

Universidade Federal de Santa Catarina, INE/CTC  
INE 5366 – Arquitetura de Computadores I  
**Segunda avaliação 2007-2**

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

**Parte I - Compreensão de conceitos básicos [3,0 pontos]**

1. [valor:  $6 \times 0,5 = 3,0$ ] Responda sucintamente as perguntas abaixo:

- a) Seja o diagrama de transição de estados mostrado na Figura 1 para o datapath do Anexo IV. Afirmção: “Sem nenhuma alteração no datapath, pode-se reduzir o número de ciclos gastos nas instruções lw e sw para 4 e 3 ciclos, respectivamente, modificando-se a unidade de controle de forma a realizar o cômputo de endereços no estado 1, às custas de um ciclo extra para a execução de instruções de desvio condicional.” A afirmação é verdadeira ou falsa? **Justifique.**

**Resposta:** \_\_\_\_\_ **Justificativa:** \_\_\_\_\_

- b) A instrução add \$s1, \$s2, \$s3 está em execução no datapath do Anexo IV, controlado pela máquina de estados da Figura 1. Os conteúdos dos registradores \$s1, \$s2 e \$s3, imediatamente antes de esta instrução iniciar sua execução, eram 0x00000000, 0x7FFFFFFF e 0x00000001, respectivamente. Afirmção: “No instante em que a próxima instrução é buscada, o conteúdo de \$s1 é 0x80000000 e o de PC é 0xC0000000” A afirmação é verdadeira ou falsa? **Justifique**, indicando a sequência de estados percorridos (na execução de add até a próxima instrução ser buscada) e invocando evento(s) relevante(s) ocorrido(s) no datapath.

**Resposta: Sequência de estados:** \_\_\_\_\_

**Justificativa:** \_\_\_\_\_

- c) Para o datapath monociclo do Anexo III, sejam: IM e DM as memórias de instruções e de dados, respectivamente; REG o banco de registradores; ALU a unidade lógica e aritmética; TAD o somador que calcula o endereço-alvo de um desvio e INC o somador que incrementa o PC de 4. Sejam os seguintes tempos de acesso ou de propagação:  $t(IM)=t(DM)=t(ALU)=200ps$ ;  $t(REG)=100ps$ ;  $t(TAD)=t(INC)=500ps$  (todos os outros componentes têm tempo nulo), onde  $ps = 10^{-12}$  s. Afirmção: “A frequência máxima de relógio para a operação desse datapath é de 1,25 GHz”. A afirmação é verdadeira ou falsa? **Justifique** indicando os nomes das unidades funcionais no caminho crítico e seu atraso de propagação total.

**Resposta:** \_\_\_\_\_ **Justificativa:** \_\_\_\_\_

- d) Em uma máquina com suporte de hardware para especulação, a instrução lw do código abaixo executa antes da instrução sub. Em que condições o valor lido por lw será escrito no registrador \$s3?

sub \$s2, \$t1, \$t2

Resposta:

sw \$s1, 0(\$s2)

bne \$s1, \$zero, L

lw \$s3, (\$s4)

- e) A variável x é alocada em um registrador \$s0 em um pipeline que possui um caminho de forwarding permitindo que a saída da ALU seja disponibilizada como uma de suas entradas no ciclo seguinte. Afirmção: “Para o código abaixo, todas as instruções lêem o valor atribuído a x diretamente do banco de registradores e sem pausas, exceto a instrução addi”. A afirmação é verdadeira ou falsa? **Justifique.**

add \$s0, \$t1, \$t2

Resposta:

addi \$t3, \$s0, 14

Justificativa:

sub \$t5, \$s0, \$t6

add \$t7, \$s0, \$t8

- f) O código abaixo executa numa CPU com predicação, onde p é um registrador de predicado (a sintaxe do MIPS foi usada só para facilitar). Escreva a sequência mínima equivalente de instruções do MIPS.

slt \$p, \$s1, \$zero

(p) addi \$s3, \$s3, 1

L: ...

Definição de critério de avaliação para a Questão 1: Nas questões que solicitam resposta e justificativa, a resposta não será pontuada sem a devida justificativa, nem se esta última estiver incorreta. Nas questões que perguntam se uma afirmação é verdadeira ou falsa, a resposta só pode assumir um desses valores Booleanos (respostas do tipo “Sim” e “Não” serão desconsideradas, pois geram ambiguidade).

