

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Comutador de Dados  
Digitais para TDM  
Determinístico E1, Visando  
uma Implementação em  
Microeletrônica**

por

Oscar Pedro Agurto Hoyos

Dissertação submetida como requisito parcial  
para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Tiaraju V. Wagner  
Orientador

Porto Alegre, maio de 1996.



SABi



05225351

UFRGS  
INSTITUTO DE INFORMÁTICA  
BIBLIOTECA

## CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Agurto Hoyos, Oscar Pedro

Comutador de dados digitais para TDM determinístico E1, visando uma implementação em microeletrônica / Oscar Pedro Agurto Hoyos.—Porto Alegre: CPGCC da UFRGS, 1996. 236 p.: il.

Dissertação (mestrado)—Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 1996. Orientador: Wagner, Tiaraju V.

1. Microeletrônica. 2. Comunicação de Dados. 3. Sistemas de Comutação. 4. Comutação de Circuitos. 5. Redes Corporativas. 6. Projeto de Sistemas Digitais. I. Wagner, Tiaraju V. II. Título.

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA	
N.º CHAMADA 621.38-181.4(043) A284C	N.º REG.: 32399
ORIGEM: D	DATA: 12/07/96
FUNDO: II	PREÇO: R\$ 25,00
	CA.A.: 108/07/96
	FURN.: II

*Microeletrônica - SBU/II*  
*Comunicação: Dados*  
*Redes corporativas*  
*Projeto: Sistemas digitais*  
*Comutação: circuitos*

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Hélgio Trindade

Pró-Reitor de Pesquisa e Graduação: Prof. Cláudio Scherer

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

*ENPq 3.04 03.00-6*

"A la memoria  
de mis abuelos:  
Marcial y María.  
... Dios los tenga  
a buen recaudo".

## AGRADECIMENTOS

Desejo agradecer profundamente a todos aqueles que de alguma maneira contribuíram para a realização deste trabalho. Em especial:

À minha família: Gloria, Pocho, Joel e Lito, pelo carinho e apoio que sempre souberam me dar quando precisei, apesar da distância.

À minha namorada Karim Ayma, pela paciência, amor e compreensão, e especialmente por ter sabido manter acesa "la luz de nuestro candil", durante esta longa odisséia.

Ao professor Tiaraju V. Wagner, por ter orientado excelentemente esta dissertação, partilhando comigo sua experiência e uma grande amizade.

Ao professor Juergen Rochol, pela proposta deste trabalho, e principalmente pelos comentários e sugestões ao mesmo, que ajudaram a enriquecer seu conteúdo.

Aos professores Ricardo A. L. Reis e Carlos Silva C., pelo apoio brindado, graças ao qual foi possível o prosseguimento do Curso de Mestrado.

Ao amigo Gustavo W. Kaiser, pela atitude sempre solícita para responder a minhas inúmeras perguntas acerca do seu trabalho e do "SOLO". Também, pela fé e alegria transmitidas através dos seus conselhos e seu bom humor.

Aos colegas e amigos do CPGCC, André Aita, Alvaro Terrazas, Jung Choi, Evandro Bender, Clairmont Borges, Luis F. da Silva, Marcus Kindel, Luis Pilger, José Güntzel e Reginaldo Tavares, pela grande amizade e solidariedade que demonstraram tanto dentro como fora do Curso.

E ao CNPq, pelo apoio financeiro.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS</b> . . . . .	<b>9</b>
<b>LISTA DE FIGURAS</b> . . . . .	<b>11</b>
<b>LISTA DE TABELAS</b> . . . . .	<b>18</b>
<b>RESUMO</b> . . . . .	<b>19</b>
<b>ABSTRACT</b> . . . . .	<b>20</b>
<b>1 INTRODUÇÃO</b> . . . . .	<b>21</b>
<b>2 SISTEMAS DE COMUTAÇÃO</b> . . . . .	<b>25</b>
<b>2.1 Visão Geral</b> . . . . .	25
<b>2.2 Definições</b> . . . . .	26
<b>2.3 Comutadores Digitais</b> . . . . .	26
<b>2.4 Modalidades de Comutação</b> . . . . .	28
2.4.1 Comutação de Mensagens . . . . .	30
2.4.2 Comutação de Circuitos . . . . .	30
2.4.3 Comutação de Pacotes . . . . .	32
2.4.3.1 Circuito Virtual . . . . .	33
2.4.3.2 Datagrama . . . . .	34
2.4.4 Comutação de Células . . . . .	34
<b>2.5 Componentes de um Sistema de Comutação</b> . . . . .	35
<b>3 COMUTAÇÃO DE CIRCUITOS</b> . . . . .	<b>38</b>
<b>3.1 Visão Geral</b> . . . . .	38
<b>3.2 Comutação Espacial</b> . . . . .	44
<b>3.3 Comutação Temporal</b> . . . . .	48
3.3.1 TDM Síncrona (STDM) . . . . .	48
3.3.1.1 <i>TDM Bus Switching</i> . . . . .	50
3.3.1.2 <i>Time Slot Interchange (TSI)</i> . . . . .	53

3.3.1.3	<i>Time Multiplexed Switching</i> (TMS) . . . . .	57
3.3.2	TDM Assíncrona (ATDM) . . . . .	59
3.3.2.1	<i>Time Division Multiple Access</i> (TDMA) . . . . .	60
<b>4</b>	<b>REDES CORPORATIVAS E1</b> . . . . .	<b>63</b>
4.1	Visão Geral . . . . .	63
4.2	A Hierarquia Digital E1 . . . . .	65
4.3	O Multiplexador E1 . . . . .	68
4.3.1	Descrição Funcional . . . . .	68
4.4	Comutação na Rede Corporativa E1 . . . . .	70
<b>5</b>	<b>COMUTADOR DIGITAL TDM E1</b> . . . . .	<b>72</b>
5.1	Arquitetura do Comutador . . . . .	72
5.1.1	Requerimentos . . . . .	73
5.1.2	Alternativas . . . . .	74
5.1.3	A Arquitetura . . . . .	76
5.2	Aspectos de Temporização . . . . .	78
5.2.1	Sincronização . . . . .	79
5.2.2	Armazenamento Elástico . . . . .	80
5.3	Aspectos de Programação . . . . .	82
5.3.1	A Memória de Conexões . . . . .	83
5.3.2	Canal Serial de Comunicação . . . . .	87
5.3.2.1	O Protocolo BSC . . . . .	88
5.4	Recomendações ITU-T consideradas . . . . .	91
5.4.1	Rec. G.702 - Hierarquias Digitais e Taxas de <i>Bit</i> associadas . . . . .	92
5.4.2	Rec. G.703 - Características Físicas e Elétricas das Interfaces . . . . .	92
5.4.3	Rec. G.704 - Estruturas dos Quadros Síncronos da Hierarquia Digital . . . . .	93
5.4.4	Rec. G.706 - Procedimentos relacionados com Sincronismo de Quadro e CRC, em estruturas básicas de Quadro . . . . .	94

5.4.5	Rec. G.736 - Características de um Equipamento Multiplex Digital Síncrono, operando a 2048 Kbit/s . . . . .	94
5.5	Outros Trabalhos . . . . .	95
6	<b>IMPLEMENTAÇÃO DO COMUTADOR DIGITAL . . . . .</b>	<b>97</b>
6.1	Geradores de Sinais . . . . .	97
6.1.1	Gerador de Sinal <i>Push</i> . . . . .	98
6.1.2	Gerador de Sinal <i>Save</i> . . . . .	98
6.1.3	Gerador de Sinal <i>MCLK</i> . . . . .	100
6.1.4	Gerador de Endereços . . . . .	101
6.2	Conversores . . . . .	104
6.2.1	Conversor Série a Paralelo . . . . .	104
6.2.2	Conversor Paralelo a Série . . . . .	106
6.3	Memórias . . . . .	107
6.3.1	Memória SFM . . . . .	108
6.3.2	Memória DSM . . . . .	110
6.3.3	Memória CM . . . . .	113
6.4	<i>Buffer</i> de Alinhamento . . . . .	114
6.5	Especificação do Controle . . . . .	120
7	<b>SIMULAÇÃO DO COMUTADOR DIGITAL . . . . .</b>	<b>125</b>
7.1	Comutação Local-Local . . . . .	126
7.2	Comutação Local-Remota . . . . .	130
7.3	Comutação Remota . . . . .	138
8	<b>CONCLUSÃO . . . . .</b>	<b>144</b>
	<b>ANEXO A-1 ROTINAS PARA O MÓDULO DE CONTROLE DO COMUTADOR DIGITAL . . . . .</b>	<b>147</b>
A-1.1	A Rotina <i>Start</i> . . . . .	147
A-1.2	A rotina <i>Ready</i> . . . . .	150

A-1.3	A Rotina <i>Setup</i> . . . . .	151
A-1.4	A Rotina <i>Comunic</i> . . . . .	153
A-1.5	A Rotina <i>Stop</i> . . . . .	158
A-1.6	A Rotina <i>Slave</i> . . . . .	162
ANEXO A-2	CONTEÚDO DA MEMÓRIA DE CONEXÕES . . .	164
ANEXO A-3	SUBCIRCUITOS DO MÓDULO MATRIC . . . . .	166
ANEXO A-4	SUBCIRCUITOS DOS MÓDULOS GERADORES DE SINAIS . . . . .	169
ANEXO A-5	SUBCIRCUITOS DO CONVERSOR SÉRIE A PA- RALELO . . . . .	175
ANEXO A-6	SUBCIRCUITOS DO CONVERSOR PARALELO A SÉRIE . . . . .	183
ANEXO A-7	SUBCIRCUITOS AUXILIARES I . . . . .	194
ANEXO A-8	SUBCIRCUITOS DA MEMÓRIA SFM . . . . .	200
ANEXO A-9	SUBCIRCUITOS DA MEMÓRIA DSM . . . . .	207
ANEXO A-10	SUBCIRCUITOS AUXILIARES II . . . . .	214
ANEXO A-11	SUBCIRCUITOS DO <i>BUFFER</i> DE ALINHAMENTO	224
	BIBLIOGRAFIA . . . . .	231

## LISTA DE ABREVIATURAS

A/D	- <i>Analog to Digital</i>
ABNT	- Associação Brasileira de Normas Técnicas
AMI	- <i>Alternate Mark Inversion Code</i>
ANSI	- <i>American National Standards Institute</i>
ASIC	- <i>Application Specific Integrated Circuit</i>
ATM	- <i>Asynchronous Transfer Mode</i>
B	- Canal tipo B
BCC	- <i>Block Check Character</i>
BSC	- <i>Byte Synchronous Communication</i>
CBR	- <i>Constant Bit Rate</i>
CCITT	- Comitê Consultativo Internacional de Telegrafia e Telefonia
CD	- <i>Carrier Detect</i>
CI	- Circuito Integrado
CMOS	- <i>Complementar Metal Oxid Semiconductor</i>
CRC	- <i>Cycle Redundancy Check</i>
CTS	- <i>Clear to Send</i>
D	- Canal tipo D
D/A	- Digital to Analog
E1	- Hierarquia Digital Européia de 2048 Kbit/s
E/S	- Entrada/Saída
FDDI	- <i>Fiber Distributed Data Interface</i>
FDM	- <i>Frequency Division Multiplexing</i>
FET	- <i>Field Effect Transistor</i>
FIFO	- <i>First In First Out</i>
HDB3	- <i>High Density Bipolar 3</i>
HDLC	- <i>High Level Data Link Control</i>
IEEE	- <i>Institute of Electrical and Electronics Engineers</i>
ISDN	- <i>Integrated Services Digital Network</i>
ITU-T	- <i>International Telecommunications Union</i>
LAN	- <i>Local Area Network</i>
LRC	- <i>Linear Redundancy Check</i>
LSB	- <i>Least Significant Bit</i>
MAN	- <i>Metropolitan Area Networks</i>
MSB	- <i>Most Significant Bit</i>
MUX	- Multiplexador
NRZ	- <i>No Return to Zero</i>
OSI	- <i>Open System Interconnection</i>
PBX	- <i>Private Branch Exchange</i>
PCM	- <i>Pulse Code Modulation</i>
ppm	- <i>part per million</i>
PPM	- <i>Pulse Position Modulation</i>
PROM	- <i>Programmable Read Only Memory</i>
RAM	- <i>Random Access Memory</i>

ROM	- <i>Read Only Memory</i>
RDI	- Rede Digital Integrada
RDSI	- Rede Digital de Serviços Integrados
RDSI-FE	- Rede Digital de Serviços Integrados de Faixa Estreita
RDSI-FL	- Rede Digital de Serviços Integrados de Faixa Larga
RTS	- <i>Request to Send</i>
SDLC	- <i>Synchronous Data Link Control</i>
SRAM	- <i>Static Random Access Memory</i>
T1	- Hierarquia Digital Americana de 1544 Kbit/s
TCP/IP	- <i>Transport Control Protocol / Internet Protocol</i>
TDM	- <i>Time Division Multiplexing</i>
TDMA	- <i>Time Division Multiple Access</i>
TMS	- <i>Time Multiplexed Switching</i>
TSI	- <i>Time Slot Interchange</i>
USART	- <i>Universal Synchronous Asynchronous Receiver Transmitter</i>
VBR	- <i>Variable Bit Rate</i>
VLSI	- <i>Very Large Scale of Integration</i>
WAN	- <i>Wide Area Networks</i>

## LISTA DE FIGURAS

FIGURA 2.1	- Diagrama de Tempo da Comutação de Mensagens. . . . .	31
FIGURA 2.2	- Diagrama de Tempo da Comutação de Circuitos. . . . .	32
FIGURA 2.3	- Diagrama de Tempo da Comutação de Pacotes. . . . .	33
FIGURA 2.4	- Elementos Básicos de um Sistema de Comutação. . . . .	36
FIGURA 3.1	- Rede Genérica. . . . .	39
FIGURA 3.2	- Modelo de Referência OSI. . . . .	42
FIGURA 3.3	- Comutador Espacial de Uma Etapa. . . . .	45
FIGURA 3.4	- Comutador Espacial de 3 Etapas. . . . .	46
FIGURA 3.5	- TDM Síncrona. . . . .	49
FIGURA 3.6	- Comutador por Barramento TDM. . . . .	51
FIGURA 3.7	- Controle do Barramento TDM. . . . .	52
FIGURA 3.8	- Operação de uma Unidade TSI. . . . .	53
FIGURA 3.9	- <i>Time Slot Interchange</i> . . . . .	54
FIGURA 3.10	- Escrita Aleatória e Leitura Seqüencial. . . . .	56
FIGURA 3.11	- Escrita Seqüencial e Leitura Aleatória. . . . .	57
FIGURA 3.12	- Rede STS. . . . .	58
FIGURA 3.13	- Rede TST. . . . .	59
FIGURA 3.14	- TDM Assíncrona. . . . .	60
FIGURA 3.15	- Estrutura de Quadro de um sistema TDMA. . . . .	61
FIGURA 4.1	- Rede Corporativa El ponto-a-ponto. . . . .	65
FIGURA 4.2	- Formato de Quadro El. . . . .	67
FIGURA 5.1	- Rede El. . . . .	73
FIGURA 5.2	- Multiplexadores El e Comutador Digital. . . . .	76
FIGURA 5.3	- Arquitetura do Comutador Digital. . . . .	77
FIGURA 5.4	- Sinais de Relógio para o Comutador. . . . .	79
FIGURA 5.5	- <i>Buffer</i> Elástico. . . . .	81
FIGURA 5.6	- Interface com o Comutador TDM. . . . .	82

FIGURA 5.7	- Descrição da Programação da Memória de Conexões. . . . .	85
FIGURA 5.8	- Estrutura Típica de Bloco no BSC. . . . .	89
FIGURA 5.9	- Transferência de Informações usando BSC. . . . .	90
FIGURA 5.10	- Blocos transmitidos em Transparência. . . . .	91
FIGURA 6.1	- Módulo gen-push. . . . .	98
FIGURA 6.2	- Simulação do Gerador de Sinal Push. . . . .	99
FIGURA 6.3	- Módulo gen-save. . . . .	99
FIGURA 6.4	- Simulação do Gerador de Sinal Save. . . . .	100
FIGURA 6.5	- Módulo gera-mclk. . . . .	101
FIGURA 6.6	- Simulação do Gerador de Sinal MCLK . . . . .	102
FIGURA 6.7	- Módulo mac. . . . .	102
FIGURA 6.8	- Simulação do Gerador de Endereços . . . . .	103
FIGURA 6.9	- Módulo b-sp-a. . . . .	104
FIGURA 6.10	- Diagrama de Blocos do Conversor Série a Paralelo. . . . .	105
FIGURA 6.11	- Módulo b-ps. . . . .	106
FIGURA 6.12	- Diagrama de Blocos do Conversor Paralelo a Série. . . . .	107
FIGURA 6.13	- Módulo b-sfm. . . . .	108
FIGURA 6.14	- Unidade de Memória Básica SFM. . . . .	109
FIGURA 6.15	- Simulação da Memória SFM. . . . .	110
FIGURA 6.16	- Módulo b-dsm. . . . .	111
FIGURA 6.17	- Módulo u-dsm. . . . .	112
FIGURA 6.18	- Simulação da Memória DSM. . . . .	113
FIGURA 6.19	- Memória romx. . . . .	114
FIGURA 6.20	- Módulo <i>align.buffer</i> . . . . .	115
FIGURA 6.21	- Buffer de Alinhamento. . . . .	117
FIGURA 6.22	- Simulação do Caso Típico. . . . .	118
FIGURA 6.23	- Simulação do Pior Caso. . . . .	119
FIGURA 6.24	- Diagrama de Estados. . . . .	121
FIGURA 6.25	- Diagrama de Blocos do Sistema de Comutação. . . . .	124

FIGURA 7.1	- Diagrama Esquemático do Comutador Digital. . . . .	126
FIGURA 7.2	- Simulação da Comutação Local-Local (vista geral). . . . .	128
FIGURA 7.3	- Entradas ao Comutador Digital, Comutação Local-Local. . .	131
FIGURA 7.4	- Saídas do Comutador Digital, Comutação Local-Local. . . .	132
FIGURA 7.5	- Simulação da Comutação Local-Remota (vista geral). . . .	134
FIGURA 7.6	- Entradas ao Comutador Digital, Comutação Local-Remota. .	136
FIGURA 7.7	- Saídas do Comutador Digital, Comutação Local-Remota. . .	137
FIGURA 7.8	- Simulação da Comutação Remota (vista geral). . . . .	139
FIGURA 7.9	- Entradas ao Comutador Digital, Comutação Remota. . . .	141
FIGURA 7.10	- Saídas do Comutador Digital, Comutação Remota. . . . .	143
FIGURA A-1.1	- Diagrama de Fluxo da Rotina <i>Start</i> . . . . .	148
FIGURA A-1.2	- Diagrama de Fluxo da Rotina <i>Ready (1/2)</i> . . . . .	151
FIGURA A-1.3	- Diagrama de Fluxo da Rotina <i>Ready (2/2)</i> . . . . .	152
FIGURA A-1.4	- Diagrama de Fluxo da Rotina <i>Setup</i> . . . . .	154
FIGURA A-1.5	- Diagrama de Fluxo da Rotina <i>Comunic</i> . . . . .	155
FIGURA A-1.6	- Diagrama de Fluxo da Rotina <i>Tx-Char</i> . . . . .	156
FIGURA A-1.7	- Diagrama de Fluxo da Rotina <i>Ext-Change</i> . . . . .	157
FIGURA A-1.8	- Diagrama de Fluxo da Rotina <i>Rx-Special</i> . . . . .	158
FIGURA A-1.9	- Diagrama de Fluxo da Rotina <i>Rx-Char (1/3)</i> . . . . .	159
FIGURA A-1.10	- Diagrama de Fluxo da Rotina <i>Rx-Char (2/3)</i> . . . . .	160
FIGURA A-1.11	- Diagrama de Fluxo da Rotina <i>Rx-Char (3/3)</i> . . . . .	161
FIGURA A-1.12	- Diagrama de Fluxo da Rotina <i>Stop</i> . . . . .	162
FIGURA A-1.13	- Diagrama de Fluxo da Rotina <i>Slave</i> . . . . .	163
FIGURA A-3.1	- Módulo MATRIC (1/2). . . . .	167
FIGURA A-3.2	- Módulo MATRIC (2/2). . . . .	168
FIGURA A-4.1	- Módulo gera-mclk. . . . .	170
FIGURA A-4.2	- Módulo gen-push. . . . .	170
FIGURA A-4.3	- Módulo gen-save. . . . .	171
FIGURA A-4.4	- Módulo mac (1/3). . . . .	172

FIGURA A-4.5	- Módulo mac (2/3).	173
FIGURA A-4.6	- Módulo mac (3/3).	174
FIGURA A-5.1	- Módulo b-sp-a (1/12).	176
FIGURA A-5.2	- Módulo b-sp-a (2/12).	176
FIGURA A-5.3	- Módulo b-sp-a (3/12).	177
FIGURA A-5.4	- Módulo b-sp-a (4/12).	177
FIGURA A-5.5	- Módulo b-sp-a (5/12).	178
FIGURA A-5.6	- Módulo b-sp-a (6/12).	178
FIGURA A-5.7	- Módulo b-sp-a (7/12).	179
FIGURA A-5.8	- Módulo b-sp-a (8/12).	179
FIGURA A-5.9	- Módulo b-sp-a (9/12).	180
FIGURA A-5.10	- Módulo b-sp-a (10/12).	180
FIGURA A-5.11	- Módulo b-sp-a (11/12).	181
FIGURA A-5.12	- Módulo b-sp-a (12/12).	181
FIGURA A-5.13	- Módulo s-p.	182
FIGURA A-6.1	- Módulo b-ps.	184
FIGURA A-6.2	- Módulo bloc-ps (etapa A, 1/8).	185
FIGURA A-6.3	- Módulo bloc-ps (etapa A, 2/8).	185
FIGURA A-6.4	- Módulo bloc-ps (etapa A, 3/8).	186
FIGURA A-6.5	- Módulo bloc-ps (etapa A, 4/8).	186
FIGURA A-6.6	- Módulo bloc-ps (etapa A, 5/8).	187
FIGURA A-6.7	- Módulo bloc-ps (etapa A, 6/8).	187
FIGURA A-6.8	- Módulo bloc-ps (etapa A, 7/8).	188
FIGURA A-6.9	- Módulo bloc-ps (etapa A, 8/8).	188
FIGURA A-6.10	- Módulo bloc-ps (etapa B, 1/3).	189
FIGURA A-6.11	- Módulo bloc-ps (etapa B, 2/3).	189
FIGURA A-6.12	- Módulo bloc-ps (etapa B, 3/3).	190
FIGURA A-6.13	- Módulo bloc-ps (etapa C).	190
FIGURA A-6.14	- Módulo cont-256 (1/3).	191

FIGURA A-6.15 – Módulo cont-256 (2/3).	192
FIGURA A-6.16 – Módulo cont-256 (3/3).	193
FIGURA A-7.1 – Módulo b-latch (1/8).	195
FIGURA A-7.2 – Módulo b-latch (2/8).	195
FIGURA A-7.3 – Módulo b-latch (3/8).	196
FIGURA A-7.4 – Módulo b-latch (4/8).	196
FIGURA A-7.5 – Módulo b-latch (5/8).	197
FIGURA A-7.6 – Módulo b-latch (6/8).	197
FIGURA A-7.7 – Módulo b-latch (7/8).	198
FIGURA A-7.8 – Módulo b-latch (8/8).	198
FIGURA A-7.9 – Módulo latch-8.	199
FIGURA A-8.1 – Módulo b-sfm (1/2).	201
FIGURA A-8.2 – Módulo b-sfm (2/2).	202
FIGURA A-8.3 – Módulo sfm (1/8).	203
FIGURA A-8.4 – Módulo sfm (2/8).	203
FIGURA A-8.5 – Módulo sfm (3/8).	204
FIGURA A-8.6 – Módulo sfm (4/8).	204
FIGURA A-8.7 – Módulo sfm (5/8).	205
FIGURA A-8.8 – Módulo sfm (6/8).	205
FIGURA A-8.9 – Módulo sfm (7/8).	206
FIGURA A-8.10 – Módulo sfm (8/8).	206
FIGURA A-9.1 – Módulo b-dsm (1/2).	208
FIGURA A-9.2 – Módulo b-dsm (2/2).	209
FIGURA A-9.3 – Módulo dsm (1/8).	210
FIGURA A-9.4 – Módulo dsm (2/8).	210
FIGURA A-9.5 – Módulo dsm (3/8).	211
FIGURA A-9.6 – Módulo dsm (4/8).	211
FIGURA A-9.7 – Módulo dsm (5/8).	212
FIGURA A-9.8 – Módulo dsm (6/8).	212

FIGURA A-9.9 – Módulo dsm (7/8). . . . .	213
FIGURA A-9.10 – Módulo dsm (8/8). . . . .	213
FIGURA A-10.1 – Módulo dec-6x64 (1/17). . . . .	215
FIGURA A-10.2 – Módulo dec-6x64 (2/17). . . . .	215
FIGURA A-10.3 – Módulo dec-6x64 (3/17). . . . .	216
FIGURA A-10.4 – Módulo dec-6x64 (4/17). . . . .	216
FIGURA A-10.5 – Módulo dec-6x64 (5/17). . . . .	217
FIGURA A-10.6 – Módulo dec-6x64 (6/17). . . . .	217
FIGURA A-10.7 – Módulo dec-6x64 (7/17). . . . .	218
FIGURA A-10.8 – Módulo dec-6x64 (8/17). . . . .	218
FIGURA A-10.9 – Módulo dec-6x64 (9/17). . . . .	219
FIGURA A-10.10– Módulo dec-6x64 (10/17). . . . .	219
FIGURA A-10.11– Módulo dec-6x64 (11/17). . . . .	220
FIGURA A-10.12– Módulo dec-6x64 (12/17). . . . .	220
FIGURA A-10.13– Módulo dec-6x64 (13/17). . . . .	221
FIGURA A-10.14– Módulo dec-6x64 (14/17). . . . .	221
FIGURA A-10.15– Módulo dec-6x64 (15/17). . . . .	222
FIGURA A-10.16– Módulo dec-6x64 (16/17). . . . .	222
FIGURA A-10.17– Módulo dec-6x64 (17/17). . . . .	223
FIGURA A-10.18– Módulo nor-6. . . . .	223
FIGURA A-11.1 – Módulo 256.celda (1/2). . . . .	225
FIGURA A-11.2 – Módulo 256.celda (2/2). . . . .	226
FIGURA A-11.3 – Módulo 64.celda (1/2). . . . .	227
FIGURA A-11.4 – Módulo 64.celda (2/2). . . . .	227
FIGURA A-11.5 – Módulo 8.celda. . . . .	228
FIGURA A-11.6 – Módulo celda. . . . .	228
FIGURA A-11.7 – Módulo add.gen.strt. . . . .	229
FIGURA A-11.8 – Módulo dec2x4. . . . .	229
FIGURA A-11.9 – Módulo dec2x4.en. . . . .	229

FIGURA A-11.10- Módulo dec3x8. . . . .	230
FIGURA A-11.11- Módulo dec3x8.en. . . . .	230

## LISTA DE TABELAS

TABELA 3.1	- Aplicações da Comutação de Circuitos . . . . .	44
TABELA 3.2	- Técnicas da Comutação de Circuitos . . . . .	44
TABELA 5.1	- Caracteres de Controle BSC . . . . .	89
TABELA 5.2	- Hierarquias Digitais ITU-T . . . . .	92
TABELA 5.3	- Descrição do Quadro E1. . . . .	93
TABELA 7.1	- Quadros de Entrada Comutação Local-Local . . . . .	129
TABELA 7.2	- Quadros de Saída Comutação Local-Local . . . . .	129
TABELA 7.3	- Quadros de Entrada Comutação Local-Remota . . . . .	135
TABELA 7.4	- Quadros de Saída Comutação Local-Remota . . . . .	135
TABELA 7.5	- Quadros de Entrada Comutação Remota . . . . .	140
TABELA 7.6	- Quadros de Saída Comutação Remota . . . . .	142
TABELA A-1.1	- Interrupções do 8051 . . . . .	147
TABELA A-1.2	- Interface do 8274 . . . . .	148
TABELA A-1.3	- Vetor de Interrupções do 8274 . . . . .	154

## RESUMO

Este trabalho consiste na especificação e desenvolvimento da arquitetura de um Comutador Digital para TDM Determinístico E1, visando sua posterior implementação em microeletrônica.

Inicialmente são apresentados os conceitos gerais sobre os Sistemas de Comutação, bem como das principais modalidades de comutação, seguidos de um estudo aprofundado da Comutação de Circuitos e suas técnicas mais utilizadas, devido a sua íntima relação com a multiplexação TDM e a hierarquia E1. Do mesmo modo, são descritas as características das Redes Corporativas E1 e dos multiplexadores E1, junto com as funções principais do Comutador dentro do ambiente de uma rede ponto-a-ponto.

Com base no estudo previo, é proposta a arquitetura de um Comutador Digital baseado em técnicas TSI capaz de fornecer funções de comutação local e remota entre os dispositivos conectados aos multiplexadores E1, que formam os nós de uma Rede Corporativa com controle centralizado. O projeto lógico e a simulação do Comutador Digital foram realizados dentro do *framework SOLO/Cadence*, usando a biblioteca de *Standard Cells* da tecnologia CMOS de  $1.2\mu$ . O simulador lógico *SILOS*, disponível no *SOLO/Cadence*, foi utilizado para validar a arquitetura proposta. Detalhes de implementação e resultados de simulação são apresentados. O módulo de controle do Comutador Digital é apenas especificado.

**PALAVRAS-CHAVE:** Microeletrônica, Comunicação de Dados, Sistemas de Comutação, Comutação de Circuitos, Redes Corporativas, Projeto de Sistemas Digitais.

**TITLE:** "DATA DIGITAL SWITCH FOR E1 DETERMINISTIC TDM, LOOKING TOWARD A MICROELECTRONICS IMPLEMENTATION."

## ABSTRACT

This work consists in the specification and development of a Digital Circuit Switch architecture for E1 Deterministic TDM, looking toward a future microelectronics implementation.

First, general concepts about Switching Systems and its basic elements, as well as the main kinds of switching are presented. Also, a meticulous study about Circuit Switching and its more used techniques is realized, because of the intrinsic relation with TDM and E1 hierarchy. In the same way, the characteristics of E1 Corporate Networks and E1 multiplexers are described, along with the main functions of the Digital Switch into an end-to-end network.

Taking into account the previous study, the architecture of a Digital Switch based on TSI techniques, is proposed. This architecture is able to perform local and remote switching between the devices connected to E1 multiplexers, which form the network nodes of an end-to-end Corporate Network. The logic design and the circuit simulation of the Digital Switch were performed within SOLO/Cadence Standard Cells design framework, using CMOS  $1.2\mu$  technology. The logic simulator SILOS was used to validate the proposed architecture. Implementation details and simulation results are presented. The Control module of the Digital Switch is only specified.

**KEY WORDS:** Microelectronics, Data Communications, Switching Systems, Circuit Switching, Corporate Networks, Digital Systems Design.

# 1 INTRODUÇÃO

Historicamente, as comunicações de voz e de dados têm sido manipuladas por redes de comunicação diferentes uma vez que estas redes de propósito específico têm sido grandemente satisfatórias no que concerne às aplicações até então existentes.

Outra razão fundamental para o uso de redes diferentes para voz e dados são as diferenças existentes nas características destes sinais. O sinal de voz acontece inerentemente em tempo real, visto que é um sinal analógico gerado por pessoas que falam entre si. A densidade espectral de energia e a atividade média do sinal de voz são bem conhecidas e consistentes entre diferentes interlocutores. Por outro lado, a maioria de dados são gerados por uma máquina e são de natureza digital. A taxa de *bit* e o comprimento das mensagens de dados, variam amplamente dependendo de uma aplicação particular. Pouco a pouco estas linhas divisórias estão desaparecendo como consequência das Técnicas de Digitalização de Voz, e aplicações tais como Correio Eletrônico, Fax e Multimídia.

Por outro lado, existe atualmente uma corrida para a fusão entre computadores e telefonia, dirigida pela necessidade de integrar aplicações de voz e dados. Em sistemas futuros, esta fusão estará compreendida por áudio de alta fidelidade, vídeo e TV, e eletrônica de consumo avançada [BUR 93]. A digitalização abriu novas possibilidades para arquitetura de sistemas e estruturas de rede, que podem finalmente conduzir a uma rede de telecomunicações universal para vários serviços de comunicação [SCH 82]. A Rede Digital de Serviços Integrados (RDSI), ou *Integrated Services Digital Network* (ISDN), está quase tornando-se uma realidade. Esta rede digital de cobertura mundial procura oferecer um amplo conjunto de serviços de voz e de dados baseados em canais de 64 Kbit/s (tipo B). O fato que uma única rede ofereça serviços de voz e de dados promete vários benefícios, por exemplo, os usuários ganham conveniência, flexibilidade e economia. Uma interface de usuário integrada permitirá que equipamentos terminais diferentes sejam deslocados e conectados da

mesma maneira que os aparelhos elétricos podem se conectar a uma tomada elétrica padronizada [CHE 88]. Além disso, os serviços podem ser personalizados segundo as necessidades individuais, sem preocupar-se com a compatibilidade das redes de propósito especial. Como resultado do compartilhamento de facilidades, a eficiência é incrementada, e as operações e a manutenção da rede são simplificadas. Estes itens se tornarão complexos rapidamente dentro de uma rede não-integrada, por causa da proliferação de novos serviços. Também a redução de custos de uma rede deve resultar em um menor preço pelo serviço aos usuários [CHE 88], [MAL 88].

Portanto, nos próximos anos, a fronteira entre as comunicações de voz e de dados será indistinguível na luta para suportar o amplo conjunto de novas aplicações multimídia, que são possíveis graças ao surgimento das tecnologias de processamento digital. Estas incluem: compressão de áudio e de vídeo, imagens de alta resolução, reconhecimento e síntese de voz. A oportunidade de integrar novos serviços, e de desenvolver novas aplicações multimídia, é o resultado do progresso em três áreas: padronização, interconexão de redes e microeletrônica.

Instituições como o *International Telecommunications Union* (ITU-T), e o *Institute of Electrical and Electronics Engineers* (IEEE), estão continuamente definindo padrões para permitir aos sistemas de comunicações interagir entre si, e também permitir o desenvolvimento de aplicações em uma plataforma comum [BUR 93]. Por outro lado, os recentes avanços na tecnologia VLSI (*Very Large Scale of Integration*) têm reduzido o tamanho, a complexidade e o custo da circuitaria necessária nas redes digitais [WES 93]. Segundo a perspectiva das aplicações multimídia, a técnica *full-custom* VLSI, é a mais promissora [ACK 94]. Cada uma das tecnologias de processamento digital confiam na microeletrônica para sua implementação com custos reduzidos. Segundo a perspectiva dos projetistas de microeletrônica (VLSI), a multimídia é uma área importante de aplicação que demanda requisitos extremos de tecnologia em termos de processos, arquitetura, projeto de circuitos e ferramentas de CAD [ACK 94].

Dentro de uma rede de comunicações podemos reconhecer 3 elementos básicos, a saber: os Terminais, o Sistema de Transmissão e os Comutadores [BEJ 91]. Os comutadores são os elementos menos visíveis para os usuários, porém, são os mais importantes em termos de oferecimento dos serviços disponíveis na rede. Eles formam o núcleo básico de um Sistema de Comutação.

Um Sistema de Comutação é um sistema baseado em *hardware* e *software* que fornece roteamento do fluxo de dados, bem como um certo grau de inteligência dentro das redes de comunicações [PER 93]. A comutação tem mudado dramaticamente nos últimos 15 anos. Sistemas de Comutação Digitais foram introduzidos nas redes de voz e de dados como consequência do avanço da tecnologia dos semicondutores, que resultou em uma importante redução de custos [SCH 82]. Os dois principais protagonistas da evolução nos comutadores foram: o microprocessador e os Circuitos Integrados de Aplicação Específica ou ASICs (*Application Specific Integrated Circuits*).

O desenvolvimento do microprocessador como uma poderosa e versátil ferramenta para o uso pelos engenheiros de projeto em transmissão e comutação digital, juntou-se aos circuitos integrados de aplicação específica, para oferecer *software* personalizado segundo as necessidades individuais dos sistemas [MEL 78], [PIT 79]. Atualmente, os Circuitos Integrados de Aplicação Específica e os microprocessadores estão muito difundidos dentro dos sistemas de comunicações (telefones celulares, sistemas de comutação, centrais PBX), fornecendo uma nova e ampla gama de serviços e benefícios.

Dentro deste contexto, este trabalho propõe uma arquitetura para um Comutador Digital para TDM, capaz de operar dentro do ambiente de uma Rede Corporativa ponto-a-ponto, cujos nós são formados por multiplexadores da hierarquia digital E1, tal e como são descritos em [KAI 95a]. A arquitetura foi definida e desenvolvida visando sua posterior implementação como ASIC, e considerando que o Comutador deve ser governado por um controle externo baseado em um microcontrolador.

Neste trabalho são inicialmente apresentados conceitos gerais sobre os Sistemas de Comutação e seus componentes básicos, bem como das principais modalidades com as quais são implementados. Em seguida, é feito um estudo aprofundado da Comutação de Circuitos e suas variantes, destacando sua intrínseca relação com as técnicas de multiplexação TDM, que são a base da hierarquia digital E1. Depois, é abordado o tema das Redes Corporativas E1 descrevendo, além de suas principais características, a hierarquia digital E1, os equipamentos de multiplexação E1, e as funções principais do Comutador dentro de um ambiente de rede ponto-a-ponto.

Finalmente, é proposta uma arquitetura para o Comutador Digital, fundamentando sua estrutura com base nos requerimentos da rede e nas recomendações ITU-T correspondentes à hierarquia digital E1. Na parte de implementação, é mostrado o mapeamento a nível de portas lógicas dos módulos da arquitetura proposta, assim como a simulação lógica dos circuitos, ambas tarefas realizadas dentro do ambiente de projeto *SOLO/Cadence* [CAD 88]. A implementação foi realizada usando o estilo de projeto *Standard Cells* e com tecnologia CMOS de  $1.2\mu$ . Como conclusão, são analisados os resultados obtidos e é especificado o controle do comutador visando sua futura implementação.

O uso de um ambiente profissional de projeto e os resultados das simulações devidamente interpretados asseguram que o projeto, como um todo, está correto e completo.

## 2 SISTEMAS DE COMUTAÇÃO

### 2.1 Visão Geral

A Transmissão Digital, denominação usada quando o sinal de mensagem transmitido é codificado como uma seqüência de pulsos elétricos discretos, não apenas têm tornado possível a comunicação de alta velocidade entre terminais oferecendo sinais discretos, senão também estimulou uma nova era na comutação, onde a rota através dos Nós de Comutação é igualmente projetada para suportar sinais digitais. O ambiente no qual os nós de comutação devem operar mudou desde uma relativamente pequena coleção de circuitos analógicos de voz até um grande número de circuitos com diferente largura de banda e um incremento na quantidade de serviços digitais disponíveis.

Embora utilizando uma tecnologia diferente, a comutação foi a primeira aplicação de processamento de informação elétrica da forma como a conhecemos atualmente. Os sistemas baseados em válvulas eletrônicas para operações lógicas foram aplicados em: computadores, transmissão usando *Time Division* ("tempo compartilhado") e comutação.

As primeiras redes de comutação de circuitos estabeleciam caminhos de transmissão contínuos e separados em 2 vias entre um par de canais. Para isto, eram utilizados diferentes dispositivos tais como: chaves eletro-mecânicas, comutadores *step-by-step*, chaves *cross-bar* ou dispositivos de estado sólido (semi-condutores). Esta técnica é conhecida como *Space Division* [JOE 83].

Depois da invenção do transistor, a lógica eletrônica para os controles da comutação, tornou-se viável. Com o uso dos denominados Controles Comuns para estabelecer conexões entre circuitos, foi observada a inerente função de memória nas redes de comutação. Os controles comuns armazenam, transferem, interpretam, traduzem e reúnem informação, a maioria em forma digital, com o propósito de

estabelecer conexões através de uma ou mais redes de comutação, fornecendo uma rota através dos comutadores para sinais de controle e sinais de voz.

As idéias de utilizar memórias eletrônicas volumosas com acesso aleatório, e lógica de semi-condutores foram combinadas para produzir o primeiro Controle Comum Programado em tempo real. Esta técnica, conhecida como Controle com Programa Armazenado (*Stored Program Control*), representou uma mudança fundamental nas funções de comutação. O controle por Programa Armazenado abriu as portas para a implementação de vários e inovadores serviços de usuário e para a uma grande simplificação da administração e manutenção dos sistemas de comutação digital.

## 2.2 Definições

*Comutação* é um meio de alocação de recursos tais como espaço, largura de banda ou tempo, a pessoas ou máquinas que usam tais recursos para comunicar-se a distância.

*Comutação Digital* significa geralmente que a informação sendo transmitida através do comutador (voz ou dados) está na forma digital.

*Circuito* ou *Conexão* é usado para destacar que a informação passa simultaneamente em ambas as direções e sem possibilidade de atraso no caso de sinais de voz.

*Call-Setup* é o procedimento de estabelecimento de uma conexão entre dois dispositivos conectados a uma rede.

## 2.3 Comutadores Digitais

Até 1960, os comutadores digitais mais utilizados eram os comutadores de mensagens. Em um *Comutador de Mensagens*, a informação é enviada a um ou

mais pontos intermediários onde a mensagem é armazenada (e possivelmente processada e regenerada) até que a informação seja solicitada, ou até que um enlace seja liberado para finalmente enviar a informação ao seu destino. Os Comutadores de Mensagens eram classicamente digitais devido a que a informação, por exemplo os dados telegráficos, era naturalmente digital e as velocidades de processamento requeridas eram baixas. Contudo, a maior parte da informação no mundo é transmitida não por comutadores de mensagens mas por comutadores de circuitos, e foi apenas na década de 60 que a digitalização dos comutadores de circuitos tornou-se viável.

Em um *Comutador de Circuitos* não é requerido nenhum tipo de armazenamento da forma como existe nos comutadores de mensagens. Pelo contrário, um caminho ponto-a-ponto é estabelecido através de comutadores intermediários, com um mínimo de atraso possível. Depois deste *call-setup*, o caminho é dedicado aos usuários até que o seu uso termine, quando então os enlaces utilizados são realocados a outros usuários. O domínio clássico da Comutação de Circuitos é a transmissão de voz, desde que um tempo consideravelmente longo de *call-setup* é usualmente aceitável antes que a comunicação tenha lugar. Contudo, o controle dos *call-setup* é complexo, e a digitalização de voz precisa de circuitos digitais de relativa alta velocidade.

No início dos anos 70, incrementou-se o uso da *Comutação de Pacotes*, uma nova forma de comutação digital que combina alguns aspectos da Comutação de Circuitos e da Comutação de Mensagens. Na Comutação de Pacotes a informação é enviada no modo *store-and-forward* semelhante à Comutação de Mensagens. Porém, no caso da Comutação de Pacotes, as mensagens são divididas em pacotes de 100 a 2000 bits, cada um dos quais é logo enviado na forma *store-and-forward* e reconstruído no destino para conformar a mensagem original. A vantagem fornecida pela Comutação de Pacotes é o tempo curto de ocupação das linhas de longa distância para o tráfego de dados do tipo *burst*, apenas a quantidade do tempo necessário para transmitir um pacote pequeno. Portanto, pode ser obtida uma eficiente utilização dos recursos. Um comutador de pacotes pode realizar algumas das funções de um

comutador de circuitos, por exemplo o estabelecimento de um caminho virtual ou canal lógico ponto-a-ponto, que pode ser alocado a um par de usuários determinados. Assim, para tráfego de dados interativo, o comutador de pacotes combina a vantagem do comutador de circuitos ao obter uma rota pre-estabelecida, e a vantagem do comutador de mensagens na utilização eficiente dos recursos disponíveis mediante o armazenamento e envio da informação. A Comutação de Pacotes foi desenvolvida para satisfazer as necessidades dos dados gerados pelos computadores e terminais, e tornou-se possível pelo surgimento do minicomputador.

Na década de 80, as Redes Locais (LAN), cujo domínio primário é uma área geográfica limitada, apareceram paralelamente como um novo tipo de comutador. Embora, possa ser novidade pensar em uma LAN como um comutador digital, não existe dúvida que esta realiza as funções de um comutador digital, pois aloca digitalmente recursos para a transmissão de informação digital. As funções de comutação estão de tal forma distribuídas através da rede que pode ser dito que o comutador é a totalidade da rede [LIS 83].

Atualmente, a Comutação de Circuitos e a Comutação de Pacotes continuam tendo um importante papel no âmbito das redes de comunicações. Os Serviços de Dados Comutados da Rede Telefônica Pública e a Rede Digital de Serviços Integrados de Faixa Estreita (RDSI-FE), são aplicações atuais de Comutação de Circuitos. Já a Comutação de Pacotes com seu primeiro protocolo o X.25, deu origem aos atuais *Frame Relay*, *Fast Packet* e *ATM (Asynchronous Transfer Mode)*, sendo este último a base da futura Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL).

## 2.4 Modalidades de Comutação

As modalidades de comutação de informação digital, devem atender às seguintes prerrogativas básicas [ROC 93]:

- Permitir diversos modos de conexão:

- Ponto-a-ponto
  - Multi-ponto
  - Difusão
- Atender taxas de *bits* elevadas e diversas para:
    - Canais de Voz
    - Canais de Dados
    - Canais de Imagens (multimídia)

Desde o início, duas técnicas de comutação vêm concorrendo na área de comunicações, em relação à utilização da largura de banda [INF 80]:

- Pré-Alocação
- Alocação Dinâmica

São sistemas de circuito comutado: a Rede Telefônica Pública Comutada, o Serviço de Telex e as Centrais PBX, neles é alocada uma quantidade fixa de largura de banda pelo tempo de duração de uma conexão. Os sistemas onde utiliza-se alocação dinâmica são: a Comutação de Mensagens e a Comutação de Pacotes.

Os avanços na tecnologia de computadores, com o conseqüente incremento da disponibilidade e a redução de custos de potência de processamento e de capacidade de armazenamento, têm tornado possível que os sistemas de comunicação de dados utilizem alocação dinâmica de forma mais eficiente que antes.

A seguir são descritas as três modalidades tradicionais que são utilizadas na comutação de dados que fluem ao longo de uma rede, desde a fonte até o destino. Também, faz-se menção da denominada *Comutação de Células* ou ATM.

### 2.4.1 Comutação de Mensagens

Consiste em armazenar os dados em uma estação próxima ao usuário e daí enviá-los, em alta velocidade, ao seu destino. Esta forma de serviço permite, mais facilmente, a adição de uma série de características à transmissão, tais como mudanças de velocidade e multiplexação de várias estações transmissoras. Além do mais, para tráfego intenso, a comutação de mensagens otimiza o uso dos meios de transmissão a longa distância, às custas de um acréscimo no custo dos comutadores. Esta modalidade é também conhecida como *store-and-forward* (armazene e envie) [TAN 88].

Esta modalidade envia a mensagem completa através da rede. A transmissão é realizada em etapas: as mensagens são transmitidas ao centro mais próximo, são armazenadas em *buffers*, e esperam em filas pela liberação da linha para serem transmitidas ao próximo centro. Para identificar as mensagens é colocado um cabeçalho em cada uma destas. Este cabeçalho é freqüentemente de dezenas de caracteres de comprimento, o que representa para mensagens pequenas um *overhead* considerável, e torna ineficientes as comunicações. A figura 2.1 ilustra as características desta técnica.

Devido a que os protocolos dos equipamentos de usuário possuem formatos de mensagens grandes e procedimentos de estabelecimento de conexões relativamente longos, existe um atraso demasiadamente grande. Portanto, para poder atender aos diferentes comprimentos das mensagens, os custos de armazenamento nos pontos intermediários são bastante elevados [TAR 84].

### 2.4.2 Comutação de Circuitos

Em uma rede de comutação de circuitos, a tarefa dos nós de comutação é estabelecer uma conexão direta desde um dispositivo até o ponto de destino para transmitir a informação por ele gerada. Após do estabelecimento da conexão, a informação é transmitida sem interrupção da conexão até o final, e só então o circuito

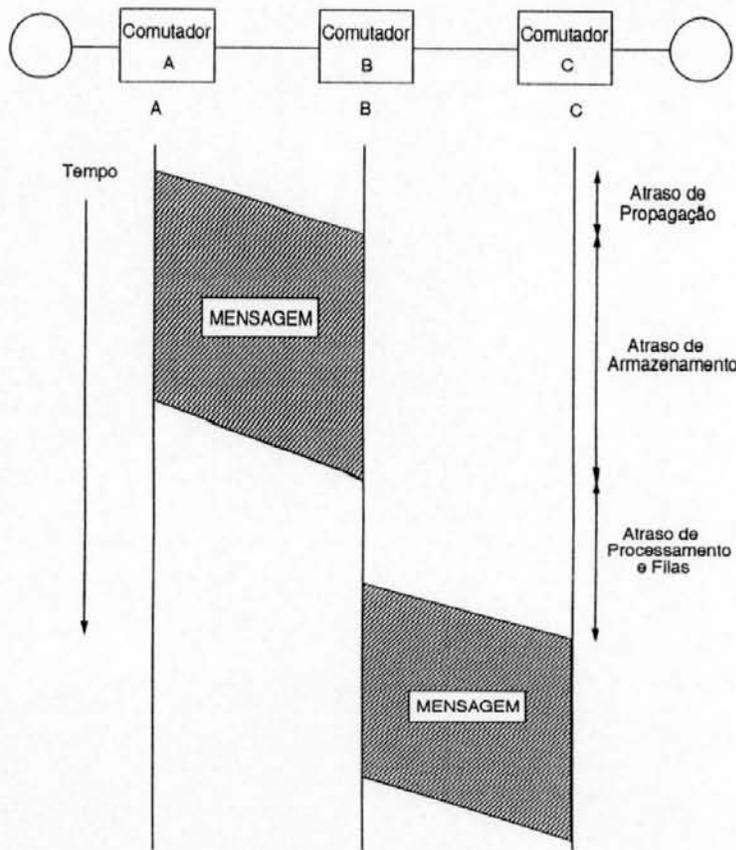


FIGURA 2.1 – Diagrama de Tempo da Comutação de Mensagens.

é desconectado e habilitado para uso por um outro par de usuários, tal como mostra a figura 2.2. Observa-se que nesta modalidade a largura de banda do canal em termos da taxa de transmissão, e o atraso, são constantes para aquela particular conexão.

Pelo fato que a utilização do canal é limitada pela natureza do tráfego transmitido, a comutação de circuitos é relativamente ineficiente para a transmissão de tráfego de dados interativo ou tipo *burst*. Além disso, a largura de banda não pode ser compartilhada pelos outros usuários.

As redes de circuito comutado utilizam métodos de *software* para estabelecer circuitos. Uma vez que o circuito é estabelecido, um recurso fixo da rede é alocado ao usuário, mas a rede não intervém na fase de transferência de dados durante o enlace. Isto é, a rede comuta via *hardware* o caminho entre o canal de entrada e o canal de saída de um circuito TDM, mas não precisa alocar estatisticamente os recursos. Em outras palavras: estabelece e esquece [INF 80]. Em uma situação

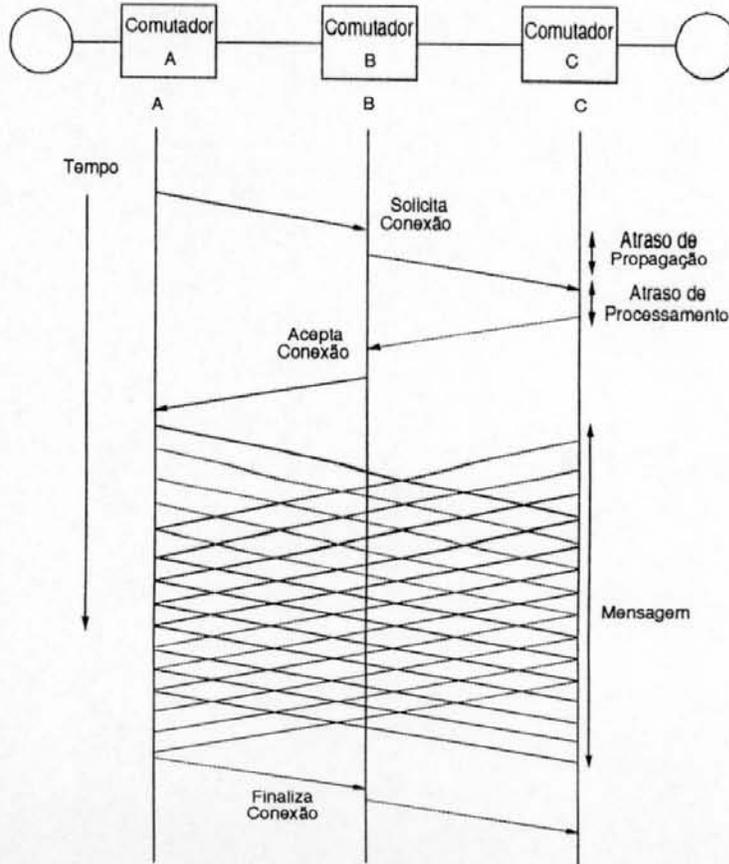


FIGURA 2.2 – Diagrama de Tempo da Comutação de Circuitos.

potencial de sobrecarga, uma rede de circuito comutado rejeitará o estabelecimento de novos enlaces, por tanto é inerentemente mais confiável.

### 2.4.3 Comutação de Pacotes

Contrastando com a modalidade anterior, a comutação de pacotes tenta multiplexar o uso dos circuitos de comunicação entre todos os usuários conectados. As mensagens são divididas tipicamente em segmentos curtos de comprimento fixo denominados pacotes, que são roteados independentemente ao seu destino usando processos *store-and-forward*. Cada pacote possui um cabeçalho ou *header* que contém informação de seqüência, que informa ao nó receptor como processá-lo, de tal forma que a mensagem possa ser reconstruída no destino. Os pacotes seguintes entre os mesmos usuários são tratados em forma independente. A característica básica desta modalidade é que a largura de banda do canal de dados entre

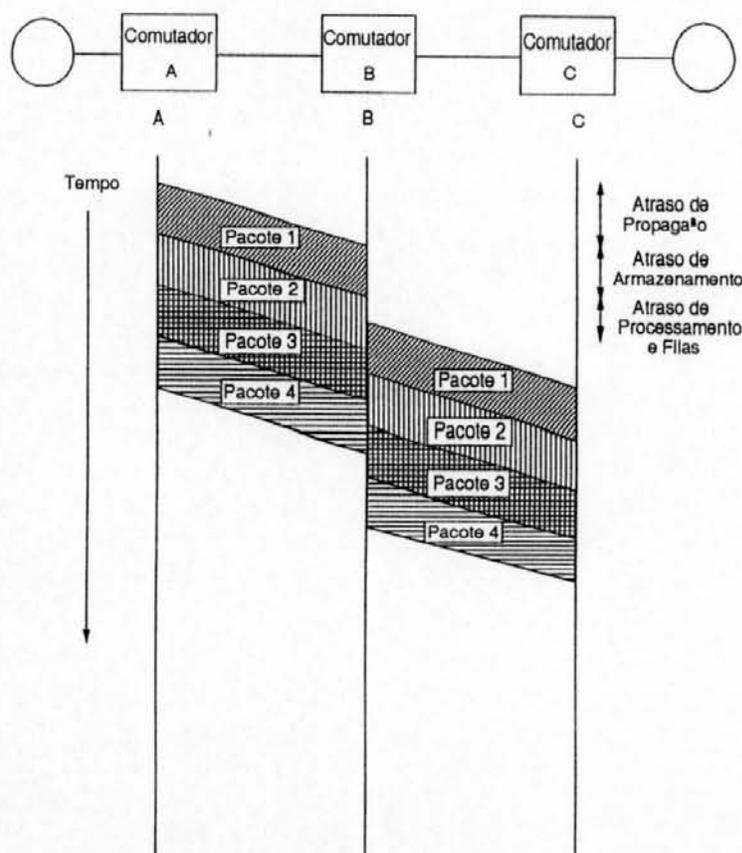


FIGURA 2.3 – Diagrama de Tempo da Comutação de Pacotes.

dois usuários não é constante, mas variável segundo o requerimento dos usuários. A figura 2.3 mostra esta técnica.

A comutação de pacotes é uma reaplicação das técnicas básicas de alocação dinâmica, e foi desenvolvida para atender mensagens de comprimento variável, simplificar o uso de *buffers* e melhorar o desempenho pelo atraso. Entre suas principais propriedades estão uma alta disponibilidade de caminhos alternados, baixa taxa de erros, atraso e largura de banda variáveis e cabeçalho em cada pacote. Possui duas variantes: Circuito Virtual e Datagrama.

#### 2.4.3.1 Circuito Virtual

Ao serem iniciados os procedimentos de conexão, os dispositivos trocam uma série de informações sobre a conexão, após o que são compostas tabelas nos centros de comutação mais próximos da fonte e do destino e, a partir de então,

todos os dados subseqüentes são dirigidos automaticamente entre ambos. Em esses sistemas, os dados sempre seguem uma mesma trajetória ("Direcionamento Fixo").

#### 2.4.3.2 Datagrama

Nesta modalidade de serviço, cada pacote contém informações completas para seu direcionamento, de modo que não é necessário estabelecer um circuito virtual na rede. Os dados podem ser conduzidos por trajetórias diferentes, dependendo da disponibilidade instantânea de rotas específicas ("Funcionamento Adaptativo"). Quando os dados podem seguir por diferentes trajetórias, pode acontecer, que um pacote chegue primeiro que um seu antecessor. Uma vez que um pacote na modalidade Datagrama tenha sido enviado, o emissor pode ou não ser informado de sua entrega. Esta abordagem é mais adequada para transmissões curtas.

#### 2.4.4 Comutação de Células

O Modo de Transferência Assíncrono (ATM) é uma técnica de comutação para sinais de banda larga, e com freqüência é designado como o padrão para a Comutação de Células ou *Fast Packet*. No mais puro sentido, a comutação ATM assemelha-se à verdadeira definição de *fast packet* devido a que manipula todos os tipos de tráfego através de comutação rápida de pacotes, reduzindo o processamento de protocolos e utilizando multiplexação estatística. ATM é basicamente uma técnica orientada à conexão, isto é, usa caminhos virtuais para transmitir a informação dos usuários, e pode transportar serviços que precisam ou não de conexões a taxas de *bit* constantes (CBR) ou taxas de *bit* variáveis (VBR).

ATM é uma tecnologia assíncrona porque as células transmitidas pelo usuário não são necessariamente periódicas; é projetada para fornecer o modo de transferência para serviços da RDSI-FL, e constitui-se na primeira tecnologia que fornece realmente largura de banda sob demanda, admitindo múltiplos usuários e partilhando eficientemente a largura de banda para otimizar os recursos da rede.

ATM é o primeiro padrão de pacotes que suporta velocidades superiores a T1/E1 e T3/E3, para manipular voz, dados, vídeo e imagens.

ATM proporciona conexões múltiplas de circuitos virtuais comutados aos usuários através de um único acesso à rede. Isto faz com que ATM seja a primeira tecnologia que combina efetivamente os benefícios da comutação de circuitos e da comutação de pacotes.

## 2.5 Componentes de um Sistema de Comutação

Em termos gerais, dentro de cada equipamento associado com qualquer sistema de comutação, pode-se distinguir módulos que realizam pelo menos uma das seguintes funções:

- Sinalização
- Controle
- Comutação

Os elementos básicos que constituem um Sistema de Comutação são mostrados na figura 2.4.

A função do *módulo de sinalização* é monitorar a atividade das linhas de entrada do comutador e enviar o *status* apropriado ou a informação de controle ao módulo de controle do sistema. O módulo de sinalização também é encarregado de colocar sinais de controle dentro das linhas de saída do comutador, sob o comando do módulo de controle.

