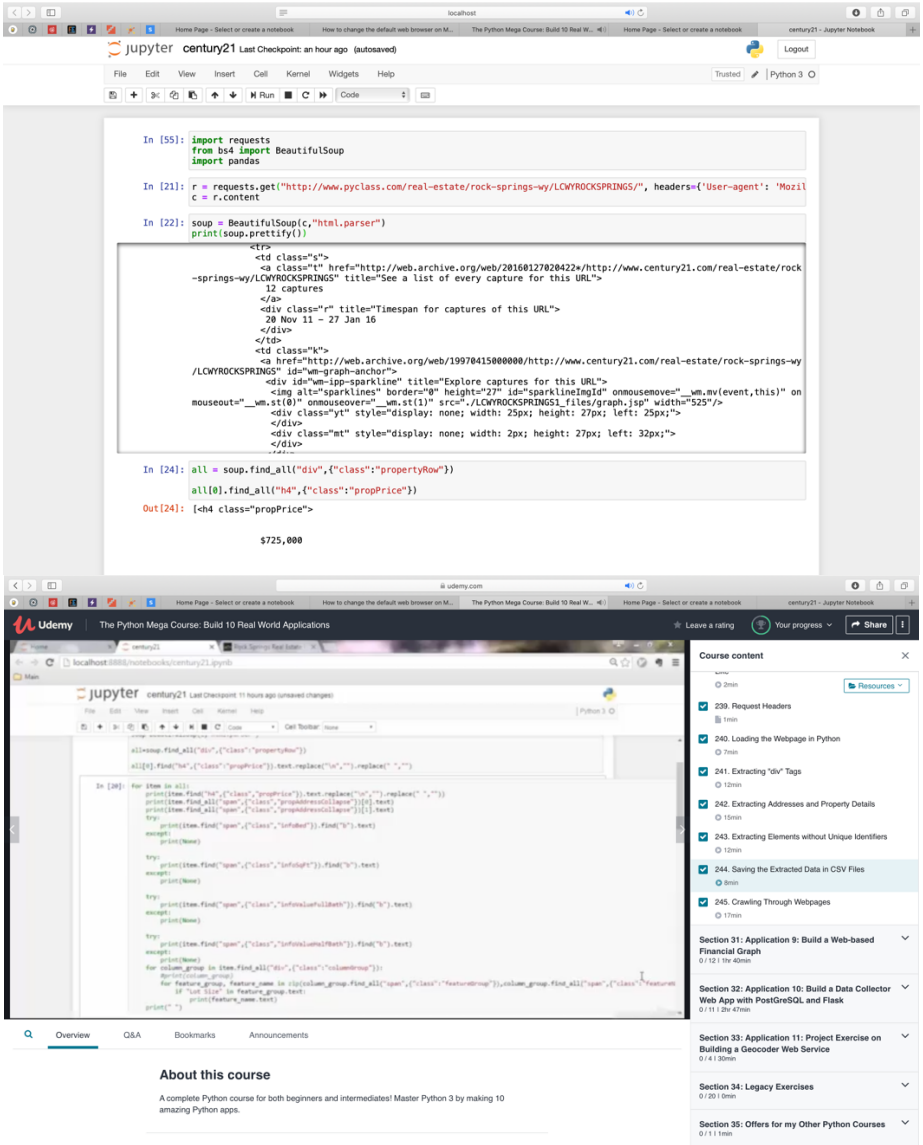
DAILY ASSESSMENT FORMAT

Date:	**03/06/2020**	**Name:**	**Prajwal Kamagethi Chakravarti P L**
Course:	**Python**	**USN:**	**4AL17EC073**
Topic:	• **Application 7: Scrape Real Estate Property Data from the Web**	**Semester & Section:**	**6 & B**
Github Repository:	**https://github.com/alvas-education-foundation/Prajwal-Kamagethi.git**		

FORENOON SESSION DETAILS

Image of the session:



