
Facens

ARQUITETURA DE COMPUTADORES – LABORATÓRIO

André Breda Carneiro
Sidney José Montebeller
Fernando Deluno Garcia
Rafael Rodrigues Da Paz

Facens

Experiência Nº 5 – AMBIENTE DE SIMULAÇÃO – CPU BOARD.

Objetivos:

- Adquirir conhecimentos em arquitetura de computadores;
- Familiarização com o módulo CPU board studio e CPU board Lab;
- Instruções básicas usando linguagem assembly;
- Manipulação de alguns periféricos.

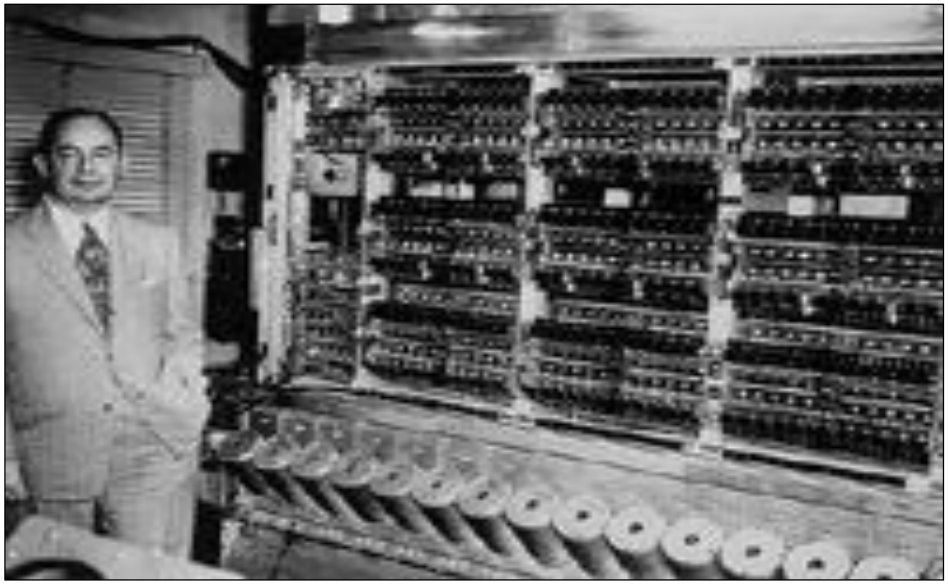
Facens

A máquina de von Neumann

1945 - First draft of a report on the EDVAC.

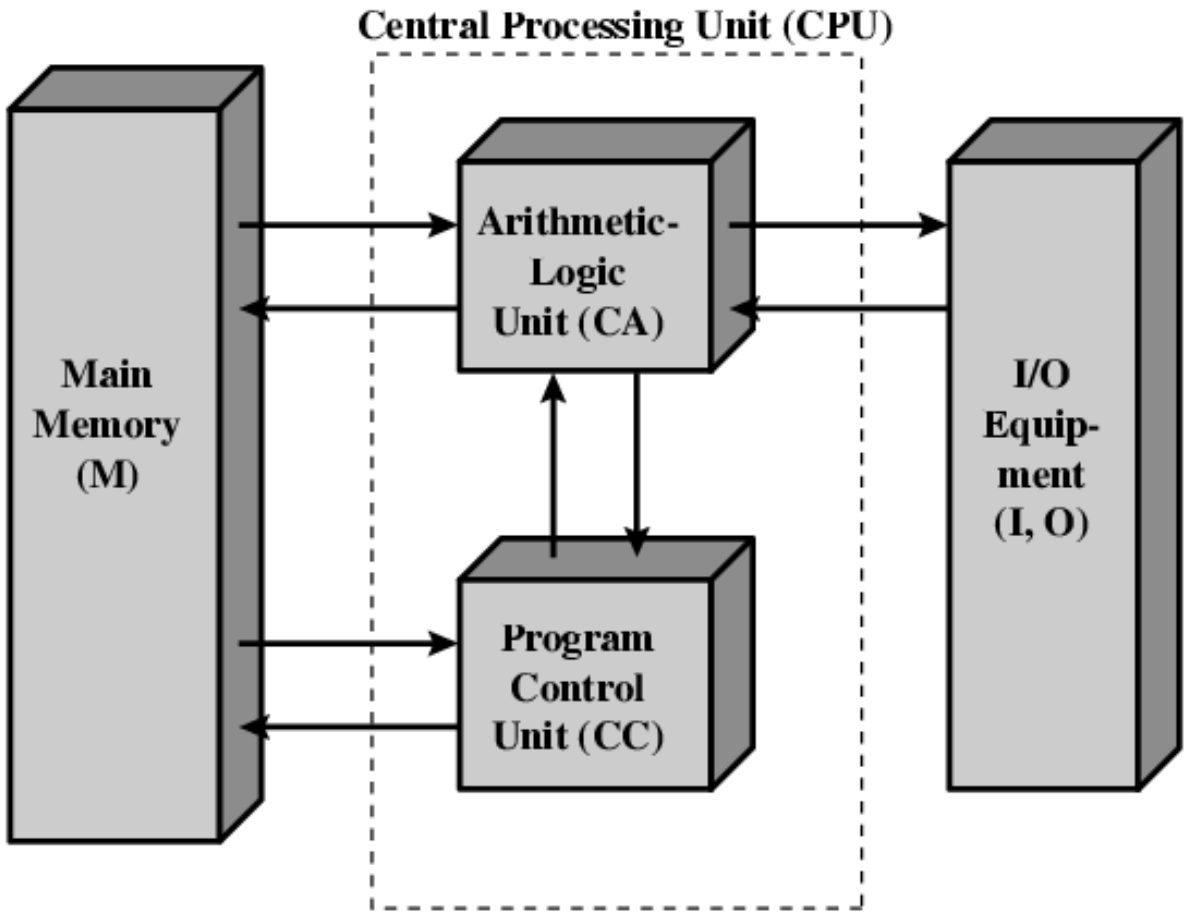
Conceito de programa armazenado.

