



## AGRADECIMENTOS

À Companhia de Processamento de Dados do Rio Grande do Sul (PROCERGS), pela iniciativa no desenvolvimento deste trabalho e pelo apoio financeiro.

Ao Curso de Pós-Graduação, pela oportunidade.

Ao CNPQ e à UFRGS, pelo apoio financeiro.

Ao meu orientador, pelo incentivo e sugestões.

Aos colegas Reginaldo Caetano, Jaime Wagner e Maurício Tazza, pelas valiosas discussões.

Ao auxiliar de pesquisa Jonas Barcellos Moraes, pela colaboração na construção do protótipo.

A todos os colegas do curso que contribuíram de uma forma ou de outra para a realização deste trabalho.

À minha família, pela compreensão.

A Mara e Márcio.

A meus pais.

## SINOPSE

Este trabalho apresenta um sistema para teste de redes de teleprocessamento. O sistema é capaz de monitorar uma linha de teleprocessamento, armazenando de maneira seletiva o fluxo de dados em uma memória de 2048 palavras. Ele é capaz também de simular elementos da rede, sendo seu protocolo montado pelo operador. O sistema pode ainda realizar medidas de taxa de erro.

Os dados armazenados durante a monitoração podem ser visualizados num mostrador de 32 caracteres, existente no painel, ou serem despejados em uma impressora.

O sistema foi implementado com microprocessador, memória e diversas interfaces que ligam o processador central ao canal de comunicação e ao painel de comando.

## ABSTRACT

This paper presents a datacommunications network test system. This system is capable of monitoring a link, selectively storing the data stream in a 2K bytes memory. It is also capable of simulating network elements, with its protocol being defined by the operator. During a simulation test the monitoring feature is active. The system is also able to measure error rates.

This system was implemented with a microprocessor, memory and interface circuits to the communications channel and to the command front panel. The data acquired during a monitoring test can be inspected in a 32 character display in the front panel or dumped in a printer.

## SUMÁRIO

1. INTRODUÇÃO .....	1
1.1. Motivação .....	1
1.2. Definição do sistema .....	2
1.3. Apresentação do trabalho .....	6
1.4. O interface V.24 .....	7
2. CONCEITOS E DEFINIÇÕES .....	11
2.1. Conceito de evento .....	11
2.2. Conceito de janela .....	12
2.2.1. Janela desligada .....	12
2.2.2. Janela positiva .....	12
2.2.3. Janela negativa .....	13
2.3. Conceito de intervalo de tempo .....	13
2.4. Conceito de erro de recepção .....	13
2.5. Definição dos campos do mostrador .....	14
2.5.1. Campo de parâmetros .....	14
2.5.2. Campo de opções .....	14
2.5.3. Campo auxiliar .....	15
2.5.4. Campo usuário .....	15
2.6. Conceito de menu .....	17
2.7. Conceito de diretiva .....	18
2.8. Definição de ECD .....	18
2.9. Definição de ETD .....	18
2.10. Definição de memória monitora .....	19

3. TEORIA DE OPERAÇÃO .....	20
3.1. Introdução .....	20
3.2. Formato .....	22
3.2.1. Código .....	23
3.2.2. No.Bits .....	23
3.2.3. Paridad .....	23
3.2.4. Modo TX .....	23
3.2.5. Velocid .....	24
3.2.6. Canal .....	24
3.2.7. Alarme .....	24
3.2.8. Relógio .....	25
3.2.9. Exemplo de programação .....	25
3.2.10. Resumo dos parâmetros e opções de formato	28
3.3. Modo Monitor .....	30
3.3.1. Testes no modo monitor .....	32
3.3.2. Programação do modo monitor .....	33
3.3.3. Ligação no modo monitor .....	41
3.3.4. Resumo dos parâmetros e opções do modo mo nitor .....	42
3.4. Modo Simulador .....	43
3.4.1. Edição dos blocos de transmissão e recep ção .....	45
3.4.2. Sintaxe do programa de simulação .....	46
3.4.3. Localização e formato do programa de simu lação .....	48

3.4.4.	Edição e verificação do programa de simulação .....	48
3.4.5.	Exemplos de programas de simulação .....	49
3.4.6.	Programação do modo simulador .....	52
3.4.7.	Ligações no modo simulador.....	55
3.4.8.	Resumo dos parâmetros e opções do modo simulador .....	57
3.5.	Modo Teste .....	58
3.5.1.	Configurações de teste .....	58
3.5.2.	Medidas de erro .....	61
3.5.3.	Padrões de teste .....	61
3.5.4.	Programação do modo teste .....	63
3.5.5.	Resumo dos parâmetros e opções do modo teste .....	66
3.6.	Diretivas de Edição .....	67
3.6.1.	Editar,0 .....	67
3.6.2.	Editar,1 .....	68
3.6.3.	Editar,2 .....	68
3.6.4.	Editar,3 .....	68
3.7.	Diretivas de Listagem .....	68
3.7.1.	Listar,0 .....	69
3.7.2.	Listar,1 .....	69
3.7.3.	Listar,2 .....	69
3.7.4.	Listar,3 .....	69
3.7.5.	Listar,4 .....	70
3.7.6.	Listar,5 .....	70

3.7.7. Listar,6 .....	70
3.7.8. Listar,7 .....	70
3.7.9. Listar,8 .....	71
3.8. Descrição funcional das teclas especiais .....	71
3.9. Condições de erro .....	75
3.10. MENU .....	76
4. TEORIA DE FUNCIONAMENTO .....	79
4.1. Introdução .....	79
4.2. Unidade central de processamento .....	82
4.2.1. Processador central .....	84
4.2.2. Controlador de interrupções .....	85
4.2.3. Controlador de ADM .....	86
4.3. Memória .....	89
4.3.1. Memória de leitura apenas .....	89
4.3.2. Memória de leitura e escrita .....	91
4.4. Interface do teclado .....	93
4.5. Interface do mostrador .....	98
4.5.1. Varredura do mostrador .....	98
4.5.2. Controle do cursor .....	100
4.5.3. Gerador de caracteres .....	101
4.5.4. Funcionamento do interface .....	104
4.6. Interface de comunicações .....	109
4.6.1. Relógio de 1 milisegundo .....	110
4.6.2. Relógio programável .....	111
4.6.3. Canais de comunicações .....	113
4.6.4. Registrador de comando do interface .....	120

4.6.5. Método de ligação ao interface V.24 .....	122
4.6.6. Medição dos intervalos de tempo .....	123
4.7. Interface da impressora .....	125
4.8. Painel de comando .....	126
4.8.1. Teclado hexadecimal .....	128
4.8.2. Diodos emissores de luz .....	128
4.8.3. Mostrador .....	129
4.8.4. Conjunto de teclas especiais .....	129
4.9. Painel traseiro .....	130
5. ORGANIZAÇÃO DO PROGRAMA RESIDENTE .....	132
5.1. Introdução .....	132
5.2. Sistema de interrupção .....	133
5.2.1. Nível 7 .....	134
5.2.2. Nível 6 .....	134
5.2.3. Nível 5 .....	135
5.2.4. Nível 4 .....	135
5.2.5. Nível 3 .....	136
5.2.6. Nível 2 .....	136
5.2.7. Nível 1 .....	136
5.2.8. Nível 0 .....	136
5.3. Estado de controle .....	137
5.3.1. Estado "0" .....	137
5.3.2. Estado "1" .....	138
5.3.3. Estado "2" .....	138
5.3.4. Estado "3" .....	140
5.3.5. Estado "4" .....	140

5.3.6. Estado "5" .....	140
5.3.7. Estado "6" .....	140
5.3.8. Estado "7" .....	141
5.3.9. Estado "8" .....	141
5.3.10. Estado "9" .....	141
5.3.11. Estado "10" .....	142
5.4. Tabela de transição .....	142
5.4.1. Funções da tabela de transição .....	144
5.5. Estado de execução .....	147
5.5.1. Rotinas do modo monitor .....	150
5.5.2. Rotinas do modo simulador .....	155
5.5.3. Rotinas do modo teste .....	158
5.5.4. Rotinas auxiliares .....	162
6. APLICAÇÕES .....	165
6.1. Introdução .....	165
6.2. Aplicações do modo monitor .....	165
6.2.1. Exemplo 1 .....	166
6.2.2. Exemplo 2 .....	168
6.2.3. Exemplo 3 .....	170
6.2.4. Exemplo 4 .....	171
6.2.5. Exemplo 5 .....	171
6.3. Aplicações do modo simulador .....	172
6.3.1. Exemplo 1 .....	173
6.3.2. Exemplo 2 .....	174
6.4. Aplicações do modo teste .....	175
6.4.1. Exemplo 1 .....	175
6.4.2. Exemplo 2 .....	177

7. CONCLUSÃO .....	179
APÊNDICE "A" Lista de definições dos circuitos do interface V.24 .....	185
APÊNDICE "B" Código ASCII .....	192
APÊNDICE "C" Código EBCDIC .....	193
APÊNDICE "D" Código BAUDOT .....	194
APÊNDICE "E" Endereços de entrada e saída .....	195
APÊNDICE "F" Alocação dos pinos no barramento .....	196
BIBLIOGRAFIA .....	197

## LISTA DE FIGURAS

Figura 1	Componentes de uma ligação .....	3
Figura 2	Campos do mostrador .....	14
Figura 3	Exemplo de campo usuário numérico .....	16
Figura 4	Exemplo de campo usuário alfanumérico .....	17
Figura 5	Configurações de operação do STTP .....	21
Figura 6	Monitoração junto ao terminal .....	30
Figura 7	Monitoração junto ao computador .....	31
Figura 8	Ligações no modo monitor .....	42
Figura 9	Ligação direta STTP - ETD .....	55
Figura 10	Ligação via modem .....	56
Figura 11	Laço digital no modem local .....	58
Figura 12	Laço analógico no modem local .....	59
Figura 13	Laço analógico no modem remoto .....	60
Figura 14	Laço digital no modem remoto .....	60
Figura 15	Padrão 1:1 .....	61
Figura 16	Padrão 3:1 .....	62
Figura 17	Padrão 1:3 .....	62
Figura 18	Padrão 7:1 .....	62
Figura 19	Padrão 1:7 .....	62
Figura 20	Geração do padrão 511 .....	63
Figura 21	Diagrama em blocos do STTP .....	81
Figura 22	Diagrama em blocos da unidade central de pro cessamento .....	83
Figura 23	Registrador de modo do controlador de ADM ..	87
Figura 24	Memória de leitura apenas .....	90

Figura 25	Memória de leitura e escrita .....	92
Figura 26	Interface do teclado .....	96
Figura 27	Técnica de "Debounce" utilizada .....	97
Figura 28	Circuito de repetição .....	97
Figura 29	Açãoamento da matriz de pontos luminosos ..	100
Figura 30	Ocupação da célula do gerador de caracteres.	103
Figura 31	Gerador de caracteres .....	103
Figura 32	Interface mostrador .....	105
Figura 33	Carga dos registradores caracter e linha ...	106
Figura 34	Diagrama de tempos do pedido de ADM .....	107
Figura 35	Circuito carregador .....	108
Figura 36	Circuito do cursor .....	109
Figura 37	Relógio de 1 milisegundo .....	110
Figura 38	Relógio programável .....	112
Figura 39	Fluxograma de programação da USART .....	115
Figura 40	Método de ligação ao interface V.24 .....	122
Figura 41	Seletor de linha de início .....	124
Figura 42	Painel de comando .....	127
Figura 43	Painel traseiro .....	131
Figura 44	Estados principais do sistema .....	132
Figura 45	Estado de controle .....	139
Figura 46	Entrada no estado de execução .....	149
Figura 47	Iniciador do modo monitor .....	151
Figura 48	Continuador do modo monitor (parte 1 de 3) .	152
Figura 49	Continuador do modo monitor (parte 2 de 3) .	153
Figura 50	Continuador do modo monitor (parte 3 de 3) .	154
Figura 51	Iniciador do modo simulador .....	155

Figura 52	Continuador do modo simulador .....	157
Figura 53	Iniciador do modo teste .....	159
Figura 54	Continuador de transmissão do modo teste ...	160
Figura 55	Continuador de recepção do modo teste .....	161
Figura 56	Teste de linha .....	176
Figura 57	Teste do modem remoto .....	177

LISTA DE TABELAS

Tabela 1	Designação dos pinos do interface V.24 .....	9
Tabela 2	Circuitos do interface V.24 por categoria ..	10
Tabela 3	Código das teclas .....	94
Tabela 4	Código das velocidades de transmissão .....	111
Tabela 5	Posicionamento das chaves H1 e H2 .....	123
Tabela 6	Níveis de interrupção .....	134
Tabela 7	Tabela de transição .....	143

## 1 INTRODUÇÃO

Neste capítulo apresenta-se a motivação e definição de um sistema de teste de teleprocessamento e o interface sobre o qual ele atua, além da apresentação do trabalho.

### 1.1 Motivação

O conceito original do computador sendo basicamente uma calculadora sofreu uma profunda mudança nas últimas décadas. O desenvolvimento tecnológico tanto em "Hardware" como em "Software" tornaram o computador uma peça extremamente versátil e útil. O desenvolvimento de redes de teleprocessamento tornou-se inevitável na medida em que permitia aumentar a sua eficiência, compartilhando seus recursos entre muitos usuários, geograficamente dispersos.

O prognóstico de que o futuro das telecomunicações e da informática desembocarão numa via comum onde a informação será distribuída, isto é, disponível onde for necessário, já é um conceito firmado e reconhecido.

Quem quer que tenha tido a seu cargo a tarefa de gerir uma rede de teleprocessamento, mesmo das mais simples, sabe das dificuldades e da necessidade de apontar o responsável, quando surge um problema em uma determinada ligação. Num sistema altamente complexo e sofisticado como este, onde são utilizados equipamentos e serviços de diversas fontes, esta é uma tarefa bastante difícil.

A Companhia de Processamento de Dados do Rio Grande do Sul - PROCERGS - sentiu este problema. Com uma rede de teleprocessamento em franca expansão através do Estado, os problemas de manutenção passaram a representar um custo considerável. Através de consultas mantidas junto à Universidade Federal do Rio Grande do Sul - UFRGS - e do Curso de Pós-Graduação em Ciência da Computação - CPGCC -, estudou-se a possibilidade de desenvolver um sistema capaz de realizar testes em redes de teleprocessamento com a finalidade de diagnosticar prontamente a causa de problemas. Este sistema deveria ser implementado até a fase de um protótipo que seria entregue à PROCERGS que em troca financiaria parte do custo de desenvolvimento.

## 1.2 Definição do sistema

Uma ligação típica é constituída de três elementos distintos, Figura 1: (1) um EQUIPAMENTO DE TRANSMISSÃO DE DADOS (ETD), que tanto pode ser um simples terminal, como um computador, (2) um EQUIPAMENTO DE COMUNICAÇÃO DE DADOS (ECD), normalmente um modem e (3) um canal de comunicação, normalmente uma linha telefônica.



Figura 1 Componentes de uma ligação

Existem três maneiras de testar uma ligação: (1) teste de protocolo, (2) teste dos dados digitais e (3) teste do canal analógico. Para cada um destes testes, existe um conjunto de medidas específicas. O teste de protocolo constitui o nível 2 de um sistema distribuído, enquanto que os testes digitais e analógicos constituem o nível 1. Os testes de nível 1 devem ser realizados por pessoal de "Hardware", enquanto que os testes de nível 2 são normalmente realizados por pessoal de "Software".

Um equipamento que abrangesse estas três áreas de teste ainda não se tornou viável tecnológica e comercialmente. Os equipamentos de teste do nível 2 são mais caros, sofisticados e, com algumas facilidades adicionais, podem se constituir em poderosas ferramentas, podendo realizar testes do nível 1 de maneira indireta. Estes equipamentos são a última palavra em "Test-Set" para redes de teleprocessamento. É, pois, um equipamento deste tipo que se concebeu desenvolver.

O interface entre ETD e ECD é padronizado pelas

normas CCITT V.24<sup>4</sup> e EIA RS-232-C<sup>10</sup>, constituindo-se no ponto ideal de acesso à ligação. Este interface é digital e contém todos os sinais de dado e controle de interesse num teste. É através deste interface que o sistema de teste deverá atuar.

Um equipamento destinado à detecção de erros não deveria interromper o funcionamento da rede. Uma monitoração não intrusiva, capaz de armazenar de maneira seletiva os dados transacionados entre duas ou mais estações, é um requisito essencial para este sistema. Esta monitoração deve ter ainda a habilidade de reconhecer seqüências específicas de caracteres, verificar a ocorrência de erros e medir o tempo entre os sinais do interface V.24.

Alguns defeitos não podem ser detectados através de uma monitoração, por exemplo, quando um terminal não responde. Nestes casos, é necessário atuar ativamente na ligação, simulando componentes que interajam com partes específicas da rede. A capacidade de simulação é outra característica importante para um sistema de teste.

A melhor maneira de saber a fidedignidade com que um sistema transmitirá dados é fornecê-los ao sistema e examiná-los na saída. Existem muitos casos em que os componentes do sistema, modem e canal de dados, parecem estar dentro das especificações, no entanto apresentam altas taxas de erro. Por outro lado, algumas vezes um canal que parece estar altamente distorcido, transmite dados adequadamente. Devido a estas anomalias, o teste de BERT (Bit Error Rate) é necessário para prover uma indicação da qualidade geral do sistema

de transmissão. Outros testes de taxa de erro, são o CERT (Character Error Rate) e o BKER (Block Error Rate) que fornecem uma distribuição estatística de primeira ordem dos erros de bit. Esta informação indica o "Throughput" do sistema. A capacidade de realizar tais testes constitui outra característica desejável para um sistema de teste.

Após uma análise de diversos instrumentos semelhantes, existentes no mercado internacional, reuniu-se o que mais interessava de cada um e elaborou-se uma lista de características que se tornou a definição do sistema em desenvolvimento. Apresentamos abaixo esta lista:

- capacidade de monitorar caracteres de controle e texto;
- capacidade de monitorar seqüência específica de caracteres;
- capacidade de armazenar mensagens recebidas ou transmitidas;
- capacidade de simular elementos da rede;
- capacidade de gerar mensagens específicas de teste;
- programação por teclado dos protocolos desejados;
- programação fixa de alguns protocolos mais comuns;
- mostrador alfanumérico;
- códigos de operação ASCII, EBCDIC;
- seleção de paridade vertical par/ímpar/nenhuma;
- seleção de paridade horizontal longitudinal/cíclica;

- capacidade de gerar erros de paridade;
- velocidade selecionável até 9600 bps;
- operação selecionável "Full/Half-Duplex";
- modo selecionável síncrono/assíncrono;
- seleção de relógio interno/externo;
- bit de "Stop" de largura variável;
- interface de comunicações RS-232-C;
- indicação de estado do interface digital;
- pontos de teste adicionais no interface digital;
- capacidade de auto-teste;
- capacidade de gerar "Delays";
- Teste BERT/CERT;
- Medições de tempos de "Turn-around";
- alarme audível programável;
- saídas para TTY, vídeo ou K7;
- equipamento portátil;
- alimentação 110/220 V, baixo consumo.

### 1.3 Apresentação do trabalho

O sistema de teste de teleprocessamento, daqui por diante designado apenas por "STTP", foi implementado com microprocessador 8080A, possuindo 8K de memória fixa, onde está armazenado o programa residente e 4K de memória volátil, dos quais 2K são utilizados como memória de monitoração. O equipamento possui uma "tela", constituída de um mostrador de 32 caracteres alfanuméricos dispostos em linha, através da qual

pode-se visualizar os dados armazenados na sua memória durante a realização do teste. Estes dados podem ainda ser despejados em uma impressora para a obtenção de uma cópia permanente. A sua operação é comandada por meio de teclas especiais e a entrada de dados é feita através de um teclado hexa.

No capítulo 2 apresentamos definições e conceitos de expressões utilizadas neste trabalho com um significado próprio.

No capítulo 3 é apresentado o sistema do ponto de vista do usuário, suas características, seus testes, sua programação e operação.

No capítulo 4 é apresentado a implementação de "Hardware", a configuração do processador, os interfaces e seus registradores.

No capítulo 5 é apresentado o programa residente montado para controlar as atividades de todo o sistema.

No capítulo 6 são dados alguns exemplos de aplicação do STTP.

Finalmente, no capítulo 7, é feita uma auto-avaliação do trabalho, bem como um histórico do mesmo.

#### 1.4 O interface V.24

O sistema acima definido deverá atuar unicamente sobre o interface existente entre o equipamento terminal de dados (ETD) e o equipamento de comunicação de dados (ECD). Este interface é definido através do Comitê Consultivo Interna-

cional sobre Telegrafia e Telefonia (CCITT), em sua recomendação V.24<sup>4</sup> e também pela Associação das Indústrias Eletrônicas (EIA), em sua norma RS-232-C<sup>10</sup>.

No Apêndice A, apresentamos uma lista de definição dos circuitos pertencentes a este interface, doravante chamado de "interface V.24". Estes sinais estão diretamente relacionados com os pinos dos conectores J1 (ECD) e J2 (ETD) localizados no painel traseiro do STTP.

A Tabela 1 apresenta a designação dos pinos dos conectores J1 e J2 com os sinais do interface V.24.

A Tabela 2 apresenta uma lista dos circuitos do interface V.24, dividindo os sinais por categoria e indicando o seu sentido.

PINO	RS232-C	CCITT	ABREV.	DESCRIÇÃO
1	AA	101	GND	Terra de proteção
2	BA	103	TD	Dados transmitidos
3	BB	104	RD	Dados recebidos
4	CA	105	RTS	Pedido para transmitir
5	CB	106	CTS	Pronto para transmitir
6	CC	107	DSR	Modem pronto
7	AB	102	GND	Terra de sinal
8	CF	109	DCD	Detetor de portadora presente
9	-	-	-	(Reservado para teste)
10	-	-	-	(Reservado para teste)
11	-	-	-	Não designado
12	SCF	122	SDCD	Detetor portadora canal secund.
13	SCB	121	SCTS	Pronto para transmitir canal secund.
14	SBA	118	STD	Dados transmitidos canal secund.
15	DB	114	TXC	Relógio de transmissão (do ECD)
16	SBB	119	SRD	Dados recebidos canal secund.
17	DD	115	RXC	Relógio de recepção (do ECD)
18	-	-	-	Não designado
19	SCA	120	SRTS	Pedido para transmitir canal secund.
20	CD	108.2	DTR	Terminal pronto
21	CG	110	SQD	Detetor de qualidade do sinal
22	CE	125	RI	Indicador de chamada
23	CH/CI	111/112		Seletor de taxa (do ETD/ECD)
24	DA	113	TXC1	Relógio de transmissão (do ETD)
25	-	-	-	Não designado

Tabela 1 Designação dos pinos do interface V24

R	S	C	2	C	3	I	2	T	C	T	DA		CON-		TEM	
											D	P	TROL	P	D	P
											O	/	O	/	O	/
											E	E	E	E	E	E
											C	C	C	C	C	C
											D	D	D	D	D	D
AA	101	Terra de proteção								X						
AB	102	Terra de sinal								X						
BA	103	Dados transmitidos									X					
BB	104	Dados recebidos								X						
CA	105	Pedido para transmitir											X			
CB	106	Pronto para transmitir										X				
CC	107	Modem pronto										X				
CD	108	Terminal pronto											X			
CE	125	Indicador de chamada										X				
CF	109	Portadora presente										X				
CG	110	Qualidade do sinal										X				
CH	111	Seletor de taxa (ETD)											X			
CI	112	Seletor de taxa (ECD)										X				
DA	113	Relógio de trans. (ETD)														X
DB	114	Relógio de trans. (ECD)													X	
DD	115	Relógio de rec. (ECD)													X	
SBA	118	Dados trans. secund.									X					
SBB	119	Dados rec. secund.								X						
SCA	120	Pedido p/trans. secund.											X			
SCB	121	Pronto p/trans. secund.										X				
SCF	122	Portadora pres. secund.										X				

Tabela 2 Circuitos do interface V24 por categoria

Este capítulo apresenta algumas definições e conceitos de termos utilizados neste trabalho e restritos ao âmbito do mesmo.

## 2.1 Conceito de EVENTO

Um EVENTO é uma ocorrência que altera o estado de uma monitoração, iniciando ou terminando um processo de armazenamento de dados. Um EVENTO pode ser caracterizado de três maneiras:

### 1. EVENTO POR SEQUÊNCIA DE CARACTERES

Seqüências de até 8 caracteres podem ser especificadas como geradoras de um EVENTO, quando detectadas na recepção ou transmissão. Esta seqüência poderá incluir caracteres especiais como o NÃO e o QUALQUER.

### 2. EVENTO POR INTERVALO DE TEMPO

Intervalos de tempo medidos entre as transições de duas linhas do interface V.24, quando ultrapassarem um tempo especificado pelo usuário, poderão gerar um EVENTO.

### 3. EVENTO POR ERRO DE RECEPÇÃO

Ao ser assim especificado, um EVENTO é gerado sempre que for detectado um erro de recepção do tipo definido em

FORMATO.

## 2.2 Conceito de JANELA

É, muitas vezes, interessante, por razões de limitação de memória, simplicidade ou clareza do teste, armazenar somente determinadas partes do fluxo de dados. Com isto, obtemos um armazenamento seletivo e racional, o que resulta numa maior flexibilidade e inteligência do teste.

Com este objetivo, foram criados três tipos de janela, apresentados a seguir:

### 2.2.1 JANELA DESLIGADA

Com esta janela não é feito nenhum tipo de seleção no armazenamento dos dados, sendo todo fluxo armazenado na sua ordem de chegada.

### 2.2.2 JANELA POSITIVA

Esta janela define uma seqüência contínua de caracteres que será armazenada na memória. Esta seqüência é constituída de "n" caracteres ( $n = 0$  a 9999) contíguos que seguem a ocorrência do EVENTO especificado.

### 2.2.3 JANELA NEGATIVA

Esta janela define uma seqüência contínua de caracteres que não será armazenada na memória. Esta seqüência é constituída de "n" caracteres (n = 0 a 9999) contíguos que seguem a ocorrência do EVENTO especificado.

### 2.3 Conceito de INTERVALO DE TEMPO

Intervalo de tempo é a medida em milisegundos (MS), de valor máximo de 131.072 ms, e precisão de 1 ms, definida entre as transições de duas linhas do interface V.24.

### 2.4 Conceito de ERRO DE RECEPÇÃO

A ocorrência de erros no fluxo de dados recebidos pelo STTP poderá ser detectada e gerar EVENTOS.

Os métodos de verificação de erros normalmente usados em transmissão de dados são: a verificação da redundância vertical (VRV) que pode ser paridade par ou ímpar por caracter, a verificação da redundância longitudinal (VRL) e a verificação da redundância cíclica (VRC ou CRC) que é feita por blocos de caracter.

## 2.5 Definição dos campos do mostrador

Durante a programação dos testes, o mostrador apresenta campos bem definidos que mantêm informações pertinentes. A figura 2 mostra a posição destes campos no mostrador.

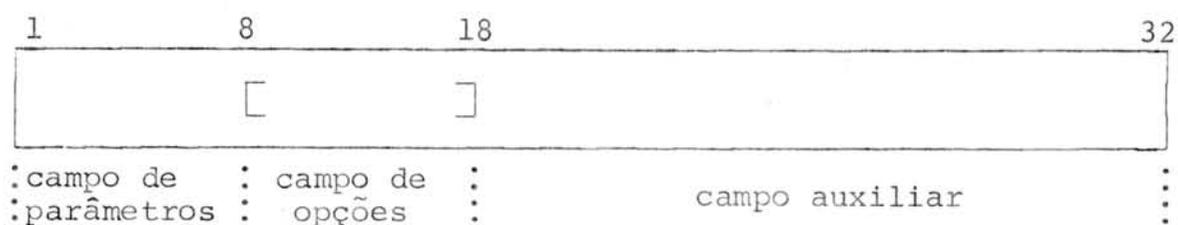


Figura 2 Campos do mostrador

### 2.5.1 Campo de parâmetros

Os parâmetros de FORMATO e dos modos TESTE, MONITOR e SIMULADOR, sempre aparecem no campo de parâmetros. Este campo é definido como as sete primeiras posições do mostrador, a partir da esquerda.

### 2.5.2 Campo de opções

As opções de um determinado parâmetro sempre aparecem no campo de opções. Este campo é definido pelas nove posições delimitadas por duas chaves [ ].

### 2.5.3 Campo auxiliar

Algumas opções possuem campo auxiliar. Este campo é todo o espaço do mostrador à direita do campo de opções. Num campo auxiliar sempre existirá um campo usuário e alguns símbolos gráficos que identificam a função deste campo usuário.

### 2.5.4 Campo usuário

Um campo usuário é delimitado por um par de parêntesis "()". Este campo sempre possui um cursor no seu interior, que indica a posição onde será colocado o caracter digitado pelo operador. Este cursor pode ser deslocado para direita ou para esquerda pelas teclas DIREITA e ESQUERDA. Sempre que o operador entrar com um caracter neste campo o cursor se desloca uma posição para a direita, nunca ultrapassando, porém, os limites deste campo.

Há dois tipos de campo usuário. Um numérico, que só aceita números de 0 a 9 e um alfanumérico que só aceita caracteres hexadecimais codificados em ASCII.

Os campos usuário, numéricos e alfanuméricos, servem para o usuário complementar as especificações de opção.

Campo usuário numérico -

São os campos usuário que só devem ser preenchidos por números. Para preencher estes campos, o operador pre-

cisa apenas digitar as teclas hexadecimais de 0 a 9, não precisando codificar os números em ASCII. Estes campos devem ter todas as suas posições preenchidas com números. Não são aceitas posições em branco. Desta maneira, sempre que o usuário quiser entrar com um número de dígitos menor que o tamanho do campo, ele deverá ajustar o número à direita e completar o campo com zeros não significativos à esquerda. Na figura 3 apresentamos um exemplo deste tipo de campo.

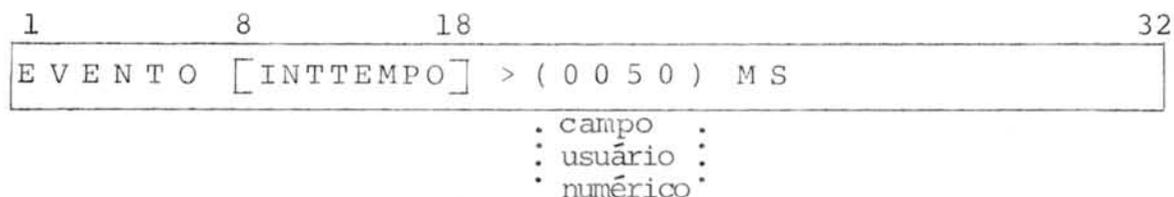


Figura 3 Exemplo de campo usuário numérico

#### Campo usuário alfanumérico -

São os campos usuário que só devem ser preenchidos por caracteres codificados em ASCII. Para preencher estes campos, o operador precisa digitar duas teclas hexadecimais para representar o código ASCII de cada caracter. Posições em branco neste campo são consideradas como caracteres branco (Hexa 20). Na figura 4, apresentamos um exemplo deste campo.

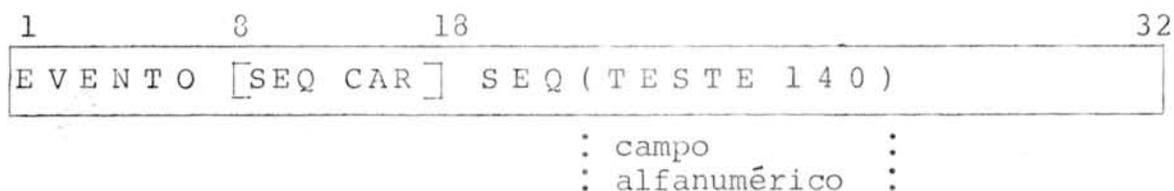


Figura 4 Exemplo de campo usuário alfanumérico

## 2.6 Conceito de MENU

MENU significa cardápio, e como tal é uma lista onde o cliente encontra as opções de pratos que lhe são oferecidos pelo estabelecimento. No nosso caso, de maneira semelhante, o operador do STTP tem diante de si uma lista de opções à sua escolha para a programação do equipamento. Esta lista pode ser consultada através das teclas PARÂMETRO e OPÇÃO.

As funções FORMATO e MODO apresentam uma lista de parâmetros que pode ser percorrida com o auxílio da tecla PARÂMETRO. Cada parâmetro, por sua vez, possui uma lista de opções que pode ser percorrida com o auxílio da tecla OPÇÃO. Programar o STTP por esta técnica significa escolher uma opção para cada parâmetro pertinente ao teste. Por exemplo, para colocar o sistema no MODO MONITOR basta apertar a tecla MODO e, a seguir, apertar a tecla OPÇÃO, até que apareça na tela

MODO [MONITOR]

A técnica de MENU tem sido utilizada de diversas maneiras em equipamentos que possuem um microprocessador para fazer o interface com o operador. Algumas vantagens desta téc

nica são a facilidade no aprendizado da programação do equipamento, o reduzido número de comandos, com a conseqüente redução do painel e a rapidez na programação.

## 2.7 Conceito de diretiva

Diretiva é um comando montado pelo operador através das teclas LISTAR ou EDITAR seguidas de um número decimal de 0 a 9. As diretivas de listagem aparecem em detalhe em 3.7, enquanto que as de edição aparecem em 3.6.

## 2.8 Definição de ECD

ECD é a abreviação para EQUIPAMENTO DE COMUNICAÇÃO DE DADOS. Este equipamento é o responsável pela transmissão e recepção de dados através de uma linha de comunicação de dados. Este equipamento normalmente é um modem (modulador demodulador).

## 2.9 Definição de ETD

ETD é a abreviação para EQUIPAMENTO TERMINAL DE DADOS. Este equipamento é utilizado para transmitir dados através de um sistema de comunicação de dados. Este equipamento poderá ser um terminal, ou um "DCP" (Processador de Comuni

cação de Dados) ou um processador de "Front-end".

## 2.10 Definição de MEMÓRIA MONITORA

MEMÓRIA MONITORA é uma área da memória de armazenamento do sistema, nos endereços 30000 a 37777 (2K bytes), onde são armazenados os dados adquiridos durante a monitoração.

Este capítulo apresenta o SISTEMA DE TESTE DE TELEPROCESSAMENTO do ponto de vista do usuário. São apresentados os modos de operação do sistema e os procedimentos de programação para a realização dos testes.

### 3.1 Introdução

O SISTEMA DE TESTE DE TELEPROCESSAMENTO é basicamente um instrumento de teste. Deve ser bem compreendido por aqueles que vão operá-lo, não só a maneira apropriada de programá-lo, mas ainda como realizar os testes disponíveis no equipamento de maneira inteligente e eficaz.

No capítulo 1 foi apresentado o ambiente, a motivação e a definição do STTP. Neste capítulo é apresentado o equipamento que foi concebido para atingir aqueles objetivos, visto sob a perspectiva do usuário. O enfoque dado aqui vai além da simples operação do STTP, apresentando também a orientação que levou a definir este ou aquele teste, esta ou aquela característica.

O STTP é extremamente versátil, podendo realizar testes sobre os mais diversos tipos de dados transmitidos serialmente através de um interface do tipo V.24. Para que ele se adapte a cada nova ligação, deve ser realizada a sua programação de FORMATO. Esta programação é apresentada em 3.2.

Uma vez programado o STTP para operar em um de-

terminado formato de dados, devemos decidir o tipo de teste que desejamos realizar. O STTP pode operar de três maneiras, ou modos: MODO MONITOR, MODO SIMULADOR e MODO TESTE. Na figura 5 vemos as configurações para cada modo de operação.

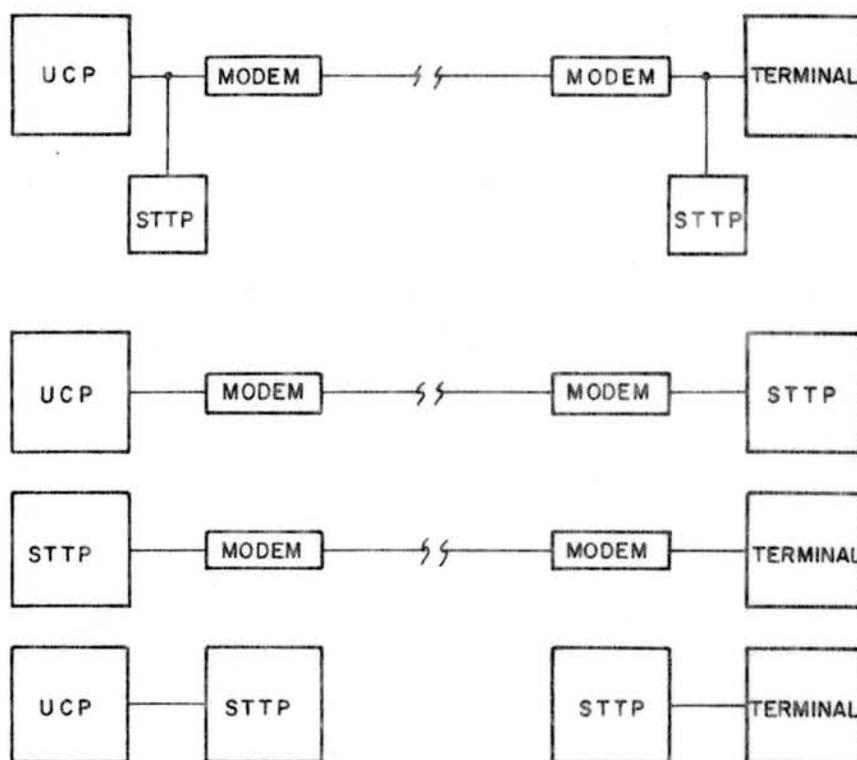


Figura 5 Configurações de operação do STTP

O MODO MONITOR é o mais comum e, possivelmente, o de maior utilidade. Este modo possibilita a monitoração, armazenamento e posterior verificação dos dados que circulam por um interface V.24, sem interferir nos mesmos.

Em alguns casos, quando a simples monitoração não é suficiente, por exemplo, quando não há atividade na linha, o MODO SIMULADOR pode ser utilizado. Neste modo, o STTP pode

simular qualquer elemento de uma ligação, com seu protocolo definido pelo próprio operador.

Por último, o STTP pode ainda realizar testes de linha através do interface V.24. Ele envia um padrão de dados pela linha, em laço e em "full-duplex", e recebe estes dados depois deles percorrerem toda a linha, comparando-os com os dados enviados e realizando a contagem de erros observados. Este teste de linha serve para testar a continuidade da mesma e dar uma idéia sobre as condições gerais da ligação, fornecendo uma medida de taxa de erros.

### 3.2           FORMATO

FORMATO é uma tecla existente no painel de comando do STTP que possibilita ao operador programar o equipamento para operar segundo as especificações da linha de comunicação à qual o sistema vai ser conectado.

Em formato são definidas as condições gerais de funcionamento do sistema. As especificações de formato influenciam as operações nos três modos de operação, TESTE, MONITOR e SIMULADOR.

A seguir, apresentamos detalhadamente os parâmetros de formato com as suas respectivas opções.

### 3.2.1 CÓDIGO

CÓDIGO - define qual o código dos dados sobre os quais o STTP vai operar. As opções de código são: EBCDIC, ASCII e BAUDOT.

### 3.2.2 NO.BITS

NÚMERO DE BITS - especifica o número de bits por carácter, dos dados sobre os quais o STTP vai operar. Este número depende principalmente do código utilizado. Por exemplo, o código BAUDOT utiliza 5 bits, enquanto o EBCDIC utiliza 8 e o ASCII 6 ou 7 bits. As opções quanto ao número de bits por carácter são: 5, 6, 7 e 8 bits.

### 3.2.3 PARIDAD

PARIDADE - especifica o tipo de verificação de erro que o STTP vai realizar. As opções de verificação de erro são: nenhuma, paridade vertical par, paridade vertical ímpar, BCC par, BCC ímpar e CRC16.

### 3.2.4 MODO TX

MODO DE TRANSMISSÃO - especifica se a transmissão

é realizada no modo síncrono ou assíncrono. Opções: síncrono e assíncrono.

### 3.2.5 VELOCID

VELOCIDADE - especifica a velocidade ou taxa de transmissão dos dados. As opções são: 9600, 4800, 2400, 1200, 600, 300, 150, 110, 75 e 50 bits por segundo.

### 3.2.6 CANAL

CANAL - define sobre qual canal será realizado o teste. Alguns modems possuem dois canais para transmissão: o canal principal, por onde são transmitidos os dados, e o canal secundário, por onde são transmitidos sinais de controle. O STTP possui a capacidade de operar seus testes sobre qualquer um dos dois canais, pela simples escolha de opção deste parâmetro. As opções são: principal e secundário.

### 3.2.7 ALARME

ALARME - define se o alarme audível estará habilitado ou não. Caso esteja habilitado, ele será acionado sempre que ocorrer um EVENTO ou for detectado um erro no MODO TESTE. Opções: ligado e desligado.

### 3.2.8 RELÓGIO

RELÓGIO - especifica a fonte do relógio utilizado para a transmissão e recepção de dados. Nas transmissões assíncronas o relógio deve ser sempre fornecido internamente. Para transmissões síncronas, o relógio poderá ser externo, caso o STTP esteja ligado a um modem que, então, fornecerá o mesmo. Opções: interno e externo.

### 3.2.9 Exemplos de programação

Apresentamos aqui um exemplo de como se programa as especificações de FORMATO.

Inicialmente, o operador deve definir uma opção para cada parâmetro apresentado acima.

#### 1) O operador define os parâmetros de FORMATO

Código: ASCII

Número de bits: 7

Paridade: ímpar

Modo de transmissão: síncrono

Velocidade: 9600

Canal: principal

Alarme: desligado

Relógio: externo

#### 2) Apertar a tecla FORMATO. Deve aparecer na tecla o primeiro

parâmetro de formato, código, com a sua opção anteriormente programada. Por exemplo:

CÓDIGO [ EBCDIC ]

- 3) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

CÓDIGO [ ASCII ]

- 4) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro número de bits com a sua opção anteriormente programada.

- 5) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

NO.BITS [ 7 ]

- 6) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro paridade com a sua opção anteriormente programada.

- 7) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

PARIDAD [ ÍMPAR ]

- 8) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro modo de transmissão com a sua opção anteriormente programada.

- 9) Se a opção da tela não for a que desejamos, apertar suces-

sivamente a tecla OPÇÃO até aparecer:

MODO TX [SÍNCRONO]

10) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro velocidade com a sua opção anteriormente programada.

11) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

VELOCID [ 9600 ]

12) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro canal com a sua opção anteriormente programada.

13) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

CANAL [PRINCIPAL]

14) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro alarme com a sua opção anteriormente programada.

15) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

ALARME [DESLICADO]

16) Apertar a tecla PARÂMETRO. Deve aparecer na tela o parâmetro relógio com a sua opção anteriormente programada.

17) Se a opção da tela não for a que desejamos, apertar sucessivamente a tecla OPÇÃO até aparecer:

sivamente a tecla OPÇÃO até aparecer:

RELÓGIO [ EXTERNO ]

- 18) Para revisar a programação de formato, basta apertar sucessivamente a tecla PARÂMETRO, verificando para cada uma a opção em que este se encontra.

Desta maneira, programamos o STTP para ser conectado a uma ligação síncrona, de 9600 bps, operando em código ASCII de 7 bits com paridade ímpar, através do canal principal.

### 3.2.10 Resumo dos parâmetros e opções de formato

CÓDIGO [ ASCII ] \*  
 [ EBCDIC ]  
 [ BAUDOT ]

NO.BITS [ 5 ]  
 [ 6 ]  
 [ 7 ] \*  
 [ 8 ]

PARIDAD [ SEM ] \*  
 [ PAR ]  
 [ ÍMPAR ]

BCC/PAR  BCC/ÍMPAR  CRC-16 MODO TX  SÍNCRONO  ASSINC  \*VELOCID  9600  4800  2400  1200  \* 600  300  150  110  75  50 CANAL  PRINCIPAL  \* SECUND. ALARME  DESLIGADO  \* LIGADO RELÓGIO  INTERNO  \* EXTERNO 

OBS.: O asterisco indica a opção inicial do parâmetro.

### 3.3 MODO MONITOR

Este modo tem por objetivo monitorar de maneira seletiva e inteligente o fluxo e dados trocados entre duas ou mais estações de teleprocessamento.

A monitoração pode ser feita junto a qualquer uma das estações. A estação junto a qual o STTP se encontrar é chamada de estação local. Na figura 6, o STTP encontra-se junto ao terminal.

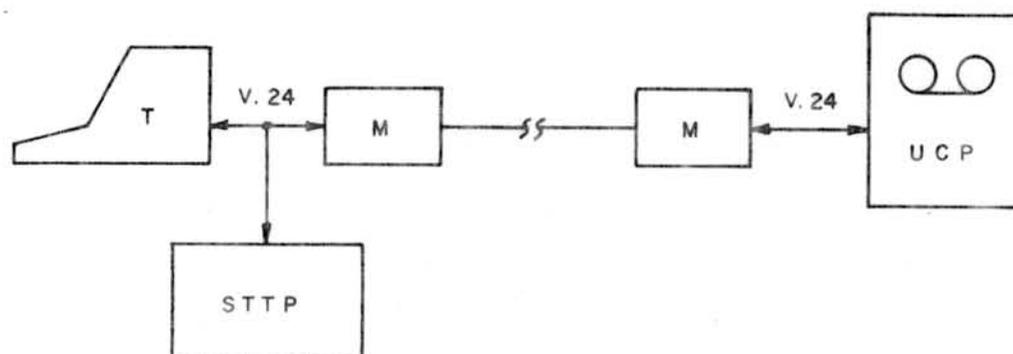


Figura 6 Monitoração junto ao terminal

Na figura 7 vemos o STTP junto ao computador. Desta maneira é possível monitorar as trocas de dados entre este e qualquer terminal ligado a ele. As mensagens trocadas com um determinado terminal são sempre identificadas por um endereço específico. Com o auxílio da janela, ver (2.2), podemos melhor visualizar as mensagens específicas de um terminal.

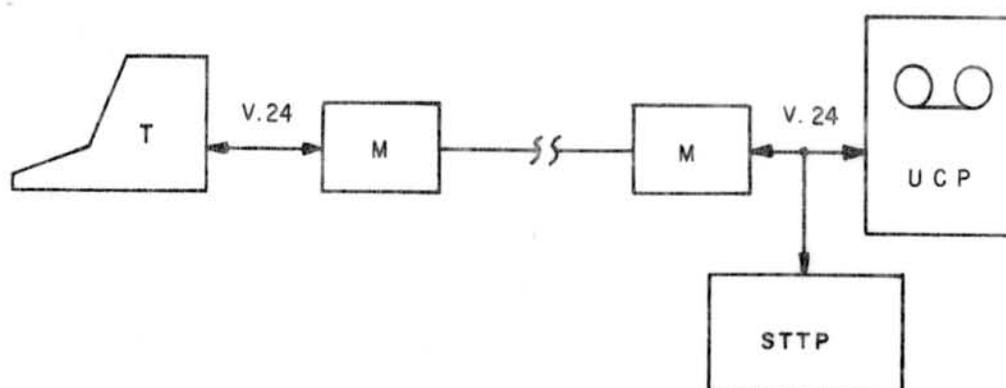


Figura 7 Monitoração junto ao computador

O STTP armazena os dados transmitidos e recebidos pela estação local, conforme seus parâmetros de teste e janela. Existe internamente uma memória estática de leitura e escrita, de 2048 palavras de 8 bits, chamada de MEMÓRIA MONITORA, onde são armazenados estes dados. Os dados recebidos pelo STTP são mostrados na tela em tempo real. Ao final do teste, o operador poderá visualizar o conteúdo da MEMÓRIA MONITORA no mostrador ou despejar o mesmo na impressora, obtendo com isto uma cópia permanente.

O operador deverá distinguir no fluxo de dados os que são transmitidos pela estação local e os que são recebidos pela mesma. Para isto, os dados transmitidos pela estação local aparecem no mostrador com o ponto decimal do "display" aceso, enquanto que os recebidos pela mesma aparecem com o ponto decimal apagado.

O operador deverá ainda reconhecer os caracteres de controle que, normalmente, não apresentam representação grá

fica. O STTP possui um gerador de caracteres especial que dá representação gráfica a estes caracteres. É importante observar aqui que estas duas facilidades, diferenciação entre caracteres recebidos e transmitidos e representação gráfica dos caracteres de controle, não estarão à disposição na cópia tirada pela impressora, já que esta apresenta um gerador de caracteres convencional. Uma maneira de contornar o problema da representação gráfica dos caracteres de controle é fazer uma cópia em código hexa, onde então, todos os caracteres terão um código representativo.

### 3.3.1 Testes no MODO MONITOR

O MODO MONITOR apresenta três tipos de teste:

#### 1. ARMAZENA E CONTA EVENTOS -

Neste teste, os caracteres transmitidos ou recebidos pela estação monitorada são armazenados segundo as especificações de JANELA e é contado o número de ocorrências do EVENTO especificado. A parada deste teste é manual, através da tecla PARAR.

Na memória ficam armazenados os últimos 2048 caracteres recebidos que podem ser examinados no mostrador ou despejados na impressora.

Ao ser interrompido o teste, é automaticamente indicado no mostrador a contagem de eventos.

## 2. ARMAZENA APÓS EVENTO -

Neste teste, não é iniciada a armazenagem de caracteres antes da ocorrência do EVENTO especificado. Após a ocorrência do mesmo, é iniciada a armazenagem dos dados transmitidos e recebidos pela estação monitorada, segundo as especificações de JANELA, até completar as 2048 posições da MEMÓRIA MONITORA, quando é encerrado o teste automaticamente.

Ao seu término é mostrada a contagem de eventos.

## 3. ARMAZENA ATÉ EVENTO -

Neste teste, os caracteres transmitidos e recebidos pela estação monitorada, são armazenados, segundo as especificações de JANELA, até a ocorrência do EVENTO especificado, após o qual ainda são armazenados mais 10 caracteres antes de encerrar.

### 3.3.2 Programação do MODO MONITOR

A programação deste modo é feita pela técnica de MENU, com o auxílio das teclas MODO, PARÂMETRO e OPÇÃO.

Apertando a tecla MODO, aparecerá no mostrador um dos três modos em que o STTP poderá se encontrar. Com a tecla OPÇÃO, levamos o sistema para o MODO MONITOR. Deverá aparecer na tela:

MODO [ MONITOR ]

Agora, apertando sucessivamente a tecla PARÂMETRO, poderemos ver os cinco parâmetros deste modo com as suas respectivas op

ções. Para cada parâmetro, devemos escolher, com o auxílio da tecla OPÇÃO, uma das diversas opções disponíveis.

A seguir apresentamos detalhadamente os parâmetros do MODO MONITOR assim como suas opções.

#### 1. EVENTO -

EVENTO - define como vai ser gerado um EVENTO, ver (2.1).

As opções de EVENTO são:

OPÇÃO 1: EVENTO [SEQ CAR] SEQ ( )

específica como EVENTO a seqüência de caracteres que deve ser digitada pelo operador no campo usuário alfanumérico de oito posições. As seqüências menores do que 8 caracteres devem ser ajustadas à esquerda e completado o restante do campo com o caracter QUALQUER.

OPÇÃO 2: EVENTO [ERRO REC]

indica que quando for detectado um erro, como definido em formato, será gerado um EVENTO.

OPÇÃO 3: EVENTO [INT TEMPO] > ( )MS

indica que, quando o intervalo de tempo medido for maior que o especificado, será gerado um EVENTO. Para a especificação deste limite máximo, o operador deverá preencher o campo usuário numérico com um número de 0 a 9.999 correspondente ao limite de tempo desejado. Para preencher este campo, basta o operador digitar as teclas numéricas de 0 a 9 do teclado hexa, ajustando à direita do campo e preenchendo com zeros não significativos à esquerda do número, caso ele tenha menos do que 4 dígitos.

## 2. TESTE -

TESTE - define qual o tipo de TESTE que será realizado neste modo.

As opções de TESTE são:

### OPÇÃO 1: TESTE [CONTA EV]

define o TESTE ARMAZENA E CONTA EVENTOS. Neste TESTE, os caracteres transmitidos ou recebidos pela estação monitorada são armazenados na MEMÓRIA MONITORA, segundo as especificações de JANELA e é contado o número de ocorrências do EVENTO especificado. A parada deste teste é manual.

### OPÇÃO 2: TESTE [EV INICIA]

define o TESTE ARMAZENA APÓS EVENTO. Neste TESTE só é iniciada a armazenagem de caracteres na memória monitora, após a ocorrência do evento especificado. Uma vez iniciada a armazenagem, esta só pára ao ser preenchida toda a memória monitora, quando é terminado o teste automaticamente.

### OPÇÃO 3: TESTE [EV FINDA]

define o TESTE ARMAZENA ATÉ EVENTO. Neste TESTE, os caracteres transmitidos e recebidos pela estação monitorada são armazenados na MEMÓRIA MONITORA, segundo as especificações de JANELA, até a ocorrência de um EVENTO, após o qual ainda são armazenados mais 10 caracteres e o teste é encerrado.

## 3. JANELA -

JANELA - especifica qual o tipo de JANELA que vai operar durante o armazenamento de dados, Ver (2.2).

As opções de JANELA SÃO:

## OPÇÃO 1: JANELA [ DESLIGADA ]

especifica que as funções de JANELA se encontram desligadas, não havendo nenhuma seleção na armazenagem de dados. Ver (2.2.1).

## OPÇÃO 2: JANELA [ EVENTO ] + ( ) CARACTS

indica que somente serão armazenados os "n" caracteres seguintes à ocorrência do evento especificado. O tamanho "n" da JANELA é definido pelo operador no campo usuário numérico de quatro posições. Este campo admite uma janela de tamanho máximo de 9.999 caracteres. Esta opção corresponde à JANELA POSITIVA definida em (2.2.2).

## OPÇÃO 3: JANELA [ N: EVENTO ] + ( ) CARACTS

é idêntica à anterior, somente que neste intervalo os "n" caracteres seguintes ao EVENTO, não são armazenados. Esta opção corresponde à JANELA NEGATIVA definida em (2.2.3).

## 4. T INIC -

INICIA INTERVALO DE TEMPO - especifica a linha do interface V.24 que vai marcar o início da contagem de tempo.

As opções de início de intervalo de tempo são:

## OPÇÃO 1: T INIC [ 0-&gt;1 ] LINHA ( )

especifica que a transição da linha indicada no campo usuário deve ser de 0 (marca, negativo) para 1 (espaço, positivo). O operador define no campo usuário numérico o número da linha do interface V.24. Este número poderá ser de 1 a 25, sendo que algumas linhas, por não apresentarem transições, não podem ser especificadas, como é o caso das linhas 1, 6, 7, 9, 10, 11, 18, 20 e 25. Caso alguma destas linhas seja especificada,

o programa indicará o seguinte erro

<<ERRO>> LINHA V.24

OPÇÃO 2: T INIC [ 1->0 ] LINHA ( )

especifica que a transição da linha especificada no campo usuário numérico deve ser de 1 (espaço, positivo) para 0 (marca, negativo).

5. T FIM -

FINDA INTERVALO DE TEMPO - especifica a linha do interface V.24 que vai marcar o fim da contagem de tempo.

As opções de fim de intervalo de tempo são:

OPÇÃO 1: T FIM [ 0->1 ] LINHA ( )

especifica que a transição da linha indicada no campo usuário deve ser de 0 (marca, negativo) para 1 (espaço, positivo). O operador define no campo usuário numérico o número da linha do interface V.24. Este número poderá ser de 1 a 25, sendo que algumas linhas, por não apresentarem transições, não podem ser especificadas, como é o caso das linhas 1, 6, 7, 9, 10, 11, 18, 20 e 25. Caso alguma destas linhas seja especificada, o programa indicará o seguinte erro

<<ERRO>> LINHA V.24

OPÇÃO 2: T FIM [ 1->0 ] LINHA ( )

especifica que a transição da linha especificada no campo usuário deve ser de 1 (espaço, positivo) para 0 (marca, negativo).

Exemplo de programação do MODO MONITOR -

Vamos supor que o teste tenha por finalidade moni

torar as respostas de um terminal AS SEQUÊNCIAS DE "POLLING" endereçadas a ele. O endereço do terminal é, por exemplo "JA".

- 1) Apertamos a tecla MODO. Deve aparecer na tela o modo em que o equipamento se encontrava previamente, por exemplo:

MODO [ SIMULA ]

- 2) Apertamos a tecla OPÇÃO, sucessivamente, até aparecer na tela:

MODO [ MONITOR ]

Com isto, colocamos o STTP no MODO MONITOR. Agora, devemos programar este modo.

- 3) Apertamos a tecla PARÂMETRO. Deve aparecer na tela o parâmetro evento com a sua respectiva opção anteriormente programada.

- 4) Apertamos a tecla OPÇÃO, sucessivamente, até aparecer na tela:

EVENTO [ SEQ CAR ] SEQ (            )

Assim, selecionamos como EVENTO uma seqüência de caracteres. Devemos, agora, entrar com a seqüência desejada no campo usuário alfanumérico existente identificado por SEQ.

- 5) Digitamos, em hexadecimal, o código ASCII dos caracteres da seqüência: "EOT,J,A,POL,ENQ".

- 6) Devemos completar o restante do campo com o caracter QUALQUER. Apertamos a tecla QUALQUER, até completar o campo

usuário. Teremos a seqüência: "EOT,J,A,POL,ENQ,X,X,X".

7) Apertamos a tecla PARÂMETRO. Aparecerá na tela o parâmetro teste com a sua respectiva opção.

8) Se a opção da tela não for a que desejamos, apertamos sucessivamente a tecla OPÇÃO até aparecer na tela:

TESTE [ EV INICIA ]

Assim, definimos que a monitoração só inicia quando for detectada a seqüência definida no passo 4 e dura até preencher a MEMÓRIA MONITORA.

9) Apertamos a tecla PARÂMETRO. Deve aparecer na tela o parâmetro janela com a sua respectiva opção.

10) Apertamos sucessivamente a tecla OPÇÃO, até aparecer na tela:

JANELA [ EVENTO ] + ( ) CARACTS

Aqui definimos uma JANELA POSITIVA. Devemos também definir o tamanho desta JANELA. Vamos supor que desejamos armazenar apenas os 10 primeiros caracteres após a identificação da seqüência de "polling".

11) Digitar no teclado hexa os números: "0,0,1,0". Na tela deverá ficar:

JANELA [ 0->1 ] +(0010)CARACTS

12) Apertamos novamente a tecla PARÂMETRO e veremos na tela

o parâmetro início de intervalo de tempo.

- 13) Apertamos sucessivamente a tecla OPÇÃO até aparecer na tela:

```
T INIC [ 0->1 ] LINHA( )
```

Definimos assim que a linha que vai marcar o início do intervalo de tempo deve ter sua transição de "0" para "1". Digamos que queremos medir o tempo de atraso do "pedido para transmitir" do terminal para a resposta "pronto para transmitir" do modem.

- 14) Digitar no teclado hexa os números "0,4". Na tela devemos ter

```
T INIC [ 0->1 ] LINHA(04)
```

Assim, a linha que marca o início do intervalo de tempo é a linha 4, "pedido para transmitir" do terminal (circuito 105).

- 15) Apertamos a tecla PARÂMETRO e deverá aparecer o parâmetro fim de intervalo de tempo.

- 16) Apertamos sucessivamente a tecla OPÇÃO até aparecer na tela:

```
T FIM [ 0->1 ] LINHA( )
```

- 17) Digitamos no teclado hexa os números "0,5" correspondentes à linha 5 do interface V.24, "pronto para transmitir do modem (circuito 106).

T FIM [ 0->1 ] LINHA(05)

18) Podemos agora verificar a montagem do teste, apertando sucessivamente a tecla PARÂMETRO e verificando se as opções e os campos usuários estão corretos.

19) Se tudo estiver em ordem, apertamos a tecla EXECUTAR.

### 3.3.3 Ligação no MODO MONITOR

Para a realização de testes no MODO MONITOR, o STTP poderá ser ligado junto ao terminal ou junto ao computador, Ver figura 8. Em ambos os casos, porém, a saída do terminal ou computador deve ser ligada através de cabo ao conector ETD(J2) localizado no painel traseiro. O modem, por sua vez, deve ser ligado ao conector ECD(J1) também localizado no painel traseiro do STTP.

Os dados transacionados entre o terminal ou computador e o modem passam internamente ao STTP sem sofrer alteração. Isto é feito para que seja retirada a informação dos sinais do interface V.24, sem introduzir distorções nos mesmos.

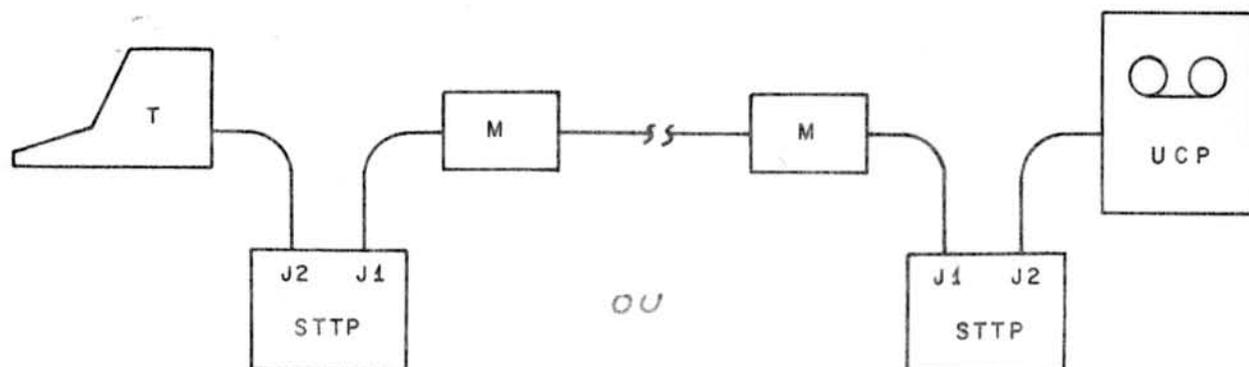


Figura 8 Ligações no MODO MONITOR

### 3.3.4 Resumo dos parâmetros e opções do MODO MONITOR

EVENTO [ SEQ CAR ] SEQ(        ) \*  
 [ ERRO REC ]  
 [ INT TEMPO ] (        ) MS

TESTE [ CONTA EV ] \*  
 [ EV INICIA ]  
 [ EV FINDA ]

JANELA [ DESLIGADA ] \*  
 [ EVENTO ] + (        ) CARACTS  
 [ N:EVENTO ] + (        ) CARACTS

T INIC [ 0->1 ] LINHA(        ) \*  
 [ 1->0 ] LINHA(        )

T FIM [ 0->1 ] LINHA( )\*  
[ 1->0 ] LINHA( )

OBS.: O asterisco indica a opção inicial do parâmetro.

### 3.4 MODO SIMULADOR

Nos casos em que não é possível realizar uma monitoração, por exemplo quando um terminal não responde, ou a linha se mantém sem atividade, torna-se necessário o uso do STTP como simulador, para atuar ativamente sobre a ligação.

O STTP possui facilidades para simular praticamente qualquer elemento da rede em qualquer protocolo não transparente que opere a nível de caracter, como é o caso do BSC (Binary Synchronous Communication) da IBM<sup>14</sup> e o poll-select da Burroughs<sup>3</sup>, ou ainda protocolos do tipo "contention".

O STTP pode atuar tanto como elemento mestre (computador), iniciador e controlador da troca de mensagens, como elemento escravo (terminal), que apenas responde aos comandos do mestre.

Nestas trocas de mensagens, o operador pode definir o tempo de espera para o sistema responder suas mensagens. Isto caracteriza a sua capacidade de gerar atrasos programados.

A filosofia de funcionamento do STTP, operando no modo simulador, se baseia na transmissão e recepção de blocos. Os blocos de transmissão são editados pelo operador em

espaços reservados de memória, onde cada um possui um nome fixo, pelo qual é referenciado.

Os blocos de recepção são identificados através de seqüências de até 8 caracteres, editados pelo operador em espaços reservados de memória onde cada seqüência possui um nome fixo, pela qual é referenciada.

Foram definidos cinco espaços para armazenar até cinco blocos de transmissão. Quatro destes espaços têm capacidade para armazenar até 28 caracteres e um tem capacidade para armazenar até 256 caracteres.

Para o armazenamento das seqüências de recepção, foram reservados quatro espaços de 8 caracteres cada um.

A identificação destes espaços é a seguinte:

- Blocos de transmissão:

A0: 28 caracteres

A1: 28 caracteres

A2: 28 caracteres

A3: 28 caracteres

A4: 256 caracteres

- Blocos de recepção:

B0: 8 caracteres

B1: 8 caracteres

B2: 8 caracteres

B3: 8 caracteres

Quando desejamos transmitir mensagens menores do que o tamanho do bloco, terminamos a mensagem com o caracter QUALQUER, que não será transmitido, mas que indicará o fim da mensagem. Quando desejamos reconhecer seqüências meno-

res do que 8 caracteres na recepção, devemos ajustar a seqüên-  
cia à esquerda e preencher com o caracter QUALQUER o resto do  
bloco. Para editar os blocos de recepção e transmissão usamos  
o teclado hexadecimal, entrando com o código ASCII correspon-  
dente a cada caracter.

Os blocos de recepção, na verdade, constituem-se  
de seqüências de até 8 caracteres que devem ser reconhecidos  
na recepção. Desta maneira, conseguimos simplificar sobrema-  
neira a montagem do programa de simulação, conforme será vis-  
to mais adiante.

#### 3.4.1 Edição dos blocos de transmissão e recepção

Para a edição dos blocos de recepção usam-se as  
seguintes diretivas:

EDITAR,1 - para a edição dos blocos de recepção,  
mostra o primeiro bloco, B0.

B0( )

ENTRAR - armazena o conteúdo do bloco mostrado  
na tela e exhibe o próximo bloco, B1.  
Após o último bloco, B3, volta a mos-  
trar o primeiro, B0.

Para a edição dos blocos de transmissão usam-se  
as seguintes diretivas:

EDITAR,2 - para a edição dos blocos de transmis-  
são A0 a A3. Mostra o primeiro bloco,  
A0.

A0 ( )

ENTRAR - armazena o conteúdo do bloco mostrado na tela e exhibe o próximo bloco. Após o último bloco A3, volta a mostrar o primeiro, A0.

EDITAR,3 - para a edição do bloco de transmissão A4 de 256 posições.

A4 (

Para a edição deste bloco não usamos a diretiva ENTRAR, visto que o seu tamanho ultrapassa o campo da tela. Cada caracter digitado na tela é automaticamente guardado na memória. Ao chegarmos ao fim da tela, os dados inseridos fazem com que os dados na tela se desloquem para a esquerda, permitindo desta maneira a edição dos 256 caracteres. No final do bloco, aparece um ")" e os dados inseridos se sobrepõem na última posição, não havendo mais deslocamento da tela para a esquerda.

Este bloco especial, A4, serve para o envio de mensagens completas, enquanto que os blocos A0, A1, A2 e A3 servem para a transmissão de mensagens de protocolo e controle.

#### 3.4.2 Sintaxe do programa de simulação

Conjunto de instruções:

## 1. A&lt;ALGARISMO DE 0 a 4&gt;

Transmite a mensagem contida no bloco de transmissão especificado pela própria instrução.

## 2. B&lt;ALGARISMO DE 0 a 3&gt;

Entra em recepção. Se durante a recepção foi recebida a mensagem especificada pela própria instrução, o programa vai para a próxima instrução, caso negativo, pula a próxima instrução.

## 3. &lt;ALGARISMO DECIMAL&gt;&lt;ALGARISMO DECIMAL&gt;

Desvia o programa para a instrução cujo número é indicado pelos dois algarismos.

## 4. BC

Entra em recepção. Se for recebido BCC (ou CRC) correto, vai para a próxima instrução, senão, pula a próxima instrução.

## Observação:

Na edição do programa de simulação não é usado o código ASCII. Observar que as diretivas deste programa utilizam apenas dígitos hexadecimais. Assim, por exemplo, para digitar a diretiva "BC", basta apertar as teclas "B" e "C" do teclado hexa; para digitar a diretiva "A0", basta apertar as teclas "A" e "0".

### 3.4.3 Localização e formato do programa de simulação

O programa de simulação é armazenado na área de programa de simulação (APS). Esta área reserva 64 posições. Cada instrução é constituída de dois caracteres, havendo área disponível para 32 instruções, numeradas de 1 a 32.

### 3.4.4 Edição e verificação do programa de simulação

A edição ou a simples verificação do programa de simulação é realizada através das diretivas:

EDITAR,0 - mostra a primeira instrução e prepara a edição.

01( )

ENTRAR - armazena o conteúdo da presente instrução na APS e mostra na tela a instrução seguinte.

02( )

Os parênteses delimitam a instrução a dois caracteres. Cada instrução, numerada e delimitada por parênteses aparece individualmente na tela. Existe um cursor no interior dos parênteses, que permite ao operador digitar e reeditar a instrução à vontade. Este cursor está sob controle das teclas DIREITA e ESQUERDA.

### 3.4.5 Exemplos de programas de simulação

#### 1. Sequência "poll" (mestre)

Conteúdo dos blocos de transmissão e recepção usados:

(A0) = EOT,AD1,AD2,POL,ENQ,X

(A1) = ACK,X

(A2) = NAK,X

(B0) = EOT,X,X,X,X,X,X,X

OBS.: "X" representa o caracter QUALQUER.

Programa:

01(A0) Envia seq. poll

02(B0) Recebeu EOT?

03(01) Sim, retorna para 01.

04(BC) Não, recebeu BCC correto?

05(08) Sim, vai para instrução 08.

06(A2) Não, envia NACK.

07(01) Retorna para 01.

08(A1) Envia ACK.

09(01) Retorna para 01.

#### 2. Sequência "SEL" (mestre)

Conteúdo dos blocos de transmissão e recepção usados:

(A0) = EOT,AD1,AD2,SEL,ENQ,X

(A4) = SOH,AD1,AD2,STX,M,E,N,S,A,G,E,M,ETX,X

(B0) = NAK,X,X,X,X,X,X,X

(B1) = ACK,X,X,X,X,X,X,X

Programa:

01(A0) Envia seq. sell

02(B0) Recebeu NACK?

