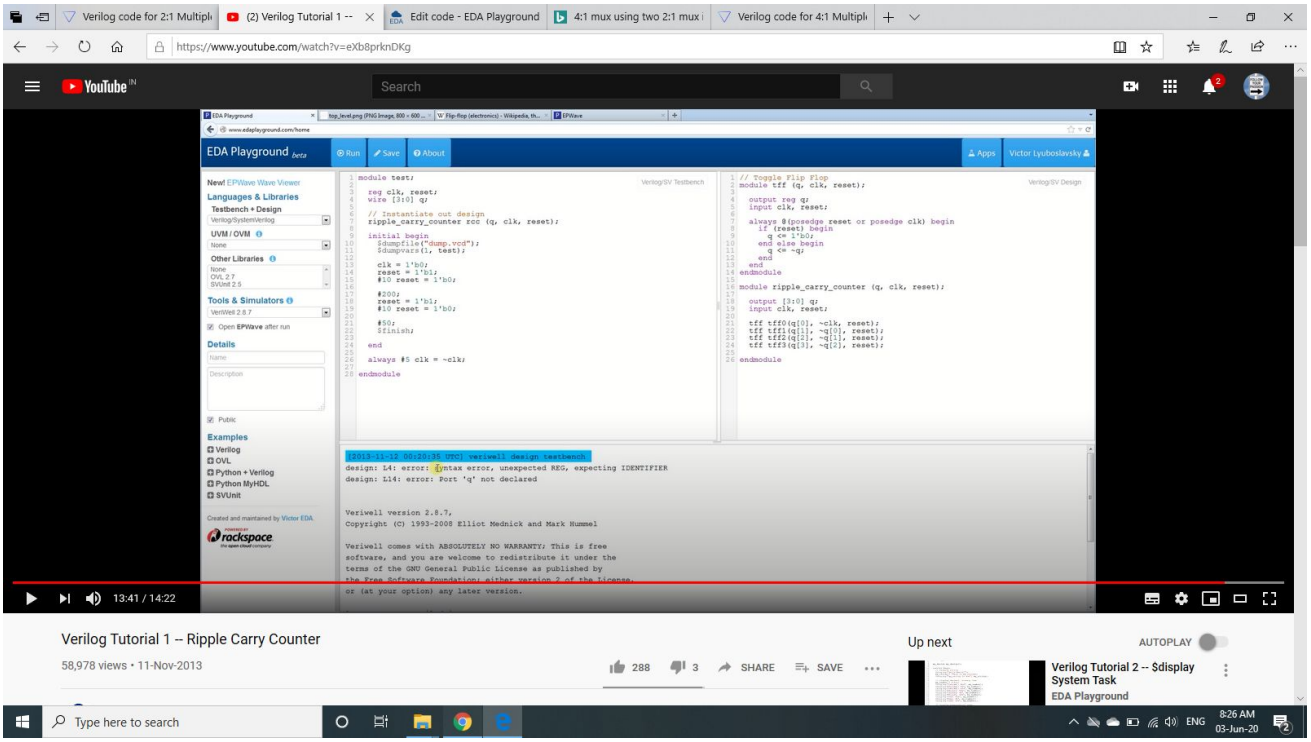
REPORT JUNE 03

Date:	03 JUNE 2020	Name:	Rakshith B
Course:	Digital Design Using HDL	USN:	4AL16EC409
Topic:	EDA Playground Online compiler , EDA Playground Tutorial Demo Video , How to Download And Install Xilinx Vivado Design Suite , Task for Day-3	Semester & Section:	6th SEM B
Github Repository:	Rakshith-B		

FORENOON SESSION DETAILS

Image of session



Report –EDA Playground Tutorial Demo
Implement Inverter Using The EDA Tool

```
module inverter(y,a);
output y;
input a;
assign y=~a;
endmodule
```

