

# Mapa da ROM do TK90X

*Álvaro Ferreira de Freitas Borja*

Marcelo Martins do Carmo – Maio 2005

[www.TK90X.com.br](http://www.TK90X.com.br)

---

## ROM DO ZX SPECTRUM

O ZX Spectrum possui um monitor ocupando 16 Kb de programas escritos em linguagem de máquina do microprocessador Z80. Este monitor está dividido em três partes principais, que são: rotinas de entrada e saída, interpretador BASIC e operação de expressões.

**Rotinas de restart e tabelas** – no início do programa monitor estão localizadas as várias rotinas de *restart* que são chamadas por meio da instrução RST. Todos os *restarts* do Z80 são usados; por exemplo, RST 08 é utilizado para mensagens de erro do sistema. Nesta parte do programa monitor estão as tabelas que contêm a forma expandida dos tokens e códigos-chave.

**Rotina do teclado** - é feita uma leitura do teclado a cada 1/60 de segundo (ou 1/50, dependendo da localidade); a rotina de teclado retorna com o código do caráter requerido.

A todo instante o registrador E retorna com um valor na faixa de 03-39, que representa cada uma das 40 teclas do teclado ou 255, caso não haja nenhuma tecla pressionada.

**Rotina de Som** - o Spectrum possui um alto-falante simples e uma nota é produzida, sendo usada repetidamente a instrução OUT, de modo apropriado, na porta 254 (FEh).

*Nota: o TK90X utiliza o alto-falante da própria televisão.*

**Rotinas de operação com o cassete** - ao contrário do ZX81, no Spectrum este é um bloco bastante extenso (1.329 Kbytes).

Programas escritos em BASIC ou blocos de dados são precedidos por um cabeçalho de 17 bytes - header que é salvo primeiramente. Este cabeçalho descreve as características do bloco de dados que é gravado após ele.

Em todos os casos, (SAVE, LOAD, MERGE e VERIFY) os pares de registros DE contêm o comprimento do bloco; o par XI contêm o endereço básico, e o registro A é carregado com 00, para o caso de um header, ou FFh (255), para um bloco de dados / programa.

**Rotinas de manipulação do vídeo e impressão** - todas as rotinas de entrada / saída do Spectrum são vetorizadas através da área de informações dos canais e fluxos.

No Spectrum, normalmente as entradas são possíveis somente pelo teclado, porém saídas podem ser direcionadas para a impressora, parte superior ou inferior do vídeo.

**Rotinas de execução** - nesta parte do programa monitor está localizado o procedimento de inicialização e o loop principal de execução do interpretador BASIC.

Toda linha BASIC quando retorna do editor é checado quanto à correta sintaxe e então é salva na área de programa, isto para o caso de iniciar com um número de linha; caso contrário, é executada imediatamente.

**Interpretação do comando e a linha BASIC** - esta parte do monitor considera uma linha BASIC como sendo um conjunto de declarações e, no seu procedimento, cada declaração é iniciada por um comando particular.

**Rotinas aritméticas** - o Spectrum opera com dois tipos de números: valores inteiros que variam de -65535 a +65535 ou números de ponto flutuante na forma de cinco bytes (foi mantido o padrão do ZX81).

**O calculador de ponto flutuante** - o calculador do Spectrum opera números e strings, e estas operações são especificadas por literais. Essa pode, portanto, ser considerada como sendo uma linguagem interna de operação do *stack* no calculador.

Esta parte do programa monitor contém rotinas para todas as funções aritméticas. As aproximações para SIN x, EXP x, LN x e ATN x são obtidas pelo desenvolvimento dos polinômios do *Chebyshev*.

