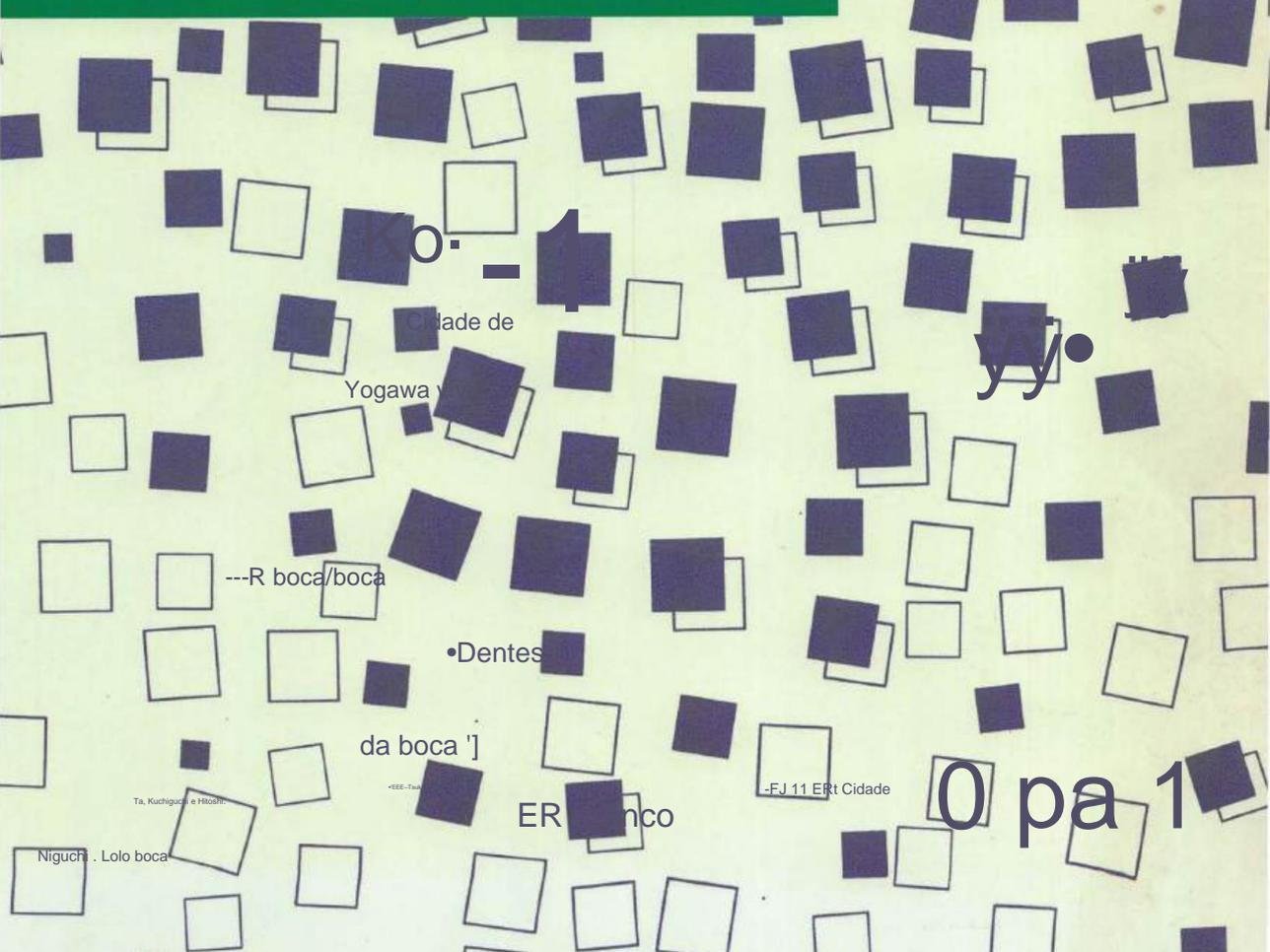


MSX マガジン編集部監修

# MSX turbo R Technical Hand Book

MSX turbo R テクニカル・ハンドブック



Yogawa

---R boca/boca

•Dentes

da boca ']

Ta. Kuchiguchi e Hoshiz

Niguchi . Lolo boca

ER nco

-FJ 11 ERt Cidade

0 pa 1

アスキー出版局



# MSXturbo R Técnico Manual



- MSX , MSX-DOS é uma marca registrada da ASCII.
- MS-DOS é uma marca comercial da Microsoft Corporation.
- OS-9 é uma marca comercial da Microsoft Corporation dos Estados Unidos e da Motorola Corporation dos Estados Unidos.
- EX é uma marca registrada da American Mathematical Society.
- Micro'IEX é uma marca registrada da Addison-Wesley Publishing, EUA.

- Outros nomes de CPU, nomes de sistemas, nomes de produtos, etc. usados neste manual são geralmente os nomes de produtos de cada fabricante de desenvolvimento.

É um sinal. Além disso, As marcas TM e @ não são especificadas no texto.

Este livro contém tudo, exceto a primeira página, o colofão e algumas ilustrações. , Baseado em ASCII `` Japonês ¥EX"  
Eu fiz o processo de composição. Gostaríamos de agradecer ao Sr. ishii@cts.dnp.co.jp, autor de msdos.sty, tecnologia de publicação  
e aos membros do departamento. Além disso, Meruhen gentilmente concordou em fazer a ilustração da primeira página.  
Porque a quantidade de livros disponíveis , Muito obrigado.  
para muitas pessoas é enorme Desta vez usaremos "MSX-DOS2 japonês". , memorando  
A descrição referente ao reamatsuper é omitida. Em relação a estes assuntos, consulte o livro ``Japanese", que será lançado em breve.  
Pretendo explicar isso no Manual Técnico do MSX-DOS2 (título provisório).



# Introdução

Bem-vindo ao mundo do MSX turbo R. Este livro contém informações privilegiadas necessárias para aproveitar ao máximo o computador pessoal MSX, que se tornou incrivelmente poderoso graças à sua CPU de alta velocidade e memória de grande capacidade, incluindo o seguinte: Esta é uma explicação detalhada.

1. Internamente de 16 bits, velocidade de processamento mais de 10 vezes mais rápida que o MSX anterior. CPU de alta velocidade , Técnicas para levar o desempenho do R800 ao limite.
2. Instruções para fazer pleno uso da fonte de som PCM e da fonte de som FM que são padrão no MSX turbo R. Informações e técnicas.
3. Como funciona o mecanismo SLOT e como utilizá-lo, que são essenciais para usar o MSX.
4. Sistema Kanji BASIC necessário para desenvolver software que lide com japonês.
5. Como usar o VDP para demonstrar suas técnicas na tela.

O MSX turbo R é o primeiro computador pessoal que usa uma CPU de 16 bits para alcançar um alto desempenho dramático sem fazer grandes alterações na arquitetura dos modelos anteriores.

é.

Outras máquinas mudaram completamente sua arquitetura quando passaram de 8 bits para 16 bits, então foram desenvolvidas por muitas pessoas em máquinas de 8 bits. Todo o software e know-how adquirido acabou sendo jogado fora.

Acreditamos que é necessário fazer a CPU de 16 bits para aumentar o desempenho do MSX, e para isso utilizaremos os ativos do software e hardware desenvolvidos para MSX. Também acreditamos que seria imprudente jogar fora. afastando o know-how do usuário. Para conseguir isso, pensamos que precisaríamos de um CPU compatível com Z80 para o novo MSX, e desenvolvemos o R800. Para alcançar total compatibilidade com o MSX anterior, desenvolvemos o MSX turbo R, que incorpora o Z80 anterior, bem como o recém-desenvolvido R800.

Desta forma, o MSX turbo R mantém a compatibilidade ideal com o MSX convencional. Portanto, os usuários podem utilizar os ativos de software que acumularam ao longo dos anos.

Basta executá-lo diretamente no MSX turbo R e você obterá uma melhoria de desempenho muitas vezes maior<sup>1</sup>. Além disso, o conhecimento necessário para desenvolver software pode ser usado como está, mas você pode usar parte do conhecimento explicado neste livro. Ao fazer isso, será possível realizar um sistema que realce ainda mais o desempenho do software. máquina e exibe excelente desempenho de custo.

Ryozo Yamashita, Gerente Geral, Departamento de Gerenciamento de Produtos 1, Divisão de Sistemas

---

<sup>1</sup> O software comercialmente disponível para 1MSX roda muito rápido no R800, tornando-o incompatível. Como o Z80 opera automaticamente, a velocidade pode não ser alta.

# Índice

<b>1 MSX turboR</b>	<b>15</b>
Hardware	16
1.1 MSX turbo R!	16
1.1.1 Estas são as características do MSX turbo R!	16
1.1.2 Configuração do sistema MSX turbo R.	16
1.1.3 Troca elegante de CPU.	18
1.1.4 Configuração ROM do MSX turbo R que pode ser empacotada com qualquer coisa.	18
1.1.5 Temporizador do sistema para ajustar a velocidade	19
1.1.6 Barco de E/S MSX turbo R.	20
1.1.7 Modo DRAM para aproveitar a velocidade	20
1.1.8 Estas são as características do R800!	20
1.1.9 Tudo sobre o R800 •	20
1.2 Como usar o MSX turbo R.	27
1.2.1 Como aproveitar a velocidade do R800. Programação.	27
1.2.2 Notas e problemas ao usar o R800.	27
1.2.3 Adicionado BIOS e sua descrição de função.	27
1.2.4 Sobre BIOS alteradas e excluídas	28
1.2.5 Notas sobre desenvolvimento de aplicações •	31
1.2.6 Exemplo de programa que alterna CPUs •	32
1.2.7	33
1.3 Como usar o P CM até o limite.	37
1.3.1 Noções básicas... Como usar com BASIC.	37
1.3.2 Comandos BASIC relacionados ao PCM •	38
1.3.3 BEEP Faça um som com PCM! •	39
1.3.4 Versão avançada... P CM em linguagem de máquina!	41
<b>2 VAGAS</b>	<b>47</b>
2.1 Slots são permitidos.	48

2.1.1 Como a CPU e a memória estão conectadas?	48
2.1.2 Explorando o interior da CPU Z80 de 8 bits •	48
2.1.3 Os tipos de memória variam dependendo da função	0,50
2.1.4 O que são slots MSX?	• 50
2.1.5 O segredo da capacidade de expansão do MSX está em seus slots.	52
2.1.6 Esses slots MSX2+ incomuns.	53
2.1.7 Faça a fenda + comprimento F largo.	0,55
2.2 Tente trocar de slot.	• 57
2.2.1 Para trocar de slot.	57
2.2.2 Como especificar o número do slot •	57
2.2.3 Funções do BIOS que operam slots.	• 58
2.2.4 Como saber a configuração do slot.	0,60
2.2.5 Explorando a área de trabalho do sistema	61
2.2.6 Especificações de hardware do MSX2+.	• 64
Ativação do dispositivo de prevenção de colisão	65
2.3 Configuração do slot MSX tubo R • 2.3.1-A	• 67
configuração do slot é finalmente unificada.	67
3 Kanji BASIC 3.1	71
Analisar Kanji BASIC	72
3.1.1 Hardware necessário para Kanji BASIC.	72.3.1.2
O que é software compatível com MSX-JE ?	• 73
3.1.3 Explicar o princípio de funcionamento do Kanji Dryper	• 73
3.1.4 Compatível com JE, Hardware e	• 75
Software.	76
3.1.5 Vários modos de tela que podem ser usados com Kanji BASIC •	3.1.6
Texto Kanji e gráficos Kanji •	3.1.7
A maneira correta de usar o driver Kanji	78
4V9958 VDP	81
4.1 Lista de registros V9958	83
4.2 Novos recursos do V9958.	• 85
4.2.1 Rolagem horizontal	85
4.2.2 Peso	0,87
4.2.3 Comando. •	0,87
4.2.4 Exibição do método YJK	87
4.3 Funções descontinuadas do V9958	89

4.4 Especificações de hardware V9958 (peças	90
alteradas) 4.5 V9958 e MSX2 +	91
4.5.1 Existem 12 tipos de modos de tela no total .	91
4.5.2 Controlando registros VDP. ÿÿ ÿÿ	92
4.5.3 Registros V9958	95
4.5.4 Rolagem horizontal usando VDP 4.5.5	95
Sob nenhuma circunstância você deve usar truques.	98
4.6 Dissecando o sistema YJK 4.6.	99
I A radiodifusão televisiva e o sistema	99
YJK 4.6.2 Estruturas de dados do sistema RGB e do sistema	99
YJK. ÿÿ 4.6.3 Programa de	101
amostragem de cores 4.6.4 É uma operação lógica mortal.	103
4.6.5 A chamada	105
descoloração 4.6.6 Qual é a diferença entre a TELA 10 e a 11	105
4.6.7 A TELA 11 também é um teloff ? Este é um truque	106
para exibir caracteres em 4.6.8 SCREEN 12 . 4.6.9 Método YJK e	108
registros VDP	108
4.7 Estudando interrupções de linha de varredura	110
4.7.1 Como funciona a exibição da tela do monitor? ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ. 4.7.2	110
Transmissão televisiva utilizando sistema entrelaçado	111
4.7.3 Tela entrelaçada no MSX2.	112
4.7.4 Explorando o princípio das interrupções da linha de varredura	113
4.7.5 Apresentando um exemplo de interrupções de	114
linha de varredura, 4.7.6 Vamos para a parte prática!	116
4.7.7 Registro VDP usado para interrupções de linha de	116
varredura. ÿÿ 4.7.8 Método de montagem e operação da peça	119
BASIC. ÿÿ 4.7.9 O princípio de funcionamento de	121
uma parte do montador 4.7. 10 Rotina de linguagem de máquina de interrupção de linha .	128
 5MÚSICA MSX	 129
5.1 O que é fonte de som FM?	130
5.1.1 História dos instrumentos musicais eletrônicos que levam	130
às fontes sonoras FM 5.1.2 Vamos analisar os sons	132
dos instrumentos musicais 5.1.3 Os intervalos não são	134
necessariamente equilibrados. ÿÿ ÿÿ ÿÿ 5.1.4 Vamos analisar	135
o MSX-MUSIC. 5.1.5 Experimentando sons rítmicos usando uma fonte de som FM	137

5.2	Controlar fonte de som FM	139
5.2.1	Tente fazer sons com um programa em linguagem de máquina •	139
5.2.2	Explique a visão geral da biblioteca • 5.2.3 Vamos	141
	compilar com MSX-É	149
5.3	Esta é a estrutura de dados de uma fonte de som FM •••••	150
5.3.1	Vamos criar dados de fonte de som FM 5.3.2 Vamos	150
	especificar os dados de som do instrumento de	152
	percussão. ••••• 5.3.4 Coisas que você não pode fazer com OP LL Dryno	154
5.3.5	Vamos tentar adicionar dados de tom 5.3.6 Explique os	156
	dados de amostra •	156
5.4	Coisas que você deve saber sobre fontes de	160
	som FM 5.4.1 Corrigido o conteúdo do poderoso método de	160
	uso. ••••• 5.4.2 Lista de dados de tons do MSX MUSIC. •••••	162
A	Tabela de instruções do R800	165
A.1	Como usar	166
	a tabela de instruções. ••••• A.2 Instrução de movimento de 8 bits • A.3	166
	Instrução de movimento de 16 bits. •••••	168
	•••••	169
A.4	Instruções de troca. • A.5.	171
	Instruções de manipulação de pilha.	171
A.6	Instruções de transferência de bloco. •••••	172
A.7	Instruções de busca de bloco •	172
A.8	Instruções de	172
	multiplicação. A.9	173
	Instrução de adição. A.10 Instrução de subtração. ••••• • A.11 Comando	175
	de comparação ••••• A.12 Instruções de operação lógica. •	176
A.13	Comando de operação de	177
	bits • A.14 Comando de rotação. •••••	178
	•.	
A.15	Comando Shift. •••••	181
A.16	Instruções da filial. •••••	182
• A.17	Instrução de chamada. •••••	183
A.18	Instruções de entrada/saída. ••••• ••••• ••••• A.19 Instruções de	185
	controle da CPU.	186



4.11 Procedimento de interrupção da linha	115
de varredura 4.12 Registro VDP que gera interrupção da linha de varredura	116
4.13 Registro VDP para detectar interrupções na linha de	117
varredura 4.14 Registro VDP para controlar a troca de tela	117
4.15 Mecanismo de rolagem vertical de hardware	118
5.1 Explore as estruturas de quatro tipos de instrumentos musicais eletrônicos	131
5.2 Analisando os sons básicos 5.3	132
Envelopes de instrumento e sintetizador	134
5.4 Dados sonoros de	153
instrumentos de	157
percussão 5.5 Dados de tom 5.6 Lista de registros OPLL /	161

# Índice

1.1 Muff I/O do MSX turbo R 0 • 1.2 Compare	.....
as velocidades de operação do Z80 e R800. • vinte e quatro	
• Lista de alterações de BIOS e BASIC na versão 1.3 MSX turbo R.	32
1.4 Porta de E/S para PC CM •	• 45
2.1 Área de trabalho do sistema para slots 2.2 Portas	61
de E/S MSX2+	64
3.1 Lista de hardware interno do MSX-JE.	73
3.2 Modo de tela Kanji BASIC. 3.3	77
Ganchos Kanji usados por secadores.	79
4.1 Modo VDP e modo de tela BASIC	• 82
4.2 Registro de modo.	0,83
4.3 Registro de Comando.	84
4.4 Registro de status ••	84
4.5 Alteração dos terminais do V9958 •	• 90
4.6 Características CC do V9958	90
4.7 Modo de tela MSX2+. •• 4,8 portas de E/	• 91
S VDP. 4.9 Local de armazenamento	• 92
do registro de controle. • 4.10 Outras áreas úteis de trabalho do	•.
sistema • 4.11 Áreas de trabalho do sistema adicionadas e alteradas	••• 94
no MSX2+. •• 4.12 Detalhes do endereço OFAFCH (MODE) • 4.13 Operação lógica •	••• 94.
	94
	104
5.1 Comparação do desempenho de instrumentos musicais	131
eletrônicos •• 5.2 Relação entre escala e frequência •••	• 134

5.3 Lista de temperamentos que podem ser configurados no MSX- . . . . .	.135
Music. 5.4 6 Som de instrumento + 1 Estrutura de dados de som de . . . . .	150
instrumento de percussão • 5.5 6 Som de instrumento + 1 Exemplo de . . . . .	.151
estrutura de dados de som de instrumento de . . . . .	• 152
percussão • 5.6 9 Estrutura de dados de som de instrumento • 5.7 Dados de som de instrumento .*****	
***yy yy yy yy yy yy yy yy yy yy 155 . . . . .	0,155
5.8 Exemplos de dados de som de instrumento • 5.9 Lista de dados de sons. . . . .	.163

1.1 第1章 MSX turbo R



Este capítulo foi publicado originalmente na edição de novembro de 1990 e na edição de dezembro de 1990 da MSX Magazine.

análise técnica do turbo R” , “Como usar o PCM em seus limites”

Esta é uma versão reeditada do artigo.

## 1.1 Hardware MSX turbo R

Equipado com CPU “R800” de 16 bits recentemente desenvolvida , 256 KB de RAM principal  
Conversamos sobre coisas como os recursos padrão do MSX-DOS2 que suportam diretórios hierárquicos.  
MSX turbo R tem muitos problemas Qual é a configuração do sistema desta máquina que chama a atenção?

Esta seção fornece uma visão geral do processo.

### 1.1.1 Esta é a característica do MSX turbo R!

- Além do Z80, é equipado com uma CPU “R800” de alta velocidade compatível com Kamisumi, resultando em média de 4 a 5 vezes mais potência e máximo

Atinge aproximadamente 10 vezes a velocidade (comparado ao MSX2+).

- Equipado com MSX-DOS2 japonês e Kanji Draino junto com MSX-DOS1. Suporta diretórios hierárquicos e variáveis de ambiente compatíveis com MS-DOS.

- Equipado com 256 kilobytes de RAM principal como padrão, compatível com mapeador de memória.

Além disso, a configuração do slot também foi padronizada.

- Equipado com função de gravação/reprodução PCM como padrão. MSX-, que anteriormente era um equipamento opcional.

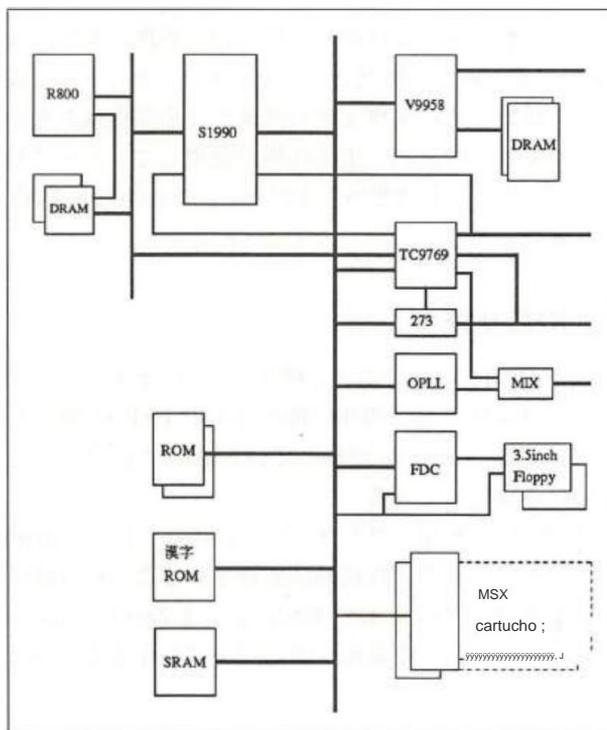
A MÚSICA agora também está incluída como equipamento padrão.

### 1.1.2 Configuração do sistema MSX turbo R

A configuração de hardware do MSX turbo R (doravante denominado turbo R) é mostrada na Figura 1.1. MSX legado  
O próximo MSX inclui o mesmo CPU compatível com “Z80” e o recentemente desenvolvido CPU . indústria  
“R800”, mas “o próximo MSX terá o Z280 da Zilog ou o proeminente HD64180 (ambos compatíveis com Z80).  
Ao contrário dos rumores de que parece que terá uma CPU de alta velocidade, o ASCII realmente criou a CPU.  
Isso mesmo.

O desempenho desse hardware é comparável ao de máquinas mais antigas de 16 bits e a velocidade da CPU é  
V30 (CPU de 16 bits desenvolvida pela NEC). Além disso, coloque um dicionário de conversão de kanji na ROM,  
Economizar RAM e espaço em disco é um princípio tradicional de design do MSX. notas recentes  
Também é usado em alguns tipos de PCs. Avalie o hardware do turbo R em uma palavra  
e , Pode-se dizer que “é isso que todos almejam”.

Figura 1.1: Configuração do sistema MSX turbo R



#### hardware turbo R

Configuração, linha de sinal de controle fino é omitido e expresso simplesmente.

A saída do V9958 é um sinal de vídeo.

Não., conectado ao TC9769(Z80)

A linha é o teclado e a alegria stick, a saída de 273 é

Lintur Turbo ySaída MIX

A energia é um sinal de áudio.

Vamos explicar a configuração do hardware com mais detalhes. Primeiro, vamos dar uma olhada no "TC9769" no diagrama.

A julgar pelo número do modelo, é um Toshiba CMOS-LSI (um tipo de LSI digital de baixo consumo de energia).

Comumente chamado de "MSX-Engine", inclui uma CPU compatível com Z80, uma fonte de som PSG, etc.

Como LSI. Doravante, quando a notação "z80" aparecer neste livro, ela se referirá a este chip.

significa.

"273" abaixo é o buffer de passagem para controlar a impressora, "OPLL" é o FM

A fonte de som, "FDC", refere-se a um controlador de disquete. Também chamado de "SRAM"

A memória serve para armazenar os resultados de aprendizagem do dicionário de kanji mesmo quando a energia está desligada.

Esta SRAM e o dicionário de conversão contínua de frases são recursos opcionais para os fabricantes.

A propósito, o R800 e a RAM principal são conectados via S1990 através do caminho (longa linha vertical no diagrama).

conectado a. Por exemplo, quando o R800 opera VDP, o S1990 recebe o sinal.

relé e, se necessário, envie um sinal de espera ao R800 para ajustar o tempo do sinal.

Execute operações que correspondam ao tempo dos sinais z80. Por outro lado, quando z80 usa RAM principal,

Quando isso acontece, o S1990 e o R800 retransmitem sinais e lidam com o mapeamento de memória.

A razão pela qual o turbo R tem uma configuração tão complicada é que não é possível usar hardware convencional ou Para manter a compatibilidade com o software. Acho que fiz um bom trabalho até agora. Apareça.

Como você pode ver, embora o turbo R possua um pequeno número de peças, seu processamento interno é extremamente complexo. Mudou para algo mais complicado. Além disso, o 81990 é um nó plano de 160 pinos: A soldagem manual por especialistas é impossível. O hardware está se tornando mais rápido e menor, mas No entanto, as técnicas de fabricação da boa e velha era dos microcomputadores de placa única não são mais aplicáveis. Eu fiz. Entretanto, os slots dos cartuchos MSX ainda são os mesmos, então o MSX continuará a ser usado. continuará a ser o material de referência para hardware introdutório. .

### 1.1.3 Troca elegante de CPU

As máquinas MSX2 da Victor, HC-90 e HC-95, permitem alternar entre dois tipos de CPUs. Eu usei trocando. No entanto, no turbo R, o LSI "81990" especialmente desenvolvido Ele gerencia o sistema, portanto, mesmo quando a energia está ligada e os programas estão em execução, a CPU está Você pode alternar e continuar executando o programa.

Graças a este hardware, o software MSX tradicional agora está no modo Z80 e turbo O software específico do R pode ser executado automaticamente no modo rápido R800. Tornou-se. Além disso, verifique o tipo de hardware, se for um MSX convencional, escolha Z80, ou se for turbo Se você quiser R, você pode escolher R800, e também podemos criar software para MSX2/turbo R. Ouvir.

## 1 Configuração ROM 1.4 do MSX turbo R que empacota tudo

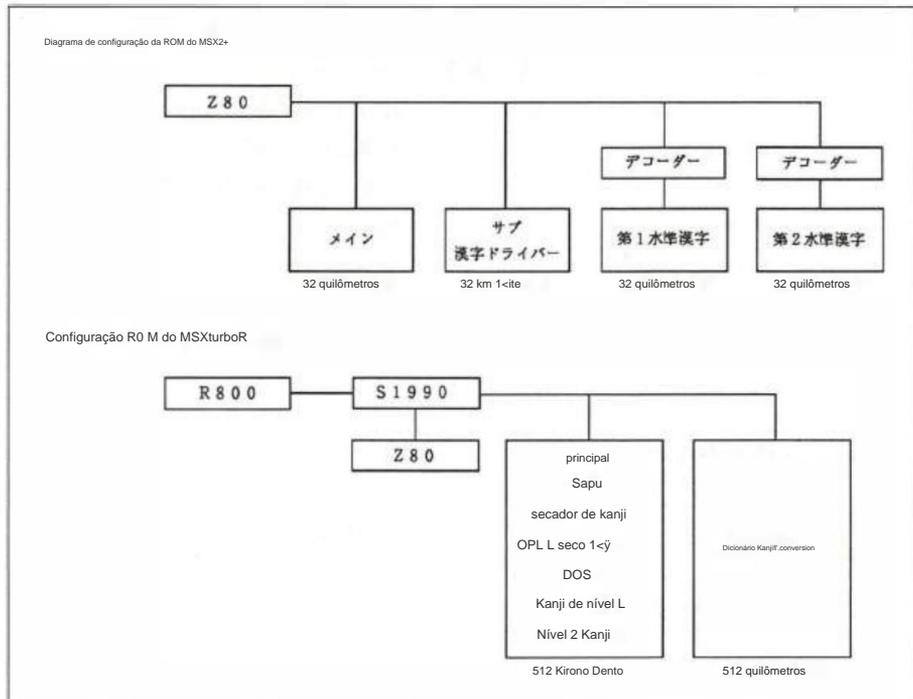
Supõe-se que o turbo R tenha muita ROM integrada, mas quando você abre a tampa, O número de ROMs é surpreendentemente pequeno. A razão para isso é a função de controle de mega ROM do 81990.

O MSX2+ possui uma ROM embutida como mostrado na parte superior da Figura 1.2. ROM principal e O bloco ROM está conectado a diferentes slots e o Kanji ROM está conectado à porta de E/S. Portanto, independente da capacidade total, cada ROM deve ser separada. cervo e usa uma ROM de 128 KB em vez de quatro ROMs de 32 KB. É mais barato, requer menos área de placa e consome menos energia.

Portanto, no turbo R, conforme mostrado na parte inferior da Figura 1.2, a ROM de 1 512 kilobytes é Principal, secundário, driver OPLL, kanji de 1º nível do DOS, kanji de 2º nível, todos Eu empacotei e guardei. No entanto, a mega ROM 81990 entre a CPU e a ROM As funções de controle permitem que o software controle, por exemplo, ROM Kanji de nível 1, a partir da porta de E/S. Parece que está conectado aos endereços D8H e D9H do alvo.

Além disso, há um total de 164 kilobytes de ROM DOS (MSX DOS tem 16 kilobytes, MSX- (DOS2 tem 48 KB), faça 4 furos no espaço de 16 KB no slot 3-2.

Figura 1.2: Mudanças na configuração da ROM no MSX turbo R



Eles estão conectados de maneira alternada.

### 1.1.5 Temporizador do sistema para ajustar a velocidade

O R800 tenta usar o V9958 (LSI que controla a exibição da tela) em intervalos de 8 microssegundos ou menos.

O circuito de interface VDP integrado no 81990 conectará automaticamente o R800.

Acenda uma luz. Isso pode causar mau funcionamento do V9958 devido ao processamento da CPU ser muito rápido.

Não há pagamento.

No entanto, outros LSIs periféricos não possuem funções de espera automática, então o software  
Você precisa ajustar seu tempo sozinho. Software mais tradicional

EX (SP), H.L.

EX (SP), HL

ou,

PRESSIONE HL

POP HL

Ao incorporar instruções demoradas, mas sem efeitos colaterais, em programas, Eu estava ajustando o tempo. No entanto, conforme explicado posteriormente, o tempo de execução de instruções do R800 Como é incerto, é impossível obter o tempo usando este método. Então turbo R possui um "temporizador de sistema" para ajustar a velocidade.

Este é um contador de 16 bits que aumenta a cada 3,911 microssegundos.

está conectado ao endereço da porta de E/S E6H, e o byte Kamizumi está conectado ao endereço E7H. no entanto, É inconveniente se você tentar ler um valor de 16 bits e o valor do contador mudar no meio. É melhor usar apenas o byte inferior ou o byte superior.

A Listagem 1.1 é um exemplo de programa que espera pelo valor no registrador B x 3,911 microssegundos. mosquito Escreva um programa para usar o valor do byte superior em vez do byte inferior do contador. Se você alterar, ele aguardará o valor no registro B x 1001,2 microssegundos. Você pode criar um programa .

#### Listagem 1.1 (TIMER.Z80)

```
.Z80
CONTAGEM      EQ      OE6H      ; Balcão inferior {de 8 bits
BAIXO CONTAGEM ALTA  EQ      OE7H      ; Contador superior de 8 bits
;
; Aguarde o registro B {Direct*3.911uS
; Erro é -3.911uS.. +OS ; Não
; defina 0 como t.
; Interrupções devem ser proibidas.
; C , UM , F é uma garota mentirosa
;
ESPERE:
    EM      A, (CONTAGEM)      ; Obtenha o valor atual do contador
    L. D.   C,A                ; salve-o
WAIT_LOOP:
    EM      A, (CONTAGEM) c    ; Obtém o valor atual do contador
    SUB                    ; Calcular o tempo decorrido
    GP      B                  ; O tempo especificado já passou?
    JR      C,WAIT_LOOP       ; Se não tiver decorrido, faça um loop.
    RET
```

### 1.1.6 Porta MSX turbo R 1/0

Os materiais de comunicado de imprensa do turbo R não incluíam um mapa de E/S, então entrevistas e As informações obtidas através da análise de hardware foram adicionadas ao mapa MSX2+ 1/0, e o departamento editorial fez o muflo 1/0 mostrado na Tabela 1.1. Os itens anotados com "R" foram adicionados recentemente ao turbo R. Esta é a porta 1/0 adicionada.

Em primeiro lugar, um "conversor D/A" é um dispositivo que pode gravar e reproduzir PCM sem passar pelo BIOS. Porta 1/0 para operação. Apresentarei os detalhes mais tarde. "Pausar controle da tecla" Porta I/O para proibir ou permitir o encerramento do programa pela tecla Z. Insira o disco

## 1.1 Hardware MSX turbo R

00000000

Tabela 1.1: Mapa MSX turbo R 1/0

Título	Uso	Nota	Hardware
OOH-3FH	caseiro		
40H ~ 7BH	Fabricante desativado. ção		
7CH-7DH	OPLL	+ B	
80H-87H	RS-232C	1 B, X	
88H-8BH	VDP externo	2xx	
90H-93H	F. Injetor	2	
98H-9BH	VDP	+1	
AOH-A2H	PSG		
A4H-A5H	Comparador D/A R Pose		
A7H	Kippine LJ Go R 8255		
CINZA-BH		1B	
ACH-AFH	Motor MSX	2	
BOH-B3H	SRAM 1xx da SONY		
B4H-B5H		2	
BBH-BBH	Suporte de relógio	2xx	
BCH - BFH	VHD atacado	2xx	
COH-C1H	MSX-Audio Interface MSX	2xx	
CBH-CCH	2 xx Disquete 2 - xx ROM Kanji de 1º nível 2		
DOH-D7H	2º 7)-Quasi Kanji ROM 2 Expansão de ROM, Kanji R -		
DAP-D9H	x R?		
DAH-DAP			
DCH			
E3H-E5H	?		
E6H-E7H	Temporizador do		
F4H	sistema R Redefinir status		
FSH	+ Bte . torta habilitar 2		
Memória	Controle AV	2x	
FCH-FFH	F6H - F7H Matsuno-ý:- 2 B		

Baseado em pesquisa do Departamento Editorial da MSX Magazine

1 compatível com MSX1.

2 Compatível com MSX2.

+ Compatível com MSX2+.

Recentemente instalado no R turbo R.

B Sempre opere através do BIOS.

fazer

— programa de aplicação

não deve ser operado.

x Recursos de opção do fabricante. Ta

No entanto, não foi implementado no

protótipo da máquina turbo R investigado desta vez.

Estava tão quente.

xx Contido em máquinas convencionais

Porém, a partir das especificações do turbo R

Excluído.

? Algo está conectado, mas o

Não está escrito na carta. grau

Para inspeção de hardware

Parece haver um registro.

Para evitar que o programa seja interrompido durante a saída e o disco seja destruído,

Parece que está preparado.

A "expansão da ROM Kanji" inclui ROM Kanji de 24 pontos e JIS que podem ser criados no futuro.

Parece ser uma função de reserva em preparação para a ROM Kanji de terceiro nível. O "?" abaixo não se aplica a nenhum material.

Embora não esteja escrito, ao ler e escrever na porta 1/0, algo acontece dentro do 81990.

O hardware parece funcionar. Não seria uma porta 1/0 do hardware turbo , para inspeção de fábrica

R? E o "temporizador do sistema" é conforme explicado acima.

Pertence a você.

Embora isso não esteja escrito na tabela, também é necessário lidar com a velocidade do turbo R. ,

PSG, joystick, mouse, impressora, teclado, relógio (bateria)

O BIOS deve ser usado para operar o IC do relógio de backup. A seguir, embora seja um recurso comum no MSX2+, eu gostaria de adicionar uma explicação adicional ao "reset status". Esta é uma porta de E/S que distingue entre uma reinicialização de hardware e uma reinicialização, saltando para o endereço 0 na ROM principal. Especificamente, quando o endereço 17AH da ROM principal é chamado, o valor deste status de reset é lido no registrador A, e quando o endereço 17DH é chamado, o valor do registrador A é escrito no status de reset Incluído. por exemplo,

LIGUE 17AH

DR80H

LIGUE 17DH

RST 0H

Seguindo estes passos e configurando o bit 7 do status de reset para 1, e então saltando para o endereço 0, você pode reiniciar o MSX de forma confiável. A

propósito, a razão pela qual você não deve usar o status de redefinição sem passar pelo BIOS é que a lógica do sinal de hardware do status de redefinição é invertida dependendo da máquina. O BIOS compensa a diferença.

Com o turbo R equipado com DOS2 como padrão, a memória tornou-se cada vez mais importante. "Mapper". Requer um procedimento um tanto complicado para usar e operar o BIOS estendido.

Como observação lateral, a Tabela 1.1 também inclui funções que foram colocadas em uso prático ou prototipadas no passado, mas não estão incluídas em MSXs recentes. Eu acho que o recente MSX se tornou um computador comum e há poucos periféricos sofisticados, mas o que você acha?

### 1.1.7 Modo DRAM para aproveitar a velocidade

Cada tipo de memória possui um limite de intervalo mínimo de tempo entre a leitura e a escrita, denominado "tempo de acesso". Se a velocidade da CPU for muito rápida, será necessário adicionar um "tempo de espera" para combinar a velocidade da CPU com a memória. Esse tempo de acesso varia dependendo do tipo de memória, e quanto mais rápido a memória puder ser usada, mais cara ela se tornará. Também, em geral, a RAM tem um tempo de acesso mais curto que a ROM.

Agora, para aproveitar a velocidade do R800, é melhor ter programas armazenados em RAM do que em ROM. Portanto, um "modo DRA" foi criado que transfere o conteúdo do BIOS, BASIC, sub ROM e ROM do driver Kanji para DRAM (RAM principal) para uso.

Isso desconecta os últimos 64 kilobytes da RAM principal do mapeador de memória e Após transferir o conteúdo da ROM, ela fica protegida contra gravação e conectada à CPU. CPU

Parece que a ROM normal foi substituída por uma ROM de alta velocidade. Ao executar programas escritos em BASIC, a ROM contendo o BIOS e o interlitter BASIC é frequentemente usada, para que a velocidade do modo DRAM possa ser aproveitada.

No entanto, ao executar programas em linguagem de máquina, especialmente programas DOS, a ROM é usada por um período de tempo relativamente curto. Portanto, em vez de usar o modo DRAM, pode ser mais vantajoso utilizar o excesso de memória como um disco RAM.

Os programas em cartuchos ROM também rodam mais rápido quando transferidos para a RAM, mas com o turbo R, as versões em disco do software se tornarão mais populares do que nunca.

### 1.1.8 O R800 é único!

- Z80 e controle de objetos. Portanto, o software para o Z80 também é  
Funciona exceto em peças que dependem do tempo.
- A velocidade do clock da CPU é de 7,16 MHz. Além disso, como o número de clocks por instrução é significativamente reduzido em comparação com o Z80, quando convertido para o Z80, é equivalente a 29 MHz (quando em modo sem espera).
- Suporta instruções de multiplicação com precisão de 16 bits x 16 bits y 32 bits. É isso  
Isso permitiu melhorar significativamente a velocidade de processamento de cálculos.
- No Z80, o endereço para cada 8 bits superior/inferior do registrador IX IY era indefinido.  
o acesso foi formalmente garantido.

### 1.9 R800 Todos 1

O R800, que foi adotado como CPU do turbo R, é um CPU de alta velocidade com software compatível com o Z80 convencional. Isso significa que o software desenvolvido para o Z80 pode ser executado em alta velocidade no R800, desde que a CPU não seja muito rápida.

Os recursos adicionados ao Z80 incluem uma instrução de multiplicação de 16 bits e uma instrução de acesso de byte de registro IX/IY, que foi considerada um "truque" no Z80. Para obter mais informações, consulte o quadro de instruções do R800 incluído no apêndice deste livro.

A frequência de clock do MSX convencional é 3,58 MHz e a frequência de clock do Turbo R é 7,16 MHz. Quando você olha para isso, parece que a velocidade simplesmente dobrou, mas na verdade não é o caso. Com o R800, o número de clocks necessários para executar uma única instrução é reduzido e não há necessidade de esperar ciclos MI para acessar a RAM, portanto a velocidade de execução do programa fica ainda mais rápida. Mesma velocidade de processamento do R800 com Z80 convencional

Aproximadamente 2,9 vezes para alcançar. Como se diz que a frequência do clock é igual à do clock original, há uma velocidade considerável.

Isso significa que houve um aumento no número de pessoas.

Tabela 1.2: Comparação da velocidade operacional do Z80 e R800

		MSX2+ turbo R (único µs) (unidade: µs)	Ampliação	倍率
L. D.	R,s	1,40	0,14 x	10,0
L. D.	r <sub>i</sub> (HL)	2,23	0,42 x	5,3
L. D.	r <sub>i</sub> (IX+n)	5,87	0,70 x	8,4
EMPURRAR	qq	3,35	0,56 x	6,0
LDIR	(BCY0)	6,43	0,98 x	6,6
ADICIONAR	A, r	1,40	0,14 x	10,0
INC.	R	1,40	0,14 x	10,0
ADICIONAR	Olá,ss	3,35	0,14 x	24,0
INC.	ss	1,96	0,14 x	14,0
JP		3,07	0,42 x	7,3
JR		3,63	0,42 x	8,7
DJNZ	(B ≠ 0)	3,91	0,42 x	9,3
CHAMAR		5,03	0,84 x	6,0
RET		3,07	0,56 x	5,5
MULTU	A,r	—	1,96	—
MULTUW	HL,rr	—	5,03	—

Agora, os resultados da comparação da velocidade do Z80 e R800 para cada tipo de instrução são mostrados na Tabela 1.2. Ré

1.2 A transferência de dados entre registradores (instrução LD) e a velocidade de adição são 10 vezes mais rápidas, para atenção. Vale a pena. No entanto, os valores desta tabela baseiam-se na velocidade medida quando o R800 se move sem esperar. de. Na realidade, o peso pode reduzir a sua velocidade, por isso tome cuidado. Além disso, Explicaremos detalhadamente posteriormente as condições que causam as isenções e como evitá-las.

A estrutura interna do R800 é mostrada na Figura 1.3. No R800, o caminho de dados externo é Embora tenha 8 bits, o caminho de dados dentro da CPU é de 16 bits. Então adição de 16 bits Instruções e similares são processados em um ciclo.

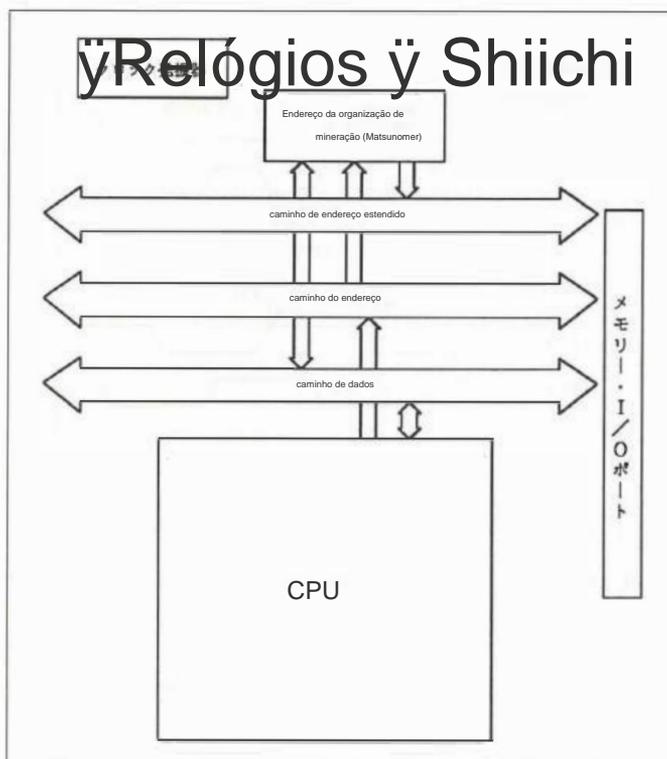
Olhando para esta configuração de hardware, o R800 tem um desempenho superior ao CPU Z80 de 8 bits. Uma CPU de 16 bits com caminho de dados de 8 bits, como "8088" da Intel ou Moto Parece estar próximo do "MC68008" de Laura.

Observe que existe algo chamado "mecanismo de expansão de endereço (Matsupar)" no topo da Figura 1.3. Porém, parece que foi preparado para usar o R800 para outras coisas além do MSX. Usado no turbo R.

Se quiser usar o R800, você pode usar o mecanismo de controle de slot integrado ao 81990 e o tapete de memória em vez do R800. Nomar controlará o sistema.

A seguir, vamos explicar detalhadamente o "acesso à página DRAM". Em primeiro lugar, este

Figura 1.3: Diagrama de blocos interno do R800

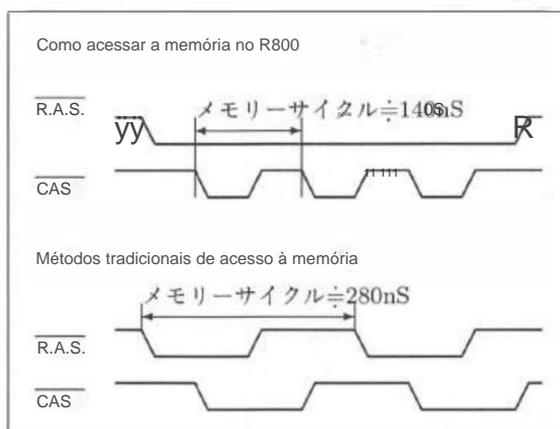


A parte inferior da Figura 1.4 mostra como acessar a memória usando o Z80. Depois de enviar o byte superior do endereço (endereço de linha) para a DRAM, diminuir o sinal RAS (strobe de endereço de linha) e enviar o byte inferior do endereço (endereço de coluna) para a DRAM, Reduza o sinal CAS (estroboscópio de endereço de coluna). Isso especifica o endereço da memória.

