

Universidade Federal de Santa Catarina  
Centro Tecnológico – Departamento de Informática e de Estatística  
INE 5607 – Organização e Arquitetura de Computadores  
Prof. Laércio Lima Pilla

Questões da área de Unidades Centrais de Processamento.

1) Sua empresa desenvolve aplicações de classificação baseada em imagens. A rede de supermercados Super Juca resolveu contratá-los para preparar um programa que classifica lotes de maçãs baseado na quantidade de maçãs podres pegadas na imagem.

O procedimento principal do programa Apple Classifier™ contém 10 bilhões de instruções, sendo 50% aritméticas, 20% loads, 10% stores e 20% branches. Tendo essa aplicação em mente, lhe é dada a responsabilidade de escolher o processador mais adequado. Quatro opções de processadores MIPS são apresentadas:

- a) Monociclops 100: processador monociclo, 100 MHz, R\$100,00
- b) Multiperf V: processador multiciclo, 800 MHz, R\$150,00
- c) Propipe FX: processador pipeline 8 estágios, 10% de bolhas, 1,5 GHz, R\$200,00
- d) Supercalc 2: processador superescalar 8 estágios, sem bolhas, despacho de 2 instruções em ordem (40% de sucesso com a aplicação), 1,5 GHz, R\$300,00

Apresente: (i) o desempenho previsto com cada processador; (ii) o speedup do processador de maior desempenho sobre os outros; e (iii) o processador com melhor custo/desempenho.

Resposta:

**(i) Desempenho previsto com cada processador:**

**Tempo(Monociclo)** = Instruções / frequência

$$= 10 \cdot 10^9 / 100 \cdot 10^6$$

$$= 10^{10} / 10^8$$

$$= 10^2$$

$$= 100 \text{ s}$$

**Tempo(Multiciclo)** = CPI médio \* Instruções / frequência

$$= (0,5 \cdot 4 + 0,2 \cdot 5 + 0,1 \cdot 4 + 0,2 \cdot 3) \cdot 10 \cdot 10^9 / 800 \cdot 10^6$$

$$= (2 + 1 + 0,4 + 0,6) \cdot 100 \cdot 10^8 / 8 \cdot 10^8$$

$$= 4 \cdot 100 / 8$$

$$= 400 / 8$$

$$= 50 \text{ s}$$

**Tempo(Pipeline)** = ((Estágios do pipeline - 1) + Instruções + (Instruções \* %bolhas)) / frequência

$$= (8 - 1 + 10 \cdot 10^9 + 10 \cdot 10^9 \cdot 0,1) / 1,5 \cdot 10^9$$

$$= (7 + 1,1 \cdot 10^9) / 1,5 \cdot 10^9$$

$$(\text{sem o } 7 \text{ que é } \lll 10^9) \approx 1,1 \cdot 10^9 / 1,5 \cdot 10^9$$

$$= 11/1,5$$

$$= 7,33 \text{ s}$$

**Tempo(Superescalar)** = ((Estágios do pipeline -1) + Instruções\*(1-%despacho\_múltiplo + %despacho\_múltiplo/paralelismo))/frequência

$$= (7 + 10 \cdot 10^9 \cdot (1 - 0,4 + 0,4/2)) / 1,5 \cdot 10^9$$

$$= (7 + 10 \cdot 10^9 \cdot (0,6 + 0,2)) / 1,5 \cdot 10^9$$

$$= (7 + 10 \cdot 10^9 \cdot 0,8) / 1,5 \cdot 10^9$$

$$(\text{sem o } 7 \text{ que é } \lll 10^9) \approx 0,8 \cdot 10^9 / 1,5 \cdot 10^9$$

$$= 8/1,5$$

$$= 5,33 \text{ s}$$

Resumindo,

Tempo(Monociclo) = 100 s

Tempo(Multiciclo) = 50 s

Tempo(Pipeline) = 7,33 s

Tempo(Superescalar) = 5,33 s

**(ii) Speedup do processador de maior desempenho sobre os outros**

Speedup superescalar sobre monociclo =  $100 / 5,33 = 18,76$

Speedup superescalar sobre multiciclo =  $50 / 5,33 = 9,38$

Speedup superescalar sobre pipeline =  $7,33 / 5,33 = 1,375$

**(iii) Processador com melhor custo/desempenho**

Custo/desempenho = Custo/(1/tempo)

Custo/desempenho monociclo =  $R\$100,00 / (1/100) = 10000$

Custo/desempenho multiciclo =  $R\$150,00 / (1/50) = 7500$

Custo/desempenho pipeline =  $R\$200,00 / (1/7,33) = 1466$

Custo/desempenho superescalar =  $R\$300,00 / (1/5,33) = 1599$

**Melhor custo/desempenho: Processador pipeline Propipe FX**

2) Considere o seguinte trecho de código em linguagem de montagem:

```

1. add $t0, $s0, $s1
2. add $t1, $s0, $s2
3. or  $t2, $t0, $t1
4. nor $t3, $s2, $zero
5. and $t4, $s1, $t3
6. or  $t5, $t2, $t4
7. sll $t6, $t1, 4
8. add $t7, $t6, $s0
9. nor $t8, $t5, $t4

```

Sabendo que esse código executará em um processador pipeline com o despacho de até 3 instruções para ULAs fora de ordem, apresente um escalonamento das instruções levando o mínimo de ciclos respeitando suas dependências.

