Digital Electronics1 4th Sem (GE4T) for Honors students

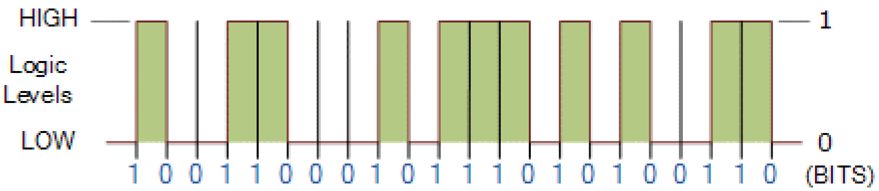
Mahadeb Pal, Assistant Professor and HOD, Physics.
SMHGC.

Difference between Analog Circuits Digital Circuits

- Analog circuits operate on continuously variable signals also known as Analog Signals. Digital Circuits operate on discretely variable signals or Digital signals i.e. the signal exists only in two levels: 0 and 1 (binary digital signalling).
- **Depending the efficiency and precision, it is quite difficult to design Analog Circuits. Digital Circuits are relatively easy to design with many automated tools available for various stages of design and analysis.**
- When interacting with the physical world, analog circuits can directly accept the signals from outside as the data is already analog. If a digital circuit has to acquire data from physical world, the analog signals must be converted to digital signals first.
- As there is no need for data conversion, there is ideally no loss of information. During the process of converting analog signals to digital signals, there might a significant amount of data loss, which can result in loss of information.
- If precision and accuracy are not a criterion, then analog circuits can be simple and inexpensive. Even with simple design techniques and at low cost, the digital circuits can provide good accuracy and precision.
- Due to the lack of skilled engineers and the complexity of the designs, analog circuits can turnout to be quite expensive. Advanced Integrated Circuits technologies and many other factors help the digital circuits to be reliable, lower in cost and smaller in size.

Binary Numbers

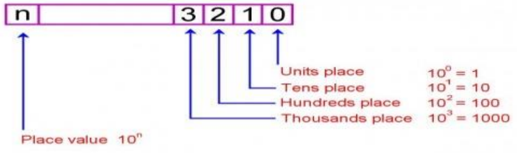
Binary Numbers are the flow of information in the form of zeros and ones used by digital computers and systems.



Generally, a logic “1” represents a higher voltage, such as 5 volts, which is commonly referred to as a HIGH value, while a logic “0” represents a low voltage, such as 0 volts or ground, and is commonly referred to as a LOW value. These two discrete voltage levels representing the digital values of “1’s” (one’s) and “0’s” (zero’s) are commonly called: **BI**nary digi**TS**, and in digital and computational circuits and applications they are normally referred to as binary **BITS**.

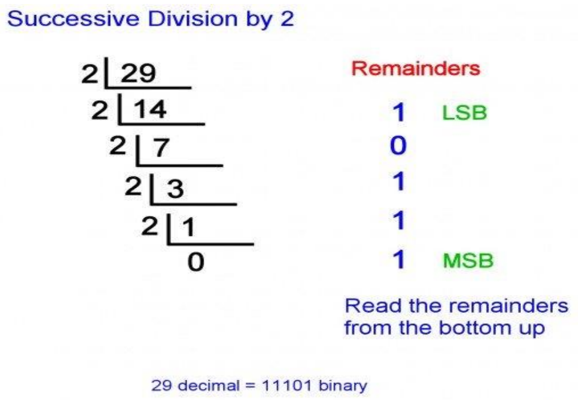
The *binary number system* is a Base-2 numbering system which follows the same set of rules in mathematics as the commonly used decimal or base-10 number system. So instead of powers of ten, ($\mathbf{10^n}$) for example: 1, 10, 100, 1000 etc, binary numbers use powers of two, ($\mathbf{2^n}$) effectively doubling the value of each successive bit as it goes, for example: 1, 2, 4, 8, 16, 32 etc.

Decimal to Binary

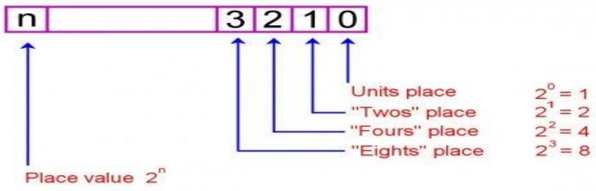


Steps to Convert from Decimal to Binary

- If you don't have a calculator to hand, you can easily convert a decimal number to binary using the remainder method. This involves dividing the number by 2 recursively until you're left with 0, while taking note of each remainder.
- Write down the decimal number.
- Divide the number by 2.
- Write the result underneath.
- Write the remainder on the right hand side. This will be 0 or 1.
- Divide the result of the division by 2 and again write down the remainder.
- Continue dividing and writing down remainders until the result of the division is 0.
- The most significant bit (MSB) is at the bottom of the column of remainders and the least significant bit (LSB) is at the top.
- Read the series of 1s and 0s on the right from the bottom up. This is the binary equivalent of the decimal number.
- What is 29 in binary?

