

## Lista de Exercícios

### PIPELINING

1. Qual a **vantagem** que a utilização da técnica de pipelining traz em relação a arquitetura MIPS-Multiciclo estudada em aula? Descreva textualmente.
2. Embora o desempenho seja superior, há um custo associado a implementação utilizando pipelining. **Qual é este custo? Em que ele implica?**
3. **O que é CPI?** Forneça uma explicação textual e a forma de cálculo. O CPI **muda se ele implementa ou não a técnica de pipelining?**
4. Quais são as **fórmulas** para se calcular desempenho para o processador multiciclo e com pipelining?
5. **Porque a técnica de pipelining não pode ser facilmente explorada em arquiteturas CISC?** Quais são os fatores complicadores?
6. **Como computadores modernos como os da família x86 implementam a técnica de pipelining? Todas as instruções são pipelinizáveis?** Justifique.
7. Um programa de computador em assembly roda em dois computadores. Computador A implementa uma organização Multiciclo e computador B uma implementação Multiciclo com Pipelining. O período de clock de ambos os computadores é de 40ns. Assuma que o programa contendo um total de 420.000 instruções é executado e que do total de instruções 75% são instruções do tipo R, 5% instruções BEQ/BNE/SLT, 10% instruções do tipo LW, 5% do tipo SW, 4% J, e 1% JAL.
  - a) Qual a frequência de clock em que ambos os computadores opera?
  - b) Quanto tempo o computador A levará para executar o programa?
  - c) Assumindo execução sem stalls, quanto tempo o computador B levará para executar o programa?
  - d) Assuma que para a resolução de todos os hazard de dados, o computador B requerirá 10% mais instruções NOP. Neste contexto quanto tempo B levará para executar o programa? Ele é mais rápido ou lento que A?
8. **Quais são os quatro princípios de projeto desejáveis para o desenvolvimento de um processador que explore o máximo possível a técnica de pipelining?** Cite-os e desenvolva a razão porque eles são desejáveis.
9. Explique o que são **hazards** no contexto de pipelining listando os três tipos e descrevendo suas características.
10. Forneça um exemplo de código em que ocorre um hazard estrutural.
11. Forneça um exemplo de código em que ocorre um **hazard de dados**.
12. Forneça um exemplo de código em que ocorre um hazard de controle.
13. **O que são stalls, porque eles ocorrem?**
14. Plote o diagrama de alocação de subsistemas para o programa abaixo. Desenvolva o diagrama utilizando stalls quando necessário para os primeiros 50 passos de alocação de subsistemas.

AOC II - Arquitetura e Organização de Computadores  
Prof. Dr. rer. nat. Daniel Duarte Abdala

```
1 #-----
2 #primeiro programa em assembly MIPS32
3 #
4 # DDA 24/11/13
5 #computa a soma dos numeros no array
6 #-----
7 #dados do programa
8 .data
9 arr1: .word 0 : 19      # "array" of words to contain fib values
10 size1: .word 19        #tamanho do array
11 #arr2: .word 1,2,3,5,7,11,13,17,19
12 arr2: .word 1,2,3,4,5,6,7,8,9
13 size2: .word 9
14 #codigo do programa
15 .text
16      addi    $t0, $zero, 0      #inicializa o contador com 0
17      add     $s0, $zero, $zero  #inicializa o registrador da soma com 0
18      la      $t1, size2        #carrega endereço de size2 em $t1
19      lw      $t1, 0($t1)       #carrega o tamanho do array arr2 em $t1
20      la      $t2, arr2         #carrega o end do 1a palavra do arr2 em $t2
21      add     $t3, $zero, $zero  #contador de posição do array
22 LOOP: beq     $t0, $t1, END     #testa se o contador é igual a size2
23      add     $t4, $t2, $t3      #calcula o end. do prox. elemento do array
24      lw      $s1, 0($t4)       #carrega o prox. elemento do array em $s1
25      add     $s0, $s0, $s1      #soma o prox. elemento ao total
26      addi    $t0, $t0, 1       #incrementa o contador
27      addi    $t3, $t3, 4       #incrementa o contador de posição
28      j       LOOP
29 END:  #apresenta um inteiro
30      add     $a0, $s0, $zero    # coloca em $a0 o resultado
31      li      $v0, 1
32      syscall
33      # Termino do programa
34      li      $v0, 10           # código de termino
35      syscall                  # chamada do sistema
```

## DESEMPENHO DE PROCESSADORES

15. Defina com suas palavras o que vem a ser **desempenho**. Como a definição se aplica ao **contexto de computadores**?
16. **Como desempenho de computadores é medido? Forneça a fórmula. Quais são os problemas com a abordagem quantitativa?**
17. Qual a diferença entre largura de banda (**throughput**) e **tempo de resposta**? Defina os termos.
18. Defina os termos tempo de execução e tempo de CPU. **Que fatores adicionais fazem com que o tempo de execução seja superior ao tempo de CPU?**
19. **O que é a lei de Amdahl e como ela impacta o desempenho de computadores multicore? Forneça a fórmula e a explique.**
20. Considere um sistema multicore composto por N cores. Construa uma tabela composta pelas seguintes colunas: número de cores (para, 1, 2, 4, 8, 16, 32, 64, 128 e 256), speedup, taxa de speedup por número de processadores. Plote também um gráfico do número de cores para o speedup ganho. para o cálculo considere que a parte não paralelizável do programa seja de **15%**.
21. **Qual a fórmula para o cálculo de CPI em processadores com pipeline? Explique a razão das alterações.**

## PARALELISMO EM NÍVEIS DE INSTRUÇÕES

22. **O que é paralelismo em nível de instruções** e que técnicas podem ser utilizadas para que tal efeito seja alcançado?

