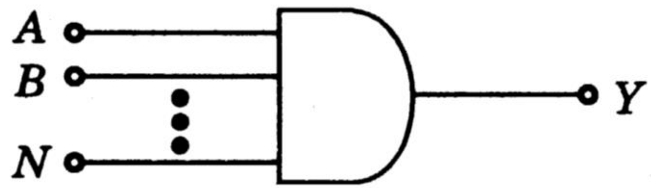
AE 242
Aerospace Measurements Laboratory

Basic digital circuits

AND operation

$$Y = A \cdot B \cdot C \cdot \ldots N$$

$$Y = ABC\ldots N$$



Standard symbol for AND gate

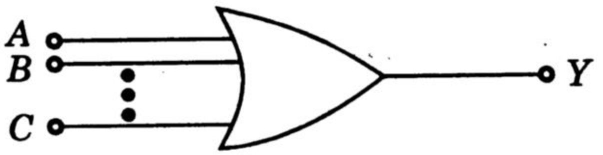
A,B,C .. N are input variables (possible values only 0 &1) and Y is output. Y will be high only when all the inputs are high (positive logic)

Inputs		Output
A	*B*	*Y*
0	0	0
0	1	0
1	0	0
1	1	1

Truth table for a 2-input AND gate

Basic digital circuits

OR operation



Standard symbol for OR gate

$$Y = A + B + C + \ldots N$$

Output of OR gate is 1, if and only if one or more inputs are 1

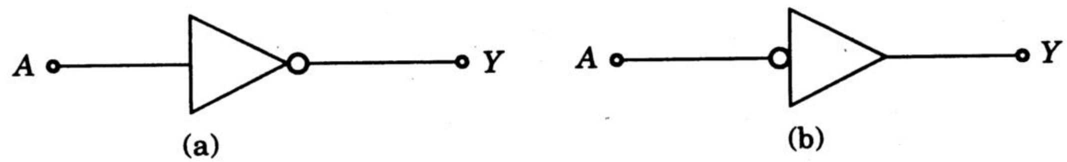
Inputs		*Output*
A	*B*	*Y*
0	0	0
0	1	1
1	0	1
1	1	1

Truth table for a 2-input OR gate

Basic digital circuits

NOT operation

$$Y = \overline{A}$$



Standard symbol for NOT gate

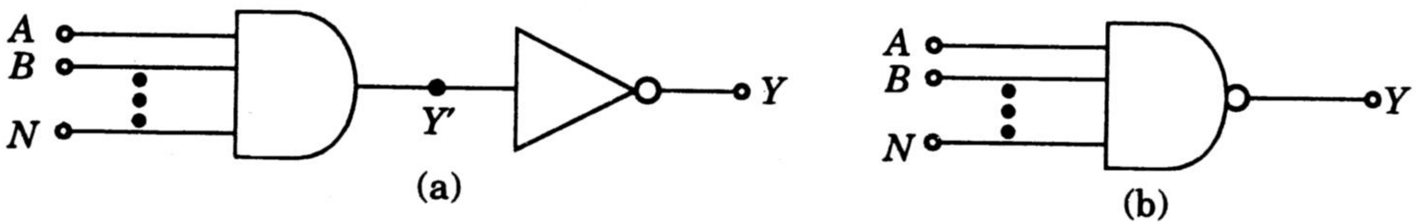
NOT gate also known as inverter. It is one input (A) and one output (Y) device. Output is complement of input. Bubble in the circuit always denotes inversion in digital circuits.

Input A	*Output* Y
0	1
1	0

Truth table for a NOT gate

NAND operations

NOT-AND operation is known as NAND, it is a AND gate followed by a NOT gate. Complemented output of AND. Standard symbol is AND with bubble.



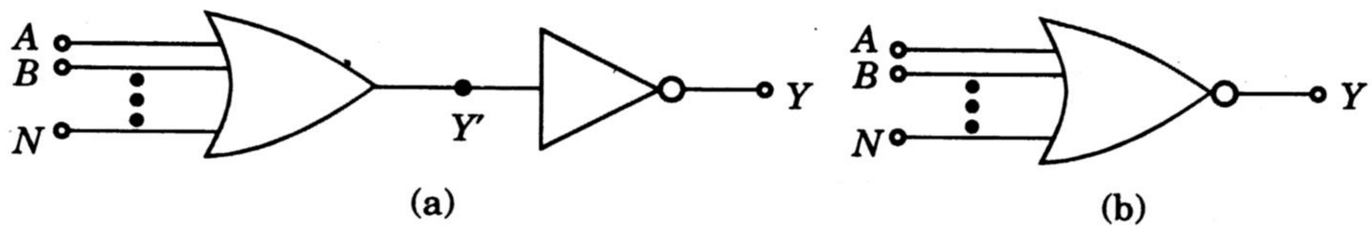
$$Y = \overline{A \cdot B \cdot C \cdot \ldots N}; Y = \overline{ABC\ldots N}$$

