Module – 1

1. Optimize the Boolean function, F (A, B, C, D) =m0+ml+m7+m13+m15 with don't care conditions d(A, B, C, D) = m2+m6+m8+m9+m10 using K-Map.
2. What is a multiplexer? Using an 8-to-1 MUX, implement the Boolean function F(A,B,C,D)=Σ (1,3,4,11,12,13,14,15)
3. Design an Octal to Binary encoder with the help of a truth table and logic (6) diagram.
4. Reduce the Boolean function specified in the truth table to its minimum SOP form by using a K-Map

Inputs			Output
X	Y	Z	S
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

5. Minimize the Boolean expression f(A,B,C,D)-2m(1,5,6,7,9,15)+d(2,3,11,13) (6) using Karnaugh map and realize it using Logic gates
6. Convert the decimal number 3.257 10^4 into IEEE-754 Single Precision (6) Floating Point binary representation
7. Minimize the Boolean expression f(A,B,C,D)=(0,1,3,5,7,8,9,11,13,15)using Karnaugh map and realize it using NAND gate.
8. Using Boolean algebra techniques, simplify the expression AB+A(B+C)+B(B+C).Express +19 and -19 in 2's complement form

Module – 2

1. A) Mention any four applications of shift registers. B) Describe the working of a Parallel in Serial Out register.
2. Draw the state diagram and logic diagram of a 3 bit up down synchronous counter
3. Explain the working of an edge triggered SR Flip-Flop with the help of a circuit (6) and timing diagram.
4. What are the basic functions of a shift register? Explain 4 bit PISO shift register (6) with a neat diagram

5. Construct a 3-bit Up/Down Synchronous Counter Show the relevant Hoolean (6) expressions
6. Impliment and explain the working of a 4-bit Parallel-In Serial-Out [PISO] (6) Shitter.
7. Explain the working of an edge triggered SR Flip flop in detail.
8. Design a 3 bit UP/DOWN synchronous counter

Module – 3

1. Explain the five classic components of a computer with diagram
2. Calculate and Compare the average execution time between instructions of a non-pipelined implementation to a pipelined implementation. The operation times for the major functional units are 200ps for memory access, 200ps for ALU operation, and 100ps for register file read or write. Consider the 3 consecutive load instructions. (Eg: lw $t1,100($t2)
3. Explain classic components of a computer system with the help of a block (0) diagrum
4. What is Addressing Mode? Describe any three Addressing Modes with (6) examples.
5. List down and briefly explain the 8 great ideas in Computer Architecture
6. Define Addressing mode. Explain 5 Addressing modes with examples
7. What you meant by addressing modes? Explain any three addressing modes (6) that have been used in recent computers
8. Explain the five classic components of a computer with figure

Module – 4

1. What is Direct Memory Access? Explain two types of bus arbitration schemes
2. List different types of pipeline hazards with examples
3. With a neat diagram, explain the operation of DMA controllers in a computer (6) system
4. What is pipeline hazard? Explain any three hazards in pipelined processors with (6) examples
5. Draw the Single Cycle Datapath for implementing Memory Reference (6) instructions and R-Format instructions.
6. Write notes on: Direct Memory Access & Interrupt Handling
7. Draw a single data path representation for R-type instruction
8. How should two or more simultaneous interrupt requests be handled? Explain (6) with figure

Module – 5

1. What is virtual memory? Explain the process of address translation
2. Illustrate the different mapping methods of Cache Memory
3. Explain virtual memory address translation using page table with the help of a (6) neat diagram
4. What is cache memory? Explain different cache mapping techniques with (6) examples
5. Explain the various Cache Mapping Techniques
6. A)What do you understand by Virtual Memory B)Construct a IKB Memory IC using 1024x4 Memory chips.
7. How the virtual address is converted into real address in a paged virtual memory system? Explain
8. A computer system uses 16-bit memory addresses. It has a 2K-byte cache organized in a direct mapped manner with 64 bytes per cache block. Assume that the size of each memory word is I byte. Calculate the number of bits in each of the Tag, Block, and Word fields of the memory address