O *módulo de controle* processa a informação de sinalização que chega, e a partir dela, configura e estabelece as conexões apropriadas. O controle de um enlace de comunicação entre dois dispositivos envolve várias funções de sinalização e supervisão para o estabelecimento, alteração, monitoração e finalização da conexão. Tanto para os sistemas de comunicação de voz como de dados, a implementação

da sinalização e supervisão não é um problema trivial, pelo contrário, é um difícil objetivo de projeto para as arquiteturas dos sistemas [BAX 82].

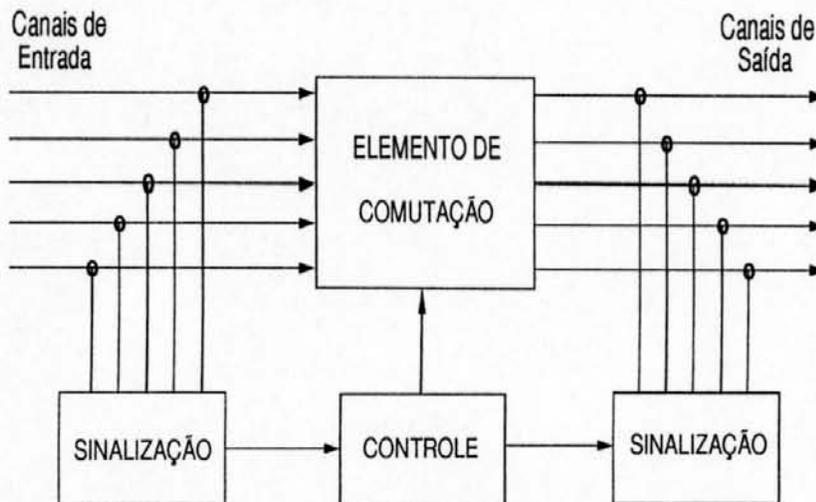


FIGURA 2.4 – Elementos Básicos de um Sistema de Comutação.

Os mecanismos para transmitir a sinalização e a supervisão envolvem duas formas: a inclusão da informação de sinalização junto com os sinais de voz ou de dados, denominada sinalização *In-Band*, ou a transmissão da informação de sinalização em um canal de comunicações separado, denominada sinalização *Out-of-Band*.

A sinalização *In-Band* tem a vantagem de usar o mesmo canal para ambos os propósitos e portanto, reduz o número de canais a serem usados. Contudo, esta estratégia complica a detecção de sinais de sinalização e supervisão, pois requer mecanismos para distinguir entre dados e informação de sinalização. Já a sinalização *Out-of-Band* é geralmente preferida, mas em sistemas de custo reduzido, pois o uso de um segundo canal é proibitivo.

A função de comutação é fornecida por um *elemento de comutação*. Este elemento de comutação, normalmente referido como *cross-point*, é um dispositivo que habilita uma conexão física determinada para estabelecer um caminho de transmissão para sinais de voz ou dados. A forma mais simples de um elemento de comutação é um dispositivo de contato metálico, um relé eletromecânico por exemplo, que ao ser instruído conecta os extremos de dois condutores.

Como a tecnologia tem amadurecido desde o relé normal até os circuitos integrados digitais, as opções disponíveis para o elemento de comutação são amplas. A escolha do elemento de comutação determina o modo de transmissão e portanto, a configuração geral da comutação. De fato, o elemento de comutação é o resultado do conceito de comutação selecionado para uma implementação particular.

## 3 COMUTAÇÃO DE CIRCUITOS

### 3.1 Visão Geral

A comunicação via Comutação de Circuitos, originou-se na Rede Telefônica Pública. Os sinais de voz ou dados são transmitidos no nível da camada física. Os sinais não são processados, apenas conduzidos através da rede sem importar seu conteúdo.

Atualmente, os computadores precisam "conversar" entre eles da mesma forma. Os enlaces de dados que utilizam comutação de circuitos funcionam mediante o estabelecimento de um caminho de comunicação dedicado entre duas estações ou terminais, mas a transmissão não é restrita a voz. Um caminho ponto-a-ponto é estabelecido como um circuito real e permanece ativo até que todos os dados sejam transmitidos. Deve-se entender que este caminho é uma seqüência de enlaces conectados entre os nós de uma rede. Uma vez concluída a comunicação, o circuito é desconectado. Cada enlace posterior é estabelecido e desconectado da mesma maneira. Nota-se que, a largura de banda alocada a cada enlace é dedicada até que todos os dados sejam transmitidos e recebidos. A Comutação de Circuitos é a tecnologia ideal para o tráfego que necessita largura de banda constante.

A Comutação de Circuitos possui 3 etapas:

- Estabelecimento do Circuito de Comunicação
- Transferência de Sinais
- Desconexão do Circuito

O estabelecimento do circuito ou *call-setup* implica que, antes que qualquer sinal seja transmitido, deve ser estabelecido um enlace ou circuito ponto-a-ponto (estação-a-estação). O seguinte exemplo usa como referência a figura 3.1,

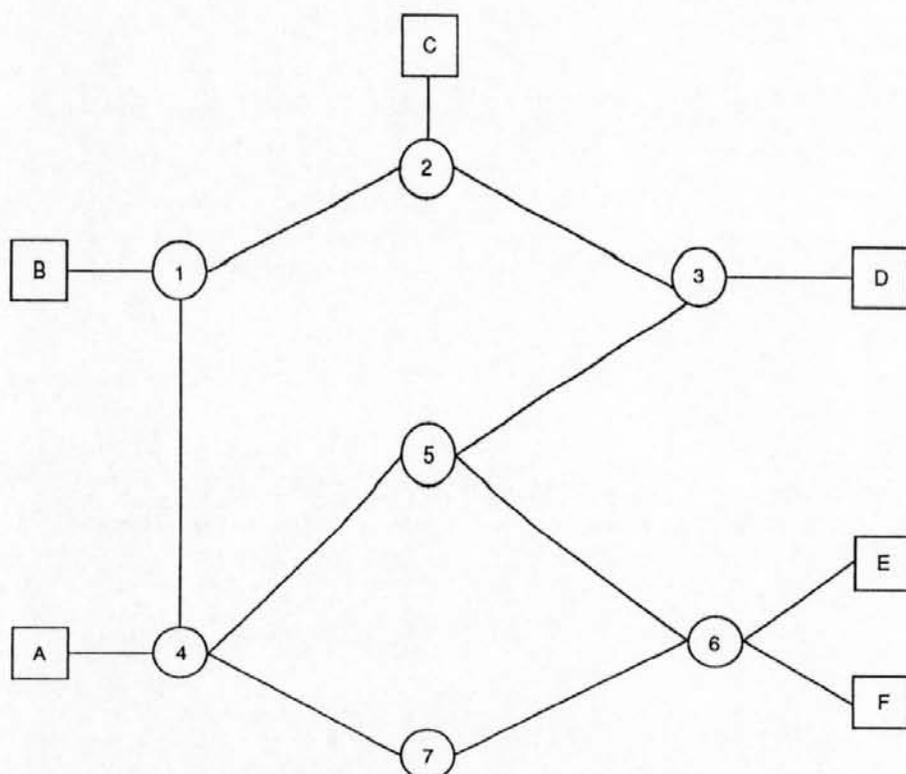


FIGURA 3.1 – Rede Genérica.

na qual se observa uma rede genérica com varios nós de comutação. A estação A envia um sinal de requerimento ao nó 4, solicitando uma conexão com a estação E. Tipicamente, o enlace desde A até o nó 4 é uma linha dedicada que já existe. O nó 4 deve encontrar o trecho seguinte que conduza ao nó 6. Baseado na informação de roteamento e nas medidas de disponibilidade, o nó 4 seleciona o enlace ao nó 5, ocupa um canal livre desta linha, usando FDM ou TDM, e envia uma mensagem de requerimento de conexão ao nó E. Deste modo, um caminho dedicado foi estabelecido desde A através dos nós 4 e 5. Agora o nó 5 dedica um canal até o nó 6 e enlaça internamente esse canal com o canal que chega do nó 4. O nó 6 completa a conexão com E. Para completar a conexão, realiza-se um teste para determinar se a estação E está ocupada ou se está pronta para aceitar a conexão.

A transferência de sinais acontece depois de ter sido estabelecido o circuito de enlace; os sinais podem ser transmitidos desde a estação A até a estação E através da rede. Os sinais transmitidos podem ser de voz analógica, voz digital ou dados binários, dependendo da natureza da rede. Com o avanço tecnológico, redes digitais integradas estão sendo o meio mais comum de comunicação. Por este motivo, o uso

da transmissão digital para voz e dados é cada vez mais predominante. Geralmente, a conexão é *full-duplex*, isto é, os dados podem ser transmitidos simultaneamente nos dois sentidos.

A desconexão do circuito significa que, logo após o período de transferência de dados, a conexão é interrompida, usualmente pela ação de uma das duas estações. Os sinais de comando devem ser propagados aos nós 4, 5 e 6 para liberar os recursos alocados.

Observa-se, que o caminho de conexão na Comutação de Circuitos é estabelecido antes que a transmissão de dados se inicie. Portanto, a capacidade do canal deve ser reservada entre cada par de nós no caminho e cada nó deve ter capacidade de comutação interna disponível para administrar a conexão requerida. Os comutadores devem possuir um certo nível de inteligência para estabelecer os caminhos de conexão e para fornecer uma rota através da rede, considerando a disponibilidade dos canais livres no momento.

A Comutação de Circuitos pode ser muito ineficiente devido ao fato do canal estar disponível durante todo o tempo de interconexão, ainda que não existam dados sendo transmitidos. Para uma conexão de sinais de voz, a percentagem de utilização do canal pode ser elevada, mas não atinge 100%. Em uma conexão terminal-a-computador, a capacidade do canal pode estar sendo desperdiçada a maior parte do tempo da conexão, devido às pausas naturais do uso. As comunicações entre computadores se caracterizam por enlaces de longa duração com média baixa, mas com instantes de muito tráfego [BUD 84]. Em termos de desempenho, existe um atraso prévio à transferência de dados devido ao período de estabelecimento do circuito. Contudo, uma vez que o circuito é estabelecido, a rede é efetivamente transparente para os usuários. Os dados são transmitidos a uma velocidade fixa sem mais atraso que o produzido pela propagação dos sinais através dos enlaces de transmissão. Portanto, o atraso em cada nó é mínimo [STA 92].

A principal consideração no projeto de um comutador de circuitos é o atraso. A redução do atraso a milisegundos pode significar um grande rendimento e

ganhos ao serviço dos fornecedores de uma rede de voz ou de dados, especialmente quando as aplicações dos usuários são sensíveis aos atrasos [SPO 93].

A Comutação de Circuitos ainda permanece como o tipo mais comum de Serviço Público de Dados Comutados. A comutação de circuitos de dados demorou mais para emergir no mercado das comunicações se comparada com a comutação de circuitos de voz, primeiramente porque são necessárias inteligência adicional e capacidade de digitalização em cada comutador de dados, para poder processar os enlaces de alta velocidade que são característicos na comunicação de dados. Também, porque as redes de comunicação de dados devem fornecer muito mais largura de banda para evitar o bloqueio. Este acontece quando toda a largura de banda disponível entre dois pontos da rede está sendo utilizada e algumas solicitações de conexão devem esperar até que haja largura de banda disponível para o circuito. Observemos que os equipamentos de comunicação de dados não simpatizam com sinais de ocupado ou bloqueio, especialmente quando a rede transporta informações valendo milhares de dólares por segundo.

Por outro lado, nos Estados Unidos, a Comutação de Circuitos de dados tem sido usada historicamente como suporte para os serviços de linha privada, e ainda permanece como a melhor opção viável considerando o custo para usuários de redes privadas, com serviços de dados comutados a 56 kbit/s (DS0) oferecidos desde 5 a 25 centavos por minuto, dependendo da hora do dia, tempo de uso, velocidade, taxa de erros, distância a transmitir, e outros fatores de confiabilidade. O serviço de Comutação de Circuitos oferecido anteriormente está baseado em técnicas TDM e fornece transporte de dados independente de protocolos. Os usuários acessam um serviço de circuito comutado mediante um canal dedicado conectado a um comutador de circuitos, o qual fornece a comunicação com outros comutadores dentro de uma rede comutada. Devido a que as redes de voz foram os usuários iniciais da Comutação de Circuitos, isto explica o fato do tráfego de voz não ser apenas compatível, mas ideal para as redes de circuito comutado.

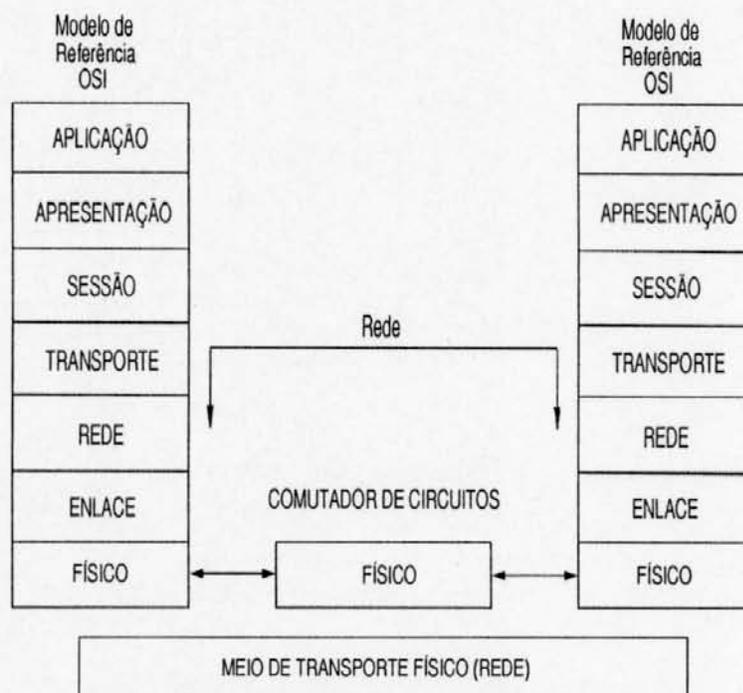


FIGURA 3.2 – Modelo de Referência OSI.

Contudo, o usuário de equipamentos de dados precisa de 3 tipos de circuitos para uma conexão:

- Circuito de Dados
- Circuito de Sinalização
- Circuito de Gerenciamento de Conexões

Os procedimentos de chamada são os mesmos para sinais de voz e sinais de dados, visto que a Comutação de Circuitos opera unicamente na primeira camada do Modelo de Referência OSI, ver figura 3.2, fornecendo métodos inteligentes de acesso ao meio físico.

A Comutação de Circuitos possui uma lista de requerimentos:

1. Estabelecer, manter, e finalizar requerimentos de conexão.
2. Fornecer transmissão *full-duplex* transparente.
3. Limitar o atraso a valores aceitáveis para conexões de voz ( $\leq 0.5s$ ).

4. Fornecer qualidade adequada para conexões de voz.
5. Ter probabilidade de bloqueio limitada.

Do anterior concluímos que os sinais de voz devem ser transmitidos virtualmente sem atraso e sem variações de volume e frequência. A taxa de transmissão dos sinais deve ser mantida constante, devido a que a transmissão e recepção ocorrem simultaneamente. Além disso, a qualidade do sinal recebido deve ser suficientemente alta para não comprometer a inteligibilidade. O último requerimento, o limite na probabilidade de bloqueio, refere-se à capacidade de um comutador para estabelecer os enlaces solicitados.

Dentro do mundo digital as ineficiências da Comutação de Circuitos são mais aparentes que reais, sendo que esta técnica é uma opção atrativa para LANs e WANs. A característica mais interessante é sua *transparência*. Uma vez que o circuito é estabelecido, ele aparece como uma conexão direta entre duas estações, e não é necessário contar com lógica de rede especial nas estações. Além disso, a Comutação de Circuitos evita os requerimentos de roteamento complexo, controle de fluxo e controle de erros que são próprios das redes de pacotes [STA 92].

Atualmente, algumas aplicações que geram tráfego em blocos de dados, e que requerem largura de banda disponível em períodos de tempo predefinidos, usam Comutação de Circuitos de alta velocidade como uma solução ideal, embora esta pareça ser uma solução a curto prazo para mais tarde dar lugar a técnicas mais avançadas como *Packet e Cell Switching*. Mas, enquanto estes serviços estejam disponíveis para a grande maioria, a Comutação de Circuitos oferece custos reduzidos e melhoras na qualidade do serviço, contrastando com as linhas privadas. As principais aplicações da Comutação de Circuitos se enumeram na tabela 3.1.

No Brasil, o uso da Rede Telefônica Pública Comutada foi autorizado desde maio de 1981, e foi adotada a hierarquia digital européia E1, sendo que a Telebrás editou através de suas "Práticas", os aspectos particulares de interesse para

TABELA 3.1 – Aplicações da Comutação de Circuitos

Voz	Dados
Rede Telefônica Pública	RDSI-FE
Centrais PBX	Comutadores de Dados
WAN privadas	

TABELA 3.2 – Técnicas da Comutação de Circuitos

Comutação Espacial ( <i>Space-Division</i> )
Comutação Temporal ( <i>Time-Division</i> )
<i>TDM Bus Switching</i>
<i>Time Slot Interchange (TSI)</i>
<i>Time Multiplexed Switching (TMS)</i>
<i>Time Division Multiple Access (TDMA)</i>

a Rede Pública de Comunicações [DOP 94]. A seguir serão descritas as principais técnicas da Comutação de Circuitos. Na tabela 3.2 estão resumidas tais técnicas.

### 3.2 Comutação Espacial

Na comutação por Divisão no Espaço ou Comutação Espacial, um simples caminho físico é estabelecido entre as entidades envolvidas no enlace de comunicação. Portanto, um comutador espacial é aquele cujos caminhos de sinal configurados estão separados fisicamente uns de outros. Cada conexão precisa do estabelecimento de um caminho físico através do comutador que é dedicado somente à transferência de sinais entre os pontos finais. O bloco básico de um comutador espacial é um *cross-point* metálico ou um elemento semicondutor (transistor FET), que pode ser habilitado ou desabilitado pelo módulo de controle do comutador.

A figura 3.3, mostra uma matriz *cross-bar* simples com N linhas *full-duplex* de Entrada/Saída (E/S). A matriz tem N entradas e N saídas, cada dispositivo conecta-se à matriz via uma linha de entrada e uma linha de saída. A interconexão é possível entre duas linhas pela habilitação do *cross-point* apropriado. Para evitar

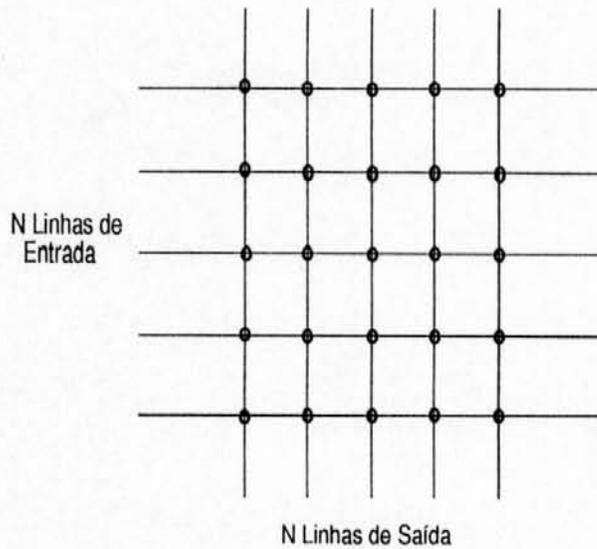


FIGURA 3.3 – Comutador Espacial de Uma Etapa.

conexões duplas, o comutador e suas linhas de controle devem ser organizadas de tal forma que apenas um único *cross-point* possa ser fechado de cada vez, em qualquer caminho horizontal ou vertical. Observa-se também, que são requeridos  $N^2$  *cross-points* no total. O comutador *cross-bar* apresenta as seguintes limitações:

- O número de *cross-points* cresce segundo o valor de  $N^2$ . Para valores grandes de  $N$ , resulta em uma alternativa de custo elevado e com uma grande carga capacitiva em qualquer caminho de mensagens.
- A perda de um *cross-point* impede a conexão entre os dois dispositivos envolvidos.
- Os *cross-points* são utilizados de forma ineficiente, pois no máximo  $N$  do total de  $N^2$  trabalham.

Para superar estas limitações, são utilizados comutadores de múltiplas etapas. As  $N$  linhas de entrada são divididas em  $N/n$  grupos de  $n$  linhas. Cada grupo de linhas entra em uma matriz da primeira etapa. As saídas das matrizes da primeira etapa constituem-se em entradas para o grupo de matrizes da segunda etapa, e assim por diante. A última etapa tem  $N$  saídas, portanto, cada dispositivo conecta sua linha de entrada à primeira etapa e sua linha de saída à última etapa. A figura 3.4 mostra uma rede genérica de comutadores de 3 etapas.

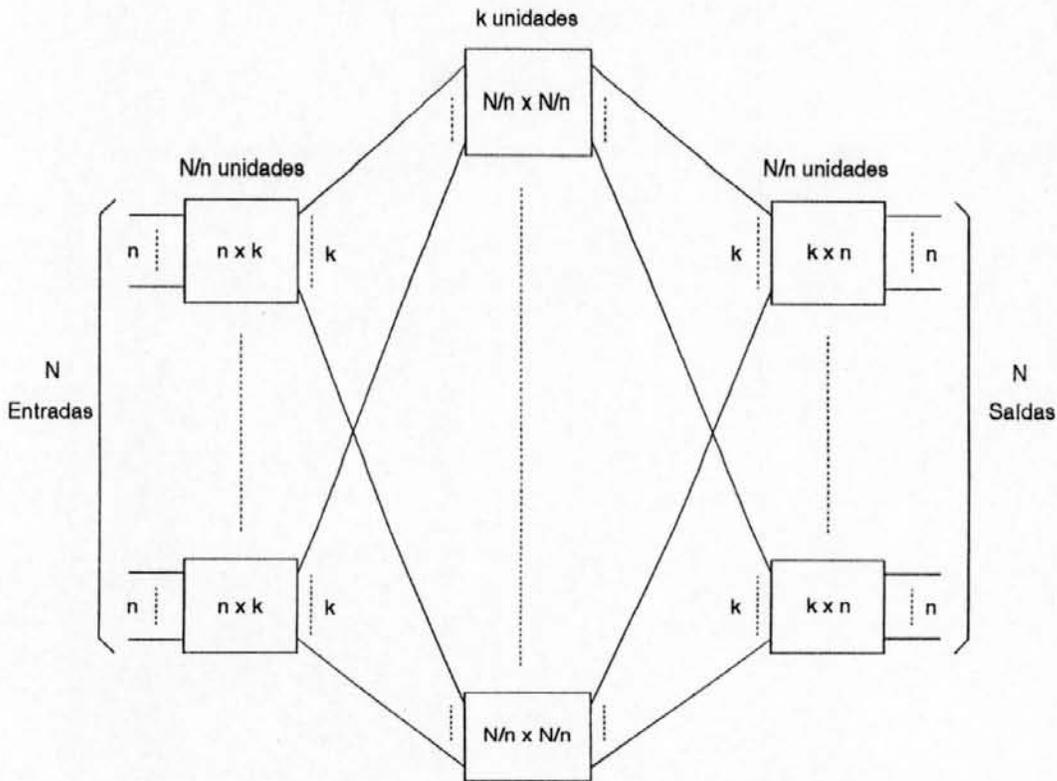


FIGURA 3.4 – Comutador Espacial de 3 Etapas.

O comutador de múltiplas etapas possui várias vantagens com relação a uma matriz simples:

- O número de *cross-points* é reduzido drasticamente.
- A utilização dos *cross-points* é incrementada.
- Possui maior confiabilidade, visto que existe mais de um caminho através do comutador para conectar dois extremos.

A principal desvantagem desta estratégia é a necessidade de um esquema de controle mais complexo. Uma outra consideração a ser levada em conta acerca dos comutadores de múltiplas etapas é que pode existir bloqueio. Isto não acontece com os comutadores de uma única etapa porque sempre existe um caminho disponível para conectar uma entrada a uma saída [STA 92].

Nas comunicações de dados que utilizam comutadores espaciais, um compromisso deve ser considerado entre a alocação de mais largura de banda no canal para obter a transparência do comutador à velocidade, e o código utilizado para o

tráfego dos dados. É importante destacar que em sistemas de comutação que usam *cross-points* metálicos, a transmissão satisfatória está relacionada ao comprimento dos caminhos através dos comutadores [BEL 90a]. Usando modulação por Posição de Pulso (PPM), a forma do sinal de dados digitais pode ser preservado através do comutador de dados, enquanto a informação de controle é transportada pelo mesmo sinal. As características do comutador de dados espacial podem ser utilizadas às custas da largura de banda e às custas de uma capacidade de canal sub-utilizada para obter maior velocidade e comutação de dados independente do protocolo.

O sinal de voz foi tradicionalmente transportado por meio de canais ou circuitos dedicados durante o tempo da ligação. A principal razão para isto, é que a voz usa efetivamente a capacidade do canal, e porque a voz é normalmente transmitida com uma largura de banda de 4KHz. As comunicações telefônicas também combinam a informação de sinalização e supervisão junto com a informação de voz. A informação de sinalização é codificada de várias maneiras, incluindo a presença ou ausência de corrente, presença de determinados tons e a presença de vários níveis de voltagem ou polaridade de voltagem. O gerenciamento de uma conexão de voz está associada exclusivamente com o estabelecimento e a finalização da mesma. Assim, não existe controle ou processamento feito enquanto dura a conexão.

Os comutadores tipo *cross-point* têm sido utilizados principalmente em sistemas de comutação de pequena escala (menos de 50 linhas), por serem mais econômicos. Por exemplo em Concentradores Locais e Centrais PBX pequenas [OWE 82].

Observe que, a comutação espacial pode ser encarada como uma simples extensão das possibilidades de comutação no espaço dos multiplexadores de entrada de um nó de comutação sem uma preocupação conceitual com a unidade que está sendo comutada, como é o caso da comutação temporal que será descrita a seguir.

### 3.3 Comutação Temporal

Com certeza, o método mais eficiente de separação de canais ou usuários em um canal digital de comunicações comum, é garantindo que eles transmitam em tempos diferentes. Esta técnica é conhecida como *Time Division Multiplexing* (TDM) ou Multiplexação por Divisão no Tempo, e possui algumas variantes.

#### 3.3.1 TDM Síncrona (STDM)

Freqüentemente, em um enlace de comunicação ponto-a-ponto, precisamos compartilhar o meio físico por várias estações de usuário. Quando isto ocorre, é apropriado utilizar a técnica denominada *Synchronous Time Division Multiplexing* ou TDM síncrona, que possibilita que vários sinais possam ser transportados por um único caminho físico, intercalando-se porções de cada sinal durante o tempo.

A técnica TDM síncrona permite que vários canais de baixa velocidade compartilhem uma linha ou canal de alta velocidade. Os canais a serem multiplexados são denominados *Canais* ou *Circuitos Tributários*, e são oferecidos diretamente aos usuários. Este conjunto de canais é amostrado seqüencialmente, e tais amostras são organizadas serialmente em *Time-Slots* para formar uma seqüência de *bits* com velocidade maior.

Na prática, um número qualquer de tributários pode ser multiplexado, e da mesma forma, um *Time-Slot* definido pode ter um número qualquer de *bits*. A organização em octetos é comum, devido a que os sistemas de voz PCM utilizam 8 *bits* para a quantização de uma amostra, e porque os sistemas de comunicação de dados transferem tipicamente grupos de 8 *bits*, conhecidos no ambiente dos computadores como *bytes*. Em ambos os casos, é necessário manter a integridade do octeto no destino, isto é, a seqüência de *bits* deve estar conformada pelos mesmos 8 *bits* definidos na fonte. Uma questão importante que deve observar-se é que na TDM síncrona, a fonte e o destino dos dados em cada *time-slot* são conhecidos, assim, não

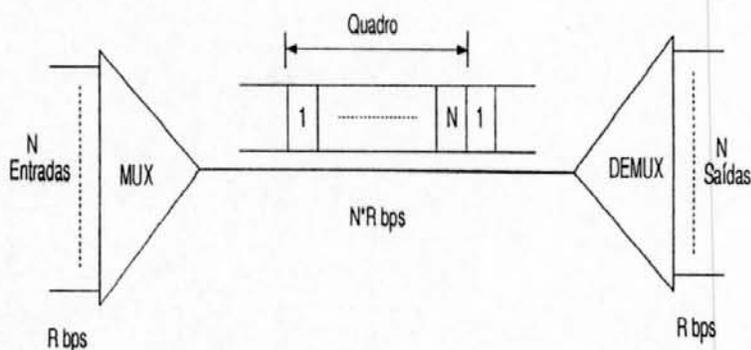


FIGURA 3.5 – TDM Síncrona.

existe motivo para colocar *bits* de endereçamento em cada *slot*. A figura 3.5 descreve a TDM síncrona.

O mecanismo da TDM síncrona é simples. Cada linha de entrada armazena seus dados em um *buffer*; o multiplexador lê estes dados seqüencialmente, toma blocos de dados de tamanho fixo de cada *buffer* e os envia através da linha. A leitura em seqüência de todos os *buffers* correspondentes a cada tributário, produz um *quadro* de dados. Para realizar a demultiplexação, os limites dos *time-slots* devem ser conhecidos. Além disso, para garantir que a correspondência entre canais tributários de entrada e saída seja mantida, a demultiplexação precisa conhecer o início de quadro. Para este propósito, o multiplexador insere *bits* adicionais dentro do quadro, denominados *Bits de Quadro*. Estes *bits* seguem um padrão determinístico que pode ser reconhecido no demultiplexador sem ser confundidos com os *bits* de informação. Quando o demultiplexador localiza este *bits*, através de um processo denominado *Recuperação de Quadro*, ele possui um ponto de referência para localizar o início de quadro, e realizar a demultiplexação, isto é, vai preenchendo os *buffers* das linhas de saída um a um.

Devido a que o Multiplexador não pode atender um número ilimitado de *bits*, é necessário garantir que a mínima taxa de *bit* do canal de saída de alta velocidade, seja maior ou pelo menos igual à soma das taxas máximas de *bits* dos canais tributários mais a taxa de *bit* requerida para os *bits* de enquadramento ou sincronização. No caso em que um dispositivo conectado a uma linha de entrada

não esteja enviando dados, o multiplexador deve enviar *slots* vazios. Portanto, a taxa de transferência de dados pode ser menor que a capacidade total do sistema.

Para o uso eficiente de uma linha digital, requer-se que a maioria de *time-slots* sejam atribuídos e utilizados. Sob estas circunstâncias é inteiramente viável que os dados de um determinado *time-slot* possam ser alocados em um *time-slot* diferente sobre a mesma ou outra linha. Este deslocamento ou intercâmbio de *time-slots* é essencial para uma Comutação Temporal eficiente, tal como é descrita para comutação e transmissão digital. Este princípio foi primeiramente reconhecido por H. Inose e J. P. Runyon em 1960 [JOE 79].

A técnica TDM síncrona permite três formas de comutação:

- *TDM Bus Switching*
- *Time Slot Interchange (TSI)*
- *Time Multiplexed Switching (TMS)*

A seguir, tais formas de comutação são descritas com detalhe. Um estudo comparativo destas técnicas pode ser encontrado em [AGU 95].

#### 3.3.1.1 *TDM Bus Switching*

Quando utiliza-se *TDM Bus Switching* ou Comutação por Barramento TDM, cada dispositivo conecta-se ao comutador através de duas linhas com *buffers*, uma correspondente à entrada e outra correspondente à saída. Estas linhas estão conectadas a um barramento digital de alta velocidade mediante chaveamento controlado. A cada linha de entrada é atribuído um *time-slot*. Durante o tempo correspondente a um *slot*, a chave da linha é habilitada, permitindo que uma porção dos dados sejam carregados no barramento. Para o mesmo *time-slot*, a chave da linha de saída também é habilitada. Assim, durante um *time-slot*, os dados são comutados desde a linha de entrada habilitada até a linha de saída habilitada.

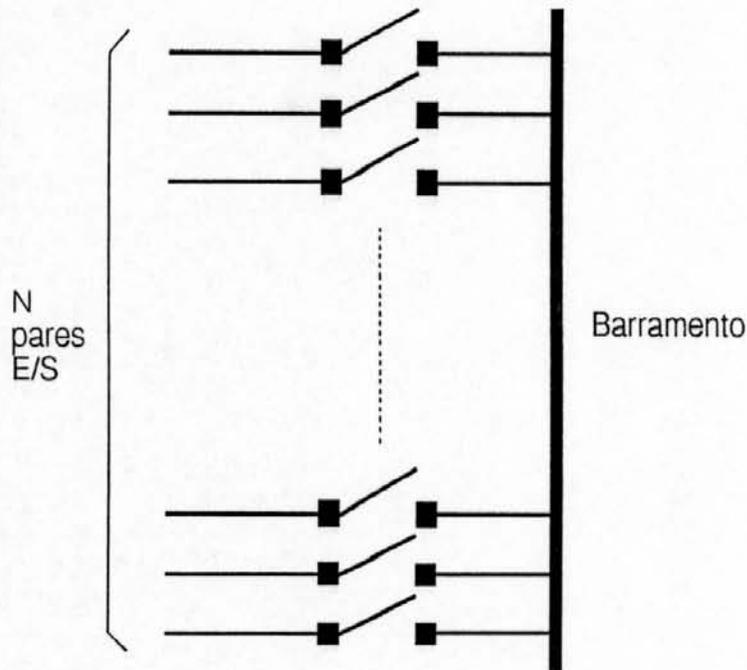


FIGURA 3.6 – Comutador por Barramento TDM.

O processo de comutação consiste em emparelhar diferentes linhas de entrada e saída, habilitando seu chaveamento em sucessivos *time-slots*. Um dispositivo conectado a um barramento TDM opera em modo *full-duplex* transmitindo durante um determinado *time-slot* e recebendo durante outro.

Um comutador que opera com esta configuração e que suporta N dispositivos, precisa de N ocorrências repetitivas de *time-slots*, cada um conectando uma linha de entrada com uma linha de saída, para formar um quadro. A figura 3.6, mostra uma implementação de um comutador por Barramento TDM.

A atribuição da entrada pode ser fixa, portanto, as atribuições nas saídas permitem diversas conexões. O tempo de duração de um *slot* deve ser igual ao tempo de transmissão da entrada mais o atraso de propagação entre a entrada e a saída no barramento. Porém, para manter a uniformidade entre todos os *time-slots*, a duração deles é definida como o tempo de transmissão mais o atraso de propagação máximo, isto é de um extremo a outro do barramento. Por questões de eficiência, o atraso de propagação deve ser muito menor que o tempo de transmissão de um *time-slot* [AGU 95].

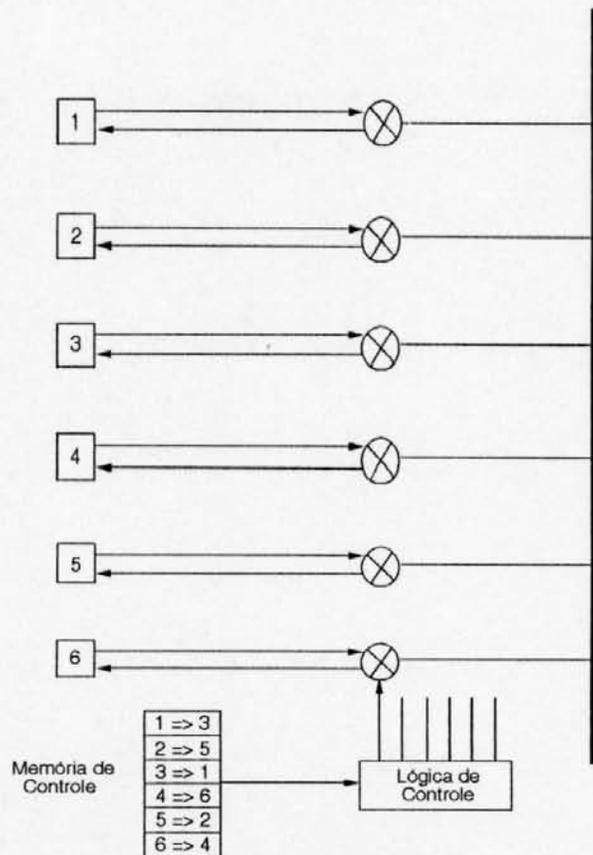


FIGURA 3.7 – Controle do Barramento TDM.

Uma implementação do controle para um comutador por Barramento TDM é mostrado na figura 3.7.

Considerando um tempo de propagação no barramento de  $0,01\mu\text{s}$  e 6 *time-slots* de  $5,01\mu\text{s}$  cada, o tempo total de ocupação do barramento será de  $30,06\mu\text{s}$ . Como a memória de controle indica quais chaves serão habilitadas durante cada *time-slot*, um ciclo do controlador deve durar  $30,06\mu\text{s}$ , tempo necessário para ler todas as posições da memória. Assim, durante o primeiro *time-slot* de cada ciclo, a chave da entrada do dispositivo 1 e a chave da saída do dispositivo 3 são habilitadas, permitindo a passagem dos dados desde o dispositivo 1 até o dispositivo 3 através do barramento. As palavras restantes da memória de controle são acessadas nos *time-slots* seguintes, e tratadas adequadamente.

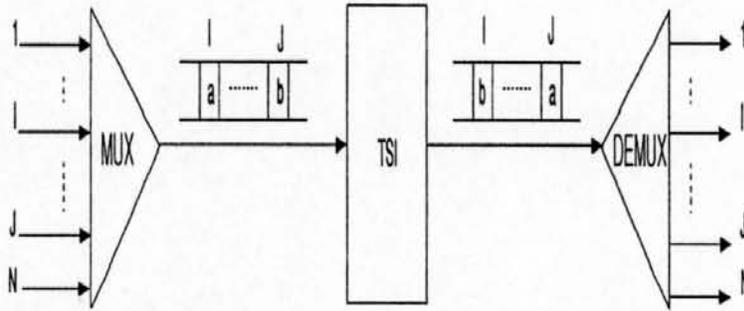


FIGURA 3.8 – Operação de uma Unidade TSI.

### 3.3.1.2 Time Slot Interchange (TSI)

Uma unidade TSI opera sobre uma seqüência TDM síncrona de *time-slots*, ou canais, trocando pares de *slots* para conseguir uma operação *full-duplex*. A figura 3.8, mostra como a linha de entrada de um dispositivo I é conectada à linha de saída do dispositivo J, e vice-versa. Observa-se também, como as linhas de entrada de N dispositivos são passadas através de um multiplexador TDM síncrono para produzir um quadro TDM com N *slots*. Para conseguir a interconexão de dois dispositivos, os *time-slots* correspondentes às duas entradas são trocados; a seqüência resultante é demultiplexada nas saídas dos N dispositivos. O resultado obtido é uma conexão *full-duplex* entre pares de linhas.

Neste ponto é oportuno mencionar que, a multiplexação geralmente não é considerada como uma função de comutação, mas quando estritamente considerada, esta inclui o acesso a memórias do tipo TSI.

Devido ao custo baixo das memórias digitais, a comutação temporal fornece funções de comutação digital de maneira mais econômica que as implementações do tipo espacial. Quando sinais digitais podem ser multiplexados em um formato TDM simples, é possível então implementar comutadores muito econômicos que realizam apenas comutação temporal. Porém, as limitações práticas de velocidade das memórias determinam o tamanho e capacidade máximos de um comutador temporal, desde que sempre é necessária alguma quantidade de comutação espacial nos comutadores de grande porte.

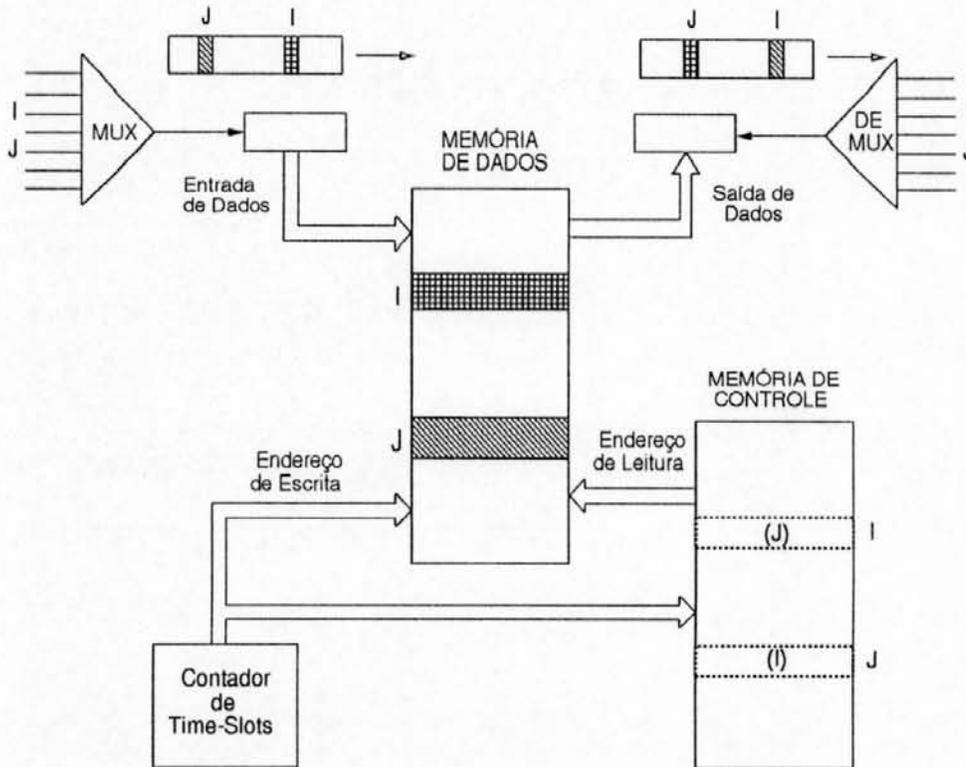


FIGURA 3.9 - *Time Slot Interchange*.

A figura 3.9 descreve a operação funcional básica de um TSI. Individualmente, os circuitos contendo os dados digitais são multiplexados e demultiplexados de uma maneira fixa para estabelecer um quadro TDM em duas direções (*full-duplex*). Como mencionado anteriormente, as funções de multiplexação e demultiplexação podem ser realizadas em conjunto com as de comutação, ou podem ser implementadas em terminais de transmissão remotos.

O intercâmbio de informação entre dois *time-slots* diferentes é realizado por uma unidade TSI. No TSI da figura anterior as palavras de dados que chegam nos *time-slots* de entrada são escritas seqüencialmente na memória de armazenamento de dados. Por outro lado, as palavras de dados para os *time-slots* de saída são lidas segundo os endereços fornecidos pela memória de controle. A conexão *full-duplex* entre o canal I e o canal J implica que o endereço I da memória de dados é lido durante o *time-slot* de saída J e vice-versa, tal como indica a memória de controle associada. A memória de dados é acessada duas vezes durante cada quadro de *time-slots*. A primeira vez, quando o circuito de controle (não mostrado na figura) seleciona o número de *time-slot* como endereço de escrita. A segunda vez, quando o

conteúdo da memória de controle para um *time-slot* particular é selecionado como endereço de leitura.

Visto que, por cada canal que entra e que sai, no TSI, são requeridas uma operação de leitura e uma operação de escrita, o número máximo de canais  $NC$  que pode ser suportado por um comutador de memória simples, calcula-se com a equação seguinte:

$$NC = \frac{125}{2 \cdot T_c} \quad (3.1)$$

$NC$  é o número de canais, o fator **125** é o tempo de duração de um quadro em microsegundos para canais de voz ou dados amostrados a 8KHz, e finalmente,  $T_c$  é o tempo de resposta da memória em microsegundos.

Nota-se que, cada memória TSI deve ter no mínimo uma palavra para cada *time-slot* por quadro. A memória de controle precisa também do mesmo número de palavras, mas cada palavra possui um número de *bits* ( $NB$ ) que depende da equação seguinte:

$$NB = \log_2 NC \quad (3.2)$$

Tanto o contador de *time-slots* como a lógica de controle para selecionar endereços e habilitar a nova informação a ser armazenada na memória de controle, podem ser implementados com circuitos integrados convencionais ou por circuitos integrados de aplicação específica (ASICs).

Nota-se que, em uma unidade TSI, para permitir o intercâmbio de dois *time-slots*, os dados de entrada de um canal devem ser armazenados até que eles possam ser enviados no canal correto no próximo ciclo de quadro. Portanto, um comutador que usa TSI introduz um atraso para poder realizar a comutação desejada. Tal atraso, denominado *tempo de latência* do equipamento, é introduzido quando as memórias são escritas, conforme os dados vão chegando, e lidas quando os dados

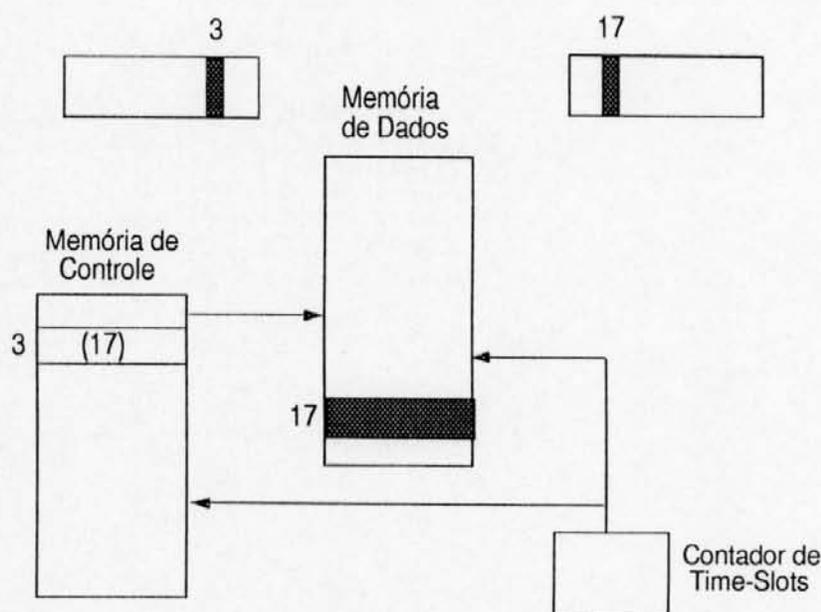


FIGURA 3.10 – Escrita Aleatória e Leitura Seqüencial.

são transferidos. Se uma posição de memória é ocupada por cada *time-slot* de um quadro TDM, a informação de cada canal TDM pode ser armazenada durante o tempo de duração de um quadro completo, sem existir a possibilidade de sobrescrita dos dados na memória. No máximo, a memória que implementa a etapa temporal, introduz um atraso de um quadro aos dados que passam através dela, devido à necessidade de armazenar os *time-slots* de um quadro, enquanto os *time-slots* do quadro precedente estão sendo lidos em outra parte da memória.

Existem duas formas básicas de gerenciamento das memórias: escrevendo seqüencialmente na memória de dados e lendo aleatoriamente sob o comando da memória de controle ou, escrevendo aleatoriamente na memória de dados e lendo-a seqüencialmente. As figuras 3.10 e 3.11, descrevem as duas formas de operação e indicam como as memórias são acessadas para transferir a informação desde o *time-slot* 3 até o *time-slot* 17. Observa-se que, as duas formas de operação usam uma memória de controle cíclica que é acessada de forma sincronizada com o contador de *time-slots*.

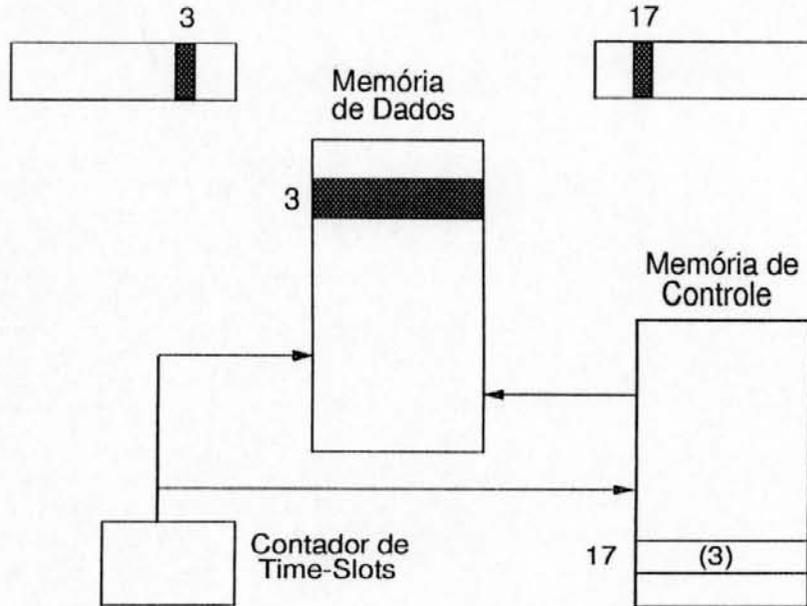


FIGURA 3.11 – Escrita Seqüencial e Leitura Aleatória.

### 3.3.1.3 *Time Multiplexed Switching* (TMS)

O TSI é uma forma simples e efetiva de comutar dados em formato TDM, porém o tamanho do comutador, em termos de número de conexões, está limitado pela velocidade de acesso das memórias.

Nota-se também que o TSI pode suportar um número limitado de conexões, pois a medida que o tamanho da unidade aumenta, mantendo fixa a velocidade de acesso, o atraso dentro do TSI também cresce. Uma alternativa para superar tais problemas é o uso de múltiplas unidades TSI [SKA 79]. Deste modo, para conectar dois canais ligados a um mesmo TSI, os seus *time-slots* podem ser trocados de maneira convencional, tal como explicado no item anterior. Porém, para conectar um canal de um quadro TDM que está ligado a um determinado TSI, com outro canal que pertence a outro quadro TDM e que está ligado a outro TSI, é necessário contar com algum tipo de interconexão especial entre unidades TSI. Esta técnica denomina-se *Time Multiplexed Switching* (TMS).

Redes de múltiplas etapas podem ser construídas concatenando etapas TMS e etapas TSI. As etapas TMS são denominadas como S (*Space*), e as etapas TSI são denominadas como T (*Time*). Os sistemas de comutação são descritos pela

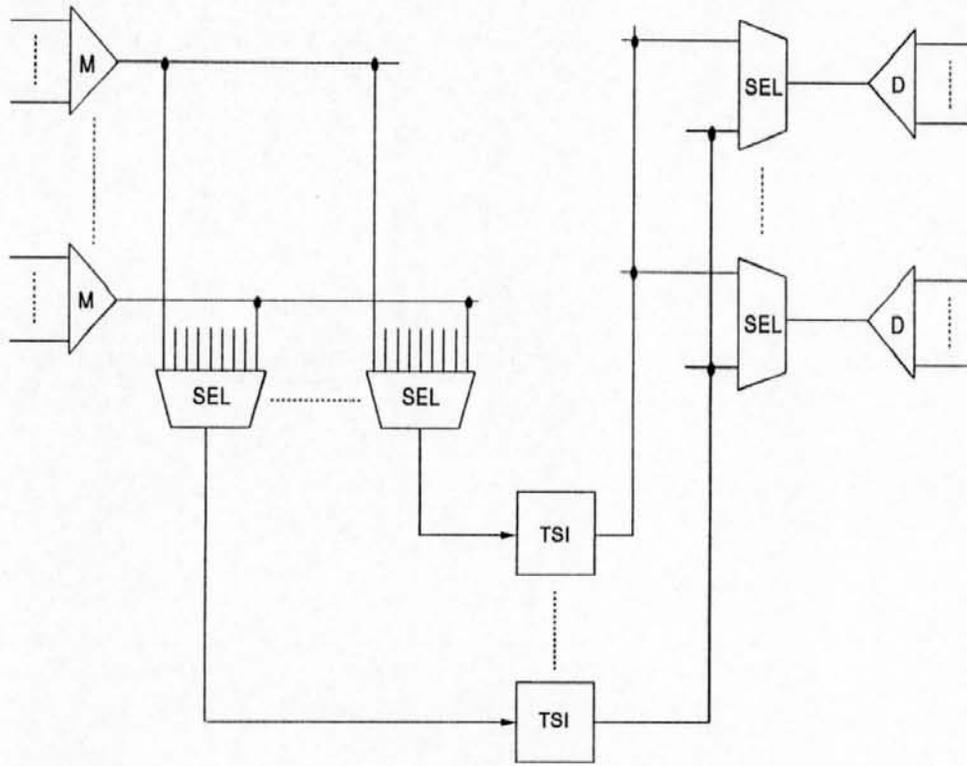


FIGURA 3.12 – Rede STS.

enumeração de suas etapas desde a entrada até a saída, usando os símbolos T e S. Por exemplo, um sistema TST é um sistema de 3 etapas que realiza sucessivamente comutações temporal, espacial e temporal desde sua entrada até sua saída. As figuras 3.12 e 3.13, mostram as arquiteturas de dois sistemas: STS e TST, respectivamente. Nos dois casos, as etapas TMS são implementadas mediante o uso de seletores digitais (SEL), os quais selecionam uma entrada por vez durante o tempo de um *time-slot*. Os seletores SEL realizam a comutação de quadros completos TDM.

Em uma arquitetura STS, o caminho entre o canal de entrada e o canal de saída possui várias rotas físicas possíveis, igual ao número de unidades TSI do sistema, mas uma rota apenas pode ser escolhida cada vez. Por outro lado, em uma arquitetura TST, as diferentes rotas entre dois canais estão todas no domínio do tempo, pois existe só um caminho físico possível. As duas arquiteturas apresentam probabilidade de bloqueio, que pode ser reduzida incrementando o número de unidades TSI adequadamente [AGU 95]. Todos os comutadores multi-etapa precisam de um algoritmo de roteamento, para determinar a rota mais adequada desde a entrada até a saída.

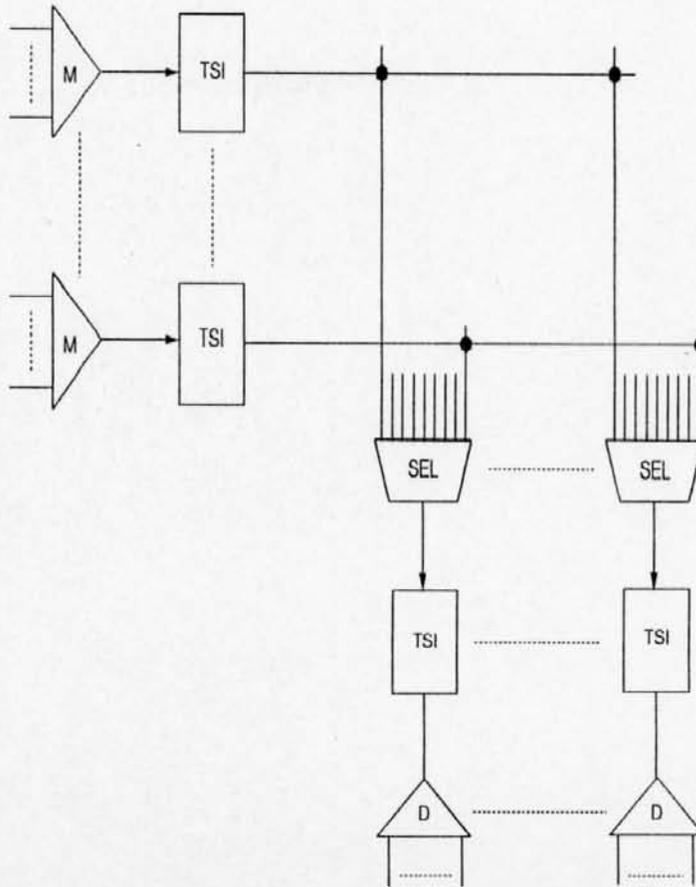


FIGURA 3.13 – Rede TST.

### 3.3.2 TDM Assíncrona (ATDM)

Nesta técnica, o tempo de utilização do meio físico de transmissão é alocado dinamicamente entre os diversos canais de comunicação sob demanda, isto é, segundo o tráfego de dados gerados entre eles. Esta estratégia permite um compartilhamento mais eficiente da capacidade de transmissão da sub-rede de comunicação, no caso de tráfego tipicamente irregular. Em razão disso, a técnica *Asynchronous Time Division Multiplexing* (ATDM) é adotada como base dos principais mecanismos de controle de acesso. Visto que, na técnica ATDM não existe alocação fixa, a priori, de intervalos de transmissão, é necessário que os dados carreguem a identificação de sua origem.

Enquanto a técnica de multiplexação STDM possui uma estratégia de controle de utilização do meio que consiste simplesmente no estabelecimento prévio de intervalos de transmissão associados a cada canal, a técnica ATDM caracteriza-

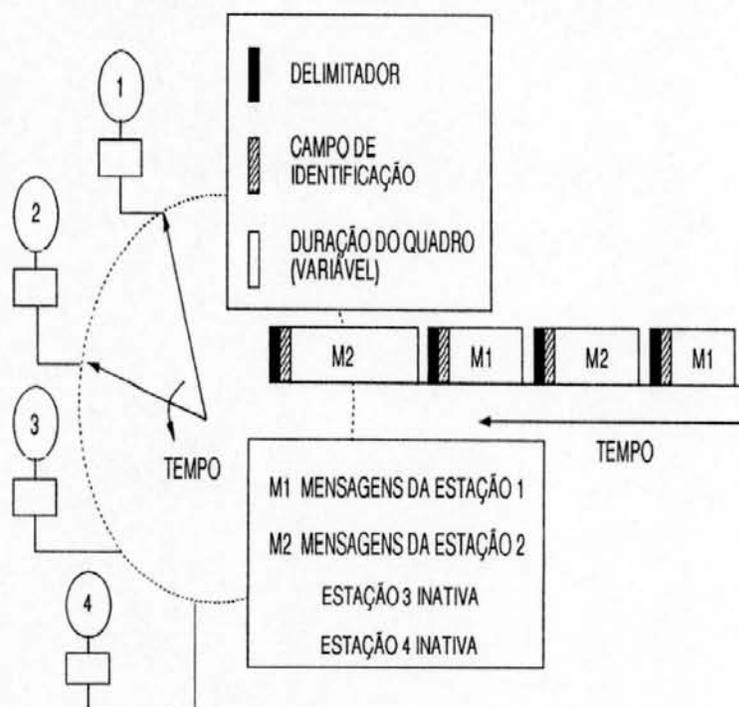


FIGURA 3.14 – TDM Assíncrona.

se pela alocação estatística dos intervalos de transmissão, necessitando, portanto, de uma estratégia de gerência e compartilhamento do meio de transmissão mais elaborada [GIO 86]. Porém, o ganho médio na eficiência da transmissão é de aproximadamente 4 a 1 [SPO 93], devido a que ATDM utiliza os *time-slots* dos usuários que não estão transmitindo dados. A figura 3.14 descreve a técnica ATDM.

Os tipos de estratégia de controle de acesso associados à técnica ATDM, deram origem aos diversos mecanismos de controle de acesso ao meio utilizados nas redes locais de computadores.

### 3.3.2.1 *Time Division Multiple Access* (TDMA)

A Comutação de Circuitos aplica-se também a topologias de rede de múltiplo acesso, tais como Barramento ou Anel. A técnica apropriada é altamente dependente da topologia. Por exemplo, na topologia em anel, pode-se definir diretamente um esquema para TDM: *Slotted Ring*. Esta técnica descreve um método para formar uma estrutura de quadro e *time-slot* fixos para uma topologia em anel, assim como a maneira na qual um circuito pode ser formado.

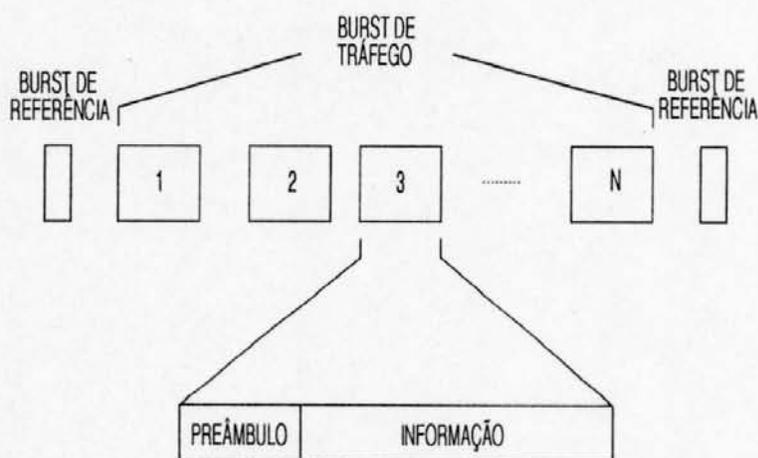


FIGURA 3.15 – Estrutura de Quadro de um sistema TDMA.

