# **Experiment 1**

# **Introduction to Logic Gates**

**Objective**

* To practice the operation of the basic gates (AND, OR and NOT).
* To verify the basics of multiple input gates and propagation delay.
* Other gates such as NAND, NOR, wired logic will also be examined.
* How we can use any type of gates to replace any other type.

**Basic concepts**

Binary logic consists of binary variables and logical operations. Each variable can have only two distinct possible values: “1” and “0”. The basic logical operations are AND, OR and NOT. These operations are represented by symbols, each symbol has input and output variables. For each combination of the values of input variables, there is a value of output specified by the definition of the logical operation. These definitions may be listed in a compact form using truth tables. A truth table is a table of all possible combinations of the variables showing the relation between the values that the input variable may take and the result of operation.

The three basic logic operations are as follows:

* ***AND:***

This operation is represented by a dot or by the absence of an operator, e.g. X.Y=Z or XY=Z, it is read "X AND Y is equal to Z". This operation means that Z=1 if and only if X=1 and Y=1; otherwise Z=0.

* ***OR:***

This operation is represented by a plus sign e.g. X+Y=Z, it is read "X OR Y is equal to Z". This operation means that Z=0 if and only if X=0 and Y=0; otherwise Z = 1.

* ***NOT:***

This operation is represented by prime or a bar, e.g. X'=Z or X = Z, it reads "X NOT equal to Z", meaning that if X=1 then Z=0 but if X=0 then Z=1.

The NAND gate is an AND gate with inverted output. The NOR gate is an OR gate with a inverted output. All gates, except the inverter (NOT) can have multiple inputs (more than 2) with the same logic operations described above.

The characteristics of IC digital logic families are usually compared by analyzing the circuit of the basic gate in each family. The most important parameters that are evaluated and compared are propagation delay, fan-out, power dissipation and noise margin. These parameters are explained as follows:

* Propagation delay is the time required for the output of a gate to change after one of its inputs has changed. If several gates are arranged in series their propagation delays are additive. There are two kinds of propagation delays:

 TPLH = Propagation delay time when the output changes from low to high.

 TPHL = Propagation delay time when the output changes from high to low.

The propagation delay of a gate is usually taken as the maximum ofT PLH and T PHL. The typical propagation delay for standard TTL gate is 10 ns.

* Fan-out specifies the number of standard loads that the output of a gate can drive with normal operation, sometimes it is referred to as loading. For standard TTL family typically the fan-out is considered to be a maximum of 10, this means that the output of a gate can be connected to a maximum of 10 input gates for correct operation.
* Power dissipation of a gate is the supplied power required to operate gate. This parameter is expressed in mw and represents the actual power dissipated in the gate.
* Noise margin is the maximum noise voltage that can be added to the input signal of a digital circuit that would not cause a change in the circuit output.

**Procedure:**

***Verification of the Logic Lab board Operation:***

***Remember that the following testing procedure (step 1 to 8) should be done for the lab board by the start of every lab session to make sure that all parts of the board are OK before starting to implement your experiment just to eliminate error sources and have a successful implementation for your design in an easy way.***

1. Verify proper line voltage setting (110v / 220v).
2. Connect the Logic Lab to the appropriate power source.
3. Set the Logic Lab LINE switch to ON.
4. Wire one data switch output connector (SW1- SW8) to one of the LED input connectors (L1-L8).
5. Verify HIGH and LOW operation of the data switch by first setting the data switch to LOW and noting the indicator illumination level.

Then setting the data switch to HIGH and noting the indicator illumination level. The six independent LED indicators are illuminated for a logic HIGH level and extinguished for a logic LOW level.

1. Repeat step 5 for all other data switches and indicators.
2. Adjust your board clock to 1 K Hz and then wire it to one of the LED indicator inputs connectors (L1-L8). Note the output indication.
3. Adjust your board clock to 10 kHz and then wire it to one of the LED indicator input connectors.

***Note:***

The 10 kHz signal is too fast for the human eye to distinguish the HIGH and LOW Output indicator. The LED will appear to dimly illuminate.

