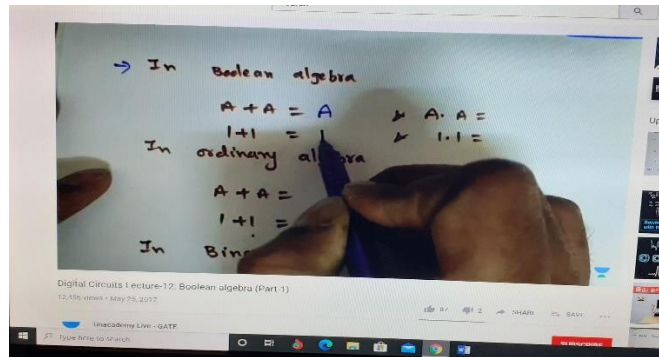| Date:	28-5-2020	Name:	Kavyashree m
Course:	Logic design	USN:	4al15ec036
Topic:	Boolean equations for digital circuits.Combinational circuits: Conversion of MUX and Decoders to logic gates. design of 7 segment decoder with common anode display	Semester & Section:	8th A
Github Repository:	Kavya		

FORENOON SESSION DETAILS

Image of session

Fig 1 : Boolean equations for digital circuits
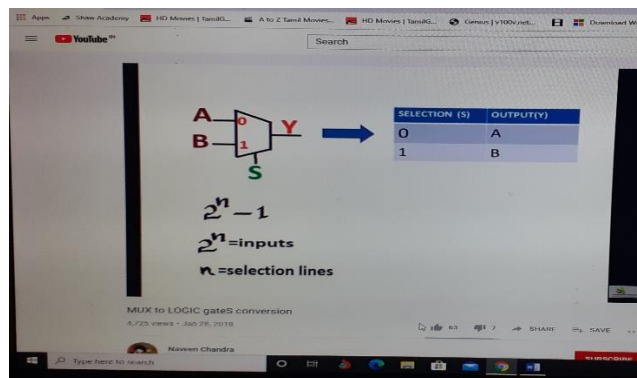
Fig 2 : Conversion of MUX and Decoders to logic gates

BCD to 7-segment decoder
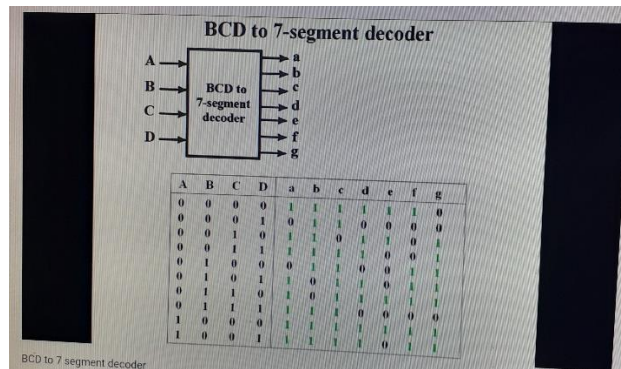
Fig 3: design of 7 segment decoder with common anode display

Boolean equations for digital circuits

Boolean algebra, a logic algebra, allows the rules used in the algebra of numbers to be applied to logic. It formalizes the rules of logic. Boolean algebra is used to simplify Boolean expressions which represent combinational logic circuits. It reduces the original expression to an equivalent expression that has fewer terms which means that less logic gates are needed to implement the combinational logic circuit.

Laws of Boolean Algebra

Boolean Algebra Laws are used to simplify boolean expressions.

Basic Boolean Laws

1. Idempotent Law

   - A * A = A

   - A + A = A

2. Associative Law

   - (A * B) * C = A * (B * C)

- ○ (A + B) + C = A + (B + C)

3. Commutative Law

    - ○ A * B = B * A

    - ○ A + B = B + A

4. Distributive Law

    - ○ A * (B + C) = A * B + A * C

    - ○ A + (B * C) = (A + B) * (A + C)

