BCS302 SIMP 2025- by RVITM review team

Module 2

1. Design a combinational circuit to convert BCD to Excess-3 or any other specified design.
2. Design a 4-bit parallel adder/subtractor circuit.
3. Explain the carry look-ahead adder with a block diagram and briefly describe a binary subtractor.
4. Explain the half adder and full adder with truth tables and logic diagrams, including designing a full adder using two half adders.
5. Define a decoder and construct a 2-to-4 line decoder with an enable input, including the truth table.
6. Design a modulo-10 synchronous counter using JK flip-flops and differentiate between asynchronous and synchronous counters.
7. Define a multiplexer and explain an 8:1 MUX with a logic diagram and logic expression. Use it to implement $F(A,B,C,D)=\Sigma m(1,2,5,6,9,12)F(A, B, C, D) = \Sigma m(1,2,5,6,9,12)$.
8. Explain the operation of SR, JK, T, and D flip-flops, focusing on a positive edge-triggered D flip-flop. Include characteristic and truth tables.
9. Write Verilog HDL code for (i) a 2-to-1 multiplexer, (ii) a 2-to-4 decoder, and (iii) a full adder using a half-adder module.

Module 3

1. Explain the basic operational concepts of a computer with a neat diagram, including the processor registers and the sequence of steps involved.
2. Compare and contrast RISC and CISC architectures, highlighting their key differences, trade-offs, and the role of instruction sequencing and branching with examples.
3. Analyze Big-Endian and Little-Endian methods of byte addressing with examples, and explain addressing modes with examples.
4. Differentiate between static RAM (SRAM) and dynamic RAM (DRAM), focusing on their operation and applications, and explain the concept of memory interleaving and its benefits in improving memory access speed.
5. Write a program to evaluate e Y= A*B+C*D or A+B*C+D using one-address, two-address, three-address, and zero-address instructions, and explain the effective address computation for various instruction formats.
6. Describe the characteristics of serial communication interfaces (e.g., RS-232), their applications, and the role of processor clock, clock rate, and the basic performance equation in performance measurement.
7. What is performance measurement? Explain the overall SPEC rating and discuss the differences between saving return addresses in processor registers, memory locations, and on a stack, including their support for subroutine nesting and recursion.

8. Registers R1 and R2 of a computer contain the decimal values 1200 and 4600. What is the effective- address of the memory operand in each of the following instructions? (a) Load 20(R1), R5 (b) Move #3000, R5 (c) Store R5,30(R1, R2) (d) Add -(R2), R5 (e) Subtract (R1)+, R5

Module 4

1. Differentiate between memory-mapped I/O and I/O-mapped I/O.
2. Explain methods for handling multiple interrupts raised by devices, using priority structures.
3. Define and describe interrupts and interrupt hardware, including enabling and disabling interrupts.
4. Explain the operation of a DMA controller with a block diagram, and define DMA bus arbitration (centralized and distributed).
5. Define exceptions and describe different types of exceptions.
6. Explain cache memory and different mapping functions with diagrams.
7. Discuss the memory hierarchy in a computer system, highlighting variations in size, speed, and cost per bit.
8. Explain input/output operations by the processor with an example program for reading and displaying characters.

Module 5

1. Write and explain the control sequence for executing an ADD instruction on a single-bus processor.
2. Explain the 3-bus organization of a data path with a diagram.
3. Explain the organization of a single-bus processor data path with fundamental concepts.
4. Briefly explain register transfer, ALU operation, and the process of fetching and storing words in memory.
5. Discuss hazards in pipelining (types, examples), pipeline performance, and the role of cache in pipelining.
6. Define and explain stalls or bubbles in a pipeline with an example.
7. Write a note on fetching a word from memory and storing it with a supporting diagram.
8. Explain pipeline hazards, types of stalls, and methods to address them.

Module 1

1. Simplify Boolean expressions using four-variable K-maps and implement them using NAND and NOR gates.

    - $F(A,B,C,D)=AD'+B'C'D+BCD'+BC'DF(A, B, C, D) = AD' + B'C'D + BCD' + BC'D$
    - $F(A,B,C,D)=\pi M(1,2,3,7,13,15)F(A, B, C, D) = \pi M(1, 2, 3, 7, 13, 15)$
2. Implement Boolean expressions using NAND and NOR gates: $F(x,y,z)=\Sigma(0,6,8,13,14)+d(1,3,10)F(x, y, z) = \Sigma(0, 6,8,13,14) + d(1, 3, 10)$.
3. Design multi-level NOR and NAND circuits for $F=CD(B+C)A+(BC'+DE')F = CD(B + C)A + (BC' + DE')$.
4. Prove Boolean identities like DeMorgan's Theorem using truth tables
5. Design a multiple-level logic circuit for $F(w,x,y,z)=\Sigma m(0,2,4,5,6,7,8,10,13,15)F(w, x, y, z) = \Sigma m(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$.
6. Draw and explain the block diagram of a multiplexer and its application in Boolean function realization, also sketch 4-bit parity generator
7. Implement a full adder circuit using NAND gates and explain its operation with a truth table.
8. Simplify and implement expressions using logic gates: $F=AB+A'B'+CF = AB + A'B' + C$.
9. Simplify the Boolean function F(w,x,y,z) using the Quine-McCluskey algorithm.

Additional Questions for practice – Only if/after you complete the above SIMP

Module 1: Introduction to Digital Design

1. Compare the Espresso algorithm and K-maps for Boolean function minimization, mentioning advantages and disadvantages.
2. Compare TTL and CMOS logic families in terms of power consumption, speed, and noise immunity.
3. Explain the concept of fan-out and its importance in digital circuit design.
4. Explain how Hamming code detects and corrects single-bit errors in data transmission.

Module 2: Combinational & Sequential Logic

1. Design an 8-to-3 priority encoder and explain its working.
2. Design a circuit to convert a 4-bit binary number to its Gray code equivalent.
3. Analyze a sequential circuit for a given input- TYPE
4. Design a 4-bit universal shift register capable of left shift, right shift, and parallel load operations.

Module 3: Basic Structure of Computers

1. Describe the characteristics and applications of the serial communication interface (e.g., RS-232).

Module 4: Input/Output Organization

1. Explain the role of a programmable interrupt controller (PIC) in managing multiple interrupts -VBQ
2. Write a program to implement interrupt-driven I/O for data transfer.
3. Compare PCI and ISA bus architectures in terms of data transfer rates, addressing, and applications.

Module 5: Basic Processing Unit

1. List and explain the micro-operations involved in fetching and decoding an instruction.
2. Compare hardwired and microprogrammed control units, highlighting their pros and cons.