

**Fifth Semester B.E. Degree Examination, Dec.2024/Jan.2025**  
**Verilog HDL**

Time: 3 hrs.

Max. Marks: 100

*Note: Answer any FIVE full questions, choosing ONE full question from each module.*

**Module-1**

- 1 a. Explain the typical design flow for designing VLSI IC circuits, with a neat flow chart. (10 Marks)
- b. Explain the different levels of abstraction used for programming in verilog and write the example in each case. (10 Marks)

**OR**

- 2 a. Explain top-down design methodology and bottom-up design methodology with example. (10 Marks)
- b. Explain the design hierarchy using 4-bit ripple carry counter. With a block diagram of 4-bit ripple carry counter, explain the design hierarchy. (10 Marks)

**Module-2**

- 3 a. Explain the lexical convention 'sized numbers and unsized numbers'. (02 Marks)
- b. Explain the following data types with an example in verilog:  
i) Registers ii) Nets iii) Arrays iv) Integers v) Time. (10 Marks)
- c. What are system tasks and compiler directives? Explain with example. (08 Marks)

**OR**

- 4 a. With a neat block diagram, explain the components of a verilog module by highlighting mandatory blocks. (08 Marks)
- b. What are the components of SR-Latch? Write verilog HDL module of SR-Latch. (08 Marks)
- c. Write ANSI C style port declaration syntax. (04 Marks)

**Module-3**

- 5 a. With the help of logic diagram, write a verilog code for 4 to 1 multiplexer using gate-level modeling. (08 Marks)
- b. What are rise, fall and turn-off delays? Explain, how they are specified in verilog. (08 Marks)
- c. Design gate-level description for 2-to-1 multiplexer using bufif 0 and bufif 1 gates. The delay specification for these gates are as follows:

Delay	Min	Typ	Max
Rise	1	2	3
Fall	3	4	5
Turn-off	5	6	7

(04 Marks)

**OR**

- 6 a. Write a verilog data-flow level of abstraction for 4-to-1 multiplexer using i) Conditional operator ii) Logical equation. (10 Marks)

- b. What would be output of the following:

$a = 8'b10100101$        $b = 8'b10110111$

- i)  $a \& b$       ii)  $a \& \& b$       iii)  $\& b$       iv)  $a >> 1$       v)  $a >>> 1$       vi)  $y = \{2\{b\}\}$   
 vii)  $a \wedge b$       viii)  $z = \{b, a\}$       ix)  $y = a + b$       x)  $!a$       (10 Marks)

#### Module-4

- 7 a. Explain the blocking assignment statements and non-blocking assignment statements with relevant examples. (08 Marks)  
 b. Explain the following control statement syntax with an example:  
     i) if-else      ii) For (08 Marks)  
 c. Write a verilog HDL code for JK-flip flop using CASE statement. (04 Marks)

#### OR

- 8 a. Bring out the difference between task and function. (08 Marks)  
 b. Write verilog program to define a function to calculate the factorial of a 4-bit number. The output is a 32-bit value. Invoke the function by using stimulus and check results. (08 Marks)  
 c. What is task definition using ANSI c-style argument declaration? (04 Marks)

#### Module-5

- 9 a. Explain the terms force and release. (06 Marks)  
 b. Discuss the system tasks related to files. (06 Marks)  
 c. Using assign and deassign statements, design a positive edge-triggered D-flipflop with a synchronous clear ( $q = 0$ ) and preset ( $q = 1$ ). (08 Marks)

#### OR

- 10 a. With a neat flow chart explain basic computer-aided logic synthesis process. (10 Marks)  
 b. What will the following statement translate to when run on a logic synthesis tool:  
     i)  $\text{assign } (c\_out, sum) = a + b + c\_in ;$   
     ii)  $\text{assign out} = (s) ? i1 : i0 ;$   
     iii)  $\text{always } @ (clk \text{ or } d)$   
         if (clk)  
              $q = d ;$  (10 Marks)

\* \* \* \* \*