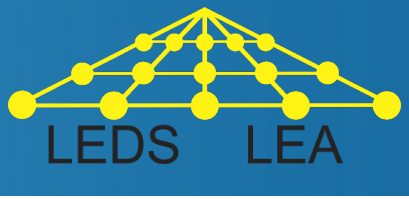
Arquitetura e Organização de Computadores

Conjunto de Instruções da Arquitetura – CompSim



Agenda

- Tipos de Instruções
- Instruções Lógicas

Tipos de Instruções

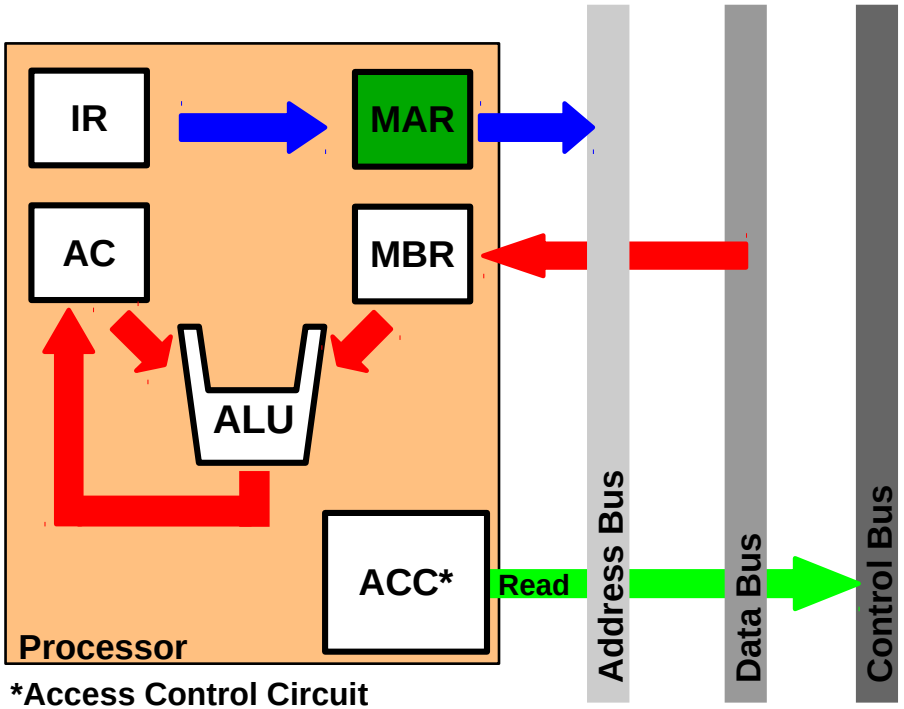
- Pseudo-Instruções do Montador (*Assembler*)
  - Segmento
    - .code, .data, .bss, .stack
  - Rótulo ou Nome
    - :
  - Delimitador de comentário
    - ;
  - Definição/Declaração de variáveis
    - DD, DB, RESD, RESB
- Conjunto de Instruções da Arquitetura (ISA)
  - Aritméticas
    - ADD, SUB
  -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

Operações Lógicas

- As operações lógicas consistem em duas instruções:
  - NAND (*Not AND*) – Função lógica binária AND com resultado negado.
  - SHIFT (*Arithmetic Shift*) – Deslocamento binário para direita/esquerda, introduzindo constantes no início/final, respectivamente.
- O AC é implícito, é um dos operandos fonte e é o de destino.
- Na instrução SHIFT, o operando em memória indica o sentido do deslocamento.
  - '0' sentido à direita (>>).
  - '1' sentido à esquerda (<<).
- Sintaxe:
  - Nand: [<rotulo>] NAND <endereco-memoria>
  - Deslocamento: [<rotulo>] SHIFT <endereco-memoria>

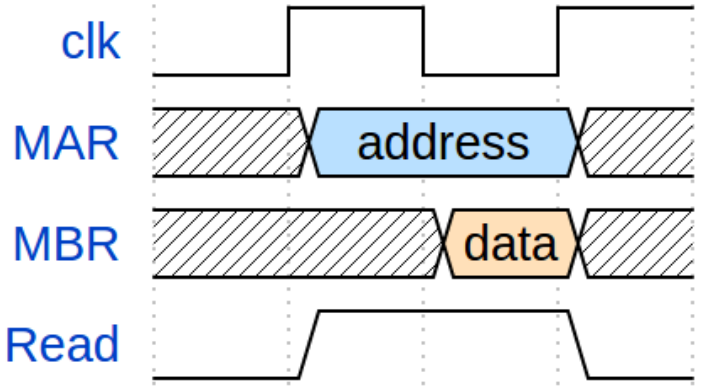
Operações Lógicas

- NAND (*Not AND*)



*Access Control Circuit

Waveform



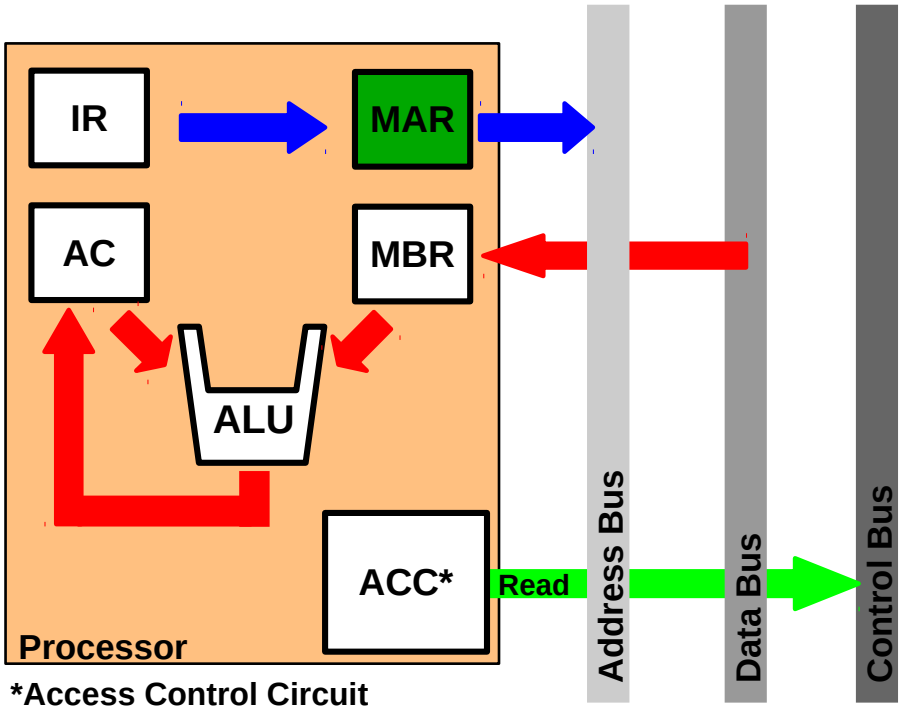
Microcode (RTN*)

```
MAR ← IR[11–0]
MBR ← M[MAR]
AC ← AC ^ MBR
```

*Register Transfer Notation

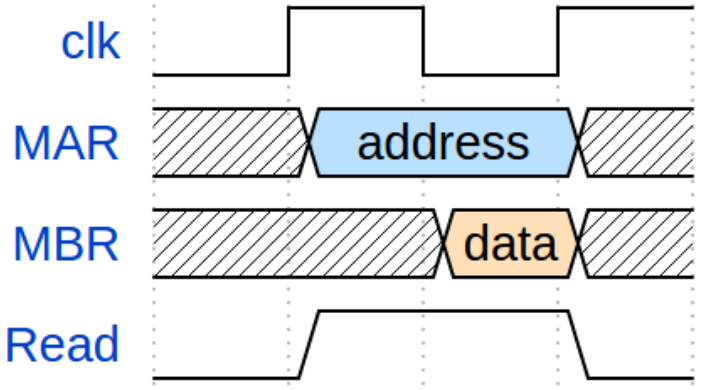
Operações Lógicas

- SHIFT (*Arithmetic Shift*)



*Access Control Circuit

Waveform



Microcode (RTN*)

```
MAR ← IR[11–0]

MBR ← M[MAR]

If MBR = 0 then
   AC ← AC >> 1
Else
   AC ← AC << 1
```

*Register Transfer Notation

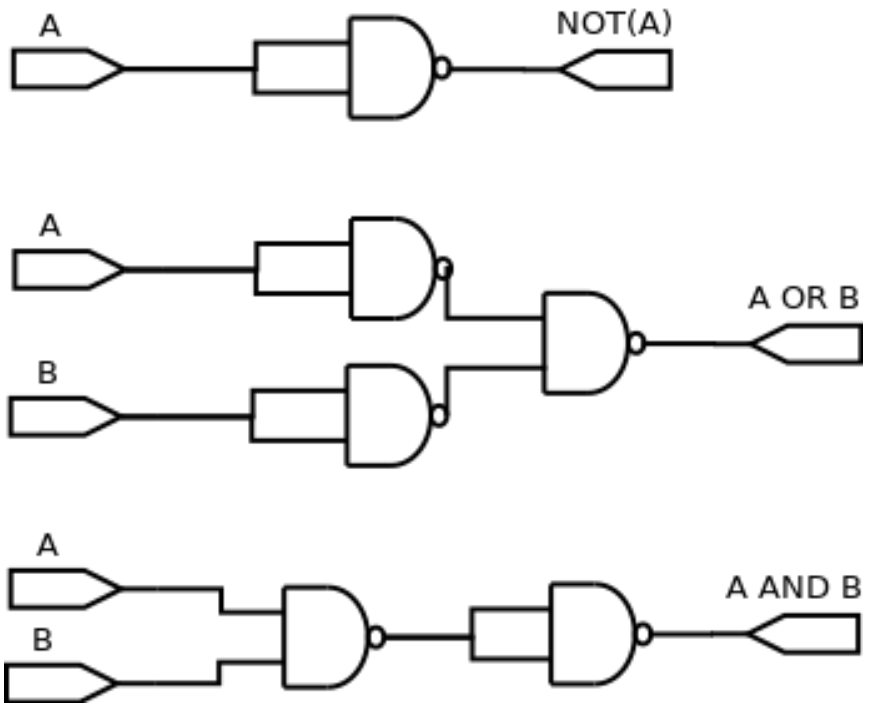
Operações Lógicas

- Exemplos práticos: Implementação das operações AND, OR e NOT.
- Procedimento:
  - Baixar e extrair o pacote:
    - 4.basic_logic_operations.zip
  - Menu “File” → “Open”
    - Ou Teclas “Ctrl+o”
  - Arquivos:
    - “and_operation.asm”
    - “or_operation.asm”
    - “not_operation.asm”

```
1       .code
2           ; c = a NAND b
3           LDA a
4           NAND b
5           STA c
6
7           ; c = c NAND c
8           LDA c
9           NAND c
10          STA c
11
12      end:
13          INT exit
14
15      .data
16          ;a = 1010b
17          a: DD 10
18
19          ;b = 1111b
20          b: DD 15
21
22          exit: DD 25
23
24      .bss
25          ;c = a AND b
26          c: RESD 1
27
28      .stack 10
29
```

Operações Lógicas

- Exemplos práticos: Implementação das operações AND, OR e NOT.
- Princípio utilizado para implementação:
  - Uso da porta lógica universal NAND




```
1       .code
2           ; c = a NAND b
3           LDA a
4           NAND b
5           STA c
6
7           ; c = c NAND c
8           LDA c
9           NAND c
10          STA c
11
12      end:
13          INT exit
14
15      .data
16          ;a = 1010b
17          a: DD 10
18
19          ;b = 1111b
20          b: DD 15
21
22          exit: DD 25
23
24      .bss
25          ;c = a AND b
26          c: RESD 1
27
28      .stack 10
29
```

Atividade Prática

1) Otimizar a operação de multiplicação/divisão por 2 da atividade anterior utilizando a instrução SHIFT

2) Implementar as operações lógicas:
    - NOR (Not OR)
    - XOR (eXclusive OR) → Somador
    - XNOR (eXclusive NOR) → Somador Invertido