

## **DAILY ASSESSMENT FORMAT**

<b>Date:</b>	<b>29-05-2020</b>	<b>Name:</b>	<b>BINDUSHRI</b>
<b>Course:</b>	<b>Logic designs</b>	<b>USN:</b>	<b>4AL17EC011</b>
<b>Topic:</b>	<b>1.analysis of clocked sequential circuits. 2.DFF analysis 3.analysis of clocked Sequential circuits.</b>		<b>6<sup>th</sup> A</b>
<b>Github Repository:</b>	<b>Bindushri</b>		

### **FORENOON SESSION DETAILS**

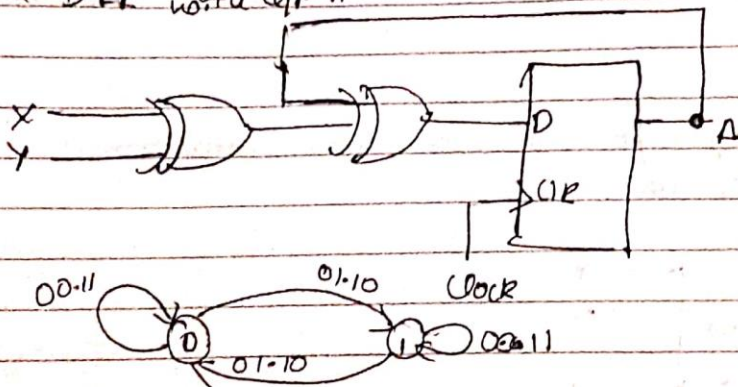
Day 2:- 29-05-2020

## Analysis of clocked sequential circuits

- \* I/P that set the FF to 1 is called preset or direct set
- \* I/P that clears the FF to 0 is called clear or direct clear
- \* the part of the circuit that generates the input to FF is described algebraically by a set of Boolean functions called FF input equation
- \* the information observable in a state table could be represented graphically in the form of a state diagram
- \* In general a sequential circuit with 'm' FF and 'n' I/Ps needs  $2^{m+n}$  rows in the ST

