**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **03--06-2020** | **Name:** | **Kavya M M** |
| **Course:** | **HDL** | **USN:** | **4AL17EC040** |
| **Topic:** | **EDA playground, Ripple carry adder,** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Kavya\_ECE040** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **EDA playground:** |
| **Ripple carry adder code:**  A **ripple carry adder** is a logic circuit in which the **carry**-out of each full **adder** is the **carry** in of the succeeding next most significant full **adder**. It is called a **ripple carry adder** because each **carry** bit gets rippled into the next stage.    Design code:  module tff(q, clk, reset);  output reg q;  input clk, reset;  always @(posedge reset or posedge clk) begin  if (reset) begin  q <= 1'b0;  end else begin  q <= ~q;  end  end  endmodule      module rippple\_carry\_counter (q, clk, reset);  output [3:0] q;  input clk, reset;    tff tff0(q[0], clk, reset);  tff tff1(q[1], q[0], reset);  tff tff2(q[2], q[1], reset);  tff tff3(q[3], q[2], reset);    endmodule      Test bench code:  module test;  reg clk,reset;  wire [3:0]q;  ripple\_carry\_counter rcc(q, clk, reset);    initial begin  $dumpfile("dump.vcd");  $dumpvars(1, test);    clk = 1'b0;  reset = 1'b1;  #10 reset = 1'b0;  #200;  reset = 1'b1;  #10 reset = 1'b0;  #50;  $finish;  end    always #5 clk = ~clk;  endmodule  Output:      Task:  Design code for 4:1 Mux using 2:1 Mux using structural style:    https://www.edaplayground.com/x/4wtj  Design code:    Testbench code:    Output: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:3june2020** |  | **Name:Kavya\_M\_M** |  | |
| **Course: python** |  | **USN:4AL17EC040** |  | |
| **Topic: application** |  | **Semester & Section:6th A** |  | |
| **AFTERNOON SESSION DETAILS** | | | |
|  | | | |
| Steps for creating the mobile app using python:   * Install kivy library using   Pip install kivy   * Kivy will help you to make Android Applications using Python * So Kivy is basically when you make Android applications using Python * Creating a "Login Page" (Frontend) * Creating a "Sign Up Page" (Frontend) for New Users * Getting user input * Implementing the “sign up” page(backend) * Implementing the “sign up success ” page * Styling the “login” and “sign up” page * Preparing the environment for deploying the android app * Creating and installing APK file | | | |