03(01) Sim, retorna para 01.

04(B1) Não, recebeu ACK?

05(07) Sim, vai para instrução 07.

06(01) Não, retorna para 01.

07(A4) Envia mensagem completa.

08(B1) Recebeu ACK?

09(01) Sim, retorna para 01.

10(B0) Não, recebeu NAK?

11(01) Sim, retorna para 01.

12(01) Não, retorna para 01.

3. Seqüência "poll-sel" (escravo)

Conteúdo dos blocos de transmissão e recepção usa

dos:

(A1) = EOT,X

(A2) = ACK,X

(A3) = NAK,X

(A4) = SOH,AD1,AD2,STX,M,E,N,S,A,G,E,M,ETX,X

(B0) = EOT,AD1,AD2,POL,ENQ,X,X,X

(B1) = EOT,AD1,AD2,SEL,ENQ,X,X,X

(B2) = ACK,X,X,X,X,X,X,X

(B3) = NAK,X,X,X,X,X,X,X

Programa:

01(B0) Recebeu seqüência poll?  
02(06) Sim, vai para 06.  
03(B1) Não, recebeu seqüência sel?  
04(14) Sim, vai para 14.  
05(01) Não, retorna para 01.  
06(A4) Envia mensagem completa.  
07(B2) Recebeu ACK?  
08(12) Sim, vai para 12.  
09(B3) Não, recebeu, NAK?  
10(01) Sim, retorna para 01. ? 06  
11(01) Não, retorna para 01.  
12(A1) Envia seqüência EOT.  
13(01) Retorna para 01.  
14(A2) Envia ACK.  
15(BC) Recebeu mensagem correta?  
16(19) Sim, vai para 19.  
17(A3) Não, envia NAK.  
18(01) Retorna para 01.  
19(A2) Envia ACK.  
20(01) Retorna para 01.

### 3.4.6 Programação do MODO SIMULADOR

A programação deste modo é feita de maneira idêntica aos outros modos. Apertando a tecla MODO aparecerá no mostrador um dos três modos em que o STTP poderá se encontrar. Com a tecla OPÇÃO poderemos levar o sistema até o MODO SIMULADOR:

MODO [ SIMULA ]

Agora, apertando sucessivamente a tecla PARÂMETRO poderemos ver os parâmetros deste modo, com as suas respectivas opções. Para cada parâmetro escolhemos, com o auxílio da tecla OPÇÃO, a opção conveniente.

Todos os parâmetros do modo monitor valem para o modo simulador. Além destes, este modo apresenta os seguintes parâmetros:

#### 1. ELEMENT

ELEMENTO - define se o STTP vai operar no modo mestre (computador) ou escravo (terminal). Operando no modo mestre, ele inicia o processo de troca de mensagens. Operando no modo escravo, ele espera uma mensagem para responder. Em ambos os casos, o operador deve especificar, no campo usuário numérico, o tempo em milisegundos que o STTP deve esperar para responder uma mensagem, após reconhecê-la.

As opções de elemento são:

OPÇÃO 1: ELEMENT [ ESCRAVO ] R APS( )MS

OPÇÃO 2: ELEMENT [ MESTRE ] R APS( )MS

O tempo máximo de atraso é 9999 ms.

## 2. SIMULA

SIMULA - define se o STTP vai simular um ECD (modem) ou um ETD (terminal ou UCP). Ao simular um ECD o STTP simula não só o modem como o equipamento que estiver no outro lado da linha. Ao simular um ETD, o STTP deve ser conectado a um modem. Não é possível fazer a simulação de um ETD e conectá-lo diretamente a outro ETD, a menos que haja entre ambos um par de modems ou um eliminador de modems.

Os opções de simula são:

OPÇÃO 1: SIMULA [ ECD ] RTS/CTS ( )MS

O operador deve especificar, no campo usuário numérico de três posições, o tempo em milisegundos do atraso RTS/CTS (pedido para transmitir para pronto para transmitir) que se deseja simular.

OPÇÃO 2: SIMULA [ ETD ]

Nesta opção é simulado o terminal ou computador, devendo o STTP ser ligado a um modem.

Exemplo de programação do MODO SIMULADOR -

Vamos supor que desejamos testar um terminal. Para tanto, devemos utilizar o STTP simulando um modem e um computador. Durante a simulação, também é realizada a monitoração. Assim, não vamos nos preocupar, neste exemplo, com a montagem dos parâmetros de monitoração, pois o mesmo já foi visto na seção 3.3.

1) Apertar a tecla MODO.

2) Apertar sucessivamente a tecla opção até aparecer na tela:

MODO [ SIMULA ]

Assim, colocamos o STTP no MODO SIMULADOR.

- 3) Apertamos a tecla PARÂMETRO, sucessivamente, até aparecer o parâmetro simula.

- 4) Apertamos a tecla OPÇÃO sucessivamente até aparecer:

SIMULA [ ECD ] RTS/CTS( )MS

para simular um modem.

- 5) Digitar no teclado hexa os números: "0,1,0,0". Deste modo estabelecemos que o atraso RTS para CTS é de 100 ms. Na tela deverá ficar:

SIMULA [ ECD ] RTS/CTS(0100)MS

- 6) Apertar a tecla PARÂMETRO. Deverá aparecer o parâmetro elemento.

- 7) Apertar sucessivamente a tecla OPÇÃO até aparecer:

ELEMEN [ MESTRE ] R APS( )MS

- 8) Digitar no teclado hexa o tempo em milisegundos que se deseja esperar entre uma mensagem e outra. Por exemplo, digitar os números: "0,1,5,0". Na tela deveremos ter:

ELEMEN [ MESTRE ] R APS(0150)MS

- 9) Pronto, podemos agora verificar se os parâmetros estão corretamente programados, apertando a tecla PARÂMETRO sucessi

vamente.

- 10) Estando tudo correto e estando também editados o programa de simulação e as respectivas mensagens de transmissão e recepção, podemos apertar a tecla EXECUTAR para iniciar o teste.

#### 3.4.7 Ligações no MODO SIMULADOR

Operando no modo simulador, o usuário poderá estar diante de duas situações: Ligar o STTP a um ETD local, através de uma ligação direta, ou ligar o STTP a um ETD remoto através de uma ligação via modem. No primeiro caso, diz-se que o STTP está simulando um eco (modem). Para este tipo de ligação, o equipamento deve ser ligado ao conector ETD(J2) do STTP, conforme figura 9.



Figura 9 Ligação direta STTP-ETD

No segundo caso, para ligação do STTP a um equipamento remoto através de modem, devemos ligar a saída ECD(J1) do STTP ao modem.

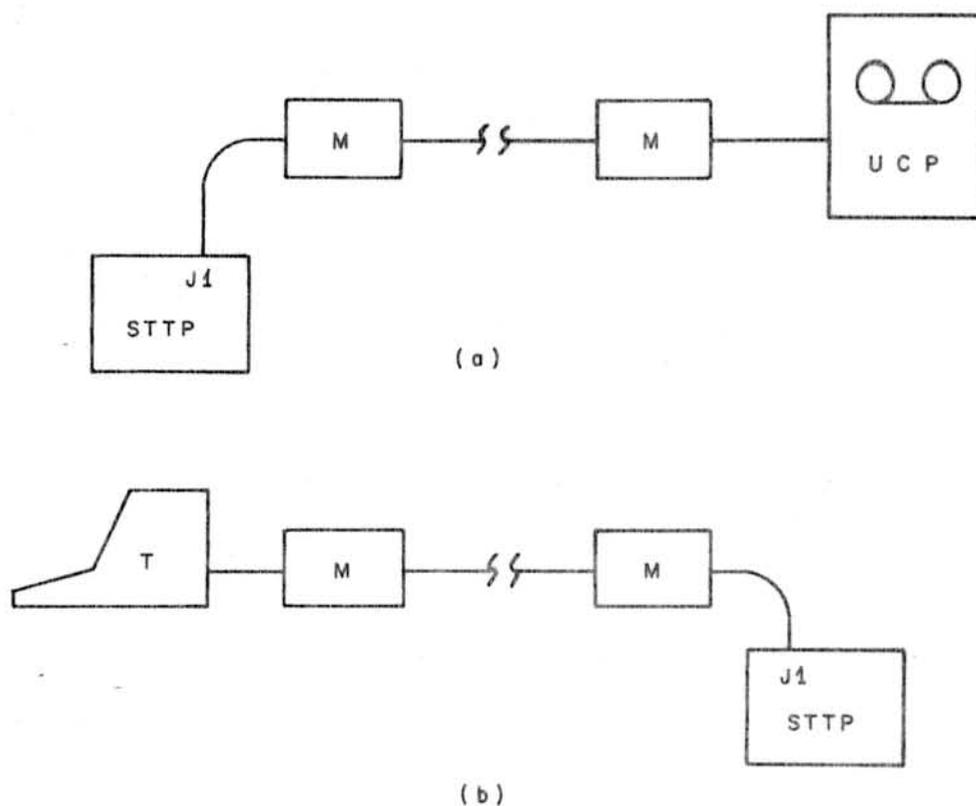


Figura 10 Ligação via modem:

(A) simulando terminal (escravo)

(B) simulando computador (mestre)

Observar que as saídas do STTP devem ser ligadas aos equipamentos de mesmo nome. Assim, a saída J1(ECD) deve ser somente ligada a um equipamento de comunicação de dados (modem), enquanto que a saída J2(ETD) deve ser somente ligada a um equipamento de transmissão de dados (terminal ou computador).

## 3.4.8 Resumo dos parâmetros e opções do MODO SIMULADOR

EVENTO  SEQ CAR  SEQ(        ) \*  
 ERRO REC   
 INT TEMP  > (        ) MS

TESTE  CONTA EV  \*  
 EV INICIA   
 EV FINDA

JANELA  DESLIGADA  \*  
 EVENTO  + (        ) CARACTS  
 N:EVENTO  + (        ) CARACTS

T INIC  0->1  LINHA(        ) \*  
 1->0  LINHA(        )

T FIN  0->1  LINHA(        ) \*  
 1->0  LINHA(        )

SIMULA  ETD   
 ECD  RTS/CTS(        ) MS \*

ELEMENT  MESTRE  R APS(        ) MS \*  
 ESCRAVO  R APS(        ) MS

OBS.: O asterisco indica a opção inicial do parâmetro.

### 3.5 MODO TESTE

Este modo tem por objetivo realizar medidas de taxa de erro no enlace analógico compreendido pelo canal e os modems.

#### 3.5.1 Configurações de teste

Este teste poderá ser configurado de quatro maneiras; segundo a ordem de apresentação.

Na configuração da figura 11, com o modem local em laço digital, o teste tem a função de verificar se o STTP está operando corretamente. Esta é uma espécie de auto-teste do equipamento. Nesta configuração, os dados transmitidos pelo STTP retornam ao mesmo sem entrar pelos circuitos do modem. Desta maneira, não devem ser detectados erros. Caso isto venha a ocorrer, indicará que o STTP está defeituoso e os próximos testes não terão mais validade.

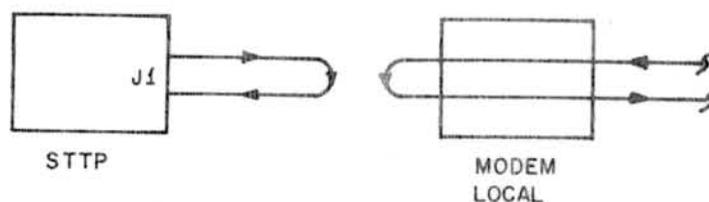


Figura 11 Laço digital no modem local

Na configuração da figura 12, com o modem local em laço analógico, o teste tem a função de verificar se os circuitos do modem local estão funcionando corretamente. Os dados transmitidos pelo STTP entram pelos circuitos de transmissão do modem e retornam pelos circuitos de recepção sem passarem pela linha física. Nesta configuração, não deverão ser contados erros. Caso isto venha a ocorrer, será uma indicação que o modem local se encontra defeituoso.

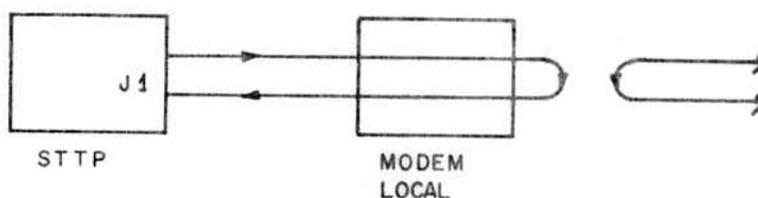


Figura 12 Laço analógico no modem local

Na configuração da figura 13, com o modem remoto em laço analógico, o teste tem a função de verificar se a linha física não está interrompida ou degradada. Caso este teste não apresente erros, teremos uma indicação segura que o modem local e a linha física estão em perfeitas condições. Caso apresente altas taxas de erros, poderemos concluir que a linha física está interrompida ou degradada.

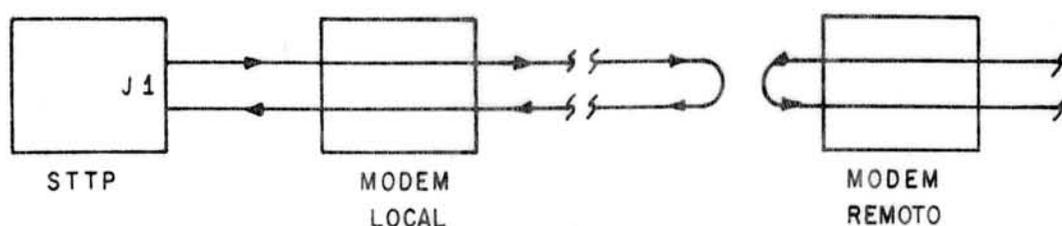


Figura 13 Laço analógico no modem remoto

Na configuração da figura 14, com o modem remoto em laço digital, o teste tem a função de verificar o funcionamento global da ligação. Os dados transmitidos pelo STTP passam pelos circuitos de transmissão e recepção de ambos os modems e pela linha física. As taxas de erro verificadas nesta configuração, indicam as condições da ligação. Segundo as Normas Telebrás (24), na configuração da figura 12, a taxa de erros de bit deve ser sempre inferior a  $1 \times 10E6$ , ou seja, menos de 1 errado em um milhão de bits transmitidos.

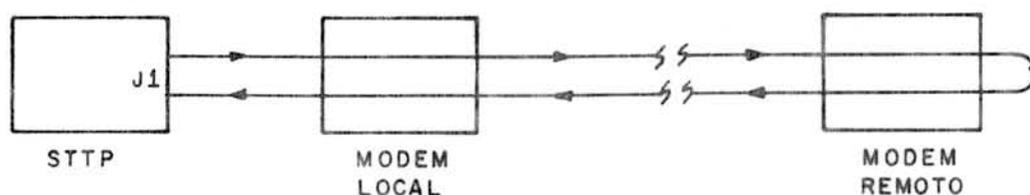


Figura 14 Laço digital no modem remoto

### 3.5.2 Medidas de erro

O modo teste realiza medições de taxas de erro de bits, caracteres ou blocos. Estas medições poderão ser feitas de três maneiras:

1. número de bits (ou caracteres ou blocos) errados em mil bits (ou caracteres ou blocos) transmitidos;
2. número de bits (ou caracteres ou blocos) errados em um milhão de bits (ou caracteres ou blocos) transmitidos;
3. contagem contínua de bits (ou caracteres ou blocos) errados.

### 3.5.3 - Padrões de teste

Os padrões enviados para a medição da taxa de erro são aqueles recomendados pela CCITT V.52(5): espaço permanente, marca permanente, 1:1, 3:1, 1:3, 7:1, 1:7 e o padrão 511.

O padrão 1:1 é uma seqüência de 0's e 1's alternados, como podemos ver na figura 15.



Figura 15 Padrão 1:1

O padrão 3:1 é a seqüência de três 1's e um 0, como podemos ver na figura 16.



Figura 16 Padrão 3:1

O padrão 1:3 é a seqüência formada por 1 e três 0's, como podemos ver na figura 17.



Figura 17 Padrão 1:3

O padrão 7:1 é formado por sete 1's e um 0, como pode-se ver na figura 18.



Figura 18 Padrão 7:1

O padrão 1:7 é a seqüência formada por um 1 e sete 0's, como podemos ver na figura 19.



Figura 19 Padrão 1:7

O padrão pseudo-aleatório 511 pode ser gerado por um registrador de deslocamento de nove estágios, cujos estágios 5 e 9 são somados em módulo dois e o resultado realimenta a entrada do registrador, ver figura 20. O somador módulo dois produz saída 1 quando as duas entradas forem diferentes e saída 0 quando as duas entradas forem iguais. No início da geração do padrão, todos os estágios do registrador de deslocamento estão em 1.

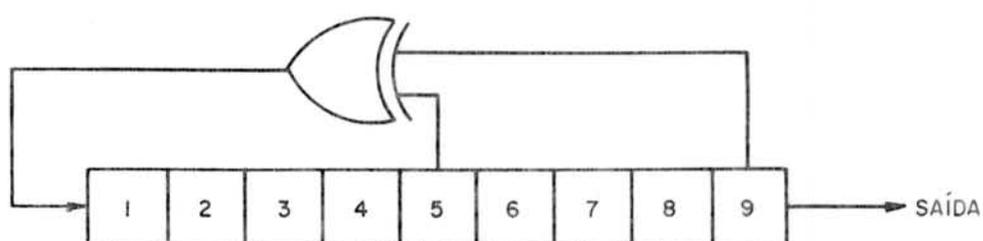


Figura 20 Geração do padrão 511

No STTP, por motivo de conveniência, economia de circuitos e espaço físico, o gerador 511 foi implementado por programa.

#### 3.5.4 Programação do MODO TESTE

A programação deste modo é feita pela técnica de MENU, descrita em 2.6, com o auxílio das teclas modo, parâmetro e opção. Apertando a tecla modo, aparecerá no mostrador um dos três modos em que o STTP poderá se encontrar. Com a

tecla opção, colocamos a mesma no MODO TESTE. A seguir, apertando sucessivamente a tecla PARÂMETRO, passaremos pelos três parâmetros do MODO TESTE: padrão, teste de erro e contagem. Para cada parâmetro escolhemos, com o auxílio da tecla opção, uma das várias opções disponíveis.

Apresentamos, a seguir, de maneira detalhada, os parâmetros e opções do MODO TESTE.

### 1. PADRÃO

PADRÃO - define o padrão de bits que será enviado pela linha para realizar a contagem de erros. Estes padrões são os recomendados pela CCITT V.52(5). As opções são: espaço permanente, marca permanente, 1:1, 3:1, 1:3, 7:1, 1:7 e o padrão 511.

### 2. TEST ER

TESTE DE ERRO - define se a contagem de erros é de bits, caracteres ou blocos errados. Um bloco é considerado como uma seqüência de 511 bits. As opções de teste de erro são; bit, caracter e bloco.

### 3. CONTAG

CONTAGEM - define o número de bits (ou caracteres ou blocos) que serão enviados durante o teste.

As opções de contagem são:

OPÇÃO 1: CONTAG [ 10E6 ]

significa que serão enviados um milhão de elementos (bits, caracteres ou blocos).

OPÇÃO 2: CONTAG [ 10E3 ]

significa que serão enviados mil elementos (bits, caracteres ou blocos).

OPÇÃO 3: CONTAG [ CONTÍNUO ]

significa que a transmissão de elementos se dará continuamente e a contagem de erros prosseguirá indenifidamente até que o operador encerre o teste pressionando a tecla PARAR.

Observação -

As contagens de erros de caracter e bloco são mostradas no painel em tempo real. A contagem de erros de bit, no entanto, somente é apresentada no final do teste. Isto ocorre devido ao tempo de programa para este teste, nas velocidades mais altas, não comportar manter um contador em tempo real no mostrador.

Exemplo de programação do MODO TESTE -

Neste exemplo, desejamos enviar o padrão CCITT 511<sup>5</sup> e testar a taxa de erros de bits por milhão.

- 1) Apertar a tecla MODO.
- 2) Levar o STTP para o MODO TESTE, apertando sucessivamente a tecla OPÇÃO até aparecer:
 

MODO [ TESTE ]
- 3) Apertar a tecla PARÂMETRO. Deve aparecer o parâmetro padrão.
- 4) Apertar sucessivamente a tecla OPÇÃO até aparecer:

PADRÃO [ 511 ]

5) Apertar a tecla PARÂMETRO. Deve aparecer o parâmetro teste de erro.

6) Apertar sucessivamente a tecla OPÇÃO até aparecer:

TEST ER [ BIT ]

7) Apertar a tecla PARÂMETRO. Deve aparecer o parâmetro contagem.

8) Apertar sucessivamente a tecla OPÇÃO até aparecer:

CONTAG [ 10E6 ]

9) Pronto, poderemos conferir a programação apertando sucessivamente a tecla PARÂMETRO.

10) Para iniciar o teste, basta apertar agora a tecla EXECUTAR.

### 3.5.5 Resumo dos parâmetros e opções do MODO TESTE

PADRÃO [ 0 ]  
 [ 1 ]  
 [ 1:1 ]  
 [ 3:1 ]  
 [ 1:3 ]

[ 7:1 ]  
 [ 1:7 ]  
 [ 511 ] \*

TEST ER [ BIT ] \*  
 [ CHARACTER ]  
 [ BLOCO ]

CONTAG [ 10E6 ] \*  
 [ 10E3 ]  
 [ CONTINUO ]

OBS.: O asterisco indica a opção inicial do parâmetro.

### 3.6 - Diretivas de edição

Estas diretivas são usadas para a edição do programa de simulação e das mensagens de transmissão e recepção.

A sintaxe desta diretiva é a seguinte:

EDITAR<NUMERO DE 0 A 3>

#### 3.6.1 Editar,0

Esta diretiva é utilizada para iniciar a edição do programa de simulação. Ao entrarmos com esta diretiva, é exibido no mostrador a primeira linha do programa simulador.

01( )

### 3.6.2 Editar,1

Esta diretiva é usada para permitir a edição das mensagens de recepção. Ao entrarmos com esta diretiva, é exibido no mostrador o primeiro bloco de mensagem de recepção.

B0 ( )

### 3.6.3 Editar,2

Esta diretiva é usada para permitir a edição das mensagens de transmissão A0 a A3. Ao entrarmos com esta diretiva, é exibido no mostrador o primeiro bloco de transmissão.

A0 ( )

### 3.6.4 Editar,3

Esta diretiva é usada para permitir a edição da mensagem de transmissão de 256 caracteres. Ao entrarmos com esta diretiva, é exibido no mostrador o início do bloco A4.

A4 (

## 3.7 Diretivas de listagem

Uma série de diretivas de listagem permite ao operador visualizar os resultados dos testes realizados pelo

STTP. A sintaxe desta diretiva é a seguinte:

LISTAR<NÚMERO DE 0 A 8>

3.7.1 Listar,0

Esta diretiva lista no mostrador as últimas 32 posições da MEMÓRIA MONITORA.

3.7.2 Listar,1

Esta diretiva lista no mostrador as primeiras 32 posições da MEMÓRIA MONITORA.

3.7.3 Listar,2

Esta diretiva lista no mostrador os últimos 32 caracteres armazenados na MEMÓRIA MONITORA.

3.7.4 Listar,3

Esta diretiva lista no mostrador a contagem de eventos ocorridos durante o teste.

Contagem eventos = 00000000

## 3.7.5 Listar,4

Esta diretiva lista no mostrador o resultado do último intervalo de tempo medido.

Int.Tempo = 000000 ms

## 3.7.6 Listar,5

Esta diretiva lista no mostrador o número de repetições ocorridas durante uma simulação.

Repetições = 000000

## 3.7.7 Listar,6

Esta diretiva lista no mostrador o número de erros detectados durante a medição de taxa de erro.

Erros = 00000000

## 3.7.8 Listar,7

Esta diretiva lista na impressora o conteúdo da MEMÓRIA MONITORA em código ASCII.

### 3.7.9 Listar,8

Esta diretiva lista na impressora o conteúdo da MEMÓRIA MONITORA em código hexa.

## 3.8 Descrição funcional das teclas especiais

Um conjunto de teclas especiais preenche as funções de comando e controle do STTP. A função destas teclas é apresentada abaixo.

### INICIO -

Esta tecla tem por finalidade colocar o sistema no estado de controle, sob comando do teclado, além de inicializar o sistema. Esta tecla é pressionada logo após o sistema ser ligado e sempre quando, por algum motivo, o sistema entrar em "dead-lock".

### MODO -

Esta tecla mostra o modo atual do sistema e permite a escolha de um dos três modos possíveis de operação: teste, monitor e simulador. Esta tecla também é utilizada para a escolha das opções dos diversos parâmetros de cada modo, ou seja, permite a programação dos modos.

## FORMATO -

Esta tecla permite a programação do formato de transmissão através da técnica de "menu", com a utilização das teclas parâmetro e opção. Os parâmetros de formato incluem: código, velocidade, paridade, modo de transmissão e relógio.

## LISTAR -

Esta tecla, seguida de um número de 0 a 8, permite listar:

Listar,0 - lista últimas 32 posições da memória monitora;

Listar,1 - lista primeiras 32 posições da memória monitora;

Listar,2 - lista últimos 32 caracteres recebidos;

Listar,3 - mostra contagem de eventos;

Listar,4 - mostra último intervalo de tempo medido;

Listar,5 - mostra número de repetições;

Listar,6 - mostra número de erros;

Listar,7 - lista memória monitora, em ASCII, na impressora;

Listar,8 - lista memória monitora, em hexa, na impressora.

## EDITAR -

Esta tecla, seguida de um número de 0 a 3, permite:

Editar,0 - edita programa de simulação;  
Editar,1 - edita mensagens de recepção;  
Editar,2 - edita mensagens de transmissão;  
Editar,3 - edita memória de transmissão de 256  
caracteres.

#### PARÂMETRO -

Esta tecla permite a visualização dos diversos pa  
râmetros de formato, ou dos modos teste, monitor  
e simulador. Ela mostra também a opção de cada pa  
râmetro chamado.

#### OPÇÃO -

Permite a programação e verificação das opções de  
cada parâmetro apresentado. Esta tecla, juntamen-  
te com a tecla parâmetro é utilizada para imple-  
mentar a técnica MENU de programação, ver (2.6).

#### QUALQUER -

Define um caracter que aceita comparação com qual  
quer caracter. Este caracter é utilizado, por e-  
xemplo, quando desejamos detectar uma seqüência  
menor que oito caracteres. As posições restantes  
são preenchidas com o caracter QUALQUER, signifi-  
cando que qualquer caracter naquelas posições se-  
rá aceito como pertencente à seqüência.

## NÃO -

Indica que o próximo caracter digitado é do tipo NÃO. Um caracter deste tipo aceita comparação com qualquer caracter exceto ele próprio.

## DIREITA -

Esta tecla permite deslocar os dados da MEMÓRIA MONITORA, como são mostrados, para a direita. Também permite, quando existir cursor no mostrador, deslocar o mesmo para a direita.

## ESQUERDA -

Esta tecla permite deslocar os dados da MEMÓRIA MONITORA, como são mostrados, para a esquerda. Também permite quando existir cursor no mostrador, deslocar o mesmo para a esquerda.

## EXECUTAR -

Esta tecla coloca o sistema no ESTADO DE EXECUÇÃO. Neste estado o sistema se encontra realizando uma monitoração, uma simulação ou um teste.

## PARAR -

Esta tecla encerra qualquer teste em execução pelo sistema, mostra o resultado do teste correspondente e retorna para o estado de controle.

**ENTRAR -**

Esta tecla é utilizada para edição e verificação das mensagens de transmissão e recepção e do programa de simulação. Na edição do programa de simulação, por exemplo, ao apertarmos esta tecla, a linha do programa mostrada na tela é guardada e é trazida a linha seguinte.

**FUNÇÃO -**

Esta tecla atualmente se encontra desligada. Sua função é prever futuras expansões do sistema.

**3.9 Condições de erro**

Ao apertarmos a tecla EXECUTAR, o STTP entra no ESTADO DE EXECUÇÃO. Caso algum parâmetro esteja programado fora das especificações, haverá um retorno para o ESTADO DE CONTROLE com uma indicação de erro presente no mostrador. Apresentamos as condições de erro detectadas pelo sistema e os possíveis motivos do erro.

**1. <<ERRO>> STOP BITS**

Indica que foi programada a opção de transmissão assíncrona e o número de "stop bits" especificado está errado. Este número de stop deve ser 1 ou 2.

## 2. &lt;&lt;ERRO&gt;&gt; CAR SINC

Indica que foi programada a opção de transmissão síncrona e o operador não digitou o(s) caracter(es) de sincronismo no campo usuário alfanumérico existente.

## 3. &lt;&lt;ERRO&gt;&gt; SINTAXE PROG.SIMUL.

Indica um erro de sintaxe no programa de simulação digitado pelo operador. Verificar os comandos do programa de simulação.

## 4. &lt;&lt;ERRO&gt;&gt; LINHA V.24

Indica que a(s) linha(s) escolhida(s) para a contagem de intervalo de tempo ou não existem no interface V.24 ou não apresentam transição para que a medida seja feita. As linhas permitidas para esta medida são: 2,3,4,5,8,12,13, 14, 15,16,17,19,21,22,23 e 24.

## 3.10 MENU

Apresentamos abaixo o MENU de operação do STTP.

## FORMATO -

Código .....: ASCII, EBCDIC e BAUDOT;

Número de bits ....: 5, 6, 7 e 8;

Paridade .....: par, ímpar, BCC/par, BCC/ímpar, CRC16;

Modo de transmissão : síncrono e assíncrono;

Velocidade .....: 9600, 4800, 2400, 1200, 600,  
 300, 150, 110, 75 e 50 bps;  
 Canal .....: principal e secundário;  
 Alarme .....: ligado e desligado;  
 Relógio .....: interno e externo.

MODO MONITOR -

Evento .....: seqüência de caracter,  
 erro de recepção e  
 intervalo de tempo;  
 Teste .....: conta eventos,  
 evento inicia armazenagem e  
 evento finda armazenagem;  
 Janela .....: desligada,  
 evento + "n" caracteres e  
 evento + "n" caracteres (ne-  
 gativo);  
 Início int. tempo .: transição "0" para "1" da lil  
 nha "n" e  
 transição "1" para "0" da lil  
 nha "n";  
 Fim intervalo tempo: transição "0" para "1" da lil  
 nha "n" e  
 transição "1" para "0" da lil  
 nha "n".

MODO SIMULADOR -

Evento .....: seqüência de caracter,

erro de recepção e  
intervalo de tempo;

Teste .....: conta eventos,  
evento inicia armazenagem e  
evento finda armazenagem;

Janela .....: desligada,  
evento + "n" caracteres e  
evento + "n" caracteres (ne-  
gativo);

Início int. tempo .: transição "0" para "1" da li  
nha "n" e  
transição "1" para "0" da li  
nha "n";

Fim intervalo tempo: transição "0" para "1" da li  
nha "n" e  
transição "1" para "0" da li  
nha "n";

Simula .....: equipamento de transmissão  
de dados e  
equipamento de comunicação  
de dados;

Elemento .....: mestre e escravo.

MODO TESTE -

Padrão .....: 0, 1, 1:1, 3:1, 1:3, 7:1,  
1:7 e 511;

Teste de erros ....: bit, caracter e bloco;

Contagem .....: 10E6, 10E3 e contínuo.

Este capítulo apresenta a solução utilizada na implementação do sistema definido nos capítulos anteriores. São descritos aqui todos os blocos funcionais que constituem o equipamento.

#### 4.1 Introdução

Devido às características almeçadas na definição deste sistema, ficou patente a necessidade do uso de um microprocessador para coordenar as atividades de aquisição de dados, processamento da informação e controle dos circuitos auxiliares. A não utilização de um microprocessador implicaria num aumento desmesurado de circuitos e de chaves no painel de comando, além de aumentar a complexidade de operação do equipamento. Por todos estes motivos, e considerando ainda que os microprocessadores constituem uma solução técnica altamente recomendável nestes casos, optamos pela sua utilização.

O microprocessador escolhido foi o 8080A<sup>19</sup> por vários motivos: existência de apoio de "Software" de desenvolvimento no HP-2100 e no SED (Sistema de Entrada de Dados), ambos disponíveis nos laboratórios da PCCC; existência de apoio de "Hardware", como estoque de interfaces para o 8080A disponíveis no almoxarifado; padronização no uso dos recursos disponíveis; baixo custo e performance adequada à tarefa; expectativa de nacionalização deste componente.

A figura 21 apresenta um diagrama de blocos de todo o sistema. Como podemos ver, o STTP é constituído de uma série de blocos funcionais estanques, conectados entre si, através de um barramento. A descrição sucinta dos sinais deste barramento aparece no apêndice F. O módulo da UCP possui uma ligação extra com o bloco de memória. Esta ligação é constituída das linhas de endereço e de comandos de leitura e escrita na memória. Estas ligações não estão no barramento por dois motivos: primeiro - porque são sinais que ligam apenas os dois blocos, e segundo - para descongestionar o barramento que, possuindo espaço para apenas 44 sinais, já se encontrava totalmente ocupado.

A unidade central de processamento (UCP) se comunica com o mundo exterior através de interfaces específicos, cuja função é compatibilizar os sinais externos com os sinais internos do sistema. Existem quatro interfaces que realizam esta tarefa.

O interface mostrador permite à UCP escrever mensagens e dados no mostrador de 32 caracteres existente no painel frontal.

O interface do teclado permite levar as informações entradas pelo operador através das teclas do painel até o processador central.

O interface da impressora permite à UCP despejar o conteúdo da sua memória monitora numa impressora, através do conector J3 existente no painel traseiro do STTP.

