

Arquitectura de Sistemas e Computadores II

3^a Frequência e Exame

Departamento de Informática
Universidade de Évora

8 de Janeiro de 2019

- Os símbolos à esquerda de cada pergunta identificam a prova ou provas a que ela pertence:
 - ♣ assinala as perguntas do exame;
 - ◊ assinala as perguntas da frequência.
- O exame está cotado para 20 valores. A frequência está cotada para 15 valores.
- Indique todos os cálculos efectuados

Perguntas rápidas

- 1. [0,5 valores] Sabendo que a execução de um programa no processador X necessita do dobro dos ciclos de relógio da execução do mesmo programa no processador Y, pode concluir que o desempenho de X para esse programa é inferior ao de Y?
- 2. [0,5 valores] Quais são as técnicas usadas, no pipeline MIPS de cinco andares, para garantir que uma instrução não usa o conteúdo desactualizado de um registo?
- ♣ ◊ 3. [0,5 valores] Se a dimensão dos blocos de cache for aumentada, o número de *compulsory misses* deverá aumentar, manter-se ou diminuir?
0,6
- ◊ 4. [0,5 valores] Em que estratégia para lidar com as operações de escrita numa cache é necessário associar um *dirty bit* aos blocos na cache?
0,6
- ♣ ◊ 5. [0,5 valores] Uma página virtual de um programa tem de residir na mesma página física durante toda a execução do programa?
0,6
- ◊ 6. [0,5 valores] A comunicação entre processos em sistemas de multiprocessamento de memória partilhada é explícita ou pode ser implícita?

Desempenho

- 7. No processador A, cujo relógio funciona a uma frequência de 2 GHz, o CPI do programa P é 3,6.
 - (a) [2 valores] Se a distribuição das instruções do programa for a apresentada abaixo, qual o CPI das instruções de salto?

Classe	Aritméticas	Acesso à memória	Saltos
%	40	40	20
CPI	2	5	?

- (b) [2,5 valores] Seja B um processador que implementa a mesma arquitectura que A. No processador B, cujo relógio funciona a uma frequência de 4 GHz, o CPI de P é 4,8.

Calcule o *speedup* que se obtém quando P é executado em B em relação à sua execução em A.

(Note que os CPI da alínea anterior não se aplicam ao processador B.)

(CONTINUA...)

Implementação MIPS monociclo

8. [4 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução `aui` (*add upper immediate*), que é uma instrução tipo-I com três argumentos:

<code>aui rt, rs, immediate</code>	<table border="1"><tr><td><code>aui</code></td><td><code>rs</code></td><td><code>rt</code></td><td><code>immediate</code></td></tr><tr><td>bits</td><td>6</td><td>5</td><td>5</td></tr><tr><td></td><td></td><td></td><td>16</td></tr></table>	<code>aui</code>	<code>rs</code>	<code>rt</code>	<code>immediate</code>	bits	6	5	5				16
<code>aui</code>	<code>rs</code>	<code>rt</code>	<code>immediate</code>										
bits	6	5	5										
			16										

Esta instrução coloca no registo `rt` o valor obtido somando `immediate` $\ll 16$ ao valor contido no registo `rs`.

- Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?
- Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- Quais os valores que os vários sinais de controlo deverão ter durante a execução desta instrução?
(Não precisa de indicar o valor de `ALUOp`, basta dizer qual será a operação executada pela ALU durante a execução desta instrução.)
- Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

Pipeline MIPS de 5 andares

9. [2 valores] Simule a execução do código à direita num processador com *forwarding*, com decisão dos saltos condicionais no andar ID, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, assumindo que o salto condicional não é efectuado. Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

- `add $t0, $a1, $a1`
- `add $t0, $t0, $t0`
- `addu $t1, $a0, $t0`
- `lw $t2, 0($t1)`
- `lw $t2, 4($t2)`
- `sw $t2, 4($a2)`
- `beq $t2, $a3, salta`
- `or $v0, $0, $t2`

Cache

10. Considere um sistema com uma cache com dois níveis em que, para um programa, a cache de primeiro nível apresenta uma *miss rate* de 4% e a de segundo nível apresenta uma *miss rate* de 50%.

- 10.3 (a) [2,5 valores] Se o *hit time* da cache de primeiro nível for 3 ciclos, o *hit time* da cache de segundo nível for 20 ciclos, e o tempo de acesso à memória física for 120 ciclos, qual o tempo médio de acesso à memória (em ciclos) na presença e na ausência da cache de segundo nível?

- 10.3 (b) [2 valores] Mantendo-se a *miss rate* da cache de primeiro nível, qual deveria ser a *miss rate* da cache de segundo nível para se obter uma *miss rate* global de 1,5%?

Memória virtual

11. [3 valores] Considere um sistema em que a dimensão das páginas de memória é de 4 KB, os endereços virtuais têm 40 bits, os endereços físicos têm 32 bits, o TLB é *direct mapped* e tem espaço para 16 blocos com uma tradução cada, e em que é acedido o endereço virtual 00 1000 2308₁₆.

Apresente o conteúdo da posição do TLB em que a tradução relativa àquele acesso é inserida e indique o seu índice, assim como o conteúdo da posição correspondente da tabela de páginas. Assuma que a página virtual é colocada na página física 734.

