

Rechnerorganisation

1. Definition: Computer Organization bezieht sich auf die Betriebsstruktur und Implementierung von Computersystemen, mit dem Fokus darauf, wie Hardwarekomponenten zusammenarbeiten, um Anweisungen auszuführen.
2. Historische Entwicklung: Verfolgt die Entwicklung von frühen mechanischen Computern bis hin zu modernen Mehrkernprozessoren.
3. Von-Neumann-Architektur: Ein grundlegendes Modell, bei dem die CPU, der Speicher und die Eingabe/Ausgabe über einen Bus miteinander verbunden sind.
4. Harvard-Architektur: Trennt Speicher- und Signalwege für Anweisungen und Daten, wodurch die Leistung verbessert wird.
5. CPU-Komponenten: Enthält die Arithmetisch-Logische Einheit (ALU), die Steuerungseinheit (CU) und Register.
6. ALU-Funktionen: Führt arithmetische und logische Operationen wie Addition, Subtraktion, UND, ODER durch.
7. Rolle der Steuerungseinheit: Leitet den Betrieb des Prozessors, indem sie Anweisungen decodiert und Steuersignale erzeugt.
8. Register: Kleine, schnelle Speicherorte innerhalb der CPU, die verwendet werden, um Daten und Anweisungen vorübergehend zu halten.
9. Cache-Speicher: Hochgeschwindigkeits-Speicher, der sich in der Nähe der CPU befindet, um die Datenzugriffszeit zu verkürzen.
10. Speicherhierarchie: Organisiert den Speicher in Ebenen basierend auf Geschwindigkeit und Kosten, einschließlich Register, Cache, RAM und Sekundärspeicher.
11. RAM (Random Access Memory): Flüchtiger Speicher, der zum Speichern von Daten und Maschinencode verwendet wird, die derzeit verwendet werden.
12. ROM (Read-Only Memory): Nicht-flüchtiger Speicher, der zur Speicherung von Firmware und Systemstartanweisungen verwendet wird.
13. Bus-Struktur: Ein Kommunikationssystem, das Daten zwischen Komponenten innerhalb oder außerhalb eines Computers überträgt.
14. Datenbus: Überträgt die tatsächlich verarbeiteten Daten.
15. Adressbus: Überträgt Informationen darüber, wohin Daten gesendet oder abgerufen werden sollen.
16. Steuerbus: Überträgt Steuersignale von der CPU zu anderen Komponenten.
17. Befehlssatzarchitektur (ISA): Definiert den Satz von Anweisungen, die eine CPU ausführen kann.

