**Week-10**

**DLD LAB-10**

**Name: shamaim noor**

**L1f24BSCS0040**

**Binary to Gray Code Conversion, Half Adder, and Full Adder Circuit**

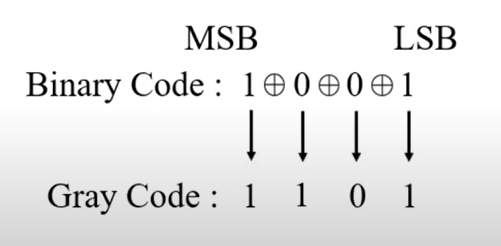
**Objectives:**

* To understand concept of Binary to Gray Code Conversion, Half Adder and Full Adder circuits.
* To validate implementation of circuits using **Circuit Maker 2000**.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Decimal Number** | **4-bit Binary Code** | | | | **4-bit Gray Code** | | | |
|  |  |  |  |  |  |  |  |  |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** |
| **2** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **1** |
| **3** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **0** |
| **4** | **0** | **1** | **0** | **0** | **0** | **1** | **1** | **0** |
| **5** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **1** |
| **6** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **1** |
| **7** | **0** | **1** | **1** | **1** | **0** | **1** | **0** | **0** |
| **8** | **1** | **0** | **0** | **0** | **1** | **1** | **0** | **0** |
| **9** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **1** |
| **10** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **1** |
| **11** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **0** |
| **12** | **1** | **1** | **0** | **0** | **1** | **0** | **1** | **0** |
| **13** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | **1** |
| **14** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **1** |
| **15** | **1** | **1** | **1** | **1** | **1** | **0** | **0** | **0** |

**Binary to Gray Code Conversion**

**Binary to Gray code Conversion:**

****

**Gray to Binary code Conversion:**



**General Rule for Conversion:**

* **G₄ = B₄**
* **G₃ = B₄ ⊕ B₃** (XOR)
* **G₂ = B₃ ⊕ B₂**
* **G₁ = B₂ ⊕ B₁**

**Example: Decimal 5**

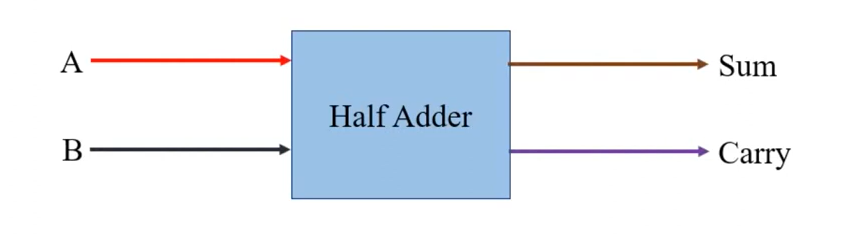
* **Binary (B₄ B₃ B₂ B₁) = 0101**
* Compute Gray Code:
  + G₄ = B₄ = 0
  + G₃ = B₄ ⊕ B₃ = 0 ⊕ 1 = 1
  + G₂ = B₃ ⊕ B₂ = 1 ⊕ 0 = 1
  + G₁ = B₂ ⊕ B₁ = 0 ⊕ 1 = 1
* **Gray Code = 0111**

**Example: Decimal 10**

* **Binary = 1010**
* G₄ = 1
* G₃ = 1 ⊕ 0 = 1
* G₂ = 0 ⊕ 1 = 1
* G₁ = 1 ⊕ 0 = 1
* **Gray Code = 1111**

**Half Adder Circuit**

In Digital Logic Design (DLD), a Half Adder is a basic combinational circuit used to add two single-bit binary numbers.



**Inputs:**

* A (1-bit)
* B (1-bit)

**Outputs:**

* **Sum** (S) → result of A + B
* **Carry** (C) → carry-out bit if there's an overflow

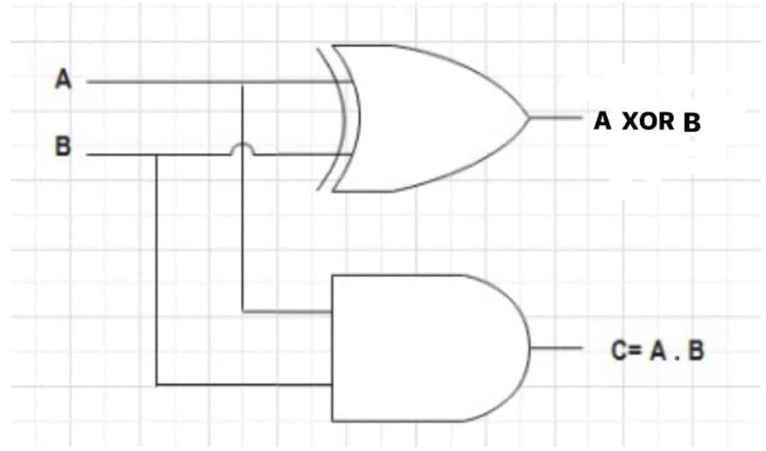
**Logic Expressions:**

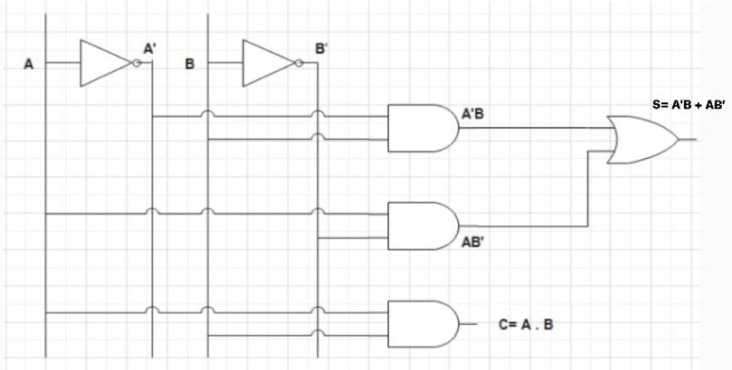
* Sum (S) = **A ⊕ B** (XOR gate)
* Carry (C) = **A · B** (AND gate)

**Truth Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Sum (A ⊕ B)** | **Carry (A · B)** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Circuit Diagram:**

****

****

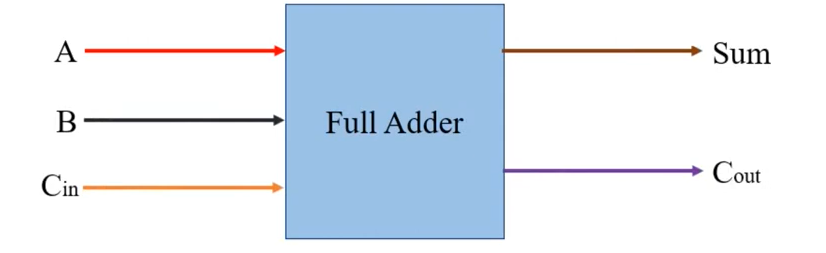
**Full Adder Circuit**

A **Full Adder** is a combinational circuit that **adds three 1-bit binary numbers**:

* Input A
* Input B
* Carry-in (**Cin** from previous stage)

It gives:

* **Sum (S)**
* **Carry-out (Cout)**

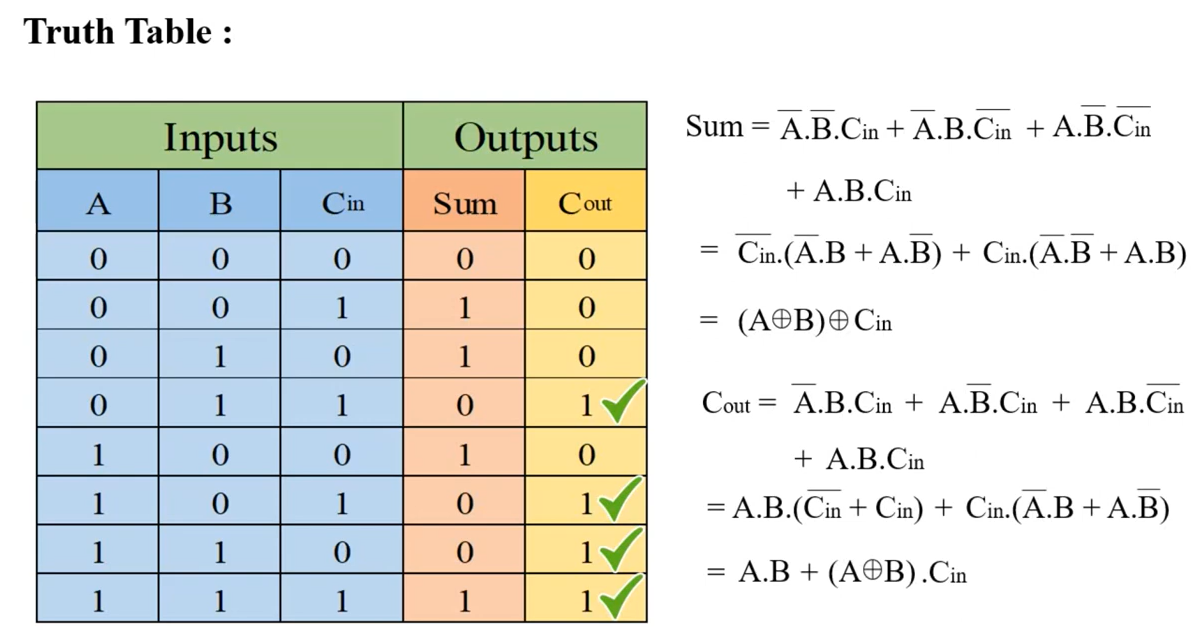
****

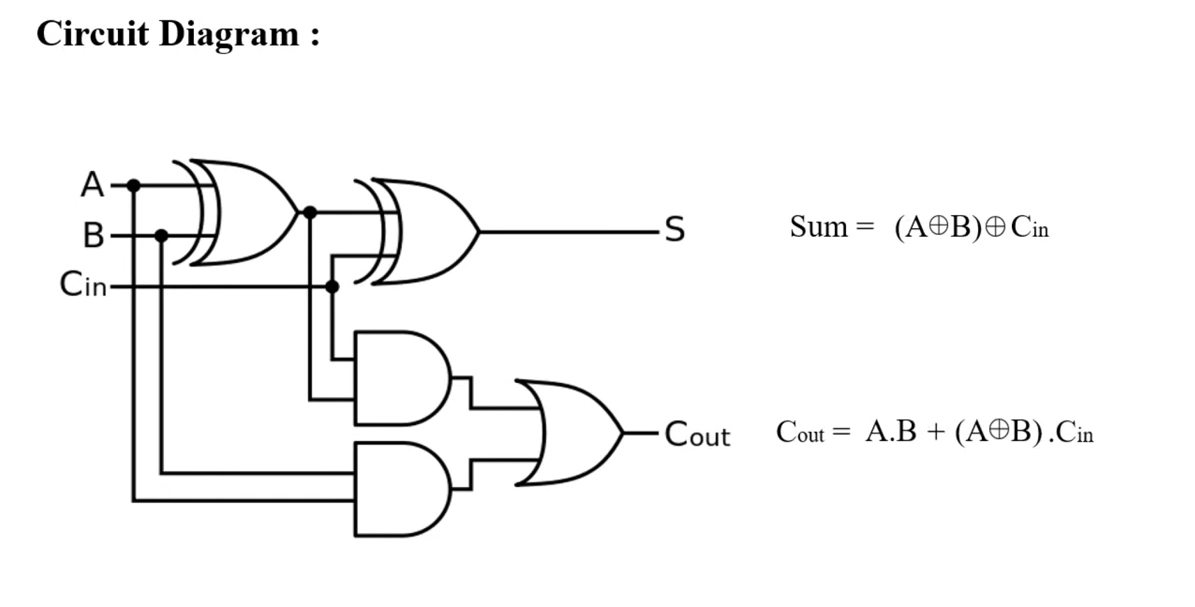
**Inputs:**

* A (1-bit)
* B (1-bit)
* Cin (Carry-in)

