

EXPANSÃO DE MEMÓRIA 64 K/128 K

LODATA

LOGODATA

INDÚSTRIA E COMÉRCIO DE EQUIPAMENTOS ELETRÔNICOS LTDA.
RUA CAMACAN, 424 - VILA ANASTÁCIO - SÃO PAULO - SP
CEP 05095 - TELEFONE (011) 832-1281
CGC 51.160.471/0001-06
I.E. 110.362.573

EXPANSÃO DE MEMÓRIA 64/128K

MANUAL DE OPERAÇÕES

PARTE NÚMERO 8001A0M128K

ÍNDICE GERAL

CAPÍTULO 1: *INTRODUÇÃO*

CAPÍTULO 2: *INSTRUÇÕES DE INSTALAÇÃO*

- 2.1 - Conector recomendado
- 2.2 - Instalação
- 2.3 - Instalação de placas de expansão adicionais

CAPÍTULO 3: *UTILIZAÇÃO DA EXPANSÃO DE 64K E 128K*

- 3.1 - Uso com Applesoft e BASIC inteiro
 - 3.1.1 - Apple II
 - 3.1.2 - Apple II plus e UNITRON
- 3.2 - Operação com DOS 3.3
- 3.3 - Tratamento do Cartão de linguagem ou exp. 16K
- 3.4 - Resenha do software
 - 3.4.1 - MOVEDOS
 - 3.4.2 - RAMEXPAND
 - 3.4.3 - DOS PSEUDO-DISK
 - 3.4.4 - PASCAL PSEUDO-DISK
 - 3.4.5 - CP/M PSEUDO-DISK

CAPÍTULO 4: *MOVEDOS - SOFTWARE DE RELOCAÇÃO DO DOS*

- 4.1 - Propriedades do DOS relocado
- 4.2 - Utilização de MOVEDOS
 - 4.2.1 - Rodando com DOS 3.2
 - 4.2.2 - Rodando com DOS 3.3
- 4.3 - Utilização do cartão de linguagem

- 4.4 - Compatibilidade do DOS relocado
 - 4.4.1 - Modificação do FID e MUFFIN
 - 4.4.2 - Modificação do BOOT13
- 4.5 - Informações técnicas e detalhes de software
 - 4.5.1 - Operação do DOS relocado
 - 4.5.2 - Utilização da memória
 - 4.5.3 - Detalhes adicionais sobre MOVEDOS
 - 4.5.4 - Rotinas diversas, apontadores, etc.

CAPÍTULO 6 : *RAMEXPAND*

- 5.1 - Características do *RAMEXPAND*
 - 5.1.1 - Terminologia
 - 5.1.2 - Requisitos e utilização de memória
 - 5.1.3 - Estrutura de dados
 - 5.1.4 - Alocação de memória
 - 5.1.4.1 - Placas de 16K e 32K
 - 5.1.4.2 - Placas de 64K e 128K
 - 5.1.5 - Pré-requisitos caso o Cartão de Linguagem esteja presente
- 5.2 - Utilização de *RAMEXPAND*
 - 5.2.1 - Instalação
 - 5.2.2 - Implantação dos comandos
 - 5.2.3 - Chamada de *RAMEXPAND*
 - 5.2.4 - Inicialização de *RAMEXPAND*
 - 5.2.5 - Acrescentando blocos de memória
 - 5.2.5.1 - Placas de 16K e 32K
 - 5.2.5.2 - Placas de 64K e 128K
 - 5.2.6 - Reserva de espaço para o BASIC
 - 5.2.7 - Utilização do cartão de firmware
- 5.3 - Precauções
 - 5.3.1 - Proteção de *RAMEXPAND*
 - 5.3.2 - Ajuste de MAXFILES
 - 5.3.3 - Recarga de *RAMEXPAND*
 - 5.3.4 - Troca de versões de BASIC
 - 5.3.5 - Reposição dos pontos de entrada de *RAMEXPAND*
 - 5.3.6 - Declaração de variáveis
- 5.4 - Manipulação de erros
 - 5.4.1 - Descrição
 - 5.4.2 - Sumário dos erros

- 5.5 - Limitações presentes
- 5.6 - Programa exemplo
 - 5.6.1 - Rodando o programa
 - 5.6.2 - Operação do programa
- 5.7 - Sumário das funções disponíveis
- 5.8 - Sintaxe e descrição dos comandos

CAPÍTULO 6 : PSEUDO DISCO PARA O DOS

- 6.1 - Requisitos e características
- 6.2 - Instalação do Pseudo Disco
- 6.3 - Uso do Pseudo Disco
- 6.4 - Reconexão após "Bootstrap"
- 6.5 - Precauções
- 6.6 - Uso do Pseudo Disco
- 6.7 - Como funciona o Pseudo Disco

CAPÍTULO 7 : PSEUDO DISCO PARA PASCAL

CAPÍTULO 8 : PSEUDO DISCO PARA PC/M

- 8.1 - Ajuste do Pseudo Disco
- 8.2 - Implantação do Pseudo Disco
- 8.3 - Utilização do Pseudo Disco
- 8.4 - Requisitos e limitações

CAPÍTULO 9 : INFORMAÇÕES TÉCNICAS

9.1 - Endereçamento das placas de 64K e 128K

9.2 - Controle das placas de 64K e 128K

9.3 - Utilização da placa de RAM

9.3.1 - DOS relocado

9.3.2 - BASIC alternativo

9.3.3 - Indicadores LED

9.3.4 - Programas de diagnóstico

9.4 - Especificações

9.4.1 - Memória

9.4.2 - Consumo

9.4.3 - Partes compatíveis

CAPÍTULO I – INTRODUÇÃO

As placas LOGODATA 64/128K são compatíveis com o software projetado para utilizar o cartão de linguagem (PASCAL, FORTRAN, LISA versão II, etc.), bem como o cartão Z80 LOGODATA (PC/M, COBOL 80, FORTRAN 80, etc.).

Esta compatibilidade é assegurada por uma organização em bancos de 16Kbyte (4 bancos em 64K e 8 em 128K) acessados um por vez.

O primeiro desses bancos de 16K é controlado pelo software, exatamente da mesma maneira como o são os Cartões de Linguagem originais. Como resultado, não é necessário modificar o software existente.

Com a utilização do DOS 3.3, a versão de BASIC não residente na máquina será carregada automaticamente no primeiro banco de 16K, durante o procedimento de "bootstrap". As instruções INT e FP permitirão intercambiar as versões.

O software fornecido com as placas LOGODATA 64/128K tira proveito da capacidade adicional de memória, permitindo:

- 1.1 - Relocar o sistema operacional, liberando aproximadamente 10Kbytes de memória a ser utilizada para execução de programas no espaço de endereçamento real (0 a 64K).
- 1.2 - Emular um ou mais acionadores de discos flexíveis sob controle de DOS, PASCAL ou PC/M.
- 1.3 - Armazenar matrizes e sub-rotinas sob BASIC inteiro ou APPLESOFT.

CAPÍTULO 2 – INSTRUÇÕES DE INSTALAÇÃO

2.1 - CONECTOR RECOMENDADO

A posição recomendada para instalação dos cartões LOGODATA 64/128K depende da presença ou não do Cartão de Linguagem (ou expansão de 16K).

2.1.1 - Se o citado Cartão de Linguagem estiver presente no conector 0, instale os cartões adicionais em qualquer dos conectores disponíveis.

2.1.2 - Se no conector 0 estiver presente um cartão com linguagem em "Firmware", remova-o do sistema, ou instale-o em qualquer outro conector, instalando no conector 0 a expansão de memória LOGODATA 64/128K.

2.1.3 - Na ausência de qualquer cartão no conector 0, instale a expansão no mesmo.

2.2 - INSTALAÇÃO

2.2.1 - Desligue a chave de alimentação (verifique se o piloto está apagado).

2.2.2 - Remova a tampa do gabinete.

2.2.3 - Selecione um dos conectores que se encontram na parte superior da placa principal, de acordo com os critérios estabelecidos no item 2.1 acima. As designações dos conectores estão estampadas ao lado dos mesmos na placa de circuito impresso e vão de 0 a 7 da esquerda para a direita.

2.2.4 - Antes de tocar qualquer componente na placa de seu computador, descarregue cargas estáticas tocando a carcaça da fonte de alimentação (a caixa metálica longa à sua esquerda, olhando o computador de frente para o teclado).

2.2.5 - Remova a placa de expansão LOGODATA 64/128K de sua embalagem. A placa é acondicionada em saco plástico com tratamento anti-estático. Ela deverá permanecer neste saco, de cor preta, sempre que não estiver sendo usada, ou quando for enviada para eventuais reparos. A NÃO observância deste cuidado invalida a garantia da mesma. Antes de tocar a placa proceda como no item 2.2.4 acima.

2.2.6 - Insira a placa no conector escolhido com a parte mais longa apontando para você. Lembre-se que a força de inserção depende do número de contatos do conector que estão sendo abertos; portanto insira a placa ligeiramente inclinada (parte mais próxima do teclado mais alta) e pressione gentilmente até que a mesma esteja completamente horizontal e encaixada no conector. NÃO ligue o equipamento antes de certificar-se de que a placa está devidamente inserida. Danos à placa e ao equipamento podem resultar da inobservância desta recomendação, que não estão cobertos pela garantia.

2.2.7 - Ao recolocar a tampa do gabinete, certifique-se de não tocar a parte protuberante da placa, para evitar que a mesma seja deslocada da posição em que foi instalada.

2.3 - Os mesmos cuidados acima devem ser observados na instalação de outras placas adicionais de expansão.

CAPÍTULO 3 – UTILIZAÇÃO DAS PLACAS LOGODATA 64/128K

Os procedimentos para utilização das placas LOGODATA 64/128K descritos neste capítulo darão ênfase especial à sua operação sob controle dos BASICS inteiro e Applesoft, com o sistema operacional DOS 3.3.

Na maioria dos casos a operação das placas de expansão é transparente ao usuário, isto é, com a placa no conector 0 o software reconhece a presença da mesma e o procedimento é automático.

A maior parte do software está hoje disponível no formato de 16 setores por trilha (DOS 3.3) como é o caso de PASCAL, FORTRAN, e o sistema operacional PC/M 44 ou PC/M 56.

Recomenda-se portanto, enfaticamente, a mudança para este formato.

Listamos abaixo alguns casos de software que reconhece automaticamente os primeiros 16K bytes das expansões 64/128K:

VISICALC	PC/M 60
PASCAL	PC/M 56
FORTRAN	FORTRAN 80
BASIC INTEIRO	COBOL 80
APPLESOFT	ETC.

3.1 - EMPREGO COM APPLESOFT E BASIC INTEIRO

Uma das propriedades das placas LOGODATA de 64 e 128K é a de permitir a disponibilidade de ter-se na expansão a versão de BASIC alternativa à residente na máquina. O acesso a esta versão é feito de modo similar àquele feito ao Cartão de Linguagem (APPLE * FIRMWARE CARD).

3.1.1 - APPLE II

Nesta versão o BASIC residente é o inteiro. A versão APPLESOFT pode ser carregada no primeiro banco de memória da expansão. Quando necessário, o acesso é feito pela instrução FP no modo imediato.

3.1.2 - APPLE II PLUS (UNITRON AP II)

Da mesma maneira, nesta máquina a versão residente é o APPLESOFT e o acesso ao BASIC alternativo é feito via instrução INT.

Em ambos os casos, é indispensável que a expansão LOGODATA 64/128K esteja instalada no conector 0.

O sistema operacional DOS 3.3 não consegue reconhecer ou fazer acesso ao BASIC alternativo em conectores diferentes do 0, entretanto, o utilitário MOVEDOS fornecido com a placa de expansão pode fazê-lo (veja cap. 4).

3.2 - OPERAÇÃO COM DOS 3.3

O programa HELLO das cópias de distribuição do DOS 3.3 testa o conector 0 quanto à presença de memória, e, em caso positivo, carrega a linguagem alternativa nessa área.

Esse procedimento é repetido sempre que o DOS 3.3 é carregado na máquina, desde que a versão alternativa e o programa HELLO estejam presentes no disco.

Tal utilização automática, entretanto, não é muito conveniente, visto que ativa somente o primeiro bloco de 16K bytes da placa LOGODATA 64/128K.

O software de aplicação fornecido com a placa é descrito nas páginas seguintes e permite utilização extensiva da memória adicional, em uma ou mais placas presentes concomitantemente no sistema; com efeito, qualquer combinação de placas que não exceda 256K bytes pode ser utilizada.

3.3 - TRATAMENTO ESPECIAL DO CARTÃO DE LINGUAGEM

O cartão de linguagem, se presente, merece consideração especial. A razão é que o mesmo contém uma memória ROM, que está habilitada sempre que a memória RAM no cartão não o esteja, razão pela qual ocorre conflito com outras memórias presentes no sistema. Este conflito é contornado no pacote de software fornecido, entretanto paga-se uma penalidade na forma de acesso proibido a alguns blocos de memória.

O que resulta do exposto acima é que para cada banco de memória de 16K bytes, um bloco de 2K, especificamente aquele que ocuparia o espaço de endereçamento do ROM Autostart no cartão de linguagem (endereços \$F800 até \$FFFF) fica inacessível, resultando na perda de 16K bytes na placa de expansão de 64K, e 32K bytes na placa de 128K.

Recomendamos portanto não utilizar o cartão de linguagem com nossas expansões, visto que não resulta em benefício palpável.

3.4 - VISÃO GERAL DO SOFTWARE

É a seguinte a descrição geral do software fornecido com os cartões de expansão LOGODATA 64/128K. O mesmo estará presente (de modo evidente ou não) nos programas mais divulgados, projetados para reconhecer a presença de memória adicional.

3.4.1 - MOVEDOS (CAP. 4)

Programa que reloca o sistema operacional DOS 3.3 no segundo banco de 16K bytes das expansões, liberando aproximadamente 10K bytes para memória de programas no espaço de endereçamento original da máquina. O BASIC alternativo também pode ser carregado no mesmo cartão. Este programa é extremamente útil quando a memória disponível para uma aplicação é limitante.

3.4.2 - RAMEXPAND (CAP. 5)

Este pacote consiste em um conjunto de sub-rotinas utilitárias, acessíveis via APPLESOFT ou BASIC inteiro, que permitem rápidas transferências de dados do espaço de endereçamento normal para o da placa de expansão e vice-versa.

Permite a transferência de vetores, matrizes, sub-rotinas e programas, mantendo o acesso a estas informações, e opera com o DOS normal, bem como com o relocado.

3.4.3 - DISCO VIRTUAL (CAP. 6)

A função deste pacote é a de permitir a utilização do cartão de expansão de memória LOGODATA 64/128K como um acionador de disco virtual, emulando sua contrapartida mecânica.

Com efeito, uma ou mais placas de expansão poderão ser identificadas como acionadores de discos magnéticos, após um breve ritual de inicialização, que as habilita a serem manipuladas por programas da mesma maneira que estes manipulam acionadores de disco. Opera com DOS normal ou relocado. O resultado é um aumento dramático na velocidade de acesso a disco, bem como redução do desgaste dos acionadores presentes no sistema, se os acessos mais freqüentes forem transferidos para o acionador virtual.

3.4.5 - DISCO VIRTUAL PARA PASCAL

Mesma função descrita acima, porém sob Sistema Operacional do PASCAL.

3.4.6 - DISCO VIRTUAL PARA PC/M

Idem, sob controle do Sistema Operacional PC/M.

CAPÍTULO 4 – MOVEDOS - SOFTWARE DE RELOCAÇÃO

Descrevemos neste capítulo o software que permite tirar partido das expansões presentes no sistema para liberar uma área de cerca de 10K bytes no espaço de endereçamento do processador pelo artifício de mover o sistema operacional para a expansão.

Num sistema de 48K bytes, o sistema operacional DOS ocupa o topo dos mesmos 48K, residindo em 10.5K bytes (de \$9600 até \$BFFF). O programa que descrevemos move o DOS para a placa de expansão LOGODATA 64/128K, introduzindo as modificações necessárias no sistema, de modo a permitir a comunicação entre o sistema operacional relocado e o espaço real de endereçamento.

Essencialmente, o sistema operacional é deslocado para uma área desativada da memória, deixando de ocupar espaço útil desnecessariamente. Quando requerido, os eventos ocorrem como segue:

- a) O banco da placa LOGODATA 64/128K que contém o sistema operacional é habilitado. Como resultado o BASIC residente é desabilitado, visto que divide o espaço de endereçamento \$D000 – \$FFFF com o banco acima mencionado. Isto não causa problemas pois o BASIC não é necessário para operação do DOS.
- b) O comando é executado pelo DOS, agora residente no sistema.
- c) O banco que contém o DOS é desativado novamente, reabilitando o BASIC, e o controle é restituído a este último.

Assim, do ponto de vista do usuário, 105K bytes foram liberados para a aplicação. Além do mais, ambas as versões de BASIC continuam disponíveis na placa de expansão, bem como o sistema operacional relocado.

O resto da memória de expansão disponível pode ser utilizada como disco rápido, melhorando em muito o desempenho do sistema.

4.1 - PROPRIEDADES DO SISTEMA OPERACIONAL RELOCADO

O sistema operacional relocado retém todas as propriedades do original, incluindo a capacidade de inicializar disquetes. Quando um disco é inicializado através do sistema operacional relocado, o mesmo sistema relocado é transferido, resultando num disco que, quando carregado, moverá automaticamente o sistema operacional para a expansão.

Assim, o programa que efetua a relocação só precisa ser executado uma vez.

O programa foi projetado para operar com as placas LOGODATA 64/128K em um conector diferente do "zero", e relocar apropriadamente o sistema operacional nesta posição.

A busca dos cartões é efetuada a partir do conector zero procedendo para cima, até que uma placa 64/128K seja encontrada.

O conflito de endereços que ocorre em presença do cartão de linguagem é evitado pela versão MOVEDOS.LANG. Esta versão evita a ocupação dos 2K bytes superiores.

4.2 - UTILIZAÇÃO DE MOVEDOS

O software que acompanha a placa 64/128K é distribuído num disco de 16 setores e pode ser utilizado diretamente com o DOS 3.3.

Recomenda-se que uma cópia seja efetuada imediatamente antes da sua utilização. Quando for utilizado o software com o DOS 3.2 sugerimos que os programas sejam modificados para o formato de 16 setores por trilha, através, por exemplo, do programa DEMUFFIN das cópias de distribuição do sistema operacional, antes da sua utilização.

Conforme já foi dito, não é necessário que a placa LOGODATA 64/128K seja instalada no conector zero; entretanto recomenda-se esta posição para permitir que a mesma seja localizada por outros programas que assumam sua existência nesta posição, tais como PASCAL, VISICALC, etc.

4.2.1 - UTILIZAÇÃO COM DOS 3.2

- a) Carregue um DOS 3.2 comum.
- b) Execute o programa DEMUFFIN ou similar para converter MOVEDOS para 13 setores/trilha.
- c) Execute MOVEDOS digitando BRUN MOVEDOS. O sistema responderá com a mensagem que indica que o carregamento do sistema operacional em RAM está sendo efetuado, a recolocação é efetuada, e o controle retorna ao BASIC.
- d) Um disco virgem pode ser inicializado, resultando numa cópia que posteriormente relocará automaticamente. Etiquetas 128K LOGODATA foram incluídas no presente manual para identificação de seus programas que contenham DOS relocado. Recomendamos rotular os disquetes para evitar confusão com aqueles que contenham DOS normal.
- e) Caso seja conveniente, o BASIC alternativo poderá ser carregado no primeiro banco de 16K da placa LOGODATA 64/128K conforme descrito anteriormente, se desejado.

4.2.2 - UTILIZAÇÃO COM DOS 3.3

- a) Carregue um DOS 3.3 comum.
- b) Insira um disco de 16 setores contendo uma cópia do programa MOVE-DOS ou próprio original de distribuição (CUIDADO: não diga que eu não avisei!), e digita BRUN MOVEDOS. O sistema responderá com a mensagem que indica que o carregamento do sistema operacional em RAM está sendo efetuado, a relocação é efetuada, e o controle retorna ao BASIC.
- c) Um disco virgem pode ser inicializado, resultando numa cópia que posteriormente relocará automaticamente. Etiquetas 128K LOGODATA foram incluídas no presente manual para identificação de seus programas que contenham DOS relocado. Recomendamos rotular os disquetes para evitar confusão com aqueles que contenham DOS normal.

- d) Caso seja conveniente, o BASIC alternativo poderá ser carregado no primeiro banco de 16K da placa LOGODATA 64/128K conforme descrito anteriormente, se desejado.

4.2.3 - UTILIZAÇÃO COM CARTÃO DE LINGUAGEM

Caso no sistema esteja presente um cartão de linguagem, utilize a versão modificada `MOVEDOS.LANG` no disquete de distribuição, em lugar de `MOVEDOS`.

4.3 - UTILIZAÇÃO COM CARTÃO DE LINGUAGEM EM OUTRO CONECTOR

O DOS relocado, por omissão, é inicializado com a busca do BASIC alternativo no conector que contém a placa de expansão.

O primeiro banco de 16K conterá o BASIC alternativo e o segundo banco de 16K conterá o sistema operacional relocado.

Estas condições podem ser alteradas, se desejado, pelo emprego dos utilitários fornecidos, de tal forma que um conector diferente seja utilizado para o BASIC. O cartão de linguagem poderá ser colocado neste conector.

Damos a seguir o procedimento passo a passo para proceder a essas modificações:

- a) Carregue um DOS (3.2 ou 3.3), normal e execute `MOVEDOS`; ou
Carregue um DOS relocado de um disco previamente inicializado.
- b) Execute o programa de configuração apropriado (`AJUSTECON.INTEIRO` ou `AJUSTECON.APPLESOFT`).
- c) Digite o número do conector que contém o cartão de linguagem quando solicitado pelo programa. O sistema operacional relocado será modificado para reconhecer a presença do cartão.
- d) Esta nova configuração poderá ser congelada, inicializando-se um disquete com este sistema operacional, para uso futuro.

NOTA: O programa `HELLO` usado com este DOS modificado deverá ser alterado para não testar a presença do cartão de linguagem no conector 0, e não carregar o BASIC alternativo.

4.4 - PRECAUÇÕES NA INICIALIZAÇÃO DO SISTEMA

Aplicam-se nos casos descritos acima. (Reconfiguração)

Recomenda-se que sempre que uma reinicialização seja necessária (via PR#6, etc.), a mesma seja executada a partir do BASIC residente no sistema (`APPLESOFT` nas versões `APII plus`, `INTEIRO` nas versões `APII`), para evitar que o cartão de linguagem permaneça ativado.

A combinação Cartão de linguagem em conector diferente do zero E sistema operacional relocado é proibida. Se não for tomada a precaução de retornar ao BASIC em ROM da placa principal, neste caso, o cartão de linguagem permanecerá ativado após o "Bootstrap", e o sistema permanecerá travado sem condições de recuperação pelo DOS.

CAPÍTULO 5 – RAMEXPAND

RAMEXPAND é um pacote que consiste numa série de funções cujo propósito é estender a memória disponível para programas em qualquer dos BASICs do sistema.

Informação, na forma de dados, programas ou sub-rotinas, poderá ser armazenada em uma ou mais placas de expansão LOGODATA 64/128K, bem como em qualquer placa de 16K presente no sistema, para utilização quando necessário.

Este pacote permite armazenar programas completos que podem, no devido tempo, serem carregados e executados.

Permite também sobrepor e encadear sub-rotinas e armazenar e recuperar matrizes.

5.1 - CARACTERÍSTICAS DO RAMEXPAND

5.1.1 - TERMINOLOGIA

Ao longo deste capítulo, o termo RAM "expandido" será utilizado para denominar a memória presente no sistema que está contida nas placas LOGODATA 64/128K e, eventualmente, em placas de 16K.

Note-se que não é necessário dispor destas últimas; entretanto, nosso software as utilizará, se presentes.

5.1.2 - REQUISITOS E UTILIZAÇÃO DA MEMÓRIA

O RAMEXPAND ocupa duas porções de memória. A maior parte deste utilitário reside na placa de expansão (64/128K) que contém o sistema operacional relocado e ocupa o banco de 4K não utilizado dentro do banco de 16K da placa. Este é o banco 2B (conforme informações técnicas no capítulo 9). Uma página de memória de \$BDOO até \$BDFF (imediatamente abaixo das rotinas de interface do DOS relocado) é utilizada por RAMEXPAND; o apontador HIMEM é ajustado para proteger esta área.

O sistema RAMEXPAND consiste de 3 arquivos:

- RAMEXPAND
- RAMEXPAND.LOW
- RAMEXPAND.LOAD

O programa RAMEXPAND.LOAD serve para carregar o sistema RAMEXPAND na memória e ajustar diversos apontadores. O arquivo RAMEXPAND é aquele que reside na placa LOGODATA 64/128K; RAMEXPAND.LOW é aquela porção que ocupa a memória em HIMEM, no espaço de endereçamento real do sistema.

5.1.3 - ESTRUTURA DE DADOS

RAMEXPAND armazena e recupera partes de programas, texto ou dados. Cada porção é armazenada em uma área independente do RAM "expandido".

Estas áreas são denominadas segmentos. Quando a informação é armazenada num segmento, deve-se-lhe atribuir um nome, que será utilizado quando de sua recuperação. Os "nomes" são, na realidade, inteiros positivos, negativos, e menores que 32.767, ou variáveis inteiras, a serem especificadas quando RAMEXPAND é chamado.

5.1.4 - ALOCAÇÃO DE MEMÓRIA

O tratamento de cada banco de 16K nos quais a memória é subdividida é feito separadamente para um bloco de 12K e outro de 4K.

A organização das expansões LOGODATA 64/128K é a seguinte:

Como todos os bancos nos quais a memória é subdividida compartilham a região da memória ocupada pelo firmware e área de E/S da máquina, o maior bloco possível é de 16K bytes de extensão, visto que tanto entrada e saída como firmware podem ser desativados sem prejuízo da capacidade de armazenamento da máquina.

Disto resulta:

5.1.4.1 - Bancos de memória expandida, Interpretador ou Monitor, e E/S são disjuntivos, ou seja, somente um pode estar ativo de cada vez.

5.1.4.2 - Como consequência de 5.1.4.1, não é possível operar sobre a memória expandida diretamente.

5.1.4.3 - Todo acesso ao armazenamento que denominamos secundário deve ser feito através de E/S.

A organização é abaixo descrita:

A placa de 64K bytes é subdividida em 4 bancos de 16K bytes e a de 128K bytes em 8 bancos.

Cada banco de 16K bytes é dividido em um bloco de 12K bytes e um de 4K bytes; este último e o bloco de 4K bytes inferior do bloco de 12K bytes compartilham o mesmo espaço para evitar área de E/S.

Pictoricamente a reprodução é mostrada na figura 5.1.

Portanto, para se especificar um determinado bloco de memória, deve-se fornecer:

a - A designação da seção de 32K

256₁₀ = 1a. seção

512₁₀ = 2a. seção

1024₁₀ = 3a. seção

2048₁₀ = 4a. seção (correspondente ao "nibble" mais significativo na fig. 5.1)

b - A designação do bloco:

8 = 1o. bloco de 12K

16 = 2o. bloco de 12K

32 = bloco de 4K associado ao 1o. de 12K

64 = bloco de 4K associado ao 2o. de 12K (correspondentes aos dois "nibbles" menos significativos na fig. 5.1)

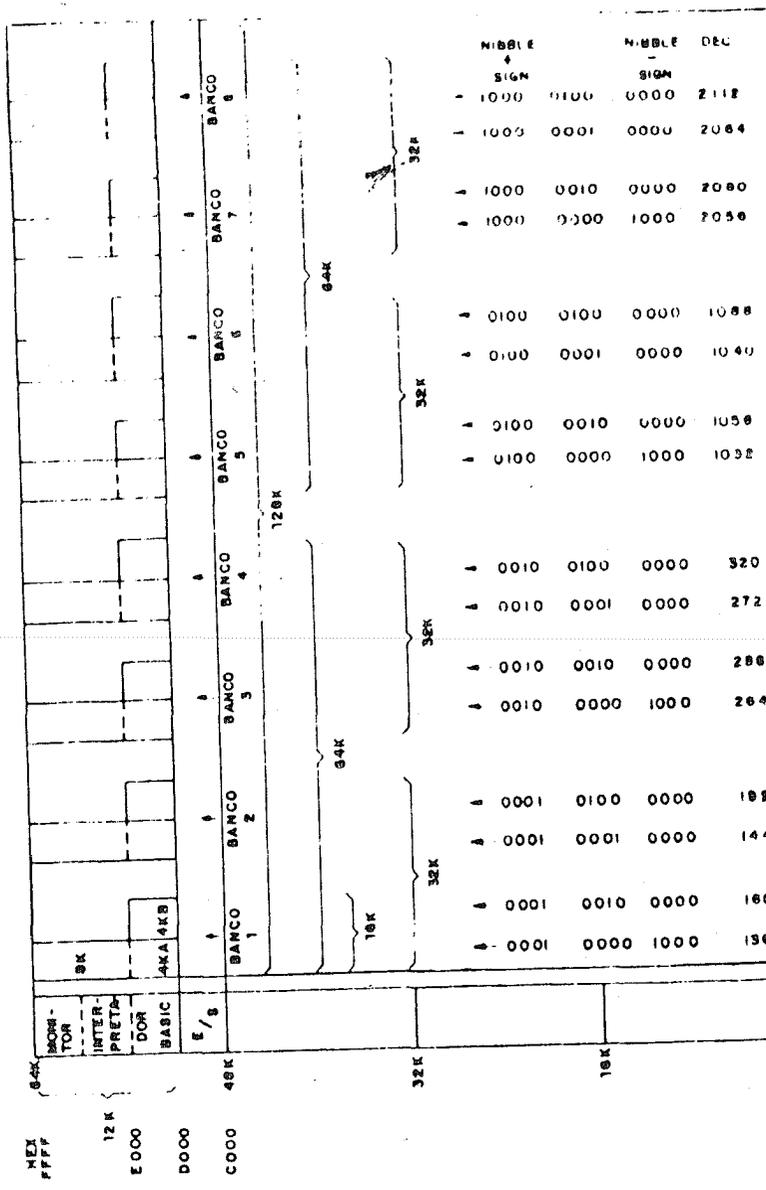


Fig. 5.1

HEX	NIBBLE SIGN	NIBBLE SIGN	NIBBLE SIGN	DEC
1000	0100	0100	0000	2112
1000	0001	0000	0000	2064
1000	0010	0000	0000	2080
1000	0000	1000	0000	2056
0100	0100	0000	0000	1088
0100	0001	0000	0000	1040
0100	0010	0000	0000	1056
0100	0000	1000	0000	1032
0010	0100	0000	0000	320
0010	0001	0000	0000	272
0010	0010	0000	0000	288
0010	0000	1000	0000	264
0001	0100	0000	0000	182
0001	0001	0000	0000	144
0001	0010	0000	0000	160
0001	0000	1000	0000	136

5.1.5 - TRATAMENTO ESPECIAL DO CARTÃO DE LINGUAGEM

Se presente, o cartão de linguagem necessita de tratamento especial devido à presença do ROM de partida automática (veja cap. 3, Item 3.2).

A designação de memória para RAMEXPAND foi ligeiramente modificada para acomodar o cartão de linguagem.

Os números de especificação do bloco permanecem os mesmos, com a diferença que os 2K superiores de cada bloco de 12K não são utilizados por RAMEXPAND. Como resultado, os blocos com designação \$10 e 16₀ passam a ser blocos de 10K (em lugar 12K declarados) na presença do cartão de linguagem. A existência deste último é verificada durante os passos de inicialização e instalação.

5.2 - USO DE RAMEXPAND

5.2.1 - INSTALAÇÃO

O procedimento abaixo descreve os passos a serem seguidos para instalar o RAMEXPAND no sistema.

Carregue RAMEXPAND através da execução do programa de carga RAMEXPAND.LOAD.

Este programa carrega os dois segmentos do sistema RAMEXPAND porção residente na memória expandida, bem como aquela que real espaço real de endereçamento entre \$BDOO e \$BDFF). O valor do parâmetro HIMEM é baixado para proteger este segundo segmento.

As posições de memória 10₁₀ até 12₁₀ são alteradas de modo a apontar endereço de entrada do RAMEXPAND. Como resultado, RAMEXPAND pode ser chamado utilizando-se o comando CALL 10 ou através do comando USER (APPLESOFT). Quando executado, RAMEXPAND.LOAD pergunta qual o conector no qual RAMEXPAND deve ser carregado:

– ENTRE COM O CONECTOR DESTINO DE RAMEXPAND

Responda com o número do conector no qual está colocada a expansão de memória onde se pretende instalar RAMEXPAND.

– TIPO DE CARTÃO

- (1) 128K OU 64K
- (2) 32K
- (3) 128K OU 64K, COM O CARTÃO DE LINGUAGEM
- (4) 32K COM O CARTÃO DE LINGUAGEM

Responda adequadamente.

Concluído o diálogo, a instalação está concluída. Dois procedimentos de inicialização são geralmente necessários antes de efetivamente moverem-se dados ou programas para a memória expandida, através das funções de RAMEXPAND.

Tais procedimentos serão descritos posteriormente.

5.2.2 - IMPLANTAÇÃO DOS COMANDOS

Antes de chamar RAMEXPAND, é indispensável criar-se uma variável de comando. O propósito deste último é passar um código de função e alguns parâmetros. O comando deve ser necessariamente denominado CM\$. RAMEXPAND buscará sempre a cadeia CM\$ quando for chamado. A seguir descreve-se sua sintaxe.

O comando CM\$ deve conter cinco valores, separados por vírgulas, cada um dos quais poderá ser um número ou uma variável.

Exemplo: CM\$ = "A, B, C, D, E"

As variáveis acima representam o seguinte:

- A - (função) — inteiro de 0 a 16 que identifica a função a ser executada.
- B - (segmento) — inteiro de 1 a 32.767, código do segmento de memória a ser utilizado.
- C - (código de erro) — deve ser uma variável; sob o nome desta variável RAMEXPAND devolve um código que representa os erros ou a execução bem sucedida de uma função.
- D - (início) — depende da função. Pode representar o início de uma área de programa ou um nome de uma matriz.
- E - (fim) — também depende da função; pode representar o fim de uma área de programa.

5.2.3 - CHAMADA DE RAMEXPAND

Conforme descrito acima, basta executar a instrução CALL 10, precedida do comando CM\$ para chamar RAMEXPAND. Esta seqüência deve ser repetida a cada comando.

5.2.4 - INICIALIZAÇÃO

Depois de instalado (5.2.1), RAMEXPAND deve ser inicializado, o que é feito através de um comando com código de função 14. A cadeia de comando para inicialização fica, portanto:

CM\$ = "14, 0, A, 0, 0"

onde,

A = variável de erro que deve ser definida antes da edição do comando

5.2.5 - ACOPLAMENTO DE BLOCOS DE MEMÓRIA

Após a inicialização é necessário especificar a memória expandida. A função de conexão (código 9) deve ser utilizada para efetuar acoplamento de blocos. Esta função pode ser utilizada para acoplar um ou mais blocos numa determinada placa de expansão.

- Para efetuar o acoplamento, siga as instruções abaixo.

5.2.5.1 - Calcule o parâmetro de seleção de bloco, somando o número do conector com 128 e a designação da seção de 32K definida na figura 5.1.

Exemplo: O parâmetro do primeiro bloco de 12K no conector 2 é dado por:

2 (número do conector)
128 (indica placa de 128K)
8 (designação do 1o. bloco de 12K)
256 (designação do 1o. bloco de 32K)

394

5.2.5.2 - Insira este parâmetro de seleção de bloco na cadeia de comando, como segundo parâmetro. Portanto no exemplo acima.

CM\$ = "9, 394, A, 0, 0"

onde,

A = variável de erro

Chamadas adicionais a RAMEXPAND (função 9) podem ser utilizadas para acoplar mais blocos, se necessário.

Exemplo: Especificar o primeiro e segundo blocos de 12K numa placa de 128 no conector 2.

2 (número do conector)
128 (tipo de placa)
8 (primeiro bloco de 12K)
16 (segundo bloco de 12K)
256 (primeira seção de 32K)

410 (parâmetro de seleção de bloco)

CM\$ = "9, 410, A, 0, 0"

5.2.6 - RESERVA DE ESPAÇO PARA O BASIC

O basic alternativo pode ter uma área reservada, evitando acoplar o primeiro bloco de 12K da placa de expansão. A utilização do segundo bloco permite o armazenamento do próprio RAMEXPAND, sem perturbar o BASIC no bloco de 12K.

5.3 - PRECAUÇÕES

5.3.1 - PROTEÇÃO DE RAMEXPAND

RAMEXPAND ocupa o bloco de 4K bytes com designação 64K. Este bloco não deve ser especificado para uso. Se o sistema operacional relocado estiver sendo utilizado, o bloco que o contém (designação 16) também não deve ser especificado.

5.3.2 - AJUSTE DE MAXFILES

Uma vez instalado, a parte de RAMEXPAND residente no espaço de endereçamento real (RAMEXPAND.LOW) não pode ser movida. Por consequência, MAXFILES não pode ser aumentada após a instalação de RAMEXPAND.

5.3.3 - RECARGA DE RAMEXPAND

Toda vez que RAMEXPAND.LOAD é executado, a parte inferior de RAMEXPAND é carregada e HIMEM reduzido para protegê-la. A execução recorrente deste programa implica na redução sistemática de HIMEM. Para evitar que o apontador HIMEM continue a ser reduzido, execute um comando MAXFILES antes de recarregar RAMEXPAND.

5.3.4 - TROCA DE VERSÕES DE BASIC

Sempre que os comandos EP ou INT forem executados é necessário declarar HIMEM, caso contrário os dados armazenados por RAMEXPAND serão perdidos. O valor requerido por HIMEM é dado quando RAMEXPAND.LOAD é executado.

5.3.5 - REPOSIÇÃO DOS PONTOS DE ENTRADA DE RAMEXPAND (ATENÇÃO)

O apontador para entrada de RAMEXPAND (normalmente coincidente com HIMEM) é armazenado inicialmente nas posições 11 e 12, se as mesmas forem alteradas, é evidentemente necessário restabelecê-las antes de usar RAMEXPAND.

5.3.6 - DECLARAÇÃO DE VARIÁVEIS

As variáveis a serem utilizadas como parâmetro na cadeia de comando devem ser declaradas antes de sua utilização.

Isto pode ser feito simplesmente atribuindo-lhes algum valor antes de chamar RAMEXPAND.

5.4 - MANIPULAÇÃO DE ERROS

5.4.1 - DESCRIÇÃO

Conforme descrito anteriormente, uma das variáveis da cadeia de comando é alterada por RAMEXPAND para devolver o código que descreve o resultado de sua execução.

- a) Se a cadeia de comando CM\$ não for encontrada não haverá indicação de erro.
- b) Se a variável de erro não for especificada, o código de erro não será devolvido. Entretanto, não havendo erros, a função será executada.
- c) Se uma variável não previamente declarada for especificada como variável de erro, o erro não será indicado, mesmo que ocorra.
- d) Se alguma das outras variáveis não for encontrada, a variável de erro será alterada convenientemente e nenhuma outra ação executada.
- e) Em geral, mas não sempre, nenhuma ação será executada na eventualidade de se encontrar uma condição de erro.

5.4.2 - SUMÁRIO DOS ERROS

- 1 - Função ilegal
- 2 - Variável não encontrada
- 3 - Bloco de memória já em uso
- 4 - Bloco de memória não encontrado
- 5 - Parâmetro de seleção de bloco ilegal
- 6 - O bloco de memória não é RAM
- 7 - Nome do segmento já em uso
- 8 - Segmento não encontrado
- 9 - Nome do segmento ilegal
- 10 - Programa não especificado
- 11 - Matriz não especificada
- 12 - Espaço insuficiente para armazenar dados na memória expandida
- 13 - Espaço insuficiente no espaço de endereçamento real para recuperar o programa
- 14 - Espaço insuficiente no espaço de endereçamento real para carregar a cadeia

5.5 - SUMÁRIO DAS FUNÇÕES DISPONÍVEIS

O primeiro valor da cadeia de comando identifica uma das seguintes funções:

- 0 - função nula
- 1 - armazena todo o programa na memória expandida
- 2 - busca o programa na memória expandida (ME) e executa
- 3 - armazena sub-rotina ou segmento de programa na ME
- 4 - buscar sub-rotina ou segmento de programa na ME
- 5 - armazena conteúdo de uma matriz na ME
- 6 - buscar o conteúdo de uma matriz na ME
- 7 - trocar o conteúdo de uma matriz entre a memória normal e a ME
- 8 - apagar segmento da ME
- 9 - conectar (acoplar) novo bloco de ME
- 10 - desacoplar bloco da ME
- 11 - buscar espaço disponível na ME
- 12 - buscar o nome do próximo segmento
- 13 - não definido
- 14 - inicializar RAMEXPAND
- 15 - armazenar setor de APPLESOFT na ME
- 16 - buscar um setor de APPLESOFT na ME

5.6 - SINTAXE E DESCRIÇÃO DOS COMANDOS

A sintaxe será baseada na cadeia de comando

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

5.6.1 - A = 0 – FUNÇÃO NULA

Descr.: Nada executa

5.6.2 - A = 1 – FUNÇÃO ARMAZENAR PROGRAMA

Descr.: Armazena o programa presente num novo segmento da memória

B = Nome do novo segmento da ME

C = Erro

D = 0 (não utilizada)

E = 0 (não utilizada)

Erros ativados – 0, 1, 2, 7, 9, 12

5.6.3 - A = 2 – FUNÇÃO BUSCA E EXECUTA

Descr.: Recupera o segmento endereçado por B e executa o programa ali residente.

B = Nome do segmento da ME

C = Erro

D = 0

E = 0

Erros ativados – 0, 1, 2, 8, 9, 13

5.6.4 - A = 3 – FUNÇÃO ARMAZENAR SUB-ROTINA OU SEGMENTO DE PROGRAMA NA ME

O nome do novo segmento não pode existir antes da execução. O programa é armazenado desde o primeiro número de linha até o último, inclusive.

B = Nome do segmento da ME

C = Erro

D = Número de linha inicial

E = Número de linha final

Erros ativados – 0, 1, 2, 7, 9, 10, 12

5.6.5 - A = 4 – FUNÇÃO BUSCA SUB-ROTINA OU SEGMENTO DE PROGRAMA

Recupera no segmento denominado por B a sub-rotina ou segmento e superpõe ao programa corrente começando pelo número de linha especificado por D.

B = Nome do segmento da ME

C = Erro

D = Número da linha inicial

E = 0

Erros ativados – 0, 1, 2, 8, 9, 13

5.6.6 - A = 5 – FUNÇÃO ARMAZENAR VETOR

(Basic inteiro) – O vetor deve ser especificado por uma variável (não pode ser número). Para vetores em Applesoft utilizar a função 15.

B = Nome do segmento da ME

C = Erro

D = Nome do vetor ou cadeia

E = 0 (não utilizada)

Erros ativados – 0, 1, 2, 7, 9, 11, 12

5.6.7 - A - 6 - FUNÇÃO RECUPERA VETOR

Recupera e armazena sob o nome a ele atribuído pela função o vetor ou cadeia alfanumérica (Basic inteiro). O valor deve ser especificado pelo nome. Se seu comprimento for especificado, RAMEXPAND devolve em lugar do comprimento, o número de bytes realmente armazenado. Para obter o número de elementos transferidos divide-se o número de bytes transferidos por 2 se os elementos forem inteiros ou por 8 se forem reais.

B = Nome do segmento da ME

C = Erro

D = Nome do vetor

E = Comprimento do vetor (retorna número de bytes transferidos)

Erros ativados - 0, 1, 2, 8, 9, 11

5.6.8 - A - 7 - FUNÇÃO TROCA CONTEÚDO DE VETORES

Com exceção da troca, é idêntica à função descrita em 5.6.7, inclusive o valor retornado. O vetor no espaço de endereçamento normal é trocado com o presente na ME.

B = Idem 5.6.7

C = Idem 5.6.7

D = Idem 5.6.7

E = Idem 5.6.7

Erros ativados - 0, 1, 2, 8, 9, 11

5.6.9 - A - 8 - APAGAR SEGMENTO

Cancela o segmento denominado por B, deixando disponível o espaço na ME para outros usos.

B = Nome do segmento

C = Erro

D = 0 (não utilizado)

E = 0 (não utilizado)

5.6.10 - A - 9 - ACOPLAR BLOCOS

Veja descrição em 5.2.2.

B = Parâmetro de seleção de bloco

C = Erro

D = 0

E = 0

Erros ativados - 0, 1, 2, 3, 5, 6

TABELA DE PARÂMETROS DE SELEÇÃO DE BLOCO

PLACA	ÁREA DA M.E. PROTEGIDAS PARA:			
	NADA	BASIC + RAMEXP.	DOS + RAMEXP.	DOS + BASIC + RAMEXP.
64K	1016+C	432+C 760+C	424+C 760+C	418+C 760+C
128K	4088+C	432+C 3832+C	424+C 3832+C	418+C 3832+C

* Para conectar estes blocos devem ser efetuadas duas chamadas consecutivas com os parâmetros mencionados. "C" é o número do conector.

5.6.11- A - 10 – FUNÇÃO DESACOPLAR BLOCOS

A função 10 desconecta os blocos da ME, se os mesmos não contiverem dados. Caso contrário, não serão os mesmos desconectados.

O formato do parâmetro de seleção de bloco é o mesmo da função anterior.

A variável D, se especificada, conterá após a execução o nome do segmento que ainda permanece armazenado no bloco.

Erros ativados – 0, 1, 2, 5, X (número do bloco)

5.6.12- A - 11 – FUNÇÃO DETERMINAR ESPAÇO DISPONÍVEL

A função computa o espaço disponível na ME. O número de bytes é desenvolvido na variável E.

B = 0 (não utilizada)

C = Erro

D = 0

E = Espaço disponível

Erros ativados – 0, 1, 2

5.6.13- A - 12 – FUNÇÃO RECUPERAR NOME DO PRÓXIMO SEGMENTO

Esta função permite varrer a ME para verificar quais nomes de segmentos foram utilizados. O nome do próximo segmento é devolvido na variável B. Se a variável D tiver sido declarada, o comprimento do novo segmento será devolvido na mesma.

Para obter o primeiro nome de segmento especifique a variável B com o valor 0.

Se o nome devolvido for 0 não há mais segmentos livres.

Erros ativados – 0, 1, 2, 8, 9

5.6.14- A - 13 – FUNÇÃO NÃO IMPLEMENTADA

5.6.15- A = 14 – FUNÇÃO INICIALIZAR RAMEXPAND

Vide descrição anterior.

Esta função desacopla todos os blocos e os dados anteriores são perdidos.

B = 0

C = Erro

D = 0

E = 0

Erros ativados – 0, 1

5.6.16- A = 15 – FUNÇÃO ARMAZENAR

Armazena o conteúdo do vetor (APPLESOFT somente) na ME. O vetor deve ser identificado por uma variável.

B = Nome do segmento

C = Erro

D = Nome do vetor

E = Não utilizado

Erros ativados – 0, 1, 2, 7, 9, 11, 12

5.6.17- A = 16 – BUSCAR CONTEÚDO DE VETOR NA ME

B = Nome do segmento

C = Erro

D = Nome do vetor

E = Não utilizado

Recupera o vetor na ME e armazena na variável designada em APPLESOFT.

Erros ativados – 0, 1, 2, 8, 9, 11, 14

PROGRAMA EXEMPLO

No diskette de distribuição foi incluído um programa exemplo que ilustra algumas funções do RAMEXPAND.

Este programa consiste dos seguintes arquivos:

DEMO

DEMO2

APPEND4

APPEND2

APPEND3

DEMOUTPUT

O programa demonstra a operação das seguintes funções:

1 - Armazenar vetor

2 - Buscar vetor

3 - Armazenar sub-rotina ou segmento de programa

4 - Buscar sub-rotina ou segmento de programa, além das funções de suporte de inicialização e acoplamento.

Para executar o programa exemplo:

- A - Reloque o D.O.S. (BRUN MOVEDOS)
- B - Execute RAMEXPA.LOAD (RUN RAMEXPA.LOAD) :
- C - Execute DEMO

OPERAÇÃO DO PROGRAMA

- A - O programa de demonstração inicializa RAMEXPAND, aloca o bloco de 4K com designação 32 usando a função (9).
- B - A seguir algumas rotinas padrão são executadas para a opção do "menu" PROGRAM LISTING. Isto é feito executando o arquivo DEMOUTPUT.
- C - Três segmentos de programa são lidos do disco e armazenados por RAMEXPAND usando a função (3).
- D - A parte principal do programa de demonstração é carregada e o "menu" apresentado.
As opções do "menu" são auto-explicativas.

CAPÍTULO 6 PSEUDO DISCO DOS (DISCO VIRTUAL)

Este pacote de software permite a utilização das placas de expansão LOGODATA 64/128K de modo similar ao de acionadores de discos flexíveis.

O sistema operacional DOS é modificado de modo a endereçar os dados à expansão assim como se a mesma fosse um acionador de discos magnéticos. Programas que utilizem DOS para armazenamento de dados verão a placa de expansão como um acionador adicional. O computador no qual as placas são instaladas deve ter o sistema operacional "remendado" de modo a reconhecer a existência do "DISCO VIRTUAL".

Para permitir estas modificações, um pacote de três rotinas foi incluído nos disquetes de distribuição que acompanham sua placa de expansão; são eles:

PSEUDO DISCO – Programa em Applesoft utilizado para instalar o Disco Virtual e modificar alguns parâmetros.

PSEUDO.NRM – Arquivo binário que contém os programas de comando do Disco Virtual e "remendos" do Sistema Operacional normal.

PSEUDO.REL – Arquivo binário com as mesmas características de PSEUDO.NRM, para ser utilizado com o sistema operacional relocado. (vide capítulos anteriores).

O arquivo PSEUDO PARÂMETROS é gerado na entrada do programa acima, e contém atualização dos parâmetros do programa PSEUDO DISCO, posição e tipo de placas adicionais de memória existentes.

6.1 CARACTERÍSTICAS E REQUISITOS DO DISCO VIRTUAL

O Disco Virtual é projetado para operar com os sistemas DOS 3.3 e DOS 3.2, normais ou relocados.

Uma ou mais expansões LOGODATA 64/128K podem ser utilizadas para armazenamento, até o máximo de 256K bytes.

Para implantar o Disco Virtual, são necessários dois procedimentos a serem seguidos, na ordem em que são mencionados.

6.2 AJUSTE DO PSEUDO DISCO

Este procedimento permite ao sistema hospedeiro conhecer a posição dos cartões de expansão e que porções dos mesmos devem ser utilizadas.

Execute o programa PSEUDO DISCO (PSEUDO.NRM e PSEUDO.REL devem estar presentes no mesmo disquete). Após a carga do programa, o seguinte "menu" de opções aparece na tela:

PSEUDO DISCO DOS

- (1) EXAMINAR AJUSTE DO DISCO VIRTUAL
- (2) AJUSTAR PSEUDO DISCO
- (3) INSTALAR PSEUDO DISCO
- (4) RECONECTAR PSEUDO DISCO
- (5) SAÍDA

SUA ESCOLHA:

Selecione a opção (2)
A tela mostrará o seguinte:

CONECTOR	BANCOS	ACIONADOR	DIR	SETORES
CONECTOR	12345678		CONECTOR	BANCOS 12345678
0			4	
1			5	
2			6	
3			7	

B - BASIC

U - USADO

N - NÃO USADO

- (1) MUDAR CONTEÚDO DO CONECTOR
 - (2) MUDAR PARÂMETROS DO DISCO
 - (3) SAÍDA
-

Selecione a opção (1) para indicar ao sistema onde os cartões se encontram. Descreva-se a seguir o diálogo com a máquina:

MÁQUINA: QUAL O CONECTOR?

Responda com o conector do primeiro cartão de memória.

MÁQUINA:

(0) VAZIO

(1) 128K RAM

(3) 32K RAM

(2) 64K RAM

(4) 16K RAM

O QUE HÁ AGORA NO CONECTOR?

Responda com o número correspondente ao conteúdo do cartão.

MÁQUINA: QUER UTILIZAR O CARTÃO INTEIRO?

Responda SIM em caso afirmativo. O "menu" de ajuste atualizará automaticamente o conteúdo do conector. Caso a resposta seja negativa, será editada pela máquina a seguinte pergunta:

MÁQUINA: QUER UTILIZAR O BANCO 1?

Responda SIM ou NÃO e repita o procedimento até que todos os bancos da placa tenham sido definidos.

Concluída esta operação, selecione a opção (3) para sair do modo de ajuste. O sistema responde a esta ação acionando o disco e atualizando o arquivo PSEUDO. PARÂMETROS. Lembre-se de que o disquete NÃO deve estar protegido contra escrita (com seu rasgo fechado).

6.3 - INSTALAÇÃO DO PSEUDO DISCO

Selecione a opção (3) do "menu" principal. O sistema carregará a versão apropriada do programa PSEUDO.XXX (XXX está por NoRMal ou RELocada), dependendo da atualização de parâmetros ter sido feita com Sistema Operacional normal ou relocado, e "inicializará" o Disco Virtual com diretório vazio.

A partir de agora o acesso ao Disco Virtual poderá ser feito da mesma forma que a um acionador real.

Por exemplo:

CATALOG,S5,D1

mostrará o diretório do Disco Virtual instalado no conector 5 (Volume 254 se o mesmo estiver vazio).

6.4 - RECONEXÃO DO PSEUDO DISCO

Cada vez que o Sistema Operacional é recarregado, as alterações introduzidas no mesmo são restauradas, resultando na desconexão do Disco Virtual. Quaisquer arquivos armazenados no mesmo, ainda estão presentes (desde que a alimentação não tenha sido desligada), mas estão temporariamente inacessíveis.

A maneira mais segura de reconectar o Disco Virtual é utilizar a opção (4) do "menu" principal para evitar a "reinicialização" da "trilha" de diretório, o que impediria acesso a esses arquivos.

6.5 - PRECAUÇÕES E OBSERVAÇÕES

6.5.1 - Embora haja alguma proteção, é recomendável evitar a utilização do Disco Virtual concomitantemente com o uso da memória expandida.

Dados do "Disco" poderão ser destruídos se for carregado, por exemplo, um programa no banco 1 do conector 0, quando o mesmo está sendo utilizado pelo Disco Virtual.

6.5.2 - Reinicialize disquetes reais com a presença do Disco Virtual. Com efeito, a cópia do Sistema Operacional (que será transferida quando o disquete for inicializado) conterá as alterações necessárias para operar com o acionador virtual. Este disquete não poderá efetuar "bootstrap", visto que o Sistema Operacional está modificado.

6.5.3 - Não é necessário "rodar" o programa de ajuste do Disco Virtual sempre que o mesmo for instalado. O passo 6.2 dos procedimentos acima pode ser eliminado se não houver modificações na configuração dos cartões dentro da máquina, visto que a mesma está armazenada no arquivo PSEUDO.PARÂMETROS.

6.6 - UTILIZAÇÃO DO DISCO VIRTUAL

O número de arquivos que podem ser armazenados no Disco Virtual depende da capacidade do cartão instalado.

Para um cartão LOGODATA de 128K bytes o "Disco" equivalente tem uma capacidade de aproximadamente 115K bytes depois de excluídos os 16K bytes necessários ao sistema operacional (se este for o Relocado), Tabela de Conteúdo do Volume e Diretório; portanto, pouco menor que a capacidade de um disquete de 16 setores por trilha, e pouco maior que a de um de 13 setores por trilha.

Devido à sua organização peculiar, (de "32 setores/trilha") programas padrão de cópia a nível de trilha e de "nibble" não poderão ser utilizados para cópia de dados do Disco Virtual. No pacote de software de distribuição, entretanto, fornecido juntamente com as placas LOGODATA 64/128K, o programa MOD foi incluído que permite modificar o programa FID padrão do Sistema Operacional de modo a permitir a operação com o Disco Virtual. Basta executar o programa MOD para introduzir numa alteração permanente, um FID que poderá continuar sendo utilizado após a modificação, mesmo com disquetes de 16 setores por trilha. O programa FID modificado pode também operar com o Sistema Operacional relocado mencionado nos capítulos anteriores.

6.7 - DESCRIÇÃO DO FUNCIONAMENTO DO DISCO VIRTUAL

O Disco Virtual consiste dos seguintes programas:

6.7.1 - PSEUDO DISCO

Programa em Applesoft que funciona como interface entre o usuário e os programas PSEUDO.NRM e PSEUDO.REL que operam efetivamente as alterações no Sistema Operacional, e instalação das rotinas diretoras do Pseudo Disco para emulação do disco real.

6.7.2 - PSEUDO.NRM

Conforme descrito acima, é o vetor da instalação do Disco Virtual. A inicialização (incluindo a trilha de diretório, se desejado) é efetuada pelo mesmo.

Suas rotinas de controle estão localizadas entre os "buffers" de arquivos e o começo do DOS, com HIMEM ajustado na direção descendente.

0.7.3 - PSEUDO.REL

Tem as mesmas funções descritas anteriormente, e é coligado automaticamente quando a máquina sente a presença do DOS relocado.

Suas rotinas estão instaladas abaixo das rotinas de interface do DOS relocado (\$BE00) com o apontador HIMEM acionado para protegê-las da escrita.

CAPÍTULO 7 – PSEUDO DISCO PARA O PASCAL

NOTA

**POR RAZÕES TÉCNICAS ESTE CAPÍTULO NÃO PODE SER INCLUÍDO
NO PRESENTE MANUAL**

CAPÍTULO 8 – PSEUDO DISCO PC/M

Este pacote de "software" prepara as expansões LOGODATA 64/128K para operação como acionadores virtuais sob controle do sistema operacional PC/M. Permite utilizar expansões de 16K, juntamente com as de 64/128K. Os benefícios em velocidade e desgaste reduzido são análogos aos descritos no capítulo 6, bem como a operação dos programas.

O "software" incluído neste pacote é constituído pelos seguintes programas:

PSEUDO.BAS – Programa em BASIC utilizado para indicar ao sistema onde se encontram e quais são as expansões presentes, bem como para alterar parâmetros do acionador (extensão do diretório, denominação do acionador etc.).

PSEUDO.COM – Programa em linguagem de montagem carregado e executado pelo programa acima, que cria os comandos INIT.COM e INIT2.COM.

INIT.COM – Estes programas, quando executados como comandos, efetivamente inicializam o Disco Virtual. O comando INIT2 age como INIT, com exceção de não inicializar a "trilha" de diretório depois de instalar as rotinas de acionamento do Disco Virtual.

PARAM.DAT – Arquivo que contém os parâmetros ajustados pelos programas acima. Dois procedimentos devem ser seguidos para instalação do Disco Virtual:

a - Ajuste de parâmetros.

b - Instalação dos "acionadores" e inicialização do sistema.

8.1 - AJUSTE DO PSEUDO DISCO

Este procedimento deve ser seguido para indicar ao sistema quais cartões de expansão estão presentes, onde se encontram, e que porções dos mesmos devem ser utilizadas.

Todos os arquivos e programas descritos na introdução acima são necessários e devem estar disponíveis em um disquete que contenha o sistema PC/M, instalado no acionador A.

8.1.1 - Para ajustar o pseudo disco carregue MBASIC e execute o programa PSEUDO como segue:

```
RUN "A:PSEUDO"
```

Como resultado a seguinte tela será apresentada:

PSEUDO DISCO PC/M

- < 1 > EXAMINAR AJUSTE DO DISCO VIRTUAL
- < 2 > AJUSTAR O DISCO VIRTUAL
- < 3 > ATUALIZAR INIT.COM E INIT2.COM
- < 4 > SAIR

QUAL É SUA OPÇÃO?

Selecione neste ponto a opção < 2 >. O sistema mostrará a seguir:

PSEUDO DISCO PC/M			
ACIONADOR C:		BLOCOS DE DIRETÓRIO 1	
CONECTOR	BANCO	CONECTOR	BANCO
0	12345678	4	12345678
1		6	
2		6	
3		7	

U = USADO	N = NÃO USADO	P = RESERVADO PC/M
-----------	---------------	--------------------

- < 1 > MUDAR CONTEÚDO DO CONECTOR
- < 2 > MUDAR PARÂMETROS DO DISCO
- < 3 > SAÍDA

QUAL SUA OPÇÃO?

Selecione agora a opção < 1 >
O sistema perguntará a seguir:

QUAL CONECTOR?

Responda com o número do conector a ser alterado.

O sistema reage perguntando qual o conteúdo do conector para identificar a placa: responda com o número associado ao tipo de placa que estiver utilizando.

Identificada a placa, você será perguntado sobre quais os bancos da mesma que deseja utilizar. Caso se trate do conector zero, será perguntado também se deve ser reservado espaço para PC/M 56.

Repita a seqüência acima tantas vezes quantas forem necessárias para poder ver na tela uma configuração que seja cópia fiel daquela que estiver usando. Selecione depois disto a opção < 2 > e altere os parâmetros do "Disco" segundo suas conveniências. As opções possíveis são:

-
- < 1 > MODIFIQUE NOME DO ACIONADOR
 - < 2 > MOD. NÚMERO DE BLOCOS P/ DIRETÓRIO
 - < 3 > SAÍDA
-

Selecione a opção < 1 > caso deseje atribuir ao novo "acionador" um nome diferente de C: (este último é selecionado por omissão).

Selecione a opção < 2 > para alterar os blocos alocados ao armazenamento da "trilha" de diretório (1 por omissão).

NOTA: Cada bloco pode conter 32 entradas de diretório.

Pela seleção da opção (3) do "menu" principal os comandos INIT e INIT2 são atualizados e ficam habilitados a "remendar" o Sistema Operacional, permitindo-lhe reconhecer a configuração recém alterada. Feito isso, o sistema retorna ao controle do Sistema Operacional.

8.2 - INSTALAÇÃO DO PSEUDO DISCO

Após os ajustes o acionador virtual está pronto para ser instalado, bastando para tanto executar, a partir do Sistema Operacional, o comando INIT ou INIT2 que efetive a instalação.

Conforme descrito acima INIT2 difere de INIT porquanto não inicializa o diretório do "disco" recém instalado.

A utilização deste último torna-se evidente no caso de um "bootstrap" eventual, sem desligamento da alimentação.

Neste caso pode-se manter o acesso aos arquivos que continuam na placa LOGODATA 64/128K, o que seria impossível se o bloco de diretório fosse reinicializado, pois os apontadores do mesmo estariam inevitavelmente perdidos impedindo acesso aos dados armazenados.

NOTA : Não é evidentemente necessário ajustar o PSEUDO DISCO toda vez que for necessário utilizá-lo. Com efeito, a configuração do sistema é parte integrante dos comandos acima, sendo necessário alterá-la somente se a configuração do sistema o for. Caso contrário, a simples execução desses comandos põe ao dispor do usuário o acionador virtual.

8.3 - UTILIZAÇÃO DO PSEUDO DISCO

Após a instalação o Disco Virtual é utilizado como se fosse realmente um acionador utilizando a denominação a ele atribuída. A transferência de arquivos pode ser efetuada por exemplo através dos comandos de programas de cópia tais como PIP. Um arquivo "Submit" pode ser utilizado para:

a - Instalar o Disco Virtual e

b - Copiar diversos arquivos empregando um único comando

Por exemplo: O arquivo abaixo, TRANS.SUB instalará o Disco Virtual (acionador C:) e copiará arquivos do acionador B digitando apenas SUBMIT TRANS

Arquivo TRANS.SUB

INIT

PIP C: - B: *.*

8.4 REQUISITOS E LIMITAÇÕES

- 8.4.1 O pseudo disco utilizará até um total de 256K byte de memória RAM, distribuída em diversos cartões de expansão.
- 8.4.2 As rotinas vetores do Disco Virtual residem no bloco de configuração de E/S reservado a vetores do usuário, portanto este espaço não estará disponível após a instalação do Disco Virtual.

CAPÍTULO 9 INFORMAÇÕES TÉCNICAS

9.1 ENDEREÇAMENTO

(Veja capítulo 5 sobre organização dos bancos de memória)

Conforme descrito anteriormente, a placa LOGODATA 64/128K é subdividida em blocos de 16K que compartilham o espaço de endereçamento do ROM da placa principal.

Além disso, como os 4K inferiores de cada bloco de 16K acabariam por ocupar a área de E/S (\$C000 a \$CFFF), este banco de 4K ocupa o endereço \$D000 até \$DFFF, sendo o endereçamento desta placa, portanto, compartilhado adicionalmente por mais este banco de 4K em cada bloco de 16K.

O acesso a cada bloco de memória é feito portanto especificando:

- 1 - O banco de 16K
- 2 - A página inferior de 4K a ser utilizada

Por convenção, o banco inferior de 4K de cada bloco de 12K é denominado por A e o bloco de 4K adjacente por B (veja figura 5.1).

9.2 CONTROLE DAS PLACAS 64/128K

Dezesseis endereços da área de E/S são utilizados no controle da placa. Os endereços estão compreendidos entre \$CON0 e \$CONF, onde N = B + número do conector onde a placa foi inserida. Estes endereços permitem:

- 1 - Selecionar a ME ou o ROM da placa principal
- 2 - Selecionar o banco de 16K desejado
- 3 - Selecionar a página de 4K inferior
- 4 - Habilitar ou desabilitar escrita na ME

Como usualmente, para colocar a ME em um determinado estado, basta fazer acesso ao endereço correspondente a este estado. Isto pode ser feito lendo ou escrevendo neste endereço (via BASIC através de POKE ou PEEK).

Os 4 bits menos significativos do endereço de controle determinam qual dos bancos de 16K será acessado, bem como o modo de operação dos mesmos.

A linha A2 é utilizada como controle de estados, quando em 0. A linha A2 em 1 habilita a seleção de até 8 bancos de 16K (no caso da placa de 128K) com as linhas A0, A1 e A3 indicando o endereço do banco. Com a linha A2 em 0, o significado das linhas A0, A1 e A3 é o seguinte:

A0 - Habilita (A0 = 0) ou desabilita escrita na ME

A1 - Em conjunto com a linha A0 é utilizada para desabilitar a ME

A3 - Seleciona a página de 4K dentro do banco de 16K (A3 = 0, página A)

A2	A3	A1	A0		LEITURA	ESCRITA
0	0	0	0	seleciona banco de 4K A	RAM	**
0	0	0	1	seleciona banco de 4K A	ROM	RAM
0	0	1	0	seleciona banco de 4K A	ROM	*
0	0	1	1	seleciona banco de 4K A	RAM	RAM

0	1	0	0	seleciona banco de 4K B	RAM	**
0	1	0	1	seleciona banco de 4K B	ROM	RAM
0	1	1	0	seleciona banco de 4K B	ROM	*
0	1	1	1	seleciona banco de 4K B	RAM	RAM
						:
1	0	0	0	seleciona banco de 16K número 1		
1	0	0	1	seleciona banco de 16K número 2		
1	0	1	0	seleciona banco de 16K número 3		
1	0	1	1	seleciona banco de 16K número 4		
1	1	0	0	seleciona banco de 16K número 5		
1	1	0	1	seleciona banco de 16K número 6		
1	1	1	0	seleciona banco de 16K número 7		
1	1	1	1	seleciona banco de 16K número 8		

* - desabilita a ME

** - "write protect"

A tabela acima mostra que a combinação apropriada permite acessar qualquer segmento da ME. Como resultado, é possível escrever na ME enquanto um programa é executado pelo interpretador residente em ROM.

Dois acessos ao endereço de controle são necessários sempre que se queira habilitar a ME para a escrita; caso a mesma já esteja habilitada, um único acesso é suficiente.

9.3 - UTILIZAÇÃO DA PLACA DE RAM

9.3.1 - DOS RELOCADO

O DOS relocado ocupa o banco 2 e o banco 2A. Este banco é habilitado pelo endereço de controle \$CON3 e pelo endereço de seleção de banco \$CON5.

9.3.2 - BASIC ALTERNATIVO

Ocupa o banco 1 e o banco 1A, e é habilitado pelo endereço de controle CON0 e pelo endereço de seleção de banco CON4.

9.3.3 - LEDS DE INDICAÇÃO

O led mais afastado do teclado (com a placa instalada) é acesso quando a ME está habilitada para escrita. Os demais fornecem a indicação binária do número de banco acessado (bit menos significativo mais afastado do teclado).



21

CX. 48

- LOGO DATA - 64/128 K