O computador IAS e a máquina de von Neumann.



Facens

Estrutura da Máquina de von Neumann



Facens

Unidade Central de Processamento

A sequência de operações realizadas por um computador é determinada a partir da unidade central de processamento.

É dividida em componentes que executam funções específicas como o caso da unidade de controle e a unidade lógica e aritmética.

Unidade Central de Processamento

A unidade de controle é responsável por interpretar e executar as instruções armazenadas na memória, além de gerar sinais de controle para os outros módulos do computador.

A unidade lógica e aritmética é utilizada para realizar operações sobre os dados.

Facens

Memória

Um dos principais componentes da arquitetura de von Neumann é a memória principal, onde são armazenadas as instruções do programa e os dados utilizados durante a sua execução.

Instruções e dados ocupam o mesmo espaço de endereços.

Dispositivos de entrada e saída

Módulo responsável por manter contato da CPU com dispositivos externos.

Facens

Interconexão

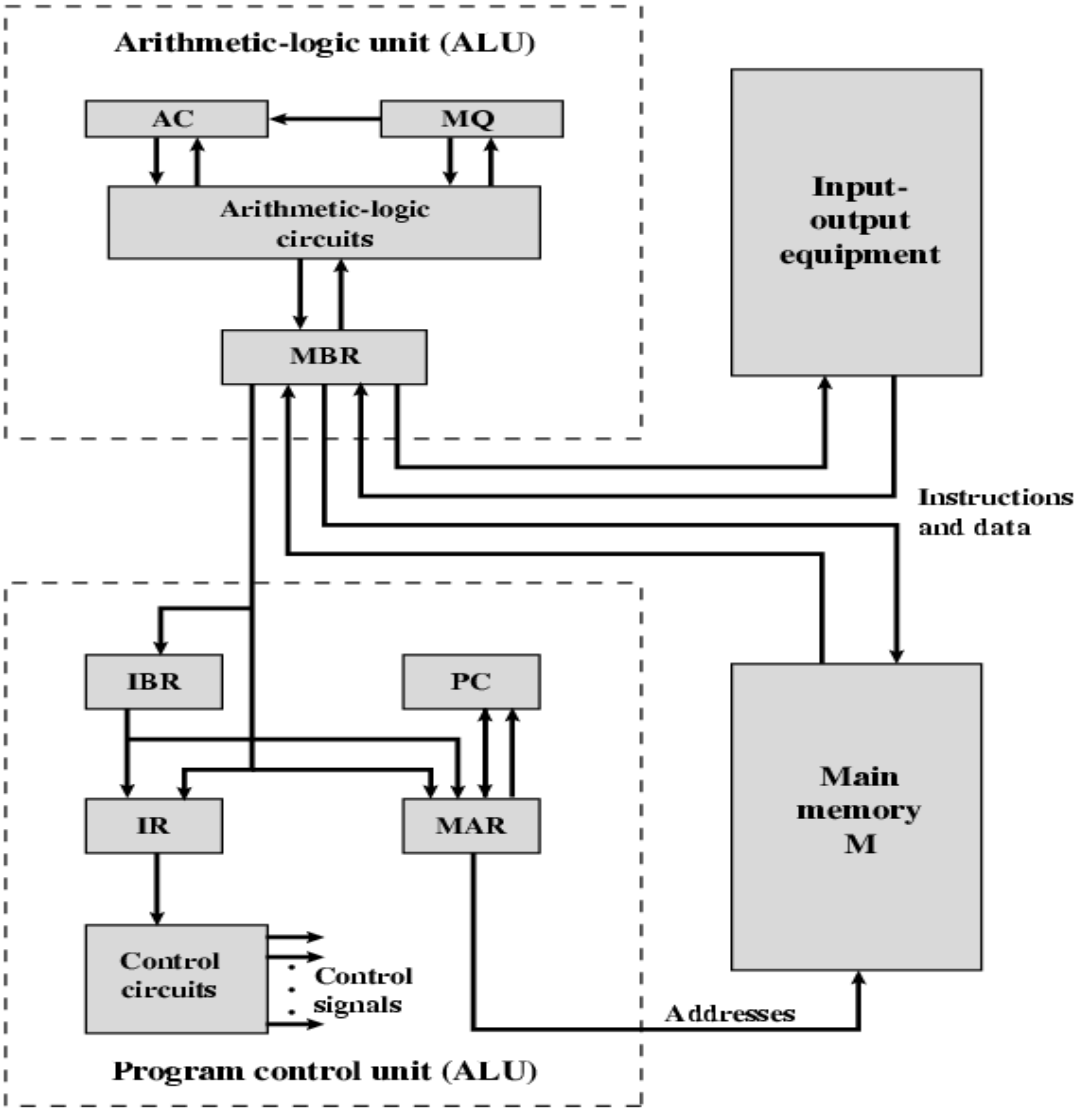
Os elementos da máquina de von Neumann estão conectados através de barramentos, que são os canais de comunicação.

Os barramentos possuem funções específicas:

- Barramento de endereços;
- Barramento de dados;
- Barramento de sinais de controle.

Facens

Estrutura geral do IAS



Facens

Registradores

AC – Acumulador

PC – Contador de programa

IR – Registrador de Instrução

MAR – Registrador de Endereço de Memória

MBR – Registrador de Dados de Memória

Facens

Programa - Instrução

Uma instrução é um valor binário (sequência de bits) armazenado em memória.

Essa sequência de bits determina para a CPU informações necessárias para a execução de uma operação.

Essas informações podem ser representadas por um único bit ou um conjunto de bits.

O formato de instrução determina o significado de cada bit ou conjunto de bits.

Facens

Tipos de instruções

Movimentação de dados entre CPU/Memória e Memória/CPU.

Operações aritméticas e lógicas.

Desvios condicionais e incondicionais.

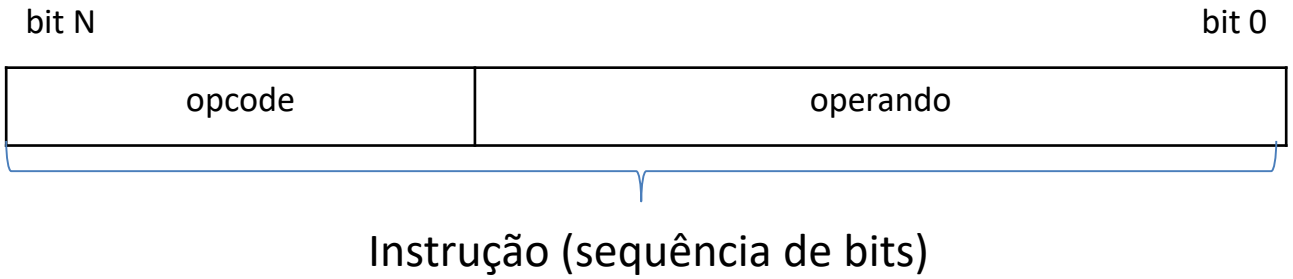
Controle e testes.

Facens

Formato de instrução

O elemento que determina a operação que será realizada é chamado de ***opcode***.

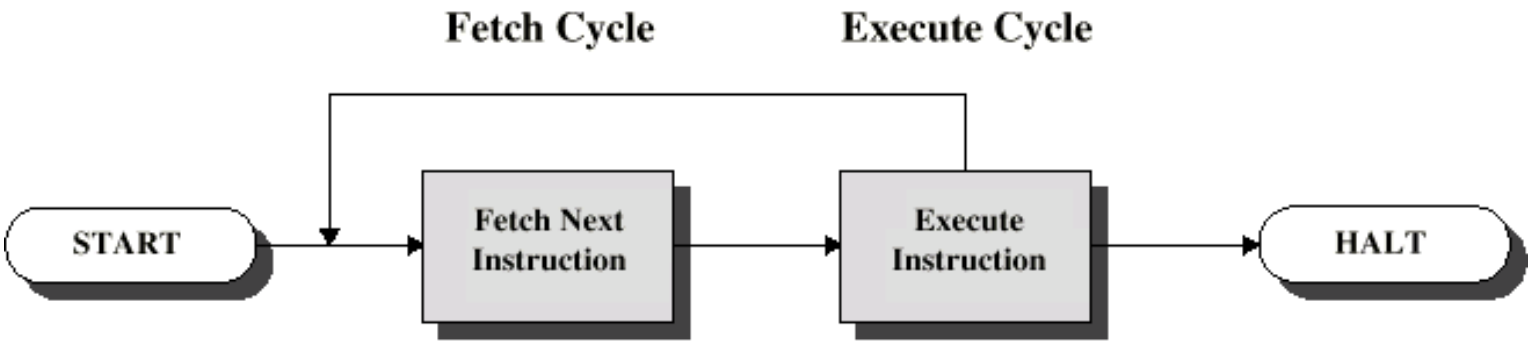
Um formato simples de uma instrução pode ser representada pelo código da operação e um operando.