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

Web scraping is used to collect large information from websites. But why does someone have to collect such large data from websites? To know about this, let's look at the applications of web scraping:

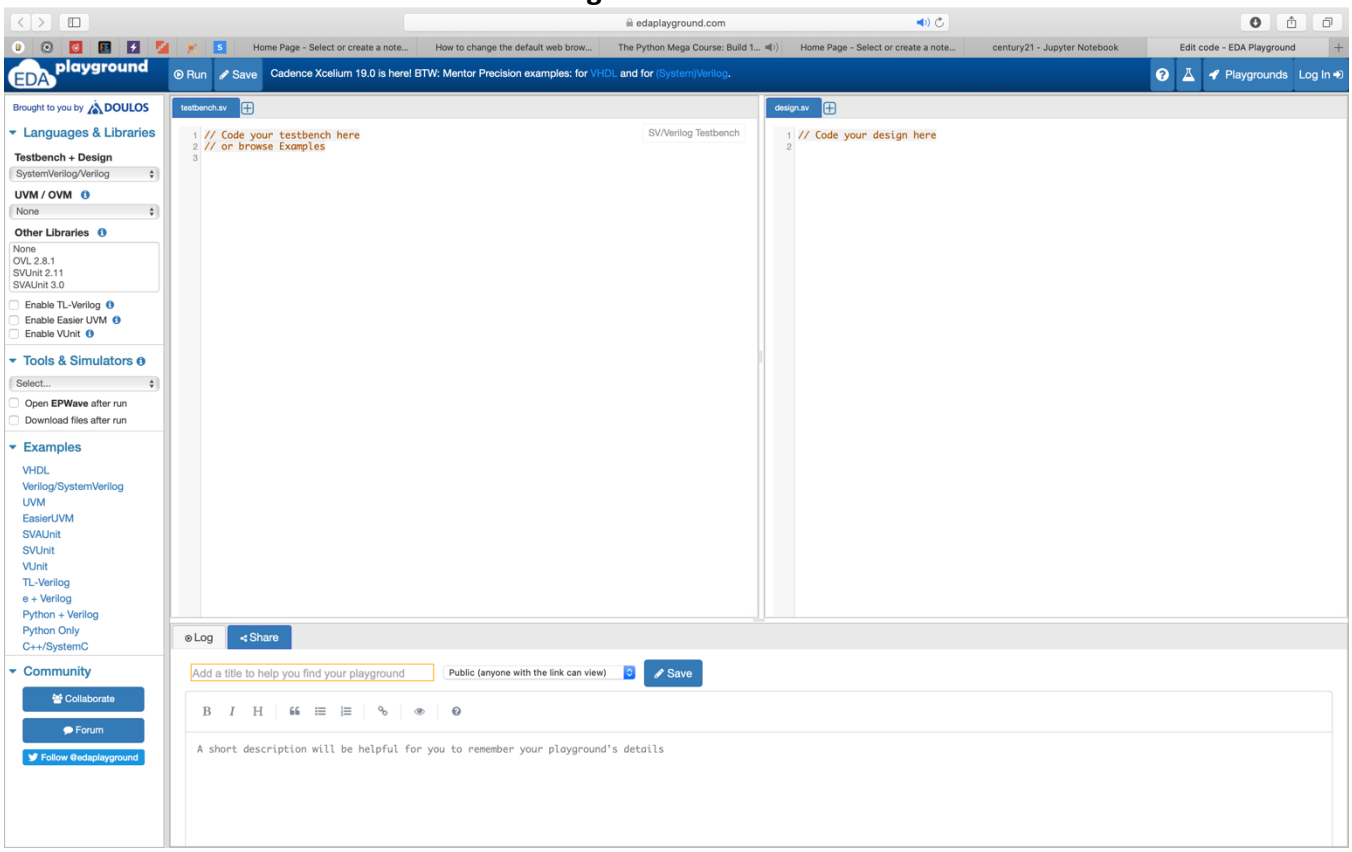
- **Price Comparison: Services such as ParseHub use web scraping to collect data from online shopping websites and use it to compare the prices of products.**
- **Email address gathering: Many companies that use email as a medium for marketing, use web scraping to collect email ID and then send bulk emails.**
- **Social Media Scraping: Web scraping is used to collect data from Social Media websites such as Twitter to find out what's trending.**
- **Research and Development: Web scraping is used to collect a large set of data (Statistics, General Information, Temperature, etc.) from websites, which are analyzed and used to carry out Surveys or for R&D.**
- **Job listings: Details regarding job openings, interviews are collected from different websites and then listed in one place so that it is easily accessible to the user.**
- **In this section we were taught how to scrap data from a real estate website.**

