

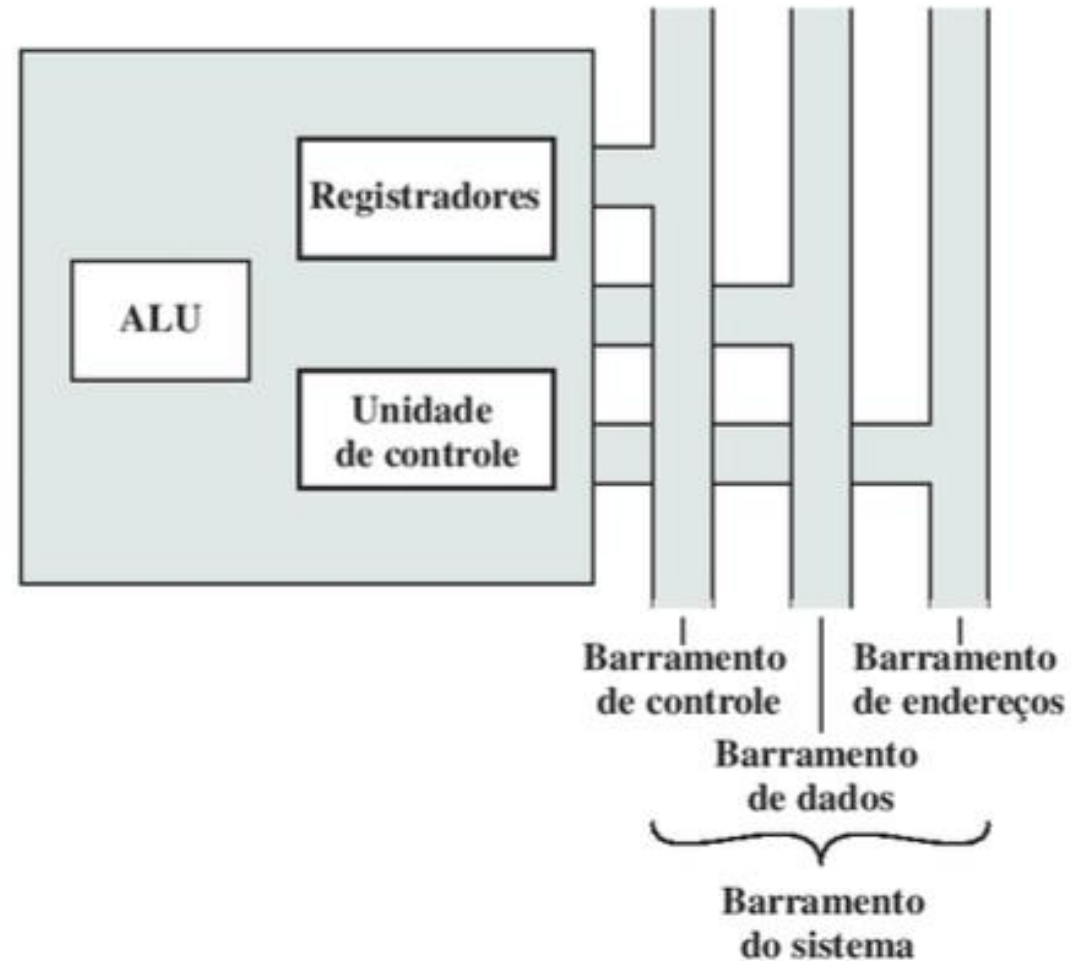
Estrutura e função do processador

Prof. Paulo André Zapparoli

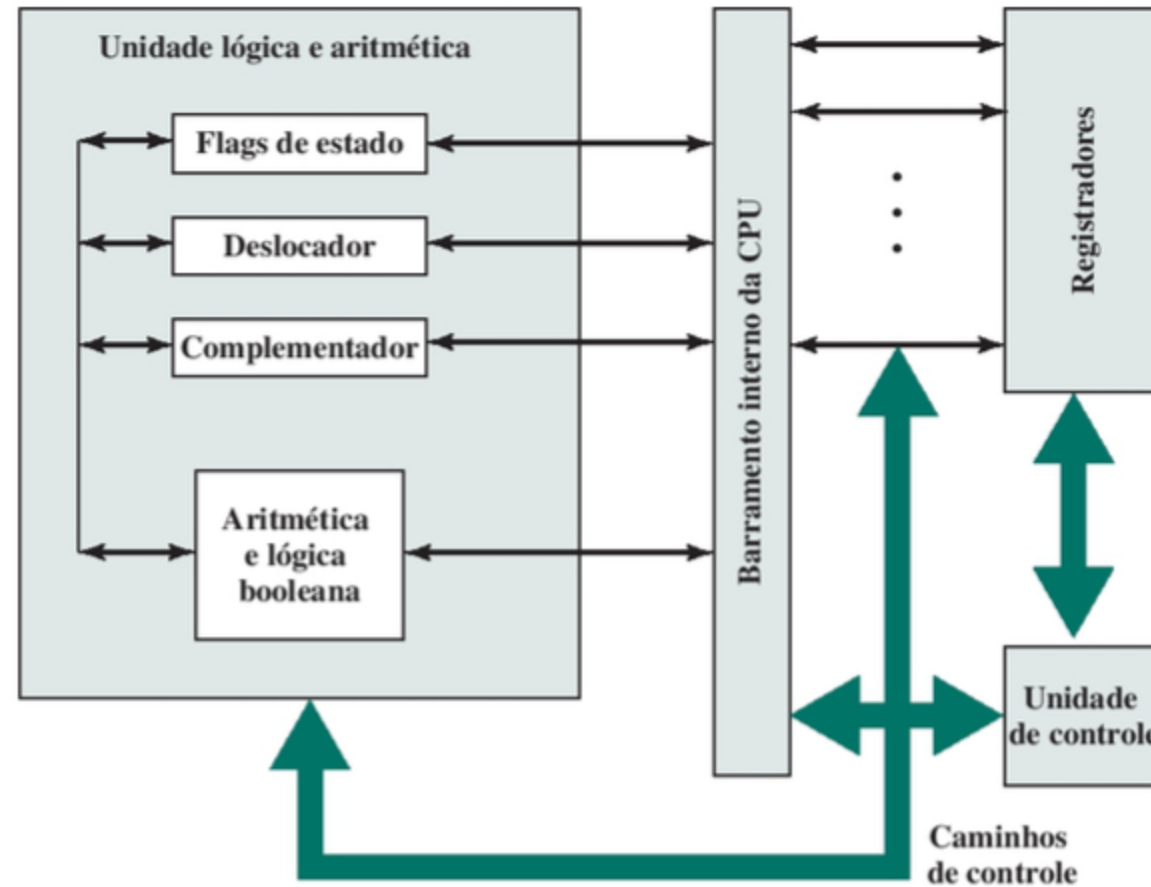
Organização do Processador

- ▶ **Busca da instrução:** o processador lê uma instrução da memória (registrador, cache, memória principal).
- ▶ **Interpretação da instrução:** a instrução é decodificada para determinar qual ação é necessária.
- ▶ **Busca dos dados:** a execução de uma instrução pode necessitar a leitura de dados da memória ou de um módulo de E/S.
- ▶ **Processamento dos dados:** a execução de uma instrução pode necessitar efetuar alguma operação aritmética ou lógica com os dados.
- ▶ **Escrita dos dados:** os resultados de uma execução podem necessitar escrever dados para a memória ou para um módulo de E/S.

Organização do Processador



Organização do Processador



Organização dos Registradores

níveis mais altos da hierarquia, a memória é mais rápida, menor e mais cara (por bit). Dentro do processador, existe um conjunto de registradores que funcionam como um nível de memória acima da memória principal e da cache dentro dessa hierarquia. Os registradores no processador desempenham dois papéis:

- ▶ **Registradores visíveis ao usuário:** possibilitam que o programador de linguagem de máquina ou de montagem minimize as referências à memória principal, pela otimização do uso de registradores.
- ▶ **Registradores de controle e de estado:** usados pela unidade de controle para controlar a operação do processador e por programas privilegiados do Sistema Operacional para controlar a execução de programas.

Organização dos Registradores

Registradores visíveis ao usuário

- ▶ Uso geral.
- ▶ Dados.
- ▶ Endereços.
- ▶ Códigos condicionais.

Organização dos Registradores

Registradores de uso geral podem ser atribuídos para uma variedade de funções pelo programador. Algumas vezes, seu uso dentro do conjunto de instruções é ortogonal para a operação. Ou seja, qualquer registrador de uso geral pode conter um operando para qualquer opcode. Isso permite o verdadeiro uso dos registradores de uso geral. No entanto, frequentemente existem restrições. Por exemplo, pode haver registradores dedicados para ponto flutuante e operações de pilha.

