

Nome: _____ Matrícula: _____

Instruções critério de avaliação: A interpretação das questões é parte integrante desta avaliação. As respostas devem ser fornecidas no espaço para elas reservado. Nas questões que solicitam justificativa, a resposta não será pontuada sem a devida justificativa, nem se esta última estiver incorreta. A pontuação de cada item (“a”, “b”, etc.) de uma questão é indivisível: um item só será pontuado se estiver totalmente correto.

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

1. [valor: $4 \times 0,5 + 1,0 = 3,0$] Responda sucintamente as perguntas abaixo:

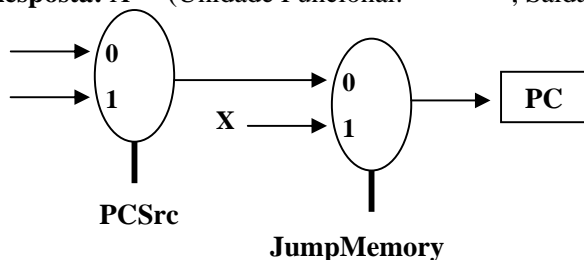
- a) O registrador \$t0 armazena o endereço 0x2000 0000. Quais os endereços mínimo e máximo referenciáveis pela instrução `lb $s0, x`, onde `x` é o endereço efetivo obtido usando \$t0 como registrador-base?
 $X_{\min} = 0x$ $X_{\max} = 0x$
- b) Num arquivo-objeto, encontram-se as representações em linguagem de máquina de instruções `beq`, `jr`, `jal` e `j`. Sabe-se que `beq` e `j` usam rótulos (*label*) locais. Afirmção: “Duas dessas instruções nunca precisam ser linkeditadas.” A afirmação é verdadeira ou falsa? **Justifique a (não-)necessidade de linkedição das instruções. Resposta: (V ou F):** . **Justificativa:**

- c) Afirmção: “Caso o sistema não inicialize \$gp para apontar para o centro da área de dados estáticos, a pseudo-instrução `lw $s0, 0x8000 8020` requer duas instruções nativas para sua implementação”. A afirmação é verdadeira ou falsa? **Justifique apresentando o código que a implementa (o qual deve garantir que os valores de quaisquer registradores não-reservados sejam preservados após sua execução).**

Resposta (V ou F): . **Justificativa:**

- d) Seja a instrução “jump memory”: `jm K(rs) # PC ← M[R[rs]+sign-extend(K)]`, onde M e R representam, respectivamente, a memória de dados e o banco de registradores. Para incluí-la no datapath do Anexo III, um novo multiplexador e um novo sinal de controle foram inseridos sem se modificar as demais conexões pré-existentes (diagrama abaixo). Identifique o nome da unidade funcional e o nome da respectiva saída a ser conectada a X, **identificado-as com os rótulos constantes do Anexo III.**

Resposta: X = (Unidade Funcional: ; Saída:)



- e) No código abaixo `beq` é “delayed branch”. Otimize o código para eliminar o `nop`.

Código original	
loop:	<code>lw \$t0, 100 (\$s0)</code>
	<code>addi \$s0, \$s0, 4</code>
	<code>beq \$s0, \$t2, loop</code>
	<code>nop</code>

Código otimizado	
loop:	<code>beq \$s0, \$t2, loop</code>

- f) Uma máquina possui um pipeline que emite uma instrução a cada ciclo, mas admite escalonamento dinâmico, graças a múltiplas unidades inteiras: MU, AR1 e AR2. A multiplicação é realizada em MU; somas e subtrações em AR1 ou AR2. Sabe-se que: para uma instrução de multiplicação, o produto fica disponível

à saída de MU ao final do décimo ciclo de execução da instrução; para as demais instruções, o resultado fica disponível à saída de AR1 ou AR2 ao final do terceiro ciclo de execução. Há atalhos permitindo que o valor à saída de cada unidade fique disponível, à entrada das demais unidades, no ciclo de relógio seguinte. Dado o código abaixo e os valores armazenados nos registradores \$t1 = 3, \$t2 = 4, \$t4 = 7, \$s1 = 5, \$s4 = 7 e \$s5=6, determine o seguinte:

mul \$t0, \$t1, \$t2 **A sequência de valores (resultados) produzidos às saídas unidades funcionais:**

add \$s2, \$s1, \$t0 (, , ,)

sub \$s3, \$s4, \$s5 **A sequência de valores (resultados) escritos no banco de registradores:**

addi \$t3, \$t4, 4 (, , ,)

Parte II – Aplicação de conceitos básicos [4,0 pontos]

Nas próximas questões envolvendo pipeline, o datapath básico do Anexo V pode ser estendido com diferentes combinações de recursos adicionais. Para simplificar o enunciado das próximas questões, as extensões estão numeradas e listadas abaixo:

