



ARQUITETURA DE COMPUTADORES

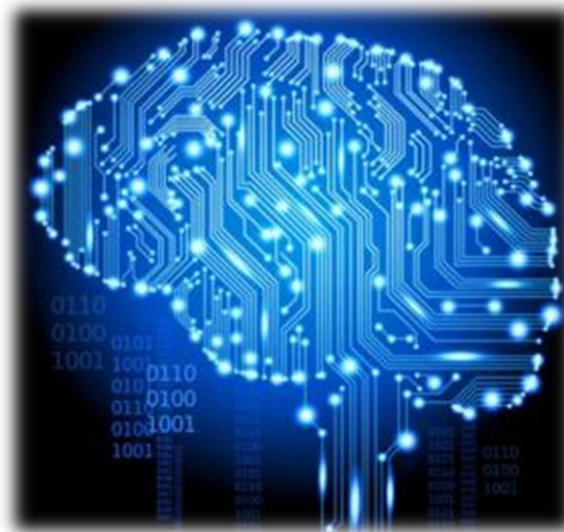
PROCESSAMENTO E INSTRUÇÕES

Prof. Dr. Daniel Caetano

2022 - 1

Compreendendo o problema

- **Situação:** sempre ouvimos que o processador é o “cérebro” do computador. Na verdade, ele apenas executa instruções que lhe foram passadas...



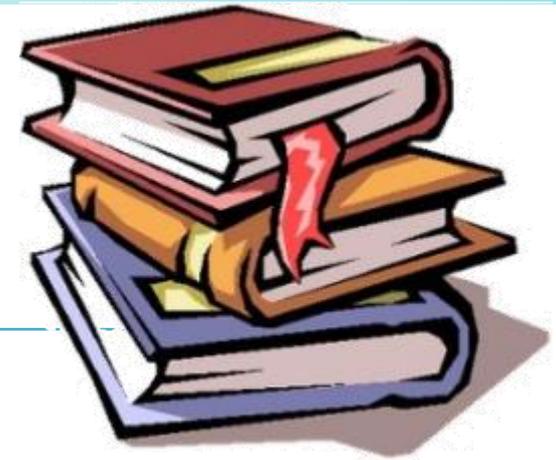
**Mas como ele faz
isso?**

Objetivos



- Compreender o ciclo de instruções de um processador comum
- Conhecer as principais características de um processador
- Compreender o conceito de conjunto de instruções
- Tomar contato com o conceito de compilação

Material de Estudo



Material	Acesso ao Material
Notas de Aula e Apresentação	https://www.caetano.eng.br/aulas/2022a/ara0039.php (Arquitetura de Computadores – Aula 04)
Material Base	<ul style="list-style-type: none">• Introdução à Organização de Computadores (Monteiro, Minha Biblioteca), cap. 6, item 6.2.• Organização de Computadores (Polli, SAVA), cap. 4.
Material Adicional	<ol style="list-style-type: none">1) Saiba o que é o processador: https://www.techtudo.com.br/artigos/noticia/2012/02/o-que-e-processador.html2) Porque o processador é importante: https://www.showmetech.com.br/porque-o-processador-e-uma-peca-importante/3) Conjunto de instruções: https://www.ime.usp.br/~adao/conjuntodeinstrucoes2018f.pdf

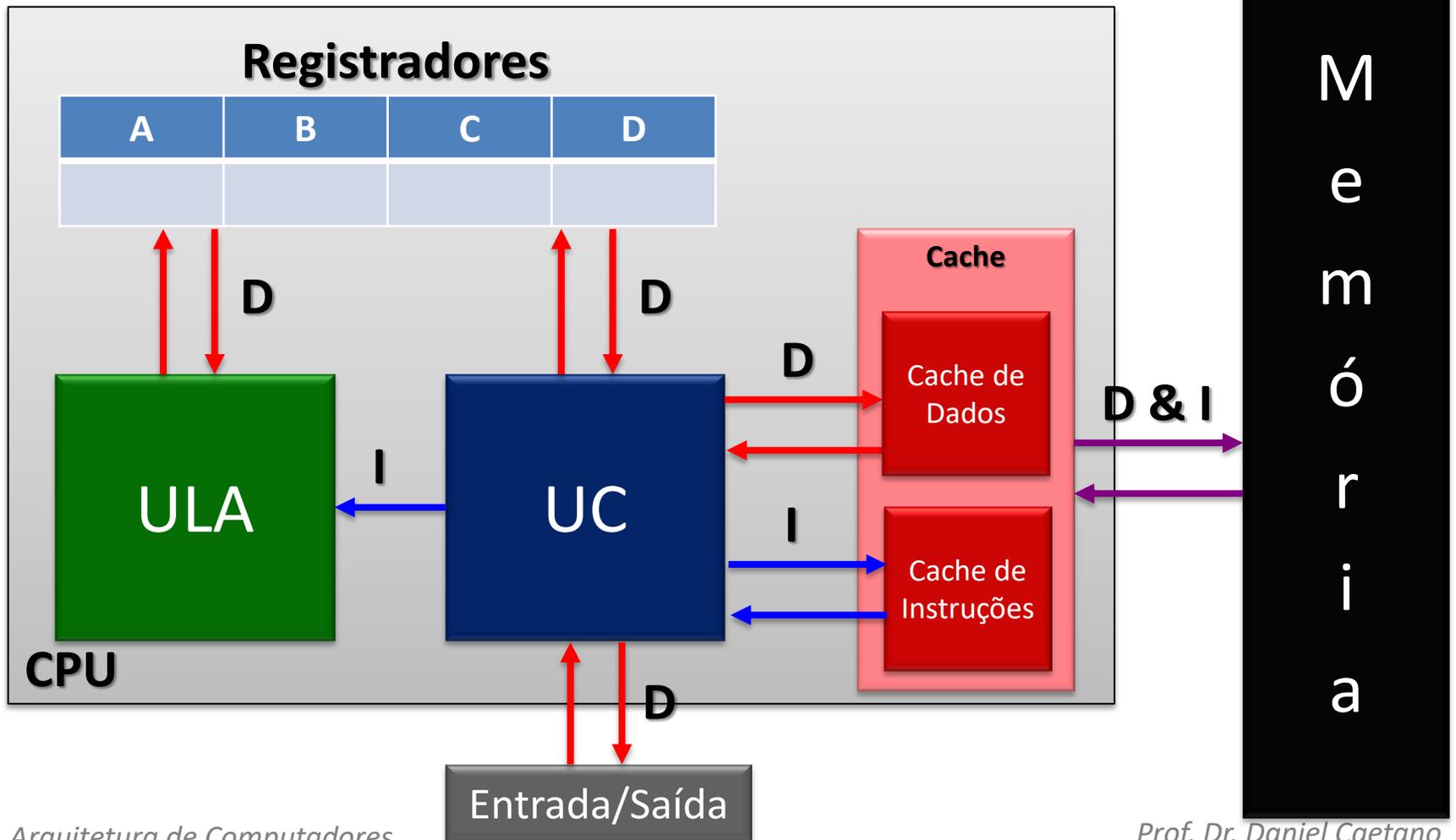


RETOMANDO:

Os “COMPONENTES” DA CPU

Os Componentes da CPU

- Arquitetura Híbrida



Os Componentes da CPU

- ULA: Faz os Cálculos
- UC:
 - Controla a execução do programa (ordem de leitura das instruções)
 - Traz dados da memória e dispositivos para os registradores
 - Comanda a ULA
- Como isso tudo ocorre?



A UNIDADE DE CONTROLE

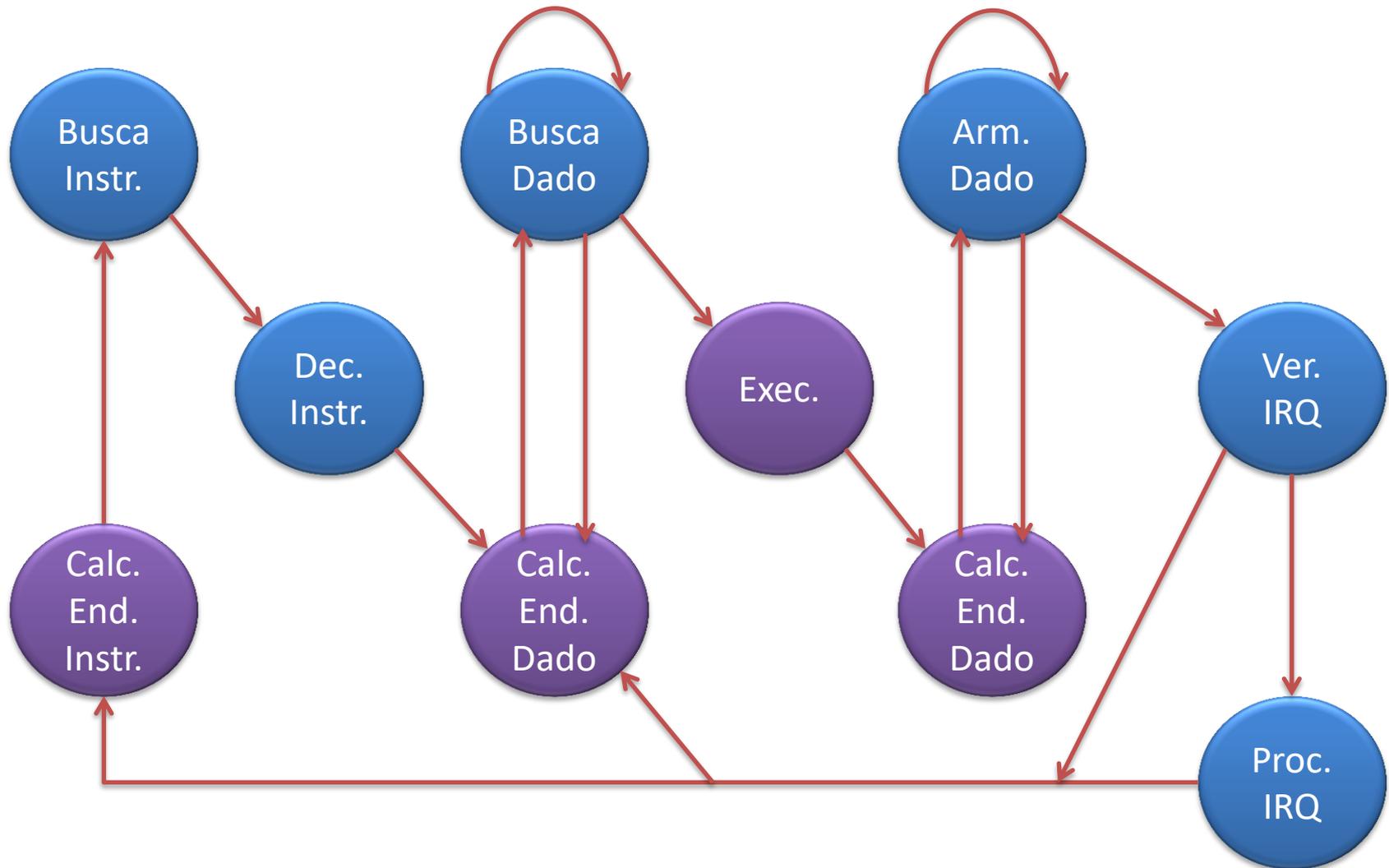
Responsabilidades da UC

- 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

Rotina de Operação da UC

- a) Busca de Instrução
- b) Interpretação da Instrução
- c) Busca dados (se necessário)
- d) Processa dados (se necessário) – **ULA**
- e) Escrita de dados (se necessário)
- f) Avaliação de Interrupções
- g) Execução de Interrupção (se necessário)
- h) Volta para (a)

Diagrama do Ciclo de Instrução





REGISTRADORES ESPECIAIS

Registradores Especiais da ULA

- Quais os registradores da ULA?
 - ULA faz “contas”...
- Acumulador: local onde os resultados são armazenados
- Mas... a ULA só faz essas “contas”?
 - Sim... mas a ULA calcula mais do que o resultado!

Registradores Especiais da ULA

- Ela também atualiza um registrador de “flags”

...	Neg	Carry	Zero	Parity

- Dão informações sobre os resultados
 - Se o resultado é zero, o bit “zero” vira 1
 - Se o resultado superou o limite, “carry” vira 1
 - Se o número de bits 1 é par, “parity” vira 1
 - Etc...

Registadores Especiais da UC

- E a UC?
 - Onde está a próxima instrução?
 - **PC** (Program Counter ou Contador de Programa)
 - Qual instrução está sendo processada?
 - **IR** (Instruction Register ou Registrador de Instrução)
 - Qual endereço sendo lido?
 - **MAR** (Memory Address Register ou Registrador de Endereço de Memória)
 - Qual é o dado sendo lido?
 - **MBR** (Memory Buffer Register ou Registrador de Buffer de Memória)
- **MAR e MBR são ligados aos barramentos**

Outros Registradores

- Registradores de Propósito Geral
 - B, C, D, E... (EBC, ECX, EDX...)
- Registradores de Pilha
 - Armazenamento de Dados (LIFO)
 - **SP** ou **BP** (Stack/Base Pointer: aponta para o topo da pilha)
- Registradores de Índices
 - **IX**, **SI**, **DI** (Index, Source Index, Destination Index)
- Registradores de Segmento (MMU)
 - **CS** (Code Segment ou Segmento de Código)
 - **DS** (Data Segment / Segmento de Dados)
 - **SS** (Stack Segment ou Segmento da Pilha)



A LINGUAGEM DE MÁQUINA

Linguagem de Máquina

- Como indicar instruções para o processador?



Linguagem de Máquina

- Como indicar instruções para o processador?

FIOS



Linguagem de Máquina

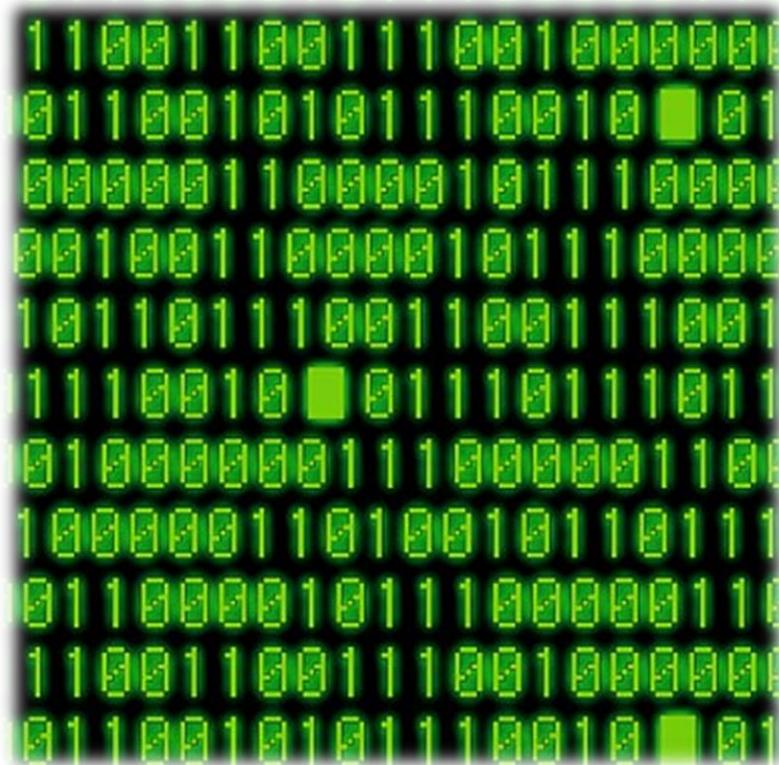
- Como indicar instruções para o processador?

10111000110010110



Linguagem de Máquina

- Como seria um programa em linguagem de máquina?





LINGUAGEM ASSEMBLY

Linguagem Assembly

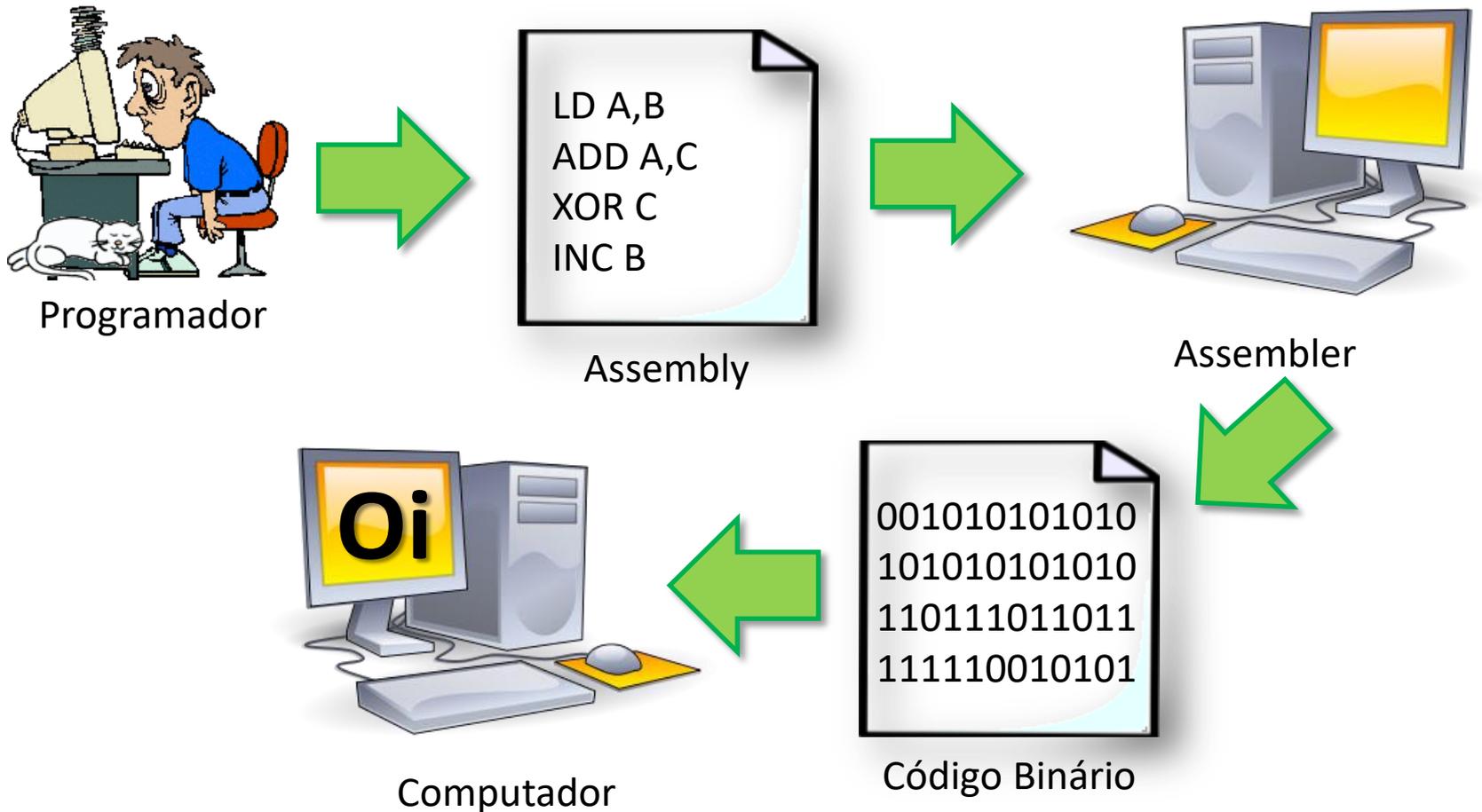
- Ninguém percebeu que LM era horrível?
 - Não demorou!
- Qual a solução?

Op1	Mnemonic
00000000	NOP
00000001	LD BC,nnmm
00000010	LD (BC),A
00000011	INC BC
00000100	INC B
00000101	DEC B
00000110	LD B,mm
00000111	RLCA

Linguagem Assembly

- Linguagem Assembly: Instruções Mnemônicas
- E quem converte?
 - No início, manualmente
 - Muito rapidamente: **assemblers**
- Qual o papel dos “assemblers”?

O que faz o Assembler?





INSTRUÇÕES DA ULA E DA UC

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)

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**



CODIFICAÇÃO DAS 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

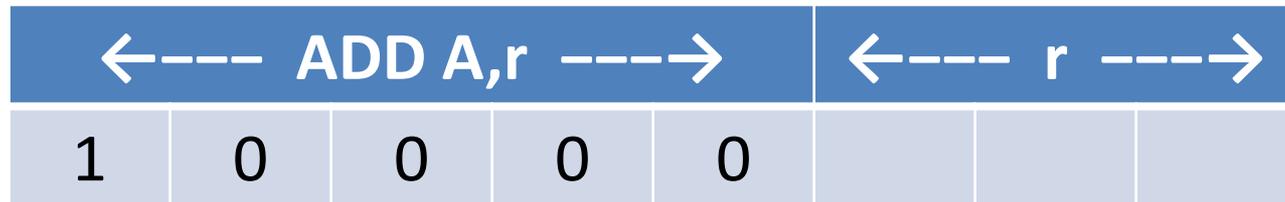
- Transferência de Dados (UC)
 - MOVE, STORE, LOAD, EXCHANGE, PUSH, POP...
- Operações de E/S (UC)
 - READ, WRITE, IN, OUT...
- Operações Aritméticas (ULA)
 - ADD, SUB, MULT, DIV, INC, DEC...
- Operações Lógicas e Conversão (ULA)
 - AND, OR, NOT, XOR, TST, CMP, SHIFT, TRANS, CONV
- Transferência de Controle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

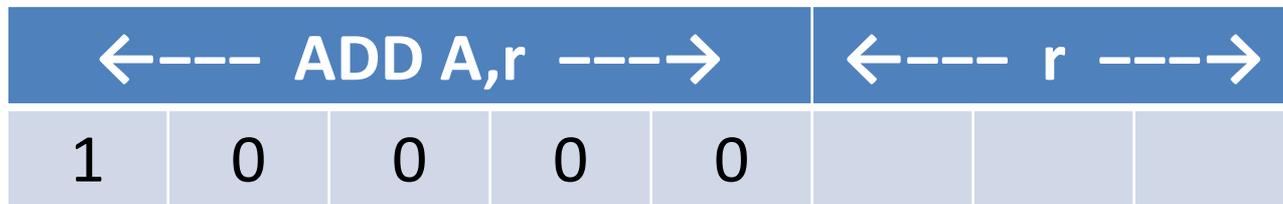
- Obviamente não são letras na memória
- Cada instrução: um conjunto de bits
- Exemplo: **ADD A,r**
 - **Função:** soma o valor de um registrador **r** em **A**



- E o que vai nos bits do “r”?

