 <p>Departamento de Sistemas e Computação</p>	<p>Universidade Federal de Campina Grande Departamento de Sistemas e Computação Disciplina: <i>Organização e Arquitetura de Computadores I</i> Profa. Joseana Macêdo Fecchine Régis de Araújo</p> <p style="text-align: center;">PROJETO FINAL (Simulador MIPS, Pipeline e Memória Cache)</p>	
Matrícula(s):	Nome(s):	Nota(s):

Obs.: Esta atividade deverá ser realizada por uma equipe de 05 (cinco) 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](#), atualizado por Fábio Fernando de Oliveira Silva, aluno do Curso de Ciência da Computação/UFCG, obrigada!).
- 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.

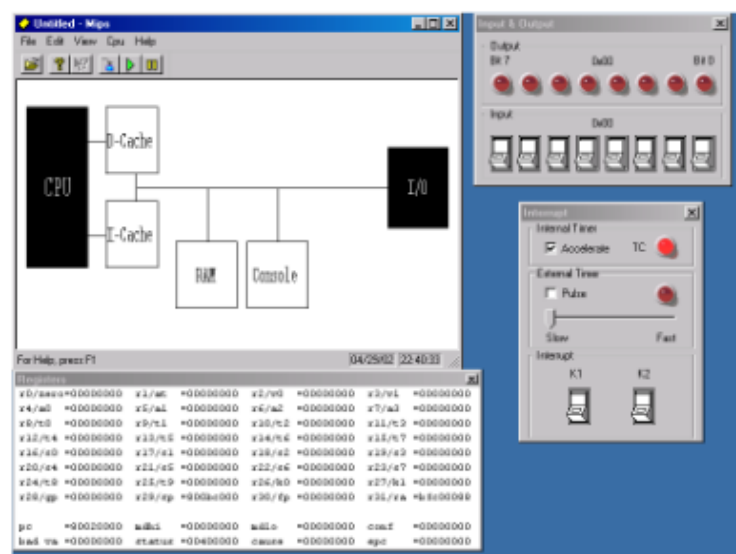


Figure . The system simulator with CPU register and I/O-device windows open.

Address	Content	Label
00100000	0F 8F 02 4C	lw \$9, 0x4C(00100000)
00100004	0F 8F 02 4C	lw \$9, 0x4C(00100004)
00100008	0F 8F 02 4C	lw \$9, 0x4C(00100008)
0010000C	0F 8F 02 4C	lw \$9, 0x4C(0010000C)
00100010	0F 8F 02 4C	lw \$9, 0x4C(00100010)
00100014	0F 8F 02 4C	lw \$9, 0x4C(00100014)
00100018	0F 8F 02 4C	lw \$9, 0x4C(00100018)
0010001C	0F 8F 02 4C	lw \$9, 0x4C(0010001C)
00100020	0F 8F 02 4C	lw \$9, 0x4C(00100020)
00100024	0F 8F 02 4C	lw \$9, 0x4C(00100024)
00100028	0F 8F 02 4C	lw \$9, 0x4C(00100028)
0010002C	0F 8F 02 4C	lw \$9, 0x4C(0010002C)
00100030	0F 8F 02 4C	lw \$9, 0x4C(00100030)
00100034	0F 8F 02 4C	lw \$9, 0x4C(00100034)
00100038	0F 8F 02 4C	lw \$9, 0x4C(00100038)
0010003C	0F 8F 02 4C	lw \$9, 0x4C(0010003C)
00100040	0F 8F 02 4C	lw \$9, 0x4C(00100040)
00100044	0F 8F 02 4C	lw \$9, 0x4C(00100044)
00100048	0F 8F 02 4C	lw \$9, 0x4C(00100048)
0010004C	0F 8F 02 4C	lw \$9, 0x4C(0010004C)
00100050	0F 8F 02 4C	lw \$9, 0x4C(00100050)
00100054	0F 8F 02 4C	lw \$9, 0x4C(00100054)
00100058	0F 8F 02 4C	lw \$9, 0x4C(00100058)
0010005C	0F 8F 02 4C	lw \$9, 0x4C(0010005C)
00100060	0F 8F 02 4C	lw \$9, 0x4C(00100060)
00100064	0F 8F 02 4C	lw \$9, 0x4C(00100064)
00100068	0F 8F 02 4C	lw \$9, 0x4C(00100068)
0010006C	0F 8F 02 4C	lw \$9, 0x4C(0010006C)
00100070	0F 8F 02 4C	lw \$9, 0x4C(00100070)
00100074	0F 8F 02 4C	lw \$9, 0x4C(00100074)
00100078	0F 8F 02 4C	lw \$9, 0x4C(00100078)
0010007C	0F 8F 02 4C	lw \$9, 0x4C(0010007C)
00100080	0F 8F 02 4C	lw \$9, 0x4C(00100080)
00100084	0F 8F 02 4C	lw \$9, 0x4C(00100084)
00100088	0F 8F 02 4C	lw \$9, 0x4C(00100088)
0010008C	0F 8F 02 4C	lw \$9, 0x4C(0010008C)
00100090	0F 8F 02 4C	lw \$9, 0x4C(00100090)
00100094	0F 8F 02 4C	lw \$9, 0x4C(00100094)
00100098	0F 8F 02 4C	lw \$9, 0x4C(00100098)
0010009C	0F 8F 02 4C	lw \$9, 0x4C(0010009C)
001000A0	0F 8F 02 4C	lw \$9, 0x4C(001000A0)
001000A4	0F 8F 02 4C	lw \$9, 0x4C(001000A4)
001000A8	0F 8F 02 4C	lw \$9, 0x4C(001000A8)
001000AC	0F 8F 02 4C	lw \$9, 0x4C(001000AC)
001000B0	0F 8F 02 4C	lw \$9, 0x4C(001000B0)
001000B4	0F 8F 02 4C	lw \$9, 0x4C(001000B4)
001000B8	0F 8F 02 4C	lw \$9, 0x4C(001000B8)
001000BC	0F 8F 02 4C	lw \$9, 0x4C(001000BC)
001000C0	0F 8F 02 4C	lw \$9, 0x4C(001000C0)
001000C4	0F 8F 02 4C	lw \$9, 0x4C(001000C4)
001000C8	0F 8F 02 4C	lw \$9, 0x4C(001000C8)
001000CC	0F 8F 02 4C	lw \$9, 0x4C(001000CC)
001000D0	0F 8F 02 4C	lw \$9, 0x4C(001000D0)
001000D4	0F 8F 02 4C	lw \$9, 0x4C(001000D4)
001000D8	0F 8F 02 4C	lw \$9, 0x4C(001000D8)
001000DC	0F 8F 02 4C	lw \$9, 0x4C(001000DC)
001000E0	0F 8F 02 4C	lw \$9, 0x4C(001000E0)
001000E4	0F 8F 02 4C	lw \$9, 0x4C(001000E4)
001000E8	0F 8F 02 4C	lw \$9, 0x4C(001000E8)
001000EC	0F 8F 02 4C	lw \$9, 0x4C(001000EC)
001000F0	0F 8F 02 4C	lw \$9, 0x4C(001000F0)
001000F4	0F 8F 02 4C	lw \$9, 0x4C(001000F4)
001000F8	0F 8F 02 4C	lw \$9, 0x4C(001000F8)
001000FC	0F 8F 02 4C	lw \$9, 0x4C(001000FC)

