

UNIVERSIDADE FEDERAL DE ALFENAS  
Curso de Ciência da Computação - Instituto de Ciências Exatas  
Disciplina: Arquitetura de Computadores I - Prova 01 (Parte A)

Instruções para a realização desta prova

- a) Esta prova deverá ser realizada em tripla
  - b) O conteúdo entregue deverá ser um arquivo em formato PDF
  - c) Você deverá organizar suas soluções de acordo com a sequência dos exercícios selecionados
  - d) Você deverá inserir o enunciado da questão e inserir a solução imediatamente abaixo. Insira as soluções abaixo do enunciado de cada sub-item de cada questão.  
Insira o enunciado da questão mesmo se você não for apresentar a solução
  - e) Apresente o desenvolvimento das soluções e as respostas de cada questão
  - f) O arquivo entregue deverá ser nomeado da seguinte forma:
    - d.1) arql-P1-parteA\_<número matrícula 1>\_<número matrícula 2>\_<número matrícula 3>.pdf
    - d.2) em que <número matrícula x> é o número de matrícula do estudante x
    - d.3) Exemplo: arql-P1-parteA\_2029.1.04.049\_2029.1.04.015\_2029.1.04.009.pdf
  - g) Provas não formatadas e sem a devida organização das soluções não serão consideradas
  - h) Edite este pdf, inclua os nomes, matrículas e anexe páginas com enunciados e soluções das questões
- Boa Prova

Livro texto:  
Arquitetura e Organização de Computadores  
William Stallings 8ª Edição, Editora Pearson



Capítulo 01 Questões de revisão: Todas

Capítulo 02 Questões de revisão: Todas e Problemas: 2.1 2.2 2.3 2.5 2.6 2.7 2.8 2.9

2.10 Capítulo 03 Questões de revisão: Todas e Problemas: 3.1 3.2 3.3 3.7

Capítulo 04 Questões de revisão: Todas e Problemas: 4.1 4.2 4.3 4.4 4.5 4.6 4.8 4.10

4.12 Capítulo 05 Questões de revisão: Todas e Problemas: 5.1 5.11 5.12 5.13

Bom Trabalho a Todos!  
Professor Eliseu César Miguel

Esta avaliação foi elaborada utilizando-se  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

## **PERGUNTAS DE REVISÃO CAP 1**

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

**R:** Em termos gerais, arquitetura de computadores é referente aos atributos que possuem um impacto direto sobre a execução lógica do computador (exemplo: mecanismos de E/S e técnicas para endereçamento de memória), enquanto a organização refere-se às unidades operacionais e suas interconexões que realizam as especificações arquiteturais (exemplo: interfaces entre o computador e periféricos e a tecnologia de memória utilizada).

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

**R:** Em termos gerais, a função de um computador se refere à operação individual de cada componente como parte da estrutura (existem apenas quatro: processamento, armazenamento e movimentação de dados e controle). Já a estrutura de um computador é referente ao modo como os componentes são inter-relacionados.

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

**R:** Processamento de dados, armazenamento de dados, movimentação de dados e controle.

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

**R:** 1 - CPU: controla a operação do computador e realiza suas funções de processamento de dados, conhecida normalmente por processador. 2 - Memória Principal: armazena os dados. 3 - E/S: move dados entre computador e seu ambiente externo. 4 - Interconexão do Sistema: algum mecanismo que oferece comunicação entre CPU, memória e E/S, pode-se citar como exemplo o barramento do sistema.

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

**R:** 1 - Unidade de Controle: controla a operação da CPU e, portanto, do computador. 2 - Unidade Aritmética e Lógica (ALU): realiza as funções de processamento de dados do computador. 3 - Registradores: oferece armazenamento interno à CPU. 4 - Interconexão da CPU: algum mecanismo que oferece comunicação entre Unidade de controle, ALU e registradores.

## **PERGUNTAS DE REVISÃO CAP 2**

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

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

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

**R:** Memória principal, Unidade Lógica e Aritmética, Unidade de Controle e Dispositivos de E/S.

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

**R:** Portas, células de memória e interconexões entre ambas.

**2.4 Explique a Lei de Moore.**

**R:** Gordon Earle Moore observou que o número de transistores que podem ser colocados em um único chip foi dobrando a cada ano e profetizou que a quantidade de transistores que poderiam ser colocados em uma mesma área continuaria dobrando no mesmo ritmo a cada ano, mantendo-se o mesmo custo de fabricação.

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

**R:** 1 - Conjunto de instruções semelhantes ou idênticas: o mesmo conjunto de instruções de uma máquina pode ser compatível com os outros membros da família, tornando possível assim a execução de um programa em outra máquina. 2 - Sistema Operacional semelhantes ou idênticos: o mesmo sistema operacional básico está disponível para toda a família. 3 - Velocidade aumentada: a taxa de execução de instruções aumenta dos membros da família mais baixos aos mais elevados. 4 - Maior número de portas de E/S: o número de portas de E/S aumenta, dos membros mais baixos aos mais altos da família. 5 - Tamanho de memória crescente: o tamanho da memória principal aumenta, dos membros mais baixos aos mais altos da família. 6 - Maior custo: o custo de um sistema aumenta dos membros mais baixos aos mais altos da família.

**2.6 Qual é a principal característica que distingue um microprocessador?**

**R:** Em um microprocessador, todos os componentes da CPU estão localizados em um único chip.

## **PROBLEMAS CAP 2**

**2.1 Considere que  $A = A(1), A(2), \dots, A(1000)$  e  $B = B(1), B(2), \dots, B(1000)$  sejam dois vetores (arrays unidimensionais) compostos de 1000 números em cada um, que são somados para formar um array C tal que  $C(I) = A(I) + B(I)$  para  $I = 1, 2, \dots, 1000$ . Usando o conjunto de instruções do IAS, escreva um programa para esse problema. Ignore o fato de que o IAS foi projetado para ter apenas 1000 palavras de armazenamento.**

2.1 INSTRUÇÃO

0	LOAD M(2000)	ADD M(2000)
1	STOR M(4000)	LOAD M(8)
2	M SOB (9)	JUMP + M(3, 20:39)
3	JUMP M(3, 00:19)	<del>ADD M(8)</del> STOR M(8)
4	ADD M(9)	ADD M(10)
5	STOR M(0, 8:19)	ADD M(10)
6	STOR M(0, 23:39)	ADD M(10)
7	STOR M(1, 8:19)	JUMP M(0, 00:19)
8	999	
9	1	
10	1000	

R:

## 2.2

a) No IAS, como ficaria a instrução de código de máquina para carregar o conteúdo do endereço de memória 2?

2.2 a)

opcode	operando
00000001	0000000010

R:

b) Quantas viagens à memória a CPU precisa fazer para completar essa instrução durante o ciclo de instrução?

R: Duas viagens.

