Jiangxi University of Science and Technology

DIGITAL DESIGN

Lecture 6: Logic gate & Universal GATE

Logic Gates

- **AND gate(.) –** The AND gate gives an output of 1 if both the two inputs are 1, it gives 0 otherwise.
- **OR gate(+) –** The OR gate gives an output of 1 if either of the two inputs are 1, it gives 0 otherwise.
- **NOT gate(‘) –** The NOT gate gives an output of 1 input is 0 and vice-versa.
- **XOR gate() –** The XOR gate gives an output of 1 if either both inputs are different, it gives 0 if they are same.

Three more logic gates are obtained if the output of above-mentioned gates is negated.

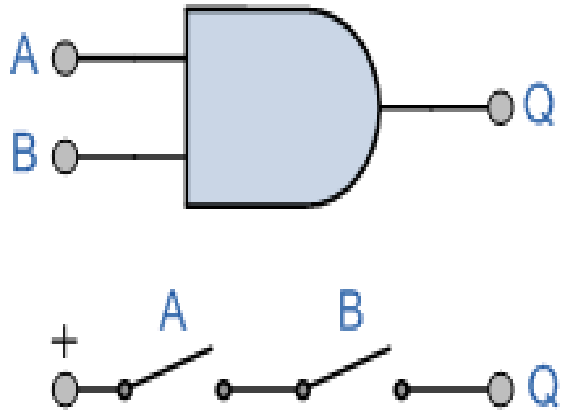
NAND gate()- The NAND gate (negated AND) gives an output of 0 if both inputs are 1, it gives 1 otherwise.

•**NOR gate()-** The NOR gate (negated OR) gives an output of 1 if both inputs are 0, it gives 1 otherwise.

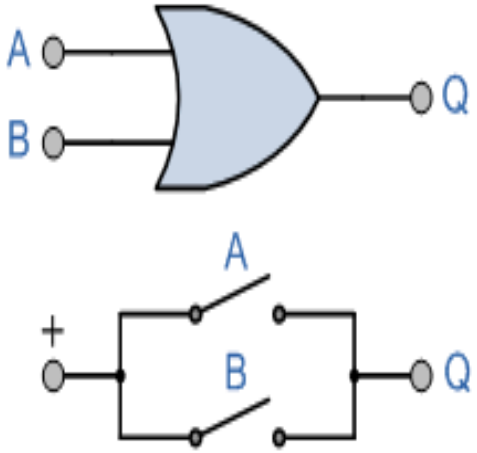
XNOR gate()- The XNOR gate (negated XOR) gives an output of 1 both inputs are same and 0 if both are different.

Every Logic gate has a graphical representation or symbol associated with it. Below is an image which shows the graphical symbols and truth tables associated with each logic gate.

Logic Gates

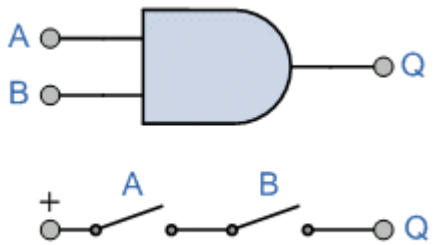
Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1

The 2-input Logic AND Gate

Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1

The 2-input Logic OR Gate

Logic Gates



Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y F: 0 0 0; 0 1 0; 1 0 0; 1 1 1
OR		$F = x + y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 1
Inverter		$F = x'$	x F: 0 1; 1 0
Buffer		$F = x$	x F: 0 0; 1 1
NAND		$F = (xy)'$	x y F: 0 0 1; 0 1 1; 1 0 1; 1 1 0
NOR		$F = (x + y)'$	x y F: 0 0 1; 0 1 0; 1 0 0; 1 1 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y F: 0 0 1; 0 1 0; 1 0 0; 1 1 1

Logic Gates

YES

INPUT	OUTPUT
A	
0	0
1	1

NOT

INPUT	OUTPUT
A	
0	1
1	0

AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

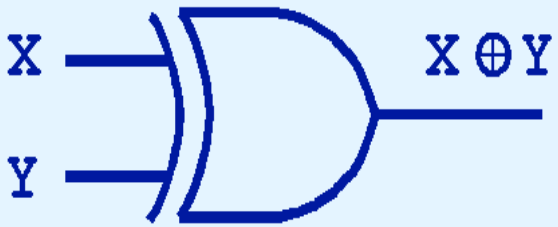
INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

Logic Gates

- Another very useful gate is the exclusive OR (XOR) gate.
- The output of the XOR operation is true only when the values of the inputs differ.

X XOR Y

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0



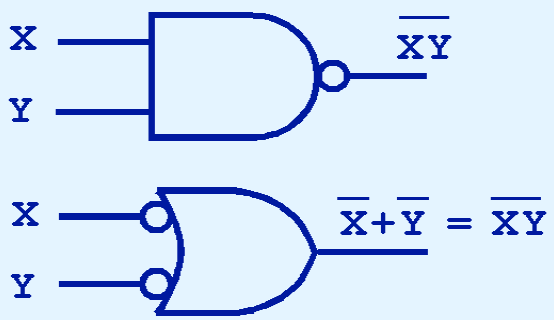
Note the special symbol ⊕ for the XOR operation.

Logic Gates

- NAND and NOR are two very important gates. Their symbols and truth tables are shown at the right.

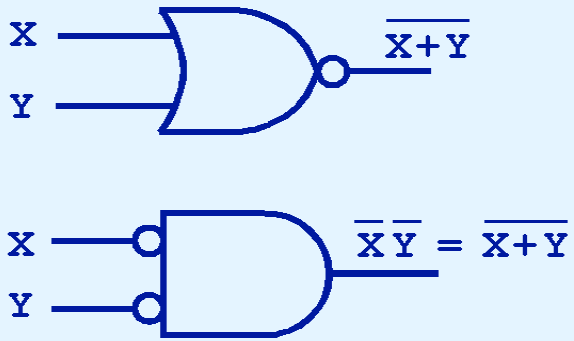
X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



X NOR Y

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



Summary Exclusive-OR Logic

The truth table for an exclusive-OR gate is

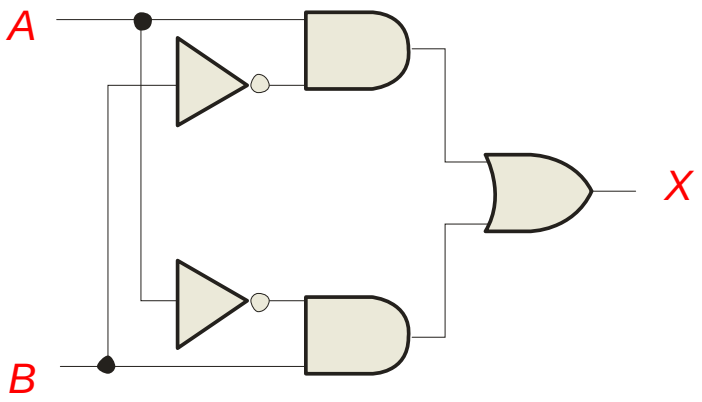
Inputs		Output
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0

Notice that the output is HIGH whenever *A* and *B* <u>disagree</u>.

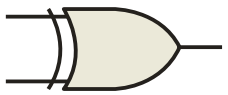
The Boolean expression is

$$X = A\overline{B} + \overline{A}B$$

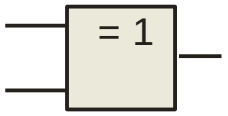
The circuit can be drawn as



Symbols:



Distinctive shape outline



Rectangular

Summary Exclusive-NOR Logic

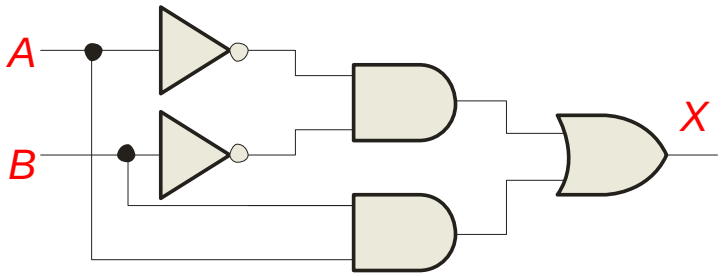
The truth table for an exclusive-NOR gate is

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

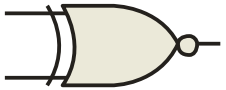
Notice that the output is HIGH whenever *A* and *B* <u>agree</u>.