12. [1,5 valores] Aponte duas vantagens do uso de memória virtual.

Multiprocessamento

13. O código abaixo vai ser executado nos dois processadores MIPS de um multiprocessador de memória partilhada. No início da execução destas instruções, o endereço guardado no registo `$a0` é $1000\ 0000_{16}$ nos dois processadores e a palavra nesse endereço tem o valor 200.

- `lw $t1, 0($a0)`
- `add $t1, $t1, 10`
- `sw $t1, 0($a0)`

13. (a) [2 valores] Mostre que, depois da execução destas instruções nos dois processadores, o valor no endereço $1000\ 0000_{16}$ pode não ser 220.

13. (b) [2 valores] Altere o código apresentado de modo a que quando for executado em paralelo nos dois processadores, o valor naquela posição de memória seja sempre 20 mais o valor original.

	\$10	\$11	\$12	\$13	MEM[1000000000]
0	?	?	?	?	200
1	200	"	"	"	"
2	198	"	200	"	"
3	1	18	200	18	198
4	"	198	"	"	"
5	"	"	0	"	"
6	"	"	"	198	"

Arquitectura de Sistemas e Computadores II

3^a Frequência e Exame

Departamento de Informática
Universidade de Évora

8 de Janeiro de 2018

- Os símbolos à esquerda de cada pergunta identificam a prova ou provas a que ela pertence:
♣ assinala as perguntas do exame; ♦ assinala as perguntas da 2^a frequência.
- Indique todos os cálculos efectuados

Perguntas rápidas

- ♣ 1. [0,5 valores] O desempenho do processador X ser superior ao do processador Y significa que o relógio de X funciona a uma frequência superior à do relógio de Y ?
- ♣ 2. [0,5 valores] Idealmente, quantas vezes mais rápida seria a execução de um programa num *pipeline* de cinco andares em relação à execução numa implementação monociclo?
- ♣ ♦ 3. [0,5 valores] Em quantas posições pode ser encontrado cada bloco numa cache 8-way set associative com 256 conjuntos?
- ♣ ♦ 4. [0,5 valores] O que caracteriza a estratégia *write-through* para lidar com as operações de escrita numa cache?
- ♦ 5. [0,5 valores] Se a tradução de uma página virtual para página física não é encontrada no TLB, onde é que ela deve ser procurada a seguir?
- ♦ 6. [0,5 valores] A comunicação entre processos em sistemas de multiprocessamento de memória distribuída é explícita ou implícita?

Desempenho

- ♣ 7. [2 valores] A execução do programa P no computador A , cujo relógio funciona a uma frequência de 1 GHz, demora 2,4 s. Quantas instruções de P são executadas se elas apresentarem a distribuição abaixo?

Classe	Aritméticas	Acesso à memória	Saltos
%	40	40	20
CPI	2	3	2

- ♣ 8. [2 valores] Se um programa corre em 20 segundos e as instruções de acesso à memória são responsáveis por 15 segundos desse tempo, quanto é necessário aumentar a velocidade dos acessos à memória para que o programa corra 2,5 vezes mais depressa?

(CONTINUA...)

Implementação MIPS monociclo

9. [4 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução *jri* (*jump register indexed*), que é uma instrução tipo-I com dois argumentos:

jri rs, rt	bits	6	5	5	16	0
------------	------	---	---	---	----	---

Esta instrução provoca a continuação da execução na instrução cujo endereço é obtido somando os conteúdos dos registo *rs* e *rt*.

- Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?
- Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- Quais os valores que os vários sinais de controlo deverão ter durante a execução desta instrução?
(Não precisa de indicar o valor de ALUOp, basta dizer qual será a operação executada pela ALU durante a execução desta instrução.)
- Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

Pipeline MIPS de 5 andares

10. [2 valores] Simule a execução do código seguinte num processador com *forwarding*, com decisão dos saltos condicionais no andar ID, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, assumindo que nenhum dos saltos condicionais é efectuado. Apresente a evolução do estado do pipeline durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

```
1.          or  $t0, $0, $a0
2.  ciclo:   lw   $t1, 0($t0)
3.          beq $t1, $a1, fim
4.          lw   $t0, 4($t0)
5.          sw   $t0, 0($a0)
6.          bne $t0, $0, ciclo
7.  fim:     slt $v0, $0, $t0
8.          jr   $ra
```

Quantos ciclos de relógio são necessários para executar o código nas condições acima?

Quantos ciclos de relógio seriam necessários para executar o código se o ciclo (instruções 2 a 6) fosse executado 100 vezes?

Cache

11. [2 valores] Considere que uma palavra tem 64 bits e que os endereços seguintes são acedidos pela ordem indicada:

24 40 36 56 80 24

Simule o funcionamento de uma cache 2-way set associative, com 8 palavras e blocos de 2 palavras, para a sequência de acessos indicada. Assuma que a cache inicialmente está vazia e, para cada acesso, indique a palavra acedida, o número do bloco a que pertence a palavra, o índice da posição da cache que irá ocupar, o *tag*, se há um *hit* ou um *miss* e, quando aplicável, o número do bloco que será substituído. Apresente o conteúdo final da cache, tão completo quanto possível, e calcule a *miss rate* verificada.

(CONTINUA...)

