INSTITUTO FEDERAL SUL-RIO-GRANDENSE
UNIVERSIDADE ABERTA DO BRASIL
Programa de Fomento ao Uso das
TECNOLOGIAS DE COMUNICAÇÃO E INFORMAÇÃO NOS CURSOS DE GRADUAÇÃO - TICS

Arquitetura e organização de computadores

Lisandro Lemos Machado

Ministério da Educação
GOVERNO FEDERAL BRASIL PAÍS RICO É PAÍS SEM POBREZA

Apostila de Arquitetura e Organização de Computadores

MACHADO, Lisandro Lemos.

2011/2

Produzido pela Equipe de Produção de Material Didático da
Universidade Aberta do Brasil do Instituto Federal Sul-rio-grandense

EQUIPE DE PRODUÇÃO DE MATERIAL DIDÁTICO – UAB/IFSUL

Lisiane Corrêa Gomes Silveira
GESTORA DA EQUIPE DE DESIGN

Denise Zarnottz Knabach
Felipe Rommel
Helena Guimarães de Faria
Lucas Quaresma Lopes
EQUIPE DE DESIGN

Catiúcia Klug Schneider
GESTORA DE PRODUÇÃO DE VÍDEO

Gladimir Pinto da Silva
PRODUTOR DE ÁUDIO E VÍDEO

Marcus Freitas Neves
EDITOR DE VÍDEO

João Eliézer Ribeiro Schaun
GESTOR DO AMBIENTE VIRTUAL DE APRENDIZAGEM

Giovani Portelinha Maia
GESTOR DE MANUTENÇÃO E SISTEMA DA INFORMAÇÃO

Carlo Camani Schneider
Efrain Becker Bartz
Jeferson de Oliveira Oliveira
Mishell Ferreira Weber
EQUIPE DE PROGRAMAÇÃO PARA WEB

SUMÁRIO

GUIA DIDÁTICO GD

APRESENTAÇÃO

Prezado (a) aluno (a),

Bem-vindo(a) ao espaço de estudo da Disciplina de Arquitetura e Organização de Computadores.

Ao final desta disciplina você deverá será capaz de compreender a arquitetura e a organização dos computadores através do estudo das características e funcionalidades de seus componentes.

Nesta disciplina, estudaremos as características e funcionalidades de componentes, visando compreender a arquitetura e a organização de computadores. Ela também se utiliza de conceitos de Fundamentos Matemáticos da Computação e elementos de Lógica.

Nas unidades, serão abordados os seguintes conteúdos: Conceitos básicos, Unidade Central de Processamento, Sistema de memória, Entrada e saída e Tipos de organização de computadores.

Através dela, pretende-se prover a base teórica sobre o funcionamento dos componentes e a arquitetura que formam o computador, alinhando-se com conceitos trabalhados em disciplinas como Montagem e Manutenção de Computadores, Sistemas Operacionais e outras que referenciarem o funcionamento do computador em seus conteúdos.

Bom trabalho para todos!

Objetivo Geral

Compreender a arquitetura e a organização dos computadores através do estudo das características e funcionalidades de seus componentes.

Habilidades

- Conhecer os conceitos básicos relativos ao funcionamento de computadores.
- Compreender o funcionamento dos componentes da Unidade Central de Processamento.
- Identificar os tipos de memória que compõem o sistema de memórias existente, bem como suas características.
- Conhecer os dispositivos de entrada e saída e a forma como realizam suas tarefas.
- Reconhecer os diferentes tipos de organização de computadores.

Metodologia

A disciplina será desenvolvida em 60h através do Ambiente Virtual de Aprendizado Moodle. Nele, serão disponibilizados os materiais utilizados na disciplina, contando ainda com recursos de fórum, e-mail, textos de apoio e exercícios on-line.

Avaliação

A avaliação se dará mediante a participação nos fóruns e a realização das atividades propostas, tanto presenciais como a distância.

Programação

Primeira semana:

As atividades a serem desenvolvidas na primeira semana são:

1. Leitura e estudo do Conteúdo: Conceitos básicos
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Segunda semana:

As atividades a serem desenvolvidas na segunda semana são:

1. Leitura e estudo do Conteúdo: Unidade Central de Processamento
2. Realização de Atividades.
3. Participação em Fórum de discussão.

Terceira semana:

As atividades a serem desenvolvidas na terceira semana são:

1. Leitura e estudo do Conteúdo: Processador hipotético.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Quarta semana:

As atividades a serem desenvolvidas na quarta semana são:

1. Leitura e estudo do Conteúdo: Simulador de processador.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Quinta semana:

As atividades a serem desenvolvidas na quinta semana são:

1. Leitura e estudo do Conteúdo: Simulador de processador.
2. Realização de atividades.
3. Fórum de discussão para dúvidas.

Sexta semana:

As atividades a serem desenvolvidas na sexta semana são:

1. Leitura e estudo do Conteúdo: Hierarquia do sistema de memória.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Sétima semana:

As atividades a serem desenvolvidas na sétima semana são:

1. Leitura e estudo do Conteúdo: Memória Principal.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Oitava semana:

As atividades a serem desenvolvidas na oitava semana são:

1. Leitura e estudo do Conteúdo: Memória cachê.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Nona semana:

As atividades a serem desenvolvidas na nona semana são:

1. Leitura e estudo do Conteúdo: Memória Secundária.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Décima semana:

As atividades a serem desenvolvidas na décima semana são:

1. Leitura e estudo do Conteúdo: Entrada e Saída.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Décima primeira semana:

As atividades a serem desenvolvidas na décima primeira semana são:

1. Leitura e estudo do Conteúdo: Arquiteturas paralelas e Multiprocessamento.
2. Realização de Atividades.
3. Fórum de discussão para dúvidas.

Décima segunda semana:

As atividades a serem desenvolvidas na décima segunda semana são:

1. Participação em fórum de discussão.

Referências

CARTER, Nicholas. **Arquitetura de Computadores**. São Paulo: Bookman, 2003.

HENNESSY, John L**. Arquitetura de Computadores: uma abordagem quantitativa**. 3ª ed. Rio de Janeiro: Campus, 2003.

MORIMOTO, Carlos Eduardo. **Hardware, o guia definitivo**. Porto Alegre: Sul Editores, 2009.

MONTEIRO, Mário A. **Introdução à organização de computadores**. 5ª ed.. Rio de Janeiro: LTC, 2007.

PATTERSON, D. A.; HENNESSY, J. L. **Projeto e Organização de Computadores: A Interface Hardware/Software**. 3ª ed.. Rio de Janeiro: Campus/Elsevier, 2005.

STALLINGS, W. **Arquitetura e Organização de Computadores**. 5ª ed.. São Paulo: Prentice Hall, 2002.

TANENBAUM, A. S. **Organização Estruturada de Computadores**. 5ª ed.. Prentice-Hall, 2006.

TORRES, Gabriel. **Hardware: curso completo.** 4ª ed. Rio de Janeiro: Axcel Books, 2001.

VASCONCELOS, Laércio. **Hardware na prática**. Rio de Janeiro: edição do autor, 2007.

WEBER, Raul Fernando. **Arquitetura de computadores pessoais**. Porto Alegre: Sagra Luzzatto, 2001.

WEBER, Raul Fernando. **Fundamentos de arquitetura de computadores**. Porto Alegre: Bookman; UFRGS, 2008.

Currículo Professor-Autor

Possui graduação em Ciência da Computação pela Universidade de Passo Fundo (2003), Especialização em Informática Aplicada à Educação (2005) e Formação Pedagógica de Docentes para atuação na Educação Profissional (2005) - Antigo Esquema I, ambos pela mesma instituição. É mestrando em Educação pela Universidade de Passo Fundo e atualmente é professor do Instituto Federal Sul-Riograndense (IF Sul), atuando na graduação e ensino técnico. Tem experiência na área de Ciência da Computação, atuando principalmente nos seguintes temas: informática na educação, ensino técnico, recursos didático-pedagógicos, objetos de aprendizagem, ensino-aprendizagem e interatividade.

Lattes: <http://lattes.cnpq.br/2047240300453280 >

TICs

A

Conceitos básicos

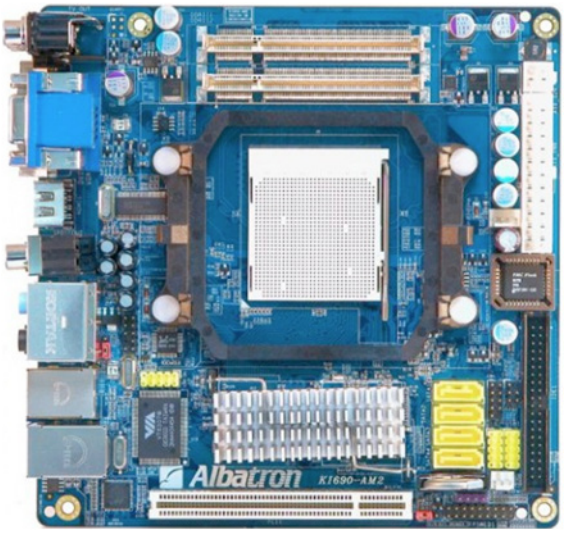
Unidade A
Arquitetura e Organização de Computadores

UNIDADE A

CONCEITOS BÁSICOS

Introdução:

Antes de iniciarmos o nosso módulo, observe a seguinte imagem:



Com certeza muitos conhecem este elemento, é um modelo de placa-mãe que encontramos em nossos computadores. Mas você consegue identificar os componentes que são conectados nela?

Podemos destacar, dentre os componentes que podemos encontrar conectados em uma placa-mãe, o processador, módulos de memória RAM, disco rígido, placa de vídeo, placas de expansão, etc.

Sabemos que cada um dos componentes citados acima desempenha um papel específico no funcionamento do computador. Mas você sabe como cada um deles funciona internamente para desempenhar suas funções?

Vivemos em uma sociedade rodeada pelas mais diversas tecnologias que são utilizadas nas mais variadas atividades. Assim, "em relação ao número de usuários e de unidades instaladas, computadores pessoais são, sem dúvida, os computadores mais populares. Seu grande sucesso deve-se ao baixo custo, à flexibilidade de serem adaptados a um grande número de aplicações, à grande quantidade de software disponível e à facilidade de encontrar profissionais familiarizados com sua arquitetura" (WEBER, 2008, p. 1). Para a compreensão sobre o funcionamento dos computadores, é necessário o estudo dos elementos que compõem sua arquitetura, sendo importante para aqueles que o utilizam que conheçam as relações existentes internamente e entendam o papel dessas no desempenho da máquina.

Informação

A finalidade básica de um computador é a de realizar operações com informações em formato digital. Agora, para entendermos o porquê das informações serem digitais e entendermos o que isso significa, devemos distinguir os tipos de informações existentes.

Informação analógica

Na natureza, todo tipo de informação pode assumir qualquer valor em um intervalo de - ∞ a + ∞. É possível distinguir uma cor verde que esteja um pouco mais clara de outro tom de verde, em uma variação quase infinita de tons de mesma cor, do mais claro até o mais escuro. Pode-se distinguir um som mais alto do que outro, assim como perceber quanto um ambiente está mais claro do que outro. Todo esse tipo de informação é conhecido como *informação analógica* (TORRES, 2001). Os sinais analógicos citados são lidos de forma direta, sem que seja necessário ocorrer qualquer tipo de decodificação complexa para compreendê-los.

Atenção

Situação: uma música gravada em uma fita cassete.

Após, passado certo tempo, a mesma música fica com um som mais "abafado", com "chiados" e "estalos"; que são os ruídos.

A razão disso se deve ao fato de que a informação da música na fita cassete foi gravada de maneira analógica, na hora de reproduzir a música, o gravador simplesmente achou que os ruídos fizessem parte dela.

Isso porque, como a informação foi gravada analogicamente, poderia assumir qualquer valor, inclusive o valor "ruído".

(TORRES, 2001)

O sinal armazenado se degrada com o tempo, e existe sempre certa perda de qualidade ao se fazer cópias. No caso de uma fita, o aparelho interpreta os ruídos gerados pela degradação como parte da música. Quando em uma transmissão, caso o sinal enviado sofra alguma interferência, torna-se difícil identificar se houve alguma alteração ou não no caminho entre o emissor e o receptor, conforme figura abaixo, onde um tom de verde foi enviado e um tom de verde foi recebido, embora de tons diferentes, sendo difícil identificar a alteração:

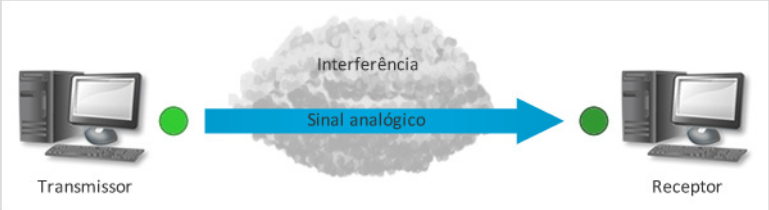


Figura A.1 - Transmissão de sinal analógico sofrendo interferência
Fonte: do autor

Informação digital

Dispositivos eletrônicos no processamento de informações trabalham com o sistema binário. No sistema binário, ao contrário do sistema de numeração decimal (que utilizamos), só há dois algarismos: "0" e "1". Nisto há uma grande vantagem: qualquer valor diferente desses será completamente desprezado pelo circuito eletrônico, gerando confiabilidade e funcionalidade. Esse sistema também é chamado de sistema digital. Cada algarismo binário é chamado de *bit* (contração de *binary digit).* (TORRES, 2001).

Atenção

Situação: uma música gravada em uma fita DAT (Digital Audio Tape, que grava informações digitalmente).

Com o passar do tempo estaria, como uma fita cassete analógica, cheia de interferências em sua camada magnética, na forma de ruído.

Mas por ter sido gravada sob a forma de informações digitais, a música está codificada sob a forma de vários "0"s e "1"s.

Qualquer outro valor diferente de "0" ou "1" será simplesmente ignorado pelo aparelho reprodutor.

(TORRES, 2001)

A vantagem do sistema digital sobre o analógico é que as informações são gravadas em forma de números. No caso de um CD, por exemplo, o que há gravado não são músicas ou sons, mas sim números. Com isso, há como se utilizar mecanismos de correção de erros a fim de verificar a integridade dos dados, tornando-o mais confiável (TORRES, 2001). No caso de uma transmissão, caso o sinal enviado sofra alguma interferência, é possível identificar se houve alguma alteração ou não no caminho entre o emissor e o receptor, visto que os valores de "0"s e "1"s podem ser conferidos, conforme figura abaixo, onde existe uma diferença entre o conjunto de bits enviados e o conjunto de bits recebidos, facilitando a identificação:

Figura A.2 - Transmissão de sinal digital sofrendo interferência
Fonte: do autor

Números binários

Conjuntos de algarismos binários (bits) formam palavras binárias, sendo que cada casa binária só poderá ser preenchida com dois algarismos (0 ou 1), enquanto cada casa decimal pode ser preenchida com dez algarismos (de 0 a 9). As palavras binárias recebem nomes especiais conforme a quantidade de bits utilizada pelas mesmas, representando uma variação de números bastante definida: (TORRES, 2001)

- Nibble: 4 bits (24 = 16 variações)
- Byte: 8 bits (28 = 256 variações)
- Word: 16 bits (216 = 65.536 variações)
- Double Word = 32 bits (232 = 4.294.967.296 variações)
- Quad Word = 64 bits (264 = 18.446.744.073.709.551.616 variações)

O número máximo que pode ser expresso por palavra binária é determinado pela quantidade de bits que ela formada, sendo assim, com um byte é possível representar 256 números (2^8), por exemplo. Os números "inteiros" em binário, pelo fato de ser utilizada a base 2 ao invés da base 10, quando representados em decimal parecem "quebrados". Por exemplo, 1.024 é um número inteiro em binário, pois representa 2^7, sendo que o valor "inteiro" equivalente a ele em decimal seria 1000.

O sufixo K (kilo-), que, em decimal, representa 1.000 vezes (como em Km e Kg), em binário representa 2^{10} vezes (1.024). Logo, 1 Kbyte representa 1.024 bytes, 2 Kbytes representam 2.048 bytes e assim sucessivamente. Do mesmo modo, o sufixo M (mega-) representa 2^{20} vezes (1.048.576) e o sufixo G (giga-) representa 2^{30} vezes (1.073.741.824), diferenciando-se completamente da representação decimal (TORRES, 2001, p. 7). Conforme tabelas a seguir:

Potência de 2		
Kilo (K)	2^{10}	1.024
Mega (M)	2^{20}	1.048.576
Giga (G)	2^{30}	1.073.741.824
Tera (T)	2^{40}	1.099.511.627.776
Peta (P)	2^{50}	1.125.899.906.843.624
Exa (E)	2^{60}	1.152.921.504.607.870.976
Zeta (Z)	2^{70}	1.180.591.620.718.458.879.424
Yotta (Y)	2^{80}	1.208.925.819.615.701.892.530.176

Tabela A.1 – Representação dos sufixos em binário

Potência de 10		
Kilo (K)	10^{3}	1.000
Mega (M)	10^{6}	1.000.000
Giga (G)	10^{9}	1.000.000.000
Tera (T)	10^{12}	1.000.000.000.000
Peta (P)	10^{15}	1.000.000.000.000.000
Exa (E)	10^{18}	1. 000.000.000.000.000.000
Zeta (Z)	10^{21}	1. 000.000.000.000.000.000.000
Yotta (Y)	10^{24}	1. 000.000.000.000.000.000.000.000

Tabela A.2 – Representação dos sufixos em decimal

O byte é a palavra binária mais utilizada, principalmente porque os microprocessadores passaram a ser largamente utilizados (década de 1970) com modelos de oito bits. Um aspecto fundamental é o de representar as palavras binárias byte e bit. Enquanto abreviamos bit com "b" *(b* minúsculo), abreviamos byte com "B" *(b* maiúsculo). Assim, 1 KB é a representação de um kilobyte (1.024 bytes = 8.192 bits), enquanto 1 Kb é a representação de um kilobit (1.024 bits). (TORRES, 2001, p. 9)

Clock

A transmissão de dados no computador, entre um dispositivo receptor e um dispositivo transmissor, é controlada por um sinal de controle chamado clock. Esse sinal é usado para sincronizar o transmissor com o receptor, isto é, para informar ao receptor que um dado está sendo transmitido (TORRES, 2001). Na figura abaixo, é demonstrado um clock com 4 ciclos, sendo que em cada um deles é enviado um dado do processador para a memória RAM:

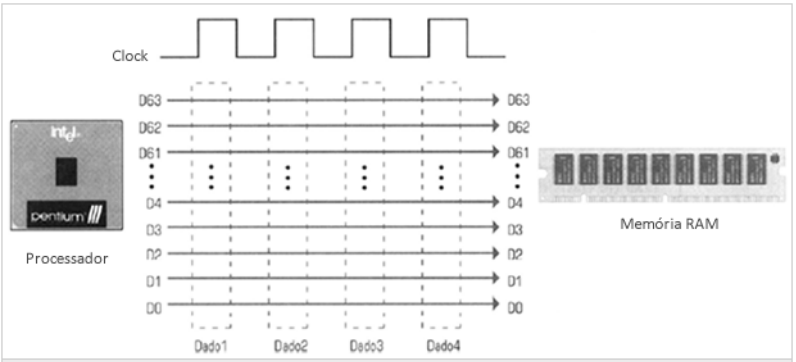


Figura A.3 - Transferência de dados sob a sincronia de um sinal de clock
Fonte: TORRES, 2001, p. 11.

A frequência do clock (quantidade de pulsos por segundo) determina a velocidade da transmissão (frequência de operação). Esta frequência é medida em Hertz (Hz). Por exemplo, um clock de 100 MHz significa que em um segundo temos 100 milhões de pulsos e em cada um desses pulsos existe a possibilidade da transmissão de dados.

Como em cada pulso de clock um dado pode ser transmitido, aumentando-se a frequência do clock, aumenta-se a velocidade com que os dados são transmitidos.

Atenção

Os sistemas de clock utilizados para a comunicação entre os dispositivos são independentes.
A comunicação do disco rígido com a placa-mãe utiliza um sistema de clock, assim como a comunicação da placa de vídeo com a placa-mãe e do processador com a memória RAM.

Transmissão de dados

Cada dispositivo digital trabalha com um determinado número de bits, sendo que o canal de comunicação deste dispositivo transmite e recebe essa quantidade de bits por vez. (TORRES, 2001)

Atenção

Cada dispositivo só pode comunicar-se com dispositivos que manipulem a mesma quantidade de bits. Um dispositivo de 8 bits somente se comunica diretamente com outro dispositivo de 8 bits, assim como um de 32 bits somente se comunica diretamente com outro de 32 bits, e assim sucessivamente.

A comunicação entre dispositivos ocorre de duas formas: transmissão paralela ou transmissão em série.

Transmissão Paralela

Neste tipo de transmissão todos os bits que o dispositivo transmissor é capaz de manipular são transmitidos simultaneamente ao receptor, através de vias (caminhos por onde os dados trafegam) paralelas. Conforme pode ser vista na figura:

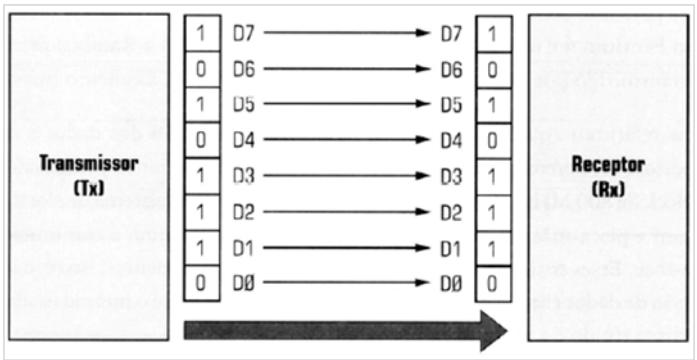


Figura A.4 - Comunicação paralela
Fonte: TORRES, 2001, p. 9.

Interferência eletromagnética

Na transmissão paralela pode ocorrer o problema de ruído interferindo na transmissão dos dados. Isso se deve ao fato de que quando uma corrente elétrica passa por um fio condutor é criado um campo eletromagnético ao redor e, se esse campo eletromagnético for muito forte, será gerado um ruído no fio ao lado, corrompendo a informação transmitida. Quanto maior for a frequência de operação do dispositivo, maior será a possibilidade de ocorrer ruído.

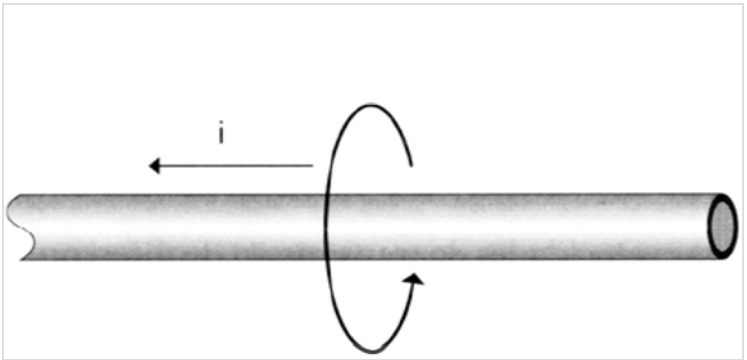


Figura A.5 - Campo eletromagnético gerado ao redor de fio
Fonte: TORRES, 2001, p. 12.

Atenuação

Outra situação que pode ocorrer é de o sinal transmitido enfraquecer na medida em que trafega no fio condutor, sendo que quanto mais longo for o fio, mais fraco fica o sinal ao longo da distância percorrida, tornando-se atenuado.

Transmissão em Série

Este tipo de transmissão é caracterizado por enviar um bit por vez. Como ela possui um único fio condutor utilizado para transmitir as informações, sofre menos com os problemas de ruído e de atenuação. Dessa forma, é possível atingir uma frequência de operação mais alta.

Transmissão em Série Síncrona

Nela, é utilizado um fio independente para a transmissão do sinal de clock, que é usado pelo receptor para saber onde começa e onde termina cada dado que está sendo transmitido e outro para o envio e recebimento dos dados (TORRES, 2001).

Transmissão em Série Assíncrona

Nela, o mesmo canal onde os dados são transmitidos é usado para a transmissão dos sinais de sincronismo entre o transmissor e o receptor. São transmitidos dois sinais de sincronismo, chamados start bit e stop bit, indicando, respectivamente, o início e o fim da transmissão de um grupo de bits (TORRES, 2001).

Taxa de Transferência

Além do clock, a velocidade de transmissão depende também da quantidade de bits que são transmitidos por vez (TORRES, 2001). Considere os 3 dispositivos abaixo:

- Dispositivo de 64 bits com clock de 100 MHz
- Dispositivo de 128 bits com clock de 50 MHz
- Dispositivo de 32 bits com clock de 200 MHz

Calculando a taxa de transferência máxima teórica através da fórmula

Taxa de transferência = clock (em Hz) x quantidade de bits ÷ 8

Podemos verificar que os três dispositivos, apesar de terem quantidade de bits e frequência de clock diferentes apresentam a mesma taxa de transferência máxima teórica de 800 MB/s. Dessa forma, podemos constatar que ambos os fatores (frequência de operação e quantidade de bits) influenciam na transmissão de dados.

Atenção

A velocidade de transmissão paralela é padronizada em bytes por segundo (B/s), assim como a transmissão em série é em bits por segundo (b/s).

Arquitetura física de sistemas computacionais

Antes de conhecermos os componentes que formam a arquitetura dos sistemas computacionais, vamos conhecer a história da informática até chegar ao que conhecemos hoje, para isso, assista ao vídeo disponível em:

http://www.youtube.com/watch?v=F3qWg1JBPZg.

Histórico

Máquinas de calcular e computadores vêm sendo inventados e desenvolvidos ao longo da história da humanidade (WEBER, 2004), conforme pode ser visto no breve histórico a seguir:

Blaise Pascal

Em 1642 ele desenvolve a primeira máquina calculadora mecânica, a pascaline, que era não programável e utilizada para a realização de somas e subtrações;

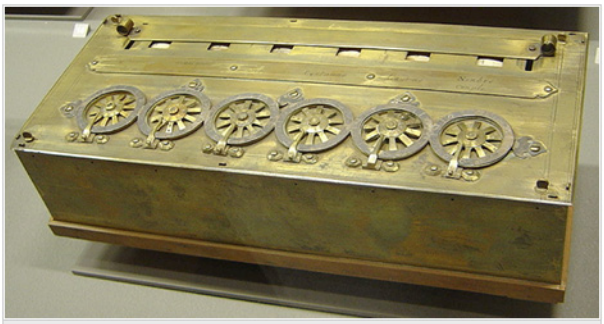


Figura A.6 - Pascaline
Fonte: Wikipedia

Charles Babbage

Ele projeta dois computadores que, embora não tenham sido concluídos, resultam em consideráveis avanços científicos na época. Em 1823 ele projeta o "Dispositivo Diferencial" para a resolução automática de tabelas

matemáticas. Em 1834 ele projeta o "Dispositivo analítico" com intuito de que realizasse qualquer operação matemática automaticamente, sendo que nela já havia módulos de armazenamento (memória) e uma unidade operadora com a entrada e a saída de dados ocorrendo através de cartões perfurados e com a possibilidade de alterar a sequência dos comandos executados (programável). (WEBER, 2004).

Os principais avanços tecnológicos da computação podem ser vistos na imagem abaixo:

Data	Inventor: máquina	Capacidade	Inovações técnicas
1642	Pascal: Calculadora	adição, subtração	transferência automática de vai-um: representação em complemento
1671	Leibnitz: Calculadora	adição, subtração, multiplicação, divisão	mecanismo para multiplicação e divisão
1827	Babbage: Difference Engine	avaliação polinomial por diferenças finitas	operação automática com diversos passos
1834	Babbage: Analytical Engine	computador de propósitos gerais	mecanismo automático de controle de sequência (programa)
1941	Zuse: Z3	computador de propósitos gerais	primeiros computadores de propósitos gerais operacionais
1944	Aiken: Harward Mark I	computador de propósitos gerais	primeiros computadores de propósitos gerais operacionais

Figura A.7 - Avanços tecnológicos. Fonte: Weber, 2004

O primeiro computador eletrônico de propósitos gerais foi provavelmente o ENIAC (Eletronic Numerical Integrator and Calculator), construído entre 1943 e 1946 devido à necessidade de construir tabelas balísticas por interesse do sistema militar americano. Era uma máquina de 30 toneladas, contendo 18000 válvulas (WEBER, 2004).

