



Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Disciplina: *Organização e Arquitetura de Computadores I*
Profa. *Joseana Macêdo Fachine Régis de Araújo*

LISTA DE EXERCÍCIOS FINAL
(Simulador MIPS, Pipeline e Memória Cache)

Matrícula(s):

Nome(s):

Nota(s):

Obs.: Esta atividade deverá ser realizada por uma equipe de 02 (dois) alunos.

ATIVIDADE PRÁTICA – SIMULAÇÃO DE UMA MÁQUINA MIPS, PIPELINE E MEMÓRIA CACHE

Objetivos da Atividade: Conhecer um simulador para uma Máquina MIPS, o comportamento de um processador com implementação usando pipeline e analisar o funcionamento da Memória Cache (Mapeamento Direto e Mapeamento Associativo).

SIMULADOR:

- Download ([clique aqui](#))
- Tutoriais:
 - [Simulador MIPS - Tutorial](#)
 - Informações complementares para instalação estão disponíveis [aqui](#).
 - Tutorial: [MipsIT Integrated Development Environment and the Mips Simulator](#)
 - **Todo o material necessário está disponível [aqui](#).**
- Simuladores que implementam o MIPS com pipeline:
 - MipsPipeS.exe (emula um processador com pipeline simples sem resolução de conflitos por hardware);
 - MipsPipeXL.exe (emula um processador implementado em pipeline que possui unidade de adiantamento).
 - **IMPORTANTE:** Os simuladores de Pipeline estão na pasta MipsIt.

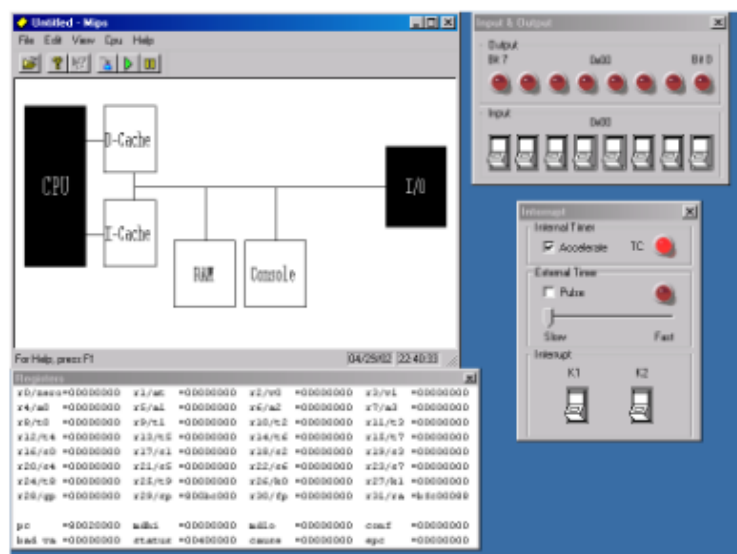


Figure . The system simulator with CPU register and I/O-device windows open.

Address	Content	Label
01200000	0F 87 02 4C	...
01200004	0F 85 01 0E	...
01200008	0F 83 00 30	...
0120000C	0F 81 00 40	...
01200010	0F 7F 00 50	...
01200014	0F 7D 00 60	...
01200018	0F 7B 00 70	...
0120001C	0F 79 00 80	...
01200020	0F 77 00 90	...
01200024	0F 75 00 A0	...
01200028	0F 73 00 B0	...
0120002C	0F 71 00 C0	...
01200030	0F 6F 00 D0	...
01200034	0F 6D 00 E0	...
01200038	0F 6B 00 F0	...
0120003C	0F 69 00 00	...
01200040	0F 67 00 10	...
01200044	0F 65 00 20	...
01200048	0F 63 00 30	...
0120004C	0F 61 00 40	...
01200050	0F 5F 00 50	...
01200054	0F 5D 00 60	...
01200058	0F 5B 00 70	...
0120005C	0F 59 00 80	...
01200060	0F 57 00 90	...
01200064	0F 55 00 A0	...
01200068	0F 53 00 B0	...
0120006C	0F 51 00 C0	...
01200070	0F 4F 00 D0	...
01200074	0F 4D 00 E0	...
01200078	0F 4B 00 F0	...
0120007C	0F 49 00 00	...
01200080	0F 47 00 10	...
01200084	0F 45 00 20	...
01200088	0F 43 00 30	...
0120008C	0F 41 00 40	...
01200090	0F 3F 00 50	...
01200094	0F 3D 00 60	...
01200098	0F 3B 00 70	...
0120009C	0F 39 00 80	...
012000A0	0F 37 00 90	...
012000A4	0F 35 00 A0	...
012000A8	0F 33 00 B0	...
012000AC	0F 31 00 C0	...
012000B0	0F 2F 00 D0	...
012000B4	0F 2D 00 E0	...
012000B8	0F 2B 00 F0	...
012000BC	0F 29 00 00	...
012000C0	0F 27 00 10	...
012000C4	0F 25 00 20	...
012000C8	0F 23 00 30	...
012000CC	0F 21 00 40	...
012000D0	0F 1F 00 50	...
012000D4	0F 1D 00 60	...
012000D8	0F 1B 00 70	...
012000DC	0F 19 00 80	...
012000E0	0F 17 00 90	...
012000E4	0F 15 00 A0	...
012000E8	0F 13 00 B0	...
012000EC	0F 11 00 C0	...
012000F0	0F 0F 00 D0	...
012000F4	0F 0D 00 E0	...
012000F8	0F 0B 00 F0	...
012000FC	0F 09 00 00	...

Figure . The memory view in the simulator.

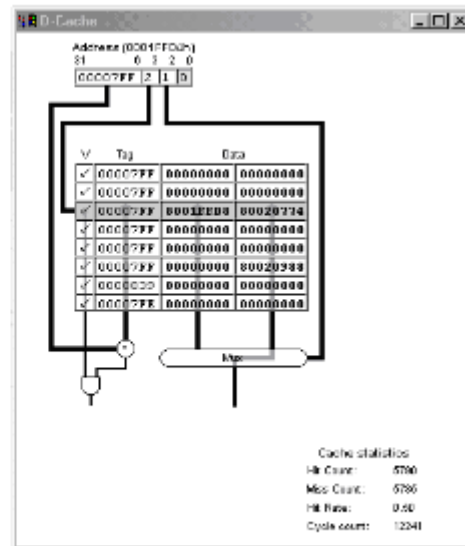


Figure . The animated cache view.

ATIVIDADE 1: Avaliar o funcionamento do simulador MipsIt para executar programas escritos em linguagem de máquina do MIPS. Instruções de uso disponíveis em [MipsIt](#). Uma vez que o código foi carregado no simulador, a janela de simulação é aberta e é possível verificar os registradores (clcando no desenho da CPU) ou a Memória (clcando no desenho RAM). Responder as questões a seguir (capturar telas para facilitar a explicação de cada item).

- Qual o número de registradores? Explicar a função de cada registrador.
- Qual o tamanho do dado a ser processado?
- Qual o tamanho das instruções?
- Clicar apenas na CPU e na RAM e descrever o que é possível observar em cada um destes módulos.
- Executar o programa passo a passo e descrever o que faz cada instrução e o que faz o programa e informar o conteúdo final dos registradores envolvidos.

```
.data
a:    .word 10
b:    .word 5
c:    .word 5
      .text
      .globl start
      .ent start
start: lw    $8, a
      lw    $9, b
      lw    $10, c
      add   $11, $10, $9
      sub   $11, $11, $8
      sw    $11, a
      .end start
```

ATIVIDADE 2: Criar um novo projeto e digitar o programa abaixo. Carregar o programa no simulador (MipsPipeS.exe).

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start

start: add t2, t0, t1
      nop
      nop
      nop

      .end start
```

- Iniciar t0 e t1 com os valores 11 e 6 (diretamente na janela que mostra os registradores) e executar o programa passo a passo. Descrever o que acontece quando as instruções entram em cada estágio do pipeline (IF, ID, EX, MEM, WB)? Descrever todos os sinais, as mudanças nos registradores e outros efeitos (de forma detalhada).
- Quantos ciclos são necessários para execução da instrução add t2, t0, t1. Qual o tempo necessário para execução desta instrução?
- Quantos ciclos são necessários para obtenção do resultado da operação no registrador?
- Qual o tempo total necessário para execução do programa?

ATIVIDADE 3: Executar o programa abaixo atribuindo os valores 11, 6 e 7 aos registradores t0, t1 e t4, respectivamente (editar os valores diretamente na janela dos registradores).

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start

start: add t2, t0, t1
      sub t3, t2, t4
      nop
      nop
      nop

      .end start
```

- Após quantos ciclos o registrador t2 recebe o resultado correto?
- Após quantos ciclos este valor é necessário na segunda instrução?
- Qual o problema neste programa e como esse problema pode ser resolvido?
- Executar o mesmo programa usando o simulador MipsPipeXL.exe.
- O pipeline executa corretamente? Justifique sua resposta.

