

Lista de Exercício

Exercício 1

Considere o seguinte trecho de código do MIPS:

```
LW    $t1, 100($t0)
LW    $t2, 104($t0)
LW    $t3, 108($t0)
LW    $t4, 112($t0)
ADD   $S0, $t1, $t2
MUL   $S1, $t3, $t4
SUB   $A0, $S1, $S0
```

- (a) Traduza o código acima, de uma forma eficiente, para uma arquitetura do tipo acumulador. Deixe claro as considerações necessárias. Se desejar, apresente uma imagem ilustrativa desta máquina.
- (b) Apresente as vantagens e as desvantagens da arquitetura proposta no item (a) em comparação ao MIPS.
- (c) Traduza o código acima, de uma forma eficiente, para uma arquitetura do tipo pilha. Deixe claro as considerações necessárias. Se desejar, apresente uma imagem ilustrativa desta máquina.
- (d) Apresente as vantagens e as desvantagens da arquitetura proposta no item (c) em comparação ao MIPS.

Exercício 2

Considere o seguinte trecho de código:

```
0: ADD      R15, R2, R3
1: SUB      R1, R12, R16
2: ADDIU    R11, R10, 1
3: MUL      R5, R1, R4
4: MUL      R7, R5, R6
5: ADDIU    R18, R11, 1
6: ADDIU    R14, R18, 1
7: ADDIU    R13, R18, 2
8: SW       R5, 0(R14)
9: SW       R7, 0(R14)
```

- (a) Indique todas as dependências de dados.
- (b) Mostre o diagrama de pipeline para o trecho acima considerando o pipeline de 5 estágios do MIPS para instruções inteiras. A instrução MUL é executada pela própria ALU por 1 estágio. Considere todos os tipos de encaminhamento (forwarding).
- (c) Mostre o diagrama de pipeline para o trecho acima considerando o pipeline de 5 estágios do MIPS para instruções inteiras. A instrução MUL é executada por uma unidade funcional exclusiva por 4 estágios. Considere apenas uma ALU e uma unidade funcional de multiplicação. Considere todos os tipos de encaminhamento (forwarding).

Exercício 3

Considere o seguinte trecho de código:

0: ADD	R1, R2, R3
1: LW	R4, 100(R1)
2: SUB	R2, R5, R4
3: DIV	R5, R2, R1
4: ADD	R4, R1, R4
5: SUB	R5, R6, R7
6: MUL	R2, R5, R4
7: MUL	R6, R7, R8

(a) Indique todas as dependências de dados.

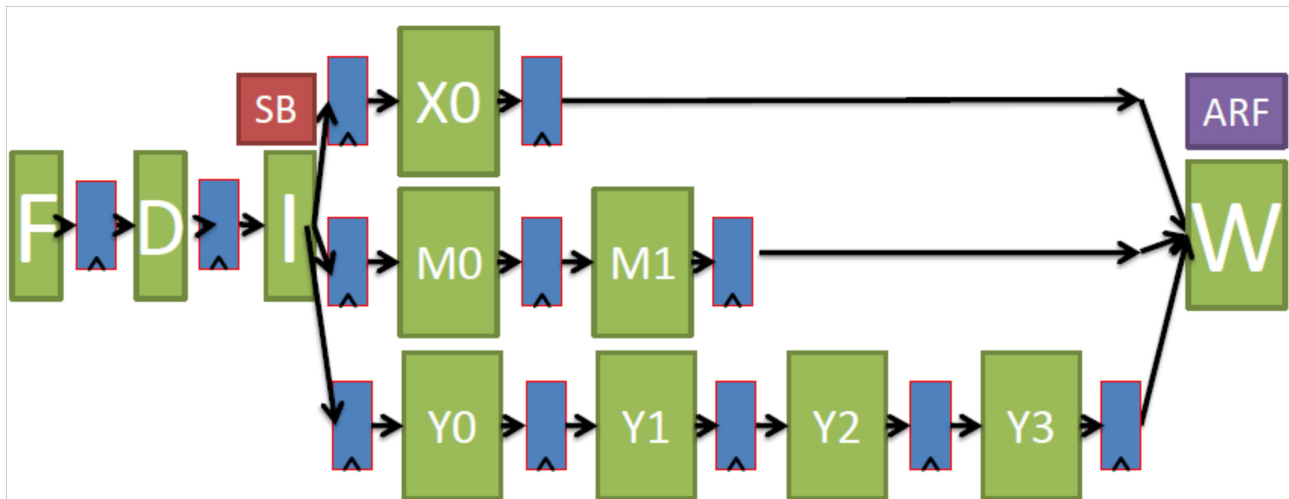
(b) Mostre o diagrama de pipeline para o trecho acima considerando o pipeline de 5 estágios do MIPS para instruções inteiras. As instruções MUL e DIV são executada pela própria ALU por 1 estágio. Considere todos os tipos de encaminhamento (forwarding).

