



ORGANIZAÇÃO DE COMPUTADORES

CONJUNTO DE INSTRUÇÕES

Prof. Dr. Daniel Caetano

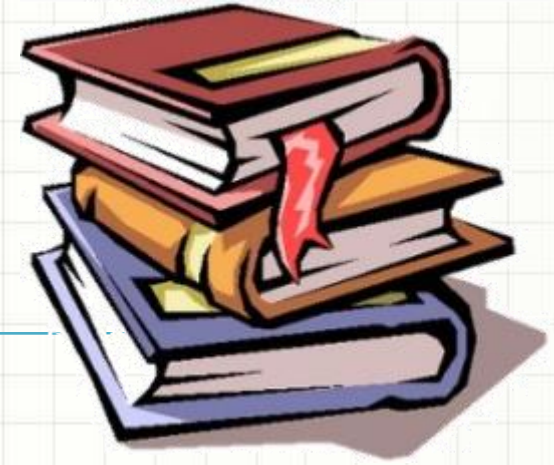
2014 - 1

Objetivos

- Conhecer o formato de uma instrução
- Entender os modos de endereçamento de uma instrução



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>
(Aula 7)

Apresentação

<http://www.caetano.eng.br/>
(Aula 7)

Material Didático

Fundamentos da Arquitetura de Computadores,
páginas 27 a 48

Biblioteca Virtual

Arquitetura e Organização de Computadores, páginas
287 a 426
Introdução à Organização de Computadores, páginas
153 a 203