2.3 No IAS, descreva em português o processo que a CPU precisa assegurar para ler um valor da memória e escrever um valor na memória em termos do que é colocado em MAR, MBR, barramento de endereço, barramento de dados e barramento de controle.

R: Para ler um valor da memória, a CPU carrega o endereço do valor que ela quer no MAR, então ativa a linha de controle de leitura de memória e o coloca o endereço no barramento de endereços. A memória coloca o conteúdo da localização passando-o para o barramento de dados. Este dado é então transferido para o MBR. Para escrever um valor na memória, a CPU também coloca o endereço no MAR e no MBR, então afirma na linha de controle de gravação para a memória e coloca o endereço no barramento de endereços e os dados no barramento de dados. Transfere os dados da memória para o barramento de dados para o correspondente local de memória.

2.4 Dado o conteúdo de memória do computador IAS, mostrado a seguir,

Endereço	Conteúdo
08A	010FA210FB
08B	010FA0F08D
08C	020FA210FB

mostre o código em linguagem de montagem para o programa, começando no endereço 08A. Explique o que esse programa faz.

R: Este programa armazena o valor absoluto do conteúdo da posição de memória 0FA na memória local 0FB.

08A	LOAD M(0FA)
	STOR M(0FB)
08B	LOAD M(0FA)
	JUMP+M(08D)
08C	LOAD-M(0FA)
	STOR M(0FB)
08D	

2.5 Na Figura 2.3, indique a largura, em bits, de cada caminho de dados (por exemplo, entre AC e ALU).

R: Os caminhos de dados da MP para o MBR são de 40 bits, os caminhos de dados que chegam ao MAR são de 12 bits. Todos os caminhos que vão para AC são 40 bits e o mesmo para o MQ.

2.6 No IBM 360 Modelos 65 e 75, os endereços são espalhados em duas unidades separadas da memória principal (por exemplo, todas as palavras de número par em uma unidade e todas as palavras de número ímpar em outra). Qual poderia ser a finalidade dessa técnica?

R: Aumentar o desempenho.

2.7 Com referência à Tabela 2.4, vemos que o desempenho relativo do IBM 360 Modelo 75 é 50 vezes o do 360 Modelo 30, embora o tempo de ciclo de instrução seja apenas 5 vezes mais rápido. Como você explica essa discrepância?

R: A discrepância pode ser explicada pelo de outros componentes do sistema, além da velocidade do relógio fazer uma grande diferença na velocidade do clock faz uma grande diferença na velocidade geral do sistema.

**2.8 Enquanto analisa a loja de computadores de Billy Bob, você escuta um cliente perguntando a ele qual é o computador mais rápido na loja que ele possa comprar. Billy Bob responde: “Você está olhando para nossos Macintoshes. O Mac mais rápido que temos trabalha com uma velocidade de clock de 1,2 gigahertz. Se você realmente quer a máquina mais rápida, então deve comprar nosso Intel Pentium IV de 2,4 gigahertz em vez disso”. Billy Bob está certo? O que você diria para ajudar esse cliente?**

**2.9 O ENIAC era uma máquina decimal, onde um registrador era representado por um anel de 10 válvulas. A qualquer momento, somente uma válvula estava no estado On, representando um dos 10 dígitos. Supondo que o ENIAC tivesse a capacidade de ter várias válvulas no estado ON e OFF simultaneamente, por que essa representação é “esbanjadora” e que faixa de valores inteiros poderíamos representar usando 10 válvulas?**

**R:** Com dez dígitos poderíamos representar  $2^{10} = 1024$ . Para inteiros, podemos usar para representar de 0 a 1023.

**2.10 Um programa de benchmark é executado em um processador a 40 MHz. O programa executado consiste em 100.000 execuções de instrução, com a seguinte mistura de instruções e quantidade de ciclos de clock:**

Tipo de instrução	Quantidade de instruções	Ciclos por instrução
Aritmética de inteiros	45 000	1
Transferência de dados	32 000	2
Ponto flutuante	15 000	2
Transferência de controle	8 000	2

**Determine o CPI efetivo, a taxa de MIPS e o tempo de execução para esse programa.**

**R:** CPI efetivo = 1,55. Taxa de MIPS = 25,8. Tempo de Execução = 3,87 ns.

**2.11 Considere duas máquinas diferentes, com dois conjuntos de instruções diferentes, ambos tendo uma taxa de clock de 200MHz. As medições a seguir são registradas nas duas máquinas rodando determinado conjunto de programas de benchmark:**

Tipo de instrução	Quantidade de instruções (milhões)	Ciclos por instrução
Máquina A		
Aritmética e lógica	8	1
Load e store	4	3
Desvio	2	4
Outros	4	3
Máquina B		
Aritmética e lógica	10	1
Load e store	8	2
Desvio	2	4
Outros	4	3

a) Determine o CPI efetivo, a taxa MIPS e o tempo de execução para cada máquina.

2.11) a)  $CPI_A = \frac{(8 \times 1 + 4 \times 3 + 2 \times 4 + 4 \times 3) \times 10^6}{(8 + 4 + 2 + 4) \times 10^6} \approx 2,22$

$MIPS_A = \frac{200 \times 10^6}{2,22 \times 10^6} = 90$

$TE_A = \frac{18 \times 10^6 \times 2,22}{200 \times 10^6} = 0,23s$

$CPI_B = 1,92$        $TE_B = 0,23s$

$MIPS_B = 104$

R:

b) Comente os resultados.

2.13 Quatro programas de benchmark são executados em três computadores com os seguintes resultados:

	Computador A	Computador B	Computador C
Programa 1	1	10	20
Programa 2	1 000	100	20
Programa 3	500	1 000	50
Programa 4	100	800	100

A tabela mostra o tempo de execução em segundos, com 100 000 000 instruções executadas em cada um dos quatro programas. Calcule os valores de MIPS para cada computador para cada programa. Depois, calcule as médias aritmética e harmônica considerando pesos iguais para os quatro programas, e classifique os computadores com base na média aritmética e a média harmônica.

**R:** No computador A, o programa 1 terá valor MIPS = 100, o programa 2 MIPS = 0,1, o programa 3 MIPS = 0,2 e no programa 4 MIPS = 2. No computador B, programa 1 MIPS = 10, programa 2 = 1, programa 3 = 0,1, programa 4 = 0,125. Já no computador C, o programa 1 apresenta taxa MIPS = 5, igual ao programa 2 que apresenta a mesma taxa, o programa 3 apresenta MIPS = 2 e o programa 4 MIPS = 1.

