# ISAT 252 - Analytical Methods IV
## Lab1: Digital Logic and Computers

**Objectives:**

- Gain a basic understanding of digital logic
- Experiment with digital logic programming using the MMLogic MultiMedia logic design system from Softronics, Inc .

**Instructions:**

- Download the MMLogic software module from blackboard and save it to the desktop.
- You should also download the help file to gain a better understanding of the digital components and their functionality as used in this lab.

**Deliverables**

- *Parts B, C and D. Include the printed copies of the worksheets for Parts B, C and D and the electronic submission of your circuit file for Parts C and D.*

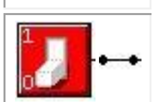
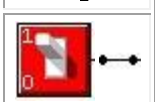
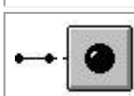
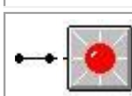
**Introduction**

All digital computers are built with the basic components shown in this lab. All other circuits, including the **Xor Gate** shown below, can be constructed with **Inverter Gates**, **Or Gates,** and **And Gates**. What makes computers so complicated is that they are composed of millions of these basic gates, interconnected in very intricate ways. When you have finished studying these concepts, you will have a basic understanding of digital logic and the circuits used to implement it.

**Definitions**

As you study this lab, here are some things to consider:

- A **State** is a *way of being.*
- An **Input** is a pin, connection, or electrical signal that goes **INTO** a device that *affects* its **current state** in some way.
- An **Output** is a pin, connection, or electrical signal that comes **OUT** of a device and *reflects* its **current state**.
- For **Binary Digital Computers**:

| **Binary States** | | |
|---|---|---|
| **Inputs** and **Outputs** | **Off** | **On** |
| | **0** | **1** |
| | **False** | **True** |
| **Switches** on this page | **Down** | **Up** |
| |  |  |
| **LED's** on this page | **Dark** | **Lit** |
| |  |  |

- In general, **Outputs** are connected to **Inputs.**
- The **Switches** are **Input Devices.** On this page, all switches have **ONE OUTPUT.** which can be connected to **ONE OR MANY INPUTS**
- The **LED's** are **Output Devices.** On this page, all LED's have **ONE INPUT** which can be connected to just **ONE OUTPUT.**
- All **Logic Gates** on this page have either **ONE** or **TWO** or **THREE INPUTS,** with only **ONE OUTPUT.**
- A **Truth Table** is a list of all the *possible* ***input*** states of a device along with the *resulting* ***output*** states.

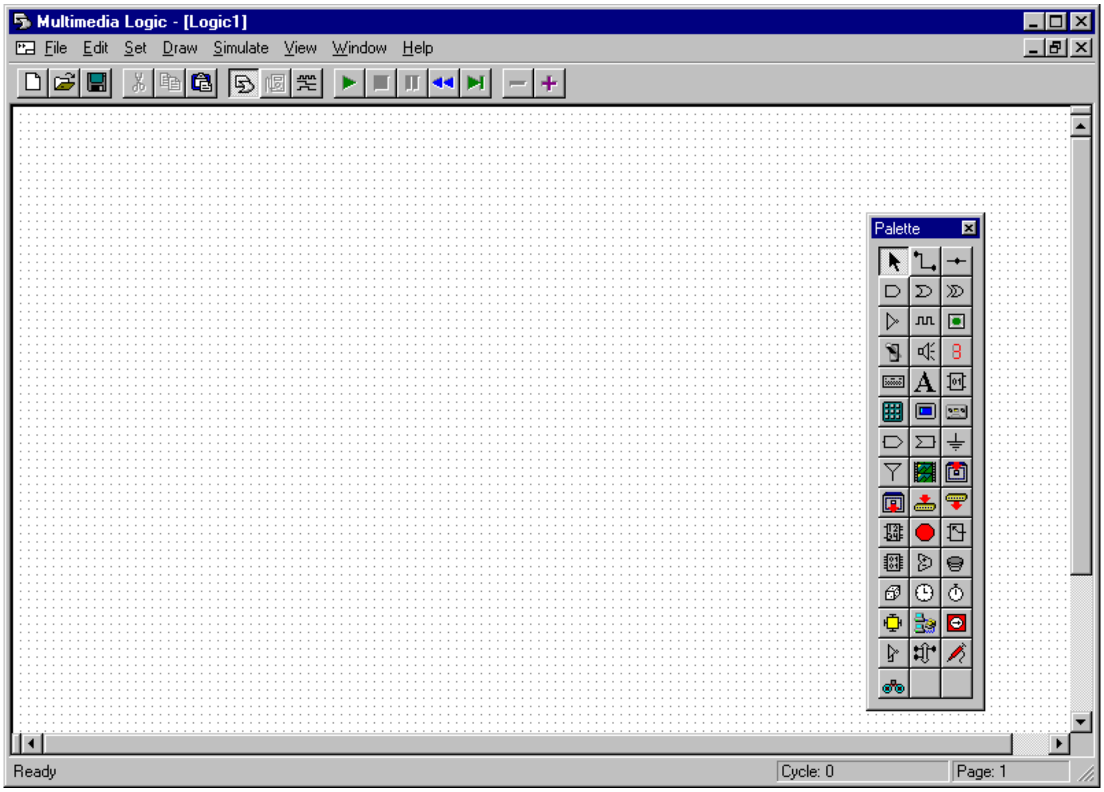
As you proceed with this lab, you should experiment with the switches on this page. Clicking them will change their state -- that is, from a **"Down / 0"** to a **"Up / 1"**, and vise versa. Then watch what happens to the LED's in the same box.

## Part A – Tutorial Walkthrough: Using the Multimedia Logic Environment

## Starting the program

- Double click on the Multi Media Logic  icon to start the program.
- A window with a message about MSW may appear;  if it does, just click the  OK button and it will go away.
- A window with a tip about using the program may appear;  if it does, just click the Close  button and it will go away.
- Click on the Full screen button of the Multimedia Logic window (marked □ at the top right of the window).
- Inside the Multimedia Logic window there is another window called Logic 1. Click on its Full screen button.

  The screen should now look like this:




- If the Palette is not visible, click on the first button in the third group of three icons at the top of the window.
- Use the mouse to move the palette to the right-hand side of the window by pointing at its title and dragging it across (push the left-hand mouse button and hold it down while moving the mouse).

## Drawing a circuit

- Click on the symbol for an **AND** gate, which is the first icon in the second row of the palette; the cursor will change to a plus sign (+).
- Move the mouse to the middle of the window and click the left-hand button; a large **AND** gate will appear. Be careful not to click the button more than once or several gates will appear.
- Click on the arrow symbol, which is the first icon in the top row of the palette.
- Point at the **AND** gate and drag it round the screen.
- Press the grey DELETE key and the **AND** gate will disappear.
- Move the mouse to the list of menus at the top of the window and click on Edit; a menu will drop down.
- Click on the word Undo and the **AND** gate will reappear.