Com o avanço da pesquisa e o consequente desenvolvimento tecnológico ao longo do tempo, a tecnologia e os estilos usados na construção e programação de computadores formaram várias gerações de computadores (WEBER, 2004), conforme imagem:

Geração	Tecnologias	Característica de hardware	Característica de software	Exemplo
1ª (1946-1954)	válvulas, memórias de tubos catódicos	aritmética de ponto fixo	linguagem de máquina, linguagem assembler	IAS, UNIVAC
2ª (1955-1964)	transistores, núcleos de ferrite, discos magnéticos	ponto flutuante, registrador índice, processadores E/S	linguagens de alto nível, bibliotecas de rotinas, processamento em lote	IBM7094 CDC1604
3ª (1965-1974)	circuitos integrados (SSI e MSI)	microprogramação, pipeline, memória cache	multiprogramação, multiprocessamento, sistema operacional, memória virtual	IBM S/360; DEC PDP-8
4ª(1975-?)	circuitos LSI, memórias semicondutoras			Amdahl 470; Intel 8748

Figura A.8 - Gerações de computadores

Modelo de von Neumann

Em 1946, von Neumann e sua equipe iniciaram o projeto de um novo computador de programa armazenado: o computador IAS. Ela usava uma memória principal de acesso randômico, o que permitia o acesso a uma palavra inteira em uma palavra inteira em uma única operação. Essa máquina acabou por influenciar os projetos subsequentes de outras máquinas (WEBER, 2004).

Atenção

Blocos básicos:

- uma unidade de processamento central, para execução de operações aritméticas e lógicas.
- uma unidade de controle de programa, para determinar o sequenciamento das instruções a serem executadas e gerar os sinais de controle para as outras unidades. Esses sinais determinam as ações a serem executadas.
- uma unidade de memória principal.
- uma unidade de entrada e saída.

(WEBER, 2004)

Ele apresentava sua estrutura da seguinte maneira:

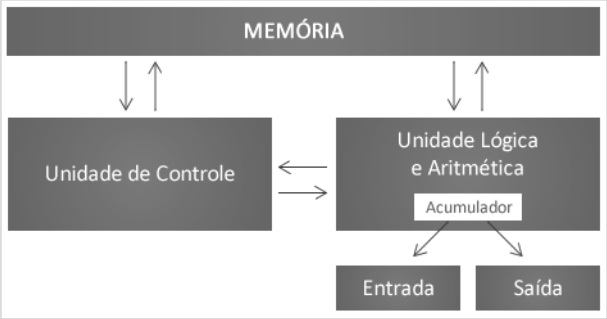


Figura A.9 - Estrutura do IAS
Fonte: do autor

Princípios básicos

Cada computador tem um conjunto de operações e convenções para determinar as posições dos dados com os quais a operação será realizada.

As ações a serem executadas em um computador são definidas por instruções, que são compostas por:

- Operação: especifica a operação que será desempenhada.
- Operandos: indicam a posição dos dados com os quais a operação será realizada.

Um **programa** é formado por uma sequência pré-determinada de instruções. O programa e seus dados ficam armazenados na memória da máquina.

Para que um programa armazenado na memória seja processado, é necessário que suas instruções sejam interpretadas. Isso se deve ao fato de que os programas são formados por instruções de uma linguagem de alto nível, mais convenientes aos programadores (digamos que seja L1), ao contrário da máquina, que trabalha com uma linguagem baseada em instruções de baixo nível (digamos que seja L0). Apesar disso, os programas escritos em L1 têm de executar em um computador programado em L0.

Atenção

Os métodos para executar um programa L1 em L0 são os seguintes:

- Tradução: consiste em substituir cada instrução escrita no programa por uma sequência equivalente de instruções em L0. O computador executa o novo programa L0 em vez do programa L1.
- Interpretação: o programa em L0 considera os programas em L1 como dados de entrada. Ele os executa examinando cada instrução por vez, executando diretamente a sequência equivalente de instruções L0.

(TANENBAUM, 2006)

A maioria dos computadores modernos apresenta dois ou mais níveis no processo de conversão entre a linguagem de programas e a do computador.

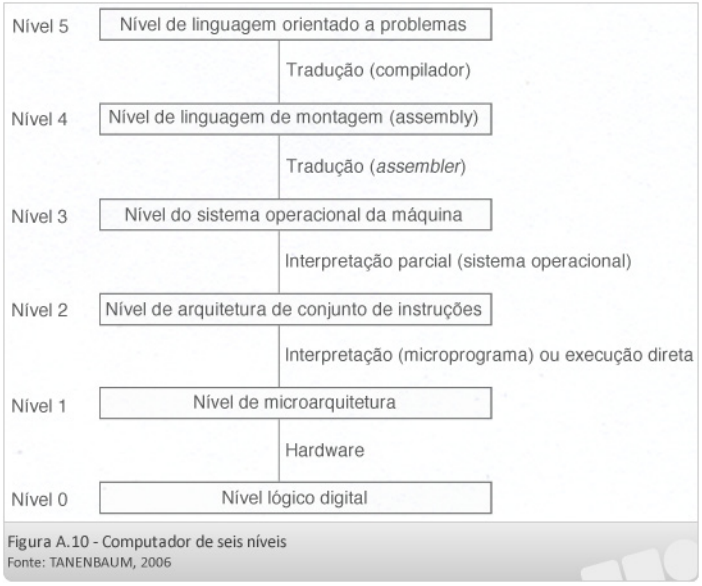


Figura A.10 - Computador de seis níveis
Fonte: TANENBAUM, 2006

Níveis:

- Nível lógico digital: composto por portas que possuem uma ou mais entradas digitais (0 ou 1) e que computam como saída alguma função dessas entradas, como E (AND), OU (OR), etc.
- Nível de microarquitetura: composto por um conjunto de registradores (que formam uma memória local) e um circuito denominado ULA (Unidade Lógica e Aritmética) que executa operações lógicas e aritméticas.
- Nível de arquitetura do conjunto de instruções: conhecido como ISA (Instructiun Set Architectue). São as instruções executadas por interpretação pelo microprograma ou pelos circuitos de execução do hardware.
- Nível de máquina de sistema operacional: caracteriza-se por ser um nível híbrido. Parte das instruções em sua linguagem também está no nível ISA, assim como existe um conjunto de novas instruções, que permitem a ele se relacionar com os níveis superiores.
- Nível da linguagem de montagem: baseado na linguagem assembly, que fornece um método para que sejam escritos programas para os níveis 1, 2 e 3 em uma forma que não seja tão desagradável quanto as linguagens de máquina real em si.
- Nível de linguagem orientado a problemas: consiste em linguagens projetadas para ser usadas por programadores de aplicações que tenham um problema a resolver. Essas linguagens costumam ser denominadas linguagens de alto nível (TANENBAUM, 2006).

Componentes dos sistemas computacionais

Os circuitos eletrônicos digitais, que formam os elementos do computador, são construídos com uma pastilha de material semicondutor, chamado silício. Cada pastilha agrupa milhões de transistores.

Com o avanço tecnológico, tem sido possível a construção de pastilhas de silício cada vez menores e com maior densidade, isto é, maior concentração de transistores. O motivo é diminuição das trilhas que compõem a pastilha de silício. A distância entre elas geralmente é dada em micrômetro (μm) que equivale a 10-6 (0,000001 metros), chegando aos nanômetros (nm) que equivale a 10-9 (0,000000001 metros).

Atenção

Quanto menor a distância das trilhas da pastilha de silício, menos corrente é necessária para deslocar elétrons dentro das trilhas, influindo em:

- Os elétrons chegam ao destino em menos tempo.
- Maior frequência de operação (clock).
- Menor consumo elétrico.
- Menor produção de calor.
- Tensão de alimentação (“voltagem”) menor.

(TORRES, 2001)

Componentes dos sistemas computacionais

Os elementos básicos dos computadores relacionam-se na seguinte estrutura:

Figura A.11 - Relação entre elementos do computador
Fonte: do autor

Dentre os elementos que compõem um sistema computacional atual podemos destacar:

- **Placa-Mãe:** é a responsável pela interconexão de todas as peças que formam o sistema computacional. Ela é desenvolvida de modo a tornar possível conectar todos os dispositivos do computador, oferecendo conexões para o processador, para a memória RAM, para o HD, para os dispositivos de entrada e saída, entre outros. Dois de seus chipsets exercem importantes funções:
  - Ponte norte (northbridge): interliga o processador a dispositivos rápidos como memória RAM e placa de vídeo.
  - Ponte sul (southbridge): interliga os demais dispositivos da máquina à ponte norte, que por sua vez faz a ligação com o processador.
- **Processador**: a CPU (Central Processing Unit) ou UCP (Unidade Central de Processamento) é composta por circuitos integrados passíveis de programação, que manipulam e processam dados. Em seu processamento ele segue instruções (que compõem os programas) que se traduzem em comandos executados por ele com base em seu conjunto de instruções.

Figura A.12 - Processamento de dados
Fonte: do autor

O processador entende uma quantidade finita de instruções que são listadas em uma tabela conhecida como conjunto de instruções. Cada processador pode ter um conjunto de instruções diferentes.

- **Memória**: ela tem a função de armazenar dados e instruções dos programas, sendo estruturada na forma de uma matriz organizada em posições, sendo que cada informação armazenada nela ocupa um endereço específico. Quando se executa um programa, ele é transferido, normalmente, do disco rígido (memória secundária) para a memória RAM (memória principal). O processador busca as instruções dos programas na memória RAM.

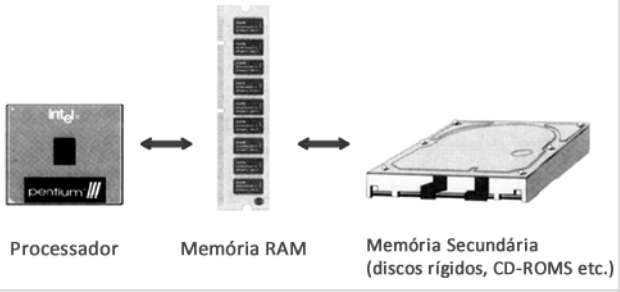


Figura A.13 - Relação entre processador e memórias
Fonte: TORRES, 2001

A transferência de informações entre a CPU e a memória principal ocorre através de **palavras**. A unidade palavra indica a unidade de transferência e processamento de um computador. As palavras são múltiplos de 1 byte, sendo que, se um microprocessador utilizar 32bits serão 4 bytes como tamanho da palavra.

- **Barramento**: é um caminho para a troca de dados entre circuitos, formado por um conjunto de condutores chamados de trilhas, por onde trafegam os bits. Ele apresenta as seguintes características:
  - largura do barramento: número de bits transportados numa operação (ex: 32 bits).
  - frequência de operação: velocidade com que os dados são transmitidos (ex: 400 MHz).

  Geralmente possui duas linhas:
  - Linhas de controle: por onde são transmitidas informações de sinalização, como o tipo de operação que está sendo realizada (leitura, escrita, etc.).
  - Linhas de dados: por onde trafegam instruções, operandos e endereços.

- **Dispositivos de Entrada e Saída**: os dispositivos de I/O (Input/Output) ou dispositivos de E/S (Entrada/Saída) são utilizados para comunicar o computador com o meio externo para receber dados ou responder ao processamento executado (ex: teclado, monitor, impressora, CD, pen drive, etc).

Arquiteturas

Nos primórdios da informática existiam vários fabricantes diferentes e cada um desenvolvia todos os componentes de seus próprios computadores, que eram incompatíveis entre os diferentes fabricantes.
Com o desenvolvimento dos computadores pessoais, surge a plataforma PC, que se trata de uma **arquitetura aberta** a qual permite o uso de componentes de diversos fabricantes e de diferentes sistemas operacionais. Essa arquitetura é baseada em padrões definidos, a partir dos quais produtores podem desenvolver seus próprios componentes, existindo assim compatibilidade entre os componentes de diferentes fabricantes. Dessa forma, ela favorece o desenvolvimento a partir da concorrência entre fabricantes, criando uma demanda maior, permitindo preços mais baixos.
Também existe a **arquitetura fechada**, que se trata de uma arquitetura restrita pelo fato de os padrões, para desenvolvimento de componentes, serem proprietários, não permitindo seu uso por outros fabricantes. Com ela, os conflitos de hardware diminuem, fazendo com que o computador apresente melhor desempenho. Geralmente são encontrados em mainframes, servidores e supercomputadores, sendo que a assistência e as peças para substituição são encontradas somente com o próprio fabricante.

Resumo

Ao final da presente unidade, vimos que:

- O computador trabalha com informações digitais, ao invés de informações analógicas como na natureza.
- O sistema de numeração binário é a base do sistema digital utilizado pelo computador, onde a partir de um bit (0 ou 1) temos um sistema de palavras binárias (nibble, byte, etc) e um sistema de unidades (Kbyte, MByte, etc).
- A transmissão de dados em um computador é sincronizada através do sinal de clock, que indica a frequência de operação na comunicação entre dispositivos.
- Na transmissão de dados, existem dois tipos de dispositivos: em série, que transmitem um bit por vez, e os paralelos, que transmitem um conjunto de bits por vez.
- Tanto a frequência de operação quanto a quantidade de bits influenciam na transmissão de dados.
- O modelo de von Neumann é baseado na existência de uma unidade de processamento central, uma unidade de controle de programa, uma unidade de memória principal e uma unidade de entrada e saída.
- Todo programa que é executado no computador é formado por uma sequência de instruções (formada por operação e operandos) que ficam armazenadas na memória.
- Os programas são escritos em uma linguagem de alto nível, sendo que o computador trabalha em uma linguagem de baixo nível. Para que ocorra comunicação entre elas, é necessário utilizar os métodos de tradução e de interpretação, podendo existir vários níveis nesse processo.
- A distância das trilhas da pastilha de silício influencia em série de aspectos como a distância a ser percorrida pelos elétrons, frequência de operação, consumo elétrico, etc.
- Dentre os componentes de um sistema computacional podemos destacar a placa-mãe, o processador, a memória, o barramento e os dispositivos de entrada e saída.
- Podemos encontrar dois tipos de arquiteturas de computadores atualmente: a aberta e a fechada.

Questões de Revisão

a) Por que os sistemas computacionais não utilizam sinal analógico ao invés de sinal digital?
b) Em que consiste o sistema binário?
c) Conceitue bit, Byte, Word, KByte, MByte e GByte.
d) Diferencie transmissão paralela de transmissão em série.
e) O que é o clock, como ele é medido e o que ele influencia na transmissão de dados?
f) Descreva o princípio básico da arquitetura de Von Neuman e seus elementos básicos.
g) Que problema existe entre programa e computador? Que métodos existem para resolvê-lo?
h) Qual é o papel de uma placa-mãe? E qual é a finalidade da ponte norte e da ponte sul?
i) Qual é a função do processador? Quais são suas características?
j) Como funciona a memória principal?
k) Comente sobre o barramento do computador.
l) Qual é a importância dos dispositivos de entrada e saída?
m) O que a distância entre as trilhas de uma partícula de silício influencia no micro?
n) Quais são as arquiteturas de computadores existentes? Quais são suas características?

Referências

TANENBAUM, Andrew S. **Organização Estruturada de Computadores.** 5ª ed. São Paulo: Pearson, 2006.

TORRES,, Gabriel. **Hardware: curso completo**. 4ª ed. Rio de Janeiro: Axcel Books, 2001.

WEBER, Raul Fernando. **Arquitetura de computadores pessoais**. Porto Alegre: Sagra Luzzatto, 2004.

WEBER, Raul Fernando. **Fundamentos de arquitetura de computadores.** Porto Alegre: Bookman; UFRGS, 2008.

Atividades

1. Considere as seguintes afirmações:

I. Um bit é o conjunto de 8 bytes.

II. Na potência de 2, um Giga equivale a 230 Bytes.

III. Uma informação, no formato digital, pode assumir qualquer valor.

IV. Um sinal digital pode ser verificado quanto a sua integridade.

V. O sistema binário baseia-se em dois estados de tensão em que o computador trabalha: ligado (1) e desligado (0).

Estão corretas as afirmativas

a) II, III e V.
b) I e IV.
c) II, IV e V.
d) III e V.

2. Em relação aos sufixos binários, qual é a alternativa correta?

a) Um GB representa 1.000.000.000 de bits.
b) Um KB representa 1024 bytes.
c) Um TB representa 1.073.741.824 bytes.
d) Um Mb representa 1.000.000 de bits.

3. Sobre o clock é incorreto afirmar que:

a) é utilizado para sincronizar a comunicação entre componentes.
b) os sistemas de clock utilizados para a comunicação entre dispositivos são independentes.
c) a cada ciclo de clock é possível a transmissão de dados.
d) é medido em Hz, que indica a quantidade de bytes transmitidos em um segundo.

4. Considere as seguintes afirmações:

( ) Um dispositivo se comunica apenas com outro que manipule a mesma quantidade de bits.

( ) A atenuação de sinal se caracteriza pela presença de ruídos no meio condutor.

( ) Um dispositivo paralelo transmite um conjunto de bits a cada transferência.

( ) Um dispositivo serial permite maior frequência de operação.

( ) Um dispositivo serial sofre mais problemas de interferência eletromagnética.

A alternativa que contém a ordem correta das afirmações quanto a serem verdadeiras ou falsas é:

a) V – F – V – V - F
b) F – V – F – F - V
c) F – F – V – F - V
d) V – F – F – V - V

5. Sobre a taxa de transferência de dispositivos, é correto afirmar que:

a) a taxa de transferência de dispositivos paralelos é medida em bps, enquanto que a de dispositivos seriais é medida em Bps.

b) a frequência de operação é o fator fundamental que influencia na transmissão de dados.

c) um dispositivo de 64 bits que trabalha em uma frequência de 400 MHz apresenta uma taxa de transferência máxima teórica de 25.600 MB/s.

d) um dispositivo de 32 bits que trabalha em uma frequência de 800 MHz apresenta uma taxa de transferência máxima teórica de 3.200 MB/s.

6. Sobre a arquitetura de Von Neuman, é correto afirmar que:

a) as instruções a serem executadas ficavam armazenadas na memória principal.
b) utilizava um disco rígido para armazenar todas as informações do sistema.
c) a ULA era utilizada para controlar as ações dos outros elementos do sistema.
d) não existiam meios de entrar e sair dados do sistema.

7. Sobre a execução de programas no computador é incorreto afirmar que:

a) as instruções, que formam os programas, são compostas pela operação, que determinam o que será realizado, e por operandos, que indicam os dados a serem utilizados.
b) todo programa a ser executado fica na armazenado na memória.
c) as instruções que formam os programas e as que a máquina utiliza são de linguagens diferentes entre si.
d) o método de tradução consiste em interpretar cada instrução de um programa e gera uma sequência de ações em linguagem de máquina.

8. Considere as seguintes características:

I. Maior frequência de operação.

II. Maior capacidade de armazenamento.

III. Menos corrente é necessária para deslocar elétrons dentro das trilhas.

IV. Diminui a quantidade de componentes.

V. Menor consumo elétrico.

Que características acima são influenciadas pela distância entre as trilhas da pastilha de silício?

a) III e V
b) I, II e III
c) II, IV e V
d) I, III e V

9. Sobre o barramento, é incorreto afirmar que:

a) é um dispositivo utilizado para controlar a troca de dados entre circuitos.
b) é formado por um conjunto de condutores por onde passam os bits.
c) sua largura determina a quantidade de bits que podem ser transportados.
d) a frequência de operação determina a velocidade com que os dados são transmitidos.

10. Sobre a arquitetura de computadores, é incorreto afirmar que:

a) a placa-mãe é responsável por interligar os dispositivos que formam o sistema computacional.
b) um processador executa um conjunto fixo de instruções, iguais para todos os modelos de processadores.
c) a unidade de transferência de dados entre memória principal e processador é conhecida como palavra.
d) todas as informações de um programa a ser executado pelo processador ficam armazenadas na memória RAM, ocupando um endereço específico

TICs

B

Unidade central de processamento

Unidade B

Arquitetura e Organização de Computadores

UNIDADE B

UNIDADE CENTRAL DE PROCESSAMENTO

Introdução:

O processador, também conhecido como CPU (central processing unit, em inglês), ou UCP (unidade central de processamento, em português), é formado por chips responsáveis pela execução de cálculos, decisões lógicas e instruções que resultam em todas as tarefas que um computador pode fazer. Afinal de contas, todos os programas que estão em utilização em um computador, obrigatoriamente, devem ser executados pelo processador.

Atenção

Lembrando o que vimos na unidade anterior, um programa consiste em uma série de instruções que o processador deverá executar para que a tarefa solicitada seja realizada.

Antes de estudarmos como funciona um processador, você tem ideia de como ele opera no computador? Como ele se relaciona com os demais componentes do sistema computacional?

Assista a animação disponível em <http://www.youtube.com/watch?v=oui_qEhe3P4>, que trata o processador como um personagem, demonstrando-o em diversas situações de interação com os demais componentes da máquina com a responsabilidade de fazer com que tudo funcione. Após assisti-lo, comente no fórum suas impressões sobre o papel do processador.

Execução de programas

Para que um programa seja executado, é necessário que sejam transferidos todos os dados necessários, a partir de algum dispositivo de armazenamento para a memória RAM, de onde serão acessados pelo processador. Ao ser processado, ou seja, após o processador executar as instruções que compõem o programa, o resultado é entregue ao programa que será o responsável por determinar o que será feito com ele.

Relação CPU/Memória RAM

O processador trabalha apenas com valores armazenados em registradores na execução de suas atividades, ou seja, ele não acessa diretamente as informações da memória RAM.

Atenção

Os registradores são áreas de armazenamento temporário de valores presentes no processador. Neles são carregados os valores da memória necessários para a execução de uma instrução, sendo que também são utilizados como local de armazenamento para os resultados das execuções das instruções.

Portanto, para que uma instrução seja executada é necessário acessá-la em seu endereço na memória RAM e transferi-la para o registrador do processador.

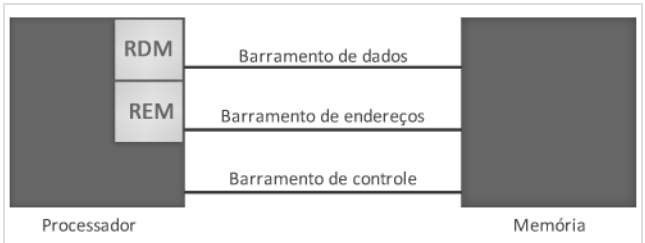


Figura B.1 - Relação CPU/Memória
Fonte: do autor

A transferência de dados entre processador e memória é realizada através do barramento local, utilizando-se dos seguintes componentes:

- REM (Registrador de Endereço de Memória): armazena o endereço da memória onde será lido ou escrito o dado.
- RDM (Registrador de Dados da Memória): armazena o dado a ser escrito ou lido na memória.
- Barramento de dados: liga o RDM à memória, sendo o caminho por onde é feita a transferência de dados.
- Barramento de endereços: liga o REM à memória fornecendo o endereço a ser lido ou escrito.
- Barramento de controle: liga o processador à memória para enviar sinais de controle como leitura (READ), escrita (WRITE) ou espera (WAIT).

As operações, que são realizadas pelo processador na memória, envolvem a leitura de dados da memória para armazená-los nos registradores, já a escrita desses dados estão armazenados em registradores na memória.

A operação de leitura (READ), quando o conteúdo da posição de memória endereçada por REM é copiado em RDM, envolve a seguinte sequência de operações:

REM ← *endereço*
Comando READ
RDM ← Memória[REM]

A operação de escrita (WRITE), quando a posição de memória endereçada por REM recebe o conteúdo de RDM, envolve a seguinte sequência de operações:

REM ← endereço
RDM ← dado
Comando WRITE
Memória[REM] ← RDM

Clock

O processador, no desempenho de suas atividades, necessita realizar operações internamente e também comunicar-se com os demais componentes do micro (para ler e gravar informações na memória principal, por exemplo). Segundo Torres:

> *Os processadores atualmente utilizam um esquema de multiplicação de clock, onde o clock do barramento local é muito inferior ao clock interno do processador. Esse esquema de multiplicação de clock foi criado porque é difícil construir placas-mãe e circuitos de apoio que consigam operar em frequências de operação tão altas como aquelas que os processadores conseguem trabalhar internamente. (2001, p. 34)*

Portanto, o clock interno do processador é a frequência com a qual o processador trabalha internamente na execução de suas atividades, sendo utilizado o clock externo para a comunicação com os demais componentes do computador através da placa-mãe.

Atenção

Imagine um processador que possui uma frequência de operação (clock interno) de 1,8 GHz e que se comunica com uma placa-mãe que possui uma frequência de operação de 200 MHz. Como a frequência de operação entre ambos é completamente diferente, é necessário que o processador reduza sua frequência quando necessitar se comunicar com a placa-mãe, para isso, é utilizado um multiplicador que, neste caso, é 9. Sendo assim, dividindo o clock interno do processador, que é 1,8 GHz, por 9, obteremos o valor de clock de 200 MHz (1,8 GHz / 9 = 200 MHz), que é o necessário para que possa ocorrer comunicação com o restante do micro.

Técnicas para minimizar a diferença de clock

Visando minimizar a diferença existente entre o clock interno e o clock externo do processador são utilizadas algumas técnicas. Uma solução envolve a utilização de memória cache (que será abordada no próximo módulo, juntamente com o sistema de memórias), que se trata de uma memória de maior velocidade de acesso que a memória principal.

A transferência de mais de um dado por ciclo de clock também é uma técnica empregada para compensar essa diferença. Ela consiste em transferir dois dados por ciclo de clock, a chamada DDR (Dual Data Rate), assim como transferir quatro dados por ciclo de clock, a QDR (Quad Data Rate).

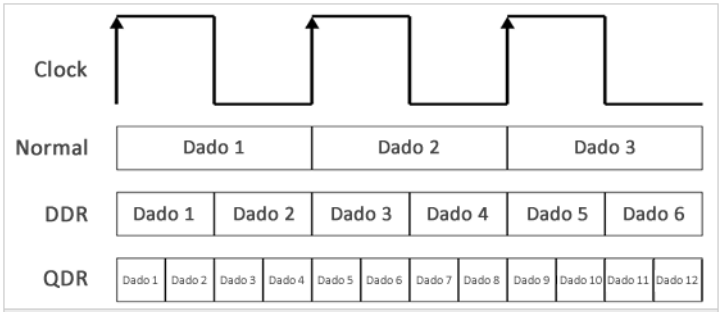


Figura B.2 - Transferência de mais de um dado por clock
Fonte: Clube do Hardware

Arquitetura do Processador

Basicamente, um processador, em sua arquitetura, pode ser dividido em vários blocos:

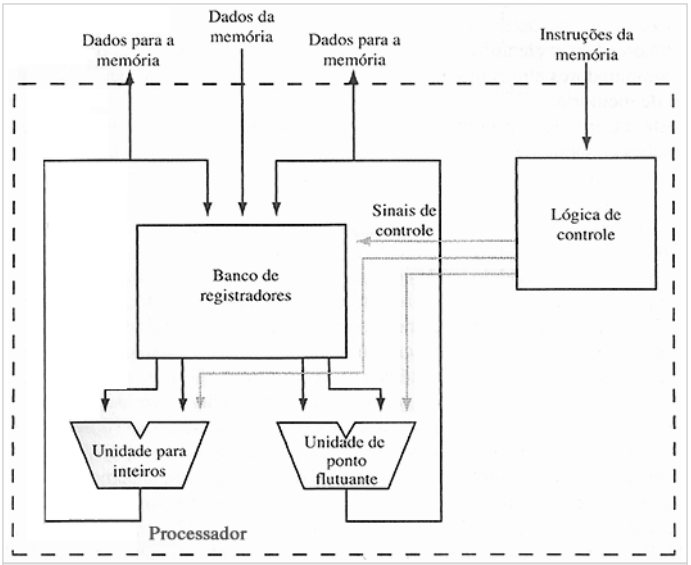


Figura B.3 - Blocos do processador
Fonte: WEBER, 2004

- Unidade operacional (execução)

  Contém o hardware que executa as instruções, inclusive os responsáveis pela busca e decodificação de instruções e a unidade lógica e a aritmética que executam os cálculos.
- Banco de registradores

  Pequena área de armazenamento para os dados que o processador está usando, possibilitando acesso rápido.
- Unidade de controle

  Responsável pelo controle do restante do processador, determinando instruções a serem executadas e quais operações são necessárias para executar cada instrução.

