Digital Logic Design
(EL-1005)
LABORATORY MANUAL
Spring-2025

LAB 01
Introduction to DLD, Lab Tools & Basic Logic Gates

STUDENT NAME ROLL NO SEC

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /10

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI

Lab Session 01

OBJECTIVES:

The objectives of this lab are:

- To familiarize with the Lab Hardware and Software Tools.
- To learn and understand the working of basic gates (AND, OR, NOT).
- To learn and understand troubleshooting of logic circuits
- Identify gates by their symbols
- Write logical expressions of gates and draw their truth tables.

Introductory Concepts:

Digital circuits have two discrete voltage levels to represent the binary digits *(bits)* 1 and 0. All digital circuits are switching circuits that use high-speed transistors to represent either an ON condition or an OFF condition.

Hardware Tools:

Below are some important Hardware tools which will be used throughout the labs.

The Power Supply:

For any integrated circuit (IC) to function properly, power and ground must be connected. All active electronic devices, such as the integrated circuits used in digital electronics, require a stable source of dc voltage to function properly. The power supply provides the proper level of dc voltage. It is very important that the correct voltage be set before connecting it to the ICs on your board .For nearly all of the circuits, the power supply should be set to +5.0 V. When testing a faulty circuit, one of the first checks is to verify that the supply voltage is correct and that there is no ac component to the power supply output.

Analogue & Digital Electronic Trainer (ETS-7000):

The ETS-7000 Analogue and Digital Trainer provide a complete digital electronics workstation and have a huge number of features in one compact unit. This trainer is housed in a compact and robust casing, and is equally ideal for constructing and testing both analogue and digital electronics circuits. It has DC Power Supplies, Function Generators for generating various types of Waves with different frequencies, Logic & Pulse Switches, 7-Segment Display and Breadboard etc.

Figure 1: ETS-700 Trainer

Logic Probe

Another handheld instrument that is useful for tracing simple logic circuits is the logic probe. The logic probe can be used to determine the logic level of a point in a circuit or to determine whether there is pulse activity at the point by LED (light-emitting diode) displays. A simple logic probe can determine if logic levels are HIGH, LOW, or INVALID.

Figure 2: Logic Probe

Protoboards/ Breadboards:

Protoboards/Breadboards are a convenient way to construct circuits for testing and experimenting. A breadboard is used to make up **temporary circuits** for testing or to try out an idea. No soldering is required so it is easy to change connections and replace components. Parts will not be damaged so they will be available to re-use afterwards. The photograph shows a circuit on a typical small breadboard which is suitable for beginners building simple circuits with one or two ICs

IC (Integrated Circuit):

Integrated Circuit usually called as a chip or microchip is a small electronic device made up of semiconductor material. ICs contain the electronic components for the digital gates. The chip is mounted in a ceramic or plastic container and connections are welded to external pins to form the IC. The number of pins may range from 14 in a small IC package to 64 or more in a larger IC. The following figures show an IC with 14 pins, both the Device package and the pin layout in its top view. If you are looking at the IC from above you will see a circle or notch. This is the top of the chip. The top-left pin is almost always Pin 1. In coming labs we will be using various series of ICs for learning the functionality of all kinds of gates like NOT, AND, OR gates etc and will also implement circuits of Binary Adder, Subtractor and Multiplexers etc.

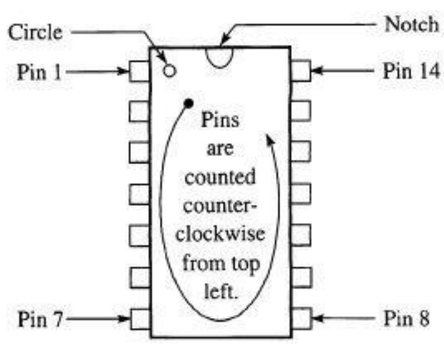


Figure 3: General IC Package

Basic Circuit Connection Procedure:

The components used to build digital circuits are very delicate and can easily be damaged if not handled properly. Following are the steps to build a digital circuit:

- Turn power switch OFF.
- Place diode / transistor / IC chip on the breadboard. Make sure their pins are not short circuited.
- Connect the GND pins of all IC chips to ground.
- Connect the Vcc pins of all chips to +5V on your trainer.
- Connect all other input and output signals
- Turn power ON.