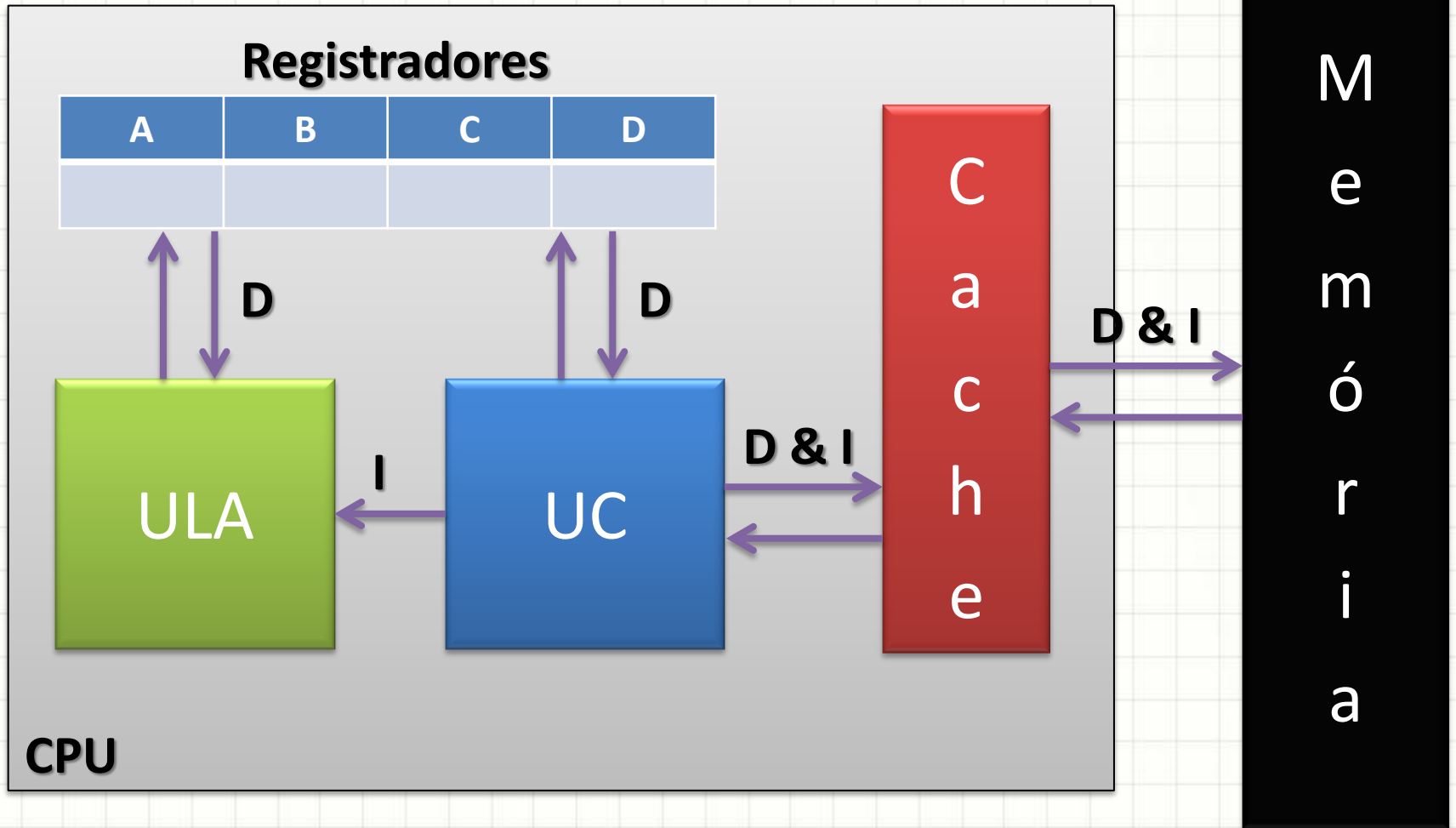
INTRODUÇÃO

Introdução

- Já conhecemos:
 - As partes do computador
 - As partes da CPU
- As instruções orientam a CPU
- Mas como são as instruções?
- Vamos relembrar um pouco sobre a CPU

O Processador e os Registradores

- Diagrama

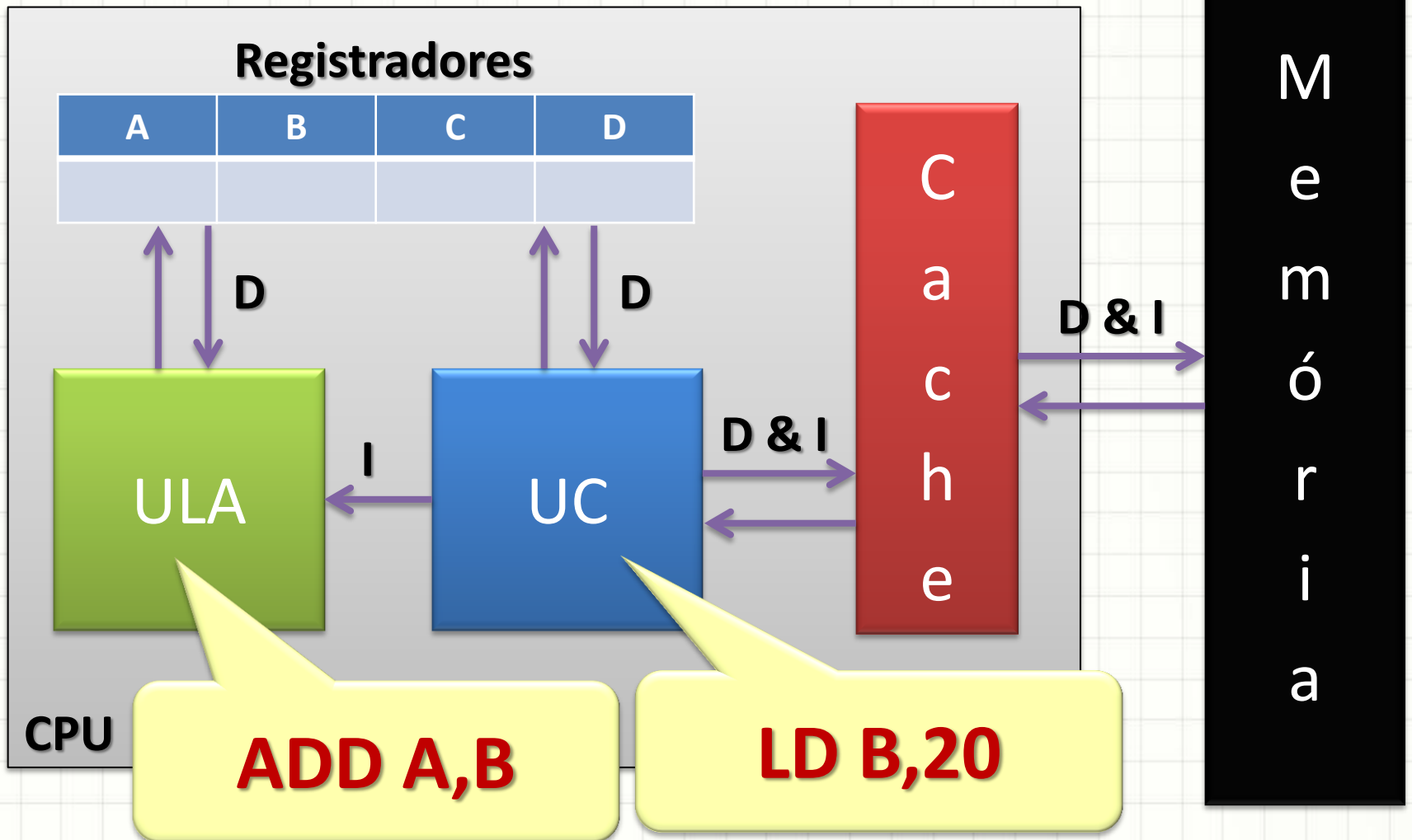


O Processador e os Registradores

- Procedimento de execução:
- UC busca dado na origem (memória, cache, dispositivo)
- UC armazena informação em um registrador
- UC comanda que ULA processe
- ULA lê informação do registrador e processa
- ULA armazena resultado em um registrador
- UC transfere dado do registrador para o destino (memória, cache, dispositivo)

O Processador e os Registradores

- Diagrama





OPERAÇÕES EXECUTADAS PELA ULA

Operações da ULA

- Que tipo de operação a ULA executa?
- Ela é uma espécie de calculadora...
 - Adição (ADD)
 - Subtração (SUB)
 - Multiplicação (MUL)
 - Divisão (DIV)
 - E (AND)
 - Ou (OR)
 - Não (NOT)
 - Ou Exclusivo (XOR)

Operações da ULA

- Em que dados ela opera?
 - SEMPRE nos registradores
 - Usualmente o resultado é armazenado no primeiro registrador
 - ADD A,B Soma B em A
 - NOT A Inverte os bits de A
- Em geral existe um registrador preferencial para armazenar os resultados
 - **Acumulador**
 - A, AX, EAX... (na arquitetura x86)



A UNIDADE DE CONTROLE

Responsabilidades da UC

- Analogia: Pessoa (UC) usando Calculadora (ULA)
- Responsabilidades
 - Controlar a execução de instruções na ordem certa
 - Leitura da memória principal e E/S
 - Escrita na memória principal e E/S
 - Controlar os ciclos de interrupção

Operações da UC

- ULA: **NÃO** executa instruções para acessar a memória e dispositivos
 - Ler (LD)
 - Armazenar (STOR)
 - Movimentar (MOV)
 - Escrita em Dispositivo (OUT)
 - Leitura de Dispositivo (IN)
- Essas instruções são **interpretadas pela UC**



INSTRUÇÕES DA CPU

Instruções da CPU

- Ok... vimos que algumas instruções são executadas pela ULA e outras pela UC...
- Mas quais são as instruções?
- **COMO** são as instruções?

- Vamos começar por “**quais são**”
 - Organizando em categorias

Instruções da CPU

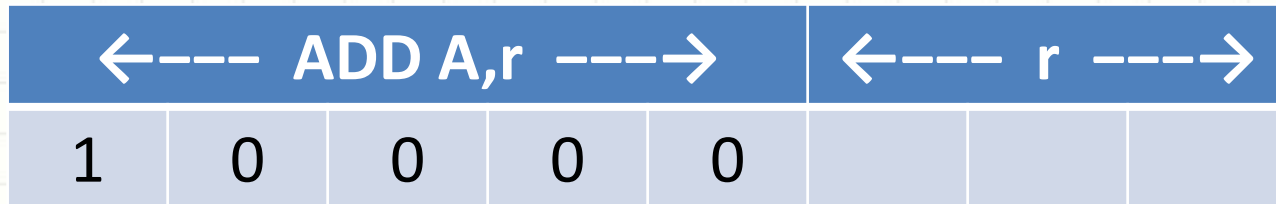
- Transferência de Dados (UC)
 - MOVE, STORE, LOAD, EXCHANGE, PUSH, POP...
- Operações de E/S (UC)
 - READ, WRITE, IN, OUT...
- Operações Aritméticas (ULA)
 - ADD, SUB, MULT, DIV, INC, DEC...
- Operações Lógicas e Conversão (ULA)
 - AND, OR, NOT, XOR, TST, CMP, SHIFT, TRANS, CONV
- Transferência de Controle
 - JUMP, CALL, HALT, BRANCH

Instruções da CPU

- Nem todos os processadores possuem todas essas instruções!
- O conjunto de instruções que um processador aceita é chamado de...
 - **CONJUNTO DE INSTRUÇÕES**
- Ou, em inglês
 - **INSTRUCTION SET**
- Ok... mas **como** são essas instruções?
 - Essas palavras estarão na memória?