---

## As rotinas de restart e tabelas

- 0000 Rotina de inicialização (START). As interrupções mascaráveis são desabilitadas, e o par de registradores DE é carregado com o valor máximo para o topo da RAM .
- 0008 Rotina das mensagens de erro do sistema. O ponteiro de erro é ajustado para a posição de erro.
- 0010 Imprime a caráter cujo código está no registrador A.
- 0018 O conteúdo da posição corrente endereçada por CH-ADD é buscada: um retorno é feito se o valor presente for de um caráter imprimível, caso contrário, CH-ADD é incrementado e o teste repetido.
- 0020 Quando uma linha BASIC é interpretada, esta rotina é chamada repetidamente ao longo da linha.
- 0028 Desvia para o calculador de ponto flutuante.
- 0030 Esta rotina cria espaço na área de trabalho. O número de espaços é determinado pelo valor corrente no par BC.
- 0038 Interrupção mascarável, sempre que ocorre o clock é incrementado e é feita uma varredura no teclado.
- 0066 Interrupção não mascarável (NMI). Se a variável do sistema NMIVCT (23728) for 0, ocorrerá um reset (no TK90X, executará um salto para o endereço que estiver armazenado nesta variável, porém, 0 não reseta).
- 007D O valor presente no registro A é testado para ver se este é imprimível.
- 0095-0204 Tabela dos *tokens* o último byte de cada *token* está com o bit 7 setado.
- 0205-022B Tabela das chaves principais (modo L e CAPS SHIFT).
- 022C-025F Tabela do modo estendido.
- 0260-0269 Tabela dos códigos de controle (tecla digitada + CAPS SHIFT).
- 026A-0283 Tabela dos símbolos (tecla + SYMBOL SHIFT).
- 0284-028D Tabela do modo estendido (CAPS SHIFT + SYMBOL SHIFT + tecla).

---

## Rotinas do teclado

- 028E Rotina de leitura do teclado; se nenhuma tecla estiver pressionada, o par DE retorna com o valor 65535.
- 02BF Esta rotina é chamada sempre que ocorre uma interrupção mascarável. Em uso normal, ocorre uma vez a cada 20 ms. A finalidade desta rotina é ler o teclado e determinar o valor da tecla pressionada.
- 0310 Sub-rotina de repetição de tecla. Haverá uma repetição de tecla, se na primeira ocasião ela ficar pressionada por um período de 0,7 segundos, e, posteriormente, se ficar pressionada 0,1 segundo.
- 031E O valor da tecla é testado e um retorno é feito, se não houver tecla pressionada ou somente SHIFT; caso contrário, o código desta tecla é achado.
- 0333 Rotina de decodificação do teclado. Esta rotina é entrada com o código principal do registro E, o valor das FLAGS em D, o valor no modo do registro C e o byte invertido no registro B.

---

## Rotinas de som

- 03B5 Sub-rotina que gera som no alto-falante. A entrada é feita com o par de registradores DE contendo o valor "F\*T", onde uma nota de frequência F terá uma duração de T segundos, e o par HL conterá um valor igual ao número de T *states do loop* de temporização dividido por 4.
- 03F8 Comando SOUND; esta sub-rotina é entrada com dois números na pilha do calculador. O número no topo representa a tonalidade da nota e o número abaixo, a duração.
- 046E-04A9 Tabela dos SEMI-TONS.

---

## Rotinas de operação com cassete

- 04C2 Salva o cabeçalho (header) e posteriormente o programa corrente na memória ou bloco de dados.
- 053F Esta sub-rotina é comum tanto para o SAVE como para o LOAD. A borda é setada com a cor original, e a tecla BREAK é testada.
- 0556 Carrega o header e depois carrega (ou verifica) um bloco de dados.
- 0605 Este ponto de entrada é usado por todas as quatro instruções (SAVE, LOAD, VERIFY e MERGE). O valor contido na variável T-ADDR (23668) é que permitirá fazer a distinção entre as quatro instruções.
- 0642 Indica erro do tipo F – – – nome inválido.

- 0670 Idem erro tipo 2 – – – variável inexistente.
- 07CB Rotina de controle do VERIFY .
- 0802 Sub-rotina para carregar um bloco de dados. É utilizada por todas as rotinas de carregamento (LOAD, VERIFY e MERCE).
- 0808 Rotina de controle do carregamento de um programa BASIC, variáveis ou um array.
- 08B6 Rotina de controle do MERGE.
- 092C Sub-rotina para executar MERGE numa linha ou variável.
- 09A1 - 09F3 Tabela de mensagens do cassete. Cada mensagem é mostrada com o último caráter invertido.

