



**UFC**

**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**EXERCÍCIOS RESOLVIDOS  
ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES  
WILLIAN STALLINGS**

**RONILDO OLIVEIRA DA SILVA**

**QUIXADÁ - CEARÁ  
2014**

# INTRODUÇÃO

## 1.1. Qual é, em termos gerais, a distinção entre a organização e a arquitetura do computador?

### Arquitetura

- Atributos de Sistema que são visíveis para o programador, atributos que tem impacto direto sobre a execução lógica de um programa.
  - Conjunto de instruções, número de bits utilizados para representar dados, mecanismos de E/S e técnicas de endereçamento de memória.

### Organização

- Refere-se às unidades operacionais e suas interconexões que implementam as especificações da sua arquitetura.
  - Detalhes de hardware transparentes ao programador, tais como, sinais de controle, as interfaces entre o computador e os periféricos e a tecnologia de memória utilizada.

## 1.2. Qual é, em termos gerais, a distinção entre a estrutura e a função do computador?

- Estrutura: Modo como os componentes são inter-relacionados.
- Função: Operação individual de cada componente como parte da estrutura.

## 1.3. Quais são as quatro funções principais de um computador?

Processamento, armazenamento, movimentação e controle.

## 1.4. Lista e defina resumidamente os principais componentes estruturais de um computador.

- CPU: Controla a operação do computador e desempenha funções de processamento de dados.
- Memória: Armazena dados e instruções.
- E/S: Transfere dados entre o computador e o ambiente externo (periféricos e comunicação de dados).
- Sistema de interconexão: Mecanismos que estabelecem a comunicação entre a CPU, a memória principal e os dispositivos de E/S.

## 1.5. Lista e defina resumidamente os principais componentes estruturais de um processador.

- Unidade de Controle: Controla a operação da CPU e, portanto do computador.
- Unidade Lógica Aritmética: desempenha as funções de processamento de dados do computador.
- Registradores: Fornecem o armazenamento que possibilita a comunicação entre a unidade de controle, a ULA e os registradores.
- Interconexão de CPU: Mecanismo que possibilita a comunicação entre a unidade de controle, a ULA e os registradores.

# DESEMPENHO E EVOLUÇÃO

## 2.1. O que é um computador de programa armazenado?

É um computador que obtém dados e instruções a partir da memória, pode ser modificado apenas pela atribuição de valores às posições de memória.

## 2.2. Quais são os quatro componentes principais de qualquer computador de uso geral?

- **Memória principal:** armazenadora de dados e instruções;
- **Unidade Lógica e Aritmética (ULA):** realiza operações com dados binários;
- **Unidade de Controle:** interpreta e executa instruções armazenadas na memória;
- **Dispositivos de E/S:** Eles permitem a interação do processador com o usuário.

### **2.3. No nível de circuito integrado, quais são os três constituintes principais de um sistema de computação?**

- Unidade de processamento (CPU);
- Controlador de trânsito de dados (Multiplexador);
- Armazenamento de dados e instruções (Memória).

### **2.4. Explique a lei de Moore.**

A lei de Moore observa que o número de transistores que podiam ser impressos em uma única pastilha dobrava a cada ano e previa corretamente que esse crescimento continuou, ano após anos e década após década.

### **2.5. Liste e explique as principais características de uma família de computadores.**

- Conjunto de instruções idêntico ou semelhante: Em muitos casos, o mesmo conjunto de instruções de máquina é usado em todos os membros da família. Membros mais simples da família possui em conjunto de instruções que constituem um subconjunto de instruções das instruções usadas nos computadores do topo de linha da família.
- Sistema operacional idêntico ou semelhante: O mesmo sistema operacional básico está disponível em todos os membros da família.
- Número crescente de portas de E/S.
- Custo crescente.
- Capacidade de memória crescente.

### **2.6 O clock do processador é uma das medidas utilizadas para medir sua velocidade. Explique o motivo pelo qual comparar apenas esse clock não é uma forma correta de comparar a capacidade de dois processadores de executarem instruções. Mostre ainda outras evoluções na arquitetura e organização dos processadores que melhoraram a eficiência de execução de instruções.**

Existem vários outros atributos de um processador que devem ser levados em conta. Além do clock, tem-se a velocidade básica, o tamanho e o grau de simultaneidade que implica em processamentos paralelos e transferências entre a memória e a CPU.

### **2.7 Comente as dificuldades em melhorar a eficiência dos processadores atuais e explique porque as arquiteturas paralelas são uma alternativa.**

A dificuldade de melhoria dos processamentos é devida a falta de correspondência de velocidade da taxa de transferência de dados entre a memória principal e a CPU.

### **2.8 O fator crítico de desempenho é na comunicação entre processador e memória principal. Explique algumas medidas que podem ser tomadas para melhorar essa comunicação.**

- Ampliar o número de bits obtidos em cada acesso à memória, aumentando a largura das memórias DRAM em vez de sua capacidade e utilizando barramentos de dados mais largos.
- Mudar a interface de memória DRAM para torná-las mais eficiente, usando uma memória cache ou outro esquema de armazenamento temporário na pastilha de memória DRAM.
- Reduzir a frequência de acesso à memória, incorporando estruturas de memórias cache eficientes e complexas entre o processador e a memória principal. Isso inclui o uso de memórias cache tanto na pastilha do processador como fora dele.
- Aumentar a largura de banda da conexão entre processadores e memórias usando barramento de alta velocidade e uma hierarquia de barramentos para estruturar o fluxo de dados e armazenar os dados temporariamente.

## BARRAMENTOS



### 3.11 Liste e defina resumidamente os estados possíveis que definem a execução de uma instrução .

#### Transferências de valores entre o processador e a memória ou E/S

- **Cálculo de endereço de instrução** – o endereço da próxima instrução a ser executada é determinado (geralmente é um incremento ao endereço da instrução anterior).
- **Busca de instrução** – Uma instrução é lida na memória e armazenada no processador.
- **Decodificação de instrução** – O código da instrução a ser executada é analisado, para determinar qual é a operação a ser realizada e os operandos a serem usados

#### Operações internas no processador.

- **Cálculo de endereço de operando** – Se a operação envolver a referência a um operando na memória ou estiver disponível via E/S, o endereço do operando será determinado.
- **Busca de operando** – O operando é localizado na memória ou é lido do dispositivo de E/S.
- **Execução da operação** – A operação indicada na instrução é executada.
- **Armazenamento de resultados** – O resultado é escrito na memória ou no dispositivo de E/S.

### 3.2. Qual o benefício em utilizar a arquitetura de barramento múltiplo em relação a estrutura de barramento único?

Usar um barramento múltiplo é mais eficiente pois é composto de vários barramentos únicos postos hierarquicamente, com funções e barramentos específicos (local, do sistema, de expansão e de alta velocidade em alguns casos). Ao contrário do barramento singular, pode receber mais conexões de dispositivos sem perder significativamente o desempenho.

