

# Nestor, a base de todos os computadores



Este é um micro feito para montar e aprender a "mexer" com os computadores. Além disso, é um dos modelos mais baratos já projetados no Brasil

**L**onge de ser "mais um" computador pessoal, com terminal de vídeo, unidade de disquete ou fita cassete, linguagem de alto nível, impressora e teclado alfanumérico, o Nestor pode ser melhor definido como um micro experimental, dirigido a estudantes. De fato, ele foi concebido de forma a ocupar uma única placa de 23x18 cm, que abriga todos os seus componentes, desde o teclado compacto até o visor de LEDs.

Todos esses componentes passaram por uma pesquisa de mercado, a fim de assegurar sua fácil obtenção e também que o projeto fosse o mais barato possível. Assim, por exemplo, optamos por utilizar o tradicional microprocessador Z-80A — que acha-

mos ainda "em forma" e, o que é melhor, facilmente encontrável nas lojas, a um custo relativamente baixo.

O mesmo processo norteou a escolha das demais peças; o teclado, por exemplo, foi totalmente composto com pequenas teclas individuais, de fabricação nacional, bastante confiáveis e baratas.

O Nestor trabalha em linguagem de máquina, com o auxílio de sete funções básicas (veja a tabela de características). Pode ser empregado como um pequeno sistema de desenvolvimento, ou como um pequeno controlador de processos industriais (ele dispõe de saídas para interfaces de potência), ou ainda como ferramenta didática, no ensino de hardware e software de microcomputadores.

## Dividindo o assunto

Dada a extensão da matéria, ela será dividida, a princípio, em três partes. Nesta primeira, vamos tratar da descrição do hardware adotado, em nível básico, sem entrarmos em muitos detalhes, já que os montadores poderão se aprofundar nessa parte consultando a bibliografia que fornecemos no final do artigo.

Na edição de março, publicaremos o projeto da placa de circuito impresso, detalhes de montagem, seqüência para teste de circuitos, incluindo uma série de macetes práticos. Por fim, em abril, vamos apresentar toda a lista de programas monitor, com seu respectivo fluxograma, além do mapeamento de memórias, formatação

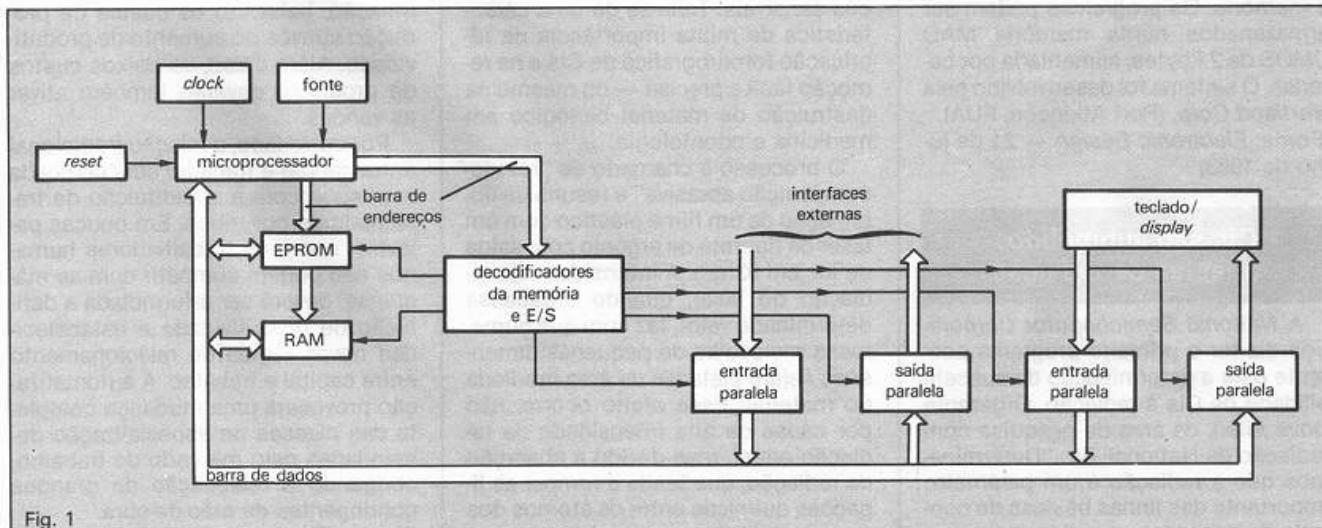


Fig. 1

Diagrama de blocos completo do Nestor.

de caracteres para o *display*, mapeamento de portas, alguns programas como sugestão e ainda uma tabela contendo todo o conjunto de instruções do microprocessador Z-80A. E já que falamos nele, vamos começar a primeira parte dando uma visão geral desse componente.

### O microprocessador Z-80A

Para entender melhor o projeto de hardware do Nestor (cujo diagrama de blocos aparece na figura 1), é essencial algum conhecimento sobre o microprocessador nele utilizado. Acompanhe a explanação pelo diagrama da estrutura interna desse integrado, que está na figura 2.

O Z-80 possui, internamente, 18 registradores de 8 bits e mais 4 de 16 bits, todos acessíveis ao programador e que funcionam como uma memória programável estática. Podem ser divididos da seguinte forma (figura 3):

**Registrador acumulador (A) e de flag (F)** — São independentes, sendo que o acumulador recebe os resultados de todas as operações lógicas e aritméticas de 8 bits, enquanto que o registrador de *flag* indica a ocorrência de condições específicas, tais como paridade, zero, sinal, transporte, excesso numérico (*overflow*).

**Registradores de aplicação geral** — Dividem-se em dois conjuntos idênticos. O principal contém 6 registradores de 8 bits, chamados B, C, D, E, H e L; o conjunto alternativo possui o mesmo número de registradores, com notação B', C', D', E', H' e L'. No caso de operação de 16 bits, tais registradores podem ser agrupados em pares (BC, DE, HL ou BC', DE', HL').

**Registradores de aplicação especial** — São apenas dois: o PC ou contador de programa e o SP ou indicador de pilha. O primeiro é de 16 bits e possui sempre o endereço no qual a instrução em curso será executada. Prosseguindo na execução, o PC é incrementado, caso o programa necessite do próximo byte na memória; caso contrário, o valor correto desse registrador é substituído por outro, se deve ser executada uma instrução de salto (*jump*) ou chamada (*call*).

Para permitir o uso de vários níveis de sub-rotinas, utiliza-se a chamada "pilha" e o correspondente **indicador de pilha**. Assim, sempre que determinadas instruções são executadas ou quando são feitas chamadas para

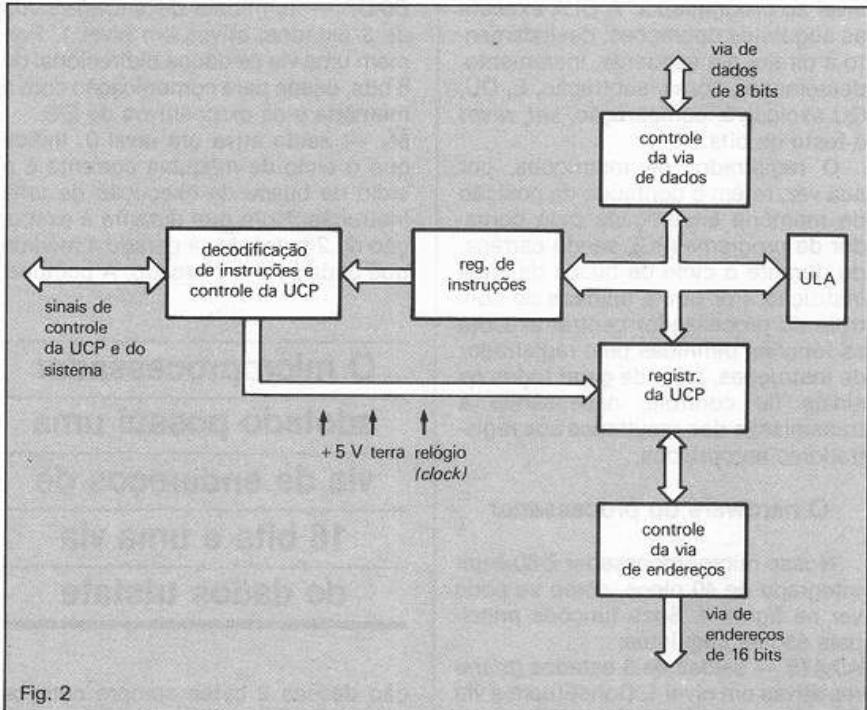


Fig. 2 Estrutura interna simplificada do microprocessador Z-80.

sub-rotinas, o registrador PC pode ser armazenado nessa pilha.

Na verdade, a "pilha" é uma área reservada de várias posições de memória, o topo da qual é "memorizado" pelo conteúdo do indicador de pilha; em outras palavras, o indicador mostra o endereço de entrada mais recente, já

tre si, permitindo transformar a base de endereços usada em modos de endereçamento indexado, além de apontar para as posições de memória onde o dado respectivo deve ser armazenado ou restaurado. Incorporado às instruções de indexação existe um número inteiro, em complemento de dois, que especifica o deslocamento do endereço básico.

**Registrador de endereço de página de interrupção (I)** — É um registrador de 8 bits, que pode ser carregado com um endereço da página de uma rotina de interrupção. Um programa de controle de interrupção irá votoriar esse endereço de página.

**Registrador de restauração de memória (R)** — Para permitir o uso de memórias dinâmicas com o Z-80, utilize-se este registrador de 7 bits para restaurar os dados; ele é automaticamente incrementado após cada ciclo de busca (*fetch cycle*).

O Z-80 dispõe ainda de uma unidade lógica e aritmética (ULA), um registrador de instruções e um controle do processador central. Todas as operações lógicas e aritméticas do processador são efetuadas pela ULA, 8 bits por vez; ela se comunica internamente com os registradores do processador central e não é diretamente aces-

## O Z-80 dispõe de um total de 22 registradores, sendo 18 de 8 bits e 4 de 16 bits

que as posições de memória são armazenadas como um arquivo do tipo LIFO (último a entrar, primeiro a sair). Ao término da sub-rotina, o processador central retoma o programa principal, através do endereço existente no topo da pilha.

**Indexadores IX e IY** — Facilitam a manipulação de tabelas de dados. São ambos de 16 bits e independentes en-

sível ao programador. A ULA executa as seguintes operações: deslocamento à direita e à esquerda, incremento, decremento, soma, subtração, E, OU, OU exclusivo, comparação, set, reset e teste de bits.

O registrador de instruções, por sua vez, retém o conteúdo da posição de memória endereçada pelo contador de programa (PC), sendo carregado durante o ciclo de busca de cada instrução. Por fim, a unidade de controle do processador central executa as funções definidas pelo registrador de instruções, além de gerar todos os sinais de controle, necessários à transmissão dos resultados aos registradores apropriados.

#### O hardware do processador

Nosso microprocessador Z-80 é um integrado de 40 pinos, como se pode ver na figura 4. Suas funções principais são as seguintes:

**A0-A15** — saídas de 3 estados (*tristate*), ativas em nível 1. Constituem a via de endereços de 16 bits e fornecem o endereço para as mudanças de dados na memória (até 64 kbytes) e nos dispositivos E/S. O endereçamento de entrada/saída pode utilizar somente os 8 bits menos significativos do endereço, podendo selecionar diretamente 256 dispositivos diferentes. Durante o tempo de restauração, os 7 bits menos significativos possuem um endereço válido de restauração.

**D0-D7** — terminais de entrada/saída de 3 estados, ativos em nível 1. Formam uma via de dados bidirecional de 8 bits, usada para comunicação com a memória e os dispositivos de E/S.

**M<sub>1</sub>** — saída ativa em nível 0. Indica que o ciclo de máquina corrente é o ciclo de busca da execução de uma instrução. Note que durante a execução de 2 bytes, M<sub>1</sub> é gerado à medida que cada byte é buscado. A codifica-

### O microprocessador adotado possui uma via de endereços de 16 bits e uma via de dados tristate

ção desses 2 bytes sempre começa com CBH, DDH, EDH ou FDH. A função M<sub>1</sub> é também utilizada em conjunto com IORQ, para indicar o reconhecimento de um ciclo de interrupção. **IORQ** — saída de 3 estados, ativa em nível 0. Indica que a metade menos significativa da via de endereços possui um endereço válido de E/S, para uma operação de leitura ou escrita em uma E/S. O sinal IORQ também é gerado com o sinal M<sub>1</sub> quando uma in-

terrupção está sendo reconhecida, a fim de indicar que um vetor de resposta de interrupção pode ser colocado na via de dados. A operação de reconhecimento de interrupção pode ocorrer durante um tempo de M<sub>1</sub>, enquanto as operações de E/S estão proibidas.

**MREQ** — saída de 3 estados, ativa em nível 0. O sinal de pedido de memória indica que a via de endereços possui um endereço válido para uma operação de leitura ou escrita na memória.

**RD** — saída de 3 estados, ativa em nível 0. Indica que o processador central deseja ler na memória ou um dispositivo de E/S; tanto a memória como o dispositivo endereçados devem usar esse sinal para colocar as informações na via de dados do processador central.

**WR** — saída de 3 estados, ativa em nível 0. Indica que a via de dados do processador central possui um dado válido para ser armazenado na memória ou no dispositivo de E/S endereçado.

**RFSH** — saída ativa em nível 0. Indica que os 7 bits menos significativos da via de endereços contém um endereço de restauração (*refresh*) para memórias dinâmicas e que o sinal MREQ deve ser usado para fazer uma leitura correspondente para todas as memórias.

**HALT** — saída ativa em nível 0. Indica que o processador central executou uma instrução HALT (parada) e está esperando uma interrupção, antes de reassumir a operação. Enquanto parado, o processador central executa a instrução NOPs (fora de operação), a fim de manter ativa a restauração de memória.

**WAIT** — entrada ativa em nível 0. Indica, para o processador central, que o dispositivo de E/S ou a memória endereçada não estão prontos para transferência de dados. O processador permanece no estado de espera enquanto o sinal WAIT está ativo. Ele permite, assim, a sincronização da memória ou do dispositivo de E/S com o processador central.

**INT** — entrada ativa em nível 0. Este sinal, de pedido de interrupção, é gerado pelos dispositivos de E/S. Cada pedido será atendido ao final da instrução em curso, caso o flip-flop de interrupção controlado por software esteja habilitado e o sinal BUSRQ não estiver ativo. Quando o processador central aceita a interrupção, um sinal de reconhecimento (IORQ, durante M<sub>1</sub>) é gerado, para dar início ao próximo ciclo de instrução. O processador

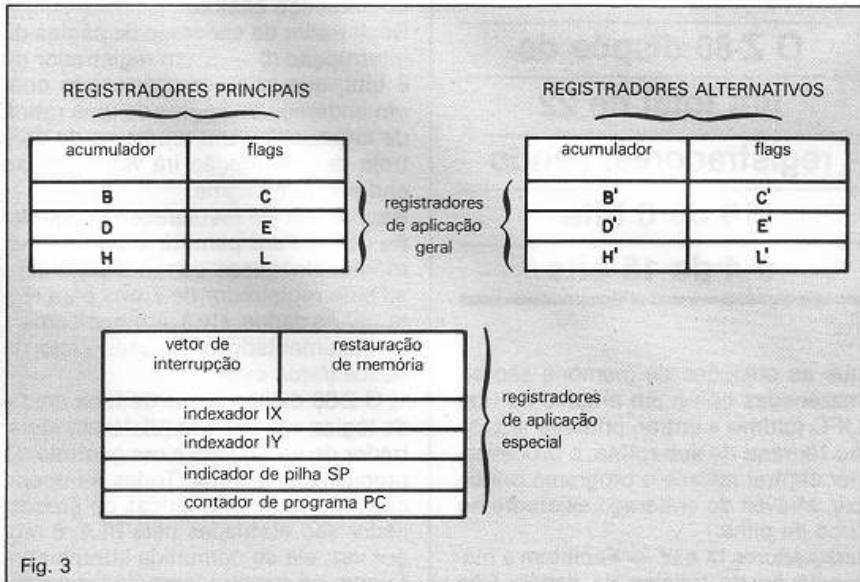


Fig. 3

Estrutura de registradores do processador central do Z-80.

pode responder a uma interrupção de três modos diferentes.

**NMI** — entrada gatilhada na descida do pulso. Esta linha, de pedido de interrupção não mascarável, tem prioridade superior à da INT e é sempre recolhida no final da instrução presente, independentemente do estado do flip-flop de interrupção. O sinal NMI força o processador a recomeçar da posição 0066H. O contador de programa é automaticamente preservado na pilha externa, para que o usuário possa retornar ao programa que foi interrompido. Note que ciclos contínuos de WAIT podem impedir o término da instrução em curso e que um BUSRQ tem o poder de anular um NMI.

#### Tipos de instruções do Z-80

O microprocessador Z-80 é capaz de executar 158 instruções, incluindo todas as 78 do 8080, que podem ser agrupadas da seguinte forma: carga e troca, aritmética e lógica, transferência de bloco e procura, rotação e deslocamento, manipulação de bits, salto, chamada e retorno, entrada e saída, controle de UCP. As instruções se-

rão melhor detalhadas na parte dedicada ao software.

#### O circuito do Nestor

Na figura 5 aparece o circuito completo de nosso micro. Para facilitar um pouco sua análise, dividimos esse esquema em partes, que podem ser melhor identificadas pelo diagrama de blocos da figura 1.

**Geração de relógio (clock)** — O Nestor utiliza um *clock* (sinal de sincronismo) de aproximadamente 3,58 MHz, ou seja, um pouco abaixo da frequência padrão do Z-80A, que é de 4 MHz. O circuito projetado é simplesmente um oscilador clássico a cristal, utilizando dois inversores na geração do sinal e um terceiro apenas para quadra esse sinal.

Em termos de operação, o cristal não precisa ser necessariamente o sugerido; aliás, o oscilador nem mesmo precisa ser a cristal, caso não exista preocupação com os tempos de processamento (outro assunto que veremos na parte de software). Nesse caso, o montador poderá optar pelo circuito da figura 6, que é mais simples e econômico.

**Círcuito de reset** — O *reset* é um dos controles mais necessários em microcomputadores, pois ele tem a função de parar o programa e carregar o contador de programa (PC) com 0000H (no mais baixo endereço de memória), permitindo ao usuário reiniciar seu programa.

O *reset* pode ser manual, automático ou combinado. Em nosso projeto,

**Nestor é um computador completo, com RAM, EPROM, teclado, display e clock a cristal**

utilizamos um circuito padrão de *reset*, com o qual pode ser dado um comando manual e também um *reset* automático para a UCP, quando todo o circuito é ligado, através da malha formada por R8 e C2. Assim que a alimentação é fornecida ao circuito, o capacitor — totalmente descarregado — levará um certo tempo para produzir um nível lógico 1 na entrada RESET da UCP; e, quando o micro é desligado, utiliza-se um diodo para descarregar o capacitor rapidamente. **Decodificação de E/S** — Lembrando que um endereço qualquer pode tanto referir-se à memória quanto a uma entrada/saída, é preciso diferenciar, entre essas funções, qual o dispositivo externo a ser acionado. Isto é perfeitamente possível, pois a UCP nos fornece, na sua via de controle, os sinais de MREQ (requisição de memória), IORQ (requisição de E/S), RD (pedido de leitura) e WR (pedido de escrita), todos citados no tópico referente ao Z-80.

Em nosso circuito, por exemplo, utilizamos inicialmente uma combinação NOU entre os sinais RD e IORQ (C116), gerando um sinal de leitura de entrada (IORD); empregamos a mesma combinação para WR e IORQ, produzindo um sinal para leitura de saída (IOWR). O primeiro é utilizado para selecionar os pontos de entrada e o segundo, os pontos de saída.

Como nosso circuito é muito simples, tendo apenas 3 portas de saída

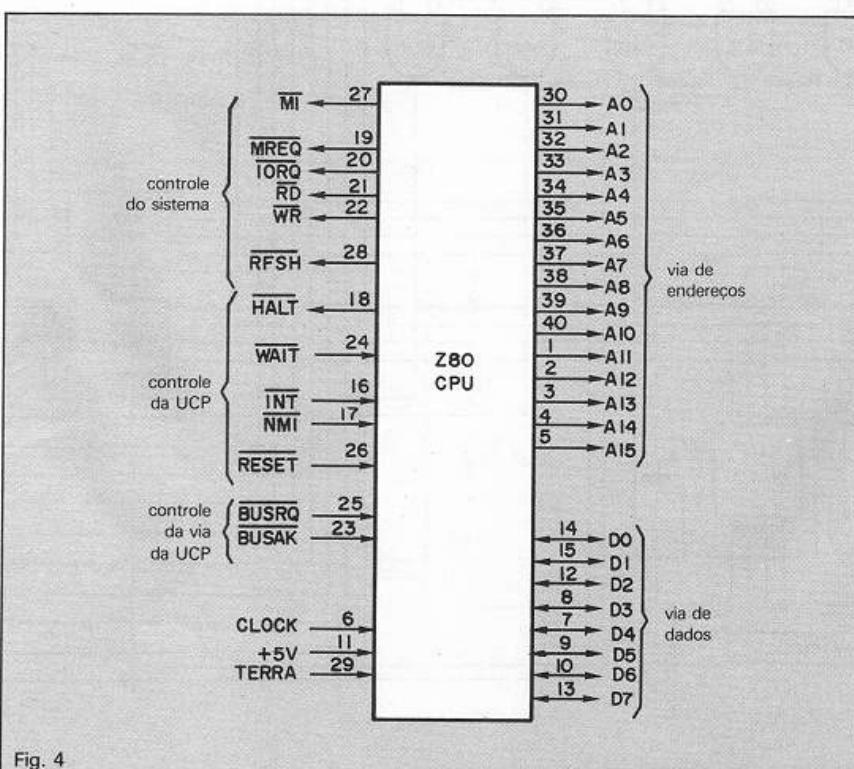


Fig. 4

Pinagem e funções do mesmo processador.

