

# Lab 1

# ASSEMBLY

- Linguagem básica estrutural característica de cada equipamento.
- Assembler: traduz o código interno de instruções da máquina.
- Os programas de apoio e inclusive partes importantes do sistema operacional são construídos, na maioria dos casos, a partir do mesmo.

# COMPILADORES

- Funcionam como tradutores de uma linguagem de uso mais compreensível pelo homem (linguagem de Alto Nível) para linguagem de máquina.
- Traduzem todo o código num novo código;
  - Executam com a mesma rapidez que os programas escritos diretamente em linguagem máquina

# COMPILADORES

- Interpretadores: interpretam instrução a instrução;
- São mais lentos do que os programas escritos diretamente em linguagem máquina.
- Assemblers: interpretam a linguagem simbólica em linguagem máquina.

# Linguagens de Programação

- Representam a interface entre o programador e o hardware, ou a ferramenta utilizada para materializar uma tarefa que será desenvolvida pelo sistema
- Na base está a linguagem de máquina, que é a versão final que o computador realmente processa.
- Há ainda a linguagem Assembly, intimamente ligada ao conjunto de instruções do processador.

# Linguagens de Programação

- **Nível intermediário:** encontram-se as linguagens que apresentam facilidade de programação com comandos, porém conseguem amplo acesso ao hardware, como as linguagens de baixo nível.
- Ideais para elaboração de programas como sistemas operacionais ou rotinas que interagem com dispositivos de I/O, memórias, periféricos.
- Exemplo: Linguagem C

# Registadores Microprocessador 8085

- 7 registradores de uso geral, todos de 8 bits

8 bits	8 bits
Acc	
B	C
D	E
H	L

- A (ACUMULADOR): utilizado em todas as operações de entrada/saída e em quase todas as instruções de lógica e aritmética.

# Registadores Microprocessador 8085

- Demais registradores de uso geral: importantes, principalmente porque podem ser utilizados em pares, formando registradores de 16 bits, organizados como: par BC, DE e HL
- Outros registradores: apontador de pilha (SP), o contador de programa (PC), o registrador de instruções e o registrador para FLAGS
- Indispensáveis para o funcionamento do 8085



# Conjunto de Instruções do 8085

- A operação de qualquer microprocessador é baseada em um conjunto de códigos que é denominado de CONJUNTO DE INSTRUÇÕES.
- Esses códigos deverão ser inseridos necessariamente em uma memória, e em muitos casos dados adicionais são necessários.
- O microprocessador pode apresentar diferentes modos para acesso à memória, chamados de *modos de endereçamento*.

# Conjunto de Instruções do 8085

- Organização das instruções no 8085:
- 1º byte → código (sempre)
- 2º byte → dado ou byte menos significativo de endereço(LSB)
- 3º byte → dado ou byte mais significativo de endereço(MSB)

# Conjunto de Instruções do 8085

- O conjunto de instruções do 8085 está organizado em 5 grupos básicos:
  - **1 - transferência de dados:** permitem a troca de dados entre registradores e memória;
  - **2 – operações lógicas:** permitem a implementação de funções lógicas, como AND, OR, XOR, comparações, etc.;
  - **3 – operações aritméticas:** permitem a realização de somas e subtrações;
  - **4 – controle de programa:** permitem desvios condicionados ou não, chamadas de subrotinas, etc.
  - **5 – manipulação de PILHA** e acesso aos dispositivos de entrada/saída.

# Conjunto de Instruções do 8085

- **Modos de Endereçamento no 8085**
- Implícito: o microprocessador “sabe” a priori onde está o dado (ex.: CMA, DAA, etc.)
- Por registro: operações envolvendo registradores internos (ex.: ADD B, SUB D, etc)
- Imediato: o dado segue o código da instrução (ex.: MVI A,25; CPI 00, etc)
- Direto: o endereço do dado segue o código (ex.: STA 2550, LDA 4001, etc)
- Indireto: o endereço do dado é passado via um par de registros (ex.: LDAX B, etc)

# Instruções de Transferência de Dados do 8085

- As instruções deste grupo realizam as seguintes funções:
  - Transferência de dados entre registradores e entre registrador e memória;
  - Atribuir valor ao registrador ou à memória;
  - Transferência de dados entre acumulador e memória;
  - Transferência de dados entre os registradores H e L e a memória;
  - Troca de valores entre o par de registradores H,L e o par D,E.

# Instruções de Transferência de Dados do 8085

- Considerações iniciais:
  - r, r1, r2 : registradores - A, B, C, D, E, H, L;
  - M : símbolo indicando endereço de memória formado pelo conteúdo do par H, L;
  - Data8 – Valor de 8 bits
  - Data16 – Valor de 16 bits (dado)
  - Adr16 – Valor de 16 bits (endereço)
  - DDD – registrador de destino do byte (dados)
  - FFF – registrador fonte do byte (dados)

# Instruções de Transferência de Dados do 8085

Registrador	Composição de Bits DDD ou FFF
<b>B</b>	<b>000</b>
<b>C</b>	<b>001</b>
<b>D</b>	<b>010</b>
<b>E</b>	<b>011</b>
<b>H</b>	<b>100</b>
<b>L</b>	<b>101</b>
<b>A</b>	<b>111</b>

NOME	REGISTRADORES	Composição de Bits PR
<b>B</b>	<b>B, C</b>	<b>00</b>
<b>D</b>	<b>D, E</b>	<b>01</b>
<b>H</b>	<b>H, L</b>	<b>10</b>
<b>SP</b>	<b>SP</b>	<b>11</b>

