1. **Quais as vantagens de um processador multiciclo em relação a um uniciclo?**

**Resposta:** possibilidade de fazer com que as instruções sejam executadas em quantidades diferentes de períodos de clock e compartilhamento de unidades funcionais dentro do espaço de tempo necessário à execução de uma única instrução.

1. **Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?**

**Resposta:** inserção de bancos de registradores depois da memória de instruções e da unidade de controle e na saída ULA

**3) Considerando o pipeline do MIPS (simples com MEM compartilhada para instrução e dados) e uma iteração de loop conforme o trecho de programa abaixo, relacione os conflitos que podem ocorrer e seus consequentes stalls. Qual o speedup (por iteração) para o programa em relação à versão sem pipeline?**

Loop: **subi $t2**, $t2, 4

**lw $t1**, 0(**$t2**)

**add $t3**, **$t1**, $t4

**add $t4**, **$t3**, **$t3**

**sw $t4**, 0(**$t2**)

**beq $t2**, $0, loop

**Resposta:** conflitos em $t3 e $t4, foi carregado $t1 e registrador $t4 não foi tratado, $t4 está operando com o registrador $t3 onde usa $t4 na instrução anterior. O STALL gerado em $t4 na instrução ADD $t3 será obtido na instrução SW $t4. O SPEEDUP (número de estágios) do programa em execução é de 4 em um ciclo e 6 no próximo ciclo.

**4) No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) existentes.**

**div.d** F1, F2, F3

**sub.d** F4, F5, F1

**s.d** F4, 4(F10)

**add.d** F5, F6, F7

**div.d** F4, F5, F6

Resposta:

Instrução 1 = **div.d** F1, F2, F3 = WAR (Write After Read)

Instrução 2 = **sub.d** F4, F5, F1 = WAW (Write After Write)

Instrução 3 = **s.d** F4, 4(F10) = WAR

Instrução 4 = **add.d** F5, F6, F7 = WAR

Instrução 5 = **div.d** F4, F5, F6 = RAW (Read After Write)

**5) Em relação a memória cache. Um computador tem CPI 1 quando todos os acessos à memória acertam no cache. Loads e Stores totalizam 50% das instruções. Se a penalidade por miss é de 25 ciclos e o miss rate é 2%, qual o desempenho relativo se o computador acertar todos os acessos?**

Resposta:

CPU execution time = (CPU clock cycles + **Memory stall cycles**) X Clock cycle = (IC X CPI + 0) X Clock cycle = IC X 1,0 X

**Memory stall cycles** = IC X Mem accesses/Instruction X Miss penalty = IC X (1 + 0,5) X 0,02 X 25 = IC X 0,75

CPU execution time cache= (IC X 1,0 + IC X 0,75) X Clock cycle

CPU execution time / CPU execution time cache = IC X (1,0 + 0,75) X Clock cycle / IC X 1,0 X Clock cycle = 1,75 / 1,0 = 1,75

**6) Descreva os seguintes conceitos:**

**a) Write through**

**b) Write back**

**c) Localidade Temporal**

**d) Localidade Espacial**

**Resposta:**

1. Write through = escrita simultânea em ambas as memorias cache e principal.
2. Write Back = escrita apenas em memória cache e atualização quando o bloco da memória for substituída.
3. Localidade Temporal = se um dado já possui referencia na memória, ele tenderá a ser novamente referenciado na próxima atualização. Como por exemplo os Loops de instruções e dados
4. Localidade Espacial = se um dado já possui referencia na memória, os endereços próximos tenderão a ser referenciados também. Como por exemplo o acesso a dados de um array.