**Department of Electrical Engineering**

**Faculty Member:** Arshad Nazir **Dated:** 15/10/2021

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

**Group No.: 7**

**EE-221: Digital Logic Design**

Lab 4: Design and Implementation of Simple Practical Circuits with Full Adder

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **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 4: Design and Implementation of Simple Practical Circuits with Full Adder

This Lab has been divided into two parts.

In first part you are required to design and implement simple practical circuits.

In the second part you are required to write Verilog code for the practical circuit

**Objectives:**

* Design the Simple practical circuits using digital logic Hardware Implementation of Basic Logic Circuits
* Verilog coding of the practical circuit

**Lab Instructions**

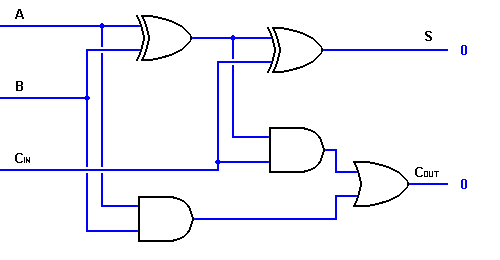
* This lab activity comprises three parts, namely Pre-lab, Lab tasks, and Post-Lab Viva session.
* The lab report will be uploaded on LMS three days before scheduled lab date. The students will get hard copy of lab report, complete the Pre-lab task before coming to the lab and deposit it with teacher/lab 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.

**Lab Tasks**

1. **Lab Task 1:**

Implement the Full Adder Circuit given below, make its truth table and Schematic Diagram. Show your results to the lab engineer.