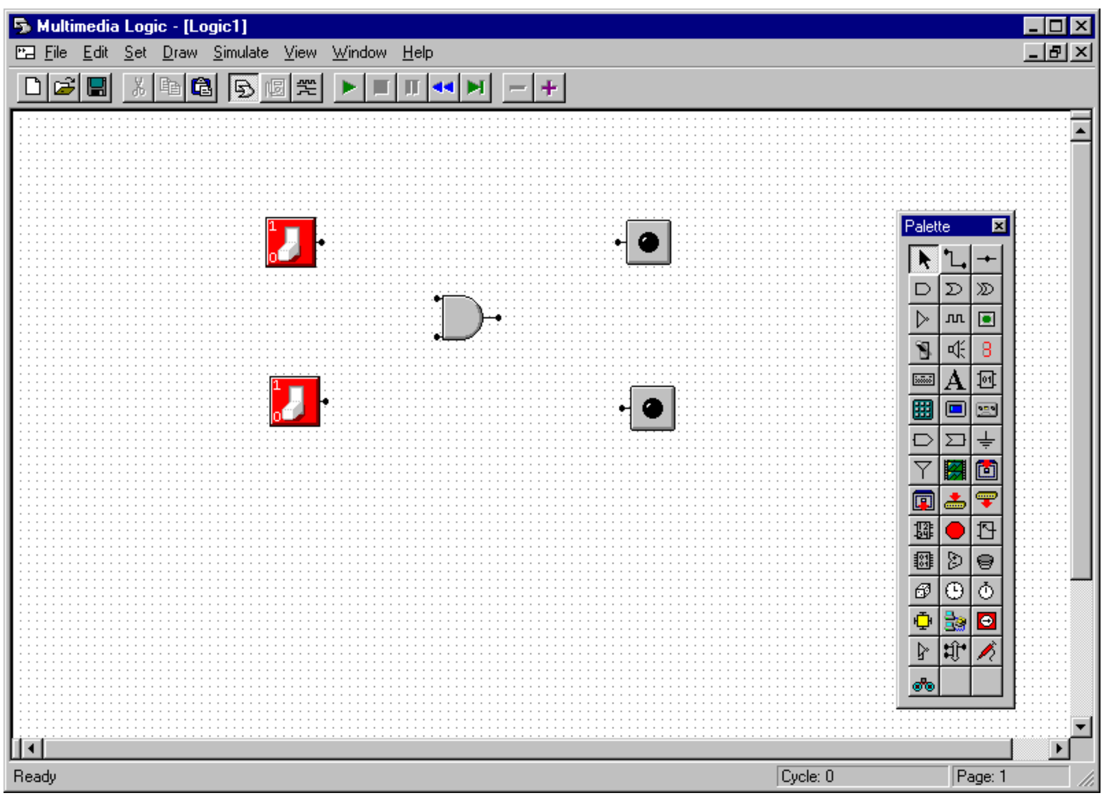
Remember these tricks in case you make any mistakes. If you put part of the circuit in the wrong place, you can select it with the arrow and then drag it to the right place. If you accidentally draw something that is not wanted, you can select it and delete it. If you make any mistake, you can always undo it.

A surrounding box highlights the currently selected part of the circuit. You can click anywhere on the background to deselect everything (which may help you see details in the circuit).

## Adding more components

- Click on the symbol for a switch, which is the first icon in the fourth row of the palette.
- Move the mouse to the left-hand side of the window and click the left-hand button twice to make two copies of the switch, one just above the level of the AND gate and one just below.
- Click on the symbol for a light-emitting diode (LED), which is the third icon in the third row of the palette.
- Move the mouse to the right-hand side of the window and click the left-hand button twice to make copies of the LED, roughly level with the two switches.

The screen should now look like this:

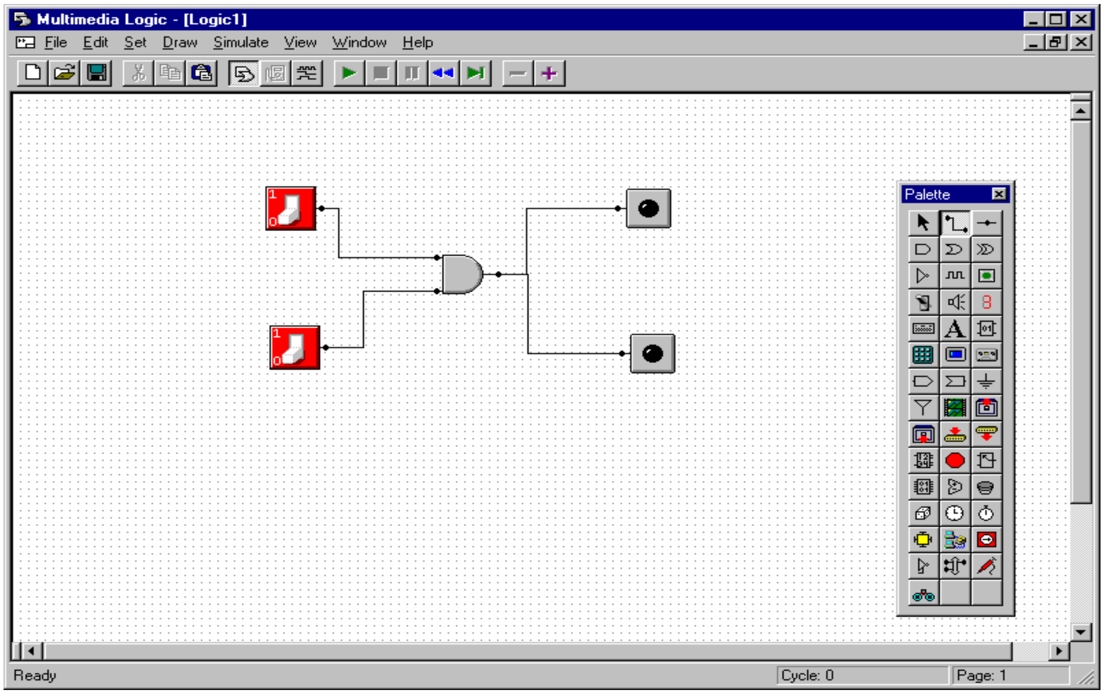


Notice the little black dots on the stalks to the left and right of the components; these are connection points.  Each switch has one connection point on its right-hand side, each LED has one on its left-hand side and the AND  gate has two on its left-hand-side and one on its right-hand side.

## Wiring the components together

- Click on the symbol for a wire, which is the second icon in the top row of the palette; the cursor will change shape to a plus sign.
- Point carefully at the connection point on the right-hand side of the top, left switch.
- Press the left-hand button and drag a wire across to the upper connection point on the left-hand side of the AND gate, then release the button.
- If you miss the connection point on the switch, a little window will pop up saying “Error Nothing to connect to”.  Just click on OK and it will go away so you can try again.
- If you miss the connection point on the AND gate, the wire will disappear. Just try to draw it again.
- Join the lower switch to the lower connection point on the left-hand side of the AND  gate similarly.
- Join the connection point on the right-hand side of the AND gate to the upper LED similarly.
- Now join the connection point on the right-hand side of the AND  gate to the lower LED.  Notice that you can have two wires connected to a single connection point.

The screen should now look like this:




If there are any problems, just correct them by deleting and replacing components or wires. When selecting a wire, you have to point at its end; if you point to the right-hand connection point on the AND gate, you will select both of the wires.

Try dragging the components and notice how the wires stay connected.

## Testing a circuit

- Click on the green triangle which is the first icon in the fourth group at the top of the screen.
- A window may pop up inviting you to save the logic. Just click on the Cancel button and it will go away.

    *The palette will disappear and the box labelled Cycle at the bottom right-hand side of the window will start counting up.*

- Click on the two switches to turn them on and observe that the two LEDs come on. (They are American switches, so up is on and down is off. ☺)
- Turn them on and off a few times to convince yourself that the AND gate is working correctly. Write the results here:

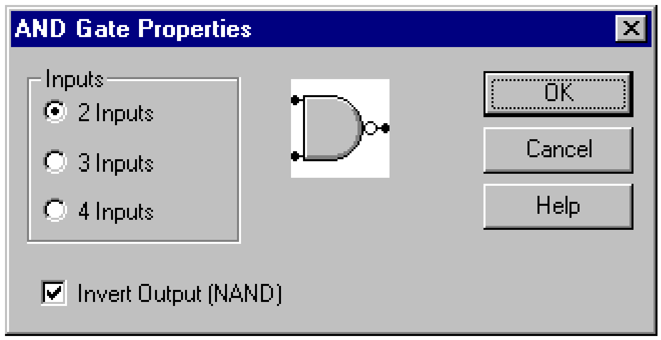
| Upper switch | Lower switch | LEDs |
|---|---|---|
| Off | Off | |
| Off | On | |
| On | Off | |
| On | On | |

- Explain the results (Don't do this here. Space is provided in part B)

- Click on the red square next to the green triangle in the fourth group of icons at the top of the window and the simulator will stop. The palette should re-appear.

