# Lab Experiment # 08

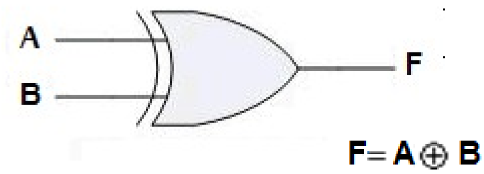
## ***XOR and XNOR gates: Basics and Applications***

### Objectives

To learn how to build XOR gates from basic gates
To learn how to build a Half Adder and a Full Adder using XOR gates.
To learn how to build a parallel adder, subtracter, incrementer and decrementer using full adders.

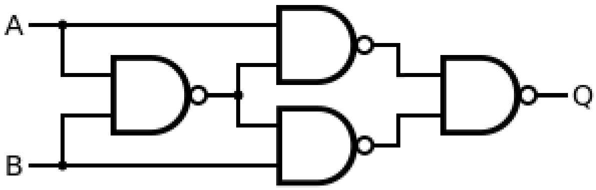
### Background

The XOR gate (sometimes EOR gate, or EXOR gate) is a digital logic gate that implements an exclusive or; that is, a true output (1) results if one, and only one, of the inputs to the gate is true (1). If both inputs are false (0) or both are true (1), a false output (0) results. Next is the circuit representation of the XOR gate and its truth table.

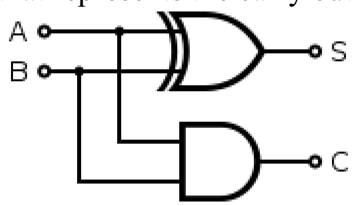



| A | B | F1 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Next is one way to build an XOR gate using NAND gates only

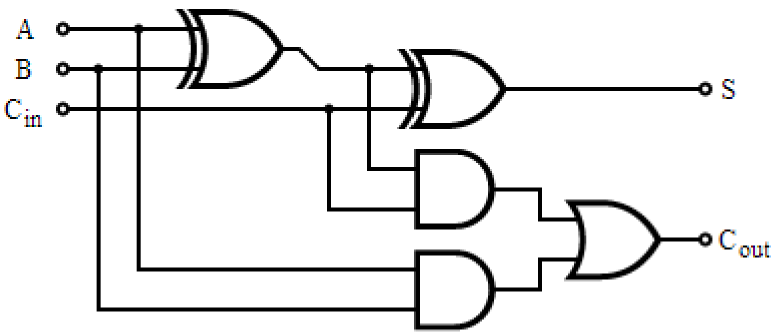



The XOR logic gate can be used as a one-bit adder (or a Half-Adder; HA)that adds any two bits together to output one bit (the sum) and another bit that represents the carry out. As shown below




The XOR logic gate can be used as a one-bit full adder that adds any three bits together to output one bit (the sum) and another bit that represents the carry out. As shown below

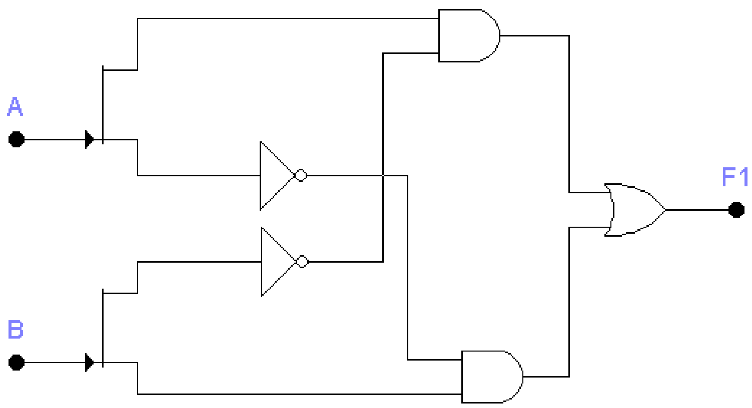
| Inputs | | | Outputs | |
|---|---|---|---|---|
| $A$ | $B$ | $C_{in}$ | $C_{out}$ | $S$ |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | |




## ***Lab Tasks***

**Task 1: XOR built from basic gates**

Draw using EWB the following circuits then fill their truth tables:

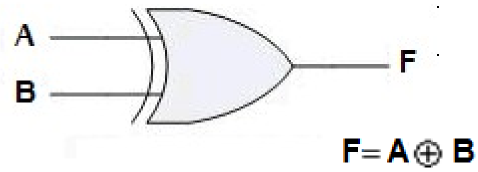



| A | B | F1 |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |

| | | |
|---|---|---|
| 1 | 1 | |

What do you notice?

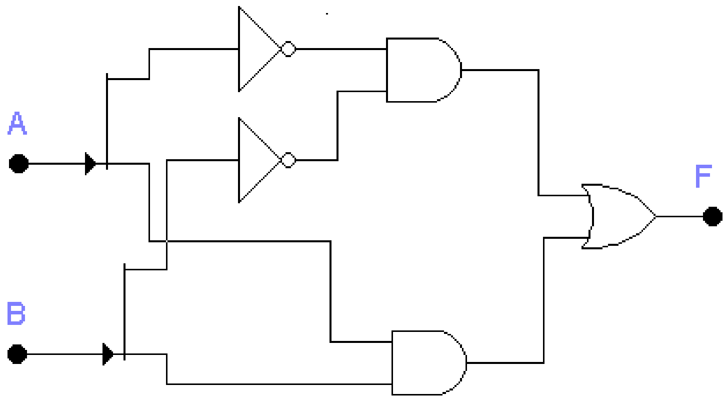
Each one of the above circuits can be replaced with one single logic gate that gives the same truth table, that's the Exclusive OR Gate or XOR.




| A | B | F | No. of 1's |
|---|---|---|---|
| 0 | 0 | 0 | Even |
| 0 | 1 | 1 | Odd |
| 1 | 0 | 1 | Odd |
| 1 | 1 | 0 | Even |

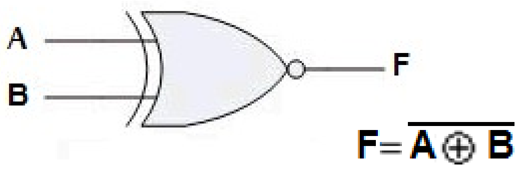
**Task 2: XNOR Gate**

Draw using EWB the following circuit then fill its truth table:




| A | B | F |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

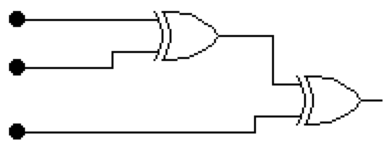
The above circuit can be replaced with one single logic gate that gives the same truth table, that's the Exclusive NOR Gate or XNOR.



| A | B | F | No. of 1's |
|---|---|---|---|
| 0 | 0 | 1 | Odd |
| 0 | 1 | 0 | Even |
| 1 | 0 | 0 | Even |
| 1 | 1 | 1 | Odd |

**Task 3: 3-input XOR Gate**

Draw using EWB a three-input XOR gate. Check the circuit using a Logic converter.



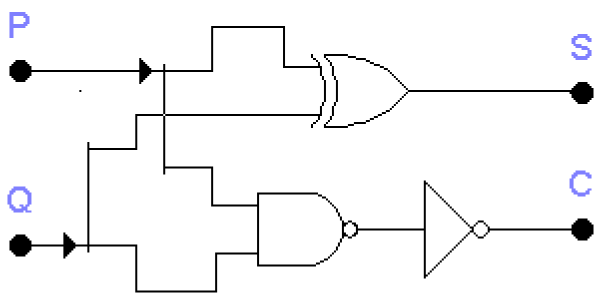
| A | B | C | A B C |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

**Task 4: Half adder circuit**

The following diagram represents the Half Adder ( HA is a Logic Circuit that performs 1-bit binary addition). Given that P and Q are two 1-bit binary numbers, S is the 1-bit Sum of P and Q, and C is the CARRY bit.