Por outro lado, a parte superior da Figura 1.4 mostra o acesso à página DRAM no R800. Mantendo fixo o byte superior do endereço e do sinal RAS e alterando apenas o byte inferior do endereço e do sinal CAS, a DRAM é usada com o dobro da velocidade dos métodos convencionais. Desta maneira, No R800, o acesso à página é executado automaticamente quando a DRAM é usada continuamente sem alterar o byte superior do endereço.

O tipo de DRAM fácil de conectar ao R800 é de 256 kilobits. (32 kilobytes), Imecha. Os exemplos incluem broca (128 quilômetros) e 4 brocas mecânicas (512 quilômetros). O valor mínimo para a capacidade da RAM principal foi definido em 256 kilobytes.

Figura 1.4: Diferença no método de acesso à memória entre Z80 e R800



Mesmo com turbo R, apenas duas unidades de DRAM de 1 bit mecânico são suficientes.

1 O primeiro MSX, desenvolvido em 1983, usava oito DRAMs de 16 kilobits,

Mesmo assim, a capacidade principal da RAM era de apenas 16 kilobytes. Quando penso nisso, eu

As capacidades tecnológicas atuais permitem-nos atingir uma capacidade de RAM de 256 kilobytes com apenas duas DRAMs.

incrível. A funcionalidade do MSX aumentou, mas o tamanho do hardware e o consumo de energia diminuíram.

Está ficando mais frio. Os resultados da aplicação da mais recente tecnologia de semicondutores do Japão tornaram-se recentemente um tema quente.

Pode-se dizer que este é o advento dos laptops e do Turbo R.

## 1.2 Como usar o MSX turbo R

### 1.2.1 Programação que aproveita a velocidade do R800

É verdade que o RSOO é rápido, mas para aproveitar ao máximo sua velocidade, ou seja, para evitar esperas e aproveitar o poder do RSOO, é necessária alguma engenhosidade de programação. Lembre-se de que há 3 esperas para acessar um slot externo, 2 esperas para acessar a ROM interna e 1 espera quando a DRAM interna não pode ser paginada. É algo que acontece.

Idealmente, o programa deve ser colocado em um intervalo de 256 bytes (intervalo acessível por página), onde os bytes superiores dos endereços internos da RAM são iguais e os dados devem ser colocados em registros. Neste caso, não ocorre acesso à memória para os dados, e o acesso à memória para a CPU ler o programa também é realizado em modo página, portanto não há espera na CPU. Embora seja difícil criar todos os programas desta forma, seria uma boa ideia abordar esta condição mesmo para as sub-rotinas que requerem mais velocidade.

Endereços de programa, dados e pilha estão relacionados ao acesso à página.  
fazer. por exemplo,

```
EMPURRAR      H.L
```

O tempo de execução de uma instrução é de 4 clocks se o byte superior do endereço onde a instrução está colocada corresponder ao byte Kamizumi do ponteiro da pilha. Se não corresponder, são 5 relógios. Provavelmente não há necessidade de pensar até aqui ao criar um programa, mas é importante lembrar que o tempo de execução dos comandos difere dependendo da situação.

### 1.2.2 Notas e problemas ao usar o R800

Com o zso, a DRAM era atualizada cada vez que uma única instrução era executada. No entanto, com o RSOO, são necessários 280 nanossegundos para atualizar a DRAM a cada 31 microssegundos. Observe que não é possível prever com precisão o tempo de execução do programa RSOO devido ao tempo necessário para esta atualização e às condições de acesso à página DRAM mencionadas anteriormente.

Portanto, para ajustar a velocidade do programa, usamos algo chamado "temporizador do sistema". Explicarei mais tarde como alterar o timer do sistema e como ajustar a velocidade entre a CPU e o V DP, então aguarde.

Além disso, embora isso possa ser dito sobre qualquer CPU nova, um possível problema com o RSOO é a falta de equipamentos de desenvolvimento. Especialmente ao desenvolver software

ICE (emulador no circuito) pode ser usado para depuração. É inconveniente.

Portanto, para criar software para turbo R, primeiro precisamos do MSX convencional e do Z80. Depurei-o completamente usando um ICE personalizado e ele deve funcionar de maneira confiável. programa para turbo R. Seria bom ter uma maneira de consertar isso usando um arquivo compatível com Z80. Crie um arquivo de log, verifique a operação e, em seguida, interrompa-o. Apenas a parte que utiliza multiplicação deve ser reescrita para R800. Neste momento, a rotina também é uma boa ideia verificar o funcionamento. Se for esse o caso, Por fim, monte tudo e mova-o. caso... tudo o que posso fazer é olhar a lista de fontes e pensar a respeito.

### 1.2.3 Adicionado BIOS e sua descrição de função

Para controlar os novos recursos de hardware do turbo R, comutação de CPU e PCM Adicionado BIOS para gravação e reprodução.

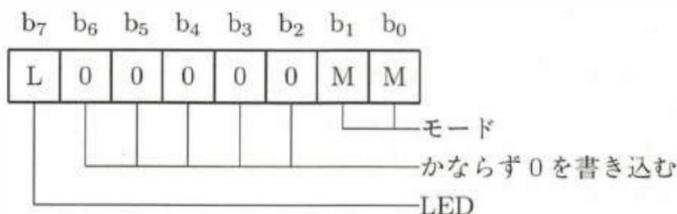
Aqui explicaremos o nome do BIOS Endereço de entrada (número da casa), função e cada (rótulo) e os registros nessa ordem. Os símbolos usados para representar as funções do BIOS são como e Está aqui. Em primeiro lugar, os registros físicos são registros cujos valores devem ser definidos antes de chamar o BIOS. O problema é Nos registros para os quais o BIOS retorna valores, há momentos em que o BIOS grava valores sem sentido, ou seja, o original Representa um registro cujo conteúdo será destruído. IYH também se refere aos principais compradores do registro IY. O conteúdo da parte inferior é ignorado.

#### Endereço CHGCPU 01soH

Defina o modo como segue nos

bits 1 e 0 do registro do país A onde você deseja alternar a CPU atual. Por aqui O "R8 00DRAM" transfere o conteúdo da ROM do BIOS para a DRAM.

Estou falando sobre o modo de mudança.



モード	
00	Z80
01	R800 ROM
10	R800 DRAM

Além disso, se o bit 7 do registrador A for 1, qual CPU está em execução?  
 O LED que indica o status atual mudará. Por outro lado, se o bit 7 do registrador A for 0, então  
 A CPU é trocada, mas o LED não muda.

Não sólido

Figura AF

O conteúdo dos registros anteriores é alterado, exceto AF e R.

Ele será transferido como está para a CPU subsequente. Além disso, depois de mudar e interromper  
 é permitido. Veja abaixo detalhes sobre precauções para troca de CPU.

explique claramente .

Endereço GETCPU 01s3H

Verifique a CPU em execução .

Não sólido

Dependendo de qual CPU está com problemas , os seguintes valores são retornados no registro A:

0	Z80
1	R800 ROM
2	RAM R800D

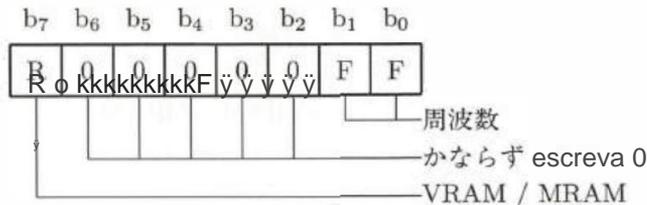
vezes F

Certifique-se de que o hardware seja turbo R usando o método explicado posteriormente .  
 , você precisa chamar esse BIOS.

Endereço PCMPPLY 01s6H

Reproduza som PCM .

Difícil A



EHL (endereço de dados)

DBC (comprimento dos dados)

Se o bit 7 do registro A for 1, é para RAM de vídeo, e se for 0, é para principal.

Os dados da fonte de som PCM são colocados na RAM. Observe que os dados são armazenados na RAM de vídeo.

Os valores dos registros D e E têm significado apenas em certos casos.

Define a frequência de amostragem com os bits 1 e 0 do registro A.

No entanto, a 15,75 kHz, o turbo R opera no modo R800 DRAM.

Isso só pode ser especificado se houver.

.. 1	5,75 quilohertz
01	7,875 quilohertz
10	5,25 quilohertz
11	3,9375 quilohertz

país carrega bandeira

0 Conclusão bem sucedida

1 rescisão anormal

A (Causa da anormalidade)

1 Erro de especificação de frequência

2 Barra média pela tecla STOP

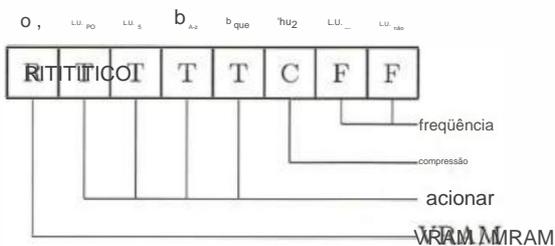
EHL (endereço de interrupção)

Figura tudo

Endereço PCMREC 01s9H

Grave o som do seu próprio PCM.

Difícil A



EHL (endereço de dados)

DBC (comprimento dos dados)

O método para definir os bits 7, 1 e 0 do registro A é explicado em PCMPLY.

O mesmo que. Os bits 6 a 3 do registro A são o "nível de disparo".

OK, especifique o volume do som que aciona a gravação. Se este valor for 0, a gravação começará imediatamente.

Além disso, se o bit 2 do registrador A for 1, os dados gravados serão compactados. Se for, não pode ser compactado.

problemas para carregar a bandeira

0 Conclusão bem sucedida

1 rescisão anormal

A (Causa da anormalidade) 1

Erro de especificação de frequência

2 Interrupção com tecla STOP

EHL (endereço de interrupção)

todas as vezes

## 1.2.4 Sobre BIOS modificado e removido

O BIOS alterado ou removido no turbo R é mostrado na Tabela 1.3. para cada um Vou explicar brevemente.

Primeiro de tudo, o turboR não possui mais uma interface de fita cassete, então "TAPION" "TAP IN" "TAP I OF" no BIOS, "TAPOON", "TAPOUT", Chamar "TAPOOF" define o sinalizador de transporte e retorna como um erro. Além disso, "STMOTR" não está mais disponível e, mesmo que você o chame, ele retornará sem nenhum erro. Além disso, o BIOS do paddle e da light pen foram removidos para adicionar novos recursos sem alterar o tamanho da ROM principal.

Quando você chama "GTPDL" no BIOS, o registro A Sempre retorna com um O. Da mesma forma, com "GTPAD" ou "NEWPAD", Mesmo que você coloque um número direto de 8 a 11 no registro A para determinar a caneta óptica, a chamada sempre retornará com um O no registro A.

Quanto ao BIOS alterado, "ROM version ID" é usado para saber a versão do MSX em uso. Isso pode ser visto no conteúdo do endereço 002DH na ROM principal, que foi alterado para 03H no caso do turboR. Se você estiver desenvolvendo um programa para turboR, primeiro certifique-se de que o valor deste endereço seja 03H ou superior, e caso contrário, execute-o como um programa para MSX2, ou exiba uma mensagem de erro e aborte-o.

Observe que um programa que só roda quando o conteúdo do endereço 002DH é 03H não funcionará mais quando o MSX for atualizado no futuro, então é necessário criar um programa que rode apenas quando o conteúdo do endereço 002DH for 03H ou superior. Em geral, em relação às versões de hardware e sistema operacional,

Tabela 1.3: Lista de alterações do BIOS e BASIC no MSX turbo R

Adicionada entrada de BIOS	
CHGCPU	0180H
GETCPU	0183H
PCMPPLY	0186H
Gravação PCM	0189H
Entradas de BIOS modificadas	
ID da versão da ROM	002DH
Entrada de BIOS excluída	
GTDL	OODEH
TAPIÃO	OOE1H
PIN TA	OOE4H
TAPIOF	OOE7H
TAPÃO	OOEAH
RETIRE	OOEDH
TAPOOF	OOF0H
STMOTR	OOF3H
GTPAD	OODBH
NOVOPAD	SUB01ADH

Declaração adicionada
LIGUE PARA CMREC
LIGUE PARA CMPLAY
PAUSA DE CHAMADA
= declaração modificada
<b>COPY</b>
Declaração excluída
CLO ANÚNCIO
CSAVE
MOTOR

Se você obtiver um número de versão maior ou igual ao número de versão necessário, o software será executado.

Por favor me ajude a fazer isso. Vamos fazer um programa .

Isso realmente aconteceu no passado, mas a verificação da versão do MSX

Como resultado de um erro, alguns programas MSX2 não funcionam no MSX2+, e alguns que possuem função de aprendizagem.

Se você combiná-lo com o MSX-JE, você acabará com aplicações que não funcionam. que Para evitar isso, lembre-se que “Se for 03H ou superior, faça funcionar”.

Sim.

Além disso, como o BIOS, adições, alterações e exclusões podem ser feitas nas funções BASIC do turboR.

Houve. Para isso, consulte a Tabela 1.3 e o manual BASIC que acompanha sua máquina.

Quero isso.

### 1.2.5 Notas sobre desenvolvimento de aplicações

No MSX turbo R, o R800 nem sempre opera sem peso. fora

3 esperas ao acessar slots internos, 2 esperas ao acessar a ROM interna.

É necessária uma espera quando a DRAM interna causa uma quebra de página.

é. Portanto, para acelerar o programa, estes pesos devem ser reduzidos ao máximo.

Você tem que trabalhar pensando no que vai acontecer. Aqui estão três coisas que você deve ter em mente.

Tentei impedir, então vamos lembrar. .

O primeiro passo é transferir o próprio programa para a RAM e depois executá-lo. Rebanho  
 Não há problema com o software fornecido como cópia porque ele roda naturalmente na RAM.  
 Porém, você deve ter cuidado com o programa que acompanha o cartucho ROM no slot.  
 Ao transferir apenas as peças necessárias para a RAM e depois executá-las, é possível acelerar significativamente o processo.  
 Tornar-se.

Também é importante codificar de uma forma que não cause quebras de página. Por R800,  
 Este recurso é útil porque possui um caminho dedicado para suportar o acesso à página DRAM.  
 Vamos aproveitar ao máximo. Especificamente, nesse caso, apenas os 8 bits inferiores do endereço mudam.  
 Memória contínua, claro? ? OH ~? ? Memória na faixa de 256 bytes até FFH  
 É eficaz programar de forma que o acesso seja realizado.

A propósito, quando ocorre uma quebra de página, a memória é  
 Quando ocorre um acesso, ou seja, quando os 8 bits superiores do endereço mudam.  
 está ligando.

Eu escrevi brevemente sobre isso anteriormente, mas diferentemente do MSX2+, o turboR usa um programa  
 O tempo exato de execução de uma instrução não é conhecido na fase de codificação da RAM. A razão  
 Um exemplo disso são as quebras de página DRAM, que podem ocorrer em momentos imprevisíveis.  
 E, diferentemente do Z80, a atualização da DRAM é realizada de forma assíncrona com a execução das instruções.  
 Isso ocorre porque há um silêncio.

Além disso, para criar um programa que rode tanto no turboR quanto no MSX2+, é necessário usar software.  
 A cronometragem por loops de software não é recomendada. Portanto, o turboR foi recentemente equipado  
 com um temporizador de sistema que conta a cada 3.911 microssegundos.  
 Estava feito. De agora em diante, usarei este cronômetro do sistema para definir o tempo.  
 vamos .

## 1.2.6 Exemplo de programa para troca de CPUs

Listagem 1.2 alterna a CPU “CHGCPU.COM” のソースリストだ。turbo  
 Em R, quando o DOS2 está em execução,

```
CHGCPU0
```

No modo Z80,

```
CHGCPU 1
```

O modo ROM do R800 é

```
CHGCPU2
```

O modo DRAM do R800 é selecionado respectivamente. Explique o conteúdo do programa e , A partir do endereço 5DH na área de trabalho do DOS (área FCB padrão, para ser exato), execute o comando. Obtenha o primeiro caractere do I-ésimo parâmetro e defina o valor do registro A de acordo. Então Em seguida, chama o BIOS "CHGCPU" no endereço 180H da ROM principal.

Além disso, verifique o número da versão do DOS para tornar o programa mais prático. O processo de verificação também é adicionado. Especificamente, primeiro, o conteúdo do endereço 2DH na ROM principal é 03H. Confirme se o acima é verdadeiro, ou seja, turbo R, e execute a chamada de sistema DOS 6FH. usando , Certifique-se de que o número da versão do kernel DOS seja 2 ou superior. .

### Listagem 1.2 (CHGCPU.Z80)

```

; Z80
EQU RDSLTL      0000CH      ; leitura entre slots ; chamada
CALLT EQU      0001CH      entre slots : número do slot
EXPTBL EQU      0FCC1H      da ROM principal
;
;
Eu      a, (EXPTBL)      ;
      haha, 2dh          ; endereço para ler ;
vou ligar RDSLTL3
CP
Jr.     não, TURBOR
Eu      de, MSG_NDTR
      c,9                ; y y -STROUT
vou ligar
primeiro 5.              ; retornar ao DOS
TURBOR:
Eu      c,6fh 5          ; _DOSVER
ligaria
para você a,b 2          ; versão do kernel DOS
CP
Jr.     c, NOTDOS2
ld      uma,d            ; versão do MSXDOSSYS.
      2
CP Jr.   c. NOTDOS2
;
ld      uma, (005ch+1) ' ;parâmetro de comando
sub     0 '              ;O:Z80 , 1: R800ROM ; , 2: R800RAM
ret     c                aborta se parâmetro < '0'
cp      3                ;
ret     nc               ; abortar se '3' <= sinalizador de LED de
ou      80h              alteração do conjunto de parâmetros
veho    ix, 180h         ; endereço do CHGCPU ; slot
      sim, (EXPTBL-1)    da ROM principal ;
      CALSLT .
;
engenheiro
ligaria primeiro
■
NOTDDS2.ldld
      de, MSG_NDTDOS2
      c,9                ■ -STROUT

```

```

chamar
primeiro 5 . ; retornar ao DOS
;
MSG_NOTR:
D. B. 'MSX turbo R' , Odh, Oah, 'não é
MSG NOTDOS2:
D. B. 'não MSX-DOS 2' , Odh, Oah , 'y$'
FIM

```

Da mesma forma, a Listagem 1.3 abaixo engana um programa para MSX2 para rodar no modo R800. Esta é a lista de fontes de "GAMEBOOT.COM". Este programa começa com DOS2. Se você tentar iniciar um programa em outro disco com R800 selecionado.

Algo para seguir em frente. Ou seja, existem programas (jogos, etc.) que não incluem o sistema DOS2. etc.) para operá-los com força no modo R800.

Deixe-me explicar brevemente o programa. , Primeiro, exiba uma mensagem na tela e depois Aguarde até que seja substituído. A seguir, leia o setor de inicialização do disco substituído e faça acontecer. O ambiente então é tal que o setor de inicialização é codificado uma segunda vez da maneira normal. A página 1 tem o sinalizador de transporte do DOS ROM , Outras páginas são RAM , definido.

Além disso, uma função para lembrar o ponteiro para o programa de tratamento de erros. Defina a área de trabalho do DOS (endereço F323H) para o registro HL e a página 1 para a RAM. Altere o endereço do programa (F368H) para o DOS ROM no registro DE. são definidos respectivamente. .

## Listagem 1.3 (GAMEBOOT.Z80)

```

.z80

_COiili equ 01h
_strout equ 09h
_setdta equ 1h
_rdabs equ 2h

dos equi 0005h
enaslt equi 0024h

não é o equi De340h
primeiro mestre equi De348h

ld sp, (6)
ld de, prompt c, ; imprimir mensagem de alerta
ld _strout dos
ligar
c _con1n dos ; aguarde a digitação
para ld
de, Oc000h ; leia o setor de inicialização em Oc000h
c _setdta dos
ligar de, O
1, 0 h, 1 ; setor lógico 0 ; unidade A ::

para ld ld ligar para ld ld ld
ld c _rdabs dos
ligar h, 40h
a,
(mestre) enaslt
para ld, hl, Of323h
ld ligar para ld de, Of368h
ld
xor um
ld (não primeiro), um
scf
jp Oc01eh

incitar:
----- 'Insira o disco do jogo na unidade A: ' ÿ d pressione ' ,Odh,Oah
----- Muiy key s'

fim

```

## 1.3 Como usar o PCM em seus limites

Um novo recurso adicionado ao turbo R é o PC M. Por se tratar de uma função preparada com muito esforço, É da natureza humana querer obter o máximo desempenho de uma máquina. Então, do B ASIC à máquina Apresentaremos como utilizar o PCM, incluindo uso especial usando interrupções de linha de varredura horizontal.

### 1.3.1 Como usar o Básico H/H/BASIC

Primeiro, vamos apresentar o básico usando BASIC.

Para começar, o PCM converte a entrada de áudio de um microfone etc. em dados digitais e os armazena na memória. Ele pode ser armazenado na memória e reproduzido à vontade.

No caso do turbo R, os dados PCM são armazenados na RAM principal ou na RAM de vídeo. sa As taxas de amostragem são 15,75 kHz, 7,875 kHz, 5,25 kHz, 3,9375 kHz.

Existem quatro tipos de quilohertz para escolher. Quanto maior o valor, melhor a qualidade. Sangh. Isso significa que um anel pode ser feito.

Ao usar o PCM do BASIC, você só precisa se lembrar de dois comandos. Como usar Eu resumi aqui, então use-o como referência. Basicamente, ele executa essas instruções.

Isso permite a gravação e reprodução de PCM. No entanto, o endereço inicial e o endereço final para armazenamento de dados são Você precisa ter muito cuidado ao definir o endereço.

Em primeiro lugar, o B ASIC “CLEAR” 命令で PCM データ用のメモリー領域を確保 Se você não controlar isso, definitivamente ficará fora de controle. Por exemplo, do endereço COOO H para DOOOH. Ao usar dados PC M,

```
LIMPAR 200, &HCOOO
```

Vamos fazer assim. Por enquanto, a Listagem 1.4 mostra um programa de exemplo simples. Eu coloquei lá em cima, então você pode querer inseri-lo e brincar com ele.

Obviamente, se você usar RAM de vídeo para dados PCM, poderá colocar os dados em qualquer endereço. Não há necessidade de se preocupar com o endereço inicial ou final. E um bidê No caso do ORAM, os dados PCM podem ser verificados visualmente. primeiro

TELA 8

Se você definir o modo de tela e depois gravar PCM, os dados aparecerão na tela.

Pode ser interessante vê-lo exibido em cascata.

O método básico de gravação e reprodução do PC M é adequado, desde que você preste atenção aos pontos acima. além disso Ao alterar a taxa de amostragem de reprodução, você pode reproduzir em quatro velocidades. Ru. No entanto, o problema ocorre ao jogar PC M. turbo R pega.

Infelizmente, não quero jogar nada enquanto jogo PCM porque vou acabar ficando preso.

Eu não posso fazer isso.

## Listagem 1.4 (PCMLBAS)

```
10 CLEAR100 ,&H9000
20 IMPRIMIR "Imakara Lok em Simas." ;
30 A$=INPUT$(1): PRINT 40
_PCMREC yyH9000 50 ,&HCFFF , Ó)
PRINT ``Sai Sei Shi7s." ;
60 A$= INPUT$(1): PRINT 70
_PCMPLAY yyH9000 80 GOTO ,&HCFFF , Ó)
20
```

### 1.3.2 Instruções BASIC relacionadas ao PCM

#### CALL IL DE fL M DH E" fL



- Gravação na RAM principal ou RAM de vídeo.

CALL PCMREC (@Endereço inicial, Endereço final, Taxa de amostragem [, [Nível de disparo]  
jré! ? chave de compressão] [, SJ)

- Gravação em uma variável de array.

CALL PCMREC(nome da variável da matriz, [comprimento], taxa de amostragem [, [Acione Le Benore! ?  
interruptor de compressão])

Definir a taxa de amostragem (Taxa de amostragem de valor especificado)	
.	15,7500 KHZ
1	7,8750 KHZ
2	5,2500 KHZ
3	3,9375 KHZ

No nível de disparo, quando a gravação começa Defina o nível de entrada. Os valores variam de 0 a 127.

A gravação começará quando o nível de entrada exceder este valor.

Se definido como 0 ou omitido, a gravação começa imediatamente.

A configuração da chave de compressão é 1 para suprimir o silêncio.

Se definido como 0, ou omitido, nenhuma compactação será executada.

## Formato CALL PCM

### PLAY

- Reprodução da RAM principal ou RAM de vídeo

CALL PCMPPLAY (@Endereço inicial, Endereço final, Taxa de amostragem [, SJ)

yy Reprodução de variáveis de array

CALL PCPLAY(nome da variável de array, [comprimento], taxa de amostragem)

Quando PCMREC e PCMPPLAY não estão no modo de alta velocidade é executado temporariamente no modo de alta velocidade e, ao sair, retorna ao seu estado original. Ele também retorna para 15,75KHz no modo ROM do R80 0.

Se for especificado, a linha de gravação ou

reprodução. Quando a linha de gravação ou reprodução do programa termina, o resultado será interrompido. O formato dos dados PCM é de até 1.255. Nos dados, 0 é especial. Continuação <1 byte vezes especificadas

Emite nível 0 (127) por apenas alguns minutos •

### 1.3.3 Reproduza o som BEEP com PCM!

Ao executar um programa BASIC, pressione as teclas Mamego e Iji ao mesmo tempo.

Você sabe que quando interrompe um programa, você ouve um pip. (LIST) é sabido. "LIST" não exibir a lista com o comando interrompe-la com "Ono" @, você ouvirá o mesmo som de bipe. Você pode alterar o som usando o comando "SETBEEP" do BASIC, mas existem quatro tipos disponíveis. faltou , impacto.

Então, o que acontece quando você reproduz esse som BEEP em um PCM? Defina uma linha estranha Se você ouvir, o MSX falará toda vez que você fizer alguma coisa, então pode ser bem barulhento e divertido.

não tenho.

É por isso que postei a Listagem 1.5. Quando você executa o programa, o som BEE P é Pÿ M Você será capaz de fazer um som com Claro, é exclusivo do turbo R. Não é tanto tempo, então Por favor insira 9 comprimento.

Este programa está escrito na página 1 (endereços 4000H a 60FFH) da RAM principal. Após executar o programa, utilize o comando "C LEAR" para movê-lo acima da área do usuário. Não importa se o limite é maior que o endereço BOOO H. No , Itens relacionados ao disco de memória não podem ser usados. entanto, por favor não diga acidentalmente "C ALL MEMINI". Então, BÁSICO Ao usar o comando "BEEP",

IMPRIMIR CHR\$ (7)

Se você não usar, o som do BEEP não se tornará PC M. Tome cuidado.

Explique como usar o programa.

Conjunto de BEEP de 1 PCM

Depois disso, o som do BEEP torna-se PC M. Uma vez executadas, as configurações permanecerão até que a energia seja desligada. válido. Você também pode alterar as configurações da instrução C LEAR.

2 PC f reinicialização do BEEP

Retorne o som do BEEP ao seu estado original. Defina para DOS ou D OS2 com "C ALL SY STEM"

Quando isso acontecer, certifique-se de executar este comando.

3 PC vergonha de reprodução de dados

Reproduz os dados do PC M atualmente definidos. Use-o para confirmação.

4 gravação de dados PCM

Grave dados do PC M a 15,75 quilohertz. Ao gravar, endereço BOOO H-C FFFH

Usa memória até.

5 CARGA de dados PCM

Lê dados PC M salvos no formato BSAVE com a extensão ".PC M".

6 Dados PCM SALVAR

Grave os dados PC M gravados usando "Gravação de dados PC M" em um disco.

FIM

Saia do programa. claro , [Não me importo com feijão @ ÿ.

Observe que uma mensagem simples será exibida na tela, portanto consulte-a.

### Listagem 1.5 (PCM2.BAS)

```

10 SCREEN=WIDTH40:DEFINT AZ 20 CLEAR100,&HBOOO
30
DEFUSR=&HD800:DEFUSR1=&HD806:DEFUSR2=&HD803 40 FOR I=Aguilha. 800 TD &
HD87F 50 LEIA A$: POKE I, VAL("&H"&A$):
PRÓXIMO
100 IMPRESSÃO
110 PRINT"1) PCM BEEP Definir 120 "
PRINT"2) PCM BEEP Redefinir H
130 IMPRIMIR"3) DADOS PCM DIZ DIZ"
140 IMPRIMIR"4) BLOQUEIO DE DADOS PCM LIGADO"
150 IMPRIMIR"5) CARGA DE DADOS PCM"
160 IMPRIMIR"6) SALVAR DADOS PCM"
170 IMPRIMIR"0) FIM"
180 PRINT" .... APERTE TECLA 0-6y" ; 190
A$=INPUT$(1) : I=ASC(A$)-ASC("O")+1 200 SE I>0 E I<8 ENTÃO
ELSE190 210 ON I GOTO 230.240.310.220.340.390.430 220
PRINTCHR$(7) : GOTO 190 230 GOSUB 470:END 240 GOSUB 470: I=USR(O) :
I=USR(O)

250 IMPRESSÃO"PCM BEEP
260 PRINT"DOS 7TAHA DOS2 WO TSUKOU PARA KIHA KANARAZ PCM BEEP
Não é fácil de configurar. "
270 PRINT"PCM BEEP DATA Página 1 (4100H 60FFH) PERTO
truta. "
280 IMPRIMIR"CLEAR MEIL TE' BOOOH CONFIGURAÇÃO MOKA
Meu 7 Senga, LIGUE PARA MEMINI
Eu não sei o que fazer. "

```

```

290 PRINT"Nao, BEEP Mail I Wo Show Sunore To Ki Ha PRINT CHR$(7) Wo Tsukatteku Dasai."

300 END

310 GOSUB 470:POKE &HFDA4,&HC9 320
PRINT"PCM BEEP reset."
330 GOTO 100 340
GOSUB 470 350
PRINT"PCM Lokuon wo Hajime Mass. (HIT QUALQUER TECLA!)"; )

370 IMPRIMIR"Camisetas Lokuon Shu Ryo."
380 GOTO 100 390
GOSUB 470 400
IMPRIMIR"CARREGAR dados PCM"
410 ENTRADA" NOME DO ARQUIVO (8 modi)=" ;A$
420 BLOAD A$+".PCM" : I=USR(O):GOTO 100 430 GOSUB
470 440 PRINT"PCM
data SAVE"
450 ENTRADA" NOME DO ARQUIVO (8 modi)=" ;A$
460 I=USR2(0) :BSAVE A$+".PCM" , &HBOOO,&HCFFF:GOTO 100 470 PRINT CHR$(
(I+47) :PRINT:PRINT:RETURN 480 DADOS C3,4D,D8,C3,5F,
D8,CD,6D 490 DADOS D8,3A,42,F3,32,2A,D8,21 500
DADOS 2E,D8,11 ,00,40,01 ,00,01 510 DADOS
ED,BO,CD ,76,D8 ,21 ,29,D8 520 DADOS 11,A4,FD
,01,05,00 ,ED,BO 530 DADOS C9,F7 ,00,00,40,C9,FE,07
540 DADOS co,01 ,oo,20, 21 ,oo,41,3E 550 DADOS
03,D3,A5 ,F3,DB,A4,D6,01 560 DADOS
38,FA,7E,D3,A4,23,0B,79 570 DADOS B0,
20,F1 ,FB ,C9,CD,6D,D8 580 DADOS 21,00,B0,
11,00,41 ,01 ,00

```

### 1.3.4 HH PCM avançado em linguagem de máquina!

Ao usar PCM em linguagem de máquina, a maneira mais rápida é usar BI OS. amostra Configurações como ótimo e nível de disparo são quase iguais às do BASIC, então não há problema. Provavelmente não.

Como esta é uma versão avançada, você pode gravar PCM sem usar este BIOS. Apresentarei dois programas que podem ser reproduzidos.

Ao usar o BIOS, a taxa de amostragem é 15,75 kHz, 7,875 kHz, Existem apenas quatro tipos para escolher: 5,25 quilohertz e 3,9375 quilohertz. a cada 63,5 microssegundos Como é utilizado um contador cujo valor muda, quatro ou mais taxas de amostragem podem ser definidas. Não posso. Defina esse contador como uma variável de sistema cujo valor muda a cada 3,91 microssegundos. O programa apresentado aqui é aquele que assumi do Mutimer.

Primeiro, vamos explicar como usar o programa de gravação. O registro HL contém dados PC M.

O endereço inicial da memória que armazena os dados é armazenado no registrador BC, e o tamanho dos dados a serem gravados é armazenado no registrador BC.

Defina o tamanho de cada um. E no registro E, qual é o temporizador do sistema?

Defina se deseja incluir um peso para a contagem. 16, aproximadamente 15,75 quilohers

Eu acho que corresponde a tsu.

Reprodução. O mesmo vale para o programa. Endereço inicial dos dados PCM a serem reproduzidos no registro HL endereço, o tamanho dos dados no registro BC e o contador de peso no registro E.

Basta definir o número de entradas.

No meio da lista de programas de gravação PCM,

OEDH, 70H

Há algo estranho nisso, isso é

IN (HL), (C)

É sobre um comando. Leia o valor da porta do registrador C e aplique-o apenas aos flags

Este é um comando exclusivo do R800.

Independentemente do princípio do programa, vamos tentar usá-lo de qualquer maneira. alterar o valor do registro E Ao fazer isso, você poderá aproveitar as várias mudanças no som.

## Listagem 1.6 (PCMREC.MAC)

```
PMDAC EQU OA4H
PMCNT EQU OA4H
PMCNTL EQU OA5H
PMSTAT EQU OA5H
SISTML EQU OE6H ; porta do temporizador do sistema
```

REC:

```
L.D. A,00001100B
FORA (PMCNTL),A ; MODO A/D
D.I.
XOR A
FORA (SYSTML),A ; redefinir o temporizador
```

REC1:

```
EM A, (SISTML)
CP- E
JR C, REC1 ; espere
XOR A
FORA (SYSTML),A ; redefinir o temporizador
```

PRESSIONE BC

```
L.D. A,00011100B
FORA (PMCNTL),A ;
L.D. A,80H
L.D. C, PMSTAT
FORA (PMDAC),A ;
```

```
DEFB OEDH,70H ; IN (HL) , (C)
JP M,RECAD0
E 01111111B
RECAD0:
OU 010000008

FORA (PMDAC),A
DEFB OEDH,70H
JP M,RECAD1
E 10111111B
RECAD1:
OU 00100000B

FORA (PMDAC),A
DEFB OEDH,70H
JP M, RECAD2
E 11011111B
RECAD2:
OU 00010000B

FORA (PMDAC),A
DEFB OEDH,70H
JP M, RECAD3
E 11101111B
RECAD3:
OU 00001000B

FORA (PMDAC),A
DEFB OEDH,70H
JP M, RECAD4
E 11110111B
RECAD4:
OU 00000100B

FORA (PMDAC),A
DEFB OEDH,70H
JP M, RECAD5
E 11111011B
RECAD5:
OU 00000010B

FORA (PMDAC),A
DEFB OEDH,70H
JP M, RECAD6
E 11111101B
RECAD6:
OU 00000001B

FORA (PMDAC),A
DEFB OEDH,70H
JPM , RECAD7
E 11111110B
RECAD7:
OU 0000000B
```

```

LD (HL),A
LD A, 000011 OOB
FORA (PMCNTL),A

POP BC
INC.HL
DEZEMBRO AC
LD A,C
ORB ; fim dos dados ?
JR Nova Zelândia , REC1

LD A , 00000011B
FORA (PMCNTL),A;MODO D/A
E.I.
RET

FIM

```

## Listagem 1.7 (PCMPLAY.MAC)

```

PMDAC EQU OA4H
PMCNT EQU OA4H
PMCNTL EQU OA5H
PMSTAT EQU OA5H
SISTML EQU OE6H ; porta do temporizador do sistema

JOGAR :
LD A , 00000011b
FORA (PMCNTL),A ; MODO D/A
D.I.
XOR A
FORA (SYSTML),A ; redefinir o temporizador

JOGAR :
EM A, (SYS bateu L)
CP- E
JR C , JOGAR ; espere
XOR A
FORA (SYSTML),A ; redefinir o temporizador

LD A, (HL)
SAÍDA (PMDAC),A ; reproduzir 1 byte
INC.HL
DEZEMBRO AC
LD A,C
ORB ; fim dos dados ?
JR Nova Zelândia , JOGAR
E.I.
RET

FIM

```

Tabela 1.4: Portas de E/S para PC M

番地	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Eu bit0
Escreva OASH .		■	■		SMPL SEL FILT	MUDO	ADDA	
OASH Leia COMP .			■		SMPL SEL FILT	MUTE	BUFF DA3	DA2 DAI .
OA4H Grava DA7			DA6	DAS	DA4			DAO
OA4H Leia .		■	■	■		■	CTI	CTO

- **ADDA (BUFF):** modo buffer  
Especifique a saída do conversor D/A  
Ru. 0 em D/A (buffer único duf),  
Defina como 1 (buffer único) durante A/D.  
vamos. Além disso, ao reiniciar, é ruim. Lubat  
Está em estado de fogo.
- **Controle de silenciamento MUTE**  
Saída de áudio para todo o sistema ativada  
ou desligue-o.  
0 Saída de áudio desligada (na reinicialização)  
1 Saída de áudio ligada
- **Sinal de entrada do circuito de retenção de amostra FILT**  
Selecione o número  
Entra no circuito sample e hold durante A/D  
O sinal de entrada é o sinal de saída do filtro  
ou como um sinal de referência.  
0 . 0 para sinal de referência, 1 para saída de filtro  
sinal de força torna-se um som. Ó quando reiniciado
- **SEL:** Seleção do sinal de entrada do filtro  
A entrada de sinal para o filtro passa-baixa é  
Torne-o o sinal de saída do conversor D/A  
Ou use o sinal de saída do amplificador de microfone?  
Selecione. Saída do conversor D/A em 0  
Som do sinal de saída do amplificador de microfone no sinal de alimentação 1.
- **SMPL** · Sinal de retenção de amostra  
Amostra ou espera o sinal de entrada  
Escolha se deseja fazer isso.  
0 amostras (na redefinição)  
1 espera
- **COMP:** Sinal de saída do comparador  
Amostra e retém sinal de saída e D/A  
Compare com o sinal de saída do conversor.  
0 Saída D/A > saída de retenção de amostra  
1 saída D/A < saída de retenção de amostra
- **Dados de saída DA7~DAO .D/A**  
Usado ao reproduzir dados PCM.  
produza os dados desejados aqui  
Você pode reproduzir o som do PC.  
O formato dos dados é binário absoluto.  
Em Lee, 127 corresponde ao nível 0 0
- **CTI, dados do contador CTO**  
Conta a cada 63,5 microssegundos  
ser feito. D/A às vezes conta  
dados gravados no endereço OA4H.  
Os dados são gerados repetidamente. Também OA4H  
Quando os dados são gravados, o contador é zerado.  
Criado.



2<sup>a</sup> Class | SLOT  
第 2 章

# SLOT



Este capítulo contém os artigos "MSX2 + Technical Expedition" nas edições de fevereiro de 1989 e março de 1989 da MSX Magazine, e "Technical Analysis" na edição de novembro de 1990. Esta é uma versão reeditada.

## 2.1 O que é um slot?

O orifício para inserir um cartucho no MSX é o "slot da coluna".

”。でもス

O que Lott se refere é a capacidade de gerenciar a memória do MSX. Neste capítulo, explicaremos os slots mais importantes e difíceis.

### 2.1.1 Como a CPU e a memória estão conectadas?

Os componentes mais importantes que compõem um computador são a CPU e a memória.

CPU é uma abreviatura de "Unidade Central de Processamento", que é um dispositivo que gerencia todo o computador e realiza cálculos. A memória, por outro lado, refere-se a um dispositivo semelhante a um bloco de notas que armazena informações manipuladas pela CPU. As

informações manipuladas por um computador são representadas por um número binário, que é uma combinação dos números 0 e 1.

Como você sabe, isso vai acontecer. Um dígito desse número binário é chamado de "bit" (byte). ”、8桁を“バイト

Além disso, em listas de programas, etc., se os números binários forem expressos como estão, o número de dígitos aumentará, portanto, os números binários de 4 bits são expressos como "números hexadecimais" usando os caracteres de 0 a 9 e de A a F. . também é frequentemente usado.

Não me entenda mal, eu quero isso. , No mundo da informática, a unidade "quilômetro" significa 1.024 vezes, não 1.000 vezes. Por exemplo, 64 kilobytes de memória equivalem a  $64 \times 1024 = 65536$  bytes de memória. Isso também é  $65.536 \times 8 = 524.288$  bits. Para gerenciar essa memória, muitos microcomputadores

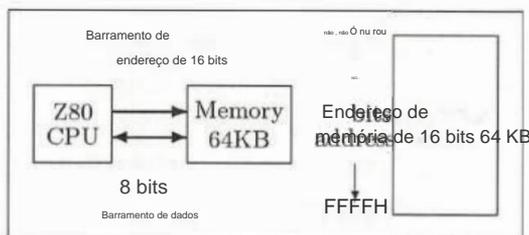
usam um único

Cada item de memória é numerado. Isso é chamado de "número da casa" ou "endereço". Este é frequentemente o caso em programas em linguagem de máquina que dizem algo como "O endereço inicial de execução é 8000H".

### 2.1.2 Veja os componentes internos da CPU Z8 0 de 8 bits

A CPU e a memória são conectadas por um "caminho de endereço" e um "caminho de dados" conforme mostrado na Figura 2.1. Um caminho de endereço é um fio elétrico que envia um sinal da CPU para a memória que especifica o endereço de memória que a CPU deseja ler ou escrever. Um caminho de dados é um fio elétrico que comunica o conteúdo da memória. Observe que enquanto o primeiro é um caminho unidirecional, da CPU para a memória, o último é um caminho bidirecional.

Figura 2.1: Memória PU Z80 C



A CPU Z80 usada no MSX é basicamente para, um cartão de memória de 64KB.

Um 16 bits

Sinal do caminho do endereço 0000H ~ 1 barra na memória até FFFFH

Isso significa que você pode especificar o site. Vejo você de novo Caminho de dados para troca de dados é de 8 bits.

"Z80", CPU do MS X antes do turbo R, era de 8 bits (fisicamente 8 fios)

Possui um caminho de dados de 16 bits e um caminho de endereço de 16 bits. Isso resultou em 64 kilobytes

Isso significa que você pode ler e gravar na memória do seu computador um byte de cada vez. Este tipo de CPU pode ser usado como um "8 bits CPU", 8 ビット CPU が組み込まれたコンピューターを "8 ビットコンピューター"

É chamado. É por isso que o MS X é um computador de 8 bits.

Agora, para explicar especificamente sobre o caminho do endereço, podemos especificar um caminho de endereço de 16 bits.

O endereço de memória que pode ser especificado é um número binário.

OOOOOOOOOOOOOOOO

De (B é um símbolo que representa um número binário)

11111 111111111111 libras

para. Isso é expresso em números decimais de 0 a 65535 e em números hexadecimais de 0000H a FFFFH.

(H é um símbolo hexadecimal). Cada endereço contém 8 bits

(= 1 byte) e representa um valor de 0 a 255 em notação decimal. Também é expresso em bytes.

Se convertermos essas memórias em kilobytes únicos, obteremos 64 computadores de 8 bits.

Isso significa que 64 kilobytes de memória podem ser conectados ao computador.

Escrevi anteriormente que o MS X é um computador de 8 bits. Hoje em dia, visualização de computador de 16 bits

Os computadores (computadores equipados com CPUs de 16 bits) também estão se tornando populares. Nesse caso

O caminho de dados de 16 bits permite armazenar o dobro de informações de uma vez em comparação com um computador de 8 bits.

Sabe ler e escrever. Existem mais caminhos de endereço e mais memória pode ser conectada.

São muitas vantagens como No entanto, é relativamente caro porque a fiação é complicada.

Esta é a situação atual. Além disso, muitos computadores grandes suportam dados de 32 ou 64 bits.

Tenho tapas e um passe de endereço.

Observe que o MS X turbo R está equipado com a CPU R800 de 16 bits, mas

O caminho de dados permanece de 8 bits para permitir armazenamento subsequente.

### 2.1.3 Os tipos de memória variam dependendo da função

Existem muitos tipos de memória. Primeiro, eles são classificados por tipo de componente em "ROM" e "RAM". ROM (Read Only Memory) é um tipo de memória cujo conteúdo não pode ser reescrito, mas cujo conteúdo permanece mesmo quando a energia é desligada. Todo o software embutido no corpo do MSX (como o BASIC) e o software fornecido pelo cartucho são gravados nesta ROM. Além disso, a ROM Kanji que vem como padrão com o MSX2+ é uma ROM dedicada que contém as formas dos caracteres Kanji.

RAM (Random Access Memory) é uma memória cujo conteúdo pode ser reescrito livremente, mas cujo conteúdo é apagado quando a energia é desligada. É usado para armazenar temporariamente resultados de cálculos em um programa ou para carregar e executar um programa a partir de um disquete. Por exemplo, quando você digita um pequeno programa de uma revista M para jogar, ele também é armazenado nesta RAM.

"SRAM" é um tipo de RAM com baixo consumo de energia que pode ser usado em notebooks alimentados por bateria, processadores de texto portáteis e carrinhos de jogos com notebooks MSX e Famicom. Também é possível classificar a memória por uso, conforme mostrado na Figura 2.1.

