

# USP - ICMC - SSC

## SSC 0511 - Sist. Informação - 2o. Semestre 2014

# Disciplina de Organização de Computadores Digitais

**Prof. Fernando Santos Osório**

**Email: fosorio [at] { icmc. usp. br , gmail. com }**

**Página Pessoal: <http://www.icmc.usp.br/~fosorio/>**

**Material on-line: Wiki ICMC:**

**[http://wiki.icmc.usp.br/index.php/SSC-511-2014\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-511-2014(fosorio))**



Lab. de Robótica Móvel



Centro de Robótica da USP

## Agenda:

- 1. Arquiteturas de Von Neumann:**  
**Colocando o processador a funcionar**
  - > UC
  - > Registradores
  - > ULA
  - > Memória
  - > E/S
- 2. Arquitetura do NEANDER**

## 4. Arquiteturas Didáticas

### **Neander - Computador Hipotético [Weber 2001\*]**

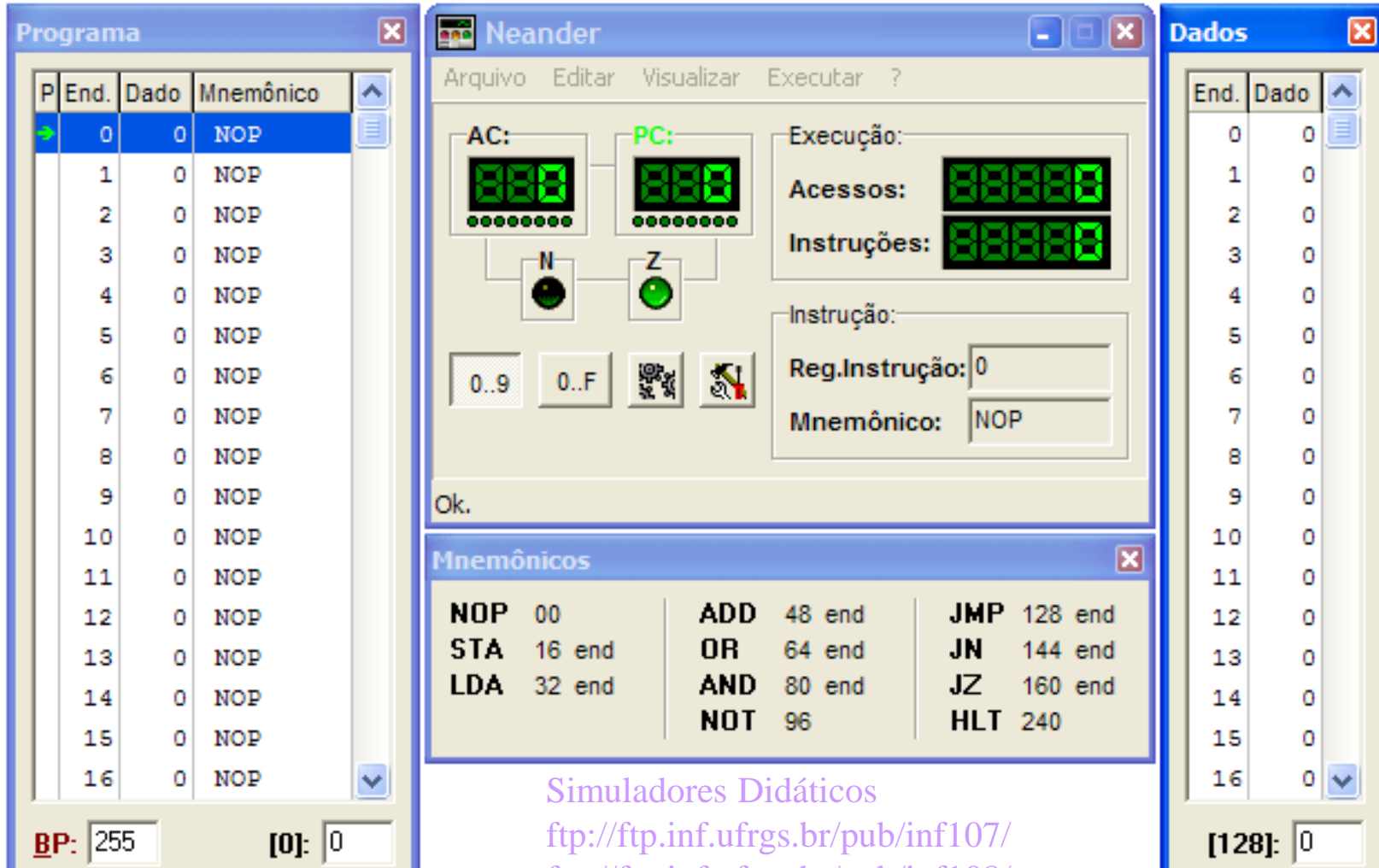
#### **Arquitetura: características gerais**

- **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**

## 4. Arquiteturas Didáticas

Neander => Simulador WNeander

 **Neander** Versão 2.1  
Julho 2002  
Autores: Raul Fernando Weber  
Taizy Silva Weber  
Versão: Fabio Augusto Dal Castel Win32



The screenshot displays the Neander simulator interface, which is divided into several windows:

- Programa:** A table showing the program memory. The first row is highlighted, indicating the current instruction.
- Neander:** The main simulator window showing the CPU status. It includes:
  - AC (Accumulator):** A 3-digit display showing 000.
  - PC (Program Counter):** A 3-digit display showing 000.
  - Execução:** A section with two 5-digit displays for **Acessos** (00000) and **Instruções** (00000).
  - Instrução:** A section with two input fields: **Reg.Instrução:** 0 and **Mnemônico:** NOP.
  - Flags:** Two indicator lights, **N** (Not) and **Z** (Zero), both currently off.
  - Buttons:** **0..9**, **0..F**, and two function buttons.
  - Ok.** button at the bottom.
- Mnemônicos:** A table listing the instruction set:

<b>NOP</b>	00	<b>ADD</b>	48 end	<b>JMP</b>	128 end
<b>STA</b>	16 end	<b>OR</b>	64 end	<b>JN</b>	144 end
<b>LDA</b>	32 end	<b>AND</b>	80 end	<b>JZ</b>	160 end
		<b>NOT</b>	96	<b>HLT</b>	240
- Dados:** A table showing the data memory. The first row is highlighted, indicating the current data location.

At the bottom of the simulator, there are two input fields: **BP:** 255 and **[0]:** 0. The **Dados** window also has an input field: **[128]:** 0.