(c) Mostre o diagrama de pipeline para o trecho acima considerando o pipeline de 5 estágios do MIPS para instruções inteiras. A instrução MUL é executada por uma unidade funcional exclusiva por 4 estágios, enquanto a instrução DIV é executada por uma unidade funcional exclusiva por 8 estágios. Considere apenas uma ALU e uma unidade funcional de multiplicação. Considere todos os tipos de encaminhamento (forwarding).

Exercício 4

Considere o processador I2O2 visto em sala e apresentado abaixo. Além disso, resolva todos os itens da questão considerando o seguinte código:

Processador I2O2



Código:

```

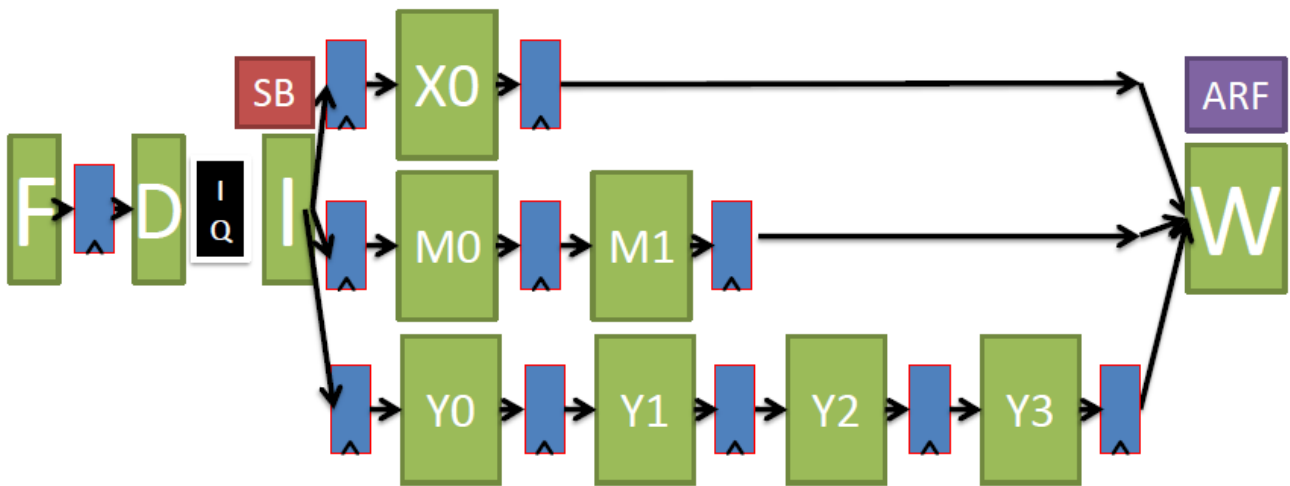
1:    MUL    R1, R2, R3
2:    ADD    R4, R2, R1
3:    MUL    R2, R7, R8
4:    LW     R10, 0(R12)
5:    MUL    R4, R10, R1
6:    SW     R4, 0(R2)
7:    ADD    R5, R6, R7
8:    ADD    R4, R3, R1
9:    BNE    R4, R0
10:   ADDI   R3, R1, 10

```

- (a) Mostre o diagrama do pipeline deste trecho de código, considerando que o desvio da instrução 9 é não-tomado e a instrução 10 pode ser disparada em seguida. Encaminhamentos são permitidos.
- (b) Mostre o estado do Scoreboard quando a instrução 3 está no estágio I do pipeline. Encaminhamentos são permitidos.

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

- (c) Considere agora o processador IO3 visto em aula e apresentado abaixo. Como fica o novo diagrama do pipeline? Apenas uma instrução pode ser disparada por vez. Quando a instrução está no IQ esperando o disparo, considere como estágio i.

