

ARQ1 \_ Aula\_15

Tema: Introdução à linguagem de máquina (*assembly*)

Orientação geral:

Apresentar todas as soluções em apenas um arquivo com formato texto (.txt).  
Sugere-se usar como nome Guia\_xx.txt, onde xx indicará o guia, exemplo Guia\_01.txt.  
Todos os arquivos deverão conter identificações iniciais com o nome e matrícula, no caso de programas, usar comentários.  
As implementações e testes dos exemplos em Verilog (.v) fornecidos como pontos de partida, também fazem parte da atividade e deverão ter os códigos fontes entregues **separadamente**, com o código fonte e os módulos de testes, a fim de que possam ser compilados e verificados.  
Sugere-se usar como nomes Guia\_01yy.v, onde yy indicará a questão, exemplo Guia\_0101.v  
As saídas de resultados, opcionalmente, poderão ser copiadas ao final do código, em comentários.  
Quaisquer outras anotações, observações ou comentários poderão ser colocadas em arquivo texto (README.txt) acompanhando a entrega.

Outras formas de solução serão **opcionais**; não servirão para substituir as atividades a serem avaliadas. Caso entregues, poderão contar apenas como atividades extras.  
Os programas com funções desenvolvidas em C, Java ou Python (c, .java, py), como os modelos usados para verificação automática de testes das respostas; caso entregues, também deverão estar em arquivos **separados**, com o código fonte e os módulos de testes, a fim de serem compilados e testados.  
As execuções deverão, preferencialmente, serão testadas mediante uso de redirecionamento de entradas e saídas padrões, cujos dados/resultados deverão ser armazenados em arquivos textos.  
Os resultados poderão ser anexados ao código, ao final, como comentários  
Planilhas, caso venham a ser solicitadas, deverão ser **programadas** e/ou usar funções nativas.  
Serão suplementares e opcionais, e deverão ser entregues em formato texto, preferencialmente, com colunas separadas por tabulações ou no formato (.csv), acompanhando a solução em texto.  
Arquivos em formato (.pdf), fotos, cópias de tela ou soluções manuscritas também poderão ser aceitos como recursos suplementares para visualização, mas não servirão como substitutos e **não** terão validade para fins de avaliação.  
Os *layouts* de circuitos deverão ser entregues no formato (.circ), identificados internamente.  
Os *layouts* de diagramas deverão ser entregues no formato (.jff), identificados externamente.  
Figuras exportadas pela ferramenta serão aceitas apenas como arquivos para visualização, mas **não** terão validade para fins de avaliação. Separar versões completas (a) e simplificadas (b).

## Atividade: Arquitetura de Computador – Intel 8085

Todos os programas deverão ser testados em simulador.

Dado o exemplo abaixo:

```
// Guia_1500.txt
// 999999 - Xxx Yyy Zzz
//
// Somar dois numeros de 8 bits
// Armazenar o primeiro dado em 0100, por exemplo e atualizar as instrucoes
// Armazenar o segundo dado em 0101
// Armazenar o resultado em 0103

// Area de código      ; Significado      // Comentario
    LXI  H,dado1        ; HL <- addr(dado1) // endereco do dado1
    MOV  A,M            ; A <- MEM [HL]     // A = MEM [dado1]
    INX  H              ; HL <- HL+1        // endereco do dado2
    ADD  M              ; A <- A + MEM [HL]  // A = A + MEM [dado2]
    STA  dado3          ; MEM [dado3] <- A  // MEM [dado3] = A
    HLT                  ; halt             // parar

; Area de dados
; dado1: 02h           ; primeiro dado em hexadecimal
; dado2: 03h           ; segundo dado em hexadecimal
; dado3: 00h           ; resultado em hexadecimal
```

### Exercícios

01.) Implementar um programa (Guia\_1501.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits.

$$\text{dado04} = \text{dado01} - \text{dado02} - \text{dado03}$$

02.) Implementar um programa (Guia\_1502.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 16 bits (sinal+amplitude).  
Sugestão: Usar complemento de 2 para a subtração.

$$\text{dado05} = \text{dado01} - \text{dado02} + \text{dado03} - \text{dado04}$$

- 03.) Implementar um programa (Guia\_1503.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits codificados em BCD.  
DICA: Se passar de 9, precisará ajustar o próximo bit para converter a saída.

$$\text{dado05} = \text{dado01} - \text{dado02} - \text{dado03} - \text{dado04}$$

- 04.) Implementar um programa (Guia\_1504.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits.  
DICA: Para o dobro, somar o acumulador com ele mesmo.

$$\text{dado03} = 2 * \text{dado01} - 3 * \text{dado02}$$

- 05.) Implementar um programa (Guia\_1505.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits.  
DICA: Se usar deslocamentos/rotações para multiplicar e dividir, levar em consideração o bit do sinal.  
Observar que há instruções de rotação e não para simples deslocamentos.  
Melhor usar somas e comparações.

$$\text{dado03} = \text{dado01} * 4 + \text{dado02} / 2$$

#### Extras

- 06.) Implementar um programa (Guia\_1506.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits.

$$\text{dado03} = 3 * (\text{dado01} - 4 * \text{dado02})$$

- 07.) Implementar um programa (Guia\_1507.txt) para o processador 8085 para calcular o resultado da expressão aritmética abaixo, usando dados com 8 bits.  
DICA: Se usar deslocamentos/rotações para multiplicar e dividir, levar em consideração o bit do sinal.  
Observar que há instruções de rotação e não de simples deslocamentos.  
Melhor usar somas e comparações.

$$\text{dado03} = \text{dado01} \% 2 + \text{dado02} / 4$$