**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 3 June 2020 | **Name:** | Anupama J S |
| **Course:** | Digital design using HDL | **USN:** | 4AL16EC005 |
| **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:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **EDA Playground Help**  Quick Start  Log in. Click the Log in button (top right) Then either  click on Google or Facebook or  register by clicking on ‘Register for a full account’ (which enables all the simulators on EDA Playground)  Select your language from the Testbench + Design menu.  Select your simulator from the Tools & Simulators menu. Using certain simulators will require you to supply additional identifcation information.  Type in your code in the testbench and design windows.  Click Run.  Tutorial <http://eda-playground.readthedocs.io/en/latest/tutorial.html>  EDA Playground on YouTube - Tutorials for Verilog, SystemVerilog, UVM, and VHDL, interview questions, news and features, etc.  **What is EDA Playground?**  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 out  Try out a language feature with a small example.  Try out a library that you’re thinking of using.  EDA Playground In a separate web browser window, log in to EDA Playground at: http://www.edaplayground.com  Log in. Click the Log in button (top right) Then either click on Google or Facebook or register by clicking on ‘Register for a full account’ (which enables all the simulators on EDA Playground)  Select ‘Aldec Riviera Pro’ from the Tools & Simulators menu. This selects the Aldec Riviera Pro simulator, which can be used however you logged in. Using certain other simulators will require you to have registered for a full account.  In either the Design or Testbench window pane, type in the following code:    module test;  initial  $display("Hello World!");  endmodule  Click Run (top left)  Yes, running a simulation is this  In the bottom pane, we see real-time results as our code is being compiled and then run. A run typically takes 1-5 seconds, depending on network traffic and simulator. Near the bottom of result output, we see:    Hello World!  Now, Click the Share tab near in the bottom pane and then type in a name and description. Then click Save.    The browser page will reload and the browser address bar will change. This is a persistent link to saved code. We can send the link by email, post it on a web page, post it on Stack Overflow forums, etc. Here is what the link looks like for one user’s Hello World! playground: http://www.edaplayground.com/s/3/12    Now, let’s try modifying existing code. Load the following example: RAM    On the left editor pane, before the end of initial block, add the following:    write\_enable = 1;  data\_write = 8'h2C;  toggle\_clk\_write;  toggle\_clk\_read;  $display ("data[%0h]: %0h",address\_read, data\_read);    Run the sim. In the results you should see this new message:  data[1b]: 2c  Click Copy to save a personal version of the modified RAM code, including the simulation results.  Loading Waves from EDA Playground  You can run a simulation on EDA Playground and load the resulting waves in EPWave.  Loading Waves for System Verilog and Verilog Simulations   Go to your code on EDA Playground. For example: RAM Design and Test   Make sure your code contains appropriate function calls to create a \*.vcd file. For example:  initial begin  $dumpfile("dump.vcd");  $dumpvars(1);  End   Select a simulator and check the Open EPWave after run checkbox. (Not all simulators may have this run option.)   Click Run. After the run completes, the resulting waves will load in a new EPWave window. (Pop-ups must be enabled.)  Loading Waves for VHDL Simulation   Check the Open EPWave after run checkbox.   Specify the Top entity to simulate.   Click Run. After the run completes, the resulting waves will load in a new EPWave window. (Pop-ups must be enabled.)  The waves for all signals in the specified Top entity and any of its components will be dumped.  In EPWave window, click Get Signals to select the signals to view.    **How to Download And Install Xilinx Vivado Design Suite**  Vivado Design Suite - HLx Editions Update 1 - 2019.2  Important Information  This is a common updater. You do not need to re-run it for Vitis if you have already run it for Vivado and vice versa.  Vivado Design Suite 2019.2.1 is now available with support for:  Additional Zynq UltraScale+ RFSoCs devices enabled:- (XCZU46DR, XCZU47DR, XCZU48DR, XCZU49DR)  For customers using these devices, Xilinx recommends installing Vivado 2019.2.1. For other devices, please continue to use Vivado 2019.2.  Note: Download verification is only supported with Google Chrome and Microsoft Internet Explorer web browsers.    The Vivado Design Suite supports the following established industry design standards:  • Tcl  • AXI4, IP-XACT  • Synopsys design constraints (SDC)  • Verilog, VHDL, VHDL-2008, SystemVerilog  • SystemC, C, C++  The Vivado Design Suite solution is native Tcl based with support for SDC and Xilinx design constraints (XDC) formats. Extensive Verilog, VHDL, and SystemVerilog support for synthesis enables easier FPGA adoption. Vivado High-Level Synthesis (HLS) enables the use of native C, C++, or SystemC languages to define logic. Using standard IP interconnect protocol, such as AXI4 and IP-XACT, enables faster and easier system-level design integration. Support for these industry standards also enables the electronic design automation (EDA) ecosystem to better support the Vivado Design Suite. In addition, many new third-party tools are integrated with the Vivado Design Suite.  **Task 3 - Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler.**    module mux4to1(a,sel,out);  input [3:0] a;  input [1:0] sel;  output out;  wire mux[2:0];  mux2to1 m1 (a[3],a[2],sel[0],mux\_1),  m2 (a[1],a[4],sel[0],mux\_2),  m3 (mux\_1,mux\_2,sel[1],out);  endmodule. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 3 June 2020 | **Name:** | Anupama J S |
| **Course:** | Python | **USN:** | 4AL16EC005 |
| **Topic:** | 1. Application 8: Scrape Real Estate Property Data from the Web | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| **Why scrape data from real estate sites?**  The large property listing companies target an entire nation and work on millions of properties. But in case you are a real estate agent, or if you are setting up shop and targeting a specific state or region, it is better that instead of trying to gather data yourself, you scrape it from a major real estate listing website.  You can also build Machine Learning models to predict the prices of properties and compare your predictions with Zillow™’s Zestimates™ and see which one is better or closer to real values.  **How to set things up?**  In case you have followed any of our previous “How to scrape” articles, you might already have the necessary setup ready on your computers. In case you have not, I recommend you to follow this [article](https://www.promptcloud.com/blog/tutorial-how-to-scrape-amazon-product-details-prices-using-python/) to set up Python, its packages and the text editor before you can get your hands dirty with the code.  **Where is the code?**  Without much ado, we decided to bring you the code for the scraper that will help you extract information from a property listing. It is written in Python and subsequently, I will show you how to run it and what you will get once you run it.  **Web Scraping Using Beautiful Soup**  Beautiful Soup is a Python library designed for quick turnaround projects like screen scraping. Three features make it powerful:  Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn’t take much code to write an application. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn’t take muchcode to write an application | | | |