

3ª Lista de Exercícios

Observação:

Os exercícios desta lista não serão cobrados. Porém, é altamente recomendável que os alunos tentem resolvê-los, a fim de se prepararem minimamente para a 2ª prova semestral.

Exercício 1 (ULA do MIPS monociclo: estrutura e funcionamento)

A Fig. 1 mostra o esquemático do nível RT para a ULA do MIPS monociclo. Esta ULA é semelhante àquela vista em aula, exceto pelo fato de possuir recursos que lhe permitem realizar a operação “S = 1 se A < B senão, S = 0” (bloco “SLT” e um multiplexador extra controlado pelo sinal “ctrl”) e recursos para identificar se S=0 (bloco “=0?”).

- Desenhe o esquemático para “SLT”, sabendo que ele é composto por conexões (fios) e porta(s) lógica(s). (Não use multiplexador.) “SLT” deve realizar a operação “S = 1 se A < B senão, S = 0” corretamente **mesmo quando o somador/subtrator detectar overflow**!
- Escreva a equação para o sinal “ctrl”.
- Desenhe o esquemático no nível lógico para o bloco “=0?”, o qual é responsável pela saída “Zero” da ULA (“flag de zero”).

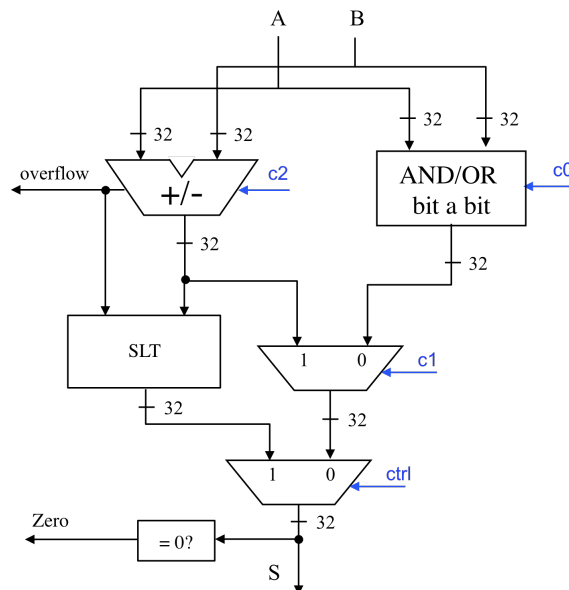


Fig. 1– Esquemático no nível RT da ULA do MIPS monociclo.

Exercício 2 (ULA do MIPS monociclo: custo)

Partindo das estimativas de custo vistas em aula, estime o custo de uma implementação CMOS da ULA da Fig. 1, em termos de número de transistores. Considere que:

- Um mux 2:1 (de 1 bit) necessite de 4 transistores;
- O bloco “=0?” deve utilizar portas NAND e/ou NOR de duas entradas e inversores. Porém, o número de inversores deve ser mínimo.

Exercício 3 (Deslocador do MIPS)

Desenhe o esquemático do nível lógico para o deslocador à esquerda de 2 bits utilizado no MIPS monociclo para o cálculo do endereço de desvio, no caso de uma instrução beq.

Exercício 4

a) Quais instruções do processador MIPS acessam a memória de dados? Quais seriam as implicações no projeto do processador MIPS, caso houvesse instruções aritméticas cujos operandos (de entrada) tivessem que ser lidos da memória de dados?

b) Considere que o registrador *\$pc* (PC) do processador MIPS armazene o valor *0x000C0000* em uma instrução *beq*. Após a execução da instrução *beq* qual o maior endereço que pode ser atingido? Qual o menor endereço?

c) Considere ainda que o registrador *\$pc* (PC) armazene o valor *0x000C0000*. Caso seja utilizada uma instrução *jump* para realizar o desvio, qual o maior endereço que pode ser atingido? Qual o menor endereço?

d) Proponha um trecho de código MIPS que realize $e = (a+b) - (c-d)$. Considere que *a*, *b*, *c* e *d* estão nos registradores *\$s0*, *\$s1*, *\$s2* e *\$s3* e que a variável *e* corresponde ao registrador *\$s4*.