**(a)** Find out the Boolean functions S and C, and write them in the corresponding blanks.

**(b)** Draw using EWB the HA circuit then find its truth table by using the logic converter.



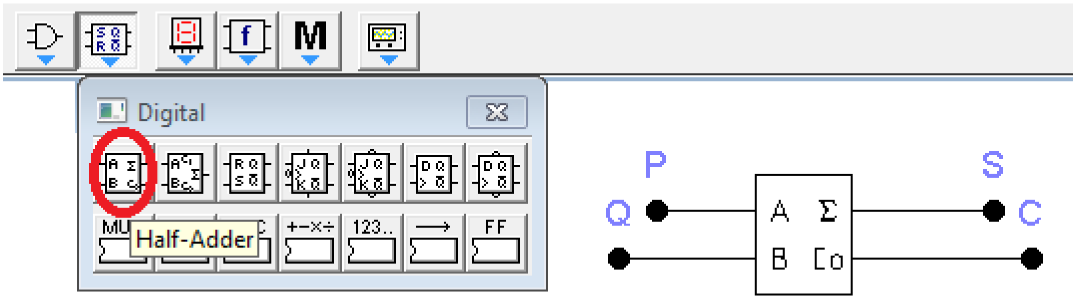

| **S =** |
|---|

| **C =** |
|---|

| P | Q | S | C |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

**Task 5: Implementing HA circuit using EWB**

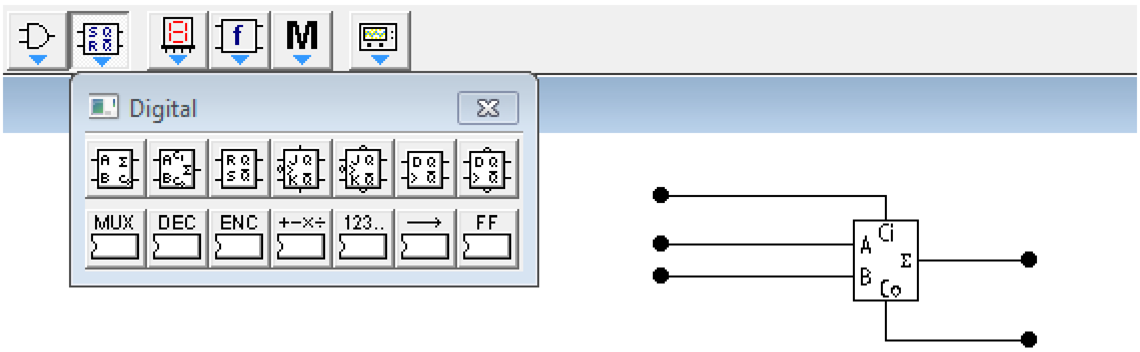
Draw using EWB the HA circuit shown in the figure below then find its truth table by using the logic converter, compare the truth table obtained with the one in Task5, what do you notice?




| P | Q | S | C |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

**Task 6: Implementing FA circuit using EWB**

Draw using EWB a full adder circuit; find out its truth table and Boolean functions.

