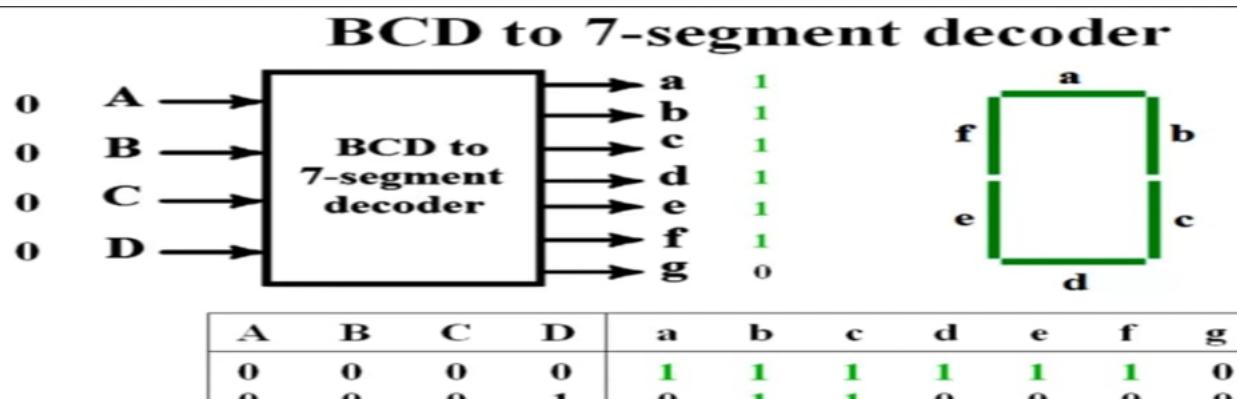


DAILY ASSESSMENT FORMAT

Date:	28/05/2020	Name:	Akshatha M Deshpande
Course:	Logic design	USN:	4AL17EC006
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:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

FORENOON SESSION DETAILS

Image of session



Report – Report can be typed or hand written for up to two pages.

Boolean equations for digital circuits, Combinational Circuits:-

Boolean Algebra: '0' & '1'

- * Cost of the circuit
 - * Simple realization of a circuit
- "In 1854, George Boole developed in algebraic system now called Boolean Algebra".

- * Boolean algebra is a system of mathematics logic
- * It is depended with set of elements a set of operators and a number of axioms or postulates.

Set of elements - {0, 1}

Two binary operators - OR & And
+ .

Unary operator - NOT

- * In boolean algebra

$$\begin{array}{ll} A+A = A & A \cdot A = A \\ 1+1 = 1 & 1 \cdot 1 = 1 \end{array}$$

- * In ordinary Algebra

$$\begin{array}{ll} A+A = 2A & A \cdot A = A^2 \\ 1+1 = 2 & 1 \cdot 1 = 1 \end{array}$$

- * In Binary number system

$$1+1 = (10) \quad 1 \cdot 1 = 1$$

Laws of Boolean Algebra

1) Commutative law :- $x+y = y+x$ or $x \cdot y = y \cdot x$

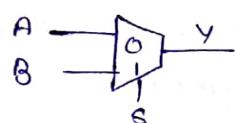
2) Associative law :- $x+(y+z) = (x+y)+z$
 $x \cdot (y \cdot z) = (x \cdot y) \cdot z$

3) Distributive : $x(y+z) = xy+xz$
 $x+yz = (x+y)(x+z)$



MUX To logic gates :-

- * Nand, nor \rightarrow Universal gate.
- * MUX, decoder \rightarrow Universal logic.
- * Now we see how a 2:1 mux can be used to create different logic gates.



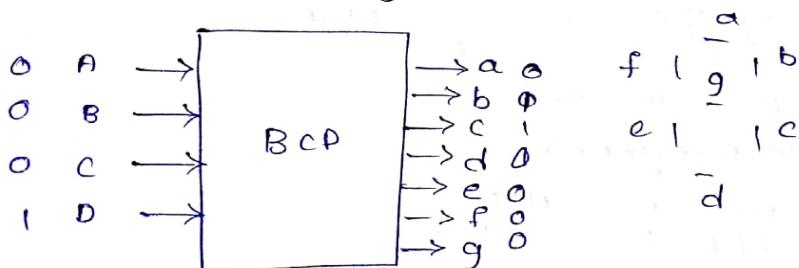
S	y
0	A
1	B

\downarrow

$y = A\bar{S} + BS$

Design of 7 segment decoder:-

BCD to 7-segment



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	0	1	0	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1

Date: 28/05/2020
Course: Python
Topic: Application :Build a desktop database application

Name: Akshatha M Deshpande
USN: 4AL17EC006
Semester & 6th Sem A sec
Section:

AFTERNOON SESSION DETAILS

Image of session

The screenshot shows a Windows desktop environment with the Atom code editor open. The left sidebar shows a project structure with files: Demo, __pycache__, backend.py, books.db, and frontend.py. The right pane contains the content of the frontend.py file.

```
184. Connecting the Frontend to the Backend, Part 1
frontend.py — D:\pp\Tkinter_sqlite3\Demo — Atom
File Edit View Selection Find Packages Help
Demo
backend.py
frontend.py
books.db
frontend.py

1 import backend
2
3
4 def view_command():
5     list1.delete(0,END)
6     for row in backend.view():
7         list1.insert(END,row)
8
9 def search_command():
10    list1.delete(0,END)
11    for row in backend.search(title_text.get(),author_text.get(),year_text.get(),isbn_text.get()):
12        list1.insert(END,row)
13
14 def add_command():
15     backend.insert(title_text.get(),author_text.get(),year_text.get(),isbn_text.get())
16     list1.delete(0,END)
17     list1.insert(0,"")
18     f insert
19 window INSERT
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
99]
```

The terminal window at the bottom shows the command-line interface for running the application:

```
PS D:\pp\Tkinter_sqlite3\Demo> python .\backend.py
[]
PS D:\pp\Tkinter_sqlite3\Demo> python .\backend.py
[(4, 'The moon', 'John Smooth', 1917, 99999)]
PS D:\pp\Tkinter_sqlite3\Demo> python frontend.py
[(4, 'The moon', 'John Smooth', 1917, 99999)]
PS D:\pp\Tkinter_sqlite3\Demo> python frontend.py
[(4, 'The moon', 'John Smooth', 1917, 99999)]
PS D:\pp\Tkinter_sqlite3\Demo>
```

Report – Report can be typed or hand written for up to two pages.

Build a Web/Desktop DataBase Application.

- * We have to use two parts in this
 - front end
 - Backend
- * We are going to use graphical user interface in this.
 - front end → code of tkinter library.
 - Backend → sqlite3 , the program that interacts with databaseWe can choose any first.
- * We want to show our program to open,choose and close the window.
- * We can just backend and print the program if we don't have frontend function that is GUI.
- * A sketch has to be made to let the program know how we want the window box.
 - This is between " "
- * Add this is called as User interface design.
- * This opens a window.

Frontend :-

- * This is build using tkinter function.
- * Title , year , Author , ISBN text boxes are created.



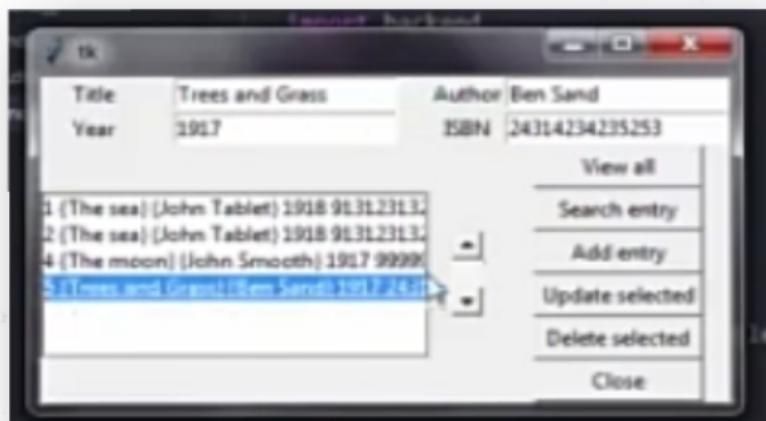
- * Then the list boxes are created
 - Giving required height & width
 - rowspan & colspan is used.
 - scroll bar is added using

