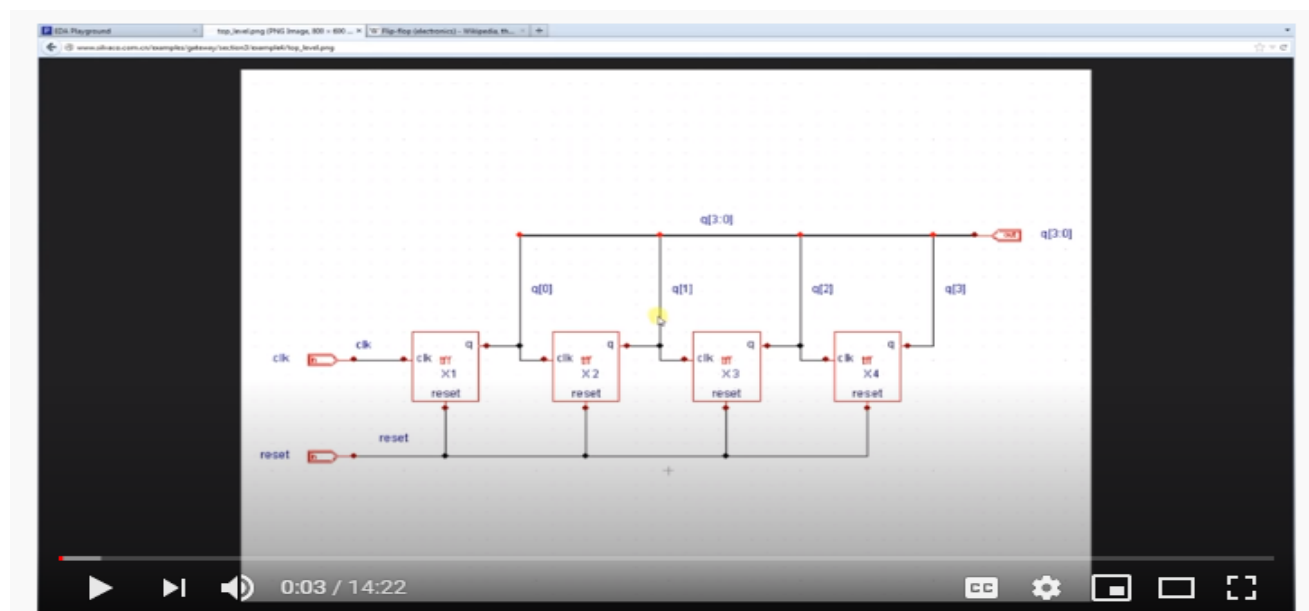
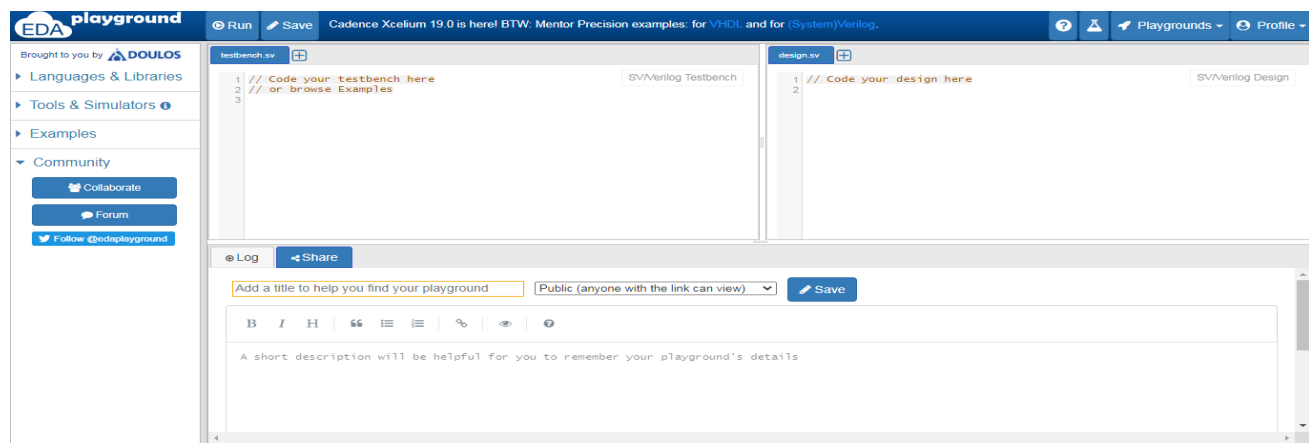


# DAILY ASSESSMENT FORMAT

Date:	3/06/2020	Name:	Akshatha M Deshpande
Course:	DIGITAL DESIGN USING HDL	USN:	4AL17EC006
Topic:	EDA Playground Tutorial Demo Video „How to Download And Install Xilinx Viva do Design Suite ,Viva do Design Suite for implementation of HDLcode	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

## FORENOON SESSION DETAILS

### Image of session



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

## Sample of EDA:

The screenshot displays the EDA Playground web interface. The top navigation bar includes the EDA Playground logo, buttons for Run, Save, and Copy, and a status message: "Cadence Xcelium 19.0 is here! BTW: Mentor Precision examples: for VHDL and for (System)Verilog." On the right of the bar are icons for help, a flask, Playgrounds, and Profile.

