

Ex. 1 Considere a execução do trecho de código abaixo no pipeline de 5 estágios, SEM adiantamento. (a) Modifique o código para que ele execute corretamente num processador SEM lógica de bloqueios (stalls). (b) Reescreva o código para que ele execute com um mínimo de bolhas (pipeline sem adiantamento e nem bloqueios). (c) Qual o ganho de desempenho da resposta de (b) com relação à resposta de (a)? O sufixo `.s` indica instrução PF tipo *single precision* (float), e `f0..f8` são registradores de ponto flutuante de 32 bits.

```
loop:
  ld.s f0, 0(r4)    # load FP single precision
  ld.s f2, 0(r6)
  mul.s f4, f2, f0  # 1 ciclo, mult FP single precision
  add.s f6, f4, f8  # f8 escalar
  st.s f4, 0(r8)    # store FP single precision
  st.s f6, 4(r8)
  addi r4, r4, 4
  addi r6, r6, 4
  addi r8, r8, 8
  bne r8, r9, loop # r9 limite
```

Ex. 2 Considere a execução do trecho de programa acima no pipeline original de cinco estágios (com bloqueios e com adiantamento). Quantos ciclos serão necessários para executar este código? Desenhe um diagrama que mostra as dependências, e outro diagrama que mostra como o código será executado, mostrando as bolhas e/ou adiantamento.

```
add r5, r6, r7
lw r6, 100(r2)
sub r7, r6, r8
sw r7, 200(r2)
```

Ex. 3 O seguinte programa é executado no processador segmentado.

```
la r20, 0x4000.0000
la r21, 800
move r5, r0
Sum: lw r10, 0(r20)
     add r5, r5, r10
     addiu r20, r20, 4
     addiu r21, r21, -4
     bne r21, r0, Sum
```

(a) Considere a CPU segmentada simples, sem nenhuma forma de adiantamento e nem bloqueios. Acrescente ao código o que for necessário para garantir a execução correta deste programa. Qual o número total de ciclos necessários para computar a redução do vetor apontado por `r20`? (b) Otimize o código para reduzir o número de ciclos. Qual o novo número total de ciclos? Qual o ganho? (c) Repita (a) considerando adiantamento. (d) Qual o ganho considerando as respostas de (b) e (c)?

Ex. 4 Projete um circuito eficiente e eficaz de previsão de desvios, com capacidade para conter informações sobre 1024 instruções de desvio. Justifique a eficiência e eficácia do seu projeto.

Ex. 5 (a) Enumere e descreva todas as causas para exceções no processador segmentado visto em aula. Desenhe um diagrama simplificado do processador e mostre os locais onde as exceções

ocorrem e onde são detectadas. (b) Liste a sequencia de eventos associados ao tratamento de uma exceção, considerando todos os efeitos no processador.

Ex. 6 Enuncie a Lei de Amdahl. Como esta lei descreve/modela o desempenho de processadores com (a) ciclo longo e (b) segmentados?

Ex. 7 Circuito de Adiantamento – (i) Escreva as equações lógicas que descrevem o controle dos dois seletores nas entradas A e B da ULA. (ii) Escreva as equações lógicas que descrevem o controle do adiantamento para o estágio de memória. (iii) Escreva as equações que lógicas que descrevem o controle do adiantamento para desvios no estágio de decodificação.

Ex. 8 Controle de Bloqueios – (i) Escreva as equações lógicas que detectam dependências de dados (ALU, uso dos LDs, desvios). (ii) Escreva as equações lógicas que detectam dependências de controle.

Ex. 9 Processadores superescalares – Discuta os problemas ocasionados pela execução de duas instruções por ciclo num processador segmentado de cinco estágios.

Ex. 10 Mostre como implementar as instruções abaixo no processador segmentado com cinco estágios. Sua implementação não pode introduzir nenhum risco estrutural e deve ter um custo relativamente baixo. Indique quaisquer modificações necessárias e mostre a tabela de sinais de controle ativos em cada um dos estágios de execução da instrução. SE não é possível implementar a instrução sem a adição de riscos estruturais ou custo elevado, justifique. A vírgula significa "execução simultânea". Note que estas instruções não fazem parte do conjunto de instruções do MIPS para os fins desta disciplina.

```
lwpi rt, desl(rs)   # R[rt] <- M[ R[rs]+desl ] , R[rs] <- R[rs]+desl
swpi rt, desl(rs)   # M[ R[rs]+desl ] <- R[rt] , R[rs] <- R[rs]+desl
ldi rd,rs,rt        # R[rd] <- M[ R[rs] + R[rt] ] - load indexado
bal desl             # R[r31] <- PC+8 , PC <- (PC+4) + ext( desl<<2 )
bgtzal rs,desl      # if(rs >= 0) R[31] <- PC+8 , PC <- (PC+4)+ext(desl<<2)
b desl              # PC <- (PC+4) + ext( desl<<2 ) - branch always
```

Ex. 11 Enquanto estudamos processadores segmentados, discutimos várias técnicas para aumentar o desempenho dos processadores, algumas delas listadas abaixo. Para cada um dos itens abaixo, explique clara e sucintamente como/porque este produz ganho de desempenho. (a) Segmentação em cinco estágios; (b) eliminação de riscos estruturais; (c) eliminação de riscos de dados [causados por dependências de dados]; (d) eliminação de riscos de controle [causados por dependências de controle]; (e) super-escalaridade; (f) previsão dinâmica de desvios; (g) execução especulativa.

Ex. 12 Considere um circuito combinacional que não é segmentado e têm latência T ; para este circuito, a taxa de computação é $1/T$. O ganho de desempenho de uma implementação segmentada pode ser modelada pela Equação que define G , na qual T é a latência do circuito original sem segmentação, S é o atraso adicional introduzido pelos registradores e k é o número de estágios, supondo que a computação possa ser dividida em k estágios iguais.

$$G = 1/(T/k + S) \quad (1)$$

Assim, $(T/k + S)$ é o tempo mínimo necessário para a computação em cada estágio do novo sistema, e $1/(T/k + S)$ é sua taxa de computação (*throughput*). A Equação 1 é similar à Lei de Amdahl. (a) Descreva a relação entre estas duas equações, e (b) dê uma explicação intuitiva para a semelhança entre elas.