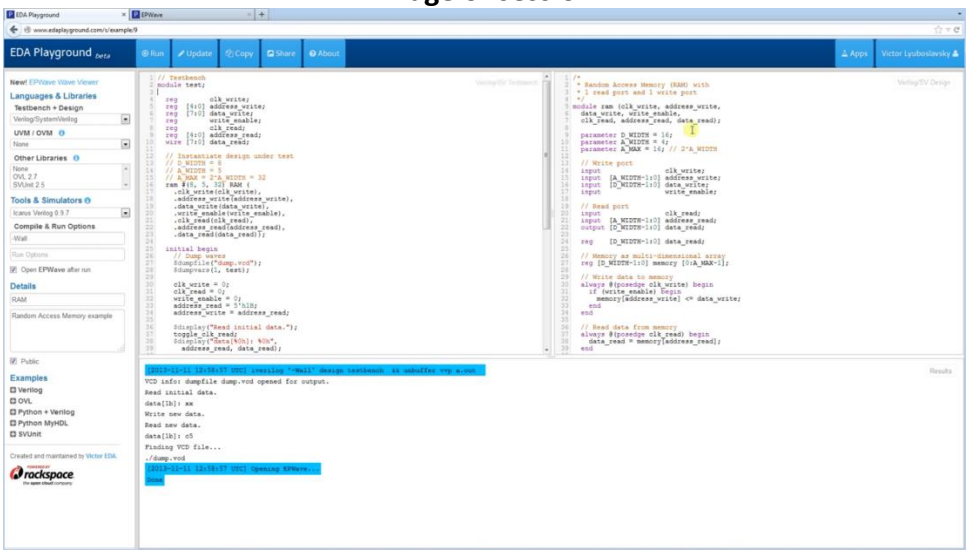
DAILY ASSESSMENT FORMAT

Date:	**3/6/2020**	**Name:**	**Kishan shetty**
Course:	**Digital Design using HDL**	**USN:**	**4AL17EC041**
Topic:	**1. EDA Playground Online complier** **2. EDA Playground Tutorial Demo Video** **3. How to Download And Install Xilinx Vivado Design Suite** **4. Vivado Design Suite for implementation of HDL code**	**Semester & Section :**	**6th-'A'**
Github Repository:	**Kishanshetty-041**	**E-mail:**	**Shettykishan983@gmail.com**

FORENOON SESSION DETAILS

Image of session



What is EDA Playground?

EDA Playground gives engineers immediate hands – on exposure to simulating system Verilog, Verilog, VHDL, C++/System C, and other HDLs. All you need is a web browser. The goal is to accelerate learning of design/test bench development with easier code sharing and simpler access to EDA tools and libraries. With a simple click, run your code and see console output in real time. View waves for your simulation using EP Wave browser-based wave viewer. Save your code snippets ("Playgrounds").Share your code and simulation results with a web link. Perfect for web forum discussions or emails. Great for asking questions or sharing your knowledge. Quickly try something out. Try out a language feature with a small example. Try out a library that you're thinking of using. Example Use cases Quick prototyping –try out syntax a library/language feature. When asking questions on Stack Overflow or other online forums, attach a link to the code and simulation results.

Task: Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux4to1 is
port(s1,s2,d00,d01,d10,d11 : in std_logic;
z_out : out std_logic);
end mux4to1;

architecture arc of mux4to1 is

component mux2to1
port(sx1,sx2,d0,d1 : in std_logic;
z : out std_logic);
end component;

component or_2
port(a,b : in std_logic;
c : out std_logic);
end component;

signal intr1, intr2, intr3, intr4 : std_logic;
begin
mux1 : mux2to1 port map(s1,s2,d00,d01,intr1);
mux2 : mux2to1 port map(not s1,s2, d10,d11,intr2);
o1 : or_2 port map(intr1, intr2, z_out);
end arc;

library ieee;
use ieee.std_logic_1164.all;
```

```
entity mux2to1 is
port(sx1,sx2,d0,d1 :in std_logic;
z1,z2: inout std_logic;
z: out std_logic);
end mux2to1;

architecture arch of mux2to1 is
begin
z1 <= d0 and (not sx1) and (not sx2);
z2 <= (d1 and (not sx1) and sx2);
z<= z1 or z2;
end arch;