**2.16 Considere o exemplo na Seção 2.5 para o cálculo da taxa média de CPI e MIPS, que produziram o resultado de CPI = 2,24 e taxa MIPS = 178.**

**Agora, suponha que o programa possa ser executado em oito tarefas paralelas ou threads com aproximadamente o mesmo número de instruções executadas em cada tarefa. A execução é em um sistema de 8 processadores, com cada processador (core) tendo o mesmo desempenho do único processador usado originalmente. A coordenação e a sincronização entre as partes acrescentam mais 25.000 execuções de instrução a cada tarefa. Considere a mesma mistura de instruções do exemplo para cada tarefa, mas aumente o CPI para referência à memória com cada perda de cache para 12 ciclos, devido à disputa pela memória.**

**a) Determine o CPI médio.**

**R:** Assumindo o mesmo mix de instruções significa que as instruções adicionais para cada tarefa deve ser alocada proporcionalmente entre os tipos de instruções, o que nos dá:

Tipo de Instrução	CPI	Instrução Mix
Aritmética e Lógica	1	60%
Load/Store com acerto de cache	2	18%
Branch	4	12%
Memória com falta de cache	12	10%

$CPI = 0,6 + (2 * 0,18) + (4 * 0,12) + (12 * 0,1) = 2,64$ . Observa-se que o CPI aumenta devido ao tempo de acesso de à memória que também teve um aumento.

**b) Determine a taxa MIPS correspondente.**

**R:**  $MIPS = 400/2,64 = 152$ , mostrando que há uma queda na taxa MIPS também.

**c) Calcule o fator de speedup.**

**R:** Tempo de execução =  $(2 * 10^6) / (178 * 10^6) = 11$  ms.

Tempo de execução para cada processador =  $((2 * 10^6) / 8) + (0,025 * 10^6) / 152 * 10^6 = 1,8$  ms.

$Speedup = 11 / 1,8 = 6,11$ .

## PERGUNTAS DE REVISÃO CAPÍTULO 3

**3.1 Que categorias gerais de funções são especificadas pelas instruções do computador?**

**R:** Processadores de memória, Processador de E/S, Processamento de Dados, Controle.



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

**R:** 1 - Cálculo de endereço de instrução: o endereço da próxima instrução a ser executada é determinado. 2 - Busca de instrução: uma instrução é lida na memória e armazenada no processador. 3 - Decodificação de instrução: o código da instrução a ser executada é analisado, para determinar qual é a operação a ser realizada e os operandos a serem usados. 4 - Cálculo de endereço de operando: se a operação envolver a referência a um operando na memória ou estiver disponível via E/S, o endereço do operando será determinado. 5 - Busca de operando: o operando é localizado na memória ou é lido do dispositivo de E/S. 6 - Execução da operação: a operação indicada na instrução é executada. 7 - Armazenamento de resultados: o resultado é escrito na memória ou no dispositivo de E/S.

### **3.3 Liste e defina resumidamente duas técnicas para lidar com múltiplas interrupções.**

**R:** 1 - Desabilitar todas as interrupções enquanto uma interrupção está sendo processada. 2 - Definir prioridades para interrupções e permitir uma interrupção de maior prioridade sob uma interrupção de menor prioridade.

### **3.4 Que tipos de transferências a estrutura de interconexão de um computador precisa aceitar?**

**R:** Memória para processador, Processador para memória, E/S para processador, Processador para E/s e E/S de ou para a memória.

### **3.5 Qual é o benefício de usar a arquitetura de barramento múltiplo em comparação com uma arquitetura de barramento único?**

**R:** Usar um barramento múltiplo é mais eficiente pois é composto de vários barramentos únicos postos hierarquicamente com funções e barramentos específicos. Ao contrário do barramento único, pode receber mais conexões de dispositivos sem perder significativamente o desempenho.

### **3.6 Liste e defina resumidamente os grupos funcionais das linhas de sinal para o barramento PCI.**

**R:** 1 - Pinos do sistema: incluem os pinos de clock e reset. 2 - Pinos de endereços e de dados: incluem 32 linhas que são multiplexadas no tempo para endereços e dados. 3 - Pinos de controle da interface: controlam a temporização de transações e oferecem coordenação entre iniciadores e destinos. 4 - Pinos de arbitração: diferente das outras linhas de sinal do PCI, estas linhas não são compartilhadas. Em vez disso, cada mestre PCI tem seu próprio par de linhas de arbitração que a conectam diretamente ao arbitrador do barramento PCI. 5 - Pinos de erros: usado para indicar erros de paridade e outros. 6 - Pinos de interrupção: estes são disponíveis para dispositivos PCI que precisam gerar solicitações de serviço. 7 - Pinos de suporte à cache: esses pinos são necessários para dar suporte a uma memória no PCI que possa ser armazenada em uma memória cache do processador ou de outro dispositivo. 8 - Pinos de extensão de barramento de 64 bits: incluem 32 linhas que são multiplexadas no tempo para endereços e dados, e que são combinadas com linhas de endereço/dados obrigatórias para formar um barramento de endereço/dados de 64 bits. 9 - JTAG/pinos de testes: essas linhas de sinal admitem procedimentos de teste definidos no padrão IEEE 1149.1.

## **RESPOSTAS PARA OS PROBLEMAS CAP 3**

**3.1 A máquina hipotética da Figura 3.4 também tem duas instruções de E/S:**

**0011 = Carregar AC de E/S**

**0011 = Armazenar AC em E/S**

**Nesses casos, o endereço de 12 bits identifica um dispositivo de E/S em particular. Mostre a execução do programa (usando o formato da Figura 3.5) para o programa a seguir:**

- 1. Carregar AC do dispositivo 5.**
- 2. Somar o conteúdo do local de memória 940.**
- 3. Armazenar AC no dispositivo 6.**

**Suponha que o próximo valor apanhado do dispositivo 5 seja 3 e que o local 940 contenha o valor 2.**

**R:**

300	3005
301	5940
302	7006

1	3005 → IR
2	3 → AC
3	5940 → IR
4	3 + 2 = 5 → AC
5	7006 → IR
6	AC → Dispositivo 6

**3.2 A execução do programa da Figura 3.5 é descrita no texto usando seis etapas. Expanda essa descrição para mostrar o uso do MAR e do MBR.**

**3.3 Considere um microprocessador de 32 bits hipotético com instruções de 32 bits, compostas de dois campos: o primeiro byte contém o opcode e o restante, o operando imediato ou o endereço de um operando.**

**a) Qual é a capacidade de memória máxima endereçável diretamente (em bytes)?**

**R:**  $2^{24} = 16$  MBytes

**b) Discuta o impacto sobre a velocidade do sistema se o barramento do microprocessador tiver**

**R:** 1 - o endereço completo pode ser transferido de uma vez só e decodificado na memória. Porém, como o endereço de dados é de apenas 16 bit, isso exige dois ciclos para buscar uma instrução de 32 bits ou operando.

2 - os 16 bits do endereço de barramento não conseguem acessar toda a memória, assim, um controle de interface de memória mais complexa se torna necessário para travar a primeira parte do endereço e, em seguida, a segunda parte.

