

**USP - ICMC - SSC  
SSC 0610 - Eng. Comp. - 2o. Semestre 2010**

## **Disciplina de Organização de Computadores I**

**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: maccddias [at] gmail.com**

**Material on-line Wiki ICMC - <http://wiki.icmc.usp.br/index.php/Ssc-610>**

*Aula 03q*

## **Apresentação da Disciplina**

### **Agenda:**

- 1. Arquitetura do NEANDER**
- 2. Programação em Linguagem de Máquina**
- 3. Programando o Neander**

## 1. Arquitetura do 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

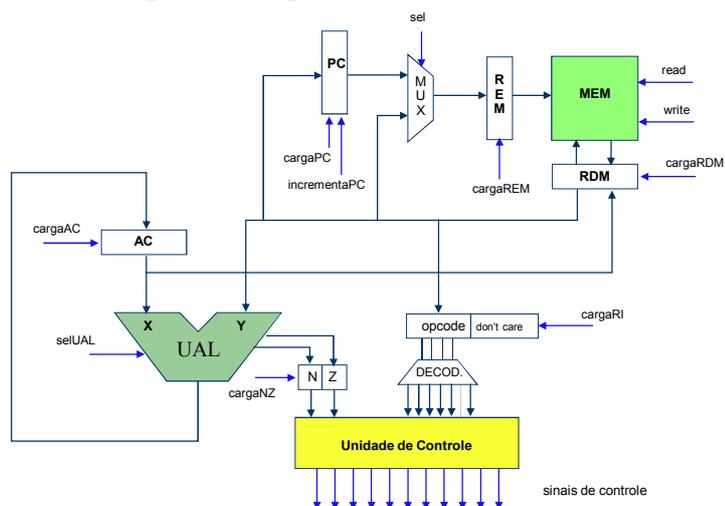
3

Agosto 2009

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

## 1. Arquitetura do Neander

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



4

Agosto 2009

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

Versão Web: <http://inf.ufrgs.br/~vbuaraujo/sw/neander/>

## 1. Arquitetura do Neander

### Neander => Simulador WNeander

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

9

Agosto 2009

## 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

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

10

Agosto 2009

## 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)

## 2. 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)

## 2. Arquiteturas Didáticas

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

The Ahmes simulator interface includes the following components:

- Programa:** A list of instructions with columns for 'P', 'End.', 'Dado', and 'Mnemônico'. The first instruction is NOP at address 0.
- Ahmes:** The main execution window showing:
  - Registers: AC (000000), PC (000000).
  - Status flags: N (off), Z (on), V (off), C (off), B (off).
  - Execution: Accessos (000000), Instr.: (000000).
  - Instruction: RI: 0, Mnem: NOP.
  - Buttons: 0..9, 0..F, and control icons.
- Dados:** A memory dump window with columns for 'End.', 'Dado', and 'Mnemônico'. Address 128 is highlighted.
- Mnemônicos:** A table of instruction codes:
 

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	JNZ 164 end	
SUB 112 end	JC 176 end	
	JNC 180 end	
	JB 184 end	
	JNB 188 end	

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

## 2. Arquiteturas Didáticas

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

The Ramses v1.2 simulator interface includes the following components:

- Programa:** A list of instructions with columns for 'P', 'End.', 'Dado', and 'Mnemônico'. The first instruction is NOP at address 0.
- Ramses v1.2:** The main execution window showing:
  - Registers: RA (000000), RB (000000), RX (000000), PC (000000).
  - Status flags: N (off), Z (on), C (off).
  - Execution: Accessos (000000), Instr.: (000000).
  - Instruction: RI: 0, Mnem: NOP.
  - Buttons: 0..9, 0..F, and control icons.
- Dados:** A memory dump window with columns for 'End.', 'Dado', and 'Mnemônico'. Address 128 is highlighted.
- Códigos das instruções:** A table of instruction codes and modes:
 

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

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



**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 I / Eng. Comp.**

**Estagiário PAE: Maurício A. Dias**

**Web disciplina: <http://wiki.icmc.usp.br/index.php/Ssc-610>**

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

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