JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -3 EXAMINATIONS- 2026

B.Tech.-VI Semester (CSE/IT/BT/**ECE**/CE)

COURSE CODE (CREDITS): 18B1WEC744 (3) MAX MARKS: 35

COURSE NAME: : FPGA Based Instrumentation System Design

COURSE INSTRUCTOR: Dr. Harsh Sohal MAX. TIME: 2 Hours

Note: *(a) All questions are compulsory.*

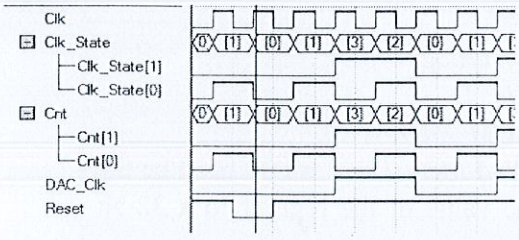
(b) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

(c) Use of calculator is not allowed

Q.No	Question	CO	Marks
Q1	(a) Declare the following variables in Verilog: 1) An integer called *Counter_Up*. 2) An16-bit vector net called *ROM_Address_in*. 3) A 32-bit storage register called *Mem_Data*. Bit 31 must be the most significant bit. Set the value of the register to a 32-bit decimal number equal to 3. 4) A time variable called *time_1*. 5) A parameter *number_of_taps* equal to 128. 6) An array called delays. Array contains 20 elements of the type integer. (b) What would be the output/effect of the following statements? 1) in_reg = 4'd20; $monitor($time, " In register value = %b\n", in_reg[2:0]); 2) `define MEM_SIZE 1024 $display("The maximum memory size is %h", 'MEM_SIZE);	1	3+1

Q2	(a) Compare the Precision vs Accuracy w.r.t. measuring instruments. Is it possible to have an instrument with high degree of precision and poor in accuracy? Discuss.
	(b) What is the order of execution of statements in the following Verilog code? Is there any ambiguity in the order of execution? What are the final values of *a, b, c, d*?
	`initial`
	`begin`
	`a = 1'b0;`
	`#0 c = b;`
	`end`
	`initial`
	`begin`
	`b = 1'b1;`
		5	2+2+2

	`#0 d = a;`
	`End`
	(c). What is the final value of d in the following example? (Hint: See intra-assignment delays.)
	`initial`
	`begin`
	`b = 1'b1; c = 1'b0;`
	`#10 b = 1'b0;`
	`initial`
	`begin`
	`d = #25 (b | c);`
	`end`		

Q3	Write the verilog Code for the following waveform: (Clk, Reset are the inputs; rest are outputs.) 	4	4

Q4	(a) What are blocking and non-blocking assignments? How can they be used to eliminate race conditions? Give suitable Verilog codes to explain.
	(b) Compare and contrast the outputs of the following codes: Code 1 and Code 2. How the outputs of the two will differ? Draw and compare the waveforms with explanation.
	Code 1.
	`reg x, y;`
	`reg [1:0] z, w;`
	`initial`
	`fork`
	`x = 1'b0;`
	`#5 y = 1'b1;`
	`#10 z = {x, y};`
	`#20 w = {y, x};`
	`join`
	Code 2.
	`reg x, y;`
	`reg [1:0] z, w;`
	`initial`
	`begin`
	`x = 1'b0;`
	`#5 y = 1'b1;`
		4	4+4

	`#10 z = {x, y};`
	`#20 w = {y, x};`
	`end`		

Q5.	(a) What sequential and parallel blocks? Differentiate between the two using suitable Verilog codes. Also suggest suitable applications where one may be preferred over the other. (b) How are tasks and function different in Verilog HDL? Differentiate the two in detail by giving example codes.	2,3	4+4
Q6.	Consider the following specifications and design a Finite State Machine diagram for a highway traffic light: o The traffic signal for the main highway gets highest priority because cars are continuously present on the main highway. Thus, the main highway signal remains green by default. o Occasionally, cars from the country road arrive at the traffic signal. The traffic signal for the country road must turn green only long enough to let the cars on the country road go. o As soon as there are no cars on the country road, the country road traffic signal turns yellow and then red and the traffic signal on the main highway turns green again. o There is a sensor to detect cars waiting on the country road. The sensor sends a signal X as input to the controller. X = 1 if there are cars on the country road; otherwise, X= 0. Using above information design a state transition *table* and state *diagram.*	5	5