A memória que está diretamente conectada à CPU é chamada de "memória principal" ou "memória principal". Em outras palavras, a ROM diferente da ROM Kanji no corpo do MSX e os 64 kilobytes de RAM principal são a memória principal do MSX.

Além disso, o MSX também possui uma memória chamada "vídeo RAM (VRAM)". RAM de vídeo é a RAM que armazena os gráficos e caracteres exibidos na tela da TV. Dependendo do modelo do computador, a RAM de vídeo pode ser conectada diretamente à CPU, mas no MSX ela é conectada através de uma parte dedicada chamada VDP (abreviação de processador de exibição de vídeo e RAM de vídeo).

As informações básicas até este ponto abrangem não apenas MSX, mas também o conhecimento mais básico sobre computadores. Eu acho que o livro introdutório ao BASIC que vem com o MSX tem informações detalhadas, então por favor consulte-o.

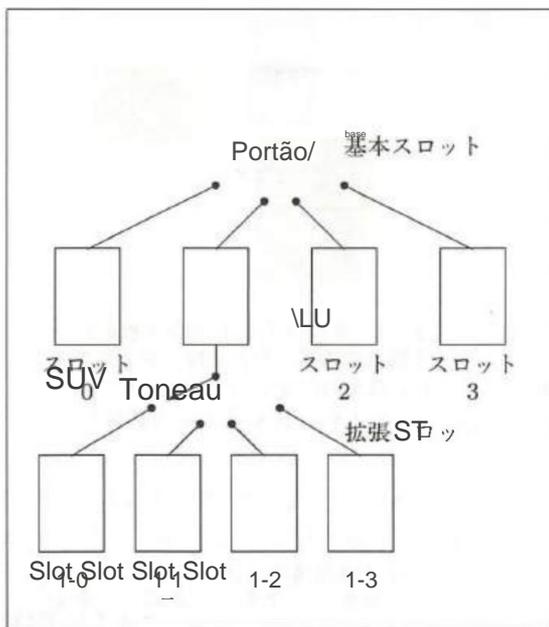
### 2.1.4 Quais são os slots no Ran1SX?

Como escrevi na introdução, a memória principal que pode ser conectada a uma CPU de 8 bits é de 64 kb. No entanto, isso só era verdade para os primeiros computadores de 8 bits. Hoje em dia, vários métodos estão disponíveis para anexar mais de 64 kilobytes de memória.

Para MSX, o método é chamado de "troca de slot". Conforme mostrado na Figura 2.2, ao preparar quatro conjuntos de 64 kilobytes de memória e alternar entre eles, você pode atingir um máximo de 256 kilobytes.

memória. Essas memórias são chamadas de "slots de base".  
Também é atribuído ao slot de cartucho fornecido na máquina.

Figura 2.2: Configuração do slot MSX (Parte 1)



Para MSX, 64kB  
Lida com mais de 1000 ienes de memória  
Use o , "Troca de slot  
método ``Utame-e".  
64 KB de memória  
Mude para até 256  
Lida com kilobytes de memória  
Uau! Esses quatro conjuntos de notas  
``Slot básico"  
cada um expandido a partir daqui  
4 slots como "slot de expansão"  
É chamado de "lote".

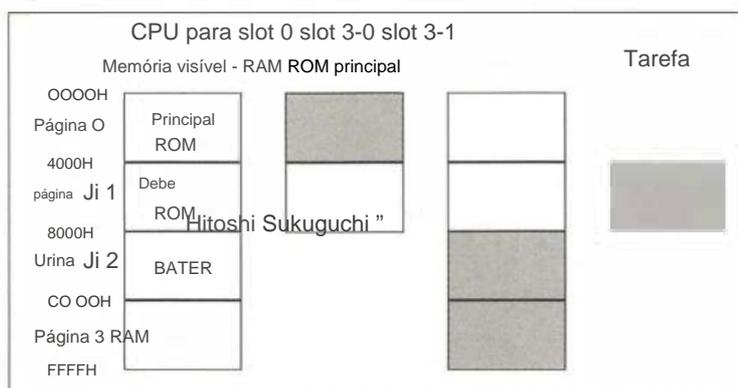
Além disso, você pode alternar entre quatro conjuntos de "slots de expansão" em vez de um slot básico.  
Também existe uma maneira. Nesse Um total de 16 conjuntos de 64 kilobytes de memória. Em outras palavras, o mais  
caso, até 1 megabyte (1.024 kilobytes) de memória pode ser conectada. No entanto, a expansão  
O slot não pode ser mais alongado.

Agora, ao trocar de slot, você pode lidar com mais de 64 kilobytes de memória.  
Isso é verdade, . É inconveniente que toda a memória seja trocada ao mesmo tempo.  
mas no MSX a idéia era dividir a memória em unidades individuais chamadas "páginas" e tratá-las.  
Está sendo

Os 16 kilobytes dos endereços de memória 0000 H a 3FFFH são paginados, 4000H-8000H  
os 16 kilobytes dos endereços 0 à página . Da mesma forma, página do endereço 8000H para BFFFH.  
1 são a página 2, os endereços C000 H a FFFFH são a página 3 e assim por diante. 16 KB cada  
Isso significa que você pode selecionar um slot diferente para cada bloco de páginas.

Por exemplo, quando instruções relacionadas à E/S do disco BASIC estão sendo processadas, a página  
As páginas 2 e 3 são trocadas para a RAM principal , A página 1 está no disco  
(veja a Figura 2.3).

Figura 23: . Configuração do slot M SX (parte 2)



O endereço de 64 kilobytes de memória é , Quatro páginas de 16 kilobytes cada. Está dividido em páginas e um slot pode ser selecionado para cada página. Por exemplo, ao inserir um disco Ao processar instruções para saída, a página O é a página ROM principal do BASIC. , A página 1 é o disco ROM, as páginas 2 e 3 são o RA 4 principal e assim por diante.

O intérprete A B ASIC é um programa que processa programas escritos em B ASIC.

Um programa escrito na ROM do M SX. 32 km em MSXI

Embora tenha sido incluído na ROM do computador, são 48 km no M SX2. Então, em M SX2

A ROM é dividida em duas partes, e a parte comum com o M SX1 é a "ROM principal" de 32 kilobytes.

Além disso, as funções expandidas com o MSX 2 são escritas em uma "sub ROM" de 16 kilobytes.

Além disso, você pode usar cartuchos de interface para disco interno M SX e unidades externas.

A câmera possui uma ROM integrada de 16 kilobytes. B Para lidar com entrada/saída de disco ASIC

Isto significa que o programa (DISK B ASIC) foi escrito. Então o disco

Durante a entrada/saída, a situação será conforme mostrado na Figura 2.3.

### 2.1.5 O segredo da escalabilidade do MSX está nos slots

Os slots do M SX são usados não apenas para adicionar memória, mas também para expandir a funcionalidade do M SX.

Também é usado para Conectar um cartucho de jogo também é igual a conectar um cartucho de modem.

Isso também é chamado de slot.

Conecte o cartucho de interface de disco ao M SX conforme descrito.

e os 16 kilobytes de ROM do cartucho são conectados ao slot. E então Di

Quando ocorre E/S de disco, a memória é automaticamente transferida para a interface do disco.

Isso significa que ele pode ser alternado para o slot ROM. Portanto, a interface do disco

Quer esteja integrado no próprio dispositivo ou conectado como um cartucho, o programa

Não há problema com a operação do sistema.

Além disso, quando uma interface de disco é conectada, o comando "C ALL FORMAT" é emitido. mas, Quando o cartucho de comunicação está conectado, o comando BASIC estendido "C ALL T ELC OM" é exibido. O comando se torna utilizável. Essas instruções estendidas são transportadas pela ROM dentro do cartucho. Será processado. Muitos computadores além do MSX exigem que você use dispositivos periféricos ao usá-los. O programa que os controla deve ser carregado do disco. No entanto, no MSX, As instruções BASIC são expandidas automaticamente simplesmente conectando um cartucho de interface. Isso será feito.

Também oferecemos o "MSX-DOS2 japonês", que é uma versão atualizada do MSX- DOS e software integrado. O tão falado "H AL NOTE" e a GUI dedicada do turboR (Graphical User I) "MSXView", que foi introduzido como uma interface de software ( Um dos. Desta forma, você pode conectar facilmente o cartucho ROM ao slot e ele funcionará. poder estender , Esta é uma vantagem do MSX que outros computadores não possuem.

Os slots são um recurso útil, mas , Desenvolver um programa que faça pleno uso dele é É uma coisa bastante difícil. Um programador que pode escrever livremente programas em linguagem de máquina para o Z80 C PU. No entanto, pode levar mais de um ano para realmente entender o conceito de slots.

## 2.1.6 Este slot MSX2+ incomum

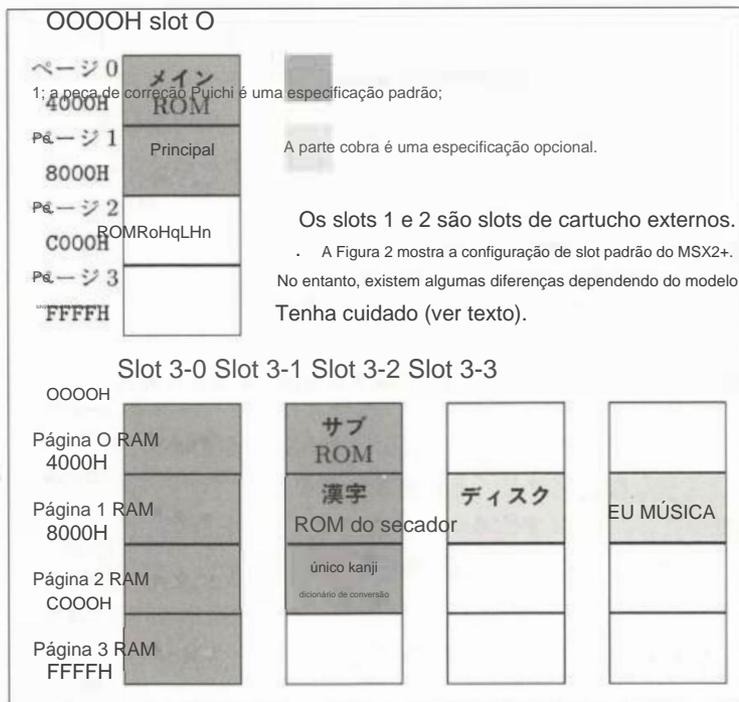
Como escrevi até agora, a máquina MSX possui extensibilidade e características ricas. O que ele me deu foi algo chamado slot. Porém, este slot também é um ponto fraco do MSX. havia. Isso ocorre porque "o MSX convencional possui diferentes configurações de slots dependendo do modelo, então É provável que ocorram problemas de compatibilidade de software."

Por exemplo, em um modelo onde os slots 1 e 3 são atribuídos aos slots de cartucho, Certos softwares não funcionam. Além disso, a RAM é colocada no slot de expansão do slot 3. Se estiver, talvez você não consiga usar funções de sub-ROM do DOS. programe com cuidado Você pode evitar esses problemas criando um sistema e testando-o em todas as máquinas MSX. Eu deveria levar um chute. Porém, se você torná-lo compatível com máquinas com todas as configurações de slot, o programa As desvantagens são que a RAM fica mais longa e a velocidade de execução fica mais lenta. De uma forma simples Slots são aqueles que não funcionam.

Com o desenvolvimento do MSX2+, alguns padrões relativos à configuração de slots foram finalmente decididos. Era. As Figuras 2.4 e 2.5 mostram exemplos de configurações de slots MSX2+. Dentro do corpo principal Dependendo do número de software que você deseja armazenar, você pode querer expandir apenas o slot 3 ou o slot 0. e ao expandir o slot 3.

A Figura 2.4 mostra o caso em que apenas o slot 3 é expandido. Slot básico O, BÁSICO A ROM principal é colocada lá, e o slot 1 e o slot 2 são usados como slots de cartucho externo.

Figura 2.4: Exemplo de configuração do slot MSX2+ (ao expandir apenas o slot 3)



Em seguida, instale 64 kilobytes de RAM em um dos slots de expansão do slot 3.

Coloque a RAM no mesmo slot para as páginas 0 a 3. mesmo

A sub-ROM, o driver de kanji e o dicionário de conversão de kanji único totalizam 48 kilobytes de ROM. e coloque-o em um dos slots de expansão no slot 3. Na figura, o slot

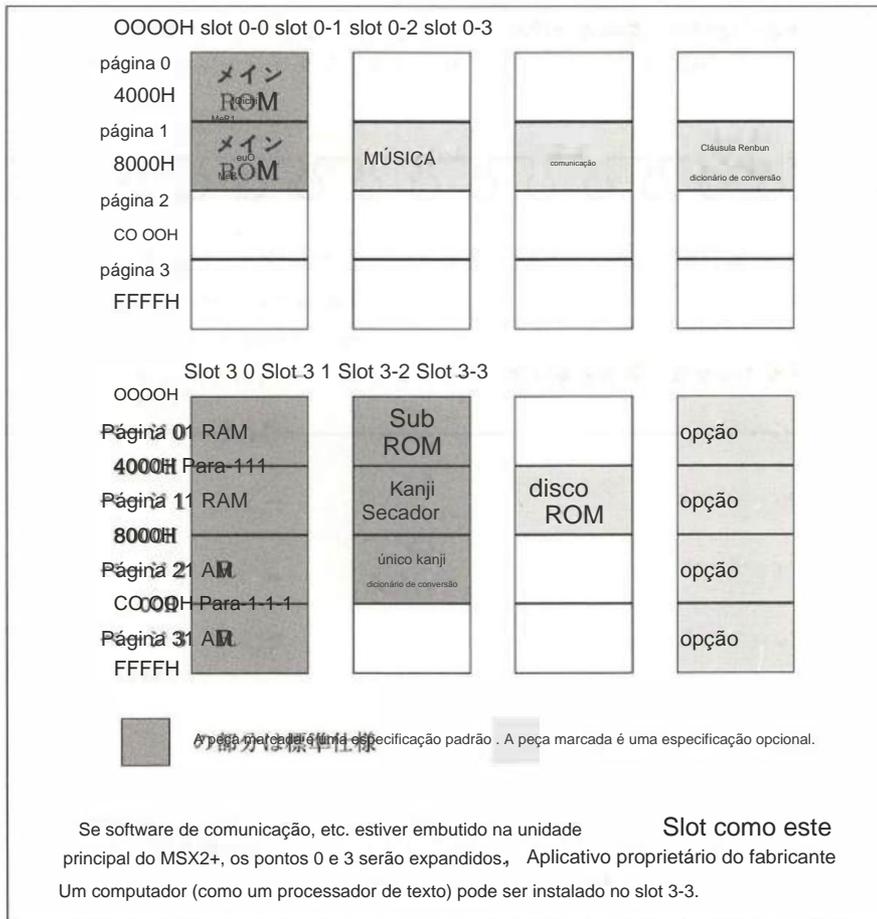
A RAM é colocada em 0 de 3 (outro slot de expansão do slot básico 3) e a ROM é colocada em 1 de 3. Embora seja , isso varia dependendo do modelo.

Em contraste, a Figura 2.5 mostra o caso em que tanto o slot 0 quanto o slot 3 são expandidos. Slot básico A ROM principal é colocada no slot de expansão O da porta 0. A configuração do slot 3 é mostrada na Figura Quase o mesmo que 2.4. Se você tiver uma interface de disco integrada, slot 0 Observe que a ROM é sempre colocada no slot de expansão 3, e não no slot de expansão 3.

As .4 e a área cinza claro na Figura 2.5. Ou seja, a interface do disco, especificações opcionais mostradas Em relação a cada ROM de dicionário de comunicação e conversão contínua de frases, consulte MSX2+ na Figura 2 são MSX-MU SIC (fonte de som FM). Portanto, não está embutido no corpo principal. Também pode ser conectado como um

Agora, o MSX2+, que possui a mesma quantidade de software integrado mostrado na Figura 2.5, possui 36 possibilidades teóricas.

Figura 2.5: Exemplo de configuração de slot MSX2+ (ao expandir os slots 0 e 3)



Várias configurações de slots são possíveis. Posso ouvir as pessoas dizendo: "Uau, são tantos!" Isso é verdade, mas o número de combinações ainda é menor que a configuração do slot MSX2. Não. Além disso, a sub-ROM, o driver de kanji e o dicionário de conversão de kanji único devem ocupar o mesmo slot. Ao colocá-lo na placa, a entrada/saída de kanji do MSX2+ deveria ter se tornado mais rápida que o esperado.

### 2.1.7 Expandir slot

As máquinas MSX possuem um ou dois slots de cartucho externos (geralmente usados para jogos). (onde você insere o cartucho) é um slot básico.

Você pode expandi-lo para 4 slots de expansão conectando um dispositivo. Por exemplo, 2 slots de expansão na unidade principal.

Se você conectar um expansor de slot a ambos os slots, serão criados um total de 8 slots.

É por isso que eu faço isso.

A única coisa com a qual você deve ter cuidado é que alguns cartuchos não funcionam no slot de expansão. MSX - DOS2 (tipo com RAM integrada) é um exemplo.

Certifique-se de conhecer o software que deseja usar antes de expandi-lo.



## 2.2 Experimentando a troca de slots

Quando se fala em MSX, o conceito de slots não pode ser evitado. Aqui, explicaremos principalmente como controlar slots com software e mudanças nas especificações do MSX2+.

### 2.2.1 Para trocar de slot

Embora eu tenha explicado o significado dos slots no MSX, não expliquei como trocar de slot. Então, a partir daqui, apresentarei o método. Claro, não é algo que possa ser alterado por uma pessoa usando um switch, mas pode ser alterado por um programa.

Não.

Primeiro, a CPU do MSX, Z80, tem algo chamado de “porta 1/0”. É como uma linha telefônica que permite à CPU se comunicar com o mundo exterior (ou seja, periféricos como fontes de som V DP e FM). O Z80 possui um total de 256 barcos de I/O, mas eles se diferenciam atribuindo endereços de 0 a 255 (OOH a FFH em hexadecimal) para cada um.

Para lidar com essas portas em BASIC, leia um valor de 1 byte da porta de E/S com a função “INP” e escreva um valor de 1 byte com a instrução “OUT”. Em linguagem de máquina, o comando “IN” “O comando OUT” faz a mesma coisa.

O slot básico é comutado dependendo do valor escrito no endereço ASH desta porta de E/S. Por outro lado, a leitura do valor neste endereço informará o status atual do slot. Os bits 7 e 6 correspondem à página 3, os bits 5 e 4 correspondem à página 2, os bits 3 e 2 correspondem à página 1, os bits 1 e 0 correspondem à página 0 e assim por diante, então IIIIOOOB (B é 2 Se você escrever o valor (ou seja, um número hexadecimal), as páginas 3 e 2 serão comutadas para o slot 3 e as páginas 1 e 0 serão comutadas para o slot 0. Além disso, o endereço de memória FFFFH é usado para alternar slots de expansão, mas isso é complicado, por isso vamos

omitir aqui. A propósito, trocar diretamente de slots usando um programa não é apenas problemático, mas também tende a causar problemas de compatibilidade, como o programa não funcionar em algumas máquinas. Portanto, os slots são realmente trocados usando o “BIOS”. BIOS significa “Sistema Básico de Entrada e Saída”. Como explicarei em detalhes mais tarde, é uma coleção de sub-rotinas em linguagem de máquina para controlar o hardware.

### 2.2.2 Como especificar o número do slot

Ao alternar slots usando o BIOS, você pode especificar o número do slot básico e o número do slot de expansão separadamente. Mas para isso precisamos de dois registros.

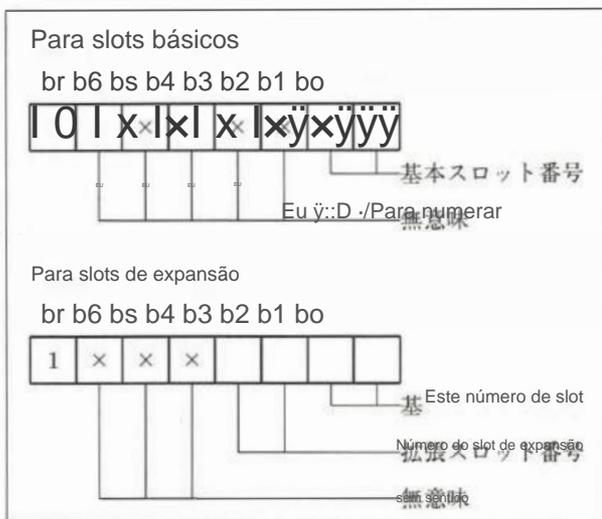
(um local de armazenamento temporário para dados dentro da CPU), o que não é econômico (?). Portanto, a Fig.

Como em 2.6, cada bit de 8 bits (1 byte) é usado com sabedoria para slots básicos e expansão. O método usado é especificar todos os slots de uma vez.

Por exemplo, para especificar o slot base 0, use o valor 0000000B (OOH em hexadecimal). Para especificar o slot de expansão 1 do slot básico 3, use o valor 10000 111B(87H).

Tudo que você precisa fazer é especificar.

Figura 2.6: Como especificar números de slot



O número do slot é mostrado na figura. 8 bits (= 1 byte) ser feito. Por exemplo, base Número do slot do livro O Para especificar Lote 3 slot de expansão Para especificar o ponto 1 10000 111B e Basta usar o valor É por isso. Além disso, na figura , cada um marcado com um O conteúdo dos bits é ignorado.

### 2.2.3 Funções do BIOS que operam slots

Primeiro, vamos aprender os símbolos usados para representar as funções do BIOS. O que é difícil é o BIOS 11 Cadastro cujo valor deve ser definido antes da emissão. O país é um registro cujo valor é retornado pela BIOS. Os tempos são Representa um registro no qual o BIOS grava valores sem sentido, o que significa que seu conteúdo original é destruído. Mãe IYH representa o byte superior do registrador IY e o conteúdo do byte inferior é ignorado. Ru. O primeiro endereço escrito é o endereço de entrada para chamar o BIOS. Sf.

- Caixa R Endereço 000CH
- DSLTL própria O número especificado no registro HL do slot especificado no registro A. Leia o conteúdo da terra.
- vezes Um número de slot Endereço HL

A 読んだ値  
 AF、BC、DE  
 割り込みが禁止 さい。

Endereço WRSLT 0014H

O número especificado no registro HL do slot especificado no registro A.  
 Escreva o conteúdo do registrador E no

Difícil A slot y y número

Endereço HL

E O que escrever

Não sólido

Figuras As interrupções' D

AF e BC estão desabilitadas.

Endereço CALSLT 001cH

Chame a rotina "Saf" nos slots de Gunji e de outros e envie-a.

Endereço IX fixo para ligar

Número do slot IY H

Depende da pessoa que está ligando para você

Figura IX Salva , Ei, registro de volta

o estado atual do slot na pilha e chama a sub-rotina desejada .

arquivo. AF, BC , O conteúdo dos registradores DE e HL é armazenado diretamente na sub-rotina.

Chin e quando a sub-rotina executa a instrução RET, o programa original

Voltar para Neste momento também , Os valores dos registros DE, HL são sub-azuis

é passado de AF e BC Chin. Quantos bytes de pilha são usados depende do slot

Varia dependendo da configuração do host.

E Endereço NASLT 0024H

Gunji Trocar slots.

país Um número de slot

Página H (2 bits superiores)

figura nenhum

foto AF, BC , DE, HL

Por exemplo , para alternar entre 2 páginas, defina 80H como BFH no registro H. Basta definir o valor de . Interrupções são proibidas.

### Endereço CALLF oo3oH

Publiquei as dimensões do sub-noro no slot de Iro et al .

Conforme mostrado no programa abaixo, o comando ST30H" é seguido pelo número do slot e Escreva a localização no programa.

```

Horário de início 30H
Número do slot do banco de dados
Endereço DW

```

Depende da pessoa que está ligando

O mesmo que , 1 registro anterior

CAL Sguchi, exceto pelo slot rígido Nagi IX e método de especificação de endereço. propósito especial ).

### Endereço EXTROM 01scH

Chame a sub ROM yE.

```

sólido      Endereço IX para ligar
país       Depende de quem você liga
vezes      IX , Ei, registro de volta
sólido      Funciona da mesma forma que C ALSLT, exceto que o slot de sub ROM é selecionado automaticamente.
Ouvir.

```

No entanto, o BIOS apresentado acima tem algumas limitações. . Nenhum para a página 3 Não pode ser usado. Para a página O, só pode ser usado ao chamar a ROM principal do DOS. Quer dizer, ele não se move e não pode usar páginas 2 e 3. ,

No entanto, dependendo da configuração do slot, você poderá usá-lo dependendo da configuração do slot. Tenha cuidado para não criar "programas", especialmente ao chamar subROMs do DOS.

Usando C ALSLT como

Às vezes ele se move e às vezes não.

#### 2.2.4 Como saber a configuração do slot

Conforme explicado anteriormente, a configuração do slot do MSX difere dependendo do modelo. disco eu Existem também especificações opcionais, como interfaces, para que você possa configurar tantos slots quanto máquinas.

Não é exagero dizer que existe. Então, deixe-me mostrar como verificar a configuração do slot da sua máquina e a presença ou ausência de dispositivos conectados. Opções

A área de memória do F380H ao FFF EH é chamada de "área de trabalho do sistema" e está localizada aqui. Ele armazena informações importantes para o BI OS, etc. Interface de disco

Uma vez conectado, a "área de trabalho do disco" será criada em um endereço um pouco menor que a área de trabalho do sistema.

Uma "consulta" é preparada. As informações relacionadas aos slots são armazenadas na .como 1

Tabela 2, Área de Trabalho do Sistema e Área de Trabalho do Disco.

A questão de qual slot está a RAM principal é importante, mas , "RA-MADO" na Tabela 2.1 está localizado na área de trabalho do disco. Portanto, se não houver disco, isso não funcionará. Há também o problema de não saber suas informações.

Lista 2 Este .1, as configurações de slots examinadas nessas áreas de trabalho do sistema são é um programa que exibe informações facilmente. A configuração difere dependendo do modelo, portanto verifique seu MSX Mas vamos tentar .

Além daquelas listadas na Tabela 2.1, existem outras áreas de trabalho do sistema que são úteis para o programa. No entanto, consulte o "Manual Técnico do MSX 2" para obter detalhes. Então isso Estas áreas de trabalho do sistema estão abertas a candidaturas, salvo indicação em contrário. O programa não deve ser reescrito. O sistema está ficando sem memória e está ficando sem memória. Eu tenho um programa que usa uma área de consulta. , Tenha cuidado, pois isso pode causar incompatibilidade. .

Tabela 2.1: Área de trabalho do sistema relacionada aos slots

Nome	Endereço	意味
RAMADO	F341H	pagina O Número do slot de RAM (1)
RAMAD1	F342H	pagina 1 (1) Número do slot de RAM do F342H
RAMAD2	F343H	pagina 2 Número do slot de RAM (1)
RAMAD3	F344H	pagina 3 Número do slot de RAM (1)
	F348H	Número do slot da interface MASTER Drive A (1)
	F348H	Número do slot sub ROM EXBRSA FAF8H (0 para MSX1)
	FCC1H	Número do slot ROM principal FCC1H
EXPTBL	FCC2H	Se o slot 1 do FCC2H está expandido (2)
	FCC3H	Se o slot 2 do FCC3H é expandido (2)
	FCC4H	Se o slot 3 do FCC4H é expandido (2)
(1) Válido somente se um disco estiver presente.		
(2) 80H se expandido, caso contrário 0.		

## 2.2.5 Explorar a área de trabalho do sistema

No software do jogo para MSX 2, se você executá-lo no MSX 20, poderá ver a imagem do título da TELA 12. Há algo que exibe a superfície. Além disso, estou tentando me comunicar mesmo que não haja cartucho de modem.

Neste caso, existem programas que exibem mensagens de erro amigáveis. Para criar . Este tipo de software um programa, apresentarei como um programa pode verificar o tipo e a configuração do hardware.

Primeiramente, para verificar se existe disco, leia o conteúdo do endereço FFA7H.

Se for C9H, não há disco. , Qualquer outro valor significa que existe um disco. .

Para descobrir o "tipo MSX", leia o endereço 2DH na ROM principal. Se 0 MSXI , 1 Se MSX2 em , Se for 2, é MSX2+, e se for 3, é turbo R.

geral , O conteúdo dos endereços 2BH e 2CH é O, mas são "feitos para exportação para o exterior". MSX" contém números que representam o tipo de teclado e o símbolo da moeda. Software de exportação Você só precisa se preocupar com isso ao criar software, por isso omitiremos a lista de números. .

Esta é uma observação lateral, mas os computadores pessoais MSX foram amplamente utilizados na Europa, na União Soviética e no futuro médio e próximo.

Um número considerável deles também é exportado para países do leste, como o Kuwait. Além disso, na vizinha Coreia do Sul, são utilizados em escolas.

Muitos foram introduzidos e considerados úteis nas aulas. É uma máquina muito internacional.

É N.

Pois bem, Para saber a "capacidade da RAM de vídeo", leia o endereço FAFCH. Bit 2 e Bit

Se o valor 1 for 00, são 16 quilômetros, e se for 10, são 01 então 64 kilobytes ,

É um robô. Os outros bits parecem ser usados para outros propósitos, então vamos ignorá-los.

Na lista da próxima página, use "AND 6" para extrair o valor correto do bit 2 e do bit 1 e, em seguida, Basta dividir isso por 2.

Esta lista também verifica a presença de um "BIOS de expansão" como um bônus adicional. Isso é comumente Para controlar hardware opcional, como modem de comunicação, fonte de som FM e dicionário de kanji. Esta é a função de Se o conteúdo do bit O do endereço FB20H for 1 e o conteúdo do endereço FFAH for C9H. Nesse caso, existe algum tipo de recurso avançado do BIOS. Para descobrir o que é, Requer um programa de linguagem de máquina complexo, então desta vez vou passar. Então este trabalho Embora não esteja escrito na especificação, o conteúdo do endereço FFCBH é aquele que possui funções estendidas do BIOS. Parece ser o número do slot do programa que está sendo usado.

Observe que os bits cujos significados não são determinados, como bits em um valor que representa um número de slot,

Não se sabe o que está escrito nos bits 6 a 4, etc. Portanto, "E"

Você pode ver como estou usando isso e ignorando seu conteúdo?

Parece que já escrevi isso muitas vezes, mas mesmo que as máquinas sejam do mesmo fabricante, varia dependendo do modelo.

As configurações dos slots podem variar, então , Depois de tentar em minha própria máquina experimente também na máquina de um amigo.. Testado em muitas máquinas , Exibir os resultados Eu acho que você deveria tentar.

## Listagem 2.1 (WHO\_AM".BAS)

```

100 ' Analisando estrutura de slots do MSX 110 ' por nao-i em 9
de janeiro de 1989 120 CLEAR : DEFINT AZ : CLS
130 VE = PEEK(&H2D) : ' Versão No. do BASIC
140 IF VE=0 THEN PRINT "I a Prato MSXI"

150 SE VE=1 ENTÃO IMPRIMA "Eu sou MSX2"
160 SE VE=2 ENTÃO IMPRIMA "Eu sou MSX2+"
165 SE VE=3 ENTÃO IMPRIMA "Eu sou MSX turbo R"
170 SE VE>3 ENTÃO IMPRIMA "Quem sou eu?"
180 VR = (PEEK(&HFAFC) AND 6) ¥ 2¥ ' tamanho da VRAM 190 IF VR=0 ENTÃO IMPRIMA "VRAM
16KB"
200 SE VR=1 ENTÃO IMPRIMA "VRAM 64KB"
210 SE VR>1 ENTÃO IMPRIMA "VRAM 128KB"
220 MR = PEEK(&HFC49) : ' tamanho da RAM principal 230 IF MR >=
&HE0 THEN PRINT "RAM 8KB"
240 SE MR < &HE0 E MR >= &HCO ENTÃO IMPRIMA "RAM 16KB"
250 SE MR <CO ENTÃO IMPRIMA "RAM >= 32KB"
260 FOR SS = 0 TO 3 : EXPTBL 270 PRINT USING
`` Slot # é " SS ; SS 310 PRINT 320 SS = PEEK(&HFCC1) :
PRINT "ROM principal está em" • GOSUB 530 330 SS =
PEEK(&HFAF8) : PRINT "Sub ROM está em" : GOSUB 530 340 IF PEEK(&HFFA7) <> &HC9 THEN 360 350
PRINT "Eu tenho
disco(s) : GOTO
450 360 PRINT "Eu não tenho disco(s)." "

370 SS = PEEK(&HF348) : PRINT "FDC ROM está em"; está em " : GOSUB 530 400 SS = PEEK(&HF343) :
PRINT "P2 RAM está em " : GOSUB 530 410 SS = PEEK(&HF344) : PRINT `` P3 RAM está em " : O
endereço inferior da área de trabalho do disco é " : 430 PRINT RIGHT$("00"+HEX $(PEEK(&HFC4B) )
,2) : 450 ' detecção de BIOS estendido 460 IF (PEEK(&HFB20) AND 1) = 0 THEN GOTO 520 470 IF
PEEK(&HFFCB) = &HC9 THEN GOTO 520 480 PRINT : PRINT "Eu estendi o BIOS."

490 SS = PEEK(&HFFCB)
500 PRINT "ROM do BIOS estendido pode estar em 510 GOSUB 530 520 END

530' exibindo o número do slot
540 IMPRIMIR USANDO "slot preparado #";
550 SE (SS E 128) = 0 ENTÃO 570 slot estendido #"
560 IMPRIMIR USANDO" (SS E 12) ¥ 4;
570 IMPRIMIR: RETORNAR

```

## 2.2.6 Especificações de hardware MSX2+

MSX2+ inclui pequenas melhorias de hardware. A Tabela 2.2 resume o

mas, Esta é uma porta de E/S para a qual as especificações foram definidas ou adicionadas recentemente.

Tabela 2.2: Portas de E/S MSX2+

Uso de Enjira Bancho
Fonte de som FM integrada de 7 canais
Fonte de som FM 7DH integrada
ROM Kanji DAH Nível 2
ROM Kanji DBH Nível 2
Controle de inicialização F4H
Dispositivo F5H habilitado

Novas especificações são definidas ou adicionadas. O que foi adicionado. No , verdadeiro profissional entanto, em gramas, a porta 1/0 é usada diretamente. É melhor usar o BIOS.

Os endereços 7C H e 7 DH são portas de E/S para operar a fonte de som FM integrada na unidade principal. para. Ao contrário disso, o tipo de fonte de som FM fornecida por um cartucho (que poderá ser lançado no futuro) , Usa a mesma porta de E/ S do "FM- PAC" da Nomanasonic .

Para verificar se a fonte de som FM está integrada na unidade principal, verifique cada slot. e leia do endereço 40 18 H até o endereço 40 1 FH. Seu conteúdo é a frase "APRLOPLL" Se a string corresponder, significa que a ROM do programa que controla a fonte de som FM está nesse , slot e que a fonte de som FM está incorporada no dispositivo.

Por outro lado, no caso de um cartucho de fonte de som FM, o conteúdo do endereço 40 18 H é "PAC20 PLL" Parece que será composto por quatro caracteres que representam o tipo de produto e a palavra "OPLL".

" Para MSX-Write e alguns modelos de cartuchos, você pode selecionar "BASICy" no primeiro menu. Ao selecioná-lo, a tela de título do MSX aparece como se tivesse sido redefinida. esse Para preparar o software, vá para o endereço 0 da ROM principal e reinicie.

Isso ocorre porque ele executa o mesmo processamento que o .

Anteriormente, não havia como distinguir com segurança entre um salto para o endereço 0 e um reset verdadeiro. Como resultado, podem ocorrer mau funcionamento do software. Do MSX2+, E/S Hardware foi adicionado ao endereço da porta F4H para verificar o status de redefinição. Ta É isso , Na verdade, a BIOS adicionada à ROM principal do MSX2+ é usada conforme mostrado abaixo.

LIGUE 17AH  
DR80H  
LIGUE 17DH  
JP 0

O programa do cartucho ROM chamado e emitido durante a inicialização é

## LIGUE 1 7AH

Faça o seguinte. Se o bit 7 do registrador A for 0, é um reset verdadeiro. Se 1 Jump O permite que você saiba que foi chamado.

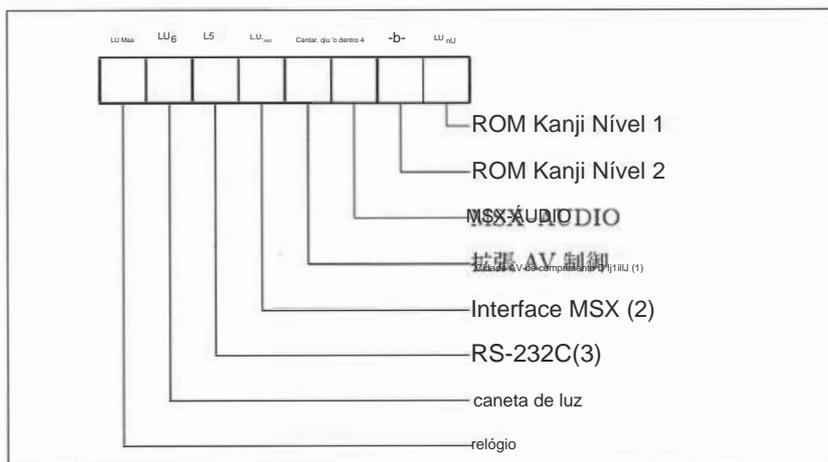
## 2.2.7 Habilitação do Dispositivo de Prevenção de Colisões

Se você conectar um cartucho Kanji ROM a um MSX que tenha uma Kanji ROM integrada, os caracteres Kanji serão exibidos. Não só não será exibido corretamente, mas também existe o risco de colisão e falha de hardware. criança Um recurso valioso que evita isso é a função "Device Enable" controlada pelo endereço F5H da porta de E/S. Chama-se "Touro".

O hardware mostrado na Figura 2.7 está desconectado do caminho na reinicialização. Em seguida, escreva o valor de 1 byte (8 bits) no endereço F5H da porta de E/S para defini-lo como 1. O hardware integrado correspondente ao bit alterado é conectado ao caminho. esses O processamento é redefinido ou salta para 0 (dependendo do software, o endereço ROM principal 0 é Isso é feito automaticamente após o início da execução do programa.

Não havia , se você escrever 0 no endereço F5H da porta de E/S, ela já está conectada. regulamentações sobre separar ou não o hardware usado no MSX2. Por esta razão "MSX -

Figura 2.7: Habilitação do dispositivo



O endereço F5H da porta 1/0 ativa ou desativa o hardware integrado. (O hardware correspondente ao bit escrito como 1 é válido).

(1) Função super pose etc. controlada pelo endereço F7H da porta 1/0.

(2) Embora esteja nas especificações, não foi colocado em uso

prático. (3) Não está relacionado ao modem.

Se algo como "Write" pular para o endereço 0 na ROM e tentar reinicializar o BIOS, poderá ocorrer confusão.

No MSX2+, se você escrever 0, o hardware interno será desconectado do caminho.

Foi organizado para que pudesse ser realizado. Isto permite utilizar os endereços F4H e F5H da porta de E/S.

Ao criar o software básico para MSX2+, a compatibilidade e a confiabilidade são agora maiores do que nunca.

Vai melhorar.

## 2.3 Configuração do slot MSX tubo R

Aqui, explicaremos a configuração dos slots do recém-anunciado M SX turbo R. O que chama a atenção é que a estrutura dos slots foi finalmente unificada. criança Isto é extremamente significativo.

### 2.3.1 A configuração do slot é finalmente unificada.

A Figura 2.8 mostra a configuração de slot do turboR. Compatível com velocidades de CPU mais rápidas , Aplicativo A estrutura do slot foi padronizada para facilitar o desenvolvimento e a depuração de aplicativos de programas.

Neste diagrama, parece que há 64 kilobytes de RAM no slot 3-0. , fruta 256 kilobytes de RAM principal são conectados através do mapeador de memória. Há. Deste total, a porção que excede 64 kilobytes é a área de trabalho japonesa MSX - DOS2. Geralmente é usado para o "modo DRAM", que é explicado em outro capítulo. , Porém, o programa aplicativo usa o BIOS estendido para alternar o Matsuno f e isso acontece. Também é possível usar essas RAMs.

Além disso, a página 1 do slot 3-2 contém a ROM do sistema DOS. e Porém, aqui temos 16 kilobytes de ROM DOSI (M SX - DOS) e 48 kilobytes. Uma ROM DOS2 padrão está conectada e pode ser trocada automaticamente conforme necessário. Está melhorando.

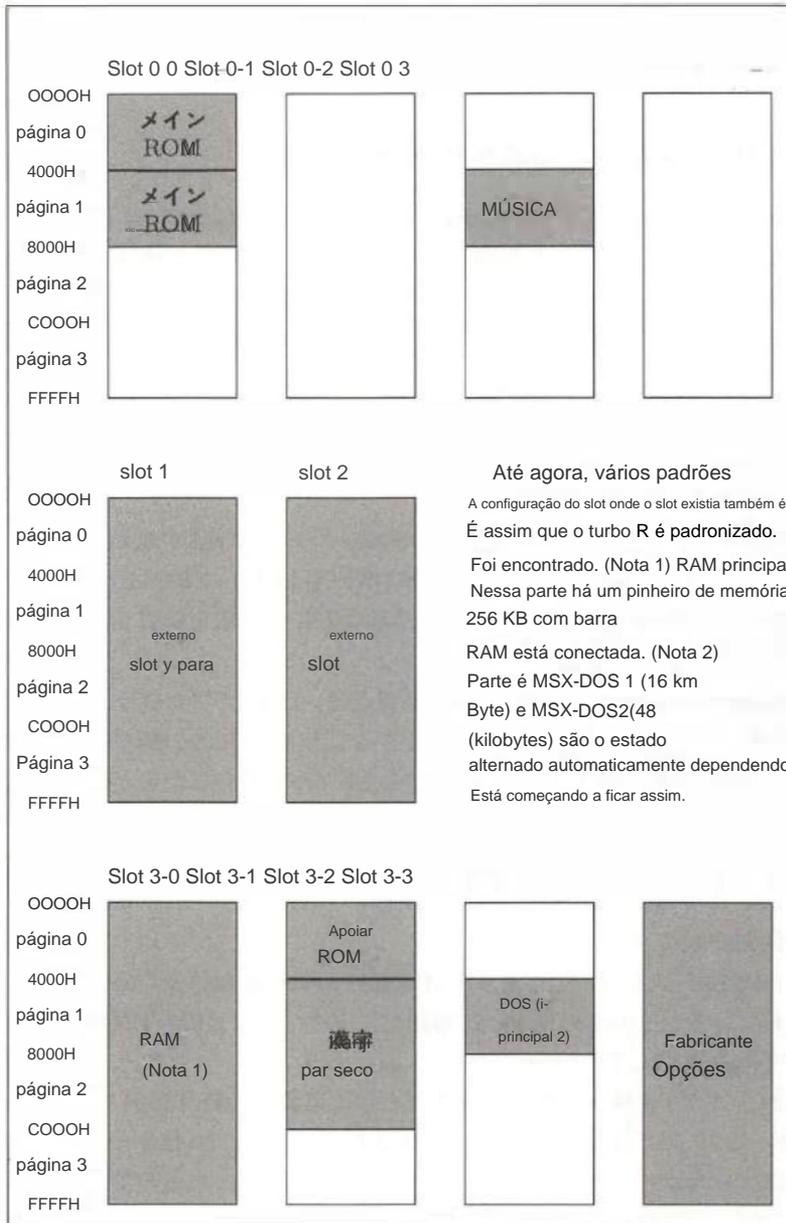
A maior vantagem desta configuração de slot padronizada é que os programas DOS podem ser O programa de processamento de interrupções do DOS Você pode colocar Ram em qualquer endereço. Eu apresentei isso em uma antiga revista M. Se, o slot 0 expandido tiver RAM e sub ROM, a instalação do M SX - DOS Havia a possibilidade de que a função de chamada entre slots e o programa de tratamento de interrupções ficassem fora de controle. Porém, com o turboR, o slot 3 expandido possui RAM e sub ROM, portanto isso não é possível. Problemas como este não ocorrem.

Além disso, a ROM do OPLL Draino, ou seja, FM-BIOS, é sempre colocada no slot 02. Portanto, o software dedicado turboR é

Você também pode pular a etapa de busca do alvo.

Além disso, como um exemplo especial dos benefícios da configuração unificada de slots, "Konami "Cartucho 10x". Isso requer a configuração do cartucho 10x no slot 1 e o cartucho de jogo no lote 2 e é compatível com alguns M SXI. e a máquina M SX2 não funcionou. No entanto, com M SX2+ e turboR, a velocidade externa Como o lote é determinado nos slots 1 e 2, tais programas especiais podem ser facilmente executados. O ambiente foi agora criado para garantir que isso possa ser alcançado.

Figura 2.8: Configuração do slot MSX turbo R



Além disso, para empresas de software, a configuração unificada de slots facilita a  
A maior vantagem é que reduz o número de problemas causados por problemas de configuração de slots.  
Do ponto de vista da criação de software, , Maior velocidade da CPU e maior capacidade de RAM  
a consistência da configuração do slot é muito mais atraente do que o design real. Viva o turbo R!



1 第 3 章 漢字 BASIC



Este capítulo foi retirado da "M SX2 + Technical Expedition" da edição de abril de 1989 da M SX Makatsun.

Esta é uma versão reeditada do artigo.

