**Lab 1: Verification Of AND Gate**

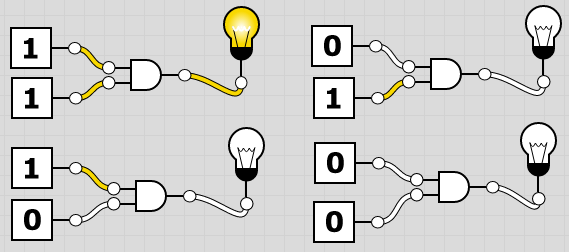
Objective:To verify the truth table of AND Gate.

Theory: The AND Gate is a basic logic gate with two or more inputs and one output that performs logical conjunction. The output of an AND gate is true when all of the inputs are true.

Experiment/Tool: Logic Gate Simulator

Logic Symbol: Boolean Expression:Y = A.B

Simulation:



Working Mechanism**:** In AND Gate, the following truth table illustrates that the output is true when both inputs are high otherwise it is false. In the simulation above, it is seen that the bulb is ON when both inputs are 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A.B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth Table:

Conclusion**:**  From the truth table, the AND gate has been verified.

**Lab 2: Verification Of OR Gate**

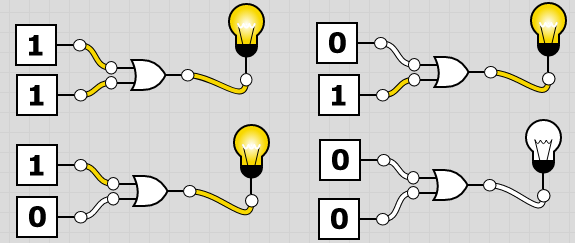
Objective:To verify the truth table of OR Gate.

Theory: The OR Gate is a basic logic gate with two or more inputs and one output that performs logical conjunction. The output of an OR Gate is false when all of the inputs are false.

Logic Symbol: Boolean Expression:Y = A+B

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** In OR Gate, the following truth table illustrates that the output is false when both inputs are low otherwise it is true. In the simulation above, it is seen that the bulb is OFF when both inputs are 0.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A+B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth Table:

Conclusion**:**  From the truth table, the OR Gate has been verified.

**Lab 3: Verification Of NOT Gate**

Objective:To verify the truth table of NOT Gate.

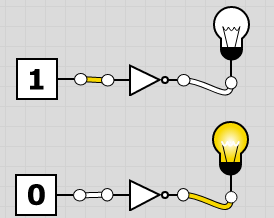
Theory: A NOT Gate is also called as inverter because it implements logical negation.

It uses just one input to generate one output. It inverts the input – the output is 1 if the input is 0 and output is 0 if the input is 1.

Logic Symbol: Boolean Expression:Y = A’

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** In NOT Gate, the following truth table illustrates that the output is true when both inputs are true otherwise it is false. In the simulation above, it is seen that the bulb is ON when both inputs are 1.

|  |  |
| --- | --- |
| Input | Output  Y= A’ |
| 0 | 1 |
| 1 | 0 |

Truth Table:

Conclusion**:**  From the truth table, the NOT gate has been verified.

**Lab 4: Verification Of NAND Gate**

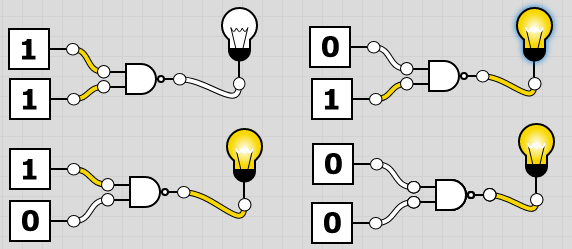
Objective:To verify the truth table of NAND Gate.

Theory: The NAND gate is a universal logic gate with two or more inputs and one output that performs logical conjunction. The output of a NAND Gate is false when all of the inputs are true.

Logic Symbol: Boolean Expression:Y = (A.B)’

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** In NAND Gate, the following truth table illustrates that the output is false when both inputs are low otherwise it is true. In the simulation above, it is seen that the bulb is OFF when both inputs are 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= (A.B)’ |
| A | B |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth Table:

Conclusion**:**  From the truth table, the NAND Gate has been verified.

