**DAY 14 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **02-06-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL16EC008** |
| **Topic:** | 1. **FPGA Basics: Architecture, Applications and Uses** 2. **Verilog HDL Basics by Intel** 3. **Verilog Testbench code to verify the design under test (DUT)** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**      **FPGA Basics: Architecture, Applications and Uses**  FPGA (Field Programmable Gate Array) is an integrated circuit containing gate matrix which can be programmed by the user “in the field” without using expensive equipment. An FPGA contains a set of programmable logic gates and rich interconnect resources, making it possible to implement complex digital circuits. FPGA devices are produced by a number of semiconductor companies: Xilinx, Altera, Actel, Lattice, QuickLogic and Atmel. FPGA Architecture FPGAs are prefabricated silicon chips that can be programmed electrically to implement digital designs. The first static memory based FPGA called SRAM is used for configuring both logic and interconnection using a stream of configuration bits. Today’s modern EPGA contains approximately 3,30,000 logic blocks and around 1,100 inputs and outputs.  The FPGA Architecture consists of three major components   * Programmable Logic Blocks, which implement logic functions * Programmable Routing (interconnects), which implements functions * I/O blocks, which are used to make off-chip connections     **Applications of FPGA**   * FPGAs have gained a quick acceptance over the past decades. Here are the some of the applications of FPGAs in various technologies. * Users can apply them to the wide range of applications like random logics, SPLDs, device controllers, communication encoding and filtering. * The emulation of entire large hardware systems via the use of many interconnected FPGAs. * They offer a powerful solution for meeting machine vision, industrial networking, motor control and video surveillance. * FPGAs are used in custom computing machines. * FPGAs provide a unique combination of highly parallel custom computation and low-cost computation.   **Implement a 4:1 MUX and write the test bench code to verify the module**    **Verilog design**  module mux41(input i0,i1,i2,i3,sel0,sel1,output reg y);     always @(\*)      begin        case ({sel0,sel1})        2'b00 : y = i0;        2'b01 : y = i1;        2'b10 : y = i2;        2'b11 : y = i3;        endcase     end      endmodule    **TestBench**  module tb\_mux41;     reg I0,I1,I2,I3,SEL0,SEL1;    wire Y;       mux41 MUX (.i0(I0),.i1(I1),.i2(I2),.i3(I3),.sel0(SEL0),.sel1(SEL1),.y(Y));       initial begin        I0 =1'b0;        I1= 1'b0;        I2 =1'b0;        I3 =1'b0;        SEL0 =1'b0;        SEL1 =1'b0;        #45 $finish;    end       always #2 I0 = ~I0;    always #4 I1 =~I1;    always #6 I2 =~I1;    always #8 I3 =~I1;    always #3 SEL0 = ~SEL0;    always #3 SEL1 = ~SEL1;       always @(Y)    $display( "time =%0t INPUT VALUES: \t I0=%b I1 =%b I2 =%b I3 =%b SEL0 =%b SEL1 =%b \t output value Y =%b ",$time,I0,I1,I2,I3,SEL0,SEL1,Y); endmodule  OUTPUT  time =0 INPUT VALUES:    I0=0 I1 =0 I2 =0 I3 =0 SEL0 =0 SEL1 =0          output value Y =0 time =2 INPUT VALUES:    I0=1 I1 =0 I2 =0 I3 =0 SEL0 =0 SEL1 =0          output value Y =1 time =3 INPUT VALUES:    I0=1 I1 =0 I2 =0 I3 =0 SEL0 =1 SEL1 =1          output value Y =0 time =6 INPUT VALUES:    I0=1 I1 =1 I2 =0 I3 =0 SEL0 =0 SEL1 =0          output value Y =1 time =8 INPUT VALUES:    I0=0 I1 =0 I2 =0 I3 =0 SEL0 =0 SEL1 =0          output value Y =0 time =14 INPUT VALUES:   I0=1 I1 =1 I2 =1 I3 =0 SEL0 =0 SEL1 =0          output value Y =1 time =15 INPUT VALUES:   I0=1 I1 =1 I2 =1 I3 =0 SEL0 =1 SEL1 =1          output value Y =0 |

**DAY 14 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **02-06-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **PYTHON** | **USN:** | **4AL16EC008** |
| **Topic:** | 1. **Interactive Data Visualization with Bokeh** 2. **Webscraping with Python Beautiful Soup** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**      **Interactive Data Visualization with Bokeh**  Bokeh prides itself on being a library for *interactive* data visualization. Unlike popular counterparts in the Python visualization space, like Matplotlib and Seaborn, Bokeh renders its graphics using HTML and JavaScript. This makes it a great candidate for building web-based dashboards and applications. However, it’s an equally powerful tool for exploring and understanding your data or creating beautiful custom charts for a project or report.  Building a visualization with Bokeh involves the following steps:   * Prepare the data * Determine where the visualization will be rendered * Set up the figure(s) * Connect to and draw your data * Organize the layout * Preview and save your beautiful data creation   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)  **Webscraping with Python Beautiful Soup**  Web Scraping is the process of downloading data from websites and extracting valuable information from that data. The need for Web Scraping is increasing, and so it’s the perfect time to get comfortable using it. The process of web scraping and cleaning the scraped data is logical and can therefore be implemented easily and become second nature after a few attempts. Getting Started Library wise we have a few different choices, including:   * Request * Beautiful Soup * Scrapy * Selenium   Scrapy is a complete web scraping framework which takes care of everything from getting the HTML, to processing the data. Selenium is a browser automation tool that can for example enable you to navigate between multiple pages. These two libraries have a steeper learning curve than Request which is used to get HTML data and BeautifulSoup which is used as a parser for the HTML. |