Por outro lado, TDM é difícil de aplicar em uma topologia tipo barramento. A razão é simplesmente que o TDM requer um quadro fixo conhecido por todos os nós da rede, mas em um barramento, particularmente em uma rede *broadcast* geograficamente grande (rede de satélites), os atrasos de propagação no meio de transmissão serão tipicamente grandes para o tempo de um *bit*.

Contudo, é comum ainda aplicar técnicas TDM nesta situação, mas devem ser modificadas considerando os atrasos de propagação significativos entre usuários. Esta modificação origina uma outra técnica conhecida como *Time Division Multiple Access* (TDMA) [LEE 90]. Esta técnica de multiplexação é derivada da combinação dos princípios das técnicas STDM e ATDM.

Esta técnica é aplicável a qualquer topologia tipo barramento ou *broadcast*, onde existe um conjunto de transmissores e receptores, que "escutam" simultaneamente cada transmissão. TDMA tem sido aplicada extensivamente nas redes de satélites, ver figura 3.15.

TDMA requer um nó de controle centralizado, uma característica que pode ser evitada usando técnicas de acesso aleatório. A função primária deste nó de controle é transmitir um *burst* de referência periódico, semelhante aos *bits* de quadro adicionados em TDM, que define um quadro e fornece uma forma de sincronização para todos os outros nós.

O quadro assim definido, é dividido em *time-slots*, como em TDM, e a cada nó é atribuído um único *time-slot*, pelo qual transmite sua informação. Cada nó, de um grupo de  $N$ , transmite um *burst* de tráfego dentro do seu *time-slot* atribuído. Portanto, a técnica é similar a TDM, porém possui diferenças substanciais:

1. Precisa de *tempos de guarda*, que considerem a diferença de fase com que cada nó recebe o *burst* de referência, e que evite o *overlap* entre o tráfego de cada nó.
2. É necessário um *preâmbulo* no início de cada *burst* de tráfego, que permita ao receptor obter a fase de temporização e a fase do sinal portador.
3. Deve existir um mecanismo de controle centralizado para alocar *time-slots* e comunicar a informação de tais alocações aos nós. Os *time-slots* podem ser pré-alocados, considerando que os câmbios são pouco frequentes, e ocorrem apenas como resultado da demanda de tráfego.

## 4 REDES CORPORATIVAS E1

### 4.1 Visão Geral

Com a abertura do mercado de informática, que possibilitou às empresas sediadas no Brasil, o acesso às tecnologias existentes nos países mais adiantados, e a quebra do monopólio estatal no setor das comunicações, no ano 1990, a estruturação de redes particulares para comunicação de dados, tornou-se viável, utilizando-se em alguns casos nós de rede E1 [COR 93]. Estas redes particulares foram denominadas *Redes Corporativas*.

A principal finalidade destas redes é atender às necessidades de intercâmbio de informação entre os diferentes setores de uma corporação. Entenda-se por corporação, um local industrial, uma universidade, um banco. A corporação estabelece sua própria infra-estrutura de comunicações, adequando esta a suas necessidades particulares. Isto origina que, no início, sejam feitos investimentos em equipamentos e instalações, por parte da corporação. Entretanto, o baixo custo operacional das redes corporativas comparado aos serviços públicos, proporciona uma economia que compensa grandemente o investimento inicial.

Embora os sistemas atuais de informação utilizados pelas corporações, estejam constituídos basicamente por Redes Locais de Computadores, com frequência é necessário interconectar tais redes, de forma que possam compor um sistema de informação global da corporação [KAI 95a]. Esta rede deve ser capaz de suportar, além de tráfego de dados, canais de voz e eventualmente, vídeo digital, caracterizando assim uma *Rede Corporativa Integrada*. Em redes deste tipo, o multiplexador E1, que abordaremos posteriormente, pode ser utilizado como elemento básico de implementação, em diferentes topologias de rede, tais como:

1. Rede E1 como *backbone* para interligação de centros de tráfego. Neste tipo de rede os nós E1 devem ter, além da multiplexação, capacidade de roteamento.
2. Rede E1 com controle fixo e centralizado. São aplicáveis a estruturas mais modestas, com no máximo três nós e roteamento fixo, controlado centralmente. Os multiplexadores E1 constituem os nós da rede, com capacidade de comutação por comando de operador.
3. Rede E1 com gerenciamento e controle distribuídos e auto-configurável. Todos os nós deste tipo de rede possuem capacidade de processamento, podendo assim executar tarefas de operação e controle localmente. Cada nó recebe recursos de *software* e *hardware* que o habilitam para a execução de funções de controle e de gerenciamento da rede. A grande vantagem desta topologia é sua maior autonomia e robustez, visto que não existe um nó essencial para o funcionamento do sistema. Contudo, a complexidade e os custos de implementação mais elevados, fazem com que seja mais indicada para grandes redes, ou redes nas quais seja preciso uma alta confiabilidade.

Geralmente, as Redes Corporativas contam com algum tipo de acesso à Rede Pública de Comunicações, de modo a ser possível o estabelecimento de enlaces de comunicação sobre distâncias muito grandes, não cobertas pela estrutura própria da rede. Esta característica é interessante para pequenas e médias corporações, para as quais não é viável manter o aluguel de canais privativos. Da mesma forma, permite a interconexão eventual entre Redes Corporativas distintas, onde a Rede Pública serve como elemento de interligação entre elas. A figura 4.1 mostra a configuração típica de uma Rede Corporativa E1.

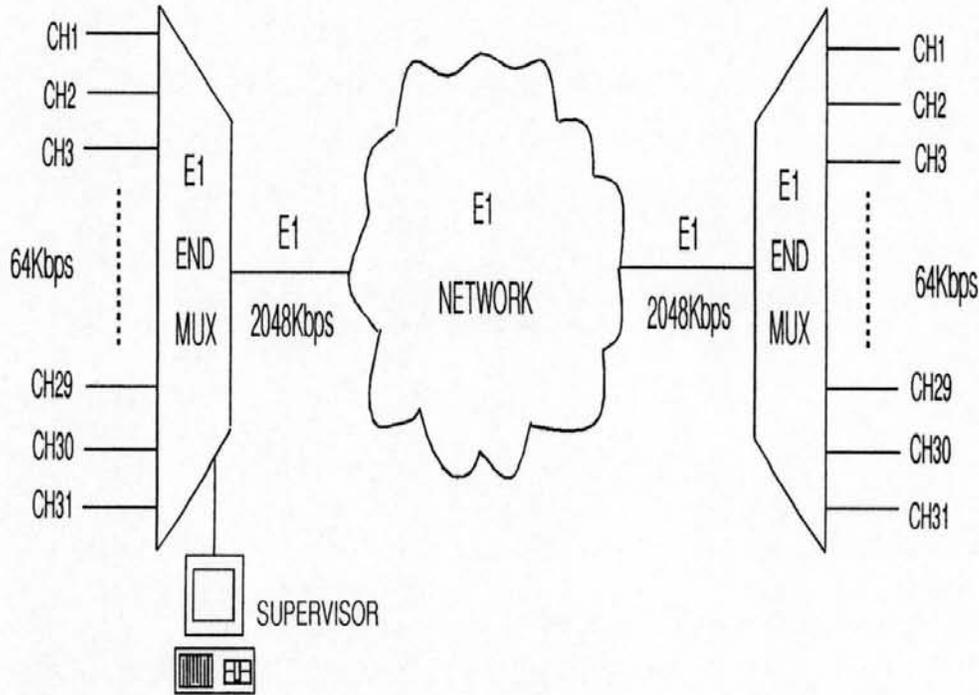


FIGURA 4.1 – Rede Corporativa E1 ponto-a-ponto.

## 4.2 A Hierarquia Digital E1

O ITU-T (*International Telecommunications Union*) definiu a hierarquia digital E1, na sua *Recomendação G.702: Hierarquia Digital por Velocidade*. O termo E1 refere-se a 32 canais de 64 Kbit/s para transmissão de voz ou dados digitais, formando um sinal de banda básica com agregado de 2.048 Mbit/s. Este padrão hierárquico originou-se na Europa, daí sua denominação de padrão europeu, e posteriormente foi adotado por outros países, entre eles o Brasil. Nos Estados Unidos o padrão utilizado é denominado T1, o qual foi originariamente definido pela AT&T, e compreende 24 canais de 64 Kbit/s multiplexados em um sinal agregado de 1.544 Mbit/s.

As principais características da hierarquia E1 são:

1. Possui circuitos a 4-fios: Devido a que esta hierarquia evoluiu a partir do tradicional par trançado. Dois fios são para transmissão, e dois para recepção. Ambos os pares são balanceados.

2. Permite enlaces *full-duplex*: A transmissão e recepção de dados pode realizar-se simultaneamente. Alguns usuários derivam outros usos tais como um caminho unicamente para transferência de arquivos ou para impressão remota. Também um serviço alternado a 2-fios, tal como as comunicações de voz.
3. É digital: Este é um serviço completamente digital. Dados analógicos, fax analógico e voz analógica, são convertidos em pulsos digitais para transmissão na linha.
4. Usa TDM: 32 canais tipo B (64 Kbit/s) são multiplexados juntos para formar um agregado de 2.048 Mbit/s. TDM permite que cada canal utilize um intervalo de tempo igual a 1/32 do tempo total do agregado. Estes intervalos, denominados *time-slots*, podem ser fixos para cada canal. O *time-slot* de cada canal está sempre presente no agregado, mesmo quando não existe informação a ser transmitida.
5. Usa PCM: O *Pulse Code Modulation* (PCM) ou Modulação por Código de Pulso, permite amostrar um sinal de voz analógico ou qualquer outro serviço 8000 vezes por segundo e atribuir uma palavra de 8 *bits* para cada amostra, produzindo assim uma capacidade de canal de 64 Kbit/s.
6. Possui um formato de quadro: Como é usado TDM, os 32 canais são multiplexados em um quadro que será transportado através da linha. Cada quadro resulta em uma seqüência de 256 *bits*, correspondendo 8 *bits* por cada canal. Visto que são gerados 8000 quadros por segundo, um quadro têm 125 $\mu$ s de duração. Ver figura 4.2.
7. Usa codificação bipolar: A hierarquia E1 utiliza código HDB3 para transportar os dados através da linha. Este código bipolar possui duas grandes vantagens:
  - Apresenta um espectro de energia concentrado em uma faixa de freqüência menor, em torno de uma freqüência igual à metade da

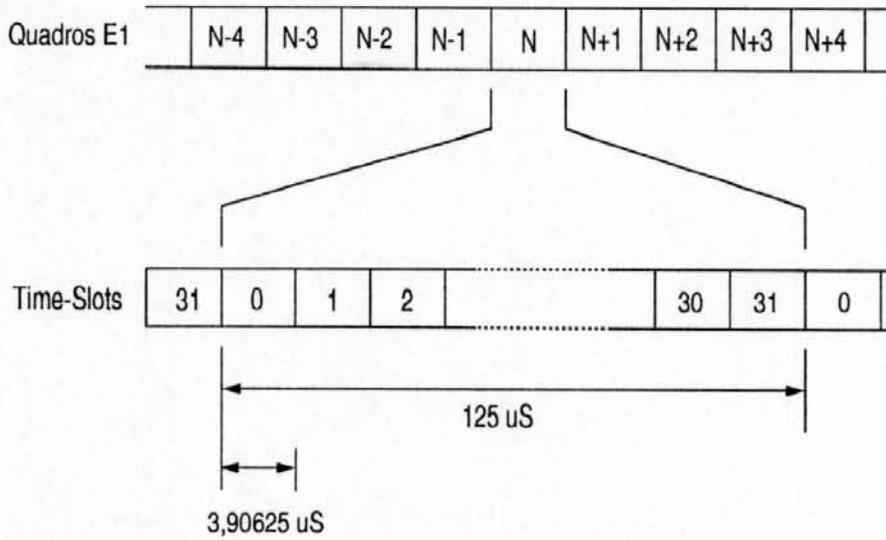


FIGURA 4.2 – Formato de Quadro E1.

taxa de sinalização dos bits de dados. Isto permite incrementar o espaçamento entre repetidores, e possibilita uma filtragem linear da informação de relógio [GIO 86].

- Permite o uso de transformadores de isolamento para conexões mais econômicas. Isto é possível porque o código HDB3 não possui nível DC nem componentes nas frequências baixas [STA 90].
8. Oferece diversas configurações: A hierarquia E1 possui 31 canais de 64 Kbit/s cada, mais um canal adicional de 64 Kbit/s. Este é considerado Canal de Serviço. Contudo, o equipamento de multiplexação pode ser configurado em várias formas [BAT 92]:
- Como um único canal de 2.048 Mbit/s para vídeo-conferência ponto-a-ponto.
  - Como dois canais de dados de alta velocidade (512 Kbit/s) mais 16 canais de baixa velocidade para voz e dados a 64 Kbit/s.
  - Qualquer outra combinação de serviços.

## 4.3 O Multiplexador E1

Os multiplexadores E1 são equipamentos de multiplexação que utilizam facilidades de transmissão digitais, podendo integrar redes *backbone* que manipulam sinais de voz, dados e imagens, atuando apenas no nível físico do modelo de referência OSI. Portanto, são estruturas transparentes para os terminais usuários nos níveis lógicos tais como protocolos de enlace, que evoluíram diretamente dos sistemas TDM/PCM públicos. As características mais comuns destes dispositivos são:

- Multiplexação TDM de sinais digitais.
- Compressão de voz analógica
- Comutação através de comandos de operador.
- Gerenciamento Centralizado.
- Compatibilidade com padrões TDM/PCM públicos e sua hierarquia.
- Facilidades de interconexão com outras redes: X.25, TCP/IP, *Frame Relay*, FDDI e ATM.

### 4.3.1 Descrição Funcional

O multiplexador pode ser imaginado como uma chave rotativa, que dá uma volta completa a cada  $T$  segundos, e que a largura de contato corresponde a  $t$  [KAI 94]. Este dispositivo amostra seqüencialmente os dados provenientes de cada canal tributário de 64 Kbit/s durante intervalos de tempo  $t$ , a cada período  $T$ , exigindo um sincronismo de amostragem entre o circuito multiplexador e o demultiplexador.

Os dados provenientes dos canais tributários são interfaceados por módulos de recepção que sincronizam (mediante técnicas de armazenamento elástico) e

regeneram os sinais provenientes das linhas de 64 Kbit/s, e fornecem em sua saída um sinal binário NRZ (*No Return to Zero*). À medida que os canais tributários vão sendo amostrados, o agregado de 2048 Kbit/s é transmitido em direção ao equipamento remoto (demultiplexador), através de um módulo transmissor que converte os sinais NRZ para HDB3.

No equipamento remoto, uma interface de recepção HDB3, regenera o sinal de 2048 Kbit/s, convertendo-o em sinais binários NRZ, que são armazenados em um *buffer* elástico de recepção. A partir deste *buffer*, no instante correspondente a cada canal tributário, os sinais de cada um deles são direcionados ao seu respectivo módulo de transmissão de 64 Kbit/s, através dos quais, os sinais de cada tributário serão transmitidos aos seus destinos via linha.

No sistema E1 são transmitidos 32 canais, sendo que destes, dois canais são reservados à transmissão de sinalização de serviço e sincronismo do sistema. O *canal zero* transporta informação de sincronismo, enquanto que o *canal 16* transporta sinalização de serviços de voz. No caso do multiplexador estar operando somente com dados digitais, o *canal 16* funciona como um canal de usuário transportando dados. A estrutura de quadro básica para o padrão E1 estabelece um comprimento de quadro igual a 256 bits, 8 bits por cada canal de usuário, com uma taxa de amostragem de 8KHz. O *canal zero* transporta a informação de enquadramento através de *palavras de alinhamento de quadro* que sincronizam o emissor e o receptor do sistema E1. A palavra de alinhamento de quadro é transmitida alternadamente com a *palavra de serviço* no *canal zero*. O primeiro bit do *canal zero* é reservado para uso internacional. Os bits 2 a 8 da palavra de alinhamento têm sempre o mesmo formato *0011011*. O receptor determina a posição do quadro de pulsos baseado nas palavras de alinhamento dos quadros entrantes. Considera-se a condição de *perda de alinhamento*, quando 3 palavras de alinhamento de quadro consecutivas forem recebidas com erro. Uma análise aprofundada do multiplexador E1 encontra-se em [KAI 95a].

## 4.4 Comutação na Rede Corporativa E1

A função de comutação em uma rede de comunicações refere-se à alocação dos diferentes recursos da rede: meios de transmissão, repetidores, sistemas intermediários, etc., para a transmissão pelos diversos dispositivos conectados [SOA 95]. A alocação de tais recursos está intimamente ligada à forma de multiplexação dos meios de transmissão.

Dentro de uma Rede Corporativa E1 ponto-a-ponto com controle centralizado, a comutação consiste basicamente no estabelecimento de um caminho de comunicação dedicado entre dois usuários conectados à rede, que permita o intercâmbio de informação entre eles. As funções de comutação são fornecidas por um Sistema de Comutação integrado no multiplexador que opera como mestre. O coração do Sistema de Comutação é um Comutador Digital que apesar de ser um elemento não visível pelo usuário, possui grande importância em termos de oferecimento dos serviços disponíveis da rede [BEJ 91]. A adequada programação do comutador permite trocar o fluxo de dados de um canal para outro dentro da rede de forma transparente para os usuários.

O Sistema de Comutação manipula os 31 canais de usuário tipo B (64 Kbit/s) conectados a cada nó da rede, e realiza suas funções de comutação tanto no multiplexador local como no multiplexador remoto. Isto significa que o sistema pode estabelecer conexões entre dois dispositivos conectados no mesmo multiplexador, seja este local ou remoto, e entre um dispositivo conectado no multiplexador local e outro dispositivo conectado no multiplexador remoto.

A configuração do Comutador Digital é realizada por meio de um microcomputador (*PC*) conectado ao Multiplexador Mestre através de um canal serial síncrono, e que trabalha emulando um terminal que roda um *software* de usuário. O processo de configuração começa quando o operador da rede acessa o terminal e solicita um conjunto de conexões entre os usuários. Logo após a verificação da coerência das conexões, o *software* de usuário converte este conjunto de conexões em

comandos de programação e os envia até o Sistema de Comutação através do canal serial síncrono. Este canal deve ser governado por um protocolo de comunicação a nível de enlace do modelo OSI. No outro extremo, o Sistema de Comutação recebe a informação e estabelece as conexões solicitadas através de seu módulo de controle, o qual habilita o comutador segundo a programação recebida. Depois, o Sistema de Comutação fica à espera de uma nova seqüência de comandos de programação.

No capítulo seguinte serão descritos o funcionamento e as principais características do Comutador Digital, bem como do Sistema de Comutação, para trabalhar no ambiente de uma Rede Corporativa El ponto-a-ponto.

## 5 COMUTADOR DIGITAL TDM E1

Até este ponto foram tratados aspectos concernentes aos Sistemas de Comutação, a Comutação de Circuitos e as Redes Corporativas E1. O primeiro tópico pretende dar uma introdução não apenas ao conceito de Comutação, mas também descrever suas principais modalidades e especialmente destacar que os elementos de um Sistema de Comutação são constantes, independentemente da modalidade que é utilizada. Isto é, em um Sistema de Comutação sempre vamos encontrar módulos com as funções de Sinalização, Controle e Comutação.

Já o tópico dedicado à Comutação de Circuitos, oferece uma abordagem detalhada desta modalidade e faz um estudo comparativo das principais técnicas utilizadas para a implementação de Comutadores de Circuitos, ressaltando sua relação intrínseca com as técnicas TDM e sua característica de fornecer transporte de dados independente de protocolos. Desta maneira, ao apresentar o tópico relacionado com as Redes Corporativas E1 é possível intuir que ditas redes conformam um área de aplicação de tais comutadores, e que a escolha da arquitetura de um comutador para sistemas da hierarquia E1 se enquadra dentro da Comutação de Circuitos.

Neste capítulo será descrita a arquitetura de um Comutador Digital TDM E1, capaz de proporcionar funções de comutação entre os usuários conectados aos nós de uma Rede Corporativa E1 ponto-a-ponto. Os aspectos relacionados com a sincronização e programação do comutador, bem como as normas internacionais consideradas também serão descritas. Finalmente, servindo como referência, menciona-se outros trabalhos sobre o mesmo tema pesquisados na literatura técnica e industrial.

### 5.1 Arquitetura do Comutador

A figura 5.1 mostra uma Rede Corporativa ponto-a-ponto, cujos nós estão formados por Multiplexadores E1. O nó local, considerado Mestre, é formado por um multiplexador que terá embutido o comutador, e portanto, vai possuir funções

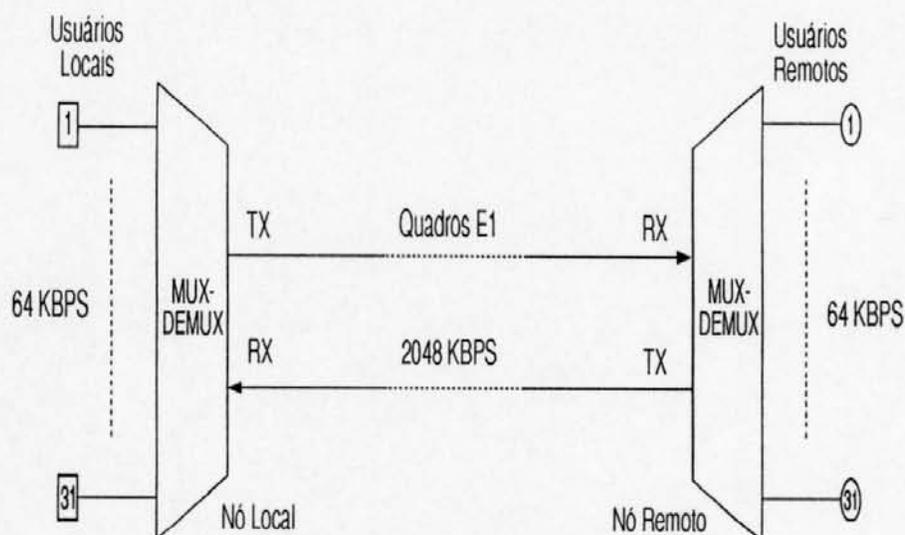


FIGURA 5.1 – Rede E1.

de multiplexação e funções de comutação. Por outro lado, o nó remoto vai ter apenas funções de multiplexação e é considerado, dentro do ambiente da rede, como Escravo. Ambos os nós atendem 31 canais de usuários com taxa de 64 Kbit/s.

### 5.1.1 Requerimentos

Certos requerimentos básicos tiveram de ser atendidos antes da escolha de uma arquitetura para o comutador do sistema. As seguintes características para o comutador foram fundamentais:

- Deve ter capacidade de estabelecer conexões entre todos os usuários conectados à rede. Isto significa que um usuário qualquer deve poder trocar informações com qualquer outro usuário, independentemente do multiplexador no qual esteja conectado. Tendo em vista o que foi explicado anteriormente, existem dois tipos ou possibilidades de comutação: Local e Remota.

*Comutação Local* significa estabelecer um enlace entre dois usuários conectados ao mesmo multiplexador. Assim, considerando o estabelecimento de um enlace entre um par de usuários, ambos pertencentes ao multiplexador local, esta é uma operação de *Comutação Local-Local*. Da mesma forma, um enlace entre um par de usuários ambos conectados

ao multiplexador remoto, implicará uma *Comutação Local-Remota*. Por último, quando um enlace é estabelecido entre um par de usuários pertencentes a diferentes multiplexadores, esta operação é denominada *Comutação Remota*.

- Deve ser configurável. Isto é, os enlaces ou conexões a serem estabelecidos para um determinado período de transmissão devem ser programados previamente através de comandos de operador. Em outras palavras, o operador da rede deve ser capaz de configurar os enlaces solicitados pelos usuários mediante a programação adequada do comutador.
- Visto que, o comutador vai formar parte de um multiplexador E1, este deve atender às recomendações nacionais e internacionais relacionadas com esses dispositivos.

### 5.1.2 Alternativas

Devido à natureza síncrona dos multiplexadores E1, e o fato deles ter evoluído diretamente dos sistemas TDM e PCM públicos, a escolha de uma arquitetura para o comutador do sistema ficou enquadrada dentro da modalidade de Comutação de Circuitos.

Segundo visto na seção 3.3.1, sobre as técnicas de acesso a enlaces ponto-a-ponto, quando sinais digitais podem ser multiplexados em um formato TDM simples, é possível então implementar comutadores muito econômicos baseados apenas em Comutação Temporal, sendo a unidade operacional básica destes comutadores o *Time Slot Interchange* (TSI).

Até aqui, torna-se evidente que a Comutação Remota pode ser facilmente implementada usando unidades TSI, uma por cada sinal de agregado de 2048 Kbit/s. Porém, a forma de implementar a Comutação Local ainda não foi discutida.

Uma alternativa bastante simples para obter Comutação Local em ambos os nós da rede é mediante o uso de comutadores espaciais do tipo Matrizes *Cross-point*. Os canais tributários de 64 Kbit/s correspondentes aos usuários da rede podem ser conectados diretamente às linhas de entrada e às linhas de saída de uma matriz de chaves semi-condutoras, uma matriz para cada nó E1. Os enlaces seriam estabelecidos pela simples habilitação de uma chave. Contudo, o elevado número de chaves requerido por cada matriz ( $31 \times 31 = 961$  chaves), bem como o grande número de linhas de controle necessárias (uma por cada chave), fazem com que esta proposta não seja viável, visto que se pretende alcançar uma solução integrada e simplificar o máximo possível o controle do sistema. Sendo assim, optou-se por procurar uma arquitetura que ofereceria uma solução conjunta ao sistema baseada em técnicas TSI.

A partir de uma nova análise do problema, observou-se que a única forma que uma unidade TSI podia fornecer Comutação Local e Remota era tendo acesso simultâneo a ambos os quadros E1 que são produzidos pelos multiplexadores da rede e que estão disponíveis nos sinais de agregado de 2048 Kbit/s que formam o enlace *full-duplex* da rede. Acessando estes quadros, tem-se a disposição os *time-slots* correspondentes ao total de usuários da rede. Contudo, acessar as linhas multiplexadas traz consigo mais uma complicação: os sinais transmitidos e recebidos são de natureza plesiócrona se comparados mutuamente, portanto, antes de realizar qualquer operação de comutação com os quadros é necessário realizar um alinhamento temporal entre estes e sincronizá-los com o relógio do multiplexador usado como referência, neste caso o relógio do multiplexador local. Para isto, é necessário o uso de técnicas de *Armazenamento Elástico* tal e como são descritas em [KAI 95].

Considerando que, o uso de armazenamento elástico e unidades TSI, simplificavam bastante a arquitetura do comutador permitindo sua integração em apenas uma pastilha de silício, optou-se por desenvolver esta solução. Na figura 5.2, observa-se como o comutador vai conectado ao multiplexador local ou mestre. Q1 e Q2 são os quadros E1 de entrada ao comutador e, Q3 e Q4 são os quadros E1 de saída com os dados comutados. Q3 é o quadro comutado que é enviado ao multiplexador remoto, e Q4 é o quadro comutado que é enviado ao multiplexador local.

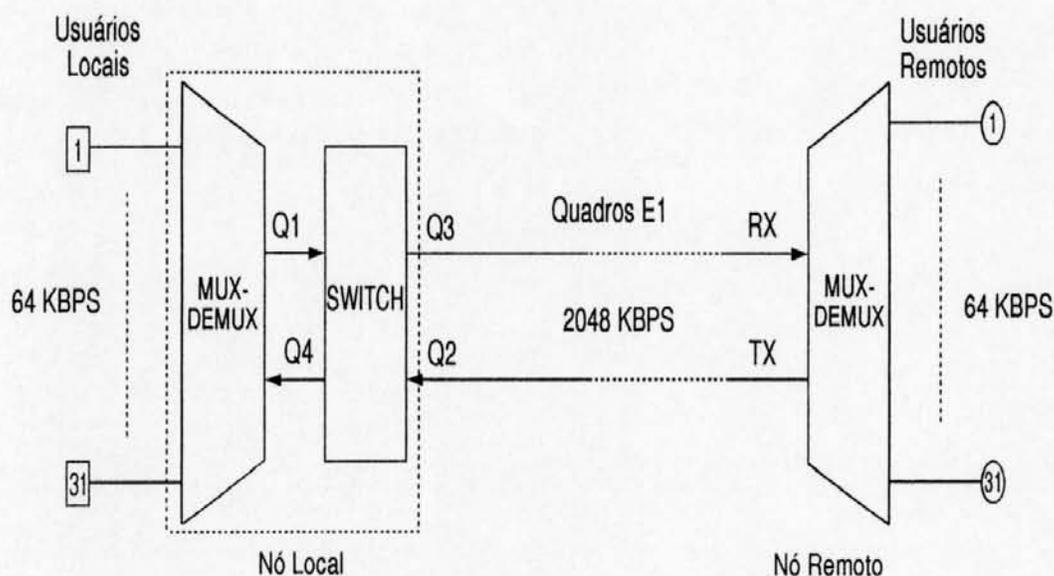


FIGURA 5.2 – Multiplexadores E1 e Comutador Digital.

Como pode ser visto na figura, o Comutador Digital é inserido nas duas linhas do agregado, sendo completamente independente do sistema E1, que continua operando sem modificações.

### 5.1.3 A Arquitetura

Na figura 5.3 é mostrada a arquitetura do Comutador Digital proposto. O mecanismo de comutação é baseado em Divisão de Tempo, usando a técnica TSI de leitura seqüencial e escrita aleatória (vide seção 3.3.1.2). O comutador funciona assim:

Primeiro, os quadros Q1 e Q2 chegam ao comutador e devem passar através do bloco *Conversor Série a Paralelo*, onde serão divididos nos 32 *time-slots* correspondentes, um ao Canal de Serviço e 31 aos Canais Tributários. Neste ponto, é necessário um *buffer elástico* que forneça alinhamento temporal entre os quadros, devido à natureza plesiócrons destes, e que os sincronize em relação ao relógio local [AGU 96]. O quadro Q2 passa pelo *buffer elástico* e a sincronização é obtida: o quadro de saída QD2 é síncrono com Q1 e ambos estão alinhados. Uma vez síncronos e alinhados, cada quadro é dividido nos seus 32 *time-slots*, 64 no total considerando ambos os quadros, e logo estes são escritos em forma paralela na memória SFM

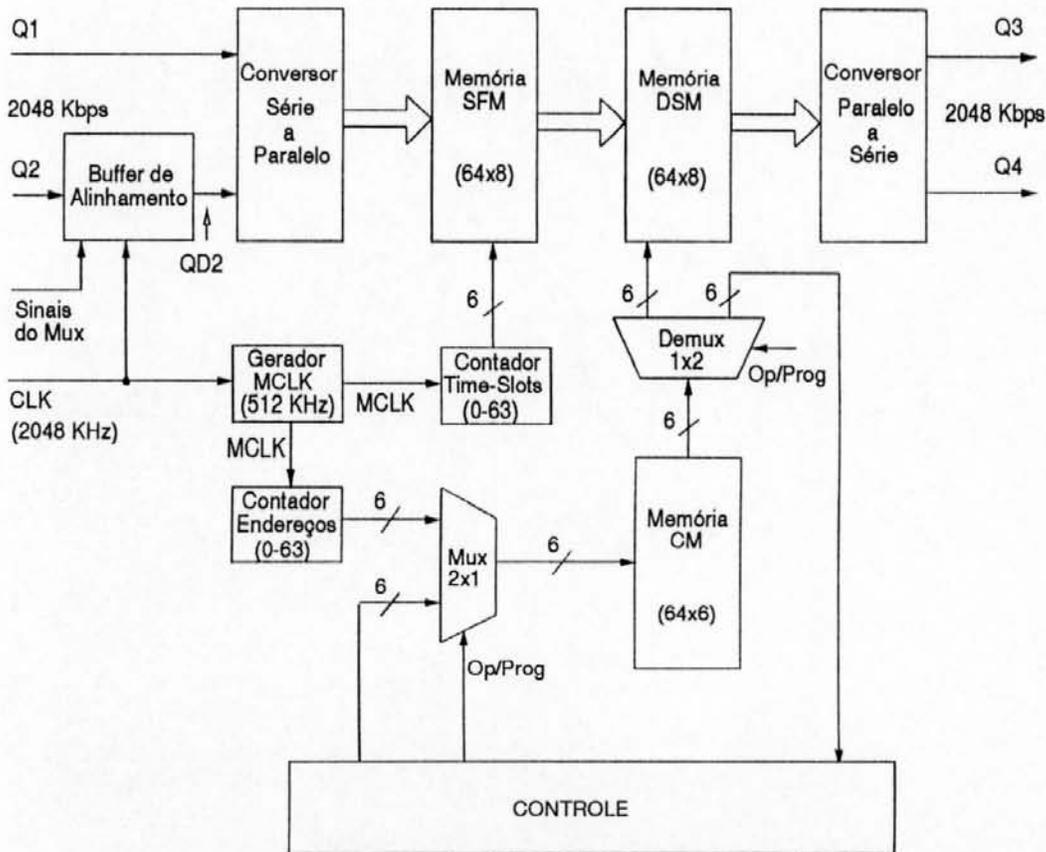


FIGURA 5.3 – Arquitetura do Comutador Digital.

(*Synchronized Frames Memory*). Esta é uma memória de dados de 64 palavras de 8 *bits* cada, e armazena os 32 *time-slots* correspondentes a cada quadro.

Em seguida, os *time-slots* armazenados na memória SFM são lidos seqüencialmente e transferidos aleatoriamente para a memória DSM (*Data Switch Memory*). Esta memória também é de 64 palavras de 8 *bits* cada. O bloco *Contador de Time-Slots* proporciona endereços seqüenciais à memória SFM, e a memória CM (*Connection Memory*) especifica qual endereço da memória DSM deve ser acessado para realizar a função de comutação. A memória CM possui 64 palavras de 6 *bits* cada, e deve ser programada previamente pelo operador através do bloco de *Controle*. As posições da memória CM, estão divididas em duas partes: Inferior (*Low*) e Superior (*High*). Ambas as partes estão associadas com os *time-slots* dos quadros Q1 e Q2 respectivamente. O bloco *Contador de Endereços* permite a leitura seqüencial da memória CM.

Finalmente, a memória DSM é lida simultaneamente pelo bloco *Conversor Paralelo a Série*, para obter os quadros E1 de saída Q3 e Q4. Onde Q3 é o quadro comutado remoto, e Q4 é o quadro comutado local.

Os blocos Contador de *Time-Slots* e Contador de Endereços são contadores síncronos de 6 *bits* cada. O sinal MCLK é o sinal de relógio de 512 KHz, obtido a partir do sinal de relógio local de 2048 KHz, fornecido pelo Multiplexador. Os sinais do Mux representam o conjunto de sinais que são necessários para o alinhamento e sincronização dos quadros. O bloco *Mux 2x1* e *Demux 1x2* permitem ao bloco de controle acessar a memória de conexões (CM) do comutador e programá-la.

É importante ressaltar que as operações de comutação são completamente síncronas, e introduzem um atraso igual a dois quadros E1, isto significa, que o comutador possui uma *latência* de 250 $\mu$ s. O atraso introduzido pelo *buffer* elástico é dependente da diferença de fase existente entre os *bits* de início de cada quadro. No pior caso, os quadros estarão desfasados em 255 *bits*, adicionando ao comutador um atraso de 124.5 $\mu$ s. Em termos práticos, a latência total do comutador não ultrapassa três quadros E1 ou 375 $\mu$ s [AGU 96].

## 5.2 Aspectos de Temporização

Devido às características de sincronismo exigidas tanto para a transmissão como para a comutação digitais, são necessários geradores de pulso ou osciladores, que forneçam tanto a cadência de sincronismo utilizada na formação dos quadros de transmissão multiplexada entre os nós da rede como a referência para a própria comutação.

A seguir, serão descritos alguns detalhes relacionados com a sincronização do comutador, e com o uso do *buffer* elástico no processo de alinhamento dos quadros.

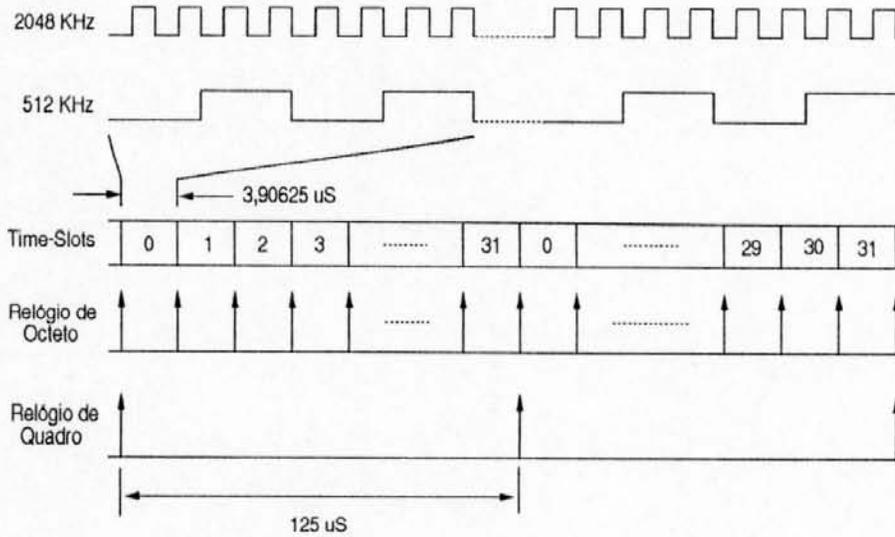


FIGURA 5.4 – Sinais de Relógio para o Computador.

### 5.2.1 Sincronização

A Sincronização Digital é um processo que consiste de manter uma pre-determinada relação de tempo entre os processos ou sistemas digitais [HAR 78]. Todos os sistemas digitais requerem inerentemente uma fonte de frequência ou sinal de relógio como meio de temporizar suas operações internas e externas.

Na hierarquia E1, a temporização e a sincronização são obtidas a partir dos *bits* transmitidos e recebidos. O emissor e o receptor se sincronizam a nível de *byte*, devido a que cada canal transporta 8 *bits* de informação. Portanto, para o correto funcionamento síncrono do comutador pelo menos os seguintes tipos de sinais, mostrados na figura 5.4, devem estar presentes:

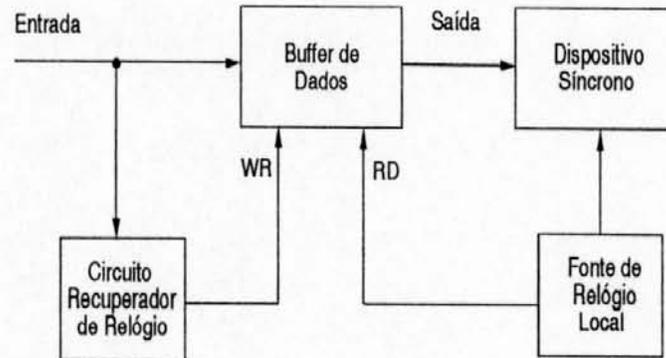
- Um sinal de relógio de 2048KHz, a partir do qual os demais sinais de relógio necessários serão derivados. Este sinal fornece o sincronismo a nível de *bit*. Segundo a *Rec. G.736*, esta fonte de sinal de relógio pode ser obtido a partir de um oscilador interno, de uma fonte externa de sincronismo ou a partir do sinal de relógio recuperado da linha. O equipamento multiplexador é quem deve fornecer este sinal ao comutador a partir de seu módulo Gerador de Base de Tempo como descrito em [KAI 95a].

- Um sinal para temporização de octeto, isto é, que sinalize a ocorrência de *time-slots*. Este sinal é importante pois permite o sincronismo a nível de *byte*.
- Um sinal para temporização de quadro, isto é, que marque o início dos quadros entrantes. Este sinal deve ser utilizado para sincronizar o tempo de deslocamento dos dados através dos conversores e das memórias.
- Um sinal de relógio de 512KHz. O processo de transferência de dados entre a memória SFM e a memória DSM é a nível de *byte*. Dito de outra maneira, os dados são escritos de uma memória para outra preservando a estrutura dos *time-slots* (grupos de 8 *bits*). Visto que, as operações de escrita e leitura nas memórias devem ser feitas em um intervalo de apenas  $125\mu\text{s}$  (tempo de um quadro) e existem 64 *time-slots* envolvidos, concluímos que o tempo de transferência máximo de cada *time-slot* é de  $1.95\mu\text{s}$ . Com o valor anterior calculamos a frequência do sinal de relógio necessário e obtemos 512KHz.

### 5.2.2 Armazenamento Elástico

Todas as fontes de sinal de relógio apresentam uma certa quantidade de instabilidades que, durante um período longo de tempo, pode afetar a integridade da transferência de dados no ambiente de uma rede. As principais fontes de instabilidade de relógio em uma rede são o ruído e interferência, as mudanças na distância média de transmissão e na velocidade de propagação, o efeito Doppler em terminais móveis e informação de tempo irregular. Portanto, geralmente é necessário utilizar sempre que possível, osciladores tão estáveis e econômicos quanto possível em ambos os extremos de um enlace de transmissão.

Contudo, a ocorrência de instabilidades do sinal de relógio, conhecida como *jitter*, nas linhas de transmissão é inevitável, tornando-se uma consideração

FIGURA 5.5 – *Buffer Elástico*.

importante no projeto de enlaces de transmissão digitais, especialmente pela acumulação de *jitter* nos circuitos recuperadores de sinal de relógio.

Quando os pontos finais de um enlace de transmissão trabalham com um sinal de relógio local, a diferença entre o relógio recebido (recuperado) e o relógio local, relativamente estável, deve ser conciliada por meio de técnicas de *Armazenamento Elástico* [BEJ 91].

Básicamente, um *buffer* de dados que é escrito de acordo com um sinal de relógio, e lido a partir de outro sinal de relógio, realiza armazenamento elástico. Se existem pequenas instabilidades em qualquer um dos relógios, o *buffer elástico* absorve as diferenças entre os dados transmitidos e os dados recebidos. Um *buffer* elástico pode apenas compensar pequenas instabilidades, que produzem uma diferença limitada entre os dados recebidos e transmitidos. Se existe uma diferença substancial, como acontece com relógios altamente precisos mas desincronizados, o *buffer* elástico eventualmente se saturará (*overflow*), ou desperdiçará sua capacidade (*underflow*).

A figura 5.5 mostra o diagrama de blocos de um dispositivo capaz de realizar o armazenamento elástico. Na entrada, a informação de temporização do sinal recebido é extraída pelo circuito recuperador de relógio, e é utilizada para escrever os dados no *buffer*, isto é, os dados são escritos no *buffer* usando seu próprio sinal de relógio. No outro extremo, o dispositivo síncrono lerá os dados do *buffer* usando um sinal de relógio gerado localmente. As operações de leitura e escrita são realizadas independentemente, portanto, sob o ponto de vista temporal, a entrada e

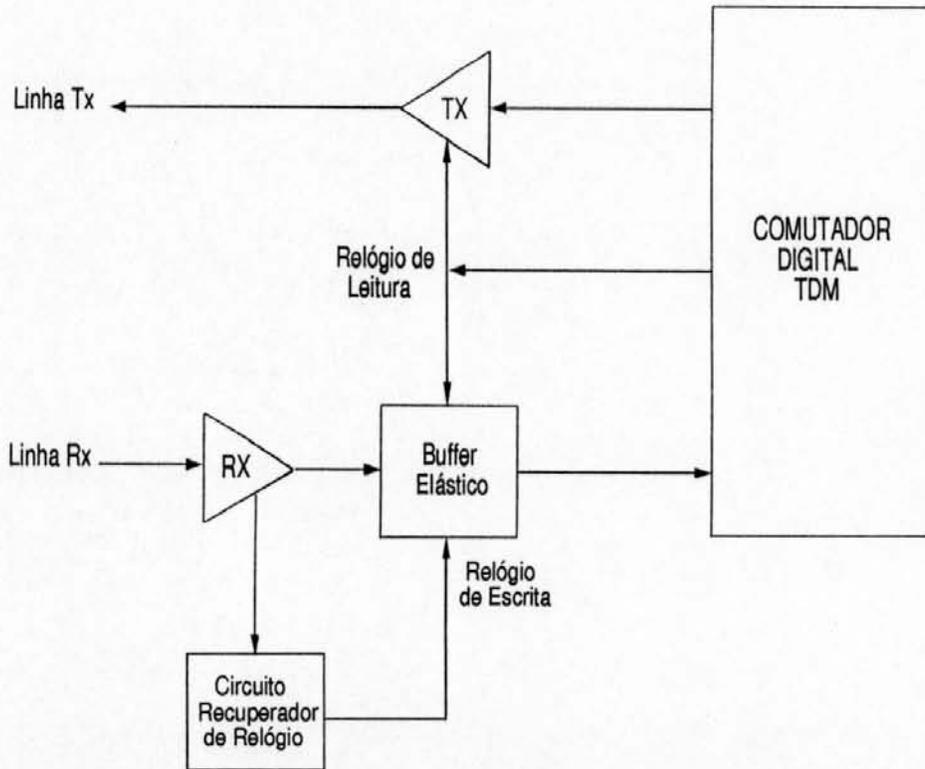


FIGURA 5.6 – Interface com o Comutador TDM.

a saída operam como se estivessem conectadas diretamente uma com outra usando uma fonte comum de temporização.

Se um enlace de transmissão digital é interfaceado com um comutador TDM, o uso de um *buffer* elástico é uma necessidade [BEJ 91]. Nesse caso, a principal função do *buffer* elástico é a de sincronizar o sinal de relógio dos dados recebidos com o sinal de relógio dos dados a serem transmitidos. Considera-se que tanto os dados recebidos como os dados a serem transmitidos são dados binários NRZ sem *jitter*. O *jitter* da linha deve ser extraído pelo multiplexador antes de passar ao comutador. A figura 5.6 descreve a interface do *buffer* elástico com o comutador TDM.

### 5.3 Aspectos de Programação

A programação do Comutador Digital possibilita a efetiva execução de suas funções de comutação. O estabelecimento de enlaces ou conexões entre os

usuários da rede é realizado a partir da informação armazenada na memória de conexões do comutador, portanto, a adequada programação desta memória é essencial.

O modo de programar a memória de conexões, e a forma de acessá-la, serão descritos nos itens seguintes.

### 5.3.1 A Memória de Conexões

A arquitetura do comutador proposta está baseada na técnica TSI. Operações de escrita e leitura de dados de uma memória caracterizam este tipo de dispositivo. A comutação propriamente dita é conseguida mediante o intercâmbio dos *time-slots* correspondentes aos usuários que estão se comunicando. Este intercâmbio é determinado pela informação contida na memória de conexões CM.

Considerando que, o comutador proposto opera mediante a leitura seqüencial da memória SFM, e a escrita aleatória de dados na memória DSM, cada palavra da memória CM deve conter informação acerca da nova posição do *time-slot* associado, e fornece-la seqüencialmente.

A memória CM possui 64 palavras de 6 *bits* cada. O número de *bits* de cada palavra pode ser obtido usando a equação (3.2), porém, é fácil deduzir que são necessários 6 *bits* para poder acessar as 64 palavras da memória DSM. CM é dividida em duas partes iguais: Inferior (*Low*) e Superior (*High*). A parte inferior corresponde aos *time-slots* do quadro Q1, e a parte superior corresponde aos *time-slots* do quadro Q2. Isto é, as palavras 0 a 31 da memória CM representam os *time-slots* 0 a 31 do quadro Q1, e da mesma maneira, as palavras 32 a 63 de CM representam os *time-slots* 0 a 31 do quadro Q2. Portanto, cada palavra da memória CM deve conter informação sobre o destino do *time-slot* respectivo, sendo necessário para isto, indicar o endereço da palavra da memória DSM no qual será escrito.

De forma semelhante, a memória DSM também é dividida em duas partes iguais: Inferior (*Low*) e Superior (*High*), correspondendo estas ao quadro Q3 e ao

quadro Q4 respectivamente. É bom lembrar que Q3 é o quadro que é enviado ao multiplexador remoto, e Q4 é o quadro que é enviado ao multiplexador local.

Na figura 5.7, resume-se a descrição dos passos de programação da memória de conexões para as três possíveis formas de comutação. Após, descreve-se a seqüência de operações de cada possibilidade com mais detalhe.

*Comutação Remota.* Considere que é solicitada a conexão entre o usuário local  $i$  e o usuário remoto  $j$ . A informação do usuário  $i$  deve ser enviada até o multiplexador remoto, portanto, deve ser colocada no quadro Q3. Da mesma forma, a informação do usuário  $j$ , deve alcançar o multiplexador local, isto implica que deve ser colocada no quadro Q4. Para  $i$  acessar  $j$ , seu *time-slot* correspondente deve ser colocado na posição  $j$  do quadro Q3. Assim na palavra  $i$  da memória CM deve estar escrito o valor de  $j$  (passo 1). Agora, como  $j$  está associado ao quadro Q2, sua posição relativa na memória CM é a palavra  $32+j$ ; assim, para conectar-se com  $i$ , esta palavra deve conter o valor  $32+i$  (passo 2).

*Comutação Local-Local.* Considere  $i$  e  $j$  dois usuários locais que desejam se conectar. Ambos os *time-slots* correspondentes a estes usuários estão presentes no quadro Q1, e como ambos têm que retornar ao multiplexador local, devem ser posicionados adequadamente no quadro Q4. Para fazer a troca, o *time-slot*  $i$  é enviado à posição  $j$  do quadro Q4, isto é, a palavra  $i$  de CM deve conter o valor  $32+j$  (passo 1). Igualmente, o *time-slot*  $j$  deve ser enviado à posição  $i$  do quadro Q4. Isto consegue-se armazenando o valor  $32+i$  na palavra  $j$  de CM (passo 2).

*Comutação Local-Remota.* Neste caso, tanto  $i$  quanto  $j$ , são usuários remotos. Seus *time-slots* estão presentes no quadro Q2, e para retornarem ao multiplexador remoto devem ser posicionados no quadro Q3. De maneira semelhante ao caso anterior, a palavra  $32+i$  de CM deve conter o valor  $j$  (passo 1), e a palavra  $32+j$  deve conter o valor  $i$  (passo 2).

Do que foi explicado acima conclue-se:

CHI ←→ CHj

i → Usuário i

j → Usuário j

TS = Time-Slot

COMUTAÇÃO REMOTA

- 1) TS(fonte) = i  
TS(destino) = j
- 2) TS(fonte) = 32 + j  
TS(destino) = 32 + i

COMUTAÇÃO LOCAL-LOCAL

- 1) TS(fonte) = i  
TS(destino) = 32 + j
- 2) TS(fonte) = j  
TS(destino) = 32 + i

COMUTAÇÃO LOCAL-REMOTA

- 1) TS(fonte) = 32 + i  
TS(destino) = j
- 2) TS(fonte) = 32 + j  
TS(destino) = i

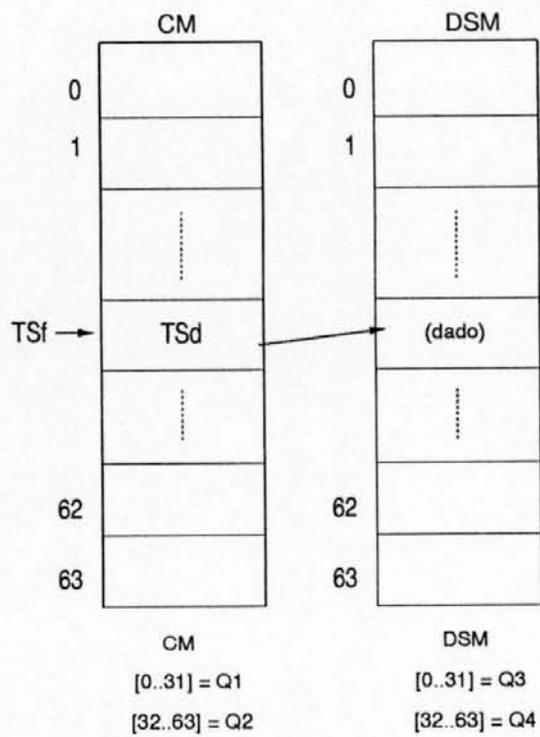


FIGURA 5.7 – Descrição da Programação da Memória de Conexões.

- O estabelecimento dos enlaces deve ser realizado pelo operador da rede ou por um *software* de usuário, portanto, este deve determinar as prioridades no caso em que dois ou mais usuários solicitarem o mesmo enlace.
- O *time-slot* zero ou Canal de Serviço, dos quadros Q1 e Q2, *sempre* deve ser enviado no *time-slot* zero dos quadros Q3 e Q4 respectivamente. Isto significa que, no processo de programação as palavras 0 e 32 da memória CM, já possuem um valor pre-definido em 0 e 32 respectivamente. O *time-slot* zero de cada quadro não é um *time-slot* de usuário, portanto, não pode ser comutado como tal.
- O estabelecimento de um enlace do tipo usuário local  $i$  com usuário remoto  $j$ , não implica necessariamente que o usuário local  $j$  deve conectar-se com o usuário remoto  $i$ . Pelo contrário, o Comutador possibilita que estes usuários possam conectar-se com qualquer outro usuário da rede, seja local ou remoto, sempre que no momento da solicitação do enlace o usuário destino esteja disponível. A decisão sobre a prioridade dos enlaces fica por conta do operador da rede, como foi mencionado anteriormente.
- Quando um usuário determinado não consegue se conectar com o usuário solicitado, deve permanecer em espera até a liberação desse canal, caso não tenha outras opções. O estado de espera se descreve na memória de conexões como uma comutação local do usuário consigo mesmo. Por exemplo: Se o usuário  $i$  do multiplexador local não conseguiu se conectar com um outro usuário, a palavra  $i$  da memória CM, correspondente a este usuário, deve conter o valor  $32+i$ .
- O estabelecimento de laços de teste pode utilizar a estratégia anteriormente descrita, sempre que estes sejam de natureza local ou a nível de quadro completo. Neste último caso, a memória CM deve ser programada para fornecer um *laço total*, isto é, o quadro Q1 deve retornar completo no quadro Q4, ou o quadro Q2 deve retornar completo no quadro Q3. Ou-

tras possibilidades de laço devem ser fornecidas via *hardware* localizado no multiplexador.

- A memória CM é, em princípio, uma RAM que é programada previamente pelo controle do sistema com base na informação enviada pelo operador da rede mediante um terminal de supervisão. Porém, o uso de uma memória do tipo EPROM, no lugar de uma RAM, produz o mesmo resultado.

### 5.3.2 Canal Serial de Comunicação

Este canal deve permitir ao operador da rede enviar dados de programação ao controle do comutador, a partir de um terminal de usuário. Além disso, deve oferecer acesso ao Canal de Enlace de Dados de 20Kbit/s, disponível no Canal de Serviço do multiplexador. O canal de 20Kbit/s possibilita uma sub-rede para troca de mensagens de gerenciamento e controle entre os nós da rede propriamente dita [KAI 95a].

Considerando estas premissas, determinou-se a necessidade de implementar um canal serial síncrono para estabelecer o enlace entre o terminal de usuário e o controle do comutador. Por ser um enlace de comunicação ponto-a-ponto, é necessário o uso de um protocolo de comunicação, de forma que seja possível controlar o sincronismo e a detecção de erros, garantindo a integridade dos dados.

Um protocolo de controle de comunicação relativamente simples, e ainda muito utilizado é o denominado BSC (*Byte Synchronous Communication*) ou *Bisync*, criado pela IBM em 1968 [MON 90]. As características de transparência e o algoritmo CRC para controle de erros deste protocolo, foram fundamentais para sua escolha.

### 5.3.2.1 O Protocolo BSC

Todos os dados em BSC são transmitidos em uma seqüência serial de *bits* 0 e 1. A principal função do BSC é estabelecer um protocolo que permita efetuar a transferência ordenada de caracteres codificados em binário, que contenham as informações da mensagem, eventualmente precedidas por informações de cabeçalho, bem como a identificação da mensagem e o seu destino. Adicionalmente, insere caracteres reservados para o controle da linha, que delimitam as várias porções da mensagem, e também controlam o fluxo da transmissão. O BSC é um protocolo baseado no caractere, e suporta diferentes conjuntos de códigos de transmissão [GIO 86]:

- EBCDIC (*Extended Binary Coded Decimal Interchange Code*)
- ASCII (*American Standard Code for Information Interchange*)
- CBII (Código Brasileiro para Intercâmbio de Informações)
- *Six Bit Transcode*

Cada um dos códigos anteriores oferece diferentes capacidades de assinalamento gráfico e funcional, o que reflete a respectiva flexibilidade, vantagens e desvantagens. Quando o código permite usar a facilidade de transparência, a flexibilidade do enlace de comunicação aumenta sensivelmente.

Quando uma linha opera em um enlace ponto-a-ponto (CPU-Terminal), pode ocorrer a situação denominada *contenção*, onde a CPU e o terminal tentam transmitir simultaneamente. Para evitar essa possibilidade, uma estação faz um convite enviando para a outra estação um caractere de linha de consulta, e que consiste em pedir o controle da linha para poder transmitir uma mensagem ao receptor.

Uma mensagem pode consistir de um ou mais blocos de dados, isto é, uma mensagem pode ser dividida e transmitida por blocos, para facilitar sua ma-

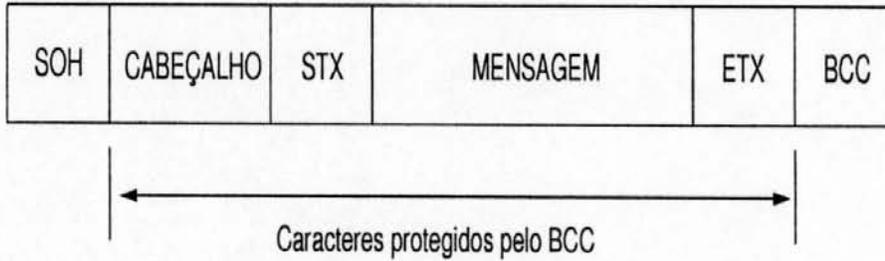


FIGURA 5.8 – Estrutura Típica de Bloco no BSC.

TABELA 5.1 – Caracteres de Controle BSC

Caractere	Significado
SYN	Caractere de Sincronização
SOH	Início de Cabeçalho
STX	Início de Texto
ETB	Fim de Bloco
ETX	Fim de Texto
ENQ	Verifica Estado da Estação
ACK	Reconhecimento Positivo
NAK	Reconhecimento Negativo
DLE	Usado para permitir Transparência
EOT	Fim de Transmissão

nipulação e reduzir a possibilidade de erros de transmissão. A estrutura típica de uma mensagem de informação é mostrada na figura 5.8. BCC (*Block Check Character*) é o nome genérico dado ao *byte* de verificação LRC (*Linear Redundancy Check*) ou CRC (*Cyclic Redundancy Check*) que é inserido durante a transmissão, imediatamente após o fim do bloco de uma mensagem.

Os caracteres de controle e supervisão usados com o protocolo BSC são apresentados na tabela 5.1. Um exemplo de procedimento de transmissão em uma configuração ponto-a-ponto aparece na figura 5.9.

Para obter transparência, ou seja, permitir que qualquer configuração de *bits* seja transmitida pelo usuário, o BSC tem um *modo transparente*. Esta modalidade oferece uma maior versatilidade quanto ao código de dados, pois, neste caso, qualquer *byte*, mesmo os caracteres reservados para o controle da linha, pode ser incluído e transmitido como parte de dado de uma mensagem. Esta facilidade é muito importante para enviar os dados binários, os números em ponto flutuante,

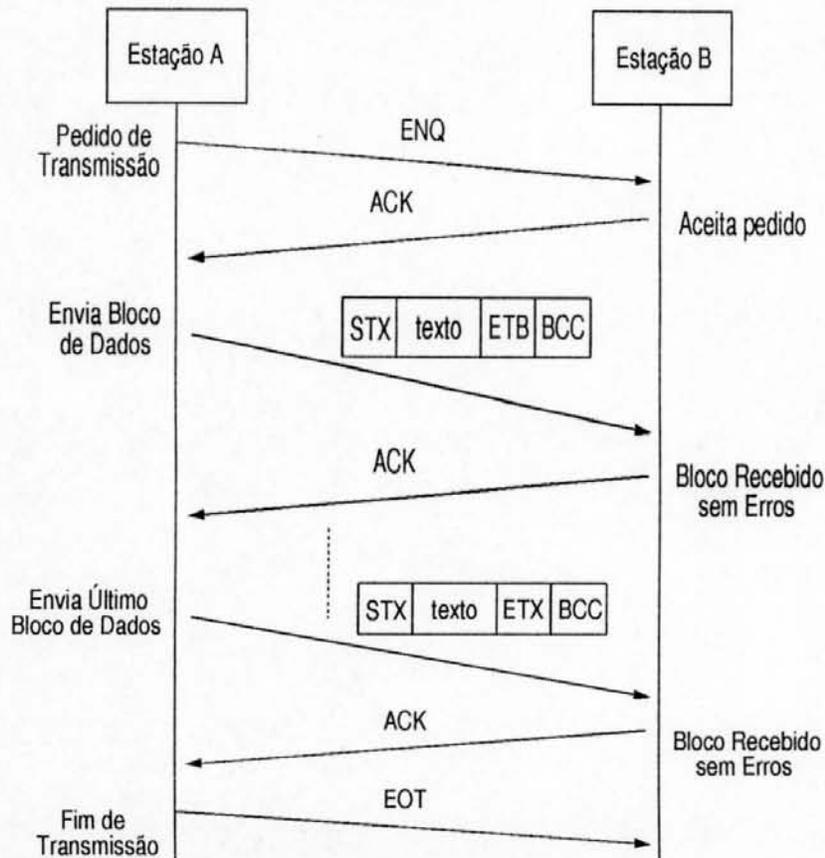
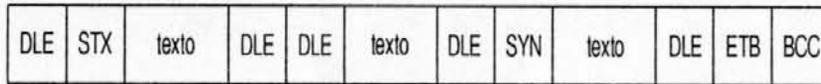


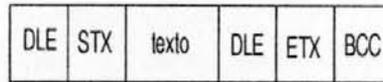
FIGURA 5.9 – Transferência de Informações usando BSC.

os dados numéricos em decimal compactado, os códigos especiais ou o programa de computador em linguagem de máquina [PEN 88]. Qualquer caractere de controle de linha no modo transparente é precedido do caractere DLE para não ser reconhecido como uma seqüência de controle. Caso o texto do usuário contenha o caractere DLE, um DLE adicional é inserido na transmissão, formando a seqüência DLE DLE. Na recepção, todo DLE seguido de outro DLE é retirado e o outro é tratado como texto [GIO 86]. A modalidade de transparência é iniciada pelo caractere de controle SYN SYN seguido de DLE STX e terminada por DLE ETB ou DLE ETX.

Para verificação de erro, os caracteres de verificação CRC-16 (16 bits) são normalmente usados no modo transparente. Exemplo de um bloco transmitido em transparência descreve-se na figura 5.10.



Primeiro Bloco em Transparência



Último Bloco em Transparência

FIGURA 5.10 – Blocos transmitidos em Transparência.

## 5.4 Recomendações ITU-T consideradas

Desde 1968, o Grupo de Estudo XVIII (*Digital Networks*) do Comitê Consultivo Internacional para Telefonia e Telegrafia (CCITT), hoje ITU-T, vem estudando e especificando as características da futura rede digital. Uma ênfase particular tem sido dada às redes de serviços integrados.

Como resultado deste trabalho, um certo número de recomendações foram publicadas nas denominadas *Séries G.700, G.800 e G.900*. A maioria destas recomendações estão dirigidas tanto aos sistemas digitais usados nos Estados Unidos, Canadá e Japão, como aos sistemas digitais usados na Europa e outras partes do mundo. O ITU-T tem dedicado especial atenção às especificações sobre o desempenho dos Sistemas Telefônicos e de Dados, em termos das características da conversão A/D, a qualidade de áudio dos canais PCM, independência das seqüências de *bits* e dos *time-slots* em canais digitais de 64Kbit/s, e nos princípios de comutação.

No Brasil, a Associação Brasileira de Normas Técnicas (ABNT) é a instituição que acolhe as Recomendações ITU-T, e as adapta à realidade do país.

A seguir, faz-se uma descrição das principais recomendações levadas em conta para a implementação do comutador, que em princípio, são as mesmas para o multiplexador. Isto se deve a que o comutador vai formar parte do multiplexador como se fosse um novo módulo do mesmo.

TABELA 5.2 – Hierarquias Digitais ITU-T

Nível	USA (T)	Europa (E)
0	64 Kbit/s	64 Kbit/s
1	1544 Kbit/s	2048 Kbit/s
2	6312 Kbit/s	8448 Kbit/s
3	44736 Kbit/s	34368 Kbit/s
4	97728 Kbit/s	139264 Kbit/s

#### 5.4.1 Rec. G.702 - Hierarquias Digitais e Taxas de *Bit* associadas

Nesta recomendação, o ITU-T define duas hierarquias digitais para multiplexação, cada uma constituída por 4 níveis segundo mostrado na tabela 5.2.

A hierarquia T1 é usada nos Estados Unidos, Canadá e Japão. Por outro lado, no Brasil, a exemplo da Europa, foi adotada a hierarquia E1.

#### 5.4.2 Rec. G.703 - Características Físicas e Elétricas das Interfaces

A Recomendação *G.703* especifica e padroniza as interfaces necessárias à viabilização da interconexão dos componentes de uma rede digital. Assim, para os canais tributários foi definida a interface de 64 Kbit/s e para o agregado definiu-se a interface de 2048 Kbit/s.

Na interface de 64 Kbit/s, três sinais devem ser transportados em ambos os sentidos de transmissão: um sinal de informação a 64 Kbit/s, um sinal de temporização de 64 KHz e um sinal de temporização de 8 KHz. O sinal de informação a 64 Kbit/s corresponde aos dados NRZ codificados segundo a regra *G.703*. O sinal de temporização de 64 KHz é incorporado à seqüência transmitida ao longo da linha, através das transições inter-níveis do sinal codificado. No lado receptor, estas transições possibilitarão a recuperação do relógio de temporização. O sinal de temporização de 8 KHz é gerado a partir da regra de violação do código *G.703*; sua

TABELA 5.3 – Descrição do Quadro E1.

<i>Time-Slot</i>	Tipo de Informação
0	Sincronismo
1 - 15	Voz/Dados
16	Sinalização/Dados
17 - 31	Voz/Dados

finalidade é possibilitar a organização dos dados de tributário em *octetos*, que serão utilizados na composição do quadro de dados a 2048 Kbit/s.

Por outro lado, na interface do agregado, o sinal possui uma taxa nominal de 2048 Kbit/s, com uma tolerância de +/- 50 ppm e combina o sinal de dados com o de sincronismo de *bit* por meio do código *HDB3*, uma variante do código AMI, que visa eliminar algumas características indesejáveis deste último.

#### 5.4.3 Rec. G.704 - Estruturas dos Quadros Síncronos da Hierarquia Digital

Esta recomendação fornece as características funcionais das interfaces associadas com equipamentos multiplexadores digitais. Descreve a estrutura dos quadros, incluindo detalhes sobre o comprimento de quadro, a palavra de alinhamento de quadro e outras informações básicas.

No sistema E1, são transmitidos 32 canais, porém destes, somente 30 são disponibilizados para a transmissão de canais de voz digital, sendo os dois canais restantes, *time-slots* 0 e 16, destinados ao transporte de informação relacionada com a sinalização de serviço e o sincronismo do sistema. Quando se transmite exclusivamente dados, o *time-slot* 16 é utilizado como um canal a mais. A tabela 5.3 resume a distribuição da informação por cada *time-slot*.

A estrutura do quadro básico E1 estabelece um comprimento de quadro de 256 *bits*, numerados de 0 a 255, com uma taxa de repetição de 8 KHz. Portanto, a duração de um quadro básico é de 125 $\mu$ s.

Os primeiros 8 *bits* de cada quadro (canal zero), são usados para transmitir alternadamente a *Palavra de Alinhamento de Quadro* e a *Palavra de Serviço*. As palavras de alinhamento de quadro sincronizam o emissor e o receptor do sistema E1. Os *bits* *b1* e *b7* desta palavra têm sempre o mesmo formato *0011011*. O receptor determina a posição do quadro de pulsos baseado nas palavras de alinhamento de quadro entrantes, e assim consegue distribuir corretamente os *bits* entrantes ao seu respectivo canal.

#### 5.4.4 Rec. G.706 - Procedimentos relacionados com Sincronismo de Quadro e CRC, em estruturas básicas de Quadro

Os procedimentos de controle aplicados nas estruturas de quadro básico T1 e E1, são descritos nesta recomendação. Encontramos estratégias para os casos de Perda de Sincronismo de Quadro, Alinhamento de Quadro e Alinhamento de Multiquadro. É de interesse aqui a Estratégia de Alinhamento de Quadro.

O Alinhamento de Quadro caracteriza o sincronismo entre a transmissão e recepção de quadros. Esta recomendação especifica um procedimento a ser adotado no estabelecimento do sincronismo, a partir do reconhecimento da palavra de alinhamento de quadro.

#### 5.4.5 Rec. G.736 - Características de um Equipamento Multiplex Digital Síncrono, operando a 2048 Kbit/s

Esta recomendação diz respeito às características funcionais dos multiplexadores da hierarquia E1, destacando: Taxa de *Bit* e Tolerância, Características da Fonte do Sinal de Relógio e Condições de Falha e Ações.

## 5.5 Outros Trabalhos

Diversas propostas para a implementação da arquitetura de um comutador de circuitos foram encontradas durante a fase de pesquisa bibliográfica.

Em [SKA 79], são apresentadas varias estruturas como alternativas para implementar Comutadores Digitais baseados em *Time Division*. Redes combinadas de espaço-tempo: T, TST, STS, TSSST, SSTSS e TSTST são descritas a nível funcional, e faz-se uma análise da complexidade do sistema de controle conforme as redes de comutação crescem. Finalmente, são citados alguns problemas de transmissão a serem considerados, e faz-se uma estimativa dos custos envolvidos no projeto dos comutadores com a tecnologia da época. Já em [CHA 79] é mostrada a arquitetura de um bloco básico, denominado DSM (*Digital Symmetrical Matrices*), que possui características combinadas de unidade TSI e matriz *cross-point* (TSSST). Este bloco básico é utilizado como unidade para a implementação de comutadores de grande escala para circuitos de voz. Resultados práticos da implementação de um módulo DSM que atende 8 enlaces PCM de entrada em um circuito integrado LSI também são apresentados.

Uma nova arquitetura para comutadores TSI que trabalham em velocidades da ordem de Gigabit por segundo é proposta em [YAM 86]. A arquitetura está baseada em uma técnica denominada "Escrita Lenta e Leitura Rápida". Nesta técnica o período de tempo de escrita de cada dado é duplicado mediante o uso de mais uma memória de dados. Desta forma, o tempo de acesso às memórias é aumentado sem reduzir a velocidade de comutação, e evita a perda de dados quando se trabalha a taxas de velocidade elevadas. Segundo os autores, tanto o *hardware* requerido quanto a potência total dissipada por esta estrutura é apenas 1.2 vezes maior que a estrutura convencional (TSI). Resultados de implementação não são apresentados. Em [SRI 94], são comparadas as estratégias de comutação existentes para sistemas que usam multiprocessadores e é descrita a arquitetura de um comutador de circuitos adaptivo para redes baseadas em multiprocessadores com uma topologia de interconexão arbitrária.

Até este ponto, com exceção do trabalho descrito em [CHA 79], orientado para canais de voz PCM, vemos que todas as publicações consultadas descrevem apenas alternativas para a arquitetura de comutadores, mas não apresentam implementações.

Por outro lado, na área comercial foram encontrados dois componentes interessantes. Em [MIT 86] estão descritos dois circuitos integrados disponíveis no mercado, que fornecem comutação temporal e espacial para canais de dados ou voz codificados em PCM, sob o controle de um microprocessador. O modelo MT8980D atende 256 canais de 64Kbit/s, isto é, 8 linhas de 2048Kbit/s, enquanto que o modelo MT8981D atende 128 canais de 64Kbit/s cada. Ambos os dispositivos são ISOS-CMOS, encapsulados em pastilhas de 40 pinos. A característica mais interessante é sua capacidade de permitir ao microprocessador acessar, tanto para leitura como para escrita, a cada um dos canais de 64Kbit/s através de sua interface de controle.

Comparado com a arquitetura proposta neste trabalho, existem duas diferenças básicas: primeira, os circuitos integrados de [MIT 86] não apresentam *buffer* elástico na sua entrada, portanto, eles trabalham só com quadros síncronos e alinhados; e segunda, a capacidade de acesso do microprocessador a cada canal de usuário dentro do comutador, para leitura e escrita de dados.

No que concerne ao número de canais a serem atendidos, esta é apenas uma característica do tipo de aplicação na qual o comutador está envolvido. Em outras palavras, a capacidade do comutador proposto aqui pode ser ampliada facilmente, sem alterar suas características funcionais.

## 6 IMPLEMENTAÇÃO DO COMUTADOR DIGITAL

Neste capítulo será apresentada a implementação dos módulos funcionais do Comutador Digital, descritos no capítulo anterior. O Módulo de Controle, que deve ser implementado utilizando o microcontrolador 8051 da Intel, e o Canal Serial Síncrono da Interface serão apenas especificados.

A implementação a nível de portas lógicas, dos circuitos da cada módulo, bem como as simulações dos mesmos, foram realizados dentro do ambiente de projeto *SOLO/Cadence* [CAD 88], usando o estilo de projeto *Standard Cells* e tecnologia CMOS de  $1.2\mu$ . As simulações foram realizadas usando o simulador lógico *SILOS* [SIL 88], disponível no *framework SOLO/Cadence*. Devido a que o *SILOS* modela os transistores CMOS em forma de chaves e considera o efeito de carga das capacitâncias entre as etapas dos circuitos, foi possível avaliar os atrasos em cada etapa e calcular a latência total do Comutador Digital, que é um parâmetro comumente utilizado para avaliar o desempenho de uma arquitetura de comunicações [SRI 94].

Os diagramas esquemáticos em detalhe, dos circuitos correspondentes aos módulos do Comutador Digital apresentados na figura 5.3, aparecem nos anexos A-3, A-4, A-5, A-6, A-7, A-8, A-9, A-10 e A-11.

### 6.1 Geradores de Sinais

No capítulo anterior foi determinada a necessidade de um conjunto de sinais de temporização para o adequado funcionamento do Comutador Digital. A saber: um sinal de relógio de 2048KHz, um sinal de relógio de 512KHz, um sinal de temporização de octeto e um sinal de temporização de quadro. Da mesma forma, um gerador de endereços para permitir o acesso às memórias do Comutador. Em

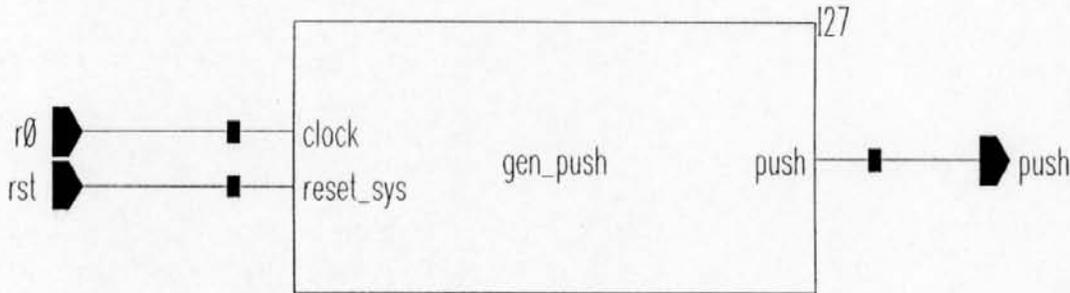


FIGURA 6.1 – Módulo gen-push.

seguida, são apresentados os módulos geradores dos sinais requeridos, com exceção do sinal de relógio de 2048KHz, o qual deve ser fornecido pelo multiplexador local.

### 6.1.1 Gerador de Sinal *Push*

Este módulo encarrega-se de fornecer a temporização de octeto. A figura 6.1 mostra o módulo *gen-push*, que produz um sinal de saída denominado *push*, que consiste em um pulso estreito de largura constante, depois da contagem de 8 *bits* de dados (octeto). O sinal *r0* é o sinal de relógio de 2048KHz fornecida pela fonte de base de tempo local, e o sinal *rst* é o sinal de *reset* do comutador. (Este sinal de *reset* é fornecido pelo *Buffer* de Alinhamento uma vez concluída a tarefa de alinhamento dos quadros como será descrito mais adiante).

Na simulação da figura 6.2 podem ser observados os sinais descritos anteriormente. A contagem do gerador começa logo que o sinal */rst* vai para o nível lógico zero, e considerando os flancos de subida do sinal */r0*. A largura do pulso *push*, que na figura aparece como o sinal */I25/I9/I308.push*, é de 7.7ns.

### 6.1.2 Gerador de Sinal *Save*

O sinal *save* corresponde ao sinal de temporização de quadro. O módulo que fornece este sinal, chamado *gen-save*, é mostrado na figura 6.3. A saída deste módulo é um pulso estreito de largura constante, que indica a ocorrência de um

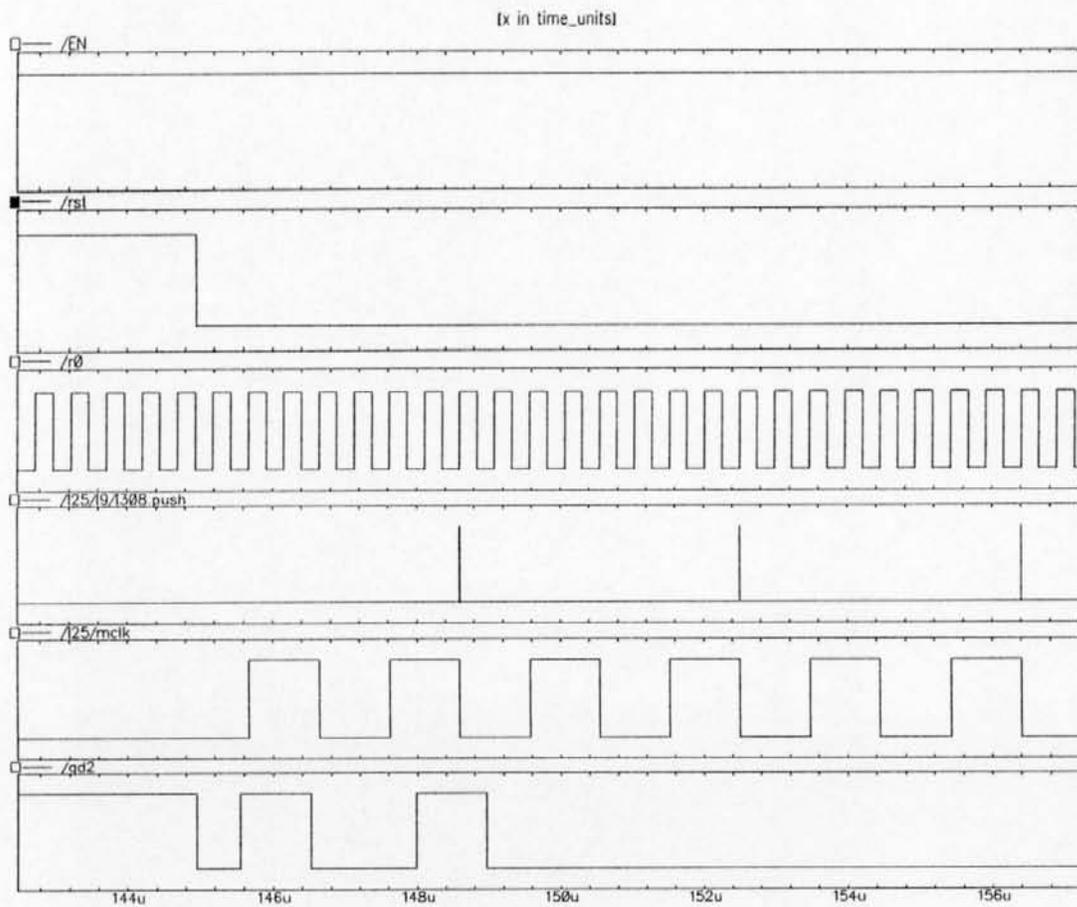


FIGURA 6.2 – Simulação do Gerador de Sinal Push.

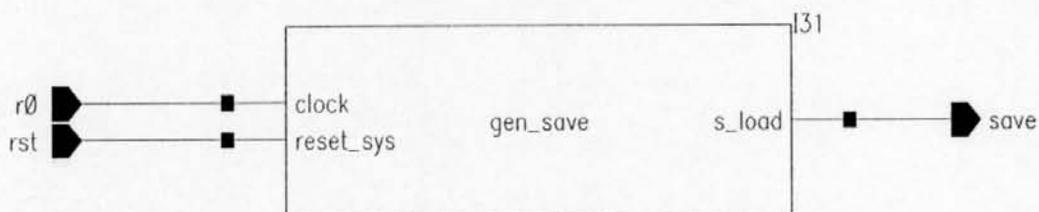


FIGURA 6.3 – Módulo gen-save.

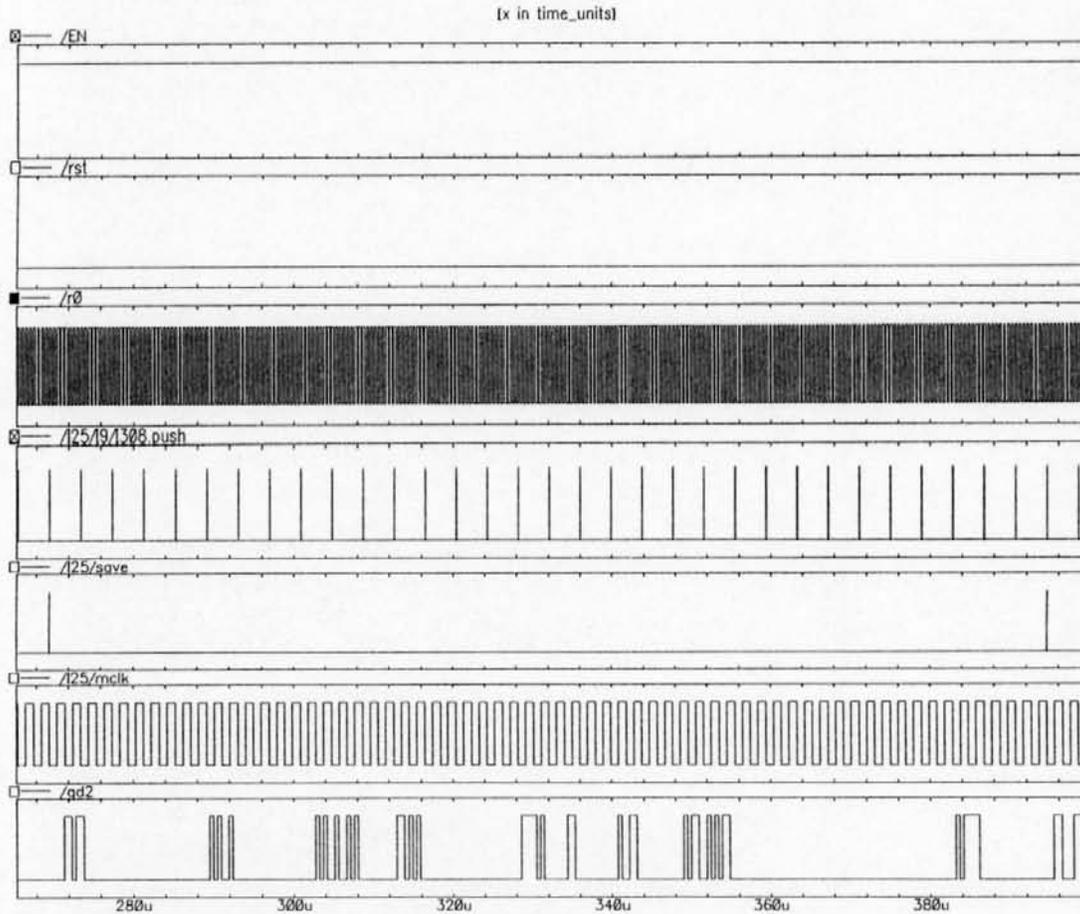


FIGURA 6.4 – Simulação do Gerador de Sinal Save.

quadro completo (256 *bits*), ou seja 32 octetos. O sinal *r0* é o sinal de relógio local de 2048KHz, e o sinal *rst* é o sinal de *reset* do computador.

O sinal *save* é obtido mediante a contagem de 32 pulsos do tipo *push*, como pode ser observado na figura 6.4. A contagem inicia-se só depois que sinal */rst* vai para zero lógico. Os sinais */I25/I9/I308.push* e */I25/save* correspondem aos sinais *push* e *save* respectivamente. A largura do pulso *save* é de 8.7ns.

### 6.1.3 Gerador de Sinal *MCLK*

O módulo *gera-mclk* mostrado na figura 6.5 fornece o sinal *mclk*, que é um sinal de relógio de 512KHz. Este sinal é utilizado para temporizar as operações de leitura e escrita das memórias do Computador Digital. Os sinais de entrada *r0* e

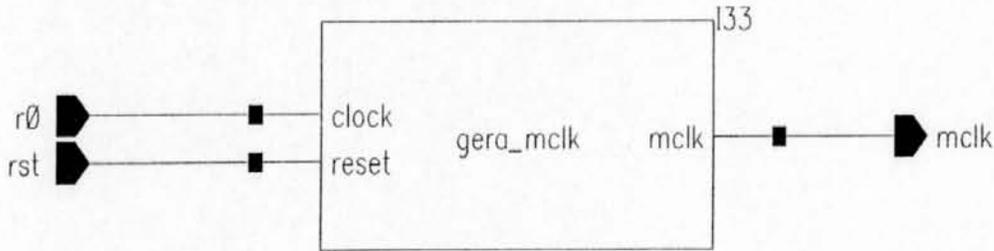


FIGURA 6.5 – Módulo gera-mclk.

*rst*, são o sinal de relógio de 2048KHz e o sinal de *reset* do comutador. O sinal *mclk* é obtido mediante um divisor de frequência a partir do sinal de relógio local.

Na figura 6.6, mostra-se o sinal */I25/mclk* correspondente ao sinal de relógio de 512KHz, e pode ser comparado com os sinais *r0* e *r1*, ambos sinais de relógio de 2048KHz. O sinal *mclk* é um sinal de relógio interno ao Comutador Digital e só está disponível depois que o sinal */rst* toma o valor zero lógico.

#### 6.1.4 Gerador de Endereços

A geração seqüencial de endereços permite o acesso à informação armazenada nas memórias SFM e CM do Comutador. O módulo encarregado de fornecer tais endereços é o *MAC (Master Address Counter)*, contador síncrono de 6 *bits*, mostrado na figura 6.7. Este módulo possui como entradas o sinal *rst* (*reset* do comutador), e o sinal de relógio *mclk*. Na saída, entrega uma palavra de 6 *bits* capaz de endereçar as 64 posições de cada memória. O *bit ad.ctl5* representa o MSB, enquanto que o *bit ad.ctl0* representa o LSB.

A figura 6.8 apresenta as saídas do módulo *mac*, */ad.ctl5* até */ad.ctl0*, temporizadas pelo sinal de relógio */I25/mclk*, e habilitadas a partir do valor zero do sinal */rst*. Os endereços são gerados ciclicamente a cada 125 $\mu$ s.

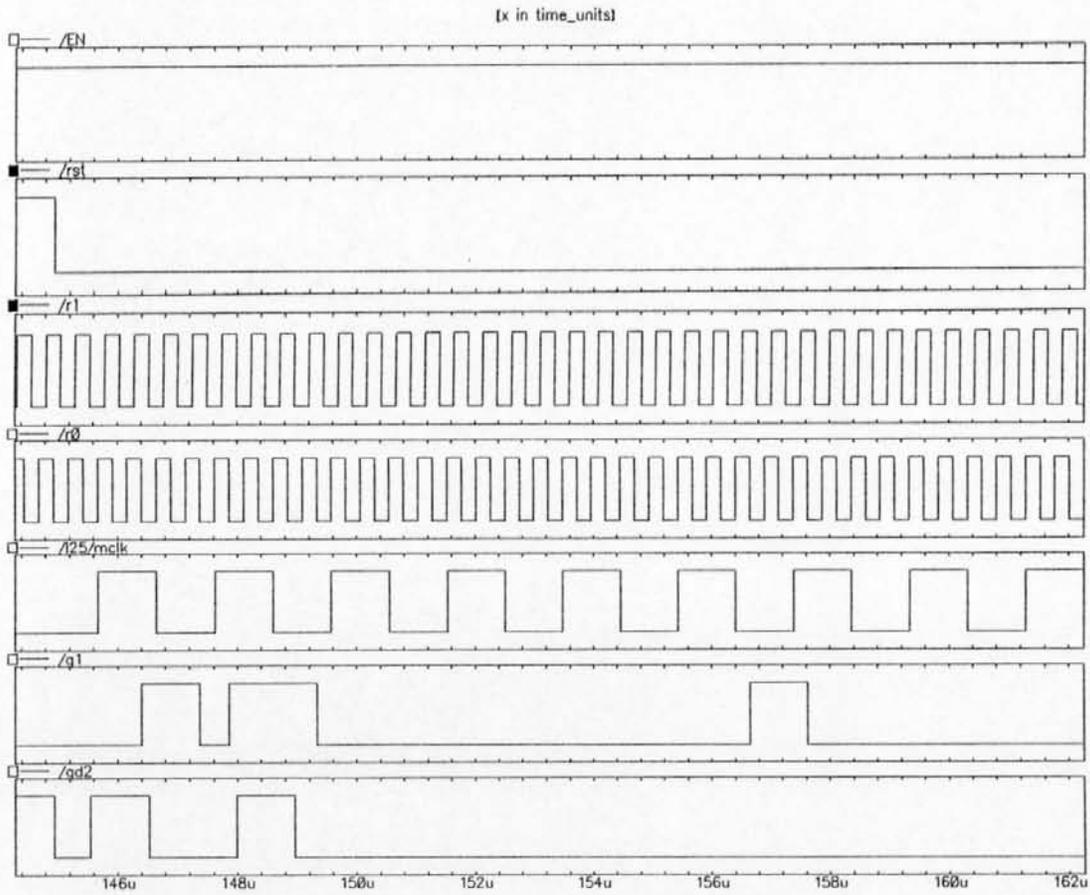


FIGURA 6.6 – Simulação do Gerador de Sinal MCLK

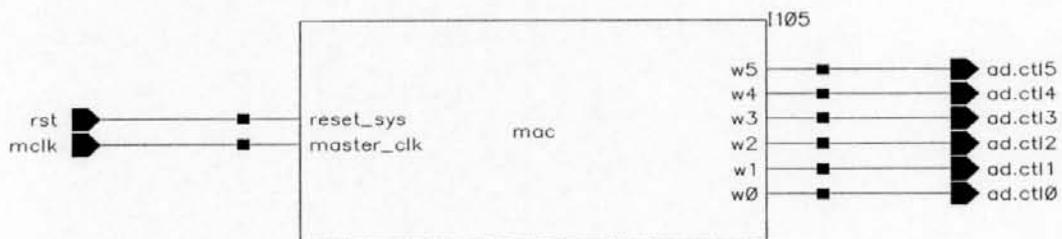


FIGURA 6.7 – Módulo mac.

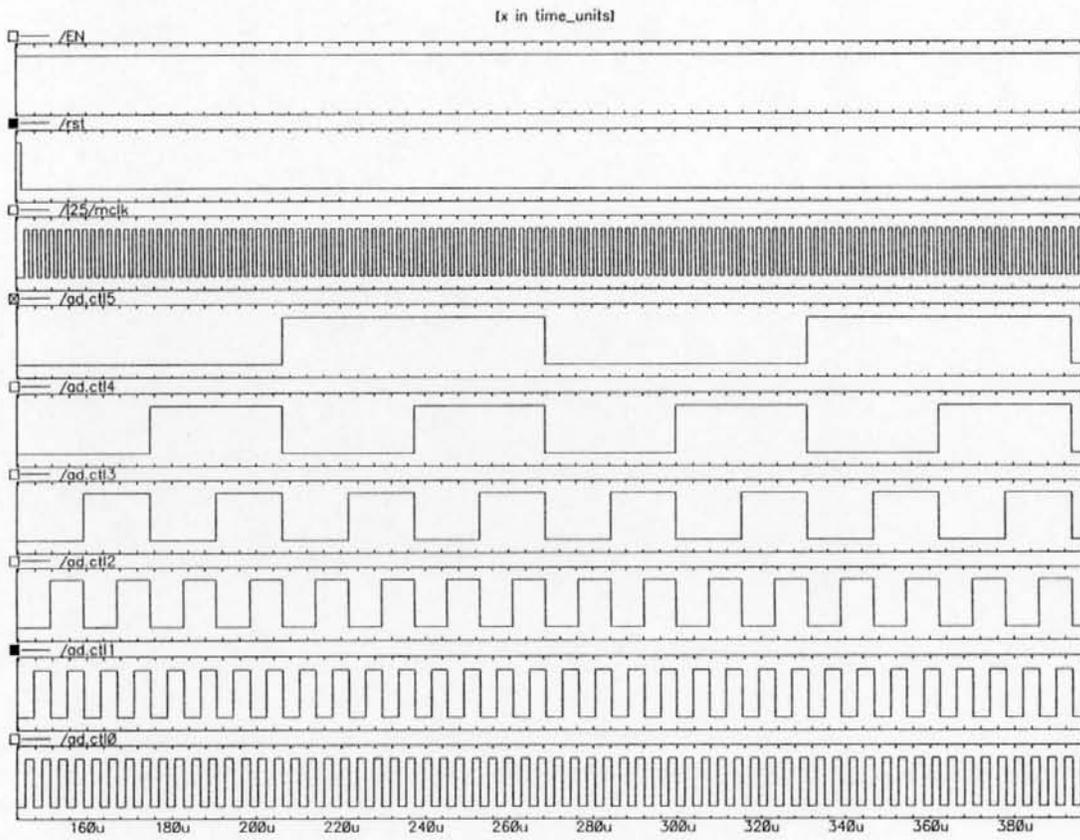


FIGURA 6.8 – Simulação do Gerador de Endereços

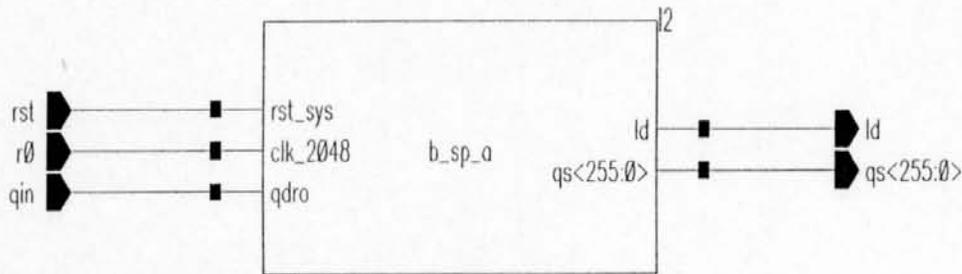


FIGURA 6.9 – Módulo b-sp-a.

## 6.2 Conversores

Uma das primeiras tarefas a serem realizadas antes de iniciar o intercâmbio de dados, é a desmontagem dos quadros E1 de entrada em suas unidades básicas ou *time-slots*. Concluída a re-alocação dos *time-slots*, a montagem dos "novos" quadros E1 de saída é necessária. Os módulos responsáveis pela montagem e desmontagem dos quadros E1, denominados conversores de Série a Paralelo e de Paralelo a Série, são apresentados a seguir.

### 6.2.1 Conversor Série a Paralelo

O módulo conversor de dados Série a Paralelo para quadros E1 de 256 *bits*, chamado *b-sp-a*, é mostrado na figura 6.9. Este módulo recebe como entradas o sinal *rst*, o sinal de relógio *r0* de 2048KHz, e o quadro E1 de entrada *qin*. Na sua saída produz um vetor paralelo de 256 *bits* formado pelos dados do quadro de entrada. Também fornece o sinal *ld*, que é um pulso estreito que ocorre cada vez que uma operação de conversão de um quadro é completada.

Na figura 6.10, o circuito deste conversor é descrito a nível de blocos. O funcionamento do circuito baseia-se no armazenamento seqüencial dos *time-slots* de quadro E1 de entrada em um conjunto de 32 registradores de 8 *bits*, a partir dos quais podem ser lidos de forma paralela. Primeiro, o quadro E1 de entrada *qin* passa através de um registrador de deslocamento de 8 *bits*. O bloco *gen-push*

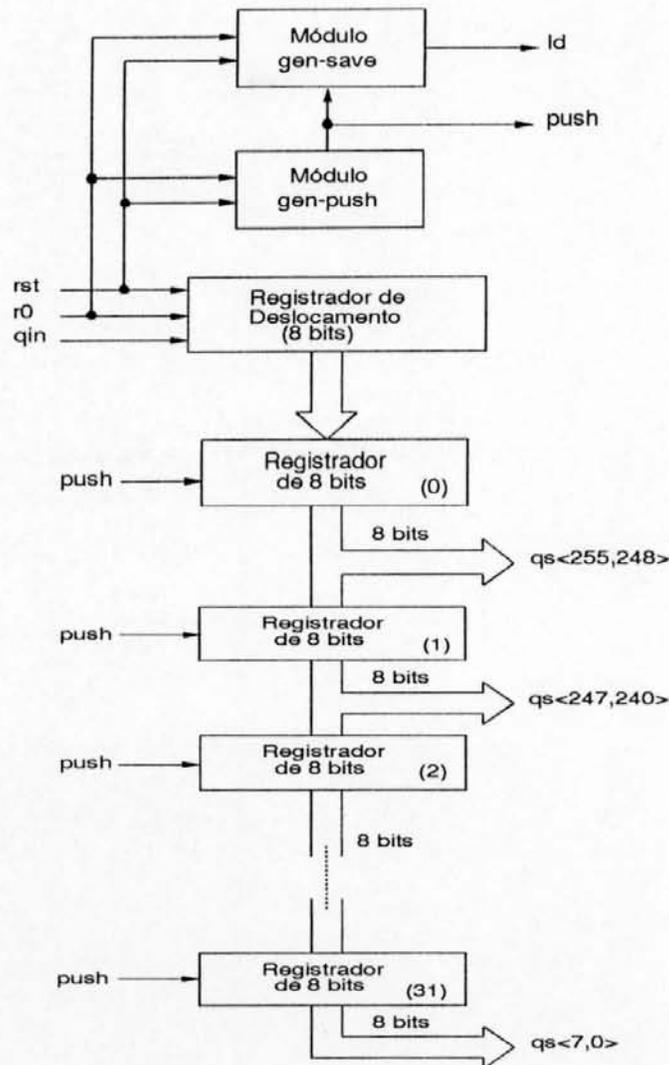


FIGURA 6.10 – Diagrama de Blocos do Conversor Série a Paralelo.

conta o número de *bits* entrantes, e gera um pulso de saída *push* a cada ocorrência de um octeto. Portanto, a cada 8 *bits*, 1 *time-slot*, o conteúdo do registrador de deslocamento é transferido ao registrador de 8 *bits* adjacente, e o conteúdo deste último ao registrador seguinte e assim sucesivamente. Entretanto, o bloco *gen-save* vai contando o número de pulsos *push* até alcançar os 32 que equivalem a um quadro E1 completo. Uma vez contados 32 pulsos *push*, gera um pulso de saída *ld* (sinal *save*) que indica que as saídas *qs* do conversor estão disponíveis para sua transferência à etapa seguinte do Comutador Digital. O processo é cíclico e se repete a cada 125 $\mu$ s. O Comutador precisa de dois destes conversores para atender aos dois quadros E1 de entrada.

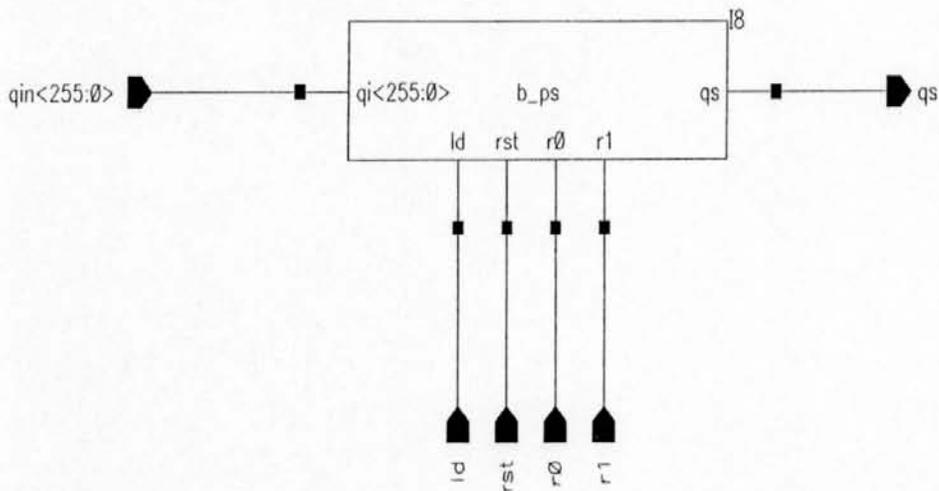


FIGURA 6.11 – Módulo b-ps.

### 6.2.2 Conversor Paralelo a Série

Este conversor encarrega-se da montagem dos quadros E1 de saída que contêm os dados comutados. A figura 6.11 mostra o módulo *b-ps*, que recebe um vetor paralelo de entrada de 256 *bits*, e produz uma saída serial de dados. Os sinais necessários para realizar a conversão são um sinal de *rst* (do Comutador), um sinal de carga *ld*, e dois sinais de relógio de 2048KHz *r0* e *r1*. O sinal *r1* tem sua fase atrasada 90 graus em relação ao sinal *r0*.

O conversor recebe em paralelo os dados provenientes da memória DSM, que correspondem a um quadro com os *time-slots* já comutados, e os armazena em um bloco de *latches* para depois amostrá-los seqüencialmente, a maneira de uma chave rotativa, e montar um quadro E1 de saída. A figura 6.12 mostra o circuito a nível de blocos do conversor. Os *bits* do quadro *qin* chegam ao conversor em paralelo desde a memória DSM e são armazenados no bloco de *latches* depois de um pulso de carga *ld*, fornecido pelo conversor Série a Paralelo (saída *ld*). O módulo *cont-256* é um contador síncrono de 8 *bits* que se encarrega de selecionar seqüencialmente as entradas do multiplexador. O sinal de relógio *r0* de 2048KHz, é a base de tempo do contador e sincroniza a amostragem. O bloco *FFD* representa um *flip-flop* tipo D que é utilizado para evitar a presença de espículas no quadro E1 de saída provocadas pelo chaveamento. O uso do sinal de relógio *r1*, para habilitar o *flip-flop* D garante

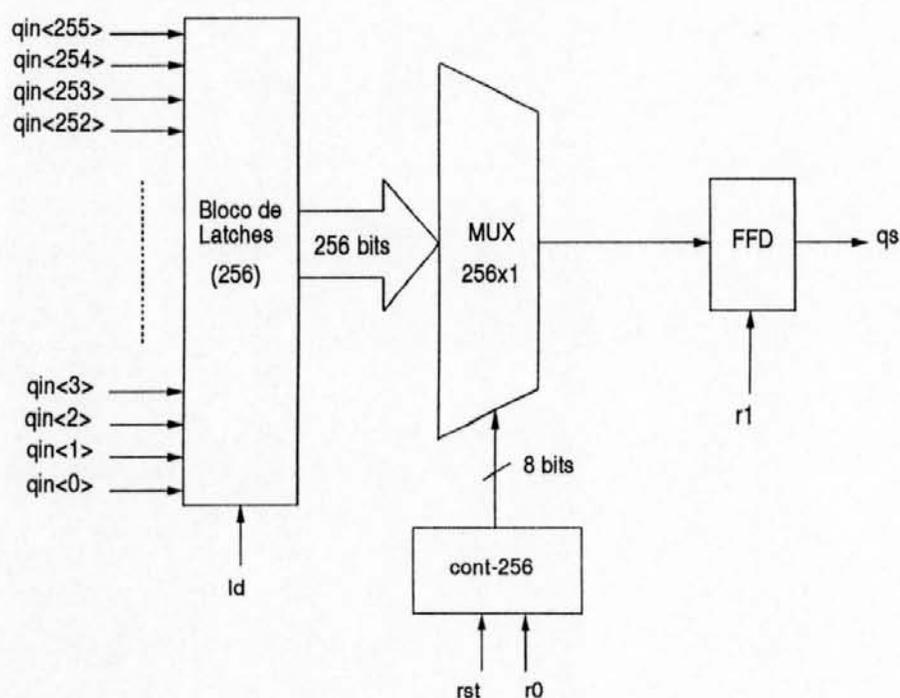


FIGURA 6.12 – Diagrama de Blocos do Conversor Paralelo a Série.

a estabilidade do sinal  $qs$  na saída. O processo de conversão é cíclico e se repete a cada  $125\mu s$  (o tempo que dura um quadro E1). No Computador Digital são utilizados dois destes conversores para realizar a montagem dos dois quadros E1 de saída.

### 6.3 Memórias

A função básica do Computador Digital TDM consiste na alocação dos dados de um determinado *time-slot* em um *time-slot* diferente, seja no mesmo quadro E1 ou em outro quadro E1. Para realizar esta função os *time-slots* de ambos os quadros de entrada devem ser previamente armazenados para que, no próximo ciclo de quadro, depois de ser re-allocados, possam ser enviados no quadro E1 de saída correto. Da mesma maneira, a informação de roteamento deve estar previamente armazenada, e ser acessível ao Computador para a efetiva execução das funções de comutação.

Duas memórias internas (SFM e DSM), e uma memória externa (CM) ao Computador Digital, são as responsáveis pelo armazenamento dos dados de usuário

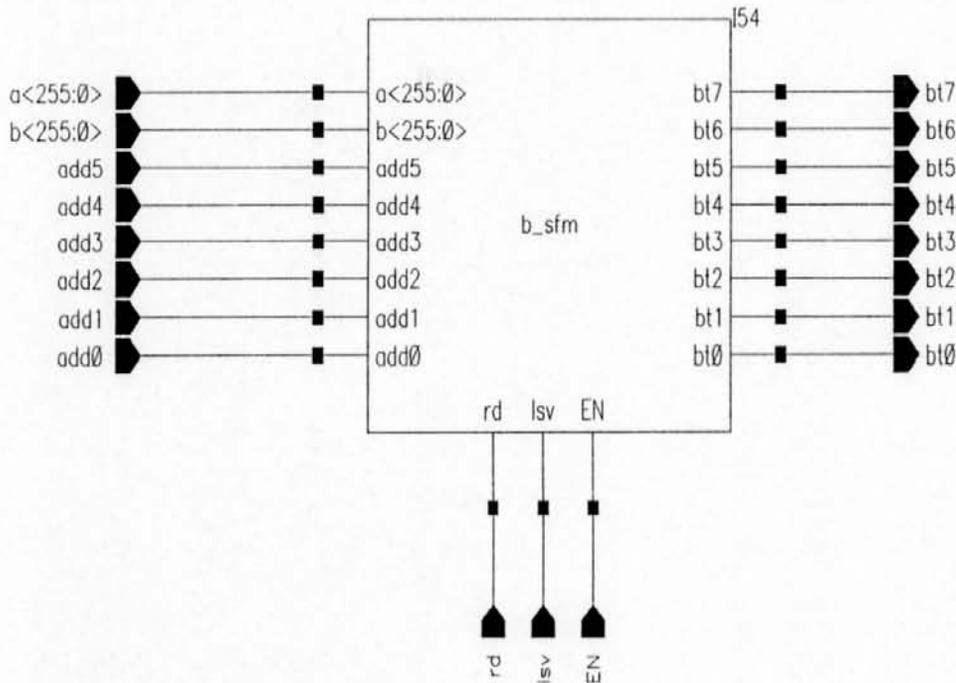


FIGURA 6.13 – Módulo b-sfm.

e dos dados de controle, respectivamente. A implementação destas memórias é apresentada a seguir.

### 6.3.1 Memória SFM

O módulo *b-sfm*, mostrado na figura 6.13, corresponde à memória SFM (*Synchronized Frame Memory*). Esta memória possui 64 palavras de 8 *bits* cada, que podem ser escritas em forma paralela mediante um único pulso de escrita, e lidas individualmente como uma memória convencional. As entradas *a(255:0)* e *b(255:0)* correspondem aos dois barramentos de 256 *bits* cada, através dos quais os *bits* de dados dos quadros E1 sincronizados são carregados de forma paralela. Os *bits* de entrada *add5* até *add0* correspondem à palavra de endereço de 6 *bits* que permite o acesso a cada posição da memória SFM, sendo que o *bit add5* é o MSB, e o *bit add0* é o LSB. Na saída, uma palavra de 8 *bits*, *bt7* até *bt0*. A entrada *EN* é o sinal de *enable* que habilita a memória, e as entradas *rd* e *lsv* são o sinal de leitura e de escrita respectivamente. O sinal *lsv* é fornecido pelo bloco Conversor Série a Paralelo através de sua saída *ld* (sinal *save*).

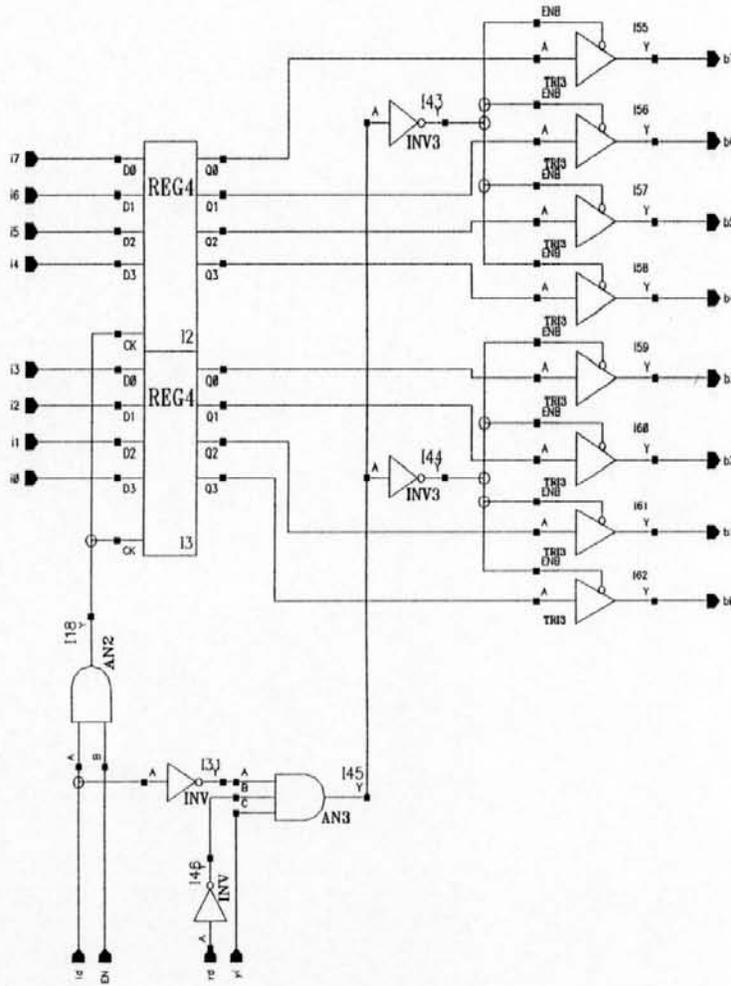


FIGURA 6.14 – Unidade de Memória Básica SFM.

Na figura 6.14 é apresentada a unidade de memória básica de SFM. Esta unidade corresponde a um *byte* de informação (8 *bits*). O sinal  $yi$  é o resultado da decodificação do endereço correspondente à posição de memória atual. Observa-se que, os sinais  $yi$  e  $rd$  permitem a leitura do conteúdo da memória, sempre que não esteja sendo realizada uma operação de escrita no mesmo instante (sinal  $ld$  em nível lógico 1). Desta maneira evita-se que uma operação de escrita e uma operação de leitura ocorram simultaneamente. O sinal  $ld$  é o sinal de temporização de quadro (sinal *save*) proveniente do conversor Série a Paralelo.

A simulação de uma operação de leitura da posição 32 da memória SFM, mostra-se na figura 6.15. O sinal  $/I25/I2/I3/I394.Y$  representa o sinal  $rd$ , que habilita a operação de leitura a uma frequência de 512KHz. Desta forma, consegue-se ler toda a memória em um período de 125 $\mu$ s. O sinal  $/I25/I2/y(32)$  é o resultado da

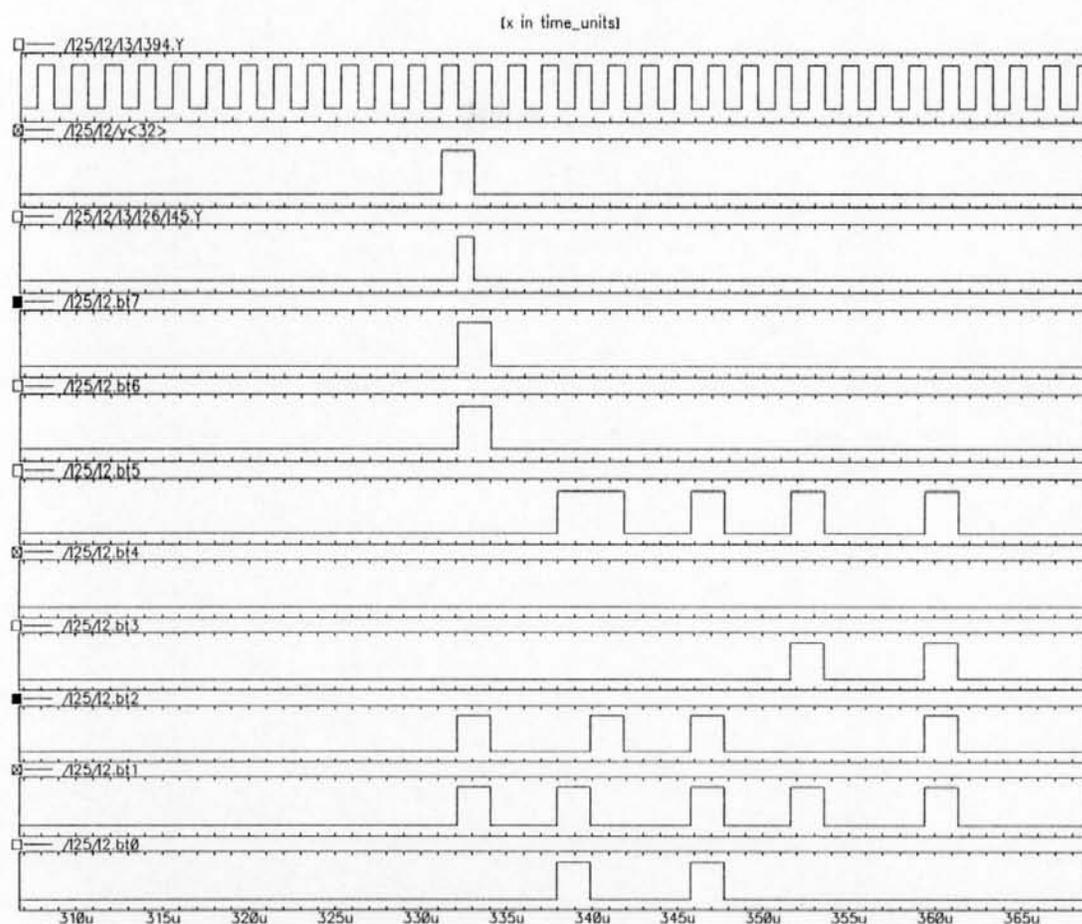


FIGURA 6.15 – Simulação da Memória SFM.

decodificação do endereço de memória correspondente à posição 32. A combinação destes sinais através de uma porta lógica *AND* (ver porta lógica *AN3* na figura 6.14), produzem o sinal de habilitação */I25/I2/I3/I26/I45.Y* que realmente disponibiliza o conteúdo da posição de memória 32 (01100011) nas saídas */I25/I2.bt7* até */I25/I2.bt0*. Esta estratégia é utilizada para evitar a ocorrência de espículas provocadas pelas transições dos sinais de decodificação, quando da leitura seqüencial da memória SFM. A informação armazenada na memória SFM é lida seqüencialmente, *time-slot a time-slot*, e fica disponível para sua transferência à memória DSM.

### 6.3.2 Memória DSM

A memória DSM (*Digital Switch Memory*) recebe os dados da memória SFM e os aloca nas posições determinadas pelos endereços fornecidos pela memória CM. Assim, depois de um período de quadro igual a  $125\mu\text{s}$ , a memória DSM contém

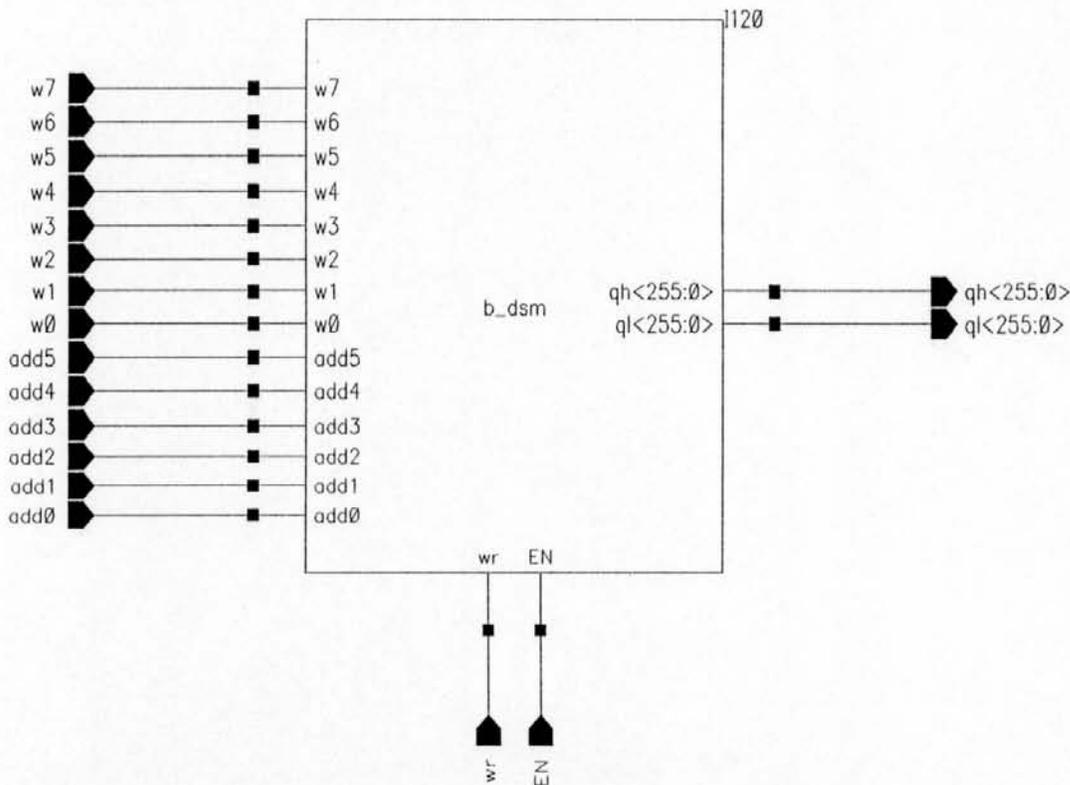


FIGURA 6.16 – Módulo b-dsm.

efetivamente os quadros E1 comutados de saída. Esta memória possui 64 palavras de 8 *bits* cada, que podem ser escritas individualmente como uma memória convencional, e lidas em forma paralela. A memória DSM está dividida em duas partes iguais, a parte inferior (*Low*) armazena os *time-slots* correspondentes ao quadro de saída Q3, e a parte superior (*High*) armazena os *time-slots* correspondentes ao quadro de saída Q4. O módulo *b-dsm*, mostrado na figura 6.16, corresponde à memória DSM. As entradas *w7* até *w0*, e *add5* até *add0* correspondem à palavra de dados de entrada, e à palavra de endereços respectivamente. Os *bits* de índice maior representam o MSB, e os *bits* de índice menor o LSB. Na saída, os vetores *qh* e *ql*, disponibilizam os 64 *time-slots* armazenados na memória DSM, correspondentes aos quadros já comutados, para sua leitura em forma paralela pela etapa seguinte. A entrada *EN* é o sinal de *enable* que habilita a memória, e a entrada *wr* é o sinal que controla a operação de escrita.

