

**INSTITUTO FEDERAL**  
Sul-rio-grandense

Câmpus  
Venâncio Aires

# Arquitetura e Organização de Computadores

Professor: Fernando Luís Herrmann

E-mail: [fernandoherrmann@ifsul.edu.br](mailto:fernandoherrmann@ifsul.edu.br)



# Material de aula:

<https://github.com/herrmannfl/tads-aoc-2022>



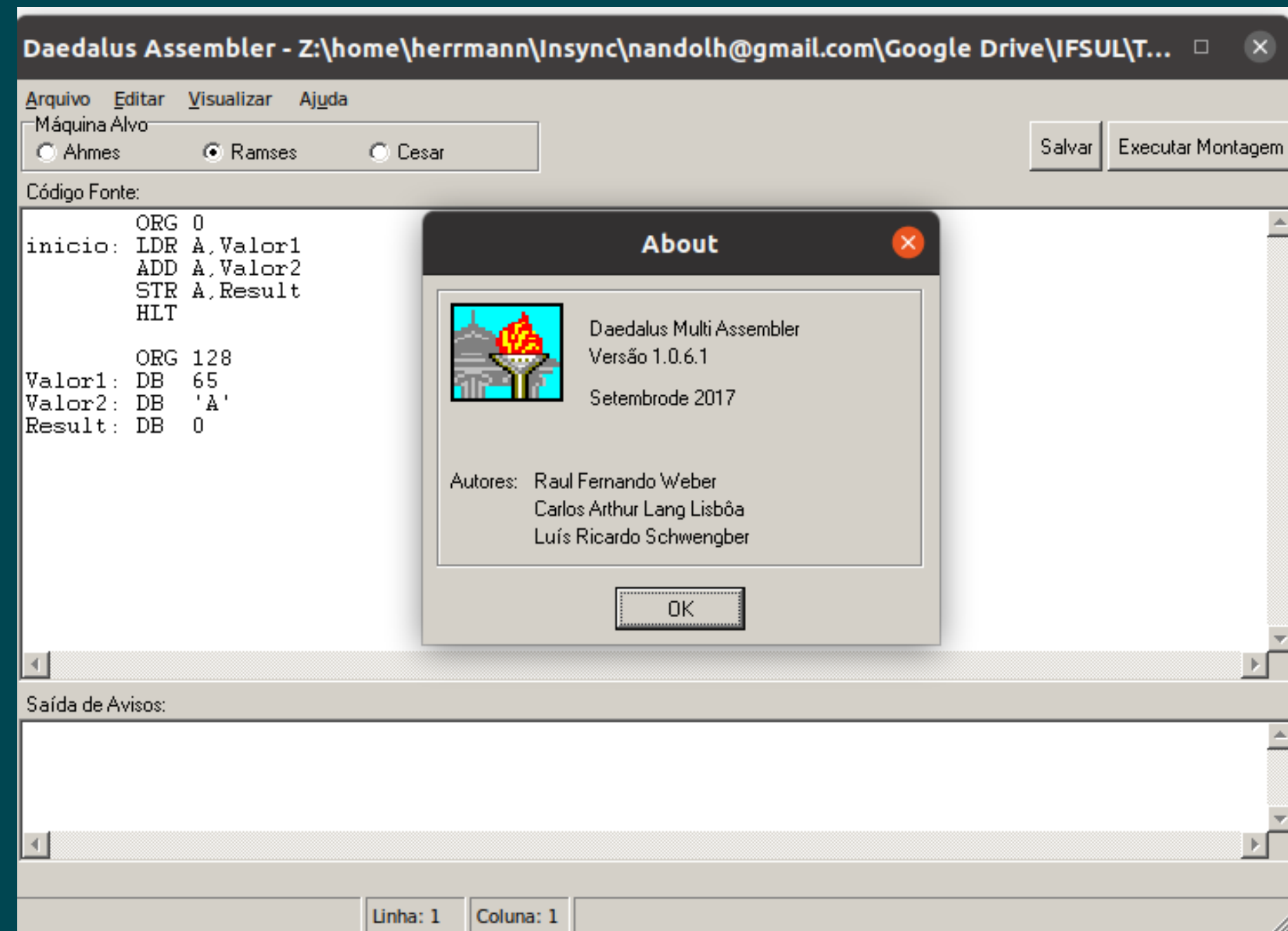
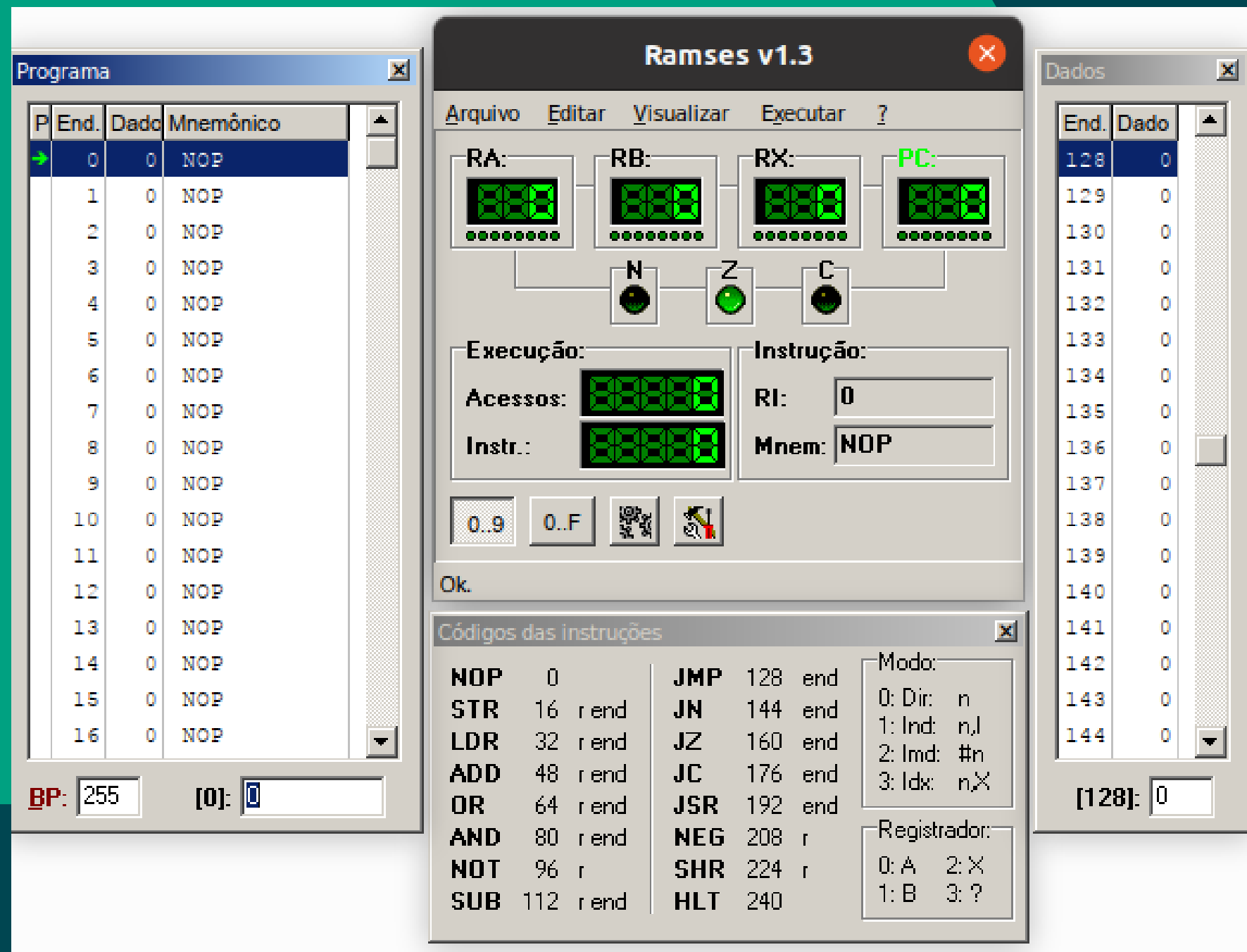
# Arquitetura Didáticas

Arquitetura	Endereços	Dados	Instruções	Registradores
NEANDER	8 bits 256 bytes	8 bits	11 instruções (opcode: 4bits)	AC, PC, IR, Flags (N,Z) REM, RDM
AHMES	8 bits 256 bytes	8 bits	24 instruções Neander extendido	PC, IR, Flags (N, Z, C,, V) REM, RDM
RAMSES	8 bits	8 bits	Modos de endereçamento 4 modos x 16 instruções	PC, IR, RA, RB, RX, Flags (N,Z, C, V) REM, RDM
CESAR	16 bits 64 KBytes	16 bits	Inúmeras	R0 a R6 (uso geral) R7 (PC)

# Simulador Ramses



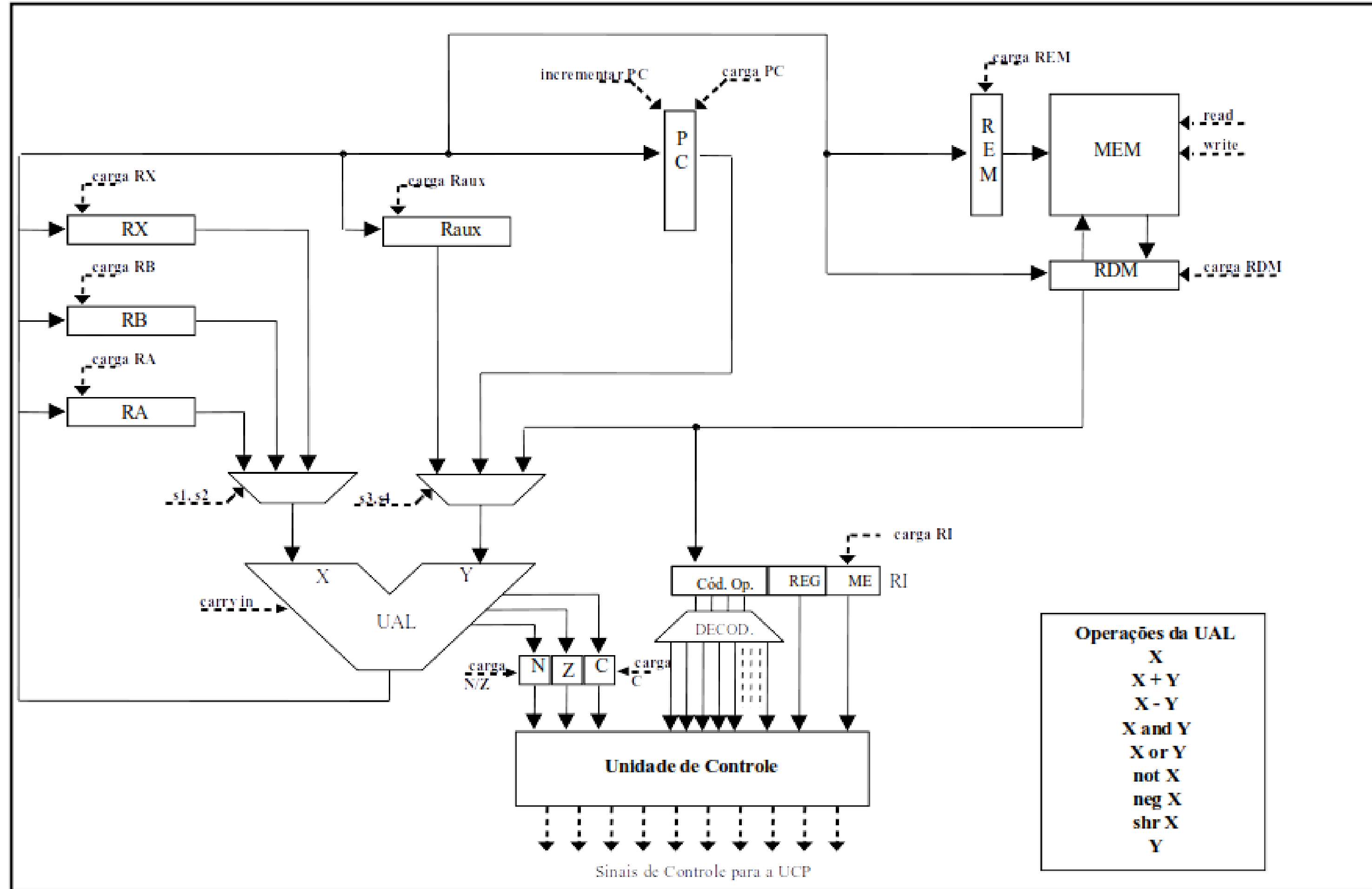
# Montador DAEDALUS



# Características gerais

- Largura de dados e de endereços de 8 bits
- Dados representados em complemento de dois
- 4 modos de endereçamento: direto, indireto, imediato e indexado
- 2 registradores de uso geral de 8 bits
- 1 registrador de índice de 8 bits
- 1 apontador de programa de 8 bits
- 1 registrador de estado com 3 códigos de condição: negativo, zero e carry

# Arquitetura Ramses



# Conjunto de Instruções RAMSES

Código	Instrução	Operação
0000	NOP	Nenhuma operação
0001	STR r,end	Armazena registrador "r" no endereço "end" da memória
0010	LDR r,end	Carrega o registrador "r" com o conteúdo do endereço "end" da memória
0011	ADD r,end	Soma o conteúdo do endereço "end" da memória ao registrador "r"
0100	OR r,end	Efetua operação lógica "OU" do conteúdo do endereço "end" da memória ao registrador "r"
0101	AND r,end	Efetua operação lógica "E" do conteúdo do endereço "end" da memória ao registrador "r"
0110	NOT r	Inverte todos os bits do registrador "r"
0111	SUB r,end	Subtrai o conteúdo do endereço "end" da memória ao registrador "r"
1000	JMP end	Desvio incondicional para o endereço "end" da memória
1001	JN end	Desvio condicional, se "N=1", para o endereço "end" da memória
1010	JZ end	Desvio condicional, se "Z=1", para o endereço "end" da memória
1011	JC end	Desvio condicional, se "C=1", para o endereço "end" da memória
1100	JSR end	Desvio para subrotina no endereço "end" da memória
1101	NEG r	Troca o sinal do registrador "r"
1110	SHR r	Desloca o registrador "r" um bit para a direita
1111	HLT	Para o ciclo de busca-decodificação-execução

## Códigos das instruções

<b>NOP</b>	0					<b>JMP</b>	128	end
<b>STR</b>	16	r	end			<b>JN</b>	144	end
<b>LDR</b>	32	r	end			<b>JZ</b>	160	end
<b>ADD</b>	48	r	end			<b>JC</b>	176	end
<b>OR</b>	64	r	end			<b>JSR</b>	192	end
<b>AND</b>	80	r	end			<b>NEG</b>	208	r
<b>NOT</b>	96	r				<b>SHR</b>	224	r
<b>SUB</b>	112	r	end			<b>HLT</b>	240	

Modo:

0: Dir: n  
1: Ind: n,l  
2: Ind: #n  
3: Idx: n,X

Registrador:

0: A 2: X  
1: B 3: ?

Daedalus Assembler

Arquivo

Editar

Visualizar

Ajuda

Máquina Alvo

☐ Ahmes

☒ Ramses

☐ Cesar

Salvar

Executar Montagem

Código Fonte:

Início: ORG 0  
LDR A Valor1 ; Carrega A com o primeiro valor  
ADD A Valor2 ; Adiciona valor2 ao registrador A  
STR A Result ; Armazena resultado  
HLT  
  
ORG 128 ; Define area de dados começando no end. 128  
  
Valor1: DB H10  
Valor2: DB H80  
Result: DB 0

Código gerado sem erros

Linha: 7

Coluna: 25

Mapa da Mem...

Visualizar

End.	Valor
0	20
1	80
2	30
3	81
4	10
5	82
6	FF
7	0
8	0
9	0
A	0
B	0
C	0
D	0
E	0

Códigos das instruções

<b>NOP</b>	0					<b>JMP</b>	128	end	
<b>STR</b>	16	r	end			<b>JN</b>	144	end	
<b>LDR</b>	32	r	end			<b>JZ</b>	160	end	
<b>ADD</b>	48	r	end			<b>JC</b>	176	end	
<b>OR</b>	64	r	end			<b>JSR</b>	192	end	
<b>AND</b>	80	r	end			<b>NEG</b>	208	r	
<b>NOT</b>	96	r				<b>SHR</b>	224	r	
<b>SUB</b>	112	r	end			<b>HLT</b>	240		

Modo:

0: Dir: n  
1: Ind: n,l  
2: Ind: #n  
3: Idx: nX

Registrador:

0: A 2: X  
1: B 3: ?