Lab#1

To Demonstrate ETS- 5000 training board, Logicly Software, Logic gates & Verification of their Truth Tables

Equipment Required:

ETS-5000 training board

Logic gates (ICs)

Introduction:

Logic Gates:

A Digital Logic Gate is an electronic circuit which makes logical decisions based on the combination of digital signals present on its inputs. Digital logic gates can have more than one input, for example, inputs A, B, C, D etc., but generally only have one digital output, (Q). Individual logic gates can be connected or cascaded together to form a logic gate function with any desired number of inputs, or to form combinational and sequential type circuits, or to produce different logic gate functions from standard gates. Standard commercially available digital logic gates are available in two basic families or forms, TTL which stands for Transistor-Transistor Logic such as the 7400 series, and CMOS which stands for Complementary Metal-Oxide-Silicon which is the 4000 series of chips. This notation of TTL or CMOS refers to the logic technology used to manufacture the integrated circuit, (IC) or a “chip”.

Integrated Circuits or IC’s, can be grouped together into families according to the number of transistors or “gates” that they contain. Integrated circuits are categorized according to the number of logic gates or the complexity of the circuits within a single chip with the general classification for the number of individual gates given as:

Small Scale Integration or (SSI) – Contain up to 10 transistors or a few gates within a single package such as AND, OR, NOT gates.

Medium Scale Integration or (MSI) – between 10 and 100 transistors or tens of gates within a single package and perform digital operations such as adders, decoders, counters, flip-flops and multiplexers.

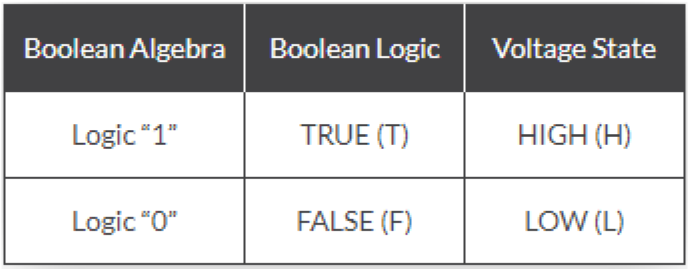
Large Scale Integration or (LSI) – between 100 and 1,000 transistors or hundreds of gates and perform specific digital operations such as I/O chips, memory, arithmetic and logic units.

Very-Large Scale Integration or (VLSI) – between 1,000 and 10,000 transistors or thousands of gates and perform computational operations such as processors, large memory arrays and programmable logic devices.

Super-Large Scale Integration or (SLSI) – between 10,000 and 100,000 transistors within a single package and perform computational operations such as microprocessor chips, micro-controllers, basic PICs and calculators.

Ultra-Large Scale Integration or (ULSI) – more than 1 million transistors – the big boys that are used in computers CPUs, GPUs, video processors, micro-controllers, FPGAs and complex PICs.

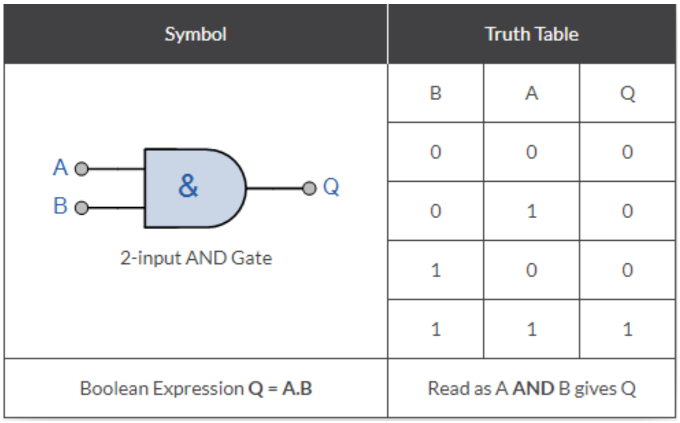
In digital logic design only two voltage levels or states are allowed and these states are generally referred to as Logic “1” and Logic “0”, or HIGH and LOW, or TRUE and FALSE. These two states are represented in Boolean Algebra and standard truth tables by the binary digits of “1” and “0” respectively. A good example of a digital state is a simple light switch. The switch can be either “ON” or “OFF”, one state or the other, but not both at the same time. Then we can summarize the relationship between these various digital states as being:



Boolean Algebra	Boolean Logic	Voltage State
Logic “1”	TRUE (T)	HIGH (H)
Logic “0”	FALSE (F)	LOW (L)