## 3.1 Analisar Kanji BÁSICO

Uma das características do M SX2+ e das máquinas posteriores e do DOS2 era que os kanji eram mais fáceis de usar. Além das cadeias de caracteres e nomes de arquivos de programas, Pode usar kanji. este capítulo BASIC, apresentaremos relatórios sobre as funções Kanji BASIC.

### 3.1.1 Hardware necessário para Kanji BASIC

Como posso usar o Kanji BASIC? Primeiro, os mais básicos são M SX2+ e livro turbo R (ou cartucho DOS2). Kanji BASIC em ambos os casos. Possui uma ROM integrada. A diferença entre os dois em termos de funcionalidade é TELA 10 a 12 modos de imagem natural podem ser usados, e a unidade principal possui uma ROM Kanji integrada. Estar lá.

Além disso, um exemplo um pouco especial para o qual gostaria de chamar a atenção é o divulgado pela Sony. Com o cartucho japonês "HBI-JI", você Kanji BASIC e Kanji ROM integrados pode tornar seu M SX2 existente compatível com kanji.

DOS2, MSX2+ e turbo R possuem uma função de "conversão de kanji único". Isso é "leitura" Este método usa "código J IS" ou "código J IS" para especificar caracteres kanji por letra. Mas esta é uma frase longa Como é inconveniente inserir capítulos, adicionamos uma função de "conversão contínua de cláusulas" chamada "MSX-JE". Ru. Por exemplo, se você escrever uma frase em hiragana como "Hoje é um bom dia", você poderá dizer "Hoje é A conversão de oração contínua converte a frase "É um lindo dia" em uma frase que inclui kanji e kana. Utiliza o mesmo método utilizado pelos profissionais.

A Tabela 3.1 lista o hardware que incorpora o MSX-JE. embutido no corpo principal Nesse caso, você pode usar a função de conversão de cláusula contínua como está, caso contrário, poderá conectar um cartucho. Apenas faça. No entanto, observe que M SX -JE integrado = não é compatível com Kanji BASIC. coisa. Como escrevi no início, em combinação com M SX2+, turbo R DOS2, etc. Isso permite que você use o Kanji BASIC, que desativou a conversão contínua de cláusulas .

M SX -JE tem uma função de aprendizagem. Por exemplo, converta o hiragana "kanji" em "kanji". Suponha que três tipos de candidatos sejam exibidos: "personagem", "secretária" e "feel". Entre estes, selecione "secretária". Depois de selecioná-lo, na próxima vez que tentar converter a mesma palavra, "secretária" será sua primeira escolha. Como funciona. As palavras que você usa com frequência serão priorizadas; portanto, quanto mais você as usar, mais palavras usará. Se o livro for adequado para você, você poderá convertê-lo com mais eficiência. O item "SR AM" na Tabela 3.1 é "Sim"

O que torna isso possível é que esse resultado de aprendizagem é obtido através da utilização de uma memória que retém seu conteúdo mesmo quando a energia é desligada.

Tabela 3.1: Lista de hardware integrado do MS X-JE

Fabricante	Nome do produto	forma	漢字ROM I SRAM I	
Nof Nasonic FS-AST	Panasonic	MSX turbo-R	1, 2	
FS-AIWX	MSX2+	Nof Nasonic FS AIWX MSX2+	1, 2 disponíveis	
Panasonic FS-SR021	Cartucho (1)	Nof Nasonic FS	1, 2	Sim
4600F	—		1, 2	Sim
	—		1	Sim
Impressora MSX2	SONIC FS PWI (2)		1	Sim
sony	HB FIXV	MSX2+	1, 2	Sim
sony	HB-FIXDJ	MSX2+	1, 2	Sim
sony	Cartucho HBI-JI	Sanyo PHC-77	1, 2	Sim
		MSX2	1	無
Laboratório HAL	Cartucho HAL NOTE (3)	Cartucho de gravação	1, 2	Sim
ASCII MSX	ASCII		1	無
	Cartucho MSX-Write	JI	1, 2	Sim

O item kanji ROM indica a presença ou ausência de kanji ROM de 1º e 2º nível. SRAM "Sim" é , Mesmo que a energia seja desligada, o aprendizado permanece. (1) Carrinho Processador de texto AIWX uma crista. (2) Conjunto de cartucho e impressora dedicada. (3) Cartucho SO dedicado com disco e disco .

Ele foi projetado para deixar para trás.

### 3.1.2 O que é software compatível com MSX-JE?

#### EM

Em combinação com a aplicação, é possível utilizar a função de conversão contínua de cláusulas. Kanji Como ROMs e ROMs para dicionários de conversão de cláusulas contínuas são relativamente caros, a MS Compartilhar software é muito econômico.

A forma como um aplicativo usa o MS X-JE é determinada pelas especificações, portanto Software marcado como "Compatível com MSX JE" funcionará com qualquer MSX JE. Agora é possível fazer isso. Entretanto, a H ALNOTE na Tabela 3.1 indica que o cartucho e Devido à combinação de discos Ele usa um sistema operacional dedicado chamado HALOS. de haste, existem restrições à combinação com software diferente do HALOS.

### 3.1.3 Explicando o princípio de funcionamento do secador Kanji

Esta é uma explicação simples de como os kanji são representados dentro dos computadores, não apenas no MS X.

Primeiro, os caracteres alfabéticos e katakana podem ser representados por valores de 1 byte (8 bits), mas os kanji não podem ser representados por valores de 1 byte (8 bits).

Para fazer isso, é necessário um código de 2 bytes. De acordo com JIS (Padrões Industriais Japoneses), os kanji são

é expresso usando um código de 2 bytes.

Sub...3021H (1º nível)

Braço...-4F53H (1º nível)

2...5021H (2º nível)

Aves...737EH (2º nível)

No entanto, este "código JIS" é considerado difícil de processar quando caracteres alfabéticos e kanji são misturados. Há um problema. Portanto, convertamos o código JIS para facilitar o processamento dos computadores. O "código Shift JIS" é usado em muitos computadores pessoais, incluindo MSX.

Sim. Algumas pessoas o chamam de "Código Kanji da Microsoft", mas isso é apenas

Somente quem estiver interessado deve se

lembrar disso. Em qualquer caso, o código Shift JIS pode ser usado diretamente para usar software criado para caracteres ingleses.

Ou, com algumas modificações, você pode usar uma versão conveniente projetada para enganá-lo em japonês.

Código Kanji. O sistema operacional mais popular para computadores pessoais de 16 bits.

Sistemas operacionais como MS-DOS e OS-9, que se tornaram populares recentemente, também exigem uma mudança.

O código JIS Kanji é usado. Os arquivos de documentos podem ser compartilhados entre diferentes computadores.

Isso significa que também é possível trocar.

A propósito, quando olhei a tabela de códigos kanji, percebi que números e letras estavam misturados.

Para evitar confusão, usaremos o termo geral para nos referirmos a caracteres alfabéticos, kanji, etc. que são representados por códigos kanji de 2 bytes.

"Caracteres de largura total". Por outro lado, os caracteres representados por um código de caracteres de 1 byte são chamados de "caracteres de meia largura".

Faremos uma distinção entre os dois. Por exemplo, "ABCD" para caracteres de meia largura e "ABCD" para caracteres de largura total.

Observe que "A B C D" são tratados como caracteres completamente diferentes.

Pois bem, vamos ao assunto principal. MSX2+ e turbo R, função de entrada/saída kanji DOS2 é chamado de "Kanji Draper". Isto não está limitado ao BASIC ou DOS, mas muitos

Um programa aplicativo é disponibilizado para você.

A Figura 3.1 mostra o princípio de funcionamento do Secador Kanji. ) Quando explico passo a passo, começo pela chave.

Leia (julgar) a entrada "kana" ou "romaji" de largura total do quadro e exiba-a na tela.

Exibição ativada. Em seguida, chame o MSX-JE e converta hiragana em kanji (caracteres de largura total). durar

Em seguida, o código Kanji convertido é enviado para a aplicação através de BASIC ou DOS.

Isso significa que ele será enviado para o programa de download.

Por outro lado, se um programa aplicativo exibir caracteres na tela,

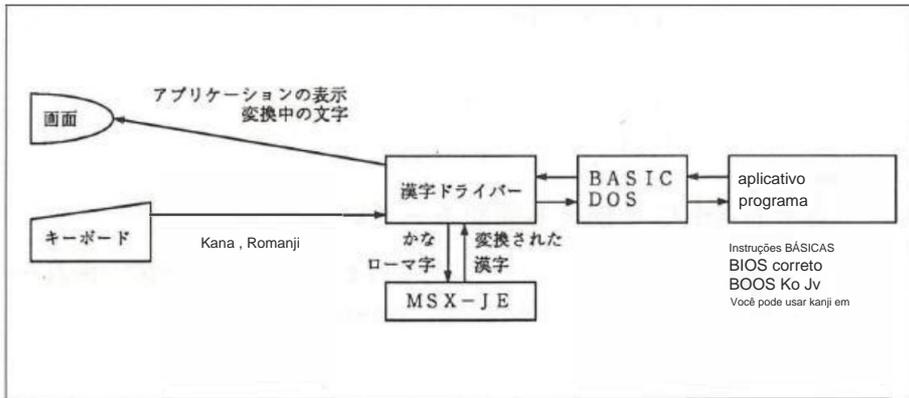
Tudo que você precisa fazer é enviar o código kanji para o driver kanji.

Para programas aplicativos, este driver Kanji usa comandos BASIC.

Chamadas de BIOS, chamadas BDOS (chamadas de BIOS para entrada/saída por programas DOS)

Parece que uma função de entrada/saída kanji foi adicionada a uma função semelhante. Em particular,

Figura 3.1: Princípio de funcionamento do Secador Kanji



O fato de o próprio programa aplicativo não precisar operar o dicionário MSX JE é uma grande vantagem do código Kanji. Essa é uma grande vantagem do Leipah.

### 3.1.4 Hardware e software compatíveis com JE

Para referência, aqui está uma lista de hardware e software compatíveis com MSX-JE lançados até o momento. Eu parei.

#### •Cartucho de modem

Pana Sonic FS C MI –  
 Pana Sonic FS C M820  
 Sony HBI 1200  
 cânone V M-300  
 Myojo Eletricidade V-3

#### •MSX com modem integrado

Pana Sonic FS AIFM –  
 Sony HB-T7  
 — HB-7600  
 — ML-7S2H

- Software

sony Documento Sakuzaemon

Sony Postal Shoemon

FERRAMENTAS ASCII MSX-

TERMASCII MSX-DOS2

MSXView ASCII

### 3.1.5 Vários modos de tela que podem ser usados com Kanji BASIC

Assim como existem muitos modos de tela para MSX2+ e turbo R, existem muitos modos de tela para Kanji BASIC. C também é complicado. A lista é mostrada na Tabela 3.2. Em BÁSICO,

LIGUE PARA O KANJI

LARGURA

Por comando. Em DOS2

KM EDO

MOD0

Cada comando especifica o modo de tela. por exemplo,

CHAMADA KANJI: LARGURA 32

Ao executar este comando, a tela exibirá 32 caracteres x 12 linhas. A mesma coisa com DOS2

Para fazer isso,

KMODE 0

MOD0 32

Tudo que você precisa fazer é Entretanto, se você iniciar a versão em inglês do sistema, o driver Kanji irá

Como não está carregado, vamos voltar ao BASIC e acessar o modo Kanji. Agora , vamos explicar o significado da

**Tabela 3.2 em detalhes.** Em primeiro lugar, "contagem de pontos na tela" refere-se ao número de pontos exibidos na tela.

Número de pontos mostrados. Kanji são representados por combinações desses pontos.

No modo de tela vertical de 424 pontos, é usado um método de exibição denominado "entrelaçado".

Sim. Isso exhibe alternadamente duas telas deslocadas verticalmente em meio ponto, resultando em

Como aumentar o número de pontos. No entanto, a desvantagem é que a tela pisca, facilitando o cansaço dos olhos.

"Contagem de pontos Kanji" é o número de pontos para representar um kanji. Normalmente os kanji são 16 x 16

Embora seja expresso em pontos, ele é compactado em 12 pontos horizontais x 16 pontos verticais e 40 pixels são exibidos em uma tela de 512 pontos horizontais.

Também é possível exibir os kanji dos caracteres. Eu também tenho um cartucho de modem Panasonic.

Uma vez conectado, a ROM Kanji integrada de 12 x 12 pontos será selecionada automaticamente.

Tabela 3.2: Modos de tela Kanji BASIC

画面下の点数	漢字下の点数	半角文字数	Método de configuração
Número de pontos na tela	Número de pontos kanji		
256x212	16x16	32 x12	CHAMADA KANJI0: LARGURA 32
256x212	12x16	40 x12	CHAMADA KANJI1 : LARGURA 40
256 x 424	16x16	32 x24	CHAMADA KANJI2: LARGURA 32
256 x 424	12x16	40 x24	CHAMADA KANJI3: WID 100 I 40
512 x 212	16x16	64 x12	CHAMADA KANJI0: LARGURA 64
512x212	12x16	80 x12	CHAMADA KANJI1: LARGURA 80
512 x 424	16x16	64 x24	CHAMADA KANJI2: LARGURA 64
512 x 424	12x16	80 x24	CHAMADA KANJI3: WID 100 I 80

"Número de caracteres de meia largura" é o número de dígitos e linhas de caracteres de meia largura exibidos na tela. Naturalmente, todos

No caso de caracteres quadrados (kanji), o número de dígitos é metade do valor da tabela.

Uma coisa a notar aqui é que todas as linhas da tabela são BASIC.

Não pode ser usado em todos os lugares. Exibição de teclas de função e funções para conversão de kanji.

Na visão traseira, são utilizadas uma ou duas linhas na parte inferior da tela.

### 3.1.6 Texto Kanji e gráficos Kanji

Agora, na verdade, existe um problema mais complicado com os modos de tela. Em BÁSICO,

```
LIGUE PARA KANJI1
```

```
LARGURA 40
```

```
TELA 0
```

É claro que quando você executa este comando, o modo de tela se torna o segundo a partir do topo na Tabela 3.2.

O importante a notar aqui é que mesmo que "SCREEN 0" seja especificado,

O VDP deve estar no estado "TELA 5" com 256 x 212 pontos. Esse tipo de situação

Este modo é denominado "modo de texto Kanji".

Neste modo, você pode inserir ou modificar programas BASIC, inserir usando o comando INPUT, A saída usando a instrução PRINT pode ser executada sem problemas. No entanto, LINHA, PINTURA, etc. Funções gráficas não podem ser usadas.

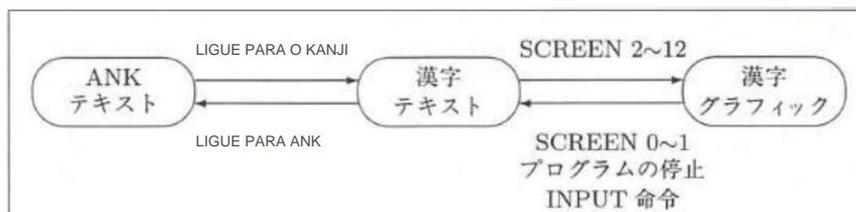
Para manipular gráficos no modo Kanji, use um comando como "TELA 5" para rolar a tela.

É necessário mudar para o "Modo gráfico Kanji". Eu só quero que você se lembre

Neste modo, funções gráficas e saída de kanji podem ser usadas, mas como regra geral, kanji

Resumi que não é possível. Certifique-se de não cometer um erro. A Figura 3.2 mostra como alternar entre os modos de tela acima. inserir informações.

Figura 3.2: Alternando modo de tela



Quando um programa começa a ser executado, quando termina ou quando ocorre um erro, você pode verificar se o modo de tela está definido conforme o esperado. . Ukkari “LIGUE PARA KANJI” Se você esquecer, o programa funcionará corretamente imediatamente após inseri-lo, mas assim que você desligar a energia, às vezes, quando tento recarregar o programa, ele não funciona.

Além disso, no modo gráfico Kanji, use os comandos “LINE” e “LINE INPUT”. e , reverte automaticamente para o modo de texto Kanji. Para evitar isso, digite “INPUT\$”. ou função “INKEY\$” para ler o teclado.

### 3.1.7 Esta é a maneira correta de usar o kanji dreipÿ.

Kanji Dryper derruba a sabedoria convencional de que as funções kanji do MSX não são práticas. O software que desenvolvemos é “Incrível”. No entanto, foi “adicionado” ao BASIC. No entanto, há uma armadilha surpreendente. O texto kanji e o gráfico kanji que acabei de explicar As diferenças nos ganchos são um desses segredos. Aqui, usaremos o kanji “dripa”.

Aqui estão algumas coisas que você deve ter em mente ao fazer isso.

A primeira vez que uma instrução “CALL KANJI” for executada após reiniciar o MSX, Uma área de trabalho é fornecida para usar o driver de personagem e o MSX-JE. Naquela hora O conteúdo das variáveis BASIC desaparece e o número de repetições do comando “FOR-NEXT” e “RETURN” Uma área de trabalho chamada “Pilha de Software” para registrar o número da linha para a qual uma instrução retorna. será inicializado . por exemplo,

```
10A=1
vinteLIGUE PARA KANJIO
30 IMPRIMIR A
```

Quando o programa é executado, “0” é exibido pela primeira vez (ou seja, o valor da variável A (está definido como 0), mas a partir da segunda vez, “1” é exibido corretamente. Também,

```

10 GOSUB 80
    █
70 FIM
80 LIGUE PARA KANJIO
90 RETORNO

```

Algo assim. Quando executo um `CALL KANJIO`, “RETURN” quando “CALL KANJIO” é executado programa, esqueço para onde o comando retorna e o programa se comporta de maneira estranha.

Além disso, quando o driver Kanji aloca uma área de trabalho na memória, `BÁSICO` Existe também o problema de que o número de áreas de trabalho diminuirá proporcionalmente. Tamanho da memória Pai não fu. Programas e programas que usam sub-rotinas em linguagem de máquina têm memória limitada. Pode haver momentos em que ele pare de se mover.

O driver Kanji converte o código shift JIS dentro do programa em código JIS.

Esta função tem a função de imprimir em uma impressora kanji.

Funciona em comandos e BIOS “LPTOUT”. No entanto, acabei de ligar a impressora.

A “impressão de imagens em bits”, que imprime gráficos controlando bits por vez, tem efeitos colaterais. É . Portanto, antes de imprimir a imagem bit, selecione o número F418H na área de trabalho do sistema. necessário escrever um valor diferente de zero no campo “RAWPRT” (chamado RAWPRT) para proibir a conversão do código kanji.

Devido à .

necessidade , a discussão a seguir é voltada para programadores avançados. Em primeiro lugar, o que está listado na Tabela 3.3 Aqui está uma lista de ganchos que podem ser reescritos por Kanji Dryper. Outros programas podem acessar esses arquivos. Se você reescrever o código, o driver Kanji poderá não funcionar corretamente, portanto, tome cuidado.

Em seguida, quando o gancho chama o driver Kanji com uma chamada entre slots.

Tabela 3.3: Ganchos usados pelo secador Kanji

Nome do	função
endereço Ij FDA4H H.	Exibir um caractere na tela do CHPU
Exibir o cursor FDA9H H.DSPC	
FDAEH H.	Apagar cursor ERAC
FDB3H H.	Display Teclas de função DSPF
FDBSH H.	Limpa a exibição da tecla de função ERAF.
FDBDH H.	Mude a tela TOTÉ para modo texto.
FDC2H H.	Ler l caractere do teclado CHGE.
O editor FDDBH H.	PINL BASIC lê uma linha
Leia a linha FDE5H	INLI 1.
FFB6H H.	Escreva a letra l na impressora LPTD

Os programas aplicativos podem usar esses ganchos. Caso contrário, o driver kanji poderá não funcionar corretamente. •

Supõe-se que o conteúdo dos , O conteúdo do registro IY será destruído. Portanto, os kanjis registros anteriores e do BIOS relacionado à entrada/saída IX seja salvo.

Os programas criados com kanji não funcionam corretamente no modo kanji.

Capítulo 4  
第 4 章

# V9958VDP



As seções 1 a 4 deste capítulo referem-se ao "V9938 MSX-VIDEO Technical Day". Esta é uma versão reeditada do "Tab Book" e das "Especificações V9958" pelo departamento editorial, e as funções comuns do V9958 são omitidas, portanto "MSX-Datapack" etc. Consulte o seguinte.

As seções 5 a 7 deste capítulo são da "MSX Magazine" edição de dezembro de 1988, 1989 "MSX2+Technical Exploration" nas edições de janeiro, novembro, dezembro e janeiro de 1990. Esta é uma versão reeditada do artigo "Tai."

Em documentos de hardware como "Especificações V9958", "modo VDP" é usado na explicação, mas neste livro é usado em conjunto com o artigo da MSX Magazine. Use o modo de tela BASIC. O modo VDP e o modo de tela BASIC correspondem à Tabela 4.1.

Tabela 4.1: Modos VDP e modos de tela BASIC

Modo VDP	Modo de tela BÁSICO
TEXTO 1	TELA 0: LARGURA 40
TEXTO 2	TELA 0: LARGURA 80
TELA MULTI COLORIDA	TELA 3
GRÁFICO 1	TELA 1
GRÁFICO 2	TELA 2
GRÁFICO 3	TELA 4
GRÁFICO 4	TELA 5
GRÁFICO 5	TELA 6
GRÁFICO 6	TELA 7
GRÁFICO 7	TELA 8 (TELA 10~12)



Tabela 4.3: Registro de Comando

	b7	b5	bs	b4	b3	b2	b1	bo	SX7	SX5	SXs	SX4	SX3	
R#32	SX2	SX1	sx											Fonte X (Baixa)
R#33	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	▪	Fonte X (Alta)
R#34	SY7	SY5	SYs	SY4	SY3	SY2	SY1	SYo	SYg	SYs				Fonte Y (Baixa)
R#35	▪	▪	▪	▪	▪	▪	▪	▪	DX7	DX5	DXs			Fonte Y (Alta)
R#36	DX4	DX3	DX2	DX1	DXo	DXs	DY7	DY5	DYs	DY4				Destino X (Baixo)
R#37	▪	▪	▪	▪	▪	▪	▪	▪	▪	DY3	DY2			Destino X (Alto)
R#38	DY1	DYo	DYg	DYs	NX7	NX5	NXs	NX4	NX3	NX2				Destino Y (Baixo)
R#39	▪	▪	▪	▪	▪	▪	▪	▪	NX1	NXo	NXs			Destino Y (Alto)
R#40	NY7	NY5	NYs	NY4	NY3	NY2	NY1	NYo	NYg	NYs				Número de pontos X (baixo)
R#41	▪	▪	▪	▪	▪	▪	▪	▪	▪	CH3	CH2			Número de pontos
R#42	CH1	ChO	CL3	CL2	CL1	CLo	MXC	MXD	MXS	DIY				Número de ponto Y (baixo)
R#43	▪	▪	▪	▪	▪	▪	▪	▪	▪	DIX	EQ	MAJ		Número do ponto Y (alto)
R#44	CM3	CM2	CM1	CMo	L03	LD2	LD1	LDo						Cor
R#45														t
R#46														Saída AC. <sup>a</sup>

Tabela 4.4: Registro de status

	b7	b5	bs	b4	b3	b2	b1	bo	F	5SF	c	554	553	552	551	5So	FLy	
ENTÃO	LPSy	ID4	ID3	ID2	ID1	IDo	FH	TR	HR	VR	BD	1	1	ED	CE	X7	X5	Estado 0
S#1	Xs	X4	X3	X2	X1	1	1											Estado 1
S#2																		Estado 2
S#3																Xô		Coluna (Baixa)
S#4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	X	Coluna (alta)
S#5	A7	A5	Sim	A4	A3	A2	A1										Ei	Linha (baixa)
S#6	1	1	1	1	1	1	1	1	ED	Sim								Linha (alta)
S#7	C7	C5	Cs	C4	C3	C2	C1	Co										Cor
S#8	BX7	BX5	BXs	BX4	BX3	BX2	BX1	BXo										Borda X (baixa)
S#9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BX	Borda X (alta)

ÿ São bits relacionados a funções que existem no V9938, mas não no V9958.

O valor de não tem sentido. .

O ID do V9958 é 000IOB .

## 4.2 Novos recursos no V9958

### 4.2.1 Rolagem horizontal

	b7	b5	b4	b2	R#25	VDS	YAE	YJK	MSK	b1	bom
SP2 HOs	H01	H06	HOs	H04	H03	H02	H01	H0b			
R#26		■									
R#27		■	■				■				

HO a a HOo é a quantidade de rolagem horizontal da tela, em unidades de 2 pontos para as TELA 6 e 7. Em outros modos de tela, defina em unidades de 1 ponto.

Se SP2 = 0 (valor inicial), o tamanho da tela horizontal é de 1 página.

Se SP2 = 1, o tamanho da tela horizontal será de 2 páginas.

Se MSK = 0 (valor inicial), a borda esquerda da tela não fica mascarada.

Se MSK = 1, os 16 pontos mais à esquerda da tela nas TELA 6 e 7 são

Neste modo, os 8 pontos na borda esquerda da tela são mascarados e a cor da borda é exibida.

De HO a a H03, a tela se move para a esquerda pelo valor definido em unidades de 8 pontos (SCREEN Para 6 e 7, mude 16 pontos (pontos únicos).

Figura 4.1: Rolagem horizontal (quando SP2=0)

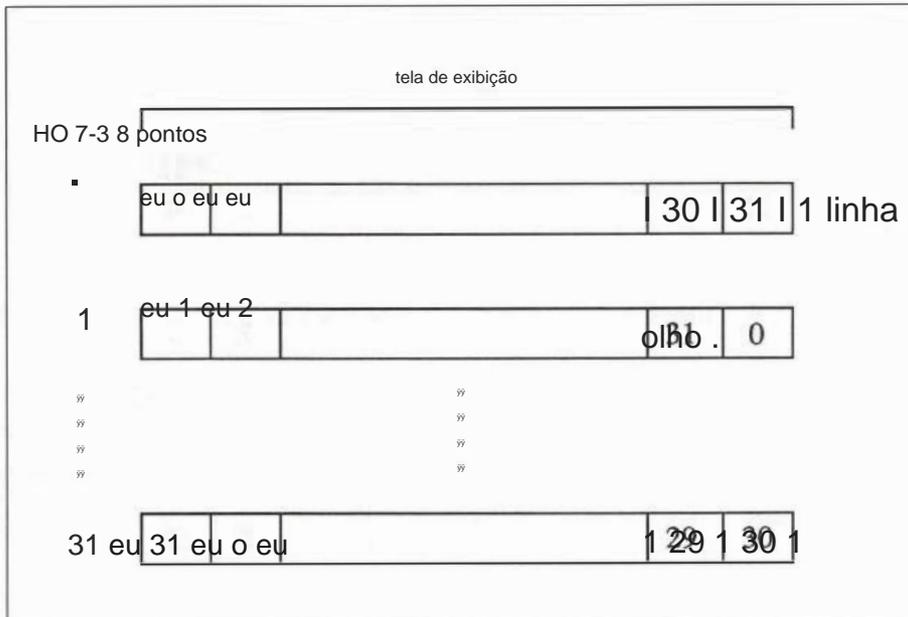
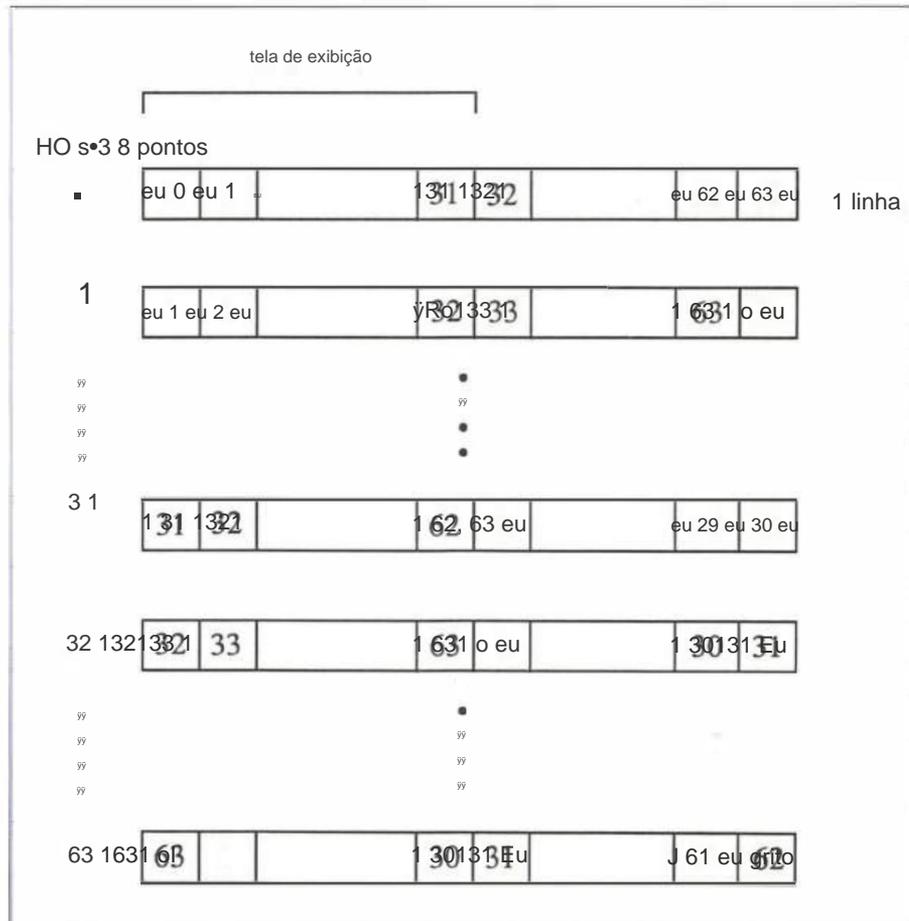


Figura 4.2: Rolagem horizontal (quando SP2=1)



Quando  $SP2 = 0$ , Os dados de uma tela são exibidos com rolagem horizontal. HO's ignorados

ser encontrado .

Quando  $SP2 = 1$ , duas telas com dados são exibidas com rolagem horizontal. padrão Defina 1 como A15 na tabela de nomes do endereço base da tabela de nomes Nomaturn O endereço base do endereço base é de 0 a 31, que é o valor de configuração A15 = 0, e 32 a 63, que é o valor de configuração. A base da mesa do gerador nomaturn e a mesa de cores conectadas diretamente entre si com A15=l. O endereço permanece inalterado e não muda durante a rolagem.

Para HO's até H03, a tela do display é movida para a direita pelo valor definido, e a tela do display tem 1 ponto único (SCREEN 6 e 7, mude 2 pontos)

## 4.2.2 Peso

	b7	b6		b4	b3	b2	b1	b0m
R#25	0	CMD	VDS	YAE	JK	MA	MSK	SP2

Se WTE = 0 (valor inicial), a função de espera está desabilitada.

Se WTE = 1, a função de espera está habilitada. VRAM acessada pela CPU

Todas as portas do V9958 são acessadas até que o acesso VRAM do V9958 seja concluído.

Demora muito para acessar o site. Acesso a registros e paletas de cores

Não há função de espera devido a processos incompletos ou dados de comando prontos.

## 4.2.3 Comando

	b7	b6		b4	b3	b2	b1	b0m
R#25	0	CMD	VDS	YAE	JK	WTE	MSK	SP2

Se CMD = 0 (valor inicial), a função de comando estará disponível apenas nos modos de tela da TELA 5 a 12. torna-se eficaz.

Se CMD = 1, a funcionalidade do comando será habilitada no modo de tela inteira.

Nos modos de tela diferentes da TELA 5 a 12, ela funciona como TELA 8. Portanto, o Os parâmetros são definidos no sistema de coordenadas XY da TELA 8. .

## 4.2.4 Exibição do método YJK

Registrar configurações

	b7	b6		b4	b3	b2	b1	b0
R#25	0	CMD	VDS	YAE	JK	WTE	MSK	SP2

Se YJK = 0 (valor inicial), os dados na VRAM são tratados no formato RGB (cada 3. , 3, 2 bits)  
As cores de exibição dos sprites são as mesmas de antes.

Se YJK = 1, os dados na VRAM são considerados no formato YJK e são usados como um sinal RGB. (5 bits cada) e gera dados analógicos do terminal RGB. Cor de exibição do Sprite A paleta está habilitada.

YAE seleciona o formato de dados YJK.

Se EJA = 0

Não há atributo. O formato dos dados é mostrado abaixo. Grupo 4 pontos consecutivos Expressado por loop.

C1 c6 Cs C4 C3 C2 C 1 c.

	KL
aaa	KH
	h
e , e,	JH

Se EJA = 1

Existe um atributo para cada ponto. O formato dos dados é mostrado abaixo. contínuo Representado pelo agrupamento de 4 pontos.

C1 c6 Cs C4 C3 C2 C1 Co e

	UM	KL
sim	UM	KH
sim	UM	h
yichiA JH		

Se A= 0 (valor inicial), então Y , J , Todo K são dados do método YJK.

Se A=1, os dados Y tornam-se um código de cores e são convertidos para RGB através do registro de cores.

Saída. J e K são dados do método YJK.

Fórmula de conversão entre sistema YJK e sistema RGB (referência)

$$\begin{aligned}
 &= Y+J \\
 11 \text{ RGB } &Y + K \\
 &\ddot{y}Y - \ddot{y} J - tK \\
 &= \ddot{y}B + tR + \ddot{y}G \\
 &= RY \\
 \text{YJK} &= GY
 \end{aligned}$$

(Nota do Editor) O valor de Y é um número inteiro de 0 a 31 se não houver atributo;

Se houver um número, é um número par entre 0 e 30. Os valores de J e K são números inteiros entre -32 e 31.

O resultado da conversão de YJK para RGB pode ser cortado entre 0 e 31.

### 4.3 Recursos obsoletos do V9958

Os seguintes recursos que existiam no V9938 foram descontinuados.

- Saída de vídeo composto
- Interface mouse/caneta óptica

(Nota do Editor) O mouse MSX usa a função de interface de mouse do V9938. Portanto, a remoção deste recurso não terá efeito.

### 4.4 Especificações de hardware V9958 (alterações)

Tabela 4.5: Mudanças de pinos V9958

番号	V9958	V9938
Nome de família	eu eu; o eu	Nome eu eu; o eu
4VRESET	I Separação de entradas VDS lógicas de 3 valores para HSYNC / CSYNC	
5HSYNC	. Saída HSYNC ou saída HSYNC 1/0 lag output	
6CSYNC		CSYNC 1/0 8
CPUCLK / VDS	. Saída CPUCLK ou saída VDS CPUCLK .	
21AVDD (DAC)	Fonte de alimentação analógica	VÍDEO
26 ESPERE	. Saída de espera 1/0	LPS
27 HRESET	I Separação de entradas de 3 LPDs lógicos oficiais de HSYNC / CSYNC	I

Registro de controle 25 Se o sinalizador VDS no bit 5 for 0, o pino 8 será definido como CPUCLK.

Ao lado da saída , Quando o sinalizador VDS é 1, o terminal 8 se torna a saída .

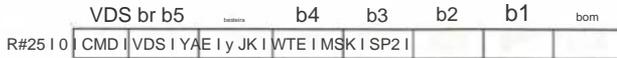


Tabela 4.6: Características DC do V9958

VRESET, item de símbolo

HRESET	最小	標準	最大	單位
	Minimum	Padrão	Maximo	
Tensão de entrada de baixo nível $V_{OL}$	-0,3		0,8	V
Tensão de entrada de alto nível $V_{IH}$	2.2		$V_{CC}$	V

HSYNC, CSYNC, CPUCLK / VDS, página de engenharia do símbolo

WAIT	Condições de medição	mínimo	padrão	unidade	máxima
V(L	tensão de saída de baixo nível $I_{oL} = 1,6mA$	0,4	v		
= 0.ImA	2,4				V

Símbolo

G, R, B	Item (itl)	Unidade	máxima	padrão	mínima	condicional
	Tensão máxima de saída VRGB31	RL = 470!			2.8	V
1	Tensão mínima de saída VnaBo	RL = 470!			2,0	V
1	$V_{p\_p}$ VnaB31 - VnaBo	RL = 470!1	Desvio		0,8	V
	da linha inferior P do DnaB	RL = 470!1			5%	%

## 4.5 V9958 e MSX2+

O componente que controla a exibição da tela do MSX é chamado de "Vídeo Display Processor" ou VDP, abreviadamente. O VDP para MSX2 era chamado de "V9938", mas em máquinas com MSX2+ ou posterior As funções foram atualizadas para "V9958". Aqui apresentamos as funções adicionadas ao V9958. fazer.

### 4.5.1 Total de 12 modos de tela

O modo de tela é o estado da tela que pode ser alterado usando o comando SCREEN do BASIC. Por exemplo, se você quiser escrever um programa BASIC com 40 dígitos horizontais,

TELA 0 : LARGURA 40

Para mudar a tela para o modo de texto (exibição de caracteres), Um lugar para desenhar gráficos

TELA 8

Se houver muitos modos de tela, como alternar para o modo gráfico (exibição de diagrama), É um pouco complicado, então se pudesse ser feito menos, não seria grande coisa. Até MSX2+ Existem razões pelas quais existem tantos modos de tela. .

Tabela 4.7: Modos de tela do MSX2+

№	Número	Método de exibição	Resolução	Caractere	numero de cores
(1)	1 caractere		80 x 24 caracteres	Especifique a cor do caractere e a cor de fundo	
(2)	2 caracteres		32 x 24 caracteres	Especifique a cor do texto e a cor de fundo	
	alfanuméricos (3)	3	256 x 192 pontos	16 cores	2 cores para cada 8 pontos horizontais
	bitmap (4)	64 x 48 pontos			16 cores
	4 bolinhas de		256 x 192 pontos	16 cores	2 cores para cada 8 pontos horizontais
	guê 5 bitmap	256 x 212 pontos		16 cores	
	Mapa de 6 bits	512 x 212 pontos		4 cores	
	7 bitmap	512 x 212 pontos		16 cores	
	Muff de 8 bits.	256 x	256 x 212 pontos	256 cores	
		212 pontos,	12.499 cores,	o que não requer uma tabela de	
	10 YJK I	RGB 256 x		caracteres de 9 pedaços ( ver texto )	
	11 YJK I	RGB 256 x 212 pontos		12.499 cores (ver texto)	
	12 YJK		212 pontos	19.268 cores (ver texto)	

- (1) 6 pontos x 8 pontos representam um caractere e exibem caracteres alfabéticos e katakana
  - (2) 8 pontos x 8 pontos representam 1 caractere e exibem caracteres alfabéticos Exibe katakana e hiragana
  - (3) Uma combinação de padrões de 8 x 8 pontos. Criando uma tela (4)
- Com a exibição de bitmap, é possível exibir a cor de um ponto independentemente das cores vizinhas.

A décima questão é a velocidade. Quando você envia caracteres para a tela no modo gráfico, o display Isso leva tempo. Portanto, ao inserir um programa do zero, você pode inserir números e kanji.

Embora não seja utilizável, o modo de texto é conveniente porque é exibido rapidamente.

A segunda razão é que quanto mais funcional for a tela (quanto mais pontos e cores ela puder exibir), mais Por exemplo, os dados da tela SCREEN 8 requerem 54.272 telas.

Há também uma lista de emblemas, que só pode conter 12 imagens em um disco (2DD). o Portanto, os jogos em cartucho ROM possuem um número limitado de cores, mas possuem menos dados. A TELA 2 é frequentemente usada para economizar memória porque é cada vez mais rápida.

Terceiro, adicionamos recursos enquanto mantemos a compatibilidade, permitindo vários modos de tela. tornou-se necessário. O primeiro MSX tinha quatro tipos de telas, SCREEN 0 a 3.

Havia apenas o modo enxuto. Isso se tornou o MSX2. Para exibição gráfica de alta resolução Adicionada TELA 4 a 8. Além disso, o MSX2+ usa SCREEN para aumentar o número de cores. Portanto, 10 a 12 foram adicionados.

A Tabela 4.7 resume esses modos de tela. Mas na realidade, a tela Este não é o fim dos problemas com o entrelaçamento, que duplica a resolução vertical. Há também um modo kanji que foi adicionado recentemente no MSX2+.

#### 4.5.2 Controlando registros VDP

O VDP que controla a tela do MSX é semelhante ao coração da máquina, a CPU.

Existe também um "registro". E esse registro VDP é a porta 1/0.

Porque a CPU controla o VDP pressionando o botão. Os registradores mais comumente usados são chamados de "registros de controle". Para a CPU saber o estado do VDP Os registros usados para esse propósito são chamados de "registros de status". Além disso, PDV disso, existe um "registro de comando" que a CPU manipula para executar instruções de alto nível. Qual é o objetivo deste livro? Por não ser utilizado no programa, sua explicação será omitida.

O endereço da porta 1/0 que conecta a CPU e o VDP geralmente é de 98H a 9BH. são. Contudo, para ser mais preciso, como mostrado na Tabela 4.8, o conteúdo dos endereços ROM 6 e 7 são

Tabela 4.8: Portas de E/S VDP

Nome da porta	R/W	Endereço de E/S Porta	Propósito
0 READ		Endereço ROM 6 conteúdo VRAM lido	
Porta 1 READ		ROM endereço 6 conteúdo + 1 registro de status	
Porta 0 WRITE		ROM endereço 7 conteúdo gravação VRAM	
Porta 1 WRITE		ROM endereço 7 conteúdo + registro do controlador	
Porta 2 WRITE		ROM endereço 7 conteúdo + 2 f Deixe o registro	
Porta 3 WRITE		ROM endereço 7 conteúdo + 3 registro de especificação indireta	

Está decidido. Isto foi feito para permitir que o VDP fosse adicionado fora da unidade principal do MSX.

Observe que mesmo que a mesma porta seja usada para escrita e leitura, o endereço da porta I/O é o mesmo durante a escrita e a leitura.

Observe que os valores podem ser diferentes. .

Agora, para definir o valor do registro de controle VDP, primeiro defina o valor na Tabela 4.8.

Grave os dados que deseja definir na porta 1 e, em seguida, grave os dados na mesma porta 1. “Número de registro 128”. Esta condição de “continuamente” é surpreendentemente importante e é necessário escrevê-la duas vezes. Se ocorrer uma interrupção durante o processo, o VDP ficará confuso. Então, neste caso, primeiro Como escrever dois dados em sucessão após desabilitar interrupções usando o comando “DI” é comumente usado.

A propósito, os registradores de controle são registradores somente de escrita. Uma vez definido O valor armazenado não pode ser lido. Portanto, você pode escrever apenas bits específicos de um registrador. Se você quiser alterá-lo, será inconveniente. Então geralmente o controle

Os valores gravados nos registradores também são gravados na área de software do sistema (RAM) conforme mostrado na Tabela 4.9. O método é anotar. Por exemplo, o bit 4 do registrador de controle I

Se quiser mudar o bit 4 para 1, leia o conteúdo do endereço F3E0H na RAM e mude o bit 4 para 1.

Em seguida, o valor é escrito no registrador de controle 1 e no endereço RAM F3E0H.

É assim que é. Devo apresentá-lo mais tarde? No exemplo de programa de interrupção de linha de varredura na Listagem 4.9:

Neste exemplo, uma sub-rotina chamada “WRTVDP” realiza a mesma operação.

A seguir, explicaremos como ler o valor do registro de status. Este é um controle

Defina o número do registro de status que deseja ler no registro de função 15 e defina a porta VDP.

O procedimento de leitura do valor do registrador 1 e retorno do registrador de controle 15 para 0 é realizado por meio de uma interrupção.

A ideia é praticar o ato enquanto o ato é proibido. No programa de exemplo da Listagem 4.9,

A sub-rotina “\_VDPSTA” corresponde a isto.

Tabela 4.9: Localização dos registradores de controle

レジスター番号 Número de registro	VDP 関数番号 Número da função VDP	保存番地 Salvar endereço	ラベル
■	■	F3DFH	RG0SAV
⋮	⋮	⋮	⋮
7	7	F3E6H	RG7SAV
8	9	FFE7H	RG8SAV
⋮	⋮	⋮	⋮
...	...	FFF6H	RG23SA
...	26	FFFAH	RG25SA
26	27	FFFBH	RG26SA
27	28	FFFBH	RG27SA

Tabela 4.10: Outras áreas úteis de trabalho do sistema

Endereço	Etiqueta	significado
F341H	RAM ADO página 0	Número do slot de RAM*
F342H	RAMAD1 Página 1	Livro de números do slot de RAM
F343H	RAMAD2 Página 2	Número do slot de RAM*
F344H	RAMAD3 Página 3	Número do slot de RAM*
	Número da página de exibição	FAF5H DPPAGE
	Número de página ativa	FAF6H ACPAGE
	Gancho de interrupção	FD9AH H.KEYI
FD9FH H	Temporizador TIMI	prejudica l; gancho

\*Válido apenas se um disco estiver disponível.

Tabela 4.11: Áreas de trabalho do sistema adicionadas e alteradas no MSX2+

Endereço	Etiqueta	significado
MODO OFAFCH		Veja a próxima tabela
Área de trabalho de OFAFDH	NORUSE	Kanji Dofuipa
OFDOAH SLTWRK+1		Área de trabalho de Kanji Draper
⋮	⋮	
OFDOFH SLTWRK+6		
OFFFAH RG25SA		Salvando registros VDP
OFFFBH RG26SA		
DE FF CH RG27SA		

Tabela 4.12: Detalhes do endereço OFAFCH (MODE)

Endereço	significado
b7 1	significa bakatakana, 0 significa hiragana
	Se for 1, há ROM de kanji de segundo nível.
b6 bs 1	então TELA 1 1 0 então TELA 10 Usado dentro de b4
	Se for 1, mascare 3FFFH para endereço VRAM com SCREEN 0-3
b3 b2	Capacidade VRAM
b1 00	16 KB 0 1 64 KB b. 1 Se; conversão, 1 0 1 28 KB
	romanji kana

Também e também , MSX2 e 2+ ROM possuem uma BIOS com as mesmas funções dessas sub-rotinas.