The Boolean expression is $X = A\overline{B} \overline{+} AB$

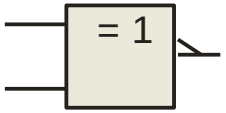
The circuit can be drawn as



Symbols:



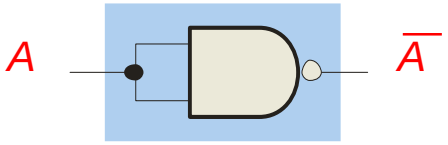
Distinctive shape outline



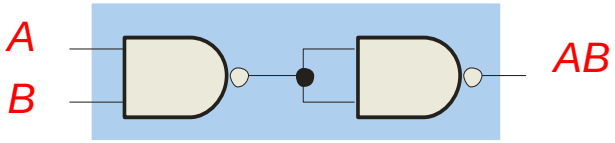
Rectangular

Summary Universal Gates

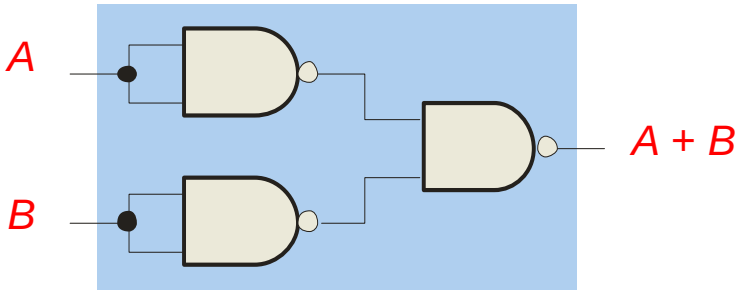
NAND gates are sometimes called **universal** gates because they can be used to produce the other basic Boolean functions.



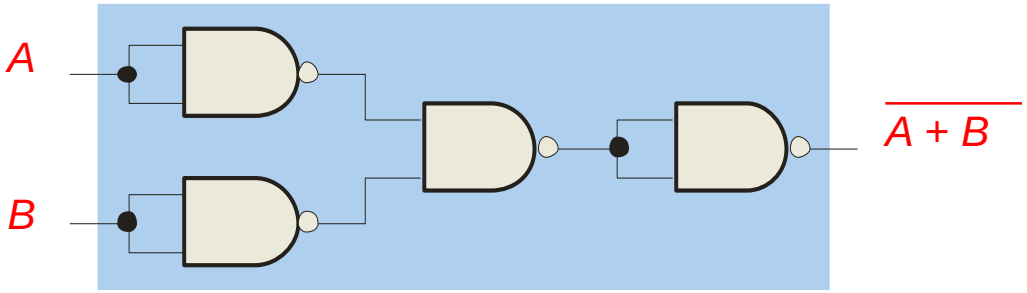
Inverter



AND gate



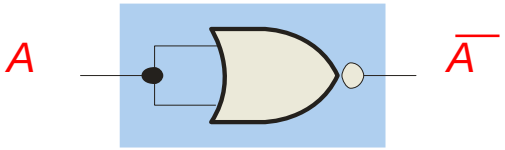
OR gate



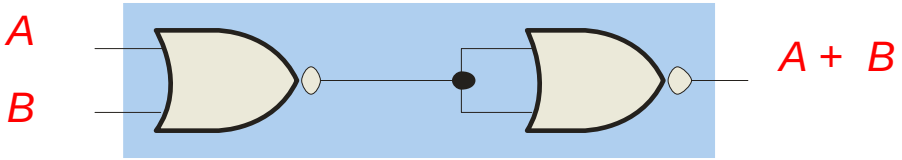
NOR gate

Summary Universal Gates

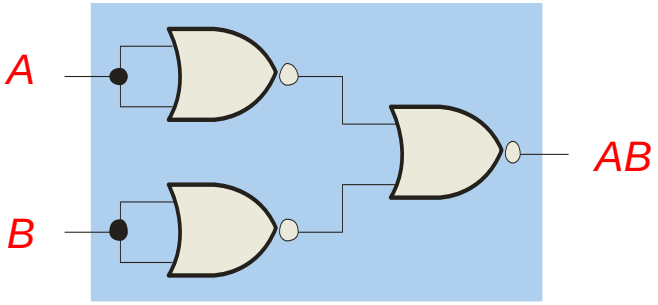
NOR gates are also **universal** gates and can form all of the basic gates.



