

Lista 1 - Arquitetura e Organização de Computadores

REPOSTAS NÃO OFICIAIS ELABORADAS PELO MONITOR

19 de novembro de 2024

Questões

Q.1 Quais os três principais componentes de qualquer computador de propósito geral? Quais destes componentes residem atualmente no circuito integrado do processador?

Q.2 Por que, ao contrário de comandos em linguagem de alto nível, instruções representam apenas uma única operação?

Q.3 Camadas de abstração estão presentes em vários pontos da estrutura de um sistema computacional. Descreva as principais abstrações vistas em aula, listando os programas que operam em sua fronteira.

Q.4 Suponha que você está construindo uma abstração para acesso a disco. Quais seriam as operações básicas de sua abstração?

Q.5 Por que não é possível executar um programa MIPS em um PC com processador x86?

Q.6 Qual a principal estratégia utilizada para aumentar o poder de processamento de computadores? Qual a relação desta estratégia com a Lei de Moore?

Q.7 Por que observamos uma redução da taxa de aumento do ciclo de relógio dos processadores modernos?

Q.8 Um programa é executado em um processador de 40 MHz. Este programa executa 100.000 instruções, com a seguinte mistura de ciclos de relógio por instrução:

Tipo	Quantidade	CPI
Aritmética de inteiros	40.000	1
Transferência de dados	32.000	2
Ponto flutuante	20.000	2
Transferência de controle	8.000	2

Determine o CPI efetivo e o tempo de execução deste programa.

Q.9 Duas arquiteturas diferentes, executando o mesmo programa compilado para cada arquitetura, produzem os seguintes resultados:

Arquitetura	Frequência Relógio	Quantidade Instruções	Tempo
A	5 MHz	12.000.000	12 s
B	25 MHz	18.000.000	1 s

a) Qual o valor de CPI para cada arquitetura?

b) Usando apenas frequência de relógio e CPI, calcule quantas vezes B é mais rápido que A. Compare este número com a diferença em tempo de execução do programa nas duas arquiteturas. Explique a diferença.

Q.10 Suponha que uma CPU pode realizar uma multiplicação em 10 ns e uma subtração em 1 ns. Quanto tempo demorará o cálculo de $d = a * b - a * c$? Você consegue otimizar a equação de forma a levar menos tempo?

Q.11 O que distingue os registradores da memória? Qual o efeito disto nas instruções?

Q.12 Por que os vários tipos de memória estão organizados em uma hierarquia?

Q.13 Tomando-se o acesso a memória na arquitetura MIPS:

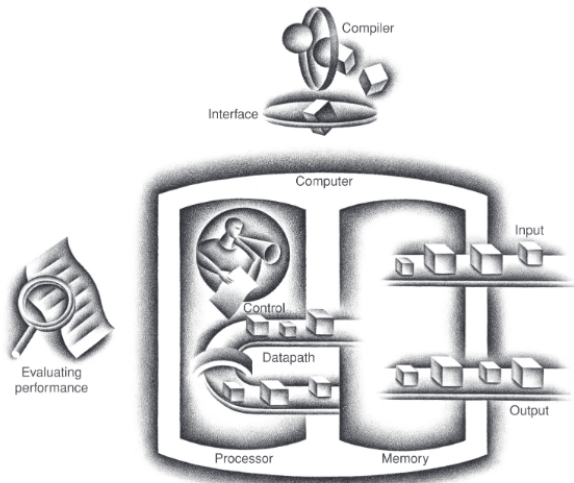
a) Qual a capacidade máxima desta memória em bits, em bytes e em palavras?

b) Qual o último endereço em notação binária e decimal e qual o total de posições endereçáveis.

c) Supondo que uma palavra leva 30 ns para ser transferida da memória para a CPU, quanto tempo é necessário para se ler todo o conteúdo de uma memória de um computador com 2GB?

Respostas

R.1 No geral, todos os computadores realizam as mesmas funções básicas: inserir, processar, armazenar dados e gerar saída dos dados. Para realizar essas tarefas são necessários três componentes principais: E/S, memória e um processador (combinação de caminho de dados e controle).



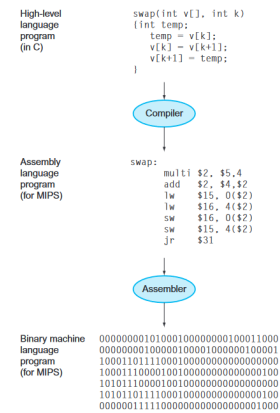
Ao analisar o circuito integrado de um processador, encontramos seus componentes: o caminho de dados, que realiza as operações, e o controle, que comanda o funcionamento do caminho de dados, memória e dispositivos de E/S. Além desses dois, ainda dentro do processador, temos um tipo de memória, o cache, que atua como um buffer rápido para a memória principal.



R.2 O conjunto de instruções é projetado para representar o funcionamento direto do hardware, que executa operações básicas e atômicas. Instruções mais simples podem ser executadas pelo processador em poucos ciclos. No geral, os aspectos do projeto visam construir um projeto que seja eficiente.