Unidade operacional

Ela é a responsável por executar as transformações sobre dados especificados pelas instruções do computador.

O número, tamanho e uso dos registradores e a quantidade e tipo de operações que a unidade lógica e aritmética realiza são alguns dos fatores que determinam o porte de um processador.

Unidade lógica e aritmética (ULA)

Nela são realizadas as operações aritméticas e operações lógicas sobre um ou mais operandos. Muitas vezes, as operações da ULA são indicadas por instruções simples. As funções mais complexas, exigidas pelas instruções de máquina, são realizadas pela ativação sequencial das várias operações básicas disponíveis.

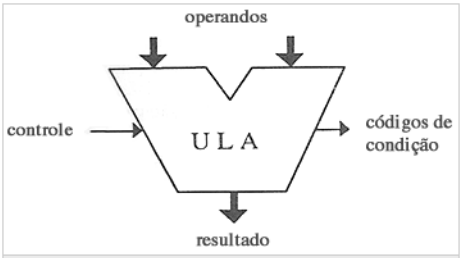


Figura B.4 - Modelo estrutural da ULA
Fonte: WEBER, 2004

A ULA apresenta as seguintes características:

- comprimento em bits dos operandos
- número e tipo de operações
- códigos de condição gerados

Os códigos de condição gerados servem como indicações sobre as operações realizadas, podendo ser, segundo Weber (2004):

- Overflow: resultado de uma operação não pode ser representada no espaço disponível (tamanho da palavra insuficiente).
- Sinal: indica se o resultado da operação é negativo ou positivo.
- Carry: dependendo da operação realizada (soma ou subtração) pode representar o bit de vai-um (carry-out) ou vem-um (borrow-out).
- Zero: indica se o resultado da operação realizada é nulo.

Acumulador (AC)

É um registrador que tem como função armazenar um operando que será utilizado, ou o resultado fornecido pela ULA. Nos computadores mais simples pode existir um acumulador e nos mais complexos podem ser encontrados vários registradores com a função de um acumulador. Dentre sua principal característica está o seu comprimento em bits, que indica a quantidade de bits que ele pode armazenar, sendo que em cada nova operação, um dado é copiado para o seu interior, fazendo com que o conteúdo antigo seja perdido (WEBER, 2004).

Unidade de controle (UC)

Sua principal finalidade é a de fornecer os sinais de controle que gerenciam o fluxo interno de dados no processador. É ele quem coordena o instante preciso em que ocorrem as transferências entre um componente do processador e outro na execução de uma instrução, conforme figura abaixo:

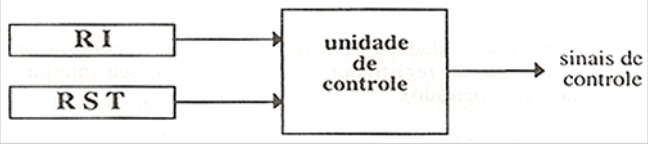


Figura B.5 - Esquema de funcionamento da UC
Fonte: WEBER, 2004

Cada sinal de controle gerado por ela comanda uma micro-operação a ser realizada. Ela recebe como entrada o valor do Registrador de Instrução e decodifica-o, juntamente com os sinais de saída da ULA (RST). Para cada código de instrução ela gera uma sequência de sinais diferentes, ativando os circuitos correspondentes para que cada uma das tarefas necessárias para a busca e execução da instrução seja completada.

As tarefas podem incluir a carga de valores em um registrador, seleção de um dado para entrada em componente, ativação da memória, seleção de uma operação da ULA, habilitação de um circuito lógico, etc. (WEBER, 2004)

Banco de registradores

No computador existem registradores que executam funções específicas e que, dependendo da arquitetura, podem ser encontrados em diferentes blocos.

- Apontador de instruções (Contador de programa – PC)

  Sua função é manter atualizado o endereço de memória da próxima instrução que deve ser executada após a atual.
- Registrador de instruções (RI)

  Armazena o código da instrução que está sendo executada. É em função dele que a unidade de controle determina quais sinais de controle devem ser gerados para executar as operações determinadas pela instrução. O comprimento em bits do RI está ligado ao tamanho e à codificação das instruções do computador.
- Registrador de estado (RST)

  Nele ficam armazenados os códigos de condição gerados pela unidade lógica e aritmética. Em função dele, em conjunto com o RI, que a UC toma decisões sobre a geração ou não de certos sinais de controle.

Demais elementos do processador

Além dos blocos que compõem o processador, também podem ser encontrados os seguintes elementos:

- Clock (relógio)

  É usado para manter o sincronismo do funcionamento entre todos os componentes do processador.
- Decodificador de instruções

  É um dispositivo utilizado para identificar as operações a serem realizadas com base na instrução a ser executada.
- Barramento Interno de Dados

  É o caminho utilizado na transferência de dados entre os registradores e entre registradores e a ULA.
- Barramento Interno de Endereços

  É o caminho utilizado para a transferência de endereços entre os registradores.
- Barramento Interno de Controle

  É o caminho utilizado para transmitir os sinais da unidade de controle que comandam o funcionamento de cada circuito do processador.

O conjunto de todos os elementos que formam o processador será organizado de acordo com a arquitetura utilizada, como, por exemplo, pode ser visto na figura abaixo:

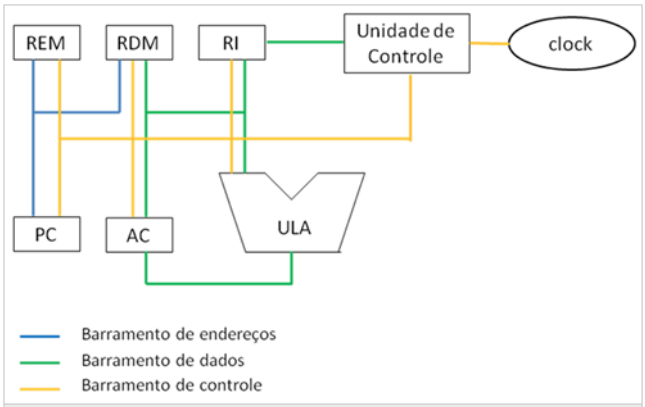


Figura B.6 - Exemplo de arquitetura de um processador
Fonte: do autor

Execução de Instruções

Um conjunto de instruções é um conjunto de bits devidamente codificados que indica ao computador que sequência de micro-operações ele deve realizar. Elas podem ser classificadas, dentre outras formas, como (WEBER, 2004):

- Instruções de transferência de dados
- Instruções aritméticas e lógicas
- Instruções de teste e desvio

O conjunto de instruções envolve o conjunto de todas as instruções que um computador reconhece e pode executar. Todo programa é formado por uma sequência finita de instruções de um determinado conjunto de instruções. A maioria das instruções que formam um programa realiza operações que envolvem operandos (dados), que podem estar em qualquer posição de memória ou em algum registrador.

Para que a UC consiga localizar o operando é necessário que o endereço dele seja indicado junto com a instrução. Nas instruções que envolvem a operação de desvio é necessário indicar para qual posição ou endereço de programa será realizado o desvio.

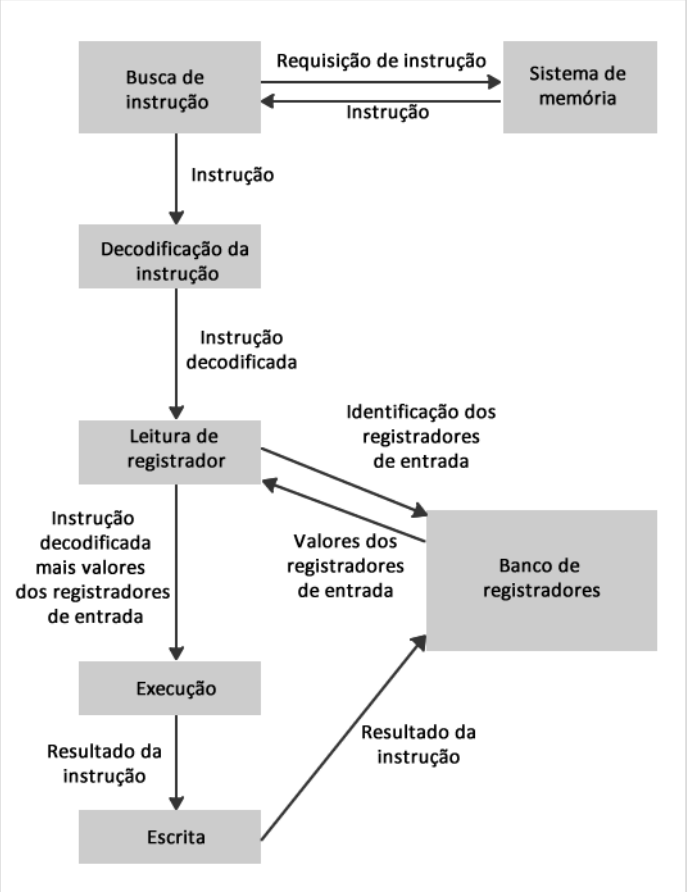


Figura B.7 - Execução de instruções
Fonte: CARTER, Nicholas. Arquitetura de computadores. Porto Alegre: Bookman, 2002.

Sequência de Funcionamento

A sequência para a realização de uma instrução pelo processador é conhecida como ciclo de "busca – decodificação – execução" de instruções.

Uma instrução é buscada por vez para ser executada, de acordo com os seguintes passos:

a) Busca da próxima instrução na memória;
b) Decodificação da instrução;
c) Dados que servirão como operandos são buscados na memória, se for necessário;
d) Cálculo do endereço da próxima instrução;
e) Execução da instrução;
f) Armazenamento do resultado.

No ciclo de instrução, as seguintes ações são executadas:

a) Transferência do conteúdo do endereço de memória apontado pelo contador de programa (PC) como sendo da próxima instrução a ser executada para o RI.
b) Atualizar o valor de PC para indicar o endereço da próxima instrução a ser executada.
c) O decodificador de instruções recebe o código da operação e o decodifica, enviando-o para a UC.
d) A UC gera os sinais necessários para que a instrução possa ser executada.

A sequência do ciclo de instruções é executada para cada nova instrução a ser executada repetidamente e se constitui nas seguintes etapas (WEBER, 2004):

1. Busca
    a) Copiar o PC para o registrador de endereços da memória (REM).
    b) Ler uma instrução da memória.
    c) Copiar o registrador de dados da memória (RDM) para o registrador de instruções (RI).
    d) Atualizar o contador de programa (PC).
2. Decodificação
    a) Determina qual instrução deve ser executada.
3. Execução
    a) Cálculo de endereço dos operandos (se houver).
    b) Busca dos operandos na memória (se houverem).
    c) Seleção da operação a ser realizada pela ULA.
    d) Carga de registradores.
    e) Escrita de operandos na memória.
    f) Atualização do PC (somente no caso das instruções serem desvios).

Pipelines

Nos computadores mais antigos as instruções eram executadas uma por uma, com o processador buscando uma instrução na memória, decodificando-a (determinar qual instrução era), lendo as entradas das instruções (no banco de registradores), executando os cálculos exigidos pela instrução e escrevendo os resultados de volta no banco de registradores. O problema desta abordagem é que o *hardware* necessário para executar cada um desses passos é diferente, de modo que a maior parte dele fica ocioso em um determinado momento, esperando que as outras partes do processador completem a sua parte de execução da instrução. (CARTER, 2002)

Com o objetivo de tornar ágil este processo, evita-se que elementos do processador fiquem parados, é utilizada a técnica de *pipelining* (*pipe* = tubo; *line* = sequência). Ela consiste em reduzir o tempo de execução de um conjunto de instruções, sendo que o tempo para executar uma instrução continua o mesmo, mas a quantidade de instruções executadas por um período de tempo aumenta. (CARTER, 2002)

Atenção

Ele utiliza a metodologia de uma linha de montagem, onde a fabricação de um produto qualquer é subdividida em partes (implementadas em setores) nas quais, tarefas independentes estão sendo desenvolvidas ao mesmo tempo. O tempo para a produção de um produto não diminui, mas sim o tempo entre eles.

É uma técnica de implementação de CPU onde múltiplas instruções podem estar em execução ao mesmo tempo em estágios de processamento diferentes, sendo que cada um deles é responsável pela execução de uma parte da instrução e possui o seu próprio bloco de controle. Assim que um estágio completa sua tarefa com uma instrução, passa ela para o estágio seguinte e começa a tratar da próxima instrução. Sendo que várias instruções são executadas ao mesmo tempo, ocorre um acréscimo no desempenho do processador.

A execução de uma instrução, por exemplo, pode ser dividida em 5 estágios básicos:

1. Busca de instruções (fetch)
2. Decodificação de instruções (decode)
3. Busca dos operandos e armazenamento em registradores (operand fetch)
4. Execução de instruções (execute)
5. Armazenamento do resultado em registradores (store)

Sem a utilização de pipeline a execução de uma instrução ocorreria da seguinte maneira:

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2		processo1			

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3			processo1		

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4				processo1	

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4					
5					processo1

continua...

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4					
5					
6	processo2				

Tabela B.1 - Processo sem utilização de pipelines

Dessa forma, quando um estágio estava ocupado executando sua função, os outros 4 estágios ficavam ociosos, assim como apenas um processo ocupa o processador, levando cinco unidades de tempo para executar e só então outro processo pode começar a ser processado.

Com a utilização de pipelines, o primeiro processo continua demorando cinco unidades de tempo para ser executado, mas logo a seguir, em cada unidade de tempo um novo processo é executado. Conforme segue:

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	

continua...

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	
5	processo5	processo4	processo3	processo2	processo1

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	
5	processo5	processo4	processo3	processo2	processo1
6	processo6	processo5	processo4	processo3	processo2

Tabela B.2 - Processo com utilização de pipelines

Alguns estágios apresentam tempo de execução diferente dos outros, fazendo com que o tempo necessário para encher o pipeline e esvaziá-lo seja variável, com instruções ficando presas no pipeline, aguardando que as instruções que geram suas entradas sejam executadas. Também existe uma série de fatores que limitam a capacidade de um pipeline de executar instruções, como dependências entre instruções (leitura ou escrita de registradores que estão sendo utilizados por outras instruções), desvios (processador não sabe qual instrução deve buscar até que ocorra o desvio) e o tempo necessário para acessar a memória. (CARTER, 2002)

Devido à existência de fatores que podem comprometer o desempenho do uso de pipelines, existem técnicas que são utilizadas para amenizá-los:

- Pré-decodificação: o processador pode realizar a decodificação de instruções (paralelamente) antes do momento de elas serem executadas.
- Execução fora-de-sequência: o processador pode executar previamente um determinado número de instruções. Posteriormente, a ordem de execução é verificada e os resultados das operações são repassados na sua ordem correta.
- Previsão de desvio: caso exista uma instrução de desvio dentro do pipeline, ela pode ser calculada mais cedo (para determinar qual caminho será percorrido e descartar as instruções que não serão necessárias) no pipeline ou prever o destino de cada possibilidade de desvio, para que o processador possa procurar as instruções que serão necessárias anteriormente.

Medidas de Desempenho

Existem diversas formas de medir o desempenho de sistemas computacionais. Dentre elas podemos destacar:

- MIPS (Milhões de Instruções por Segundo): mede a execução de instruções, dividindo o número de instruções executadas de um programa pelo tempo necessário para executá-lo, não considerando que existem diferentes instruções, com tempos de execução diferentes.

- FLOPS (Operações de Ponto Flutuante por Segundo): mede basicamente o desempenho da ULA, analisando apenas quantas instruções complexas são executadas.
- Tempo de acesso: está relacionado à velocidade de cada componente e a do barramento que interliga o processador à memória, tratando o tempo gasto entre o instante em que foi realizada uma solicitação e o instante em que o sistema entregou a resposta.

Programação de processador

A única linguagem que o processador trabalha é a de máquina, baseada em linguagem binária, que é utilizada para a codificação do conjunto de instruções de um computador. Para facilitar a tarefa de programação e de depuração são utilizados mnemônicos, que são associados aos códigos das instruções, nomes aos operandos e rótulos (labels) às posições ocupadas pelo programa, a chamada linguagem simbólica. (WEBER, 2004)

Nesse processo, é utilizado o montador, pois o programa escrito em linguagem simbólica precisa ser traduzido em linguagem de máquina para que possa ser executado, passando por um processo conhecido como montagem.

Resumo

Até este momento vimos que:

- Os programas a serem executados ficam armazenados na memória principal (RAM) de onde serão acessados pelo processador.
- Os dados necessários para a execução de um programa são armazenados, a partir da memória principal, em registradores (memória temporária interna ao processador), assim como o resultado do processamento.
- A transferência de informações envolvendo processador e memória, seja na leitura ou na escrita, ocorre através de elementos como registradores (REM e RDM) e barramentos (de dados, de endereços e de controle).
- O processador trabalha com dois clocks, um para realizar operações internamente e outro para se comunicar com demais componentes do computador (mais lentos), sendo que existem técnicas de compensação como o uso de memória cache e o envio de mais informações por ciclo de clock.
- Internamente, o processador é dividido nos blocos de unidade operacional (composta por ULA e acumulador), de banco de registradores (formada por PC, RI e RST) e da unidade de controle, bem como o clock, o decodificador de instruções e barramentos internos.
- O conjunto de instruções de um processador, que determina o que ele pode fazer (geralmente operações sobre operandos), pode ser dividido em três tipos de instruções: de transferência de dados, aritméticas e lógicas e as de teste e desvio.
- A sequência para a realização de uma instrução pelo processador é conhecida como ciclo de “busca – decodificação – execução” de instruções. Nela os elementos dos blocos básicos são ativados para que ocorra o processamento;
- A técnica de pipeline é praticada no processador visando dividir a execução de uma instrução em estágios de processamento diferentes, sendo que cada um deles é responsável pela execução de uma parte da instrução, ocorrendo ao mesmo tempo e possuem o seu próprio bloco de controle.
- As formas existentes de medir o desempenho de processadores, podem ser destacadas como MIPS, a FLOPS e o tempo de acesso, dentre outras.

Questões de revisão

a) Qual é a relação existente entre o processador e os programas a serem executados? Que elementos são utilizados e como eles se relacionam na transferência de valores (na leitura e na escrita) entre a memória RAM e o processador?

b) Comente a diferença entre clock interno e clock externo. Por que existem dois clocks no processador? Que técnicas são utilizadas para minimizar a diferença entre eles?

c) Em que blocos pode ser dividida a arquitetura de um processador? Comente sobre as características de cada bloco.

d) Conceitue conjunto de instruções, instrução, programa e operandos.

e) A sequência da execução de uma instrução envolve que passos? Como funciona cada um dos passos da sequência do ciclo de instruções?

f) Que medidas de desempenho podem ser utilizadas para um processador?

g) Qual é a finalidade de uso e como funciona o pipeline em um processador?

h) Que técnicas são utilizadas para compensar a diferença entre estágios de um pipeline?

i) Como é realizada a programação de um processador?

Processador Hipotético

Como vimos anteriormente, a execução de um programa envolve a tradução do mesmo na linguagem que a máquina compreende, a chamada linguagem de máquina, que é baseada em um sistema numérico, ao contrário do que estamos acostumados quando utilizamos a máquina ou programamos algo nela.

Quando tratamos de programação do processador não é diferente, conforme veremos a seguir. Na etapa final, quando o programa (escrito em uma linguagem de alto nível pelo programador) chega ao processador, através do processo de tradução e de interpretação (vistos na unidade A), é como se o tivéssemos escrito diretamente em linguagem numérica, que nem é ilustrado na imagem abaixo:

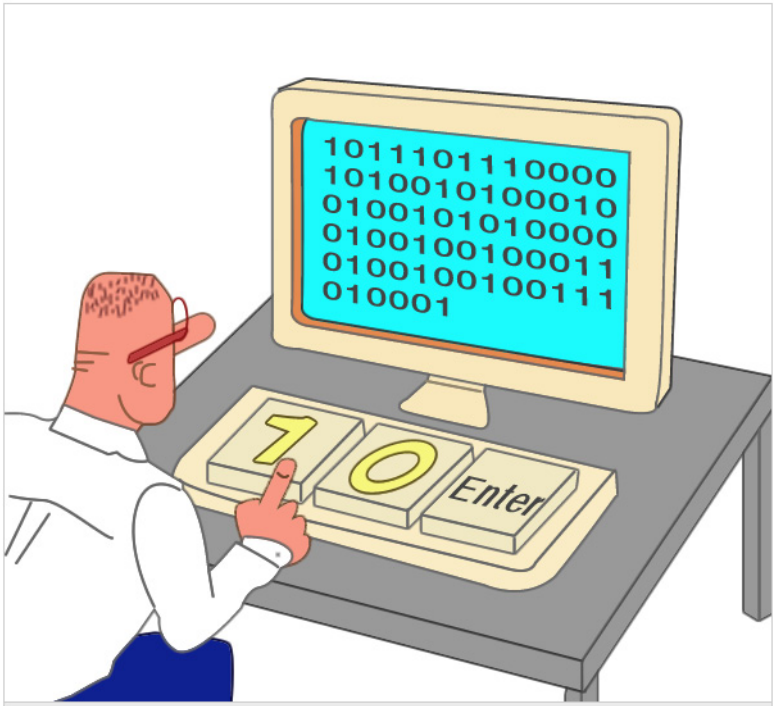


Figura B.8 - Programando em Binário
Fonte: http://www.boadica.com.br/humor/75/esta-e-pra-quem-progamou-em-binario

Como funciona este processo em relação ao processador é o que nós estudaremos a partir de agora.

Para facilitar a compreensão sobre a arquitetura e o funcionamento de processadores são utilizadas versões mais simples, chamadas de processadores hipotéticos. Exemplos de processadores hipotéticos são o Neander, o Ramses, o Cesar, Simuladores MIPS, etc. O PH1 é um desses processadores, com o propósito de apresentar os principais conceitos básicos sobre arquitetura e organização de computadores.

Notação RTL

A notação RTL (Linguagem de Transferência de Registradores) é utilizada como forma de as transferências de dados entre blocos, seja entre registradores e entre registradores e memória.

Notação	Definição	Exemplo
<-	Transferência de conteúdo entre registrador ou entre registrador e memória.	reg1 <- reg2
MEM [end]	Aponta para uma posição na memória.	reg1 <- MEM[50]
$REG_{m..n}$	Seleciona somente o conjunto de bits que iniciam na posição m e terminam na posição n	reg1 <- $reg2_{3..0}$
+ - * /	Operadores aritméticos de soma, subtração, multiplicação e divisão, respectivamente.	reg1 <- reg1 + reg2
& | ^ !	Operadores bit-a-bit AND, OR, XOR e NOT, respectivamente	reg1 <- reg1 & reg2
SE condição ENTÃO transferência	Trasnferência condicional	SE reg1 > 0 ENTÃO reg2 <- reg3
== != <> <= >=	Operadores lógicos da comparação. Respectivamente, igual, diferente, menor, maior, menor que, maior que.	SE reg1 != 0 ENTÃO reg1 ,-0

TABELA B.3 - Lista de notações

Fonte: http://usuarios.upf.br/~brusso/arq1/phsim_doc/prog_sim.htm

Organização

O PH1 possui uma organização interna mais simples em comparação com processadores comerciais. Ele é um processador que trabalha com palavras (dados) e endereços de 8 bits. Ele possui um conjunto de 16 instruções que podem, ou não, ter um operando, que sempre é um endereço de memória. Ele também possui barramentos, registradores, uma unidade lógica aritmética e uma unidade de controle, conforme demonstrados na figura baixo:

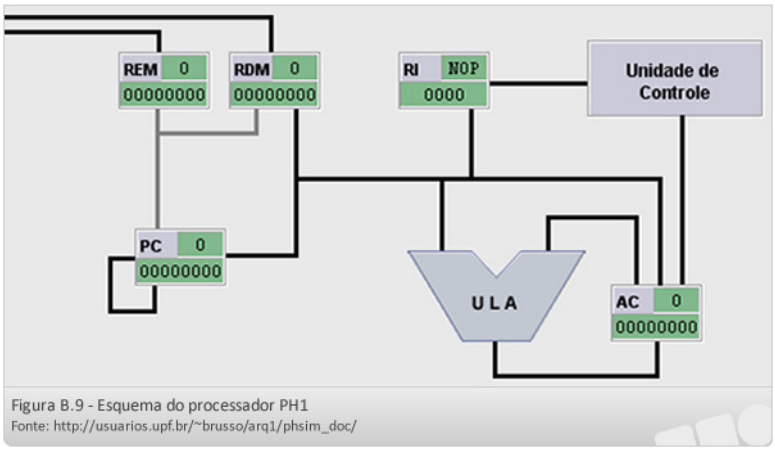


Figura B.9 - Esquema do processador PH1
Fonte: http://usuarios.upf.br/~brusso/arq1/phsim_doc/

Ele possui um conjunto de cinco registradores, com quatro deles com largura de 8 bits, sendo que a exceção é o RI que possui somente 4 bits. Eles são classificados em dois grupos: de uso reservado ao processador e de uso geral (o único dos cinco registradores que é de uso geral, e disponível ao programador, é o AC).

A função atribuída a cada registrador é:

- AC: é o acumulador. Com uma largura de 8 bits, nele ficam armazenados os valores a serem processados, assim como o resultado de uma operação executada, pelo processador.
- PC: é contador de programa. Com uma largura de 8 bits, ele armazena o endereço da próxima instrução a ser executada pelo processador.
- RI: é o registrador de instrução. Com uma largura de 4 bits (devido ao fato de o PH1 possuir apenas 16 instruções:

24 = 16), ele armazena o código binário da instrução que está sendo executada no momento pelo processador.
- RDM: é o registrador de dados da memória. Com uma largura de 8 bits, ele armazena os valores recebidos ou que serão enviados para a memória.
- REM: é o registrador de endereços da memória. Com uma largura de 8 bits, ele armazena o endereço de memória em que será realizada a próxima operação de leitura ou de escrita de dados.

A transferência de informações entre os registradores do PH1 ocorre através dos barramentos internos, que são três:

- Barramento de dados: realiza o transporte de dados da memória à ULA; da ULA para os registradores e entre os registradores. Esse transporte de dados pode ser realizado em ambos os sentidos;.
- Barramento de endereços: transporta os endereços entre os registradores, sendo utilizado pelo PC, o REM e o RDM.
- Barramento de controle: envia sinais a partir da UC a fim de comandar o funcionamento de cada componente do processador.

A Unidade Lógica Aritmética trabalha com o tamanho de palavra de 1 byte, com números inteiros em complemento de dois. Ela recebe geralmente como entrada dois operandos: um deles vem do AC e o outro é um valor vindo da memória que é armazenado temporariamente no RDM. O resultado da operação é armazenado em AC.

A Unidade de Controle é a responsável por receber a instrução armazenada no RI e a decodificar, gerando uma série de sinais que ativam os componentes correspondentes para a realização da execução da instrução.

Para que exista um sincronismo entre a unidade de controle e os demais componentes do processador é utilizado o sinal de clock.

A memória que o PH1 utiliza é composta por 256 endereços, numerados de 0 a 255, e organizada da seguinte forma:

- Endereço 0 até 127: área de armazenamento de instruções.
- Endereço 128 até 255: área de armazenamento de dados.

Instruções

O conjunto de instruções do PH1 é formado por 16 instruções, que estão codificadas em dois formatos:

- Instruções que possuem operando: elas referenciam operandos na memória e endereços de desvio, que necessitam de dois bytes (uma posição de memória para cada byte) para serem codificadas, sendo que o primeiro contém o código da instrução e outro contém o endereço do operando;
- Instruções que não possuem operando: elas não referenciam a memória, utilizando somente um byte (uma posição de memória) para a instrução.

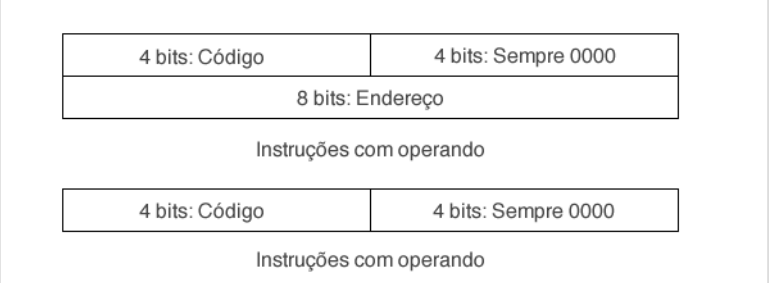


Figura B.10 - Formato das instruções do PH1

O conjunto de instruções é a principal característica da arquitetura de um computador. Ele define quais são as operações suportadas e qual o código binário correspondente pelo processador. No PH1, cada uma das instruções possui o seu código binário (formado por 4 bits), o seu mnemônico, indica uma função a ser desempenhada, bem como gera uma série de relações entre os elementos do processador (descrição em RTL) para que a operação seja efetuada.

