

Segundo teste – 30 de maio de 2014 – Duração 1 hora e 30 minutos

Responda às questões no espaço reservado para o efeito após cada alínea, ou na última página caso necessite mais espaço. Apresente todos os cálculos que efectuar, e esclareça qualquer pressuposto que tenha de fazer para resolver uma questão. Qualquer caso de fraude ou plágio será punido com a reprovação à disciplina, de acordo com os regulamentos em vigor.

Número		Nome	
--------	--	------	--

Grupo I (10 valores)

1. Pretende-se adicionar ao conjunto de instruções do MIPS uma instrução chamada **loadSum** que efectua a soma entre um valor lido da memória e um valor imediato, armazenando o resultado num registo.

A especificação da instrução “**loadSum rt, rs, imed**” é a seguinte:

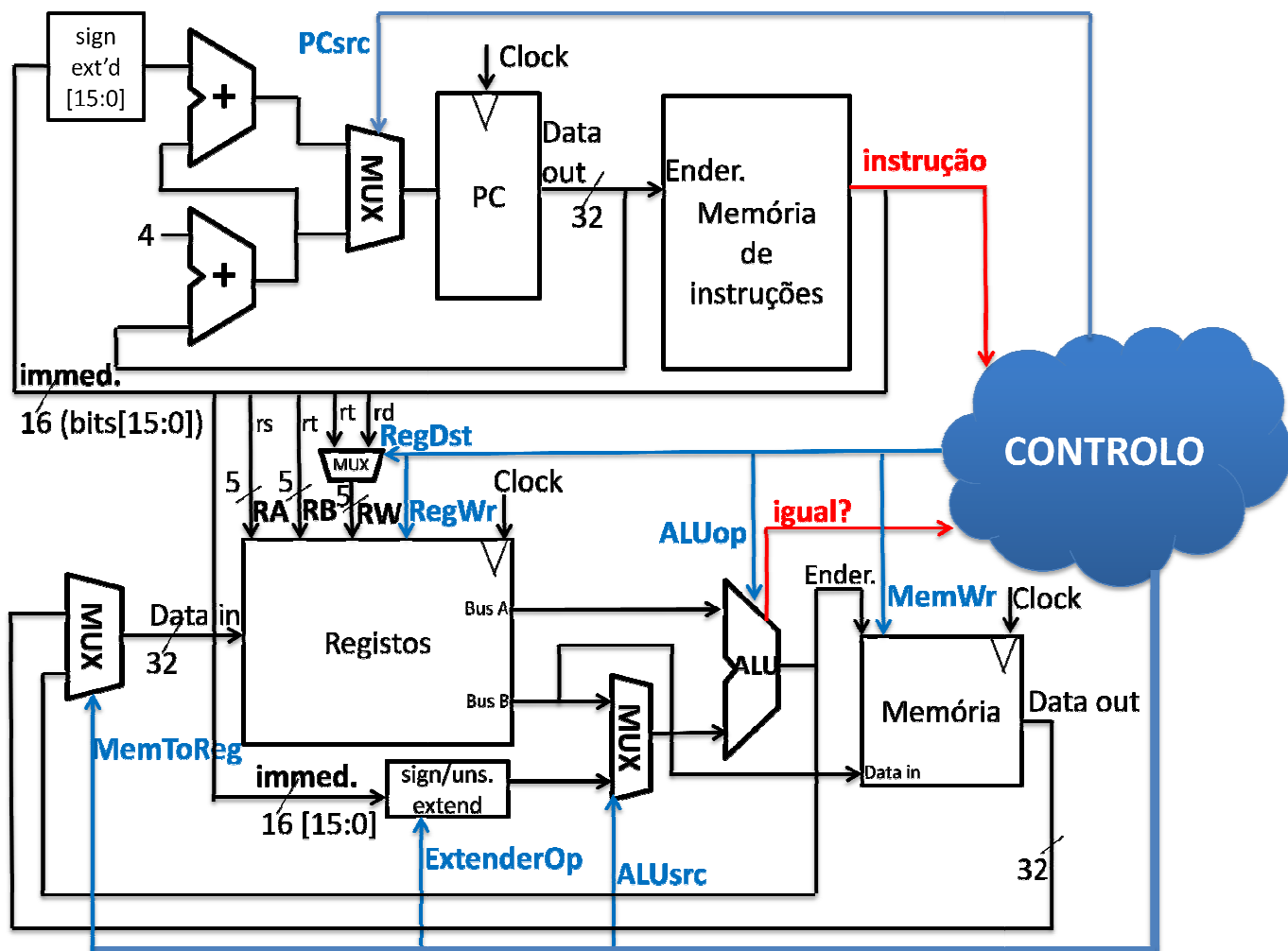
$\$pc \leftarrow \$pc + 4$

$R[rt] \leftarrow MEM[rs] + ext_sinal(imed)$

- a) Qual o formato (R, I, ou J) da nova instrução em linguagem máquina MIPS? Justifique brevemente.

- b) Para suportar esta instrução, quais as alterações que são necessárias efectuar ao hardware do CPU de ciclo único? Deverá em particular indicar quais os componentes novos que têm de ser adicionados ou modificados. Pode adicionar *multiplexers* e fios, mas **não pode adicionar novos somadores**. A resposta está dividida em três partes:

- b.1) Efectue as alterações directamente no circuito seguinte.



b.2) Liste os componentes que foram adicionados e modificados, e para cada componente indique claramente a que entradas ou saídas de outros componentes estão ligados.

b.3) Liste as novas linhas de controlo que foram adicionadas.

c) Qual é o valor para todas as linhas de controlo do CPU quando esta instrução estiver a ser executada? Deve indicar o valor da respectiva linha (0 ou 1) ou X no caso de ser indiferente (*"don't care"*). Deve também adicionar à tabela abaixo as novas linhas de controlo ou indicar as alterações que fez nas linhas existentes.

Número:

Nome:

PCSrc	RegDst	RegWr	ALUSrc	MemWr
