|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Hardware Description Language Lab (2UXL401)** | **Semester:** | **IV** |
| **Date of Performance:** | **04 / 05 / 2021** | **Batch No:** | **B2** |
| **Faculty Name:** | **Prof. Bhargavi Kaslikar** | **Roll No:** | **1912060** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** |  |

**Experiment No: 9**

**Title:** Asynchronous Counter using Verilog

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Write a Verilog code for a T FF and implement a 4 bit asynchronous counter using this as a structured model. It has an asynchronous reset. For T ff when t = 1, the output toggles, otherwise the output doesn’t change  Test using test bench and Implement the same on CPLD Kit. |

|  |
| --- |
| **COs to be achieved:** |
| **CO 1**: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications  **CO 2**: Test a VHDL code and verify the circuit model.  **CO 3**: Synthesize and Implement the designed circuits on CPLD/ FPGA. |

|  |
| --- |
| **Work to be done** |
| Write a verilog code for T FF. Use this to Upload Verilog codes. Also upload test bench and simulation for the same. |
| **Main Code:** |
| module Async\_counter(q,clk,reset);  output [3:0] q;  input clk, reset;  T\_FF tff0(q[0],clk, reset);  T\_FF tff1(q[1],q[0], reset);  T\_FF tff2(q[2],q[1], reset);  T\_FF tff3(q[3],q[2], reset);  endmodule  module T\_FF(q\_t,clk\_t,reset\_t);  output q\_t;  input clk\_t, reset\_t;  wire d;  D\_FF dff0(q\_t,d, clk\_t, reset\_t);  not n1(d,q\_t);  endmodule  module D\_FF(q\_d,d\_d,clk\_d,reset\_d);  output q\_d;  input d\_d, clk\_d, reset\_d;  reg q\_d;  always @(posedge reset\_d or negedge clk\_d)  if (reset\_d)  q\_d <= 1'b0;  else  q\_d <= d\_d;  endmodule |
| **Testbench:** |
| module Async\_counter\_tb;  reg clk;  reg reset;  wire [3:0] q;  Async\_counter uut(.clk(clk), .reset(reset), .q(q));    initial  begin  clk=0;  end  always begin #10 clk=~clk;  end    initial  begin  reset=1;  #10 reset=0;  #500$stop;  end  endmodule |
|  |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| Upload Answer of following question before coming to next laboratory.   1. Which of the following is true about the following verilog module?   module guess (A, B, C, D, E, F, Y);  input A,B,C,D,E,F; output Y, wire t1, t2, y;  nand #1 G1(t1,A,B,C) G2(t2,D,E,F);  and #1  G3 (y,t1,t2); end module   * 1. The module represents a design at the structural level.   2. The module represents a design at the functional level.   3. The module represents a design at the layout level.   4. None of the above.   Ans:  **Option a). The module represents a design at the structural level.**   1. For the following Verilog code segment: wire [7:0] A;   wire B;  assign B = ~|A;  if the value of A is 8’b00111001, what will be the value of {A[5:3], 3{B}}?  a. 6’b111111  b. 6’b011000  c. 6’b111000  d. None of the above  Ans:  **Option c). 6’b111000**   1. Identify the valid statements for Verilog operators. 2. The bitwise operators like & and | can acts as unary as well as binary operator. 3. The LHS of “assign” operator can be register type or net type variable. 4. The shift operators X>>2 and X>>>2 yields similar results when X is a signed number. 5. The equality operator == and === returns same logic value when either of the operands contains all possible logic values   Ans: Option a). The bitwise operators like & and | can acts as unary as well as binary operator.   1. Analyze the following Verilog code   module Q4(clk, reset ,s, q);  input clk,s,reset;  output reg [3:0] q;  always @ (posedge clk)  begin  if (reset)  q<=4'b0010;  else  q<={s,q[3:1]};  end  endmodule  Ans:  In the given code, clk, reset and s are the input registers, q is a 4 bit output register. The clock is a positive edge triggered clock which begins the execution of code. Reset is synchronous.  When reset is synchronized with clk, value of output q is “0010”. When reset is not in synchronization with clk, the value of output q depends on ‘s’ i.e.  If s = ‘1’ 🡪 q = “1001”  If s = ‘0’ 🡪 q = “0001” |

|  |
| --- |
| **Conclusion:**  **Thus, in this experiment we have implemented an asynchronous counter in Verilog using T flip flops and the T flip flops used D flip flops. Hence, we have used a Top Down approach to implement the counter.** |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |