

1) Considere a execução do seguinte código no caminho de dados pipeline :

```

Lw $4, 8($4)
Sw $5, 4($4)
Lw $5, $5, $4 Lw $5, 8($4)
Sw $5, 4($5)
Addi $4, $5, 4
Beq $4, $5, L1
L2: Bne $6, $4, L3
L1: J L2
L3: Lw $6, 8($4)
Add $6, $4, $6
Lw $6, 4($6)
And $4, $4, $6
Sw $6, 8($4)

```

B D Ex M Er
 B D D D Ex M Er
 B B B D Ex M Er
 B D D D

End:

Levando-se em conta as instruções acima e os seguintes tempos associados as unidades funcionais: memória 3 ns (leitura/escrita), alu 3 ns e registradores 1ns (leitura/escrita), e zero nanosegundos nos demais componentes.

- 1.1. Supondo que a Unidade de Adiantamento tenha sido implementada na fase de execução e que a Unidade de Detecção de Conflito tenha sido implementada na fase de decodificação e que os desvios condicionais ocorram. Esboce o diagrama estilizado do código acima com solução dos conflitos, de um modo explícito, objetivando um melhor desempenho possível numa arquitetura Não-Otimizada (2,5 pontos);
- 1.2. Supondo que a Unidade de Adiantamento tenha sido implementada na fase de decodificação e que a Unidade de Detecção de Conflito (Estendida) tenha sido implementada e que os desvios condicionais ocorram. Esboce o diagrama estilizado do código acima com solução dos conflitos, de um modo explícito, objetivando um melhor desempenho possível numa arquitetura Otimizada (2,5 pontos);
- 1.3. Supondo que a Unidade de Adiantamento tenha sido implementada na fase de memória e que a Unidade de Detecção de Conflito não seja implementada e que os desvios ocorram. Esboce o diagrama estilizado do código acima com solução dos conflitos, de um modo explícito, objetivando um melhor desempenho possível numa arquitetura Não-Otimizada (2,5 pontos);

2. Suponha que os sinais Aluop1 e Aluop0 tenham as seguintes combinações de bits:

Aluop1 Aluop0

- | | | |
|---|---|----------------------------------|
| 0 | 1 | → instruções de desvio |
| 1 | 0 | → instruções tipo-R |
| 1 | 1 | → Instruções de acesso a memória |

Supor ainda que os sinais de controle Aluop1, Branch e MemRead não tenham sido implementados na unidade de controle de uma arquitetura monociclo. É possível executar programas contendo instruções Tipo-R, Load (Lw), Store (Sw), desvio condicional (BEQ) e desvio Incondicional (J) nesta arquitetura por meio da utilização dos demais sinais de controle ? Sim ou Não? Fundamente a sua resposta com base na teoria estudada. (2,5 pontos)

9.50

Arquitetura de Computadores Alexandre Diego Santos Silva

12/12/16

① Memória - 3ns ALU - 3ns Registradores - 1ns

1.1. lw B D Ex M+Er ✓
 sw B D D#Ex M+Er ✓
 lw B B D Ex M+Er ✓
 sw B D D#Ex M+Er ✓
 addi B B D Ex M+Er ✓
 beg B D#Ex M+Er ✓
 bne/NOP B D#Ex 0 0 ✓
 j/NOP B D 0 0 0 0 ✓
 lw/NOP B 0 0 0 0 0 ✓
 lw/NOP B D Ex M+Er ✓
 bne B 0 0 0 0 0 ✓
 j/NOP B D Ex M+Er ✓
 lw/NOP B D Ex 0 0 0 ✓
 add/NOP B D 0 0 0 0 ✓
 lw B D Ex M+Er ✓
 add B D D#Ex M+Er ✓
 lw B B D#Ex M+Er ✓
 and B D D#Ex M+Er ✓
 sw B B D#Ex M+Er ✓

Repetindo o lw para facilitar visualização

$$\text{Tempo} = 28 \times 3 = 84\text{ns}$$

Repetindo o lw/NOP para continuação

lw/NOP B 0 0 0 0 ✓

bne B D Ex M Er ✓

j/NOP B D Ex 0 0 ✓

lw/NOP B 0 0 0 0 ✓

add/NOP B 0 0 0 0 ✓

lw B D Ex M Er ✓

NOP 0 0 0 0 0 ✓

NOP 0 0 0 0 0 ✓

add B D Ex M Er ✓

NOP 0 0 0 0 0 ✓

NOP 0 0 0 0 0 ✓

lw B D Ex M Er ✓

NOP 0 0 0 0 0 ✓

NOP 0 0 0 0 0 ✓

and B D Ex M Er ✓

sw B D Ex M Er ✓

$$\text{Tempo} = 35 \times 3 = 105 \text{ ns}$$

(2)	Reg Dst	Alu Src	Mem Read	Mem Write	Reg Write	Mem To Reg	Branch	DVI	Aluop 1	Aluop 0
Tipo-R	1	0	0	0	1	0	0	0	1	0
lw	0	1	1	0	1	1	0	0	1	1
sw	X	1	0	1	0	X	0	0	1	1
beg	X	0	0	0	0	X	1	0	0	1
j	X	X	0	0	0	X	X	1	0	1

Para que as instruções possam ser executadas os sinais que não foram implementados devem ser substituídos por outros que equivalem ao seu conjunto, logo buscando por equivalências, temos:

$$Alwops \equiv (MemWrite) \text{ or } (RegWrite)$$

1	1
1	1
1	1
0	0
0	0

O sinal "Alwops" pode ser obtido através da combinação dos sinais "MemWrite" e "RegWrite" através de uma porta lógica "or".

$$Branch \equiv NOT((MemWrite) \text{ or } (RegWrite) \text{ or } (DVS))$$

0	0
0	0
0	0
1	1
x	0

25

O sinal "Branch" pode ser obtido através da combinação dos sinais "MemWrite", "RegWrite" e "DVS" através de uma porta lógica "Nor".

$$MemRead \equiv (AlwSrc) \text{ and } (RegWrite)$$

0	0
1	1
0	0
0	0
0	0

Enfi da
acord. com
o enunciado.

O sinal "MemRead" pode ser obtido através da combinação dos sinais "AlwSrc" e "RegWrite" através de uma porta lógica "AND".

~~Como exemplo, vamos considerar as instruções ADD, SUB, MUL, DIV, AND, OR, XOR, NOT, LUI, SW, BEQ, D.~~

Logo, programas contendo as instruções determinadas (Tipo-R, lw, sw, beq, d) podem ser executados uma vez que os sinais Alwops, Branch e MemRead sejam trazidos por suas equivalências, conforme demonstrado acima.

Caso não fossem encontradas equivalências para todos os sinais, não seria possível a execução de tais programas.