Figura 21 Diagrama em blocos do STTP

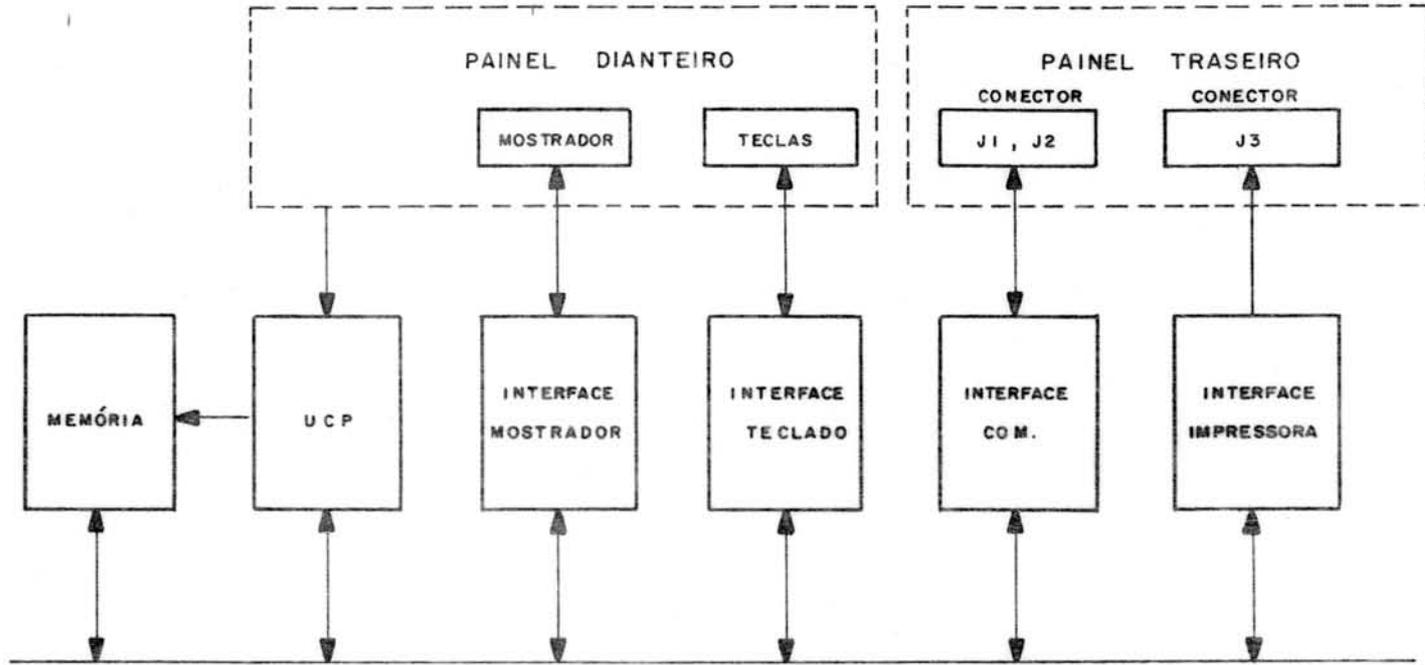


DIAGRAMA EM BLOCOS DO STTP

Finalmente temos o interface de comunicações que possibilita ao sistema interagir com o canal de comunicações (V.24) através dos conectores J1 e J2 existentes no painel traseiro.

#### 4.2 Unidade Central de Processamento

Esta unidade é responsável pela execução do programa armazenado na memória de leitura apenas, o que, em suma, exerce controle sobre todo o sistema, realiza aquisições de dados e informações de entrada, envia dados pelos canais de comunicação e comandos pelas portas de saída.

Esta unidade está localizada em uma placa específica, placa 1.

Na figura 22, apresentamos um diagrama em blocos da unidade central de processamento. Fazem parte desta unidade: o processador central (8080A), o controlador de acesso direto à memória (8257) e o controlador de interrupções (8214).

Figura 22 Diagrama em blocos da unidade central de processamento (UCP)

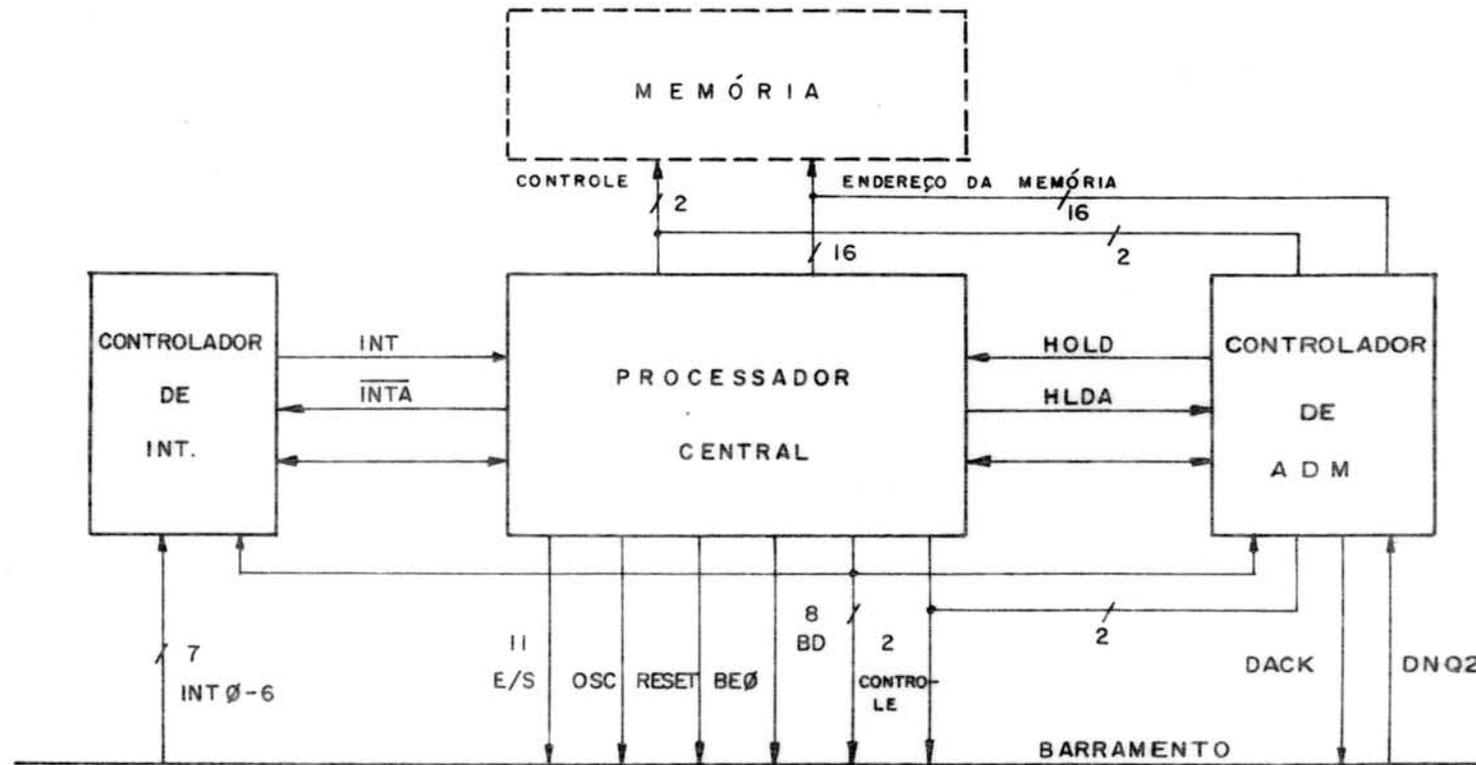


DIAGRAMA EM BLOCOS DA UNIDADE CENTRAL DE PROCESSAMENTO

#### 4.2.1 Processador Central

O processador central liga-se com o controlador de interrupções e com o controlador de acesso direto à memória. Todos os blocos se ligam ao barramento. A unidade central de processamento também se liga diretamente à memória, sem passar pelo barramento. Esta ligação é constituída das linhas de endereço e de dois sinais que controlam a leitura e escrita de dados da memória. O processador central decodifica os endereços das portas de entrada e saída, colocando diretamente no barramento o endereço de cada porta. Esta decodificação central se deve ao fato de ser pequeno o número de portas de entrada e saída. Desta maneira as diversas placas não precisam de decodificadores locais. O processador também envia pelo barramento os seguintes sinais:

- OSC - frequência de 18,432 MHz utilizada pela interface de comunicações para gerar as diversas taxas de transmissão de dados;
- RESET - que serve para colocar no estado inicial todos os interfaces;
- CONTROLE- que são dois sinais que controlam a leitura e escrita nas portas de entrada e saída;
- BE0 - linha zero de endereços. Serve para endereçar registradores internos das USART's.
- BD - barramento de dados, 8 linhas.

#### 4.2.2 Controlador de interrupções

O controlador de interrupções (8214) permite o atendimento de vários periféricos que funcionam à base de interrupção. Ele possui oito níveis de interrupção, podendo ser desabilitado segundo a ordem de prioridade. Chegam através do barramento 7 dos níveis de interrupção, que são utilizados pelos diversos interfaces. Este controlador é programável, de modo que o processador pode habilitar e desabilitar cada um dos níveis de interrupção. Ao receber um pedido de interrupção, através das linhas INTO-7, o controlador envia o sinal INT ao processador. Este responde com o sinal INTA se estiver em condições de atender a interrupção. O controlador, por sua vez, coloca no barramento de dados a instrução RESTART com o código correspondente ao nível que interrompeu. O processador executa a instrução desviando o programa para o início da rotina de atendimento do periférico que pediu interrupção.

Apresentamos, a seguir, um exemplo de programação deste controlador de interrupção. Observar que os mnemônicos utilizados para o assembler são os descritos em [21]. O exemplo habilita todos os níveis de interrupção:

```
      :  
      LAI 10  
      OUT 20  
      :  
      :
```

#### 4.2.3 Controlador de ADM

O controlador de ADM(8257) é utilizado para realizar transferências de dados da memória para o mostrador sem utilizar o microprocessador. Utilizou-se a técnica de ADM para evitar sobrecarregar o microprocessador com uma tarefa simples, porém, de grande consumo de tempo. Como podemos ver pela figura 21, o controlador de ADM está ligado às linhas de endereço e de controle de leitura e escrita da memória e das portas de entrada e saída. Isto se deve ao fato do controlador assumir o controle destes sinais durante as transferências de ADM. O controlador recebe o pedido de ADM através do sinal DRQ2 e envia ao processador o sinal HOLD, pedindo permissão para realizar a transferência de ADM. O processador responde com o sinal HLDA, indicando que a transferência pode ser realizada. O controlador envia o sinal DACK2 ao periférico, indicando que o pedido de ADM foi aceito, assume o controle dos barramentos de dado e endereço e inicia a transferência de dados conforme estiver programado.

A programação é feita pelo programa residente durante a inicialização do sistema. Esta programação indica o endereço inicial de memória, o número de caracteres a transferir e o modo da operação de ADM.

O endereço inicial de memória é enviado em duas partes ao registrador RACH2, cujo endereço é 4. Primeiro os 8 bits menos significativos e após os 8 bits mais significativos.

O número de caracteres a transferir é também en-

viado em duas partes ao registrador RTCH2, cujo endereço é 5. Primeiro os 8 bits menos significativos e depois os 8 bits mais significativos. Nos bits 6 e 7 do byte mais significativo vai a informação de leitura (bit 7 = 1) ou escrita (bit 6 = 1) da memória.

O registrador de modo, localizado no endereço 10, tem a sua configuração apresentada na figura 23.

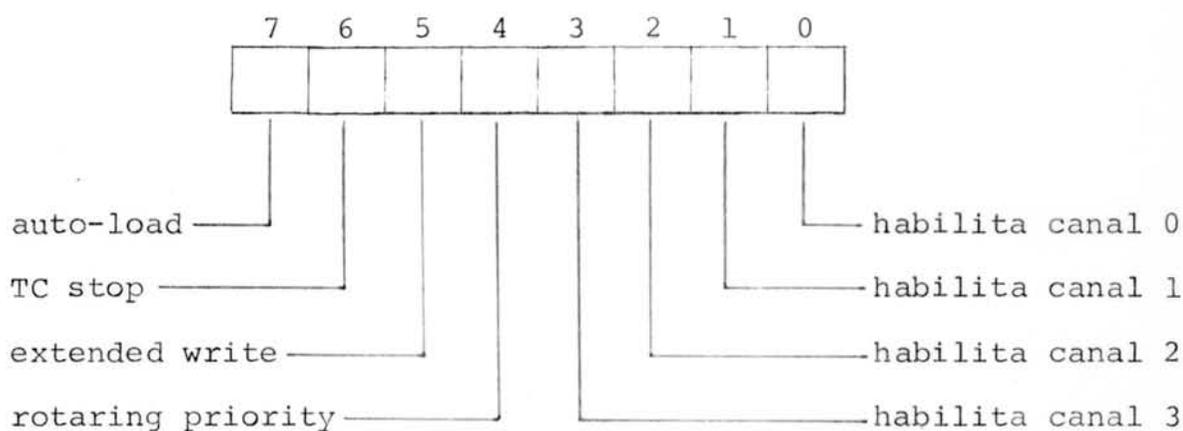


Figura 23 Registrador de modo do controlador de ADM

O modo "auto-load" permite ao canal 2 ser usado para operações de repetição ou encadeamento de blocos, sem a intervenção imediata do programa entre os blocos. Os registradores do canal 2 são inicializados normalmente: os registradores do canal 3, porém, são usados para guardar os parâmetros de reinicialização do canal 2 (endereço inicial, número de caracteres a transferir e modo de transferência). Após o primeiro bloco de ADM ser transferido, os parâmetros no canal 3 são transferidos para o canal 2 automaticamente. Assim, uma vez programado, o controlador continua indefinidamente fazendo

transferências de ADM da mesma área, no caso a área de tela. Isto é necessário para manter uma varredura atualizada de todas as linhas do "display" já que o mesmo mantém armazenado apenas o conteúdo de uma linha de cada vez.

Apresentamos, a seguir, um exemplo de programação deste controlador de ADM:

```

:
LHLI TELA * endereça início da área de tela
LAL
OUT 4 * programa canal 2
LAH
OUT 4 * coloca endereço inicial (tela)
LAI 37
OUT 5
LAI 200 * coloca número de bytes a transferir
OUT 5 * 32 caracteres.
LAL
OUT 6 * programa canal 3 idêntico ao canal 2
LAH
OUT 6 * coloca endereço inicial
LAI 37
OUT 7
LAI 200 * coloca número de bytes a transferir
OUT 7
LAI 204 * habilita canal 2 na opção de auto-load
OUT 10
```

```

:-
```

### 4.3 Memória

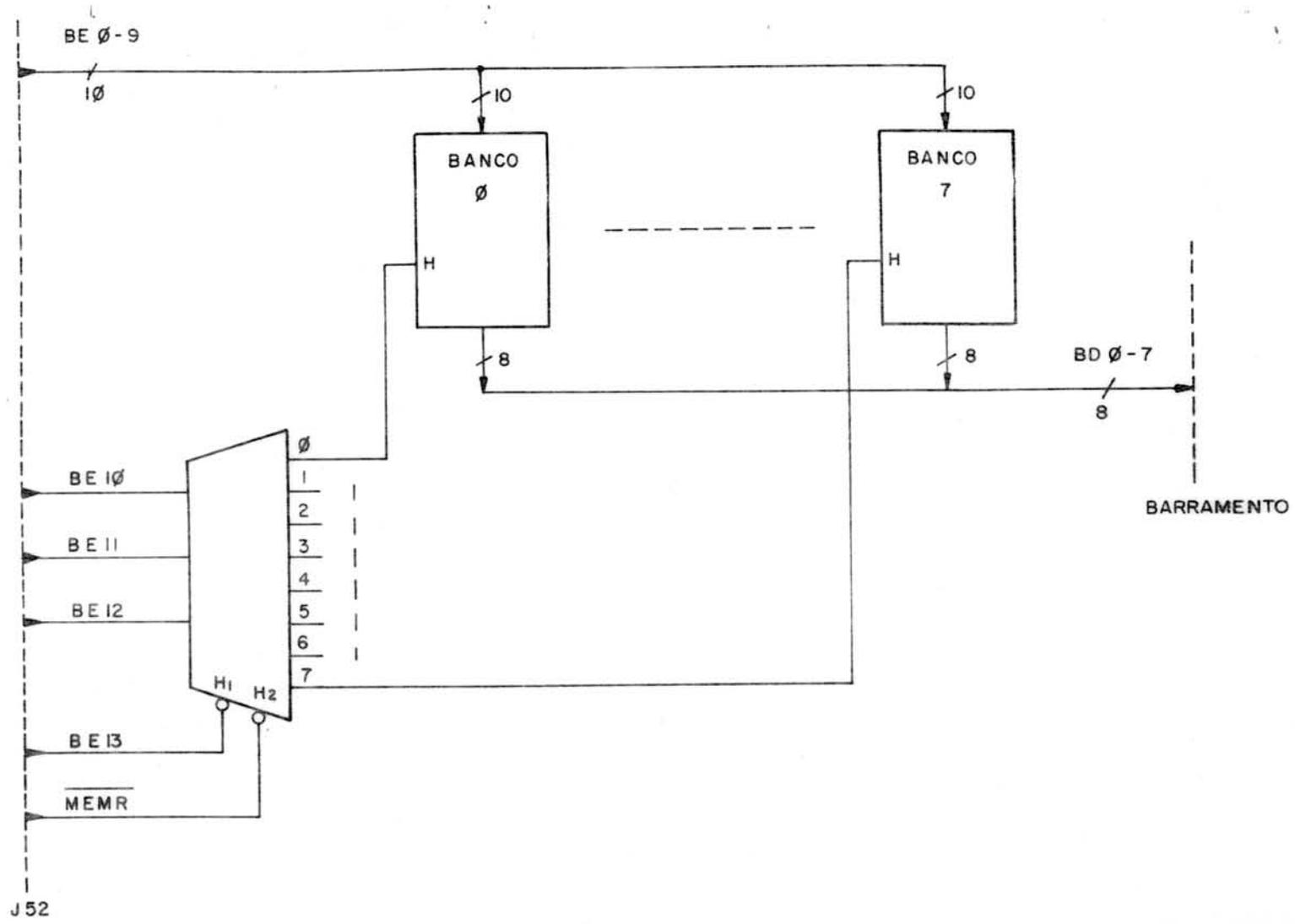
A memória do sistema ocupa uma placa específica, placa 2. Temos nesta placa dois tipos de memória: uma de leitura apenas e outra de leitura e escrita.

#### 4.3.1 Memória de leitura apenas

A memória de leitura apenas contém o programa que coordena todas as atividades do STTP. Foi construída a partir de pastilhas de REEPROM (memória reprogramável de leitura apenas) do tipo 2708, cuja capacidade é de 1024 palavras de 8 bits. Foram utilizadas oito destas pastilhas, totalizando uma capacidade de memória de 8192 palavras de 8 bits (8K bytes). Na figura 24, apresentamos um diagrama em blocos da memória de leitura apenas. Observar que o único sinal de comando é o de leitura da memória, MEMR. Os sinais de endereço, BE0 a BE9, endereçam as 1024 posições de todos os blocos da memória, mas cada bloco deve ser habilitado pelo decodificador de blocos. Este decodifica as linhas de endereço BE10 a BE12, habilitando os 8 blocos individualmente. A linha BE13 deve estar em "0" para permitir o endereçamento desta memória.

O endereçamento desta memória é de 0000 a 17777 (octal).

Figura 24 Memória de leitura apenas

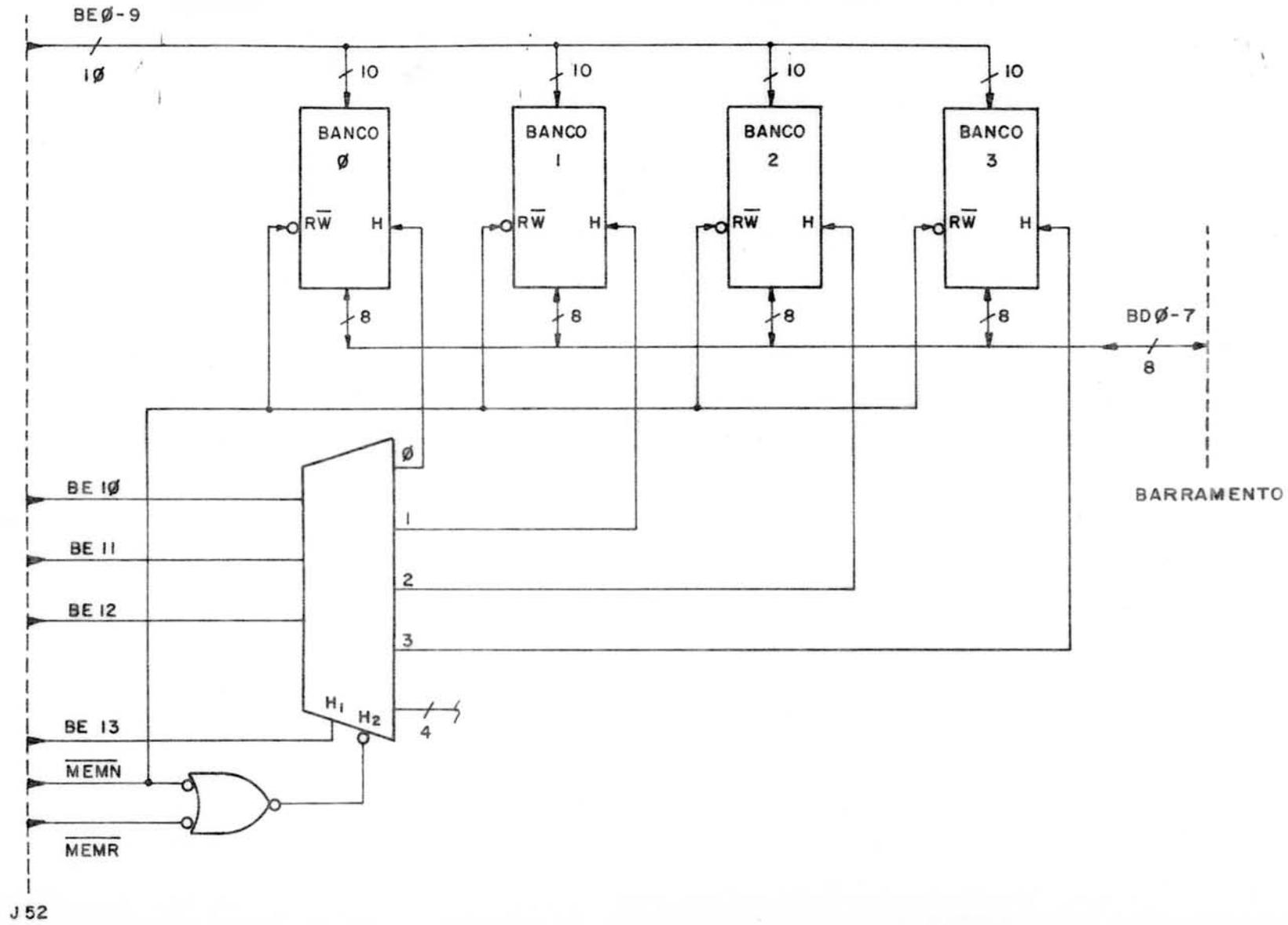


#### 4.3.2 Memória de leitura e escrita

A memória de leitura e escrita está dividida em duas partes. A primeira parte, nos endereços 20000 a 27777 (octal) (2K bytes) constitui a memória de rascunho onde a UCP coloca seus registradores temporários, a pilha e as mensagens de transmissão e recepção definidas pelo usuário. Na segunda parte, nos endereços 30000 a 37777 (octal) (2K bytes) está a memória monitora, onde é armazenado o fluxo de dados monitorados. Ela foi construída com pastilhas de RAM estática (memórias de acesso randômico) do tipo 2114, cuja capacidade é de 1024 palavras de 4 bits. Foram utilizadas oito pastilhas para totalizar 4096 palavras de 8 bits (4K bytes). Na figura 25, apresentamos um diagrama em blocos da memória de leitura e escrita. Aqui temos dois sinais de comando, o de leitura da memória, MEMR, e o de escrita na memória, MEMW. Os sinais de endereço, BE0 a BE9 endereçam as 1024 posições de todos os blocos. As linhas de endereço, BE10 a BE12, são decodificadas pelo decodificador de blocos que habilita individualmente cada bloco. A linha BE13 deve estar em 1 para esta memória ser endereçada.

O endereçamento desta memória é de 20000 a 37777 (octal).

Figura 25 Memória de leitura e escrita



J 52

#### 4.4 Interface do teclado

Este interface está localizado na placa 3. Ele tem por finalidade transferir para a UCP as informações entradas pelo operador através do teclado. O teclado é constituído de duas partes: teclado hexadecimal e conjunto de teclas especiais. Internamente, no entanto, esta diferença não existe, e todas as teclas são codificadas pelo interface de modo seqüencial. A tecla INÍCIO não é atendida por este interface, ver 4.8.4.

Uma vez o operador tendo pressionado uma tecla, o interface se encarrega de codificar a mesma, guardar este código em um registrador e avisar a UCP, através de uma linha de interrupção, que uma tecla foi acionada. Daí por diante, é função da UCP ler o código da tecla digitada no registrador, onde o mesmo foi guardado pelo interface.

As teclas do painel e os seus respectivos códigos são apresentadas na tabela 3.

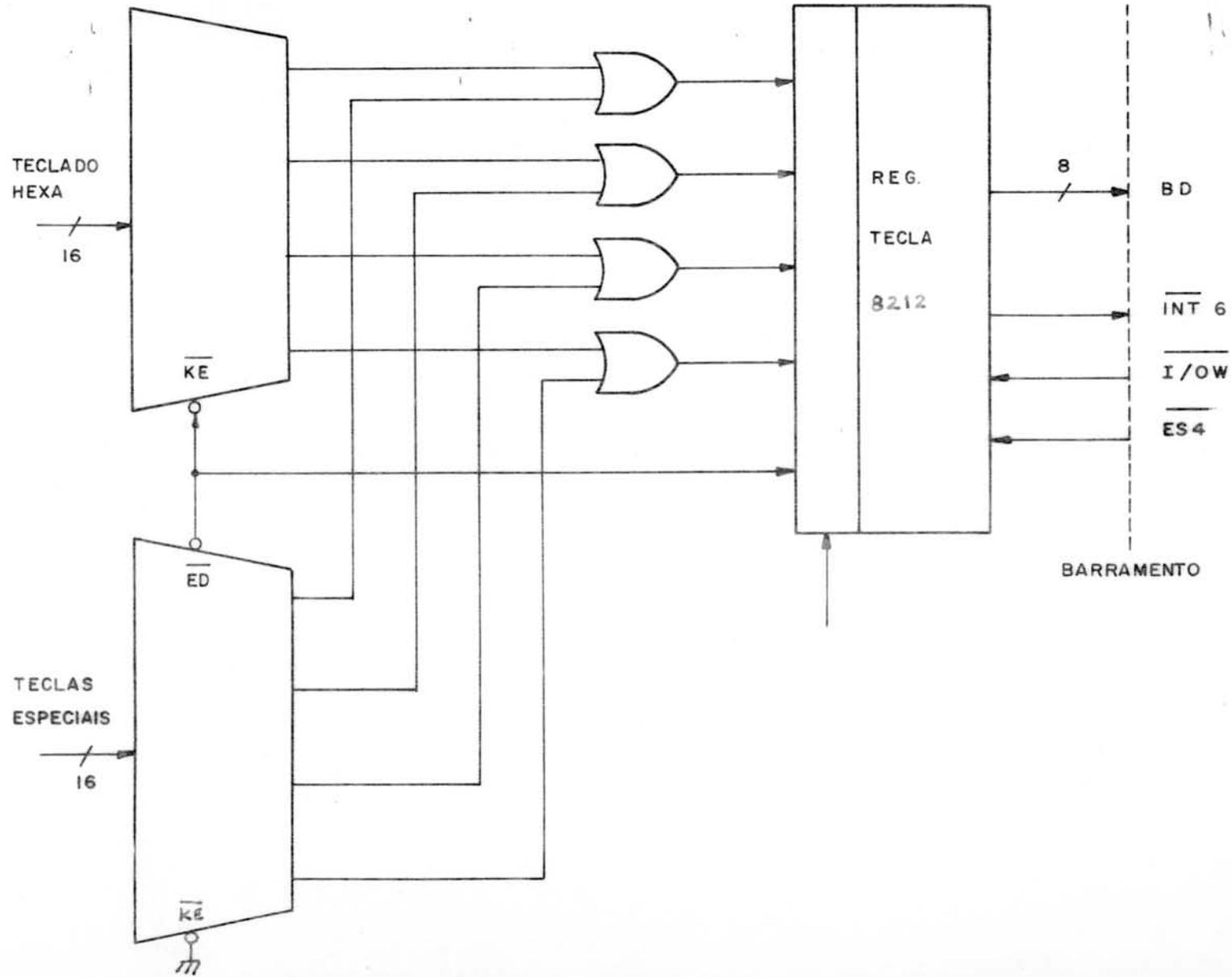
TECLA	CÓDIGO OCTAL
Hexa 0	00
Hexa 1	01
Hexa 2	02
Hexa 3	03
Hexa 4	04
Hexa 5	05
Hexa 6	06
Hexa 7	07
Hexa 8	10
Hexa 9	11
Hexa A	12
Hexa B	13
Hexa C	14
Hexa D	15
Hexa E	16
Hexa F	17
Esquerda	20
Direita	21
Qualquer	22
Não	23
Entrar	24
Listar	25
Executar	26
Parâmetro	27
Opção	30
Formato	31
Modo	32
Editar	33
Parar	34
Função	35

Tabela 3 Código das teclas

Na figura 26, vemos o diagrama simplificado do

interface do teclado. Há dois decodificadores de 16 para 4, um para o teclado hexa e outro para as teclas especiais. Estas têm prioridade sobre o teclado hexa. Os dados são codificados em 5 bits e guardados no registrador TECLA. Ao ser pressionada uma tecla, o interface guarda o seu código neste registrador e aciona a linha de interrupção INT6. Ao ser atendida a interrupção, o programa lê o conteúdo do registrador tecla pela instrução IN 100. Ao executar esta instrução, o processador ativa as linhas E/S4 (endereço do registrador) e I/OR (comando de leitura). Com isto, o interface coloca no barramento de dados o conteúdo do registrador tecla e retira o pedido de interrupção.

Figura 26 Interface do teclado



Na figura 27, apresentamos o processo de "Debounce" empregado. São utilizados circuitos CMOS com histerese obtida através de realimentação positiva.

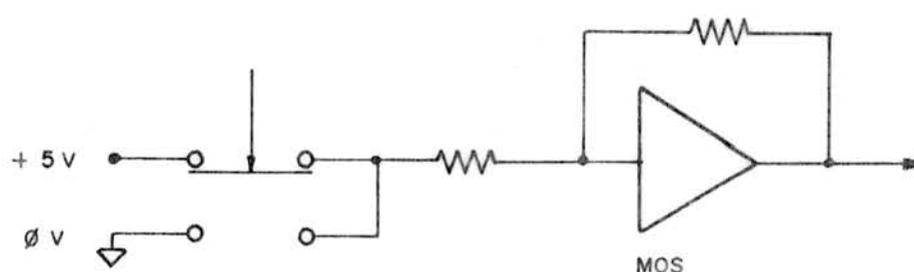


Figura 27 Técnica de "Debounce" utilizada

As teclas ESQUERDA e DIREITA possuem a facilidade de repetição. Isto permite que a ação da tecla seja repetida enquanto ela estiver sendo pressionada. Na figura 28, apresentamos a solução encontrada. Foi utilizado um multivibrador estável, à base de circuitos CMOS, comandado pela ação da tecla. A constante de tempo RC determina a frequência de oscilação do circuito.

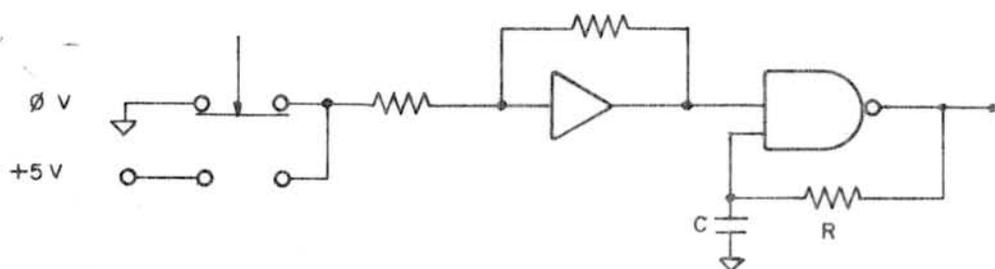


Figura 28 Circuito de repetição

## 4.5 Interface mostrador

A finalidade deste interface é transferir, de maneira adequada, os dados da memória para o mostrador.

### 4.5.1 Varredura do mostrador

Existe uma área da memória de leitura e escrita, chamada área de tela, de 32 posições, que é constantemente exibida pelo mostrador. Sempre que a UCP deseja exibir alguma coisa no mostrador, ela preenche esta área com o conteúdo a mostrar. É função do interface mostrador mostrar constantemente o conteúdo da área de tela.

A maneira como isto é feito é através da técnica de acesso direto à memória (ADM). O mostrador é constituído de 32 células. Cada célula é uma matriz 5x7 de pontos luminosos. Não é possível acender todos os pontos da matriz ao mesmo tempo. Mas, por um processo de varredura e levando em conta a persistência do olho humano, tem-se a impressão de ver os vários pontos da célula acesos a um só instante. Existem dois processos de varredura de células deste tipo (TIL501): varredura por coluna e varredura por linha. Na varredura por linha, acendemos os pontos de uma linha por vez. Na varredura por coluna, acendemos os pontos de cada coluna por vez. Como o mostrador possui 32 células colocadas lado a lado, temos na realidade  $32 \times 5 = 160$  colunas. Varrer 160 colunas uma após a outra é impraticável, porque teríamos um regime de trabalho

para cada ponto luminoso de  $1/160$ , ou seja,  $0,63\%$ , quando o recomendado é de  $20\%$ . Outra solução é realizar a varredura por linha o que nos dá um regime de trabalho de  $1/7$ , ou seja,  $14\%$ , o que, se não é ideal, é bem mais razoável. Na prática, esta última solução mostrou-se bastante satisfatória. Adotando-se uma varredura por quadro de 60 ciclos, teremos que cada linha deverá permanecer acesa por  $1/(60 \times 7)$ , ou seja,  $2,381$  ms. Para manter uma linha acesa durante este período, tornou-se necessário armazenar a informação da linha em registradores. Foram utilizados 32 registradores de 6 bits, um para cada célula. Na figura 29, vemos uma célula do mostrador com o registrador de linha que mantém a informação dos bits da linha acionada pelo seletor de linhas. Os 6 bits mantêm a informação das 5 colunas mais o ponto decimal existente na sétima linha. Estes 32 registradores devem ser carregados com a informação da nova linha a cada período de  $2,381$  ms. Desta maneira, a cada período de  $2,381$  ms, o interface realiza um pedido de ADM. O controlador de ADM aceita este pedido ao final da execução da instrução em andamento na UCP. Esta pára as suas funções e entrega o controle dos barramentos de dado e endereço ao controlador de ADM que, então, realiza as transferências conforme estiver programado. Após transferir 32 caracteres, o interface encerra o pedido de ADM. O controlador de ADM reconhece o fim do ciclo de transferência e retorna o controle para o microprocessador que continua suas funções de processamento a partir do ponto onde havia parado.

