

## Lista de Exercício 2 – OC II

### Exercício 1

Considere as seguintes características para erro de previsão. Os desvios são 20% do processamento. O CPI sem parada é igual a 1,5.

	Previsão Não-Tomado	Previsão Tomado
Desvio Tomado	4	1
Desvio Não-Tomado	0	5

(a) Considerando que os desvios são tomados 75% do tempo. Determine o CPI desta máquina usando um sistema de previsão estático tomado. Faça o mesmo para o não-tomado.

(b) Considere o uso de uma branch history table de 1 bit. Assuma que o predictor é inicializado no estado T. Dada a sequência de desvios TTTTNNNTNTNTTTTNTNTNT, qual é a taxa de acerto e CPI correspondente?

(c) Repita o item acima para o caso de uma branch history table de 2 bits. Assuma que o predictor é inicializado no estado PT. Os estados são:

PT – Pouco Tomado  
 MT – Muito Tomado  
 PN – Pouco Não-Tomado  
 MN – Muito Não-Tomado

### Exercício 2

Para este problema, assuma um processador VLIW com três unidades de inteiros (X, Y, Z), uma unidade de multiplicação (M) e duas unidades de *load* e *store* (LS0, LS1). Instruções de ALU tem uma latência de 1, multiplicação tem uma latência de 5 e *loads* tem uma latência de 2. Um desvio (*branch*) pode ser executado por ciclo e executa no pipeline Z. O seguinte código foi gerado assumindo uma técnica de escalonamento conhecida como EQ.

	X	Y	Z	M	LS0	LS1
1	ADDI R9, R0, 9	ADDI R10, R0, 10				
2	ADDI R6, R0, 6	ADDI R8, R0, 8	ADDI R5, R0, 5			
3					LW R6, 0(R7)	LW R8, 4(R7)
4	ADDI R12, R6, 1	ADDI R13 R8, 2				
5				MUL R7, R6, R9		
6				MUL R5, R8, R10		
7					LW R14, 8(R7)	
8	ADD R15, R16, R17					
9	ADD R14, R14, R5					
10	SUB R19, R18, R22					
11	ADD R5, R7, R5					

**R0 tem valor zero!**

**Assuma que os endereços 0(R7), 4(R7) e 8(R7) contêm os valores 0, 10 e 1, respectivamente.**

(a) Quais os valores de R12, R13 e R14 após a execução do código?

(b) Sem alterar os nomes dos registradores, reescale o código acima de forma a melhorar o desempenho sem impactar na lógica correta.

	X	Y	Z	M	LS0	LS1
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						

### Exercício 3

Um fabricante acabou de lançar um processador que possui uma cache L1 de 128KB, associativa de 2 vias (2 way) com um tempo de acesso de 5 ciclos. O tamanho do bloco é de 64 bytes. Já o acesso à memória principal demora 150 ciclos.

Um programador escreve um programa que repetidamente acessa (em um loop) somente os seguintes dois blocos A e B. O loop é executado por bilhões de iterações.

(a) Qual o tempo médio de acesso a memória esperado? Por quê?

(b) Ao realizar testes, o programador mediu um tempo médio de acesso a memória de 7,5 ciclos. O que você pode dizer sobre esse processador? Descreva tudo que tenha a dizer concretamente, mas de forma concisa. Mostre como chegou a esta conclusão.

(c) A cache de vítimas ajudaria neste caso? Por quê?

### Exercício 4

Este problema investigará os efeitos da **Previsão de Desvios (Branch Prediction)**. Ao longo do problema vamos considerar o seguinte código:

loop:

```
LW    R4, 0(R3)
ADDI  R3, R3, 4
SUBI  R1, R1, 1
```

b1:

```
BEQZ  R4, b2
ADDI  R2, R2, 1
```

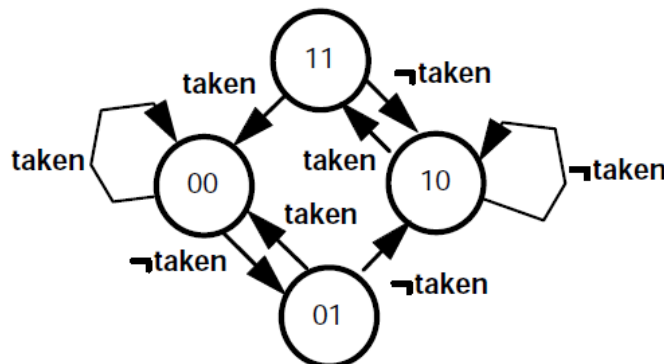
b2:

```
BNEZ  R1, loop
```

Assuma que o valor inicial de R1 é n ( $n > 0$ )

