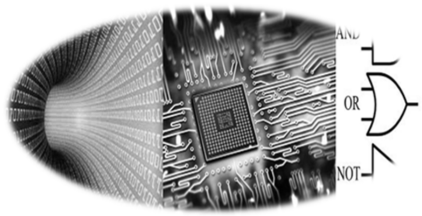
DIGITAL ELECTRONICS

Lecture Note 02: Boolean Algebra and Logic Gates

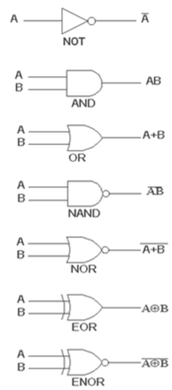


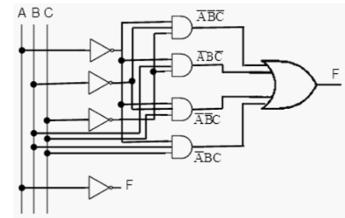
Dr. Tushar Kanti Bera

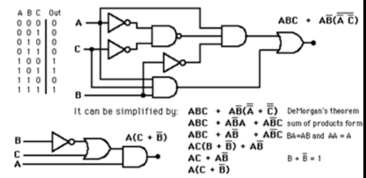
Department of Electrical Engineering
National Institute of Technology Durgapur (NITD)
Mahatma Gandhi Rd, A-Zone, Durgapur, West Bengal 713209, India
Date: Jan-Jun, 2019

Introduction to Logic Gates

- NOT
- AND
- OR
- NAND
- NOR
- XOR
- XNOR

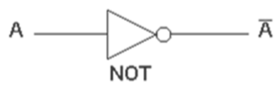






NOT Gate

- NOT provides the inversion of the digital signal
- NOT gate is also called as Inverter



- Boolean expression or Gate Equation

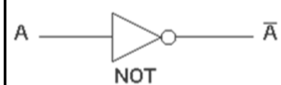
$$X = \overline{A}$$

- Truth Table

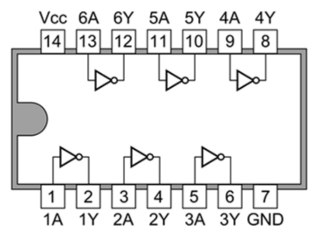
A	X
0	1
1	0

74LS04 NOT Gate IC

- NOT Gate

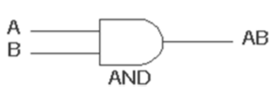


7404 Hex Inverters



AND Gate

- AND gate provides the AND operation of the digital signal



- Boolean expression or Gate Equation

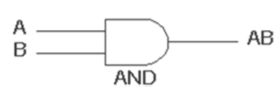
$$X = A.B$$

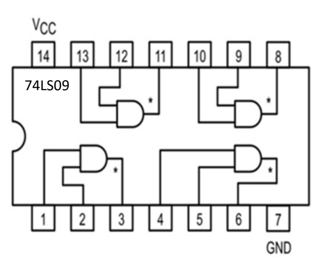
- Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Gate IC

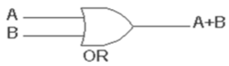
- AND gate provides the AND operation of the digital signal





OR Gate

- OR gate provides the OR operation of the digital signal



- Boolean expression or Gate Equation

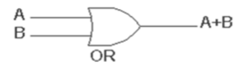
$$X = A+B$$

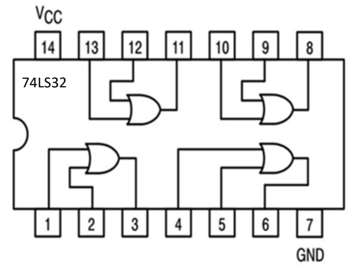
- Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Gate IC

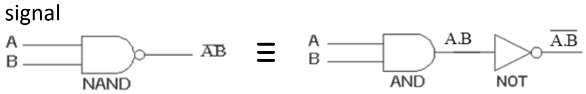
- OR gate provides the OR operation of the digital signal





NAND Gate

- NAND gate provides the (AND+NOT) operation of the digital signal



- Boolean expression or Gate Equation

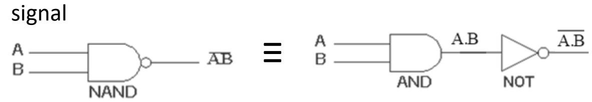
$$X = \overline{A.B}$$

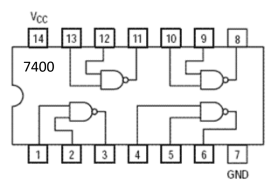
- Truth Table

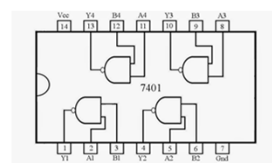
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

NAND Gate IC

- NAND gate provides the (AND+NOT) operation of the digital signal

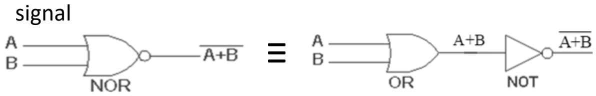






NOR Gate

- NOR gate provides the (OR+ NOT) operation of the digital signal



- Boolean expression or Gate Equation

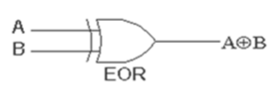
$$X = \overline{A + B}$$

- Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

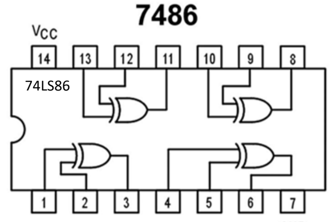
XOR Gate

- XOR gate provides the exclusive OR operation of the digital signal



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

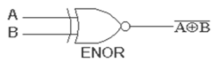
7486



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

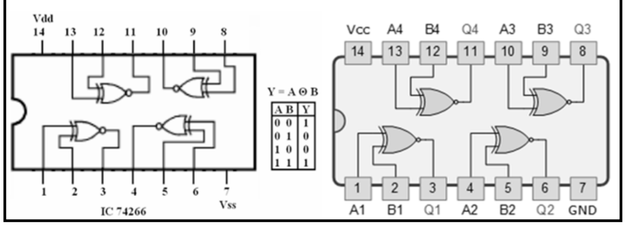
XNOR Gate IC

- XNOR gate provides the exclusive NOR operation of the digital signal

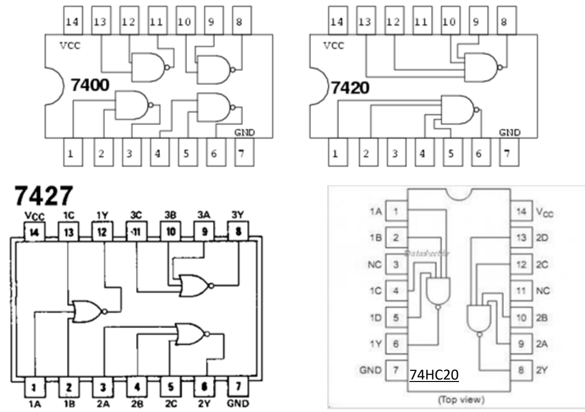


$$X = A.B + \overline{A}.\overline{B}$$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

