**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **01-06-2020** | **Name:** | **Anand kumar k** |
| **Course:** |  | **USN:** | **4al16ec002** |
| **Topic:** | **FPGA** | **Semester & Section:** | **8thsem ‘A’ sec** |
| **Github Repository:** | **Anand-courses** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    FPGA-strengths |
|  |
| **Verilog code to implement NAND gate in all different styles.**  **1.Gate level modeling in verilog**  **Module nand\_2(output y, input A,B)**  **Wire yd;**  **And(yd,A,B);**  **Not (y,yd);**  **Endmodule;**  **2.Data flow modeling in verilog**  **module nand\_2\_data\_flow (output Y, input A,B);**  **assign Y= ~(A & B);**  **endmodule**  **3.Behaviorial modeling in verilog**  **module nand\_2\_behavioral (output reg Y, input A,B);**  **always @ (A or B)begin**  **if (A == 1’b1 & B == 1’b1)begin**  **Y = 1’b0;**  **End**  **Else**  **Y = 1’b1;**  **End**  **endmodule**  **4. Test bench of the Nand gate using Verilog**  **include “nand\_2\_behavioral.v”**  **module nand\_2\_behavioral\_tb;**  **reg A, B;**  **wire Y;**  **nand\_2\_behavioral indtance0 (Y, A, B);**  **initial begin**  **A = 0; B = 0;**  **#1 A = 0; B = 1;**  **#1 A = 1; B = 0;**  **#1 A = 1; B = 1;**  **End**  **Initial begin**  **$monitor (“%t | A = %d| B = %d| Y = %d”, $time, A, B, Y);**  **$dumpfile(“dump.vcd”);**  **$dumpyars();**  **End**  **Endmodule** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **01-06-2020** | **Name:** | **Anand kumar k** | |
| **Course:** |  | **USN:** | **4al16ec002** | |
| **Topic:** | **python** | **Semester & Section:** | **8thsem ‘A’ sec** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| In this section you learned that:   * we can read an existing file with Python:  1. with open("file.txt") as file: 2. content = file.read()  * we can create a new file with Python and write some text on it:  1. with open("file.txt", "w") as file: 2. content = file.write("Sample text")  * we can append text to an existing file without overwriting it:  1. with open("file.txt", "a") as file: 2. content = file.write("More sample text")  * we can both append and read a file with:  1. with open("file.txt", "a+") as file: 2. content = file.write("Even more sample text") 3. file.seek(0) 4. content = file.read() | | | |