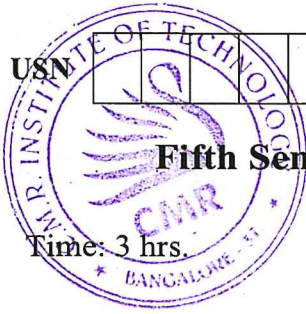


CBCS SCHEME

18EC56



Fifth Semester B.E. Degree Examination, July/August 2021

Verilog HDL

Time: 3 hrs.

Max. Marks: 100

Note: Answer any FIVE full questions.

- 1
 - a. Explain the various stages used in VLSI design with a neat flow diagram. (08 Marks)
 - b. Design a 4-bit ripple carry counter using a top-down design methodology. (08 Marks)
 - c. Compare the HDL programming to traditional software programming. (04 Marks)

- 2
 - a. Give the importance of stimulus block. Explain the different styles of stimulus block used for testing the design. (08 Marks)
 - b. Explain the different levels of abstraction. (06 Marks)
 - c. Write a pseudo verilog code for 4-bit ripple carry adder with following description.
 - i) Define a module FA with input A, B C in, sum and carry with no internals.
 - ii) Instantiate 4 full adders of the type FA in the module Ripple-Add and name them as FA0, FA1, FA2 and FA3. (06 Marks)

- 3
 - a. Illustrate with examples the data types used to define nets, registers, vectors and arrays. (08 Marks)
 - b. Differentiate i) $\$display$ and $\$monitor$ ii) $\$stop$ and $\$finish$ with examples. (06 Marks)
 - c. Declare a top-level module as TOP for stimulus. Define a constant N of size 8, IN_REG (8 bit) LOAD_EN(1-bit), LOAD_VAL (8-bit) and CLK(1bit) as register variables, and OUI_REG (8-bit) as wire. Instantiate the module shift_reg and call it as SR1. Connect the port by named list. (06 Marks)

- 4
 - a. Illustrate with example the post connection rule of verilog HDL programming. (08 Marks)
 - b. Draw the logic diagram of SR latch. Develop the verilog code for SR latch. Identify the components and hence write the test bench to verify the functionality. (08 Marks)
 - c. Declare the following variables in verilog.
 - i) Net 'A' is fixed to logic value '0' at declaration
 - ii) Vector register, Address_bus of 41 bit wide
 - iii) A memory MEM containing 256 words of 64 bit each
 - iv) An integer called count. (04 Marks)

- 5
 - a. Design a 4-bit ripple carry full adder using 1-bit full adder. Develop the verilog code for a 4-bit ripple carry full adder using gate level modeling. Verify the functionality with appropriate test bench. (08 Marks)
 - b. Given $A = 5'b10101$; $B = 5'b11101$; $C = 5'b11001$; $D = 5'b10011$. Evaluate.
 - i) $Y = A \& B$
 - ii) $Y = \sim(\& C)$
 - iii) $Y = C \wedge D$
 - iv) $Y = C \% A$
 - v) $Y = A + (D \gg \gg 1)$
 - vi) $Y = \{B[3], C[2], A\}$ (06 Marks)
 - c. Discuss the gate delays along with its types of delay specification. (06 Marks)

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg. 42+8 = 50, will be treated as malpractice.

- 6 a. Design a 4-bit ripple carry counter using TFF. Write the verilog code using data flow modeling. Verify the code with appropriate test bench. (08 Marks)
- b. Design a 2×1 MUX using bufif0 and bufif1 gates. Write the verilog code using gate level modeling for the given delay specification.

	Min	Max	Typ
Rise	1	3	2
Fall	3	5	4
Turnoff	5	7	6

- c. Discuss the types of delays used in the continuous assignment statement. (06 Marks)
- 7 a. i) Differentiate blocking and non-blocking statement with appropriate examples. (06 Marks)
- ii) Design a clock with period 40 and a duty cycle of 25% by using the always and initial statement. The value of clock at time = 0 is initialized to 0. Display the value. (08 Marks)
- b. Design a 4×1 MUX and develop a verilog code using case statement. (06 Marks)
- c. Bring out the differences and similarities between task and function. (06 Marks)

CMRIT LIBRARY
BANGALORE - 560 037

- 8 a. Compare sequential and parallel block with appropriate example. (06 Marks)
- b. Define a task to compute the parity of a 16-bit data. Write a verilog code to call task calc-parity to compute the parity. Display the message as even or odd parity. (08 Marks)
- c. Discuss the for loop and forever statement with example. (06 Marks)
- 9 a. Illustrate with examples the system tasks related to files. (06 Marks)
- b. Write a verilog program for a positive edge triggered DFF with asynchronous clear ($q = 0$) and preset ($q = 1$) using assign and deassign statements. (06 Marks)
- c. Give the importance of parameter overriding. Explain the two techniques of parameter overriding with examples. (08 Marks)
- 10 a. List the limitation of manually obtained gate level synthesis of design. How these are analyzed and addressed using automated logic synthesis tools. (08 Marks)
- b. Discuss in detail the steps involved in the logic synthesis flow from RTL to gates with a neat flow diagram. (08 Marks)
- c. Interpret the gatelevel netlist diagram for the following when run on a synthesis tool. (04 Marks)
- i) assign out = (Sel)? I1 : I0 ;
- ii) always @ (posedge clk)
- q ← d;