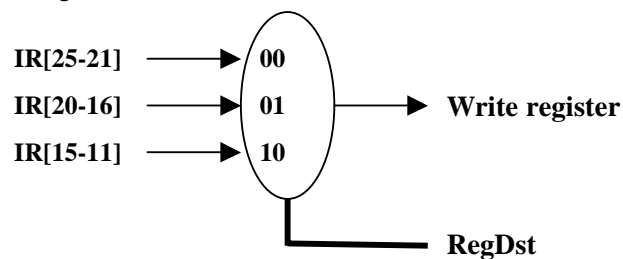
## Parte II – Aplicação de conceitos básicos [4,5 pontos]

2. [valor: 20 x 0,1 = 2,0] Vamos incluir um novo modo de endereçamento para as instruções load e store no MIPS, denominado modo “pós-incrementado”, no qual o registrador-base é atualizado com um novo valor, conforme especificado abaixo:

lw rt, K(rs)++      #  $R[rt] = M[R[rs]+K]$ ,  $R[rs] = R[rs] + K$ ;

sw rt, K(rs)++      #  $M[R[rs]+K] = R[rt]$ ,  $R[rs] = R[rs] + K$ ;

onde R, M e K representam, respectivamente, o banco de registradores, a memória de dados e uma constante inteira. Assuma que o datapath multiciclo do Anexo IV sofra uma única modificação para suportar o novo modo: o multiplexador controlado pelo sinal RegDst é substituído por um novo multiplexador de 3 entradas conectado conforme ilustra a figura abaixo:



Sem alterar o número total de estados, complete as tabelas abaixo fazendo as alterações necessárias na máquina de estados para acomodar o novo modo de endereçamento. (Penalidade: 10 ou mais sinais errados no total anulam a questão).

### a) Estado 2

ALUSrcA	ALUSrcB	lorD	MemRead	MemWrite	MemtoReg	RegDst	RegWrite
		X	X		X	X	

### b) Estado 3

ALUSrcA	ALUSrcB	lorD	MemRead	MemWrite	MemtoReg	RegDst	RegWrite
X	X	X					

### c) Estado 4

ALUSrcA	ALUSrcB	lorD	MemRead	MemWrite	MemtoReg	RegDst	RegWrite
X	X	X	X				

### d) Estado 5

ALUSrcA	ALUSrcB	lorD	MemRead	MemWrite	MemtoReg	RegDst	RegWrite
X	X	X					

Para as Questões 3 e 4, você deve completar as tabelas de ocupação do pipeline do datapath do Anexo V de acordo com as seguintes **regras obrigatórias de preenchimento**:

- Use os acrônimos IF, ID, EX, ME e WB para indicar a ocupação de um estágio por uma dada instrução.
- Uma instrução deve iniciar sua execução o mais cedo possível, mas só deve iniciá-la em um ciclo se, e somente se, a partir daquele ciclo ela puder continuar sua execução sem pausa até terminar (ou seja, se houver necessidade de pausa, ela ocorre antes da ocupação do estágio IF).
- Os ciclos em que uma instrução não ocupa um estágio (porque o já estágio está ocupado ou porque está em pausa) devem ser deixados em branco.
- Os ciclos em que a execução de uma instrução é anulada (por ter sido detectada previsão incorreta de desvio) devem ser marcados com **A**, somente durante os ciclos de relógio em que a instrução anulada ainda estiver no pipeline.

3. [valor:  $4 \times 0,25 = 1,0$ ] Suponha que o datapath do Anexo V seja modificado da seguinte forma:
- Há um caminho de “forwarding” que permite que o valor à saída da memória seja disponibilizado como uma das entradas da ALU no ciclo seguinte.
  - A operação relacional “ $\neq$ ” do desvio condicional é realizada na ALU.
  - O processador faz previsão estática de desvio assumindo sempre a hipótese de desvio não tomado.
- Nota: A instrução bne tem o comportamento tradicional (não é “delayed branch”).

Para o segmento de código abaixo, ilustre a ocupação dos estágios do pipeline, supondo que a previsão resulte incorreta.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw \$s1, 0(\$s0)	IF	ID	EX	ME	WB									
bne \$s1, \$zero, L														
add \$s2, \$s4, \$s3														
j exit														
sub \$s2, \$s4, \$s3														

4. [valor:  $3 \times 0,5 = 1,5$ ] Para o segmento de código abaixo, ilustre a ocupação dos estágios do pipeline do datapath do Anexo V, considerando que ele seja modificado com a introdução de caminhos de forwarding em três cenários distintos.