Registradores de dados podem ser usados apenas para guardar dados e não podem ser empregados para calcular o endereço de um operando.

Organização dos Registradores

Registradores de endereços podem ser, de certa forma, de uso geral ou podem ser dedicados para um modo de endereçamento em particular. Os exemplos incluem o seguinte:

- ▶ **Ponteiros de segmento:** em uma máquina com endereçamento segmentado (veja a Seção 8.3), um registrador de segmento guarda o endereço base do segmento. Pode haver múltiplos registradores: por exemplo, um para o sistema operacional e um para o processo atual.
- ▶ **Registradores de índice:** estes são usados para indexar endereços e podem ser autoindexados.
- ▶ **Ponteiros de pilha:** se houver endereçamento de pilha visível ao usuário, então normalmente haverá um registrador dedicado que aponta para o topo da pilha. Isso permite o endereçamento implícito: ou seja, as instruções de pilha (como push, pop e outras) não precisam conter um operando de pilha explícito.

Organização dos Registradores

- **Códigos Condicionais**

(também chamados de *flags*). Códigos condicionais são bits definidos pelo hardware do processador como resultado das operações. Por exemplo, uma operação aritmética pode produzir um resultado positivo, negativo, zero ou *overflow*. Além do resultado que é guardado no registrador ou na memória, um código condicional também é definido. O código pode ser testado na sequência como parte de uma operação de desvio condicional.

Organização dos Registradores

Vantagens	Desvantagens
1. Como os códigos condicionais são definidos por instruções normais aritméticas e de movimentação de dados, eles devem reduzir o número de instruções de comparação e teste (COMPARE, TEST) necessárias.	1. Códigos condicionais acrescentam complexidade, tanto para hardware como para software. Os bits dos códigos condicionais são frequentemente modificados de maneiras diferentes por instruções distintas, tornando a vida do microprogramador e do desenvolvedor de compiladores mais difícil.
2. Instruções condicionais, como BRANCH, são simplificadas em relação a instruções compostas como TEST e BRANCH.	2. Códigos condicionais são irregulares; em geral, eles não fazem parte do caminho principal de dados e, por isso, requerem conexões extras de hardware.
3. Códigos condicionais facilitam desvios múltiplos. Por exemplo, uma instrução TEST pode ser seguida de dois desvios, um para menor ou igual a zero e outro para maior que zero.	3. Muitas vezes máquinas com códigos condicionais devem adicionar instruções especiais que não usam códigos condicionais para situações especiais de qualquer forma, como verificação de bits, controle de loop e operações atômicas de semáforos.
4. Códigos condicionais podem ser salvos em uma pilha durante chamadas sub-rotina junto com outra informação de registro.	4. Em uma implementação de pipeline, códigos condicionais requerem sincronização especial para evitar conflitos.

Organização dos Registradores

Registradores de controle e de estado

- ▶ **Contador de programas (PC):** contém o endereço de uma instrução a ser lida.
- ▶ **Registrador da instrução (IR):** contém a instrução lida mais recentemente.
- ▶ **Registrador de endereço de memória (MAR):** contém o endereço de um local de memória.
- ▶ **Registrador de buffer de memória (MBR):** contém uma palavra de dados para ser escrita na memória ou a palavra lida mais recentemente.

Organização dos Registradores

Registradores de controle e de estado

- ▶ **Sinal:** contém o bit de sinal do resultado da última operação aritmética.
- ▶ **Zero:** definido em 1 quando o resultado é 0.
- ▶ **Carry:** definido em 1 se uma operação resultou em um *carry* (adição) ou um empréstimo (subtração) de um bit de ordem maior. Usado para operações aritméticas de múltiplas palavras.
- ▶ **Igual:** definido em 1 se uma comparação lógica resultou em igualdade.
- ▶ **Overflow:** usado para indicar *overflow* aritmético.
- ▶ **Habilitar/desabilitar interrupção:** usado para habilitar ou desabilitar interrupções.
- ▶ **Supervisor:** indica se o processador está executando no modo supervisor ou usuário. Algumas instruções privilegiadas podem ser executadas apenas no modo supervisor e algumas áreas de memória podem ser acessadas apenas no modo supervisor.

