Sri Adichunchanagiri Shikshana Trust®

SJB INSTITUTE OF TECHNOLOGY

Accredited by NBA & NAAC with 'A' Grade

No. 67, BGS Health & Education City, Dr. Vishnuvardhan Road
Kengeri, Bangalore – 560 060

Department of Electronics & Communication Engineering

VHDL [18EC56]

QUESTION BANK

V SEMESTER – B. E

Academic Year: 2021 – 2022 (ODD)

Course Coordinator :	**Mrs Latha S**
Designation :	**Assistant Professor**

NOTE: Programs and Exercise examples which are there in notes are important

MODULE 1

1. Discuss in brief about the evolution of CAD tools and HDLs used in digital system design
2. Explain the typical VLSI IC design flow with the help of flow chart.
3. Discuss the trends in HDLs?
4. Why Verilog HDL has evolved as popular HDL in digital circuit design?
5. Explain the advantages of using HDLs over traditional schematic based design.
6. Describe the digital system design using hierarchical design methodologies.
7. Apply the top-down design methodology to demonstrate the design of ripple carry counter.
8. Apply the bottom-up design methodology to demonstrate the design of 4-bit ripple carry adder.
9. Write Verilog HDL program to describe the 4-bit ripple carry counter.
10. Define Module and an Instance. Describe 4 different description styles of Verilog HDL
11. Differentiate simulation and synthesis. What is stimulus?
12. Write a test bench to test the 4-bit ripple carry counter.
13. Write a test bench to test the 4-bit ripple carry adder.
14. **Exercise examples and Programs**

MODULE 2

1. Describe the lexical conventions used in Verilog HDL with examples.
2. Explain different data types of Verilog HDL with examples
3. What are system tasks and compiler directives?
4. What are the uses of $monitor, $display and $finish system tasks? Explain with examples.
5. Explain `define, `timescale and `include compiler directives.
6. Explain the components of Verilog HDL module.
7. What are the components of SR latch? Write Verilog HDL module of SR latch.
8. Explain the different types of ports supported by Verilog HDL withexamples.
9. Explain the port connection rules of Verilog HDL with examples.
10. How hierarchical names helps in addressing any identifier used in the design from any other level of hierarchy? Explain with examples.
11. What are the basic components of a module? Which components are mandatory?
12. **Exercise examples and Programs**

MODULE 3

1. Write the truth table of all the basic gates. Input values consisting of '0', '1', 'x', 'z'.
2. What are the primitive gates supported by Verilog HDL? Write the Verilog HDL statements to instantiate all the primitive gates.
3. Use gate level description of Verilog HDL to design 4 to 1 multiplexer. Write truth table, top-level block, logic expression and logic diagram. Also write the stimulus block for the same.
4. Explain the different types of buffers and not gates with the help of truth table, logic symbol, logic expression (bufif, buf, not, notif).
5. Use gate level description of Verilog HDL to describe the 4-bit ripple carry counter. Also write a stimulus block for 4-bit ripple carry adder.
6. How to model the delays of a logic gate using Verilog HDL? Give examples. Also explain the different delays associated with digital circuits.

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

7. Write gate level description to implement function y = a.b + c, with 5 and 4 time units of gate delay for AND and OR gate respectively. Also write the stimulus block and simulation waveform.
8. With syntax describe the continuous assignment statement.
9. Show how different delays associated with logic circuit are modelled using dataflow description.
10. Explain different operators supported by Verilog HDL.
11. What is an expression associated with dataflow description? What are the different types of operands in an expression?
12. Discuss the precedence of operators.
13. Use dataflow description style of Verilog HDL to design 4:1 multiplexer with and without using conditional operator.
14. Use dataflow description style of Verilog HDL to design 4-bit adder using
    i. Ripple carry logic.
    ii. Carry look ahead logic.
15. Use dataflow description style, gate level description of Verilog HDL to design 4-bit ripple carry counter. Also write the stimulus block to verify the same.
16. **Exercise examples and Programs**