### 3.3. Descreva como acontece a sequência de instruções na figura do slide 13 .

- É efetuado o cálculo da próxima instrução (incremento 1 no valor da instrução anterior, por exemplo).
- Busca a instrução lendo-a do seu local da memória para o processador.
- Analisa a instrução para determinar o tipo de operação a ser realizado e o operando a serem utilizados.
- Se a operação envolver referência a um operando na memória ou dispositivo E/S, então determina ou lê da E/S.
  - Realiza a operação indicada na instrução.
- Escreve o resultado na memória ou envia para a E/S.

### 3.4. Explique o que é uma interrupção, quais as suas vantagens e desvantagens e como o ciclo de interrupção é adicionado ao ciclo normal de uma instrução. Fale ainda sobre as múltiplas interrupções.

- Interrupção é um método de paralisação da espera do processador de uma instrução que está sendo concluída. Como o processador é mais rápido que as demais comunicações, ele interrompe a espera e processa uma nova instrução antes da anterior acabar.
  - As vantagens das interrupções é de não permitir a ociosidade do processador em meio as transferências de dados entre ele e dispositivos externos (que são mais lentos que a CPU).
  - As desvantagens podem ser resumidas no fato de a instrução anterior acabar no meio do processo de uma outra instrução ou interrupção. Daí, tem-se a necessidade de aplicar um método de manipulação de interrupções.
- Ciclo de interrupção é adicionado ao ciclo de de uma instrução para acomodar as interrupções. A cada execução de instrução, há a verificação de interrupção. Se a mesma não houver concluído, novamente busca-se mais uma instrução. Caso a instrução esteja finalizada, conclui-se o ciclo de normal de instrução.
- Considerando que haja várias instruções, haverá também múltiplas interrupções, isso é, o processador pode gerar uma interrupção para processar uma outra instrução, e nesse meio tempo,

poderá haver mais uma interrupção. O tratamento dessas múltiplas interrupções podem ser feitos em forma de fila, em ordem de “chegada” ou por prioridades das interrupções.



### 3.5. Fale sobre os três tipos de barramentos: endereço, dados e controle.

- Linhas de dados – fornecem um caminho para a transferência de dados entre os módulos do sistema. Possui tipicamente 8, 16 ou 32 linhas é conhecido como largura do barramento de dados, constitui um parâmetro fundamental para o desempenho global do sistema.
- Linhas de endereço – São utilizadas para designar a fonte ou o destino dos dados transferidos pelo barramento de dados. Endereçam também as postas de E/S.
- Linhas de controle – Usadas para controlar o acesso e a utilização das linhas de dados e de endereço. Proporcionam uma maneira de controle já que as linhas de dados e de endereço são compartilhadas por todos os componentes.

### 3.6. Explique a arquitetura de barramento de alto desempenho, também conhecida como Arquitetura Mezanino, e mostre suas vantagens e desvantagens.

Na arquitetura de barramento tradicional, pode-se conectar controladores de E/S diretamente no barramento, Porém não é tão eficaz quando se tem um desempenho maior nos dispositivos de E/S. Uma solução mais eficiente foi a utilização de um barramento de alta velocidade que esteja integrado ao restante do sistema, exigindo apenas um ponte entre o barramento do processador e o de alta velocidade.

### 3.7. Explique a sequência de ações nos diagramas de temporização síncrono e assíncrono dos slides 51, 52 e 53.

O processador coloca um endereço de memória nas linhas de endereço no primeiro ciclo e ativa as linhas de estado logo em seguida. Ativa o endereço logo após a estabilização das linhas de estado. Emite um comando de leitura no início do segundo ciclo.

Um módulo de memória reconhece o endereço e após um ciclo coloca os dados na linha de dados. O processador lê os dados da linha e remove o sinal de leitura. No caso de uma escrita, o processador coloca os dados no início da segundo ciclo e emite um sinal de escrita após a estabilização. O módulo de memória copia a informação das linhas de dados no terceiro ciclo.

#### Temporização assíncrona

Nestes casos, os eventos dependem da ocorrência do evento anterior.

- O processador coloca um sinal de endereço e estado no barramento.
- Após a sua estabilização, é emitido o comando de leitura, indicando sinais válidos de endereço e controle.
- A memória decodifica os endereços e coloca os dados na linha de dados.
- Quando os dados se estabilizam, a memória ativa a linha de confirmação para sinalizar que os dados estão disponíveis.
- Quando os dados forem lidos, é emitido o sinal de leitura, e a memória retira os dados e a linha de confirmação.
- É removida a informação do endereço.

#### Temporização assíncrona simples

- O mestre coloca os dados na linha de dados e sinais nas linhas de endereço e de estado simultaneamente.
- A memória responde ao comando de escrita copiando os dados da linha de dados e ativando a linha de confirmação.
- O mestre remove o sinal de escrita e o módulo de memória remove o sinal de confirmação.

## 4.1. No contexto da memória interna, defina o que é uma Palavra.

Palavra pode ser um conjunto de bits que representam um número inteiro, pode ser também uma unidade máxima de transferência por ciclo ou até mesmo a largura do barramento de uma memória.

## 4.2. Sobre os princípios da memória cache, explique o funcionamento apresentado nas figuras dos slides 20 e 24.

### Slide 20:

**a - Cache única)** A função da memória cache é auxiliar o processador a receber dados da memória de forma mais eficiente e rápida, alguns bytes de endereço de dados ou instruções são carregados nela já antecipando o acesso à memória principal. Caso o endereço não esteja na cache, é necessário esse acesso mais lento.

**b- Vários níveis de cache)** Já com mais níveis de cache, o funcionamento é praticamente o mesmo, algumas diferenças são os tamanhos que são indiretamente proporcionais a velocidade e que dois dos três níveis são partes do processador, assim assumindo maior eficiência na transferência de dados e reduzindo esse gargalo.

### Slide 24:

#### **Operação de leitura de cache.**

- É solicitada e recebida o endereço de leitura da palavra;
- Verifica se o endereço da palavra solicitada está em cache;
- Se estiver, recebe a palavra e entrega ao processador;
- Finaliza a operação.
  - Em caso contrário (o endereço da palavra não esteja em cache);
- Acessa a memória principal no bloco onde está o endereço de leitura;
- Aloca espaço na cache para o bloco de memória;
- Entrega a palavra ao processador;
- Finaliza a operação.

## 4.3. Explique o princípio da localidade.

Como a cache acessa um bloco da memória principal, decorre um aumento da probabilidade das demais consultas de dados por ela feita ser novamente endereçada para o mesmo bloco.

## 4.4. Sobre os elementos da memória cache, fale sobre as funções de mapeamento direto e associativo. Cite pelo menos uma vantagem e desvantagem de cada.

- **Mapeamento direto:** Uma única linha da memória cache é relacionada com um bloco da memória principal.
  - Fácil implementação
  - Blocos implementados em uma linha fixa da cache.
- **Mapeamento associativo:**
  - Qualquer bloco da memória principal pode ser relacionado a uma linha da memória cache.
  - Implementação de circuitos complexas.

## 4.5. Explique a necessidade dos algoritmos de substituição para o mapeamento associativo e porque esses algoritmos não servem quando se trabalha com o mapeamento direto. Ainda sobre os algoritmos de substituição explique rapidamente os algoritmos LRU, FIFO, LFU e Aleatório.

