**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Yashaswini R** |
| **Course:** | **Digital Design Using HDL** | **USN:** | **4AL17EC098** |
| **Topic:** | **Verilog Tutorials and practice**  **programs, Building/ Demo projects**  **using FPGA** | **Semester & Section:** | **6th B** |
| **Github Repository:** | **Yashaswini** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| Design style:   * Bottom−up design * Top−down methodology.   Abstract levels of Verilog:   * Behavioural level * Register−Transfer Level * Gate Level   History Of Verilog  Verilog was started initially as a proprietary hardware modeling language by Gateway DesignAutomation Inc. around 1984. It is rumored that the original language was designed by takingfeatures from the most popular HDL language of the time, called HiLo as well as from traditionalcomputer language such as C. At that time, Verilog was not standardized and the languagemodified itself in almost all the revisions that came out within 1984 to 1990.  Various stage of ASIC/FPGA:   * Specification: Word processor like Word, Kwriter, AbiWord, Open Office. * High Level Design: Word processor like Word, Kwriter, AbiWord, for drawing waveform use tools like wave former or testbenches or Word, Open Office. * Micro Design/Low level design: Word processor like Word, Kwriter, AbiWord, for drawing waveform use tools like wave former or test bencher or Word. For FSM State CAD or some similar tool, Open Office. * RTL Coding: Vim, Emacs, context, HDL TurboWriter * Simulation: Modalism, VCS, Verilog−XL, Veriwell, Finsim, iVerilog, VeriDOS. * Synthesis: Design Compiler, FPGA Compiler, Synplify, Leonardo Spectrum. You candownload this from FPGA vendors like Altera and Xilinx for free. * Place & Route: For FPGA use FPGA' vendors P&R tool. ASIC tools require expensiveP&R tools like Apollo. Students can use LASI, Magic. * Post Si Validation: For ASIC and FPGA, the chip needs to be tested in real environment.Board design, device drivers needs to be in place.   Verilog operators:  Arithmetic operators:  •Binary: +, −, \*, /, % (the modulus operator)  • Unary: +, − (This is used to specify the sign)  • Integer division truncates any fractional part  • The result of a modulus operation takes the sign of the first operand  • If any operand bit value is the unknown value x, then the entire result value is x  Register data types are used as unsigned values  Verilog HDL abstract level:  • Behavioral Models: Higher level of modeling where behavior of logic is modeled.  • RTL Models: Logic is modeled at register level  • Structural Models: Logic is modeled at both register level and gate level.  Task:  Implement a verilog module to count number of 0’s in a 16 bit number in  compiler.  module num\_zeros\_for( input  [15:0] A, output reg  [4:0]  ones  );  integer i;  always@(A)  begin  ones = 0;  for(i=0;i<16;i=i+1)  if(A[i] == 0'b1)  ones = ones + 1;  end  endmodule  output  Input = "1010\_0010\_1011\_0010" => Output = "01001" ( 9 in decimal)  Input = "0011\_0110\_1000\_1011" => Output = "01000" ( 8 in decimal) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Yashaswini R** | |
| **Course:** | **Python** | **USN:** | **4AL17EC098** | |
| **Topic:** | **Application** | **Semester & Section:** | **6th B** | |
| **Github Repository** | **Yashaswini** |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
|  | | | |
| Bokeh:   * Bokeh is python library used in data visualization * We can grab data from various source data file formats such as csv, python list, json file so on * Bokeh produces interactive graph     #Plotting percentage of women who received an engineering degree over years  #importing bokeh and pandas  from bokeh.plotting import figure  from bokeh.io import output\_file, show  import pandas  #prepare some data  df=pandas.read\_csv("http://pythonhow.com/data/bachelors.csv")  x=df["Year"]  y=df["Engineering"]  #prepare the output file  output\_file("Line\_from\_bachelors.html")  #create a figure object  f=figure()  #create line plot  f.line(x,y)  #write the plot in the figure object  Show(f) | | | |