MODULE 4

1. Describe the following statements with an example: initial and always
2. What are blocking and non-blocking assignment statements? Explain with examples.
3. With syntax explain conditional, branching and loop statements available in Verilog HDL behavioural description.
4. Describe sequential and parallel blocks of Verilog HDL.
5. Write Verilog HDL program of 4:1 mux using CASE statement.
6. Write Verilog HDL program of 4:1 mux using If-else statement.
7. Write Verilog HDL program of 4-bit synchronous up counter.
8. Write Verilog HDL program of 4-bit asynchronous down counter.
9. Write Verilog HDL program to simulate traffic signal controller.
10. **Exercise examples and Programs.**

MODULE 5

1. Discuss the evolution of VHDL.
2. List and explain the advantages and benefits of using VHDL?
3. In brief discuss the shortcomings of VHDL.
4. Explain the digital system synthesis using VHDL in detail.
5. With tool flow diagram explain design tool flow.
6. Explain the components of VHDL program.
7. Define entity. Explain different types of ports used in VHDL entity.
8. Explain different description styles supported by VHDL with example.
9. Compare different description styles of VHDL with examples.
10. Explain the different data objects, data types of VHDL.
11. What are attributes? Explain with examples.
12. Differentiate between signal assignment and variable assignment statements.
13. Write the entity declaration 2-bit magnitude comparator.

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060

Exercise Examples

1: An interconnect switch (IS) contains the following components, a shared memory (MEM), a system controller (SC) and a data crossbar (Xbar).

a. Define the modules MEM, SC, and Xbar, using the module/endmodule keywords. You do not need to define the internals. Assume that the modules have no terminal lists.

b. Define the module IS, using the module/endmodule keywords. Instantiate the modules MEM, SC, Xbar and call the instances mem1, sc1, and xbar1, respectively. You do not need to define the internals. Assume that the module IS has no terminals.

c. Define a stimulus block (Top), using the module/endmodule keywords. Instantiate the design block IS and call the instance is1. This is the final step in building the simulation environment.

2: A 4-bit ripple carry adder (Ripple_Add) contains four 1-bit full adders (FA).

a. Define the module FA. Do not define the internals or the terminal list.

b. Define the module Ripple_Add. Do not define the internals or the terminal list. Instantiate four full adders of the type FA in the module Ripple_Add and call them fa0, fa1, fa2, and fa3.

3: A 4-bit parallel shift register has I/O pins as shown in the figure below. Write the module definition for this module shift_reg. Include the list of ports and port declarations. You do not need to show the internals.

4: What are the basic components of a module? Which components are mandatory?

5. Declare a top-level module stimulus. Define REG_IN (4 bit) and CLK (1 bit) as reg register variables and REG_OUT (4 bit) as wire. Instantiate the module shift_reg and call it sr1. Connect the ports by orderedlist.

6. A 2-input xor gate can be built from my_and, my_or and my_not gates. Construct an xor module in Verilog that realizes the logic function, z = xy' +x'y. Inputs are x and y, and z is the output. Write a stimulus module that exercises all four combinations of x and y inputs.

7. The 1-bit full adder described in the chapter can be expressed in a sum of products form.
sum = a.b.c_in + a'.b.c_in' + a'.b'.c_in + a.b'.c_in'
c_out = a.b + b.c_in + a.c_in
Assuming a, b, c_in are the inputs and sum and c_out are the outputs, design a logic circuit to implement the 1-bit full adder, using only and, not, and or gates. Write the Verilog description for the circuit. You may use up to 4-input Verilog primitive and and or gates. Write the stimulus for the full adder and check the functionality for all input combinations.

8. A full subtractor has three 1-bit inputs x, y, and z (previous borrow) and two 1-bit outputs D (difference) and B (borrow). The logic equations for D and B areas follows:

Head
Dept. of Electronics & Communication Engg.
SJB Institute of Technology
Bengaluru-560060