Uma vez que a cache estiver cheia, e um novo bloco for trazido para a cache, um dos blocos existentes precisa ser substituído. Para o mapeamento direto, existe apenas uma linha possível para qualquer bloco em particular e nenhum a escolha é possível.

**LRU:** Substitua aquele bloco no conjunto que permaneceu na cache por mais tempo sem qualquer referência a ele.

**FIFO:** substitua o bloco no conjunto que esteve na cachê por mais tempo.

**LFU:** substitua aquele bloco no conjunto que teve menos referencia.

**Aleatório:** Escolhe uma linha aleatória dentre as linhas candidatas.



#### 4.6. Explique as três técnicas de coerência de cache: observação do barramento, transparência de hardware e memória não cacheável.

- **Monitoramento do barramento:** Cada controlador de memória cache monitora as linhas de endereço para detectar operações de escrita na memória feitas por outros mestres do barramento.
- **Transparência em hardware:** Um hardware adicional é utilizado para assegurar que todas as atualizações feitas na memória principal por meio de uma memória cache sejam refletidas em todas as demais memórias cache. Se um processador modifica uma palavra em sua memória cache, além dessa atualização ser feita na memória principal, as palavras correspondentes nas demais memórias cache são também atualizadas.
- **Memória não-cacheável:** É apenas uma parte da memória principal é compartilhada por mais de um processador e não pode ser associada à memória cache.

#### 4.7. Explique como funciona internamente as duas tecnologias de acesso aleatório, dinâmica e estática, utilizadas para construção de memória cache e principal. Fale sobre as vantagens e desvantagens de cada uma em termos de custo, capacidade e tempo de resposta.

A SDRAM são diferente das memórias DRAM clássicas, que possuem uma interface assíncrona, e por isto respondem tão rápido quanto possível a mudanças nas entradas de controle, a SDRAM possui uma interface síncrona, significando que ela espera pelo sinal do clock antes de responder às entradas de comando e é portanto sincronizada com o barramento do sistema do computador.

#### 4.8. O que são as memórias de acesso sequencial, direto e aleatório?

- **Acesso sequencial:** é feito de forma linear e para chegar a determinada parte e preciso passar pelos seus intermediários.
- **Acesso direto:** blocos tem seu endereço baseado no local físico, para acessar determinada parte da memoria faz-se o acesso direto aos blocos e sequencial ate determinado endereço.
- **Acesso aleatório:** cada local endereçável tem um endereço baseado na localização física, assim qualquer um pode ser acessado de forma direta.

#### 4.9. Qual o relacionamento entre tempo de acesso, custo de memória e capacidade?

- Tempo de acesso mais rápido, maior custo por bit.
- Maior capacidade menor custo por bit.
- Maior capacidade tempo de acesso maior.

#### 4.10. Como o princípio da localidade se relaciona com o uso de múltiplos níveis de memória?

Quanto mais perto do processador os blocos guardados na cache serão os que tem mais chance de serem mapeados mais frequentemente (*princípio de localização temporal*) como os blocos da L1 são menores há mais chance de linhas próximas serem mapeadas (*princípio de localização espacial*).

## 5.1. Explique como acontece a gravação e leitura em um disco magnético.

- **Gravação:** Por uma bobina chamada cabeça. Os pulsos são enviados á cabeça de gravação e os padrões magnéticos são gravados na superfície abaixo.
- **Leitura:** Quando a superfície do disco passa pela cabeça gera uma corrente de mesma polaridade da gravada.

## 5.2. Como o uso de vários discos pode melhorar a confiabilidade ou o tempo de acesso às informações armazenadas?

Com diversos discos, diferentes requisições de E/S podem ser processados em paralelo, desde que os dados requeridos residam em discos separados. Também, uma única requisição de E/S pode ser executada em paralelo se o bloco de dados for distribuídos em vários discos.

Vários discos podem ser utilizados como armazenadores tanto de cópias completas do disco, como o de blocos e bits de paridade. Tais técnicas de redundância e de paridade, trazem segurança ao ambiente.

## 5.3. Defina os termos tempo de busca, atraso rotacional, tempo de acesso e taxa de transferência.

- Tempo de busca: tempo para posicionar a cabeça na trilha
- Atraso rotacional: Tempo para início do setor alcance a cabeça.
- Tempo de acesso: Soma do tempo de busca e do atraso rotacional.
- Tempo de transferência: Tempo necessário para uma operação de leitura ou escrita.

## 5.4. Que características comuns são compartilhadas por todos os níveis de RAID?

RAID é um conjunto de discos físicos vistos pelo Sistema Operacional como uma única unidade lógica.

Os dados são distribuídos pelos discos físicos pelo esquema de *striping*.

A capacidade do disco redundante é usada para armazenar informações de paridade usada para recuperar dados em uma facha de disco.

## 5.5. Defina resumidamente os 6 primeiros níveis de RAID.

- **RAID 0** : Não redundante, intercalação de dados (*striping*), alta capacidade de transferência (paralela);
- **RAID 1**: Redundante, espelhamento de discos, utiliza o dobro de discos;
- **RAID 2**: Acesso paralelo, redundante, paridade via código de *Hamming*, utiliza *m* discos a mais;
- **RAID 3**: Acesso paralelo, paridade de bit intercalada nos discos, utiliza apenas um disco para isso;
- **RAID 4**: Acesso independente, paridade de bloco intercalada, utiliza apenas um disco para isso;;
- **RAID 5**: Acesso independente, paridade de bloco intercalada e distribuída;
- **RAID 6**: Acesso independente, paridade de bloco dupla intercalada e distribuída;

## 5.6. No contexto do RAID, mostre a diferença entre acesso paralelo e acesso independente.

- **Paralelo:** Todos os discos membros participam na execução de cada solicitação de E/S. Normalmente os eixos das unidades individuais são sincronizadas de modo que cada cabeça de disco esteja na mesma posição em cada disco a qualquer instante.
- **Independentes:** Cada disco opera independente, assim solicitações de E/S podem ser feitas separadamente e em paralelo.

# ENTRADA E SAÍDA



## 6.1. Liste três classificações gerais de dispositivos externos ou periféricos.

- Dispositivos voltados para a comunicação com o usuário;
- Dispositivos voltados para a comunicação com a máquina;
- Dispositivos voltados para a comunicação com dispositivos remotos;

## 6.2. Quais as principais funções de um módulo de E/S?

- Controle e temporização.
- Comunicação com o processador.
- Comunicação com o dispositivo.
- Armazenamento temporário (*buffering*) de dados.
- Detecção de erro.

## 6.3. Liste e defina resumidamente três técnicas para realizar E/S.

- **E/S programada:** a E/S ocorre sob o controle direto e contínuo do programa solicitando a operação de E/S.
- **E/S controlada por interrupção:** um programa emite um comando de E/S e depois continua a executar, até que seja interrompido pelo hardware de E/S para sinalizar o final da operação de E/S.
- **Acesso direto à memória (DMA):** um processador de E/S especializado assume o controle de uma operação E/S para mover um grande bloco de dados.

