कंप्यूटर संगठन

1. परिभाषा: कंप्यूटर संगठन कंप्यूटर प्रणालियों के संचालन संरचना और कार्यान्वयन को संदर्भित करता है, जो हार्डवेयर घटकों के बीच कैसे संचारित होते हैं ताकि निर्देशों का कार्यान्वयन किया जा सके।
2. ऐतिहासिक विकास: प्रारंभिक यांत्रिक कंप्यूटर से आधुनिक बहु-कोर प्रोसेसर तक के विकास का पता लगाता है।
3. वॉन न्यूमैन आर्किटेक्चर: एक आधारभूत मॉडल जहां □□□, मेमोरी और □/□ एक बस के माध्यम से जुड़े होते हैं।
4. हार्वर्ड आर्किटेक्चर: निर्देशों और डेटा के लिए भंडारण और सिग्नल पथ अलग करते हैं, जिससे प्रदर्शन में वृद्धि होती है।
5. □□□ घटक: इसमें गणितीय तर्क इकाई (□□□), नियंत्रण इकाई (□□) और रजिस्टर शामिल हैं।
6. □□□ कार्य: गणितीय और तर्कीय संचालन जैसे जोड़ना, घटाना, □□□, □□ का कार्य करता है।
7. नियंत्रण इकाई का भूमिका: प्रोसेसर का संचालन निर्देशों को डिकोड करने और नियंत्रण सिग्नल उत्पन्न करने के द्वारा निर्देशित करता है।
8. रजिस्टर: □□□ के भीतर डेटा और निर्देशों को अस्थायी रूप से रखने के लिए छोटे, तेज़ भंडारण स्थान।
9. कैश मेमोरी: डेटा एक्सेस समय को कम करने के लिए □□□ के पास उच्च-गति मेमोरी।
10. मेमोरी हायरार्की: गति और लागत के आधार पर मेमोरी को स्तरों में संगठित करता है, जिसमें रजिस्टर, कैश, □□□ और द्वितीयक भंडारण शामिल हैं।
11. □□□ (रैंडम एक्सेस मेमोरी): वर्तमान में उपयोग में आने वाले डेटा और मशीन कोड को भंडारित करने के लिए अस्थायी मेमोरी।
12. □□□ (रिड-ओनली मेमोरी): फर्मवेयर और सिस्टम बूट निर्देशों को भंडारित करने के लिए स्थायी मेमोरी।
13. बस संरचना: कंप्यूटर के भीतर या बाहर घटकों के बीच डेटा को स्थानांतरित करने का संचार प्रणाली।
14. डेटा बस: संचालित होने वाले वास्तविक डेटा को ले जाती है।
15. एड्रेस बस: डेटा को कहां भेजा जाना चाहिए या प्राप्त किया जाना चाहिए, इसके बारे में जानकारी ले जाती है।
16. नियंत्रण बस: □□□ से अन्य घटकों को नियंत्रण सिग्नल ले जाती है।
17. निर्देश सेट आर्किटेक्चर (□□□): □□□ द्वारा कार्यान्वित किए जा सकने वाले निर्देशों की सेट को परिभाषित करता है।
18. □□□□ (रिड्यूस्ड इंस्ट्रक्शन सेट कंप्यूटिंग): एक □□□ डिजाइन दर्शन जो एक छोटे, अत्यधिक अनुकूलित निर्देशों की सेट का उपयोग करता है।
19. □□□□ (कम्प्लेक्स इंस्ट्रक्शन सेट कंप्यूटिंग): एक □□□ जिसमें एक बड़ा सेट निर्देश शामिल है, जिनमें से कुछ जटिल कार्य कर सकते हैं।
20. पाइपलाइनिंग: □□□ थ्रूपुट को बढ़ाने के लिए कई निर्देश चरणों को ओवरलैप करने का तकनीक।
21. पाइपलाइन चरण: आम तौर पर □□□□□, □□□□□□, □□□□□□□, □□□□□□ □□□□□□, और □□□□□ □□□□ शामिल होते हैं।
22. पाइपलाइनिंग में खतरे: डेटा खतरे, नियंत्रण खतरे और संरचनात्मक खतरे जैसे मुद्दे जो पाइपलाइन फ्लो को बाधित कर सकते हैं।
23. ब्रांच प्रेडिक्शन: पाइपलाइन को भरा रखने के लिए ब्रांच निर्देशों की दिशा का अनुमान लगाने का एक तरीका।
24. सुपरस्केलर आर्किटेक्चर: एकल पाइपलाइन चरण में कई निर्देशों को एक साथ संचालित करने की अनुमति देता है।
25. समांतर संचालन: निर्देशों को एक साथ संचालित करने के लिए कई प्रोसेसर या कोर का उपयोग करता है।
26. बहु-कोर प्रोसेसर: एकल चिप में एक साथ कई संचालन कोर शामिल हैं।

