

Acesso à Memória

Temos basicamente 2 tipos de abordagem para o acesso à memória com o 8085. A primeira é com endereços fixos e a segunda com endereços indexados pelo par HL.

Endereços fixos

Este tipo de acesso é feito com as instruções LDA e STA. Veja o vídeo sobre o exemplo abaixo e veja como funciona no simulador.

Exemplo 1:

```
MVI A, 5
STA 0880H
MVI A, 0
LDA 0880H
HLT
```

Neste programa simples, é atribuído 5 ao acumulador e na sequência a instrução diz que deve ser armazenado o valor de A no endereço de memória 0880H. Em seguida, zeramos o A e fazemos uma leitura da memória no endereço 0880H (onde havia sido armazenado o 5) e é copiado para o A.

Endereços indexados pelo HL

A forma anterior é interessante quando temos poucos dados para fazer acesso à memória. A medida que precisamos trabalhar com vetores (sequência de endereços de memória associada a uma operação), este tipo de acesso é complicado pois precisamos escrever um LDA ou STA para cada endereço de memória, aumentando o esforço de programação e a quantidade de memória para aquele programa. Uma outra forma de acesso à memória que resolve este problema é usar a letra M (memória) nas instruções MOV, indexada pelo par de registradores HL.

Como exemplo a instrução MOV A, M fará a mesma coisa que LDA, ou seja, transferirá um byte da memória para o A. O endereço de memória acessado será dado pelo par HL. As instruções LDA e STA limitam a comunicação entre a memória e o registrador A, sendo que as instruções MOV podem usar o M e qualquer registrador de 8 bits diferente do A. O exemplo abaixo deixa claro como isto é uma vantagem na transferência de muitos dados usando um loop e incrementando o par HL para mudar o endereço de memória

Exemplo 2

```
                MVI B, 0
                MVI C, 5
                LXI H, 0880H
loop:           MOV M, B
                INR B
                INX H
                DCR C
                JNZ loop
                HLT
```

Veja o vídeo explicativo sobre este exemplo. B e C são atribuídos com o valor 0 e 5 respectivamente, sendo o C o contador do loop deste programa. Neste caso o LXI H, 0880H indica que o par HL é inicializado com o valor 0880H, sendo que para as instruções com a letra M se torna um endereço de memória. MOV M, B coloca o conteúdo de B (inicialmente) no endereço 0880H. Na sequência o B é incrementado e a instrução INX H é uma instrução que incrementa o par HL (como se fossem um número de 16 bits) e isto serve para que o HL mude de endereço de memória (para o byte seguinte). DCR C e JNZ loop fazem parte do controle do loop e indicam que ele deve ocorrer o número de vezes inicial do C (no caso 5). Quanto executar novamente o corpo do loop, o novo B será copiado para a memória no novo endereço e isso até que C seja 0.

Exemplo 3

```
                MVI A, 0
                MVI C, 16
                LXI H, 0880H
loop:           ADD M
                INX H
                DCR C
                JNZ loop
                HLT
```

Este exemplo faz o somatório dos 16 conteúdos de memória a partir de 0880H (incluindo). Visualmente é uma linha da representação da memória no simulador. Podemos editar essas linhas para testar o programa. Veja o vídeo deste exemplo para maiores detalhes.