Digital Logic Design Lab
Date: _____
Total Marks: 30

Lab Manual _____
Roll Number_____

HDL Models for Combinational Logic Circuits

Objective:

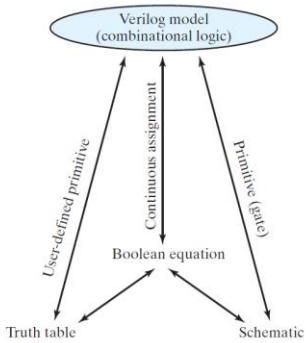
In this lab students will learn how to implement

- Design and implement combinational circuits (Decoder, Adder, Ripple Carry Adder, Comparator, MUX) using Verilog.
- Apply behavioral modeling techniques (always, case, if-else, procedural assignments) in Verilog code.
- Simulate and verify digital circuit functionality using appropriate testbenches.
- Understand structural relationships between basic modules (e.g., using adders to build ripple-carry adder).

Brief Theory Description of the Lab

This lab focuses on implementing essential combinational logic circuits using Verilog Hardware Description Language (HDL). Students design modules for a decoder, half/full adder, ripple carry adder, comparator, and multiplexer using behavioral modeling approaches. Through these exercises, students learn how to express digital logic using always blocks, case statements, and procedural assignments.

The lab emphasizes understanding how basic combinational circuits function and how they can be interconnected to form more complex systems—for example, using multiple full adders to construct a ripple carry adder. Students also develop testbenches to simulate each design, observe waveform outputs, and verify that the circuits behave as expected. By completing this lab, students gain confidence in writing Verilog code, interpreting simulation results, and understanding the practical implementation of fundamental digital logic components.



This picture illustrates the different ways a combinational logic circuit can be represented and implemented in Verilog. It shows that a single Verilog model can be created starting from multiple equivalent descriptions of the same logic.

Brief Explanation

The diagram shows that combinational logic can be described in Verilog using several approaches:

- Truth Table → can be converted into a User-Defined Primitive (UDP) in Verilog.
- Boolean Equation → can be written using continuous assignment statements (assign) in Verilog.
- Schematic or Gate Diagram → can be implemented using primitive gate-level modeling (e.g., and, or, not).

All these representations—truth table, Boolean expressions, and gate schematics—ultimately describe the same underlying combinational logic. Verilog allows the designer to choose any of these modeling styles, and each of them can be transformed into a valid Verilog model.

Exercise

Write down the Verilog HDL code for designing 2 to 4-line decoder circuit using Gate level description or data flow modeling. [5]

Digital Logic Design Lab
Date: _____
Total Marks: 30

Lab Manual _____
Roll Number_____

Test Bench Code

Write down the Verilog code for designing a full adder circuit using the concept of instantiation of two half adders. Also write code for its test bench waveforms. [5]

Digital Logic Design Lab
Date: _____
Total Marks: 30

Lab Manual _____
Roll Number_____

Write code for designing four bit ripple carry adder by instantiating the chain of four full adders. [5]

Digital Logic Design Lab
Date: _____
Total Marks: 30

Lab Manual _____
Roll Number_____

Write code for designing four bit comparator circuit in Verilog HDL. Also write code for its test bench waveform. [5]

Write Verilog code for giving the Behavioral description of four-to-one line multiplexer. Also write code for testing its waveform on test bench. [10]

Digital Logic Design Lab
Date: _____
Total Marks: 30

Lab Manual _____
Roll Number_____