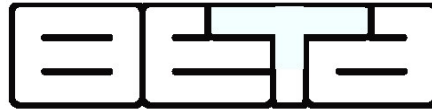


# ***INTERFACE DE DRIVE TK90X/95***



Manual do Usuário

# ÍNDICE

<b>I - APRESENTAÇÃO</b> .....	3
<b>II - INTERLIGANDO O EQUIPAMENTO</b> .....	4
<b>III - CONVENÇÕES</b> .....	5
<b>IV - COMANDOS DO DOS</b> .....	5
1 - RETURN .....	5
2 - RAND USR 15360 .....	5
3 - CAT .....	5
4 - * .....	6
5 - FORMAT .....	6
6 - LOAD .....	7
7 - RUN .....	7
8 - ERASE .....	7
9 - MOVE .....	7
10 - MERGE .....	8
11 - NEW .....	8
12 - COPY .....	8
13 - PEEK .....	8
14 - POKE .....	8
15 - 40 .....	8
16 - 80 .....	9
17 - GOTO .....	9
18 - RAND USR 15363:REM: .....	9
19 - SAVE .....	9
20 - ATTR .....	9
<b>V - COMANDOS DE IMPRESSORA</b> .....	10
1 - LPRINT .....	10
2 - COPY .....	10
<b>VI - ARQUIVOS SEQUENCIAIS E ALEATÓRIOS</b> .....	10
1 - TIPOS DE ARQUIVOS .....	10
2 - ARQUIVOS SEQUENCIAIS E ALEATÓRIOS .....	11
3 - COMANDOS DE ARQ. SEQ. E ALEATÓRIOS .....	11
<b>VII - FITA CASSETE – DISQUETE</b> .....	14
<b>VIII - DISCO – DISCO</b> .....	15
<b>IX - INTERFACE DE IMPRESSORA</b> .....	15
<b>X - TRATAMENTO DE ERROS</b> .....	16
<b>XI - APROFUNDANDO-SE NA IDS-91</b> .....	16
<b>XII - ESQUEMAS</b> .....	23
<b>XIII - LAYOUT</b> .....	27
<b>XIV - FOTOS</b> .....	28
<b>XV - RELATORIO BETA</b> .....	32
1 - GENERALIDADES .....	33
2 - CIRCUITO DE APOIO .....	37
3 - CIRCUITO OPERACIONAL .....	44
4 - AVISOS .....	46

# I. APRESENTAÇÃO

Parabéns, você acaba de adquirir o mais útil dos periféricos para seu TK-90X e TK-95. Abra cuidadosamente a embalagem que contém a sua **IDS-91 (Interface de Drive Synchron)**, com ela você encontrará o seguinte material:

- Este manual de instruções;
- Uma interface de drive IDS-91 com o cabo para drive já conectado
- Um disquete com programas utilitários

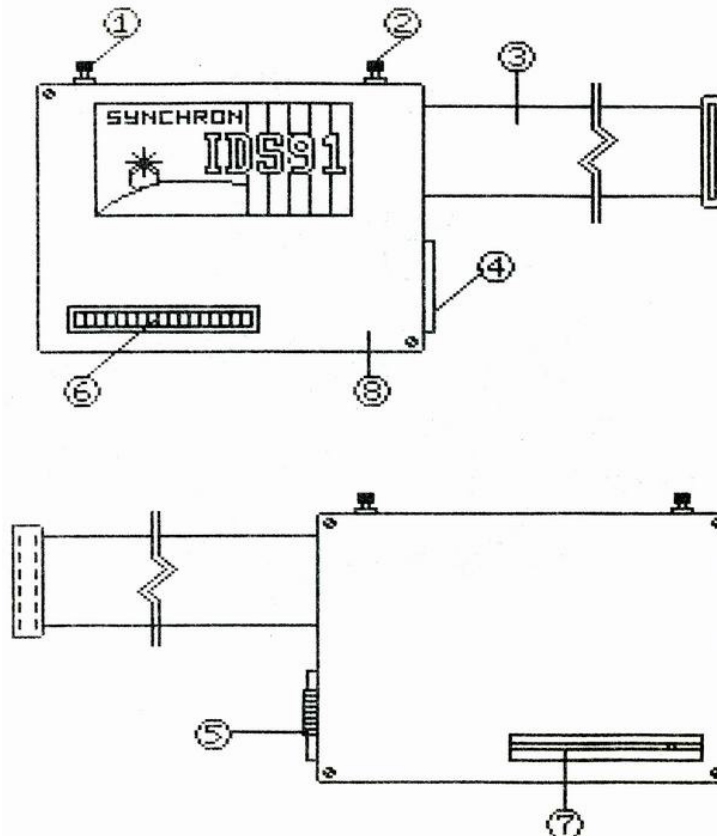
**NÃO LIGUE NADA AINDA!!!**

Tenha paciência, antes de mais nada leia atentamente às instruções que seguem:

Reconhecendo as partes:

Guie-se pela Figura 1 para as descrições a seguir:

- 1- **Botão de Reset:** Acostume-se a usá-lo sempre que precisar reiniciar o sistema, ligar e desligar a fonte reduz a vida útil de seu equipamento.
- 2- **Botão 48K:** Sua função é fazer uma cópia no disquete, de toda a RAM do micro.
- 3- **Cabo de drive:** Está conectado no interior da IDS-91
- 4- **Conector de impressora:** padrão Centronics.
- 5- **Plug de Alimentação:** alimenta o micro com 9 VDC nominais.
- 6- **Conector para o micro:** encaixa-se firmemente no slot de expansões na traseira do micro, na operação de encaixe verifique que há uma pequena placa guia neste conector, sua finalidade é impedir uma conexão defasada que pode resultar em danos tanto para o micro quanto para a interface.
- 7- **Extensor do slot do micro:** É uma cópia quase igual ao slot de expansões na traseira do micro, apenas o sinal IOREQ é filtrado pela IDS-91 para garantir que nenhum outro periférico tenha acesso, enquanto a IDS-91 estiver fazendo uma operação de I/O.
- 8- **Caixa:** Abriga a placa principal e extensora da IDS-91.



**FIGURA 1**

## II. INTERLIGANDO O EQUIPAMENTO

- 1- Mantenha o micro desligado.
- 2- Conecte a IDS-91 no barramento de expansão do micro (fig.2). Fique atento para a plaquinha guia que deverá encaixar-se com precisão no corte do slot de expansão de seu micro.
- 3- Conecte o plug de alimentação da fonte de seu micro no plug da IDS-91.
- 4- Conecte o cabo de drive no conector de seu drive.
- 5- Certifique-se que a TV, o drive e o micro estejam ligados à rede elétrica de sua residência. Verifique também as tensões de 110 e 220 V.
- 6- Coloque o disquete no drive e deixe-o com a tampa aberta.
- 7- Ligue a TV, o drive e por último o micro.

(1) Providencie para que a sua TV já esteja previamente sintonizada com o canal de seu micro. Veja também se os cabos estão firmemente encaixados

(2) O led do drive deverá acender e, após alguns segundos apagar-se. **Se o led ficar aceso permanentemente, desligue o micro de imediato e inverta a conexão do cabo no drive!!!**

(3) Ao mesmo instante que o led apagar-se, surgirá no canto inferior esquerdo da tela o prompt da IDS-91 seguido pelo cursor piscante do TK. Se a tela apresentar listas pretas horizontais e alguns quadrinhos coloridos, acione o botão reset. **Cuidado para não acionar o botão 48K no lugar do botão do reset!!!** Caso isso ocorra com a tampa do drive fechada e com o disco desprotegido para escrita, o botão mágico poderá sobrepor informações sobre as previamente existentes no disco.

- 8- Aparecendo o prompt e o cursor, tudo correu bem, caso contrário, consulte o guia de pequenos defeitos, ou procure a assistência técnica.
- 9- Feche a tampa do drive. Seu equipamento está preparado para uso.

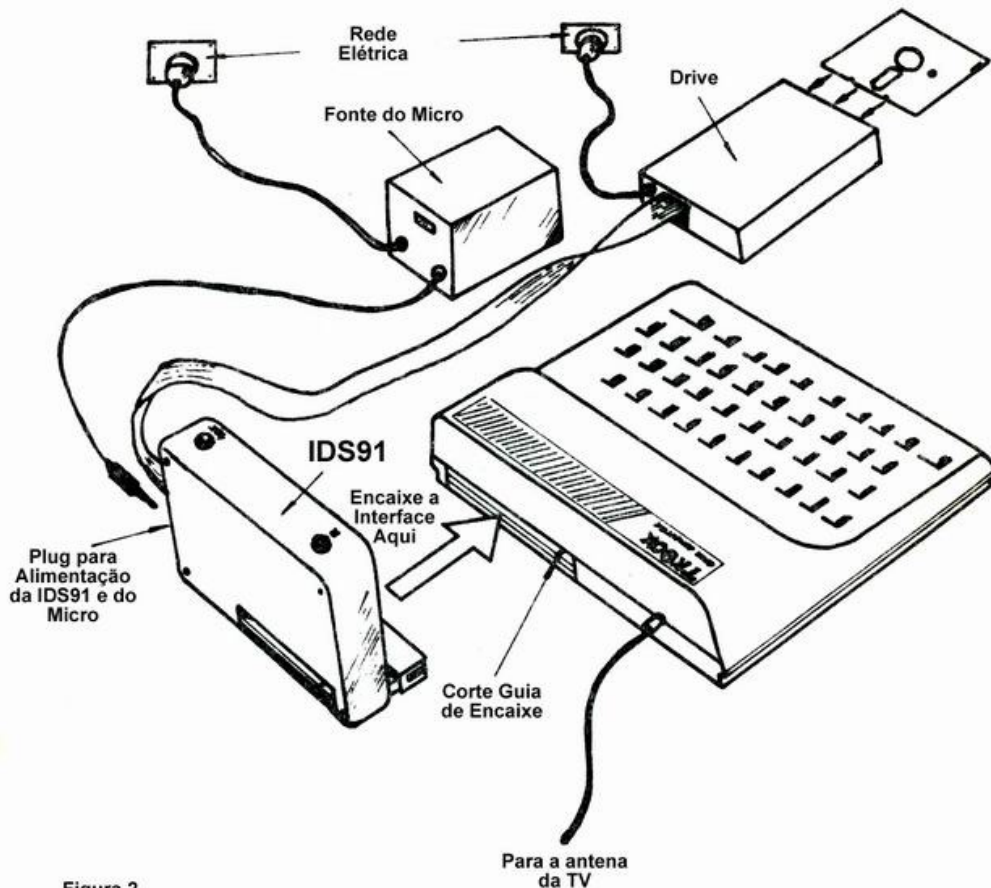


Figura 2

### III. CONVENÇÕES

Neste manual, adotaremos as seguintes convenções:

1. **<comando>** tudo o que estiver entre os sinais de maior e menor significa uma tecla apertada. Ex: **<go to>**.
2. Se não for especificado o contrário, os comandos funcionam de modo semelhante ao do BASIC.

### IV. COMANDOS DO D.O.S.

D.O.S. é o sistema operacional de disco (Disk Operating System), na IDS-91 este programa está gravado permanentemente em uma memória dentro da interface, como existe um sistema contínuo de revezamento entre a memória IDS-91 e a do micro isto significa que a memória RAM não é utilizada para armazenamento do D.O.S.

Reconhecemos que o D.O.S. está ativo em determinado momento pela presença do prompt **A: >**, **B: >**, **C: >** e **D: >** seguido pelo cursor **█**, sempre que o sistema é inicializado, o drive ativado será o 'A'

D.O.S. ativo significa que qualquer comando introduzido pelo teclado será interpretado pelo D.O.S. Vamos começar então pelo comando que nos permite voltar ao BASIC.

#### 1- RETURN:

Digite **<return>** seguido de **<enter>** para retornar ao BASIC normal do TK. No primeiro retorno a tela de abertura do TK será apresentada

#### 2- RAND USR 15360:

**<rand><usr>** 15360 **<enter>** leva o usuário para o sistema operacional, e o prompt do D.O.S. voltará a surgir na tela.

#### 3- CAT:

Para saber quais programas estão gravados no disco, usamos:

**<cat>** no, ds

Onde:

no e ds são parâmetros opcionais.

Após alguns instantes serão enviadas as seguintes informações:

```
TITULO: NOME
N1 arquivo (s)
N2 apagado (s)

D:arquivo 1<T> S1:arquivo 4 <T> S2
D:arquivo 3<T> S3:arquivo 4 <T> S4
"
"
S livre (s)
```

Onde:

NOME é o nome do disquete atribuído pelo usuário na formatação,

N1 é o número de arquivos no disco,

N2 é o número de arquivos apagados (ver o comando move)

D: é o drive selecionado para o diretório,

<T> é o tipo de arquivo (B=BASIC, C=código de máquina, D=matrizes e #=dados),  
S1...S4 é o número de setores que cada arquivo está ocupando no disco  
S é o número de setores ainda disponíveis no disco

4 canais são abertos para a inicialização do TK:

Canal #0: entrada de dados via teclado, saída via parte inferior da tela  
Canal #1: imprime na parte inferior da tela, normalmente usado pelo editor BASIC  
Canal #2: Imprime na parte de cima da tela, não possui rotina de entrada de dados, normalmente usado pelo comando <Print>  
Canal #3: envia dados para a impressora, normalmente utilizado pelo comando LPrint. Não possui rotina para a entrada de dados, quando a IDS-91 está conectada, a saída deixa de ser para a impressora ZX e passa a ser via porta paralela centronics

Se omitida a especificação de canal, o diretório será enviado para o canal #2 (tela).

Veja bibliografia no final deste manual para ter maiores informações sobre canais do BASIC.

#### 4- Comando \* (seleção do drive):

Quatro drives podem ser conectados ao mesmo tempo na interface IDS-91, estes drives são denominados A:, B:, C: e D:. Ao ser ligada à interface assume o drive A: automaticamente e assim fica até que o usuário queira modificar o drive corrente, isto pode ser feito de duas formas diferentes:

1- Especificando o drive em uma linha de comando, ex.:

```
A> load "B: SYNCHRON"
```

Após a execução do comando, o drive corrente continua sendo o A:

2- Usando o comando "\*", neste caso o drive corrente passará a ser específico no comando e a letra correspondente será apresentada na linha de edição seguida de ">".

```
* do
```

Onde do= "A\$", "B\$", "C\$", "D\$", ou conforme a convenção.

#### 5- FORMAT:

Antes de ser usado, o disquete precisa ser formatado, isto quer dizer que o D.O.S. irá marcar magneticamente as trilhas e os setores na superfície do disco, posteriormente estas marcações servirão de guia para os arquivos serem encontrados.

O comando format verifica as características do drive e formata o disco de acordo com elas.

```
A> format "drive: titulo"
```

Onde:

drive = A: ... D:

titulo = título do disquete, a formatação será no drive especificado, ausência de especificação formatada no drive corrente. Ex:

```
A> format "C: IDS-91"
```

O disquete no drive C: será formatado e receberá o título "IDS-91"

Para formatar um disco com face simples use:

```
A> format "drive: $titulo"
```

## Atenção !!!

A formatação destruirá de forma irrecuperável , tudo o que estiver previamente gravado no disco.

