**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **2/June/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4al17ec074** |
| **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:** | **6th b** |
| **Github Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **FPGA Basics: Architecture, Applications and Uses**  **FPGA Architecture**    ***Figure 2: A simplified CLB: The four-input LUT is formed from two three-input units. (Image source: Wikipedia)***  **The number and arrangement of components in the CLB varies by device; the simplified example in Figure 2 contains two three-input LUTs (1), an FA (3) and a D-type flip-flop (5), plus a standard mux (2) and two muxes, (4) and (6), that are configured during FPGA programming.**  **This simplified CLB has two modes of operation. In normal mode, the LUTs are combined with**  **Current-generation FPGAs include more complex CLBs capable of multiple operations with a single block**    ***Figure 3: The Cyclone V SoC internal block diagram (Image Source: Cornell University)***  **At the other end of the scale, the Stratix 10 SX targets high-performance applications in communications, data center acceleration, high-performance computing (HPC), radar processing and ASIC prototyping; that FPGA includes a quad-core 64-bit Arm Cortex-A53 running at up to 1.5 GHz.**  **FPGA Design**    ***Figure 4: Sample VHDL code for a signed adder (Image Source: Wikipedia)***  **Once the FPGA design has been created and verified using HDL, the compiler takes the text-based file and generates a configuration file that contains information on how the components should be wired together.**  **FPGA Applications**  **Many applications rely on the parallel execution of identical operations; the ability to configure the FPGA’s CLBs into hundreds or thousands of identical processing blocks has applications in image processing, artificial intelligence (AI), data center hardware accelerators, enterprise networking and automotive advanced driver assistance systems (ADAS).**  **Many of these application areas are changing very quickly as requirements evolve and new protocols and standards are adopted. FPGAs enable manufacturers to implement systems that can be updated when necessary.**  **FPGA History: What Comes Next?**  **With these emerging applications, the FPGA market is growing at a healthy clip: It was valued at $5.34 billion in 2016 and is expected to grow to $9.50 billion in 2023, according to industry researchers MarketsandMarkets. That’s a compound annual growth rate (CAGR) of 8.5 percent, compared to a CAGR of about 2 percent for the much larger ($74 billion) general microprocessor market.**  **The exponential growth of data, and the emergence of fast-changing fields such as AI, machine learning, HPC and genomics, require architectures that are fast, flexible and adaptable. FPGAs are well-positioned to take advantage of these new opportunities.**      **Implement a 4:1 MUX and write the test bench code to verify the module**  **module top;**  **wire out;**  **reg a;**  **reg b;**  **reg c;**  **reg d;**  **reg s0, s1;**  **m41 name(.out(out), .a(a), .b(b), .c(c), .d(d), .s0(s0), .s1(s1));**  **initial**  **begin**  **a=1'b0; b=1'b0; c=1'b0; d=1'b0;**  **s0=1'b0; s1=1'b0;**  **#500 $finish;**  **end**  **always #40 a=~a;**  **always #20 b=~b;**  **always #10 c=~c;**  **always #5 d=~d;**  **always #80 s0=~s0;**  **always #160 s1=~s1;**  **always@(a or b or c or d or s0 or s1)**  **$monitor("At time = %t, Output = %d", $time, out);**  **endmodule;** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **2/June/2020** | **Name:** | **Prashantha naik** | |
| **Course:** | **Python** | **USN:** | **4al17ec074** | |
| **Topic:** | 1. **Interactive Data Visualization with Bokeh.** 2. **Web scraping with Python Beautiful Soup** | **Semester&Section:** | **6th b** | |
| **Git hub repository** | **prashanth\_couse** |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  Plotting Weather Data (Practice)   1. import pandas 3. from bokeh.plotting import figure, output\_file, show 5. df=pandas.read\_excel("http://pythonhow.com/data/verlegenhuken.xlsx",sheet\_name=0) 6. df["Temperature"]=df["Temperature"]/10 7. df["Pressure"]=df["Pressure"]/10 9. p=figure(plot\_width=500,plot\_height=400,tools='pan') 11. p.title.text="Temperature and Air Pressure" 12. p.title.text\_color="Gray" 13. p.title.text\_font="arial" 14. p.title.text\_font\_style="bold" 15. p.xaxis.minor\_tick\_line\_color=None 16. p.yaxis.minor\_tick\_line\_color=None 17. p.xaxis.axis\_label="Temperature (°C)" 18. p.yaxis.axis\_label="Pressure (hPa)" 20. p.circle(df["Temperature"],df["Pressure"],size=0.5) 21. output\_file("Weather.html") 22. show(p)   **Web scraping with Python Beautiful Soup**   * **learnt concept behind the web scraping** * **learnt about Request headers** | | | |