**Problem no.02**

**Problem specification:**

Derive the equations for a 3-bit gray to binary converter from Truth table and implement

those with the required gates.

**Required Instruments:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Model** | **Quantity** |
| **01** | **Logisim Software** |  |  |
| **02** | **IC(Hex-Invertor)** | **74LS04** | **01** |
| **03** | **IC (Quad 2 input AND)** | **74LS08** | **03** |
| **04** | **IC (Quad 2 input OR)** | **74LS32** | **01** |
| **05** | **Wires** |  |  |
| **06** | **Input Pin** |  | **03** |
| **07** | **Output Pin** |  | **03** |

**Truth table:**

Truth table for the 3-bit gray code to binary representation conversion is presented below. Here ABC is gray code and XYZ is its binary representation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **X** | **Y** | **Z** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **1** | **1** | **1** |

**Required Equation:**

**Equation for X:**

X = ABC’ + ABC + AB’C + AB’C’

= AB(C+C’) + AB’(C+C’)

= AB + AB’

= A(B+B’)

= A

**Equation for Y:**

Y = A’BC + A’BC’ + AB’C + AB’C’

= A’B (C + C’) + AB’ (C + C’)

= A’B + AB’

=A ⊕B

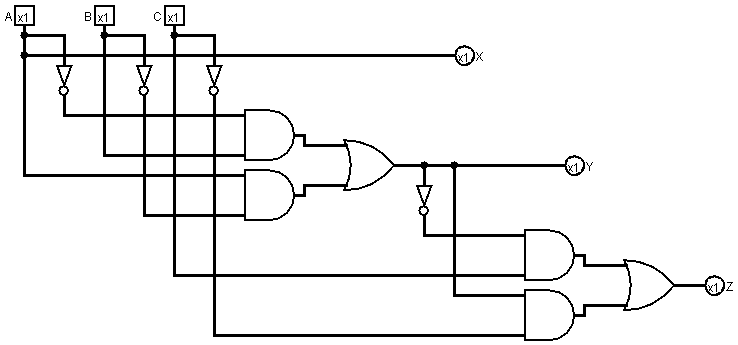
**Equation for Z:**

Z = A’B’C + A’BC’ + ABC + AB’C’

= A’ (B ⊕ C) + A (B ⊕ C)’

= A ⊕ (B ⊕ C)

**Circuit Diagram:**



**Observation:**

We tried to make the circuit such a way that it was not too dense with wires and we also used the documentations of the ICs to make sure the connections were given through the right pins. We checked the output according to the truth table.

**Problem no.03**

**Problem specification:**

Derive the truth table and corresponding output equations for the given condition and implement those with the required gates.

Condition: There are 3 inputs into a system. The system will glow LED1 and LED0 in such a way that the pattern represents the number of set bits in the input.

**Required Instruments:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Model** | **Quantity** |
| **01** | **Logisim Software** |  |  |
| **02** | **IC(Hex-Invertor)** | **74LS04** | **1** |
| **03** | **IC (Quad 2 input AND)** | **74LS08** | **2** |
| **04** | **IC (Quad 2 input OR)** | **74LS32** | **2** |
| **05** | **Wires** |  |  |
| **06** | **Input Pin** |  | **3** |
| **07** | **Output Pin** |  | **2** |

**Truth Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **LED1** | **LED0** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** |

**Required Equation:**

**Equation for LED0:**

LED0 = A’B’C + A’BC’ + AB’C’ + ABC

= A’ (B ⊕ C) + A (B ⊕ C)’

= A ⊕ (B ⊕ C)

**Equation for LED1:**

LED1 = A’BC + AB’C + ABC’ + ABC

= BC (A + A’) + A (BC’ + B’C)

= BC + AB’C +ABC’

= C (B + AB’) + ABC’

= C (A+B) + ABC’

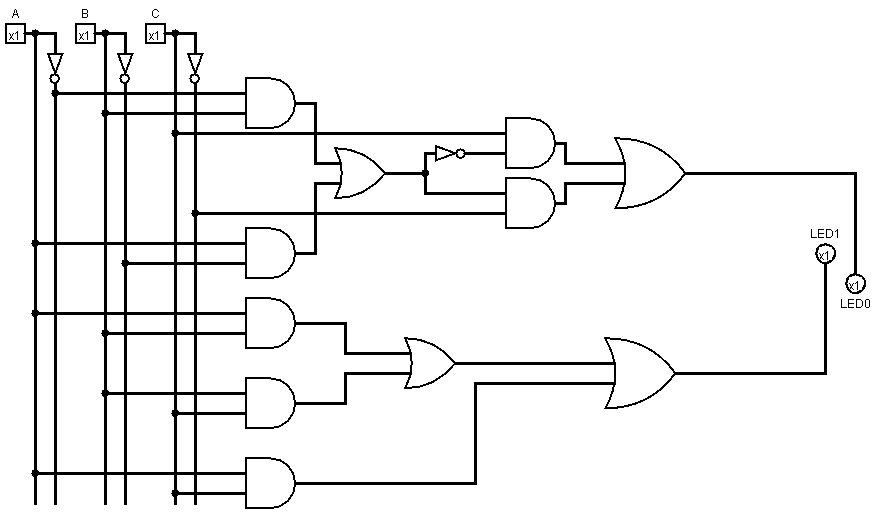
= AC + BC +ABC’

= BC + A (BC’ + C)

= BC + A(B+C)

= AB + BC + AC

**Circuit Diagram:**



**Observation:**

We tried to make the circuit such a way that it was not too dense with wires and we also used the documentations of the ICs to make sure the connections were given through the right pins. We checked the output according to the truth table.