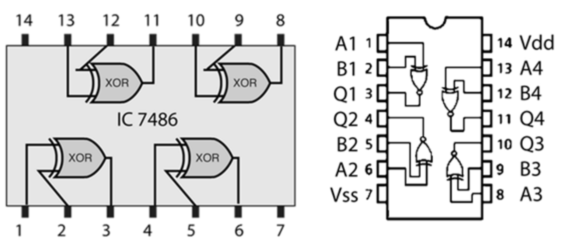


Gate IC Pin Diagram



Gate IC Pin Diagram

Pinout diagram of the 74HC266N, 74LS266 and CD4077 quad XNOR plastic dual in-line package 14-pin package (**PDIP-14**) ICs.



http://www.wikiwand.com/en/XNOR_

http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/xor.html

Timing Diagram

- Timing diagram of the AND gate:

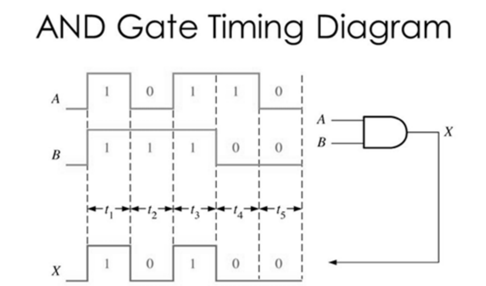
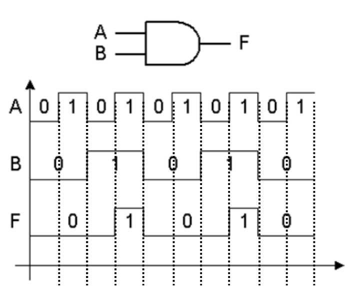


Figure 3–10 Example of pulsed AND gate operation with a timing diagram showing input and output relationships.

10

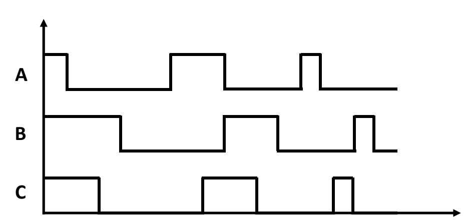
Timing Diagram

- Timing diagram of the AND gate:



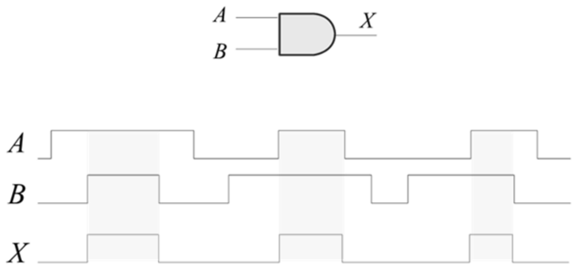
Timing Diagram

- Timing diagram of the gates:
- Timing diagram of the digital signal is the representation of the signal amplitude with respect to time.
- The figure below shows a timing diagram of three digital signals A, B and C



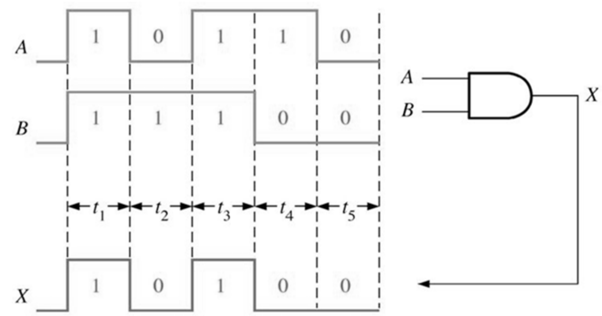
Timing Diagram

- Timing diagram of a two-input AND gate:



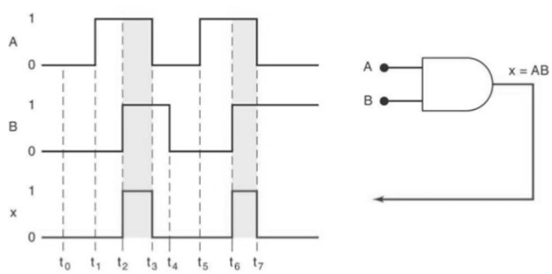
Timing Diagram: AND Gate

- Timing diagram of a two-input AND gate:



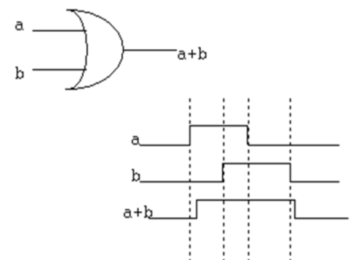
Timing Diagram: AND Gate

- Timing diagram of a two-input AND gate:



Timing Diagram: OR Gate

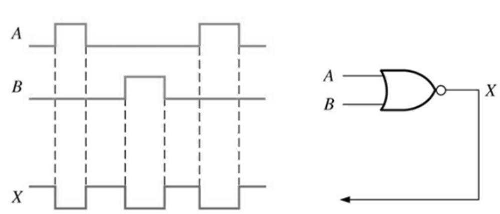
- Timing diagram of a two-input OR gate:



Timing Diagram: NOR Gate

- Timing diagram of a two-input NOR gate:

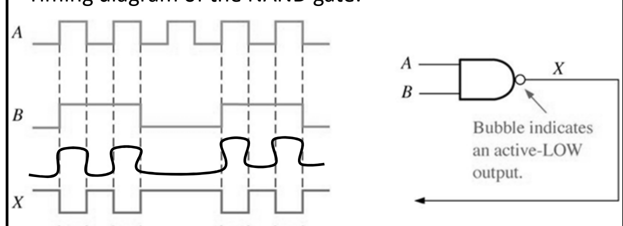
NOR Gate Timing Diagram



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

Timing Diagram: NAND Gate

- Timing diagram of the NAND gate:

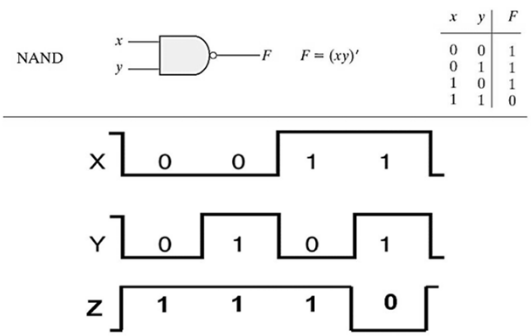


A and *B* are both HIGH during these four time intervals. Therefore *X* is LOW.

$$X = \overline{AB}$$

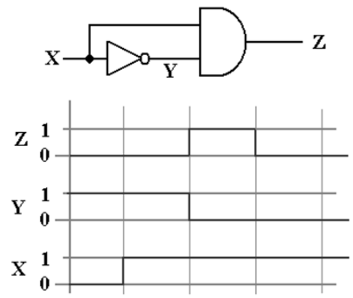
Timing Diagram: NAND Gate

• Timing diagram of the NAND gate:



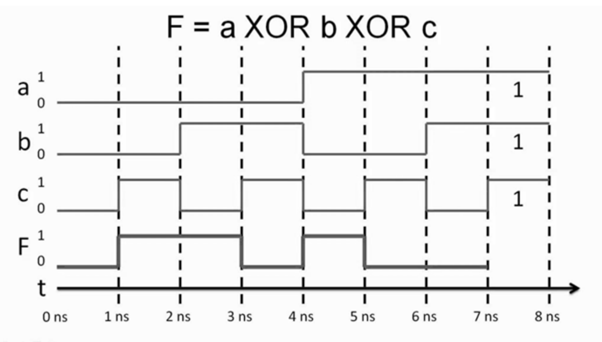
Timing Diagram: Gate Combination

• Timing diagram of the gate combination:



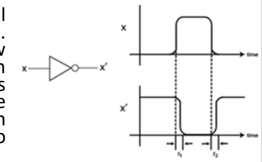
Timing Diagram: XOR Gate

• Timing diagram of the XOR gate:



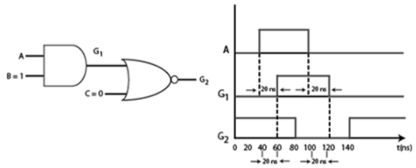
Delays in Gates and Timing Diagrams

- If the input of a logic gate is changed, the output will not change instantaneously. The reason behind this is elements that switch the inputs within the gate take a fixed time to react to change, so the resulting change output is delayed with respect to the input. The propagation delay in an inverter figure shown below any possible waveforms of input and output for an inverter. For a change in output delayed by time, ε, taken respectfully to the input, is said to have a propagation delay of ε.
- A propagation delay for a 0 to 1 output change may be different than that of a delay for a 1 to 0 change. Some propagation delays can be neglected if they are as short as a few nanoseconds. However, it is good practice to analyze these sequential circuits, no matter how short the delay may be.
- More than likely, a timing diagram will be used to analyze sequential circuits. Timing diagrams can be used to show different signals in a circuit as a function of time. When plotting numerous variables, they are plotted along the same time scale so the times at which these variables change with respect to each other can be easily understood.



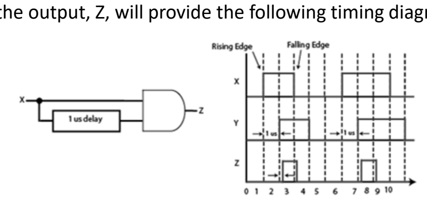
Delays in Gates and Timing Diagrams

- For a circuit with two gates, as shown below, each gate is assumed to have propagation delay of 20 ns. The diagram specifies what will happen if gate inputs B and C are held at regular values of 1 and 0, respectively, and gate input A is changed to 1 at t = 40 ns where it is then changed back to 0 at t = 100 ns. The gate output of G_1 changes exactly 20 ns after A changes, and finally the gate output of G_2 changes exactly 20 ns after G_1 changes.

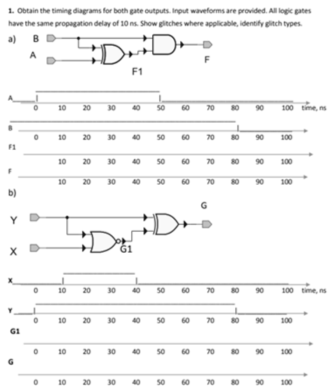


Delays in Gates and Timing Diagrams

- The figure below depicts a timing diagram for a logic circuit with a delay element. Input X consists of two pulses: the first pulse is 2 microseconds wide and the second pulse is 3 microseconds wide. The delay element that tangents off of X has output Y, which is identical to the input, X, but it is delayed by 1 microsecond. What this means is that Y changes from a value of 1, 1 microsecond after the rising edge of the X pulse and then returns to a value of 0, 1 microsecond after the falling edge of the X pulse. Z, which is the output of this AND gate, should always be 1 during the time interval for which both X and Y have a value of 1. Assuming that there is a small propagation delay in the AND gate of ε
- , then the output, Z, will provide the following timing diagram:

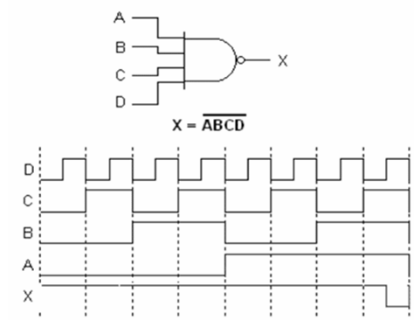


Delays in Gates and Timing Diagrams



Gates with More Inputs

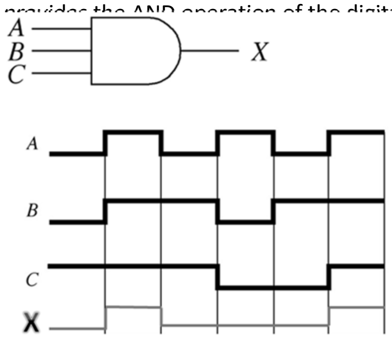
- AND gate provides the AND operation of the digital signal



