1)

Todo processador trabalha em dois ciclos principais, o Ciclo de Busca e o Ciclo de Execução.

Assim que o computador é iniciado, o processador entra no ciclo de busca, em seguida passa para o ciclo de execuçao e logo volta para o ciclo de busca novamente.

Durante o Ciclo de Busca, é a Unidade de Controle que atua. Uma nova instrução é buscada da Memória para que possa ser decodificada. Nessa fase os registradores [[PC(contador de programas)]](http://producao.virtual.ufpb.br/books/edusantana/introducao-a-arquitetura-de-computadores-livro/livro/livro.chunked/go01.html#PC) e [[IR(registrador de instruçao)]](http://producao.virtual.ufpb.br/books/edusantana/introducao-a-arquitetura-de-computadores-livro/livro/livro.chunked/go01.html#IR) são utilizado. O PC é logo lido para se saber que instrução será executada, essa instrução é trazida para o IR e, finalmente, é decodificada pela Unidade de Controle. Assim que esse processo termina, caso a instrução não diga respeito à um laço, ou à uma repetição, o conteúdo de PC é incrementado. Ou seja, PC recebe PC + 1. Assim, no próximo Ciclo de Busca a instrução do endereço seguinte será carregada da memória e executada. Esse comportamento garante a característica de execução sequencial dos programas.

No passo seguinte a CPU entra em Ciclo de Execução. Nessa etapa atua a Unidade de Ciclo de Dados. Agora a Unidade de Controle já sabe exatamente que operação será executada, com quais dados e o que fazer com o resultado. Essa informação é passada para a ULA e os registradores envolvidos. Durante o Ciclo de Execução há cinco possíveis tipos de operação que podem ser executadas:

Processador e memória

trata simplesmente da transferência de dados entre CPU e memória principal;

Processador e Entrada e Saída

diz respeito à transferência de dados entre a CPU e um dispositivo de Entrada e Saída, como teclado, mouse, monitor, rede, impressora etc.;

Processamento de Dados

são operações simplesmente de processamento dos dados, como operação aritmética ou lógica sobre os registradores da CPU;

Controle

são instruções que servem para controlar os dispositivos do computador, como para ligar um periférico, iniciar uma operação do disco rígido, ou transferir um dado que acabou de chegar pela Internet para a Memória Principal;

Operações compostas

são operações que combinam uma ou mais instruções das outras em uma mesma operação.

2)

**Proposito geral**: executar qualquer tipo de software

**Contador de programa:** Sinaliza para a próxima instrução a ser executada;

**Estado de programa**: sinaliza a instruçao de processamento em curso

4)

Monociclica:

São baseadas em um único ciclo de clock, grande o suficiente para acomodar todas as instruções a serem consideradas. Toda a instrução começa sua execução em uma transição ativa do clock e completa a execução na próxima transição ativa do sinal do clock. Assim sendo, todas as instruções gastam o mesmo tempo para serem executadas, tempo este que vai corresponder ao tempo gasto na execução da instrução mais demorada, e que deve obrigatoriamente ser igual ao ciclo do clock.  
  
Apesar de muito simples de entender, este esquema não se revela prático, pois é muito mais lento do que uma outra implementação que permita que classes diferentes de instruções gastem exatamente o tempo necessário às suas execuções. A vantagem desse tipo de implementaçãoestá no fato de que os tempos de execução de cada classe de instrução variam substancialmente.

Multiciclica:

Neste tipo de implementação, cada passo de execução gasta um período do clock. A implementação multiciclo permite que uma unidade funcional seja utilizada mais de uma vez por instrução, uma vez que ela está sendo usada em ciclos diferentes do clock. Estapossibilidade de compartilhamento pode ajudar a reduzir a quantidade de hardware necessário à implementação. Em resumo, as principais vantagens da implementação multiciclo são a possibilidade de fazer com que instruções sejam executadas em quantidades diferentes de períodos do clock, e a capacidade de compartilhar unidades funcionais dentro do espaço de tempo necessário à execução de uma únicainstrução.  
  
Ao final de um ciclo de clock, todos os dados que precisarem ser usados em ciclos subseqüentes devem ser armazenados em um elemento de estado. Os dados a serem usados em outras instruções devem ser armazenados em elementos de estado visíveis ao programador, ou seja, no banco de registradores, no PC ou na memória.

**Pepiline**:

 É uma técnica de [*hardware*](https://pt.wikipedia.org/wiki/Hardware) que permite que a [CPU](https://pt.wikipedia.org/wiki/CPU) realize a busca de uma ou mais instruções além da próxima a ser executada. Estas instruções são colocadas em uma fila de [memória](https://pt.wikipedia.org/wiki/Mem%C3%B3ria_(computador)) dentro do processador (CPU) onde aguardam o momento de serem executadas: assim que uma instrução termina o primeiro estágio e parte para o segundo, a próxima instrução já ocupa o primeiro estágio

Superescalar:

Exploram paralelismo em nível de instruções de maneira a capacitar a execução de mais de uma instrução por ciclo de clock. Este tipo de processador decodiﬁca múltiplas instruções de uma vez e o resultado de instruções de desvio condicional são geralmente preditas antecipadamente, durante a fase de busca, para assegurar um ﬂuxo ininterrupto.

Na **arquitetura superescalar**, várias instruções podem ser iniciadas simultaneamente e executadas independentemente umas das outras. A arquitetura *pipeline* permite que diversas instruções sejam executadas ao mesmo tempo, desde que estejam em estágios diferentes do *pipeline*.