DAILY ASSESSMENT

Date:	**28/05/2020**	**Name:**	**CHANDANA.R**
Course:	Logic design	**USN:**	**4AL16EC017**
Topic:	Boolean algebra, MUX to LOGIC gate conversion and 7 segment display	**Semester & Section:**	**8(A)**
Github Repository:	**Chandana-shaiva**		

FORENOON SESSION DETAILS

REPORT:

- Digital Circuits Boolean algebra

Boolean Algebra is an algebra, which deals with binary numbers & binary variables. Hence, it is also called as Binary Algebra or logical Algebra. A mathematician, named George Boole had developed this algebra in 1854. The variables used in this algebra are also called as Boolean variables.

The range of voltages corresponding to Logic 'High' is represented with '1' and the range of voltages corresponding to logic 'Low' is represented with '0'.

Consider the binary numbers 0 and 1, Boolean variable x and its complement x′. Either the Boolean variable or complement of it is known as **literal**. The four possible **logical OR** operations among these literals and binary numbers are shown below.

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + x' = 1$$

Similarly, the four possible **logical AND** operations among those literals and binary numbers are shown below.

$$x.1 = x$$

$$x.0 = 0$$

$$x.x = x$$

$$x.x' = 0$$

These are the simple Boolean postulates. We can verify these postulates easily, by substituting the Boolean variable with '0' or '1'.

Boolean Laws

There are six types of Boolean Laws.

Commutative law

Any binary operation which satisfies the following expression is referred to as commutative operation.

$$\text{(i) } A.B = B.A \qquad \text{(ii) } A + B = B + A$$

Commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit.

Associative law

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

$$\text{(i) } (A.B).C = A.(B.C) \qquad \text{(ii) } (A + B) + C = A + (B + C)$$

Distributive law

Distributive law states the following condition.

$$A.(B + C) = A.B + A.C$$

AND law

These laws use the AND operation. Therefore they are called as **AND** laws.

$$\text{(i) } A.0 = 0 \qquad \text{(ii) } A.1 = A$$

$$\text{(iii) } A.A = A \qquad \text{(iv) } A.\overline{A} = 0$$

OR law

These laws use the OR operation. Therefore they are called as **OR** laws.

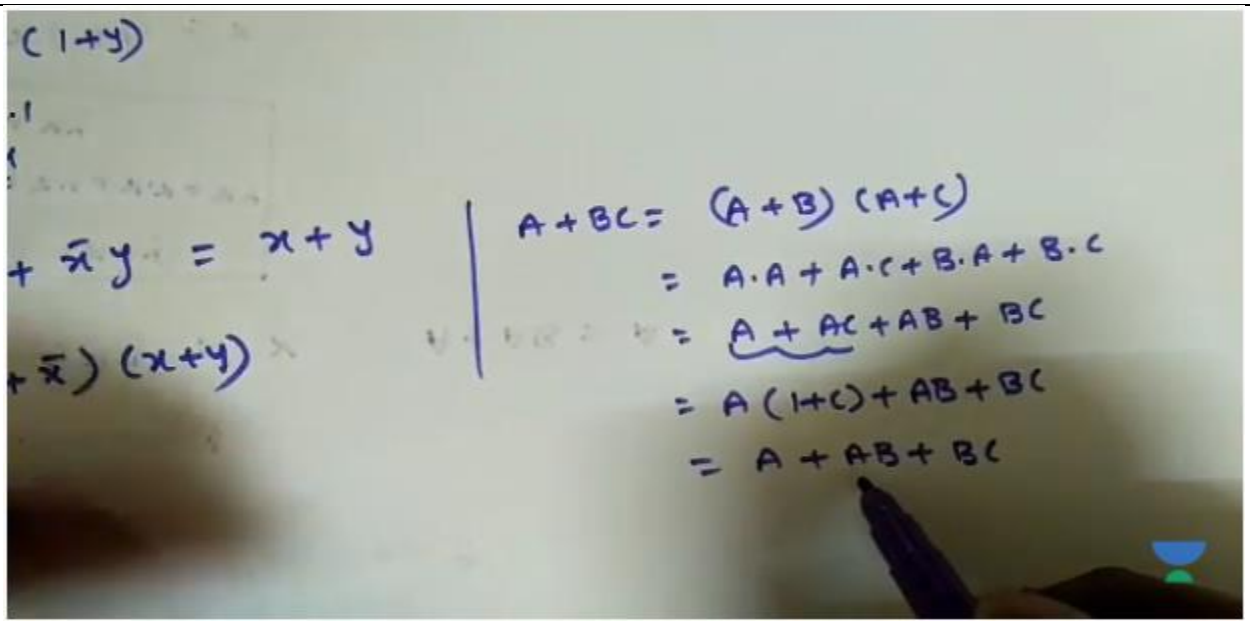
$$\text{(i) } A + 0 = A \qquad \text{(ii) } A + 1 = 1$$