- ◊ 12. Considere um sistema com uma cache com dois níveis em que, para um programa, que faz 1 milhão de acessos à memória, a cache de segundo nível é acedida 80 mil vezes.
- [2,5 valores] Qual a *miss rate* da cache de primeiro nível e qual deverá ser a *miss rate* da cache de segundo nível para se obter uma *miss rate* global de 2%?
 - [2,5 valores] Se o *hit time* da cache de primeiro nível for 4 ciclos, o *hit time* da cache de segundo nível for 25 ciclos, e o tempo de acesso à memória física for 100 ciclos, qual o tempo médio de acesso à memória (em ciclos) na presença e na ausência da cache de segundo nível?

Memória virtual

- ◊ 13. Considere um sistema MIPS em que a dimensão das páginas de memória é de 4 KB. Num momento da execução de um programa, a sua tabela de páginas apresenta o conteúdo (parcialmente) mostrado abaixo.

- [2 valores] Assumindo que as páginas virtuais foram acedidas pela ordem 17, 18, 19, 17 e 21, apresente o conteúdo do TLB do sistema (*2-way set associative*, com 4 blocos de uma tradução e substituição por LRU).
- [2 valores] Calcule o endereço virtual correspondente ao endereço físico $00\ 7C\ 40_{16}$.

Tabela de páginas

	Dirty	Pág. física
17	1	7
18	0	12
19	1	23
20	0	DISCO
21	0	3
	...	

Multiprocessamento

- ◊ 14. [2 valores] Num multiprocessador de memória partilhada é executado um programa paralelo, sendo as instruções seguintes executadas no processador indicado:

Processador 1	Processador 2
$x = x * 5;$	$x = x + x;$

Liste todas as possíveis sequências de valores que a variável x poderá assumir durante a execução deste código se, inicialmente, o valor de x for 2.

- ◊ 15. [2 valores] Num multiprocessador de memória partilhada com dois processadores MIPS, os processadores vão executar as instruções seguintes, pela ordem apresentada:

	Processador 1	Processador 2
1.	11 \$10, 0(\$4)	
2.	addi \$10, \$10, -2	11 \$12, 0(\$4)
3.	sc \$10, 0(\$4)	addi \$12, \$12, 10
4.	lw \$11, 0(\$4)	
5.		sc \$12, 0(\$4)
6.		lw \$13, 0(\$4)

Antes da execução destas instruções, o endereço guardado no registo $\$4$ é $1000\ 0000_{16}$ nos dois processadores e a palavra nesse endereço tem o valor 200.

Considerando que as instruções na mesma linha são executadas em simultâneo, diga quais os valores nos registos $\$10$ e $\$11$ do Processador 1, nos registos $\$12$ e $\$13$ do Processador 2, e na posição de memória com endereço $1000\ 0000_{16}$, depois da execução de *cada* instrução.

Arquitectura de Sistemas e Computadores II

2^a Frequência

Departamento de Informática
Universidade de Évora

13 de Dezembro de 2016

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] Idealmente, quantas vezes mais rápida será a execução de um programa num *pipeline* de cinco andares em relação à execução numa implementação monociclo?

4x

2. [1 valor] Quantos ciclos de relógio demora a execução da instrução add no *pipeline* MIPS de cinco andares?

5 ciclos

3. [1 valor] Se a instrução que está no andar EX do *pipeline* MIPS gera uma exceção, as instruções que se encontram nos andares MEM e WB terminam a sua execução normalmente ou essa execução é interrompida?

4. [1 valor] Quando existe execução de instruções fora de ordem, o resultado do programa pode ser diferente de quando as instruções são executadas estritamente por ordem? *não porque pode originar dependências e causar conflitos*

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 1. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

5. Considere que o significado e o efeito do código MIPS seguinte são exactamente aqueles que teria se fosse executado na implementação monociclo do processador (onde não existem *delay slots*). No fim da execução do código, os valores presentes nos registos usados não são importantes.

```
1.      beq    $a0, $0, fim
2.      ciclo: lw    $t0, 0($a1)
3.          lw    $t1, 0($a2)
4.          addu   $t0, $t0, $t1
5.          sw    $t0, 0($a3)
6.          addiu  $a1, $a1, 4
7.          addiu  $a2, $a2, 4
8.          addiu  $a3, $a3, 4
9.          addi   $a0, $a0, -1
10.         bne   $a0, $0, ciclo
11.         fim:   jr    $ra
```

(a) [3 valores] Identifique todas as dependências (de dados) existentes no código apresentado.

(b) [4 valores] Simule a execução do código num processador com *forwarding*, com decisão dos saltos condicionais no andar ID, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, assumindo que o valor inicial no registo \$a0 é 1. Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

Quantos ciclos de relógio são necessários para executar o código nas condições acima?

Quantos ciclos de relógio seriam necessários para executar o código se o ciclo (instruções 2 a 10) fosse executado 1000 vezes?

(CONTINUA...)

