

## Lista de Exercício 3 – OC II

### Exercício 1

(a) Em um multiprocessador com uma cache privada L1 e cache compartilhada L2, se dados compartilhados (shared) são lidos por todos os cores todos os ciclos, mas escritos apenas uma vez em 1000 ciclos por um core, qual tipo de protocolo de coerência devo usar? Update ou Invalidate? Por quê?

(b) Considere uma máquina multiprocessada com 3-cores. Assuma que cada processador possui uma cache com associatividade completa com 4 blocos de 2 palavras. Na tabela abaixo, considere o estado de cada bloco, na respectiva linha, como C (compartilhado), I (Invalido) e E (Exclusivo). Quando apenas uma cache possui o bloco, considere como C. Indique também quem fornece o dado e se o compartilhamento é verdadeiro ou falso, quando for o caso. Considere todas as caches vazias inicialmente e o protocolo Snoopy.

Obs.: considere que os blocos sempre apresentam palavras em pares. p.ex: w0 e w1; w2 e w3; w4 e w5 ....

Ação do Processador	C-P1	C-P2	C-P3	Ação do Barramento	Fornecedor do dado	Compartilhamento
P1 lê w3						
P1 lê w5						
P2 lê w3						
P3 lê w4						
P3 escreve w1						
P1 escreve w3						
P3 lê w4						
P2 lê w3						
P2 escreve w5						
P2 escreve w2						
P1 lê w4						

### Exercício 2

(a) Você precisa executar o código abaixo na arquitetura “VMIPS adaptada”. Assuma o comprimento de vetor máximo igual a 128. Mostre o diagrama de pipeline deste código sendo executando em uma arquitetura de uma raia que tem uma unidade independente de load, uma de store, uma de multiplicação e uma de ALU. Loads demoram 3 ciclos (L0, L1, L2), store 2 ciclos (S0, S1), multiplicação 5 (Y0, Y1, Y2, Y3, Y4) e operações da ALU 2 ciclos (X0, X1). Assuma que há pipeline total nas unidades funcionais. O pipeline tem estágios dedicados de leitura e escrita de registradores, mas só há uma porta de leitura e uma de escrita no registrador de arquivo. Não há encaminhamento.

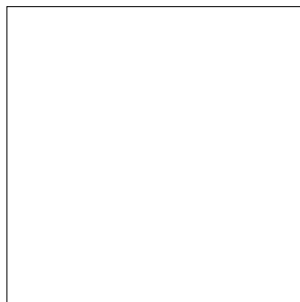
Código “VMIPS adaptada”

LDI	VLR, 4
LV	V1, R4
MULVS.DI	V3, V1, 7
LV	V2, R5
ADD	V4, V3, V2
SV	R6, V4

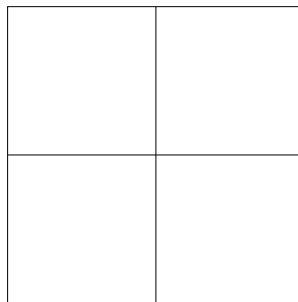
(b) Repita do diagrama do pipeline acima considerando que o pipeline tem agora duas portas de escrita e duas de leitura no registrador de arquivo por raia. Assuma também que a arquitetura possui duas raias (duas fontes duplicadas de unidades funcionais).

### Exercício 3

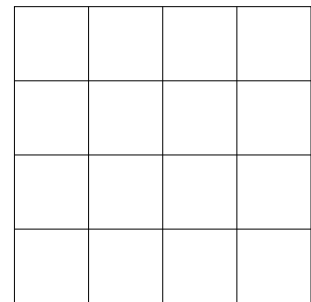
Considere os seguintes três processadores (X, Y e Z) que são fabricados em uma área constante de silício de 16A. Assuma que uma thread simples em um core tem um desempenho que cresce com a raiz quadrada da sua área.