A	B	C	D	X
0	0	0	0	1
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	0

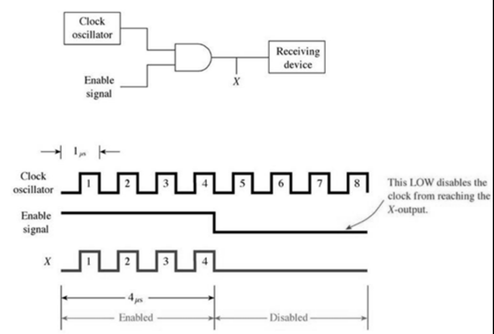
Gates with More Inputs

- AND gate provides the AND operation of the digital signal

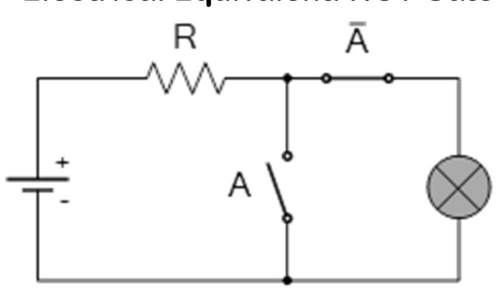


Gates with More Inputs

- AND gate provides the AND operation of the digital signal

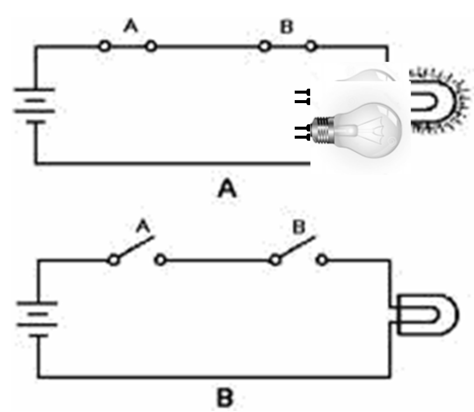


Electrical Equivalent: NOT Gate

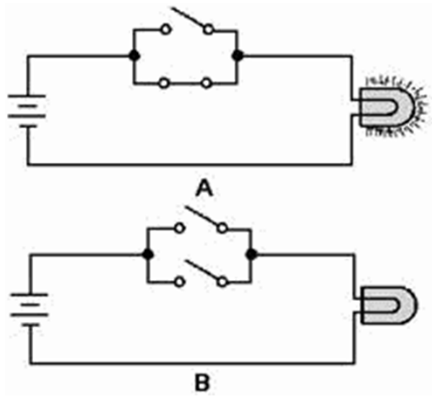


Switch A - Open = "0", Lamp - ON = "1"
Switch A - Closed = "1", Lamp - OFF = "0"

Electrical Equivalent: AND Gate



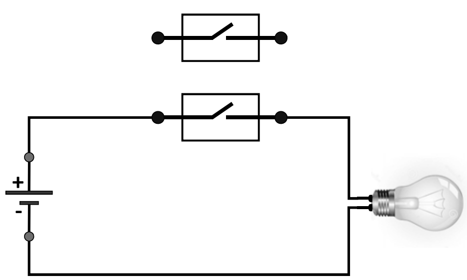
Electrical Equivalent: OR Gate



Electrical Equivalent: NOR Gate

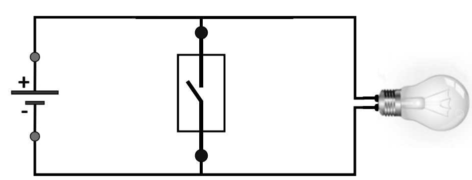
Electrical Equivalent: Gate or Switch

- A switch is a electrical/electronic component which make or break a circuit.



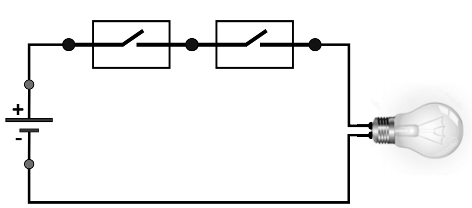
Electrical Equivalent: NOT Gate

- A switch is a electrical/electronic component which make or break a circuit.



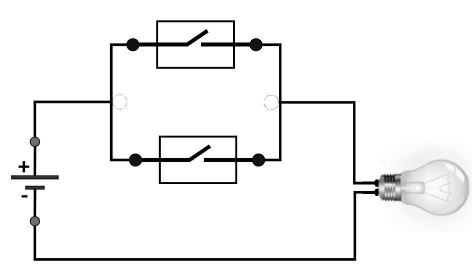
Electrical Equivalent: AND Gate

- A switch is a electrical/electronic component which make or break a circuit.



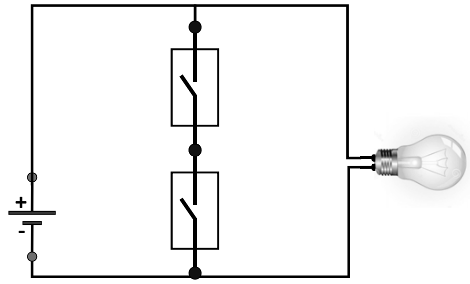
Electrical Equivalent: OR Gate

- A switch is a electrical/electronic component which make or break a circuit.



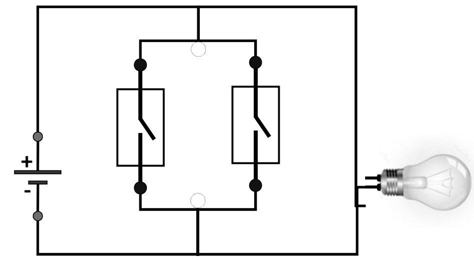
Electrical Equivalent: NAND Gate

- **A switch is a electrical/electronic component which make or break a circuit.**



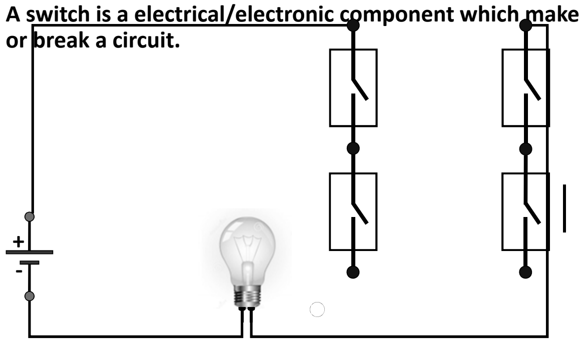
Electrical Equivalent: NOR Gate

- **A switch is a electrical/electronic component which make or break a circuit.**



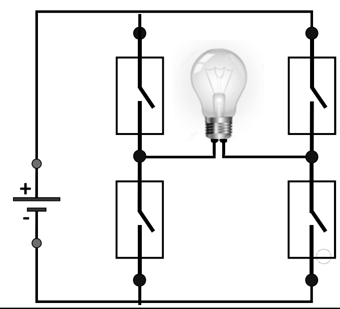
Electrical Equivalent: XOR Gate

- **A switch is a electrical/electronic component which make or break a circuit.**



Electrical Equivalent: XOR Gate

- **A switch is a electrical/electronic component which make or break a circuit.**



Boolean Operations and Expressions

- Addition

  0 + 0 = 0

  0 + 1 = 1

  1 + 0 = 1

  1 + 1 = 1

- Multiplication

  0 * 0 = 0

  0 * 1 = 0

  1 * 0 = 0

  1 * 1 = 1

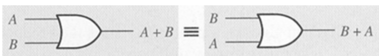
Laws of Boolean Algebra

- Commutative Laws
- Associative Laws
- Distributive Law

Commutative Law: Addition

- Commutative Law of Addition:

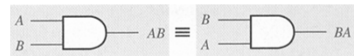
A + B = B + A



Commutative Law: Multiplication

- Commutative Law of Multiplication:

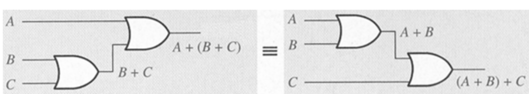
A * B = B * A



Associative Law: Addition

- Associative Law of Addition:

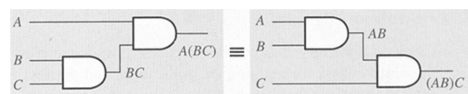
A + (B + C) = (A + B) + C



Associative Law: Multiplication

- Associative Law of Multiplication:

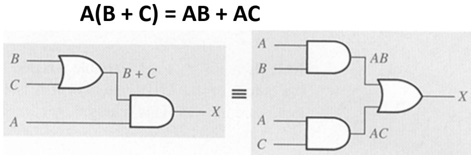
A * (B * C) = (A * B) * C



Distributive Law

- Distributive Law:

A(B + C) = AB + AC



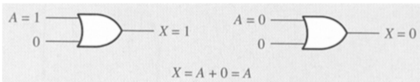
$X = A(B + C)$ $X = AB + AC$

Rules of Boolean Algebra

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = A$
5. $A + A = A$
6. $A + \overline{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

Rules of Boolean Algebra: Rule 1

• Rule 1

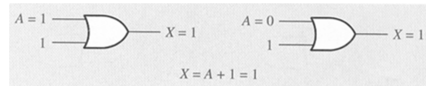


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 2

• Rule 2



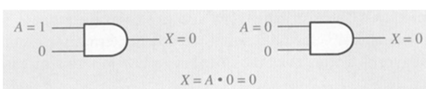
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 3

Rules of Boolean Algebra

• Rule 3



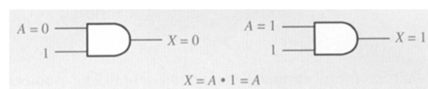
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra: Rule 4

Rules of Boolean Algebra

• Rule 4



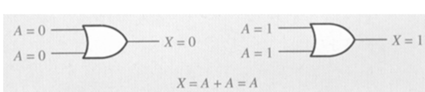
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra: Rule 5

Rules of Boolean Algebra

• Rule 5



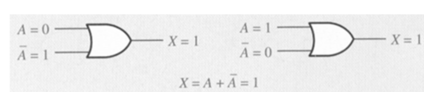
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 6

Rules of Boolean Algebra

• Rule 6



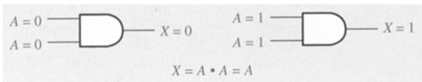
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 7

Rules of Boolean Algebra

- Rule 7



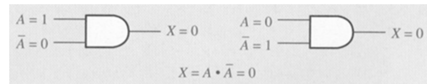
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra: Rule 8

Rules of Boolean Algebra

- Rule 8



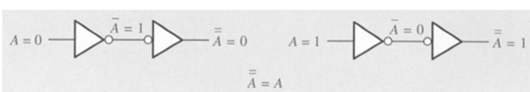
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Rules of Boolean Algebra: Rule 9

Rules of Boolean Algebra

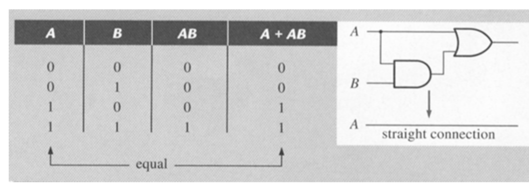
- Rule 9



Rules of Boolean Algebra: Rule 10

Rules of Boolean Algebra

- Rule 10: A + AB = A



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

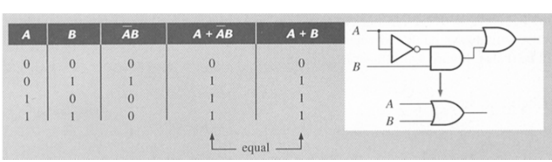
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 11

Rules of Boolean Algebra

- Rule 11: $A + \overline{A}B = A + B$



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

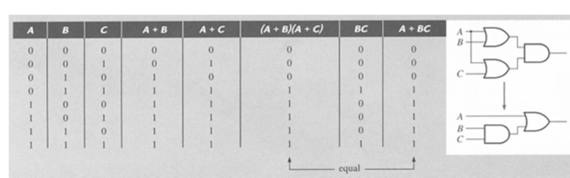
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Rules of Boolean Algebra: Rule 12

Rules of Boolean Algebra

- Rule 12: (A + B)(A + C) = A + BC



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

DeMorgan's Theorem

- Theorem 1

$$\overline{XY} = \overline{X} + \overline{Y}$$

- Theorem 2

$$\overline{X+Y} = \overline{X}\,\overline{Y}$$

- Theorem 1: for n inputs

$$\overline{X_1X_2X_3...X_{n-1}X_n} = \overline{X_1} + \overline{X_2} + \overline{X_2} + ...+ \overline{X_{n-1}} + \overline{X_n}$$

- Theorem 2: for n inputs

$$\overline{X_1 + X_2 + X_3 + ...+ X_{n-1} + X_n} = \overline{X_1}\cdot\overline{X_2}\cdot\overline{X_2}\cdot.....+\overline{X_{n-1}}\cdot\overline{X_n}$$

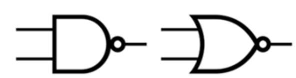
Remember:

BBCTS:

"Break the bar, change the sign"

Universal Gate – NAND

- **Universal Gate:** The GATES which can be used to develop other GATES
- **NAND** and **NOR** Gates



- We will learn
  - **NAND** gate can be used to replace an **AND** gate, **OR** gate, or an **INVERTER** gate.
  - How a logic circuit implemented with **AND-OR-Invert (AOI)** logic gates can be re-implemented using only **NAND** gates.
  - That using a single gate type, in this case **NAND**, will reduce the number of integrated circuits (IC) required to implement a logic circuit.

