

- Projeto da Unidade II
- Valor: 4,0
- Atividade INDIVIDUAL
- Prazo de entrega: 30/04

Simulador de pipeline considerando conflitos de dados, estruturais e de controle

Descrição: Implementar uma ferramenta que simule a execução em pipeline 5 estágios do MIPS. O simulador recebe como entrada um conjunto de instruções em Assembly MIPS, simula a execução dessas instruções considerando todas as dependências de dados e conflitos estruturais e, tem como saída, a quantidade de ciclos necessários para executar essas instruções e a informação sobre os estágios em cada ciclo.

CONSIDERAÇÕES:

- 1) Considere um pipeline de 5 estágios na arquitetura Harvard onde os estágios:
 - a. IF: Busca a próxima instrução;
 - b. ID: Decodifica a instrução e busca operandos que são registradores;
 - c. EX: Executa a instrução;
 - d. MEM: leitura da memória;
 - e. WB: salva na memória ou no registrador.
- 2) O programa recebe um código escrito em assembly e montará, ao final, a sua execução no pipeline ciclo por ciclo. Não é obrigatório que o programa faça a análise dos conflitos em tempo de execução, ele pode processar todo o código e em seguida montar/imprimir o pipeline, conforme exemplo na frente.
- 3) As instruções utilizadas serão ADD, SUB, BEQ, BNE, LW, SW e JUMP. Não se preocupe com as demais instruções.
- 4) Não se preocupe com a execução completa das instruções, i.e., os registradores das instruções são apenas referências para que você saiba onde tem conflito e que possa resolver tal conflito. EX: Em “add \$t0, \$t1, \$t2” não é preciso carregar os valores dos registradores \$t1 e \$t2 nem salvar em \$t0. Apenas utilize-os para montar a execução ciclo a ciclo do pipeline.
- 5) **Não** deve ser realizada reordenação
- 6) Considere que a arquitetura **não** tem redirecionamento (forwarding)
- 7) Quanto aos conflitos, considere:

Conflitos de dados:

A ferramenta deve ser capaz de detectar as dependências de dados e parar o pipeline até que a dependência seja resolvida. Exemplo:

1. add \$t0, \$t1, \$t2	1	IF	ID	EX	MEM	WB				
2. sub \$s0, \$t0, \$t1	2					IF	ID	EX	MEM	WB

Conflitos estruturais:

Para considerar os conflitos estruturais, a ferramenta deve assumir **sempre** uma arquitetura harvard. Ou seja, não haverão casos de conflitos por memória. Considere apenas a disputa pelo **estágio** do pipeline.

Conflitos de controle:

Assuma que os saltos serão **SEMPRE** tomados. Nesse caso, ao encontrar um salto, é preciso verificar qual o Label e preencher o pipeline com a instruções a partir do label.

1. beq \$s0, \$s1, DENTRO	1	IF	ID	EX	ME M	WB								
2. add \$t1, \$t2, \$t3														
3. lw \$t0, 128(\$t1)	4		IF	ID	EX	ME M	WB							
DENTRO:														
4. sub \$t1, \$t2, \$t3	5							IF	ID	EX	ME M	WB		
5. add \$s5, \$t1, \$t4														

Saída da Ferramenta:

A saída da ferramenta deve mostrar:

1. A quantidade de ciclos total para executar as instruções
2. O estágio que cada instrução se encontra por ciclo.

Exemplo de funcionamento da ferramenta:

Entrada:

add \$t0, \$t1, \$t2

sub \$s0, \$t0, \$t1

Saída:

Quantidade de ciclos total: 9

----- Ciclo 1

IF: add \$t0, \$t1, \$t2

ID: 0

EX: 0

MEM: 0

WB: 0

----- Ciclo 2

IF: 0

ID: add \$t0, \$t1, \$t2

EX: 0

MEM: 0

WB: 0

----- Ciclo 3

IF: 0

ID: 0

EX: add \$t0, \$t1, \$t2

MEM: 0

WB: 0

----- Ciclo 4

IF: 0

ID: 0

EX: 0

MEM: add \$t0, \$t1, \$t2

WB: 0

----- Ciclo 5

IF: sub \$s0, \$t0, \$t1

ID: 0

EX: 0

MEM: 0

WB: add \$t0, \$t1, \$t2

Ciclo 6

IF: 0

ID: sub \$s0, \$t0, \$t1

EX: 0

MEM: 0

WB: 0

Ciclo 7

IF: 0

ID: 0

EX: sub \$s0, \$t0, \$t1

MEM: 0

WB: 0

Ciclo 8

IF: 0

ID: 0

EX: 0

MEM: sub \$s0, \$t0, \$t1

WB: 0

Ciclo 9

IF: 0

ID: 0

EX: 0

MEM: 0

WB: sub \$s0, \$t0, \$t1

Exemplo em sala:

```
1. add    $t0,    $zero,    $v0
2. add    $t1,    $t0,     $t2
3. sub    $s0,    $t3,     $t1
4. lw     $t4,    4($s0)
5. beq    $s0,    $t4,     DENTRO
6. add    $t1,    $t2,     $t3
7. lw     $t0,    128($t1)
8. j      EXIT
```

DENTRO:

```
9. sw     $t8,    0($s8)
10. sub   $t1,    $t2,     $t3
11. lw    $t0,    8($s1)
```

EXIT:

```
12. ...
```

Como será testado:

Todas as ferramentas serão testadas com um caso de teste criado especificamente para avaliar a corretude.

Se não funcionar ou a contagem de ciclos ou a exibição estiver errada, a nota será diminuída **consideravelmente (tende a zero)**.