23. Como se pode identificar se duas instruções podem ser executadas em paralelo sem implicar em algum tipo de hazard? Forneça exemplos.
24. O que é um bloco básico de código? Porque este conceito é importante.
25. Defina o conceito de profundidade de pipeline. Qual a profundidade do pipeline do MIPS?
26. Defina o conceito de interlock no contexto de processadores com pipeline.
27. Desenvolva um programa em assembly do MIPS que implemente a seguinte parcela de código:

```
for(i=1; i<=999; i++)  
    x[i] = x[i] + y[i]
```

A seguir identifique os blocos básicos de código no programa.

28. Reescreva o programa do exercício anterior considerando que ele será executado em um processadores com pipeline sem interlock, ou seja, o compilador deverá incluir todos os NOPS necessários.
29. Do que trata a técnica de execução de instruções fora de ordem. Seria possível remover alguns dos NOPS do programa resultante no exercício anterior por meio da utilização desta técnica? Se sim, forneça o programa resultante.
30. Explique e forneça um exemplo de como endereços de memória podem resultar em dependência de dados. Discuta porque eles são difíceis de se detectar.
31. Descreva a ideia geral e a motivação da técnica de desdobramento de loops para a resolução de hazard de dados.
32. Considere a seguinte parcela de código a seguir:

```
LOOP: LW    $S0, 0($S1)  
      NOP  
      NOP  
      ADDI $S4, $S0, 42  
      NOP  
      SW    $S4, 0($S1)  
      ADDI $S1, $S1, -4  
      NOP  
      BNE $S1, $zero, LOOP
```

- a) Considere que o registrador S1 inicia com o valor 4000. Quantos NOPS seriam executados?
- b) Execute um desdobramento triplo do código acima.
- c) qual o ganho em número de execuções (quantos NOPS não seriam executados) no programa resultante do item b)?
- 33.

### PARALELISMO EM NÍVEL DE DADOS

34. O que é paralelismo em nível de dados e de que formas ele pode ser alcançado?
35. Defina os termos MIMD e SIMD e exemplifique algumas arquiteturas que utilizam um ou outro.
36. Qual a maior vantagem do modo de instrução SIMD em relação ao modo MIMD?

37. Forneça pseudo código vetorial para o programa a seguir. Anote qualquer suposição quanto a arquitetura que você fizer.

```
L.D    F0,a
DADDIU R4,RX,#512
LOOP: L.D    F2, 0(RX)
      MUL.D  F2, F2, F0
      L.D    F4, 0(RY)
      ADD.D  F4, F4, F2
      S.D    F4, 0(RY)
      DADDIU RX, RX, 8
      DADDIU RY, RY, 8
      DSUBU  R20, R4, RX
      BNEZ   R20, LOOP
```

38. Com relação ao tempo de execução vetorial de um programa, quais são os fatores que o impactam?
39. Como a banda de transferência de dados de memória pode impactar o desempenho de processadores vetoriais e qual o papel da memória cache na mitigação de impacto deste problema?
40. Escreva um programa em assembly MIMD que executa a multiplicação de dois vetores coluna de N posições. A seguir forneça uma versão SIMD (vetorial) que utilize registradores vetoriais de 4 variáveis.

### PARALELISMO EM NÍVENS DE THREADS

41. Qual seria o ganho real de se utilizar threads em um programa que roda em um uniprocessador?
42. Quais são os dois modelos de memória utilizados em multiprocessadores MIMD? Descreva cada um deles e forneça diagramas que os exemplifiquem.
43. Descreva uma possível estratégia para a manutenção da coerência de memória nos modelos de memória SMP e DSM.

### Principais Opcodes com seus Tempos de Execução por Subsistema

opcode	BI	DI	EXE	MEM	RW	Total
add	40	20	40		20	120ns
addi	40	20	40		20	120ns
sub	40	20	40		20	120ns
and	40	20	40		20	120ns
andi	40	20	40		20	120ns
or	40	20	40		20	120ns
ori	40	20	40		20	120ns
nor	40	20	40		20	120ns
xor	40	20	40		20	120ns
lw	40	20	40	40	20	160ns
sw	40	20	40	40	–	140ns
beq	40	20	20		–	80ns
bne	40	20	20		–	80ns
slt	40	20	20		20	80ns
j	40	20			–	60ns
jr	40	20			–	60ns
jal	40	20			20	80ns
lui	40	20			20	80ns

### Formulário

Desempenho	$Desempenho_x = \frac{1}{TempoExec_x}$
Desempenho Relativo	$n = \frac{Desempenho_x}{Desempenho_y}$
Desempenho da CPU	$tempo\ de\ exec\ da\ CPU = \#ciclos\ de\ clock \times período\ do\ clock$
$CPI_{multi}$	$CPI = \frac{\sum_{i=1}^N \#ciclos\ da\ instrução\ tipo(i)}{N}$
$Tempo_{CPU}$	$tempo_{CPU} = \frac{\#instruções \times CPI}{período\ de\ clock}$
Lei de Amdahl	$s(n) = \frac{1}{B + \frac{1-B}{n}}$