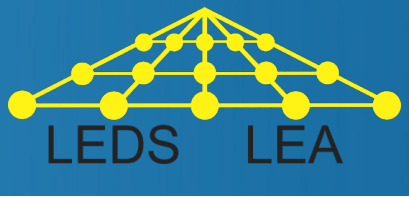
Arquitetura e Organização de Computadores

Conjunto de Instruções da Arquitetura – CompSim



Agenda

- Tipos de Instruções
- Ciclo de Instrução
- Operações Básicas de Acesso à Memória

Tipos de Instruções

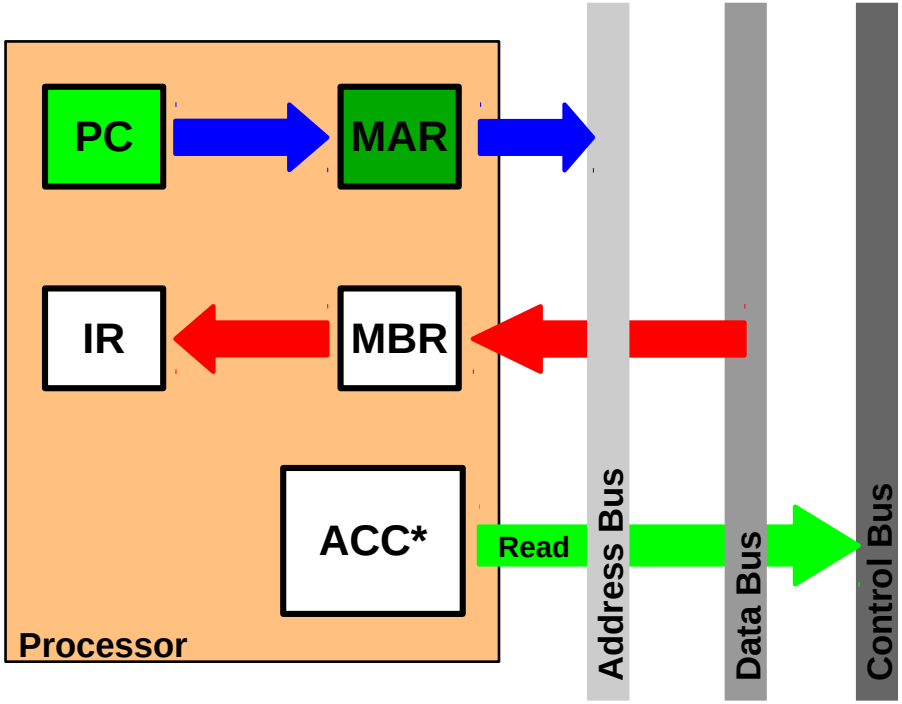
- Pseudo-Instruções do Montador (*Assembler*)
  - Segmento
    - .code, .data, .bss, .stack
  - Rótulo ou Nome
    - :
  - Delimitador de comentário
    - ;
  - Definição/Declaração de variáveis
    - DD, DB, RESD, RESB
- Conjunto de Instruções da Arquitetura (ISA)
  - Aritméticas
    - ADD, SUB
  - Lógicas
    - NAND, SHIFT
  - Transferência de dados
    - MOV, LDA, STA, LDI, STI, SOP
  - Transferência de controle
    - JMP, JN, JZ, CALL, RET, INT
  - Entrada/Saída
    - INT

Ciclo de Instrução

- Vimos anteriormente os registradores CS, DS, SS e SP.
- Além desses, temos:
  - PC (*Program Counter*) – Endereça a próxima instrução que será executada;
  - MAR (*Memory Address Register*) – Endereça uma posição de memória em operações de leitura e escrita;
  - MBR (*Memory Buffer Register*) – Armazena um dado que será escrito ou lido da memória.
  - AC (*Accumulator*) – Registrador de propósito geral. Pode ser utilizado em diferentes tipos de contexto;
  - IR (*Instruction Register*) – Armazena a instrução que está sendo executada. O processo de decodificação de uma instrução inicia no IR.

Ciclo de Instrução

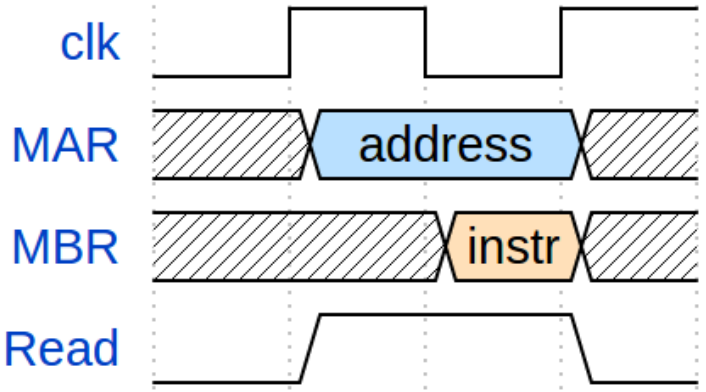
- **Fetch** (Busca de Instrução)



*Access Control Circuit

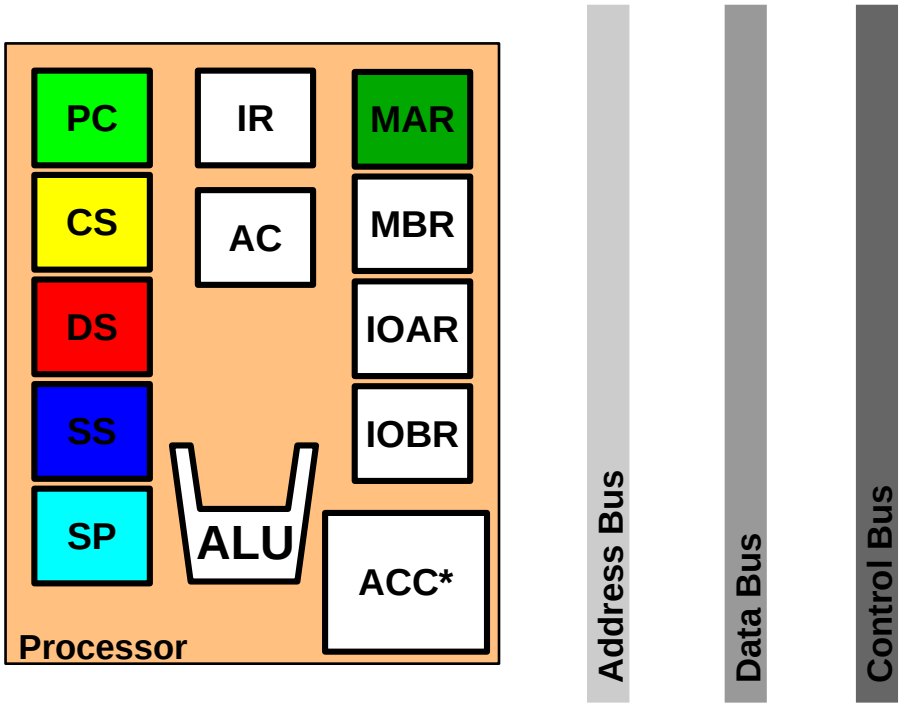
- Para a busca de uma intrução na memória:
  - O endereço em PC é copiado para MAR;
  - A memória recebe o endereço de MAR, através do barramento de endereços;
  - A memória envia a instrução solicitada para o barramento de dados;
  - A instrução é copiada para MBR, o qual copia-a para IR.

Waveform



Ciclo de Instrução

- **Execução** de Instrução



*Access Control Circuit

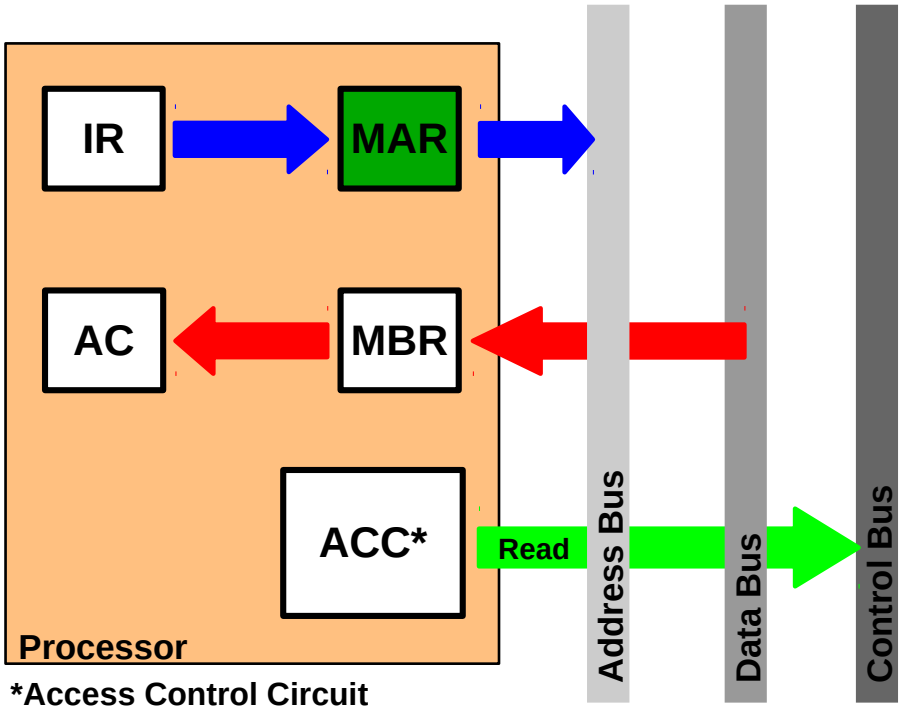
- A execução das instruções não segue um modelo fixo.
  - Podem (ou não) ser envolvidos o registrador de propósito geral (AC) e os específicos.
  - Em caso de operandos externos ao processador, pode haver novos acessos à memória ou aos periféricos;
  - Pode ainda utilizar a ULA.

Operações Básicas de Acesso à Memória

- O acesso básico à memória pode ser realizado através de duas instruções:
  - LDA (*LoaD from Address*) – Leitura de um dado a partir de um determinado endereço de memória.
  - STA (*Store To Address*) – Escrita de um dado em um determinado endereço de memória.
- Para tanto, utilizam o AC como destino (LDA) ou fonte (STA) de dados.
- Sintaxe:
  - Leitura: [<rotulo>] LDA <endereco-memoria>
  - Escrita: [<rotulo>] STA <endereco-memoria>

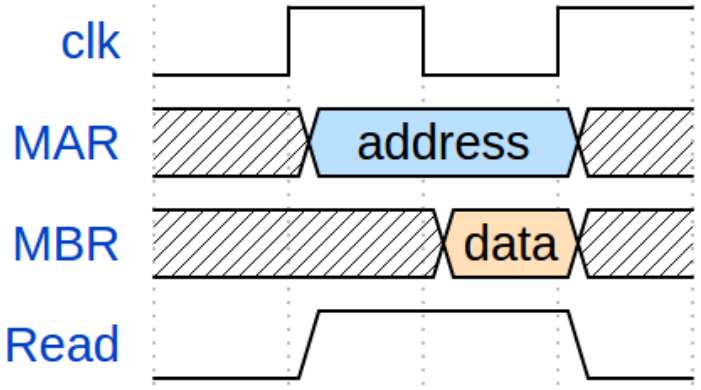
Operações Básicas de Acesso à Memória

- LDA (*LoaD from Address*)



*Access Control Circuit

Waveform



Microcode (RTN*)

MAR ← IR[11–0]

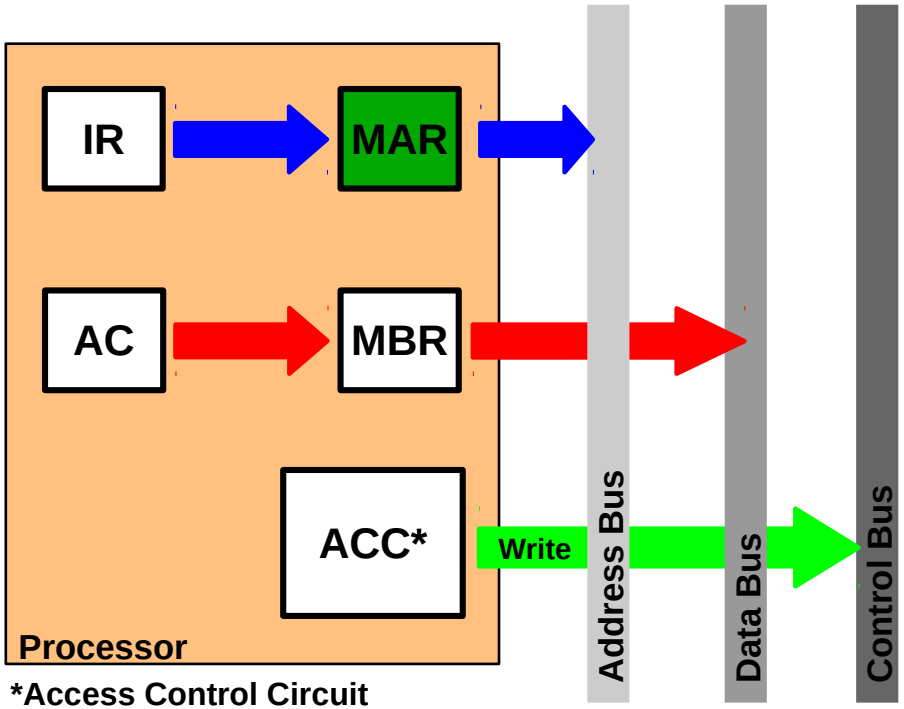
MBR ← M[MAR]

AC ← MBR

*Register Transfer Notation

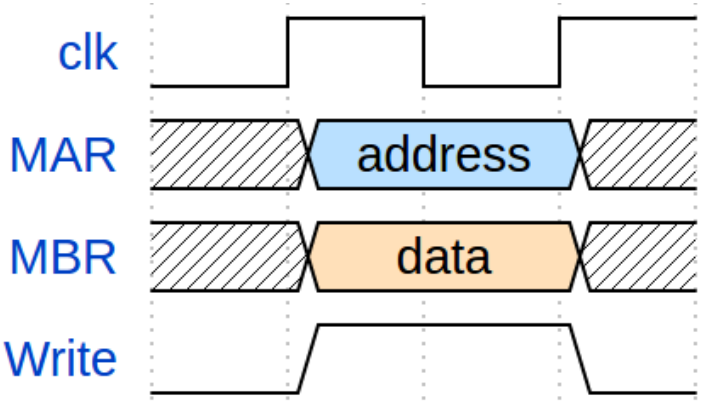
Operações Básicas de Acesso à Memória

- STA (*Store To Address*)



*Access Control Circuit

Waveform



Microcode (RTN*)

MAR ← IR[11–0]

MBR ← AC

M[MAR] ← MBR

*Register Transfer Notation

Operações Básicas de Acesso à Memória

- Exemplo prático: Acessando a memória
- Procedimento:
  - Baixar e extrair o pacote:
    - 2.basic_memory_access.zip
  - Menu “File” → “Open”
    - Ou Teclas “Ctrl+o”
  - Arquivo “accessing_memory.asm”

```
1   .code
2
3           ; Le valor de a para AC
4           LDA a
5
6           ; Grava valor de AC em b
7           STA b
8
9           ;finaliza programa
10  end:
11          INT exit
12
13  .data
14          ;int a = 10;
15          a: DD 10
16
17          ; syscall exit
18          exit: DD 25
19
20          ; secao .bss eh opcional
21  .bss
22          ;int b;
23          b: RESD 1
24
25  ; pilha de 10 palavras
26  .stack 10
27
28
29
```