The left sidebar contains a "Brought to you by DOULOS" section, followed by "Languages & Libraries". Under "Testbench + Design", there is a dropdown for "SystemVerilog/Verilog", a dropdown for "UVM / OVM" set to "None", and a section for "Other Libraries" with "None", "OVL 2.8.1", and "SVUnit 2.11". Below these are checkboxes for "Enable TL-Verilog", "Enable Easier UVM", and "Enable VUnit". The "Tools & Simulators" section shows "Icarus Verilog 0.9.7" selected. Under "Compile & Run Options", there is a dropdown for "-Wall", a "Run Options" field, and checkboxes for "Open EPWave after run" and "Download files after run". There are also links for "Examples" and "Community". At the bottom of the sidebar is a "www.doulos.com/collaborate" link.

The main workspace is divided into two panels. The left panel, titled "testbench.sv", contains the following Verilog code:

```
1
2 module test;
3   initial
4     begin
5       $display("Quarantine Life!");
6     end
7 endmodule
8
```

The right panel, titled "design.sv", contains the following Verilog code:

```
1 // Code your design here
2
```

Below the code panels is a terminal window with a "Log" button and a "Share" button. The terminal output shows the command: "[2020-06-03 02:24:07 EDT] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out" and the output: "Quarantine Life!" followed by a "Done" status.

**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 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: 3/06/2020  
Course: Python  
Topic: Application 7: Scrape  
Real Estate Property  
Data from the Web

Name:  
USN:  
Semester &  
Section:

Akshatha M Deshpande  
4AL17EC006  
6th Sem A sec

## AFTERNOON SESSION DETAILS

Image of session

The screenshot shows a Udemy video player interface. The video title is "The Python Mega Course: Build 10 Real World Applications". The main content is a Jupyter Notebook window titled "century21". The notebook displays a table of real estate data with columns: Address, Area, Beds, Full Baths, Half Baths, Locality, Lot Size, and Price. The table contains 10 rows of data. Below the table, there is a code cell with the following code:

```
In [36]: df
Out[36]:
```

	Address	Area	Beds	Full Baths	Half Baths	Locality	Lot Size	Price
0	Gateway	None	None	None	None	Rock Springs, WY 82901	NaN	\$725,000
1	1003 Winchester Blvd	None	4	4	None	Rock Springs, WY 82901	0.21 Acres	\$462,900
2	3239 Spearhead Way	3,076	4	3	1	Rock Springs, WY 82901	Under 1/2 Acre	\$379,900
3	600 Takadega	3,154	5	3	None	Rock Springs, WY 82901	NaN	\$379,000
4	3487 Broad Avenue	3,236	5	3	None	Rock Springs, WY 82901	0.34 Acres	\$349,900
5	234 Via Spoleto	2,688	4	3	None	Rock Springs, WY 82901	Under 1/2 Acre	\$330,000
6	2425 Cripple Creek	8,263	4	35	None	Rock Springs, WY 82901	NaN	\$279,900
7	522 Emerald Street	1,172	3	3	None	Rock Springs, WY 82901	Under 1/2 Acre	\$254,000
8	1302 Veteran's Drive	1,932	4	2	None	Rock Springs, WY 82901	0.27 Acres	\$252,900
9	343 Via Ruzze	None	3	2	1	Rock Springs, WY 82901	0.16 Acres	\$219,900

Below the table, there is a code cell with the following code:

```
In [37]: df.to_csv("Output.csv")
In [ ]: base_url="http://www.cen
for page in range(0,10)
```

A text overlay on the video says: "So at each iteration we will increase by 10. For now don't worry about the". The video player shows a progress bar at 4:45 / 17:15.

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

### Application 7: Scrape Real Estate Property Data from the Web:

- ◆ Scraped Website Data - How The Output Will Look Like
- ◆ Loading the Web page in Python
- ◆ Extracting "div" Tags
- ◆ Extracting Addresses and Property Details
- ◆ Extracting Elements without Unique Identifiers
- ◆ Saving the Extracted Data in CSV Files
- ◆ Crawling through web pages
- ◆ use web scraping to collect email ID
- ◆ Web scraping is used to collect a large set of data
- ◆ Email address gathering
- ◆ Web scraping is used to collect data from Social Media websites
- ◆ Services such as ParseHub use web scraping to collect data from online shopping
- ◆ Details regarding job openings, interviews are collected from different websites