Como são as Instruções

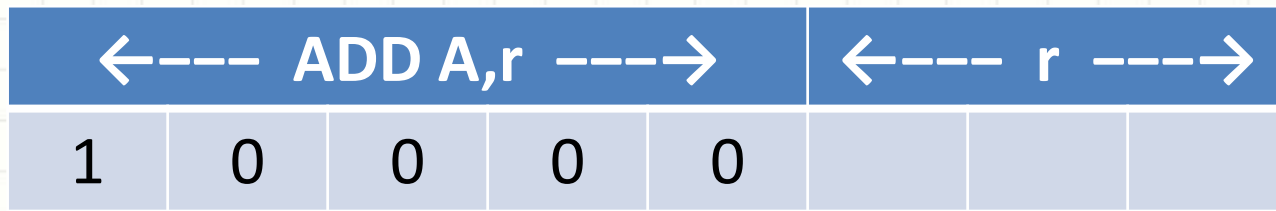
- Obviamente não são letras na memória
- Cada instrução: um conjunto de bits
- Exemplo: **ADD A,r**
 - **Função:** soma o valor de um registrador **r** em **A**



- E o que vai nos bits do “r”?

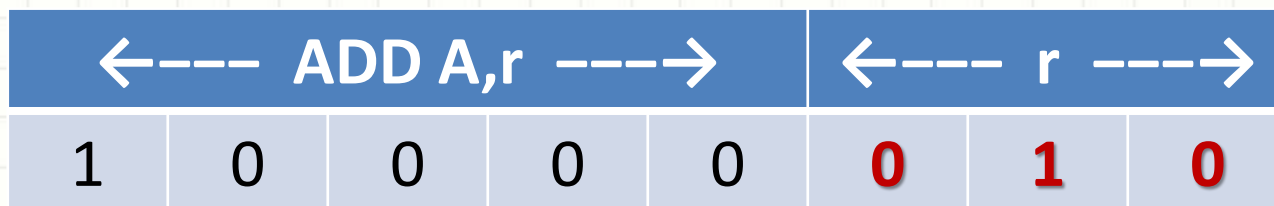
Como são as Instruções

- Exemplo: **ADD A,r**
 - **Função:** soma o valor de um registrador **r** em **A**



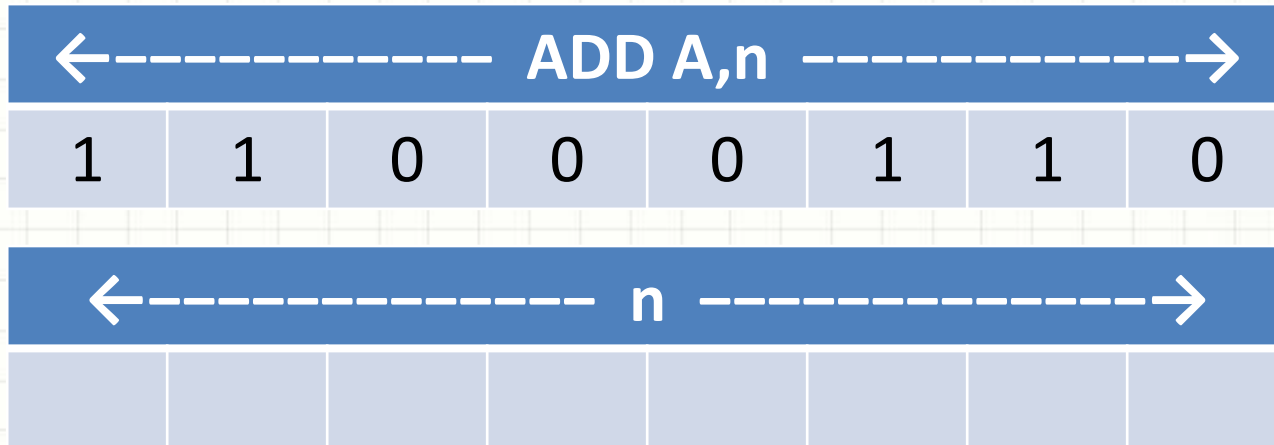
Registrador	A	B	C	D	E	H	L
Bits	111	000	001	010	011	100	101

- Exemplo: **ADD A,D**



Como são as Instruções

- Há instruções que são executadas...
 - Parte pela ULA
 - Parte pela UC
- Exemplo: **ADD A,n**
 - **Função:** soma um número **n** em A

