

# INF01113 - Organização de Computadores B

## 1ª Lista de Exercícios

Exercícios selecionados do livro-texto da disciplina:

D.Patterson e J.Hennessy

Organização e Projeto de Computadores: a Interface Hardware/Software.

LTC, Rio de Janeiro, 2000 – 2ª edição.

1 – Descreva o efeito causado por manter em 0 todos os sinais de controle dos multiplexadores no caminho de dados **monociclo** da Figura 1 [Figura 5.19]. Quais das instruções, se houver alguma, continuariam a funcionar apesar disto? Considere cada uma das situações separadamente: RegDst = 0, ALUSrc [UALFonte] = 0, MemtoReg [MemParaReg] = 0, Zero = 0.

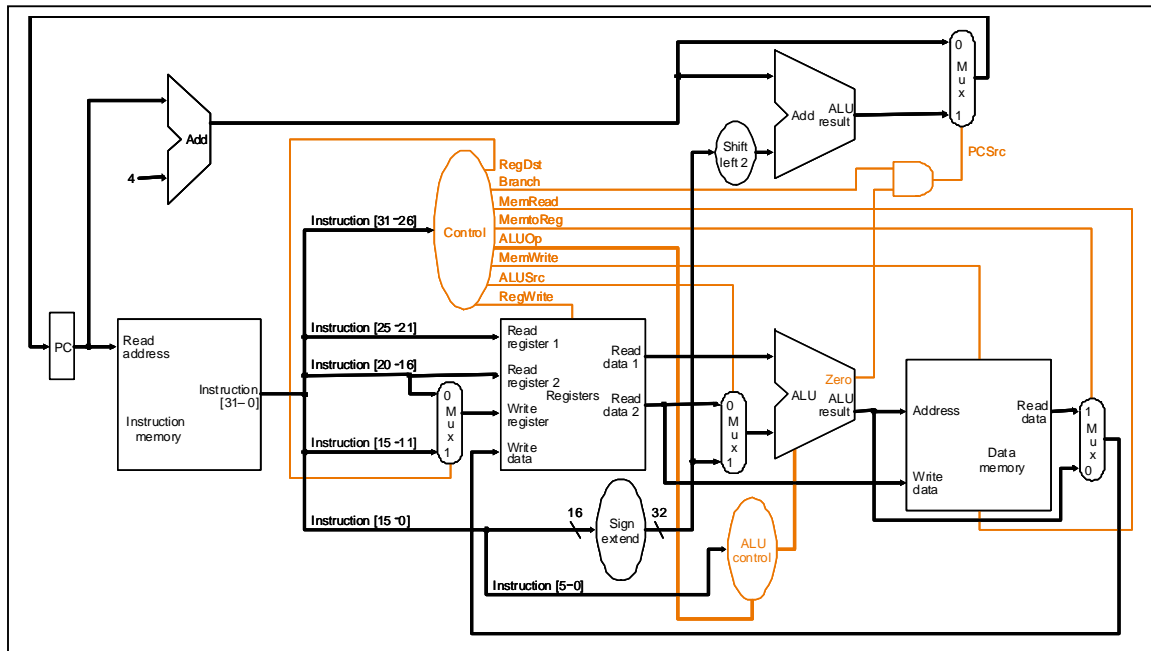


Figura 1 [5.19]

2 – Precisamos adicionar a instrução `addi` (soma imediata) ao caminho de dados monociclo. Acrescente o que for necessário, em termos de elementos e de sinais de controle, ao caminho de dados da Figura 1 [Figura 5.19]. Mostre as modificações na Figura 2 [Figura 5.20], em função desta nova instrução.

Instrução	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
<code>lw</code>	0	1	1	1	1	0	0	0	0
<code>sw</code>	x	1	x	0	0	1	0	0	0
<code>beq</code>	x	0	x	0	0	0	1	0	1

Figura 2 [5.20]

3 – Precisamos acrescentar ao caminho de dados multiciclo a instrução `addi` (soma imediata) descrita abaixo.

