|  |  |  |
| --- | --- | --- |
| **American University of Sharjah**  **College of Engineering**  Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE | A picture containing logo  Description automatically generated | **Lab Instructor:** Eng. Donthi Sankalpa  **Office:** ESB – 1036C  **Phone**: 971-6-5154826  **e-mail**: [dsankalpa@aus.edu](mailto:DSANKALPA@aus.edu)  **Semester**: Fall 2023 |

Lab #4 - Simplification of Boolean Functions Using Karnaugh Map

**Objectives**

1. Study and demonstrate the use of basic logic gates: NAND, NOT, AND, and OR.
2. To use Karnaugh map to simplify Boolean expressions.

**Lab Equipment and Circuit Components**

**a. Equipment**

* + - NI ELVIS iii Board
    - Connectors.

**b. Circuit Components**

* + - 7400, 7404, 7408, 7432 chips

**Introduction:**

Karnaugh maps or K-maps for short provide another means of simplifying and optimizing logical expressions. This is a graphical technique that utilizes a sum of product (SOP) form or product of sums.

In order to use a K-map, the truth table for a logical expression is transferred to a K-map grid. The grid for three and four input expressions are provided in the figures below. Each cell corresponds to one row in a truth table or one given state in the logical expression. The order of the items in the grid is not random at all; they are set so that any adjacent cell differs in value by the change in only one variable. Because of this, items can be grouped together easily in rectangular blocks of two, four, and eight to find the minimal number of groupings that can cover the entire expression. Note that diagonal cells require that the value of more than two inputs change, and hence are not considered neighbors and therefore cannot be combined to form larger cells.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C’D’ 0 0 | C’D  0 1 | CD  1 1 | CD’  1 0 |
| A’B’  0 0 |  |  |  |  |
| A’B  0 1 |  |  |  |  |
| AB  1 1 |  |  |  |  |
| AB’  1 0 |  |  |  |  |
| Figure 1. 4-input K-map | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | B’C’ 0 0 | B’C  0 1 | BC  1 1 | BC’  1 0 |
| A’  0 |  |  |  |  |
| A  1 |  |  |  |  |
| Figure 2. 3-input K-map | | | | |

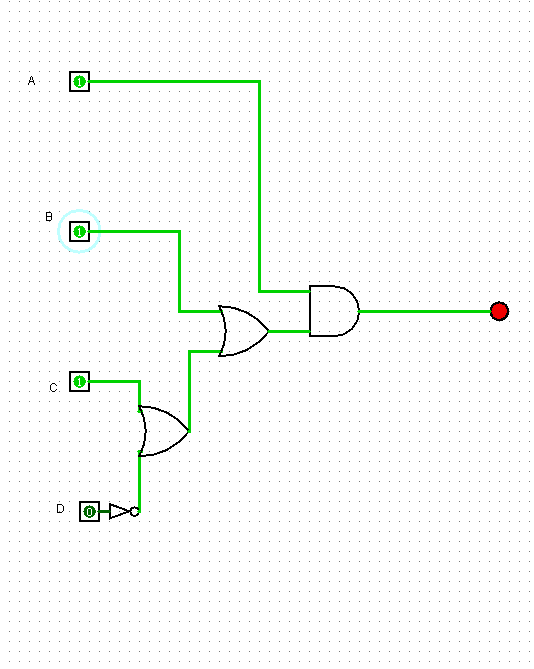
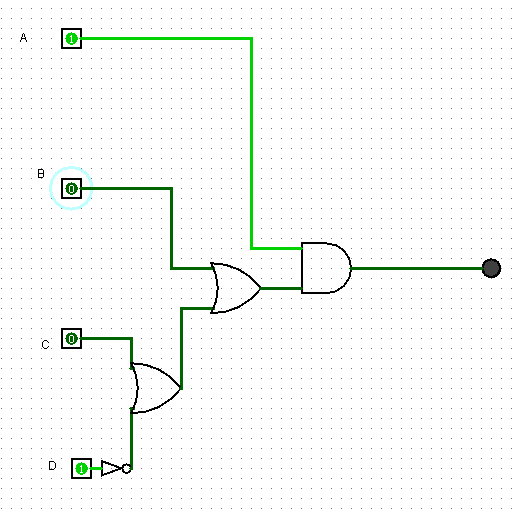
**Lab Assignment**

**Question 1:**

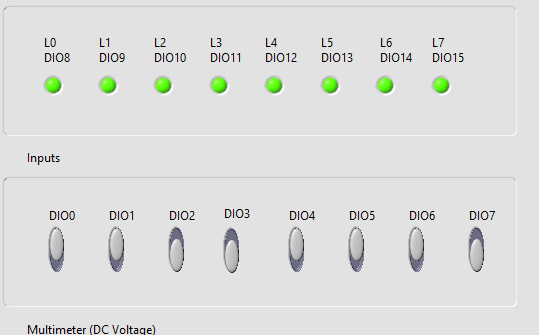
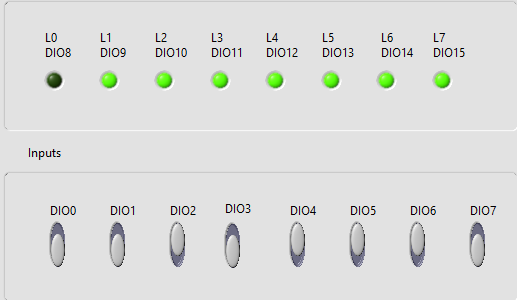
From the K-map in Figure 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C’D’ 0 0 | C’D  0 1 | CD  1 1 | CD’  1 0 |
| A’B’  0 0 | 0 | 0 | 0 | 0 |
| A’B  0 1 | 0 | 0 | 0 | 0 |
| AB  1 1 | 1 | 1 | 1 | 1 |
| AB’  1 0 | 1 | 0 | 1 | 1 |
| Figure 3. F (A, B, C, D) | | | | |
| AB + AC + AD’ = A(B+C +D’) | | | | |
|  | | | | |

