Sr.NO	Question
1	Take a 2 state variable and assign X and Z to the variable and display the output.
2	Take a logic type variable and use it in always block and assign block. Check the output
3	Check the difference between 4’b1000 and 4’d1000.
4	Write an example for Concatenation operator in verilog.
5	Write a task to add two arrays.
6	Write a function with return type as array.
7	enum {IDLE, START, STOP=’bx} state; Display the above enum type variable
8	enum integer {IDLE, START=’bx, STOP} state; Display the above enum type variable
9	How to come out of “forever” loop in verilog. Give an example?
10	How to come out of “forever” loop in System verilog. Give an example?
11	for(int i=9; i >=-7; i--) Using the above for loop, write an example that displays “i” value only for positive “i” values.
12	Declare a two dimensional(2D) array and initialize the arrray any values and display using verilog constructs and system verilog constru
13	How can we reduce the manual repetaed work in SV?
14	Write a simple code using pre-increment and post-increment operators?

15	integer state;
	initial
	begin
	int count=’d10;
	#10 state=’d20;
	end
	initial
	begin
	#1 $display(“count=%d”,count);
	#15 $display(“state=%d”,state);
	end
	endmodule
	Write the above program, simulate it and observe the output.

16	module eg;
	reg[1:0] state, state_next, count_next;
	always @(posedge clk)
	state<=state_next;
	always @(posedge clk)
	state<=count_next;
	endmodule
	a. Write the above program, simulate it and observe the output.
	b. Replace the above always block with always_ff and observe the output.

17	always @(*)
	begin
	if(s1)
	out=c;
	else
	if(s2)
	out=1;
	end
	endmodule
	a. Write the above program, simulate it and observe the output.
	b. Replace the above always block with always_comb and observe the output.

18	18. Write adddition of two varialbes using task and functions in verilog and system verilog .
19	19. Take single bit logic variables and compare using ==? and !=? operators. Apply all possible scenarioes.
20	a. Write code for the below state machine using Verilog constructs IDLE | START | STOP b. Write code for the above state machine using System Verilog constructs

21	typedef enum {IDLE, START,STOP, WAIT} state;
	state s1;
	always_comb
	begin
	case(s1)
	IDLE:$display(“state = %d”, s1);
	START:$display(“state = %d”, s1);
	STOP:$display(“state = %d”, s1);
	WAIT:$display(“state = %d”, s1);
	endcase
	end
	endmodule
	Execute the above program.check what will be the state value.Describe your analysis

22	module eg;
	typedef enum {IDLE, START,STOP} state;
	parameter WAIT=’d3;
	state s1;
	always_comb
	begin
	case(s1)
	IDLE : s1 = START;
	START: s1 = STOP;
	STOP : s1 = WAIT;
	endcase
	end
	endmodule
	Execute the above program.List dwon the observations.