**Logic Expressions:**

* **Sum = A ⊕ B ⊕ Cin**
* **Carryout = (A·B) + (Cin.(A ⊕ B)**

****

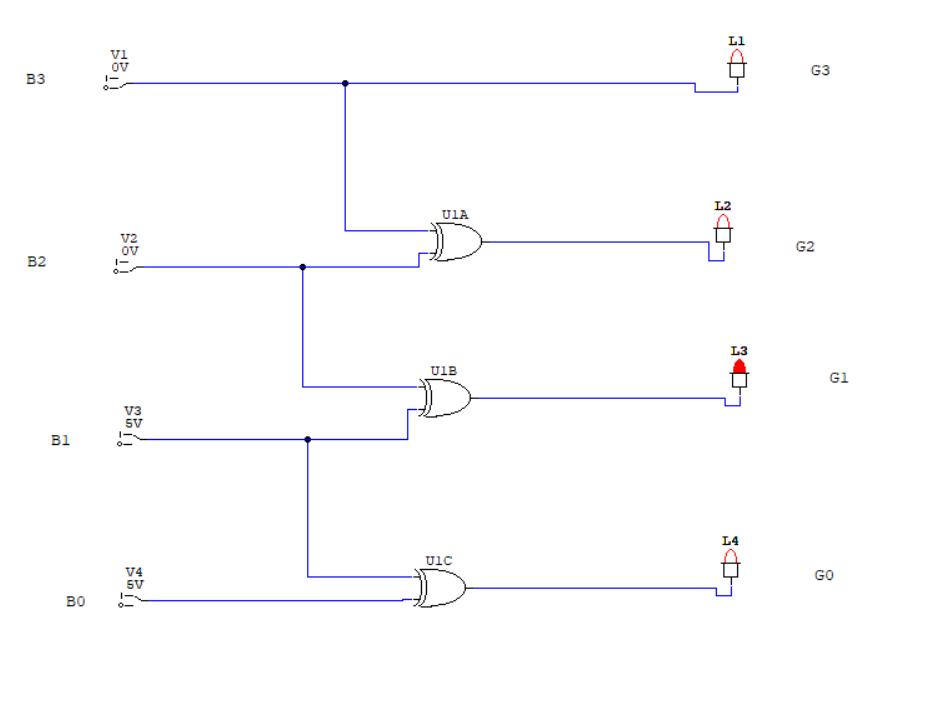
****

**LAB TASKS**

**TASK-01**

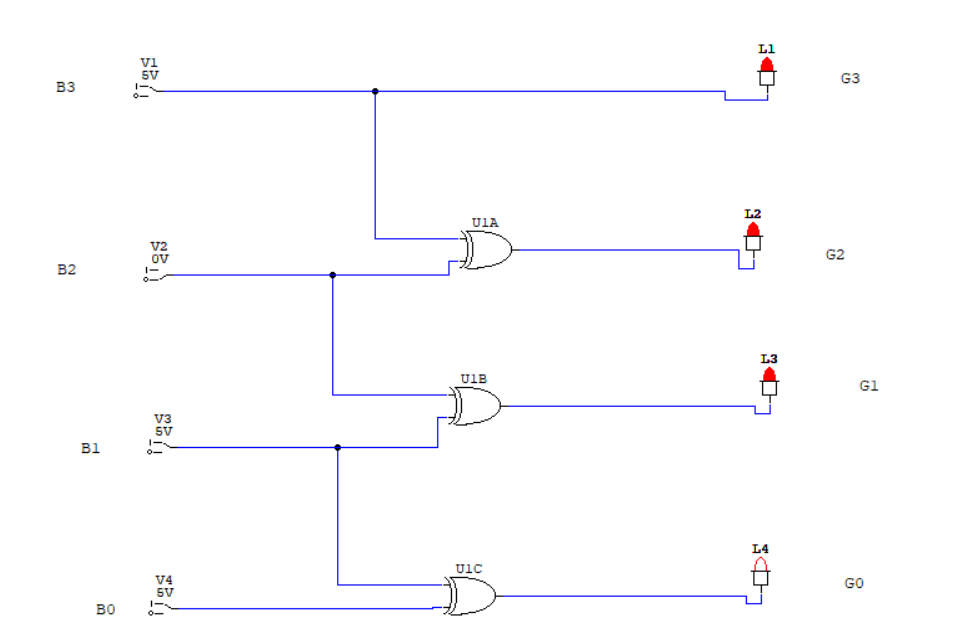
**Implement a 4-bit Binary to Gray Converter on Circuit Maker and verify the results using the above given table. Please provide the circuit snapshots against highlighted combinations.**

**Solution:**

****

**A diagram of a circuit

AI-generated content may be incorrect.**

****

**A diagram of a circuit

AI-generated content may be incorrect.**

**TASK-02**

**Implement a 4-bit Gray to Binary Converter on Circuit Maker and verify the results using the above given table. Please provide the circuit snapshots against highlighted combinations.**