AND Gate

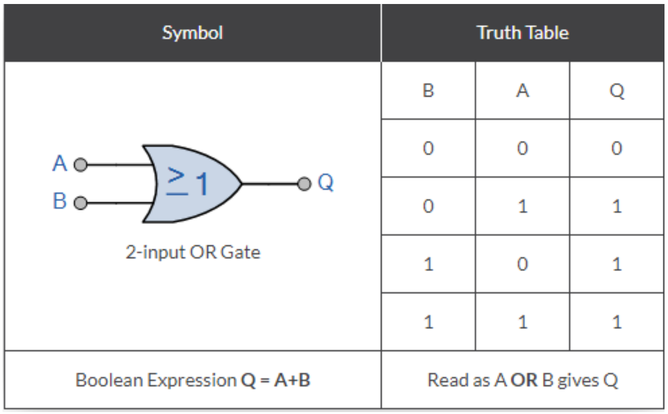
The Logic AND Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when all of its inputs are HIGH. The output state of a digital logic AND gate only returns “LOW” again when ANY of its inputs are at a logic level “0”. In other words, for a logic AND gate, any LOW input will give a LOW output. The logic or Boolean expression given for a digital logic AND gate is that for Logical Multiplication which is denoted by a single dot or full stop symbol, (.) giving us the Boolean expression of: A.B = Q.



Symbol	Truth Table		
	B	A	Q
2-input AND Gate	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression **Q = A.B**	Read as A **AND** B gives Q		

OR Gate

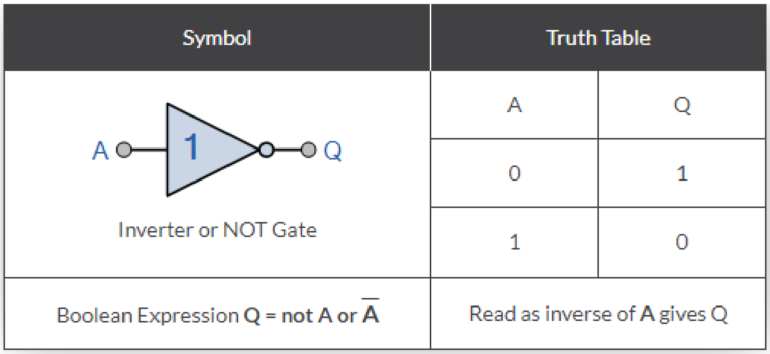
The Logic OR Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when one or more of its inputs are HIGH. The output, Q of a “Logic OR Gate” only returns “LOW” again when ALL of its inputs are at a logic level “0”. In other words, for a logic OR gate, any “HIGH” input will give a “HIGH”, logic level “1” output. The logic or Boolean expression given for a digital logic OR gate is that for Logical Addition which is denoted by a plus sign, (+) giving us the Boolean expression of: A+B = Q.



Symbol	Truth Table		
2-input OR Gate	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression **Q = A+B**	Read as A **OR** B gives Q		

NOT Gate

The Logic NOT Gate is the most basic of all the logical gates and is often referred to as an Inverting Buffer or simply an Inverter. Inverting NOT gates are single input devices which have an output level that is normally at logic level “1” and goes “LOW” to a logic level “0” when its single input is at logic level “1”, in other words it “inverts” (complements) its input signal. The output from a NOT gate only returns “HIGH” again when its input is at logic level “0” giving us the Boolean expression of: A = Q’.

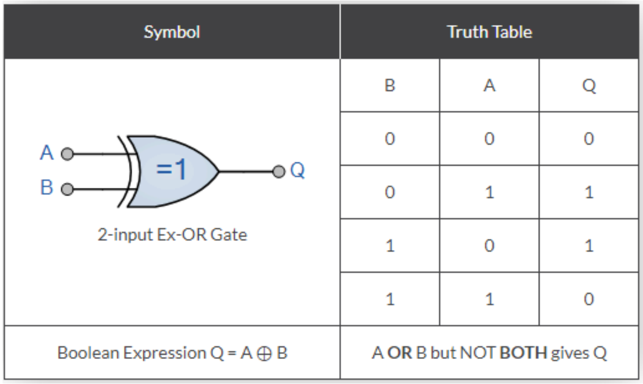


Symbol	Truth Table	
Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression **Q = not A or $\overline{A}$**	Read as inverse of **A** gives Q	

