Estrutura e Função do Processador

Arquitetura e Organização de Computadores

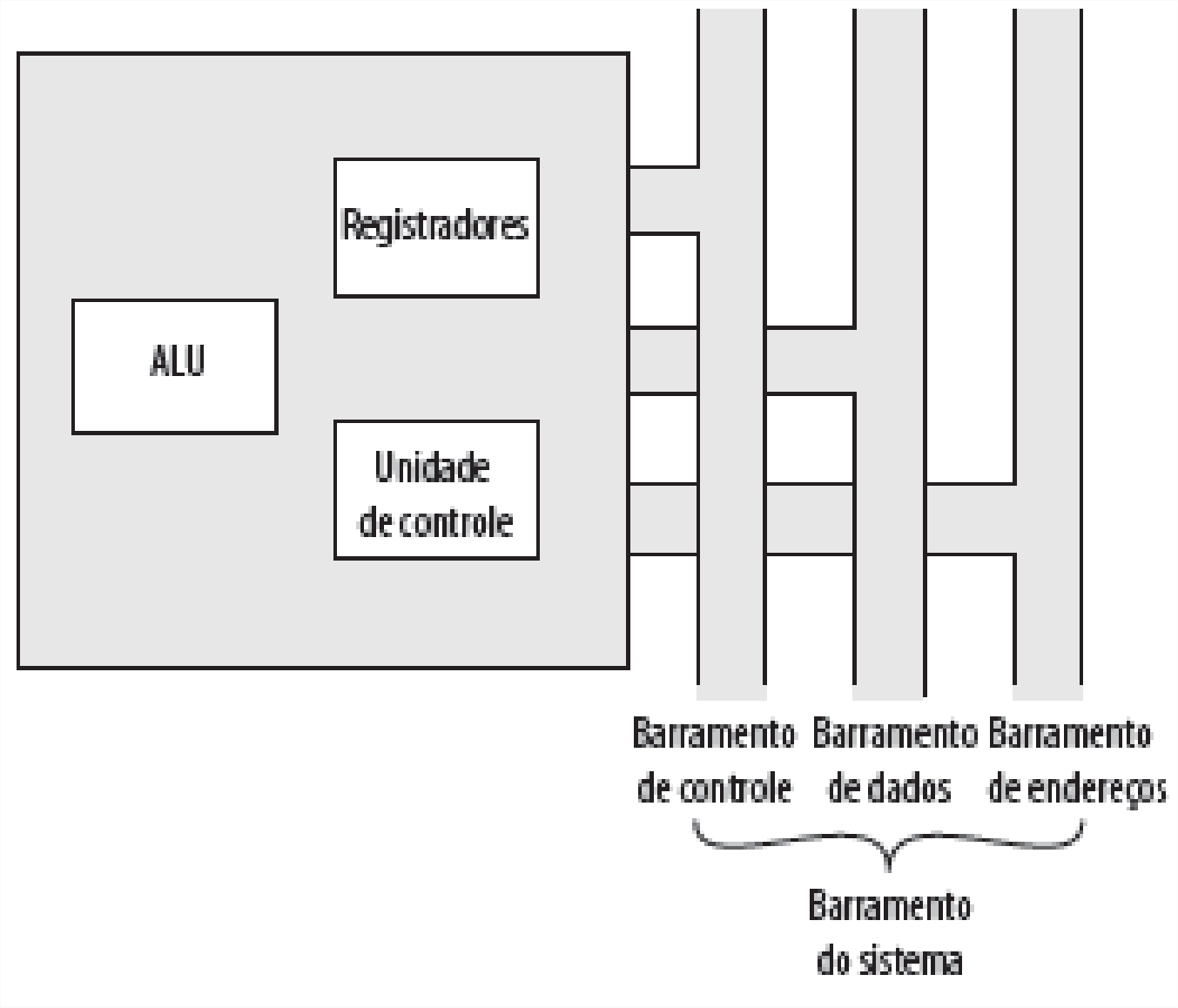
Prof. Lucas de Oliveira Teixeira

UEM

Estrutura do Processador

Estrutura da Processador

Estrutura externa do processador:



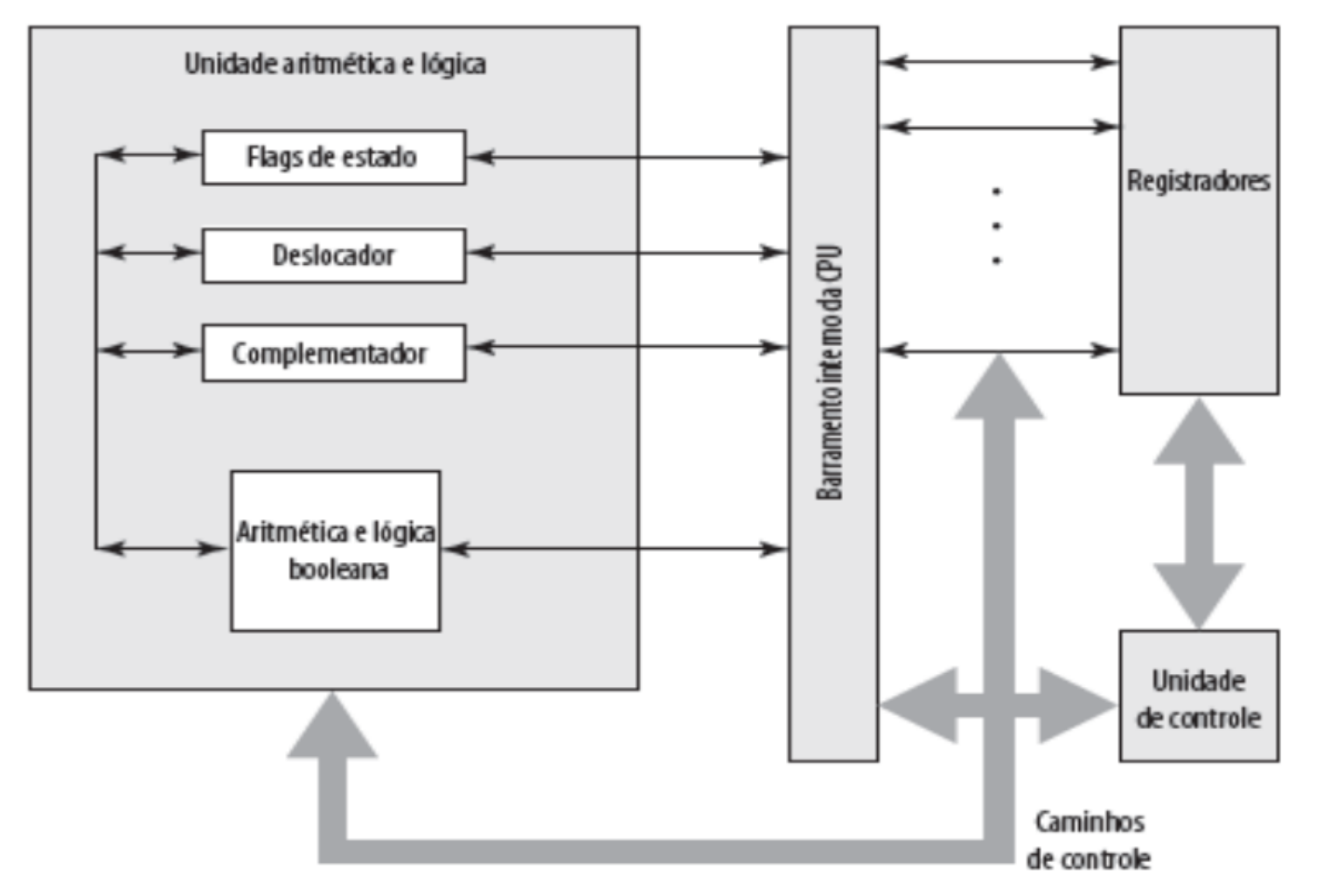
Estrutura da Processador

Estrutura externa do processador:

- Os barramentos são a ponte de comunicação entre o processador e os componentes externos.
- A ULA realiza as operações aritméticas e lógicas.
- Os registradores são uma memória pequena e de alto desempenho dentro do processador.
- A unidade de controle realiza o controle da ULA e do acesso aos registradores e barramentos.

Estrutura da Processador

Estrutura interna do processador:



Registradores

Registradores

- É um dos componente mais importantes da CPU, é necessário para um armazenamento temporário para as operações.
- Quantidade e função de cada registrador variam entre projetos de processadores. É uma das principais decisões de projeto.
- Os registradores são o nível mais alto da hierarquia de memória.