a) **Cenário 1:** Datapath sem caminho de forwarding algum.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
add \$s1, \$t0, \$t1	IF	ID	EX	ME	WB									
lw \$s0, 0(\$s1)														
sw \$s0, 0(\$s2)														

b) **Cenário 2** - Datapath com um caminho de forwarding: valor da saída da ALU é disponibilizado como uma das entradas da ALU no ciclo seguinte.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
add \$s1, \$t0, \$t1	IF	ID	EX	ME	WB									
lw \$s0, 0(\$s1)														
sw \$s0, 0(\$s2)														

c) **Cenário 3** - Datapath com 2 caminhos de forwarding: além do especificado no Cenário 2, há um outro caminho onde o valor à porta de leitura da memória é disponibilizado à porta de entrada da mesma, no ciclo seguinte.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12	13	14
add \$s1, \$t0, \$t1	IF	ID	EX	ME	WB									
lw \$s0, 0(\$s1)														
sw \$s0, 0(\$s2)														

Critério de avaliação para a Questão 4: um item só será pontuado se todo o seu diagrama estiver correto (não haverá pontuação parcial dentro de um mesmo item).

### Parte III – Generalização a partir de conceitos básicos [2,5 pontos]

5. [valor: 0,25+0,25+0,5 = 1,0] Para caracterizar a duração de cada “hazard”, o manual de um processador reporta as latências entre duas instruções quando ocorre um “hazard”. Dadas duas instruções  $u$  e  $v$ , tais que  $v$  é sucessora imediata de  $u$  e  $v$  consome um valor produzido por  $u$ , a *latência* entre  $u$  e  $v$  é o número de ciclos adicionais que se deve intercalar antes do instante previsto para  $v$  iniciar execução, de forma que, uma vez iniciada,  $v$  possa executar sem pausa até terminar. A tabela abaixo ilustra as combinações possíveis entre uma instrução load e sua sucessora imediata, a qual pode ou não consumir o valor produzido. Para cada caso, mostra-se sua latência e sua frequência de ocorrência em um programa, depois do escalonamento de código.

Caso	u (instrução produtora)	v (instrução sucessora imediata)	latência (u,v)	Ocorrência
1	load	aritmético-lógica consumidora	1	30%
2	load	store consumidora	1	5%
3	load	desvio condicional consumidor	1	5%
4	load	qualquer outro tipo consumidor	0	10%
5	load	qualquer tipo <b>não</b> consumidor	0	50%

Sabendo-se que o número total de instruções executadas é  $I$  e a frequência do relógio é  $f$ , determine o impacto no desempenho devido à latência da instrução load, para o programa com o “mix” de instruções abaixo, supondo que nenhum outro tipo de “hazard” se manifeste durante sua execução.

load	store	Aritmético-lógica	Desvio condicional	Desvio incondicional
25%	10%	52%	11%	2%

a) Qual o número de ciclos de parada resultantes do Caso 1?

$\text{ciclos}_{\text{stall}}(\text{caso1}) =$

b) Qual o número de ciclos de parada resultantes do Caso 2?

$\text{ciclos}_{\text{stall}}(\text{caso2}) =$

c) Qual o tempo de execução do programa?

$\text{tex} =$

Critério de avaliação: neste exercício será pontuada apenas a resposta final de cada item.

6. [3 x 0,5 = 1,5] Dado o código abaixo, otimize-o para os 3 cenários especificados abaixo, supondo que:
- O programa é executado em pipelines de 5 estágios (IF, ID, EX, ME, WB). Cada pipeline resolve o teste de desvios condicionais no estágio ID através de uma unidade TEST nele especialmente alocada.
  - Há caminhos de forwarding permitindo que o valor à saída da ALU (EX) seja disponibilizado, no ciclo de relógio seguinte, como uma das entradas da própria ALU (EX) e da unidade TEST (ID).

**Definição:** Diz-se que duas instruções têm uma *dependência de saída* se não possuem entre si dependência de dados alguma, mas escrevem em um mesmo registrador destino, não podendo por isso ser reordenadas.

a) **Cenário 1** - Corpo do laço desenrolado duas vezes nas seguintes condições:

- Todas as dependências de saída devem ser eliminadas através de renomeamento de registradores;
- O renomeamento deve ser usado somente para eliminar dependências de saída.
- Ao renomear um registrador  $\$ti$ , o novo registrador escolhido deve ser o  $\$ti+4$ ;
- Deve-se eliminar instruções de controle do laço duplicadas, preservando a semântica do código original, através de ajustes de constante imediata e compensação de “offset”.
- Qualquer outra otimização não citada acima é proibida.

b) **Cenário 2** – Código desenrolado como no Cenário 1 e escalonado para eliminar todos os ciclos de pausa em um datapath com emissão de uma única instrução por ciclo.

c) **Cenário 3** – Código desenrolado como no Cenário 1 e escalonado para eliminar todos os ciclos de pausa em um datapath com emissão de duas instruções por ciclo, sendo que a primeira deve ser do tipo R ou desvio e a segunda deve ser de acesso à memória