Código	Mnemônico	Função	RTL
0000	NOP	Não executa operação alguma.	
0001	LDR end	Carrega um valor em registrador.	AC ← MEM[end]
0010	STR end	Armazena um valor na memória.	MEM[end] ← AC
0011	ADD end	Adição.	AC ← AC + MEM[end]
0100	SUB sub	Subtração.	AC ← AC - MEM[end]
0101	MUL end	Multiplicação.	AC ← AC * MEM[end]
0110	DIV end	Divisão.	AC ← AC / MEM[end]
0111	NOT	Operação lógica de Negação bit-a-bit.	AC ← !AC
1000	AND end	Operação lógica “E” bit-a-bit.	AC ← AC & MEM[end]
1001	OR end	Operação lógica “OU” bit-a-bit.	AC ← AC | MEM[end]
1010	XOR end	Operação lógica “OU” exclusivo bit-a-bit.	AC ← AC ^ MEM[end]
1011	JMP end	Desvio incondicional.	PC ← end
1100	JEQ end	Desvio condicional, caso AC == 0.	se AC==0 entao PC ← end
1101	JG end	Desvio condicional, caso AC > 0.	se AC>0 entao PC ← end
1110	JL end	Desvio condicional, caso AC < 0.	se AC<0 entao PC ← end
1111	HLT	Término de execução.	

Tabela B.4 – Lista de instruções do PH1

Conforme visto no conteúdo anterior, cada programa de computador é formado por um conjunto de instruções executadas uma a uma pelo processador. Para que ocorra o processamento das instruções é necessário que elas e os dados utilizados por elas sejam carregados na memória no formato binário (linguagem de máquina).

Para programar em nível de máquina é necessário, portanto, que se conheça o código binário e o formato de cada instrução e também escolher a posição de memória (os endereços) onde os dados (valores numéricos), utilizados como operandos, ficarão armazenados.

Atenção

As instruções ficam armazenadas sequencialmente a partir do endereço 0, enquanto que os dados estão guardados a partir do endereço 128 da memória;

Estrutura

Existe um caminho de dados por onde ocorrem as transferências entre registradores para a execução de cada instrução no processador. Ele integra os registradores, os barramentos por onde ocorrem as transferências e a ULA, a responsável pelas operações com os dados. Quem os controla é a UC, que comanda a necessidade de um ciclo de busca de operando de uma instrução.

O ciclo de busca da instrução é o mesmo para todas as instruções do PH1. Ele inicia na busca do código da instrução, que deve ser buscado na memória e armazenado no registrador RI. Após, o valor do registrador PC é incrementado para que passe a apontar para o próximo endereço de memória:

```
REM ← PC
RDM ← MEM[REM]
PC ← PC + 1
RI ← RDM7..4
```

Na sequência, o código da instrução, que está armazenado no RI, passa pelo processo de decodificação onde é interpretado. A partir disso, o bloco de controle, através da UC, especifica a instrução a ser executada e busca de operando, se existir, comandando a sequência necessária para a sua execução, através da geração de sinais de controle.

O próximo estágio envolve a busca do operando, caso a instrução que foi decodificada o possua, onde ele deve ser buscado na memória e armazenado no RDM, com o valor de PC sendo incrementado para apontar para o endereço onde está a próxima instrução:

```
REM ← PC
RDM ← MEM[REM]
PC ← PC + 1
```

A próxima etapa envolve a execução das instruções, que consiste em transferências de valores entre registradores, trocas de sinais entre os circuitos e operações realizadas pela ULA com a finalidade de completar seu processamento.

Execução de um programa

Imagine um programa que realiza a soma de dois valores armazenados na memória e armazena o resultado em outra posição da memória (A = B + C). Nesse caso, A ocuparia o endereço 128; B o endereço 129 e C o endereço 130. Cada instrução necessária para realizar a operação é convertida em seu código binário equivalente em linguagem de máquina. A execução de um programa em linguagem de máquina corresponde à busca e à execução de cada uma das suas instruções de forma sequencial (exceto em caso de instruções de desvio) até que o final do programa seja alcançado.

No caso do programa descrito anteriormente, que realiza a soma de dois valores (A = B + C), as instruções do PH1 necessárias para realizar tal operação, utilizando seus mnemônicos para representá-las, ficariam da seguinte forma, no formato de instruções:

```
LDR 129
ADD 130
STR 128
HLT
```

Conforme a tabela de instruções do PH1, podemos ver a instrução LDR é utilizada para carregar um valor da memória, neste caso da posição 129 que corresponde à B, no registrador AC. A instrução seguinte ADD indica que o valor carregado em AC pela instrução anterior deve ser somado com o conteúdo da posição 130 da memória, que corresponde à C. O resultado da soma automaticamente fica armazenado

no registrador AC. Com a instrução STR é ordenado que o conteúdo de AC seja armazenado na memória na posição 128, correspondendo à A. Como marca de fim de programa, é utilizada a instrução HLT, que não indica nenhuma posição de memória, mas apenas o fim da execução.

Quando da execução dessas instruções, elas devem ser convertidas para a linguagem de máquina:

$Endereço_{10}$	$Conteúdo_{2}$
0	00010000
1	10000001
2	00110000
3	10000010
4	00100000
5	10000000
6	11110000
...	
128	00000000
129	00000101
130	00000010

Tabela B-5

Conforme vimos antes, a área referente às instruções inicia na posição 0 da memória e a de dados começa a partir da posição 128. Podemos observar, então, que as instruções do programa ocupam da posição 0 até a posição 6 da memória e os dados ocupam da posição 128 até a 130. É importante lembrar, também, que existem instruções que possuem operandos e outras que não os possuem, no caso das que possuem elas ocuparão duas posições de memória e as que não possuem ocuparão apenas uma. Sendo assim, podemos perceber que as instruções LDR, ADD e STR possuem operandos, pois elas indicam uma posição de memória onde estão os dados necessários para executá-las, já a instrução HLT não possui, pois não indica posição alguma.

Todo programa sempre iniciará com uma instrução. O PH1 possui 16 instruções, portanto, para representá-las em binário, são necessários 4 bits, pois 24 representa 16 variações. Como o tamanho da palavra utilizada no PH1 é de 8 bits, as instruções também ocuparão um espaço de 8 bits (embora necessitem apenas de 4), sendo assim, os 4 últimos bits (os mais à esquerda) representam o código da instrução e os 4 primeiros bits (os mais à direita) são completados com 0000, que devem ser desconsiderados.

Por exemplo, na posição 0 da memória encontra-se o valor binário 00010000. Como é uma instrução, os quatro primeiros bits (0000) são desconsiderados, restando os quatro últimos bits (0001). Para descobrir que instrução equivale a este código, deve-se olhar na tabela de instruções do PH1 na coluna código, dessa forma descobrimos que o código 0001 equivale ao mnemônico LDR.

Se a instrução não possuir operando, na posição seguinte encontraremos uma nova instrução a ser decodificada, caso contrário, se ela possuir operando, o endereço imediatamente abaixo indicará obrigatoriamente a posição de memória em que está o dado a ser utilizado para a operação, considerando

os 8 bits para tanto. A partir da conversão do valor binário armazenado na posição, descobre-se onde está o dado necessário.

Por exemplo, descobrimos que na posição 0 da memória existe o código referente à instrução LDR, que é um instrução que possui um operando, sendo assim, na posição de memória 1 estará a indicação da posição da área de dados (a partir da posição 128 até a 255 da memória) em que se encontra o operando. Para isso, basta converter o binário 10000001 para decimal e descobrir que na posição 129 da memória está o valor necessário. Consequentemente, na posição 2 da memória encontraremos uma instrução, repetindo-se os passos anteriores.

A interpretação do código em linguagem de máquina do exemplo ficará da seguinte maneira:

Endereço$_{10}$	Conteúdo$_2$	
0	0001~~0000~~	LDR
1	10000001	129
2	0011~~0000~~	ADD
3	10000010	130
4	0010~~0000~~	STR
5	10000000	128
6	1111~~0000~~	HLT
...		
128	00000000	
129	00000101	
130	00000010	

Tabela B-6

Podemos descrever as relações estabelecidas para cada instrução executada por um programa através da RTL, que encontramos na tabela de instruções do PH1 na coluna RTL. A execução das instruções gera uma série de ações entre os registradores (conforme vimos no item B.8.4) para, basicamente, conduzir os dados ao AC, que é o registrador de propósito geral do PH1. Vamos analisar então as instruções do programa de exemplo:

- A instrução LDR 129 indica que o valor da posição 129 da memória deve ser carregado no registrador AC;
  AC ← MEM [129]
- A instrução ADD 130 indica que o valor armazenado em AC (que foi carregado da posição 129 da memória na instrução anterior) deve ser somado com o conteúdo da posição 130 da memória e o resultado dessa operação deve ser armazenado em AC, sobrescrevendo o conteúdo anterior;
  AC ← AC + MEM [130]
- A instrução STR 128 indica que o conteúdo de AC deve ser armazenado permanentemente na posição 128 da memória;
  MEM [128] ← AC

Sendo assim, as instruções

LDR 129
ADD 130
STR 128
HLT

representadas em RTL ficam da seguinte forma:

AC ← MEM [129]
AC ← AC + MEM [130]
MEM [128] ← AC

Exemplo

Partindo do programa abaixo, que está representado em linguagem de máquina, tente traduzi-lo para o formato de instruções, utilizando seus mnemônicos e também em RTL. Abaixo, na Tabela B-7 você encontra o exemplo resolvido para comparar com sua resposta

Memória	
End_{10}	$Conteúdo_2$
0	00010000
1	10000000
2	01110000
3	10000000
4	10000001
5	00100000
6	10000001
7	11110000
...	
128	10110101
129	01010111

Tabela B-7

Formato de instruções

Lembrando que as instruções utilizam apenas os 4 últimos bits, devemos desconsiderar os 4 primeiros bits. Outro detalhe importante é que devemos identificar se a instrução possui operando ou não. Se ela possuir, o endereço seguinte indicará a posição da memória onde ele está, caso contrário, a próxima posição da memória será uma nova instrução, conforme segue na Tabela B-8:

Memória	
End_{10}	$Conteúdo_2$
0	0001~~0000~~
1	10000000
2	0111~~0000~~
3	1000~~0000~~
4	10000001
5	0010~~0000~~
6	10000001
7	1111~~0000~~
...	
128	10110101
129	01010111
$Registradores_2$	
AC	00000000
PC	00000000

LDR
129
ADD
130
STR
128
HLT

Tabela B-8

Portanto, após identificar o significado de cada linha, a sequência de instruções correspondente à linguagem binária é:

LDR 128
NOT
AND 129
STR 129
HLT

Observe que a instrução NOT (posição 2 da memória) não possui um operando, sendo assim, na posição seguinte (posição 3 da memória) existe uma nova instrução (AND).

RTL

A sequência equivalente de instruções em RTL:

AC ← MEM [128]
AC ← ! AC
AC ← AC & MEM [129]
MEM [129]

Resumo

Ao final deste conteúdo, vimos que:

- A estrutura de um processador comercial, que encontramos em nossos computadores, é extremamente complexa. Para fim de estudos, existem os processadores hipotéticos, de estrutura mais simples, que nos permitem compreender como um processador funciona internamente.
- Para representar a relação, na transferência de informações, entre os elementos internos do processador existe a notação RTL.
- O processador hipotético que utilizaremos é o PH1, que trabalha com informações de tamanho de palavra de 8 bits e formado pelos elementos RDM, REM, PC, AC, RI, ULA, UC, Clock e barramentos internos de endereços, dados e de controle. Ele trabalha com uma memória formada por 256 endereços com capacidade de 8 bits cada, sendo que da posição 0 até a 127 ficam armazenadas instruções e da posição 128 até a 255 ficam armazenados os dados.
- O PH1 possui 16 instruções (conforme Tabela B.1), sendo que algumas indicam operação com operando (ocupando dois endereços de memória) e outras não (ocupando apenas um endereço de memória). Elas são representadas em código binário utilizando-se 4 bits e também através de um mnemônico específico. Cada uma delas desempenha uma função específica, que desencadeia uma série de relações entre os elementos do processador.
- Para a execução de uma instrução, é necessário lê-la da memória e armazenar seu código em RI para posterior decodificação, que é utilizada pela UC para gerar os sinais de controle para ativar os elementos necessários à realização da operação e, se existir, para buscar operando na memória, encerrando com a execução da instrução pela ULA.
- Na representação do processo de funcionamento do processador, podemos encontrar as seguintes representações: através de instruções, utilizando o mnemônico delas; através da linguagem de máquina, utilizando o código binário correspondente a elas; através da notação RTL, demonstrando a relação existente entre os elementos do processador na execução da instrução.

Atividades

1. Que instruções estão sendo executadas pelo seguinte programa em linguagem de máquina? Escrever o programa de linguagem de máquina para linguagem de instruções e RTL.

a)

Memória	
End10	Conteúdo
0	00010000
1	10000000
2	01010000
3	10000010
4	01110000
5	00100000
6	10000001
7	11110000
...	
128	10001100
129	00000000
130	00011011

b)

Memória	
End_{10}	Conteúdo
0	00010000
1	10000000
2	00110000
3	10000001
4	11000000
5	00010010
6	11010000
7	00001010
8	11100000
9	00001110
10	01000000
11	10000010
12	10110000
13	00000100
14	00110000
15	10000011
16	10110000
17	00000100
18	11110000
...	
128	10001001
129	11011001
130	00000001
131	00000010

2. Reescrever o programa abaixo em RTL para a linguagem de máquina e para o formato de instruções:

AC ← MEM [129]
AC ← AC - MEM [130]
MEM [130] ← AC
AC ← MEM [128]
AC ← AC ^ MEM [131]

3. Escrever os programas abaixo, escritos em instruções, para linguagem de máquina e RTL:

a)

LDR 128
ADD 129
XOR 130
STR 131
HLT

b)

LDR 129
MUL 130
JEQ 10
SUB 128
JMP 4
HLT

Atividade no fórum

Com base em tudo que foi estudado na unidade B, referente à unidade central de processamento, pesquise sobre os assuntos a seguir e poste no fórum suas impressões a respeito deles até o final da semana como parte da avaliação da etapa:

- Como é caracterizada a arquitetura de processadores RISC? E a arquitetura CISC? Como são encontradas estas arquiteturas nos processadores atuais?
- Considerando os componentes de um processador, vistos no material de aula, como eles estão presentes nos processadores reais atuais como os Intel I7 e AMD Phenom II? Como funcionam os seguintes elementos nestes processadores: cache, frequência de operação, controlador de memória, barramentos, e pipelines?

Referências

CARTER, Nicholas. **Arquitetura de computadores**. Porto Alegre: Bookman, 2002 PHSim 2.0. Disponível em <http://usuarios.upf.br/~brusso/arq1/phsim_doc>. Acesso em: 30 abr. 2011.

TORRES, Gabriel. **Como os Processadores Funcionam**. Disponível em: <http://www.clubedohardware.com.br/printpage/Como-os-Processadores-Funcionam/1145>. Acesso em: 30 abr. 2011.

TORRES, Gabriel. **Hardware: curso completo**. 4ª ed. Rio de Janeiro: Axcel Books, 2001.

WEBER, Raul Fernando. **Arquitetura de computadores pessoais**. Porto Alegre: Sagra Luzzatto, 2004.

Atividades

1. Sobre o processador é correto afirmar que

a) consegue executar instruções a partir de qualquer dispositivo do computador.
b) possui um conjunto de instruções que podem ser atualizadas periodicamente e utilizadas pelos programas.
c) existe um conjunto de instruções padrão para todos os modelos de processadores.
d) tem a finalidade de executar o processamento de dados conforme programação pré-definida dos programas.

2. Em um conjunto de instruções de um processador não encontraremos instruções

a) de transferência de dados.
b) de operação de periféricos.
c) aritméticas e lógicas.
d) de teste e desvio.

3. Sobre clock do processador é correto afirmar que

a) é utilizado o clock do processador para realizar os processamentos e a comunicação com os outros componentes do computador.
b) o clock interno é utilizado na comunicação com a memória RAM.
c) o clock externo sincroniza a comunicação entre os elementos internos do processador.
d) o uso de memória cache é uma técnica que visa amenizar a diferença entre clocks do processador.

4. Em relação aos blocos do processador, é incorreto afirmar que

a) a unidade operacional é composta pela unidade lógica e aritmética e o acumulador.
b) no banco de registradores é onde ocorre o processamento das instruções.
c) a unidade de controle baseia suas decisões no conteúdo do RI e do RST.
d) o clock do processador sincroniza o funcionamento dos demais componentes dos blocos.

5. Sobre as medidas de desempenho de processador é correto afirmar que

a) o MIPS é utilizado para medir o desempenho global do sistema.
b) o FLOPS analisa a execução apenas das instruções mais complexas.
c) o tempo de acesso mede a execução de diferentes tipos de instruções.
d) n.d.a.

6. Sobre o pipeline é correto afirmar que

a) após a execução completa de uma instrução é que outra pode começar a ser executada.
b) existem diversos estágios distintos, cada um responsável pela execução de uma parte da instrução.
c) o tempo para a execução de uma instrução diminui.
d) a execução fora-de-sequência é uma das etapas de um pipeline.

7. Tendo os elementos internos de um processador à esquerda e suas características, fora de ordem, à direita,

(1) REM	( ) armazenar o endereço a ser acessado na memória.
(2) UC	( ) indicar a posição da próxima informação a ser acessada.
(3) RDM	( ) realizar as operações lógicas e aritméticas.
(4) PC	( ) comanda as ações a serem executadas pelos demais elementos.
(5) ULA	( ) armazena valores recebidos da, ou a serem enviados para a, memória.

A alternativa que apresenta a ordem correta contendo a associação dos elementos com suas características é:

a) 1 - 2 - 3 - 4 - 5
b) 4 - 1 - 5 - 3 - 2
c) 1 - 4 - 5 - 2 - 3

d) 3 - 5 - 2 - 1 - 4

8. Em relação aos elementos internos do processador é correto afirmar que

a) o barramento de controle é o caminho por onde são enviados os endereços a serem acessados.
b) o barramento de dados é o caminho utilizado para a transmissão de valores entre os registradores.
c) o decodificador de instruções armazena o código da instrução a ser executada.
d) o AC indica o que cada elemento deve fazer para que a instrução possa ser executada.

9. Na sequência de funcionamento do processador, é incorreto afirmar que

a) antes de buscar a instrução ela é decodificada.
b) ela é composta pelo ciclo de busca – decodificação – execução de uma instrução.
c) na etapa de execução devem ser buscados os operandos, caso existam.
d) a UC recebe a instrução decodificada e gera os sinais para que ela seja executada

10. Referente à programação de processador, é incorreto afirmar que

a) os programas são escritos em linguagem simbólica.
b) o processador trabalha baseado em linguagem binária.
c) o montador traduz a linguagem do programa para uma que a máquina entenda.
d) na linguagem simbólica valores binários são associados aos códigos das instruções.

TICs

C

Sistema de Memória

Unidade C
Arquitetura e Organização de Computadores

UNIDADE C

SISTEMA DE MEMÓRIA

Introdução:

Como vimos na unidade A, o computador possui memórias para armazenar informações necessárias para a execução de dados. Acontece que não existe apenas um tipo de memória, mas várias dependendo da necessidade. Podemos fazer uma analogia com o nosso próprio sistema de memória, conforme pode ser visto a partir do texto abaixo:

Atenção

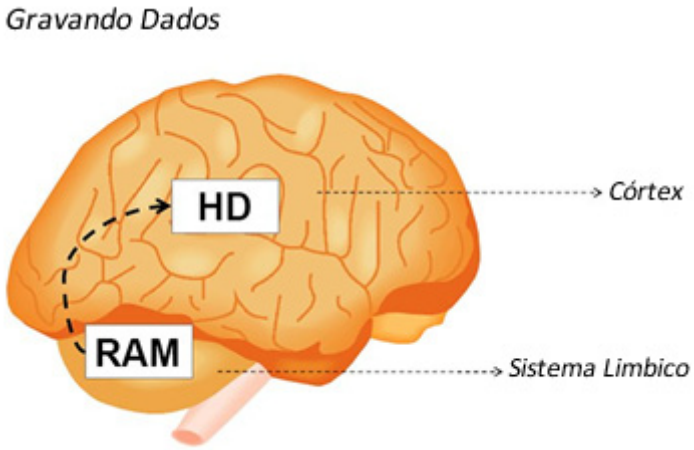
O conteúdo desse post é baseado nas ideias do livro "Aprendendo Inteligência" do Prof. Pier. Ele mesmo classifica o seu livro como um "manual de instruções para o cérebro de alunos em geral".

Cá entre nós, na era da informação quem não quer ter a possibilidade de ler, aprender e fixar o conhecimento. Não precisa ser aluno, basta ter vontade de saber um determinado assunto ou atividade.

Defende Professor Pier que nosso cérebro tem um mecanismo de funcionamento muito parecido com o funcionamento de um computador.

Sabemos que numa máquina existem dois tipos de memoria:

A chamada memoria RAM e a chamada memoria do DISCO.



Pois bem a memoria RAM é uma memoria rápida e volátil e sendo assim se perde todas vez que o computador é desligado, a menos que os dados antes tenham sido salvos.

Quantas vezes não nos ocorreu de perder um texto ou alguma informação do office que ainda não tinha sido salva?

O mesmo ocorre com a nossa memoria de curto prazo ela é processada no sistema límbico.

Assim todas as informações novas .... leituras , aulas, conversas, momentos.... tudo, absolutamente tudo é processado no sistema límbico. É a memoria de curta duração

E como se faz para guardar a informação processada no sistema límbico? Afinal boa parte do que vemos

e temos contato é esquecida.. A resposta é simples. A Emoção

Emoção. Por meio da emoção a memoria de curta duração se torna parte de nossa memória permanente. É a emoção, sentimento... até mesmo a tristeza, a tragédia, a dor ,o amor, o prazer....tudo que é sentimento.

Assim, se você possui prazer em alguma atividade ou leitura... se algo despertou seu interesse é porque lhe dá prazer e assim esse material vai fazer parte da sua memoria permanente. Vai fazer parte de você.

Fonte: <http://oracleandcoffee.blogspot.com/2009/04/coffee-moment-aprendendo-inteligencia.html>

Hierarquia de memórias

Para que o computador apresente um melhor desempenho, quando da manipulação de informações na memória, surge a necessidade de existirem diferentes tipos de memória. O uso desses vários tipos de memória que encontramos no computador depende das necessidades das atividades a serem executadas, o que basicamente podemos dividir em duas situações:

- velocidade na transferência de informações;
- capacidade de armazenamento de informações.

Os diferentes tipos de memória encontradas no computador são interligados, formando um sistema entre si, conhecido como subsistema de memória, sendo que seus componentes são organizados hierarquicamente, conforme podemos observar na figura C.1:

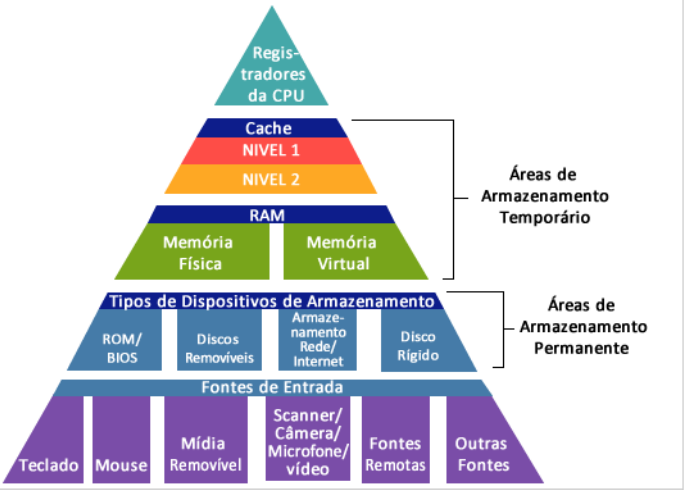


Figura C.1 - Estrutura da hierarquia de memórias de um computador
Fonte: http://informatica.hsw.uol.com.br/memoria-do-computador1.htm

Disposição das memórias

Nesta estrutura, quanto mais alto na hierarquia,

- mais próximo será do processador;
- menor será a quantidade de memória;
- mais rápida será a implementação em tecnologia, o que resulta em menor tempo de acesso a uma informação contida nela.

Por outro lado, quanto mais baixo na hierarquia,

- maior será a capacidade de armazenamento de informações;
- mais lento será o tempo de acesso do que o nível acima.

Dessa forma, como podemos observar na imagem abaixo, quanto mais na parte superior da pirâmide, maior será a frequência de operação, menor será a capacidade de armazenamento e, consequentemente, maior será o custo para desenvolver essa memória, onde encontramos memórias como registradores e cache. Conforme se desce na hierarquia em direção ao nível inferior, a frequência de operação diminui, o custo para a produção é reduzido e a capacidade de armazenamento de informações aumenta, como podemos encontrar no disco rígido.

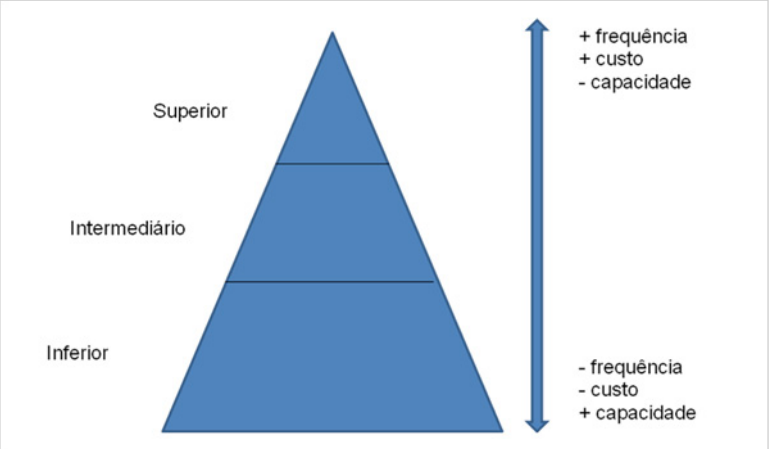


Figura C.2 - Posição das memórias na hierarquia

Através disto, percebemos o porquê das características das memórias que conhecemos. A memória cache, com menor capacidade de armazenamento, comparativamente com as outras, mas com frequência de operação mais próxima a do processador, já o disco rígido apresenta muito mais capacidade de armazenamento, em relação às outras memórias, mas possui um tempo de acesso muito inferior.

O objetivo dessa disposição das memórias é o de manter as informações mais referenciadas nos níveis mais altos da hierarquia, fazendo com que as solicitações de acesso à memória sejam tratadas nesses níveis, visto que, assim, o processador as recebe de forma mais rápida e agiliza o processo como um todo.

Não há como se prever qual posição de memória será acessada com maior frequência, portanto, é utilizado um sistema que se baseia na demanda para determinar quais dados serão mantidos nos níveis mais altos da hierarquia. Assim, quando uma solicitação de acesso é recebida, o nível mais alto é verificado, caso não seja encontrada a informação (falha), o próximo nível mais baixo será acessado e assim sucessivamente até ela ser localizada. Quando encontrada a informação (acerto), um bloco de posições seqüenciais é copiado do nível que a contém para todos os níveis acima dele.

Características das memórias

Quando se trata de memórias, cada uma delas possui características distintas que podem ser enquadradas em seis categorias:

- Tempo de acesso: indica quanto tempo a memória leva para entregar uma informação no barramento de dados após uma de suas posições ter sido endereçada.
- Capacidade: quantidade de informação que pode ser armazenada em uma memória.
- Volatilidade: divide-se em dois grupos:
    - Memória não volátil: que mantêm a informação armazenada quando a energia elétrica é interrompida;
    - Memória volátil: que perde a informação armazenada quando a energia elétrica é interrompida.