Processador X  
1 core grande de 16A



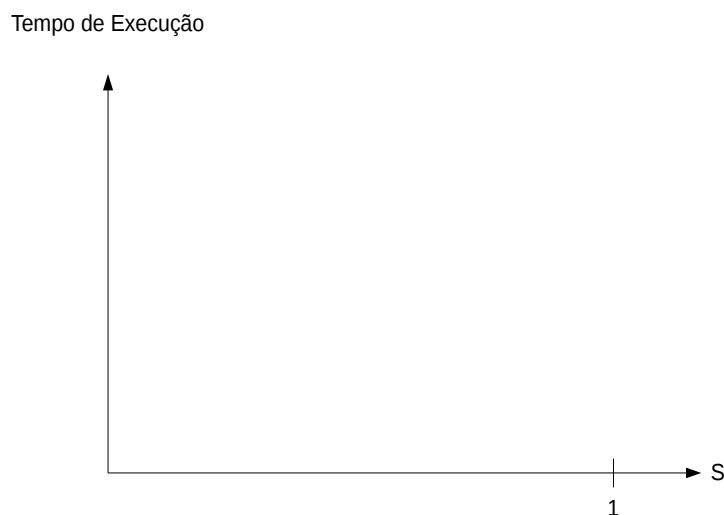
Processador Y  
4 cores médios de 4A



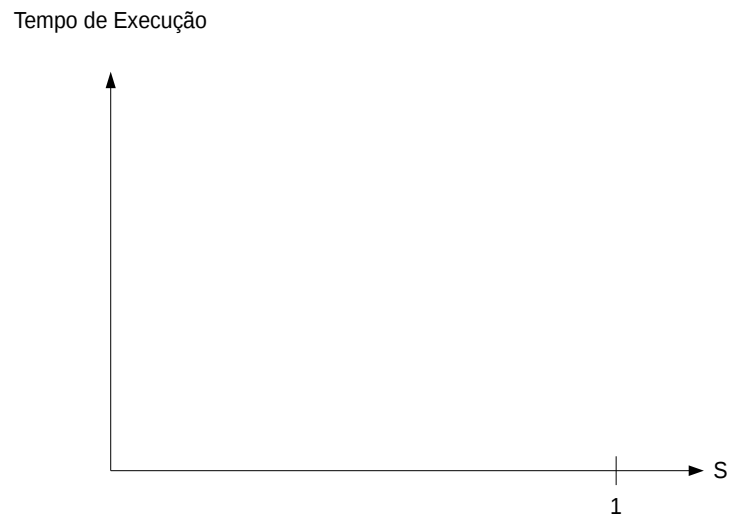
Processador Z  
16 cores pequenos de 1A

Em cada um dos três processadores, nós vamos executar um programa onde **a fração S do trabalho é serial e a fração 1-S do trabalho é infinitamente paralelizável**. Em função de S, plote o tempo de execução do programa em cada um dos três processadores. Assuma que o programa leva o tempo T para ser executado usando apenas um pequeno core do processador Z. Mostre a reta de tempo, indicando o valor onde o eixo y é interceptado e a inclinação.

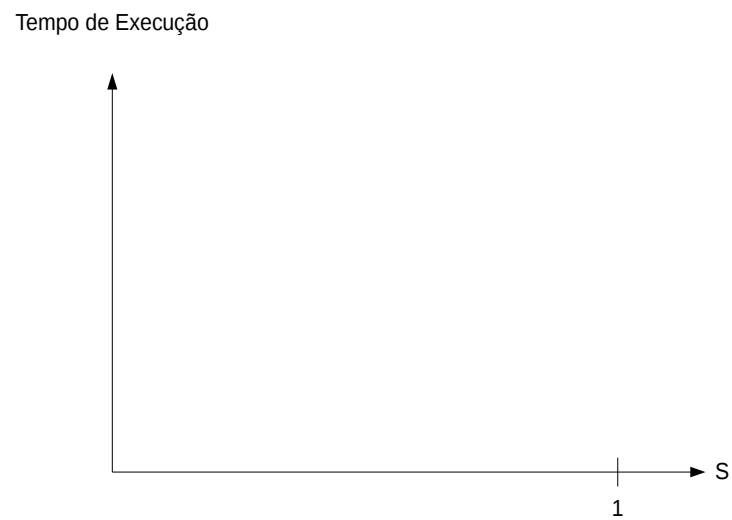
(a) Processador X



(b) Processador Y



(c) Processador Z



(d) Tipicamente, para um programa realístico, a fração paralela não é infinitamente paralelizável. Cite três razões para isso.

#### Exercício 4

Considere uma máquina multiprocessada com 4-cores, usando o protocolo *write-invalidate*. Mostre para cada bloco da cache e cada cache qual o estado em cada ciclo considerando a execução do código intervalado abaixo:

Tempo	P1	P2	P3	P4
1	lw R6, 128(R0)			
2			sw R0, 264(R0)	
3				lw R7, 132(R0)
4				sw R10, 128(R0)
5		sw R0, 384(R0)		
6		lw R11, 144(R0)		
7		lw R12, 388(R0)		
8			lw R9, 128(R0)	
9			lw R10, 390(R0)	
10				lw R15, 312(R0)
11				sw R7, 316(R0)
12	sw R6, 384(R0)			
13	lw R9, 228(R0)			
14			sw R10, 200(R0)	

Considere: C (compartilhado), I (Invalido) e E (Exclusivo).

Assuma que cada processador possui uma cache de mapeamento direto de 16KB de tamanho e com blocos de 128-bytes.

Tempo	Bloco	C-P1	C- P2	C- P3	C-P4
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					