- (c) [2,5 valores] Altere o código apresentado, reordenando as instruções e, se considerar útil, modificando o offset das instruções de acesso à memória, de modo a eliminar o maior número possível de atrasos e de ciclos desperdiçados durante a sua execução no pipeline com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

6. [2 valores] Durante a execução de um programa, o conteúdo do pipeline MIPS da Figura 1, num determinado ciclo de relógio, é o seguinte:

IF	ID	EX	MEM	WB
or \$t5, \$t4, \$t6	addu \$t2, \$t3, \$t7	addi \$t1, \$t0, 5	beq \$0, \$0, 11	sw \$s0, 0(\$s1)

Diga qual deverá ser o valor dos sinais de controlo *em uso*, neste ciclo, em cada um dos andares, e qual a operação realizada pela ALU. (Não é necessário referir o valor de ALUOp.)

7. [2 valores] Se as latências dos andares do pipeline forem as apresentadas abaixo, qual o andar que determina a frequência máxima a que poderá funcionar o relógio do processador e qual é essa frequência?

Andar	IF	ID	EX	MEM	WB
Latência	360 ps	235 ps	385 ps	365 ps	255 ps

ILP

8. [2,5 valores] Organize o código original da pergunta 5, introduzindo as alterações que considerar convenientes, para ser executado no pipeline MIPS *double issue* (com *forwarding*, decisão dos saltos condicionais no andar ID, previsão perfeita e sem *delay slot*), em que cada *issue packet* pode conter uma instrução aritmética ou de salto, e uma instrução de acesso à memória, de modo a não haver a necessidade da introdução de atrasos durante a sua execução.

Arquitectura de Sistemas e Computadores II

2^a Frequência

Departamento de Informática
Universidade de Évora

28 de Novembro de 2018

Indique todos os cálculos efectuados

Perguntas rápidas

- [1 valor] No *pipeline* MIPS, a leitura dos registos a usar por uma instrução acontece antes, em simultâneo, ou depois da descodificação da instrução?
- [1 valor] Quando, no *pipeline* MIPS, uma instrução é atrasada um ciclo devido a um conflito de dados que a envolve, é a entrada da instrução no *pipeline* que é atrasada, ou a instrução fica um ciclo adicional nalgum andar do *pipeline*?
- [1 valor] Se a instrução que está no andar EX do *pipeline* MIPS gera uma excepção, o que acontece à instrução está no andar IF quando a excepção é gerada?
- [1 valor] Numa cache com *no write-allocate*, quando há um *miss* numa operação de escrita, o conteúdo da cache pode ser alterado devido a essa operação ou fica na mesma?

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 1. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

- Considere que o significado e o efeito do código MIPS seguinte são exactamente aqueles que teria se fosse executado na implementação monociclo do processador (onde não existem *delay slots*). No fim da execução do código, os valores presentes nos registos usados não são importantes, excepto o do registo \$v0.

```
1.          ori    $t1, $0, 0
2.          or     $t0, $t1, $a2
3.  ciclo:   beq    $a1, $0, fim -
4.          lw     $t2, 0($a2)
5.          bne    $t2, $a0, copia
6.          addi   $t1, $t1, 1
7.          beq    $0, $0, avanca
8.  copia:   addiu  $t0, $t0, 4
9.          sw     $t2, -4($t0)
10.  avanca:  addiu $a2, $a2, 4
11.          addi   $a1, $a1, -1
12.          beq    $0, $0, ciclo .
13.  fim:    or     $v0, $t1, $0
14.          jr     $ra
```

- [2,5 valores] Identifique todas as dependências (de dados) existentes no código apresentado.

(CONTINUA...)

- (b) [3 valores] Simule a execução do código apresentado num processador com *forwarding*, com decisão dos saltos condicionais no andar ID, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, com o salto correspondente à instrução 5 a ser efectuado e o correspondente à instrução 3 a só ser efectuado na segunda vez que a instrução for executada.

Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

- (c) [2,5 valores] Altere o código apresentado, reordenando as instruções e, se considerar útil, modificando o offset das instruções de acesso à memória, de modo a eliminar o maior número possível de atrasos e de ciclos desperdiçados durante a sua execução no *pipeline* com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

ILP

6. [2,5 valores] Organize o código original da pergunta 5, introduzindo as alterações que considerar convenientes, para ser executado no *pipeline MIPS double issue* (com *forwarding*, decisão dos saltos condicionais no andar ID, previsão perfeita e sem *delay slot*), em que cada *issue packet* pode conter uma instrução aritmética ou de salto, e uma instrução de acesso à memória, de modo a não haver a necessidade da introdução de atrasos durante a sua execução.

Caches

7. Considere uma cache 3-way set associative, com dois conjuntos, blocos de 2 palavras, palavras de 32 bits, e aplicando a estratégia LRU para a escolha do bloco a substituir. O conteúdo dessa cache é parcialmente apresentado na figura seguinte (onde $M[p]$ representa o valor da palavra de memória p):

Índice	Valid	Tag	Palavras			Valid	Tag	Palavras			Valid	Tag	Palavras		
0	1	6	$M[24]$			0					6				
1	1	4				1			$M[15]$	0					

- (a) [2,5 valores] Complete o que puder da figura.

- (b) [3 valores] Simule o funcionamento da cache, a partir do estado apresentado acima, para a sequência de acessos aos endereços seguintes:

103 94 62 40 74

Assuma que o último bloco da cache acedido foi o que se encontra na posição central, no conjunto com índice 1. Para cada acesso, indique a palavra acedida, o número do bloco a que pertence a palavra, o índice da posição da cache que irá ocupar, o tag, se há um hit ou um miss e, quando aplicável, o número do bloco que será substituído. Apresente o conteúdo final da cache, tão completo quanto possível, e calcule a miss rate verificada.