27. □□□□ (सिंगल इंस्ट्रक्शन, मल्टीपल डेटा): एक समांतर संचालन आर्किटेक्चर जहां एकल निर्देश कई डेटा बिंदुओं पर एक साथ कार्य करता है।

28. □□□□ (मल्टीपल इंस्ट्रक्शन, मल्टीपल डेटा): एक समांतर आर्किटेक्चर जहां कई प्रोसेसर अलग-अलग डेटा पर अलग-अलग निर्देशों का कार्यान्वयन करते हैं।

29. मेमोरी प्रबंधन: मेमोरी को प्रभावी ढंग से प्रबंधित और आवंटित करने के तकनीक, जिसमें पेजिंग और सेगमेंटेशन शामिल हैं।

30. वर्चुअल मेमोरी: भौतिक मेमोरी को डिस्क भंडारण पर बढ़ाता है, जिससे प्रणालियों को बड़े वर्कलोड संभालने की अनुमति मिलती है।

31. पेजिंग: मेमोरी को स्थिर आकार के पेजों में विभाजित करता है ताकि मेमोरी प्रबंधन को सरल बनाया जा सके और फ्रैगमेंटेशन कम हो सके।

32. सेगमेंटेशन: मेमोरी को फंक्शन या डेटा संरचनाओं जैसे तार्किक विभाजनों के आधार पर परिवर्तनीय आकार के सेगमेंटों में विभाजित करता है।

33. कैश मैपिंग तकनीक: डायरेक्ट-मैप्ड, फुली एसोसिएटिव और सेट-एसोसिएटिव कैश शामिल हैं।

34. कैश प्रतिस्थापन नीति: कैश प्रविष्टि को प्रतिस्थापित करने का निर्णय लेती है, जैसे कि □□□□□ □□□□□□□□□ □□□□ (□□□) या □□□□□-□□-□□□□□-□□□ (□□□□□)।

35. कैश सहजता: बहु-प्रोसेसर प्रणाली में कई कैशों में भंडारित डेटा की एकरूपता सुनिश्चित करता है।

36. मेमोरी सहजता मॉडल: प्रणाली सहजता बनाए रखने के लिए संचालन का क्रम परिभाषित करता है।

37. इनपुट/आउटपुट प्रणालियाँ: कंप्यूटर और बाहरी उपकरणों के बीच संचार का प्रबंधन करती हैं।

38. □/□ उपकरण वर्गीकरण: इनपुट उपकरण, आउटपुट उपकरण और भंडारण उपकरण शामिल हैं।

39. □/□ इंटरफेस: □□□, □□□□ और □□□□ जैसे मानक जो उपकरणों को मदरबोर्ड के साथ संचार करने का तरीका परिभाषित करते हैं।

40. डायरेक्ट मेमोरी एक्सेस (□□□): □□□ हस्तक्षेप के बिना उपकरणों को मेमोरी में डेटा स्थानांतरित करने की अनुमति देता है।