Why Python for Web Scraping?

- **Ease of Use: Python is simple to code. You do not have to add semi-colons ";" or curly-braces "{}" anywhere. This makes it less messy and easy to use.**
- **Large Collection of Libraries: Python has a huge collection of libraries such as Numpy, Matlplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.**
- **Dynamically typed: In Python, you don't have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.**
- **Easily Understandable Syntax: Python syntax is easily understandable mainly because reading a Python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in Python also helps the user to differentiate between different scope/blocks in the code.**
- **Small code, large task: Web scraping is used to save time. But what's the use if you spend more time writing the code? Well, you don't have to. In Python, you can write small codes to do large tasks. Hence, you save time even while writing the code.**
- **Community: What if you get stuck while writing the code? You don't have to worry. Python community has one of the biggest and most active communities, where you can seek help from.**

Date:	**03-06-2020**	**Name:**	**Prajwal Kamagethi Chakravarti P L**
Course:	**DIGITAL DESIGN USING HDL**	**USN:**	**4AL17EC073**
Topic:	• **EDA Playground Online complier** • **EDA Playground Tutorial Demo Video** • **How to Download And Install Xilinx Vivado Design Suite** • **Vivado Design Suite for implementation of HDL code**	**Semester & Section:**	**6TH & B**
Github Repository:	**https://github.com/alvas-education-foundation/Prajwal-Kamagethi.git**		

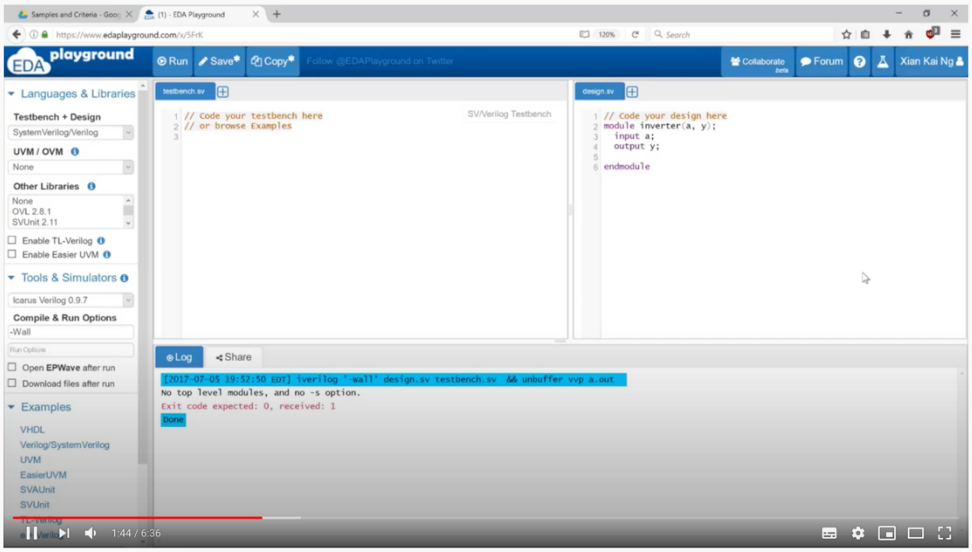
Image of the session:



youtube.com

Home Page - Select or create... | How to change the default we... | The Python Mega Course:... | Home Page - Select or create... | century21 - Jupyter Notebook | Edit code - EDA Playground | (6) Verilog on EDA Playgro...