**Lab 5: Verification Of NOR Gate**

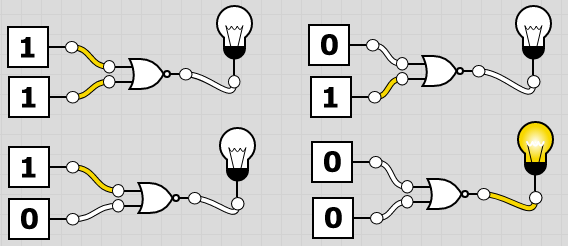
Objective:To verify the truth table of NOR Gate.

Theory: The NOR Gate is a universal logic gate with two or more inputs and one output that performs logical conjunction. The output of an NOR Gate is true only when all of the inputs are false.

Logic Symbol: Boolean Expression:Y = (A+B)’

Experiment/Tool: Logic Gate Simulator

Simulation: `



Working Mechanism**:** In NOR Gate, the following truth table illustrates that the output is true when both inputs are false otherwise it is true. In the simulation above, it is seen that the bulb is ON when both inputs are 0.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= (A+B)’ |
| A | B |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Truth Table:

Conclusion**:**  From the truth table, the NOR Gate has been verified.

**Lab 6: Verification Of XOR Gate**

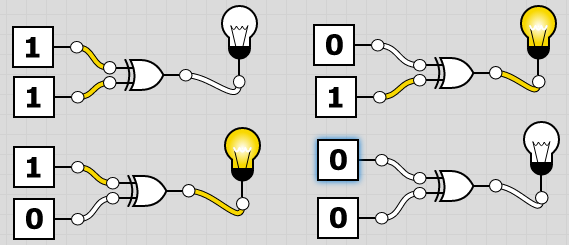
Objective:To verify the truth table of XOR Gate.

Theory: The XOR gate is a universal logic gate with two or more inputs and one output that performs logical conjunction. The output of an XOR Gate is true when inputs are either 0 or 1.

Logic Symbol: Boolean Expression:Y = (A+B). (A’+B’)

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** In XOR Gate, the following truth table illustrates that the output is true when one input is low or other is high otherwise it is false. In the simulation above, it is seen that the bulb is ON when inputs are either 0 or 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y = (A+B). (A’+B’) |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth Table:

Conclusion**:**  From the truth table, the XOR Gate has been verified.

**Lab 7: Verification Of XNOR Gate**

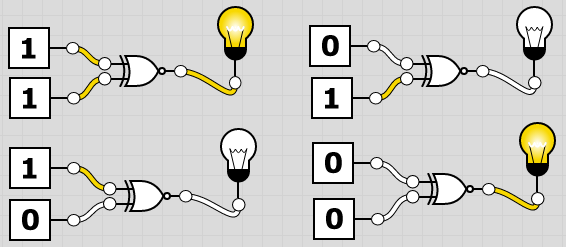
Objective:To verify the truth table of XNOR Gate.

Theory: The XNOR Gate is a universal logic gate with two or more inputs and one output that performs logical conjunction. The output of an XNOR Gate is false when inputs are either 0 or 1.

Logic Symbol: Boolean Expression:Y = (A.B) + (A’.B’)

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** In XNOR Gate, the following truth table illustrates that the output is false when one input is low or other is high otherwise it is true. In the simulation above, it is seen that the bulb is OFF when the inputs are either 0 or 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  (A.B) + (A’. B’) |
| A | B |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth Table:

Conclusion**:**  From the truth table, the XNOR Gate has been verified.

**Lab 8: Verification Of NAND Gate As NOT Gate**

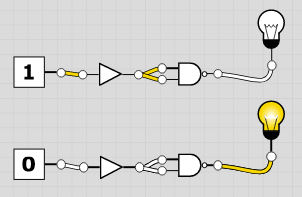
Objective:To verify the truth table of NAND Gate as NOT Gate.

Theory: NAND Gate is a universal logic gate which can be used to derive basic AND, OR and NOT gates. A simple input can be passed to the NAND Gate to get logical complement of the input which makes the NAND Gate act as a NOT Gate.

Boolean Expression:Y = (A.B)’= A’ **[By Idempotent Law]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NAND Gate to get their complements. As seen in simulation above, doing that results in both the output of the NAND Gate and NOT Gate being the same which turns ON when input is 0 and OFF when input is 1.

|  |  |
| --- | --- |
| Input | Output |
| 0 | 1 |
| 1 | 0 |

|  |  |
| --- | --- |
| Input | Output  Y= A’ |
| 0 | 1 |
| 1 | 0 |

Truth Table:

**Truth Table for NOT Gate Truth Table for NAND Gate as NOT Gate**

Conclusion**:**  From the truth table, the NOT Gate can be derived from NAND Gate.