## Changing a circuit

- Move the mouse to the AND gate and click the right-hand button; a menu will pop up. Click on Properties… and a window labelled AND Gate Properties will pop up:




- Click in the box at the bottom left-hand side of the window so that a tick appears just before the text Invert Output (NAND).
- Click OK.

*Notice how the AND gate now has a little circle on its right-hand side, showing that its output has been inverted; it is now a NAND gate. (You may need to deselect the gate before this becomes clear.)*

- Simulate (RUN) the new circuit and see the difference in its behaviour. Write the results here:

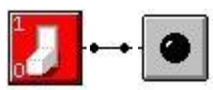
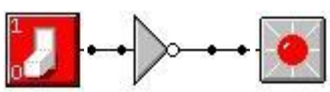
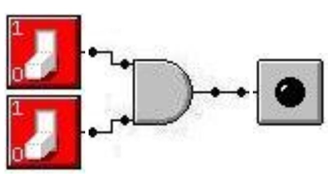
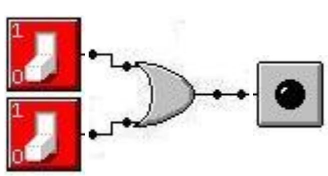
| Upper switch | Lower switch | LEDs |
|---|---|---|
| Off | Off | |
| Off | On | |
| On | Off | |
| On | On | |

- Click on the red square next to the green triangle in the fourth group of icons at the top of the window to stop the simulator when you have finished.

**Name:** Kadar Anwar **Section:** 003

## Part B – Building and Testing Basic Logic Gates Using the Multimedia Logic Design and Simulation Environment. Completing Truth Tables.

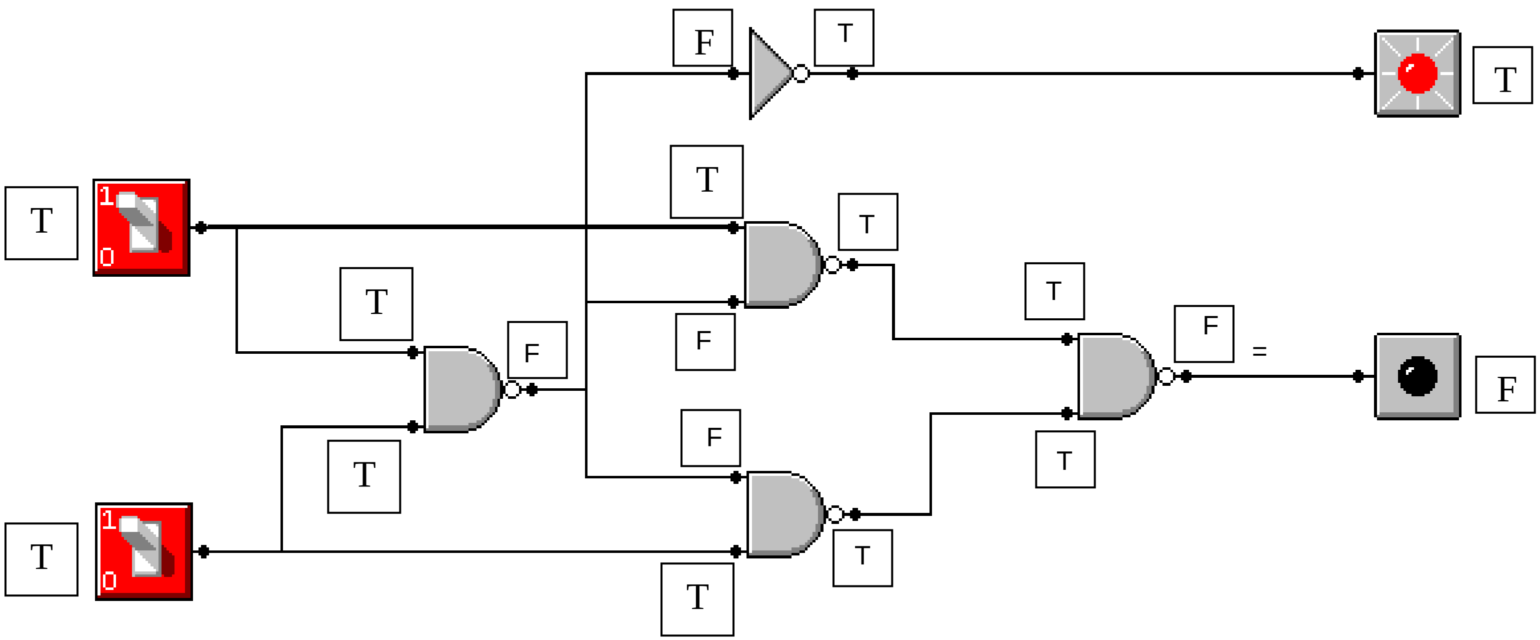
The circuits below are the most basic found in a **Binary Digital Computer**. Build all of the circuits and fill in the corresponding Truth Table. Also, give a brief description and explanation of what you have observed as you toggle through all possible switch combinations. *The first one has been done for you.*