YouTube IN — Search



Verilog on EDA Playground Starting Tutorial

6,442 views • 5 Jul 2017

50 | 1 | SHARE | SAVE

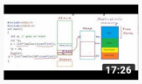
King X KoK
4.03K subscribers

SUBSCRIBE

A Getting Started Guide
Remember to rate & comment!

Up next — AUTOPLAY


Aalto Talk with Linus Torvalds [Full-length]
aaltouniversityace
1.6M views • 7 years ago


Pointers and dynamic memory - stack vs heap
mycodeschool
817K views • 7 years ago

OSI Model Explained | OSI Animation | Open System...
TechTerms
1.6M views • 1 year ago

Support Vector Machine (SVM) - Fun and Easy Machine...
Augmented Startups
368K views • 2 years ago

What is HART Protocol?
RealPars
552K views • 1 year ago


4.4 Bellman Ford Algorithm - Single Source Shortest Path -...
Abdul Bari
527K views • 2 years ago

Natural Language Processing in Python
PyOhio
510K views • 1 year ago

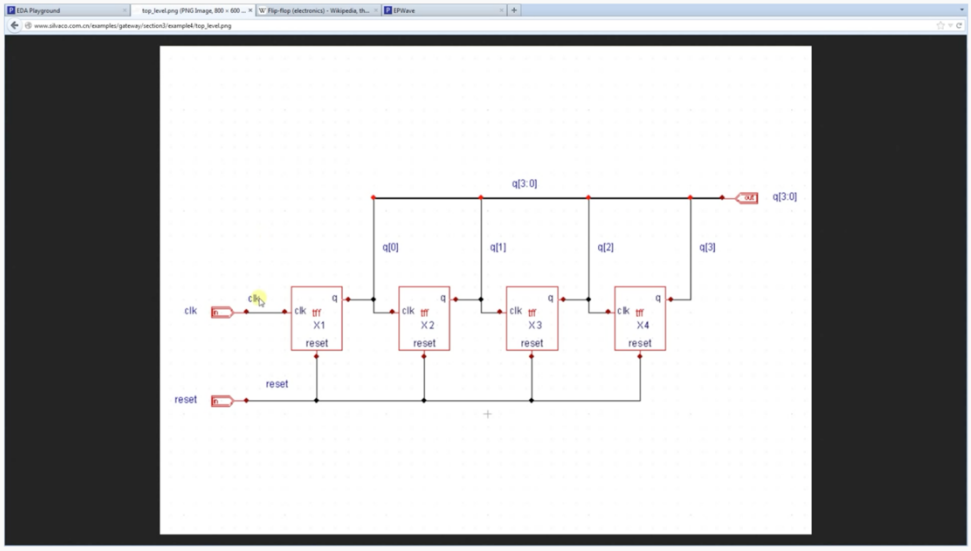
Live: TinyPICO Nano - Test Jig Adapter PCB - Frogo pins!
Unexpected Maker
129 watching
LIVE NOW


Introduction to Microservices, Docker, and Kubernetes

youtube.com

Home Page - Select or cr... | How to change the defaul... | The Python Mega Cou... | Home Page - Select or cr... | century21 - Jupyter Note... | Edit code - EDA Playgrou... | on EDA Playground St... | Tutorial 1 -- Ripple Ca...

YouTube IN — Search



Verilog Tutorial 1 -- Ripple Carry Counter

58,978 views • 11 Nov 2013

288 | 3 | SHARE | SAVE

EDA Playground
5.12K subscribers

SUBSCRIBE

In this Verilog tutorial, we implement a basic Ripple Carry Counter design and test using Verilog.