Registradores

Organização:

- Registradores visíveis ao usuário: Possibilitam que o programador de linguagem de máquina minimize as referências à memória, otimizando o uso dos registradores.
- Registradores de controle e estado: Utilizados pela unidade de controle para comandar a operação do processador; e programas privilegiados do sistema operacional para controlar a execução dos programas (processos).

Registradores

Registradores visíveis ao usuário:

- Podem ser referenciados pela linguagem de máquina do computador.
- Subdivididos em:
  - Uso geral: Podem ser utilizados em qualquer ocasião.
  - Dados: Usados apenas para guardar dados.
  - Endereços: Usados para um modo de endereçamento em particular (registradores de índice, ponteiros de pilha).
  - Códigos condicionais: Bits definidos como resultados de operações. Ex.: Operações aritméticas podem gerar resultados positivos, negativos, zero, overflow, etc.

Registradores

Registradores de uso geral:

- Podem ser de propósito geral verdadeiro: Pode conter um operando para qualquer código de operação.
- Podem ser restritos: Dedicados exclusivamente para algumas operações. Exemplo: Ponto flutuante ou operações de multiplicação.
- Podem ser usados para dados ou cálculo de endereçamento: Registradores de segmento base, índice ou topo da pilha.

Registradores

Registradores de uso geral vs uso específico:

- Uso geral:
  - Aumenta o tamanho e complexidade das instrução.
  - Aumenta a flexibilidade e opções na programação.
- Uso específico:
  - Menor flexibilidade e opções na programação.
  - Permite referência implícita aos registradores.
  - Diminui o tamanho e complexidade das instruções.
- A tendência é utilizar registradores especializados.

Registradores

Quantidade de registradores de uso geral:

- A quantidade afeta o projeto do conjunto de instruções: Quanto mais registradores, maior o número de bits para especificar os operandos.
- O número comum é entre 8 e 32.
- Quanto menos, mais referências à memória são necessárias.
- Porém, um número grande não reduz necessariamente as referências à memória e ocupa espaço no processador.
- Obs.: Algumas arquiteturas RISC podem ter centenas de registradores.

Registradores

Tamanho dos registradores de uso geral:

- Grande o suficiente para conter o maior endereço do sistema.
- Grande o suficiente para conter uma palavra de memória.
- É desejável que se possa combinar dois registradores de dados para conter valores de tamanho duplo. Por exemplo, na linguagem C podemos fazer isto:

```
1  int  a;
2  long a;
3
4  float  b;
5  double b;
```

Registradores

Registradores de código condicional:

- Conjuntos de bits individuais (flags), atualizados pelo hardware da CPU como resultado de operações. Exemplo: Se o resultado da última operação foi zero.
- Podem ser lidos (implicitamente) por instruções de máquina. Exemplo: Jump If Zero (JZ).
- Normalmente não podem ser alterados diretamente por instruções.

Registradores

Registradores de controle e estado:

- São utilizados pela UC para controlar a operação da CPU e por programas do SO para controlar a execução de programas.
- Não são visíveis ao usuário na maioria das máquinas.
- Program Counter (PC): O endereço da instrução a ser buscada.
- Instruction Register (IR): A última instrução buscada.
- Memory Address Register (MAR): O endereço de uma posição de memória a ser escrita ou lida.
- Memory Buffer Register (MBR): Um dado a ser escrito na memória ou lido recentemente.

Registradores

Outros registradores:

- Dependendo da CPU, podem existir outros registradores de uso específico:
  - Blocos de controle de processo (visto em SO).
  - Vetores de interrupção (visto em SO).
  - Controle de operações de I/O.
- No projeto da organização dos registradores, as necessidades do SO são determinantes.

Registradores

Processador 8086:

- Registradores de uso geral: AX, BX, CX e DX.
- Registradores de índicoes: SP (stack pointer), BP (base pointer), SI (source register) e DI (destination register).
- Registradores de segmentos: CS (code), DS (data), SS (stack), ES (extra).
- Registradores de estado de programa: flags e ponteiro de instrução.
- Ele possui registradores de uso geral, com registradores de dados de 16 ou 8 bits e índices de 16 bits.