Testbench Code

```
timescale ins/lps
```

```
module testbench();
reg a1;
wire y1;
inverter inv1(a1,y1);
initial begin
    a1=a'b1;
    $display("a=%b",a1);
end
endmodule
```

Ripple Carry Counter

```
module ripple_counter_4_bit(q,clk,reset);
input clk,reset;
output[3:0]q;
T_FF tff0(q[0],clk,reset);
T_FF tff1(q[1],q[0],reset);
T_FF tff2(q[2],q[1],reset);
T_FF tff3(q[3],q[2],reset);
endmodule
module T_FF(q,clk,reset);
input clk,reset;
output q;
wire d;
D_FF dff0(q,d,clk,reset);
not n1(d,q);
endmodule
module D_FF(q,d,clk,reset);
input d,clk,reset;
output reg q;
always@(negedge clk or posedge reset)
begin
if(reset)
q<=1'b0;
else
q<=d;
end
endmodule
```

TestBench Code

```
module test
  reg clk,reset;
  wire(3:0)q;
  ripple_carry_counter rcc(q,clk,reset);
  initial begin
```

```
    $dumpfile("dump.vcd");
    $dumpvars(1,test);
    clk=1'b0;
    reset=1'b1;
    #10 reset=1'b0;
    #200;
  end
  always #5 clk=~clk;
endmodule
```

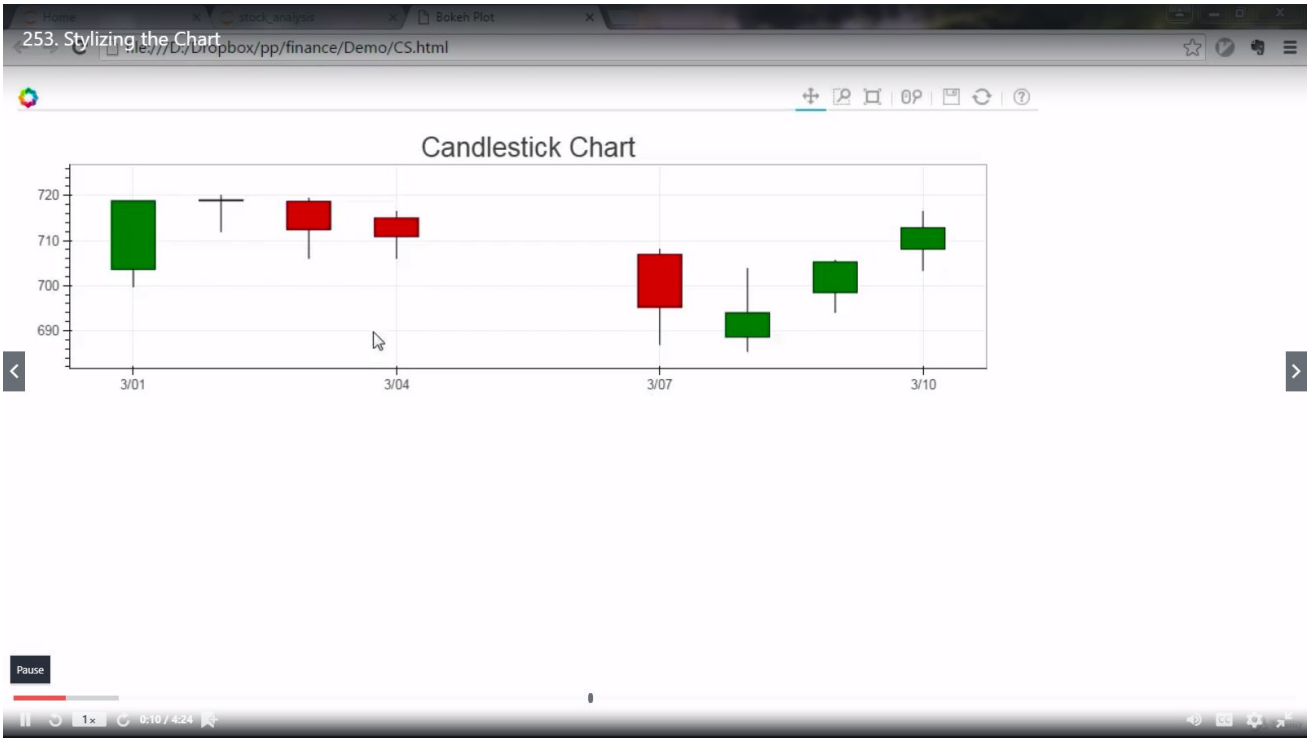
Implement 4 to 1 MUX using structural modelling style and test the module in an online/offline compiler.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux2_1 is
    port(A,B : in STD_LOGIC;
    S: in STD_LOGIC;
    Z: out STD_LOGIC);
    end mux2_1;
architecture Behavioral of mux2_1 is
begin
process (A,B,S) is
begin
 if (S ='0') then
  Z <= A;
  else
  Z <= B;
 end if;
end process;
end behavioral;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux4_1 is
port(
A,B,C,D : in STD_LOGIC;
S0,S1: in STD_LOGIC;
Z: out STD_LOGIC
);
end mux4_1;

