

Computador Hipotético Neander

Disciplina: Introdução à Arquitetura de Computadores

Luciano Moraes Da Luz Brum

Universidade Federal do Pampa – Unipampa – Campus Bagé

Email: lucianobrum18@gmail.com

➤ Largura de dados e endereços de 8 bits;

- Largura de dados e endereços de 8 bits;
- Dados representados em complemento de dois;

- Largura de dados e endereços de 8 bits;
- Dados representados em complemento de dois;
- 1 acumulador de 8 bits (AC);

- Largura de dados e endereços de 8 bits;
- Dados representados em complemento de dois;
- 1 acumulador de 8 bits (AC);
- 1 contador de programa de 8 bits (PC);

- Largura de dados e endereços de 8 bits;
- Dados representados em complemento de dois;
- 1 acumulador de 8 bits (AC);
- 1 contador de programa de 8 bits (PC);
- 1 registrador de estado com 2 códigos de condição: negativo (N) e zero (Z);

➤ Modos de endereçamento: Direto.

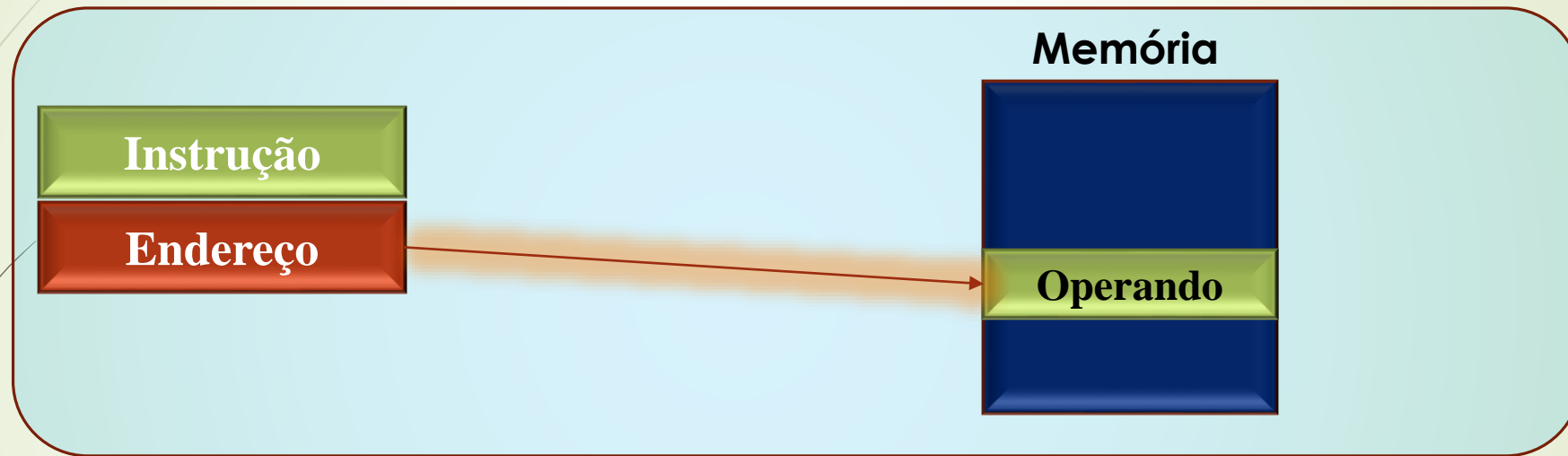


Figura 1: Endereçamento Direto. Fonte: Adaptado de Weber, 2001.

- Obs: em instruções de desvio, o endereço contido na instrução é a posição de memória onde está a instrução a ser executada.

ISA (*Instruction Set Architecture*)

Código	Instrução	Comentário
0000	NOP	nenhuma operação
0001	STA end	armazena acumulador - (store)
0010	LDA end	carrega acumulador - (load)
0011	ADD end	soma
0100	OR end	“ou” lógico
0101	AND end	“e” lógico
0110	NOT	inverte (complementa) acumulador
1000	JMP end	desvio incondicional - (jump)
1001	JN end	desvio condicional - (jump on negative)
1010	JZ end	desvio condicional - (jump on zero)
1111	HLT	término de execução - (halt)

Tabela 1: Conjunto de instruções do Neander. Fonte: Weber, 2001.