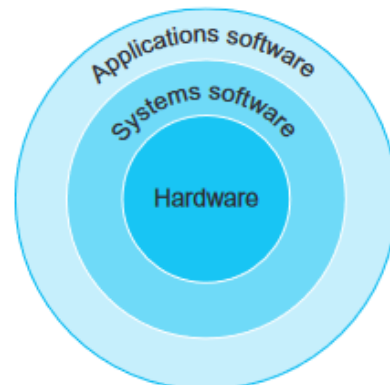
R.3 De cima para baixo, temos:

Aplicação: Escrita em linguagem de alto nível. Entretanto, as aplicações precisam ser traduzidas em instruções que o computador de fato consiga executar. Uma linguagem compilada, por exemplo, utiliza um software de sistema chamado compilador, que traduz o código alto nível para linguagem assembly (instruções). As instruções são posteriormente convertidas para binário com outro software de sistema, chamado montador (assembler).



Softwares de sistema: Fornecem serviços que são comumente utilizados, incluindo sistemas operacionais, compiladores e montadores. Tomando o sistema operacional como exemplo, uma vez que as aplicações foram traduzidas e estão prontas para serem executadas, elas são gerenciadas pelo SO, que disponibiliza recursos e interfaces, fazendo a ligação entre o programa e o hardware.

Hardware: Finalmente, a instrução chega no hardware para ser executada, passando por diversos estágios como busca, decodificação, execução, etc.



R.4 Algumas operações básicas devem ser implementadas:

disk_init(): Inicializa o disco para acesso, permitindo que outras operações sejam realizadas.

disk_info(): Retorna o tamanho do disco em setores. O tamanho total pode ser recuperado multiplicando os setores pelo tamanho de cada setor.

disk_write(): Escreve dados em um setor do disco.

disk_read(): Lê dados de um setor do disco.

R.5 Os dois processadores têm uma arquitetura de conjunto de instruções diferentes. O que impossibilita de um programa com instruções MIPS seja entendido por um processador x86. **a)**

R.6 Combinar vários transistores em um único chip foi a principal estratégia por trás dos aumentos no poder de processamento. A Lei de Moore diz que essa capacidade em transistores dobra a cada 18 a 24 meses.

R.7 A potência consumida por um circuito é proporcional ao produto do número de transistores pela frequência com que são chaveados.

$$P \propto \frac{1}{2} CV^2 f$$

Portanto, em geral, velocidades de clock mais altas ou números de transistores mais altos ocasionam maior consumo. Entretanto, especialmente na tecnologia CMOS, energia não é consumida quando ela está ociosa. Assim sendo, velocidades de clock mais baixas permitem que o processador "durma" e poupe energia.

R.8

$$\begin{aligned} \text{ciclos de clock da CPU} &= \sum_{i=1}^n (CPI_i \times C_i) \\ &= (40000 \times 1) + (32000 \times 2) \\ &\quad + (20000 \times 2) + (8000 \times 2) \\ &= 160000 \end{aligned}$$

$$\begin{aligned} CPI &= \frac{\text{ciclos de clock da CPU}}{\text{quantidade de instruções}} \\ &= \frac{1.6 \times 10^5}{1 \times 10^5} \\ &= 1.6 \end{aligned}$$

$$\begin{aligned} \text{tempo} &= \frac{\text{ciclos de clock da CPU para um programa}}{\text{Frequência de clock}} \\ &= \frac{1.6 \times 10^5}{4 \times 10^7} \\ &= 0.004 \text{ s} \end{aligned}$$

R.9

$$\begin{aligned} CPI_A &= \frac{\text{ciclos totais}}{\text{instruções}} \\ &= \frac{5 \times 10^6 \text{ Hz} \times 12 \text{ s}}{1.2 \times 10^7} \\ &= \frac{6 \times 10^7}{1.2 \times 10^7} \\ &= 5 \end{aligned}$$

$$\begin{aligned} CPI_B &= \frac{2.5 \times 10^7 \text{ Hz} \times 1 \text{ s}}{1.8 \times 10^7} \\ &\approx 1.39 \end{aligned}$$

b)

$$\begin{aligned} \text{Fator velocidade} &= \frac{CPI_A \times \text{Frequência}_B}{CPI_B \times \text{Frequência}_A} \\ &= \frac{5 \times 2.5 \times 10^7 \text{ Hz}}{1.38 \times 5 \times 10^6 \text{ Hz}} \\ &\approx 18 \end{aligned}$$

A arquitetura B é 18 vezes mais rápida que a A. Caso a quantidade de instruções fosse 18 milhões para as duas, o tempo da A aumentaria para 18s, deixando bem evidente a diferença de tempo.

R.10 Nessa organização, a CPU executa $x = a * b$ (10ns), depois $y = a * c$ (10ns) e por fim, $x - y$ (1ns). Totalizando 21ns. Para otimizar a equação, basta utilizar a propriedade distributiva e reorganizar da seguinte maneira: $a * (b - c)$, dessa forma, o processador executaria $b - c$ (1ns) e o resultado disso vezes a (10ns), dessa forma a execução levaria apenas 11ns.

R.11 Os registradores são um grupo limitado de locais especiais, embutidos no hardware do processador, e que são muito mais rápidos que a memória. O registrador está no mesmo patamar de veloci-

dade que o processador. Portanto, as instruções só podem ser executadas com valores contidos neles. Caso contrário, o processador teria que esperar pela memória para cada instrução, aumentando muito a latência do sistema.

R.12 A hierarquia de memória combina os diferentes níveis de memória para maximizar o desempenho e diminuir o custo, utilizando memórias mais rápidas e caras para dados mais acessados e memórias mais lentas e baratas para dados menos acessados.

R.13