- Tecnologia de fabricação: é dividida em dois grupos:
  - Memórias de semicondutores: são memórias eletrônicas, rápidas e mais caras de serem produzidas;
  - Memórias de meio magnético: armazenam informações sobre a forma de campos magnéticos, sendo mais lentas e mais baratas para serem produzidas.
- Temporariedade: é o tempo que a informação permanece nos tipos de memória, indicando se ela é permanente ou transitória.
- Custo: ele varia de acordo com os fatores citados acima como a tecnologia de fabricação, que resulta em maior ou menor tempo de acesso, ciclos de memória, quantidade de bits por espaço físico, dentre outros.

Memória ROM

A memória ROM (Read-Only Memory) é uma memória conhecida como de somente leitura. Em sua definição clássica, suas informações são gravadas uma única vez e, após, não podem ser alteradas ou apagadas, somente acessadas e seu conteúdo não é apagado quando a alimentação elétrica é cortada.

Atualmente, o termo é usado para indicar um conjunto de tipos de memória que são usadas para a leitura de dados em dispositivos eletrônicos digitais, mas que podem ser escritas por meio de mecanismos especiais.

Podemos destacar alguns tipos existentes de memória ROM (Disponível em http://informatica.hsw.uol.com.br/memoria-rom.htm. Acesso em 01 jul 2010).

- ROM (Read-Only Memory)
  - Pode ser escrita apenas uma vez, em seu processo de fabriação.
  - É composta por uma matriz de linhas e colunas, com os endereços sendo formados na intersecção de cada linha e coluna. Em cada endereço é usado um diodo para conectá-lo, permitindo a passagem de corrente e, com isso, fazê-la valer 1. Se o valor a ser registrado for 0, a posição não é conectada com o diodo, impedindo a passagem de corrente.

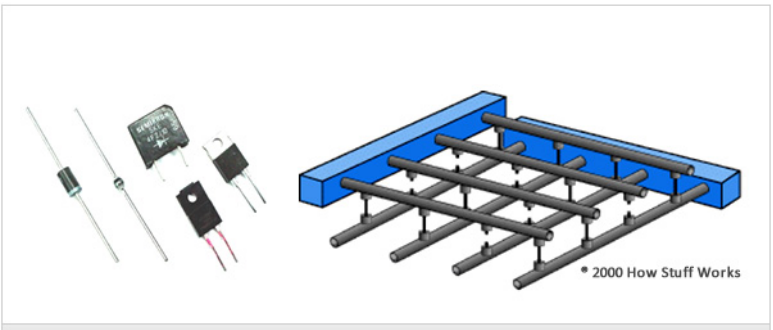


Figura C.3 - Exemplos de diodo à esquerda e à direita uma imagem da composição de uma ROM
Fonte: http://informatica.hsw.uol.com.br/memoria-rom.htm

  - Se um diodo está presente na célula, a carga será conduzida até a terra e a célula será lida como sendo "ligada" (valor 1).
  - Se o valor da célula é 0, não há diodo na intersecção para ligar a linha à coluna, então a carga na coluna não é transferida para a linha.
  - Não se pode programar ou regravar um circuito impresso ROM.

- PROM (Programmable Read-Only Memory)
  - Pode ser escrita com dispositivos especiais apenas uma vez, não podendo mais ser apagada ou modificada após isso.
  - Seus endereços são formados na intersecção de linhas e colunas no formato de uma matriz, onde em cada intersecção existe um fusível, ligando-as.

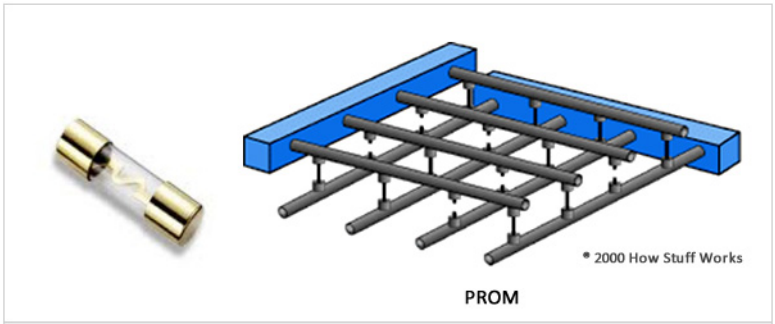


Figura C.4 - Exemplo de um fusível à esquerda e à direita uma imagem da composição de uma PROM
Fonte: http://informatica.hsw.uol.com.br/memoria-rom.htm

- Todas as células possuem um fusível, cujo estado inicial (vazio) de um chip de PROM, é quando todas as posições valerem 1.
- Para alterar o valor das células para 0, é usado um programador para enviar uma tensão mais alta na conexão entre a linha e a coluna, queimando o fusível (processo conhecido como "queimar" uma PROM).

- EPROM (Erasable Programmable Read-Only Memory)
  - Pode ser apagada pelo uso de uma frequência específica de radiação ultravioleta, permitindo sua reutilização.
  - Cada célula de cada interseção possui dois transistores (porta flutuante e porta de controle), que são separados um do outro por uma fina camada de óxido.
  - Inicialmente, todas as portas de uma EPROM vêm com o valor 1. A única ligação da porta flutuante com a linha (wordline) acontece através da porta de controle.
  - Para mudar o valor para 0 é necessário alterar a disposição dos elétrons na porta flutuante. A tensão vem da coluna (bitline), entra pela porta flutuante e é canalizada para a terra.
  - Após a alteração, esses elétrons carregados negativamente atuam como uma barreira entre a porta de controle e a porta flutuante.
  - Para regravar uma EPROM, é necessário apagá-la, sendo necessário um nível de energia suficientemente forte para romper completamente o bloqueio de elétrons negativos na porta flutuante. Esse processo não é seletivo, ou seja, todo o conteúdo da memória é apagado nesse processo.

- EEPROM (Electrically Erasable Programmable Read-Only Memory)
  - Seu princípio de funcionamento é semelhante ao da EEPROM, com a diferença de que pode ter seu conteúdo modificado eletricamente, mesmo quando já estiver sendo utilizada em um circuito eletrônico.
  - No processo de regravação, o chip não precisa ser removido, não tem de ser completamente apagado e não requer equipamento adicional.
  - Para que os elétrons da célula possam retornar ao seu estado normal, é necessária a aplicação localizada de um campo elétrico em cada célula.
  - As mudanças são feitas em um byte de cada vez, o que as torna versáteis, mas lentas.

Utilidade da memória ROM

As ROMs costumam ser utilizadas para armazenar o software básico de funcionamento e equipamentos, o chamado firmware. O firmware de um aparelho é uma espécie de sistema operacional, que realiza a comunicação entre o usuário e o aparelho, bem como armazena funções para execução de tarefas solicitadas, sendo que devido aos tipos de ROM existente atualmente, o firmware pode ser atualizado quando necessário.

No computador, a memória ROM é utilizada para armazenar os programas básicos do sistema como o BIOS (Basic Input Output System) que é o sistema básico de entrada e saída do sistema, o POST (Power On Self Test) que é o autoteste que o computador executa sempre que é ligado e o SETUP que armazena as configurações da máquina.

Resumo

Ao final da presente unidade, nós vimos que:

- Diferentes são os tipos de memória no computador, cada uma com uma finalidade específica, organizadas em uma estrutura que pode ser chamada de hierarquia de memórias.
- A necessidade do uso de memórias depende basicamente de dois fatores: velocidade de transmissão ou capacidade de armazenamento.
- Na hierarquia, as memórias nos níveis mais superiores ficam mais próximas ao processador, tendo frequência de operação mais elevada, assim como seu custo e menor capacidade de armazenamento. Quanto mais próximas dos níveis inferiores, aumenta a capacidade de armazenamento, reduzem-se os custos e diminui a frequência de operação.
- Sempre se tenta manter as informações mais referenciadas nos níveis mais altos da hierarquia, buscando assim evitar que se perca tempo acessando uma informação em memórias mais lentas.
- A tentativa de acesso a uma informação sempre inicia pelo nível mais alto e vai descendo nos níveis até ela ser encontrada, sendo realizada uma cópia do conteúdo em cada um dos níveis superiores.
- As memórias apresentam características específicas como tempo de acesso, capacidade, volatilidade, tecnologia de fabricação, temporariedade e custo.
- Um dos tipos de memória existente é a ROM (Read Only Memory), utilizada basicamente como uma memória para leitura de informações.
- A memória ROM se divide nos seguintes tipos, cada um com suas particularidades: ROM (gravada de fábrica), PROM (permite a programação apenas uma vez), EPROM (permite que o conteúdo seja reescrito apagando todas as informações de cada vez) e EEPROM (permite que o conteúdo seja reescrito e que seja selecionado o conteúdo a ser apagado).

Questões de revisão

a) Por que existem diferentes tipos de memória em um computador?
b) O que é a hierarquia de memórias? O que a posição de uma memória na hierarquia indica?
c) Como funciona o acesso a uma informação na hierarquia de memórias?
d) Quais são as características das memórias e o que cada uma determina?
e) Diferencie os tipos existentes de memória ROM existentes em relação a sua característica de funcionamento e a seus componentes.

Memória RAM

Conforme visto na semana anterior, o computador é composto por um sistema de memória, formado por diferentes tipos de memória, exercendo papéis específicos no funcionamento do computador. Uma delas é a memória RAM, também conhecida como principal que, como o nome diz, executa uma importante função de armazenar dados a serem executados pelo processador.

É uma memória de acesso aleatório (Random Access Memory) que permite o acesso direto a qualquer um de seus endereços, sem a necessidade de percorrer outros endereços para chegar até ele, possibilitando maior agilidade. Seu chip é formado por um grande número de células idênticas, organizadas na forma de linhas e colunas.

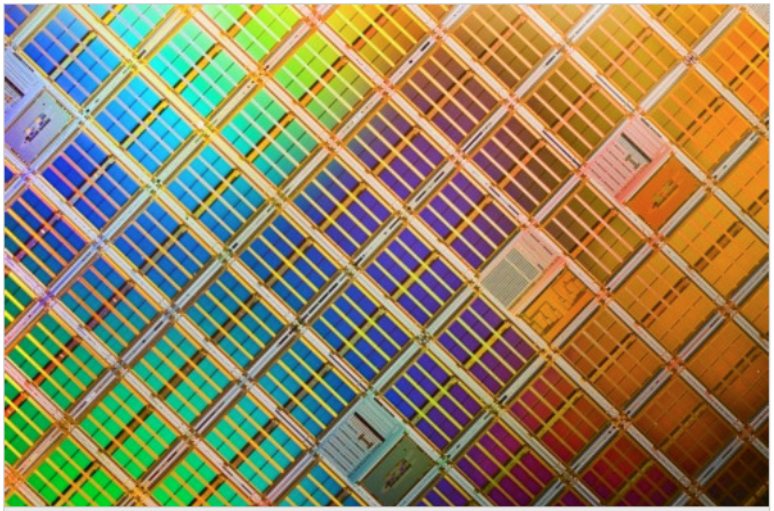


Figura C.5 - Componentes de um módulo de RAM
Fonte: http://www.hardware.com.br/tutoriais/memoria-ram/

Quando um dado precisa ser carregado, primeiramente ele é lido do disco rígido, ou de alguma outra mídia de armazenamento permanente, e transferido para a memória RAM, para então poder ser executado futuramente pelo processador.

A memória RAM oferece tempos de acesso mais baixos que o disco rígido, assim como taxas de transferência mais altas. Porém, ela é uma memória de armazenamento volátil, possuindo a desvantagem de perder os dados armazenados quando a alimentação elétrica é suspensa, existindo a necessidade de salvar os arquivos periodicamente em uma mídia de armazenamento permanente.

Atenção

O chip de memória serve para armazenar dados, não realizando nenhum tipo de processamento. Serve apenas como armazenamento de dados, para tanto, existe o controlador de memória.

Componentes do acesso à memória

No acesso a informações armazenadas na memória RAM, alguns elementos desempenham papéis específicos:

- CPU: centraliza todos os processos que ocorrem na máquina.
- Controlador de memória: é responsável pela comunicação entre a CPU e memória RAM.
- Barramento: são vias de comunicação entre componentes distintos. O Barramento de memória conecta o controlador de memória com a memória RAM.

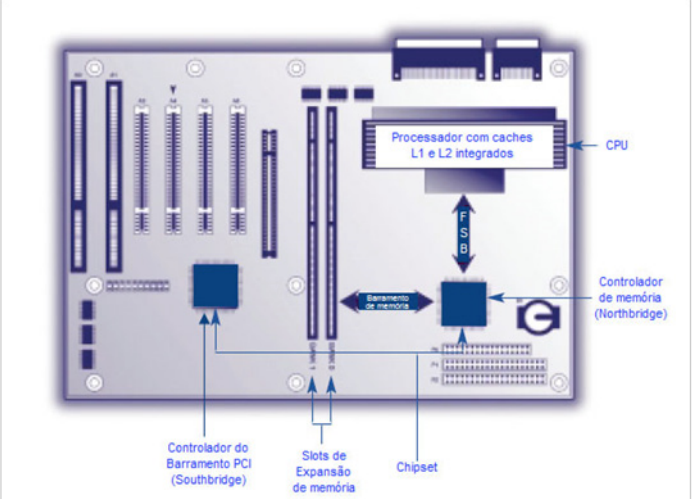


Figura C.6: Componentes relacionados com a memória RAM
Fonte: http://www.forumpcs.com.br/comunidade/viewtopic.php?t=105011

Módulo de memória RAM

O chip de memória RAM mais utilizado é o DRAM (Dynamic RAM), onde cada bit é formado pelo conjunto de um transistor e um capacitor.

Atenção

O transistor controla a passagem da corrente elétrica, enquanto o capacitor a armazena por um curto período;

Quando o capacitor (dois blocos de metal ligados ao transistor) contém um pulso elétrico, temos um bit 1 e quando ele está descarregado temos um bit 0.

Figura C.7: Exemplo de capacitores, à esquerda, e de transistor, à direita
Fontes: http://www.splung.com/content/sid/3/page/capacitors e http://static.infoescola.com/wp-content/uploads/2011/03/transistor.jpg.

Os capacitores conservam o pulso elétrico por apenas uma fração de segundo. Isso acaba gerando um problema que é a perda de dados armazenados nele, imagine que um programa é carregado na memória e logo em seguida é descarregado dela. Para evitar que ocorra essa situação, a placa-mãe possui um circuito de refresh, que é responsável por regravar o conteúdo da memória várias vezes por segundo (cerca de 64 milissegundos).

Atenção

O processo de refresh faz com que aumente o consumo de energia, que também é transformada em calor, e também torna o acesso à memória mais lento, pois o refresh acaba por ocupar ciclos de clock em seu processo.

A solução encontrada para contornar o problema da DRAM foi a de substituir pelo chip de memória SRAM (Static RAM). Ela é formada por cerca de 4 ou 6 transistores, sendo que dois deles controlam a leitura e gravação de dados e os demais formam a célula que armazena o impulso elétrico (cada par forma um inversor, sendo que existem geralmente dois deles). Por esse motivo, ela se torna mais rápida, pois não precisam de refresh, resultando em menor consumo de energia, que acabam sendo mais caras de ser produzidas. Podemos encontrá-la na memória cache.

Podemos destacar, também, alguns outros elementos que compõem o módulo de memória RAM:

- Placa de circuito impresso (Printed Circuit Board – PCB): é a placa onde os chips de DRAM são fixados, possuindo várias camadas formadas por trilhas internas onde são conectados os chips de DRAM do módulo.
- Contatos metálicos: são contatos elétricos do módulo que o conectam na placa-mãe.
- Encapsulamento dos chips (CSP - Chip Scale Package): não usa pinos para se conectar ao PCB, possuindo pequenas esferas de metal em sua parte inferior.

Atualmente, os módulos de DRAM também são conhecidos como SDRAM (Synchronous DRAM). Ele permite que as memórias sejam sincronizadas com o processador, permitindo ao controlador de memória saber exatamente em que ciclo de clock a informação estará disponível para o processador.

Funcionamento da memória RAM

O controlador de memória é o elemento através do qual o processador acessa a memória principal. Sua função é a de coordenar o processo de leitura e gravação de dados na memória, controlando todo o trânsito de dados entre a memória e os demais componentes.

Os módulos de memória estão divididos em linhas e colunas e para acessar um determinado endereço (seja para gravar ou ler dados) o controlador de memória gera a seguinte sequência:

- 1º à valor RAS (Row Address Strobe), que indica o número da linha onde se encontra o dado a ser acessado.
- 2º à valor CAS (Collum Address Strobe), que indica o número correspondente da coluna onde se encontra o dado a ser acessado.

A memória está organizada da seguinte maneira exposta:

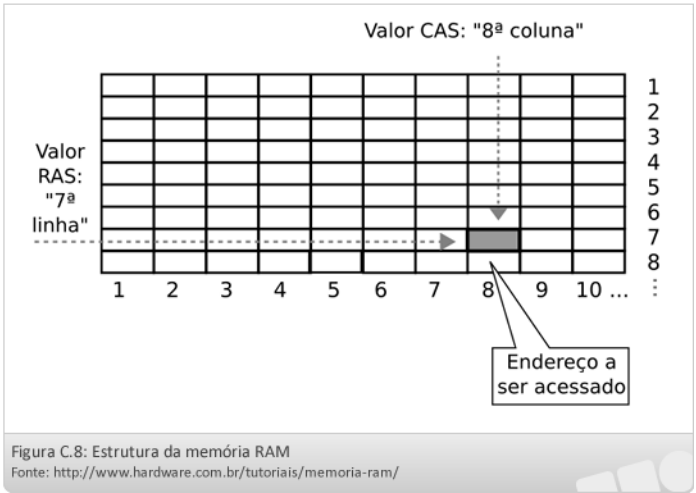


Figura C.8: Estrutura da memória RAM
Fonte: http://www.hardware.com.br/tutoriais/memoria-ram/

A memória RAM é organizada em diversas unidades de armazenamento chamadas de célula, cada uma ocupando um endereço de memória. A célula é a menor unidade da memória, composta por um número fixo de bits (geralmente 8 bits) e identificada por um endereço único e fixo. Existe um barramento comum, que é compartilhado por todos os endereços do módulo para o envio e o recebimento de informações.

Atenção

Conforme abordado anteriormente, a palavra é a unidade utilizada para representar a transferência de dados entre o processador e memória, representando a quantidade de informação que pode ser processada, armazenada ou transferida em uma única operação.

O procedimento de leitura de informações na memória passa, portanto, pelo controlador de memória que envia pares de endereços RAS e CAS em sequências e recebe de volta uma quantidade de informações conforme a largura do barramento de dados (ex: 64 bits). Mesmo que sejam necessários apenas alguns bytes a serem lidos, todo o bloco de bits adjacente é enviando. Por exemplo, se a quantidade de dados solicitada for de 42 bits e a largura do barramento for de 64 bits, serão entregues 64 bits, sendo os 42 bits pedidos mais a quantidade de bits que complete a largura do barramento.

A memória RAM é conectada ao controlador de memória através de três barramentos distintos:

- Barramento de Dados: por onde são transportadas as informações lidas ou escritas.
- Barramento de Endereço: por onde é indicado o endereço de memória em que lidas ou armazenadas as informações.
- Barramento de Controle: indica o tipo de operação que será realizada, bem como o sinal de sincronização (clock).

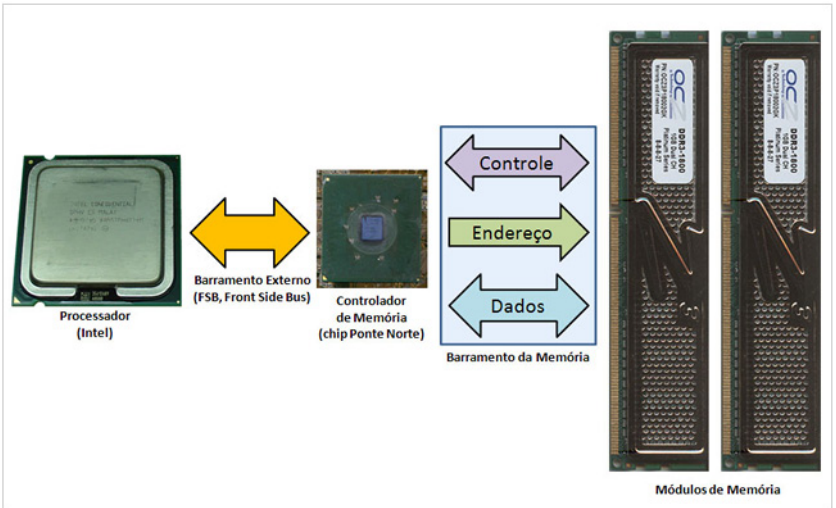


Figura C.9: Relação Processador/Memória
Fonte: http://www.clubedohardware.com.br/artigos/Tudo-o-Que-Voce-Precisa-Saber-Sobre-Memorias-Dual-Channel/673

Nessa relação, o processador é muito mais rápido do que a memória RAM. Dessa forma, são necessárias técnicas de compensação, justamente para tentar minimizar a diferença de desempenho existente entre ambas, de onde podemos destacar:

- Utilização de memória cache (que será abordada posteriormente).
- Aumento da velocidade de acesso aos dados armazenados na memória a partir do aumento da largura do barramento de dados.
- Aumento de canais: possível em chipsets e processadores com controladores de memória dual-channel. Ele consiste em “agrupar” dois módulos de memória distintos como se fossem um só, de onde o controlador de

memória é capaz de acessar dois endereços diferentes (cada um em um dos módulos de memória) a cada ciclo de clock. Isso possibilita a transferência do dobro de dados por ciclo, já que cada módulo possui seu próprio conjunto de barramentos (dados, endereço, controle) e faz com que o processador precise esperar menos tempo, conforme Figura C.10:

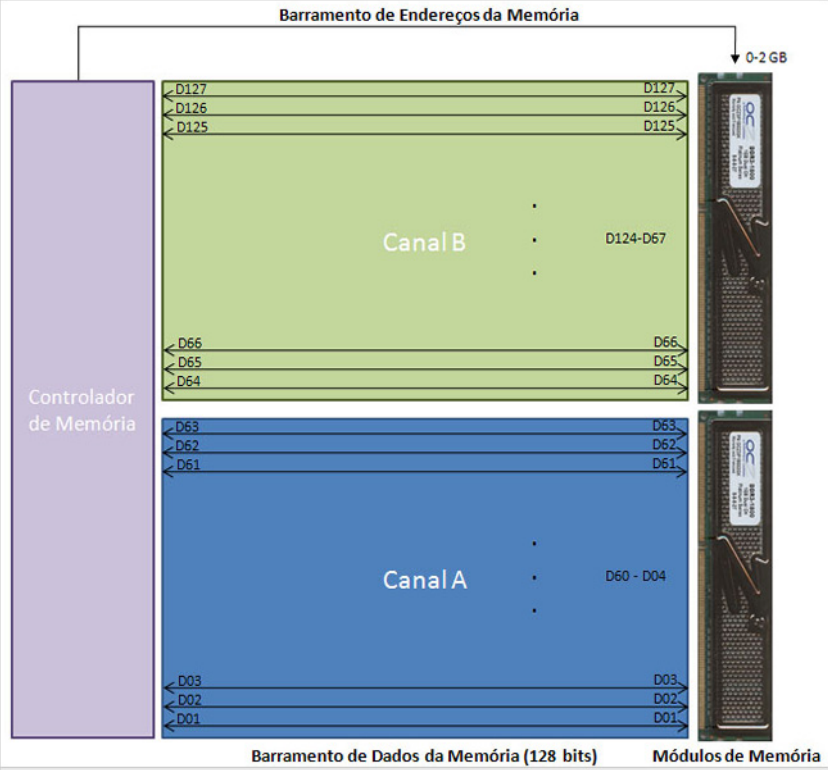


Figura C.10: Dual Channel
Fonte: http://www.clubedohardware.com.br/artigos/Tudo-o-Que-Voce-Precisa-Saber-Sobre-Memorias-Dual-Channel/673/3

Técnicas de Memória RAM

Assim como as técnicas descritas anteriormente, existem outras formas utilizadas visando reduzir a diferença de desempenho entre CPU e RAM como é o caso do DDR.

As memórias DDR são memórias do tipo SDRAM que implementam o duplo fluxo de dados (Double Data Rate), transmitindo duas vezes por ciclo de clock. Ela permite que sejam realizadas duas operações por vez, com cada um dos comandos enviados para um endereço diferente, porém na mesma linha. Ao contrário de como funcionava anteriormente nas memórias SDR (Single Data Rate), onde era realizada uma transferência por ciclo, conforme figura C11.

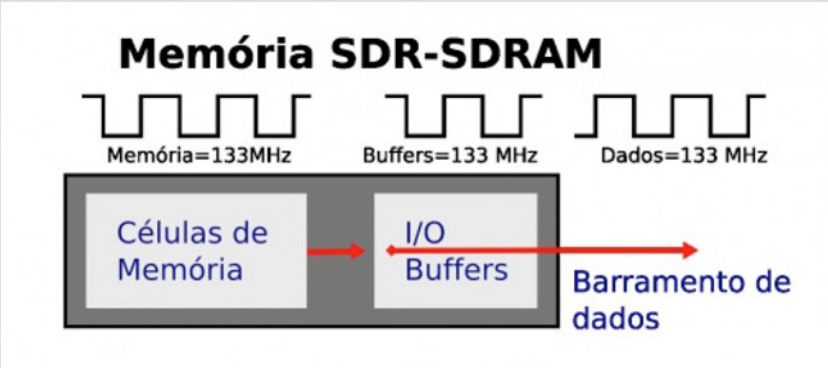


Figura C.11: Operações por ciclo de clock em memória SDR
Fonte: http://www.hardware.com.br/guias/memoria-ram/ddr.html

As duas operações realizadas nas memórias DDR são enviadas através do barramento de dados na forma de duas transferências separadas, uma realizada no início e a outra no final do ciclo de clock, conforme figura C12.

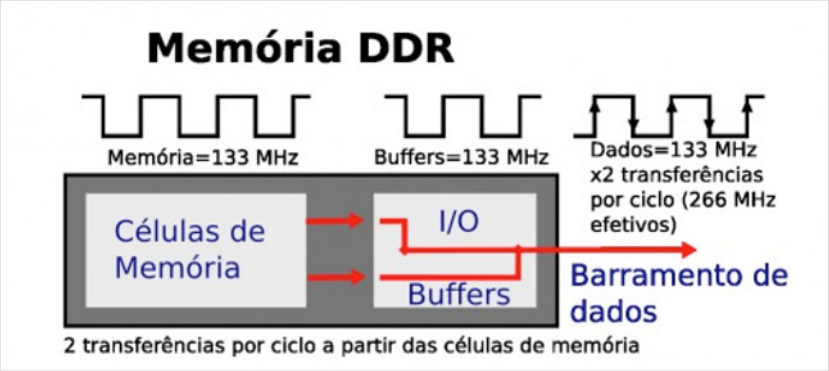


Figura C.12: Operações por ciclo de clock em memória DDR
Fonte: http://www.hardware.com.br/guias/memoria-ram/ddr.html

Dessa forma, como ela é capaz de realizar duas operações por ciclo, essa memória funciona como se operasse com o dobro de seu clock real. Por exemplo, um módulo DDR trabalha a 133 MHz de clock real, mas como são feitas duas transferências por ciclo, o desempenho equivale ao que pode ser obtido por um módulo de 266 MHz, portanto, ela é denominada de DDR266. Esse módulo também é conhecido como PC2100, que indica que a taxa de transferência máxima da memória é de 2100 MB/s.

A taxa de transferência máxima teórica da memória é dobrada com o uso desta tecnologia. Para encontrá-la utilizamos a fórmula

TTMT = Clock DDR x quantidade de bits transferidos por pulso de clock / 8

A memória acima é dita PC2100, pois multiplicando seu clock DDR (266 MHz) pela quantidade de bits que podem ser transferidos por ciclo de clock (64 bits) e dividido por 8 encontramos o valor aproximado a 2100 MB/s (2128 MB/s na realidade).

Com estas características, as memórias DDR seguem a classificação:

DDRccc / PCtttt

Onde temos a seguinte representação:

- ccc = representa o clock máximo em que as memória trabalham.
- tttt = representa a taxa de transferência máxima teórica da memória.

De acordo com essa classificação, encontramos os seguintes módulos de memórias DDR:

- DDR-200 (100 MHz real) = PC1600
- DDR-266 (133 MHz real) = PC2100
- DDR-333 (166 MHz real) = PC2700
- DDR-400 (200 MHz real) = PC3200
- DDR-466 (233 MHz real) = PC3700
- DDR-500 (250 MHz real) = PC4000

Na evolução das memórias DDR surgem as DDR2, que trabalham sob o mesmo princípio, porém duplicam a taxa de transferência, realizando quatro operações por ciclo de clock, mas mantendo as mesmas frequências de operação.