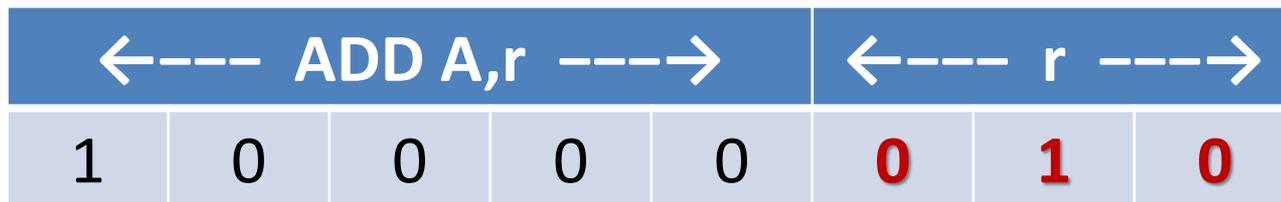
Como são as Instruções

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



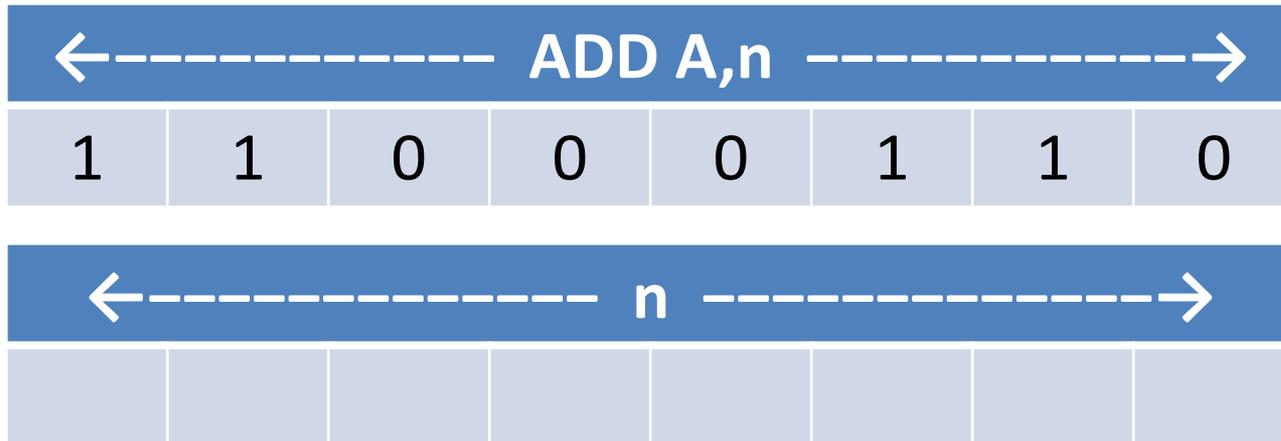
Registrador	A	B	C	D	E	H	L
Bits	111	000	001	010	011	100	101