Facens

Ciclo de instrução

Basicamente o ciclo de instrução da CPU é dividido em duas etapas: busca (*fetch cycle*) e execução (*execute cycle*).



Facens

O ciclo de busca

A primeira etapa é responsável por buscar uma instrução na memória principal.

A instrução é transferida da memória principal para o registrador de instruções.

A instrução armazenada é decodificada para que a unidade de controle execute a operação da instrução.

O ciclo de execução

A segunda etapa é responsável por realizar a execução da instrução.

Nessa etapa as operações realizadas dependem da instrução decodificada.

Facens

Módulo Didático CPU Board

Facens

Registradores da CPU Board

Registrador de endereço (MAR) de 12 bits.

Registrador de dados de (MBR) de 8 bits.

Contador de programa (PC) de 12 bits.

Registrador de instruções (IR) de 16 bits.

Registrador acumulador (AC) de 8 bits.

Registrador com ***flags de controle*** (SF, OF, ZF, GF e LF).

Facens

Conjunto de instruções

Tipos de operações:

- Movimentação de dados entre CPU/Memória e Memória/CPU.
- Operações aritméticas e lógicas.
- Desvios condicionais e incondicionais.
- Controle e testes.

Instruções de tamanho fixo: 2 bytes

- 4 bits para o opcode.
- 12 bits para o operando.

Facens

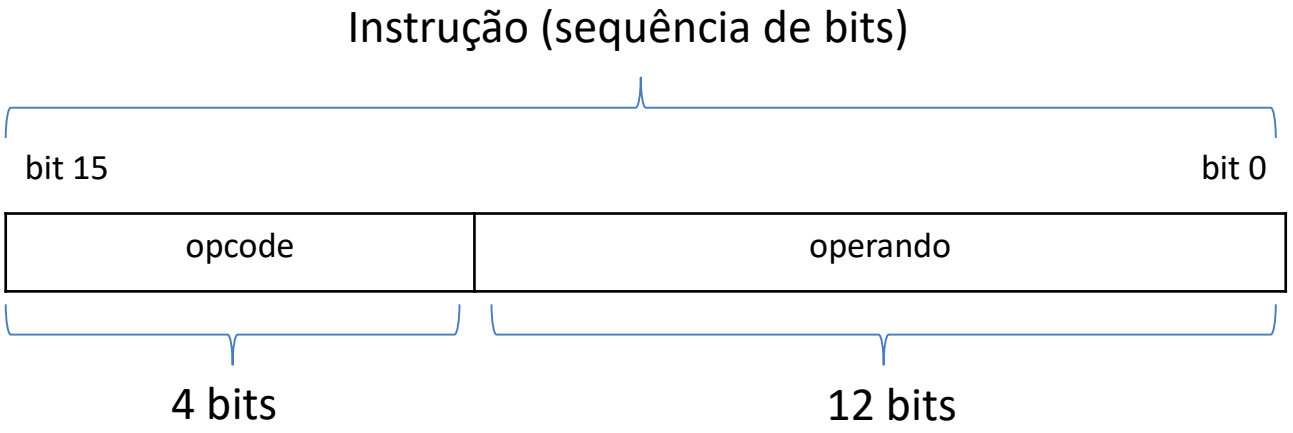
Conjunto de instruções

Mnemônico	Descrição	Opcode	Operação
LDA	Load Accumulator	0000	AC = M[Endereço]
STR	Store	0001	M[Endereço] = AC
ADD	Add	0010	AC = AC + M[Endereço]
SUB	Subtract	0011	AC = AC – M[Endereço]
AND	Logical AND	0100	AC = AC & M[Endereço]
OR	Logical OR	0101	AC = AC | M[Endereço]
NOT	One's Complement	0110	AC = ~AC
XOR	Exclusive OR	0111	AC = AC ^ M[Endereço]
JMP	Indirect Jump	1000	Desvio indireto
JE	Jump if Equal	1001	Desvia se ZF = 1
JL	Jump if Less Than	1010	Desvia se LF = 1
JG	Jump if Greater Than	1011	Desvia se GF = 1
JLE	Jump if Less or Equal	1100	Desvia se ZF = 1 ou LF = 1
JGE	Jump if Greater or Equal	1101	Desvia se ZF = 1 ou GF = 1
HLT	Halt	1110	Para o processamento
NOP	No Operation	1111	Sem operação

Facens

Formato de instrução

Instrução com tamanho de 2 bytes e composta por um opcode e um operando.



O operando representa o endereço de memória que será utilizado no ciclo de execução da instrução.

Facens

Modo de endereçamento

O modo de endereçamento utilizado é o direto, onde o operando corresponde ao endereço real da memória principal.

Unidade lógica e aritmética

Qualquer operação realizada na ULA atualiza as flags de controle.

As operações aritméticas utilizam números sinalizados.

No módulo CPU Board a comparação de dois valores deve ser realizada a partir da operação de subtração.

As flags podem ser utilizadas posteriormente nas instruções de desvio condicional.

Facens

Unidade lógica e aritmética

Flag	bit	Significado
SF (do inglês, sign flag)	1	Resultado é negativo
	0	Resultado é positivo
ZF (do inglês, zero flag)	1	Resultado é zero
	0	Resultado é diferente de zero
GF (do inglês, greater than flag)	1	Indica que na última operação AC > AC2
LF (do inglês, less than flag)	1	Indica que na última operação AC < AC2
OF (do inglês, overflow flag)	1	Indica que na última operação ocorreu overflow
	0	indica que na última operação não ocorreu overflow

Facens

Detalhes do hardware

Memória RAM de 4kBytes:

- 4096 palavras de 8 bits.

Lógica de seleção de endereços:

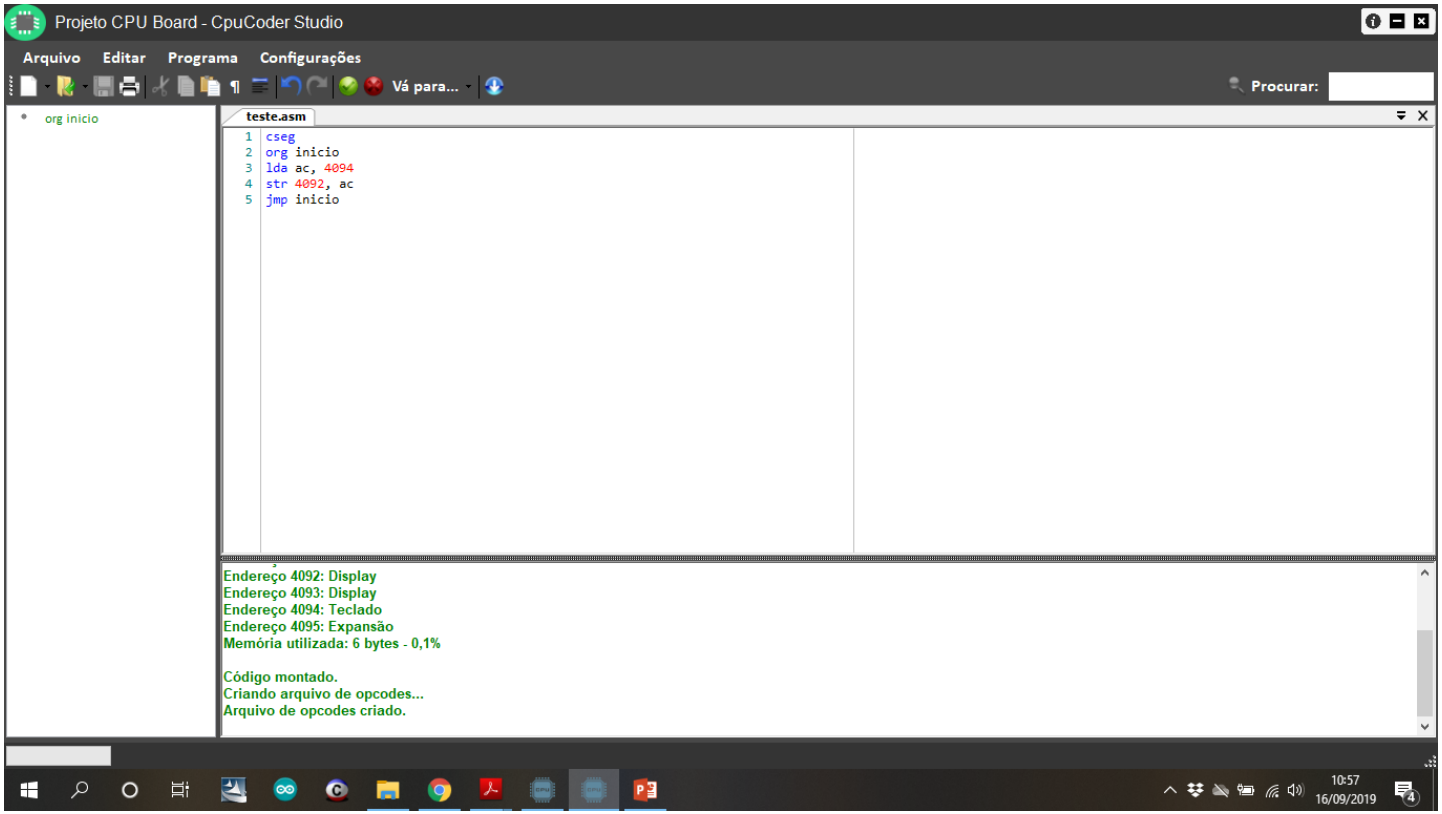
- Endereço 0~4091: memória RAM.
- Endereço 4092: display linha 1.
- Endereço 4093: display linha 2.
- Endereço 4094: chaves.
- Endereço 4095: barramento de expansão.