**Logic Diagram**

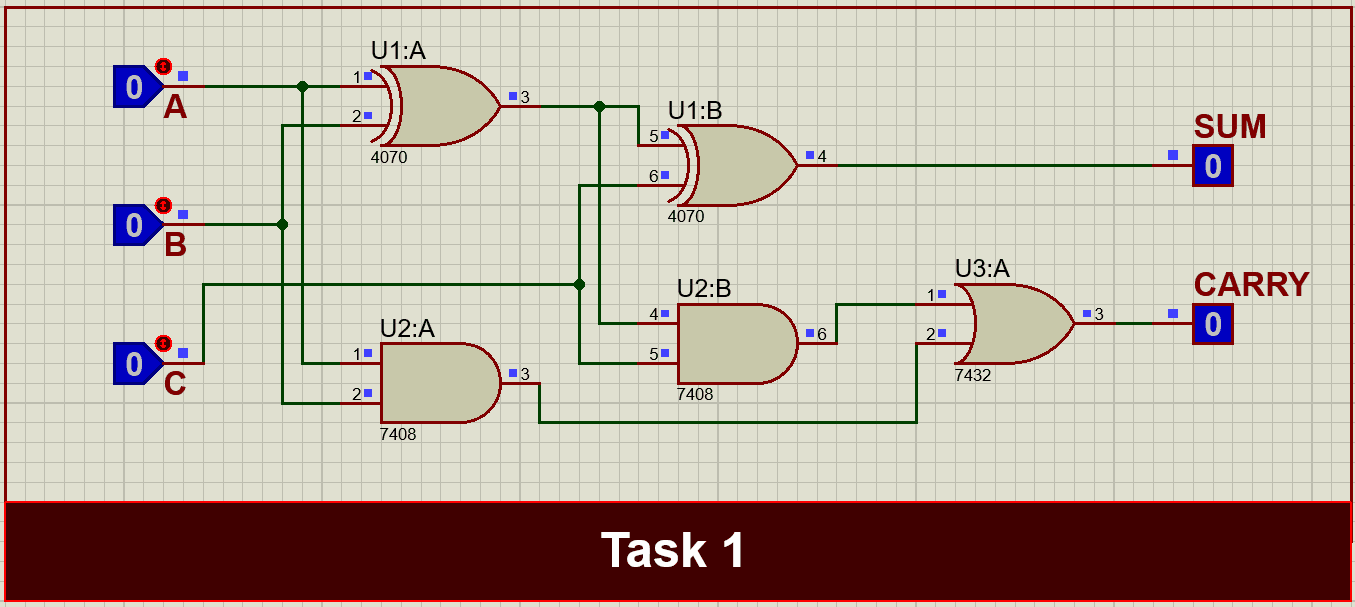


**Truth Table**

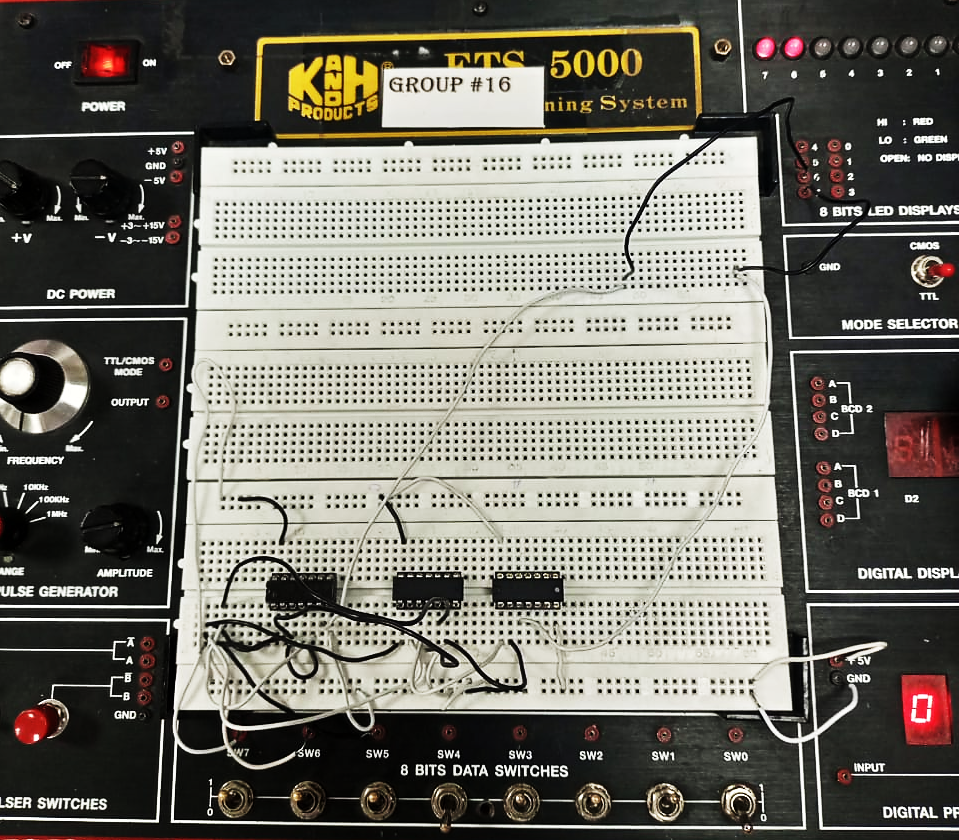
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **Cin** | **Cout** | **S** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** |

Which are the results we would expect from a FULL ADDER circuit, and to ensure integrity, we have implemented the given schematic on Proteus as well

**Proteus Simulation**

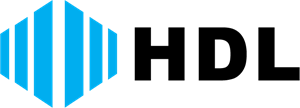


**Hardware Implementation**



1. **Lab Task 2:**

Design and simulate the gate-level model of the circuit you patched in task 1. Give the code (including test bench) and resulted waveform in the space provided below.



module task1(sum, carry, a, b, c);

    input a, b, c;

    output sum, carry;

    wire w1, w2, w3, w4, w5;

    xor x1(w1, a, b);

    xor x2(w3, w1, c);

    and a1(w2, a, b);

    and a2(w4, w1, c);

    or o1(w5, w4, w2);

    assign {carry} = w5;

    assign {sum} = w3;

endmodule

module testbench();

    reg A, B, C;

    wire SUM, CARRY;

    task1 t1(SUM, CARRY, A, B, C);

    initial begin

        #100ps A=0; B=0; C=0;

        #100ps A=0; B=0; C=1;

        #100ps A=0; B=1; C=0;

        #100ps A=0; B=1; C=1;

        #100ps A=1; B=0; C=0;

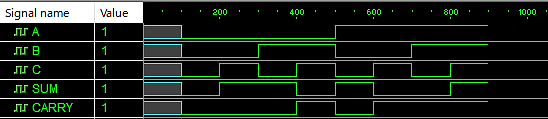
        #100ps A=1; B=0; C=1;

        #100ps A=1; B=1; C=0;

        #100ps A=1; B=1; C=1;

    end

endmodule

****

**Alternative HDL Implementation (Calling Half-Adder Twice)**

module halfadder(out1, out2, a, b);

    input a, b;

    output out1, out2;

    xor x1(out1, a, b);

    and a1(out2, a, b);

endmodule

module testbench;

    reg A, B, C;

    wire SUM, CARRY;

    halfadder h1(OUT1, OUT2, A, B);

    halfadder h2(SUM, OUT4, OUT1, C);

    or o1(CARRY, OUT4, OUT2);

    initial begin

        #100ps A=0; B=0; C=0;

        #100ps A=0; B=0; C=1;

        #100ps A=0; B=1; C=0;

        #100ps A=0; B=1; C=1;

        #100ps A=1; B=0; C=0;

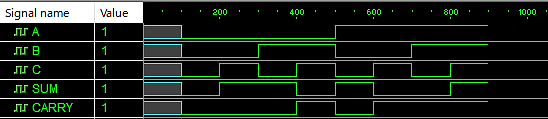
        #100ps A=1; B=0; C=1;

        #100ps A=1; B=1; C=0;

        #100ps A=1; B=1; C=1;

    end

endmodule

****

**Observations/Comments:**

In this lab, we deployed the concepts learned in Lab 2, *Implementation of a Half Adder circuit*, to implement a Full Adder. Two Half Adders concatenated together and through use of an additional OR gate give us a Full Adder circuit, essentially implying that:

**Full Adder = Half Adder + Half Adder**

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.