Complete Ripple Carry Counter from the Verilog tutorial:

Up next — AUTOPLAY


Verilog Tutorial 2 -- $display System Task
EDA Playground
14K views • 6 years ago


Verilog Basics
Paul Franzon
158K views • 7 years ago


Learning Verilog for FPGAs: Flip Flops
HACKADAY
3.6K views • 4 years ago


Hello UVM
SK B
24K views • 6 years ago

Joji
88rising

Verilog tutorial for beginners 2 : D Flip Flop Implementation in...
Rajput Sandeep
58K views • 5 years ago

PixHawk/Mission Planner/ArduPlane Build for...
Painless360
Recommended for you

Live: TinyPICO Nano - Test Jig Adapter PCB - Frogo pins!
Unexpected Maker
129 watching
LIVE NOW


Access Any Android Device Remotely Without Touching...

Report:

- **In today's session we noted how to use a EDA Playground Online complier**
- **EDA Playground Tutorial Demo Video helped us in getting familiarized with the tool.**
- **How to Download And Install Xilinx Vivado Design Suite and Vivado Design Suite for implementation of HDL code video taught us how to download and use the Xilinx to implement Verilog code.**

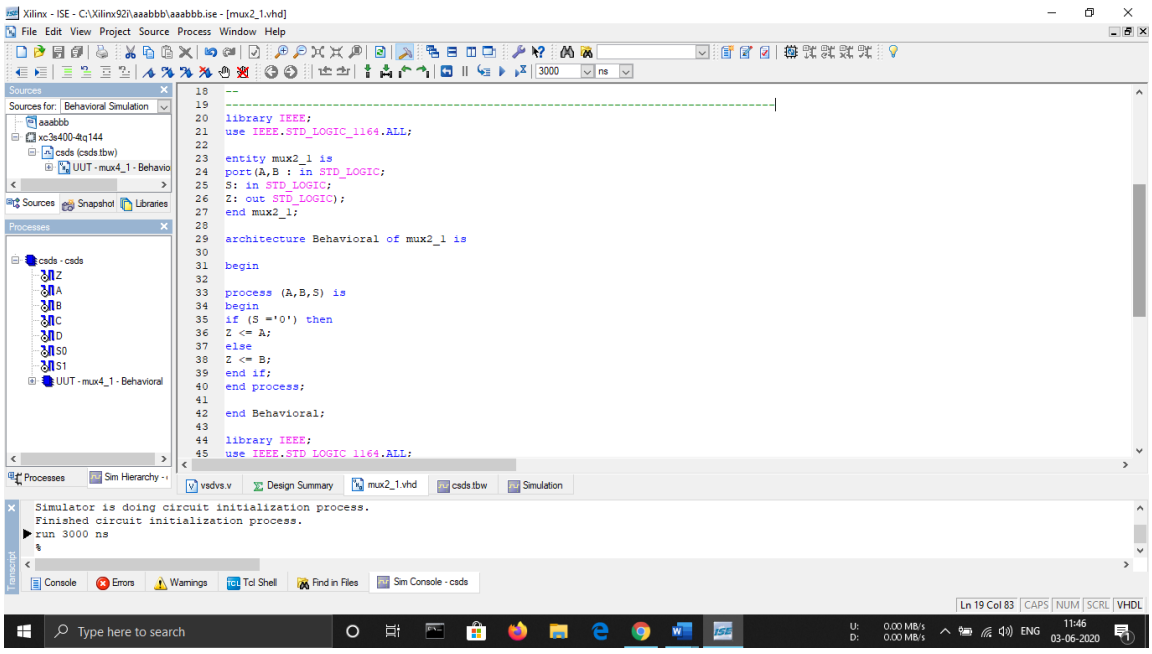
Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler.

