

1. Descreva os tipos de *hazards* que podem ocorrer na arquitetura MIPS e as formas para evitá-los.
2. Considere o seguinte fragmento de código na linguagem C:

```

Código em C
A = B + E;
C = B + F;
```

Aqui está o código MIPS gerado para esse segmento, supondo que todas as variáveis estejam na memória e sejam endereçáveis com *offsets* a partir de \$t0:

```

Código em MIPS
lw  $t1, 0($t0)
lw  $t2, 4($t0)
add $t3, $t1, $t2
sw  $t3, 12($t0)
lw  $t4, 8($t0)
add $t5, $t1, $t4
sw  $t5, 16($t0)
```

Encontre os *hazards* neste segmento de código e reordene as instruções para evitar quaisquer *pipeline stalls* (Suponha que o processador implementa *forwarding*).

3. Para cada sequência de código a seguir, indique se ela deverá sofrer *stall*, pode evitar *stalls* usando apenas *forwarding*, ou pode ser executada sem *stall* ou *forwarding*.

```

Sequência 1
lw  $t0, 0($t0)
add $t1, $t0, $t0
```

```

Sequência 2
add $t1, $t0, $t0
addi $t2, $t0, 5
addi $t4, $t1, 5
```

```

Sequência 3
addi $t1, $t0, 1
addi $t2, $t0, 2
addi $t3, $t0, 2
addi $t3, $t0, 4
addi $t3, $t0, 4
addi $t5, $t0, 5
```

4. Uma arquiteta de computadores precisa projetar o *pipeline* de um novo microprocessador. Ela tem um núcleo de um programa de exemplo com  $10^6$  instruções. Cada instrução exige 100ps para ser executada.
  - (a) Quanto tempo será necessário para executar este núcleo de programa em um processador sem *pipeline*?
  - (b) O microprocessador mais moderno tem 20 estágios de *pipeline*. Suponha que ele tenha um *pipeline* perfeito. Quanto é o ganho de velocidade conseguido em comparação com o processador sem *pipeline*?
5. Compare o desempenho para os controles: (1) ciclo único; (2) multiciclo; (3) *pipelining*; usando o mix de instruções SPECint2000:

- 25% de loads e 10% de stores
- 11% de branches e 2% de jumps
- 52% de instruções da ALU

Para a execução em *pipeline*, considere que metade das instruções de *load* é seguida imediatamente por uma instrução que utiliza o resultado, que o atraso de desvio na previsão errada é de 1 ciclo de clock, e que a quarta parte dos desvios tem previsão errada. Considere que os jumps sempre pagam um ciclo de clock inteiro de atraso, de modo que seu tempo médio é de 2 ciclos de clock. Ignore quaisquer outros *hazards*. Os tempos das unidades funcionais são:

- 200ps para acesso à memória;
- 100ps para operação da ALU;
- 50ps para leitura ou escrita no banco de registradores

6. Em um programa com várias instruções em sequência no formato *lw, add, lw, add, . . .* onde a instrução *add* depende apenas da instrução *lw* imediatamente antes dela. A instrução *lw* também depende apenas da instrução *add* imediatamente antes dela. Se o programa for executado no caminho de dados em pipeline, responda:

- Qual seria o CPI real?
- Sem encaminhamento (*forwarding*), qual seria o CPI real?

7. Temos um núcleo de programa com cinco desvios condicionais. O núcleo do programa será executado milhares de vezes. A seguir estão os resultados de cada desvio para execução do núcleo do programa (T para tomado, N para não tomado):

```

Branch 1: T-T-T
Branch 2: N-N-N-N
Branch 3: T-N-T-N-T-N
Branch 4: T-T-T-N-T
Branch 5: T-T-N-T-T-N-T
```

Considere que o comportamento de cada desvio permanece o mesmo para cada execução do programa. Para esquemas dinâmicos, suponha de que cada desvio tenha seu buffer de previsão e que cada buffer seja iniciado com o mesmo estado antes de cada execução. Liste as previsões para os seguintes esquemas de previsão de desvios:

- Sempre tomado
- Sempre não tomado
- Previsor de 1 bit, iniciado para prever "tomado"
- Previsor de 2 bits, iniciado para prever "tomado" levemente

Qual é a exatidão das previsões?