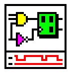
Precautions:

- Handle components from their top plastic part. Avoid touching the pins.
- Be careful not to connect any circuit output to Vcc or any other output.
- Check IC diagram before making any connections.
- Turn power OFF before you modify the circuit.

<u>Software Tool (LogicWorks):</u>

Logic-Works is an interactive circuit design tool intended for teaching and learning digital logic. It is a program that we can use for designing and simulating circuits. This lab material will introduce you the basic features of Logic-Works and familiarize you with the program by stepping you through the construction of a simple circuit.

Startup Logic-Works:

You start Logic Works by clicking on the icon  or selecting it from the Microsoft Windows Start menu. Once the tool is open, you will see the "Welcome to the Logic-Works" screen.

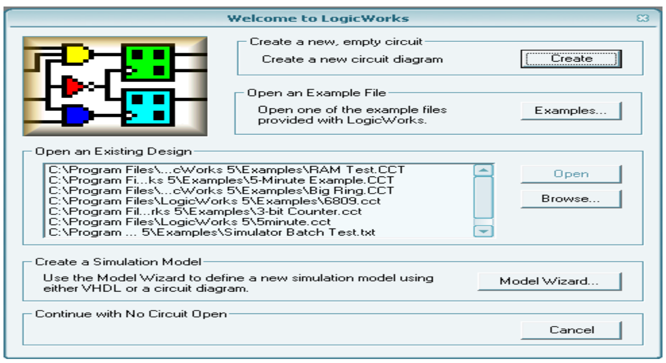


Figure 4: Start-up screen

Select "Create a new circuit diagram" by clicking on the "Create" button. Then the following screen is displayed.

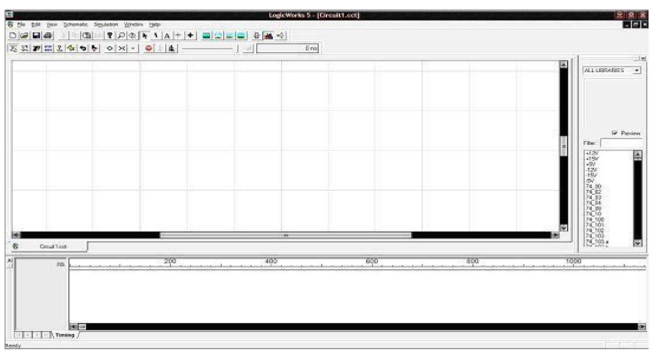


Figure 5: Main Panel

There are three primary window components in Logic-Works.

1. Circuit Window
2. Timing Window
3. Parts Palette.

1. **<u>Circuit Window:</u>**
   The circuit window is where you build your circuits. For example, you can select different parts from the Parts Palette and place the device in the circuit Window to build your circuits. The following is a screen capture of the circuit window with a sample circuit already drawn. You can select any part of the circuit by left clicking on it, once selected, a right click can bring up a pop menu that displays the options for manipulating the selected component.

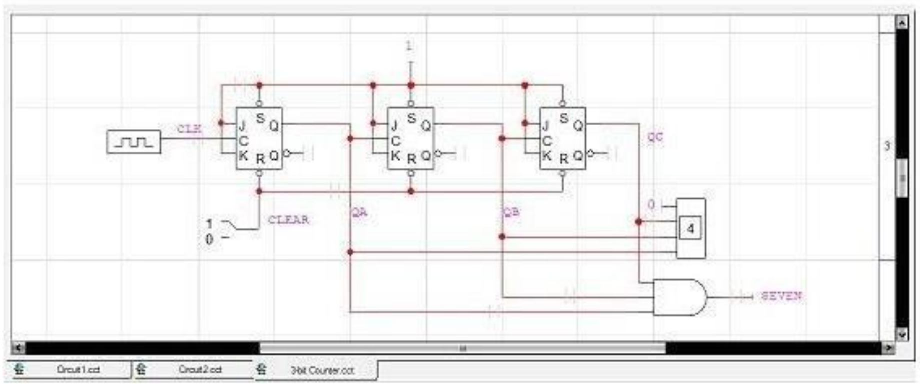


Figure 6: Simple Circuit Diagram

2. **<u>Timing Window:</u>**

   When a circuit is simulated, the Timing diagram window can be displayed to show signal values versus time. Only one Timing window can be displayed at any given time, and it gives waveforms generated by the current design. Closing the Timing window does not close the circuit design file. Like the Circuit window, the intervals of the timing diagram can be selected by using click and drag the mouse over the desired interval. Once a section of the diagram is selected, right click brings up a pop menu with options that can be used to manipulate the Timing diagram.