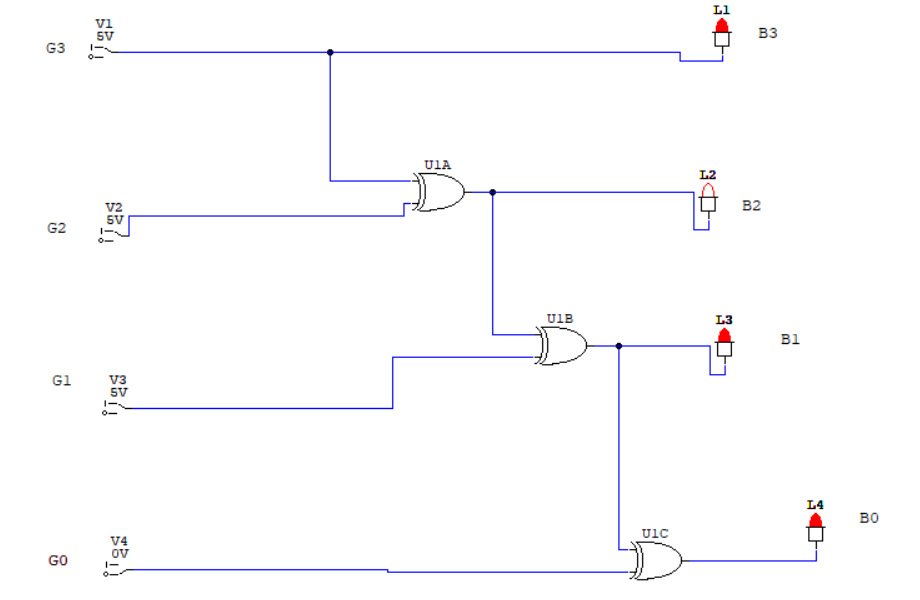
**Solution:**

**A diagram of a circuit

AI-generated content may be incorrect.**

**A diagram of a circuit

AI-generated content may be incorrect.**

****

**A diagram of a circuit

AI-generated content may be incorrect.**

**TASK-03**

1. Draw a truth table for **half adder**. Clearly mention all inputs and outputs.
2. For both **CARRY** and **SUM**, write standard SOP Expression.
3. For both **CARRY** and **SUM**, write the Minimized SOP using **K-Map** grouping techniques.
4. For both **standard** and **minimized expression** on Circuit Maker and verify the results.

**Solution:**

| **A** | **B** | **Sum (S)** | **Carry (C)** |
| --- | --- | --- | --- |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** |

**b) SOP expression for Sum:**

**Sum (S)=A′B+AB′**

**SOP expression for Carry:**

**carry (C)=AB**

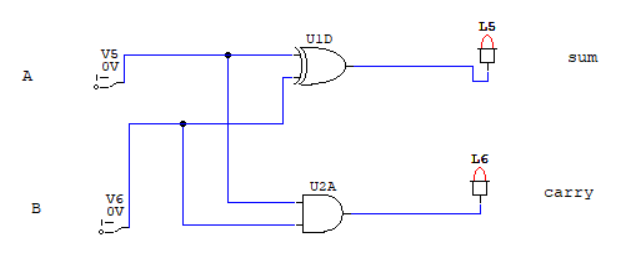
**c) Minimized SOP for Sum:**

**Sum (S)= A′B+AB′**

**Minimized SOP for Carry:**

**Carry (C)= AB**

**d)**

****

**A diagram of a wire

AI-generated content may be incorrect.**

**A diagram of a block diagram

AI-generated content may be incorrect.**

**A diagram of a circuit

AI-generated content may be incorrect.**

**TASK-04**

1. Draw a truth table for **full adder**. Clearly mention all inputs and outputs.
2. For both **CARRY** and **SUM**, write standard SOP Expression.
3. For both **CARRY** and **SUM**, write the Minimized SOP using **K-Map** grouping techniques.
4. For both **standard** and **minimized expression** on Circuit Maker and verify the results.