e) Proponha um trecho de código MIPS que realize $A[20] = a + A[10]$. Considere que o endereço base do array *A* esteja no registrador *\$s1* e a variável *a* esteja no registrador *\$s0*.

f) As instruções **add**, **sub**, **and** e **or** são instruções do tipo R que buscam seus operandos em registradores. Caso um dos operandos seja uma constante, o programador pode, alternativamente, utilizar certas instruções do tipo I para realizar somas, subtrações, ands ou or. Consulte o manual do MIPS (no livro) e apresente essas instruções juntamente com os seus códigos de operação.

Exercício 5

Para resolver a presente questão, considere as instruções do processador MIPS descritas a seguir.

Mnemônico	Instrução	Linguagem de Montagem	Significado
addi	Adição imediata	addi \$s1, \$s2, const	$\$s1 \leftarrow \$s2 + \text{const}$
add	Adição	add \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 + \$s3$
sub	Subtração	sub \$s1, \$s2, \$s3	$\$s1 \leftarrow \$s2 - \$s3$
slt	Set on less than	slt \$s1, \$s2, \$s3	if($\$s2 < \$s3$) $\$s1 \leftarrow 1$; else $\$s1 \leftarrow 0$
lw	Load word	lw \$s1, des1(\$s2)	$\$s1 \leftarrow \text{Mem}[\$s2 + \text{des1}]$
sw	Store word	sw \$s1, des1(\$s2)	$\text{Mem}[\$s2 + \text{des1}] \leftarrow \$s1$
beq	Branch on equal	beq \$s1, \$s2, L	if($\$s1 == \$s2$) desvia para L
j	Jump	j L	desvia para a linha cujo label é "L"

Complete o código em linguagem de montagem gerado pelo compilador do processador MIPS para o trecho de código em linguagem C dado a seguir, onde *X* é um vetor de *n* elementos que foi alocado estaticamente na memória e seu endereço-base está armazenado no registrador *\$s0*. Considere também que neste trecho de código as variáveis *i*, *total* e *n* estejam nos registradores *\$s1*, *\$s2* e *\$s3*, respectivamente, e que o registrador *\$zero* contenha a constante zero (i.e., *\$zero*=0).

```
i=0;
total=0;
while( i < n ) {
    total=total+X[i];
    i=i+1;
}
```

Trecho de programa em linguagem de montagem do processador MIPS. Há exatamente **quatro** instruções que devem ser completadas!

	add	\$s1, \$zero, \$zero
	add	\$s2, \$zero, \$zero
Loop:	slt	\$t0, \$s1, \$s3
	beq	\$t0, \$zero, Exit

add	\$t1, \$s1, \$s1
	\$t1,
addi	\$s1, \$s1, 1
j	Loop
Exit:	próxima instrução

Exercício 3

A Fig. 2 mostra o diagrama do processador MIPS, versão monociclo.

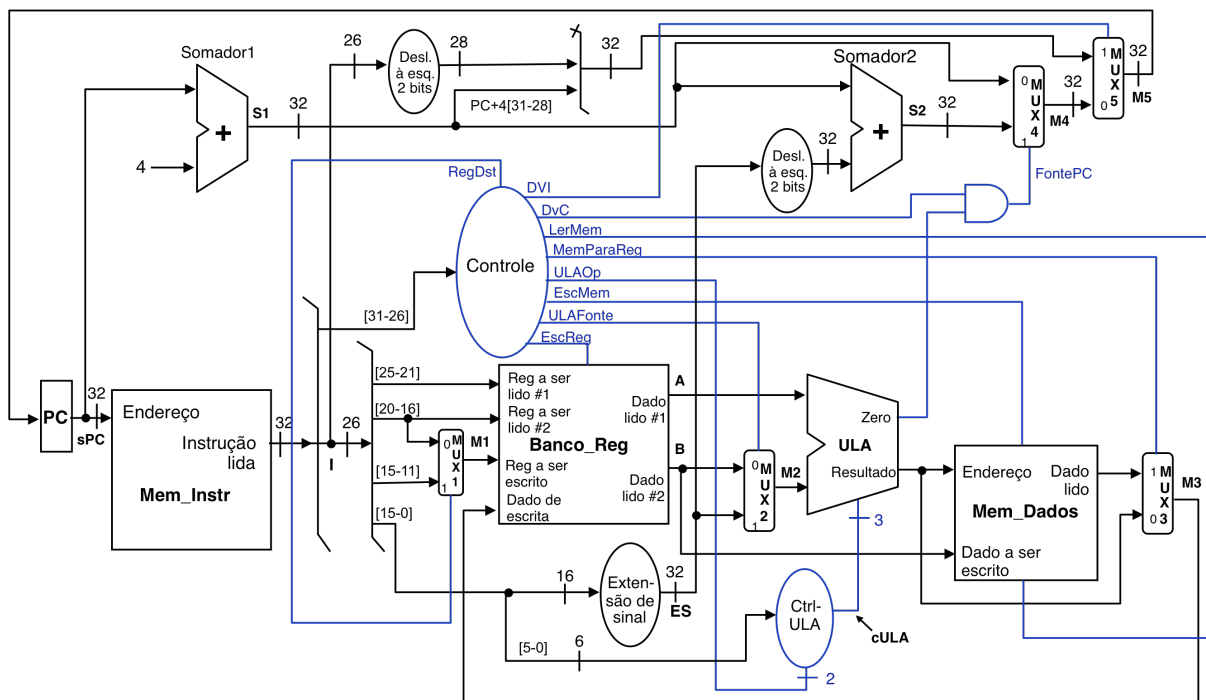


Fig. 2 – Diagrama do processador MIPS, versão monociclo.

- a) O bloco de controle do MIPS monociclo é a um circuito combinacional ou a um circuito sequencial do tipo “máquina de estados” (FSM)? Justifique.

A Tab. 1 mostra como as instruções add, sub, lw, sw, beq e jump estão definidas no MIPS.

Tab. 1 – Descrição das instruções add, sub, lw, sw, beq e jump do MIPS.

Mnemônico	Instrução	Linguagem de Montagem	Significado
add	Adição	add \$s1, \$s2, \$s3	\$s1 := \$s2 + \$s3
sub	Subtração	add \$s1, \$s2, \$s3	\$s1 := \$s2 - \$s3
lw	Load word	lw \$s1, deslocam(\$s2)	\$s1 ← Mem[\$s2 + deslocam]
sw	Load word	sw \$s1, deslocam (\$s2)	Mem[\$s2 + deslocam] ← \$s1

beq	branch on equal	beq \$s1, \$s2, deslocam	if(\$s1 == \$s2) then PC ← PC+4+4xdeslocam
j	jump	j, endereço	PC ← endereço

Nesta definição \$s1, \$s2 e \$s3 são registradores e deslocam é uma constante inteira de 32 bits, obtida a partir da extensão de sinal dos 16 bits menos significativos da instrução (formato "I").

- b) Preencha a Tab. 2 com os valores adequados dos sinais de controle que configuram o bloco operativo (*datapath*) do MIPS monociclo para a execução de cada uma das instruções da Tab. 1. Lembre-se que, conforme discutido em aula, os sinais que controlam leitura e escrita de elementos de memória (memórias e registradores) não podem valer "X" (*don't care*).

Tab. 2 – Sinais de controle para o bloco operativo do MIPS monociclo executar add, sub, lw, sw, beq e jump (a completar).

	DVI	DvC	ULAFonte	ULAOp	LerMem	EscMem	MemParaReg	RegDst	EscReg
add				10					
sub				10					
lw				00					
sw				00					
beq				01					
j				XX					

A Tab. 3 mostra valores para as características temporais dos componentes do MIPS monociclo da Fig. 2.