### The AND gate:

1. Study the schematic diagram of I C 7408 (quad, four in one package,

2-input AND gate) from the TTL sheets in the lab.

1. Insert a 7408 into your board and connect pin 14 to +5V, pin 7 to common (GND).
2. Connect SW1 to pin 1 and SW2 to pin2. Connect pin3 to L1.
3. Set the data switches (SW1 and SW2) to test all different input combinations and observe the output (L1). Record the results in the form of a truth table.
4. Construct a two level 4-input AND gate which implements the function F= ABC using 2-input AND gates. Verify the output of your design and record it in the form of a truth table.
5. Design the two level 4-input using workbench in a gate format and also in a chip format.
6. Obtain the truth table of the two level4-input AND using workbench.
7. Using the Algebraic rules, obtain another design for the 4-input AND gate using any other gates, get EWB simulation for it.
8. Compare the results of step 8 & step7 to verify that your design of step 8 is correct because it verifies the same truth table as the three inputs AND gate.

### The NOT gate:

1. Study the schematic diagram of I C 7404 (hex, six in one package,

1-input NOT gate) from the sheets in the lab.

1. Insert a 7404 into the logic breadboard.
2. Connect pin 1 to SW1 and pin 2 to L1. Observe the output for all possible combinations of the input. Record your results in the form of a truth table.
3. Construct a series of six NOT gates by connecting the output of one inverter to the input of another (the first gate's input should not be connected to another gate as well as the last gate's output).
4. Connect the input of the first gate to the 1k Hz signal on your board. Connect the output from the second gate to L1. Connect the output from the fourth gate to L2. Connect the output from the sixth gate to L3. Watch L1, L2 and L3.

What do you notice? Are you able to deduce the propagation delay for all the six inverter, and hence compute it for one gate?

1. Design the above part using workbench in a gate format and also in a chip format.

### NAND gate:

1. Repeat the procedure in A- 1 to 7 above using the 7400 IC instead of the 7408 IC.
2. Does the truth table resemble a 4-input NAND gate? If it does not,

correct your design to represent the 4-input NAND gate, then simulate your design using EWB and make sure that it verifies the 4-input NAND gate truth table using EWB.

1. Using the Algebraic rules, obtain another design for the 4-input NAND gate using any other gates, then simulate your design using EWB and make sure that it verifies the three input NAND gate truth table using EWB.
2. Compare the results of step 3 & corrected design of step2 to verify that your design of step 3 is correct because it verifies the same truth table as the 4-input NAND gate.

### NOR gate:

1. Repeat the procedure in A- 1 to 7 above using the 7402 IC instead of the 7408 IC.
2. Does the truth table resembles a 4-input NOR gate?

If it does not, correct your design to represent the 4-input NOR gate, then simulate your design using EWB and make sure that it verifies the 4-input NOR gate truth table using EWB.

1. Using the Algebraic rules, obtain another design for the 4-input NOR gate using any other gates, then simulate your design using EWB and make sure that it verifies the 4-input NOR gate truth table using EWB.
2. Compare the results of step 3 & corrected design of step2 to verify that your design of step 3 is correct because it verifies the same truth table as the 4-input NOR gate.

###  Wired Logic:

The wired AND gate is a condition where the output of two or more gates are directly connected to each other. In a wired AND configuration both inputs must be HIGH to yield a HIGH output.

In a wired OR configuration both inputs must be LOW to yield a LOW output. Although wired logic circuit configuration eliminates additional circuitry, it is not good practice (except for specified gate called open-collector which requires external pull-up (connected to Vcc) or pull-down (connected to Ground) resistor for two reasons:

1. In standard TTL 74XX series logic, only 2 outputs can be tied in the wired AND configuration.
2. A gate can appear to give an enormous output which makes circuit troubleshooting externally difficult.
3. Connect two NOR gate outputs (on the same 7402) together and then connect the common point to L1.
4. Connect SW1 through SW4 to the inputs of the two NOR gates.
5. Set SW1 to SW4 so that SW1=SW2=SW3= 0 and SW4 = 1.
6. Note the output at L1. Is it what you would expect?