**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **4th June 2020** | **Name:** | **Rashmitha** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL17EC077** |
| **Topic:** | **Hardware modeling using Verilog & fpga and asic** | **Semester & Section:** | **6th sem ‘B’ sec** |
| **Github Repository:** | **Rashmitha** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (245).png  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (246).png  **Hardware modeling using Verilog**    Hardware modeling using Verilog.it uses various digital circuit modeling issues using Verilog ,writing test benches and some case studies.    **Fpga and asic**    FPGA Basics – A Look Under the Hood An introductory look inside Field Programmable Gate Arrays. We’ll go over:Strengths & Weaknesses of FPGAs How FPGAs work What’s inside an FPGA So you keep hearing about FPGAs being utilized in more and more applications, but aren’t sure whether it makes sense to switch to a new technology. Or maybe you’re just getting into the embedded world and want to figure out if an FPGA-based system makes sense for you or not.This paper provides an overview of some of the key elements of FPGAs for engineers interested in utilizing FPGA-based technologies. It’s worth noting that this is a complex topic, and as such, some topics are not covered, some are just introductory, and others will evolve over time.  This paper should still give you a lot of helpful information if you’re new to the world of FPGAs.What are the most important things you should know right away?Get out of the software mindset – You’re not writing software. Let me say that again because this is the single most important point if you’re thinking about working with FPGAs.You-are-NOT-writingsoftware.You’re designing a digital circuit. You’re using code to tell the chip how to configure itself.Plan for lots of bugs – yes, plan for them. They are going to happen. Way more than you expected. If you’re a newbie developer, you need to pull in someone that has experience with FPGA development to help with this estimate.Application-specific realities – you ought to concern yourself with realities revolving around cyber security and safety, as FPGAs are a different animal than what you’re likely used to.What is an FPGA?An FPGA is a (mostly) digital, (re-)configurable ASIC. I say mostly because there are analog and mixed-signal aspects to modern FPGAs. For example, some have A/D converters and PLLs. I put re- in parenthesis because there are actually one-timeprogrammable FPGAs, where once you configure them, that’s it, never again. However, most FPGAs you’ll come across are going to be re-configurable. So what do I mean by digitally configurable ASIC?I mean that at the core of it, you’re designing a digital logic circuit, as in AND, OR, NOT, flip-flops, etc. Of course that’s not entirely accurate and there’s much more to it than that, but that is the gist at its core.he players –There are currently two big boys: Altera (part of Intel) and Xilinx, and some supporting players (e.g. Actel (owned by Microsemi)).The main underlying technology options are SRAM-based (this is the most common technology), flash, and anti-fuse. As you might imagine, each option has its own pros and cons. Check this out for some more details.Strengths / best suited for:Much of what will make it worthwhile to utilize an FPGA comes down to the low-level functions being performed within the device. There are four processing/algorithm attributes defined below that FPGAs are generally well-suited for. While just one of these needs may drive you toward an FPGA, the more of these your application has, the more an FPGA-based solution will appeal.Parallel processes – if you need to process several input channels of information (e.g. many simultaneous A/D channels) or control several channels at once (e.g. several PID loops). High data-to-clock-rateratio – if you’ve got lots of calculations that need to be executed over and over and over again, essentially continuously. The advantage is that you’re not tying up a centralized processor. Each function can operate on its own. Large quantities of deterministic I/O – the amount of determinism that you can achieve with an FPGA will usually far surpass that of a typical sequential processor. If there are too many operations within your required loop rate on a sequential processor, you may not even have enough time to close the loop to update all of the I/O within the allotted time. Signal processing – includes algorithms such as digital filtering, demodulation, detection algorithms, frequency domain processing, image processing, or control algorithms. Weaknesses / not optimal for:With any significant benefit, there’s often times a corresponding cost.  blocks that allow for various voltage standards (e.g. LVCMOS, LVDS) as well as timing delay elements to help align multiple signals with one another (e.g. for a parallel bus to an external RAM chip).Clocking and routing –This is really a more advanced topic, but critical enough to at least introduce. You’ll likely use an external oscillator and feed it into clocking resources that can multiply, divide, and provide phase-shifted versions of your clock to various parts of the FPGA.Routing resources not only route your clock to various parts of the FPGA, but also your data. Routing resources within an FPGA are one of the most underappreciated elements, but so critical. Check out this sea of madness:What’s Inside – Advanced componentsHard cores – These are functional blocks that (at least for the most part) have their own dedicated logical resources. In other words, they are already embedded into your FPGA silicon. You configure them with various parameters and tell the tools to enable them for you. This could include functions such as high-speed communications (e.g. high-speed serial, Ethernet), low-speed A/D converters for things like measuring slowly varying voltages, and microprocessor cores to handle some of the functions that FPGA logic is not as well suited for.Soft cores – These are functional blocks that don’t have their own dedicated logical resources. |
|  |

**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **4th June 2020** | **Name:** | **Rashmitha** |
| **Course:** | **Python** | **USN:** | **4AL17EC077** |
| **Topic:** | **Build a web based financial graph** | **Semester & Section:** | **6th sem ‘B’ sec** |
| **Github Repository:** | **Rashmitha** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
| **Image of session**  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (247).png  C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (248).png  **BUILD A WEB BASED FINANCIAL GRAPH**      The study aims to inspect the stableness of interactive affinity between search interest of prices of the stock and evidence stock market out comes on worldwide equity market indices. This study represents and develop former exploration into financial graph by registering the attributes and magnitudes of the graph use and embarkment from representational impartiality.such a paradox could also be derived through investors behavior and degree of disclosure inclusion. |
|  |