

Arquitetura e Organização de Computadores

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

27 de Agosto de 2024

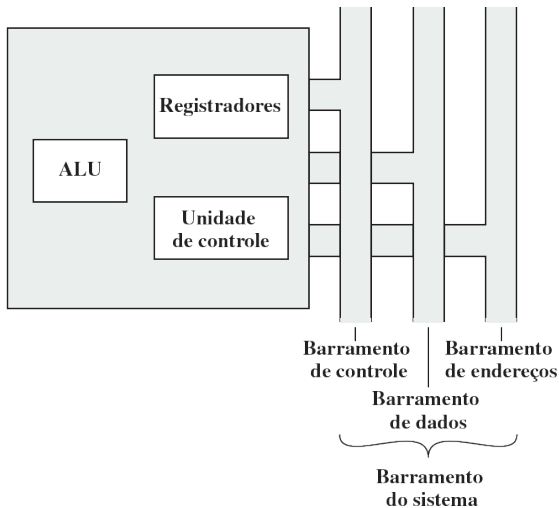
Organização do processador

- Para entender a organização do processador, vamos considerar os requisitos necessários:
- **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.
- Para fazer essas coisas, deve estar claro que o processador precisa armazenar alguns dados temporariamente.

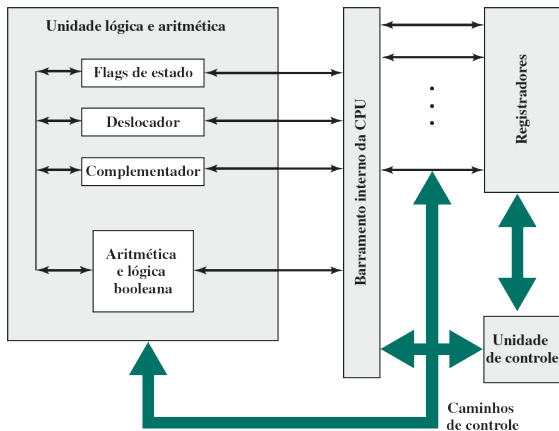
Organização do processador

- CPU com barramento do sistema:



Organização do processador

- Estrutura interna da CPU:



Organização dos registradores

- 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.
- Não há separação clara de registradores nessas duas categorias.

- Um **registrador visível ao usuário** é aquele que pode ser referenciado pelos recursos da linguagem de máquina que o processador executa.
- Podemos dividi-los nas seguintes categorias:
 - Uso geral.
 - Dados.
 - Endereços.
 - Códigos condicionais.

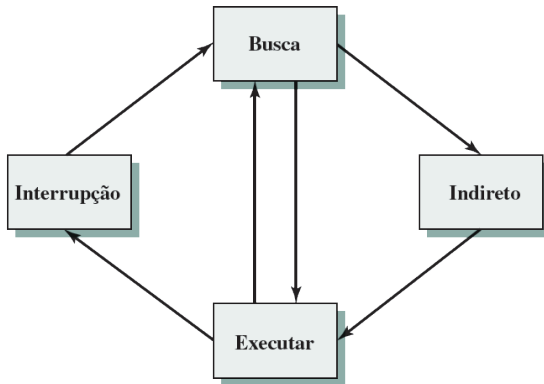
- Quatro registradores são essenciais:
 - **Contador de programas (PC):** contém o endereço de uma instrução a ser lida.
 - **Registrador da instrução (IR):** contém a instrução lida mais recentemente.
 - **Registrador de endereço de memória (MAR):** contém o endereço de um local de memória.
 - **Registrador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

- Muitos modelos de processador incluem um registrador ou conjunto de registradores frequentemente conhecido como **palavra de estado do programa (PSW)**, que contém as informações de estado.
- Em geral, a PSW contém códigos condicionais e outras informações de estado. Campos comuns ou flags incluem:
- **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 ou um empréstimo de um bit de ordem maior.
- **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.

- Um **ciclo de instrução** inclui os seguintes estágios:
 - **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.