Arquitectura de Sistemas e Computadores II

1^a Frequência

Departamento de Informática
Universidade de Évora

31 de Outubro de 2018

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] Se um programa demora mais tempo a executar no computador X do que no computador Y , qual o computador que apresenta melhor desempenho para esse programa?
2. [1 valor] O que é igual em dois processadores que implementam a mesma arquitectura, o CPI, as instruções que executam, ou a frequência do relógio?
3. [1 valor] Na implementação MIPS monociclo, a execução da instrução `and` é mais rápida do que a da instrução `sw`?
4. [1 valor] Qual o *speedup* máximo que se pode esperar obter ao passar de uma implementação monociclo de um processador para uma implementação *pipelined* com 8 andares?

Desempenho

5. Durante a execução do programa P são executadas 2 mil milhões de instruções, com a seguinte distribuição:

Classe	Aritméticas	Leitura da memória	Escrita na memória	Saltos
%	40	25	10	25
CPI	2	6	5	4

- (a) [1,5 valores] Qual o CPI de P ?
- (b) [2 valores] Se a execução do programa demorar 7,6 s, qual é o período do relógio do processador? (Se não resolveu a alínea anterior e precisar de saber o CPI de P , use o valor 15,2.)
6. [1,5 valores] Quantas vezes é necessário reduzir o tempo das instruções correspondentes a 80% do tempo total de execução de um programa para obter um *speedup* de 2,5?

Implementação MIPS monociclo

7. [6 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução `addipc` (*add immediate to PC*), que é uma instrução tipo-I com dois argumentos:

addipc rt, immediate	31	0	0	0
	bits	6	5	5

Esta instrução coloca, no registo `rt`, o valor obtido somando o endereço da instrução a `immediate` \times 4, sendo `immediate` um valor com sinal.

- (a) Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução? (Identifique um *multiplexer* através do sinal que o controla.)
- (b) Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- (c) Quais os valores que os vários sinais de controlo deverão ter e qual a operação realizada pela ALU durante a execução desta instrução? (Não é necessário apresentar o valor de ALUOp.)
- (d) Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

8. Seja PC = 004C1FA8₁₆ o endereço da instrução sub \$t0, \$v1, \$t0, cuja codificação binária completa é a seguinte:

bit	31	26	21	16	11	6	0
	000000	00011	01000	01000	00000	100010	
	op code	rs	rt	rd	shamt	funct	

- (a) [2,5 valores] Sejam os seguintes os valores contidos em alguns dos registos do processador, quando a execução da instrução se inicia:

Registo	1	3	4	8	10	11	16	23	24	31
Valor	88	100	3201	122	56	0	32	93000	4	4444444

Indique os valores que estão presentes, no fim do ciclo em que a instrução executa, nos pontos A, B, C, e D do circuito da Figura 2. Use a base de numeração que achar conveniente para cada um dos valores.

(Se necessitar do conteúdo de um registo não contemplado na tabela acima, considere que esse registo contém o valor obtido adicionando 1000 ao número do registo.)

- (b) [2,5 valores] Sejam as seguintes as latências das várias componentes do processador:

PC	Memórias	Banco registos	ALU	Somadores	Shift left 2	Extensão com sinal	Multiplexers	Controlo	Controlo da ALU
10 ps	250 ps	100 ps	200 ps	180 ps	2 ps	3 ps	25 ps	35 ps	13 ps

(Considere que os restantes elementos lógicos têm latência zero.)

Calcule o tempo que demora, desde o início do ciclo de relógio em que a instrução é executada, até que os valores correctos estejam disponíveis nos pontos A, B, C e D do circuito da Figura 2. Explique todos os tempos que considerou, nos cálculos que fez, para chegar aos valores que obteve.

(Sugestão: Inclua esses valores na figura.)

Arquitectura de Sistemas e Computadores II

Exame de Recurso

Departamento de Informática
Universidade de Évora

25 de Janeiro de 2018

Indique todos os cálculos efectuados

Perguntas rápidas

1. [0,5 valores] O CPI de um programa é uma característica da arquitectura ou do processador que a implementa?
2. [0,5 valores] Nos processadores reais, a que correspondem as duas memórias distintas, de instruções e de dados, visíveis nos diagramas de blocos do MIPS?
3. [0,5 valores] De que tipo (ou tipos) de localidade de acessos é possível tirar partido com uma cache em que os blocos só têm uma palavra?
4. [0,5 valores] A página física em que reside uma dada página virtual de um processo pode estar no TLB e não se encontrar na tabela de páginas do processo?

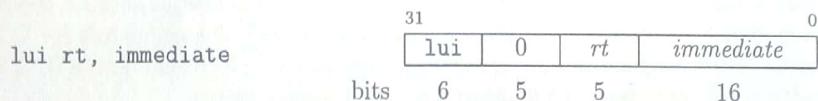
Desempenho

5. [2 valores] Durante a execução de um programa, que demora 30 s, são executadas 10^{10} instruções, com a distribuição apresentada na tabela abaixo. Qual a frequência do relógio do processador em que o programa é executado?