Após a formatação, a tela mostrará o seguinte:

```
T i t u l o  
S1/S2
```

Onde: S1 = número de setores livres (bons).  
S2 = número total de setores.

## 6- LOAD:

Carrega um arquivo do disco para a memória.

```
<load> S$  
<load> S$ <code>  
<load> S$ <code> endereço  
<load> S$ <data> A$ ( ),...
```

Onde S\$ é uma string ou expressão.  
O token *screen\$* deve ser substituído por 16384,6912

## 7- RUN:

Equivalente ao comando `load` seguido de `run`, se o programa for salvo com o número de linha (`save S$ line M`) a execução começa por ela, caso contrário a execução será a partir da primeira linha do programa.

```
<run> S$  
<run> S$ <code>
```

A segunda forma permite a execução de um programa em linguagem de máquina a partir do endereço inicial do bloco, se o endereço inicial for outro, então deve ser usado o comando `<rand><usr>` endereço.

## 8- ERASE:

Apaga arquivos no disquete.

Para apagar um arquivo que não seja em BASIC, o tipo de arquivo deverá ser especificado.

```
<erase> S$ tipo
```

Onde tipo = `<code>`, `<data>` ou `<#>`

Devido a forma de armazenamento no disquete, utilizada pelo D.O.S., o arquivo que for apagado, e estiver entre outros arquivos, deixará um espaço vazio que não será aproveitado para o armazenamento de outros arquivos, isto ocorre porque o D.O.S. sempre grava um arquivo após o último que foi gravado, ou seja, os setores são preenchidos na ordem crescente dentro das trilhas.

Quando houver espaço vazio no disquete o diretório acusará um ou mais arquivos apagados, para sanar este inconveniente existe o comando `move` descrito a seguir.

## 9- MOVE:

Permite compactar os arquivos no disco quando há espaços deixados pelo comando `erase`, liberando mais espaço no fim do disquete.

## 10- MERGE:

Funciona exatamente como o comando merge do cassete, acrescentando um programa BASIC a outro já existente. As linhas já existentes com o mesmo número da nova , serão substituídas pela atual.

```
<merge> S$
```

Também impede a auto-execução de um programa salvo com save ... line ...

## 11- NEW:

Permite a mudança de nome de um arquivo, se não for arquivo em BASIC o tipo deve ser especificado:

```
<new> "novonome", "velhonome"<tipo>
```

## 12- COPY:

Usado para execução de cópias de arquivos de um drive para outro, se não for arquivo BASIC o tipo deve ser especificado.

```
<copy> "drivedestino:novonome", "drivefonte:velhonome"<tipo>
```

Se no "drivedestino:" houver algum arquivo com o mesmo nome que o "novonome" o usuário será perguntado se deseja ou não substituí-lo pelo atual.

Para copiar TODOS os arquivos de um drive para outro use:

```
<copy> "drivedestino: *", "drivefonte: *"
```

Para copiar arquivos com apenas um drive ou para se obter uma cópia-imagem para backup o usuário dispõe do programa copyfil no disco que acompanha a interface.

Este comando precisa ser de, no mínimo, 4 kbytes livres na memória para ser executado, caso contrário a mensagem "ram lotada" será emitida e o comando abortado.

## 13- PEEK:

Permite ler qualquer setor de um arquivo e transferi-lo para uma localização na RAM.

```
<peek> S$ <tipo> enderecoram, setor
```

S\$ contém o nome do arquivo, enderecoram e o endereço na ram onde o setor será armazenado e setor e o número no arquivo do setor desejado.

## 14- POKE:

Armazena o conteúdo de uma área de ram em um setor especificado de um arquivo em disquete

```
<poke> S$<tipo> enderecoram, setor
```

## 15- 40:

Caso o drive utilizado seja antigo, com o mecanismo de acionamento da cabeça tipo CAM, o D.O.S. pode assumir erroneamente, nos testes iniciais para se determinar o tipo de drive, que o drive é de 80 trilhas, o comando 40 sana o problema determinado indicando que o drive corrente é de 40 trilhas.



## 16- 80:

É o contrário do anterior, atualmente pouco utilizado.

## 17- GOTO:

Exclusivamente usado para carregar um programa salvo pelo "Botão 48K". Durante o processo, algumas linhas estranhas aparecerão na tela mas fazem parte do processo e logo após desaparecerão.

```
<go to> "nome" <code>
```

Onde "nome" é o nome do arquivo dado pelo usuário conforme será explicado no capítulo VII.

## 18- RAND USR 15363: REM:

Usado para execução de comandos do D.O.S. a partir do BASIC.

A key-word **REM**, presente na linha de comando, impõe duas restrições ao programa:

- 1- Só pode existir um comando D.O.S. por linha BASIC.
- 2- Este comando tem que ser o último da linha.

```
<rand> <usr> 15363: <rem> : comando
```

## 19- SAVE:

Salva um programa em BASIC, blocos de bytes ou matrizes no disquete

```
<save> D$  
<save> D$ <line> linha  
<save> D$ <code> inicio, comprimento  
<save> D$ <data> A$ ( )  
<save> D$ <code> 16384, 6912
```

Para salvar telas não é possível usar <screen\$>, daí a existência da última forma.

## 20- ATTR:

Muda os atributos Ink e paper da tela sem alterar seu conteúdo

```
D$: <attr> paper, ink
```

Onde:

D\$ = A:, B:, C: ou D:

Paper = 0,1,...,7

Ink = 0,1,...,7

## V. COMANDOS DE IMPRESSORA

### 1- LPRINT:

O TK pode suportar dois tipos de impressoras:

- 1- ZX (ou equivalente Alphacom 32 e Timex 2040),
- 2- Paralela padrão Centronics

Quando o sistema micro – IDS-91 é ligado, o canal #3 é conectado a rotina da interface paralela, esta situação pode ser alterada por este comando:

```
A) <lprint>#<enter> ou  
<rand> <usr> 15363: <rem> : <lprint>#
```

Não confundir a segunda forma com o comando BASIC <LPRINT> que é usado para enviar dados a impressora.

Este comando só é usado para mudar o tipo de impressora para a qual serão enviados os dados de <copy>, <llist> e <print> do BASIC.

### 2- COPY:

Já vimos este comando quando tratamos da cópia de arquivos do D.O.S., veremos agora uma nova aplicação para este comando na interface da impressora.

Existem dois modos de utilização deste comando com a impressora paralela:

#### 1- “dump” de texto:

Tudo o que estiver na tela e for texto será enviado a impressora, os caracteres gráficos, UDG e figuras serão substituídos por espaço

```
<rand> <usr> 15363: <rem> : <copy>
```

#### 2- “dump” total:

Transfere texto, UDG e gráficos da tela para a impressora (exceto os atributos de cor).

```
<rand> <usr> 15363: <rem> : <copy> <screen#>#
```

Onde # = 1 ou # = 2

Se # = 1 a cópia será no tamanho 1x1.

Se # = 2 a cópia será no tamanho 2x2.

A impressora deve ser gráfica e capaz de reconhecer comandos padrão Epson de densidade simples.

## VI. ARQUIVOS SEQUENCIAIS E ALEATÓRIOS

### 1- Tipos de Arquivos:

Como já visto anteriormente, temos os seguintes tipos de arquivos:

- BASIC (B)
- Código de máquina (C)
- Dados de Matrizes (D)
- Dados seqüenciais e aleatórios (#)

Trataremos a seguir desse último tipo.

## 2- Arquivos seqüenciais e aleatórios

**Arquivos seqüenciais:** Os itens serão recuperados na ordem em que foram originalmente armazenados. Significa dizer que, uma vez lido o item 100, não será possível ler o item 99 sem retornar a leitora a partir do item 1.

**Arquivos aleatórios:** O acesso ao item é feito mediante referência ao seu número de posição dentro do arquivo. O tempo de acesso do primeiro item é praticamente o mesmo de acesso ao último item.

Uma limitação do tipo de acesso aleatório é que todos os itens têm que ter o mesmo comprimento, especificado no momento em que é criado o arquivo, se for armazenado um item menor do que o espaço previsto, este será acrescido de um enter no seu final (**CHR# 13**), se for maior, serão "cortados" todos os caracteres excedentes.

Para usar um arquivo devemos primeiro abri-lo e ao término do uso fecha-lo, fechar um arquivo é essencial, caso isso não seja feito corre-se o risco de perda de dados que ainda restarem no "buffer" e o D.O.S. não poderá encontrar o fim do arquivo nas operações de leitura.

Quando abrimos um arquivo estamos ligando-o a um dos canais de informação do TK. Existem 16 canais no TK destinados a troca de informações entre diversos periféricos do computador. Destes, quatro já são usados nos periféricos-padrões, portanto restam-nos 12 canais para uso dos arquivos.

Até 12 arquivos podem ser abertos simultaneamente, e verificar se existe memória disponível antes de abrir um arquivo, porque o D.O.S. aloca um "buffer" de 292 bytes de memória para cada arquivo aberto.

Os arquivos seqüenciais e aleatórios podem ser apagados, renomeados ou copiados com os comandos utilizados para outros tipos de arquivos, por exemplo:

```
<erase> "nome"#  
<new> "novonome", "velhonome"#  
<copy> "drive:novonome", "drive:velhonome"#
```

## 3- Comandos de arquivos seqüenciais e aleatórios

### 1- Open #:

Sua função é abrir um arquivo de dados, não é possível abrir um arquivo seqüencial para escrita e leitura ao mesmo tempo como ocorre com os arquivos aleatórios.

Para abrir um arquivo seqüencial para escrita usamos:

```
<open #> canal, "nome", #
```

Onde:

canal é o número do canal;

nome é o nome do arquivo o qual pode ser também uma expressão ou variável, como ocorre com todos os comandos.

Usa-se o comando print # para escrever no arquivo, após o final da escrita o arquivo deve ser fechado, para abri-lo novamente para leitura usamos:

```
<open #> canal, "nome", R
```

Os comandos input e inkey\$ serão usados para a leitura dos dados do arquivo, ao término da leitura devemos fechar os arquivos.

Os arquivos aleatórios podem ser abertos tanto para leitura quanto para escrita simultâneos.

```
<open #> canal, "nome"<RND>, tamanho
```

Onde:

tamanho é o tamanho do registro a ser usado e o valor máximo de tamanho será 255.

O registro (item) é armazenado na forma de um string simples, se esta for menor que o tamanho especificado para o registro então receberá um carriage return no fim (`chr$ 13`), se for maior que o registro não haverá o byte `chr$ 13` e o excedente será perdido.

## 2- Close #

Permite que o arquivo seja fechado, bastando pra isso fechar o canal com o qual o arquivo foi aberto.

```
<close #> canal
```

## 3- Print #

Escreve dados no arquivo, deve ser usado apenas no BASIC TK, o que significa que não precisa ser precedido pela seqüência `RAND USR 15363: REM:`

```
<print #> canal; reg1,...regN
```

Onde:

reg1...regN são optativos no caso de arquivos seqüenciais.

Os apóstrofos são necessários para separar os registros.

Para escrita em arquivos aleatórios usamos a forma:

```
<print #> canal, (nr), reg
```

Onde:

nr é o numero do registro.

reg é um item a ser escrito no registro, no caso de arquivos aleatórios só podemos escrever um registro de cada vez.

## 4. Input #

Obtém um item do arquivo e o aloca a uma variável especificada como se tivesse sido digitado do teclado, para arquivos seqüenciais o formato é:

```
<input #> canal, variavel
```

Onde:

Variável será string ou numérica conforme o caso, atentar para a consistência dos dados evitando ler um item numérico em uma variável string ou vice-versa.

No caso de arquivos aleatórios temos:

```
<input #> canal; (nr), variavel
```

## 5. Inkey #

Como ocorre no teclado, esta função devolve apenas um caractere obtido do arquivo.

```
<inkey$> <#> canal
```

## 6. O Fim do arquivo

A tentativa de ler um item após o fim de um arquivo seqüencial resultará na mensagem de erro "8 FIM DE ARQUIVO" com execução do programa sendo interrompida, para se evitar este problema deve-se simular a função EOF (End OF File), que devolve o valor 1 se o ultimo caracter do arquivo já foi lido:

```
1 DEF FN B(J)=PEEK 23631+256*PEEK 23632 +  
PEEK (23574+2*J)+ 256 * PEEK (23575+2*J) -1  
2 DEF FN E(J)=PEEK (FN B(J)+13)= PEEK (FN  
B(J)+27) AND PEEK (FN B(J)+14)= PEEK (FN  
B(J)+28)
```

Se o ultimo caracter do arquivo tiver sido lido, a função FN E(J), onde J é o numero do canal a que se liga o arquivo, devolverá o valor 1, caso contrario devolverá o valor 0.  
Quanto aos arquivos aleatórios, se tentarmos uma leitura após o fim do arquivo ocorrerá erro de sintaxe na linha do **INPUT**.

## 7. Exemplos:

### Arquivo Seqüencial:

O programa a seguir escreve 5 itens na tela e os numera para uma referencia futura. Em seguida lê o arquivo e imprime os itens na tela.

```
1 RAND USR 15363: REM: OPEN #4,"TSTARQ", W
2 FOR I=1 TO 5: PRINT #4,"ITEM NR"; I
3 NEXT I: RAND USR 15363: REM: CLOSE #4
4 REM-----
5 RAND USR 15363: REM: OPEN#4, "TSTARQ", R
6 FOR I=1 TO 5: INPUT #4;A$
7 PRINT A$: NEXT I
8 RAND USR 15363: REM: CLOSE #4
```

### Arquivo Aleatório:

Neste caso os itens são recuperados aleatoriamente 10 vezes através do uso da função "RND" do BASIC TK.

```
1 RAND USR 15363:REM:OPEN #4,"TSTARQ1"
RND,10
2 FOR I=1 TO 5: PRINT #4; (I), "ITEM NR"; I
3 NEXT I: RAND USR 15363: REM: CLOSE #4
4 REM-----
5 RAND USR 15363:REM:OPEN #4,"TSTARQ1"
RND,10
6 FOR I=1 TO 10:INPUT #4; (INT (RND*9)+1),
A$
7 PRINT A$: NEXT I
8 RAND USR 15363: REM: CLOSE #4
```

### Fim de Arquivos:

Neste caso são armazenados 10 itens em um arquivo de 10 registros, observe o que ocorre quando se tenta ler o registro 10 inexistente, depois mude a linha 6 para que o programa leia o item 10, por exemplo, e veja o que ocorre, não esqueça de garantir que o arquivo esteja fechado antes de rodar o programa depois da primeira vez.

```
1 RAND USR 15363:REM:OPEN #4,"TSTARQ2"
RND,10
2 FOR A=1 TO 10: PRINT #4; (A), ? : NEXT A
3 RAND USR 15363: REM: CLOSE #4
4 REM-----
5 RAND USR 15363:REM:OPEN #4,"TSTARQ2"
RND,10
6 INPUT #4; (20), B
7 PRINT B
8 RAND USR 15363: REM: CLOSE #4
```

## VII. FITA CASSETE – DISQUETE

Para transferência de uma fita cassete para o disco podemos adotar 3 procedimentos conforme o caso:

### 1- Manual:

Neste processo o usuário deverá carregar na memória cada um dos blocos componentes do programa e ir salvando um a um, usando o endereço inicial e tamanho. Este processo, apesar de ser o que apresenta melhores resultados, exige um bom conhecimento do sistema operacional do micro e de linguagem de máquina, isto inviabiliza sua descrição detalhada neste manual.

### 2- Copytape:

É um dos programas utilitários que se encontram no disquete que acompanha seu equipamento.

1. Inicialize seu sistema com o disquete IDS-91, escolha a opção 1 na parte inferior da tela de apresentação da SYNCHRON.
2. Posicione o gravador no início do programa a ser transferido e ligue-o.
3. Enquanto cada bloco estiver sendo transferido siga as mensagens de ligar e desligar o gravador.
4. Reset a IDS-91 e rode o programa COPYFIL.
5. Utilizando o comando **CAT** do COPYFIL anote os dados de cada bloco do programa carregado.
6. Usando o comando **MERGE** do D.O.S., carregue um a um os blocos, substituindo os **SAVE** e **LOAD** de fita por **SAVE** e **LOAD** no formato disco.
7. Apague a versão fita que está no disquete e, em seu lugar salve a versão disco.

A limitação deste método é que somente programas do HEADER (pequeno bloco de 17 bytes gravados previamente pelo **SAVE** em fita e que contém informações sobre o programa) podem ser transferidos.

### 3- Botão 48K

É o meio mais simples e rápido para a transferência de programas para disco, se o programa se utilizar do modo 1 de interrupção do Z-80 o Botão 48K não funcionará.

O nome deste botão vem do seu modo de operar em que toda a memória RAM do micro (48K) é transferida para o disco na configuração em se encontrar no instante em que o botão for pressionado.

Seqüência para a correta operação do Botão 48K.

1. Formate um disco em face simples.
2. Coloque o disco no drive A
3. Retorne ao BASIC e limpe toda a memória usando **RAND USR 0**.
4. Carregue e rode o programa do cassete normalmente.
5. Aperte o Botão 48K por um instante.
6. Aguarde o fim da operação após a qual o programa deverá continuar rodando normalmente. Caso isto não ocorra pode ser indicio de que a transferência não foi bem sucedida.
7. Teste o resultado da transferência usando o comando **GOTO** do D.O.S.

Observações:

1. O nome do arquivo é " " e seu tipo é CODE.
2. Você pode mudar este nome por outro a seu gosto, usando o comando **NEW** do D.O.S.
3. O Arquivo salvo no disco consome 192 setores e se instala nas trilhas de 1 a 12 do disco, por isso nessa área não deve haver nenhum outro arquivo sob pena de completa destruição.

4. Podemos copiar este arquivo para outro disco usando o comando **COPY** do D.O.S. ou do COPYFIL, recomendamos que isso seja feito para que o disco fique simples e liberado para outras operações com o Botão 48K.
5. Se um programa não funcionar após sua carga em disco pelo botão 48K, pode estar ocorrendo do programa estar usando sua própria rotina de teclado, diferente da rotina do sistema operacional do TK. Neste caso mude o nome do programa para que comece com um \$.

## VIII. DISCO – DISCO

O D.O.S. dispõe de comandos próprios para cópias de arquivos de disco para disco, como já vimos anteriormente. Um programa disponível no disco de utilitários que acompanha a IDS-91, chamado COPYFIL, poderá auxiliar o usuário na realização de cópias, principalmente quando só dispuser de um drive, os comandos abaixo, entre outros que pertencem ao COPYFIL, apresentam características dignas de mencionarmos:

- 1- **CAT**: apresenta o nome dos arquivos linha após linha, além do endereço de início, comprimento e **LINE**.
- 2- **COPY** S"nomenovo", "nomevelho": copia arquivos em sistemas com um só drive, como o arquivo é copiado em blocos de 24 bytes, dependendo do tamanho serão necessárias mais de uma troca de discos, o sistema emite todas as mensagens necessárias para orientar o usuário. **Disco Fonte** é o que contém o arquivo a ser copiado, **Disco Destino** é o que receberá o arquivo.
- 3- **<COPY>** B: Para obter cópias de segurança (BACKUP) de nossos discos usamos este comando, neste caso ocorre uma cópia-imagem do disco fonte para o disco destino, ou seja, a cópia é feita trilha por trilha, serão necessárias 14 operações de troca de discos, se forem de 40 trilhas, uma vez que o comando copia de 6 em 6 trilhas (24 Kbytes).

## IX. INTERFACE DE IMPRESSORA

- 1- Padrão Centronics.
- 2- Software de operação em EPROM (embutido na interface).
- 3- Configuração na impressora: ao receber o código de retorno de carro a impressora não deveria dar salto de linha (LF).
- 4- Quando o sistema é ligado à saída do canal #3 (impressora) é desviada para a interface paralela.
- 5- O comando **NEW** ou **PRINT USR @** a partir do BASIC realocará o canal #3 para a impressora ZX. Para acionar a impressora novamente use o comando:  

```
<rand> <usr> 15363: <rem>: <lprint>1
```
- 6- Forma como são tratados os caracteres enviados a interface:
  - (1) Caracteres de controle do BASIC TK:
    - (A) Os caracteres ASCII de 16 a 22 (INK a AT) serão ignorados.
    - (B) O código ASCII 23 (TAB) posicionará a cabeça na coluna especificada pelo argumento, dentro da linha, o maior valor para tabulação é 127.
  - (2) UDG'S e caracteres gráficos:  
O padrão EPSON será acionado e será impresso como são vistos na tela.
  - (3) Palavras reservadas do BASIC (Tokens):  
Os códigos ASCII de 165 a 255 serão impressos expandidos, por ex.: o código 181 será impresso com **"ASN"**.
  - (4) Acentuação e escolha de padrão:  
O software providenciará o caracter acentuado correspondente ao padrão escolhido se receber uma seqüência correta de caracteres (caracter+backspace+acento).
  - (5) Envio de caracteres de controle à impressora:  
Para permitir que os caracteres de controle sejam enviados á impressora devemos usar o comando:

```
<POKE> 23681, Numero de Caracteres
```

Os caracteres serão enviados à impressora sem receberem qualquer tratamento do software, por ex. : O Tokem 181 será enviado diretamente e não mais os caracteres **ASN**, a função **CHR#** se comportará normalmente após terem sido enviados o número de caracteres especificado.

(6) Uso com linguagem de máquina:

Para imprimir o caracter cujo código esteja no acumulador usamos a instrumentação ASSEMBLY:

```
CALL 3CF5H   OU   RST #10H
```

Esta última depois de o canal #3 ter sido aberto.

OBS.: existe um JUMPER interno na IDS-91 o qual permite a escolha entre os padrões ABICOMP e MSX 1.1.

## X. TRATAMENTO DE ERROS

É o procedimento que o D.O.S. ou o BASIC adotam quando ocorre um erro qualquer no processamento dos programas ou comando, os erros podem ser de dois tipos:

- 1- Quando o D.O.S. esta ativo (sinal ">" na linha de edição):  
O comando é abortado e reapresentado para que o usuário faça as correções necessárias, uma mensagem é apresentada elucidando o tipo de erro.
- 2- Quando o BASIC está ativo:  
Alguns erros apenas causam a interrupção do processamento, mas não são seguidos de mensagem alguma, para se obter um número relacionado ao erro ocorrido use os comandos do D.O.S. na forma:

```
<let> variavel = <usr> 15363: <rem>: comando
```

Onde:

variável conterà um numero cuja interrupção é:

- (0) Comando executado sem problemas
- (1) Arquivo inexistente no disquete
- (2) Arquivo já existente no disco. Ocorre quando se tenta salvar um arquivo de mesmo nome e tipo já existente no disquete
- (3) Sem espaço no disco
- (4)
- (5) Número do registro acima do especificado
- (6)
- (7) Erro no disco
- (8) Erro de sintaxe nos comandos

Desta forma basta se verificar a variável para a tomada de decisões e impressão de mensagens nos programas.

## XI. APROFUNDANDO-SE NA IDS-91

### 1- Introdução

A IDS-91 é constituída principalmente por dois circuitos integrados: O controlador de disco flexível FDC 1793/7 e uma EPROM 27128 (16 Kbytes) contendo o D.O.S.

O FDC comunica-se com o DOS através das portas de entrada e saída de numero 31 (1F), 63 (3F), 95 (5F) e 127 (7F).

A EPROM é acessada através da faixa de endereço de 256 bytes entre 15360 e 15615 (3C00 e 3CFF) e, em determinados instantes pode ocupar os 16 Bytes iniciais do mapa de memória do TK.

A IDS-91 utiliza 7 portas de I/O e isso pode, ocasionalmente, resultar em conflito com outro periférico ligado ao expansor, o qual também pode se utilizar de uma destas portas; buscando



evitar tal problema o sinal IOREQ passa por um mecanismo de bloqueio que desativa o barramento de expansão nas operações de I/O da IDS-91 e desativa a IDS-91 nas operações de I/O com outros periféricos. Por este motivo recomenda-se que periféricos sejam conectados na IDS-91 e esta ao micro.

Como acima dito, existe um revezamento entre a EPROM da IDS-91 e a ROM do micro, não acontecendo nunca das duas estarem ativas ao mesmo tempo. Para a implementação deste revezamento existe um mecanismo de paginação das EPROMs. Contudo, para que o BASIC possa acessar o DOS é reservada a área de memória a que referimos no início, e que é sempre prioritária para o BASIC. Alguns jogos podem incompatibilizar com este mecanismo e, para isto, foi prevista a porta 252 (FC) de I/O, que faz com que a EPROM do DOS fique desabilitada mesmo naquela área dos 256 Bytes.

## 2- Principais entradas para o D.O.S.

15360 (3C00)	Chamada ao DOS pelo usuário. Permite a execução dos comandos em modo imediato.
15363 (3C03)	Chamada ao DOS pelo BASIC-TK, permite o acesso ao drive pelo programa através do comando já visto.
15467 (3C6B)	Ponto de entrada para paginar o restante da EPROM do DOS, preserva todos os registradores, mas deixa as interrupções desabilitadas.
15484 (3C7C)	Ponto de entrada para despagnar a EPROM do DOS, preserva todos os registradores e reabilita as interrupções.
15511 (3C97)	Ponto de entrada para habilitar I/O com a IDS-91 e desabilitar o conector de expansão, podemos acessar o FDC diretamente.
15517 (3C9D)	Executa a operação anterior de modo inverso.
15605 (3CF5)	Ponto de entrada para enviar a impressora paralela o caracter cujo código esteja no registrador A, não preserva nenhum registrador e deixa as interrupções habilitadas.

## 3- Algumas rotinas do D.O.S.

Após a chamada a 15467(3C6B), ponto de entrada que vai paginar a EPROM do D.O.S. por completo, teremos o D.O.S. em sua principal atividade que é o gerenciamento de todos os acessos ao disco. Seguem-se algumas de suas rotinas:

9073 (2371)	Lê o setor de configuração do disco na área destinada ao buffer da impressora (23296 a 23551) e ajusta as variáveis do sistema para o tipo do disco, dependendo contudo do conteúdo da variável em 23831(5D17), salva antes o conteúdo do buffer no setor 9 da trilha 0.
11983 (2ECF)	Aciona o drive corrente e lê a trilha em que a cabeça estiver posicionada, guardando-a nas variáveis adequadas. É necessário chamar esta rotina antes de chamar as rotinas de leitura e escrita de setor. Na entrada, o registrador B deve estar com o valor 0.
11990 (2ED6)	Realiza a leitura de setores do disco. Na entrada HL deve conter o endereço da RAM para onde os setores serão lidos. O registrador D deverá conter a trilha onde se localiza o setor e o registrador E, o setor correspondente. B deverá conter o número de setores a serem lidos. No caso de mais de um setor, DE contém a trilha e o setor iniciais sendo que o D.O.S. fará a cabeça avançar automaticamente.
12027 (2EFB)	O mesmo que o anterior para escrita de discos.
12429 (308D)	Seleciona o lado 0 do disco, é necessário antes de formatar o disco.
12440 (3098)	Seleciona o lado 1 do disco, também necessário antes de formatar o disco.

12447 (309F)	Formata a trilha do disco, a trilha será formatada com o número contido no registrador E, mesmo que o drive não esteja posicionado nesta trilha. Na entrada, as variáveis 23768 e 23769 (5CD8 e 5CD9) devem conter o valor 0.
15728 (3D70)	Seleciona o drive cujo número esteja no registrador A (de 0 e 3) para o drive corrente, se A estiver com o conteúdo da variável do sistema 23833 (5D19), será selecionado o drive default, ou seja, aquele cuja letra aparece quando o D.O.S. é chamado.

#### 4- Tabelas

16321 (3FC1)	Contêm os 21 comandos do D.O.S., um byte por comando com o código do mesmo.
16342 (3FD6)	Tabela com os endereços de execução dos 21 comandos do D.O.S., dois bytes para cada comando, na mesma ordem da tabela anterior.

#### 5- Formato da trilha do sistema: Trilha 0

A trilha 0 contém o diretório e informações sobre o tipo de formatação empregada no disco.

Setores 0 a 7	Contêm o diretório do disco, cada entrada do diretório compreende 16 bytes, logo podem existir no máximo 128 arquivos no disco.
Setor 9	Setor usado para armazenamento temporário da área do buffer da impressora, esta área passa então a ser usada pelo D.O.S. para descarregar seqüencialmente os setores do diretório durante uma busca de arquivo.
Setor 10	Usado pela rotina do Botão 48K.
Setor 11 a 15	Não utilizados pelo D.O.S., acessíveis somente por linguagem de máquina.

#### 6- Formato das entradas do diretório

Bytes 0 a 7	Nome do arquivo. Byte 0=1: O arquivo foi apagado pelo comando ERASE. Byte 0=0: Fim do diretório.
Byte 8	Letra correspondente ao tipo de arquivo (B,C,D, #).
Bytes 9 e 10	Tamanho do arquivo em bytes para tipo BASIC e endereço inicial do bloco para tipo CODE.
Bytes 11 e 12	O conteúdo varia de acordo com o tipo de arquivo: CODE= Comprimento do bloco, BASIC= Tamanho do BASIC...
Byte 13	Número de setores ocupados pelo arquivo.
Byte 14	Setor inicial do arquivo.
Byte 15	Trilha inicial do arquivo.

#### 7- Formato do setor de configuração

Bytes 0 a 244	Vazios ( 0 ).
---------------	---------------

Byte 255	Próximo setor livre no fim do disco.
Byte 226	Trilha do próximo setor livre.
Byte 227	Contém o tipo de formatação empregada no disco:
16H	80 trilhas face dupla.
17H	40 trilhas face dupla.
18H	80 trilhas face simples.
19H	40 trilhas face simples.
Byte 228	Número total de arquivos no disco incluindo os apagados.
Byte 229 e 230	Número de setores livres do disco.
Byte 231	Contém o valor 16 indicando que o disco é padrão BETA, este byte é checado pela rotina 2371H.
Bytes 232 e 233	Não usados, contêm 0.
Bytes 234 a 242	Não usados, contêm espaços.
Byte 243	Não usado.
Byte 244	Número de arquivos apagados no disco.
Bytes 245 a 252	Nome do disco conforme definido pelo comando FORMAT, se o nome tiver menos que 8 caracteres, os demais serão preenchidos com espaços.
Bytes 253 a 255	Não usados ( 0 ).

## 8- Formato das portas de entrada

Porta 31 (1F)	Registrador de status do controlador.
Porta 63 (3F)	Registrador de trilha do controlador.
Porta 95 (5F)	Registrador de setor do controlador.
Porta 127 (7F)	Registrador de dados do controlador.
Porta 255 (FF)	Porta status secundaria:
Bit 7	Leitura do pino 39(INTRQ) do controlador: 1 = Comando terminado. 0 = Comando em andamento.
Bit 6	Leitura do pino 38(DRQ) do controlador: 1 = FDC requisitando dados. 0 = FDC não tem dado disponível.
Bit 5	Leitura do sinal de BUSY da impressora paralela: 1 = Impressora ocupada. 0 = impressora livre.
Bit 4	Leitura do padrão de impressora: 1 = Padrão de impressão ABICOMP. 0 = Padrão de impressão MSX 1.1.
Bit 3 a 0	Não utilizados.

## 9- Formato das portas de saída

Porta 31 (1F)	Registrador de comandos do FDC.
Porta 63 (3F)	Registrador de trilha do controlador.
Porta 95 (5F)	Registrador de setor do controlador.
Porta 127 (7F)	Registrador de dados do controlador.
Porta 251 (FB)	O dado a ser impresso deve ser enviado para esta porta onde ficará armazenado.
Porta 252 (FC)	Porta de configuração da interface. A ULA do TK também será acionada pelos circuitos por se tratar de uma porta PAR.
Bits 0 a 2	Cor da borda da tela.
Bit 3	Aciona a saída de som pela TV.
Bit 4	Aciona saída MIC.
Bit 5	Não usado.

Bit 6	Se = 1 Desabilita acesso á EPROM do D.O.S. pela área prioritária (3000 a 3DFF).															
Bit 7	Se = 0 Habilita o acesso á EPROM do D.O.S. pela área prioritária. Se = 1 desabilita I/O com a IDS-91 e habilita I/O com o conector de expansão. Se = 0 habilita I/O com a IDS-91 e desabilita I/O com o conector de expansão.															
Porta 255 (FF) Bits 0 e 1	Porta de configuração secundária: Seleção do drive: <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">Bit 1</td> <td style="padding-right: 10px;">Bit 0</td> <td>DRIVE</td> </tr> <tr> <td>0</td> <td>0</td> <td>A</td> </tr> <tr> <td>0</td> <td>1</td> <td>B</td> </tr> <tr> <td>1</td> <td>0</td> <td>C</td> </tr> <tr> <td>1</td> <td>1</td> <td>D</td> </tr> </table>	Bit 1	Bit 0	DRIVE	0	0	A	0	1	B	1	0	C	1	1	D
Bit 1	Bit 0	DRIVE														
0	0	A														
0	1	B														
1	0	C														
1	1	D														
Bit 2	Reset do FDC. Se = 1 não ocorre RESET do FDC. Se = 0 ocorre RESET. Deve ser mantido em 1.															
Bit 3	Se = 1 Informa ao FDC que a cabeça está no disco pronta para atuar. Se = 0 A cabeça não está pronta, manter em 1.															
Bit 4	Se = 1 Seleciona o lado 1 do disco. Se = 0 Seleciona o lado 0 do disco.															
Bit 5	Strob da impressora, se em 1 vai para 0 e vice-versa. Indo de 0 para 1 e depois voltando para 0, envia os dados para a impressora (após testar se ela está pronta).															
Bit 6	Seleciona a densidade de formatação: Se = 1 Densidade simples (FM). Se = 0 Densidade dupla (MFM). Manter em 0.															
Bit 7	Paginação das memórias: Se = 1 o D.O.S. aciona a ROM BASIC. Se = 0 o D.O.S. aciona sua própria EPROM, o BASIC recebe os comandos via teclado e, ao teclar <ENTER>, o D.O.S. faz a interpretação.															

## 10-O sinal motor-on

Este sinal encontra-se ligado ao pino HLD do controlador, portanto o drive só será acionado se este pino estiver em 1, envie um comando tipo 1 ao FDC (Bit 3 em 1) antes de enviar qualquer comando de escrita ou leitura de setor, para que o motor seja acionado.

## 11-Formato do disco

O disco é dividido em 40 trilhas, cada uma com 16 setores. Cada setor contém 256 bytes. A trilha 0 pertence ao sistema e contém o diretório. Um disco pode armazenar um total de  $40 \times 16 \times 256 \times 2 = 327680$  bytes (320 Kbytes). Como a trilha zero não armazena dados do usuário temos  $327680 - 4096 = 323584$  Bytes úteis (316 Kbytes), estes cálculos se referem a discos de 360 K. Em drives de 3 1/2" ou nos drives de 5 1/4 " e 1,2 Mbytes a capacidade pode ser calculada de modo semelhante.

Os arquivos são armazenados a partir da trilha 0 do lado 1, setor a setor, na ordem crescente. Após o último setor ser preenchido, o D.O.S. passa para o primeiro setor do lado 0, na trilha 1 (a trilha 0 esta ocupada pelo diretório), o processo segue desta forma até que o disco seja totalmente preenchido ou que se atinjam o número limite de arquivos permitidos no disco (128 arquivos).

No disco, os setores seguem uma ordem de modo que ficam entrelaçados permitindo que o sistema tenha tempo de processar as informações enquanto o setor seguinte se aproxima da cabeça do drive. A seqüência física dos setores é: 1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16.

## 12-Variáveis do sistema

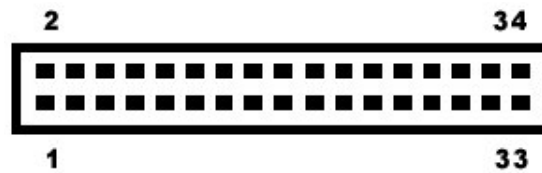
Ocupam a área de 23734 a 23845 (5CB6 a 5D25) num total de 112 bytes, e mais uma área de armazenamento temporário de 23846 a 23903 (5D26 a 5D60) totalizando 58 Kbytes, as duas áreas somam 170 Bytes.

5CC8 a 5CCB	Flags de configuração do tipo de disco presente em cada um dos quatro drives.
5CCC	Uso geral.
5CCD	Flag, armazena o Bit 7 do registrador de status do controlador e comanda ou não um atraso de 450ms antes de ler cada setor do disco.
5CCE	Flag, 00 = Leitura, FF = Escrita.
5CD7	Numero de trilhas de drive corrente.
5CD8	Flag durante o comando FORMAT. 00 = Verifica, FF = Não Verifica.
5CD9	Flag durante o comando FORMAT. 00 = Normal, FF = Teste.
5CDA	Flag durante o comando FORMAT. 00 = Face simples, FF = Face dupla.
5CDB e 5CDC	Armazenam resultados de expressões calculadas pelo acumulador de comandos do D.O.S.
5CDD e 5CE4	Workspace para nomes de arquivos de disco.
5CF4 e 5CF5	Setor e trilhas iniciais a serem lidos.
5CF6	Número do drive corrente selecionado.
5CFA a 5CFD	Flags do tipo de drive e step rate para cada um dos drives A, B, C, D.
5CFE	Comando a ser enviado ao controlador ( 88 ou A8 ).
5CFF	Armazenamento temporário do setor inicial.
5D00 e 5D01	Armazenamento temporário do endereço de carga.
5D02 e 5D03	Armazenamento temporário do par HL em chamadas a ROM BASIC.
5D04 e 5D05	Armazenamento temporário do par DE em chamadas a ROM BASIC.
5D11 e 5D12	Endereço do inicio do comando D.O.S. ( após <REM> ou diretamente do editor BASIC ).
5D13 e 5D14	Armazena a variável ERR_SP (23613 / 14), cujo valor é alterado pelo D.O.S.
5D15	Flag: 00 = D.O.S. foi chamado por RAND USR 15360. FF = D.O.S. foi chamado por RAND USR 15363.
	Impede a emissão de mensagens de erro.
5D16	Copia do dado enviado a porta de configuração secundaria 255 (FF).
5D17	Flag: 00 = Buffer de impressora não deve ser salvo no disco. FF = O buffer deve ser salvo e recuperado após comando.
5D18	Flag: 00 = Interface ONE não está presente. FF = Interface ONE está presente e as variáveis do microdrive devem ser salvas.
5D19	Número do drive DEFAULT (cuja letra aparece seguida do sinal ">").
5D1A e 5D1B	Endereço de retorno quando termina a execução dos comandos do D.O.S.
5D1C e 5D1D	Antigo valor do SP do Z-80, alterado em duas posições pelo D.O.S.
5D1F	Indica que a forma de ponto flutuante criada durante a analise da sintaxe dos comandos deve ser eliminada antes da representação do comando USUAR.

### 13-Pinagem dos conectores:

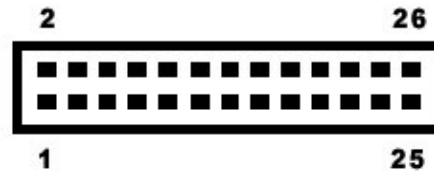
#### DRIVE (VISTO DE FRENTE)

Pinos 01 a 33	TERRA
Pino 02	N/C
Pino 04	N/C
Pino 06	Drive Select 3
Pino 08	INDEX
Pino 10	Drive Select 0
Pino 12	Drive Select 1
Pino 14	Drive Select 2
Pino 16	Motor On
Pino 18	Direction
Pino 20	STEP
Pino 22	Write Data
Pino 24	Write Gate
Pino 26	Track 00
Pino 28	Write protected
Pino 30	Read Data
Pino 32	Side Select
Pino 34	Ready

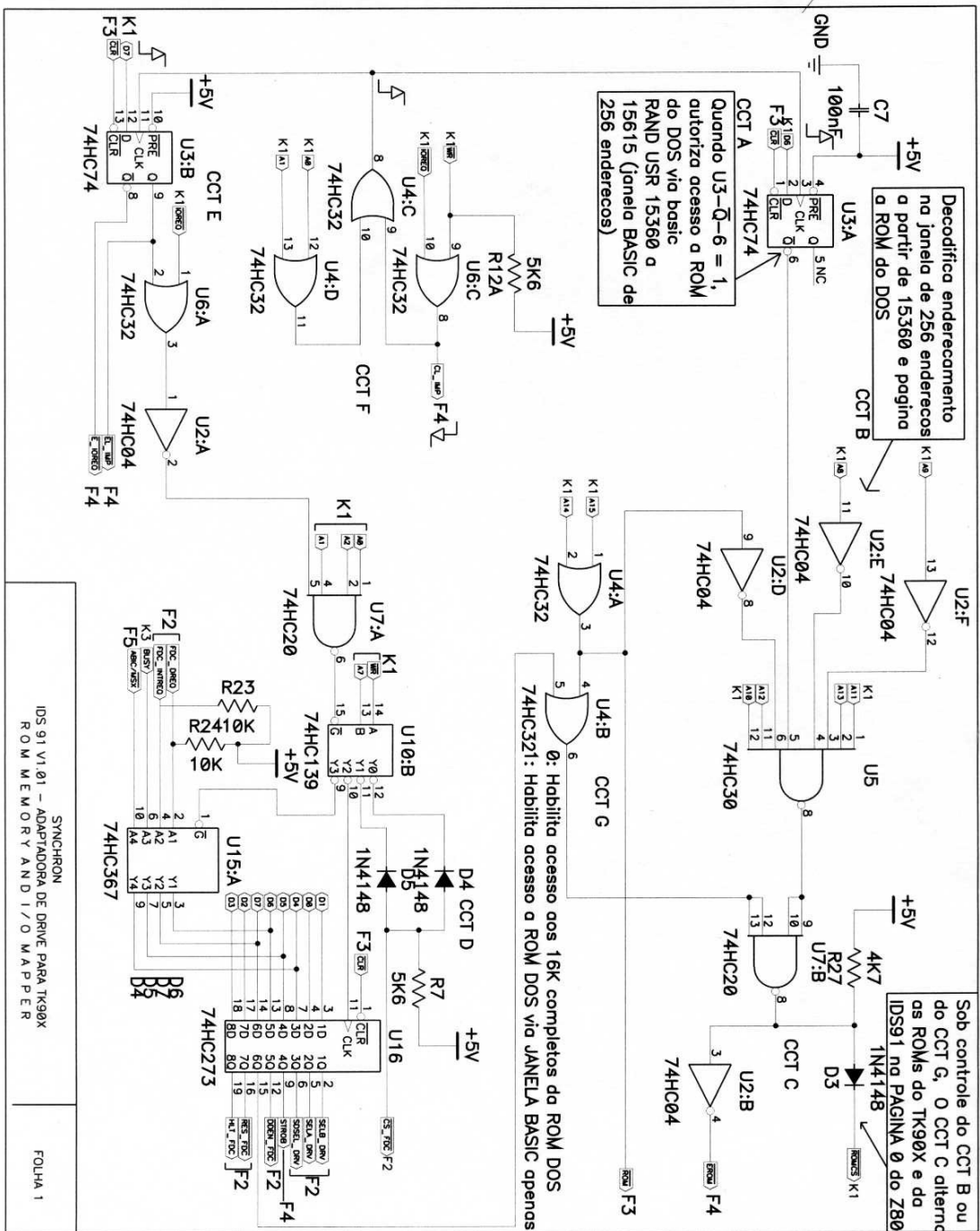


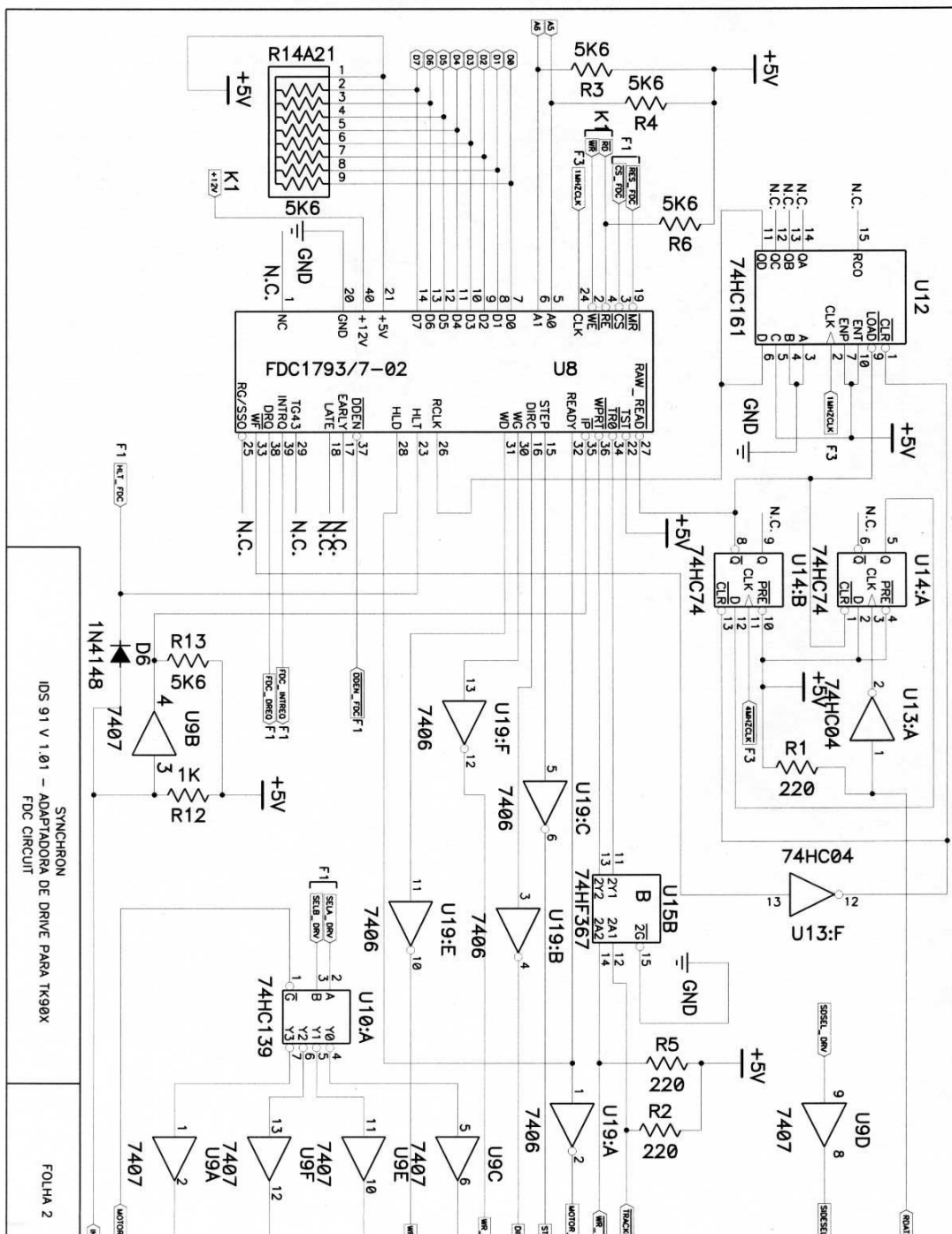
#### IMPRESSORA (VISTO DE FRENTE)

Pinos 01 a 25	TERRA
Pino 02	N.C.
Pino 04	N.C.
Pino 06	BUSY
Pino 08	N.C.
Pino 10	Dado 7
Pino 12	Dado 6
Pino 14	Dado 5
Pino 16	Dado 4
Pino 18	Dado 3
Pino 20	Dado 2
Pino 22	Dado 1
Pino 24	Dado 0
Pino 26	STROBE



## XII. ESQUEMAS:

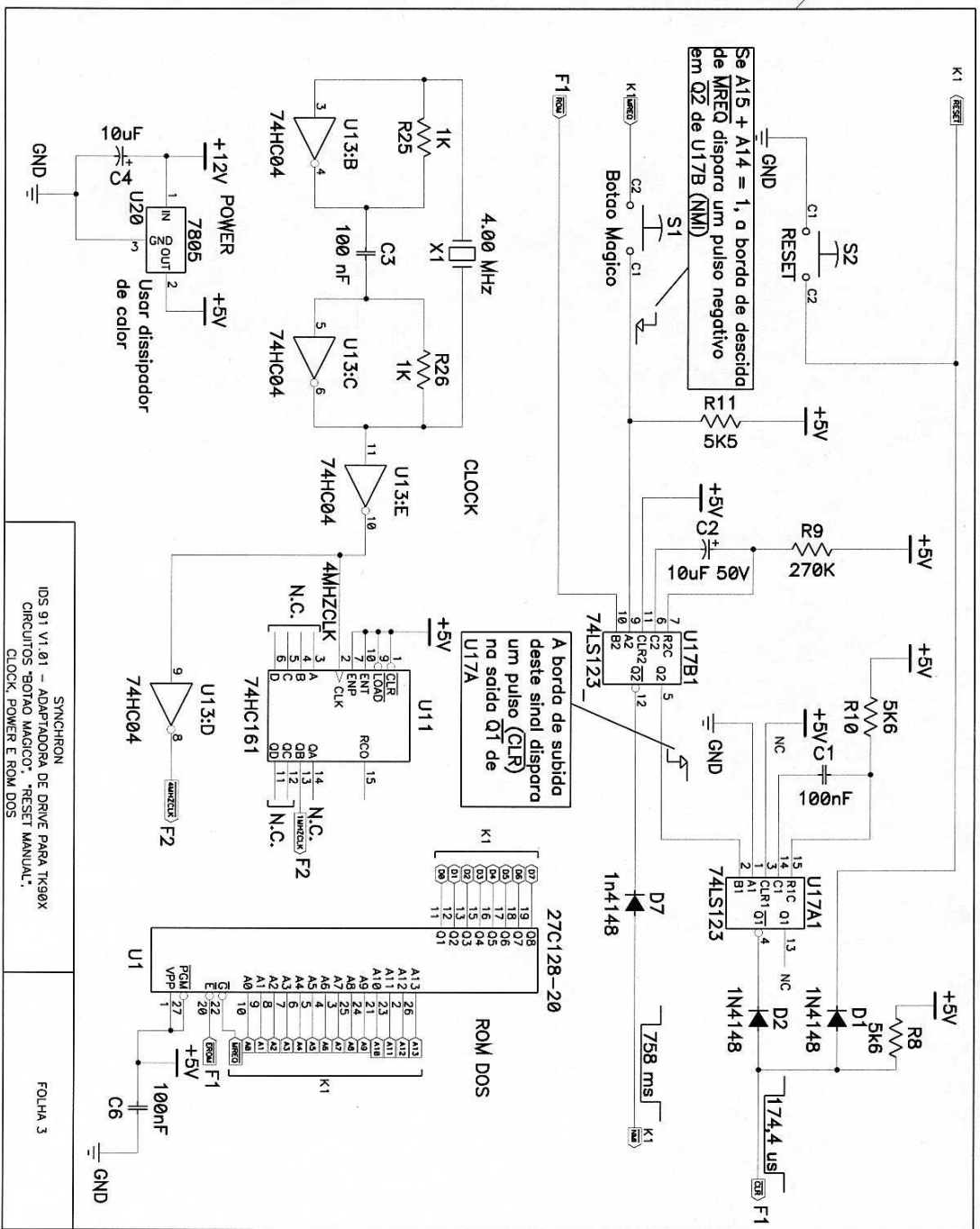


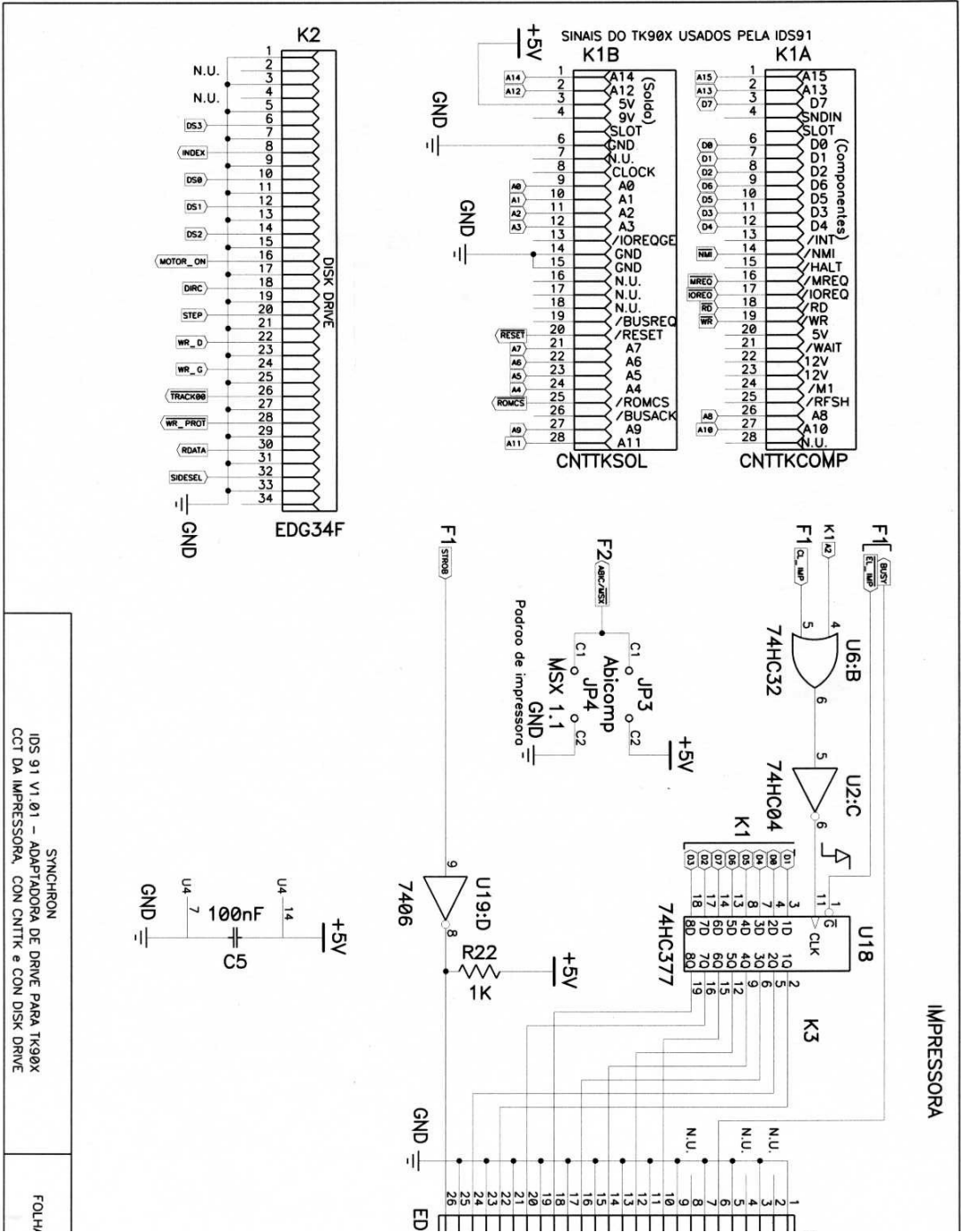


IDS 91 V 1.01 - ADAPTADORA DE DRIVE PARA TK90X  
 SYNCHRON  
 FDC CIRCUIT

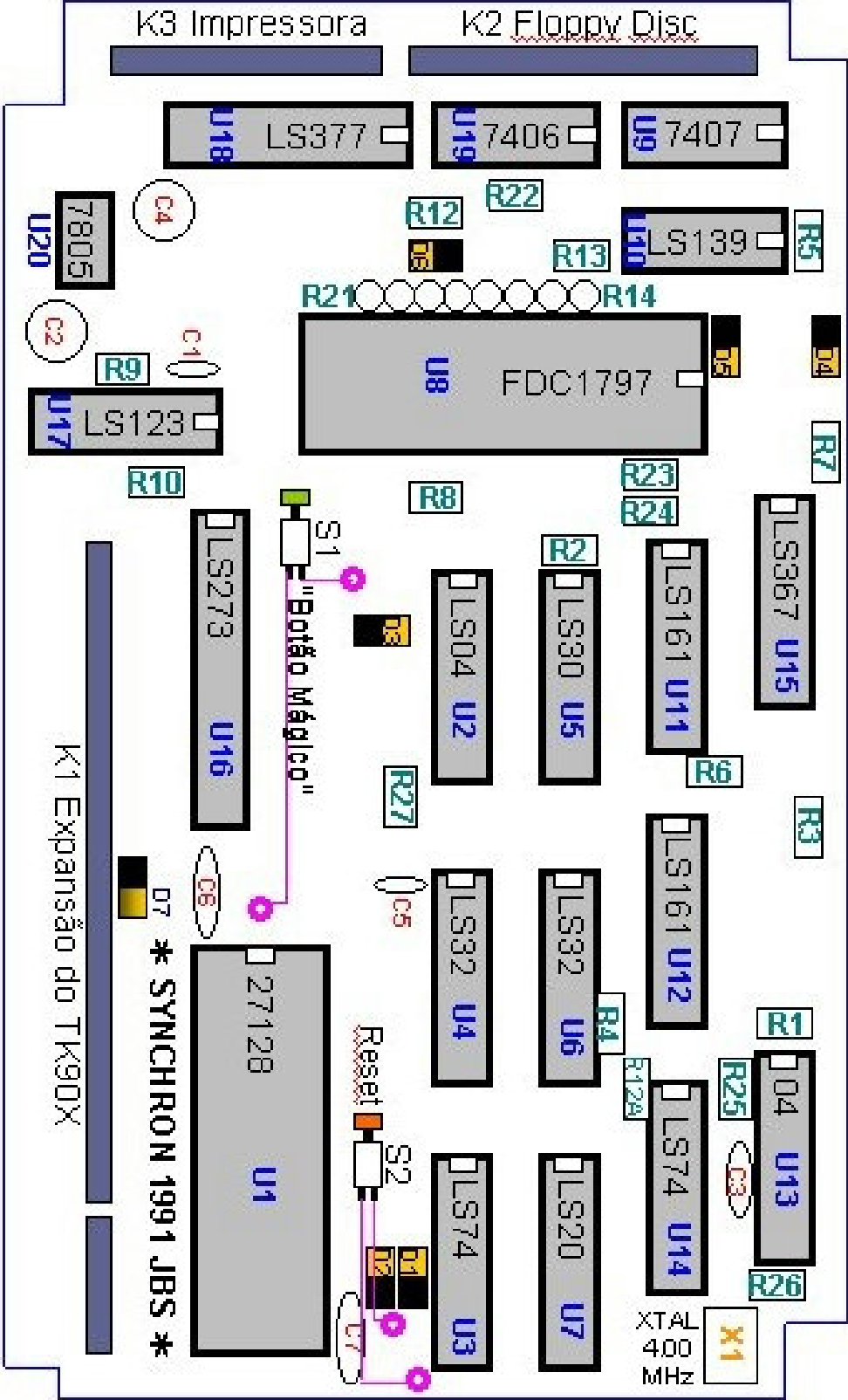
FOLHA 2







XIII. LAYOUT

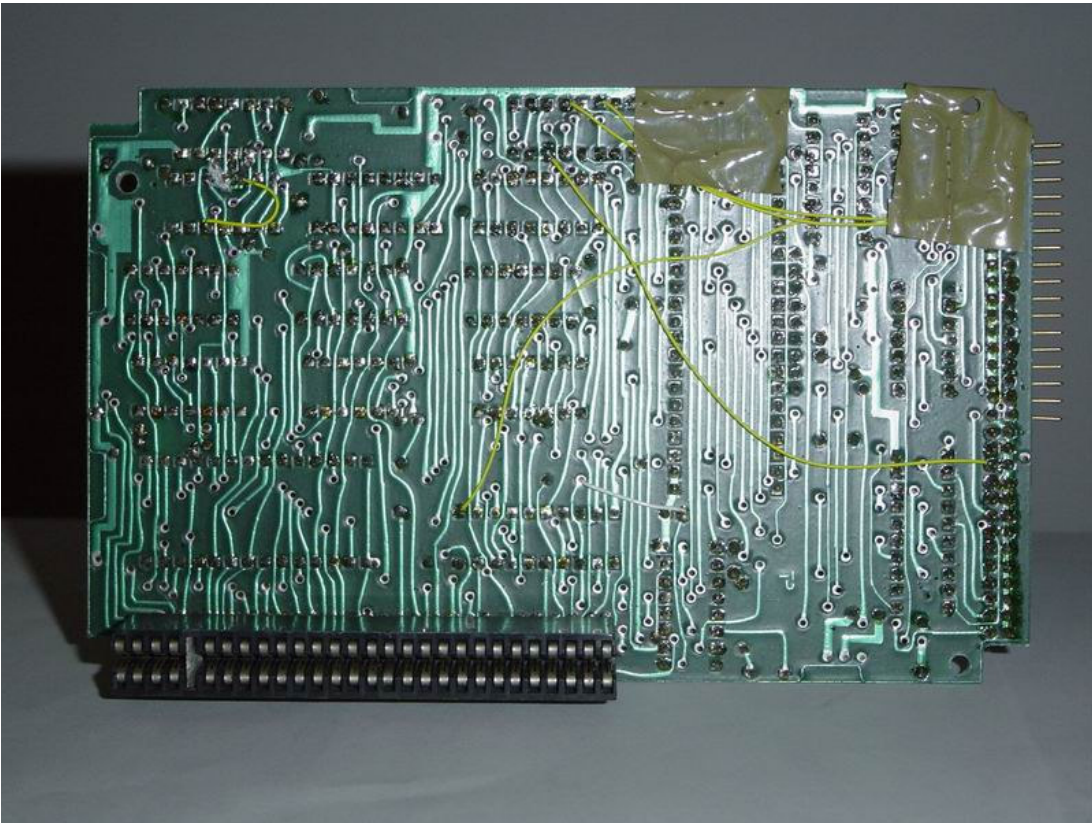
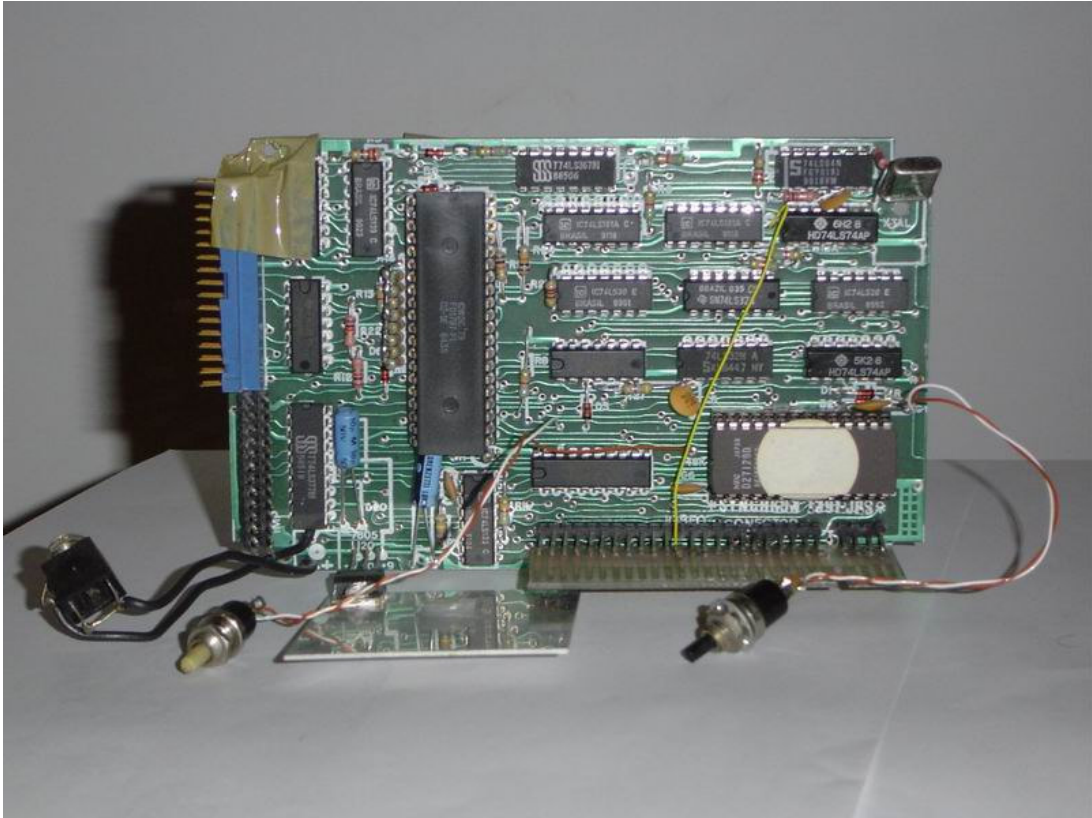


XIV. FOTOS









A SYNCHRON cabe o direito de, a qualquer tempo, modificar, sem aviso prévio, as especificações deste equipamento, é proibida a reprodução total ou parcial deste manual, sem prévia autorização da SYNCHRON.

Este manual foi copiado do seu original, foram feitas modificações a fim de corrigir erros de digitação do manual original.

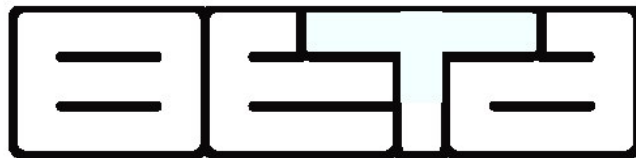
A Interface de Drive IDS-91 foi projetada e desenvolvida por:  
Jorge Braga da Silva @=> jorbs@itelefonica.com.br

#### 1ª REVISÃO

Completada / Revisada em 16 de Dezembro de 2004  
Por Clóvis Friolani @=> frio\_tk@yahoo.com.br

Agradecendo desde já o meu amigo Fábio Lalau que fez o trabalho com as imagens e o pessoal do Grupo Yahoo TK90X.

# RELATÓRIO BETA IDS2001<sup>ne</sup>



Desenvolvido e escrito por  
Jorge Braga da Silva  
jorbs@itelefonica.com.br



# I – GENERALIDADES

=====

## 1 - MAPA DE MEMÓRIA GENERALIZADO E AS MEMÓRIAS DA IDS2001ne

NÃO IMPLEMENTADA NO TK90X	DD...DD	<b>BLOCO B<sub>n</sub></b> 16 KBytes RAM	HH...HH		
	DD...DD+1-16K		HH...HH+1-16K		
	DD...DD-16K	<b>BLOCO B<sub>n-1</sub></b> 16 KBytes RAM	HH...HH		
	DD...DD+1-32K		HH...HH+1-32K		
	.	.	.		
	.	.	.		
	.	.	.		
MEMÓRIA ON-BOARD DO TK90X	73.327	<b>BLOCO 5</b> 16 KBytes RAM	11FFF		
	69.632		11000		
	69.631	<b>BLOCO 4</b> 16 KBytes RAM	10FFF		
	65.536		10000		
	65.535	<b>BLOCO 3</b> 16 KBytes RAM	FFFF		
	49.152		C000		
	49.151	<b>BLOCO 2</b> 16 KBytes RAM	BFFF		
	32.768		8000		
	32.767	<b>BLOCO 1</b> 16 KBytes RAM	7FFF		
	16.384		4000		
	16.383		3FFF		
	<b>BLOCO 0</b> 16 KBytes ROM-BASIC		16.383d	<b>MEMÓRIAS DA BETA</b>	3FFF
00000		0000	00000d	<b>BLOCO 0.1</b> 16 KBytes (ROM-DOS na BETA)	0000

**Figura 1 - Mapa de memória generalizado e as memórias da IDS2001ne.** Os quatro primeiros blocos existem fisicamente no TK90X e seus endereços têm uma correspondência biunívoca com os endereços das páginas de memória do Z80. O Bloco 0.1 é acrescentado com a inclusão da IDS2001ne e o circuito chaveia a página 0 do Z80 entre os Blocos 0 e 0.1. Os Blocos de 4 a B<sub>n</sub> representam genericamente possíveis ampliações de memória física, como por exemplo os Spectrum 128 e Spectrum +3, nos quais se implementou circuitos para que as quatro páginas do Z80 sejam chaveadas em 12 blocos de memória física.

## 2 – DEFINIÇÕES E CONVENÇÕES:

**Âmbito deste relatório:** Em princípio, as definições e terminologias usadas a seguir são aplicáveis somente aos relatórios e outros documentos elaborados pela synchron. O sentido dos termos podem estar de acordo com o uso corrente ou não, mas todo esforço foi feito para que refletissem exatamente o sentido já consagrado no meio técnico.

**Bases numéricas:** Todos os números neste relatório estão na base dez exceto quando especificado ao contrário. Ex: 123o (octal), 123d (decimal), 0101b (binário), 1234h (hexadecimal). Os números representados em decimal não terão indicador. Ex: 1234 é um decimal.

**Memória física:** Toda memória contida em circuitos integrados.

**Memória primária:** Memória física instalada na motherboard do TK90X. Os 64 K bytes do TK90X são uma memória primária.

**Bloco:** 16 K bytes contínuos e lineares de memória física. Vários blocos podem ser concatenados para formar um banco de memória física, por exemplo, no Spectrum 128K existem 12 blocos, perfazendo um total de 192 KB de memórias RAM e ROM.

**Página:** Faixa de 16 K bytes contínuos e lineares dentro da qual o microprocessador pode gerar endereços. O Z80 pode endereçar numa faixa de endereços lineares de 0 a 65.535. Diz-se, então, para efeitos desse relatório, que sua capacidade de endereçamento é de 4 páginas. Já o 8088 tem capacidade para endereçar 64 páginas.

**Paginar um bloco:** No TK90X, um modo comum do sistema operacional (SO) vencer os limites de endereçamento do seu microprocessador é fazendo paginação em um conjunto de blocos com mais de 64 KB de memória. Para isso o SO manipula um circuito eletrônico do hardware para desviar a página endereçada pelo microprocessador para um dos n blocos de modo que o processamento não sofra descontinuidade nem perdas de informações. No caso objeto desse relatório, no qual se considera a IDS 2001 conectada, temos normalmente o TK90X operando sobre cinco blocos de memórias, quatro do próprio micro e um da interface, totalizando  $5 \times 16K = 80K$  total de memória física. Portanto, um circuito eletrônico terá a incumbência de chavear ou paginar a ROM BETA no lugar da ROM BASIC, e vice-versa, conforme seja adequado a cada momento. É obvio que as memórias físicas estão presas na placa e, portanto, não irão se mover para ocupar uma o lugar da outra. Na verdade a página 0 (veja figura 1) do Z80 é que é desviada pelo circuito, a comando do SO, para ser acessada no bloco físico correto. Neste caso a ROM BETA no interior da interface.

**Banco de memórias:** Toda a memória de um sistema, por exemplo, o banco de memória do TK90X é de 64 K bytes (16 K bytes de ROM e 48 K bytes de RAM) e da IDS2001 é 16 K bytes de ROM.

**Banco de memória RAM:** Toda a RAM de um sistema.

**Banco de memória ROM:** Toda a ROM de um sistema.

**/\_ABCD:** Sinal qualquer ativo em nível lógico zero.

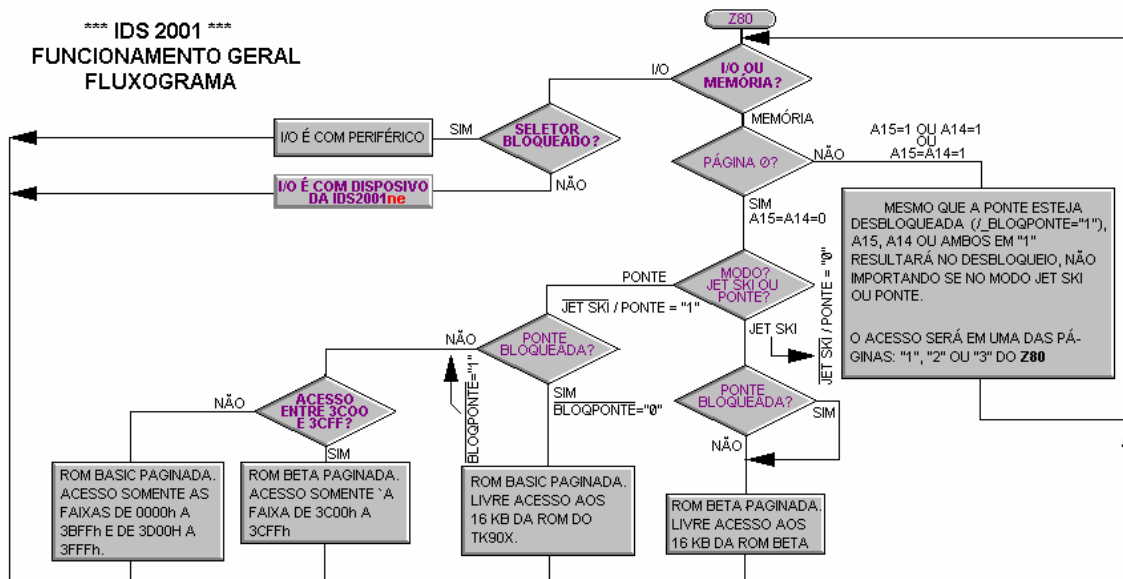
**ABCD:** Sinal qualquer ativo em nível lógico um.

**SODIS BETA:** Sistema Operacional de Disco da BETA.

**PONTE:** Faixa de endereços de memória na página 0 (no caso da IDS 2001, os endereços de 3C00h a 3CFFh) que, através de mecanismos de hardware de paginação de memórias físicas, permite ao SODIS BETA acessar rotinas na ROM BASIC e até mesmo na RAM do micro. Quando o Z80 acessa a página 0 nesta faixa, estando o SODIS BETA no MODO PONTE e com a ponte não bloqueada, é a ROM BETA que estará visível ao Z80. Rotinas aí contidas preparam o ambiente para o acesso à ROM BASIC e posterior retorno ao ponto original do programa. Se o Z80 endereçar fora dessa faixa, mesmo que seja na página 0 e respeitadas as demais condições, é a ROM BASIC ou a RAM que estará visível. Neste caso isso ocorre nos acessos entre 0000h a 3BFFh, entre 3D00h a 3FFFh e de 4000h a FFFFh.

**JET SKI:** Faixa de endereços de memória de 0000h a 3FFFh que, através de mecanismos de hardware de paginação de memórias físicas, permite ao SODIS BETA acessar rotinas em toda a extensão de sua própria memória. Neste caso está ativado o MODO JET SKI e somente ROM BETA estará visível ao Z80 na página 0. O fluxograma da figura 2 detalha melhor o que foi considerado para os modos JET SKI e PONTE.

### 3 – FLUXOGRAMA DO FUNCIONAMENTO GERAL DA IDS2001ne



**Figura 2 - Fluxograma do funcionamento geral Da IDS2001ne.** Trata apenas da lógica do circuito elétrico que compatibiliza os demais circuitos (impressora e FDC) às características do hardware do TK90X.

#### 4 – DIAGRAMA EM BLOCOS DA IDS2001ne

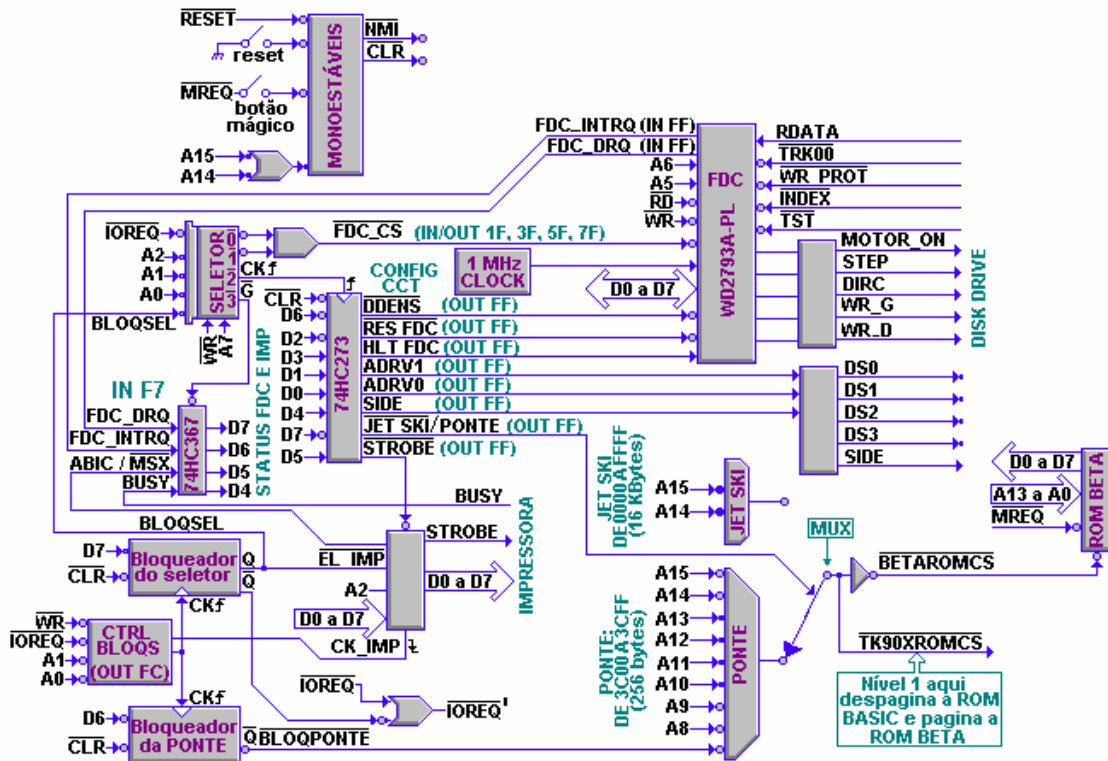


Figura 3 – Diagrama em blocos da IDS2001ne.

## II – CIRCUITOS DE APOIO DA IDS2001ne

=====

### 1 – CIRCUITO DE PAGINAÇÃO NO POWER-ON E RESET:

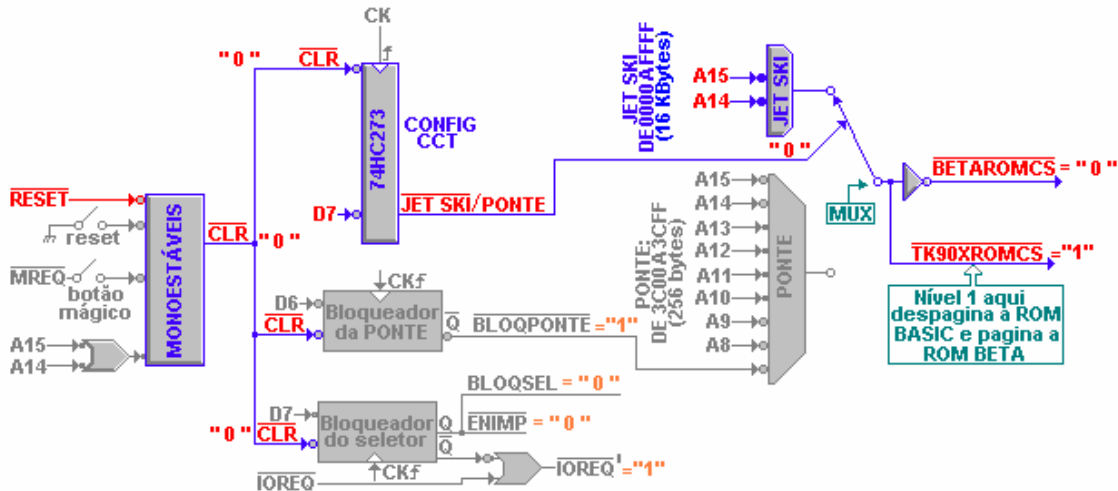


Figura 4 – Circuito de paginação ao ligar e ao resetar o micro.

#### A - NO POWER ON (AO LIGAR O TK90X):

1 - O circuito dos monoestáveis (veja Figura 4) produz o pulso negativo /\_CLR que limpa (configura para "0") os flip-flops do BLOQUEADOR DA PONTE, do BLOQUEADOR DO SELETOR e também todos os flip-flops do CONFIG CCT.

1.1 - A saída /\_BLOQPONTE do BLOQUEADOR DA PONTE vai para "1" deixando a PONTE (faixa de endereços de 3C00h a 3CFFh na ROM BETA) desbloqueada.

1.2 – A saída BLOQSEL do BLOQUEADOR DO SELETOR vai para "0", desbloqueando o SELETOR.

1.3 – Todas as saídas do 74LS273, no CONFIG CCT, vão para "0". Então /\_JET SKI / PONTE = "0" implicando que o MUX seleciona as linhas de endereço A15 e A14 para acessarem a saída /\_TK90XROMCS. Com isso todo acesso do Z80 endereçado para memória entre 0000h e 3FFFh irá colocar a ROM BETA visível (paginada, chaveada) na PÁGINA 0 do microprocessador. Este é o chamado MODO JET SKI de operação da IDS2001ne. Veja abaixo o mapa de endereços do MODO JET SKI no POWER ON.

HEX	DEC	BINÁRIO																/_JET SKI / PONTE	BLOQSEL	/_BLOQPONTE
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0			
0000	00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	•	•
3FFF	16383	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	•	•

**LEGENDA:**

- Sinal decodificado fisicamente no circuito
- Nível lógico do sinal é logicamente irrelevante

**Figura 5 - MODO JET SKI no POWER ON e RESET de hardware.**

2 – O SODIS BETA será executado e fará o papel do BASIC na configuração e inicialização do ambiente. Entre outras atividades, fará o teste de RAM disponível e inicializará as variáveis de sistema do BASIC, da BETA e da microdrive, se estiver presente.

2.1 – Como /\_IOREQ' = "1" porque o flip-flop do BLOQUEADOR DO SELETOR foi resetado, não será possível o I/O com as interfaces plugadas no conector de expansão da IDS2001ne. Quando o SODIS BETA não necessitar de I/O com a IDS2001ne, deixará o SELETOR desbloqueado, permitindo que as interfaces sejam reconhecidas pelo TK90X, como veremos mais adiante.

**B - NO RESET DE HARDWARE DO Z80:**

1 – Idem acima para o POWER ON.

**C - OBSERVAÇÕES:**

1 – Como visto acima, no POWER ON ou no caso de RESET de hardware, a IDS entra no modo JET SKI, no qual toda ROM BETA ficará visível para a PÁGINA 0 do Z80. Pela Figura 5 observa-se que para manter-se neste modo basta que /\_JET SKI / PONTE = "0", não influenciando se o SELETOR está bloqueado ou não ou se a PONTE está bloqueada ou não. O SELETOR e a PONTE estarão desbloqueados obrigatoriamente na ocorrência de POWER ON e RESET de hardware. Podemos ver, pela Figura 2, que no MODO JET SKI não é relevante o bloqueio ou não da PONTE.

2 - Mas o desbloqueio do SELETOR implica na impossibilidade de I/O com outras interfaces (desde que conectadas no barramento de expansão da IDS). Já o bloqueio do SELETOR impede que se troque o modo (no caso, de JET SKI para PONTE) porém não impede o funcionamento normal do MODO JET SKI que está ativado; impede também o I/O com a IDS (impressora e FDC) mas libera I/O com outras interfaces ligadas à IDS.

## 2 – CIRCUITO DE PAGINAÇÃO COM RAND USR 15360 E RAND USR 15363:

### A - COM RAND USR 15360:

1 – O TK90X está no modo imediato (não está rodando programas do usuário).

2 – O ambiente do SODIS BETA já foi inicializado, embora o TK90X esteja no modo BASIC (está com um dos cursores do BASIC na tela).

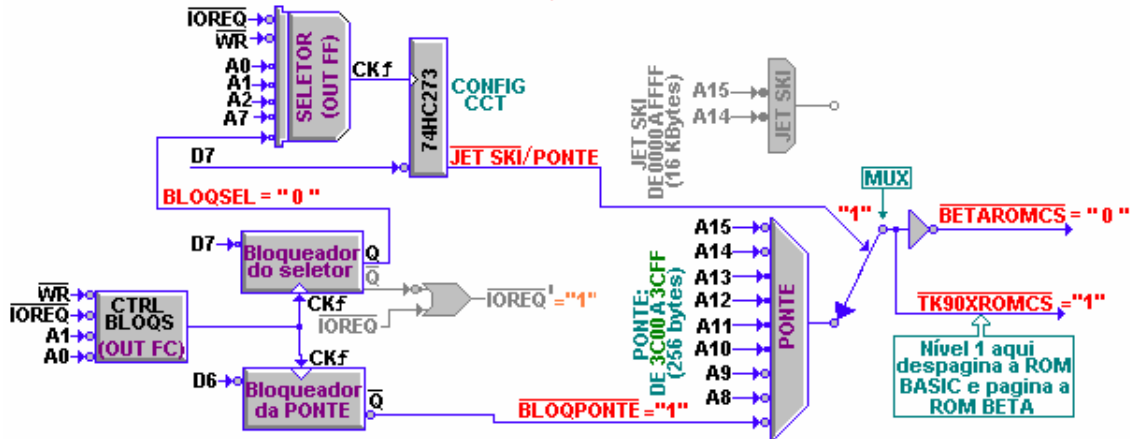


Figura 5 – Circuito de paginação com RAND USR 15360 e RAND USR 15363.

3 – Para que o RAND USR 15360 seja corretamente executado, em algum momento anterior o SODIS BETA fez o seguinte (acompanhe pela Figura 5):

3.1 – Escreveu um dado na PORTA FCh, com a seguinte configuração para os bits D7 e D6:

D7 = "0": leva BLOQSEL a "0", cujo efeito é desbloquear o SELETOR. O I/O com periféricos do TK90X, plugados no conector de expansão da IDS2001ne, fica bloqueado.

D6 = "0": leva /\_BLOQPONTE a "1", cujo efeito é desbloquear a PONTE.

3.2 – Escreveu um dado na PORTA FFh, com o bit D7 = "1", alterando em CONFIG CCT somente a saída /\_JET SKI / PONTE para nível "1", cujo efeito é fazer o MUX selecionar o MODO PONTE.

3.3 – Escreveu um dado na PORTA FCh, com a seguinte configuração para os bits D7 e D6:

D7 = "1": leva BLOQSEL a "1", cujo efeito é bloquear o SELETOR. O I/O com periféricos do TK90X, plugados no conector de expansão da IDS2001ne, fica desbloqueado.

D6 = "0": mantém /\_BLOQPONTE em "1", cujo efeito é manter desbloqueada a PONTE.

4 – Após a seqüência acima, o circuito da IDS2001ne termina no MODO PONTE com a PONTE desbloqueada. Todo acesso do Z80 à sua PÁGINA 0 terá a seguinte visibilidade de memórias:

ENDEREÇOS	VISIBILIDADE
De 0000h a 3DFFh	ROM DO TK90X
De 3C00h a 3CFFh	ROM DA IDS2001ne
De 3D00h a 3FFFh	ROM DO TK90X

5 – O usuário digita RAND USR 15360 no teclado do TK90X. Ao apertar a tecla ENTER, o fluxo do BASIC se desvia para o endereço solicitado, que está dentro da área de endereços da PONTE. Quem está visível é a ROM BETA. Rotinas na PONTE vão dar o devido tratamento ao comando do usuário. O cursor muda para o cursor do SODIS BETA, A>, B>, C> ou D>, conforme o drive default. Agora o TK90X só reconhecerá os comandos do SODIS BETA. Um deles é justamente o RETURN, que fará o micro

retornar ao ambiente nativo do BASIC. Veja abaixo o mapa de decodificação dos endereços de memória para o MODO PONTE.

HEX	DEC	BINÁRIO														/_JET SKI /PONTE	BLOQSEL	/_BLOQPONTE		
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2				A1	A0
3C00	15360	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	•	1
3CFF	15615	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	•	1

**Figura 6 - MODO PONTE – Decodificação dos endereços de memória.** Sinais que são decodificados fisicamente no circuito: A15 a A8, /\_JET SKI / PONTE e /\_BLOQPONTE.

#### B - COM RAND USR 15363:

1 – O TK90X poderá estar rodando um programa do usuário (modo programado) ou estar no modo imediato. A diferença é que RAND USR 15363 retorna o controle para o BASIC no modo imediato ou no programado. Isto é útil para permitir o acesso dos programas do usuário aos comandos do SODIS BETA.

2 – São válidas todas as considerações dos itens 3 e 4 anteriores.

3 – No modo imediato, o usuário digita RAND USR 15363 no teclado do TK90X. Ao apertar a tecla ENTER, o fluxo do BASIC se desvia para o endereço solicitado, que está dentro da área de endereços da PONTE. Quem está visível é a ROM BETA. Rotinas na PONTE vão dar o devido tratamento ao comando do usuário. Como não há um programa do usuário sendo executado, automaticamente o SODIS BETA faz um RETURN, o que trará o micro de volta ao ambiente nativo do BASIC.

4 - No modo programado, o BASIC executa RAND USR 15363, seguido de um REM para dar tempo ao chaveamento dos circuitos da IDS, o fluxo do BASIC se desvia para o endereço solicitado, que está dentro da área de endereços da PONTE. Quem está visível é a ROM BETA. Rotinas na PONTE vão dar o devido tratamento ao comando de disco que se segue. Como há um programa do usuário sendo executado, o SODIS BETA faz um RETURN ao ponto seguinte no BASIC de onde se originou a chamada ao DOS.

#### C - OBSERVAÇÕES:

1 – Se no MODO PONTE (/\_JET SKI / PONTE = "1") esta for bloqueada (/\_BLOQPONTE = "0"), todo acesso do Z80 à sua PÁGINA 0 terá visibilidade para a ROM do TK90X. Portanto, uma vez inicializado o SODIS BETA, a PONTE deverá ficar sempre desbloqueada, de modo que tanto o usuário quanto seus programas possam acessar os recursos da BETA.

2 – Quanto ao SELETOR, aqui também valem as observações 2 para o MODO JET SKI, o desbloqueio do SELETOR implica na impossibilidade de I/O com outras interfaces (desde que conectadas no barramento de expansão da IDS). Já o bloqueio do SELETOR impede que se troque o modo (no caso, de PONTE para JET SKI) porém não impede o funcionamento normal do MODO PONTE que está ativado; impede também o I/O com a IDS (impressora e FDC) mas libera I/O com outras interfaces ligadas à IDS.

3 – Na ROM BETA, os 256 bytes de 3C00h a 3CFFh que constituem a ponte entre o SODIS e o usuário e seus programas, contêm programas de controle dos circuitos da IDS para permitir o acesso de que trata o item anterior, sem que ocorram perdas de dados ou travamentos dos softwares. Já na ROMBASIC, os bytes correspondentes na mesma área de endereços são todos FFs - parte de uma área livre um pouco maior.



### 3 – CIRCUITOS DE CONTROLE E DE CONFIGURAÇÃO

#### A – BLOQUEADOR DO SELETOR E DA PONTE

A Figura 5 mostra os blocos que representam os circuitos do BLOQUEADOR do SELETOR e da PONTE.

O BLOQUEADOR utiliza a porta de I/O FCh (SAÍDA) da IDS conforme o seguinte mapa de decodificação do endereço:

HEX	DEC	BINÁRIO														/_IOREQ	/_WR	D7	D6		
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2					A1	A0
FC	252	•	•	•	•	•	•	•	•	1	1	1	1	1	1	0	0	0	0	0/1	0/1

Figura 7 - PORTA DE I/O FCh – BLOQUEADOR do SELETOR e da PONTE.

Esta é a única porta de I/O da IDS2001ne que não pode ser bloqueada. Os sinais A1, A0, /\_IOREQ, /\_WR, D7 e D6 são decodificados fisicamente no circuito, enquanto para os sinais de A7 a A2 ocorre decodificação lógica, ou seja são decodificados fisicamente em outros circuitos, obrigando seus valores lógicos na porta FCh, mesmo não participando da decodificação física, para evitar conflitos com os referidos outros circuitos.

OUT (FC) bloqueia e desbloqueia os circuitos do SELETOR e da PONTE conforme D7 e D6 sejam 0 ou 1 (veja FIGURA 7).

BITS DO DADO		STATUS RESULTANTE	
D7	D6	SELETOR	PONTE
0	0	DESBLOQUEADO	DESBLOQUEADO
0	1	DESBLOQUEADO	BLOQUEADO
1	0	BLOQUEADO	DESBLOQUEADO
1	1	BLOQUEADO	BLOQUEADO

Figura 8 – Como o BLOQ CCT bloqueia e desbloqueia os circuitos do SELETOR e PONTE.

D7 é usado no bloqueio do SELETOR e D6 no bloqueio da PONTE. Os estados de bloqueios são armazenados em dois flip-flops e permanecem atuando até que novos estado sejam estabelecidos mediante um outro OUT (FC). Como já visto, no POWER ON e no RESET de hardware, os flip-flops são limpos pelo circuito e assumem a condição inicial de desbloqueio tanto do SELETOR quanto da PONTE.

#### B – SELETOR & CONFIG CCT:

O SELETOR usa seis portas de I/O da IDS, que podem ser bloqueadas pelo circuito BLOQUEADOR. São as portas 1Fh, 3Fh, 5Fh, 7Fh, F7h e FFh.

O SELETOR, como diz o próprio nome, seleciona qual circuito será ativado para I/O na IDS. Sempre que o SELETOR for desbloqueado significa que o SODIS BETA irá acessar uma das portas de I/O da IDS, seqüentemente, e automaticamente, o sinal /\_IOREQ do bus do Z80 é bloqueado no circuito e, em seu lugar, vai para o conector de expansão da BETA o sinal /\_IOREQ' igual a "1". Para as outras interfaces do TK plugadas na IDS tudo se passa como se não houvesse requisição de I/O.

A Figura 8, abaixo, mostra o mapa dos endereços para as seis portas do SELETOR.

HEX	DEC	BINÁRIO														BLOQSEL	/_IOREQ	/_RD	/_WR		
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2					A1	A0
1F	31	•	•	•	•	•	•	•	•	0	0	0	1	1	1	1	1	0	0	1/0	0/1
3F	63	•	•	•	•	•	•	•	•	0	0	1	1	1	1	1	1	0	0	1/0	0/1
5F	95	•	•	•	•	•	•	•	•	0	1	0	1	1	1	1	1	0	0	1/0	0/1
7F	127	•	•	•	•	•	•	•	•	0	1	1	1	1	1	1	1	0	0	1/0	0/1
F7	247	•	•	•	•	•	•	•	•	1	1	1	1	1	1	1	1	0	0	0	1
FF	255	•	•	•	•	•	•	•	•	1	1	1	1	1	1	1	1	0	0	1	0

**Figura 9 – Mapa de decodificação dos endereços das portas dos dispositivos de I/O da IDS2001ne selecionáveis pelo SELETOR.** Note que /\_WR participa fisicamente da decodificação, conforme detalhamento na Figura 9. Os outros sinais que participam da decodificação física no circuito são A7, A2, A1, A0, BLOQSEL e /\_IOREQ. Já A6, A5, A4, A3 e /\_RD têm atuação lógica para evitarem conflito com outros circuitos.

Os sinais A7, A2, A1, A0, BLOQSEL, /\_IOREQ, e /\_WR são decodificados fisicamente no circuito para o acesso às portas do SELETOR. Os sinais A6, A5, A4, A3 e /\_RD participam logicamente da decodificação para evitar conflitos com outros circuitos.

A tabela a seguir mostra quais são os dispositivos de I/O selecionáveis pelo SELETOR, conforme seja feita uma leitura ou uma escrita de dado.

CÓDIGO DE SELEÇÃO		IN ou OUT ?	PORTA	DISPOSITIVO SELECIONADO
A7	/_WR			
0	0	OUT	1F	Registro de COMANDO do FDC
0	0	OUT	3F	Registro de TRILHA do FDC
0	0	OUT	5F	Registro de SETOR do FDC
0	0	OUT	7F	Registro de DADOS do FDC
0	1	IN	1F	Registro de COMANDO do FDC
0	1	IN	3F	Registro de TRILHA do FDC
0	1	IN	5F	Registro de SETOR do FDC
0	1	IN	7F	Registro de DADOS do FDC
1	0	OUT	FF	Registrador do CONFIG CCT
1	1	IN	F7	Status do FDC e da impressora

**Figura 10 – Identificação dos dispositivos de I/O da IDS2001ne selecionáveis pelo SELETOR.**

C – O CONFIG CCT:

O CONFIG CCT é um dos dispositivos de I/O controlados pelo SELETOR. Portanto, qualquer acesso a esse dispositivo depende do prévio desbloqueio do SELETOR. Trata-se de um registro de 8 bits que armazena diversas configurações para o FDC, o circuito de seleção de drives, o modo de operação para acesso as memórias da BETA e do TK90X (/\_JET SKI / PONTE) e o controle da impressora (STROBE). O registro do CONFIG CCT não pode ser lido, apenas escrito. Por isso o SODIS BETA armazena seu conteúdo em SAVE\_FF\_5D16h, uma das variáveis de sistema da BETA.

A porta de acesso do SELETOR para o registro de configurações, como vimos acima, é a FFh, somente no modo escrita. A tabela a seguir demonstra para cada valor dos bits do dado enviado ao registro qual a configuração final resultante.

SINAL	BIT DO DADO		CONFIGURAÇÃO	
/_JET SKI / PONTE	D7	= "0"	Modo JET SKI	
		= "1"	Modo PONTE	
/_DEDENS	D6	= "0"	Dupla densidade	
		= "1"	Densidade simples	
/_STROBE	D5	= "0"	Controla o envio de dados para a impressora através de um pulso negativo de duração adequada.	
		= "1"		
SIDE	D4	= "0"	Lado zero do disco	
		= "1"	Lado 1 do disco	
HLTFDC	D3	= "0"	A cabeça do drive não está posicionada	
		= "1"	A cabeça do drive está posicionada	
/_RESFDC	D2	= "0"	Reseta o FDC	
		= "1"	Não reseta o FDC	
	D1	D0	<b>SELECIONA O DRIVE:</b>	
ADRV1	D1	0	0	DRIVE 0
		0	1	DRIVE 1
ADRV0	D0	1	0	DRIVE 2
		1	1	DRIVE 3

Figura 11 – Configuração do registro de CONFIG CCT.

### III – CIRCUITOS OPERACIONAIS DA IDS2001ne

=====

#### 1 – IMPRESSORA

A porta de saída FBh é utilizada para envio de caracteres imprimíveis e de controle da impressora. O SELETOR deverá estar desbloqueado (BLOQSEL = "0"). Veja abaixo o mapa de decodificação do endereço da porta da impressora :

HEX	DEC	BINÁRIO														BLOQSEL	/_IOREQ	/_RD	/_WR		
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2					A1	A0
FB	251	•	•	•	•	•	•	•	•	1	1	1	1	1	0	1	1	0	0	1	0

Figura 12 – Mapa de decodificação do endereço da porta da impressora.

Na Figura 11 vemos que apenas os sinais A2, BLOQUESEL, /\_IOREQ, e /\_WR são fisicamente usados na decodificação do endereço da porta de saída da impressora. Os sinais /\_RD, e A7 a A4 são usados logicamente para não causar interferências em outros circuitos. O sinal BLOQSEL deverá ser posto em "0" numa etapa anterior (desbloqueio do SELETOR), caso contrário não será possível acessar a impressora.

O circuito elétrico está representado na Figura 12 de forma simplificada. Na inicialização do sistema o SODIS BETA identificou o padrão de impressora (ABICOMP ou MSX 1.1) fazendo um acesso em IN F7h. Para escrever um dado na impressora, seja para controle ou para impressão, deve-se proceder como se segue:

1 – Desbloquear o SELETOR (fazer um OUT FCh, *dado*, onde *dado* tem o bit D7 = "0").

2 – Verificar se a impressora está desocupada (fazer IN F7h e verificar se o bit D4 do *dado* é "0"). Se a impressora estiver ocupada fazer pooling nesta porta até a liberação da impressora.

3 – Escrever o *dado* no registrador da impressora (REG\_IMP) acessando a porta FBh (OUT FBh, *dado*).

4 – Escrever o *dado* na impressora da seguinte forma:

a – Ler a variável de sistema SAVE\_FF\_5D16 (área de variáveis da BETA), cujo conteúdo representa a configuração do circuito presente em CONFIG CCT.

b – Alterar o bit D5 (/\_STROB\_IMP) do *dado* lido em SAVE\_FF\_5D16 de "1" para "0".

c - Escrever o *dado* na porta FFh, o registro de configuração do circuito.

d - Aguardar o tempo de strobe previsto para a norma da porta paralela.

e – Alterar o bit D5 de "0" para "1" e escrevê-lo novamente na porta FFh. O *dado* existente no registro da impressora terá sido transmitido para a impressora.

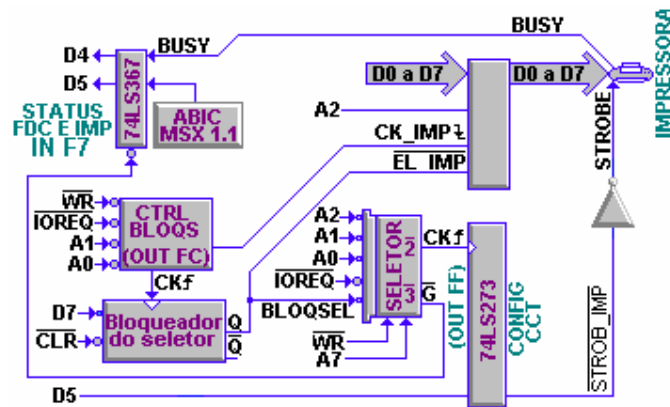


Figura 13 – Diagrama em blocos do circuito da impressora.

## 2 – FLOPPY DISK CONTROL (FDC)

As portas de saída e entrada (registros do FDC) 1Fh, 3Fh, 5Fh e 7Fh são utilizadas para o controle do FDC. O SELETOR deverá estar desbloqueado (BLOQSEL = "0"). Veja abaixo o mapa de decodificação dos endereços do FDC :

HEX	DEC	BINÁRIO																			
		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BLOQSEL	/IOREQ	/RD	/WR
1F	31	•	•	•	•	•	•	•	•	0	0	0	1	1	1	1	1	0	0	0	1
		•	•	•	•	•	•	•	•	0	0	0	1	1	1	1	1	0	0	1	0
3F	63	•	•	•	•	•	•	•	•	0	0	1	1	1	1	1	0	0	0	1	
		•	•	•	•	•	•	•	•	0	0	1	1	1	1	1	0	0	1	0	
5F	95	•	•	•	•	•	•	•	•	0	1	0	1	1	1	1	0	0	0	1	
		•	•	•	•	•	•	•	•	0	1	0	1	1	1	1	0	0	1	0	
7F	127	•	•	•	•	•	•	•	•	0	1	1	1	1	1	1	0	0	0	1	
		•	•	•	•	•	•	•	•	0	1	1	1	1	1	1	0	0	1	0	

Figura 14 – Mapa de decodificação dos endereços do FDC.

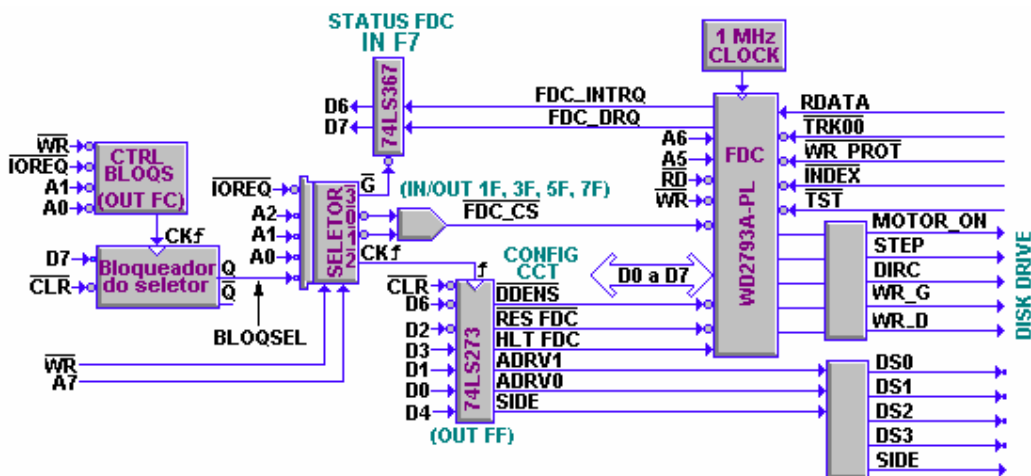


Figura 15 – Diagrama em blocos do circuito do FDC (floppy disk control). Não estão sendo mostrados alguns circuitos auxiliares do FDC.

## IV – A V I S O

=====

As informações constantes deste relatório foram obtidas mediante a análise dos circuitos e do firmware da interface original devido a não existência de manuais técnicos no Brasil para as versões da BETA aqui encontradas. Portanto use com os cuidados necessários as informações aqui contidas. O autor não teve quaisquer problemas na aplicação destas informações ao usar a interface, mas alerta que será de inteira responsabilidade do usuário, se ocorrerem danos ou falhas no seu hardware, decorrentes do uso inadequado ou não deste relatório.

O uso dos circuitos da IDS2001ne não está permitido para fins comerciais.

O autor espera receber consultas, informações, sugestões e críticas a respeito deste trabalho.

Este relatório deverá passar por atualizações conforme novas informações ficarem disponíveis.

Desenvolvido e escrito por  
Jorge Braga da Silva  
jorbs@itelefonica.com.br