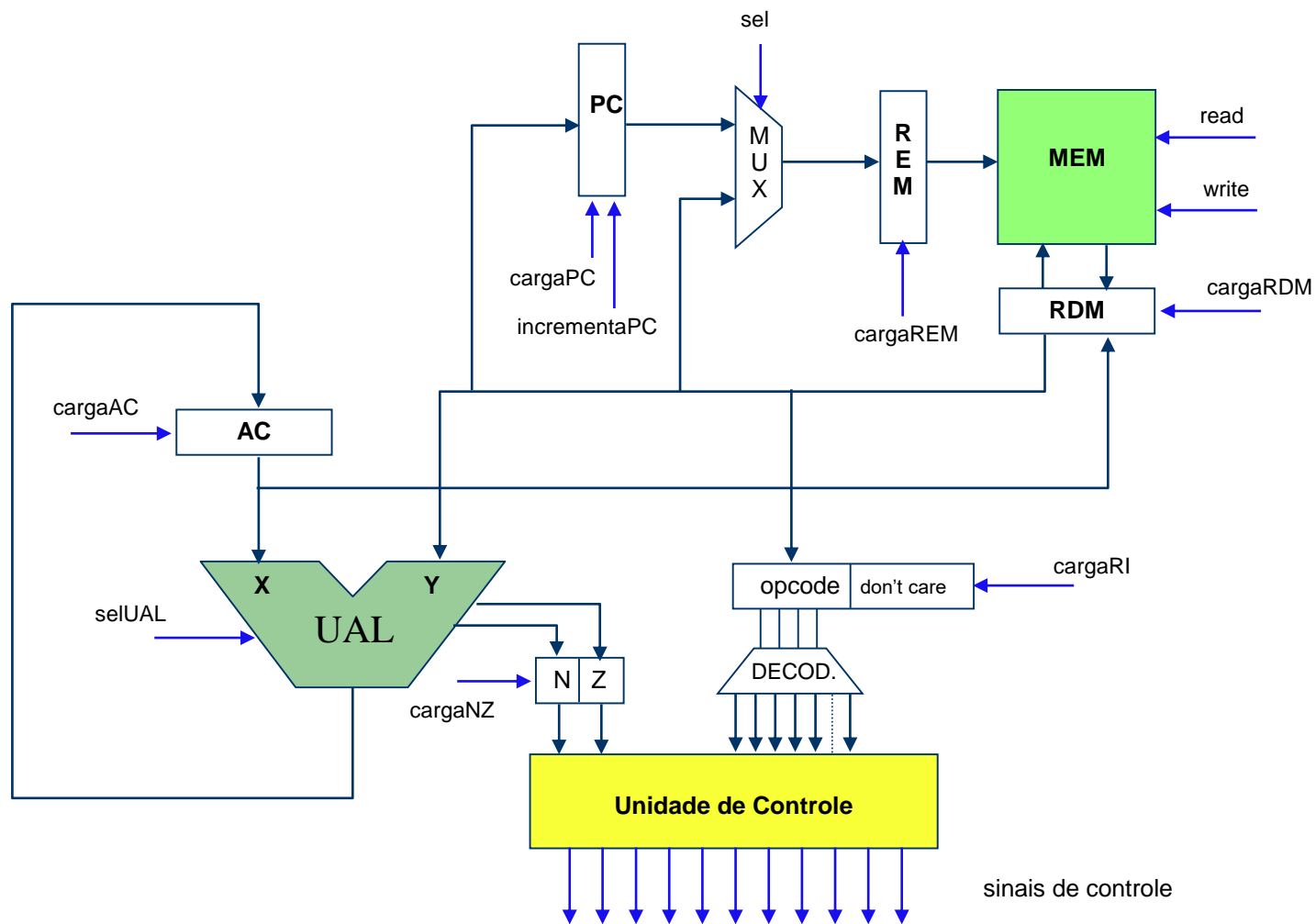
Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>

## 4. Arquiteturas Didáticas

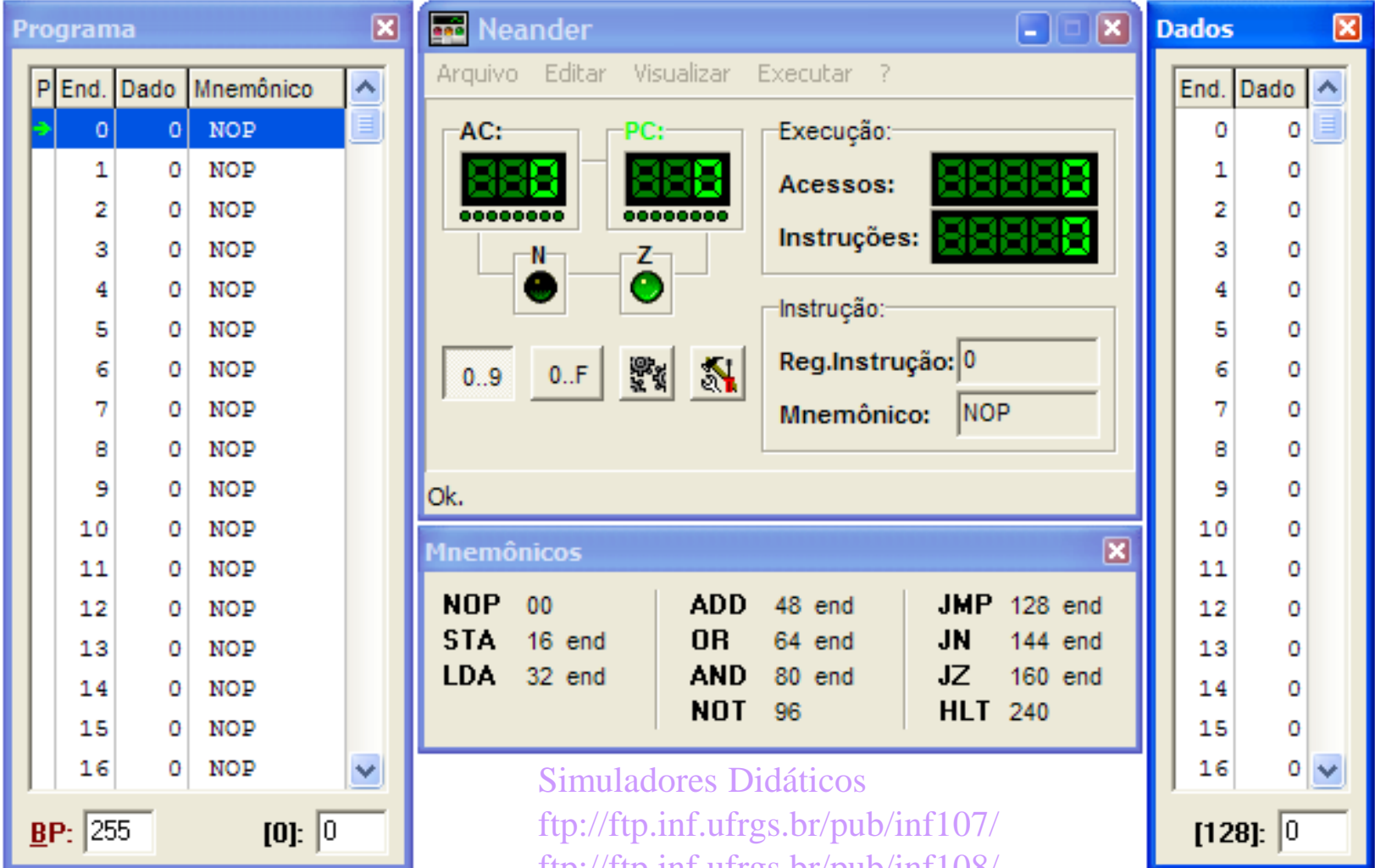
### Neander - Computador Hipotético [Weber 2001\*]



## 4. Arquiteturas Didáticas

Neander => Simulador WNeander

 **Neander** Versão 2.1  
Julho 2002  
Autores: Raul Fernando Weber  
Taizy Silva Weber  
Versão: Fabio Augusto Dal Castel Win32



The screenshot displays the Neander simulator interface, which is divided into several windows:

- Programa:** A table showing the program memory. The first row is highlighted, indicating the current instruction.
- Neander:** The main simulator window showing the CPU status. It includes:
  - AC (Accumulator):** A 3-digit display showing 000.
  - PC (Program Counter):** A 3-digit display showing 000.
  - Execução:** A section with two 5-digit displays for **Acessos** (00000) and **Instruções** (00000).
  - Instrução:** A section with two input fields: **Reg.Instrução:** 0 and **Mnemônico:** NOP.
  - Flags:** Two indicator lights, **N** (Not) and **Z** (Zero), both currently off.
  - Buttons:** **0..9**, **0..F**, and two function buttons.
  - Ok.** button at the bottom.
- Mnemônicos:** A table listing the instruction set and their addresses.
- Dados:** A table showing the data memory, currently empty.

At the bottom of the main window, there are two input fields: **BP:** 255 and **[0]:** 0.

Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>

## 2. Programação do Neander

### Programação do Neander

Código Binário	Instrução em Hexa	instrução	comentário
0000	00	NOP	Nenhuma operação
0001	10 XX	STA end	MEM(end) ← AC
0010	20 XX	LDA end	AC ← MEM(end)
0011	30 XX	ADD end	AC ← MEM(end) + AC
0100	40 XX	OR end	AC ← MEM(end) OR AC
0101	50 XX	AND end	AC ← MEM(end) AND AC
0110	60	NOT	AC ← NOT AC
1000	80 XX	JMP end	PC ← end
1001	90 XX	JN end	IF N=1 THEN PC ← end
1010	A0 XX	JZ end	IF Z=1 THEN PC ← end
1111	F0	HLT	pára processamento

## 2. Programação do Neander

### Programação do Neander :

#### Programação do Neander – Exercícios

- 1) Somar vários valores de 8 bits ( $A + B + C + D + E$ )
- 2) Subtrair valores de 8 bits ( $A - B$ )
- 3) Contador: Laço de contagem até 10
- 4) Somar os dados de um vetor
- 5) Somar valores com mais de 8 bits (!)
- 6) Multiplicar 2 valores
- 7) Pesquisar um dado em uma tabela



## 2. Programação do Neander

### Programação do Neander : Soma Valores

#### Linguagem de Montagem (Código Comentado)

```
Valor1 EQU 200 ; Endereço da variável Valor1 definido como 200 ($C8)
Valor2 EQU 201 ; Endereço da variável Valor2 definido como 201
Valor3 EQU 202 ; Endereço da variável Valor3 definido como 202
Valor4 EQU 203 ; Endereço da variável Valor4 definido como 203
Valor5 EQU 204 ; Endereço da variável Valor5 definido como 204
Result EQU 210 ; Endereço da variável Result definido como 210 ($D2)

ORG $00 ; Endereço inicial da execução do Prog. (Inicia com PC:00)

Ini: LDA Valor1 ; Acumulador AC recebe conteúdo de Valor1 (End. 200)
      ADD Valor2 ; Soma AC = AC + conteúdo do end. Valor2 (End. 201)
      ADD Valor3 ; Soma AC = AC + conteúdo do end. Valor2 (End. 202)
      ADD Valor4 ; Soma AC = AC + conteúdo do end. Valor2 (End. 203)
      ADD Valor5 ; Soma AC = AC + conteúdo do end. Valor2 (End. 204)
      STA Result ; Salva o resultado do AC na memória(End.210)

Fim: HLT ; Termina a execução;

      END ; Fim do código de Montagem
```

## 2. Programação do Neander

### Programação do Neander : Soma Valores

Memória	Instrução em Hexa	instrução
\$00	20 C8	LDA Valor1
\$02	30 C9	ADD Valor2
\$04	30 CA	ADD Valor3
\$06	30 CB	ADD Valor4
\$08	30 CC	ADD Valor5
\$0A	10 D2	STA Result
\$0C	F0	HLT
\$C8	01	Valor1
\$C9	02	Valor2
\$CA	03	Valor3
\$CB	04	Valor4
\$CC	05	Valor5
\$D2	00	Result

## 2. Programação do Neander

### Programação do Neander :

#### Programação do Neander – Exercícios

- 1) Somar vários valores de 8 bits (A + B + C + D + E)
- 2) Subtrair valores de 8 bits (A – B)
- 3) Contador: Laço de contagem até 10
- 4) Somar os dados de um vetor de 10 elementos
- 5) Somar valores com mais de 8 bits (!)
- 6) Multiplicar 2 valores
- 7) Pesquisar um dado em uma tabela