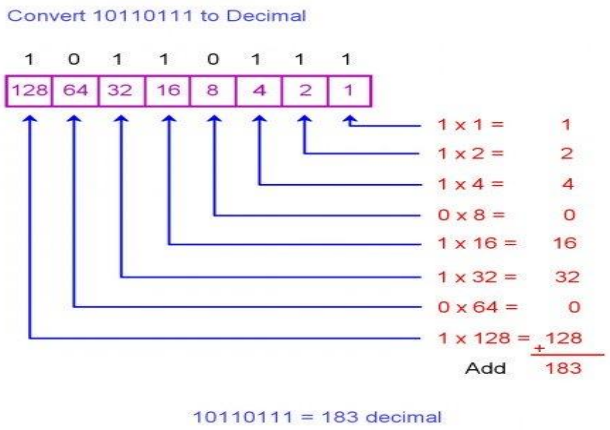


Binary to decimal



Steps to Convert Binary to Decimal

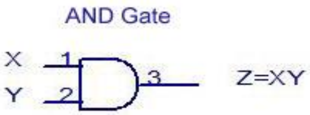
- Converting from binary to decimal involves multiplying the value of each digit (i.e. 1 or 0) by the value of the placeholder in the number
- Write down the number.
- Starting with the LSB, multiply the digit by the value of the place holder.
- Continue doing this until you reach the MSB.
- Add the results together.



AND GATE

An AND gate requires two or more inputs and produce only one output. The AND gate produces an output of logic 1 state when each of the inputs are at logic 1 state and also produces an output of logic 0 state even if any of its inputs are at logic 0 state. The symbol for AND operation is ‘.’, or we use no symbol for representing. If the inputs are of X and Y, then the output can be expressed as Z=XY. The AND gate is so named because, if 0 is called “false” and 1 is called “true,” the gate performs in the same way as the logical “and” operator. The AND gate is also named as all or nothing gate. The logic symbols and truth tables of two-input and three-input AND gates are given below.

2 Input AND Gate



TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

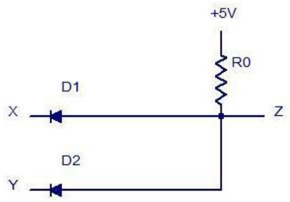
Discrete AND Gate

TRUTH TABLE

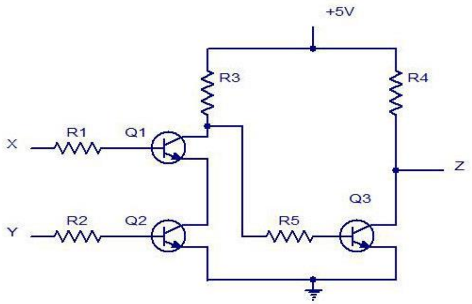
INPUTS		OUTPUT
X	Y	Z
0 V	0 V	0 V
0 V	5 V	0 V
5 V	0 V	0 V
5 V	5 V	5 V

Discrete AND gates may be realized by using diodes or transistors. The inputs represented as X and Y may be either 0V or +5V correspondingly. The output is represented by Z . In the diode of AND gate, when both the inputs are of same value, X=+5V and Y= +5V, then the diodes are in OFF condition. As a result, no current flows through the resistor and there will not be any voltage drop across the resistor. Here the output will be Z=+5V. Similarly, when both the inputs such as X and Y are equal to 0V, then the corresponding diodes such as either D1 or D2 or both the diodes are at ON state and act as short circuits. Here the output will be Z corresponds to 0V. In practical cases the output z corresponds to 0.6V or 0.7V, which is treated as logic 0 state.

2 Input Diode AND Gate



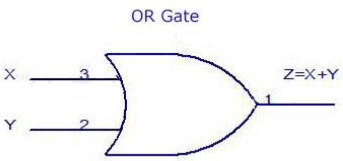
2 Input Transistor AND Gate



OR GATE

Similar to AND gate, an OR gate may also have two or more inputs but produce only one output. The OR gate produces an output of logic 1 state even if any of its inputs is in logic 1 state and also produces an output of logic 0 state if any of its inputs is in logic 0 state. The symbol for OR operation is ‘+’. If the inputs are of X and Y, then the output can be represented as Z=X+Y. An OR gate may also be defined as a device whose output is 1, even if one of its input is 1. OR gate is also called as any or all gate. It is also called as an inclusive OR gate because it consists of the condition of ‘both the inputs can be present’. The logic symbols and truth table for two-input and three-input OR gates are given below.