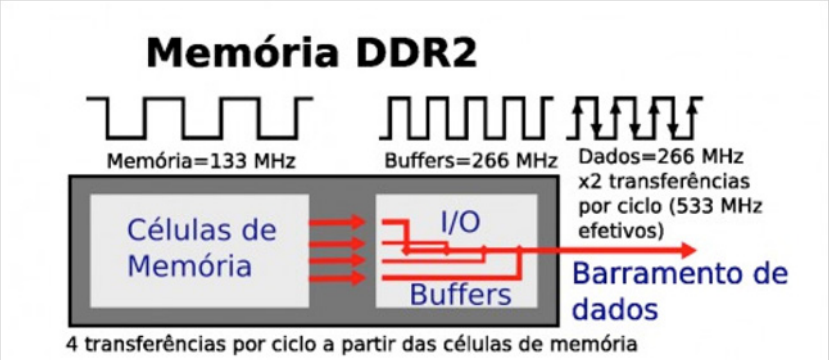


Figura C.13: Operações por ciclo de clock em memória DDR2
Fonte: http://www.hardware.com.br/guias/memoria-ram/ddr2.html

Podemos encontrar os seguintes módulos DDR2 (repare no clock real deles em relação às DDR):

- DDR2-400 (100 MHz real) = PC2-3200
- DDR2-533 (133 MHz real) = PC2-4200
- DDR2-667 (166 MHz real) = PC2-5300
- DDR2-800 (200 MHz real) = PC2-6400
- DDR2-933 (233 MHz real) = PC2-7500
- DDR2-1066 (266 MHz real) = PC2-8500
- DDR2-1200 (300 MHz real) = PC2-9600

As memórias DDR2 seguem o mesmo princípio das DDR, sendo rotuladas com o dobro do seu clock real. Assim, uma memória DDR2-800 opera com um clock real de 200 MHz, mas como realizam 4 operações por ciclo de clock são classificadas como fossem de 800 MHz. Ainda são classificadas como PC2-6400, devido a sua transferência máxima teórica ser de 6.400 MB/s.

Atenção

Se forem utilizados dois canais (dual channel), esse módulo DDR2-800 terá sua taxa de transferência máxima teórica dobrada, de 6.400 MB/s passando para 12.800 MB/s (800 MHz x 128 / 8), já que é transferido o dobro da quantidade de dados, com o barramento trabalhando como se fosse de 128 bits ao invés de 64 bits, a cada pulso de clock.

Atualmente existem as memórias DDR3, que operam com a frequência dos módulos DDR2 duplicada, realizando 8 transferências por ciclo de clock. Esse aumento de frequência é obtido através do aumento na capacidade de acesso simultâneo a endereços que estejam adjacentes e não através do aumento da frequência real.

Nesse caso, em um módulo DDR3-1066, as células de memória operam a 133 MHz de clock real, com os buffers de dados operando a 266 MHz (o dobro do real) e realizando 4 transferências por ciclo (8 vezes o real), resultando em uma frequência efetiva de 1066 MHz.

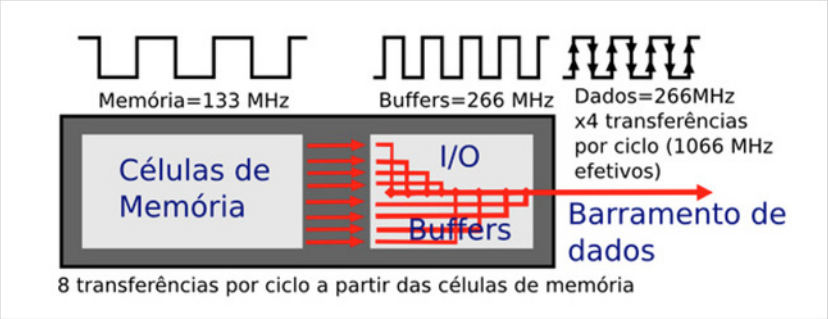


Figura C.14: Operações por ciclo de clock em memória DDR3
Fonte: http://www.hardware.com.br/guias/memoria-ram/ddr3.html

As memórias DDR3 possuem os seguintes módulos (note o clock real delas em relação às DDR e DDR2):

- DDR3-800(100 MHZ real) = PC3-6400
- DDR3-1066 (133 MHz real) = PC3-8500
- DDR3-1333 (166 MHz real) = PC3-10667
- DDR3-1600 (200 MHz real) = PC3-12800

Temporizações de Memória

Além da velocidade, outro fator que igualmente é importante no desempenho da memória são as suas temporizações. Devido a elas, dois módulos de memória que apresentem a mesma taxa de transferência máxima teórica podem apresentar desempenhos diferentes.

As temporizações indicam o tempo em que o módulo de memória RAM leva para realizar internamente uma operação. Como, por exemplo, existe o parâmetro CL, CAS Latency (Latência do CAS), que indica a quantidade de pulsos de clock que o módulo de memória demora em retornar um dado solicitado pelo processador. Um módulo de memória com valor de CL igual a 4 levará quatro pulsos de clock para entregar um dado, enquanto que um módulo com CL igual a 3 levará três pulsos de clock para entregar o mesmo dado.

As temporizações de um módulo de memória são demonstradas através de uma série de números, os quais indicam a quantidade de pulsos de clock que a memória demora para realizar determinadas operações, conforme:

CL – tRCD – tRP – tRAS - CMD

Atenção

Vale lembrar que, conforme visto anteriormente, a memória é organizada internamente em forma de uma matriz, com os dados sendo armazenados na interseção de linhas e colunas, sendo necessário percorrê-las para acessá-lo.

Sendo assim, cada sigla representa uma temporalização de uma determinada operação a ser realizada pela memória, como segue:

- CL: CAS Latency. Tempo demorado entre um comando ter sido enviado para a memória e ela começar a responder.
- tRCD: RAS to CAS Delay. Tempo demorado entre a ativação da linha (RAS) e a coluna (CAS) onde o dado está armazenado na matriz.
- tRP: RAS Precharge. Tempo demorado entre desativar o acesso a uma linha de dados e iniciar o acesso a outra linha de dados.

- tRAS: Active to Precharge Delay. O quanto a memória tem que esperar até que o próximo acesso à memória possa ser iniciado.
- CMD: Command Rate. Tempo demorado entre o chip de memória ter sido ativado (quando o computador é recém-ligado) e o primeiro comando poder ser enviado para a ela. Algumas vezes esse valor pode não ser informado. Normalmente, possui o valor T1 (1 clock) ou T2 (2 clocks).

Dessa forma, podemos encontrar módulos de memória apresentando temporizações como 2 – 3 – 2 – 6 – T1 ou 5 – 5 – 5 – 15 dentre outras, sendo que cada valor indicado representa a mesma ordem, da esquerda para a direita: CL - tRCD - tRP - tRAS - CMD.

Resumo

Ao final da presente unidade vimos que:

- A memória RAM é caracterizada por permitir que qualquer um de seus endereços seja acessado diretamente.
- Ela cumpre o papel de armazenar os dados a serem processados pelo processador, sendo que o acesso ocorre através do controlador de memória por meio dos barramentos.
- A memória DRAM, utilizada como memória principal, é formada por um transistor (passagem da corrente elétrica) e um capacitor (armazenamento), que retêm a informação por um curto período de tempo, necessitando de um sistema de refresh.
- A memória SRAM, utilizada como memória cache, é formada por mais de um transistor, onde alguns são utilizados para controlar leitura e gravação, e outros para ajudar no armazenamento da corrente elétrica pelo capacitor.
- Para o acesso à memória, o controlador de memória gera, nessa ordem, o valor do RAS e o valor do CAS, onde na intersecção destes valores está a célula e onde ficam armazenados os bits.
- Dentre várias técnicas para compensar a diferença de velocidade entre memória RAM e processador existe o uso de memória cache, dual channel e as técnicas de DDR, DDR2 e DDR3.
- Outro fator que determina o desempenho de uma memória são suas temporizações, que indicam quanto tempo a memória leva internamente para realizar determinadas operações.

Questões de revisão

a) Quais são as características da memória RAM e qual é o seu papel em um computador?
b) Quais são os componentes que estão envolvidos em um acesso à memória RAM, e quais são suas utilidades?
c) Diferencie o chip de memória DRAM do SRAM. Como funciona um módulo SDRAM?
d) Como está organizada a memória RAM? Descreva como ocorre o acesso aos dados na memória RAM.
e) Qual é o principio de funcionamento de uma memória DDR? E de uma DDR2?
f) Indique o significado da seguinte classificação de uma memória DDR: DDR-466 PC3700.
g) Calcule a taxa de transferência máxima teórica das seguintes memórias RAM (64 bits): DDR-333, DDR2-400, DDR com clock real de 233 MHz, DDR2 com 233 de clock real.
h) O que é a temporização de memória? O que significam os números que a representam?

Memória Cache

Outro elemento importante do sistema de memória de um computador é a memória cache que, conforme já comentado na unidade B, executa uma importante função, intermediando a relação entre memória RAM e processador.

Cache é um bloco de memória que serve para armazenar, de forma temporária, determinados dados que possivelmente serão utilizados ou que foram utilizados recentemente. Referente a ela, podemos destacar os seguintes fatores:

- tem como característica ser de acesso rápido, intermediando a relação entre quem requisita uma informação e o dispositivo que a armazena;
- busca evitar o acesso aos dispositivos de armazenamento mais lentos, armazenando os dados de forma a estarem mais próximos de quem os solicita;
- é composta por uma fila de elementos, que são cópias exatas de dados presentes em algum outro local (no caso, o local original de armazenamento). Cada elemento possui uma etiqueta que indica o local de armazenamento original (de onde ele foi copiado).

Essa técnica é empregada em diversas situações na informática, sempre com o objetivo principal de agilizar o acesso a determinadas informações que são necessárias a um determinado elemento, como:

- em processadores: onde disponibilizam dados já requisitados e outros a serem processados.
- em navegadores: armazenando localmente páginas acessadas, evitando consultas constantes à rede.
- em redes de computadores: quando o acesso externo à rede ocorre através de serviço de proxy (que compartilha a conexão/link), pode-se armazenar uma lista de sites visitados por usuários da rede.
- em discos rígidos: onde ficam armazenados os dados recentemente acessados nele.

Acesso à cache

Quando o cliente de cache deseja acessar um dado no local de armazenamento original (memória RAM, HD, Internet, etc.), primeiramente é verificado se ele não está presente na cache. Se o dado necessário for encontrado na cache com a etiqueta correspondente ao desejado, o que está na cache é enviado ao invés do original.

Nessa situação, quando o cliente acessa somente a cache , ocorre o chamado cache hit (acerto do cache), que justamente é a situação em que a informação a ser acessada é localizada na cache, evitando o acesso à mídia de armazenamento inferior na hierarquia, que consequentemente é mais lenta. A quantidade de acertos é conhecida como a taxa de acerto (hit rate ou hit ratio) da cache. Quanto maior essa taxa, melhor é o desempenho.

Por outro lado, quando a informação requerida não é localizada na cache, ocorre o cache miss (erro do cache). Dessa forma, a informação deve ser copiada do local original de armazenamento e inserida na cache, sendo disponibilizada para o acesso.

Sendo necessário armazenar um novo dado na cache e não havendo espaço disponível, pois todas as posições estão ocupadas, devem-se remover determinados elementos para liberar espaço. A forma utilizada para selecionar o elemento a ser retirado é comandada por uma política de troca (replacement policy). Uma das políticas de troca mais utilizada é a LRU (least recently used), que remove o elemento recentemente menos usado.

Armazenamento de informações da Cache

Quando um dado é armazenado na cache, como resultado de um processamento realizado pelo processador, em algum momento ele deve ser gravado de volta no local de armazenamento original.

Atenção

Como o processador trabalha apenas com informações em registradores, após executá-las, grava os resultados na memória cache, liberando os registradores para as novas informações a serem processadas.

É função da política de escrita (write policy) determinar o momento em que ocorrerá a gravação dos dados da cache em seu local original. Para isso, existem os seguintes tipos:

- **Política write-through** (escrita através): cada vez que um elemento é escrito na cache, ele também é gravado no local de armazenamento original. Assim, a memória que está hierarquicamente abaixo possuirá a informação atualizada, facilitando a remoção do conteúdo da cache. Por outro lado, cada operação na cache resulta uma operação na memória, o que torna o acesso mais lento.
- **Política write-back** (escrever de volta): as escritas são realizadas apenas na cache. Quando a informação for ser removida, são identificados quais dos elementos foram alterados e somente essas posições são escritas de volta nos locais de armazenamento originais. Com isso, é reduzido o tempo que se leva para escrever na memória, pois é reduzida a quantidade de operações de escrita na memória. No entanto, quando um dado que deve ser removido (para liberar espaço para outro) foi modificado, é necessário que o novo conteúdo aguarde o conteúdo a ser removido ser escrito no outro nível da hierarquia. Outro problema é que nem sempre existirá consistência entre os dados que estão na cache e na memória.

Atenção

Em relação à política de write-back, os dados que estão armazenados nos locais originais de armazenamento podem ser alterados por outros elementos além da cache. Assim, a cópia que está na cache pode não ser mais válida como, por exemplo, a existência de mais de uma cache pode resultar que o mesmo conteúdo que está em caches diferentes pode ser alterado em uma delas, tornando a outra inválida.

É necessário, então, que os protocolos de comunicação entre gerentes de cada uma das caches mantenham os dados consistentes (protocolos de coerência).

Organização de cache

A cache pode ser organizada de duas formas: separando dados das instruções ou unificada.

Na cache separada, são utilizados locais distintos para armazenar os dados e as instruções, permitindo que o processador acesse simultaneamente instruções (na cache de instruções) e dados (na cache de dados). Como geralmente as instruções dos programas não são alteradas, estas informações podem ser facilmente descartadas e como instruções ocupam menos espaço, é possível destinar mais espaço para armazenamento de dados.

Já na cache unificada é utilizado o mesmo espaço de armazenamento tanto para instruções quanto para dados.

Cache de processador

Como forma de minimizar a diferença de velocidade entre processador e memória RAM passou a ser usada a memória cache, baseada na memória SRAM. Trata-se de uma memória de acesso rápido, servindo para armazenar os dados mais frequentemente usados pelo processador.

O seu objetivo é o de evitar, na maioria das vezes, que o processador tenha que acessar a memória principal, que, comparativamente, é mais lenta. Atualmente os sistemas utilizam mais de um nível de cache na hierarquia de memória, sendo a cache primária, ou cache L1 (level 1), cache L2 (level 2) e cache L3 (level 3).

A cache L1 fica próxima ao núcleo do processador e o acompanha em velocidade, apresentando tempos de latência extremamente baixos. Ela é organizada separando a área de instruções da área de dados.

As demais caches são organizadas de forma unificada, sendo que cada nível hierarquicamente inferior (L2, L3) apresenta maior capacidade de armazenamento, porém maior distância do processador, resultando em tempos de acesso superior em relação às que se encontram nos níveis superiores.

A leitura de dados no processador começa sempre no cache L1, caso o dado seja encontrado, o processador não perderá tempo. Caso o dado não esteja no cache L1, o próximo a ser acessado será o cache L2, encontrando o que procura nele, o processador já perderá algum tempo, mas não tanto quanto na RAM, e assim sucessivamente. Caso os dados não estejam em nenhum dos níveis de cache, a memória RAM deverá ser acessada, resultando em um processo mais lento. Quando o processador precisa ler dados na memória principal, o controlador de cache transfere blocos de dados da RAM para a cache do nível mais baixo até a cache L1 que será acessada pelo processador. Conforme segue:

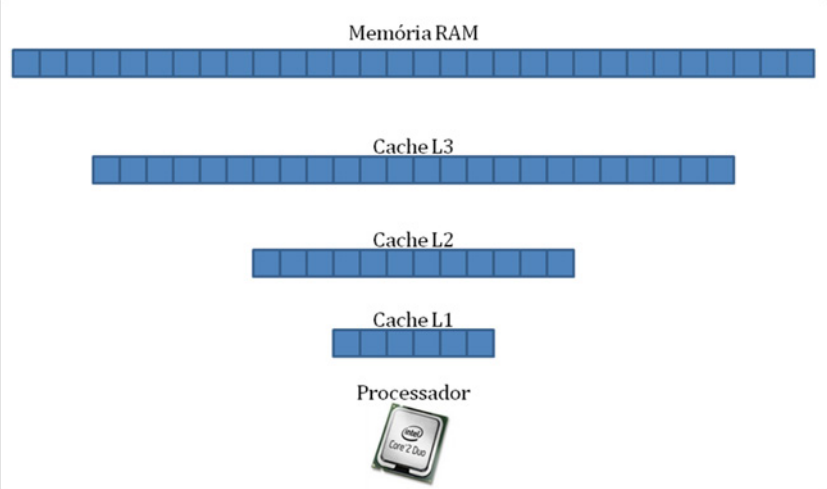


Figura C.15: Posição das memórias em relação ao processador.
Fonte: http://1.bp.blogspot.com/_gVb00JTqXR4/SXRmwEsiTiI/AAAAAAAAAdE/jwRTf7LbGJo/s1600-h

Referente ao tempo de acesso, para efeitos de comparação, temos os seguintes valores:

- Cache L1: cerca de 3 a 4 ciclos
- Cache L2: cerca de 10 a 15 ciclos
- Memória RAM: cerca de 140 ciclos

Função da cache de processador

A cachê, anteriormente, era limitada a armazenar cópias das últimas informações acessadas pelo processador, descartando as informações mais antigas ou menos acessadas trabalhando de forma unificada.

Atualmente, a cache, como visto anteriormente, pode trabalhar de forma unificada ou separando instruções de dados. Ela também conta com sistema de prefetch, que monitora o fluxo de instruções, carregando antecipadamente dados que poderão ser necessários nos ciclos seguintes.

Ela acelera a operação de escrita, pois o processador grava diretamente nela e o controlador de cache

fica encarregado de gravar os dados na memória RAM posteriormente.

A cache L1 passou a ser dividida em dois blocos independentes, sendo um destinado para o armazenamento de dados e outro para o de instruções, o que possibilita que o processador leia dados e instruções simultaneamente.

A cache exerce contato direto com a memória principal, seja gravando ou lendo informações. Para agilizar essa relação existe um elemento chamado TLB (Translation lookaside buffer), que fica localizado entre a cache L2 (ou L3, se for o caso) e a memória RAM. Ele armazena os endereços de memória, convertendo os endereços lógicos (que são usados por aplicativos em execução) em endereços físicos presentes na memória principal. Isso é necessário, pois no momento da execução de um processo, ele referencia endereços virtuais, com os quais eles trabalham, mas que devem ser traduzidos para endereços reais, que podem ser acessados pelo processador.

Resumo

Ao final da presente unidade vimos que:

- O conceito de cache é utilizado nas mais diversas situações (processadores, navegadores, discos rígidos, etc.) para armazenar temporariamente dados que possivelmente serão, ou que recentemente foram utilizados.
- O acesso a uma informação é feito a partir da cache, evitando acessar a memória de armazenamento sempre que possível.
- Quando a informação solicitada está na cache, ocorre o cache hit, evitando acesso a meios mais lentos;
- Quando a informação não está na cache, ocorre o cache miss, sendo necessário buscá-la da memória de armazenamento original e, com isso, pode surgir a necessidade de liberar espaço para essa nova informação, de acordo com a política de troca.
- A informação que é armazenada na cache em algum momento deve ser repassada para o meio de armazenamento original. Isso é determinado pela política de escrita, que são duas: write-through, que a cada alteração na cache a repassa para o local orginal, e a write-back, que somente grava no local original quando a informação foi alterada, o que pode acarretar em situações como a coerência de cachê.
- As caches podem ser organizadas dividindo a área de dados da de instruções, o que permite acesso simultâneo a elas, ou, por outro lado, podem ser unificadas, com dados e instruções dividindo a mesma área.
- No processador, a memória cache é utilizada para evitar o acesso à memória RAM, dividida em vários níveis (L1, L2, L3), sendo que a busca por informações começa a partir do primeiro nível, descendo sucessivamente a hierarquia até encontrá-la.
- A função da cache, com o passar da evolução, foi sendo aperfeiçoada, passando do simples armazenar às últimas informações solicitadas, para auxiliar a agilizar o desempenho do processador na manipulação de informações.

Questões de revisão

a) Qual é o princípio básico da cache e onde ela é utilizada?
b) Como funciona o acesso na cache? Que tipos de situações em que podem ocorrer o acesso?
c) Quais são e como funcionam as políticas de escrita da cache?
d) Detalhe como a memória cache é utilizada no processador e qual sua função desempenhada.
e) Quais são as principais características desempenhadas pelas cache atualmente?
f) O que é TLB?

Memória Secundária

No sistema de memória de um computador, além das memórias voláteis que armazenam os dados e instruções para que sejam acessados e executados pelo processador, também são necessárias memórias que armazenem permanentemente estas informações para posterior acesso.

A memória secundária é utilizada justamente com a finalidade de armazenar permanentemente as informações em um computador, de forma que, mesmo sem alimentação elétrica, ele mantenha as informações nele contidas. O principal elemento que desempenha essa função em um computador é o disco rígido (hard disk = HD).

Conforme visto anteriormente, quando uma informação necessita ser carregada para a memória principal, geralmente ela é carregada do disco rígido, como, por exemplo, o sistema operacional que, quando está instalado no computador, no momento em que a máquina é ligada deve ser carregado do disco rígido, assim como os programas e arquivos.

Além dessa função, o disco rígido também pode se relacionar com a memória RAM em outras duas circunstâncias: memória virtual e cache de disco.

Técnicas envolvendo Disco Rígido e Memória RAM

Quando toda a área de armazenamento da memória RAM está ocupada, pode-se utilizar uma parte da memória secundária para simular a existência de mais endereços na principal. Essa técnica é conhecida como memória virtual ou também como swap. Dependendo do sistema que a utiliza, ela pode ser armazenada em um arquivo no disco rígido ou em uma partição.

Essa técnica é possível devido à utilização dos endereços virtuais pelos programas, que possibilitam o carregamento e o armazenamento dos dados dos programas nas mais diversas localizações. Dessa forma, o programa pode ser dividido em diversas partes e cada uma delas pode ser alocada na memória da forma que melhor convir ao sistema, bem como, parte do programa pode estar na memória RAM (trecho essencial do programa) e outra parte pode estar na memória virtual (HD).

Quanto menor for a quantidade existente de memória RAM, mais será utilizada a memória de swap. Com isso ocorre perda de desempenho no sistema já que o tempo para acessar um HD é superior ao da memória RAM. Além disso, os programas em execução não acessam dados seus armazenados na memória secundária, sendo assim, quando for necessário um dado que está na memória virtual, ele deve ser conduzido até a memória principal.

Atenção

A área de endereços da memória é dividida em partes denominadas páginas, que armazenam blocos contínuos de dados, que podem estar na RAM ou no HD, sendo que quando uma página de dados é referenciada ela deve ser necessariamente copiada para a memória principal, caso não esteja lá.

Quando um programa está em execução, ele apenas trabalha com endereços virtuais, que são os que ele vê, porém, na verdade, ele ocupa endereços físicos na memória principal. É necessário que ocorra a tradução entre esses endereços, convertendo endereços virtuais em físicos no momento da execução do programa pelo processador. Para facilitar esse processo existe uma tabela de páginas que mantém um mapeamento relacionando os endereços virtuais dos programas em endereços físicos da memória

principal. Através dessa tabela é possível identificar se uma página solicitada já está na memória RAM ou se ela está na memória virtual e deve, conseqüentemente, ser carregada para a RAM.

Para lidar com essa situação existem dois métodos. Um deles é o de paginação por demanda, onde as páginas somente são transferidas para a memória principal quando solicitadas pelo programa, dessa forma apenas as páginas necessárias ficam na memória, perdendo menos tempo com transferências, porém pode acontecer de faltar páginas durante a execução do programa e elas terem de ser buscadas do HD, acarretando em demora. Outro método empregado é o de swapping, que carrega todos os dados de um programa como um bloco único, com isso a transferência dos dados do programa é realizada em uma única operação, com isso todos os dados necessários estarão na memória, porém levará mais tempo para realizar as transferências.

Outra técnica que envolve o disco rígido e a memória RAM é a cache de disco. Nela, quando existem posições livres na memória principal, são copiados dados a mais do disco rígido para a memória principal, agilizando o acesso a eles, quando necessário. Através disso, cria-se a impressão de que o acesso ao HD é mais rápido do que na realidade.

Composição do disco rígido

O disco rígido armazena as informações em formato magnético, o que mantêm a gravação permanentemente. Os dados são gravados em discos magnéticos, chamados de platters (pratos), que são compostos por duas camadas:

- substrato: formada por um disco metálico, feito de ligas de alumínio, perfeitamente plano;
- superfície magnética nas duas faces do disco, recoberta por uma camada protetora contra pequenos impactos;

Na fabricação dos discos, a superfície magnética é formada por grãos microscópicos depositados de forma uniforme por toda a superfície. Quanto menores forem os grãos, mais altas serão as densidades do disco, medida em gigabits por polegada quadrada, sendo que quanto maior a densidade, maior será a quantidade de grãos depositados e, conseqüentemente, a quantidade de informações que podem ser armazenadas no disco.

Todos os discos que compõem o HD são montados em um eixo de alumínio para evitar vibrações, sendo que existem espaçadores entre eles. Um motor de rotação é o responsável por manter um ritmo constante para que os discos rotacionem por minuto (motores de 5.400, 7.200, 10.000 RPM).

Estrutura do disco rígido

No processo de leitura e gravação de dados no disco rígido, são usadas cabeças de leitura eletromagnéticas (heads) que são presas a um braço móvel (arm), que é movimentado pelo actuator, o que permite acesso a toda a área do disco.

Dependendo da capacidade, os HDs podem ser compostos por vários discos. Nesse sentido, o HD possui duas cabeças de leitura para cada disco (sendo uma para cada face), de forma que se ele for formado por 4 discos, existirão 8 cabeças de leitura, todas elas presas ao mesmo braço móvel. Isso significa que quando uma das cabeças de leitura precisa ser movimentada, todas as outras serão movimentadas também, sobre suas respectivas faces. Podemos observar estes elementos a partir da figura abaixo:

Figura C.16 - Elementos do disco rígido
Fonte: http://www.hardware.com.br/guias/hds/como-hds-funcionam.html

Para que a estrutura de um disco rígido permita o acesso e armazenamento de elementos, cada disco que o compõem é organizado em setores, trilhas e cilindros, formando o que também é conhecido como geometria dos discos. Podemos visualizá-la na figura abaixo:

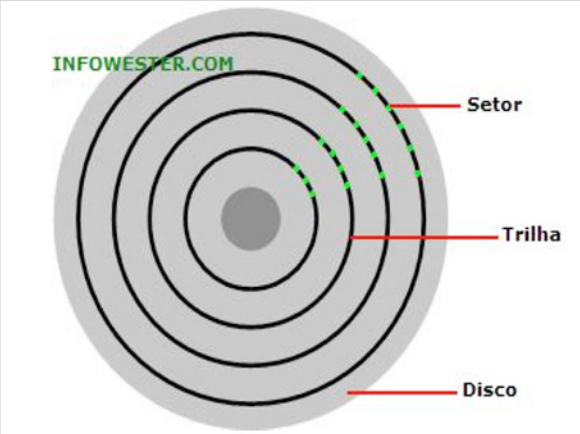


Figura C.17 - Geometria de um disco
Fonte: http://www.infowester.com/hd.php

Na organização acima, as trilhas são as circunferências, que são numeradas seqüencialmente a partir da borda do disco em direção ao centro (reduzindo seu diâmetro gradualmente). A trilha que mais próxima da extremidade do disco é denominada trilha 0, a próxima trilha é a trilha 1, e assim por diante, até chegar à trilha mais próxima do centro. Isso se repete para cada face de cada disco. Por sua vez, as trilhas são divididas em setores, que são áreas de tamanho fixo, onde efetivamente ficam armazenados os dados em um disco.

Para ler os dados que ficam localizados nesta estrutura são utilizadas as cabeças de leitura e gravação que são movimentadas até um local específico do disco pelos braços moveis. Conforme vimos anteriormente, esses braços são todos unidos a um mesmo ponto, sendo que quando um dos braços é movimentado, todos os demais também serão. Por exemplo, para ler um dado que está na trilha 100 da face de disco 2 do 1º disco, a controladora do disco aciona a cabeça de leitura especifica pela face de disco 2 do 1º disco, juntamente com ela todas as outras cabeças também serão movimentadas para a trilha 100 de suas respectivas faces.

