

Gabarito Lista 1 – Arquitetura de computadores

1. O tradutor converte o programa original em um outro programa com a nova linguagem, ou seja, na tradução um programa em um nível acima substitui cada instrução escrita por uma equivalente do nível a baixo, então o computador executa este novo programa (totalmente traduzido) e descarta o antigo. Já a interpretação consiste em buscar a primeira instrução, executar, buscar a próxima, executar, e assim por diante... esta técnica não requer a geração prévia de um novo programa. Uma máquina virtual é uma máquina conceitual, não existe.

2. An interpreter executes a program by fetching the first instruction, carrying it out, then fetching the next one, and so on. A translator first converts the original program into an equivalent one in another language and then runs the new program.

3. It is possible, but there are problems. One difficulty is the large amount of code produced. Since one ISA instruction does the work of many microinstructions, the resulting program will be much bigger. Another problem is that the compiler will have to deal with a more primitive output language, hence it, itself, will become more complex. Also, on many machines, the microprogram is in ROM. Making it user-changeable would require putting it in RAM, which is much slower than ROM. On the positive side, the resulting program might well be much faster, since the overhead of one level of interpretation would be eliminated.

4. Durante o projeto detalhado de um novo computador, os níveis da nova máquina de dispositivos e da lógica digital podem muito bem serem simulados em uma antiga máquina, o que os coloca em torno de nível 5 ou 6.

5. Cada nível de interpretação retarda a máquina por um fator de n/m . Assim, os tempos de execução para os níveis 2, 3 e 4 são kn/m , kn^2/m^2 , e kn^3/m^3 , respectivamente.

6. Cada nível adicional de interpretação custa algum tempo. Se não for necessário, deve ser evitado.

7. Você perde um fator de n em cada nível, por isso os tempos de execução de instruções em níveis 2, 3 e 4 são kn , kn^2 , e kn^3 , respectivamente.

8. Hardware e software são funcionalmente equivalentes. Qualquer função feita por um pode-se, em princípio, ser realizado pelo outro. E não são equivalentes no sentido de que para fazer a máquina realmente executar, o nível mais baixo (hardware) deve ser implementado. Eles também se diferem em performance.

9. Letra **d**.

10. Memória, unidade de lógica e aritmética, unidade de controle e equipamento de entrada e saída.

11. De modo nenhum. Se você quiser alterar o programa do mecanismo onde a mudança correu, você precisa trocar o computador inteiro, ou seja construir um novo. Um computador moderno não tem de ser substituído porque você quer mudar o programa. Ele pode ler muitos programas de muitos CD-ROMs.

12. A typical example is a program that computes the inner product of two arrays, A and B. The first two instructions might fetch A[0] and B[0], respectively. At the end of the iteration, these instructions could be incremented to point to A[1] and B[1], respectively. Before indexing and indirect addressing were invented, this was done.

13. De acordo com a Lei de Moore, o número de transistores em um chip dobra a cada 18 meses, isso equivale a um aumento de 60% no número de transistores por ano. Essa não é uma lei real, mas uma simples observação empírica.

14. A primeira lei do software de Nathan diz que “O software é um gás. Ele se expande até preencher o recipiente que o contém.” Ou seja, o software, que continua a adquirir características, cria uma demanda constante por processadores mais velozes, memórias maiores e mais capacidade de E/S.

15. De acordo com a lei de Moore, no próximo ano o mesmo chip terá 1,6 vezes mais transistores. Isto significa que a área de cada transistor será $1/1,6$ ou 0.625 vezes o tamanho dos transistores deste ano. Uma vez que a área é calculada com o quadrado do diâmetro, o diâmetro de transistores do próximo ano deve ser 0,079 microns.

