Short Questions

1. How is overflow detected in computer arithmetic?
2. What are the key differences between binary and decimal arithmetic units?
3. Define the term "mantissa" in floating-point representation.
4. Discuss the concept of guard, round, and sticky bits in floating-point rounding.
5. Why is binary-coded decimal (BCD) used in decimal arithmetic operations?
6. Compare the advantages and disadvantages of fixed-point and floating-point representation.
7. Describe the significance of the IEEE 754 standard in floating-point representation.
8. How does the computer handle negative numbers in binary arithmetic?
9. Explain the concept of normalization in fixed-point representation.
10. What is the purpose of the least significant bit (LSB) in binary representation?
11. Discuss the limitations of fixed-point arithmetic in numerical precision.
12. How are carry and overflow related in computer arithmetic?
13. Explain the role of a carry look-ahead adder in addition operations.
14. What is the role of the quotient and remainder in division algorithms?
15. Discuss the challenges of implementing division in hardware.
16. Define the term "rounding error" in floating-point arithmetic.
17. How does the computer handle denormalized numbers in floating-point representation?
18. Explain the process of converting decimal numbers to binary.
19. Why is radix complement used in binary subtraction?

20. Discuss the impact of precision on numerical computations in floating-point arithmetic.

21. What is the role of guard bits in rounding floating-point numbers?

22. How is the normalization process different in fixed-point and floating-point representation?

23. Define the term "floating-point exception" in computer arithmetic.

24. Discuss the trade-offs between hardware complexity and performance in arithmetic units.

25. Explain the concept of round-to-nearest in floating-point rounding.

26. What is the significance of Input-Output Interface in computer systems?

27. Explain the concept of Asynchronous data transfer.

28. Differentiate between Synchronous and Asynchronous data transfer.

29. List and explain the Modes of Transfer in Input-Output Organization.

30. What is Priority Interrupt, and how does it affect system operation?

31. Define Direct Memory Access (DMA) and its role in data transfer.

32. How does DMA differ from programmed I/O in terms of data transfer?

33. Explain the concept of Memory Hierarchy in computer systems.

34. Define Main Memory and its primary function.

35. What is the purpose of Auxiliary Memory in computer architecture?

36. Differentiate between Main Memory and Auxiliary Memory.

37. How does Associative Memory differ from Main Memory?

38. Explain the role and importance of Cache Memory in a computer system.

39. List the types of Cache Memory and briefly describe each.

40. What are the advantages of using Cache Memory in a computer system?

41. Discuss the trade-offs associated with the use of Cache Memory.

42. How does Cache Memory contribute to improving system performance?

43. Describe the concept of Input-Output Interface in computer architecture.

44. Discuss the importance of a well-designed Input-Output Interface.

45. Explain the term "interrupt" in the context of computer systems.

46. How does interrupt handling contribute to efficient system operation?

47. What is the primary purpose of Asynchronous data transfer in I/O operations?

48. Compare and contrast Asynchronous and Synchronous data transfer.

49. Enumerate the different Modes of Transfer in Input-Output Organization.

50. How does Priority Interrupt enhance system responsiveness?

51. Discuss the challenges associated with Priority Interrupt handling.

52. Explain the significance of Direct Memory Access in data transfer.

53. Differentiate between memory hierarchy and memory organization.

54. Why is Memory Hierarchy essential for computer systems?

55. Describe the role of Main Memory in storing program instructions.

56. What are the characteristics of Auxiliary Memory?

57. How does Auxiliary Memory complement Main Memory in a computer system?

58. Define Associative Memory and provide an example of its use.

59. Discuss the advantages of using Associative Memory in specific applications.

60. How does the organization of Cache Memory contribute to speed improvement?

61. Explain the role of Cache Memory in reducing memory access time.

62. Enumerate the key features of a well-designed Input-Output Interface.

63. What factors influence the choice of a specific data transfer mode?

64. Describe the challenges associated with Asynchronous data transfer.

65. How does Priority Interrupt affect the execution flow of a computer system?

66. Discuss the role of Direct Memory Access in offloading the CPU during data transfer.

67. What is the purpose of prioritizing interrupts in system design?

68. Compare the benefits and drawbacks of various Modes of Transfer.

69. How does Cache Memory help in overcoming the speed difference between the CPU and main memory?

70. Explain the concept of memory mapping in the context of Input-Output Interface.

71. How does memory organization impact the overall performance of a computer system?

72. Discuss the role of Main Memory in providing fast access to frequently used data.

73. What are the common challenges associated with Auxiliary Memory systems?

74. Explain the concept of virtual memory in the context of Memory Hierarchy.

75. How does Cache Memory contribute to reducing bus contention in a computer system?

76. What does CISC stand for, and what are its characteristics?

77. Define RISC and list its key characteristics.

78. Explain the concept of pipeline processing.

79. Differentiate between parallel processing and pipelining.

80. What is an arithmetic pipeline, and how does it improve processing efficiency?

81. Define an instruction pipeline and its role in instruction execution.

82. How does RISC architecture implement a pipeline?

83. Describe the key features of vector processing.

84. Explain the role of an array processor in vector processing.

85. What are the characteristics of multiprocessors?

86. Define and explain the importance of interconnection structures in multiprocessor systems.

87. How is interprocessor arbitration handled in multiprocessor systems?

88. What is the significance of interprocessor communication in multiprocessor architecture?

89. Define and explain the concept of cache coherence.

90. How does cache coherence impact the performance of multiprocessor systems?

91. Differentiate between CISC and RISC instructions.

92. Why is pipelining considered an effective technique for instruction execution?

93. How does vector processing contribute to improved computational performance?

94. Describe the key components of an arithmetic pipeline.

95. What challenges are associated with implementing pipelining in computer architecture?

96. Explain the role of interprocessor synchronization in multiprocessor systems.

97. How does a CISC instruction set differ from a RISC instruction set?

98. Why is reducing the number of instructions per program crucial for RISC architecture?

99. Discuss the advantages of parallel processing in computer systems.

100. What are the primary goals of a vector processor?

101. How does pipelining enhance instruction throughput in a processor?

102. Define and provide examples of complex instructions in CISC architecture.

103. Discuss the impact of interprocessor communication on overall system performance.

104. Explain the concept of cache coherence protocols.

105. How does vector processing improve the efficiency of numerical computations?

106. Describe the role of an array processor in scientific and engineering applications.

107. Compare and contrast the characteristics of CISC and RISC architectures.

108. What is the purpose of interprocessor arbitration in multiprocessor systems?

109. How does cache coherence contribute to data consistency in multiprocessor environments?

110. Discuss the challenges of implementing vector processing in modern computing systems.

111. Define superscalar architecture and its relationship to instruction parallelism.

112. Explain the concept of instruction-level parallelism.

113. What factors contribute to the efficiency of an arithmetic pipeline?

114. Describe the impact of cache size on cache coherence in multiprocessor systems.

115. How do interconnection networks affect the scalability of multiprocessor systems?

116. Discuss the trade-offs between pipelining and parallel processing in computer architecture.

117. Explain the role of interprocessor synchronization in preventing race conditions.

118. How does the instruction pipeline in RISC architecture differ from CISC?

119. Define and provide examples of vector operations in computing.

120. Discuss the challenges of maintaining cache coherence in a multiprocessor environment.

121. What are the benefits of using an array processor for matrix computations?

122. Explain the importance of cache memory in modern computer architectures.

123. How does the concept of pipelining relate to the fetch-decode-execute cycle?

124. Compare and contrast the scalability of different interconnection structures in multiprocessor systems.

125. Discuss the impact of cache misses on the performance of multiprocessor systems.