Exclusive-OR Gate

Previously, we saw that for a 2-input OR gate, if A = “1”, OR B = “1”, OR BOTH A + B = “1” then the output from the digital gate must also be at a logic level “1” and because of this, this type of logic gate is known as an Inclusive-OR function. The logic gate gets its name from the fact that it includes the case of Q = “1” when both A and B = “1”. If however, an logic output “1” is obtained when ONLY A = “1” or when ONLY B = “1” but NOT both together at the same time, giving the binary inputs of “01” or “10”, then the output will be “1”. This type of gate is known as an Exclusive-OR function or more commonly an Ex-Or function for short. This is because its boolean expression excludes the “OR BOTH” case of Q = “1” when both A and B = “1”.

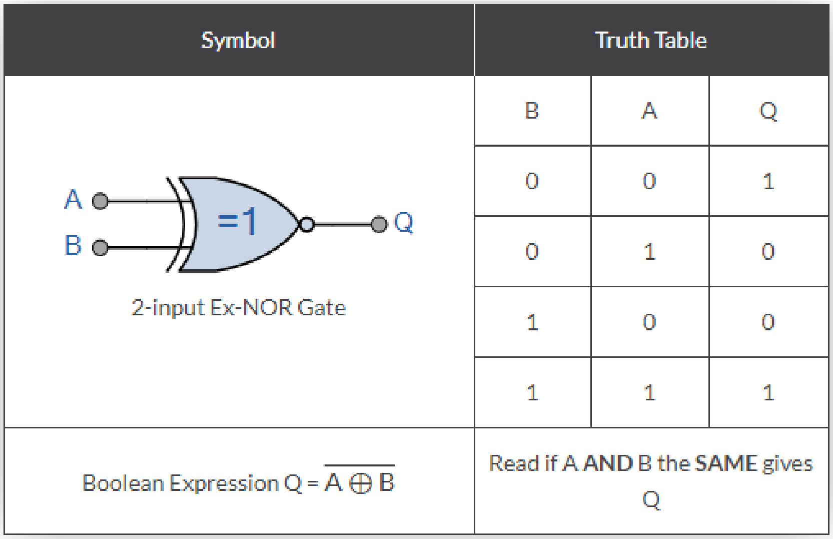
In other words the output of an Exclusive-OR gate ONLY goes “HIGH” when its two input terminals are at “DIFFERENT” logic levels with respect to each other. An odd number of logic “1’s” on its inputs gives a logic “1” at the output. These two inputs can be at logic level “1” or at logic level “0” giving us the Boolean expression of: $Q = (A \oplus B) = A'.B + A.B'$



Symbol	Truth Table		
	B	A	Q
2-input Ex-OR Gate	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression Q = A ⊕ B	A **OR** B but NOT **BOTH** gives Q		

Exclusive-NOR Gate

The Exclusive-NOR Gate function is a digital logic gate that is the reverse or complementary form of the Exclusive-OR function. Basically the “Exclusive-NOR” gate is a combination of the Exclusive-OR gate and the NOT gate but has a truth table similar to the standard NOR gate in that it has an output that is normally at logic level “1” and goes “LOW” to logic level “0” when ANY of its inputs are at logic level “1”. However, an output “1” is only obtained if BOTH of its inputs are at the same logic level, either binary “1” or “0”. For example, “00” or “11”. This input combination would then give us the Boolean expression of: $Q = (A \oplus B)' = A'.B' + A.B$. Then the output of a digital logic Exclusive-NOR gate ONLY goes “HIGH” when its two input terminals, A and B are at the “SAME” logic level which can be either at a logic level “1” or at a logic level “0”. In other words, an even number of logic “1’s” on its inputs gives a logic “1” at the output, otherwise is at logic level “0”. Then this type of gate gives and output “1” when its inputs are “logically equal” or “equivalent” to each other, which is why an Exclusive-NOR gate is sometimes called an Equivalence Gate. The logic symbol for an Exclusive-NOR gate is simply an Exclusive-OR gate with a circle or “inversion bubble”, (o) at its output to represent the NOT function. Then the Logic Exclusive-NOR Gate is the reverse or “Complementary” form of the Exclusive-OR gate, $(A \oplus B)$ we have seen previously.



Symbol	Truth Table		
2-input Ex-NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression Q = $\overline{A \oplus B}$	Read if A **AND** B the **SAME** gives Q		