entity or_2 is
port(a,b : in bit;
        c : out bit);
end or_2;
architecture arc of or_2 is
begin
c<=a or b;
end arc;
```

Testbench code:

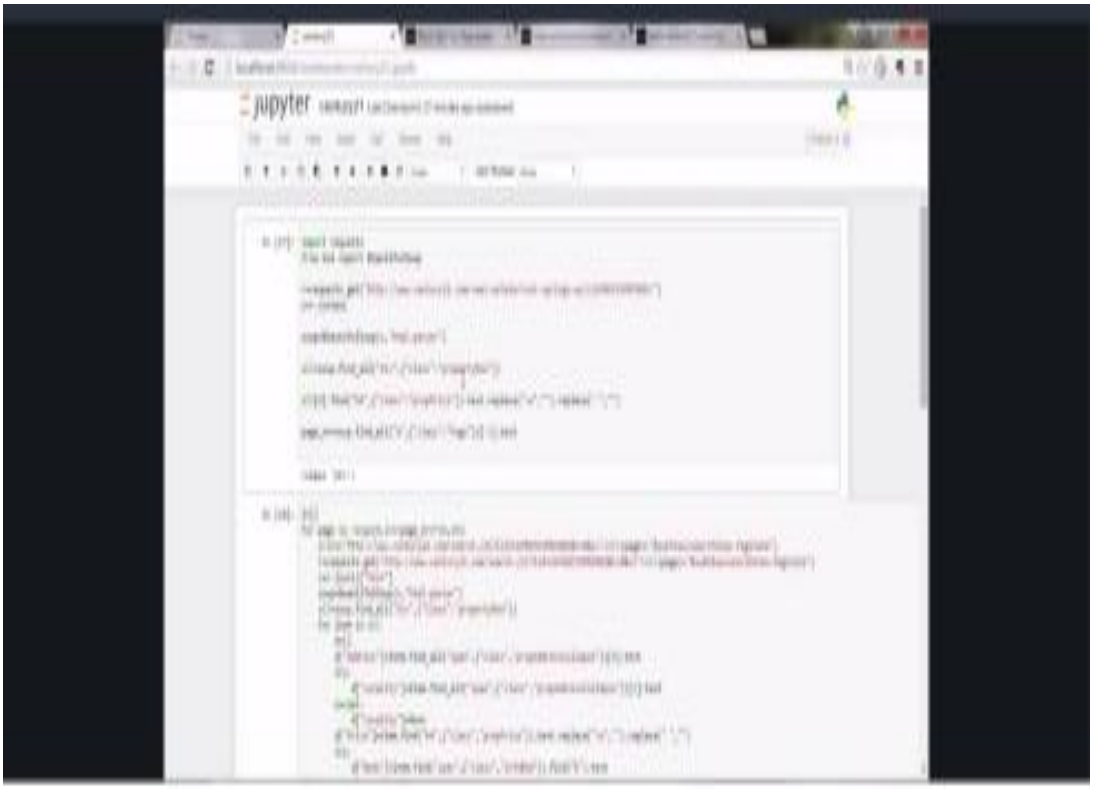
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY mux4_tb IS
END mux4_tb;

ARCHITECTURE mux4_tb OF mux4_tb IS

   COMPONENT mux4
   PORT(
       d0 : IN  bit;
       d1 : IN  bit;
       d2 : IN  bit;
       d3 : IN  bit;
       s0 : IN  bit;
       s1 : IN  bit;
       y : OUT  bit
      );
   END COMPONENT;

  signal d0 : bit := '1';
```

```
  signal d1 : bit := '0';
  signal d2 : bit := '1';
  signal d3 : bit := '0';
  signal s0 : bit := '0';
  signal s1 : bit := '0';
  signal y : bit;

BEGIN

  uut: mux4 PORT MAP (
        d0 => d0,
        d1 => d1,
        d2 => d2,
        d3 => d3,
        s0 => s0,
        s1 => s1,
        y => y
      );

 stim_proc: process
 begin

 s0 <= '0';
 s1 <= '0';
 wait for 50 ns;

 s0 <= '0';
 s1 <= '1';
 wait for 50 ns;

 s0 <= '1';
 s1 <= '0';
 wait for 50 ns;

 s0 <= '1';
 s1 <= '1';
 wait;

 end process;

END;
```

Date:	3/6/2020	Name:	Kishan shetty
Course:	python	USN:	4AL17EC041
Topic:	Application 6: Building a Mobile App with Python	Semester & Section:	6th-'A'

AFTERNOON SESSION DETAILS

Image of session



Report – Report can be typed or hand written for up to two pages.

Web scraping with Python Beautiful Soup

- Introduction to web scraping using BeautifulSoup from bs4 library and requests module.
- BeautifulSoup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.
- The requests module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data.
- **Web scraping:** Web scraping is a term used to describe the use of a program or algorithm to extract and process large amounts of data from the web. Whether you are a data scientist, engineer, or anybody who analyzes large amounts of datasets, the ability to scrape data from the web is a useful skill to have. Let's say you find data from the web, and there is no direct way to download it, web scraping using Python is a skill you can use to extract the data into a useful form that can be imported.