Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

USP - ICMC - SSC
SSC 0510 - Informática - 2o. Semestre 2009

Disciplina de Arquitetura de Computadores

Prof. Fernando Santos Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Página Pessoal: http://www.icmc.usp.br/~fosorio/

Estagiário PAE Maurício Dias - Email: acdias29 [at] yahoo.com.br

Material on-line: COTEIA - http://coteia.icmc.usp.br

Aula 04

Aula 04 - Tópicos Abordados

Conteúdos Abordados:

1. Neander: Breve Revisão
2. Neander: Linguagem de Máquina (*Assembly*)
   Características do Conjunto de Instruções (*Instruction Set*)
   Programando no Neander
3. Conceitos de Programação em *Assembly*
   Modos de Endereçamento: *Direto, Indireto, Imediato, Indexado*
   Tipos de Endereço: *Absoluto, Relativo*
4. Microprocessador 6502
   Arquitetura
   Programação

Arquiteturas Didáticas: Neander

Neander - Computador Hipotético [Weber 2001*]

Arquitetura: características gerais

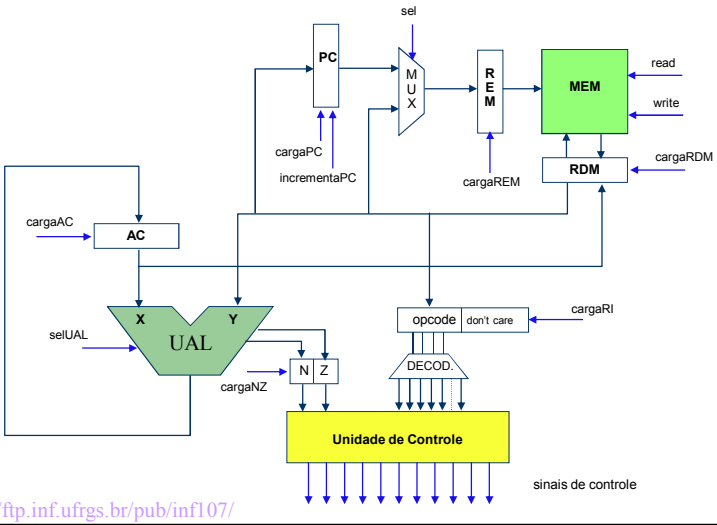
- Largura de dados e endereços de 8 bits (bus)
- Dados representados em complemento de 2
- Acumulador de 8 bits (AC - Accumulator)
- Apontador de programa de 8 bits (PC - Program Counter)
- Registrador de Instruções de 8 bits (IR - Instruction Reg.)
- Registrador de estado (flags) com 2 códigos de condição: Negativo (N) e Zero (Z)
- Endereçamento de memória total de 256 bytes

* ftp://ftp.inf.ufrgs.br/pub/inf107/

USP - SSC-510 - Informática T1 / 2009-2
Arquitetura de Computadores
Prof. Fernando Osório

Arquiteturas Didáticas: Neander

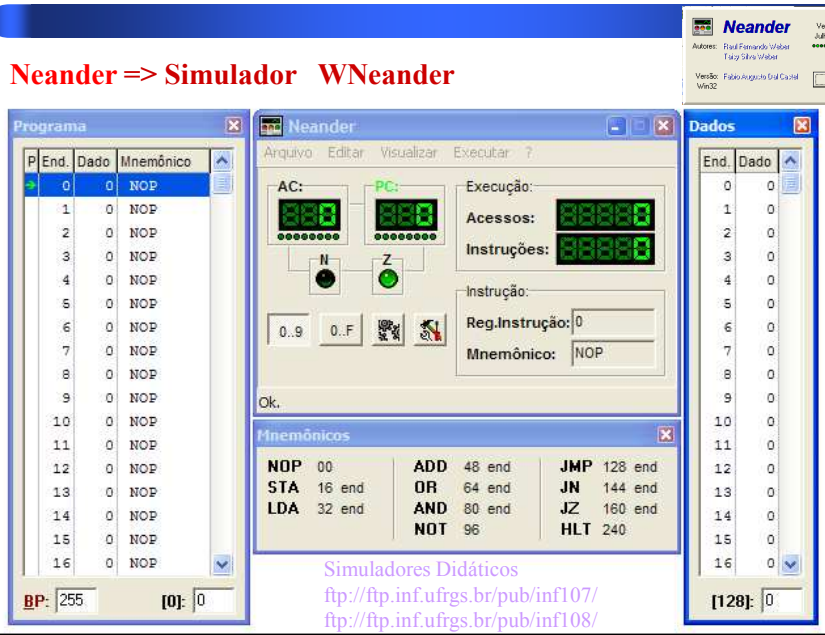
Neander - Computador Hipotético [Weber 2001*]



* ftp://ftp.inf.ufrgs.br/pub/inf107/

Neander: Simulação

Neander => Simulador WNeander



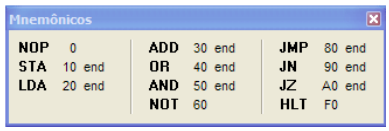
Simuladores Didáticos
ftp://ftp.inf.ufrgs.br/pub/inf107/
ftp://ftp.inf.ufrgs.br/pub/inf108/

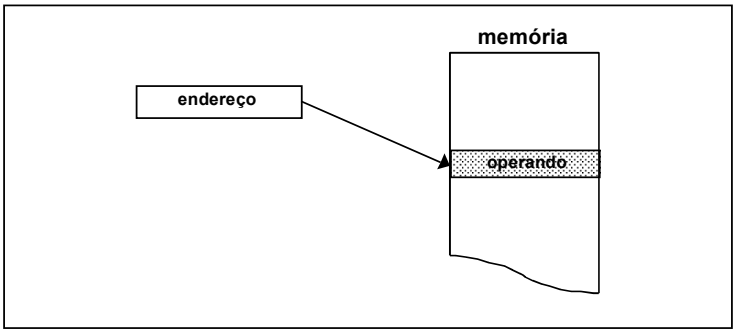
Neander: Programação

Neander - Modos de Endereçamento

NEANDER - Somente UM modo de endereçamento:

Modo Direto

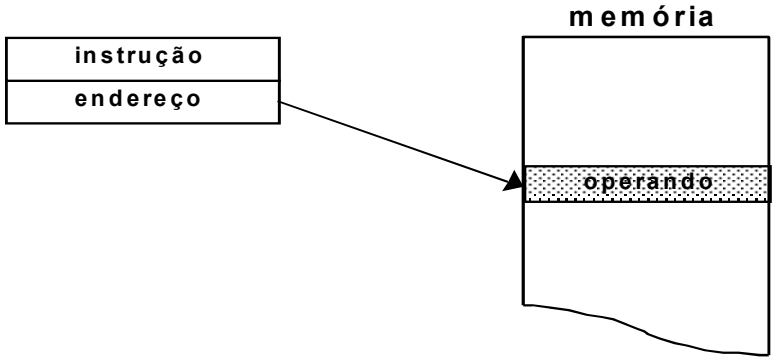




Neander: Programação

Neander - Modos de Endereçamento

Endereçamento direto (absoluto)



Neander: Programação

Neander - Gerando o código...

Mnemônicos

NOP	0	ADD	30 end	JMP	80 end
STA	10 end	OR	40 end	JN	90 end
LDA	20 end	AND	50 end	JZ	A0 end
		NOT	60	HLT	F0

Linguagem de Montagem (Assembly Language)

```
LDA  Valor1   ; Valor1 <=> End. $50
ADD  Valor2   ; Valor2 <=> End. $51
STA  Valor3   ; Valor3 <=> End. $52
```

Montador (Assembler)

