

GABARITO AD1 - Organização de Computadores 2006.1

1. Descreva passo a passo as operações de leitura da memória e de escrita na memória, indicando como os registradores RDM e REM são utilizados.

Resposta :

Na leitura, inicialmente o REM é atualizado com um endereço contido em um registrador qualquer da UCP.

O endereço armazenado no REM é colocado no barramento de endereços. A unidade de controle gera um sinal de leitura no barramento de controle. São feitas a decodificação do endereço e localização da célula e finalmente o RDM é carregado com o valor da memória principal relativo ao endereço contido no REM, através do barramento de dados. Um outro registrador da UCP é atualizado com o conteúdo do RDM.

Na escrita, inicialmente o REM é atualizado com um endereço contido em um registrador qualquer da UCP.

O endereço armazenado no REM é colocado no barramento de endereços.

Da mesma forma o RDM é atualizado com um valor contido em um registrador qualquer da UCP.

A unidade de controle gera um sinal de escrita no barramento de controle. A memória é atualizada com o valor do RDM no endereço dado pelo REM.

2. Um computador possui uma capacidade máxima de memória principal com 32K células, cada uma capaz de armazenar uma palavra de 8 bits.

- a) Qual é o maior endereço em decimal desta memória ?

Resposta: Os endereços variam de 0 até $2^{15} - 1$, ou seja, de 0 até 32767. O maior endereço é portanto 32767.

- b) Qual é o tamanho do barramento de endereços deste sistema ?

Resposta: Como a máquina endereça 32K células = 2^{15} células, são necessários 15 bits para endereçá-las. Logo o barramento de endereços tem que ter tamanho igual a 15 bits.

- c) Quantos bits podem ser armazenados no RDM e no REM ?

Resposta: RDM tem 8 bits, tamanho da palavra da memória principal, REM tem a mesma quantidade de bits do barramento de endereços, 15 bits.

- d) Qual é o número máximo de bits que pode existir na memória ?

Resposta: A capacidade da memória é igual ao número de células multiplicado pelo tamanho da célula, ou seja, $32K \times 8 = 256K$ bits

3. Considere uma máquina que possa endereçar 512 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 32 bytes. Ela possui uma memória cache que pode armazenar 8K blocos, sendo um bloco por linha. Mostre o formato da memória cache, indicando os campos necessários (tag, bloco) e o número de bits para cada campo, e o formato de um endereço da memória principal, indicando os bits que referenciam os campos da cache, para os seguintes mapeamentos:
- a) Mapeamento direto.

Resposta:

Como há 16 M blocos na memória principal e 8 K blocos na cache, então cada linha deverá acomodar 2 K blocos, um de cada vez, é claro.

Formato da memória cache:

A memória cache terá 8K linhas (ou quadros) em cada linha teremos tag de 11 bits e bloco de 32 bytes.

O endereço da memória principal terá o seguinte formato:

Campo de número do bloco: de 11 bits (para endereçar um dos 2 K blocos)

Campo de número da linha: de 13 bits (já que temos 8 K blocos)

Campo de endereço da palavra: 5 bits (já que temos 32 bytes por bloco)

- b) Mapeamento totalmente associativo.

Resposta:

Como há 16 M blocos na MP e eles podem ficar em quaisquer dos 8 K blocos da Cache, temos:

Formato da memória cache:

A memória cache terá 8K linhas (ou quadros), em cada linha teremos tag de 24 bits e bloco de 32 bytes.

O endereço da memória principal terá o seguinte formato:

Campo de número do bloco: de 24 bits (para endereçar um dos 16 M blocos)

Campo de endereço da palavra: 5 bits (já que temos 32 bytes por bloco)

- c) Mapeamento associativo por conjunto, onde cada conjunto possui quatro linhas, cada uma de um bloco.

Resposta:

A memória cache neste caso terá um número de conjuntos igual a $8K \text{ linhas} / 4 = 2K$. Onde, cada um dos 16M blocos pode ficar em um dos 2K conjuntos, ou seja, cada conjunto poderá alocar 8K blocos.

Assim, o formato da memória cache será: 2k conjuntos de 4 linhas e cada linha com bloco de 32 K.

