**Department of Electrical Engineering**

**Faculty Member:** Arshad Nazir **Dated:** 30/10/2021

**Semester:** 3rd **Section:** BEE 12C

**Group No.: 7**

**EE-221: Digital Logic Design**

Lab 5: Minimization of Boolean Functions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4/CLO4** | **PLO4/CLO4** | **PLO5/CLO5** | **PLO8/CLO6** | **PLO9/CLO7** |  |
| **Name** | **Reg. No** | **Viva / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** | **Total marks Obtained** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **25 Marks** |
| **Muhammad Adil Azeem** | 342459 |  |  |  |  |  |  |
| **Muhammad Umer** | 345834 |  |  |  |  |  |  |
| **Raahim Ahmad Waqar** | 342287 |  |  |  |  |  |  |
| **Saad Bakhtiar** | 341150 |  |  |  |  |  |  |
| **Syed Aun Ali Kazmi** | 342384 |  |  |  |  |  |  |

## Lab 5: Minimization of Boolean Functions

This Lab has been divided into two parts.

The first part is the hardware implementation of a Boolean function given to you. But you have to first minimize the Boolean functions to minimum number of literals.

In next part you will simulate the same circuit using Verilog.

**Objectives:**

* Understand Minimization of Boolean Functions
* Simulate Basic Circuits using Verilog
* Hardware Implementation of Basic Logic Circuits

**Lab Instructions**

* This lab activity comprises three parts, namely Pre-lab, Lab tasks, and Post-Lab Viva session.
* The lab report will be uploaded on LMS three days before scheduled lab date. The students will get hard copy of lab report, complete the Pre-lab task before coming to the lab and deposit it with teacher/lab engineer for necessary evaluation. Alternately each group to upload completed lab report on LMS for grading.
* The students will start lab task and demonstrate design steps separately for step-wise evaluation ( course instructor/lab engineer will sign each step after ascertaining functional verification)
* Remember that a neat logic diagram with pins numbered coupled with nicely patched circuit will simplify trouble-shooting process.
* After the lab, students are expected to unwire the circuit and deposit back components before leaving.
* The students will complete lab task and submit complete report to Lab Engineer before leaving lab
* There are related questions at the end of this activity. Give complete answers.

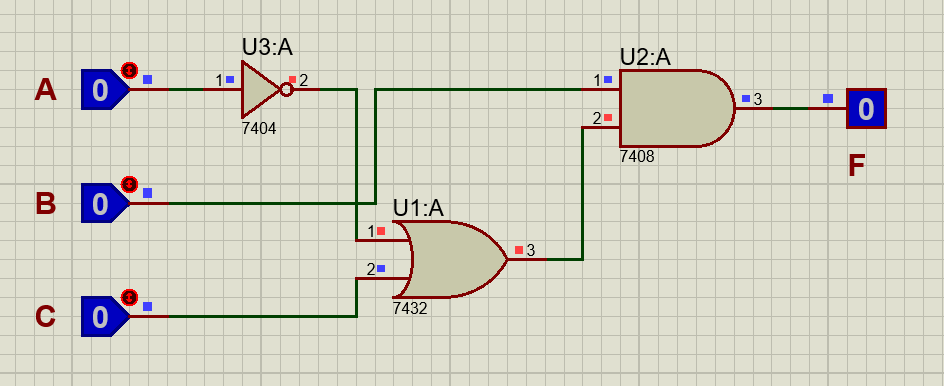
**Pre – Lab Tasks**

**Pre – Lab Task 1:**

Write the Boolean expression of the following two functions. Simplify the expression using algebraic manipulation and draw the **logic diagram**.

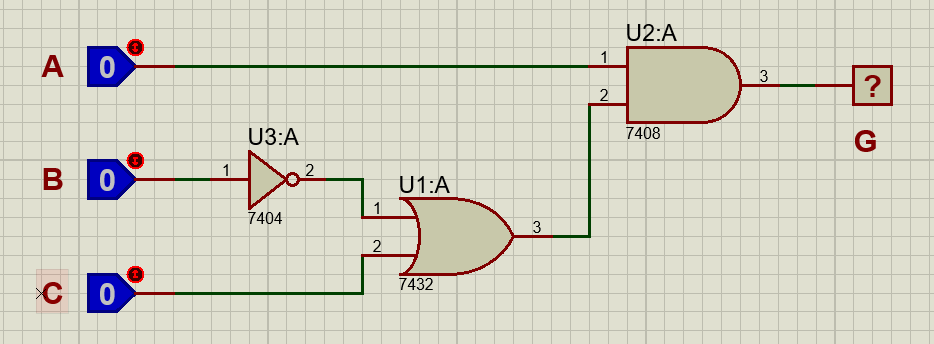
1. **F (A, B, C) = ∑ (2, 3, 7)**

* A’BC’ + A’BC + ABC
* A’BC’ + BC(A + A’)
* A’BC’ + BC(1)
* A’BC’ + BC
* B(A’C’+C)
* B(A’+C)
* A’B+BC
* B(A’+C)

****

1. **G (A, B, C) = ∑ (4, 5, 7)**

* AB’C’ + AB’C + ABC
* AB’C’ + AC(B’+B)
* AB’C’ + AC(1)
* AB’C’ + AC
* A(B’C’+C)
* A(B’+C)
* AB’+AC
* **A(B’+C)**

****

**Pre – Lab Task 2:**

Mention the number of literals and gates needed for implementing the above function in hardware.