2 Input OR Gate

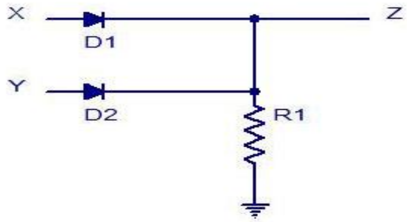


TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Discrete OR gates may be realized by using diodes or transistors. The inputs represented as X and Y may be either 0V or +5V correspondingly. The output is represented by Z . In the diode of OR gate, when both the inputs are of same value, X=0V and Y= 0V, then both the diodes are in OFF condition. As a result, no current flows through the resistor and there will not be any voltage drop across the resistor. Here the output will be Z=0V. Similarly, when both the inputs or either the inputs such as X and Y are equal to +5V, then the corresponding diodes either D1 or D2 or both the diodes are at ON state and act as short circuits. Here the output will be Z corresponds to +5V. In practical cases the output Z corresponds to +5V-diode drop = +5V – 0.7V = +4.3V, which is regarded as Logic 1 state.

2 Input Diode OR Gate



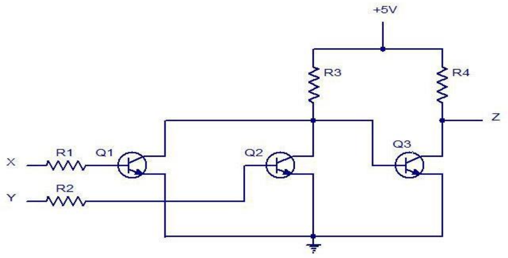
Discrete OR Gate

TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0 V	0 V	0 V
0 V	5 V	5 V
5 V	0 V	5 V
5 V	5 V	5 V

In the case of transistor OR gate, when the inputs X=0V and Y = 0V both the transistors Q1 and Q2 are at OFF state. At the same time, Transistor Q3 gets enough base drive from the supply +5V through Resistor R3 and so transistor Q3 will be ON. Thereby the output voltage Z= $V_{ce(sat)}$ corresponds to 0V. When either the inputs X and Y or both the inputs are equal to +5V, then the corresponding transistors either Q1 or Q2 will be ON or both the transistors Q1 and Q2 will be ON and therefore the voltage at the collector of transistor Q1 is $V_{CE(sat)}$ corresponds to 0V. Due to this reason the transistor Q3 doesn’t forward bias the base-emitter junction and turns OFF. So the final output voltage corresponds to +5V (logic 1 state). The truth table for this gate circuit is shown below:

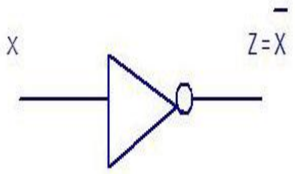
2 Input Transistor OR Gate

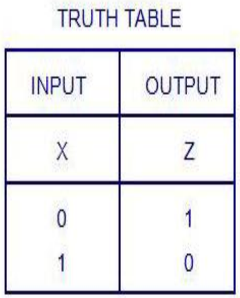


NOT GATE

The NOT gate is also called as an inverter, simply because it changes the input to its opposite. The NOT gate is having only one input and one corresponding output. It is a device whose output is always the compliment of the given input. That means, the NOT gate produces an output of logic 1 state when the input is of logic 0 state and also produce the output of logic 0 state when the input is of logic 1 state. The NOT operation is denoted by '-'(bar). When the input variable to the NOT gate is represented by 'X' and the output is represented by 'Z'. In the NOT operation it can be read as 'Z is equal to X bar'. The logic symbol and truth table are given below:

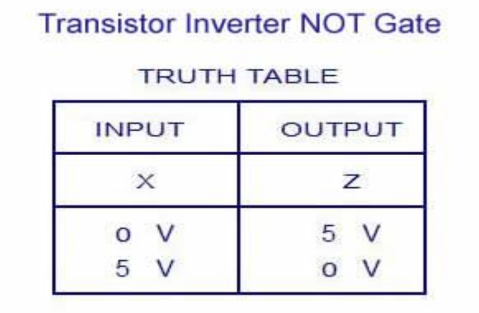
NOT Gate





TRUTH TABLE

INPUT	OUTPUT
X	Z
0 1	1 0



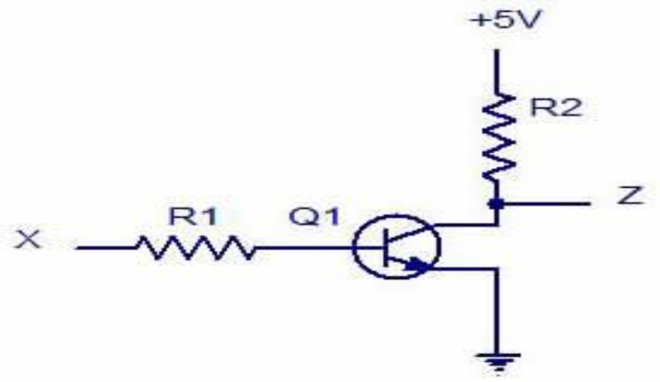
Transistor Inverter NOT Gate

TRUTH TABLE

INPUT	OUTPUT
X	Z
0 V 5 V	5 V 0 V

Discrete NOT gate may be realized by using transistors. The inputs represented as X may be either 0V or +5V correspondingly. The output is represented by Z. When the input X = 0V, then the transistor Q1 will be reverse biased and therefore it remains OFF. As a result no current flows through the resistor and thereby there will not be any voltage drop across the resistor. As a result, the output voltage Z corresponds to +5V. When the input X= +5V, transistor Q1 is ON and the output voltage Z=V $_{\text{ce(sat)}}$ corresponds to 0V. The truth table for the NOT gate is given below:

Transistor Inverter NOT Gate



Universal GATE

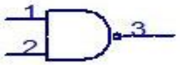
The NAND and NOR gates are the universal gates. Each of this gates can realize the logic circuits single handedly. The NAND and NOR are also called as universal building blocks. Both NAND and NOR has the ability to perform three basic logic functions such as AND,OR and NOT.

NAND GATE

NAND gate is a combination of an AND gate and a NOT gate. The expression for the NAND gate is ‘—‘whole bar. The output of the NAND gate is at logic 0 level only when each of the inputs assumes a logic 1 level. The truth table of two-input NAND gate is given below:

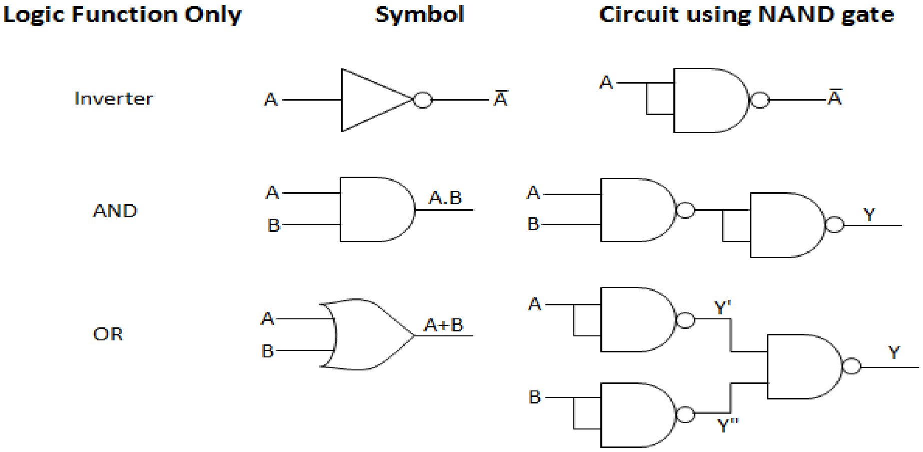
2 Input NAND Gate

NAND Gate



TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

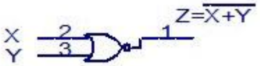


NOR GATE

NOR means NOT OR. That means, NOR gate is a combination of an OR gate and a NOT gate. The output is logic 1 level, only when each of its inputs assumes a logic 0 level. For any other combination of inputs, the output is a logic 0 level. The truth table of two-input NOR gate is given below:

2 Input NOR Gate

NOR Gate



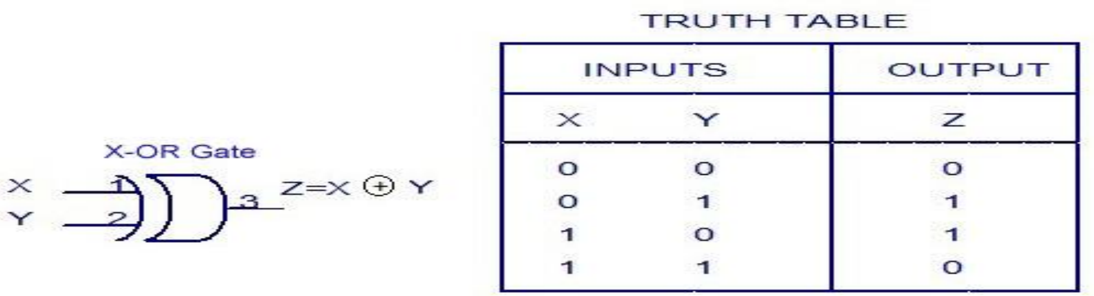
TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