### DE analysis

I/P eqn of a DFF is given by  $D_A = A \oplus X \oplus Y$  - DFF with I/P A



Analysis with JK :-  $J_A = B$ ;  $K_A = Bx'$

$$J_B = x'; K_B = A'x + Ax' = A \oplus x$$

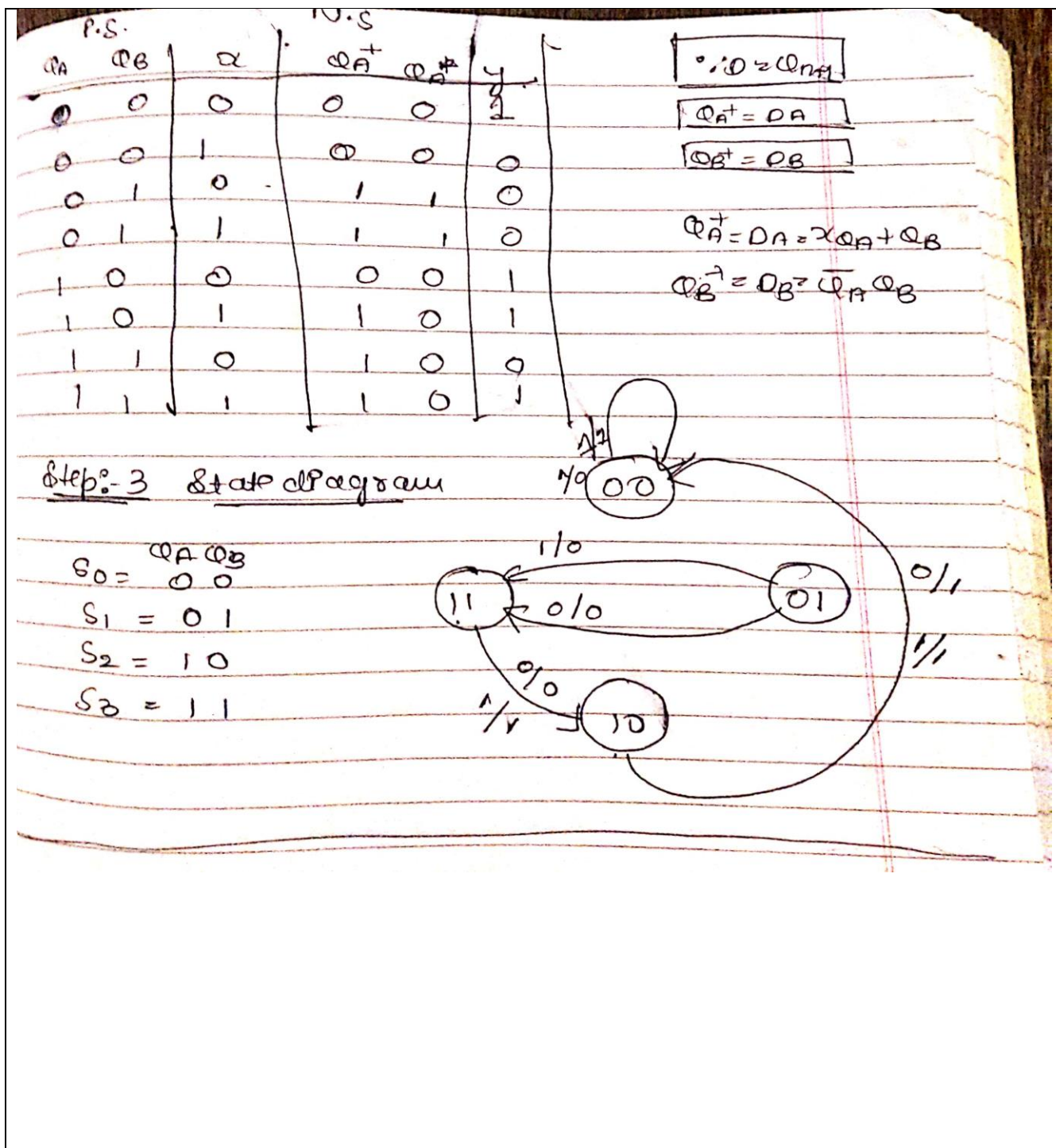
### Analysis of clocked SC (with DFF)

Step 1:- Find o/p the I/P & o/p eqn.

$$D_A = 2Q_A + 0B, D_B = \overline{Q_A}Q_B, Y = 2Q_A + \overline{2}Q_B$$

Step 2: State table

P.S



Date:29may2020  
 Course: python  
 Topic:  
 Basics:sec25

Name:Bindushri  
 USN:4AL17EC011  
 Sem&Sec:6<sup>th</sup> A

#### AFTERNOON SESSION DETAILS

Image of session



29-05-2020

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

lec: 25 Python for Image and Video Processing  
with opencv

Method to open cv ~~code~~:-

→ open python & typed `import cv2`. If ~~the~~ you don't get error then opencv is installed.

\* to install

1. pip install opencv-python
2. open python then type `import cv2`

⇒ Reading, Displaying, Resizing and writing image  
→ galaxy.jpg ~~script in~~  
→ script.py.

1. `import cv2`

2. `img = cv2.imread("galaxy.jpg", 0)`

3. `print(type(img))`

4. `print(img)`

5. `print(img.shape)`

6. `print(img.ndim)`

7.

10. `resized_image = cv2.resize(img, (1000, 500))`

10. `resized_image = cv2.resize(img, (int(img.shape[1]/2), int(img.shape[0]/2)))`

11. `cv2.imshow("galaxy", resized_image)`

12. `cv2.imwrite("galaxy resized.jpg", resized_image)`

13. `cv2.waitKey(0)`

14. `cv2.destroyAllWindows()`

## \* Face Detection

```
1. import cv2
2.
3. face_cascade = cv2.CascadeClassifier('haarcascade_
   frontalface_default.xml')
4.
5. img = cv2.imread('photo.jpg')
6. gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
7.
8. faces = face_cascade.detectMultiScale(gray_img,
9.    scaleFactor=1.05,
10.    minNeighbors=5)
11. for x, y, w, h in faces:
12.     img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 3)
13. print("# of faces")
14. print(faces)
15. resized = cv2.resize(img, (int(img.shape[1]/3),
   (img.shape[0]/3)))

cv2.imshow("gray", gray_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## \* Capturing video