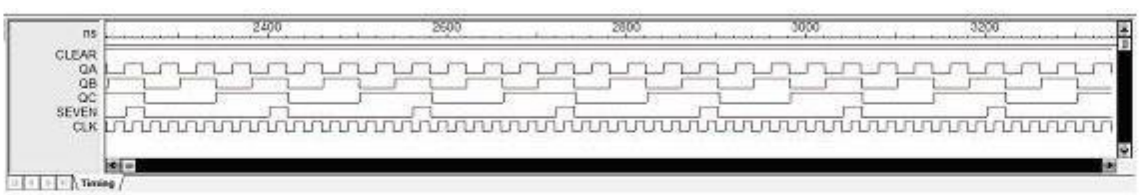


Figure 7: Simple Timing Diagram Example

3. **<u>Parts Palette:</u>**
   Parts Palette is another very important part of Logic-Works interface. By default, it is located to the right of the Circuit Window. The location can be changed by holding down the left mouse button on the double stripped title bar near the top and dragging the window to the desired location. Usually, I would recommend that you keep it in the default location.

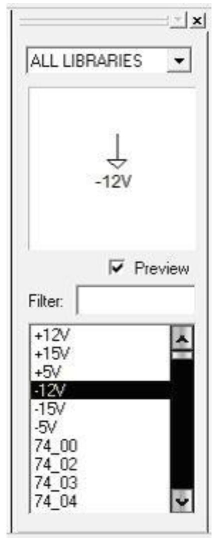


Figure 8: Parts Palette

Building a Simple Circuit:

In this section, you will build and test a circuit that implements the following Boolean equation:

Z = AB + BC + AC'

This requires the following components

- Three AND gates with two inputs (AND-2)
- Two OR gates with two inputs (OR-2)
- A single NOT gate commonly called an inverter
- Three switches to provide a way to modify the input values for testing
- A binary probe to show the results of the circuit

The first three components can be found in the "Simulation Gates.clf" library:

1. Click on the pull-down menu on the Parts Window
2. Click on the "Simulation Gates.clf" library.

Simulation Gates.clf" should now be displayed in the pull-down menu window of the *Parts Window* as shown in Fig. 1. The next portion of the *Parts Window* displays a list of the logic gates contained in this library. The scroll bar can be used to view them all.

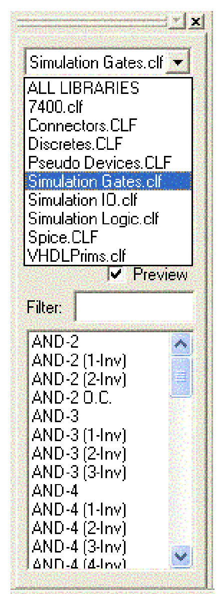


Figure 9: Selection of the libraries

Put the desired devices on the schematic (The Circuit Design Window):

AND Gates:

- Find the AND-2 gate in the Parts Window and double click on it.
- Move the mouse pointer to the Circuit Design Window and place three of these gates where you want them by clicking once for each gate
- Hit Esc/Space so that no more AND gates are selected.

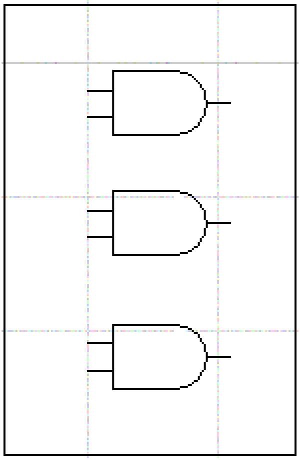


Figure 10: AND Gate

OR Gates:

- Find the OR-2 gate in the *Parts Window* and double click on it.
- Move the mouse pointer to the *Circuit Design Window* and place two of these gates to the right of the AND gates.