Assuma que o valor inicial de R2 é 0 (R2 guarda o resultado do programa)  
 Assuma que o valor inicial de R1 é p (um ponteiro para o início de um array de inteiros de 32 bits)

Todas as previsões de desvios neste problema vai ser baseada em um predictor de 2 bits , como mostrado abaixo:



O estado **1X** é considerado como **não-tomado**. Já o estado **0X** é considerado como **tomado**.  
 Assumam que b1 e b2 não possuem conflito no BHT.

(a) O que o programa calcula? Ou seja, qual o valor contido em R2 quando o loop terminar?

(b) Vamos investigar o funcionamento do predictor proposto. Assuma que as entradas do programa são  $n = 8$  e  $p[0] = 1, p[1] = 0, p[2] = 1, p[3] = 0, \dots$  etc.; isto é, o array exibe elementos alternados de 1s e 0s. Preencha o restante da tabela abaixo. Qual é o número de previsões incorretas?

A tabela contém uma entrada para cada desvio (b1 e b2) que é executado. Os bits do Predictor de Desvio (PD) na tabela são bits da BHT (Branch History Table). Para cada desvio, verifique os bits de PD correspondente para realizar a previsão. Em seguida, atualize os bits de PD da entrada.

Estado do Sistema			Predictor de Desvio (PD)		Comportamento do Desvio	
PC	R3	R4	b1 bits	b2 bits	Previsto	Ocorrido
b1	4	1	10	10	N	N
b2	4	1	10	10	N	T
b1						
b2						
b1						
b2						
b1						
b2						
b1						
b2						
b1						
b2						
b1						
b2						
b1						
b2						

(c) Agora adicionamos um **bit de histórico global (desvios correlacionados)**. Preencha a tabela

abaixo e novamente informe o número total de predições incorretas. Considere as mesmas condições do item b.

O bit histórico igual a 0 corresponde a não tomado e 1 a tomado.

Estado do Sistema				Previsor de Desvio (PD)				Comportamento do Desvio	
PC	R3	R4	Bit História	b1 bits		b2 bits		Previsto	Ocorrido
				0	1	0	1		
b1	4	1	1	10	10	10	10	N	N
b2	4	1	0	10	10	10	10	N	T
b1									
b2									
b1									
b2									
b1									
b2									
b1									
b2									
b1									
b2									
b1									
b2									
b1									
b2									

(d) Refaça o item c, mas agora considere um segundo bit de histórico global. Qual o número de predições incorretas?

Estado do Sistema				Previsor de Desvio (PD)								Comportamento do Desvio	
PC	R3	R4	Bit História	b1 bits				b2 bits				Previsto	Ocorrido
				0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1		
b1	4	1	1 1	10	10	10	10	10	10	10	10	N	N
b2	4	1	0 1	10	10	10	10	10	10	10	10	N	T
b1													
b2													
b1													
b2													
b1													
b2													
b1													
b2													
b1													
b2													
b1													
b2													

(e) Compare os resultados do itens b, c e d. Onde ocorre a maior parte das previsões incorretas em cada caso (no início, periodicamente, no fim, etc)? O que isto diz sobre a previsão com bits de histórico global? O que ocorreria para um valor de n maior? Explique resumidamente.

(f) As entradas utilizadas neste problema são muito regulares. O que você esperaria que ocorresse caso as entradas fossem aleatórias (igual probabilidade do elemento do array ser 0 ou 1)? Dos 3 previsores que vimos neste problema, qual deles seria o melhor para este tipo de entradas? O resultado é o mesmo para n pequeno e grande?