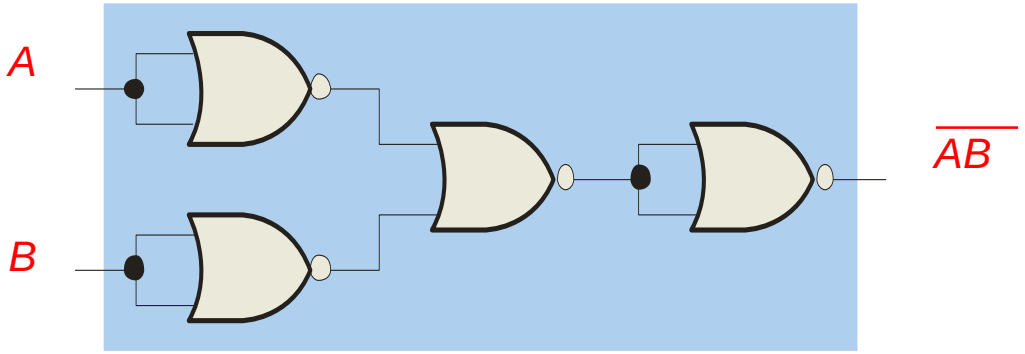
Inverter



OR gate

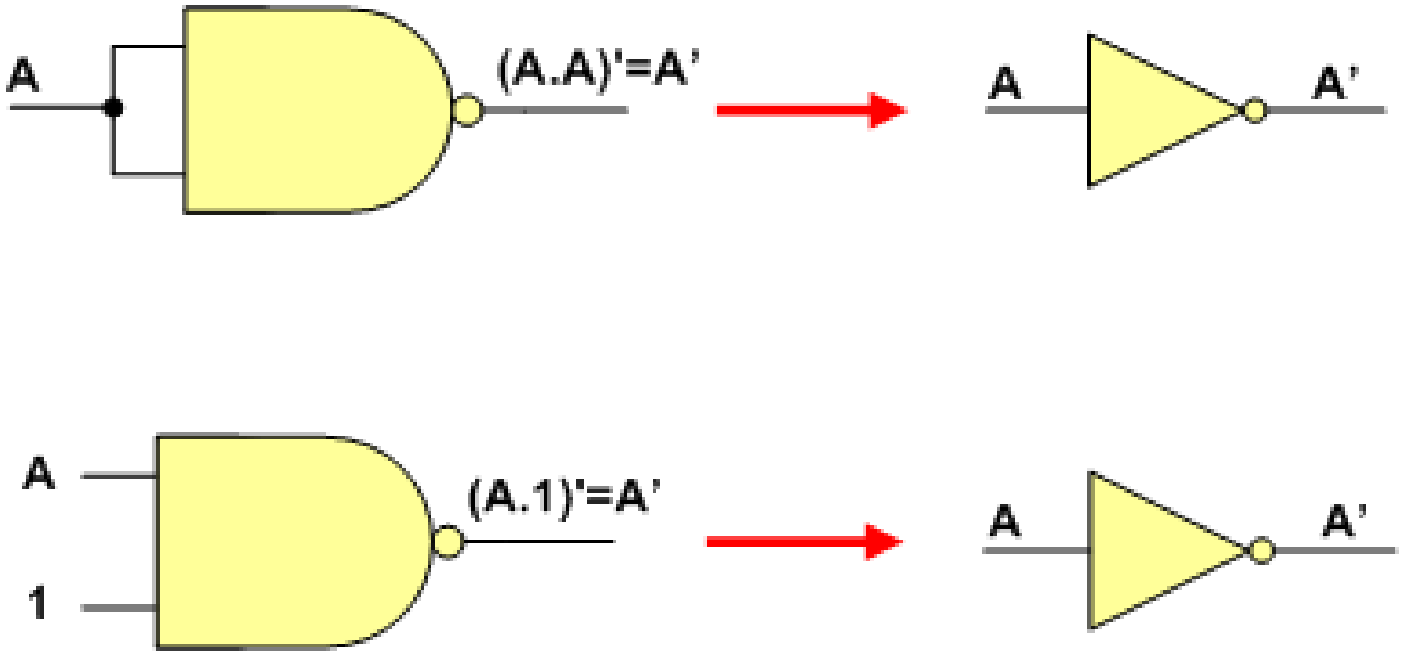


AND gate

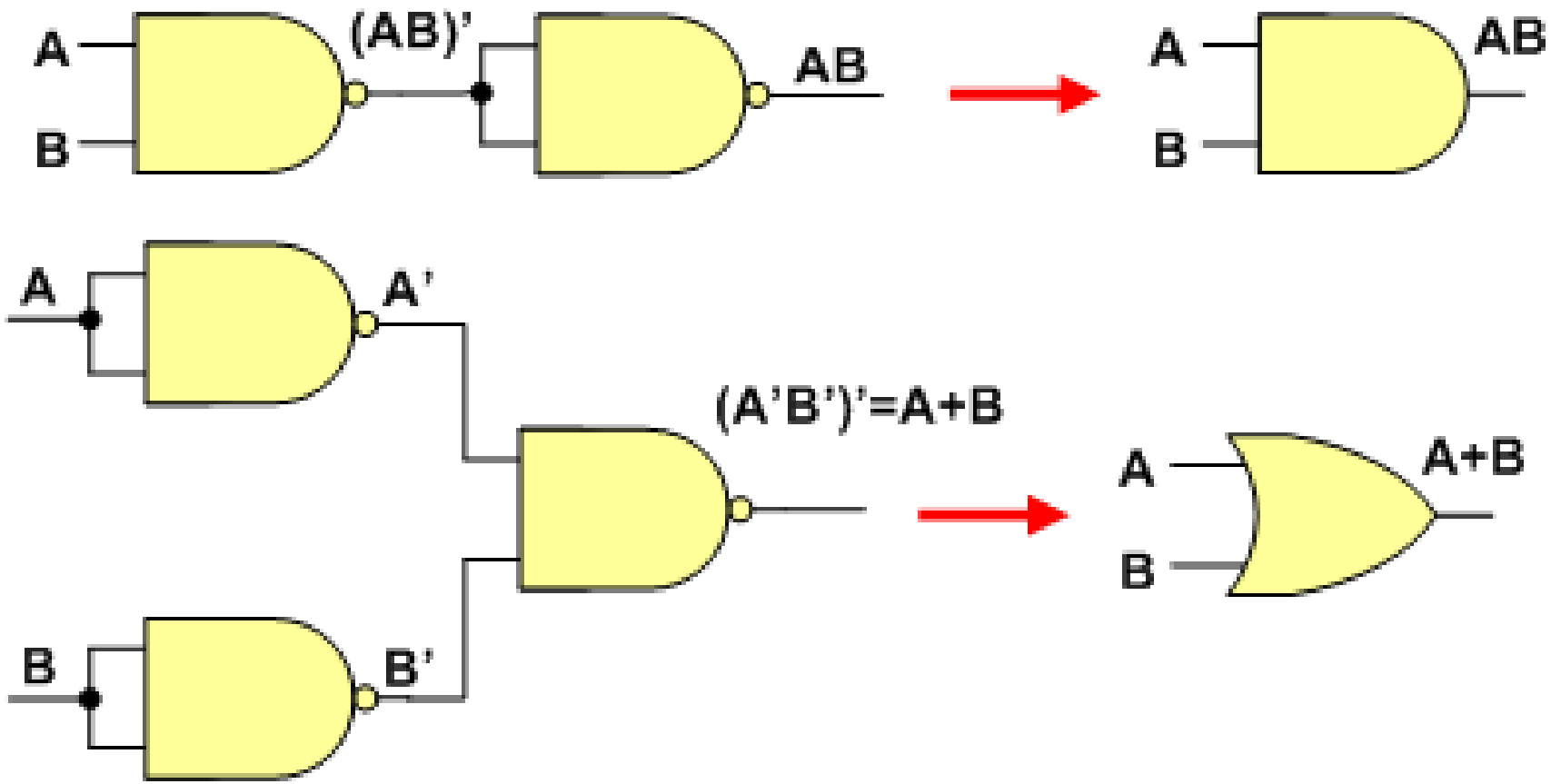


NAND gate

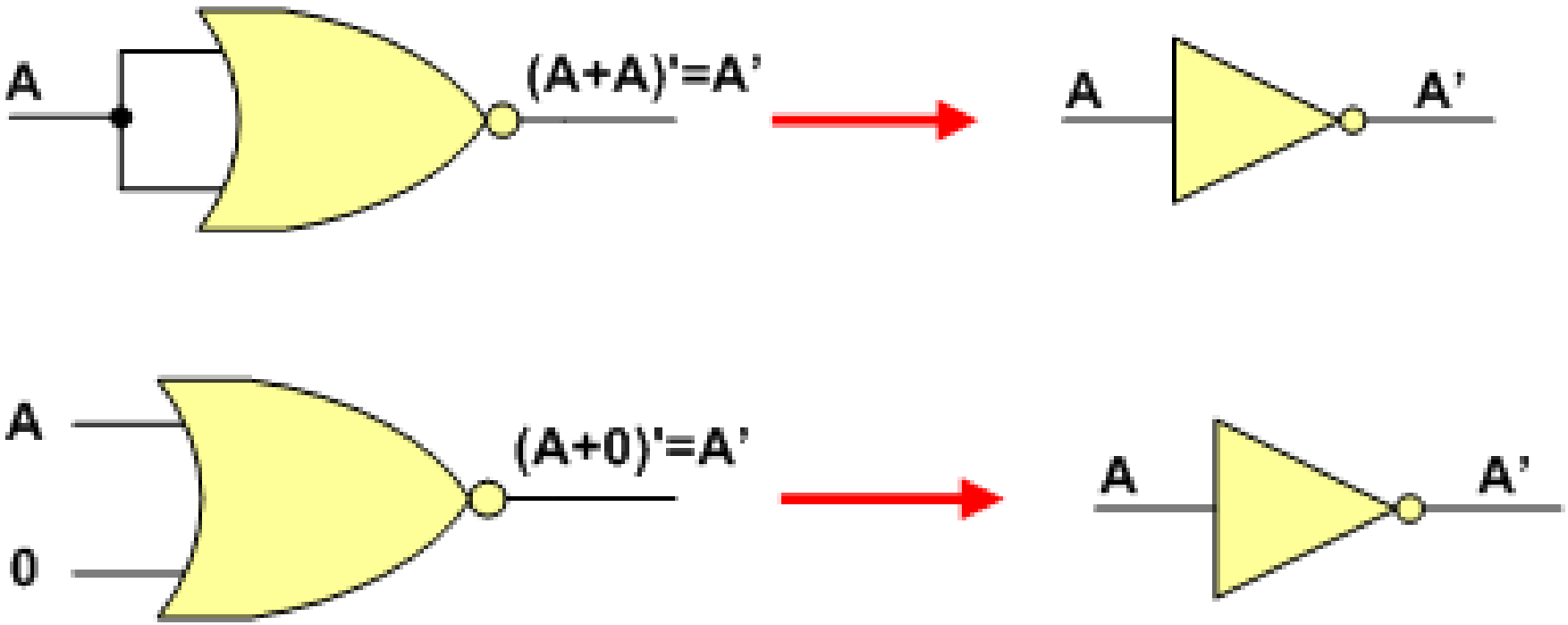
Universal NAND



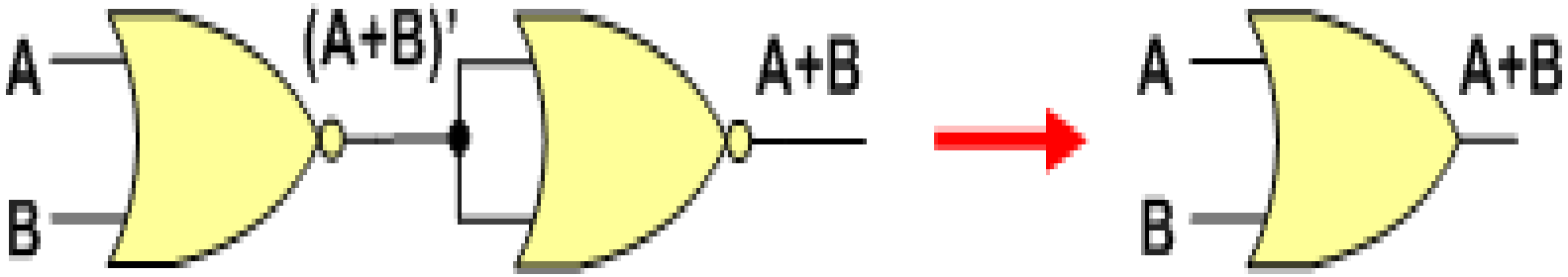
Universal NAND

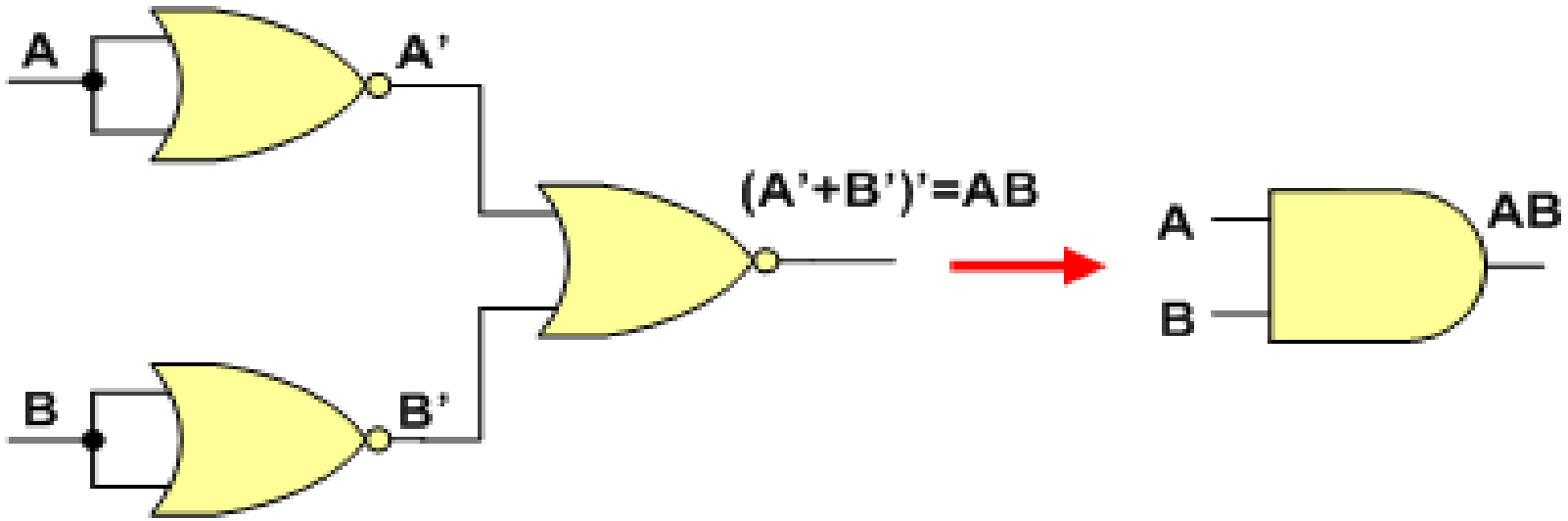


