**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Prajwal Kamagethi Chakravarti P L** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL17EC073** |
| **Topic:** | * **Verilog Tutorials and practice programs** * **Building/ Demo projects using FPGA** | **Semester & Section:** | **6th & B** |
| **GitHub Repository:** | **https://www.github.com/alvas-education-foundation/Prajwal-Kamagethi.git** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report –**   * **FPGA stands for Field Programmable Gate Array. It is an integrated circuit which can be “field” programmed to work as per the intended design.** * **Verilog like any other hardware description language, permits the designers to design a design in either Bottom−up or Top−down methodology.** * **Verilog simulator was first used beginning in 1985 and was extended substantially through 1987.** * **Various stages of ASIC/FPGA:**   **Specification**  **High Level Design**  **Micro Design/Low level design:**  **RTL coding**  **Simulation**  **Synthesis**  **Place and route**   * **An FPGA designer likes working on this due to these reasons:**   **Very fast on-chip (FPGA) demonstration** **Simple and fast design process on FPGA** **FPGA's programmability** **FPGA's high performance** **FPGA's flexibility**  * **We also learnt the different ways and techniques of designing FPGA.**   **Implement a verilog module to count number of 0’s in a 16 bit number in compiler.**  **Verilog code:**  **module num\_zero\_ones**  **(input [15:0] In,**  **output reg [4:0] ones,**  **output reg [4:0] zeros);**  **integer i, o, z;**  **always**  **@(in)**  **begin**  **o = 0;**  **z = 0;**  **for(i=0;i<16;i=i+1)**  **if(In[i] == 1'b1)**  **o = o + 1;**  **z = 16-o;**  **ones = o;**  **zeros = z;**  **end**  **endmodule** |
| [**VHDL**](https://en.wikipedia.org/wiki/VHDL)**code for Clock divider on**[**FPGA**](https://www.fpga4student.com/p/fpga-projects.html)**:** **library** **IEEE**;  **use** **IEEE.STD\_LOGIC\_1164.ALL**;  **USE** **IEEE.STD\_LOGIC\_UNSIGNED.ALL**;  **entity** **clock\_div** **is**  **port** (  clk\_in: **in** **std\_logic**; -- clock input on FPGA  clk\_out: **out** **std\_logic** -- clock output  );  **end** **clock\_div**;  **architecture** **Behavioral** **of** **clock\_div** **is**  **signal** divisor: **std\_logic\_vector**(**27** **downto** **0**):=(**others** =>'0');  **begin**  **process**(clk\_in)  **begin**  **if**(rising\_edge(clk\_in)) **then**  divisor <= divisor + x"0000001";  -- If(divisor>=x"2FAF07F") then -- for running on FPGA -- comment when running simulation  -- Modify the divisor (x"2FAF07F"=49999999) above to get the clock frequency you want:  -- Frequency of clk\_out = Frequency of (clk\_in) divided by (divisor + 1)  -- If the frequency of clk\_in is 50MHz and the divisor is 49999999=x"2FAF07F",  -- the frequency of clk\_out is 1Hz  **if**(divisor>=x"0000001") **then** -- for running simulation -- comment when running on FPGA  -- divisor = 1 => divide clock by half for simulation purposes  divisor <= x"0000000";  **end** **if**;  **end** **if**;  **end** **process**;  --clk\_out <= '0' when divisor < x"17D7840" else '1'; -- half of the clock divisor x"17D7840" = (divisor + 1)/2  clk\_out <= '0' **when** divisor < x"0000001" **else** '1';-- divide clock by half  **end** **Behavioral**; |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Prajwal Kamagethi Chakravarti P L** | |
| **Course:** | **Udemy-python** | **USN:** | **4AL17EC073** | |
| **Topic:** | **Build a data collector web app with PostGreSQL and Flask** | **Semester &Section:** | **6th & B** | |
| **GitHub repository** | **https://www.github.com/alvas-education-foundation/Prajwal-Kamagethi.git** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**      **Report –**   * **In this section we learnt to build a data collector web application, which collects height data from the user and sends the survey result via e-mail.** * **To deploy the application to a live web server, Heroku platform is used and to save the data from the users, database is created in Heroku using credentials** * **We can install Heroku CLI using this shell script.**  **Then we have to Create python virtual environment for the project.**  * **To activate this environment, use this common inside book server directory.** * **Then we install flask and write the code and run it.** * **Then writing the database we need for our application is necessary.** * **We need to define configurations for deploying environments** * **Then we need to use flask package for database manipulations.**  **After finishing the code completely Commit changes using git and push to Heroku.** | | | |