Na figura 6.17 é apresentada a unidade de memória básica de DSM. Esta unidade corresponde a um *byte* de informação (8 *bits*). O sinal *yi* é o resultado da

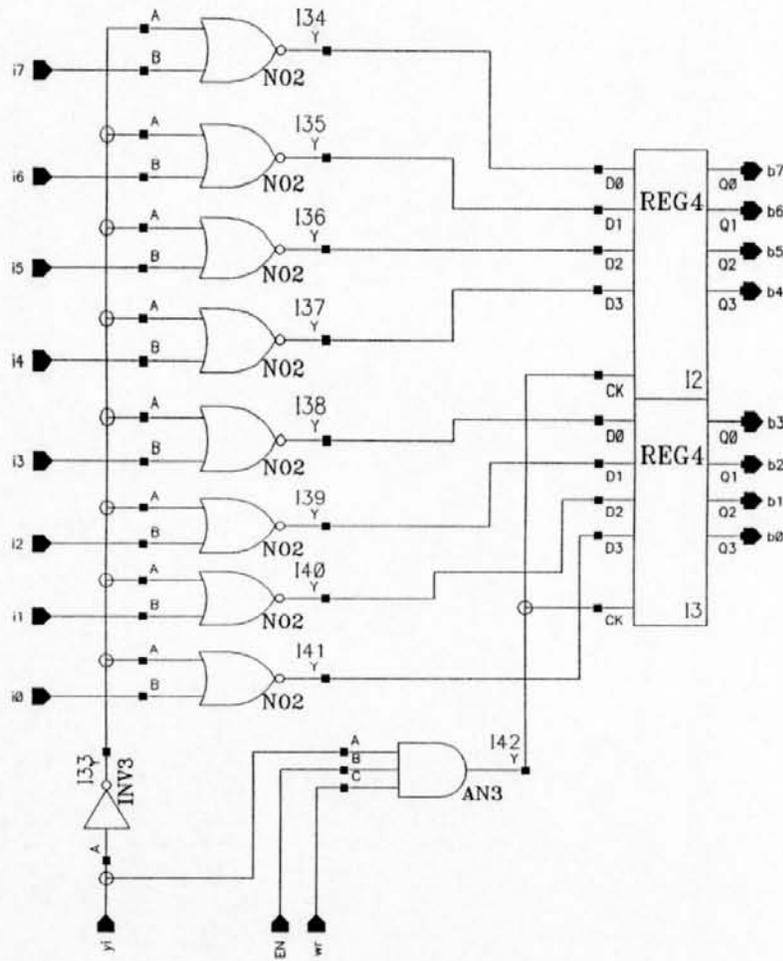


FIGURA 6.17 – Módulo u-dsm.

decodificação da palavra de endereço correspondente à posição de memória atual. Observa-se que, os sinais  $y_i$  e  $w_r$  permitem a escrita de dados na memória, sempre que a memória esteja habilitada (sinal  $EN$  em nível lógico 1). No caso que a posição de memória atual não seja acessada, as portas lógicas  $NOR$  colocam em nível lógico zero as entradas dos registradores, porém, o conteúdo destes não é alterado, e suas saídas sempre estão disponíveis para leitura.

A simulação da seleção de escrita da posição 32 da memória DSM, mostra-se na figura 6.18. O sinal  $/I_{25}/I_4/I_3/I_{110}.Y$  representa o sinal de escrita  $w_r$ , que está disponível a uma frequência de 512KHz. Desta forma, consegue-se escrever em toda a memória em um período de  $125\mu s$ . O sinal  $/I_{25}/I_4/y(32)$  é o resultado da decodificação do endereço de memória correspondente à posição 32. (Como mencionado anteriormente, a memória CM fornece tal endereço segundo os dados nela armazena-

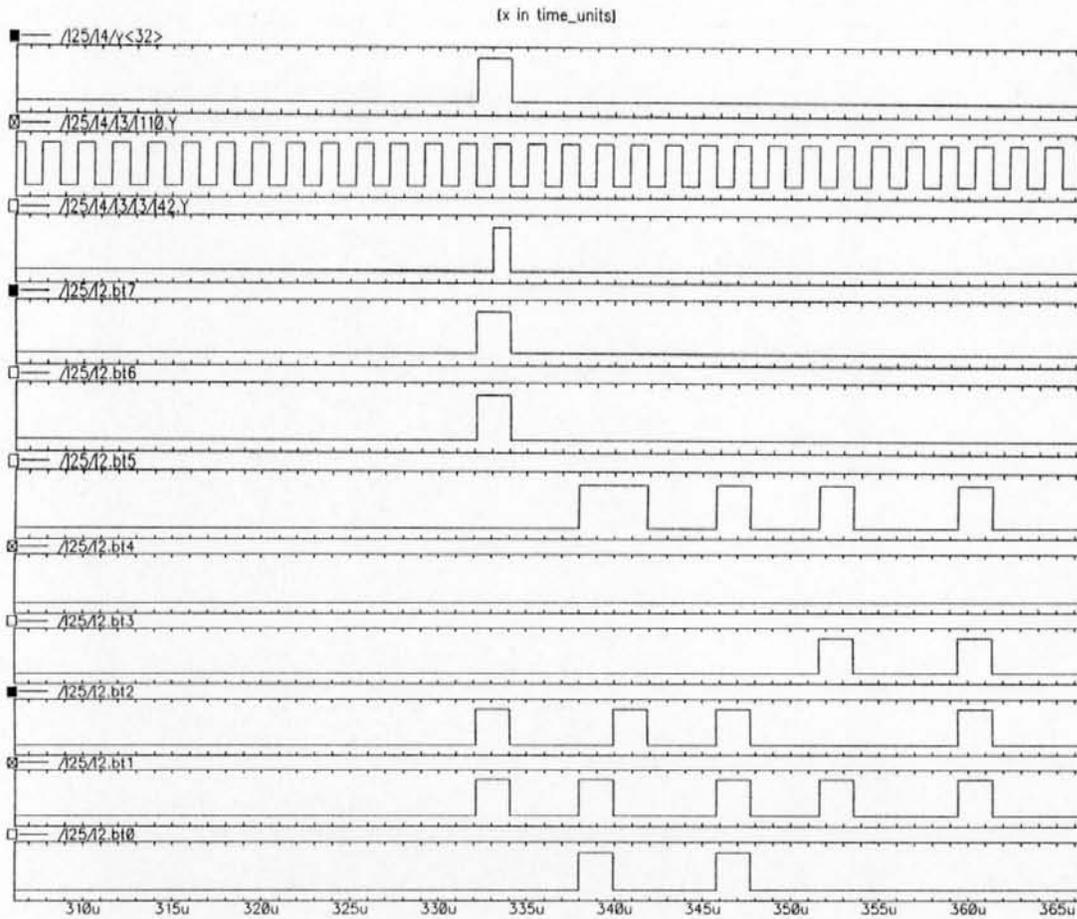


FIGURA 6.18 – Simulação da Memória DSM.

dos). A combinação destes sinais através de uma porta lógica *AND* (ver porta lógica *AN3* na figura 6.17), produzem o sinal de habilitação */I25/I4/I3/I3/I42.Y*. Este sinal habilita a escrita dos dados presentes nas entradas */I25/I2.bit7* até */I25/I2.bit0* durante a metade do tempo de um *bit* (aproximadamente  $1\mu\text{s}$ ). Esta estratégia é utilizada para garantir que os dados a serem escritos estejam estáveis nas entradas, e evitar a superposição. A cada  $125\mu\text{s}$  a informação armazenada na memória DSM é lida em forma paralela, pela etapa seguinte, o conversor Paralelo a Série.

### 6.3.3 Memória CM

O estabelecimento de enlaces ou conexões entre os usuários da rede é realizado a partir da informação armazenada na memória CM (*Connection Memory*). O Computador Digital opera mediante a leitura seqüencial dos *time-slots* armazenados na memória SFM, e a escrita aleatória destes na memória DSM. A escrita na

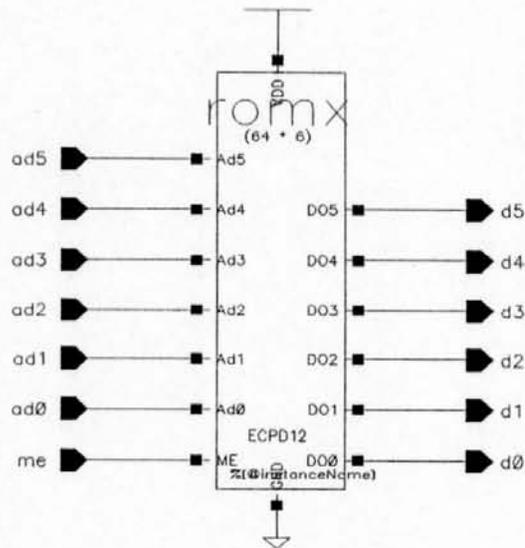


FIGURA 6.19 – Memória romx.

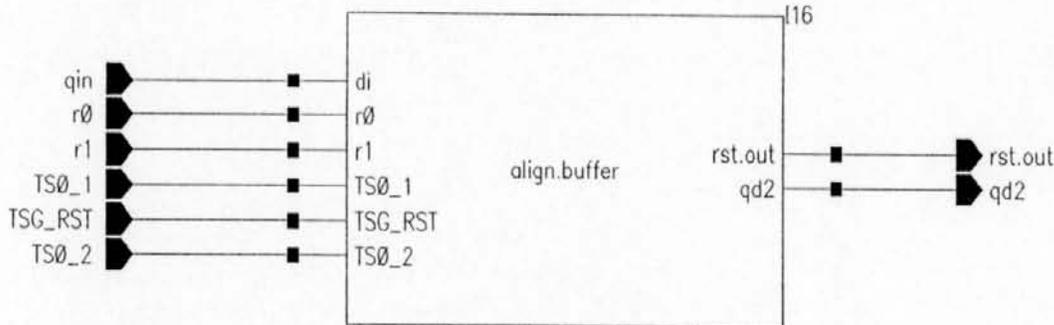
memória DSM é controlada pela memória CM que contém informação acerca da nova posição dos *time-slots* realizando desta forma a comutação. Cada palavra da memória CM está associada a um determinado *time-slot* armazenado em SFM, e contém o endereço do novo destino em DSM.

Para realizar as simulações do Comutador Digital foram utilizadas memórias tipo ROM com 64 palavras de 6 bits cada, geradas usando o Compilador de Macrocelulas (*Macrocells Compiler*) disponível no ambiente de projeto *SOLO/Cadence*. A figura 6.19 mostra a memória *romx*, usada nas simulações da arquitetura completa do Comutador Digital, cujos resultados são apresentados mais adiante.

Na prática, a memória CM deve ser implementada usando uma RAM 64x6, de maneira que possa ser programada pelo controle do Comutador Digital. Isto não impede que seja utilizada uma EPROM no seu lugar, obtendo-se os mesmos resultados.

## 6.4 Buffer de Alinhamento

Como visto no capítulo 5, antes de realizar a comutação dos dados de usuário, é necessário o estabelecimento do sincronismo entre os quadros E1 que

FIGURA 6.20 – Módulo *align.buffer*.

acessam o Comutador Digital, tomando como referência a fonte de relógio do multiplexador local (nó local). O procedimento de sincronização entre o quadro E1 transmitido e o quadro E1 recebido se denomina *Alinhamento de Quadro*. A *Recomendação G.706* descreve a estratégia a ser adotada para o estabelecimento de sincronismo a partir do reconhecimento da Palavra de Alinhamento de Quadro. O procedimento é o seguinte:

1. Monitorar o sinal de recepção e reconhecer a palavra de alinhamento de quadro *0011011*, nos *bits b1* a *b7* do quadro básico (canal 0).
2. No próximo quadro, o *bit b1* da palavra de serviço deve estar em nível lógico 1.
3. Reconhecer novamente, no quadro seguinte, a palavra de alinhamento de quadro.

Se as três condições anteriores forem positivas, então o sincronismo entre o transmissor e o receptor foi estabelecido.

Dentro da rede, o processo anterior é realizado pelo módulo de Alinhamento de Quadro localizado no Multiplexador Local [KAI 95a]. Portanto, o multiplexador local recebe o quadro E1 proveniente do multiplexador remoto, extrai o *jitter* acumulado nele, e sincroniza este quadro com o relógio local. Neste ponto, o quadro E1 recebido e o quadro E1 gerado localmente são síncronos entre si, e ambos são enviados ao Comutador Digital. O sincronismo obtido é a nível de bit, mas não a

nível de quadro, isto é, os quadros E1 que chegam ao Comutador precisam ainda ser alinhados. Para realizar o alinhamento, o multiplexador local fornece ao Comutador Digital de três sinais: o sinal de *reset* do Gerador de *Time-Slots* (TSG-RST), e as janelas de tempo correspondentes ao *time-slot* zero ou canal de serviço do quadro E1 local (TS0-1) e do quadro E1 remoto (TS0-2).

O alinhamento de quadro no Comutador Digital é realizado por um circuito de armazenamento elástico contido dentro do módulo *align.buffer* mostrado na figura 6.20. A entrada *qin* recebe o quadro E1 remoto desalinhado. As entradas *r0* e *r1* correspondem aos sinais de relógio usados para temporizar a escrita e a leitura do *buffer* elástico. Tanto *r0* como *r1* são sinais de relógio de 2048KHz, estando a fase de *r1* atrasada 90 graus em relação à fase de *r0*. As entradas *TSG-RST*, *TS0-1* e *TS0-2* correspondem aos sinais fornecidos pelo multiplexador local descritos previamente. A saída *qd2* entrega o quadro E1 recebido, alinhado com o quadro local, e a saída *rst.out*, é o sinal de *reset* que inicializa o processo de comutação.

A figura 6.21 mostra os sub-módulos que constituem o *Buffer de Alinhamento*. Os módulos *add.gen.strt*, são circuitos de disparo que habilitam os contadores de 8 bits, módulos *cont-256*, os quais fornecem o endereço para a escrita e a leitura do *buffer* elástico contido no módulo *256.celda*. O circuito de Alinhamento de Quadro funciona assim:

1. O módulo de Alinhamento de Quadro localizado no multiplexador local procura a palavra de alinhamento de quadro *0011011* no *time-slot* zero do quadro E1 recebido, e uma vez identificada, realiza a sincronização a nível de *bit* entre o quadro E1 recebido e o quadro E1 local. Concluída esta tarefa habilita o sinal *TSG-RST* (sinal de *reset* do módulo Gerador de *Time-Slots* do multiplexador), nível lógico 1, e disponibiliza os sinais *TS0-2* e *TS0-1* ao Comutador Digital.
2. Já com *TSG-RST* habilitado, o *buffer* de alinhamento espera pela primeira ocorrência do sinal *TS0-2*, provido pelo módulo gerador de *time-slots* do multiplexador, e habilita o módulo *add.gen.strt* (inferior), corres-

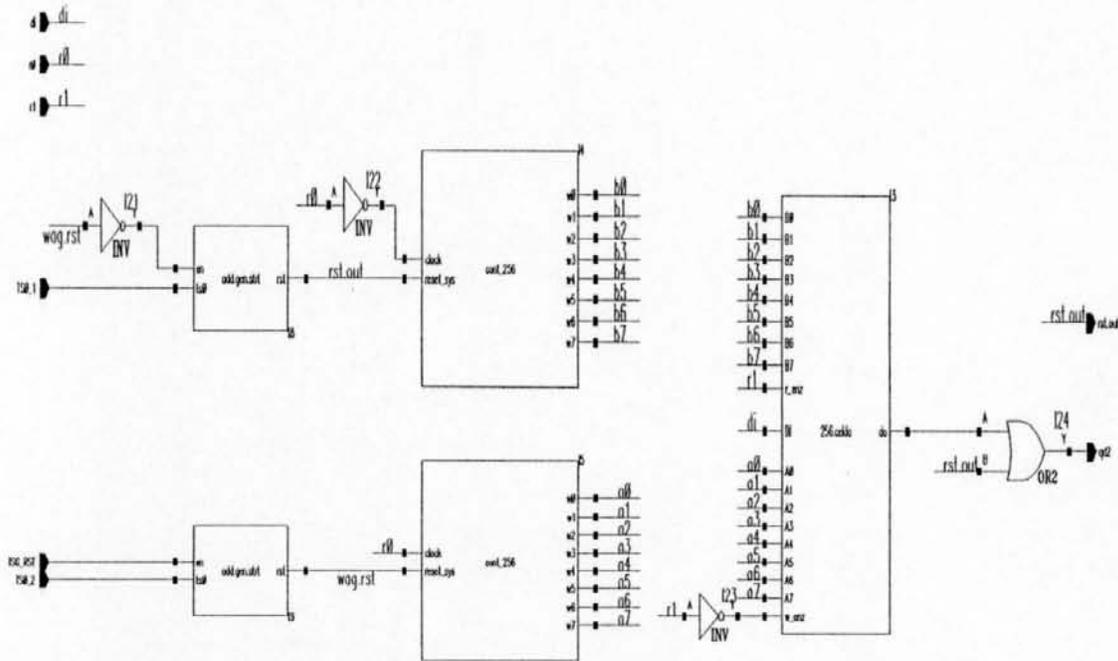


FIGURA 6.21 – Buffer de Alinhamento.

pondente ao contador de endereços de escrita, e inicia o armazenamento de dados no *buffer* elástico (módulo *256.cel*da).

3. Depois do início da escrita, o módulo *add.gen.strt* (superior), correspondente ao contador de endereços de leitura, é habilitado pelo sinal *wag.rst* e espera pela ocorrência do sinal *TS0-1*. A partir daí, inicia-se a leitura dos dados do *buffer* elástico, operação que disponibiliza o quadro alinhado *qd2*, e inicializa ao mesmo tempo o processo de comutação mediante o sinal *rst.out*.

A figura 6.22 apresenta os resultados de simulação de um caso típico de alinhamento de quadro. O sinal */q1* é o quadro E1 gerado localmente, e */q2* é o quadro E1 recebido do multiplexador remoto. Observa-se que, com o sinal */TSG-RST* habilitado, a ocorrência de */TS0-2* e */TS0-1*, nessa ordem, habilita ao módulo *align.buffer* para disponibilizar o sinal */qd2*, que corresponde ao quadro */q2* alinhado com o quadro */q1*. O primeiro *time-slot* de */q2* (01100011), e o primeiro *time-slot* de */q1* aparecem alinhados aos 145 $\mu$ s.

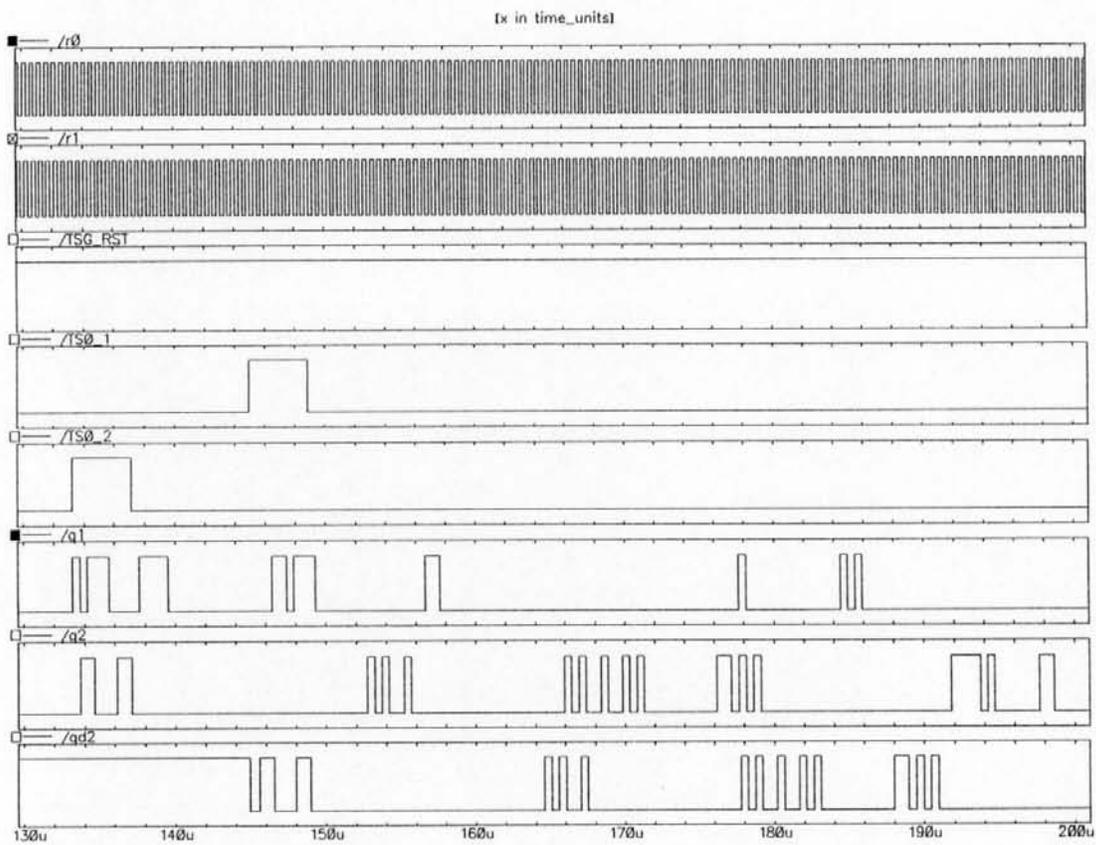


FIGURA 6.22 - Simulação do Caso Típico.

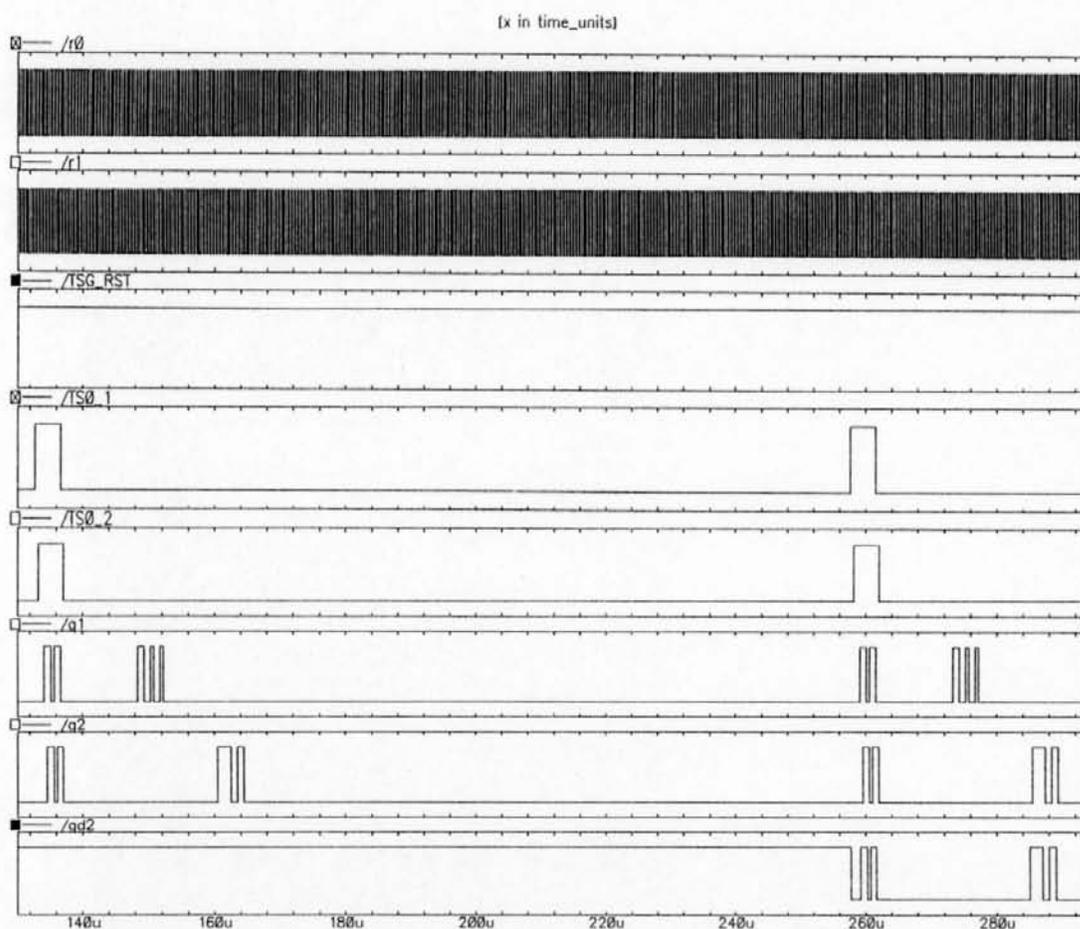


FIGURA 6.23 – Simulação do Pior Caso.

Por outro lado, a figura 6.23 mostra os resultados da simulação do pior caso do alinhamento de quadro. Neste caso, o sinal  $/TS0-1$  ocorre  $1\text{ bit}$  antes do sinal  $/TS0-2$ , e obriga ao *buffer* de alinhamento a esperar pela próxima ocorrência de  $/TS0-1$ , que acontece quase  $125\mu\text{s}$  depois ( $255\text{ bits}$ ), e como consequência a usar praticamente toda sua capacidade de armazenamento (o *buffer* possui  $256\text{ bits}$ ). O *buffer* elástico começa a ser escrito aos  $133\mu\text{s}$  e, aproximadamente  $125\mu\text{s}$  depois (1 quadro E1), começa a ser lido a partir dos  $258\mu\text{s}$ . Neste instante, os quadros  $/q1$  e  $/q2$  aparecem alinhados.

Uma vez concluído o processo de alinhamento dos quadros, da-se início às operações de comutação propriamente ditas.

## 6.5 Especificação do Controle

A implementação do módulo de Controle do Comutador Digital forma parte das atividades a serem desenvolvidas em trabalhos futuros. Entretanto, considerando que este módulo, junto ao Comutador Digital, é imprescindível para constituir um Sistema de Comutação, suas características mais importantes serão especificadas aqui.

Basicamente, o módulo de controle deve gerenciar as tarefas de programação da memória de conexões CM, bem como as de atendimento a um canal serial externo de comunicações, mediante o qual o operador da rede envia ao Comutador a nova configuração dos enlaces entre usuários. Além disso, deve permitir o acesso ao canal interno de 20 Kbit/s disponível no *time-slot* zero dos quadros gerados pelo multiplexador. Este canal constitui o meio de comunicação através do qual dar-se-á o tráfego das mensagens do futuro Sistema de Gerenciamento e Controle da Rede, a ser implementado com base no protocolo *LAP-D*.

A utilização de um microcontrolador para a implementação do módulo de Controle é uma opção de projeto atrativa, pois permite a personalização das funções de controle mediante *software*, bem como armazenar informação sobre os usuários, e manipular protocolos de comunicação síncrona ou assíncrona. Outras vantagens relativas ao uso de um microcontrolador são a modularidade e a facilidade de extensão que oferece ao projeto. Na figura 6.24 é mostrado o diagrama de estados correspondente ao Controle do Comutador Digital, a ser implementado usando um microcontrolador. Observa-se que, o controle é determinado por quatro (4) estados básicos, e é sensível a duas (2) rotinas de interrupção.

O estado *Start*, agrupa os passos necessários para a inicialização do sistema: habilitação de interrupções, estabelecimento de prioridades entre elas, configuração do canal serial externo de comunicações, definição do *buffer* para a recepção de dados. Dependendo do nó ao qual esteja conectado, o sistema trabalhará em modo *Mestre*, ou em modo *Escravo*. No caso de operar neste último, o sistema

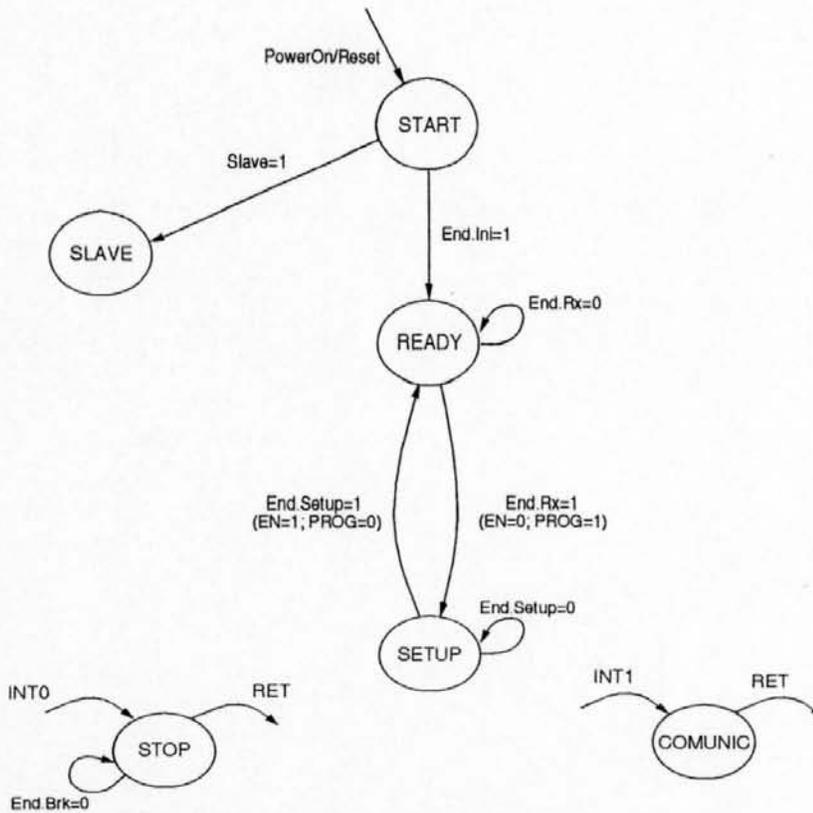


FIGURA 6.24 – Diagrama de Estados.

entrará no estado *Slave*, onde toda a informação relacionada com a comutação e a habilitação de laços de teste, é recebida via o canal interno de 20 Kbit/s do multiplexador. No modo Mestre, o sistema passa pelos estados *Ready* e *Setup*, e fica sensível às rotinas de interrupção *Stop* e *Comunic*. O estado *Ready*, consiste basicamente de um *loop* de espera, durante o qual o sistema fica aguardando pela chegada de um novo dado de configuração do Comutador, enquanto supervisa o funcionamento do comutador e do multiplexador. A chegada de um dado pelo canal serial externo, habilita a rotina de comunicação chamada *Comunic*. Uma vez concluída a recepção dos dados ( $end.rx=1$ ), o sistema passa para o estado *Setup*, durante o qual a memória de conexões CM é programada segundo a informação de configuração recebida. Finalizado este processo, o sistema volta ao estado *Ready*. Caso for necessário, a rotina de interrupção *Stop* permitirá em qualquer instante, levar o sistema a um estado de repouso no qual não realiza tarefa alguma. O sistema abandona este estado via um sinal externo ( $end.brk=1$ ), e retorna ao ponto no qual foi interrompido.

Um dispositivo bastante conhecido e amplamente utilizado em aplicações de controle e sistemas de comunicação de dados, é o microcontrolador de 8 *bits* da família Intel MCS-51 [STE 93]. Esta família, tipificada pelo 8051, é orientada principalmente para aplicações de controle seqüenciais. O 8051 contém 4 *Kbytes* de ROM, 128 *bytes* de RAM, 32 linhas E/S, 2 contadores ou temporizadores de 16 *bits* cada, 5 fontes de interrupção (sendo 2 externas), uma porta serial *duplex*, e um processador booleano a nível de *bit* [INT 86]. As características deste microcontrolador enquadram-se dentro das necessidades do sistema, sendo esta a razão principal de sua escolha. Porém, isto não significa que não possa ser utilizado um outro microcontrolador no seu lugar.

Embora o 8051, possua uma interface serial para transmissão e recepção de dados, para obter o mesmo nível de desempenho que um processador como o 8085, o uso de uma USART (*Universal Synchronous Asynchronous Receiver Transmitter*) dedicada é recomendado [STE 93]. Contudo, a necessidade de usar um protocolo síncrono de comunicações como o BSC, exige a presença de um *hardware* adicional para as tarefas de inclusão e reconhecimento de caracteres de sincronismo, e para a aplicação do algoritmo CRC de controle de erros.

Dentro do grupo de componentes orientados a comunicações entre microcomputadores oferecidos pela Intel [INT 87], foi encontrado um controlador serial de comunicações que suporta o protocolo BSC. O Controlador Serial Multi-Protocolo MPSC 8274 da Intel, é capaz de manipular protocolos de comunicação síncrona e assíncrona. Sua característica programável lhe permite ser configurado em vários modos de operação, proporcionando otimização de uma determinada aplicação de comunicação de dados. Mas, a característica mais importante deste controlador de comunicações, é sua completa compatibilidade com os microcontroladores da família 8051.

O MPSC 8274 implementa o protocolo BSC com um ou dois caracteres de sincronismo SYNC, e realiza a geração e verificação automática de CRC. Também suporta os protocolos síncronos SDLC (*Synchronous Data Link Control*) e HDLC

(*High Level Data Link Control*) compatíveis com a Recomendação X.25 da CCITT. Este dispositivo possui uma interface serial com 2 canais independentes A e B. Cada canal consiste de um bloco de transmissão, um bloco de recepção, e um banco de registradores de leitura e escrita utilizados para inicialização e supervisão. Além disso, um bloco de lógica de controle proporciona os pinos de interface para conexão com modem [INT 87].

Como visto no capítulo anterior, pretende-se estabelecer um enlace Terminal de Usuário e Controle do Comutador (ponto-a-ponto), baseado no protocolo BSC, operando em seu modo transparente. A transparência é necessária para poder transmitir os dados binários correspondentes à programação da memória de conexões, e os códigos de operação. Os caracteres de controle de erro CRC-16, são usados comumente no modo transparente. O MPSC 8274 deve ser programado com estas características.

Finalmente, considerando-se que a medida em que os dados chegam pelo canal serial, estes devem ser armazenados até o instante em que serão processados, fazendo-se assim necessário o uso de uma RAM externa. O circuito integrado 8155 é um dispositivo periférico tipicamente utilizado na interface do microcontrolador 8051. Consiste de uma RAM estática de 256x8, e de 3 portos de E/S programáveis. Sendo utilizado, o 8155 oferece além de uma RAM externa, mais três portas de E/S adicionais.

A figura 6.25 descreve o diagrama de blocos do Sistema de Comutação (Controle e Comutador Digital), e as interfaces com o multiplexador E1. Os diagramas de fluxo das rotinas que implementam os estados do Controle do Comutador, são mostrados no Anexo A-1. As rotinas estão baseadas no *software* do microcontrolador MCS 8051 e do controlador serial de protocolos MPSC 8274.

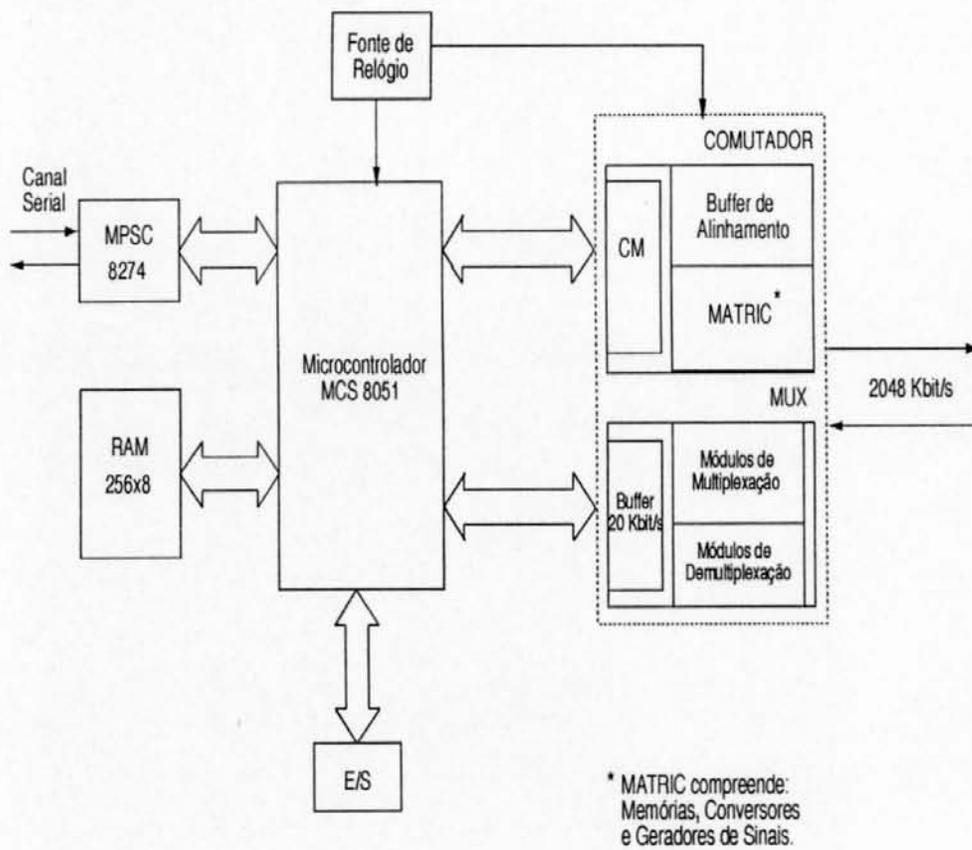


FIGURA 6.25 – Diagrama de Blocos do Sistema de Comutação.

## 7 SIMULAÇÃO DO COMUTADOR DIGITAL

Os resultados de simulação da arquitetura completa do Comutador Digital são apresentados neste capítulo. Diversas simulações foram realizadas usando o simulador lógico *SILOS*, considerando as três possibilidades de comutação (enlaces entre usuários locais em ambos os nós da rede, local e remoto, e enlaces entre usuários pertencentes a diferentes nós), e o caso crítico (pior caso) no alinhamento dos quadros. Os resultados obtidos foram positivos e permitiram validar a arquitetura proposta.

Para uma melhor explicação e compreensão dos resultados a serem apresentados, optou-se pelo uso de uma única memória de conexões (memória *romx*) contendo informação sobre as diversas conexões a serem estabelecidas entre o total de 64 usuários da rede. Assim, dividiu-se o total de usuários em três grupos: o primeiro agrupa aqueles usuários que solicitaram um enlace local dentro do nó local (comutação local-local); no segundo grupo estão aqueles usuários que solicitaram um enlace local dentro do nó remoto (comutação local-remota); e finalmente, no terceiro grupo estão aqueles usuários que solicitaram um enlace com usuários de um nó diferente (comutação remota). Para conseguir descrever os resultados de cada caso por separado, utilizou-se vetores de teste de entrada diferentes, modificando o conteúdo dos quadros de entrada E1 mediante a inclusão de dados apenas nos *time-slots* dos usuários envolvidos em cada caso de comutação. Desta maneira, pode-se avaliar melhor e com mais facilidade, o desempenho do Comutador Digital.

Considerando que, os resultados de simulação do processo de Alinhamento de Quadro já foram apresentados no capítulo anterior, aqui o maior ênfase será dado ao processo de Comutação. Porém, não deve perder-se de vista que a Comutação começa a partir do instante seguinte ao final do processo de alinhamento, e portanto, ambos processos caracterizam ao Comutador Digital. A figura 7.1 mostra o diagrama esquemático total do Comutador Digital usado na simulação. O módulo

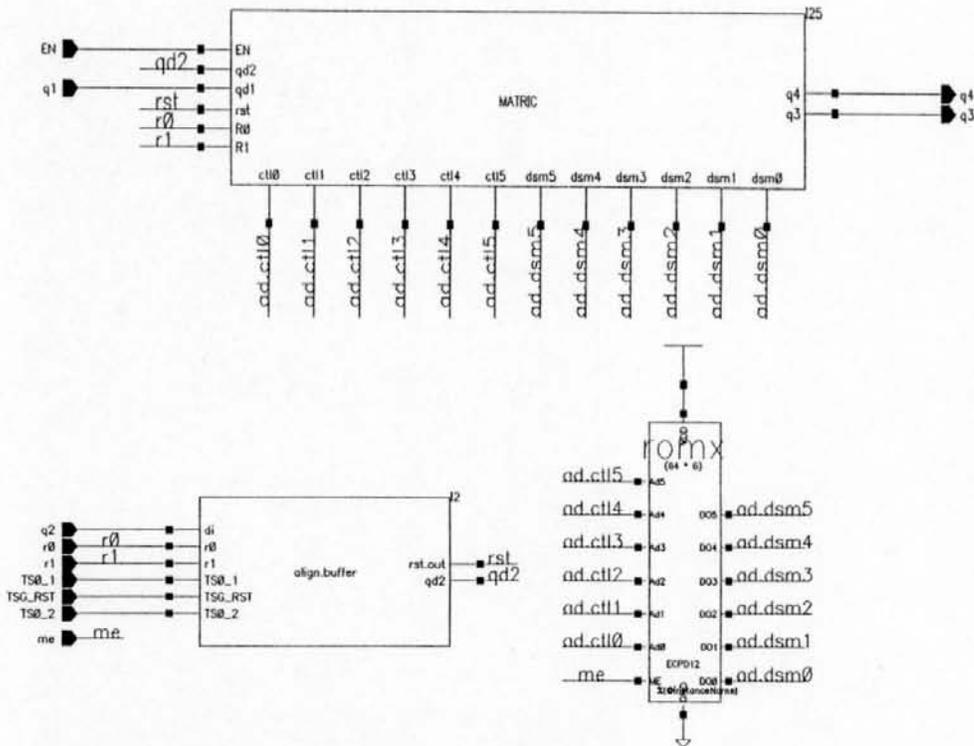


FIGURA 7.1 – Diagrama Esquemático do Computador Digital.

*MATRIC* consiste do Conversor Série a Paralelo, o Conversor Paralelo a Série, as memórias SFM e DSM, e o Gerador de Endereços MAC. O módulo *align.buffer* e a memória *romx*, correspondem ao circuito do *Buffer* de Alinhamento e a Memória de Conexões respectivamente.

## 7.1 Comutação Local-Local

Neste caso, um grupo de usuários conectados ao multiplexador local (nó local), solicitam ao Computador Digital o estabelecimento de um enlace de comunicação entre eles. Isto significa que, o Computador deve deslocar os dados destes usuários, contidos nos *time-slots* do quadro de entrada Q1, de maneira tal que apareçam adequadamente localizados nos *time-slots* do quadro de saída Q4, que é o que retorna ao multiplexador local.

A memória de conexões *romx* foi programada para estabelecer os seguintes enlaces entre usuários locais: usuário 6 com usuário 22; usuário 11 com usuário 31; usuário 12 com usuário 15; usuário 14 com usuário 16; e usuário 23 com usuário 24.

Para facilitar a identificação dos *time-slots* correspondentes a cada usuário local dentro dos quadros E1, cada *time-slot* carrega como dado o número correspondente a sua posição dentro do quadro. Assim, por exemplo, o *time-slot* 6, correspondente ao usuário local 6, transporta o número binário 01100000 (o primeiro *bit* é o LSB); o *time-slot* 22, transporta o número binário 01101000. A tabela 7.1, resume o conteúdo dos quadros de entrada Q1 e Q2, segundo a informação contida em cada *time-slot*. O quadro Q2 carrega informação apenas no seu canal de serviço (*time-slot* zero). Isto fez-se com o propósito de facilitar a observação da comutação local-local. A presença de dados no *time-slot* zero de Q2 é importante porque permite observar como é tratada a informação de serviço, relevante apenas para o multiplexador local. Esta informação deve formar parte do *time-slot* zero do quadro Q4.

A figura 7.2 mostra uma vista geral no tempo do processo de comutação. Este processo começa logo após ser obtido o alinhamento dos quadros de entrada Q1 e Q2, aos  $145\mu\text{s}$ , passa por um período de latência que estende-se até os  $395\mu\text{s}$ , instante a partir do qual os quadros Q3 e Q4 ficam disponíveis na saída do Comutador Digital. O sinal */r0* é o sinal de relógio local de frequência igual a 2048KHz. O sinal */TS0-1* corresponde à janela de tempo que marca o início e o fim do *time-slot* zero do quadro de entrada Q1, que aparece na figura como o sinal */q1*. Alinhado com este último, o sinal */qd2* corresponde ao quadro de entrada Q2, logo após de atravessar pelo *Buffer de Alinhamento*. Os sinais */q3* e */q4* representam os quadros de saída Q3 e Q4 do Comutador Digital. A maneira de auxílio à visualização dos quadros e dos *time-slots*, foram incluídos na figura o sinal de temporização de quadro, sinal */I25/save*, e o sinal de temporização de octeto, */I25/I10/I308.push*.

O conteúdo de cada quadro de entrada ao Comutador Digital pode ser verificado na figura 7.3. Tomando como referência os sinais */TS0-1*, */I25/save* e */I25/I10/I308.push*, observa-se quais os dados entrantes ao Comutador, depois de

FIGURA 7.2 - Simulação da Comutação Local-Local (vista geral).

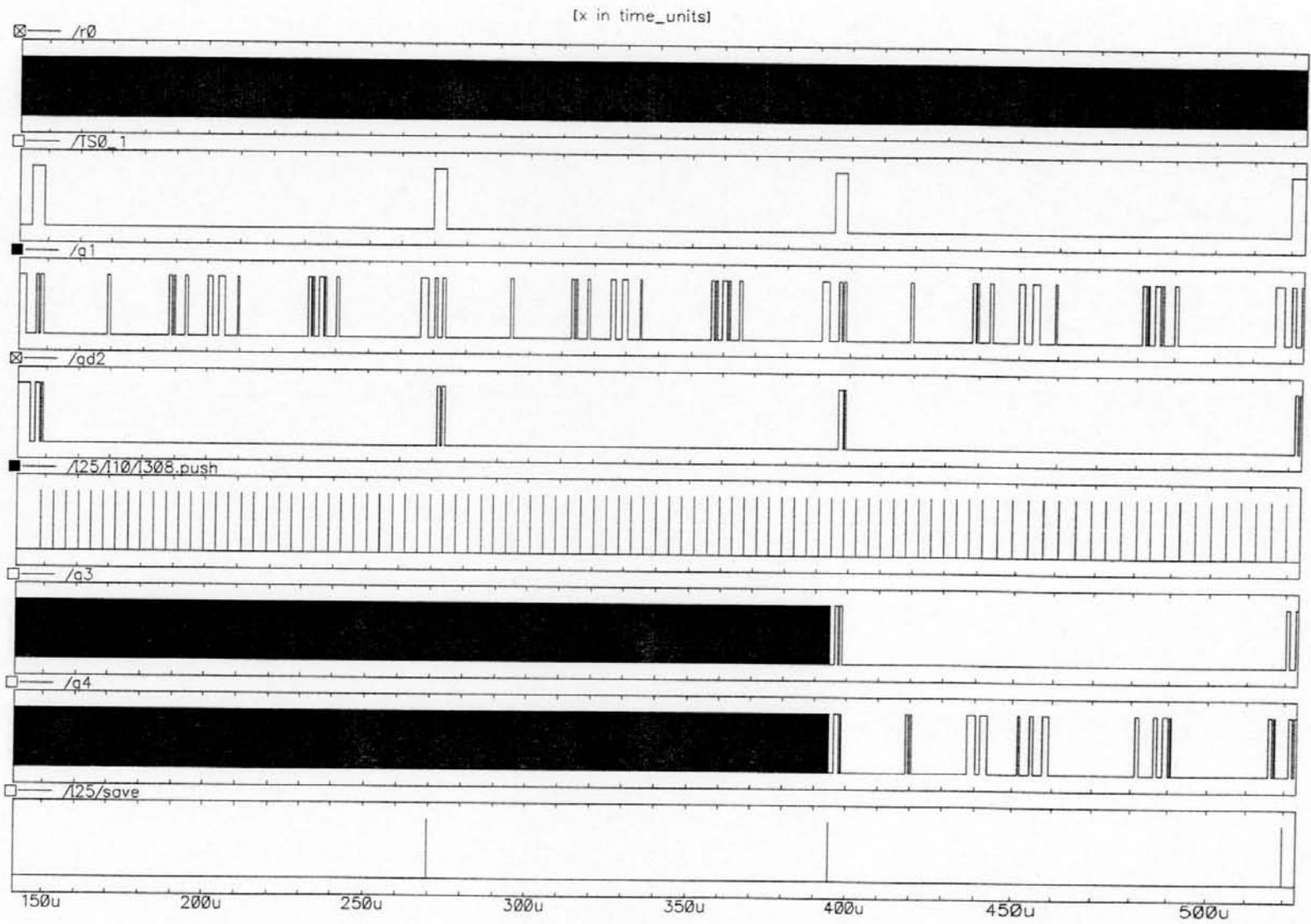


TABELA 7.1 – Quadros de Entrada Comutação Local-Local

Quadros de Entrada	
Q1	Q2
TS(00) = 00011011	TS(00) = 00011101
TS(06) = 01100000	-
TS(11) = 11010000	-
TS(12) = 00110000	-
TS(14) = 01110000	-
TS(15) = 11110000	-
TS(16) = 00001000	-
TS(22) = 01101000	-
TS(23) = 11101000	-
TS(24) = 00011000	-
TS(31) = 11111000	-

TABELA 7.2 – Quadros de Saída Comutação Local-Local

Quadros de Saída	
Q3	Q4
TS(00) = 00011011	TS(00) = 00011101
-	TS(06) = 01101000
-	TS(11) = 11111000
-	TS(12) = 11110000
-	TS(14) = 00001000
-	TS(15) = 00110000
-	TS(16) = 01110000
-	TS(22) = 01100000
-	TS(23) = 00011000
-	TS(24) = 11101000
-	TS(31) = 11010000

conseguido o alinhamento dos quadros, e o valor não determinado das saídas  $/q3$  e  $/q4$  durante intervalo de tempo mostrado, devido ao período de latência.

Na figura 7.4 podem observar-se os resultados da simulação. Aos  $395\mu s$  (use como referência o sinal  $/I25/save$ ),  $250\mu s$  depois do início da comutação, os sinais  $/q3$  e  $/q4$  apresentam o conteúdo resultante dos quadros de saída Q3 e Q4. Conforme o esperado, o quadro Q3 apenas contém informação no seu *time-slot* zero, que deve ser enviado até o multiplexador remoto. Já o quadro Q4 transporta os dados de usuário comutados segundo a programação da memória de conexões. A verificação das saídas pode ser feita com ajuda do sinal de temporização de octeto  $/I25/I10/I308.push$ . A tabela 7.2 resume o conteúdo de cada *time-slot* dos quadros de saída Q3 e Q4.

## 7.2 Comutação Local-Remota

De maneira similar ao caso anterior, um grupo de usuários conectados ao multiplexador remoto (nó remoto) solicitam ao Comutador Digital o estabelecimento de conexões entre eles. Neste caso, o Comutador deve deslocar os dados destes usuários, contidos nos *time-slots* do quadro de entrada Q2, até posições adequadas nos *time-slots* do quadro de saída Q3 (que vai até o multiplexador remoto), que permitam o intercâmbio solicitado.

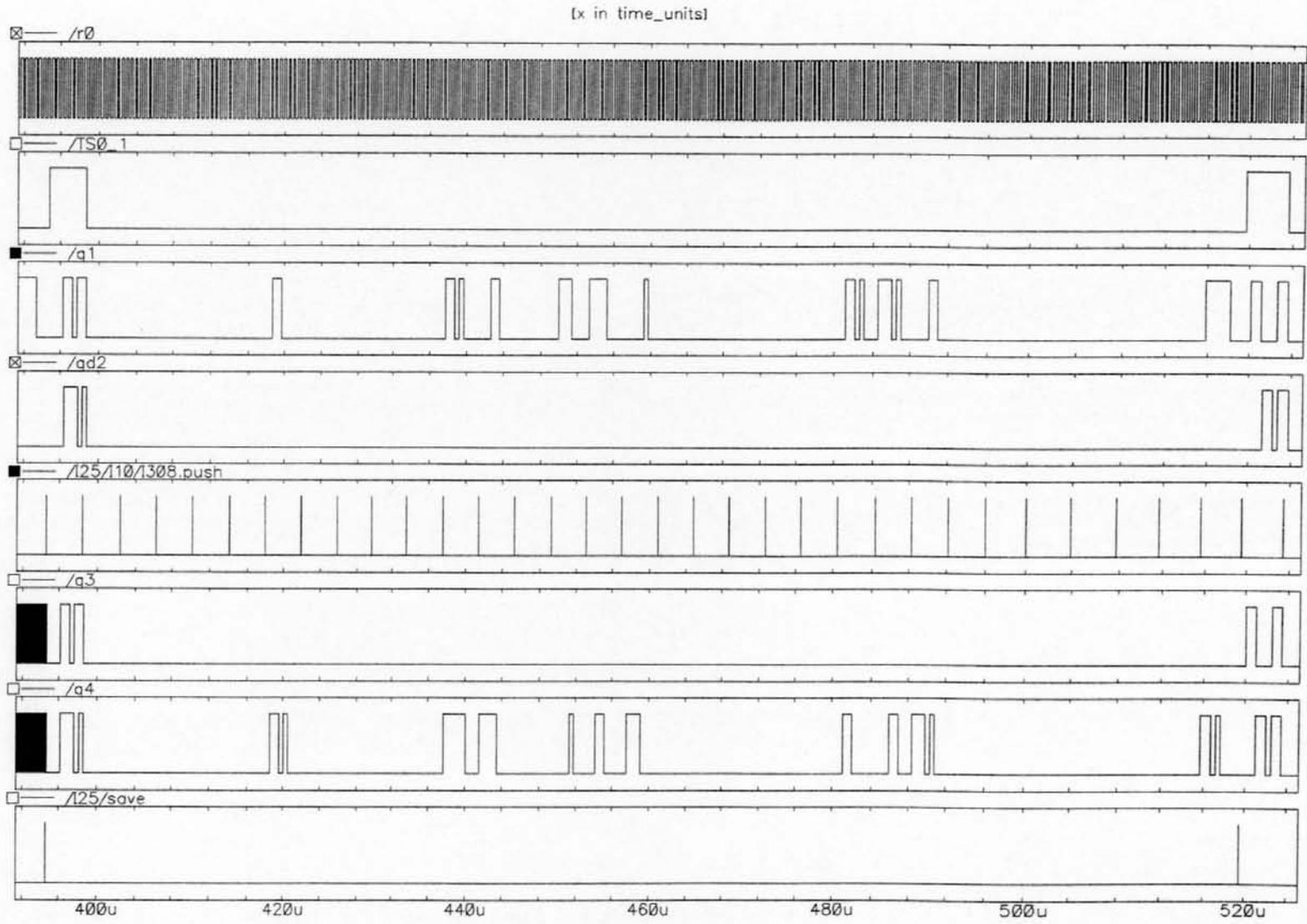
Os enlaces solicitados pelos usuários remotos foram os seguintes: usuário 3 com usuário 7; usuário 4 com usuário 30; usuário 10 com usuário 14; usuário 22 com usuário 24; e usuário 23 com usuário 28.

A informação contida nos *time-slots* de cada usuário remoto dentro dos quadros E1, corresponde a sua posição dentro do quadro somada ao valor 32. Assim, por exemplo, o *time-slot* 3, correspondente ao usuário remoto 3, transporta o número binário 11000100 (o primeiro *bit* é o LSB) que equivale ao número decimal 35. Isto permite uma fácil identificação dos *time-slots*, e associa seu valor à sua procedência. A tabela 7.3, resume a informação transportada pelos quadros de entrada Q1 e



FIGURA 7.3 - Entradas ao Computador Digital, Comutação Local-Local.

FIGURA 7.4 – Saídas do Computador Digital, Comutação Local-Local.



Q2 em seus respectivos *time-slots*. O quadro Q1 carrega informação apenas no seu canal de serviço (*time-slot zero*). Esta informação deve aparecer no *time-slot zero* do quadro Q3, e deve ser enviada até o multiplexador remoto.

A figura 7.5 mostra uma vista geral no tempo do processo de comutação, que iniciou-se aos  $145\mu\text{s}$ , imediatamente após de ser obtido o alinhamento dos quadros de entrada Q1 e Q2. Os quadros de saída Q3 e Q4 ficam disponíveis na saída do Comutador Digital aos  $395\mu\text{s}$  depois de passar por um período de latência de  $250\mu\text{s}$ . O sinal  $/r0$  é o sinal de relógio local de frequência igual a 2048KHz. O sinal  $/TS0-1$  corresponde à janela de tempo que marca o início e o fim do *time-slot zero* do quadro de entrada Q1, representado na figura como o sinal  $/q1$ . O sinal  $/qd2$  corresponde ao quadro de entrada Q2 alinhado com o quadro Q1, e é a saída do *Buffer de Alinhamento*. Os sinais  $/q3$  e  $/q4$  representam os quadros de saída Q3 e Q4 do Comutador Digital. A maneira de auxílio à visualização dos quadros e dos *time-slots*, foram incluídos na figura o sinal de temporização de quadro, sinal  $/I25/save$ , e o sinal de temporização de octeto,  $/I25/I10/I308.push$ .

Na figura 7.6, pode ser verificado o conteúdo de cada quadro de entrada ao Comutador Digital. Utilizando como referência os sinais  $/TS0-1$ ,  $/I25/save$  e  $/I25/I10/I308.push$ , observa-se quais os dados entrantes ao Comutador, depois de realizado o alinhamento dos quadros, e o valor não determinado das saídas  $/q3$  e  $/q4$  durante o intervalo de tempo mostrado, devido ao período de latência.

Os resultados da simulação podem observar-se na figura 7.7. Os sinais de saída  $/q3$  e  $/q4$  disponibilizam o conteúdo resultante dos quadros de Q3 e Q4, aos  $395\mu\text{s}$  (use como referência o sinal  $/I25/save$ ), isto é,  $250\mu\text{s}$  depois do início da comutação. Conforme o esperado, o quadro Q4 apenas contém informação no seu *time-slot zero*, que deve ser enviado até o multiplexador local. Já o quadro Q3 transporta os dados de usuário comutados segundo a programação da memória de conexões. A verificação das saídas pode ser feita com ajuda do sinal de temporização de octeto  $/I25/I10/I308.push$ . A tabela 7.4 resume o conteúdo de cada *time-slot* dos quadros de saída Q3 e Q4.

FIGURA 7.5 - Simulação da Comutação Local-Remota (vista geral).