<table>
<tr><th colspan="4">Basic Circuits</th></tr>
<tr><th>Name</th><th>Schematic Symbol</th><th>Truth Table</th><th>Description</th></tr>
<tr><td>Switch and LED</td><td></td><td><table><tr><th>Switch</th><th>LED</th></tr><tr><td>Down / 0</td><td>Dark / 0</td></tr><tr><td>Up / 1</td><td>Lit / 1</td></tr></table></td><td>The most basic of all circuits. The switch on the left will turn the LED (Light Emitting Diode) on the right on. Click it and see.</td></tr>
<tr><td>Inverter Gate</td><td></td><td><table><tr><th>Switch</th><th>LED</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td><td>The inverter gate causes the result to be opposite of what is expected. That is, if the switch is set to 0 the LED will be ON; if the switch is set to 1 the LED will be OFF. The Inverter acts as a "NOT" logical operator.</td></tr>
<tr><td>And Gate</td><td></td><td><table><tr><th colspan="2">Switches</th><th>LED</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>The AND gate requires both switches to be ON in order for the LED to turn on. Any other combination will not cause the LED to remain OFF.</td></tr>
<tr><td>Or Gate</td><td></td><td><table><tr><th colspan="2">Switches</th><th>LED</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>The OR gate requires only one switch to be ON to power on the LED. The LED will remain ON as long as at least one, or both, switches are on. It will only turn off if both switches are turned OFF.</td></tr>
</table>

## Part C – Other Circuits: Applying What You Have Learned.

As an application of what you have learned, you will design and analyze an important circuit of your own. This circuit is central to all digital electronic devices (cell phones, computers, PDAs, etc)

1) In the diagram below, fill in the empty boxes with a True (T) or False (F). Look at the inputs to each gate and use the truth tables to determine the outputs. Note that four of the gates are NAND gates, and the triangular gate is an Inverter (i.e. a NOT gate). Your final input into the top LED will be True (since it's ON), and the input to the lower LED will be False (since it's OFF).



2) Create the above circuit in MMLogic and change the input switches. Notice how the output changes. Now, think of True as the binary digit 1, and think of False as the binary digit 0. Compare the input and output to the binary addition examples below. Note your observations and fill in the truth table for this circuit. SAVE YOUR COMPLETED CIRCUIT AND ELECTRONICALLY SUBMIT IT TO YOUR FTP FOLDER.