Facens

Primeiro passo (CPU Studio):

Baixar o arquivo: **Software CPU.zip**

Descompactar e abrir a pasta **CpuStudio**, dentro da pasta executar o arquivo: **CPUStudio.exe**

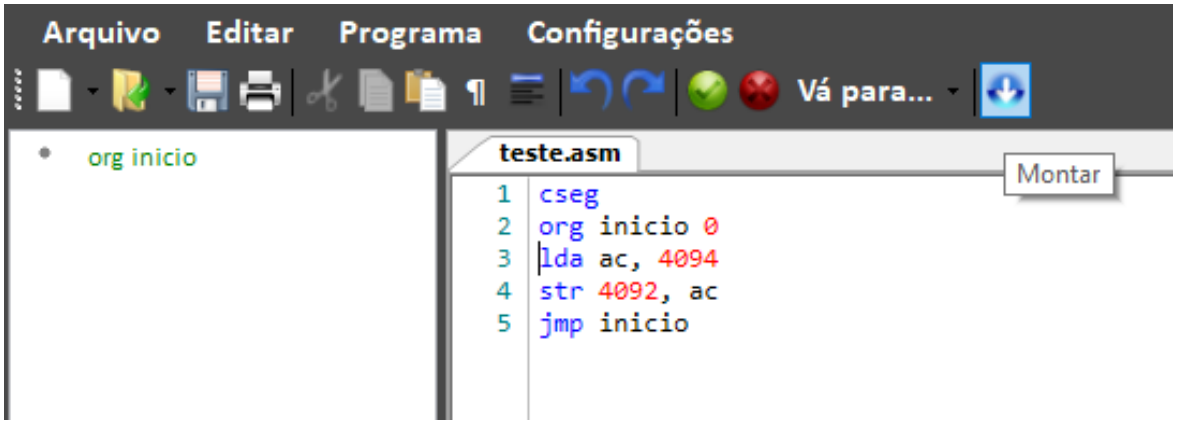


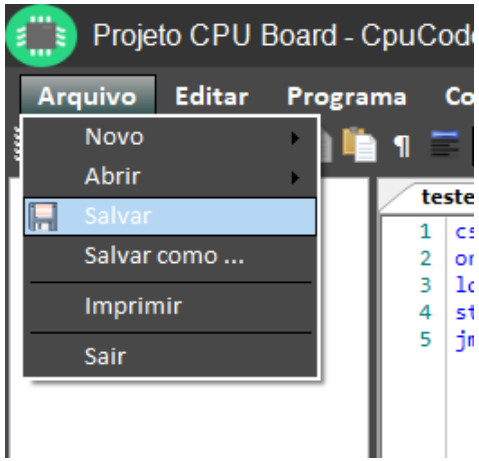
Facens

Primeiro passo (CPU Studio):

Escrever o código, montar e salvar o arquivo .asm em uma pasta criada somente para ele.

Feito isso, ir para o segundo passo.