AOI Logic

More ICs = More $$

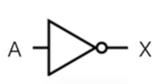
NAND Logic

Less ICs = Less $$

70

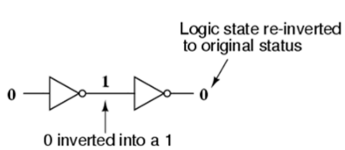
NOT Gate

symbol of NOT gate



where, X = **NOT** A

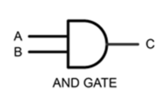
Double inversion



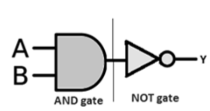
71

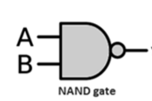
NAND Gate

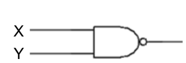
AND Gate



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1







$Z = \overline{X\,Y} = \overline{X} + \overline{Y}$

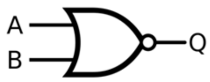
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

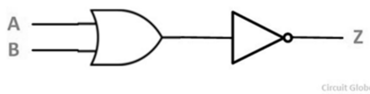
72

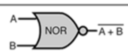
NOR Gate

OR Gate

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1





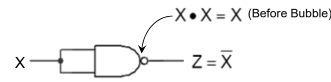


A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

73

NAND Gate as an Inverter Gate

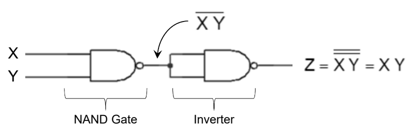


X	Z
0	1
1	0

Equivalent to Inverter

74

NAND Gate as an AND Gate

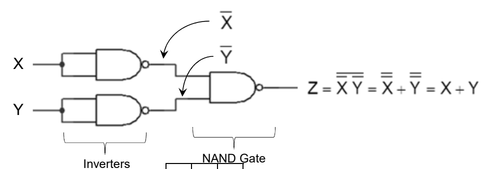


X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Equivalent to AND Gate

75

NAND Gate as an OR Gate

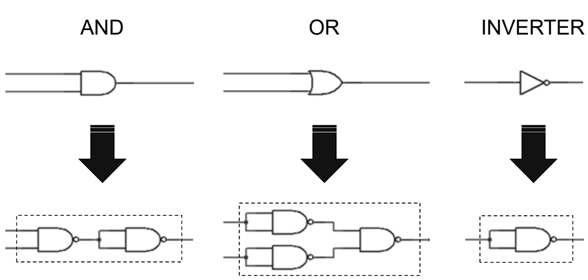


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Equivalent to OR Gate

76

NAND Gate Equivalent to AOI Gates

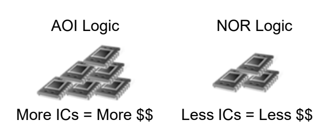


77

Universal Gate – NOR

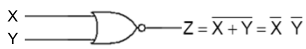
This presentation will demonstrate…

- The basic function of the **NOR** gate.
- How an **NOR** gate can be using to replace an **AND** gate, an **OR** gate or an **INVERTER** gate.
- How a logic circuit implemented with **AOI** logic gates could be re-implemented using only **NOR** gates
- That using a single gate type, in this case **NOR**, will reduce the number of integrated circuits (IC) required to implement a logic circuit.



78

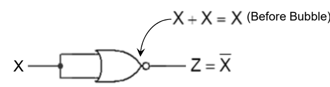
NOR Gate



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

79

NOR Gate as an Inverter Gate

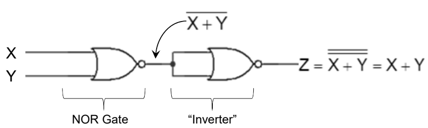


X	Z
0	1
1	0

Equivalent to Inverter

80

NOR Gate as an OR Gate

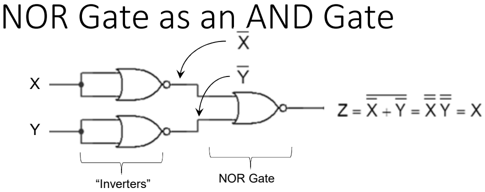


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Equivalent to OR Gate

81

NOR Gate as an AND Gate



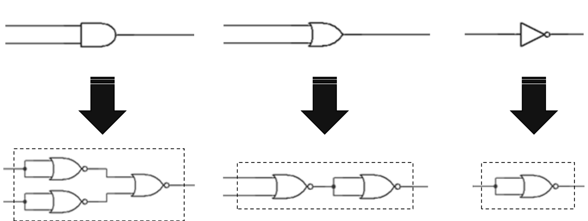
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Equivalent to AND Gate

82

NOR Gate Equivalent of AOI Gates

AND OR INVERTER



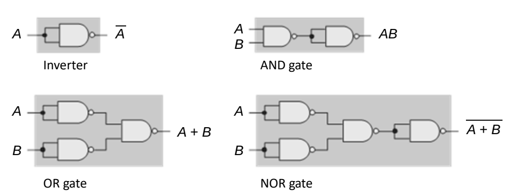
83

Logic Gates

Abc…

Universal Gates

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



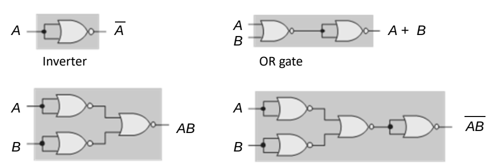
84

Logic Gates

Abc…

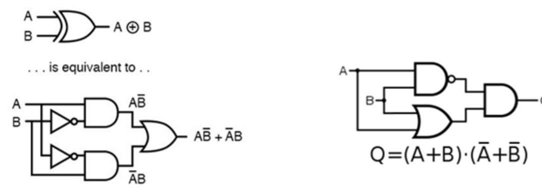
Universal Gates

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



85

XOR Gate with NAND Gate



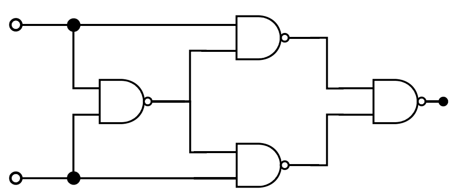
$A \oplus B = A\overline{B} + \overline{A}B$

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

$$X = A \oplus B$$

86

XOR Gate with NAND Gate

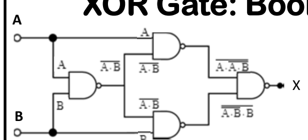


$$X = A\cdot\overline{B} + \overline{A}\cdot B$$

$$X = A \oplus B$$

87

XOR Gate: Boolean Expression



$$X = \overline{\left(\overline{A\cdot\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}\cdot B}\right)}$$

$$X = \overline{\left(\overline{A\cdot\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}\cdot B}\right)}$$

$$X = \left(\overline{\overline{A\cdot\overline{A\cdot B}}}\right) + \left(\overline{\overline{\overline{A\cdot B}\cdot B}}\right)$$