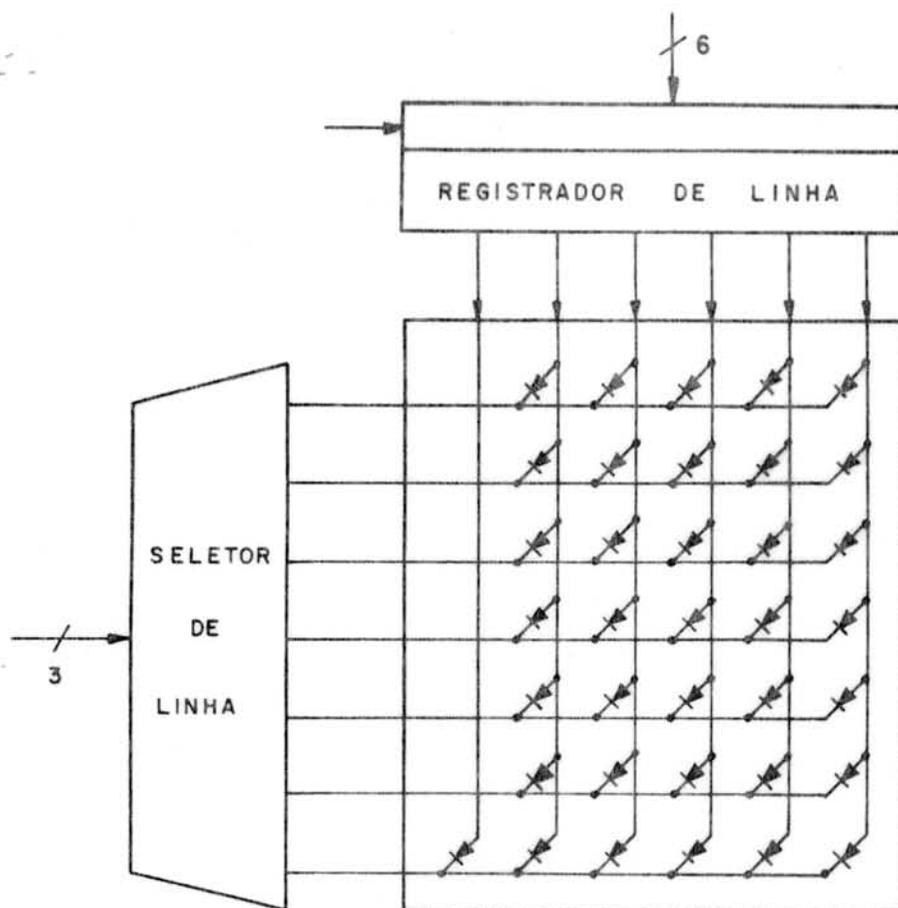


Figura 29 Acionamento da matriz de pontos luminosos

#### 4.5.2 Controle do cursor

Neste interface existe ainda um cursor que faz piscar a célula correspondente, acendendo todos os seus pontos luminosos, de modo a destacá-la das demais células. Existem dois comandos que controlam este cursor. O primeiro é o comando que habilita e desabilita o cursor. Para isto a UCP deve:

Escrever 000 no endereço HCursor=60 para desabilitar o cursor

Escrever 001 no endereço HCursor=60 para habilitar o cursor

O outro comando coloca o cursor em qualquer posição no mostrador. Para isto, a UCP deve escrever no endereço Cursor=40 o código binário da posição em que deseja colocar o cursor. Esta posição pode variar de 0 a 31.

Exemplo de programação do cursor:

⋮

XRA

OUT 60 \* Apaga cursor

⋮

LAI 20D

OUT 40 \* Coloca cursor na posição 20

LAI 1

OUT 60 \* Acende cursor

⋮

#### 4.5.3 Gerador de caracteres

Devido à necessidade de exibir caracteres especiais, como é o caso dos caracteres de controle de protocolo, foi construído um gerador de caracteres especial. Este gerador possui o conjunto completo dos caracteres representáveis em código ASCII mais os caracteres de controle que normalmente não possuem representação gráfica, num total de 128 caracte-

teres.

Um gerador de caracteres nada mais é do que uma memória que possui como entrada as linhas de endereço e como saída as linhas de dados lidos. O código ASCII de um determinado caracter, colocado à sua entrada, endereça uma determinada posição da memória, que aparecerá na saída e que representará graficamente o caracter. É lógico que uma simples posição de memória de 8 bits não pode armazenar informação suficiente para representar graficamente um caracter. A representação gráfica de um caracter é feita através de uma matriz de pontos. No nosso caso esta matriz é de 5x7 pontos. Para armazenar esta informação gráfica precisamos de no mínimo 5x7 bits, um para cada ponto. Por motivo de simplificação no endereçamento do gerador, foram utilizadas oito posições de memória para conter toda a informação do caracter. Utilizando uma memória do tipo 2708 de 1024 posições de 8 bits cada, temos um gerador com capacidade de  $1024/8 = 128$  caracteres. Isto é mais do que suficiente. Estas 8 posições de memória constituem a célula do caracter. Esta célula possui  $8 \times 8 = 64$  bits, mais do que o suficiente para armazenar os  $5 \times 7 = 35$  pontos de que precisamos para desenhar um caracter no "display". Na figura 30 apresentamos a célula de caracter do gerador. A área achuriada representa a ocupação real da célula.

Observamos que para representar um caracter completo, precisamos endereçar o gerador 7 vezes consecutivas. O gerador necessita de 10 bits para endereçar as suas 1024 posições. Os 7 bits mais significativos do endereço endereçam os 128 caracteres, enquanto que os 3 bits menos significativos

endereçam as linhas de cada caracter. É função do interface mostrador fornecer os 3 bits menos significativos, realizando a varredura das linhas de cada caracter.

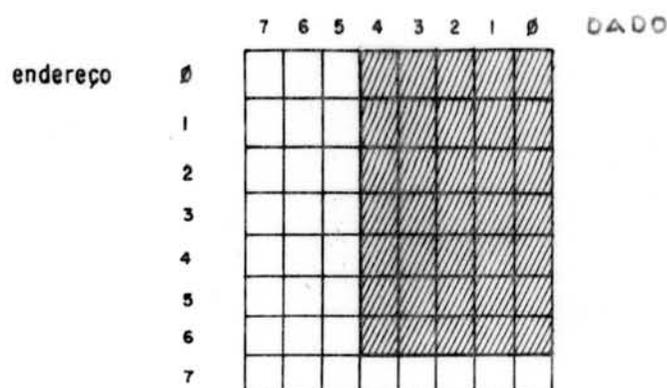


Figura 30 Ocupação da célula do gerador de caracteres

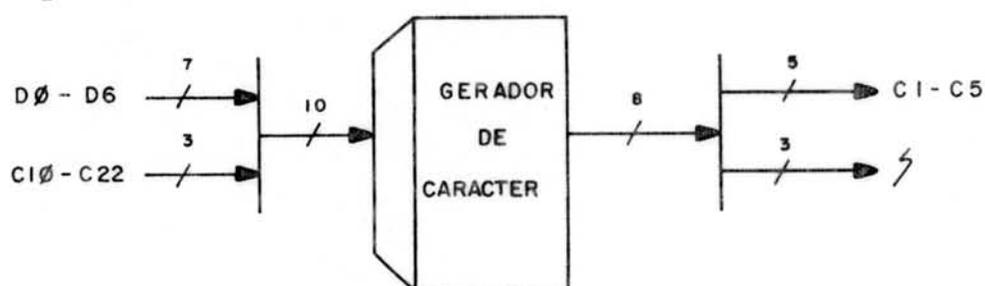
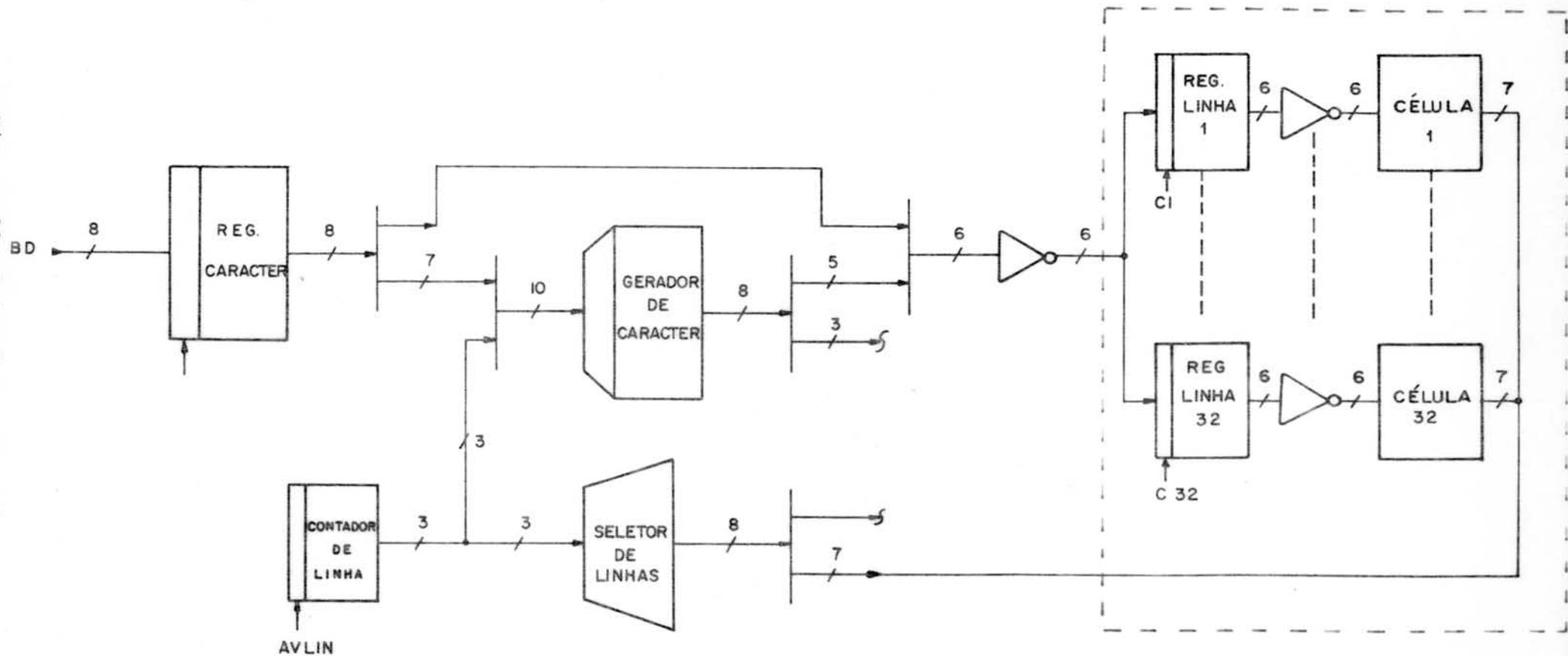


Figura 31 Gerador de caracteres

#### 4.5.4 Funcionamento do interface

Na figura 32, apresentamos um diagrama de blocos do interface mostrador. O caracter a ser exibido chega até o interface através das linhas de dado do barramento e é guardado no registrador de caracteres. O bit mais significativo contém a informação do ponto decimal e não passa pelo gerador de caracteres. Os outros 7 bits formam os bits mais significativos que endereçam o gerador. À saída do gerador, temos a informação de uma linha do caracter recebido. Esta informação é guardada no registrador de linha e mostrada na matriz de pontos luminosos. Existem 32 registradores de linha, cada um de 6 bits. Durante uma transferência de ADM, são enviados 32 caracteres ao interface. De cada caracter, é mostrada uma linha apenas. Na chamada de ADM seguinte são transferidos os mesmos 32 caracteres, só que é mostrada a linha seguinte dos mesmos. Assim por diante, até serem mostradas todas as linhas, quando o processo se repete. São necessárias 7 chamadas de ADM para mostrar um quadro completo. São varridos 60 quadros por segundo.

Figura 32 Interface mostrador



Para cada caracter transferido, é enviado ao interface os sinais I/O $\bar{W}$  e DACK2/. Estes sinais servem para armazenar o caracter no registrador de caracteres e também para armazenar o conteúdo gráfico da linha deste caracter no registrador de linha. Na figura 33, apresentamos os tempos em que são realizados estes dois armazenamentos. O tempo entre armazenar o caracter e armazenar a sua representação gráfica é de aproximadamente 750 nanosegundos. Este tempo deve ser maior que o tempo de acesso do gerador de caracteres que é de 450 ns. Assim, temos uma folga de 67% no acesso ao gerador.

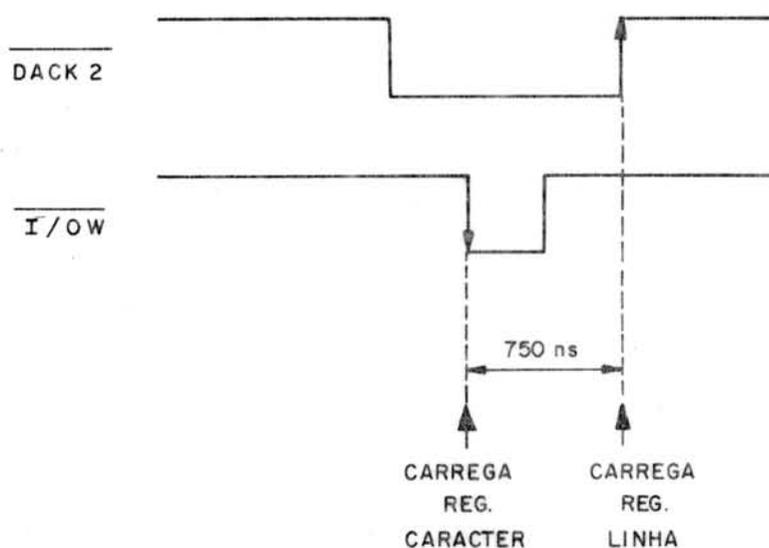


Figura 33 Carga dos registradores caracter e linha

O circuito contador de linhas fornece os 3 bits menos significativos de endereço ao gerador de caracteres, dando assim a informação da linha a ser exibida. Esta informação também deve chegar à matriz de pontos luminosos, para ser

acesa a linha correspondente. Isto é feito, passando os 3 bits que indicam a linha pelo circuito seletor de linhas. As linhas de todas as matrizes estão em paralelo, de modo que elas são acionadas simultaneamente. Como as matrizes possuem apenas 7 linhas, o contador de linhas é um contador binário truncado, que conta de 0 a 6. O contador de linhas avança a sua contagem na subida do sinal AVLIN, gerado pelo controle. Este sinal ocorre sempre ao término de cada transferência de ADM.

Na figura 34, apresentamos um diagrama de tempos onde aparece a ocorrência de AVLIN. Também aparece neste diagrama o sinal DRQ2 que é o pedido de ADM feito pelo interface. Este sinal é levantado a cada 3,28 ms por meio de um monoestável e baixado após a transferência dos 32 caracteres.

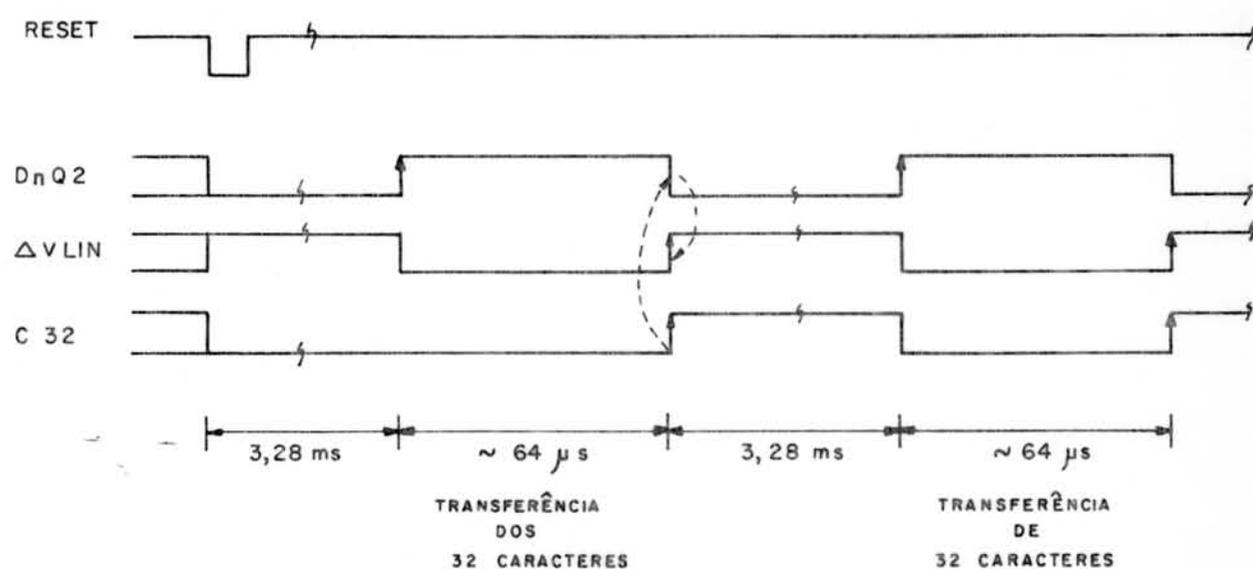


Figura 34 Diagrama de tempos do pedido de ADM (DRQ2)

Na figura 32, o retângulo pontilhado separa os circuitos montados na placa painel, junto ao mostrador. Nesta figura, representamos os circuitos que se repetem 32 vezes, um para cada caracter do mostrador. São eles o registrador de linha, o "driver" inversor e a própria matriz de pontos luminosos.

O circuito carregador, mostrado na figura 35, possui 32 saídas, exclusivas, que carregam um após o outro os 32 registradores de linha. Este circuito é constituído de um registrador deslocador de 32 estágios com saídas em paralelo. O sinal AVCAR faz avançar o sinal de carga pelos 32 estágios. Ao ser carregado o trigésimo segundo registrador de linha, é avisado ao controle que pode ser encerrado o pedido de ADM.

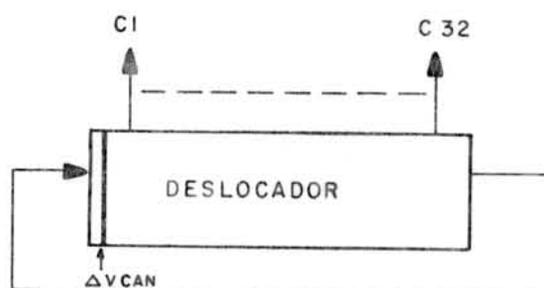


Figura 35 Circuito carregador

O circuito que gera o cursor é constituído de um registrador decodificador de 5 para 32, ver figura 36. A saída deste circuito está ligada à entrada "RESET" dos registradores de linha. Só uma saída estará ativa por vez. A saída ativa limpa o conteúdo do respectivo registrador. Como a saída

dos registradores de linha sofre uma inversão, o registrador em zero, faz acender todos os pontos da linha do "display" correspondente. Um circuito multivibrador astável, dentro do bloco de controle, ativa e desativa o cursor a uma frequência bem inferior às varreduras do mostrador. Assim, durante o período do ciclo em que o cursor estiver ativo, ocorrerão várias varreduras, o que mostrará todos os pontos acesos da matriz respectiva. Durante o período do ciclo em que o cursor estiver desabilitado, a matriz aparecerá com o seu conteúdo normal. Assim, o cursor "pisca" na frequência do astável.

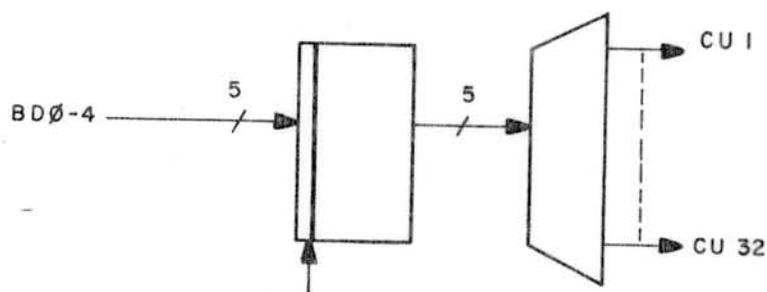


Figura 36 Circuito do cursor

#### 4.6 Interface de comunicações

Este interface é responsável pela ligação entre a UCP e as linhas de comunicação de dados do interface V.24.

#### 4.6.1 Relógio de 1 milisegundo

O relógio de 1 milisegundo serve como temporizador para serem feitas medidas de tempo em milisegundos. Este relógio é obtido pela divisão da frequência de 1,2288 MHz por 205 e após por 6, ver figura 37. O resultado é 999.024 Hz, o que nos dá um período de 1,001 ms, com uma precisão de 0,098%. Este relógio interrompe a UCP a cada ciclo. Como ele não é sincronizado, as medidas de tempo assim realizadas apresentam um erro de 1 ms. Esta precisão não parece, à primeira vista, muito satisfatória, porém ela está dentro de 10% nos piores casos, e em torno de 1% na maioria das medições, que se situam na ordem das centenas de milisegundos. Acrescente-se a isto o fato de todos os equipamentos semelhantes consultados apresentarem esta mesma precisão para as suas medidas de tempo.

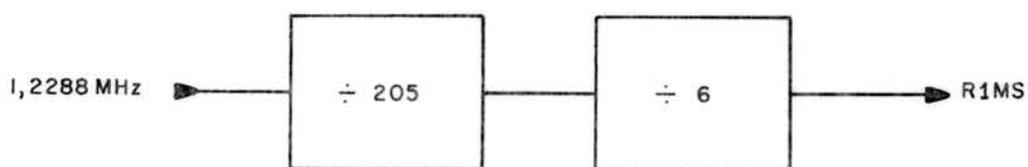


Figura 37 Relógio de 1 milisegundo

#### 4.6.2 Relógio programável

O relógio programável fornece às usart's a taxa de transferência dos dados. Este relógio permite a transferência de dados nas taxas de 9600, 4800, 2400, 1200, 600, 300, 150, 110, 75 e 50 bits por segundo. Ele consiste de um divisor programável cuja frequência de entrada é 18,432 MHz, ver figura 38. Esta frequência é inicialmente dividida por 15, fornecendo 1,228 MHz, que também alimenta o relógio de 1 ms discutido acima. O divisor programável realiza divisões inteiras conforme o código que é colocado nas suas linhas de programação (V0 a V7). Os códigos de programação das diversas velocidades são apresentados na tabela 4.

CÓDIGO OCTAL	DIVISÃO	FREQUÊNCIA (X64)
376	2	9600
375	4	4800
373	8	2400
367	16	1200
357	32	600
337	64	300
277	128	150
247	176	110
177	256	75
077	384	50

Tabela 4 Código das velocidades de transmissão

Esta frequência poderá passar por outro divisor fixo de 64 ou não. Caso a transmissão seja assíncrona, o relógio deverá ser 64 vezes maior que a taxa de transmissão. Caso a transmissão seja síncrona, o relógio deve manter a mesma frequência da taxa de transmissão.

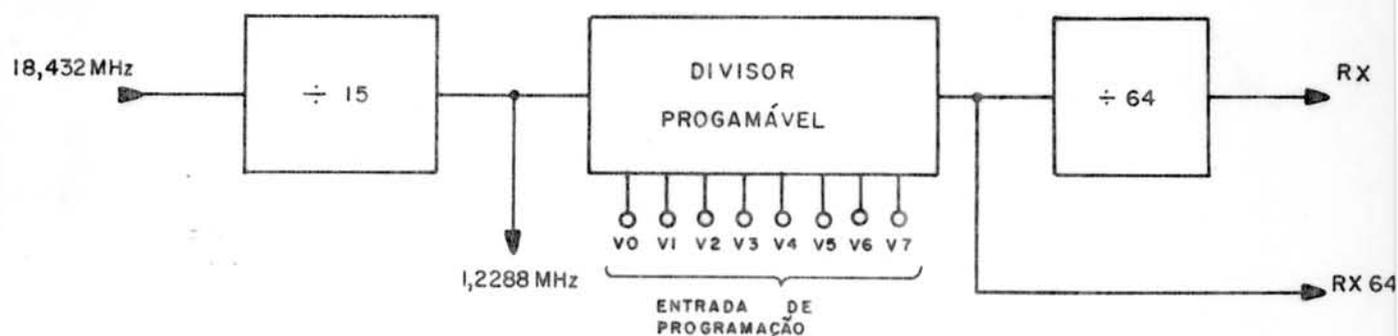


Figura 38 Relógio programável

A programação deste relógio é feita pelo programa que carrega no registrador RVeloc=120 o código correspondente à taxa desejada. Apresentamos abaixo um exemplo de programação do relógio programável.

```

:
:
LAI 367 * programa relógio para
OUT 120 * taxa de 1200 bps
:
:

```

#### 4.6.3 Canais de comunicação

Para que o STTP possa manter comunicação com o interface V.24, foi necessário usar dois canais de comunicação: o canal de comunicação 1 e o canal de comunicação 2. Estes canais recebem dados transmitidos serialmente pelo interface V.24 e os transformam em caracteres de até 8 bits em paralelo para a UCP. Na transmissão eles recebem os dados em paralelo da UCP e os transmitem serialmente para o interface V.24.

A razão de se usar dois canais adveio da necessidade de monitorar uma linha em "full-duplex". Neste caso, a estação monitorada transmite e recebe dados ao mesmo tempo. O STTP deve receber tanto os dados transmitidos como os dados recebidos por ela. Assim, usamos o canal de comunicação 1 para monitorar os dados recebidos pela estação local, enquanto o canal de comunicação 2 monitora os dados transmitidos pela estação local. No caso de uma simulação de ETD, apenas o canal 1 é utilizado, enquanto que na simulação de ECD, apenas o canal 2 é utilizado. Deste modo, as coisas se simplificaram para o usuário, já que o canal 1 (conector J1) sempre é ligado a um ECD (modem), enquanto que o canal 2 (conector J2) sempre é ligado a um ETD (terminal ou computador).

Os dois canais funcionam de maneira idêntica, sendo ambos implementador com a usart (receptor e transmissor síncrono assíncrono universal) 8251. Estas usart's são totalmente programáveis, permitindo ao sistema uma flexibilidade muito grande. Cada usart possui 2 registradores, um de comando e

outro de dado. O registrador de comando, na leitura, passa a ser um registrador de estado da usart. Na usart do canal 1 estes registradores apresentam os seguintes endereços:

Registrador de dados (leitura e escrita)	- endereço 160
Registrador de comando (escrita)	- endereço 161
Registrador de estado (leitura)	- endereço 161

A usart do canal 2 apresenta seus registradores com os seguintes endereços:

Registrador de dados (leitura e escrita)	- endereço 200
Registrador de comando (escrita)	- endereço 201
Registrador de estado (leitura)	- endereço 201

#### Programação das usart's -

A programação das usart's está descrita pelo fluxograma apresentado na figura 39. Após receber o sinal RESET, a usart espera por uma instrução de modo. Se a instrução de modo indicar uma transmissão síncrona, devemos enviar um ou dois caracteres de sincronismo. Para transmissões assíncronas estes caracteres não são enviados à usart. Após, devemos enviar a instrução de comando e só então a usart se encontra pronta a receber ou transmitir dados.

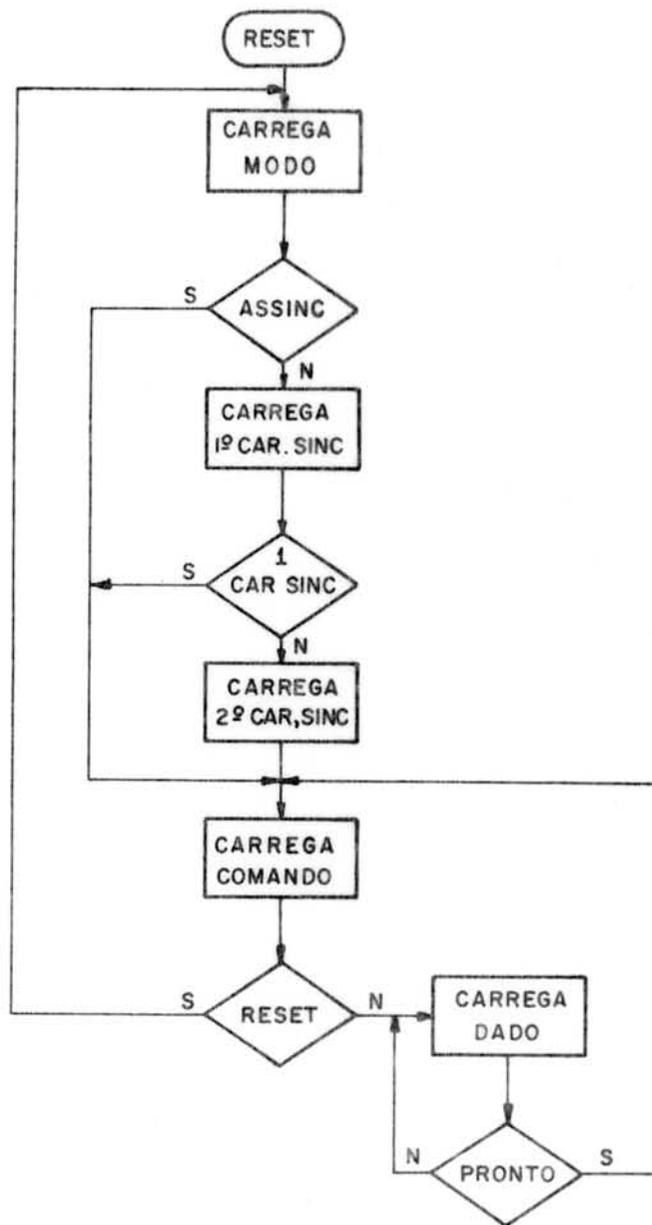


Figura 39 Fluxograma de programação da usart

- Formato da instrução de modo (assíncrono)

7	6	5	4	3	2	1	0
S2	S1	EP	PEN	L2	L1	B2	B1

B2,B1 - Baud rate factor

0 , 0 - (modo síncrono)

1 , 1 - (1X)

1 , 0 - (16X)

1 , 1 - (64X)

L2,L1 - Tamanho do caracter

0 , 0 - 5 bits

0 , 1 - 6 bits

1 , 0 - 7 bits

1 , 1 - 8 bits

PEN - habilita paridade

0 - desabilita

1 - habilita

EP - Paridade

0 - ímpar

1 - par

S2,S1 - Número de Stop bits

0 , 1 - inválido

0 , 1 - 1 stop bit

1 , 0 - 1,5 stop bits

1 , 1 - 2 stop bits

- Formato da instrução de modo (síncrono)

7	6	5	4	3	2	1	0
SCS	ESD	EP	PEN	L2	L1	0	0

L2,L1 - Tamanho do caracter

0 , 0 - 5 bits

0 , 1 - 6 bits

1 , 0 - 7 bits

1 , 1 - 8 bits

PEN - Habilita paridade

0 - desabilita

1 - habilita

EP - Paridade

0 - ímpar

1 - par

ESD - Detector de sincronismo externo

0 - syndet é saída

1 - syndet é entrada

SCS - Caracter de sincronismo simples

0 - caracter de sincronismo duplo

1 - caracter de sincronismo simples

## - Formato da instrução de comando

7	6	5	4	3	2	1	0
EH	IR	RTS	ER	SBK	RXE	DTR	TXE

TXE - Transmissor habilitado  
 0 - desabilitado  
 1 - habilitado

DTR - Terminal pronto  
 0 - não pronto  
 1 - pronto

RXE - Receptor habilitado  
 0 - não habilitado  
 1 - habilitado

SBK - Envia caracter "break"  
 0 - operação normal  
 1 - envia caracter "break"

ER - Reseta erros  
 1 - reseta flags de erro: PE, OE, FE

RTS - Pedido para transmitir  
 1 - envia pedido para transmitir

IR - Reset interno  
 1 - coloca usart à espera da instrução de formato

EH - Entra no modo "hunt"  
 1 - habilita procura do caracter de sincronismo

- Formato do registrador de estado

7	6	5	4	3	2	1	0
DSR	SYNDE	FE	OE	PE	TXEMP	RXRDY	TXRDY

TXRDY - Transmissor pronto

RXRDY - Receptor pronto

TXEMP - Transmissor vazio

PE - Erro de paridade

OE - Erro de "overrun"

ocorre quando um caracter não é lido a tempo pela UCP

FE - Erro de "framing"

ocorre quando não é detectado o bit de stop numa recepção assíncrona

SYND - Detectado caracter(es) de sincronismo

DSR - Modem pronto

- Exemplo de programação da usart

```

:
LAI 100
OUT 161 * Reseta usart 1
:
LAI 113 * Envia instrução de modo
OUT 161 * 64X,7 bits, s/parid., 1 stop bit
LAI 26 * Envia instrução de comando
OUT 161 * DTR, RXE, ER: entra em recepção
:
LAI 63 * Envia instrução de comando
OUT 161 * TXE, DTR, ER: entra em transmissão
:
LDMI DADO
OUT 160 * Envia dado
:

```

#### 4.6.4 Registrador de comando do interface

Este registrador é responsável pelo controle global do interface de comunicações. O seu endereço é CCOM=260. Apresentamos abaixo a configuração deste registrador.

7	6	5	4	3	2	1	0
TST	TLF	TLI	MT	SC	H2	H1	REL

REL - Relógio  
 0 - externo  
 1 - interno

H2,H1 - Operação  
 0 , 0 - monitora  
 0 , 1 - simula ECD  
 1 , 0 - simula ETD  
 1 , 1 - não permitido

SC - Selecciona canal  
 0 - canal secundário  
 1 - canal principal

MT - Modo de transmissão  
 0 - síncrono  
 1 - assíncrono

TLI - Transição linha início  
 0 - 0->1  
 1 - 1->0

TLF - Transição linha fim  
 0 - 0->1  
 1 - 1->0

TST - Modo teste  
 0 - outros modos  
 1 - modo teste

## 4.6.5 Método de ligação ao interface V.24

Na figura 40 é mostrada a maneira como o STTP se liga ao interface V.24. Primeiramente existem receptores e transmissores que ajustam os níveis dos sinais internos aos níveis dos sinais do interface.

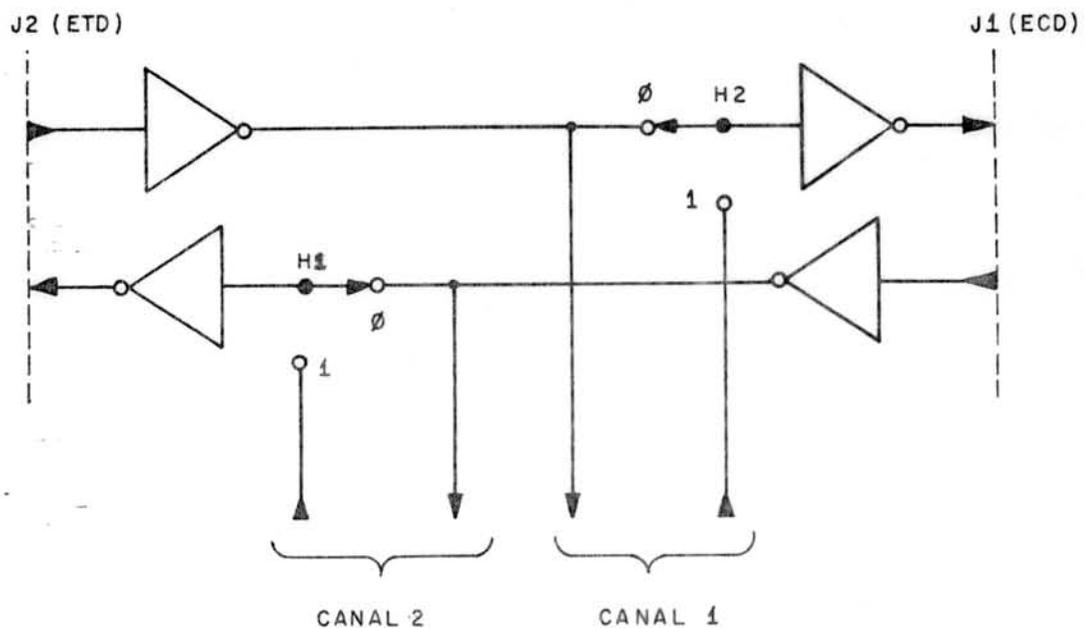


Figura 40 Método de ligação ao interface V.24

As chaves  $H_1$  e  $H_2$ , comandadas pelo registrador de controle do interface, comandam o fluxo de dados. No caso de uma monitoração, ambas deverão estar na posição "0". Desta maneira, os dados passam através do sistema sem sofrer interferência, sendo apenas captados pelo STTP. No caso de uma simulação, uma das chaves estará na posição "1", permitindo assim o diâ-

logo entre o STTP e o interface V.24. Na tabela 5, apresentamos as diversas posições em que podem se encontrar as chaves.