Universal gate NOR

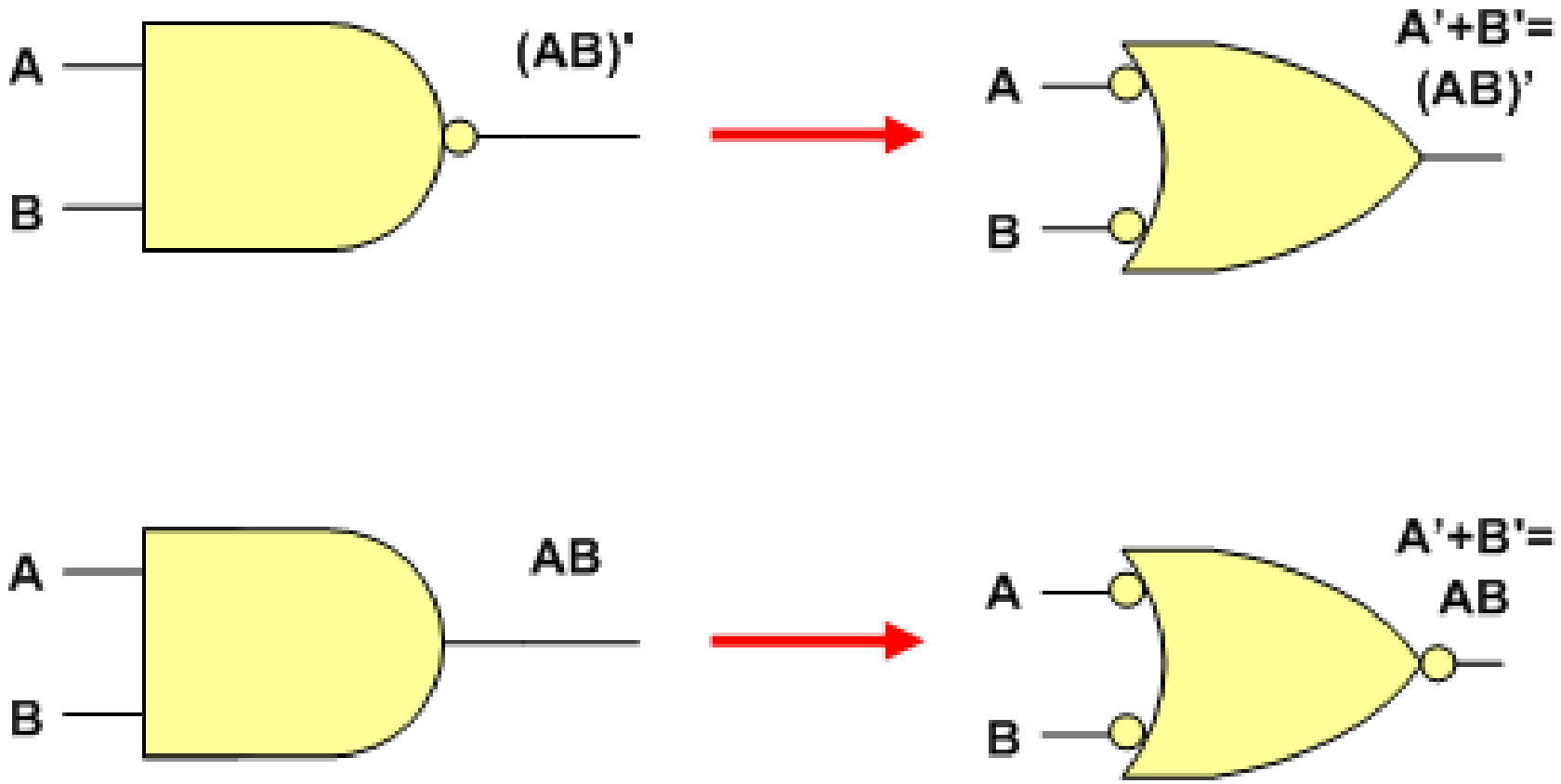


Universal gate NOR

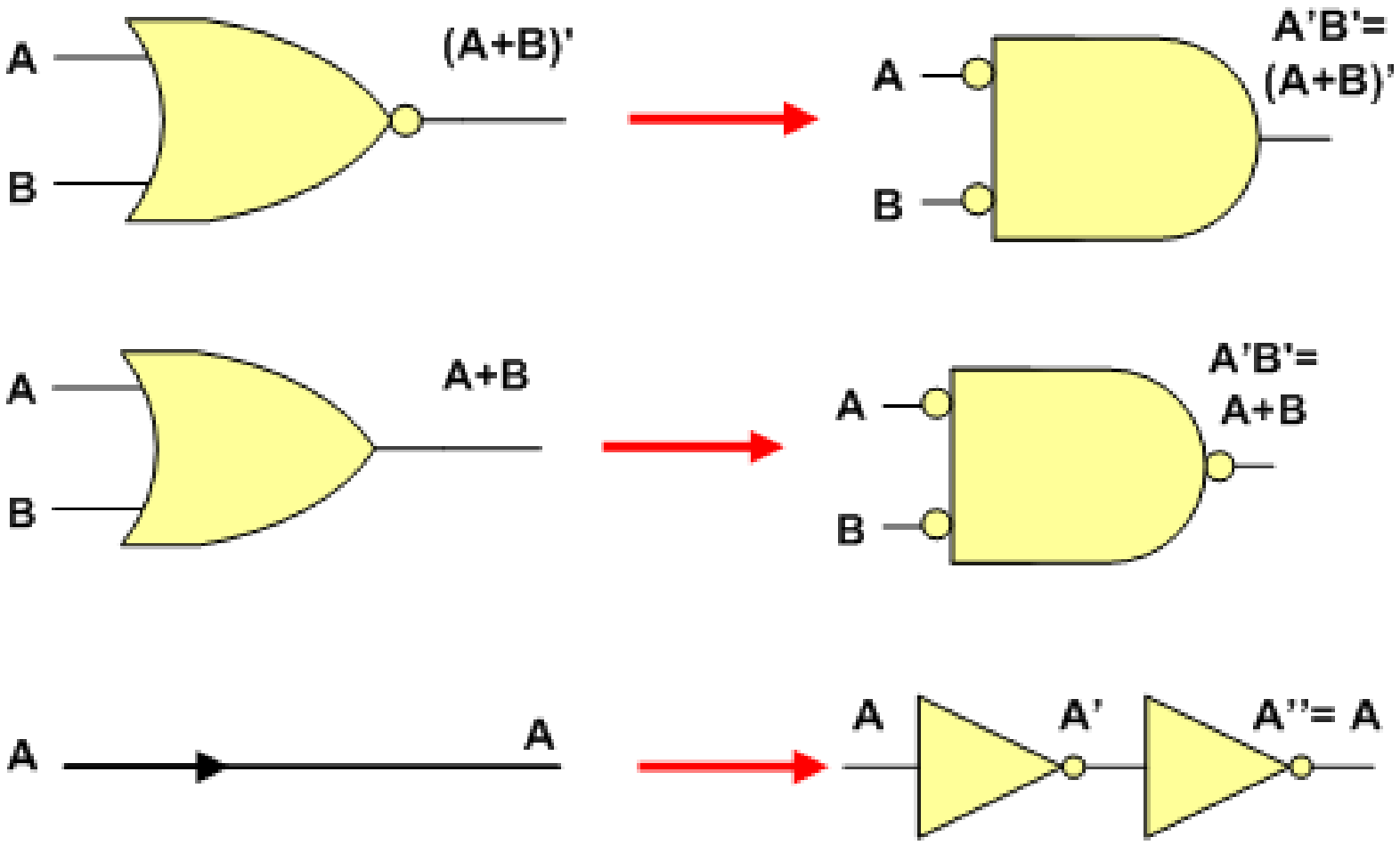




Universal gate NOR

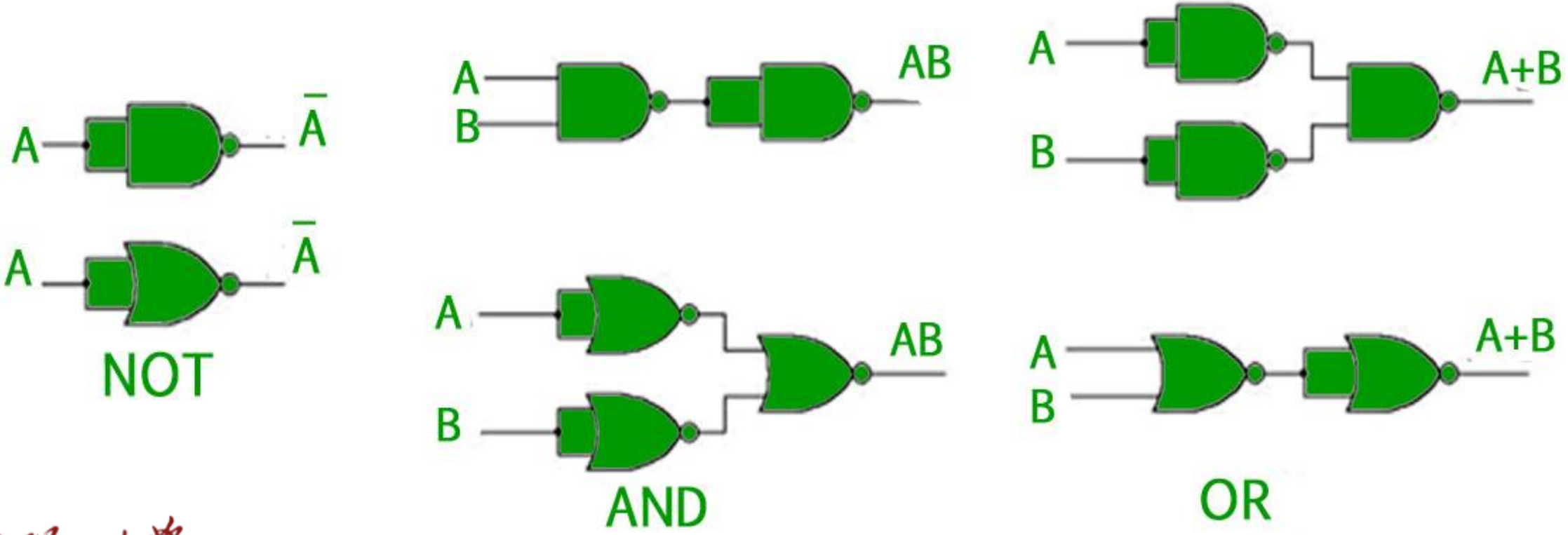


Universal gate NOR



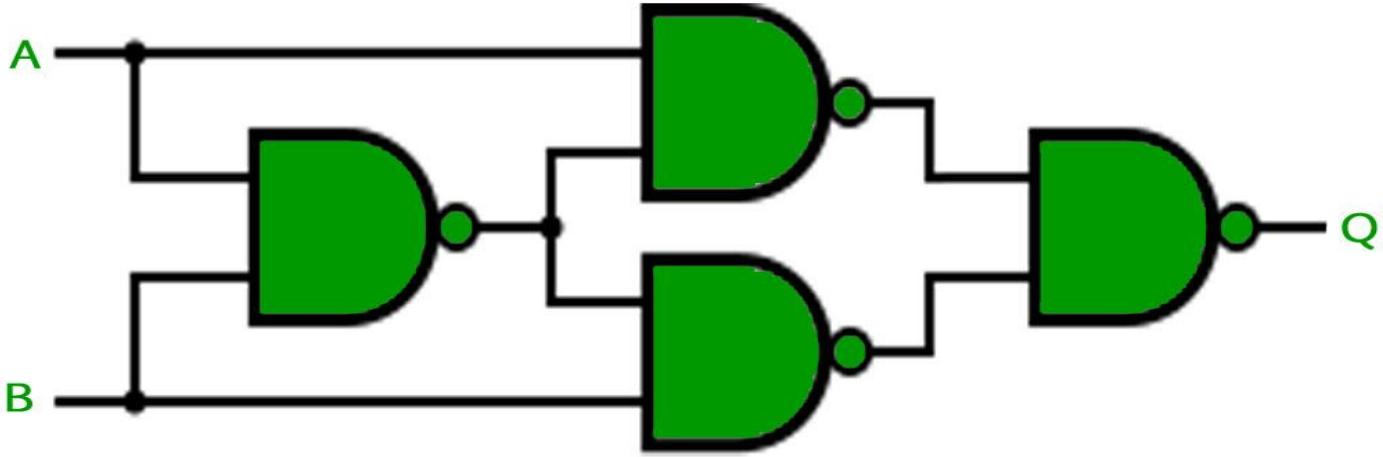
Universal Logic Gates

- Out of the seven logic gates discussed above, NAND and NOR are also known as **universal gates** since they can be used to implement any digital circuit without using any other gate. This means that every gate can be created by NAND or NOR gates only.
- Implementation of three basic gates using NAND and NOR gates is shown below –