Inputs		Output
A	*B*	*Y*
0	0	1
0	1	1
1	0	1
1	1	0

Truth table for a NAND gate

NOR operations

NOT-OR operation is known as NOR. It is OR gate followed by a NOT gate. Complemented output of OR gate. Standard symbol is OR gate with a bubble.



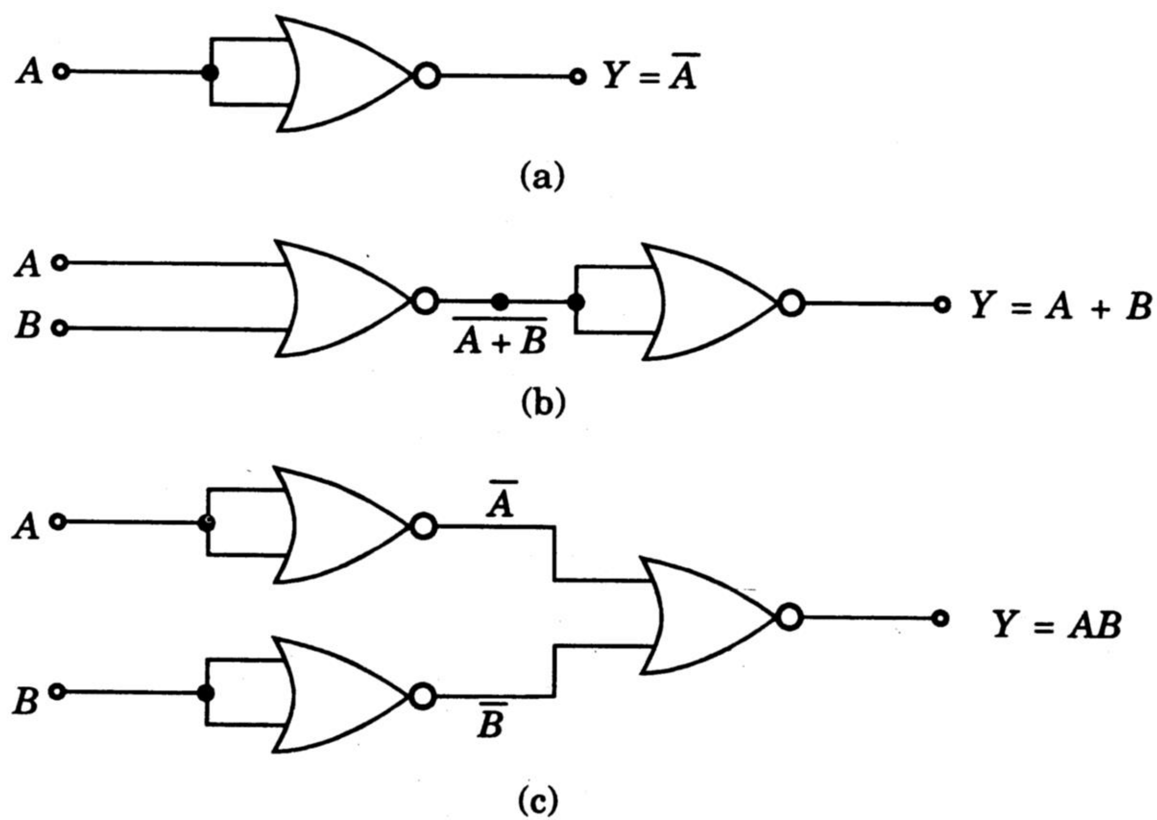
$$Y = \overline{A + B + C + \ldots N}$$

Inputs		Output
A	*B*	*Y*
0	0	1
0	1	0
1	0	0
1	1	0

Truth table for a two input NOR gate

Using NOR as universal gate

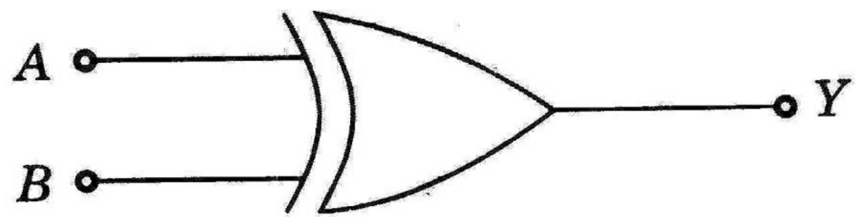
Using NOR gate, basic digital circuits i.e. AND, OR, NOT can be obtained and this property makes it a universal gate.