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

$$X = A\cdot\left(\overline{A} + \overline{B}\right) + \left(\overline{A} + \overline{B}\right)\cdot B$$

$$X = A\cdot\left(\overline{A} + \overline{B}\right) + \left(\overline{A} + \overline{B}\right)\cdot B$$

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

Now,

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

$$X = 0 + A\cdot\overline{B} + \overline{A}\cdot B + 0$$

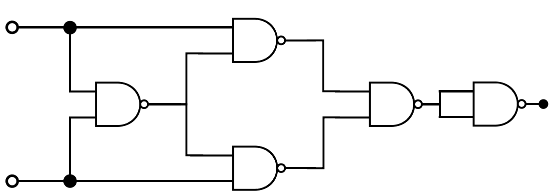
$$X = 0 + 0 + A\cdot\overline{B} + \overline{A}\cdot B$$

$$X = 0 + A\cdot\overline{B} + \overline{A}\cdot B$$

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

88

XNOR Gate with NAND Gate



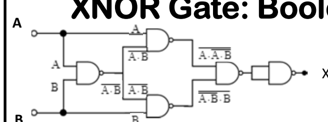
$$X = A\cdot B + \overline{A}\cdot\overline{B}$$

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

$$X = A \odot B$$

89

XNOR Gate: Boolean Expression



$$X = \overline{\overline{\left(\overline{A\cdot\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}\cdot B}\right)}}$$

$$X = \overline{\overline{\left(\overline{A\cdot\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}\cdot B}\right)}}$$

$$X = \left(\overline{A\cdot\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}\cdot B}\right)$$

$$X = \left(\overline{A} + \overline{\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}} + \overline{B}\right)$$

$$X = \left(\overline{A} + \overline{\overline{A\cdot B}}\right)\cdot\left(\overline{\overline{A\cdot B}} + \overline{B}\right)$$

$$X = \left(\overline{A} + A\cdot B\right)\cdot\left(A\cdot B + \overline{B}\right)$$

$$X = \left(\overline{A} + A\cdot B\right)\cdot\left(A\cdot B + \overline{B}\right)$$

Now,

$$X = \overline{A}\cdot(A\cdot B) + \overline{A}\cdot\overline{B} + (A\cdot B)\cdot(A\cdot B) + (A\cdot B)\cdot\overline{B}$$

$$X = \left(\overline{A}\cdot A\right)\cdot B + \overline{A}\cdot\overline{B} + (A\cdot A)\cdot(B\cdot B) + A\cdot\left(B\cdot\overline{B}\right)$$

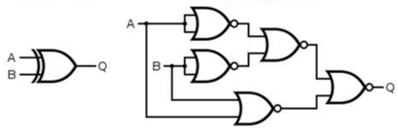
$$X = 0\cdot B + \overline{A}\cdot\overline{B} + (A)\cdot(B) + A\cdot 0$$

$$X = 0 + \overline{A}\cdot\overline{B} + A\cdot B + 0$$

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

90

XOR Gate with NOR Gate



Truth Table

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

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

$$X = A \oplus B$$

91

Logic Gate Applications

LOGIC GATE APPLICATIONS

AND GATE APPLICATION :

In a simple application, an AND gate can be used to detect the existence of a specified number of conditions and, in response, to activate an appropriate action.

For example, an automobile's safety system requires that an audible signal must be produced to warn the driver when the seat belt is not engaged. The conditions for this are :

i. That the ignition switch is on.

ii. That the seat belt is unbuckled.

iii. That the warning signal should last for a specified time and then turn off automatically.

The first two conditions can be represented by switches and the third by a timer circuit.

Fig.20 shows an AND gate whose high output activates a buzzer when these three conditions are met on its inputs.

Logic Gate Applications

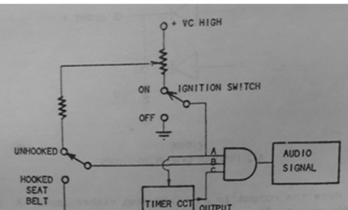


FIGURE 20

EXAMPLE OF AN AND GATE APPLICATION

i. When the ignition switch is closed (i.e.ON) a HIGH is connected to the gate input A.

ii. When the belt is not buckled, a HIGH is connected to the gate input B.

iii. At the instant the ignition switch is closed, the timer is activated and produces a HIGH on gate input C.

iv. The resulting HIGH gate output activates the alarm.

v. After a specified time the output of timer circuit goes LOW. This disables the AND gate and turns of the alarm.

Logic Gate Applications

OR GATE APPLICATION :

Let us assume that in a room with three doors an indicator lamp must be turned on, when any of the door is not closed.

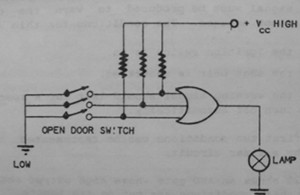
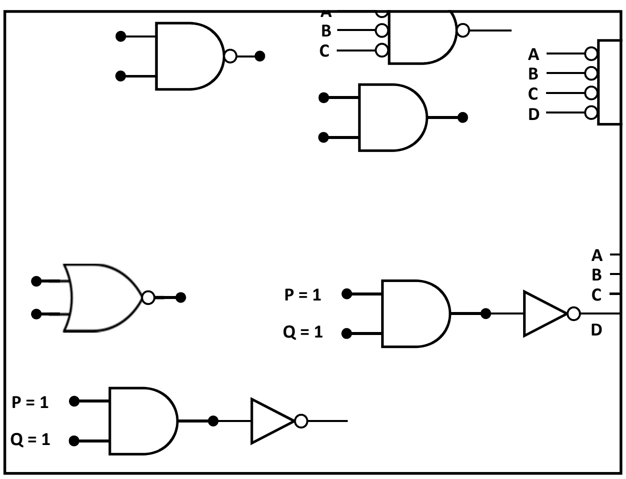


FIGURE 21

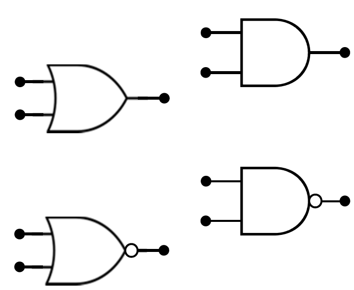
EXAMPLE OF AN OR GATE APPLICATION

The sensors are switches which are open when door is open.

This open switch creates HIGH level for the OR gate input as shown in Fig.21. If ANY or ALL of the doors are open, the gate output is HIGH. This HIGH level is then used to illuminate the indicator lamp.



Digital Circuit from Truth Table

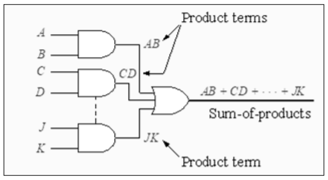


Combinational Logic

Abc…

Combinational Logic Circuits

In Sum-of-Products (SOP) form, basic combinational circuits can be directly implemented with AND-OR combinations if the necessary complement terms are available.