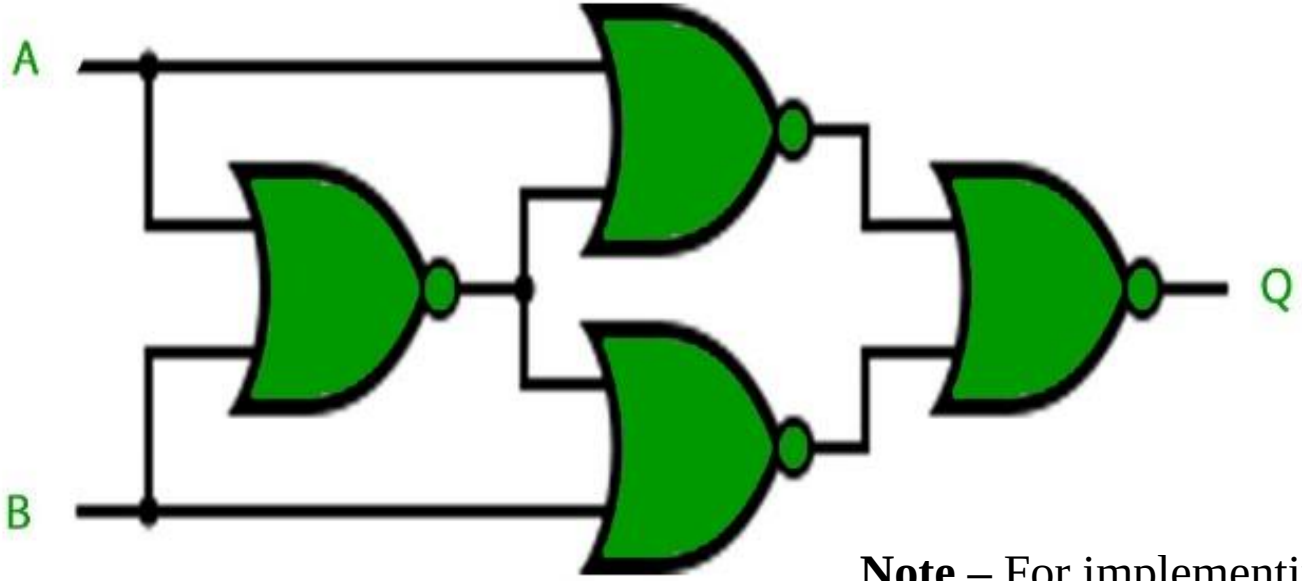
XOR gate

- For the **XOR gate**, NAND and NOR implementation is –
- Implemented **Using NAND** –



NOR

- Implemented **using NOR –**

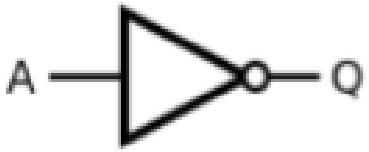


Note – For implementing XNOR gate, a single NAND or NOR gate can be added to the above circuits to negate the output of the XOR gate.

NOT

A NOT gate is made by joining the inputs of a NAND gate together. Since a NAND gate is equivalent to an AND gate followed by a NOT gate, joining the inputs of a NAND gate leaves only the NOT gate.

Desired NOT Gate



Q = NOT(**A**)

NAND Construction

= **A** NAND **A**

Truth Table

Input A	Output Q
0	1
1	0

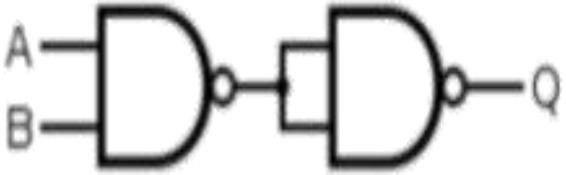
AND

- An AND gate is made by inverting the output of a NAND gate as shown below.

Desired AND Gate

Q = A AND B

NAND Construction



= (A NAND B) NAND (A NAND B)

Truth Table

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

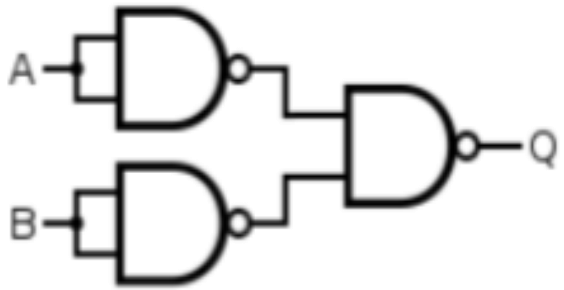
OR

- 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 0.
- To be an OR gate, however, the output must be 1 if any input is 1. Therefore, if the inputs are inverted, any high input will trigger a high output.

Desired OR Gate

Q = A OR B

NAND Construction



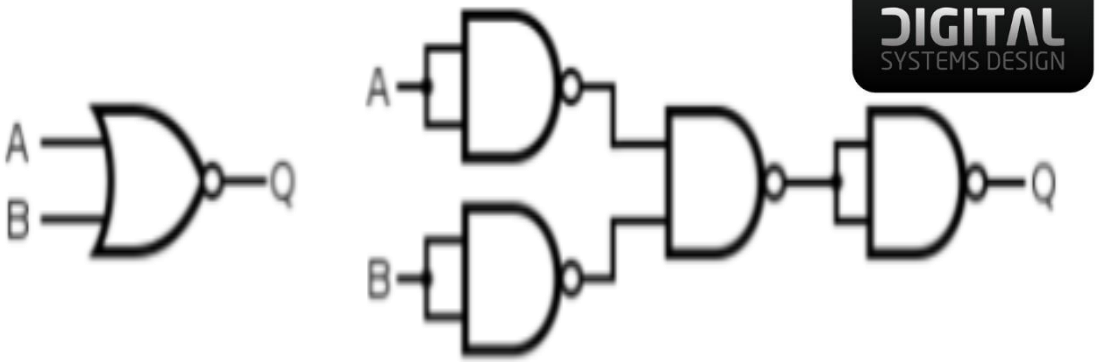
= (A NAND A) NAND (B NAND B)

Truth Table

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

NOR

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



Q = A NOR B

= [(A NAND A) NAND (B NAND B)] NAND [(A NAND A) NAND (B NAND B)]

Truth Table

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

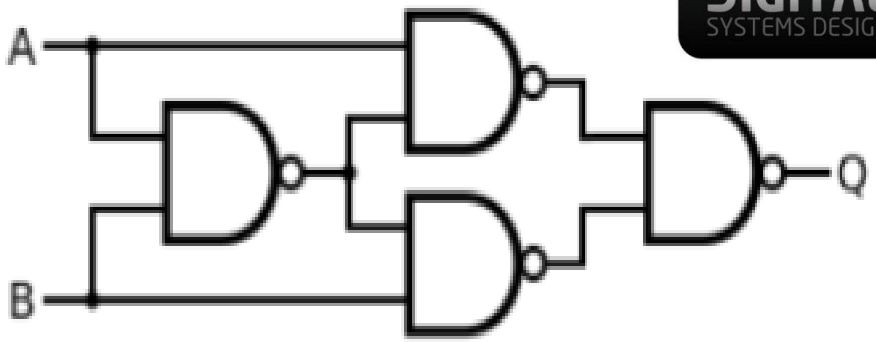
XOR

- An XOR gate is made by connecting four NAND gates as shown below. This construction entails a propagation delay three times that of a single NAND gate.