$$\text{(iii) } A + A = A \qquad \text{(iv) } A + \overline{A} = 1$$

INVERSION law

This law uses the NOT operation. The inversion law states that double inversion of a variable results in the original variable itself.

$$\overline{\overline{A}} = A$$



- **MUX to LOGIC gates conversion**

A 2^n-input mux has n select lines. It can be used to implement logic functions by implementing LUT (Look-Up Table) for that function. A 2-input mux can implement any 2-input function, a 4-input mux can implement any 3-input, an 8-input mux can implement any 4-input function, and so on. This property of muxes makes FPGAs implement programmable hardware with the help of LUT muxes. In this post, we will be discussing the implementation of 2-input AND, OR, NAND, NOR, XOR and XNOR gates using a 2-input mux.

-input NAND gate using 2:1 mux: below shows the truth table of a 2-input NAND gate. If we observe carefully, OUT equals '1' when A is '0'. Similarly, when A is '1', OUT is B'. So, if we connect SEL pin of mux to A, D0 pin of mux to '1' and D1 to B', then it will act as a NAND gate.

A	B	OUT
0	0	1
0	1	1
1	0	1
1	1	0

OUT = 1 when A = 0

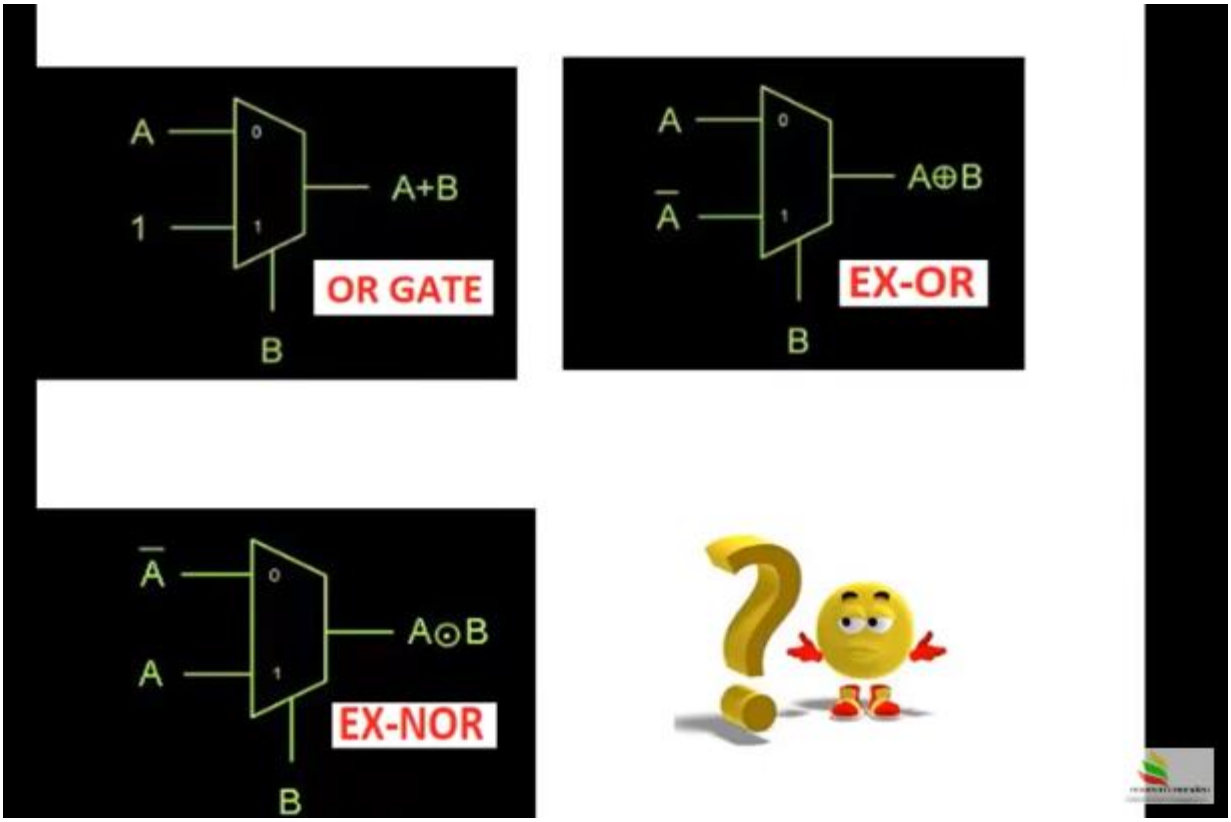
OUT = B' when A = 1

<u>2-input OR gate using 2x1 mux</u>: Figure 5 below shows the truth table for a 2-input OR gate. If we observe carefully, OUT equals B when A is '0'. Similarly, OUT is '1' (or A), when A is '1'. So, we can make a 2:1 mux act like a 2-input OR gate, if we connect D0 pin to B and D1 pin to A, with select connected to A.

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

OUT = B when A = 0

OUT = 1 when A = 1



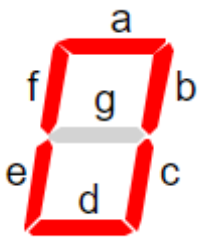
➢ BCD to 7 segment decoder:

A seven-segment display is an electronic display device for displaying decimal numerals. Seven-segment displays are widely used in digital clocks, electronic meters and other electronic devices that display numerical information.

The schematic shows a BCD to 7 Segment Display for one of the digits of a digital clock. The following components are used.

A 7 Segment LED display generally has 8 input connections, one for each LED segment and one that acts as a common terminal. There are 2 types of 7 Segment LED digital display.

- Common Cathode Display – all the cathode connections of the LEDs are connected to ground. A logic '1' applied to the anode terminal of the individual segment illuminates it.
- Common Anode Display – all the anode connections of the LEDs are connected to VCC. A logic '0' applied to the cathode terminal of the individual segment illuminates it.



DAILY ASSIGEMENTS DETAILS:

Date:	**28/05/2020**	**Name:**	**CHANDANA.R**
Course:	Python	**USN:**	**4AL16C017**
Topic:	Application 5: Build a Desktop Database Application	**Semester & Section:**	**8(A)**
Github Repository:	**Chandana-shaiva**		

AFTERNOON SESSION

Session image:

REPORT:

- Python has been the most trending programming language used for object oriented programming.
- With python you can run simple statement over and over again without having to compile a whole program of which it's output functionality.

- Python is an interactive programming which has a diverse range of options for GUI (Graphical User Interface) framework (help developers create GUI applications in an easy and secure manner).

- This article discusses the Best Python framework for building a desktop application and GUI.

➢ PyQT:

- PyQt is a Graphical User Interface widget toolkit. It is one of the most powerful and popular Python interfaces.
- It is a combination of the Qt (owned by Nokia) library and Python programming language which leaves a developer to decide whether to create a program by coding or create visual dialogs using Qt Designer.

➢ Tkinter:

- Tkinter is the most popular programming package for graphical user interface or desktop apps. It is so named because of its simplicity.
- Tkinter is the combination of Tk and Python's standard GUI framework. TKinter comes with an abundance of resources of codes and reference books which is the major merit of choosing it as a package.
- It provides diverse widgets, such as labels, buttons, and text boxes used in a graphical user interface application. The Button control also called widgets are used to display buttons in developed application while the Canvas widget is used to draw shapes (lines, ovals, polygon...) in your application.

➢ KIVY:

- Kivy as an OpenGL ES 2 accelerated framework for the creation of new user interfaces empowers you with the ease to write your code once and have it run on different platforms or Operating Systems.

➢ WxPython:

- WxPython is a python extension module. It is also an open source wrapper for cross-platform graphical user interface library Widget**.**

➢ PyGUI:

- PyGUI is the simplest and most lightweight of all GUIs because it is purely in sync with Python programming language.

- It is a cross-platform graphical application framework for Windows, Mac OS, and Unix.
- A PyGUI developer inserts very less code between the GUI platform and Python application which in turn displays the natural GUI of the platform.