Classe	A	B	C	D
%	40	35	15	10
CPI	6	10	12	13

Implementação MIPS monociclo

6. [4 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução lui (*load upper immediate*), que é uma instrução tipo-I com dois argumentos:



Esta instrução coloca no registo rt o valor $\text{immediate} \times 2^{16}$ (=

immediate	dezasseis 0's
-----------	---------------

).

- (a) Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?
- (b) Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- (c) Quais os valores que os vários sinais de controlo deverão ter e qual a operação realizada pela ALU durante a execução desta instrução? (Não é necessário apresentar o valor de ALUOp.)
- (d) Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

(CONTINUA...)

Pipeline MIPS de 5 andares

7. [2 valores] Altere o código à direita (escrito sem ter em conta a existência de *delay slots*) de modo a poder ser executado, sem a introdução de qualquer atraso, num processador com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *delay slot*.

Identifique os conflitos de dados presentes no código alterado, diga quais os registos cujos valores terão de ser *forwarded* e entre que andares do *pipeline* será feito o *forwarding*.

```
1.      or    $v0, $0, $0
2.      beq   $a1, $0, fim
3.      ciclo: lw    $t2, 0($a0)
4.          add   $v0, $v0, $t2
5.          addiu $a0, $a0, 4
6.          addi   $a1, $a1, -1
7.          bne   $a1, $0, ciclo
8.      fim:   jr    $ra
```

Cache

8. [2 valores] Considere um sistema com palavras e endereços de 32 bits, com uma cache 8-way set associative, com blocos de 64 bytes, e em que os blocos ocupam um total de 16 KB.

Após ter sido feito um acesso ao endereço $04DC\ C7C4_{16}$, em que índice da cache se encontrará a palavra acedida, quantos bits terá o *tag* correspondente e qual o seu valor, e que outras palavras se encontrarão mesma posição?

Memória virtual

9. [2 valores] Partindo da tabela de páginas com o conteúdo parcialmente mostrado e de um TLB (*fully associative*, com 4 blocos de uma tradução e substituição por LRU) vazio, apresente o conteúdo do TLB depois da seguinte sequência de operações:

1. Leitura de uma posição de memória da página virtual 35;
2. Leitura de uma posição de memória da página virtual 36;
3. Leitura de uma posição de memória da página virtual 37;
4. Escrita de uma posição de memória da página virtual 36.

Tabela de páginas

Pág. virtual	Dirty	Pág. física
35	0	98
36	0	13
37	1	100
38	0	123
39	0	75
		...

10. [2 valores] Num sistema, em que as tabelas de páginas têm dois níveis, os endereços virtuais têm 40 bits e uma página contém 32 KB. Se a tabela de primeiro nível tiver 4096 posições, quantas posições terão as tabelas de segundo nível?

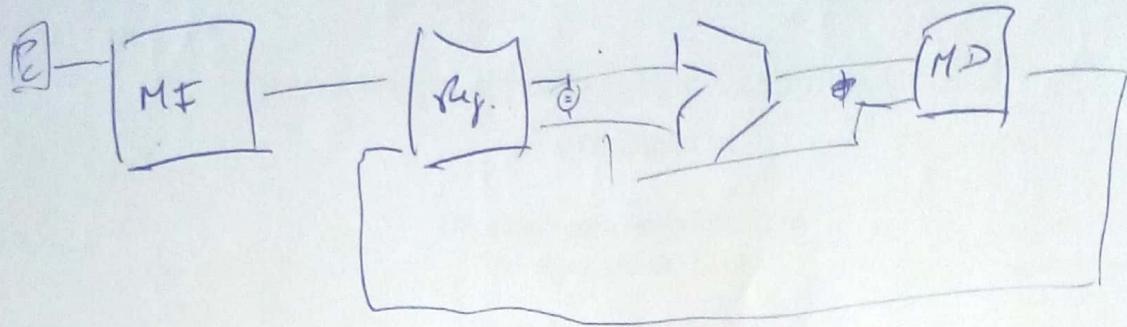
Multiprocessamento

11. [2 valores] A execução sequencial de um programa demora 20 s, em que 2,5% correspondem à parte não paralelizável do programa. Qual o *speedup* obtido se o programa for paralelizado e executado em 40 processadores, se um dos processadores ficar com 2% do trabalho, sendo o restante distribuído igualmente pelos outros processadores?

12. [2 valores] O uso de *semáforos* é uma das técnicas para controlar o acesso às secções críticas de um programa paralelo. Os semáforos podem ter o valor 0 ou 1 e são manipulados através das primitivas P e V. A primitiva P, que é usada à entrada da secção crítica, espera que o valor do semáforo seja 1 e põe o semáforo a 0. A primitiva V limita-se a repor o valor do semáforo a 1, para permitir de novo o acesso à secção crítica.

Apresente o código MIPS que implementa a primitiva P, como uma função, que recebe o endereço da posição de memória que contém o valor do semáforo no registo \$a0.

IF ID EX MEM WB



1: slt

2: beg slt

3: lw beg slt(sto,...)

4: lw beg(sto,..) selne slt(sto,...)

5: lw lw beg Cselne slt

	1°	2°	3°	4°	5°		1°	2°	3°	4°	5°
slt	IF	ID	EX	MEM	WB		slt	if	id	ex	mem
beg	IF	ID	ID	EX			beg	if	id	id	
lw	IF	IF	ID				lw				
lw				IF							

Arquitectura de Sistemas e Computadores II

2^a Frequência

Departamento de Informática

Universidade de Évora

23 de Novembro de 2017

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] A execução *pipelined* é uma técnica que aumenta o *throughput* do processador ou que diminui o tempo que uma instrução demora a executar?