Categoria	Instrução	Exemplo	Significado	Comentário
Aritmética	<code>add immediate</code>	<code>addi \$s1, \$s2, 100</code>	$\$s1 = \$s2 + 100$	Coloca a soma do conteúdo do

				registrador fonte(\$s2) com a constante imediata com sinal estendido no registrador destino(\$s1).
--	--	--	--	--

Adicione tudo que for preciso, em termos de elementos de hardware e de sinais de controle, ao caminho de dados da Figura 3 [Figura 5.33].

4 – Se for o caso, mostre as modificações na máquina de controle do MIPS Multiciclo para executar a instrução addi. Por favor, explicita quantos ciclos são gastos para executar a nova instrução no nosso caminho de dados modificado.

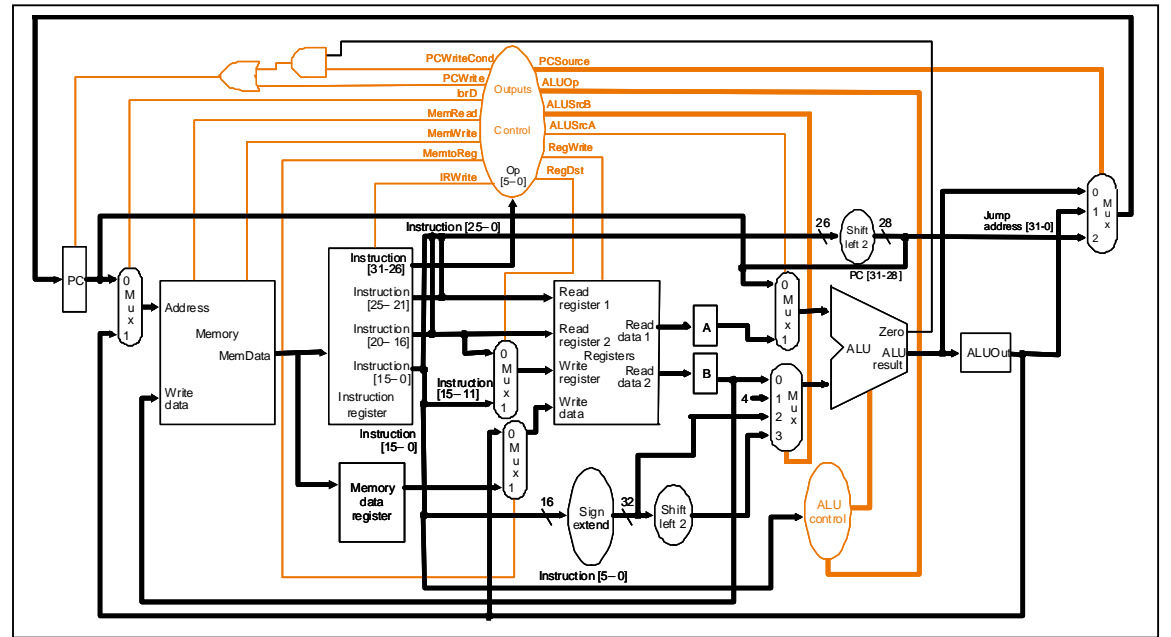


Figura 3 [5.33]

5 - Desejamos comparar a performance de duas máquinas diferentes, M1 e M2. As medidas a seguir foram realizadas nestas máquinas:

Programa	Tempo em M1	Tempo em M2
1	10 segundos	5 segundos
2	3 segundos	4 segundos

Programa	Instruções executadas em M1	Instruções executadas em M2
1	200 x 10 <sup>6</sup>	160 x 10 <sup>6</sup>

Com base em tais medidas, pergunta-se: qual das máquinas é mais rápida na execução de cada um dos programas? Escolhida a mais rápida, calcule quão mais rápida ela é. Encontre também, a velocidade de execução de instruções (medidas em instruções por segundo) para cada uma das máquinas, ao rodar o programa1.