Figure . The memory view in the simulator.

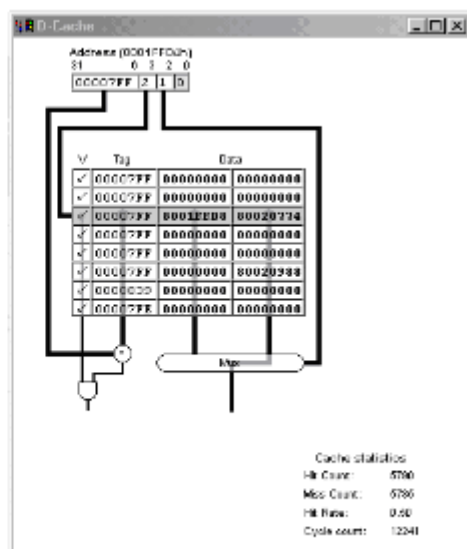


Figure . The animated cache view.

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?
- 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.
- Descrever o que faz o programa.

```
.data
a:    .word 5
b:    .word 10
c:    .word 5

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

ATIVIDADE 2: Crie um novo projeto e digite o programa abaixo. Carregue o programa no simulador (MipsPipeS.exe)

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

start:  sub  t2, t0, t1
        nop
        nop
        nop

        .end start
```

- Inicie t0 e t1 com valores distintos (diretamente na janela que mostra os registradores) e execute o programa passo a passo. Descreva o que acontece quando as instruções entram em cada estágio do pipeline (IF, ID, EX, MEM, WB)? Descreva todos os sinais, as mudanças nos registradores e outros efeitos em detalhes.
- Quantos ciclos são necessários para obtenção do resultado da operação no registrador?

ATIVIDADE 3: Execute o programa abaixo atribuindo valores distintos aos registradores t0, t1 e t4 (edite os valores diretamente na janela dos registradores).

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

start:  sub  t2, t0, t1
        add  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?
- Execute 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, responda 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.

```
/*
 * 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 ("As somas sao %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 10 de março 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 10 de março de 2018.

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

Disponível em <http://www.docstoc.com/docs/15480715/MipsIt%E2%80%94Simulation-and-Development-Environment-Using-Animation> . Último acesso em 24 de maio de 2016.

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 10 de março de 2018.

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

Disponível <<http://www.cin.ufpe.br/~if674>>. Último acesso em 10 de março de 2018.

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

Disponível em <<https://www.coursehero.com/file/6940575/LearningMaterialICT4v51/>>. Último acesso em 10 de março 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 10 de março de 2018.

SUCESSO!