5. Identity Law

    - ○ A * 0 = 0   A * 1 = A

    - ○ A + 1 = 1   A + 0 = A

6. Complement Law

    - ○ A * ~A = 0

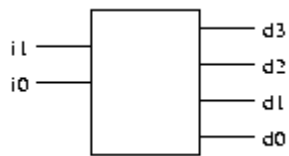   - ○ A + ~A = 1

7. Involution Law

    - ○ ~(~A) = A

8. DeMorgan's Law

    - ○ ~(A * B) = ~A + ~B

    - ○ ~(A + B) = ~A * ~B

Conversion of MUX and Decoders to logic gates

Decoders

A decoder is a circuit which has n inputs and 2^n outputs, and outputs 1 on the wire corresponding to the binary number represented by the inputs. For example, a 2-4 decoder might be drawn like this:
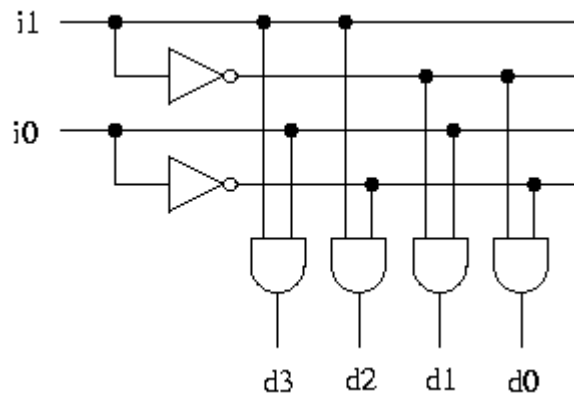

and its truth table (again, really four truth tables, one for each output) is:

i_1	i_0	d_3	d_2	d_1	d_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

The Decoder Circuit

The following circuit generates all four minterms from two inputs, and implements the 2-4 decoder.

d3   d2   d1   d0

Larger decoders can be implemented in the same way. Here is a 3-8 decoder.

d7   d6   d5   d4   d3   d2   d1   d0
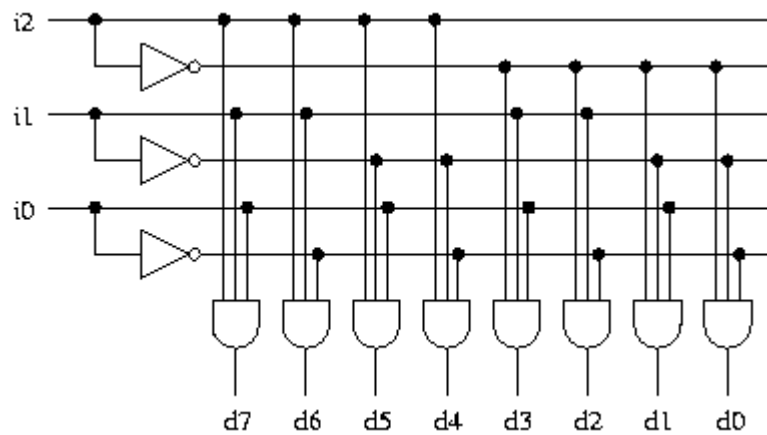
Multiplexers

A multiplexer is a device which allows one of a number of inputs to be routed to a single output. Here is a 4-1 multiplexer.
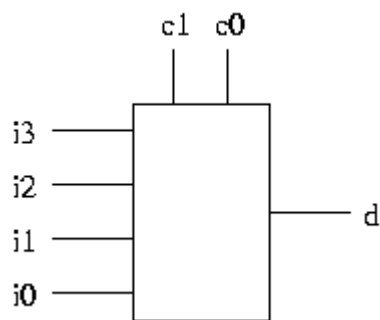
The control inputs c_0 and c_1 represent a 2-bit binary number, which determines which of the inputs i_0◆i_3 is connected to the output d.

If $c_1 c_0$ represents the number n in binary, then the value of the output d is the value of input i_n.

Multiplexers are useful in many situations. For example, in a CPU, data being written to memory might come from one of a number of sources - from a register, from the result of a calculation, etc - so a multiplexer would be used to select data from the appropriate source.

Another application is where we want to be able to choose one of several operations to carry out on some data - all the operations can be calculated, and a multiplexer can be used to select the desired result (more on this later).

Implementing Multiplexers

The implementation of a multiplexer is straightforward, and uses a decoder. Here is a 4-1 multiplexer.
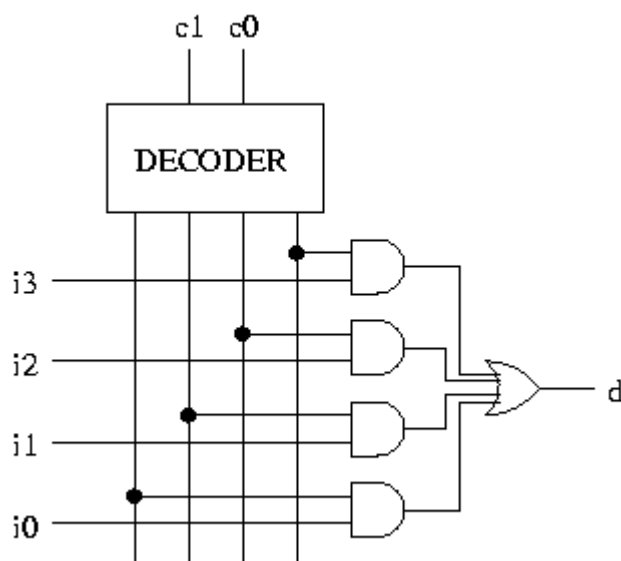
All the outputs of the decoder are 0, apart from one. The inputs $c_1 c_0$ determine which of the outputs is non-zero.