- Exemplo: **ADD A,D**



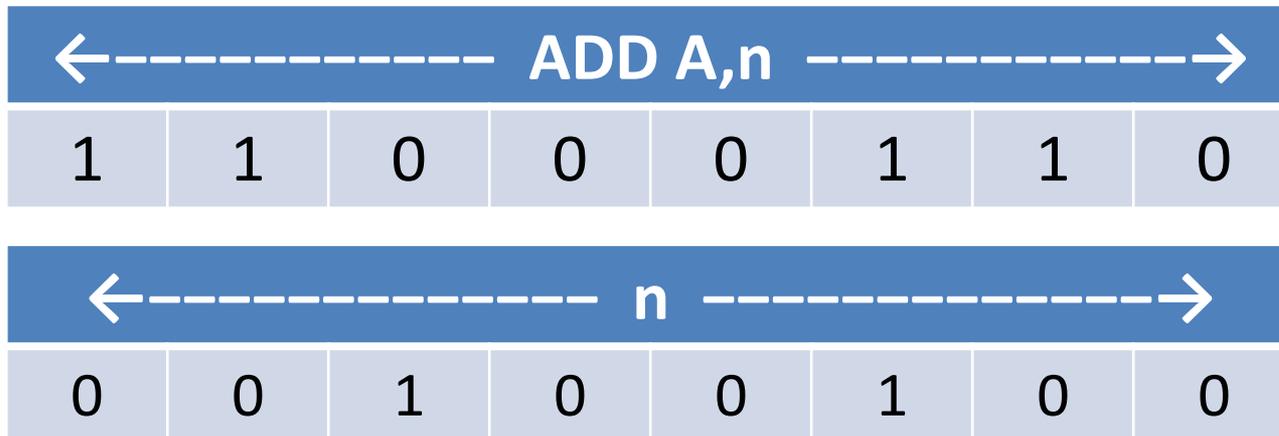
Como são as Instruções

- 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



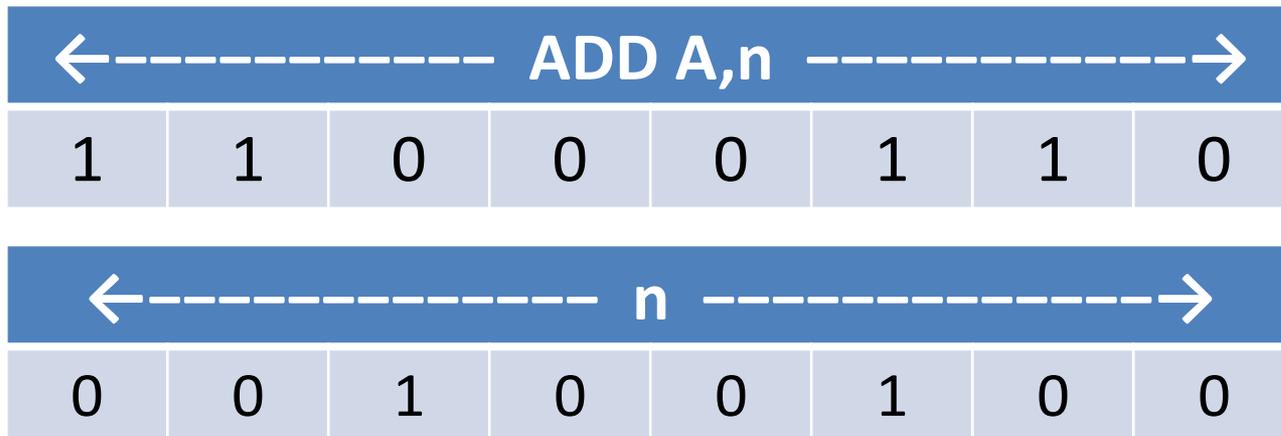
- Na prática, a UC carrega 0x24 em um registrador temporário **X** e solicita que a ULA some **ADD A,X**



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



LINGUAGENS DE PROGRAMAÇÃO DE ALTO NÍVEL

Por que não usar Assembly?

- Muito burocrático!
- Exige compreensão do hardware
- Instruções muito “simples”
- Difícil de ler e de escrever!

LD	A,16
LD	B,32
ADD	A,B
SRA	A

$$A = (16 + 32)/2$$

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- FORTRAN

```
PROGRAM TRIVIAL
  INTEGER I
  I=2
  IF(I .GE. 2) CALL PRINTIT
  STOP
END
SUBROUTINE PRINTIT
  PRINT *, 'Hola Mundo'
  RETURN
END
```

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- COBOL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. HELLO-WORLD.  
PROCEDURE DIVISION.  
    DISPLAY 'Hello, world'.  
    STOP RUN.
```

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- LISP

```
;;; Hello World in Common Lisp

(defun helloworld ()
  (print "Hello World!")
)
```

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- C/C++

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- BASIC

```
10 PRINT "HELLO WORLD"
```

Linguagens de Alto Nível

- Linguagens mais próximas da humana
- Pascal

```
PROGRAM hello;  
USES CRT;  
BEGIN  
  WriteLn('Hello World!');  
END.
```

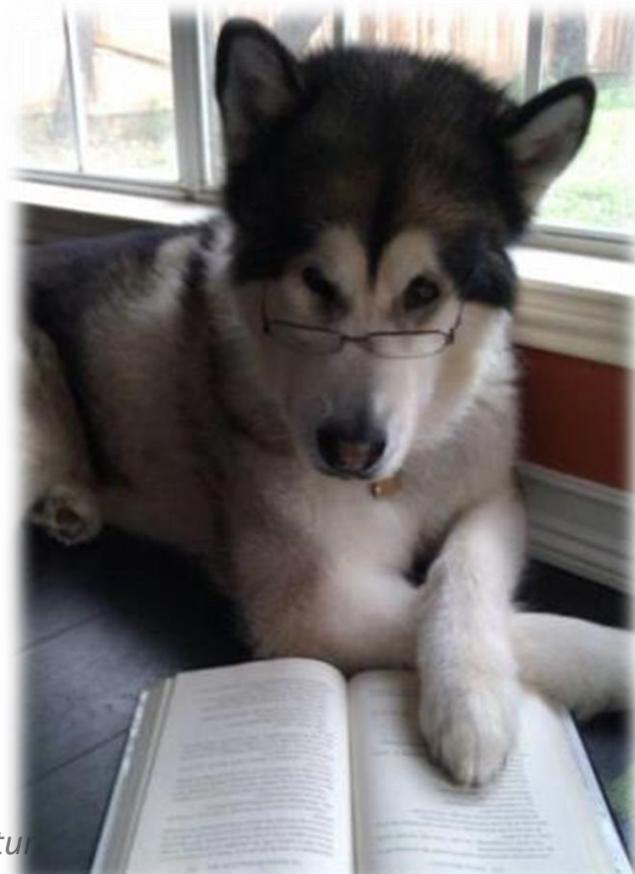
Linguagens de Alto Nível

- Linguagens mais próximas da humana
- Java

