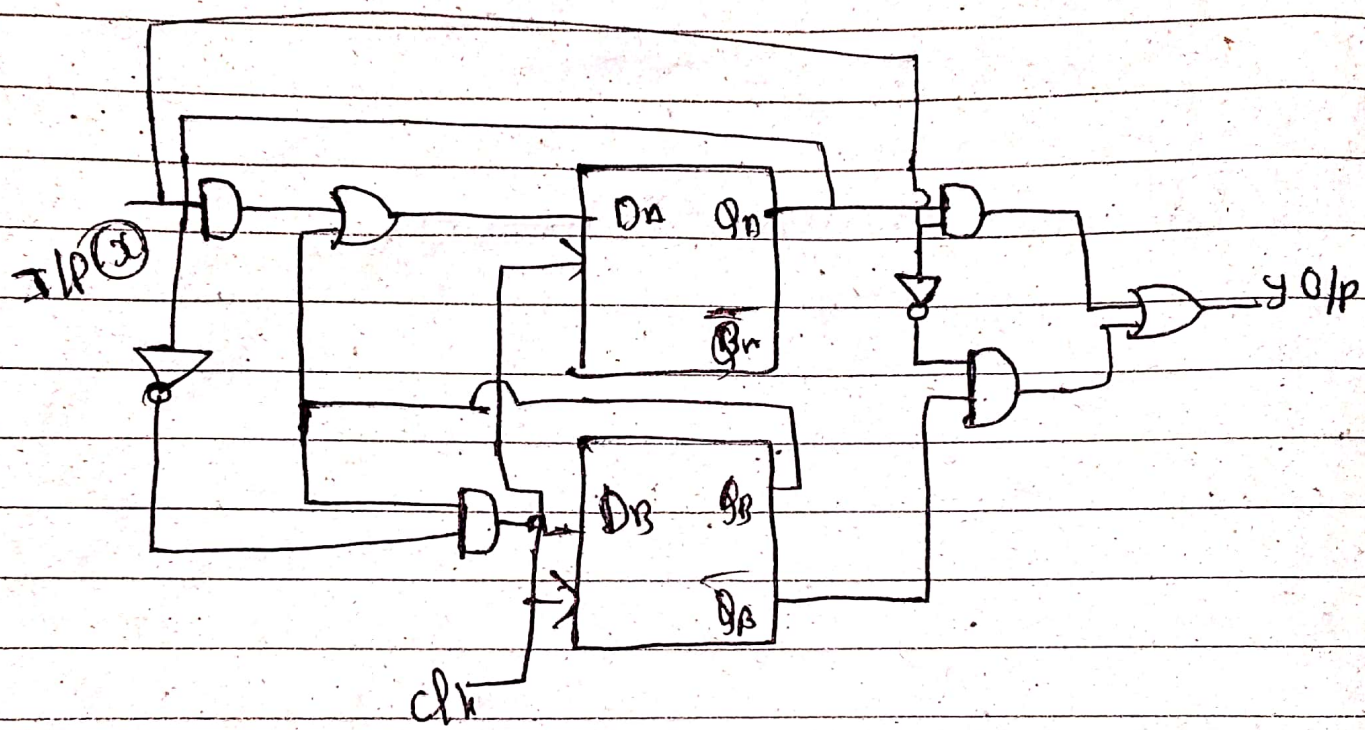
Logic design

Poojary Sushmita S
4AL16EC046
GB → 8 29/5/2020

Analysis of clocked sequential ckt (with DFF)



step 1 = Find out the i/p o/p eqn.

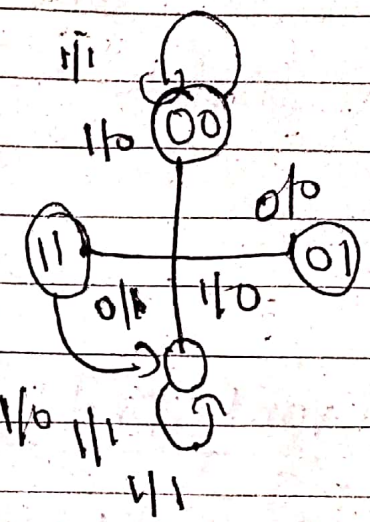
$D_A = x\bar{Q}_A + Q_B$

$D_B = \bar{Q}_A Q_B$

$Y = \bar{x}\bar{Q}_B + xQ_B$

step 2 state table.

P.S			N.S		
Q_A	Q_B	x	Q_A	Q_B	Y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	1	1	1	0
0	1	0	1	1	0
1	0	1	1	1	0
1	0	0	0	0	1



Digital clk design.

- digital clk is a type of clk that displays the time digitaly.
- digital clks are often associated with electronic device the digital description refers only the display not the desire mechanism.
- Because digital clk can be very small & inspires device.

Python programing:-

Poojary Sushmita
4AL16EC046 → 6B

* Object oriented programing.

- Oops the way to organise the code
- Backend

```
import Sqlite3
class Database
    def init_(self,
        conn= sqlite.connect('books.db')
        cur = conn.cursec()
        cur.execute ("CREATE TABLE")
        conn.commit()
        conn.close()
```

- front end:-

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

* Inheritence:- Is the process of create a new class from a base class.

Ex:-

```
class Account:
```

```
def __init__(self, filepath):
    self.filepath = 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) -> desired class
    def __init__(self, filepath, fee):
        Account.__init__(self, filepath)
        self.fee = fee

    def tranfer(self, amount):
        self.balance = self.balance - amount - self.fee

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