Registradores

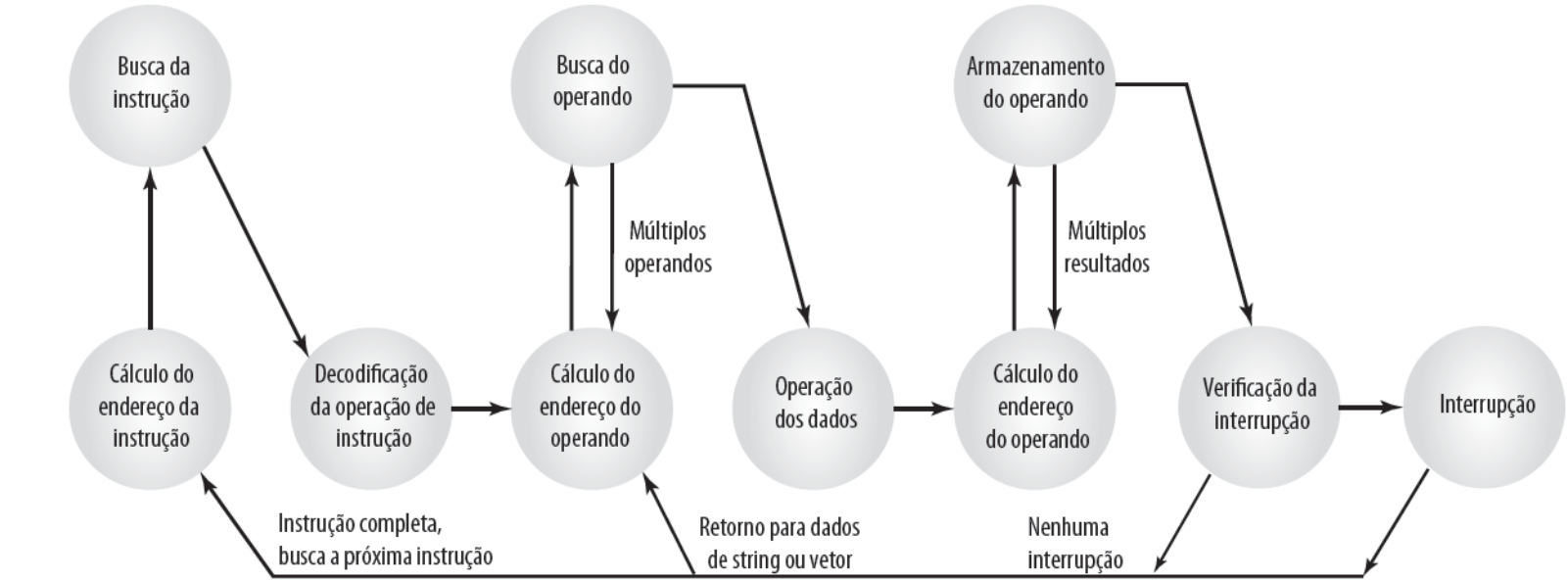
Processador 80386 (Pentium 4):

- Todos os registradores de 32 bits mantendo a organização do 8086.
- Registradores de uso geral: EAX, EBX, ECX e EDX.
- Registradores de índicoes: ESP (stack pointer), EBP (base pointer), ESI (source register) e EDI (destination register).

Ciclo de Instrução

Ciclo de Instrução

Figura 3.12 Diagrama de estado do ciclo de instruções, com interrupções



Ciclo de Instrução

Indireção:

- Uma instrução pode exigir acessos à memória para obter operandos.
- Porém, o endereçamento indireto requer mais acessos à memória. Lembra-se que EA = [A].
- Isto obriga o ciclo de instrução a realizar passos a mais.