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

## 2. Programação do Neander

### Programação do Neander :

#### Arquitetura do Neander – Críticas?

- Possui apenas 1 modo de endereçamento (Direto Absoluto)
- Possui apenas 1 registrador de uso geral (Acumulador)
- Possui apenas 2 flags de status da ULA (Flip-flops N e Z)
- Possui apenas 11 instruções de máquina (incluindo NOP e HLT)
- Não possui flags de “vai-um” (Carry In, Carry Out)
- Não possui instruções de desvio/retorno de sub-rotina (JSR, RTS)
- Não possui uma pilha auxiliar para dados/endereços (Push, Pop)
- Não possui instruções de acesso imediato a memória (LDA #)
- Não possui instruções de acesso indexado a memória (LDA \$,X)
- Não possui instruções dedicadas de E/S (In, Out)

## 4. Arquiteturas Didáticas

### Evolução do Neander... Ahmes, Ramses, Cesar

#### Quadro comparativo

Arquitetura	Endereços	Dados	Nro. Instruções	Registradores
<b>NEANDER</b>	8 bits 256 bytes	8 bits Compl.2	11 instruções (OpCode: 4bits)	AC, PC, IR, Flags (N,Z) REM, RDM
<b>AHMES</b>	8 bits	8 bits	24 instruções (Neander ext.)	PC, IR, REM, RDM Flags (N, Z, C, B, V)
<b>RAMSES</b>	8 bits	8 bits	Modos de End. 4 modos x 16 instr.	PC, IR, RA, RB, RX Flags (N, Z, V, C)
<b>CESAR</b>	16 bits 64 Kbytes	16 bits	Inúmeras	R0 a R6 (uso geral) R7 (PC)

#### Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>

[http://pt.wikipedia.org/wiki/Máquinas\\_hipotéticas\\_da\\_Universidade\\_Federal\\_do\\_Rio\\_Grande\\_do\\_Sul](http://pt.wikipedia.org/wiki/Máquinas_hipotéticas_da_Universidade_Federal_do_Rio_Grande_do_Sul)

## 4. Arquiteturas Didáticas

### Evolução do Neander... Ahmes, Ramses, Cesar

The image displays the Ahmes educational simulator interface, which is divided into several windows:

- Programa:** A table showing the program memory layout.
 

P	End.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP
16	0	0	NOP
- Ahmes:** The main CPU window showing execution status.
  - AC: 000
  - PC: 000
  - Flags: N (off), Z (on), V (off), C (off), B (off)
  - Execução: 00000
  - Acessos: 00000
  - Instr.: 00000
  - Instrução: R I: 0, Mnem: NOP
  - Buttons: 0..9, 0..F, and a Run button.
- Dados:** A table showing the data memory layout.
 

End.	Dado
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0
144	0
- Mnemônicos:** A table listing the instruction set.
 

NOP	00	JMP	128 end	SHR	224
STA	16 end	JN	144 end	SHL	225
LDA	32 end	JP	148 end	ROR	226
ADD	48 end	JV	152 end	ROL	227
OR	64 end	JNV	156 end	HLT	240
AND	80 end	JZ	160 end		
NOT	96 end	JNZ	164 end		
SUB	112 end	JC	176 end		
		JNC	180 end		
		JB	184 end		
		JNB	188 end		

At the bottom right, there is a footer for the Ahmes simulator, including the version (Versão 2.1, Julho 2002), authors (Fábio Fernando Weber, Taicy Silva Weber), and a version by Fábio Augusto Dal Castel (Win32).

Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>  
<ftp://ftp.inf.ufrgs.br/pub/inf108/>

## 4. Arquiteturas Didáticas

### Evolução do Neander... Ahmes, Ramses, Cesar

**Programa**

P	End.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

BP: 255 [0]: 0

**Ramses v1.2**

Arquivo Editar Visualizar Executar ?

RA: [000] RB: [000] RX: [000] PC: [000]

N [ ] Z [ ] C [ ]

Execução: Acessos: [0000] Instr.: [0000]

Instrução: RI: 0 Mnem: NOP

0..9 0..F [ ] [ ]

Ok.

