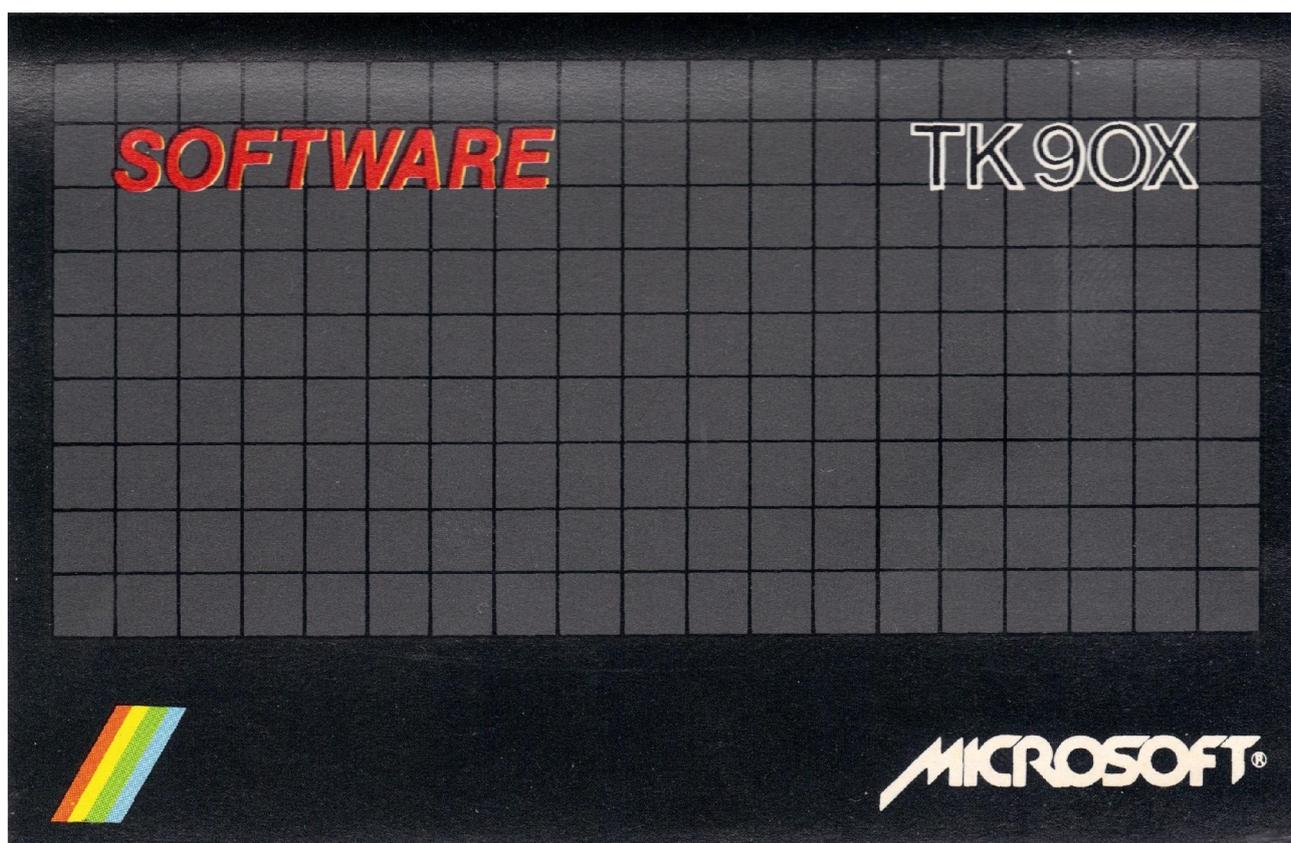
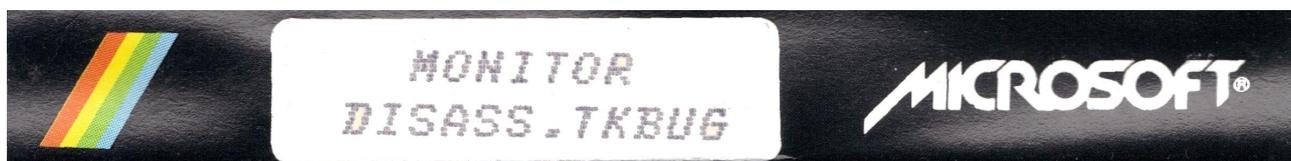


MANUAL

MONITOR DISASSEMBLER/TKBUG



Redigitado e
Scaneado por:
Clóvis frio_tk

ÍNDICE

INSTRUÇÕES	2
COMANDOS DISPONÍVEIS	4
CAPS SHIFT 1	4
SYMBOL SHIFT 3	4
SYMBOL SHIFT 4 ou \$	5
ENTER	5
CAPS SHIFT 5	5
CAPS SHIFT 7	5
CAPS SHIFT 8	5
G	5
H	6
I	6
J	6
SYMBOL SHIFT K	6
L	6
M	7
N	7
SYMBOL SHIFT N	7
O	8
P	8
SYMBOL SHIFT P	8
Q	8
SYMBOL SHIFT T	9
T	9
Imprimir	10
Gerar o Programa Fonte	12
U	13
V	14
W	14
X	16
Y	16
SYMBOL SHIFT Z	17
MODIFICANDO A MEMÓRIA	17
MODIFICANDO OS REGISTRADORES	18
MANUAL ORIGINAL	19
BOX E CASSETE	20

MONITOR DISSASSEMBLER/TKBUG

INSTRUÇÕES

O TKBUG é provido na forma de tabela relocada. Para utilizá-lo, basta seguir estes passos:

- Carregue o programa no endereço desejado, digitando:
LOAD " " CODE endereço escolhido ENTER
- Entre no TKBUG, digitando:
RAND USR endereço escolhido para a carga ENTER

Ex:

```
LOAD " " CODE 49152
```

lê o programa e carrega no endereço #C000

Após carregado o programa aparece a mensagem:

```
0 Executado 0:1
```

```
RAND USR 49152 - entra no programa desassembler.
```

A tela apresenta a seguinte mensagem:

```
-----  
! C1985*MULTISOFT inf. ltda.*TKBUG !  
-----
```

A seguir, a tabela, aparece na outra página:

```

0000    F3        DI
>PC CB89    F3 22 87 D3 2A 8E D3
  SP FF40    2B 2D 65 33 32 90 ED
  IY 5C3A    FF ED 00 54 FF 00 00
  IX 03D4    04 0C 0D 20 FD 0E 3F
  HL 0000    F3 AF 11 FF FF C3 CB
  DE 0012    15 AF D3 DF C9 0D 2A
  BC 0053    E1 6E FD 75 00 ED 7B
  AF 0044    Z  V
  IR 3F4A

      FFF4 10  FFFC  42  0004 FF
      FFF5 10  FFFD  42  0005 C3
      FFF6 10  FFFE  3C  0006 CB
      FFF7 00  FFFF  00  0007 11
      FFF8 00 >0000 F3< 0008 2A
      FFF9 42  0001  AF  0009 5D
      FFFA 42  0002  11  000A 5C
      FFFB 42  0003  FF  000B 22

```

>C

Onde:

- 0000** - endereço de memória apontado.
- F3** - mnemônicos do endereço.
- DI** - instrução gerada pelos mnemônicos.
- >** - apontador de registrador, para modificar o conteúdo dos registradores.
- PC a IR** - nomes dos registradores do Z-80.
- CB89 a 3F4A** - valor atual do conteúdo dos registradores.
- FB a D3** - sete localizações de memória a partir do endereço apontado pelo registrador PC (igual para os outros registradores).
- F** - registrador de "flags". É exposto de acordo com os "flags" afetados na instrução corrente (SZ HVNC), indicando se o sinal, zero, half carry, paridade/ overflow, adição/subtração e carry são setados.
- IR** - registrador de "refresh". É alterado a cada instrução.
- FFF4 10 a 000B 22** - seção de 24 bytes de memória, organizados com endereços (2 bytes) e seu conteúdo (1 byte).
- >0000 F3<** - valor do apontador.

Os comandos do teclado entram em resposta ao símbolo de pronto (>), no lado esquerdo inferior da tela.

Obs: Alguns comandos poderosos podem ser desastrosos se usados erroneamente. Assim, não se deve esquecer de pressionar **SYMBOL SHIFT** juntamente com a letra do comando.

Os comandos inválidos são simplesmente ignorados.

Acima da tela frontal os comandos são processados. Muitos comandos necessitam entrar com um número hexadecimal. Se você entra mais de 4 dígitos para número hexadecimal, apenas os 4 últimos dígitos são conservados e colocados na tela.

- Para voltar ao **BASIC**,

.pressione **CAPS SHIFT 1**

- Para retornar ao **TKBUG**,

.execute o endereço original acrescido de 29 (decimal):

RAND USR endereço escolhido para a carga somado a 29

No exemplo acima: **RAND USR 49181**

COMANDOS DISPONÍVEIS

Os comandos seguintes são disponíveis no TKBUG:

*** CAPS SHIFT 1**

Retorna ao BASIC.

Para entrar novamente no TKBUG, execute **RAND USR** endereço apontado somado a 29.

*** SYMBOL SHIFT 3**

Troca para decimal os endereços dessassemblados, os endereços de memória da seção de 24 bytes e os endereços do registradores.

Para retornar à forma inicial, ou seja, hexadecimal, pressione novamente **SYMBOL SHIFT 3**.

*** SYMBOL SHIFT 4 ou \$**

Dessassembla o programa a partir do apontador de memória (apresentado na parte superior da tela inicial).

Para dessassemblar outra página, pressione qualquer tecla.

Pra retornar à tecla inicial, pressione novamente **SYMBOL SHIFT 4**.

*** ENTER**

Incrementa o apontador de memória uma vez e a seção de 24 bytes de memória.

*** CAPS SHIFT 5**

Decrementa o apontador de memória de oito.

*** CAPS SHIFT 7**

Decrementa o apontador de memória de um.

*** CAPS SHIFT 8**

Incrementa o apontador de memória de oito.

*** G**

Pesquisa a memória para um string específico. Depois da introdução dos caracteres, a pesquisa é realizada a partir do apontador de memória atual até encontrar o string. Ao encontrá-lo, o apontador é atualizado com o endereço onde se encontra o string.

Ex: M: 8000 **ENTER** - atualiza o apontador para 8000
 G: D3 **ENTER** =\
 : D9 } string a ser procurado a partir
 : **ENTER** =/ do endereço 8000

Obs: Coloque apenas 2 bytes em cada linha. Sendo colocados mais bytes, o comando aceitara somente os 2 últimos.

Se desejar que sejam encontrados mais strings D3D9 [OUT (#D9),A], pressione **N**.

*** H**

Converte um número na base decimal para hexadecimal.

Ex: H: 4000 = 0FA0

*** I**

Usado para transferir um bloco de memória para outra posição de memória.

Inicial = endereço inicial do bloco de memória a ser transferido

Final = endereço final do bloco de memória a ser transferido

Para = endereço inicial para onde deve ser transferido

Ex:

Inicial = C000

Final = C100

Para = 8000

O conteúdo de C000 até C100 é transferido para 8000 a 8100.

*** J**

Este comando permite que um programa em assembler seja executado, partindo de um endereço especificado em hexadecimal. Quando este comando é executado, a tela é limpa.

Ex: J: C01D - endereço de reentrada do TKBUG.

*** SYMBOL SHIFT K**

Este comando executa um programa em assembler, partindo do endereço estipulado no Contador de Programa (PC).

Ex: Se >PC C01D, ao executar **SYMBOL SHIFT K**, opera-se nova entrada no TKBUG.

Obs: Este comando pode funcionar com o comando **W**.

*** L**

Lista os dados a partir do endereço dado pelo apontador de memória em código ASCII.

São listados 80 bytes de cada vez. Para continuar a listagem, pressione **ENTER**.

Para voltar à tela principal, tecele **CAPS SHIFT 5**.

*** M**

Muda o apontador de memória para o endereço estipulado em hexadecimal.

Usado para ler alguma sub-rotina ou parte do programa, ou modificar algum endereço.

Ex: M: 8000

*** N**

Usado em conjunto com o comando **G**, continua a busca do string estipulado pelo comando **G**, partindo do apontador de memória.

Ex: M: 0000
G: CD ENTER
: FB ENTER
: 3C ENTER

Se deseja saber se há mais alguma instrução CDFB3C (CALL #3CFB), pressione **N**. A cada pressão da tecla **N**, aparecem, no apontador de memória, os endereços que contêm essa instrução. No exemplo acima, primeiramente aparece:

0D1C CDFB3C CALL #3CFB

Após pressionar **N**

D3C2 CDFB3C CALL #3CFB

*** SYMBOL SHIFT N**

Transfere o conteúdo do endereço do SP (stack pointer) para o apontador de memória. Isto é usado quando você necessita olhar outra parte do programa ou executar o endereço dado por (SP).

Para sair, basta dar o comando **M** e o endereço que desejar.

*** O**

Este comando é usado para realizar um desvio relativo (JR).

Suponhamos que no endereço de memória #67FF haja o código hexadecimal #20 (JR NZ,disp), e no endereço de memória #6800 haja #16; e o apontador de memória esteja no endereço #6800. Ao executar o comando **O**, há um desvio relativo para o endereço #6817, que é o deslocamento que aparece na instrução.

Ex: 67FF 2016 JR NZ,#6817

*** P)**

Este comando permite encher uma determinada região de memória com apenas um código. Os endereços são introduzidos em hexadecimal.

Inicial - primeiro endereço a ser preenchido
Final - último endereço a ser preenchido
Com - valor do código em hexadecimal

Ex: Inicial: 8000
 Final : 8010
 Com : 20

Os endereços de 8000 a 8010 vão estar com o código 20.

*** SYMBOL SHIFT P**

Este comando é idêntico ao comando **L**, sendo a saída feita através da impressora.

Para retornar à tela principal, use **CAPS SHIFTS 5**.

Para continuar, pressione qualquer tecla.

O procedimento para utilizar a impressora é idêntico ao descrito no comando **T**.

*** Q**

Este comando mostra, na tela principal, o conteúdo dos registradores alternativos (AF', HL', DE', BC') no lugar dos registradores (AF, HL, DE, BC). Para retornar pressione **Q** novamente.

* SYMBOL SHIFT T

Este comando Seta um "breakpoint" após a instrução corrente e continua a execução.

```
EX:      9000 B7      OR      A
          9001 CD0098 CALL    #9800
          9004 010000 LD      BC,#0000
          .          .      .          .
          9800 21FFFF LD      HL, -1
```

Sendo usado **SYMBOL SHIFT Z** para executar as instruções a partir de #9000, ao chegar a #9001, este comando dá um desvio absoluto para #9800. Não desejando assim, você deve usar o comando **SYMBOL SHIFT T**. O programa continua a seqüência do programa, executando #9004.

* T

Este comando desassembla um bloco de memória, com opção para impressora e geração de programa-fonte (SOURCER).

Inicial : primeiro endereço do bloco de memória a ser desassemblado

Final : último endereço do bloco de memória a ser desassemblado

Imprimir : opção para impressora

Texto : endereço onde inicia o texto

Inicial : primeiro endereço da área de memória que deve ser interpretado como caractere ASCII

Final : último endereço da área de memória que deve ser interpretado como caractere ASCII.

Para voltar à primeira página, depois de desassemblado o bloco de memória, pressione qualquer tecla.

Para voltar à primeira página, antes que todo o bloco tenha sido desassemblado, dê uma pausa e pressione:

CAPS SHIFT 5.

Para dar uma pausa no desassembler, pressione:

BREAK / SPACE ou **ENTER.**

Para continuar, pressione qualquer tecla exceto:
CAPS SHIFT 5 ou **CAPS SHIFT 1**.

Se houver algum opcode inválido, o comando desassembla com um NOP e imprime um asterisco (*) no lado direito do opcode.

Para desassemblar em DEFB, dê a área de memória, na segunda vez em que aparecer Inicial (end. inicial) e Final (end. final).

Ex:

```
Inicial : 8B ENTER
Final   : 9E ENTER
Imprimir: S(im) ou N(ão)
Texto   : ENTER
Inicial : 95 ENTER
Final   : 9E ENTER
Inicial : ENTER
Final   : ENTER
008B FE16 CP #16
008D 3801 JR C,L0090
008F 23 INC HL
0090 37 SCF
0091 225D5C LD (#5C5D), HL
0094 C9 RET
0095 BF524E DEFB #BF, "R", "N"
0098 C4494E DEFB #C4, "I", "N"
009B 4B4559 DEFB "K", "E", "Y"
009E A4 DEFB #A4
```

* IMPRIMIR

Para imprimir, siga o procedimento abaixo:

Inicial : endereço inicial da área de memória a ser impressa

Final : endereço final da área de memória a ser impressa

Imprimir : S(im)

Região : região onde os LABELS do programa estarão na memória. Desejando apenas imprimir, sem a geração do SOURCER, não entre valor neste campo.

Texto : desejando apenas imprimir, deixe este campo vazio
Inicial : havendo área de memória que deva ser interpretada como ASCII, dê o primeiro endereço da área.
Em caso contrário, deixe vazio este campo.
Final : havendo área de memória que deva ser interpretada como ASCII, dê o último endereço da área.
Em caso contrário, deixe vazio este campo.

Ex:

Inicial : 9000 **ENTER**
Final : 9030 **ENTER**
Imprimir : S
Região : 9200 (início da tabela; necessário para a geração do SOURCER).
Texto : **ENTER** (apenas imprimir)
Inicial : **ENTER** (não havendo ASCII)
Final : **ENTER** (não havendo ASCII)

O conteúdo de 9000 a 9030 é impresso como segue:

```
9000      0E46          LD      C,#46
9002      CD1690  L9002  CALL   L9016
9005      B7          OR      A
.
.
.
9016      C5          L9016  PUSH  BC
.
.
```

Nota: A impressora TK PRINTER 500 deve ser conectada diretamente à expansão do TK90X. Após a conexão, carregue os programas Assembler e TKBUG.

Outas impressoras devem ser conectadas ao TK90X através de interface de impressora. Após a conexão, carregue o software de impressora e, depois, os programas.

* GERAR O PROGRAMA FONTE (SOURCER)

Para gerar um programa fonte (SOURCER) proceda do seguinte modo:

Tendo sido carregados os dois softwares (Assembler e TKBUG). Entre no Assembler e use o comando **X**, para saber onde começa a área reservada ao início do texto.

EX: Sendo colocado o Assembler em 24064, o início do texto é 32921 ou 8099 em hexadecimal.

- Retorne ao BASIC e, então, ao TKBUG.

Acione o comando **T** e digite:

Inicial : endereço inicial do programa a ser gerado

Final : endereço final do programa a ser gerado

Imprimir : S(im) ou N(ão) se deseja ou não imprimir o programa

Texto : endereço dado no Assembler através do comando **X** (em hexadecimal). Se o Assembler foi lido no endereço 24064, digite neste campo 8099.

Região : região da memória onde será iniciada a tabela de LABELS. Se nada é digitado neste campo, o computador coloca a tabela aleatoriamente na memória.

Inicial : endereço inicial da área de memória que deve ser interpretada como ASCII.

Final : endereço final da área de memória que deve ser interpretada como ASCII.

Ex:

```
Inicial : 9000 ENTER
Final   : 9032 ENTER
Imprimir: N(ão) ou S(im)
Texto   : 8099
Região  : 9300 (endereço inicial da tabela de
             LABELS)
Inicial : ENTER (sem interpretação ASCII)
Final   : ENTER (sem interpretação ASCII)
```

Obs: O programa pode ser simultaneamente gerado e impresso via impressora.

A tela mostra o programa desassemblado e a mensagem:

Fim do Texto =#(endereço final do texto em hexadecimal)

- Pressione qualquer tecla para retornar a tela inicial
- Transforme o primeiro e o segundo byte do endereço do final do texto em hexadecimal

Sugestão: Use o comando **M** e digite o 1º byte após pressionar **SYMBOL SHIFT 3**: este muda para decimal. Proceda da mesma forma com o 2º byte.

- Retorne ao BASIC e digite os seguintes comandos:

POKE: (endereço onde foi lido o programa Assembler)
+54, 2º byte decimal

POKE: (endereço onde foi lido o programa Assembler)
+55, 1º byte decimal

EX: Se o Assembler foi lido em 24064, e o endereço final do texto é 129 (#81) e 193 (#C1), então:

```
POKE 24064+54, 193
POKE 24064+55, 129
```

- Retorne ao Assembler e liste o programa fonte.

* U

Usado em conjunto com o comando **O**.

Lembre-se de que o comando **O** realiza um desvio relativo (JR). O comando **U** retorna ao ponto de origem do desvio relativo.

EX: 67FF 2016 JR NZ, #6817

Estando o apontador com #6800, ao pressionar **O**, executa-se um desvio relativo para #6817, e o apontador fica com o endereço #6817.

Para retornar ao original, use o comando **U**, e o apontador de programa ficará com #6800 novamente.

* **V**

Este comando retorna de um desvio absoluto (dado por um comando **X**). É usado em conjunto com o comando **X** de forma semelhante ao **U** e **O**.

EX: 8000 D4ES80 CALL #80E8

Colocando o apontador em #8001 e executando o comando **X**, este comando vai para #80E8. Para retornar desse desvio, utilize o comando **V**.

* **W**

Este comando seta no apontador de memória, um "breakpoint": uma instrução de CALL impressa no painel frontal no apontador de programa (CDA0CB). Esta instrução (CALL) habilita o programa para uma execução e analisa os registradores do Z-80, os flags e algumas localizações de memória importantes.

Desejando parar a execução de um programa num endereço específico, use o comando **M** para setar o apontador de memória com esse endereço e, então, use **W**. Os 3 bytes do código que estava originalmente neste endereço, são salvos e substituídos por CALL #CBA0. Sendo executado, o programa irá até esse endereço e parará, dando o conteúdo dos registradores, restaurando esse endereço com o conteúdo original e colocando-o no apontador de memória.

Obs.: O TKBUG usa a área do fim do programa para armazenar informações do "breakpoint". Se o comando **T** é acionado, todos os breakpoints usados no programa são perdidos, pois este comando também usa a área do fim do programa.

Exemplo:

```
7080 2A0072    LD      HL, (#7200)
7083 ED5B0272  LD      DE, (#7202)
7087 CD0071    CALL   #7100
708A 7C        LD      A,L
708B CD1D71    CALL   #7110
708E 7D        LD      A,L
708F CD1D71    CALL   #711D
7092 210000    LD      HL,#0000
7100 AF        XOR     A
7101 ED52     SBC    HL, DE
7103 1 9      ADD    HL, DE
7104 3001     JR     NC,#7107
7106 EB       EX     DE, HL
7107 B2       OR     D
7108 B7       SCF
7109 C0       RET    NZ
710A B3       OR     E
710B 5A       LD     E,D
710C 2007     JR     NZ,#7115
710E EB       EX     DE, HL
710F C9       RET
7110 EB       EX     DE, HL
7111 19       ADD    HL,DE
7112 EB       EX     DE, HL
7113 29       ADD    HL, HL
7114 D8       RET    C
7115 1F       RRA
7116 30FB     JR     NC,#7113
.
.
.
7200 1B2A     DEFW   10779
7202 0300     DEFW   3
```

M: 7080 ENTER = seta apontador de memória com #7080
7080^M = seta Contador de Programa (PC) com #7080
^Z = executa rotina LD HL, (#7200)
^Z = executa rotina LD DE, (#7202)
^Z = executa rotina CALL#7100
M: 7115 ENTER = seta apontador de memória com #7115
W = seta um "breakpoint"
^K = executa programa até #7115, restaurando o endereço #7115 com conteúdo original e para.

Obs: O sinal ^ corresponde a **SYMBOL SHIFT**.

*** X**

Este comando é usado para realizar um desvio absoluto (JP ou CALL).

Ex:

Suponha que na tela principal esteja:

```

      8702      AF
      8703      CD
      >8704     2F<
      8705      94
  
```

Sendo executado o comando X 8703 CD, há um desvio relativo para o endereço #942F.

*** Y**

Este comando permite a introdução de caracteres ASCII na memória, a partir do endereço dado no apontador de memória.

Para sair deste comando, use **CAPS SHIFT 5**.

EX: M: 8000
 Posiciona o apontador em 8000.
 Depois tecle **Y** e escreva TK90X. Retorne à tela inicial, usando **CAPS SHIFT 5**, posicione novamente o apontador em 8000 e pressione **L**.

No lado direito, aparece, em ASCII, TK90X, e, à esquerda, o código equivalente de cada letra nas posições de memória.

*** SYMBOL SHIFT Z**

Este comando é muito utilizado para a depuração de programas em assembler. Ele executa as instruções contidas no Contador de Programa (PC), os registradores são atualizados após cada instrução executada, e o apontador assume o valor do Contador de Programa (PC).

MODIFICANDO A MEMÓRIA

O conteúdo dos endereços do apontador de memória podem ser modificados se for entrado com um número hexadecimal seguido por um comando. Os dois dígitos hexadecimais menos significativos (se apenas um dígito entrar, o dígito da esquerda será um zero) entram no endereço corrente do apontador de memória e, então, o comando especificado é executado. Se o comando não é válido, é ignorado.

Sendo dado o comando **ENTER**, o valor entra na memória avançando uma vez

Ex: F3 ENTER

Sendo dado o comando **CAPS SHIFT 8**, o valor entra na memória e o apontador avança oito vezes.

Ex: 23 CAPS SHIFT 8
E5 M:8000

Sendo usado o comando **O** (desvio relativo), este coloca no endereço do apontador o valor do código e realiza um desvio igual ao valor do código.

Ex: 10

O endereço do apontador estará com 10, e seu deslocamento será de 10 vezes o endereço do apontador.

Sendo dado o comando "-" (**SYMBOL SHIFT 0**), o valor entra na memória, e o apontador permanece na mesma posição.

Ex: 2A-

Obs: Pode-se também dar **BREAK** ou **SPACE** após o código hexadecimal. Este código entra na memória como o exemplo acima. Outros comandos podem ser usados também: **X**, **CAPS SHIFT 7**, **M**, etc.

MODIFICANDO OS REGISTRADORES

Note que na tela principal existe um símbolo de pronto (>) ao lado esquerdo do registrador PC.

Para mudar este símbolo de pronto para outro, use o comando **SYMBOL SHIFT M**.

Para modificar o conteúdo do registrador, posicione o símbolo no registrador. Em seguida, digite o novo valor em hexadecimal, seguido de **SYMBOL SHIFT M**.

Obs: Os registradores SP e IR não podem ser modificados.