Como existe uma sub-rotina, normalmente você pode usar o BIOS sem criar uma sub-rotina sozinho. Mas isso Seu BIOS está localizado na sub ROM ou chama a sub ROM durante o processamento, então

A desvantagem é que leva algum tempo. Portanto, para processamento como interrupções de linha de varredura, É inconveniente, então não uso o BIOS.

As Tabelas 4.10 a 4.12 resumem outras áreas de trabalho do sistema para sua referência.

Eu gostaria que você pensasse sobre isso.

### 4.5.3 Registros V9958

A Figura 4.3 mostra os três registradores de controle adicionados ao V9958. Esses registros são somente para gravação e os valores escritos são armazenados no software do sistema na Tabela 4.9. Não há registro de controle 24 registrado no consultador, e registro número 1 e Observe que os números das funções VDP do BASIC são diferentes.

A maioria dos recursos adicionados no V9958 são controlados pelo registro de controle 25.

Apresentarei primeiro as funções restantes, deixando a famosa exibição YJK (imagem natural) e a rolagem horizontal para depois.

Vamos fazê-lo.

Certifique-se de escrever 0 no bit 7 do registro 25. O bit 5 é "VDS" É chamado e controla a função do pino 8 do VDP. Mas é apenas um fu normal. programa é este bit. É proibido reescrever o texto.

Além disso, se você escrever 0 no bit 6, funcionará para as TELA 5 a 8 como o V9938. "Comandos VDP" só podem ser usados quando .

É por isso que você pode usar comandos VDP no modo de tela. Este comando VDP é um BASIC Uma função que permite ao VDP executar tarefas como comandos COPY e LINE. Vamos entrar em detalhes. Os comandos VDP para telas diferentes das TELA 5 a 8 usam 128 KB.

localização na RAM de vídeo, Assim como na TELA 8, a coordenada X vai de 0 a 255 , A coordenada Y é Especifique um valor de 0 a 511.

Além disso, escrever 1 no bit 2 habilita a "função de espera" do VDP. criança Isso significa que quando a CPU lê e grava a RAM de vídeo, se a CPU operar muito rápido, o VDP Esta função envia um "sinal WAIT" à CPU para fazê-la esperar. , para registro de paletes Entretanto, não há função de espera para gravações e transferências usando comandos VDP.

### 4.5.4 Rolagem horizontal com VDP

Para rolagem horizontal, existem dois métodos: usar dados de imagem para uma tela e usar dados de imagem para duas telas.

Existe uma maneira de usá-lo . Deixe-me explicar cada caso por vez . Primeiro, se o bit .

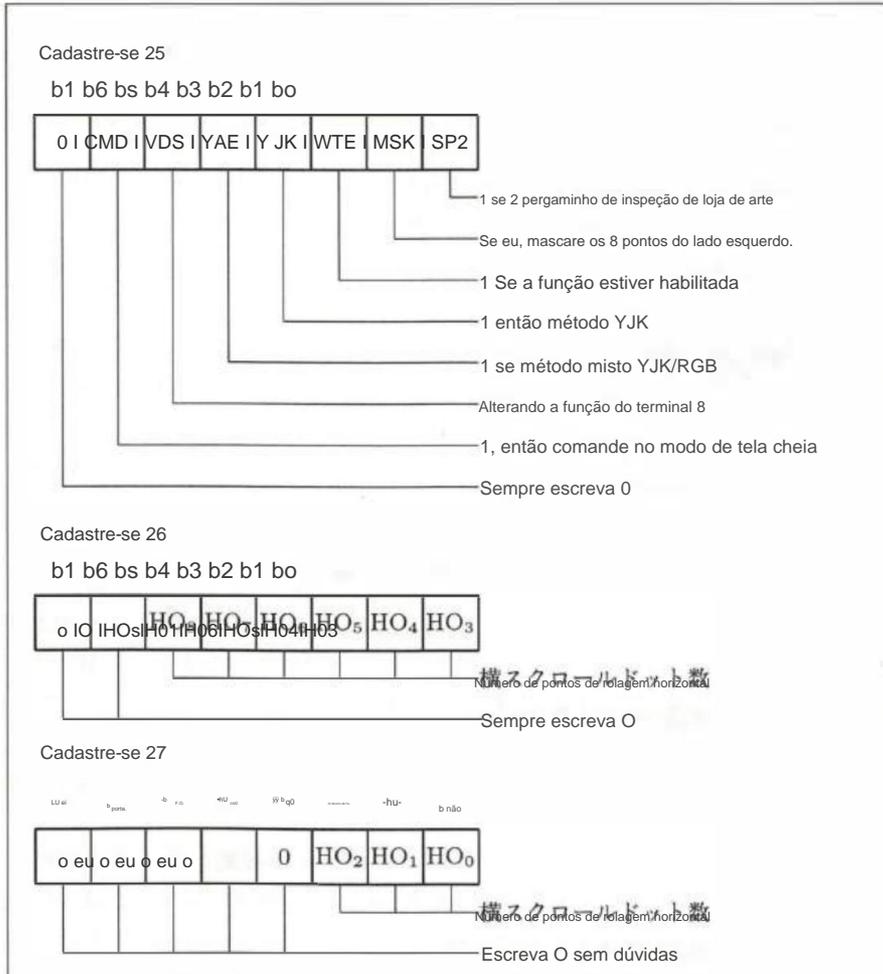
0 "SP2" do registrador de controle 25 for 0, então o bit 0 "SP2" do registrador de controle 25 é 0.

O número de pontos de rolagem que causam a rolagem horizontal usando dados de uma tela, conforme mostrado acima.

Um valor de 0 a 63 no registro 26 pode ser especificado nos registros 26 e 27.

Quando você escreve , a tela rola para a esquerda naquele valor x 8 pontos e grava no registro 27.

Figura 4.3: Lista de funções de registro de controle adicionadas ao V9958



Se você escrever um valor entre 0 e 7, ele se moverá para a direita desse número. Porém, nas TELA 6 e 7, o especificado  
 Observe que a rolagem ocorrerá o dobro de pontos que você definiu..

Aparece quando você aumenta o número , a tela rolará para a esquerda e a parte saliente ficará na borda direita.  
 sequencialmente até 255.

Por outro lado, se o bit 0 do registro 25 for 1, conforme mostrado na parte inferior da Figura 4.4, serão exibidas duas telas com dados.  
 A parte especificada dos dados da imagem é exibida e a rolagem horizontal começa.

Os dados da imagem neste momento são armazenados nas páginas O e L ou 2 e 3 da RAM de vídeo.

Figura 4.4: Dois tipos de mecanismos de rolagem horizontal

Quando SP2=0

+1111+ Número de pontos especificados nos registros 26 e 27

registro de controle  
Pontos designados por 26 e 27

Conforme você aumenta o número, ele aparecerá na tela.  
A imagem mostrada rola para a esquerda.  
projetando-se da borda esquerda da caixa de arte  
A parte que você adicionou aparece na borda direita.  
Uma caixa de bactérias à esquerda e à direita  
Isso coloca você em um estado conectado.

111 Conteúdo VRAM111

Quando SP2=1

+1111 + número de pontos especificados nos registros 26 e 27

+111 página VRAM O VjiVRAM página 1111+

Uma combinação de páginas de RAM de vídeo 0 e 1 (ou 2 e 3).  
Somente a parte especificada é exibida em duas telas horizontais de dados de imagem.  
Então, conforme a tela rola, as duas imagens são conectadas.  
É hora de mudar.

são. Se as páginas forem 0 e 1, defina a página de exibição como l, e se as páginas forem 2 e 3, defina-a como te. estômago  
Tente definir a página de pulverização . O número de pontos para rolagem horizontal é de 0 a 511 (ou seja, 1  
para 3 (duas vezes a rolagem da tela).

Além disso, quando o bit 1 do registro 25 é definido como 1, os 8 pontos na extremidade esquerda da tela (SCREEN6  
e 7, os 16 pontos) não são exibidos e a cor ao redor da tela é exibida.  
Isso é um problema ao rolar horizontalmente, especialmente ao rolar horizontalmente usando l tela de dados.  
Isso é útil para ocultar peças que se projetam da superfície e que logo aparecerão do outro lado. Lista 4.1  
Publiquei um exemplo de programa de rolagem horizontal abaixo, então fique à vontade para usá-lo como referência. .

## Listagem 4.1 (HSCROLL.BAS)

```

100 'hscroll.bas
110 ' por nao-i em 26 de outubro de 1989 1 20 ' 130
ONSTOP
GOSUB 290 : STOP ON 140 DEFINT AZ : TELA 5
150 COR 15 ,1,1: CLS 160 LINE (0,0)ÿ
(255 ,211. )

170 CONJUNTO PÁGINA 1,1: COR 15, 8,1: CLS 180 LINHA
(0,0)Um (255 ,211)
190 V6=VDP ( 26): VDP (26)=V6 OU 3 200 ÿ ÿÿ ho
scroll 210 FOR V7=0 TO
63 220 GOSUB 320 : VDP ( 27)
=V7 : VDP ( 28)=7 230 FOR V8= 6 TO 0 STEP -1 GOSUB 320: VDP
(28) =V8 240 250 NEXT VS 260 NEXT V7 270
VDP      GOTO 200 280 ÿbookÿ restaurar
290 STOP OFF: VDP
(26) =V6: SET
PAGE 0,0 300
COLOR 15 ,4,7: TELA 0: END 310
ÿ *Livro principal aguarde próximo vsync 320 TIME=O 330 IF
TIME=O GOTO 330 340 IF (VDP (-2) AND 64) =0
GOTO 340 350 RETURN

```

### 4.5.5 Sob nenhuma circunstância você deve usar truques.

Sempre escreva 0 ou 1 no registro VDP.

Há algo que foi especificado, como ignorar esta especificação e escrever um valor estranho.

Você nunca sabe o que vai acontecer, então você nunca deve fazer isso.

Além disso, o comportamento do VDP não é determinado pelas especificações, como a combinação de YJK=O e YAE=1.

Existem algumas configurações que não estão especificadas, mas você também pode usar esses "truques" que não estão escritos nas especificações.

**Não posso. Mesmo que a máquina que você possui funcione sem problemas, pode**

**Acabei de me mudar . Não há garantia de que funcionará corretamente em outras máquinas MSX.**

não tenho. Além disso, embora este truque seja eficaz para o V9958 atual, é importante observar que o VDP será melhorado no futuro.

**Se um fabricante que não seja a Yamaha fabricar um VDP compatível com V9958 (se**

Todos os rolos (V9958) são fabricados pela Yamaha, portanto os mesmos truques podem não ser necessariamente eficazes.

Da mesma forma, você não deve usar nenhum dos truques da CPU (oficialmente chamados de "instruções indefinidas").

Sem chance. Mesmo em exemplos anteriores , Turbo HC-95 do Pictor devido ao truque do Z80

Software que ficou fora de controle (usando HD64180, uma CPU compatível com versões superiores do Z80)

Havia ar.

## 4.6 Dissecando o método YJK

### 4.6.1 Transmissão televisiva e formato YJK

O modo de imagem natural do MSX2+ usa o formato YJK em vez do formato RGB tradicional.

Este sistema YJK é semelhante aos sinais de transmissão de televisão em cores.

Muitos leitores deste livro podem não saber disso, mas no passado as transmissões de televisão eram em preto e branco. Naturalmente, todos os receptores de televisão eram receptores preto e branco. Depois disso, quando começamos a transmissão em cores, duas coisas se tornaram um problema. A tela colorida é separada em três cores: RGB (vermelho, verde e azul). Ao transmitir, são usados três canais. Portanto, mesmo um receptor preto e branco pode receber dados coloridos. Deve ser capaz de receber transmissões.

Portanto, as propriedades do olho humano foram utilizadas. A retina possui células que detectam o brilho e células que detectam a cor. O número de células que detectam o brilho é grande e o número de células que detectam a cor é grande. Como existem tão poucas cores, o olho humano é insensível às mudanças de cor.

Na transmissão de televisão em cores, são usados um sinal Y representando "brilho" e um sinal UV representando "matiz". As cores são expressas por meio de combinações. Aproveitando o facto de o olho humano ser insensível a mudanças de cor, a quantidade de sinal UV (tecnicamente referida como banda de frequência) é pequena e um canal pode enviar sinais YUV de uma só vez. Além disso, o sinal Y que representa o brilho é igual ao sinal de transmissão em preto e branco. Portanto, quando você assiste a uma transmissão colorida em um receptor em preto e branco, apenas o sinal Y é exibido e uma imagem normal em preto e branco é exibida.

Parece um rosto.

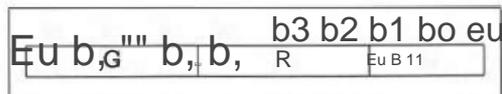
Dessa forma, as televisões podem exibir uma tela colorida sem aumentar a quantidade de ondas de rádio (ou seja, o número de canais). O sistema YUV foi usado para transmitir o vídeo. Por outro lado, com o MSX2+, você pode aumentar a RAM de vídeo. Para aumentar o número de cores sem aumentar o número de cores, utiliza-se o método YJK, que é semelhante ao método YUV.

### 4.6.2 Estrutura de dados do método RGB e método YJK

#### Método RGB (TELA 8)

Na TELA 8, o brilho de R, G e B é representado por 3, 3 e 2 bits, respectivamente. Cada ponto pode exibir qualquer cor entre 256 cores.

Figura 4.5: Estrutura de dados da tela do método RGB



## Método YJK (TELA 12)

A TELA 12 usa o método YJK, que consiste em 4 grupos de 4 pontos horizontais. Figura 4.6 De Yo a Y3, o brilho de cada ponto é de 5 bits (ou seja, 32 passos) e K e J são de 4 bits. A tonalidade de todo o ponto é expressa em 12 bits (4.096 cores). Dados do formato YJK para o formato RGB O método para convertê-lo é o seguinte.

$$\begin{aligned} &= Y+J \\ &= Y+K \quad I \\ \text{RGB} &= 0,25Y - 0,5J \quad 0,25K \end{aligned}$$

Porém, os valores de Y variam de 0 a 31, e os valores de J e K variam de -32 a 31. Calculado usando a fórmula acima Se os valores calculados de R, G e B forem menores que 0, 0 é gerado em vez desse valor e Da mesma forma, se for maior que 31, 31 será gerado. Este tipo de processamento pode ser realizado de "0 a 31". recorte.

Por exemplo, os 4 bytes de dados a seguir são  $y = 0$ ,  $J = K = -32$  representa preto Você sabe o que é? Vamos fazer os cálculos em uma calculadora.



Por outro lado, vamos considerar a conversão de dados RGB em dados YJK.

Em outras palavras, a fórmula de conversão para RGB é considerada uma equação linear simultânea tridimensional e é expressa como Y, J, K

Tudo que você precisa fazer é resolver cada um. É um problema de nível de "álgebra/geometria" do ensino médio. A resposta é a próxima As três fórmulas dadas em Eu fiz isso corretamente?

$$\begin{aligned} &= (2R + G + 4B)/8 \\ &= (6R - G - 4B)/8 \\ \text{YJK} &= (-2R + 7G - 4B)/8 \end{aligned}$$

Mesmo ao exibir imagens no formato YJK, os sinais emitidos pelo VDP são conforme explicado acima. O sinal é convertido no mesmo sinal RGB analógico de antes e em um sinal composto (vídeo) usando a fórmula. Não há necessidade de um monitor de TV especial para MSX2+.

Nas telas no formato YJK, 1 byte corresponde a 1 ponto, semelhante à TELA 8. Responder. No entanto, os valores de J e K são especificados a cada 4 pontos horizontais, por exemplo,

PSET (0,0) ,0

Quando você faz isso, o valor de K dos 4 pontos de (0,0) a (3,0) muda. Isso . Além disso resulta em "descoloração". Em relação a isso, Explicarei novamente mais tarde .



Está sendo Conforme explicado até agora, o valor de Y, que representa a luminância (brilho), e a cor Este é um método de especificação de cores usando combinações de J e K, que representam fases.

A Listagem 4.2 é um programa que exibe todas as cores que podem ser representadas por YJK. apenas Porém, aqui usamos a instrução PSET 32.768 vezes para definir os valores de J e K. Demorou muito, então melhorei a lista usando os comandos LINE e COPY.

4.3 . Escreva o valor de K em PSET para apenas 1 yj à esquerda. , Copiando usando o comando COPY.

### Listagem 4.2 (YJKI.BAS)

```

100' Amostra de cores da tela 1 1 e 1 2 por PSET 110'
por nao-1 em 2 de novembro de 1988.
120 CHAMADA KANJIO: LARGURA 64: DEFINT AZ: YY=O
130 LIMPAR: DEFINIR AZ: YY=O
140 EM PARADA GOSUB 400: PARAR EM
150 TELA 12: COR &HF8 ,0,0:CHAMAR CLS
160' Conjunto J e K da VRAM
170 PARA K=-32 TD 31 : YP=1 12+K+K
180 PARA J=-32 TD 31 : XP=128+JHo4
190 PSET (XP , YP), K E 7
200 PSET (XP,YP+1) , K E 7
210 PSET (XP+1 , SIM) , (K E 56) ¥ 8
220 PSET (XP+1 , YP+1 ),(K E 56)¥8
230 PSET (XP+2 , YP), J E 7
240 PSET (XP+2 , YP+1), J E 7
250 PSET (XP+3 , SIM) , (J E 56) ¥8
260 PSET (XP+3 , YP+1 ),(J E 56) ¥8
270 NEXT : NEXT
280 'Altera o valor de y
290 SC=12 : NS=1 : TELA 12
300 PARA Y=1 TD 3 1: GOSUB 360 • PRÓXIMO
310 PARA Y=30 TD 0 PASSO -1: GOSUB 360: PRÓXIMO
320 SC=11 : NS=1 : TELA 11
330 PARA Y=2 TD 30 PASSO 2: GOSUB 360: PRÓXIMO
340 PARA Y=28 TD 0 PASSO -2: GOSUB 360: PRÓXIMO
350 VAI PARA 280
360 ' Sub-rotina para reescrever Y na VRAM
370 IF NS THEN LOCATE 1,0: PRINT USING N yÿ"; " TELA
SC:NS=O 380
LOCATE y1: PRINT USING"O valor Y (brilho) é
##é." yS
390 LINHA(0,48) - (255 R : , 175) , (Y XOR YY) Mamoru 8 , ----, XO
YY=Y : RETORNO
400' chamado para parar
410 TELA O: COR 15,4,7:FIM

```

## Listagem 4.3 (YJK2.BAS)

```

100 ' Amostras de cores para tela 1 1 e 1 2
110 ' por nao-i em 2 de novembro de 1988 120
CALL KANJIO : WID 64 : DEFINT AZ : YY=O 130 CLEAR: DEFINT
AZ : YY=O 140 ON STOP GOSUB 380 :
STOP ON 150 SCREEN 12: COLOR &HF8 ,0,0:
CALL CLS 160 'V Conjunto J e K de RAM

170 PARA Kyÿ32 TD 31 : YP=1 12+K+K 180
PSET (O,YP) ,K E 7 190 PSET
(O,YP+1) ,K E 7
200 PSET (1, YP) , (K E 56)¥8 210
PSET (1,YP+1) , (K E 56)¥8: PRÓXIMO
220 PARA J=-32 A 31: XP=128+J*4 LINHA
230(XP+2 ,48) - (XP+2 , 175) , J E 7
LINHA 240(XP+3 ,48) - (XP+3 , 175) , (J E 56)¥8
250 CÓPIA(0 ,48) -(1 , 175) PARA (XP , VOCÊ): PRÓXIMO
260 'altera o valor de y
270 SC = 12: NS = 1: TELA 12 280 PARA Y = 1
A 3 1: GOSUB 340: PRÓXIMO 290 PARA Y = 30 A 0 PASSO -1:
GOSUB 340: PRÓXIMO 300 SC = 11: NS = 1: TELA 11 310 FOR Y=2 A 30
PASSO 2: GOSUB 340: NEXT 320 FOR Y=28 A
0 PASSO -2: GOSUB 340: NEXT 330 GOTO 260 340 ' Sub-rotina para reescrever
Y de VR AM

350 IF NS THEN LOCATE 1,0:PRINT USING" SCREE N ÿÿ" ;

##é." ÿS
370 LINHA(0,48) - (255 R : , 175) , (S X OU AA) *8 , BF,XO
YY=Y : RETORNO
380 ' chamado para parar
390 TELA O: COR 15,4,7:FIM

```

## 4.6.4 É um desempenho lógico mortal.

Operação lógica significa operação lógica em japonês. . Para cada bit do número binário significa um cálculo a ser realizado. Isso inclui E , OU , XOR , Existem quatro tipos de NÃO. .

Uma operação AND é uma operação que define bits que são 1 em dois valores como 1. Ta e

Se for assim,

```

xÿÿB00000011
Y%=&B00000101

```

Ao realizar uma operação AND em , bit 0 (bit mais à direita) da variável X3 e da variável Y3 Como apenas é 1,

X% E Y%=&B00000001

É por isso. Da mesma forma, na operação OR, ambos ou um dos dois valores tornam-se 1.

Na operação XOR, o bit que é definido como 1 em apenas um dos dois valores é definido como 1.

Além disso, NOT é uma operação que inverte cada bit de um valor, o que inverte os dados da imagem.

A conversão e a gravação também são chamadas de operação PRESET. É difícil entender quando explicado em palavras.

Portanto, resumi o conteúdo na Tabela 4.13.

Tabela 4.13: Operação lógica

記号	意味	値
PSET	Grava a cor especificada como está.	
	Escreva o produto lógico de AND <b>cor redundante e</b>	0011 E 0101 y 0001
	Escreva a soma lógica da cor redundante e	0011 OU 0101 y 0111
	Escreva o OR exclusivo da <b>cor redundante XOR e 0011</b>	XOR 0101 y 01 10
PRESET	Inversão de bit de gravação da cor especificada NOT 0101 y 1010	

Agora, para lidar com RAM de vídeo no MSX2 e MSX2+, precisamos usar estas operações lógicas. É necessário considerar a situação. Por exemplo, no programa da Listagem 4.3 postado anteriormente, Considere aumentar o valor Y da RAM de vídeo de 1 (00001 em binário) para 2 (00010).

Vamos ver. Em outras palavras,

```
b1 b5 bs b4 b3 b2 b1 bo
0 0 0 0 1 ? . . .
```

O conteúdo da RAM de vídeo é

```
b1 b5 bs b4 b3 b2 b1 bo
0 0 0 1 0 ? . . .
```

Eu tenho que mudar para . Os bits 2 a 0 contêm o valor J ou K, então altere isso.

Não desista.

O autor pensou em multiplicar 3, que é o XOR do valor Y original 1 e do valor alvo 2, por 8,

```
LINHA (0,48) - (255 , 175) , ---- , XOR
```

realizar . O valor Y muda de 1 para 2 em I reescritas. Esta é a lista 4.3, linha 370,

```
LINHA (O , YOyUm (255 , A0+127) , (S X OU AA) *8 , ---- , XOR
```

O significado de Y é o valor alvo de Y e yy é o valor original de Y. .

Se você especificar uma operação lógica com o comando LINE, será feita uma tentativa de escrita. O resultado da operação entre a cor atual e a cor original é gravado na RAM de vídeo. . Ta e Por exemplo, para a tela SCREEN 12,

LINHA (0,0) - (255 , 21 1) , &B1 111 1000 , , E

Vamos fazê-lo , Os bits 7 a 3 da RAM de vídeo permanecem como estão, bit 2 ou  
A partir daí, até o bit 1 passa a 0. , a tela inteira é exibida em sombreado preto e branco.

#### 4.6.5 Mudança de cor

A estrutura de dados da tela YJK é resumida anteriormente na Figura 4.6. . Igual à TELA 8  
Basicamente, cada ponto L na tela corresponde a um byte de RAM de vídeo. Largura 256 x Altura 212  
A tela de pontos é exibida usando 54.272 bytes de RAM de vídeo. No entanto, especifique o matiz  
Os valores de J e K são especificados para cada 4 pontos horizontais, por exemplo, na TELA 12,

PSET (3 , 0) ,3

Quando executamos, obtemos não apenas 1 ponto em (3,0), mas também 4 pontos J de (0,0) a (3,0).  
O valor muda e esta área fica vermelha. Salve a tela SCREEN 12 como "sample.s12"

Se você salvar em um arquivo chamado "color change" e então executar a Listagem 4.4, você verá o que é chamado de "color discoloration".

Deve ser pesado. Vamos experimentar. Como regra geral, a TELA 12 permite que caracteres e linhas  
Não pode ser exibido.

#### Listagem 4.4 (S12.BAS)

```
10 (Exemplo de descoloração)
20 CALL KANJI 30
TELA 12: COLOR &HF9,2 40 BLOAD
"amostra. s12",S 50 LINE (0,0)Y(211,
211) '3 60 LOCATE 4,6: COLOR &HF9,2
70 IMPRIMIR "TELA 12 Então as cores
mudam."
80 VAI PARA 80
```

#### 4.6.6 Qual é a diferença entre a TELA 10 e a 11?

A estrutura de dados das TELA 10 e 11 está resumida na Figura 4.7 acima. Ambas as máquinas  
Funcionalmente eles são exatamente iguais. Essas telas também são usadas ao exibir a tela BLOAD.

Não há diferença no modo enxuto. Então qual é a diferença? Com comandos BASIC, etc.

Surge um problema ao escrever figuras ou caracteres na tela.

A Listagem 4.5 é um exemplo de programa que mostra a diferença. Dados de tela gravados no formato YJK  
BLOAD os dados e

CÍRCULO (128 , 106) , 100,6

Na TELA 10, o bit 3 da RAM de vídeo torna-se 1, e 6 (o número de cor vermelha, Pedaco especificado pela instrução do círculo; 0110 em binário) é escrito nos bits 7 a 4.

É sugado. Portanto, você pode desenhar um círculo vermelho sem destruir a tela YJK de fundo.

No entanto, na TELA 11, 6 (ou seja, 00000110 em 8 bits) é enviado diretamente para a RAM de vídeo. Como você pode ver neste exemplo, a TELA

10 é usada para desenhar caracteres e figuras sobrepostos em uma tela YJK. Você precisa usar a TELA 11 para processar os dados da tela YJK.

### Lista 4.5 (SIO.BAS)

100 (Como evitar a descoloração)

110 TELA 12 120 BLDAD

\*amostra. s 12", S 130 TELA 10 140 CÍRCULO

(128 150 PARA I% = 1 A

50: BEEP: PRÓXIMO 160 , 106) , 100 , 6

TELA 12 170 BLOAD \*amostra. s 12", S 180 TELA 11

190 LE (128 200 IR PARA

200

, 106) , 100 , 6

### 4.6.7 Como usar legendas mesmo na TELA 11

A Listagem 4.6 é um programa que grava caracteres na tela da TELA 11. caracteres de exibição Acabei de escrever que a TELA 10 é usada para impressão, mas devido ao comando PRINT, a TELA 10 é usada. Há momentos em que 11 é mais conveniente.

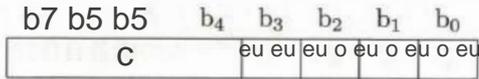
Por exemplo, se você usar os comandos LINE ou PUT KANJI na TELA 10,, A tela YJK tem Linhas e caracteres são escritos no formato RGB. Entretanto, se você usar o comando PRINT, um fundo quadrado aparecerá. Um quadro aparece e o texto está escrito dentro dele. Parece que não está sendo usado como legenda.

Não é adequado para ouriços-do-mar.

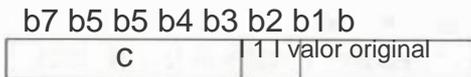
O que entra em jogo é mudar para a TELA. Página 1 da RAM de vídeo com a instrução SET PAGE 11 e BLOADing a tela de fundo lá. Defina a cor de primeiro. Em seguida, redefina a página para 0 e, em seguida, plano para 7. Mude a cor de fundo para 0 e exiba os caracteres usando o comando PRINT. Além disso, esses personagens de, Especifique "TAND" no comando COPY para mover para a parte desejada dos dados da imagem na página 1. É por isso que faço cópias.

TAND não copia a parte com número de cor 0 (transparente), mas faz AND apenas as partes de outras cores. Uma função que permite pesquisar enquanto realiza cálculos. Como resultado, a caixa de caracteres com o número de cor 0 é ignorada e Somente os caracteres escritos em 7 (azul claro) serão copiados. RAM de vídeo da área onde você deseja exibir os caracteres Isso significa que os bits 7 a 3 se tornam 0.

A seguir, escreva a mesma letra na página O na cor especificada por (C×16+8). Este valor de C é Número da cor inserido no início do programa.



Os bits 7 a 4 são o número da cor desejada e os bits que especificam a exibição RGB.  
O bit 3 torna-se 1 e os bits restantes de 2 a 0 tornam-se 0. Este personagem, desta vez TOR  
Especifique e copie , O conteúdo da RAM de vídeo dos personagens é o seguinte:



Isso significa que os caracteres podem ser exibidos no formato RGB sem destruir a tela YJK de fundo.

Desta forma, quando caracteres escritos em locais diferentes são copiados através do comando COPY, eles são copiados para o comando PRINT.

Pode compensar a desvantagem de não ter uma função de operação lógica .

O método YJK do MSX2+ é um modo difícil, , Se você usá-lo com sabedoria, poderá obter resultados surpreendentes.  
mas factível. Vamos todos fazer o nosso melhor e pesquisar.

## Listagem 4.6 (SH.BAS)

```

100' TELA 1 1 Terror desligado.
110 'por nao-i em 4. nov. 1988 1 20 SCREEN
O:WIDTH 32: CALL KANJIO 130 DEFINT AZ: ON STOP
GOSUB300: STOP ON 140 FILES: PRINT: INPUT "nome do arquivo
"; FF$
150 ABRIR FF$ PARA ENTRADA COMO #1 : FECHAR #1 ;

180 ENTRADA "X(O ... 240)";X
190 ENTRADA "Y(O ... 196)" :Y
200 TELA 1 2 : SET PÁGINA 1.: BLOAD FF$,S 210 TELA 11 220 SET
PÁGINA O,O:COLOR
7,0:CALL CLS 230 L=LEN(SS$) *8-1 240 LOCATE O ,O:IMPRIMIR
SS$ 250 CÓPIA (0 0) - (L, 15)

, , 0 PARA(X, S) , 1,TAND
260 LOCALIZAR 0,0:COR C* 16+8 ,0: IMPRIMIR SS$ 270 CÓPIA
(0 0 TO (X , 0) - (L, 15) , , S) , 1,TOR
280 DEFINIR PÁGINA 1, 1
290 GOTO 290 300
' *** chamado por STOP ***
310 CONJUNTO PÁGINA O,O: COR 15,4,7

```

#### 4.6.8 Um truque para exibir texto na TELA 12

Em princípio, não é possível exibir caracteres ou linhas na TELA 12, que está totalmente no formato YJK. Não tenho. No entanto, existe um truque para exibir texto branco usando operações lógicas. Deixe-me apresentar o programa da Listagem 4.7. A estrutura do programa em si é quase igual à lista anterior 4.6. Mas por favor tenha cuidado,

```
COR &HF8 , .
```

e ,

```
COPIAR .. ,
```

Existem duas partes. Agora você pode definir o valor Y da parte onde o texto será exibido para no máximo 31, e o texto será exibido em uma cor mais branca que o fundo. Por outro lado, mude a cor do texto para 3, Lógica. Se você definir a operação do menu como TAND, um texto enegrecido será exibido.

Quando você executa o programa , Primeiro, uma lista de arquivos no disco será exibida.

Especifique o arquivo de imagem que deseja usar. Em seguida, escreva a legenda que deseja exibir na tela e posicione-a.

Insira a localização em coordenadas. A legenda agora deve ser exibida na tela TELA 12. centro. Depois de ler o programa, verifiquei. Pode ser divertido distribuir muitos deles.

#### Listagem 4.7 (S12T.BAS)

100' TELA 12 Terror desligado.

110 ' por nao-i em 4 de novembro de 1988

120 TELA 0 • LARGURA 32: LIGUE PARA KANJIO

130 DEFINT AZ: ON STOP GOSUB260: STDP ON

140 ARQUIVOS: IMPRIMIR: INPUT "nome do arquivo " ; FF\$

150 ABRIR FF\$ PARA ENTRADA COMO # 1 : FECHAR #1 160

INPUT " string 170 INPUT ;SS\$

"X(O ... 240)";X 180 INPUT "Y(O ... 196)

";Y 190 TELA 12: SET PAGE 1,1: BLOAD

FF\$ ,S 200 SET PAGE 0,0: COLOR &HF8 ,0: CALL CLS 210

L=LEN(SS\$) Livro 8- 1

220 LOCATE O,O:PRINT SS\$

230 COPY (0,0)Y(L , 15) ,0 TD (X, Y) , 1, TdR

240 SET PAGE 1, 1 250

GOTO 250 260 '

\*\*Livro chamado por STOP \*\*\* 270 SET PAGE

O,O:COLOR 15,4,7

#### 4.6.9 Método YJK e registros VDP

Exibição em formato YJK usando programa em linguagem de máquina em vez de BASIC SCREEN 10 a 12. Deixe-me mostrar como fazer isso manipulando os registradores VDP. . controlar

Bit 3 "YJK" e bit 4 "YAE" do registro 25 selecionam a exibição da tela no formato RGB e a exibição no formato YJK pode ser selecionada.

Primeiro, registros diferentes do registro 25, Na linguagem BASIC, defina as configurações da mesma forma que para SCREEN 8. Selecione o método YJK especificando o modo de tela 10 a 12. Você tem que escolher. Porém, em função do VDP, a tela da TELA 8 tem formato RGB e formato YJK. É mais fácil entender se você pensar que é possível mudar para um método diferente. Bit 3 YJK é 0, Se o bit 4 YAE também for 0, a própria TELA 8 será definida. Então, Se você deixar esse registro como está e mudar o bit 3 YJK para 1, SCREEN 12 O mesmo formato de exibição YJK está definido.

Para exibir uma tela mista YJK e RGB na TELA 10 ou 11, 1 para YJK, Basta definir YAE como 1 também. Observe que se YJK for 0, a emenda A cor do alvo não é afetada pela paleta. Se YJK for 1, altere-o com a paleta.

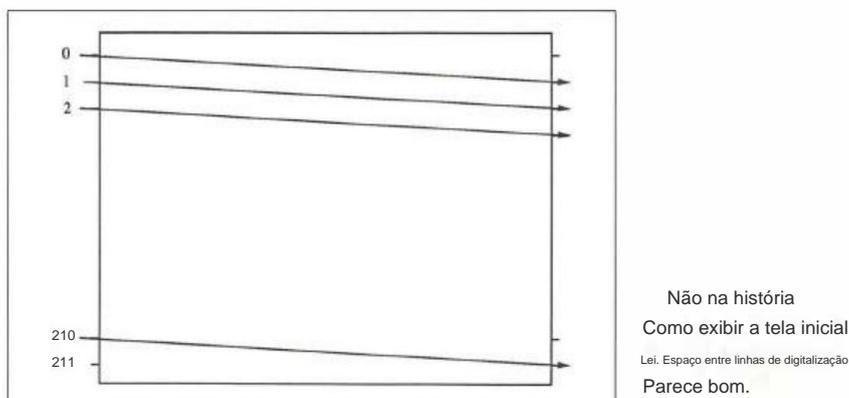
Eu posso queimar e queimar.

## 4.7 Estudando a interrupção da linha de varredura

### 4.7.1 Como funciona a exibição da tela do monitor?

A tela do MSX2 SCREENS é uma coleção de pontos (pixels) medindo 256 na horizontal x 212 na vertical. Você sabe que isso é expresso em . O mecanismo para realmente exibir isso no monitor é mostrado na figura. Como mostrado em 4.8. Cada pixel é varrido uma linha por vez, do canto superior esquerdo ao canto inferior direito da tela. Ao enviar dados, a tela é exibida. Neste momento, 256 Tsukuda estavam alinhados lado a lado. Uma coleção de pixels é chamada de "linha de varredura". São 256 pixels horizontalmente. 212 linhas verticais de varredura se unem para formar uma tela. Tornou-se um Shinodanmi.

Figura 4.8: Varrendo linhas em uma tela de TV



Esta linha de varredura é usada não apenas para telas de computador, mas também para transmissões de televisão comuns. É expresso como No sistema de transmissão de televisão "NTSC" do Japão e da América, 525 A tela é exibida usando as linhas de digitalização do livro. No entanto, o número realmente exibido na tela é aproximadamente 490. Resíduo As outras linhas de varredura contêm sinais para controlar a televisão chamados "sinais de sincronização" e sinais de multiplexação de caracteres. Contém sinais para transmissão. O número de imagens exibidas por segundo é 30. contém 525 linhas de digitalização, então  $525 \times 30 = 15.750$  linhas de digitalização Ele é exibido na tela por um segundo. Especificações para transmissão de TV e exibição na tela do MSX Os valores "frequência de varredura vertical 30 Hz, frequência de varredura horizontal 15,75 kHz" são baseados nesta ideia. Tem gosto.

Uma tela de 640 pontos horizontais x 400 pontos verticais, frequentemente vista em computadores de 16 bits. Aqueles com esta função usam uma frequência de varredura horizontal de 24kHz para exibir muitos pontos. É necessário um monitor dedicado e os computadores mais recentes possuem telas ainda mais detalhadas.

Kunari, Monitores de 31,5kHz ou 34kHz são usados.

"Multi-scanning" é um dispositivo que pode lidar com telas com várias frequências de varredura horizontal. Pode monitorar" . Suporta qualquer frequência de 15,75kHz a 34kHz?

Há uma variedade de modelos, incluindo aqueles que podem alternar entre 15,75kHz e 24kHz. criança Se você decidir comprar um novo monitor de outra pessoa, considere usar o computador ou computador existente.

Verifique cuidadosamente as especificações do computador que deseja usar e certifique-se de que ele suporta muitas frequências de varredura horizontal.

Escolha um monitor multi-scan

Para sua referência, gostaria de mencionar, a PAL nos países da Europa Ocidental, No Japão, são usados monitores com um sistema chamado "SECAM", e esses sistemas incluem sistemas NTSC e NTSC. O número de linhas de varredura é diferente. É por isso que os computadores fabricados na Europa e exportam MSX Mesmo quando conectado a um monitor japonês, a tela não pode ser exibida. Sony e Victor e outras empresas fabricam monitores que podem alternar entre os formatos NTSC, PAL e SECAM. No entanto, uma vez que é utilizado para fins especiais, como inspecionar computadores para exportação, Muito caro (Recentemente, a Panasonic lançou um Vitae Odeck que pode reproduzir vídeos de todo o mundo. Parece que a nova versão foi lançada).

Além disso, esta é uma digressão, mas, "Resolução horizontal" indica o desempenho de VTRs e discos de vídeo. "graus" não tem nada a ver com o número de linhas de varredura.

Uma medida da finura da tela, que corresponde ao número de pixels. A resolução vertical é igual ao número de linhas de varredura, O mesmo vale para qualquer videocassete. Mas também, "EDY" e "SVHS" apresentam menor consumo de água que os VTRs convencionais. A resolução horizontal melhorou.

## 4.7.2 Transmissão televisiva entrelaçada

O mecanismo de exibição da tela do computador era o mostrado na Figura 4.8, mas, Esta é a televisão quando se tratava de transmissão,.. A rigor, é um pouco diferente. Primeiro, dê uma olhada na Figura 4.9.

Eu quero que você olhe assim.

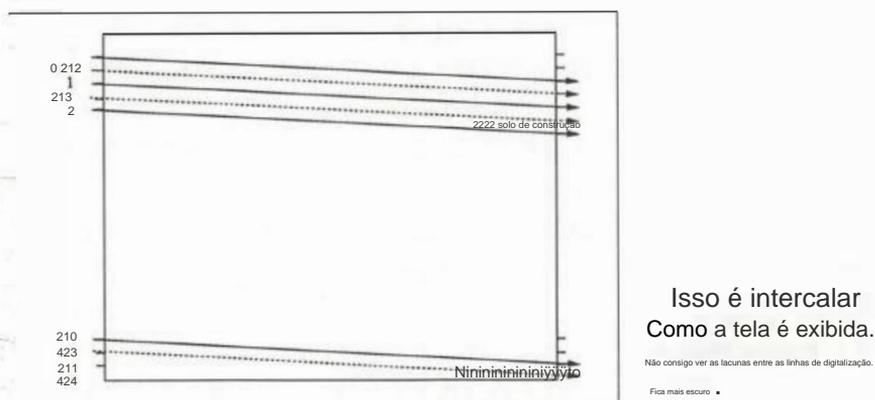
Esta é uma explicação de um método de exibição de tela chamado "entrelaçamento". televisão de verdade É por isso que este método é usado na transmissão. Primeiro, defina a linha de digitalização na parte superior da tela como o número 0. Se você exibir as linhas de varredura L, 2 e 3 da mesma maneira explicada na Figura 4.8, e quando chegar ao final da tela, você verá cada linha de varredura que acabou de exibir, para preencher o número. espaços em branco. As linhas de digitalização são exibidas em ordem, como os números 213 e 214.

A tradução literal da palavra "entrelaçar" é "tecer junto". Conforme explicado na Figura 4.8. Ao preencher as lacunas entre as linhas digitalizadas com outra linha de digitalização, a lacuna entre os pixels é reduzida. Estou tentando me livrar disso. A propósito, conforme mostrado na Figura 4.8, todos " É denominado "não entrelaçado".

Um método para exibir sequencialmente as linhas de varredura de

Então, por que esse método de entrelaçamento foi adotado para transmissão televisiva?

Figura 4.9: Isto é o que acontece no modo entrelaçado.



Destaca o ruído gerado por diversas causas e os distúrbios da tela causados por ele.

Para economizar dinheiro. Por exemplo, mesmo que as linhas de varredura O e L sejam perturbadas por ruído, alguns Após um período de tempo, se a 212ª linha de varredura exibida durante esse período for normal, a tela ficará distorcida.

Também é imperceptível. As ondas de rádio viajam pelo ar e estão sujeitas a ruídos e interferências de outras ondas de rádio.

Ruído gerado por produtos eletrônicos, ruído captado em tomadas elétricas, etc.

Este entrelaçamento é importante porque é facilmente afetado por vários tipos de ruído.

É um método eficaz.

Por outro lado, a desvantagem do método entrelaçado é que as posições das linhas de varredura adjacentes são ligeiramente diferentes umas das outras.

A tela pode parecer instável devido ao desalinhamento de telas com movimento, como aquelas vistas em transmissões de televisão.

Eu realmente não me importo com isso, mas quando congelo pequenos textos como na tela de um computador.

A cintilação é surpreendentemente perceptível ao exibir uma imagem fechada. Por esta razão, o MSX

A exibição de tela não entrelaçada é a norma para muitos computadores, incluindo

é usado

### 4.7.3 Tela entrelaçada no MSX2

Muitos computadores "geralmente" usam telas não entrelaçadas...

A razão pela qual escrevi isso condicionalmente é porque telas entrelaçadas também podem ser usadas no MSX2 e posteriores.

O primeiro motivo pelo qual adotamos o método entrelaçado, que apresenta muitas oscilações, foi Monitores de TV domésticos (ou seja, os chamados televisores domésticos que não são dedicados a computadores)

Isso porque ele exibe uma tela com 424 pontos verticais. MSX2 foi desenvolvido

Em 1985, os monitores multi-scan não eram comuns e a frequência de varredura horizontal

Um monitor de 24kHz também é caro, custando mais de 100.000 ienes. Era. Portanto, combinei-o com o MSX2.

O monitor a ser combinado é uma televisão doméstica geral ou um monitor com frequência de varredura horizontal de 15,75kHz. Monitores de baixa resolução tornaram-se populares. Por exemplo, com MSX2+,

LIGUE PARA KANJI2

Ao executar este comando e mudar a tela para o modo entrelaçado, 24 linhas verticais de Isso tornou possível exibir kanji. No entanto, como escrevi anteriormente, entrelace A tela pisca muito, o que pode facilmente cansar seus olhos, portanto, faça uma pausa de vez em quando. .

O segundo propósito é mais agressivo que o primeiro, e é conectar o MSX2 a uma tela de TV. para permitir "sobreposição" e "digitalização de vídeo" por é. A tela do MSX2 usa uma frequência de varredura horizontal de 15,75 kHz e um método entrelaçado. Sobrepor a tela do computador na tela da TV, ou sobrepor a tela do computador na tela da TV, ou Ideal para digitalização de vídeo, importação de imagens de uma câmera digital para um computador. Contra O FS-5500 da Panasonic, que possui todas essas funções integradas em um único visualizador, Produtos como o HC-95 da Victor, que pode ser usado conectando uma placa opcional, tornaram-se muito populares. Embora a máquina tenha sido lançada antes dos anos 2000, ela ainda é capaz de realizar tarefas como capturar e processar dados de imagens. Dizem que ele é muito ativo.

A terceira vantagem do método entrelaçado é que fica bem em fotografias. . Não-inter Nas capturas de tela de renda, lacunas nas linhas de digitalização são visíveis e, quando impressas, aparece um padrão listrado chamado "moiré". Fácil de aparecer. Entretanto, em uma tela entrelaçada, não há espaços entre as linhas de varredura, causando moiré. Não houve acordo.

