

**MANUAL DE INSTALAÇÃO
E OPERAÇÃO DO CARTÃO
DE EXPANSÃO DE MEMÓRIA
PARA 128K/192K**

2ª Edição — 1984

SPECTRUM

MICRO **engenhos**²

MICRO engenho²

**MANUAL DE INSTALAÇÃO
E OPERAÇÃO
DO CARTÃO DE EXPANSÃO
DE MEMÓRIA PARA 128K/192K**

ÍNDICE

PREFÁCIO	03
<i>PROPÓSITOS DESTES MANUAL</i>	03
1. INTRODUÇÃO	07
2. INSTALANDO O CARTÃO	11
<i>REMOVENDO A TAMPA DA UNIDADE DE</i> <i>PROCESSAMENTO</i>	11
<i>LOCALIZANDO OS CONECTORES DE EXPAN-</i> <i>SÃO</i>	12
<i>CONECTANDO O CARTÃO</i>	13
<i>RECOLOCANDO A TAMPA DA UNIDADE DE</i> <i>PROCESSAMENTO</i>	14
3. O PROGRAMA MOVESOD	17
<i>A MEMÓRIA DO COMPUTADOR</i>	18
<i>O MECANISMO DO PROGRAMA MOVESOD</i> ...	20
<i>AS CARACTERÍSTICAS DO SOD RELOCADO</i> ..	22
<i>USANDO O PROGRAMA MOVESOD</i>	23
<i>A COMPATIBILIDADE DO SOD RELOCADO</i> <i>COM O SOFTWARE EXISTENTE</i>	23
<i>AS MODIFICAÇÕES NO PROGRAMA FID</i>	24
4. O PROGRAMA RAMEXPANDE	27
<i>AS CARACTERÍSTICAS DO PROGRAMA RA-</i> <i>MEXPANDE</i>	27
<i>O MECANISMO DO PROGRAMA RAMEXPAN-</i> <i>DE</i>	29
<i>USANDO O PROGRAMA RAMEXPANDE</i>	30
<i>SUMÁRIO DAS FUNÇÕES</i>	33
<i>SUMÁRIO DE CÓDIGOS DE ERROS</i>	34
<i>CUIDADOS A SEREM TOMADOS</i>	35
<i>DESCRIÇÃO DOS COMANDOS</i>	36

5. O PROGRAMA PSEUDO DISCO	49
<i>AS CARACTERÍSTICAS DO PROGRAMA PSEUDO DISCO</i>	49
<i>O MECANISMO DO PROGRAMA PSEUDO DISCO</i>	49
<i>USANDO O PROGRAMA PSEUDO DISCO</i>	50
<i>CUIDADOS A SEREM TOMADOS</i>	51
6. INFORMAÇÕES TÉCNICAS	55
<i>ENDEREÇAMENTO DA MEMÓRIA DO CARTÃO</i>	55
<i>CONTROLANDO A EXPANSÃO</i>	57
<i>USO DE MEMÓRIA</i>	60
<i>EXECUTANDO O PROGRAMA DE TESTE DE MEMÓRIA</i>	60
7. ESPECIFICAÇÕES	65
<i>MEMÓRIA</i>	65
<i>CONSUMO MÉDIO DE ENERGIA</i>	65

É proibida a reprodução total ou parcial deste Manual, sem prévia autorização do autor.

PROPÓSITOS DESTES MANUAL 03

PREFÁCIO

PROPÓSITOS DESTA MANUAL

Este manual tem dois propósitos: ensiná-lo a instalar o Cartão de Expansão de Memória para 128/192K bytes, e ainda, ensiná-lo a usar os programas aplicativos do disco que acompanha o Cartão. Se você tem pouca experiência em programação, provavelmente sentirá certa dificuldade para usar esses programas. Pensando nisso, escrevemos este manual em uma linguagem simples e de fácil compreensão.

Evidentemente que fizemos isso sem reduzir o grau de informação, para não prejudicar os usuários mais experientes.

Os programas discutidos neste manual têm uma gama muito grande de aplicações. Por isso, achamos que você não deve simplesmente ler o manual, mas sim, estudá-lo com muita atenção.

Como você deve ter notado pelo nome do Cartão, ele possui duas versões: uma que expande a memória para 128K e outra que expande para 192 K bytes. A única diferença entre elas reside no número de componentes eletrônicos, mas ambas podem ser instaladas, indistintamente, no microENGENHO ou no microENGENHO².

.....

1. INTRODUÇÃO

1. INTRODUÇÃO

O Cartão que expande a memória para 128K bytes ou o Cartão que expande a memória para 192K bytes ampliam consideravelmente os recursos do seu microENGENHO. Você pode usar mais de um Cartão, o limite será o número de conectores disponíveis na Placa Expansora de Periféricos.

Ao adquirir uma das versões do Cartão você recebe:

- Uma placa que contém os componentes eletrônicos e que a partir de agora será chamada de "Cartão";
- Um disco contendo alguns programas aplicativos;
- Um Certificado de Garantia;
- Este Manual de Instalação e Operação.

O microENGENHO equipado com 64K bytes de memória (Pascal ou Fortran) ou o microENGENHO equipado com o Cartão Z-80 (CP/M, Cobol 80, Fortran 80, etc.) é compatível com este novo Cartão. Como você verá, esta compatibilidade é possível devido à organização da memória RAM do Cartão, que é feita através de vários bancos de 16K bytes cada. Dessa forma, é como se tivéssemos vários cartões de 16K ocupando um único conector da Placa Expansora de Periféricos.

2. INSTALANDO O CARTÃO

REMOVENDO A TAMPA DA UNIDADE DE PROCESSAMENTO 11

LOCALIZANDO OS CONECTORES DE EXPANSÃO 12

CONECTANDO O CARTÃO 13

RECOLOCANDO A TAMPA DA UNIDADE DE PROCESSAMENTO 14

2. INSTALANDO O CARTÃO

REMOVENDO A TAMPA DA UNIDADE DE PROCESSAMENTO

Antes de mais nada, esteja certo de que o Me2 está desligado da rede de energia elétrica. A seguir, retire os quatro parafusos da tampa da UNIDADE DE PROCESSAMENTO com o auxílio de chave de fenda. Cumprida esta etapa, remova a tampa e coloque-a em lugar seguro de arranhões. A figura 01 apresenta uma vista dessa desmontagem.

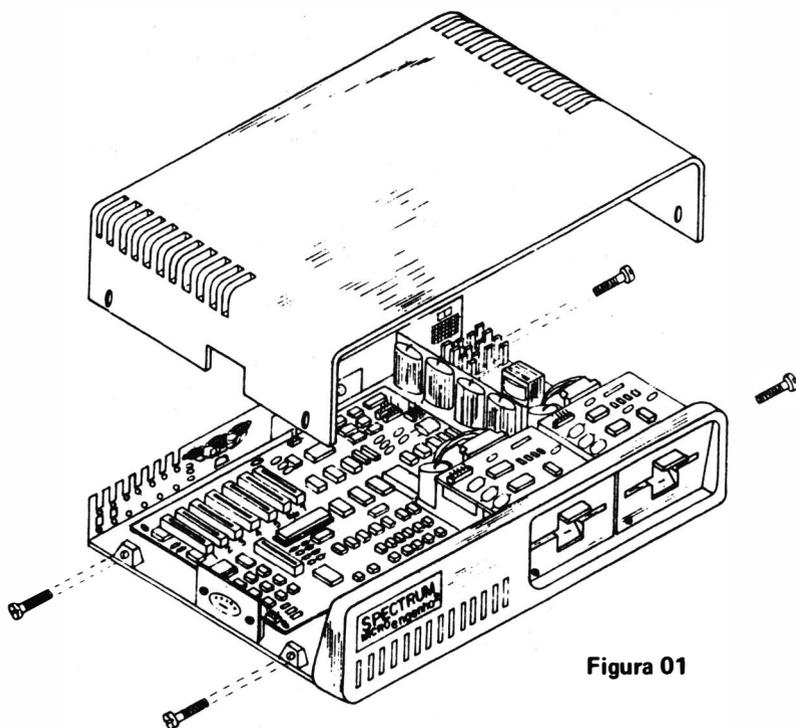


Figura 01

LOCALIZANDO OS CONECTORES DE EXPANSÃO

Com o Me2 aberto, de acordo com as instruções anteriores, olhe para seu canto esquerdo traseiro. Nessa região você encontrará uma série de conectores cuja finalidade é possibilitar a ligação do Me2 a acessórios, periféricos, expansões, etc. Observe que esses conectores estão numerados de 1 a 7. O Cartão pode ser acoplado em qualquer um desses conectores, nós escolhemos o conector número 5 apenas para facilitar os exemplos futuros. Veja agora a figura 02.

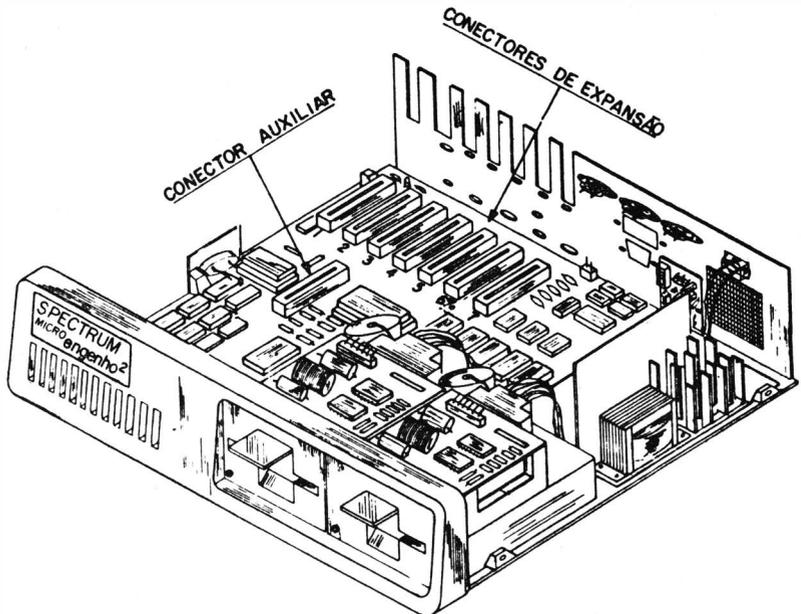


Figura 02

CONECTANDO O CARTÃO

Segurando firmemente o Cartão, evitando tocar em seus contatos dourados ou em seus componentes eletrônicos encaixe-o no conector 5, exatamente como mostra a figura 03.

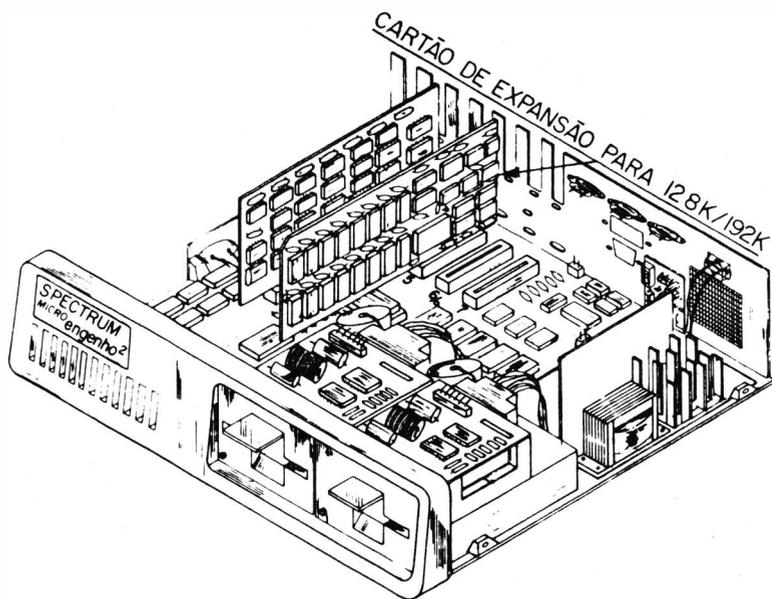


Figura 03

RECOLOCANDO A TAMPA DA UNIDADE DE PROCESSAMENTO

Após efetuar a ligação do Cartão, recoloque a tampa da UNIDADE DE PROCESSAMENTO, reaperte os quatro parafusos e finalmente ligue o Me2 à rede de energia elétrica. Faça tudo isso com bastante cuidado e continue a leitura deste manual a fim de inteirar-se de outras informações.

3. O PROGRAMA MOVESOD

A MEMÓRIA DO COMPUTADOR 18

O MECANISMO DO PROGRAMA MOVESOD 20

AS CARACTERÍSTICAS DO SOD RELOCADO 22

USANDO O PROGRAMA MOVESOD 23

*A COMPATIBILIDADE DO SOD RELOCADO COM O SOFTWARE
EXISTENTE 23*

AS MODIFICAÇÕES DO PROGRAMA FID 24

3. O PROGRAMA MOVESOD

Na embalagem do Cartão você recebeu um disco com alguns programas aplicativos. Esses programas, que são: **o MOVESOD, o RAMEXPANDE e o PSEUDO DISCO**, permitem que você use a memória disponível em seu Cartão. Primeiramente descreveremos o MOVESOD.

Com este programa, você pode usar a área de memória que até aqui era de uso exclusivo do Sistema de Operação em Disco (SOD). Para compreender como o MOVESOD faz isto, antes você precisa entender alguns conceitos fundamentais.

A MEMÓRIA DO COMPUTADOR

A memória principal do microENGENHO possui 64K bytes. Esses 64K estão distribuídos de acordo com a figura 04, que mostra o mapeamento dessa memória. Assim, existem alguns espaços previamente reservados para exercerem funções específicas. Observando a figura 04, você notará que o espaço de \$0000 à \$9600 está reservado aos programas do usuário, o espaço de \$9600 à \$C000 está reservado ao SOD, o espaço de \$C000 à \$D000 está reservado à entrada e saída de dados e o espaço de \$D000 à \$FFFF está reservado ao Basic do microENGENHO.



Figura 04

Observações:

- *A área de memória de \$0000 à \$BFFF é denominada "Área de Memória RAM";*
- *A área de memória de \$D000 à \$FFFF é denominada "Área de Memória EPROM";*
- *Os endereços estão indicados em hexadecimal.*

O importante da explicação anterior é notar que existe uma área de memória RAM exclusivamente reservada ao SOD. O programa MOVESOD transfere o SOD desta área para uma área de memória do Cartão e, conseqüentemente, libera a área que até então ele vinha ocupando. Em resumo, o programa MOVESOD transfere o SOD para a memória do Cartão liberando assim uma memória adicional de 10K bytes, que você pode usar em seus programas.

Antes de prosseguir, é preciso entender mais um detalhe. O Cartão que você possui, que pode ser o que expande a memória para 128K ou o que expande a memória para 192K, é constituído por vários bancos de 16K, por exemplo, o Cartão que expande a memória para 192K é formado por oito bancos. Quando o programa MOVESOD transfere o SOD para o Cartão, ele o coloca em seu segundo banco.

Agora, para encerrar esta explicação, veja a figura 05, onde os bancos de 16K são colocados. Note que na realidade cada banco é formado por dois blocos, um de 12K e outro de 4K, que são colocados em paralelo compartilhando o espaço da memória EPROM.

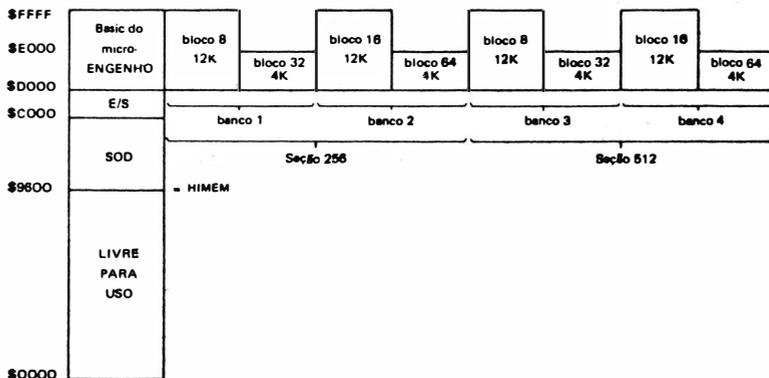


Figura 05

Observação: este mapa se refere ao Cartão que expande a memória para 128K. Para o Cartão que expande a memória para 192K basta aumentar quatro bancos ou duas seções.

O MECANISMO DO PROGRAMA MOVESOD

Como dissemos anteriormente, o programa MOVESOD faz exatamente o que o seu nome diz, ou seja, move o SOD da área de memória RAM do microENGENHO, para o segundo banco de 16K do Cartão. Na transferência, todo o espaço do banco menos 4K é utilizado pelo SOD e pelas rotinas associadas ao monitor. Esses

4K que sobram serão utilizados por um outro programa que será visto mais adiante.

A seguir, apresentamos a figura 06, que ilustra tudo o que foi dito até aqui. Note nessa figura, que existe uma pequena parte de SOD que não é transferida para o segundo banco do Cartão. Essa pequena parte, de \$BE000 a \$BFFF, permanece na memória principal com as rotinas que habilitam e desabilitam o segundo banco do Cartão.

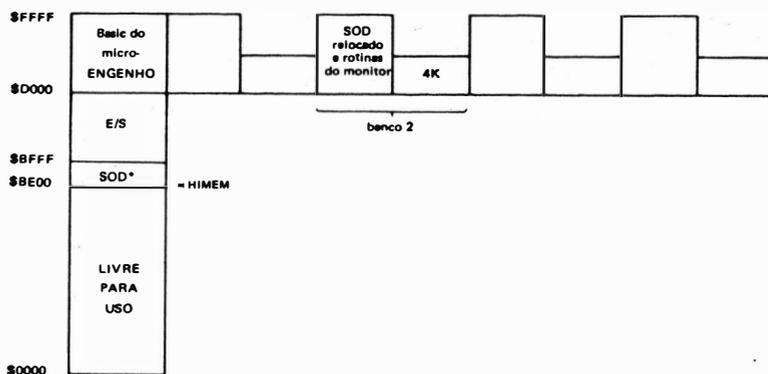


Figura 06

Observação: o comando *MAXFILES* do Basic estabelece o número de buffers para o uso do SOD. O segundo banco do Cartão suporta todo o SOD e mais três buffers. Sendo assim, se você usar um comando *MAXFILES* maior que três, os buffers adicionais serão colocados na memória principal em blocos de 595 bytes e a partir do endereço \$BE00. Veja a figura anterior, nela este endereço é chamado de HIMEM.

AS CARACTERÍSTICAS DO SOD RELOCADO

Após o SOD ter sido transferido para o Cartão, ele permanece desativado e à sua disposição. Assim, quando um programa o ativa, ele age da seguinte maneira:

- O Segundo banco do Cartão fica habilitado e com isso o Basic do microENGENHO é desativado, uma vez que eles compartilham o mesmo endereço da memória;
- O comando do SOD que foi solicitado pelo programa é executado e a seguir o Basic é novamente recocado no sistema.

Note que: *após o comando do SOD ter sido executado, o segundo banco é desabilitado para que o Basic do microENGENHO reassuma.*

Todas as características do SOD normal são mantidas no SOD relocado, inclusive a capacidade de inicializar discos. Para facilitar o trabalho, o programa MOVESOD só precisa ser executado uma vez, pois, ao inicializar um disco o SOD relocado fica nele gravado, assim, quando mais tarde este disco for executado, o SOD será automaticamente transferido para o segundo banco do Cartão.

Um outro detalhe importante é que você pode usar mais de um Cartão. O programa MOVESOD procura pelo primeiro Cartão, a partir do conector número 0 da Placa Expansora de Periféricos.

Observações:

- *Faça uma cópia do disco que contém o programa MOVESOD assim que recebê-lo;*
- *O programa MOVESOD faz todas as alterações necessárias para possibilitar a comunicação com o SOD relocado.*

USANDO O PROGRAMA MOVESOD

Agora vamos ensiná-lo a executar o programa MOVESOD a partir do SOD normal. Para isso, siga os seguintes passos:

Primeiro: carregue o SOD normal no microENGENHO com o disquete mestre;

Segundo: insira o disco que contém uma cópia do programa MOVESOD ou então o próprio original e a seguir digite: **BRUN MOVESOD CR.** Na tela será apresentada uma mensagem indicando que a relocação do SOD está sendo efetuada. Imediatamente após o término desta o controle retornará ao Basic.

Observação: *a partir de agora, aconselhamos você a rotular todos os discos que forem inicializados com este SOD relocado, porque desse modo não os confundirá com aqueles inicializados pelo SOD normal.*

A COMPATIBILIDADE DO SOD RELOCADO COM O SOFTWARE EXISTENTE

Na maioria dos programas existentes você poderá tratar o SOD relocado como o SOD normal. Entretanto, podem surgir problemas com aqueles programas que usam rotinas do SOD normal. Nesse caso você precisará fazer modificações nesses programas, como é o caso do programa FID que será comentado a seguir.

AS MODIFICAÇÕES NO PROGRAMA FID

O programa FID, oferecido com o disquete mestre, só pode ser usado com o SOD relocado após sofrer algumas modificações. Para realizar essas modificações use o programa MOD, que também faz parte do disco que contém o MOVESOD, da seguinte forma: carregue o disco que você recebeu no Acionador de Discos e digite **BRUN MOD CR**. A partir desse ponto, basta acompanhar as instruções da tela que são auto-instrutivas.

4. PROGRAMA RAMEXPANDE

AS CARACTERÍSTICAS DO PROGRAMA RAMEXPANDE 27

O MECANISMO DO PROGRAMA RAMEXPANDE 29

USANDO O PROGRAMA RAMEXPANDE 30

SUMÁRIO DAS FUNÇÕES 33

SUMÁRIO DE CÓDIGOS DE ERRO 34

CUIDADOS A SEREM TOMADOS 35

DESCRIÇÃO DOS COMANDOS 36

4. O PROGRAMA RAMEXPANDE

Este programa possibilita que o Basic do microENGENHO e o Basic Inteiro utilizem a memória RAM, que é adicionada com a presença de uma das versões dos Cartões no sistema. Dessa maneira, informações na forma de dados e até mesmo programas podem ser armazenados nos bancos de 16K do Cartão para serem utilizados futuramente.

AS CARACTERÍSTICAS DO PROGRAMA RAMEXPANDE

O programa RAMEXPANDE ocupa duas áreas distintas de memória. Sua maior parte está localizada naqueles 4K do segundo banco, que não foram ocupados pelo programa MOVESOD, enquanto que o restante ocupa uma área da memória principal do microENGENHO, denominada página \$BD.

O programa RAMEXPANDE é composto por outros três programas, que são: o RAMEXPANDE propriamente dito, o RAMEXPANDE.LOW e o RAMEXPANDE.LOAD. O RAMEXPANDE.LOAD serve para carregar a memória e posicionar alguns ponteiros, o RAMEXPANDE é o programa que está localizado nos 4K do segundo banco do Cartão e o RAMEXPANDE.LOW é o programa que reside na página \$BD da memória principal.

Basicamente, o programa RAMEXPANDE armazena programas e informações em áreas independentes da memória RAM do Cartão. Cada uma dessas áreas independentes, que são chamadas de segmentos, deve receber um número inteiro de 1 a 32727, para que possam ser identificados quando for necessário.

O RAMEXPANDE trata cada banco de 16K como um conjunto de dois blocos de memória: um de 12K e outro de 4K. Desta forma, o Cartão que expande a memória para 192K possui 16 blocos, oito de 12K e oito de 4K, enquanto que o Cartão que expande a memória para 128K possui oito, quatro de 12K e quatro de 4K. Agora, analise a figura 05 e observe que esses blocos são facilmente identificados, uma vez que cada um possui um número. Note também, que cada seção de 32K recebe uma numeração, que será útil futuramente.

A seguir, apresentamos também a tabela 01 para que você revise tudo o que foi explicado até aqui a respeito do mapeamento de memória.

Banco	Bloco	Memória
<i>banco1</i>	8	12K
<i>banco 1</i>	32	4K
<i>banco2</i>	16	12K
<i>banco2</i>	64	4K
<i>banco3</i>	8	12K
<i>banco3</i>	32	4K
<i>banco4</i>	16	12K
<i>banco4</i>	64	4K

Tabela 01

Observação: a figura 05 e a tabela 01 se referem ao Cartão de Expansão de Memória para 128K Bytes. Para o Cartão de Expansão de Memória para 192K Bytes basta aumentar quatro bancos na figura 05 e analisar a tabela 02 a seguir.

Banco	Bloco	Memória
<i>banco1</i>	8	12K
<i>banco1</i>	32	4K
<i>banco2</i>	16	12K
<i>banco2</i>	64	4K
<i>banco3</i>	8	12K
<i>banco3</i>	32	4K
<i>banco4</i>	16	12K
<i>banco4</i>	64	4K
<i>banco5</i>	8	12K
<i>banco5</i>	32	4K
<i>banco6</i>	16	12K
<i>banco6</i>	64	4K
<i>banco7</i>	8	12K
<i>banco7</i>	32	4K
<i>banco8</i>	16	12K
<i>banco8</i>	64	4K

Tabela 02

O MECANISMO DO PROGRAMA RAMEXPANDE

Para usar o programa RAMEXPANDE, você primeiro deve armazená-lo na memória do microENGENHO e em seguida dizer a ele o que pretende fazer. O mecanismo para mandar o programa RAMEXPANDE realizar uma tarefa, se resume na construção de uma variável denominada "variável STRING". Essa variável tem a seguinte forma:

CM\$ = "A, B, E, D, L"

Então, é atribuindo valores aos parâmetros **A, B, E, D** e **L**, que você determina a tarefa que o programa RAMEXPANDE deve realizar.

Observação: *nem sempre você precisa definir todos os parâmetros para que ele realize uma tarefa, ou seja, existem algumas tarefas que ele realiza conhecendo menos que cinco parâmetros.*

Em geral, esses parâmetros representam o seguinte:

Parâmetro A: identifica a função a ser executada e deve receber valores de 0 a 16. Para encontrar o número associado a cada função consulte a tabela 03.

Parâmetro B: identifica o segmento em uso. Pode receber valores de 1 a 32727 (se você não se recorda o que é segmento, leia novamente "AS CARACTERÍSTICAS DO PROGRAMA RAMEXPANDE na página 27 deste manual).

Parâmetro E: trata-se de um código de erros, deve ser sempre uma variável. Sua função é informá-lo se a tarefa foi ou não bem sucedida. O programa RAMEXPANDE passa essa informação através de um número, para interpretá-lo consulte a tabela 04.

Parâmetro D e L: estes dois parâmetros dependem da função que está sendo usada, isto é, dependem do parâmetro A, mas este assunto será detalhado mais adiante.

USANDO O PROGRAMA RAMEXPANDE

Agora, ensinaremos você a usar o programa RAMEXPANDE, para isso, siga atentamente as instruções e à

medida que for assimilando pratique o máximo possível. Então, vamos lá:

Primeiro: introduza o disco que contém o programa no Acionador de Discos, carregue-o digitando **PR#6** e digite **RUN RAMEXPANDE.LOAD CR**. Através da tela o computador lhe perguntará em qual conector encontra-se o Cartão. Responda digitando o número do conector da Placa Expansora de Periféricos em que você colocou o Cartão.

Segundo: em seguida você deve inicializar o programa. Essa inicialização é feita digitando a seguinte instrução:
E = 0: CM\$ = "14, 0, E": CALL 10: PRINT E CR.

Veja bem! Definimos os parâmetros da seguinte forma:
A = 14, porque de acordo com a tabela 03, 14 define a inicialização do RAMEXPAND;

B = 0, porque nesse caso não estamos usando segmento;

E = variável previamente definida como sendo 0 para garantir a sua existência.

Terceiro: finalmente, você deve dizer ao programa qual bloco da memória deseja usar. Para isso, construa a variável STRING como a seguir:

CM\$ "A, B, E, D, L"

A = 9, de acordo com a tabela 03.

B = 5 + 128 + 8 + 256 = 397

As parcelas desta soma representam:

5 = número do conector em que o Cartão está encaixado;

128 = use sempre este valor, pois ele se refere ao modelo do Cartão;

8 = número do primeiro bloco de 12K. Veja a tabela 01;

256 = número da primeira seção de 32K. Veja a figura 05 que apresenta esta numeração;

E = variável através da qual retorna o estado de erro.

Note que: neste caso não precisamos definir os parâmetros *D* e *L*. Logo, para selecionar o primeiro bloco de memória do Cartão, basta digitar: $CM\$ = "9, 397, E"$: CALL 10: PRINT E CR.

Observações:

- O comando CALL 10 que aparece na instrução tem o objetivo de chamar o programa RAMEXPANDE;
- O comando PRINT é usado para que o programa imprima o código de erro;
- Você pode selecionar até quatro blocos dentro de uma seção, para fazer isto basta redefinir o parâmetro *B*. Por exemplo, para selecionar o primeiro e segundo blocos de 12K, faríamos assim:

$$B = 5 + 128 + 8 + 16 + 256 = 413.$$

Veja que neste caso apenas adicionamos o número do segundo bloco, que é 16.

Importante: como o programa RAMEXPANDE ocupa os 4K do segundo banco, você não deve especificar esse bloco, que é o de número 64 da seção 256.

Veja agora as tabelas 03 e 04 que mostram o sumário das funções e o sumário de códigos de erros, respectivamente.

SUMÁRIO DAS FUNÇÕES

Parâmetro A	Função
0	<i>Nenhuma operação.</i>
1	<i>Armazenar programas completos na memória expandida.</i>
2	<i>Buscar e executar programas que se encontram na memória expandida.</i>
3	<i>Não implementada.</i>
4	<i>Não implementada.</i>
5	<i>Armazenar conteúdo de um ARRAY (números reais ou inteiros) na memória expandida.</i>
6	<i>Buscar o conteúdo de um ARRAY na memória expandida.</i>
7	<i>Trocar o conteúdo de um ARRAY entre a memória normal e a expandida.</i>
8	<i>Apagar segmentos da memória expandida para torná-los úteis.</i>
9	<i>Selecionar novos blocos da memória expandida.</i>
10	<i>Não implementada.</i>
11	<i>Conhecer a quantidade total de espaço livre na memória expandida.</i>
12	<i>Buscar o próximo segmento ocupado.</i>
13	<i>Não implementada.</i>
14	<i>Inicializar o RAMEXPANDE.</i>
15	<i>Guardar o conteúdo de um ARRAY STRING (Basic do micro-ENGENHO) na memória expandida.</i>
16	<i>Buscar o conteúdo do ARRAY STRING na memória expandida.</i>

Tabela 03

SUMÁRIO DE CÓDIGOS DE ERROS

Número da mensagem	Significado
0	<i>Êxito na operação.</i>
1	<i>Função ilegal.</i>
2	<i>Variável não encontrada.</i>
3	<i>Bloco de memória já em uso.</i>
4	<i>Bloco de memória não encontrado.</i>
5	<i>Parâmetro de seleção de bloco ilegal.</i>
6	<i>Bloco de memória não é RAM.</i>
7	<i>Segmento utilizado.</i>
8	<i>Segmento não encontrado.</i>
9	<i>Número de segmento ilegal.</i>
10	<i>Nenhum programa é especificado, o número de fim de linha precede o número de começo de linha.</i>
11	<i>ARRAY não especificado.</i>
12	<i>Não há espaço suficiente para guardar informação na memória expandida.</i>
13	<i>Não há espaço suficiente para carregar STRING na memória normal.</i>

Tabela 04

CUIDADOS A SEREM TOMADOS

Primeiro: o SOD permite que até 16 arquivos sejam usados ao mesmo tempo. Ao ser carregado, se nada for especificado, ele assume como sendo três o número de arquivos. Para alterar essa configuração você deve usar o comando MAXFILES. Os valores atribuídos a esse comando podem ser mudados, mesmo quando você já estiver executando o programa RAMEXPANDE. Para contar com esse recurso, antes de executar o programa RAMEXPANDE.LOAD, atribua ao MAXFILES o maior valor que você usará, assim poderá alterá-lo durante a execução do programa;

Observação: *se você não se recorda do comando MAXFILES, consulte o Manual do Sistema de Operação em Disco.*

Segundo: como você deve ter notado, ao carregar o programa RAMEXPANDE o ponteiro denominado HIMEM é posicionado, de forma a indicar ao usuário a memória que está disponível. Se você não se recorda, veja a figura 06 e note também, que ao carregar o programa aparece uma mensagem na tela indicando a nova posição do HIMEM. Desse modo, se você recarregar o programa, o HIMEM terá seu valor decrescido porque ele novamente será reposicionado. Para evitar que isso aconteça, sempre antes de carregar o programa RAMEXPAND, execute um comando MAXFILES com o número de arquivos que pretende usar com o SOD relocado. Lembre-se, o valor normal é três.

Terceiro: Existem dois comandos do SOD, FP e INT, que possibilitam o uso do Basic com ponto flutuante e do Basic inteiro, respectivamente. Não use esses comandos sem antes declarar o valor de HIMEM, sob o risco de perder informações contidas na memória principal.

DESCRIÇÃO DOS COMANDOS

Como dissemos anteriormente, antes de executar qualquer tarefa, você deve informar ao RAMEXPANDE que tarefa espera que ele execute. Essa informação é prestada através da construção de uma variável STRING. Para facilitar o trabalho, faremos agora uma descrição de cada uma dessas variáveis.

1 – Armazenar programas na memória expandida

Forma: CM\$ = "A, B, E"

A = 1, de acordo com a tabela 03;

B = número do segmento de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro.

Descrição:

Esta função guarda o programa em um segmento que não pode ser usado para outro fim. Após sua execução você poderá receber as seguintes mensagens como códigos de erros:

0 = êxito

1 = função ilegal

2 = variável não encontrada

7 = segmento já utilizado

9 = número de segmento ilegal

12 = espaço de memória não suficiente

2 – Buscar e executar programas que se encontram na memória expandida

Forma: CM\$ = "A, B, E"

A = 2, de acordo com a tabela 03;

B = número do segmento, de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

Descrição:

Esta função busca e executa o programa localizado no segmento especificado pelo parâmetro B. Após sua execução você pode receber as seguintes mensagens como códigos de erros:

0 = êxito

1 = função ilegal

2 = variável não encontrada

8 = segmento não encontrado

9 = número de segmento ilegal

13 = memória ocupada

5 – Armazenar ARRAY

Forma: CM\$ "A, B, E, D"

A = 5, de acordo com a tabela 03;

B = número do segmento, de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

D = nome do ARRAY

Descrição:

Esta função guarda o conteúdo de um ARRAY de números reais ou inteiros num segmento livre. O ARRAY deve ser especificado por um nome, não pode ser por número. Após sua execução você pode receber as seguintes mensagens como códigos de erros:

- 0 = êxito
- 1 = função ilegal
- 2 = variável não encontrada
- 7 = segmento já utilizado
- 9 = número de segmento ilegal
- 11 = nenhum ARRAY especificado
- 12 = espaço não suficiente na memória expandida

6 — Buscar o conteúdo de um ARRAY na memória expandida.

Forma: CM\$ "A, B, E, D, L"

- A = 6, de acordo com a tabela 03;
- B = número do segmento, de 1 a 32727;
- E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;
- D = nome do ARRAY;
- L = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o comprimento total do ARRAY.

Descrição:

Esta função recupera o conteúdo do ARRAY que se encontra no segmento especificado pelo parâmetro B. Os códigos de erros para esta função são apresentados a seguir:

- 0 = êxito
- 1 = função ilegal
- 2 = variável não encontrada
- 8 = segmento não encontrado
- 9 = número de segmento ilegal
- 11 = ARRAY não especificado

7 – Trocar o conteúdo de um ARRAY entre a memória normal e a expandida

Forma: CM\$ = "A, B, E, D, L"

- A = 7, de acordo com a tabela 03;
- B = número do segmento, entre 1 e 32727;
- E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;
- D = nome do ARRAY a ser colocado na memória expandida;
- L = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que é retornado o comprimento do ARRAY.

Descrição:

Esta função troca o conteúdo de um ARRAY na memória principal com o conteúdo de um ARRAY na memória expandida, desde que eles tenham o mesmo comprimento. Veja agora os códigos de erros para esta função.

-
- 0 = êxito
 - 1 = função ilegal
 - 2 = variável não encontrada
 - 8 = segmento não encontrado
 - 9 = número de segmento ilegal
 - 11 = ARRAY não especificado

8 — Apagar segmentos na memória expandida para torná-los úteis

Forma: CM\$ = "A, B, E"

- A = 8, de acordo com a tabela 03;
- B = número do segmento, de 1 a 32727;
- E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

Descrição:

Esta função apaga o conteúdo do segmento especificado pelo parâmetro B, tornando esse segmento livre para que possa ser ocupado. Para esta função os códigos de erros são os seguintes:

- 0 = êxito
- 1 = função ilegal
- 2 = variável não encontrada
- 8 = segmento não encontrado
- 9 = número de segmento ilegal

9 – Selecionar novos blocos da memória expandida

Forma: CM\$ = "A, B, E"

A = 9, de acordo com a tabela 03;

B = número do segmento, de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

Descrição:

Esta função permite selecionar novos blocos para serem unidos àqueles já em uso. Para este caso os códigos de erros são os seguintes:

0 = êxito

1 = função ilegal

2 = variável não encontrada

3 = bloco de memória já em uso

5 = parâmetro de seleção de bloco ilegal

6 = bloco de memória não é RAM

11 – Conhecer a quantidade total de espaço livre na memória expandida

Forma: CM\$ = "A, B, E, D, L"

A = 11, de acordo com a tabela 03;

B = 0, neste caso B não tem significado;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que será retornado o código de erro;

D = 0, neste caso D não tem significado;

L = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que é retornada a quantidade total de espaço livre na memória expandida.

Descrição:

Esta função permite conhecer a quantidade total de espaço livre na memória expandida. Os códigos de erros para este caso são:

0 = êxito

1 = função ilegal

2 = variável não encontrada

12 – Buscar o próximo segmento ocupado

Forma: CM\$ = "A, B, C"

A = 12, de acordo com a tabela 03;

B = variável que deve ser previamente definida como sendo 0 para garantir a sua existência;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que é retornado o número do próximo segmento, se estiver ocupado.

Descrição:

Esta função serve para revisar quais segmentos estão sendo utilizados. O parâmetro B deve ser uma variável para que nela retorne o número do próximo segmento. Se o próximo segmento não estiver ocupado o valor que retornará será 0. Os códigos de erros para este caso são.

- 0 = êxito
- 1 = função ilegal
- 2 = variável não encontrada
- 8 = segmento não encontrado
- 9 = número de segmento ilegal

14 – Inicializar o RAMEXPANDE

CM\$ = "A, B, E"

- A = 14, de acordo com a tabela 03;
- B = 0, neste caso B não tem significado;
- E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

Descrição:

Esta função inicializa o programa RAMEXPANDE. Todos os blocos são desconectados e qualquer informação é perdida. Os códigos de erros são os seguintes:

- 0 = êxito
- 1 = função ilegal.

15 – Guardar o conteúdo de um ARRAY STRING (Basic do microENGENHO) na memória expandida

Forma: CM\$ = "A, B, E, D"

A = 15, de acordo com a tabela 03;

B = número do segmento, de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0 para garantir a sua existência. É através dessa variável que retorna o código de erro;

D = nome do ARRAY STRING.

Descrição:

Esta função guarda o conteúdo de um ARRAY STRING (Basic do microENGENHO) em um segmento livre, o qual deve ser numerado convenientemente. O ARRAY deve ser especificado por um nome e não por um número. Neste caso os códigos de erros são:

0 = êxito

1 = função ilegal

2 = variável não encontrada

7 = segmento utilizado

9 = número de segmento ilegal

11 = ARRAY não especificado

12 = não há espaço suficiente para guardar informação na memória expandida.

16 – Buscar o conteúdo do ARRAY STRING na memória expandida

Forma: CM\$ = "A, B, E, D"

A = 16, de acordo com a tabela 03;

B = número do segmento, de 1 a 32727;

E = variável que deve ser previamente definida como sendo 0. É através dessa variável que retorna o código de erro;

D = nome do ARRAY.

Descrição:

Esta função recupera o conteúdo de um ARRAY STRING (Basic do microENGENHO) do segmento especificado pelo parâmetro B, trazendo-o para a memória principal. O ARRAY deve ser especificado por um nome e não por um número. Para esta função os códigos de erros são os seguintes:

- 0 = êxito
- 1 = função ilegal
- 2 = variável não encontrada
- 8 = segmento não encontrado
- 9 = número de segmento ilegal
- 11 = ARRAY não especificado
- 14 = memória ocupada

Importante:

Como mostrou o item anterior, cada função do programa RAMEXPAND tem alguns códigos de erros associados. Vamos agora descrever as características referentes aos tratamentos de erros:

- *O erro só será indicado se houver a variável CM\$;*
- *Se o parâmetro B não for devidamente especificado, nenhum código de erro retornará, embora a função seja executada;*
- *Se a variável erro não for previamente definida, não haverá indicação de erro mesmo que ele ocorra;*

-
- *Se a variável de erro for devidamente especificada, porém faltar a definição de algum outro parâmetro, haverá indicação de um erro;*
 - *Quando houver um erro, a execução de uma função depende tanto da situação como do erro. Na maioria das vezes, após a constatação de erro nenhuma ação ocorrerá.*

5. O PROGRAMA PSEUDO DISCO

AS CARACTERÍSTICAS DO PROGRAMA PSEUDO DISCO 49

O MECANISMO DO PROGRAMA PSEUDO DISCO 49

USANDO O PROGRAMA PSEUDO DISCO 50

CUIDADOS A SEREM TOMADOS 51

5. O PROGRAMA PSEUDO DISCO

Este programa possibilita o uso do Cartão para armazenamento de arquivos e informações como se ele fosse um disco. Evidentemente que as características do PSEUDO DISCO não se comparam aquelas apresentadas por um disco, porém, através de software apropriado o SOD pode interpretar o Cartão como se ele fosse um outro disco.

AS CARACTERÍSTICAS DO PROGRAMA PSEUDO DISCO

O PSEUDO DISCO opera tanto com o SOD normal quanto com o SOD relocado, ele também permite usar a memória de um ou mais Cartões para armazenamento. Basicamente, para usar o PSEUDO DISCO primeiramente você deve prepará-lo e em seguida instalá-lo, mas este procedimento será descrito mais adiante.

O MECANISMO DO PROGRAMA PSEUDO DISCO

O PSEUDO DISCO consiste de três programas: o PSEUDO DISCO, o PSEUDO NRM e o PSEUDO MVD. O PSEUDO DISCO é um programa em Basic que faz a ligação entre o usuário e os outros dois programas que estão escritos em linguagem de máquina. O PSEUDO NRM instala o PSEUDO DISCO e inicializa o sistema se o

SOD normal estiver sendo usado. O PSEUDO MVD instala o PSEUDO DISCO se o SOD relocado estiver sendo usado.

USANDO O PROGRAMA PSEUDO DISCO

Antes de usar o PSEUDO DISCO, como já foi dito anteriormente, você deve proceder sua preparação e a sua instalação, para isso siga as seguintes instruções:

- Preparando o PSEUDO DISCO

Primeiro: introduza o disco que contém o PSEUDO DISCO no Acionador de Discos e digite **PR#6 CR**;

Segundo: digite **RUN PSEUDO DISCO CR** para executar o programa;

Terceiro: nesse momento a tela deve estar apresentando um menu contendo cinco opções, escolha aquela que prepara o PSEUDO DISCO e que é a opção (2);

Quarto: novas três opções aparecerão na tela digite 1 para dizer ao programa em que conector o Cartão está instalado e para quantos K Bytes ele expande a memória. Além dessas perguntas, o programa também perguntará se você pretende usar todos os bancos do Cartão como PSEUDO DISCO, responda com **S** (Sim) e **N** (Não). Se você responder que não o programa perguntará um banco por vez, novamente responda com **S** (Sim) ou **N** (Não);

Quinto: repita a quarta instrução para todos os Cartões que pretende usar e depois digite 2 para alterar os parâmetros do disco;

Sexto: nesse ponto devem existir três novas opções na

tela, a opção 1 possibilita definir o número do conector (slot) e o número do drive para que mais tarde você possa se referir ao PSEUDO DISCO. Por exemplo, se você definisse como slot 4 e drive 1 teria de usar esses números para acessar o PSEUDO DISCO, como por exemplo: **CATALOG, S4, D1;**

Sétimo: com a opção 2, se você quiser, pode redefinir o comprimento do diretório, caso não defina ele terá 4 setores como comprimento;

Oitavo: como todos os parâmetros estão definidos, retorne ao menu principal digitando 3.

- Instalando o PSEUDO DISCO

Após efetuar a correta preparação do programa você deve efetuar a sua instalação no sistema, o que possibilitará seu uso. Para efetuar a instalação selecione a opção 3 do menu principal. Feito isto, o próprio sistema carrega o programa apropriado, PSEUDO NRM ou PSEUDO MVD, dependendo do SOD relocado estar ou não presente.

Para ter acesso ao PSEUDO DISCO proceda da mesma forma que para um disco real, ou seja, usando as designações de conector (slot) e drive que você definiu no item sexto.

CUIDADOS A SEREM TOMADOS

- Devido a organização do PSEUDO DISCO o programa FID, que possibilita a transferência de arquivos do ou para o PSEUDO DISCO, deve ser alterado para possibilitar a transferência de um disco real para o PSEUDO DISCO. Esta alteração é feita usando o programa MOD exatamente como se encontra descrito na página 24 deste Manual;

-
- Você deve tomar cuidado na escolha dos bancos a serem usados pelo PSEUDO DISCO, principalmente quando o MOVESOD estiver sendo usado e, conseqüentemente estiver ocupando o banco 2;
 - Quando um disco for inicializado com o PSEUDO DISCO já instalado, uma cópia de conexão ao SOD será escrita nele. Desse modo, este disco não mais poderá ser carregado apropriadamente. Para evitar este problema, aconselhamos você a inicializar os discos antes de instalar o PSEUDO DISCO. As informações permanecem no PSEUDO DISCO enquanto o computador estiver alimentado pela energia elétrica ou até que elas sejam apagadas com o comando DELETE;
 - Você só precisa executar a preparação do programa PSEUDO DISCO quando quiser efetuar uma mudança em sua configuração, caso contrário basta executar sua instalação.

6. INFORMAÇÕES TÉCNICAS

ENDEREÇAMENTO DA MEMÓRIA DO CARTÃO 55

CONTROLANDO A EXPANSÃO 57

USO DE MEMÓRIA 60

EXECUTANDO O PROGRAMA DE TESTE DE MEMÓRIA 60

6. INFORMAÇÕES TÉCNICAS

ENDEREÇAMENTO DA MEMÓRIA DO CARTÃO

Sob muitos aspectos, o Cartão que expande a memória para 128K pode ser entendido como quatro cartões de 16K cada, e o Cartão que expande a memória para 192K como oito cartões também de 16K cada, com a grande vantagem de ocuparem um único conector da Placa Expansora de Periféricos.

Esses bancos de 16K compartilham o mesmo espaço de memória ocupado pelas EPROMS da Placa de Circuito Principal do microGENHNO, assim, em um determinado instante ou as EPROMS são acessadas ou um banco de 16K é acessado. Aqui surge a necessidade de uma explicação: como o espaço de memória utilizado pelas EPROMS tem apenas 12K de comprimento, aparentemente, os bancos de 16K não poderiam ai ser colocados, uma vez que não haveria espaço para eles. Diante disso, cada banco de 16K foi subdividido em três blocos, um de 8K, que ocupa o espaço de \$E000 a \$FFFF e dois blocos de 4K que compartilham o espaço de \$D000 a \$DFFF. Veja a figura 07.

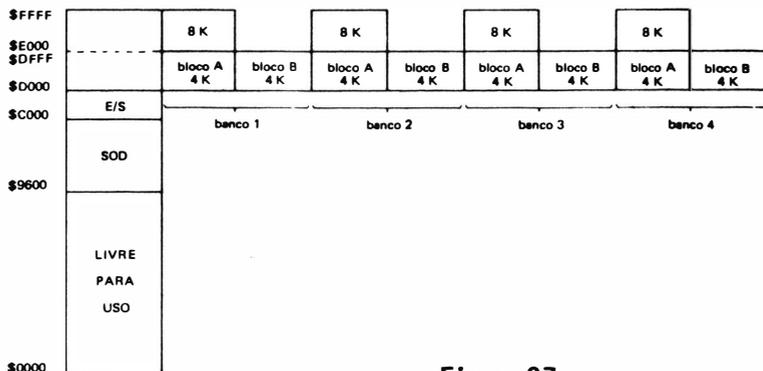


Figura 07

Observação: esta figura se refere ao Cartão que expande a memória para 128K, para o Cartão que expande a memória para 192K, basta aumentarmos quatro bancos.

A partir dessa forma de subdividir cada banco adotamos a seguinte convenção:

- Os bancos são denotados por banco 1, banco 2, banco 3 e banco 4, no caso do Cartão que expande a memória para 128K e por banco 1, banco 2, banco 3, banco 4, banco 5, banco 6, banco 7 e banco 8 para o Cartão que expande a memória para 192K;
- Para especificarmos os dois blocos de 4K, utilizamos as letras A e B. Desta forma, a expressão BANCO 1A se refere ao bloco A de 4K do Banco 1, BANCO 1B se refere ao bloco B de 4K do Banco 1, e assim por diante.

CONTROLANDO A EXPANSÃO

O Cartão é controlado usando dezesseis posições de memória, que estão no espaço de \$CONO a \$CONF (onde N = 8 + número do conector onde o Cartão está instalado). Desse modo, há a possibilidade de controlarmos as seguintes funções:

- Seleção da RAM da memória expandida ou da EPROM da memória principal;
- Seleção do banco de 16K a ser acessado;
- Seleção do bloco (A ou B) dentro do banco a ser usado;
- Habilitação da RAM para escrita ou proteção da RAM contra escrita.

Atenção: *para colocar a memória expandida em um estado, basta acessar o endereço de controle que determina este estado. No Basic as funções PEEK e POKE fazem o acesso, porém se estiver operando em linguagem de máquina, use as funções LDA e STA que correspondem a CARREGAR e GUARDAR, respectivamente.*

A tabela 05 descreve os endereços e suas respectivas funções.

Endereço de controle	Função
\$CON0	<i>Lê RAM do bloco A e a protege contra escrita.</i>
\$CON1	<i>Lê EPROM e habilita a escrita na RAM do bloco A.</i>
* \$CON2	<i>Lê EPROM e protege a RAM do bloco A contra escrita.</i>
\$CON3	<i>Lê RAM do bloco A e a habilita para escrita.</i>
\$CON8	<i>Lê RAM do bloco A e a protege contra escrita.</i>
\$CON9	<i>Lê EPROM e habilita a escrita na RAM do bloco B.</i>
* \$CONA	<i>Lê EPROM e protege a RAM do bloco B contra escrita.</i>
\$CONB	<i>Lê RAM do bloco B e a habilita para escrita.</i>
\$CON4	<i>Seleciona o banco 1.</i>
\$CON5	<i>Seleciona o banco 2.</i>
\$CON6	<i>Seleciona o banco 3.</i>
\$CON7	<i>Seleciona o banco 4.</i>
\$CONC	<i>Seleciona o banco 5.</i>
\$COND	<i>Seleciona o banco 6.</i>
\$CONE	<i>Seleciona o banco 7.</i>
\$CONF	<i>Seleciona o banco 8.</i>

Tabela 05

(*) *Esta Combinação (lê EPROM e RAM protegida contra escrita) desabilita a memória expandida.*

Observações:

- *Quando está ocorrendo leitura na EPROM, a RAM da expansão é desabilitada para leitura e a EPROM da memória principal é habilitada;*
- *Quando está ocorrendo leitura na RAM, a RAM da expansão é habilitada e conseqüentemente a EPROM da memória principal é desabilitada;*
- *Nos endereços da tabela anterior $N = 8 +$ número do conector em que o Cartão está instalado.*

Os quatro bits do endereço de controle escolhem o modo de operação do Cartão, assim como determinam qual o banco de 16K será acessado.

Note que:

- *Quando a linha 2 de endereço é “baixa” ($A_2 = 0$) a seleção de estado é habilitada e neste caso, as linhas 0, 1 e 3 determinam estado da seguinte forma:*
 - a. *A linha 0 de endereço determina se o Cartão será habilitado para escrita ou protegido contra escrita ($A_0 = 1$ para habilitar escrita e $A_0 = 0$ para proteger contra escrita).*
 - b. *A linha 3 de endereço determina qual bloco de 4K dentro do banco de 16K será acessado ($A_3 = 0$ para o bloco A e $A_3 = 1$ para o bloco B).*
- *Quando a linha de endereço é “alta” ($A_2 = 1$) a seleção de bancos de 16K habilitada e as linhas 0, 1 e 3 de endereço determinam qual banco de 16K é escolhido.*

Nos modos CON1 e CON9 as EPROMS da memória principal são acessadas durante uma leitura de \$D000 a \$FFFF enquanto a RAM na memória expandida é gra-

vada durante uma operação de escrita. Isto possibilita escrever na RAM enquanto se executa o Basic na EPROM que ocupa o mesmo espaço de endereço. Isto é possível desde que o Basic esteja sendo lido pela EPROM enquanto o programa deve estar escrevendo na RAM através de uma instrução POKE.

Para que a escrita na RAM da memória expandida seja habilitada, o endereço de controle do modo desejado deve ser acessado duas vezes, isto é imprescindível para qualquer dos modos de habilitar escrita. Se o Cartão já está no estado de escrita habilitada (\$CON1), então basta um único acesso ao endereço de controle. Por outro lado, se o Cartão está no estado protegido contra escrita (CONO), são necessários dois acessos para o endereço de controle para que a escrita seja habilitada.

USO DE MEMÓRIA

- **SOD RELOCADO**

O SOD relocado ocupa o bloco de 8K e o bloco A de 4K do banco 2, utilizando todos os 12K de memória, de \$D000 a \$FFFF para o SOD, rotinas do monitor e buffers de arquivos associados. Este banco é habilitado usando o endereço de controle \$CON3 e o endereço de seleção do banco \$CON5. O bloco 8 do banco 2 não é usado pelo SOD relocado.

- **Programas de teste de memória**

No disco que você recebeu também existe um programa de teste da memória do Cartão. Com esse programa você testa a confiabilidade do sistema.

EXECUTANDO O PROGRAMA DE TESTE DE MEMÓRIA

Primeiro: introduza o disco que contém o teste no Acionador de Discos e digite: **PR# 6 CR;**

Segundo: se o Cartão for o que expande a memória para 128K DIGITE: **RUN TESTE DE MEMÓRIA-128K CR** e se o Cartão for aquele que expande a memória para 192K digite: **RUN TESTE DE MEMÓRIA-192K CR**;

Terceiro: o programa perguntará pelo número do conector (slot). Responda com o número do conector da Placa Expansora de Periféricos onde o Cartão está instalado;

Quarto: se você desejar um completo teste de memória, incluindo testes de refrescamento, após o número do conector digite F. Por exemplo se o Cartão está instalado no conector 5, digite: **5F CR**.

Quinto: o teste será executado por vários minutos e se tudo estiver em ordem será novamente solicitado o número do conector;

A seguir apresentamos a tabela 06 que mostra uma relação dos possíveis erros no teste de memória.

Erro	Significado
<i>A</i>	<i>ROM não habilitada.</i>
<i>B</i>	<i>Usados dois POKES para habilitar a ROM.</i>
<i>C</i>	<i>Escrita na ROM.</i>
<i>D</i>	<i>Escrita na RAM habilitada após um POKE.</i>
<i>E</i>	<i>Escrita na RAM não habilitada após dois POKES.</i>
<i>F</i>	<i>Proteção contra escrita permitiu escrita após um POKE.</i>
<i>G</i>	<i>Proteção contra escrita permitiu escrita após dois POKES.</i>
<i>H</i>	<i>Leitura na ROM/escrita na RAM habilitada após um POKE.</i>
<i>I</i>	<i>Leitura na ROM/escrita na RAM habilitada após dois POKES.</i>
<i>J</i>	<i>ROM não habilitada.</i>

Tabela 06

7. ESPECIFICAÇÕES

MEMÓRIA 65

CONSUMO MÉDIO DE ENERGIA 65

7. ESPECIFICAÇÕES

MEMÓRIA

Tipo: dinâmica
Tempo de acesso: 200ns
Refrescamento: 7 bits

CONSUMO MÉDIO DE ENERGIA

+ 5V 160 mA

.....

.....

SPECTRUM EQUIPAMENTOS ELETRÔNICOS IND. E COM. LTDA.

Rua Félix Guilhem, 913 ● CEP 05069 ● São Paulo ● SP ● Fone: 260-0488

CGC 51.235.703/0001-30 - INDÚSTRIA BRASILEIRA