**c) Quantos bits são necessários para o contador de programa e o registrador de instrução?**

**R:** O PC deve ter pelo menos 24 bits. Se o registro de instrução é para conter toda a instrução, ele terá de ser 32 bits de comprimento, se ele irá conter apenas o código operação então ela terá que ser de 8 bits de comprimento.

**3.4 Considere um microprocessador hipotético gerando um endereço de 16 bits (por exemplo, suponha que o contador de programa e os registradores de endereço tenham 16 bits de largura) e tendo um barramento de dados de 16 bits.**

**a) Qual é o espaço de endereço de memória máximo que o processador pode acessar diretamente se estiver conectado a uma “memória de 16 bits”?**

**R:** O microprocessador acessa a  $2^{16} = 64K$  bytes, com uma memória de 16 bits pode-se transferir um byte ou uma palavra de 16 bytes.

**b) Qual é o espaço de endereço de memória máximo que o processador pode acessar diretamente se estiver conectado a uma “memória de 8 bits”?**

**R:** O microprocessador acessa a  $2^{16} = 64K$  bytes, com uma memória de 8 bits pode-se transferir um byte.

**c) Que recursos arquiteturais permitirão que esse microprocessador acesse um “espaço de E/S” separado?**

**d) Se uma instrução de entrada e saída pode especificar um número de porta de E/S de 8 bits, quantas portas de E/S de 8 bits o microprocessador pode aceitar? Quantas portas de E/S de 16 bits? Explique.**

**R:** Pode suportar  $2^8 = 256$  portas de entradas e também  $2^8 = 256$  portas de saídas. Sendo necessário em ambos os casos 16 bits.

**3.5 Considere um microprocessador de 32 bits, com um barramento de dados de 16 bits, controlado por um clock de entrada de 8 MHz. Suponha que esse microprocessador tenha um ciclo de barramento cuja duração mínima seja igual a 4 ciclos de clock. Qual é a taxa de transferência de dados máxima pelo barramento que esse microprocessador pode sustentar, em bytes/s? Para aumentar seu desempenho, seria melhor tornar seu barramento de dados externo de 32 bits ou dobrar a frequência de clock externa fornecida ao microprocessador? Informe quaisquer outras suposições que você faça e explique. Dica: determine o número de bytes que podem ser transferidos por ciclo de barramento.**

**R:** Ciclo de clock =  $1/8 \text{ MHz} = 125 \text{ ns}$

Ciclo de Barramento =  $4 \times 125 \text{ ns} = 500 \text{ ns}$

Como há a transferência de 2 bytes a cada 500 ns, a taxa de transferência é igual há 4 Mbytes por segundo. Ao duplicar a frequência pode-se adotar uma nova tecnologia de fabricação de chips, onde assume-se que as instruções terão o mesmo número de ciclos de clock. Ao dobrar o barramento externo de dados, significa que serão mais amplos. No primeiro caso, a velocidade dos chips de memória, onde também será necessário duplicar (aproximadamente) para não retardar o microprocessador, no segundo caso, o "comprimento de palavra" de memória terá de dobrar para ser capaz de enviar/receber 32 quantidades bits.

**3.6 Considere um sistema de computação que contenha um módulo de E/S controlando um teletipo teclado/impressora simples. Os registradores a seguir estão contidos no processador e conectados diretamente ao barramento do sistema:**

<b>INPR:</b>	<b>Registrador de entrada, 8 bits</b>
<b>OUTR:</b>	<b>Registrador de saída, 8 bits</b>
<b>FGI:</b>	<b>Flag de entrada, 1 bit</b>
<b>FGO:</b>	<b>Flag de saída, 1 bit</b>
<b>IEN:</b>	<b>Ativação de interrupção, 1 bit</b>

**A entrada de teclado do teletipo e a saída da impressora no teletipo são controladas pelo módulo de E/S. O teletipo é capaz de codificar um símbolo alfanumérico para uma palavra de 8 bits e decodificar uma palavra de 8 bits para um símbolo alfanumérico.**

**a) Descreva como o processador, usando os quatro primeiros registradores listados neste problema, pode alcançar a E/S com o teletipo.**

**R:** A entrada do teletipo é armazenada em INPR. O INPR só aceita dados do teletipo quando FGI = 0. Quando os dados chegam, ele é armazenado em INPR e FGI é definido como 1. A CPU verifica periodicamente FGI. Se FGI = 1, a CPU transfere os conteúdos da INPR à AC e define FGI a 0. Quando a CPU tem dados para enviar para o teletipo, ele verifica FGO. Se FGO igual à 0, a CPU deve esperar. Se FGO = 1, a CPU transfere os conteúdos da AC para OUTR e define FGO em 0. O teletipo define FGI a 1 após a palavra ser escrita.

**b) Descreva como a função pode ser realizada de forma mais eficiente também empregando a IEN.**

**R:** O processo descrito na letra a é muito custoso. A CPU, que é muito mais rápida do que o teletipo, deve verificar repetidamente FGI e FGO. Se forem utilizadas as interrupções, o teletipo pode emitir uma interrupção para a CPU sempre que ele está pronto para aceitar ou enviar dados. registro IEN pode ser definido pela CPU.

**3.7 Considere dois microprocessadores tendo barramentos de dados externos de 8 e 16 bits, respectivamente. Os dois processadores são idênticos em outros aspectos e seus ciclos de barramento ocupam o mesmo espaço.**

**a) Suponha que todas as instruções e operandos tenham 2 bytes de extensão. Por que fator diferem as taxas de transferência de dados máximas?**

**R:** O microprocessador de 16 bits tem o dobro da taxa de transferência de dados.

**b) Repita considerando que metade dos operandos e instruções tenham 1 byte de extensão.**

**3.8 A Figura 3.26 indica um esquema de arbitração distribuído que pode ser usado com um barramento obsoleto conhecido como Multibus I. Os agentes são encadeados fisicamente em forma de margarida em ordem de prioridade. O agente mais à esquerda no diagrama recebe um sinal bus priority in (BPRN) indicando que nenhum agente com prioridade mais alta deseja o barramento. Se o agente não requisitar o barramento, ele ativa sua linha bus priority out (BPRO). no início de um ciclo de clock, qualquer agente pode requisitar o controle do barramento reduzindo sua linha BPRO. Isso abaixa a linha BPRN do próximo agente na cadeia, que por sua vez, precisa abaixar sua linha BPRO. Assim, o sinal propaga a extensão da cadeia. Ao final dessa reação em cadeia, deverá haver apenas um agente cujo BPRN está ativado e cujo BPRO não está. Esse agente tem prioridade. Se, no início de um ciclo de barramento, o**

barramento não estiver ocupado (BUSY inativo), o agente que tem prioridade pode apanhar o controle do barramento ativando a linha BUSY.