H1,H2	FUNÇÃO
0 0	- Modo monitor
0 1	- Simula ECD
1 0	- Simula ETD
1 1	- Não permitido

Tabela 5 Posicionamento das chaves H1 e H2

No modo teste devemos ter H1=1 e H2=0, ou seja, a mesma configuração de SIMULA ETD, já que devemos nos conectar com um modem.

#### 4.6.6 Medição dos intervalos de tempo

O STTP realiza medições de intervalo de tempo entre as transições de duas linhas quaisquer do interface V.24. Para tanto existem dois circuitos seletores de 16 para 1. Uma entrada de programação, de 4 bits, seleciona uma das 16 linhas. Estas são selecionadas entre as linhas do interface V.24 que realizam transições. Na figura 41, apresentamos o circuito seletor da linha de início do intervalo de tempo. A linha de fim tem um circuito idêntico. A saída do circuito seletor passa por um inversor programável controlado pelo sinal TLI (ver 4.6.4), que inverte ou não a linha selecionada para ge-

rar início. Uma vez ocorrida a transição desejada da linha selecionada, é ligado o flip-flop início. Este flip-flop interrompe o processador indicando o início do intervalo de tempo. O programa, então, zera o contador de intervalos de tempo e habilita o relógio de milisegundos a incrementar este contador a cada ciclo de 1 ms. O mesmo processo é utilizado para gerar o sinal de fim de intervalo de tempo. Quando este sinal interromper o processador, o programa armazena o resultado do contador e fica à espera de um novo início.

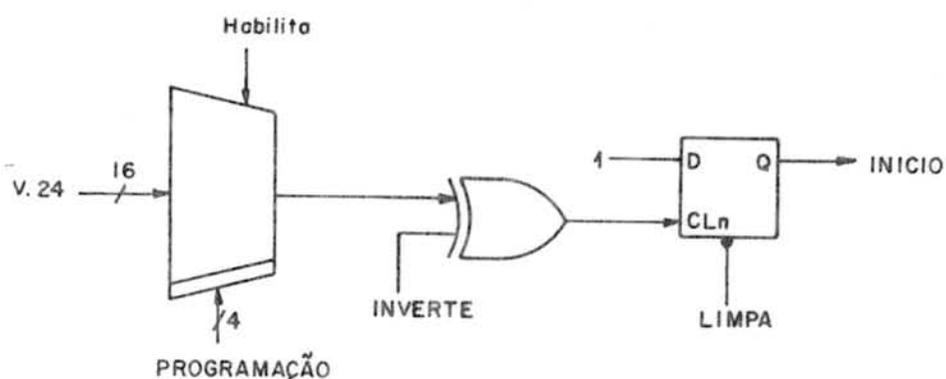
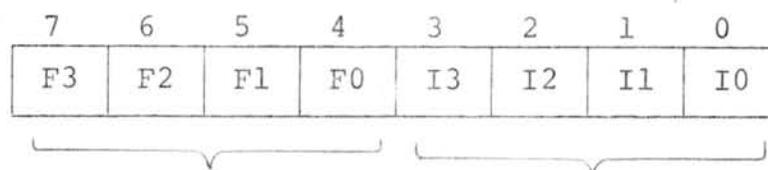


Figura 41 Seletor de linha de início

Para programar os seletores de início e fim, existe um registrador cujo endereço é SLIF=240. Apresentamos abaixo o formato deste registrador.

Registrador de linhas início e fim (SLIF=240)



Seleciona linha fim      Seleciona linha início

#### 4.7 Interface da impressora

Este interface recebe os dados paralelos enviados pelo processador central e os envia serialmente à impressora, através do conector J3 (IMP) existente no painel traseiro do STTP.

Para realizar esta tarefa foi utilizado unicamente uma usart. A operação da usart já foi detalhada durante a descrição do interface de comunicações, dispensando aqui maiores comentários. Os endereços deste interface são:

Endereço de dados (saída)	US3D=340
Endereço de controle (entrada e saída)	US3C=341

A operação deste interface é a única que não se realiza através de interrupção. Durante uma transferência de dados para a impressora, o processador não realiza outra tarefa que não seja pertinente ao processo. No entanto, ele sempre poderá ser interrompido pelo teclado.

A operação de impressão consiste em enviar atra-

vés deste interface todo o conteúdo da memória monitora, a partir do seu endereço inicial. As duas opções existentes possibilitam enviar os dados codificados em ASCII ou hexa.

#### 4.8 Painel de comando

O painel do STTP é constituído de um mostrador de 32 caracteres alfanuméricos, através do qual o operador coordena a programação do equipamento e visualiza os dados armazenados na memória monitora; um teclado hexadecimal para entrada de dados codificados em ASCII; um conjunto de teclas especiais para entrada de comandos e um conjunto de 25 diodos emissores de luz para indicar o estado das linhas do interface V.24. Na figura 42, apresentamos uma visão do painel de comando.



#### 4.8.1 Teclado hexadecimal

Por questão de espaço físico, desejava-se um equipamento o mais portátil possível, foi escolhido um teclado hexadecimal para entrada de dados. Os dados aceitos pelo sistema através deste teclado são codificados em ASCII. Desta maneira, devemos entrar com dois caracteres hexadecimais de cada vez. Porém, algumas vezes, o operador é solicitado a preencher um campo apenas com números (campo usuário numérico). Neste caso, basta entrar com a tecla correspondente ao número decimal desejado de 0 a 9.

#### 4.8.2 Diodos emissores de luz

Para indicar o estado das 25 linhas do interface V.24, foram utilizados 25 diodos emissores de luz. Estes diodos acendem quando a respectiva linha estiver na condição de "espaço" (positivo), e se apagam quando a linha estiver na condição de "marca" (negativo).

Algumas linhas do interface V.24 não são utilizadas, como é o caso das linhas 9, 10, 11, 18 e 25. Outras duas linhas, 1 e 7, são terras. Neste caso, os diodos correspondentes a estas linhas não devem nunca acender.

#### 4.8.3 Mostrador

O mostrador é composto de 32 matrizes 5x7 de pontos luminosos (TIL501). Cada matriz é capaz de representar qualquer caracter codificado em ASCII. Além destes, elas também podem representar caracteres especiais, como é o caso dos caracteres de controle. Cada célula possui um ponto decimal que é utilizado para diferenciar dados recebidos de dados transmitidos por uma estação local. Os transmitidos aparecem com o ponto decimal aceso. O cursor para este tipo de mostrador, acende alternadamente todos os pontos da matriz e o conteúdo da mesma.

#### 4.8.4 Conjunto de teclas especiais

Um conjunto de 15 teclas especiais é utilizado para comandar e programar o STTP. A definição funcional destas teclas foi apresentada em 3.6. Destas teclas, apenas uma não é tratada pelo interface do teclado. É a tecla vermelha do painel, início, que se liga diretamente ao processador. Esta tecla inicializa o processador levando-o a iniciar a execução a partir do endereço 0 da memória. Neste endereço está o início do programa residente. Esta tecla não age por interrupção, não havendo nada que possa impedir a sua atuação. Ela é utilizada caso o sistema, por algum motivo, entre em "dead-lock", isto é, uma situação sem saída.

#### 4.9 Painel traseiro

Na figura 43, apresentamos uma visão do painel traseiro. Neste painel estão localizados os três conectores que ligam o STTP com o mundo exterior.

O conector ECD (J1), do tipo DB-25P, de 25 pinos, padrão que liga o STTP ao equipamento de mesmo nome, ECD (modem).

O conector ETD (J2), do tipo DB-25S, de 25 pinos, padrão, que liga o STTP ao equipamento de mesmo nome, ETD (terminal ou computador).

O conector IMP (J3), do tipo DB-25P, que liga o STTP a uma impressora do tipo EXTEL, através do qual podemos imprimir o conteúdo da memória monitora.

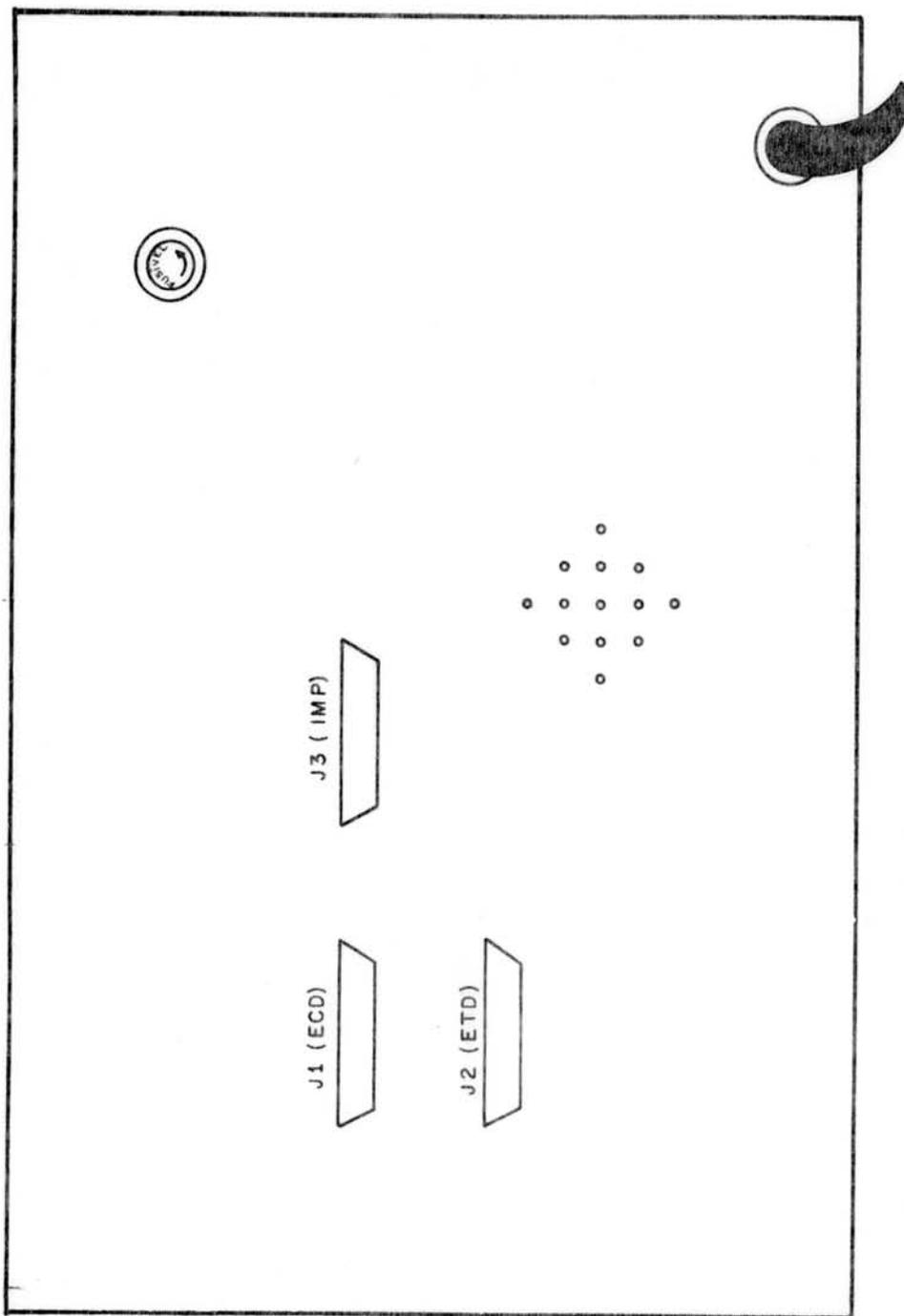


Figura 43 Painel traseiro

5

ORGANIZAÇÃO DO PROGRAMA RESIDENTE

Apresenta a organização do programa residente juntamente com as suas rotinas principais.

## 5.1 Introdução

O programa residente do sistema de teste de redes de teleprocessamento apresenta dois estados distintos: o ESTADO DE CONTROLE e o ESTADO DE EXECUÇÃO, ver figura 44.

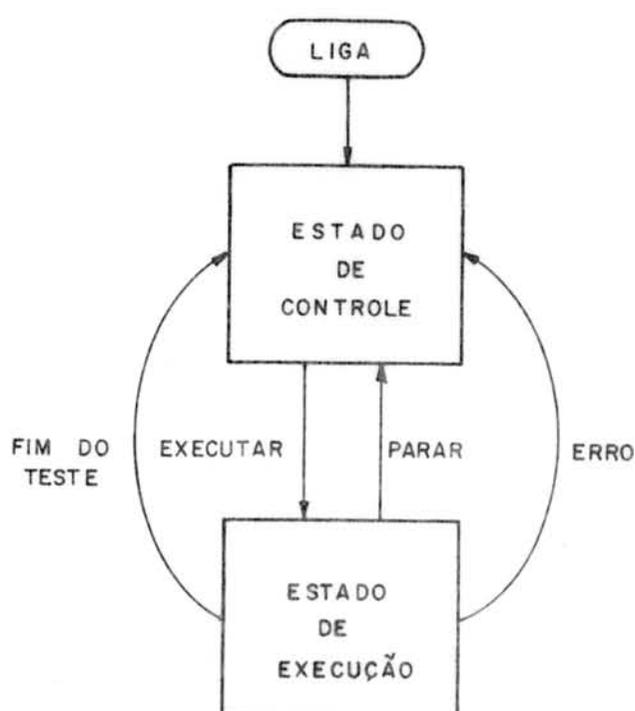


Figura 44 Estados principais do sistema

Ao ser ligado o sistema, ele assume imediatamente o ESTADO DE CONTROLE. A saída deste estado só pode conduzir ao ESTADO DE EXECUÇÃO ao ser apertada a tecla EXECUTAR. A partir do ESTADO DE EXECUÇÃO, o sistema pode retornar ao ESTADO DE CONTROLE de três maneiras: ao apertarmos a tecla PARAR, por uma saída automática ao término de um teste ou ainda por uma saída por erro, quando entramos no ESTADO DE EXECUÇÃO com algum erro de programação.

As atividades do sistema operam à base de interrupção. Estas interrupções, no entanto, são controladas pelo programa residente. No ESTADO DE CONTROLE, por exemplo, só são aceitas as interrupções vindas do teclado. Todas as demais interrupções são inibidas neste estado. Já no ESTADO DE EXECUÇÃO, todas as interrupções podem atuar, dependendo do tipo de teste que se realiza.

## 5.2 Sistema de interrupção

O STTP funciona à base de interrupção. Para controlar estas interrupções, existe um CONTROLADOR DE INTERRUPTÕES situado na UNIDADE CENTRAL DE PROCESSAMENTO. Este controlador aceita até 8 níveis de interrupção, que foram assim definidos, por ordem de prioridade, na tabela 6.

Nível 7	-	Não utilizado
Nível 6	-	Teclas do painel de comando
Nível 5	-	Canal de comunicações 1
Nível 4	-	Canal de comunicações 2
Nível 3	-	Pedido para transmitir do ETD
Nível 2	-	Início da contagem de tempo
Nível 1	-	Fim da contagem de tempo
Nível 0	-	Relógio de 1 ms

Tabela 6 Níveis de interrupção

#### 5.2.1 NÍVEL 7

Este é o nível de maior prioridade. Este nível não foi utilizado porque leva o sistema para o endereço 000 de inicialização do STTP. A tecla início, ligada diretamente ao processador, também leva o sistema para este endereço. A diferença está que, ao atuar diretamente sobre o processador, não há nada que possa impedir a sua atuação, enquanto que atuando através do nível 7, poderia ser inibida por acaso e o sistema ficaria impedido de se reinicializar.

#### 5.2.2 NÍVEL 6

Após o NÍVEL 7, este é o de maior prioridade. Interrompem neste nível todas as teclas do painel, exceto a te-

cla INÍCIO. Este nível nunca é desabilitado pelo programa residente, sendo sempre atendido. No seu atendimento, existe a tabela de transição que realiza a seleção das teclas que podem ser atendidas, dependendo do estado em que o sistema se encontrar. O endereço de atendimento deste nível é 10.

#### 5.2.3 NÍVEL 5

Neste nível interrompe o CANAL DE COMUNICAÇÕES 1. Este canal está ligado ao conector J1(ECD) do painel traseiro que liga o STTP ao modem. Este canal interrompe, quando habilitado, sempre que um caracter for recebido do ECD ou transferido ao ECD. O endereço de atendimento deste nível é 20.

#### 5.2.4 NÍVEL 4

Neste nível interrompe o CANAL DE COMUNICAÇÕES 2. Este canal está ligado ao conector J2(ETD) do painel traseiro e liga o STTP a um EQUIPAMENTO DE TRANSMISSÃO DE DADO (ETD). Este canal interrompe sempre que um caracter for transmitido ou recebido através do respectivo interface. O endereço deste nível é 30.

#### 5.2.5 NÍVEL 3

Neste nível interrompe o sinal RTS (pedido para transmitir), vindo do ETD, quando o sistema se encontra no modo SIMULA ECD. O endereço de atendimento deste nível é 40.

#### 5.2.6 NÍVEL 2

Neste nível interrompe a linha INÍCIO DE INTERVALO DE TEMPO. Esta linha, definida pelo operador, marca o início da contagem de tempo entre duas linhas do interface V.24. O endereço de atendimento deste nível é 50.

#### 5.2.7 NÍVEL 1

Neste nível interrompe a linha FIM DE INTERVALO DE TEMPO. Esta linha, definida pelo operador, marca o fim da contagem de tempo entre duas linhas do interface V.24. O endereço de atendimento deste nível é 60.

#### 5.2.8 NÍVEL 0

Neste nível interrompe o relógio de 1 milissegundo. Este relógio executa todas as funções de temporização do sistema. É com ele que o sistema mede o intervalo de tempo em

#### 5.2.5 NÍVEL 3

Neste nível interrompe o sinal RTS (pedido para transmitir), vindo do ETD, quando o sistema se encontra no modo SIMULA ECD. O endereço de atendimento deste nível é 40.

#### 5.2.6 NÍVEL 2

Neste nível interrompe a linha INÍCIO DE INTERVALO DE TEMPO. Esta linha, definida pelo operador, marca o início da contagem de tempo entre duas linhas do interface V.24. O endereço de atendimento deste nível é 50.

#### 5.2.7 NÍVEL 1

Neste nível interrompe a linha FIM DE INTERVALO DE TEMPO. Esta linha, definida pelo operador, marca o fim da contagem de tempo entre duas linhas do interface V.24. O endereço de atendimento deste nível é 60.

#### 5.2.8 NÍVEL 0

Neste nível interrompe o relógio de 1 milissegundo. Este relógio executa todas as funções de temporização do sistema. É com ele que o sistema mede o intervalo de tempo em

tre as transições de duas linhas do interface V.24 e simula o atraso RTS/CTS. Ele serve ainda para gerar o tempo de espera entre mensagens em uma simulação. O endereço de atendimento deste nível é de 70.

### 5.3 Estado de controle

Neste estado o sistema aceita as teclas do painel de comando e o usuário pode configurar o sistema para operar conforme suas especificações. O ESTADO DE CONTROLE pode ser descrito pelo diagrama de estados apresentado na figura 45. Segundo este diagrama, as teclas constituem as entradas que levam o sistema a executar uma função e ir para outro estado. O sistema assim descrito divide as teclas em duas classes. A primeira, formada pelas teclas ditas fundamentais, que são: FORMATO, MODO, EDITAR, LISTAR e EXECUTAR. Estas teclas são sempre aceitas em qualquer estado exceto no ESTADO DE EXECUÇÃO. A segunda classe é formada pelas demais teclas, chamadas de teclas auxiliares e que são aceitas somente em determinados estados.

A seguir, passamos a descrever os estados dentro de CONTROLE.

#### 5.3.1 Estado "0"

Este é o chamado estado inicial ou geral. Este

estado aceita apenas as cinco teclas fundamentais: FORMATO, MODO, EDITAR, LISTAR e EXECUTAR. Este estado é atingido pelas teclas INÍCIO, na inicialização do sistema, e PARAR, quando realizamos uma parada manual do teste.

### 5.3.2 Estado "1"

Este é também chamado de estado FORMATO, pois ele é alcançado ao ser acionada a tecla FORMATO. Este estado permite a programação dos parâmetros de FORMATO. Para esta programação, necessita-se apenas das teclas PARÂMETRO, OPÇÃO, ESQUERDA, DIREITA e HEXA. As demais teclas, com exceção das cinco fundamentais, não são atendidas neste estado.

### 5.3.3 Estado "2"

Este estado é atingido através da tecla MODO. Ele permite a programação dos três modos de operação do STTP: monitor, simulação e teste. Neste estado, são habilitadas, além das cinco fundamentais, as seguintes teclas: HEXA, ESQUERDA, DIREITA, QUALQUER e NÃO.



#### 5.3.4 Estado "3"

Este estado é atingido através da tecla EDITAR. Ele permite a execução das diretivas de edição. Neste estado é habilitado apenas o teclado HEXA, já que é esperado um dígito hexa para completar a diretiva de edição.

#### 5.3.5 Estado "4"

Este estado é atingido, a partir do estado "3", pela tecla HEXA=0. Este estado configura a diretiva EDITAR,0 que permite a edição do programa de simulação. Neste estado são habilitadas as teclas: HEXA, ESQUERDA, DIREITA e ENTRAR.

#### 5.3.6 Estado "5"

Este estado é atingido, a partir do estado "3", pela tecla HEXA=1. Este estado configura a diretiva EDITAR,1 que permite a edição das mensagens de recepção. Neste estado são habilitadas as teclas: HEXA, ESQUERDA, DIREITA, QUALQUER, NÃO e ENTRAR.

#### 5.3.7 Estado "6"

Este estado é também atingido a partir do estado

"3", pela tecla HEXA=2. Este estado configura a diretiva EDITAR,2, que permite a edição das mensagens de transmissão. Neste estado são habilitadas as teclas: HEXA, ESQUERDA, DIREITA, QUALQUER e ENTRAR.

#### 5.3.8 Estado "7"

Este estado é atingido a partir do estado "3", pela tecla HEXA=3. Este estado configura a diretiva EDITAR,3 que permite a edição da mensagem de transmissão de 256 caracteres. Neste estado estão habilitadas as teclas: HEXA, ESQUERDA, DIREITA e QUALQUER.

#### 5.3.9 Estado "8"

Este estado é atingido através da tecla LISTAR. Ele prepara as diretivas de listagem. Só são aceitas, neste estado, as teclas HEXA, que devem completar de maneira adequada a diretiva LISTAR.

#### 5.3.10 Estado "9"

Este estado é atingido a partir do estado "8", ou seja, é o estado que configura as diretivas LISTAR,0, LISTAR,1 e LISTAR,2. Estas diretivas listam a memória no mostrador. As-

sim que, para auxiliar na visualização desta memória monitora, devemos dispor das teclas DIREITA e ESQUERDA, que são as únicas habilitadas neste estado.

#### 5.3.11 Estado "10"

Este estado é atingido através do acionamento da tecla EXECUTAR. Este é o chamado ESTADO DE EXECUÇÃO, descrito em detalhe mais adiante.

#### 5.4 Tabela de transição

Para implementar a função descrita pelo diagrama de estados apresentado acima, foi construída uma tabela de transição. Esta tabela apresenta como colunas as teclas de entrada e como linhas os 11 estados em que o sistema pode se encontrar. A célula desta tabela é constituída de duas partes: uma indicação da função a ser executada e o próximo estado a ser atingido. As teclas de entrada desta tabela apresentam as seguintes abreviações:

HEX	-	Qualquer tecla hexadecimal
ESQ	-	ESQUERDA
DIR	-	DIREITA
QUA	-	QUALQUER
NÃO	-	NÃO
ENT	-	ENTRAR

LIS - LISTAR  
 EXE - EXECUTAR  
 PRM - PARÂMETRO  
 OPÇ - OPÇÃO  
 FOR - FORMATO  
 MOD - MODO  
 EDI - EDITAR  
 PAR - PARAR

	HEX	ESQ	DIR	QUA	NÃO	ENT	LIS	EXE	PRM	OPÇ	FOR	MOD	EDI	PAR
0	RET	RET	RET	RET	RET	RET	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	0	0	0	0	0	0	8	10	0	0	1	2	3	0
1	H12	E/C	D/C	RET	RET	RET	LIS	EXE	PRM	OPÇ	FOR	MOD	EDI	RET
	1	1	1	1	1	1	8	10	1	1	1	2	3	1
2	H12	E/C	D/C	QUA	NÃO	RET	LIS	EXE	PRM	OPÇ	FOR	MOD	EDI	RET
	2	2	2	2	2	2	8	10	2	2	1	2	3	2
3	HED	RET	RET	RET	RET	RET	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	0	0	0	0	0	0	8	10	0	0	1	2	3	0
4	HX1	E/C	D/C	RET	RET	EN0	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	4	4	4	4	4	4	8	10	4	4	1	2	3	4
5	HX2	E/C	D/C	QUA	NÃO	EN1	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	5	5	5	5	5	5	8	10	5	5	1	2	3	5
6	HX2	E/C	D/C	QUA	RET	EN2	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	6	6	6	6	6	6	8	10	6	6	1	2	3	6
7	HX2	E/C	D/C	QUA	RET	EN2	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	7	7	7	7	7	7	8	10	7	7	1	2	3	7
8	HLI	RET	RET	RET	RET	RET	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	0	0	0	0	0	0	8	10	0	0	1	2	3	0
9	RET	E/D	D/D	RET	RET	RET	LIS	EXE	RET	RET	FOR	MOD	EDI	RET
	9	9	9	9	9	9	8	10	9	9	1	2	3	9
10	RET	PAR												
	10	10	10	10	10	10	10	10	10	10	10	10	10	0

Tabela 7 Tabela de transição

#### 5.4.1 Funções da tabela de transição

Apresentamos abaixo uma descrição das funções executadas pela tabela de transição. Estas funções foram implementadas por rotinas do mesmo nome que se encontram, por ordem alfabética, na listagem do programa residente sob o título de ROTINAS DO ESTADO DE CONTROLE.

RET - RETORNA. Não executa função alguma. O sistema simplesmente retorna ao estado indicado. Esta função desabilita a tecla no estado respectivo.

D/C - DIR/CURSOR. Desloca cursor para a direita dentro do campo usuário.

D/D - DIR/DADO. Desloca dados da tela para a direita.

DCD - DIR/CD. Desloca cursor ou dados para a direita. Se o cursor atingir o fim da tela - 1 posição, desloca os dados para a esquerda. Esta função é utilizada para editar a mensagem de transmissão A4.

EDI - EDITAR. Leva o sistema ao estado 3 como preparação para receber uma diretiva de edição. Esta função é executada ao ser pressionada a tecla EDITAR.

ENO - ENTRA0. Armazena a linha do programa de simulação presente na tela e mostra a próxima linha para edição ou

conferência.

- EN1 - ENTRA1. Armazena a mensagem de recepção presente na tela e mostra a próxima mensagem de recepção, para edição ou conferência.
- EN2 - ENTRA2. Armazena a mensagem de transmissão presente na tela e mostra a próxima mensagem de transmissão, para edição ou conferência.
- E/C - ESQ/CURSOR. Desloca cursor para a esquerda dentro do campo usuário.
- E/D - ESQ/DADO. Desloca dados na tela para a esquerda.
- ECD - ESQ/CD. Desloca cursor ou dados para a esquerda. Se o cursor atingir a posição 4 da tela, desloca os dados para a direita até alcançar o início da memória de transmissão.
- FOR - FORMATO. Prepara a programação de FORMATO. Mostra na tela o primeiro parâmetro de FORMATO.
- HED - HEDITA. Analisa a tecla digitada durante o estado 3, verificando a correção da diretiva de edição entrada e desviando para o estado que atende a edição correspondente.

- HX1 - HEXA1. Coloca caracter hexa recebido na posição do cursor.
- HX2 - HEXA2. Coloca caracter alfanumérico (ASCII) na posição do cursor. Este caracter é formado por dois dígitos hexa.
- H12 - HEXA12. Se o campo usuário atual for numérico, executa HEXA1, se for alfanumérico, executa HEXA2.
- HLI - HLISTA. Recebe dígitos hexa para completar a diretiva LISTAR.
- LIS - LISTAR. Prepara as diretivas LISTAR.
- MOD - MODO. Prepara a programação de MODO. Mostra na tela o modo atual.
- NÃO - NÃO. Faz com que o próximo caracter alfanumérico entrado pelo operador seja do tipo NÃO.
- OPÇ - OPÇÃO. Mostra a próxima opção do parâmetro mostrado na tela.
- PRM - PARÂMETRO. Mostra o próximo parâmetro.
- PAR - PARAR. Faz o sistema retornar do ESTADO EXECUÇÃO.

QUA - QUALQUER. Coloca o caracter QUALQUER na posição do cursor.

QCD - QUALCD. Coloca o caracter qualquer na posição do cursor e avança cursor ou dados, durante a edição do bloco de mensagem A4.

#### 5.5 ESTADO DE EXECUÇÃO

Neste estado, o sistema aceita todos os níveis de interrupção. No entanto, das interrupções de nível 6, no teclado, só é aceita a tecla PARAR, que faz o sistema voltar ao ESTADO DE CONTROLE. As demais teclas são inibidas neste estado. Ver estado 10 da TABELA DE TRANSIÇÃO.

No MODO TESTE, este estado aceita apenas as interrupções dos níveis 5, 6 e 7, inibindo as demais. Os níveis 6 e 7 não são nunca inibidos. O nível 5 atende as interrupções do CANAL DE COMUNICAÇÕES 1 e se constitui na única fonte de interrupções na execução deste modo.

No MODO MONITOR, são aceitos todos os níveis de interrupção menos o nível 3. Através dos CANAIS DE COMUNICAÇÃO 1 e 2, são recebidos os dados de monitoração, não havendo transmissão alguma. Os níveis 0, 1 e 2 são utilizados para medir os intervalos de tempo entre as linhas do interface V. 24.

No MODO SIMULADOR, existem duas situações distintas. Uma quando simulamos um ECD (modem). Neste caso, a comu-

nicação de dados se faz apenas através do canal 1, ficando o canal 2 desabilitado. As interrupções dos níveis 0, 1 e 2 são atendidas da mesma maneira que no MODO MONITOR. A interrupção de nível 3 (RTS) é habilitada para a simulação do atraso RTS/CTS. A outra situação de simulação ocorre quando simulamos um ETD. Neste caso a comunicação de dados se faz através do canal 2, ficando o canal 1 desabilitado. As interrupções dos níveis 0, 1 e 2 são atendidas da mesma maneira que no MODO MONITOR. A interrupção de nível 3 é desabilitada.

Na figura 46 apresentamos um fluxograma da entrada do ESTADO DE EXECUÇÃO. Após a inicialização, é feita uma verificação de erros. Sendo detectado algum erro, o sistema apresenta a indicação do erro no mostrador e retorna para o ESTADO DE CONTROLE. Não havendo erros, o programa prossegue para desviar para a rotina iniciadora do modo em que o sistema se encontrar.

Na execução de cada um dos três MODOS do sistema, o programa apresenta duas partes distintas:

Iniciador

Continuador

Apresentamos, a seguir, os fluxogramas destas rotinas para cada MODO de operação.

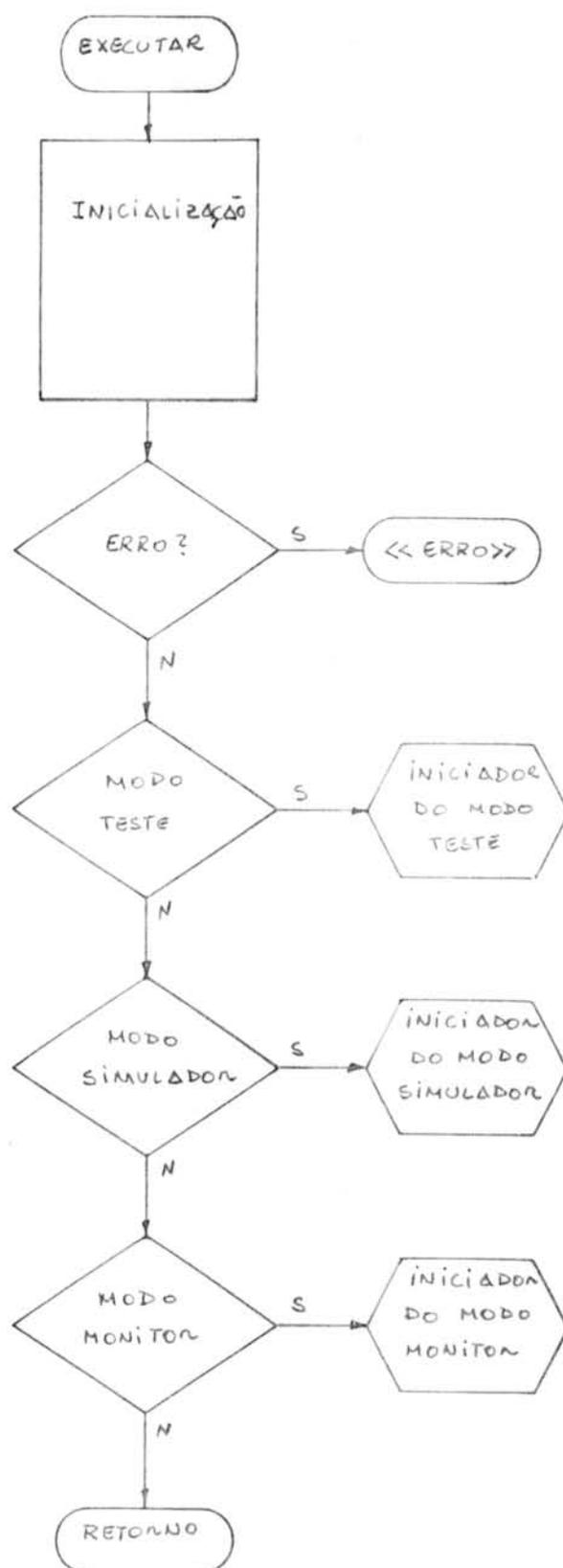


Figura 46 Entrada no ESTADO DE EXECUÇÃO

### 5.5.1 Rotinas do MODO MONITOR

O programa no modo monitor pode ser dividido em duas partes: iniciador e continuador. Na figura 47, apresentamos um fluxograma do iniciador do MODO MONITOR. Este iniciador inicializa os parâmetros, contadores e apontadores do teste e coloca os canais 1 e 2 em recepção.

Após ter iniciado o teste de monitoração, o sistema fica aguardando a ocorrência de interrupções dos canais de comunicação. Estas interrupções desviam o programa para a rotina continuadora do MODO MONITOR. Nas figuras 48, 49 e 50 apresentamos um fluxograma simplificado desta rotina. Ela verifica a ocorrência do EVENTO especificado, verifica o tipo de JANELA selecionado e desvia para um dos três testes do modo: ARMAZENA E CONTA EVENTOS, ARMAZENA APÓS EVENTO e ARMAZENA ATÉ EVENTO. A descrição destes testes já foi vista em 3.3.1.

