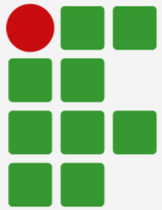


INSTITUTO FEDERAL
Sul-rio-grandense | Câmpus Venâncio Aires

Arquitetura e Organização de Computadores

Professor: Fernando Luís Herrmann
E-mail: fernandoherrmann@ifsul.edu.br

Tecnólogo em Análise e Desenvolvimento de Sistemas 2022

Material de aula:

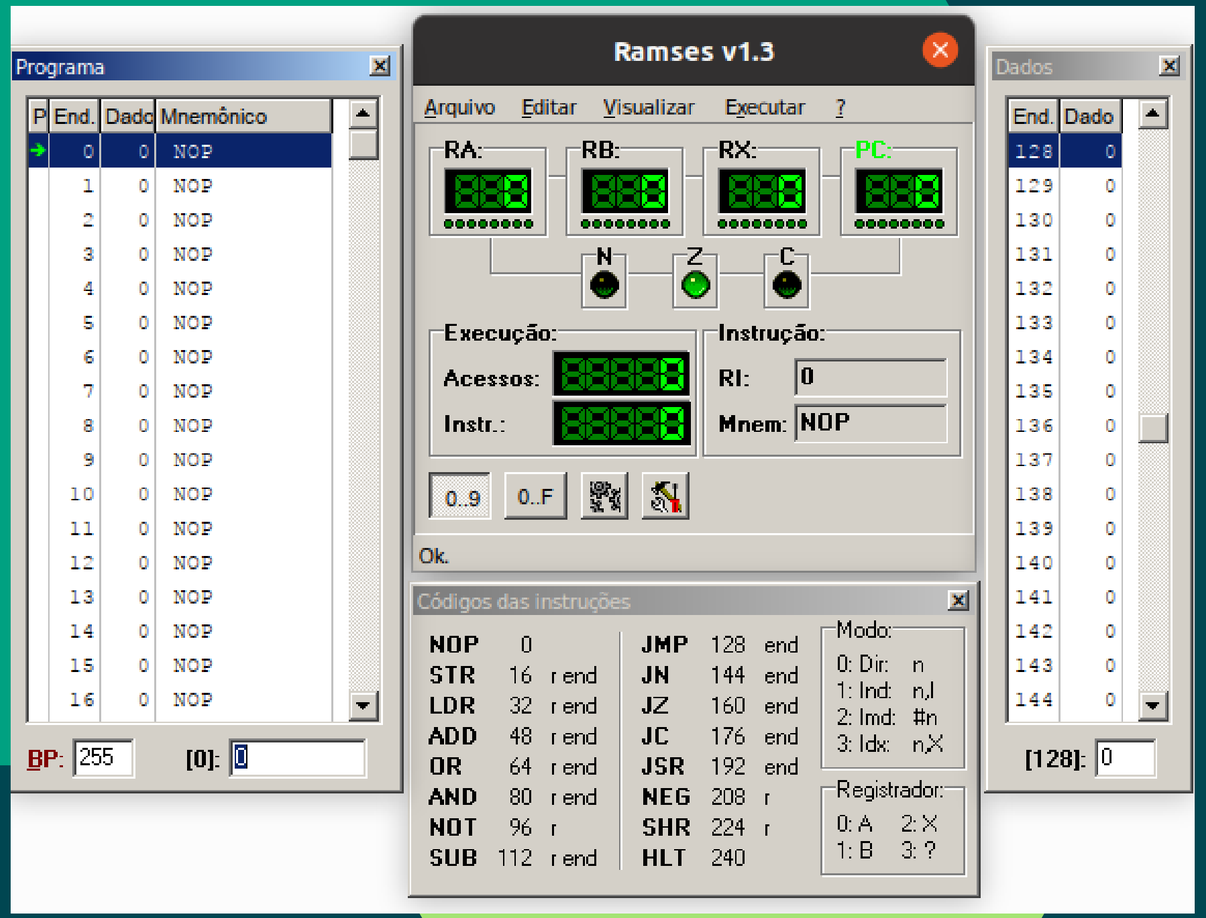
https://github.com/herrmannfl/tads-aoc-2022