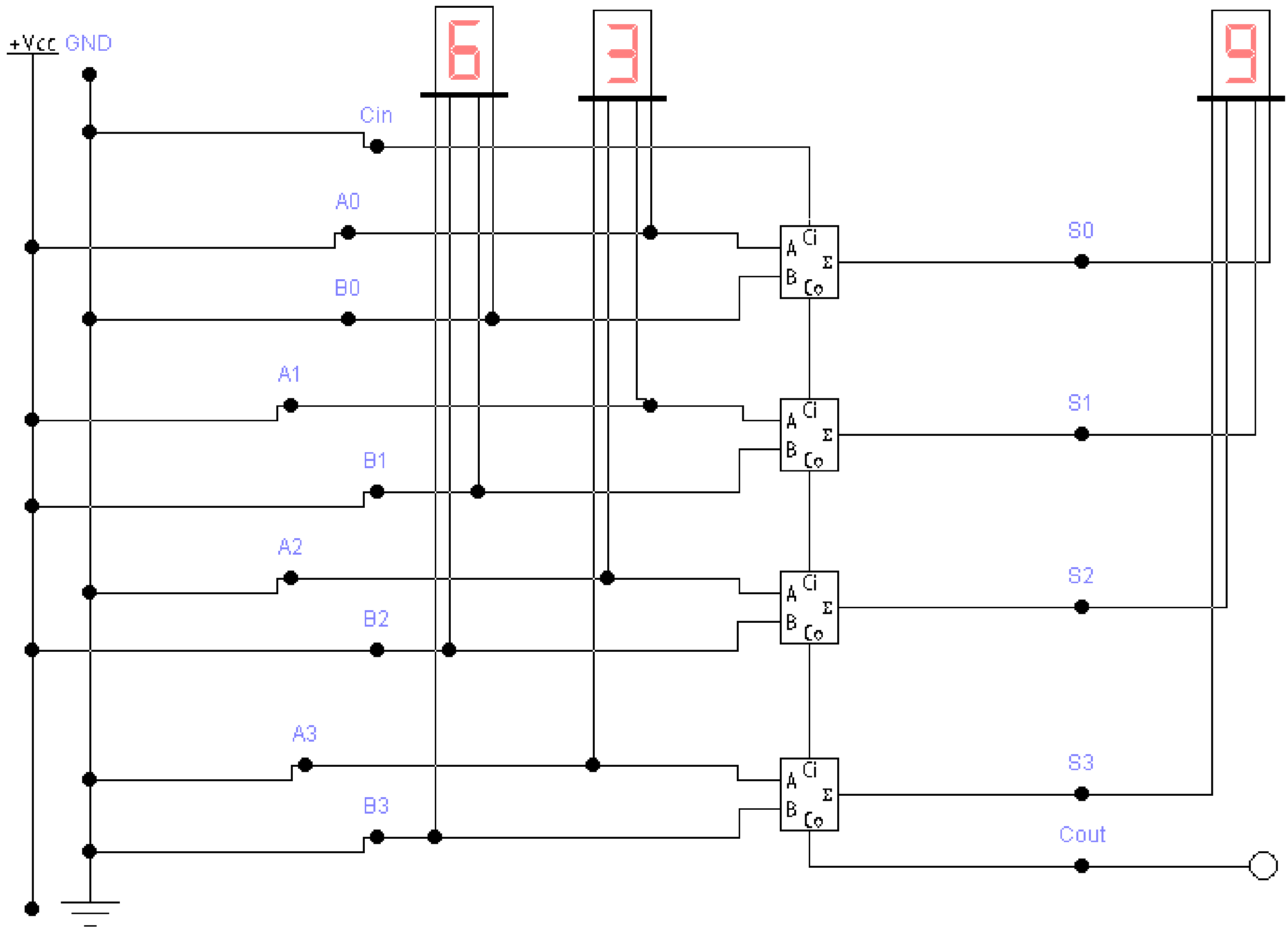


| Cin | P | Q | Sum | Cout |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