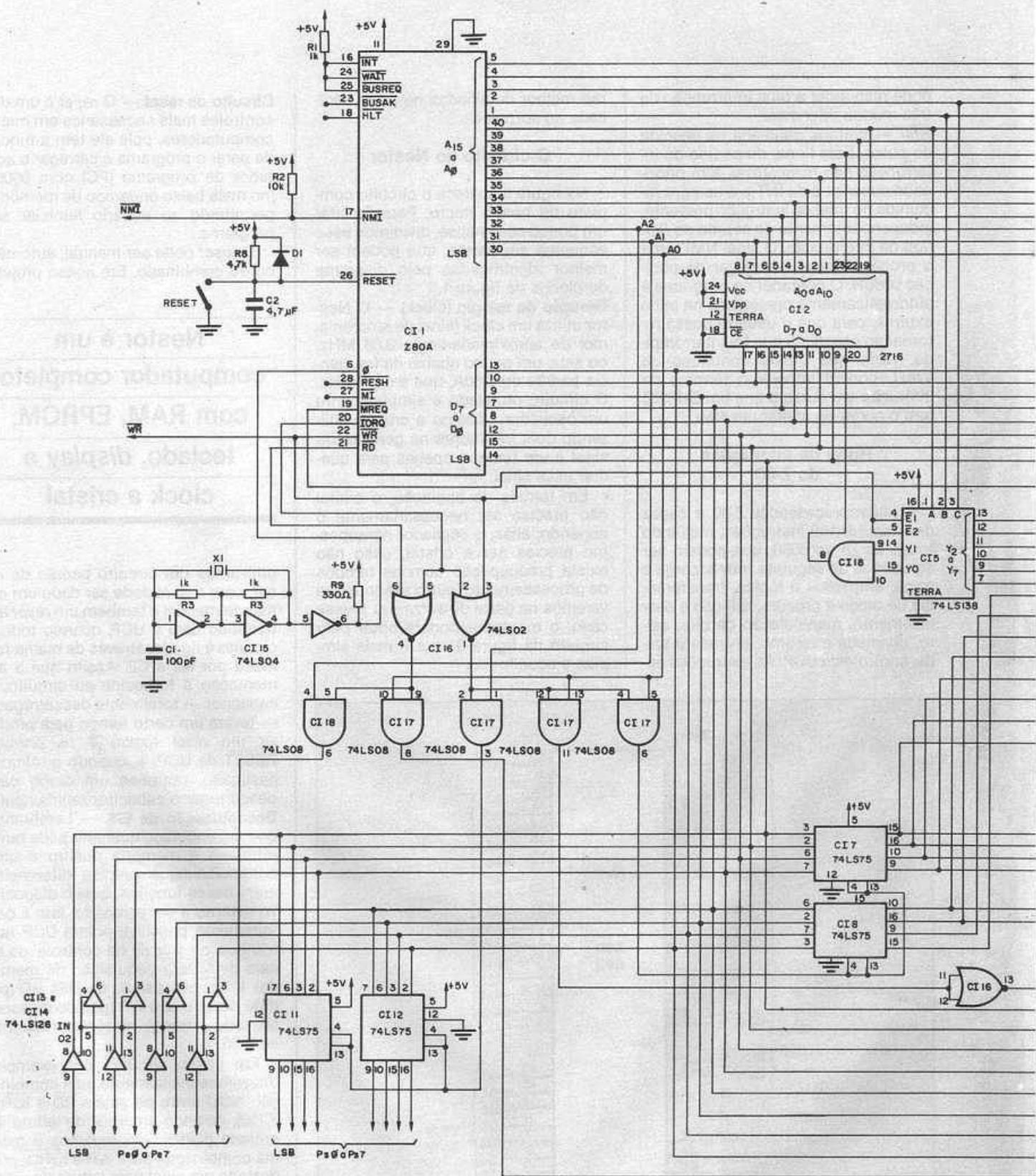
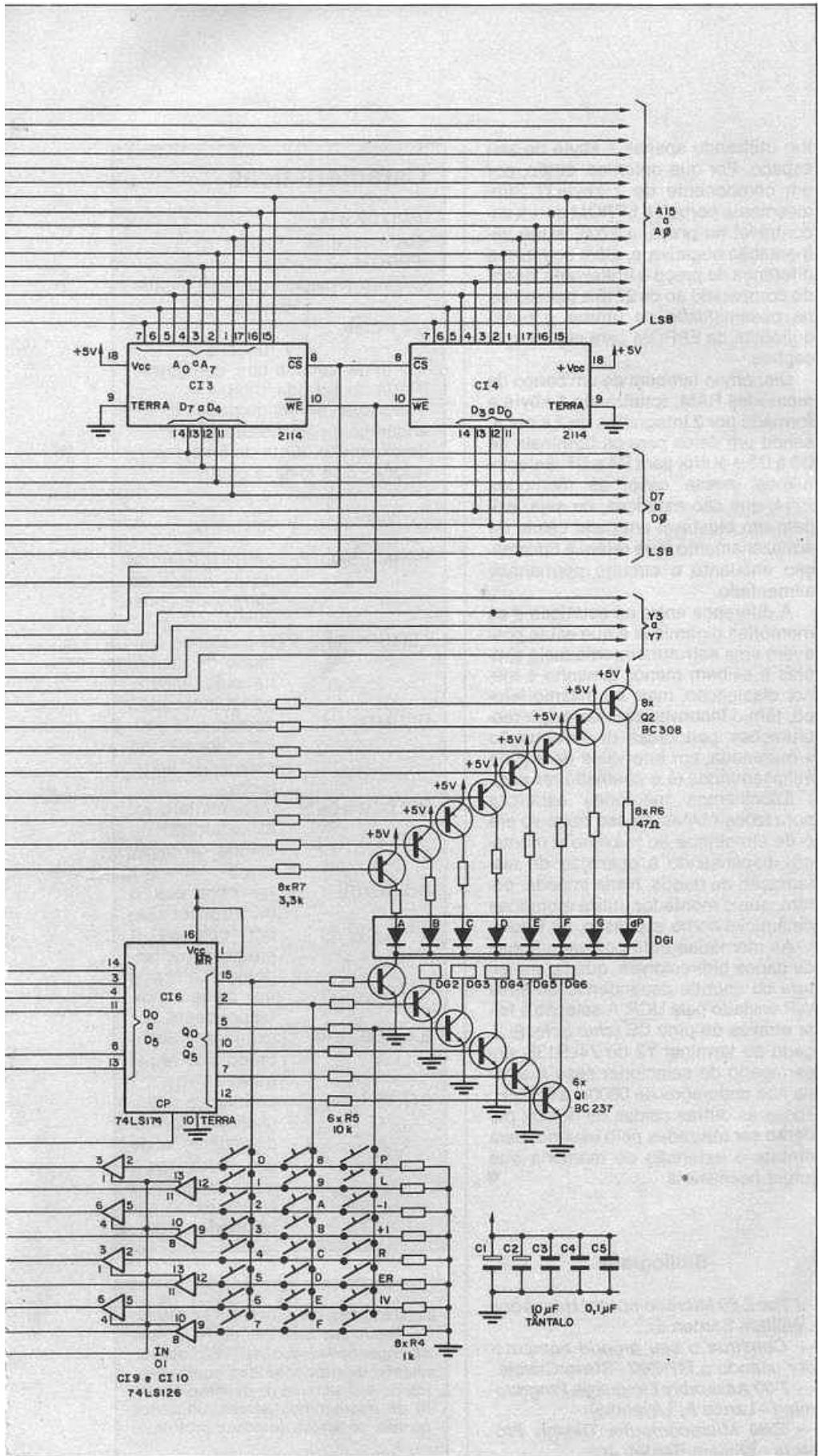


Fig. 5

*Esquema completo do Nestor, com todos os componentes e ligações.*



(sendo duas para o *display*) e 2 portas de entrada (das quais uma para o *display*), não foi necessária a utilização de decodificação para as linhas de endereço (A0 a A7) — as quais, devidamente decodificadas, podem endereçar até 256 portas diretamente.

Empregamos diretamente apenas as linhas de endereço A0, A1 e A2, combinadas com os sinais IORD e IOWR, de forma a termos os seguintes endereços para as portas: CI6 = 02H (saída), CI7 e CI8 = 01H (saída), CI9 e CI10 = 01H (entrada), CI11 e CI12 = 04H (saída), CI13 e CI14 = 02 (entrada).

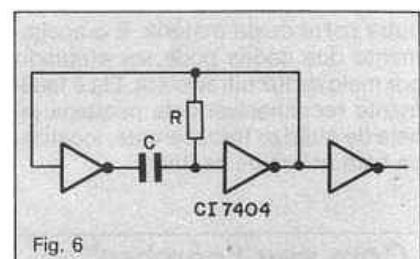


Fig. 6  
Sistema alternativo de clock para o Nestor, sem utilizar cristal.

Como todos devem saber, existem portas de E/S bastante sofisticadas, inclusive bidirecionais, como a PIO Z-80, por exemplo; existem, também, interfaces bastante poderosas para *displays* e teclados. Tais dispositivos, porém, chegaram a ter preços proibitivos nos últimos meses, especialmente para aqueles que desejam montar seu primeiro micro da forma mais econômica possível. Por essa razão, preferimos optar, como portas de saída, por simples biestáveis tipo D da família TTL LS e como portas de entrada, simples buffers tristate, da mesma família.

Nosso projeto possui ainda uma porta de saída de 8 bits com *latch* e uma porta *tristate* de 8 bits para entrada, a fim de proporcionar ligações externas para o micro, mas que não estão sendo utilizadas no sistema mínimo sugerido (CI11 a CI14).

As outras portas (duas de saída, uma de entrada), como se pode ver, são utilizadas para o conjunto *display/teclado*. A explicação sobre a interpretação de teclas e multiplexação do *display* será dada na parte referente ao programa monitor, já que tais funções são produzidas por *software*.

**Decodificação de memória** — A deco-

dificação necessária à seleção de memórias é efetuada por um demultiplexador 1 por 8 (74LS138), com suas entradas de seleção ligadas a A10, A11 e A12. Com as combinações resultantes, é possível endereçar 8 kbytes de memória, sendo 1 kbyte por canal. Note que o Nestor utiliza, em sua versão básica, apenas 3 kbytes de memória (2 k de EPROM e 1 k de RAM), permitindo, portanto, expansão de até 5 k.

No projeto usamos, no endereço 0000H-07FF, uma memória EPROM, do tipo que aceita apenas leitura, mas pode ser reprogramada pelo usuário. Sua gravação é feita através de uma interface especial, como veremos em outra parte desta matéria. E o apagamento dos dados pode ser efetuado por meio de luz ultravioleta. Ela é facilmente reconhecível pela pequena janela de quartzo transparente, localizada bem acima da pastilha.

## Com seu "cérebro" — um programa monitor de 1 kbyte — Nestor é capaz de efetuar vários programas

Pelo fato de ser uma memória de 2 kbytes, utilizamos uma porta E nos terminais Y0 e Y1 do demultiplexador, a fim de selecioná-la. Observe que ela sendo selecionada apenas pelo pino OE (*output enable*), que no nível lógico 1 coloca as saídas da EPROM em seu terceiro estado.

Quanto ao pino CE (*chip enable*), foi conectado diretamente à terra, fazendo com que a memória esteja sempre "pronta". Esse tipo de ligação não é usual, pois faz com que a memória continue "puxando" corrente da fonte, mesmo sem estar sendo requisitada (ela responde por um consumo 50% superior ao normal). Por outro lado, porém, apresenta a vantagem de possibilitar um tempo de acesso mais breve, o que julgamos uma preocupação justa em uma montagem experimental como o Nestor.

Nessa EPROM fica armazenado o programa monitor, "cérebro" do Nes-

tor, utilizando apenas 1 kbyte de seu espaço. Por que optamos, então, por um componente de 2 kbytes? Simplesmente porque a EPROM de 1 k encontrável na praça, a 2708, exige alimentação negativa; e, além do mais, a diferença de preço é irrelevante, quando comparado ao da 2716 e pensamos na possibilidade de utilizar o outro quilobyte da EPROM para outras aplicações.

Dispomos também de um banco de memórias RAM, totalizando 1 kbyte e formado por 2 integrados de 1 k por 4, sendo um deles para os terminais de D0 a D3 e outro, para D4 a D7. Selecione-nos, nesse caso, as memórias 2114, que são estáticas, ou seja, exibem um biestável em cada célula de armazenamento, que retém a informação enquanto o circuito permanece alimentado.

A diferença entre as estáticas e as memórias dinâmicas é que estas possuem uma estrutura interna mais simples e exibem menor tamanho e menor dissipação, mas, ao mesmo tempo, têm o inconveniente de exigir restaurações periódicas da informação armazenada, em intervalos de alguns milissegundos (é o chamado *refresh*).

Escolhemos memórias estáticas por razões óbvias. Nossa objetivo era o de simplificar ao máximo a montagem, dispensando a operação de restauração de dados. Nada impede, porém, que o montador utilize memórias dinâmicas como expansão, no futuro.

As memórias 2114 possuem linhas de dados bidirecionais, que fazem leitura ou escrita, dependendo do sinal WR enviado pela UCP. A seleção é feita através do pino CS (*chip select*), ligado ao terminal Y2 do 74LS138, encarregado de selecionar essa memória nos endereços de 0800H a 0BFFH. Todas as outras saídas do demux poderão ser utilizadas pelo usuário, para efetuar a extensão de memória que julgar necessária.

### Bibliografia

- *The Z-80 Microcomputer Handbook* - William Barden Jr.
- *Construa o seu próprio computador usando o MP-Z80* - Steve Ciarcia
- *Z-80 Assembly Language Programming* - Lance A. Leventhal
- *Z-80 Microcomputer Design Projects* - William Barden Jr.

### Características

- CPU - Z-80A
- RAM estática - 1 kbyte
- EPROM - 2 kbytes
- Display hexadecimal de 6 dígitos      } 16 hexa
- 24 teclas      } 7 funções
- Porta de saída, 8 bits, com *latch*
- Porta de entrada, 8 bits
- Acesso às linhas de dados e endereços, com possibilidade de expansão direta de 5 kbytes
- Clock - 3,58 MHz, a cristal

### Funções

PERMISSÃO (P)	- permite a entrada de dados no campo de endereços
LEITURA (L)	- lê o valor do conteúdo da memória cujo endereço foi colocado
ÚLTIMO (-1)	- decrementa o valor colocado no campo de endereços
PRÓXIMO (+1)	- incrementa o valor colocado no campo de endereços
RODAR (R)	- faz com que o microprocessador comece o programa no endereço colocado no campo correspondente
EX. REG. (ER)	- examina o conteúdo dos registradores
INT. VET. (IV)	- permite a colocação, no campo de endereços, de um endereço que dê partida ao micro, ao ser acionado o NMI

Gostaríamos de agradecer ao Centro de Pesquisas Elétricas da Faculdade de Engenharia Industrial (FEI) pela permissão de utilização dos equipamentos de seu sistema de desenvolvimento de microcomputadores, utilizados na fase de testes de nosso protótipo.

NESTOR, A BASE DE TODOS  
OS COMPUTADORES — 2<sup>a</sup> PARTE

# Montagem do hardware



Possuindo na apresentação do computador da NE, passamos agora à parte de montagem e testes. São dados aqui, também, a lista completa de material e o projeto do circuito impresso

**C**omo havíamos anunciado na edição anterior, eis aqui todas as informações necessárias para se montar o Nestor, além de uma seqüência rigorosa de testes de *hardware*, concebida para garantir a perfeita operação do micro. A parte referente ao *software* e às aplicações será apresentada na última parte da matéria, na NE nº 86.

Se você já optou pela montagem de seu computador, aconselhamos que, antes de mais nada, leia cuidadosamente esta segunda parte e informe-se sobre todos os detalhes da montagem. Tenha o cuidado de verificar se na sua região é possível adquirir facilmente todos os componentes da lista de material e, só depois, comece a fazer suas compras. Tivemos o cuidado de projetar o Nestor utilizando apenas componentes de fácil obtenção no mercado nacional; poderá acontecer, no entanto, de você ter que encomendar alguma coisa às lojas de São Paulo ou Rio.

A seqüência de testes também foi formulada de modo a evitar ao máximo os problemas de operação do micro e a torná-la acessível mesmo aos montadores que não dispõem de instrumentos sofisticados de medida. Assim, uma vez montado e testado, qualquer "irmão" do Nestor deverá funcionar perfeitamente.

## Informações iniciais

A montagem do Nestor será um pouco diferente das tradicionais, devido à maior complexidade e às caracte-

rísticas específicas de um circuito baseado num microprocessador. Assim, ao invés de ser montado todo de uma vez, o Nestor deve ir se formando aos poucos, numa seqüência combinada de montagem e testes. Aconselhamos você a seguir este nosso método, pois é quase certo que, se for pelo processo normal, terá algum problema ao fim da montagem — seja com curtos, pistas abertas, soldas frias — e será bastante difícil localizar a falha.

Desse modo, você verá que o computador estará sendo testado à medida que é montado, uma prática que permite isolar e sanar mais facilmente os defeitos. A seqüência poderá pare-

cer um tanto tediosa e mecânica, no início; mas lembre-se sempre do investimento feito e de como é compensador ter um equipamento funcionando depois de uma montagem — mesmo demorada.

Além disso, o sistema imaginado por nós utiliza apenas instrumentos simples de bancada, tornando-o acessível a todos os montadores. Mês que vem, quando publicaremos o programa monitor do micro, será dada outra bateria de testes, que permitirá analisar o comportamento dinâmico do Nestor — tal como a análise de formas de onda, por exemplo.

Uma última sugestão, para os mon-

## Relação de componentes

### SEMICONDUTORES

CI1- Z80A (CPU de 4 MHz)  
CI2-2716 ou 2516 (EPROM;  
tempo de acesso = 350 ns)  
CI3, CI4- 2114 (RAMs;  
tempo de acesso = 350 ns)  
CI5- 74LS138  
CI6- 74LS174  
CI7, CI8, CI11, CI12- 74LS75  
CI9, CI10, CI13, CI14- 74LS126  
CI15- 74LS04  
CI16- 74LS02  
CI17, CI18- 74LS08  
Q1- BC 237 ou equivalente(6 x)  
Q2- BC 308 ou equivalente(8 x)  
D1- FDH 660 ou equivalente  
D<sub>G1</sub> a D<sub>G6</sub> - NSN 7663 (*display*  
âmbar) ou NSN 7673 (verde)

### RESISTORES (todos de 1/8 W)

R1, R3(2 x), R4(8 x)- 1 kΩ  
R2, R5(6 x)- 10 kΩ  
R6(8 x)- 47Ω  
R7(8 x)- 3,3 kΩ  
R8- 4,7 kΩ  
R9- 330 Ω

### CAPACITORES

C1, C2(2 x)- 10 µF/15 V (tântalo)  
C3, C4, C5- 0,1 µF (poliéster)

### MISCELÂNEA

Mini-cristal para TV  
em cores- 3,578 MHz  
Teclas IPM (24 x)  
Soquetes para os Cls (um de 40 pinos, um de 24 pinos, 2 de 18 pinos, 6 de 16 pinos e 8 de 14 pinos)

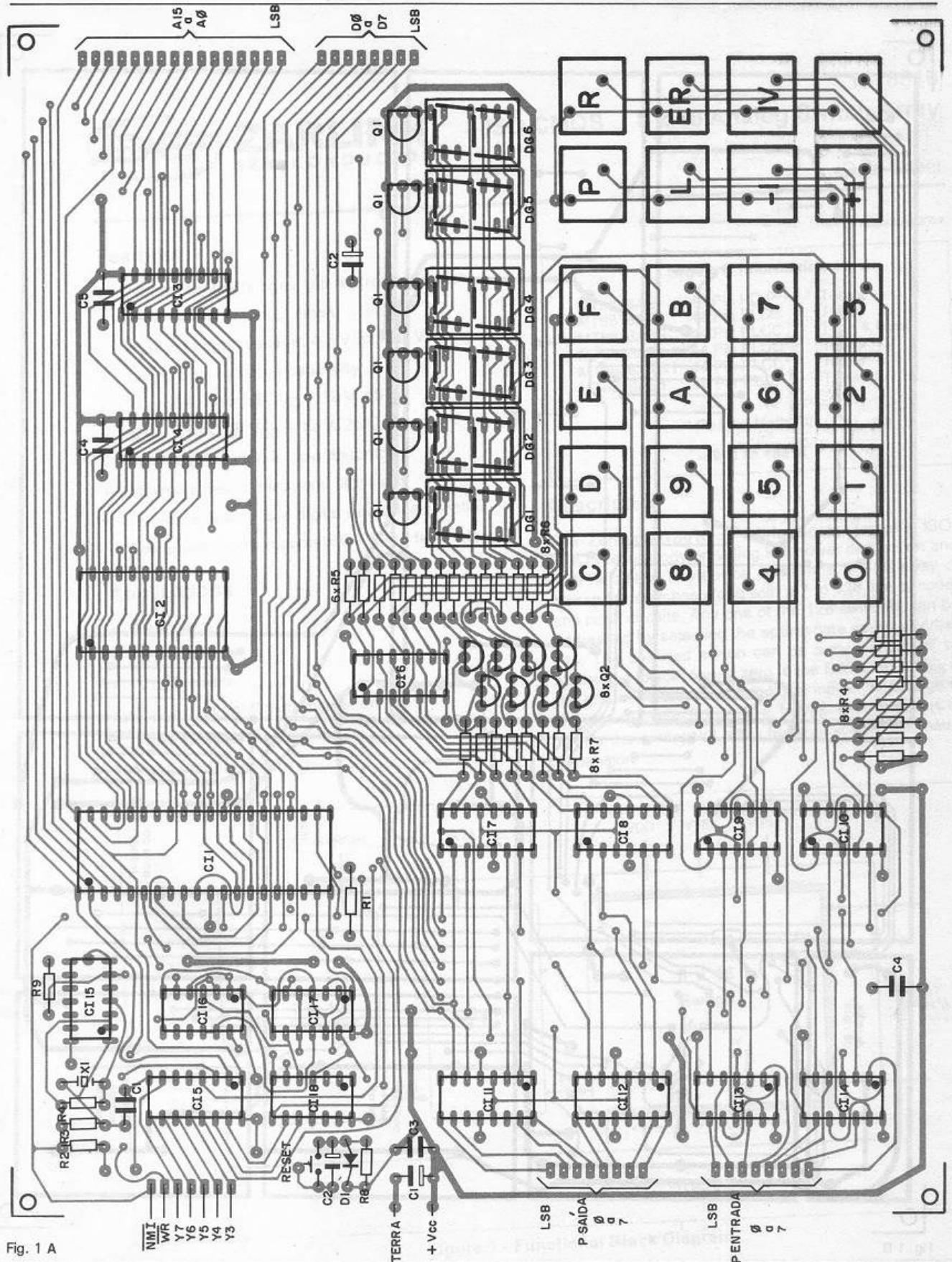


Fig. 1 A

A placa do Nestor é dupla face. Aqui ela aparece vista pelo lado dos componentes, com o traçado da outra face em transparência.