Desired XOR Gate

Q = A XOR B

NAND Construction

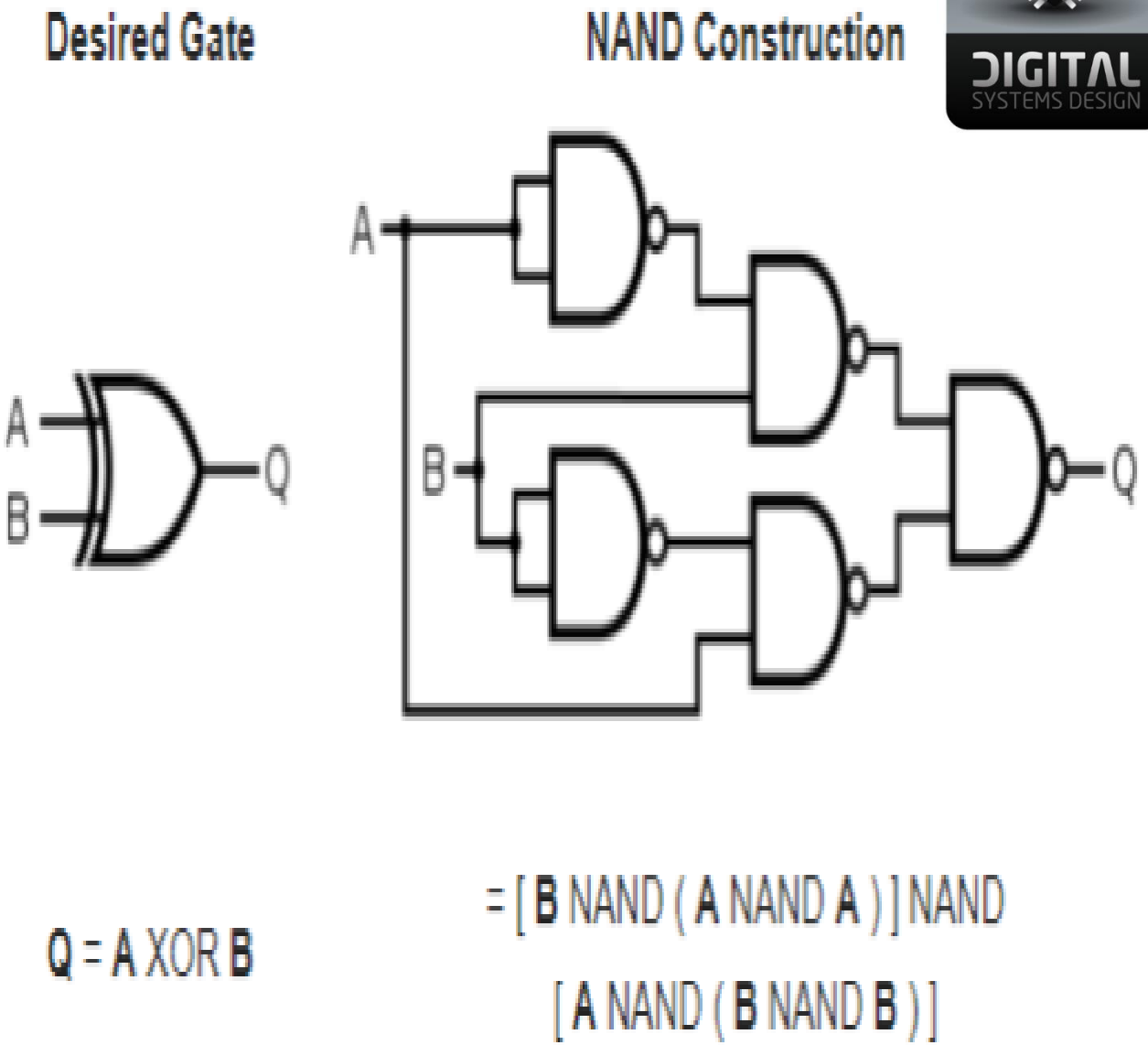


= [A NAND (A NAND B)] NAND [B NAND (A NAND B)]

Truth Table

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

- Alternatively, the B-input of the XNOR gate with the 3-gate propagation delay can be inverted. This construction uses five gates instead of four.

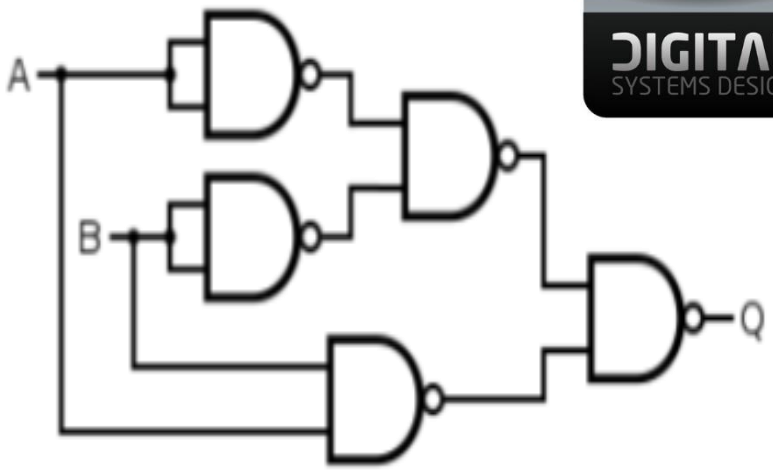


XNOR

- An XNOR gate is made by connecting the output of 3 NAND gates (connected as an OR gate) and the output of a NAND gate to the respective inputs of a NAND gate.
- This construction entails a propagation delay three times that of a single NAND gate and uses five gates.

Desired XNOR Gate

NAND Construction



Q = A XNOR B

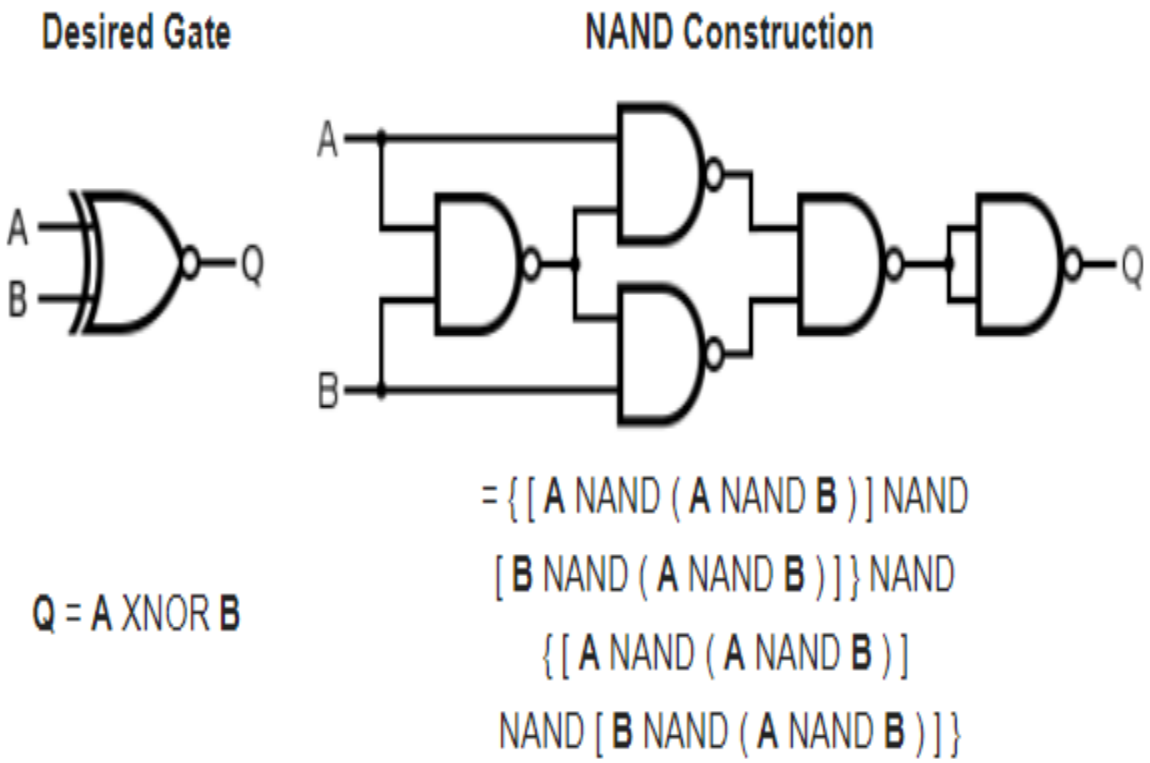
= [(A NAND A) NAND (B NAND B)] NAND (A NAND B)