.**Lab 9: Verification Of NAND Gate As AND Gate**

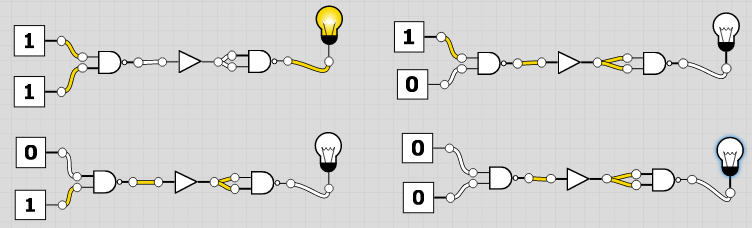
Objective:To verify the truth table of NAND Gate as AND Gate.

Theory: NAND Gate is a universal logic gate which can be used to derive basic AND, OR and NOT gates. AND Gate can be derived from NAND Gate by passing the inputs through a NAND Gate and passing the result to another NAND Gate again.

Boolean Expression:Y = ((A. B)’)’ = (A.B) **[De-Morgan’s Theorem]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NAND Gate and generated output is again passed through another NAND Gate which generates the output of AND Gate. The output is 1 only if both inputs are 1. As seen above in simulation, the bulb is ON as both inputs are 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A. B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| Inputs | | Output |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth Table:

**Truth Table for AND Gate Truth Table for NAND Gate as AND Gate**

Conclusion**:**  From the truth table, the AND Gate can be derived from NAND Gate.

**Lab 10: Verification Of NAND Gate As OR Gate**

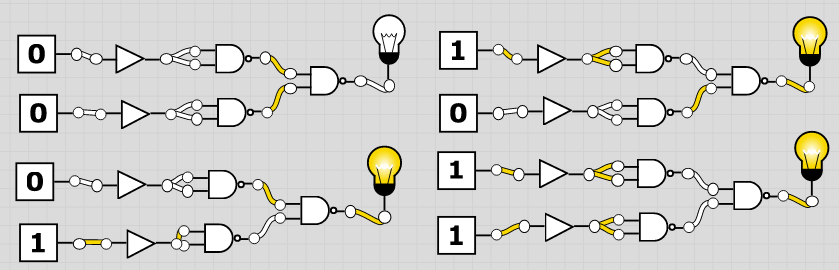
Objective:To verify the truth table of NAND Gate as OR Gate.

Theory: NAND Gate is a universal logic gate which can be used to derive basic AND, OR and NOT gates. OR Gate can be derived from NAND Gate by passing the inputs through a NAND Gate and passing the result to another NAND Gate again.

Boolean Expression:Y = (A’. B’)’ = (A+B) **[By Idempotent Law]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NAND Gate and generated output is again passed through another NAND Gate which generates the output of OR Gate. The output is 0 only if both inputs are 0. As seen above in simulation, the bulb is OFF as both inputs are 0.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A+B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A+B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth Table:

**Truth Table for OR Gate Truth Table for NAND Gate as OR Gate**

Conclusion**:**  From the truth table, the OR Gate can be derived from NAND Gate.

**Lab 11: Verification Of NOR Gate As NOT Gate**

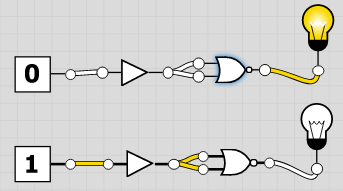
Objective:To verify the truth table of NOR Gate as NOT Gate.

Theory: NOR Gate is a universal logic gate which can be used to derive basic AND, OR and NOT gates. A simple input can be passed to the NOR Gate to get logical complement of the input which makes the NOR Gate act as a NOT Gate.

Boolean Expression:Y = (A.B)’= A’ **[By Idempotent Law]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NOR Gate to get their complements. As seen in simulation above, doing that results in both the output of the NOR Gate and NOT Gate being the same which turns ON when input is 0 and OFF when input is 1.

|  |  |
| --- | --- |
| Input | Output |
| 0 | 1 |
| 1 | 0 |

|  |  |
| --- | --- |
| Input | Output  Y= A’ |
| 0 | 1 |
| 1 | 0 |

Truth Table:

**Truth Table for NOT Gate Truth Table for NOR Gate as NOT Gate**

Conclusion**:**  From the truth table, the NOT Gate can be derived from NOR Gate.