Resposta:

Para definir quais instruções podem executar antes das outras, a melhor opção é montar um grafo de dependência entre as instruções. Uma instrução depende de outras se algum de seus registradores fonte é o registrador de destino de outra instrução.

Exemplo: add \$t1, \$s0, \$s0 depende de add \$s0, \$s1, \$s2.

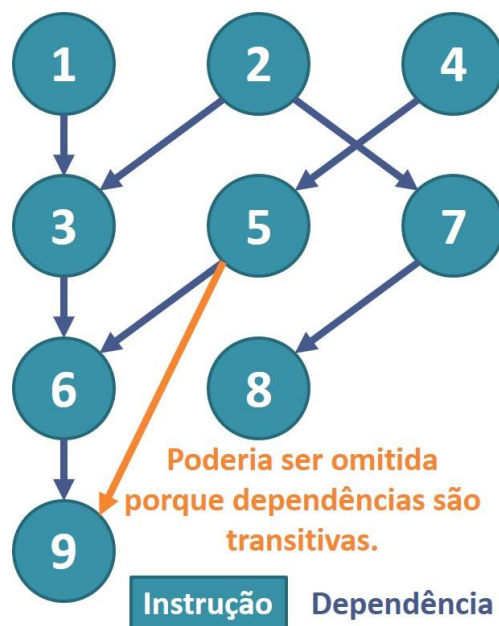


Figura 1. Grafo de dependência entre instruções segundo sua numeração. O formato X->Y indica que Y só pode executar após X.

	ULA 1	ULA 2	ULA 3
<b>Ciclo 1</b>	1. add \$t0, \$s0, \$s1	2. add \$t1, \$s0, \$s2	4. nor \$t3, \$s2, \$zero
<b>Ciclo 2</b>	3. or \$t2, \$t0, \$t1	5. and \$t4, \$s1, \$t3	7. sll \$t6, \$t1, 4
<b>Ciclo 3</b>	6. or \$t5, \$t2, \$t4	8. add \$t7, \$t6, \$s0	
<b>Ciclo 4</b>	9. nor \$t8, \$t5, \$t4		

Figura 2. Escalonamento das instruções. Instruções em uma mesma linha poderiam se encontrar em colunas diferentes.

3) Uma nova versão da principal aplicação de sua empresa, TamagoshiSim, está prevista para o fim do ano. Entre os melhoramentos feitos no código estão a paralelização de 95% do código, enquanto na parte de simulação houve um aumento de 10x na complexidade (o que exige 10x mais tempo para computar).

Sabendo de seus conhecimentos na área de arquitetura e organização de computadores, a sua chefe lhe nomeou para definir o hardware base para rodar a aplicação. Tendo em vista que se deseja manter o mesmo tempo de simulação da aplicação anterior, qual o número mínimo de núcleos da nova máquina?

Resposta:

A questão trata de um simples caso de uso do cálculo de *speedup* pela Lei de Amdahl:

$$\text{Speedup}(n) = 1/(B+(1/n)*(1-B))$$

No nosso caso, nós sabemos que o *speedup* a ser alcançando com paralelismo tem que ser igual a dez. Isso acontece porque a nova versão do código é 10x mais complexa e, logo, precisa rodar em 1/10 do tempo da versão original, o que significa que um *speedup* igual a 10 é necessário. Tendo isso em vista, nós podemos reorganizar a fórmula para evidenciar n.

$$\text{Speedup}(n) = 1/(B+(1/n)*(1-B))$$

$$B+(1/n)*(1-B) = 1/\text{Speedup}(n)$$

Nós sabemos também o percentual de código sequencial B (5%). Substituindo os valores na equação, temos

$$B+(1/n)*(1-B) = 1/\text{Speedup}(n)$$

$$0,05+(1/n)*0,95 = 1/10$$

$$0,05+0,95/n = 0,1$$

$$0,95/n = 0,05$$

$$0,95/0,05 = n = 19 \text{ núcleos}$$

Sendo assim, a empresa precisa de uma máquina com um mínimo de 19 núcleos.