41. इंटरप्ट: □□□ को तत्काल ध्यान देने की आवश्यकता वाले घटनाओं की सूचना देने वाले सिग्नल हैं, जिससे असंयोजित संचालन की अनुमति मिलती है।

42. इंटरप्ट हैंडलिंग: □□□ इंटरप्टों का जवाब देने की प्रक्रिया, जिसमें स्टेट को बचाना और इंटरप्ट सेवा रूटीन का कार्यान्वयन शामिल है।

43. □□□ नियंत्रक: □□□ संचालन का प्रबंधन करने वाले हार्डवेयर घटक, जिससे □□□ को डेटा स्थानांतरण से मुक्त हो जाता है।

44. डिवाइस ड्राइवर: ऑपरेटिंग सिस्टम को हार्डवेयर उपकरणों के साथ संचार करने की अनुमति देने वाला सॉफ्टवेयर।

45. पेरिफेरल कंपोनेंट इंटरकनेक्ट (□□□): पेरिफेरल्स को मदरबोर्ड से जोड़ने के लिए एक मानक।

46. सीरियल बनाम समांतर संचार: सीरियल एक बार में एक बिट भेजता है, जबकि समांतर एक साथ कई बिट भेजता है।

47. सीरियल पोर्ट: □□-232 जैसे इंटरफेस जो उपकरणों के साथ सीरियल संचार के लिए उपयोग होते हैं।

48. समांतर पोर्ट: प्रिंटर और अन्य पेरिफेरल्स के साथ समांतर संचार के लिए उपयोग किए जाने वाले इंटरफेस।

49. बस अर्बिट्रेशन: कई उपकरणों के बीच बस तक पहुंच का प्रबंधन करने की प्रक्रिया ताकि टकराव से बचा जा सके।

50. सिस्टम बस बनाम पेरिफेरल बस: सिस्टम बस □□□, मेमोरी और मुख्य घटकों को जोड़ते हैं, जबकि पेरिफेरल बस बाहरी उपकरणों को जोड़ते हैं।

51. इंटरप्ट वेक्टर टेबल: इंटरप्ट सेवा रूटीनों के पते भंडारित करने के लिए एक डेटा संरचना।

52. प्रोग्रामेबल इंटरप्ट नियंत्रक: कई इंटरप्ट अनुरोधों का प्रबंधन करने और उन्हें प्राथमिकता देना वाले हार्डवेयर।

53. बस चौड़ाई: एक बार में बस के माध्यम से भेजे जा सकने वाले बिटों की संख्या।

54. क्लॉक स्पीड: □□□ द्वारा निर्देशों का कार्यान्वयन करने की दर, □□□ में मापी जाती है।

55. क्लॉक साइकिल: □□□ द्वारा एक मूल संचालन करने के लिए एक मूल समय इकाई।

56. क्लॉक स्क्यू: सर्किट के विभिन्न हिस्सों पर क्लॉक सिग्नल की पहुंच के समय में अंतर।

57. क्लॉक वितरण: □□□ के सभी घटकों को क्लॉक सिग्नल पहुंचाने का तरीका।

58. गर्मी निष्कासन: □□□ को ओवरहिट होने से बचाने के लिए अतिरिक्त गर्मी को हटाने की प्रक्रिया।

59. ठंडा करने के समाधान: गर्मी को प्रबंधित करने के लिए उपयोग किए जाने वाले हिट सिंक, फैन और लिक्विड कूलिंग सिस्टम शामिल हैं।

60. पावर सप्लाई यूनिट्स (□□□□): सभी कंप्यूटर घटकों को आवश्यक शक्ति प्रदान करते हैं।

61. वोल्टेज रेगुलेटर्स: □□□ और अन्य घटकों को स्थिर वोल्टेज स्तर प्रदान करने के लिए।

62. मदरबोर्ड आर्किटेक्चर: मुख्य सर्किट बोर्ड जो □□□, मेमोरी और अन्य महत्वपूर्ण घटकों को घर देता है।

