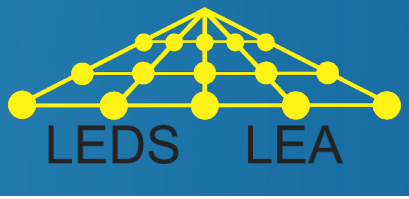
Arquitetura e Organização de Computadores

Conjunto de Instruções da Arquitetura – CompSim



Agenda

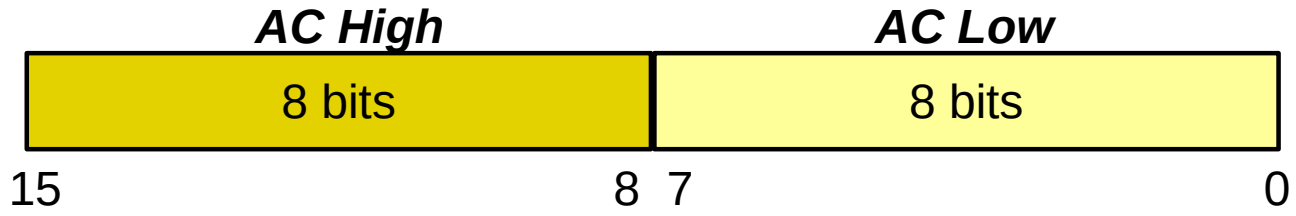
- Tipos de Instruções
- Operações Básicas de Entrada e Saída

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 Básicas de Entrada e Saída

- **Interface de E/S**
  - Em operações de E/S o Registrador Acumulador (AC) passa a incluir dois campos:



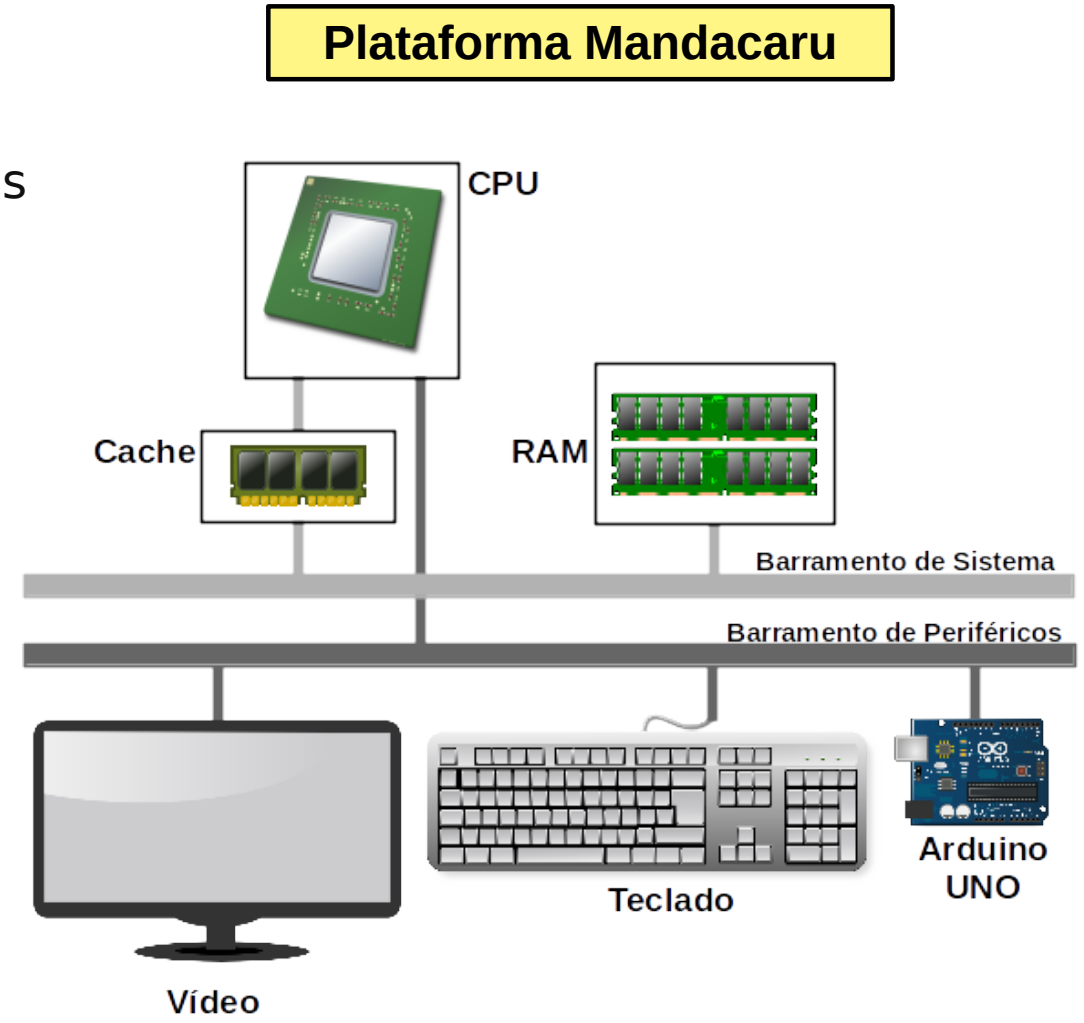
  - Onde:
    - ***AC High*** – Endereço de 8-bits do periférico (2^8 = 256 periféricos!)
    - ***AC Low*** – Dado de 8-bits que será lido/escrito

Operações Básicas de Entrada e Saída

- **Interface de E/S**
  - Vimos que a instrução INT é utilizada na operação de ***HALT*** de execução do processador.
  - Contudo, no CompSim, a instrução INT também é utilizada em operações de E/S.
  - Sintaxe:
    - Operação de E/S: [<rotulo>] INT <endereco-memoria>
      - Onde:
        - O AC é implícito e seus campos (***AC High*** e ***AC Low***) devem ser utilizados para endereço e dados, respetivamente.
        - O operando em memória indica o tipo de operação:
          - 0 – Entrada – Leitura de dado de 8-bits de um periférico
          - 1 – Saída – Escrita de dado de 8-bits em um periférico

Operações Básicas de Entrada e Saída

- **Periféricos do CompSim**
  - Plataforma Mandacaru
    - Barramento de periféricos
      - Vídeo
      - Teclado
      - Arduino
        - UNO
        - MEGA



Operações Básicas de Entrada e Saída

- **Periféricos do CompSim**

Periférico	Modo de Operação	Endereço (*AC High*)		Dado (*AC Low*)	Retorno (AC)
		Bin	Dec		
Vídeo	Output (Digital)	00000000	0	Byte	-
Teclado	Input (Digital)	00000001	1	-	Byte
Arduino – Porta B (Pinos 8 a 13)	Input/Output (Digital)	00000010	2	Byte*	Byte
Arduino – Porta D (Pinos 0 a 7)	Input/Output (Digital)	00000011	3	Byte**	Byte
Arduino – Porta C (Pinos A0 a A5)	Input (Analógico)	00000100	4	Byte***	Inteiro+
Arduino – Porta PWM (Pino 3)	Output (Analógico)	00000101	5	Byte	-
Arduino – Porta PWM (Pino 5)	Output (Analógico)	00000110	6	Byte	-
Arduino – Porta PWM (Pino 6)	Output (Analógico)	00000111	7	Byte	-
Arduino – Porta PWM (Pino 7)	Output (Analógico)	00001000	8	Byte	-
Arduino – Porta PWM (Pino 10)	Output (Analógico)	00001001	9	Byte	-
Arduino – Porta PWM (Pino 11)	Output (Analógico)	00001010	10	Byte	-

* Os 2 bits mais significativos são descartados (Mask 00111111).
** Os 2 bits menos significativos são descartados (Mask 11111100).
*** Utiliza apenas 3 bits para representar o intervalo de 0 a 5. Os 5 bits mais significativos são descartados (Mask 00000111).
\+ Utiliza *AC High* e *AC Low* para retornar um inteiro entre 0 e 1024.

Operações Básicas de Entrada e Saída

- Exemplos práticos: Acessando registradores.
- Procedimento:
  - Baixar e extrair o pacote:
    - 9.input_output_operations..zip
  - Menu “File” → “Open”
    - Ou Teclas “Ctrl+o”
  - Arquivos:
    - Vários exemplos práticos.

```
1   .code
2
3       ;set keyboard address
4       LDA keyboard_address
5
6       ;read char from keyboard
7       INT input
8
9       ;grava char em 'a'
10      STA a
11
12      ;encerra a aplicacao
13      INT exit
14
15  .data
16      ;A0 pin
17      pin: DD 0
18
19      arduino_portC: DD 1024
20
21      ;input interruption
22      input: DD 20
23
24      ;syscall exit
25      exit: DD 25
26
27  .bss
28      a: RESD 1
29
```

Atividade Prática

- Criar um programa onde:
  - O usuário deve entrar com um número inteiro no intervalo de 0 a 4;
  - O programa multiplicará a entrada por 2; e imprimirá o resultado.