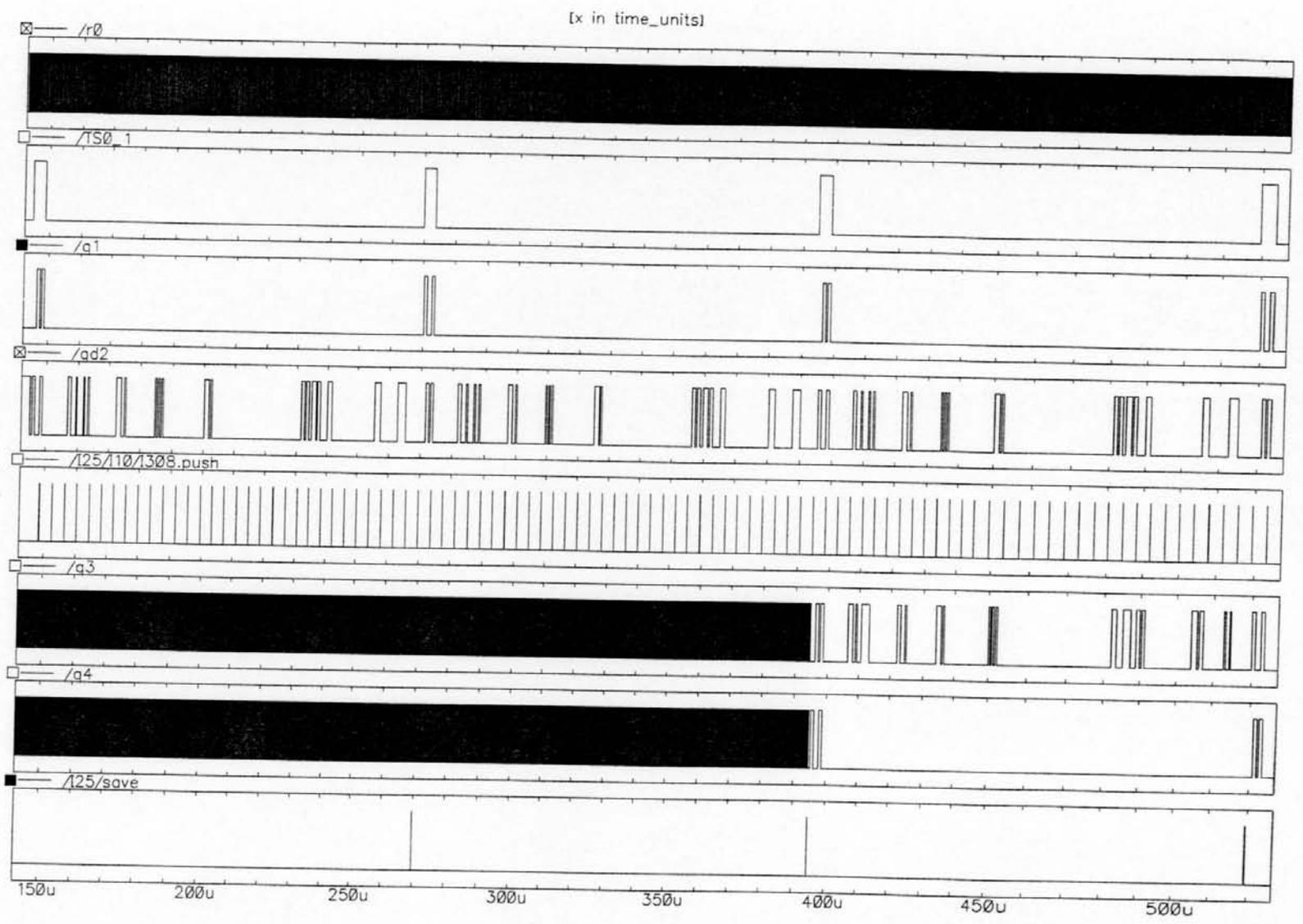


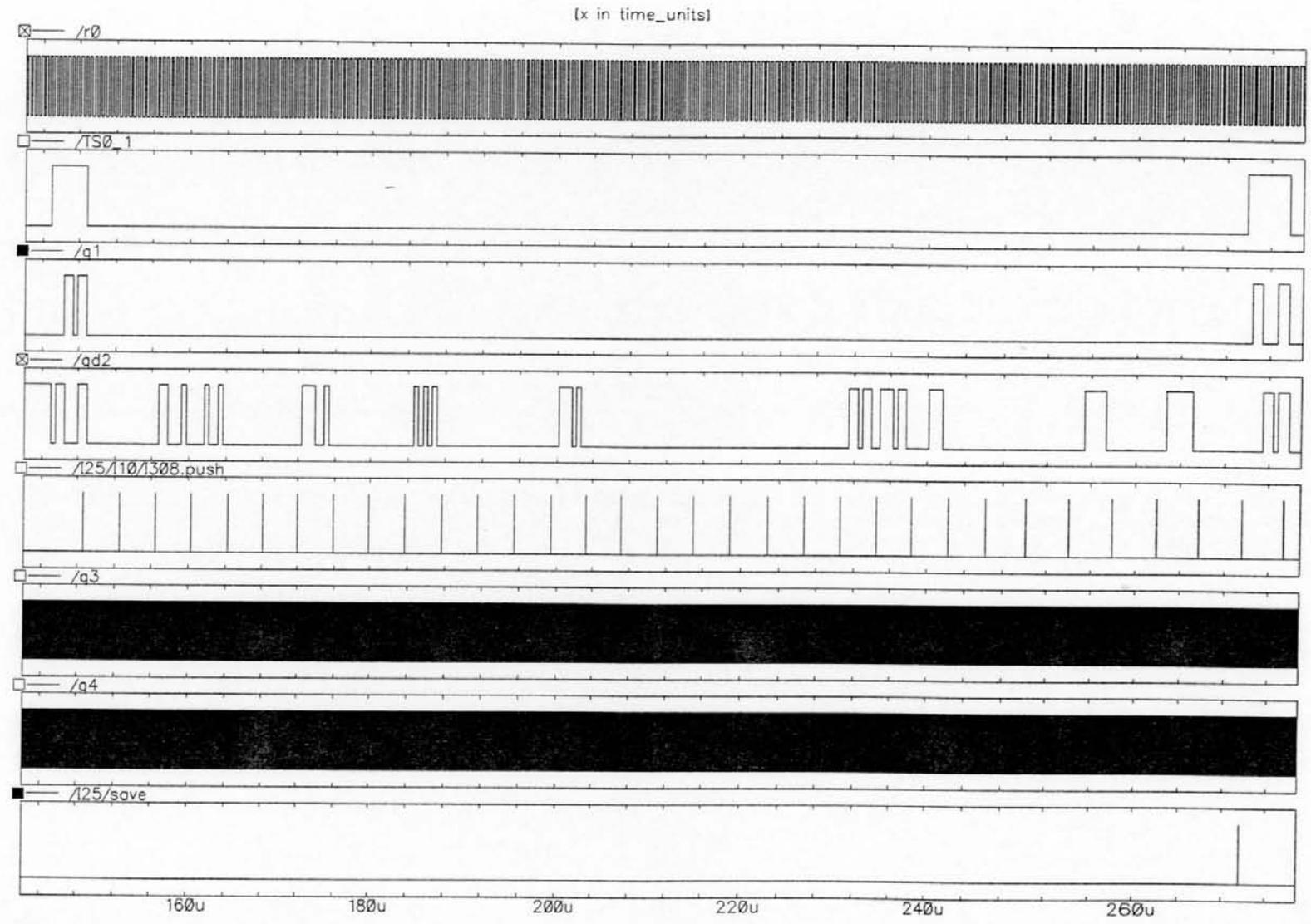
TABELA 7.3 – Quadros de Entrada Comutação Local-Remota

Quadros de Entrada	
Q1	Q2
TS(00) = 00011011	TS(00) = 01100011
-	TS(03) = 11000100
-	TS(04) = 00100100
-	TS(07) = 11100100
-	TS(10) = 01010100
-	TS(14) = 01110100
-	TS(22) = 01101100
-	TS(23) = 11101100
-	TS(24) = 00011100
-	TS(28) = 00111100
-	TS(30) = 01111100

TABELA 7.4 – Quadros de Saída Comutação Local-Remota

Quadros de Saída	
Q3	Q4
TS(00) = 00011011	TS(00) = 01100011
TS(03) = 11100100	-
TS(04) = 01111100	-
TS(07) = 11000100	-
TS(10) = 01110100	-
TS(14) = 01010100	-
TS(22) = 00011100	-
TS(23) = 00111100	-
TS(24) = 01101100	-
TS(28) = 11101100	-
TS(30) = 00100100	-

FIGURA 7.6 – Entradas ao Computador Digital, Comutação Local-Remota.



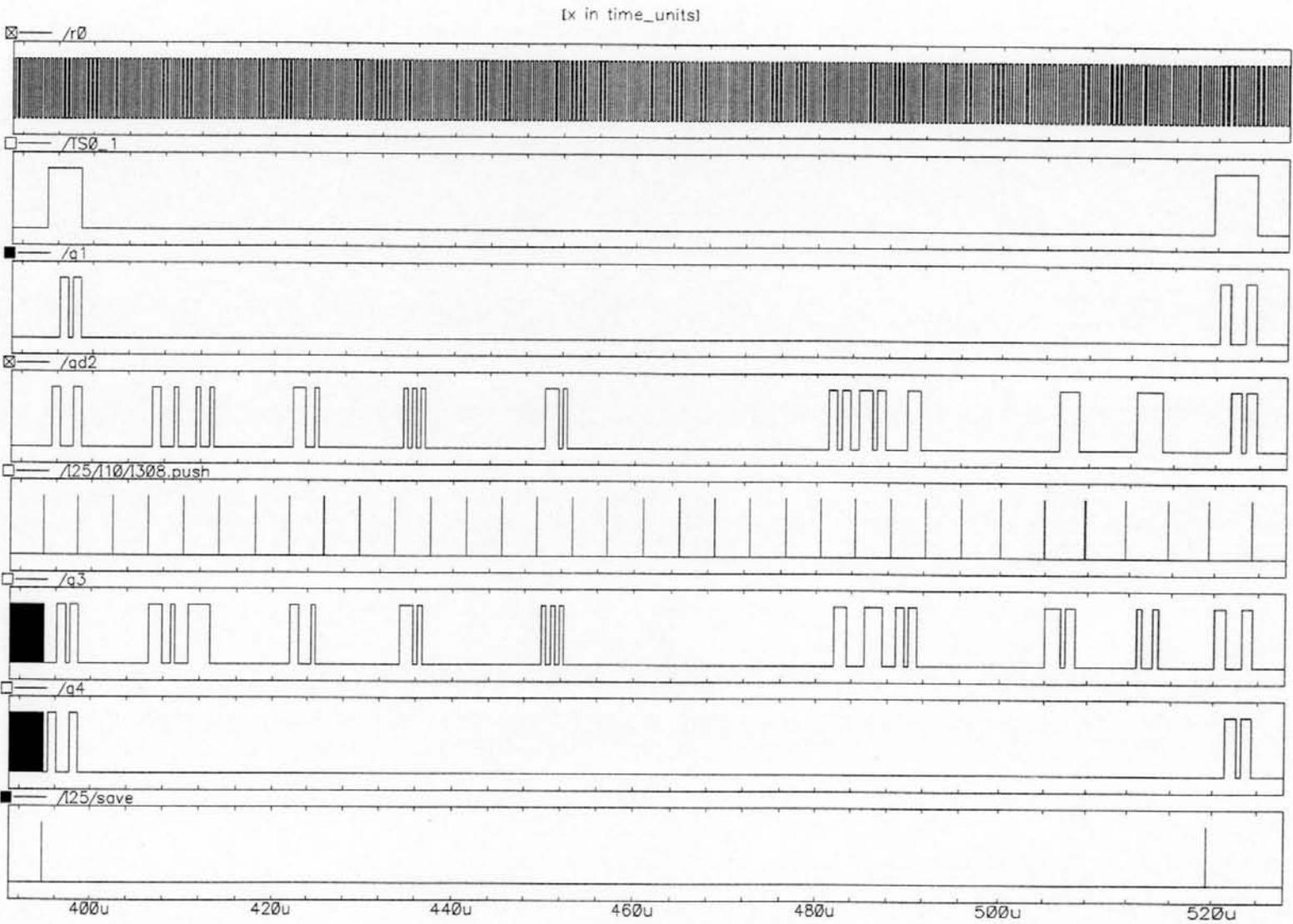


FIGURA 7.7 – Saídas do Computador Digital, Comutação Local-Remota.

### 7.3 Comutação Remota

Este é o caso mais interessante de análise, pois permite visualizar o verdadeiro potencial do Comutador Digital, bem como destacar sua importância dentro do ambiente da rede. Neste caso, um grupo de usuários conectados ao multiplexador local (nó local) necessitam conectar-se com outro grupo de usuários conectados ao multiplexador remoto (nó remoto). Para isto, o Comutador deve deslocar a informação contida nos *time-slots* pertencentes aos quadros de entrada Q1 e Q2, até posições adequadas nos *time-slots* dos quadros de saída Q3 e Q4, de maneira que permitam o intercâmbio solicitado e um fluxo de dados contínuo entre ambos os extremos da rede.

Os enlaces a serem estabelecidos são os seguintes: usuário local 1 com usuário remoto 16; usuário local 3 com usuário remoto 11; usuário local 8 com usuário remoto 8; usuário local 10 com usuário remoto 21; usuário local 17 com usuário remoto 18; usuário local 20 com usuário remoto 5; usuário local 25 com usuário remoto 20; usuário local 27 com usuário remoto 15; usuário local 29 com usuário remoto 9; e usuário local 30 com usuário remoto 29.

Dentro do quadro de entrada Q1, os *time-slots* de cada usuário local, carregam como dado o número correspondente a sua posição dentro do quadro. Assim, por exemplo, o *time-slot* 8, correspondente ao usuário local 8, transporta o número binário 00010000 (o primeiro *bit* é o LSB). No quadro de entrada Q2, os *time-slots* pertencentes aos usuários remotos, carregam como dado o número correspondente a sua posição dentro do quadro somado ao valor 32. Por exemplo, o *time-slot* 8, correspondente ao usuário remoto 8, transporta o número binário 00010100 (o primeiro *bit* é o LSB), que equivale ao número decimal 40. Desta maneira, o valor transportado por cada *time-slot* está associado à sua procedência (local ou remoto). Os canais de serviço de ambos os quadros de entrada possuem informação que deve ser encaminhada aos multiplexadores, portanto a nova posição destes *time-slots* é sempre conhecida. Na tabela 7.5, resume a informação transportada pelos quadros de entrada Q1 e Q2 em seus respectivos *time-slots*.

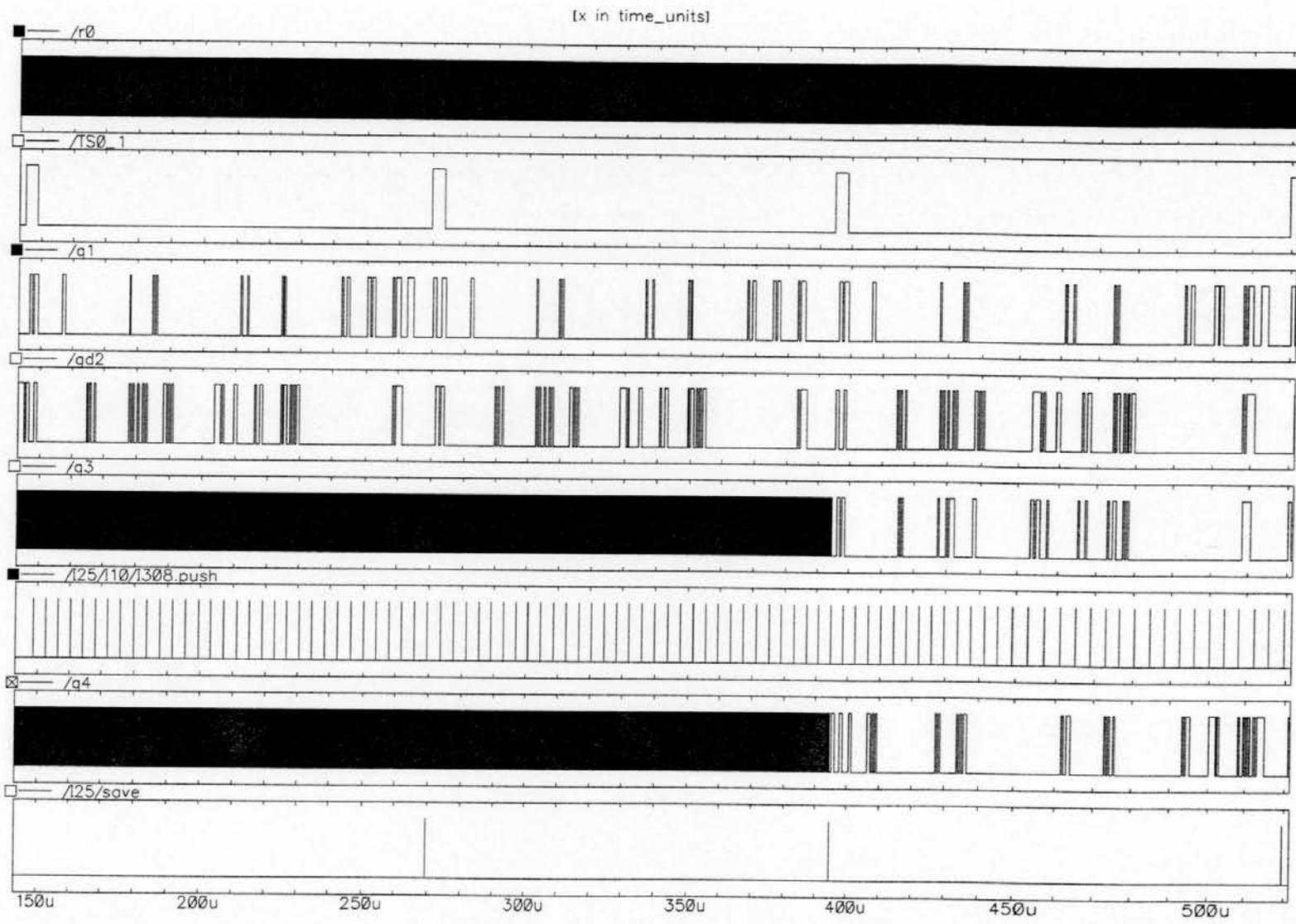


FIGURA 7.8 - Simulação da Computação Remota (vista geral).

TABELA 7.5 – Quadros de Entrada Comutação Remota

Quadros de Entrada	
Q1	Q2
TS(00) = 00011011	TS(00) = 01100011
TS(01) = 10000000	TS(05) = 10100100
TS(03) = 11000000	TS(08) = 00010100
TS(08) = 00010000	TS(09) = 10010100
TS(10) = 01010000	TS(11) = 11010100
TS(17) = 10001000	TS(15) = 11110100
TS(20) = 00101000	TS(16) = 00001100
TS(25) = 10011000	TS(18) = 01001100
TS(27) = 11011000	TS(20) = 00101100
TS(29) = 10111000	TS(21) = 10101100
TS(30) = 01111000	TS(29) = 10111100

A figura 7.8 mostra uma vista geral no tempo do processo de comutação, iniciado aos  $145\mu s$ , imediatamente após de ser obtido o alinhamento dos quadros de entrada Q1 e Q2. Os quadros de saída Q3 e Q4 ficam disponíveis na saída do Comutador Digital aos  $395\mu s$ , devido à latência do Comutador Digital. O sinal  $/r0$  é o sinal de relógio local de frequência igual a 2048KHz. O sinal  $/TS0-1$  corresponde à janela de tempo que determina o início e o fim do *time-slot* zero do quadro de entrada Q1, representado na figura como o sinal  $/q1$ . O sinal  $/qd2$  corresponde ao quadro de entrada Q2 alinhado com o quadro Q1, e é a saída do *Buffer de Alinhamento*. Os quadros de saída Q3 e Q4 do Comutador Digital, aparecem como os sinais  $/q3$  e  $/q4$  respectivamente. Também são incluídos na figura, o sinal de temporização de quadro, sinal  $/I25/save$ , e o sinal de temporização de octeto,  $/I25/I10/I308.push$ , que ajudam à visualização dos quadros e dos *time-slots*.

O conteúdo de cada quadro de entrada ao Comutador Digital pode ser verificado na figura 7.9. Utilizando como referência os sinais  $/TS0-1$ ,  $/I25/save$  e  $/I25/I10/I308.push$ , observa-se quais os dados entrantes ao Comutador, depois de realizado o alinhamento dos quadros, e o valor não determinado das saídas  $/q3$  e  $/q4$  durante o intervalo de tempo mostrado, devido ao período de latência.

FIGURA 7.9 – Entradas ao Computador Digital, Comutação Remota.

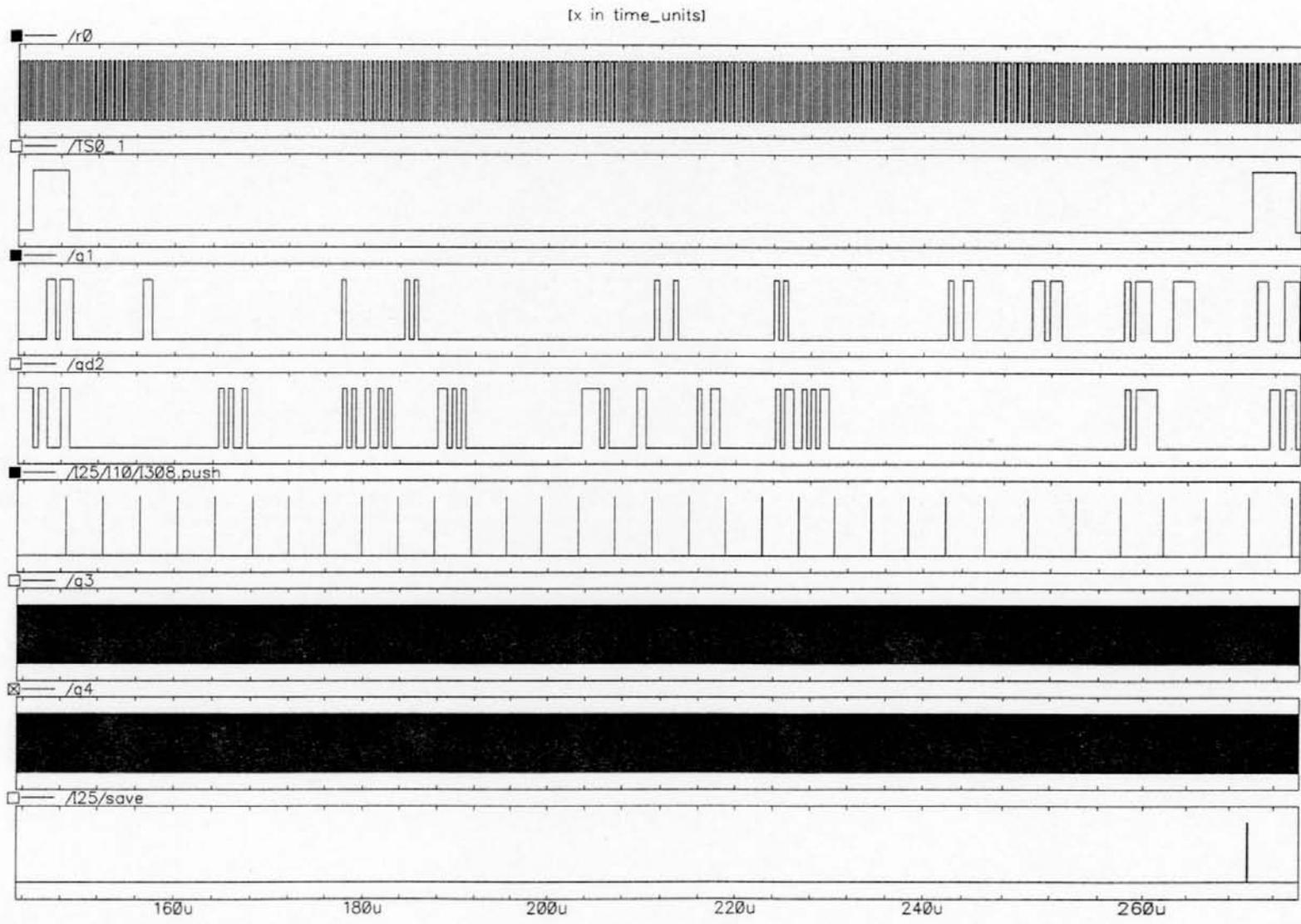
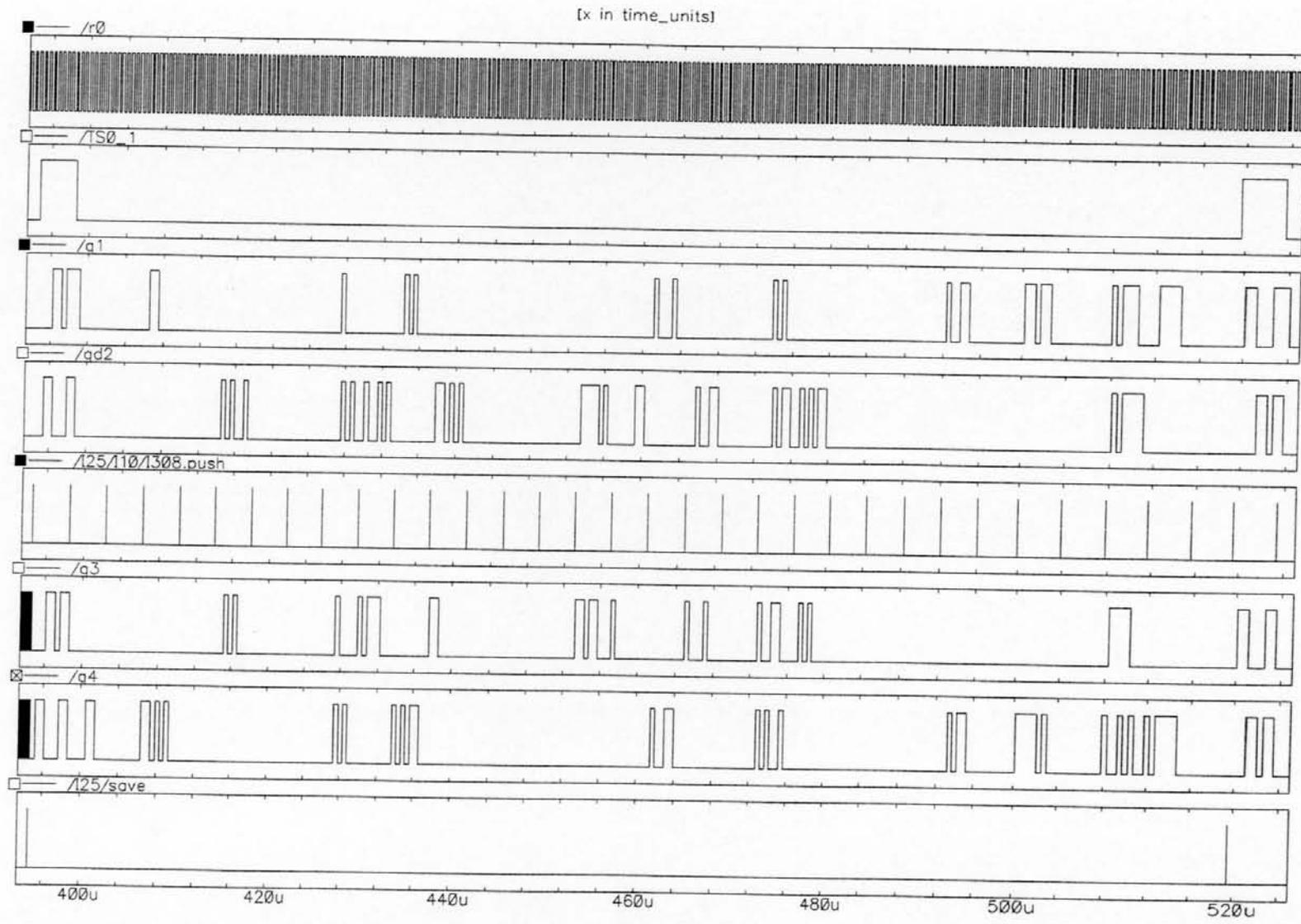


TABELA 7.6 – Quadros de Saída Comutação Remota

Quadros de Saída	
Q3	Q4
TS(00) = 00011011	TS(00) = 01110011
TS(05) = 00101000	TS(01) = 00001100
TS(08) = 00010000	TS(03) = 11010100
TS(09) = 10111000	TS(08) = 00010100
TS(11) = 11000000	TS(10) = 10101100
TS(15) = 11011000	TS(17) = 01001100
TS(16) = 10000000	TS(20) = 10100100
TS(18) = 10001000	TS(25) = 00101100
TS(20) = 10011000	TS(27) = 11110100
TS(21) = 01010000	TS(29) = 10010100
TS(29) = 01111000	TS(30) = 10111100

Os resultados da simulação são mostrados na figura 7.10. Aos  $395\mu s$ , os sinais de saída  $/q3$  e  $/q4$  disponibilizam o conteúdo resultante dos quadros de Q3 e Q4. O quadro Q3 é enviado ao multiplexador remoto, e o quadro Q4 é enviado ao multiplexador local. A verificação das saídas pode ser feita com ajuda do sinal de temporização de octeto  $/I25/I10/I308.push$ . A tabela 7.6 resume o conteúdo de cada *time-slot* dos quadros de saída Q3 e Q4.

FIGURA 7.10 – Saídas do Computador Digital, Comutação Remota.



## 8 CONCLUSÃO

Neste trabalho foi proposta, especificada, desenvolvida e validada mediante simulações lógicas, a arquitetura de um Comutador Digital para TDM Determinístico E1, capaz de fornecer comutação local e remota dentro de uma Rede Corporativa ponto-a-ponto, cujos nós estão constituídos por multiplexadores E1. A arquitetura foi desenvolvida visando sua posterior implementação como um ASIC.

Inicialmente foi desenvolvido um estudo aprofundado dos aspectos relacionados com os Sistemas de Comutação, a Comutação de Circuitos e as Redes Corporativas E1. A hierarquia E1 foi analisada detalhadamente com base na especificação feita pelo ITU-T através de suas recomendações da Série G. A descrição do funcionamento do Multiplexador E1 e de suas principais aplicações, foi realizada utilizando como referência os trabalhos prévios a respeito apresentados em [KAI 94] e [KAI 95a]. Uma exposição da necessidade das funções de comutação dentro das Redes Corporativas E1 e sua importância em termos de oferecimento de serviços aos usuários conectados a tais redes, foi apresentada, juntamente com a descrição do funcionamento de um Sistema de Comutação orientado a fornecer enlaces de comunicação entre usuários locais e remotos, mediante comandos de operador.

Uma vez definido o ambiente de operação do Comutador, foram estabelecidos os requisitos básicos que caracterizam a arquitetura deste dispositivo. Considerou-se que o Comutador Digital deve fornecer comutação local e remota entre qualquer par de usuários pertencentes à rede, independentemente do multiplexador ou nó ao qual estejam conectados; deve também ser configurável mediante comandos de operador, e ser compatível com as características do multiplexador E1 do qual fará parte. Com base nestes requisitos, optou-se por uma arquitetura do Comutador Digital considerando o uso de técnicas TSI, manipulando os sinais de agregado de 2048 Kbit/s presentes na interface de cada multiplexador. Após determinar a necessidade do emprego de técnicas de armazenamento elástico na interface do Comutador Digital, propôs-se a arquitetura de um sistema digital integrado capaz

de implementá-lo. Após esta etapa, foram definidos os módulos funcionais do sistema, e foram especificados os aspectos relativos à temporização e à programação do Comutador Digital. O uso de um protocolo síncrono de comunicações para controlar o enlace ponto-a-ponto entre o Terminal de Operador e o Controle do Comutador foi determinado. As características de transmissão com transparência e controle de erros usando CRC justificaram a escolha do protocolo BSC. Foram também apresentados as recomendações ITU-T consideradas na especificação da arquitetura do Comutador Digital, e foram feitas referências a outros trabalhos sobre comutadores de circuitos pesquisados na literatura científica e industrial.

Finalizando o trabalho, realizou-se a implementação dos módulos funcionais do Comutador Digital, utilizando a biblioteca *Standard Cells* da tecnologia CMOS de  $1.2\mu$ , no ambiente de projeto *SOLO/Cadence*. Descreveu-se os circuitos implementados correspondentes aos geradores de sinais de temporização, aos conversores de dados, às memórias e ao *buffer* elástico. Na maioria dos casos foram incluídos os resultados de simulação dos circuitos de cada módulo. As simulações foram realizadas usando o simulador lógico *SILOS*. A seguir, foi realizada a simulação lógica da arquitetura completa do Comutador Digital. Diversas simulações foram realizadas considerando os três tipos de comutação possíveis: local-local, local-remota, e remota, bem como o caso crítico no alinhamento dos quadros. Os resultados obtidos foram absolutamente satisfatórios validando a arquitetura proposta. Devido a que o simulador *SILOS* modela os transistores CMOS como chaves e considera o efeito de carga das capacitâncias entre as etapas dos circuitos, foi possível avaliar os atrasos decorrentes de cada etapa, bem como calcular a latência do Comutador Digital. A latência máxima de  $375\mu$ s, equivalente a três quadros E1, prevista pela especificação inicial, foi comprovada, e representa um atraso aceitável para este tipo de dispositivo. Os resultados de simulação do Comutador Digital para os três tipos de comutação foram apresentados. Com o objetivo de nortear uma implementação posterior e permitir a continuidade deste trabalho, foi especificado o módulo de controle do Comutador Digital. Foi definido o diagrama de estados do controle, e foi sugerido o uso do microcontrolador MCS 8051 e do controlador serial de mul-

tiprotocolos MPSC 8274, ambos da Intel, como base do sistema. A modularidade e facilidade de extensão que estes dispositivos oferecem ao sistema justificam sua indicação. A última atividade foi descrever mediante diagramas de fluxo, as rotinas correspondentes a cada estado do controle, considerando o *software* do MCS 8051 e do MPSC 8274.

A grande contribuição deste trabalho reside em duas áreas: Comunicação de Dados e Microeletrônica. Na área de Comunicação de Dados, este trabalho contribui com a proposta de uma arquitetura para um equipamento Comutador de dados TDM capaz de fornecer funções de comutação dentro do ambiente de um End-Mux E1. A arquitetura do Comutador foi projetada atendendo as recomendações ITU-T relativas à hierarquia E1 vigentes no Brasil, e com o objetivo de constituir-se em um módulo do multiplexador E1 descrito em [KAI 95a], aumentando assim o número de funções disponíveis deste equipamento. Na área de Microeletrônica, contribui mostrando que o desenvolvimento de um ASIC para sistemas deste tipo, é uma solução atrativa para incrementar o desempenho e a confiabilidade do sistema. Do mesmo modo, este trabalho amplia o espectro de equipamentos para comunicação de dados digitais desenvolvidos pelo Grupo de Microeletrônica (GME) da UFRGS.

Como atividades futuras pode-se citar:

- Geração do *layout* do circuito do Comutador Digital, e sua posterior fabricação como Circuito Integrado (CI).
- Implementação das rotinas correspondentes a cada estado do Controle do Comutador Digital em linguagem *Assembler* do 8051.
- Projeto e implementação da placa de circuito impresso do Sistema de Comutação (Comutador e Controle) utilizando o ASIC do Comutador, o 8051, o 8274 e componentes discretos.

## ANEXO A-1 ROTINAS PARA O MÓDULO DE CONTROLE DO COMUTADOR DIGITAL

### A-1.1 A Rotina *Start*

Esta rotina realiza as operações necessárias para a inicialização do sistema: habilitação das interrupções externas do 8051, estabelecimento de prioridades entre elas, programação do MPSC 8274, e definição do *buffer* de recepção de dados na memória externa. O microcontrolador deve verificar também, o modo de operação do nó ao qual está conectado. Caso estiver operando em modo *Escravo*, deve passar ao estado *Slave*.

A figura A-1.1 mostra o diagrama de fluxo correspondente à rotina que implementa o estado *Start*. O 8051 deve ser inicializado atribuindo a interrupção externa 0 (EXT0) à rotina *Stop*, e a interrupção externa 1 (EXT1) à rotina *Comunic*. A prioridade maior deve ser atribuída a EXT0. Os endereços correspondentes aos vetores de interrupção do 8051 [INT 86], aparecem na tabela A-1.1.

O MPSC 8274 deve ser programado para operar com o protocolo BSC, em modo transparente através do canal A, com inclusão de caracteres CRC-16 e com dois caracteres de sincronismo (BISYNC). A transmissão e a recepção devem ser ativadas pelo modem mediante os sinais CTS (*Clear To Send*) e CD (*Carrier Detect*). O 8274 gera uma interrupção por cada caractere recebido [INT 87]. A interface de acesso

TABELA A-1.1 – Interrupções do 8051

Símbolo	Endereço	Fonte de Interrupção
RESET	00H	<i>Power Up or Reset</i>
EXTI0	03H	Interrupção Externa 0
EXTI1	13H	Interrupção Externa 1
SINT	23H	Interrupção do Porto Serial

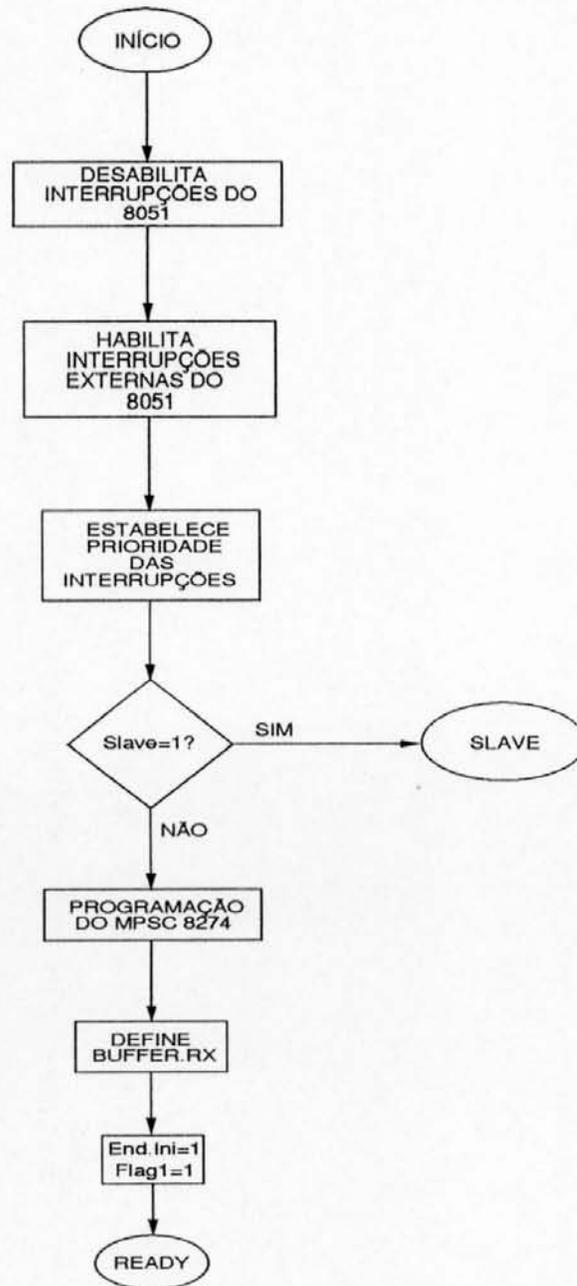
FIGURA A-1.1 – Diagrama de Fluxo da Rotina *Start*.

TABELA A-1.2 – Interface do 8274

/CS	A1	A0	Leitura	Escrita
0	0	0	Dados no Canal A	Dados no Canal A
0	1	0	Status (RR0, RR1)	Comando/Parâmetro (WR0, WR7)
0	0	1	Dados no Canal B	Dados no Canal B
0	1	1	Status (RR0, RR1)	Comando/Parâmetro (WR0, WR7)
1	X	X	Z	Z

ao 8274 é definida pela tabela A-1.2. As instruções de inicialização e programação do 8051 e do 8274 mostram-se a seguir<sup>1</sup>:

-- 8051

MOV IE, #00H ; Desabilita Interrupcoes

MOV SP, 40H ; Move a pilha 'a parte superior da RAM interna

MOV IP, #01H ; Estabelece a prioridade maior a EXT0

MOV R0, #BUFF.RX ; O endereco de inicio do buffer de recepcao de  
; dados, e' armazenado no registrador R0

-- 8274

; Canal A

WR0: 00H, 18H ; Reseta o canal

WR0: 00H, 80H ; Reseta o Gerador de CRC do Transmissor

WR2: 02H, 10H ; Prioridade 0, Ambos canais em modo Interrupcao

WR4: 04H, 10H ; Define SYNC de 16 bits e nao paridade

WR1: 01H, 1BH ; Habilita interrupcao Rx por cada dado recebido,  
; interrupcao Tx e interrupcoes externas

WR3: 03H, F3H ; Define Rx com 8 bits/dado, Modo Auto Enable,  
; Fase Hunt, inibe SYNC, habilita Rx

WR5: 05H, EFH ; Define Tx com 8 bits/dado, DTR=1, RTS=1,  
; habilita Tx e CRC-16

WR6: 06H, 16H ; Byte inferior do caractere SYNC

WR7: 07H, 16H ; Byte superior do caractere SYNC

; Canal B

<sup>1</sup>O 8274 possui registradores de escrita e leitura identificados como WR e RR respectivamente. Assim, WR5 é o registrador de escrita número 5, RR1 é o registrador de leitura número 1, etc.

WR2: 02H, 00H ; Vetor de Interrupcao

WR1: 01H, 1CH ; Habilita Status Affect Vector

## A-1.2 A rotina *Ready*

O estado *Ready* corresponde à operação normal do sistema. Uma vez inicializado, a variável *flag1* habilitada (nível lógico 1) leva o sistema a programar o Comutador Digital com enlaces pre-estabelecidos armazenados na ROM do microcontrolador (programação *default*). Depois disto, o sistema fica na espera de dados de programação enviados pelo operador da rede através do canal serial gerenciado pelo 8274. Enquanto fica aguardando dados, rotinas de supervisão do multiplexador e de atualização dos portos de saída do microcontrolador (conectados a um painel de controle), podem ser executadas. A variável *ST.CTS*, corresponde ao status do sinal CTS fornecido pelo modem. Se CTS é ativado ( $CTS=0$ ), significa que o modem está pronto para transmitir. Para manter o sincronismo no canal serial quando não existem dados para enviar, é recomendável transmitir o caractere de sincronismo SYNC.

Concluída a recepção de dados ( $end.rx=1$ ), realiza-se a identificação do tipo de operação solicitado. Isto implica que entre os dados recebidos devem estar presentes códigos de operação, que devem ter sido definidos previamente. Foram considerados dois tipos de operação: programação e teste. No caso da programação, esta pode ser realizada usando os dados recebidos, ou usando os enlaces *default*, isto é, com os dados armazenados na ROM do sistema. No caso do teste, este considera o estabelecimento de laços locais internos entre canais do multiplexador. Uma interface adequada entre o MUX e o controle deve ser definida para habilitar tais laços. Os laços de teste podem ser locais ou remotos, sendo que estes últimos podem ser implementados usando o comutador, tal e como descrito no capítulo 5. Toda informação sobre laços de teste deve ser conhecida por ambos os nós da

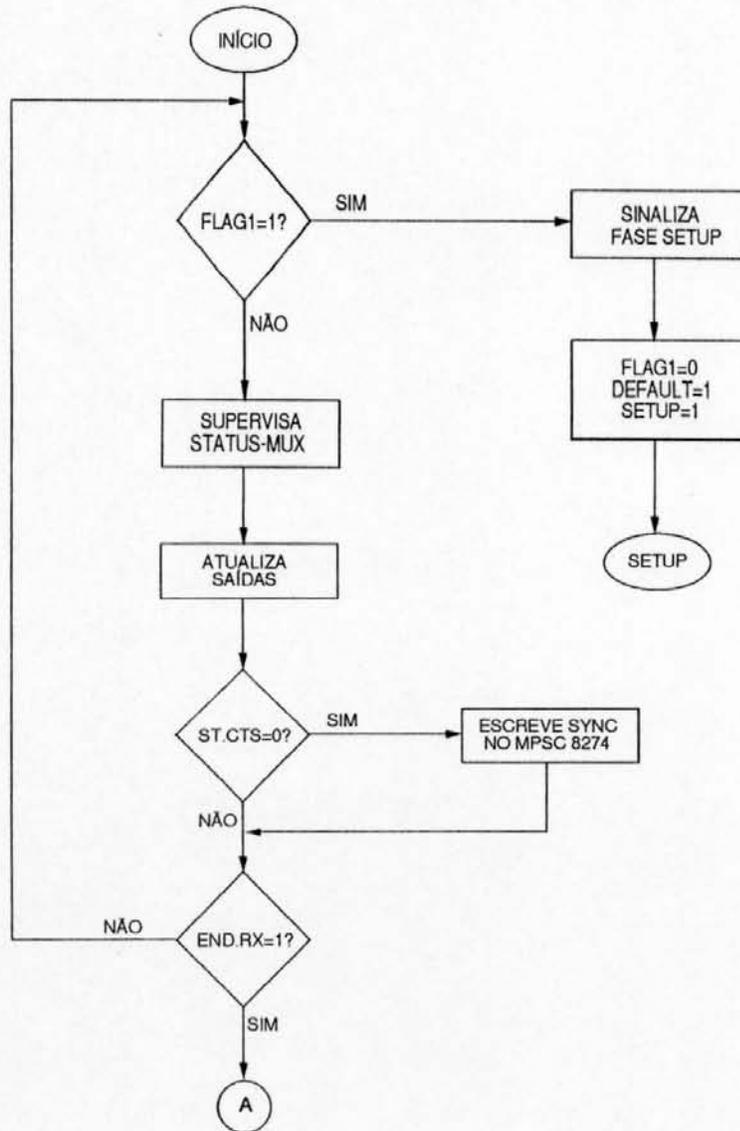
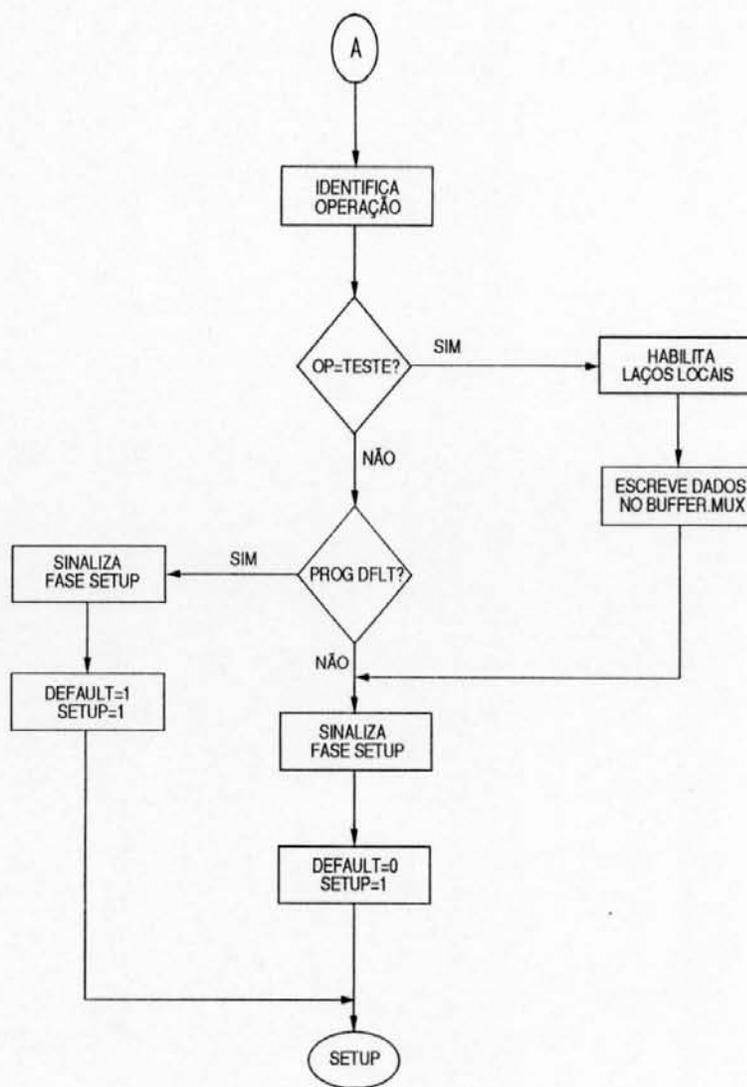


FIGURA A-1.2 – Diagrama de Fluxo da Rotina *Ready* (1/2).

rede, portanto, o controle deve utilizar o canal de 20 Kbit/s do multiplexador para transmitir tal informação. Para isto, o controle deve escrever os dados adequados no *buffer* do multiplexador (*buffer.mux*). As figuras A-1.2 e A-1.3 mostram o diagrama de fluxo correspondente à rotina que implementa o estado *Ready*.

### A-1.3 A Rotina *Setup*

Ao entrar no estado *Setup*, o sistema inicia o processo de configuração da memória de conexões CM. Antes de começar a escrita da memória CM, o controle

FIGURA A-1.3 – Diagrama de Fluxo da Rotina *Ready* (2/2).

verifica se o Comutador está pronto para ser reconfigurado. Isto depende principalmente do estado atual do multiplexador, pois é necessário que todos os enlaces de comunicação atuais tenham concluído para estabelecer os novos enlaces. Verificado o status do multiplexador, o pino de habilitação do Comutador EN é desativado (EN=0), e o pino PROG é ativado (PROG=1). Em seguida, o controle escreve os dados de programação na memória CM. Estes dados são lidos a partir do *buffer* de recepção (*buffer.rx*) definido na RAM externa, ou a partir da lista de enlaces definidos na ROM do sistema (*buffer.dflt*), no caso *default*. Finalizada a programação, o pino PROG é desativado (PROG=0), e habilita-se o Comutador fazendo EN=1. O diagrama de fluxo correspondente à rotina que implementa o estado *Setup* é mostrado na figura A-1.4.

#### A-1.4 A Rotina *Comunic*

Quando a interrupção EXT1 é habilitada, o controle passa a executar uma rotina de transmissão e recepção de dados. Esta rotina é chamada *Comunic* mostrada na figura A-1.5.

O MPSC 8274 foi programado para estabelecer um enlace ponto-a-ponto usando o protocolo BSC em modo transparente, com dois caracteres de sincronismo (BISYNC) e CRC-16. A transmissão de dados foi condicionada ao estado ativo do sinal CTS (CTS=1, pronto para transmitir), e a recepção de dados ao estado ativo do sinal CD (CD=1, presença de portadora), ambos sinais fornecidos pelo modem. Cada interrupção gerada pelo MPSC 8274 corresponde ao vetor de interrupções mostrado na tabela A-1.3 [INT 87].

As figuras A-1.6, A-1.7, A-1.8, A-1.9, A-1.10 e A-1.11 mostram os diagramas de fluxo das rotinas que atendem o vetor de interrupção do 8274. Para iniciar a transmissão, o primeiro caractere de dados deve ser escrito no *buffer* de transmissão do MPSC 8274. Este caractere será enviado se CTS está ativo. Uma nova interrupção será gerada cada vez que o *buffer* de transmissão fique vazio. Neste

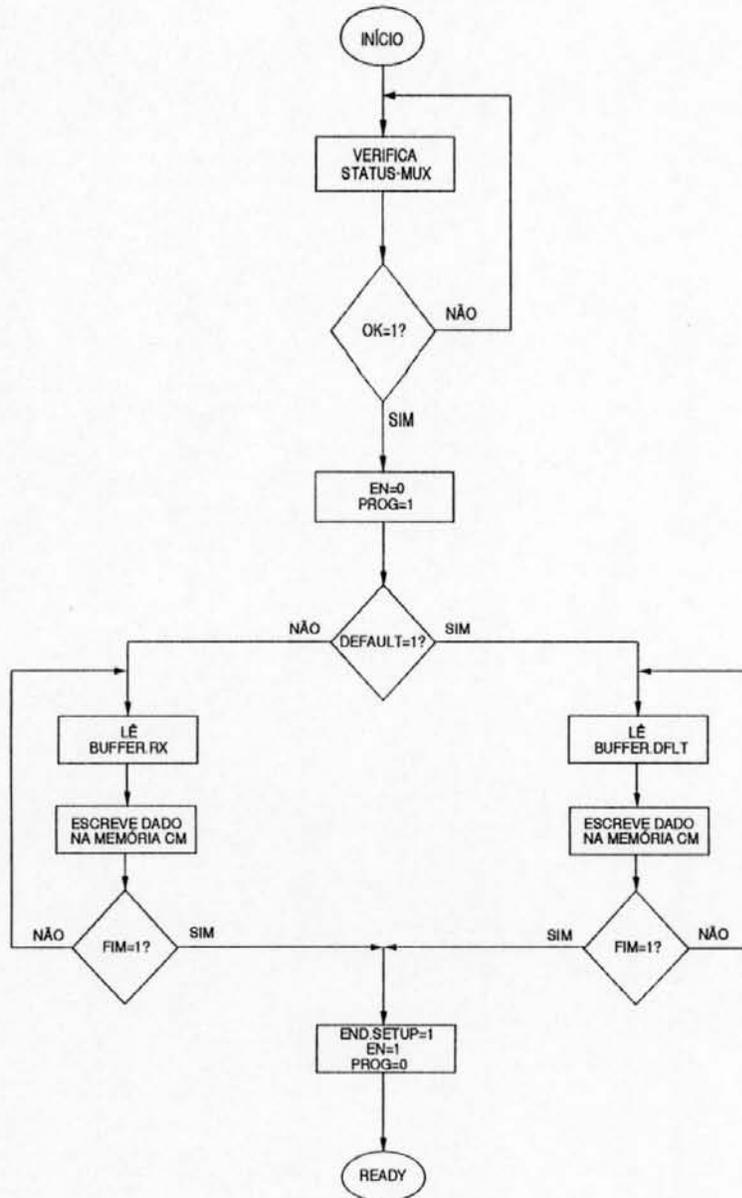
FIGURA A-1.4 – Diagrama de Fluxo da Rotina *Setup*.

TABELA A-1.3 – Vetor de Interrupções do 8274

Canal	V2 V1 V0	Fonte de Interrupção
B	0 0 0	<i>Tx Buffer Empty</i>
	0 0 1	<i>Ext. Status Change</i>
	0 1 0	<i>Rx Character Available</i>
	0 1 1	<i>Special Rx Condition</i>
A	1 0 0	<i>Tx Buffer Empty</i>
	1 0 1	<i>Ext. Status Change</i>
	1 1 0	<i>Rx Character Available</i>
	1 1 1	<i>Special Rx Condition</i>

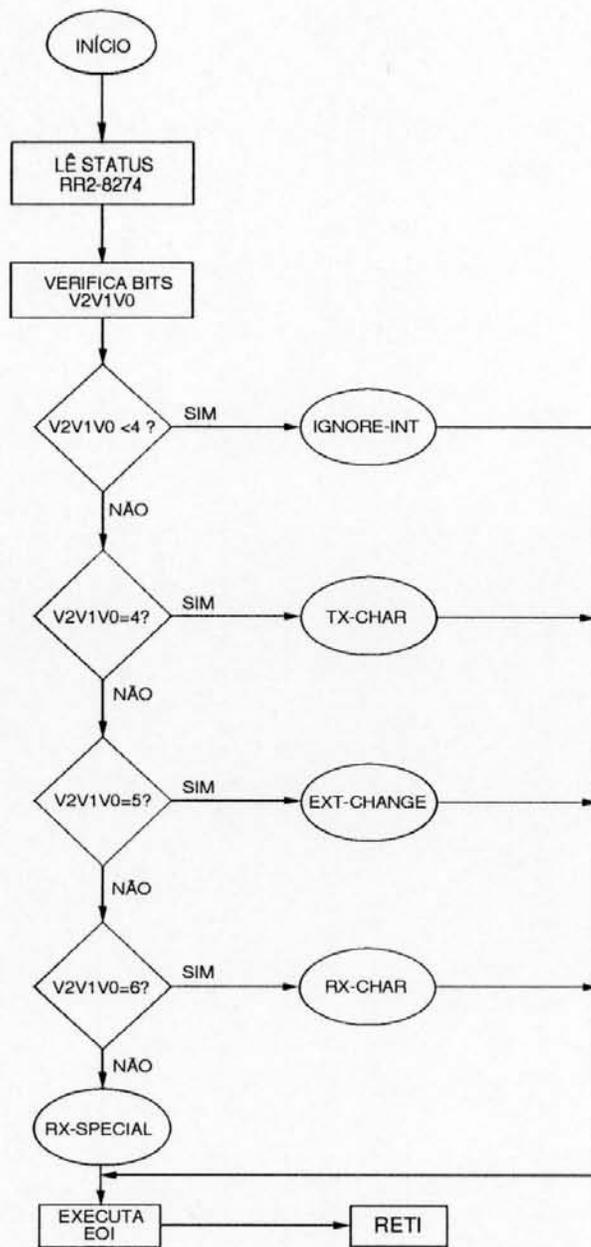


FIGURA A-1.5 – Diagrama de Fluxo da Rotina *Comunic*.

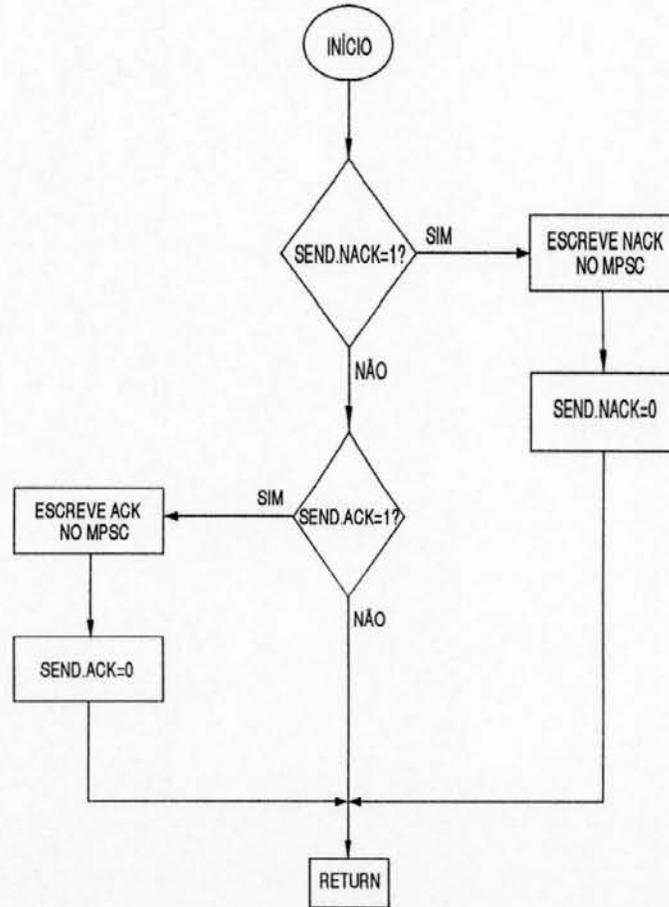


FIGURA A-1.6 – Diagrama de Fluxo da Rotina *Tx-Char*.

caso, existem duas coisas a fazer: enviam-se os dados restantes, ou espera-se pela condição *Transmit Underrun*. Esta condição enviará o CRC gerado, se o comando *Reset Transmit Underrun/EOM* foi utilizado previamente. É recomendável executar este comando (escrever em WR0 o dado 11000000)<sup>2</sup>, logo após de ser enviado o primeiro caractere válido [INT 87]. Assim, quando todos os caracteres tenham sido enviados, O CRC será transmitido na primeira ocorrência da condição *Transmit Underrun*.

