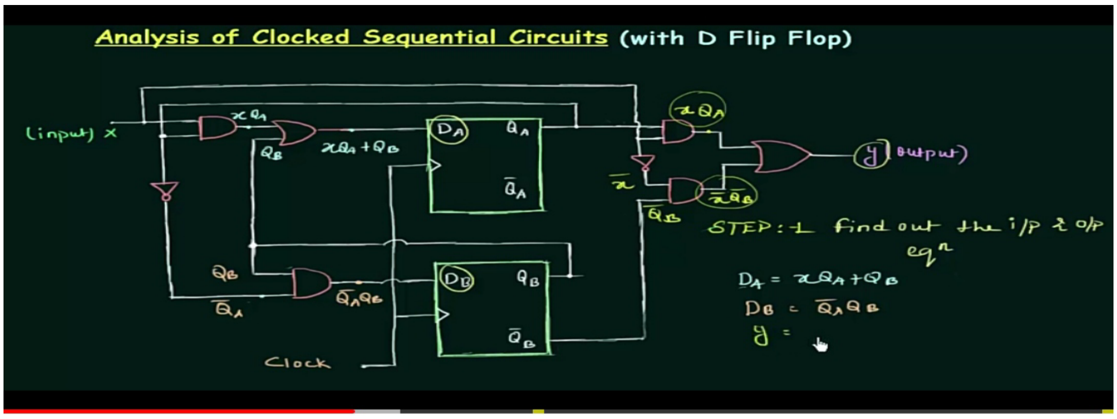
Date:	**29-05-2020**	**Name:**	**Rajeshwari Gadagi**
Course:	**Logic design**	**USN:**	**4AL17EC076**
Topic:	Analysis of clocked sequential circuit	**Semester and section:**	**6th sem and B sec**



Analysis of Clocked Sequential Circuits (with D Flip Flop)

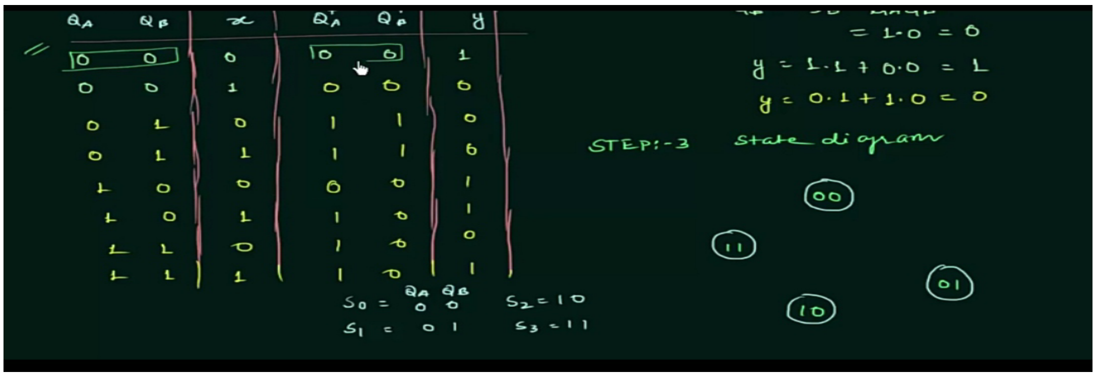
STEP:-1 find out the i/p & o/p eq^n

$D_A = xQ_A + Q_B$

$D_B = \bar{Q}_A Q_B$

$y =$

Analysis of Clocked Sequential Circuits (with D Flip Flop)