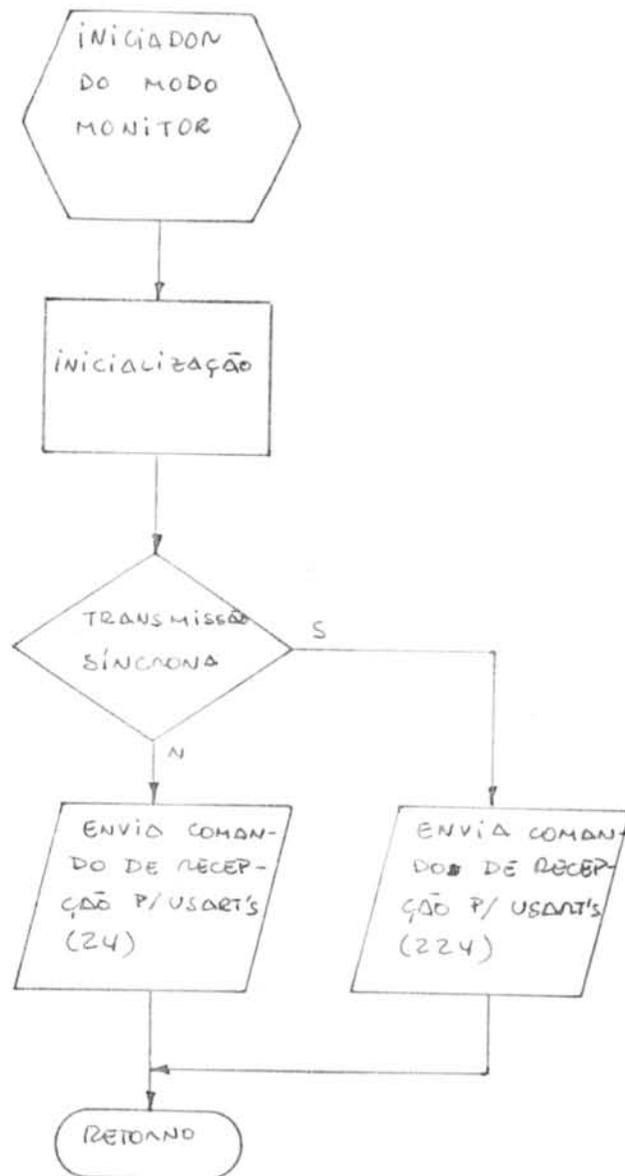


Figura 47 Inicializador do MODO MONITOR

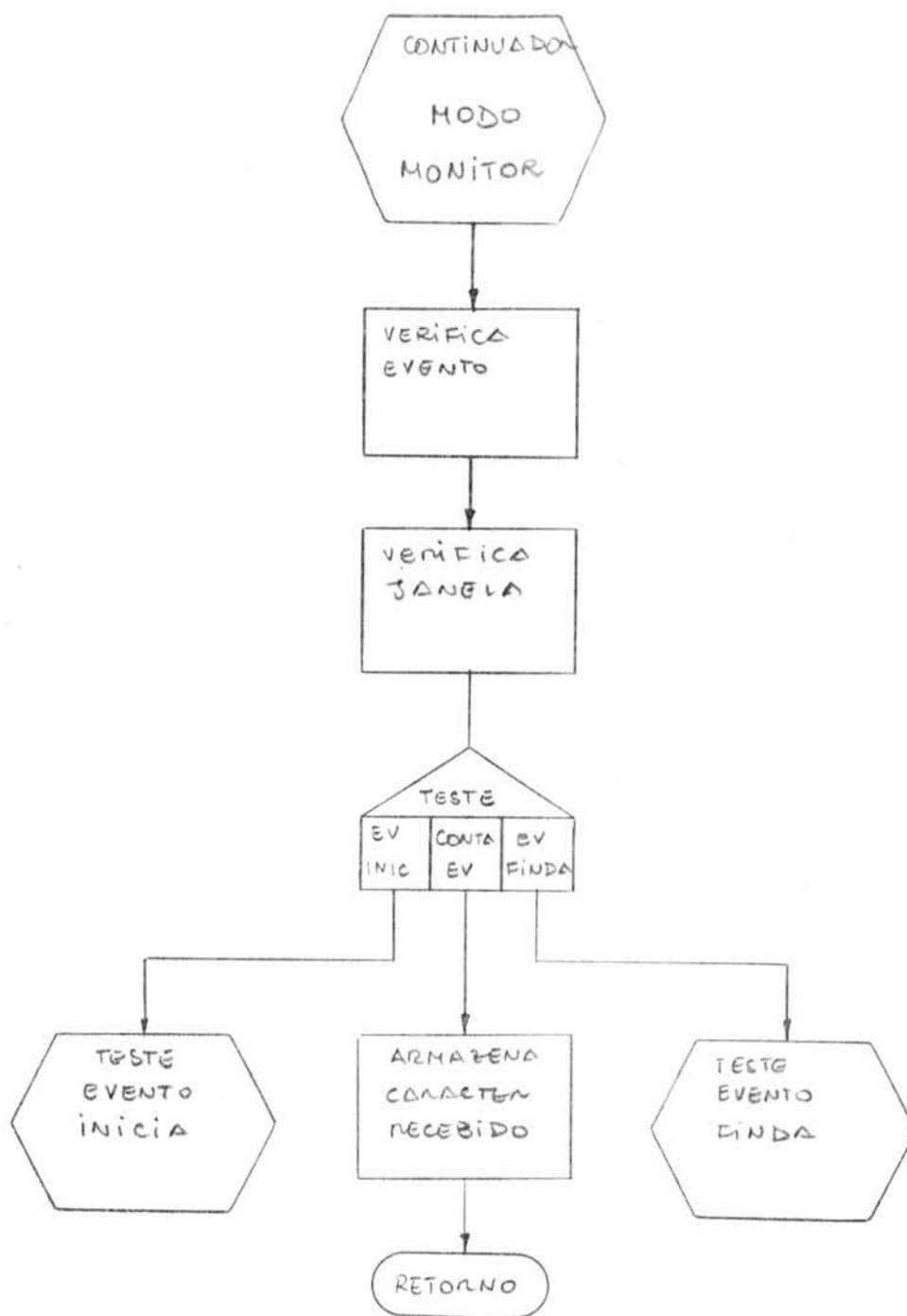


Figura 48 Continuidor do MODO MONITOR (parte 1 de 3)

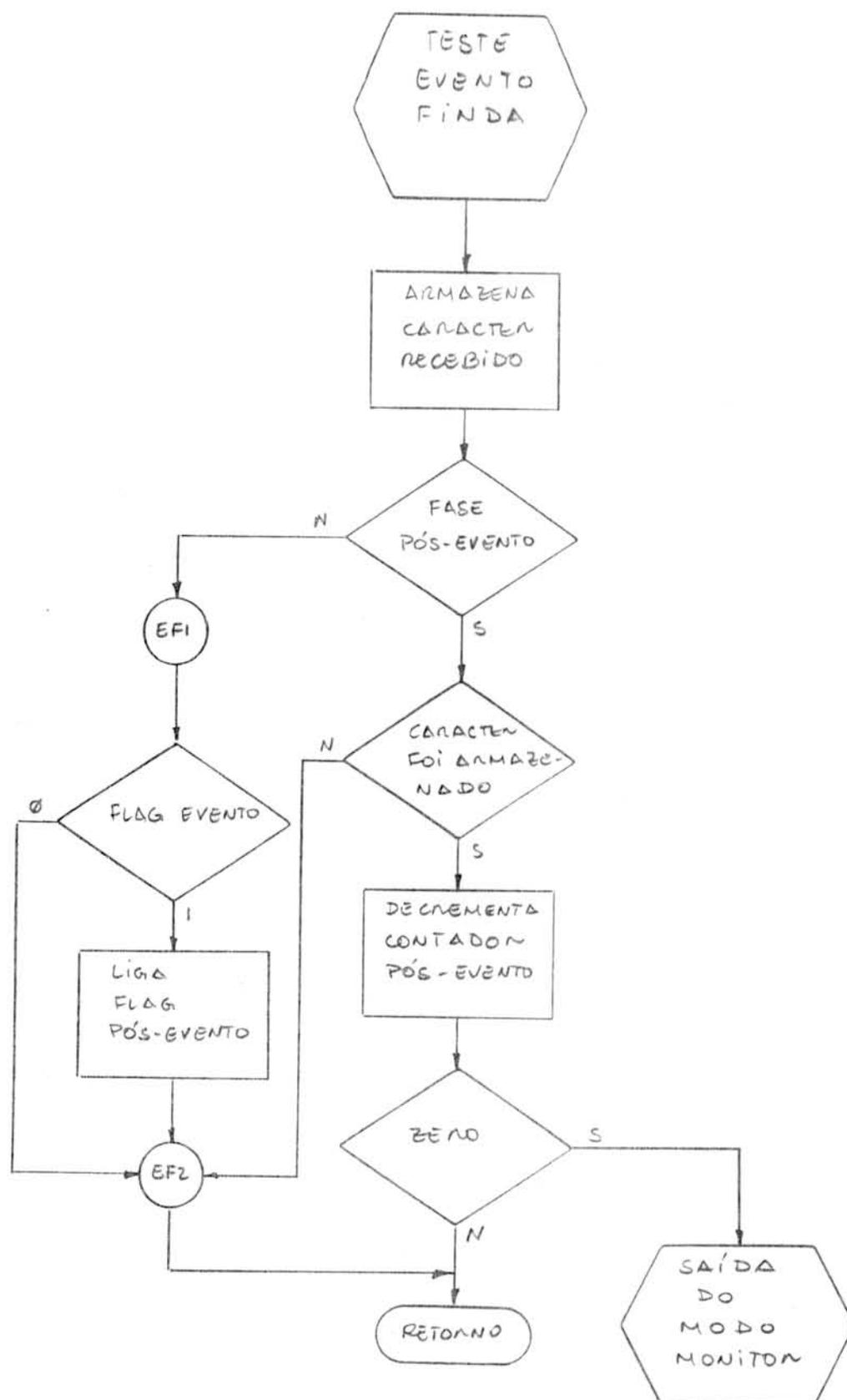


Figura 49 Continuidor do MODO MONITOR (parte 2 de 3)

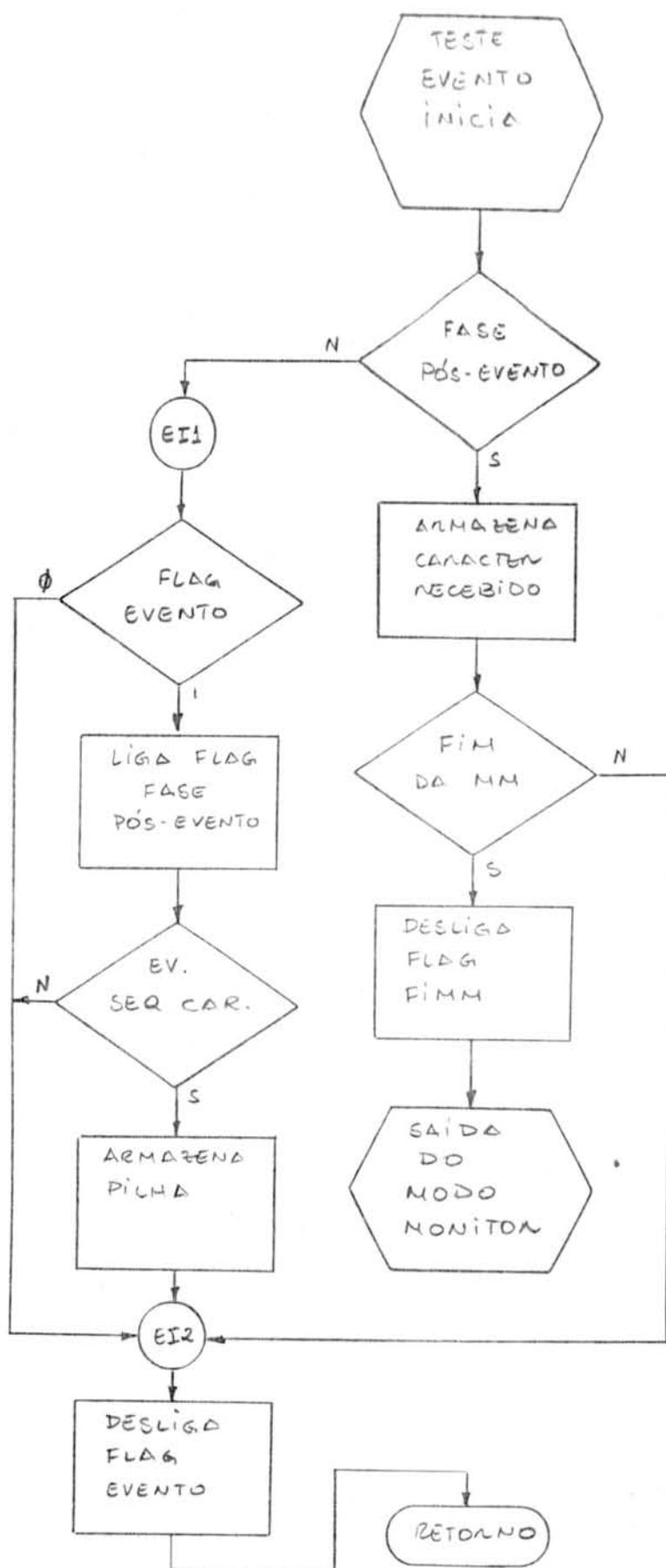


Figura 50 Continuator do MODO MONITOR (parte 3 de 3)

## 5.5.2 Rotinas do MODO SIMULADOR

A exemplo do MODO MONITOR, o MODO SIMULADOR também apresenta uma rotina iniciadora e uma continuadora. Na figura 51, apresentamos um fluxograma simplificado da rotina iniciadora. Após a inicialização, a rotina executa a primeira instrução do programa de simulação. Esta primeira instrução poderá colocar o canal de comunicações correspondente em transmissão ou recepção.

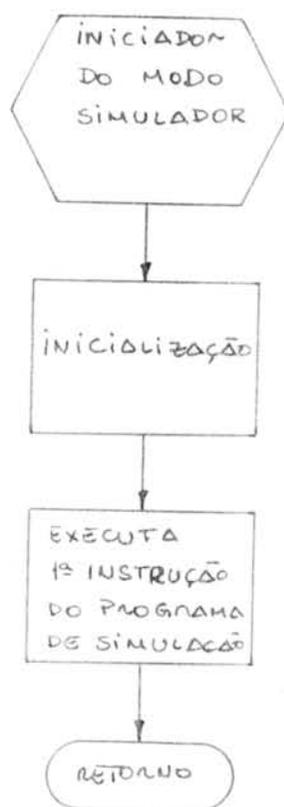


Figura 51 Iniciador do MODO SIMULADOR

O continuador do MODO SIMULADOR aparece simplificado no fluxograma da figura 52. Se o sistema estiver em transmissão, a rotina envia o próximo caracter da mensagem de transmissão correspondente. Se estiver em recepção, compara os últimos 8 caracteres recebidos com a(s) seqüência(s) de recepção. Em ambos os casos, o programa desvia após para a rotina continuadora do MODO MONITOR já que os testes de monitoração também são realizados durante a simulação.

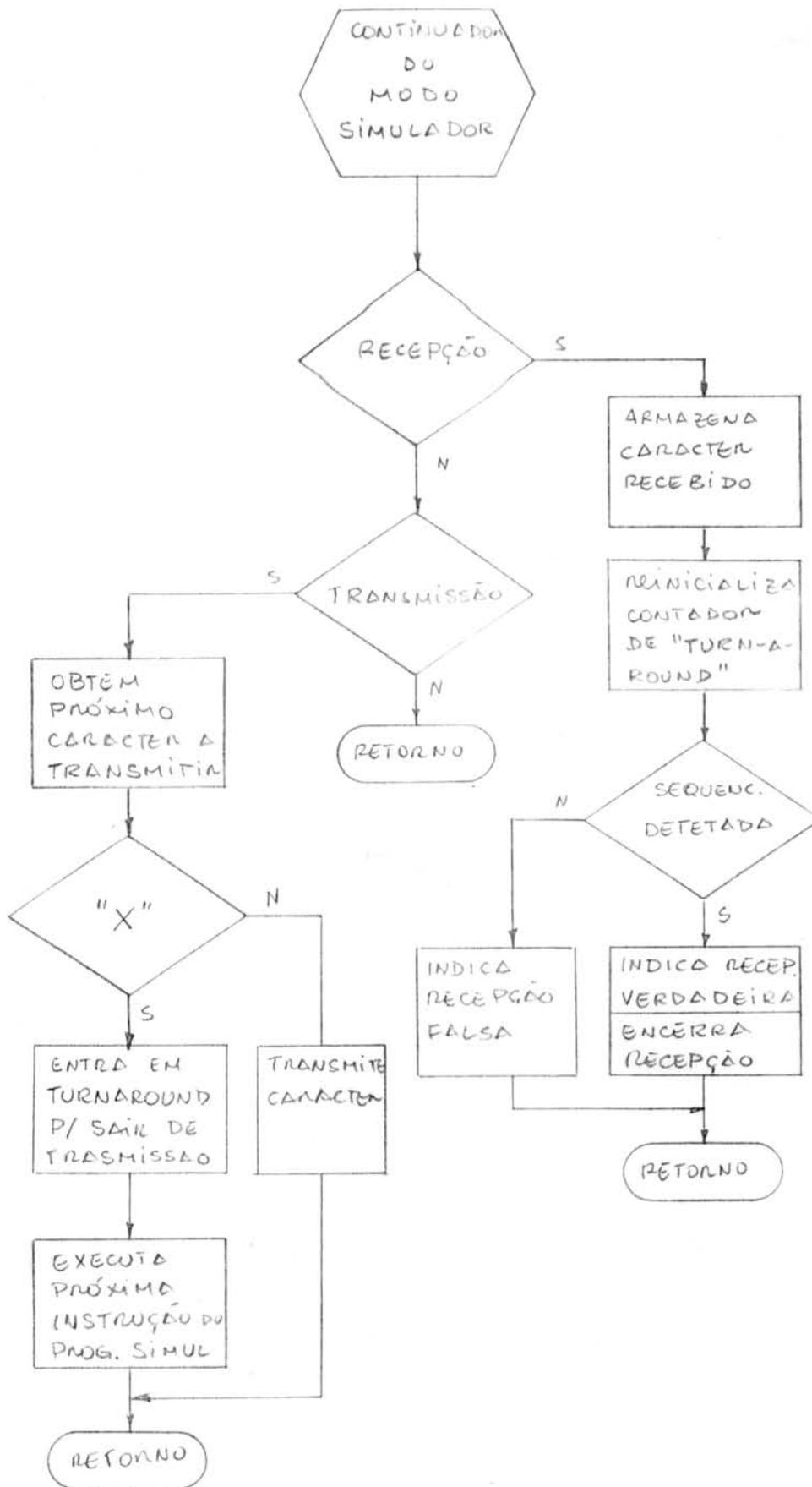


Figura 52 Continuidade do MODO SIMULADOR

### 5.5.3 Rotinas do MODO TESTE

Na figura 53 apresentamos um fluxograma da rotina iniciadora deste modo. Após a inicialização dos parâmetros e contadores do teste, é colocado o canal de comunicações 1 em transmissão e recepção.

Nas figuras 54 e 55, apresentamos um fluxograma resumido do continuador do MODO TESTE. Este continuador apresenta duas partes. Na transmissão, ele envia o próximo caracter segundo o padrão especificado. Na recepção é verificado o tipo de teste, bit, caracter ou bloco e desviado para a rotina correspondente para a contagem de erros observados.

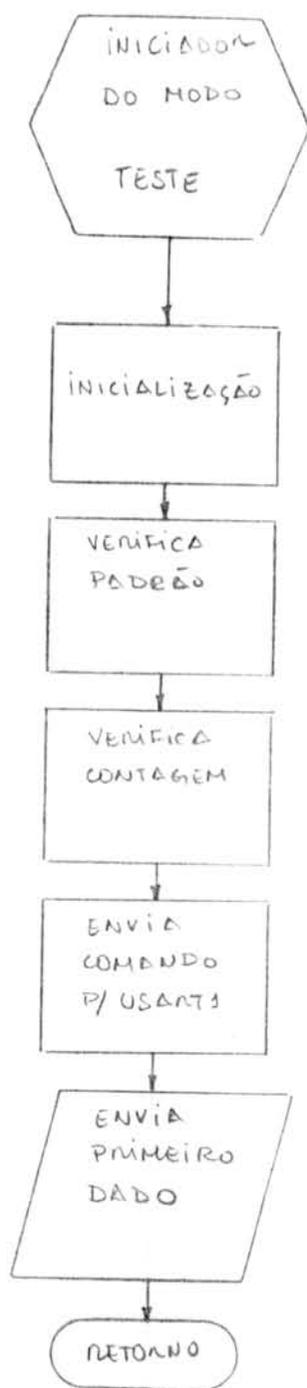


Figura 53 Iniciador do MODO TESTE

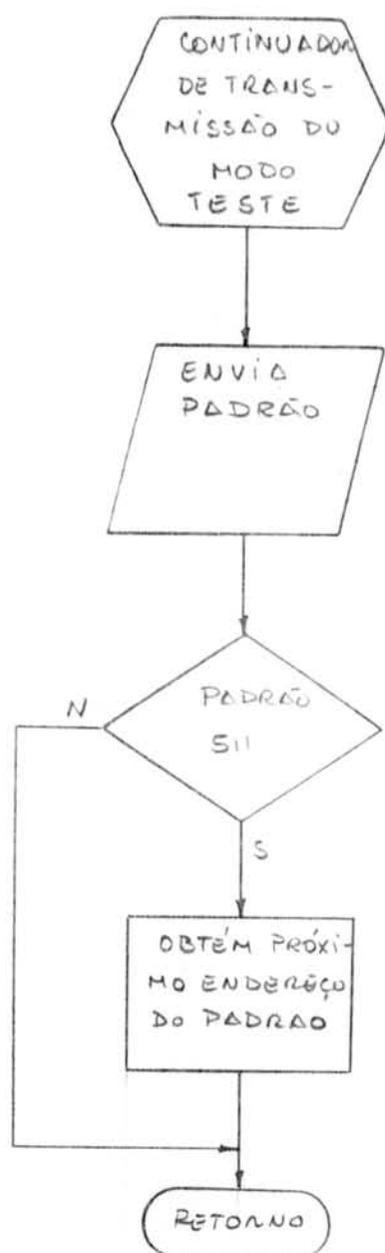


Figura 54 Continuidade de transmissão do MODO TESTE

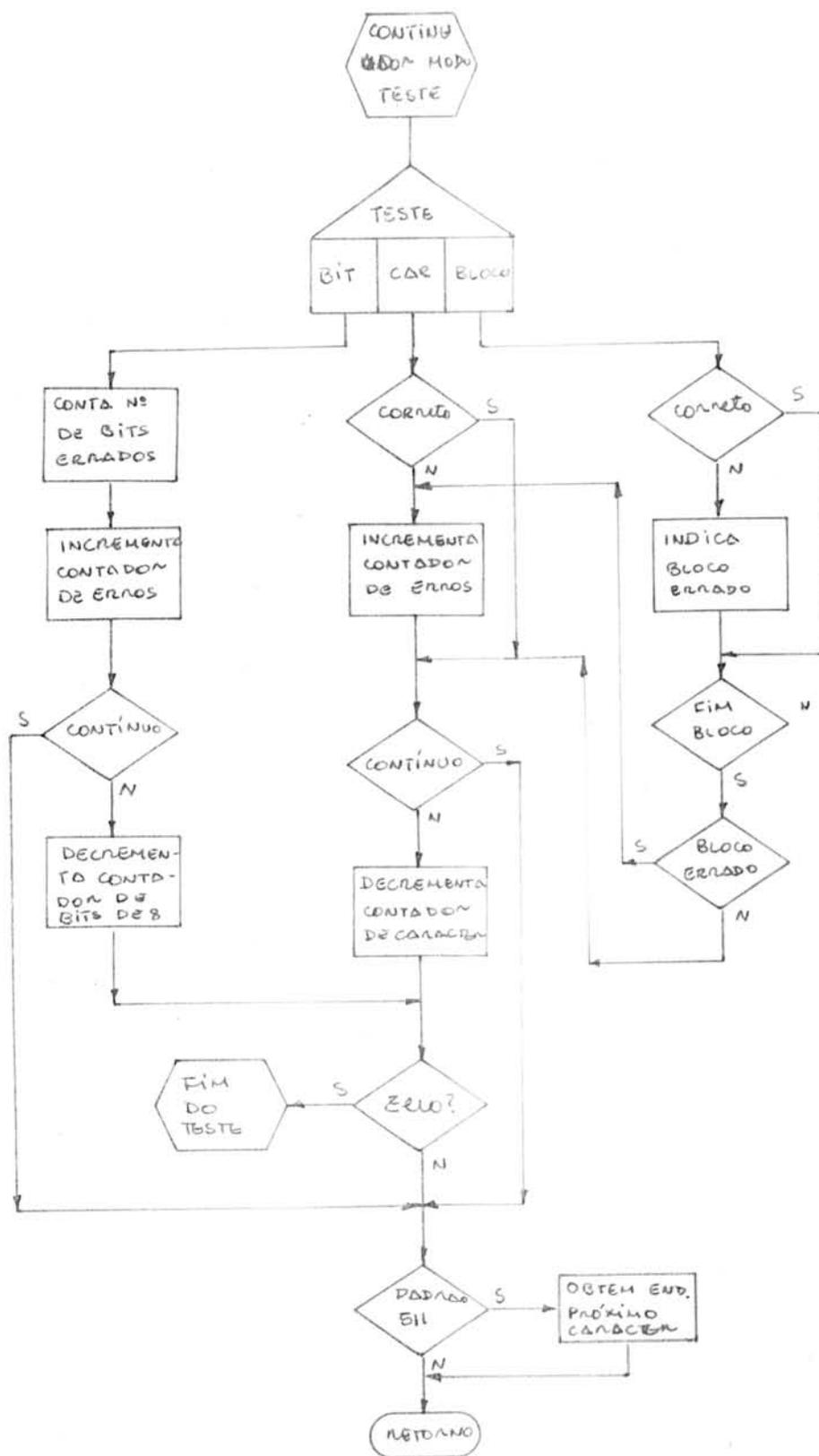


Figura 55 Continuidor de recepção do MODO TESTE

#### 5.5.4 Rotinas auxiliares

Uma série de rotinas auxiliares foram utilizadas na implementação das rotinas apresentadas acima. Apresentamos abaixo uma relação destas rotinas que se encontram, por ordem alfabética, na listagem do programa residente sob o título de ROTINAS AUXILIARES.

ACENDECURSOR	- Acende o cursor do mostrador.
ALARME	- Toca alarme audível.
APAGACURSOR	- Apaga cursor do mostrador.
ARMAZENA	- Armazena caracter na memória monitora de acordo com a JANELA.
ARMPILHA	- Armazena pilha na memória monitora.
ATUALIZA	- Atualiza tabela de campos do usuário a partir da última tela.
AVCDC	- Avança cursor dentro do campo usuário.
AVCURSOR	- Avança cursor na tela.
AVMM	- Avança ponteiros da memória monitora.
AVPAMM	- Avança ponteiro PAMM da memória monitora.
AVPTELAS	- Avança ponteiros das telas 1 e 2.
BCDBIN	- Converte número BCD em binário.
BINBCD	- Converte número binário (24 bits) em BCD.
CAMPOCURSOR	- Coloca cursor na primeira posição do campo usuário exibido na tela.
CAMP.OP	- Preenche campo opção e o campo auxiliar se houver.
CAMP.PA	- Preenche campo parâmetro.

CBCC	- Computa BCC.
CRC	- Computa CRC-16.
COMPARA	- Compara dois números de 16 bits.
COMPSEQ	- Compara duas seqüências de 8 caracteres.
CONTADOR	- Incrementa contador decimal.
DCCOBIN	- Decrementa contador binário.
DCPMM	- Decrementa ponteiro da memória monitora.
ENDCURSOR	- Encontra endereço do cursor na memória.
FIFO	- Armazena caracter na pilha FIFO.
GCPC	- Guarda caracter na posição do cursor.
GPPA511	- Gera padrão 511.
I.CONT.D	- Incrementa contador decimal.
ICURSOR	- Coloca cursor no início da tela.
IEVENTO	- Indica ocorrência de evento.
INCOBIN	- Incrementa contador binário de 24 bits.
LIMPA	- Coloca brancos na área especificada.
LIMPATELA	- Limpa mostrador.
MMM	- Mostra, em tempo real, os caracteres recebidos durante a monitoração.
MOSTRA	- Transfere para tela área especificada.
MUL7	- Multiplica número de 8 bits por 7.
MUL10	- Multiplica número de 8 bits por 10.
NIPS	- Numera instrução de programa simulação.
NMENR	- Numera mensagens de recepção.
NMENT	- Numera mensagens de transmissão.
PROGRADM	- Programa controlador de ADM.
RETCURSOR	- Desloca cursor uma posição para a esquerda.
SALVAREG	- Salva registradores do processador.

TELANORMAL	- Programa varredura normal da tela.
TELAL	- Monta tela 1.
TRANSFERE	- Transfere "n" caracteres seguidos de uma posição da memória para outra.
TRANSF1	- Transfere conteúdo do campo usuário para tabela de campo usuário e vice-versa.
TRCONT	- Transfere contador de 6 dígitos para tela.
TRTELA	- Transfere mensagem para a tela.
VSEQREC	- Verifica seqüência de recepção.
ZERA	- Coloca "n" zeros (ASCII) a partir de uma posição de memória.

## 6            APLICAÇÕES

Onde são apresentados alguns exemplos sobre o diagnóstico de falhas em redes de teleprocessamento com o uso do STTP.

### 6.1            Introdução

Neste capítulo apresentamos alguns exemplos de aplicação do sistema descrito nos capítulos anteriores. Estes exemplos estão divididos em três partes, uma para cada modo de operação do STTP. Embora os exemplos de cada modo apareçam aqui em situações independentes, é bastante provável que durante uma sessão de testes, o operador utilize os diversos modos para chegar a uma conclusão satisfatória acerca do problema.

### 6.2            Aplicações do MODO MONITOR

Como primeiro exemplo, citaremos um caso real, encontrado durante os testes de campo do protótipo.

## 6.2.1 Exemplo 1

## Descrição do problema:

O terminal escrevia determinada mensagem a partir do final da linha da tela do vídeo, ao invés do início. A mensagem aparecia assim:

```
?020 - BATCH ALUNOS DE GRADUACAO          ?ES
TA CORRETO? (SIM OU NAO)
```

## Descrição da ligação:

- Terminal - TD800
- Computador - B6700
- Velocidade - 1200 bps
- Ligação - assíncrona
- Código - ASCII/7 bits

## Programação de formato do STTP:

- Código  ASCII
- No.Bits  7
- Paridad  SEM
- Modo TX  ASSINC
- Velocid  1200
- Canal  PRINCIPAL
- Alarme  DESLIGADO

## Programação do modo monitor:

- Evento  SEQ CAR  SEQ(?020 -)
- Teste  EV INICIA

- Janela [ EVENTO ] + (0072) CARACTS
- T Inic [ 0->1 ] Linha (04)
- T Fim [ 0->1 ] Linha (05)

Descrição do teste:

Nesta configuração, o STTP não armazena nada até reconhecer a seqüência que gera o evento. A partir daí ele prossegue a armazenagem até encher a sua memória monitora. A janela especifica que só serão guardados os caracteres que compõem a seqüência de evento e mais os 72 caracteres seguintes.

Através do terminal levamos o computador a enviar a mensagem com problema. Após ser recebida a mensagem e constatado o problema, fazemos uma parada manual no teste, apertando a tecla PARAR, com a diretiva LISTAR,1. É mostrado na tela o início da memória monitora com os 32 primeiros caracteres ali guardados. Com o auxílio da tecla ESQUERDA, vamos deslocando os dados da memória pelo mostrador e podemos observar os 80 caracteres armazenados. Nesta oportunidade observamos o seguinte:

?020 - NNNNBATCH ALUNOS DE GRADUACAO

Aparecem na memória quatro caracteres nulos (código 0 octal) antes da palavra "batch". Ora, este caracter não apresenta representação gráfica e por isto ele não faz o cursor se deslocar. O computador tem em sua conta certa que enviou os 80 caracteres da linha, mas o terminal desloca o cursor de apenas 72 posições. As quatro posições que faltavam

para completar a linha da mensagem são os nulos que o terminal não conta como caracteres representáveis. Logo o cursor, ao final da mensagem se posiciona perto do final da linha, e a próxima mensagem inicia ali, e não no começo como era de se esperar. Isto ocorre devido ao fato do computador não enviar os comandos CR (retorno de carro) e LF (nova linha) antes de cada mensagem.

#### 6.2.2 Exemplo 2

Neste exemplo, não apresentamos detalhes da ligação que só concernem à programação de FORMATO e não ao teste propriamente.

Descrição do problema:

Algumas mensagens recebidas pelo terminal eram duplicadas.

Programação do modo monitor:

Evento [ SEQ CAR ] SEQ( )

Teste [ CONTA EV ]

Janela [ DESLIGADA ]

TInic [ 0->1 ] Linha(04)

TFim [ 0->1 ] Linha(05)

Descrição do teste:

A seqüência que gera eventos foi definida como

sendo:

SEQ(EOT,B,A,SEL,ENQ,X,X,X)

Esta seqüência inicia a transmissão de mensagens do computador ao terminal. Deste modo, geramos um evento para toda mensagem recebida pelo terminal.

O teste simplesmente vai armazenando os dados na memória monitora, e contando eventos. Neste teste não é feita nenhuma seleção nos dados armazenados, já que a JANELA se encontra desligada.

O teste vai armazenando continuamente, até a ocorrência do problema, ou seja, uma linha é duplicada. Paramos o teste manualmente, pressionando a tecla PARAR. Pela diretiva LISTAR,2, exibimos na tela os últimos 32 caracteres recebidos. Pela tecla DIREITA, vamos deslocando os dados da memória monitora na tela, até encontrar o diálogo que encerrou a transmissão da mensagem que foi repetida. Observamos a seguinte seqüência:

... ETX,BCC,ACK,EOT,AD1,AD2, .....

Nesta seqüência aparece sublinhada a mensagem enviada pelo terminal ao computador. É enviado ACK indicando que o terminal recebeu a mensagem corretamente, tanto que ela foi colocada no vídeo. Porém o computador enviou novamente a mesma mensagem. Isto só pode ter um motivo: o computador não reconheceu o ACK enviado pelo terminal e retransmitiu a mensagem. Uma maneira de se comprovar este fato é realizando o teste de monitoração junto ao computador e verificando se ele recebe o ACK ou não.

## 6.2.3 Exemplo 3

Descrição do problema:

O terminal não responde.