Organização dos Registradores

Registradores de dados	
D0	
D1	
D2	
D3	
D4	
D5	
D6	
D7	

Registradores de endereços	
A0	
A1	
A2	
A3	
A4	
A5	
A6	
A7	

Estado do programa	
Contador do programa	
Registrador de estado	

(a) MC68000

Registradores gerais	
AX	Acumulador
BX	Base
CX	Contador
DX	Dados

Ponteiros e índice	
SP	Ponteiro de pilha
BP	Ponteiro de base
SI	Índice de origem
DI	Índice de destino

Segmento	
CS	Código
DS	Dados
SS	Pilha
ES	Extrato

Estado do programa	
Flags	
Ponteiro de instrução	

(b) 8086

Registradores gerais		
EAX		AX
EBX		BX
ECX		CX
EDX		DX
ESP		SP
EBP		BP
ESI		SI
EDI		DI

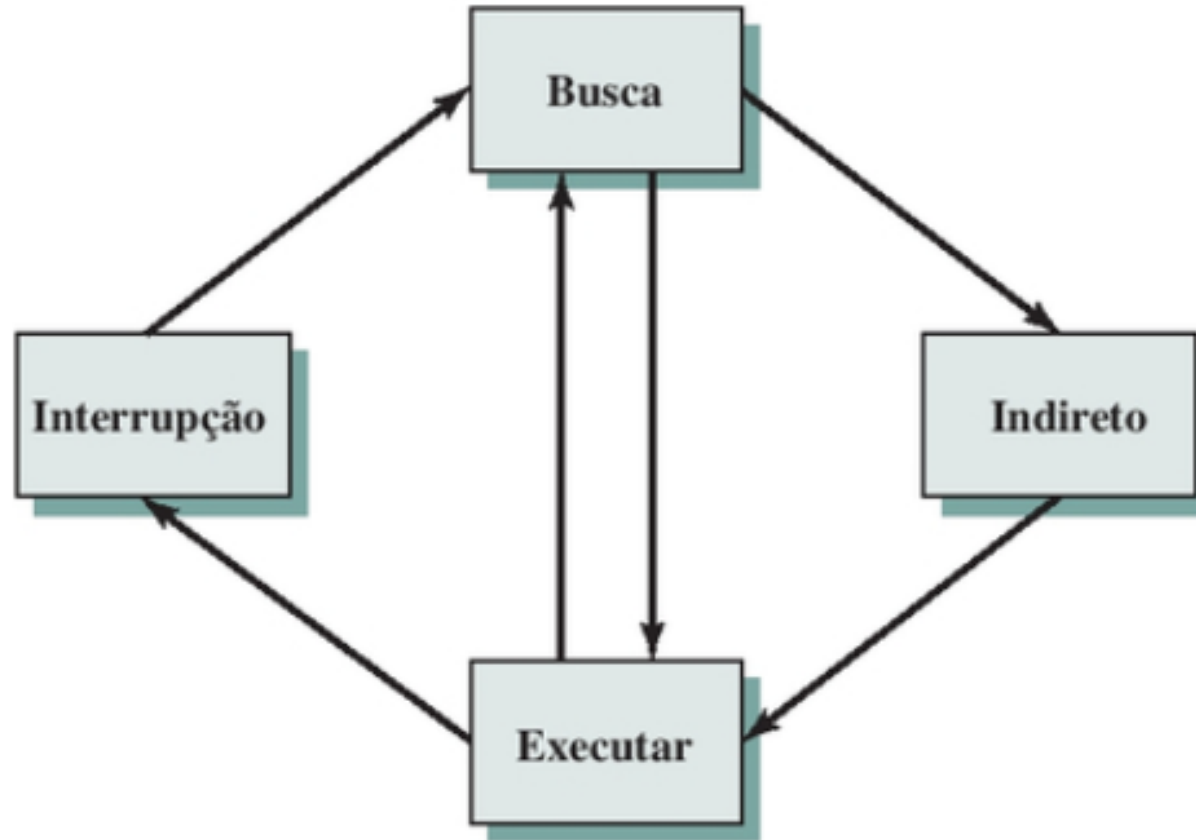
Estado do programa	
Registrador de FLAGS	
Ponteiro de instrução	