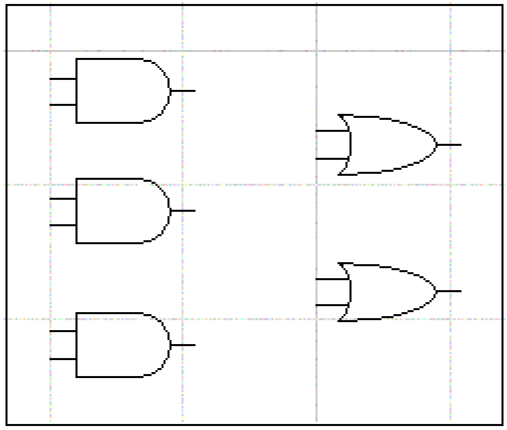


Figure 11: Or Gate Placement

NOT Gates (Inverters):

- Find the NOT gate in the *Parts Window* and double click on it.
- Move the mouse pointer to the *Circuit Design Window* and place one NOT gate to the left of the bottom AND gate.

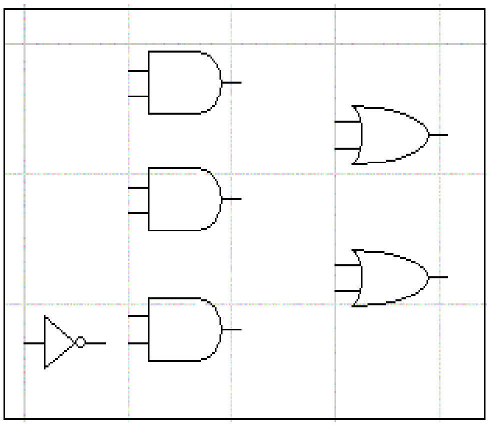


Figure 12: Not Gate Placement

Moving and deleting existing gates:

- By single clicking on an item on the schematic (The Circuit Design Window) the device will be selected and highlighted. While the device is selected, the Delete key will remove the item from the circuit window.
- To move a device, point at it and hold down the left mouse button, then move the mouse to the desired location. Release the mouse button.

Adding switches:

- Click on the pull-down menu in the *Parts Window*.
- Click on the "Simulation IO.clf" library.
- Double click on the binary switch entry and place three of them on the circuit window on the left of your design.
- Double click on the Binary Probe entry and place it on the circuit window on the right of your design.

Your design should now resemble the circuit window in Fig. 5 below.

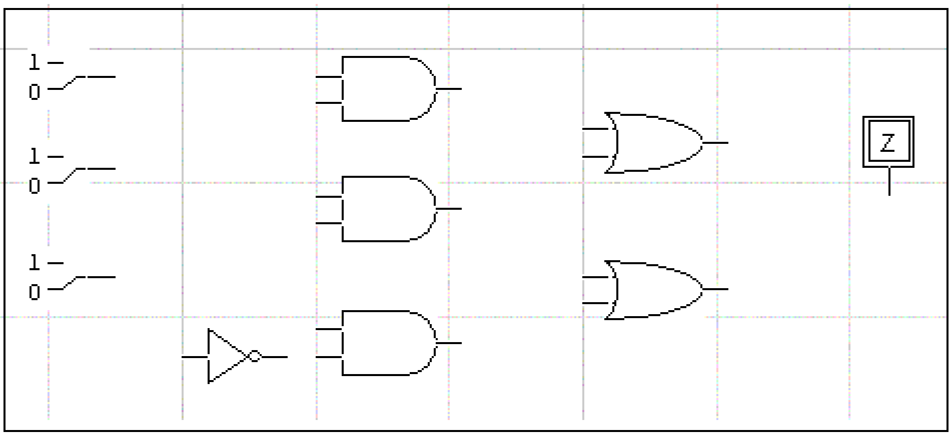


Figure: 13 Binary Switch and Probe Placement

Connecting the Devices

Adding wires:

- Place the cursor on the right edge of the switch and hold down the left mouse button.
- Drag the mouse a half-inch or so to the right and release the button. A red wire should now be attached to the switch, ending in the middle of nowhere.
- Repeat this for each of the three switches. Right-click on the wire you want to name.
- Select Name from the box that appears.
- Type the name of the wire in the text box. Be sure to check Visible as in Fig

