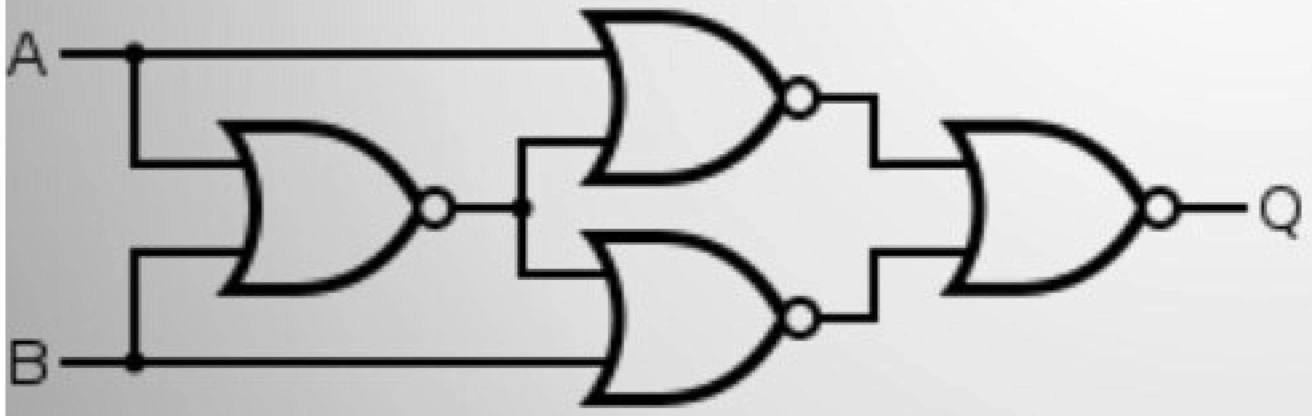


NOR Construction

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

Implementation of XNOR gate using NOR

- An XNOR gate can be constructed from four NOR gates implementing the expression "(A NOR N) NOR (B NOR N) where N = A NOR B". This construction has a propagation delay three times that of a single NOR gate, and uses more gates.



NOR Construction

Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	1

The End!!

Any Question?