Truth Table

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

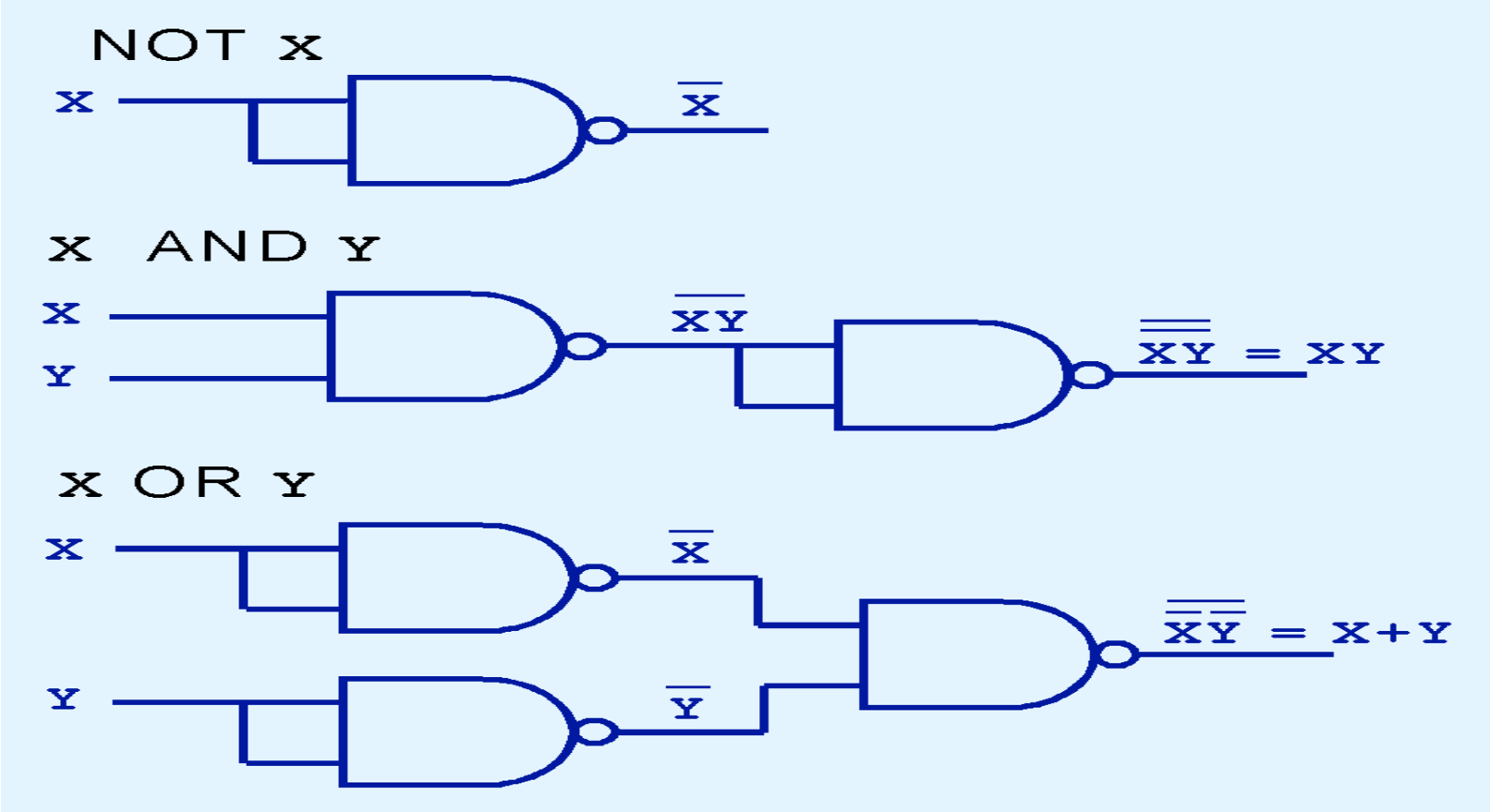
XOR gate

- Alternatively, the 4-gate version of the XOR gate can be used with an inverter.
- This construction has a propagation delay four times (instead of three times) that of a single NAND gate.

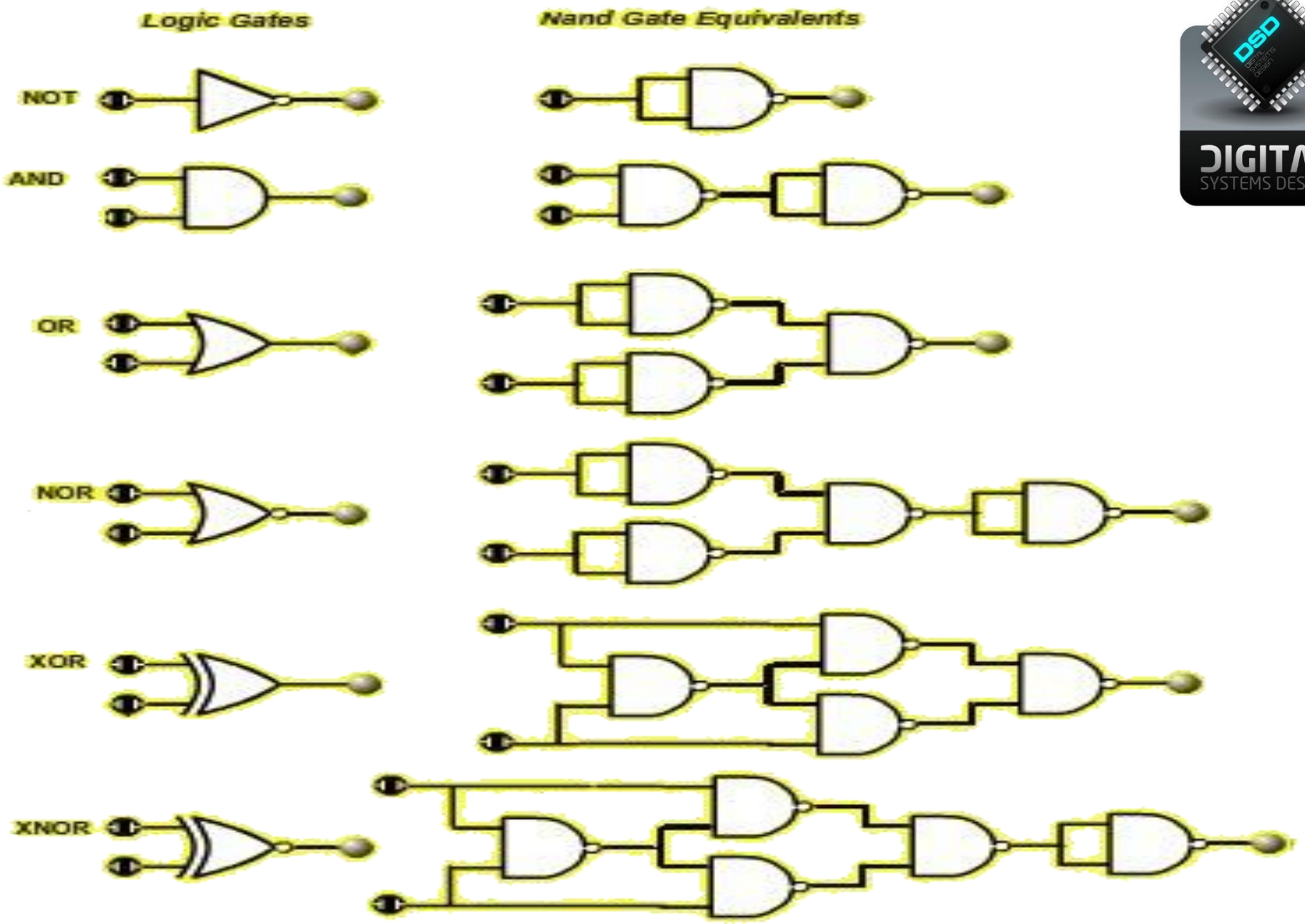


Logic Gates

- NAND and NOR are known as universal gates because they are inexpensive to manufacture and any Boolean function can be constructed using only NAND or only NOR gates.



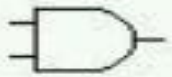
Summary

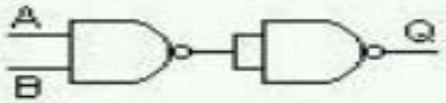


Universal Gate

All gates can be made from a CD4011 or equiv (quad NAND gate IC)

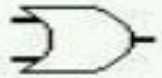
AND

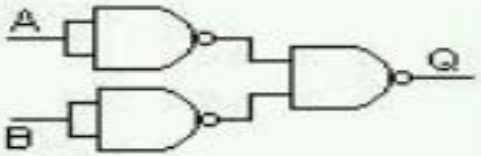




A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

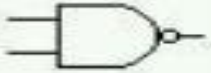
OR

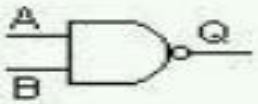




A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

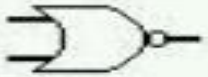
NAND

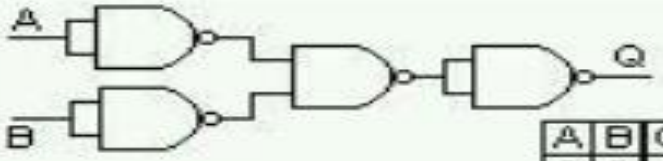




A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

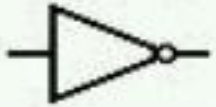
NOR

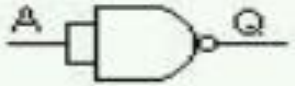




A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

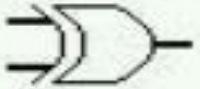
NOT

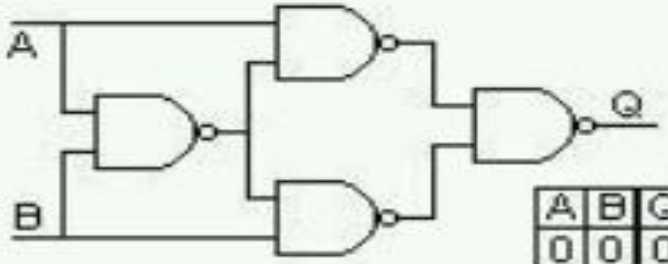




A	Q
0	1
1	0

XOR





A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Reference

http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/

https://www.quora.com/Why-is-the-NAND-GATE-called-a-universal-gate