Em máquinas MSX2 e posteriores, mude a tela para o modo entrelaçado da seguinte forma: Posso aprender e fazer coisas. . Entretanto, o modo Kanji só pode ser especificado no MSX2+ ou posterior.

Para máquinas MSX2                    TELA    ,, ,, 1

Para máquinas MSX2+ ou posteriores CALL KANJI3

#### 4.7.4 Explorando o princípio da interrupção da linha de varredura

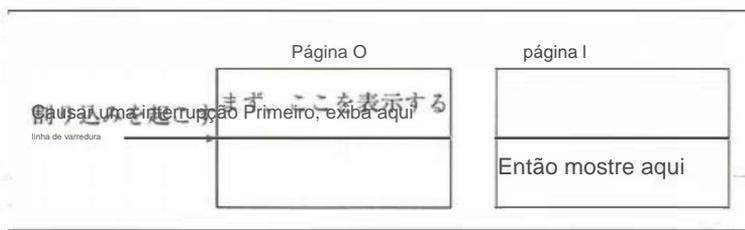
"Interrupção" refere-se à alteração do fluxo de um programa devido a condições excepcionais. Ta e Por exemplo, alterar o fluxo de um programa BASIC quando um botão do joystick é pressionado A instrução "ON STRIG GOSUB" também é um tipo de instrução que trata desta interrupção. Eu fiz.

O princípio de "interrupção da linha de varredura" é o mesmo e ocorre quando a linha de varredura com o número especificado termina de ser exibida. Quando isso ocorre, o processamento de interrupção é executado. No entanto, as interrupções da linha de varredura são processadas rapidamente. Por necessitar de atualização, a programação em BASIC não será suficiente. máquina É necessário o tratamento de interrupções por um programa de linguagem.

Agora, vamos explicar brevemente o princípio do processamento de entrada de dados. Primeiro, o VDP controla a exibição da tela. A "Light Pen" pode enviar um sinal de interrupção para a CPU dependendo de uma condição específica. "、"走査線"という3種類があるけど、MSX2ではライ As interrupções de Topen não são ignoradas.

Em primeiro lugar, "interrupção de retraço vertical" ocorre quando a exibição na parte inferior da tela termina e a exibição na próxima tela é iniciada. Esta é uma interrupção que ocorre durante a preparação e ocorre a cada 1/60 de segundo. Isso é A "interrupção do temporizador" do MSX é usada para ajustar o tempo de reprodução da música durante jogos, etc. Qual é a verdadeira natureza da "visão"? O problema desta vez é o que acontece quando a exibição de uma linha de digitalização específica é concluída. Esta é a interrupção da linha de varredura que ocorre. Essa interrupção também ocorre a cada 1/60 segundo.

Figura 4.10: Este é o princípio da interrupção da linha de varredura.



Ao combinar interrupção de retraço vertical e interrupção de linha de varredura, a borda superior da tela e qualquer É possível causar intrusão em dois locais ao longo da linha. Isto permite que a tela seja apagada verticalmente.

A parte superior da interrupção de linha até a interrupção de linha e da interrupção de linha até a interrupção de linha de retraço vertical.

Isso significa que pode ser dividido em duas partes na parte inferior.

MSX2 também permite que você tenha múltiplas telas chamadas "páginas". vídeo No MSX2 com 128 KB de RAM, 4 telas com SCREEN 5 ou 6; Isso significa que 7 e 8 podem ter duas páginas de tela. Alternando entre eles usando o comando BASIC "SET PAGE", você pode exibir múltiplas telas instantaneamente.

Esta função é usada para verificar interrupções de linha, que são usadas em softwares de jogos e outros aplicativos.

Crie um programa de processamento de interrupções em linguagem de máquina e troque de página usando interrupções de linha de varredura.

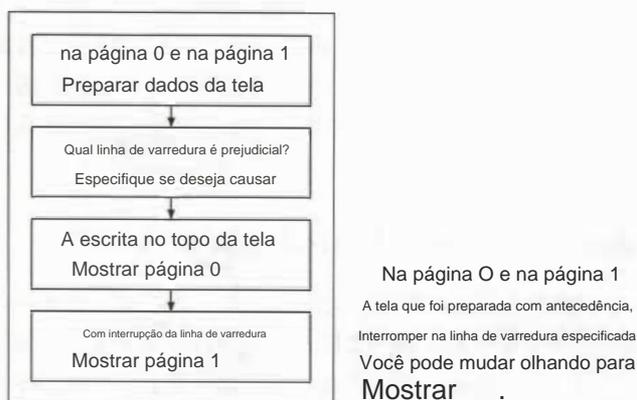
Isso permite exibir páginas separadas na parte superior e inferior da tela. Se você , Tela VDP também combinar isso com a função de rolagem, também será possível rolar apenas um lado da tela.

#### 4.7.5 Apresentando um exemplo de interrupção de linha de varredura

Para ser honesto, quando vi as especificações do VDP pela primeira vez, , "Interrupção da linha de digitalização" Eu me perguntei: "Qual é a utilidade do MSX2 Technical Hand?"

Embora a existência de uma função de interrupção de linha de varredura tenha sido introduzida no "Livro", etc.,

Figura 4.11: Procedimento de interrupção da linha de varredura



Não foram listados exemplos de programas específicos ou exemplos de aplicação. Na verdade, é um software de jogo. Aquele que foi aplicado ao mundo e deixou todos felizes foi o Pony Car desenvolvido pela Compile. Acho que foi depois do lançamento de um jogo chamado "Zanuck EX" lançado pela Nion.

Um dos recursos expandidos do MSX quando ele se tornou MSX2 foi a "rolagem vertical de hardware". Como . Este foi um recurso muito útil para fazer jogos de tiro, mas a tela inteira irá rolar até que haja uma pontuação, isso é essencial para o jogo.

Não foi possível fixar a exibição etc. na tela. No entanto, em "Zanac EX",

Enquanto a tela rola verticalmente em alta velocidade, a pontuação é exibida em uma posição fixa na parte superior da tela. Aí estava. Na época, no departamento editorial do M My Game, falava-se muito sobre que tipo de técnicas ele usava. Mas , Da cintilação no limite entre a parte da partitura e a parte da rolagem , Acabou sendo uma interrupção da linha de varredura.

fez. Jogos , Bem, quando você percebe isso, é como um "ovo de Colombo" e de agora em diante, a divisão da linha de varredura de tiro que aproveitam esse recurso começam a ser desenvolvidos um após o outro. linha de varredura Quando você usa a tecla de pausa para interromper o software do jogo que usa interrupçõesduas telas aparecem.

Apenas um deles pode ser exibido. Vamos realmente tentar! .

Além disso, o MSX2+ combina interrupções de linha de varredura e "rolagem horizontal de hardware". Além disso, também é possível rolar horizontalmente apenas uma parte da tela. Da Konami No lançado "F-1 Spirit 3D Speciale", apenas a tela do jogo pode ser rolada horizontalmente.

Ao mesmo tempo, uma interrupção na linha de varredura faz com que uma imagem da cabine de uma máquina FI apareça na parte inferior da tela. Parece estar exibindo uma imagem em branco.

## 4.7.6 Vamos para a parte prática!

Vamos apresentar um exemplo de programa que realmente usa interrupções de linha de varredura. Os detalhes serão explicados posteriormente. No entanto, este programa foi escrito em linguagem de máquina. No entanto, pode ser chamado a partir da linguagem BASIC. Como ele foi projetado para ser utilizável, você também poderá aplicá-lo aos seus próprios jogos. é. Também tornarei pública a lista de fontes, então se você conhece assembler, faça o seu melhor. Gostaria que você analisasse.

### 4.7.7 Registros VDP usados para interrupções de linha de varredura

As etapas para causar uma interrupção na linha de varredura são as seguintes: Primeiro registro de controle  
Escreva o número da linha de varredura na qual você deseja causar interferência ou prejudicial em 19 e registre o controle em 0.  
Mude o bit 4 para 1. Suru e , Quando a linha de varredura especificada terminar de ser exibida, o VDP interrompe a CPU. Além disso, quando ocorre uma interrupção, o estator  
Se o bit O do registrador de thread I for 1, a causa da interrupção é uma interrupção da linha de varredura.  
Eu entendo isso. As Figuras 4.12 e 4.13 resumem essas questões.

Figura 4.12: Registro VDP gerando interrupção de scanline

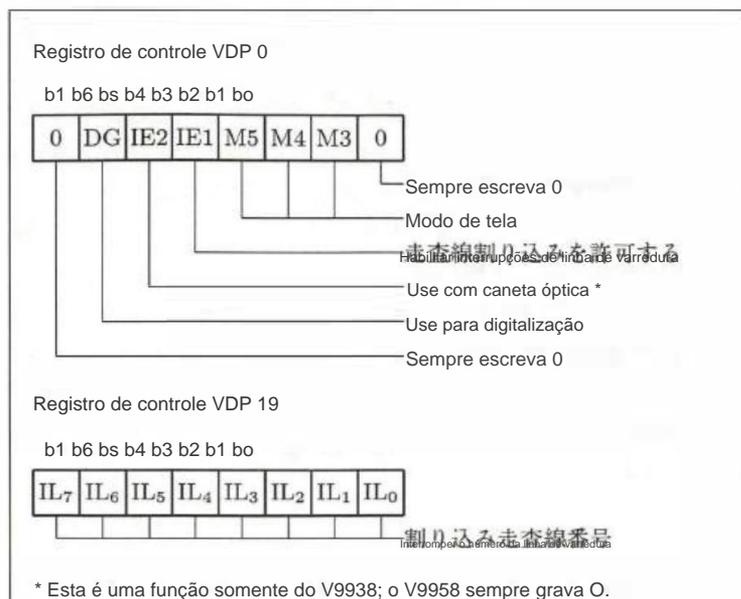
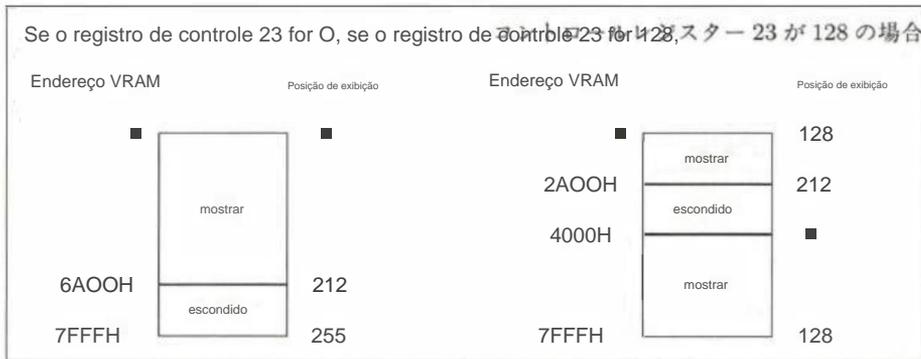




Figura 4.15: Mecanismo de rolagem vertical de hardware



Escrever permite que você troque de página.

Além disso, se você usar o registro de controle 23, , Execute a rolagem vertical do hardware

A . Especificamente, conforme mostrado na Figura 4.15, a tela muda dependendo do valor definido no registro.

correspondência entre o espaço de exibição e o endereço da RAM de vídeo muda e a tela rola verticalmente.

É assim que funciona. No entanto, se você usar essa rolagem vertical de hardware, poderá ter problemas.

Os números das linhas de varredura também mudarão, então faça correções, como adicionar a quantidade de rolagem vertical aos números das linhas de varredura.

Esta correção está sendo . No programa de exemplo , Sub-rotina começando em “ON\_VSYNC”

implementada nos casos em que uma correção é necessária.

#### 4.7.8 Método de montagem e operação da peça BASIC

O programa de exemplo para causar uma interrupção na linha de varredura é escrito em assembler. É composto por peças e peças criadas em linguagem BASIC. Princípio de funcionamento da peça montadora O processo será explicado a seguir, mas primeiro começaremos com o método de montagem e o princípio de funcionamento da peça BASIC. O programa .

(lista de fontes) do assembler introdutório é mostrado na Listagem 4.9 na página seguinte. Eu postei. Estou tentando inserir isso em um editor de tela MSX-DOS e salvá-lo com o nome de arquivo "ON-SCAN.Z80", mas estou tentando . Então entre em "MSX DOS TOOLS" salvá-lo com o nome de arquivo "M80.COM" e "180.COM" . Usando o programa, siga estas etapas: Monte e vincule. Se um arquivo de linguagem de máquina chamado "ONSCAN.BIN" for criado, Está feito. Observe que a peça "DEL ONSCAN.BIN" será alterada durante a remontagem. Também para excluir arquivos antigos. Você não precisa disso na primeira vez.

```
M80 , =ONSCAN.Z80/R/Z
180 ONSCAN , ONSCAN/N/E
DEL ONSCAN . LIXO
REN ONSCAN.COM ONSCAN . LIXO
```

A seguir, vamos controlar o arquivo de linguagem de máquina. , Programa BASIC (lista) (Ver Seção 4.8).

Primeiro, as linhas 190 a 200 são para inicializar a página 0 da RAM de vídeo. de. Se quiser rolar a tela verticalmente, basta inicializar a página inteira O. Porém, o comando "CLS" só pode inicializar a parte que está exibida na tela. Portanto, use o comando "COPY" para alterar o conteúdo de (0,0) (255.127) na tela para (0,128)-(255.255). Isso significa que ele está copiando e inicializando toda a página O. Também 210 linhas, 240 linhas, 250 linhas são as partes que exibem os padrões de teste.

BÁSICO. Do Lokallam à linguagem de máquina. Lok. Para ligar para RAM, use "Número de construção USR" Faça uso disso. A linha 260 especifica a linha de varredura que causará uma interrupção e coloca a página 0 no topo da tela. Esta é a parte que exhibe a página 1 na parte inferior da tela.

```
USR (256 + número da linha de varredura)
```

Se você alterar o valor do número da linha de varredura de acordo com o formato, o local onde ocorre a interrupção mudará. Além .

disso , a página 0 exibida na parte superior da tela é rolada verticalmente. 290 linhas.

```
USR (512 + número da linha de varredura)
```

Especifique o valor no formato . Além disso, não usei desta vez. ,

```
USR (768 + número da linha de varredura)
```

Se você especificar , poderá rolar verticalmente a página 1 exibida na parte inferior da tela.

pode . Para cancelar a interrupção da linha de varredura, use

```
USR(O)
```

Tudo que você precisa fazer é executar.

Observe que neste programa os parâmetros da função USR são alterados para simplificar a parte do assembler.

O medidor é limitado a números inteiros. . é por isso ,

```
USR(868,0)
```

Ou algo assim

```
USR(100+A !)
```

Parâmetros de números reais, como , não podem ser usados.

## Listagem 4.8 (ONSCAN.BAS)

```
100,
110' onscan1 bas: interrupção de teste no mar e
120' por nao-i em 24 de setembro de 1989 130, 140
CLEAR
300 150 DEFINT AZ , &:HAFF
160 OPEN "GRP:" FOR
OUTPUT AS #1 170 BLOAD "onscan. bin"

180 TELA 5
190 DEFINIR PÁGINA 0,0: COR 1 5 , 6,1: CLS
200 CÓPIA (0,0) - (255 128) , 127) PARA (0 ,
210 CÍRCULO ( 128 , 106) , 80: PINTURA (128, 106)
220 CONJUNTO PÁGINA 1,1: COR 15 , 1,1: CLS
230 DEFUSRO=&:HBOOO
240 PSET (8, 192) ,0,PRÉ-DEFINIDO
250 IMPRESSÃO #1 , "Esta correção não irá rolar"
260 JK=USR (256+100)
270 PARA J=1 A 5
280 PARA I=O A 255
290 JK=USR (512 + engenharia)
300 PRÓXIMO I
310 PRÓXIMO J
320 JK=USR (O): COR 15,4,4•TELA 0
330 FIM
```

#### 4.7.9 Montador Parte do princípio de funcionamento

Agora vamos passar para o programa principal da Listagem 4.9 que realmente trata as interrupções da linha de varredura. Explique o princípio de funcionamento do grama.

Primeiro, as três linhas que começam com "ASEG" são usadas para criar um subsistema BASIC usando M80.COM ou 180.COM. Instruções para criar objetos especiais, como rotinas. M80.COM e 180.COM são Geralmente usado em pares, é possível converter programas escritos em assembler para arquivos em linguagem de máquina. É algo que cria algo. Porém, a extensão do arquivo criado neste momento é "COM". MSX se torna um arquivo em linguagem de máquina que pode ser executado em DOS. Se você deseja criar um arquivo que possa ser manipulado pelo BASIC, você precisará de instruções deste ASEG. É por isso.

Além disso, um programa em linguagem de máquina em um formato que pode ser BLOADED em BASIC possui 7 bytes no início. Há um cabeçalho de ,

```
FEH
Carregar endereço inicial
Carregar endereço final
endereço de início de execução
```

Os seguintes dados são gravados respectivamente. Estes são especificados pela ASEG.

As próximas quatro linhas.

Bem, então , A parte após "AD-LOAD:" torna-se o corpo principal do programa. linha primeiro Este é o processamento dos parâmetros da função USR. Se o parâmetro for um número inteiro, o nível A é O conteúdo do registro é 2, então estou verificando. Além disso, o nome da entrada O valor dos dados é registrado nos endereços HL+2 e HL+3. Neste momento, acima do parâmetro Se o byte ativo for 0, ele será usado para limpeza, se for 1, será usado para definir interrupções de linha de varredura e, se for 2, será usado para definir a página O verticalmente. Rolar, 3 significa rolagem vertical da página 1.

Quando realmente ocorre uma interrupção, o endereço FD9AH (parte HOOKDT) é chamado. Isso vai acontecer. Nas próximas 5 notas,

```
Horário de início 30H
Número do slot do banco de dados
Endereço DW
RET
```

Ao escrever isso, você pode chamar o programa especificado quando ocorrer uma interrupção. pode ser feito. Um lugar como este, que é chamado sob certas condições, é chamado de "gancho". aqui Agora estamos nos preparando para que "ON\_H.KEYI:" seja chamado.

Do endereço FD9FH, interrupção do temporizador (ON\_H.TIMI:), Interrupção de retraço vertical A parte que chama. Aqui mostraremos o registro de status do VDP quando ocorrer a interrupção.

Não reescreva o registro AF porque o valor 0 está armazenado no registro A.

Se você realmente precisar reescrever o gancho, escreva-o como no programa de exemplo.

Salve o valor original de e, em seguida, pule para o gancho salvo quando o processamento da interrupção for concluído.

Vamos tentar forçar isso.

Um programa que é chamado quando ocorre uma interrupção e opera o VDP, "ON\_VSYNC:" e , "ON\_SCAN: Eu os resumi em letras sub-j começando em y' Reescreva aqui.

Nesse caso, poderíamos usar interrupções de linha de varredura para outros fins. .

A subrotina "\_VDPSTA" lê o registro de status do VDP. sub azul

O comando "WRTVDP" grava o valor especificado no registro de controle VDP.

e salve os valores em seus respectivos locais de armazenamento (ver Tabela 4.9). Eu escrevi isso antes também.

Da mesma forma, ROMs MSX2 e 2+ possuem um BIOS com as mesmas funções dessas sub-rotinas.

Como existe uma sub-rotina, normalmente você pode usar o BIOS sem criar uma sub-rotina sozinho. Mas isso

Como o BIOS está localizado em uma sub-ROM ou chama a sub-ROM durante o processamento,

A desvantagem é que leva algum tempo. Portanto, interromper o processamento como esse exige muito esforço.

Não usei o BIOS propositalmente porque não combinava bem.

Isto não tem nada a ver com as sub-rotinas da linguagem de máquina do BASIC. , Programa DOS

Na RAM, existe uma função de processamento de interrupções. o programa deve ser colocado em um endereço maior que 4000H. criança

Isto ocorre porque ocorrem problemas com MSXs com configurações de slots específicas abaixo deste valor.

## Listagem 4.9 (ONSCAN.Z80)

```

;
;
;      onscan.z80: programa de teste para interrupção na varredura por
;      nao-i em 26 de setembro de 1989 .
;
;
;      chamada como função USR do BASIC
;      USR(&H00xx) restaura registros e ganchos
;      USR(&H01xx) definir linha de interrupção
;
;      USR(&H02xx) define linha de deslocamento de exibição da página 0
;      USR(&H03xx) define linha de deslocamento de exibição da página 1
;
;
;      .Z80
INICIAR EQU USE      OBOOOOH      ; endereço para carregar e executar 0
SUB      _      EQ
USE_WRTVDP      EQ      ■
;
;      SE      USAR_WRTVDP
WRTVDP      EQ      0047H
;
;      TERMINAR SE
EXTROM      EQ      015FH
VDPSTA      EQ      0131H
SETPAG      EQ      013DH
;
;
RAMAD2      EQ      OF343H
RAMAD3      EQ      OF344H
RGOSAV      EQ      OF3DFH
DPPAGE      EQ      OFAFSH
ACPAGE      EQ      OFAF6H
H.KEY1 H.    EQ      OFD9AH
TIMI        EQ      OFD9FH
RG8SAV      EQ      OFFE7H
;
;
;      ASEG
;      ORGANIZAÇÃO      100H      ; para criar um arquivo .COM
;      .FASE      INÍCIO-7
;
;
;      OFEH      ; cabeçalho para BLOAD ;
;      CARREGAMENTO DE ANUNCIO      endereço para carregar ;
;      AD_NEXT-1
;      NÃO_FAÇA NADA
;
;
;      CARREGAMENTO DE ANUNCIO
EMPURRAR      AF
EMPURRAR      H. L.
EMPURRAR      D.E.
EMPURRAR      2
CP      AC
JR      , RESET_SCAN      ; o parâmetro não é inteiro
INC.      H. L.
INC.      H. L.
L. D.      E, (HL)
INC.      H. L.
L. D.      D, (HL)      ;DE = pmuametro de USR()

```

```

L. D. ANUNCIO
OU LIM
JR Z , RESET_SCAN
DEZEMBRO LIM
JR Z , SET_SCAN
DEZEMBRO LIM
JR Z , SET_DO
DEZEMBRO LIM
JR Z,SET_D1
JR REINICIAR VARREDURA
;
DEFINIR VERIFICAÇÃO:
L. D. A,E
L. D. (ILSAV),A
CHAMAR DEFINIR ILREG ; definir linha de interrupção
L. D. A, (enganchado)
OU LIM
JR RET_BASIC ; o gancho já está definido
;
L. D. HL, H. KEYI DE
L. D. HOOKSA
L. D. a.C. , 10
LDIR ; salvar ganchos
L. D. H. L. , HOOKDT
L. D. DE,H.KEYI AC 10
L. D.
D.I.
LDIR ; definir ganchos
L. D. A, (RAMAD2)
L. D. (H. KEYI+1) ,A (H.
L. D. TIMI+1) ,A A, (RGOSAV)
L. D.
OU 00010000B
L. D. B,A
L. D. c,o
CHAMAR WRTVDP ; interromper em
L. D. A, 1
L. D. (HOOKED) ,A
JR RET_BASIC
;
RESET_SCAN:
CHAMAR RESET_SCAN_SUB
JR RET_BÁS IC
;
SET_DO: ; definir o deslocamento de exibição da página 0
L. D. A,E
L. D. (DOVAL) ,A
JR RET_BÁS IC
DEFINIR Di: ; definir o conjunto de exibição da página 1
L. D. A,E
L. D. (D1VAL),A
;
RET_BASIC:
E.I.
POP a.C.

```

```

        POP        D.E.
        POP        H.L.
        POP        AF
NÃO_FAÇA_NADA:
        RET
;
RESET_SCAN_SUB:
        D.I.
        L.D.      A, (RGOSAV) 1
        E        11 011118
        L.D.      B,A
        L.D.      C,0
        CHAMAR   WRTVDP          ; interromper off ; escrever
        L.D.      a.C. , ----    0 no reg#23 ;
        LIGUE PARA WRTVDP
        XOR      UM
        L.D.      (DPPAGE) ,A
        L.D.      a.C. , 1F02H    ; escreva 1FH em reg#2 ;
        CHAMAR   WRTVDP
        L.D.      A, (enganchado)
        OU      UM
        RET      Z
        L.D.      H.L. , HOOKSA
        L.D.      DE, H.KEYI BC
        L.D.      , 10
        LDIR     ; restaurar ganchos
        XOR      UM
        L.D.      (Enganchado), Um
        RET
;
DEFINIR_ILREG:
        L.D.      A, (ILSAV)
        L.D.      HL, DOVAL
        ADICIONAR A, (HL)          ; linha de interrupção = (ILSAV) + (DOVAL)
        L.D.      B,A
        L.D.      C,19
        JP       WRTVDP          ; escreva a linha de interrupção # no reg#19
;
EM_H.KEYI:
        L.D.      UM,1
        CHAMAR   _VDPSTA1        ; leia o registro de status nº 1
        E
        CHAMAR   Nova Zelândia, ON_SCAN
        JR       HOOKSA
;
EM_H.TIMI:
        L.D.      ; chamado de H.
        PUSH     AF
        CALL    ON_VSYNC
        POP      AF              ; não altere AF em H. TIMI
        EI
        JR      GANCHOSA+5
;
GANCHO:
        RST     30H .
        D.B.

```





## 4.7.10 Rotina em linguagem de máquina para interrupções de linha de varredura

Finalmente, um arquivo para a linha de varredura é interrompido. É problemático digitar a lista de fontes do programa.

Crie automaticamente arquivos de linguagem de máquina para pessoas que não possuem montador ou que não possuem montador.

Poste o programa BASIC. Digite o programa abaixo e execute-o.

Ele cria dinamicamente um arquivo chamado "ONSCAN.BIN".

```
10 LIMPAR 100, &HCFFF: DIMD(15)
```

```
20 PRINT "Fazendo onscan.bin" : AD=&HBOOO: C=O:L=O
```

```
30 PARA I=OT015: LEIA A$: SE A$="y" GOT0100 40
```

```
A=VAL("&h"+A$): C=(C+A) AND 255: D(I)=(D(I)+A) E 255 45 POKE AD,A: AD=AD+1: PRÓXIMOS
```

```
50 ADAS: E=VAL("&h"+D(I)): L=L+1
```

```
60 ADAS: E=VAL("&h"+L): L=L+1
```

```
55 IF C<>A THEN PRINT "Erro na linha" ; 990+10*L: END 60 GOTO 30 100'
```

```
60 ADAS: E=VAL("&h"+L): L=L+1
```

QI

```

ng
an
"
5, 53 5E , B7 ,
D, 3D 1E , D9 ,
1, 20 9A , B0 ,
0, 11 01 , ED ,
3, A0 DF , 47 ,
D, CC OF , 18 ,
2, 52 FE , F1 ,
A, OE F4 , 00 ,
0, 02 , F4 , B0 ,
1, 01 EE , CC ,
A, 86 13 , 3E ,
A, B1 F5 , F1 ,
0, F7 BC , 00 ,
0, 00 2 E1 , 3E ,
7 D, E7 3F 424 FC 50 78 , 00 ,
0, 79 F1 , 30 ,
F, AF 72 , 07 ,
D, ED E1 , F5 ,
2, D7 OE , B0 ,
0, 01 CE , D8 ,
E,

```

第 5 章 MSX MUSIC



Este capítulo cobre MSX Mariki, Jin da edição de julho de 1990 até a edição de outubro de 1990.

Esta é uma versão reeditada do artigo "MSX2 + Technical Expedition".

## 5.1 Que tipo de fonte de som FM é?

Uma fonte de som FM cujas especificações foram determinadas sob o nome MSX-MUSIC. Efeitos sonoros de jogo poderosos  
Eu sei que isso pode te ajudar com algumas coisas, mas como funciona? Esta página

Vamos dar uma olhada mais de perto no mistério.

### 5.1.1 História dos instrumentos musicais eletrônicos que levam às fontes sonoras FM

Antes de explicar as fontes de som FM, vamos relembrar a história  
dos instrumentos musicais eletrônicos .

Combinando filtros, ele criou um instrumento chamado "sintetizador estilo Moog". 1968  
O primeiro disco com isso foi lançado em 2007 e, na década de 1970, muitos músicos já o usavam.  
começou a usar sintetizadores. No entanto, este sintetizador tornou-se recentemente popular.  
Ao contrário dos "sintetizadores digitais" usados no  
Algo feito combinando estradas. "Sintetizador analógico" versus digital  
também é chamado.

No entanto, este sintetizador analógico tinha algumas desvantagens. Isso é calor.

Eles são sensíveis a mudanças de temperatura, caros e suscetíveis a ruídos. O autor também viveu no outono da década de 1970.  
Comprei um IC na Habara e construí meu próprio sintetizador, mas fiquei impressionado com a dificuldade de fazer ajustes.  
restos.

Ben, então , A música eletrônica usando circuitos digitais foi desenvolvida para superar essas desvantagens.  
navio. A fonte de som digital mais simples é um "gerador de som programável".

É abreviado como "PSG". Isto converte as saídas de cerca de quatro osciladores digitais em um analisador digital.

Um conversor de log (D/A) que converte o sinal em um sinal de áudio e o envia. É barato e fácil de usar, ,  
por isso está integrado em muitos computadores pessoais, como o MSX.

Uma maneira de criar sons mais complexos do que o PSG é usar uma "fonte de som amostrada". esse  
recebe o som de outros instrumentos com um microfone e o converte em um conversor A/D (analógico/digital).  
converte-o em um sinal digital, armazena-o na memória e converte-o novamente em um sinal analógico com um conversor D/A.  
e jogue. Um instrumento que aplica isso é chamado de "sintetizador de amostragem".

Sem chance. Existe um alto grau de liberdade na criação de sons, mas requer uma grande quantidade de memória, etc.

A desvantagem é que as roupas são caras.

Agora, o preço barato do PSG e , Está atraindo a atenção como uma fonte sonora que possui a flexibilidade de uma fonte sonora amostrada.

O que é usado é uma fonte de som FM. FM é o mesmo que transmissão FM e sinais FM de modem.

Tabela 5.1: Comparação do desempenho de instrumentos musicais eletrônicos

	アナログシンセ	PSG	サンプリングシンセ	FM音源
Preço da quantidade de dados	複雑	LSI LSI+	memória de grande capacidade	LSI
tom de	Estável, vulnerável a mudanças de temperatura,			estável
	versátil, pobre, versátil, pequeno, extremamente			Variedade
estabilidade		barato .		pequeno
complexo de hardware	高価			meda

“Modulação”. Conforme mostrado na Figura 5.1, a saída de um oscilador digital é modulada por outro oscilador digital. Modula a frequência de um oscilador digital para criar sons mais complexos que o PSG. 1 digital de Tsukuda Este oscilador também é chamado de “operator” e, conforme mostrado na Figura 5.1, um som FM que inclui dois osciladores A fonte é chamada de “fonte de som FM com conjunto de 2 operadores”. A propósito, “MSX-MUSIC” está oficialmente O nome é “OPLL YM2413” e possui uma fonte de som FM integrada com 9 conjuntos de 2 operadores. É o LSI.

Figura 5.1: Explorando a estrutura de quatro tipos de instrumentos musicais eletrônicos

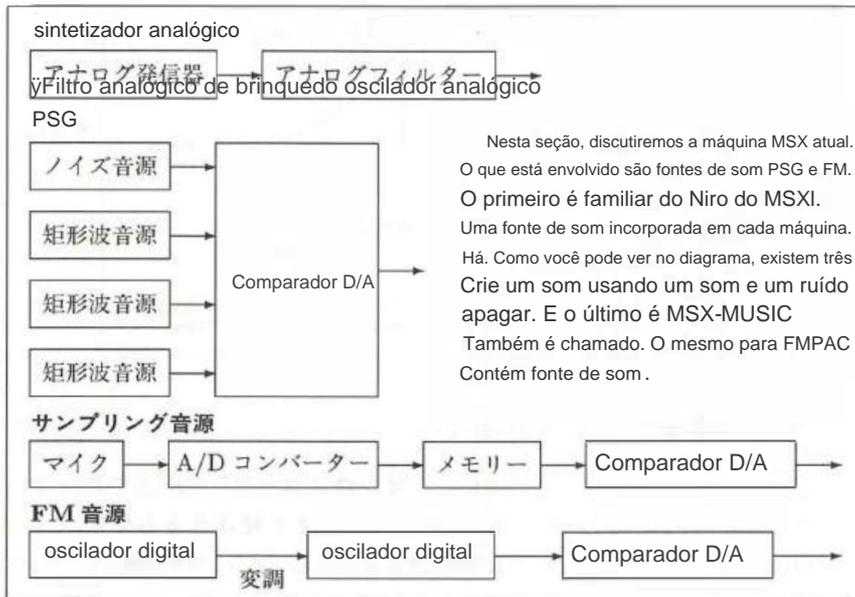
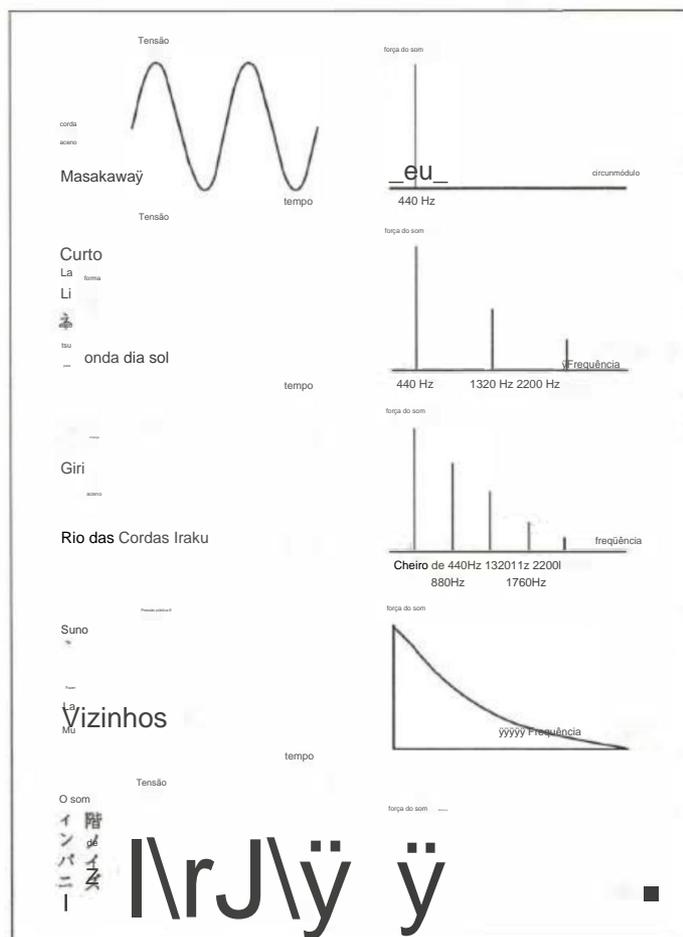


Figura 5.2: Analisando os sons básicos



### 5.1.2 sons de instrumentos musicais analisar o Vamos tentar

Para “ver” o som, basta olhar a voltagem do sinal sonoro na tela de um “osciloscópio”.

Além disso, existe um dispositivo denominado “analisador de espectro” (comumente abreviado como analisador de espectro).

Ao decompor o som em seus componentes de frequência, podemos compreender as características do som. O lado esquerdo da Figura 5.2 mostra o osciloscópio.

Este é um diagrama exagerado das características da forma de onda do som de um instrumento visto através do osciloscópio, com o analisador de espectro à direita.

ft] componentes de frequência.

O som mais básico é chamado de “onda senoidal”, cuja forma de onda é expressa pela função trigonométrica seno.

Rumono. Ele contém apenas uma frequência de som. A seguir, a forma de onda é quadrada.

onda”, 440 Hz 1320 Hz , 2200Hz...A frequência fundamental e seus múltiplos ímpares

Também inclui sons numéricos. Num instrumento musical real, o som de um clarinete está próximo desta onda quadrada. Próximo O que é fundamental para isso é a “onda dente de serra”. Isto inclui sons na frequência fundamental e seus múltiplos. A qualidade do som é semelhante à dos instrumentos de corda. Os sintetizadores analógicos são como serras. As ondas são processadas para criar sons semelhantes aos instrumentos musicais reais. Agora, instrumentos de percussão, especialmente caixa. O som da cachaça é muito diferente de outros instrumentos musicais. Não há regularidade, mas sim É semelhante ao “ruído”. Quando visualizado com um analisador de espectro, uma ampla gama de Contém o som da frequência .

O som de um tímpano está em algum lugar entre um instrumento de cordas e um instrumento de percussão, com uma frequência próxima à frequência fundamental. Uma combinação de sons com frequências diferentes. Também é chamado de “ruído de escala”. Além disso, esse ruído de escala Ao processar o som, sons como vento, ondas e assobios podem ser sintetizados. Os sons . Nas aulas de educação física, etc., o professor de flautas e instrumentos de sopro tocados por amadores também são ruídos de escala.

Em vez do som desses instrumentos, ou seja, uma forma de onda como a mostrada na Figura 5.2, a fonte sonora FM utiliza modulação. Ele distorce uma onda senoidal para criar uma forma de onda complexa contendo a frequência fundamental e frequências que são múltiplos da frequência fundamental. Esta é uma técnica difícil, e a única maneira de imitar o som de um instrumento é ajustando os valores por tentativa e erro. Lá Um programa para sintetizar vários sons instrumentais em uma fonte sonora FM. Possui um programa integrado, que Tornou-se comum escolher entre uma variedade de sons e usá-los. .

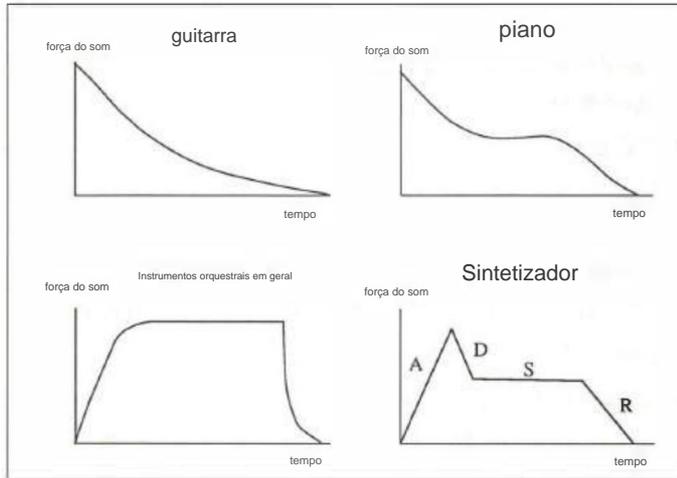
Além da forma de onda básica, os elementos que caracterizam o som incluem mudanças na intensidade do som. Esta “força” também se refere a “envelope” no sentido de “envolver” a forma de onda básica. Também existe algo chamado “pu”.

Por exemplo, quando você toca um violão ou um instrumento de percussão, um som forte é emitido naquele momento e continua por um tempo. O som fica mais fraco. Quando você pressiona uma tecla de um piano, primeiro é produzido um som alto, depois fica mais baixo. O som enfraquece gradualmente e continua em volume constante enquanto a tecla for pressionada. parar Quando você solta uma tecla, o som fica mais fraco. O som foi suave, seguido por um som com o mesmo volume, e o final foi quase igual à subida. (Ver Figura 5.3).

Por outro lado, com sintetizadores analógicos e fontes de som FM, a velocidade de subida (A: Tack), decadência imediatamente após (D: Decay), força de sustentação (S: Sustain), velocidade de desaparecimento (R: Solte) e sintetize o envelope. O dispositivo para isso é “ADSR” Também é chamado.

Era uma vez, um grupo de rock chamado Pinroid tinha uma música chamada “Blow the Wind, Call the Storm”. No interior, ao girar a fita na qual o som do simul foi gravado, o som fica gradualmente mais alto e, de repente, fica mais alto. Criei um efeito sonoro que me parou. No entanto, se você usar um sintetizador, poderá sintetizar , Dia esses sons simplesmente desacelerando o ataque, apertando a tonalidade e ajustando a sustentação para 0. Sintetizar um envelope é mais fácil do que sintetizar uma forma de onda de fonte sonora FM, então você mesmo pode sintonizar o ADSR. Seria divertido organizá-los e criar efeitos sonoros.

Figura 5.3: Envelopes de instrumento e sintetizador



### 5.1.3 Intervalos não são necessariamente temperamentos iguais

De repente, vamos falar um pouco sobre música clássica, primeiro, escala e frequência

Organizando a partir da relação de correspondência de , conforme mostra a Tabela 5.2.

Tabela 5.2: Relação entre escala e frequência

UM	440,0Hz
UM#	466,2Hz
B	493,9Hz
c	523,3Hz
C#	554,4Hz
D	587,3 Hz
D#	622,3 Hz
E	659,3Hz
F	698,5Hz
Fá#	740,0Hz
G	784,0Hz
So#	830,6Hz
um	880,0Hz

Entre estes, a frequência do som A é 440Hz, que é uma oitava. A frequência da nota a acima é 880 Hz, com uma oitava de diferença. Isso significa que a frequência dobrou. Além disso, a frequência dos sons separados por um semitom é a proporção dos números de onda é de aproximadamente 1,0595 vezes, que é a diferença entre sons separados por 12 semitons (1 oitava). A proporção de frequências é  $1,059512$  ou  $2^{1/12}$ .

Desta forma, todas as frequências da escala são expressas como uma progressão geométrica. Esta regra é chamada de "temperamento igual perfeito". Este temperamento é um clássico desde a era barroca. Parece ter sido estabelecido durante a era da seita. Quando eu era estudante do ensino médio, disseram-me que "Bach criou a Igualdade de Temperatura Bem Temperada", mas pesquisas recentes mostram que a teoria mais provável é que o temperamento igual foi criado depois de Bach. atual A maior parte da música moderna é executada com base no temperamento igual de D.

Vamos , Antes da criação do temperamento perfeitamente igual, havia um temperamento em que a proporção de frequências era um número racional. foi usado. Isso difere do temperamento igual porque a proporção de frequência dos semitons difere dependendo da localização. Por exemplo, a característica é que a proporção entre C e C# é diferente da proporção entre B e C. De temperamento igual Como as regras são complexas, foram estabelecidos vários temperamentos, como mostra a Tabela 5.3. apenas Com temperamentos diferentes do temperamento igual, os acordes soam lindos, mas o problema é que fica difícil transpor a música.

Tabela 5.3: Lista de temperamentos que podem ser configurados no MSX-Music

Temperamento definido por Atsushi	番号	Temperamento a ser definido
10 Pita Goffs	11	Apenas entonação cis maior (b menor)
1 tom médio	12	apenas entonação d jl jar (h menor)
2 Welck Meister	13	Apenas entonação em es maior (dó menor)
3 Werckmeister (Modificação 1)	14	Apenas entonação em mi maior (cis menor)
4 Werkmeister (Modificação 2)	15	Apenas entonação em fá maior (ré 7 em uma)
5 Kirnberger	16	apenas entonação fis maior (es menor)
6 Kirnberger	18	(fixo) 17 Apenas Temperamento Sol Maior (Mi Menor)
7 Vualoi Young	19	apenas entonação sol maior (fá menor)
8 Fumo	9	Entonação Lá Maior (S Branco? Menor)
Temperamento igual perfeito (configuração padrão)		20 Apenas entonação Si maior (Sol menor)
10	Apenas entonação Dó maior (Lá menor)	21 Apenas entonação H maior (GIS menor)

Existem também.

Para instrumentos como o violino, que podem mudar de frequência continuamente, que tipo de temperamento pode ser usado? Nós também podemos lidar com isso. Mas para mudar o temperamento de um piano, é preciso reafinar todas as cordas. Se necessário, Porém, em termos práticos, não é possível mudar o temperamento. Entretanto, o som FM instalado no MSX você pode escolher o temperamento conforme mostrado na Tabela 5.3. Com isso, apenas entonação de guitarra Isso significa que é possível criar sons de instrumentos musicais que não são facilmente possíveis. Utilizando esse recurso, FM Manipulando os registros da fonte sonora e ajustando o temperamento, você pode ajustar com precisão a música gagaku, a música Ryukyu, etc.

Não sei se consigo jogar .

Observe que a frequência de uma fonte sonora FM é tão precisa que o temperamento se torna um problema, mas a frequência de um PSG é Nem tanto. Portanto, é perigoso usar o PSG como padrão para afinação de instrumentos ou prática vocal. Dito isto, tenho um péssimo senso de afinação, então não entendo realmente a diferença de temperamento. Fonte de som FM Para poder usá-lo, você precisa de conhecimento de matemática, eletricidade e teoria musical, bem como um senso preciso de altura e sentido. No entanto, essas pessoas são raras. Compositores e timbres também são usados para criar músicas de jogos. Será necessário dividir funções para que designers e programadores trabalhem juntos.

#### 5.1.4 Analisar MÚSICA MSX

MSX-MUSIC vem com 63 timbres predefinidos. 15 destes Os tipos são os tons incorporados no LSI da fonte de som FM, e os 48 tipos restantes são gravados na ROM. É o tom. Os dados de tom gravados na ROM são

CÓPIA DE VOZ DE CHAMADA

Pode ser chamado com o comando . A Listagem 5.1 mostra esses dados.

É um programa para mostrar a você. Observe que se você especificar o número do tom incorporado na fonte de som FM,

ra acorda , Para o programa da Listagem 5.1,

Voz Não. . \*não possui dados.

Uma mensagem será exibida agora.