Exclusive-OR (EX-OR) operations

It is not a basic gate, and the operation can be performed using the basic gates - AND, OR and NOT. Output is logic one when odd number of inputs are one.

Standard notation $Y = A \oplus B$ $\qquad Y = A\overline{B} + \overline{A}B$

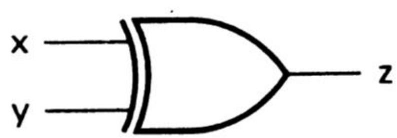


Inputs A	Inputs B	Output Y
0	0	0
0	1	1
1	0	1
1	1	0

Truth table for a EX-OR gate

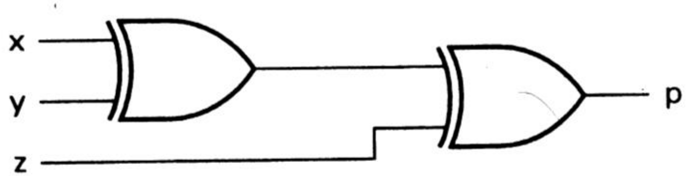
Exclusive-OR (EX-OR) operations

Truth table for three inputs. Output is logic one when odd number of inputs are one.



$z = x \oplus y$

(a) 2–Input EX-OR Gate



$p = x \oplus y \oplus z$

(b) 3–Input EX-OR function derived from Two 2-input-EX-OR gates

Input			Output
x	y	z	p
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth table for a EX-OR gate

