

1.

Interpretação de instruções: a unidade de controle interpreta o código em binário da instrução em curso recebida da memória principal e consequentemente sabe o que deve ser feito em seguida.

Gerar sinais de controle: gera sinais de controle e envia o Caminho de Dados consigo efetuar a operação correspondente à instrução em curso.

2.

Como o Ciclo de clock indicam quando as tarefas iniciam e Terminam. Assim, se houver mais pulso de clock para executar diferentes instruções o tempo será otimizado, pois cada instrução possui um tempo de execução diferente.

3.

Cada posição na memória possui um endereço fixamente conectado a ela, assim o acesso a qualquer posição é constante e independe de acessos anteriores. Como consequência, o tempo para acessar qualquer posição aleatória na memória é o mesmo devido ao fato do decodificador, elemento que faz o endereçamento na memória, ter um acesso aleatório.

4.

O pipeline, consiste em subdividir a execução em várias instruções em um projeto único, começando a próxima instrução antes da anterior ter sido finalizada. Assim, otimizamos o tempo médio de execução, visto que não espera uma instrução finalizar para começar a próxima, entretanto, a taxa do pipeline é limitada pelo estágio mais lento.

5.

Como balanceamento é a medida da uniformidade do tempo gasto em cada estágio. Assim, quando todos os estágios gastam o mesmo tempo para serem executados, a performance de um sistema computacional aumenta, pois o balanceamento é o mais ideal possível. Por outro lado, quando o balanceamento não é o ideal a taxa do pipeline é limitada pelo estágio mais lento e conseqüentemente a performance diminui.

Logo, o pipeline melhora a performance por meio do aumento do throughput das instruções (aumentando o número de instruções por unidade de tempo, e não por meio da diminuição de execução de uma instrução individual).