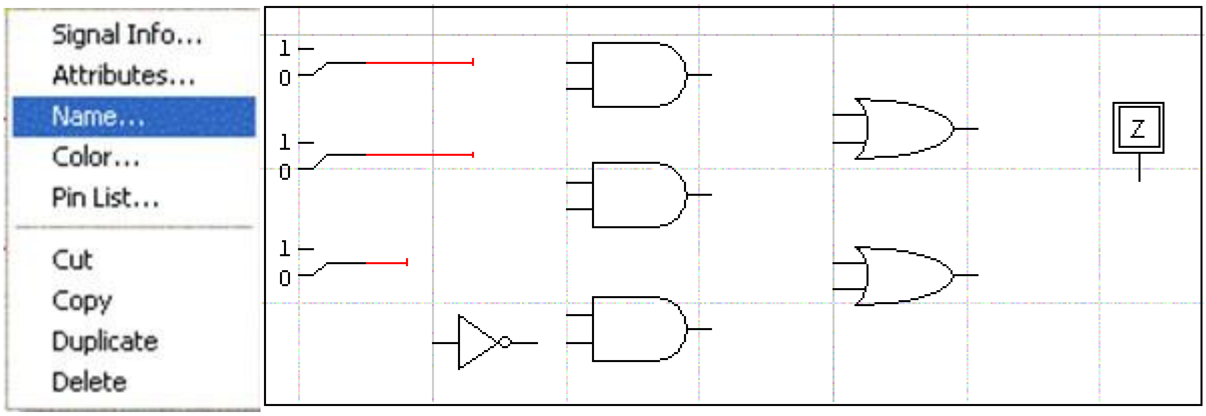
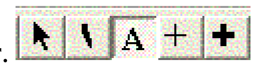


Figure: 14 Wire Connection and labeling

Recall that we are trying to implement Z = AB + BC + AC'. To clarify your design and show its relation to the equation above, we will label the wires A, B, C, and Z.

This process can also be done by clicking on the button **A** on the top tool bar. 

The mouse pointer then looks like a little pencil. Point this pencil at one of the wires (remember only the red parts are wires, if you click on a black part you are labeling a pin) and click on it. A text box appears and you can type a name for the wire. Hit *Esc/Space* to exit the name process.

Connect the devices together with wires:

- Place the pointer on the end of the wire extending from switch A and hold down the left button, then drag the mouse to an input of the first AND gate. You have now connected the switch for input A to the first input of the AND gate.
- Connect the wire from switch B to other input of the first AND gate in a similar manner.
- Click on the first input of the second AND gate.

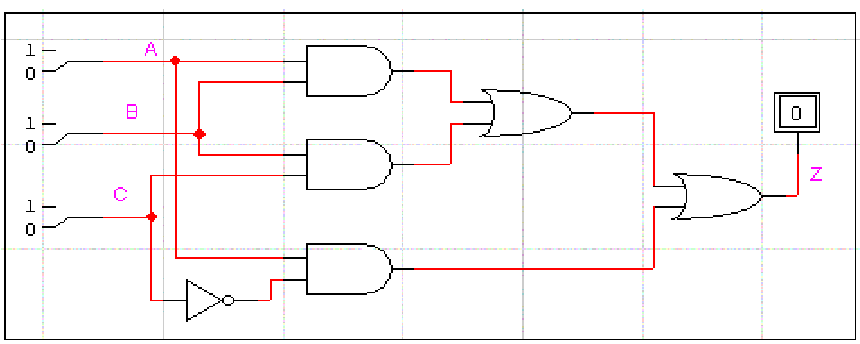
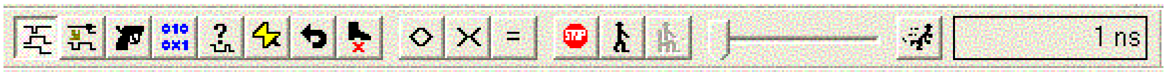


Figure: 15 Connection

- Drag a wire to any point of the wire extending from switch B. A dot will appear on the intersection if the wires were successfully connected.
- Connect the C switch wire to the input of the inverter.
- Connect the output of this inverter to one of the inputs of the last AND gate and connect the other AND gate input to switch A. Your design should now resemble the circuit window in Fig.11

Simulation Controls

Here is the Simulator Toolbar given below:



Position your mouse pointer on each button, you will find the function of each of them. You may try it on your own.
For example,

1. Clicking on the <> or >< button can adjust the range of the time values to suit the display data.
2. Clicking on the Reset button can make the simulation restart at time 0.
3. Clicking on the Run button can start the simulation.