**É preciso um certo tempo para que o sinal BPR se propague do agente de prioridade mais alta para o de prioridade mais baixa. Esse tempo deve ser menor que o ciclo de clock? Explique.**

**R:** O ponto central do clock é para definir os horários dos eventos no barramento, por isso, desejamos para uma operação de barramento arbitrária, a ser feita a cada ciclo de clock. Isto exige que o sinal de prioridade se propague ao longo da cadeia em um período de clock. Assim, o número máximo de mestres é determinado dividindo-se a quantidade de tempo que leva um mestre de barramento para passar através da prioridade barramento pelo período do relógio.

**3.10 No SBI do VAX, o dispositivo com menor prioridade normalmente tem o tempo médio de espera mais baixo. Por esse motivo, o processador normalmente recebe a prioridade mais baixa no SBI. Por que o dispositivo de prioridade 16 normalmente tem o tempo médio de espera mais baixo? Sob que circunstâncias isso não seria verdadeiro?**

**R:** No início de qualquer ranhura, se nenhuma das linhas de TR é afirmado, apenas a prioridade 16 do dispositivo pode transmitir. Isto lhe dá o tempo de espera médio mais baixo na maioria das circunstâncias. Somente quando há grande demanda no barramento, o que significa que a maior parte do tempo, há pelo menos um pedido pendente, será o dispositivo de prioridade 16 que não terá o tempo de espera médio menor.

**3.11 Para uma operação de leitura síncrona (Figura 3.19), o módulo de memória precisa colocar os dados no barramento suficientemente antes da transição de descida do sinal Read para permitir o estabelecimento do sinal. Suponha que o barramento do microprocessador tenha um clock de 10 MHz e que o sinal Read comece a cair no meio da segunda metade de T3.**

**a) Determine a extensão do ciclo de instrução de leitura de memória.**

**R:** O comprimento do ciclo de leitura de memória é de 300ns.

**b) Quando, no máximo, os dados da memória devem ser colocados no barramento? Permita 20ns para o estabelecimento das linhas de dados.**

**R:** O sinal de leitura começa a cair em 75ns a partir do início do terceiro ciclo de relógio. Assim, a memória deve colocar os dados em no barramento no mais tardar até 55ns desde o início da T3.

**3.12 Considere um microprocessador que tenha uma sincronização de leitura de memória conforme mostra a Figura 3.19. Após alguma análise, um projetista determina que a memória não consegue oferecer dados de leitura a tempo por cerca de 180ns.**

**a) Quantos estados de espera (ciclos de clock) precisam ser inseridos para a operação apropriada do sistema se a frequência do sinal de clock do barramento é de 8 MHz?**

**R:** Dois ciclos de relógio devem ser inseridos.

**b) Para forçar os estados de espera, uma linha de estado Ready é empregada. Quando o processador tiver emitido um comando Read, ele precisa esperar até que a linha Ready seja ativada antes de tentar ler dados. Em que intervalo de tempo devemos manter a linha Ready baixa a fim de forçar o processador a inserir o número requisitado de estados de espera?**

**R:** Para inserir dois ciclos de relógio, o ponto da linha pode ser colocado embaixo no início do T2 e mantido abaixo de 250ns.

**3.13 Um microprocessador tem uma sincronização de escrita na memória conforme mostra a Figura 3.19. Seu fabricante especifica que a largura do sinal Write pode ser determinada por  $T - 50$ , onde  $T$  é o período de clock em ns.**

**a) Que largura devemos esperar para o sinal Write se a frequência do sinal de clock do barramento é de 5 Mhz?**

**R:** Um relógio de 5 MHz corresponde a um período de relógio de 200ns. Portanto, o sinal Write tem uma duração de 150ns.

**b) O manual do microprocessador especifica que os dados permanecem válidos por 20ns após a transição de descida do sinal Write. Qual é a duração total da apresentação de dados válida para a memória?**

**R:** Os dados continuam a ser válidos para  $150 + 20 = 170$ ns.

**c) Quantos estados de espera devemos inserir se a memória exigir a apresentação de dados válida por pelo menos 190 ns?**

**R:** Um estado de espera.

**3.14 Um microprocessador tem uma instrução direta de incremento de memória, que soma 1 ao valor em um local da memória. A instrução tem cinco estágios: busca do opcode (quatro ciclos de clock do barramento), busca do endereço do operando (três ciclos), busca do operando (três ciclos), soma de 1 ao operando (três ciclos) e armazenamento do operando (três ciclos).**

**a) Por que quantidade (em percentagem) que a duração da instrução aumentará se tivermos que inserir dois estados de espera do barramento em cada operação de leitura e escrita de memória?**

**R:** Sem os estados de espera, a instrução leva 16 ciclos de clock de barramento. A instrução requer quatro acessos à memória, resultando em oito estados de espera. A instrução, com estados de espera, leva 24 ciclos de clock, para um aumento de 50%.

**b) Repita considerando que a operação de incremento use 13 ciclos em vez de 3 ciclos.**

**R:** Assim a instrução leva 26 ciclos sem estados de espera e 34 ciclos com estados de espera, para um aumento de 33%.

**3.15 O microprocessador Intel 8088 tem uma temporização do barramento de leitura semelhante à da Figura 3.19, mas requer quatro ciclos de clock do processador. Os dados válidos estão no barramento por uma quantidade de tempo que se estende para o quarto ciclo de clock do processador. Considere uma frequência do sinal de clock do processador de 8 MHz.**

**a) Qual é a taxa máxima de transferência de dados?**

**R:** 25 MBytes por segundo.

**b) Repita, mas considere a necessidade de inserir um estado de espera por byte transferido.**

**R:** A taxa de transferência de dados correspondente é  $1/0.625 = 1.6$ MB/s.

**3.16 O Intel 8086 é um processador de 16 bits semelhante, de várias maneiras, ao 8088 de 8 bits. O 8086 utiliza um barramento de 16 bits que pode transferir 2 bytes de cada vez, desde que o byte de mais baixa ordem tenha um endereço par. Porém, o 8086 permite operandos de palavras com alinhamento par ou ímpar. Se uma palavra com alinhamento ímpar for referenciada, dois ciclos de memória, cada um consistindo em quatro ciclos de barramento, são necessários para transferir a palavra. Considere uma instrução no 8086 que envolva dois operandos de 16 bits. Quanto tempo é necessário para buscar os operandos? Dê a faixa de respostas possíveis. Considere uma frequência de sinal de clock de 4MHz e nenhum estado de espera.**

**R:** Um ciclo de barramento leva 0,25us, portanto, um ciclo de memória tem 1us. Se ambos os operandos estiverem alinhados, leva 2us para buscar os dois operandos. Se um é ímpar, o tempo necessário é de 3 us. Se ambos estão alinhado, o tempo necessário é de 4us.