2. [1 valor] Quantos ciclos de relógio demora a execução da instrução *sw* no *pipeline* MIPS de cinco andares, na ausência de atrasos devido a acessos à memória?

3. [1 valor] Se a instrução que está no andar *MEM* do *pipeline* MIPS gera uma excepção, o que pode acontecer a essa instrução quando o tratamento da excepção termina? Ser reexecutada desde o início, ser reexecutada a partir do andar *MEM*, ou não voltar a ser executada, devido ao programa ser abortado?

4. [1 valor] Quantas tabelas de páginas existem, num sistema com memória virtual, se, além do sistema operativo, estiverem em execução, em simultâneo, 5 programas?

Pipeline MIPS de 5 andares

Para este grupo, use como referência o *pipeline* da Figura 1. Tenha, no entanto, em atenção as caracterizações do funcionamento do *pipeline* feitas nas várias alíneas.

5. Considere que o significado e o efeito do código MIPS seguinte são exactamente aqueles que teria se fosse executado na implementação monociclo do processador (onde não existem *delay slots*). No fim da execução do código, os valores presentes nos registos usados não são importantes.

```
1.    ciclo: slt    $t0, $a0, $a1
2.          beq    $t0, $0, fim
3.          lw     $t1, 0($a0)
4.          lw     $t2, 0($a1)
5.          sw     $t2, 0($a0)
6.          sw     $t1, 0($a1)
7.          addiu $a0, $a0, 4
8.          addiu $a1, $a1, -4
9.          beq    $0, $0, ciclo
10.         fim:   jr     $ra
```

(a) [2,5 valores] Identifique todas as dependências (de dados) existentes no código apresentado.

(b) [3 valores] Simule a execução do código apresentado num processador com *forwarding*, com decisão dos saltos condicionais no andar *ID*, com previsão perfeita do resultado das instruções de salto condicional e sem *delay slots*, com o salto correspondente à instrução 2 a ser efectuado na segunda vez que a instrução for executada. Apresente a evolução do estado do *pipeline* durante a execução, indicando todos os atrasos introduzidos e todos os pontos onde foi necessário o *forwarding* de algum valor, identificando claramente entre que andares o *forwarding* foi feito.

(CONTINUA...)

- (c) [2,5 valores] Altere o código apresentado, reordenando as instruções e, se considerar útil, modificando o *offset* das instruções de acesso à memória, de modo a eliminar o maior número possível de atrasos e de ciclos desperdiçados durante a sua execução no *pipeline* com *forwarding*, com decisão dos saltos condicionais no andar ID e com um *branch delay slot*.

ILP

6. [2,5 valores] Organize o código original da pergunta 5, introduzindo as alterações que considerar convenientes, para ser executado no *pipeline* MIPS *double issue* (com *forwarding*, decisão dos saltos condicionais no andar ID, previsão perfeita e sem *delay slot*), em que cada *issue packet* pode conter uma instrução aritmética ou de salto, e uma instrução de acesso à memória, de modo a não haver a necessidade da introdução de atrasos durante a sua execução.

Caches

7. Considere uma cache 2-way *set associative*, com dois conjuntos, blocos de 2 palavras, palavras de 64 ~~bytes~~ bits, e aplicando a estratégia LRU para a escolha do bloco a substituir. O conteúdo dessa cache é parcialmente apresentado na figura seguinte (onde $M[p]$ representa o valor da palavra de memória p):

Índice	Valid		Tag	Palavras		Valid		Tag	Palavras	
	0	1			M[21]	1	2		M[8]	
0	1	.								
1	1	3				0				

- (a) [2,5 valores] Complete o que puder da figura.
 (b) [3 valores] Simule funcionamento da cache, a partir do estado apresentado acima, para a sequência de acessos aos endereços seguintes:

74 170 160 80 100

Assuma que o último bloco da cache acedido foi o que se encontra na posição da direita, no conjunto com índice 0. Para cada acesso, indique a palavra acedida, o número do bloco a que pertence a palavra, o índice da posição da cache que irá ocupar, o *tag*, se há um *hit* ou um *miss* e, quando aplicável, o número do bloco que será substituído. Apresente o conteúdo final da cache, tão completo quanto possível, e calcule a *miss rate* verificada.

Arquitectura de Sistemas e Computadores II

1^a Frequência

Departamento de Informática
Universidade de Évora

19 de Outubro de 2017

Indique todos os cálculos efectuados

Perguntas rápidas

1. [1 valor] Um processador pode apresentar melhor desempenho do que outro processador cujo relógio funciona a uma frequência 5 vezes superior?
2. [1 valor] Se a execução de um programa é mais rápida no computador X do que no computador Y , pode-se concluir que a execução de qualquer programa é mais rápida em X do que em Y ?
3. [1 valor] Demora mais tempo a execução de 1000 instruções `lw` na implementação MIPS monociclo ou a execução de 1000 instruções `add`?
4. [1 valor] Quantas instruções podem estar a ser executadas no *pipeline* MIPS em simultâneo, no máximo?