Binary Addition Examples: 0 + 0 = 00 0 + 1 = 01 1 + 0 = 01 1 + 1 = 10

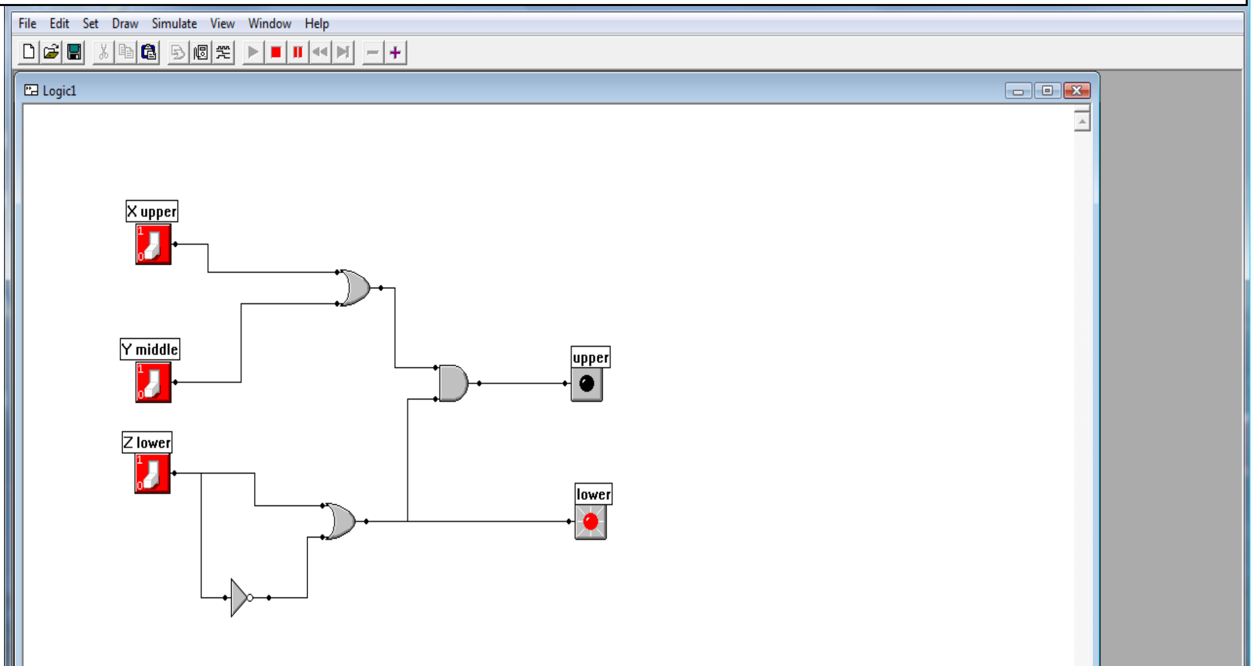
- Turn the switches on and off a few times to investigate the circuit's behaviour. Fill in the results here:

TRUTH TABLE

| Left upper switch | Left lower switch | Upper LED | Lower LED |
|---|---|---|---|
| Off | Off | | |
| Off | On | | |
| On | Off | | |
| On | On | | |

- In your own words, explain the results.

**Part D: Building and Testing Logic Gates- Truth Tables and Boolean Logic Expression**
Build and save the circuit found below as Part D. Upload the file to your ftp folder.
Hand in hard copy of work at the beginning of class




2. Complete the Truth Table for the circuit.

**TRUTH TABLE**

| X upper switch | Y middle switch | Z lower switch | Upper LED | Lower LED | |
|---|---|---|---|---|---|
| Off | Off | Off | | | |
| Off | Off | On | | | |
| Off | On | Off | | | |
| Off | On | On | | | |
| On | Off | Off | | | |
| On | Off | On | | | |
| On | On | Off | | | |
| On | On | On | | | |

3. Write the corresponding **expressions (one for the upper LED, one for the lower LED)** using the logical connectives AND, OR, and NOT.