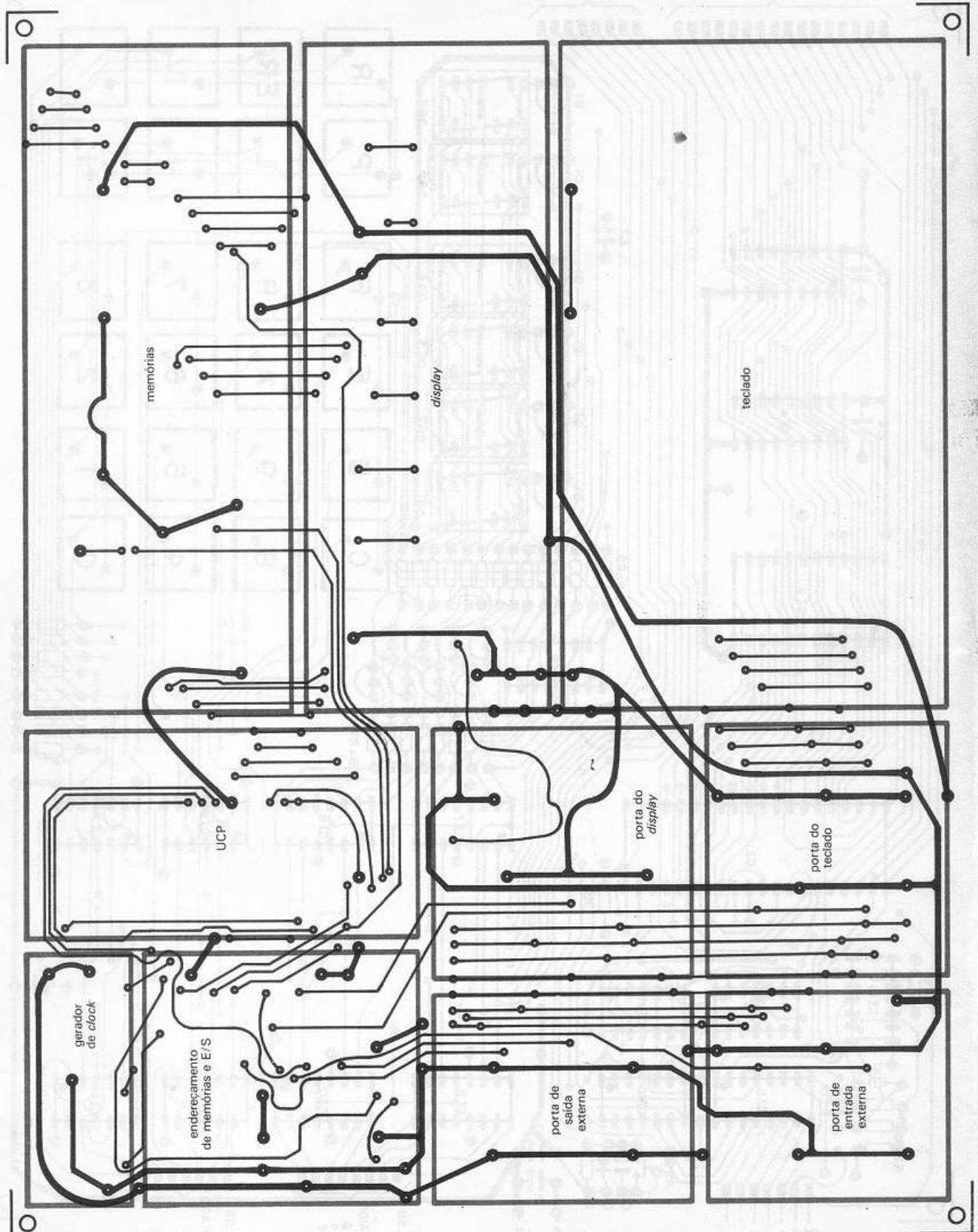


Fig. 1 B

Traçado da face dos componentes, com a sobreposição de uma máscara (opcional) indicando a localização dos vários estágios do Nestor.

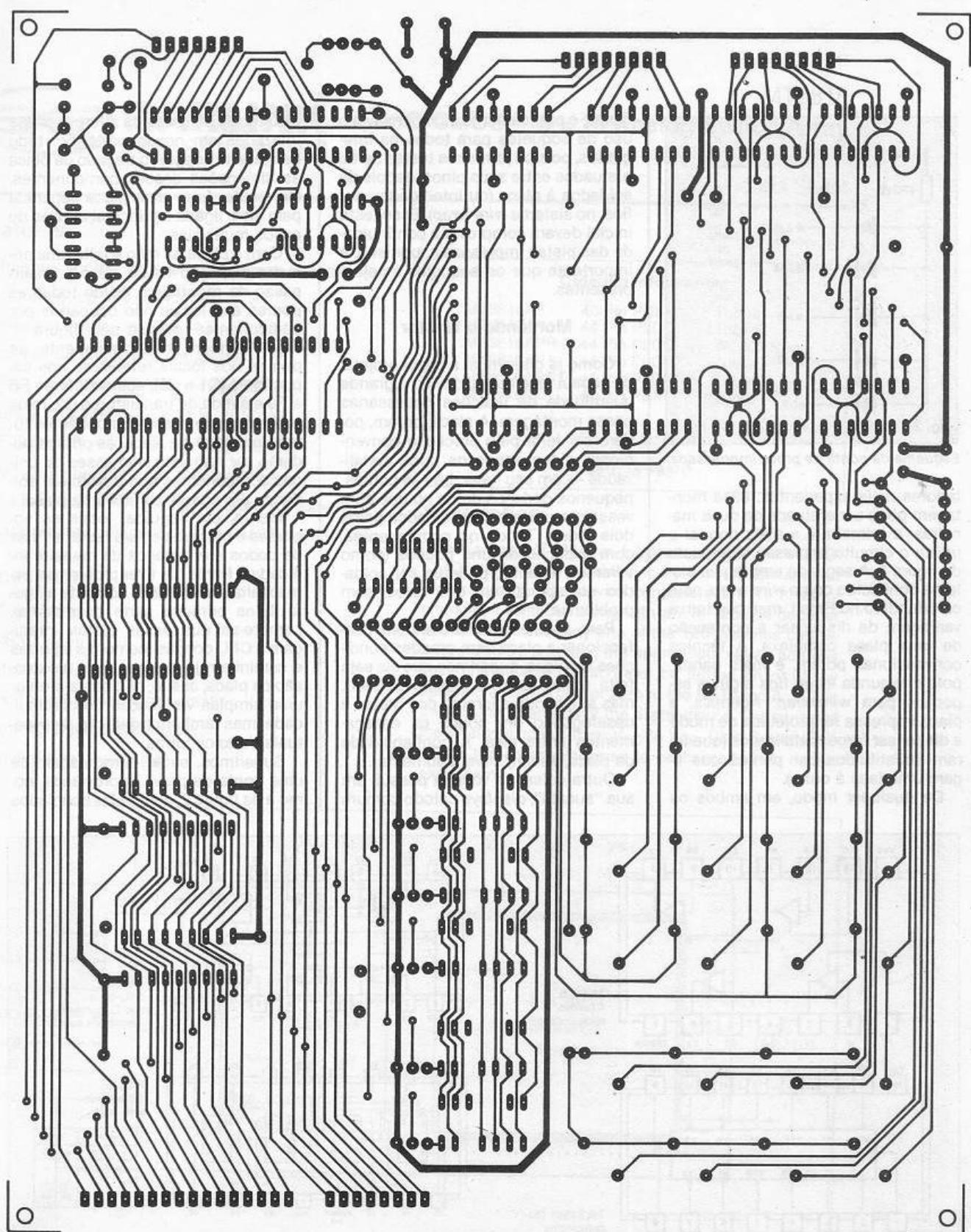


Fig. 1 C

Aqui aparece novamente a face oposta à dos componentes, mas agora na posição certa para ser copiada (a escala é 1:1).

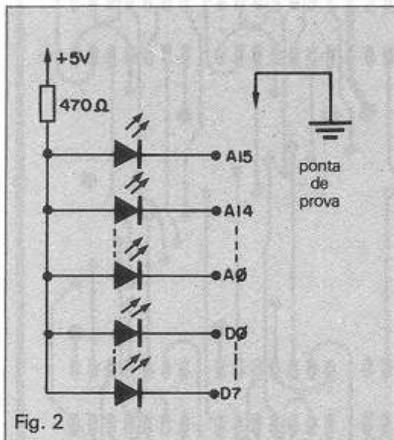


Fig. 2

Esquema da ponta de prova improvisada.

tadores mais experientes: essa montagem pode ser efetuada de duas maneiras; a primeira é a mais normal e utiliza o circuito impresso face dupla da figura 1. A segunda emprega o sistema conhecido como *wire-wrap*, pouco difundido no Brasil, mas que tem a vantagem de dispensar a confecção de uma placa complexa. A técnica convencional, porém, é mais barata, pois a segunda exige fios e pinça especiais para *wire-wrap*. Ademais, a placa impressa foi projetada de modo a dispensar furos metalizados (que foram substituídos por pontes que ligam uma face à outra).

De qualquer modo, em ambos os

casos, é praticamente indispensável o uso de soquetes para **todos** os integrados, pois os primeiros testes serão efetuados entre seus pinos, depois de soldados à placa (ou interligados por fios, no sistema *wire-wrap*). Esse teste inicial deverá comprovar a continuidade das pistas impressas e, por isso, é importante que os soquetes estejam presentes.

### Montando o Nestor

Como já dissemos, a placa projetada possui face dupla, devido à grande quantidade de ligações necessárias nessa montagem. A placa, porém, poderá ser feita pelo processo convencional, dispensando os furos metalizados — em seu lugar, irão as pontes, pequenos pedaços de fio nu que atravessam a placa e são soldados pelos dois lados. Todos os componentes, com exceção de uns poucos (como veremos adiante), poderão ser soldados em apenas um dos lados, sem problemas.

Para aqueles que pretendem confeccionar a placa sem grandes condições técnicas, sugerimos que ela seja feita maior, com o mesmo traçado, mas ampliando a largura dos filetes e desafogando um pouco os espaçamentos entre eles. A confiabilidade da placa, assim, deverá aumentar.

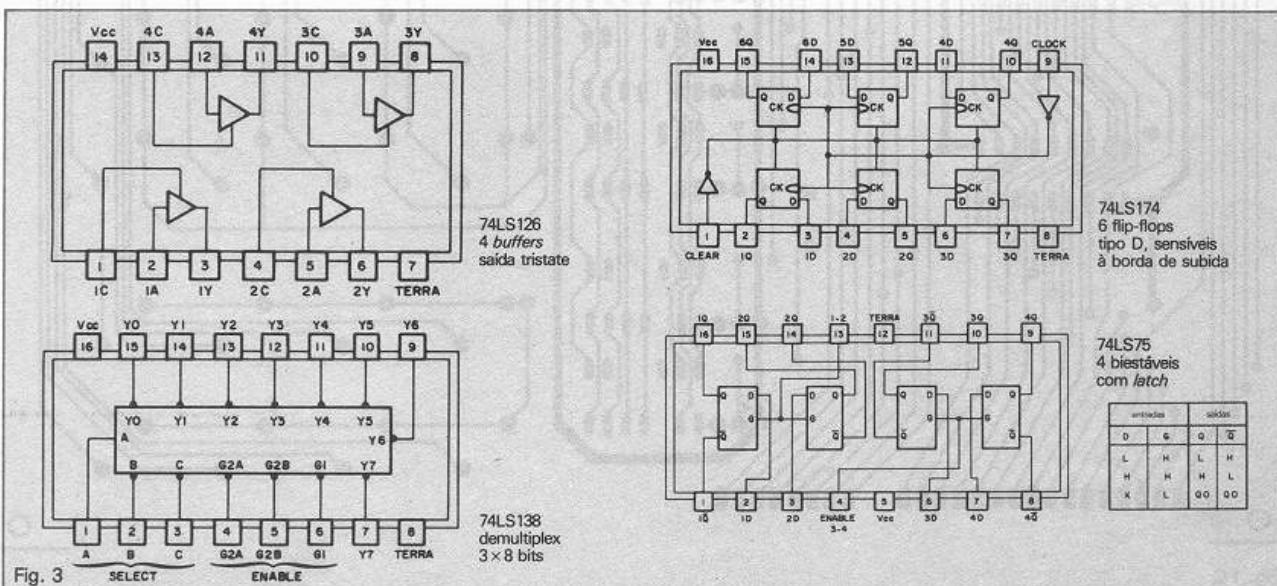
Outra coisa: se você já possui, em sua "sucata", *displays* catodo comum

e/ou teclas de modelos diferentes dos utilizados em nosso protótipo, tudo bem; basta adaptar o traçado da placa às dimensões desses componentes. Ou, então, fazer uma placa separada para eles, ligada à principal através de cabos multiveias.

Com a placa na mão e feita uma primeira inspeção visual, dê o primeiro passo da montagem; soldie todas as pontes aos locais não ocupados por componentes (guie-se pela figura 1). Ligue também, temporariamente, as pontes nos locais referentes aos capacitores (C1 a C5), aos resistores R8 e R9 e à rede de transistores Q2; mais tarde, quando esses componentes forem incluídos na placa, as pontes poderão ser retiradas (são esses os únicos componentes que devem ser soldados em ambas as faces da placa).

Monte, em seguida, todos os soquetes e prepare-se para testar as vias de dados e endereços da maneira indicada à frente (se tiver problemas para localizar essas vias, consulte a figura 4, na primeira parte da matéria). Lembre-se que essas pistas interligam a CPU com as memórias e portas e caminham juntas por toda a extensão da placa; assim, não basta efetuar uma simples verificação de continuidade, mas também a detecção de eventuais curto-circuitos.

Sugerimos, então, a montagem de uma ponta de prova improvisada, como a da figura 2, com LEDs acoplados



Estrutura interna e pinagem de 4 dos CIs empregados na montagem do Nestor. Os demais são apenas conjuntos de portas simples.



Exemplos do formato das palavras de dados.

ao barramento de dados e endereços e um pino metálico ligado à terra. Com ela, teste a continuidade e curtos ocasionais pela indicação dos diodos (no caso de curto-circuito, mais de um LED irá acender por vez).

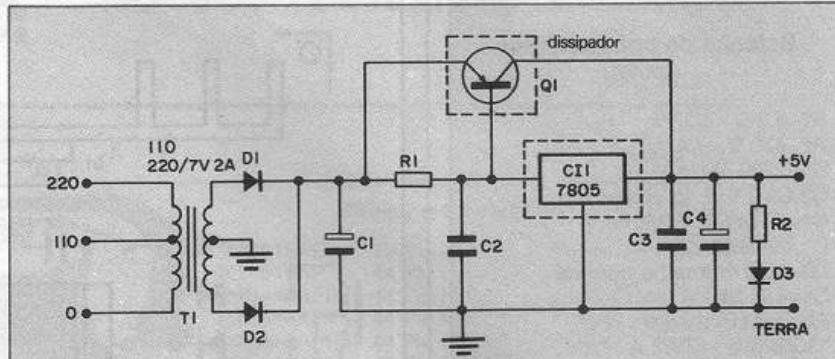
Para ter plena certeza da integridade das linhas em toda sua extensão, lembre-se de checar cada uma em todos os pinos de soquetes pelos quais passam. Assim, por exemplo, seguindo pelo esquema a linha D0, é preciso verificar-la no pino 14 de C11, no pino 9 de C12, no 11 de C14, no 3 de C17, novamente no 3 em C19, no 7 de C11 e, por fim, no pino 8 de C13; o mesmo procedimento vale para todas as outras linhas, sejam de dados ou endereços.

Uma vez concluída essa primeira etapa dos testes, você poderá ter certeza de que essas linhas estão em perfeito estado e poderá então passar às demais, como as de controle, do teclado/display e alimentação; nesses casos, porém, bastará um teste de continuidade com o multímetro, o que vai facilitar bastante o processo.

### Passando aos integrados

Terminada essa parte, pode-se tratar dos testes com os Cls já montados em seus soquetes e alimentados. Para facilitar a vida do montador, reunimos na figura 3 quatro dos componentes empregados na montagem, com sua pinagem, estrutura interna e função. Os demais são Cls de 14 pinos contendo portas simples, facilmente encontráveis em manuais ou revistas.

Mas, vamos lá. Coloque nos soquetes os integrados referentes à portas de saída (C16, C17, C18, C11, C12) e seu respectivo endereçamento (C16, C17, C18); para facilitar a leitura na saída dessas portas, você já pode montar também o conjunto referente



Esquema da fonte sugerida para o Nestor, cuja capacidade já prevê expansões futuras.

ao *display*, ou seja, as redes de resistores R5, R6 e R7, as redes de transistores Q1 e Q2 e os visores D<sub>G1</sub> a D<sub>G6</sub>. Alimente o circuito, então, com +5 V (veja mais adiante, o adendo sobre a fonte de alimentação).

Todo o controle dessas portas deve ser feito através de uma simulação de sinais, aplicados diretamente aos pinos do Z-80A, e que são os seguintes: D0 a D7 (informação), A0, A1, A2, IORQ e WR (controle). Os níveis lógicos poderão ser obtidos diretamente do +Vcc e terra da fonte, através de fios.

A CPU tem acesso a essas portas através de uma instrução de E/S — OUT aa — onde “aa” é o endereço que vai ativar a porta. Tal instrução produz os sinais de IORQ e WR (ambos ativos em 0) e coloca “aa” na via de endereços.

As portas de saída estão mapeadas da seguinte maneira: a dos dígitos (C16) com endereço 02H, a dos segmentos (C17 e C18) com 01H e a porta para uso externo (C11 e C12) com 04H. Estas duas últimas utilizam biesstáveis tipo D sensíveis a nível, enquanto a primeira é sensível à borda de subida.

Para selecionar um dígito com um determinado caractere, no *display*, proceda da seguinte forma: simule, primeiramente, a instrução OUT 03 com A0 e A1 = 1, IORQ e WR = 0 e selecionando, por exemplo, o primeiro dígito da esquerda; para isso, coloque 01H na via de dados e, em seguida, simplesmente mude o nível lógico em IORQ ou WR para 1, para que os biesstáveis retenham a informação. Nesse momento deverá acender, no primeiro visor da esquerda, apenas o segmento “a”, já que o dado encontra-se nas

duas portas correspondentes (01H). Veja a figura 4.

Para formatar outro caractere no *display*, basta simular agora a instrução OUT 01, fazer apenas A0 = 1 e seguir o mesmo processo já visto, mandando informações diferentes pela linha de dados. Para os outros dígitos, ou mais de um dígito por vez, siga o mesmo esquema de testes, até que toda essa parte esteja funcionando bem.

Para a porta de uso externo faz-se a mesma coisa, só que simulando a instrução OUT 04 (A2 = 1) e medindo as

**Relação de componentes  
(fonte)**

R1- 3,3 Ω - 1/2 W  
 R2- 470 Ω - 1/4 W  
 C1- 2200 μF/15 V  
 C2,C3-0,1 μF (poliéster)  
 C4- 10 μF/25 V (tântalo)  
 D1,D2- SK3/02  
 D3- LED vermelho comum  
 Q1- TIP 34  
 T1- transformador  
 110/220 V – 7V/2 A

saídas com um simples voltímetro, a fim de conferir a informação colocada.

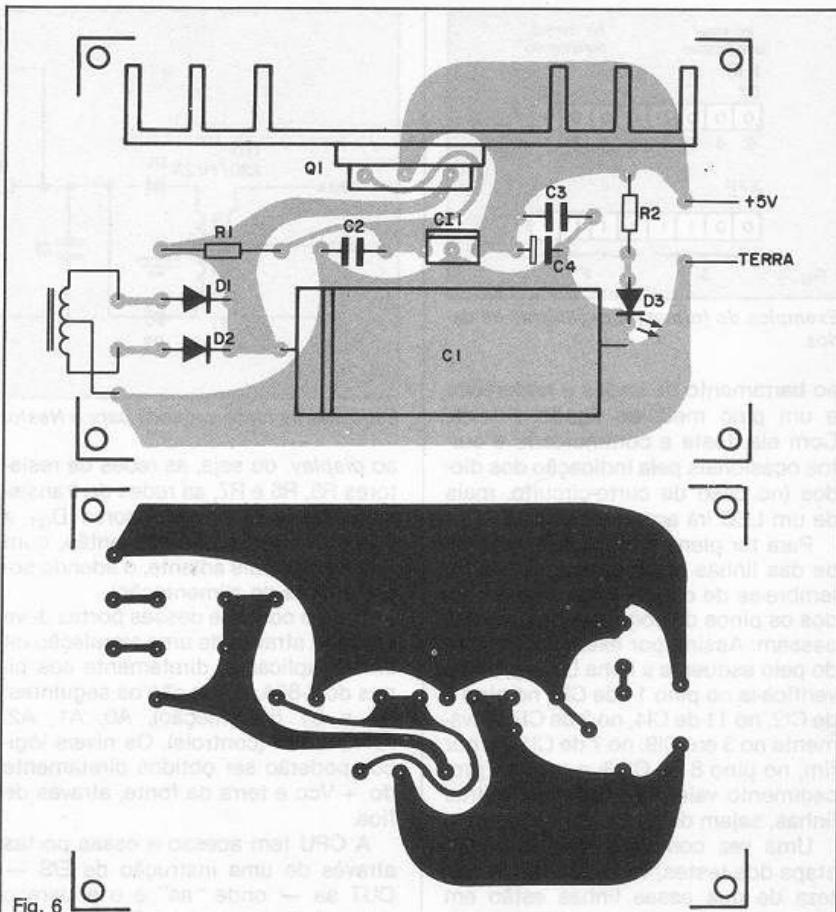
**Continuando...**

Coloque agora nos soquetes os integrados referentes às portas de entrada (C19, C10) e solda a rede de resistores R4, além de todas as teclas. Para facilitar esse teste, retire C16 para poder simular os sinais das colunas, na matriz do teclado.

Este teste exige a simulação da instrução IN aa, que ativa os pinos IORQ, RD e endereça a porta em "aa". No Nestor, a porta de teclado está mapeada com 01H e a de uso externo, com 02H; portanto, coloque a primeira coluna da matriz em nível 1, os pinos IORQ e RD em 0 e A0 em 1 e, ao apertar as teclas relativas a essa 1<sup>a</sup> coluna, "meça" a informação nos pinos da CPU correspondentes à via de dados e compare-a com a tecla pressionada. O mesmo procedimento deve ser estendido depois às outras teclas, desse e das demais colunas.

Faça o mesmo para a porta de uso externo, só que simulando a instrução IN 02 (A1 = 1) e aplicando níveis lógicos na entrada.

Chegou a vez do multiplexador (C15), que deve também estar inserido em seu soquete, a partir de agora. Para testá-lo, você pode utilizar os pinos MREQ, A10, A11 e A12 do soquete da CPU. Comece fazendo MREQ=0 e produzindo todas as 8 combinações possíveis com A10, A11 e A12 — com isso, apenas uma das saídas de C15 estará ativada por vez (pinos Y0 a Y7), em nível 0. Atenção, porém, para as três exceções desse teste: as combinações 000 e 001 devem ser verificadas no pino 20 de C12, enquanto a 010, nos pinos 8 de C13 e C14.



Placa de circuito impresso da fonte, em tamanho natural. O transistor deve receber um dissipador.

**Últimas observações**

Encerrados esses testes, você já pode montar todo o restante do circuito, pois caso surja algum problema de hardware daí para a frente, ele será facilmente resolvido. Como já dissemos, a análise dinâmica do micro só será possível com o programa monitor gravado na EPROM. Veremos isso na próxima edição, onde o programa será publicado integralmente e mais uma seqüência de testes será feita. Daremos ainda todas as formas de onda mais importante do circuito, já com o programa rodando.

**Uma fonte para o Nestor**

Como você deve ter notado, não foi previsto espaço para uma fonte no circuito impresso do micro. De fato, é preferível que a etapa de alimentação ocu-

pe uma placa separada, devido às correntes relativamente elevadas que são necessárias a essa montagem. Na configuração básica em que foi apresentado, o Nestor drena uma corrente de 450 mA, aproximadamente, a +5V.

No entanto, se você ainda não possui uma fonte com essa capacidade de corrente, é preferível montar uma com 1 ou 2 ampéres, prevendo expansões futuras, tais como memórias, interfaces e até um gravador de EPROM.

Sugerimos, por isso, o circuito clássico da figura 5, que pode inclusive ser aproveitado por aqueles que já dispõem de uma fonte comum, utilizando o regulador 7805; basta apenas acrescentar o transistor reforçador de corrente. Na figura 6 aparece a placa de circuito impresso desse mesmo circuito, em tamanho natural.

(Conclui no próximo número)

NESTOR, A BASE DE TODOS  
OS COMPUTADORES — CONCLUSÃO

# Software e operação



Grave o programa monitor — publicado aqui integralmente — na EPROM de seu micro, faça os primeiros testes de funcionamento e utilize os programas aplicativos sugeridos

**L**evando em conta que, a esta altura, todos os montadores devem estar com o *hardware* de seus Nestores devidamente acertado, vamos prosseguir (e concluir) com a apresentação do programa monitor, os testes com o micro ligado, operação básica do mesmo e, por fim, alguns pequenos aplicativos, como o princípio de um relógio digital e o Jogo da Forca.

Gostaríamos apenas de relembrar, antes de continuar, que o Nestor é, basicamente, um pequeno sistema de desenvolvimento para o microprocessador Z-80, de baixo custo, servindo como ferramenta para *hardware* e *software* de vários tipos de microprocessadores — devido à sua compatibilidade de *software* com o 8080, NSC-800, NEO-780 e o 8085, da Intel.

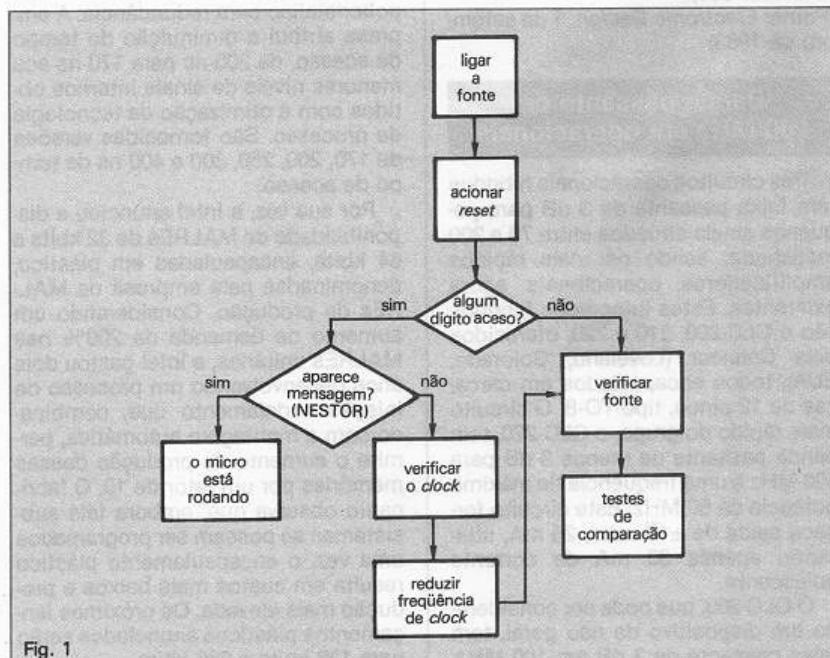
**O programa monitor** — Como sabemos, todo computador exige um sistema operacional que possa servir de interface entre o homem e a máquina, a fim de proporcionar um bom entendimento de parte a parte. Os sistemas operacionais podem ser classificados em diversos tipos, de acordo com sua capacidade e o tipo de máquina a que é destinado.

O chamado “programa monitor”, que não passa de um sistema operacional de pequeno porte, deve ser concebido em perfeita “sintonia” com o projeto do *hardware* — donde se conclui que, para cada *hardware*, existe um sistema operacional específico.

Encontramos hoje, nos diversos tipos de microcomputadores existentes, soluções dos mais variados tipos. Vê-se, por exemplo, computadores com uma estrutura física bastante simplificada, mas com um *software* poderosíssimo — ou vice-versa. As considerações de projeto mais importantes em qualquer caso, porém, são basicamente a velocidade de operação do sistema e seu custo.

No caso específico do Nestor, o projeto do sistema (tanto *hardware* como *software*) foi feito visando construir um sistema de baixo custo, de maneira a viabilizá-lo como montagem experimental. Em outras palavras, temos um *hardware* relativamente “pobre”, compensado, por outro lado, por um programa monitor bem estruturado, como veremos em seguida.

Nosso programa possui um total de



Fluxograma para a seqüência de testes do Nestor.

## Endereços do programa monitor

Tabela 1

- 0000 - Programa principal (PP1): inicialização, mensagem de *reset*
- 0090 - Varredura (VARR): leitura de teclas e varredura dos dígitos
- 009C - Restauração (REST): varredura dos dígitos (catodos)
- 00A9 - Tempo (TEMPO): tempo de varredura
- 00BD - Ajuste de coluna (AJCOL): ajuste do valor da coluna introduzido pela tecla
- 00D8 - Ajuste tecla (AJTEC): correção valor da linha da tecla
- 00F0 - Alterar comando (ALTCOM): recebe flag de comando
- 00F5 - Permissão (ROTPER): apaga o *display* e libera o campo de endereços
- 010F - Campo endereços (ROTCP): entrada de dados, seqüencialmente, no campo de endereços do visor
- 011E - Leitura (ROTLT): mostra valor do dado respectivo do endereço visado
- 012F - Campo dados (ROTCD): permite alteração do dado
- 0140 - Atualização (ATUAL): rotina que permite alteração de valores (endereços e dados)
- 0152 - Decodificação (DECOD): prepara caractere para decodificação
- 0161 - Ajuste endereço memória (RAME): atualiza par DE que fica como indicador de memória
- 016E - Display (DISPLAY): prepara buffer de visor p/ formatação
- 0186 - Formatação (FORMAT): formata valor para 7 segmentos
- 019F - Próximo (ROTPR): incrementa valor do endereço presente no *display*
- 01AF - Último (ROTUL): decrementa valor do endereço presente no *display*
- 01C8 - Rodar (ROTRD): inicialização p/ partida do endereço colocado
- 01E2 - Exame registradores (ROTER): inicializa seqüência dos registradores a serem mostrados no visor
- 01EA - Inicialização (INICIAL): formata *display* para a modalidade de exame de registradores
- 01F4 - Seleção de registradores (SELREG): seleciona o registrador seguinte a ser mostrado
- 0205 - Alteração registradores (ALTREG): rotina de alteração do conteúdo do registrador armado
- 021E - Saída (ROTS): saída com valor do registrador armado
- 02F1 - Interrupção vetorizada (RIV): recebe o endereço colocado como vetor
- 0C00 - Área de pilha do Ex. Registradores
- 0BE4 - Área de pilha do usuário (monitor)
- 0800 a 0BFF - Área de RAM

835 bytes (ou 343 H), contendo um programa principal, 24 sub-rotinas e tabelas de formatos e saltos. Devido à sua extensão, preferimos não esquematizá-lo em diagramas de blocos (seriam necessários mais de vinte deles). Estamos, portanto, publicando somente a listagem com os mnemônicos em *Assembler* e os respectivos códigos de máquina. Veja, na Tabela 1, os endereços principais do programa monitor, seguidos de suas respectivas funções; a forma de se utilizar essa biblioteca de software será vista na parte relativa aos programas, no final desta matéria.

De posse do programa monitor, já gravado em EPROM, estamos prontos para empreender a seqüência de testes dinâmicos, já prometida na 2<sup>a</sup> parte da série. Aconselhamos que seja seguida à risca, para garantir o perfeito funcionamento do micro.

**Novos testes de hardware** — Antes de mais nada, é preciso ter uma

EPROM 2716 gravada com o programa apresentado e conferir cuidadosamente essa gravação — lembre-se que não é perda de tempo, pois esses cuidados evitam muita dor de cabeça depois.

Feita essa checagem, instale a memória na placa, em sua posição correta; se todos os demais componentes estiverem montados e conferidos, basta ligar a fonte e acionar o *reset* do micro (o que deve ser feito com a micro-chave instalada no extremo esquerdo da placa). Ele deverá funcionar imediatamente, o que poderá ser constatado com o surgimento da palavra "Nestor" em seu visor.

Para facilitar ainda mais essa etapa, montamos o fluxograma da figura 1, que prevê todas as possibilidades de operação ou problemas com o computador. Ele só é válido, porém, caso tenham sido efetuados os testes passivos da edição passada — considerando, assim, placa e Cls como já testados e idôneos.

Se, por acaso, você chegar até o

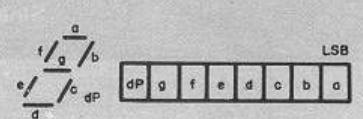


Fig. 2

Formato da palavra do display e códigos dos caracteres mais usados.

bloco "testes de comparação", será necessário efetuar algumas verificações em pontos estratégicos do circuito. As Tabelas 2 e 3 servirão de apoio nesse caso, pois trazem os tipos de formas de onda encontrados em tais pontos, **após o acionamento do reset**.

Há um outro bloco, no fluxograma, que diz: "diminuir a freqüência de *clock*". Para isso, basta acrescentar, em paralelo com o cristal, um capacitor da ordem de algumas centenas de picofarads. Essa redução da freqüência poderá tornar-se necessária no caso de você estar usando (mesmo sem saber) memórias mais lentas que as previstas.

Dedique também uma certa atenção à forma de onda do *clock*, que deverá estar perfeitamente quadrado; nesse caso, a única modificação a fazer é a substituição de CI15 (74LS04) por outro. Esse tipo de problema, contudo, é bastante raro.

Com relação à fonte, não basta que ela esteja fornecendo os 5 volts prescritos, já que podem surgir ruídos devido ao chaveamento do *display* e das memórias, que, por L.di/dt, podem produzir ruídos (pulsos breves) de cerca de 20% de Vcc; além disso, tais pulsos dificilmente são captados por osciloscópios comuns. Esse problema deverá variar de montador para montador, devido às diferenças entre Cls do mesmo tipo e a pequenas variações no traçado da placa.

Para todos os casos, porém, a solução é a mesma: ligar, soldando diretamente através de fios, o +5 V e o terra da fonte aos pinos correspondentes da CPU, sem desfazer a ligação original; em seguida, basta verificar se a operação foi alterada em alguma coisa.

Consideramos essas informações

endereço	instrução cód.máq.	label	mnemônico	endereço	instrução cód.máq.	label	mnemônico
09BA	C2B109		JP NZ, A16	09D7	FE32		CP 32H
09BD	215909		LD HL, NDV	09D9	CAD009		JP Z, PERDEU
09C0	7E		LD A, (HL)	09DC	60	A17	LD H, B
09C1	FE07		CP 07H	09DD	69		LD L, C
09C3	CAD009		JP Z, PERDEU	09DE	C5		PUSH BC
09C6	C39308		JP, A9	09DF	CD9C00		CALL REST
09D0	012009	PERDEU	LD BC, MDP	09E2	C1		POP BC
09D3	1620	A18	LD D, TEMPO	09E3	15		DEC D
09D5	03		INC BC	09E4	C2DC09		JP NZ, A17
09D6	79		LD A, C	09E7	C3D309		JP, A18

## Tabelas de mensagens

## "jogo da força"

endereço	cód. máq.
0900	FF
0901	FF
0902	FF
0903	FF
0904	FF
0905	FF
0906	E1
0907	C0
0908	82
0909	C0
090A	FF
090B	FF
090C	A1
090D	88
090E	FF
090F	FF
0910	8E
0911	C0
0912	CE
0913	C6
0914	88
0915	a
091A	FF

## "perdeu"

endereço	cód. máq.
0920	FF
0921	FF
0922	FF
0923	FF
0924	FF
0925	FF
0926	C7
0927	86
0928	FF
0929	FF
092A	A1
092B	88
092C	C8
092D	C6
092E	86
092F	FF
0934	FF

## "erro"

endereço	cód. máq.
0961	F9
0962	A4
0963	B0
0964	99
0965	92
0966	82
0967	F8

**Conclusão** — Este foi um trabalho feito com muito empenho. Acreditamos que todos os leitores — tenham ou não montado seu próprio micro — usufruiram alguma coisa dele, algo que poderá ser de grande utilidade numa sociedade cada vez mais informatizada.

Nosso objetivo foi o de transmitir informações àqueles que gostariam de aprender realmente a "mexer" em

microcomputadores e sem gastar muita verba com isso. Deixamos isso claro quando anunciamos que o Nestor é um micro dirigido a quem quer aprender a projetar sistemas e não apenas operá-los — ou seja, foi projetado especificamente para os estudantes, profissionais ou autodidatas de eletrônica.

Assim, com o Nestor e um curso de *Assembler* (ou mesmo um bom livro

que trate desse tema), qualquer montador poderá ter em pouco tempo um bom conhecimento dessa área. Sem falar no mercado de trabalho em informática, que ainda resiste à crise e pode oferecer boas oportunidades, esse pequeno micro pode significar mais um passo no domínio da tecnologia por brasileiros e para brasileiros — um importante fator de independência industrial e de pesquisa.

Sinais da CPU

Tabela 2

PINOS DA CPU	FUNÇÃO	NÍVEIS LÓGICOS
30 a 40, 1 a 5	A0 a A15	pulsados
7 a 10, 12 a 15	D0 a D7	pulsados
6	clock	pulsados
19,20, 21,22,27	M1, MREQ, IORQ, RD, WR	pulsados
16,17,24 25,26	WAIT, INT, NMI, RESET, BUSREQ	nível alto

sobre o hardware como suficientes para a montagem do Nestor, sem tropeços. Estamos, no entanto, à disposição de todos os montadores para sanar eventuais dúvidas através de carta.

**Operação: primeiros passos** — Vamos dar início à seção de operação, explicando detalhadamente a função de cada tecla do micro, onde adotaremos a seguinte notação:

DISPLAY — **1 2 3 4 5 6**

campo de endereços   campo de dados

TECLADO — **P**

Tecla **[RS] (reset)** — como você deve ter notado, não foi previsto espaço no teclado para essa função; ela foi propositalmente separada, sendo realizada por uma micro-chave de pressão colocada pouca acima da entrada da fonte e fixada na placa. A razão de isolá-la foi evitar que, accidentalmente, provoque a interrupção de um programa que esteja sendo rodado.

Esse é o único comando por hardware e é responsável pela inicialização do programa monitor, pois coloca o registrador PC (contador de programa) em 0000. Quando acionado, faz aparecer no visor a mensagem de *reset* — "Nestor" — indicando que o micro já foi inicializado. O *reset* não altera o conteúdo das memórias, embora o conteúdo dos registradores (exceto o do PC) da CPU não seja garantido pelo fabricante.

Tecla **[P] (permissão)** — tem como função a abertura do campo de endereços, que são os quatro primeiros dígitos do visor, da esquerda para a direita.

Ao ser pressionada, essa tecla faz com que o *display* apague totalmente, indicando permissão para a entrada dos algarismos em hexadecimal. Ao serem teclados, os algarismos vão en-

trando nesse campo da direita para a esquerda e podem ser carregados indefinidamente, sempre "empurrando" os outros algarismos no mesmo sentido. Isso evita ter que pressionar a tecla de permissão novamente ao ocorrer um erro de digitação, bastando apenas colocar o valor correto em sequida.

O endereço mostrado no visor servirá em três processos diferentes: listagem e modificação do dado dessa locação, como endereço de partida para rodar e como vetor de interrup-

ção — correspondentes às teclas que veremos a seguir.

Tecla **[L] (leitura)** — tem a função de mostrar, no 5º e 6º dígitos do visor (campo de dados), o valor em hexadecimal da locação mostrada no campo de endereços. Ao ser pressionada, além de apresentar esse conteúdo, permite a substituição direta do mesmo, a exemplo do que ocorre com o campo de endereços, ou seja, com os algarismos entrando no mesmo sentido; permite também fazer correções diretamente, como já visto.

É importante notar que o valor mostrado no *display*, tenha sido alterado ou não, já é o conteúdo real da locação respectiva, não necessitando de um comando posterior para confirmação, como acontece em outros micros.

Tecla **[+1] (próximo)** — incrementa de uma locação o valor indicado no campo de endereços, mostrando o conteúdo no campo de dados, que é operado exatamente como foi descrito acima.

Tecla **[-1] (último)** — decrementa de uma locação o valor indicado no campo de endereços.

Tecla **[R] (rodar)** — faz com que o processador rode o programa que começa na locação apresentada pelo campo de endereços. Em outras palavras, para essa operação é necessária a colocação prévia de um endereço válido, através da tecla de permissão. A execução do programa é imediata, ou seja, a partir do momento em que a tecla "R" é acionada.

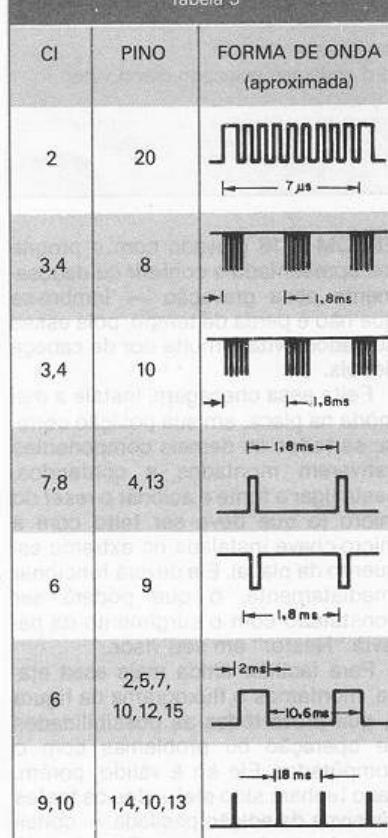
Tecla **[ER] (exame de registradores)** — tem como objetivo a leitura e alteração do conteúdo dos registradores. Quando acionada, então, faz surgir no visor o termo **[A = XX]**, onde XX é o conteúdo do registrador A. Se for pressionada, a seguir, a tecla **[+1]**, aparecerá o conteúdo de B, e assim por diante.

Se for acionada uma tecla hexadecimal, o valor entrará no campo de dados da direita para a esquerda, alterando o conteúdo do registrador. Nota-se que esta função é menos operacional, embora seja bastante didática. Num dos exemplos que vêm a seguir, você verá como utilizá-la.

Tecla **[IV] (interrupção vetorizada)** — como se pode ver pelo esquema do Nestor, publicado na 1ª parte, existe a opção de liberação do NMI para uso externo. A função dessa tecla, então, é a de colocar um endereço válido — que servirá como vetor — quando houver uma transição de 1 para 0 no NMI. Em outras palavras, quando isso ocorrer, o programa partirá da locação escolhida como vetor. A operação é simples, bastando colocar o endereço através da tecla de permissão e, a seguir, acionar esta tecla.

Sinais em alguns Cls

Tabela 3



Todas as formas de onda  
para clock de 3,58 MHz

```

***** PROGRAMA MONITOR NESTOR *****

PROGRAMA MONITOR NESTOR
NOVA ELETRONICA
SAO PAULO S.P. 01/84 MCS5/JRP
***** PROGRAMA PRINCIPAL *****
END. INSTR. LABEL MNEMONICO
0000 31000C PP1 LD SP,0C00H
0003 C5 PUSH BC
0004 D5 PUSH DE
0005 E5 PUSH HL
0006 F5 PUSH RF
0007 21E60B LD HL,FLAG
0008 3619 LD (HL),19
000C 215C00 LD HL,MENS
000F 11E70B LD DE,BUFDIS
0012 010600 LD BC,0006H
0015 ED80 LDIR
0017 31E40B INICIO LD SP,0BE4H
001A 21E70B LD HL,BUFDIS
001D 0600 LD B,00H
001F 1E06 LD E,06H
0021 3E01 LD A,01H
0023 CD9000 NE1 CALL VRAR
0026 79 LD A,C
0027 87 ADD A,A
0028 1D DEC E
0029 C22300 JP NZ,NE1
002C 78 LD A,B
002D FE00 CP 00H
002F CR1700 JP Z,INICIO
0032 21E70B NE2 LD HL,BUFDIS
0035 CD9C00 CALL REST
0038 7A LD A,D
0039 D303 OUT PDIG
003B DB01 IN A,PTEC
003D FE00 CP 00H
003F C23200 JP NZ,NE2
0042 CDRD00 CALL AJCOL
0045 CDD800 CALL AJTEC
0048 FE13 CP 13H
004A CR7700 JP Z,NE3
004D FE10 CP 10H
004F DR8300 JP C,NE4
0052 CDF000 NE5 CALL ALTCOM
0055 21E60B NE6 LD HL,FLAG
0058 76 LD A,(HL)
0059 C36900 JP RECCOM
005C C8 DEFB
005D 86 DEFB
005E 92 DEFB
005F 87 DEFB
0060 C0 DEFB
0061 CE DEFB
0062 00 NOOP
0063 00 NOOP
0064 00 NOOP
0065 00 NOOP
***** PONTO DE ENTRADA DE INTERRUPCAO NMI *****
0066 C3F20B JP RIV
0069 212003 RECCOM LD HL,TBRCOM
006C 85 RD A,L
006D 6F LD L,A
006E 7E LD A,(HL)
006F 4F LD C,A
0070 3E0R LD A,0AH
0072 85 ADD A,L
0073 6F LD L,(HL)
0074 6E LD H,C
0075 61 LD (HL)
0076 E9 JP (HL)
0077 21E60B NE3 LD HL,FLAG
0078 7E LD A,(HL)
007B FE15 CP 15H
007D CRF401 JP Z,SELREG
0080 C35200 JP NE5
0083 21E60B NE4 LD HL,FLAG
0086 7E LD A,(HL)
0087 FE15 CP 15H
0089 C0502 JP Z,ALTREG
008C 00 NOOP
008D C35500 JP NE6
***** ROTINA DE VARREDURA *****
0090 4F VRAR LD C,A
0091 CDA900 CALL TEMPO
0094 DB01 IN A,TECLADO
0096 FE00 CP 00H
0098 C8 RET Z
0099 47 LD B,A LINHA
009A 51 LD D,C COLONA
009B C9 RET
***** ROTINA DE RESTAURACAO *****
009C 0E06 REST LD C,06H
009E 3E01 LD A,01H
0090 CDA900 NE7 CALL TEMPO
0093 87 ADD A,A
0094 0D DEC C
0095 C8 RET Z
0096 C3R000 JP NE7
***** ROTINA DE TEMPO *****
0099 FS TEMPO PUSH RF
009A D303 OUT PDIG
009C 7E LD A,(HL)
009D D301 OUT PSEG
009F D9 EXX
00B0 110001 LD DE,ATRASO
00B3 1B LOOP1 DEC DE
00B4 7A LD A,D
00B5 B3 OR E
00B6 C2B300 JP NZ,LOOP1
00B9 D9 EXX
00B8 23 INC HL
00B9 F1 POP RF
00BC C9 RET
***** ROTINA DE AJUSTE DE COLUNA *****
00C5 87 ADD A,A
00C6 0E08 LD C,06H
00C8 BA CP D
00C2 C4D600 JP Z,NE8
00C9 C4D600 JP Z,NE8
00C8 BA CP D
00C5 87 ADD A,A
00C6 0E08 LD C,06H
00C8 BA CP D
00C9 C4D600 JP Z,NE8
00D3 C31700 JP INICIO
***** ROTINA DE AJUSTE DE TECLA *****
00D8 1E0B AJTEC LD E,08H
00D9 0E00 LD C,00H
00DC 3E01 LD A,01H
00DE 88 NE10 CP B
00DF CREB00 JP Z,NE9
00E2 0C INC C
00E3 87 ADD A,A
00E4 1D DEC E
00E5 C2DE00 JP NZ,NE10
00E8 C31700 JP INICIO
00EB 79 NE9 LD A,C
00EC 82 ADD A,D
00ED 47 LD B,A
00EE 00 NOOP
00EF C9 RET
***** ROTINA DE ALTERAR COMANDO *****
00F0 21E60B ALTCOM LD HL,FLAG
00F3 70 LD (HL),B
00F4 C9 RET
***** ROTINA DE PERMISSAO *****
00F5 0E06 ROTPER LD C,06H
00F7 21E60B LD HL,FLAG
00F8 3617 LD (HL),17
00FC 23 INC HL
00FD 36FF NE11 LD (HL),FFF
00FF 23 INC HL
0100 00 DEC C
0101 C2FD00 JP NZ,NE11
0104 3600 LD (HL),00H
0106 23 INC HL
0107 3600 LD (HL),00H
0109 23 INC HL
010R 3600 LD (HL),00H
010C C31700 JP INICIO
***** ROTINA DE CAMPO DE ENDERECONS *****
010F 21EE0B ROTCP LD HL,PBE
0112 78 LD A,B
0113 ED6F RLD
0115 23 INC HL
0116 ED6F RLD
0118 CDS201 CALL DECOD
011B C31700 JP INICIO
***** ROTINA DE LEITURA *****
011E CD4001 ROTLT CALL ATUAL
0121 21E60B LD HL,FLAG
0124 3618 LD (HL),18
0126 21EFOB LD HL,PRE
0129 CDS201 CALL DECOD
012C C31700 JP INICIO
***** ROTINA DE CAMPO DE DADOS *****
012F CD4001 ROTCD CALL ATUAL
0132 78 LD A,B
0133 ED6F RLD
0135 7E LD A,(HL)