- **Extensão 1a:** Para resolver o hazard estrutural resultante do acesso simultâneo ao banco de registradores nos estágios ID e WB, a escrita de um registrador ocorre no primeiro semi-ciclo de relógio; a leitura, no segundo semi-ciclo.
- **Extensão 1b:** Para resolver o hazard estrutural resultante do acesso simultâneo ao banco de registradores nos estágios ID e WB, a leitura de um registrador ocorre no primeiro semi-ciclo de relógio; a escrita, no segundo semi-ciclo.
- **Extensão 2:** As operações relacionais “≠” (ne) e “=0” (eqz) de um desvio condicional são realizadas em uma unidade funcional dedicada, denominada TEST, que faz parte do estágio ID.
- **Extensão 3:** Há um atalho permitindo que o valor à saída da ALU (EX) seja disponibilizado, no ciclo de relógio seguinte, como uma das entradas da unidade TEST (ID).
- **Extensão 4:** Há um atalho permitindo que o valor à porta de leitura da memória de dados (MEM) seja disponibilizado, no ciclo de relógio seguinte, à porta de escrita da própria memória de dados (MEM).
- **Extensão 5:** Há um atalho permitindo que o valor à saída da ALU (EX) seja disponibilizado, no ciclo de relógio seguinte, como uma das entradas da própria ALU (EX).
- **Extensão 6:** Há um atalho permitindo que o valor à saída da ALU (EX) seja disponibilizado, no segundo ciclo de relógio seguinte, como uma das entradas da própria ALU (EX).
- **Extensão 7:** Há um atalho permitindo que o valor à porta de leitura da memória de dados (MEM) seja disponibilizado, no ciclo de relógio seguinte, como uma das entradas da ALU (EX).

Nas questões envolvendo pipeline, 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 estágio já está ocupado ou porque está em pausa) devem ser deixados em branco.
- Se uma instrução for anulada a partir de um determinado ciclo (por não ter sido satisfeita a hipótese de previsão de desvio), os ciclos em que a instrução resulta anulada devem ser preenchidos com o acrônimo A, para indicar o efeito da anulação.

2. [valor: 0,5+0,5] Para o código abaixo, ilustre a ocupação dos estágios do pipeline, nos seguintes cenários:

- a) Cenário 1: Ao datapath do Anexo V foram acrescentadas só as Extensões 1b, 5 e 6.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12
add \$s0, \$t0, \$t1	IF	ID	EX	ME	WB							
sub \$s1, \$s0, \$t3												
add \$t4, \$s0, \$t5												
add \$s2, \$s0, \$s4												
add \$s3, \$s1, \$s5												

- b) Cenário 2: Ao datapath do Anexo V foram acrescentadas só as Extensões 1a, 5 e 6.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12
add \$s0, \$t0, \$t1	IF	ID	EX	ME	WB							
sub \$s1, \$s0, \$t3												
add \$t4, \$s0, \$t5												
add \$s2, \$s0, \$s4												
add \$s3, \$s1, \$s5												

3. [valor: 0,5] Para o código abaixo, ilustre a ocupação dos estágios do pipeline, sob as seguintes hipóteses:

- Ao datapath do Anexo V foram acrescentadas só as Extensões 1a, 2, 3, 7.
- A instrução bne tem o comportamento tradicional (não é “delayed branch”).

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12
Loop: lw \$t0, 0(\$s1)	IF	ID	EX	ME	WB							
addu \$t0, \$t0, \$s2												
sw \$t0, 0(\$s1)												
addi \$s1, \$s1, -4												
bne \$s1, \$zero, Loop												

4. [valor: 0,5+0,5] Para o código abaixo, ilustre a ocupação dos estágios do pipeline, nos seguintes cenários, onde o desvio não é “delayed branch” e a hipótese de previsão é de desvio não-tomado.

```

add $s6, $zero, $s0
Loop: sub $s4, $s1, $s6
      add $s1, $s2, $s3
      beqz $s4, loop
      add $t1, $t4, $t5
      lw $t0, 0($t1)
      add $s4, $s1, $t0

```

- a) **Cenário 1:** Ao datapath do Anexo V foram acrescentadas só as Extensões 1a, 2 e 5. Sabe-se que o corpo do laço será executado mais do que uma vez. **Atenção:** nas 3 últimas linhas, você deve completar as instruções faltantes e a respectiva ocupação de estágios.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12
add \$s6, \$zero, \$s0	IF	ID	EX	ME	WB							
sub \$s4, \$s5, \$s6												
add \$s1, \$s2, \$s3												
beqz \$s4, loop												

- b) **Cenário 2:** Ao datapath do Anexo V foram acrescentadas só as Extensões 1a, 2 e 5. Sabe-se que o corpo do laço será executado pela última vez. **Atenção:** nas 3 últimas linhas, você deve completar as instruções faltantes e a respectiva ocupação de estágios.