* **F:** 4 Literals, Two 2-I AND Gates, One 2-I OR Gate, 1 NOT Gate
* **G:** 4 Literals, Two 2-I AND Gates, One 2-I OR Gate, 1 NOT Gate

**Lab Tasks**

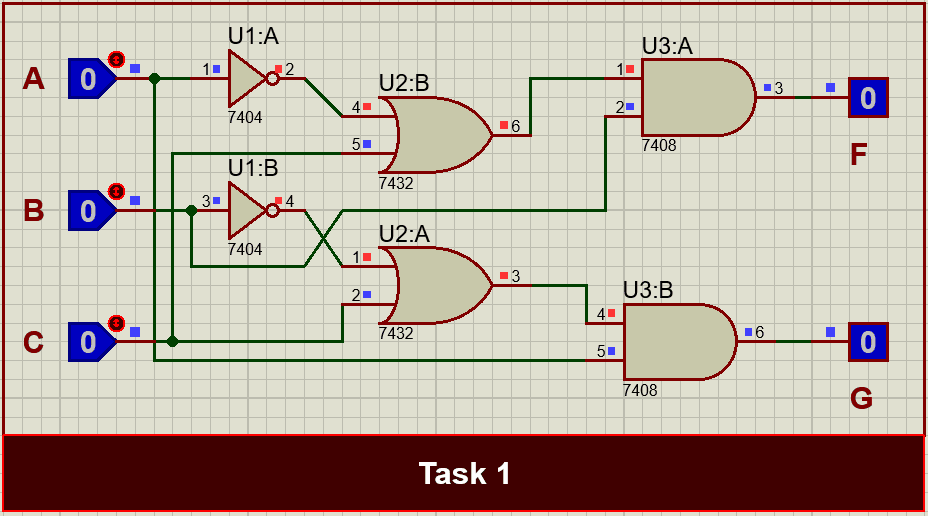
**Lab Task 1:**

Implement the Boolean functions in hardware you simplified in your Pre-Lab Task. Make truth table and **Schematic**. Mention what and how many gates you would be using? The following gates are available to you.

**Truth Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **F** | **G** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**Proteus Simulation (Schematic)**



**Hardware Implementation**



**Lab Task 2:**

Write Verilog code for the minimized functions at gate-level and perform simulation. Attach the relevant snapshots below.

module task1(f, g, a, b, c);

    input a, b, c;

    output f, g;

    wire w1, w2, w3, w4;

    not n1(w3, a);

    not n2(w4, b);

    or o1(w1, w3, c);

    and a1(f, w1, b);

    or o2(w2, w4, c);

    and a2(g, w2, a);

endmodule

module testbench();

    reg A, B, C;

    wire F, G;

    task1 t1(F, G, A, B, C);

    initial begin

        #100 A = 0; B = 0; C = 0;

        #100 A = 0; B = 0; C = 1;

        #100 A = 0; B = 1; C = 0;

        #100 A = 0; B = 1; C = 1;

        #100 A = 1; B = 0; C = 0;

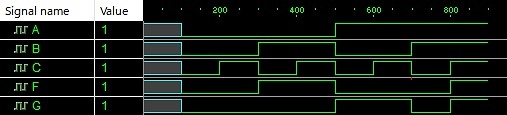
        #100 A = 1; B = 0; C = 1;

        #100 A = 1; B = 1; C = 0;

        #100 A = 1; B = 1; C = 1;

    end

endmodule

****

**Dataflow Modeling**

module task1(f, g, a, b, c);

    input a, b, c;

    output f, g;

    assign {f} = !a\*b\*!c + !a\*b\*c + a\*b\*c;

    assign {g} = a\*!b\*!c + a\*!b\*c + a\*b\*c;

endmodule

module testbench();

    reg A, B, C;

    wire F, G;

    task1 t1(F, G, A, B, C);

    initial begin

        #100 A = 0; B = 0; C = 0;

        #100 A = 0; B = 0; C = 1;

        #100 A = 0; B = 1; C = 0;

        #100 A = 0; B = 1; C = 1;

        #100 A = 1; B = 0; C = 0;

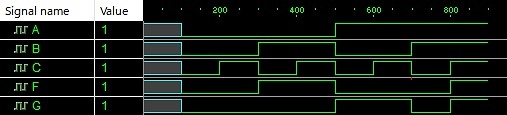
        #100 A = 1; B = 0; C = 1;

        #100 A = 1; B = 1; C = 0;

        #100 A = 1; B = 1; C = 1;

    end

endmodule

****

**Observations/Comments:**

In this lab, we implemented a logical circuit that is more optimal than its original counterpart, i.e. we simplified the sum of standard minterms to reduce and minimize the use of gates, essentially making it both efficient and cost effective.

We have also confirmed/verified the circuit’s output as it conforms to the truth table laid out in the Pre – lab and that it serves the purpose for what it was designed conclusively.