---

## Rotinas de manipulação do vídeo e impressora

- 09F4 Esta rotina é entrada com o registrador A contendo o código de um caráter de controle, um caráter printável ou um *token*.
- 0A11-0A22 Tabela dos caracteres de controle.
- 0A23 Sub-rotina para mover o cursor para a esquerda.
- 0A3D Sub-rotina para mover o cursor para a direita.
- 0A4F Sub-rotina de *carriage return*.
- 0A5F Sub-rotina para imprimir usando a tabulação com vírgula (coluna 0 ou 16). Ex.: PRINT,,"ok!".
- 0A69 Imprimir uma interrogação sempre que for feita uma tentativa de imprimir um código não printável.
- 0A6D Rotina de caracteres de controle com operandos.
- 0ADC Esta sub-rotina guarda a posição da linha e coluna, bem como o endereço do pixel na variável do sistema apropriada.
- 0B03 Sub-rotina para buscar os parâmetros da posição corrente na variável do sistema apropriada.
- 0B24 Sub-rotina para imprimir um caráter qualquer.
- 0B7F Sub-rotina usada para imprimir todos os bits dos caracteres (matriz 8x8). Na entrada, o par de registradores DE contém o endereço base da tabela de caracteres; o par HL, o endereço de destino; e o par BC, os valores da linha e coluna correntes.
- 0BDB Seta byte na área de atributos.
- 0C0A Sub-rotina usada para imprimir mensagens e *tokens*. O registro A é carregado com o número da mensagem ou do *token* da tabela. O par DE é carregado com o endereço base da tabela.
- 0C41 Retorna com o par DE apontando para o caráter inicial da entrada requerida.
- 0C55 Esta rotina é chamada a qualquer momento para forçar a necessidade de um scroll na tela.
- 0D4D Faz uma cópia temporário dos itens de cor.
- 0D6B Comando CLS.
- 0DAF Limpa toda a área de vídeo, B usada pelo CLS, listagem automática e rotina principal de execução.
- 0DD9 Seta parâmetros para CLS.
- 0DFE Sub-rotina de scroll. O registro B é carregado com o número de linhas a serem roladas (B = 23).
- 0E44 Esta sub-rotina limpa as "n" linhas da parte inferior do vídeo, sendo que o número de linhas deverá ser carregado no registro B.

- 0E88 Limpa a área de atributos.
- 0EAC Comando COPY.
- 0ECD Envia para a impressora o conteúdo do buffer de impressão.
- 0EDF Limpa o buffer da impressora.
- 0EF4 Copia uma linha na impressora.
- 0F2C Rotina do editor, O editor é chamado em duas ocasiões: quando da entrada de uma linha BASIC e pela rotina do comando INPUT.
- 0F81 Esta sub-rotina adiciona um código para o corrente EDIT ou linha INPUT.
- 0FA0-0FA8 Tabela das teclas de edição.
- 0FA9 Sub-rotina de edição.
- 0FF3 Cursor para baixo, no modo de edição.
- 1007 Cursor para esquerda, no modo de edição.
- 100C Cursor para direita, no modo de edição.
- 1015 Sub-rotina para apagar um caráter, no modo de edição.
- 101E Os dois próximos códigos da rotina de entrada são ignorados.
- 1024 Sub-rotina de entrada para a edição.
- 1031 Sub-rotina de corte. O endereço do cursor é carregado no par HL e este é decrementado, a não ser que o cursor já esteja no início da linha. Cuidado é tomado para não colocar o cursor entre os caracteres de controle e seus parâmetros.
- 103E Loop para garantir que os caracteres de controle não sejam separados de seus parâmetros.
- 1059 Cursor para cima, no modo de edição.
- 1097 Sub-rotina para limpar a área de edição e espaço de trabalho.
- 10A8 Esta é uma das importantes sub-rotinas; ela retorna o código da ultima tecla pressionada no modo CAPS LOCK. A troca no modo e parâmetros de controle da cor são manejados dentro da sub-rotina.
- 111D Esta sub-rotina é chamada sempre que a linha na área de edição ou INPUT precisem ser printados na parte inferior do vídeo.
- 1190 Esta sub-rotina retorna com o par HL apontando para a primeira posição e o par DE para a última posição da área de edição ou espaço de trabalho.
- 11A7 Sub-rotina para remover a *máscara* da marca de início da representação em ponto flutuante em uma linha BASIC (código 14).

---

## ROTINAS DE EXECUÇÃO

- 11B7 Comando NEW.
- 12A2 Loop de execução principal. O loop principal estende-se do endereço 12A2h para o endereço 15AEh e ele controle o modo de edição, a execução de comandos diretos e a produção de mensagens.
- 1391-1554 Tabela das mensagens de erro.
- 155D Sub-rotina para adicionar / mudar uma linha em um programa BASIC que esteja na área de trabalho.
- 15AF Informações iniciais dos quatro canais de comunicação (K - teclado, S - vídeo, R - área de trabalho e P - impressora).
- 15C6-15D3 Tabela de dados correntes para o canal.
- 15D4 Sub-rotina para esperar por uma tecla.

- 15F7 Sub-rotina para abrir um canal. O registrador A deverá conter o número do canal (normalmente FDh a 03). Então, dependendo do dado corrente, um canal particular será aberto.
- 1615 Seta as flags apropriadas para os diferentes canais.
- 162D-1633 Tabela de códigos para os canais.
- 1634 Seta as flags para o canal K.
- 1642 Seta as flags para o canal S.
- 164D Seta a flag para o canal P.
- 1652 Sub-rotina para criar um espaço em uma área. O par de registros HL deverá conter o endereço da posição onde se deseja criar o espaço.
- 1655 Esta sub-rotina é usada pela anterior, sendo que permite criar N espaços. O par BC deverá conter o número de espaços que se deseja criar e o par HL o endereço da primeira posição a partir da qual deverá ser criado.
- 169E Esta sub-rotina é chamada quando se usa em RST 30h. Ela cria espaço entre a área de trabalho existente e a pilha do calculador.
- 16B0 Sub-rotina para limpar a área de edição e todas as posteriores.
- 16E5 Comando CLOSE #
- 1736-161B Tabela de dados para fechar o canal corrente.
- 1736 Comando OPEN #
- 1795 Comandos LIST e LLIST.
- 17F5 Ponto de entrada do LLIST. Abre o canal para a impressora.
- 17F9 Ponto de entrada do LIST. Abre o canal para o vídeo.
- 1855 Sub-rotina para imprimir uma linha BASIC. O par de registradores HL deverá conter a endereço de início da linha, isto é, o endereço do byte de maior valor do número da linha.
- 18C1 Imprimir um carácter em FLASH. O cursor de erro e os outros cursores são usados por esta sub-rotina.
- 18E1 Imprime o cursor (modo C, E, G, K ou L).
- 1925 Todos os caracteres / *tokens* em uma linha de um programa BASIC são impressos fazendo-se repetidas chamadas a esta sub-rotina.
- 196F Acha o endereço de uma linha. O número da linha deverá ser carregado no par de registradores HL. Esta sub-rotina retoma com o endereço inicial da linha ou da primeira linha seguinte (caso esta não exista) no par HL, e da linha anterior, no par DE.
- 19B8 Usada para achar a "próxima linha" na área de programa ou a próxima variável na área de variáveis.
- 19FB Esta rotina é usada para ler o número, da linha na área de edição. Se esta não tiver número, isto é, uma instrução direta, então o número da linha é considerado 0 (zero).
- 1A1B Imprime o número das linhas de um programa em BASIC. Valores acima de 9.999 serão impressos incorretamente.
- 1A48-1B16 Tabela dos comandos BASIC.
- 1B17 Cheque da sintaxe.
- 1BB2 Comando REM.
- 1C01-1C0C Tabela da classe dos comandos.
- 1C0D Comandos classe 03. São comandos que podem ou não serem seguidos por um número. Exemplo: RUN e RUN 100.
- 1C10 Comandos classe 00. Não precisam de operando. Exemplo: COPY.
- 1C11 Comando classe 05. Exemplo: PRINT.
- 1C1F Comandos classe 01, 02 e 04. São usados por comando contendo variáveis. Exemplo: LET, FOR e NEXT.
- 1C22 Fornece os valores apropriados para as variáveis do sistema DEST (23621) e STRLEN (23666).
- 1C56 É usada pelas instruções LET, READ e INPUT para primeiramente avaliar e, então, atribuir valores para as variáveis previamente designadas.