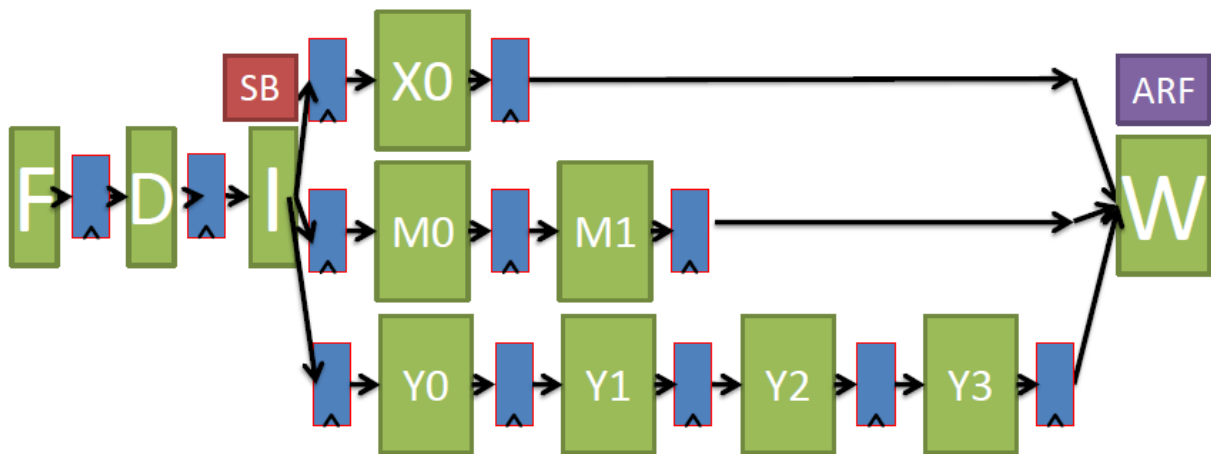


(d) Se ocorrer um problema de desalinhamento de memória no estágio M1 da instrução 6, quais problemas para o tratamento da interrupção.

(f) Como resolver este problema sem reescalonar o código e sem perder desempenho? Mostre uma imagem que ilustre a sua alternativa e descreva os detalhes.

Exercício 5

Considere o processador I2O2 visto em sala e apresentado abaixo. Mostre o estado do Scoreboard a cada ciclo da execução do código abaixo. Encaminhamentos são permitidos.



Processador I2O2

Código:

```
MUL R1, R2, R3
ADD R4, R2, R1
LW R3, 20(R4)
ADD R4, R3, R2
```

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							
	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							
	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

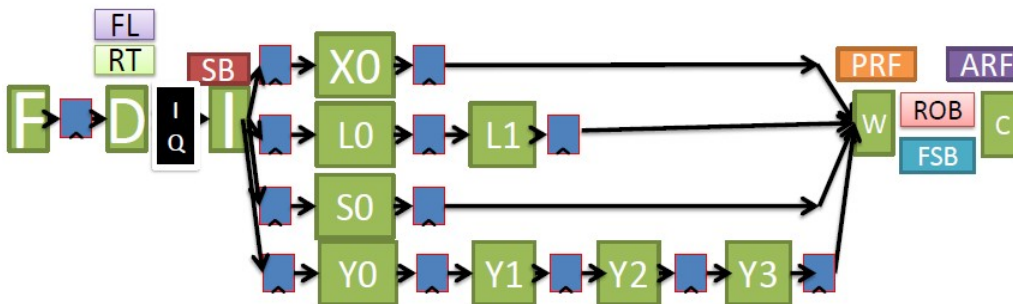
	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

	P	F	4	3	2	1	0
R1							
R2							
R3							
R4							

Exercício 6

Considere o processador IO2I visto em aula e apresentado abaixo. Apenas uma instrução pode ser buscada, decodificada, disparada e escrita por ciclo. Encaminhamentos são permitidos. Considere:

- Pipeline X executa Desvios e operações da ALU;
- Pipeline L executa loads;
- Pipeline S executa stores;
- Pipeline Y executa multiplicação.



(a) Considerando o código abaixo, mostre o diagrama do pipeline. Encaminhamentos são permitidos.

```

0:  MUL  R5, R2, R3
1:  MUL  R4, R2, R5
2:  SUB  R3, R1, R4
3:  ADDI R4, R6, 1

```

(b) Considerando o código acima e as condições iniciais do FL e RT apresentados, **Mostre o estado do Re-order Buffer (ROB), do Free List (FL) e Rename Table (RT)** ao final do ciclo 10.

Considere que o Fetch (busca) da instrução 0 ocorre no ciclo 0.

FL – Inicial

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
0	0	0	0	0	0	1	1	1	1	1	1

RT – Inicial

Registrador	R1	R2	R3	R4	R5	R6
Pendente	0	0	0	0	0	0
P-Reg	P1	P2	P3	P4	P5	P6

Ciclo 10

ROB

Estado	Especulativo	Store Bit	Destino Válido	P-Reg	A-Reg	P P-Reg

FL

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12

RT

Registrador	R1	R2	R3	R4	R5	R6
Pendente						
P-Reg						