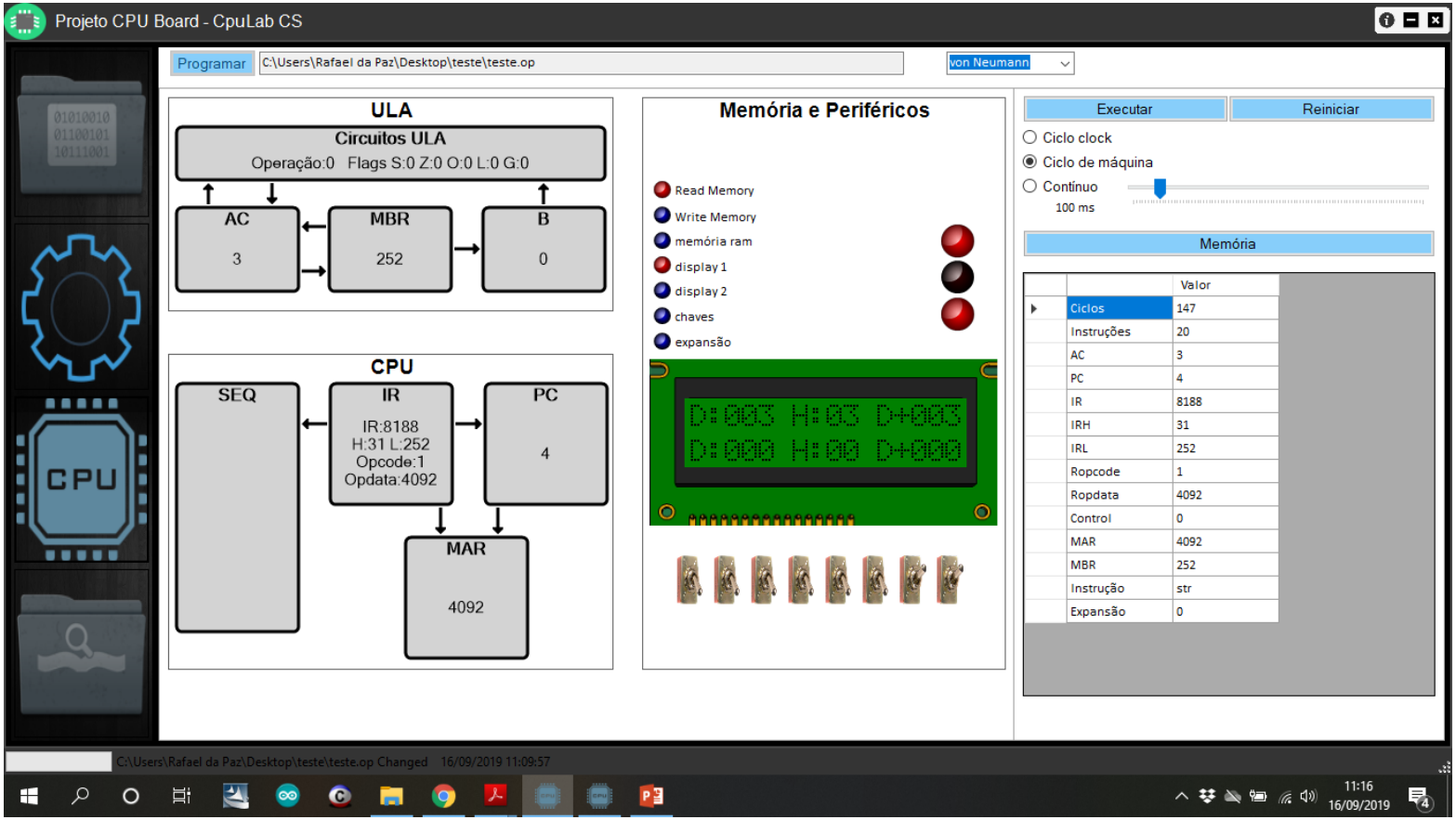




Facens

Segundo passo (CPU Lab):

