



**UNIVERSIDADE FEDERAL DE ALAGOAS – UFAL**  
**INSTITUTO DE COMPUTAÇÃO - IC**

Disciplina: ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Curso: Engenharia e Ciência da computação

Docente: Erick de Andrade Barboza

### **Lista – Prática: Pipeline MIPS**

O objetivo desta lista será a utilização de simuladores que emulam o comportamento de um processador com implementação usando pipeline.

Serão utilizados dois simuladores que implementam o MIPS com pipeline:

- O simulador MipsPipeS.exe emula um processador com pipeline simples sem resolução de conflitos por hardware.
- O simulador MipsPipeXL.exe emula um processador implementado em pipeline que possui unidade de adiantamento (curto-circuito) e adiantamento do desvio.

Para a utilização destes simuladores é necessário baixar este [arquivo](#) e seguir este [tutorial](#) sobre a utilização do Mipisl. **ATENÇÃO: Existe uma incompatibilidade entre o simulador acima e o Windows 10. Portanto, quem utilizar este S.O. deverá criar uma máquina virtual com Windows 7-8.**

O efeito da utilização do pipeline na execução das 4 classes de instruções do MIPS: aritméticas, load, store e de desvio serão discutidos nas questões abaixo.

1. Crie um novo projeto e digite o programa abaixo e carregue o mesmo no simulador (MipsPipeS.exe)

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start
start:  add t0, t1, t2
        nop
        nop
        nop
        nop
.end start
```

- a. Inicie t1 e t2 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.
- b. Quantos ciclos são necessários para se ter o resultado da operação no registrador?



2. Substitua a instrução `add t0, t1, t2` no programa acima pela instrução `lw t0, 0(t1)`, inicializando o registrador `t1` com algum endereço de memória editando o conteúdo do registrador na janela de registradores. Coloque um valor inicial no endereço de memória colocado no registrador editando a janela da memória. Compile o programa e analise sua execução no pipeline.

- O que acontece em cada um dos estágios do pipeline?
- Quais operações são feitas na ALU?
- Quantos ciclos são necessários para que o registrador tenha o valor correto?
- Todos os estágios do pipeline são utilizados?
- Faça a mesma análise para a instrução `sw t0, 4(t1)`
- Repita o mesmo tipo de análise para a instrução `beq t0, t1, Dest`. Note que você deve inserir um rótulo no programa.

3. Execute o programa abaixo atribuindo valores distintos aos registradores `t0`, `t1` e `t3` (edite os valores diretamente na janela dos registradores).

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start
start:  add t2, t0, t1
        add t4, t2, t3
        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 ele pode ser resolvido?
- Execute o mesmo programa usando o simulador `MipsPipeXL.exe`.
- O pipeline executa corretamente? Porque?

4. Execute o programa abaixo no simulador (`MipsPipeS.exe`) atribuindo o mesmo valor aos registradores `t0` e `t1` (edite os valores diretamente na janela de registradores).

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start
start: nop
        nop
        beq t0, t1, start
        addi t0, t0, 1
```



```
    nop
    nop
    nop
.end start
```

- Quantos ciclos são necessários até que o desvio possa ser executado
- O que acontece com a instrução `addi` enquanto o `branch` é calculado?
- Quais os problemas e quais as soluções para resolve-lo?
- Use agora a versão `MipsPipeXL.exe`, como será a execução do programa?

5. Execute o programa abaixo e atribua diferentes valores para os registradores `t1`, `t2` e `t3` (edite os valores diretamente na janela de registradores). O registrador `t2` contém um endereço de memória.

```
#include <reg.h>
.set noreorder
.text
.globl start
.ent start
start:add t0, t1, t2
      lw t1, 0(t2)
      sub t2, t1, t3
      nop
      nop
      nop
.end start
```

- Após quantos ciclos o registrador `t1` recebe o valor correto?
- Em que ciclo este valor deveria estar disponível?
- Qual o problema neste caso e como o mesmo pode ser resolvido?
- Execute o programa usando o simulador `MipsPipeXL.exe`, qual foi o resultado? Porque o resultado difere?

6. Considere o programa em linguagem assembly do MIPS abaixo. Estude o programa para entender sua funcionalidade.

```
#include <reg.h>
.data
A: .word 5, 6, 2, 3, 1
.text
.globl start
.ent start
start:lui a0,0x8002
      ori a0, a0, 0x0090
      addi a1, zero,0x5
      xor t0,t0,t0
CW1:  beq t0,a1,EW1
```



```
        addi t1,t0,1
CW2:    beq t1, a1, EW2
        add t2, t0, t0
        add t2, t2, t2
        add t2, a0, t2
        add t3, t1, t1
        add t3, t3, t3
        add t3, a0, t3
        lw t4, 0(t2)
        nop
        lw t5, 0(t3)
        nop
        slt t6, t4, t5
        bne t6, zero, IGN
        beq t4, t5, IGN
        sw t5, 0(t2)
        sw t4, 0(t3)
IGN:    addi t1, t1, 1
        beq $0, $0, CW2
EW2:    addi t0, t0, 1
        beq $0, $0, CW1
EW1:    beq $0, $0, FIM
FIM:    nop
        break
.end start
```

- Descreva a funcionalidade do programa
- Calcule o tempo de execução (em ciclos de relógio) usando o simulador MIPSPipeS.exe. Neste caso, devem ser inseridos NOP's na ocorrência de conflito de dados e de controle. Após a inserção dos NOPs o valor da segunda instrução ori a0, a0, 0x0090 deve ser ajustado de forma que a0 possua o endereço inicial do vetor A.
- Calcule o tempo de execução do programa usando o simulador MIPSPipeXL.exe, retirando os NOPS que não são necessários. Após a retirada dos NOPs o valor imediato da segunda instrução deverá ser ajustado.
- Compare com o resultado da letra (a). Quais as técnicas que foram usadas para a resolução dos conflitos? Mostre no código quais as técnicas que foram utilizadas.
- Você consegue um tempo de execução menor? Como?