

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

TCC04.070-Organização de Computadores I – Turma :A1 – Lista 2

- Um computador pode endereçar 256M células da memória principal, cada uma capaz de armazenar uma palavra de 8 bits. Em cada acesso à memória, realiza-se o acesso a uma palavra.
 - Qual é o maior endereço em decimal desta memória?
 - Qual é o tamanho do barramento de endereços deste sistema?
 - Qual é o tamanho do barramento de dados?
 - Qual é o número máximo de bits que pode existir na memória?
- Considere uma máquina que possa endereçar 64 Mbytes de memória física, utilizando endereço referenciando byte, e que tenha a sua memória organizada em blocos de 4 bytes. Ela possui uma memória cache que pode armazenar 2K blocos. Mostre o formato da memória cache, indicando os campos necessários (válido, 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 cada um dos mapeamentos abaixo:
 - mapeamento direto
 - mapeamento totalmente associativo
 - mapeamento associativo por conjunto com 4 linhas por conjunto.
- Uma maneira utilizada para medir o desempenho de um sistema que possui memória cache é calcular o tempo médio de acesso à memória (TMAM). Este tempo considera tanto os eventos de acerto quanto falta na cache e a frequência com que ocorrem. Ele é dado por:
$$\text{TMAM} = \text{tempo de acerto} + \text{taxa de faltas} \times \text{penalidade por falta}$$
onde tempo de acerto engloba o tempo de descobrir que ocorreu um acerto e o acesso à memória cache, penalidade por falta engloba o tempo de descoberta da ocorrência da falta adicionado ao tempo de cópia de um bloco da memória principal para a cache e o tempo de acesso a cache e a taxa de falta indica o número de faltas que ocorrem por instrução.

Considere uma máquina que possui um relógio com um ciclo de 2ns, apresenta uma penalidade de falta igual a 20 ciclos de relógio, uma taxa de faltas de 0.05 faltas por instrução e o tempo de acerto igual a 1 ciclo de relógio.

 - Calcule TMAM para esta máquina.
 - Suponha que se aumente a quantidade de memória cache desta máquina, e, com esta mudança, a taxa de faltas mude para 0.03 faltas por instrução e o tempo de acerto passa para 1.2 ciclos de relógio. Utilizando TMAM, mostre se esta mudança irá melhorar o desempenho da máquina.
- Considere um sistema de computação que possui uma UCP com um registrador RI (Registrador de Instrução) de 32 bits e 16 registradores, e pode endereçar no máximo 1M células de memória principal. Suas instruções possuem três campos: um para o código de operação, um para um operando que se encontra em um registrador e o terceiro para um operando que se encontra na memória. Indique:
 - Número de bits da instrução
 - Número de bits do código de operação
 - Número máximo de operações diferentes
- Converta o seguinte trecho de código em C para a linguagem de montagem vista em aula. Este programa compara cada elemento na posição K do vetor A com o elemento na mesma posição no vetor B. Caso eles sejam iguais, o elemento K do vetor C tem o valor 1 atribuído a ele, caso contrário, o valor atribuído é 0. Assuma que a variável inteira K está alocada ao registrador 1, o endereço inicial do vetor A está no registrador 2, o endereço inicial do vetor B está no registrador 3 e o endereço inicial do vetor C está no registrador 4. Utilize o programa SimulaPC para verificar se a sua conversão está correta.

```
for(K=0; K < 5; K++) {
```

```

if(A[K]==B[K]) {
    C[K]=1;
}
else {
    C[K]=0;
}

```

6. Converta o programa abaixo escrito em linguagem de montagem para pseudo-código ou C. Indique o conteúdo dos registradores 0, 1, 2, 3 e 4 ao final da execução deste programa, supondo que ele seja carregado a partir do endereço 0 da memória.

```

                lw 0 1 endar
                lw 0 2 tamar
                add 1 2 3
L1             beq 1 3 L2
                lw 1 4 0
                add 4 4 4
                sw 1 4 0
                addi 1 1 1
                beq 0 0 L1
L2             halt
endar         .fill array
tamar        .fill 4
array        .fill -5
array1       .fill -3
array2       .fill 4
array3       .fill 9

```

7. Suponha que um programa escrito na linguagem de montagem vista em aula foi convertido para linguagem de máquina e esteja carregado a partir do endereço de memória 10 até o endereço 1A, inclusive. O conteúdo da memória se encontra a seguir (endereços e conteúdos expressos em hexadecimal):

End.	Conteúdo	End.	Conteúdo
10	00000001	1B	F0000000
11	00420004	1C	00000000
12	010A0007	1D	00010000
13	00190004	1E	10000000
14	00A50000	1F	7FFF0000
15	002D0005	20	00000002
16	002D0005	21	FFFFFFFF
17	00E50000	22	0000000F
18	00490001	23	00000003
19	0100FFF8	24	00000010
1A	01800000	25	000000AF

- a) Mostre o código em linguagem de montagem
b) Suponha que os registradores, no início da execução deste programa, apresentem o conteúdo abaixo:
- ```

0-00000000
1-FFF01AD0
2-00012789
3-00000020

```

4-00010FDE

5-10100000

6-AC012345

7-12345678

Indique o conteúdo dos registradores e das posições de memória nos endereços 10 a 25, após a execução deste programa.