U.S.N.										

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

Autonomous Institute Affiliated to VTU

September / October 2023 Supplementary Examinations

Programme: B.E **Semester: IV**
Branch: Electronics and Communication Engineering **Duration: 3 hrs.**
Course Code: 19EC4PCHDL **Max Marks: 100**
Course: HDL Programming

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

UNIT - I

1 a) Design a 4-to-16 decoder using 2-to-4 decoders with enable using bottom-up design methodology

i. Draw the logic block diagram. **06**

ii. Code the same in Verilog HDL. **04**

b) Design a Verilog module for a D flipflop and write a testbench to verify the functionality. Also show the expected waveforms. **10**

UNIT - II

2 a) Use only Verilog logical, ternary and shift operators to perform following operations. Assume A= 4'b1110; B=8'b11010010.

i. A divided by 2 and A multiplied by 2 **02**

ii. Arithmetic right shift on B once and Logical left shift on A twice **02**

iii. Implement 'Y' as an output that selects 'A' when S=0 and selects 'B' when S=1 **02**

b) Implement an S-R latch in Verilog using dataflow modeling. Also write a testbench to verify the functionality of the same. **06**

c) Design a Gate-level schematic for a 2-input unsigned multiplier and implement the same in Verilog using Gate-level modeling. **08**

UNIT - III

3 a) Write a task that generates output 'Y' =1 when input 'X' is high for 3 clock cycles. The output 'Y' must be high for one clock cycle. **08**

b) Design a N-Bit Parallel Adder using generate block, using the 1-bit full adder as a building block. **06**

c) Write a Verilog code for 3-bit universal shift register. **06**

OR

4 a) The table given below shows the logic to generate 'Y', where A, B, C, D are inputs:

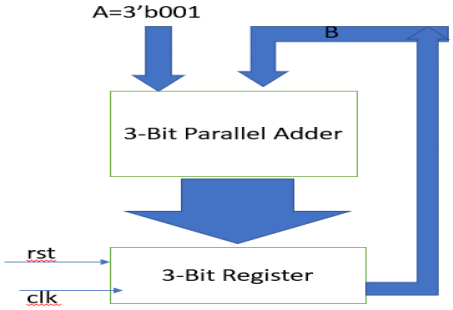
A	B	C	D	Y[2]	Y[1]	Y[0]
X	X	1	X	0	1	0
X	1	0	X	0	0	1
X	0	0	1	0	1	1
1	0	0	0	1	0	0
0	0	0	0	0	0	0

i. Write the behavioural description for the output 'Y' as per the above table using ternary operators only. **05**

ii. Implement the output 'Y' as per the above table using the appropriate case construct. **05**

iii. If the last condition (for inputs: A=0, B=0, C=0, D=0) were not specified in the above table, how would the Verilog code be written using if-else-if statements? **05**

b) For the circuit given below, write the Verilog behavioral code. **05**



UNIT - IV

5 a) Draw the equivalent logic circuit for the given code-1 and code-2. **10**

CODE-1	CODE-2
module SYNTH_CODE1(A,B,C,D);	module SYNTH_CODE1(A,B,C,D);
input A,B,C;	input A,B,C;
output reg D;	output reg D;
always@(posedge C or posedge B)	reg Q;
begin	assign D = B ? 1'b0:Q;
if(B) D <= 1'b0;	always@(posedge C)
else D <= A;	Q <= A;
end	endmodule
endmodule	

b) Observe the code below and draw the equivalent logic circuit for the same. **05**

```
module SYNTH_CODE3(A,B,Y);
input[1:0] A,B;
output[3:0] Y;
assign Y = (A-B)^2;
endmodule
```

c) For the Verilog code SYNTH_CODE3 shown below, draw the equivalent logic circuit. **05**

```
module SYNTH_CODE3(A,B,Y);
input[1:0] A,B;
output[3:0] Y;
always@(A)
begin
if(B==2'b11)
Y = A ^ 2;
else
Y = A*2;
end
endmodule
```

UNIT - V

6	a)	Briefly explain the building blocks of an FPGA with the help of a neat diagram.	**10**
	b)	Implement a half adder using 4-input Look-up Tables inside an FPGA.	**04**
	c)	Design a synthesizable Verilog module to generate the sequence 2-3-1-0-4-2.	**06**

OR

7	a)	Design a Verilog Mealy type FSM that detects 010 and 001 (with overlaps) and sets output Z=1.	**12**
	b)	With the help of a neat flowchart, briefly describe the process of implementing a digital system on an FPGA.	**08**

******