- 1C96 Seta os parâmetros de cores.
- 1CBE Rotina usada pelas instruções PLOT, DRAW e CIRCLE para especificar a condição *default* (FLASH 8, BRIGHT 8 e PAPER 8).
- 1CDB Rotina usada pelas instruções SAVE, LOAD, VERIFY e MERGE.
- 1CEE Comando STOP.
- 1CF0 Comando IF. Na entrada o valor da expressão entre o IF e o THEN é o último valor na pilha do calculador. Se este é de lógica verdadeira, então, a próxima declaração é executada; caso contrário, a linha é considerada encerrada.
- 1D03 Comando FOR. A entrada é feita com o valor inicial e o limite de FOR no topo da pilha do calculador.
- 1D86 Esta sub-rotina é usada para detectar a ocorrência de outros DATA, DEF FN ou NEXT em um programa BASIC.
- 1DAB Comando NEXT.
- 1DDA Sub-rotina de controle do loop do NEXT.
- 1DEC Instrução READ.
- 1E27 Instrução DATA. Durante a checagem da sintaxe, uma instrução DATA é checada para ver se contém uma série de expressões válidas, separadas por vírgulas. Porém, durante a execução do programa, ela é pulada.
- 1E42 Comando RESTORE.
- 1E4F Comando RANDOMIZE.
- 1E5F Comando CONTINUE.
- 1E67 Comando GOTO.
- 1E7A Função OUT. Os dois parâmetros da declaração OUT são colocados na pilha do calculador e usados diretamente.
- 1E80 Instrução POKE. Funciona de modo similar ao OUT.
- 1E85 Coloca os valores do topo da pilha do calculador em registradores simples (cada um em um registrador).
- 1EA1 Comando RUN.
- 1EAC Comando CLEAR.
- 1EED Comando GOSUB.
- 1F05 Testa a quantidade de memória disponível.
- 1F23 Comando RETURN.
- 1F3A Comando PAUSE. O período da pausa é determinado pela contagem do número de interrupções mascaráveis que ocorrem a cada 1/60 segundos.
- 1F54 Esta rotina é executada constantemente para fazer a leitura da tecla BREAK. Retorna com a flag de Carry setada, caso as teclas SHIFT e BREAK estejam pressionadas ao mesmo tempo.
- 1F60 Função DEF FN.
- 1FC9 Comandos LPRINT e PRTNT. O canal apropriado é necessariamente aberto.
- 1FF5 Sub-rotina usada para pausar a impressão e retornar o carro.
- 1FFC Esta sub-rotina é chamada pelas rotinas dos comandos PRINT, LPRINT e INPUT.
- 2045 Teste do fim de impressão.
- 2089 Comando INPUT.
- 21B9 Sub-rotina chamada repetidamente, para cada valor *inputado*.
- 21D6 Retorna com a flag de zero resetada, somente quando o canal K está em uso.
- 21E1 Rotinas de manipulação dos itens de cores.
- 2211 Rotinas de manipulação das variáveis do sistema que controlam as cores.
- 2294 Comando BORDER.
- 22AA Sub-rotina usada pelo comando PLOT para determinar o endereço de um pixel. A entrada é feita com o par de registradores BC, contendo as coordenadas do pixel, e retorna com o par HL, contendo o endereço do byte da área de vídeo que contém o pixel e o registro A apontando para a posição do pixel dentro do byte.

- 22CB Sub-rotina usada pela varredura da função POINT. Ela é entrada com as coordenadas de um pixel na pilha do calculador e retorna com o valor 1, se o pixel é cor do INK; ou 0, se da cor do PAPER.
- 22DC Instrução PLOT.
- 2307 Esta sub-rotina transfere dois números em ponto flutuante para o par BC.
- 2314 Carrega o registro A com um número, em ponto flutuante que esteja na topo da pilha do calculador. O número deve ser entre 0 e 255.
- 2320 Instrução CIRCLE.
- 2382 Instrução DRAW.
- 247D Sub-rotina usada pelas instruções CIRCLE e DRAW para setar os parâmetros iniciais.
- 24B7 Sub-rotina usada pela instrução DRAW para traçar uma linha reta.

---

## AVALIAÇÃO DE EXPRESSÕES

- 24FB Esta sub-rotina é usada para executar uma avaliação do resultado da próxima expressão. O resultado retornado é o último valor na pilha do calculador. Para um resultado numérico, o último valor será um valor em ponto flutuante. Porém, para um resultado string o último valor consiste de um conjunto de parâmetros. O primeiro valor dos cinco bytes é desprezado, o segundo e o terceiro bytes contêm o endereço de início da string, e o quarto e o quinto bytes contêm o tamanho da string.
- 2596-25AE Tabela das funções de avaliação.
- 25AF Rotinas da avaliação de funções.
- 26C9 Rotina de avaliação de variável.
- 2795-27AF Tabela dos operadores OR, AND, <, >, =, +, etc.
- 27B0-27B6 Tabela das prioridades dos operadores.
- 27BD Esta sub-rotina é chamada para avaliar a função definida por DEF FN.
- 28B2 Esta sub-rotina é chamada a qualquer momento para fazer uma procura na área de variáveis ou de argumentos sempre que uma declaração DEF FN for requerida.
- 2996 Sub-rotina normalmente usada para encontrar os parâmetros que definem uma string existente na área de variáveis ou para retornar com o par HL apontando para o endereço base de um elemento particular ou um número array. Quando chamada pela declaração DIM, esta sub-rotina somente faz um cheque da sintaxe.
- 2AB1 Passa os valores que estão nos registradores A, B, C, D e E para a pilha do calculador.
- 2AFF Comando LET.
- 2BA6 Passa qualquer valor numérico (da pilha do calculador) ou string (da área de trabalho), para a devida posição dentro da área de variáveis.
- 2BAF Continuação do comando LET.
- 2BC6 Pega os parâmetros de uma nova string para testar se há espaço na memória para então a string ser finalmente transferida.
- 2BF1 Transfere o valor no topo da pilha do calculador (ponto flutuante) para os registros B, C, D, E e A, nesta ordem, sendo que B assume o valor do byte mais no topo e assim sucessivamente.
- 2C88 Retorna com a flag de Carry setada, se o valor presente no registro A representa um dígito ou letra válida.
- 2C8D Retorna com a flag de Carry setada, se o valor presente m, registra A representa uma letra válida do alfabeto.
- 2C9B Converte um número decimal em ponto flutuante. Esta sub-rotina lê o número decimal, digito por digito, e coloca o resultado no topo da pilha do calculador.

- 2D1B Retorna com a flag de Carry resetada, se o valor presente no registro A representa um dígito válida.
- 2D28 Converte um valor binário absoluto que esteja no registrador A em um número com ponto flutuante e o coloca no topo da pilha do computador.
- 2D2B Converte um valor binário absoluto que esteja no par de registros BC em um número com ponto flutuante e a coloca no topo da pilha do computador.
- 2D3B Converte um número inteiro para ponto flutuante.

---

## ROTINAS ARITMÉTICAS

- 2D4F Esta sub-rotina mostra o valor que está no topo da pilha do computador, que é o resultado da conversão de um número na forma  $xEn$ , onde  $n$  é um inteiro negativo ou positivo. A entrada na sub-rotina é feita com  $x$  no topo da pilha do computador e  $n$  no registro A.
- 2D7F Esta sub-rotina coloca em DE um valor inteiro  $n$  (onde  $n$  varia de  $-65535$  a  $+65535$ ) da posição endereçada por HL. Sendo  $n$  normalmente o primeiro valor no topo da pilha do computador.
- 2D8C Empilha um valor inteiro  $n$  (onde  $n$  varia de  $-65535$  a  $+65535$ ) da posição endereçada por HL e as quatro posições seguintes, ou seja, recoloca o primeiro valor do topo da pilha do computador.
- 2DA2 Transfere um número em ponto flutuante que esteja no topo da pilha para o par de registradores BC. Se o resultado for maior que  $65535$ , retorna com a flag de Carry setada. Se o resultado for negativo, então, a flag de zero é resetada. O byte de menor valor é copiado também no registro A.
- 2DD5 Transfere um número em ponto flutuante que esteja no topo da pilha para o registrador A.
- 2DE3 Imprime um número em um ponto flutuante.
- 2F8B Multiplica cada byte DE por 10 e retorna com a parte inteira do resultado no registro C.
- 2F9B Repara um número em ponto flutuante para a adição.
- 300F Operação de subtração.
- 3014 Operação de adição.
- 30C0 Prepara um número em ponto flutuante para divisão ou multiplicação, retornando com a flag de Carry setada, se o número for zero, colocando o sinal do resultado no registro A.
- 30CA Operação de multiplicação.
- 31AD Operação de divisão.
- 3293 Recoloca na pilha do computador dois números inteiros transformando para cinco bytes em ponto flutuante, provenientes da operação binária da adição, multiplicação e divisão.
- 3297 Recoloca um número no topo da pilha sob a forma de cinco bytes em ponto flutuante.