```

0136 12 LD (DE),A	018C 1A LD A,(DE)	01EF 23 INC HL
0137 21EF08 LD HL,PRE	018D D1 POP DE	01F0 36FF LD (HL),FFH
0138 CD5201 CALL DECOD	018E C9 RET	01F2 23 INC HL
013D C31700 JP INICIO		01F3 C9 RET
<b>TABELA DE FORMATS ALFA</b>		
<b>ROTINA DE ATUALIZACAO</b>	<b>ROTINA DE SELECAO DE REGISTRADORES</b>	
0140 21EE0B ATUAL LD HL,PRE	018F C0 DEF8	01F4 110403 SELREG LD DE,TABREG
0143 5E LD E,(HL)	0190 F9 DEF8	01F7 21E50B LD HL,MENCON
0144 23 INC HL	0191 A4 DEF8	01FA 34 INC (HL)
0145 56 LD D,(HL)	0192 B0 DEF8	01FB 34 INC (HL)
0146 1A LD A,(DE)	0193 99 DEF8	01FC 7B LD A,E
0147 2B DEC HL	0194 92 DEF8	01FD 86 RDO A,(HL)
0148 2B DEC HL	0195 82 DEF8	- 01FE 57 - NE12 LD E,A
0149 77 LD (HL),A	0196 F8 DEF8	01FF 1A LD A,(DE)
014A E5 PUSH HL	0197 80 DEF8	0200 6F LD L,A
014B 21E60B LD HL,FLAG	0198 90 DEF8	0201 13 INC DE
014E 3618 LD (HL),18H	0199 88 DEF8	0202 1A LD A,(DE)
0150 E1 POP HL	019A 83 DEF8	0203 67 LD H,A
0151 C9 RET	019B C6 DEF8	0204 E9 JP (HL)
<b>ROTINA DE DECODIFICACAO</b>	<b>ROTINA DE ALTERACAO DE REGISTRADORES</b>	
0152 01E70B DECOD LD BC,BUFDIS	019C A1 DEF8	0205 21E50B ALTREG LD HL,MENCON
0155 CD6E01 CALL DISPLAY	019D 86 DEF8	0208 111B03 LD DE,TABALT
0158 2B DEC HL	019E 8E DEF8	020B 7B LD A,E
0159 CD6E01 CALL DISPLAY		020C 86 RDO A,(HL)
015C 2B DEC HL		020D C3FE01 JP NE12
015D CD6E01 CALL DISPLAY		<b>REGISTRADOR A</b>
0160 C9 RET		0210 CD4001 REGA CALL INICIAL
<b>ROTINA DE AJUSTE MEMORIA DE ENDERECO</b>	<b>ROTINA DE PROXIMO</b>	0213 3688 LD (HL),88H
0161 21EE0B RAME LD HL,PRE	019F CD4001 ROTPR CALL ATUAL	0215 23 INC HL
0164 73 LD (HL),E	01B2 1B DEC DE	0216 3687 LD (HL),B7H
0165 23 INC HL	01B3 CD6101 CALL RAME	0218 21F90B LD HL,MENA
0166 72 LD (HL),D	01B6 21EF02 LD HL,PRE	021B C31E02 JP ROTSA
0167 2B DEC HL	01B9 CD5201 CALL DECOD	<b>ROTINA DE SAIDA</b>
0168 2B DEC HL	01BC C31700 JP INICIO	021E 01EB0B ROTSA LD BC,BYTES
0169 1A LD A,(DE)	01BF RTE 01C7 - NAO USADA	0221 CD6E01 CALL DISPLAY
016A 77 LD (HL),A		0224 C31700 JP INICIO
016B C9 RET		<b>ALTERA REGISTRADOR A</b>
016C FF LOC. NAO USADA		0227 21F90B ALTA LD HL,MENA
016D FF LOC. NAO USADA		0229 7B LD A,B
<b>ROTINA DE DISPLAY</b>	<b>ROTINA DE ULTIMO</b>	022B ED6F RLD
016E 7E DIS- LD A,(HL)	01C8 3EFF ROTRD LD A,FFH	022D C31E02 JP ROTSA
016F E5 PLAY PUSH HL	01C9 D303 OUT PDIG	<b>REGISTRADOR B</b>
0170 21F00B LD HL,ROT	01CC 3E7F LD A,7FH	0230 CD4001 REGB CALL INICIAL
0173 77 LD (HL),A	01CE D301 OUT PSEG	0233 3683 LD (HL),33H
0174 AF XOR A	01D0 21ED0B LD HL,MEND	0235 23 INC HL
0175 ED6F RLD	01D3 36C3 LD (HL),C3H	0236 3687 LD (HL),B7H
0177 CD8601 CALL FORMAT	01D5 31FB0B LD SP,0BE4H	0238 21FF0B LD HL,MENB
017A 02 LD (BC),A	01D8 F1 PUSH RF	023B C31E02 JP ROTSA
017B 03 INC BC	01D9 E1 PUSH RL	<b>ALTERA REGISTRADOR B</b>
017C AF XOR A	01DA D1 PUSH DE	023E 21FF0B ALTB LD HL,MENB
017D ED6F RLD	01DB C1 PUSH BC	0241 7B LD A,B
017F CD8601 CALL FORMAT	01DC 31E40B LD SP,0BE4H	0242 ED6F RLD
0182 02 LD (BC),A	01DF C3ED0B JP ENDISP	0244 C31E02 JP ROTSA
0183 03 INC BC		
0184 E1 POP HL		
0185 C9 RET		
<b>ROTINA DE FORMATACAO</b>	<b>ROTINA DE EXAME DE REGISTRADORES</b>	
0186 D5 FORMAT PUSH DE	01E2 21E50B ROTER LD HL,MENCON	
0187 118F01 LD DE,INTD	01E5 36FE LD (HL),FEH	
0188 83 RDO A,E	01E7 C3F401 JP SELREG	
018B 5F LD E,A		
<b>ROTINA DE INICIALIZACAO</b>	<b>ROTINA DE INICIALIZACAO</b>	
0186 D5 FORMAT PUSH DE	01EA 21E70B INICIAL LD HL,BUFDISP	
0187 118F01 LD DE,INTD	01EB 36FF LD (HL),FFH	
0188 83 RDO A,E		
018B 5F LD E,A		

REGISTRADOR C		REGISTRADOR L		0300 02 0301 C31700	LD (BC),A JP INICIO
0247	CDER01 REGC	CALL	INICIAL	0283 CDER01 REGL	CALL INICIAL
0248	36C6	LD	(HL),C6H	0286 36C7	LD (HL),C7H
024C	23	INC	HL	0288 23	INC HL
024D	36B7	LD	(HL),B7H	0289 36B7	LD (HL),B7H
024F	21FE0B	LD	HL,MENL	028B 21FB0B	LD HL,MENL
0252	C31E02	JP	ROTSa	028E C31E02	JP ROTSA
ALTERA REGISTRADOR C		ALTERA REGISTRADOR L		TABELA DE SELEÇÃO DE REGISTRADORES	
0255	21FE0B	ALTC	LD HL,MENL	0281 21FB0B	ALTL LD HL,MENL
0258	78	LD	A,B	0284 78	LD A,B
0259	ED6F	RLD		0285 ED6F	RLD
025B	C31E02	JP	ROTSa	0287 C31E02	JP ROTSA
REGISTRADOR D		REGISTRADOR F		0304 10 0305 02 0306 30 0307 02 0308 47 0309 02 030A 5E 030B 02 030C 75 030D 02 030E 8C 030F 02 0310 A3 0311 02 0312 B9 0313 02 0314 D1 0315 02 0316 18 0317 03 0318 C3E201 JP ROTER	
025E	CDER01 REGD	CALL	INICIAL	028A CDER01 REGF	CALL INICIAL
0261	36A1	LD	(HL),A1H	028D 368E	LD (HL),SEH
0263	23	INC	HL	028F 23	INC HL
0264	36B7	LD	(HL),B7H	0290 36B7	LD (HL),B7H
0266	21FD0B	LD	HL,MEND	0292 21FB0B	LD HL,MEND
0269	C31E02	JP	ROTSa	0295 C31E02	JP ROTSA
ALTERA REGISTRADOR D		ALTERA REGISTRADOR F		TABELA DE ALTERAÇÃO DE REGISTRADORES	
026C	21FD0B	ALTD	LD HL,MEND	0298 21FB0B	ALTF LD HL,MEND
026F	78	LD	A,B	02CB 78	LD A,B
027A	ED6F	RLD		02CC ED6F	RLD
0272	C31E02	JP	ROTSa	02CE C31E02	JP ROTSA
REGISTRADOR E		REGISTRADOR I		031B 27 031C 02 031D 3E 031E 02 031F 55 0320 02 0321 6C 0322 02 0323 83 0324 02 0325 9A 0326 02 0327 B1 0328 02 0329 C8 032A 02 032B E5 032C 02	
0275	CDER01 REGE	CALL	INICIAL	02D1 ED57 REGI	LD A,I
0278	3686	LD	(HL),86H	02D3 21F70B	LD HL,MENI
0279	23	INC	HL	02D6 77	LD (HL),A
027B	36B7	LD	(HL),B7H	02D7 CDER01	CALL INICIAL
027D	21FC0B	LD	HL,MENL	02DA 36F9	LD (HL),F9H
0280	C31E02	JP	ROTSa	02DC 23	INC HL
ALTERA REGISTRADOR E		ALTERA REGISTRADOR I		02DD 36B7	LD (HL),B7H
0283	21FC0B	ALTE	LD HL,MENE	02DE 21F70B	LD HL,MENI
0286	78	LD	A,B	02E2 C31E02	JP ROTSA
0287	ED6F	RLD		0324 02 0325 9A 0326 02 0327 B1 0328 02 0329 C8 032A 02 032B E5 032C 02	
REGISTRADOR H		ROTINA PARA O VETOR DE INTERRUPÇÃO		TABELA DE SELEÇÃO DE COMANDOS	
028C	CDER01 REGH	CALL	INICIAL	02E5 21F70B	ALTI LD HL,MENI
028F	36B9	LD	(HL),B9H	02E8 78	LD A,B
0291	23	INC	HL	02E9 ED6F	RLD
0292	36B7	LD	(HL),B7	02EB 7E	LD A,(HL)
0294	21FB0B	LD	HL,MENH	02EC ED47	LD I,A
0297	C31E02	JP	ROTSa	02EE C31E02	JP ROTSA
ALTERA REGISTRADOR H		0330 00 0331 01 0332 01 0333 01 0334 01 0335 01 0336 02 0337 01 0338 01 0339 00 0340 F5 0341 0F 0342 2F 0343 17		0330 00 0331 01 0332 01 0333 01 0334 01 0335 01 0336 02 0337 01 0338 01 0339 00 0340 F5 0341 0F 0342 2F 0343 17	
029A	21FB0B	ALTH	LD HL,MENH	02F1 01F20B RIV	LD BC,0BF2H
029D	78	LD	A,B	02F4 3EC3	LD A,C3H
029E	ED6F	RLD		02F6 02	LD (BC),A
0290	C31E02	JP	ROTSa	02F7 03	INC BC
ALTERA REGISTRADOR H		02F8 21EE0B		0330 9F 033E C8 033F E2 0340 F1 0341 0F 0342 2F 0343 17	
02F9	21FB0B	LD	HL,MENH	02F9 7E	LD A,(HL)
02FD	03	INC	BC	02FC 02	LD (BC),A
02FE	23	INC	HL	02FD 03	INC BC
02FF	7E	LD	A,(HL)	02FE 23	INC HL

### Alguns testes práticos

O primeiro teste do sistema deve ser, de preferência, o menor programa possível, como a instrução HLT, que deixa o micro na condição de *halt*:

#### TECLA DISPLAY OBSERVAÇÕES

[RS]	NEST OR	
[P]		
[0800]	0800	
[L]	0800 XX	XX indica valor anterior
[76]	0800 76	76 é o código de máquina da instrução HLT
[R]	.....	acendem apenas os pontos decimais

Outro programa curto: o retorno à rotina de *reset*, utilizando somente a instrução *jump* incondicional:

#### TECLA DISPLAY OBSERVAÇÕES

[RS]	NEST OR	
[P]		
[0800]	0800	
[L]	0800 XX	
[C3]	0800 C3	1º byte da instrução
[+1]	0801 XX	
[00]	0801 00	parte baixa do endereço do salto
[+1]	0802 XX	
[00]	0802 00	parte alta do endereço do salto
[-1][-1]	0800 C3	colocação de endereço para partida
[R]	NEST OR	retorno à mensagem de <i>reset</i>

Programa para o uso de exame de registradores. Carrega o registrador HL com 1234H, examina os conteúdos e faz a retificação:

#### TECLA DISPLAY OBSERVAÇÕES

[RS]	NEST OR	
[P]		
[0800]	0800	
[L21]	0800 21	coloca 1º byte da instrução LD HL, dddd
[+134]	0801 34	byte menos significativo
[+112]	0802 12	byte mais significativo
[+1C3]	0803 C3	salta p/ início programa JP end
[+100]	0804 00	
[+100]	0805 00	
[P08002]	NEST OR	roda o programa
[ER]	A = XX	
[+1+1+1]	H = 12	passa por A, B, C, D e E
[+1]	L = 34	

Um programa para teste de IV; ao ser acionado o NMI (*hardware*), faz o processador ficar em HLT\*:

#### TECLA DISPLAY OBSERVAÇÕES

[RS]	NEST OR	
[P]		
[0900]	0900	
[IV]	0900	coloca o vetor em 0900
[P]		
[0900]	0900	
[L]	0900 XX	
[76]	0900 76	coloca o HLT
[RS]	NEST OR	
	.....	aciona NMI ( <i>hardware</i> )
	.....	coloca o pino 17 da CPU no nível lógico 0

Após essa seqüência de testes, você já poderá começar a fazer seus programas, pois o Nestor já está funcionando perfeitamente.

## Programas aplicativos

Com o traquejo conferido pelos 4 programas de teste já sugeridos, você tem condições agora de passar a um nível mais ambicioso; tente, então, mais estes quatro programas, reunidos em grau crescente de complexidade. As indicações referentes às teclas e ao *display* foram eliminadas, pois devem ser desnecessárias daqui para a frente.

Nos três primeiros programas, incluímos comentários detalhados, principalmente com relação à utilização das sub-rotinas da biblioteca de software existente no programa monitor. Quanto à utilização do visor, a figura 2 apresenta a palavra dos segmentos, juntamente com as formatações mais importantes. Sua aplicação prática será vista com mais clareza nos próprios programas.

### 1. Escrever uma palavra de seis dígitos no visor

endereço	instrução	label	mnemônico	comentários
cód.máq.				
0800	211008	início	LD HL,	utiliza o par HL como indicador no endereço inicial da tabela de caracteres a serem mostrados
0803	CD9C00		CALL REST	chama a rotina de restauração mandando p/ o <i>display</i> os 6 caracteres, a partir do endereço apontado por HL
0806	C30008		JP INÍCIO	salta para o início, para manter a informação no visor
0810	C8	tabela	DFB	N para escolher
0811	86	de	DFB	E a frase, é pre-
0812	92	carac-	DFB	S ciso utilizar os
0813	87	teres	DFB	T códigos da fi-
0814	C0		DFB	O gura 2
0815	CE		DFB	R

### 2. Circulação de palavras

Este programa faz circular, por tempo indefinido, no *display*, uma frase da direita para a esquerda.

endereço	instrução	label	mnemônico	comentários
cód.máq.				
0800	012008	inicio	LD BC, BMD	endereço inicial buffer mensagem do <i>display</i>

endereço	instrução	label	mnemônico	comentários
cód.máq.				
0803	1620	L2	LD D, TEMPO	fixa tempo (velocidade avanço visor)
0805	03		INC BC	avança buffer da mensagem
0806	79		LD A,C	carrega A c/ final do buffer
0807	FE36		CP 36H	compara c/ final da mensagem
0809	CA0008		JP Z,INÍCIO	recicla, se chegou ao final
080C	60	L1	LD H,B	carrega HL comindicador do buffer
080D	69		LD L,C	p/ utilizar rotina de restauração
080E	C5		PUSH BC	salva par BC
080F	CD9C00		CALL REST	chama rotina p/ <i>display</i>
0812	C1		POP BC	restaura BC
08013	15		DEC D	decrementa a constante de tempo
0814	C20C08		JP NZ,L1	não sendo zero, continua na mensagem
0817	C30308		JP L2	sendo zero, evoluí de 1 dígito
0820	FF	tabela	DFB	
	FF	de	DFB	
	FF	carac-	DFB	
	FF	teres	DFB	
	FF		DFB	
	FF		DFB	
	86		DFB	
	C1		DFB	
	FF		DFB	
	FF		DFB	
	86		DFB	
	C1		DFB	
	FF		DFB	
	FF		DFB	
	92		DFB	
	CO		DFB	
	C1		DFB	
	FF		DFB	
	FF		DFB	
	C8		DFB	
	86		DFB	
	92		DFB	
	87		DFB	
	CO		DFB	
	CE		DFB	
	FF		DFB	
	FF		DFB	
083A	FF		DFB	
	FF		DFB	

**Observações:** As seis palavras iniciais não determinam dígito algum, para que o visor comece apagado. Para mudar a frase, basta usar a tabela de códigos da figura 2. Para mudar a extensão da mensagem, basta alterar a locação 0808 com o comprimento da nova mensagem. Para mudar a velocidade, é só alterar a locação 0804 com o tempo desejado.

### 3. Contador de 0 a 9 (no 6º dígito)

endereço	instrução	label	mnemônico	comentários
cód.maq.				
0800	210009	início	LD HL,MBD	coloca HL como indicador do buffer do display
0803	36FF	A1	LD (HL),FF	apaga o dígito
0805	23		INC HL	ajusta próximo dígito
0806	7D		LD A,L	verifica se todo o
0807	FE06		CP 06H	visor foi apagado
0809	C20308		JP NZ,A1	retorna, se não apagou
080C	210609	A2	LD HL, MENCON	se apagou,arma memória de contagem
080F	3600		LD (HL),00	zera essa memória
0811	210609	A4	LD HL, MENCON	arma memória de contagem
0814	34		INC (HL)	incremente memória de contagem
0815	1620		LD D, TEMPO	tempo que determina velocidade contagem
0817	7E		LD A,(HL)	carrega acumulador c/a contagem
0818	FE0A		CP 0AH	verifica se contagem chegou a 10
081A	CA0C08		JP Z,A2	salta se é 10, p/zerar contador e reciclar
081D	CD8601		CALL FORMAT	formata o conteúdo de A em 7 segmentos
0820	210509		LD HL,DIG6	HL é indicador do último dígito do visor
0823	77		LD (HL),A	dá o valor formatado a esse dígito
0824	210009	A3	LD HL, MENBUF	HL é indicador do buffer do display
0827	CD9C00		CALL REST	contagem no último dígito e o restante apagado
082A	15		DEC D	decremente tempo
082B	C22408		JP NZ,A3	se não acabou, a contagem não evolui
082E	C31108		JP A4	se acabou, incrementa o dígito

**Observações:** Para alterar o tempo, basta mudar o valor do registrador D na locação 0816. Como sugestão para o leitor, se o programa for ampliado, é possível implementar um relógio digital; basta continuar a mesma análise para os demais dígitos do visor.

### 4. Jogo da forca

Neste caso, devido à extensão do programa, dispensamos os comentários. Para jogar, deve-se rodar o programa a partir de 0800, que fará circular no display o título "jogo da forca"; com isso, o jogo já foi inicializado. A palavra oculta deve ser introduzida com o auxílio dos códigos mostrados na figura 2, os quais devem ser teclados de acordo com os caracteres desejados.

Em outras palavras, a cada duas teclas acionadas, aparecerá o caractere correspondente no display, até que tenham sido completados os 6 dígitos. A palavra introduzida permanece por mais uns 3 segundos e depois se apaga.

Nas tentativas obedece-se o mesmo processo, entrando com os códigos correspondentes. No caso de acerto, na posição onde deveria estar o caractere ele surge e permanece. Em caso de erro, o visor mostrará a mensagem "erro nº X", até o máximo de 7 erros; nesse limite, o jogo termina com a mensagem "perdeu". Boas jogadas.

endereço	instrução	label	mnemônico
cod. máq.			
0800	010009	início	LD BC,EIT
0803	1620	A4	LD DE,20
0805	03		INC BC
0806	79		LD A,C
0807	FE16		CP 16H
0809	CA0008		JP Z,INÍCIO
080C	IE06	A3	LD E,06H
080F	3E01		LD A,01H
0810	60		LD H,B
0811	69		LD L,C
0812	CDA900	A2	CALL TEMPO
0815	08		EX AF
0816	DB01		IN, TECLADO
0818	FE00		CP 00H
081A	C22A08		JP NZ,A1
081D	08		EX AF
081E	87		ADD A,A
081F	1D		DEC E
0820	C21208		JP NZ,A2
0823	15		DEC D
0824	C20C08		JNZ, A3
0827	C30308		JP, A4
082A	08	A1	EX AF
082B	214909		LD HL,MENCON 1
082E	3600		LD(HL), 00
0830	215609		LD HL, MENCON 2
0833	3600		LD (HL), 00
0835	215909		LD HL, MENE
0838	3600		LD (HL), 00
083A	D9		EXX
083B	215009		LD HL, PONT
083E	D9		EXX
083F	214A09		LD HL, ZM
0842	36FF	A5	LD (HL), FF
0844	23		INC HL
0845	7D		LD A, L
0846	FE56		CP 56H
0848	C24208		JP NZ,A5
084B	215009	A6	LD HL, A6
084E	CD7009		CALL RIP
0851	78		LD A, B

endereço	instrução cód.máq.	label	mnemônico	endereço	instrução cód.máq.	label	mnemônico
0852	FE00		CP 00H	08DE	79		LD A, C
0854	CA4B08		JP Z, A6	08DF	C3D208		JP A13
0857	215009	A7	LD HL, DIG 1	08E2	7B	A12	LD A, E
085A	CD9C00		CALL REST	08E3	FE00		CP 00
085D	7A		LD A, D	08E5	CA9009		JP Z, ERROU
085E	D303		OUT, PDIG	08E8	C39308		JP A9
0860	DB01		IN A, 01	08EB a 08EF não são usadas			
0862	FE00		CP 00H	08F0	215709	A13	LD HL TAB
0864	C25708		JP NZ, A7	08F3	78		LD A, B
0867	CDBD00		CALL AJCOL	08F4	ED6F		RLD
086A	CDD800		CALL AJTEC	08F6	7E		LD A, (HL)
086D	215609		LD HL, MENCON	08F7	D9		EXX
0870	7E		LD A, (HL)	08F8	77		LD (HL), A
0871	CB47		BIT 0, A	08F9	23		INC HL
0873	C2F008		JP NZ, A13	08FA	D9		EXX
0876	215709		LD HL, MET	08FB	C37A08		JP, A14
0879	70		LD (HL), B	08FE e 08FF não são usadas			
087A	215609	A14	LD HL, MENCON	0970	0600	RIP	LD D, 00
087D	34		INC (HL)	0972	1E06		LD E, 06
087E	7E		LD A, (HL)	0974	3E01		LD A, 01
087F	FE0C		CP OCH	0976	CD9000	A15	CALL VARR
0881	C24B08		JP NZ, A6	0979	79		LD A, C
0884	116000		LD DE, TEMPO	097A	87		ADD A, A
0887	215009	A8	LD HL, BUF	097B	1D		DEC E
088A	CD9C00		CALL REST	097C	C27609		JP NZ, A15
088D	B		DEC DE	097F	C9		RET
088E	7A		OR E	0980	4F	ACERTOU	LD C, A
088F	B3		JP NZ, A8	0981	7D		LD A, L
0893	214A09	A9	LD HL, BUFT	0982	E5		PUSH HL
0896	CD7009		CALL RIP	0983	D606		SUB 06
0899	78		LD A, B	0985	6F		LD L, A
089A	FE00		CP 00	0986	71		LD (HL), C
089C	CA9308		JP Z, A9	0987	E1		POP HL
089F	214A09	A10	LD HL, MET	0988	23		INC HL
08A2	CD9C00		CALL REST	0989	79		LD A, C
08A5	7A		LD A, D	098A	1E01		LD E, 01
08A6	D303		OUT PDIG	098C	C3D708		JP, A16
08A8	DB01		IN A, 01	098F não é usado			
08AA	FE00		CP 00	0990	215909	ERROU	LD HL, MENCON
08AC	C29F08		JP NZ, A10	0993	34		INC (HL)
08AF	CDBD00		CALL AJCOL	0994	7E		LD A, (HL)
08B2	CDD800		CALL AJTEC	0995	216009		LD HL, TFE
08B5	214909		LD HL, MENCON	0998	85		ADD A, L
08B8	CB46		BIT 0, (HL)	0999	6F		LD L, A
08BA	C2C508		JP NZ, A11	099A	7E		LD A, (HL)
08BD	34		INC (HL)	099B	215A09		LD HL, BUFE
08BE	215809		LD HL, MET	099E	3686		LD (HL), COD
08C1	70		LD (HL), B	09A0	23		INC HL
08C2	C39308		JP A9	09A1	36CE		LD (HL), COD
08C5	34	A11	INC (HL)	09A3	23		INC HL
08C6	78		LD A, B	09A4	36CE		LD (HL), CEH
08C7	215809		LD HL, MET	09A6	23		INC HL
08CA	ED6F		RLD	09A7	36C0		LD (HL), COH
08CC	7E		LD A, (HL)	09A9	23		INC HL
08CD	1E00		LD E, 00	09AA	36FF		LD (HL), FFH
08CF	215009		LD HL, MENE	09AC	23		INC HL
08D2	BE	A13	CP (HL)	09AD	77		LD (HL), A
08D3	CA8009		JP Z, ACERTOU	09AE	116000		LD DE, TEMPO
08D6	23		INC HL	09B1	215A09	A16	LD HL, MENE
08D7	4F	A16	LD C, A	09B4	CD9C00		CALL REST
08D8	7D		LD A, L	09B7	1B		DEC DE
08D9	FE56		CP 56H	09B8	7A		LD A, D
08DB	CAE208		JP Z, A12	09B9	B3		OR E

endereço	instrução cód.máq.	label	mnemônico	endereço	instrução cód.máq.	label	mnemônico
09BA	C2B109		JP NZ, A16	09D7	FE32		CP 32H
09BD	215909		LD HL, NDV	09D9	CAD009		JP Z, PERDEU
09C0	7E		LD A, (HL)	09DC	60	A17	LD H, B
09C1	FE07		CP 07H	09DD	69		LD L, C
09C3	CAD009		JP Z, PERDEU	09DE	C5		PUSH BC
09C6	C39308		JP, A9	09DF	CD9C00		CALL REST
09D0	012009	PERDEU	LD BC, MDP	09E2	C1		POP BC
09D3	1620	A18	LD D, TEMPO	09E3	15		DEC D
09D5	03		INC BC	09E4	C2DC09		JP NZ, A17
09D6	79		LD A, C	09E7	C3D309		JP, A18

Tabelas de mensagens

**"jogo da força"**

endereço	cód. máq.
0900	FF
0901	FF
0902	FF
0903	FF
0904	FF
0905	FF
0906	E1
0907	C0
0908	82
0909	C0
090A	FF
090B	FF
090C	A1
090D	88
090E	FF
090F	FF
0910	8E
0911	C0
0912	CE
0913	C6
0914	88
0915	a
091A	FF

**"perdeu"**

endereço	cód. máq.
0920	FF
0921	FF
0922	FF
0923	FF
0924	FF
0925	FF
0926	C7
0927	86
0928	FF
0929	FF
092A	A1
092B	88
092C	C8
092D	C6
092E	86
092F	
0934	

**"erro"**

endereço	cód. máq.
0961	F9
0962	A4
0963	B0
0964	99
0965	92
0966	82
0967	F8

**Conclusão** — Este foi um trabalho feito com muito empenho. Acreditamos que todos os leitores — tenham ou não montado seu próprio micro — usufruiram alguma coisa dele, algo que poderá ser de grande utilidade numa sociedade cada vez mais informatizada.

Nossa objetivo foi o de transmitir informações àqueles que gostariam de aprender realmente a "mexer" em

microcomputadores e sem gastar muita verba com isso. Deixamos isso claro quando anunciamos que o Nestor é um micro dirigido a quem quer aprender a projetar sistemas e não apenas operá-los — ou seja, foi projetado especificamente para os estudantes, profissionais ou autodidatas de eletrônica.

Assim, com o Nestor e um curso de *Assembler* (ou mesmo um bom livro

que trate desse tema), qualquer montador poderá ter em pouco tempo um bom conhecimento dessa área. Sem falar no mercado de trabalho em informática, que ainda resiste à crise e pode oferecer boas oportunidades, esse pequeno micro pode significar mais um passo no domínio da tecnologia por brasileiros e para brasileiros — um importante fator de independência industrial e de pesquisa.

NESTOR, A BASE DE TODOS  
OS COMPUTADORES — 1º SUPLEMENTO

# Interligações com o mundo exterior

O Nestor não é apenas um computador didático. Com os portais de entrada/saída de que dispõe, pode ser transformado num controlador lógico. Ou num voltímetro, como veremos aqui



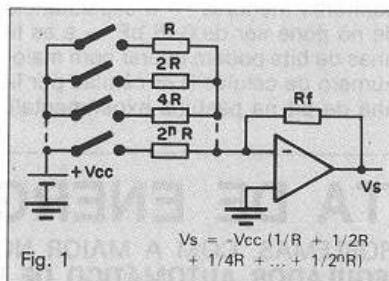
**C**omo já havíamos comentado, o objetivo de nosso projeto não abrange o desenvolvimento de sistemas voltados para a área de micros pessoais ou terminais inteligentes. Desde o início, nossa idéia foi voltar o Nestor para o treinamento de técnicos e engenheiros, além de pequenas aplicações relacionadas com a eletrônica.

No último artigo da série básica propusemos alguns programas simples, a fim de familiarizar o usuário com a programação do Nestor. Sua utilização, porém, não precisa ficar só nisso. Nossa preocupação, neste primeiro suplemento sobre o micro, será mostrar e demonstrar, através de circuitos práticos, como "interfacear" o Nestor com dispositivos externos, fazendo-o executar uma série de tarefas.

Pelo fato do Nestor possuir portais de entrada e saída, assemelha-se bastante a um controlador lógico digital, sugerindo uma extensa gama de aplicações; ele é capaz, por exemplo, de receber dados paralelos, processá-los e gerar sinais para controle e/ou sinalização. No caso do controle de velocidade de um motor CC, digamos, teríamos a rotação e a corrente de armadura como entrada e a tensão como saída.

Existem, porém, vários tipos de sistemas de controle, para as mais variadas aplicações, e uma explanação mais profunda sobre o assunto tornaria um artigo inteiro. No final deste artigo falaremos um pouco mais sobre esse assunto.

Este suplemento está dividido em duas partes básicas, a primeira tratan-



do de entradas e saídas digitais e a segunda apresentando dois circuitos práticos: um conversor A/D e outro, D/A, para permitir a manipulação de sinais analógicos através do Nestor.

**A parte digital** — O micro conta com um portal para 8 entradas digitais, utilizando dois integrados (Cl13 e Cl14, no canto inferior esquerdo da placa) que são simples buffers 74LS126, com saídas *tristate* e entradas compatíveis com TTL. Os sinais ai aplicados, contudo, poderão vir de contatos de relé, coletores abertos com vários níveis de alimentação etc. É sempre necessário, porém, fazer o interfaceamento adequado, para proteger o Nestor — utilizando acopladores óticos, por exemplo. Esse tipo de proteção atua contra ruídos e sobretensões.

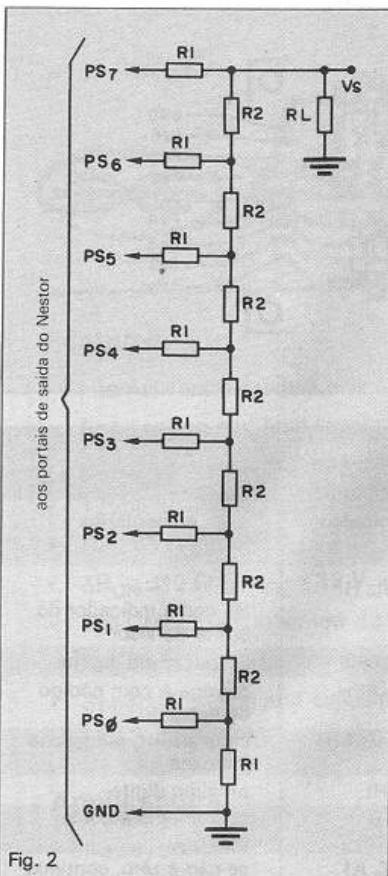
Para efetuar a aquisição dessas 8 entradas, utiliza-se a instrução IN A, (02H), cujo código é DB02, que coloca esse dado no acumulador. Para detectar se as entradas estãoativas ou não, há diversas formas; pode-se usar, por exemplo, as instruções de mani-

pulação de bits do Z-80, capazes, entre outras coisas, de testar um bit isoladamente com uma instrução BIT n, REG — o que afeta o bit Z (zero) do registrador F (*status flag*), facilitando bastante a programação.

O sistema, cujas saídas deverão ser monitoradas, poderá requerer ou não leituras síncronas. No caso assíncrono, efetua-se a leitura por varredura (ou periódica); no síncrono, costuma-se empregar a entrada de interrupção (NMI) do Nestor como entrada de sincronismo. Um bom exemplo de leitura por varredura é dado no programa relativo ao conversor A/D, mais adiante, neste mesmo artigo.

O Nestor dispõe também de 8 saídas digitais, tomando mais dois Cls (Cl11 e Cl12), estes do tipo 74LS75, que são biestáveis com trava (*Latch*) e sensíveis a nível. Esses dois integrados também são compatíveis com TTL e podem ser usados na excitação de relés eletromecânicos (com transistores excitadores), relés de estado sólido (por meio de fotoacopladores) ou mesmo circuitos de coletor aberto. Poderão servir ainda para sinalização, acionando LEDs ou lâmpadas de painel (através de transistores). Dependendo da utilização, é sempre necessário providenciar um interfaceamento adequado, visando a proteção do micro.

Para gerar uma saída, utiliza-se instruções do tipo OUT, tal como OUT (04H),A, que coloca o conteúdo do acumulador na porta de endereços 04H. Essa informação ficará retida nos biestáveis, para que o processa-



dor continue seu programa, mantendo as saídas ativadas.

**Parte analógica: o conversor D/A** — Por meio de interfaces adequadas, o Nestor tem a possibilidade de receber grandezas analógicas e também de fornecer esse tipo de sinal, respectivamente para aquisição de dados vindos de sensores e gerar saídas analógicas de controle.

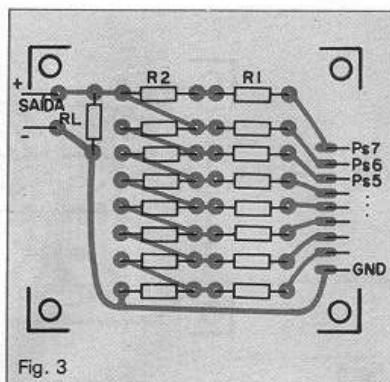
A função do conversor digital/análogo é a de entregar uma informação analógica proporcional a dados digitais. O sinal de entrada (digital) poderá ser binário puro, BCD (binário codificado em decimal) etc.

O importante, nesse caso, é saber que a quantidade binária apresentada pelo computador é a representação de um valor fracionário, que será multiplicado por uma tensão de referência. O valor decimal pode ser expresso por

$$\sum a_n \cdot 2^n$$

onde  $n$  é um número inteiro e  $a_n$  assume o valor 0 ou 1.

Portanto, é preciso montar um somador com as parcelas proporcionais a  $n$ , ou seja, com grandes contribuições de bits mais significativos e pequenas dos menos significativos. Es-



### Relação de componentes

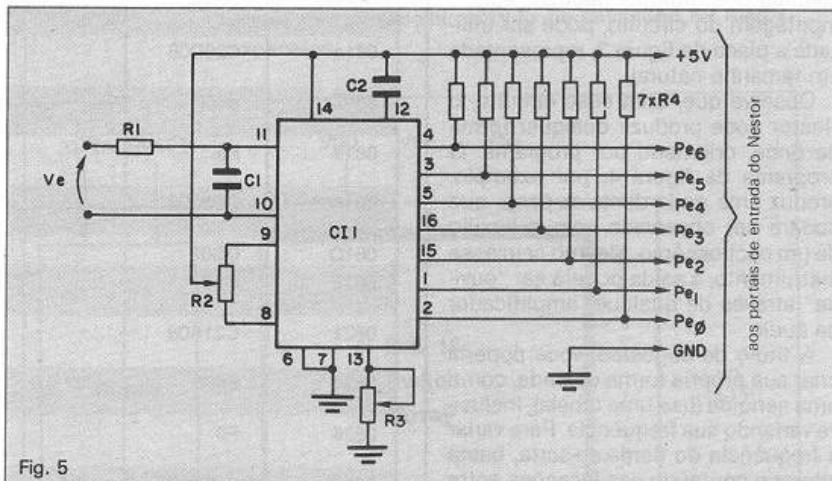
**CONVERSOR D/A**  
R1-20 kΩ - 1/4 W (filme metálico)  
R2-10 kΩ - 1/4 W (filme metálico)  
RL-100 kΩ - 1/8 W

**CONVERSOR A/D**  
R1, (7x)R4- 10 kΩ - 1/8 W  
R2- 50 kΩ - helipot (multivoltas)  
R3- 10 kΩ - helipot (multivoltas)  
C1- 0,1 μF (poliéster)  
C2- 0,22 μF (poliéster)  
CI1- CA3162

### Programa dente-de-serra (com conversor D/A)

endereço	código de máquina	label	mnemônico	comentários
0800	97	início	SUB A	zera acumulador
0801	D304	A1	OUT (Ps),A	coloca o valor no portal de saída
0803	215000		LD HL, (TEMPO)	fixa valor de tempo (freqüência)
0806	2B	A2	DEC HL	decrementa valor de tempo
0807	C20608		JPNZ, A2	se não é zero, continua a decrementar
080A	3C		INC A	se zerou, incrementa o acumulador
080B	C30108		JP, A1	retorna para mandar novo valor

Fig. 4



se tipo de conversão pode ser feito simplesmente através de uma malha resistiva, seja com resistores proporcionais ou com uma rede R-2R (*ladder* ou escada).

O primeiro tipo é conceitualmente mais simples, pois os resistores possuem pesos específicos (um resistor por bit), exibindo valores crescentes de  $R$ ,  $2R$ ,  $4R$ ,  $2^R$ , a fim de produzir as

várias correntes fracionárias. Sua visualização fica mais fácil através da figura 1, onde a saída digital foi substituída por chaves.

Como se pode notar, esse tipo de conversor depende muito do valor individual dos resistores. E a propagação através deles torna-se um problema quando é preciso representar um número elevado de bits.

Como alternativa, costuma-se adotar a rede R-2R — que, inclusive, é bem mais utilizada. Estamos apresentando, na figura 2, um circuito pronto que já pode ser acoplado ao Nestor, com as vantagens de dispensar alimentação e exhibir um ótimo desempenho. A limitação desse projeto está na excursão do sinal analógico, que é de 0,3 a 4 V para uma variação de 0 a 256, em decimal, ou de 0 a FF, em hexadecimal. Não acrescentamos um amplificador à rede por saber que essa é uma parte que vai depender diretamente do circuito a que o Nestor será acoplado.

Seu princípio de operação é bastante simples. Temos, sempre em série a cada bit, um resistor de valor  $2R$  e entre eles (na linha de soma), outro de valor  $R$  — forma-se, assim, uma rede tipo  $\pi$ . Isso faz com que a impedância, em qualquer nó de soma, seja sempre a mesma, fazendo com que a corrente produzida por um bit passe para a ramificação seguinte com metade de seu valor. Essa corrente continuará sendo dividida por 2, até completar todos os nós da malha.

A resolução desse tipo de conversor é dada pelo número de bits utilizado; para 8 bits, essa resolução é de 1/256, valor que consideramos adequado ao nosso projeto. A precisão, por outro lado, é determinada pela tolerância dos resistores; aconselhamos, por isso, a utilização de componentes de filme metálico, os quais podem ser encontrados com relativa facilidade nos grandes centros. Para a montagem do circuito, pode ser utilizada a placa da figura 3, representada em tamanho natural.

Observe que, com esse circuito, o Nestor pode produzir qualquer forma de onda, orientado por programa. O programa da figura 4, por exemplo, produz uma onda dente-de-serra, que poderá ser observada com o auxílio de um osciloscópio. Mesmo sem esse instrumento, a saída poderá ser "ouvida" através de qualquer amplificador de áudio.

A título de sugestão, você poderia criar sua própria forma de onda, como uma senóide (use uma tabela), inclusive variando sua frequência. Para variar a frequência do dente-de-serra, basta alterar o conteúdo das locações entre 0804 e 0805.

**Um conversor analógico/digital** — Como o próprio nome diz, esse circuito deve converter tensões analógicas em números binários. São válidas, para ele, as mesmas considerações de resolução feitas para o conversor D/A. Existem conversores A/D para 8, 10 ou 12 bits, de alta velocidade, e de vários tipos de operação, tais como: largura

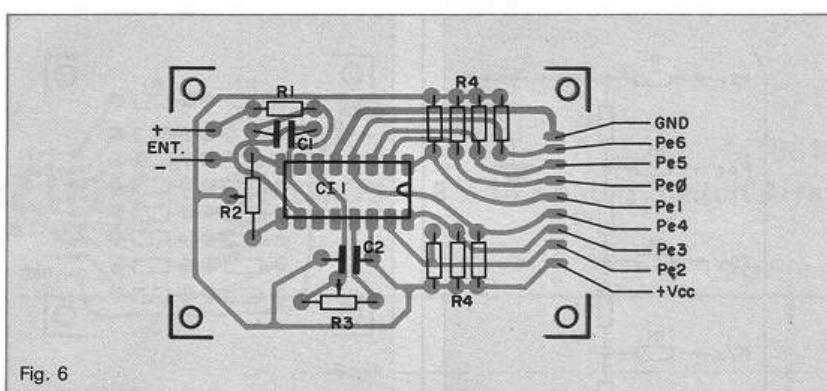


Fig. 6

Programa voltímetro (com conversor A/D)				
endereço	código de máquina	label	mnmônico	comentários
0800	216008	inicio	LD HL, (1º dig.)	HL como indicador do buffer do visor
0803	0606		LD B,06H	B com nº de dígitos
0805	3EFF		LD A,FFH	carrega A com código de apagado
0807	77	A1	LD (HL),A	coloca FFH na locação indicada
0808	23		INC HL	próximo dígito
0809	05		DEC B	decrementa a contagem
080A	C20708		JPNZ, A1	se não é zero, continua para outros dígitos
080D	CD5308	A2	CALL AMOST	se é zero manda p/ display
0810	DB02		IN A, (02)	lê valor do portal
0812	CB77		BIT 6	testa bit relativo à 3ª coluna (LSD)
0814	C20D08		JPNZ, A2	se não é zero, continua p/ restante do display
0817	E60F		AND 0F	se é zero, mascara os 4 bits de maior ordem
0819	F5		PUSH AF	preserva o valor em BCD na pilha
081A	CD5308	A3	CALL AMOST	continua com o restante do display
081D	DB02		IN A,(02H)	lê portal de entrada
081F	CB6F		BIT 5	testa bit relativo à 2ª coluna
0821	C21A08		JPNZ, A3	se não é zero, continua no restante do display
0824	E60F		AND 0F	se é zero, mascara os 4 bits de maior ordem
0826	F5		PUSH AF	preserva na pilha o acumulador
0827	CD5308	A4	CALL AMOST	restaura o display
082A	DB02		IN A, (02H)	lê portal de entrada
082C	CB67		BIT 4	testa bit relativo à 1ª coluna (MSD)
082E	C22708		JPNZ, A4	se não é zero, continua no restante do display
0831	E60F		AND 0F	se é zero, mascara os 4 bits de maior ordem
0833	F5		PUSH AF	preserva na pilha
0834	216508		LD HL, (3º dig.)	HL como indicador do 3º dígito

endereço	código de máquina	label	mnemônico	comentários
0837	0603		LD B,03H	B com nº de dígitos a serem usados
0839	F1	A5	POP AF	restaura valor BCD na pilha
083A	CD8601		CALL FORMAT	formata esse valor para 7 segmentos
083D	77		LD (HL),A	coloca o valor no buffer correspondente
083E	2B		DEC L	indicador para próximo dígito
083F	05		DEC B	decrementa contagem
0840	C23908		JPNZ, A5	se não chegou a zero, volta p/ dígito seguinte
0843	3A6308		LD A,(3º dig.)	se é zero, coloca no acumulador valor do 3º dígito
0846	FE88		CP 88H	compara com o código 88 para testar valor negativo
0848	C25008		JPNZ, A6	se não transbordou, retorna ao inicio
084B	3EBF		LD A, BFH	se transbordou, coloca o código (-) em A
084D	326308		LD (3º dig.), A	coloca o dígito no buffer
0850	C30D08	A6	JP A2	início
0853	216008	AMOST	LD HL(BUFF)	endereço inicial do buffer do visor
0856	CD9C00		CALL REST	sub-rotina restauração (Nestor)
0859	C9		RET	retorno

**Nota:** Este programa produz, no visor do Nestor, o valor direto de tensão, em milivolts, aplicado na entrada do CA3162. No caso de transbordo positivo, o visor indicará um código não válido (bbb); para o negativo, o código será (-AA). O conversor usado é adequado apenas para sistemas de controle lento, como temperatura, por exemplo.

Fig. 7

de pulso, contagem de rampa (simples ou dupla), aproximação sucessiva, entre outros.

Os conversores A/D são mais caros que os D/A e seu custo também está relacionado com sua resolução e velocidade de conversão. Mas falar sobre todos os tipos de conversores tomaria muito espaço; por isso, vamos propor diretamente um circuito prático, que possa ser utilizado com o Nestor.

O circuito proposto usa um único circuito integrado da RCA, o CA3162, encapsulado em 16 pinos e já existente no mercado nacional. Em aplicações normais, ele é utilizado em conjunto com o CA3161, um decodificador/excitador de 7 segmentos. Aqui ele foi usado isoladamente, por apresentar inúmeras vantagens: baixo custo, alimentação simples (+ 5 V), entrada de modo diferencial, precisão equivalente à de um conversor de 10 bits (pois trabalha multiplexado entre -99 a 999,

em BCD) e facilidade de processamento das informações (já que os dados entram diretamente em BCD).

Sua única desvantagem, embora seja relevante em alguns casos, reside no relativamente lento tempo de conversão, que no CA3162 é de 10 milissegundos; além disso, há também a saída multiplexada, exigindo um programa mais extenso na aquisição de dados.

Para ser interligado ao Nestor, o circuito do conversor deve ser o da figura 5 — que não prevê escala na entrada, permanecendo então a original do CI, que abrange de -99 a + 999 mV. A sugestão de placa para ele pode ser vista na figura 6, em escala natural; aconselhamos a montagem do integrado através de soquete.

Na figura 7 temos um programa ilustrativo, que faz a aquisição e o processamento dos dados analógicos e os apresenta no *display* do micro diretamente em milivolts — transformado o conjunto Nestor-conversor em um voltímetro digital. Da mesma forma, poderíamos ter medidas de corrente, resistência, temperatura e outras grandezas, desde que fossem usados os sensores e circuitos adequados.

**Algumas sugestões** — Já apresentamos circuitos e programas que permitem uma ampliação dos usos do Nestor, todos testados no próprio micro. Vamos agora sugerir duas aplicações simples, baseadas na vida prática e de fácil simulação na bancada: um controle *on-off* (liga-desliga) e outro proporcional.

Vamos imaginar uma máquina hipotética, que deve furar blocos de madeira numa determinada posição e com a máxima eficiência possível; o sistema

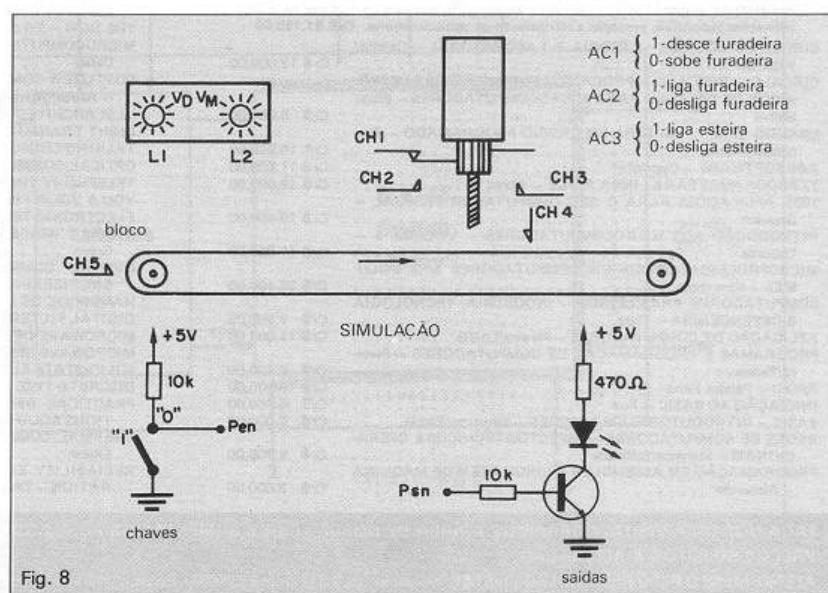


Fig. 8

já em posição está representado na figura 8. A entrada dos blocos é feita por um operador humano, que deve colocá-los apenas quando acender L1 (lâmpada verde) e evitar de introduzi-los com a esteira em movimento.

Temos, como entradas, as informações fornecidas pelas cinco chaves do sistema, assim distribuídas: CH1 — chave de início de curso da furadeira; CH2 — chave de fim de curso da furadeira (sem que a broca tenha tocado o bloco); CH3 — chave de fim de curso, dependente da profundidade do furo; CH4 — chave sensora da posição do bloco; e CH5 — chave sensora que indica a introdução dos blocos pelo operador (se o operador colocar um bloco sem permissão, as saídas devem ser zeradas e permanecer nessa condição).

Como saídas, consideramos os controles AC1 (desce ou sobe furadeira), AC2 (liga ou desliga furadeira) e AC3 (liga ou desliga esteira), além de L1 e L2 (sinalização para o operador). No Nestor essa simulação pode ser

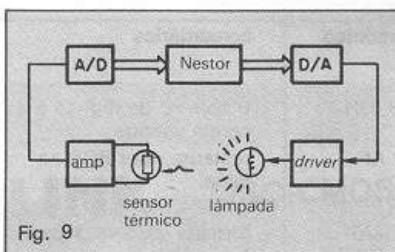


Fig. 9

implementada com facilidade, utilizando-se chaves comuns no lugar de CH1/CH5, ligadas diretamente aos portais de entrada, e cinco LEDs fazendo as vezes de saídas e ligados aos portais correspondentes. Na própria figura 8 estão esquematizados os circuitos para as chaves e os LEDs.

A segunda sugestão é a de um controle proporcional de temperatura, empregando os conversores A/D e D/A (que podem ser aqueles aqui expositos ou quaisquer outros), uma lâmpada como saída (e um transistor como excitador da mesma) e na entrada um sensor adequado, ligado convenientemente.

O objetivo é manter constante a temperatura sobre o sensor, de acordo com a montagem representada na figura 9. Essa montagem tem um efeito muito interessante, pois é possível sentir a atuação do controle (através

do brilho da lâmpada), ao se produzir certas perturbações no sistema, tais como o distanciamento e aproximação entre a lâmpada e o sensor térmico. Vê-se claramente as buscas variações de brilho da lâmpada, ao procurar manter constante a temperatura sobre o sensor.

Como se pode ver, esse programa já não é tão simples como os demais, pois envolve também alguns conceitos de controle e servomecanismos. É difícil, para nós, abordar esse assunto específico nesta seção da revista. Mesmo assim, fica sugerida essa implementação para os mais adiantados, que desejarem "mexer" um pouco mais em controle e automação.

Estamos analisando a possibilidade de continuar com a série de artigos relativos ao Nestor, abordando novas interfaces e até mesmo um gravador de EPROM completo. Pedimos, assim, a colaboração de todos os leitores interessados, para que enviem sugestões através de cartas, dirigidas à nossa redação. Até lá. ●

# Interface para cassete e programação de tons

O Nestor abre ainda mais sua gama de possibilidades: agora pode ter seus programas guardados em fita cassete e produzir trechos musicais

**N**a seqüência de suplementos que visam apresentar o micro Nestor sob todos os seus aspectos — os outros artigos apareceram nos n°s 84, 85, 86 e 88 —, estamos apresentando neste número uma interface para gravador cassete, que permite o armazenamento de programas em fita. Isso facilita bastante a vida do operador, que não precisa digitar um programa toda vez que quiser reutilizá-lo.

Essa interface é interessante tanto para quem ainda vai montar como para quem já montou o computador, já que seu hardware é muito simples e barato, pois consiste apenas de casadores de nível e impedância e aproveita os portais E/S do Nestor.

O maior trabalho, na verdade, é o de carregar a EPROM do micro com o programa de gravação/leitura. É mais um programa para ocupar parte do espaço livre daquela memória (lembre-se que o programa monitor ocupa cerca de 1 kbyte de EPROM e o outro quilobyte ficou vago prevendo a implementação desses aplicativos). Como o programa para cassete ocupa 591 bytes, fica ainda sobrando uma boa área para outras aplicações.

Na parte física da interface foi prevista também a ligação de um alto-falante, que durante a gravação e a leitura fará a sinalização acústica da operação (as freqüências utilizadas são da faixa de áudio, como veremos adiante). Outra função do alto-falante é servir como transdutor do gerador de tons programável, pois o programa possui uma sub-rotina específica para essa função — que também será vista mais adian-

te, juntamente com exemplos práticos.

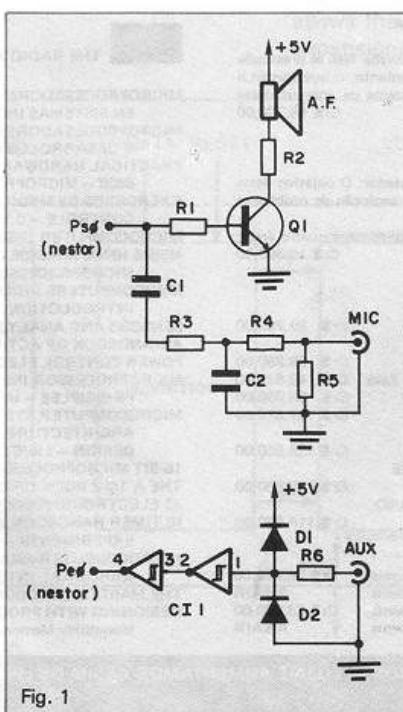
Resumindo as características básicas dessa interface, ela exibe uma velocidade de transmissão de aproximadamente 170 bps (bits por segundo) e permite a inclusão do nome do programa a ser armazenado (de até 4 caracteres hexadecimais), permitindo que na leitura o micro seja carregado com o programa de título igual ao requisitado. Isso representa outra grande vantagem, pois permite que a mes-

ma fita armazene diversos programas e evita preocupações com dados falsos e "sujeiras" da própria fita. Em outras palavras, o Nestor fica esperando pelo programa, cujo título seja igual ao pedido pelo teclado.

A inclusão de mensagens de erro é outra característica interessante do programa. Caso ocorra algum problema na gravação ou leitura — como bits perdidos, por exemplo —, o operador será alertado pelo próprio display do computador, pois foi prevista uma checagem automática das informações.

**Operação: hardware** — Como já dissemos, são poucos os componentes necessários, já que a interface irá apenas "casar" níveis TTL com os de gravação/leitura de gravadores cassete comuns. O circuito completo aparece na figura 1; a etapa de saída tem um transistor como excitador do alto-falante e uma rede resistiva para casamento do nível de gravação com os do micro. A etapa de entrada conta com R6, D1 e D2 para proteção de entrada da interface e com dois inversores Schmitt, a fim de "quadrar" o sinal vin-  
do do gravador e compatibilizá-lo com os níveis TTL.

O volume do gravador deverá ser ajustado na prática, para cada caso específico; em nosso laboratório, por exemplo, o nível 2 (no caso de controles de volume graduados de 0 a 10) apresentou resultados satisfatórios. A placa de circuito impresso sugerida por nós aparece na figura 2; ela engloba os dois circuitos e até mesmo as tomadas fêmeas de entrada e saída, que podem ser do tipo para circuito impresso.



**Software** — O programa de gravação/leitura ocupa, na EPROM, a área compreendida entre os endereços 0350 e 059F, num total de 591 bytes, o que deixa ainda 622 bytes para outras aplicações. Devido à extensão desse programa, tornou-se praticamente impossível a inclusão dos fluxogramas das sub-rotinas e também dos comentários em cada instrução. Portanto, optamos por apresentar essas explicações aqui mesmo, no corpo do artigo. Na obtenção da listagem foi empregado um editor-assembler para TRS-80 seguido de testes, o que tornou o programa completamente isento de erros.

Na figura 3 podemos ver os tipos de formatos utilizados na transmissão de dados ao gravador cassete. Como vemos, são usados dois valores de frequência, ambos situados dentro da faixa de resposta do gravador (1 e 2 kHz). Esses sinais são totalmente gerados por software e, pela análise dos formatos, podemos comprovar a relativa segurança que o sistema oferece.

O formato de bytes, por sua vez, já que a comunicação é assíncrona, possui um bit de partida e outro de parada — que é um dos formatos-padrão para esse tipo de comunicação.

**Comentários do programa** — Eis aqui os pontos de maior interesse do programa de gravação/leitura:

— 0350 a 03FC(*inicialização para o Nestor*) — é um programa que fica em loop e serve como ponto de entrada/saída para operar a interface cassete com o teclado e o *display*, usando identificação de teclas e mensagens de *display* — ou seja, um pequeno monitor que utiliza sub-rotinas do monitor original, de modo a facilitar ao usuário a operação de gravação/leitura. É responsável ainda pelo tratamento dos dados introduzidos via teclado; depois de colocados todos os dados, sai para GFITA ou LFITA.

— 03FF a 0431(*GFITA*) — programa principal de gravação em fita, que chama diversas sub-rotinas para efetuar a gravação, organizando e gerando o *file* completo.

— 0434 a 0438(*CBYTES*) — sub-rotina que calcula o número de bytes no bloco de memória, cujos valores já estão armazenados no *buffer* de gravação. — 0439 a 0440(*SOMA*) — sub-rotina que efetua a soma acima citada. — 0441 a 0453(*PARAM*) — sub-rotina que apanha os valores contidos no *buffer* (endereço fonte e endereço final),

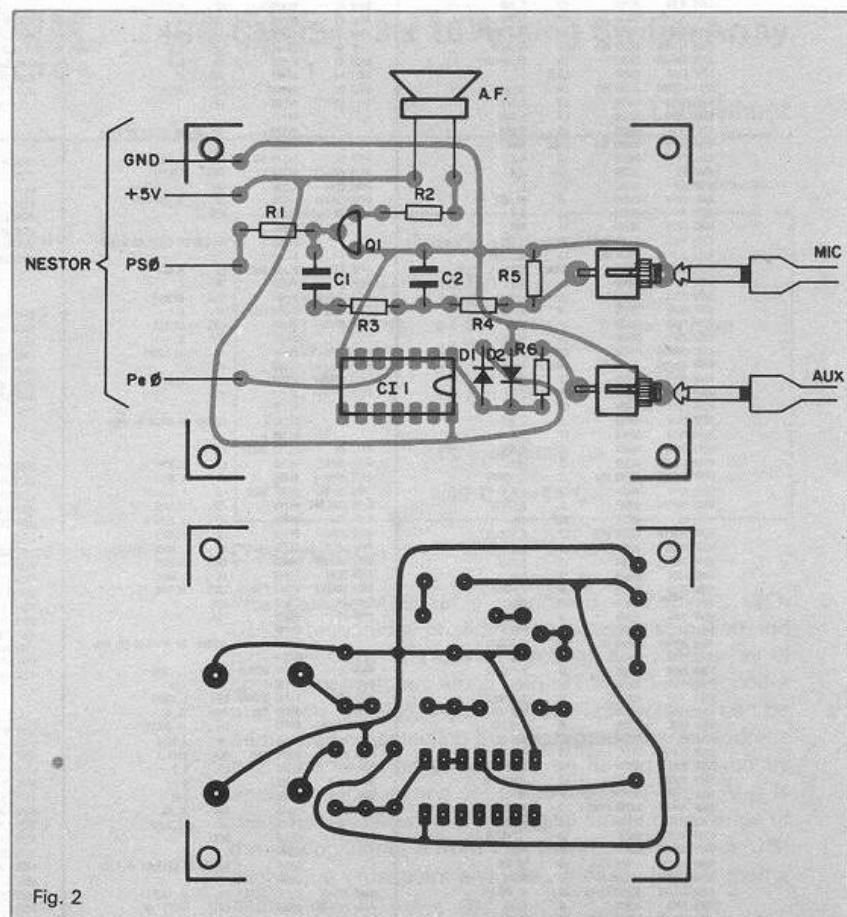


Fig. 2

calcula e detecta erros de comprimento — ou seja, valores colocados não válidos.

— 0454 a 045B(*SAFITA*) — sub-rotina de saída de fita. Tem a função de colocar o bloco de memória para o cassete.

— 045E a 0461(*SABYTE*) — sub-rotina de saída de um byte do bloco. Tem a função de gerar o sinal no formato da figura 3b.

— 0471 a 048A(*SAIBIT*) — sub-rotina de saída de um bit do byte. Tem a função de gerar, de acordo com o bit a ser en-

viado, os formatos apresentados na figura 3a.

— 048B a 04A2(*GE 1 kHz, GE 2 kHz e TOM*) — esta sub-rotina tem basicamente 3 pontos de entrada: em 048B, para geração do tom fixo de 1 kHz; em 048F, para geração do de 2 kHz; e em 0491, que é a entrada para gerador de tons programável, onde o registrador C terá o valor de frequência (veja o cálculo no fim do artigo) e o registrador HL, o número de pulsos desejados. A sub-rotina coloca a informação serial

#### Relação de componentes

R1 - 1 kΩ - 1/8 W  
 R2 - 47Ω - 1/4 W  
 R3, R4 - 10 kΩ - 1/8 W  
 R5 - 330Ω - 1/8 W  
 R6 - 180Ω - 1/8 W  
 C1, C2 - 0,01 μF (poliéster)

D1, D2- 1N914 ou equivalente  
 Q1- BC 237  
 CI1- 74LS14  
 Dois jacks fêmea para circuito impresso  
 Alto-falante de 2"

00100 : ****		01330 : ROTINA CRLC. PARAMETROS		02540 : ROTINA DE GERACAO DE PERIODOS	
00110 ; NOVA ELETRONICA		01340 ;		02570 ;	ROTINA DE GERACAO DE PERIODOS
00120 ; INTERFACE DISKETTE		0441 5157208 01350 PRMM	LD H, BUF+2	0517 1100004 02190 PER100	LD DE, 0000H
00130 ; MICRO MESTOR		0114 3E 0136d	LD E, (HL)	0518 0802 02590 LOOP3	IN R, 1023
00140 ; ****		0442 23 01370	INC H	051C 13 02600	INC DE
00150 ;		0115 56 01380	LD 0, (HL)	0510 CB17 02610	BL R
00160 ;		0447 23 01390	INC IL	051F 38F9 02620	JR C,LOOP3
00170 ;		0116 4E 01400	LD C, (HL)	0521 37F 02630	LD H,7FH
00180 ;		0449 23 01410	INC IL	0523 0302 02640	DUT (02),R
00190 ; 00190 ORS 0359H		0117 46 01420	LD H, (HL)	0525 0802 02650 LOOP4	IN R,1023
00200 ; LD R,00H		0448 49 01430	LD L,C	0527 13 02660	INC DE
00210 ; LD (CONT),R		0118 87 01440	OR R	0528 C817 02670	BL R
00220 ; LD H,L,TITULO		0449 ED52 01450	SBC HL,DE	0529 36F9 02680	JR NC,LOOP4
00230 ; ME1 LD DC,DISP		0119 40 01460	LD C,L	052C 34F 02690	LD A,OFFN
00240 ; LD 0000H		0450 44 01470	LD B,H	052E 0304 02700	DUT (04),R
00250 ; LD18		0151 03 01480	INC BC	0530 78 02710	LD A,E
00260 ; ME2 LD H,L,DISP		0452 EB 01490	EX DE,N	0531 FE66 02720	CP REROP
00270 ; LD 8,00H		0152 09 01500	RET	0531 CY 02730	RET
00280 ; E,04H		0453 C9		02740 ;	
00290 ; LD R,01H		01530 ;		02750 ;	ROTINA DE ENTRADA DE FILA
00299 ; CALL VRRA		0454 5E 01540 SFRTR	LD E, (HL)	0534 RF 02770 ENTRIR	XIR R
00300 ; LD R,C		01550 CALL SFRTR		0535 08 02780	EX RF,RF
00301 ; LD R,R		0455 CD5004 01560 CP1		0536 CD4105 02790 LOOPS	CALL LBITE
00302 ; DEC E		0156 ED81 01570 JR PE,SFRTR		0539 73 02800	LD (HL),E
00303 ; JR HZ,NE3		0458 ED5403 01580 RET		053B EDH1 02810	CP1
00304 ; JR HZ,NE4		01590 ;		053D CR8405 02820	JR PE,LOOP5
00305 ; JR Z,ME2		0460 09 01600		053F 08 02830	EX RF,RF
00306 ; LD H,L,DISP		01610 ;		0540 C9 02840	RET
00307 ; LD R,01H		0461 1608 01620 SBYTE	LD 0,00H	02850 ;	
00308 ; LD R,0		01630 OR R		02860 ;	ROTINA DE LEITORA DE BYTES
00309 ; OUT (03),R		0461 CD7104 01640 CALL SRIBIT		02870 ;	
00310 ; IN R,(01)		01650 CB18 01650 LOOP1	RR E	0541 005205 02880 LEYTE	CALL LBITE
00311 ; FE00		0466 CD7104 01660 CALL SRIBIT		0544 1A08 02890	LD 0,00H
00312 ; FE01		0167 15 01670 RET	O	0544 005205 02900 LOOP4	CALL LBITE
00313 ; FE02		0468 ED4400 01680 JR NZ,LOOP1		0549 C818 02910	RR E
00314 ; FE14		01690 SCF		0549 1B27 02920	DEC D
00315 ; FE403		0469 CD7104 01700 CALL SRIBIT		054C 26F8 02930	JR NZ,LOOP6
00316 ; FE10		01710 RET		054E CD5205 02940	CALL LBITE
00317 ; FE13		01720 ;		0551 C9 02950	RET
00318 ; FE13		04730 ;		02880 ;	
00319 ; FE03		01730 ;		02870 ;	ROTINA DE LEITORA DE BITS
00320 ; JR Z,ME3		04740 09 01740 SRIBIT	EXI	02889 ;	
00321 ; LD R,01H		01750 7600 01760 LD H,00H		0552 99 02990 LBITE	EXX
00322 ; LD C,SRH		0474 3B09 01770 JR C,SRH		0553 210000 03000	LD H,000H
00323 ; CF,00H		0178 2E10 01780 SRIO 1,0	L,ZEROZ	0554 CD7105 03010 LOOP1	CALL PER100
00324 ; CR0304		0478 CD8F04 01790 CRBL GE2XHZ		0559 14 03020	INC O
00325 ; C7		0179 2E04 01800 LD L,ZEROZ		0559 2011 03030	JR HZ,ERR0T
00326 ; RST 00H		0479 1807 01810 FINH2		055C 2804 03040	JL C,PERP
00327 ; RST 00H		01810 2E08 01820 SRIT LD L,UMHZ		0555 29 03050	DEC L
00328 ; Z,FILE		0481 CD8F04 01830 CRBL GE2XHZ		0557 20 03060	DEC L
00329 ; FE01		01840 2E08 01840 LD L,UMHZ		0560 CDCA 03070	SET 0,H
00330 ; FE02		0482 CD8004 01850 FINH2	CALL GE1XHZ	0561 1B27 03080	JR LOOP7
00331 ; FE03		01850 09 01860 EXI		0564 26 03090	SCF L
00332 ; FE04		0483 C970 01870 RET		0565 26 03100	INC L
00333 ; FE05		01880 ;		0567 28D3 03110	JR Z,LOOP7
00334 ; FE06		0488 00520		0568 99 03130	EXX L
00335 ; FE07		01890 ;		0569 C915 03120	BL L
00336 ; FE08		0489 00520		02890 ;	
00337 ; FE09		01900 ;		02900 ;	ROTINA DE ERRO DE TRANSMISSAO
00338 ; FE09		0490 00520		02910 ;	
00339 ; Z,FONTE		01910 BE1HZ	LD C,V1HZ	0545 C9 02910	RET
00340 ; LD R,(01)		0492 16100 01920 JR TON		02910 ;	
00341 ; LD C,SRH		01930 BE2HZ	LD C,V2HZ	0546 08 02918	EXI AF,AF
00342 ; LD 02H		0494 4E40 01940 TON	RD	0547 32 03110	SCF O,H
00343 ; Z,FILE		01950 29 01960 LD R,00H		0548 08 03200	EX AF,AF
00344 ; C7		0496 00510 01970 QUD	OUT (04),R	0549 37 03210	EXX
00345 ; RST 00H		0197 41 01980 LD B,C		0550 29 03210	EXX
00346 ; C21905		0498 10F6 01990 DMJZ \$		0551 C944 03220	RET
00347 ; C31903		01990 BE01 02010 XOR 01H		0552 218205 03230 HERR0	LD H,ERR0
00348 ; C21905		0499 00520 02010 SBC HL,DE		0575 CD9004 03240	REST
00349 ; C21905		01990 2D 02020 JR NZ,QUD		0578 1B27 03250	JR HERR0
00350 ; C21903		0499 00520 02030 RET		0578 218805 03260 MFH LD H,LFIN	REST
00351 ; C21908		01990 2D 02040 JR MC,COND3		0570 CD9004 03270	CALL REST
00352 ; C21908		0499 00520 02050 LD H,BUF		0580 1B27 03280	JR HFIN
00353 ; C21908		01990 2D 02060 LD H,BUF		0582 04 03290 EXR DEFB 0AH	
00354 ; C21908		0499 00520 02070 LD H,BUF		0583 CE 03300 DEFB 0CEH	
00355 ; C21908		01990 2D 02080 LD (BUF),H		0584 1E 03310 DEFB 0CEH	
00356 ; C21908		0499 00520 02090 LD R,0FFH		0585 C9 03320 DEFB 0CEH	
00357 ; C21908		01990 2D 02100 OUT (03),R		0586 F1 03330 DEFB OFFH	
00358 ; C21908		0499 00520 02110 LD R,0FFH		0587 FF 03340 DEFB OFFH	
00359 ; C21908		01990 2D 02120 OUT (01),R		0588 8E 03350 FIN DEFB	
00360 ; C21908		0499 00520 02130 LD H,1000		0589 F9 03360 DEFB 0FH	
00361 ; C21908		01990 2D 02140 CWD2	CALL PER100	0588 C9 03370 DEFB 0CBH	
00362 ; C21908		0499 00520 02150 C,COND1		0589 FF 03380 DEFB OFFH	
00363 ; C21908		01990 2D 02160 DEC H		0590 FF 03390 DEFB OFFH	
00364 ; C21908		0499 00520 02170 LD R,H		0590 FF 03400 DEFB OFFH	
00365 ; C21908		01990 2D 02180 OR L		0591 37 03410 TITULO	DEFB 0FH
00366 ; C21908		0499 00520 02190 LD H,COND2		0592 F9 03420 DEFB 0FH	
00367 ; C21908		01990 2D 02200 LD H,COND3		0593 87 03430 DEFB 0FH	
00368 ; C21908		0499 00520 02210 LD H,BUF		0594 27 03450 DEFB 0FH	
00369 ; C21908		01990 2D 02220 LD H,BUF		0595 C9 03460 DEFB 0FH	
00370 ; C21908		0499 00520 02230 LD H,BUF		0596 27 03470 DEFB 0FH	
00371 ; C21908		01990 2D 02240 LD H,BUF		0597 27 03480 DEFB 0FH	
00372 ; C21908		0499 00520 02250 LD H,BUF		0598 27 03490 DEFB 0FH	
00373 ; C21908		01990 2D 02260 LD H,BUF		0599 27 03500 DEFB 0FH	
00374 ; C21908		0499 00520 02270 LD H,BUF		0600 27 03510 DEFB 0FH	
00375 ; C21908		01990 2D 02280 LD H,BUF		0601 27 03520 DEFB 0FH	
00376 ; C21908		0499 00520 02290 LD H,BUF		0602 27 03530 DEFB 0FH	
00377 ; C21908		01990 2D 02300 LD H,BUF		0603 27 03540 DEFB 0FH	
00378 ; C21908		0499 00520 02310 LD H,BUF		0604 27 03550 DEFB 0FH	
00379 ; C21908		01990 2D 02320 LD H,BUF		0605 27 03560 DEFB 0FH	
00380 ; C21908		0499 00520 02330 LD H,BUF		0606 27 03570 DEFB 0FH	
00381 ; C21908		01990 2D 02340 LD H,BUF		0607 27 03580 DEFB 0FH	
00382 ; C21908		0499 00520 02350 LD H,BUF		0608 27 03590 DEFB 0FH	
00383 ; C21908		01990 2D 02360 LD H,BUF		0609 27 03600 DEFB 0FH	
00384 ; C21908		0499 00520 02370 LD H,BUF		0610 27 03610 DEFB 0FH	
00385 ; C21908		01990 2D 02380 LD H,BUF		0611 27 03620 DEFB 0FH	
00386 ; C21908		0499 00520 02390 LD H,BUF		0612 27 03630 DEFB 0FH	
00387 ; C21908		01990 2D 02400 LD H,BUF		0613 27 03640 DEFB 0FH	
00388 ; C21908		0499 00520 02410 LD H,BUF		0614 27 03650 DEFB 0FH	
00389 ; C21908		01990 2D 02420 LD H,BUF		0615 27 03660 DEFB 0FH	
00390 ; C21908		0499 00520 02430 LD H,BUF		0616 27 03670 DEFB 0FH	
00391 ; C21908		01990 2D 02440 LD H,BUF		0617 27 03680 DEFB 0FH	
00392 ; C21908		0499 00520 02450 LD H,BUF		0618 27 03690 DEFB 0FH	
00393 ; C21908		01990 2D 02460 LD H,BUF		0619 27 03700 DEFB 0FH	
00394 ; C21908		0499 00520 02470 LD H,BUF		0620 27 03710 DEFB 0FH	
00395 ; C21908		01990 2D 02480 LD H,BUF		0621 27 03720 DEFB 0FH	
00396 ; C21908		0499 00520 02490 LD H,BUF		0622 27 03730 DEFB 0FH	
00397 ; C21908		01990 2D 02500 LD H,BUF		0623 27 03740 DEFB 0FH	
00398 ; C21908		0499 00520 02510 LD H,BUF		0624 27 03750 DEFB 0FH	
00399 ; C21908		01990 2D 02520 LD H,BUF		0625 27 03760 DEFB 0FH	
00400 ; C21908		0499 00520 02530 LD H,BUF		0626 27 03770 DEFB 0FH	
00401 ; C21908		01990 2D 02540 LD H,BUF		0627 27 03780 DEFB 0FH	
00402 ; C21908		0499 00520 02550 LD H,BUF		0628 27 03790 DEFB 0FH	
00403 ; C21908		01990 2D 02560 LD H,BUF		0629 27 03800 DEFB 0FH	
00404 ; C21908		0499 00520 02570 LD H,BUF		0630 27 03810 DEFB 0FH	
00405 ; C21908		01990 2D 02580 LD H,BUF		0631 27 03820 DEFB 0FH	
00406 ; C21908		0499 00520 02590 LD H,BUF		0632 27 03830 DEFB 0FH	
00407 ; C21908		01990 2D 02600 LD H,BUF		0633 27 03840 DEFB 0FH	
00408 ; C21908		0499 00520 02610 LD H,BUF		0634 27 03850 DEFB 0FH	
00409 ; C21908		01990 2D 02620 LD H,BUF		0635 27 03860 DEFB 0FH	
00410 ; C21908		0499 00520 02630 LD H,BUF		0636 27 03870 DEFB 0FH	
00411 ; C21908		01990 2D 02640 LD H,BUF		0637 27 03880 DEFB 0FH	
00412 ; C21908		0499 00520 02650 LD H,BUF		0638 27 03890 DEFB 0FH	
00413 ; C21908		01990 2D 02660 LD H,BUF		0639 27 03900 DEFB 0FH	
00414 ; C21908		0499 00520 02670 LD H,BUF		0640 27 03910 DEFB 0FH	
00415 ; C21908		01990 2D 02680 LD H,BUF		0641 27 03920 DEFB 0FH	
00416 ; C21908		0499 00520 02690 LD H,BUF		0642 27 03930 DEFB 0FH	
00417 ; C21908		01990 2D 02700 LD H,BUF		0643 27 03940 DEFB 0FH	
00418 ; C21908		0499 00520 02710 LD H,BUF		0644 27 03950 DEFB 0FH	
00419 ; C21908		01990 2D 02720 LD H,BUF			

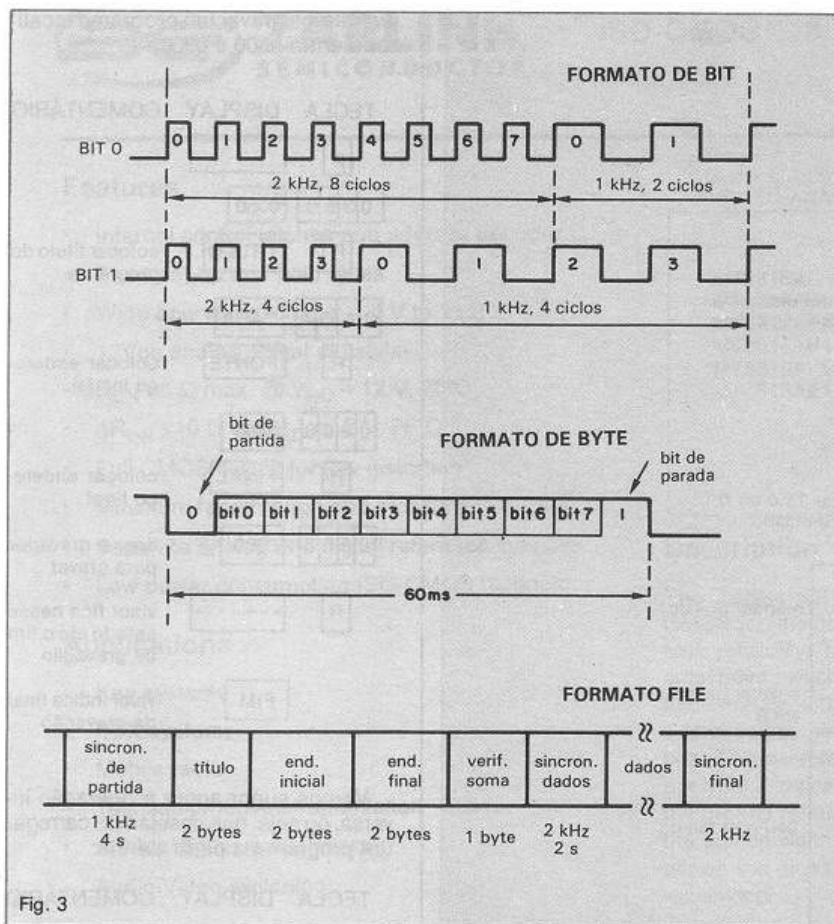


Fig. 3

no portal de saída (bit 0).

— 04A3 a 0514(LFITA) — programa principal de leitura de fita, que utiliza as várias sub-rotinas seguintes, de modo a detectar as informações contidas no file (sincronismo, endereço inicial e final e dados) e organizá-las convenientemente na memória.

— 0517 a 0533(PERIOD) — sub-rotina com a função de ler a informação no portal de entrada (bit 0) e detectar seu período (ou seja, se é relativo a 1 ou 2 kHz); o resultado sai no bit de transporte (C-2 kHz, NC-1 kHz).

— 0534 a 0540(ENFITA) — sub-rotina de entrada de fita; carrega um bloco de memória da fita, onde HL deve ter o endereço inicial do bloco e BC, o comprimento do mesmo.

— 0541 a 0551(LEBYTE) — sub-rotina de leitura de byte (ou seja, lê um byte da fita). O valor do byte é lido no registrador E.

— 0552 a 0571(LEBIT) — sub-rotina de leitura de um bit. Lê um bit da fita, ou

seja, detecta o formato da figura 3a; se é detectado um formato errado, também indica erro de transmissão.

— 056D a 056F(ERRO T) — sub-rotina de detecção de erro; tem a função de provocar um set no bit de transporte do registrador F', para indicar erro.

— 0572 a 0578(MERRO) — programa para escrever no display a mensagem de erro.

— 057A a 0580(MFIM) — programa para escrever no display a mensagem de fim de transmissão.

— 0582 a 059F — tabelas de mensagens de erro, fim de mensagem, título, endereço fonte e endereço final.

*Obs.: Todo o programa foi feito para rodar com clock de 4 MHz ou no projeto original, 3,58 MHz.*

**Gravação e leitura** — Para demonstrar melhor a operação conjunta do Nestor com o novo programa e da interface, vamos dar dois exemplos práticos, um de cada tipo. Suponha que

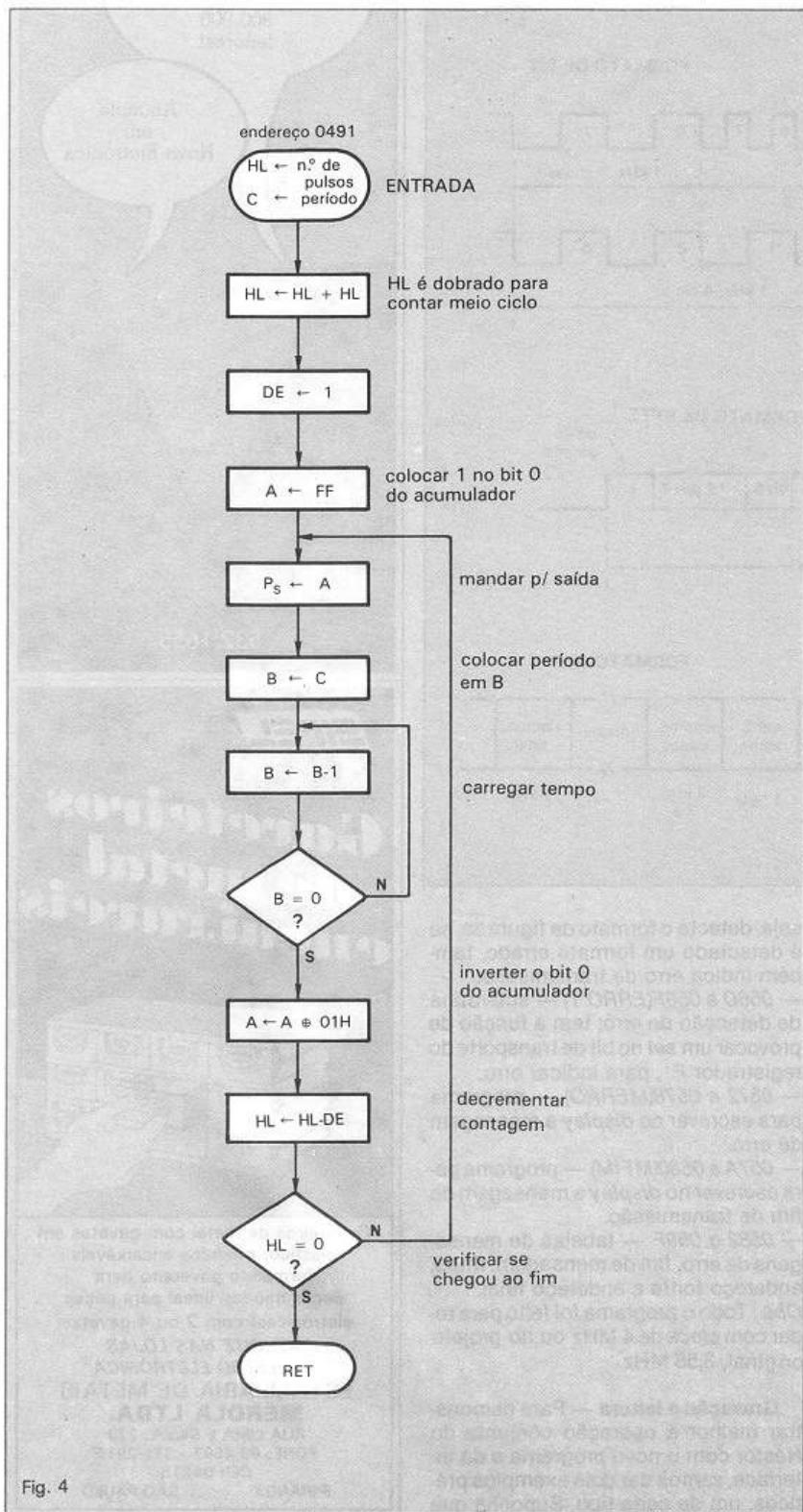


Fig. 4

queiramos gravar um programa localizado entre 0800 e 0900:

TECLA	DISPLAY	COMENTÁRIO
P	[ ]	
0 3 5 0	0350	
R	TÍTULO	colocar título do programa
C A C A	CACA	
R	FONTE	colocar endereço inicial
0 8 0 0	0800	
R	FINAL	colocar endereço final
0 9 0 0	0900	ligar o gravador para gravar
R	[ ----- ]	visor fica nesse estado até o fim da gravação
	FIM	visor indica final da gravação

Vamos supor agora a operação inversa, ou seja, que desejamos carregar um programa a partir da fita:

TECLA	DISPLAY	COMENTÁRIO
P	[ ]	
0 3 5 0	0350	
R	TÍTULO	colocar o nome do programa desejado
C A C A	CACA	
+ 1	[ ----- ]	depois de acionar essa tecla, deve-se ligar o gravador em reprodução, com a fita posicionada na área do programa

*Obs.:* Quando o micro receber o sincronismo, aparecerá no display o símbolo [ ----- ] e, em seguida, o título do programa por alguns segundos. Se o nome for igual ao pedido, o display mostrará o símbolo [ ----- ], indicando que o Nestor passou a receber os dados. Caso

endereço	código operacional	label	mnemônico
0800	0E00	loop	LD C,00
0802	21C000		LD HL,00C0
0805	CD9104		CALL TOM
0808	0EC0		LD C,C0
080A	210001		LD HL,0100
080D	CD9104		CALL TOM
0810	C30008		JP LOOP

encontre um título diferente, o micro fica esperando indefinidamente, até que surja na fita um programa com o título requisitado. Ao fim da gravação, aparecerá no visor a mensagem de final de transmissão.

**A sub-rotina TOM** — Como dissemos anteriormente, essa sub-rotina permite usar o Nestor como um gerador de tons programável. Ela pode ser

facilmente visualizada no fluxograma simplificado da figura 4.

A freqüência do sinal é calculada usando-se os chamados *T states*, que são o número de pulsos de *clock* necessários para perfazer uma instrução. Ficamos, então (com dados fornecidos pelos manuais):

- tempo de DJNZ: 13 pulsos
- tempo do *loop* de verificação de contagem (sem DJNZ): 44 pulsos

$$(n.º \text{ de } T \text{ states}) \\ período \text{ total} = \overbrace{2 \cdot (44 + 13 \cdot C)}^{\text{(n.º de } T \text{ states)}}.$$

período de 1 ciclo de *clock*( $t_{clock}$ )

Portanto, temos:

$$C = \left( \frac{\text{período desejado}}{t_{clock} \cdot 2} - 44 \right) / 13$$

onde C é dado em decimal e  $t_{clock}$ , para o Nestor, é 1/3,58 MHz.

Podemos, então, gerar tons de freqüência relativamente precisa, sem muitos problemas. Veja na figura 5 um exemplo de programa usando essa sub-rotina, simulando o som de uma sirenóis inglesa; ela utiliza duas freqüências diferentes e também dois tempos diferentes de duração, alternadamente. Pode-se gerar programas para as mais variadas aplicações em áudio; mas eles ficam por conta da imaginação do montador. ■