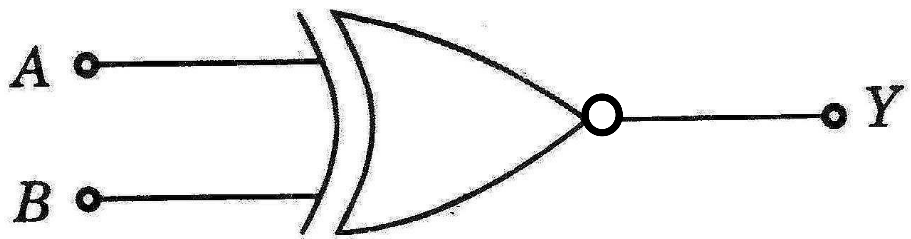
Exclusive-NOR (EX-NOR) operations

It is EX-OR gate followed by a NOT

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

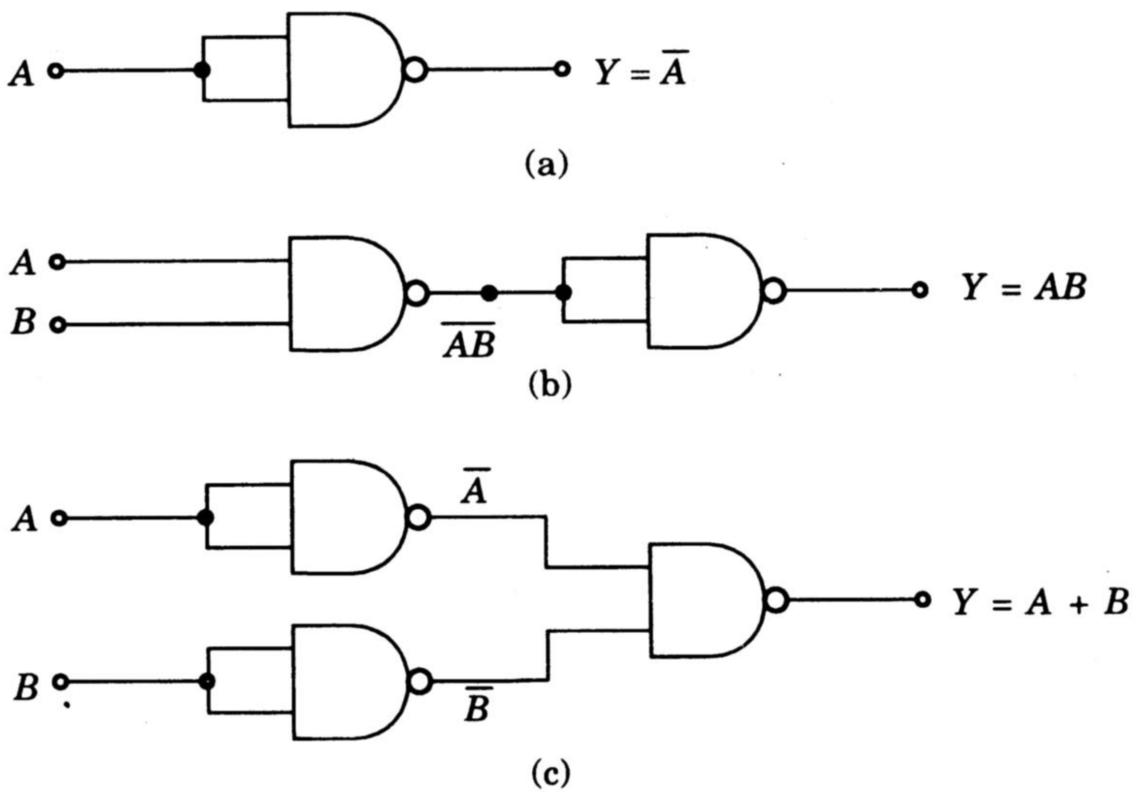
Standard notation $\quad Y = A \odot B$

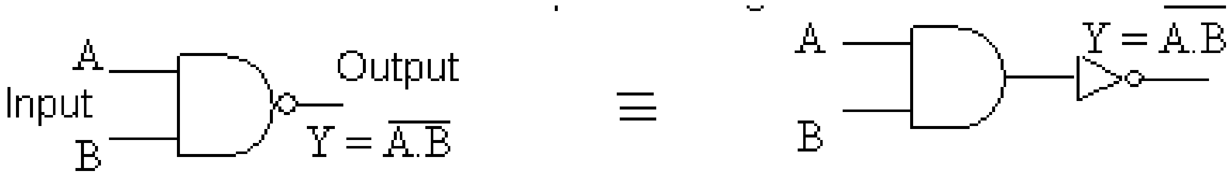
$$Y = \overline{A}\,\overline{B} + AB$$



NAND as universal gate

Using NAND gate, basic digital circuits i.e. AND, OR, NOT can be obtained and this property makes it a universal gate.





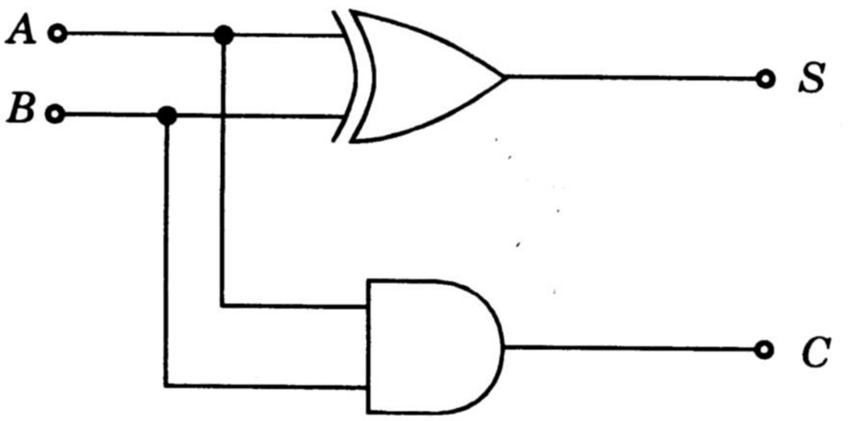
Truth Table		
Input		Output
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Components to be use for experiment		Quantity
7400	2 input quad NAND gate	1
7805	Five volt regulator	1
220 Ω resistance		2
LED		2
Bread board		1
9 V battery		1

Experiment: a) Create NOT Gate using NAND Gate
b) Create AND Gate using NAND Gate
c) Create OR Gate using NAND Gate

Half adder

A logic circuit adding two bits is called as half adder. A and B are two inputs. S is sum and C is carry as output.

