**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **3rd June 2020** | **Name:** | **Yashwitha C N** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL17EC099** |
| **Topic:** | **EDA playground** | **Semester & Section:** | **6th sem ‘B’ sec** |
| **Github Repository:** | **Yashwitha-coures** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (236).png  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (237).png  **What is EDA Playground?**    EDA Playground gives engineers immediate hands-on exposure to simulating SystemVerilog, Verilog, VHDL, C++/SystemC, and other HDLs. All you need is a web browser. The goal is to accelerate learning of design/testbench development with easier code sharing and simpler access to EDA tools and libraries.With a simple click, run your code and see console output in real time. View waves for your simulation using EPWave browser-based wave viewer.Save your code snippets (“Playgrounds”).Share your code and simulation results with a web link. Perfect for web forum discussions or emails. Great for asking questions or sharing your knowledge.Quickly try something outTry out a language feature with a small example.Try out a library that you’re thinking of using. Example UsecasesQuick prototyping – try out syntax or a library/language feature.When asking questions on Stack Overflow or other online forums, attach a link to the code and simulation results.Use during technical interviews to test candidates’ SystemVerilog/Verilog coding and debug skills.Try verifying using different verification frameworks: UVM, SVUnit, plain Verilog, or Python.Tools & Simulators For settings and options documentation, see Tools & Simulators OptionsAvailable tools and simulators are below. EDA Playground can support many different tools. Contact us to add your EDA tool to EDA Playground    **Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style:**  module and\_gate( output a,input b,c);  assign a=b&c;  endmodule  module not\_gate(output d ,input e) ;  assign d= ~e;  endmodule  module or\_gate(output l, input m,n);  assign l=m | n;  endmodule  module m21(Y,D0,D1,S);  output Y;  input D0,D1,S;  wire T1,T2,T3;  and\_gate u1(T1,D1,S);  not\_gate u2(T2,S);  and\_gate u3(T3,D0,T2);  or\_gate u4(Y,T1,T3);  endmodule |
|  |