**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **02/06/2020** | **Name:** | **Lepakshi T V** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL17EC044** |
| **Topic:** | **FPGA basics, Verilog HDL basics** | **Semester & Section:** | **6th sem, A sec** |
| **Github Repository:** | **Lepakshi-044** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  0918_FPGA-basics--architecture-applications-and-uses_main |
| **Report – Report can be typed or hand written for up to two pages.**  **FPGA:**  The [**field-programmable gate array (FPGA)**](https://www.arrow.com/en/categories/programmable-devices/programmable-logic-devices/fpgas) is an integrated circuit that consists of internal hardware blocks with user-programmable interconnects to customize operation for a specific application.  A basic FPGA architecture consists of thousands of fundamental elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects, called a fabric, that routes signals between CLBs. Input/output (I/O) blocks interface between the FPGA and external devices.  A lookup table (LUT) is a characteristic feature of an FPGA. An LUT stores a predefined list of logic outputs for any combination of inputs.  **FPGA Applications:**  Image processing  Artificial intelligence (AI)  Data center hardware accelerators  Enterprise networking and automotive advanced driver assistance systems (ADAS)  **Verilog HDL:**  Introduced in 1984 by Gateway Design Automation.  HDL-Hardware Description Language  Behavioural Modelling:  Only the Functionality of circuit, no structure  RTL-Register transfer logic  **Module Structure:**  Begins with keywords module and ends with endmodule, case-sensitive.  White space is used for readability.  All keywords are lowercase.  Port Types:  input-input port  output-output port  Inout-bidirectional port  Port Declarations:  <port\_type> <port\_name>  Two types of datatypes i.e., net and variable data type.  Bus declarations:  reg[MSB:LSB]<signal name>  reg[MSB:LSB]<signal name>  Instantiation format:  <component\_name>#<delay> <instance\_name> (port\_list);  Two methods to define connections i.e., By ordered list and By name.  Parameter-value assigned to a symbolic name, can be overwritten.  **Operators:**  Arthimetic operators: +(add), -(sub), \*(mul), /(div), %(modulus), \*\*(exponent)  Relational operators: > (Greater than), < (Lesser than), >=(Greater than or equal to), <= (lesser than or equal )  Equality operators: == (Equality), != (inequality), === (case-equality), !==(case-inequality)  Shift operators: << (logical shift left), >> (logical shift right)  Miscellaneous operators: ?: (conditional test), {} (concatenate), {{}} (replica)  Making assignments:  Continuous assignment statements and Procedural assignments blocks (initial and always)  Two types of Procedural assignments i.e., blocking and nonblocking assignment.  Two types of RTL processes i.e., combinational and clocked.  Functions and Tasks are subprograms which are defined within a module.  **Writing Test Benches:**  Test benches for combinational designs.  Test benches for sequential designs.  Generate clock and synchronize the applied inputs.  Generating random test vectors.  The system task $random can be used to generate a random number.  It is called as $random(<seed>)  The value of <seed> is optional and is used to ensure that the same sequence of random numbers are generated each time the test is run. |

**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **02/06/2020** | **Name:** | **Lepakshi T V** |
| **Course:** | **Python** | **USN:** | **4AL17EC044** |
| **Topic:** | **Interactive Data Visualization with brokeh** | **Semester & Section:** | **6th sem, A sec** |
| **Github Repository:** | **Lepakshi-044** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  Snippet producing the triangle based plot  #Making a basic Bokeh line graph  #importing Bokeh  from bokeh.plotting import figure  from bokeh.io import output\_file, show  #prepare some data  x=[3,7.5,10]  y=[3,6,9]  #prepare the output file  output\_file("Line.html")  #create a figure object  f=figure()  #create line plot  f.triangle(x,y)  #write the plot in the figure object  show(f)  Snippet producing the circle based plot  #Making a basic Bokeh line graph  #importing Bokeh  from bokeh.plotting import figure  from bokeh.io import output\_file, show  #prepare some data  x=[3,7.5,10]  y=[3,6,9]  #prepare the output file  output\_file("Line.html")  #create a figure object  f=figure()  #create line plot  f.circle(x,y)  #write the plot in the figure object  show(f)  import pandas  from bokeh.plotting import figure, output\_file, show  df=pandas.read\_excel("http://pythonhow.com/data/verlegenhuken.xlsx",sheet\_name=0)  df["Temperature"]=df["Temperature"]/10  df["Pressure"]=df["Pressure"]/10  p=figure(plot\_width=500,plot\_height=400,tools='pan')  p.title.text="Temperature and Air Pressure"  p.title.text\_color="Gray"  p.title.text\_font="arial"  p.title.text\_font\_style="bold"  p.xaxis.minor\_tick\_line\_color=None  p.yaxis.minor\_tick\_line\_color=None  p.xaxis.axis\_label="Temperature (°C)"  p.yaxis.axis\_label="Pressure (hPa)"  p.circle(df["Temperature"],df["Pressure"],size=0.5)  output\_file("Weather.html")  show(p) |