0=PC+4; 1=PC+4+(sign_ext (immed)<<2)	0=rt (bits [20:16]); 1=rd (bits [15:11])	0=Não escrever; 1=escrever (registro RW)	0=RegB ; 1=imediato (bits menos signif. instr.)	0=Não escrever 1=escrever (memória)
valor: _____				

MemToReg	ExtenderOp	ALUop		
0=da ALU; 1=da memória	0=zero extend; 1=sign extend	00=ADD; 01=SUB; 10=OR;11=COMPARE		

d) Quais as consequências de introduzir esta nova instrução “loadSum” na versão do CPU com “*pipelining*” que estudámos na cadeira? (Escolha **apenas uma** das opções – Respostas erradas descontam 1/5 da cotação.)

- ☐ Não altera o desenho do CPU com pipelining, bastando juntar o hardware novo a um dos patamares, podendo tal afetar ou não a duração do ciclo de relógio.
- ☐ Altera, pois pode gerar novos conflitos de controlo (*control hazards*).
- ☐ Altera, pois a ordem de execução dos patamares (*stages*) é diferente nesta instrução.
- ☐ Altera necessariamente a frequência do CPU, pois ao adicionar hardware temos sempre de aumentar a duração do ciclo de relógio.
- ☐ Altera, pois deixa de ser necessário o forwarding para minimizar os conflitos de dados.

2. Qual das seguintes afirmações é **falsa** sobre o protocolo MESI de coerência de caches em multi-cores? (Escolha **apenas uma** das opções – Respostas erradas descontam 1/5 da cotação.)

- ☐ Um processador com uma dada entrada no estado "partilhado" tem de escutar ("*snoop*") a interligação para determinar se algum outro processador quer escrever nesses mesmos dados.
- ☐ Se um processador tem uma entrada no estado "modificado", nenhum outro processador pode ter uma entrada para o mesmo endereço no estado "partilhado".
- ☐ Um *store* (escrita) para uma entrada que está em cache num estado diferente de inválido é servido inteiramente pela cache, isto é, não necessita nenhuma comunicação pela interligação com outros processadores.
- ☐ Se existirem em simultâneo duas entradas em processadores diferentes para o mesmo endereço, elas têm de estar ambas no estado "partilhado" e ter o mesmo valor.
- ☐ Em cada instante, apenas um processador pode ter um dado endereço em cache no estado "exclusivo".

