Experiment No 1

BASIC LOGIC GATES

Before this experiment students are expected to:

1. Make a revision of digital logic gate, truth table and how to draw electrical wiring diagram.
2. Search about digital logic in general and write down some practical devices and usage of such gates.

Objectives:

- Understand the function of each logic gate and how it being used in the industrial automation field.
- Be able to draw the electrical wiring diagram and construct the hardware connections.

Introduction:

Automation is the use of control systems (such as numerical control, programmable logic control, and other application information technology (such as computer aided technology) to control industrial machinery and process, in order to reduce the need of human's intervention.

There are many advantages of using automation in industrial, some of those advantages are:

1- Replace human operations in tedious task.
2- Replace human operations in dangerous environments (fire, nuclear facilities, under the water etc).
3- Make tasks that beyond human capability.
4- Economy improvements.

Theory:

A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions low (0) or high (1), represented by different voltage levels.

The logic state of a terminal can, and generally does, change often, as the circuit processes data. In most logic gates, the low state is approximately zero volts (0 V), while the high state is approximately five volts positive (+5 V).

There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR. In this experiment we will discuss and implement each of them,

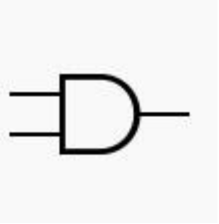
Basic Logic Gates:

AND gate:

The AND gate is so named because, if 0 is called "false" and 1 is called "true," the gate acts in the same way as the logical "and" operator. Figure 1 illustration and table show the circuit symbol and logic combinations for an AND gate. (In the symbol, the input terminals are at left and the output terminal is at right.) The output is "true" when both inputs are "true." Otherwise, the output is "false."

Truth table:

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1



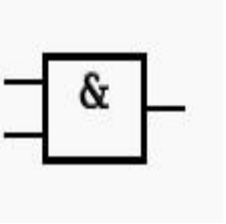


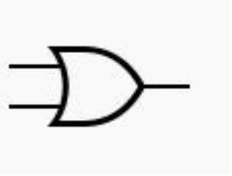
Figure (1): AND logic gate, a. Distinctive shape, b. Rectangular shape.

OR gate:

The OR gate gets its name from the fact that it behaves after the fashion of the logical inclusive "or." The output is "true" if either or both of the inputs are "true." If both inputs are "false," then the output is "false". See figure2.

Truth table:

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1



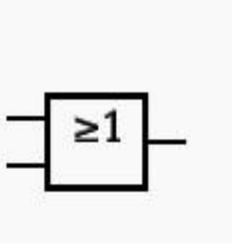


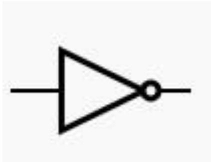
Figure (2): OR logic gate, a. Distinctive shape, b. Rectangular shape.

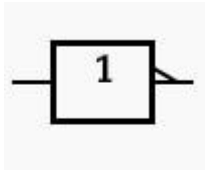
Inverter gate:

A logical inverter, sometimes called a NOT gate to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state shown in figure3.

Input 1	Output
0	1

1	0





Truth table:

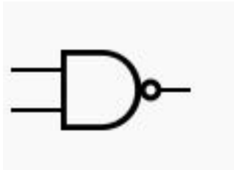
Figure (3): NOT logic gate, a. Distinctive shape, b. Rectangular shape.

NAND gate:

The NAND gate operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is "false" if both inputs are "true." Otherwise, the output is "true”. See figure4

Truth table:

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0



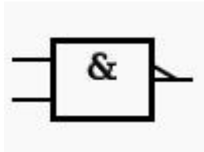


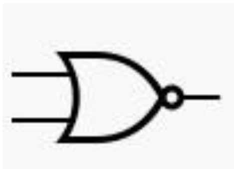
Figure (4): NAND logic gate, a. Distinctive shape, b. Rectangular shape.