## 6.4. Qual a diferença entre E/S mapeada na memória e E/S independente?

Na E/S mapeada na memória, existe um único espaço de endereçamento para posições de memória e dispositivos de E/S.

Os registradores de dados e de estado dos módulos de E/S são vistos pelo processador como posições de memória, e as mesmas instruções de máquina são usadas para acessar a memória ou os dispositivos de E/S.

Alternativamente, ele pode incluir, além das linhas de leitura e escrita, linhas de comando de entrada e saída. Nesse caso, uma linha de comando especifica se um endereço corresponde a uma posição de memória ou a um dispositivo de E/S. Qualquer endereço pode corresponder a uma posição de memória ou a um dispositivo de E/S.

Com dez linhas de endereço, o sistema pode incluir 1024 posições de memória e 1024 endereços de E/S. Como o espaço de endereçamento de E/S é independente do espaço de endereçamento de memória, esse modo é denominado E/S independente.

## 6.5. Quando ocorre uma interrupção de dispositivo, como o processador determina qual dispositivo emitiu a interrupção?

- Linhas de interrupção individuais;
- Software *polling*;
- *Daisy chain* ou hardware *polling*;
- Arbitragem do barramento (*bus mastering*).

## 6.6. Quando o módulo de DMA toma o controle de um barramento, e enquanto retém o controle do barramento, o que o processador faz?

Depois de enviar o comando, o processador pode continuar executando outras instruções. A execução da operação de E/S é delegada ao módulo de DMA.

Ele transfere diretamente todo o bloco de dados, uma palavra de cada vez, diretamente de ou para a memória, sem a intervenção do processador. Quando a transferência é concluída, o módulo de DMA envia

um sinal de interrupção para o processador. Dessa maneira, o processador é envolvido apenas no início e no fim da transferência do bloco de dados.

## ARITMÉTICA COMPUTACIONAL

**7.1. Explique resumidamente as seguintes representações: sinal-magnitude, complemento a dois, polarizada.**

- Sinal-magnitude:

É alocado o bit mais significativo para o sinal (0 corresponde ao positivo e 1 ao negativo), o restante dos bits são para a representação do número inteiro.

- Complemento a dois:

O bit mais significativo representa o sinal do número, porém os números negativos são compostos do complemento bit a bit do número positivo acrescido de 1.

- Polarizada:

Um valor fixo chamado polarização é subtraído do campo para obter o verdadeiro valor do expoente.

**7.2. Explique como determinar se um número é negativo nas seguintes representações: sinal-magnitude, complemento a dois, viesada.**

- Sinal-magnitude:

O bit mais significativo será 1 (bit de sinal).

- Complemento a dois:

O bit mais significativo será 1 (bit de sinal).

- Viesada:

É um número negativo, se o valor da representação é inferior ao viés.

**7.3. Qual a regra de extensão de sinal para números de complemento de dois?**

Acrescenta-se posições de bit à esquerda e preenche-se esses bits com o valor do bit do sinal original.

**7.4. Como você pode formar a negação de um inteiro na representação de complemento de dois?**

Toma-se o complemento booleano bit a bit do número e acrescenta-se 1 ao número obtido.

**7.5. Em termos gerais, quando a operação de complemento de dois em inteiro de n bits produz o mesmo inteiro?**

Quando o bit mais significativo é igual a 1 e os demais são iguais a 0.

**7.6. Qual é a diferença entre a representação de complemento de dois de um número e o complemento a dois de um número?**

- No complemento a 2 há duas representações para o 0 (-0 e +0).
- No complemento de 2 existe apenas uma representação para o 0, porém existem quantidades de valores positivos e negativos diferentes.

**7.7. Se tratarmos 2 números de complemento de dois como inteiros sem sinal para fins de adição, o resultado é correto se interpretado como um número de complemento de dois. Isso não é verdade para a multiplicação. Por quê?**



O algoritmo para a realização de complemento de 2 envolve simplesmente a adição dos dois números da mesma maneira que para números sem sinal, com um teste de “estouro” (overflow).

Para a multiplicação, se tratarmos os padrões de bits como números sem sinal, a sua magnitude é diferente das versões complemento de dois e por isso a magnitude do resultado será diferente.

#### 7.8. Quais são os quatro elementos essenciais de um número na notação de ponto flutuante?

- Sinal;
- Mantissa;
- Base;
- Expoente.

#### 7.9. Qual é o benefício de usar a representação polarizada para a parte de expoente de um número de ponto flutuante?

A polarização do expoente para a precisão simples é 127, ou seja, o expoente é representado como o expoente real do número somado a 127. Com esta polarização, é possível representar os expoentes reais no intervalo  $[-127 \dots +128]$  e com isso temos que os expoentes polarizados estarão no intervalo  $[0 \dots 255]$ .

A polarização é usada para facilitar as operações de ponto flutuante feitas pelos circuitos, pois ela permite que o expoente seja manipulado apenas como número positivo.

#### 7.10. Quais são as diferenças entre overflow positivo, overflow do expoente e overflow do significado?

- Overflow positivo;  
Quando o número a ser representado supera a quantidade de bits da palavra.
- Overflow do expoente;  
Quando a representação binária do expoente é maior que o tamanho que é alocado.
- Overflow do significado.  
Ocorre quando a adição de 2 significandos de mesmo sinal resulta num “vai-um” no bit mais significativo.

#### 7.11. Quais são os elementos básicos da adição e subtração de ponto flutuante?

Necessidade de alinhamento, Existem quatro fases básicas do algoritmo para a adição e subtração:

- Verificar zeros;
- Alinhar os significandos;
- Somar ou subtrair os significandos;
- Normalizar o resultado.

#### 7.12. Dê um motivo para o uso dos bits de guarda.

Bits de guarda (bit adicional) garantem precisão melhor que metade do bit menos significativo da fração. Sem os bits de guarda há perda de precisão a cada operação.

#### 7.13. Liste quatro métodos alternativos de arredondamento do resultado de uma operação de ponto flutuante.

- Arredondar para o mais próximo: o resultado é arredondado para o número representável mais próximo.

- Arredondar para cima: o resultado é arredondado para cima, na direção de infinito positivo.
- Arredondar para baixo: o resultado é arredondado para baixo, na direção de infinito negativo. Arredondar para 0: o resultado é arredondado na direção de zero.

## CONJUNTO DE INSTRUÇÕES: CARACTERÍSTICAS E FUNÇÕES

### 8.1. Quais os elementos típicos de uma instrução de máquina?

#### Código de operação

Especifica a operação a ser realizada (ADD, E/S). A operação é especificada por um código binário conhecido como **opcode** (*operation code*).

#### Referência à operando fonte

A operação pode envolver um ou mais operandos fontes, ou seja, operandos que são entradas para a operação.

#### Referência à operando destino

A operação pode produzir um resultado.

#### Referência à próxima instrução

Isso diz ao processador onde buscar a próxima instrução após o término de execução desta instrução.

### 8.2. Que tipos de locais podem manter operandos de origem e destino?

#### Memória principal ou virtual

Assim como as referências à próxima instrução, o endereço da memória principal ou virtual precisa ser fornecido.

#### Registrador do processador

Com raras exceções, um processador contém um ou mais registradores que podem ser referenciados por instruções de máquina.