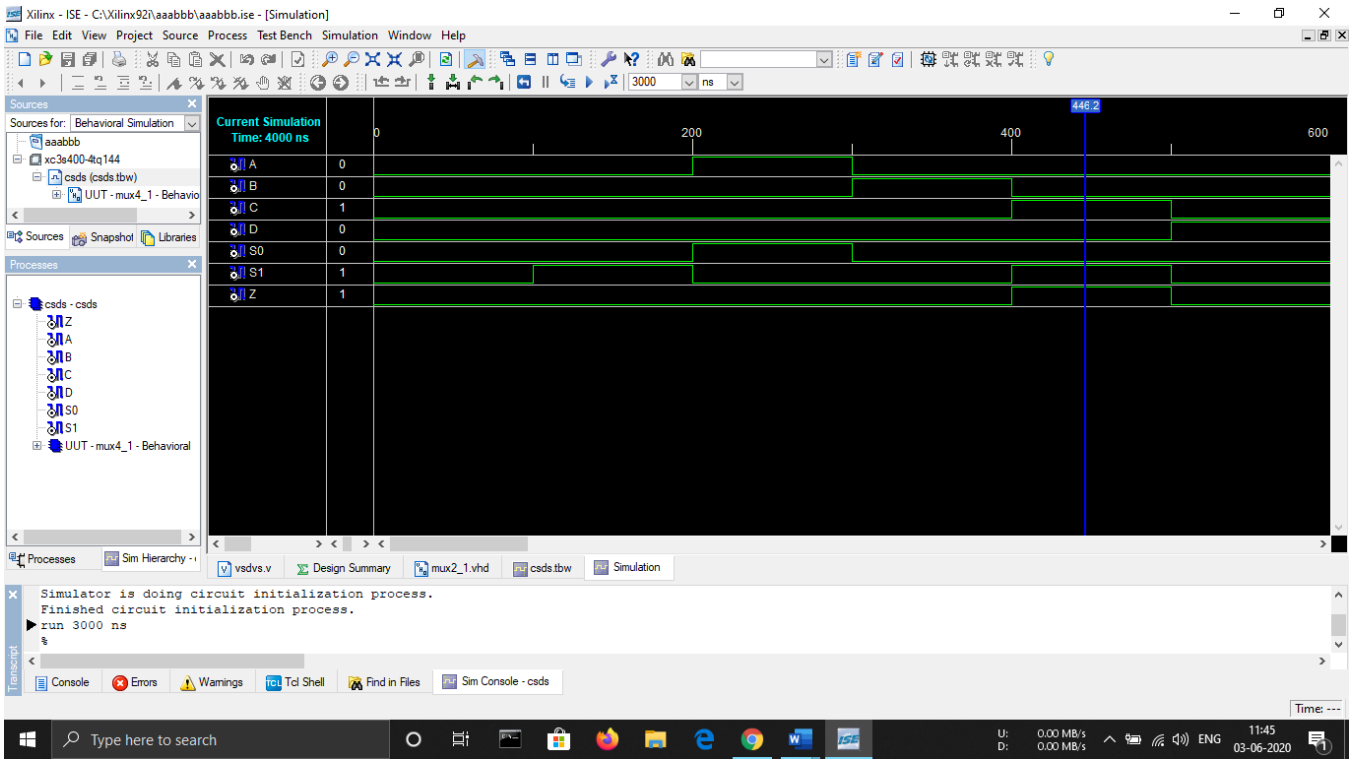
Code:

```
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;
```

Output:




Xilinx - ISE - C:\Xilinx92i\aaabbb\aaabbb.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
Sources
Sources for: Behavioral Simulation
aaabbb
xc3s400-4tq144
csds (csds.tbw)
UUT - mux4_1 - Behavio
Sources Snapshot Libraries
Processes
csds - csds
Z
A
B
C
D
S0
S1
UUT - mux4_1 - Behavioral
Processes Sim Hierarchy
Current Simulation Time: 4000 ns
0
200
400
600
446.2
A 0
B 0
C 1
D 0
S0 0
S1 1
Z 1
vsdvs.v Design Summary mux2_1.vhd csds.tbw Simulation
Simulator is doing circuit initialization process.
Finished circuit initialization process.
run 3000 ns
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - csds
Time: ---
Type here to search
U: 0.00 MB/s
D: 0.00 MB/s
ENG
11:45
03-06-2020