NOR gate:

The NOR gate is a combination OR gate followed by an inverter. Its output is "true" if both inputs are "false." Otherwise, the output is "false." See figure5

Truth table:

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0



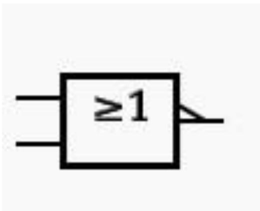


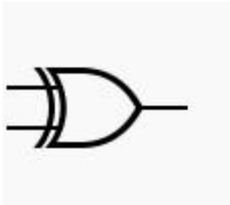
Figure (5): NOR logic gate, a. Distinctive shape, b. rectangular shape.

XOR gate:

The XOR (exclusive-OR) gate acts in the same way as the logical "either/or". The output is "true" if either, but not both, of the inputs are "true". The output is "false" if both inputs are "false" or if both inputs are "true". Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same. Figure 6 illustrate the truth table and symbol.

Truth table:

Input 1	**Input 2**	**Output**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**



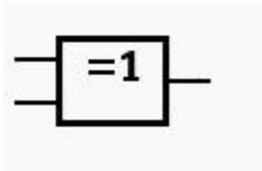


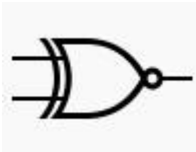
Figure (6): XOR logic gate, a. Distinctive shape, b. Rectangular shape.

XNOR gate:

The XNOR (exclusive-NOR) gate is a combination XOR gate followed by an inverter. Its output is "true" if the inputs are the same and “false" if the inputs are different. See figure 7.

Truth table:

Input 1	**Input 2**	**Output**
0	**0**	**1**
0	**1**	**0**
1	**0**	**0**
1	**1**	**1**



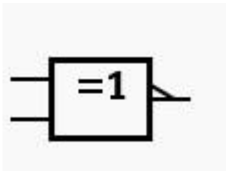


Figure (7): XNOR logic gate, a. Distinctive shape, b. Rectangular shape.

Component and apparatus:

The components required for the experiment are:

- **Powers supply 24 VDC.**
- **Electrical LEDs**
- **Bush button kit (PB “N/O,N/C”).**
- **Electrical relay kit.**
- **Wires.**

Procedure:

Part 1:

For each logic gate

1. Draw the wiring diagram representing the inputs as switches and output as LEDs.
2. Construct the drawn connection of yours.

Part 2:

Construct the connections to achieve the followings:

A- When push button PB1 is activated and push button PB2 (manual stop) is deactivated, LED1 turns on.

B- When PB1 or PB2 or both are activated and PB3 (manual stop) deactivated, then LED1 turns on, else LED1 must be off.

C- When PB1 and PB2 are activated and PB3 (manual stop) is deactivated, then LED1 turns on, and LED2 (emergency indicator) turns off. but at the moment the manual stop being activated (PB3) , LED1 and LED2 will be toggled**.**

Part 3:

Application in industrial field and automation: Entrance Control Systems (access control system): In the entrance and exit of parking, there is a sensor before the barriers to detect movement of cars and automatically control the barriers.

- Draw electrical wiring diagram for such system in order to detect any car and opened the barrier then closed it.
- Make a connection in order to build such system use LEDs to simulate the barrier and switch for sensor.

Figure (8): barrier.

Hint: you need two sensors, before and after the barrier to ensure that the car is completely lift.

In the report:

Part 2:

A- The following Truth table must be filled

PB1	PB2	LED1
0	0	
0	1	
1	0	
1	1	

B- The following Truth table must be filled

PB1	PB2	PB3 (manual stop)	LED1
0	0	0	
0	1	0	
1	0	0	
1	1	0	

PB1	PB2	PB3 (manual stop)	LED1
0	0	1	
0	1	1	
1	0	1	
1	1	1	

C- The following Truth table must be filled

PB1	PB2	PB3 (manual stop)	LED1	LED2
0	0	0		
0	1	0		
1	0	0		
1	1	0		

PB1	PB2	PB3 (manual stop)	LED1	LED2
0	0	1		
0	1	1		
1	0	1		
1	1	1		