**Solution:**

**a)**

| **A** | **B** | **Cin** | **Sum (S)** | **Carry (Cout)** |
| --- | --- | --- | --- | --- |
| **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** |

**b)**

**Standard SOP expression for Sum:**

**Sum (S)= A′B′Cin+A′BCin′+AB′Cin′+ABCin**

**Standard SOP expression for Carry:**

**Carry (Cout)= A′BCin+AB′Cin+ABCin′+ABCin**

**c)** **Minimized SOP for Sum:**

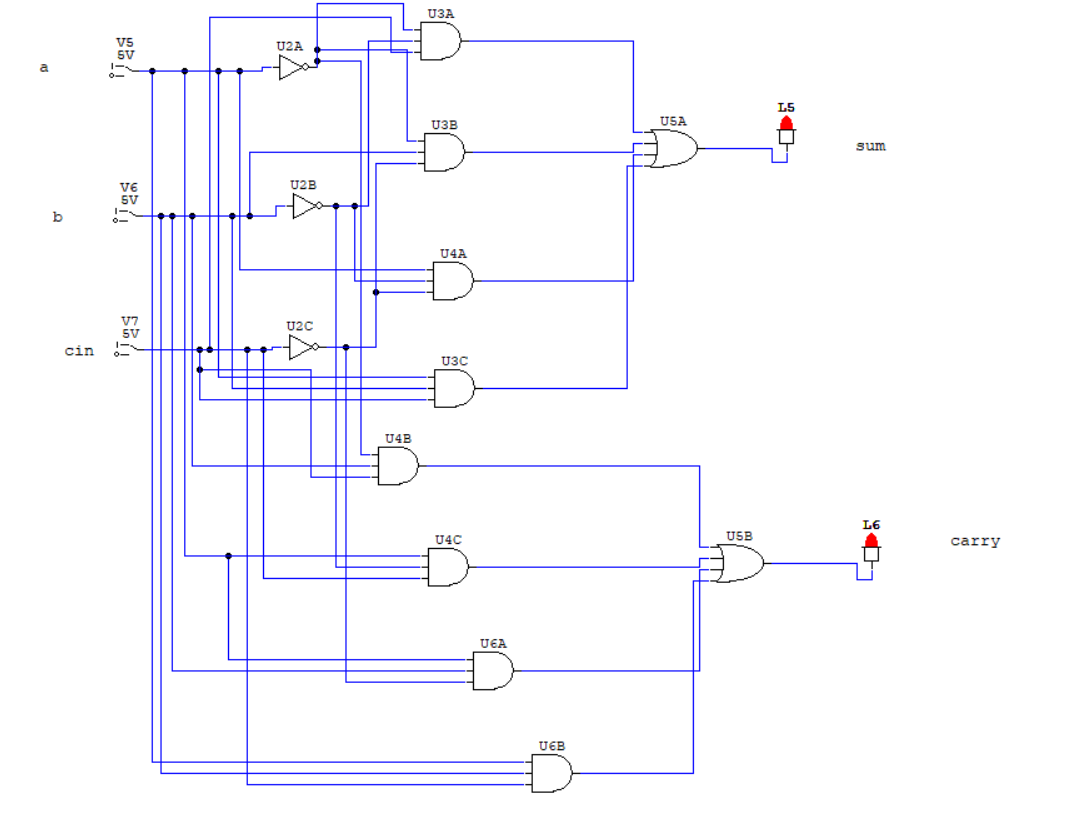
**Sum (S)= A⊕B⊕Cin**

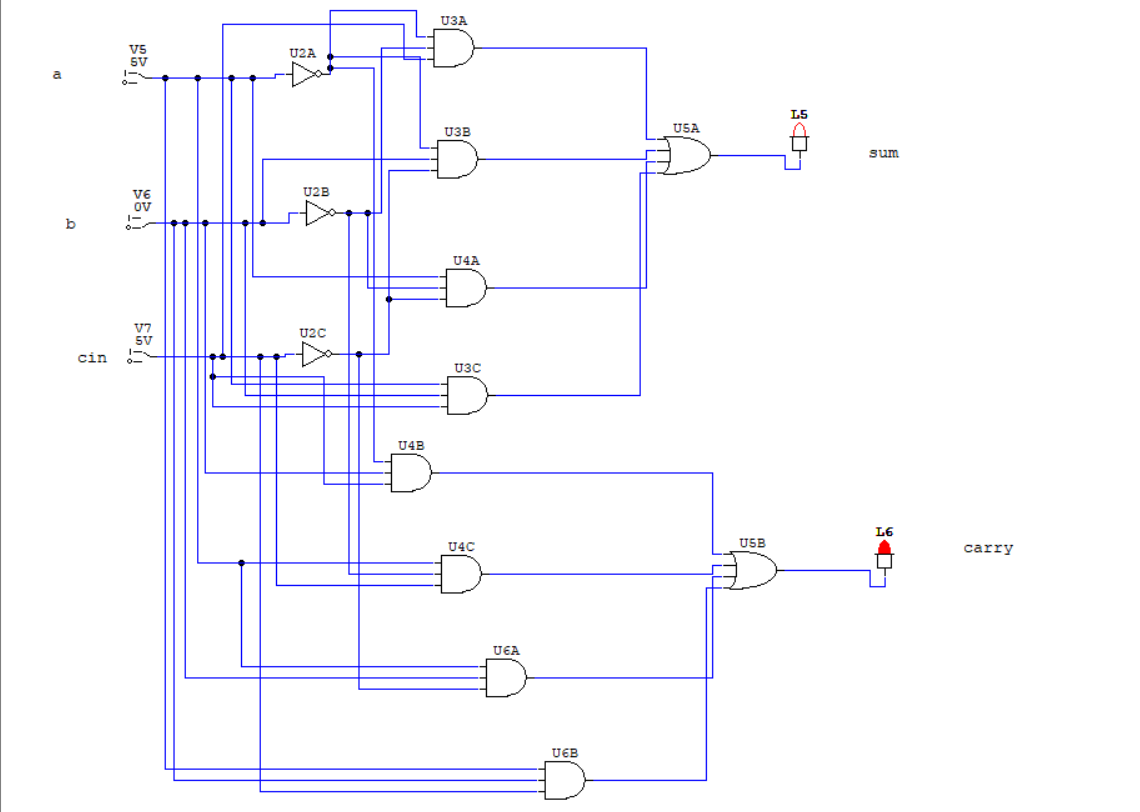
**Minimized SOP for Carry:**

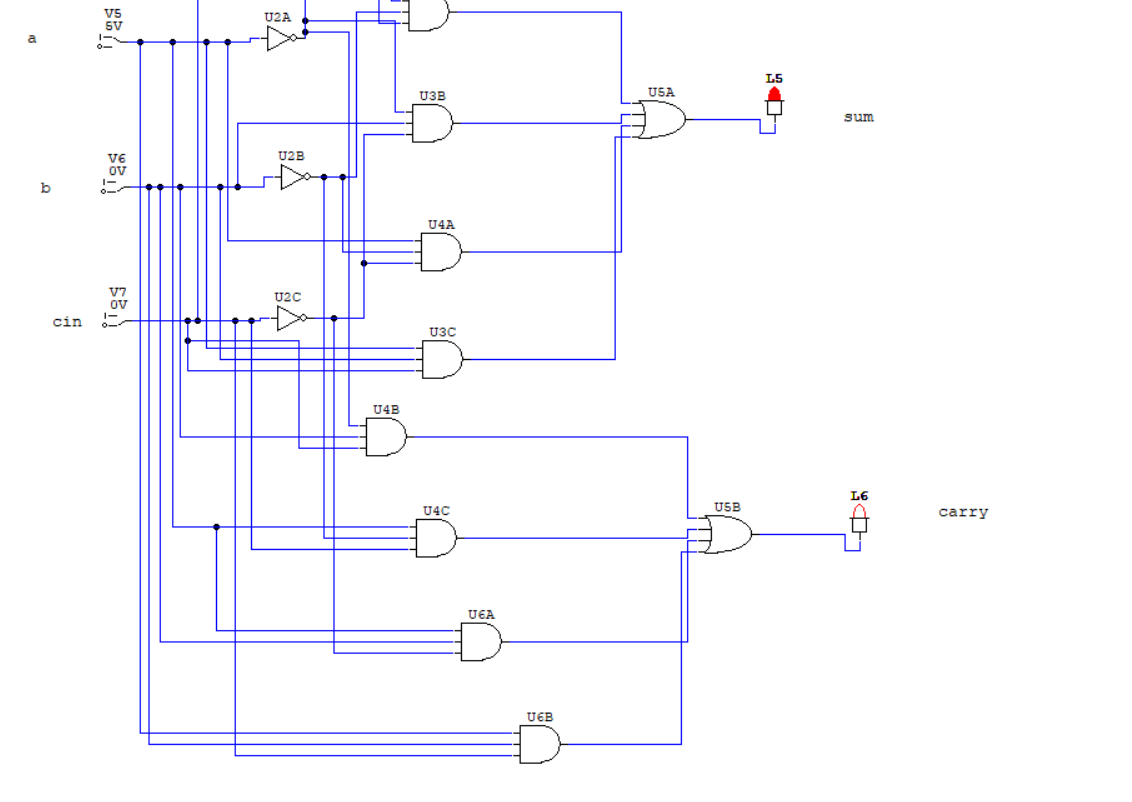
**Carry (Cout)= AB+ACin+Bcin**

**d)**

**standard circuit:**