**3.17 Considere um microprocessador de 32 bits cujo ciclo de barramento tenha a mesma duração de um microprocessador de 16 bits. Suponha que, na média, 20% dos operandos e instruções tenham 32 bits de tamanho, 40% tenham 16 bits de tamanho e 40% tenham 8 bits de tamanho. Calcule o ganho ao buscar instruções e operandos com o microprocessador de 32 bits.**

**3.18 O microprocessador do Problema 3.14 inicia o estágio de busca de operando da instrução de incremento direto da memória ao mesmo tempo em que um teclado ativa uma linha de requisição de interrupção. Depois de quanto tempo o processador entra no ciclo de processamento de interrupção? Considere uma frequência do sinal de clock do barramento de 10MHz.**

**R:** O processador precisa de mais nove ciclos de clock para completar a instrução. Assim, a interrupção Reconheça começará depois de 900ns.

## **Perguntas de revisão CAP 4**

**4.1 Quais são as diferenças entre acesso sequencial, acesso direto e acesso aleatório?**

**R:** No acesso sequencial, a memória é organizada dentro de unidades de dados, chamados registros. O endereço de armazenamento é utilizado para separar registros e auxiliar no processo de recuperação. No acesso direto, blocos individuais e registros têm um único endereço baseado na localização física. No acesso sequencial e direto, diferente do acesso aleatório, existe um mecanismo de leitura-escrita compartilhada e o tempo de acesso é variável. No acesso aleatório cada endereço na memória é único e um endereço pode ser encontrado independente da sequência de acesso precedente.

**4.2 Qual é o relacionamento geral entre tempo de acesso, custo de memória e capacidade?**

**R:** Tempo de acesso mais rápido, maior custo da memória. Quanto maior a capacidade, menor o custo, porém quanto maior a capacidade maior o tempo de acesso.

↓ Tempo de acesso    ↑ Custo da memória

↑ Capacidade de memória ↓ Custo da memória  
↑ Capacidade de memória ↑ Tempo de acesso

A memória ideal seria aquela de grande capacidade, rápido tempo de acesso e barata.

#### **4.3 Como o princípio de localidade se relaciona com o uso de múltiplos níveis de memória?**

R: Com o intuito de reduzir o tempo médio de acesso, o uso de múltiplos níveis de memória funciona condicionalmente se a diminuição do custo por bit e o aumento do tempo de acesso se aplicarem. Para a condição do aumento de tempo se tornar válida é usado o princípio de localidade, onde dados acessados recentemente e endereços próximos àqueles acessados recentemente têm mais chances de serem acessados novamente. Sendo assim o sistema de memória tende a manter os dados próximos e recentes no topo da hierarquia de memória.

#### **4.4 Quais são as diferenças entre mapeamento direto, mapeamento associativo e mapeamento associativo em conjunto?**

R: A escolha do tipo de mapeamento determina como a memória cache é organizada. O mapeamento direto é simples, cada bloco da memória principal mapeia uma linha da cache, cada endereço da memória principal pode ser entendido com três bits para identificar qual bloco está mapeando uma determinada linha. Entretanto, existe uma desvantagem, se um programa acessa repetidamente e alternadamente dois blocos de memória mapeados na mesma posição na cache, esses blocos serão inseridos e retirados constantemente da cache, resultando num fenômeno chamado *thrashing*. Já no mapeamento associativo, cada bloco da memória principal pode ser carregado em qualquer linha da cache e, um campo tag de uma linha informa qual bloco está carregado na linha. Para saber se um bloco já está na cache, todas as tags são examinadas simultaneamente, sendo essa a desvantagem do mapeamento associativo, a complexidade para comparar todas as tags em paralelo. Quando a cache está cheia, para substituir um bloco de memória é usado um algoritmo de substituição implantado em hardware para maior velocidade, escolhendo qual linha deve receber o novo bloco. O mapeamento associativo por conjunto é a junção do direto e associativo, unindo a vantagem de ambos. A cache é dividida em conjuntos e usa-se o último dígito do conjunto para determinar em qual conjunto o bloco vai (mapeamento direto), porém dentro do conjunto, o bloco pode ser colocado em qualquer linha e usa-se o acesso associativo em cada conjunto para saber se o bloco já está ou não presente (mapeamento associativo).

#### **4.5 Para uma cache mapeada diretamente, um endereço de memória principal é visto como consistindo em três campos. Liste e defina os três campos.**

R: Os três campos são definidos como  $s$  (os bits restantes que especificam um dos  $2^s$  blocos da memória principal) -  $r$  (parte mais significativa de uma das  $2^r$  linhas de cache). Assim somente  $s - r$  bits são necessários para apontar qual bloco está mapeado, pois os últimos  $r$  bits são iguais a posição da linha.



**4.6 Para uma cache associativa, um endereço de memória principal é visto como consistindo em dois campos. Liste e defina os dois campos.**

R: Usa-se uma tag, suponha uma memória de  $2^s$  blocos, o campo s de uma linha informa qual bloco está carregado na linha.

**4.7 Para uma cache associativa em conjunto, um endereço da memória principal é visto como consistindo em três campos. Liste e defina os três campos.**

R: Os três campos são tag, set e palavra. Os bits do campo tag e set especificam um dos  $2^s$  blocos da memória principal.

**4.8 Qual é a diferença entre localidade espacial e localidade temporal.**

R: Localidade espacial, um dado ou instrução é acessado recentemente existe uma probabilidade maior de acessar um dado ou instrução próximos a ele. Na localidade temporal, um dado ou instrução acessado recentemente têm maior probabilidade de ser acessado novamente do que uma acessada a mais tempo.

**4.9 Em geral, quais são as estratégias para explorar a localidade espacial e a localidade temporal?**

R: Para diminuir o tempo de processamento, é usado o princípio da localidade, onde os dados e instruções são organizados no topo da hierarquia de memória de acordo com a localidade espacial e temporal.

## **Problemas CAP 4**

**4.1 Uma cache associativa em conjunto consiste em 64 linhas, ou slots, divididas em conjuntos de quatro linhas. A memória principal contém 4 K blocos de 128 palavras cada. Mostre o formato dos endereços da memória principal.**

R: A cache é dividida em 16 conjuntos de 4 linhas cada, com quatro bits para identificar o número de série. A memória principal consiste em  $4K = 2^{12}$  blocos, então, os comprimentos de tag conjunto devem ser de 12 bits e, o comprimento tag é de 8 bits. Cada bloco possui 128 palavras e para especificar a palavra usa-se sete bits.

**4.2 Uma cache associativa em conjunto com duas linhas por conjuntos possui linhas de 16 bytes e um tamanho total de 8 KBytes. A memória principal de 64 MBytes é endereçável por byte. Mostre o formato dos endereços da memória principal.**

R: Existem 16 bytes = 512 linhas na cache, assim a cache consiste em 256 conjuntos de duas linhas cada. São necessários para identificar o conjunto 8 bits. Para a memória principal de 64 MBytes é necessário um endereço de 26 bits, e é constituída por  $64 \text{ MByte} / 16 \text{ bytes} = 2^{22}$  blocos, portanto o comprimento de tag deve ser de 22 bits. O comprimento de tag é 14 bits e o comprimento do campo palavra é de 4 bits.