Para transmitir em modo transparente, o *bit Tx CRC enable* de WR5 deve ser desabilitado (*bit 0* de WR5 igual a zero) cada vez que vai ser enviado um caractere especial. Os casos possíveis são: DLE STX, DLE ETB, DLE ETX, DLE SYN, e DLE DLE. Se o caractere seguinte aos caracteres de controle deve ser incluído no CRC, então habilitar novamente o *bit Tx CRC enable*. Para finalizar a transmissão,

<sup>2</sup>o MSB é o primeiro número à esquerda, bit 7

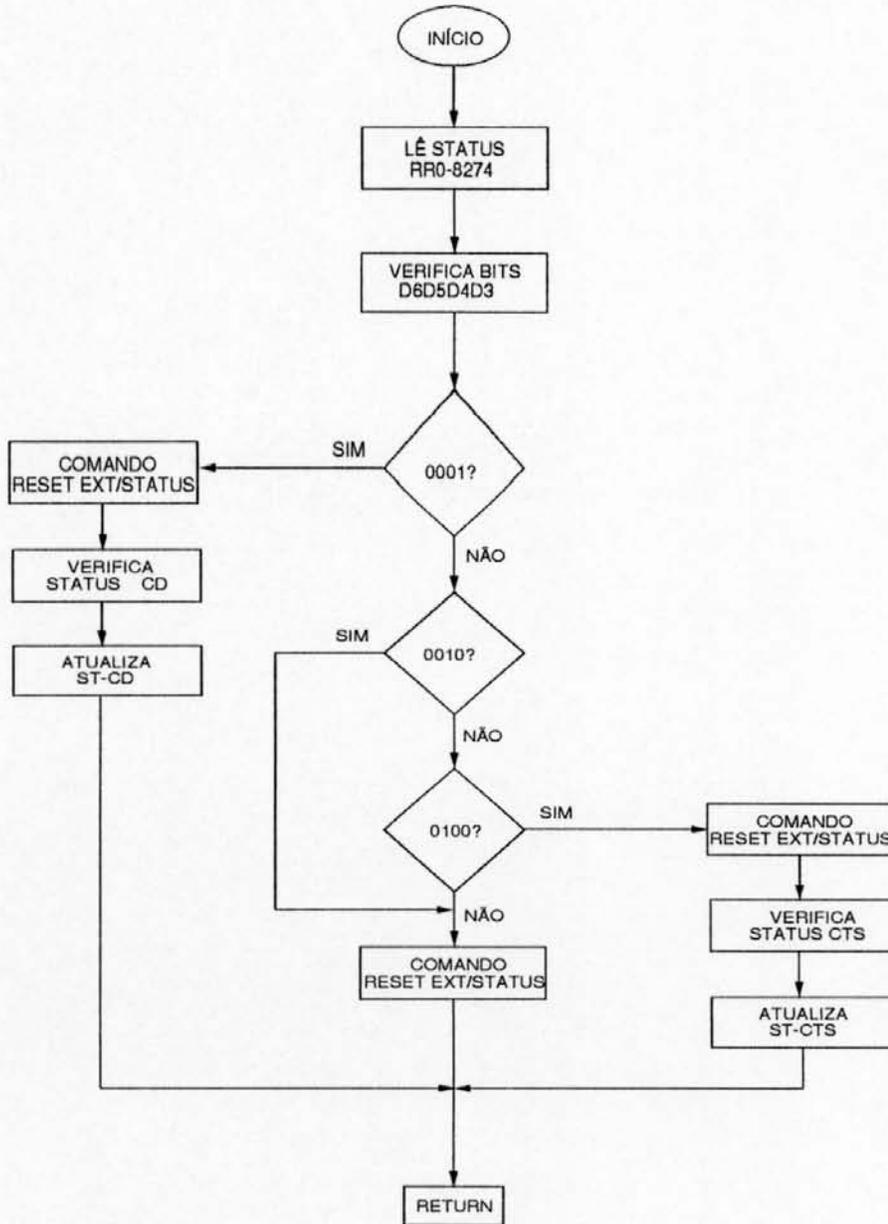


FIGURA A-1.7 – Diagrama de Fluxo da Rotina *Ext-Change*.

desabilitar os *bits Tx Enable e RTS* do registrador WR5 (escrever nos *bits 3 e 1* de WR5 o valor 0).

A recepção começa quando o sinal CD é ativado (CD=1). Depois de receber o primeiro dado válido, é recomendável inicializar o verificador de CRC (escrever nos *bits 7 e 6* de WR0 o valor 0 e 1, respectivamente), e seguidamente habilitar o gerador de CRC do receptor (escrever no *bit 3* de WR3 o valor 1). Durante a recepção é possível excluir os caracteres especiais recebidos do cálculo do CRC, para isto é preciso desabilitar o gerador de CRC do receptor depois de receber

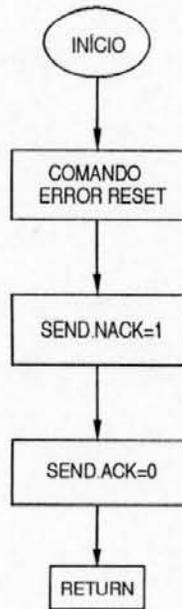


FIGURA A-1.8 – Diagrama de Fluxo da Rotina *Rx-Special*.

um caractere especial (escrever no *bit* 3 de WR3 o valor 0). Devido a que cada dado recebido passa pelo verificador de CRC 8 tempos de *bit* depois de chegar ao 8274, durante este tempo, o microcontrolador deve ler o dado e decidir se deve ser incluído ou não no cálculo do CRC.

Depois de ler os *bytes* CRC recebidos, o microcontrolador deve ler mais dois caracteres (possivelmente SYNC), antes de verificar o resultado do CRC no *bit* 6 do registrador RR1. Se uma interrupção foi gerada por uma *Special Rx Condition*, a causa é uma condição *Receive Underrun*. Neste caso, executar o comando *Error Reset* (escrever o dado 00110000 no registrador WR0). No caso de uma interrupção por *Ext. Status Change* escrever o dado 00010000 no registrador WR0.

## A-1.5 A Rotina *Stop*

Esta rotina provoca uma pausa na operação do controle. O retorno ao modo normal está condicionado à habilitação do sinal *end.brk* (nível lógico 1). A figura A-1.12 mostra o diagrama de fluxo desta rotina.

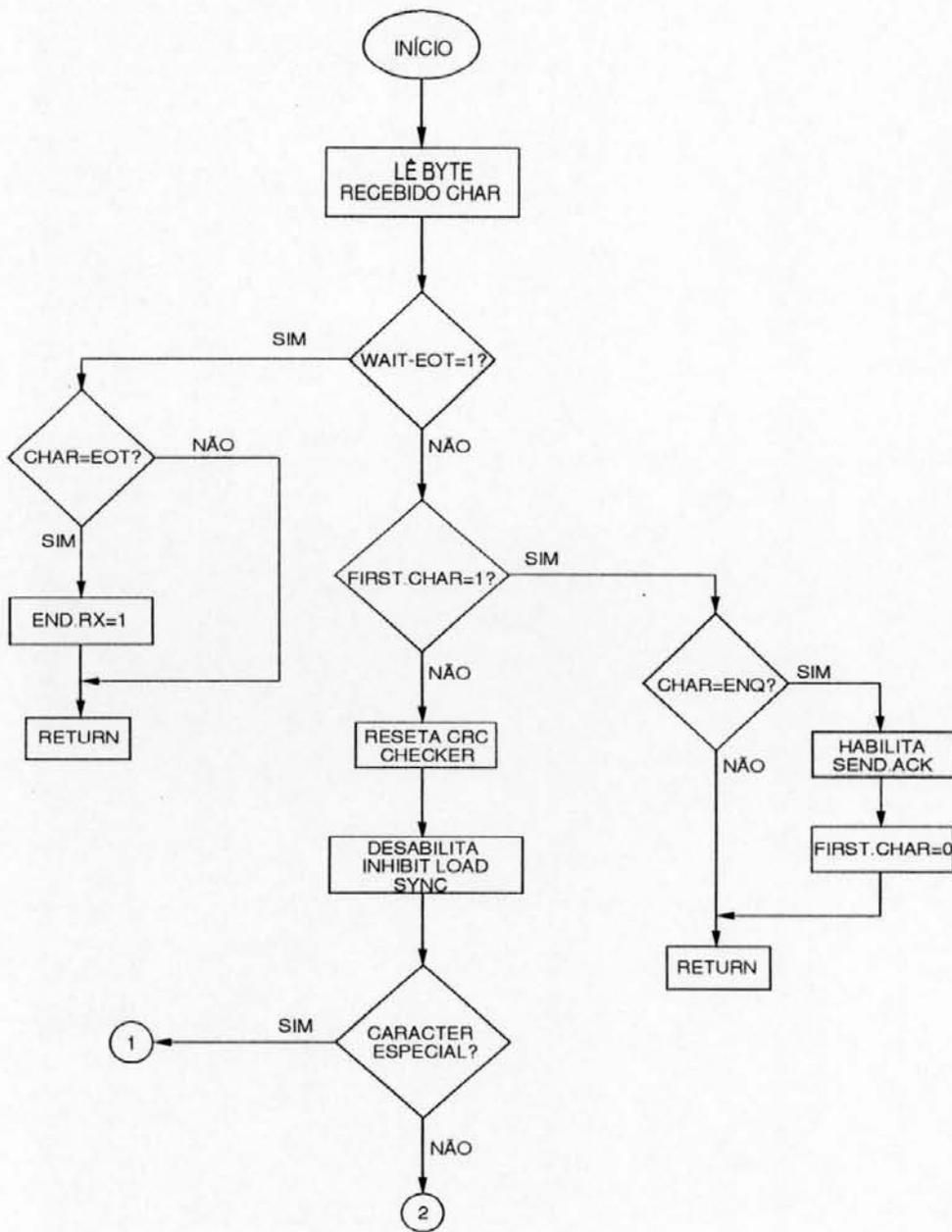


FIGURA A-1.9 – Diagrama de Fluxo da Rotina *Rx-Char* (1/3).

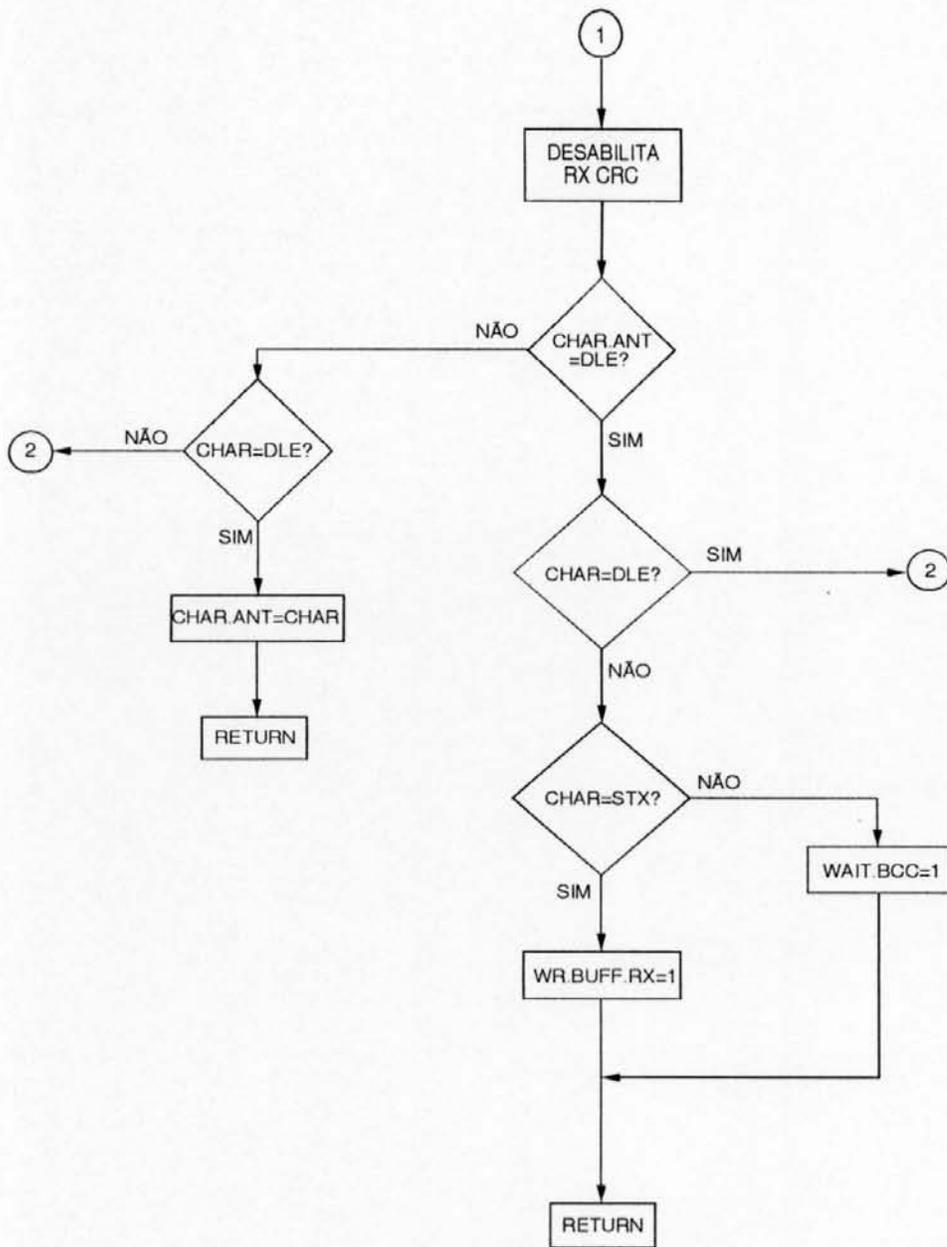


FIGURA A-1.10 – Diagrama de Fluxo da Rotina *Rx-Char* (2/3).

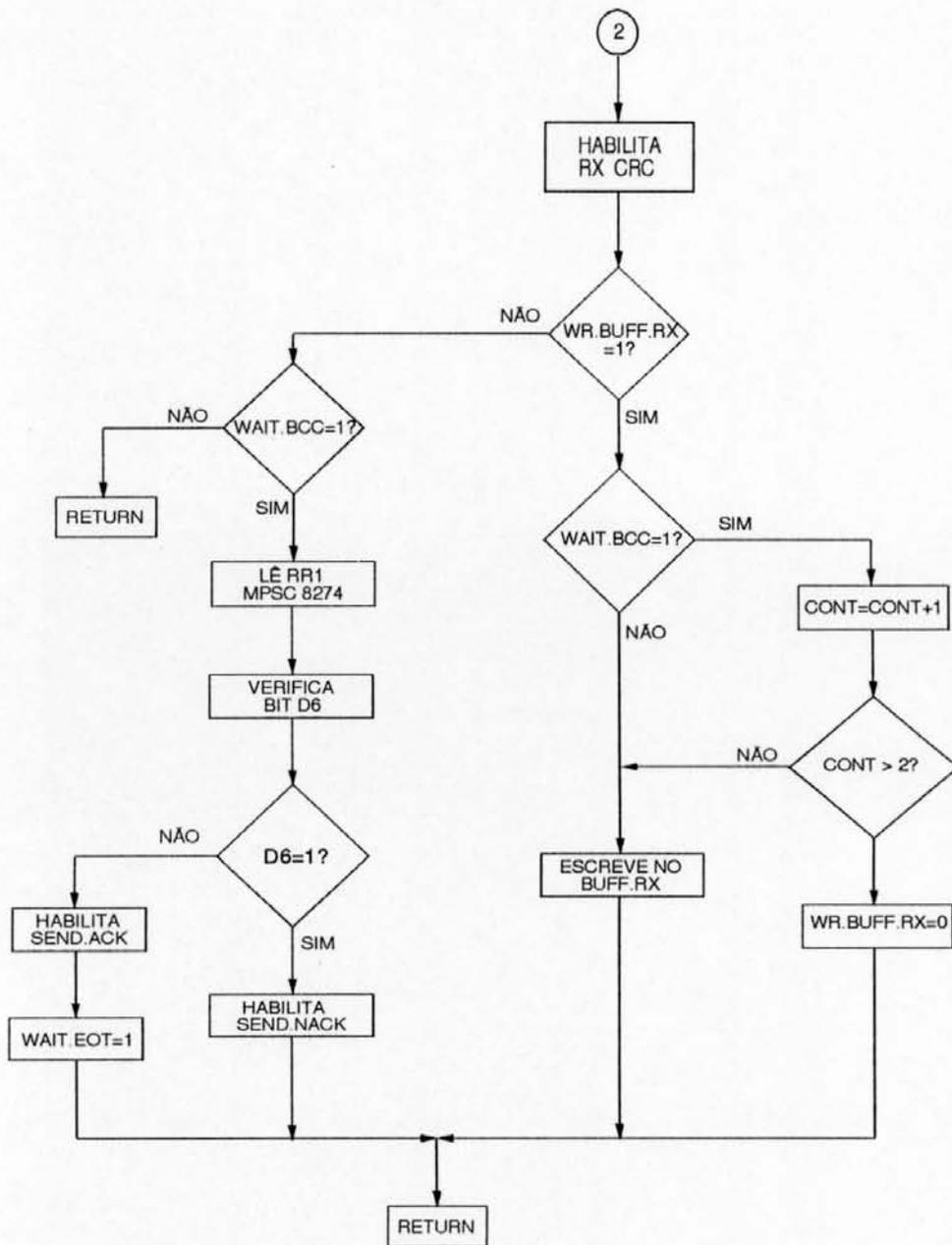
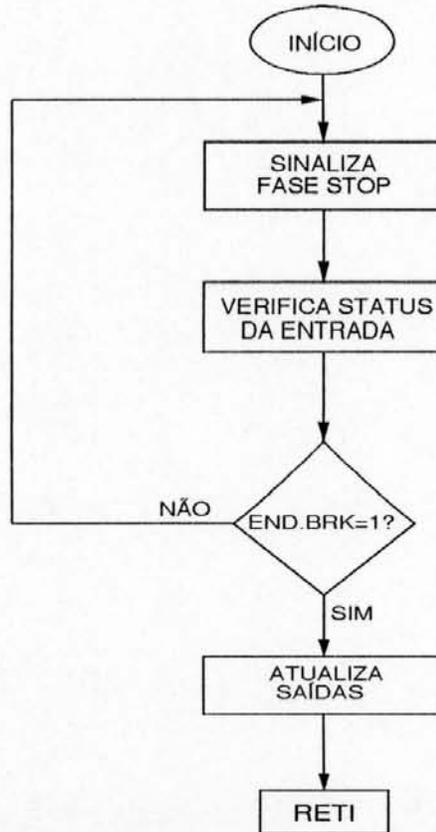
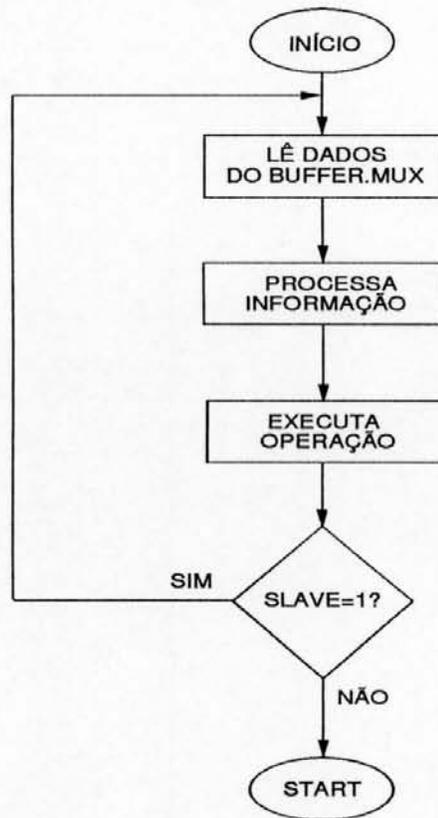


FIGURA A-1.11 - Diagrama de Fluxo da Rotina *Rx-Char* (3/3).

FIGURA A-1.12 – Diagrama de Fluxo da Rotina *Stop*.

### A-1.6 A Rotina *Slave*

Neste modo de operação, o controle espera pela chegada de dados através do canal 20 Kbit/s do multiplexador. A informação recebida é processada, e deve originar a execução de uma tarefa. Aqui considerou-se a habilitação de laços de teste locais no multiplexador remoto. O diagrama de fluxo desta rotina aparece na figura A-1.13.

FIGURA A-1.13 – Diagrama de Fluxo da Rotina *Slave*.

## ANEXO A-2 CONTEÚDO DA MEMÓRIA DE CONEXÕES

A seguir, faz-se a listagem do conteúdo da memória *romx*, utilizada para a simulação do Computador Digital.

### Low Memory

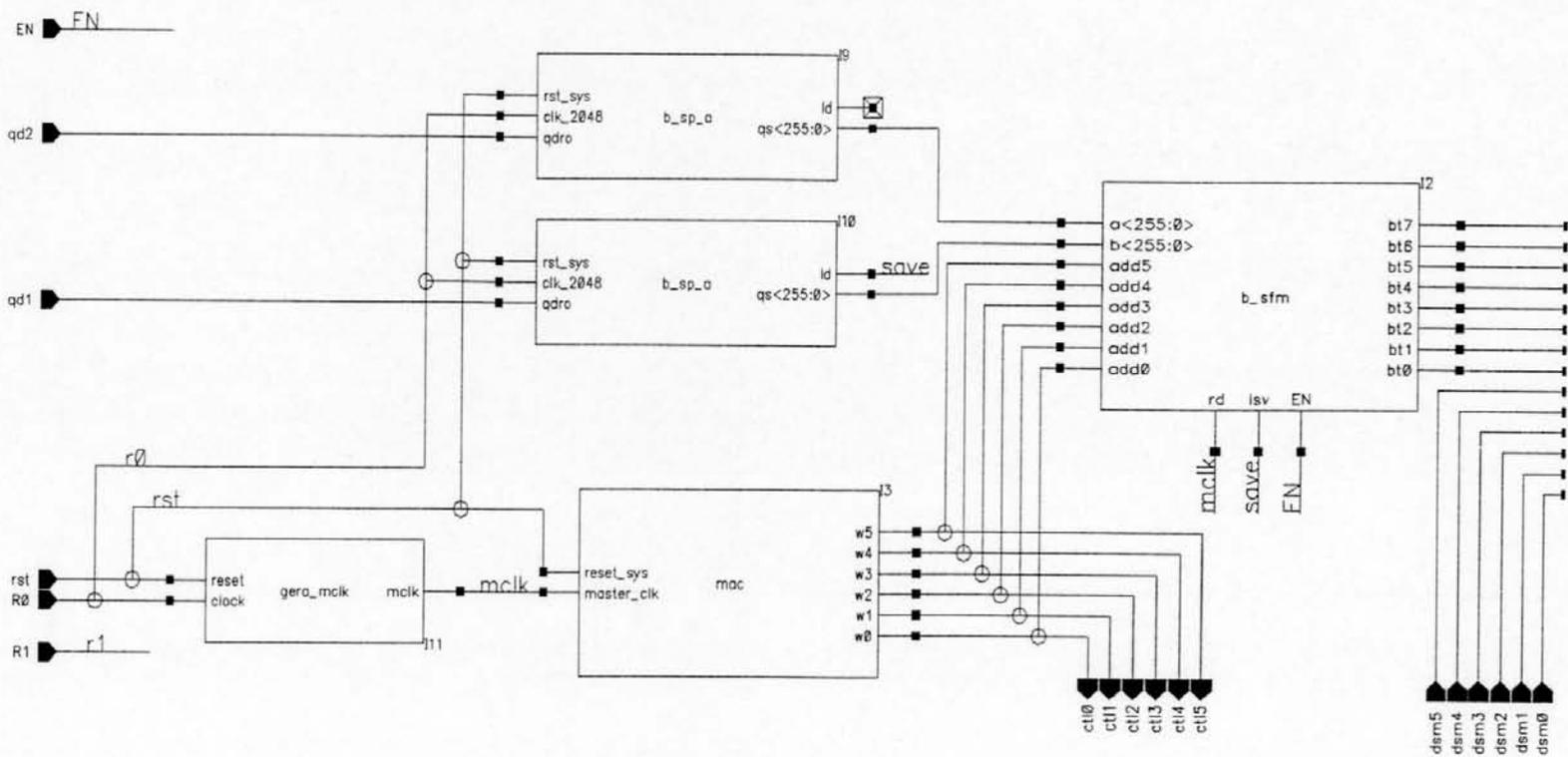
```
0 000000
1 010000
2 011010
3 001011
4 001100
5 011111
6 110110
7 000001
8 001000
9 000010
10 010101
11 111111
12 101111
13 001101
14 110000
15 101100
16 101110
17 010010
18 010001
19 010011
20 000101
21 000110
22 100110
23 111000
24 110111
25 010100
26 111010
27 001111
28 011011
29 001001
30 011101
31 101011
```

## High Memory

32 100000  
33 100111  
34 101001  
35 000111  
36 011110  
37 110100  
38 110101  
39 000011  
40 101000  
41 111101  
42 001110  
43 100011  
44 100100  
45 101101  
46 001010  
47 111011  
48 100001  
49 110010  
50 110001  
51 110011  
52 111001  
53 101010  
54 011000  
55 011100  
56 010110  
57 011001  
58 100010  
59 111100  
60 010111  
61 111110  
62 000100  
63 100101

**ANEXO A-3    SUBCIRCUITOS DO MÓDULO  
MATRIC**

FIGURA A-3.1 – Módulo MATRIC (1/2).



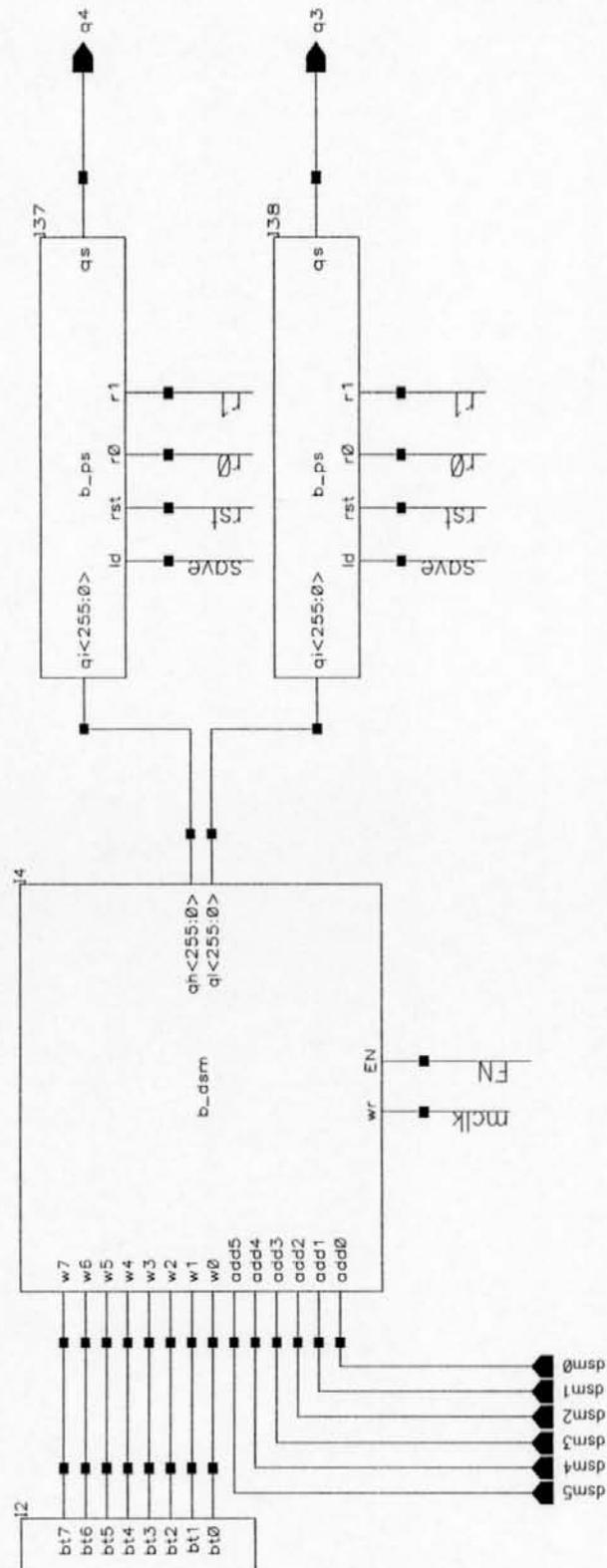


FIGURA A-3.2 – Módulo MATRIC (2/2).

**ANEXO A-4    SUBCIRCUITOS DOS  
MÓDULOS GERADORES DE  
SINAIS**



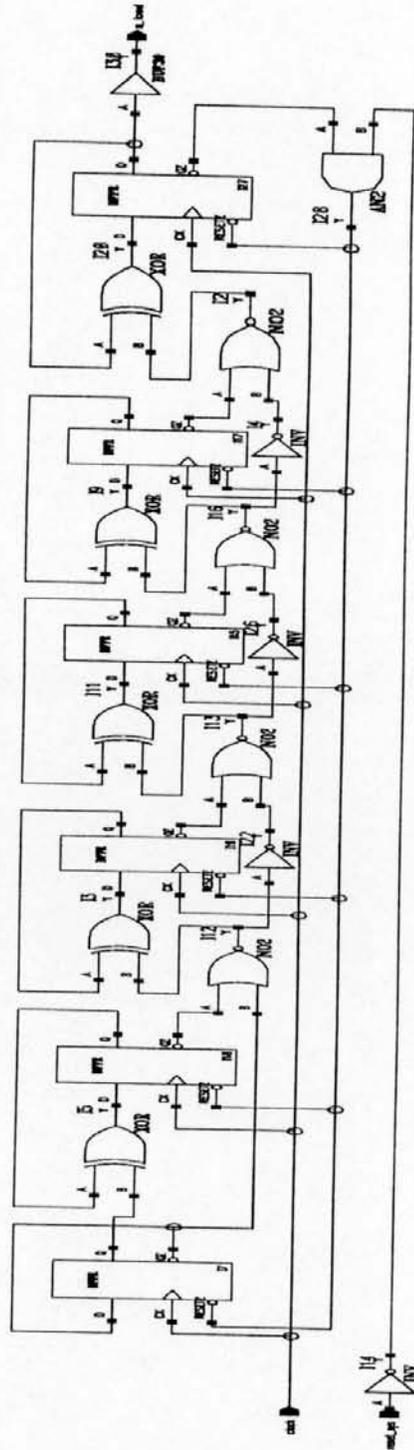
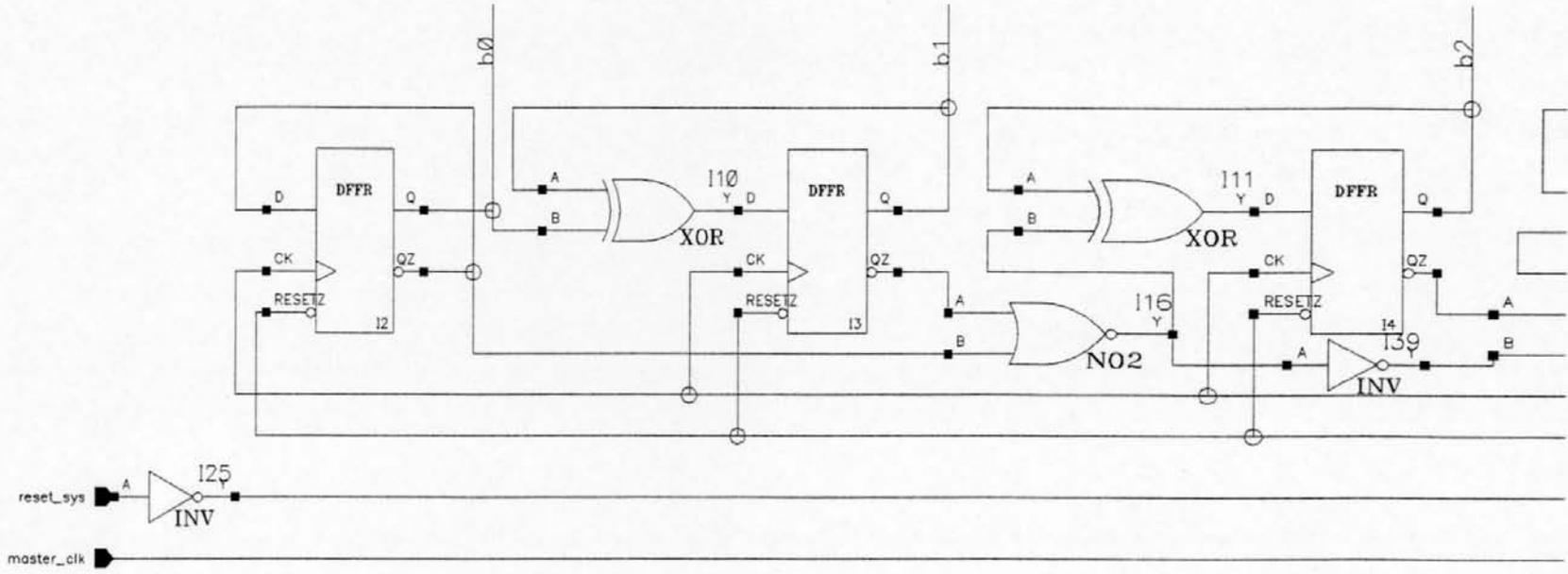


FIGURA A-4.3 - Módulo gen-save.

FIGURA A-4.4 - Módulo mac (1/3).



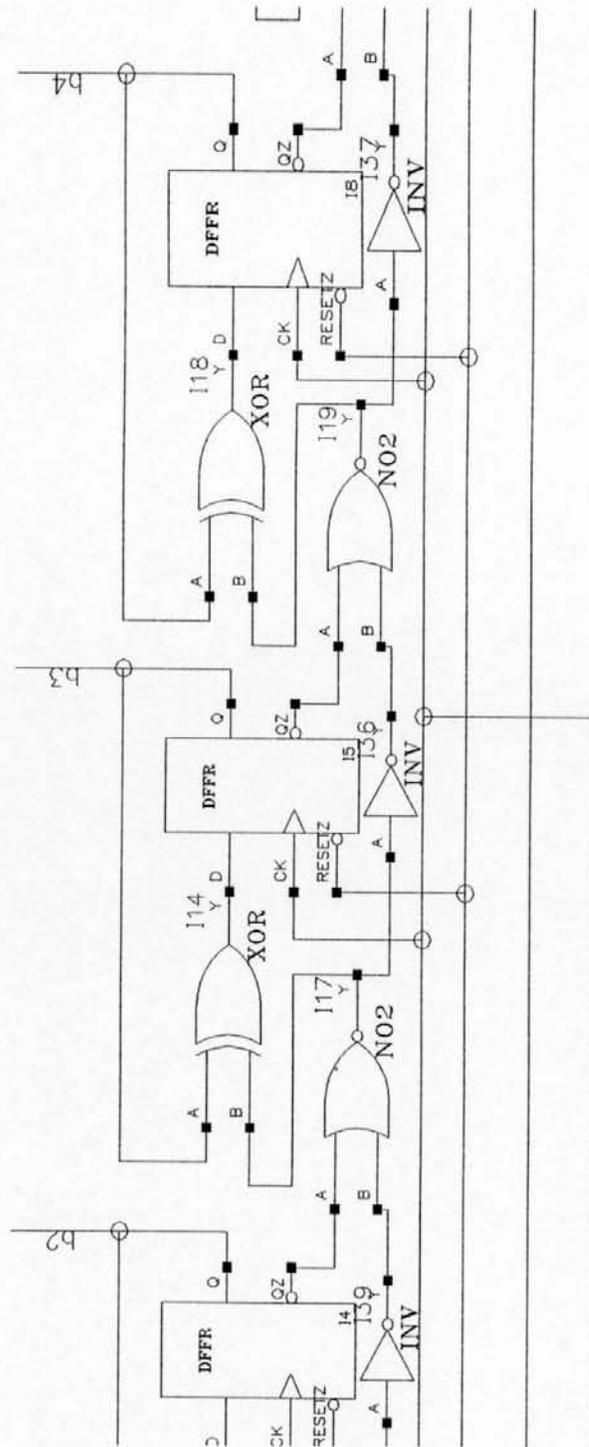


FIGURA A-4.5 – Módulo mac (2/3).

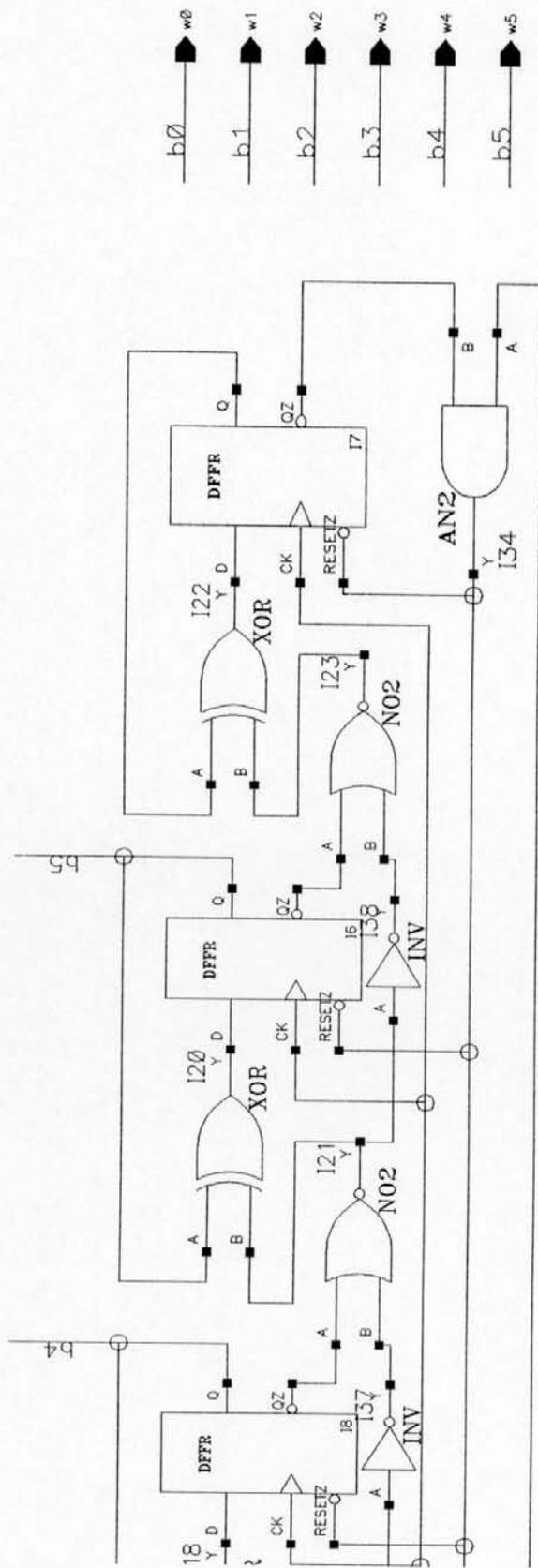


FIGURA A-4.6 - Módulo mac (3/3).

**ANEXO A-5    SUBCIRCUITOS DO  
CONVERTOR SÉRIE A  
PARALELO**

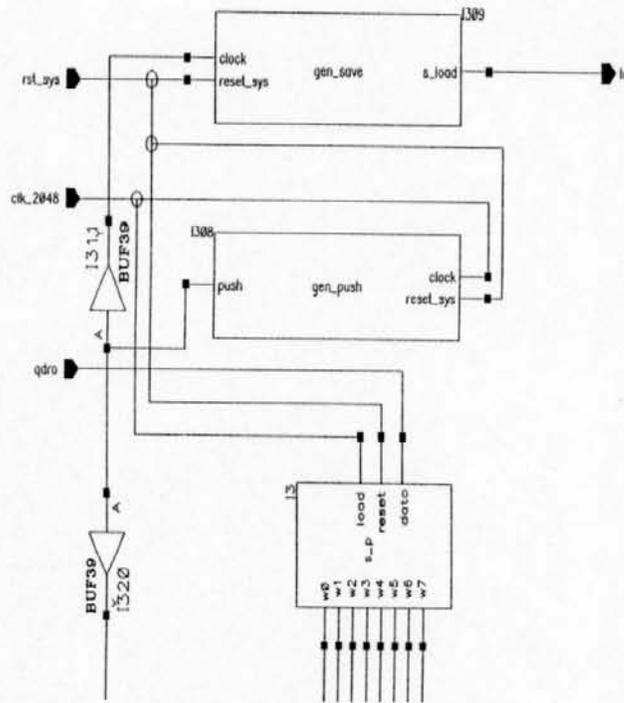


FIGURA A-5.1 – Módulo b-sp-a (1/12).

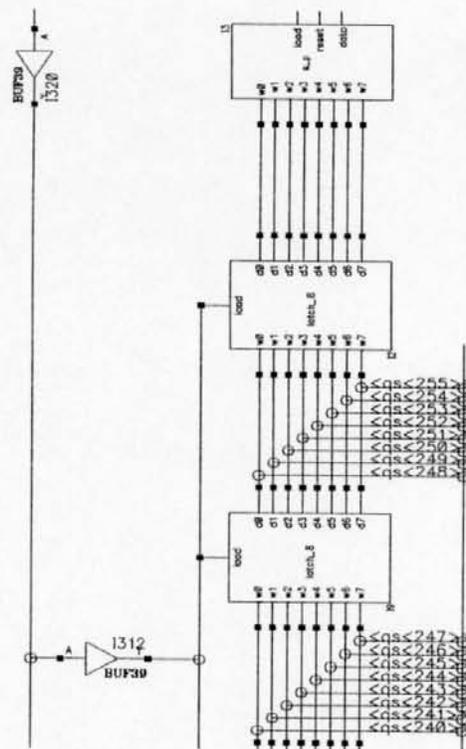


FIGURA A-5.2 – Módulo b-sp-a (2/12).

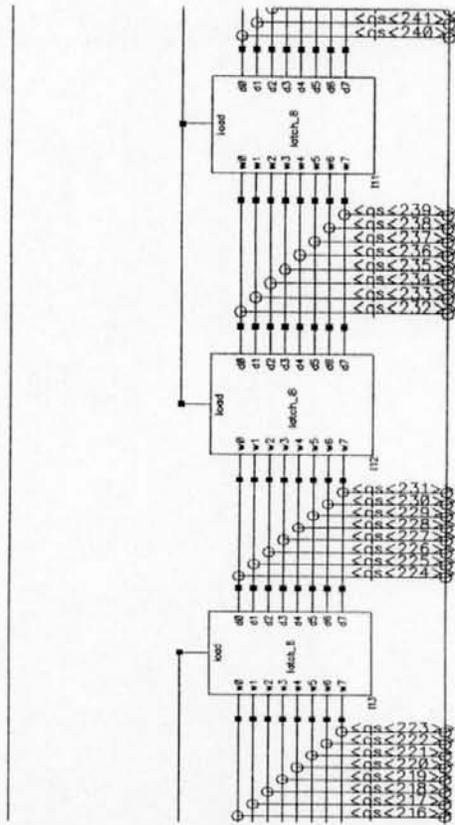


FIGURA A-5.3 – Módulo b-sp-a (3/12).

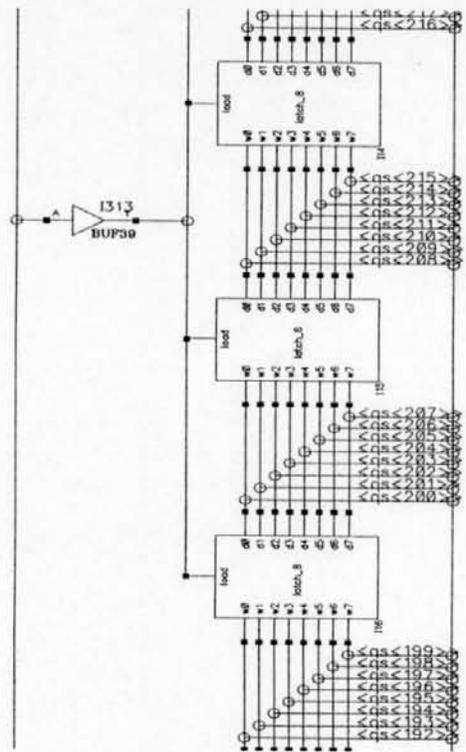


FIGURA A-5.4 – Módulo b-sp-a (4/12).

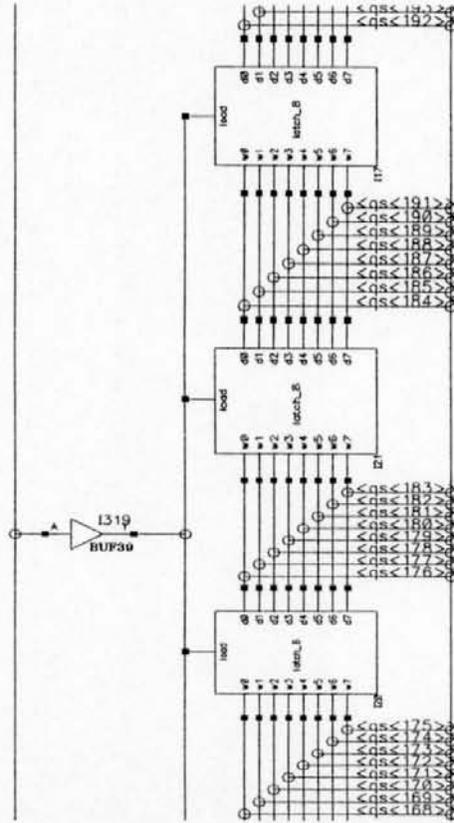


FIGURA A-5.5 – Módulo b-sp-a (5/12).

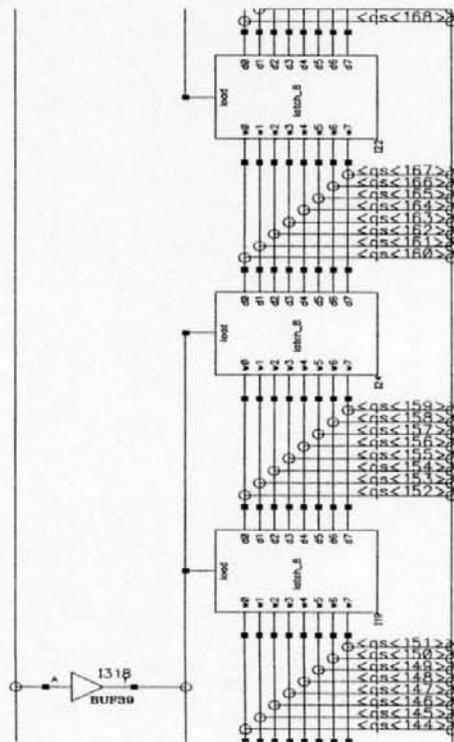


FIGURA A-5.6 – Módulo b-sp-a (6/12).

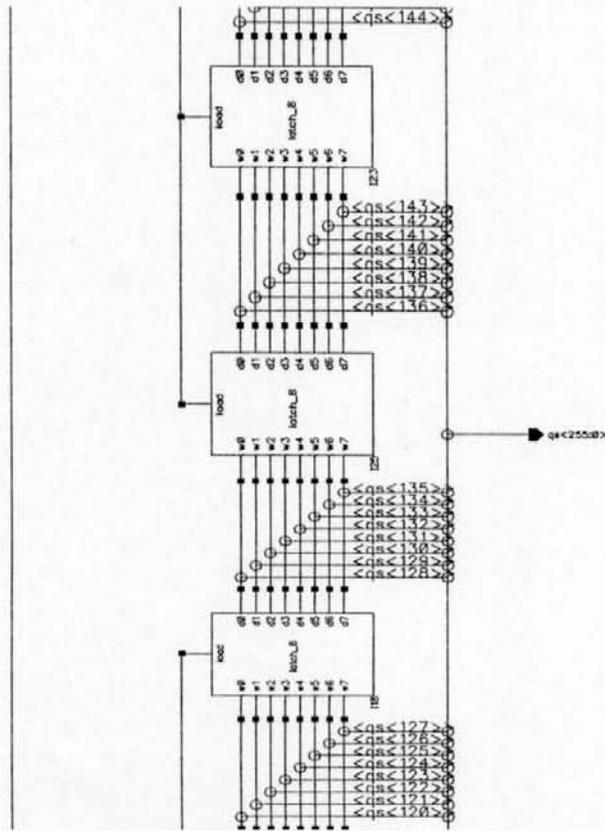


FIGURA A-5.7 – Módulo b-sp-a (7/12).

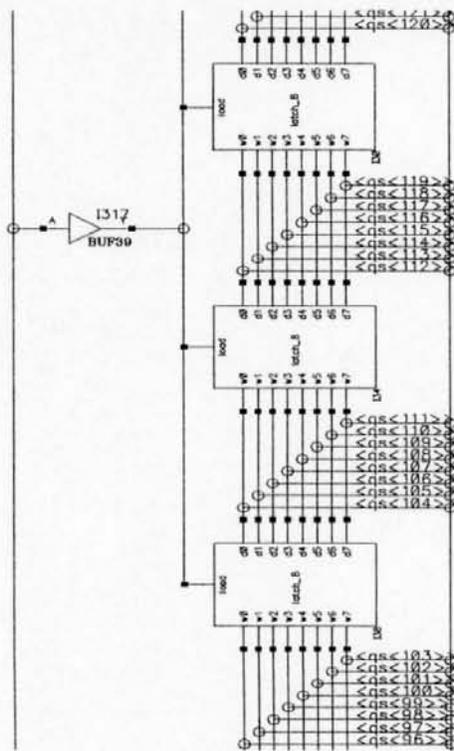


FIGURA A-5.8 – Módulo b-sp-a (8/12).

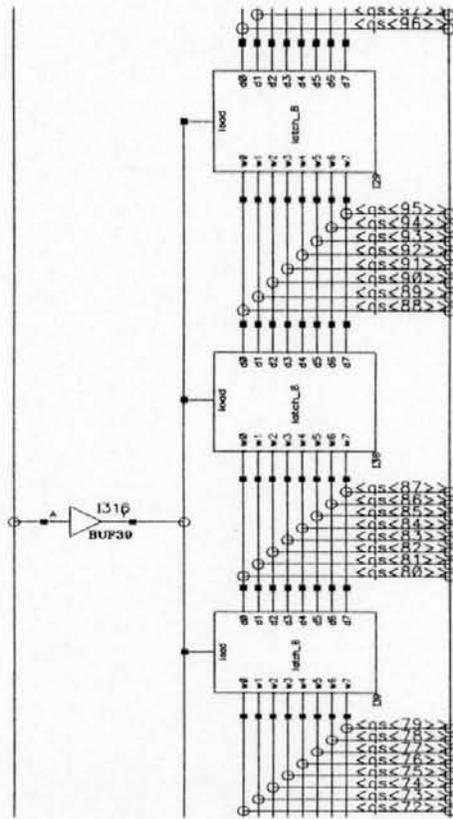


FIGURA A-5.9 – Módulo b-sp-a (9/12).

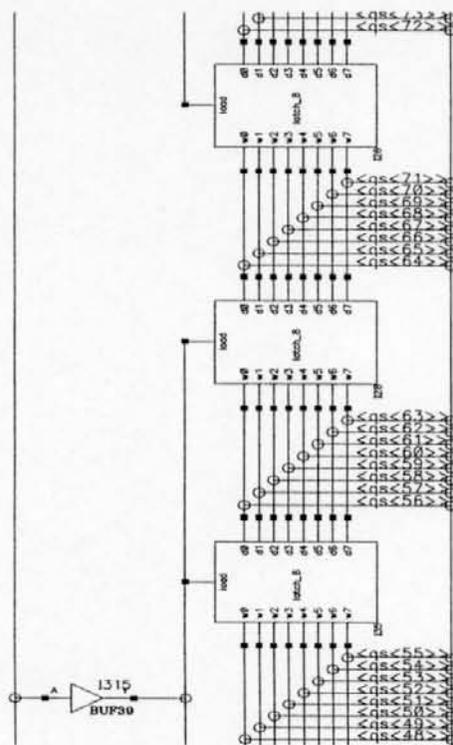


FIGURA A-5.10 – Módulo b-sp-a (10/12).

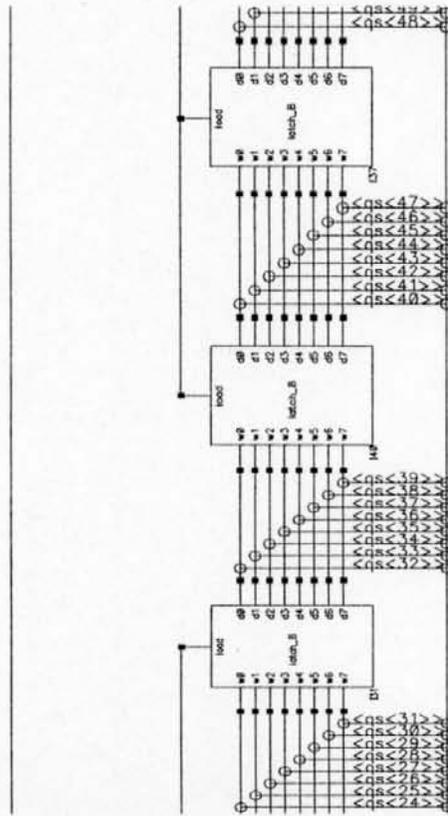


FIGURA A-5.11 – Módulo b-sp-a (11/12).

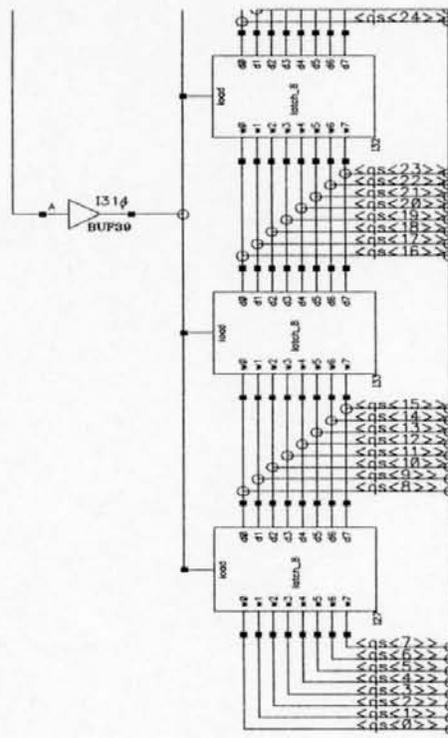


FIGURA A-5.12 – Módulo b-sp-a (12/12).

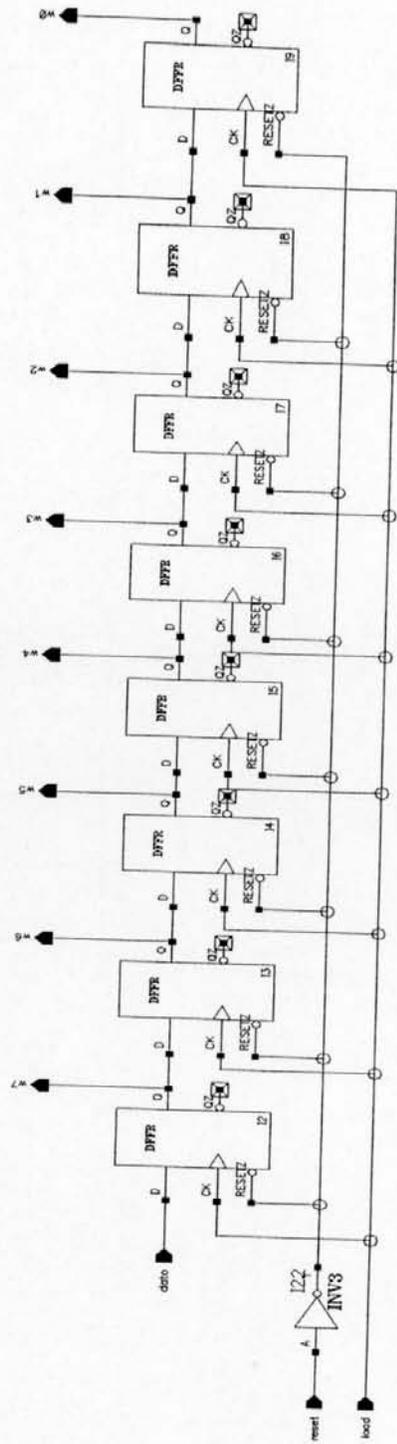


FIGURA A-5.13 - Módulo s-p.

**ANEXO A-6    SUBCIRCUITOS DO  
CONVERTOR PARALELO A  
SÉRIE**

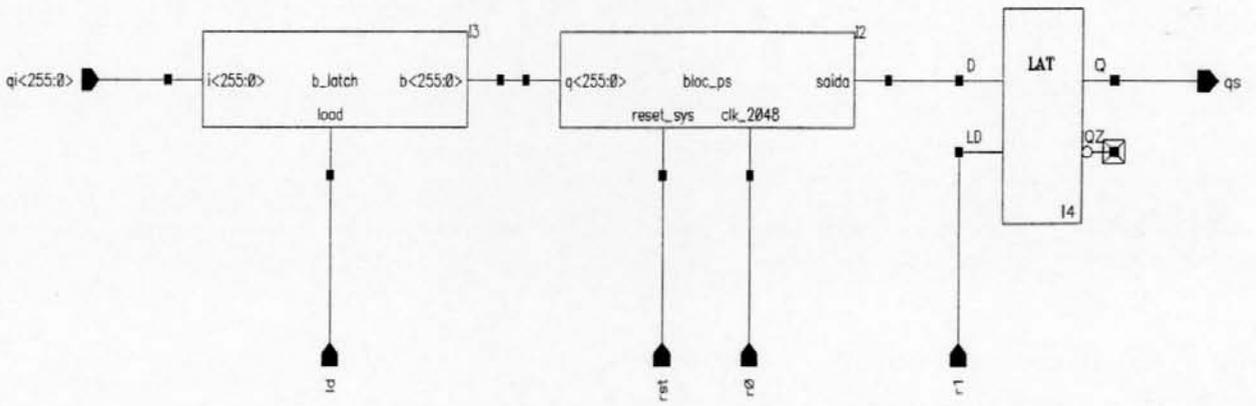


FIGURA A-6.1 – Módulo b-ps.

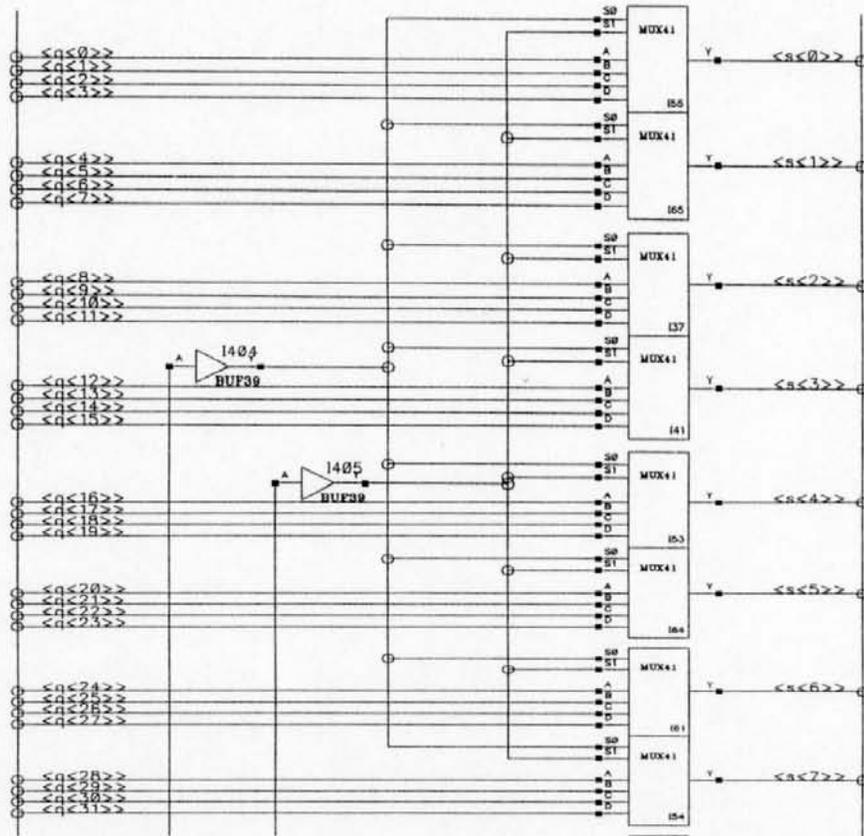


FIGURA A-6.2 – Módulo bloc-ps (etapa A, 1/8).