ATIVIDADE 4: Avaliar o funcionamento do simulador conforme instruções disponíveis em [MipsIT Integrated Development Environment and the Mips Simulator](#). Para tanto, deverão ser executados os programas [strcpy.c](#), e [matriz.c](#) (informações complementares, clique [aqui](#) e [aqui](#)). Para cada programa, responder as questões a seguir.

- Descrever o que faz o programa;
- Para cada execução, informar o que está contido no Módulo CPU;
- Para cada execução, informar o que está contido no Módulo Memória Principal (RAM);
- Na primeira execução, desabilitar o uso do Módulo Memória Cache, o qual é composto da Cache de Dados (D-Cache) e da Cache de Instruções (I-Cache) e informar o que é possível observar;
- Na segunda execução, habilitar o uso do Módulo Memória Cache, o qual é composto da Cache de Dados (D-Cache) e da Cache de Instruções (I-Cache) e informar o que é possível observar.

ATIVIDADE 5: O programa em C abaixo contém duas subrotinas as quais retornam a soma de todos elementos da matriz. A única diferença entre as duas subrotinas é que elas visitam os elementos da matriz em uma ordem diferente. Este fato não parece ser importante, entretanto com o uso da memória cache, a diferença pode ser substancial.

- Analisar o código cuidadosamente de forma a entender em qual ordem os elementos da matriz são chamados. Explicar como este posicionamento se reflete na Memória Principal ou física.
- O que ocorre quando há uma falta na cache?
- O que ocorre quando há um acerto na cache?
- Qual a função da tag ou rótulo de memória?

```

/*
 * Laboratory Exercise 8, Home Assignment 1
 * Written by Jan Eric Larsson, 24 February 1999
 *
 */
#include <stdio.h>
#include <idt_entrypt.h>
#define N 16

int A[N][N];

int SumByColRow (int Matrix[N][N])
{
    int i, j, Sum = 0, Time;

    timer_start();
    for (j = 0; j < N; j++) {
        for (i = 0; i < N; i++) {
            Sum += Matrix[i][j];
        }
    }
    Time = timer_stop();
    printf("SumByColRow time: %d\n", Time);
    return Sum;
}

```

```

int SumByRowCol (int Matrix[N][N])
{
    int i, j, Sum = 0, Time;

    flush_cache();
    timer_start();
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            Sum += Matrix[i][j];
        }
    }

    Time = timer_stop();
    printf("SumByRowCol time: %d\n", Time);
    return Sum;
}

main ()
{
    int a, b;

    printf ("Lista de Exercícios de Cache:\n");
    a = SumByColRow (A);
    b = SumByRowCol (A);
    printf ("Resultados das somas %d e %d\n", a, b);
}

```

Fontes:

1. **Usando o simulador MIPS.**

Disponível em http://www.cin.ufpe.br/~if674/index_arquivos/2012_02/tutorialMipsIt.pdf. Último acesso em 25 de julho de 2018.

2. **IS1200 Datorteknik - Getting started with the Cache Exercise.**

Disponível em <https://www.kth.se/social/course/IS1500/page/labs-56/>. Último acesso em 12 de dezembro de 2018.

3. **MipsIt-A Simulation and Development Environment Using Animation.**

Disponível em <http://www.docstoc.com/docs/15480715/MipsIt%E2%80%94A-Simulation-and-Development-Environment-Using-Animation> . Último acesso em 12 de dezembro de 2018.

4. **Lista de Exercício – Memória Cache – Infra-Estrutura de Hardware.**

Disponível em www.cin.ufpe.br/~rosf/public_html/ListaCache-04-2.doc . Último acesso em 12 de dezembro de 2018.

5. **Infra-Estrutura de Hardware - IF674, CIn, UFPE.**

Disponível <<http://www.cin.ufpe.br/~if674>>. Último acesso em 12 de dezembro de 2018.

6. **Laboratory Exercise 1 Introduction to MipsIt Studio 2000.**

Disponível em <<https://www.coursehero.com/file/6940575/LearningMaterialICT4v51/>>. Último acesso em 12 de dezembro de 2018.

7. **Software.** Disponível em

<http://cial.csie.ncku.edu.tw/course/2005_Spring_Computer_Organization/CD/Content/Software/index.html#MipsIt>. Último acesso em 12 de dezembro de 2018.