6 - Se os clocks das máquinas M1 e M2 do Exercício 25 [2.1] forem de 200 MHz e 300 MHz, respectivamente, encontre a quantidade de ciclos gastos por instrução (CPI), considerando o programa1, em ambas as máquinas, usando os dados do exercício 25 [2.1 e 2.2].

7 - Suponha que você tenha sido designado líder do projeto de um sistema computacional. Tanto o projeto do processador quanto o do compilador estão prontos, e você precisa decidir se manda o projeto para produção em seu estágio atual ou se adia o início da produção para esperar por melhoras a serem introduzidas. Para tomar esta decisão, você chama o seu engenheiro de hardware, discute o problema com ele e fica com as seguintes opções:

a. Deixar o projeto como está. Vamos chamar a máquina resultante desta decisão de *máquina básica*, ou *Mbásica*. Ela tem um clock de 500 MHz, e nela foram realizadas as seguintes medidas:

Classe de Instruções	CPI	Frequência
A	2	40%
B	3	25%
C	3	25%
D	5	10%

b. Otimizar o hardware. O pessoal do hardware afirma ser capaz de melhorar o projeto do processador, fazendo-o aceitar um clock de 600 MHz. A máquina resultante é chamada de *Mopt*, e nela foram realizadas as seguintes medidas:

Classe de Instruções	CPI	Frequência
A	2	40%
B	2	25%
C	3	25%
D	4	10%

Com base nessas informações, pede-se a CPI de cada uma das máquinas.

8 - Qual o valor do MIPS nativo para as máquinas *Mbase* e *Mopt* definidas no Exercício 7?

9 [6.2] – Usando um desenho similar ao da Figura 7 [6.8], mostre os caminhos para adiantamento de dados necessários à execução das seguintes instruções:

```

add    $2, $3, $4
add    $4, $5, $6
add    $5, $3, $4
  
```

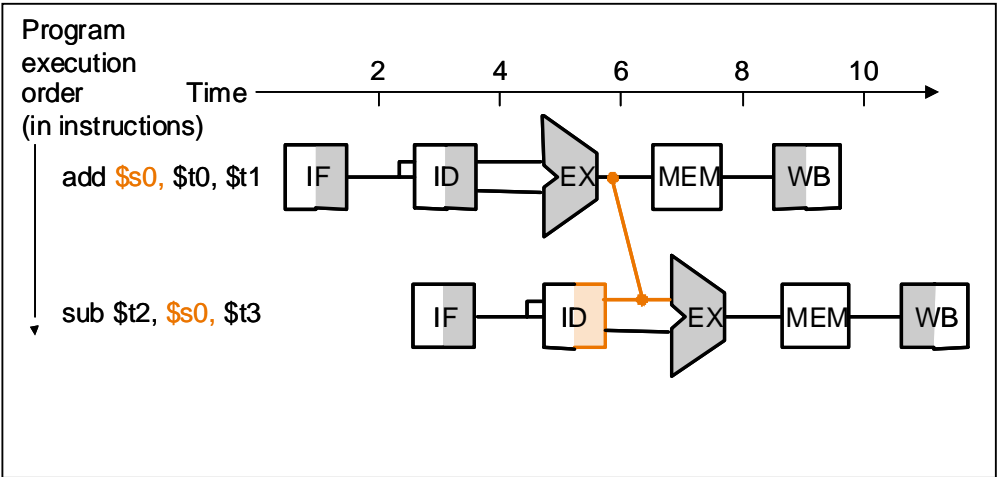


Figura 7 [6.8]

10 [6.11] – Considere a execução do seguinte código, no caminho de dados pipeline da Figura 8 [6.46]:

```
add    $1, $2, $3
add    $4, $5, $6
add    $7, $8, $9
add    $10, $11, $12
add    $13, $14, $15
```

No final do quinto ciclo da execução, quais registradores estão sendo lidos, e quais serão escritos ?