## CALCULADOR DE PONTO FLUTUANTE

- 32C5-32D6 Tabela das constantes (0, 1, 1/2,  $\sqrt{2}$ , 10).
- 32D7-335A Tabela dos endereços.
- 335B Sub-rotina do calculador. Usada para executar os cálculos em ponto flutuante. Estes podem ser considerados de três tipos:
- a) Operações binárias. Ex.: divisão, onde dois números em ponto flutuante são subtraídos para gerar um valor no topo da pilha do calculador;
  - b) Operações únicas. Ex.: co-seno, onde o valor no topo da pilha é trocado pelo resultado da função, gerando um novo valor no topo da pilha;
  - c) Operações de manipulação, onde o valor no topo da pilha é copiado para os principais cinco bytes da área de memória do calculador.
- 33A1 Sub-rotinas DELETE.
- 33A2 Executa uma operação aritmética simples.
- 33A9 Teste de memória suficiente para colocar outro número em ponto flutuante na pilha do calculador.
- 33C0 Move um número em ponto flutuante para o topo da pilha do calculador ou do topo da pilha para a área de memória do calculador.
- 3406 Localiza o endereço base para cada porção de cinco bytes da área de memória do calculador para (ou de) onde um número em ponto flutuante será movido da (ou para a) pilha do calculador.
- 341B Usada para empilhar uma constante.
- 342D Armazenar dados na área de memória do calculador.
- 343C Troca o primeiro número do topo da pilha do calculador com o segundo, tipo um SWAP do FORTH.
- 3449 Rotina para gerar as Séries de *Chebyshev*.
- 346A Função ABS.
- 346E NEGATE. Troca o sinal do valor no topo da pilha do calculador.
- 3492 Função SGN.
- 34A5 Função IN.
- 34AC Função PEEK.
- 34B3 Função USR.
- 34BC Função USR string.
- 34E9 Testa se um número em ponto flutuante é zero. Este teste existe que os quatro primeiros bytes do número sejam zero. Retorna com a flag de Carry setada se o número é zero.
- 34F9 Testa se o valor no topo da pilha é maior que zero. Coloca 1, se o valor for igual a zero, e 0, caso contrário.
- 3506 Testa se o valor no topo da pilha é menor que zero. Coloca 1, se o valor for menor que zero, e 0, caso contrário.
- 350B Seta o valor no topo da pilha para 0, se a flag de Carry está resetada; caso contrário, coloca 1.
- 351B Operação lógica OR.
- 3524 Operação lógica AND.
- 352D Operação AND entre um número e uma string.
- 353B Operações de comparação ( $\leq$ ,  $<$ ,  $>$ ,  $\geq$ , etc.).
- 359C Executa concatenação de strings.
- 35C9 Função CHR\$.
- 35DE Função VAL E VAL\$.
- 361F Função STR\$.

- 3669 Função CODE.
- 3674 Função LEN.
- 367A Esta sub-rotina é usada somente pelo gerador de séries, o seu efeito é semelhante a um DJNZ, porém o contador é a variável do sistema BREG (23655).
- 368F Executa um salto condicional, se o valor no topo da pilha do calculador é verdadeiro.
- 369B Finaliza as operações no calculador (byte 38h).
- 36A0 Retorna o valor inteiro do quociente INT (X/Y) no topo da pilha do calculador, e o resto X-INT (X/Y) é colocado abaixo na pilha do calculador. Onde Y é um número inteiro positivo colocado no topo da pilha e X um valor inteiro que esteja na pilha, porém, logo abaixo de Y.
- 36AF Função INT.
- 36C4 Função EXP.
- 3713 Função LN.
- 37AA Função COS.
- 37B5 Função SIN.
- 37DA Função TAN.
- 37E2 Função ATN.
- 3833 Função ASN.
- 3843 Função ACS.
- 384A Função SQR.
- 3851 Executa uma operação binária tipo X elevado a Y ( $X ^ Y$ ).
- 386E-3CFF Área livre.
- 3D00-3FFF Tabela do conjunto de caracteres, 8 bytes para cada caráter.