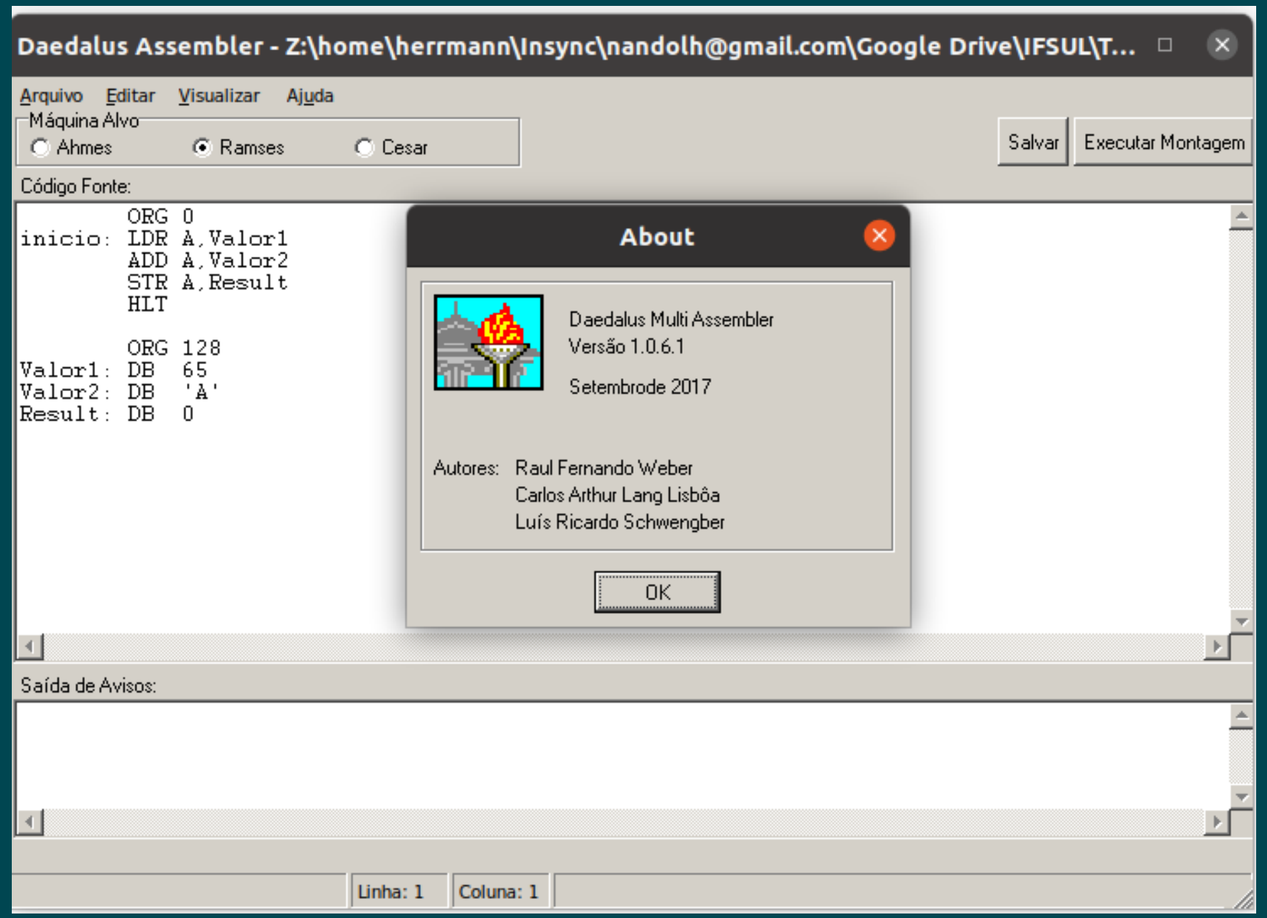
Arquiteturas Didáticas

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) REM, RDM
RAMSES	8 bits 256 bytes	8 bits	Modos de endereçamento 4 modos x 16 instruções	PC, IR, RA, RB, RX, Flags (N, Z, C) REM, RDM
CESAR	16 bits 64 KBytes	16 bits	Inúmeras	R0 a R6 (uso geral) R7 (PC)

Simulador Ramses

Montador DAEDALUS

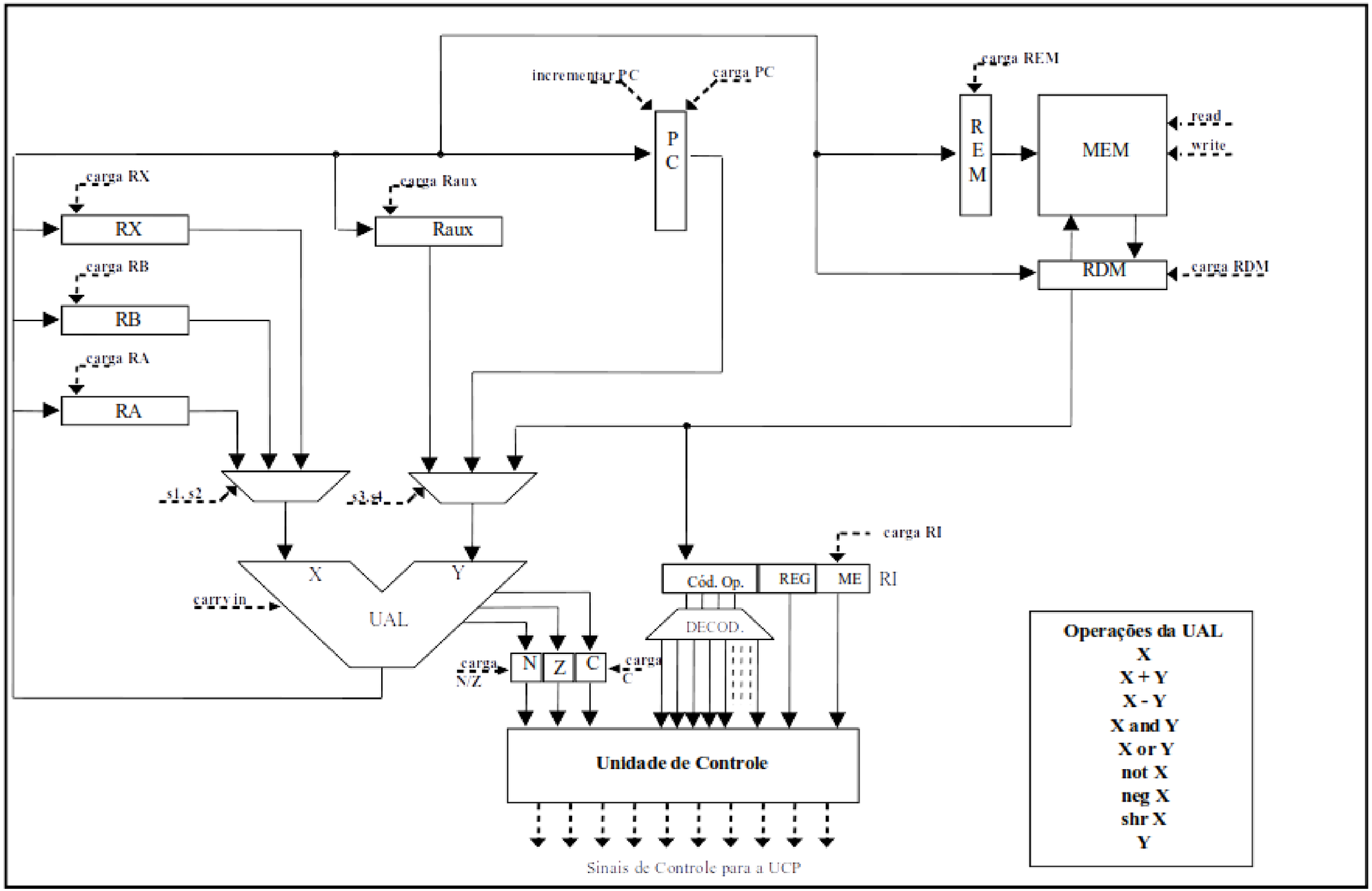




Características gerais

