**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **02-06-2020** | **Name:** | **Kiran N** |
| **Course:** | **Logic design** | **USN:** | **4al16ec031** |
| **Topic:** |  | **Semester & Section:** | **8th and A** |
| **Github Repository:** | **Kiran-course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    **REPORT**  **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).  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.  **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 |

|  |
| --- |
| **2.Verilog HDL basics in intel**    **3. Verilog test bench code to verify the design under test**    **EXAMPLE**  Y=(b’.c’)+(a.b’)  module sillyfunction(input a, b, c,  output y);  assign y = ~b & ~c | a & ~b;  endmodule  TEST BENCH CODE  module testbench1();  reg a, b, c;  wire y;  sillyfunctiondut (.a(a), .b(b), .c(c), .y(y) );d  initial begin  a = 0; b = 0; c = 0; #10;  c = 1; #10;  b = 1; c = 0; #10;  c = 1; #10;  a = 1; b = 0; c = 0; #10;  end  endmodule  **Task 2**  Implement a 4:1 MUX and write the test bench code to verify the module  module mux\_4to1\_assign ( input [3:0] a, b, c, d, input [1:0] sel, output [3:0] out);  assign out = sel[1] ? (sel[0] ? d : c) : (sel[0] ? b : a);  endmodule  Testbench  module tb\_4to1\_mux;  reg [3:0] a, b, c, d;  wire [3:0] out;  reg [1:0] sel;  integer i;  mux\_4to1\_case mux0 ( .a (a), .b (b), .c (c), .d (d), .sel (sel), .out (out));  initial begin  $monitor ("[%0t] sel=0x%0h a=0x%0h b=0x%0h c=0x%0h d=0x%0h out=0x  %0h", $time, sel, a, b, c, d, out);  sel<= 0; a <= $random; b <= $random; c <= $random; d <= $random;  for (i = 1; i < 4; i=i+1) begin  #5 sel<= i;  end  #5 $finish;  end  endmodule |