63. चिपसेट: □□□, मेमोरी और पेरिफेरल्स के बीच डेटा प्रवाह का प्रबंधन करने वाले एकीकृत सर्किटों के समूह।

64. फर्मवेयर: हार्डवेयर कार्यों को नियंत्रित करने के लिए रिड-ओनली मेमोरी में प्रोग्राम किए गए स्थायी सॉफ्टवेयर।

65. □□□□/□□□□: हार्डवेयर को बूटिंग प्रक्रिया के दौरान प्रारंभ करने और रनटाइम सेवाएं प्रदान करने वाले फर्मवेयर इंटरफेस।

66. बूट प्रक्रिया: प्रणाली को शक्ति प्रदान करने पर प्रणाली को प्रारंभ करने के लिए संचालन क्रम।

67. निर्देश पाइपलाइन चरण: आम तौर पर □□□□□, □□□□□□, □□□□□□□, □□□□□□ □□□□□□, और □□□□□ □□□□ शामिल होते हैं।

68. पाइपलाइन गहराई: पाइपलाइन में चरणों की संख्या, जो निर्देश थ्रूपुट और देरी को प्रभावित करती है।

69. पाइपलाइन संतुलन: प्रत्येक चरण के पास लगभग समान कार्यान्वयन समय हो ताकि दक्षता को अधिकतम किया जा सके।

70. डेटा खतरे: पाइपलाइन में पूर्ववर्ती निर्देशों के परिणामों पर निर्भर होने वाले संचालन की स्थिति।

71. नियंत्रण खतरे: पाइपलाइन फ्लो को बाधित करने वाले ब्रांच निर्देशों के कारण होते हैं।

72. संरचनात्मक खतरे: सभी संभव निर्देशों का कार्यान्वयन समर्थन करने के लिए हार्डवेयर संसाधनों की कमी के कारण होते हैं।

73. फॉरवर्डिंग (डेटा बायपासिंग): डेटा खतरे को कम करने के लिए डेटा को पाइपलाइन चरणों के बीच सीधे मार्गित करने का एक तकनीक।

74. स्टॉल (पाइपलाइन बबल): खतरे को हल करने के लिए पाइपलाइन में निष्क्रिय चक्रों को डालना।

75. अनुक्रमित कार्यान्वयन: संसाधनों के उपलब्ध होने पर निर्देशों का कार्यान्वयन, मूल कार्यक्रम क्रम में नहीं।

76. अनुमानित कार्यान्वयन: कार्यान्वयन करने से पहले पता नहीं होता कि क्या वे आवश्यक हैं, प्रदर्शन को बढ़ाने के लिए निर्देशों का कार्यान्वयन।

77. ब्रांच प्रेडिक्शन एल्गोरिथम: ब्रांच दिशाओं का अनुमान लगाने के लिए उपयोग किए जाने वाले तकनीक जैसे स्टैटिक प्रेडिक्शन, डायनामीक प्रेडिक्शन और दो-स्तर अनुकूलित प्रेडिक्शन।

78. निर्देश स्तर समांतरता (□□□): एकल □□□ साइकिल में कई निर्देशों का एक साथ कार्यान्वयन करने की क्षमता।

79. लूप अनरोलिंग: लूप नियंत्रण ओवरहेड को कम करने के लिए लूप बॉडी को बढ़ाने का एक अनुकूलन तकनीक।

80. सुपरपाइपलाइनिंग: उच्च क्लॉक स्पीड की अनुमति देने के लिए पाइपलाइन चरणों की संख्या बढ़ाना।

81. □□□□ (वेरी लॉंग इंस्ट्रक्शन वर्ड): एक आर्किटेक्चर जो एकल निर्देश वर्ड में कई संचालनों को एनकोड करने की अनुमति देता है।

