U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: V**

Branch: Electronics & Telecommunication Engineering **Duration: 3 hrs.**

Course Code: 19ET5PE2VH **Max Marks: 100**

Course: Verilog HDL

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

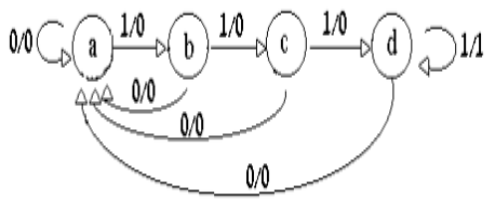
			CO	PO	Marks
		UNIT - I			
1	a)	Explain the top-down methodology with an example.	CO1	PO1	**6**
	b)	Explain briefly the four levels of abstraction in digital design.	CO1	PO1	**8**
	c)	What are the useful features of Verilog HDL for hardware design?	CO1	PO1	**6**
		OR			
2	a)	Explain the importance of Verilog HDL	CO1	PO1	**6**
	b)	Explain Dataflow, Gate level, Behavioral and Switch level abstractions in Verilog	CO1	PO1	**8**
	c)	Explain full adder using Top- Down and Bottom-Up design methodologies	CO1	PO1	**6**
		UNIT - II			
3	a)	Explain the following with an example. i)Nets ii)Registers iii)Parameters iv)Arrays	CO1	PO1	**06**
	b)	Explain the rules to be followed while connecting the ports in the Verilog module.	CO1	PO1	**06**
	c)	What does $display, $monitor, $finish and $stop statements specify in Verilog?	CO1	PO1	**08**
		OR			
4	a)	Explain different data types in VERILOG	CO1	PO1	**08**
	b)	What are the basic components of a module? Which components are mandatory?	CO1	PO1	**06**
	c)	Explain different ports in Verilog	CO1	PO1	**06**

			UNIT - III			

	5	a)	What function do the following module implement?
			(i) input [31:0] x;
			input [0:4] y;
			input sel;
			output z;
			assign z = sel ? x[y] : 1'b0;
			(ii) assign d = ~(c & b);
			assign c = ~(a & d);
			(iii) assign p = q[r];
			(iv) assign q[r] =p;
			where "p", "q" and "r" are variables
				CO2	*PO2*	**06**

		b)	With truth tables, explain the following Verilog primitives: (i) bufif0 (ii) nor (iii) xor (iv) notif1	*CO2*	*PO1*	**08**
		c)	Design and write VERILOG code to implement the functionality of a 1-Bit comparator.	*CO2*	*PO2*	**06**
			OR			

	6	a)	In the Verilog code, find the values of f1, f2 and f3;
			wire [7:0] a, b, c; wire f1, f2, f3;
			assign a=4'b0111;
			assign b=4'b1100;
			assign c=4'b0100;
			assign f1= ^ a;
			assign f2= &(a^b);
			assign f3=^a&~ ^b;
				CO2	*PO2*	**06**

		b)	Derive the advantage and disadvantage of using continuous assignment statement in a Verilog module with an example.	*CO2*	*PO2*	**06**
		c)	Write a Verilog code for all the gates in different modules (any 5).	*CO2*	*PO2*	**08**
			UNIT - IV			
	7	a)	Write a behavioral model for a BCD synchronous counter with the following specification: It should be a synchronous (4-bit) up decade counter with output count that works as follows: All state changes occur on the rising edge of the CLK input, except the asynchronous reset (reset). When reset=1, the counter reset regardless of the values of the other inputs.	*CO2*	*PO2*	**06**
		b)	Realize CMOS Inverter using MOS switches and write the Verilog description for the same.	*CO2*	*PO2*	**06**
		c)	Write a Verilog model of the FSM described by the state diagram. Develop a test bench 			**08**

			OR			
	8	a)	Explain the blocking and non-blocking assignment statements with relevant examples.	*CO2*	*PO2*	**06**
		b)	Draw the circuit diagram for a XOR gate using NMOS and PMOS switches. Write the Verilog description for the same.	*CO2*	*PO2*	**06**

		c)	Analyze the following code and write the state machine
			module function (out,in,reset,clk);
			input in, clk,reset;
			output reg out;
			reg [1:0] state;
			parameter P = 2'b00, Q = 2'b01, R = 2'b10, S = 2'b11;
			
			always@(negedge clk, reset)
			begin
			if (reset) state =P;
			else
			begin
			case (state)
			P: state <= in? Q:P;
			Q: state <= in? Q:R;
			R: state <= in? S:P;
			S: state <= in? Q:R;
			endcase
			end
			end
			always@(state)
			out = (state ==S)? 1:0;
			endmodule
				CO2	*PO2*	**08**

			UNIT - V			
	9	a)	What is logic synthesis and the impact of the same?	*CO3*	*PO1*	**06**
		b)	For the following Verilog constructs what is the interpretation of the synthesis tools to translate for gate-level representation. i)The if statement ii)The case statement iii)The always statement Write a simple Verilog code to justify the above statements.	*CO3*	*PO1*	**06**
		c)	Which are the Verilog operators supported for synthesis (any 4).	*CO3*	*PO3*	**08**
			OR			
	10	a)	Explain the importance of Synthesis	*CO3*	*PO1*	**06**
		b)	Obtain the Synthesis model for the following code: module ckt (a,b,c,d,e,f,g,y); input a,b,c,d,e,f,g; output y; reg y; always@(*) begin	*CO3*	*PO1*	**06**

			if (a= =1) y=d; else
			if (b= =0) y=e; else
			if (c= =1) y=f; else
			y=g;
			end
			endmodule			

		c)	Analyze the following Verilog process and give the circuit generated by the code.
			
			module syn (clk,clr,Dout, Din,load);
			input clk,clr;
			input [3:0] Din;
			output reg [3:0] Dout;
			always @ (posedge clk or posedge clr)
			begin
			if (clr == 1'b1) Dout <= 4'b0;
			else if (load ==1'b1) Dout <=Din;
			end
			endmodule
				CO3	*PO3*	**08**

******