```
import cv2, time
video = cv2.VideoCapture(0)
a = 0
while True:
    a = a + 1
    check, frame = video.read()

    print(check)
    print(frame)
```



```

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
time.sleep(3)
cv2.imshow("frame", frame)
cv2.imshow("capturing", frame)
cv2.imshow("capturing", gray)

```

```

key = cv2.waitKey(1)

```

```

if key == ord('q'):
    break

```

```

print(a)

```

```

webcam.release()

```

```

cv2.destroyAllWindows()

```

X ——— X ——— X

**Tkinter code:**

FileEditSelectionViewGoRunTerminalHelp

script2.py - Untitled (Wo

EXPLORER

1

OPEN EDITORS1 UNSAVED

script2.py9+

UNTITLED (WORKSPACE)

tkinter

script2.py9+

script2.py

tkinter > script2.py > ...

1from tkinter import \*  
2window = Tk()  
3  
4def km\_to\_miles():  
5 print(e1\_value.get())  
6 miles=float(e1.value.get())\*1.6  
7 t1.insert(END,miles)  
8  
9  
10b1=Button(window,text="Execute",command=km\_to\_miles)  
11b1.grid(row=0,column=0)  
12  
13e1\_value=Stringar()  
14e1=Entry(window,textvariable=e1\_value)  
15e1.grid(row=0,column=1)  
16  
17t1=Text(window,height=1,width=20)  
18t1.grid(row=0,column=2)  
19  
20window.mainloop()

PROBLEMS100OUTPUTTERMINALDEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell <https://aka.ms/pscore6>  
  
PS C:\Users\hp\Desktop\tkinter> python script2.py  
Traceback (most recent call last):  
 File "script2.py", line 20, in <module>  
 window.mainloop()  
 File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.8  
 self.tk.mainloop(n)  
KeyboardInterrupt  
PS C:\Users\hp\Desktop\tkinter> python script2.py  
█

OUTLINE  
TIMELINE

## EXPLORER

## OPEN EDITORS

X script3.py 1

## UNTITL...

## database

lite.db

multi.py

script3.py 1

script3.py X

database &gt; script3.py &gt; insert

```
1 import sqlite3
2
3 def create_table():
4     conn = sqlite3.connect('lite.db')
5     cur = conn.cursor()
6     cur.execute("CREATE TABLE IF NOT EXISTS store (item TEXT,
7     conn.commit()
8     conn.close()
9
10
11
12 def insert(item, quantity, price ):
13     conn = sqlite3.connect('lite.db')
14     cur = conn.cursor()
15     cur.execute("INSERT INTO store VALUES(?,?,?)", (item, quan
16     conn.commit()
17     conn.close()
18 create_table()
19 insert("water glass",500,20000)
20
21
22 def view():
23     conn = sqlite3.connect('lite.db')
24     cur = conn.cursor()
25     cur.execute("SELECT * FROM store")
26     rows = cur.fetchall()
27     return rows
28     conn.close()
29
30 print(view())
```

PROBLEMS 1

OUTPUT

TERMINAL

DEBUG CONSOLE

```
[('biscuits', 500, 20000.0)]
PS C:\Users\hp\Desktop\database> python script3.py
[('biscuits', 500, 20000.0), ('coffee cup', 500, 20000.0)]
PS C:\Users\hp\Desktop\database> python script3.py
[('biscuits', 500, 20000.0), ('coffee cup', 500, 20000.0), ('water glass
PS C:\Users\hp\Desktop\database> ]
```

&gt; OUTLINE

&gt; TIMELINE



EXPLORER

OPEN EDITORS

multi.py9+

UNTITL...

database

- lite.db
- multi.py9+
- script3.py

multi.py

database > multi.py > ...

25

26 e2\_value=StringVar()

27 e2=Entry(window,textvariable=e2\_value)

28 e2.grid(row=0,column=1)

29

30

31 b1=Button(window,text="Convert",command=from\_kg)

32 b1.grid(row=0,column=2)

33

34

35 t1=Text(window,height=1,width=20)

36 t1.grid(row=1,column=0)

37

38 t2=Text(window,height=1,width=20)

39 t2.grid(row=1,column=1)

40

41 t3=Text(window,height=1,width=20)

42 t3.grid(row=1,column=2)

43

44

45 window.mainloop()

PROBLEMS100

OUTPUT

TERMINAL

DEBUG CONSOLE

File "script3.py", line 25

print(view())

^

IndentationError: unindent does not match any outer indentation level

PS C:\Users\hp\Desktop\database> python script3.py

PS C:\Users\hp\Desktop\database> python script3.py

PS C:\Users\hp\Desktop\database> python script3.py

PS C:\Users\hp\Desktop\database> python script3.py

Traceback (most recent call last):

File "script3.py", line 25, in <module>

print(view())

File "script3.py", line 21, in view

cur.execute("SELECT\*FROM store")

sqlite3.OperationalError: no such table: store

PS C:\Users\hp\Desktop\database> python multi.py

> OUTLINE

> TIMELINE

thon 3.8.3 32-bit 0 100 Connect

## Script2.py code:

```
from tkinter import *
window = Tk()

def km_to_miles():
    print(e1_value.get())
    miles=float(e1.value.get())*1.6
    t1.insert(END,miles)

b1=Button(window,text="Execute",command=km_to_miles)
b1.grid(row=0,column=0)

e1_value=StringVar()
e1=Entry(window,textvariable=e1_value)
e1.grid(row=0,column=1)

t1=Text(window,height=1,width=20)
t1.grid(row=0,column=2)

window.mainloop()
```

## script3.py code:

```
import sqlite3

def create_table():
    conn = sqlite3.connect('lite.db')
    cur = conn.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS store (item TEXT, quantity INTEGER, price REAL)")
    conn.commit()
    conn.close()

def insert(item, quantity, price ):
    conn = sqlite3.connect('lite.db')
    cur = conn.cursor()
    cur.execute("INSERT INTO store VALUES(?,?,?)", (item, quantity, price))
    conn.commit()
```

```
conn.close()
create_table()
insert("water glass",500,20000)

def view():
    conn = sqlite3.connect('lite.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM store")
    rows = cur.fetchall()
    return rows
    conn.close()

print(view())
```

## multi.py code:

```
from tkinter import *

window=Tk()

def from_kg():

    gram=float(e2_value.get())*1000

    pound=float(e2_value.get())*2.20462

    ounce=float(e2_value.get())*35.274

    t1.delete("1.0", END)
    t1.insert(END,gram)
    t2.delete("1.0", END)
    t2.insert(END,pound)
    t3.delete("1.0", END)
    t3.insert(END,ounce)

e1=Label(window,text="Kg")
e1.grid(row=0,column=0)
```



```
e2_value=StringVar()
e2=Entry(window,textvariable=e2_value)
e2.grid(row=0,column=1)

b1=Button(window,text="Convert",command=from_kg)
b1.grid(row=0,column=2)

t1=Text(window,height=1,width=20)
t1.grid(row=1,column=0)

t2=Text(window,height=1,width=20)
t2.grid(row=1,column=1)

t3=Text(window,height=1,width=20)
t3.grid(row=1,column=2)

window.mainloop()
```



✓ Talk :- by Mouan. Preparation of next normal

### Business impact

- \* Spotting disruptive business and people early
- \* 75% of total workspace by 2025
- ↓ the urgency of education
- \* \$96B crowdfunding activity by 2025
- \* Continuous evolution of your social media strategy
- \* 50% of college courses will be offered online

~~Deter Daker~~ - ~~If you do~~

### Post Covid-19 world

1. Shift towards localisation
2. Digital gets a real push
3. Comes King for business



④ move towards variable cost model

Preparation for next normal

Resilient Dynamism.

→ ~~Key~~ Industries outlook:-

1) Health & Life Sciences 2) IT 3) Education  
4) Retail 5) Public Sector 6) Telecommunication  
7) Wellness 8) Insurance 9) Agriculture  
10) online platforms L&D. 11) Social media

3 R post-covid-19 strategies

1) Rescue 2) Revitalize 3) Repetition of zero

→ Data Analysis

conduct Descriptive, Diagnostic, Predictive  
~~and~~

→ ~~TECHNOLOGICAL TRENDS~~

→ A new way of learning & working

\* Design Thinking, Innovation.

↓

Desirability, Viability, Feasibility









