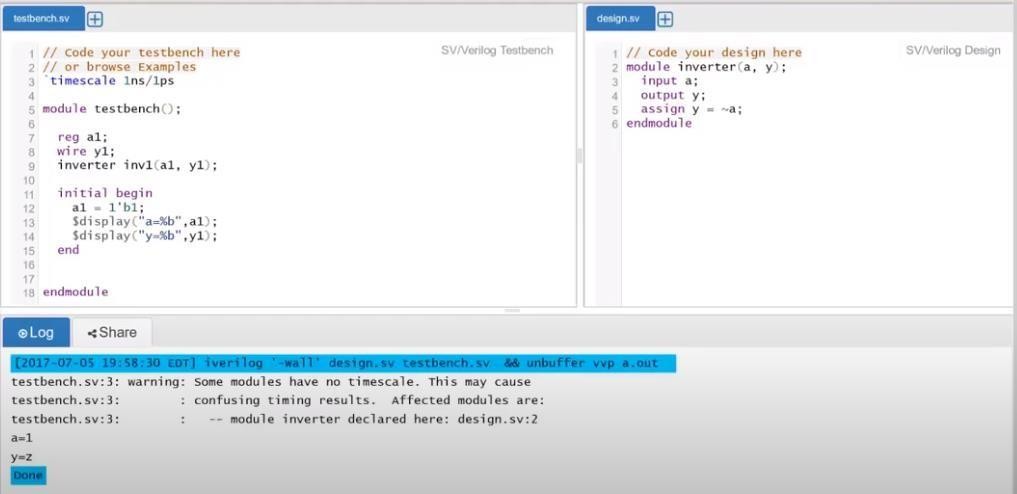
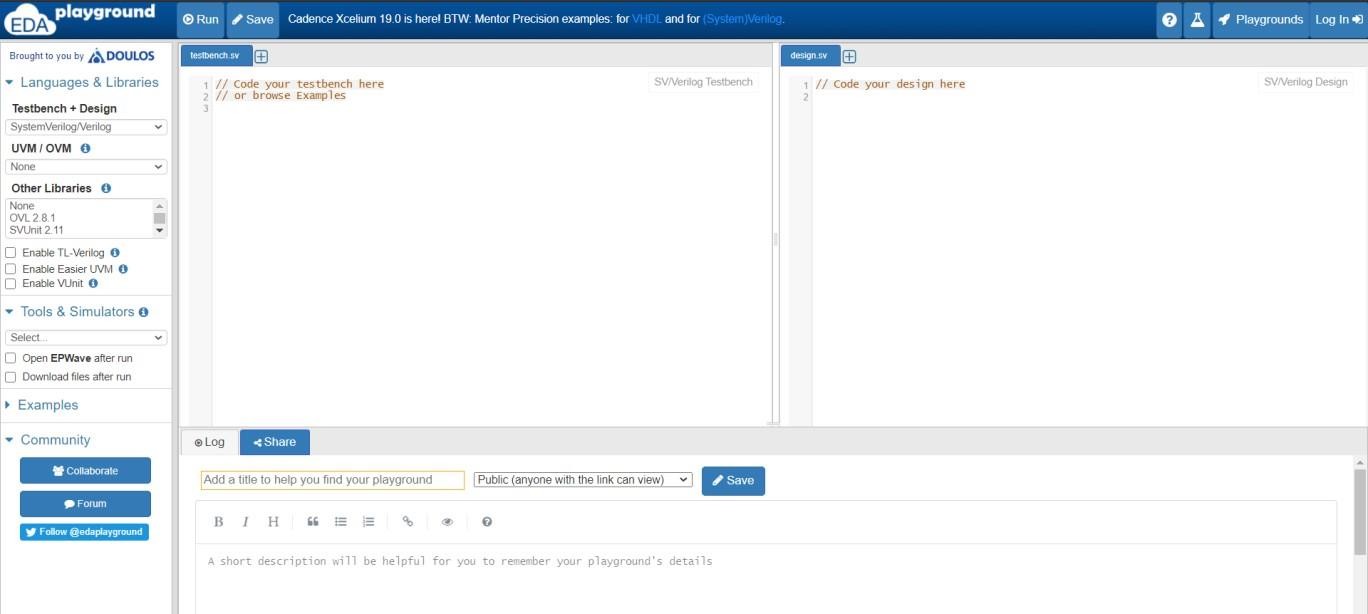
DAILY ASSESSMENT REPORT