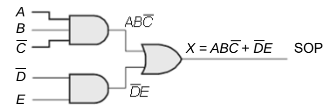
97

Logic Gates

Abc…

Combinational Logic Circuits

An example of an SOP implementation is shown. The SOP expression is an AND-OR combination of the input variables and the appropriate complements.



98

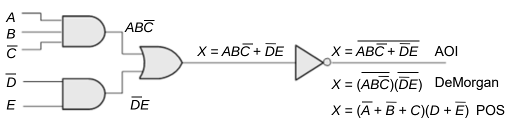
Logic Gates

Abc…

Combinational Logic Circuits

When the output of a SOP form is inverted, the circuit is called an AND-OR-Invert circuit. The AOI configuration lends itself to product-of-sums (POS) implementation.

An example of an AOI implementation is shown. The output expression can be changed to a POS expression by applying DeMorgan's theorem twice.



99

Logic Gates

Abc…

Exclusive-OR Logic

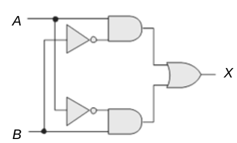
The truth table for an exclusive-OR gate is

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* disagree.

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

The circuit can be drawn as



Symbols:

Distinctive shape   Rectangular outline

100

Logic Gates

Abc…

Exclusive-NOR Logic

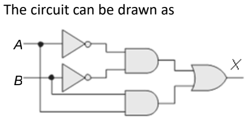
The truth table for an exclusive-NOR gate is

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* agree.

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

The circuit can be drawn as



Symbols:

Distinctive shape   Rectangular outline

101

Logic Gates

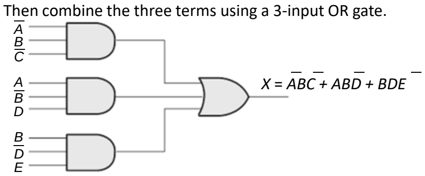
Abc…

Implementing Combinational Logic

Implementing a SOP expression is done by first forming the AND terms; then the terms are ORed together.

Example Show the circuit that will implement the Boolean expression $X = \overline{A}B\overline{C} + A\overline{B}D + B\overline{D}E$. (Assume that the variables and their complements are available.)

Solution Start by forming the terms using three 3-input AND gates. Then combine the three terms using a 3-input OR gate.



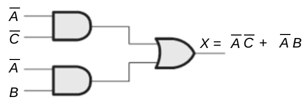
102

Logic Gates

Abc…

Solution *continued…*

Circuit:



The result is shown as a sum of products.

It is a simple matter to implement this form using only NAND gates as shown in the text and following example.

103

Logic Gates

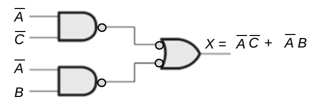
Abc…

NAND Logic

Example Convert the circuit in the previous example to one that uses only NAND gates.

Solution

Recall from Boolean algebra that double inversion cancels. By adding inverting bubbles to above circuit, it is easily converted to NAND gates:



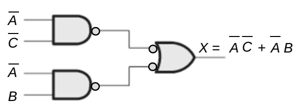
104

Logic Gates

Abc…

NAND Logic

Recall from DeMorgan's theorem that $\overline{AB} = \overline{A} + \overline{B}$. By using equivalent symbols, it is simpler to read the logic of SOP forms. The earlier example shows the idea:



The logic is easy to read if you (mentally) cancel the two connected bubbles on a line.

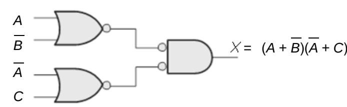
105

Logic Gates

Abc…

NOR Logic

Alternatively, DeMorgan's theorem can be written as $\overline{A + B} = \overline{A}\,\overline{B}$. By using equivalent symbols, it is simpler to read the logic of POS forms. For example,



Again, the logic is easy to read if you cancel the two connected bubbles on a line.

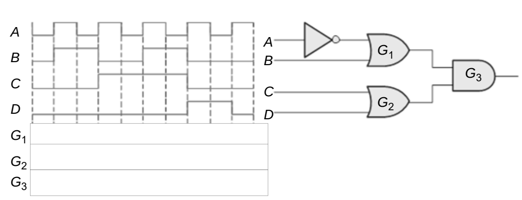
106

Logic Gates

Abc…

Pulsed Waveforms

For combinational circuits with pulsed inputs, the output can be predicted by developing intermediate outputs and combining the result. For example, the circuit shown can be analyzed at the outputs of the OR gates:



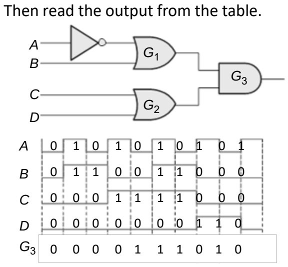
107

Logic Gates

Abc…

Pulsed Waveforms

Alternatively, you can develop the truth table for the circuit and enter 0's and 1's on the waveforms. Then read the output from the table.

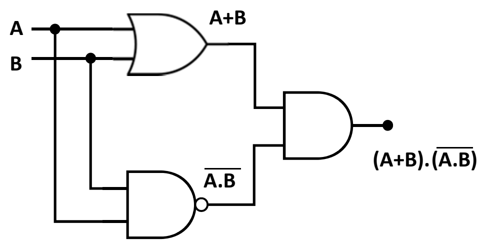


Inputs				Output
A	*B*	*C*	*D*	*X*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

108

Truth Table from Digital Circuit

• Write down the truth table indicating intermediate variables

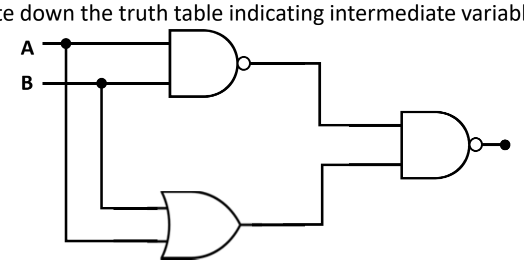


A	B	A+B	$\overline{A.B}$	$(A+B).(\overline{A.B})$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

?

Truth Table from Digital Circuit

• Write down the truth table indicating intermediate variables



A	B	A.B	$\overline{A.B}$	A+B	$(\overline{A.B}).(A+B)$	$\overline{(A.B).(A+B)}$
0	0	0	1	0	0	0
0	1	0	1	1	1	1
1	0	0	1	1	1	1
1	1	1	0	1	0	0

?

Truth Table from Digital Circuit