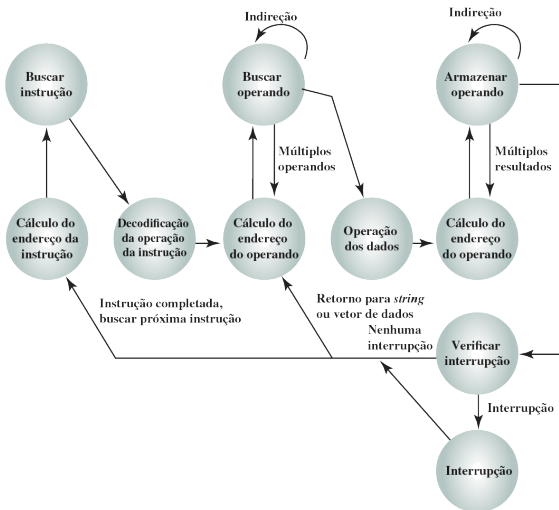
Ciclo da instrução



Ciclo indireto e interrupção

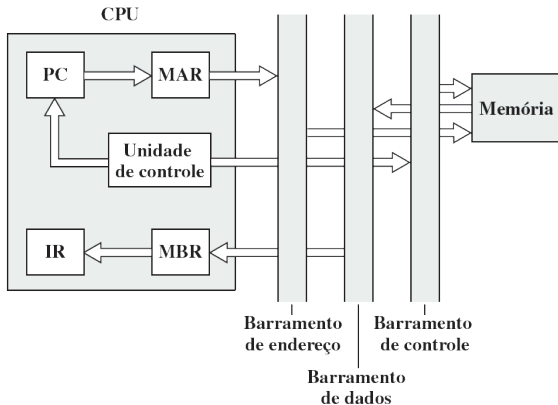
- Ciclo indireto:
 - A instrução buscada anteriormente pode continuar o ciclo de instrução e deve buscar os operandos;
 - Endereçamento indireto acontece quando uma instrução tem um endereço, que aponta para outro endereço;
- Ciclo de interrupção:
 - Após a execução ocorre o ciclo de interrupção;
 - Verifica se ocorreram interrupções no processador;

Ciclo indireto e interrupção



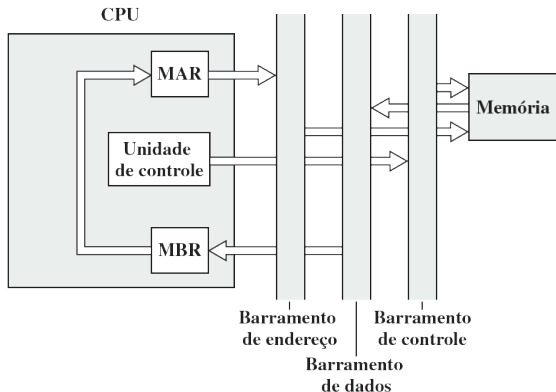
Fluxo de dados

- Fluxo de dados do ciclo de busca:

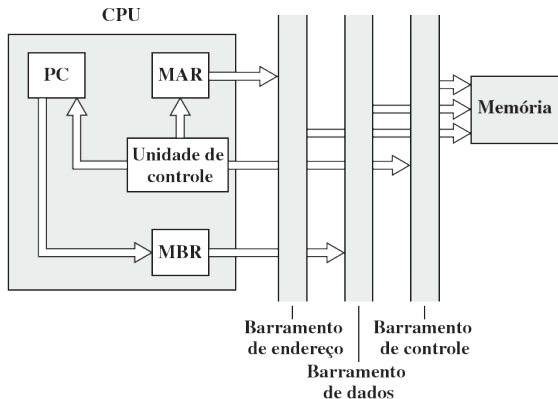


MBR = registrador de buffer de memória
MAR = registrador de endereço de memória
IR = registrador da instrução
PC = contador de programa

- Fluxo de dados do ciclo indireto:



- Fluxo de dados do ciclo de interrupção:



- O **pipeline de instruções** é semelhante ao uso de uma linha de montagem numa planta industrial.
- Para aplicar este conceito à execução da instrução, precisamos reconhecer que, de fato, uma instrução possui vários estágios.
- O pipeline possui dois estágios independentes.
- O primeiro obtém a instrução e a coloca no buffer.
- Quando o segundo estágio está livre, o primeiro passa para ele a instrução do buffer.

- Enquanto o segundo estágio está executando a instrução, o primeiro estágio aproveita qualquer ciclo de memória não utilizado para buscar a próxima instrução e colocá-la no buffer.
- Isso é chamado de **busca antecipada** (prefetch) ou busca sobreposta.
- Deve estar claro que este processo irá acelerar a execução das instruções.
- Se os estágios de leitura e execução forem de duração igual, o ciclo da instrução será reduzida pela metade.

Pipeline de instruções

- Diagrama de tempo para operação do pipeline de instrução:

Tempo →

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instrução 1	FI	DI	CO	FO	EI	WO								
Instrução 2		FI	DI	CO	FO	EI	WO							
Instrução 3			FI	DI	CO	FO	EI	WO						
Instrução 4				FI	DI	CO	FO	EI	WO					
Instrução 5					FI	DI	CO	FO	EI	WO				
Instrução 6						FI	DI	CO	FO	EI	WO			
Instrução 7							FI	DI	CO	FO	EI	WO		
Instrução 8								FI	DI	CO	FO	EI	WO	
Instrução 9									FI	DI	CO	FO	EI	WO