Abrir a pasta **CpuLab**, dentro da pasta executar o arquivo: **CPULab.exe**

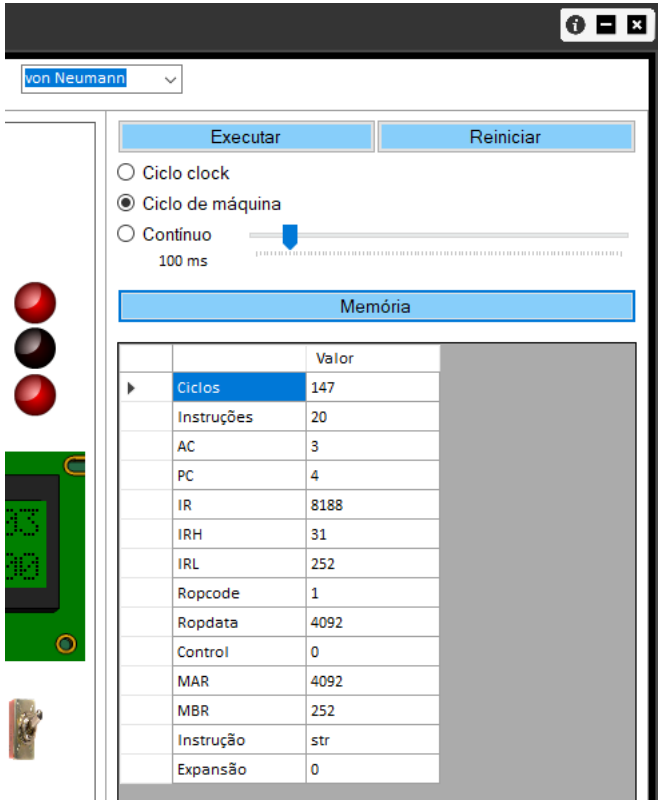


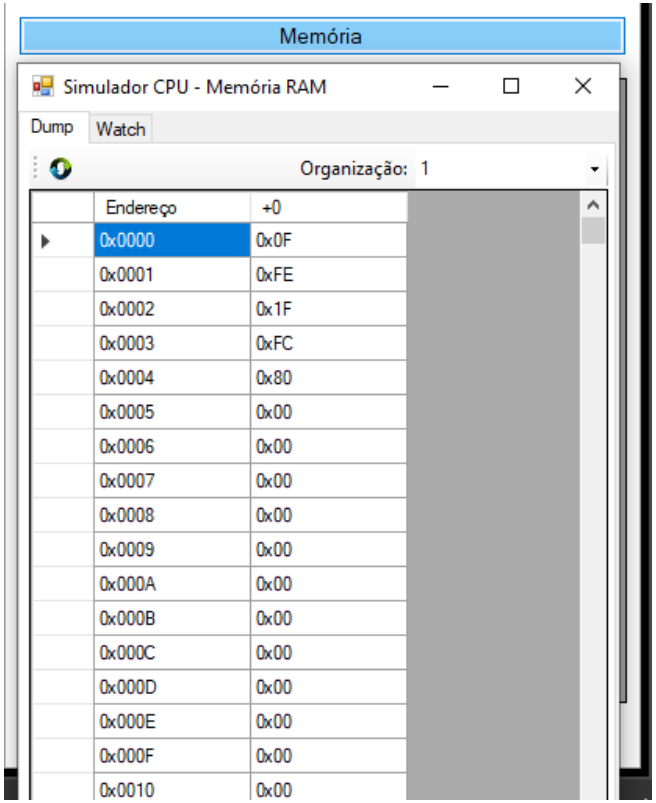
Facens

Segundo passo (CPU Lab):

Abrir a pasta **CpuLab**, dentro da pasta executar o arquivo: **CPULab.exe**

Configurar a tela no modo von Neumann e escolher as opções de execução do programa:





Facens

Rodar o exemplo:

Movimentação de dados e desvio incondicional

```
//região de dados

ram 500 25  //armazena o valor 25 no endereço 500

//região de instruções

cseg          //marca o início do segmento de código
org Inicio 0  //determina que a instrução após o org está no endereço 0
lda ac,500    //AC = M[500]
str 4092,ac   //M[4092] = AC (display linha 1)
lda ac,4094   //AC = M[4094] (chaves)
str 4093,ac   //M[4093] = AC (display linha 2)
jmp Inicio    //PC = inicio -> PC = 0, retorna para o início do programa
```

Facens

Rodar o exemplo:

Instruções lógicas e aritméticas

```
//região de dados

ram 300 0    //contador
ram 301 1    //incremento

//região de instruções

cseg            //marca o início do segmento de código
org Inicio 0    //determina que a instrução após o org está no endereço 0
lda ac,300      //AC = M[300]
add ac,301      //AC = AC + M[301]
str 300,ac      //M[300] = AC
str 4092,ac     //M[4092] = AC (display linha 1)
not ac          //AC = ~AC (invete todos os bits)
str 4093,ac     //M[4093] = AC (display linha 2)
jmp Inicio      //PC = inicio -> PC = 0, retorna para o início do programa
```

Facens

Rodar o exemplo:

Instrução de desvio condicional

```
//região de dados

ram 100 20 //numA
ram 101 10 //numB
ram 103 5  //display

//região de instruções

cseg                    //marca o início do segmento de código
org Inicio              //determina que a instrução após o org está no endereço 0
lda ac,100              //AC = M[100]
sub ac,101              //AC = AC - M[101]  -> comparação do valor numA com numB
je BLOCO_IF_INICIO      //Salta para o endereço BLOCO_IF_INICIO se numA == numB
jmp BLOCO_IF_FIM        //Salta para o endereço BLOCO_IF_FIM

org BLOCO_IF_INICIO
lda ac,100              //AC = M[100]
str 103,ac              //M[300] = AC
org BLOCO_IF_FIM
lda ac,103              //AC = M[103]
str 4092,ac             //M[4092] = AC (display linha 1)
hlt                     //fim do programa
```

Facens

Relatório 5

•Rodar os 3 exemplos apresentados;

•Criar um relógio que conte os minutos “00 até 59” e os segundos “00 até 59”, quando chegar no final 59 59, recomeçar;

Contemplar no relatório:

•Introdução;

•Procedimento experimental executado;

•Análise da arquitetura;

• Conclusão.

Links para os vídeos explicativos do módulo CPU board, com o prof. Fernando D. Garcia:

https://www.loom.com/share/ae3c9d70a2394a55be1ff80c35ef4246
https://www.loom.com/share/0be7f72841764175807dc4220be215b8

Referências

- John von Neumann ao lado do EDVAC. Fonte: Bit by Bit (Haverford College).
- STALLINGS, W. **Arquitetura e organização de computadores**. Pearson Prentice-Hall, 8ª ed, São Paulo. 2010.

Facens