Short Questions

1. What is the purpose of the sign bit in binary numbers?
2. How is division performed using the restoring division algorithm?
3. What is a non-restoring division algorithm and how does it work?
4. Explain how multiplication is handled in floating-point arithmetic.
5. What are guard digits and how do they improve accuracy in arithmetic operations?
6. Describe the algorithm for adding two floating-point numbers.
7. How does a computer handle underflow and overflow in floating-point operations?
8. What is the difference between pre-normalization and post-normalization in floating-point arithmetic?
9. How is subtraction of floating-point numbers performed differently from addition?
10. Describe the steps in converting a decimal number to floating-point format.
11. Explain the role of bias in the exponent of floating-point numbers.
12. What is the significance of the hidden bit in normalized floating-point numbers?
13. How do decimal arithmetic operations handle fractions differently from binary operations?
14. Describe the algorithm for subtracting two floating-point numbers.
15. What factors affect the speed of arithmetic operations in computers?
16. How is the modulo operation performed in computer arithmetic?
17. Explain the role of carry and borrow in addition and subtraction, respectively.
18. What is BCD (Binary-Coded Decimal) and how is it used in decimal arithmetic operations?
19. How does the IEEE 754 standard define special values like NaN (Not a Number)?
20. What are the benefits of using hardware accelerators for decimal arithmetic operations?
21. How do rounding modes affect the result of floating-point arithmetic operations?
22. Describe the significance of the alignment step in floating-point addition or subtraction.
23. What is the impact of exponent mismatch on floating-point arithmetic?
24. How does the algorithm for multiplication of two floating-point numbers ensure accuracy?
25. Explain the concept of scalable precision in floating-point arithmetic operations.
26. What is input-output organization?
27. Define the input-output interface.

28. Explain asynchronous data transfer.
29. What are the modes of data transfer?
30. Describe the concept of priority interrupt.
31. What is Direct Memory Access (DMA)?
32. How is memory organized in a computer system?
33. Explain the concept of memory hierarchy.
34. What is the role of main memory?
35. What is auxiliary memory?
36. Define associate memory.
37. How does cache memory work?
38. What are the types of input-output interfaces?
39. What advantages do asynchronous data transfers offer?
40. How do priority interrupts improve system performance?
41. Describe a scenario where DMA is used.
42. How does the memory hierarchy enhance computer performance?
43. What differentiates main memory from auxiliary memory?
44. In what situations is associate memory used?
45. What is the primary function of cache memory?
46. How do synchronous and asynchronous data transfers differ?
47. What is the importance of the modes of transfer in I/O operations?
48. How does an interrupt system work?
49. What benefits does Direct Memory Access provide?
50. How does memory hierarchy impact access speed?
51. Compare and contrast main memory and auxiliary memory.
52. How is associate memory different from other types of memory?
53. Why is cache memory crucial for modern computing?
54. What challenges are addressed by input-output organization?
55. How do interfaces facilitate data transfer between devices?
56. Describe the process of asynchronous data transfer.
57. What factors influence the selection of a data transfer mode?
58. How are devices prioritized in a priority interrupt system?
59. What is the impact of DMA on processor performance?
60. How does the concept of memory hierarchy optimize storage?
61. What are the characteristics of main memory?
62. What types of storage are considered auxiliary memory?
63. In what ways is associate memory unique?
64. How does cache memory improve processing speed?
65. What is the role of input-output interfaces in device communication?
66. How do asynchronous transfers handle data timing issues?
67. What makes a mode of transfer more suitable for certain tasks?
68. Why are priority interrupts necessary in multitasking environments?
69. How does DMA facilitate high-speed data transfers?
70. What principles underlie the memory hierarchy structure?

71. How does main memory differ from cache memory in terms of speed?
72. What is the significance of auxiliary memory in data storage?
73. How is associate memory implemented in search operations?
74. What strategies are used to optimize cache memory efficiency?
75. How do modern computers integrate various memory types effectively?
76. What are the main characteristics of CISC architecture?
77. How does RISC architecture differ from CISC?
78. What is the significance of the RISC approach in modern computing?
79. Can you list some advantages of CISC architecture?
80. What advantages does RISC architecture offer over CISC?
81. What is meant by "pipeline" in computer architecture?
82. How does parallel processing enhance computing performance?
83. Describe the concept of pipelining in CPU design.
84. What is an arithmetic pipeline, and how does it function?
85. Explain the role of an instruction pipeline in a CPU.
86. How is a RISC pipeline different from a conventional pipeline?
87. Define vector processing in the context of computer architecture.
88. What is an array processor, and how does it relate to vector processing?
89. How do multiprocessors improve computing performance?
90. What are the key characteristics of multiprocessor systems?
91. Describe different interconnection structures used in multiprocessor systems.
92. What is interprocessor arbitration, and why is it important?
93. How do processors in a multiprocessor system communicate and synchronize?
94. What is cache coherence, and why is it crucial in multiprocessor systems?
95. How does pipelining affect the throughput of a computer system?
96. What challenges arise in implementing effective pipelining in CPUs?
97. How does vector processing differ from scalar processing?
98. What types of applications benefit most from vector processing?
99. How do array processors facilitate parallel processing?
100. In what ways can pipelining be optimized in a RISC architecture?
101. What strategies are used to achieve cache coherence in multiprocessor systems?
102. How does the choice of interconnection structure impact a multiprocessor system's performance?
103. What role does interprocessor communication play in the efficiency of multiprocessor systems?
104. How is synchronization achieved between processors in a multiprocessor system?
105. Why is interprocessor arbitration necessary in a multiprocessor environment?

106. What benefits do multiprocessor systems offer for data-intensive applications?
107. How do RISC characteristics influence pipeline design?
108. What makes vector processing suited for scientific computations?
109. Can CISC architectures efficiently implement pipelining?
110. What are the challenges associated with parallel processing?
111. How does instruction pipelining improve CPU performance?
112. What is the impact of arithmetic pipelining on computational tasks?
113. How do multiprocessor systems handle data sharing and communication?
114. What are the common methods for ensuring cache coherence?
115. How do interconnection structures affect data transfer rates in multiprocessor systems?
116. What are the advantages of using an array processor for data parallel tasks?
117. How does a RISC pipeline enhance the execution of instructions?
118. What factors influence the design of a multiprocessor system?
119. How is data synchronization managed in complex multiprocessor systems?
120. What techniques are used for interprocessor arbitration?
121. How do vector processors handle multiple data elements simultaneously?
122. What are the key considerations in designing an efficient instruction pipeline?
123. How do multiprocessor systems achieve efficient parallel processing?
124. What are the challenges in maintaining cache coherence in a multiprocessor system?
125. How do the characteristics of RISC and CISC architectures affect their use in various applications?