#### Imediato

O valor do operando está contido em um campo da instrução sendo executada.

#### Dispositivos de E/S

A instrução precisa especificar o módulo e o dispositivo de E/S para a operação. Se a E/S mapeada na memória for usada, esse é apenas outro endereço da memória principal ou virtual.

### 8.3. Se uma instrução contém quatro endereços, qual poderia ser a finalidade de cada endereço?

Dois deles para referência de origem de operandos;

Um para referência de destino;

Um para referência de próxima instrução.

### 8.4. Liste e explique resumidamente cinco questões importantes no projeto do conjunto de instruções.

- Repertório de Operação: Quantos e quais operações para fornecer, e como operações complexas deve ser.
- Os tipos de dados: Os vários tipos de dados em que operações são executadas.
- Formato de Instrução: Instrução comprimento (em bits), número de endereços, o tamanho de vários campos, e assim por diante.
- Registros: Número de registros de CPU que pode ser referenciado por instruções, e à sua utilização.

- Endereçamento: O modo ou modos pelos quais o endereço de um operando é especificado.

### 8.5. Que tipos de operandos são típicos nos conjuntos de instrução de máquina?

Endereços, números, caracteres e dados lógicos.

### 8.6. Qual é o relacionamento entre o código de caracteres IRA e a representação decimal agrupada?

Para o padrão de bits IRA 011XXXX, os dígitos de 0 a 9 são representados por seu equivalentes binários, 0000 a 1001, nos 4 bits mais significativos (à direita). Esta é o mesmo código como a representação decimal-binária.

### 8.7. Qual a diferença entre um deslocamento aritmético e um deslocamento lógico?

No deslocamento lógico, os bits de uma palavra são deslocados para a esquerda ou direita. Numa extremidade, o bit deslocado para fora está perdido. Na outra extremidade, um 0 é deslocado dentro.

A operação de deslocamento aritmético trata os dados como um inteiro com sinal e não muda esse bit de sinal. Em um certo deslocamento aritmético, o bit de sinal é replicado para a posição pouco à sua direita.

Em um deslocamento para a esquerda, no deslocamento aritmético, desvio à esquerda lógico é realizado em todos os bits, mas o bit de sinal, o que é retidos.

### 8.8. Por que são necessárias instruções de transferência de controle?

Na utilização prática de computadores, é essencial que seja capaz de executar cada instrução mais de uma vez e, talvez, milhares de vezes.

Ela pode exigir milhares ou talvez milhões de instruções para implementar uma aplicação. Este seria impensável se cada instrução tiver que ser escrita separadamente. Se uma tabela ou uma lista de itens é processada, um ciclo do programa é necessário. Uma sequência de instruções é executada repetidamente para processar todos os dados.

Praticamente todos os programas envolvem alguma tomada de decisão. Queremos que o computador faça uma coisa, se um condição se mantém, e outra coisa, se outra condição se mantém.

Para compor corretamente um programa de computador grande ou mesmo de médio porte é uma tarefa bastante difícil. Mas pode ser amenizada se existir mecanismos para quebrar a tarefa em menores tamanhos que podem ser trabalhados em um de cada vez.

### 8.9. Liste e explique resumidamente duas maneiras comuns de gerar a condição a ser testada em uma instrução de desvio.

Primeiro, a maioria das máquinas proporcionam um código de condição de 1 bit ou bits múltiplos, que é definido como o resultado de algumas operações.

Outra abordagem que pode ser utilizada com um período de três endereço formato de instrução é realizar uma comparação e especificar um ramo da mesma instrução.

### 8.10. O que significa o termo aninhamento de procedimentos?

Esse termo se refere a ocorrência de um procedimento chamado dentro de outro procedimento.

### 8.11. Liste três locais possíveis para armazenar o endereço de retorno para um retorno de procedimento.

Registrador, início do procedimento e topo da pilha.

### 8.12. O que é um procedimento reentrante?

Um procedimento de reentrada é aquele em que é possível ter várias ligações abertas para ele ao mesmo tempo.

### 8.13. O que é notação polanesa invertida?

Nesta notação, o operador segue (localiza-se após) os seus dois operandos.

### 8.14. Qual a diferença entre big-endian e little-endian?

Um valor de múltiplos bytes armazenado com o byte mais significativo no menor endereço numérico está armazenado como big-endian.

Um valor de múltiplos bytes armazenado com o byte mais significativo no maior endereço numérico está armazenado na como little-endian.

## MODOS E FORMATOS DE ENDEREÇAMENTO

### 9.1. Defina resumidamente endereçamento imediato.

É considerada a forma mais simples de endereçamento, no qual o valor do operando está presente na instrução.

### 9.2. Defina resumidamente endereçamento direto.

É uma forma simples de endereçamento, onde o campo de endereço contém o endereço efetivo do operando.

### 9.3. Defina resumidamente endereçamento indireto.

Como o tamanho do endereço é normalmente menor que o tamanho da palavra, (limita o intervalo de endereços), o endereçamento indireto possui o campo de endereço se referindo ao endereço de uma palavra, o qual por sua vez, possui contém o endereço completo do operando.

### 9.4. Defina resumidamente endereçamento de registradores.

Semelhante ao endereçamento direto. Difere que o campo de endereço se refere a um registrador em vez da memória principal.

### 9.5. Defina resumidamente endereçamento indireto por registradores.

De maneira análoga ao endereçamento indireto, possui o campo de endereço se referindo ao endereço de um registrador.

### 9.6. Defina resumidamente endereçamento por deslocamento.

É a união de endereçamento direto e indireto, onde o endereçamento indireto (*opcode*), refere-se a um registrador cujo os conteúdos são adicionados ao registrador direto.

### 9.7. Defina resumidamente endereçamento relativo.

O registrador implicitamente referenciado é o contador de programa (PC). O endereço efetivo é o deslocamento relativo ao endereço da instrução.

### 9.8. Qual a vantagem da autoindexação?

É típico que existe uma necessidade de incrementar ou diminuir o registro de índice depois de cada referência a ele. Como esta é uma operação muito comum, alguns sistemas fazem isso automaticamente como parte do mesmo ciclo de instrução.

### 9.9. Qual a diferença entre pós-indexação e pré-indexação?

Essas são duas formas de endereçar, tanto das que envolvem endereçamento indireto e indexação. Com pré-indexação, a indexação é realizada antes do endereçamento indireto. Com a pós-indexação, a indexação é realizada após o endereçamento.

### 9.10. Quais fatores devem ser levados em conta para determinar o uso de bits de endereçamento de uma instrução?

- Número de modos de endereçamento:
- Número de operandos:
- Registrador vs memória:
- Número de conjuntos de registros:
- Intervalo de endereço:
- Granularidade de endereço:

Página 341

### 9.11. Quais as vantagens e as desvantagens de usar o formato da instrução de tamanho variável?

Vantagens: É fácil dar um grande repertório de *opcodes*, com diferentes comprimentos. O endereçamento pode ser mais flexível, com várias combinações de registros e referências de memória além de modos de endereçamento.

Desvantagens: um aumento na complexidade do processador.

