**LAB-11**

**Even/Odd Parity, magnitude comparator and BCD adder**

**Equipment/Components**

**Hardware:** Explorer Board

IC Type 7408 Quadruple 2-input AND gates

IC Type 7486 Quadruple 2-input XOR gate

IC Type 7432 Quadruple 2-input OR gates

IC Type 7404 Hex Inverter

**Software:** Circuit Maker, Waveform

**Description**

In this lab combinational logic circuits are implemented using ‘and’ ‘or’ and ‘not’ gate on Circuit Maker and Explorer Board. After the lab students will be able to understand the functionality of odd and even parity and how to implement the circuit for gray to binary conversion and vice versa.

**Objectives**

* To learn and understand troubleshooting of Combinational logic circuits
* To Implement Binary to Gray code converter on Hardware

**Task # 1**

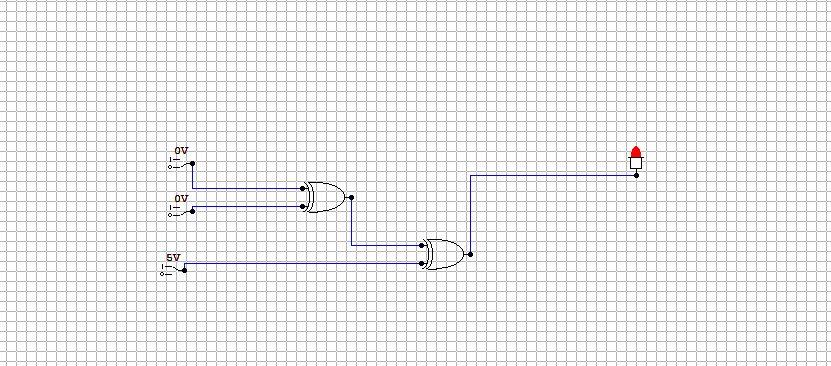
1. Fill the truth table for even parity generator.

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs (Binary) | | | Parity Bit |
| A | B | C | P |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

1. Write simplified expression for output

P= A⊕B⊕C

1. Draw a circuit on Circuit Maker



**Task # 2**

1. Fill the truth table for odd parity generator.

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs (Binary) | | | Parity Bit |
| A | B | C | P |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

1. Draw a **simplified** circuit on Circuit Maker which generates odd parity for three bit binary number.

|  |
| --- |
| Simplified Expressions for each output  P = A⊕ B Ex-NOR C |
|  |

**Task # 3:**

Design a BCD adder which adds two 4-bit binary numbers, tasks are as follows

1. Truth table

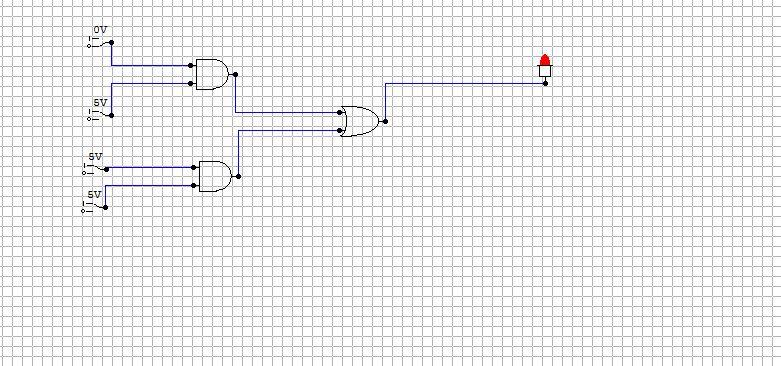
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k | Z4 | Z3 | Z2 | Z1 | C | S4 | S3 | S2 | S1 | DECIMAL |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 5 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 6 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 7 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 9 |

1. Simplified Expression for all the output bits and carry

K+Z3.(Z2+Z1)+Z3.Z1

K+Z3.Z2+Z3.Z1

1. Draw it on circuit maker



**Task # 4:**

Design a 2 bit comparator, tasks are as follows

1. Truth table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **INPUT** |  |  |  | **OUTPUT** |  |  |
| **#** | **A1** | **A0** | **B1** | **B0** | **A>B** | **A=B** | **A<B** |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

1. Simplified Expression

**A>B :**

**A1.B0’.B1’+ A0.B0’+A0.A1.B1’**

**A=B:**

**A0** Ex-NOR **B0**

**A1** Ex-NOR **B1**

**A<B**

**A0’.B0+A1’.B0.B1+A0’.A1’.B1**

1. Draw it on circuit maker

