

**ARQUITETURA E ORGANIZAÇÃO DE
COMPUTADORES**

**A UNIDADE LÓGICA ARITMÉTICA E
AS INSTRUÇÕES EM LINGUAGEM
DE MÁQUINA**

Prof. Dr. Daniel Caetano

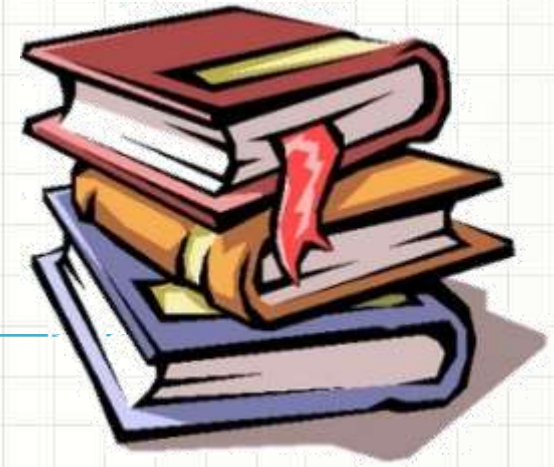
2012 - 2

Objetivos

- Conhecer o processador
- Compreender os registradores
- Conhecer as operações da ULA
- Compreender as Instruções da CPU
- Compreender os modos de Endereçamento
- **Lembretes**
 - Lista 2!



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>
(Aula 10)

Apresentação

<http://www.caetano.eng.br/>
(Aula 10)

Material Didático

Introdução à Organização de Computadores, páginas
153 a 203

Biblioteca Virtual

Arquitetura e Organização de Computadores, páginas
287 a 426



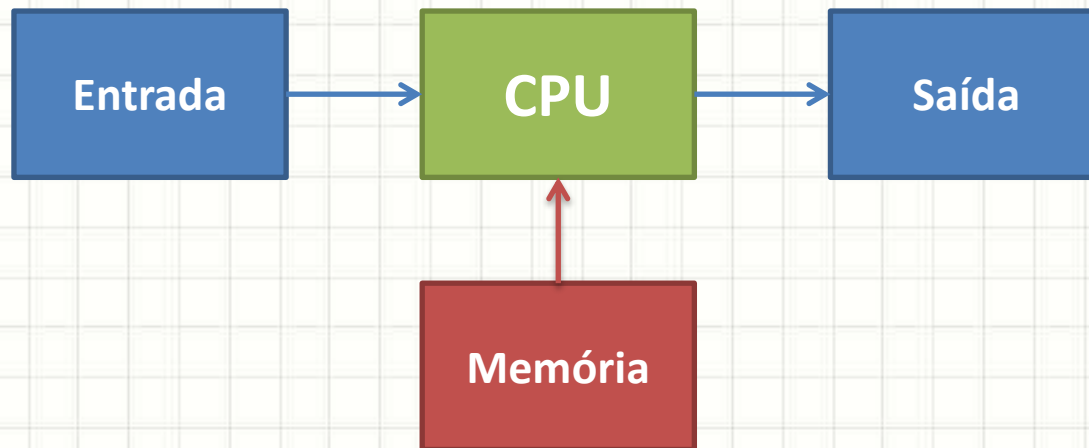
INTRODUÇÃO

Introdução

- Já foi visto o funcionamento global do computador:
 - CPU pega instrução na memória ou em um dispositivos
 - CPU processa instrução
 - CPU grava resultado na memória ou em um dispositivo
- Vamos relembrar?

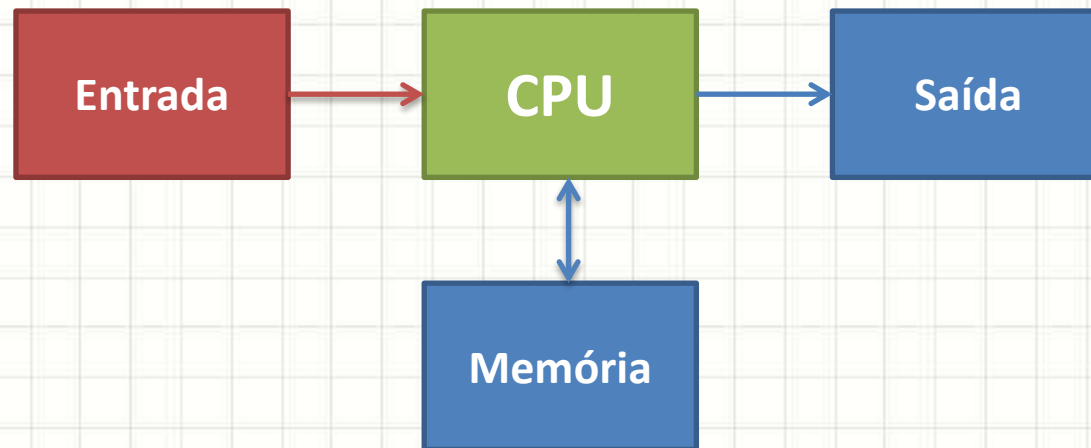
Funcionamento da CPU

- **Busca Instrução:** CPU lê a memória em busca do que deve fazer



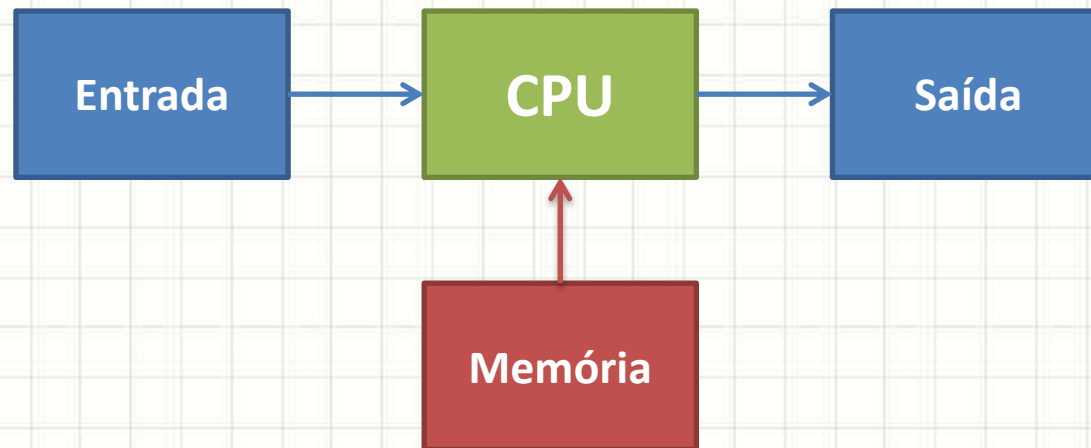
Funcionamento da CPU

- **Lê entrada:** Supondo que a instrução indica a leitura de um dado de entrada, CPU lê entrada



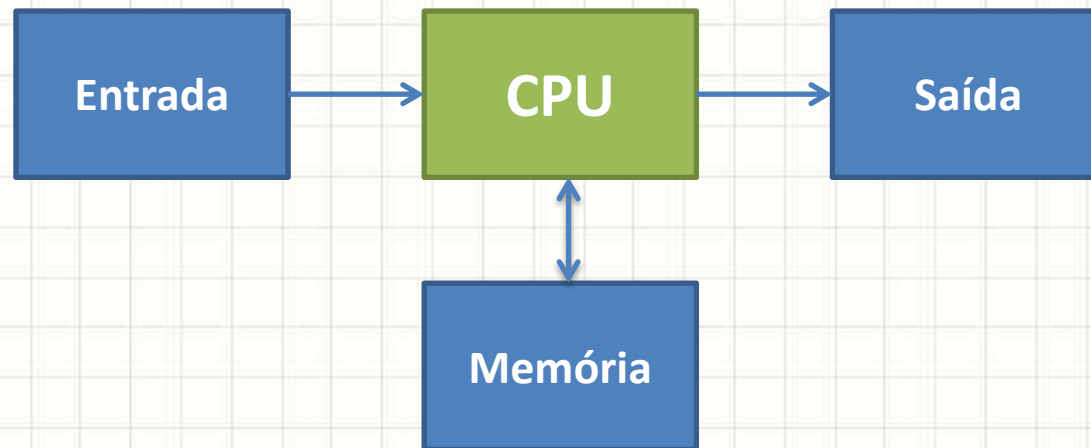
Funcionamento da CPU

- **Busca Instrução:** CPU lê a memória em busca do que deve fazer com dado lido



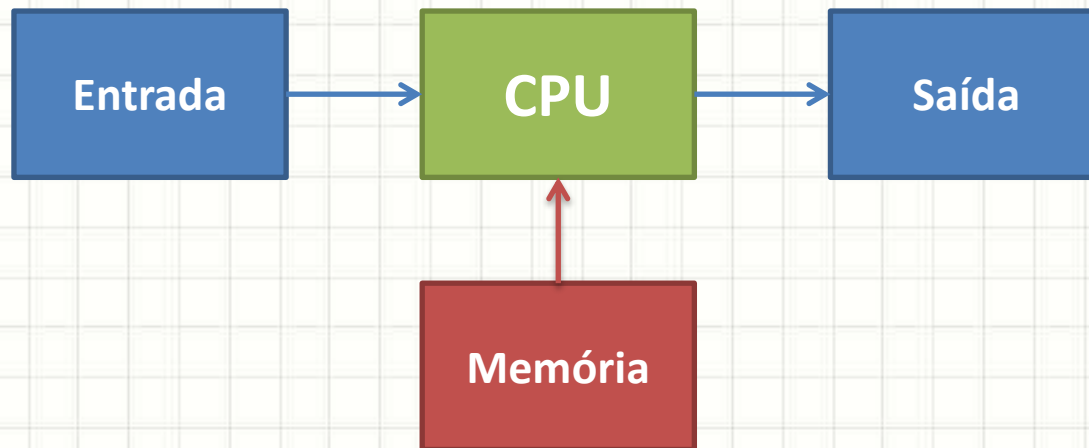
Funcionamento da CPU

- **Cálculos:** Supondo uma instrução de cálculo, a CPU realiza a operação



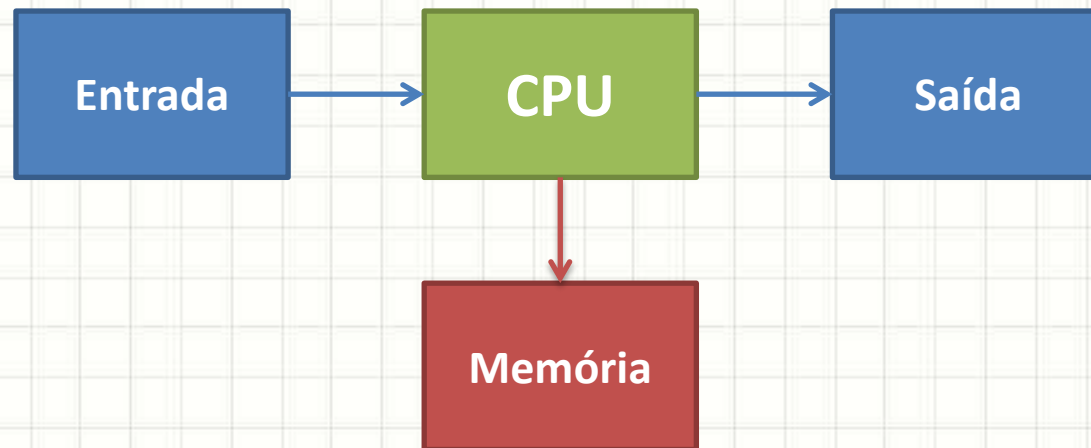
Funcionamento da CPU

- **Busca Instrução:** CPU lê a memória em busca do que deve fazer



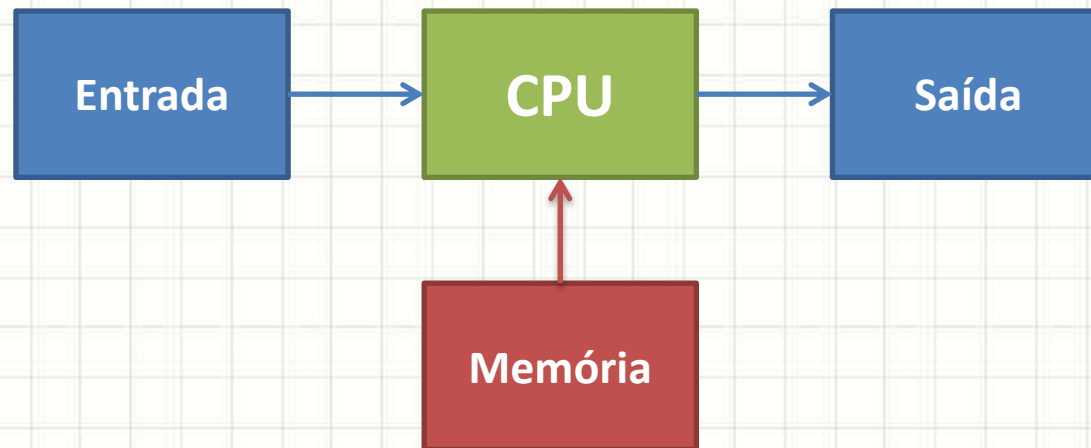
Funcionamento da CPU

- **Armazena dado:** Supondo que a instrução solicitava armazenamento do dado na memória, guarda o mesmo na memória



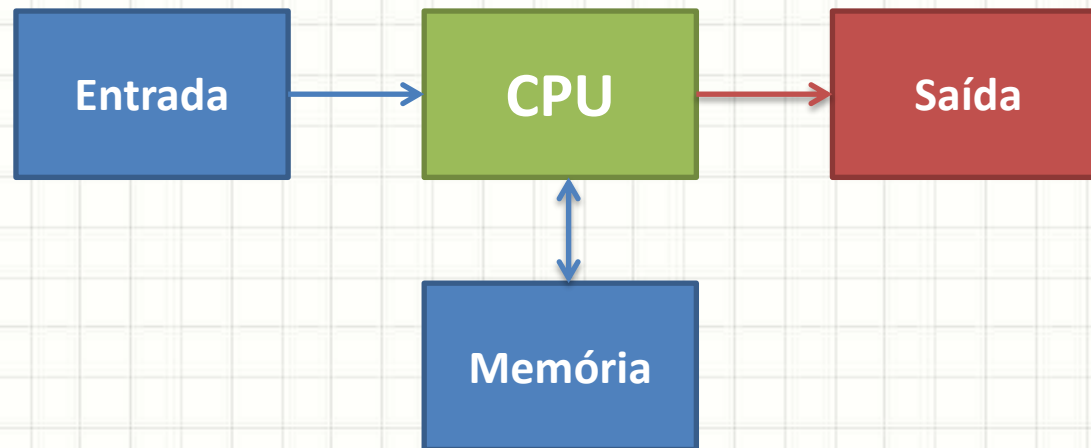
Funcionamento da CPU

- **Busca Instrução:** CPU lê a memória em busca do que deve fazer



Funcionamento da CPU

- **Apresenta saída:** Supondo que a instrução indica que um dado deve ser apresentado ao usuário, CPU escreve na saída



Partes da CPU

- Até agora: CPU como um bloco
- A partir de agora: CPU tem duas partes
 - Unidade Lógica Aritmética (ULA)
 - Unidade de Controle (UC)
- Unidade Lógica Aritmética
 - Processa os cálculos
- Unidade de Controle
 - Controla o fluxo do programa
- Vamos estudar melhor a CPU



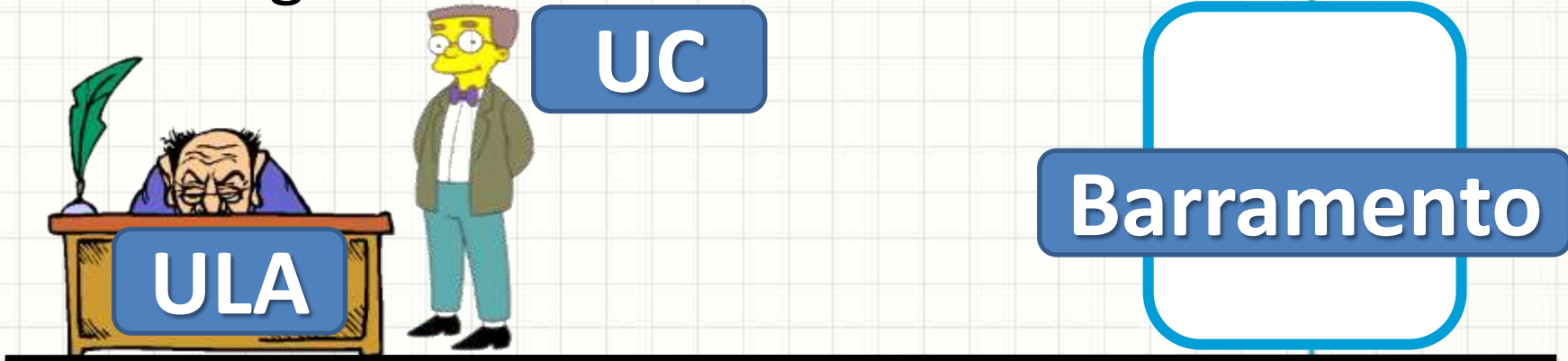
O PROCESSADOR E OS REGISTRADORES

O Processador e os Registradores

- Analogia:
 - Dois funcionários - ULA (tradutor) e UC (auxiliar)
- Objetivo:
 - Traduzir um livro
- Limitações:
 - Funcionários “desmemoriados”
 - Funcionários em um andar
 - Livro a ser traduzido no andar de cima

O Processador e os Registradores

- Analogia: Dois funcionários

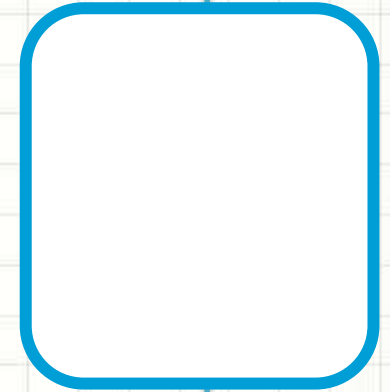


Memória



O Processador e os Registradores

- Instrução? Dois funcionários



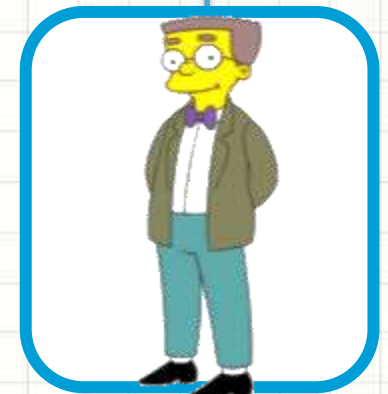
O Processador e os Registradores

- Analogia: Dois funcionários



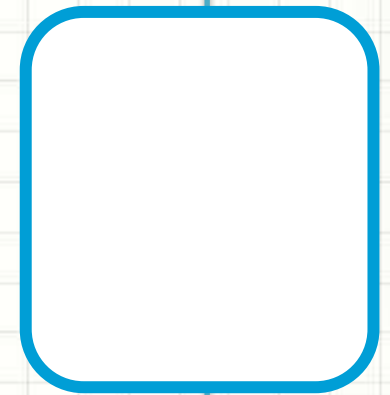
O Processador e os Registradores

- Analogia: Dois funcionários



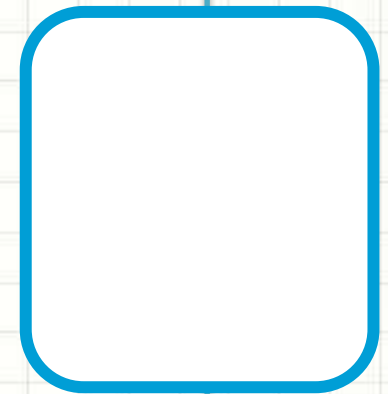
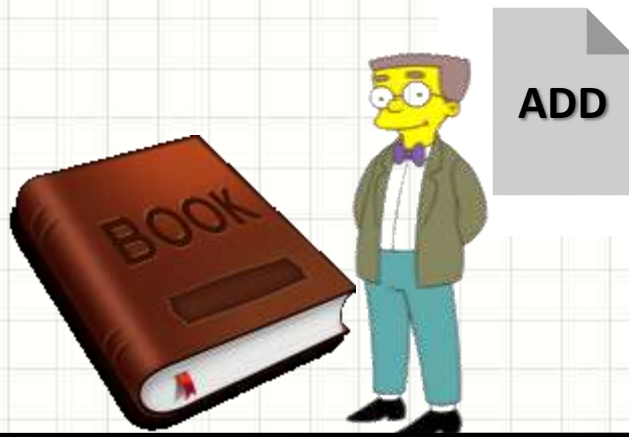
O Processador e os Registradores

- Analogia: Dois funcionários



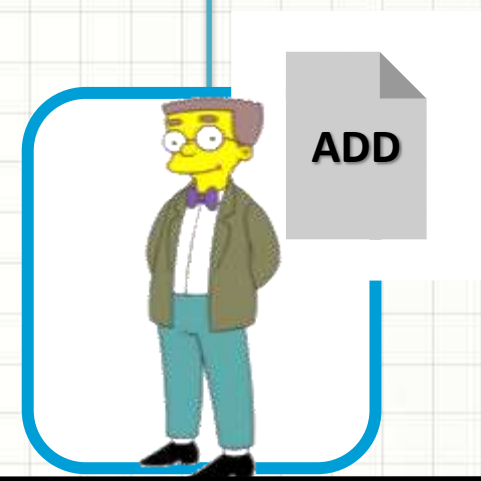
O Processador e os Registradores

- Analogia: Dois funcionários



O Processador e os Registradores

- Analogia: Dois funcionários



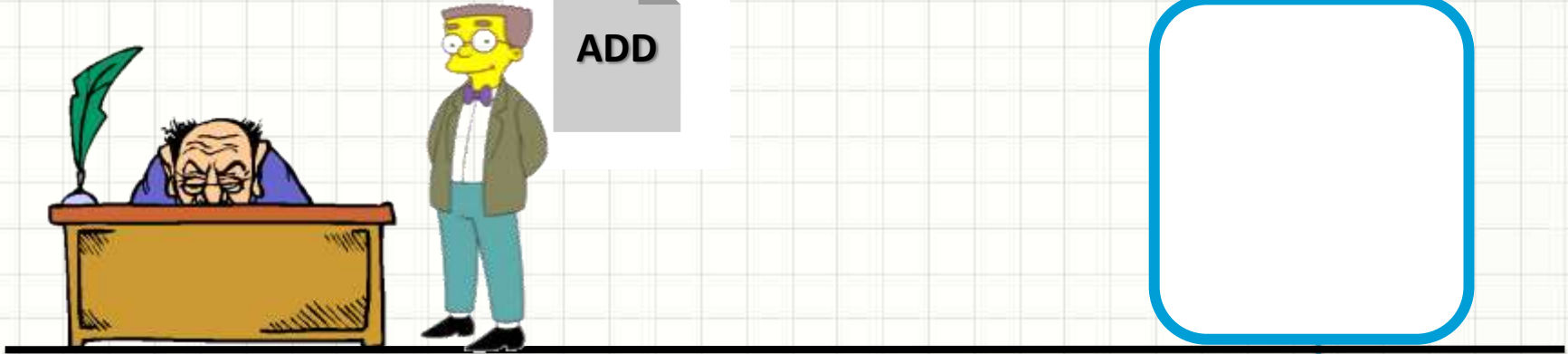
O Processador e os Registradores

- Analogia: Dois funcionários



O Processador e os Registradores

- Analogia: Dois funcionários



O Processador e os Registradores

- An ADD Dois funcionários



O Processador e os Registradores

- An Dois funcionários



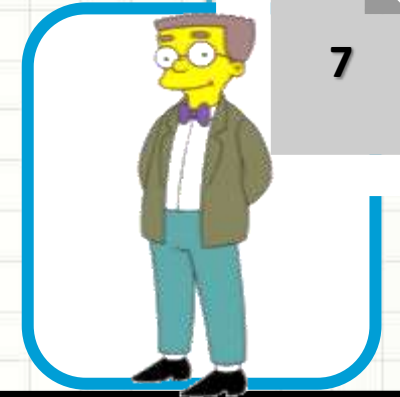
O Processador e os Registradores

- Analogia: Dois  onários



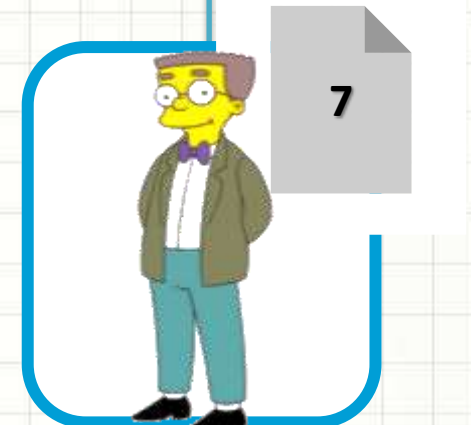
O Processador e os Registradores

- Analogia: Dois funcionários



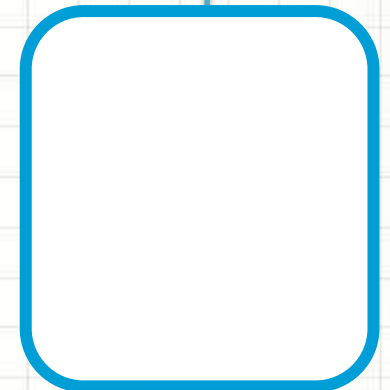
O Processador e os Registradores

- Analogia: Dois funcionários



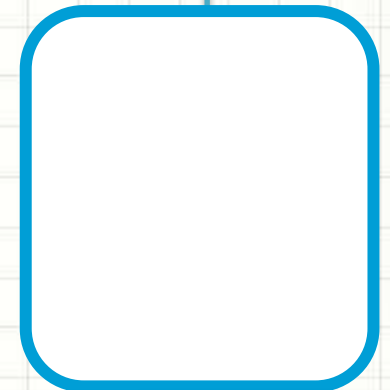
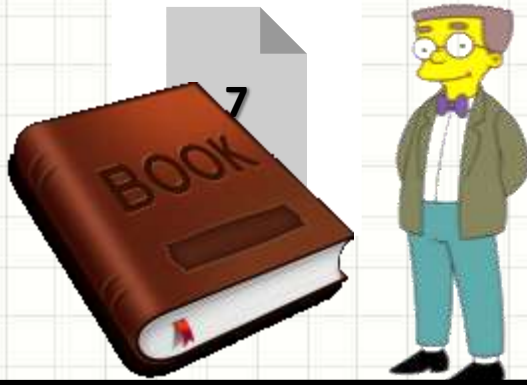
O Processador e os Registradores

- Analogia: Dois funcionários



O Processador e os Registradores

- Analogia: Dois funcionários

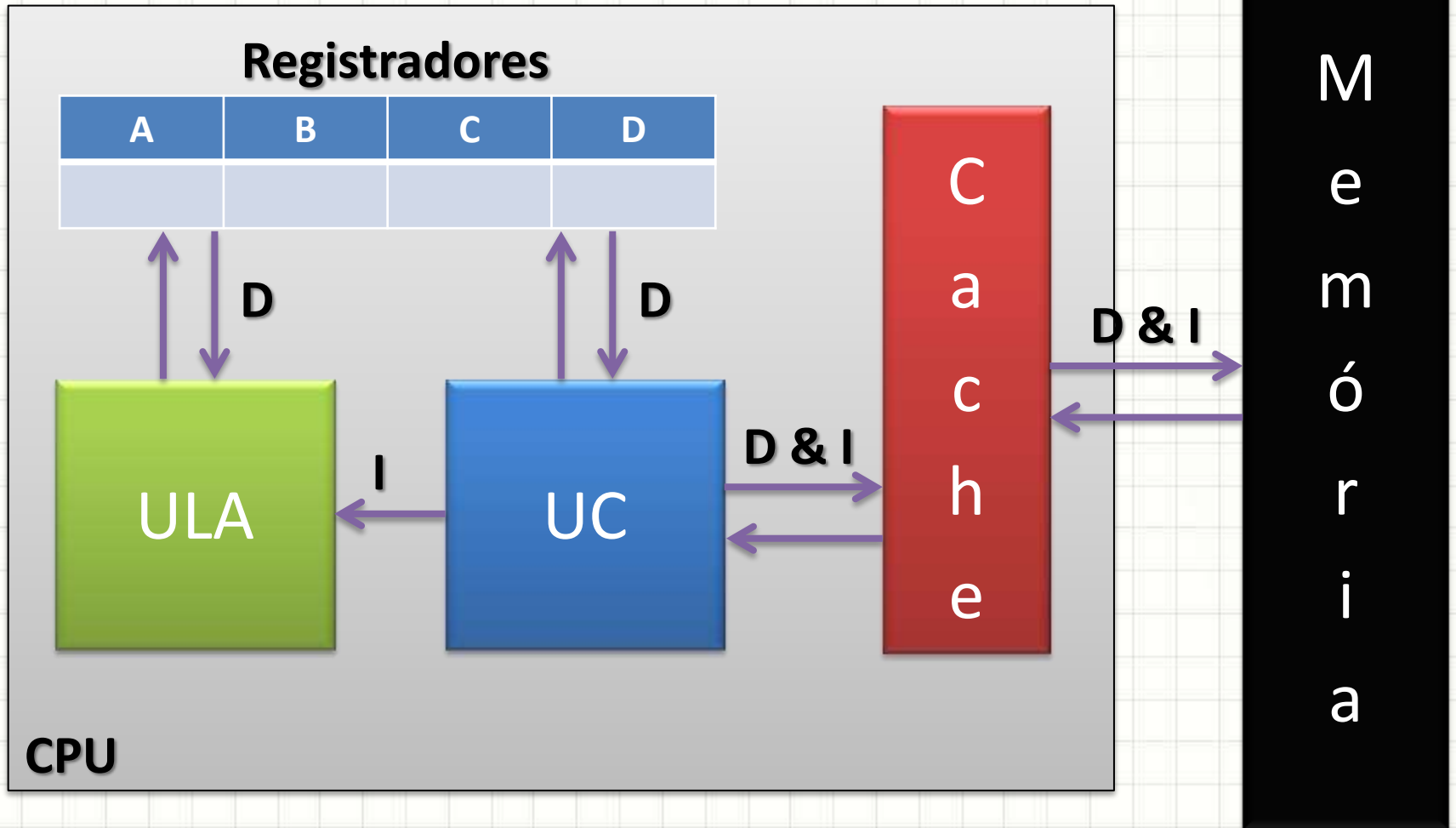


O Processador e os Registradores

- Todos os elementos da analogia existem na prática
 - ULA, UC, Cache... mas e os “papéis”?
- Papéis: registradores
 - Pequenas memórias internas à CPU
- Por que existem registradores?
 - ULA não tem contato com a memória principal...
 - ULA só faz cálculos com valores nos registradores
 - UC é responsável por transferir dados das diversas origens para os registradores

O Processador e os Registradores

- Diagrama

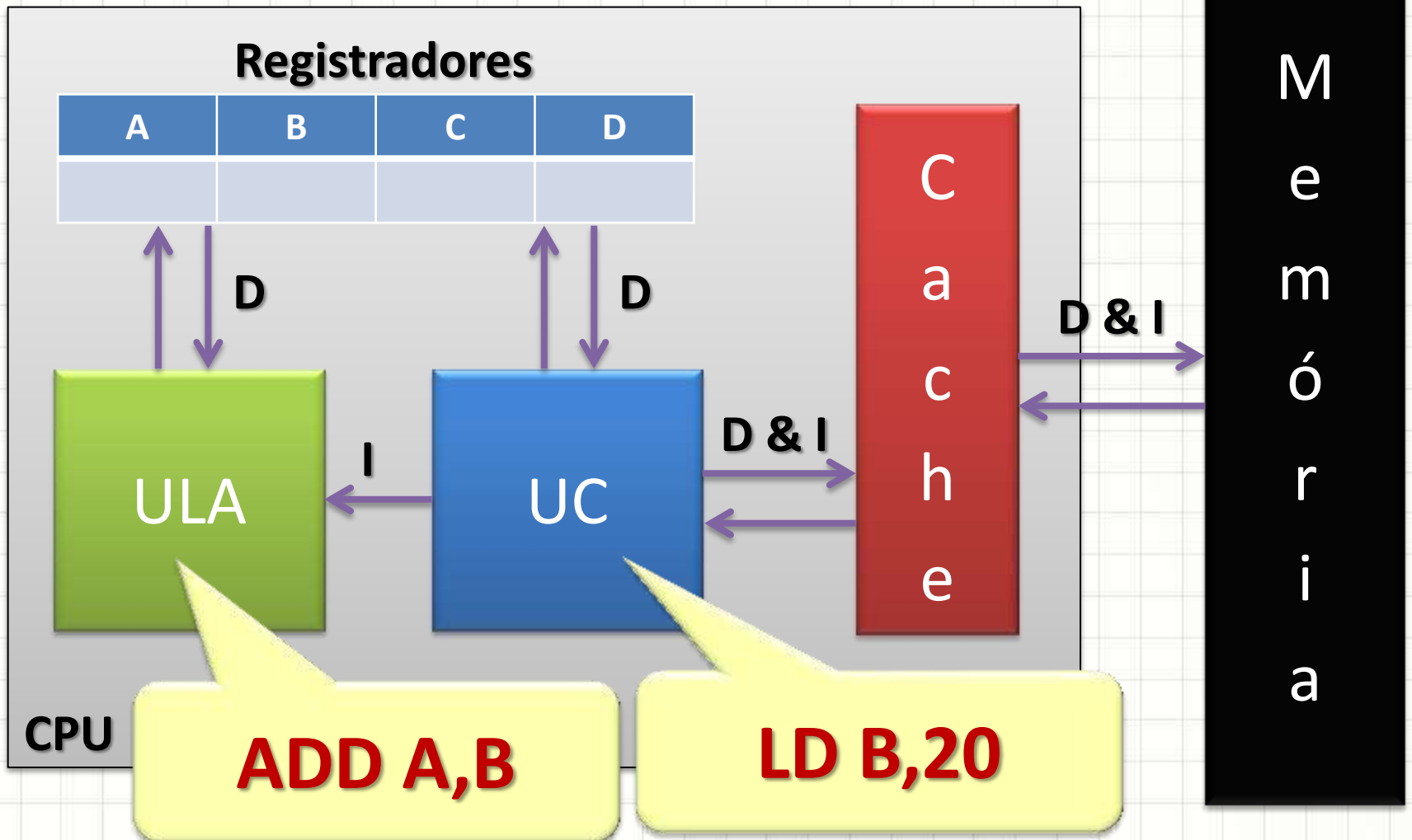


O Processador e os Registradores

- 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)



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

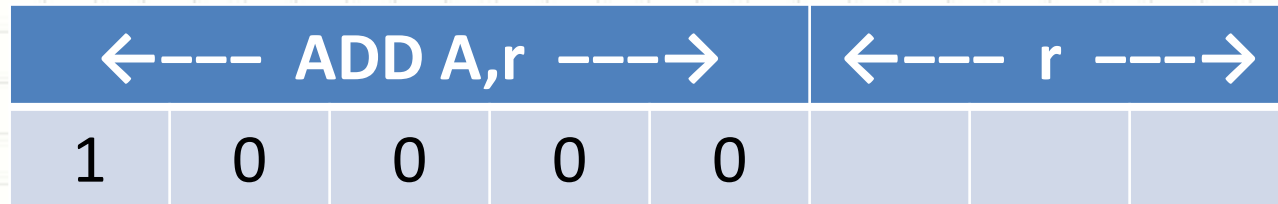
- Transferência de Dados (UC)
 - MOVE, STORE, LOAD, EXCHANGE, PUSH, POP...
- Operações de E/S (UC)
 - READ, WRITE, IN, OUT...
- Operações Aritméticas (ULA)
 - ADD, SUB, MULT, DIV, INC, DEC...
- Operações Lógicas e Conversão (ULA)
 - AND, OR, NOT, XOR, TST, CMP, SHIFT, TRANS, CONV
- Transferência de Controle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

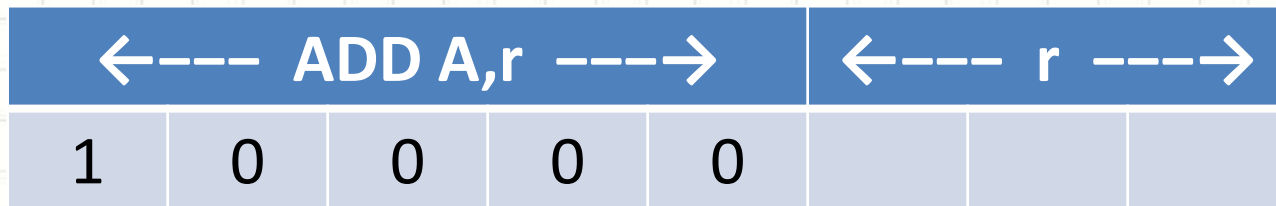


- E o que vai nos bits do “r”?

Como são as Instruções

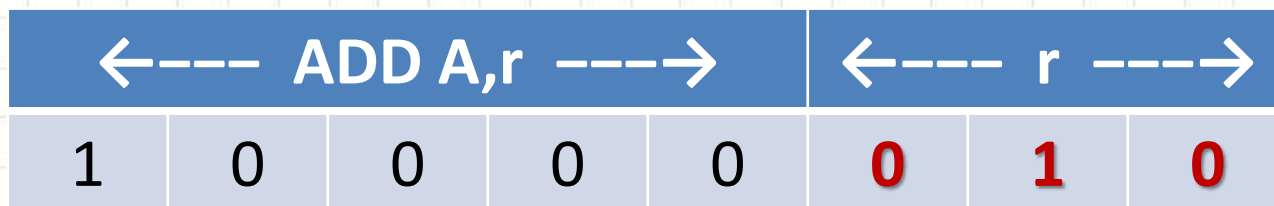
- Exemplo: **ADD A,r**

– **Função:** soma o valor de um registrador **r** em **A**



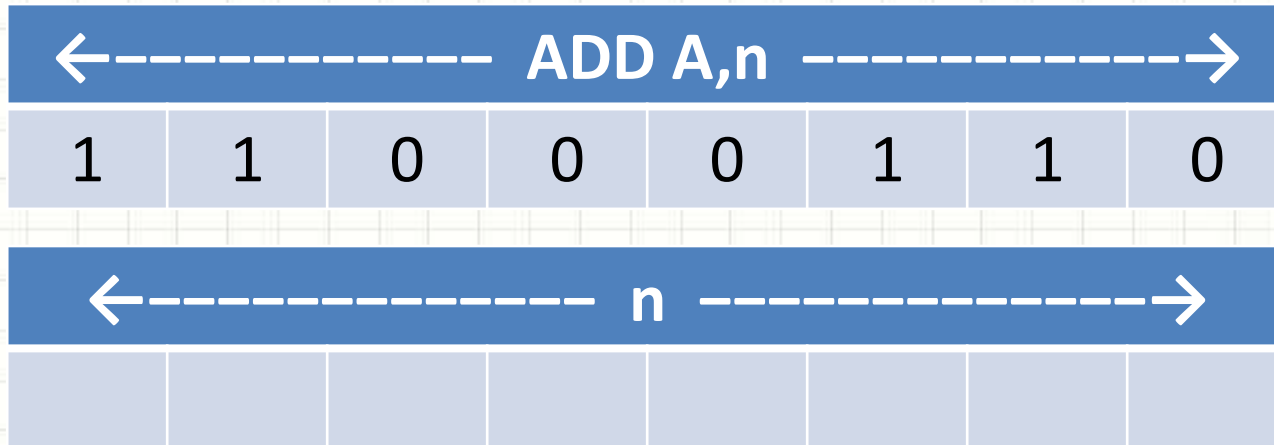
Registrador	A	B	C	D	E	H	L
Bits	111	000	001	010	011	100	101