18. RISC (Reduced Instruction Set Computing): Eine ISA-Designphilosophie, die einen kleinen, hochoptimierten Satz von Anweisungen verwendet.
19. CISC (Complex Instruction Set Computing): Eine ISA mit einem großen Satz von Anweisungen, von denen einige komplexe Aufgaben ausführen können.
20. Pipelining: Eine Technik, bei der mehrere Anweisungsphasen überlappt werden, um die CPU-Durchsatzrate zu verbessern.
21. Pipeline-Stufen: Enthält in der Regel Fetch, Decode, Execute, Memory Access und Write Back.
22. Gefahren im Pipelining: Probleme wie Datengefahren, Steuergefahren und strukturelle Gefahren, die den Pipeline-Fluss stören können.
23. Verzweigungsvorhersage: Eine Methode, um die Richtung von Verzweigungsanweisungen zu erraten, um die Pipeline voll zu halten.
24. Superskalare Architektur: Ermöglicht es mehreren Anweisungen, gleichzeitig in einer einzigen Pipeline-Stufe verarbeitet zu werden.
25. Parallelverarbeitung: Nutzung mehrerer Prozessoren oder Kerne, um Anweisungen gleichzeitig auszuführen.
26. Mehrkernprozessoren: CPUs mit mehreren Verarbeitungskernen, die in einem einzigen Chip integriert sind.
27. SIMD (Single Instruction, Multiple Data): Eine Parallelverarbeitungsarchitektur, bei der eine einzelne Anweisung gleichzeitig auf mehrere Datenpunkte wirkt.
28. MIMD (Multiple Instruction, Multiple Data): Eine parallele Architektur, bei der mehrere Prozessoren unterschiedliche Anweisungen auf unterschiedlichen Daten ausführen.
29. Speicherverwaltung: Techniken zur effizienten Verwaltung und Zuweisung von Speicher, einschließlich Paging und Segmentierung.
30. Virtueller Speicher: Erweitert den physischen Speicher auf den Festplattenspeicher, sodass Systeme größere Arbeitslasten bewältigen können.
31. Paging: Teilt den Speicher in Seiten fester Größe, um die Speicherverwaltung zu vereinfachen und die Fragmentierung zu reduzieren.
32. Segmentierung: Teilt den Speicher in Segmente variabler Größe basierend auf logischen Aufteilungen wie Funktionen oder Datenstrukturen.
33. Cache-Mapping-Techniken: Enthält direkt gemappte, vollständig assoziative und set-assoziative Caches.
34. Cache-Ersetzungsrichtlinien: Bestimmt, welchen Cache-Eintrag zu ersetzen ist, wie z.B. Least Recently Used (LRU) oder First-In-First-Out (FIFO).

35. Cache-Kohärenz: Stellt die Konsistenz der in mehreren Caches gespeicherten Daten in einem Multiprozessorsystem sicher.
36. Speicherkonsistenzmodelle: Definiert die Reihenfolge, in der Operationen erscheinen, um die Systemkonsistenz aufrechtzuerhalten.
37. Eingabe/Ausgabe-Systeme: Verwalten die Kommunikation zwischen dem Computer und externen Geräten.
38. Klassifizierung von Eingabe/Ausgabe-Geräten: Enthält Eingabegeräte, Ausgabegeräte und Speicherg-
eräte.
39. Eingabe/Ausgabe-Schnittstellen: Standards wie USB, SATA und PCIe, die definieren, wie Geräte mit der
Hauptplatine kommunizieren.
40. Direkter Speicherzugriff (DMA): Ermöglicht es Geräten, Daten zu/von Speicher zu übertragen, ohne
dass der CPU eingegriffen wird.
41. Interrupts: Signale, die die CPU über Ereignisse benachrichtigen, die sofortige Aufmerksamkeit er-
fordern, und ermöglichen asynchrones Verarbeiten.
42. Interrupt-Verarbeitung: Der Prozess, durch den die CPU auf Interrupts reagiert, einschließlich des Spe-
icherns des Zustands und des Ausführens von Interrupt-Dienstroutinen.
43. DMA-Controller: Hardwarekomponenten, die DMA-Operationen verwalten und die CPU von Datenüber-
tragungsaufgaben entlasten.
44. Gerätetreiber: Software, die es dem Betriebssystem ermöglicht, mit Hardwaregeräten zu kommu-
nizieren.
45. Peripheral Component Interconnect (PCI): Ein Standard zum Verbinden von Peripheriegeräten mit der
Hauptplatine.
46. Seriell vs. Parallelkommunikation: Seriell sendet Daten bitweise, während Parallel mehrere Bits gle-
ichzeitig sendet.
47. Serielle Schnittstellen: Schnittstellen wie RS-232, die für die serielle Kommunikation mit Geräten ver-
wendet werden.
48. Parallele Schnittstellen: Schnittstellen, die für die parallele Kommunikation, oft mit Druckern und an-
deren Peripheriegeräten, verwendet werden.
49. Bus-Arbitrierung: Der Prozess der Verwaltung des Zugriffs auf den Bus durch mehrere Geräte, um
Konflikte zu verhindern.
50. Systembusse vs. Peripheriebusse: Systembusse verbinden die CPU, den Speicher und die Hauptkom-
ponenten, während Peripheriebusse externe Geräte verbinden.
51. Interrupt-Vektor-Tabelle: Eine Datenstruktur, die verwendet wird, um die Adressen von Interrupt-
Dienstroutinen zu speichern.