Devido ao fato de que todas as cabeças ficam posicionadas sobre a mesma trilha de cada uma das faces de cada disco, surge o cilindro. Dessa forma, um cilindro é formado pelo conjunto das trilhas de mesmo número de cada face de cada disco, onde, por exemplo, o cilindro 50 é composto pela trilha 50 de cada

face de cada disco, e assim sucessivamente, conforme imagem abaixo:

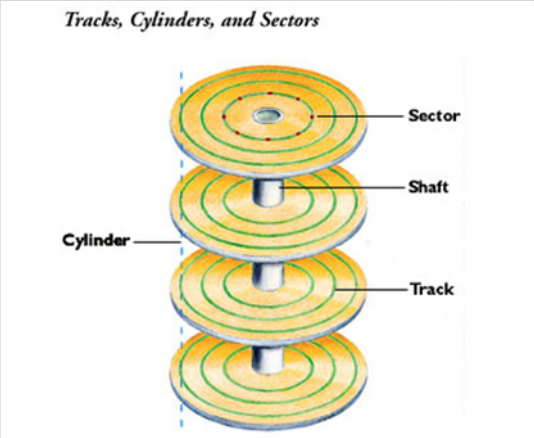


Figura C.18 - Composição do cilindro
Fonte: http://www.hardware.com.br/livros/hardware/discos.html

Operações no disco rígido

Quando da leitura ou da gravação de dados no disco rígido, os braços são movimentados através de atração e repulsão eletromagnética sob o controle do actuator. Para isso, existe um eletroímã na base do braço móvel, que permite que a placa controladora o movimente através da variação na potência e na polaridade do ímã.

Para facilitar o posicionamento da cabeça de leitura e gravação sobre a superfície dos discos, são gravados sinais magnéticos em determinados pontos da superfície dos discos que servem para orientação de que em que parte do disco a cabeça está naquele momento.

Sabendo A superfície de gravação dos pratos é composta de materiais sensíveis ao magnetismo, conforme visto no item Composição do disco rígido. Dessa forma, a cabeça de gravação manipula as moléculas da superfície da posição do disco na qual deseja fazer a gravação através de seus pólos. Nesse processo a polaridade da cabeça muda numa freqüência alta, quando ela está positiva atrai o pólo negativo das moléculas e vice-versa. De acordo com essa polaridade é que são gravados os bits (0 e 1).

Na leitura dos dados no disco, inicialmente a controladora posiciona a cabeça de leitura sobre a trilha onde está o setor a ser lido, aguardando o movimento de rotação do disco posicionar o setor sob ela. A partir desse momento a cabeça lê o campo magnético das moléculas do setor e gera uma corrente elétrica correspondente que a controladora do HD utiliza para determinar os bits da informação.

Nestes dois momentos, na gravação ou na leitura, a cabeça de leitura e gravação em momento algum encosta na superfície do disco, mesmo que a disposição dos braços seja incliná-los em direção a ela. O motivo de não ocorrer esse contato é devido à alta velocidade de rotação dos discos que acaba formando um colchão de ar que repele a cabeça de leitura, o que garante que ela sempre mantenha certa distância dos discos (geralmente medida em nanômetros).

A figura abaixo ilustra essa disposição. Nela temos um braço com duas cabeças de leitura e gravação, uma para cada face do disco, e podemos observar a tendência natural de ambas se aproximarem.

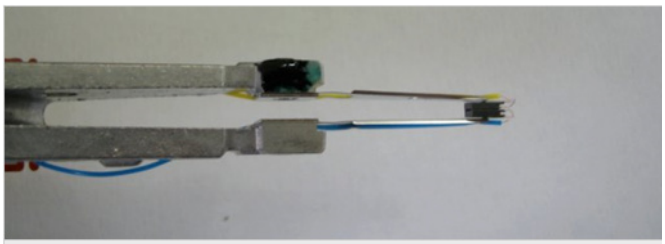


Figura c.19 - Posição das cabeças de leitura e gravação
Fonte: http://www.hardware.com.br/livros/hardware/como-funciona.html

Quando o disco está desligado, as cabeças de leitura ficam em uma posição de descanso, afastando-se dos discos, assim como no momento em que não estão sendo lidos dados.

Placa controladora

A placa controladora é a parte do HD responsável pelo processamento, realizando a interface com a placa-mãe, controlando a velocidade de rotação do motor e o movimento das cabeças de leitura, verificando as leituras, identificando erros, etc. Nele existe um chip de memória SDRAM que é utilizada como uma pequena quantidade de memória cache para o disco rígido, armazenando os últimos dados acessados, o que pode agilizar o acesso a uma informação que foi recentemente lida já que não seria necessário acessar os discos novamente.

Também faz parte o controlador principal, que executa todo o processamento do disco rígido. É ele que se comunica com a placa-mãe através de comandos padronizados de discos rígidos, sendo que internamente estes comandos são convertidos para ações a serem executadas pela cabeça de leitura e gravação. Isso garante que existe compatibilidade dos mais diversos modelos de HD com a placa-mãe, independente de seu funcionamento interno.

Os comandos gerados pelo controlador principal são recebidos pelo controlador de movimento das cabeças de leitura e da rotação do motor. A partir disso, ele envia a seqüência de ações necessárias para acessar um determinado setor.

Correção de erros

Podem ocorrer pequenas falhas na superfície da mídia magnética dos discos que podem levar a erros de leitura. Elas podem ser mais frequentes em discos com densidade de gravação maiores (100 gigabits ou mais), assim como naqueles com altas velocidades de rotação (7.200 RPM para mais).

Para tentar detectar e corrigir esses erros de leitura são utilizados sistemas de ECC (“Error Correcting Code”: “código de correção de erros”). Nele são adicionados bits a mais para cada byte armazenado no disco. Esses bits são gerados por algoritmos especiais aplicados nas informações do disco e que serão utilizados posteriormente para, através da aplicação novamente de algoritmos, verificar se houve alguma alteração nos dados armazenados.

Quando os erros de leitura se tornam cada vez mais freqüentes em um disco isso pode ser gerado por um badblock, que é um defeito físico na mídia magnética, onde os dados que ocasionalmente são armazenados nele não conseguem mais ser lidos. A solução para isso é marcar as áreas defeituosas, para que não sejam mais utilizadas. Existe então uma área reservada no início do disco chamada de defect map (mapa de defeitos). Quando a controladora do disco identifica um setor defeituoso, o substitui por um setor que não apresenta problemas, registrando isso no defect map e com isso evitando que ele seja utilizado novamente. O único porém é o de que a área de HD destinada para isso é limitada, sendo que

em algum momento pode acontecer do mapa de defeitos lotar, com isso os badblocks terão que passar a ser tratados pelo sistema operacional.

Desempenho

O desempenho do HD está diretamente ligado às etapas necessárias para se acessar uma posição específica nele. Nesse sentido, podemos destacar o tempo de busca, o tempo de latência e o tempo de acesso.

O tempo de busca (seek time) indica o tempo que a cabeça de leitura e gravação leva para ir de uma trilha para outra do disco, sendo que o seu cálculo pode ser baseado em três índices:

- Full Stroke: determina o tempo que a cabeça leva para se deslocar da primeira até a última trilha (geralmente entre 15 e 20 milissegundos);
- Track-to-Track: é o tempo que a cabeça leva para mudar de uma trilha para a seguinte (como a distância é pequena, costuma ser inferior a 1 milissegundo);
- Average (valor médio): é um valor intermediário entre os dois anteriores, indicando o tempo médio que a cabeça demora para se locomover até um setor aleatório do HD.

Após a localização da trilha, existe o tempo de latência (latency time), que é o tempo em que a cabeça fica para da sobre a trilha aguardando que o setor a ser acessado, através do movimento de rotação do disco, passe sob ela. Pode ser que o setor esteja logo a seguir ou, no pior dos casos, se tenha que aguardar uma volta inteira do disco para acessá-lo. Esse tempo é calculado dividindo 60 pela velocidade de rotação do HD em RPM e multiplicando o resultado por 1000 resultando no tempo em milissegundos. Em um disco com 5.400 RPM obteremos o tempo de latência de 11.11 ms (60 ÷ 5400 x 1000 = 11.11), sendo que é utilizado o tempo médio de latência, equivalente à metade de uma rotação do disco, nessa caso o tempo seria de 5.55 ms.

Assim que o comando para acessar uma determinada posição é recebido, a cabeça de leitura é movida para a trilha especificada (tempo de busca) e aguarda até que a rotação dos discos faça setor especificado passar por ela (tempo de latência). A eles, soma-se também o settle time (o tempo que a cabeça de leitura demora para estabilizar depois de movimentada) e o command overhead time, que é o tempo que a placa controladora demora para processar o comando e iniciar as operações. Geralmente esses dois valores juntos representam algo em torno de 0,5 ms.

O tempo de Acesso (access time) a uma posição do disco será então a soma do tempo de busca e do tempo de latência, juntamente com o settle time e o command overhead time resultando no tempo médio que é necessário para se realizar o acesso a um setor aleatório do HD. O calculo é realizado somando o tempo de busca médio (average) e o tempo de latência, adicionando 0,5 ms correspondente ao settle time e o command overhead time. Um HD com tempo de busca médio de 8,9 ms e latência de 4,15 ms, adicionando 0,5 ms do settle time e do command overhead time, resultando em um tempo de acesso de 13,55 ms.

Cache (Buffer)

Quando um arquivo vai ser lido no disco, isso acaba gerando várias leituras de setores seqüenciais. Utilizando Através do uso da cache, em cada passagem da cabeça de leitura são lidos os setores próximos, independente de terem sido solicitados ou não, dessa forma, quando for solicitada a leitura do próximo setor, ele estará carregado na cache, sendo que o dado será transferido mais rapidamente.

O cache pode ser usado também nas operações de escrita, principalmente se o disco estiver ocupado realizando outras operações. A controladora pode armazenar na cache a operação de escrita e executá-la posteriormente, quando o disco estiver liberado.

Resumo

Ao final da presente unidade nós vimos que:

- A memória secundária é utilizada para de armazenar permanentemente as informações em um computador;
- Na relação existente entre disco rígido e memória RAM podem ser utilizadas técnicas como a memória virtual e a cache de disco;
- As gravações ocorrem de forma magnética nos discos que são compostos por materiais sensíveis ao magnetismo;
- Em cada disco (platter) são utilizadas ambas as faces, tanto a superior quanto a inferior, para o armazenamento de dados, sendo que podem existir mais de um disco compondo o disco rígido e para cada face de cada um deles existe uma cabeça de leitura e gravação, fixadas no mesmo ponto, para acessá-la;
- Os discos são organizados em trilhas, circunferências que dividem a face de um disco, setores, que subdividem as trilhas e onde os dados ficam armazenados, e cilindros, compostos pela mesma trilha de todas as faces de cada disco;
- Na operação de escrita a cabeça de leitura e gravação altera sua polaridade e, com isso, a disposição dos elementos de cada setor, armazenando as seqüências de 0 e 1 necessárias. Na leitura, a cabeça de leitura e gravação lê os sinais magnéticos de cada posição, gerando os sinais equivalentes da seqüência de 0 e 1 armazenadas;
- A placa controladora é quem recebe as solicitações e coordena as ações necessárias para acessar os discos, seja na leitura ou na gravação de dados;
- Os HDs possuem um sistema para lidar com erros nos discos, utilizando códigos ECC para futura tentativa de detecção e correção de erros, bem como armazenando os badblocks para que não sejam mais utilizados;
- O desempenho do HD está ligado ao seu tempo de busca (localizar a trilha), tempo de latência (aguardar o setor a ser lido passar pela cabeça), settle time (estabilizar a cabeça sobre a trilha), command overhead time (tempo para processar a solicitação) e tempo de acesso (envolve os demais tempos para determinar tempo médio para acessar um setor no disco);
- Existe um buffer de cache utilizado para armazenar os setores recentemente lidos (agilizar posterior acesso) e para armazenar valores a serem escritos no disco enquanto ele estiver ocupado com outras operações.

Questões de revisão

a) Diferencie as memórias RAM, Cache e a secundária.
b) Qual é a diferença entre memória virtual e a cache de disco?
c) Como funciona a técnica de fabricação dos platters?
d) Como é composto um disco? Como é sua geometria?
e) Destaque o funcionamento das cabeças de leitura e gravação em relação ao movimento e ao posicionamento. Como ocorre o processo de gravação e de leitura?
f) Para que serve a memória cache (buffer) no disco rígido?
g) Como funciona a correção de erros no disco rígido?
h) Quais são os itens que influenciam no desempenho do HD? Como?

Atividades

1. Sobre a hierarquia de memória, é incorreto afirmar o seguinte:

a) O objetivo da hierarquia de memória é o de evitar o acesso às memórias mais lentas.
b) O sistema de memória do computador é organizado em uma estrutura hierárquica em que as memórias de maior capacidade de armazenamento ficam próximas ao processador.
c) Quanto mais no topo da hierarquia, maior é a frequência de operação e menor é a capacidade de armazenamento da memória.
d) A busca por uma informação solicitada inicia sempre pelos níveis mais altos da hierarquia.

2. Em relação às características das memórias, é correto afirmar que

a) a capacidade determina a quantidade de informação que pode ser armazenada na memória.
b) a tecnologia de fabricação distingue se a memória armazena permanentemente ou não as informações.
c) o tempo de acesso determina quanto tempo uma informação fica armazenada na memória.
d) a volatilidade indica se a memória é de semicondutor ou de meio magnético.

3. Sobre os tipos de memória ROM, é incorreto dizer que

a) a memória ROM utiliza um diodo na intersecção entre linhas e colunas para permitir a passagem de corrente nas posições pretendidas, sendo gravada de fábrica.
b) a memória PROM utiliza um fusível para conectar todas as linhas e colunas, permitindo que ela seja escrita uma única vez.
c) a memória EPROM possui em cada intersecção de linha e coluna dois transistores fazendo a ligação, permitindo que ela possa ser apagada aplicando radiação em um processo não seletivo.
d) a memória EEPROM permite que seja realizada apenas uma vez a gravação de informações nela.

4. Sobre memória principal, é incorreto afirmar que

a) utiliza o chip de memória SRAM, que possibilita acesso mais rápido às informações armazenadas.
b) é uma memória de acesso aleatório, formada por diversas células idênticas, organizadas em uma estrutura de matriz, onde ficam armazenados os dados.
c) o chip DRAM possui necessidade de refresh.
d) para acessar uma de suas posições é necessário o valor de RAS e o valor de CAS.

5. Sobre a organização da memória RAM, é correto afirmar que

a) a menor unidade da memória é a célula, que pode possuir um número variável de bits.
b) a palavra é a unidade utilizada para indicar a frequência de operação da memória.
c) toda a comunicação entre memória e demais componentes do micro é coordenada pelo controlador de memória.
d) existe um barramento de dados em cada posição da memória para transferência de dados para a CPU.

6. Em relação à temporização de memória, é correto afirmar que

a) o CL indica o tempo entre ligar o computador e fazer o primeiro acesso à memória.
b) o tRP indica o tempo que é necessário esperar até iniciar um novo acesso à memória.
c) o tRAS indica o tempo entre terminar o acesso a uma linha e iniciar o acesso a outra.
d) o tRCD indica o tempo entre a ativação da linha e ativação da coluna.

7. Sobre memória cachê, é incorreto afirmar que

a) ela possui cópias exatas de elementos que estão em memórias hierarquicamente inferiores.
b) na política write-through, cada alteração na cache é gravada no local original automaticamente.

c) quando ocorre um cache hit, uma informação deve ser buscada de uma memória hierarquicamente inferior.
d) a política write-back apresenta o problema de consistência entre dados da cache e da memória original.

8. Sobre memória Virtual, é correto afirmar que

a) ela utiliza espaço no disco rígido para simular existência de mais memória RAM.
b) a paginação por demanda carrega todos os dados de um programa para a memória principal.
c) armazena todos os dados e instruções dos programas em execução na memória principal.
d) na técnica de swapping os dados são carregados em partes para a memória principal e somente quando solicitados.

9. Sobre a memória secundária, é incorreto afirmar que

a) é composta por um ou mais platters, que são compostos por um disco metálico coberto, em ambos os lados, por uma segunda camada composta por uma superfície magnética.
b) o disco rígido é utilizado como forma de armazenamento permanente de dados.
c) a densidade de gravação é determinada pelo tamanho dos grãos magnéticos depositados sobre os platters.
d) para realizar a gravação e a leitura de dados a mesma cabeça de leitura e gravação eletromagnética acessa todos os platters do disco.

10. Sobre o desempenho de um disco rígido, é correto afirmar que

a) o tempo de latência indica o tempo em que a cabeça de leitura e gravação fica parada sobre a trilha aguardando a rotação do disco fazer com que o setor desejado passe sob ela.
b) o tempo de busca indica o tempo que a cabeça de leitura e gravação leva para se deslocar por todos os setores de um cilindro de um disco.
c) o tempo de acesso indica o tempo necessário para localizar uma trilha.
d) o seetle time indica o tempo que leva para ir de uma trilha para outra do disco.

Referências

ALECRIM, Emerson. **Conhecendo o disco rígido (HD)**. Disponível em: <http://www.infowester.com/hd.php>. Acesso em: 30 maio 2011.

Cache. Disponível em: <http://pt.wikipedia.org/wiki/Cache>. Acesso em: 30 maio 2011.

Como funciona a memória do computador. Disponível em: <http://informatica.hsw.uol.com.br/memoria-do computador1.htm>. Acesso em: 30 maio 2011.

Como funciona a memória ROM. Disponível em: <http://informatica.hsw.uol.com.br/memoria-rom.htm>.Acesso em: 30 maio 2011.

MORIMOTO, Carlos E. **Hardware: o guia definitivo**. São Paulo, Sul Editores, 2007.

MORIMOTO, Carlos E. **Memória RAM**. Disponível em: <http://www.hardware.com.br/guias/memoria-ram/ddr.html>. Acesso em: 30 maio 2011.

MORIMOTO, Carlos E. **Tudo sobre a memória RAM**, parte 1. Disponível em: <http://www.hardware.com.br/tutoriais/memoria-ram/>. Acesso em: 30 maio 2011.

MORIMOTO, Carlos E. **Tudo sobre os HDs, flash e armazenamento**. Disponível em: <http://www.hardware.com.br/guias/hds/como-hds-funcionam.html>. Acesso em: 30 maio 2011.

TORRES, Gabriel. **Tudo o Que Você Precisa Saber Sobre as Arquiteturas de Memória de Dois, Três e**

Quatro Canais. Disponível em:<http://www.clubedohardware.com.br/artigos/Tudo-o-Que-Voce-Precisa-Saber-Sobre-Memorias-Dual-Channel/673>. Acesso em: 30 maio 2011.

WEBER, Raul Fernando. **Arquitetura de computadores pessoais**. Porto Alegre: Sagra Luzzatto, 2004.

WEBER, Raul Fernando. **Fundamentos de arquitetura de computadores**. Porto Alegre: Bookman; UFRGS,

2008.

D

Dispositivos de Entrada e saída

Unidade D

Arquitetura e Organização de Computadores

UNIDADE D

DISPOSITIVOS DE ENTRADA E SAÍDA

De nada adianta todo o poder de processamento e a existência de uma estrutura para armazenar e conduzir informações a serem executadas em um computador se ele não tiver a capacidade de receber dados do usuário ou responder ao processamento executado. No seguinte vídeo http://www.youtube.com/watch?v=L7k06TT-8YA podemos identificar alguns dos elementos existentes para realizar a comunicação com o meio externo. Através dele podemos perceber que existem diversos dispositivos de entrada e saída de dados.

Os elementos utilizados para essa finalidade são conhecidos como dispositivos de I/O (Input/Output) ou dispositivos de E/S (Entrada/Saída).

Categorias de dispositivos de E/S

Existem duas categorias de dispositivos de E/S:

- Destinados à comunicação entre o usuário e o micro (externos): nela estão enquadrados os dispositivos voltados para a entrada de dados no computador, como o teclado, o mouse, o microfone, dentre outros, como também os destinados a realizar a saída de dados do sistema como o monitor, a impressora, a placa de som, etc.
- Destinados à comunicação entre o processador e componentes do micro (internos): nela estão contemplados os dispositivos que realizam a entrada e a saída de dados do processador como os controladores de discos, controladores de memória, controladores de barramento, etc.

Os dispositivos de E/S externos ao computador são os responsáveis por receber os dados de entrada dos usuários e, para isso, trabalham com uma baixa necessidade de largura de banda, mas com respostas imediatas às ações deles, como o mouse e o teclado. Por outro lado, os dispositivos que apresentam a saída dos dados processados para os usuários necessitam de largura de banda de saída de dados, como a placa de vídeo e a placa de som. Também existem os dispositivos que interagem, não com usuários, mas com outras máquinas, como a placa de rede, que tem a necessidade de largura de banda tanto para a entrada quanto para a saída de dados, apresentando um baixo tempo de resposta.

Endereços de IRQ

Os dispositivos de E/S podem gerar eventos em intervalos de tempo que o processador não pode prever, como, por exemplo, quando que o usuário irá pressionar uma tecla do teclado ou movimentar o cursor do mouse, bem como não é possível determinar quando será iniciada uma transferência de dados através de uma placa de rede, dentre outras situações.

Para que o processador possa executar suas atividades sem a necessidade de ficar monitorando quando que um determinado dispositivo irá realizar uma operação de E/S, são utilizados endereços de IRQ para gerar interrupções de hardware quando for necessário. Eles servem como canais através dos quais dispositivos podem chamar a atenção do processador para si, sendo que enquanto isso não ocorre o processador fica livre para realizar outras operações.

Dessa maneira, ao receber um sinal proveniente de um canal de IRQ, o processador interrompe o que está

fazendo e dá atenção ao dispositivo que o solicitou, sendo que cada endereço de IRQ funciona como se fosse uma campainha utilizada por um determinado dispositivo para chamar a atenção do processador para si.

No primórdio dos PCs, existiam 8 endereços de IRQ, sendo numerados da posição 0 até a 7, da seguinte forma:

IRQ 0 - Sinal de clock da placa-mãe
IRQ 1 - Teclado
IRQ 2 - Livre
IRQ 3 - COM 2
IRQ 4 - COM 1
IRQ 5 - Disco Rígido
IRQ 6 - Drive de disquetes
IRQ 7 - Porta paralela

Nessa organização apresentada, o número do endereço de IRQ ocupado por um dispositivo indica também a sua prioridade. Logo, o endereço de IRQ 0 possui prioridade sobre um pedido que venha do IRQ 1, bem como o endereço de IRQ 1 tem prioridade sobre o IRQ 2 e assim sucessivamente. O inconveniente dessa organização era a pouca quantidade de endereços disponíveis para que novos dispositivos pudessem ser integrados ao computador.

A forma encontrada para solucionar a falta de endereços IRQ foi manter o controlador de IRQs original, de forma a manter a compatibilidade com a estrutura antiga, e adicionar um segundo controlador com 8 novos endereços a serem utilizados. Este segundo controlador foi ligado no IRQ 2, que estava livre na estrutura original, resultando em um cascateamento de IRQs.

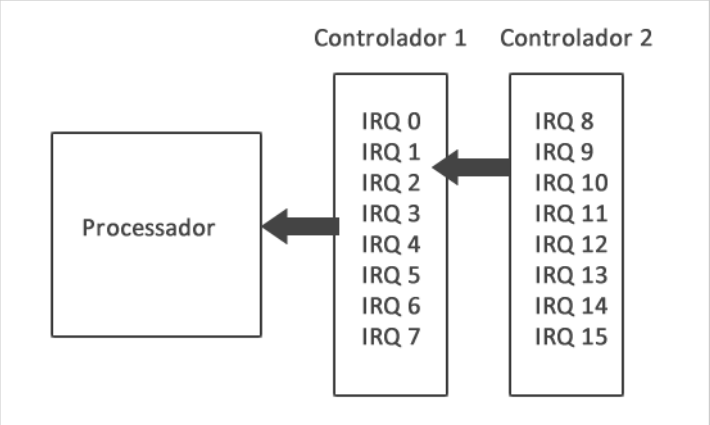


Figura D.1 - Estrutura dos endereços de IRQ
Fonte: http://www.hardware.com.br/tutoriais/irq-dma-io/

Nessa nova estrutura, a organização dos endereços de IRQ fica estabelecida da seguinte forma:

IRQ 0 - Sinal de clock da placa-mãe (fixo)
IRQ 1 - Teclado (fixo)
IRQ 2 - Cascateador de IRQs (fixo)
IRQ 3 - Porta serial 2
IRQ 4 - Porta serial 1
IRQ 5 - Livre
IRQ 6 - Drive de disquetes

IRQ 7 - Porta paralela (impressora)
IRQ 8 - Relógio do CMOS (fixo)
IRQ 9 - Placa de vídeo
IRQ 10 - Livre
IRQ 11 - Controlador USB
IRQ 12 - Porta PS/2
IRQ 13 - Coprocessador aritmético (fixo)
IRQ 14 - IDE Primária
IRQ 15 - IDE Secundária

Com o desenvolvimento dos barramentos do computador, juntamente com o barramento PCI surge o recurso de PCI Steering. Através dele, é possível que dois ou mais periféricos PCI compartilhem um mesmo endereço de IRQ. O controlador PCI passa a desempenhar um importante papel, que é o de estabelecer uma ligação entre os periféricos e o processador, sendo que é ele quem recebe todos os pedidos de interrupção dos dispositivos conectados ao barramento PCI e os encaminha para o processador e, ao receber as respostas, os encaminha de volta para o respectivo dispositivo. Com essa técnica se tornou possível utilizar menos endereços de IRQ para uma quantidade maior de dispositivos.

O controlador USB também se utiliza dessa mesma lógica de funcionamento, ocupando um único endereço IRQ e o compartilhando entre todos os dispositivos conectados às portas USB, sendo responsável por receber as solicitações, as repassar para o processador e direcionar os resultados novamente para cada dispositivo que realizou o pedido de interrupção.

Atualmente, os controladores de IRQ foram substituídos por um único controlador, de interrupção, mais aprimorado, denominado APIC (Advanced Programmable Interrupt Controller). O seu objetivo é o de melhorar a forma como se trabalha com interrupções em sistemas que possuam dois ou mais processadores. Além disso, foram adicionados novos endereços, sendo agora do IRQ 0 até o IRQ 23, preservando a compatibilidade com sistemas antigos e mantendo a possibilidade de que vários dispositivos compartilhem um mesmo endereço, o que resulta em uma maior quantidade de endereços livres para novos elementos.

Endereços de I/O

Além dos endereços destinados a que os dispositivos possam solicitar interrupções ao processador, conforme vimos, também existem os endereços utilizados para que os dispositivos possam comunicar-se entre si. Nesse sistema, cada dispositivo possui um endereço diferente e próprio, onde através do qual é possível acionar um determinado dispositivo do sistema.

Ao todo existem nos computadores 65.536 endereços de I/O, garantindo que cada dispositivo conectado no sistema possua o seu e que através disso possa ser localizado e se comunique com os demais componentes.

Barramentos de E/S

Os barramentos desempenham uma função importante no sistema de E/S, pois são através deles que os mais variados dispositivos são conectados ao sistema computacional, cabendo a cada barramento que o compõe integrar o sistema.

Eles permitem que ocorra uma abstração do processador em relação aos dispositivos de E/S existentes,

ou seja, o processador não precisa lidar diretamente com os dispositivos, pois quem trata diretamente com eles é o barramento no qual eles estão conectados, intermediando esta relação. Dessa forma, cada barramento fica responsável por lidar com os dispositivos ligados a ele.

Nessa estrutura, qualquer dispositivo que seja compatível com um dos barramentos de E/S do computador pode ser integrado ao sistema computacional, sem a necessidade de conhecer o funcionamento dos demais componentes.

O barramento é o responsável então por especificar como que os dados e comandos serão transferidos entre o processador e o dispositivo. Cabe a ele também determinar como que os diversos dispositivos competirão pelo próprio barramento, definindo protocolos a serem seguidos, assim como prioridades entre solicitações e demais elementos a serem seguidos para um correto funcionamento da comunicação entre os diferentes componentes da máquina.

Estrutura de E/S

Os computadores utilizam uma estrutura de E/S baseada na existência de barramentos, que têm como finalidade interligar o processador, a memória e os dispositivos de E/S.