## Listagem 5.1 (READFM.BAS)

```

100 ' leia VOICE DATA do MSX-MUSIC 110 ' por nao-i em
20 de abril de 1990 120 '

130 MÚSICA DE CHAMADA: DEFINT AZ 140 DIM VI
(15) ,DC (31) , VO (3)
150 PRINT "Número da voz (0, . , 63; 64 para todos; 65 para final)";
160 ENTRE-ME
170 SE 0 <= ME E ME <= 63 ENTÃO VN=ME: GOSUB 200: GOTO 150 180 SE ME = 64 ENTÃO PARA VN=O TO 63:
GOSUB 200: PRÓXIMO VN: GOTO 150
190 FIM
200'
210 EM ERRO VÁ PARA 460
220 CÓPIA DE VOZ DE CHAMADA (ÿVN,VI)
230 EM ERRO IR PARA 0
240 PARA I=O A 15
250 VD(I livro 2)=VI(I) E 255
260 VD(h2+1)=(VI(I) I 256) E 255
270 PRÓXIMO I
280 NA$=""
290 PARA I=O A 8
300 SE VD(I) ENTÃO NA$=NA$+CHR$(VD(I) )
310 PRÓXIMO I
320 IMPRIMIR: IMPRIMIR "Vo Engenharia No. " ;VN; : "NA$
330 PRINT "Tr dia pose=";VI(4);
340 IMPRIMIR "Feedback=" , (VD(10) AND 14) I 2
350 PARA I=O PARA 3: VO(I)=VD(I+16): PRÓXIMO I
360 IMPRIMIR "Operador O": GOSUB 490
370 PARA I=O PARA 3: VO(I)=VD(I+24): PRÓXIMO I
380 IMPRIMIR "bocal 1": GOSUB 490
390 CHAMADA BGM(O)
400 VOZ DE CHAMADA (ÿVNÿÿVNÿÿVN)
410 PEÇA #2, "CED<G>CR" , "V6EGF<B>ER" , "V4GBADGR"
420 VOZ DE CHAMADA(@OÿÿOÿÿ0)
430 CHAMADA BGM(1)
440 EM ERRO IR PARA 0
450 RETORNO
460*** erro
470 IMPRIMIR "Voz No. . ";VN;" não tem dados.
480 RESUMO 440
490 *** imprime dados de um operador
500 IMPRIMIR "AM=" ;(VO(O) ¥ 128) E 1;
510 IMPRIMIR "PM=" , (VO(O) ¥ 64) E 1;
520 IMPRIMIR "EG=" ; (VO(O) ¥ 32) E 1;
530 IMPRIMIR "KSR=" ; (VO(O) ¥ 16) AND 1;

```

540 IMPRIMIR "MULT="VO(O) E 15;  
 550 IMPRESSÃO "LKS=" ; (V0(1) ¥ 64) E 3;  
 560 IMPRIMIR II TL=" ;V0(1) E 63  
 570 IMPRIMIR II ADSR=" ; (V0(2) ¥ 16) AND 15 " ,";  
 580 IMPRIMIR (V0(3) ¥ 16) E 15 " ,";  
 590 RETORNO  
 600 EM ERRO IR PARA 0

Existem dois tipos de operadores operando no programa. Operador-1 é um básico  
 O operador 2 modula 1 com um "operador portador" para criar a forma de onda.  
 É um "modulador/operador" para O significado de cada valor definido  
 Uma explicação, encheria um livro por si só, então vou omiti-la aqui. Livros de referência etc.  
 Gostaria que todos investigassem por conta própria usando o arquivo . Isso mesmo, a parte "PRINT" na Listagem 5.1  
 Se você alterar para "LPRINT" e imprimir uma tabela de dados de timbre, você mesmo poderá projetar o timbre.  
 Pode ser útil como material de referência.

#### 5.1.5 Experimentando sons de ritmo e zumbido usando uma fonte de som FM

O som em que as fontes de som FM, não apenas MSX-MUSIC, são fracas são os sons de percussão. instrumentos de percussão reais,  
 O som de percussão produzido por um sintetizador analógico é um ruído irregular, mas o som de uma fonte sonora FM é  
 Os sons de percussão sofrem de muita formalidade, resultando em um som "barato" ou "mecânico".  
 Torna-se um som.

Portanto, o MSX-MUSIC gera "sons de ritmo" além dos 63 tipos de sons de instrumentos musicais.

Funções são fornecidas para esse propósito. Para começar, entre os 63 tipos de sons de instrumentos musicais, também existem instrumentos de percussão.

No entanto, estes são tons sintetizados da mesma forma que os sons dos instrumentos musicais. Estou aqui

Os sons do ritmo são coisas completamente diferentes.

Como está escrito no manual, o MSX-MUSIC possui os canais 1 a 9.

Até 9 conjuntos de fontes de som FM de 2 operadores estão integrados. Tudo isso com sons de instrumentos musicais

Se utilizado desta forma, é possível tocar nove vozes.

Entretanto, como método para criar sons rítmicos, os canais 1 a 6 são comumente usados.

Atribua os 6 operadores para 3 conjuntos de canais 7 a 9 aos sons de instrumento padrão.

Também é possível usá-lo como som rítmico. Portanto, os recursos do MSX MUSIC são

Porém, é utilizada a expressão "9 sons de instrumentos ou 6 sons de instrumentos + 1 som de instrumento de percussão".

Para utilizar estas funções do BASIC, defina os parâmetros da instrução "CALL MUSIC".

Apenas mude. por exemplo,

CHAMADA DE MÚSICA (0,0, 1,1,1,1,1,1,1,1,1)

Existem 9 sons de instrumentos.

LIGUE PARA MUS IC (1,0, 1, 1, 1, 1, 1, 1)

6 sons de instrumentos + 1 som de instrumento de percussão são selecionados. .

Para referência, o seguinte é postado. O programa é criado usando dados de timbre do MSX MUSIC e seus próprios.

Um dispositivo para ouvir as diferenças nos sons rítmicos produzidos. Primeiro, vamos dar uma olhada no som do instrumento número 31.

2 O operador toca ``Bass Drum" 4 vezes, depois 6 operações do som do ritmo.

Reproduz o som do tambor de passagem quatro vezes. O som do ritmo é mais parecido com o de uma bateria real.

Vamos digitá-lo e ouvi-lo com nossos próprios ouvidos. .

## Listagem 5.2 (BASSDRUM.BAS)

```
10CHAMADA DE MÚSICA (1,0,1,1,1,3)
20 CHAMADA BGM(O)
30 VOZ DE CHAMADA(@31)
40 REPRODUÇÃO #2, "V15CCCC" ,",",",",11 11 , "RRRRB4B4B4B411
50 VOZ DE CHAMADA (@O)
```

Além disso, o MSX permite que você toque uma fonte de som FM e PSG ao mesmo tempo, para que você possa emitir sons de instrumentos usando a fonte de som FM.

No entanto, também é possível gerar sons de instrumentos de percussão e efeitos sonoros usando PSG. Porém, depende do modelo da unidade principal do MSX.

Como o equilíbrio entre o volume da fonte de som FM e o volume do som PSG é diferente,

Você precisará de um programa para ajustar o volume de acordo com cada máquina.



```

Boi28, COMPRIMENTO DO TESTE , /* 20 : S tambor */
Boi28, COMPRIMENTO DO TESTE , l* 22: s tambor */
Boi30, COMPRIMENTO DO TESTE , l* 24 : B mm / E */
Boi28, COMPRIMENTO DO TESTE , 26 : S tambor / E */
Boi28, COMPRIMENTO DO TESTE , 28 : S tambor / E */
Boi28, COMPRIMENTO DO TESTE , 30 : S tambor /* */
FM_END, 32 : fim do ritmo */

FM_VOL + 8, /Ho 33: Cap. 1 VOL = 8 /
FM_INST + 3, ho 34: Guiter *EU */
FM_SUSON,
FM_LEGOFF,
FM_Q , 6,
FM_04 + FM_C , COMPRIMENTO DO TESTE ,
FM_04 + FM_D , COMPRIMENTO DO TESTE ,
FM_04 + FM_E, COMPRIMENTO DO TESTE ,
FM_04 + FM_F, COMPRIMENTO DO TESTE ,
FM_04 + FM_G, COMPRIMENTO DO TESTE ,
FM_04 + FM_A , COMPRIMENTO DO TESTE ,
FM_04 + FM_B , COMPRIMENTO DO TESTE ,
FM_OS + FM_C, COMPRIMENTO DE TESTE ,
FIMFM ); l* final do cap. 1 */

```

```

VOID principal (argc, argv) argc ;
int

```

```

caractere {

```

```

    auto char fmwork [FMWORK] ; o endereço l* deve ser >= 8000H */ auto char
    fmbuf [256] ; O endereço l* deve ser >= 8000H Ki/
    char fmstat;

```

```

    1f ( (fmstat = fopen(fmwork) ) == 1) { puts( "
        ) else if (fmstat ==
        2) { puts
("Endereço incorreto y");

```

```

}

```

```

printf ( "fopen: endereço da área de trabalho é% 04Xy\l" (não assinado) fmwork ) ;

```

```

    memcpy (fmbuf fmdata fmstart
(fmbuf (char) TESTTIMES); do { , sizeof(fmdata));

```

```

        fputs ("Jogando.\015" , } while saída padrão) ;
(fmtest ( ) ); fputs ( "\nFim
do jogo.\n" , fmstop() ; fputs saída padrão);
("fmstop :
complete\n" , fmclose ( ) ; fputs(" saída padrão) ;
fmclose :
complete\n" , stdout); exit (0);

```

```

}

```

Agora, deixe-me explicar brevemente o programa. . Em primeiro lugar, o tamanho é "FMWORK"  
 Prepare "fmwork", que é uma matriz automática de bytes, passe esse endereço e escreva  
 Chame "fmopen" na biblioteca. Este array deve ser colocado no endereço 8000H ou superior.  
 Como existe, vamos declará-lo como automático em vez de estático. .

Pelo procedimento acima, 0 se a fonte de som FM for preparada com sucesso, 1 se não houver fonte de som FM  
 Porém, se o endereço de "fmwork" for menor ou igual a 7FFFH, 2 é retornado de "fmopen".  
 Deveria ser.

Neste caso, você deve ter cuidado antes de usar a fonte de som FM.  
 Isso significa que você deve chamar "fmclose" antes de o programa terminar.

と。 Seria inconveniente se o programa terminasse antes de "fmclose" ser chamado.  
 Portanto, esta biblioteca ignora **CTRL + C** A chave **Tomoe** **ESC** é usada.  
 todas as funções. Se quiser escrever .

seu próprio programa usando o código FM , pressione a tecla dez vezes.

É importante lidar adequadamente com erros e erros de disco. em todo o caso ,

**Tenha cuidado ao chamar "fmclose" antes de sair. .**

Agora chame "fmstart" passando o endereço onde os dados estão armazenados e os parâmetros do número de reproduções.  
 Como foi , FM BIOS começa a tocar imediatamente. Este FM-BIOS tem temporizador de 10%  
 projetado para funcionar por chamada, você pode avançar com o programa enquanto continua jogando.  
 Também é possível. Este programa exibe "Playing" na tela enquanto você joga.  
 Eu tentei assim.

"fntest" retorna 1 se estiver tocando e 0 se tiver terminado. Também "fmstop"  
 serve para encerrar a reprodução e inicializar o FM-BIOS.

### 5.2.2 Explique a visão geral da biblioteca

A Listagem 5.4 serve para definir funções e constantes para a biblioteca de fontes sonoras FM.  
 O arquivo de cabeçalho "FMLIB.H" .

#### Listagem 5.4 (FMLIB.H)

```

EU: fmlib .h : arquivo de cabeçalho para fmlib por nao-i
* em 31 de maio de 1990 * por nao-i em 24
de fevereiro de 1991 .
*
* mas nenhuma garantia ou suporte
yy
caractere externo fmopenO, *
VOID externo fmclose (y;
ok. ab ss ee FUFM aa
yy yy PpqqMm yyjjkkllpp
1 ss

```

```

extern VOID fmwrite O; extern VOID          Eu* escrevo no registro OPLL /* gravo      ÿÿ
fmotir O ;                                no registro OPLL 0 /* música de fundo /* ... 7 */ ÿ
                                           paro a música de fundo /* leio      ÿ
VOID externo          fmstop() ; livro      dados da ROM /* agora tocando /      ÿ
char fmtest O ;      fread O; extern        Tamanho do livro da área de      ÿ ÿ
                                           trabalho ?                          ÿ
                                           ÿ ÿ ÿÿ

#define FM_VOL_          Ox0060 Ivolume GOH ... 6FH Ox0070      ÿÿ
#define FM_INST         linstulment 70H ... 7FH Ox0080 /* sustentação      ÿÿ
#define FM_SUSOFF      off Ox0081 /* sustentação em            ÿ
#define FM_SUSON       Ox0082 lki expaÿdet instulment           ÿ
#define FM_EXPINST     Ox0083 /* instulment definido pelo usuário      ÿ
#define FM_USRINST     Ox0084 /* legato off Ox0085 l* legato em Ox0086      ÿ
#define FM_LEGOFF      lho Q                                    ÿ
#define FM_LEGON       ÿ                                       ÿ
#define FM_Q           ÿ                                       ÿ
#define FM_END         Ox00ff /* fim dos dados Ox00a0 /        ÿ ÿ
#define FM_RVOL        * volume de rhythblood                  ÿ ÿÿ ÿÿ ÿÿ */

/* livro de argumentos de venda/
#define FM_C_          /* c *ÿ
#define FM_CS 1 #define Eu* C# Ho/
FM D 2 #define FM_DS 3
#define FM_E 4 #define
FM F 5 #define FM_FS 5
#define FM_G 7 #define
FM GS 8 #define FM_A 9
#define FM_AS 10 #define
FM B 11 /* outubro ho/
-
-
-

#define FM_01 1 #define          l* FM_01+FM_C significa C de oitava O      Ho/
FM_02 13 #define FM_03
25 #define FM_04 37
#define FM_05 49 #define
FM_06 61 #define FM_07 73
#define FM_08 85
-
-

```

O próximo na longa lista, número 5.5, é a biblioteca de som FM. É o começo da lista?

Em ordem, defina endereços como BIOS, defina macros que chamam e emitem FM BIOS e defina bibliotecas.

São escritas a definição da área de trabalho utilizada pela biblioteca e o corpo principal do programa da biblioteca.

Já foi feito.

## Listagem 5.5 (FMLIB.Z80)

```

fmlib .z80 : librevisedyf ou MSX-C por nao-i em 29
de maio de 1990 (C) ASCII 1988 para (C) I
sikawa 1990 search' gratuito para uso e cópia, mas sem garantia ou suporte.

.Z80

endereço da área de trabalho do sistema BIOS a e d
;
rdslt equ calslt      000ch
equ enaslt           001 canal
equ breakv          0024h
equ ramad0           De325h      Vetor de quebra C
equ ramad1           De341h      ; slot # da RAM
equ ramad2           De342h
equ ramad3           De343h
equ exptbl          De344h
equ h.timi equ      Ofcc1h
                    De d9fh      ; timer int tomoe rrupt gancho
;
;
endereço da tabela de salto FM-BIDS
ds      4018h+4
wr      p            4110h
ir      r            4113h
ms      r            4116h
me      r            4119h
rc      41 1
or      corr 41
ts      1fh 4122h
;
;
MACROS para chamar FM-BIOS
;
CALLFM MACRO Id      ENDEREÇO
      Id            ix, endereço iy,
                  (biosslot-1)
      chamada      cal slt
      ENDM
;
JUMPFM MACRO        ENDEREÇO
      Id            ix, endereço iy,
      Id            (biosslot-1) calslt
      jp
      ENDM
;
;
... ssi      c*      ; slot do FM-BIOS ;
      aka      09      salvamento - Vetor C
      rnti      ;
      sld      ;
;

```



```

adicionar hl,de
resolucao 0,1 ; certifique-se de que o endereço seja par

push hl
call ld pesquisa
ld a, (biosslot) , (p.
ld haha ontime)
de,ontime.slot-ontime hl, de (hl)

adicionar
ld , um ; modificar LD IY, ??00
ou um
ld a,1
pop hl ; endereço da área de trabalho;
ret z sem FM-BIDS
CALLFM_iniop1 ld h,
40h , (dia ad1)
ld ; porque o sistema operacional de engenharia FM-B
chamar enaslt ; não restaura o slot1

;
ld hl,h.timi de,
ld (p. gancho antigo)
ld bc,5
ldir ; salve h.timi
ld hl, hooktbl
ld de,h.timi
ld bc,5
di
*1 ; se +U h +shiu+1m

am
fin
p
im 'h

;
ld haha , (quebra)
ld (quebra), hl , ; salvar vetor de quebra
ld hl, vvv
ld (quebra), hl ; definir vetor de quebra
ei
xor um
ret

;
;
;
;
fmclosey: :
di-
ld- hl, (quebra)
ld (quebrav), hl ; restaurar vetor de quebra
ak
db1 K, NoD Sya
t iw1
111 001
GC
ldir ei ; restaurar h.timi
ret
;

```





```

      ld      b,id_string_len
chkids_loop: push
      af be push push      ;salve o endereço do slot;
      call   de            ;salve o contador ;
      rdsit             ;leia um byte ; restaure o contador ;

      -----

      PPoh,pp          CE

      11 dd            a,na " em, J AU e

      CP      C
      Jr.     nz, diferente
      pop     de            ;não. ;restaurar endereço do slot ;ponto
      inc.    de            próximo
      inc     haha

      djnz chkids_loop bc
      pop    ;restaurar
      xor    um            ambiente ;encontrado definir sinalizador zero
      ret

;
diferem:
      pop    af            ;restaurar endereço do
      pop    a.C.         slot ;restaurar ambiente ;
      xor    um
      Inc.   um
      ret

;
      fim

```

Para explicar os principais pontos do programa, primeiramente, "fmopen@" é um temporizador de "ontime". Transfira o programa principal de processamento de interrupções para outro endereço e mova-o para o espaço onde o FM-BIOS está localizado. Usado para pesquisar um lote, inicializá-lo e definir o gancho de interrupção do temporizador. Prejudicar eu] Processamento. O programa e os dados aos quais ele faz referência devem estar localizados no endereço 8000H ou superior. Portanto, transfira o programa e substitua "ld iy, 0" por "ld iy, FM-BIOS". O número do lote foi alterado para "\*256".

Encontre o slot onde o FM-BIOS está localizado. O programa em si é Ascinet MSX. Citado nas especificações FM-BIOS publicadas em Para todos os slots, Encontre o FM-BIOS verificando se a string "OPLL" está no endereço 401CH. Sim.

Agora, a área de trabalho de 192 notas passada para "fmopen@" é para o programa de tratamento de interrupções. programa, local de armazenamento do conteúdo original de "h.timi", área de trabalho FM-BIOS (160 bytes) usado para. Neste momento, a área de trabalho do FM BIOS deve iniciar a partir de um endereço par. Por isso, é preparada uma área de trabalho extra e o endereço inicial é arredondado.

Isso não foi escrito nas especificações, mas se você configurar o FM-BIOS com "CALLFM \_iniopl" Após a inicialização, a página 1 voltará para outro slot.

havia. Você deve sempre retornar a página 1 ao seu slot original.

Como escrevi antes, se o programa terminar com o gancho de interrupção do temporizador reescrito.

Como seria um problema, mudei o endereço F325H da área de trabalho do MSX DOS para [bean@+{yo}]. Eu fiz ignorar a chave. Se o programa usar disco,

Você precisa adicionar um programa para lidar com o erro. E "fmclose@" é o cronômetro Isto é para desfazer a função de interrupção e o processamento da tecla @@+@].

Para o restante da biblioteca, preencha os registros com os valores desejados e configure o FM BIOS. Você pode usá-lo simplesmente ligando para . Vamos tentar várias coisas por conta própria .

### 5.2.3 Compilar com Ran1SX-C

Se você digitar a lista em um editor como MSX-DOS MED ou KID, você obterá a Lista 5.3. de "TESTFM.C"、リスト 5.4 を "FMLIB.H"、そしてリスト 5.5 を "FMLIB.Z80"

Não importa o que você diga, você . Compile na seguinte ordem filho: "TESTFM.COM" poderá abrir o nome do arquivo . Para executar o programa, na linha de comando do MSX-DOS:

Se você digitar TatTFM [:jy, será uma montanha.

### Listagem 5.6 (FMLIB.BAT)

```
m80 ,=fmlib.z80/r/m/z cf testfm cg
testfm
```

```
m80 ,=testfm.mac/
r/m/z 180 testfm,fmlib, ck,ciib/s,crun/
s ,cend,testfm/n/y /elianain
```

### 5.3 Estrutura de dados da fonte de som FM

Esta página explica a estrutura de dados de uma fonte de som FM e como especificar os dados da fonte de som. explicar. Usado em conjunto com o programa que opera a fonte de som FM na linguagem de máquina descrita anteriormente. vamos .

#### 5.3.1 Vamos criar dados de fonte de som FM

A página anterior mostra um programa para chamar o FM BIOS a partir da linguagem de máquina e produzir som. Apresentei um exemplo de RAM. Aqui apresentaremos uma fonte de som FM para brincar com esse programa. Vamos explicar como criar os dados para .

FM-BIOS. Se fôssemos resumir a estrutura de dados do FM-BIOS, veríamos que se trata de um grande array. matriz de Cada posição de dados na matriz é contada em bytes desde o início da matriz e é No entanto, o deslocamento do início da matriz é 0. Além disso, o deslocamento do início da matriz é 0. O começo é "0" "、その次は“1バイト目”というように、呼ばれることもある。

A Tabela 5.4 mostra a estrutura de dados de 6 sons de instrumento + 1 som de percussão e um exemplo de dados reais. o que foi mostrado. Como você pode ver, o corpo dos dados está organizado no final do array. O deslocamento no qual cada dado é colocado é gravado nos primeiros 14 bytes da matriz. Está escrito nele.

Tabela 5.4: Estrutura de dados de 6 sons de instrumentos e 11 sons de instrumentos de percussão

オフセット	Contente
0	Offset dos dados sonoros do instrumento de percussão (Nota 1)
2	Deslocamento de dados do som do instrumento 1 (Nota 2)
4	Deslocamento de dados do som do instrumento 2 (Nota 2)
6	Deslocamento de dados do som do instrumento 3 (Nota 2)
8	Deslocamento de dados do som do instrumento 4 (Nota 2)
10	Deslocamento de dados do som do instrumento 5 (Nota 2)
12	Deslocamento de dados do som do instrumento 6 (Nota 2)
14	Dados sonoros de instrumentos de percussão
(Nota 3)	Dados do som do instrumento 1
(Nota 3)	Dados do som do instrumento 6

(Nota 1) Sempre , AH escreva 2 bytes de OEH.

(Nota 2) Posição inicial dos dados sonoros do instrumento Desde o início dos dados nesta tabela deslocamento em bytes. byte , inferior , Compra principal

Especifique na seguinte ordem. Além disso, use Defina 0 para canais sem Especificar.

(Nota 3) Especificado por deslocamento Os dados de som do instrumento são colocados no local .

Se explicarmos a estrutura de dados em ordem, começaremos explicando o 0º byte e o 1º byte (offset). 0 e 1) são os deslocamentos onde estão localizados os dados sonoros do instrumento de percussão, e 14 é sempre escrito. Isso será feito. Este valor é escrito na ordem do byte inferior e do byte superior para que a CPU Z80 possa entendê-lo. Quando expresso em 2 bytes, torna-se OEH , .

Os seguintes 2º a 13º bytes (offset 2 a 13) contêm o canal de som do instrumento. O deslocamento de dados de 1 a 6 é colocado. Neste caso, o deslocamento 0 é especificado.

Tabela 5.5: Exemplo de estrutura de dados para 6 sons de instrumentos + 1 sons de instrumentos de percussão

Deslocamento	Conteúdo
0	OEH OOH
2	21H
4	OH OH
6	OH OH
8	OH OH
10	OH OH
12	OOH OOH 14--
Dados sonoros de instrumentos de percussão	
33 ~	Dados do som do instrumento 1

Esta é a estrutura de dados de 6 sons de instrumentos + 1 sons de instrumentos de percussão.  
 exemplo. Fonte de som de instrumento de percussão em 14 a 32 bytes  
**Os dados têm 32 bytes.**

Os dados da fonte sonora do canal são colocados e a primeira parte do som do instrumento é  
**Os canais 2 a 6 não são usados.**

Se estiver definido, esse canal não será usado.

Por exemplo, no exemplo de estrutura de dados na Tabela 5.5, o segundo e o terceiro bytes (deslocamentos 2 e 3) 21H e OOH estão escritos nele. O que isto significa é que o som no canal 1 do som do instrumento  
 Isso significa que os dados de origem estão localizados após o deslocamento 33. E o 4º byte?

OOH é escrito no 13º byte (offset 4 a 13), então o som do instrumento

Você pode ver que os canais 2 a 6 não são usados.

O programa para controlar a fonte de som FM apresentado anteriormente é um programa em linguagem C.  
 Agora que incorporamos os dados de teste no arquivo, precisamos contar o comprimento dos dados e especificar o deslocamento.  
 Foi necessário No entanto, se você pensar bem, poderá escrever um programa assembler como o mostrado abaixo.  
 Poderia ter sido mais fácil criar os dados da fonte sonora. Por favor, anote-o para referência.

DADOS FMD:

```
DW 14
DW CH1-FMDATA
DW CH2-FMDATA
DW CH3-HyATA
DW CH4-FMDATA
DW CH5-FMDATA
DW CH6-FMDATA
```

Dados sonoros de instrumentos de percussão DB. yy yy

CH1:

Dados do canal 1 do banco de dados. yy yy

CH2:

Dados do canal 2 do banco de dados. yy yy

(Doravante, o mesmo se aplica até CH6)

Isso significa. Com este programa, ao montar a lista de fontes,  
 Isso ocorre porque o assembler MSX-DOS (M80) calcula automaticamente o deslocamento.

Mais também, Este programa não pode ser usado para fins práticos como está. Palavras BÁSICAS  
 Uma macrolinguagem musical que cumpre o papel da instrução PLAY em japonês.

(MML) para dados de fonte de som FM BIOS. Você precisará de um registro

Se

quiser tocar 9 sons de instrumentos sem nenhum som de percussão, você pode usar um conjunto de dados como o listado na Tabela 5.6. estrutura de dados. Basicamente, a estrutura de dados é a mesma de 6 sons de instrumentos + 1 som de instrumento de percussão.

Do 1º byte ao 17º byte (offset O a 17), os sons dos canais 1 a 9 do instrumento soam

Especifique o deslocamento no qual os dados de origem foram gravados. Se OOH for especificado, Isto significa que o canal correspondente não será utilizado.

Além disso, os dados da fonte de som para o canal 1 sempre começam no 18º nó (deslocamento 18). Portanto, o valor 18 na notação 12H é sempre escrito no deslocamento dos dados do , AH (10 canal 1.

Tabela 5.6: Estrutura de dados de 9 sons de instrumentos

Deslocamento (byte)	Conteúdo
2	Deslocamento de dados do som do instrumento 1 (Nota 1)
2	Deslocamento de dados do som do instrumento 2 (Nota 2)
*	
*	
16	Deslocamento de dados do som do instrumento 9 (Nota 2)
18	Dados do som do instrumento 1
*	
*	
(Nota 3)	Dados do som do instrumento 9

(Nota 1) Sempre , OOH's escreva 12H 2 bytes.

(Nota 2) Posição inicial dos dados sonoros do instrumento Desde o início dos dados nesta tabela O deslocamento em bytes do byte inferior , e do byte superior são respectivamente Especifique no topo do topo . Além disso, use O para canais que não possuem Especificar.

(Nota 3) Especificado por deslocamento Os dados de som do instrumento são colocados no local .

A propósito, esta é uma observação lateral, mas , O FM-BIOS inicia com OEH nos dados da fonte de som?

Dependendo das 12H, a presença ou ausência de sons de percussão (ou seja, 6 sons de instrumentos + 1 som de percussão ou 9 sons de instrumentos) (apenas) foi decidido. Portanto, os dados sonoros dos instrumentos de percussão sempre começam no 14º byte (offset). 14), se não houver som de instrumento de percussão, os dados de som do instrumento do canal 1 serão sempre os 18 compassos. Parece que é necessário partir do 1º ponto (offset 18).

### 5.3.2 Para especificar dados sonoros de instrumentos de percussão

A Figura 5.4 mostra os detalhes dos dados sonoros do instrumento de percussão e da sequência de dados real. Aqui, Os cinco tipos de instrumentos de percussão são representados pelo alfabeto "BSTCH" e os dados sonoros são representados por números binários. Está expresso.

Primeiro, vamos especificar o volume de cada instrumento de percussão usando dados de 2 bytes como o mostrado abaixo. .

```
101BSTCH
0000VVV
```

Figura 5.4: Dados sonoros de instrumentos de percussão

b1 b6 bs b4 b3 b2 b1 bo	
1   B s T c H 1 o   Volume (0-15)	
1   B s T c H comprimento da nota (1-255)	

101 10000 Pa: Su Tambor

00000000 volume O

Caixa 10101000

00000001 dia 音量 1

001 10000 não.

00010100 comprimento do passo 20

00101000 Tarola

00010100 comprimento da nota 20 00101000 caixa

00010100 comprimento da nota 20 00101000 caixa

00010100 comprimento da nota 20 11111111 O inverno de Theta acabou

Tambor B ÿS

a caixa

T -tom

c shin não, f le

H chapéu alto

Para selecionar instrumentos de percussão, use BSTCH

**Em relação aos 5 bits escritos como**

Especifique 1 ou 0 para cada um. o volume é máximo

A quantidade de atenuação em relação ao volume e a duração do som

Exibe o intervalo entre a reprodução do som e a reprodução do próximo som.

estou assistindo .

Neste momento, na string de 5 bits de "BSTCH", especifique o instrumento de percussão cujo volume você deseja especificar com 0 se não for 1. Além disso, na parte "vvvv", especifique o volume de O a 15. Contém um valor (na verdade, um valor binário). Porém, o "volume" neste momento é relativo ao volume máximo. Como isso especifica a quantidade de atenuação, uma configuração de 0 produzirá o som máximo e uma

Por exemplo, se você aumentar o volume de um tambor de passagem para , configuração de 15 produzirá o som mínimo, portanto, tome cuidado .

Para fazer isso,

```
10110000
00000000
10101010
00000001
```

Tudo que você precisa fazer é especificar.

Depois de especificar o volume, vamos especificar a duração da nota para cada instrumento de percussão. No entanto, instrumentos de percussão Como a duração do som em si é sempre constante, "duração do tom", neste caso, refere-se à duração de tempo entre o primeiro som e o próximo som.

Refere-se ao intervalo até o lançamento. e,

00 1BSTCH

especifica o tipo de instrumento de percussão e o próximo byte especifica o comprimento da nota (até 255). fazer. Para especificar uma duração de nota maior que 255, primeiro escreva 255 e depois insira a duração real da nota.

Especifique um valor que exceda 255. Se este valor for 255 ou mais, repita a mesma operação.

estômago. Por exemplo,

```
00110000
11111111
00000000
```

representa um tambor de passagem, comprimento de nota 255,

```
001 1 0010
111 11111
1 1 111111
1111 1111
1110 1011
```

representa um tambor de passagem, um simpal e uma duração de nota de 1000.

As especificações para a fonte de som FM não mencionaram uma única "duração do tom", mas os dados do teste mostraram que Como resultado das medições reais, a unidade de duração do som foi de 60 l segundos, que é igual ao ciclo de interrupção do temporizador.

### 5.3.3 Vamos especificar os dados sonoros do instrumento

A Tabela 5.7 mostra os detalhes dos dados sonoros dos instrumentos musicais. Nestes dados, a tabela l Somente os valores de byte têm significado e as combinações de valores subsequentes de 1 ou 2 bytes.

Existem algumas coisas que têm significado quando combinadas.

A ordem na qual você realmente especifica os dados de som do instrumento é volume, tom e sustentação. , Legato , Q A ordem é

A especificação do volume é igual aos dados do instrumento de percussão e é expressa como a quantidade de atenuação do volume máximo. Em outras palavras , O produz o som mais alto, 15 produz o som mais baixo e assim por diante.

Sustain é usado para ajustar a atenuação do som de um instrumento. O envelope do som do instrumento está na frente. Conforme explicado acima, é determinado por um valor denominado "ADSR". Para repetir, A está anexado. (velocidade de subida), D é dei k (atenuação), S significa sustentação (força de sustentação) e R significa liberação (velocidade de desaparecimento).

O "ADSR" dos tons integrados do OPLL é fixo para cada tom. Mas a sustentação Quando ativado, a liberação fica mais lenta e o som fica mais longo. Além disso, a sustentação é variável. Como você pode atribuir sons de guitarra aos canais 1 e 2, você pode Também é possível fazer pequenos ajustes, como ativar a sustentação apenas para o canal l.

Quando o legato está ativado, uma nota é conectada à próxima nota. no entanto, Se você usar muita música leger, a música perderá a nitidez, então você pode querer usar apenas alguns canais. Seria necessário inventar uma forma de ligar o legato.

Tabela 5.7: Dados sonoros do instrumento

値 Significado do valor	意味
Resto	OOH, comprimento do tom <1 byte
01H	Oitava 1 C, seguido por 1 byte é a duração do som
⋮	⋮
5FH	A# na oitava 7, o 1 byte seguinte é o comprimento do tom
60H	volume 0 (volume máximo)
⋮	⋮
6FH	volume 1 5 (volume mínimo)
70H	tom 0 (tom interno da ROM ou tom especificado pelo usuário)
71H	tom 1 (violino)
⋮	⋮
Tom	7FH 1 5 (baixo elétrico)
Sustentação	80H desligada
81H	Sustentação ativada
O 1 byte após 82H (OOH a 3FH) é o número do tom integrado da ROM.	
Os 2 bytes após 83H (inferior, superior) são o endereço dos dados de tom.	
84H	Legato desligado
85H	Legato Ligado
1 byte (OIH-08H) após 86H é Q	
Fim dos dados FFH	

Os valores que podem ser especificados para Q são de 1 a 8 e representam a razão entre a duração da nota e a duração do som real.

Por exemplo, se  $Q = 6$  e a duração da nota for 80, então uma nota de duração  $80 \times 6 \div 8 = 60$  será produzida.

,  $80 \times (8 - 6) \div 8 = 20$  dias de folga.

A combinação acima de valores especificados para legato e Q cria uma conexão entre as notas.

de sustentação determina a suavidade da música.

Tabela 5.8: Exemplo de dados sonoros de instrumentos

68H	volume 8
73H	Tom 3 (guitarra)
81H	Sustentação ativada
84H	legato desligado
86H, 06H	Q = 6
25H, 14H	oitavas 4 de dó, comprimento da nota 20
27H, 14H	Ré na oitava 4, comprimento 20
29H, FFH, FFH, FFH, EBH	oitava 4 E, comprimento do tom 1000
FFH	Fim dos dados T

A seguir, o valor de 1 byte de OOH a 5FH que especifica a escala e a duração de cada nota é a escala.  
O próximo 1 byte representa o comprimento da nota. por exemplo,

25H , 14H

Os dois bytes de dados para cada nota são C na oitava 4, e o método para Auto , Representa a duração da nota 20.  
representar comprimentos de notas de 255 ou mais é o mesmo que para instrumentos de percussão, por exemplo:

29H , FFH , FFH , FFH , EBH

Esses dados de 5 bytes representam a oitava 4 de E e o comprimento da nota 1000.

### 5.3.4 O que não pode ser feito com OPLL seco/

Dentre as funções do FM-BIOS , o temporizador é chamado por uma interrupção e dado .  
Um dispositivo que reproduz dados automaticamente é chamado de “driver OPLL”. Aqui, aqui,  
BÁSICO usando Drino.

#### CHAMADA

CHAMADA DE TEMPERAMENTO

TRANSPOSIÇÃO DE CHAMADA

Eu estava planejando explicar como obter a mesma funcionalidade do . No entanto, ao abrir o FM-BIOS,  
Quando entrei em contato com o criador, eles me pediram para fazer isso . Em outras palavras , Auto  
sozinho, e descobri que não conseguiria fazer isso sem reescrever o registro OPLL.

No final, com o driver OPLL, A normalmente é 440 Hz a uma taxa média de 12  
Diz-se que só é possível atuar com o temperamento correto. O MML suportado pelo BASIC inclui volume  
Existe uma função para definir o valor em detalhes e uma função para escrever o valor no registro do chip da fonte sonora.  
É impossível fazer a mesma coisa com o driver FM-BIOS . Além disso , o grupo de pacotes de jogos  
Também é difícil produzir efeitos sonoros enquanto toca música na rodada.

Quando você pensa dessa forma, para tornar uma fonte de som FM fácil de usar, você precisa fazer o que acabei de mencionar.  
Um driver com funções adicionais e um programa que converte MML em dados para esse driver.  
Parece que vou precisar de um grama. O artigo até agora e a fonte de som FM que será apresentada posteriormente  
Se você tiver um livro de referência, deverá ter todas as informações necessárias. Se você está confiante no programa, por favor  
— , Vamos tentar .

### 5.3.5 Adicione dados de tom e experimente

Como escrevi antes, existem 15 tipos de fonte de som FM LSI (OPLL) que podem ser controladas.  
A ROM FM-BIOS que rola tem 48 tipos de dados de tom disponíveis. Kere

Figura 5.5 Dados de tom

b1	b6	b4	b3	b2	b	b.
SOU VIB EGT	SOU VIB	Eu KSR	Múltiplo			Op.O
EGT I KSR	KSL (Op.O)	Múltiplo			Op.I	
MODELADOR DE NÍVEL Total						
KSL (Op.I)Free I DC-DW I Feedback Attack (Op.O)						
Decadência (Op.O)						
Ataque (Op.I)						
Decadência (Op.I)						
Sustentar (Op.O)						
Lançamento (Op.O)						
Sustentar (Op.I)						
Lançamento (Op.I)						

Op.O é um operador modulador, Op.I é um operador portador. Representa o operador. Esses 8 bytes são registros OPLL 0-7. escrito em. Para detalhes, veja "Como usar MSX2 + Powerful (publicado por ASCII)" Ver.

Todo , com uma estrutura de dados conforme mostrado na Figura 5.5, Você também pode adicionar mais tons. mundo está se acostumando. .

Esses dados de tom de 8 bytes são armazenados nos registros O a 7 da fonte de som FM LSI (OPLL). escrito em . É um comando estendido do BASIC. .

VOZ DE CHAMADA  
CÓPIA DE VOZ DE CHAMADA

Nos dados de 32 bytes usados para , Observe que os formatos são diferentes. o OPLL e no OPLL, o tom para cada canal é especificado como um valor de 0 a 15. . Em outras O tom O é especificado nos registros O a 7 do OPLL. O tom original a ser definido. Apenas um, se você criar seus próprios dados de tom ou se eles estiverem armazenados na ROM do FM-BIOS. tipo de dados de tom pode ser usado por vez.

Isso foi causado pela limitação do número de tons. . Esta não é uma limitação no número de canais. é por isso , Por exemplo, atribua seus próprios tons aos canais 1 e 2. , Canais 2-4 Os sons incorporados no OPLL , Também é possível alocá-lo.

Eu poderia escrever um livro inteiro apenas explicando o conteúdo dos dados de tons. Vou omitir desta vez. Em vez disso, apresentarei o livro de referência .

"MSX2 10 Powerful Usage Methods" escrito por Sei Sugitani e publicado pela ASCII Publishing Bureau.

No entanto, parece haver alguns erros neste livro. As correções são postadas na página seguinte. Vou deixar como está, então vamos consertar você mesmo. .

Além disso, se você estiver usando comunicação por computador, deverá ter acesso ao ASCII. Uma placa chamada "msx.spec" para MSX. Aqui, o programa em linguagem de máquina é MSX-

As especificações "FM BIOS" para usar MÚSICA foram lançadas. Além disso, isso é tudo Naku, Ele contém uma variedade de informações sobre MSX.

### 5.3.6 Explicando dados de amostra

Finalmente, como resumo, vamos apresentar um exemplo de especificação real de dados. A lista 5.7 faz parte da lista de programas que publiquei anteriormente.

Primeiro, a parte superior especifica o deslocamento de dados para cada canal. Bater Os dados de som do instrumento são do 14º byte ao 33º byte dos dados do canal 1 de som do instrumento? O começo, Indica que os canais 2 a 5 não são usados.

A parte central da lista contém dados sonoros de instrumentos de percussão. Primeiro, ajuste o volume de todos os instrumentos de percussão para 8. Está definido. O valor de "FM\_RVOL" é OxaO, e adicionar 31 a ele dá Oxbf, ou Isto significa que você pode especificar o volume de todos os instrumentos de percussão. Continue com o tambor de passagem na nota 20 e a caixa. O valor de "FM\_END" é Oxff e o valor de "FM\_END" é Oxff. Indica o final dos dados.

A metade inferior da lista contém os dados do canal de som do instrumento 1. Primeiro, ajuste o (Guitarra embutida), Sustain On Remifa, volume para 8 e o tom para 3 (OPLL legato desligado, Q=6). Solashido" com duração de cada nota de 20, e termina quando atinge "F LEND". Neste momento, você pode, alterar diretamente a escala de 0 a 95. Como é inconveniente especificar valores numéricos em escala de 12 tons e oitava, Eu entendo. por exemplo,

```
FM_04 + FM_D
```

Portanto, especificamos D na oitava 4. "FM-04" tem valor 37 e "FM\_D" Como o valor de é 2, somando-os dá 39, que representa Ré na oitava 4.

Além disso, para criar dados de teste usando assembler (M80), defina as constantes como segue. Seria conveniente definir.

```
F Reichi C EQU 1
H"CS EQU 2
:
:
HLVDL EQU 60H
:
:
```

Em seguida, ajuste o volume.

```
BD FM_VOL + 8
```

Como. Da mesma forma, a escala,



## Fonte de som 5.4 FM

Justamente quando pensei que tinha explicado tudo sobre as fontes de som FM, ou percebi que havia deixado algo de fora. Já faz um tempo, Por favor, cuide de mim.

### 5.4.1 Correção do conteúdo da Lei de Utilização Poderosa

"How to make full use of MSX2+" (ASCII) foi introduzido como um livro de referência para máquinas MSX2+. Vários erros foram encontrados na publicação (preço: 1.240 ienes [imposto incluído]). Como está escrito no livro Tenho certeza de que muitas pessoas iniciaram um programa, mas ele não funciona como planejado e elas ficam com a cabeça entre as mãos. criança Nesta página, corrigirei quaisquer erros atualmente conhecidos. se Além disso, se alguém encontrar algum erro diferente dos listados, Departamento Editorial da Revista M aqui, por favor me avise.

Bem, vamos começar com a primeira correção. "Tabela 4.4 Sons" na página 147 Vamos dar uma olhada na lista da biblioteca de cores. Entre estes, o tom número 10 e o nome do tom são "Guitarra". Adicione "2 guitarras" à coluna "OPLL VOICE" do item que diz "2 guitarras".

Houve também alguns erros nas "abreviaturas" listadas nesta tabela. Sobre isso Agora, ao executar o programa (READFM.BAS) listado na página 136 deste livro, a abreviatura correta será exibida. Enquanto toca cada tom, Por favor, verifique as abreviaturas.

A seguir está a explicação sobre "CÓPIA DE VOZ" na página 148. No meio da frase ``O número do tom que pode ser especificado na fonte (parâmetro - 1) é o número do tom especificado na coluna OPLL VOICE de 0 a 63", mas a resposta correta é ``especificada". ``Este é o número de OPLL

VOICE." Da mesma forma, um pouco mais tarde, ``Este é o número de OPLL VOICE na fonte." A afirmação de que "especificar um número de cor resulta em uma chamada de função ilegal" também é invertida. Corretamente O resultado é ``Se você especificar o número do timbre especificado..."

Além disso, o registro OPLL publicado nas páginas 151 a 158 Também houve vários erros na tabela. A versão corrigida está resumida na Figura 5.6. Eu preparei para você, então use-o como referência. Baseado nisso, usei o poderoso MSX2+ Por favor, revise o método de uso. Eu acho que é conveniente.

Em relação às descrições dos registros, Também houve erro, Número F da frequência alvo na fórmula de cálculo de BLOCOe BLOCO na página 155. Postado na parte inferior,

$$\text{Número F} = (440 \times 218 \div 50000) \div 24 - 1 = 288$$

A fórmula é, corretamente,

$$\text{Número F} = (440 \times 218 \div 50000) \div 2 (4 I) = 288$$

Isso é o que significa.

Finalmente, embora isto não esteja limitado ao uso poderoso do MSX2+, o som FM

Parece haver uma explicação incorreta em geral sobre como especificar os dados sonoros do instrumento em relação à fonte.

Então, por favor, corrija. .

Para especificar a escala de cada nota, OOH a 5FH é igual a C a oitava 1.

Costuma-se dizer que corresponde à oitava 7 até B, mas isso está incorreto.

Figura 5.6: Lista de registros OPLL

	b1	b5	b4	b3	b2	b1	bom							
04H	AM(M) VIB(M)	EGT(M)	KSR(M)				Múltiplo(M)							
01H	AM(C) I VIB(C)	EGT(C)	KSR(C)				Múltiplo(C)							
02H	KSL(M)		Modelador de nível total											
03H	KSL(C)		DC DMI grátis	DM	Opinião									
04H	Ataque (M)			Decadência (M)										
05H	Ataque (C)			Decadência (C)										
06H	Sustentar(M)			Rele e (M)										
07H	Mancha (C)			Rele e(C)										
O.E.H.	Vago	R	Registro de teste BD SD		CT	AH								
0FH	ROM CT													
10h	Número F													
18h								Número F						
20H														
28H	Vago	Sus.	Chave	Bloquear	Número F									
30H	Inst.			Vol.										
38H														

No modo ritmo

	b1	b5	b4	b3	b2	b1	bo
36H	vago	Bumbo					
37H	Eu oi chapéu, eu caixa						
38H	Tom Tom I Top Cymbal						

A parte marcada (M) na tabela funciona como módulo (operador 0, Os itens marcados (C) indicam o Operador 1, que atua como transportador. Para mais informações, consulte "Como fazer uso completo do MSX2+" ou a documentação técnica da Yamaha. coisa .

estômago. Corretamente, OOH é uma pausa, seguida de OIH a 5FH, que corresponde a oitava 1 C a oitava. Compatível com A# da Turbina 7.

