NAME	APEKSHA S SHETTY	USN	4AL16EC006
COURSE	Logic design	SEM AND SEC	8 A SEC
DATE	29 MAY 2020	Topic	Analysis of clocked sequential circuits, Digital clock design.
GitHub REPOSITORY	Apeksha-97		

IMAGE SECTION

COMBIBATIONAL CIRCUIT

Analysis of Clocked Sequential Circuits (with D Flip Flop)

STEP:1 find out the i/p & o/p eqn

$$D_A = x Q_A + Q_B$$
$$D_B = \bar{Q_A} Q_B$$
$$y = \bar{x} \bar{Q_B} + x Q_A$$

A sequential circuit is a logical circuit, where the output depends on the present value of the input signal as well as the sequence of past inputs. While a combinational circuit is a function of present input only.

A sequential circuit is a combination of combinational circuit and a storage element. the sequential circuits use current input variables and previous input variables which are stored and provides the data to the circuit on the next clock cycle.

Types of Sequential Circuits

The sequential circuits are classified into two types

Synchronous Circuit

Asynchronous Circuit

In synchronous sequential circuits, the state of device changes at discrete times in response to a clock signal. In asynchronous circuits, the state of the device changes in response to changing inputs.

Synchronous Circuits

In synchronous circuits, the inputs are pulses with certain restrictions on pulse width and propagation delay. Thus synchronous circuits can be divided into clocked and un-clocked or pulsed sequential circuits.

Clocked Sequential Circuit

The clocked sequential circuits have flip-flops or gated latches for its memory elements. There is a periodic clock connected to the clock inputs of all the memory elements of the circuit to synchronize all the internal changes of state. Hence the operation of the circuit is controlled and synchronized by the periodic pulse of the clock.

Asynchronous Circuits

An asynchronous circuit does not have a clock signal to synchronize its internal changes of the state. Hence the state change occurs in direct response to changes that occur in primary input lines. An asynchronous circuit does not require the precise timing control from flip-flops.

Flip Flop Circuit

A flip-flop is a sequential circuit which samples the input and changes the output at a particular instance of time. It has two stable states and can be used to store the state information. Signals are applied to one or more control inputs to change the state of the circuit and will have one or two outputs.

It is the basic storage element in sequential logic and fundamental building blocks of digital electronic systems. They can be used to keep a record of the value of a variable. Flip-flop is also used to control the functionality of a circuit.

RS Flip Flop
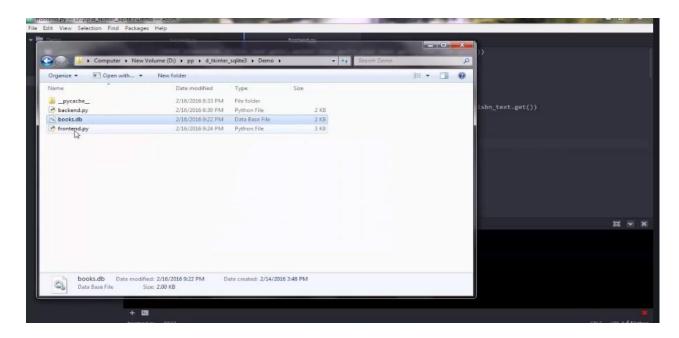
The R-S flip-flop is the simplest flip-flop. It has two outputs, one output is the reverse of the other, and two inputs. The two inputs are Set and Reset. The flip-flop basically uses NAND gates with an additional enable pin. The circuit gives output only when the enable pin is high.

JK Flip Flop

JK flip-flop is one of the important flip-flops. If the J and K inputs are one and when the clock is applied, the output changes regardless of past condition. If the J and K inputs are 0 and when the clock is applied, there will be no change in the output. There is no indeterminate condition in the JK flip-flop.