```
public class Hello {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello");  
  
    }  
  
}
```

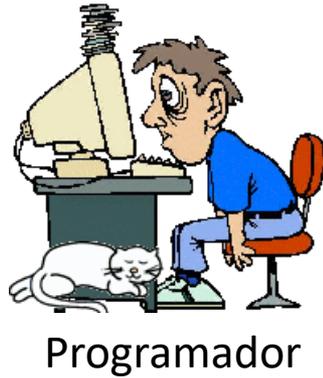
Linguagens de Alto Nível

- Cada uma tem vantagens e desvantagens
- Todas: **incompreensíveis** para o computador



HUM!?

Processo de Compilação

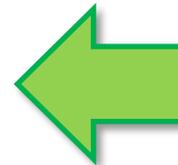


```
#include <io...  
int main(void)  
{  
    cout << "Oi";  
}
```

Código Fonte



Compilador



```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Objeto
(Binário)

Processo de Compilação

```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Objeto 1
(Binário)

```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Objeto 2
(Binário)



Linker



```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Binário
Executável



CARACTERIZAÇÃO DE UM PROCESSADOR

Como Caracterizar o Processador?

- Existem 4 características fundamentais
 - Tipo de soquete
 - Identificar compatibilidade com placa mãe1
 - Núcleos/Cores/Threads
 - Identificar o número de processos simultâneos
 - Frequência/Clock
 - Identificar a velocidade do processador
 - Identificar a compatibilidade com placa mãe
 - Identificar a compatibilidade com a memória
 - Cache
 - Identificar o tamanho da memória de trabalho.

Como Caracterizar o Processador

	Core i3-4160	Core i3-4350	Core i5-4690	Core i7-4790
Price	\$120	\$155	\$225	\$310
Cores	2	2	4	4
Threads	4	4	4	8
Frequency	3.6 GHz	3.6 GHz	3.5 GHz	3.6 GHz
Turbo	-	-	3.9 GHz	4 GHz
L3 cache	3 MB	4 MB	6 MB	8 MB
GPU model	HD Graphics 4400	HD Graphics 4600	HD Graphics 4600	HD Graphics 4600
GPU frequency	350–1150 MHz	350–1150 MHz	350–1200 MHz	350–1200 MHz
TDP	54 W	54 W	84 W	84 W

Como Caracterizar o Processador



	AMD Ryzen 9 3900X	AMD Ryzen 9 3950X	Intel Core i9-9900K	Intel Core i9-10900K
Clock Speed and Cores (Turbo)	12 x 3.8GHz (4.6GHz)	16 x 3.5GHz (4.7GHz)	8 x 3.6GHz (5.0GHz)	10 x 3.7GHz (5.3GHz)
CPU Threads	24	32	16	20
RAM Speed	3200MHz	3200MHz	2666MHz	2933MHz
L2 Cache	6MB	8MB	2MB	2MB
L3 Cache	64MB	64MB	16MB	20MB
Thermal Design	105W	105W	95W	125W
Integrated Graphics	No	No	Yes	Yes
Unlocked Multiplier	Yes	Yes	Yes	Yes



ATIVIDADE

Atividade

- Em grupo:
 1. Escolha um processador atual da Intel, procure suas especificações.
 2. Busque entre os processadores atuais na AMD aquele que mais se aproxima das características do processador Intel escolhido.
 3. Busque o preço de ambos os processadores no mercado e defina qual processador você compraria, justificando a escolha.



ENCERRAMENTO

Resumo e Próximos Passos

- O ciclo de instrução completo
 - Conjunto de Instruções e sua representação
 - Função do compilador
 - Principais características de uma CPU.
 - **Pós Aula:** Saiba Mais, A Seguir e Desafio!
 - No mural: <https://padlet.com/djcaetano/arquitetura/>
-
- Aprofundando o funcionamento...
 - Entrada e saída? Barramento?



PERGUNTAS?