



**Universidade de Brasília**

Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A  
Prof. Marcus Vinicius Lamar

2013/2

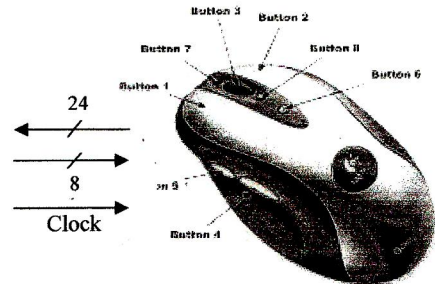
d0 d1 / d2 d3 d4 d5 d6 d7 d8

Nome: GABARITO

Matrícula:   /

## Prova 2

1)(7.0) O mouse, inventado em 1963 e patenteado em 1967 por Douglas Engelbart, tornou-se um dos dispositivos para interação humano computador mais populares. Os modernos sistemas operacionais Windows e Linux permitem a utilização simultânea de até 255 desses dispositivos. Projete uma interface que permita o uso “simultâneo” de até 256 mouses no processador MIPS Multiciclo desenvolvido em aula. Cada mouse registra as variações nas direções X, Y, com resolução de 8 bits cada em complemento de 2, e os cliques de 8 botões individuais (8 bits, 1 pressionado e 0 solto), de acordo com a tabela abaixo. Valores positivos indicam movimentos para direita ou para baixo, e valores negativos movimentos para a esquerda ou para cima. O protocolo de comunicação é muito simples e a interface física é composta de um conector de 32 pinos de dados e um de clock, que recebe 8 bits com o número do mouse e, após 2 ciclos de clock do processador, retorna os 24 bits de dados:



bits	23-16	15-8	7-0
Mouse	Botões	Desloc Y	Desloc X

- (2.0) Modifique adequadamente o Caminho de Dados e o Bloco Controlador do MIPS Multiciclo fornecido em anexo de forma a implementar a instrução: `mouse $rt, $rs. # Imm=0 Tipo-I OPCODE 0x2C` que coloca no registrador \$rt da CPU MIPS a word correspondente às informações fornecidas pelo mouse número \$rs.
- (2.0) Modifique adequadamente o Caminho de Dados e o Bloco Controlador do MIPS Multiciclo fornecido em anexo de forma a detectar a instrução `syscall`, executando-a similarmente à instrução `jal 0x0800 0000`.
- (2.0) Modifique adequadamente o Caminho de Dados e o Bloco Controlador do MIPS Multiciclo fornecido em anexo de forma a detectar a instrução `eret`, executando-a similarmente à instrução `jr $ra`.
- (1.0) Escreva um programa exemplo em Assembly MIPS que seja executável nesse processador modificado, e apresente continuamente na tela as informações dos mouses 0 a 3, de acordo com a formatação:

“Mouse:%d \t X:%d \t Y:%d \t Botoes: b<sub>7</sub>b<sub>6</sub>b<sub>5</sub>b<sub>4</sub>b<sub>3</sub>b<sub>2</sub>b<sub>1</sub>b<sub>0</sub>\n” ex.: Mouse:0 X:10 Y:-10 Botoes:00000001

Dica: Considere que as rotinas que processam as chamadas `syscalls` padrões do MARS já estão implementadas no segmento .text localizado a partir do endereço 0x08000 0000.

2) (4.0) Dada uma estrutura hierárquica de N níveis de memória cache de dados. Cada nível n possui um tempo de acesso dado por  $ta(n, N)$ , e é capaz de obter uma taxa de falhas (%)  $tf(n)$  com  $bits(n)$  de memória. O custo por bit da memória de nível n é de dada por  $custo(n)$ . O sistema possui 4GiB de memória RAM DDR3 com tempo de acesso de 800 ciclos de clock.

$$ta(n, N) = 2 + 3n^N \text{ [ciclos]} \quad tf(n) = \frac{10.2^{10}}{bits(n)} \text{ [%]} \quad custo(n) = \frac{e^{-n^2}}{120} \text{ [R$/bit]}$$

Considere que a taxa de falhas na memória cachê de instruções seja 0% e um workload composto de 1.000.000 de instruções, onde apenas 25% acessam a memória de dados.

- (1.0) Calcule o tempo de execução do workload por um processador pipeline ideal com frequência de clock de 2.8GHz que não use qualquer tipo de memória cache.
- (1.0) Qual o tamanho, em bits, que uma memória cache de dados de apenas 1 nível deve ter para que o acesso aos dados fique 10 vezes mais rápido? Qual o custo em R\$ dessa memória?
- (2.0) Defina o número de níveis N e o tamanho em KiB que cada nível da estrutura da memória cache de dados deve ter de modo a reduzir o tempo a 1% do tempo de acesso original, com um aumento do custo do sistema de no máximo R\$200,00.

**Boa Sorte!!!**





OAC-A

2013/2

## 2ª Prova Gabarito

1) a) na folha

b)

c)

d) implementar o APP i considerando

• DATA

→ Endereço: 0x0001000

MOUSE: .asciiz "mouse"

XS: .asciiz "X"

YS: .asciiz "Y"

BOT: .asciiz "BOTões"

TAB: .asciiz "\t"

NL: .asciiz "\n"

↓ caso  
LABELS de 16 bits!  
uso oddi ao  
invés de Lq.

considerando

• TEXT → Endereço: 0x00000000

AVALIAR apenas a utilização dos  
PRGIMAGENS -

2) a)  $t = I \times CPT \times T$  MEMÓRIA MEM INSTRUÇÕES!

$$t = 1.000.000 (0,25 \times 800 + 800) / 2.8G$$

$$t = 0,35714 \text{ seg}$$

b) 10x mais rápido c/ cache L1

$$t_f \times 800 + 5 = \frac{800}{10}$$

$$t_a = 2 + 3 \cdot 1 = 5 \text{ ciclos}$$

tempo acesso L1

$$t_f = \frac{80 - 5}{800} = 0,09375 \%$$

Logo:  $bits = \frac{10 \times 2^{10}}{0,09375} = 109,227 \times 10^3 \text{ bits}$

$$L = 13,33 \text{ KiB}$$

$$\text{Custo} = \frac{1}{120} \times 109,227K = R\$ 334,83$$

c) 100x mais rápido c/ cache L1

O menor tempo de acesso é  $n=1$  ✓  
independente mente do n° níveis N

Logo a estrutura rápida é com apenas 1 nível  
se p/ 10x mais rápido já é R\\$ 334,83  
Logo p/ reduzir ainda mais o custo  $\text{custo} \propto \frac{1}{t}$

$$t_f \cdot 800 + 5 = \frac{800}{100} \rightarrow t_f = 0,00375\%$$

$$bits = \frac{10 \times 2^{10}}{0,00375} = 273 \times 10^6 \text{ bits} \rightarrow 333,3 \text{ KiB}$$

$$\text{Custo} = \frac{1}{120} \times 273 \times 10^6 = R\$ 8.371,3$$

Logo B N / R\\$ 200,00 p/ 100x + rápido