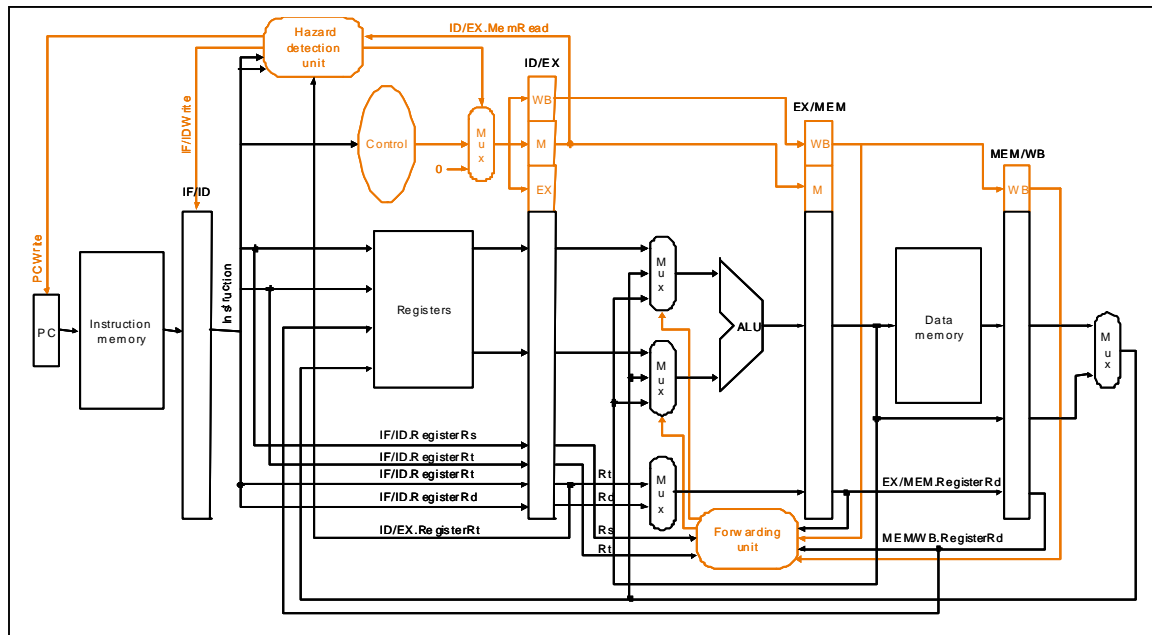


Figura 8 [6.46]

11 [5.9] – Explique por que não é possível modificar a implementação monociclo para implementar a instrução **swap** (que troca o conteúdo de dois registradores), sem modificar o banco de registradores.

Obs.: Com a instrução **swap** após completar a sequência, o registrador destino tem o valor original do registrador fonte, e o registrador fonte tem o valor original do registrador destino. A semântica da instrução **swap** é

```
swap $s0, $s1
```

**12 [5.10]** – Um amigo está propondo a eliminação do sinal de controle MemtoReg [MemParaReg]. O multiplexador que tem este sinal como entrada usaria em seu lugar o sinal MemRead [LerMem]. Você acha que esta modificação vai funcionar? Considere o caminho de dados **monociclo**.

**13 [5.12]** – Considere a seguinte idéia: vamos modificar a arquitetura do conjunto de instruções e retirar a capacidade de especificar um deslocamento para instruções com acesso à memória. Especificamente, todas as instruções `load-store` com deslocamento diferente de zero devem tornar-se pseudo-instruções e devem ser implementadas por meio de duas instruções. Por exemplo:

```
addi  $at, $t1, 104  # some o deslocamento ao conteúdo de um registrador temporário
lw     $t0, $at       # nova forma de fazer lw $t0, 104($t1)
```

Que mudanças devem ser feitas no caminho de dados monociclo e no controle para que essa arquitetura simplificada possa funcionar ?

