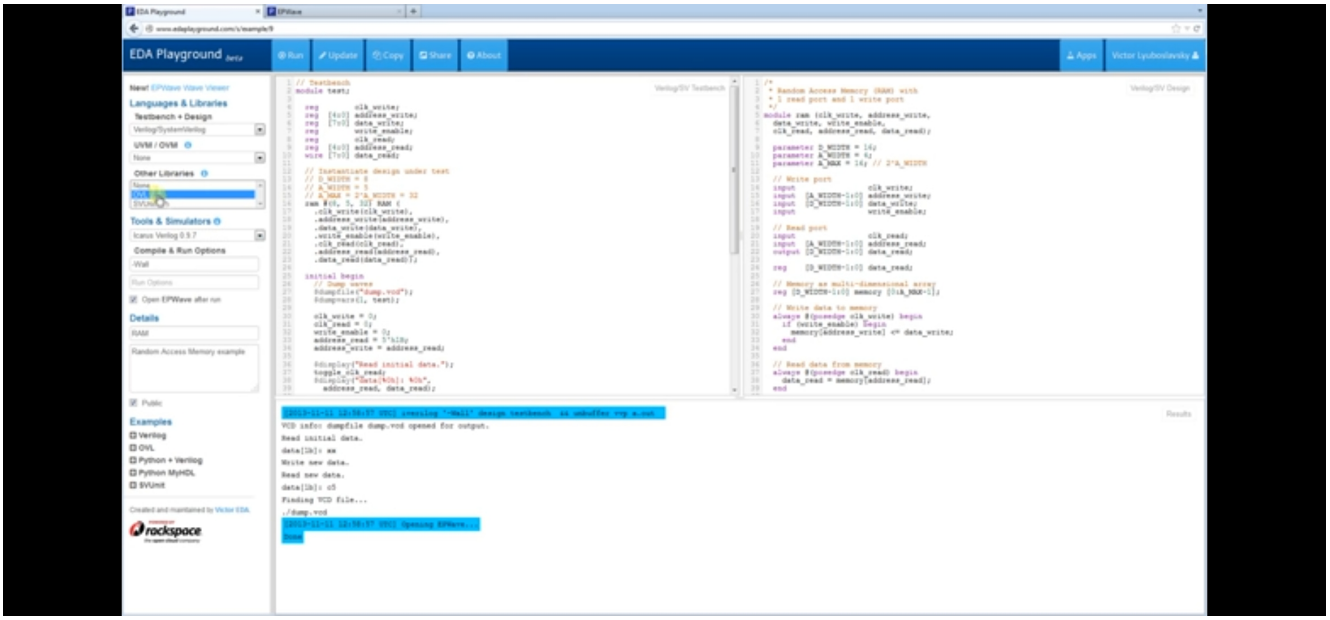
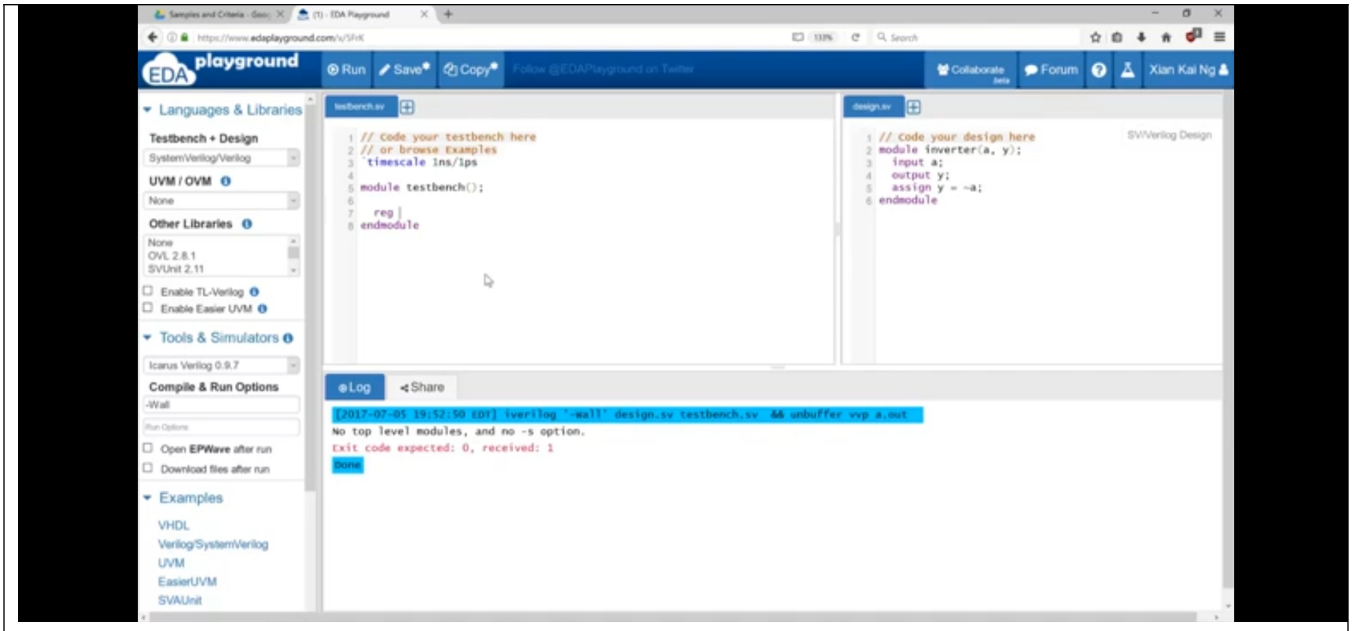