Código de Execução / Código de Máquina

```
$20  $50
$30  $51
$10  $52
```

Linguagem de Montagem (Assembly): Mnemônicos, Labels e Diretivas

```
                ORG          [PosicaoMemoria]
[NomeVariavel:] DB           [Valor_Inicial]          [; comentario]
[Rotulo:]       Mnemonico    [Operandos]              [; comentario]
```

> Existe um montador que foi desenvolvido para as arquiteturas dos processadores didáticos: Deadalus (Montador do Ahmes, Ramses e Cesar)
> Estamos adotando uma linguagem similar mas não idêntica

Neander: Programação - Exemplo

Neander - Somar 2 números

Mnemônicos

NOP	0	ADD	30 end	JMP	80 end
STA	10 end	OR	40 end	JN	90 end
LDA	20 end	AND	50 end	JZ	A0 end
		NOT	60	HLT	F0

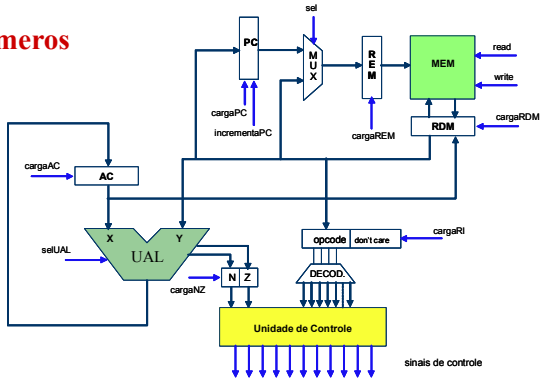
```
                ORG         $00     ; Início da área de programa

Inicio:         LDA         Valor1 ; Lê o valor contido no end. Valor1
                ADD         Valor2 ; Soma com o dado do end. Valor2
                STA         Valor3 ; Armazena no end. Valor3
                HLT

                ORG         $50    ; Início da área de dados
Valor1          DB          $10    ; Armazena o valor $10 neste end.
Valor2          DB          $23    ; Armazena o valor $23 neste end.
Valor3          DB          $00    ; Cria uma posição vazia neste end.

                END.
```

Após a montagem... (valores com $ estão em hexadecimal)

```
00: $20  $50      50: $10          Após a execução...  50: $10
02: $30  $51      51: $23                              51: $23
04: $10  $52      52: $00                              52: $33
```


Neander: Programação - Exemplo

Neander - Somar 2 números

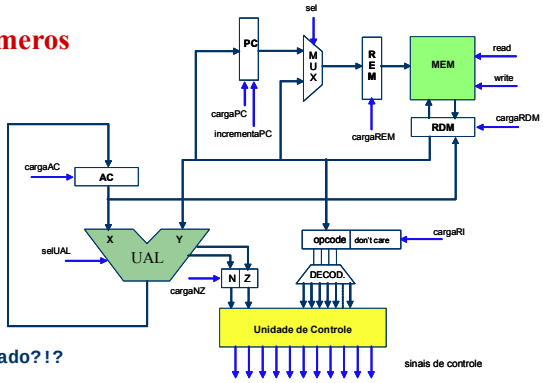



```
SOMAR 2 NÚMEROS

LDA  $50
ADD  $51
STA  $52
HLT

Acessos a memória:

LDA $50
1) Lê OP Code "LDA"
2) Lê Operando: End. $50
3) Lê Conteúdo do End. $50

ADD $51
1) Lê Op Code "ADD"
2) Lê Operando: End. $51
3) Lê Conteúdo do End. $51

STA $52
1) Lê Op Code "STA"
2) Lê Operando: End. $52
3) Armazena dado no End. $52

Total de Acessos a memória:  3x3 = 9 acessos
```

Neander: Programação - Exemplo

Neander - Somar 2 números

```
SOMAR 2 NÚMEROS

LDA  $50
ADD  $51
STA  $52
HLT

Acessos a memória:

E se o operando já fosse
o próprio dado a ser somado?!?

LDA #$10
ADD #$23
STA $51

Total de Acessos a memória:
2 x 2 + 3 = 7 acessos
```

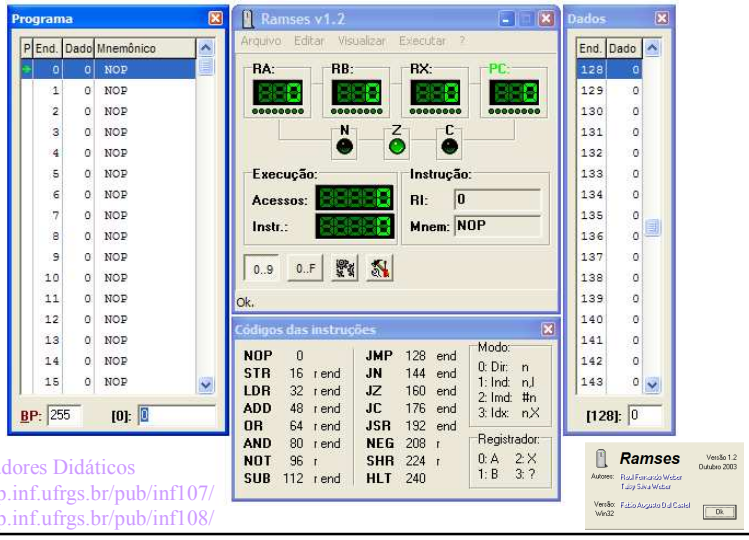


Modo Imediato

```
LDA #$10
1) Lê OP Code "LDA"
2) Lê Operando: Valor $10
3) -
```

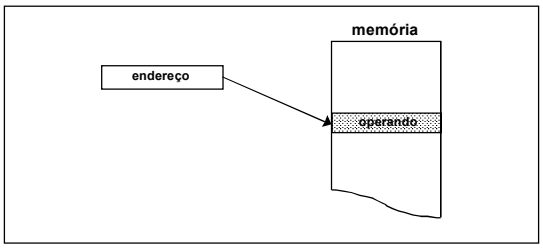

Arquiteturas Didáticas: Ramses

Neander x Ramses



Simuladores Didáticos
ftp://ftp.inf.ufrgs.br/pub/inf107/
ftp://ftp.inf.ufrgs.br/pub/inf108/

Modos de Endereçamento



NEANDER:

Registradores: AC
Modos de endereçamento:
1 - Direto

OP Code: Código + Registrador + Operador

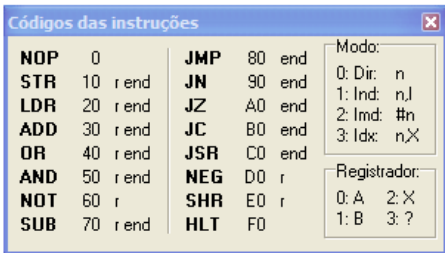
- 00 = A (registrador RA)
- 01 = B (registrador RB)
- 10 = X (registrador de índice)
- 11 = nenhum registrador

- 00 = direto
- 01 = indireto
- 10 = imediato
- 11 = indexado

RAMSES:

Registradores: RA, RB, RX
Modos de endereçamento:
1 - Direto
2 - Indireto
3 - Imediato
4 - Indexado

Modos de Endereçamento



Instrução 2 bytes:

Byte 1: Instrução
xxxx rr ee

Byte 2: Operando
endereço/dado

OP Code: Código + Registrador + Operador

- 00 = A (registrador RA)
- 01 = B (registrador RB)
- 10 = X (registrador de índice)
- 11 = nenhum registrador

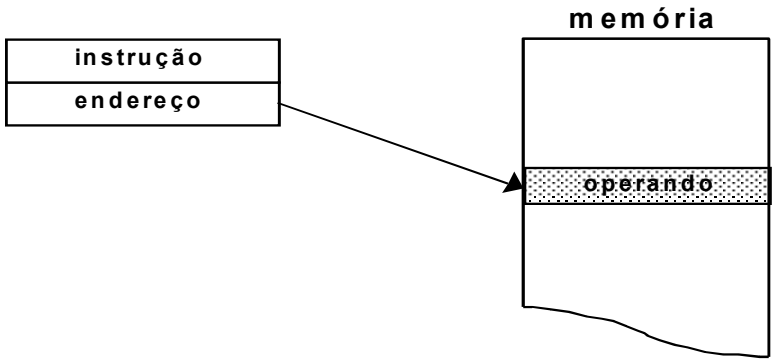
- 00 = direto
- 01 = indireto
- 10 = imediato
- 11 = indexado

RAMSES:

Registradores: RA, RB, RX
Modos de endereçamento:
1 - Direto
2 - Indireto
3 - Imediato
4 - Indexado

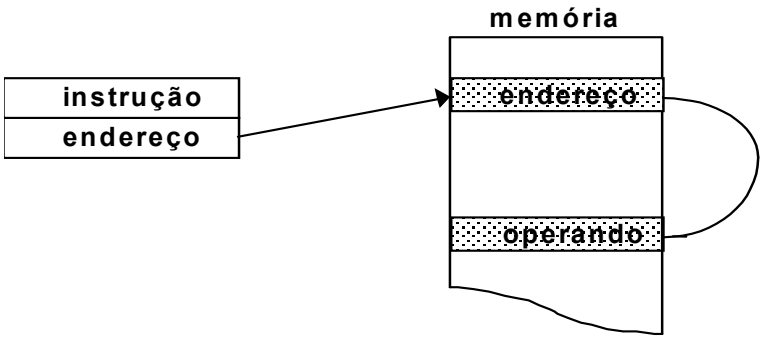
Arquiteturas: Modos de Endereçamento

Endereçamento direto (absoluto)