- 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

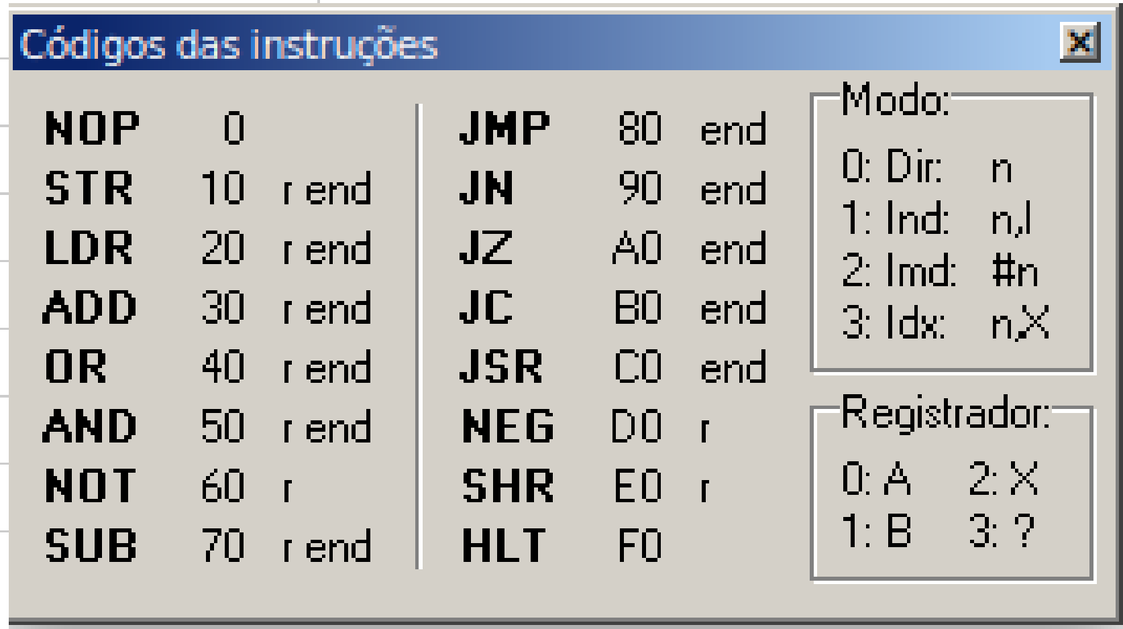
Exemplo:

0010 0100

0010 - LDR

01 - Registrador

00 - Modo



Daedalus Assembler

Arquivo Editar Visualizar Ajuda

Máquina Alvo
Ahmes Ramses Cesar

Salvar | Executar Montagem

Código Fonte:

```
        ORG 0
Inicio: 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 comecando 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

Códigos das instruções

NOP	0		JMP	80	end
STR	10	r end	JN	90	end
LDR	20	r end	JZ	A0	end
ADD	30	r end	JC	B0	end
OR	40	r end	JSR	C0	end
AND	50	r end	NEG	D0	r
NOT	60	r	SHR	E0	r
SUB	70	r end	HLT	F0	

Modo:
0: Dir: n
1: Ind: n,I
2: Imd: #n
3: Idx: n,X

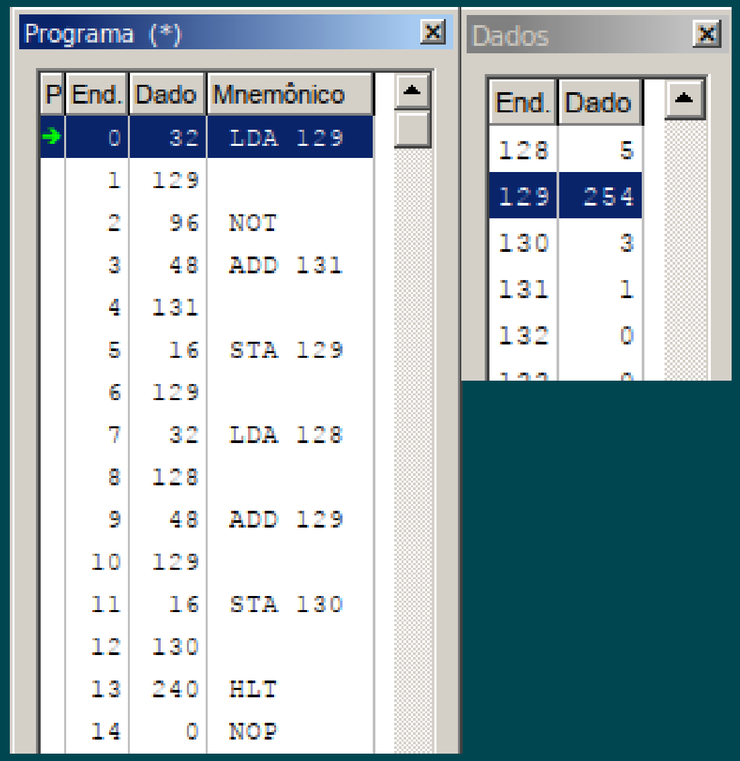
Registrador:
0: A 2: X
1: B 3: ?

Modos de Endereçamento

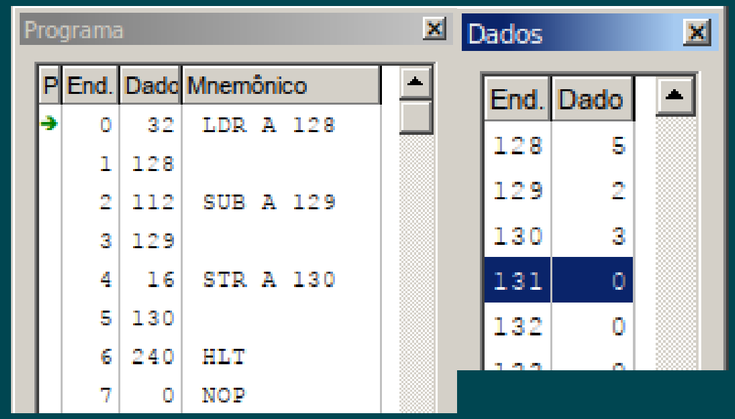
```
OpCode Oper R ME  Mnemônico       Modo Enedereçamento
  20 = 0010 0000  LDR A,Valor   - Modo Direto   A
  22 = 0010 0010  LDR A,#00     - Modo Imediato A
  24 = 0010 0100  LDR B,Valor   - Modo Direto   B
  26 = 0010 0110  LDR B,#00     - Modo Imediato B
  2A = 0010 1010  LDR X,#00     - Modo Imediato X

  20 = 0010 0000  LDR A,Valor   - Modo Direto   A
  21 = 0010 0001  LDR A,Ptr,I   - Modo Indireto A
  22 = 0010 0010  LDR A,#00     - Modo Imediato A
  23 = 0010 0011  LDR A,Vet,X   - Modo Indexado A com X
```

Questão 1. Implemente um código Assembly que seja capaz de efetuar uma subtração. As posições de memória 128 e 129 devem conter, respectivamente, o minuendo e o subtraendo. O resultado deverá ser armazenado na posição 130.

NEANDER



RAMSES



RAMSES

Questão 2. Implemente um código que seja capaz de calcular o endereço da rede a qual um IPv4 pertence dados o IPv4 e a Máscara. A memória de dados deve ser organizada da seguinte forma:
Palavras 128, 129, 130 e 131 – Bytes do IPv4
Palavras 132, 133, 134, 135 – Bytes da Máscara
Palavras 136, 137, 138, 139 – IP da rede calculado

Código Fonte:

```
          ORG 0
inicio:   LDR X, #0
loop:     LDR A, IPv4,X
          AND A, Mascara, X
          STR A, Result, X
          ADD X, #1
          STR X, Temp
          SUB X, #4
          JZ fim
          LDR X, Temp
          JMP loop
fim:      HLT

          ORG 128
IPv4:     DAB 10,10,10,1
Mascara:  DAB 255,0,0,0
Result:   DAB 0,0,0,0
Temp:     DB 0
```