architecture Behavioral of mux4_1 is
 component mux2_1
  port( A,B : in STD_LOGIC;
  S: in STD_LOGIC;
  Z: out STD_LOGIC);
 end component;
 signal temp1, temp2: std_logic;
 begin
 m1: mux2_1 port map(A,B,S0,temp1);
 m2: mux2_1 port map(C,D,S0,temp2);
 m3: mux2_1 port map(temp1,temp2,S1,Z);
end behavioral;
```

Date: 03 JUNE 2020 **Name:RAKSHITH B**
Course: Python On Udemy **USN:4AL16EC409**
Topic: Build a Web-based Financial Graph **Semester & Section:6 B**

AFTERNOON SESSION DETAILS

Image of session:Output



Build a Web-based Financial Graph

```
from flask import Flask, render_template

app=Flask(__name__)

@app.route('/plot/')
def plot():
    from pandas_datareader import data
    import datetime
    import fix_yahoo_finance as yf
    yf.pdr_override()
    from bokeh.plotting import figure, show, output_file
    from bokeh.embed import components
    from bokeh.resources import CDN

    start=datetime.datetime(2015,11,1)
    end=datetime.datetime(2016,3,10)

    df=data.get_data_yahoo(tickers="GOOG", start=start, end=end)


    def inc_dec(c, o):
        if c > o:
            value="Increase"
        elif c < o:
            value="Decrease"
        else:
            value="Equal"
        return value

    df["Status"]=[inc_dec(c,o) for c, o in zip(df.Close,df.Open)]
    df["Middle"]=(df.Open+df.Close)/2
    df["Height"]=abs(df.Close-df.Open)

    p=figure(x_axis_type='datetime', width=1000, height=300)
    p.title.text="Candlestick Chart"
    p.grid.grid_line_alpha=0.3

    hours_12=12*60*60*1000
```

```
    p.segment(df.index, df.High, df.index, df.Low, color="Black")

    p.rect(df.index[df.Status=="Increase"],df.Middle[df.Status=="Increase"],
           hours_12,
df.Height[df.Status=="Increase"],fill_color="#CCFFFF",line_color="black")

    p.rect(df.index[df.Status=="Decrease"],df.Middle[df.Status=="Decrease"],
           hours_12,
df.Height[df.Status=="Decrease"],fill_color="#FF3333",line_color="black")

    script1, div1 = components(p)
    cdn_js=CDN.js_files[0]
    cdn_css=CDN.css_files[0]
    return render_template("plot.html",
    script1=script1,
    div1=div1,
    cdn_css=cdn_css,
    cdn_js=cdn_js )



@app.route('/')
def home():
    return render_template("home.html")

@app.route('/about/')
def about():
    return render_template("about.html")

if __name__=="__main__":
    app.run(debug=True)
```