Programação do modo monitor:

Evento [ SEQ CAR ] SEQ(            )

Teste [ CONTA EV ]

Janela [ DESLIGADA ]

T Inic [ 0->1 ] Linha (04)

T Fim [ 0->1 ] Linha (05)

Descrição do teste:

Neste teste, antes de mais nada, observamos o fluxo de dados procedente do computador. Após algum tempo de monitoração, paramos o teste manualmente, apertando a tecla PARAR. Pela diretiva LISTAR,2, podemos visualizar na tela os últimos caracteres recebidos. Com o auxílio da tecla DIREITA vamos observando os dados armazenados. As seqüências se repetem:

... EOT,A,A,POL,ENQ,EOT,A,A,POL,ENQ ...

Observamos que o endereço enviado pelo computador, "AA", não corresponde ao endereço do terminal, "AB", logo este não responde porque não está sendo corretamente endereçado. Este caso é bastante simples de resolver porque basta mudar o endereço do terminal onde for mais conveniente, no próprio terminal ou no programa do computador.

#### 6.2.4 Exemplo 4

Neste exemplo deseja-se medir o "turnaround" do modem. Esta medida é o tempo entre a subida do sinal RTS (pedido para transmitir) vindo do terminal ou computador e a subida do sinal CTS (pronto para transmitir) vindo do modem. O sinal RTS está localizado no pino do interface V.24, e o sinal CTS no pino 05.

Programação do modo monitor:

A programação do modo monitor pode ser exatamente a mesma do exemplo 3, já que naquele exemplo foi definida a medida de intervalo de tempo como desejamos aqui.

Apertamos a tecla EXECUTAR e esperamos alguns segundos. Paramos o teste manualmente, apertando a tecla PARAR. Entramos com a diretiva LISTAR,4 e aparece na tela:

```
Int.tempo = 000041 ms
```

Este é o tempo entre as duas linhas, ou seja, o modem leva 41 ms para permitir o início de uma transmissão.

#### 6.2.5 Exemplo 5

Desejamos saber quanto tempo o terminal leva para passar de uma recepção para uma transmissão. Para tanto, devemos medir o tempo entre a decida do sinal DCD (PORTADORA DETECTADA) e a subida do sinal CTS (PRONTO PARA TRANSMITIR). O sinal DCD se localiza no pino 8 e o sinal CTS no pino 5 do

interface V.24.

Programação do modo teste:

T Inic [ 1->0 ] Linha(08)

T Fim [ 0->1 ] Linha(05)

Descrição do teste:

Apertamos a tecla EXECUTAR e esperamos alguns instantes para a medida ser efetuada. Paramos o teste manualmente pela tecla PARAR. Entramos com a diretiva LISTAR,4 e observamos na tela:

Int.tempo = 000054 ms

Ou seja, o terminal leva 54 ms para passar de uma recepção para transmissão.

### 6.3 Aplicações do MODO SIMULADOR

As aplicações neste modo têm um objetivo mais detalhista na tentativa de diagnosticar determinado problema. Quando todos os esforços de monitoração falham, então é a vez de se atuar ativamente na ligação. A grande diferença entre este modo e a monitoração reside no fato de, na simulação, estarmos interferindo diretamente na ligação. Neste caso a ligação é colocada em manutenção, fora da operação normal.

Em alguns casos, como no exemplo 1, a simulação se constitui numa característica altamente importante, na medida em que ela pode detectar problemas em equipamentos antes

deles serem efetivamente conectados à rede. Este seria um exemplo típico de teste de aceitação de equipamento.

### 6.3.1 Exemplo 1

Desejamos testar um terminal antes de conectá-lo à rede.

Descrição do teste:

A ligação com o terminal deve ser feita através do conector J2(ETD) do STTP. Estaremos simulando um computador que envia, inicialmente, a seqüência "pol" ao terminal. O programa de simulação pode ser o mesmo apresentado em 3.4.5.1. Os parâmetros de simulação são:

Simula [ ECD ] RTS/CTS(050)ms

Element [ MESTRE ] R APS(0500)ms

onde indicamos que estamos simulando um modem (ECD) cujo "turnaround" é de 50 ms, e somos o elemento mestre (computador) que espera 500 ms entre uma mensagem e outra.

Os parâmetros relativos à parte de monitoração são escolhidos à vontade.

Nesta situação, o STTP fica enviando mensagens de "pol" a cada 500 ms. O terminal responde ou não a esta seqüência. As trocas de mensagem são mostradas em tempo real no mostrador e armazenadas na memória de monitoração. A parada do teste é manual. Pela observação da memória monitora ao final do teste, podemos verificar se o terminal está respon-

dendo corretamente à seqüência enviada.

Uma vez constatado a correção nas respostas à seqüência "pol", podemos enviar a seqüência "sel". Para isto, devemos novamente entrar com o programa de simulação. O programa pode ser do tipo mostrado em 3.4.5.2. Repetimos o teste e verificamos se o terminal também responde corretamente a esta seqüência.

Caso tudo esteja correto, podemos conectar o terminal à rede com segurança, evitando a perda de tempo de conectar um terminal para depois verificar se tudo está funcionando corretamente.

### 6.3.2 Exemplo 2

Desejamos testar um computador a partir de um terminal remoto.

Descrição do teste:

Substituímos o terminal remoto pelo STTP. Conectamos o modem ao conector J2(ECD). Os parâmetros de simulação são:

```
Simula [ ETD ]
```

```
Element [ ESCRAVO ] R APS(0010)ms
```

onde simulamos um equipamento terminal de dados (ETD) na situação de elemento escravo que responde 10 ms após receber a mensagem.

O programa de simulação pode ser, por exemplo, o

apresentado em 3.4.5.3, que pode responder tanto às seqüências "pol" como "sel".

O teste é colocado em execução e o STTP envia uma mensagem específica, conforme o programa de simulação, a cada seqüência "pol" recebida, e está sempre pronto a receber texto, quando é inquirido por uma seqüência "sel".

Ao final do teste, pode-se observar em sua memória monitora o conteúdo da transação e analisar a sua correção.

#### 6.4 Aplicações do MODO TESTE

As aplicações deste modo visam testar a qualidade da ligação, linha e modem. Estes testes podem indicar: qual o modem com problema, se a linha está interrompida ou se a ligação encontra-se degradada.

##### 6.4.1 Exemplo 1

Neste exemplo, o computador não recebe mensagem do terminal.

Programação do modo teste:

```
Padrão  [ 511 ]  
Test er [ BIT ]  
Contag  [ 10E6 ]
```

Descrição do teste:

Verificou-se que o modem junto ao terminal não recebe sinal algum da linha. O modem junto ao computador indica que estão sendo enviados dados para a linha. Esta situação pode ter duas causas: ou a linha se encontra interrompida ou o modem remoto se encontra defeituoso.

Primeiro verificamos a linha. Colocamos o modem remoto (junto ao terminal) em laço analógico, conforme figura 56.

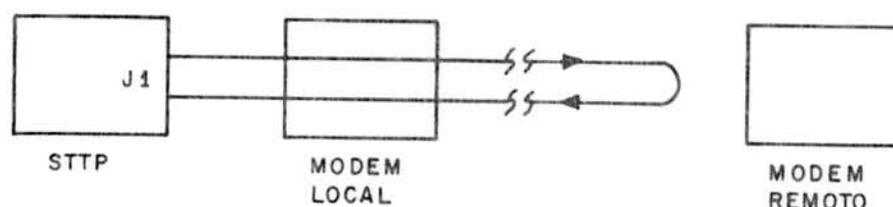


Figura 56 Teste de linha

Colocamos o STTP em execução. Ao final do teste a parece automaticamente no mostrador:

```
Erros = 00000000<<FIM TESTE>>
```

Isto indica que a linha se encontra em perfeitas condições, não havendo interrupções nem ruído observável que degrade a ligação.

Verificamos agora as condições do modem remoto. Para isto, colocamos o modem remoto em laço digital, conforme esquema da figura 57.

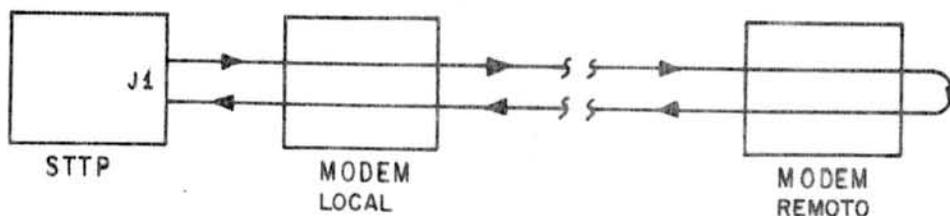


Figura 57 Teste do modem remoto

Entramos novamente em execução e esperamos o final automático do teste, quando aparece na tela:

Erros = 00556471 ms

Esta alta taxa de erros indica que o modem remoto encontra-se defeituoso. Observar que a contagem de erros neste teste não pode ser de um milhão já que o padrão 511 apresenta bits "0" e "1" em quantidades quase iguais. Se o modem estiver inoperante e recebermos apenas "0" ou apenas "1", a contagem de erros será aproximadamente a metade do número de bits enviados.

#### 6.4.2 Exemplo 2

Neste exemplo, desejamos avaliar a qualidade de uma ligação.

Descrição do teste:

Inicialmente fazemos uma medida de taxa de erros de bits/milhão, com o padrão 511. A programação do modo teste

é idêntica à do exemplo anterior. Colocamos o modem remoto em laço digital, como na figura 57. No final do teste foi observado:

Erros = 00000147 <<FIM TESTE>>

Repetimos o mesmo teste e verificamos:

Erros = 00000052 <<FIM TESTE>>

Houve uma redução drástica na contagem de erros, o que nos faz supor que os mesmos ocorrem em rajadas de tamanhos diversos.

Mudamos o teste para caracter/milhão, ainda com o padrão 511. Verificamos ao final de vários testes, os seguintes resultados:

Erros = 00000005 <<FIM TESTE>>

Erros = 00000012 <<FIM TESTE>>

Erros = 00000007 <<FIM TESTE>>

Estes resultados nos reforçam a hipótese dos erros em rajadas curtas que atingem poucos caracteres por vez.

Por último, mudamos o teste para bloco/mil com o mesmo padrão. Os resultados obtidos foram:

Erros = 00000002 <<FIM TESTE>>

Erros = 00000001 <<FIM TESTE>>

Erros = 00000001 <<FIM TESTE>>

Podemos concluir que a ligação é razoavelmente satisfatória porém apresentando erros em rajadas curtas e esparsas que interferem em 1 ou 2 blocos de 511 bits de tamanho numa transmissão de mil blocos.

CONCLUSÃO

Ao concluir este trabalho, gostaria de fazer uma auto-avaliação. Ao ser definido o STTP, procurou-se colocar nele tudo aquilo que alguém que trabalha em transmissão de dados, gostaria de ter à mão na hora de descobrir um problema numa rede complexa, com muitos equipamentos diferentes, de fabricantes diferentes. Foram comparados vários equipamentos do gênero e de cada um foi tirado aquilo que mais interessava. Reuniu-se tudo e estava definido o STTP. De todos os itens da sua definição, não foram concretizados uns poucos que passamos a enumerar:

- programação fixa de alguns protocolos mais comuns;
- capacidade de auto-teste;
- saídas para TTY, vídeo ou K7;
- equipamento portátil;
- baixo consumo.

A programação fixa de alguns protocolos, assim como a capacidade de auto-teste, não foram implementados por não haver mais espaço disponível de memória e pela absoluta falta de tempo em aumentá-la. Por outro lado, estes dois itens constituem-se em sofisticacões que não comprometem o projeto.

As saídas para TTY, vídeo ou K7, foram substituídas por uma saída para impressora.

O equipamento portátil não se tornou realizável por necessitar de recursos não disponíveis a curto prazo no PGCC. O sistema de placas padrões utilizado pelo curso, impos

sibilitou uma montagem mais compacta, descaracterizando em muito o conceito de portabilidade. Este conceito, no entanto, é muito discutível, já que inúmeros fabricantes alegam que seus produtos são portáteis apenas por possuírem uma alça, quando o seu peso é maior do que uma pessoa sozinha poderia carregar com certa facilidade. O STTP não chega a ser um equipamento pesado (5 kg), porém, sua construção quase cúbica, não predispõe à portabilidade.

O item consumo apresenta dois aspectos a considerar: o consumo da rede AC e o consumo da fonte DC. Quando se fala em equipamento portátil, o quesito consumo se refere ao tamanho e peso da fonte DC. O consumo da rede AC é inferior ao de uma lâmpada de 150 watts, o que pode ser considerado baixo. Já o consumo da fonte DC se eleva a 5 ampères. Uma fonte DC chaveada e bem dimensionada tem menor peso e tamanho que uma fonte série convencional. Mas, devido aos recursos disponíveis, limitamo-nos a construir uma fonte convencional, deixando para um desenvolvimento futuro um melhor dimensionamento da fonte.

Em termos de "Hardware", o maior problema do STTP está localizado no seu mostrador. Este é constituído de 32 "displays" alfanuméricos. A sua escolha deveu-se exclusivamente à sua disponibilidade no mercado nacional de componentes. Existem, sem dúvida, outras alternativas, muito mais convenientes, porém não eram facilmente disponíveis na época da definição do mostrador. Entre as desvantagens deste mostrador estão o seu alto consumo, no número de pastilhas auxiliares (em torno de 100) e os problemas de "lay-out" decorren-

tes de um número tão grande de componentes.

Embora o STTP possa ser reproduzido em pequena escala, a sua industrialização requer uma segunda fase de projeto. Nesta fase, passarão a ser de vital importância a redefinição do mostrador, a otimização da fonte de alimentação, o "lay-out" dos componentes e placas, os problemas de cabeamento e conectores e um novo projeto do gabinete. Estes problemas fugiram de todo aos objetivos deste trabalho.

Entre as qualidades do STTP, a sua fácil operação talvez seja a mais importante. Como este instrumento interage num alto nível com uma rede sofisticada, o operador não pode se emaranhar com uma complexa operação do instrumento.

Outra característica importante é a sua habilidade em reconhecer e disparar testes em seqüências específicas. Esta é a chave para apontar áreas de problema que possam estar ocorrendo na rede.

A sua memória de armazenamento de 2048 palavras é grande o suficiente para armazenar informação pertinente a uma determinada área com sintomas de problema. O tamanho de memória necessário é diminuído pelo aumento da capacidade do equipamento em realizar "amarrações" mais sofisticadas.

A habilidade de monitorar através do interface entre o modem e o terminal, não apenas dados, mas também informações de tempo, é de vital importância. Aproximadamente 80% dos problemas do sistema podem ser resolvidos através de uma monitoração não intrusiva. Isto economiza tempo de operação do sistema o que resulta numa diminuição do custo.

A simulação do terminal ou computador é outra ca-

racterística importante na manutenção mais detalhada de blocos específicos da rede.

Finalmente, a sua flexibilidade de operação comporta o uso em estruturas de dados de vários códigos, a diversas velocidades e com capacidade de realizar verificação de erro de inúmeras maneiras.

O crescimento contínuo de redes de teleprocessamento no país, tanto privadas como públicas, requer o uso de ferramentas adequadas para a sua manutenção. A nível nacional, não existe nenhum outro projeto de sistema semelhante, que seja do conhecimento do autor. O único caminho para os usuários tem sido a importação que, além de onerosa, tem como principal consequência o aumento da nossa já tão grande dependência do exterior. O desenvolvimento de um sistema como este traz consigo, além dos benefícios diretos às partes envolvidas, um aceno à indústria nacional que tanto se tem desgastado com o pagamento de "royalties" ao exterior para a fabricação de equipamentos sofisticados.

Para a realização deste trabalho, o autor gastou 15 meses em tempo parcial e 2 meses em tempo integral, contando apenas com o auxílio de um auxiliar de pesquisa de 20 horas/semana para a confecção da fonte de alimentação e a montagem mecânica do protótipo.

O projeto pode ser dividido em 4 fases distintas, apresentadas abaixo, com o tempo dispendido em cada uma delas:

- Definição do sistema (6 meses),
- Projeto e simulação do software (4 meses);

- Projeto e depuração do hardware (3 meses);
- Depuração do sistema a nível de protótipo (4 meses).

Na primeira fase, de definição, foi feito um levantamento dos equipamentos similares e de suas características. Logo após, foram definidos os testes que seriam realizados pelo STTP e a seguir foi definido o método de programação do mesmo. Ainda nesta fase foi definida a estrutura de hardware necessária. Esta fase foi a mais longa e também a mais difícil devido principalmente à pouca documentação obtida dos equipamentos consultados.

As fases seguintes consistiram no desenvolvimento do sistema definido na primeira fase. O projeto de software e sua simulação foi toda realizada em cima do SED (sistema de entrada de dados) desenvolvido pelo CPGCC para o desenvolvimento de sistemas à base de microprocessador. O próprio SED é implementado com um microprocessador 8080A e possui todo um software de apoio que permite a depuração de programas escritos em assembler do 8080A. Através do SED foi feita também a gravação do código objeto do programa desenvolvido nas memórias de reprom do STTP.

O projeto, montagem e depuração de hardware foi uma fase relativamente curta, onde procurou-se não reinventar a roda utilizando na medida do possível todo o apoio fornecido pelos LSI's da Intel.

A última fase consistiu na depuração do sistema já a nível de protótipo, onde tínhamos o programa residente interagindo com o hardware montado e realizando os testes de-

finidos na primeira fase. Esta depuração consistiu principalmente na depuração do programa residente, onde se localizaram a maioria das falhas encontradas.

Durante as quatro fases, cerca de 20% do tempo foi dedicado a confecção da documentação do projeto, incluindo relatórios, desenhos, diagramas e a escrita deste trabalho. Outros 20% do tempo disponível ao longo do projeto foram gastos pelo autor para a realização de tarefas secundárias ligadas à confecção do protótipo, como a compra de certos componentes especiais, a confecção da placa de painel, o projeto mecânico, a contratação de serviços de terceiros, etc..

Concluimos que uma equipe bem estruturada, onde a documentação e a confecção do protótipo não ficassem exclusivamente ao encargo do pesquisador, e mais, onde o pesquisador possuísse tempo integral para se dedicar ao projeto, o mesmo poderia ser realizado em cerca de oito meses.

APÊNDICE A LISTA DE DEFINIÇÕES DOS CIRCUITOS DO INTERFACE V.  
24 (RS-232-C)

Circuito 101 (EIA AA) - Terra de proteção

Este condutor deve ser eletricamente ligado à car  
ça do equipamento.

Circuito 102 (EIA AB) - Terra de sinal

Este condutor estabelece o potencial terra de re-  
ferência comum para todos os circuitos de ligação, exceto o  
circuito 101. Dentro do ECD, este circuito poderá ser trazido  
a um ponto onde ele poderá ser conectado ao circuito 101 atra-  
vés de uma ponte. Esta ponte pode ser colocada ou retirada na  
instalação, como for requerido por regulamentação pertinente  
ou para minimizar a introdução de ruído nos circuitos eletrô-  
nicos.

Circuito 103 (EIA BA) - Dados transmitidos

Direção: para o ECD.

Os sinais de dado gerados pelo ETD, a serem trans  
mitidos pelo canal de dados a uma ou mais estações remotas,  
são transferidos por este circuito ao ECD.

Circuito 104 (EIA BB) - Dados recebidos

Direção: do ECD.

Os sinais de dado gerados pelo ECD, em resposta  
aos sinais de linha do canal de dados recebidos de uma esta-  
ção remota, são transferidos neste circuito para o ETD.

Circuito 105 (EIA CA) - Pedido para transmitir

Direção: para o ECD.

Os sinais neste circuito controlam a função de transmissão pelo canal de dados do ECD.

A condição ON faz o ECD assumir o modo de transmissão pelo canal de dados.

A condição OFF faz o ECD assumir modo de não transmissão pelo canal de dados, após a transmissão de todos os dados do circuito 103.

Circuito 106 (EIA CB) - Pronto para transmitir

Direção: do ECD.

Os sinais neste circuito indicam se o ECD está em condições de transmitir dados no canal de dados.

A condição ON indica que o ECD está pronto para transmitir dados pelo canal de dados.

A condição OFF indica que o ECD não está pronto para transmitir dados pelo canal de dados.

Circuito 107 (EIA CC) - ECD pronto

Direção: do ECD.

Os sinais neste circuito indicam que o ECD está pronto para operar.

A condição ON indica que o ECD está conectado à linha e pronto para a troca de sinais de controle com o ETD e iniciar a troca de dados.

A condição OFF indica que o ECD não está pronto para operar.

Circuito 108/2 (EIA CD) - Terminal de dados pronto

Direção: para o ECD.

Os sinais neste circuito são usados para controlar a conexão do ECD à linha.

A condição On indica que o ETD está pronto para operar, prepara o ECD para conectar-se à linha e mantém esta conexão depois desta ser realizada por meios suplementares.

A condição OFF faz com que o ECD se desconecte da linha.

Circuito 109 (EIA CF) - Detector de portadora presente

Direção: do ECD

Os sinais neste circuito indicam se o sinal de dados recebidos está dentro de limites apropriados.

A condição ON indica que o sinal recebido está dentro de limites apropriados.

A condição OFF indica que o sinal recebido não está dentro de limites apropriados.

Circuito 110 (EIA CG) - Detector de qualidade do sinal de dados

Direção: do ECD.

Os sinais neste circuito indicam se há uma razoável probabilidade de erro nos dados recebidos no canal de dados.

A condição ON indica que não há razão para acreditar que um erro ocorreu.

A condição OFF indica que há uma probabilidade ra

zoável de erro.

Circuito 111 (EIA CH) - Seletor de taxa de transmissão

Direção: para o ECD.

Os sinais neste circuito são usados para selecionar uma de duas taxas de transmissão em ECD síncronos de dupla taxa, ou para selecionar um.

A condição ON seleciona a taxa mais alta.

A condição OFF seleciona a taxa mais baixa.

Circuito 112 (EIA CI) - Seletor de taxa de transmissão (fonte ECD)

Direção: do ECD.

Os sinais neste circuito são usados para selecionar uma de duas taxas de transmissão no ETD que coincida com a taxa usada no ECD.

A condição de ON seleciona a taxa mais alta.

A condição de OFF seleciona a taxa mais baixa.

Circuito 113 (EIA DA) - Relógio de transmissão de dados (fonte ETD)

Direção: para o ECD.

Os sinais neste circuito dão ao ECD a informação de tempo para a transmissão de dados.

A condição neste circuito deve ser ON e OFF por períodos nominalmente iguais, sendo que as transições de ON para OFF devem nominalmente indicar o centro de cada elemento de sinal do circuito 103.

Circuito 114 (EIA DB) - Relógio de transmissão de dados (fonte ECD)

Direção: do ECD.

Os sinais neste circuito dão ao ETD informação de tempo para a transmissão de dados.

As condições ON e OFF deste circuito devem ocorrer por períodos nominalmente iguais. O ETD deve apresentar um sinal de dado no circuito 103 na qual as transições entre os elementos de sinal ocorram nominalmente ao tempo em que ocorram as transições de OFF para ON do circuito 114.

Circuito 115 (EIA DD) - Relógio de recepção (fonte ECD)

Direção: do ECD.

Os sinais neste circuito dão ao ETD informação de tempo para a recepção de dados.

As condições ON e OFF deste circuito devem ocorrer por períodos nominalmente iguais e a transição de ON para OFF deve indicar o centro de cada elemento de sinal do circuito 104.

Circuito 118 (EIA SBA) - Dados transmitidos canal secundário

Direção: para o ECD.

Este circuito é equivalente ao circuito 103, exceto que ele é usado para transmitir dados pelo canal secundário.

Circuito 119 (EIA SBB) - Dados recebidos canal secundário

Direção: do ECD.

Este circuito é equivalente ao circuito 104, exceto que ele é usado para receber os dados pelo canal secundário.

Circuito 120 (EIA SCA) - Pedido para transmitir do canal secundário

Direção: para o ECD.

Este circuito é equivalente ao circuito 105, exceto que ele é usado para controlar a função de transmissão do ECD pelo canal secundário.

A condição ON faz o ECD assumir o modo de transmissão para o canal secundário.

A condição OFF faz o ECD assumir o modo de não transmissão pelo canal secundário, após todos os dados serem transmitidos à linha pelo circuito 118.

Circuito 121 (EIA SCB) - Pronto para transmitir do canal secundário

Direção: do ECD.

Indicar se o ECD está pronto para transmitir pelo canal secundário.

A condição ON indica que o ECD está pronto para transmitir pelo canal secundário.

A condição OFF indica que o ECD não está pronto para transmitir pelo canal secundário.

Circuito 122 (EIA SCF) - Detector de portadora do canal secundário

Direção: do ECD.

Este circuito é equivalente ao circuito 109, exceto que ele é usado para indicar se o sinal de linha do canal secundário está dentro de limites apropriados.

APÊNDICE B CÓDIGO ASCII

CARACTER	HEXA	CARACTER	HEXA	CARACTER	HEXA
A	41	0	30	DLE	10
B	42	1	31	EM	19
C	43	2	32	ENQ	05
D	44	3	33	EOT	04
E	45	4	34	ESC	1B
F	46	5	35	ETB	17
G	47	6	36	ETX	03
H	48	7	37	FF	0C
I	49	8	38	FS	1C
J	4A	9	39	GS	1D
K	4B	SP	20	HT	09
L	4C	!	21	LF	0A
M	4D	"	22	NAK	15
N	4E	#	23	NUL	00
O	4F	\$	24	RS	1E
O	50	%	25	SI	0F
Q	51	&	26	SO	0E
R	52	'	27	SOH	01
S	53	(	28	STX	02
T	54	)	29	SUB	1A
U	55	*	2A	SYN	16
V	56	+	2B	US	1F
W	57	,	2C	VT	0B
X	58	-	2D		
Y	59	.	2E		
Z	5A	/	2F		
a	61	:	3A		
b	62	;	3B		
c	63	<	3C		
d	64	=	3D		
e	65	>	3E		
f	66	?	3F		
g	67	@	40		
h	68	[	5B		
i	69	]	5C		
j	6A	^	5D		
k	6B	_	5E		
l	6C	`	5F		
m	6D	{	60		
n	6E		60		
o	6F	~	7B		
p	70	~	7C		
q	71	~	7D		
r	72	~	7E		
s	73	BEL	07		
t	74	BS	08		
u	75	CAN	18		
v	76	CR	0D		
w	77	DC1	11		
x	78	DC2	12		
y	79	DC3	13		
z	7A	DC4	14		
		DEL	7F		

APÉNDICE C CÓDIGO EBCDIC

CARACTER	HEXA	CARACTER	HEXA	CARACTER	HEXA
A	C1	0	F0	DEL	07
B	C2	1	F1	DLE	10
C	C3	2	F2	DS	20
D	C4	3	F3	EM	19
E	C5	4	F4	ENQ	2D
F	C6	5	F5	EOB	26
G	C7	6	F6	EOT	37
H	C8	7	F7	ESC	27
I	C9	8	F8	ETB	26
J	D1	9	F9	ETX	03
K	D2	&	50	FF	0C
L	D3	-	60	FS	22
M	D4	/	61	HT	05
N	D5	.\$	5B	IFS	1C
O	D6	¢	4A	IGS	1D
P	D7	!	5A	IL	17
Q	D8	:	7A	IRS	1E
R	D9	#	7B	IUS	1F
S	E2	,	6B	LC	06
T	E3	.	4B	LF	25
U	E4	<	4C	NAK	3D
V	E5	*	5C	NL	15
W	E6	®	6C	NUL	00
X	E7	@	7C	PF	04
Y	E8	(	4D	PB	34
Z	E9	)	5D	PRE	27
a	81	-	6D	RES	14
b	82	,	7D	RLF	09
c	83	+	4E	RS	35
d	84	;	5E	SI	0F
e	85	>	6E	SN	2A
f	86	=	7E	SMM	0A
g	87	┌	4F	SO	0E
h	88	└	5F	SOH	01
i	89	?	6F	SOS	21
j	91	"	7F	SPACE	40
k	92	{	C0	STX	02
l	93	}	D0	SUB	3F
m	94	~	E0	SYN	32
n	95	^	A1	UC	36
o	96	`	79	VT	0B
p	97	!~	6A		
q	98	BEL	2F		
r	99	BS	16		
s	A2	BYP	24		
t	A3	CAN	18		
u	A4	CC	1A		
v	A5	CR	0D		
w	A6	DC1	11		
x	A7	DC2	12		
y	A8	DC3	13		
z	A9	DC4	14		

APÉNDICE D CÓDIGO BAUDOT

CARACTER	HEXA
BLK	00
E	01
LF	02
A	03
SP	04
S	05
I	06
Y	07
CR	08
D	09
R	0A
J	0B
N	0C
F	0D
C	0E
K	0F
T	10
Z	11
L	12
W	13
H	14
Y	15
P	16
Q	17
O	18
B	19
G	1A
FIGS	1B
M	1C
X	1D
V	1E
LTRS	1F
BLK	20
3	21
LF	22
-	23
SP	24
BELL	25
8	26
7	27
CR	28
\$	29
4	2A
'	2B
/	2C
!	2D
:	2E

APÊNDICE E ENDEREÇOS DE ENTRADA E SAÍDA

SINAL	ENDEREÇO	FUNÇÃO
ES0/	000	Controlador de ADM
ES1/	020	Controlador de interrupção
ES2/	040	Registrador de posição do cursor
ES3/	060	Registrador de habilitação do cursor
ES4/	100	Registrador tecla
ES5/	120	Registrador velocidade
ES6/	140	Reseta interrupção início e fim
	141	Reseta interrupção RTS
ES7/	160	Registrador de dados da usart 1
	161	Registrador de controle da usart 1
ES8/	200	Registrador de dados da usart 2
	201	Registrador de controle da usart 2
ES9/	220	Reseta interrupção do relógio de 1 ms
ES10/	240	Registrador seleção de linhas de início e fim
ES11/	260	Registrador de controle interface comunicações
ES12/	300	Liga CTS/DTE (pronto para transmitir)
ES13/	320	Vago
ES14/	340	Registrador de dados da impressora
	341	Registrador de controle da impressora
ES15/	360	Alarme

APÊNDICE F ALOCAÇÃO DOS PINOS NO BARRAMENTO

PINO	SINAL	PINO	SINAL
C1	terra	F1	terra
C2	BD0	F2	I/OR/
C3	BD1	F3	I/OW/
C4	BD2	F4	ES2/
C5	BD3	F5	ES3/
C6	BD4	F6	ES4/
C7	BD5	F7	ES5/
C8	BD6	F8	ES6/
C9	BD7	F9	ES7/
C10	INT0/	F10	ES8/
C11	INT1/	F11	ES9/
C12	INT2/	F12	ES10/
C13	INT3/	F13	ES11/
C14	INT4/	F14	ES12/
C15	INT5/	F15	BE0
C16	INT6/	F16	F2TTL
C17	OSC	F17	Reset/
C18	DRQ2	F18	DACK2/
C19	-5V	F19	-5V
C20	-12V	F20	-12V
C21	+12V	F21	+12V
C22	+5V	F22	+5V

BIBLIOGRAFIA

- 1) ANSI. Reference model for distributed systems. In: Auerbach Data Communications Management. Philadelphia, C 1978. Rep. 51-01-30.
- 2) BJORNER, D. Finite State Automation - Definition of Data Communication Line Control Procedures. In: FALL JOINT COMPUTER CONFERENCE, Houston, Nov. 17-19, 1970. Proceedings. Montvale, AFIPS, C 1970, p. 477-91.
- 3) BURROUGHS CORPORATION. TD 700/800 Equipment Reference Manual. July 1973.
- 4) CCITT. V.24 - List of Definition for Interchange Circuits Between Data-Terminal Equipment and Circuit Terminating Equipment. In: Data Transmission over the Telephone Network. Geneva, International Telecommunication Union, 1977. (v.) VIII.1, Sec. 2, p. 79-88.
- 5) \_\_\_\_\_. V.52 - Characteristics of Distortion and Error-Rate Measuring Apparatus for Data Transmission. In: Data Transmission Over the Telephone Network. Geneva, International Telecommunication Union, 1977. (v.) VIII.1, Sec. 5, p. 186-8.
- 6) DENNIS, J.B. Concurrency in Software Systems. IEEE Transactions of Software Engineering, Austin, 5(5): 310-315, Set. 1977.
- 7) DUERR, J.R. Data Communications Testing Overview - Protocol Analysis. Computer Design, Littleton, 18(2): 10-20, Feb. 1979.
- 8) \_\_\_\_\_. Data Communications Testing Overview - Digital Testing. Computer Design, Littleton, 18(3):12-8, Mar. 1979.
- 9) \_\_\_\_\_. Data Communications Testing Overview - Analog Testing. Computer Design, Littleton, 18(4):10-3, Apr. 1979.
- 10) EIA. Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington, Electronic Industries Association, 1969.
- 11) GLASGAL, R. Basic Techniques in Data Communications. Dedham, Artech House, C 1977.
- 12) HEWLETT-PACKARD. Serial Data Analyser Model 1640A. June 1978.

- 13) \_\_\_\_\_ . Symptomatic Troubleshooting of Computer Networks. Feb. 1979.
- 14) IBM. Synchronous Data Link Control - General Information. May 1975.
- 15) \_\_\_\_ . Binary Sinchronous Communications. s.d.
- 16) INTEL CORPORATION. 8080 Microcomputer Systems User's Manual. Santa Clara, Sep. 1975.
- 17) \_\_\_\_\_ . MCS-80 User's Manual. Santa Clara, Oct. 1977.
- 18) LANCASTER, Don. CMOS Cookbook. Indianápolis, Howard W. Sams, 1978.
- 19) \_\_\_\_\_ . TTL Cookbook. Indianápolis, Howard W.Sams, 1978.
- 20) MOSER, F.B. Controlador assíncrono de teleprocessamento. Porto Alegre, PGCC da UFRGS, 1975. (Dissertação)
- 21) \_\_\_\_\_ . Manual de programação para o montador do microprocessador 8080 "AS-80". Porto Alegre, PGCC da UFRGS, 1979.
- 22) MOTOROLA. MCMOS Handbook. 1974.
- 23) NATIONAL SEMICONDUCTOR CORPORATION. TTL Databook. Santa Clara, 1976.
- 24) POISL, R.C. Concentrador de dados com microprocessador. Porto Alegre, PGCC da UFRGS, 1977. (Dissertação)
- 25) SMITH, L. Using the 8251 Universal Synchronous Assynchronous Receiver Transmitter. Santa Clara, Intel, 1976.
- 26) TELEBRÁS. Sistema de práticas Telebrás. s.l. 1978.
- 27) TEXAS INSTRUMENTS. Designing With TTL Integrated Circuits. Tokyo, McGraw-Hill, 1971.

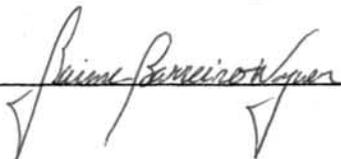
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Pós-Graduação em Ciência da Computação da UFRGS

SISTEMA DE TESTES DE REDES DE  
TELEPROCESSAMENTO

DISSERTAÇÃO APRESENTADA AOS SRS.

  
\_\_\_\_\_

Roberto Fedilhe Pinheiro  
\_\_\_\_\_

  
\_\_\_\_\_

Francisco Bernardo Moser Filho  
\_\_\_\_\_

Visto e permitida a impressão  
Porto Alegre, ..8../..4../8...



Coordenador do Curso de Pós-Graduação  
em Ciência da Computação