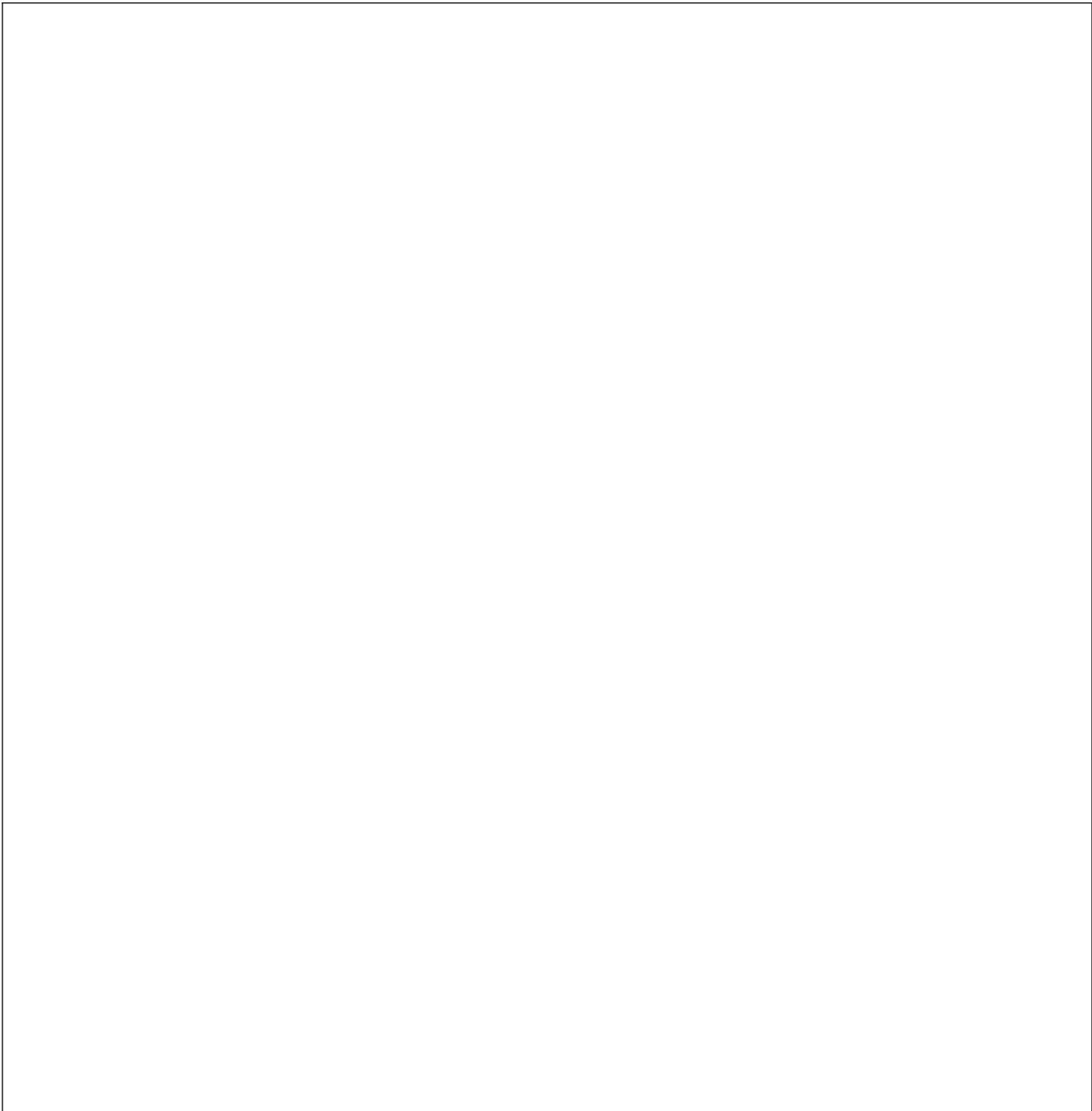

```
.scrollbar(window)
sbl.grid(row=1, column=3) # where we
require the scroll bar.
```
 - config method is used to configure.
- * Then the 5 buttons are created viewall, search entry, add entry, update, delete, close using button & grid.

Backend :-

- * sqlite 3 is used in this.
- * Database is created.

And the connection between front end & back end is made then the window of book store is opened with all the bars created.





Edit with WPS Office