NAME	APEKSHA S SHETTY	USN	4AL16EC006
COURSE	PYTHON	SEM AND SEC	8 A SEC
DATE	29 MAY 2020		
GitHub REPOSITORY	Apeksha-97		

IMAGE SECTION

```
script1.py — D:\pp\database section — Atom
File Edit View Selection Find Packages Help
  database section                    script1.py
    lite.db                    1   import sqlite3
    script1.py                 2
                               3   conn=sqlite3.connect("lite.db")
                               4   cur=conn.cursor()
                               5   cur.execute("CREATE TABLE store (item TEXT, quantity INTEGER, price REAL)")
                               6   conn.commit()
                               7   conn.close()
                               8

Traceback (most recent call last):
  File ".\script1.py", line 3, in <module>
    conn=qlite3.connect("lite.db")
NameError: name 'qlite3' is not defined
PS D:\pp\database section> python .\script1.py
PS D:\pp\database section> python .\script1.py
Traceback (most recent call last):
  File ".\script1.py", line 5, in <module>
    cur.execute("CREATE TABLE store (item TEXT, quantity INTEGER, price REAL)")
sqlite3.OperationalError: table store already exists
PS D:\pp\database section>
```

REPORT:

These are the frontend.py and backend.py scripts in OOP style. To execute this program you should execute the frontend.py file.

#frontend.py

from tkinter import *

from   backend

import

Database

database=Data

base("books.d

b"   class

Window(objec

t):

def__init

(self,window):

```python
        self.window =
        window
        self.window.wm_tit
        le("BookStore")
        l1=Label(window,t
        ext="Title")
        l1.grid(row=0,colu
        mn=0)
        l2=Label(window,t
        ext="Author")
        l2.grid(row=0,colu
        mn=2)

        l3=Label(window,text="Year")
        l3.grid(row=1,column=0)
        l4=Label(window,text="ISBN")
        l4.grid(row=1,column=2)
        self.title_text=StringVar()
        self.e1=Entry(window,textvariable
        =self.title_text)
        self.e1.grid(row=0,column=1
        self.author_text=StringVar()

        self.e2=Entry(window,textvariable=self
        .author_text)
        self.e2.grid(row=0,column=3)
```

```
self.year_text=StringVar()

self.e3=Entry(window,textvariable=self
.year_text)

self.e3.grid(row=1,column=1.

self.isbn_text=StringVar()

self.e4=Entry(window,textvariable=self
.isbn_text)

self.e4.grid(row=1,column=3)

self.list1=Listbox(window,
height=6,width=35)

self.list1.grid(row=2,column=0,rowspan=6
,columnspan=2) sb1=Scrollbar(window)

sb1.grid(row=2,column=2,rowspan=6)

self.list1.configure(yscrollcommand=sb1.s
et)

sb1.configure(command=self.list1.yview)

self.list1.bind('<<ListboxSelect>>',self.get
_selected_row)

b1=Button(window,text="View all",
width=12,command=self.view_command)

b1.grid(row=2,column=3)
```

```python
b2=Button(window,text="Search entry",
width=12,command=self.search_command)
b2.grid(row=3,column=3)

b3=Button(window,text="Add entry",
width=12,command=self.add_command)
b3.grid(row=4,column=3)

b4=Button(window,text="Update selected",
width=12,command=self.update_command) b4.grid(row=5,column=3)

b5=Button(window,text="Delete selected",
width=12,command=self.delete_command) b5.grid(row=6,column=3)

b6=Button(window,text="Close",
width=12,command=window.destroy)
b6.grid(row=7,column=3)

def
get_selected_row(self,ev
ent):
index=self.list1.curselecti
on()[0]
self.selected_tuple=self.li
st1.get(index)
self.e1.delete(0,END)
self.e1.insert(END,self.se
lected_tuple[1])
```

```python
        self.e2.delete(0,END)

        self.e2.insert(END,self.se

        lected_tuple[2])

        self.e3.delete(0,END)

        self.e3.insert(END,self.se

        lected_tuple[3])

        self.e4.delete(0,END)

        self.e4.insert(END,self.se

        lected_tuple[4])

        def view_command(self):

        self.list1.d

        elete(0,EN

        D) for row

        in

        database.v

        iew():

self.list1.in

sert(END,r

ow def

search_co

mmand(sel

f):

self.list1.delete(0,END)
```