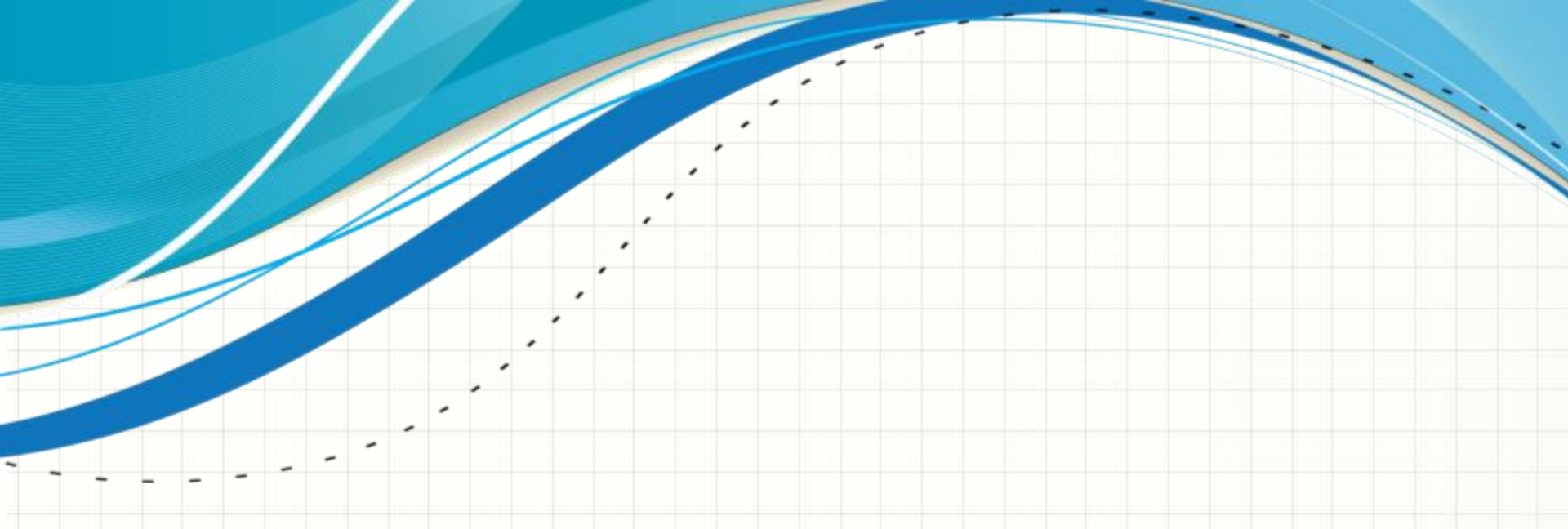


Como são as Instruções

- Exemplo: **ADD A,0x24**
 - **Função:** soma um número **0x24** em A



- Na prática, a UC carrega 0x24 em um registrador temporário **X** e solicita que a ULA some **ADD A,X**



MODOS DE ENDEREÇAMENTO

Modos de Endereçamento

- Vimos que a ULA processa... e a UC busca
- A UC só é capaz de ler um dado após uma instrução?



Modos de Endereçamento

- Vimos que a ULA processa... e a UC busca
- A UC só é capaz de ler um dado após uma instrução?
- **NÃO!**
- A UC consegue ler dados através de diversas estratégias... chamadas **modos de endereçamento**
- Estas estratégias serão apresentadas a seguir

Modos de Endereçamento

- **Endereçamento a Registradores**

- O dado já está em um registrador
- Desnecessário acessar a memória
- Ex.: LD A,B

- **Endereçamento Imediato**

- O dado “faz parte” da instrução
- O dado está em seguida à instrução na memória
- Ex.: LD A,20

Modos de Endereçamento

- **Endereçamento Direto**

- O dado está em uma posição fixa da memória
- Acesso feito indicando o endereço do dado
- Ex.: LD A,(2000)

- **Endereçamento Indireto**

- O dado está em posição variável da memória
- O endereço é indicado por um registrador
- Ex.: LD HL,2000
LD A,(HL)

Modos de Endereçamento

- **Endereçamento por Deslocamento**

- O dado está em uma posição fixa de uma tabela
- Registrador indica o início da tabela na memória
- Ex.: LD IX,2000
LD A,(IX+10)

- **Endereçamento por Pilha**

- O dado é armazenado em um local especial chamado **pilha**
- Não é preciso indicar o endereço
- Ex.: PUSH HL / POP HL



PERGUNTAS?



CONCLUSÕES

Resumo

- CPU: tem um **conjunto de instruções**
 - Instruções: sequências de bits
 - Nomes **mnemônicos** associados a elas
 - ULA não acessa a memória, quem faz é a UC
 - UC: permite muitas formas de indicar o endereço de um dado a ser lido
-
- Como a UC realiza sua tarefa?
 - Que tipo de controle ela faz?