## ESTRUTURA DE FUNÇÃO DO PROCESSADOR

### 10.1. Quais os papéis são desempenhados pelos registradores do processador?

#### Registradores Visíveis ao usuário:

Estes permitem a linguagem de máquina pelos programadores para minimizar as referências de memória principal, otimizando o uso de registradores.

#### Registradores de controle e de estado:

Estes são usados pela unidade de controle para controlar o funcionamento da CPU e, por programas de sistema operacionais privilegiados para controlar a execução de programas.

### 10.2. Quais categorias de dados são normalmente suportadas pelos registradores visíveis ao usuário?

Propósito geral, dados, endereços e código condicionais.

### 10.3. Qual é a função de códigos condicionais?

Códigos de condição são bits definidos pela CPU como o resultado das operações. Por exemplo, uma operação aritmética pode produzir um positivo, negativo, zero, ou um "estouro". Além do resultado em si que está sendo armazenado em um registro ou memória, um código de condição é também definido. O código pode ser posteriormente testado como parte de um operação de ramificação condicional.

### 10.4. O que é a palavra de estado do programa?

Todos os projetos de CPU incluem um registo ou conjunto de registros, muitas vezes conhecida como a palavra de estado de programa (PSW), que contém informações de status. A PSW normalmente contém códigos de condição além de outras informações de status.

**10.5. Por que um pipeline de instrução de dois estágios dificilmente diminuirá o tempo do ciclo da instrução pela metade, quando comparado a um sistema sem pipeline?**

- O tempo de execução geralmente será maior do que o tempo de busca. execução irá envolver a leitura e armazenamento de operandos e do desempenho de alguma operação. Assim, a fase de buscar pode ter de esperar por algum tempo antes que ele possa esvaziar o buffer.
- A instrução de desvio condicional faz com que o endereço da próxima instrução a ser desconhecido buscado. Assim, a fase de buscar deve esperar até receber o próximo endereço de instrução a partir da fase de execução. A etapa de execução pode, então, aguardar enquanto a próxima instrução é buscada.

**10.6. Liste e explique resumidamente várias formas em que um pipeline de instruções pode lidar com instruções de desvio condicionais.**

- Múltiplos fluxos;

Uma abordagem de força bruta é replicar as partes iniciais do pipeline e permitir que o pipeline busque as duas instruções.

- Busca antecipada do alvo do desvio;

Quando um desvio condicional é reconhecido, o alvo do desvio é pré-obtido, além de a instrução que segue o desvio. Este alvo é então guardado até que a instrução de desvio é executada. Se o desvio é tomado, o alvo já foi pré-obtido.

- Buffer de laço de repetição;

Um buffer de laço é uma pequena memória, alta velocidade mantida pelo pipeline de busca e contendo  $n$  instruções mais recentes.

- Previsão de desvio.

É possível melhorar o desempenho do pipeline, reorganizando automaticamente instruções dentro de um programa, de modo que instruções de desvio ocorrem mais tarde do que realmente desejado.

**10.7. Como são usados os bits de histórico para previsão de desvio?**

Um ou mais bits que refletem a o histórico recente da instrução podem ser associados com cada instrução de desvio condicional. Estes bits são referidos como chave tomada/não tomada feita que direciona o processador para fazer uma decisão em particular.

## COMPUTADORES COM CONJUNTO REDUZIDO DE INSTRUÇÕES (RISC)

**11.1. Quais são algumas das características peculiares típicas da organização RISC?**

- Um conjunto limitado de instruções com um formato fixo;
- Grande número de registros ou o uso de um compilador que otimiza a utilização do registrador;
- Ênfase otimizar o pipeline de instrução.

**11.2. Explique brevemente duas abordagens básicas para minimizar operações registrador-memória em máquinas RISC.**

Duas abordagens básicas são possíveis, uma baseada em software e o outro em hardware. A abordagem de software é contar com o compilador para maximizar o uso de registradores. O compilador tentará alocar os registros para as variáveis que irão ser usadas mais em um determinado período de

tempo. Esta abordagem requer a utilização de sofisticados algoritmos de programa de análise. A abordagem de hardware é simplesmente usar mais registradores para que mais variáveis pode ser mantidas em registros por mais tempo.

**11.3. Se um buffer circular de registradores é usado para tratar variáveis locais para procedimentos aninhados, descreva duas abordagens para lidar com variáveis globais.**

- Variáveis declaradas como global em uma HLL podem ser atribuídas a posições de memória pelo o compilador, e todas as instruções de máquina que fazem referência a estas variáveis vão usar operandos de memória de referência.
- Incorporar um conjunto de registros globais no processador. Esses registros seriam fixados em número e disponível para todos procedimentos

**11.4. Quais são algumas características típicas de uma arquitetura de conjunto de instruções RISC?**

- Uma instrução por ciclo.
- Operações registrador-para-registrador.
- Modos de endereçamento simples.
- Formatos simples de instrução.

**11.5. O que é um desvio atrasado?**

Desvio atrasado é uma forma de aumentar a eficácia do pipeline, faz uso de um desvio que não terá efeito até que após a execução da instrução seguinte.

## OPERAÇÕES DA UNIDADE DE CONTROLE

**12.1. Explique a diferença entre a sequência de escrita e a sequência de tempo de uma instrução.**

A operação de um computador, na execução de um programa, é composto por uma sequência de ciclos de instrução, com uma instrução por ciclo de máquina.

Essa sequência de ciclos de instrução não é necessariamente a mesma que a sequência de escrita de instruções que compõem o programa, por causa da existência de ramificação de instruções.

A execução real de instruções segue uma sequência de tempo de instruções.

**12.2. Qual é a relação entre instruções e micro-operações?**

A micro-operação é uma operação elementar de CPU, realizada durante um pulso de *clock*. Uma instrução consiste em uma sequência de micro-operações.

**12.3. Qual é função geral de uma unidade de controle do processador?**

A uma unidade de controle do processador executa duas funções:

- Faz com que o processador a execute micro-operações na seqüência adequada, determinada pelo programa.
- Gera os sinais de controle que fazem com que cada micro-operação seja executada.

**12.4. Defina um processador em três passos que leva caracterização da unidade de controle.**

- Define os elementos básicos do processador.
- Descreve as micro-operações que o processador executa.
- Determina as funções que a unidade de controle deve executar e fazer com que as micro-operações sejam realizadas.

**12.5. Quais tarefas básicas uma unidade de controle efetua?**

- **Sequenciamento:** A unidade de controle faz com que o processador para percorrer uma série de

micro-operações na sequência correta, com base no programa que está sendo executado.

- **Execução:** A unidade de controle faz com que cada micro-operação a ser executada.

## 12.6. Forneça uma lista típica de entradas e saídas de uma unidade de controle.

**As entradas são:**

- **Relógio:** Esta é a forma como a unidade de controle a unidade de controle "mantém o tempo." provoca uma micro- operação (ou um conjunto de micro-operações simultâneas ) para ser realizada para cada impulso de relógio . Isso às vezes é referido como o processador o tempo de ciclo , ou o tempo de ciclo de relógio .
- **Registro Instrução:** O código de operação da corrente instrução é usada para determinar qual micro-operações serão realizadas durante o ciclo de execução.
- **Flags:** Estas medidas são necessárias pela unidade de controle para determinar o estado do processador e os resultados das operações anteriores da ULA.
- **Os sinais de controle:** A parte do barramento de controle proporciona sinais para o unidade de controle, tais como sinais de interrupção e reconhecimentos.