(c) 80386—Pentium 4

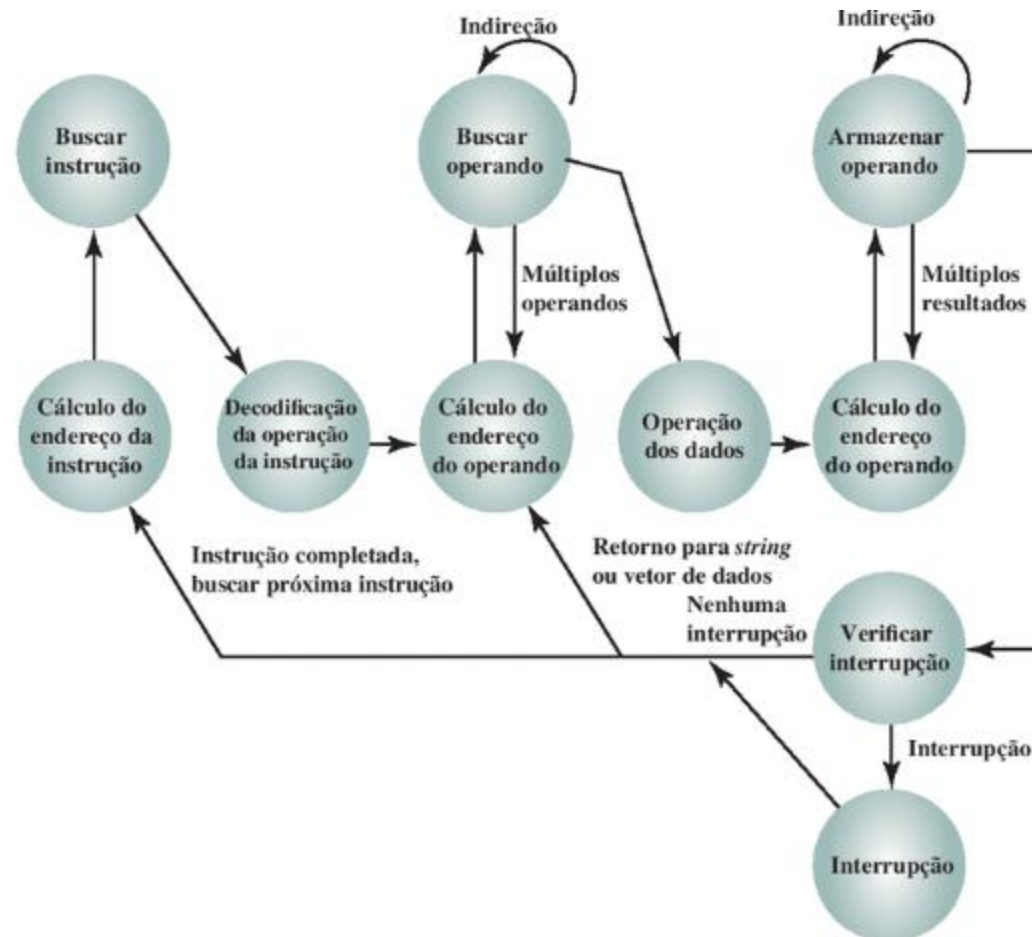
Ciclos da Instrução

- ▶ **Buscar:** lê a próxima instrução da memória para dentro do processador.
- ▶ **Executar:** interpreta opcode e efetua a operação indicada.
- ▶ **Interromper:** se as interrupções estão habilitadas e uma interrupção ocorre, salva o estado do processo atual e atende a interrupção.