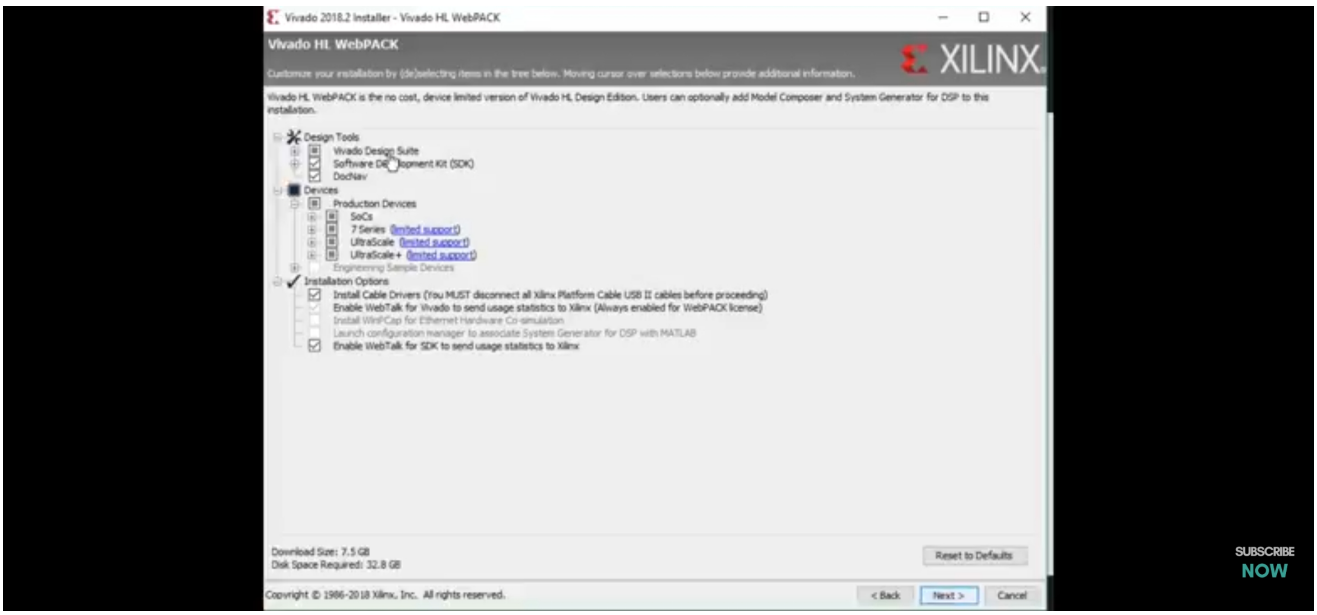
DAILY ASSESSMENT FORMAT

Date:	03-06-2020	Name:	Jagadeesha Hegde
Course:	Logic Design	USN:	4AL17EC036
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 A-sec
Github Repository:	Jagadeesha-036		

FORENOON SESSION DETAILS
Image of session




EDA playground
Run
Save
Copy
Collaborate
Forum
Xian Kai Ng
Languages & Libraries
Testbench + Design
System Verilog/Verilog
UVM / OVM
None
Other Libraries
None
OVL 2.8.1
SVUnit 2.11
Enable TL-Verilog
Enable Easier UVM
Tools & Simulators
Icarus Verilog 0.9.7
Compile & Run Options
-Wall
Open EPWave after run
Download files after run
Examples
VHDL
Verilog/SystemVerilog
UVM
EasierUVM
SVAUnit
testbench.sv
// Code your testbench here
// or browse Examples
`timescale 1ns/1ps
module testbench();
reg
endmodule
design.sv
// Code your design here
module inverter(a, y);
input a;
output y;
assign y = ~a;
endmodule
SV/Verilog Design
Log
Share
[2017-07-05 19:52:50 EDT] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out
No top level modules, and no -s option.
Exit code expected: 0, received: 1
Done


Vivado 2018.2 Installer - Vivado HL WebPACK
Vivado HL WebPACK
XILINX
Customize your installation by (de)selecting items in the tree below. Moving cursor over selections below provide additional information.
Vivado HL WebPACK is the no cost, device limited version of Vivado HL Design Edition. Users can optionally add Model Composer and System Generator for DSP to this installation.
Design Tools
Vivado Design Suite
Software Development Kit (SDK)
DocNav
Devices
Production Devices
SoCs
7 Series (limited support)
UltraScale (limited support)
UltraScale+ (limited support)
Engineering Sample Devices
Installation Options
Install Cable Drivers (You MUST disconnect all Xilinx Platform Cable USB II cables before proceeding)
Enable WebTalk for Vivado to send usage statistics to Xilinx (Always enabled for WebPACK license)
Install WinPCap for Ethernet Hardware Co-simulation
Launch configuration manager to associate System Generator for DSP with MATLAB
Enable WebTalk for SDK to send usage statistics to Xilinx
Download Size: 7.5 GB
Disk Space Required: 32.8 GB
Reset to Defaults
Copyright © 1986-2018 Xilinx, Inc. All rights reserved.
< Back
Next >
Cancel
SUBSCRIBE
NOW

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

EDA Playground gives engineers immediate hands-on exposure to simulating SystemVerilog, Verilog, VHDL, C++/SystemC, and other HDLs. All you need is a web browser. The goal is to accelerate learning of design/testbench 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 EPWave 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 outTry out a language feature with a small example.Try out a library that you’re thinking of using. Example UsecasesQuick prototyping – try out syntax or a library/language feature.When asking questions on Stack Overflow or other online forums, attach a link to the code and simulation results.Use during technical interviews to test candidates’ SystemVerilog/Verilog coding and debug skills.Try verifying using different verification frameworks: UVM, SVUnit, plain Verilog, or Python.Tools & Simulators For settings and options documentation, see Tools & Simulators OptionsAvailable tools and simulators are below. EDA Playground can support many different tools. Contact us to add your EDA tool to EDA Playground

Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style:

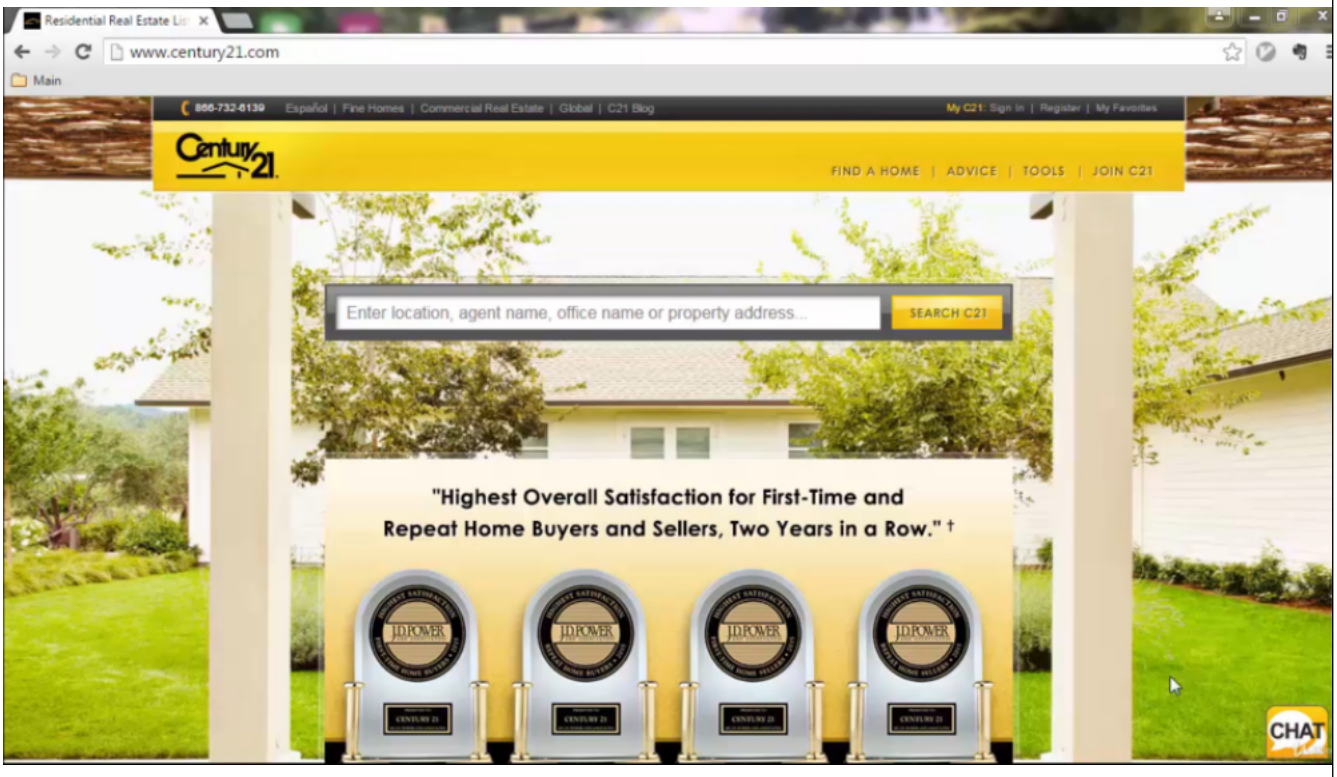
```
module and_gate( output a,input b,c);

assign a=b&c;

endmodule


module not_gate(output d ,input e) ;

assign d= ~e;

endmodule


module or_gate(output l, input m,n);

assign l=m | n;

endmodule


module m21(Y,D0,D1,S);
```

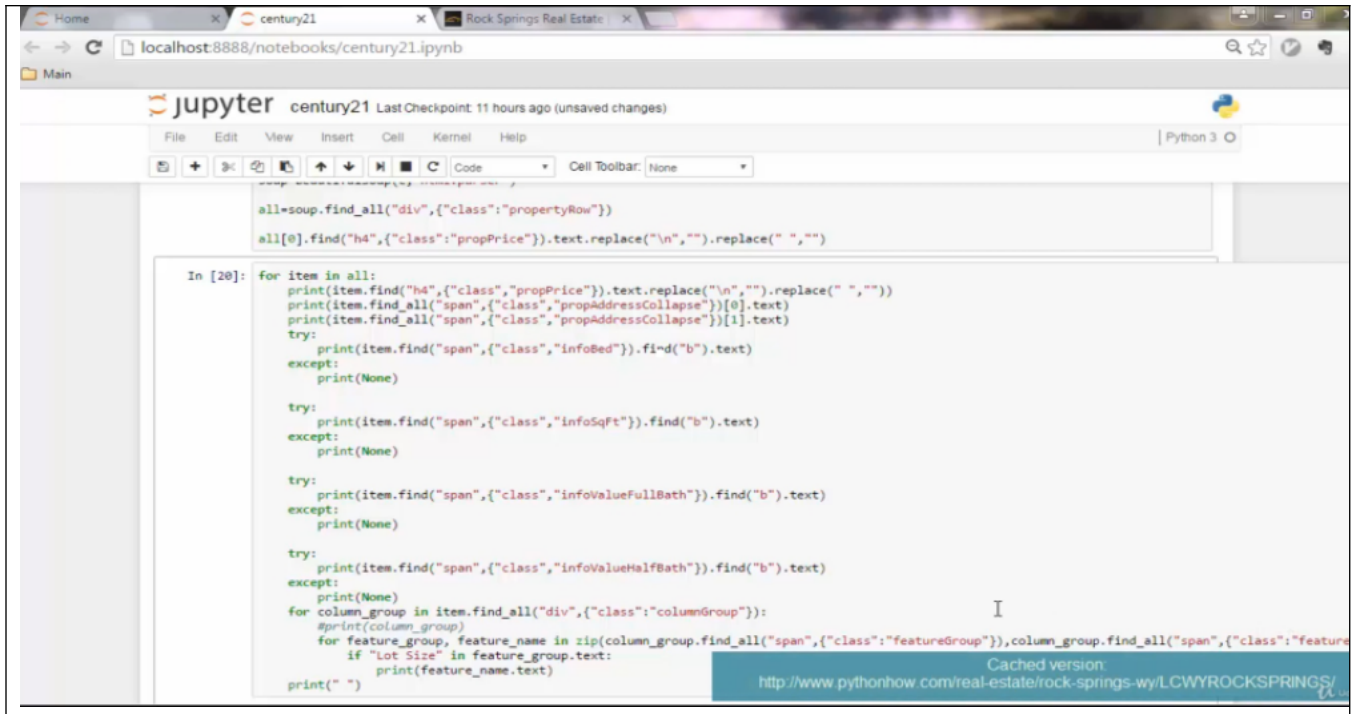
```
output Y;
input D0,D1,S;
wire T1,T2,T3;
and_gate  u1(T1,D1,S);
not_gate  u2(T2,S);
and_gate  u3(T3,D0,T2);
or_gate   u4(Y,T1,T3);
endmodule
```

Date:	03-06-2020	Name:	Jagadeesha Hegde
Course:	The Python Mega Course	USN:	4AL17EC036
Topic:	Application 8: Scrape Real Estate Property Data from the Web	Semester & Section:	6th A-sec

AFTERNOON SESSION DETAILS

Image of session




Home
century21
Rock Springs Real Estate
localhost:8888/notebooks/century21.ipynb
Main
jupyter century21 Last Checkpoint: 11 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Python 3
Code
Cell Toolbar: None
all=soup.find_all("div",{"class":"propertyRow"})
all[0].find("h4",{"class":"propPrice"}).text.replace("\n","").replace(" ","")
In [20]:
for item in all:
print(item.find("h4",{"class","propPrice"}).text.replace("\n","").replace(" ",""))
print(item.find_all("span",{"class","propAddressCollapse"})[0].text)
print(item.find_all("span",{"class","propAddressCollapse"})[1].text)
try:
print(item.find("span",{"class","infoBed"}).fi~d("b").text)
except:
print(None)
try:
print(item.find("span",{"class","infoSqFt"}).find("b").text)
except:
print(None)
try:
print(item.find("span",{"class","infoValueFullBath"}).find("b").text)
except:
print(None)
try:
print(item.find("span",{"class","infoValueHalfBath"}).find("b").text)
except:
print(None)
for column_group in item.find_all("div",{"class":"columnGroup"}):
#print(column_group)
for feature_group, feature_name in zip(column_group.find_all("span",{"class":"featureGroup"}),column_group.find_all("span",{"class":"feature
if "Lot Size" in feature_group.text:
print(feature_name.text)
print(" ")
Cached version:
http://www.pythonhow.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/

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

Scrape real estate property data from the web :

There was a time when real estate dealings were discrete, paper-based operations done on a one to one basis. With the rise of the internet and every industry finding its way into it, real estate began to realize its true potential on the web. There is no denying the fact that the internet is the most useful tool at a seller's disposal.With a large number of potential buyers online, realtors find the internet an excellent source to advertise property listings, hereby automating the whole process. Statistics suggest that 40% of buyer's inquiries stem from internet advertisements and nine out of ten people use the internet to search for property. Moreover, the same property can be enlisted on numerous sites to increase traffic and the corresponding chance of a sale.This implies endless opportunities for a realtor. But harnessing relevant data out of big data to a non-technical realtor is like looking for a needle in a haystack. The web has a staggering amount of information leading to a plethora of choices and comparisons can lead to significant confusion, making it difficult to fathom and make sense of.Web Scraping in real estate to the rescueWeb scraping is the process of sorting through overwhelming amounts of data, refine the user's searches and provide a list of relevant information. In a realtor's case, it is the go-to tool for organized property listings. Scraping the web provides parameters which the realtor can further study to determine sales and prospective buyers