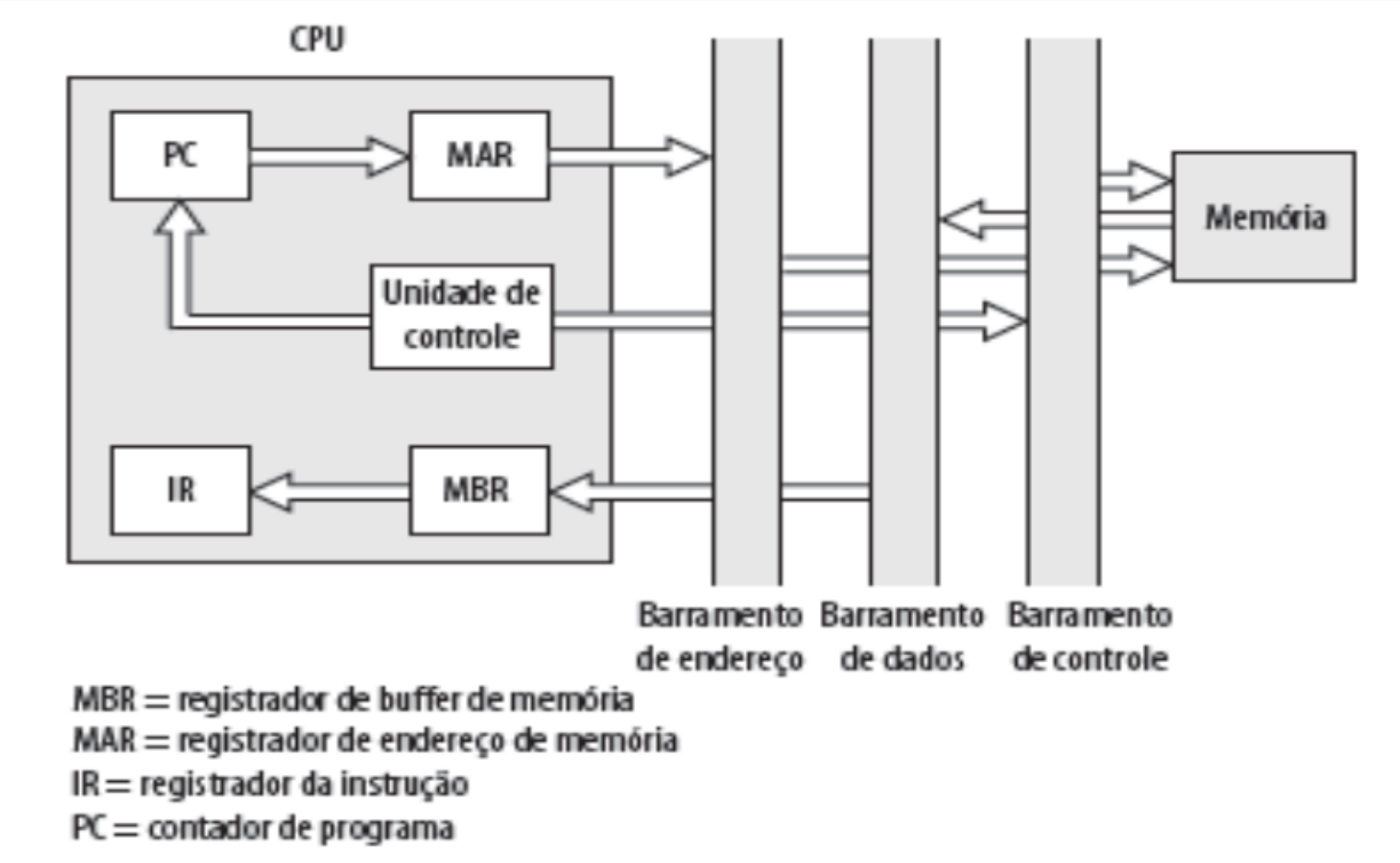
Ciclo de Instrução

Fluxo de dados da busca de instrução:

- PC contém endereço da próxima instrução.
- O endereço é carregado no MAR;
- O endereço é disponibilizado no barramento de endereços;
- A unidade de controle solicita uma operação de leitura da memória;
- O resultado é trazido pelo barramento de dados, copiado no MBR e então para o IR;
- O PC é incrementado.

Ciclo de Instrução

Fluxo de dados da busca de instrução:



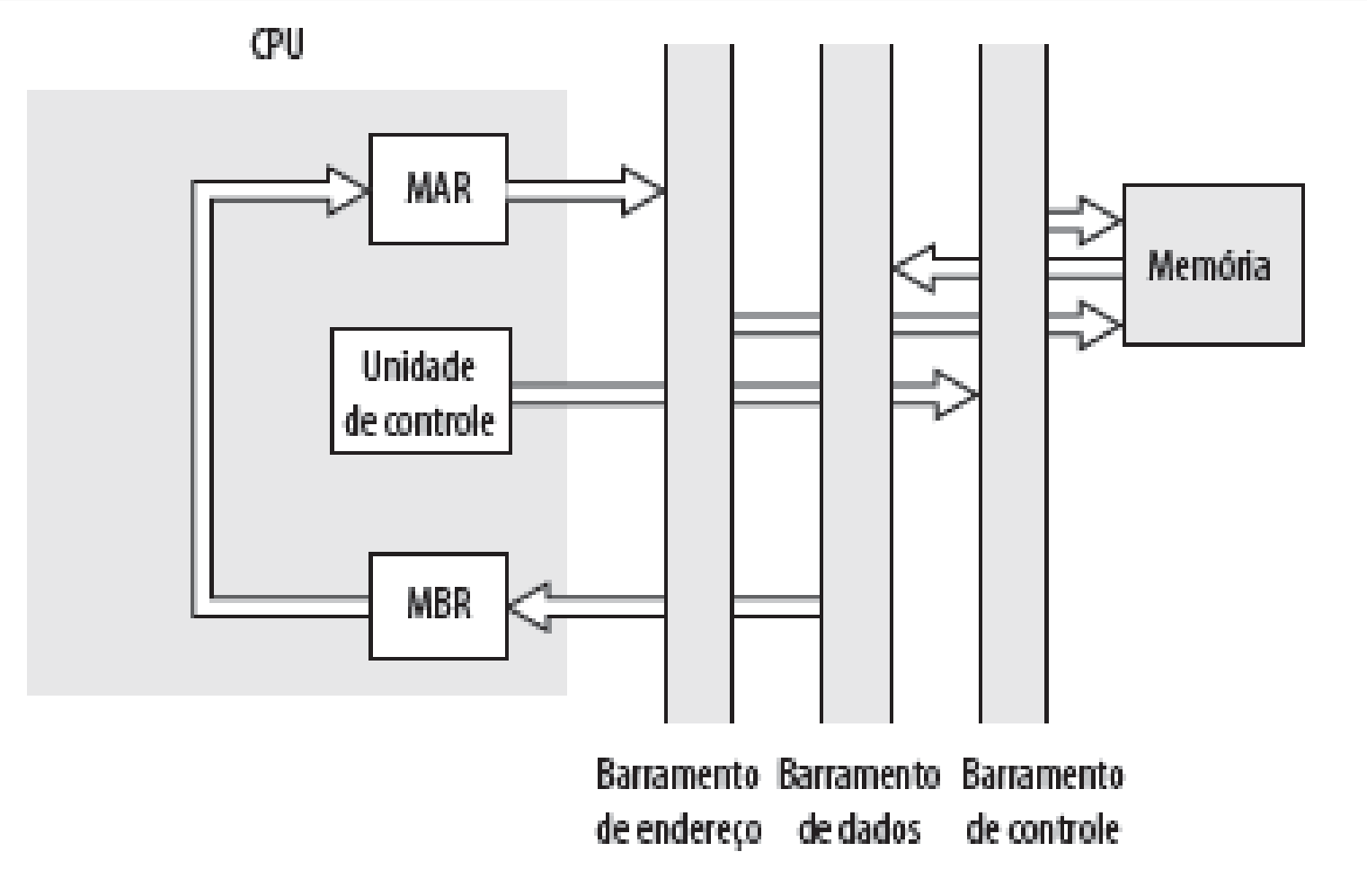
Ciclo de Instrução

Fluxo de dados da busca de instrução:

- IR é examinado.
- Se o endereçamento é indireto, o ciclo indireto é realizado:
  - Os N bits mais significativos do MBR são transferidos para o MAR;
  - A unidade de controle requisita novamente uma operação de leitura na memória;
  - O resultado (isto é, o operando efetivamente) é movido para o MBR.

Ciclo de Instrução

Fluxo de dados da busca de instrução:



Ciclo de Instrução

Fluxo de dados da execução:

- A execução depende da instrução (obviamente!).
- Pode incluir:
  - Operação de leitura ou escrita na memória;
  - Operação de Input/Output;
  - Transferências entre registradores;
  - Operações com a ULA.