**As saídas são:**

- Os sinais de controle dentro do processador:
  - Estes são dois tipos:
    - Aqueles que produzem dados para ser movidos de um registro para outro;
    - Aqueles que ativam funções específicas da ULA.
- Os sinais de controle para controlar o barramento:
  - Estes também são de dois tipos :
    - Sinais de controle para a memória;
    - Sinais de controle para os módulos de E/S.

## 12.7. Relacione três tipos de sinais de controle.

- Aqueles que ativam funções da ULA.
- Aqueles que ativam um caminho de dados.
- Aqueles que são os sinais do barramento do sistema externo ou outra interface externa.

## 12.8. Explique resumidamente o que significa uma implementação por hardware de uma unidade de controle.

Em uma implementação por hardware, a unidade de controle é essencialmente uma combinação de circuitos. Seus sinais lógicos de entrada são transformados em um conjunto de sinais lógicos de saída, que são os sinais de controle.

# CONTROLE MICROPROGRAMADO

## 13.1. Qual a diferença entre uma implementação por hardware e uma implementação microprogramada de uma unidade de controle?

Uma unidade de controle implementada por hardware é um circuito combinatório, no qual os sinais de entrada lógicos transformam-se em um conjunto de sinais lógicos de saída que funcionam como o controle sinais.

Numa unidade de controle microprogramada, a lógica é especificada por um microprograma. Um microprograma consiste em uma sequência de instruções simples em um linguagem de microprogramação.

### 13.2. Como é interpretada uma microinstrução horizontal?

Para executar uma microinstrução, ligue todas as linhas de controle indicados por um 1 bit; deixe de fora todas as linhas de controle indicado pelo bit 0. Os sinais de controle resultantes irão ativar uma ou mais micro-operações a serem realizadas.

Se a condição indicada pela condição é falsa, executar a próxima microinstrução em sequência.

Se a condição indicada pelos bits de condição é verdadeira, o próxima microinstrução a ser executada é indicado no campo de endereço.

### 13.3. Qual é o propósito de uma memória de controle?

A memória de controle contém o conjunto de microinstruções que definem a funcionalidade da unidade de controle.

### 13.4. Qual é a sequência típica na execução de uma microinstrução horizontal?

As microinstruções em cada rotina devem ser executadas sequencialmente. cada rotina termina com um ramo ou instrução de salto, indicando para onde ir.

### 13.5. Qual é a diferença entre microinstruções horizontais e verticais?

Em uma microinstrução horizontal cada bit no campo de controle é anexado a um controle linha. Em uma microinstrução vertical, um código é usado para cada ação a ser executada e o decodificador converte este código em sinais de controle individuais.

### 13.6. Quais são as tarefas básicas executadas por uma unidade de controle microprogramada?

- **Sequenciamento de Microinstrução:** Obter a próxima microinstrução a partir da memória de controle.
- **Execução de Microinstrução:** Gere os sinais de controle necessários para executar a microinstrução.

### 13.7. Qual é a diferença entre microinstruções empacotadas e não empacotadas?

O grau de empacotamento refere-se ao grau de identidade entre uma dada tarefa de controle e partes de microinstruções específicas. Como os bits se tornam mais compactados, um determinado número de bits contém mais informações.

Uma microinstrução não empacotada tem nenhuma codificação além de atribuição de funções individuais de bits individuais.

### 13.8. Qual é a diferença entre programação hardware e software?

- Programação hardware são geralmente fixos e compromete a memória só de leitura.
- Programação software são mais mutáveis e são sugestivos de microprogramação por usuário.

### 13.9. Qual é a diferença entre codificação funcional e de recursos?

**Codificação funcional** identifica funções dentro da máquina e designa campos por tipo de função. Por exemplo, se diversas fontes podem ser utilizadas para transferir dados para o acumulador, um campo pode ser designado para essa finalidade, com cada código que especifica uma fonte diferente.

**Codificação de Recursos** vê a máquina como um consistido de um conjunto de recursos independentes e dedica um campo para cada (por exemplo, E/S, memória, ULA).

### 13.10. Enumere algumas aplicações comuns de microprogramação.

- Concepção de computadores.
- Emulação.
- Suporte ao sistema operacional.
- Concepção de dispositivos para fins especiais.
- Apoio linguagem de alto nível.
- Microdiagnósticos.

## OPERAÇÕES DA UNIDADE DE CONTROLE

### 14.1. Relacione e defina brevemente três tipos de organização de sistemas computacionais?

- **Uma instrução, dados individuais (SISD):** Um único processador executa um único fluxo de instruções para operar nos dados armazenados em uma única memória.
- **Uma instrução, vários dados (SIMD):** Um único controle de instrução de máquina, execução simultânea de um número de elementos de processamento síncrono. Cada elemento de processamento de dados tem uma memória associada, de modo que cada instrução é executada em um conjunto diferente de dados pelos diferentes processadores.
- **Instrução múltiplo, dados múltiplos (MIMD):** Um conjunto de processadores executa simultaneamente sequências de instruções diferentes sobre diferentes conjuntos de dados.

### 14.2. Quais são as principais características de um SMP?

- Existem dois ou mais processadores similares de capacidade comparável.
- Estes processadores compartilham a mesma memória principal e interfaces de E/S e estão interligados por um barramento externo, de tal forma que o tempo de acesso à memória é aproximadamente o mesmo para cada processador.
- Todo o acesso processadores ação para de dispositivos E/S, ou através dos mesmos canais ou através de diferentes canais que fornecer caminhos para o mesmo dispositivo.
- Todos os processadores podem executar as mesmas funções (daí o termo simétrico).
- O sistema é controlado por um sistema operativo integrado no sistema que proporciona a interação entre os processadores e seus programas de trabalho, tarefas, arquivos e elementos de dados os níveis.

### 14.3. Quais são algumas vantagens potenciais de um SMP comparado com um uniprocessador?

- **Desempenho:** Se o trabalho a ser feito por um computador pode ser organizado de modo que algumas partes da instrução podem ser feita em paralelo, em seguida, um sistema com múltiplos processadores trará maior desempenho do que um com um único processador do mesmo tipo.
- **Disponibilidade:** Em um multiprocessador simétrico, todos os processadores podem executar as mesmas funções, a falha de um único processador de não interrompe a máquina. Em vez disso, o sistema pode continuar a funcionar com desempenho reduzido.
- **Crescimento incremental:** Um utilizador pode melhorar o desempenho de um sistema, adicionando um processador adicional.
- **Escala:** Os fornecedores podem oferecer uma gama de produtos com diferentes características de preço e desempenho com base no número de processadores configurados no sistema.