– Exemplo: **ADD A,D**



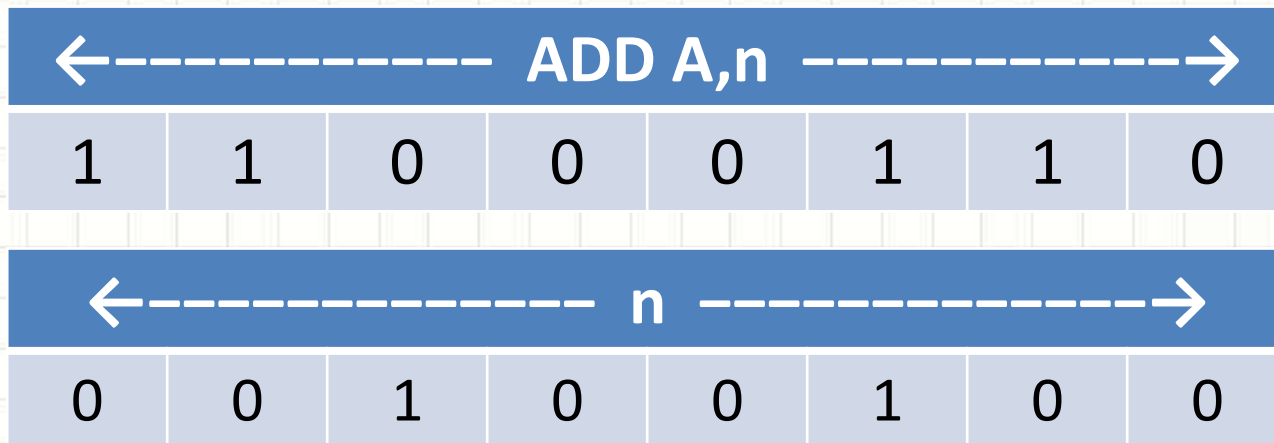
Como são as Instruções

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

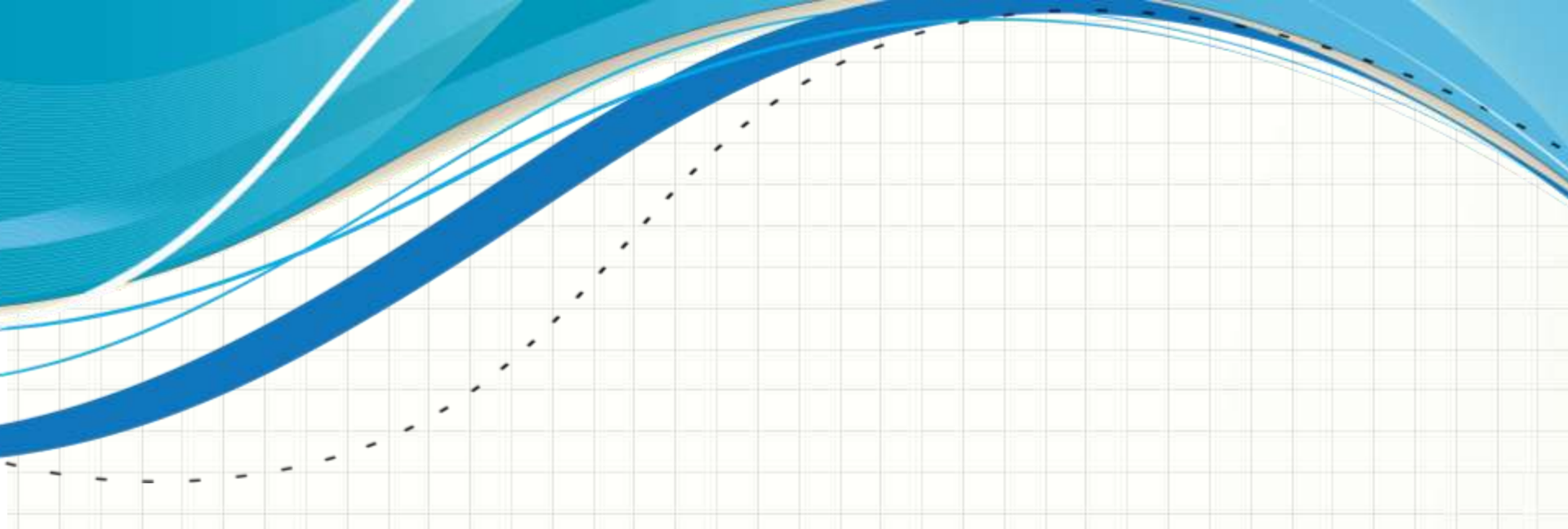


Como são as Instruções

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



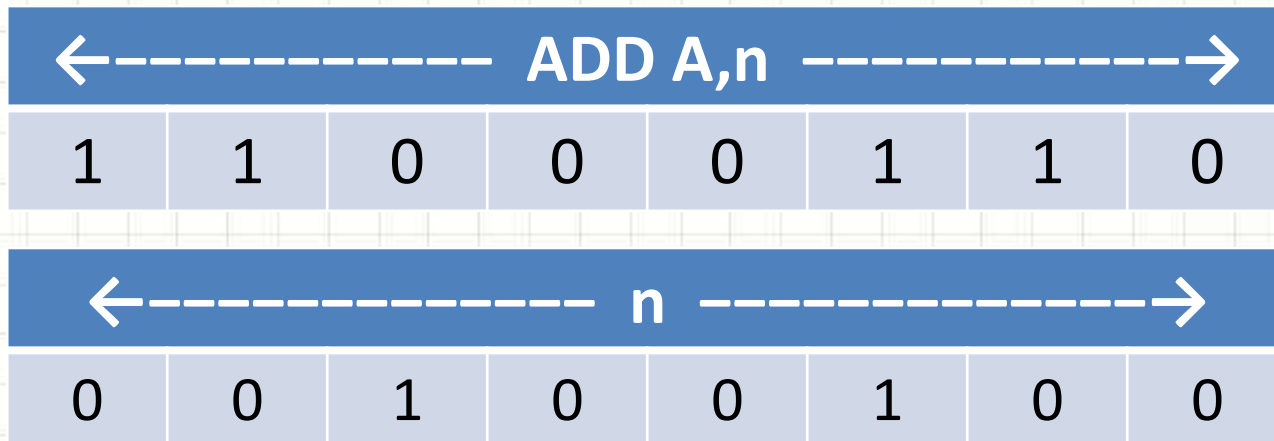
- Na prática, a UC carrega 0x24 em um registrador temporário **X** e solicita que a ULA some **ADD A,X**



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



CONCLUSÕES

Resumo

- A CPU é composta por duas partes: ULA e UC
- Uma CPU tem um determinado **conjunto de instruções**
- As instruções são sequências de bits, mas usualmente damos nomes **mnemônicos** a elas
- A ULA não acessa a memória, quem traz os dados é a UC
- A UC usualmente permite muitas formas de indicar o endereço de um dado a ser lido
- **TAREFA**
 - Lista de Exercícios 2!

Próxima Aula



- Falamos muito da ULA e instruções...
- Mas como a UC realiza sua tarefa?
 - Que tipo de controle ela faz?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**