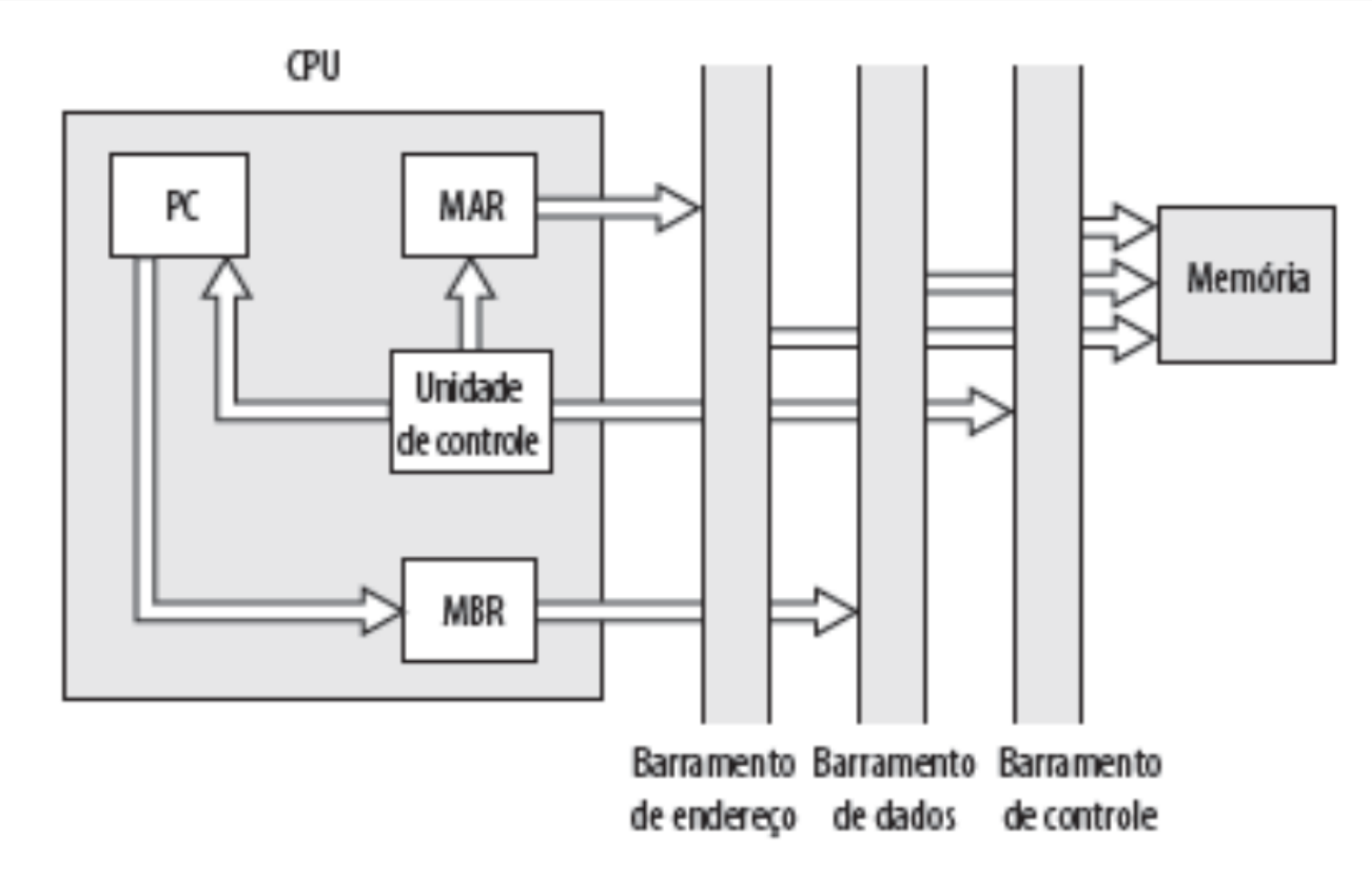
Ciclo de Instrução

Fluxo de dados da interrupção:

- A interrupção é um sinal especial para o processador.
- O conteúdo do PC é salvo para permitir o retorno ao ponto anterior de execução – É copiado para o MBR;
- O ponto de salvamento é carregado no MAR (Ex.: Ponteiro do topo da pilha);
- O MBR é escrito na memória;
- O PC é carregado com o endereço da rotina de manipulação de interrupções;
- A próxima instrução, a primeira do manipulador de interrupções, pode ser carregada.

Ciclo de Instrução

Fluxo de dados da interrupção:



Ciclo de Instrução

Prefetch (busca antecipada):

- A busca da próxima instrução acessa a memória principal.
- A execução normalmente não acessa memória principal.
- Portanto, pode-se buscar a próxima instrução durante execução da instrução atual.
- Isto é chamado busca antecipada da instrução ou instruction prefetch.

Ciclo de Instrução

Prefetch (busca antecipada):

- Aumenta a performance, mas não dobra: A operação de busca (fetch) é em geral mais curta que a execução.
- Em casos de jump (desvios) as instruções pré-buscadas (prefetched) não serão as instruções necessárias.
- Podemos adicionar mais passos para melhorar a performance: com isso surgiu o p,eipeline!