You may have noticed that in the *Timing Window* that there are small lines drawn like underscores for inputs A, B, and C. These are low because the switches are all set to the zero value.

Testing your Circuit

- Click on the handle of switch A to point to the value 1.
- Change the values of A, B, and C and observe the results in the Timing Window.
- Click on the button <> in the Simulator toolbar. This will stretch the time out, making the waveforms easier to see.

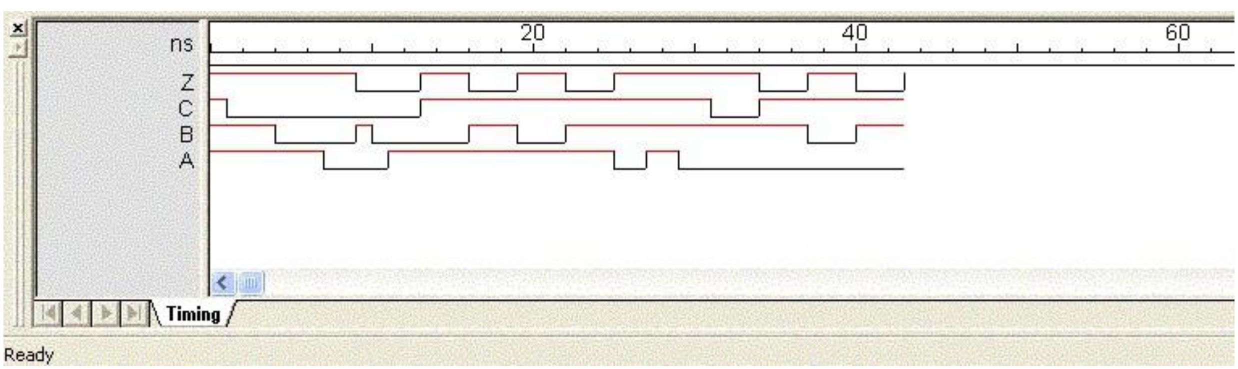


Figure: 16 Timing window

Inserting Text in the Circuit Window:

To insert the text in the circuits window, click on the button **A** in the design toolbar , then follow the following steps.

1. Position your pencil (cursor) at the desired location.
2. Type in the text
3. Change cursor to pointer by clicking on the pointer in the toolbar or press space bar or escape key.
4. Click on the text entered, then right click.
5. Select "Text Style" and then select the preferred options and then, press ok.

Saving and printing your work:

To save or print your work, click the File menu and then select the proper option. The following picture illustrates the idea.