**14 [5.13]** – Se as modificações descritas no Exercício 13 [5.12] forem implementadas, haverá algumas vantagens e algumas desvantagens com respeito à performance. Especificamente, o período do clock pode vir a ser afetado, e todas as instruções `load-store` com deslocamento diferente de zero iriam exigir a execução de uma instrução extra `addi` (um bom compilador acharia um meio de reduzir a necessidade por estas instruções extras, mas você pode ignorar este fato). Se existirem muitas instruções de `load-store` com deslocamento diferente de zero, fica claro que esta mudança não vai melhorar a performance. Admitindo os retardos especificados no item Performance de um Processador Monociclo com Instruções de Ponto Flutuante, na Seção 5.3, qual a percentagem mais alta de instruções `load-store` com deslocamentos não-zero que poderia ser tolerada? (Ou seja, o que seria necessário para que a modificação viesse a ter um impacto positivo sobre a performance?)

**13 [5.14]** – Ao estimar a performance de uma implementação monociclo, partimos do pressuposto de que somente as principais unidades funcionais têm um retardo que mereça ser considerado (ou seja, os retardos introduzidos no sinal por multiplexadores, pela unidade de controle, pelo acesso ao PC, pela unidade de extensão de sinal e pela fiação foram considerados desprezíveis). Suponha que os retardos listados no item Performance de um Processador Monociclo com Instruções de Ponto Flutuante, na Seção 5.3, tenham sido mudados em função do uso de novos somadores para adição simples:

- UAL: 2 ns
- Somador para PC + 4: X ns
- Somador para cálculo endereço de desvio condicional: Y ns

Com base nisto, pergunta-se:

- a. Qual seria o período do clock caso X = 3 e Y = 3?
- b. Qual seria o período do clock caso X = 5 e Y = 5?
- c. Qual seria o período do clock caso X = 1 e Y = 8?

**14 -** Dado um processador com 5 estágios de pipeline (F,DR,O,M,W),  $T_{ciclo} = 2ns$ , pede-se: quanto tempo será tomado para executar 1B de instruções?  
qual a aceleração em relação a:

- uma máquina single-cycle com  $T_{cycle} = 8ns$ ;
- uma máquina multi-cycle com  $T_{cycle} = 2ns$ ,  $CPI = 4$ ;

Graças a melhora na tecnologia, passa-se o pipeline a 7 estágios (FF,DR,OO,MM,W), com  $T_{ciclo} = 1ns$ .

Quanto tempo será tomado para executar 1B de instruções?  
Qual a aceleração em relação a:

- uma máquina single-cycle com  $T_{cycle} = 8ns$ ;
- uma máquina multi-cycle com  $T_{cycle} = 1ns$ ,  $CPI = 5$ ;

**15 -** O pipe de 5 estágios decide o salto no quarto estágio. A política usada é esvaziar o pipe. Qual a penalidade paga, se 30% das instruções forem branches? Qual o CPI? Melhorou-se o processador com uma política de predição de saltos com 85% de acertos. Qual a melhoria em desempenho? Qual CPI se

tem? Repita o problema para um pipe de 7 estágios, onde a decisão sobre saltos acontece no 5º estágio.

**16** - Dados os seguintes tempos e assuma M0 single cycle:

IM=4ns, Reg=1ns, ALU=3ns, DM=4ns, RegW=1ns

Qual o tempo para executar 1B de instruções?

Qual a MELHOR partição para o pipeline de 3 estágios, assumindo blocos indivisíveis? (M1) Compare o desempenho com M0.

Qual a MELHOR partição para o pipeline de 5 estágios, assumindo blocos indivisíveis? (M2) Compare o desempenho com M0.

Qual a MELHOR partição, assumindo blocos divisíveis? (M3)

Qual o desempenho de M0..3 se a taxa de branches é de 30%, sem preditor?

Qual o desempenho de M0..3 se a taxa de branches é de 30%, com preditor que acerta 90%?

**17** – altere o caminho de dados e a máquina de controle do MIPS Multiciclo para a inclusão da instrução lwp, lw com pós-incremento:

Lwp R1, R2, 100 = R1 = Mem[r2+100]; R2=R2+1

Qual o impacto no tempo de execução, se 50% das instruções de lw são lwp, e a distribuição das instruções é 20% lw, 10% sw, 50% tipo R e 30% tipo branch?

**18** - repita 17, mas agora na máquina single-cycle. Qual a penalidade?