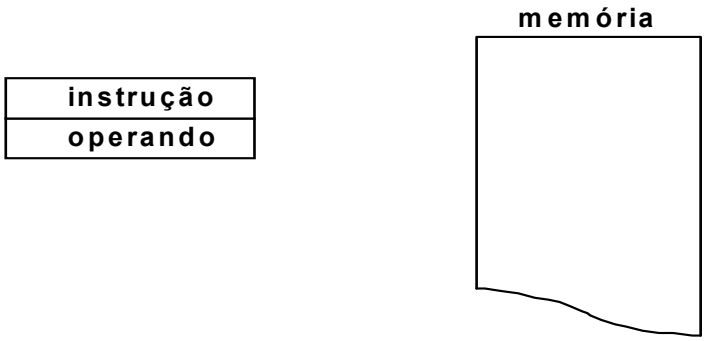
Arquiteturas: Modos de Endereçamento

Endereçamento indireto



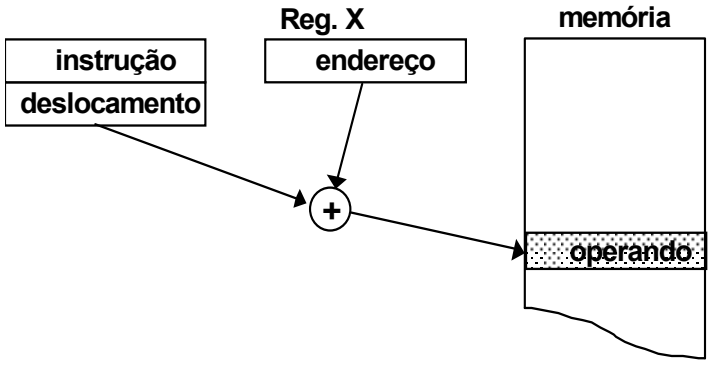
Arquiteturas: Modos de Endereçamento

Endereçamento imediato



Arquiteturas: Modos de Endereçamento

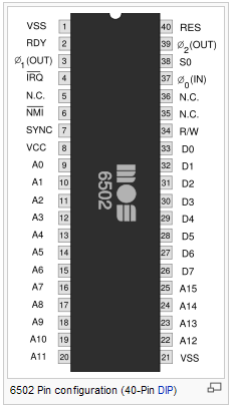
Endereçamento indexado



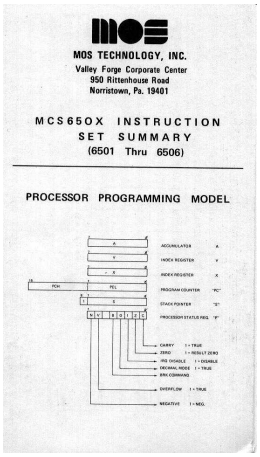
3. Arquitetura de Microprocessador Comercial

MosTech 6502 - Adotado no Apple II

Microprocessador de 8 bits dados e 16 bits de endereço



6502 Pin configuration (40-Pin DIP)


MOS TECHNOLOGY, INC.
Valley Forge Corporate Center
950 Rittenhouse Road
Norristown, Pa. 19401

MCS650X INSTRUCTION SET SUMMARY
(6501 Thru 6506)

PROCESSOR PROGRAMMING MODEL

* http://en.wikipedia.org/wiki/MOS_Technology_6502

USP - SSC-510 - Informática T1 / 2009-2
Arquitetura de Computadores
Prof. Fernando Osório

3. Arquitetura de Microprocessador Comercial

MosTech 6502 - Adotado no Apple II

Microprocessador de 8 bits dados e 16 bits de endereço

Registers

The 6502 only has 6 registers. Five are 8 bits wide, one is 16 bits wide.

A		Accumulator
X		Index Register X
Y		Index Register Y
PCH	PCL	Program Counter
S		Stack Pointer
P		Processor Status (Flags)

Processor Status Register

N	V	-	B	D	I	Z	C

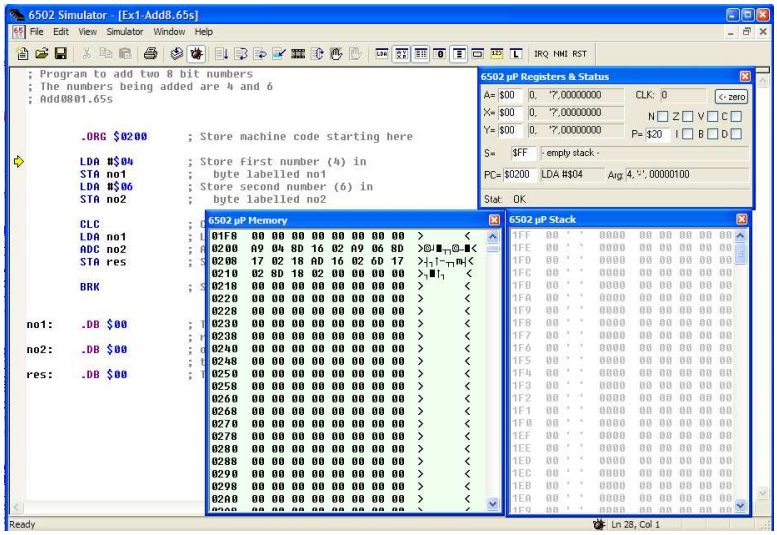
```
N - Negative Flag
V - Overflow Flag
B - Break Command
D - Decimal Mode
I - Int. Disable
Z - Zero Flag
C - Carry Flag
```

Accumulator **8 bits** **A**

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

3. Arquitetura de Microprocessador Comercial

MosTech 6502 - Adotado no Apple II



* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/ssc/
Página pessoal: http://www.icmc.usp.br/~fosorio/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Arquitetura de Computadores / Informática
Estagiário PAE: Maurício A. Dias
Web disciplina: COTEIA - Http://coteia.icmc.usp.br
> Programa, Material de Aulas, Critérios de Avaliação,
> Lista de Exercícios, Trabalhos Práticos, Datas das Provas

Visite regularmente o site da CoTeia - Material em permanente atualização