Q_A	Q_B	x	Q_A^+	Q_B^+	y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

$= 1\cdot 0 = 0$

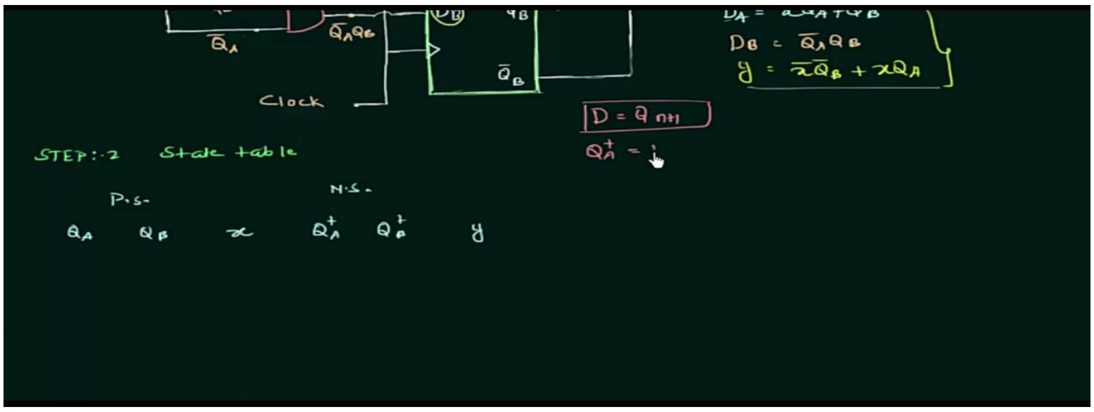
$y = 1\cdot 1 + 0\cdot 0 = 1$

$y = 0\cdot 1 + 1\cdot 0 = 0$

STEP:-3 state diagram

$S_0 = \overset{Q_A\ Q_B}{0\ \ 0}$ $\quad S_2 = 10$

$S_1 = 0\ 1$ $\quad S_3 = 11$



$D_B = \bar{Q}_A Q_B$

$y = \bar{x}\bar{Q}_B + xQ_A$

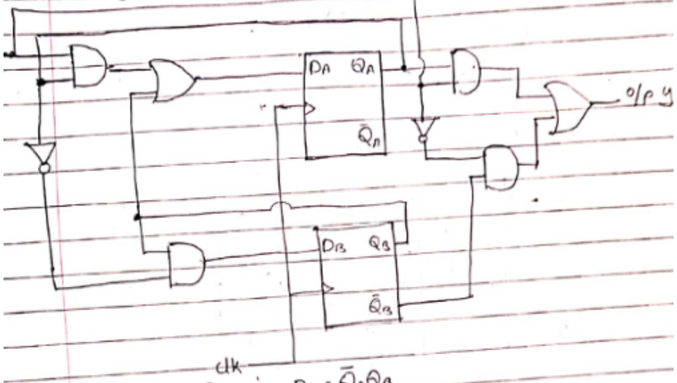
$\boxed{D = Q_{n+1}}$

$Q_A^+ =$

STEP:-2 State table

P.S.			N.S.		
Q_A	Q_B	x	Q_A^+	Q_B^+	y

Analysis Of Clocked Sequential Circuits with D FF



$D_A = xQ_A + Q_B$ $\quad$ $D_B = \bar{Q}_A Q_B$

$y = \bar{x}\bar{Q}_B + xQ_A$

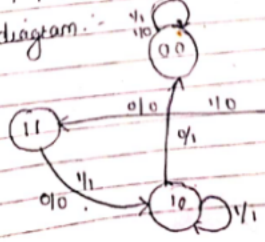
$Q_A^+ = D_A$

$Q_B^+ = D_B$

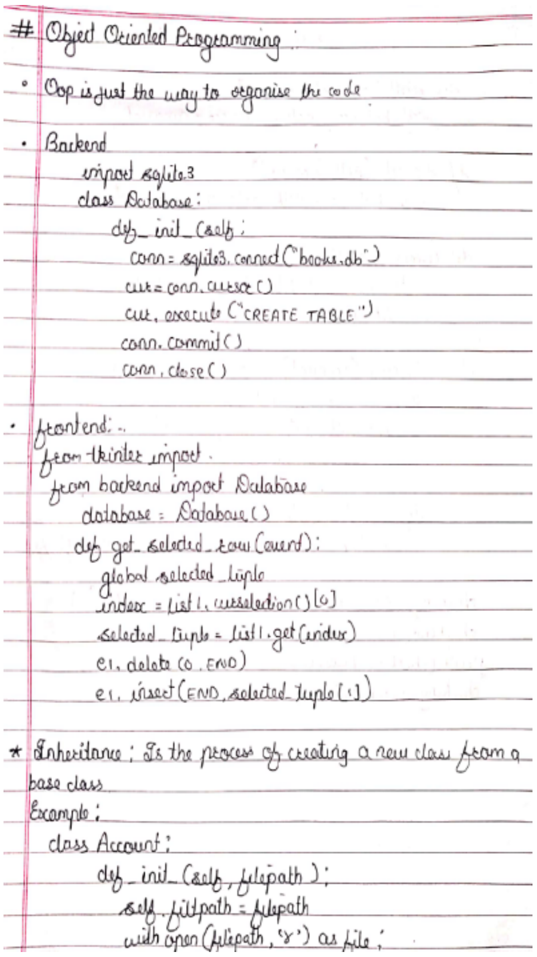
State table.