52. Programmierbare Interrupt-Controller: Hardware, die mehrere Interrupt-Anfragen verwaltet und sie priorisiert.
53. Busbreite: Die Anzahl der Bits, die gleichzeitig über einen Bus übertragen werden können.
54. Taktfrequenz: Die Rate, mit der eine CPU Anweisungen ausführt, gemessen in GHz.
55. Taktzyklus: Die grundlegende Zeiteinheit, in der eine CPU eine grundlegende Operation durchführen kann.
56. Taktversatz: Unterschiede in den Ankunftszeiten des Taktsignals an verschiedenen Teilen der Schaltung.
57. Taktverteilung: Die Methode zur Lieferung des Taktsignals an alle Komponenten in der CPU.
58. Wärmeableitung: Der Prozess des Entferns von überschüssiger Wärme von der CPU, um Überhitzung zu verhindern.
59. Kühlungslösungen: Enthält Kühlkörper, Lüfter und Flüssigkeitskühlsysteme, die zur Verwaltung der CPU-Temperaturen verwendet werden.
60. Stromversorgungseinheiten (PSUs): Stellen die notwendige Energie für alle Computerkomponenten bereit.
61. Spannungsregler: Stellen sicher, dass stabile Spannungsniveaus an die CPU und andere Komponenten geliefert werden.
62. Motherboard-Architektur: Die Hauptplatine, die die CPU, den Speicher und andere kritische Komponenten beherbergt.
63. Chipsets: Gruppen von integrierten Schaltkreisen, die den Datenfluss zwischen der CPU, dem Speicher und den Peripheriegeräten verwalten.
64. Firmware: Permanente Software, die in einen Nur-Lese-Speicher programmiert ist und Hardwarefunktionen steuert.
65. BIOS/UEFI: Firmware-Schnittstellen, die die Hardware während des Bootvorgangs initialisieren und Laufzeitdienste bereitstellen.
66. Boot-Prozess: Die Sequenz von Operationen, die das System initialisiert, wenn es eingeschaltet wird.
67. Befehlspipeline-Stufen: Enthält in der Regel Fetch, Decode, Execute, Memory Access und Write Back.
68. Pipeline-Tiefe: Die Anzahl der Stufen in einer Pipeline, die die Anweisungsdurchsatzrate und Latenz beeinflusst.
69. Pipeline-Balancierung: Sicherstellen, dass jede Stufe etwa die gleiche Ausführungszeit hat, um die Effizienz zu maximieren.
70. Datengefahren: Situationen, in denen Anweisungen von den Ergebnissen vorheriger Anweisungen in einer Pipeline abhängen.