****

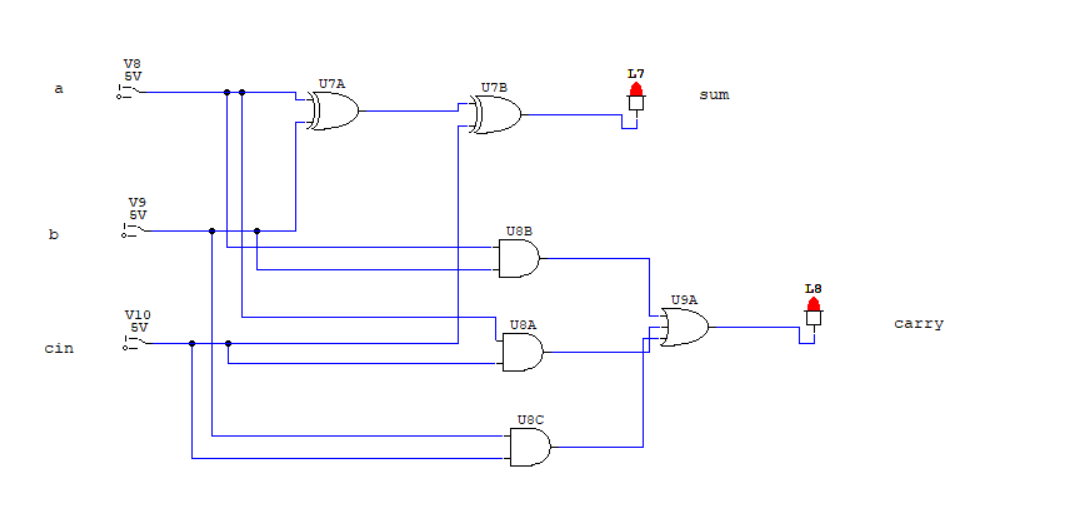
****

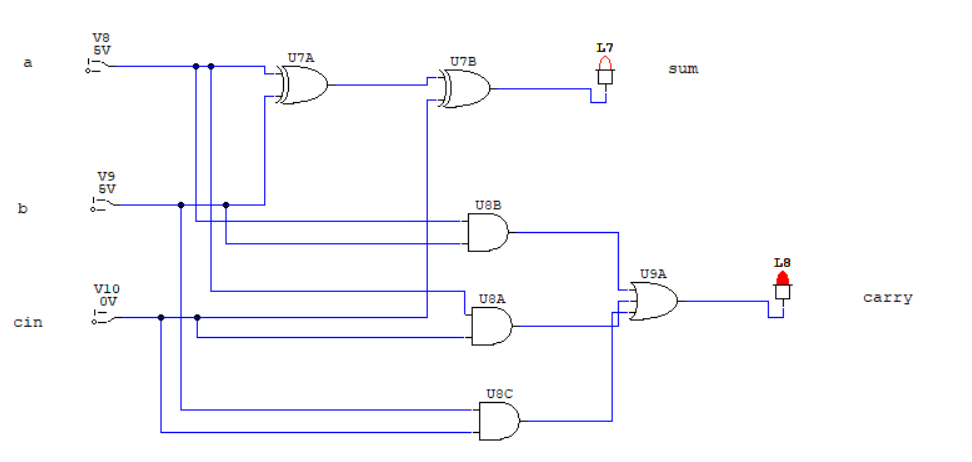
****

A diagram of a circuit

AI-generated content may be incorrect.

Simplified form:





A diagram of a circuit

AI-generated content may be incorrect.

A diagram of a circuit

AI-generated content may be incorrect.