Cada dispositivo conectado a essa estrutura é dividido em duas partes: o controlador (componentes eletrônicos presentes nos barramentos) e o próprio dispositivo. O controlador, como o próprio nome sugere, é o responsável por controlar o dispositivo de E/S, que está ligado a ele, e de gerenciar os acessos dele ao barramento.

Um controlador consegue ler e escrever blocos de dados na memória principal sem a necessidade da intervenção do processador, realizando acessos diretos à memória, o que agiliza a transferência das informações.

Para que essa estrutura possa ser utilizada, existe um árbitro do barramento, que é o responsável por definir quem terá o direito de usar o barramento quando mais de um elemento tentar utilizá-lo ao mesmo tempo, resolvendo possíveis conflitos. Cada barramento também possui um protocolo que define as regras de funcionamento dele, ou seja, o que o dispositivo deve fazer para poder utilizá-lo, permitindo assim que componentes projetados por diferentes fabricantes possam ser conectados ao sistema, desde que eles sejam produzidos, levando em consideração esses protocolos.

Por sua vez, os dispositivos que se utilizam dos barramentos se dividem em dois tipos quanto à possibilidade de uso:

- Dispositivos mestres: aqueles que podem iniciar transferências pelo barramento;
- Dispositivos escravos: aqueles que apenas aguardam uma requisição pelo barramento.

DMA

O acesso à memória principal geralmente ocorre através do processador. Esse processo centralizado na figura do processador acarretaria uma grande lentidão no processamento de informações, visto que todo e qualquer acesso dependeria de passar por um único elemento.

Visando agilizar esse acesso, existe um circuito de apoio chamado de controlador de DMA (Direct Memory Access – Acesso Direto à Memória). Através dele é possível a transferência de dados entre um determinado dispositivo e a memória principal, sem a necessidade de que a CPU participe disso.

Nessa situação, o processador apenas solicita a transferência, a partir de uma interrupção solicitada por um dispositivo, para o controlador de DMA, que passa a ser responsável por realizar a transferência. O processador então fica liberado para realizar suas demais atividades, enquanto o controlador coordena todo o processo, como, por exemplo, na gravação de um DVD, o controlador coordena a transferência dos arquivos a serem gravados do disco rígido para a memória principal e desta para a unidade de DVD. Terminada a transferência, o controlador indica para o processador, através de uma interrupção, o final da operação, devolvendo o controle para ele.

Resumo

Ao final da presente unidade nós vimos que:

- Os dispositivos de E/S fazem, tanto externamente quanto internamente no computador, a entrada de dados a serem processados pelo processador em algum momento, assim também realizam a saída dos dados processados.
- Os endereços de IRQ permitem que os dispositivos de E/S realizem suas operações e, enquanto isso, o processador fica livre para realizar outras tarefas. Quando for necessária a atenção do processador, o dispositivo gera um pedido de interrupção através de seu endereço de IRQ.
- É possível compartilhar um mesmo endereço de IRQ entre vários dispositivos, cabendo ao controlador do barramento compartilhar o endereço entre eles e intermediar a comunicação deles com o processador.
- Cada dispositivo possui um endereço de I/O único, que é diferente do endereço de E/S, que é utilizado para que o dispositivo possa ser acessado diretamente.
- Os barramentos são os responsáveis por realizar toda a comunicação entre dispositivos e o restante do computador.
- Cada barramento pode acessar diretamente a memória quando da operação de um dispositivo, devido ao DMA que permite que isso ocorra, enquanto isso libera o processador para realizar outras tarefas.

Questões de revisão

a) Qual é a função dos dispositivos de E/S no computador? Que categorias existem?
b) Como funcionam os endereços de IRQ?
c) Como funcionam os controladores PCI, USB e APIC?
d) Para que servem os endereços de I/O?
e) Qual é a função de um barramento e como está estruturada a E/S no computador?
f) O que é DMA?

Atividades

1. Sobre os dispositivos de entrada e saída, é incorreto afirmar que os dispositivos

a) que apresentam saída de dados para os usuários necessitam de largura de banda.
b) externos realizam a comunicação do computador com o usuário.
c) internos fazem a comunicação do processador com demais componentes do micro.
d) que recebem dados dos usuários não necessitam de respostas imediatas.

2. Sobre a estrutura de entrada e saída, é correto afirmar que

a) ela é composta basicamente por um barramento que interliga o processador, a memória e os dispositivos de E/S.
b) o árbitro do barramento é utilizado para controlar o acesso aos dispositivos de E/S.
c) o protocolo da barramento é utilizado para gerenciar conflitos na utilização do barramento.
d) cada dispositivo comunica-se diretamente com os outros dispositivos e com o barramento.

3. Em relação aos endereços de IRQ, é correto afirmar que

a) existe um endereço para cada dispositivo existente no computador.
b) eles servem para que o processador possa controlar a operação de dispositivos, interrompendo o funcionamento deles quando necessário.
c) é possível fazer com que um controlador de barramento compartilhe um único endereço entre mais de um dispositivo.
d) o APIC funciona compartilhando 8 endereços de IRQ entre todos os dispositivos do computador.

4. Sobre entrada e saída é correto afirmar que

a) o único papel desempenhado pelos dispositivos de E/S é o de fazer a comunicação entre usuário e micro.
b) todo acesso a um dispositivo é gerenciado através de seu controlador.
c) os endereços de I/O podem ser compartilhados entre diversos dispositivos de E/S.
d) o DMA consiste em supervisionar a transferência de dados para dispositivos sob o controle da CPU.

E

Tipos de organização de computadores

Unidade E
Arquitetura e Organização de Computadores

UNIDADE E

TIPOS DE ORGANIZAÇÃO DE COMPUTADORES

Introdução:

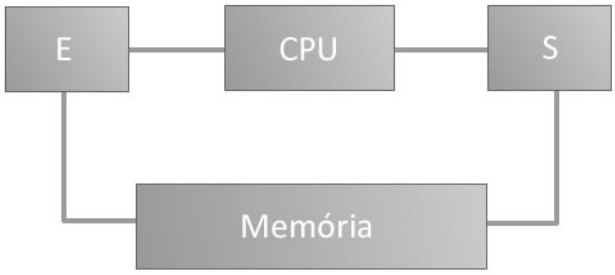
Na estrutura de computadores estudada até então, observamos os elementos existentes em uma arquitetura voltada para

Em um dado momento surgiu a necessidade de maior capacidade de processamento, surgindo então técnicas baseadas em concorrência. Nesse sentido, a utilização de diversos processadores em conjunto passou a ser empregada como uma forma de obter este ganho de desempenho e, com isso, surge o processamento paralelo, envolvendo técnicas relacionadas a esse tipo de implementação.

Com o avanço das técnicas de processamento paralelo foi aberto espaço para o surgimento de máquinas de grande capacidade de processamento, os chamados supercomputadores, amplamente utilizados em áreas de pesquisa que requerem grande processamento. No link a seguir é possível acompanhar uma reportagem referente ao supercomputador brasileiro netuno, onde podemos compreender melhor a utilidade do mesmo: <http://www.youtube.com/watch?v=dD0u7O1qDkY>.

Arquiteturas paralelas

Tradicionalmente a arquitetura dos computadores encontra-se formada da seguinte forma:



Nessa arquitetura existe uma unidade central de processamento e um sistema de entrada e de saída de dados que se relaciona com ela e diretamente com o sistema de memória (DMA). A memória oferece suporte a multiprogramação, armazenando vários programas a serem executados pela CPU, gerando um fluxo de instruções e um de dados.

Como forma de agilizar o processamento dos dados foram construídas arquiteturas com múltiplas CPU's, sendo que existem várias formas de classificar essas arquiteturas, conforme veremos a seguir.

Classificação de Flynn

Para Flynn, uma forma de classificar arquiteturas paralelas é conforme elas se relacionam com o fluxo de instruções e com o fluxo de dados. Esses fluxos podem ser múltiplos ou simples e, com base nisso, propôs uma classificação em quatro classes:

	SD (Single Data)	MD (Multiple Data)
SI(Single Instruction)	SISD Maquinas von Neumann convencionais	SIMD Máquinas Array (CM-2, MasPar)
MI (Multiple Instruction)	MISD Sem representante (até agora)	MIMD Multiprocessadores e multicomputadores (nCUBE, intel Paragon, Cray T3D)

Tabela - Classificação de Flynn (Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008)

Cada uma dessas quatro classes representa uma forma de se lidar com o processamento de instruções e de dados, conforme estudaremos a seguir.

SISD (Single Instruction Single Data)

Nessa classificação, um único fluxo de instruções trabalha sobre um único fluxo de dados. Nela, existe um único fluxo de instruções alimentando uma única unidade de controle (C) que coordena um único processador (P), que atua sobre um único fluxo de dados que é lido, processado e reescrito em uma única memória (M). Seu funcionamento é característico de máquinas compostas por um processador.

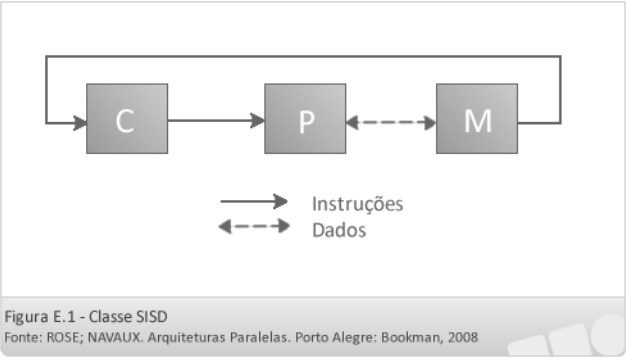


Figura E.1 - Classe SISD
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

MISD (Multiple Instruction Single Data)

Nessa classificação, existem múltiplos fluxos de instruções e um único fluxo de dados. Ela envolve múltiplos processadores (P) com sua própria unidade de controle (C) executando diferentes instruções sobre um único conjunto de dados. De forma prática, não é possível implementá-la, sendo uma classe apenas teórica.

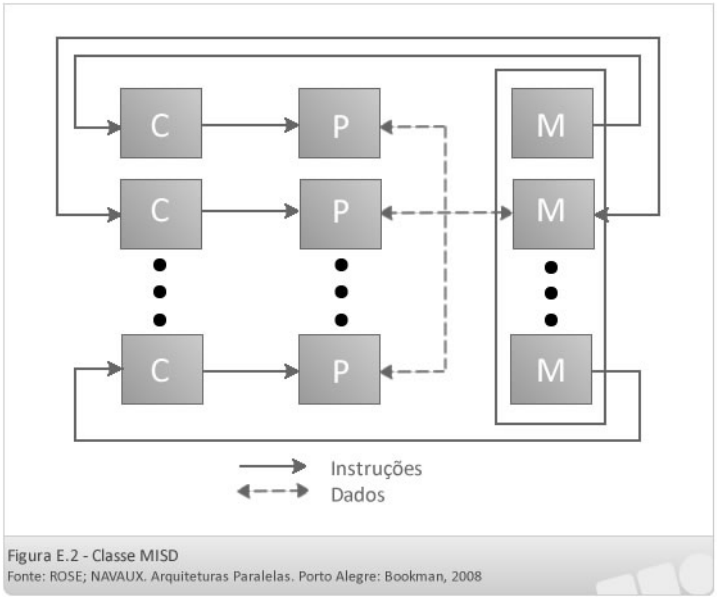


Figura E.2 - Classe MISD
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

SIMD (Single Instruction Multiple Data)

Nessa classificação, existe um único fluxo de instruções que é executado sobre múltiplos fluxos de dados. Nela, existe uma única unidade de controle (C) que comanda a execução de uma única instrução por diversos processadores (P). Cada processador executa uma parte da instrução de forma paralela (ao mesmo tempo), trabalhando sincronamente sobre diferentes fluxos de dados. Para que esse funcionamento seja possível é necessário que a memória (M) seja implementada em diversos módulos de memória, que possibilitem diversos acessos simultâneos a ela por parte dos diversos processadores.

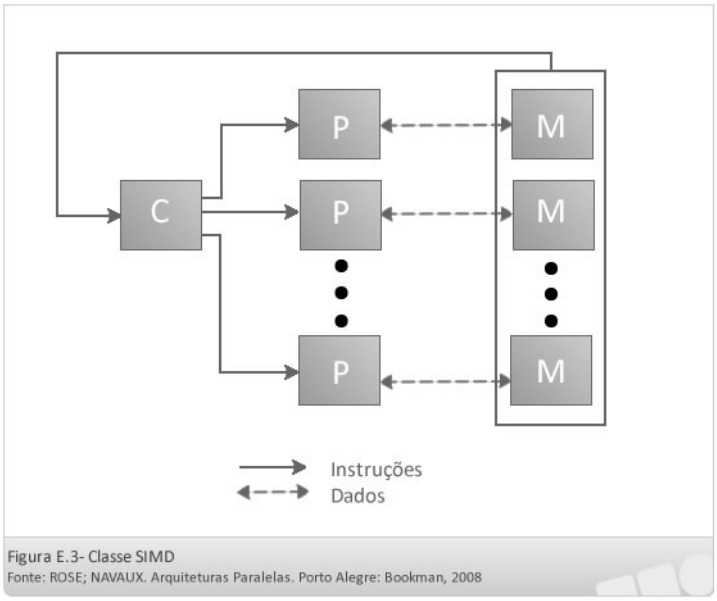


Figura E.3- Classe SIMD
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

MIMD (Multiple Instruction Multiple Data)

Nessa classificação, existem múltiplos fluxos de instruções e múltiplos fluxos de dados. Nela, cada unidade de controle (C) recebe um fluxo instruções diferente e o encaminha para o processador (P) que está sob seu controle. Dessa forma, cada um dos processadores trabalha, assincronamente, sobre instruções diferentes, sendo que cada uma delas com seus próprios dados. O módulo de memória (M) deve ser implementado em diversos módulos de memória, que possibilitem diversos acessos simultâneos a ela

por parte dos diversos processadores.

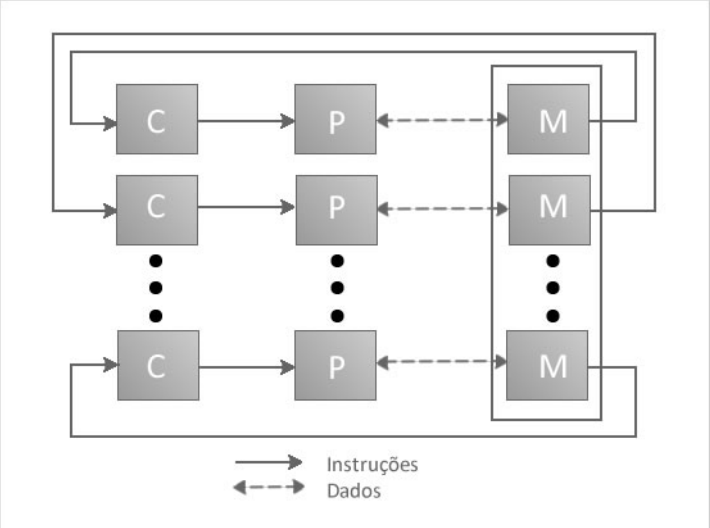


Figura E.4 - Classe MIMD
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

Com base ainda nessa classificação MIMD, ela pode se subdividir quanto a forma como a memória é acessada. Quando a memória é acessada de forma compartilhada pelos diversos processadores, a máquina é conhecida como de multiprocessadores, ou seja, uma única máquina composta por vários processadores que compartilham o mesmo espaço de endereçamento. Quando a memória não é compartilhada entre os processadores, a máquina é conhecida como de multicomputadores, sendo que, existem diferentes máquinas, com cada uma delas possuindo sua própria memória e processadores, que devem se comunicar através de trocas de mensagens para manter o correto funcionamento de todo o sistema computacional.

Classificação segundo o compartilhamento de memória

Seguindo a classificação apresentada no MIMD, de onde temos os multiprocessadores e os multicomputadores, a forma como a memória é acessada constituí-se em um elemento importante, pois determinará como que ela será compartilhada entre os diversos processadores.

Esse compartilhamento segue a seguinte classificação:

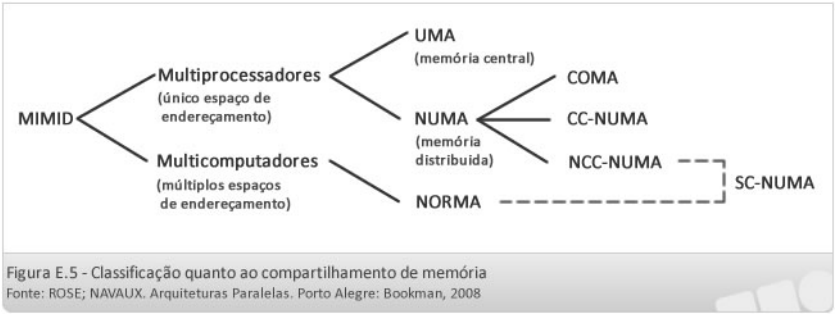


Figura E.5 - Classificação quanto ao compartilhamento de memória
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

Quanto às máquinas de multiprocessadores pode-se trabalhar com as seguintes formas de se acessar a memória:

Máquina UMA (Uniform memory access)

Nesse tipo de máquina a memória é centralizada, ficando a uma distância igual de todos os processadores que compõem o sistema, garantindo desta forma que o acesso a ela leve o mesmo tempo para cada um

deles. Como ela é centralizada, é utilizado apenas um barramento ligando-a a todos os barramentos, sendo assim, só pode ser realizado um acesso por vez à memória (quando ela está sendo acessada por um processador todos os outros devem aguardar sua vez).

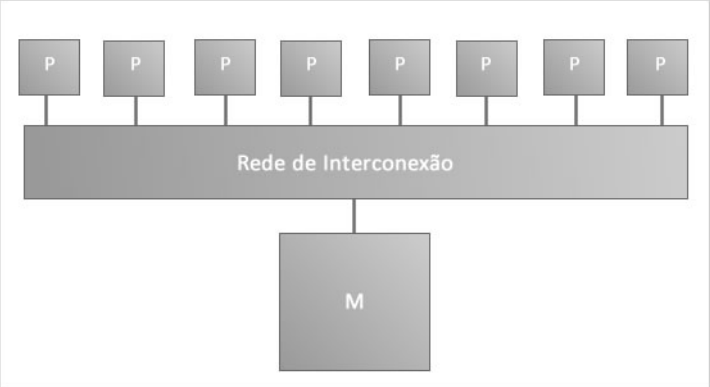


Figura E.6 - Multiprocessador UMA
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

Um fator que deve ser considerado em sua implementação é o de que cada processador possui sua própria cache, sendo que a informação que está na cache de um determinado processador pode ser alterada na cache de outro e armazenada novamente na memória, o que pode resultar em que o conteúdo da cache de um processador acabe ficando desatualizada. A possibilidade de isto ocorrer torna importante a existência de uma coerência entre as caches de cada processador, para que estejam sempre atualizadas. Esse problema costuma ser resolvido via hardware específico.

Máquina NUMA (non-uniform memory access)

Nesse tipo de máquina a memória é distribuída, sendo composta por vários módulos, cada um deles pertencendo a um processador específico. Nessa organização cada processador leva um tempo menor para acessar sua própria memória e tempos mais demorados para acessar a memória pertencente a outros processadores, resultando em acessos não-uniformes à memória, pois dependerá de qual memória deverá ser acessada, (se a do próprio processador ou de algum outro) o que acarretará em tempos diferentes de acesso.

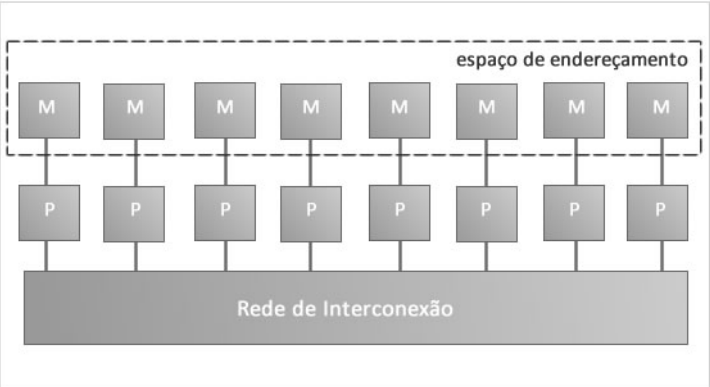


Figura E.7 - Multiprocessador NUMA
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

De acordo com a forma como é trabalhada a questão da coerência de cache a máquina NUMA pode ser subdividida em:

- NCC-NUMA (non-cache-coherent non-uniform memory access): onde não existe preocupação com a coerência de cache em hardware;

- CC-NUMA (cache-coherent non-uniform memory access): existe a garantia de coerência de cache através de hardware;
- SC-NUMA (software-coherent non-uniform memory access): nela existe a garantia de coerência de cache através de software.

Máquina COMA (cache-only memory architecture)

Nesse tipo de máquina as memórias locais são formadas na verdade por memórias cache de maior capacidade que as convencionais, possuindo hardware de replicação que permite a transferência de conteúdo entre as caches.

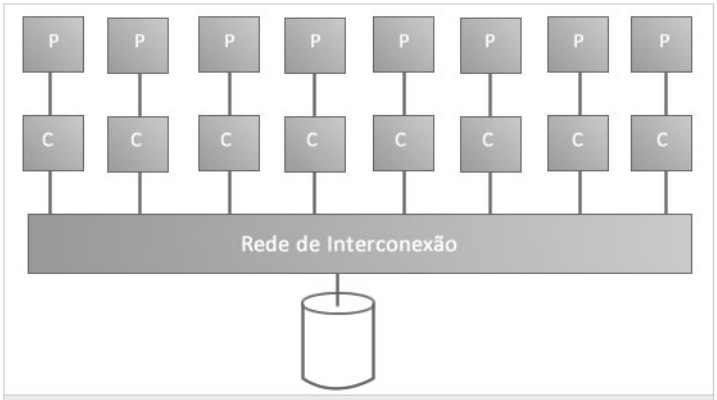


Figura E.8 - Multiprocessador COMA
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

O outro tipo de máquinas existente são as compostas por multicomputadores. Essas máquinas, quanto ao acesso à memória, podem ser classificadas como:

Máquina NORMA (non-remote memory access)

Nesse caso, como cada máquina possui uma arquitetura completa, cada processador somente acessa a área de endereçamento local de sua própria máquina.

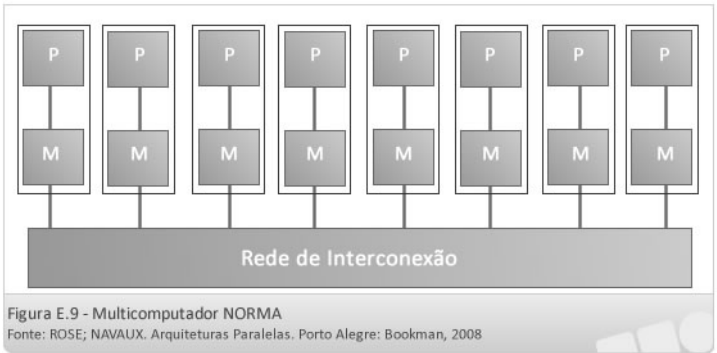


Figura E.9 - Multicomputador NORMA
Fonte: ROSE; NAVAUX. Arquiteturas Paralelas. Porto Alegre: Bookman, 2008

Medida de Desempenho de Máquinas Paralelas

Existem diversas formas de se medir o desempenho de processadores, conforme abordado no módulo B, como por exemplo o MIPS, que indica quantas milhões de instruções são executadas em um segundo. Para a medição do desempenho de máquinas paralelas é amplamente utilizada a medida FLOPS, mais especificamente o MFLOPS, que indica milhões de operações com ponto flutuante executadas por segundo.

Como forma de padronizar a medição de desempenho de diferentes máquinas paralelas são utilizados programas-padrão que são executados nas máquinas a serem comparadas, resultando na técnica de Benchmarking. Ela envolve então programas de teste que são utilizados para medir o desempenho de computadores, sendo que um dos benchmarks mais utilizados é o LINPACK, dentre outros.

Resumo

Ao final da presente unidade nós vimos que:

- As arquiteturas tradicionais dos computadores são baseadas na existência de uma única unidade de processamento e uma memória principal;
- As arquiteturas paralelas se baseiam na existência de mais de um processador, que podem ser classificadas de diversas formas;
- Uma das classificações de máquinas paralelas é a de Flynn, que se baseia na relação existente com o fluxo de instruções e com o fluxo de dados resultando em quatro classes: SISD, MISD, SIMD e MIMD;
- Outra classificação de máquinas paralelas se baseia na forma como a memória é compartilhada entre os diversos processadores, sejam multiprocessadores, quanto multicomputadores;
- De acordo com essa segunda classificação as máquinas de multiprocessadores se dividem em UMA e NUMA (COMA, CC-NUMA, NCC-NUMA, SC-NUMA) enquanto as máquinas de multicomputadores seguem a classificação NORMA;
- Como forma de medir o desempenho de máquinas paralelas são utilizados benchmarks que medem, com base em programas-padrão a quantidade de MFLOPS que cada máquina testada atinge.

Questões de revisão

a) Em que se baseia a classificação de Flynn para determinar as classes de máquinas paralelas?
b) Diferencie as classes de Flynn SISD de MISD.
c) Diferencie as classes de Flynn SIMD e MIMD.
d) Em que se baseia a classificação segundo o compartilhamento de memória?
e) Quais máquinas são as classificações de máquinas existentes segundo o compartilhamento de memória? Como cada um deles funciona?
f) Como é realizada a medição de desempenho em máquinas paralelas?

Atividade

1. Em relação à classificação de Flynn, é correto afirmar que

a) ela refere-se a máquinas que possuem um processador e como ele se relaciona com a memória do sistema.
b) classifica os computadores de acordo com o número de processadores que eles possuem.
c) classifica as arquiteturas paralelas de acordo com o fluxo de instruções e fluxo de dados.
d) ela considera apenas múltiplos fluxos de dados.

2. Sobre as classes de Flynn, é incorreto afirmar que

a) na classe SISD existe um único fluxo de instruções com múltiplos fluxos de dados sobre um único processador.
b) na classe SIMD existe um único fluxo de instruções atuando sobre um múltiplo fluxo de dados executados por um único processador.
c) na classe MIMD os processadores que compõem o sistema trabalham de forma assíncrona sobre instruções diferentes.
d) na classe MISD existem múltiplos fluxos de instruções que são executadas por vários processadores coordenados por uma única unidade de controle.

3. Sobre a classificação segundo o compartilhamento de memória, é correto afirmar que

a) ela é baseada em uma arquitetura composta por um fluxo simples de instruções.
b) ela parte da classe MIMD, de onde temos multiprocessadores e multicomputadores e na forma como eles compartilham a memória.
c) não considera o fluxo de instruções, mas sim a quantidade de processadores.
d) n.d.a.

4. Sobre as classificações de máquinas paralelas, segundo o compartilhamento de memória, é correto afirma que

a) na máquina UMA a memória é centralizada, ficando a uma mesma distância de cada processador do sistema e podendo ser acessada por todos ao mesmo tempo.
b) na máquina NUMA a memória é distribuída, com cada um de seus módulos pertencendo a um processador específico.
c) na máquina COMA a memória é composta por memória cache centralizada, servindo todos os processadores do sistema.
d) na máquina NORMA todos os processadores do sistema conseguem acessar as memórias dos outros processadores.

5. Em relação à medida de desempenho de máquina paralelas, é incorreto afirmar que

a) é utilizada como referência a medida MFLOPS.
b) é baseada na execução de programas-padrão nas máquinas a serem comparadas.
c) o LINPACK é um dos benchmarks utilizados para medição de desempenho.
d) é baseada na medição MIPS.

Atividade no fórum

Com base em tudo que foi estudado na unidade E referente aos tipos de organização de computadores, pesquise sobre os assuntos a seguir e poste no fórum suas impressões a respeito deles até o final da semana como parte da avaliação da etapa:

- Acessar o site www.top500.org. Qual é a finalidade dele? Qual é a lista mais atual? Quais são os dez supercomputadores mais potentes do mundo?
- Escolher um dos dez computadores mais potentes e por que ele está nesta localização destacar sua localização, em que é utilizado, quais as características de sua arquitetura. (Caso um dos supercomputadores já tenha sido postado no fórum por algum colega, escolher outro da lista)
- Algum computador brasileiro faz parte da lista? Se sim, Quais? Que posição ocupam?