71. Steuergefahren: Treten aufgrund von Verzweigungsanweisungen auf, die den Pipeline-Fluss stören.
72. Strukturgefahren: Treten auf, wenn Hardware-Ressourcen unzureichend sind, um alle möglichen Anweisungsausführungen gleichzeitig zu unterstützen.
73. Weiterleitung (Datenumgehung): Eine Technik zur Verringerung von Datengefahren durch Routing von Daten direkt zwischen Pipeline-Stufen.
74. Stopp (Pipeline-Blase): Einfügen von Leerzyklen in die Pipeline, um Gefahren zu lösen.
75. Ausserhalb der Reihenfolge Ausführung: Ausführen von Anweisungen, wenn Ressourcen verfügbar werden, anstatt in der ursprünglichen Programmreihenfolge.
76. Spekulative Ausführung: Ausführen von Anweisungen, bevor bekannt ist, ob sie benötigt werden, um die Leistung zu verbessern.
77. Verzweigungsvorhersage-Algorithmen: Techniken wie statische Vorhersage, dynamische Vorhersage und zweistufige adaptive Vorhersage, die verwendet werden, um Verzweigungsrichtungen zu erraten.
78. Anweisungsparallelität (ILP): Die Fähigkeit, mehrere Anweisungen gleichzeitig innerhalb eines einzigen CPU-Zyklus auszuführen.
79. Schleifenentfaltung: Eine Optimierungstechnik, die den Körper von Schleifen erhöht, um die Überkopfkosten der Schleifensteuerung zu verringern.
80. Superpipelining: Erhöhen der Anzahl der Pipeline-Stufen, um höhere Taktfrequenzen zu ermöglichen.
81. VLIW (Very Long Instruction Word): Eine Architektur, die es ermöglicht, mehrere Operationen in einem einzigen Anweisungswort zu codieren.
82. EPIC (Explicitly Parallel Instruction Computing): Eine Architektur, die parallele Anweisungsausführung durch Compilerunterstützung ermöglicht.
83. Register-Umbenennung: Eine Technik zur Eliminierung falscher Datenabhängigkeiten durch dynamische Zuweisung von Registern.
84. Hyper-Threading: Intels Technologie, die es einem einzelnen CPU-Kern ermöglicht, mehrere Threads gleichzeitig auszuführen.
85. Cache-Speicher-Ebenen: L1 (nächste zur CPU, schnellste), L2 und L3 Caches mit zunehmender Größe und Latenz.
86. Write-Through vs. Write-Back Caches: Write-Through aktualisiert sowohl Cache als auch Speicher gleichzeitig, während Write-Back nur den Cache aktualisiert und Speicheraktualisierungen aufschiebt.
87. Assoziativität in Caches: Bestimmt, wie Cache-Zeilen auf Cache-Sets abgebildet werden, was die Trefferraten und Zugriffszeiten beeinflusst.
88. Vorabholen: Laden von Daten in den Cache, bevor sie tatsächlich angefordert werden, um die Zugriffslatenz zu verringern.

89. Speicherzugriffsmuster: Sequenzieller vs. zufälliger Zugriff und deren Auswirkungen auf die Cache-Leistung.
90. NUMA (Non-Uniform Memory Access): Ein Speicherdesign, bei dem die Speicherzugriffszeit je nach Speicherort relativ zu einem Prozessor variiert.
91. SMP (Symmetric Multiprocessing): Ein System, bei dem mehrere Prozessoren einen einzigen, zentralisierten Speicher teilen.
92. Verteilte Speichersysteme: Systeme, bei denen jeder Prozessor seinen eigenen privaten Speicher hat und über ein Netzwerk kommuniziert.
93. Vernetzung: Die Topologie und Protokolle, die verwendet werden, um mehrere Prozessoren und Speichereinheiten zu verbinden.
94. Skalierbarkeit: Die Fähigkeit eines Computersystems, die Leistung durch Hinzufügen weiterer Ressourcen zu erhöhen.
95. Fehlertoleranz: Die Fähigkeit eines Systems, ordnungsgemäß weiterzuarbeiten, wenn einige seiner Komponenten ausfallen.
96. Redundanz: Integrieren zusätzlicher Komponenten, um Zuverlässigkeit und Verfügbarkeit zu erhöhen.
97. Fehlererkennung und -korrektur: Techniken wie Paritätsbits, Prüfsummen und ECC (Error-Correcting Code), um Datenfehler zu identifizieren und zu korrigieren.
98. Energieeffizienz: Designen von Systemen zur Minimierung des Energieverbrauchs bei gleichzeitiger Aufrechterhaltung der Leistung.
99. Thermische Designleistung (TDP): Die maximale Wärmemenge, die ein CPU oder GPU unter typischen Arbeitslasten erzeugen soll.
100. Zukunftstrends: Erforschung von Fortschritten wie Quantencomputing, neuromorphischen Architekturen und photonischen Prozessoren, die die Zukunft der Computerorganisation prägen.