

# Aula 10 – O conjunto de instruções

Prof. João Fernando Mari  
*joaof.mari@ufv.br*

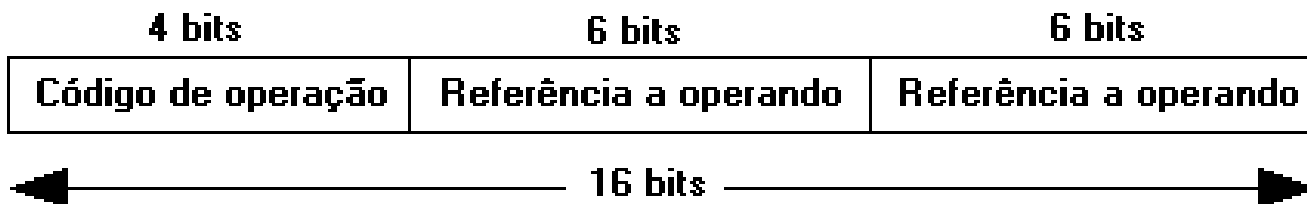
- O conjunto de instruções
- Representação de Instruções
- Representação de Instruções – Quantidade de endereços
- EXEMPLO: Linguagem de Montagem (*assembly*)

# O conjunto de instruções

- As instruções de máquina (do computador) determinam a operação que a CPU deve executar;
- A coleção desses instruções é conhecida como **conjunto de instruções** da CPU;
- Cada instrução deve conter todos os dados necessários para que a CPU possa executá-la.

# Representação de Instruções

- Cada instrução contém:
  - Código de operação;
  - Referência a operando fonte;
  - Referência a operando destino;
  - Endereço da próxima instrução.
    - Implícito pelo Contador de Programas, exceto para instruções de desvio.
- Cada arquitetura possui uma organização específica para as suas instruções.
  - EXEMPLO de uma instrução de 16 bits e duas referencias a operando.



# Representação de Instruções

- Processamento de dados:
  - ADD            → Adição
  - SUB            → Subtração
  - MPY           → Multiplicação
  - DIV            → Divisão
  - ...
  
- Movimentação de dados:
  - LOAD           → Carregar dados da memória (ou LDA)
  - STOR           → Armazenar dados na memória (ou STA)
  - ...
  
- Desvio:
  - *(necessário para as instruções de alto nível de desvio condicional (if) e loops (for, while).*
  - J → Jump (Salto incondicional)
  - ...

- Número de referencias à operandos
  - **Operação de Soma (ADD):**
  - Referencia a 1 operando:
    - ADD X
      - X: Operando de origem e destino
      - Soma o valor armazenado no registrador X ao valor atual do registrador ACUMULADOR. Armazena o resultado em X.
  - Referencia a 2 operandos:
    - ADD X, Y
      - X e Y: Operando de origem; X: Operandos de destino
      - Soma o valor ATUAL do registrador X com o valor armazenado no registrador Y. Armazena o resultado no registrador X
  - Referencia a 3 operandos:
    - ADD, X, Y, Z
      - X: Operando de destino; Y e Z: Operandos de origem.
      - Soma os valores armazenados nos registradores Y e Z e armazena o resultado em no registrador X.

# Representação de Instruções – Quantidade de endereços

- Instrução de alto-nível:
  - $Y = (A - B) / (C + D * E);$

1 operando		2 operandos		3 operandos	
LOAD D	# $AC \leftarrow D$	MOVE Y, A	# $Y \leftarrow A$	SUB Y, A, B	# $Y \leftarrow A - B$
MPY E	# $AC \leftarrow AC * E$	SUB Y, B	# $Y \leftarrow Y - B$	MPY T, D, E	# $T \leftarrow D * E$
ADD C	# $AC \leftarrow AC + C$	MOVE T, D	# $T \leftarrow D$	ADD T, T, C	# $T \leftarrow T + C$
STOR Y	# $Y \leftarrow AC$	MPY T, E	# $T \leftarrow T * E$	DIV Y, Y, T	# $Y \leftarrow Y / T$
LOAD A	# $AC \leftarrow A$	ADD T, C	# $T \leftarrow T + C$		
SUB B	# $AC \leftarrow AC - B$	DIV Y, T	# $Y \leftarrow Y / T$		
DIV Y	# $AC \leftarrow AC / Y$				
STOR Y	# $Y \leftarrow AC$				

# EXEMPLO: Linguagem de Montagem (*assembly*)

- Instrução em alto-nível:  $N = I + J + K$ ;
  - Sendo  $I=2$ ,  $J=3$ ,  $K=4$
  - As variáveis de alto-nível  $I$ ,  $J$ ,  $K$  e  $N$  fazem referencia às posições de memória:
    - 201, 202, 203 e 204, respectivamente.
  - O valor armazenado em cada uma das posições da memória:
    - Posição 201 = 2;
    - Posição 202 = 3
    - Posição 203 = 4
    - Posição 204 = 0
  - Como ocorre o processamento?
    - 1 - Carrega o conteúdo da posição de endereço 201 em AC;
      - 2
    - 2 - Adiciona o conteúdo da posição 202 a AC;
      - $2 + 3 = 5$
    - 3 - Adiciona o conteúdo da posição de memória 203 a AC;
      - $5 + 4 = 9$
    - 4 - Armazena o conteúdo de AC na posição de endereço 204.
      - $N = 9$

Memória	
0	????
...	...
201	2
202	3
203	4
204	0
...	...
N-1	????

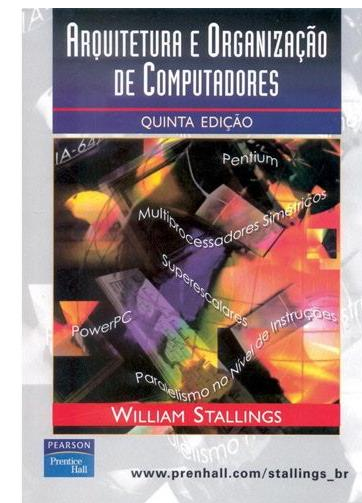


# EXEMPLO: Linguagem de Montagem (*assembly*)

Programa em Assembly			Programa Simbólico			Programa em Hexadecimal		Programa em binário	
Rotulo	Operação	Operando	Memória	Operação	Operando	Memória	Operador e Operando	Memória	Operador e Operando
FORM:	LDA	I	101	LDA	201	101	2201	101	0010 0010 0000 0001
	ADD	J	102	ADD	202	102	1202	102	0001 0010 0000 0010
	ADD	K	103	ADD	203	103	1203	103	0001 0010 0000 0011
	STA	N	104	STA	204	104	3204	104	0011 0010 0000 0100
...	...	...	...	...	...	...	...	...	...
I	DAT	2	201	DAT	2	201	0002	201	0000 0000 0000 0010
J	DAT	3	202	DAT	3	202	0003	202	0000 0000 0000 0011
K	DAT	4	203	DAT	4	203	0004	203	0000 0000 0000 0100
N	DAT	0	204	DAT	0	204	0000	204	0000 0000 0000 0000

# Referências

- STALLINGS, W. **Arquitetura e Organização de Computadores**, 8. Ed., Pearson, 2010.
  - Seções 10.1 e 10.2
- STALLINGS, W. **Arquitetura e Organização de Computadores**, 5. Ed., Pearson, 2003.
  - Seções 9.1 e 9.2



# FIM