

De um modo geral, a fluxogramação é uma técnica muito útil, principalmente no que concerne à documentação de programas, propiciando aos não-programadores condições, através de fórmulas padrões, de compreender a estrutura básica do programa. De certa forma, os fluxogramas não podem fornecer muito mais do que um esboço inicial do problema ou do programa e às vezes tendem a tornar-se confusos em função da quantidade de detalhes que se queira incluir ou transmitir.

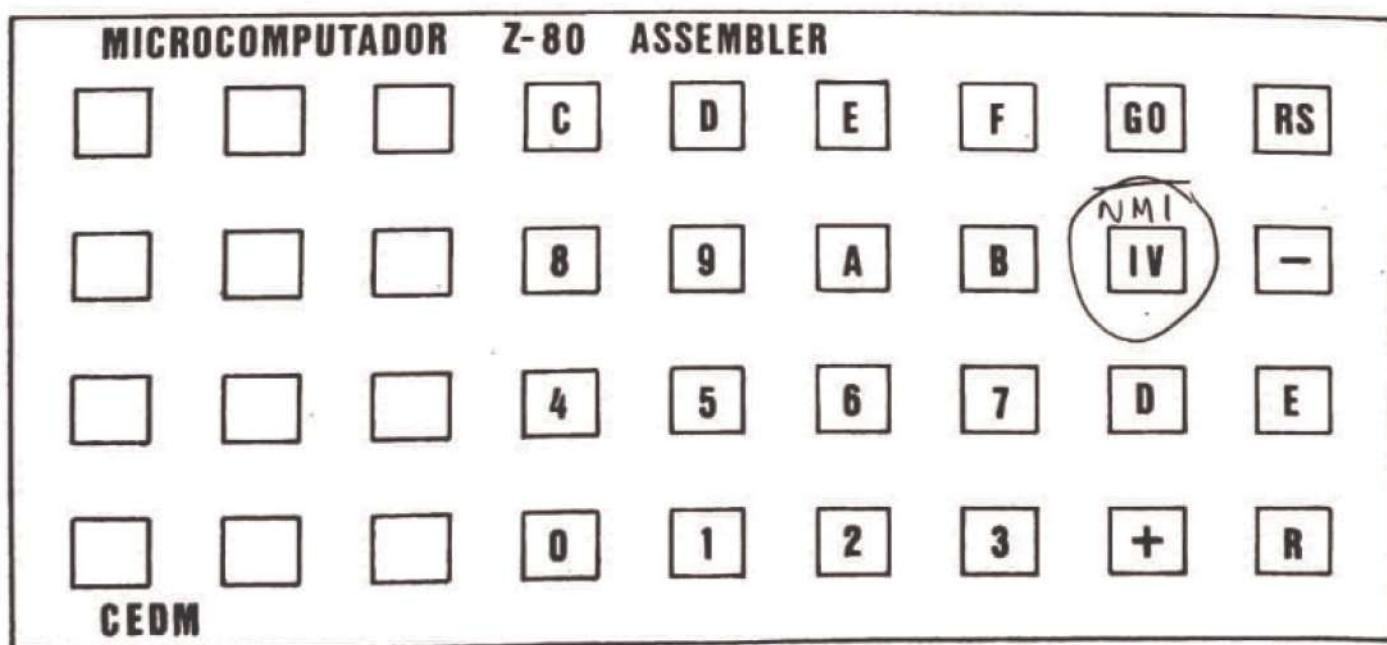
Finalizando o estudo dos fluxogramas, queremos deixar bem claro que a complexidade nem sempre é uma virtude, portanto, ao elaborar um fluxograma, seja cuidadoso e procure tornar as coisas o mais simples possível.

PROGRAMAÇÃO EM ASSEMBLER:

A linguagem Assembler é uma linguagem de baixo nível, isto é, próxima à linguagem de máquina, na qual nós entramos com as instruções de Assembler.

O método mais prático para se aprender a programar em linguagem Assembler é escrevendo programas em Assembler. Porém, antes de começarmos a escrever programas, ou seja, antes de darmos os primeiros passos, vamos passar a uma breve e sucinta descrição do teclado do microcomputador CEDM-80.

O teclado do microcomputador CEDM-80 é composto por 24 teclas, sendo 16 para numeração hexadecimal e 8 teclas de funções. A figura 12 nos mostra um lay-out do teclado do micro CEDM-80.



Visando a facilitar o estudo da função de cada tecla do microcomputador CEDM, adotaremos para o display a seguinte nomenclatura (figura 13):

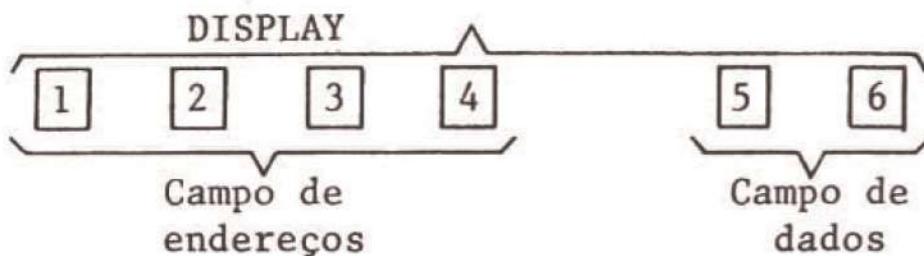


FIGURA 13

Tecla_D - Esta tecla nos possibilita verificar o conteúdo (ou valor) armazenado na posição de memória, mostrada no campo de endereços. Este valor é mostrado no 5º e 6º dígitos do display (campo de dados), sob a forma de um número hexadecimal. Ao ser pressionada, esta tecla permite, também, a substituição ou a alteração do conteúdo (o valor) da posição (ou endereço) de memória, contido no campo de endereços, com os algarismos entrando nesse campo no sentido da direita para a esquerda, indefinidamente de modo a empurrar os outros algarismos no mesmo sentido. Este fato facilita em muito a correção do valor em hexadecimal, quando este for digitado de forma errônea. Para corrigi-lo, basta colocar o valor correto, não havendo portanto, a necessidade de pressionar a tecla D novamente.

Cabe ressaltar o fato de que o valor mostrado no display, tenha ele sido ou não alterado, é o conteúdo real da respectiva posição de memória.

Tecla_E - Esta tecla tem por função a abertura do campo de endereços (1º, 2º, 3º e 4º dígitos). Ao ser pressionada, esta tecla faz com que o display apague totalmente, indicando permissão para a entrada do endereço (ou algarismos hexadecimais) cujos algarismos vão entrando da direita para a esquerda, podendo ser carregados indefinidamente, de modo a empurrar, sempre, os outros algarismos no mesmo sentido. Este fato evita termos que pressionar a tecla "E" quando da ocorrência de um erro de digitação, bastando, para a correção do erro, digitar o valor desejado, em seguida.

O endereço mostrado neste campo servirá para as seguin-

tes operações:

- vetor de interrupção;
- listagem e modificação do valor armazenado nessa posição de memória;
- endereço de partida para "rodar" um programa qualquer.

Tecla RS — Esta tecla, ao ser pressionada, inicializa o programa monitor, levando o registrador PC (Contador de Programa) à posição "0000" e colocando no display a seguinte mensagem:

" 0 0 0 0"

Esta mensagem indica que o microcomputador CEDM-80 já foi inicializado. Cabe ressaltar que o reset não altera o conteúdo das memórias, muito embora o conteúdo dos registradores (com exceção do registrador PC) possa ser, eventualmente, alterado.

Tecla GO — Ao ser pressionada, esta tecla faz com que o microcomputador execute (ou rode) o programa, a partir de um endereço válido, colocado no campo de endereços.

Tecla R — Esta tecla tem por função permitir o exame do conteúdo dos registradores, bem como a sua alteração. Toda vez que for pressionada, esta tecla irá mostrar no display o termo "A=XX", onde "A" refere-se ao acumular e "XX" é o seu conteúdo. Para verificarmos ou alterarmos o conteúdo dos outros registradores, basta pressionarmos a tecla "+" que eles irão aparecendo na ordem B, C, D, E, H, L, F e I.

Para alterar o conteúdo de algum dos registradores, basta "chamá-lo", pressionando-se a tecla "+" até que ele apareça no display e, em seguida, digitar o valor hexadecimal desejado. Note que este valor também irá entrar da direita para a esquerda.

Tecla IV — Esta tecla tem por função colocar um endereço válido (que servirá como vetor), toda vez que houver uma transição de nível lógico "0" para nível lógico "1" no NMI. Sempre que tal fato ocorrer, o programa irá partir da posição de memória, escolhida como vetor.

Tecla + - Esta tecla tem por função incrementar de uma posição o valor mostrado no campo de endereços, indicando o seu conteúdo no campo de dados.

Tecla - - A função desta tecla é semelhante à função da "tecla +", diferindo apenas no fato desta decrementar ao invés de incrementar.

Teclas 0 a F - Estas dezesseis (16) teclas correspondem à numeração hexadecimal.

Finalizando a descrição da função de todas as teclas, passaremos aos programas em linguagem Assembler, propriamente ditos. Esclarecemos, porém, que o modo como serão apresentados os primeiros programas tem por objetivo principal familiarizar o aluno com o microcomputador CEDM-80 e também com as regras básicas da programação em Assembler.

Para que o aluno possa compreender melhor os programas, aconselhamos que tenha sempre em mãos a lição "Microprocessador Z-80 III", na qual apresentamos o conjunto de instruções do microprocessador Z-80, classificadas em grupos, conforme suas funções operacionais.

PROGRAMAS EM ASSEMBLER:

O primeiro programa a ser apresentado é um programa que serve como teste do microcomputador CEDM-80 e que, ao mesmo tempo, nos permite verificar como a instrução HLT deixa o microcomputador na condição "halt" (condição de parada). Vejamos:

TECLA	DISPLAY	COMENTÁRIOS
RS	CEDM 80	Inicialização do micro.
E		O display fica totalmente apagado.
2001	2 0 0 1	
D	2 0 0 1 XX	XX - significa valor anterior.
76	2 0 0 1 76	76 - código de máquina da instrução HLT.
GO	Somente os pontos decimais se acenderão.

Observações:

- Toda vez que o micro CEDM-80 apresentar todos os pontos decimais do display acesos, isto indicará que o programa digitado foi processado (ou rodado).

- Cabe ressaltar que a área do mapa de memória destinada à memória RAM vai do endereço 2000 (inclusive) ao endereço 27FF (inclusive).

- Note que, ao ligar o microcomputador CEDM-80, este é automaticamente inicializado (ou resetado), devendo ser apresentada no display a mensagem " CEDM 80 ". Assim sendo, não é necessário pressionar a tecla de reset (RS) para iniciar o micro CEDM-80, quando este é ligado.

Ao executar um programa em Assembler, devemos, sempre que possível, ter uma "idéia" do que esperamos que ocorra ou do resultado que buscamos.

O programa a seguir nos mostra a execução de mais duas instruções do Z-80: LD r,n e ADC A,reg.

A figura 14 nos mostra um diagrama de blocos do programa a ser executado.

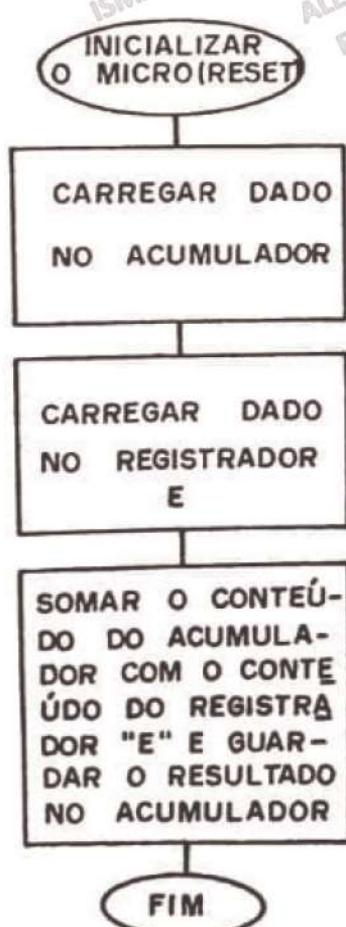


FIGURA 14

Suponhamos então, que queremos carregar o acumulador com o valor E3 e o registrador "E" com o valor 0A. Em seguida, através da instrução ADC A,E, nós iremos somar esses dois valores, obtendo então como resultado o valor ED, o qual deverá ser guardado no acumulador. Veja:

$$\begin{array}{r} \text{E3} = 1110\ 0011 \\ \text{0A} = 0000\ 1010 \\ \hline \end{array} \quad + \quad \begin{array}{c} \\ \\ \swarrow \quad \searrow \\ \text{E} \quad \text{D} \end{array}$$

Vamos agora montar o programa-fonte utilizando para tanto o mnemônico das instruções.

Programa-fonte

```
LDA, E3
LDE, 0A
ADC A,E
HALT
```

Uma vez obtido o programa-fonte, vamos então escrever o programa-objeto:

Programa-objeto

Posição de memória	Código de operação (Hexadecimal)
--------------------	-------------------------------------

2000	—	3E	LD A,E3
2001	—	E3	
2002	—	1E	LD E,0A
2003	—	0A	
2004	—	8B	ADC A,E
2005	—	76	
			HALT

Vejamos agora como digitar o programa no micro CEDM-80.

TECLA	DISPLAY	COMENTÁRIOS
RS	CEDM 80	Inicialização do micro
E		

TECLA	DISPLAY	COMENTÁRIOS
2 0 0 0	2 0 0 0	
D	2 0 0 0 XX	XX - significa valor anterior.
3 E	2 0 0 0 3E	
+	2 0 0 1 XX	
E 3	2 0 0 1 E3	
+	2 0 0 2 XX	
1 E	2 0 0 2 1E	
+	2 0 0 3 XX	
0 A	2 0 0 3 0A	
+	2 0 0 4 XX	
8 B	2 0 0 4 8B	
+	2 0 0 5 XX	
7 6	2 0 0 5 76	Final do programa.

Terminada a digitação do programa, ele está pronto para ser rodado. Para tanto, é preciso dar um endereço de partida válido. No nosso caso, o endereço de partida é a posição de memória 2000, pois é a partir dessa posição (ou endereço) de memória que começamos a digitar o nosso programa. Devemos então digitar o seguinte:

TECLA	DISPLAY	COMENTÁRIOS
E		
2 0 0 0	2 0 0 0	Endereço válido para que o programa seja rodado.
GO	Programa rodou.

AGRADECIMENTOS
SMAR RUSSANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMINO
EMILIO AOI

Feito isto, precisamos saber o resultado. No nosso caso, sabemos que o resultado da soma do conteúdo do acumulador com o conteúdo do registrador E está guardado no próprio acumulador. Portanto, para que possamos visualizar o resultado, basta pressionarmos a tecla R, vejamos:

TECLA	DISPLAY	COMENTÁRIOS
RS		
R	A = ED	O conteúdo do acumulador é ED, que confere com o resultado esperado.

Obs.: Caso fosse preciso verificar o conteúdo dos outros registradores, bastaria pressionar a tecla $\boxed{+}$, seguidamente, passando pelos registradores: B, C, D, E, H, L, F e I.

Como o programa apresentado é relativamente curto, ou seja, de poucas linhas, para endereçarmos a posição de memória válida, 2000, poderíamos ter lançado mão da tecla $\boxed{-}$. Assim, ao terminarmos a digitação do programa, bastaria voltarmos, pressionando a tecla $\boxed{-}$ 5 (cinco) vezes consecutivas, para atingirmos a posição 2000. A partir daí procede-se normalmente. Vejamos:

TECLA	DISPLAY	COMENTÁRIOS
:	:	:
7 6	2 0 0 5 76	Final do programa
- - -	:	
- -	2 0 0 0 3E	
GO	

Para verificar o resultado que deve estar armazenado no acumulador, proceda da mesma maneira que explicamos anteriormente.



CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MP-23

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) O fluxograma também é conhecido como:

- a) diagrama de blocos;
- b) diagrama de fluxo;
- c) diagrama seqüencial;
- d) N.R.A.

2) A fluxogramação se presta ao propósito de:

- a) auxiliar na visualização do melhor meio de se resolver um problema;
- b) ilustrar um sistema proposto para análise e verificação;
- c) registrar processos e atividades a serem usados como um modelo para propósitos de operação;
- d) todos os itens estão corretos.

3) Os símbolos gráficos, utilizados nos fluxogramas foram normalizados pela:

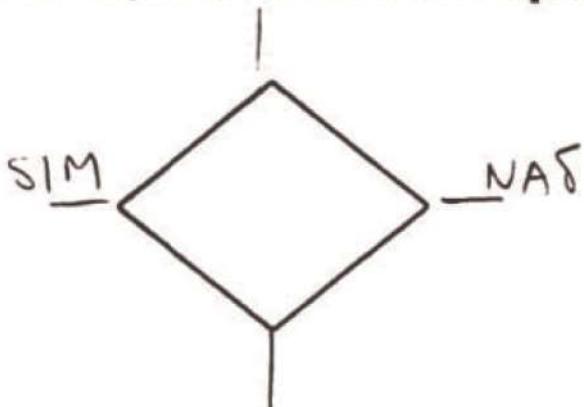
- a) ABNT;
- b) ECMA;
- c) INPM;
- d) N.R.A.

4) O símbolo, ilustrado a seguir, indica uma operação de:



- a) Processamento genérico;
- b) Decisão;
- c) Entrada/Saída;
- d) Terminal.

5) O símbolo, ilustrado a seguir, indica uma operação de:



- a) Entrada/Saída;
- b) Terminal;
- c) Conector;
- d) Decisão.

6) Os tipos básicos de estrutura que podemos encontrar dentro de um fluxograma são:

- a) Desvio, Laço e Diagrama de blocos;
- b) Desvio, Laço e Seqüencial;
- c) Desvio, Laço e Estrutura não seqüencial;
- d) N.R.A.

7) A repetição de uma instrução ou de um conjunto de instruções até o instante em que ocorra uma condição que determine o seu término é considerado como sendo um:

- a) Laço ou Loop;
- b) Desvio condicional;
- c) Desvio incondicional;
- d) N.R.A.

8) Os dígitos que compõem o campo de endereços do display do micro CEDM-80 são:

- a) 3, 4, 5 e 6;
- b) 5 e 6;
- c) 2, 3, 4 e 5;
- d) 1, 2, 3 e 4.

9) A tecla E tem por função a abertura do:

- a) campo de dados;
- b) registrador F;
- c) campo de endereços;
- d) N.R.A.

AGRADECIMENTOS
ISMAR RUSSANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMINO
EMILIO AOI

10) No programa, cujo diagrama de blocos se encontra na figura 14, se substituíssemos o valor a ser carregado no acumulador por AA e o valor a ser carregado no registrador E por 55, ao fazermos rodar o programa, obteríamos o seguinte resultado:

- a) AF;
- b) FF;**
- c) 00;
- d) 1B.

$$\begin{array}{r}
 \text{AA} \quad 1010 \ 1010 \\
 \text{55} \quad 0101 \ 0101 \\
 \hline
 1111 \ 1111
 \end{array}$$

AGRADECIMENTOS
 ISMAR RUSSANO - DOAÇÃO DO MATERIAL
 EDIÇÃO
 ALEX FIRMINO
 EMILIO AOI

$$\begin{array}{r}
 1010 \ 0101 \\
 0101 \ 1010 \\
 \hline
 1111 \ 1111
 \end{array}$$



CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LIÇÃO MP-24

MANUAL DE PRÁTICA MP-24

INTRODUÇÃO

Dando continuidade à lição anterior (MP-23), apresentaremos, nesta lição, uma série de programas, cujo objetivo é demonstrar o uso de algumas instruções do microprocessador Z-80. Nestes programas, as indicações referentes às teclas e ao display foram eliminadas, uma vez que se tornam desnecessárias daqui para frente, porém, para facilitar a compreensão dos programas, estes serão apresentados sob a forma de programa fonte (mnemônicos) e programa objeto, acompanhado de comentários.

PROGRAMA 1

O objetivo deste(s) programa(s) é o de demonstrar a execução da instrução CPL. Esta instrução tem, por função complementar, o conteúdo do acumulador, sendo que a operação de complementação é, na verdade, uma operação NOT (NO). O resultado é guardado no próprio acumulador.

Vejamos uma aplicação simples da instrução CPL. O programa fonte é mostrado na figura 1.

VEÍDA
PROIBIDA

AGRADECIMENTOS
ISMAR RUSSANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMINO
EMILIO AOI

PROGRAMA FONTE

LDA, 00
CPL
HLT

FIGURA 1

Na figura 2, apresentamos o programa objeto.

PROGRAMA OBJETO

Posição de memória Código de operação
(Hexadecimal)

Comentários

2010	3E	carregar o acumulador
2011	00	valor colocado no acumulador
2012	2F	complementa conteúdo do acumulador
2013	76	fim

FIGURA 2

Para rodar o programa (após o mesmo ter sido digitado), basta pressionar a tecla "E", digitar o endereço (válido) 2010 e, em seguida, pressionar a tecla "GO". Nos dois campos do display, deverá aparecer uma série de pontos indicando que o programa rodou. Para verificarmos o resultado, devemos pressionar a tecla "RS" e, em seguida, pressionar a tecla "R", devendo aparecer no visor "A = XX", onde XX é o complemento do valor que foi colocado no acumulador.

A instrução CPL, por si só, realiza o que nós chamamos de "complemento de 1", ou seja, complementa o conteúdo do acumulador. Entretanto, se associarmos a instrução CPL à instrução INC A, poderemos obter o "complemento de 2" do conteúdo do acumulador. A figura 3 nos mostra o programa fonte que nos permite realizar o complemento de 2, de um valor guardado no acumulador. Vamos supor que este valor seja o mesmo do programa anterior, ou seja, "00".

PROGRAMA FONTE

LDA, 00
CPL
INC A
HLT

FIGURA 3

A figura 4, por sua vez, nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2010	3E	-carregar o acumulador
2011	00	-valor colocado no acumulador
2012	2F	-complementar o acumulador
2013	3C	-incrementar o acumulador
2014	76	-fim

FIGURA 4

Para rodar o programa e verificar o resultado, o qual se encontra guardado no acumulador, procede-se da mesma forma que no programa anterior.

Este programa executa o "complemento de 2" do conteúdo do acumulador, porém o microprocessador Z-80 possui uma instrução específica para realizar esta operação que é a instrução NEG.

PROGRAMA 2

Este programa tem por objetivo mostrar a execução da instrução NEG, a qual é constituída por dois bytes. Na figura 5, apresentamos o programa fonte.

PROGRAMA FONTE

```
LDA, E8
NEG
HLT
```

FIGURA 5

Na figura 6, temos o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2001	3E	- carregar o acumulador
2002	E8	- valor colocado no acumulador
2003	ED	- 1º byte da instrução NEG
2004	44	- 2º byte da instrução NEG
2005	76	- fim

FIGURA 6

Rode o programa e verifique o resultado. Apenas, a título de curiosidade, digite novamente o programa, substituindo o valor colocado no acumulador E8 por 00. Compare este novo resultado com o resultado do programa anterior (figura 4).

PROGRAMA 3

Este programa tem por finalidade demonstrar a execução da instrução AND, a qual faz parte do grupo de instruções lógicas.

O programa fonte é mostrado na figura 7, sendo este capaz de realizar a função AND (E) entre o conteúdo do acumulador e o conteúdo do registrador B, guardando o resultado final no próprio acumulador.

PROGRAMA FONTE

```
LDA, AF
LDB, 5A
AND B
HLT
```

FIGURA 7

A figura 8 nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	3E	carrega o acumulador (A)
2001	AF	valor colocado no acumulador
2002	06	carrega o registrador B
2003	5A	valor colocado no registrador B
2004	A0	executa a função AND entre A e B
2005	76	fim

FIGURA 8

Para verificar o resultado, proceda da mesma forma que nos programas anteriores.

Uma vez rodado o programa e verificado o resultado, refaça o programa, substituindo os valores colocados no acumulador e no registrador B, isto é muito importante, pois o aluno passa a assimilar melhor a matéria.

PROGRAMA 4

O programa 4 foi desenvolvido para demonstrar a execução da instrução lógica OR (OU). Para este caso, iremos reaproveitar os mesmos dados do programa 3, mudando apenas a instrução lógica. A figura 9 nos mostra o programa fonte.

PROGRAMA FONTE

```
LD A, AF
LD B, 5A
OR B
HLT
```

FIGURA 9

Na figura 10, apresentamos o programa objeto pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2020	3E	carrega o acumulador (A)
2021	AF	valor colocado no acumulador
2022	06	carrega o registrador B
2023	5A	valor colocado no registrador B
2024	B0	executa a função OR entre A e B
2025	76	fim

FIGURA 10

Para rodar o programa, após o mesmo ter sido digitado, pressione a tecla "E" e digite o endereço 2020. Em seguida, pressione a tecla "GO". Para se verificar o resultado, uma vez o programa rodado, pressione a tecla "RS" e, em seguida, pressione a tecla "R" e o conteúdo do acumulador será mostrado no display.

As observações feitas no final do programa 3, também são válidas para este programa.

PROGRAMA 5

O presente programa tem por objetivo demonstrar a execução da instrução XOR (HL), a qual realiza a lógica OR EXCLUSIVA (OU EXCLUSIVA) entre o conteúdo da posição de memória, endereçada pelo par de registradores HL e o conteúdo do acumulador. O resultado é guardado (ou carregado) no próprio acumulador.

A figura 11 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LDA, AA
LD HL, 72
XOR (HL)
HLT
```

FIGURA 11

A figura 12 nos mostra o programa objeto pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2020	3E	carrega o acumulador (A)
2021	AA	valor colocado no acumulador
2022	36	carrega o registrador (B)
2023	72	valor colocado no registrador B
2024	AE	executa a função XOR entre A e B
2025	76	fim

FIGURA 12

Uma vez digitado o programa, rode o mesmo e verifique o resultado final. Feito isto, execute novamente este programa, porém modifique os dados para que possa fixar bem o funcionamento da operação em questão.

PROGRAMA 6

Este programa tem por objetivo demonstrar a execução da instrução RLC, a qual desloca todos os bits de um byte de dados de uma posição para a esquerda, num movimento de rotação. Assim, o bit 7 é colocado no lugar do bit 0, enquanto que os bits de 0 a 6 são deslocados de uma posição para a esquerda. O bit 7 é, também, copiado no carry.

O programa fonte a ser executado é mostrado na figura 13.

PROGRAMA FONTE

```
LD D, 92
RLC D
HLT
```

FIGURA 13

A figura 14 nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	16	carrega o registrador D
2001	92	valor colocado no registrador D
2002	CB	1º byte da instrução RLC D
2003	02	2º byte da instrução RLC D
2004	76	fim

FIGURA 14

O resultado final encontra-se guardado no registrador D.

Para verificarmos se o bit 7 foi gravado no carry, é necessário "chamarmos" o registrador F (Flag) e, em seguida, transformar o seu conteúdo, que é apresentado na forma hexadecimal, para a forma binária. O bit carry é representado pelo primeiro bit da direita, conforme nos mostra a figura 15.

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

FIGURA 15

bit carry

PROGRAMA 7

Este programa foi desenvolvido para demonstrar a execução da instrução de rotação RRC. Esta instrução é semelhante à instrução RLC, exceto quanto ao sentido de rotação, ou seja, a instrução RRC realiza a rotação para a direita. Neste caso, o bit "0" é gravado (ou copiado) no carry.

A figura 16 nos mostra o programa fonte a ser executado, o qual foi desenvolvido de modo a utilizar os mesmos dados e registros do programa anterior, permitindo, assim, que o aluno compare os resultados finais desses programas.

PROGRAMA FONTE

```
LD D, 92
RRC D
HLT
```

FIGURA 16

O programa objeto a ser digitado é apresentado na figura 17.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	16	carrega o registrador D
2001	92	valor colocado no registrador D
2002	CB	1º byte da instrução RRC D
2003	OA	2º byte da instrução RRC D
2004	76	fim

FIGURA 17

Para verificação do resultado e do valor gravado (ou copiado) no bit carry, procede-se de forma semelhante à descrita para o programa anterior.

PROGRAMA 8

O presente programa foi desenvolvido com vistas a demonstrar a execução da instrução de rotação SLA. Esta instrução desloca todos os bits de uma posição para a esquerda, reseta (zera) o bit "0" e grava ou copia o bit 7 no carry.

Esta instrução nos permite executar a operação de multiplicação por alguma potência de dois (2). Para facilitar a compreensão, vamos fazer uma analogia com o sistema decimal. Na base 10 (sistema decimal), quando multiplicamos um número qualquer por 10, simplesmente deslocamos todos os algarismos uma casa (ou posição) para a esquerda e acrescentamos um zero no final. Se multiplicarmos um número qualquer por 100, todos os algarismos serão deslocados de duas posições para a esquerda o que significa que esse mesmo número foi multiplicado por 10, duas vezes seguidas. Assim, ao executarmos a instrução SLA, estaremos multiplicando o número por 2 e, ao realizá-la duas vezes sucessivas, estaremos multiplicando o número por 4, assim por diante.

O programa fonte a ser executado é apresentado na figura 18.

PROGRAMA FONTE

```
LDA, 7F
SLA A
HLT
```

FIGURA 18

A figura 19 nos mostra o programa objeto a ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2010	3E	- carrega o acumulador (A)
2011	7F (XX)	- valor colocado no acumulador
2012	CB	- 1º byte da instrução SLA A
2013	27	- 2º byte da instrução SLA A
2014	76	- fim

FIGURA 19

Após digitar o programa, rode o mesmo e verifique o resultado final.

Lembre-se de que o maior número positivo que se pode representar por intermédio de 8 bits (utilizando-se o sistema binário e não a representação em complemento de 2) é 256 (decimal). Assim, quando o número original que irá sofrer a rotação for maior que 128 (decimal), o resultado do deslocamento não será duas vezes o número, mas sim o número multiplicado por 2, menos 256, sendo o bit carry levado para "1" (binário) para indicar este fato.

Na figura 20, apresentamos uma tabela com vários valores de XX (valor a ser colocado no acumulador), os quais deverão ser colocados no programa apresentado na figura 19, um por vez e, em seguida, o programa deverá ser rodado. Verifique o resultado final para todas as situações apresentadas, inclusive o carry bit. Note que as linhas 1 e 7 da tabela já estão preenchidas com os respectivos resultados, cabendo ao aluno a tarefa de preencher os espaços correspondentes aos resultados para as demais linhas.

XX(H)	Resultado		(F)
	Acumulador (A)	Carry (F)	
linha 1 05	0A	0	1000
linha 2 0A	14	1	0100
linha 3 78	F0	0	1100
linha 4 80	00	1	0101
linha 5 81	02	1	0011
linha 6 01	02	0	0000
linha 7 82	04	1	0000

FIGURA 20

AC	1000
AF	0100
AC	1100
AF	0101
AC	0011
AF	0000
AC	0000

O fato do carry ser igual a 1 nos permite detectar se houve um estouro, ou seja, resultado maior do que 255. Caso o carry seja igual a 1, o número que aparecer no display (transformado para decimal) mais (+) o 256 (decimal) será o valor real do número (ou resultado).

PROGRAMA 9

Este programa foi desenvolvido para demonstrar a execução da instrução SRA. Esta instrução opera de forma semelhante à instrução SLA, porém o deslocamento se dá para a direita, colocando o bit "0" no carry. O bit 7 não é alterado, o que não nos permite "dividir" números positivos maiores do que 127 (decimal) ou 7F (hexadecimal), mas, por outro lado, nos possibilita dividir números negativos de -1 a -128 (decimais).

De uma forma resumida, podemos dizer que esta instrução divide o conteúdo do registrador especificado por 2 e coloca o resto da divisão no bit carry.

A figura 21 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LDA, OF
SRA A
HLT
```

FIGURA 21

A figura 22, por sua vez, nos mostra o programa objeto, pronto para ser executado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2015	3E	-carrega o acumulador
2016	0F	-valor colocado no acumulador
2017	CB	-1º byte da instrução SRA A
2018	2F	-2º byte da instrução SRA A
2019	76	-fim

FIGURA 22

Uma vez digitado o programa, rode-o e verifique o resultado apresentado no acumulador (A) e o valor do resto da operação que é apresentado no registrador F (Flag), mais especificamente, no bit carry.

PROGRAMA 10

Este programa tem por objetivo mostrar o uso da instrução INC B, a qual incrementa o conteúdo do registrador B de uma unidade.

Na figura 23, temos o programa fonte a ser executado.

PROGRAMA FONTE

```
LD B, 01
INC B
HLT
```

FIGURA 23

A figura 24 nos mostra o programa objeto a ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	06	carrega o registrador B
2001	01	valor colocado no registrador B
2002	04	incrementa o registrador B
2003	76	fim

7 : 02

FIGURA 24

Depois de digitar o programa, este deve ser rodado. Para se verificar o resultado, após o programa ter sido rodado, basta "chamar" o registrador B e ler o valor nele armazenado.

PROGRAMA 11

Este programa tem por objetivo demonstrar a execução da instrução ADD A,r. Como sabemos, esta instrução soma o conteúdo do registrador "r" ao conteúdo do acumulador e o resultado fica no acumulador. A figura 25 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LD A, OF
LD L, FO
ADD A, L
HLT
```

FIGURA 25

Na figura 26, temos o programa objeto pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2015	3E	carrega o acumulador (A)
2016	OF	valor colocado no acumulador
2017	2E	carrega o registrador L
2018	F0	valor colocado no registrador L
2019	85	soma o conteúdo de A com o de L
201A	76	fim

FIGURA 26

Após digitar o programa, rode-o e, em seguida, verifique o resultado, o qual se encontra guardado no acumulador.

Digite novamente este programa (figura 26), porém substitua o valor colocado no acumulador e no registrador L por "FF". Note que o resultado apresentado no acumulador, após o programa ter sido rodado, é FE e o carry é igual a 1.

PROGRAMA 12

Este programa tem por objetivo demonstrar a execução da instrução ADD A,n. Como sabemos, esta instrução soma a constante "n" ao conteúdo do acumulador, sendo que o resultado fica guardado no próprio acumulador. A figura 27 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LDA, XX
ADD, n
HLT
```

FIGURA 27

A figura 28 nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2011	3E	carrega o acumulador
2012	1B	valor colocado no acumulador
2013	C6	soma a constante ao conteúdo do acumulador
2014	05	valor da constante
2015	76	fim

FIGURA 28

Após digitar o programa, "rode" o mesmo e verifique o resultado, o qual se encontra armazenado no acumulador.

PROGRAMA 13

Este programa foi desenvolvido para demonstrar a execução da instrução ADC A,r. A instrução ADC A,r soma o conteúdo do carry bit mais (+) o conteúdo do registrador r com o conteúdo do acumulador. O resultado desta operação é guardado no próprio acumulador.

Neste exemplo, iremos escolher o registrador "r" como sendo o registrador E.

A figura 29 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LD A,  
LD E,  
ADC A,E  
HLT
```

FIGURA 29

A figura 30, por sua vez, apresenta-nos o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)
--------------------	----------------------------------

2000	3E	carrega o acumulador
2001	1A	valor colocado no acumulador
2002	1E	carrega o registrador E
2003	90	valor colocado no registrador E
2004	8B	executa a instrução ADC A,E
2005	76	fim

FIGURA 30

Antes de rodar, ou mesmo de digitar o programa, "chame" o registrador F e coloque no mesmo o valor "00", o que nos garante o carry igual a zero (0). Em seguida, providencie para que o programa seja rodado e verifique o resultado.

Uma vez verificado o resultado para a condição em que o carry é igual a zero (0), vamos "chamar" novamente o registrador F e colocar nele o valor 01, valor este que nos garante que o carry seja igual a "1". Rode novamente o programa e verifique o novo resultado que está armazenado no acumulador.

No primeiro caso, para F = 00, o aluno deverá encontrar o valor AA. Para o segundo caso, em que F = 01, o aluno deverá encontrar como resultado o valor AB.

Experimente rodar o programa com outros valores, porém observando sempre o valor colocado no registrador F, de modo a garantir que o carry seja 0 ou 1, conforme se desejar.

PROGRAMA 14

O presente programa tem por objetivo demonstrar a execução da instrução SUB r. Esta instrução, como sabemos, subtrai o conteúdo do registrador r do conteúdo do acumulador, armazena o resultado da operação no próprio acumulador.

A figura 31 nos mostra o programa fonte a ser executado.

PROGRAMA FONTE

```
LD A, FF
LD B, OF
SUB B
HLT
```

FIGURA 31

A figura 32, por sua vez, nos mostra o programa objeto, pronto para ser executado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	3E	carregar o acumulador
2001	FF	valor colocado no acumulador
2002	06	carrega o registrador B
2003	0F	valor colocado no registrador B
2004	90	executa a instrução SUB B
2005	76	fim

FIGURA 32

Uma vez digitado o programa, este deve ser rodado. Para verificar o resultado, basta "chamar" o acumulador (ou simplesmente pressionar a tecla "R").

Vamos, então, conferir o resultado. Para isso, vamos resolver a operação de subtração do conteúdo do registrador B (OF) do conteúdo do acumulador (FF), vejamos:

(Hexadecimal)	(binário)
FF	= 1111 1111
OF	= 0000 1111
	<hr/>
	$\underbrace{1111}_{F} \underbrace{0000}_0$

Temos, portanto, como resultado o hexadecimal F0. Este valor (F0) é o valor que deve aparecer no display, quando formos verificar o resultado do programa.

Na próxima lição, apresentaremos mais uma série de programas para o microcomputador CEDM-80. Sugerimos ao aluno que, a partir dos programas apresentados, passe a criar pequenos programas de teste para as instruções que não foram testadas, pois, somente assim, o aluno passará a assimilar melhor o funcionamento do micro CEDM-80.

PROIBIDA

AGRADECIMENTOS
ISMAR RUSSANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMINO
EMILIO AOI



CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MP-24

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

1) Ao verificarmos o resultado do programa apresentado na figura 2, observaremos que o conteúdo do acumulador será igual a:

- a) 00
b) 1F

- c) FF
d) BA

2) O resultado do programa apresentado na figura 4 é:

- a) A = 00
b) A = FF

- c) A = 1F
d) A = F7

3) Após rodar o programa apresentado na figura 6, o conteúdo do acumulador será igual a:

- a) 08
b) 8A

- c) E8
d) 18

4) Após ter sido executado o programa apresentado na figura 8, será exibido como conteúdo do acumulador o valor:

- a) A0;
 b) OA;

- c) 00;
 d) 1A.

5) Após a execução do programa apresentado na figura 12, o acumulador passa a exibir o valor:

- a) d8;
 b) 08;

- c) AA;
 d) FF.

6) O resultado do programa apresentado na figura 17 é igual a:

- a) registrador D = 18 e registrador F = AA;
 b) registrador A = 08 e registrador F = 49;
 c) registrador D = 49 e registrador F = 08;
 d) registrador B = 49 e registrador F = 18.

7) A tabela da figura 20, após rodarmos o programa da figura 19 para os diversos valores de XX(H), deverá ter as colunas de resultado assim preenchidas:

a)

XX(H)	Resultados	
	Acumulador(A)	Carry(F)
05	0A	0
0A	14	0
78	F0	0
80	00	1
81	02	1
01	02	0
82	04	1

b)

XX(H)	Resultados	
	Acumulador(A)	Carry(F)
05	0A	0
0A	41	0
78	0F	0
80	AA	0
81	10	0
01	02	0
82	04	1

c)

XX(H)	Resultados	
	Acumulador(A)	Carry(F)
05	0A	0
0A	41	0
78	FA	0
80	F0	1
81	02	1
01	02	0
82	04	1

d)

XX(H)	Resultados	
	Acumulador(A)	Carry(F)
05	0A	0
0A	14	1
78	F0	1
80	FF	1
81	20	0
01	20	0
82	04	1

8) O programa apresentado na figura 22, após ter sido rodado, apresentará como resultado:

- a) A = F7 e F = 00;
- b) D = 27 e F = 01;
- c) A = DC e F = A1;
- d) A = 07 e F = 01.

9) Após rodarmos o programa apresentado na figura 24, o registrador B passa a exibir como resultado o valor hexadecimal:

- a) AA;
- b) 02;
- c) 9C;
- d) 71.

10) Após ter sido rodado o programa da figura 28, será apresentado como resultado:

- a) 9C;
- b) AD;
- c) 20;
- d) N.R.A.

VENDA PROIBIDA

ISMAR RUSSAN / BRADECIMENTOS
 EDIÇÃO 1 / ALEX FIRMINO
 EMILIO AOI

$$\begin{array}{r}
 1B + 0S = 2\phi \\
 \hline
 0001\mid 01\mid 1\quad 1B \\
 +\quad 0000\mid 01\mid 0\quad 0S \\
 \hline
 0010\mid 00\mid 0\quad 2\phi
 \end{array}$$



CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LICÃO MP-25

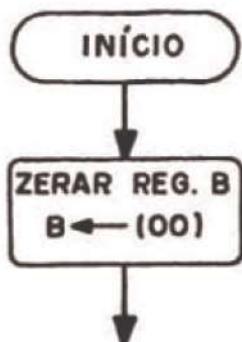
MANUAL DE PRÁTICA MP-25

INTRODUÇÃO

Nesta lição apresentaremos uma série de programas, cujo objetivo é familiarizar o aluno com as técnicas de programação em Assembler.

PROGRAMA 1

Este programa tem por objetivo apresentar a rotina de conversão de um número hexadecimal para o código BCD. Como o nosso microcomputador possui apenas dois dígitos no campo de dados, isto nos leva a concluir que o número hexadecimal a ser convertido não pode ser superior a $(63)_{16}$, pois, com dois dígitos em BCD, o maior número que conseguimos representar é $(99)_{10}$. A figura 1 nos mostra o fluxograma do programa.



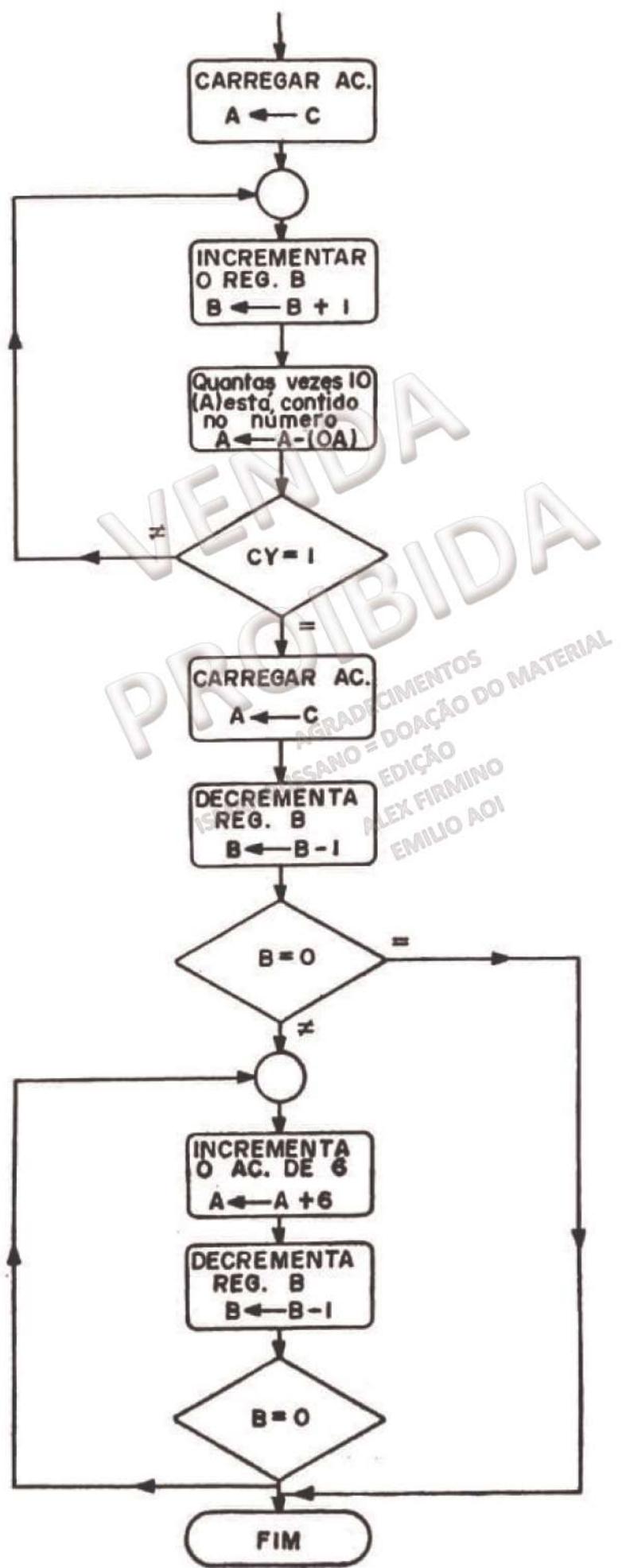


FIGURA 1

A figura 2, por sua vez, nos apresenta o programa escrito sob a forma de mnemônico, ou seja, o programa fonte.

PROGRAMA FONTE

```

LD B,00
LD A,C
INC B
SUB OA
JR NC,FB
LD A,C
DEC B
RET Z
ADD A,06
DJNZ ,FC0
HLT

```

FIGURA 2

Na figura 3, apresentamos o programa objeto, pronto para ser digitado em seu microcomputador CEDM-80.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2030	06	Iniciar o registrador B com 00
2031	00	
2032	79	Transferir o conteúdo de C para o acumulador.
2033	04	Incrementar o registrador B.
2034	D6	Verificar quantas vezes 10 está contido no número a ser transformado.
2035	0A	
2036	30	O programa prossegue no endereço relativo.
2037	FB	
2038	79	Transferir o resultado para o acumulador
2039	05	Decrementar o registrador B.
203A	C8	
203B	C6	Some seis (6) no acumulador.

Posição de memória	Código de operação (Hexadecimal)	Comentários
203C	06	
203D	10	O programa prossegue se o registrador B for diferente de zero.
203E	FC	
203F	76	Fim (HLT)

FIGURA 3

Para verificar o funcionamento do programa, o aluno deverá carregar o acumulador C com os números indicados na tabela 1 e anotar o respectivo resultado, o qual deverá ser lido no acumulador. A tabela 1 se encontra na figura 4.

Número a ser carregado no registrador C	Resultado (Acumulador)
63	99
10	16
A	10
1F	31
3A	58
5B	91

FIGURA 4

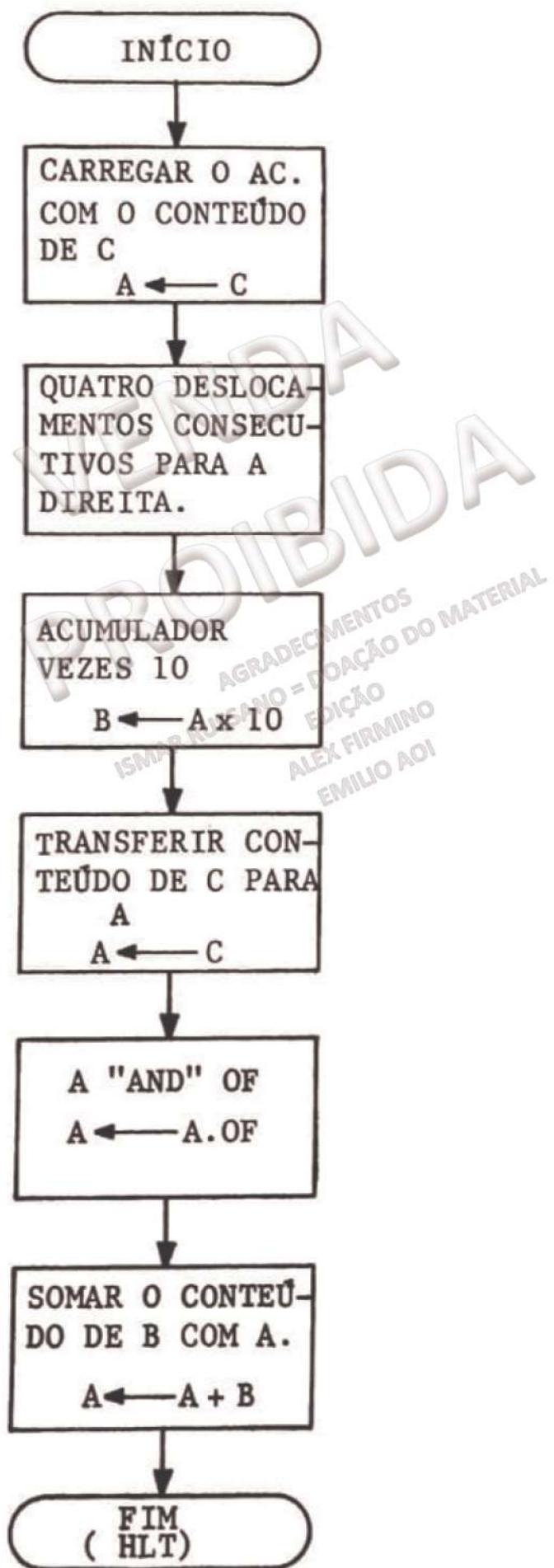
TABELA 1

PROGRAMA 2

Este programa tem por objetivo apresentar a rotina de conversão de um número escrito em BCD para hexadecimal. Este programa, comparado com o programa anterior, não apresenta praticamente nenhuma limitação, uma vez que dois dígitos em BCD nos permitem escrever no máximo o decimal 99 (equivalente ao hexadecimal 63), não ultrapassa a capacidade máxima de um registrador que é $(FF)_{16}$.

Neste programa, o número a ser convertido deverá ser carregado ou armazenado no registrador C, enquanto que o resultado deverá ser encontrado no acumulador. A figura 5 nos mostra o fluxograma do programa.

FIGURA 5



A figura 6 nos mostra o programa fonte.

PROGRAMA FONTE

```

LD A,C
SRL A
SRL A
SRL A
SRL A
ADD A,A
LD B,A
ADD A,A
ADD A,B
LD B,A
LD A,C
AND OF
ADD A,B
HLT

```

FIGURA 6

A figura 7, por sua vez, nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2040	79	Número a ser convertido.
2041	CB	SRL A
2042	3F	
2043	CB	SRL A
2044	3F	
2045	CB	SRL A
2046	3F	
2047	CB	SRL A
2048	3F	
2049	87	Multiplicar o conteúdo do Ac. por 2.
204A	47	
204B	87	Multiplicar o conteúdo do Ac. por 4.
204C	87	Multiplicar o conteúdo do Ac. por 8.

Posição de memória	Código de operação (Hexadecimal)	Comentários
204D	80	Conteúdo do Ac. vezes 10.
204E	47	
204F	79	
2050	E6	Função AND Ac. com OF.
2051	OF	
2052	80	Soma Ac. com B.
2053	76	Fim (HLT)

FIGURA 7

Para verificar o funcionamento do programa, o aluno deverá carregar o acumulador C com os números indicados na tabela 2 e anotar o respectivo resultado, o qual deverá ser lido no acumulador. A tabela 2 é mostrada na figura 8.

Número a ser carregado no registrador C	Resultado (Acumulador)
99	63
63	3F
39	27
55	37
44	2C
15	0F
29	1D
84	54
77	4D
90	5A

FIGURA 8

TABELA 2

OBS.: O aluno deverá completar os espaços vazios da tabela 2.

PROGRAMA 3

Este programa tem por objetivo mostrar como se realiza o processo de soma no microcomputador CEDM-80. O programa é relativamente simples. A figura 9 nos mostra o fluxograma do programa.

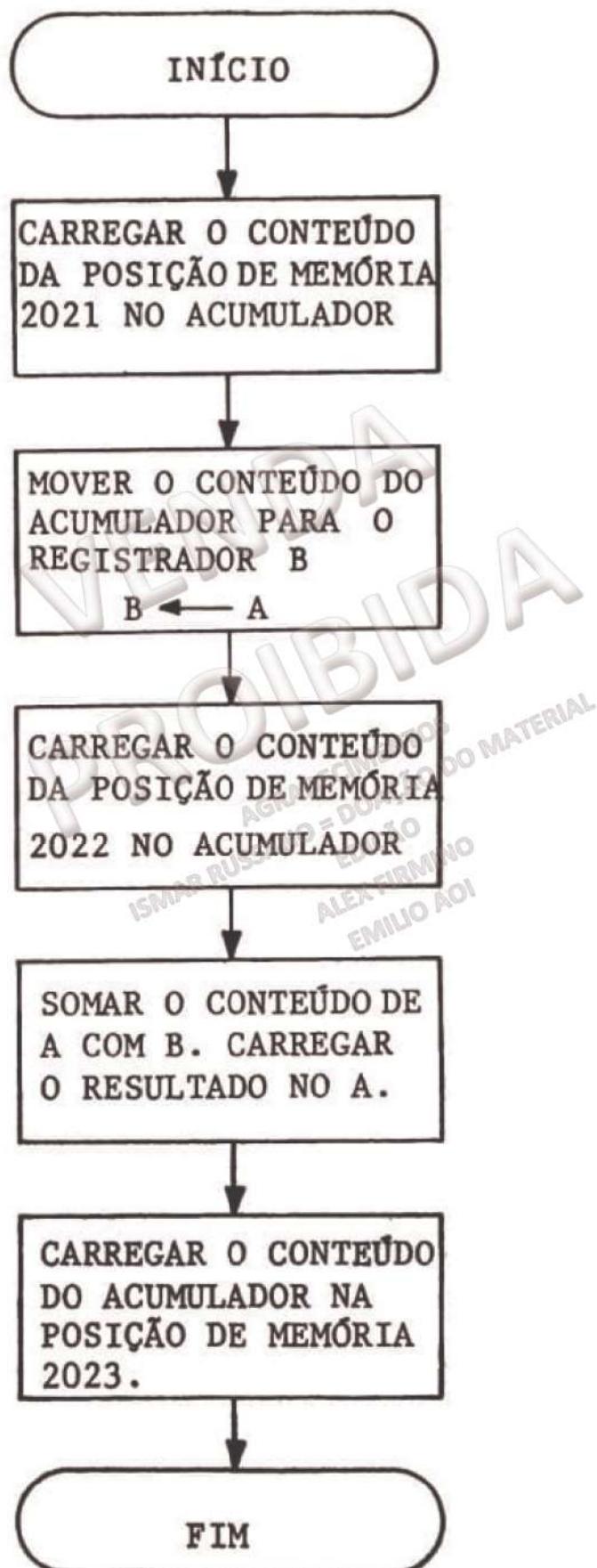


FIGURA 9

A figura 10 nos mostra o programa fonte, escrito em mne-mônicos.

PROGRAMA FONTE

```

LD A, 2021
MOV B, A
LD A, 2022
ADD A, B
LD 2023, A
HLT

```

FIGURA 10

Por outro lado, a figura 11 nos fornece o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2010	3A	Carregar o acumulador com o conteúdo da posição de memória 2021
2011	21 00	
2012	20 02	
2013	47	Mover o conteúdo do acumulador para o registrador B.
2014	3A	Carregar o acumulador com o conteúdo da posição de memória 2022
2015	22 01	
2016	20 02 (02)	
2017	80	Somar o conteúdo de A com B.
2018	32	Carregar a posição de memória 2023 com o conteúdo do acumulador.
2019	23 01	
201A	20 02 (02)	
201B	76 FF	HLT (fim).

FIGURA 11

Para que o programa opere normalmente é preciso carregar os números que se deseja somar nas posições de memórias 2021 e 2022. O resultado da operação estará armazenado na posição de memória 2023.

A título de exemplo, vamos somar os números OF_{16} e $F0_{16}$. Para tanto, vamos armazenar na posição de memória 2021 o númer

ro OF₁₆ e na posição de memória 2022 o número FO. Feito isto é só rodar o programa. O resultado encontrado deverá ser igual a FF, fato este que podemos comprovar, realizando a soma desses dois números, veja:

$$\begin{array}{r} \text{OF} = 0000 \ 1111 \\ \text{FO} = 1111 \ 0000 \\ \hline \underbrace{1111}_{\text{F}} \ \underbrace{1111}_{\text{F}} \end{array} +$$

Vejamos agora um exemplo onde ocorre carry, ou seja, vai um. Vamos somar o número AA com ele mesmo. Para tanto devemos carregar na posição de memória 2021 o valor AA e na posição de memória 2022 o valor AA. Após rodar o programa, iremos encontrar na posição de memória 2023 o valor 54 e no registrador F (Flag) o bit zero que corresponde à posição carry (C) e deverá ser igual a 1 (um). Vejamos:

Hexadecimal	Binário	Decimal
A A A A 15 4	10101010 10101010 101010100	170 170 340

Vai "um"

Como o aluno pode constatar, o número hexadecimal 154 equivale ao decimal 340.

Neste programa cabe ressaltar o fato de que o endereço no qual se encontram os números a serem somados e o endereço onde deveremos encontrar o resultado estão escritos, no programa objeto, de forma inversa. Por exemplo, na linha 2001 temos o número 21 que equivale ao byte menos significativo do endereço 2021, sendo que o byte mais significativo "20" se encontra na linha 2002. Este esquema funciona com base no processo da pilha, onde o "último a entrar é o primeiro a sair". No caso, o último a entrar foi o 20 e este deverá então ser o primeiro a sair. Assim o endereço desejado estará escrito na forma correta "2021".

PROGRAMA 4

Este programa tem por objetivo mostrar um método de se realizar a multiplicação entre dois números binários de 16

bits. Os números a serem multiplicados deverão ser carregados nas posições de memória 2070-2071 e 2072-2073. O resultado da operação de multiplicação será apresentado na posição de memória 2074-2075.

A figura 12 nos mostra o programa fonte, escrito em mnemonicos.

PROGRAMA FONTE

```

LD HL,0000
LD DE,(2070)
LD BC,(2072)
LD A,D
OR E
CALL Z,0038
MLT:SRL B
RRC
JR NC,NCF
ADD HL,DE
CALL C,0038
NCF:LD A,B
OR C
JP Z,ANS ALEX FIRMINO
SLA E
EMILIO AOI
RL D
CALL 0038
JP MLT
ANS:LD(2074),HL
RST
HLT

```

FIGURA 12

Por outro lado, a figura 13 nos mostra o programa objeto pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2010	21	HL = produto inicial igual a zero.
2011	00	
2012	00	
2013	ED	DE = multiplicando

Posição de memória	Código de operação (Hexadecimal)	Comentários
2014	5B	
2015	70 01	
2016	20 02	
2017	ED	BC = multiplicador
2018	4B	
2019	72 02	
201A	20 02	
201B	7A	Verificar se DE = 0
201C	B3	
201D	CC	Caso DE = 0, saltar para o monitor.
201E	38	
201F	00	
2020	CB	BC = deslocamento para a direita, colocando zero
2021	38	
2022	CB	O indicador C é igual ao bit que se multiplica.
2023	19	
2024	30	Comparar o indicador C.
2025	04	
2026	19	Caso indicador C = 1, somar DE ao HL
2027	DC	Caso ADD provoque arraste (vai um).
2028	38	
2029	00	
202A	78	Se o indicador C = 0, verifique se BC = 0.
202B	B1	
202C	CA	Caso BC = 0, guardar o resultado.
202D	39 39	
202E	20 02	
202F	CB	Caso BC ≠ 0, deslocar DE para a esquerda.
2030	23	
2031	CB	
2032	12	
2033	DC	Se C = 1, é porque houve sobrepassamento e é preciso voltar ao monitor.

Posição de memória	Código de operação (Hexadecimal)	Comentários
2034	38	
2035	00	
2036	C3	Caso C ≠ 1, prossiga normalmente.
2037	20	
2038	20	
2039	22	Guarde o resultado.
203A	74	
203B	20	
203C	FF	
203D	76	Pare (fim)

FIGURA 13

Para que o programa opere normalmente, é preciso carregar o multiplicando nas posições de memória 2070 e 2071 e o multiplicador nas posições de memória 2072-2073. O resultado da multiplicação será encontrado nas posições de memória 2074-2075.

Vejamos um exemplo: vamos multiplicar o número 0020 pelo número 0400. Para tanto, vamos carregar o número "20" na posição de memória 2070 e o número "00" na posição de memória 2071. Da mesma forma, vamos carregar o número "00" na posição de memória 2072 e o número "04" na posição 2073. Agora é só rodar o programa.

Para averiguar o resultado vamos verificar o conteúdo das posições de memória 2074-2075. No nosso exemplo teremos, na posição 2074, o número 00 e na posição 2075 o número 80, dando como resultado final o número 8000. Para comprovar o resultado basta resolver a operação em binário, veja:

$$\begin{array}{r}
 00000000 00100000 \\
 \times 0000010000000000 \\
 \hline
 1000000000000000
 \end{array}$$

8 0 0 0

PROGRAMA 5

Este programa tem por objetivo escrever palavras de até

6 dígitos no display do microcomputador CEDM-80. Para tanto, mostramos na figura 14 uma tabela, contendo os códigos dos caracteres mais utilizados.

CÓDIGO DOS CARACTERES

Caracter alfanumérico	Código	Caracter alfanumérico	Código
0	CO	F	8E
1	F9	G	82
2	A4	H	89
3	B0	I	F9
4	99	J	E1
5	92	L	C7
6	82	M	AA
7	F8	N	C8
8	80	O	CO
9	90	P	8C
A	88	Q	98
B	83	R	CE
C	C6	S	92
D	A1	T	87
E	86	U	C1

FIGURA 14

A figura 15 nos apresenta o programa fonte, escrito em mnemônicos.

PROGRAMA FONTE
LD HL, 2030
CALL RESET
JP INICIO

FIGURA 15

Pelo que o aluno pôde notar, o programa em si é relativamente simples, porém é necessário montarmos uma tabela com os códigos dos caracteres que compõem a palavra que pretendemos escrever. Para montagem desta tabela, lançaremos mão da pseudo instrução DFB. Suponhamos que a palavra que pretendemos escrever seja CEDM 80. Para tanto, a tabela dos caracteres a serem mostrados deve ser igual à tabela da figura 16.

Posição de memória	Código de operação (Hexadecimal)	Mnemônico
2030	C6	DFB
2031	86	DFB
2032	A1	DFB
2033	AA	DFB
2034	80	DFB
2035	CO	DFB

FIGURA 16

A figura 17 nos mostra o programa objeto, pronto para ser digitado.

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	21	Carregar o par de registradores HL com o endereço do ínicio da tabela de caracteres a serem mostrados.
2001	30	
2002	20	
2003	CD	Chama rotina de restauração, mandando para o display os 6 caracteres desejados.
2004	9C	
2005	00	
2006	C3	Saltar para o inicio do programa.
2007	00	
2008	20	

FIGURA 17

OBS.:

O endereço da rotina de restauração "009C" (veja linhas 2004 e 2005) faz parte do programa monitor do microcomputador CEDM-80.

PROGRAMA 6

Baseado no programa anterior, este programa tem por objetivo fazer com que uma frase circule no display, da direita para a esquerda, por um tempo indeterminado. A figura 18 nos mostra o programa fonte, escrito em mnemônicos.

PROGRAMA FONTE

LD BC, BMD - (buffer de mensagem do display = BMD).
 LD D, tempo - (fixar tempo - velocidade de avanço).
 INC BC
 LD A, C
 CP 36
 JP Z, inicio - (volta ao início do programa).
 LD H, B
 LD L, C
 PUSH BC
 CALL REST
 POP BC
 DEC D
 JP NZ, L1
 JP L2

FIGURA 18

Vamos supor que desejamos fazer circular pelo display a frase "EU SOU CEDM 80". Para tanto, devemos escrever uma tabela, contendo os caracteres que deverão ser mostrados, a qual, no nosso caso, inicia-se a partir da posição de memória 2020. A figura 19 nos mostra a tabela de caracteres.

TABELA DE CARACTERES

Posição de memória	Código de operação (Hexadecimal)	Mnemônico	Comentários
2020	FF	DFB	
2021	FF	DFB	
2022	FF	DFB	
2023	FF	DFB	apagado
2024	FF	DFB	
2025	FF	DFB	
2026	86	DFB	E
2027	C1	DFB	U
2028	FF	DFB	
2029	FF	DFB	apagado
202A	92	DFB	S
202B	C0	DFB	O
202C	C1	DFB	U
202D	FF	DFB	
202E	FF	DFB	apagado
202F	C8	DFB	C
2030	86	DFB	E
2031	92	DFB	D
2032	87	DFB	M

FIGURA 19

Posição de memória	Código de operação (Hexadecimal)	Mnemônico	Comentários
2033	CO	DFB	8
2034	CE	DFB	0
2035	FF	DFB	
2036	FF	DFB	
2037	FF	DFB	
2038	FF	DFB	apagado
2039	FF	DFB	
203A	FF	DFB	

O código de operação "FF" corresponde ao RESET. Assim as seis primeiras linhas da tabela nos garantem que o visor comece apagado.

A figura 20 nos mostra o programa objeto, pronto para ser digitado.

FIGURA 20

PROGRAMA OBJETO

Posição de memória	Código de operação (Hexadecimal)	Comentários
2000	01	Endereço inicial do buffer de mensagem do display.
2001	20	
2002	20	
2003	16	Fixar tempo (velocidade de avanço do visor).

Posição de memória	Código de operação (Hexadecimal)	Comentários
2004	20	
2005	03	Avança o buffer de mensagem.
2006	79	Carregar o acumulador com o final do buffer.
2007	FE	Comparar com o final da mensagem.
2008	36	
2009	CA	Reciclar, se chegou no inicio
200A	00	
200B	20	
200C	60	Carregar o par HL como indicador do buffer para utilizar a rotina de restauração.
200D	69	
200E	C5	Salvar o par BC.
200F	CD	Chamar rotina para o display.
2010	9C	
2011	00	
2012	C1	Restaurar BC.
2013	15	Decrementar constante de tempo
2014	C2	Não sendo zero, continuar a mensagem.

Posição de memória	Código de operação (Hexadecimal)	Comentários
2015	0C	
2016	20	
2017	C3	Sendo zero, adiante um dígito
2018	03	
2019	20	

Para alterar o tamanho da mensagem basta alterar a posição de memória 2008, de acordo com o comprimento da nova mensagem. Também nos é possível alterar a velocidade, bastando, para tanto, alterar a posição de memória 2004.

COMENTÁRIOS FINAIS

Com esta lição encerramos o nosso curso, uma vez que a lição MP-26 se refere apenas aos aspectos de mentagem do Kit CEDM-80.

Acreditamos termos atingido nossos objetivos com relação ao curso em si, principalmente no que diz respeito à divulgação tecnológica, bem como a formação e/ou especialização de técnicos na área de eletrônica digital e microprocessadores.

Finalizando, queremos deixar patente a nossa alegria e satisfação em podermos chegar até o final do curso com vocês. Até uma próxima oportunidade.



CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 CEP 86100 LONDRINA PARANÁ

EXAME DA LIÇÃO MP-25

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

As questões 1, 2 e 3 se referem ao programa número 1 (veja figura 4).

1) Ao se carregar o número hexadecimal 63 no registrador C, obteremos como resultado o seguinte número em BCD:

- a) 31;
- b) 91;
- c) 74;
- d) 99.

2) Idem à anterior para o número hexadecimal A:

- a) 20;
- b) 10;
- c) 93;
- d) N.R.A.

3) Idem à anterior para o número hexadecimal 3A:

- a) 58;
- b) 48;
- c) 5B;
- d) 99.

As questões 4, 5 e 6 se referem ao programa número 2 (veja figura 8).

4) Ao se carregar o número BCD 39 no registrador C, obteremos como resultado o seguinte número hexadecimal:

- a) 3F;
- b) 27;
- c) 2C;
- d) 5A.

5) Idem à questão 4 para o número BCD 55:

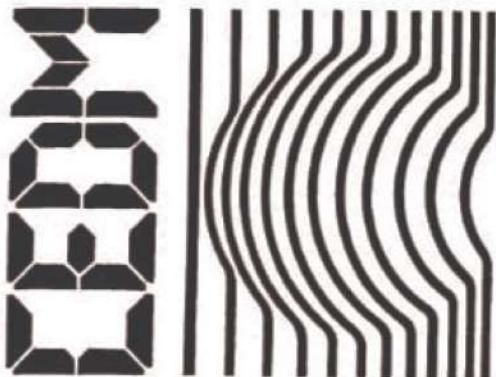
- a) 27;
- b) 2F;
- c) 37;
- d) EA.

6) Idem à anterior para o número BCD 29:

- a) A0;
- b) 3F;
- c) 59;
- d) 1D.

7) Com base no programa número 3, a soma entre OC e BO tem como resultado:

- a) BC;
b) OA;
c) BF;
d) 01.
- 8) Ainda com base no programa número 3, a soma entre 8A e 10 tem como resultado:
a) 1A;
 b) 9A;
c) 3B;
d) AO.
- 9) Com base no programa número 4, o resultado da operação de multiplicação entre os números 00FF e 00FF, tem como resultado:
a) 9010;
b) OAFE;
c) FE01;
 d) N.R.A.
- 10) Com relação ao programa número 6, se alterarmos o valor da posição de memória 2004 de 20 para 30, a velocidade de avanço do visor:
a) não se altera;
 b) diminui;
c) aumenta;
d) N.R.A.



CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

LICÃO MP-26

MANUAL DE MONTAGEM DO KIT CEDM-80

INTRODUÇÃO

O maior anseio de qualquer profissional, ou até mesmo de qualquer principiante, que se adentra à informática, é montar algo que o aproxime mais da área, que, de preferência, seja o seu próprio micro. Chegamos então, à mais importante realização prática do curso: a montagem do Kit CEDM-80, montagem esta que será, sem dúvida, o alvo das congratulações do árduo trabalho realizado por parte do aluno, do qual nos orgulhamos.

A princípio, a montagem possa parecer um tanto exaustiva. Porém, dada à importância e à destinação do Kit, partimos de uma estaca zero, fornecendo todas as informações necessárias para que se assegure do bom desempenho, conseguido com ordem e cautela. Para isto, criamos uma seqüência, a qual deverá ser seguida fielmente. Assinale a execução realizada de cada item, a fim de ter o ponto em que parou quando na interrupção da montagem.

CUIDADOS COM OS CIRCUITOS INTEGRADOS

Sabemos que os dispositivos semi-condutores são, de forma geral, muito sensíveis, cuja sensibilidade aumenta com o

VERIDA
PROIBIDA

AGRADECIMENTOS
MAR RUSIANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMÍNO
EMMA AOI

nível de integração de um circuito integrado.

Como mencionamos na lição E-1, estamos submetidos aos efeitos da eletrostática, a qual se caracteriza, mais que outros pela ocorrência de pontos com acúmulos de cargas. É, justamente, por este motivo, que os circuitos integrados (CIs) são normalmente encontrados sobre superfícies que mantêm todos os pinos sobre o mesmo potencial, evitando assim, o efeito da eletrostática. Desta maneira, os CIs deverão ser mantidos dentro do invólucro fornecido, pois este atua como protetor para os mesmos, por ser anti-estático. Assim, retire os CIs deste invólucro, somente no instante da montagem do mesmo, segurando-o como mostra a figura 01.

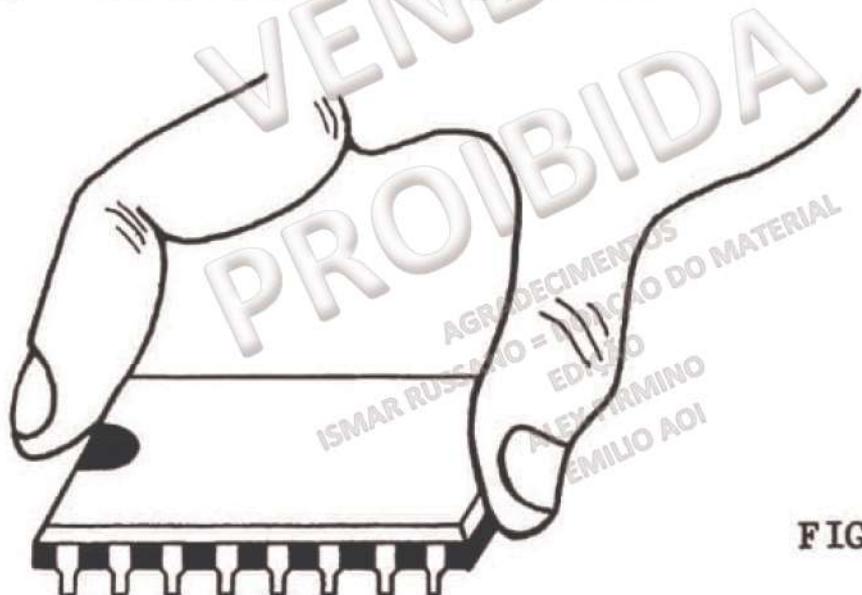


FIGURA 01

RELAÇÃO DO MATERIAL

O aluno deverá estar recebendo os seguintes componentes:

Circuitos integrados:

- () 1 - microprocessador - CPU-Z80A (CI-1)
- () 1 - memória EPROM 2716 (CI-2)
- () 1 - memória RAM 6116 (CI-3)
- () 1 - inversor tipo quadrator - 74LS14 (CI-4)
- () 1 - decodificador - 74LS138 (CI-5)
- () 1 - porta E (AND) quádrupla - 74LS08 (CI-6)
- () 1 - porta NÃO OU (NOR) quádrupla - 74LS02 (CI-7)
- () 2 - trancas bi-estáveis (LATCH) - 74LS75 (CI-8 e CI-9)

- () 2 - Bus-Buffer, TRI-STATES - 74LS126 (CI-10 e CI-11)
- () 1 - flip-flop tipo D - 74LS174 (CI-12)

Diodos:

- () 1 - diodo de comutação - 1N-4148

Transistores:

- () 6 - tipo npn - BC-338 (TR-1 a TR-6)
- () 8 - tipo pnp - BC-308 (TR-7 a TR-14)

Resistores:

- () 11 - resistores 1K Ω /0,25W (R₁; R₂; R₅; R₂₈ a R₃₅)
- () 8 - resistores 3K3 Ω /0,25W (R₆ a R₁₃)
- () 8 - resistores 10 Ω /0,25W (R₂₀ a R₂₇)
- () 7 - resistores 4K7 Ω /0,25W (R₃; R₁₄ a R₁₉)
- () 1 - resistores 330 Ω /0,25W (R₄)

Capacitores:

- () 1 - capacitor cerâmico 100pF (C₁)
- () 1 - capacitor eletrolítico 10 μ F (C₂)
- () 1 - capacitor poliéster metalizado 0,1 μ F (C₃)

Teclado:

- () 1 máscara de teclado
- () 36 teclas
- () 3 placas de borracha (chupetas)
- () 1 película isolante
- () 1 película condutora

Diversos:

- () 1 - soquete de 40 pinos
- () 2 - soquete de 24 pinos
- () 1 - placa de circuito impresso - dupla face
- () 4 - pés de borracha
- () 4 - parafusos para fixação do teclado
- () 2 m de fio nu, bitola 22 AWG

- () 60 cm de rabicho (fio encapado bitola 22 AWG)
- () 2 m de solda

Todo o material enviado foi conferido e todos os componentes devidamente testados. Porém, se alguma irregularidade for constatada, comunique-a imediatamente, não dando início à montagem, para que não fique um espaço na seqüência fornecida.

PLACA DE CIRCUITO IMPRESSO

Como suporte para o Kit CEDM-80, optamos por um material que sem dúvida, fornece a base ideal para a fixação dos componentes, cuja opção está fundamentada na resistência, durabilidade e simplicidade do circuito, onde a margem de erros é extremamente reduzida.

Uma complexidade da placa é facilmente constatada, ao termos como base o número de componentes a serem fixados, onde se faz necessária uma elevada quantia de filetes. Visando a uma compactação da placa, foi utilizada uma placa dupla face, sendo ela de fenolite com dimensões de 18 x 17,5 cm.

Antes de realizar qualquer conexão, é imprescindível um reconhecimento da placa. Já que foi fornecido o esquema do circuito do Kit, com a ajuda deste acompanhe cada filete, checando a interligação por ele efetuada. Estenda este trabalho para todos os pontos a serem conectados, para que se tenha a visão e o poder de discussão sobre a experiência que será executada.

Na figura 02, fornecemos as ilustrações de cada placa, bem como da máscara (disposição dos componentes). Observe que a placa recebida apresenta a sobreposição da máscara na face superior, caracterizando o lado pelo qual serão inseridos os componentes.

FIGURA 02

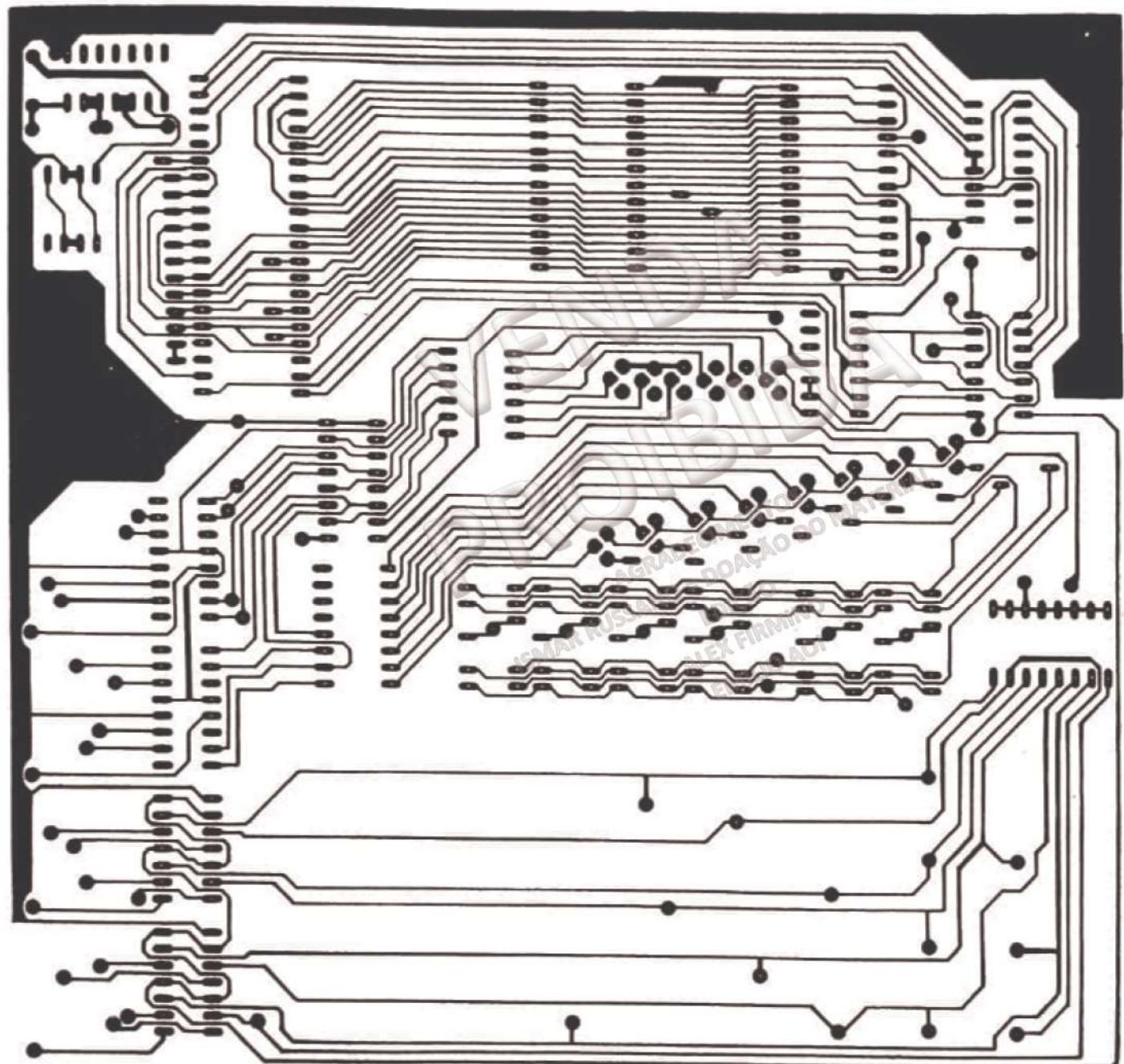


FIGURA 02-A

(FACE INFERIOR)

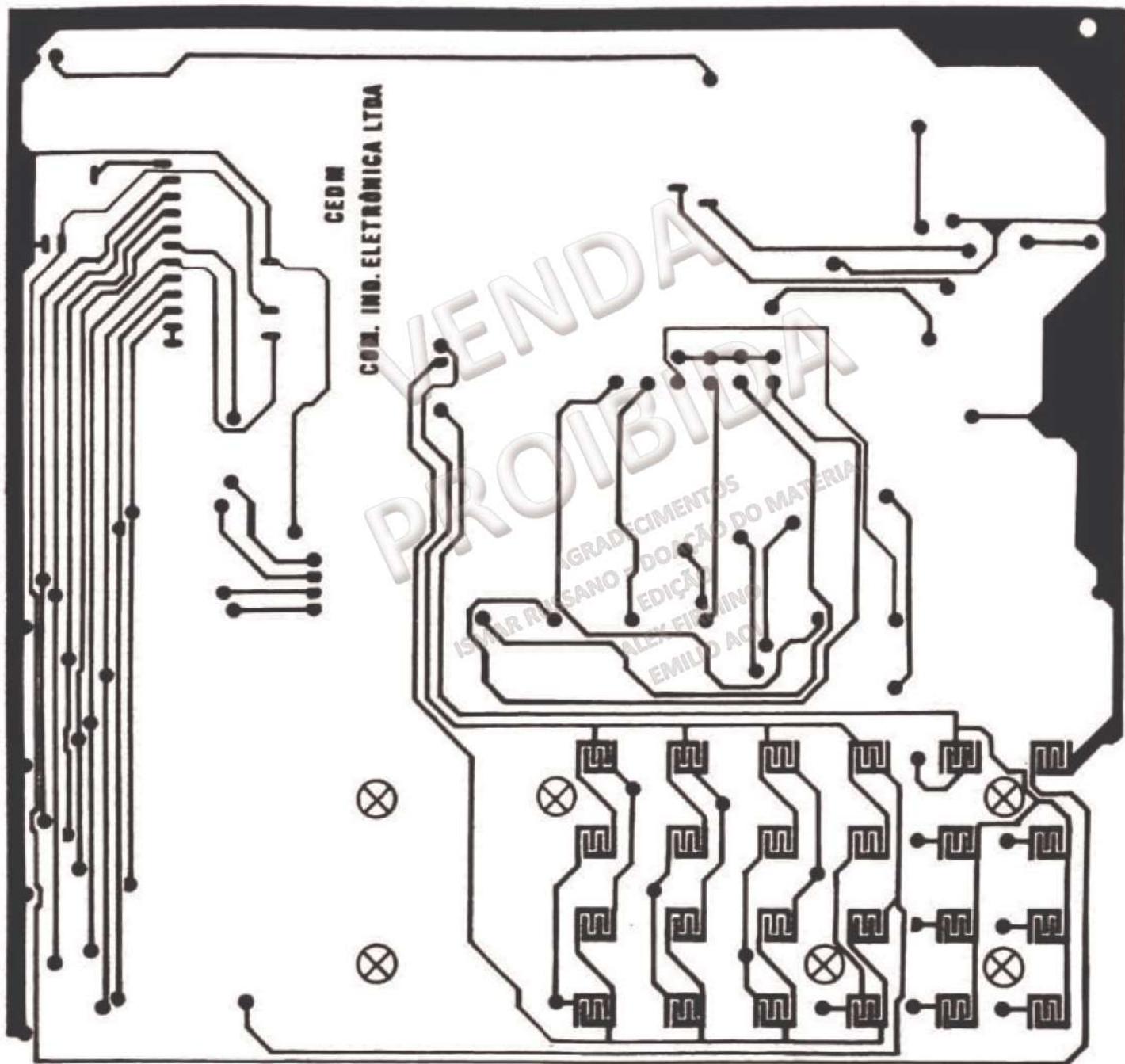


FIGURA 02-B

(FACE SUPERIOR)

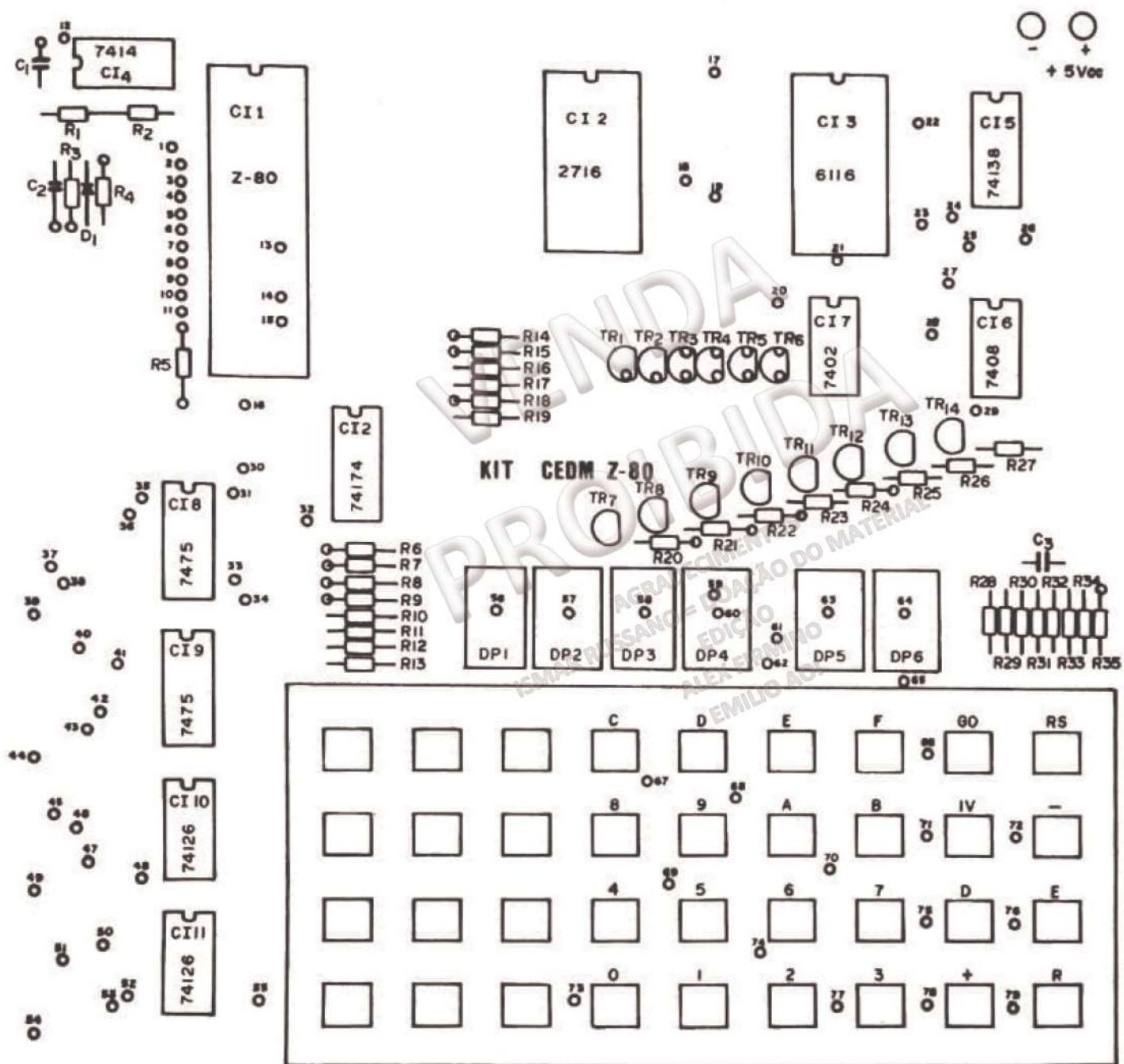


FIGURA 02 - C
(MÁSCARA)

A SOLDAGEM

O sucesso da montagem depende quase que na essência da soldagem realizada por se tratar com elementos muito sensíveis à temperatura. Se, porventura, um calor excessivo foi aplicado à placa, poderá ocorrer o deslocamento do filete ou, quando aplicado sobre os componentes, poderá danificá-los totalmente. Portanto, todo cuidado se faz necessário.

Para haver uma boa transferência de calor, o ponto primordial a ser observado é o aspecto apresentado pelo soldador, pois este não deverá apresentar uma ponta rugosa (figura 03-A), proveniente do acúmulo de sujeira, mas sim, uma ponta lisa (figura 03-B). A forma mostrada na figura 03-B é conseguida limpando a ponta do soldador, deixando obrigatoriamente um chanfro com inclinação de aproximadamente 45° .

FIGURA 03

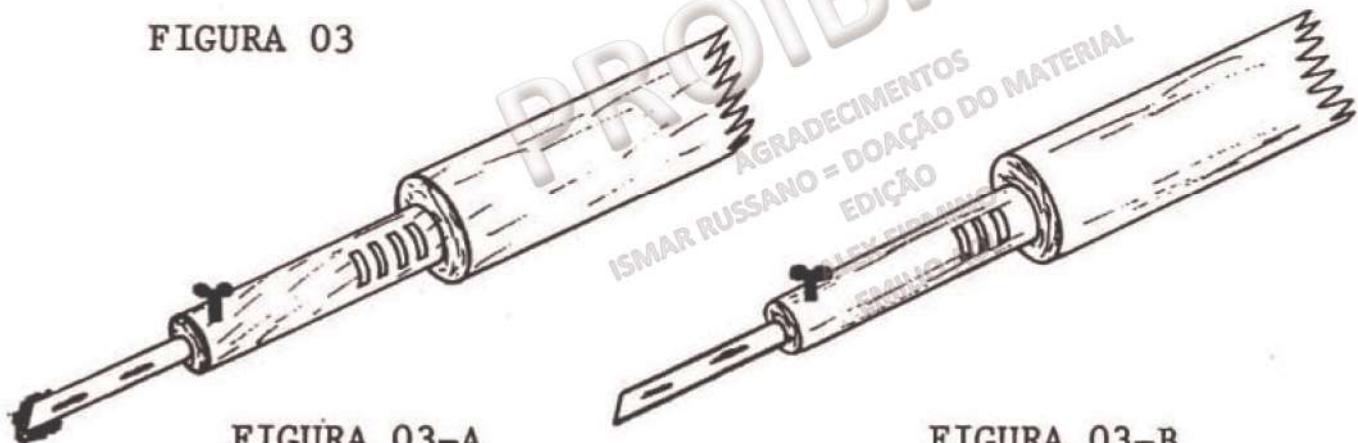


FIGURA 03-A

FIGURA 03-B

A posição do ferro de soldar também é fundamental, pois deverá tocar simultaneamente a ilha e o terminal, estando posicionado no lado oposto ao ocupado pelo fio solda, como é mostrado na figura 04.

TERMINAL

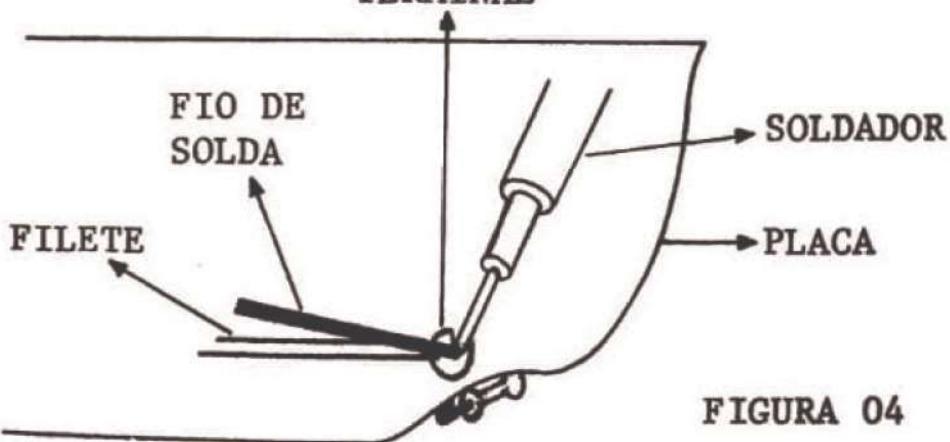


FIGURA 04

Partindo das posições indicadas na figura 04, para se con seguir uma solda perfeita, primeiro deverá ser fornecido um aquecimento prévio tanto ao terminal quanto à ilha, para então aproximar a solda, a qual se fundirá e envolverá o terminal. Estando a ilha coberta pela solda, primeiro retira-se o fio de estanho e logo após, retira-se o soldador. Este processo é realizado quase que simultaneamente e o formato final apresentado pela solda deverá ser liso e brilhante (FIGURA 05), não esparramando para além da ilha, de maneira que ocasione a formação de pontes, ou seja, fazendo contato com outras partes do circuito. Ocorrendo a formação de pontes, aconselha-se retirar o excesso de estanho com um pedaço de madeira, nunca utilizando metal ou plástico.

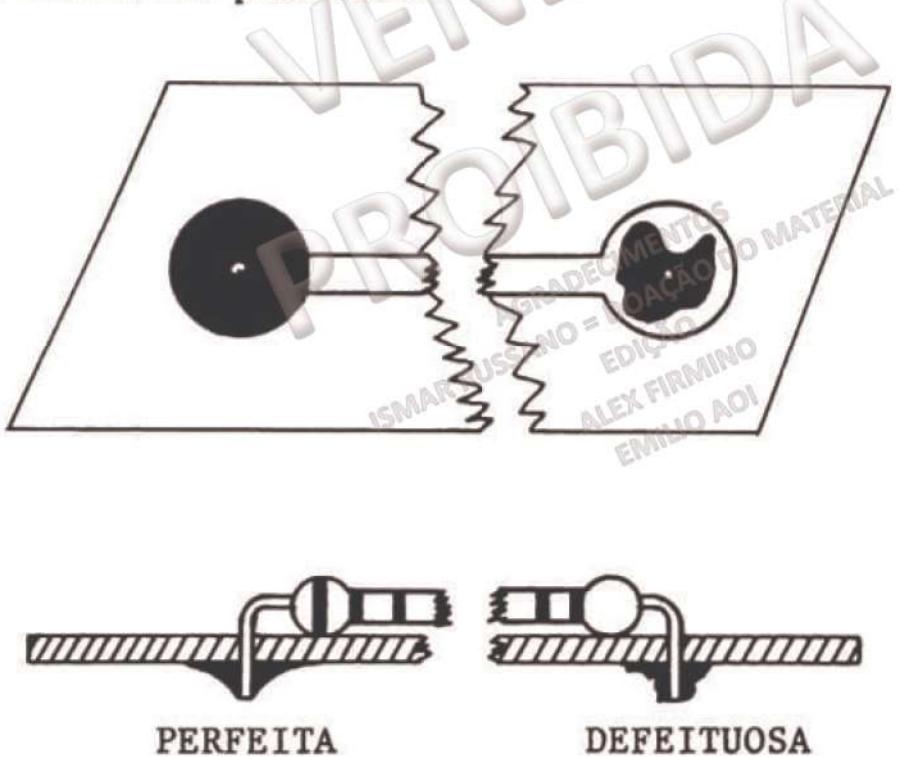


FIGURA 05

Aqui comprovamos o que fora mencionado acerca da complexidade envolvida pela placa, pois está diretamente ligada à soldagem a ser realizada. Dada a proximidade dos pontos e dos filetes, a quantidade de estanho a ser aplicado deverá ser apenas o suficiente para garantir o contato e a fixação dos componentes. Desta maneira, estando inseguro, não hesite em praticar fora da placa. Porém, acreditamos que, com estas instruções preliminares, podemos adentrar ao trabalho propriamente dito.

A MONTAGEM

1 — JUMPERS

Já que a placa de circuito impresso utilizada é de dupla face, será necessário unir filetes (em alguns pontos) situados em faces opostas. Para esta união, denominada jumper, utilizaremos pedaços de fio nu, de bitola 22 AWG, os quais serão inseridos nos pontos identificados por um circuito numerado, conforme mostra a máscara impressa sobre a face superior da placa. Observe que certos componentes apresentam esta identificação em torno de seu(s) terminal(ais), apontando também jumpers a serem efetuados, só que utilizaremos o próprio terminal, como será visto mais adiante, na fixação de tal componente.

A soldagem a ser efetuada sobre os jumpers deverá ser a mais baixa possível, visto alguns estarem situados sob alguns componentes e até mesmo sob o teclado. Desta maneira, estes jumpers não deverão, em hipótese alguma, exercer influencia sobre o apoio de tais dispositivos.

A fim de obter um ótimo rendimento, dividiremos os jumpers em grupo de 10 unidades, identificando-os na placa para a devida fixação, onde será fixado um de cada vez.

Vejamos uma maneira de realizar esta operação, conforme instruções dadas a seguir:

1 - Dobre (com o alicate de bico) os jumpers e corte (com o alicate de corte) onde se fizer necessário, para que se obtenha a seguinte forma:

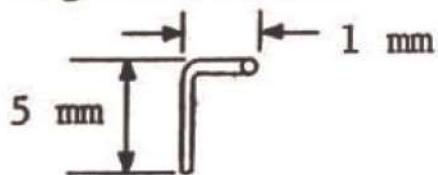


FIGURA 6

- 2 - Introduza-o no lugar correspondente da placa;
- 3 - Firme a parte situada na face superior e dobre a parte excedente da face inferior, deixando-a rente à placa (utilizar as mãos);
- 4 - Apóie a placa e, com a ponta do alicate de bico, faça pressão sobre a parte do jumper situado na face superior;

5 - Efetue a solda na face superior;

5.a - O ferro de soldar deverá tocar simultaneamente o fio jumpeador e a placa, enquanto que o fio de solda de verá tocar o fio jumpeador, para que o estanho escorra sobre ele;

6 - Vire a placa e repita o item 4;

7 - Corte a parte do fio que exceder a ilha e aplique a solda sobre o mesmo, na face inferior;

7.a - Repetir o sub-item 5.a.

A figura 07 ilustra as várias etapas:

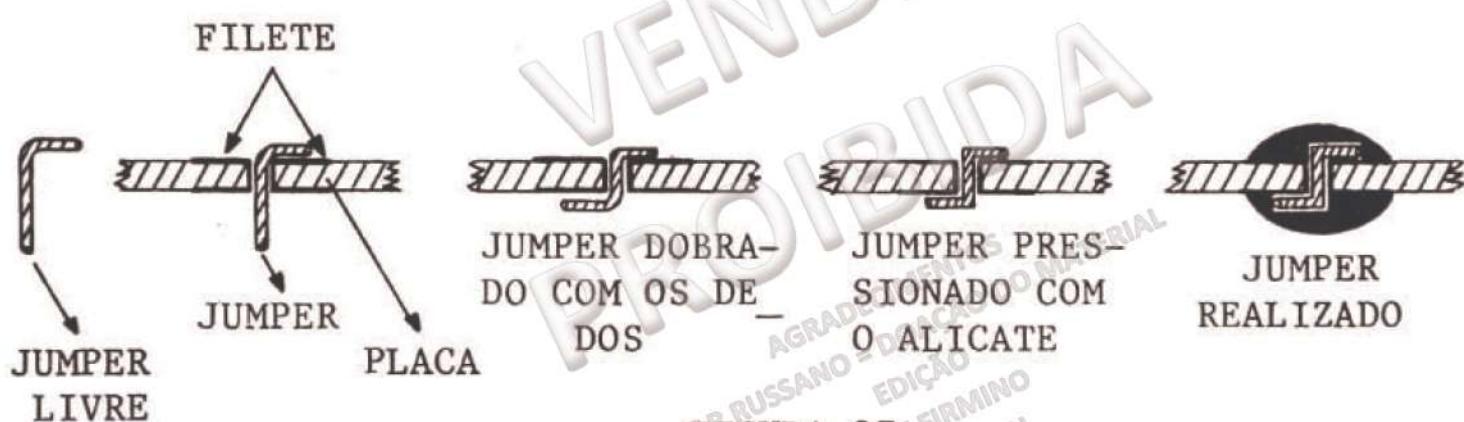


FIGURA 07

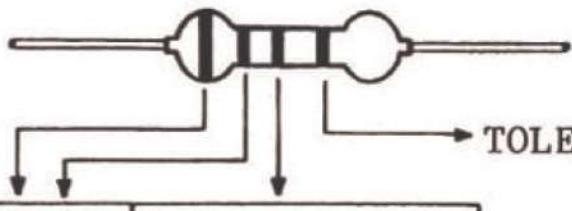
Assim:

- () solde os jumpers de 1 a 10.
- () solde os jumpers de 11 a 20.
- () solde os jumpers de 21 a 30.
- () solde os jumpers de 31 a 40.
- () solde os jumpers de 41 a 50.
- () solde os jumpers de 51 a 60.
- () solde os jumpers de 61 a 70.
- () solde os jumpers de 71 a 79.

Logo após a fixação de cada lote, pedimos checar as soldagens efetuadas.

2 - RESISTORES

De início, recorde o código de cores utilizado para a caracterização de resistências:



TOLERÂNCIA { Ouro = $\pm 5\%$
Prata = $\pm 10\%$
Incolor = $\pm 20\%$

COR	VALOR	MULTIPLICADOR
Preto	0	$1 = 10^0$
Marrom	1	$10 = 10^1$
Vermelho	2	$100 = 10^2$
Laranja	3	$1000 = 10^3$
Amarelo	4	$10000 = 10^4$
Verde	5	$100000 = 10^5$
Azul	6	$1000000 = 10^6$
Violeta	7	não usado
Cinza	8	não usado
Branco	9	não usado
Ouro	-	$\times 0,1$
Prata	-	$\times 0,01$

FIGURA 08

Após identificar o resistor a ser fixado, dobre-o antes de ser inserido na devida posição da placa, de maneira que apresente em torno de 2 mm de cada lado do corpo do resistor:

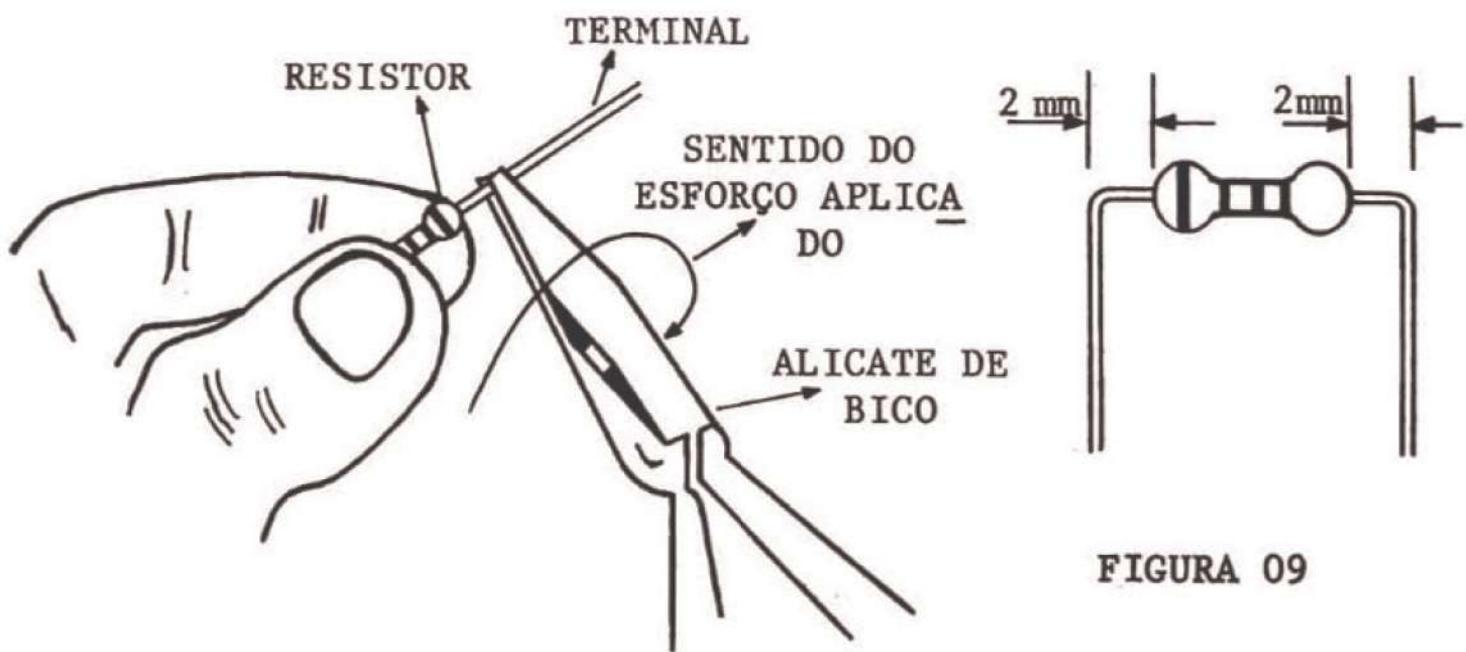


FIGURA 09

Estando com o componente dobrado, introduza-o na devida posição, de maneira que o corpo do mesmo fique apoiado sobre a placa. A parte do terminal que exceder a 2 mm da placa deve ser cortada para então se efetuar a solda.

Já que o trabalho deve ser feito com muito cuidado, aconselhamos dobrar os 2 mm do terminal sobre a placa antes de soldar, para que a soldagem apresente uma estética muito melhor.

Na parte que versava sobre os jumpers, já fora mencionado que alguns terminais seriam utilizados para efetuar conexões entre os filetes de faces opostas, os quais são identificados por um círculo. Quando ocorrerem estes jumpers, proceda normalmente, soldando primeiro na face inferior. Para realizar o jumper, vire a placa e aplique a solda junto ao terminal assinalado.



FIGURA 10

Acompanhe fielmente a seqüência a seguir, assinalando logo após a fixação do correspondente resistor:

- () resistor R₁ (1KΩ) [marrom, preto, vermelho]
 - () resistor R₂ (1KΩ) [marrom, preto, vermelho]
 - () resistor R₃ (4K7Ω) [amarelo, violeta, vermelho]
 - () jumper sobre o terminal de R₃
 - () resistor R₄ (330Ω) [laranja, laranja, marrom]
 - () jumper sobre o terminal de R₄
 - () resistor R₅ (1KΩ) [marrom, preto, vermelho]
 - () 19 jumper sobre o terminal de R₅
 - () 29 jumper sobre o terminal de R₅
 - () resistor R₆ (3K3Ω) [laranja, laranja, vermelho]
 - () jumper sobre o terminal de R₆
 - () resistor R₇ (3K3Ω) [laranja, laranja, vermelho]
 - () jumper sobre o terminal de R₇
 - () resistor R₈ (3K3Ω) [laranja, laranja, vermelho]
 - () jumper sobre o terminal de R₈

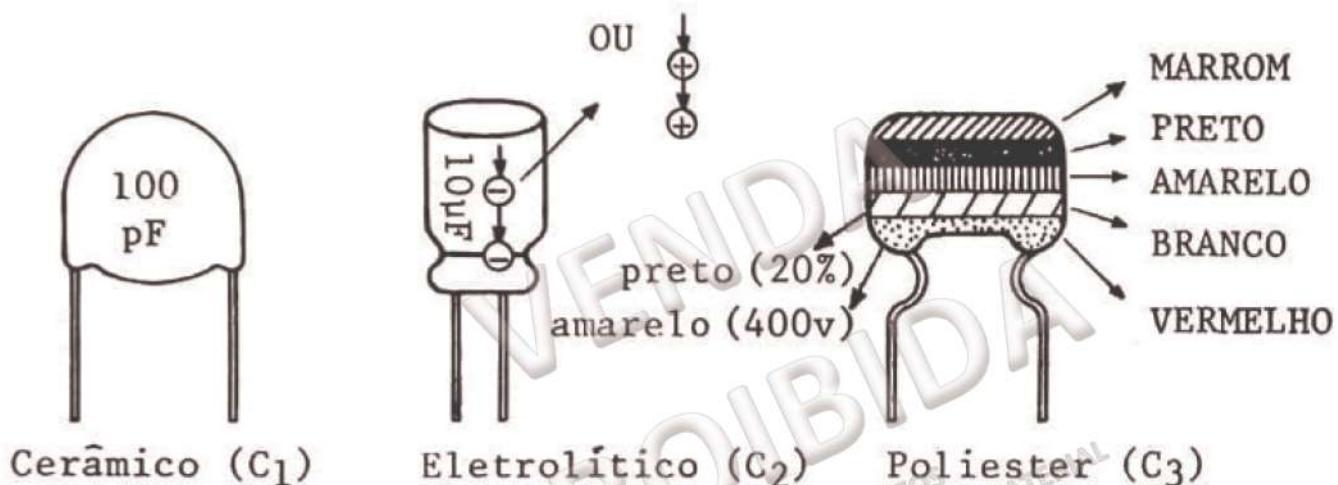
- () resistor R₉ (3K3Ω) [laranja, laranja, vermelho]
() jumper sobre o terminal de R₉
- () resistor R₁₀ (3K3Ω) [laranja, laranja, vermelho]
() resistor R₁₁ (3K3Ω) [laranja, laranja, vermelho]
() resistor R₁₂ (3K3Ω) [laranja, laranja, vermelho]
() resistor R₁₃ (3K3Ω) [laranja, laranja, vermelho]
() resistor R₁₄ (4K7Ω) [amarelo, violeta, vermelho]
() jumper sobre o terminal de R₁₄
- () resistor R₁₅ (4K7Ω) [amarelo, violeta, vermelho]
() jumper sobre o terminal de R₁₅
- () resistor R₁₆ (4K7Ω) [amarelo, violeta, vermelho]
() resistor R₁₇ (4K7Ω) [amarelo, violeta, vermelho]
() resistor R₁₈ (4K7Ω) [amarelo, violeta, vermelho]
() jumper sobre o terminal de R₁₈
- () resistor R₁₉ (4K7Ω) [amarelo, violeta, vermelho]
() resistor R₂₀ (10Ω) [marrom, preto, preto]
() jumper sobre o terminal de R₂₀
- () resistor R₂₁ (10Ω) [marrom, preto, preto]
() jumper sobre o terminal de R₂₁
- () resistor R₂₂ (10Ω) [marrom, preto, preto]
() jumper sobre o terminal de R₂₂
- () resistor R₂₃ (10Ω) [marrom, preto, preto]
() resistor R₂₄ (10Ω) [marrom, preto, preto]
() jumper sobre o terminal de R₂₄
- () resistor R₂₅ (10Ω) [marrom, preto, preto]
() resistor R₂₆ (10Ω) [marrom, preto, preto]
() resistor R₂₇ (10Ω) [marrom, preto, preto]
- () resistor R₂₈ (1KΩ) [marrom, preto, vermelho]
() resistor R₂₉ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₀ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₁ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₂ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₃ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₄ (1KΩ) [marrom, preto, vermelho]
() resistor R₃₅ (1KΩ) [marrom, preto, vermelho]
() jumper sobre o terminal de R₃₅

Atenção: Analise todas as soldagens efetuadas e certifique-se de que não ficou nenhuma por executar, antes de passar ao próximo item.

3 – CAPACITORES

Dentre os capacitores recebidos, conforme ilustra a figura 11, um deles requer cuidados especiais para a fixação, devido à polaridade de que é dotado, a qual deverá ser antes identificada para então fixar o componente.

FIGURA 11



A saber, no corpo do capacitor eletrolítico há uma seta que aponta para um dos terminais, que poderá ser o positivo ou o negativo. Esta indicação depende unicamente do fabricante, sem acarretar nenhum problema, pois indicando um, o outro terminal é logo identificado. Na placa, o terminal negativo deve ser introduzido no ponto onde será feito o jumper.

A identificação dos capacitores C_1 e C_2 é imediata, pois ambos apresentam os valores marcados no corpo do dispositivo. Já o capacitor C_3 é todo colorido, sendo que a sua identificação segue o mesmo padrão do código de cores para resistências. Isto para as 3 faixas superiores, cujo valor encontrado deverá preceder à potência 10^{-12} (pico), que é utilizado para maioria dos capacitores. As faixas inferiores poderão diferir da referenciada na ilustração correspondente, pois fornecem a tolerância e a tensão de isolação. Para o capacitor C_3 temos:

$$C_3 = \text{marrom} \times \text{preto} \times 10^{\text{amarelo}} \times 10^{-12} \text{ (Farad)}$$

Aqui também alguns terminais foram utilizados para a realização de alguns jumpers, os quais são ainda caracterizados por um círculo envolvendo o terminal.

Para a fixação, use a mesma técnica utilizada para os resistores, cortando a parte que excede 2 mm da face inferior,

soldando logo em seguida, para então soldar na face superior, se houver jumper. A título de informação abriremos uma exceção quanto à dobra do terminal do capacitor C_3 , devido ao diâmetro do mesmo.

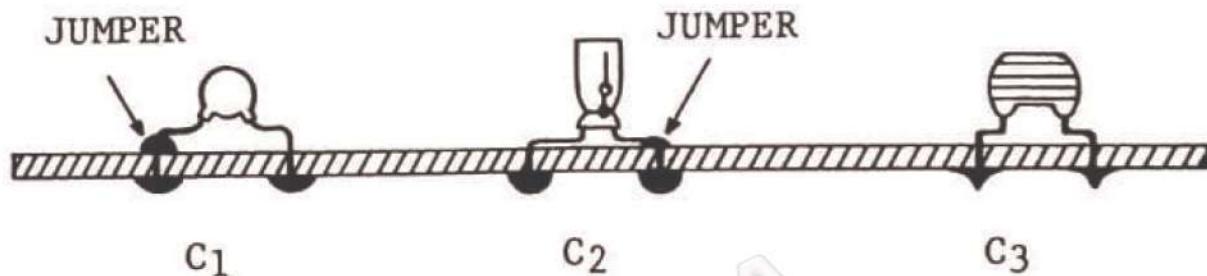


FIGURA 12

Assinale, logo após a fixação do componente, de acordo com a seguinte seqüência:

- () capacitor C_1 (cerâmica)
- () jumper do capacitor C_1
- () capacitor C_2 (eletrolítico)
- () jumper no terminal negativo de C_2
- () capacitor C_3 (poliéster)

*ISMAEL GUSSANO = GRAVAMENTOS
ALEX TAVINO = EDIÇÃO
EMÍLIO AUF = MATERIAL PROIBIDA*

Antes de prosseguir, analise o trabalho realizado!

4 — DIODO

Primeiramente devemos identificar os terminais do diodo, pois este deverá ser fixado de acordo com a simbologia gravada sobre a placa. A figura 13 fornece a correlação entre os terminais do diodo real com a convenção para o mesmo:

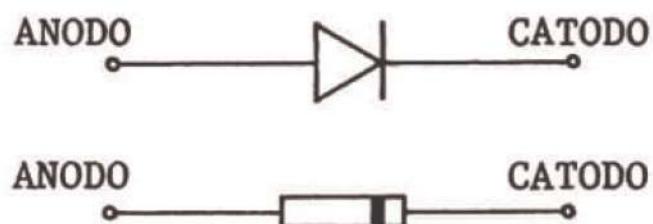


FIGURA 13

Os terminais do diodo deverão ser dobrados de maneira que apresentem 3mm de cada lado do corpo do componente, para então ser inserido pela face superior, respeitando a correlação mostrada na figura 13, deixando-o apoiado sobre a placa,

utilizando a mesma técnica de soldagem anteriormente descrita.

Sendo o diodo muito sensível à temperatura, deverá ser tomado o cuidado para não aquecer demasiadamente o terminal para não danificá-lo.

Após fixado, teremos:

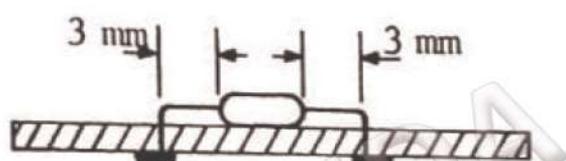
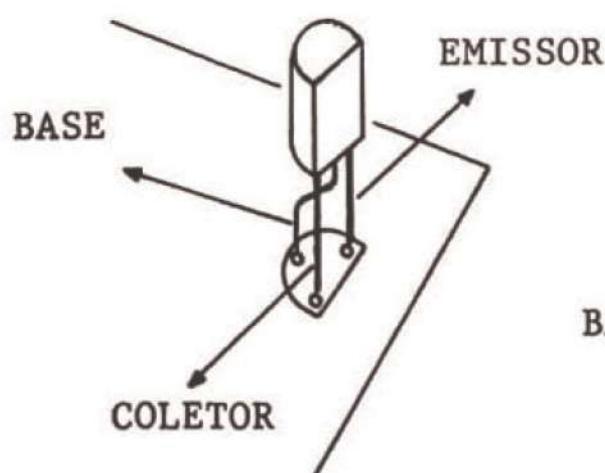


FIGURA 14

Temos apenas um diodo. Mesmo assim, certifique-se da boa soldagem realizada.

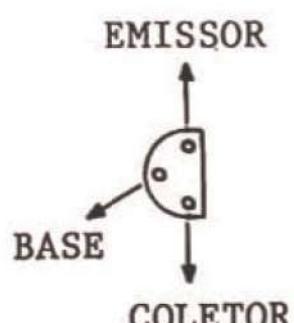
5 — TRANSISTORES

A característica fundamental para o funcionamento correto de qualquer dispositivo semicondutor está assentada na alimentação adequada de seus terminais, sendo para isso necessária a identificação dos mesmos. Os dois tipos apresentados — (BC-308 e BC-237) apresentam a mesma disposição de seus terminais, conforme ilustrado na figura 15.



PLACA E COMPONENTE

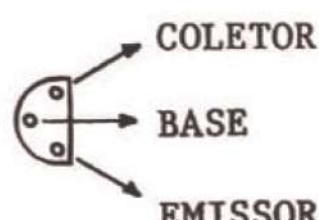
(C)



PLACA
(Vista superior)

(A)

FIGURA 15



COMPONENTE
(Vista inferior)

(B)

Para alojar o transistor na placa, não é necessário nenhuma adequação aos seus terminais, pois a placa foi confecionada de maneira que o componente encaixe diretamente, basta apenas alinhar o formato do componente com a projeção ilustrada na placa. Observe que a posição é a mesma para todos os transistores, devendo apenas tomar o cuidado para não trocar os BC-237 pelos BC-308.

A estética é, sem dúvida, um fator importantíssimo na montagem. Para isto, procure alinhar todos os transistores, deixando-os na mesma altura, que deverá girar em torno de 1 cm, medido da placa até o topo do componente. A parte que exceder 2 mm da face inferior deverá ser cortada antes de ser soldada. Não dobre o terminal sobre a face inferior para não tirar o transistor do alinhamento, o qual é conseguido tomando como base o primeiro fixado.

Como se procede com o diodo, cuidado para não aplicar um calor excessivo aos terminais dos transistores, tanto na soldagem da face inferior como dos jumpers a serem realizados.

- () Transistor TR₁ (BC-237)
 - () jumper no terminal de coletor de TR₁
- () Transistor TR₂ (BC-237)
 - () jumper no terminal de coletor de TR₂
- () Transistor TR₃ (BC-237)
 - () jumper no terminal de coletor de TR₃
 - () jumper no terminal de emissor de TR₃
- () Transistor TR₄ (BC-237)
 - () jumper no terminal de coletor de TR₄
 - () jumper no terminal de emissor de TR₄
- () Transistor TR₅ (BC-237)
 - () jumper no terminal de coletor de TR₅
 - () jumper no terminal de emissor de TR₅
- () Transistor TR₆ (BC-237)
 - () jumper no terminal de coletor de TR₆
 - () jumper no terminal de emissor de TR₆
- () Transistor TR₇ (BC-308)
- () Transistor TR₈ (BC-308)
- () Transistor TR₉ (BC-308)
- () Transistor TR₁₀ (BC-308)
- () Transistor TR₁₁ (BC-308)
- () Transistor TR₁₂ (BC-308)

- () Transistor TR₁₃ (BC-308)
 () Transistor TR₁₄ (BC-308)

Dada à proximidade dos diversos pontos soldados, certifique-se da boa soldagem realizada.

6 – CIRCUITOS INTEGRADOS

Sendo os diodos e transistores sensíveis à temperatura, a sensibilidade dos circuitos integrados passa a ser incontesteável. Esta é uma das razões que fizeram apoiamos as memórias e a CPU sobre os soquetes. Outra razão é justamente por estes CIs serem de alta integração, os quais são mais sensíveis à eletrostática e poderão estar sujeitos a ela, com o manuseio constante de seus pinos.

A figura 16 mostra a correlação existente entre a ilustração do CI na placa com o CI na realidade. Esta correspondência deverá ser fielmente acatada para a fixação do respectivo componente.

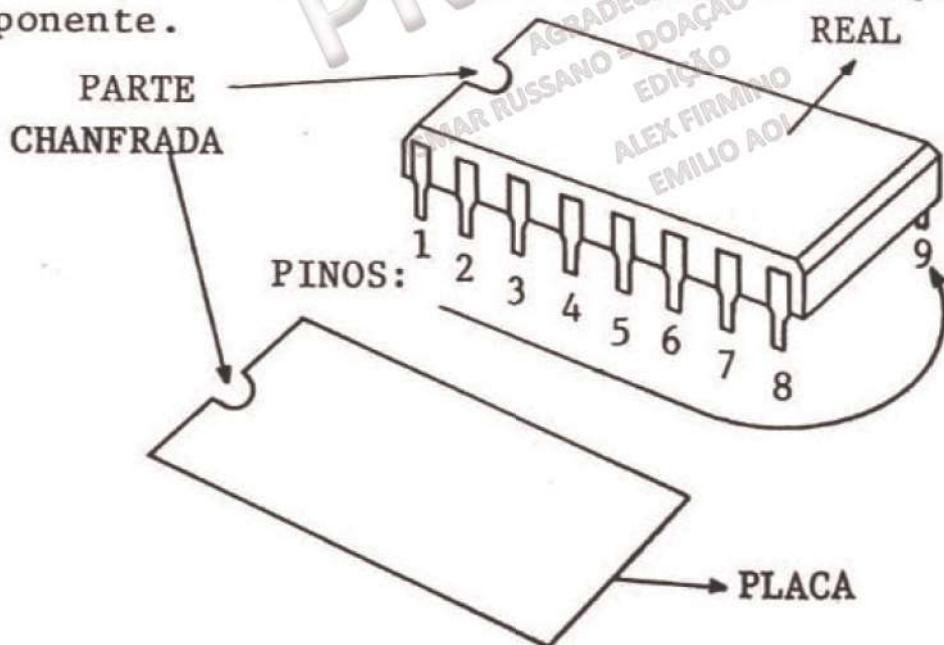


FIGURA 16

Obs.: A numeração dos pinos segue a forma de U, ordenada da esquerda para a direita.

Nesta parte da montagem, fixaremos também os soquetes. Dada à facilidade que estes apresentam de remoção ou de alocação dos circuitos integrados, os quais requerem maiores cuidados, pedimos que a CPU e as memórias sejam assentadas nos respectivos soquetes somente após o término da montagem. Contudo, siga as instruções a seguir para a fixação dos CIs:

- 1 - Pegue um componente de cada vez;

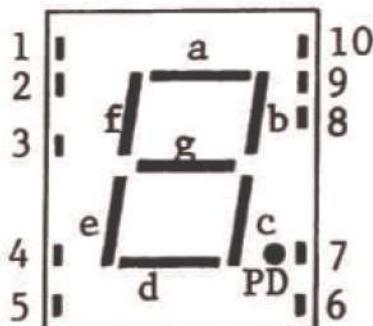
- 2 - Veja qual a posição da placa do correspondente CI e aloje-o deixando apoiado sobre a placa;
- 3 - Solde todos os pinos pela face inferior;
- 4 - Assinale, na ordem a seguir, logo após a devida fixação.

- () CI-4 (74LS14) - Solde do pino 1 ao pino 14
 () CI-5 (74LS138) - Solde do pino 1 ao pino 16
 () CI-6 (74LS08) - Solde do pino 1 ao pino 14
 () CI-7 (74LS02) - Solde do pino 1 ao pino 14
 () CI-8 (74LS75) - Solde do pino 1 ao pino 16
 () CI-9 (74LS75) - Solde do pino 1 ao pino 16
 () CI-10 (74LS126) - Solde do pino 1 ao pino 14
 () CI-11 (74LS126) - Solde do pino 1 ao pino 14
 () CI-12 (74LS174) - Solde do pino 1 ao pino 16
 () Soquete do CI-1 - Solde os 40 pinos
 () Soquete do CI-2 - Solde os 24 pinos
 () Soquete do CI-3 - Solde os 24 pinos

7 - DISPLAYS

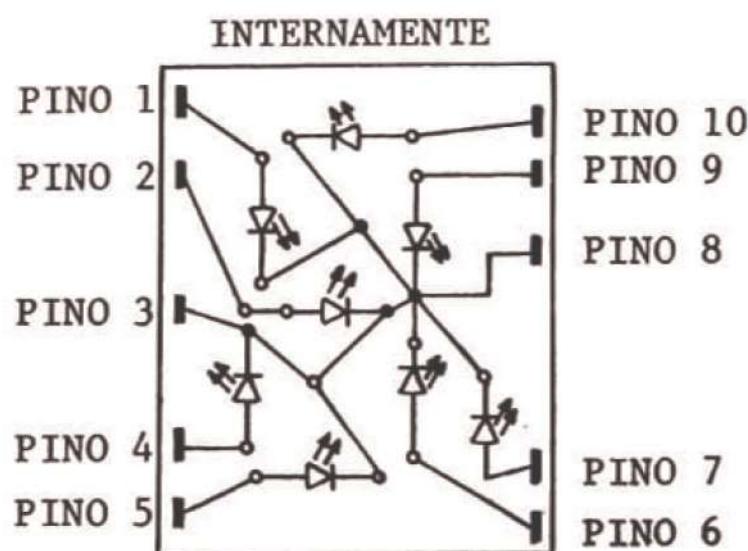
Estes são os elementos indicadores das operações realizadas com o microprocessador. São caracterizados por displays a LED, com catodo comum. A figura 17 ilustra o componente utilizado:

FIGURA 17



Configuração

- Pino 1 - segmento f
- Pino 2 - segmento g
- Pino 3 - GND (terra)
- Pino 4 - segmento e
- Pino 5 - segmento d



- Pino 6 - segmento c
- Pino 7 - ponto decimal (PD)
- Pino 8 - GND (terra)
- Pino 9 - segmento b
- Pino 10 - segmento a

Na placa, a posição do display deverá ser tal que apresente o ponto decimal voltado para a parte mais baixa da placa, isto é, para o teclado.

As mesmas instruções, bem como os mesmos cuidados para com os circuitos integrados, deverão ser seguidos para com os displays. Assim, logo após a fixação do componente, assinale na seqüência fornecida a seguir:

- () Display DP-1 - Solde do pino 1 ao pino 10
- () Display DP-2 - Solde do pino 1 ao pino 10
- () Display DP-3 - Solde do pino 1 ao pino 10
- () Display DP-4 - Solde do pino 1 ao pino 10
- () Display DP-5 - Solde do pino 1 ao pino 10
- () Display DP-6 - Solde do pino 1 ao pino 10

Certifique-se das soldagens realizadas para então prosseguir.

8 — ALIMENTAÇÃO E TESTES PRELIMINARES

Estando quase terminada a montagem, a ansiedade para ligar o Kit deve estar bem aguçada. Assim, deixaremos para o final a montagem e fixação do teclado, dando prosseguimento com uma série de testes para o Kit CEDM-80. Para isto conectaremos, de antemão, os terminais de alimentação.

No canto superior do lado direito da placa, encontram-se gravados dois círculos, caracterizando os pontos de alimentação. O terminal negativo será soldado na face superior, enquanto o terminal positivo será soldado na face inferior, da seguinte maneira:

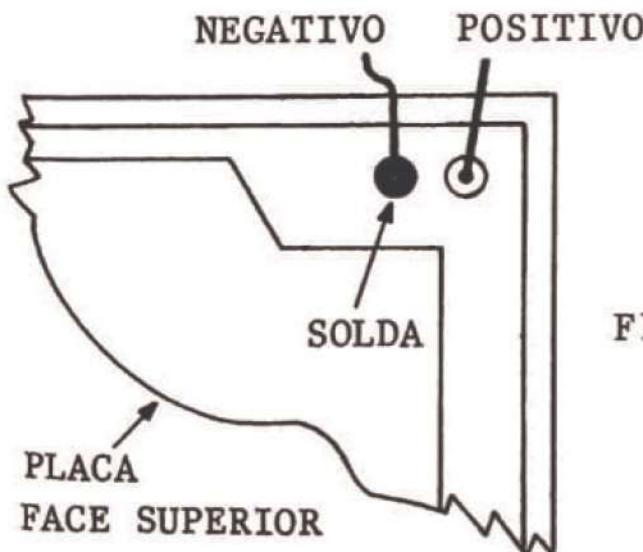
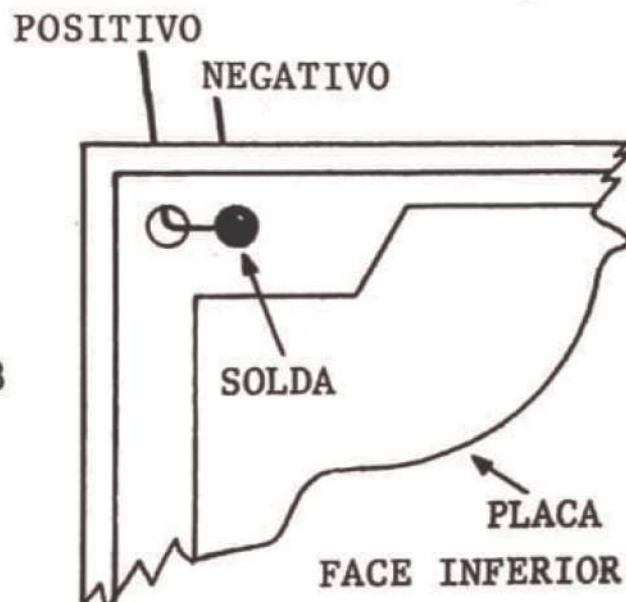


FIGURA 18



A alimentação da placa é feita com um sinal contínuo de 5 volts, o qual será proveniente da fonte CEDM-78. Assim, conecte as duas extremidades dos fios de alimentação da placa, aos correspondentes bornes da fonte (terminal positivo ao borne vermelho; terminal negativo ao borne preto).

Agora, com segurança, podem ser retiradas do invólucro as duas memórias, juntamente com a CPU e encaixá-las nos respectivos soquetes:

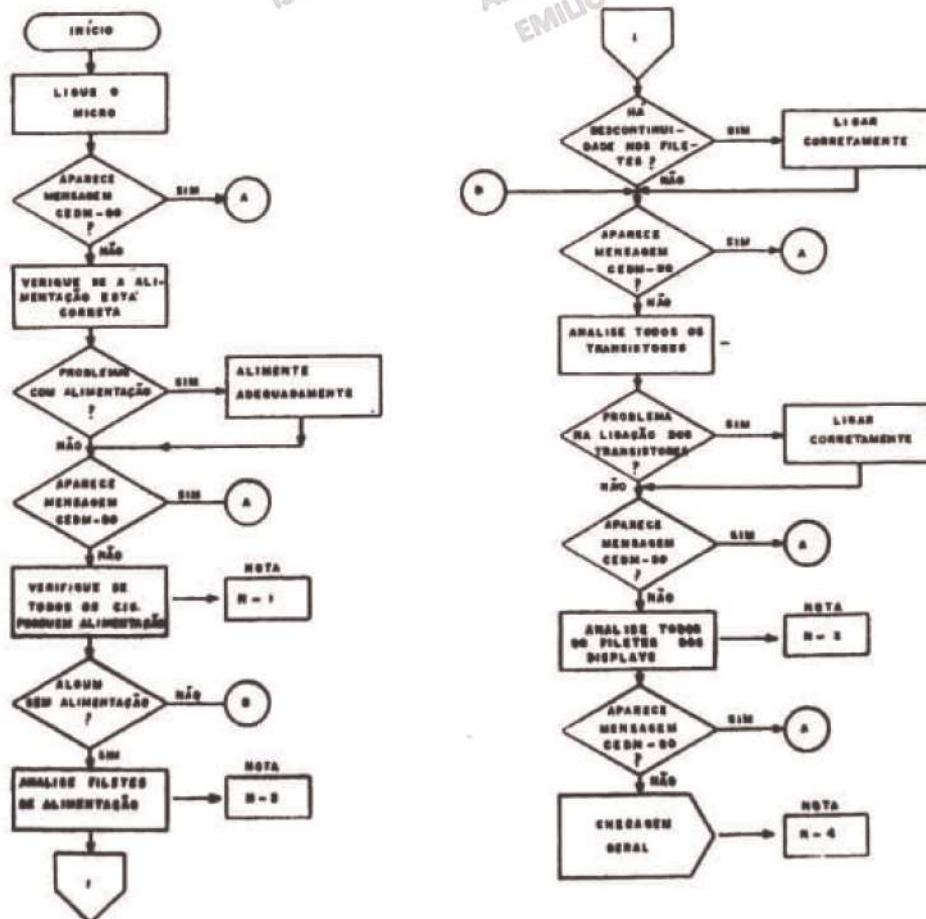
- CPU Z-80, no soquete de 40 pinos;
- Memória EPROM-2716, no soquete de 24 pinos (soquete central);
- Memória RAM-6116, no soquete de 24 pinos, situado à direita.

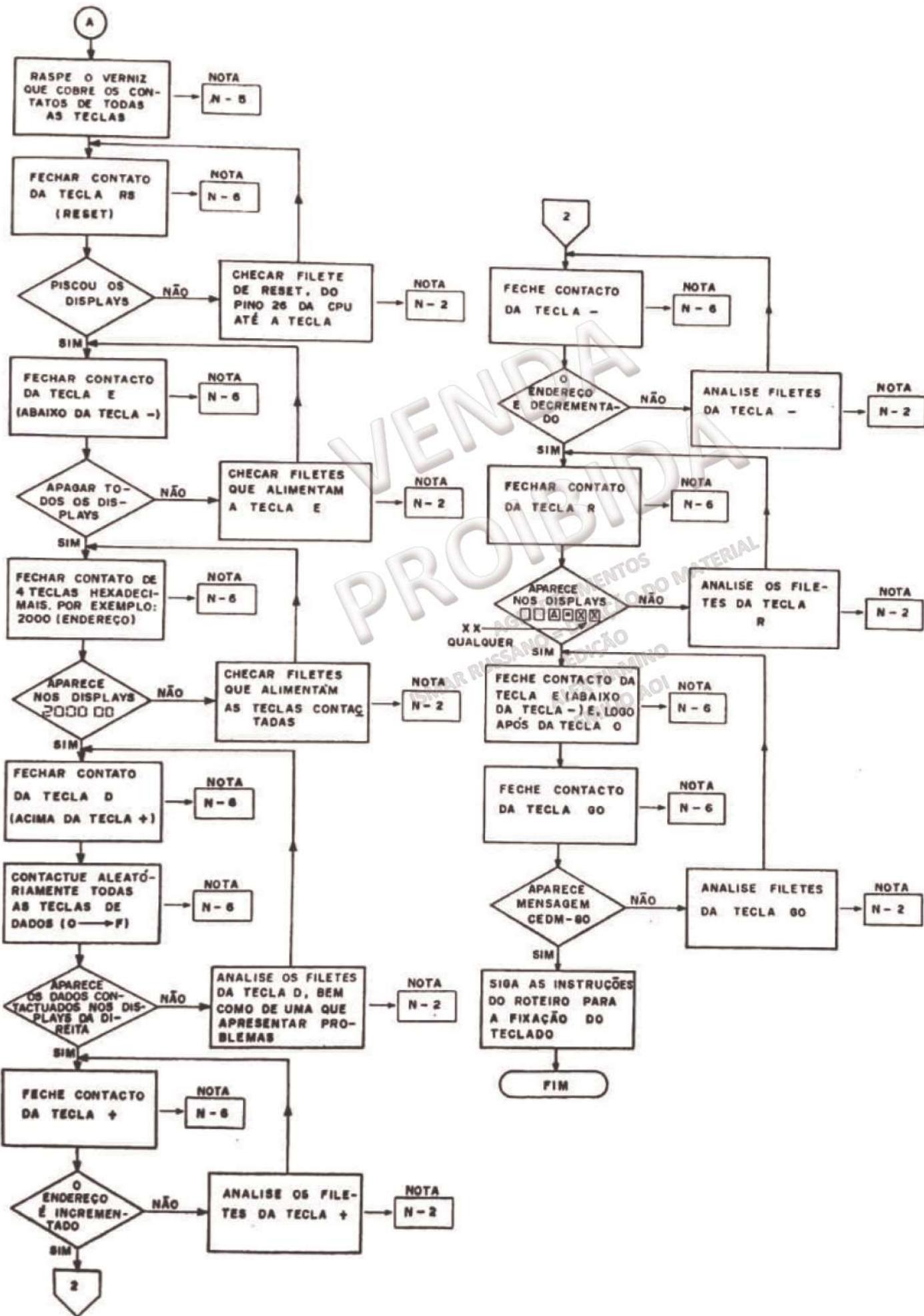
Os suportes de borracha podem também ser colocados, de preferência, o mais próximo possível dos extremos da placa.

Estando tudo pronto, ligue a fonte de alimentação na rede, o que fará aparecer nos displays do micro a mensagem:

CEDM-80

O mais prático mesmo é seguir este fluxograma (TROUBLE SHOOTING), que fornece todas as informações necessárias para o teste.





A fim de auxiliar nos testes a serem feitos, está sendo enviado juntamente com o material uma "ponta-de-testes". Veja figura.

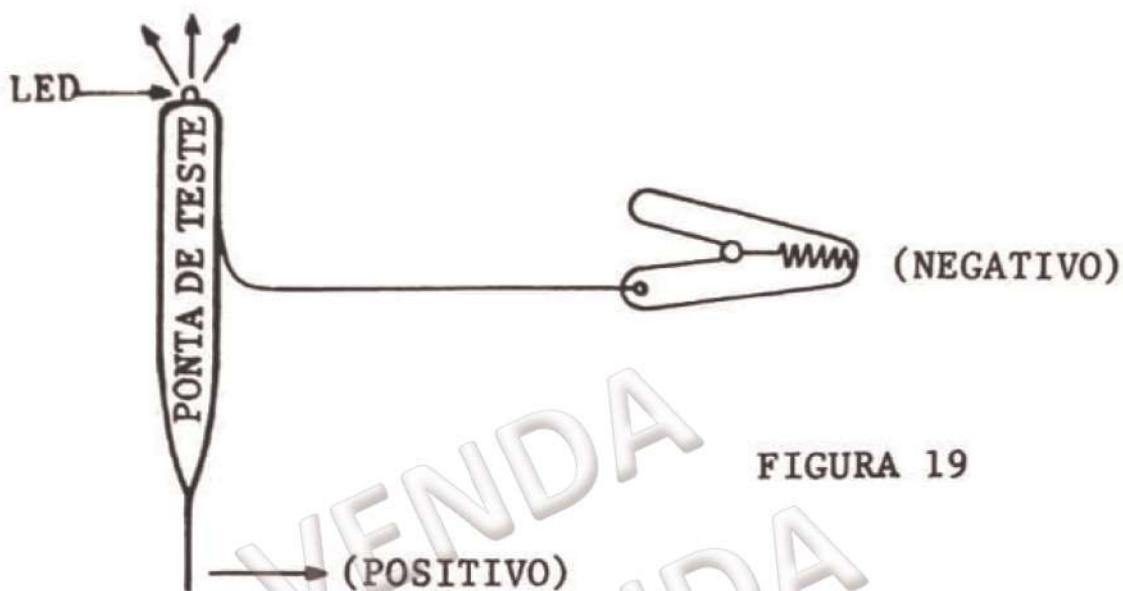


FIGURA 19

NOTAS:

N-1 - Conecte a garra da ponta-de-testes no borne negativo da fonte de alimentação e, com a outra ponta, toque no ponto de +Vcc do componente em questão. Se o LED acender, há sinal.

N-2 - Para analisar os filetes, deverá ser detectado, em todos os casos, se o mesmo apresenta continuidade. Isto é feito, desligando o micro da fonte de alimentação, para não injetar sinais indesejáveis nos pontos analisados. Conecte a garra da ponta-de-testes no polo negativo deixando livre a outra ponta. Com um fio, conecte uma das pontas no polo positivo da fonte, deixando a outra livre. Assim:

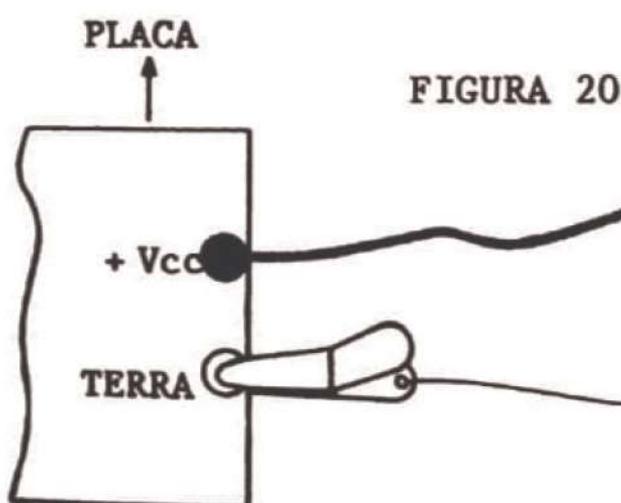
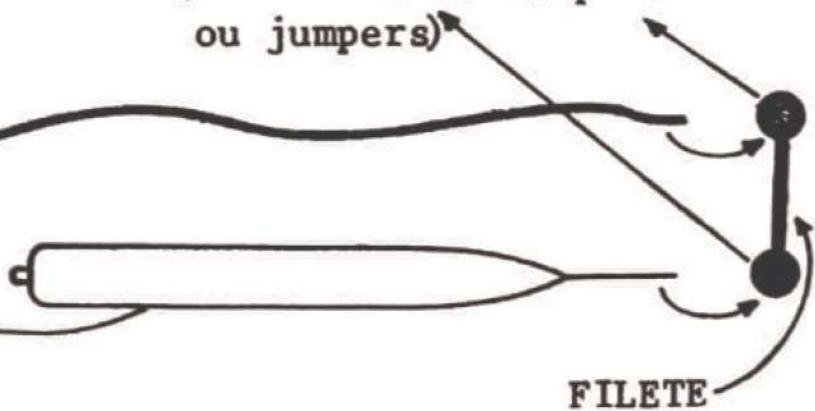


FIGURA 20

PONTOS DE CONTACTO
(terminais de componentes ou jumpers)



Se acender o led após o contacto, o filete apresenta continuidade. Se não, está rompido.

- N-3** - A emenda sobre um filete que estiver rompido é feita introduzindo uma fina camada de solda sobre o filete em questão.
- N-4** - Utilize o esquema fornecido para checar todas as ligações existentes, verificando a coerência existente para a efetuada na placa com a esquematizada. Tome como base a nota N-2.
- N-5** - Utilize uma lâmina bem fina para realizar a operação de raspagem do verniz, retirando somente sob a tecla. Cuidado para não soltar os filetes do ponto de contacto.
- N-6** - Com um material condutor, curto-circuitar os filetes do contacto sob a tecla. Poderá ser utilizada a ponta da chave de fenda.
Quando fechado o contacto das teclas de dados, poderá ser que ocorra a repetição do dado acionado (duplicação do dado). Isto é devido à não estabilidade do contacto efetuado.

9 — TECLADO

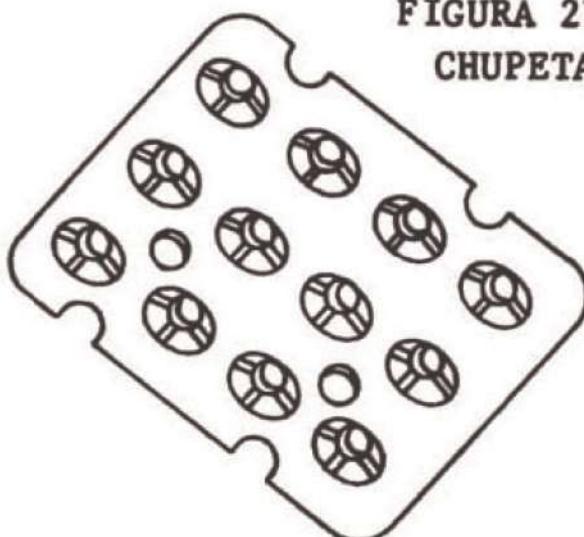
Este é o elemento responsável pela inserção de dados no micro. O conjunto compreende: a máscara, a qual funciona como suporte e invólucro para todas as partes; as teclas; as chupetas, as quais possuem a função de uma mola, agindo sobre a tecla, fazendo-a retornar à posição de repouso logo após ser acionada; uma película condutora de papel aluminizado e, finalmente, uma película isolante de plástico. Estas partes estão ilustradas na figura 21.

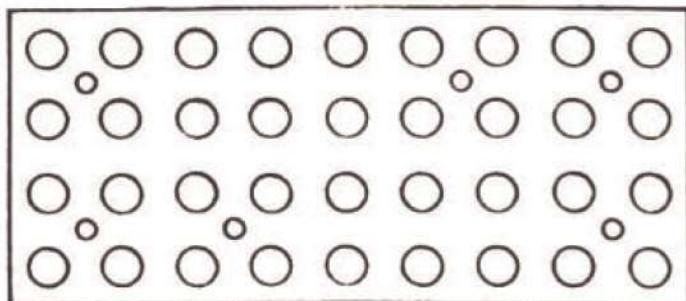
FIGURA 21



FIGURA 21-A — TECLA

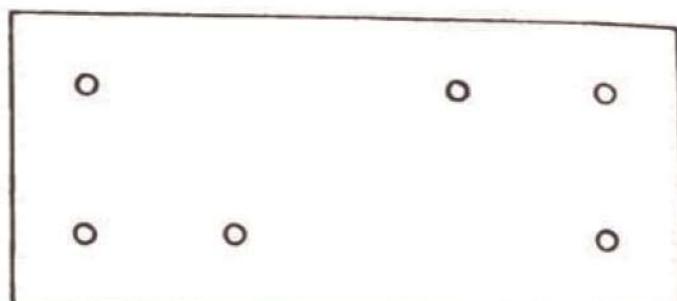
FIGURA 21-B
CHUPETA





PELÍCULA ISOLANTE

FIGURA 21 - C



PELÍCULA CONDUTORA

FIGURA 21 - D

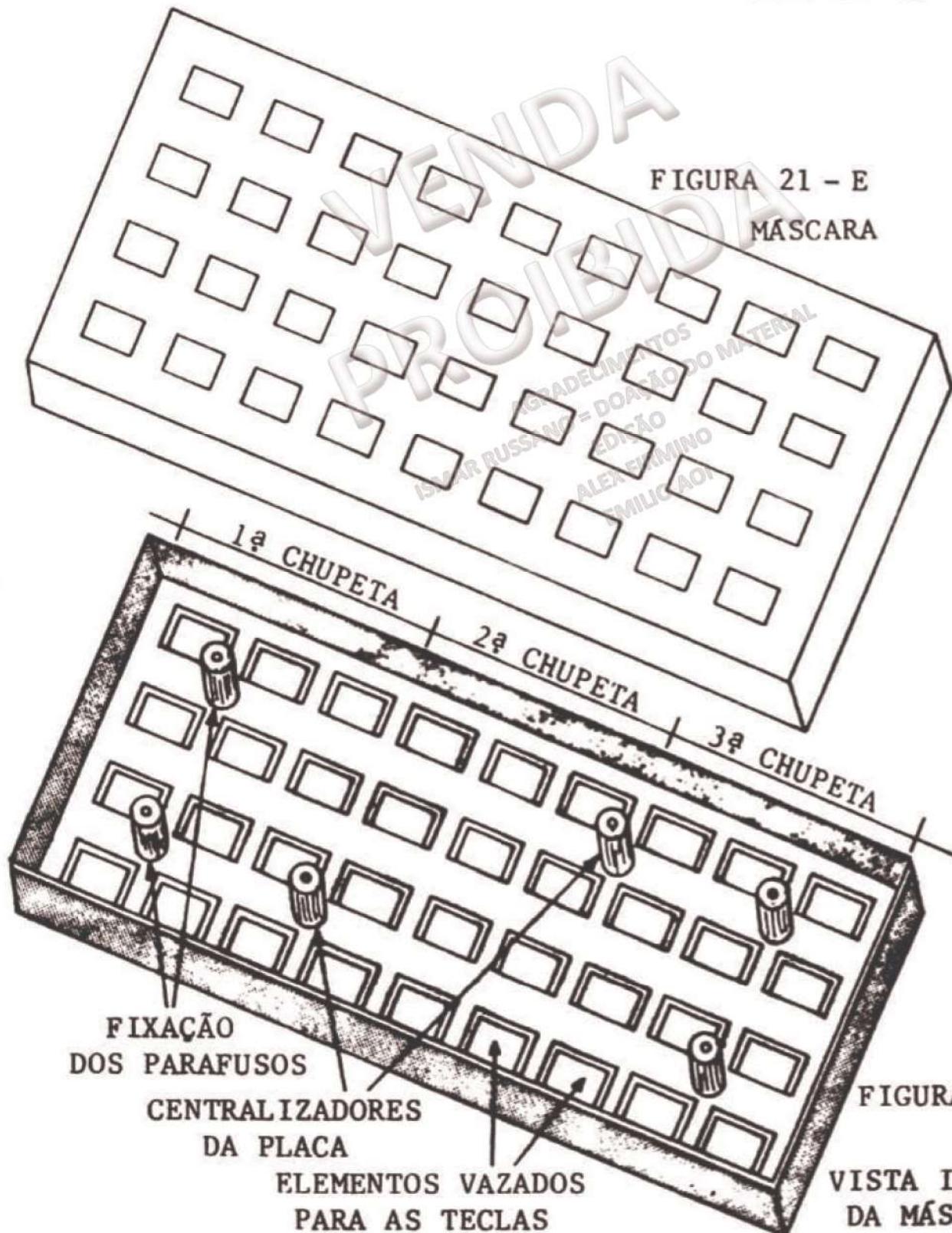


FIGURA 21 - E

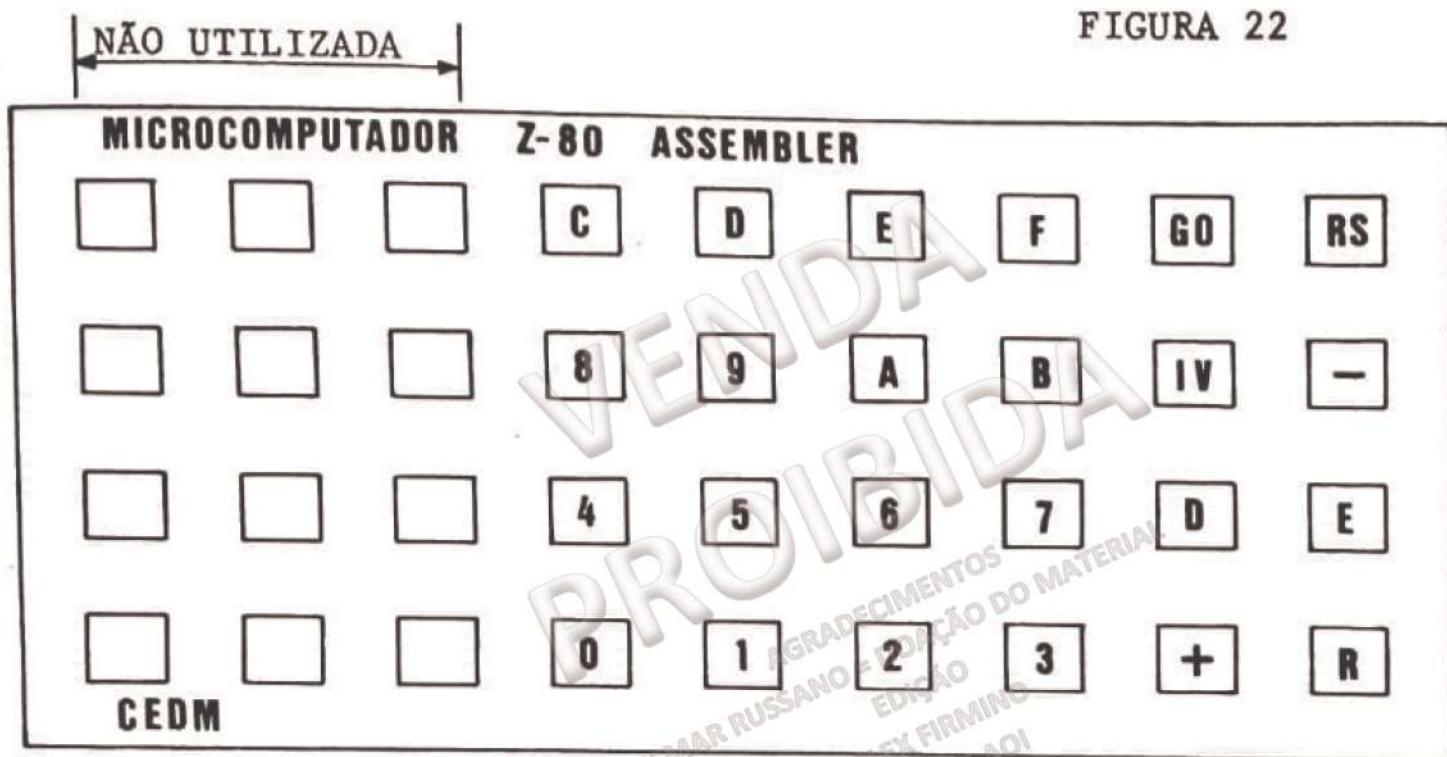
MÁSCARA

FIGURA 21-F

VISTA INTERNA
DA MÁSCARA

A fixação do teclado é simples, pois será feita através de parafusos. Antes porém, deveremos sobrepor as partes, da seguinte maneira:

- 1 - De posse da máscara, aloje todas as teclas, de maneira a apresentar a disposição mostrada na figura 22:



VISTA DO TECLADO MONTADO

- 2 - Fixe as 3 chupetas sobre as teclas, de maneira que as partes pontiagudas destas, toquem ligeiramente a base das teclas. Veja figura 23.

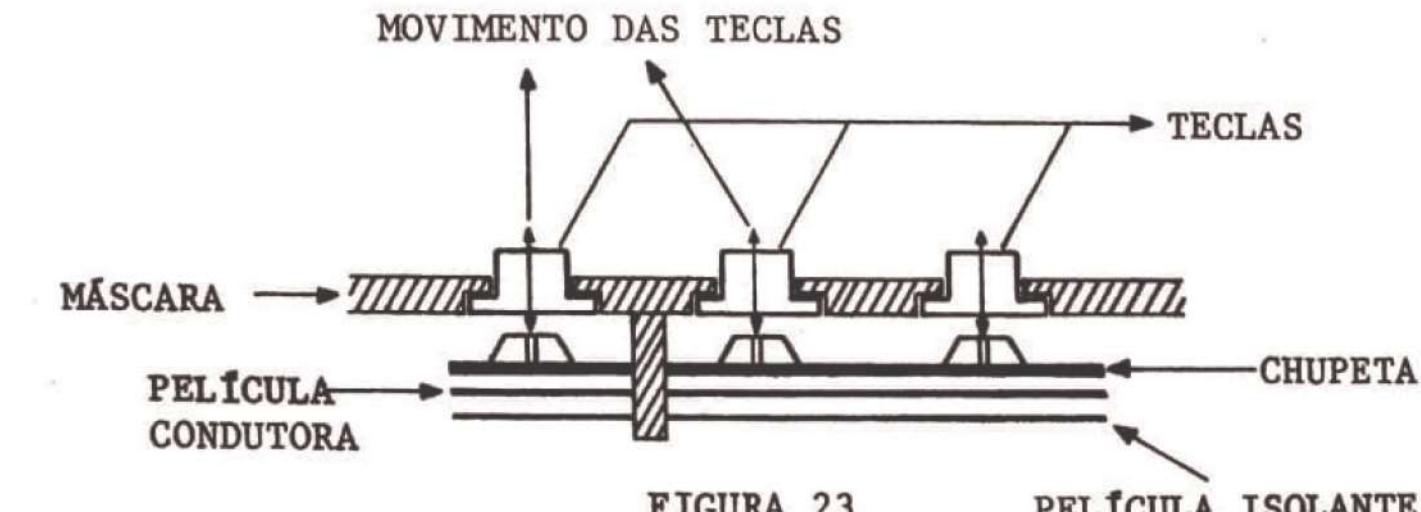


FIGURA 23

- 3 - Assente a película condutora sobre a base inferior lisa das chupetas. Se esta película estiver enrugada, deverá alisá-la com a mão, para garantir o perfeito contato.

4 - Cubra a película condutora com a película isolante.

Como já foi raspado o verniz que cobre a placa sobre os pontos de contacto das teclas, basta somente assentar o teclado. Para isso, vire o teclado com todo o cuidado possível, centralize-o na placa pelos dois centralizadores e, com os 4 parafusos recebidos, garanta a fixação, parafusando-os pela face inferior.

Com a fixação do teclado, finda-se a montagem do micro, o qual apresenta a forma ilustrada na figura 24.

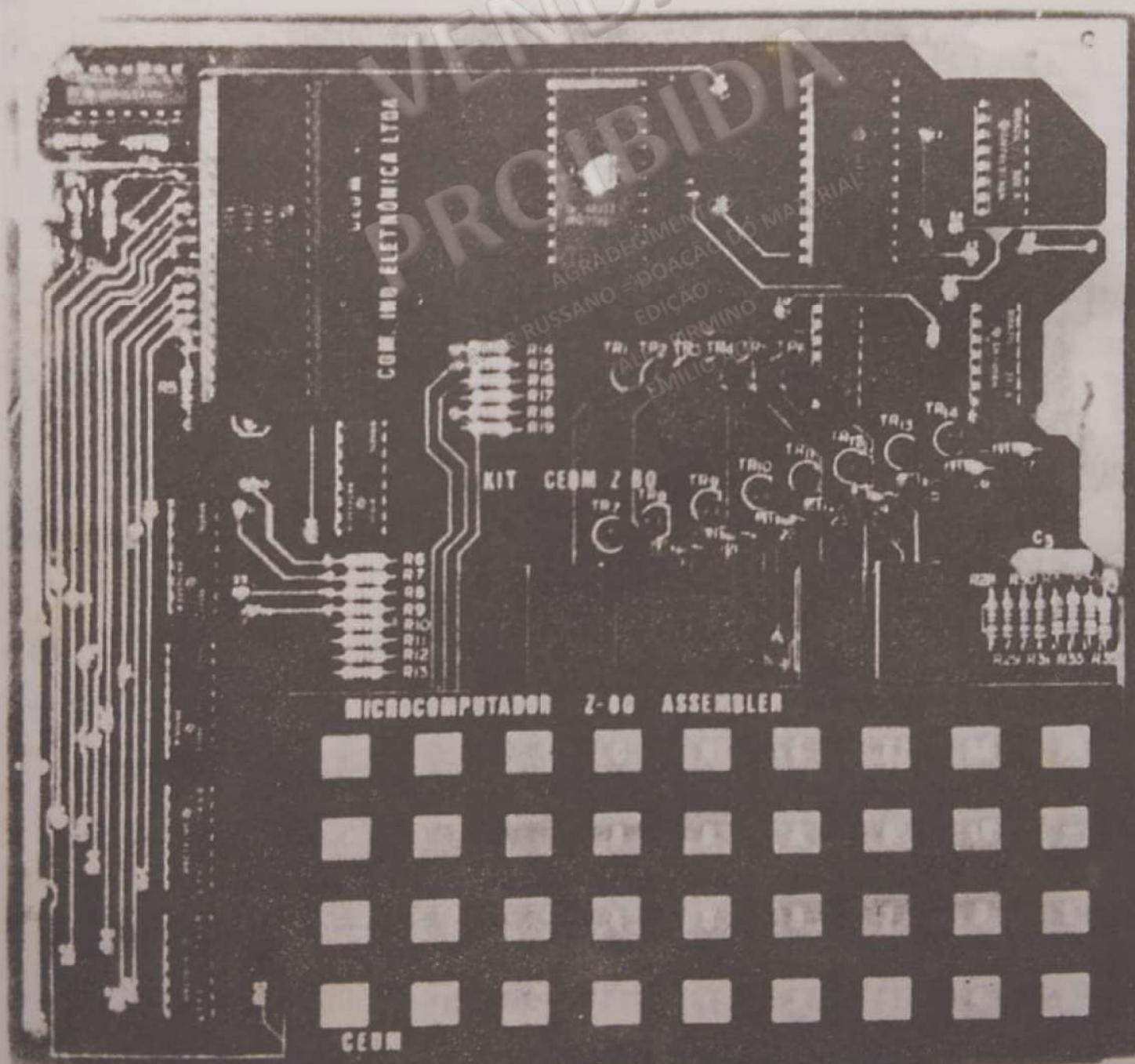


FIGURA 24



CURSO DE ELETROÔNICA DIGITAL E MICROPROCESSADORES

CAIXA POSTAL 1642 - CEP 86100 - LONDRINA - PARANÁ

EXAME DA LIÇÃO MP-26

OBSERVAÇÃO

- cada questão é composta de três ou quatro alternativas.
- só uma delas está correta.
- marque a que considerar correta.
- passe as respostas corretas para a folha de exame.

- 1) O efeito da eletrostática é devido ao:
 - a) campo magnético em torno de um filete;
 - b) efeito Joule, dadas as correntes parasitas da fonte de alimentação;
 - c) acúmulo de cargas elétricas em determinados pontos, mais que em outros;
 - d) N.R.A.
- 2) Quanto à placa de circuito impresso podemos afirmar que:
 - a) é o dispositivo que servirá para a impressão das informações;
 - b) é o elemento que, dada a sua constituição, os componentes devem ser colados e depois amarrados;
 - c) oferece a base ideal para a fixação dos componentes, pois é feita de aço, material muito resistente;
 - d) N.R.A.

3) Para o sucesso de uma soldagem deve-se:

- a) aquecer ao máximo o local de aplicação da solda, pois quanto mais quente melhor;
- b) ter a primeira preocupação com o ferro de soldar, deixando-o com uma ponta lisa;
- c) aproximar diretamente o fio de solda ao ferro de soldar e girar o ferro em torno do terminal, a fim de que a solda circunde o ponto a ser soldado;
- d) as alternativas a, b e c estão corretas.

4) Se, porventura, ocorrer a união entre filetes próximos, deve-se:

- a) ignorar a união, pois esta facilitará a propagação do sinal, dado o aumento da área do filete;
- b) aquecer e retirar o excesso com um pano molhado;
- c) aquecer e, com o ferro de soldar, procurar dividir o excesso de solda entre várias partes da placa;
- d) aquecer e, com um pedaço de madeira, retirar o excesso de solda aplicado.

5) Assinale a alternativa errada:

- a) A placa de circuito impresso, sendo dupla face, em alguns pontos será necessário unir filetes situados em faces opostas;
- b) A fixação dos jumpers é feita com aplicação da solda sómente através de uma face, pois o estanho, quando derretido, escorrerá para a outra face;
- c) Poderá ocorrer um jumper junto ao terminal do componente;
- d) Os fios jumpeadores deverão ser dobrados rente à placa para garantir uma solda baixa.

6) Os resistores que apresentam as faixas coloridas: laranja,

laranja e vermelho, caracterizam uma resistência de:

- a) $3K3\Omega$;
- b) $7K7\Omega$;
- c) 2200Ω ;
- d) 330Ω .

7) Dentre os capacitores recebidos, o único que possui a polaridade pré-fixada é o:

- a) capacitor de papel;
- b) capacitor de cerâmica;
- c) capacitor eletrolítico;
- d) capacitor de poliéster.

8) Os soquetes para os circuitos integrados servem unicamente para:

- a) colocar em destaque na placa a CPU e as memórias;
- b) proteger os circuitos integrados quanto ao manuseio constante de seus pinos, bem como facilitar a alocação e remoção dos mesmos;
- c) amplificar o sinal a ser recebido pelo circuito integrado;
- d) N.R.A.

9) Para analisar a continuidade de um filete da placa deve-se:

- a) deixar o micro ligado, tocar com as mãos em dois pontos do filete e verificar se dá choque;
- b) desligar o micro e verificar a existência de sinal, conectando um led nos extremos do filete;
- c) desligar o micro e utilizar a ponta-de-testes, que deve rá estar conectada no terminal negativo da fonte e, com um outro fio ligado ao terminal positivo, verificar se o led da ponta-de-testes acende;

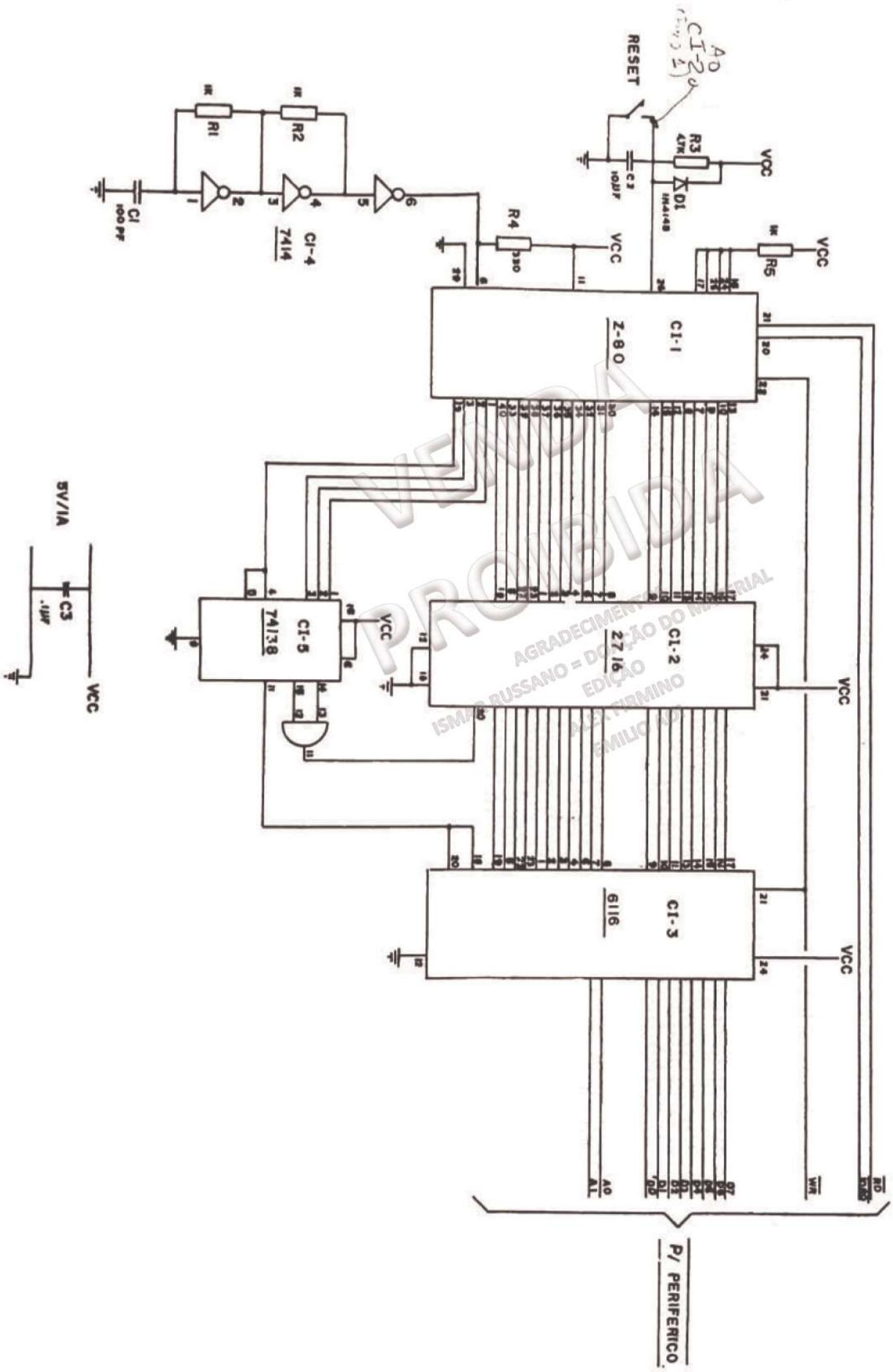
- d) utilizar qualquer procedimento mostrado nas alternativas a, b e c.

10) Assinale a alternativa correta:

- a) A película condutora de papel aluminizado serve para fechar o contacto de uma tecla, quando pressionada;
- b) A película isolante serve para permitir contacto apenas da tecla pressionada;
- c) A chupeta age como mola para a tecla, pois força esta a assumir a posição de repouso, quando pressionada
- d) as alternativas a, b e c estão corretas.

VENDEDA
PROIBIDA
AGRADECIMENTOS
ISMAR RUSSANO = DOAÇÃO DO MATERIAL
EDIÇÃO
ALEX FIRMINO
EMILIO AOI

DIAGRAMA ESQUEMÁTICO CEDM-280 – CPU



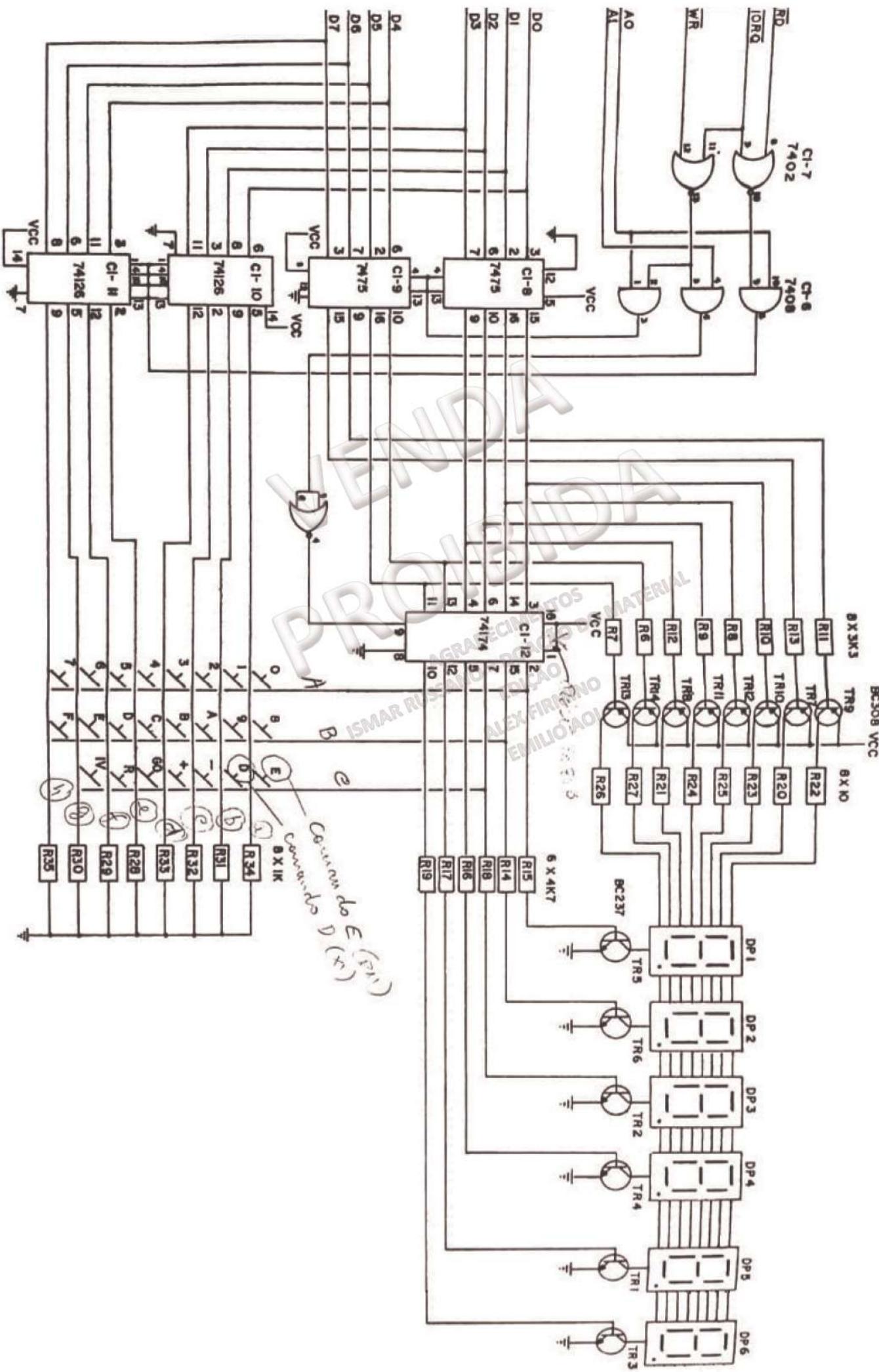


DIAGRAMA ESQUEMÁTICO CEDM-280 – PERIFÉRICOS