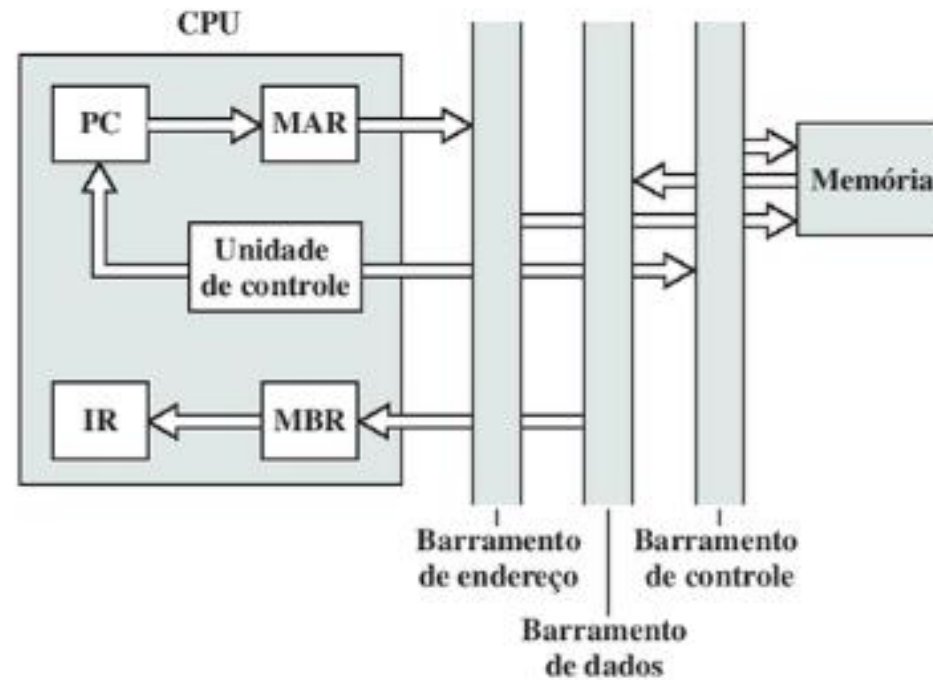
Ciclos da Instrução



Ciclos da Instrução

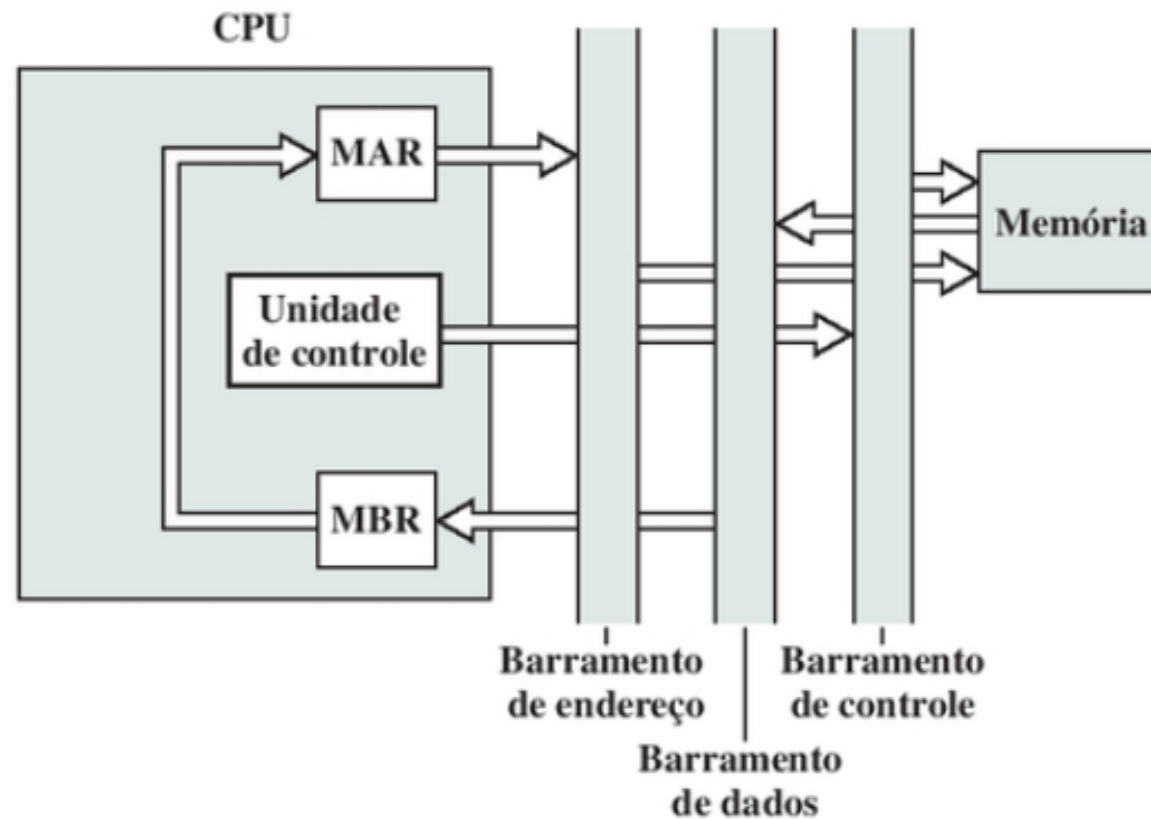


Ciclos da Instrução



MBR = registrador de buffer de memória
MAR = registrador de endereço de memória
IR = registrador da instrução
PC = contador de programa

Ciclos da Instrução



Ciclos da Instrução