P.S			N.S		
Q_A	Q_B	x	Q_A^+	Q_A^+	y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

State diagram.



Date:	29-05-2020	Name:	Rajeshwari Gadagi
Course:	Python programming	USN:	4AL17EC076
Topic:	Object oriented programming	Semester and section:	6th sem and B sec



Object Oriented Programming

- Oop is just the way to organise the code
- Backend

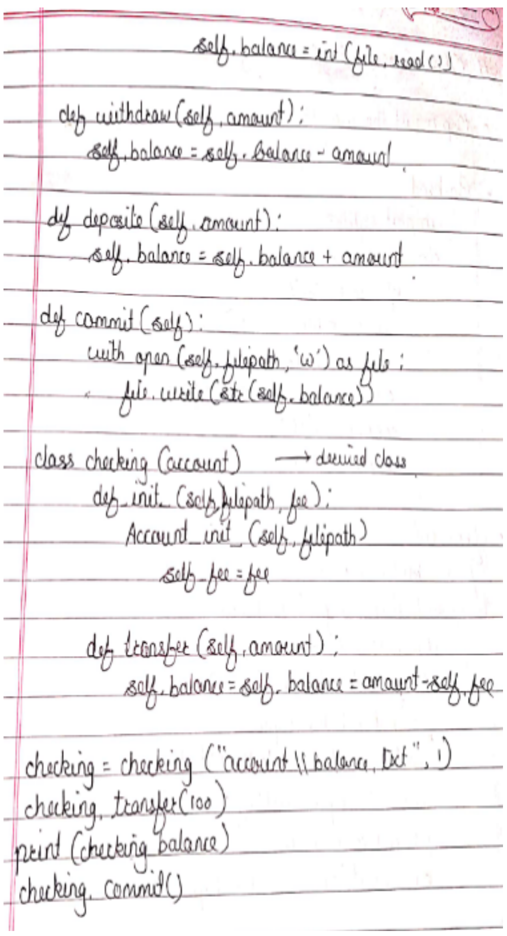
```
    import sqlite3
    class Database:
        def_init_(self;
            conn= sqlite3.connect("books.db")
            cur= conn.cursor()
            cur.execute("CREATE TABLE")
            conn.commit()
            conn.close()
```

- frontend:..

```
 from tkinter import
  from backend import Database
     database = Database()
    def get_selected_row(event):
        global selected_tuple
        index = list1.cursselection()[0]
        selected_tuple = list1.get(index)
        e1.delete(0.END)
        e1.insert(END,selected_tuple[1])
```

* Inheritance; Is the process of creating a new class from a base class
Example;

```
  class Account;
        def_init_(self,filepath);
            self.filtpath = filepath
            with open(filepath,'r') as file;
```



```
                    self.balance = int(file.read())

    def withdraw(self,amount);
        self.balance = self.balance - amount

    def deposite(self.amount):
        self.balance = self.balance + amount

    def commit(self):
        with open(self.filepath,'w') as file;
            file.write(str(self.balance))

class checking (account)      → derived class
        def_init_(self,filepath, fee);
            Account_init_(self,filepath)
                self_fee = fee

        def transfer(self,amount);
            self.balance=self.balance = amount-self.fee

checking = checking ("account\\balance.txt",1)
checking.transfer(100)
print (checking balance)
checking.commit()
```