**LISTA DE EXERCÍCIOS 2**

**QUESTÃO 1**

Cada passo é executado em um ciclo de clock; favorece o reuso de hardware; usa mais registradores para armazenar valores intermediários; instruções diferentes poderão usar quantidades diferentes de ciclos de clock.

**QUESTÃO 2**

Separar em estágios o caminho de dados, e utilizar registradores para salvar os dados entre cada um dos estágios.

**QUESTÃO 3**

**QUESTÃO 4**

RAW: Há dependência de sub.d com o resultado do F1 do primeiro div.d;

Há dependência do s.d com o resultado de F4 do primeiro sub.d;

Há dependência do segundo div.d com o resultado do F5 do add.d;

WAW: O sub.d e o segundo div.d com o registrador F4;

Há antidependência de sub.d com add.d com registrador F5.

**QUESTÃO 5**

CPU execution time = (CPU clock cycles + Memory stall cycles) X clock cycles

= (IC X CPI + 0) X clock cycle

= IC X 1.0 X clock cycle

Memory stall cycles = 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

= 1.75 X IC X clock cycle

**QUESTÃO 6**

A) **Write through**: um esquema em que as escritas sempre atualizam a cache e a memória, garantindo que os dados sejam sempre consistentes entre os dois.

B) **Write back:** um esquema que manipula escritas atualizando valores apenas no bloco da cache e, depois, escrevendo o bloco modificando o nível inferior da hierarquia quando o bloco é substituído.

C) **Localidade temporal:** o princípio em que se um local de dados é referenciado, então, ele tenderá a ser referenciado novamente em breve.

D) **Localidade espacial:** o princípio da localidade em que se um local de dados é referenciado, então, os dados com endereços próximos tenderão a ser referenciados em breve.