

## DAILY ASSESSMENT FORMAT

Date:	29-05-2020	Name:	M V Ramya
Course:	DSP	USN:	4AL17EC045
Topic:	Fourier transform	Semester & Section:	6th sem, A sec
Github Repository:	M V Ramya-045		

### FORENOON SESSION DETAILS



# 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

$S_0 = 00$   
 $S_1 = 01$   
 $S_2 = 10$   
 $S_3 = 11$

$$= 1 \cdot 0 = 0$$

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

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

STEP:-3 state diagram



10:48 / 13:29

Scroll for details



Edit with WPS Office

29/5/2020

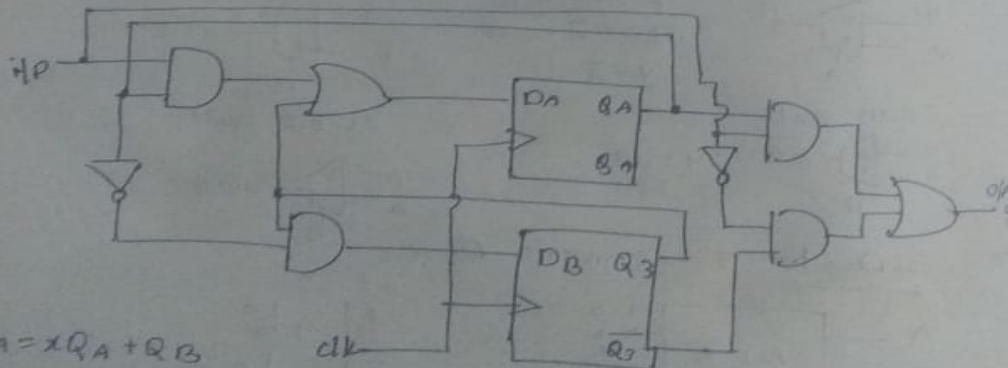
- 'A sequential circuit combination of logic circuit that consists of ip variables and op variables'
- Sequential circuit produces an op based on current input and previous input variables.
- There are two types of ip to the combination logic :- (i) External ip which not controlled by the circuit.  
(ii) Internal ip which are a function of a previous output status

### Types of sequential circuits

Asynchronous sequential ckt

Synchronous sequential ckt

\* Analysis of clocked sequential circuits with DFF



$$D_1 = xQ_1 + Q_2$$

$$Y = xQ_2 + xQ_1$$

--

Date:29may2020

Name: MV Ramya

Course: python

USN:4AL17EC045

Topic: python

Semester &  
Section:

6th sem Asec

AFTERNOON SESSION DETAILS
---------------------------

Image of session
------------------



The screenshot shows a Python IDE with a file named 'acc.py'. The code defines a class 'Checking' with a class variable 'fee' (100), an instance variable 'balance' (100), and methods 'transfer' and 'commit'. The class is instantiated twice: 'jacks\_checking' and 'johns\_checking'. The terminal output shows the execution of the script, displaying the class type and balance for each instance.

```

class Checking:
    fee = 100
    balance = 100

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

    def commit(self):
        pass

jacks_checking = Checking("account\\jack.txt", 1)
jacks_checking.transfer(100)
print(jacks_checking.balance)
jacks_checking.commit()
print(jacks_checking.type)

johns_checking = Checking("account\\john.txt", 1)
johns_checking.transfer(100)
print(johns_checking.balance)
johns_checking.commit()
print(johns_checking.type)

print(johns_checking.__doc__)

```

Terminal output:

```

PS D:\Dropbox\pp\classes\Demo> python account\acc.py
-200
checking
-100
checking
This class generates checking account objects
PS D:\Dropbox\pp\classes\Demo>

```

- The approach to solve a programming problem is by creating objects. This is known as OOP.
- An object has 2 characteristics :- attributes  
Behaviour.
- We learnt on converting the frontend and backend design and approaches involved for turning an application in OOP style.
- Inheritance :- Inheritance is a way of creating new class for using details of existing class without modifying it.
- Class :- A user-defined prototype for an object that defines a set of attributes that characterize any object of the class.
- Instance variable :- A variable that is defined inside a method and belongs only to current instance of a class.
- Object instance :- An individual object of a certain class. An object obj that belongs to a class circle.
- Method :- A special kind of function i.e. defined in class definition.
- Instantiation :- The creation of an instance of a class.
- Constructor :- A constructor is a special kind of method that python calls when it instantiates an object using the definition found in class.



Exit



13