# Instruções de Transferência de Dados do 8085

- rh : registrador superior ( B, D e H);
- rl : registrador inferior (C, E, L);
- Tempo de Execução ou Processamento de uma instrução:

$$T_E = \frac{\text{Número de Estados } T(T)}{f_{\text{clock}}} = \frac{T}{2.35 * 10^6} = 425.10^{-9} * T \quad [s]$$



# Instruções de Transferência de Dados do 8085

- **MOV r1, r2**                       $(r1) \leftarrow (r2)$                       **01 DDD FFF**  
    T = 4
- **Move**
- O conteúdo do registrador r2 é copiado no registrador r1.
- Exemplos:

Instrução (Hexadecimal)	Código (Binário)	Código
• MOV A, A	01 111 111	7F
• MOV C, D	01 001 010	4A
• MOV H, B	01 100 000	60

# Instruções de Transferência de Dados do 8085

- **MOV r, M**                       $(r) \leftarrow ((H)(L))$  **01 DDD 110**                      T = 7
  - **Move**
  - O conteúdo do endereço de memória formado pelo par H,L é copiado no registrador r.
  - Exemplos:
- | Instrução<br>(Hexadecimal) | Código (Binário) | Código |
|----------------------------|------------------|--------|
| MOV A, M                   | 01 111 110       | 7E     |
| MOV C, M                   | 01 001 110       | 4E     |

# Instruções de Transferência de Dados do 8085

- **MOV M, r**                       $((H)(L)) \leftarrow (r)$  **01110 FFF**      T = 7
- **Move**
- O conteúdo do registrador r é copiado no endereço de memória formado pelo par H,L.
- Exemplos:

Instrução (Hexadecimal)	Código (Binário)	Código
• MOV M, A	0111 0 111	77
• MOV M, H	0111 0 100	74

# Instruções de Transferência de Dados do 8085

- **MVI r, Data8** ( $r \leftarrow \text{Data8}$ )                      00 DDD 110                      T = 7
  - **Move Immediate**                      Data8
  - O byte Data 8 é transferido para o registrador r.
  - Exemplos:
- | Instrução    | Código (Binário) | Código (Hexadecimal) |
|--------------|------------------|----------------------|
| • MVI A, 45H | 00 111 110       | 3E                   |
| •            | 0100 0101        | 45                   |
| • MVI B, 06H | 00 000 110       | 06                   |
| •            | 0000 0110        | 06                   |

# Instruções de Transferência de Dados do 8085

- **MVI M, Data8** ((H)(L)) ← Data8 0011 0110 T = 10
  - **Move Immediate** Data8
- O byte Data 8 é transferido para o endereço de memória dado pelo par HL.
- Exemplo:
- | Instrução  | Código (Binário) | Código (Hexadecimal) |
|------------|------------------|----------------------|
| MVI M, DCH | 0011 0110        | 36                   |
|            | 1101 1100        | DC                   |
- | ANTES  | APÓS                |
|--|---------------------|
| (H) = 20H (L) = 30H (2030H) = XX<br>(2030H) = DC H | (H) = 20H (L) = 30H |

# Instruções de Transferência de Dados do 8085

- **LXI PR, Data16**  $(rh) \leftarrow Data16^H (rl) \leftarrow Data16^L$       00 PR 0001 T = 10
- **Load Extended Immediate**  
Data16<sup>L</sup>
- Data16<sup>H</sup>

- O byte superior de Data16 é transferido para o registrador superior rh e o byte inferior de Data16 é transferido para o registrador inferior rl.
- Exemplo:

Instrução	Código (Binário)		Código (Hexadecimal)
LXI D, 20B0H	00 01 0001	11	
	1011 0000	B0	
	0010 0000	20	

- |                     |                     |
|---------------------|---------------------|
| • ANTES             | APÓS                |
| • (D) = XX (E) = XX | (D) = 20H (E) = B0H |

# Instruções de Transferência de Dados do 8085

- **LDA Adr16**  $(A) \leftarrow (Adr16)$  0011 1010  
T = 13
- **Load Accumulator** Adr16<sup>L</sup>
- Adr16<sup>H</sup>

- O conteúdo do endereço Adr16 é copiado no acumulador.
- Exemplo:

- | Instrução<br>(Hexadecimal) | Código (Binário) | Código |
|----------------------------|------------------|--------|
| LDA 20B0H                  | 0011 1010        | 3A     |
|                            | 1011 0000        | B0     |
|                            | 0010 0000        | 20     |

- | ANTES                   | APÓS                 |
|-------------------------|----------------------|
| (20B0H) = CD H (A) = XX | (20B0H) = CD H (A) = |
| CD H                    |                      |

# Instruções de Transferência de Dados do 8085

- **STA Adr16**  $(\text{Adr16}) \leftarrow (A)$  0011 0010  
T = 13
- **Store Accumulator** Adr16<sup>L</sup>
- Adr16<sup>H</sup>

- O conteúdo do acumulador é copiado no endereço Adr16.
- Exemplo:

- | Instrução | Código (Binário) | Código (Hexadecimal) |
|-----------|------------------|----------------------|
| STA 20B0H | 0011 0010        | 32                   |
|           | 1011 0000        | B0                   |
|           | 0010 0000        | 20                   |

- | ANTES                  | APÓS                    |
|------------------------|-------------------------|
| (20B0H) = XX (A) = CDH | (20B0H) = CDH (A) = CDH |