All but one of the AND gates have 0 on one input and therefore output 0. The remaining AND gate has 1 on one input and i_n on the other input. The output of this AND gate is the value of i_n.

The OR gate has 0 on all of its inputs apart from one, and has the value of i_n on the remaining input. The output of the OR gate is therefore the value of i_n.

Larger multiplexers can be implemented in the same way.

Design of 7 segment decoder with common anode display

Step 1: The first step of the design involves analysis of the common cathode 7-segment display. A 7-segment display consists of an arrangement of LEDs in an 'H' form. A truth table is constructed with the combination of inputs for each decimal number. For example, decimal number 1 would command a combination of b and c (refer the diagram given below).

Common Cathode
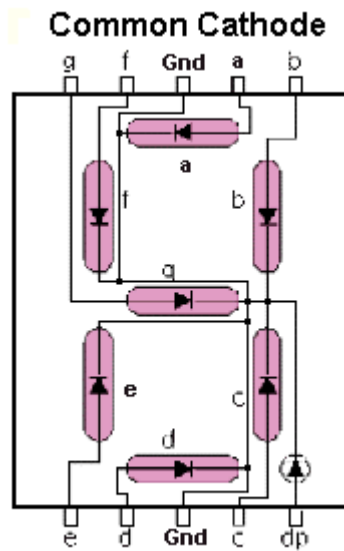
7 Segment LED

Step 2: The second step involves constructing the truth table listing the 7 display input signals, decimal number and corresponding 4 digit binary numbers

Step 3: The third step involves constructing the Karnough's map for each output term and then simplifying them to obtain a logic combination of inputs for each output.

Step 4: The final step involves drawing a combinational logic circuit for each output signal. Once the task was accomplished, a combinational logic circuit can be drawn using 4 inputs (A,B,C,D)and a 7- segment display (a,b,c,d,e,f,g) as output.

AFTERNOON SESSION DETAILS

Date:	28-5-2020	Name:	Kavyashree m
Course:	Python programming	USN:	4al15ec036
Topic:	Build a desktop database application	Semester & Section:	8th A
Github Repository:	Kavya		

Image of session
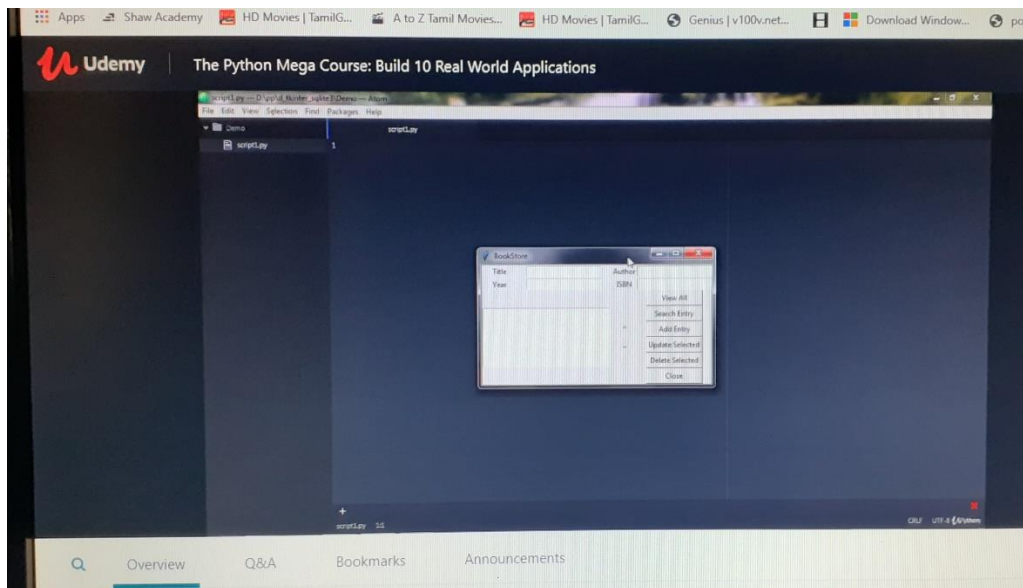
# Build a desktop database application

Python is an interactive programming which has a diverse range of options for GUI framework. This article discusses the Best Python framework for building a desktop application and GUI.

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 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
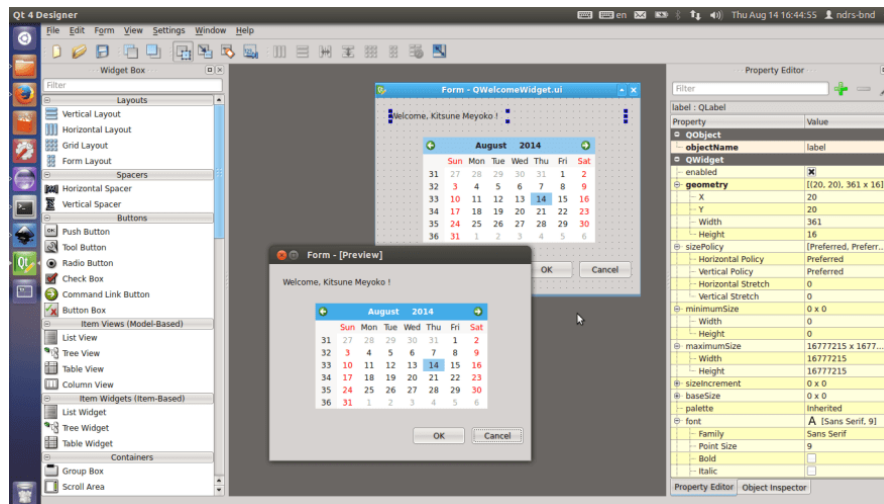
PyQt is a free Python bindings software open-source widget-toolkit Qt, implemented for cross-platform application development framework. In the free version, certain features may not be available but if your application is open source then you can use it under a free license.

User interface design

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

- Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- wxPython − This is an open-source Python interface for wxWindows.
- JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine.

Frontend interface

Front-end Web Development refers to building web interfaces, specifically the parts of the website that the user will interact with. When you're browsing the web, everything

you see, from images and headings to sliders and buttons is made using HTML, CSS and JavaScript, the main components to any website.

Backend interface

Back-end Development refers to the parts of the website that a user doesn't see or directly interact with. The back end handles application logic, algorithms, database interaction and the processing of user requests.

Connecting Frontend and Backend

There are many different ways how to structure a web application. The frontend can take different forms, and it can be daunting to understand how to connect the two.

The backend and frontend both work together to serve a single goal. It's pretty helpful to keep it in mind at all times. They are made, so a user can access them. In detail this interaction can look like this:

- The user points their browser to one of your website's urls
    - This makes the browser send out one or more requests to your server
    - The browser waits to receive responses from your server
    - Every response is used to render a part of the site
- The user waits for the browser to render the page
- The user sees a useful and usable page
- The user interacts with the page
    - Causing more requests to be sent out, to get more data and display new information, and so on.

Fixing the Bug

is a mature programming language which has established a reputation for stability Python. In order to maintain this reputation, the developers would like to know of any deficiencies you find in Python. It can be sometimes faster to fix bugs yourself and

contribute patches to Python as it streamlines the process and involves less people. Learn how to contribute.

Documentation bugs : If you find a bug in this documentation or would like to propose an improvement, please submit a bug report on the tracker. If you have a suggestion on how to fix it, include that as well.

Bug reports for Python itself should be submitted via the Python Bug Tracker. The bug tracker offers a Web form which allows pertinent information to be entered and submitted to the developers.The first step in filing a report is to determine whether the problem has already been reported. The advantage in doing so, aside from saving the developers time, is that you learn what has been done to fix it; it may be that the problem has already been fixed for the next release, or additional information is needed (in which case you are welcome to provide it if you can!). To do this, search the bug database using the search box on the top of the page.

If the problem you're reporting is not already in the bug tracker, go back to the Python Bug Tracker and log in. If you don't already have a tracker account, select the "Register" link or, if you use OpenID, one of the OpenID provider logos in the sidebar. It is not possible to submit a bug report anonymously.Being now logged in, you can submit a bug. Select the "Create New" link in the sidebar to open the bug reporting form.

The submission form has a number of fields. For the "Title" field, enter a *very* short description of the problem; less than ten words is good. In the "Type" field, select the type of your problem; also select the "Component" and "Versions" to which the bug relates.

In the "Comment" field, describe the problem in detail, including what you expected to happen and what did happen. Be sure to include whether any extension modules were involved, and what hardware and software platform you were using .

Each bug report will be assigned to a developer who will determine what needs to be done to correct the problem. You will receive an update each time action is taken on the bug.