O tag da memória cache terá 13 bits e bloco de 32 bytes.

O endereço da MP terá os seguintes campos:

Tag: 13 bits (já que cada conjunto poderá alocar 8K blocos)

Número de conjuntos: 11 bits (já que existem 2K conjuntos)
Endereço da palavra: 5 bits (já que temos 32 bytes por blocos)

4. Explique em detalhes a organização hierárquica do subsistema de memória nos computadores atuais.

Resposta:

Há muitas memórias no computador: O subsistema de memória é interligado de forma bem estruturada e organizado hierarquicamente em uma pirâmide com os níveis descritos a seguir.

No topo da pirâmide teríamos os registradores, que são pequenas unidades de memória que armazenam dados na UCP. São dispositivos de maior velocidade de transferência, menor capacidade de armazenamento e maior custo.

Em um nível abaixo teríamos a memória cache, cuja função é acelerar a velocidade de transferência das informações entre UCP e MP e, com isso, aumentar o desempenho do sistema. A UCP procura informações primeiro na Cache. Caso não as encontre, as mesmas são transferidas da MP para a Cache.

Abaixo da memória cache teríamos a memória básica de um sistema de computação, que é a memória principal. Dispositivo onde o programa (e seus dados) que vai ser executado é armazenado para que a UCP busque instrução por instrução para executá-las.

Finalmente, na base da pirâmide teríamos a memória secundária, memória auxiliar ou memória de massa, que fornece garantia de armazenamento mais permanente aos dados e programas do usuário. Alguns dispositivos são diretamente ligados: discos rígidos, outros são conectados quando necessário: disquetes, fitas de armazenamento, CD-ROM. Possuem menor velocidade de transferência, maior capacidade de armazenamento e menor custo.

5. Considere uma máquina com 64K células de memória onde cada célula armazena uma palavra e cada instrução tem o tamanho de uma palavra. Esta máquina possui um conjunto de instruções com 32 instruções distintas, sendo cada uma delas composta de um código de operação e um único operando, que indica o endereço de memória.

- a) Qual o tamanho mínimo do REM ?

Resposta: Como a máquina endereça 64K células = 2^{16} células, são necessários 16 bits para endereçá-las. Logo o REM tem que ter tamanho mínimo igual a 16 bits.

- b) Qual o tamanho mínimo do RI ?

Resposta: O RI tem que ter capacidade para armazenar uma instrução. Como podemos ter 32 instruções distintas, utilizam-se 5 bits para o campo do código de operação. O outro campo da instrução contém o endereço do operando, para o qual se necessitam de 16 bits. Logo o tamanho total da instrução é 21, e portanto o tamanho mínimo de RI é 21.

- c) Qual a capacidade da memória em bits ?

Resposta: A capacidade da memória é igual ao número de células multiplicado pelo tamanho da célula. Como cada célula armazena uma instrução, teremos que a capacidade da memória é $64K \times 21 = 1344K$ bits

- d) Se a largura do barramento de dados desta máquina for igual à metade do tamanho de uma instrução, como funcionará o ciclo de busca ?

Resposta: No ciclo de busca terão que ser executados dois acessos à memória para poder acessar os bits da instrução

6. Considere um computador que possua uma UCP com CI de 16 bits e RI de 40 bits. Suas instruções possuem dois operandos do mesmo tamanho (cada um com 16 bits) e um código de operação. Cada célula de memória tem o tamanho igual ao de uma instrução.

a) Qual o tamanho da instrução ?

Resposta: O tamanho da instrução é igual ao tamanho do RI, 40 bits

b) Qual o tamanho do código de operação ?

Resposta: A instrução possui um tamanho total de 40 bits e possui três campos, dois para operandos com tamanho de 16 bits e um para o código de operação que tem tamanho igual a $40 - 32 = 8$ bits

c) Considerando que a configuração básica dessa máquina é de 16 Kbytes de memória, mostre se é possível aumentar a quantidade de memória desta máquina. Caso seja possível, calcule a quantidade máxima que pode ser adicionada a este sistema.