EXCLUSIVE-OR GATE (X-OR) GATE

An X-OR gate is a two input, one output logic circuit. X-OR gate assumes logic 1 state when any of its two inputs assumes a logic 1 state. When both the inputs assume the logic 0 state or when both the inputs assume the logic 1 state, the output assumes a logic 0 state. The output of the X-OR gate will be the sum of the modulo sum of its inputs. X-OR gate is also termed as anti-coincidence gate or inequality detector. An X-OR gate can also be used as inverter by connecting one of the two input terminals to logic1 and also by inputting the sequence to be inverted to the other terminal.

X-OR Gate



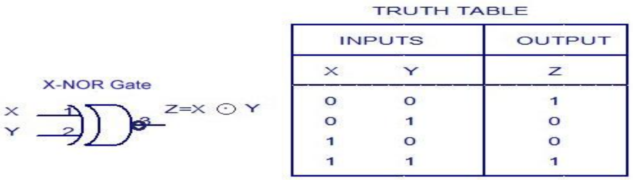
TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

EXCLUSIVE-NOR (X-NOR) GATE

An X-NOR gate is a combination of an X-OR gate and a NOT gate. The X-NOR gate is also a two input, one output concept. The output of the X-NOR gate will be logic 1 state when both the inputs assume a 0 state or when both the inputs assume a 1 state. The output of the X-NOR gate will be logic 0 state when one of the inputs assume a 0 state and the other a 1 state. It is also named as coincidence gate, because its output will be 1 only when the inputs coincide. X-NOR gate can also be used as inverter by connecting one of the two input terminals to logic 0 and also by inputting the sequence to be inverted to the other terminal.

X-NOR Gate

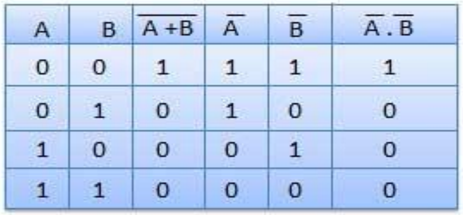


TRUTH TABLE

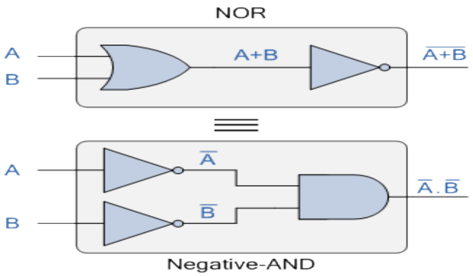
INPUTS		OUTPUT
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

DeMorgan's **First theorem** proves that when two (or more) input variables are OR'ed and negated, they are equivalent to the AND of the complements of the individual variables. Thus the equivalent of the NOR function and is a negative-AND function proving that $\overline{A+B} = \overline{A}.\overline{B}$ ……….(1) and again we can show this using the following truth table.

Truth table for verification



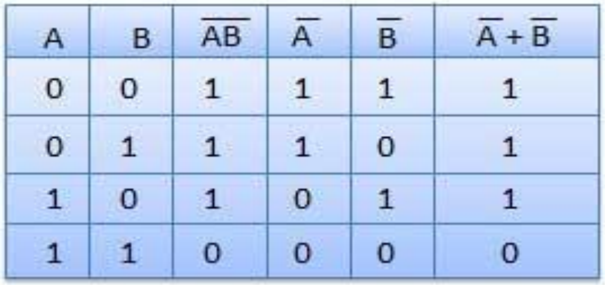
A	B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0



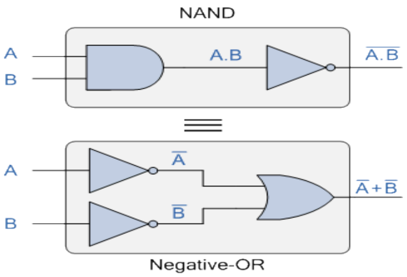
Implementation using Logic Gates

De Morgan's **Second theorem** proves that when two (or more) input variables are AND'ed and negated, they are equivalent to the OR of the complements of the individual variables. Thus the equivalent of the NAND function and is a negative-OR function proving that and we can show this $\overline{A.B} = \overline{A} + \overline{B}$ ……(2) using the following table.

Truth table for verification



A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0



Implementation using Logic Gates