.**Lab 12: Verification Of NOR Gate As AND Gate**

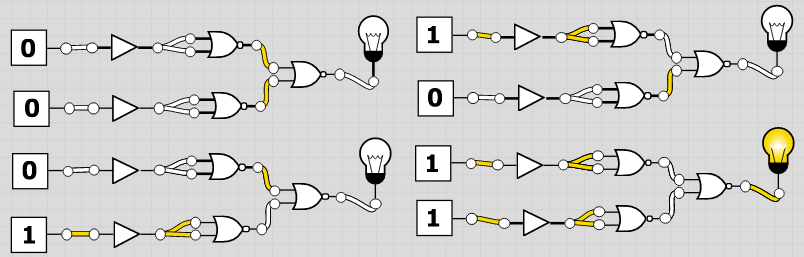
Objective:To verify the truth table of NOR Gate as AND Gate.

Theory: NOR Gate is a universal logic gate which can be used to derive basics AND, OR and NOT gates. AND Gate can be derived from NOR Gate by passing the inputs through a NAND Gate and passing the result to another NOR Gate again.

Boolean Expression:Y = (A’+B’) = (A.B) **[De-Morgan’s Theorem]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NOR Gate and generated output is again passed through another NOR Gate which generates the output of AND Gate. The output is 1 only if both inputs are 1. As seen above in simulation, the bulb is ON as both inputs are 1.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A. B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| Inputs | | Output |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth Table:

**Truth Table for AND Gate Truth Table for NOR Gate as AND Gate**

Conclusion**:**  From the truth table, the AND Gate can be derived from NOR Gate.

**Lab 13: Verification Of NOR Gate As OR Gate**

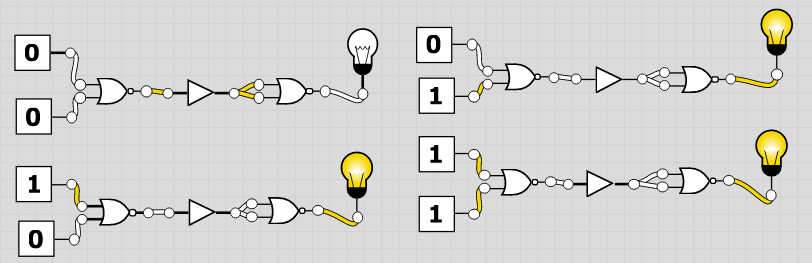
Objective:To verify the truth table of NOR Gate as OR Gate.

Theory: NAND Gate is a universal logic gate which can be used to derive basic AND, OR and NOT gates. OR Gate can be derived from NOR Gate by passing the inputs through a NAND Gate and passing the result to another NOR Gate again.

Boolean Expression:Y = ((A+B)’)’ = A+B **[By Involution Law]**

Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:** Each input is passed to the NOR Gate and generated output is again passed through another NOR Gate which generates the output of OR Gate. The output is 0 only if both inputs are 0. As seen above in simulation, the bulb is OFF as both inputs are 0.

|  |  |  |
| --- | --- | --- |
| Inputs | | Output |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| Inputs | | Output  Y= A+B |
| A | B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth Table:

**Truth Table for OR Gate Truth Table for NOR Gate as OR Gate**

Conclusion**:**  From the truth table, the OR Gate can be derived from NOR Gate.

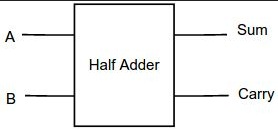
**Lab 14: Verification Of Half Adder**

Objective:To verify the truth table of Half Adder.

Theory: Half Adder is a combinational circuit which is used to add two 1-bit numbers A and B. The input variables designated the augends and addend bits and the output variables produce the SUM and CARRY. The output variables are S for SUM and C for Carry.

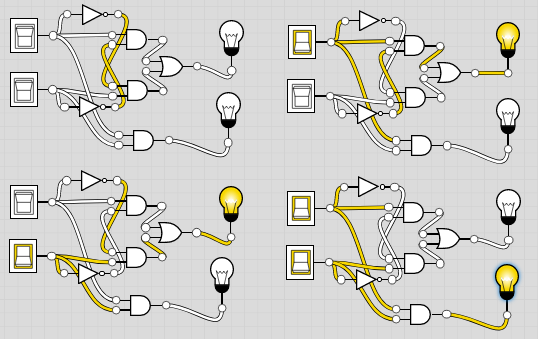
Equation:  **SUM** = (A. B’) + (A’. B) **CARRY** = A. B

Block Diagram:



Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:**

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | | Outputs | |
| A | B | Sum(S)  (A.B’) + (A’. B) | Carry(C)   1. B |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Truth Table:

Conclusion**:**  From the truth table, the Half Adder has been verified.