3. Qual das seguintes métricas de performance é melhorada (ou seja, diminui) através da introdução de *pipelining* comparado com o CPU de ciclo único? (Escolha **apenas uma** das opções – Respostas erradas descontam 1/5 da cotação.)

- ☐ Tempo (latência) de execução de uma única instrução (desde que a instrução é lida da memória até à escrita dos resultados nos registos estar concluída).
- ☐ Tempo médio de acesso à memória (métrica AMAT).
- ☐ Tempo de execução de um programa.
- ☐ Número total de instruções de um programa.
- ☐ Tempo (latência) para trazer os dados para cache após um *cache miss*.

Grupo II (10 valores)

4. Considere uma arquitetura de um CPU com endereços virtuais de 32 bits e palavras de um Byte. Considere também que esta arquitetura usa uma cache de mapeamento direto, em que cada bloco de cache contém 16 Bytes, e usa uma política “write-back”.

- a) Supondo que a capacidade da cache (ou seja, o total de dados armazenáveis em cache) é de 4K Bytes, quantos bits são usados para cada campo do endereço (tag, índice, e deslocamento)? Apresente e justifique os cálculos que efectuar.

- b) Independentemente da sua resposta à alínea anterior, considere agora a seguinte partição de endereços: 24 bits para a tag, 4 para o índice, e 4 para o deslocamento. Supondo que a cache inicialmente se encontrava vazia, isto é, todas as entradas eram inválidas, indique, para cada um dos acessos seguintes, se é um *hit* ou um *miss*, no caso do *miss* qual é o tipo de *miss* (capacidade, conflito, ou obrigatório), se gera uma escrita para a memória (*write-back*) e qual é o conteúdo dos campos válido, dirty, e tag (em hexadecimal) das duas primeiras linhas da cache após o acesso.

1 – Load do endereço: ABBA0010 (hex)

hit/miss: tipo de miss:

write back (sim/não):

conteúdo das duas primeiras linhas da cache após o acesso:

Índice	válido?	Tag	dirty
0			
1			

2 - Store do endereço: CABE0014 (hex)

hit/miss: tipo de miss:

Número:

Nome:

write back (sim/não):

conteúdo das duas primeiras linhas da cache após o acesso:

Índice	válido?	tag	dirty
0			
1			

3 - Store do endereço: CADA0012 (hex)

hit/miss: tipo de miss:

write back (sim/não):

conteúdo das duas primeiras linhas da cache após o acesso:

Índice	válido?	tag	dirty
0			
1			

4 - Load do endereço: ABBA0010 (hex)

hit/miss: tipo de miss:

write back (sim/não):

conteúdo das duas primeiras linhas da cache após o acesso:

Índice	válido?	tag	dirty
0			
1			

5 - Load ao endereço: CADA0010 (hex)

hit/miss: tipo de miss:

write back (sim/não):

conteúdo das duas primeiras linhas da cache após o acesso:

Índice	válido?	tag	Dirty
0			
1			

c) Qual o *hit rate* desta sequência de acessos? Apresente os cálculos que efectuar.

d) Considere a possibilidade de se duplicar a dimensão do bloco, mantendo a capacidade da cache. Estenda esta sequência com mais dois acessos (liste apenas os endereços destes dois acessos)

em hexadecimal) de forma a que o **hit rate seja inferior com a dimensão de bloco duplicada** do que com a dimensão original.

--

- e) Se a cache tivesse a mesma capacidade (ou seja, mantendo a quantidade total de dados armazenáveis em cache) e a dimensão do bloco da alínea b), mas fosse associativa por conjuntos com duas vias e política FIFO, qual seria a sequência de hit/miss e o hit rate? Responda preenchendo a seguinte tabela:

Endereço	Hit / Miss
ABBA0010	
CABE0014	
CADA0012	
ABBA0010	
CADA0010	

Hit rate:	
-----------	--

- f) A resposta da alínea anterior mudaria se a política fosse LRU em vez de FIFO?

Sim ☐

Não ☐

Espaço adicional para a resolução: