

Arquitetura e Organização de Computadores

Conjunto de Instruções da
Arquitetura – CompSim





Agenda

- Tipos de Instruções
- Novos Modos de Endereçamento



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`, `RESB`, `RESQ`
- 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`



Novos Modos de Endereçamento

- Vimos até agora os seguintes tipos de endereçamento:
 - **Direto**, através das instruções **LDA** e **STA**;
 - **Indireto**, através das instruções **LDI** e **STI**; e
 - **Implícito**, através da instrução **SOP** (Push e Pop).
- CompSim suporta ainda mais dois modos de endereçamento:
 - **Imediato**: o operando da instrução será copiado para AC.
 - **De Registrador**: o operando consiste de uma referência a um determinado registrador, cujo conteúdo será copiado para AC.
 - São eles: CS, DS, SS e SP.



Novos Modos de Endereçamento

- Sintaxe:
 - Imediato: [<rotulo>] **MOV** <valor>
 - Onde <valor> pode ser um número inteiro positivo, entre 0 e 4089
 - Exemplo:
MOV 10
 - De Registrador: [<rotulo>] **MOV** <endereço-registrador>
 - Onde <endereço-registrador> pode ser:
 - \$cs ou \$CS – *Code Segment*
 - \$ds ou \$DS – *Data Segment*
 - \$ss ou \$SS – *Stack Segment*
 - \$sp ou \$SP – *Stack Pointer*
 - Exemplo:
MOV \$SP



Novos Modos de Endereçamento

- Exemplos práticos: Acessando registradores.
- Procedimento:
 - Baixar e extrair o pacote:
 - [8.addressing_modes.zip](#)
 - Menu “File” → “Open”
 - Ou Teclas “Ctrl+o”
 - Arquivos:
 - “immediate_addressing.asm”
 - “register_access.asm”

```
1  .code
2
3      MOV $CS
4      STA cs
5
6      MOV $DS
7      STA ds
8
9      MOV $SS
10     STA ss
11
12     MOV $SP
13     STA sp
14
15 end:
16     INT exit
17
18 .data
19     ;syscall exit
20     exit: DD 25
21
22 .bss
23     cs: RESD 1
24     ds: RESD 1
25     ss: RESD 1
26     sp: RESD 1
27
28 .stack 10
```



Atividade Prática

- Criar um programa para guardar 5 elementos na Pilha do programa e, sem utilizar a operação “Pop” (instrução **SOP**), acessar o terceiro elemento inserido na Pilha.
 - Dica: Utilize o endereço apontado pelo registrador **SP** (*Stack Pointer*).