**Códigos das instruções**

NOP	0	JMP	128	end	Modo: 0: Dir: n 1: Ind: n,l 2: lmd: #n 3: ldx: nX
STR	16	JN	144	end	
LDR	32	JZ	160	end	
ADD	48	JC	176	end	
OR	64	JSR	192	end	Registrador: 0: A 2: X 1: B 3: ?
AND	80	NEG	208	r	
NOT	96	SHR	224	r	
SUB	112	HLT	240		

**Dados**

End.	Dado
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0

[128]: 0

**Ramses** Versão 1.2 Outubro 2003

Autores: Raul Fernando Weber  
Taisy Siva Weber

Versão: Win32 Fátio Augusto Dal Castel

Ok.

Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>

## 4. Arquiteturas Didáticas

### Evolução do Neander... Ahmes, Ramses, Cesar

**Programa**

P	Ender.	Dado	Mnemônico
	0	0	NOP
	1	0	NOP
	2	0	NOP
	3	0	NOP
	4	0	NOP
	5	0	NOP
	6	0	NOP
	7	0	NOP
	8	0	NOP
	9	0	NOP
	10	0	NOP
	11	0	NOP
	12	0	NOP
	13	0	NOP
	14	0	NOP
	15	0	NOP

BP: 65535      [0]: 0

**Cesar 16**

Arquivo Editar Visualizar Executar ?

R0: 00000    R1: 00000    R2: 00000  
R3: 00000    R4: 00000    R5: 00000  
R6: (SP) 00000    R7: (PC) 00000

Execução:    N    Z    V    C  
Acessos: 00000         
Instr.: 00000    0..9    0..F   

Instrução:  
RI: 0  
Mnem: NOP

Ok.

**Dados**

Ender.	Dado
1024	0
1025	0
1026	0
1027	0
1028	0
1029	0
1030	0
1031	0
1032	0
1033	0
1034	0
1035	0
1036	0
1037	0
1038	0
1039	0

[1024]: 0

Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>



## 4. Arquiteturas Didáticas

### Evolução do Neander... Ahmes, Ramses, Cesar

#### Quadro comparativo

Arquitetura	Endereços	Dados	Nro. Instruções	Registradores
<b>NEANDER</b>	8 bits 256 bytes	8 bits Compl.2	11 instruções (OpCode: 4bits)	AC, PC, IR, Flags (N,Z) REM, RDM
<b>AHMES</b>	8 bits	8 bits	24 instruções (Neander ext.)	PC, IR, REM, RDM Flags (N, Z, C, B, V)
<b>RAMSES</b>	8 bits	8 bits	Modos de End. 4 modos x 16 instr.	PC, IR, RA, RB, RX Flags (N, Z, V, C)
<b>CESAR</b>	16 bits 64 Kbytes	16 bits	Inúmeras	R0 a R6 (uso geral) R7 (PC)

#### Simuladores Didáticos

<ftp://ftp.inf.ufrgs.br/pub/inf107/>

<ftp://ftp.inf.ufrgs.br/pub/inf108/>

[http://pt.wikipedia.org/wiki/Máquinas\\_hipotéticas\\_da\\_Universidade\\_Federal\\_do\\_Rio\\_Grande\\_do\\_Sul](http://pt.wikipedia.org/wiki/Máquinas_hipotéticas_da_Universidade_Federal_do_Rio_Grande_do_Sul)

## 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](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. com](mailto:fosorio@gmail.com)**

**Disciplina de Organização de Computadores Digitais / BSI**

**Web disciplina: Wiki ICMC - [Http://wiki.icmc.usp.br](http://wiki.icmc.usp.br)**

**> Programa, Material de Aulas, Critérios de Avaliação,**

**> Lista de Exercícios, Trabalhos Práticos, Datas das Provas**