**4.3 Para os endereços hexadecimais da memória principal 111111, 666666, BBBBBB, mostre a seguinte informação, em formato hexadecimal:**

**a. Valores de tag, linha e palavra para uma cache de mapeamento direto, usando o formato da Figura 4.10.**

R: 11/444/166/1999/2BB/2EEE/3

**b. Valores de tag e palavra para uma cache associativa, usando o formato da Figura 4.12.**

R: 44444/1199999/22EEEE/3

**c. Valores de tag, conjunto e palavra para uma cache associativa em conjunto com duas vias, usando o formato da Figura 4.15.**

R: 22/444/1CC/1999/2177/EEE/3

**4.4 Liste os seguintes valores:**

**a. Para o exemplo de cache direta da Figura 4.10: tamanho do endereço, número de unidades endereçáveis, tamanho de bloco, número de blocos na memória principal, número de linhas na cache, tamanho da tag.**

R: Tamanho do endereço: 24, número de unidades endereçáveis:  $2^{24}$ , tamanho de bloco: 4, número de blocos na memória principal:  $2^{22}$ , número de linhas na cache:  $2^{14}$ , tamanho da tag: 8.

**b. Para o exemplo de cache associativa da Figura 4.12: tamanho do endereço, número de unidades endereçáveis, tamanho de bloco, número de blocos na memória principal, número de linhas na cache, tamanho da tag.**

R: Tamanho do endereço: 24, número de unidades endereçáveis:  $2^{24}$ , tamanho de bloco: 4, número de blocos na memória principal:  $2^{22}$ , número de linhas na cache: 4000 hex, tamanho da tag: 22.

**c. Para o exemplo de cache associativa em conjunto com duas linhas por conjunto da Figura 4.15: tamanho do endereço, número de unidades endereçáveis, tamanho de bloco, número de blocos na memória principal, número de linhas no conjunto, número de conjuntos, número de linhas na cache, tamanho da tag.**

R: Tamanho do endereço: 24, número de unidades endereçáveis:  $2^{24}$ , tamanho de bloco: 4, número de blocos na memória principal:  $2^{22}$ , número de linhas no conjunto: 2, número de conjuntos:  $2^{13}$ , número de linhas na cache:  $2^{14}$ , tamanho da tag: 9.

**4.5 Considere um microprocessador de 32 bits que tem uma cache associativa em conjunto com quatro linhas por conjunto de 16 KBytes no chip. Suponha que a cache tenha um tamanho de linha de quatro palavras de 32 bits. Desenhe um diagrama de blocos dessa cache, mostrando sua organização e como os diferentes campos de endereço são usados para determinar um acerto/falha de cache. Onde, na cache, a palavra no local de memória ABCDE8F8 é mapeada?**

R:

**4.6 Dadas as seguintes especificações para uma memória cache externa: associativa em conjunto com quatro vias; tamanho de linha de duas palavras de 16 bits; capaz de acomodar um total de 4 K palavras de 32 bits da memória principal; usada com um processador de 16 bits que emite endereços de 24 bits. Projete a estrutura de cache com todas as informações pertinentes e mostre como ela interpreta os endereços do processador.**

**4.8 Considere uma máquina com uma memória principal endereçável por byte com  $2^{16}$  bytes e um tamanho de bloco de 8 bytes. Suponha que uma cache mapeada diretamente, consistindo em 32 linhas, seja usada com essa máquina.**

**a. Como um endereço de memória de 16 bits é dividido em tag, número de linha e número de byte?**

R: Tag = 8 bits à esquerda;

Número de linha = 5 bits;

Número de byte = 3 bits à direita;

**b. Em que linha seriam armazenados os bytes com cada um dos seguintes endereços? 0001 0001 0001 1011 1100 0011 0011 0100 1101 0000 0001 1101 1010 1010 1010 1010**

R: Slot 3, ranhura 6; slot 3, ranhura 21;

**c. Suponha que o byte com endereço 0001 1010 0001 1010 seja armazenado na cache. Quais são os endereços dos outros bytes armazenados junto com ele?**

R: Bytes com endereços 0001 1010 0001 1000 através de 0001 1010 0001 1111 são armazenados no cache.

**d. Quantos bytes de memória no total podem ser armazenados na cache?**

R: 256 bytes.

**e. Por que a tag também é armazenada na cache?**

R: Porque dois dados com endereços de memória diferentes podem ser armazenados no mesmo lugar na cache, a tag é usada para diferenciar entre eles.

**4.10 Uma cache associativa em conjunto tem um tamanho de bloco de quatro palavras de 16 bits e um tamanho de conjunto de 2. A cache pode acomodar um total de 4.096 palavras. O tamanho da memória**

**principal que pode ser mantido em cache é de 64 K × 32 bits. Projete a estrutura da cache e mostre como os endereços do processador são interpretados.**

R: Tamanho do bloco = 4 palavras = 2; doublewords associatividade K = 2; tamanho da cache = 4048 palavras, c = 1024 blocos, número de conjuntos S = C / K = 512; memória principal = 64K x 32 bits = 256 KBytes =  $2^{18}$  bytes; endereço = 18 bits.

**4.12 Considere um computador com as seguintes características: total de 1 MByte de memória principal; tamanho de palavra de 1 byte; tamanho de bloco de 16 bytes; e tamanho de cache de 64 KBytes.**

**a. Para os endereços de memória principal F0010, 01234 e CABBE, indique os deslocamentos de tag, endereço de linha de cache e palavra para uma cache mapeada diretamente.**

R: Porque o tamanho do bloco é de 16 bytes e o tamanho de palavra é um byte, existem 16 palavras por bloco. Precisa-se de 4 bits para indicar a palavra desejada fora de um bloco. Cada linha de cache corresponde a um bloco de memória, então cada slot cache contém 16 bytes. Se a cache é 64 KBytes então  $64 \text{ KBytes} / 16 = 4096$  slots de cache. Para responder a estas 4096 vagas de cache, precisamos de 12 bits ( $2^{12} = 4096$ ). De modo que, dada a 20 bits principal endereço de memória: 0-3 bits indicam a palavra offset (4 bits).

**b. Indique dois endereços quaisquer da memória principal com diferentes tags que são mapeados para o mesmo slot de cache para uma cache mapeada diretamente.**

R: Endereço 1:

Compensar palavra = 1111 = Slot de 1111 1111 1111 Tag 0000

Endereço = 0FFFF

Endereço 2:

Compensar palavra = 0001 = Slot de 1111 1111 1111 Tag = 0011

Endereço = 3FFF1

**c. Para os endereços da memória principal F0010 e CABBE, indique os valores correspondentes de tag e deslocamento para uma cache totalmente associativa.**

R: Com uma cache totalmente associativa, a cache é dividida em uma tag e um campo wordoffset. Não precisa identificar qual slot de um bloco de memória, pois um bloco pode ser em qualquer slot e procura-se em paralelo. O deslocamento da palavra deve ser de 4 bits para endereçar cada palavra no bloco de 16 palavras, isso deixa 16 bits restantes para a marca.

