**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **3/06/2020** | **Name:** | **DHAMINI C L** |
| **Course:** | **Digital Design Using HDL** | **USN:** | **4AL17EC025** |
| **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 & A** |
| **Github Repository:** | **DHAMINI-CL-Course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **Report:**  **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 &quot;well-conceived , tightly integrated, blazing fast, scalable, maintainable,**  **and intuitive&quot;.**  **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 &quot;state-of-the-art comprehensive**  **EDA tool with all the latest bells and whistles in terms of data model, integration,**  **algorithms, and performance&quot;.**  **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 =&#39;0&#39;) then**  **Z &lt;= A;**  **else**  **Z &lt;= 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:3/6/2020** |  | **Name: DHAMINI C L** |  | |
| **Course:PYTHON** |  | **USN:4AL17EC025** |  | |
|  |  |  |  | |
| **Topic: Application 10: Build a Data Collector Web App with PostGreSQL and Flask** |  | **Semester & Section:6TH A SEC** |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**   1. Basic Python knowledge is required. 2. Sign up for an [OpenShift Account](https://www.openshift.com/app/account/new). It is completely free and instant . Red Hat gives every user three free Gears on which to run your applications. At the time of this writing, the combined resources allocated for each user is 1.5 GB of memory and 3 GB of disk space. 3. Install the [rhc client tool](https://openshift.redhat.com/community/get-started#cli) on your machine. The rhc is a ruby gem so you need to have ruby 1.8.7 or above on your machine. To install rhc, just typesudo gem install rhc. If you already have one, make sure it is the latest one. To update your rhc, execute the command shown below.sudo gem update rhc. For additional assistance setting up the rhc command-line tool, see the following page: https://openshift.redhat.com/community/developers/rhc-client-tools-install. 4. Setup your OpenShift account using rhc setup command. This command will help you create a namespace and upload your ssh keys to the OpenShift server.   The command shown above will create an application container for us, called a gear, and setup all of the required SELinux policies and cgroup configuration. Next, it will install all the required software on your gear. It will also install PotsgreSQL 9.2 on your application gear and will create a database with the same name as the application name. OpenShift will also setup a private git repository with some template code, and then clone the repository to your local system. Finally, OpenShift will propagate the DNS to the outside world. | | | |