16. Cada pessoa consome 730 etiquetas por ano. Multiplique por 300 milhões e obtém 219 bilhões de etiquetas por ano. Sendo um centavo uma etiqueta, o custo é de \$2.19 bilhões de dólares por ano. Com um PIB superior a \$10 trilhões, as etiquetas somam 0,02% do PIB, e não é um obstáculo enorme.

18. A ULA é o dispositivo da CPU que executa operações tais como: Adição, Subtração, Multiplicação, Divisão, Incremento, Decremento, Operação lógica AND, Operação lógica OR, Operação lógica XOR, Operação complemento, Deslocamento à direita, Deslocamento à esquerda. Tais operações podem utilizar dois operandos (adição, and, etc..) ou apenas um valor (como, por exemplo, as operações de complemento ou deslocamento). A ULA é um aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações mencionadas acima. Ela pode ser uma parte pequena da pastilha do processador, usada em pequenos sistemas, ou pode compreender um considerável conjunto de componentes lógicos de alta velocidade. A ULA é responsável por processar as operações aritméticas e lógicas requeridas nas instruções de máquina.

A área de controle de uma UCP é a parte que realiza as seguintes atividades: a) Busca de instrução que será executada, armazenando-a em um registrador especialmente projetado para esta finalidade. b) Interpretação das ações que serão processadas durante a execução da instrução. Por exemplo, se é uma soma, um complemento, etc. c) Geração dos sinais de controle apropriados para a execução da instrução identificada. Sinais estes que são enviados aos diversos componentes do sistema, sejam eles internos ou externos.

19. letra **b**.

20. O ciclo de caminho de dados é de 20 nanossegundos. O número máximo de ciclos/segundo o caminho de dados, portanto, é de 50 milhões. A melhor a máquina poderia fazer, assim, 50 MIPS.

21. As arquiteturas RISC e CISC se diferenciam no tamanho do conjunto de instruções. Na arquitetura RISC o conjunto de instruções são reduzidos, ou seja, poucas instruções, e estas são simples. Já a arquitetura CISC possui um vasto número de instruções e estas por sua vez são mais complexas. A vantagem da RISC é a velocidade, as instruções por serem mais simples e implementadas diretamente no hardware, são executadas mais rapidamente. A CISC por sua vez tem a vantagem de ser mais barata de se implementar, já que a implementação diretamente no hardware eleva o custo. Outra vantagem CISC é a compatibilidade com softwares anteriores.

22. O pipeline divide a execução da instrução em várias partes, cada uma manipulada por uma parte dedicada do hardware, e todas elas podem ser executadas em paralelo. Um pipeline dividido em cinco unidades, consiste nos seguintes estágios: 1 – busca a instrução na memória e a coloca em um buffer até que ela seja necessária. 2 – decodifica a instrução, determina seu tipo e de quais operandos ela necessita. 3 – localiza e busca os operandos, seja nos registradores, seja na memória. 4 – realiza o trabalho de executar a instrução, normalmente fazendo os operandos passar pelo caminho de dados. 5 – escreve o resultado de volta no registrador adequado.

23. Ao tamanho da palavra do processador, ou seja, o número de bits que o processador é capaz de manipular. Isto diz respeito ao número de bits (tamanho) dos registradores internos, o tamanho em bits que a ULA é capaz de processar, o tamanho do barramento de dados.

24. Um processador de pipeline é melhor. Processadores matriciais só são úteis se o problema contém paralelismo inerente.

25. O mosteiro se assemelha a Figura 2-7, página 37 o livro (5ª edição); com um mestre e muitos escravos.

26. Você não pode afirmar nada com certeza. Por exemplo, se o computador 1 tem um pipeline de cinco estágios, que pode emitir até 500 milhões de instruções/segundo e se o computador 2 não tem um pipeline, ele não pode fazer melhor do que 200 milhões de instruções/seg. assim, sem obter mais informações, você não pode dizer qual é mais rápido.

27. letra **d**.

28. letra **e**.

29. letra **c**.

30. $2 - 2 - 1 - 1 - 2$