Resposta: A capacidade máxima da memória em bytes que este sistema pode ter é igual a: número de células \times tamanho de cada célula $= 2^{16} \times 40/8 = 320$ K bytes. Então se esta máquina possui 16 Kbytes, pode-se acrescentar mais 304 Kbytes.

7. Considere o sistema visto na aula 4 que possui uma memória com 256 células, sendo que cada célula pode armazenar 12 bits. Cada instrução possui 4 bits que indicam o código de operação e 8 bits que indicam o operando. Suponha que, em um determinado momento, alguns endereços da memória contenham os seguintes conteúdos (todos os valores estão em hexadecimal):

Endereço	Conteúdo
20	170
21	529
22	150
23	351
24	250
25	170
26	451
27	270
28	820
29	000
50	006
51	001
70	003

Considere que nos endereços 20 a 29 estão armazenadas instruções de um programa e nos endereços 50, 51 e 70, valores numéricos (inteiros sem sinal).

- a) Traduza as instruções para as siglas e operandos correspondentes. Por exemplo, a instrução contida no endereço 20 é LDA 70.

Resposta:

Endereço	Conteúdo
20	LDA 70
21	JZ 29
22	LDA 50
23	ADD 51
24	STR 50
25	LDA 70
26	SUB 51
27	STR 70
28	JMP 20
29	HLT
50	006
51	001
70	003

- b) Supondo que o CI possua o endereço 20, indique como será realizada a execução de cada instrução deste programa e mostre o conteúdo de todos os endereços da tabela e dos registradores RDM, REM, RI, CI e ACC ao final da execução deste programa.

Resposta: Sequência de execução das instruções:

LDA 70 – Carrega ACC com valor 3
JZ 29 - Vai buscar instrução no próximo endereço porque ACC diferente de 0
LDA 50 - Carrega ACC com 6
ADD 51 - Soma conteúdo de ACC com 1 (ACC=7)
STR 50 - Armazena valor 7 na posição de memória de end. 50
LDA 70 – Carrega ACC com valor 3
SUB 51 – Subtrai conteúdo de ACC de 1 (ACC=2)
STR 70 – Armazena o valor 2 na posição de memória de end. 70
JMP 20 – Busca a instrução na posição de memória de endereço 20
LDA 70 – Carrega ACC com valor 2
JZ 29 - Vai buscar instrução no próximo endereço porque ACC diferente de 0
LDA 50 - Carrega ACC com 7
ADD 51 - Soma conteúdo de ACC com 1 (ACC=8)
STR 50 - Armazena valor 8 na posição de memória de end. 50
LDA 70 – Carrega ACC com valor 2
SUB 51 – Subtrai conteúdo de ACC de 1 (ACC=1)
STR 70 – Armazena o valor 1 na posição de memória de end. 70
JMP 20 – Busca a instrução na posição de memória de endereço 20

LDA 70 – Carrega ACC com valor 1
 JZ 29 - Vai buscar instrução no próximo endereço porque ACC diferente de 0
 LDA 50 - Carrega ACC com 8
 ADD 51 - Soma conteúdo de ACC com 1 (ACC=9)
 STR 50 - Armazena valor 9 na posição de memória de end. 50
 LDA 70 – Carrega ACC com valor 1
 SUB 51 – Subtrai conteúdo de ACC de 1 (ACC=0)
 STR 70 – Armazena o valor 0 na posição de memória de end. 70
 JMP 20 – Busca a instrução na posição de memória de endereço 20
 LDA 70 – Carrega ACC com valor 0
 JZ 29 - Vai buscar instrução na posição de memória de endereço 29 porque ACC igual a 0
 HLT – para execução do programa

Conteúdo da memória:

Endereço	Conteúdo
20	170
21	529
22	150
23	351
24	250
25	170
26	451
27	270
28	820
29	000
50	009
51	001
70	000

Conteúdo dos registradores ao final da execução:

RDM : 000 (instrução HLT)
 REM: 29 (endereço da instrução HLT)
 RI: : 000 (instrução HLT)
 CI: 2A (endereço seguinte ao da instrução HLT)
 ACC: 000

8. Considere a execução de 5 instruções, cada uma delas com tempo de execução igual a T_{ex} . Calcule o tempo para executar estas 5 instruções, para cada caso abaixo:
 - a) Execução em uma máquina que realiza a execução das instruções de forma sequencial
 Resposta: $T_{total} = 5 \times T_{ex}$
 - b) Execução em uma máquina que utiliza um pipeline de 3 estágios, considerando que o tempo de execução da instrução pode ser dividido igualmente entre os estágios e não há atrasos na execução do pipeline, devido a desvios condicionais.

