

Por: Frederico Pissarra

```

i-----©
| ASSEMBLY VII |
E-----¥

```

Algumas instruções afetam somente aos flags. Dentre elas, as mais utilizadas são as instruções de comparação entre dois dados.

| Comparações aritméticas:

A instrução CMP é usada quando se quer comparar dois dados, afetando somente aos flags. Eis a sintaxe:

```

+-----+
| CMP AL,1Fh
| CMP ES:[DI],1
| CMP AX,[SI]
+-----+

```

Esta instrução faz a subtração entre o operando mais a esquerda e o mais a direita, afetando somente os flags. Por exemplo, se os dois operandos tiverem valores iguais a subtração dará valores iguais e o flag de ZERO será 1. Eis a mecânica de CMP:

```

+-----+
| CMP AL,1Fh   ; AL - 1Fh, afetando somente os Flags
+-----+

```

| Comparações lógicas:

A instrução TEST é usada quando se quer comparar o estado de determinados bits de um operando. Eis a sintaxe:

```

+-----+
| TEST AL,10000000b
| TEST [BX],00001000b
+-----+

```

Esta instrução faz um AND com os dois operados, afetando apenas os flags. Os flags Z, S e P são afetados, os flags O e C serão zerados.

| Instruções que mudam o estado dos flags diretamente:

- CLC - Abreviação de CLear Carry (Zera o flag Carry).
- CLD - Abreviação de CLear Direction (Ajusta flag de direção em zero, especificando o sentido correto para instruções de bloco).
- CLI - Abreviação de CLear Interrupt (Mascara flag de interrupção, não permitindo que a CPU reconheça as interrupções mascaráveis).
- CMC - Abreviação de CoMplement Carry (Inverte o flag de carry).
- STC - Abreviação de SeT Carry (Faz carry = 1).
- STD - Abreviação de SeT Direction (flag de direção setado - indica que as instruções de bloco incrementarão os seus pointers no sentido contrário - de cima para baixo).
- STI - Abreviação de SeT Interrupt (Faz com que a CPU volte a reconhecer as interrupções mascaráveis).

Uma interrupção é um "desvio" feito pela CPU quando um dispositivo requer a atenção da mesma. Por exemplo, quando você digita uma tecla, o circuito do teclado requisita a atenção da CPU, que por sua vez, para o que está fazendo e executa uma rotina correspondente à requisição feita pelo dispositivo (ou seja, a rotina da interrupção). Ao final da rotina, a CPU retorna à tarefa que estava desempenhando antes da interrupção. Nos PCs, TODAS as interrupções são mascaráveis (podem ser ativadas e desativadas quando quisermos), com a única exceção da interrupção de checagem do sistema (o famoso MEMORY PARITY ERROR é um exemplo!).