**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **2 june 2020** | **Name:** | **Sanketh S Acharya** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL17EC084** |
| **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 SEM & ‘B’ SEC** |
| **Github Repository:** |  |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **C:\Users\cw\Desktop\2 j4.PNG** |
| **Report –**  **What is 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. The interconnects can readily be reprogrammed, allowing an FPGA to accommodate changes to a design or even support a new application during the lifetime of the part.    The FPGA has its roots in earlier devices such as programmable read-only memories (PROMs) and programmable logic devices (PLDs). These devices could be programmed either at the factory or in the field, but they used fuse technology (hence, the expression “burning a PROM”) and could not be changed once programmed. In contrast, FPGA stores its configuration information in a re-programmable medium such as static RAM (SRAM) or flash memory. FPGA manufacturers include [**Intel**](https://www.arrow.com/en/manufacturers/intel), Xilinx, [**Lattice Semiconductor**](https://www.arrow.com/en/manufacturers/lattice-semiconductor), [**Microchip Technology**](https://www.arrow.com/en/manufacturers/microchip-technology) and [**Microsemi**](https://www.arrow.com/en/manufacturers/microsemi). ****FPGA Architecture**** A basic FPGA architecture (Figure 1) 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.  Depending on the manufacturer, the CLB may also be referred to as a logic block (LB), a logic element (LE) or a logic cell (LC).  C:\Users\cw\Desktop\2 j1.PNG  Figure 1: The fundamental FPGA architecture (Image Source: National Instruments)  An individual CLB (Figure 2) is made up of several logic blocks. 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: LUTs with four to six input bits are widely used. Standard logic functions such as multiplexers (mux), full adders (FAs) and flip-flops are also common.  C:\Users\cw\Desktop\2 j2.PNG  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 Mux 2 to form a four-input LUT; in arithmetic mode, the LUT outputs are fed as inputs to the FA together with a carry input from another CLB. Mux 4 selects between the FA output or the LUT output. Mux 6 determines whether the operation is asynchronous or synchronized to the FPGA clock via the D flip-flop.  Current-generation FPGAs include more complex CLBs capable of multiple operations with a single block; CLBs can combine for more complex operations such as multipliers, registers, counters and even digital signal processing (DSP) functions FPGA Design How do we transform this collection of thousands of hardware blocks into the correct configuration to execute the application? An FPGA-based design begins by defining the required computing tasks in the development tool, then compiling them into a configuration file that contains information on how to hook up the CLBs and other modules. The process is similar to a software development cycle except that the goal is to architect the hardware itself rather than a set of instructions to run on a predefined hardware platform.  Designers have traditionally used a hardware description language (HDL) such as VHDL (Figure 4) or Verilog to design the FPGA configuration.  **C:\Users\cw\Desktop\2 j3.PNG** ****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.  A good example of FPGA use is high-speed search: Microsoft is using FPGAs in its data centers to run Bing search algorithms. The FPGA can change to support new algorithms as they are created. If needs change, the design can be repurposed to run simulation or modeling routines in an HPC application. This flexibility is difficult or impossible to achieve with an ASIC.  Other FPGA uses include aerospace and defense, medical electronics, digital television, consumer electronics, industrial motor control, scientific instruments, cybersecurity systems and wireless communications. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:2 june 2020** |  | **Name:Sanketh S Acharya** |  |
| **Course:python** |  | **USN:4AL17EC084** |  |
| **Topic:** |  | **Semester & Section:6th sem & ‘B’ sec** |  |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**  **C:\Users\cw\Desktop\2 j5.PNG** | | | |
| **Report –** | | | |