ISA (*Instruction Set Architecture*)

Instrução	Comentário
NOP	nenhuma operação
STA end	$\text{MEM}(\text{end}) \leftarrow \text{AC}$
LDA end	$\text{AC} \leftarrow \text{MEM}(\text{end})$
ADD end	$\text{AC} \leftarrow \text{MEM}(\text{end}) + \text{AC}$
OR end	$\text{AC} \leftarrow \text{MEM}(\text{end}) \text{ OR } \text{AC}$
AND end	$\text{AC} \leftarrow \text{MEM}(\text{end}) \text{ AND } \text{AC}$
NOT	$\text{AC} \leftarrow \text{NOT } \text{AC}$
JMP end	$\text{PC} \leftarrow \text{end}$
JN end	IF $\text{N}=1$ THEN $\text{PC} \leftarrow \text{end}$
JZ end	IF $\text{Z}=1$ THEN $\text{PC} \leftarrow \text{end}$

Tabela 2: Ações das instruções. Fonte: Weber, 2001.

- **Códigos de Condição:** Sinais gerados pela ULA que notificam a UC sobre determinado evento durante a realização de uma operação.

- Códigos de Condição: Sinais gerados pela ULA que notificam a UC sobre determinado evento durante a realização de uma operação.
- Utilizado pelas instruções: JN e JZ;

- **Códigos de Condição:** Sinais gerados pela ULA que notificam a UC sobre determinado evento durante a realização de uma operação.
- Utilizado pelas instruções: JN e JZ;
- N - negativo: sinal do resultado (1 se é negativo, 0 se é positivo ou nulo);

- **Códigos de Condição:** Sinais gerados pela ULA que notificam a UC sobre determinado evento durante a realização de uma operação.
- Utilizado pelas instruções: JN e JZ;
- N - negativo: sinal do resultado (1 se é negativo, 0 se é positivo ou nulo);
- Z – zero: indica se o resultado é zero (1 se é igual a 0, 0 se é diferente de 0);

- Códigos de Condição: Sinais gerados pela ULA que notificam a UC sobre determinado evento durante a realização de uma operação.
- Utilizado pelas instruções: JN e JZ;
- N - negativo: sinal do resultado (1 se é negativo, 0 se é positivo ou nulo);
- Z – zero: indica se o resultado é zero (1 se é igual a 0, 0 se é diferente de 0);
- Apenas as instruções lógicas e aritméticas (ADD, NOT, AND, OR) e de transferência (LDA) afetam os códigos de condição N e Z;

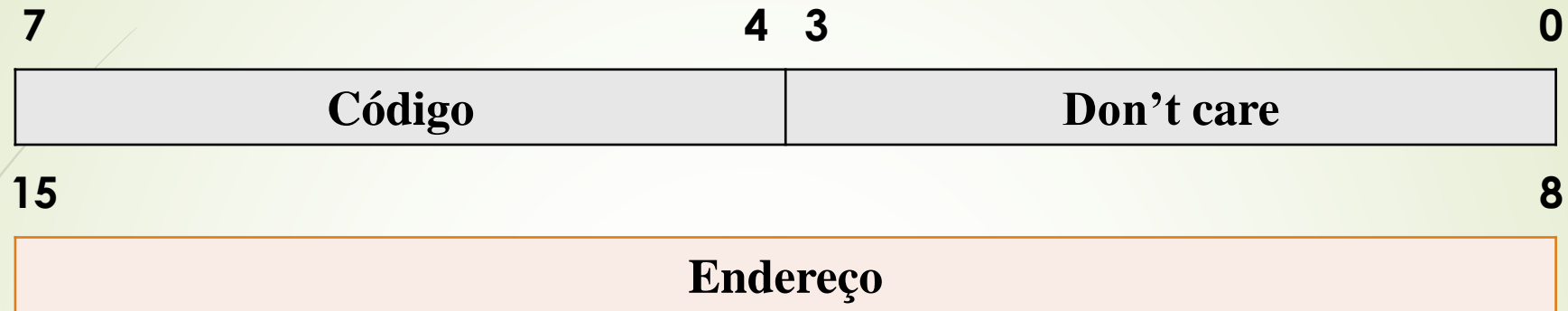


Figura 2: Formato das Instruções no Neander. Fonte: Adaptado de Weber, 2001.

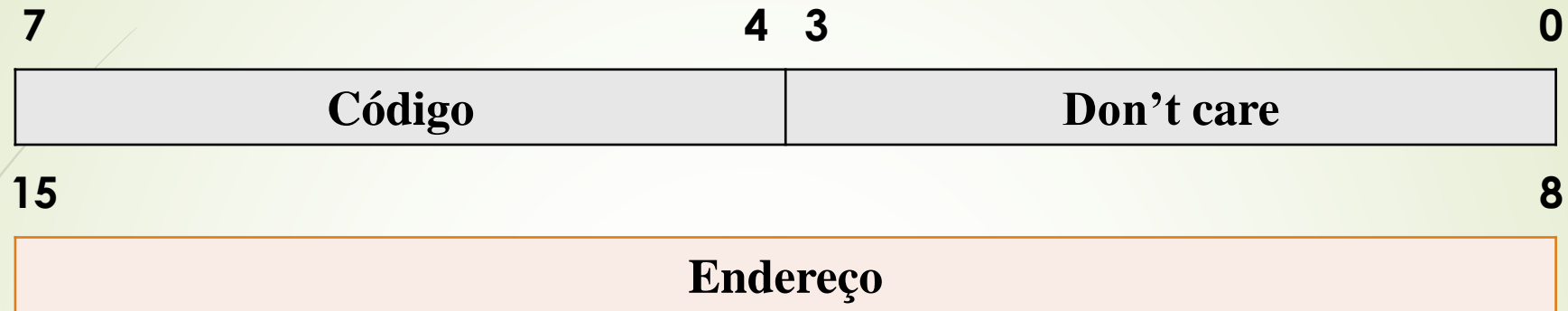


Figura 2: Formato das Instruções no Neander. Fonte: Adaptado de Weber, 2001.

- Nas instruções de 1 byte, os 4 bits mais significativos contém o código da instrução;

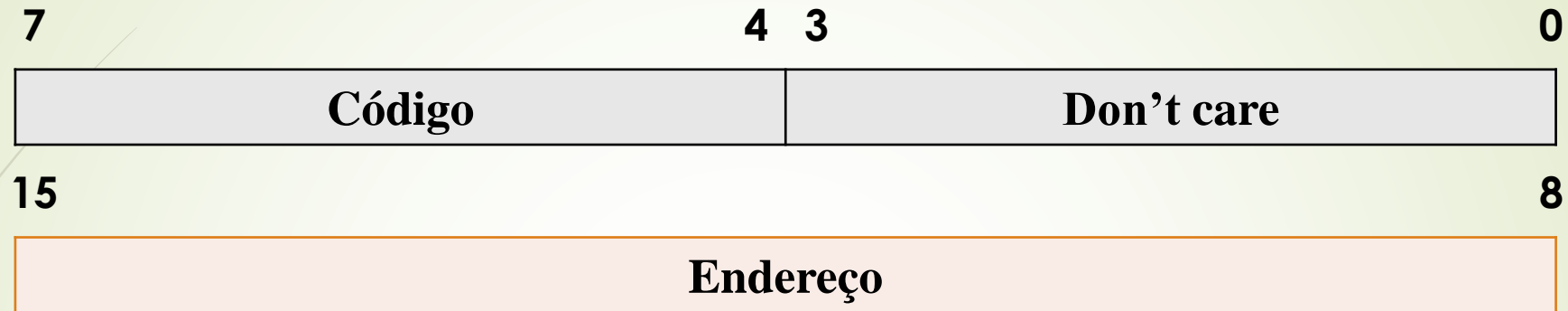


Figura 2: Formato das Instruções no Neander. Fonte: Adaptado de Weber, 2001.

- Nas instruções de 1 byte, os 4 bits mais significativos contém o código da instrução;
- Nas instruções de 2 bytes, os 4 bits mais significativos do 1º byte contém o código da instrução e o 2º byte contém um endereço;

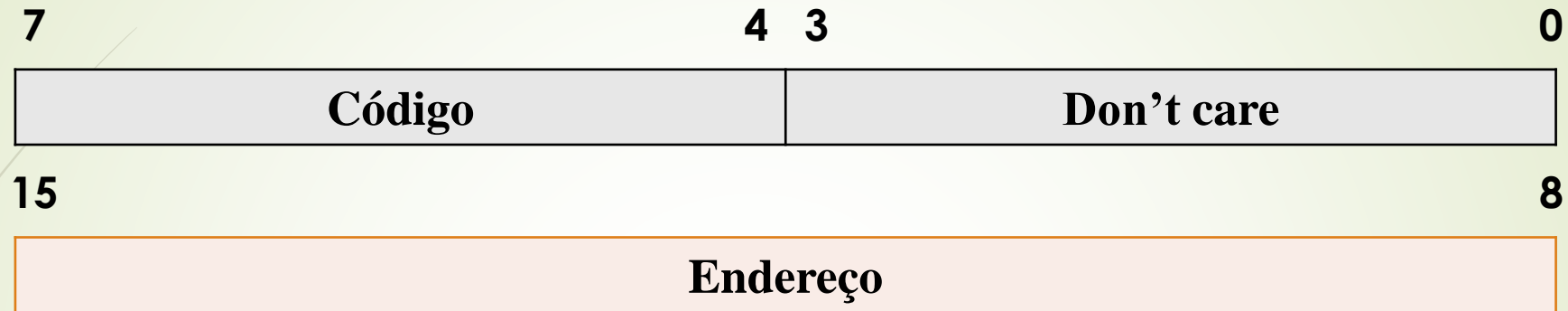


Figura 2: Formato das Instruções no Neander. Fonte: Adaptado de Weber, 2001.

- Nas instruções de 1 byte, os 4 bits mais significativos contém o código da instrução;
- Nas instruções de 2 bytes, os 4 bits mais significativos do 1º byte contém o código da instrução e o 2º byte contém um endereço;
- Instruções de 2 bytes são as que fazem referência a memória;

- A memória no Neander tem 256 posições;
- Por convenção, utiliza-se da posição 0-127 para instruções e da posição 128- 255 para dados;

Simbólico	Decimal	Comentários
LDA 128	32 128	Acumulador AC recebe conteúdo da posição 128;
ADD 129	48 129	Conteúdo do AC é somado ao conteúdo da posição 129;
ADD 130	48 130	Conteúdo do AC é somado ao conteúdo da posição 130;
STA 131	16 131	Conteúdo do AC é armazenado na posição 131 da memória;
HLT	240	Processador para, fim do programa;

Tabela 3: Exemplo de programação no Neander. Fonte: Adaptado de Weber, 2001.

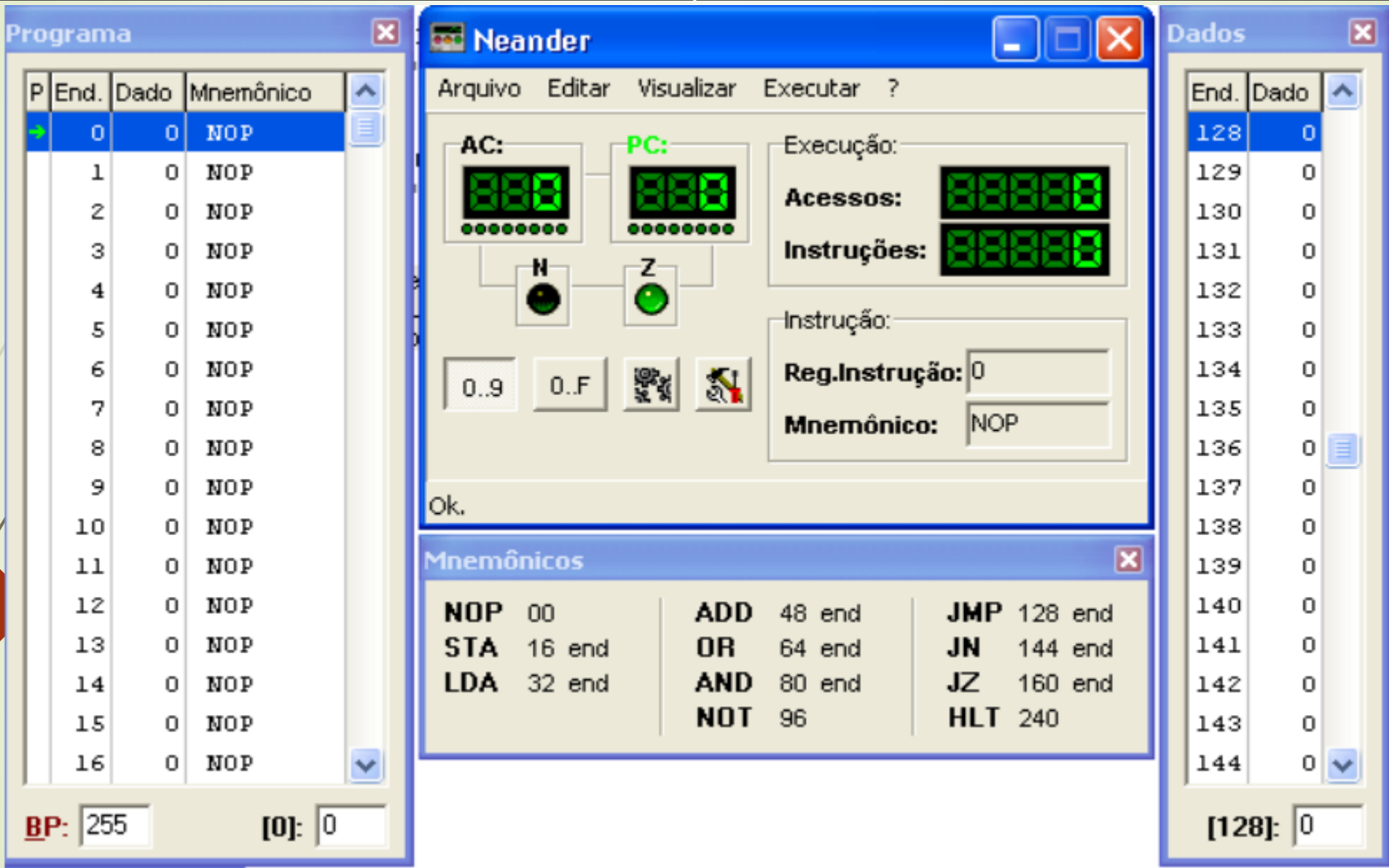


Figura 3: Simulador Neander. Fonte: Elaborada pelo autor.

Exercícios – Tentar fazer até a próxima aula

1. Faça um programa que some 2 variáveis (posição 128, 129) e armazene o resultado (posição 130).
2. Faça um programa para subtrair 2 variáveis (posição 128, 129) e armazene o resultado na posição 130 e 131 (130 – subtração em complemento de B-1, 131 – subtração em complemento de B).
3. Determine qual é o maior de 2 valores (128,129), armazenando o maior na posição 130.
4. Determine qual é o maior de 3 valores (128,129,130), armazenando o maior na posição 131.
5. Faça um programa que determine se ocorreu overflow na soma.
 - a) Operandos – posição 128 e 129;
 - b) Resultado da soma – posição 130;
 - c) Overflow – posição 131 (1 se ocorreu overflow, 0 se não ocorreu)
 - d) Constante 1 – posição 132;



Dúvidas ?