## 5.4.2 Lista de dados de tons do MSX-MUSIC

Em resposta aos pedidos dos programadores, incorporamos isto na ROM do MSX-MUSIC. Vou postar uma lista dos dados de tom que foram criados.

Listada no lado esquerdo da Tabela 5.9 está a instrução BASIC "CALL VOICE COPY".

Os 32 bytes de dados de tom obtidos do

Extraí 8 bytes de dados. Existe um processo denominado "transposição de voz" nos dados de timbre. Ele também contém dados de 2 bytes que controlam a altura do som, diretamente para o registro OPLL, mas como não são dados gravados diretamente, eu os omiti.

O tom 60 e o tom 61 parecem exatamente iguais nesta tabela, mas os valores de transposição de voz são diferentes. Na verdade, tem um tom diferente. Além disso, na tabela "usando dados do OPLL" Em relação ao número do tom escrito, é utilizado o tom embutido no OPLL, portanto a ROM não contém dados de timbre.

Os 63 tipos de dados de tons internos da ROM obtidos com FM-BIOS são baseados neste BASIC. Todos acreditavam que este era o mesmo que os dados de tom integrados da ROM. Meu coração está na verdade Recentemente ficou claro que existem diferenças entre os dois. Portanto, o lado direito da Tabela 5.9 O lado contém 8 bytes para cada tom obtido usando a função "RDDATA" do FM-BIOS. São dados de tom.

Como você pode ver comparando os lados esquerdo e direito da Tabela 5.9, os números dos tons para Extended BASIC e FM-BIOS são A correspondência entre números e nomes é a mesma, mas os dados de tom são ligeiramente diferentes. Portanto, BÁSICO Se você deseja criar um protótipo de música usando MML e converter os dados para FM-BIOS. Em alguns casos, as diferenças de timbre podem se tornar um problema.

Observe que a maioria dos dados de tom no FM-BIOS são gravados no registro 3. Como o bit 5 da estrela 3 está "vazio", Algumas pessoas podem achar isso estranho. Mas a caixa registradora o valor escrito no registrador 3 é 20H. Mesmo que sejam 20H, Mesmo que este não seja o caso, o tom realmente tocado será o mesmo.

Tabela 5.9: Lista de dados de tons

番号	音色名	拡張BASICの音色データ	Dados de tom FM-BIOS
0	Piano 1	usando dados de OPLL(3) 31 11	OE 20 D9 B2 11 F4
1	piano 2	30 10 DE 04 D9 B2 10 F4 30 10 DE	20 D9 B2 10 F3
2	violinos	usando dados de OPLL(I) 61 61 12	20 B4 56 14 17 usando dados de
3	Flauta 1	OPLL(4) usando dados de	61 31 20 20 6C 43 18 26
4	Clarinete	OPLL(5) A2 30 AO 20 88 54 14 06	usando dados de OPLL(6) 31 34 20 20
5	oboé	72 56 OA 1C usando dados de OPLL	(7) 31 71 16 20 51 52 26 24 34 30 37
6	Yrum pet	06 50 30 76 06 34 30 37 20 50 30 76 06	
7	Órgão de tubos 1		
8	Xilofone	17 52 18 05 88 D9 66 24 17 52 18 20 88 D9 66 24	
9	Órgãos	usando dados de OPLL(8) EI 63	OA 20 FC F8 28 29 usando dados
10	guitarras	de OPLL(2) 02 41 15 20 A3 A3 75 05	
11	Santool 1	19 53 OC 06 C7 F5 11 03 19 53 OC	20 C7 F5 11 03
12	Piano Elétrico 1	usando dados de OPLL (15) 23 43	09 20 DD BF 4A 05
13	C lavicódigo 1	03 09 11 06 D2 B4 F5 F6 03 09 11	20 D2 B4 F4 F5
14	Harpicódigo 1	usando dados de OPLL(II) 01 00	06 20 A3 E2 F4 F4
15	Harpicódigo 2	01 01 11 06 CO B4 01 F7 01 01 11	20 CO B4 01 F6
16	Vibrafone	usando dados de OPLL (12) F9 F1	24 20 95 DI E5 F2 13 11 oc 06 FC D2
17	Koto 1	33 84 13 11 oc 20 FC D2 33 83	
18	Taiko	01 10 OE 07 CA E6 44 24 01 10 OE	20 CA E6 44 24
19	Motor 1	EO F4 1B 87 11 FO 04 08 EO F4 1B	20 11 FO 04 08
20	OVNIs	FF 70 19 07 50 1F 05 01 FF 70 19	20 50 1F 05 01
21	Syn th Tomoe sizer Sino	13 11 11 07 FA F2 21 F5 13 11 11	20 FA F2 21 F4
22	campainha	A6 42 10 05 FB B9 11 02 A6 42 10	20 FB B9 11 02
23	Baixo Sintetizador	usando dados de OPLL(13) 40 31	89 20 C7 F9 14 04
24	Sintetizador usando dados de OPLL(10) 42 44	OB 20 94 BO 33 F6	
25	Sintetizador Percussão 01 03	OB 07 BA D9 25 06 01 03 OB 20 BA D9 25 06	
26	Sintetizar Ritmo 40 00 00 07	FA D9 37 04 40 00 00 20 FA D9 37 04	
27	Tambor de danos	02 03 09 07 CB FF 39 06 02 03 09	20 CB FF 39 06
28	Campainha	18 11 09 05 F8 F5 26 26 18 11 09	20 F8 F5 26 26
29	Fechar o chimbal	OB 04 09 07 FO F5 01 27 OB 04 09	20 FO F5 01 27
30	Caixa	40 40 07 07 FAÇA D6 01 27 40 40 07	20 FAÇA D6 01 27
31	Bumbo	00 01 07 06 CB E3 36 25 00 01 07	20 CB E3 36 25



# APÊNDICE



## Tabela de instruções R800

24 de janeiro de 1991

ASCII Co., Ltd.

Divisão de Sistemas, Departamento Editorial da Revista MSX

Se você é apaixonado por programação em nível de linguagem de máquina, definitivamente deveria tentar.

A primeira coisa a fazer é desenvolver programas no R800. Mnemônicos, operações/De comando,

Publicamos uma folha de instruções com um código de licença para você usar.

Sim. Kia , Um ventilador que aproveita a velocidade do R800. Posso programá-lo?

### A.1 Instruções Como usar a mesa

Esta tabela também resume as instruções do R800, categorizadas por tipo de instrução.

de. Na tabela, "mnemônico y" representa o nome de cada instrução e "ação de comando" representa o conteúdo de sua operação. é mostrado de forma concisa.

"y" na coluna de ação de comando indica que o conteúdo do lado direito é atribuído ao lado esquerdo. , Entre parênteses

O conteúdo da memória indicado pelo registrador vazio, etc.

Cada um tem seu próprio significado. por exemplo,

r y [.hl]

O conteúdo da memória no endereço indicado pelo registrador .hl é transferido para o registrador de 8 bits.

Isso significa atribuí-lo à estrela. No entanto, os comandos de entrada/saída [n] e [c] são os comandos de entrada/saída correspondentes. Significa o número da porta.

A coluna "Flag" descreve a operação de cada flag e a coluna "Opcode" descreve a instrução de máquina para cada instrução.

O código do idioma japonês é escrito em números binários e hexadecimais. "B" e "C" no lado direito são cada instrução

É chamado de comprimento (número , Expresse o número de clocks necessários para executar cada instrução.

de bytes).

Além disso, os nove exemplos a seguir estão resumidos em relação às abreviaturas que aparecem na tabela de ilustrações.

Por favor, use isso como referência. Além disso, os mnemônicos listados na tabela são diferentes daqueles do Z80.

A razão de estar lá é porque é um trabalho protegido por direitos autorais da Zilog. Dito isso, , Potência adicional do R800

Exceto para instruções aritméticas e instruções cujo funcionamento não foi oficialmente garantido no Z80, o mnemônico

Apesar das diferenças, todas as instruções se comportam da mesma forma. Instruções Z80

Gostaria que você programasse comparando com a tabela.



## A.2 Comando de movimento de 8 bits

11Mook	命令動作	sinalizadores de s z H também	オペコード N c 76543210 Hex B	C
ldr,r'	rŷŷr'	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	01r r'	c 1 1
ld r,n	rŷŷn	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	00 r 110	você > dar
ldr,[r,n],r'	rŷŷ-[r]r'	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	01 r 110	1 2
r,[ix+d]	ŷ.[ix+d]	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	1101 1 101 01 r 110 ŷdŷ	DD 3 5
ldr,[ix+d],r'	r←[.iy+d]	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	11 111 101 01 r 110 ŷdŷ	FD 3 5
ld [[r],r',r	[.hl]r←r	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	01 110 r	1 2
ld [[ix+d],r],r	[.ix+d]r'←r	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	11011 101 01 110r ŷdŷ	DD 3 5
ld [[iy+d],r],r	[.iy+d]r'←r	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	11111 101 01 110r ŷdŷ	FD 3 5
você, você'	vocêŷŷvocê'	ŷ*****	1 101 1 101 01 uu	DD 2 2
ldv,v'	vŷŷv'	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	11111 101 01 v v'	FD 2 2
ldu,n	vocêŷŷn	ŷ*****	11011 101 00 você vai ŷŷŷŷ	DD 3 3
ldv,n	vŷŷn	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	eu vou 101 00 v 110 ŷŷŷŷ	FD 3 3
ld [[hl],n	[.hl]to-n	ŷ*****	0011010 Conflicto - n-ŷ	36 2 3
ld [[ix+d],n	[.ix+d]ŷŷn-n	ŷŷŷŷŷŷ	11011 101 00 110 110 ŷdŷ ŷŷŷŷ	DD 4 5 36
ld [[ix+d],n	[.ix+d]ŷŷn-n	ŷ*****	11 111 101 00 10 10 ŷdŷ ŷŷŷŷ	FD 4 5 36

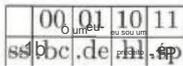
二一 番号	命令動作	código de operação de flags			
		s z H Py N c 76	543 210	Hex BC I!	
ld .a,.i	.ayy.t	---	1 1 101 10	ED 2 2	
ld.a,.r	.ayy.r				
ld .i,.a	.iyy.a			0 1 000 111 47	
ld.r,.a	.ryya .	yy yy yy yy	yy 1 1 101	101 ED 2 2	
				0 1 001 111 4F	
ld.a .bc	.ayy.bc .a	yy yy yy yy	yy 00001 0	10 OA 1 2	
ld.a .de	yy .de	yy yy yy yy	yy 0001 1	010 IA 1 2	
ld .a, nn	.ay 1nn	•yyy yy 00 111	010 3A 3 4		
		yy nn yy			
		yy nn yy			
ld [bc].a	[bc]y .a	yy yy yy yy	yy 00000 0	10 02 1	
[de].a	[de]y .a	2 y yy yy yy	yy 000100	10 12 1	
[nn].a	[nn]y .a	2 y yy yy yy	yy 00 1 100	10 32 3 4	
		yy nn t yy			
		yy nn ..			

	000	001	010	011	100	101	110	111	c
r .b	eu .d .e	eu .h	eu .1	eu .a .c					
u .b	eu .d .e	ixhl .ixl	eu .a .e	l .iyhl .iyll					
v .b	eu .a .c	ixhl .ixl	eu .a .e	l .iyhl .iyll					

A.3 Comando de movimento de 16 bits

二一 番号	命令動作	sinalizadores de código de operação			
		s z H também	N c 76 543 210	Hex	
LDS, nn	SSyy nn	y*****00 ss O OOI			BC 3 3
		yy nn t yy			
		yy nnh yy			
ld.ix,nn	.ixyy nn	yy yy yy yy	yy 11011 101	DD 4 4	
			00 100001 21		
		yy nn t yy			
		yy nnh y			
ld .iy,nn	.iyyy nn	yy yy yy yy	yy 1 1 111 101	FD 4	
			4 00 100001 21		
		yy nn t yy			
		yy nn h yy			
ld .sp, .hl		yy yy yy yy	yy 1 1 111001	F9 1	
ld .sp,.ix	.Spý.hl .Spý.ix	1 y yy yy yy	yy ll Oll 101	DD 2 2 ll	
			llOOl F9		
ld .sp,.iy	.spý.1y	yy yy yy yy	yy 1 1 111 101	FD 2 2	
			1 1 111001 F9		

Mnemônico	命令動作	operação	sz H vergonha	de código de operação	N c 76 543210 Hex BC
ld ss, [nn]	ss <sub>h</sub> ← [nn+1] SSy y yy j +lj	•••••1 1 101	101 ED 4 6	ss 10101 yy 11111 yy yy 1111 horas yy	
ld hl, [nn]	[nn] +lj : yy y <sub>n</sub> 11J	yy yy yy yy yy	yy 00 10 1 010 2A 3 5	yy nn1 yy yy nn h yy	
ld ix, [nn]	ixy [nn+1]ixy [nn]	yy yy yy yy yy yy	yy 1 1 011 101 DD 4 6	00 101 010 2A yy 1111, yy yy nn h yy	
ld iy, [nn]	.iy1yyynn J+lj 0.1 ano	yy yy yy yy yy yy	yy 111 11101 FD 4 6	00 101 010 2A yy nn1 yy yy nn h yy	
ld [nn], ss	[nn+1]y <sub>ss</sub> ← ss <sub>h</sub> [nn] ← ss <sub>l</sub>	•••••1 1 101	101 ED 4 6	ss ss ODII yy nn, umy yy nnh yy	
ld [nn], hl	[nn+1]h y <sub>l</sub> .h [n]	•••••00 100 010	10 22 3 5	<-- n n1 + yy llh yy	
ld [nn], ix	[nn+1]y <sub>ix</sub> .ixh [nn]y <sub>ixl</sub>	yy yy yy yy yy yy	yy 11011101 DD 4 6	00 100 010 22 yy nn t yy yy nn h yy	
ld [nn], iy	[nn+1]y <sub>iy</sub> .iyh [n]gritar y <sub>iy</sub>	•••••1 1 111	101 FD 4 6	DO 100 010 22 yy 11111 _ , yy nn h yy	



A.4 Instruções de troca

11Mook	命令動作	sinalizadores de código de operação						
		sz	H	P,y	N c	76 543 210	Hex	BC
xch .de, .hl	.de←.hl	yy	yy	yy	yy	yy	1 1 101	011 EB
xch .af, .af'	.af←.af'	1	1	111111	00001	000	08 1 1	
xch j.spjyhl	.hl←[.sp] ; .hl←+.hl.sp+1	yy	yy	yy	yy	yy	1 1 100011	E3 1 5
xch l.spjix	.ix←[.sp] ; .ix←+.ix.sp+1	yy	yy	yy	yy	yy	1101 1 101	DD 2 6
xch l.spjyiy	.iy←[.sp] ; .iy←+.iy.sp+1	yy	yy	yy	yy	yy	1 1 100 0 1 1	E3
xchx	.bc←+.bc ; .de←+.de ; .hl←+.hl	yy	yy	yy	yy	yy	11011001	D9 1 1

A.5 Instruções de operação da pilha

11Mook	命令動作	sinalizadores de código de operação						
		SZH	Bill	N c	76 543 210	Hex	BC	
empurrarqq	[.sp-2] ← [Sp]; Sp ← Sp-2	yy	yy	yy	yy	yy	11qq O 101	1 4
push.ix	[.sp-1] ← [ix]; Sp ← Sp-1	yy	yy	yy	yy	yy	1 1 01 1 101	DD 2 5
push.iy	[.sp-1] ← [iy]; Sp ← Sp-1	yy	yy	yy	yy	yy	1 1 100 101	E.S.
pop qq	Sp ← Sp+1; qq ← [Sp]	yy	yy	yy	yy	yy	0 001 qq	1 3
pop.ix	[ix] ← [Sp]; Sp ← Sp+1	yy	yy	yy	yy	yy	1101 1 101	DD 2 4
pop.iy	[iy] ← [Sp]; Sp ← Sp+1	yy	yy	yy	yy	yy	1 1 100 00 1	EI

oo	eu	o	eu	10	11
10	11	00	01	01	01

Quando pop .af, todos os sinalizadores mudam.

A.6 Instruções de transferência de bloco

Uma simulação	命令動作	código de operação	operação de sinalizadores	tamanho HP,y	N c 76 543 210 Hex B c
mover [.hl++] [.de++] y.hly;.bc-1;	yhvejyv.hlj;	00 1 0 • 1110	1 101 ED 2 4 10 100 000		
cidade amor y"de-1 .hl y.hl-1;.bc y.bc-1 repetir [.de y	[.de y][.hl] ; .de y.de-1 [.hl y	AO 0 0 1 0 • 1	1 1 101 101 ED 2 4 10 101000 A8		
mover +.hl y.hl+1;.bc y.be até .bc=0 repetir [.de y.hl] ;.de y.de-1	y.hlj;.de y.de+1 [.hl y] ;.de y.de+1	00 0 0 • 1 1 10	1 101 101 ED 2 4 1 0 110000		
mover [.hl y]".de-1 .hl y.hl-1;.bc y.bc-1 até .bc=0;	[.hl y]".de-1 .hl y.hl-1;.bc y.bc-1	BO 0 0 0 • 1	1 1 101 101 ED 2 4 10111000 B8		

\*1.Quando bc-1=0 O , Isso {você 1

A.7 Instruções de pesquisa de bloco

1 Mook	命令動作	código de operação	operação de sinalizadores	tamanho HP,y	N c 76 543 210 Hex B c
cmp .a[.hl++] .be y.be 1	.a[.hstop.hly.hl+1 [.hl] ;.hl y.hl-1	1 1 1 1 1 • 11	101 101 ED 2 4 10 100 001 A1		
cmp .a, [.hl y] c .bc-1 repetir .a-	[.hl] ;.hl y.hl-1	1 1 1 1 1 • 11	101 101 ED 2 4 10 101001 A9		
cm pm .a[.hl++] .be y.be ;t	[.hl] ;.hl y.hly]; l .bc=0 or .a=[.hl]	1 1 1 1 1 • 1 1	101 101 ED 2 5 10110 001 B1		
cm pm repetir .a[.hl] ;.hl y.hl-1 .a, [.hl-1] .be y.bc-1 até .bc=0 ou .a=[.hl]	repetir .a[.hl] ;.hl y.hl-1 .a, [.hl-1] .be y.bc-1 até .bc=0 ou .a=[.hl]	1 1 1 1 1 • 1 1	101 101 ED 2 5 10111001 B9		

Quando \*1.bc-1=0 O , Outros 1

Quando \*2.a=[.hl] 1 , Outros 0

A.8 Instruções de multiplicação

Mnemônico	命令動作	flags		オペコード		Hex	B	C
		S	Z H P <sub>N</sub> C	76543210				
mulub .a,r l.hly.a cidade .a*r	hl;.a*r	0	↑ • 0 • ↑	11 101 101	ED	2	14	
muluw .hl,ss l.de: .hl;.hl;.hl*ss	hl;.hl;.hl*ss	0	↑ • 0 • ↑	11 101 101	ED	2	36	

Com mulub, a operação não é garantida a menos que r seja .b, .c, .d, .e.

No muluw, a operação não é garantida exceto quando SS é be ySp.

## A.9 Instrução de adição

Uma simulação	命令動作 operação de comando	código de operação	ação de sinalizadores	sz	H	nota	N	c	76543210	Hex	BC
adicione .a,r	.aÿ.a+r .aÿ.a+p	1 1 1 v 0	10.000 rublos								1 1
adicione .a,p		1 1 1 1 v 0	11011 101 10.000p		DD	2	2				
adicione .a,q	.aÿ.a+q	!!!	11 111 101 10.000 q		FD	2	2				
adicionar .a,[hl] .aÿ.a+[hl] adicionar .a, [.ix+d] aÿ.a+[.ix+d]	.a←.a+[.ix+d]	!VÓ!	10.000 110		86	1	2				
adicione .a,[iy+d] .aÿ.a+[iy+d]		!!!	11011 101 10.000 110 ÿ d ÿ		DD	3	5				
adicione .a,[iy+d] .aÿ.a+[iy+d]		!!!	11 111 101 10.000 110 ÿ d ÿ		FD	3	5				
adicione .a,n	aÿ.a+n	!VÓ!	11000110 ÿÿÿÿ		C6	2	2				
addc.a,r	aÿ.a+r1C	!VÓ!	10.001 rublos								1 1
addc.a,p	aÿ.a+p+c	!!!	011 101 10 001p		DD	2	2	11			
addc .a,q	aÿÿ.a+q+c	!!!	11111 101 10001q		FD	2	2				
addc .a,[hl] aÿ.a+[hl]+c [.ix+d] aÿ.a+[.ix+d]+c	.a←.a+[.ix+d]+c	!VOCÊ!	10 001 110		8E	1	2				
addc .a,[ix+d] aÿ.a+[ix+d]+c		!!!	011101 10001 110 ÿ d ÿ		DD	3	5	11			
addc .a,[iy+d] aÿ.a+[iy+d]+c		!!!	11 111 101 10001 110 ÿ d ÿ		FD	3	5				
addc.a,n	.aÿ.a+n+c	!VÓ!	11001 110 ÿÿÿ+		CE	2	2				
addc.hl,ss	.hlÿ.hl+ss+c	1 eu estava v 0 1	101101 01 SS1010		ED	2	2	11			
adicionar .hl,ss	.hlÿ.hl+ss .lXÿ.1x+pp	**7*0 1 ** ?	00S 1001								1 1
adicionar .ix,pp			11011 101 00 pp 1 001		DD	2	2				
adicione .iy,rr	.iyÿ.1y+rr	••?•0 1	111 11101 00n - 1 001		FD	2	2				

二一 zombbar	命令動作	bandeiras	sopeco	一卜		
	rŷr+l	III v o •00 r 100				1 1
aumentar inc p	pŷp+l	II lv o •II OII 101 DD 2 2	00h 100			
inc q	qŷq+l	II lv o •111 1	1 101 FD 2 2			
inc[.hl]	[.hlŷr+.hl]+1	II lv o•00 110	100 34 1 4			
inc[ix+d]	[ix+dŷŷ[ix+d]+1	III v o •1101 1	1 101 DD 3 7 00 I IO			
inc[iy+d]	[iy+dŷŷ[iy+d]+1	II lv o •II III 101	FD 3 7 00 IIO 100			
incss		1 . ŷŷ ŷŷ ŷŷ ŷŷ 00	ss			1
inc.ix	SSŷss+l .ixŷ.ix+l	OOII ŷ ŷŷ ŷŷ ŷŷ ŷŷ	II OII 101 DD 2 2 00 100 OII			
inc.iy	.iyŷ.iy+l	III 101 FD 2 2 00 IO	OOII 23			

	00	01	10	11
ss	.bc	.fl	.sp	.sp
PP	.bc	.del	.ix	.sp
rr	.bc	.del	.iy	.sp

	000	001	010	011	100	101	110	111
p					.ixh	.ixl		
q					.iyh	.iyl		

## A.10 Instrução de Subtração

11mônico	命令動作	código de operação sz H também	ação de sinalizadores N c 76543210 Hex		
sub.a,r	.aÿ.ar	1 1 1 V I J 111	10010r	BC	1 1
sub.a,p	.aÿ.ap	v 1 1	11011 101 10010p	DD	2 2
sub.a,q	.aÿ.aq	111 v. 1 1	11 111 101 10010q	FD	2 2
sub.a,[hl],aÿa-[hl] sub.a,[ix+d],aÿa-	.a←.a-[ix+d]	111 v 1 1	10010110	96	1 2
[ix+d].a,[ix+d].a←.a-[ix+d]		111 v 1 1	11011101 10010110 ÿ d ÿ	DD	3 5 96
sub.a,[iy+d],aÿa-[iy+d]	.a←.a-[iy+d]	111 v. 1 1	11111101 10010110 ÿ d ÿ	FD	3 5 96
sub.a,n	.aÿ.an	111 v. 1 1	11010110 ÿÿÿÿ	D6	2 2
subc.a,r	.aÿ.arç	1 1 1 v 1	10011r		1 1
subc.a,p	.aÿ.apc	1 111 v 1 1	11011 101 10011p	DD	2 2
subc.a,q	.aÿ.aqc	111 v. 1 1	11 111 101 10011q	FD	2 2
subc.a,[hl],aÿa-[hl]c	.a←.a-[hl]c	111 v. 1 1	10011 110	9E	1 2
subc.a,[ix+d],aÿa-[ix+d]-	.a←.a-[ix+d]-	111 V I J	11011 101 10011 110 ÿ d ÿ	DD	3 5 9E
subc.a,[iy+d],aÿa-[iy+d]-c	.a←.a-[iy+d]-c	1 1 1 V eu J	11 111 101 10011 110 ÿ d ÿ	FD	3 5 9E
subc.a,n	.aÿ.anc	111 v. 1 1	11011 110 ←← n 11←	DE	2 2
subc.hl,ss	.hlÿ.hl-ss-c	JJ?	101 101 Ol ss ó oio	ED	2 2 1 1

Um Mokka	命令動作	bandeiras	peco	コード			
		sz H nota N c	76543210	Hex BC	1	1	1
cervo	rŷr-1	III vi•00	r 101				
dee p	pŷp-1	III vi•11 011 101	DD 2 2				
			00 pág . 101				
decq	qŷq- 1	III v 1 . 11 111	101 FD 2 2				
			00q 101				
dec [hl] dezero [hl]	[hl]ŷ [hl]-1	III v 1 •001101	01 35 1 4				
dec [ix+d] dezero [ix+d]	[ix+d]ŷ [ix+d]-1	III v 1 •11011	101 DD 3 7	00110101	35		
			ŷ d ŷ				
dec [iy+d] dezero [iy+d]	[iy+d]ŷ [iy+d]-	III vi •11111 101	FD 3 7 00110101	35			
			ŷ d ŷ				
decss	SSŷss- 1 ŷ	- •••••OO	ss tudo bem			1	1
dee.ix	Menos deŷ.1x-l	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	ŷŷ 11011 101 DD 2 2	00 101011 2B			
dee.iy	.iyŷ.iy-1	ŷŷ ŷŷ ŷŷ ŷŷ ŷŷ	ŷŷ 11111 101 FD 2 2	00101011 2B			

## A.11 Instruções de comparação

二一 zombar	命令動作	señalizadores de código de operação	HP, yN	c 76543210	Hex B	C
		tamanho				
cmp .a,r	.ar	III VI!	10111r			c 1 1
cmp .a,p	.ap	III VI!	1101 1 101	DD 2 2		
			10111p			
cmp.a,q	.aq	111 v. 1 1	111 101	FD 2 2 11		
			10111q			
cmp .a, [hl] cmp .a, [hl] ix+ dj .a [ix+d]	.a- [hl] .a- [ix+d]	1 1 1 v 1 1	10111 110	SER 1 2		
		111 v 1 1	11011101	DD3 5		
			10111 110	SER		
			ŷ d ŷ			
cmp .a, [iy+d] cmp .a, [iy+d]	.a- [iy+d]	111 v. 1 1	11111101	FD 3 5		
			10111 110	SER		
			ŷ d ŷ			
cmp.a,n	.um	IIIVII	11 111 110	FE 2 2		
			ŷŷŷŷŷ			

## A.12 Instruções lógicas de operação

11Mook	命令動作 operação de comando	código de operação		posição de sinalizadores	
		sz H PA,r N c	76 543 210	Hex	B C
e .a,r	.aÿ1.a8r	JJ I POO	1 1 1	1 0 100 rublos	1 1
e .a,p	.aÿ.a/p	p 0 0		haha, haha, haha 1 0 100p	DD 2 2
e .a,q	.aÿ.aÿq	1 1 1 p 0 0		101 1 0 100 q	FD 2 2 III
e .a,[hl].aÿ.aÿ[hl]	.a←.a^[.hl]	JJ I POO		1 0 100 1 10	A6 1 2
e .a,[ix+d].aÿ.aÿ[ix+d]	.a←.a^[ix+d]	JJ I POO		tudo bem, tudo bem 1 0 100 110 ÿdÿ	DD 3 5 A6
e .a,[ts+r+d].aÿ.aÿ[ty+d]	.a←.a^[iy+d]	1 1 1 p 0 0		eu vou 101 1 0 100 110 ÿdÿ	FD 3 5 A6
e .a,n	.aÿÿ.a8n	JJ I POO		1 1 100 110 ÿÿÿ*	E6 2 2
ou .a,r	.aÿÿ.aVr	JJ O POO		1 0 110 rublos	1 1
ou .a,p	.aÿÿ.aVp	JJO POO		1101 1 101 1 0 1 10 p	DD 2 2
ou .a,q	.aÿ.aVq	JJO POO		1 1 111 101 1 0 1 10 q	FD 2 2
ou .a,[hl].aÿ.aÿ[hl]	.a←.aV^[hl]	JJ O POO 1 1		1 0 110 1 10	B6 1 2
ou .a,[ix+d].aÿ.aÿ[ix+d]	.a←.aV^[ix+d]	0 p 0 0		1101 1 101 1 0 1 10 1 10 ÿdÿ	DD 3 5 B6
ou .a,[iy+d].aÿ.aÿ[iy+d]	.aÿ.aV^[iy+d]	1 1 0 p 0 0		1 1 111 101 1 0 1 10 110 ÿdÿ	FD 3 5 B6
ou .a,n	.aÿÿ.aVn	119 POO		1 1 110 110 ++nÿ	F6 2 2
xor .a,r	.aÿÿ.aÿr	1 1 0 p 0 0		1 0 101 r	1 1
xor .a,p	.aÿÿ.aÿp	JJ O POO		1101 1 101 1 0 101 p	DD 2 2
xor .a,q	.aÿÿ.aÿq	1 1 0 p 0 0		1111 1 1 01 10 101 q	FD 2 2
xor .a,[hl].aÿ.aÿ[hl]	.a←.aV^[hl]	1 1 0 p 0 0		10 101 110	EA 1 2
xor .a,[ix+d].aÿ.aÿ[ix+d]	.a←.aV^[ix+d]	JJ O PO O		1101 1 101 1 0 101 110 ÿdÿ	DD 3 5 A.E.
xor .a,[iy+d].aÿ.aÿ[iy+d]	.aÿ.aV^[iy+d]	1 1 0 p 0 0		1 1 111 101 1 0 101 110 ÿdÿ	FD 3 5 AE
xor .a,n	.aÿÿ.aÿn	JJO POO		1 1 101 1 10 Conflito - n -ÿ	EEE 2 2

## A. Instruções de operação de 13 bits

Uma simulação	operação	sinalizadores de código de operação	sz HP, v N c	76 543 210 Hex
pouco b,r	zÿNÃO r(bJ	BC 1 1 0•11	00 1 011 CB 2 2	01 b r
bit b <sub>i</sub> [h] zÿNOT [h]1bJ [h] (b)		?1 1wa 0•11	001 011 CB 2 3	01 b IIO
bit b <sub>i</sub> [ix+d] zÿNÃO [ix+d](bJ [ix+d] (b)		?1 1 ?0 •110	1 1 101 DD 4 5	11001011 CB ÿdÿ 01 b IIO
bit b <sub>i</sub> [iy+d] zÿNOT [iy+d] (b) [iy+d]		?1 1 ?0 •111	11101 FD 4 5	11001 011 CB ÿdÿ 01 b IIO
definir b,r	f(b)ÿÿl	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001011 CB 2 2	11 r
conjunto b [hl]	[hl](b)ÿÿ1-1	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001 011 CB 2 5	11bIIO
definir b <sub>i</sub> [ix+d] dj [ix+d](b)ÿÿ1-1		ÿ•••1101 1 101	DD 4 7	11001011 CB ÿdÿ
conjunto b [iy+d] [iy+d](b)ÿÿ1-1		11bIIO ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ eu vou eu OI FD 4 7	11001011 CB ÿdÿ 11 b eu eu O
clr b,r	f(b)ÿ0	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001011 CB 2 2	10h
clr b <sub>i</sub> [hl]	[hl](b)ÿ0-0	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001011 CB 2 5	10 b IIO
clr b <sub>i</sub> [ix+d] [ix+d](b)ÿ0-0		ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11011101 DD 4 7	11001011 CB ÿdÿ 10 b IIO
clr b <sub>i</sub> [iy+d] [iy+d](b)ÿ0-0		ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 111 101 FD 4 7	11001011 CB ÿdÿ 10 b IIO







## A.16 Instruções da filial

mnemônico	命令動作	código de operação de sinalizadores			
		s z H também	N c 76 543 210 Hex	BC	
br nn	.pcým	yy yy yy yy yy	yy 11000 011 C8 3 3		
			ý nn ýý yy nn h yy		
bnz nn	se Z=0 .pcým	••••••••1100001	0 C2 3 3		
			osso um nn1- + yy nn h yy		
bz nn	se Z=1 .pcýnn	yy yy yy yy yy	yy 1100 1 010 CA 3 3		
			yy nn1 yy < -nnh ->		
bnc nn	se c=0 .pcýnn	yy yy yy yy yy	yy 1 1 010010 D2 3 3		
			< -nn1 -ý yy nn h yy		
ser nn	se c=1 .pcýnn	•yy yy 1 1 01	1 010 DA 3 3		
			yy nn1 yy yy nn h yy		
bpo nn	se .conta=0 pcým	yy yy yy yy yy	yy 1 1 100 010 E2 3 3		
			- nn1 -ý yy nnh yy		
bpe nn	se pt...=1 .pcýnn	yy yy yy yy yy	yy 1 1 101 010 EA 3 3		
			< -nnýý +- nnhý		
bp nn	se s=0 .pcýnn	yy yy yy yy yy	yy 1 1 110010 F2 3 3		
			Oso - nn1 umý ý nnhý ý		
bm nn	se S = eu .pcým	•yy yy 1 1 11	010 FA 3 3		
			yy nn t yy yy nnh yy		
br [.hl]	.pcýy.h] lpcý	yy yy yy yy yy	yy 1 1 10 1 001 E9 1 1		
br [.ix]	ý.ix]	yy yy yy yy yy	yy 11011101 DD 2 2		
			1 1 10 1 001 E9		
br [.iy]	.pcýy.ty]	yy yy yy yy yy	yy 11111101 FD 2 2		
			1 1 101 001 E9		

11Mook -	命令動作	código de operação	ação de sinalizadores		
			s z H também	N c 76 543	210 Hex
curto br e	.pcÿ.pc+e	B c ÿ ÿÿ ÿÿ ÿÿ	ÿÿ ÿÿ 00011	000	18 2 3
curto bnz e	se Z=O .pcÿ.pc+e	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 00 100	000 20	2 2
curto bz e	se Z=l .pcÿ.pc+e	3 . ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 00 101	000 28	2
curto bnc e	se c=O .pcÿ.pc+e	2 3 ÿ ÿÿ ÿÿ ÿÿ	ÿÿ ÿÿ 00 11	0000 30	2 2
curto be e	se	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 00 111	000 38	2 2
dbnz e	c=l .pcÿ.pc+e .bÿ.b- l;se .b:f=O .pcÿ.pc+e	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 00010	000 10	2 2

## A.17 Instrução de Chamada

二一 zomb ar	命令動作	bandeirasopeo	一ド		
			s z H Nove c 76	543 210	Hex BC
ligar	[.sp-2]ÿ.pci [.sp-1]ÿ.pci .spÿ .sp .pc h 2;pcÿnn -	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001	101 CD	3 5
ligue paranz,nn	se z = O [.sp-2]ÿ.pc1 ; [.sp-l]ÿ.pCh .spÿ.sp-2;	•ÿÿÿÿÿ 11000	00 C4 3 3 5		
liguez,nn	[.sp-2]ÿ.pc1 ; [.sp-l]ÿ.pCh .spÿ ÿ.sp-2;pcÿÿnn se c=O	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 11001	100cc 3 3	5
ligar para n,nn	[.sp-2]ÿ.pci ; [.sp 1]ÿ.pctr.sp ÿ.sp-2;	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 010	100 D4 3 3	5
ligue, nn	[.sp-2]ÿ.pc1; [.sp 1]ÿ.pctr.Sp ÿ .sp-2; .pch .spÿ.sp-2; .pcÿm	•••1101 1 100	DC 3 3 5		
callpo,nn	se p.<,- =1 [.sp-2]ÿ.pc1 ; .pcÿm se s=O	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 100	100 E4 3 3	5
callpe,nn		ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 101	100 CE 3 3	5
ligar,nn	[.sp-2]ÿ.pc1 ; [.sp-l]ÿ.pch .sp ÿ.sp-2 ; ]ÿ.pch .spÿ.sp 2;	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 1 10	100 F4 3 3	5
ligue, nn	-	ÿÿ ÿÿ ÿÿ ÿÿ ÿÿ	ÿÿ 1 1 1 11	100 FC 3 3	5

二一 五 bar	命令動作	bandeiras	オペコード		
		s z H vergonha Nc 76 543	210 Hex B		
ret	d:pc1y1.sp1.pch1/sp+1;.sp1.sp+2 y y y y y y y y 11 001 001 C9	1 3 y y y y y y y y			
ret nz	se Z= O .pc1y1.sp1;.pch y y.sp+ 1]; .sp y.sp+2 se	1 1 000 000 co 1 1 3 y y y y y y y y			
ret z	c=O .pc1y1.sp1;.pch y y.sp+ 1]; .sp y.sp+2 se	11001 000 C8 1 1 3 y y y y y y y y	1 1 010000		
ret nc	C=l .pc1y1.sp1;.pch y y.sp+ 1]; .sp y.sp+2 se	DO 1 1 3 y y y y y y y y	11011000		
ret c	vergonha=0 .pc1y1.sp1;.pch y y.sp+ 1]; .sp y.sp+2	D8 1 1 3 y y y y y y y y	1 1 100000		
ret po	se mo=1 .Pct y [ .sp]; .pch y [ .sp+ 1];	EO 1 1 3 y y y y y y y y	1 1 101		
ret pe		000 E8 1 1 3 y y y y y y y y	11110 000 FO 1		1
ret p	se s=O .pct y y.sp1;.pch y y.sp+ 1];	3 y y y y y y y y y y y y 11111000	F8 1 1		
ret m	se S = eu .pc1y1.sp1;.pch y y.sp+ 1];	3 y y y y y y y y y y y y 1 1 101	101 ED 2		
reti		5 01 001 101	Retorno de interrupção não mascarável		
retn	4D y y y y y y y y y y 1 1 101 101 ED 2 5 01 000 101 45				
kkk	[.sp 2]y.pct1;.sp1;.pch y y.sp+ 2];	y y y y y y y y k/8 111			1 4

## A.18 Instruções de entrada/saída

二一 zōmbar	命令動作	sinalizadores de código de operação sz H também N c 76 543 210 Hex BC	código de operação		
			banco de dados	2	3
* :amf]	:ayyif]	99 99 99 99 99	11011011	banco de dados	2 3
* senhor f, [c]	rÿy.c]	1 1 0 p 0 •	1 1 1 0 1 1 0 1 0 1 r 0 0 0	ED 2 3	
* f, [c]	[.c]	1 1 0 p 0 •	1 1 1 0 1 1 0 1 0 1 1 1 0 0 0 0	DE 2 3 70	
* eu [.hl++J, [.c].hlÿ.hl+l .hlÿ	[.hl+ [.c]; b←.b-1	? 1 ?	1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0	ED 2 4 A2	
* eu [.c].hlÿ.hl+l 1; repetir:	[.c].by.b-1 [.hl+ [.c]; b←.b-1	? 1 ? • 1	1 1 1 0 1 1 0 1 1 0 1 0 1 0	ED 2 4 AA10	
* Liffi [.c].hlÿ.hl+l até .b=O repetir [.hlÿ	repetir [.hl+ [.c]; b←.b-1	? 1 ?	1 0 1 1 0 1 1 0 1 1 0 0 1 0	ED 2 4 1 1 B23	
* l; até .b=O nÿ.a out ln J .a	repetir [.hl+ [.c]; b←.b-1	? 1 ?	1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 0	ED 2 4 BA3	
* [n], a	[n]←.a	9999999999999999	1 1 1 0 1 0 1 1 9999999999999999	D3 2 3	
* fora l.c; f, r	[.c]←r	99 99 99 99 99 99	1 1 1 0 1 1 0 1 0 1 r 0 0 1	ED 2 3	
* fora [.c], [.hl++].hlÿ.hl+l [.c]ÿl	[.c] [mesmo, by, b-1	? 1 ? • 1	1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 1	ED 2 4 A3	
* fora [.hl--].hlÿ.hl-1	hlÿ, by, b-1 [.c],	? 1 ? • 1	1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 1	ED 2 4 AB	
* fora [.hl++].hlÿ.hl+l até .b=O repetir [.c]ÿl.	repetir [.c]ÿl; by, b-1 [.c],	? 1 ?	1 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1	ED 2 4 B3 3	
* fora l; até .b=O	hlÿ, by, b-1 [.c], [.hl--].hlÿ.hl-1	Wa 1 ?	1 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1	ED 2 4 BB3	

\*Quando 1.b-1=0, 1 , Outros são 0

em .f,[c] simplesmente muda o sinalizador dependendo do conteúdo da porta indicada pelo registro .c.

O conteúdo de não pode ser armazenado em lugar nenhum



# Índice

## difficuldade

Porta 1/0. 57

• Tempo de acesso . . . . . 48

Endereço. 48

Adresnow . . . . . 48

111

• Sinterlace • Função de

111

SECAM . . . . . 111

SRAM. . . . . 50

NTSC . . . . . 110

BIOS FM . . . . . 139

MSX-Engine • 17 MSX-JE

72

MSX-MÚSICA . . . . .

Tipos de MSX . . . . . 62

Envelope 0... 133

OPLL YM2413 . . . . .

OPLL Dryno f. 156

Operador - . . . . . 131

ruido de escala . . . . . 133

## caixa

MSX feito para exportação no exterior . . . . .

62





OM.....	52
Emoji — .....	50
マッパ — .....	113
Yu Ré — 夜 — . Abra ..de — .....	137
vezes	
Função USR.....	119
país	
BATER .....	50
som de ritmo .....	137
Redefinir status .....	
Conversão de cláusula contínua .....	72
ROM .....	50

## Referências

- [1] Sede ASCII Microsoft FE, Nippon Gakki Mfg. Co., Ltd., “V9938 MSX-VÍDEO LIVRO DE DADOS TÉCNICOS”、アスキー、1985年(絶版)
- [2] ASCII Co., Ltd., “Documento de construção V9958”、非売品、1988年
- [3] Supervisionado por ASCII Microsoft FE, “1SX2 Technical Handbook”、ASCII, 1986
- [4] Seiichi Sugitani, “Como fazer uso completo de h-sei sx2 + no f waffle”、アスキー、1989年
- [5] Asuki Mori Co., Ltd., “RanrISX-Datapack”、アスキー、1991年

•Biografia do autor

Ishikawana., ta

Naota Ishikawa

横浜国立大学卒業後、アスキーに入社、MSXの開発に携わる。その後、東京理科大学理学部第二部数学課、

Graduado no programa de mestrado da Divisão de Pós-Graduação em Ciências. Atualmente com Sogijuku

Certificação da Escola de Pós-Graduação em Pesquisa em Ciências e Engenharia, e atualmente está matriculado em um curso de doutorado.

MSX <O artigo da Technica J-Res serializado na revista

Eu também sou o autor. [naota@slab](mailto:naota@slab).

## Manual Técnico MSX turbo R

Publicado pela primeira vez em 31 de julho de 1991

Folha normal 2 , 500 liness (unidade principal 2 , 427 liness)

Autor Ishikawa |

Editor Keiichiro Tsukamoto

Editor Fumitaka Kojima

Editora : **Empresa ASCII**

3F Edifício Minami Aoyama, 6-11-1 Minami Aoyama, Minato-ku, Tóquio 107-24

Transferir Tóquio 4-161144

Representante (03) 3486-7111

Departamento de Vendas da Editora (03) 3486-1977 (Diamante/Rain)

Este livro é protegido pela lei de direitos autorais. Parte ou todo o edifício principal

Em relação (incluindo software e programas), ASCII Co., Ltd.

Copiar sem permissão de qualquer maneira sem permissão por escrito da

A reprodução é proibida.

Produzido por Tokyo Shoseki Printing Co., Ltd.

Impressão Tokyo Shoseki Printing Co., Ltd.

---

Editado pelo Departamento Editorial da MSX Magazine

ISBN 4-7561-0621-8

Impresso no Japão



### 内容一覧

- MSX turbo R のシステム構成
- R800 のプログラミング・テクニック
- BASIC コマンドの追加、削除、変更一覧
- 拡張 BIOS の追加、削除、変更一覧
- PCM 録音、再生機能活用法
- MSX2+と MSX turbo R のスロット
- 漢字 BASIC 概要
- V9958VDP 仕様書
- 自然画表示モード詳細
- 走査線割り込みの実践
- MSX-MUSIC のコントロール
- BASIC と BIOS の音色データ一覧
- R800CPU インストラクションコード表

boca Boca

boca

1 Boca de punição!

boca • ÿy; boca •

