Digital Logic Design Lab Manual

## Lab#1: Introduction to Lab Equipment

1. What is an integrated circuit?
2. What is a logic probe? Purpose of logic probe? How is a probe used?
3. What is a breadboard? What is the purpose of this board? How is a breadboard used?

## Lab#2: Using Debugger

1. Introduction to debugger
2. Using debugger to view contents of registers.

## Lab#3: Using basic Logic gates

1. Test the functionality of 2-Input AND Gate.
2. Test the functionality of 2-Input OR Gate.
3. Test the functionality of 2-Input NAND Gate.
4. Test the functionality of 2-Input NOR Gate.
5. Test the functionality of 2-Input XOR Gate.
6. Test the functionality of 2-Input XNOR Gate using XOR and NOT gates.
7. Test the functionality of 2-Input NOR Gate using OR and NOT gates.
8. Test the functionality of 2-Input NAND Gate using AND and NOT gates.
9. Test the functionality of 2-Input OR Gate using NOR and NOT gates.
10. Test the functionality of 2-Input AND Gate using NAND and NOT gates.
11. Test the functionality of 3-Input NAND Gate.
12. Test the functionality of 3-Input NAND Gate using 2-Input NAND gates.

## Lab#4: Using Universal Gate (NAND)

1. Implement the XOR function F(x,y) = xy’ + x’y using AND, OR, and NOT gates.
2. Implement the XNOR function F(x,y) = xy + x’y’ using AND, OR, and NOT gates.
3. Prove that 2-input AND-Invert produces the same result as 2-Input Invert-OR.
4. Implement a NOT gate using NAND Gate.
5. Implement a 2-input AND gate using NAND Gates.
6. Implement a 2-input OR gate using NAND Gates.
7. Implement a 2-input XOR gate using NAND Gates.
8. Implement a 2-input XNOR gate using NAND Gates.

## Lab#05: Using Universal Gate (NOR), K-Map Reduction

1. Implement the XOR function F(x,y) = xy’ + x’y using NOR gates only.
2. Implement the XNOR function F(x,y) = xy + x’y’ using NOR gates only.

**Consider the following Boolean expression for the remaining experiments**

1. Make truth table for the expression and implement the circuit using AND, OR, and NOT Gates.
2. Reduce the expression using 3-variable K-Map. Make the truth table for the reduced expression. Implement the circuit using AND, OR, and NOT Gates. Verify that the results of experiment 3 and 4 are same.
3. Provide NAND-ONLY implementation for the reduced expression from Exp 4. Prove that the results from Exp 4 and 5 match.
4. Provide NOR-ONLY implementation for the reduced expression from Exp 4. Prove that the results from Exp 4 and 6 match.

## Lab#06: Implementation of functions using logic gates

**Consider the following Boolean expression**

1. Make truth table for the expression and Provide AND-NOR implementation of the expression.
2. Provide NAND-AND implementation of the expression, verify that the results of experiment 1 and 2 are same.
3. Provide OR-NAND implementation of the expression; verify that the results of exp 2 and 3 are same.
4. Provide NOR-OR implementation of the expression, verify that the results of exp 3 and 4 are same.

## Lab#07: Adders

1. Design and implement the circuitry for a Half adder.
2. Implement a half adder using AND, OR and NOT gates only.
3. Design and implement the circuitry of a full adder.
4. Design a simplest circuit that determines how many of the bits in a 3-bit unsigned number are equal to 0.

## Lab#08: Subtractors and Code Converters

1. Design and implement the circuitry for a Full Subtractor.
2. Design and implement the circuitry for a BCD-to-Excess 3 Code Converter.

## Lab#09: Magnitude Comparator and Decoder

1. Design and implement the circuitry for a 2-bit magnitude comparator.
2. Design and implement the circuitry for a 2 to 4 decoder, with active-high enable.

## Lab#10: Decoders and Multiplexers

1. Design and implement the circuitry for a 4 to 1 Multiplexer.
2. Design and implement 3-to-8 decoder using two 2-to-4 decoders of the IC 74LS139.
3. Using the circuitry of Exp 2, implement a full-subtractor.
4. Using the circuitry of Exp 2, implement a full-Adder.

## Lab#11: Encoders and Multiplexers

1. Test the functionality of a 1x4 Demultiplexer using the IC-74LS139
2. Implement a half adder using IC-74LS139
3. Implement a half subtractor using IC-74LS139.
4. Test the functionality of a 8x1 multiplexer using the IC-74LS151
5. Test the functionality of a 4x1 multiplexer using the IC-74LS153

## Lab#12: Implementation of SR and D latches

1. Implement an SR-latch using NOR-gates.
2. Implement an SR-latch using NAND-gates.
3. Implement an SR-latch with enable using NAND-gates.
4. Implement a D-latch using the circuitry in used in Exp 3.

## Lab#13: Flipflops

1. Test the functionality of an SR-Latch using the IC-74LS279.
2. Test the functionality of an edge triggered D-flipflop using IC-74LS74.
3. Test the functionality of a Master Slave JK-flipflop using IC-74LS74.

## Lab#14: Registers and Counters

1. Test the functionality of a 4-bit shift register using the IC-74LS95.
2. Test the functionality of a 4-bit binary counter using the IC-74LS93.
3. Test the functionality of a Ring counter using the IC-74LS164.