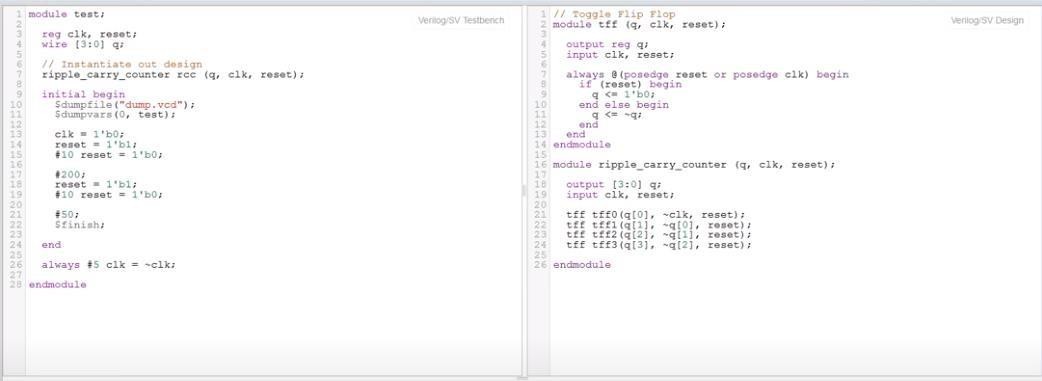
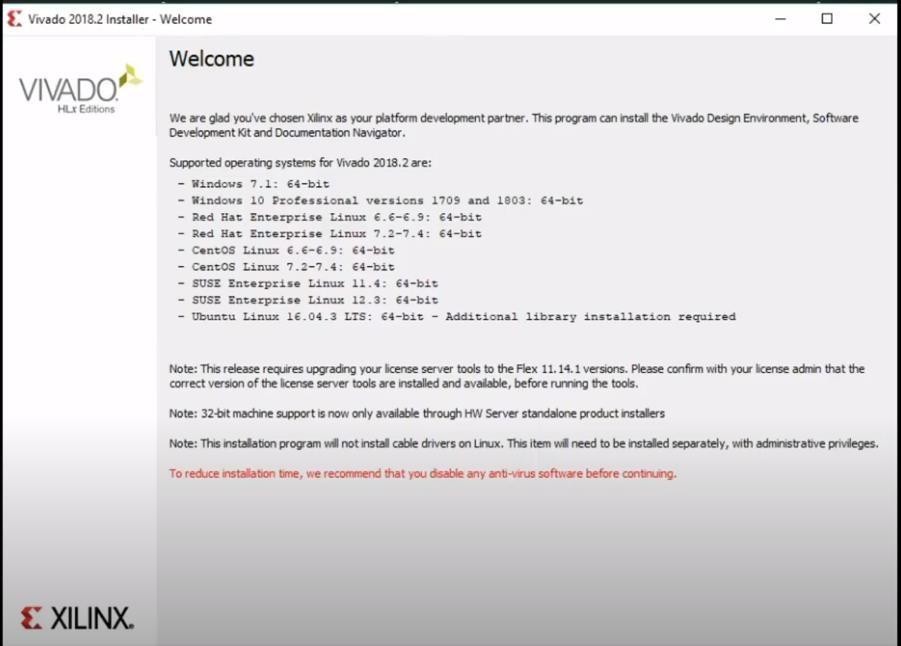
|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **03 June 2020** | **Name:** | **GAURAV NR** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL15EC025** |
| **Topic:** | * **EDA Playground Online complier** * **EDA Playground Tutorial Demo Video** * **How to Download And Install Xilinx Vivado Design Suite** * **Vivado Design Suite for implementation of HDL code** | **Semester & Section:** | **8th sem A sec** |
| **Github Repository:** | **gaurav** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    Report:  Vivado Design Suite for implementation of HDL code  1.In a separate web browser window, log in to EDA Playground  .  2. 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)   1. 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.   1. In either the Design or Testbench window pane, type in the following code: 2. Module test;   4. initial  5. $display("Hello World!");  6. endmodule  7. Click Run(top left) Yes, running a simulation is as simple as that!  8. In the bottom pane, you should see real time results as your code is being  compiled and then run. A run typically takes 5seconds, depending on network traffic and simulator. Near the bottom of result output, you should see:  9. Hello World!  10. Now, let’s save our good work. Click the Share tab near in the bottom pane and  then type in a name and description. Then click Save  11. The browser page will reload and the browser address bar will change.This is a  persistent link to your saved code. You can send the link by email, post it on a  web page, post it on Stack Overflow forums, etc.  12. Now, let’s try modifying existing code. Load the following example: RAM  13. On the left editor pane, before the end of initial block, add the following:  14. write\_enable = 1;  15. data\_write = 8'h2C;  16. toggle\_clk\_write;  17. toggle\_clk\_read;  18. $display("data[%0h]: %0h",  19. address\_read, data\_read);  20. Run the sim. In the results you should see this new message:  21. data[1b]: 2c  22.Optional. 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 SystemVerilog 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. Click Run  .  After the run completes, the resulting waves will load in a new EPWave window.  Loading Waves for VHDL Simulations, 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. |