### 14.4. Quais são algumas das principais questões a respeito de projeto de um sistema operacional para um SMP?

- **Processos concorrentes simultâneos:** Rotinas de SO precisam ser reentrante para permitir vários processadores para executar o mesmo código ao mesmo tempo. com múltiplos processadores que executam as mesmas ou diferentes partes do sistema operacional, tabelas de sistema operacional e estruturas de gestão deve ser gerida de forma adequada para evitar um conflito ou operações

inválidas.

- **Agendamento** : Qualquer processador pode realizar o agendamento , para que os conflitos sejam evitados. O programador deve atribuir processos prontos para processadores disponíveis .
- **Sincronização**: Com vários processos ativos ter acesso potencial aos espaços de endereços compartilhados ou recursos de E/S comuns, devem ser tomados cuidados para fornecer eficaz sincronização.
- **Confiabilidade e tolerância a falhas** : O sistema operacional deve fornecer métodos de reconhecimento de falha no processador. o programador e outras partes do sistema operativo tem de reconhecer a perda de um processador e tabelas de gerenciamento.

#### 14.5. Qual é a diferença entre esquemas de coerência de cache por software e hardware?

Esquemas de coerência de cache por Software tentam evitar a necessidade de adicional circuito de hardware e lógica, baseando-se no compilador e sistema operacional para lidar com o problema. Em esquemas de hardware, a lógica de coerência de cache é implementado em hardware.

#### 14.6. Qual é o significado de cada um dos quatro estados do protocolo MESI?

**Modificado**: A linha no cache foi modificada (diferente de memória principal) e está disponível apenas neste cache.

**Exclusivo**: A linha no cache é o mesmo que na memória principal e não está presente em nenhum outro cache.

**Compartilhada**: A linha no cache é o mesmo que o da memória principal e pode estar presente em outro cache.

**Inválido**: A linha no cache não contém dados válidos.

#### 14.7. Quais são alguns dos principais benefícios de clusters?

- Escalabilidade absoluta: É possível criar grandes aglomerados que superam o poder do mesmo os maiores máquinas autônomas.
- Escalabilidade incremental: Um cluster é configurado de tal maneira que é possível adicionar novos sistemas do aglomerado em pequenos incrementos. Assim, um usuário pode começar com um sistema modesto e expandi-lo de acordo com a necessidades, sem ter que passar por uma grande atualização em que um existente pequeno sistema é substituído com um sistema maior.
- Alta disponibilidade: Porque cada nó de um cluster é um computador autônomo, a falha de um nó não significa perda de serviço.
- Preço Superior / desempenho: Usando blocos de construção de commodities, é possível montar um cluster com poder de computação igual ou superior a uma única grande máquina, a um custo muito menor.

#### 14.8. Qual é a diferença entre *failover* e *failback*?

A função de comutação um aplicativos e dados sobre os recursos a partir de uma falha de sistema para um sistema alternativo no *cluster* é chamado de *failover*. Uma função relatada a restauração de aplicativos e recursos de dados para o sistema original, uma vez que ele tenha sido fixado; este é referido como o retorno de falha.

#### 14.9. Quais são as diferenças entre UMA, NUMA e CC-NUMA?

- Acesso à memória uniforme (UMA): Todos os processadores têm acesso a todas as partes da memória principal utilizando cargas e lojas. O tempo de acesso à memória de um processador para todas as regiões de memória é a mesma. Os tempos de acesso por diferentes processadores são os mesmos.
- Acesso não uniforme a memória (NUMA): Todos os processadores tem acesso a todas as partes da memória principal. O tempo de acesso a memória é diferente, dependendo de qual região da

memória principal é acessada.

- **cache-NUMA coerente (CC-NUMA):** Um sistema NUMA em que a coerência de cache é mantido entre os caches dos vários processadores.

## COMPUTADORES MULTICORE

### 15.1. Resuma a diferença entre pipeline de instruções simples, superescalar e multithreading simultâneo.

- **Pipeline:** instruções individuais são executados através de um pipeline de estágios assim que, enquanto uma instrução está sendo executada em um estágio do pipeline, uma outra instrução está sendo executada em outro estágio do pipeline.
- **Superscalar:** Vários pipelines são construídos por meio da replicação de recursos de execução. Isso permite a execução paralela de instruções em pipelines paralelos.
- **Multithreading simultânea (SMT):** Bancos de registros são replicados para que múltiplas threads possam compartilhar o uso dos recursos de pipelines.

### 15.2. Dê várias razões para a escolha dos projetistas para migrar para uma organização multicore em vez de aumentar o paralelismo dentro de um único processador

No caso do pipeline, simples pipelines de 3 estágios foram substituídos por pipelines com 5 estágios, e, em seguida, muito mais estágios, com algumas implementações com mais de um dúzia deles. Há um limite prático para o quão longe esta tendência pode ser tomada, porque com mais estágios, há a necessidade de mais lógica, mais interconexões, e mais sinais de controle.

Com organização superescalar, aumentos de desempenho pode ser conseguido através do aumento do número de pipelines paralelos. Novamente, existem reduções de desempenho conforme o número de pipelines aumenta. Uma maior lógica é necessária para gerenciar os riscos de instrução.

Eventualmente, um único segmento de execução atinge o ponto onde os perigos e dependências de recursos impede a plena utilização dos múltiplos pipelines disponíveis. Este mesmo ponto de retornos decrescentes é alcançado com SMT, como a complexidade do gerenciamento de várias threads sobre um conjunto de pipelines limita o número de linhas e número de pipelines que podem ser utilizada de forma eficaz.

### 15.3. Por que há uma tendência para se aumentar a fração da área do chip para memória cache?

A memória cache usa menos energia do que a lógica.

### 15.4. Relacione alguns exemplos de aplicações que se beneficiam diretamente da habilidade de aumentar rendimento com número de núcleos.

- Aplicações nativas multi-threaded : são caracterizado por ter um pequeno número de processos.
- Aplicações multi- processo : aplicações multi- processo são caracterizados pela presença de muitos processos single-threaded . aplicações de banco de dados.
- Aplicações Java : aplicações de máquinas virtuais.
- Aplicações de várias instâncias : Execução de várias instâncias do aplicativo em paralelo.

### 15.5. No nível mais alto, quais são as principais variáveis do projeto em uma organização multicore?

- O número de processadores de núcleo no chip
- O número de níveis de memória cache
- A quantidade de memória cache que é compartilhada

### 15.6. Relacione algumas vantagens de cache L2 compartilhada entre núcleos comparada com caches L2 separadas dedicadas para cada núcleo?

- Pode reduzir as taxas gerais de perda.
- Os dados compartilhados por vários núcleos não são replicados no nível de cache compartilhado.

- Com adequados algoritmos de substituição, a quantidade de cache compartilhado alocado para cada núcleo é dinâmico, de modo que os segmentos que têm uma menor localidade pode empregar mais cache.
- Comunicação Interprocessador é fácil de implementar, através de memória compartilhada.
- O uso de um cache L2 compartilhado limita o problema de coerência para o nível de cache L1, o que pode proporcionar alguma vantagem adicional de desempenho.

### REFERÊNCIAS BIBLIOGRÁFICAS

STALLINGS, William. Arquitetura e organização de computadores 5.ed. São Paulo: Pearson Prentice Hall, 2002.