Instrução/Ciclo	1	2	3	4	5	6	7	8	9	10	11	12
sub \$s4, \$s5, \$s6	IF	ID	EX	ME	WB							
add \$s1, \$s2, \$s3												
beqz \$s4, loop												

5. [valor: 0,5+0,5+0,5] Um segmento de código consiste de 2 desvios condicionais. Abaixo a sequência de resultados observados para cada desvio executado para uma dada invocação daquele segmento, onde T e N significam, respectivamente, tomado e não-tomado:

- Desvio 1: (T, N, T, N, T, N)
- Desvio 2: (T, T, N, T, N, N, T)

Supondo que cada desvio tenha seu próprio “buffer” (exclusivo) de previsão, mostre a sequência de previsões realizadas em cada um dos cenários abaixo.

a) Cenário A: Previsor de 1 bit, inicializado como abaixo.

Resposta: Desvio 1: (T, , , , ,) Desvio 2: (T, , , , ,)

b) Cenário B: Previsor de 2 bits, inicializado como abaixo.

Resposta: Desvio 1: (T, T, T, , ,) Desvio 2: (T, T, T, T, , ,)

c) Supondo que o código é executado milhares de vezes e que cada “buffer” é inicializado da mesma forma para cada invocação do código, qual a percentagem de previsões corretas nos Cenários 1 e 2 (indicar o valor com precisão de uma casa depois da vírgula)?

Resposta: Cenário A: %; Cenário B: %

Parte III – Generalização a partir de conceitos básicos [3,0 pontos]

6. [valor: 1,0] É muito comum testar-se a desigualdade $0 \leq i < L$ (onde i é o índice e L é o número de elementos de um arranjo) e desviar-se para um endereço `IndexOutOfBounds` em caso contrário. **Afirmção:** “São necessárias no mínimo três instruções do MIPS para realizar o teste e o desvio”. A afirmação é verdadeira ou falsa? **Supondo a alocação $(i, L) \rightarrow (\$a1, \$t2)$, justifique completando o código necessário.**

Resposta: (V ou F) **Justificativa:**

beq \$t0, \$zero, IndexOutOfBounds

7. [valor: 0,5+0,5] O MIPS tem dois tipos de desvio condicional: com atraso regular ou *delayed branch* (ex., `bne`) e com atraso-e-anulação, denominado de *branch-likely* (ex., `bnel`). A instrução *branch-likely* anula a instrução no *slot* quando o desvio **não** é tomado (impedindo que a instrução no *slot* escreva em memória ou registrador nos estágios ME e WB). Otimize o código abaixo, eliminando a instrução `nop` no *slot*, em dois cenários: a) *delayed-branch* regular (Cenário 1); b) *branch-likely*. **Restrição: Você pode incluir e mover instruções para manter a semântica original, mas nenhuma instrução já impressa pode ser alterada.**

Sem otimização	a) Cenário 1	b) Cenário 2
addi \$s1, \$zero, -1	addi \$s1, \$zero, -1	addi \$s1, \$zero, -1
loop: addi \$s1, \$s1, 1	loop: lb \$s0, 0(\$s1)	loop: lb \$s0, 0(\$s1)
lb \$s0, 0(\$s1)	slti \$t0, \$s1, 100	slti \$t0, \$s1, 100
slti \$t0, \$s1, 100	addi \$s0, \$s0, 1	addi \$s0, \$s0, 1
addi \$s0, \$s0, 1	sw \$s0, 0(\$s1)	sw \$s0, 0(\$s1)
sw \$s0, 0(\$s1)	bne \$t0, \$zero, loop	bnel \$t0, \$zero, loop
bne \$t0, \$zero, loop		
nop		
add \$s2, \$s1, \$s3	add \$s2, \$s1, \$s3	add \$s2, \$s1, \$s3

8. [valor: 0,5+0,5] No código abaixo, sabe-se que nenhuma instrução a partir do endereço `exit` usa o valor no registrador `$t2`. Os símbolos **rx** e **ry** designam registradores arbitrários.

beqz \$s1, else

nop

then: sub rx, \$s5, \$s6

add \$t1, ry, \$t3

lw \$t0, 0(\$t1)

j exit

else: add \$t2, \$t4, \$t5

lw \$t0, 0(\$t2)

exit: add \$s1, \$s1, \$t0

...

a) **Afirmção:** “Lembrando que `beqz` não é *branch-likely*, a instrução `add` residente no endereço `else` pode ser movida para o *slot* do desvio se e somente se $ry \neq t2$.” A afirmação é verdadeira ou falsa? Justifique.

Resposta: (V ou F). **Justificativa:**

b) **Afirmção:** “Substituindo `bnez` por um *branch-likely*, digamos, `bnezl`, a instrução `add` residente em `else` sempre pode ser movida para o *slot* do desvio.” A afirmação é verdadeira ou falsa? **Justifique.**

Resposta: (V ou F). **Justificativa:**