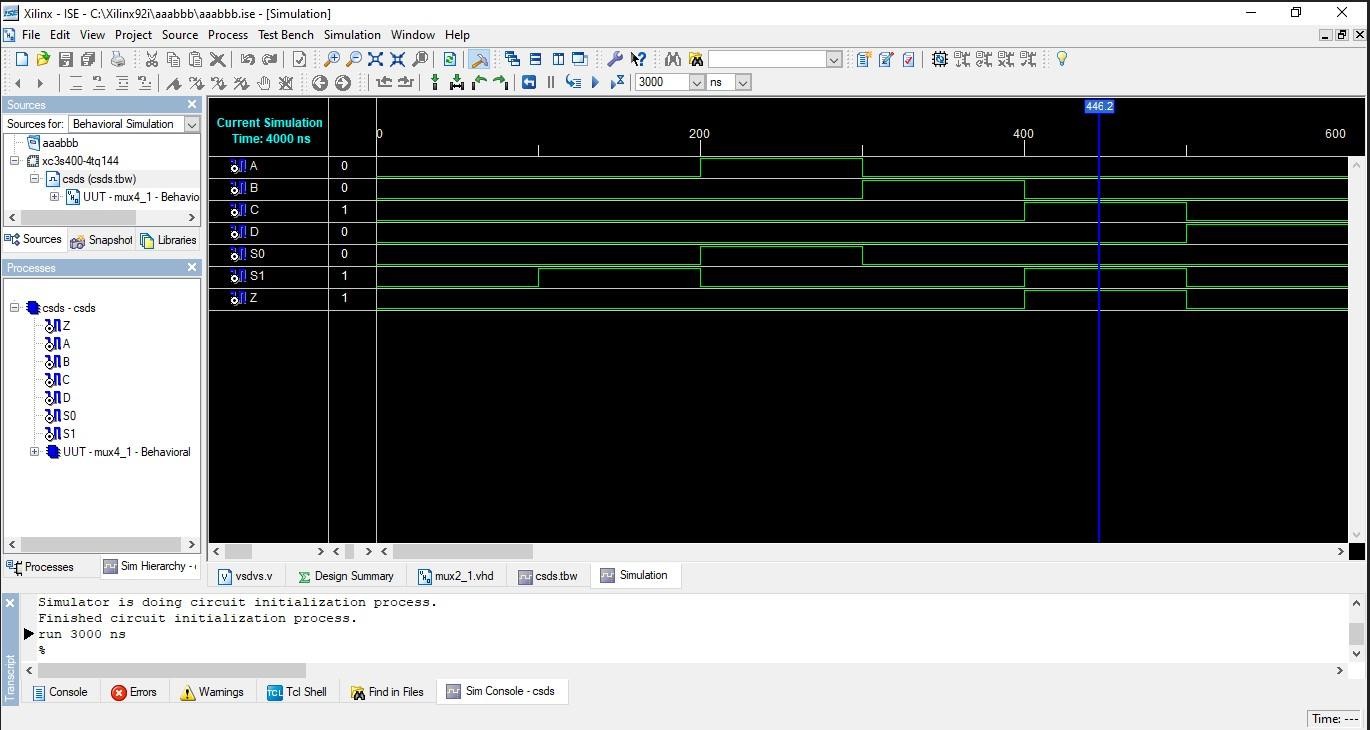
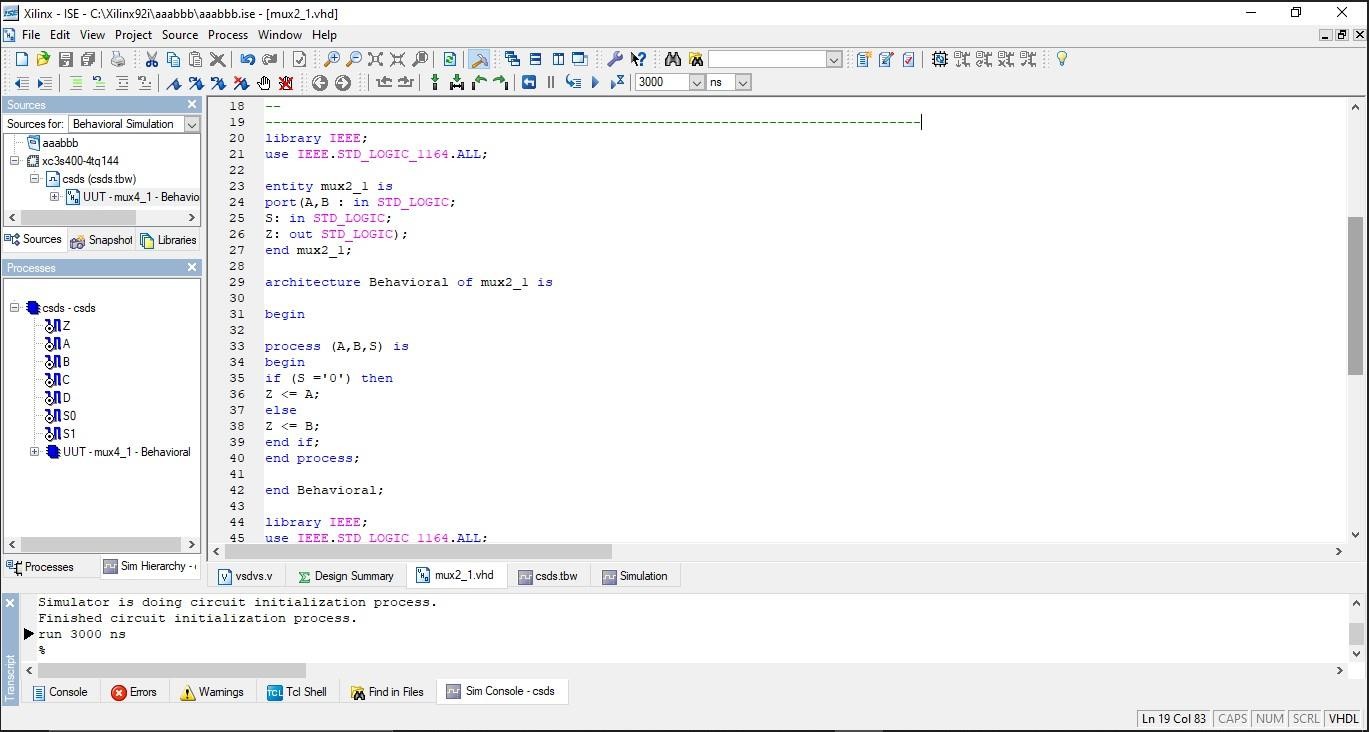
# EDA Playground Online complier:

**EDA Playground Tutorial Demo Video:**



**How to Download And Install Xilinx Vivado Design Suite:**

**Vivado Design Suite for implementation of HDL code:**



**TASK:**

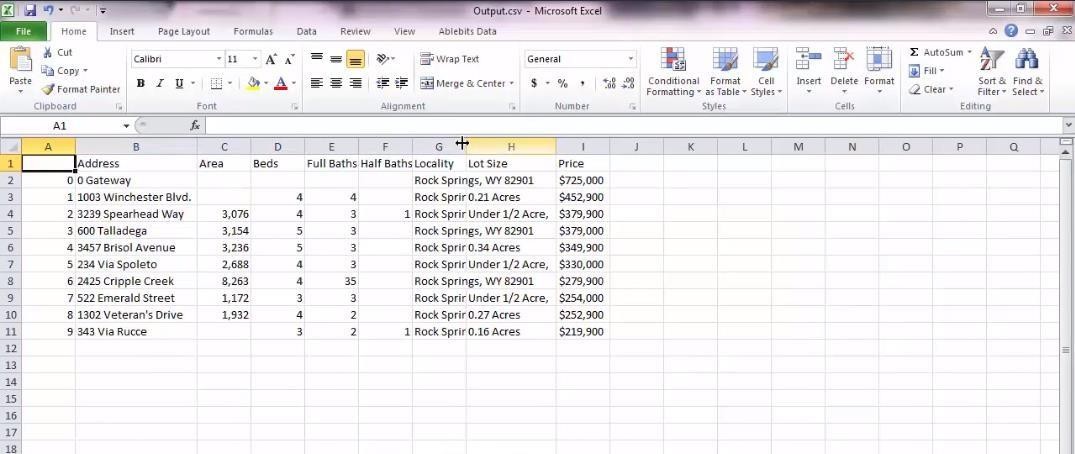
Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler.

# OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **03 June 2020** | **Name:** | **Chesmi B R** |
| **Course:** | **The Python Mega Course** | **USN:** | **4AL16EC100** |
| **Topic:** | **Application 8: Scrape Real Estate Property Data from the Web** | **Semester & Section:** | **8th sem A sec** |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
| **Image of session:** |

Report –



# Scrape Real Estate Property Data from the Web:

* **In this application we learnt how to collect data from various websites using python.**
* **We learnt about loading the webpages in python.**
* **We learnt to extract the “div” tags.**
* **We learnt about extracting addresses and property details.**
* **We learnt about extracting elements without unique identifiers.**
* **We learnt how to save the obtained data in .csv format.**
* **And also we learnt to extract data from various websites at a time using crawling through websites using python.**
* **The output obtained from website is saved in excel sheet as shown below.**

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

–

Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you

to try out different parsing strategies or trade speed for flexibility