**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **1/06/2020** | **Name:** | **PADMINI M** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL17EC066** |
| **Topic:** | **Industry Applications of FPGA**  **FPGA Business Fundamentals**  **FPGA vs ASIC Design Flow** | **Semester & Section:** | **6th Bsec** |
| **Github Repository:** | **Padmini** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**   * The FPGA is Field Programmable Gate Array. It is a type of device that is widely used in electronic circuits. FPGAs are semiconductor devices which contain programmable logic blocks and interconnection circuits. It can be programmed or reprogrammed to the required functionality after manufacturing. * When a circuit board is manufactured and if it contains an FPGA as a part of it. This is programmed during the manufacturing process and further can be reprogrammed later to create an update or make necessary changes. * This feature of FPGA makes it unique from ASIC. Application Specific Integrated Circuits (ASIC) are custom manufactured for specific design task. In past FPGAs are used to develop low speed, complex and volume design, but today FPGA easily pushes the performance barrier up to 500MHz. * In microcontrollers, the chip is designed for a customer and they have to write the software and compile it to hex file to load onto the microcontroller. This software can be easily replaced as it is stored in flash memory. * In FPGAs, there is no processor to run the software and we are the one designing the circuit. We can configure an FPGA as simple as an AND gate or a complex as the multi-core processor.      * To create a design we write Hardware Description Language (HDL), which is of two types – Verilog and VHDL. Then the HDL is synthesized into a bit file using a BITGEN to configure the FPGA. * The FPGA stores the configuration in RAM, that is the configuration is lost when there is no power connectivity. Hence, they must be configured every time power is supplied. * 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 * Programmable Logic Blocks * The programmable logic block provides basic computation and storage elements used in digital systems. A basic logic element consists of programmable combinational logic, a flip-flop, and some fast carry logic to reduce area and delay cost.   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. * The adoption of FPGA technology continues to increase as higher-level tools such as LabVIEW are making FPGAs more accessible. It is still important, however, to look inside the FPGA and appreciate how much is actually happening when block diagrams are compiled down to execute in silicon. Comparing and selecting hardware targets based on flip-flops, LUTs, multipliers, and block RAM is the best way to choose the right FPGA chip for your application. Understanding resource usage is extremely helpful during development, especially when optimizing for size and speed. This paper is not meant to be a comprehensive list of all FPGA fundamental building blocks. You can learn more about FPGAs and digital hardware design from the resources below. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **1/06/2020** | **Name:** | **PADMINI M** | |
| **Course:** | **PYTHON** | **USN:** | **4AL17EC066** | |
| **Topic:** | **Build a**  **Web-based Financial Graph** | **Semester & Section:** | **6th Bsec** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**   * Dash is a Python-based framework which is predominantly used to create web applications that include intense data visualizations. * As the name suggests, Dash is specialized for creating web-based dashboard applications. But this is achieved by using purely Python with the help of some Python libraries. * Dash not only helps you to create interactive graphs, but also to embed those graphs into a user-friendly UI. As it’s written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces. * It’s particularly suited to anyone who works with data in Python. And Dash abstracts away all of the technologies and protocols that are required to build an interactive web-based application, so it makes it much more convenient for the developers. * As Dash’s callbacks are functional in nature, it’s easy to add memoization caching. Memoization stores the results of a function after it is called and re-uses the result if the function is called with the same arguments. * Dash applications are usually dashboards which will be deployed across multiple platforms. Furthermore, they are frequently deployed across multiple processes or threads. * The stock market is one of the most interesting places for a data scientist to play. There is a lot of data, and the possibilities for analysis and prediction are unlimited. It is also one of the hot topics students love to use when they start to learn Machine Learning, after all, who doesn’t want to know if a share will have a higher or lower price * However, work with this kind of data can be a little bit tricky, especially out of the comfort zone of Kaggle datasets, when you want to select the stocks that interest you to analyze. Also, what is the correct price we should use to make visualizations and predictions? And how to make a simple yet effective visualization * Candlestick charts are one of the best ways to visualize stock data because they give us very detailed information about the evolution of share prices. In fact, they give us information about four major values at the same time.   **RPA CERTIFICATE** | | | |