Para resolver os itens que seguem, considere o diagrama de blocos do MIPS monociclo mostrado na Fig. 2 e os atrasos de seus componentes apresentados na Tab. 2. Considere também a seguinte nomenclatura: dado um sinal “nome_do_sinal”, TE(nome_do_sinal) designa seu tempo de estabilização (nesta questão, em picossegundos – ps).

- Calcule TE(I), TE(S1), e TE(S2). Estes TEs dependem da instrução a ser executada? Explique.
- Calcule os tempos de estabilização (TEs) dos sinais de controle e do sinal cULA do MIPS da Fig. 2.
- Calcule TE(A) e TE(B). Quais instruções fazem uso do valor fornecido pelo sinal B? Estes TEs dependem da instrução a ser executada? Explique.
- Considerando a instrução add, calcule TE(M5). Quais instruções não possuem o mesmo valor para TE(5)? Explique.
- Considerando ainda a instrução add, calcule TE(M3).
- Qual é o atraso crítico para a instrução add? Escreva o caminho crítico para esta instrução.
- Considerando a instrução lw, calcule TE(M3).
- Qual é o atraso crítico para a instrução lw? Escreva o caminho crítico desta instrução.
- Para determinar o atraso crítico do MIPS monociclo (considerando as características temporais da Tab. 2), é preciso realizar a análise timing para as demais instruções vistas em aula? (Sim ou não?) Explique.
- Qual é atraso crítico do MIPS monociclo desta questão? Justifique e ou mostre o cálculo.
- Calcule a frequência máxima de operação para o processador MIPS monociclo considerando as características temporais da Tab. 2. Indique seu valor em GHz (gigahertz) e mostre o cálculo.

Tab. 3 – Características temporais dos componentes do MIPS, versão monociclo. Obs: 1 ps = 1×10^{-12} s

Componente	Característica	Símbolo	Valor
Memória de instruções	tempo para leitura	td _{LMEM}	300 ps
Memória de dados	tempo para leitura	td _{LMEM}	300 ps
Memória de dados	tempo para escrita (setup)	td _{EMEM}	300 ps
Banco de Registradores	tempo para leitura	td _{LREG}	70 ps

Banco de Registradores	tempo para escrita (setup)	td_{EREG}	70 ps
ULA	atraso para qualquer operação	td_{ULA}	60 ps
Qualquer somador	atraso	td_{soma}	20 ps
Qualquer mux 2:1	atraso	td_{mux}	2 ps
PC	tempo de setup	tsu	5 ps
PC	tempo de carga	tco	5 ps
PC	tempo de hold	th	Desprezível (0 ps)
Deslocador, extensão de sinal, porta AND	atraso	-	Desprezível (0 ps)
Controle	atraso	td_{controle}	75 ps
Controle da ULA	atraso	td_{ctrlULA}	5 ps

Exercício 4

Considere a instrução `addi`, a qual é definida da seguinte forma:

Mnemônico	Instrução	Linguagem de Montagem	Significado
<code>addi</code>	Adição imediata	<code>addi \$s1, \$s2, deslocam</code>	$\$s1 := \$s2 + \text{deslocam}$

Considere que esta instrução utiliza o **formato I** (o mesmo utilizado nas instruções `lw` e `sw`), possuindo porém um código de operação ("opcode") próprio, diferente daquele utilizado nas instruções aritméticas e lógicas que operam somente com registradores (`add`, `sub`, `and`, `or`, `slt`). O valor `deslocam` é uma constante inteira de 32 bits, obtida a partir da extensão de sinal dos 16 bits menos significativos da instrução.

- Quais alterações são necessárias no bloco operativo (*datapath*) do MIPS monociclo para que `addi` possa ser executada?
- Quais alterações são necessárias no bloco de controle do MIPS monociclo para que `addi` possa ser executada?
- Complete na tabela a seguir os valores dos sinais de controle que devem ser aplicados ao bloco operativo (*datapath*) do MIPS monociclo para que `addi` seja executada corretamente.

	DVI	DvC	ULAFonte	ULAOp	LerMem	EscMem	MemParaReg	RegDst	EscReg
<code>addi</code>				00					