1. Simulate the simplified circuit on LogiSim.

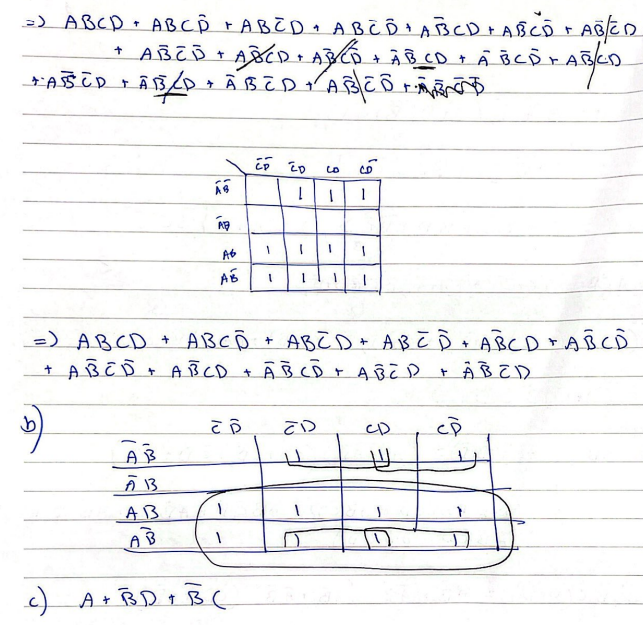
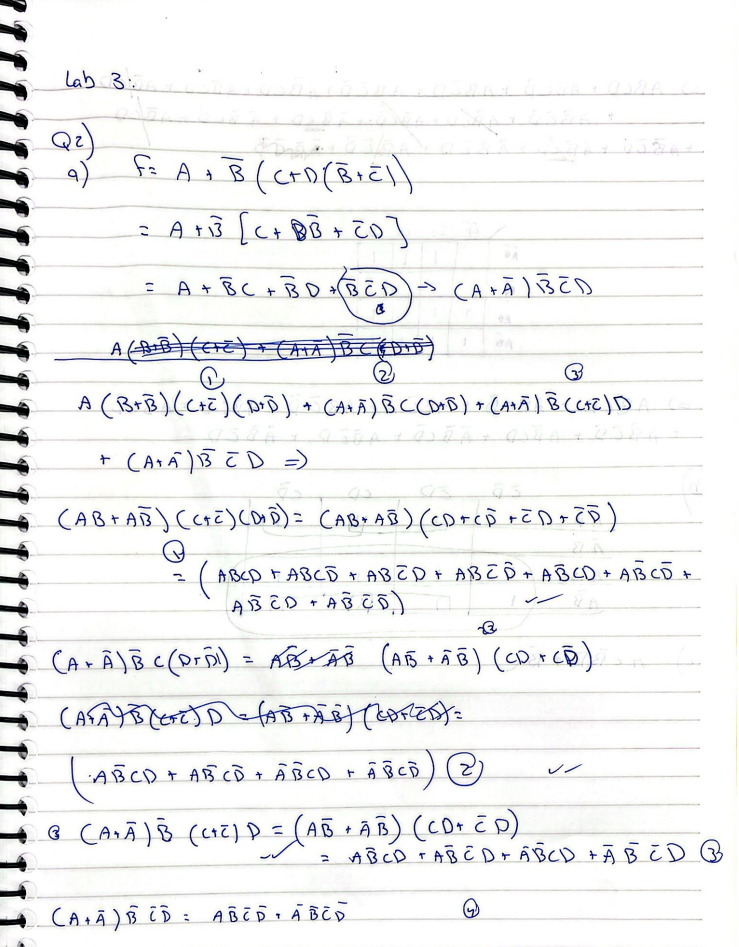
1. Implement the simplified circuit on the ELVISS board.

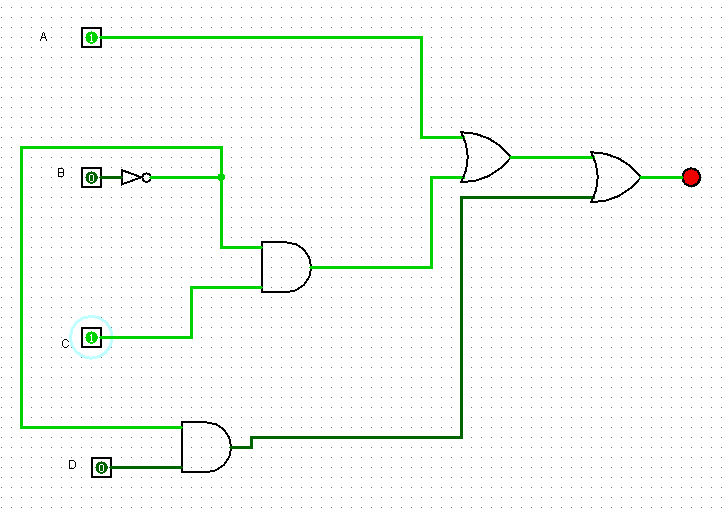
**Question 2:**

From the Expression

1. Convert the expression into standard form (let’s choose SOP form).



1. Draw the K-map of this function.
2. Use the K-map to get the simplified SOP form (essential prime implicant terms), show your grouping.
3. Simulate the simplified circuit on LogiSim.



1. Implement the simplified circuit on the ELVISS board.