F0010:

Wordoffset = 0h Tag = F001h

CABBE:

Compensar palavra = Eh Tag = CABBh

**d. Para os endereços da memória principal F0010 e CABBE, indique os valores correspondentes de tag, conjunto de cache e deslocamento para uma cache associativa em conjunto com duas vias.**

R: 0-3 bits: indicam a palavra compensar;

4-14 bits: indicar o conjunto de cache;

15-20 bits: indicam a tag;  
F0010 = 1111 0000 0000 0001 0000  
Wordoffset = 0000 = 0  
Cache Set = 000 0000 0001 = 001  
Tag = 11110 = 1 1110 = 1E  
CABBE = 1100 1010 1011 1011 1110  
Compensar palavra = 1110 = E  
Cache set = 010 1011 1011 = 2BB  
Tag = 11001 = 1 1001 = 19

## **Questões de revisão CAP 5**

### **5.1 Quais são as principais propriedades da memória semicondutora?**

R: As memórias semicondutoras compreendem dois estados estáveis, que podem ser usados para representar 0 e 1. São capazes de ser escritas, pelo menos uma vez, para determinar o estado, e de serem lidas, para verificar o estado.

### **5.2 Quais são os dois sentidos em que o termo memória de acesso aleatório é usado?**

R: O termo memória de acesso aleatório é usado para identificar memória semicondutora RAM e outras memórias de apenas leitura (ROM, por exemplo) que são memórias que as palavras são acessadas diretamente por meio da lógica de endereçamento interna.

### **5.3 Qual é a diferença entre DRAM e SRAM em termos de aplicação?**

R: A DRAM é utilizada na memória principal e a SRAM é usada na cache.

### **5.4 Qual é a diferença entre DRAM e SRAM em termos das características como velocidade, tamanho e custo?**

R: As SRAM são, geralmente, um pouco mais rápidas que as DRAM. Uma DRAM é mais simples e menor que uma célula SRAM e a DRAM é mais densa e mais barata que uma SRAM correspondente.

### **5.5 Explique por que um tipo de RAM é considerado como analógico e o outro digital.**

R: Uma RAM é tida como analógica pois armazena dados como cargas em capacitores, onde a presença ou ausência de carga representa um binário. Já a SRAM é vista como digital, pois os valores binários são armazenados por meio de configurações das portas lógicas de um flip flop tradicional.

## **5.6 Quais são algumas aplicações para a ROM?**

R: Microprogramação; biblioteca de funções de uso frequente; programas do sistema; tabelas de função.

## **5.7 Quais são as diferenças entre EPROM, EEPROM e memória flash?**

R: A EPROM é lida e escrita eletricamente, porém antes da operação de escrita todas as células de armazenamento precisam ser apagadas para retornar ao estado inicial e, cada apagamento pode levar até 20 minutos para acontecer. A EEPROM é uma memória principalmente de leitura que pode ser escrita sem apagar o conteúdo anterior, somente os bytes endereçados são atualizados. E a memória flash, é o meio termo entre a EPROM e a EEPROM, assim como no custo e na funcionalidade. Ela pode ser apagada em alguns segundos, pois usa uma tecnologia elétrica de apagamento, porém a memória flash não disponibiliza apagamento a nível de byte.

## **5.8 Explique a função de cada pino na Figura 5.4b.**

R: Pinos A0 - A19: endereço da palavra sendo acessada.

Pinos D0 - D7: dados a serem lidos (pinos de entrada e saída).

Pino Vcc: alimentação para o chip.

Pino Vss: pino de terra.

Pino CE: usado para indicar se o endereço é válido ou não pra esse chip.

Pino Vpp: voltagem de programa fornecida durante as operações de escrita.

Pinos WE e OE: indicam se é uma operação de escrita ou leitura.

Pino RAS: sinais de seleção de endereço de linha.

Pino CAS: sinais de seleção de endereço de coluna.

Pino NC: pra ter um número par de pinos.

## **5.9 O que é bit de paridade?**

R: Bit de paridade é utilizado para detectar um erro no envio da mensagem. Para evitar erros de transmissão um bit de paridade é adicionado a mensagem, cada bit de paridade é escolhido de modo que o número total de 1s em seu círculo seja par. Assim se houver discrepâncias, o computador saberá que houve um erro.

## **5.10 Como é interpretada a palavra síndrome para o código de hamming?**

R: A palavra síndrome indica se algum erro foi detectado e em qual bit estava o erro.

## **5.11 Como a SDRAM difere de uma DRAM comum?**

R: SRAM tem uma frequência de clock e uma taxa de transferência menor que uma DRAM, porém um tempo de acesso maior. A SDRAM, diferente da DRAM, troca dados com o processador sincronizado com um sinal de

clock externo e executa na velocidade plena do barramento do processador/memória, sem necessidades de estados de espera.

## Problemas CAP 5

**5.1 Sugira motivos pelos quais as RAM têm sido tradicionalmente organizadas como 1 bit por chip, enquanto as ROM normalmente são organizadas com múltiplos bits por chip.**

R: Além do menor custo, organizar uma memória como 1 bit por chip tem várias vantagens, ela requer menos pinos e, portanto, uma maior densidade de bits pode ser alcançada por um determinado pacote.

**5.11 Suponha que uma palavra de dados de 8 bits armazenada na memória seja 11000010. Usando o algoritmo de hamming, determine quais bits de verificação seriam armazenados na memória com a palavra de dados. Mostre como você chegou a sua resposta.**

R: Os bits de dados com o valor 1 se encontrarem em posição de bit 12, 11, 5, 4, 2 e 1;

Posição 12110987654321

Bits D8D7D6D5C8D4D3D2D1C2C1

Bloco 11000010

Códigos 110010110101

Os bits de verificação são: 0 0 1 0

**5.12 Para uma palavra de 8 bits 00111001, os bits de verificação armazenados com ela seriam 0111. Suponha, quando a palavra for lida da memória, que os bits de verificação são calculados como 1101. Qual palavra de dados foi lida da memória?**

R: A palavra inicialmente calculada foi: 001101001111

A palavra de dados lida da memória foi: 00011001.

**5.13 Quantos bits de verificação são necessários se o código de correção de erro de hamming for usado para detectar erros de único bit em uma palavra de dados de 1024 bits?**

R: Precisa de X bits de verificação,  $\{ X \in \mathbb{I} \mid 1024 + X \leq 2^X - 1 \}$  e o valor mínimo que satisfaz essa condição é 11.