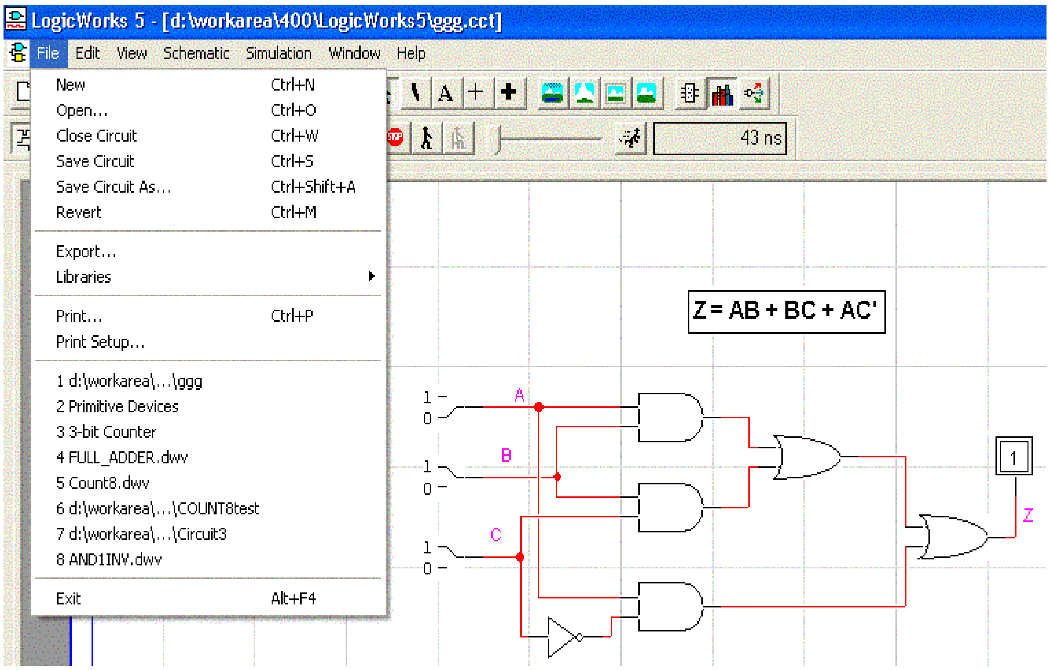


Figure. 17 Print and save option

Basic Logic Gates:

Digital circuits are the electronic circuits that manipulate binary information. Logic gates are the basic building blocks in constructing digital circuits. Logic gate has one output and one or more inputs. Each logic gate performs a specific logical operation. The interconnection between inputs and outputs of gates form a digital circuit. Any digital circuit can implement using three basic logical operations called AND, OR, and NOT. That is why AND, OR, and NOT gates are referred as basic logic gates. AND OR logic functions exhibit the phenomenon of dominance. In both cases, there is an input value that will force the output of the gate to a known value regardless of the state of other inputs. This value is known as the dominant value of the gate. The dominant value of an AND gate is zero, while the dominant value of an OR gate is one. The **Output** of logic gate also provides two nominal values of voltage only e.g. 0V and 5V representing logic 0 and logic 1 respectively. There is always a time delay between an input being applied and the output responding

In this experiment, we will use 74LS08, 74LS32, and 74LS04 ICs for the implementation of AND, OR, and NOT logical operations.

And Gate IC And Its Functionality:

74LS08 IC contains four 2-input AND gates. AND gate is an electronic circuit that gives a high output only when all of its inputs are high & if there is a single input which is low its output is low. So basically, AND gate works on the principle of multiplication.

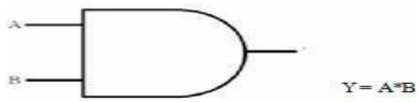


Figure 18: AND Gate Symbol

Function Table:

Inputs		Output
A	**B**	**Y**
L	L	
L	H	
H	L	
H	H	

Table 1: AND Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS08 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the ICs pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
4. For output indication use any one of the LEDs
5. Supply the VCC = +5V and GND as indicated in the diagram
6. Test at least two gates of the IC being used
7. Test the output for all possible combination of inputs and record your results in above Truth Table

Connection Diagram:

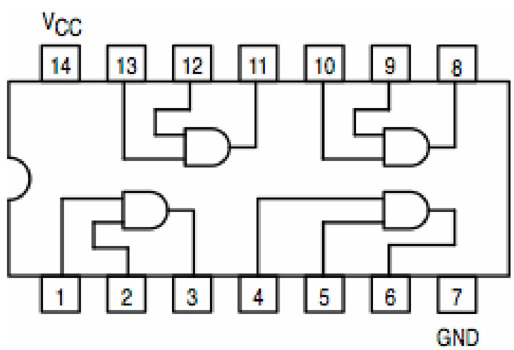


Figure 19: AND Gate IC Configuration

OR Gate IC And Its Functionality:

74LS32 IC contains four 2-input OR gates. OR gate is an electronic circuit that gives a high output if one or more of its inputs are high & gives low output when all of its inputs are low. So basically, OR gate works on the principle of addition.

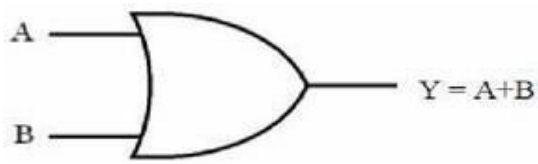


Figure 20: OR Gate Symbol

Function Table:

Inputs		Output
A	**B**	**Y**
L	L	
L	H	
H	L	
H	H	

Table 2: OR Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS32 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the ICs pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
3. For output indication use any one of the LEDs
4. Supply the VCC = +5V and GND as indicated in the diagram
5. Test at least two gates of the IC being used
6. Test the output for all possible combination of inputs and record your results in above Truth Table

Connection Diagram:

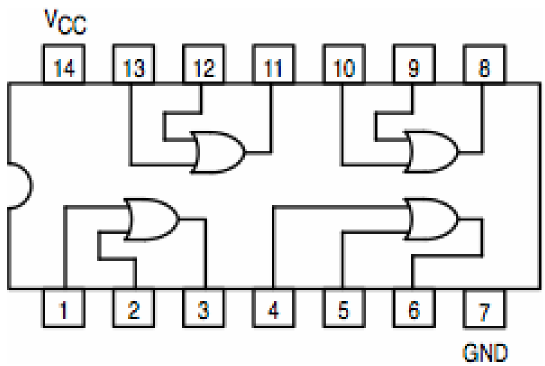


Figure 21: OR Gate IC Configuration

NOT Gate IC And Its Functionality:

74LS04 IC contains Six 1-input NOT gates. NOT gate is an electronic circuit that is used to invert a digital logic, hence called as an **inverter**. It always has exactly a single input and a single output. Whatever logical state is applied to the input, the opposite state will appear at the output.

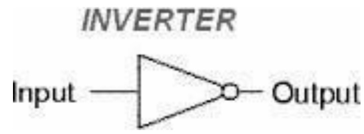


Figure 22: NOT Gate Symbol

Function Table:

Input	Output
A	**Y**
L	
H	

Table 3: NOT Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS04 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the ICs pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
3. For output indication use any one of the LEDs
4. Supply the VCC = +5V and GND as indicated in the diagram
5. Test at least two gates of the IC being used
6. Test the output for all possible combination of inputs and record your results in following Truth Table

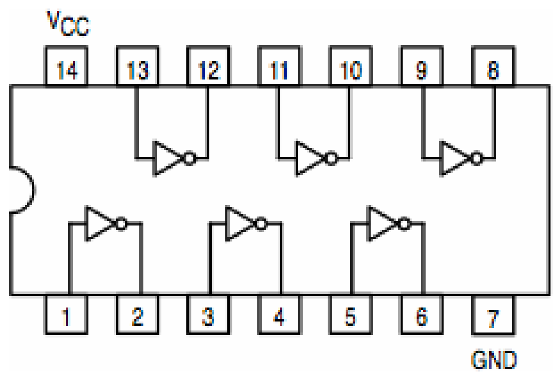


Figure 23: NOT Gate IC Configuration

Testing of ICs:

Before starting implementation of a specific logic circuit, all basic gate ICs should be tested in order to make sure that the ICs are working properly. Using the function table (truth table) for each gate, in a particular IC, apply all input combinations one by one and check its output logic level on LED.

Troubleshooting:

After testing all required number and type of ICs we need to implement a digital circuit, we start implementing the circuit on logic trainer. Once we complete the implementation, we need to test the output of the circuit to make sure that whether the circuit is working accurately or not. Using the truth table that represents the functionality of the logic circuit, we apply all input combinations one by one and check its output logic level on LED.

Lab Task:

Lab Task#1:

Implement the following logic circuit on logic trainer, and write **truth table** in the space provided below:

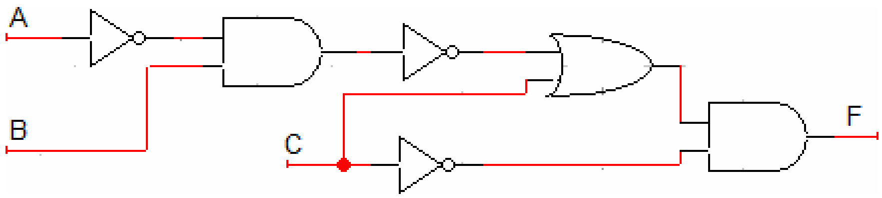


Figure 24: Combinational Circuit

Lab Task#2

Draw the logic circuit represented by each of the following expression:

1. A+B+C+D
2. ABCD
3. A+BC
4. ABC+D

Lab Task#3:

Implement the following logic circuit on logic trainer, and write truth table in the space provided below:

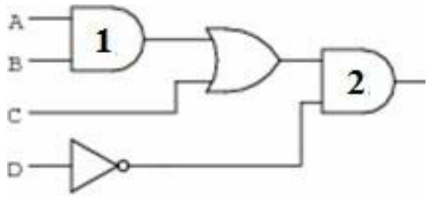


Figure 25: Combinational Circuit