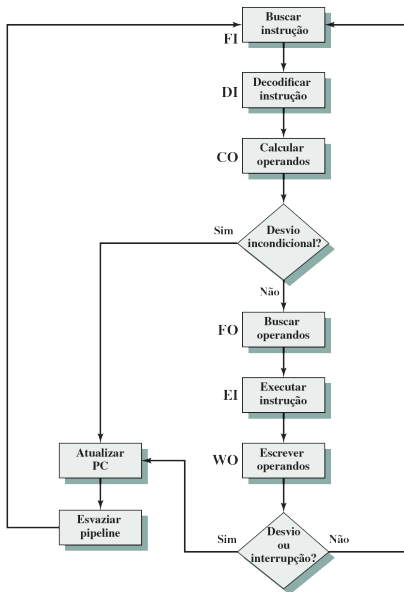
Pipeline de instruções

- O efeito de um desvio condicional na operação do pipeline da instrução:

	Tempo →							← Penalidade por desvio						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instrução 1	FI	DI	CO	FO	EI	WO								
Instrução 2		FI	DI	CO	FO	EI	WO							
Instrução 3			FI	DI	CO	FO	EI	WO						
Instrução 4				FI	DI	CO	FO							
Instrução 5					FI	DI	CO							
Instrução 6						FI	DI							
Instrução 7							FI							
Instrução 15								FI	DI	CO	FO	EI	WO	
Instrução 16									FI	DI	CO	FO	EI	WO

Pipeline de instruções

- Pipeline de instrução de uma CPU de seis estágios:



Pipeline de instruções

- Descrição alternativa de um pipeline:

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I8	I7	I6	I5	I4	I3
9	I9	I8	I7	I6	I5	I4
10		I9	I8	I7	I6	I5
11			I9	I8	I7	I6
12				I9	I8	I7
13					I9	I8
14						I9

Tempo
↓

	FI	DI	CO	FO	EI	WO
1	I1					
2	I2	I1				
3	I3	I2	I1			
4	I4	I3	I2	I1		
5	I5	I4	I3	I2	I1	
6	I6	I5	I4	I3	I2	I1
7	I7	I6	I5	I4	I3	I2
8	I15					I3
9	I16	I15				
10		I16	I15			
11			I16	I15		
12				I16	I15	
13					I16	I15
14						I16

Hazards do pipeline

- O hazard de pipeline ocorre quando o pipeline, ou alguma parte dele, deve parar porque as condições não permitem a execução contínua.
- Tal parada do pipeline é também conhecida como bolha de pipeline.
- Existem três tipos de hazards:
 - Recursos
 - Dados
 - Controle

- Um dos principais problemas ao se projetar um pipeline de instruções é garantir um fluxo estável de instruções para os estágios iniciais do pipeline.
- Uma série de abordagens foram implementadas para lidar com desvios condicionais:
 - Múltiplos fluxos.
 - Busca antecipada do alvo do desvio.
 - Buffer de loops.
 - Previsão de desvios.
 - Desvios atrasados.

- **Múltiplos fluxos**

- Um pipeline simples tem penalidades na execução de uma instrução de desvio.
- Esta abordagem envolve a execução simultânea de múltiplas instruções em paralelo. O processador é projetado para identificar e executar instruções independentes em paralelo.

- **Busca antecipada do alvo do desvio**

- Nesta abordagem, o processador tenta prever qual será o próximo endereço de instrução a ser executado, mesmo antes de saber se haverá um desvio

- **Buffer de loops**

- Se um desvio está para ser tomado, o hardware primeiro verifica se o alvo do desvio já está no buffer.
- Se estiver, a próxima instrução é buscada do buffer.
- O buffer de loops é semelhante, em princípio, a uma cache dedicada para instruções.
- A diferença é que buffer de loop guarda apenas instruções na sequência e tem um tamanho menor, tendo assim um custo menor também.

- **Previsão de desvios**

- Várias técnicas podem ser usadas para prever se um desvio será tomado.
- Entre as mais comuns estão as seguintes:
- Previsão nunca tomada.
- Previsão sempre tomada.
- Previsão por opcode.
- Chave tomada/não tomada.
- Tabela de histórico de desvio.

- **Desvios atrasados**

- O processador introduz um atraso na execução de um desvio condicional. Isso permite que as instruções após o desvio sejam executadas enquanto o processador aguarda a confirmação sobre qual caminho o desvio seguirá.
- Um exemplo instrutivo de um pipeline de instruções é o do **Intel 80486**.
- Ele implementa um pipeline de cinco estágios.

- STALLINGS, W. **Arquitetura e Organização de Computadores**. 10 ed. São Paulo: Pearson, 2017;
- TANENBAUM, A. S. **Organização Estruturada de Computadores**. 5 ed. Pearson 2007;
- HENNESY, J. PATTERSON, D. **Organização e Projeto de Computadores**. 3 ed. Editora Campus, 2005.

Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

guilhermenakahata@gmail.com

<https://github.com/GuilhermeNakahata/UNESPAR-2024>