

Exercício

1. Execute o trecho de código abaixo colocando todos os estágios do pipeline (B, D, E, M, W). Caso um estágio não tenha sua unidade principal acessada, represente por '-'. Bolhas podem ser representadas por 'X' ou 'XK' onde K significa o estágio onde instrução está parada. Considere que, caso haja uma dependência, esta deve ser resolvida com bolhas

| Instrução | Ciclo |
|----------------------------|---|
| | 1 2 3 4 5 6 7 8 9 10 11 12 |
| lui \$s1, 0 | |
| addi \$s1, \$s1, 2 | |
| lui \$s2, 0 | |
| sw \$s1, (0x11000000) \$s3 | |
| sw \$s2, (0x11000000) \$s3 | |

Exercício

2. (Baseado no POSCOMP 2005 - 21) Considere uma CPU usando pipeline com 5 estágios (B, D, E, M, W), com memórias de dados e de instruções separadas, escrita no banco de registradores na borda de subida do relógio e leitura na borda de descida do relógio e o conjunto de instruções a seguir:
- I1: lw \$2, 100(\$5)
I2: add \$1, \$2, \$3
I3: sub \$3, \$2, \$1
I4: sw \$2, 50(\$1)
I5: add \$2, \$3, \$3
I6: sub \$2, \$2, \$4
- Quantos ciclos são gastos para a execução deste código?
- a) 30
b) 17
c) 16
d) 11
e) 10

[illegible]

Exercício

3. (POSCOMP 2008 - 54) Um processador tem cinco estágios de pipeline. Suponha que cada uma das etapas do processador (busca, decodificação, execução, leitura ou escrita de dados em memória e escrita em registrador) seja executada em 5ns. O tempo total para que 5 instruções sejam executadas em pipeline, supondo que não haja dependência de dados entre as instruções é:
- a) 15ns
 - b) 25ns
 - c) 30ns
 - d) 45ns
 - e) 50ns

Exercício

4. Ilustre a execução do trecho de código no MIPS pipeline, colocando os estágios e bolhas e flush necessários para sua execução correta. Considere que o código inicia com `$t1 = 3`, `$s0 = &vet[0]` e `$t2 = 4`. Considere também que `vet` é um vetor de 4 inteiros iniciados com todas as posições em 0 e endereço de memória 0x100. Ao final apresente o conteúdo de `vet` e o valor dos registradores utilizados. OBS. Coloque F em um estágio que está em flush

| Instrução | Ciclo | | | | | | | | | |
|-------------------------|-------|---|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| ini: addi \$t1, \$t1, 1 | | | | | | | | | | |
| bgt \$t1, \$t2, fim | | | | | | | | | | |
| sw \$t1, 0(\$s0) | | | | | | | | | | |
| addi \$s0, \$s0, 4 | | | | | | | | | | |
| j ini | | | | | | | | | | |
| fim: li \$v0, 10 | | | | | | | | | | |
| syscall | | | | | | | | | | |

vet[] = { , , , } \$t1 = \$t2 = \$s0 = \$v0 =

Exercício

5. Um diagrama que relaciona ciclos de instrução com estágios do pipeline, tal como ilustrado na direita, permite compreender melhor a execução em um pipeline

- Reescreva a execução abaixo no formato proposto
- Diga quantas bolhas foram inseridas devido aos dados e devido ao controle
- Quantos ciclos a mais ocorreram por ter hazards de controle e dados?
- Qual a fórmula básica do pipeline sem considerar hazards? Entenda e comente o efeito das bolhas de dados e de controle na fórmula



| | B | D | E | M | W |
|----|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |
| 19 | | | | | |
| 20 | | | | | |

| Instrução | Ciclo | | | | | | | | | | | | | | | | | | | |
|-------------------------|-------|---|---|----------------|----------------|---|---|---|---|----|----------------|----------------|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| ini: addi \$t1, \$t1, 1 | B | D | E | - | W | | | B | D | E | - | W | | | | | | | | |
| bgt \$t1, \$t2, fim | | B | D | X ^D | X ^D | E | - | - | B | D | X ^D | X ^D | E | - | - | | | | | |
| sw \$t1, 0(\$s0) | | | B | X ^B | X ^B | D | E | M | - | B | X ^B | X ^B | D | F | F | F | | | | |
| addi \$s0, \$s0, 4 | | | | | | B | D | E | - | W | | | B | F | F | F | F | | | |
| j ini | | | | | | | B | D | - | - | - | | | | | | | | | |
| fim: li \$v0, 10 | | | | | | | | B | F | F | F | F | | B | D | E | - | W | | |
| syscall | | | | | | | | | | | | | | | B | D | E | - | - | |

```
Bolhas(dados) = , Bolhas(controle) = , #ciclosAdicionais =
```

Fórmula =

Exercício

6. Dado o trecho de código abaixo em linguagem de montagem do MIPS, preencha o diagrama de execução de execução abaixo.

Convenção: X^k [bolha no estágio k], F [Flush], - [estágio sem operação], B [Busca], D [Decodificação], E [Execução], M [operação com a memória de dados], W [Write-back]

Conteúdos iniciais da memória e dos registradores relevantes:

\$t0=0x100, \$t2=0x000, \$t3=0x100, \$t4=0x000

```
Mem [0x100-0x103] = 0x00000001
```

```
Mem [0x200-0x203] = 0x00000010
```

```
Mem [0x300-0x303] = 0x000000100
Mem [0x400-0x403] = 0x000000001
```

```
Mem [0x400 0x403] = 0x00000001
Mem [0x4E0 0x4E3] = 0x00000010
```

[illegible][illegible]