Resposta: O tempo de execução total de um grupo de instruções em uma arquitetura pipeline é dado pelo tempo de execução de uma instrução (T_{instr}) somado a $((\text{Número de instruções} - 1) \times \text{Tempo de um estágio } (T_{estag}))$

$$T_{total} = T_{ex} + 4 \times T_{ex}/3 = 2,33 T_{ex}$$

- c) Execução em uma máquina que utiliza um pipeline de 4 estágios, considerando que o tempo de execução da instrução pode ser dividido igualmente entre os estágios e não há atrasos na execução do pipeline, devido a desvios condicionais.

$$\text{Resposta: } T_{total} = T_{ex} + 4 \times T_{ex}/4 = 2 T_{ex}$$

- d) Execução em uma máquina que utiliza um pipeline de 5 estágios, considerando que o tempo de execução da instrução pode ser dividido igualmente entre os estágios e não há atrasos na execução do pipeline, devido a desvios condicionais.

$$\text{Resposta: } T_{total} = T_{ex} + 4 \times T_{ex}/5 = 1,8 T_{ex}$$

- e) Execução em uma máquina que utiliza um pipeline de 6 estágios, considerando que o tempo de execução da instrução pode ser dividido igualmente entre os estágios e não há atrasos na execução do pipeline, devido a desvios condicionais.

$$T_{total} = T_{ex} + 4 \times T_{ex}/6 = 1,67 T_{ex}$$

9. O barramento PCI possui algumas versões diferentes relacionadas à taxa de transferência (número de bytes que podem ser transmitidos em um segundo) e a frequência do relógio, conforme mostrado na tabela abaixo:

Tipo de barramento	Taxa de transferência (Mbytes/s)	Frequência do relógio (MHz)
V1	132	33
V2	264	66
V3	264	33
V4	512	66

Nota: Os nomes V1, V2, V3 e V4 não fazem parte do padrão, só foram utilizados no exercício para referenciar as diversas versões.

Indique para cada uma das versões, quantos ciclos de relógio são necessários para realizar a transferência de 128 bits no barramento.

Resposta: O número de bits que podem ser transferidos em um ciclo de relógio é dado por: $N_{bitsporciclo} = \text{Taxa de transferência} / \text{Frequência do relógio}$. Então para se transferir 128 bits, serão necessários $128 / N_{bitsporciclo}$ ciclos de relógio.

Para V1, temos que $N_{bitsporciclo} = (132 \times 8) M / 33 M = 32$, logo Número de ciclos = $128/32 = 4$

Para V2, temos que $N_{bitsporciclo} = (264 \times 8) M / 66 M = 32$, logo Número de ciclos = $128/32 = 4$

Para V3, temos que $N_{bitsporciclo} = (264 \times 8) M / 33 M = 64$, logo Número de ciclos = $128/64 = 2$

Para V4, temos que $N_{bitsporciclo} = (512 \times 8) M / 66 M = 64$, logo Número de ciclos = $128/64 = 2$

10. Escreva a sequência de microoperações que devem ser realizadas pelo ciclo de execução das seguintes instruções:

- a) SUB Op. $ACC \longleftarrow ACC - (Op.)$

Resposta:

t1: REM $\longleftarrow (RI(\text{Endereço}))$

t2: RDM \leftarrow Memória
t3: ACC \leftarrow (ACC) - (RDM)

b) JMP Op. CI \leftarrow Op

Resposta:

t1: CI \leftarrow (RI(Endereço))

c) INC Op. (Op.) \leftarrow (Op.) + 1

Resposta:

t1: REM \leftarrow (RI(Endereço))

t2: RDM \leftarrow Memória

t3: ACC \leftarrow (RDM) + 1

t4: RDM \leftarrow ACC

t5: Memória \leftarrow RDM