\

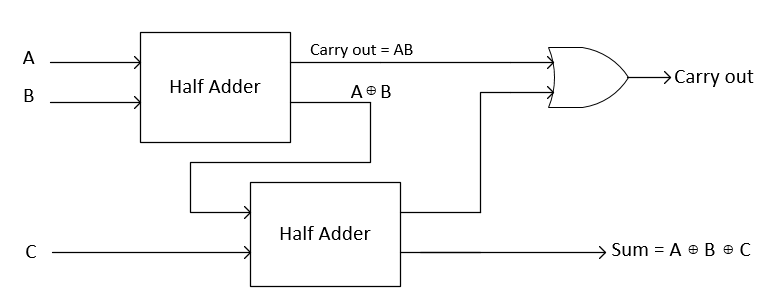
**Lab 15: Verification Of Full Adder**

Objective:To verify the truth table of Full Adder.

Theory: Full Adder is a combinational circuit which is used to add three 1-bit numbers A, B and C. The input variables designated the augends and addend bits and the output variables produce the SUM and CARRY. The output variables are S for SUM and C for Carry.

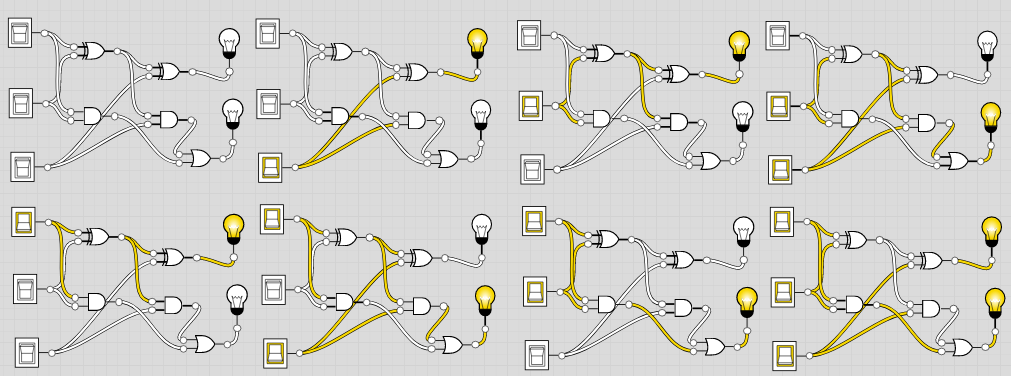
Equation:  **SUM** = (A. B’) + (A’. B) **CARRY** = A. B

Block Diagram:



Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inputs | | | Outputs | |
| A | B | C | Sum(S)  (A.B’) + (A’. B) | Carry(C)   1. B |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Truth Table:

Conclusion**:**  From the truth table, the Full Adder has been verified.

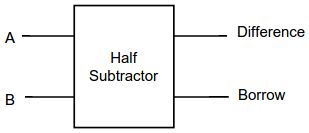
**Lab 16: Verification Of Half Subtractor**

Objective:To verify the truth table of Half Subtractor.

Theory: Half Subtractor is a combinational circuit which is used to subtract two 1-bit numbers A and B. The input variables designated the minuend and subtrahend bits and the output variables produce the DIFFERENCE and BORROW. The output variables are D for DIFFERENCE and B’ for BORROW.

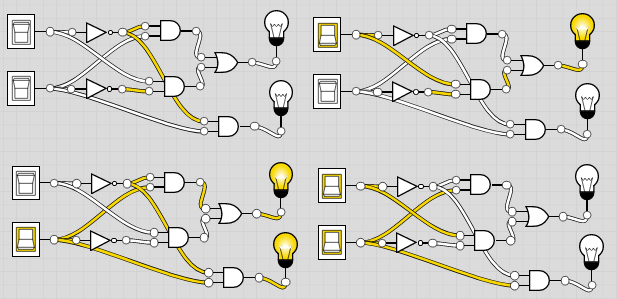
Equation:  **DIFFERENCE =** A’B + AB**’ BORROW =** A’B

Block Diagram:



Experiment/Tool: Logic Gate Simulator

Simulation:



Working Mechanism**:**

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | | Outputs | |
| A | B | Difference(D)  A’B + AB’ | Borrow(B’)  A’. B |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Truth Table:

Conclusion**:**  From the truth table, the Half Subtractor has been verified.