**Task 7: Implementing a 4-bit parallel adder using 4 FA's**

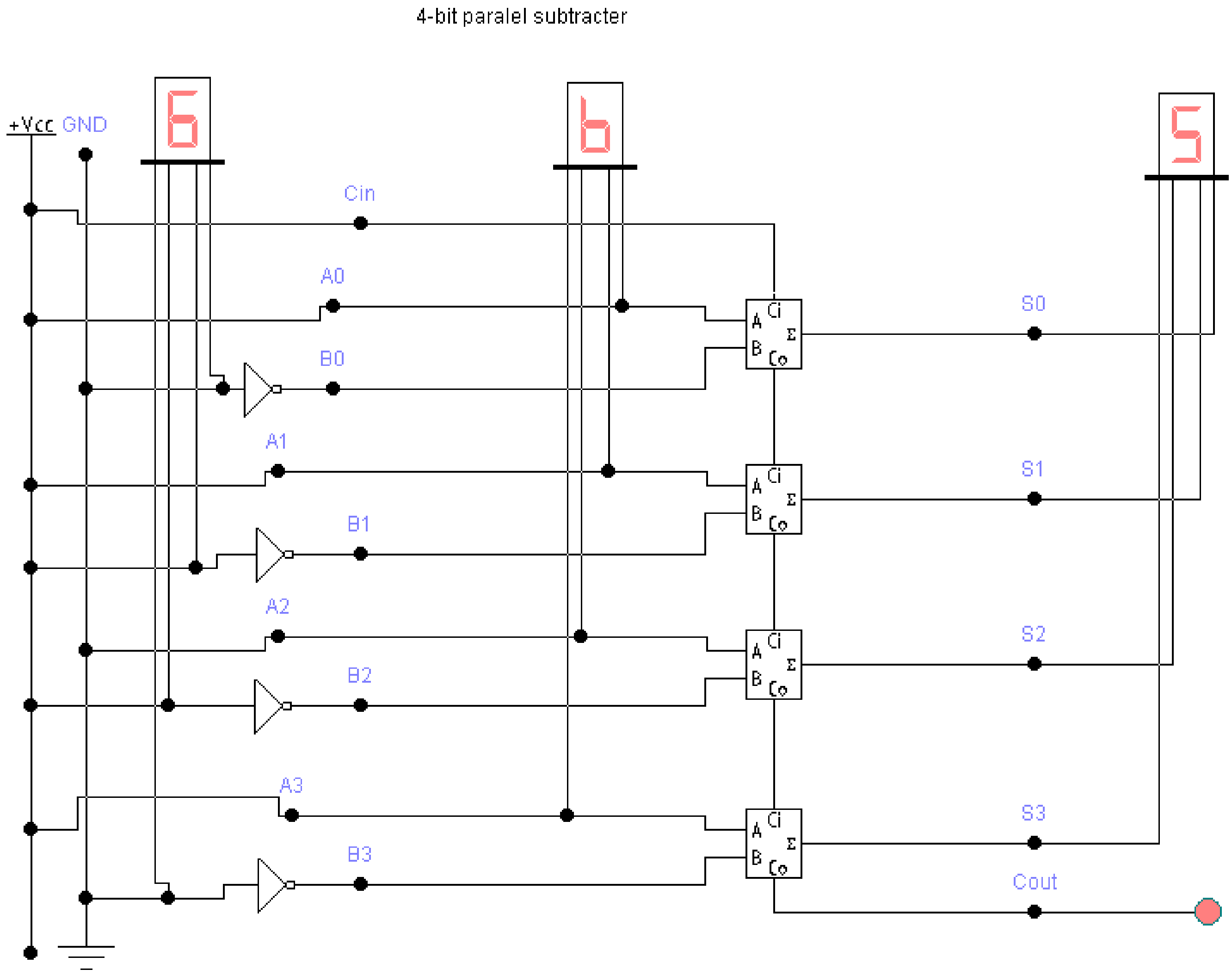
Draw using EWB a 4-bit parallel adder circuit (the circuit below shows (6+3=9))

**Note**: you can find the decoded 7-segment under the "indicators" toolbar.



**Task 8: Implementing a 4-bit parallel subtracter using 4 FA's**

Draw using EWB a 4-bit parallel adder circuit (the circuit below shows (b-6=5))




**Task 9: Implementing a 4-bit incrementer using 4 FA's**

Draw using EWB a 4-bit incrementer circuit.

**Task 10: Implementing a 4-bit decrementer using 4 FA's**

Draw using EWB a 4-bit decrementer circuit.