

## Aula 23: Controle de Fluxo do Programa-Parte 2

## 1 Objetivos

- Trabalhar o mapeamento de estruturas de decisão em assembly;
- Trabalhar o mapeamento de estruturas de repetição em assembly;

## 2 Mapeamento de estruturas decisórias em Assembly

Usando as instruções que foram repassadas na aula anterior vamos mapear algumas estruturas decisórias conhecidas no assembly;

### 2.1 alfa==beta

```
1 if (alfa==beta) {
2     /* bloco1 */
3 } else {
4     /* bloco2 */
5 }
```

Assumindo que alfa é o registrador *A* e beta é um dado armazenado em 20H temos:

```
1 beta EQU 20H
2     cjne A,beta,bloco2
3     ; inicio bloco1
4     sjmp fim_if
5 bloco2:
6     ; inicio_bloco2
7 fim_if: ; final da estrutura if
```

### 2.2 alfa >= beta

```
1 if (alfa>=beta) {
2     /* bloco1 */
3 } else {
4     /* bloco2 */
5 }
```

```
1 beta EQU 20H
2     cjne A,beta,ponto_aux ; a -> alfa
3 bloco1: ; inicio bloco1
4     sjmp fim_if
5 ponto_aux:
6     jnc bloco1
7 bloco2: ; inicio_bloco2
8     :
9 fim_if: ; final da estrutura if
```

### 3 Exercícios

1. Mapear em assembly a estrutura decisória:

```
1  if (alfa>beta) {  
2      /* bloco1 */  
3  } else {  
4      /* bloco2 */  
5  }
```

2. Construir um programa e um fluxograma para ler um código de 8 bits da porta P1 e acionar um relé pela porta P2.0 se o código for correto. Supor existência de uma subrotina `esc_liberado_acesso` e `esc_acesso_negado`. Use o bit P2.0 para validar a entrada do código.
3. Construir uma estrutura em assembly equivalente a um comando *case* do C;
4. Construir uma estrutura equivalente a um *for* conforme mostrado abaixo. Use o acumulador como variável de a variável de controle.

```
1  for (i=5;i<100;i++) {  
2      /* bloco */  
3  }
```