**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **3/June/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4al17ec074** |
| **Topic:** | 1. **EDA Playground Online complier** 2. **EDA Playground Tutorial Demo Video** 3. **How to Download and Install Xilinx Vivado Design Suite** 4. **Vivado Design Suite for implementation of HDL code** | **Semester & Section:** | **6th b** |
| **Github Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **EDA Playground** is a web browser-based integrated development environment (IDE) for simulation of System Verilog, Verilog, VHDL, C++/System C and other HDLs. **EDA Playground** is a free web application that allows users to edit, simulate, share, synthesize, and view waves for hardware description language (HDL) code.  **About Xilinx Vivado**  Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of HDL designs, superseding Xilinx ISE with additional features for system on a chip development and high-level synthesis. Vivado represents a ground-up rewrite and re-thinking of the entire design flow (compared to ISE), and has been described by reviewers as "well-conceived , tightly integrated, blazing fast, scalable, maintainable, and intuitive".  Like the later versions of ISE, Vivado includes the in-built logic simulator ISIM. Vivado also introduces high-level synthesis, with a toolchain that converts C code into programmable logic. Vivado has been described as a "state-of-the-art comprehensive EDA tool with all the latest bells and whistles in terms of data model, integration, algorithms, and performance".  Implement 4 to 1 MUX using structural modelling style and  test the module in an 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/June/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **Python** | **USN:** | **4al17ec074** | |
| **Topic:** | **Application 7: Scrape Real Estate Property Data from the Web** | **Semester&Section:** | **6th b** | |
| **Git hub repository** | **prashanth\_couse** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **Program**  **import requests**  **from bs4 import BeautifulSoup**  **r=requests.get(\"http://www.pythonhow.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS")**  **c=r.content**  **soup=BeautifulSoup(c,\"html.parser\")**  **all=soup.find\_all(\"div\",{\"class\":\"propertyRow\"})**  **all[0].find(\"h4\",{\"class\":\"propPrice\"}).text.replace(\"\\n\",\"\").replace**  **page\_nr=soup.find\_all(\"a\",{\"class\":\"Page\"})[-1].text**  **print(page\_nr,\"number of pages were found\")**  **base\_url=\"http://www.pythonhow.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/t=0&s=\"\n",**  **for page in range(0,int(page\_nr)\*10,10)**  **print( )**  **r=requests.get(base\_url+str(page)+\".html\")**  **c=r.content\n**  **#c=r.json()[\"list\"]**  **soup=BeautifulSoup(c,\"html.parser\")\n**  **all=soup.find\_all(\"div\",{\"class\":\"propertyRow\"})**  **for item in all**  **d={}**  **d[\"Address\"]=item.find\_all(\"span\",{\"class\",\"propAddressCollapse\"})[0].text**  **try:**  **d[\"Locality\"]=item.find\_all(\"span\",{\"class\",\"propAddressCollapse\"})[1].text**  **except:**  **d[\"Locality\"]=None**  **d[\"Price\"]=item.find(\"h4\",{\"class\",\"propPrice\"}).text.replace(\"\\n\",\"\").replace(\" \",\"\")\n**  **try:\n**  **d[\"Beds\"]=item.find(\"span\",{\"class\",\"infoBed\"}).find(\"b\").text\n**  **except:**  **d[\"Beds\"]=None**  **try:**  **d[\"Area\"]=item.find(\"span\",{\"class\",\"infoSqFt\"}).find(\"b\").text\n**  **except:**  **d[\"Area\"]=None**  **try:**  **d[\"Full Baths\"]=item.find(\"span\",{\"class\",\"infoValueFullBath\"}).find(\"b\").text\n",**  **except:**  **d[\"Full Baths\"]=None**  **try:**  **d[\"Half Baths\"]=item.find(\"span\",{\"class\",\"infoValueHalfBath\"}).find(\"b\").text\n**  **except:**  **d[\"Half Baths\"]=None**  **for column\_group in item.find\_all(\"div\",{\"class\":\"columnGroup\"})**  **for feature\_group, feature\_name in**  **if \"Lot Size\" in feature\_group.text:**  **d[\"Lot Size\"]=feature\_name.text**  **l.append(d)** | | | |