Desempenho

5. Seja A um computador com uma frequência de relógio de 800 MHz. Durante a execução do programa P em A , que demora 4 s, são executadas 1000 milhões de instruções.

(a) [2 valores] Qual o CPI de P ?

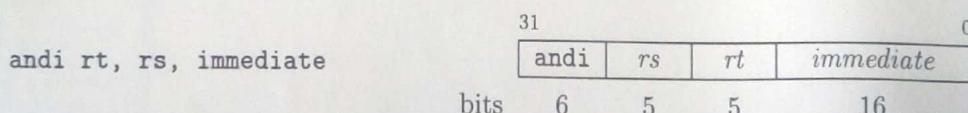
(b) [1,5 valores] Se a distribuição das instruções executadas for a apresentada na tabela abaixo, qual o CPI das instruções de salto?

Classe	Aritméticas	Acesso à memória	Saltos
%	30	40	30
CPI	1	5	?

(c) [1,5 valores] Qual é o *speedup* que se obtém se for possível reduzir o CPI das instruções de acesso à memória em 40%?

Implementação MIPS monociclo

6. [6 valores] Pretende-se que a implementação MIPS monociclo da Figura 1 suporte a execução da instrução `andi` (*and immediate*), que é uma instrução tipo-I com três argumentos:



O efeito desta instrução é colocar, no registo `rt`, o valor do E-lógico *bit a bit* entre o conteúdo do registo `rs` e `immediate`, que é um número sem sinal.

(a) Quais das unidades funcionais e dos *multiplexers* existentes serão usados na execução desta instrução?

(CONTINUA...)

- (b) Que unidades funcionais (incluindo *multiplexers*) e que sinais de controlo é necessário acrescentar?
- (c) Quais os valores que os vários sinais de controlo deverão ter durante a execução desta instrução?
(Não precisa de indicar o valor de ALUOp, basta dizer qual será a operação efectuada pela ALU durante a execução desta instrução.)
- (d) Apresente na Figura 1 as alterações à implementação que considerar necessário fazer.

7. Seja PC = 0040 003C₁₆ o endereço da instrução *beq* cuja codificação binária completa é a seguinte:

bit	31	26	21	16	0
	000100	00101	11000	11111111 11110000	

- (a) [2,5 valores] Sejam os seguintes os valores contidos em alguns dos registo do processador, quando a execução da instrução se inicia:

Registo	1	5	6	8	10	12	16	23	24	31
Valor	1	37	413	9931	9	100	310	4193000	37	4194304

Indique os valores que estão presentes, no fim do ciclo em que a instrução executa, nos pontos Ⓐ, Ⓑ, Ⓒ, Ⓓ do circuito da Figura 2.

(Se necessitar do conteúdo de um registo não contemplado na tabela acima, considere que esse registo contém o valor obtido adicionando 1000 ao número do registo.)

- (b) [2,5 valores] Sejam as seguintes as latências das várias componentes do processador:

PC	Memória	Banco registo	ALU	Somadores	Shift left 2	Extensão com sinal	Multiplexers	Controlo	Controlo da ALU
10 ps	300 ps	120 ps	220 ps	190 ps	2 ps	3 ps	30 ps	40 ps	15 ps

(Considere que os restantes elementos lógicos têm latência zero.)

Calcule o tempo que demora, desde o início do ciclo de relógio em que a instrução é executada, até que os valores correctos estejam disponíveis nos pontos Ⓐ, Ⓑ, Ⓒ e Ⓓ do circuito da Figura 2. Explicite todos os tempos que considerou, nos cálculos que fez, para chegar aos valores que obteve.

(Sugestão: Inclua esses valores na figura.)

3ª freq (11 janeiro 2017)

Perguntas rápidas

3 → menor miss rate no cache fully associative

4 → write-back

5 → execução Page Fault

6 → implícita.

cache

10. 32 bits

● 8 palavras

→ blocos 2 palavras

$$\text{miss rate} = \frac{6}{8}$$

endereço	48	56	52	48	170	60	24	14
palavra	12	19	13	12	30	15	6	9
bloco	6	7	6	6	15	7	3	2
índice	2	3	2	2	3	3	3	2
tag	1	1	1	1	3	1	0	0
H/M	M	M	M	M	M	M	M	M
sust	-	-	-	-	-	7	-	76

11.

16 conjuntos

blocos de 4 palavras

$$\text{tag} = 20_{16}$$

$$20_{16} = 32$$

$$\begin{aligned} \text{palavra} &= \text{bloco} \times \frac{\text{nº palavras}}{\text{bloco}} \\ &= 517 \times 4 \\ &= 2068_{10} \end{aligned}$$

bloco nº entradas
Tag
índice

n | 16
32

5

$$\underline{\underline{n = 517}}$$

2068 2069 2070 ...

12.

	valid	dirty	tag	Pag. R.
0	1	0	7/8	19/23
1	1	0	7	12
2	1	0	7	5
3	1	0	7	

(29) $\text{TLB} = 29 \times 4 = 1$
tag 1

(30) $\text{TLB} = 30 \times 4 = 2$
tag = 7

(31) $\text{TLB} = 31 \times 4 = 3$
tag = 7

(33) $\text{TLB} = 33 \times 4 = 9$
tag = 8