82. □□□□ (एक्सप्लिसिटली पैरलल इंस्ट्रक्शन कंप्यूटिंग): एक आर्किटेक्चर जो कंपाइलर सहायता के माध्यम से समांतर निर्देश कार्यान्वयन की अनुमति देता है।

83. रजिस्टर रेनेमिंग: रजिस्टरों को डायनामिक रूप से आवंटित करके झूठे डेटा निर्भरताओं को हटाने का एक तकनीक।

84. हाइपर-थ्रेडिंग: एकल □□□ कोर को एक साथ कई थ्रेड्स का कार्यान्वयन करने की अनुमति देने वाला इंटेल का टेक्नोलॉजी।

85. कैश मेमोरी स्तर: □1 (□□□ के सबसे करीब, सबसे तेज़), □2 और □3 कैश, जिसमें आकार और देरी बढ़ती जाती है।

86. राइट-थ्रू बनाम राइट-बैक कैश: राइट-थ्रू कैश और मेमोरी दोनों को एक साथ अपडेट करता है, जबकि राइट-बैक केवल कैश को अपडेट करता है और मेमोरी अपडेट को टाल देता है।

87. कैश में एसोसिएटिविटी: कैश लाइनों को कैश सेट्स में मैप करने का तरीका, जो हिट रेट और एक्सेस समय को प्रभावित करता है।

88. प्रीफेट्चिंग: डेटा को वास्तव में अनुरोध किए जाने से पहले कैश में लोड करने के लिए, एक्सेस देरी को कम करने के लिए।

89. मेमोरी एक्सेस पैटर्न: अनुक्रमिक बनाम रैंडम एक्सेस और उनके कैश प्रदर्शन पर प्रभाव।

90. □□□□ (नॉन-यूनिफॉर्म मेमोरी एक्सेस): एक मेमोरी डिजाइन जहां मेमोरी एक्सेस समय प्रोसेसर के संबंध में मेमोरी स्थान पर निर्भर करता है।

91. □□□ (सिमेट्रिक मल्टीप्रोसेसिंग): एक प्रणाली जहां कई प्रोसेसर एकल, केंद्रित मेमोरी को साझा करते हैं।

92. वितरित मेमोरी प्रणालियाँ: प्रत्येक प्रोसेसर के पास अपना अपना निजी मेमोरी होता है, जो एक नेटवर्क के माध्यम से संचार करता है।

93. इंटरकनेक्ट नेटवर्क: कई प्रोसेसर और मेमोरी इकाइयों को जोड़ने के लिए उपयोग किए जाने वाले टोपोलॉजी और प्रोटोकॉल।

94. स्केलेबिलिटी: एक कंप्यूटर प्रणाली को अधिक संसाधनों को जोड़कर प्रदर्शन बढ़ाने की क्षमता।

95. फॉल्ट टॉलरेंस: प्रणाली के कुछ घटकों में विफलता होने के बावजूद सही ढंग से संचालित रहने की प्रणाली की क्षमता।

96. रेडंडेंसी: विश्वसनीयता और उपलब्धता को बढ़ाने के लिए अतिरिक्त घटकों को शामिल करना।

97. त्रुटि पता लगाना और संशोधन: त्रुटि को पहचानने और संशोधित करने के लिए तकनीक जैसे पैरिटी बिट, चेकसम और □□□ (एरर-कॉरेक्टिंग कोड)।

98. शक्ति दक्षता: प्रदर्शन को बनाए रखते हुए शक्ति उपभोग को कम करने के लिए प्रणालियों का डिजाइन करना।

99. थर्मल डिजाइन पावर (□□□): सामान्य वर्कलोड के तहत □□□ या □□□ द्वारा अपेक्षित अधिकतम गर्मी का उत्पादन।

100. भविष्य के ट्रेंड: क्वांटम कंप्यूटिंग, न्यूरोमॉर्फिक आर्किटेक्चर और फोटोनिक प्रोसेसर जैसे विकास कंप्यूटर संगठन के भविष्य को आकार देने में शामिल हैं।