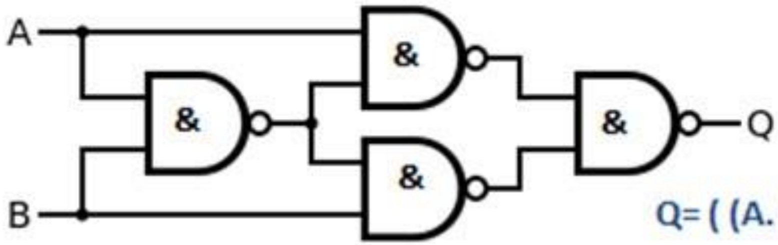


Sum $S = A \oplus B$

Carry $C = AB$

Inputs		Outputs	
A	*B*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Experiment: a) Create half adder using NAND and AND Gates



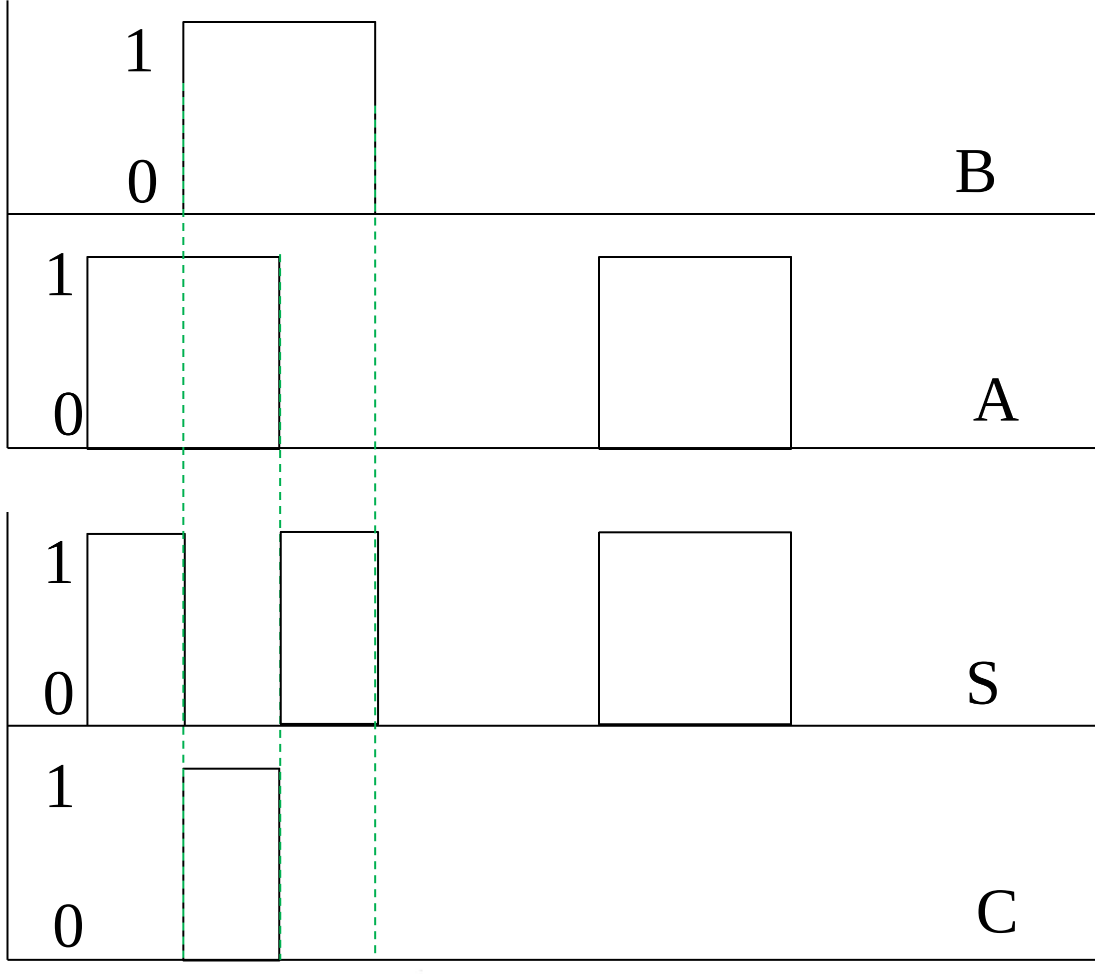
$$Q = A \oplus B = \overline{\overline{A.\overline{AB}}\,.\,\overline{B.\overline{AB}}}$$

Q= ((A. (AB)')'. (B. (AB)'))'

Truth Table		
Input		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Components to be use for experiment		Quantity
7400	2 input quad NAND gate	1
7408	2 input AND gate	1
7805	Five volt regulator	1
220 Ω resistance		2
LED		2
Bread board		1
9 V battery		1

Experiment: XOR gate using NAND Gates



Inputs		*Outputs*	
A	*B*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Experiment: Give the inputs as shown above and observe the output