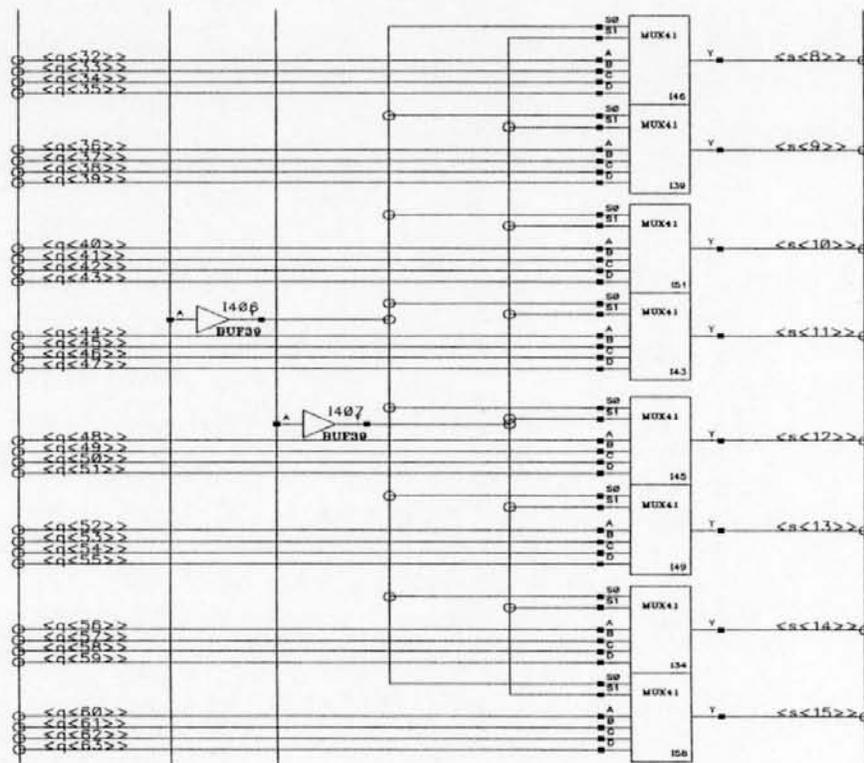


FIGURA A-6.3 – Módulo bloc-ps (etapa A, 2/8).

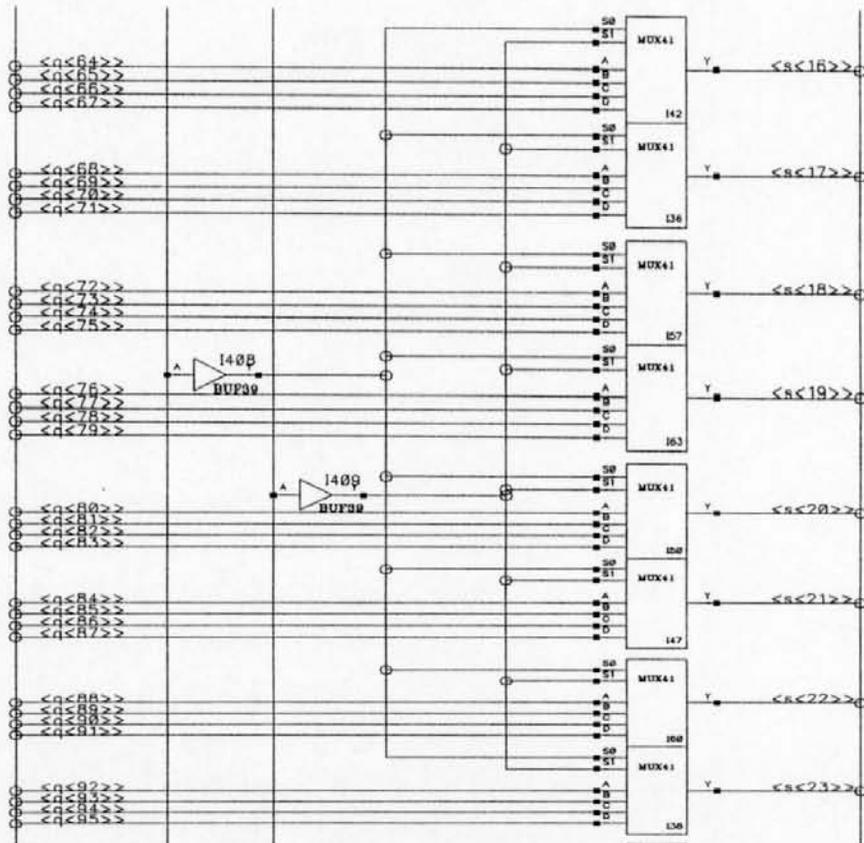


FIGURA A-6.4 – Módulo bloc-ps (etapa A, 3/8).

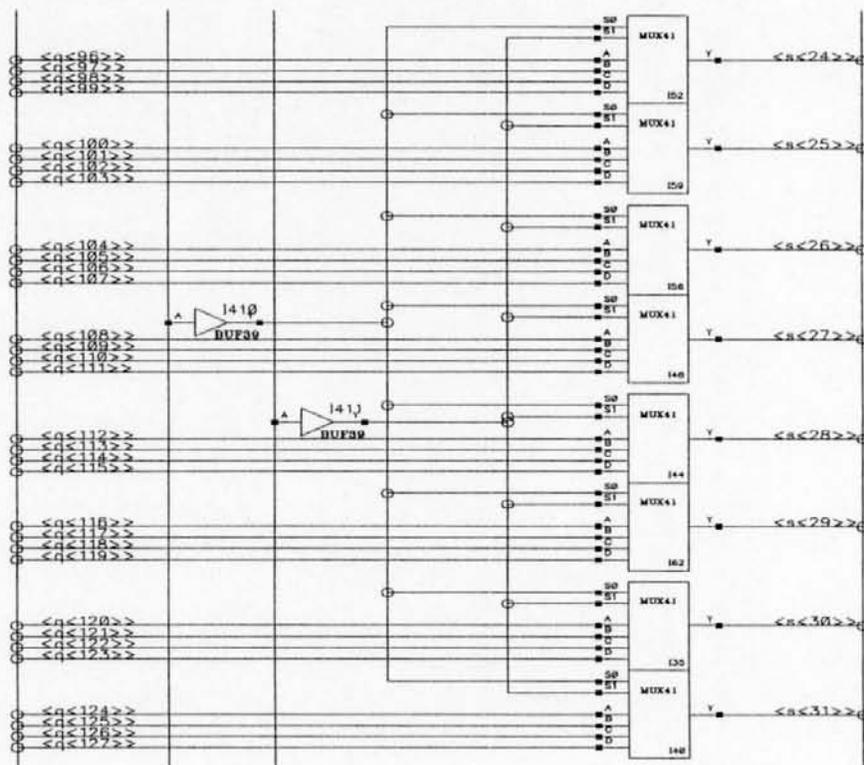


FIGURA A-6.5 – Módulo bloc-ps (etapa A, 4/8).

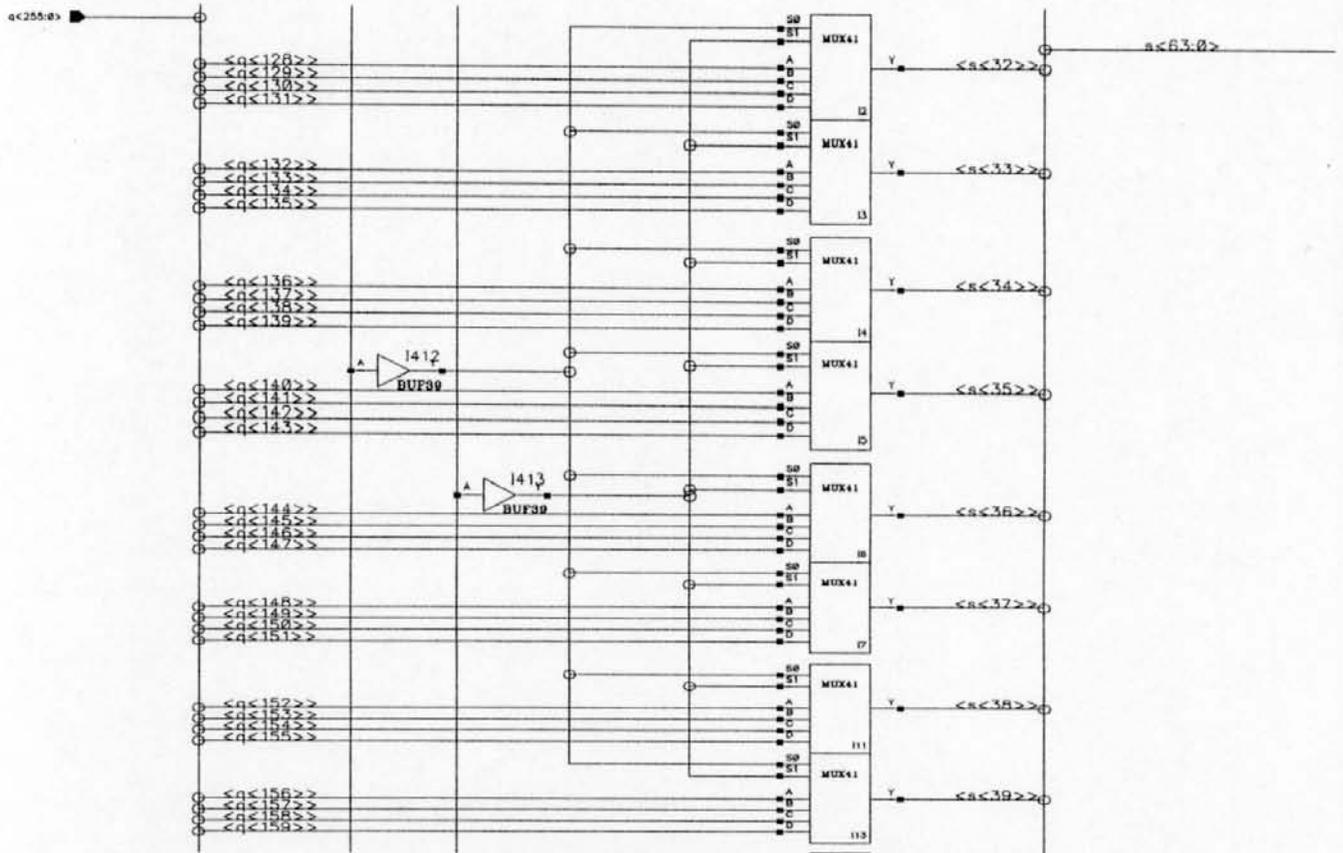


FIGURA A-6.6 – Módulo bloc-ps (etapa A, 5/8).

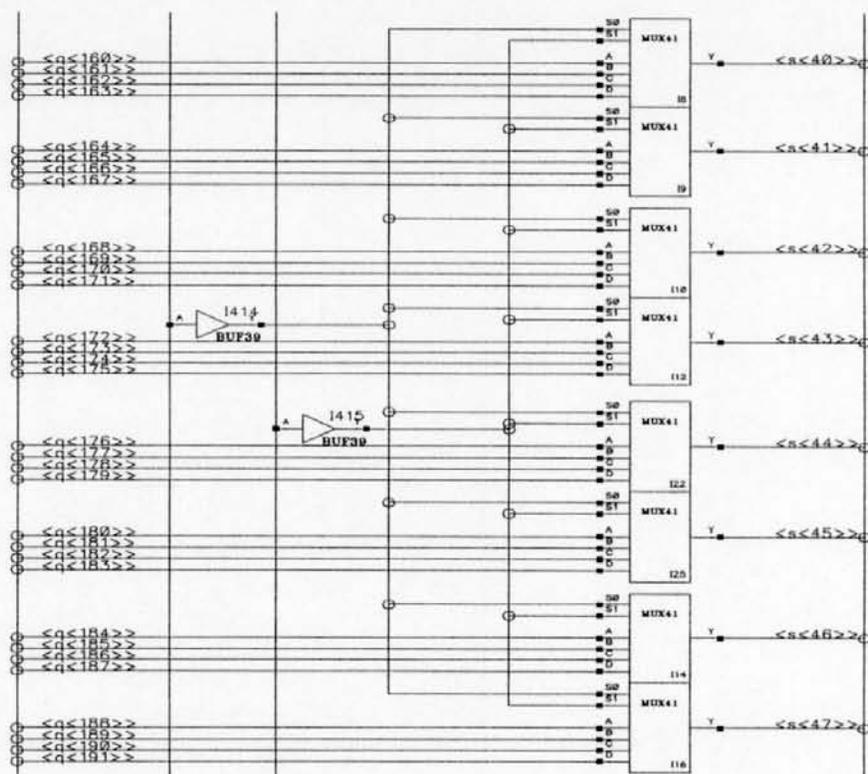


FIGURA A-6.7 – Módulo bloc-ps (etapa A, 6/8).

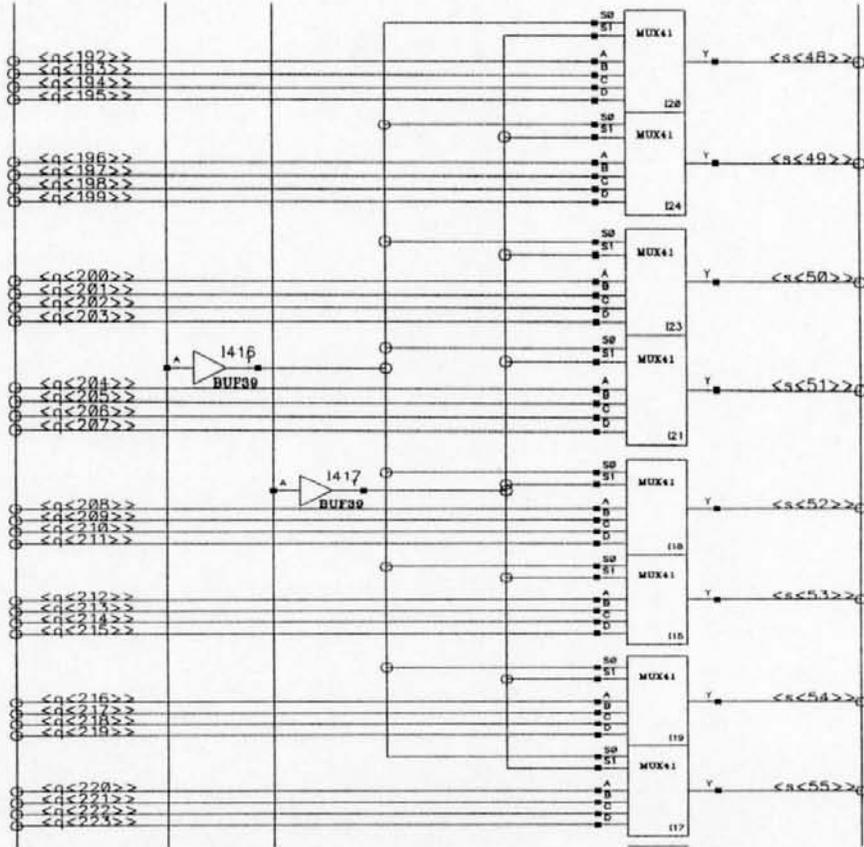


FIGURA A-6.8 – Módulo bloc-ps (etapa A, 7/8).

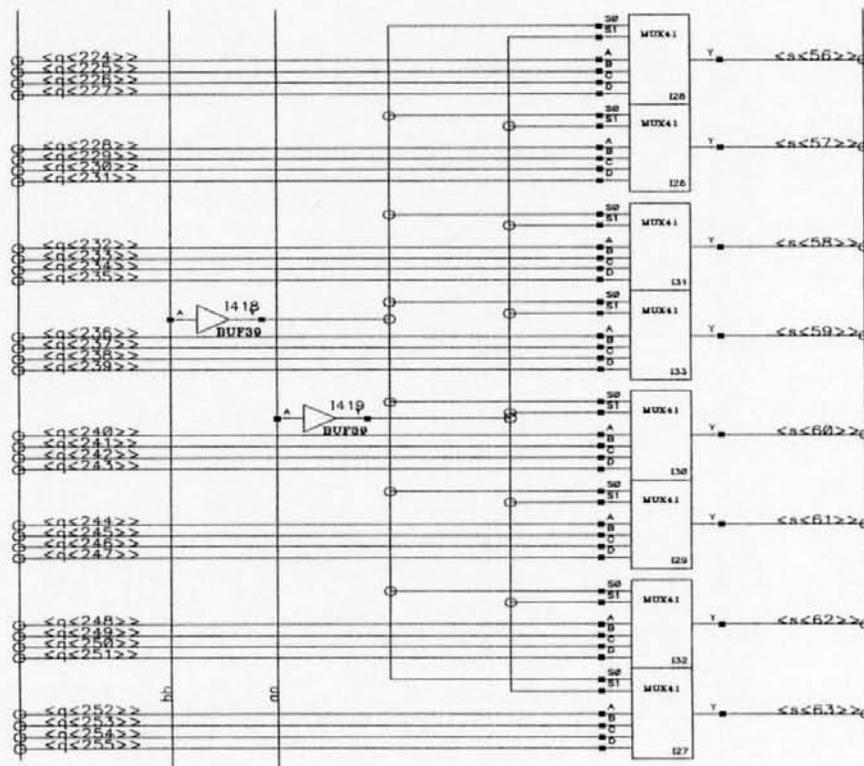


FIGURA A-6.9 – Módulo bloc-ps (etapa A, 8/8).



FIGURA A-6.10 - Módulo bloc-ps (etapa B, 1/3).

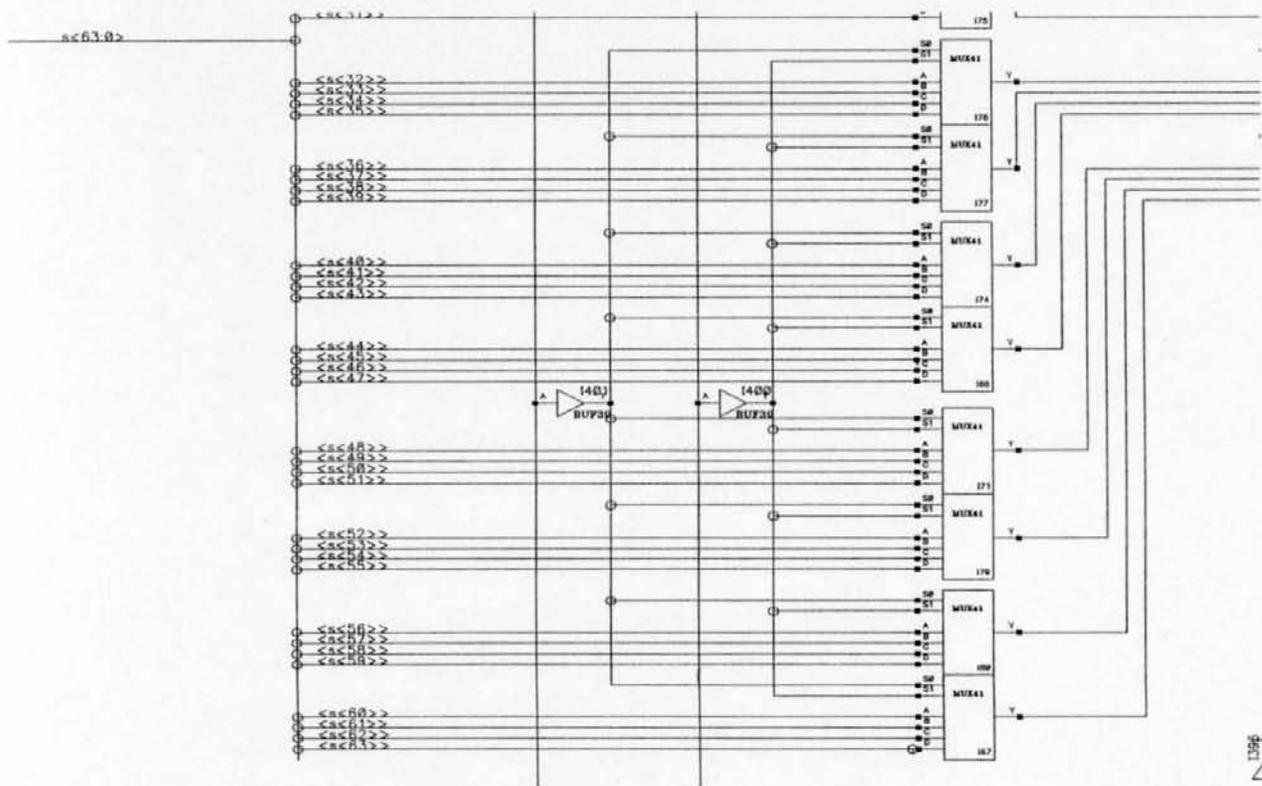


FIGURA A-6.11 - Módulo bloc-ps (etapa B, 2/3).

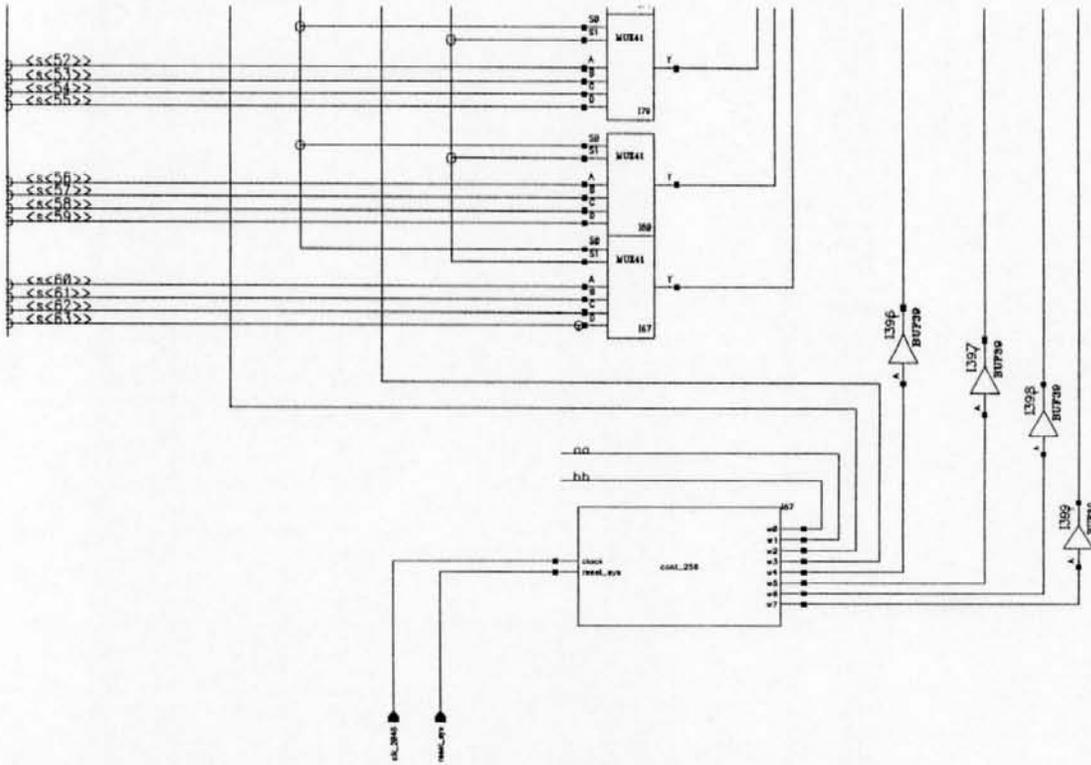


FIGURA A-6.12 – Módulo bloc-ps (etapa B, 3/3).

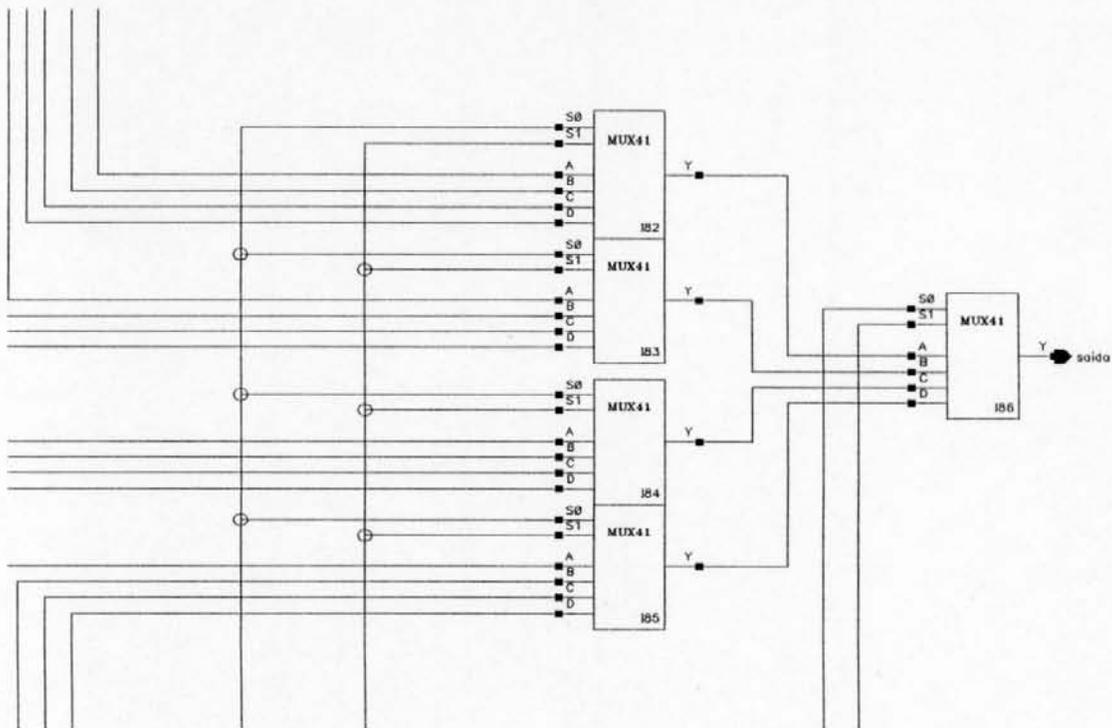


FIGURA A-6.13 – Módulo bloc-ps (etapa C).

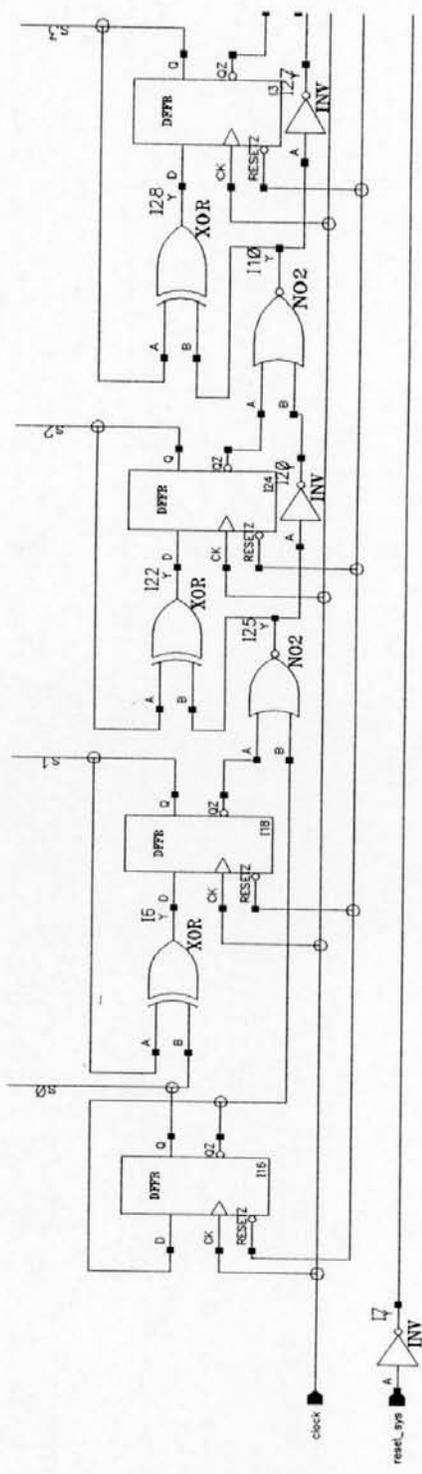
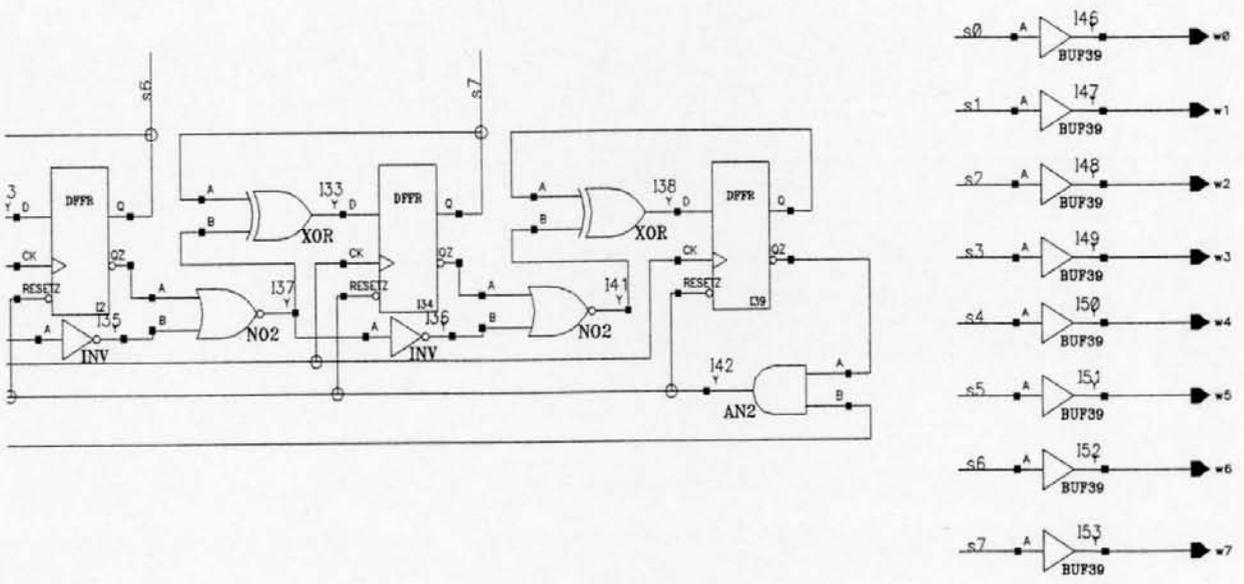


FIGURA A-6.14 – Módulo cont-256 (1/3).



FIGURA A-6.16 – Módulo cont-256 (3/3).



**ANEXO A-7    SUBCIRCUITOS AUXILIARES**  
**I**

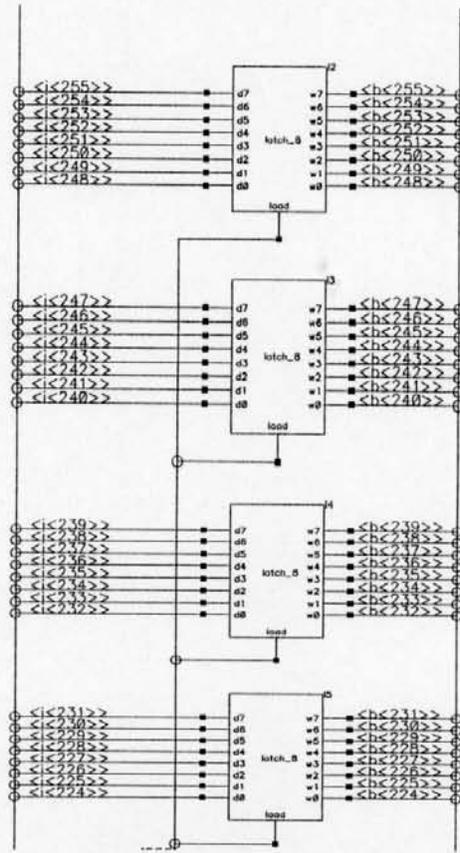


FIGURA A-7.1 – Módulo b-latch (1/8).

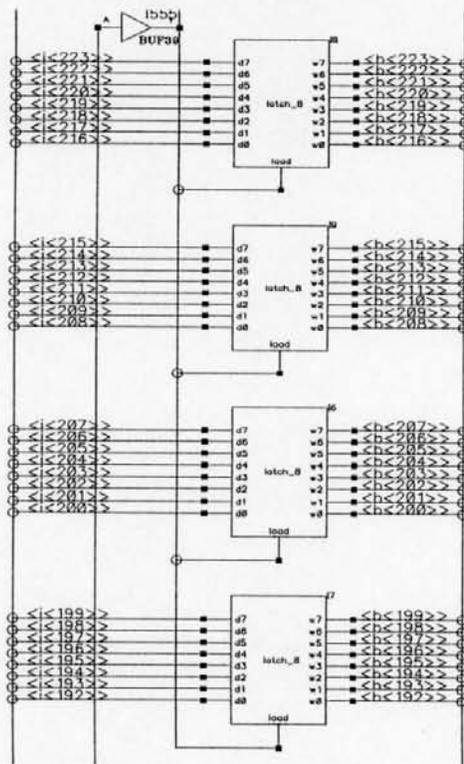


FIGURA A-7.2 – Módulo b-latch (2/8).

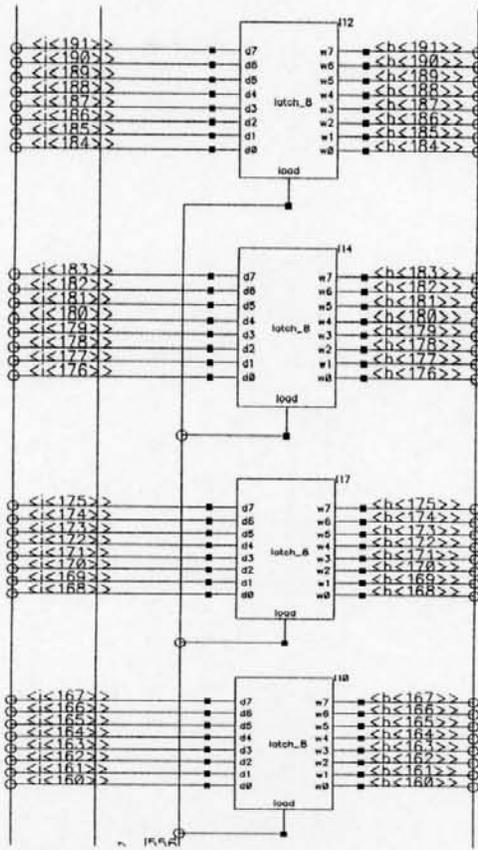


FIGURA A-7.3 – Módulo b-latch (3/8).

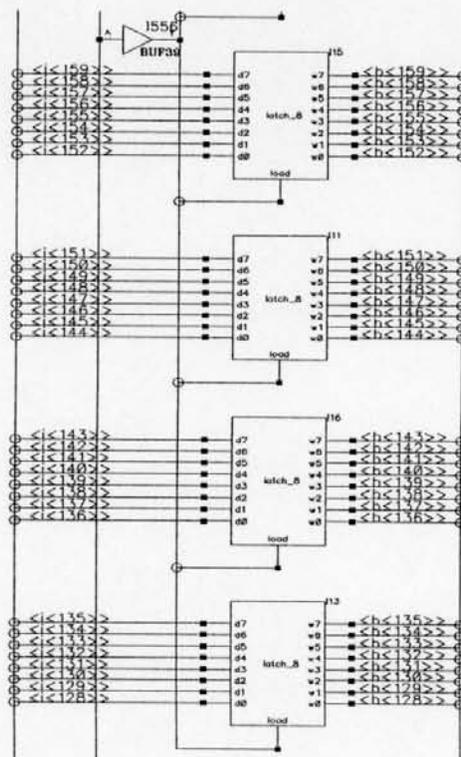


FIGURA A-7.4 – Módulo b-latch (4/8).

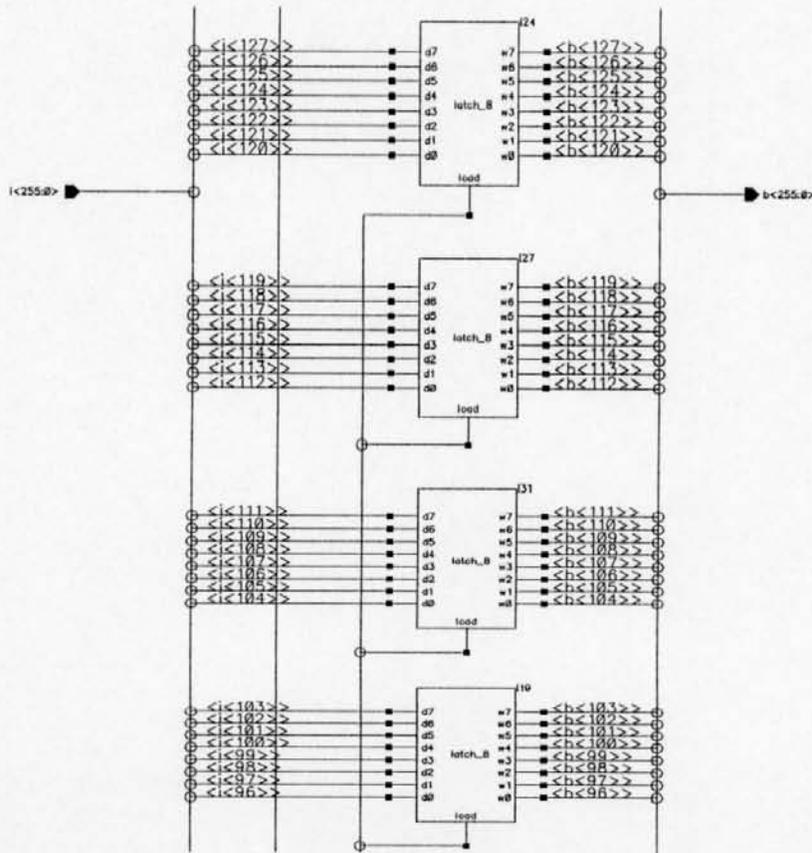


FIGURA A-7.5 – Módulo b-latch (5/8).

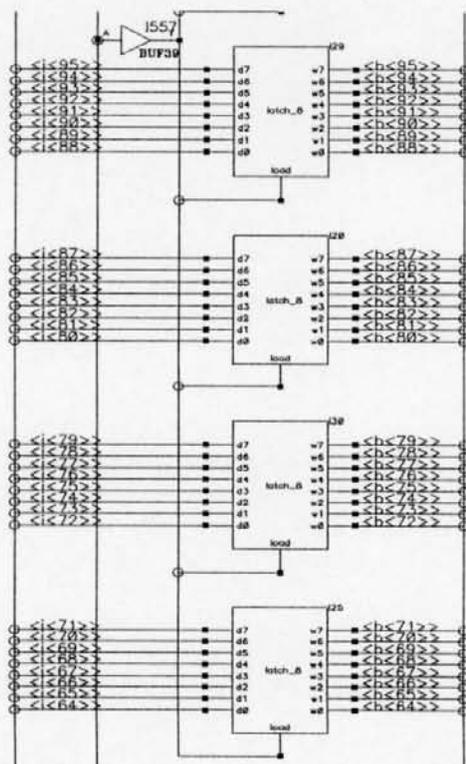


FIGURA A-7.6 – Módulo b-latch (6/8).

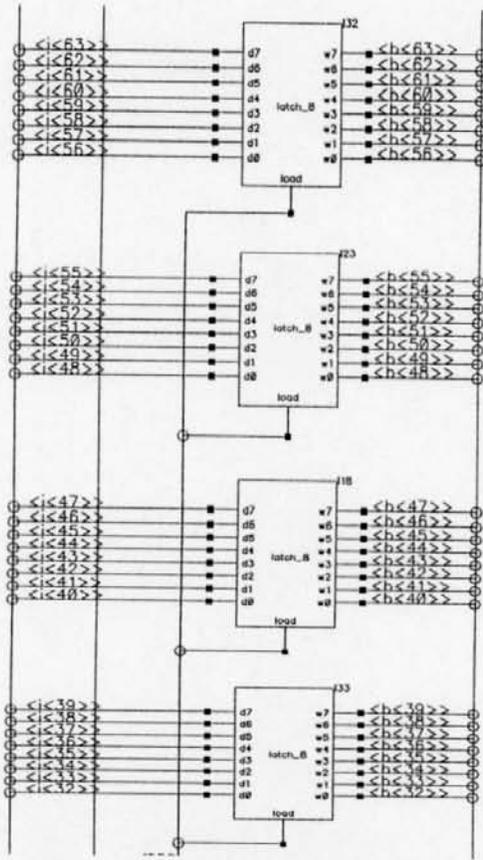


FIGURA A-7.7 – Módulo b-latch (7/8).

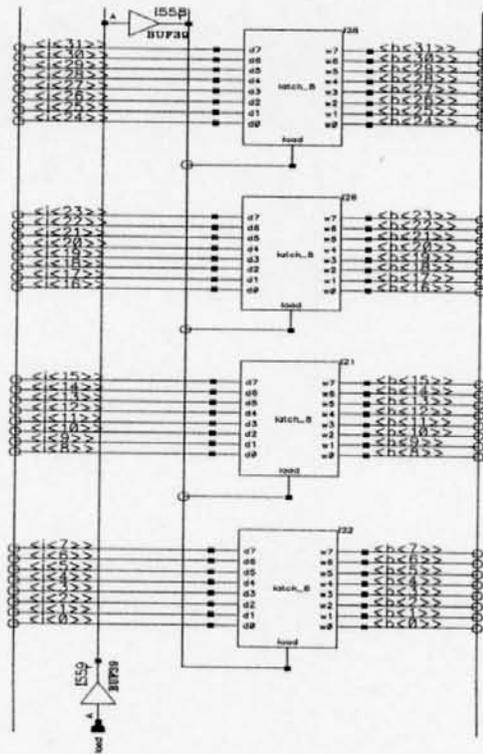


FIGURA A-7.8 – Módulo b-latch (8/8).

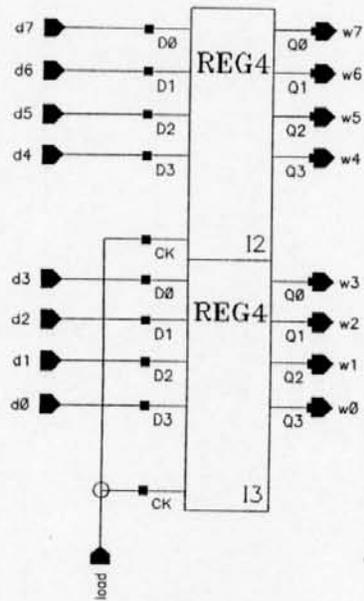


FIGURA A-7.9 – Módulo latch-8.

**ANEXO A-8    SUBCIRCUITOS DA  
MEMÓRIA SFM**

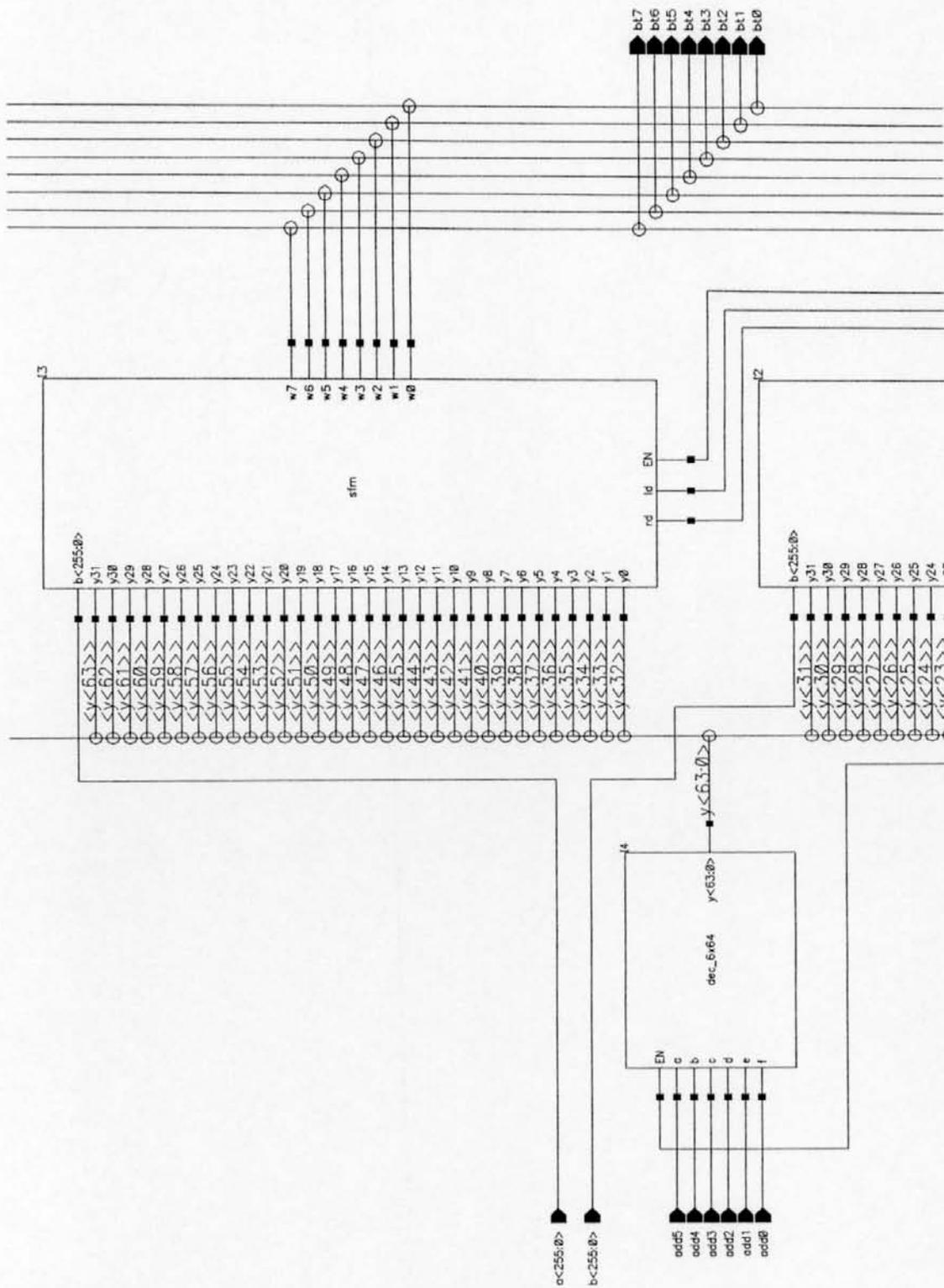


FIGURA A-8.1 - Módulo b-sfm (1/2).

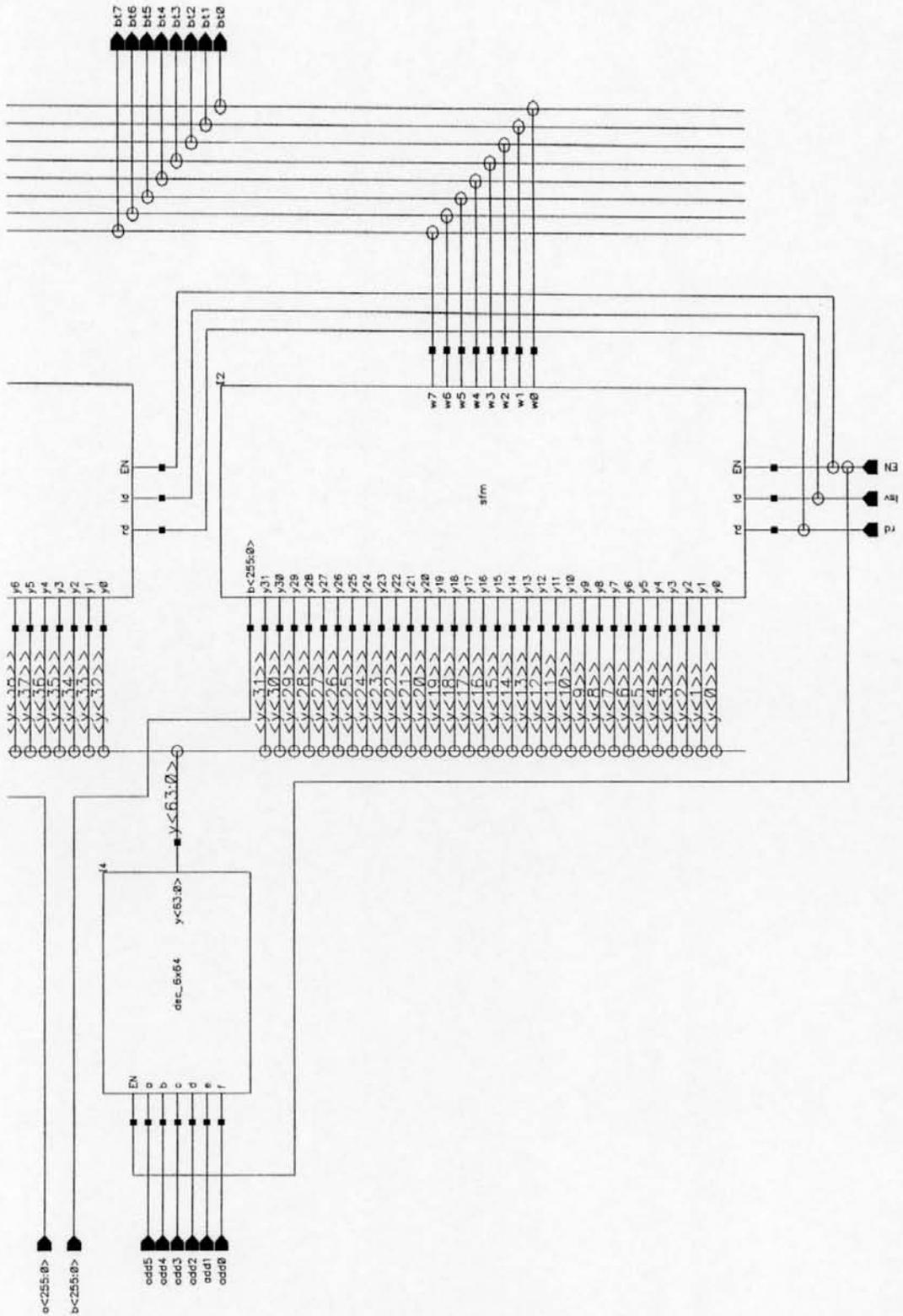


FIGURA A-8.2 – Módulo b-sfm (2/2).

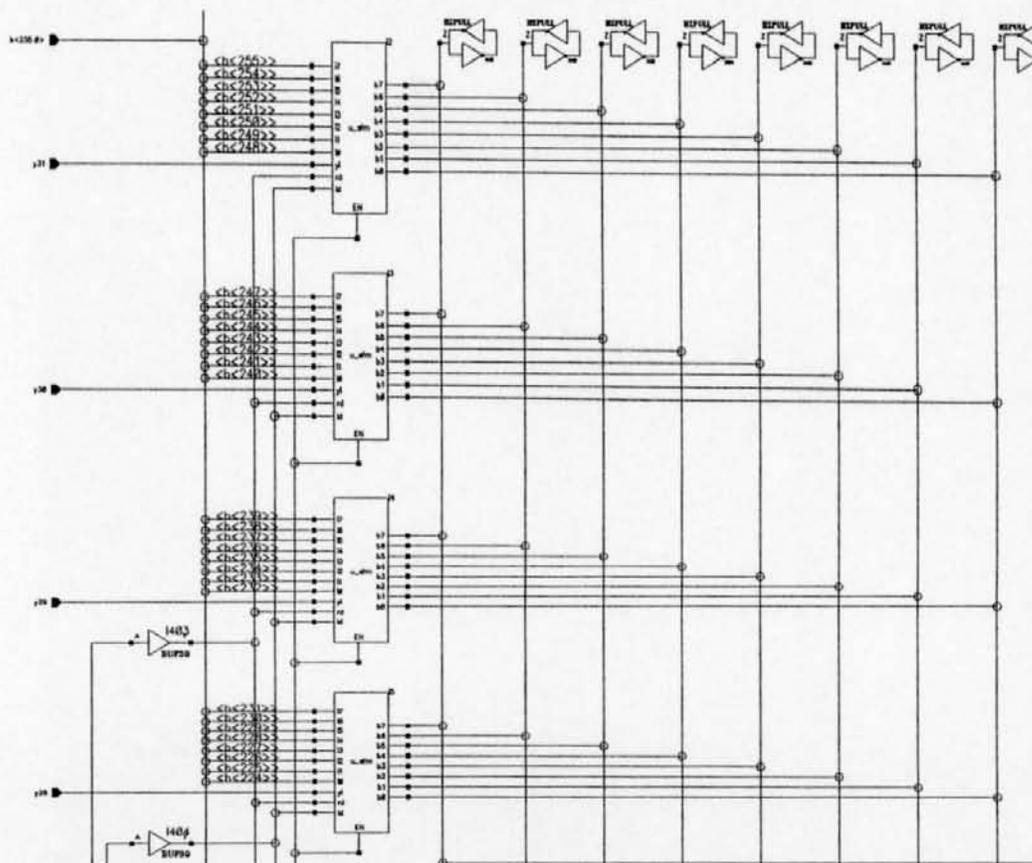


FIGURA A-8.3 - Módulo sfm (1/8).

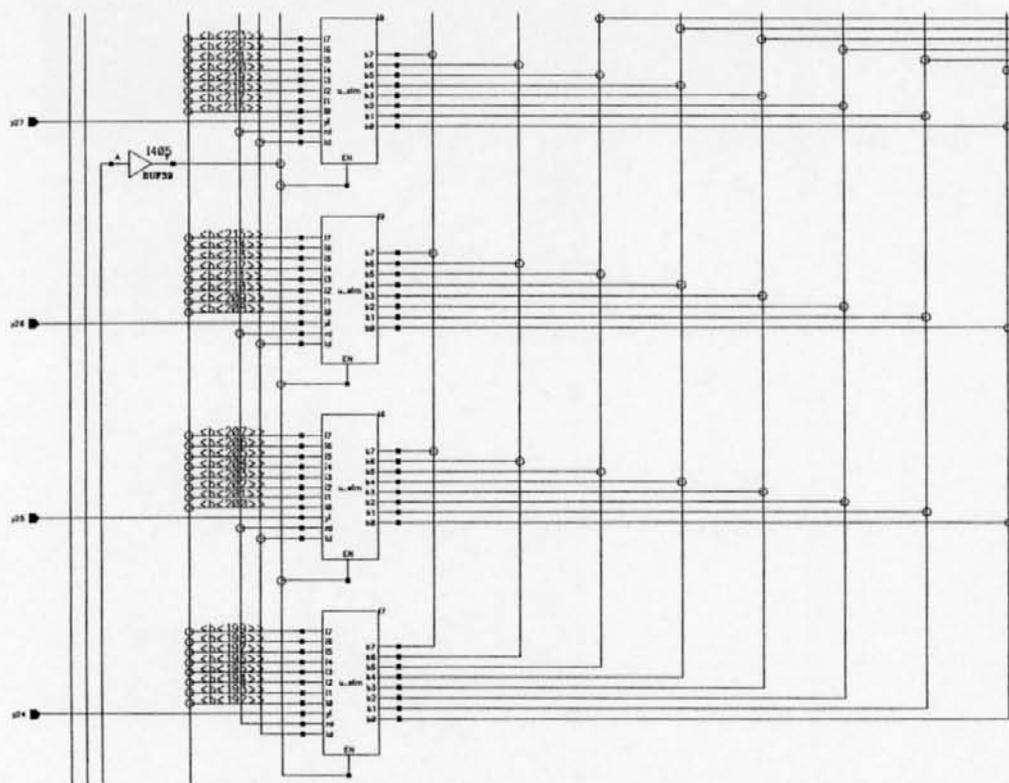


FIGURA A-8.4 - Módulo sfm (2/8).

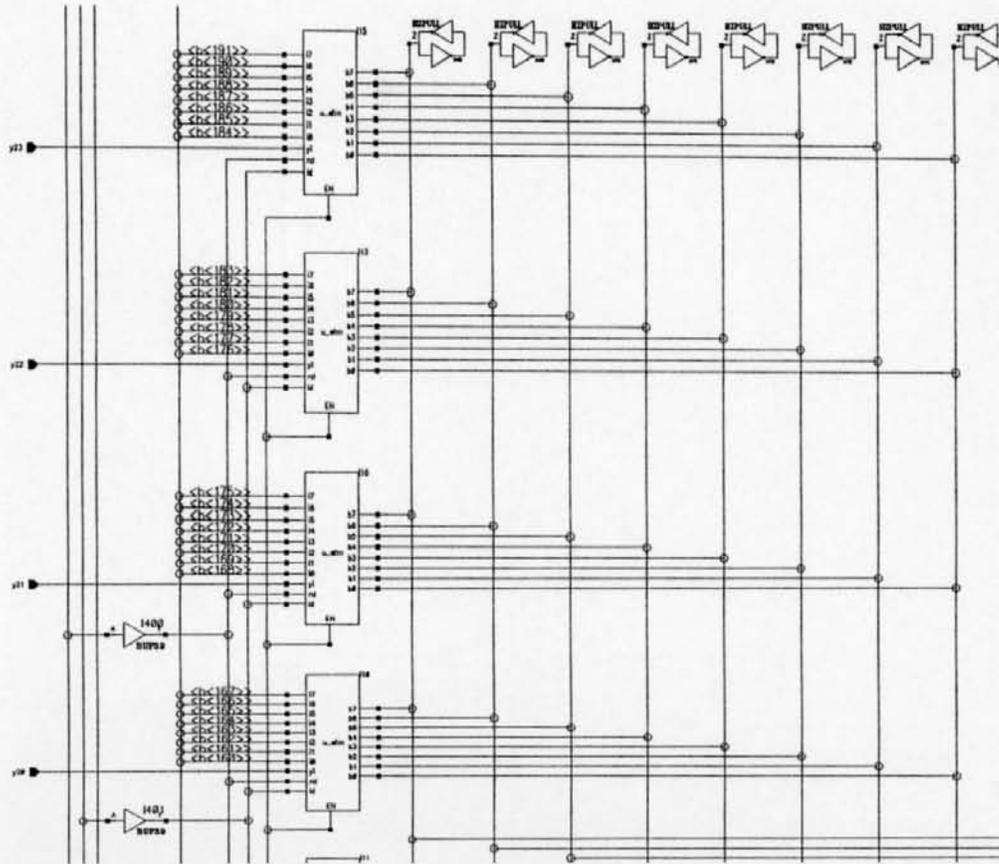


FIGURA A-8.5 - Módulo sfm (3/8).

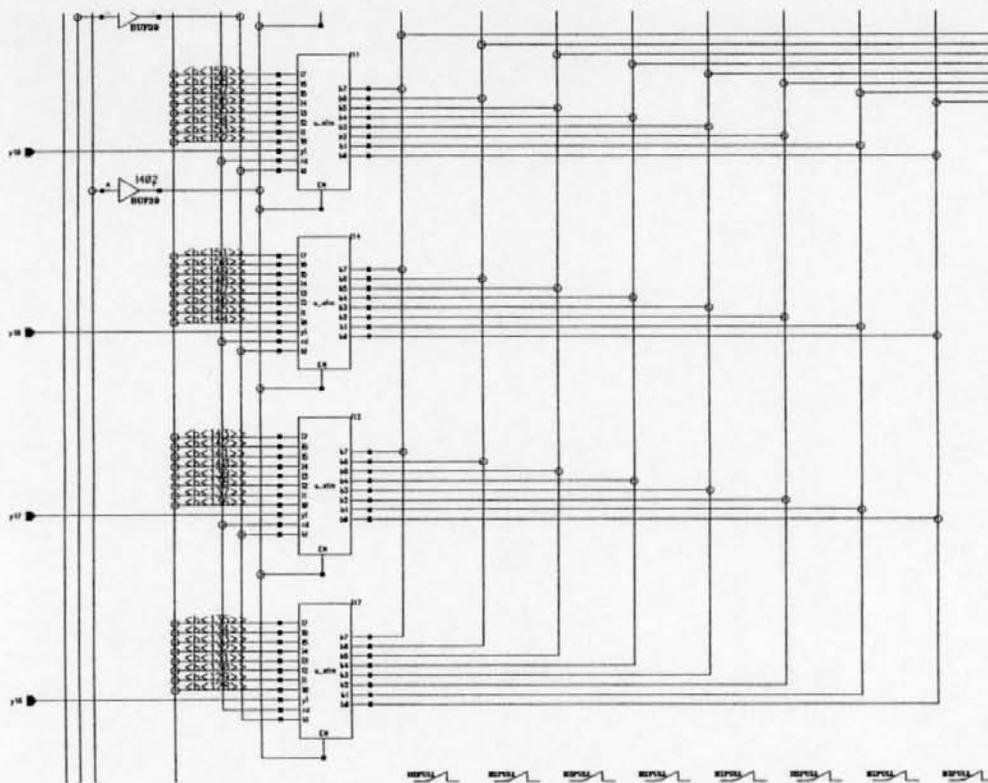


FIGURA A-8.6 - Módulo sfm (4/8).

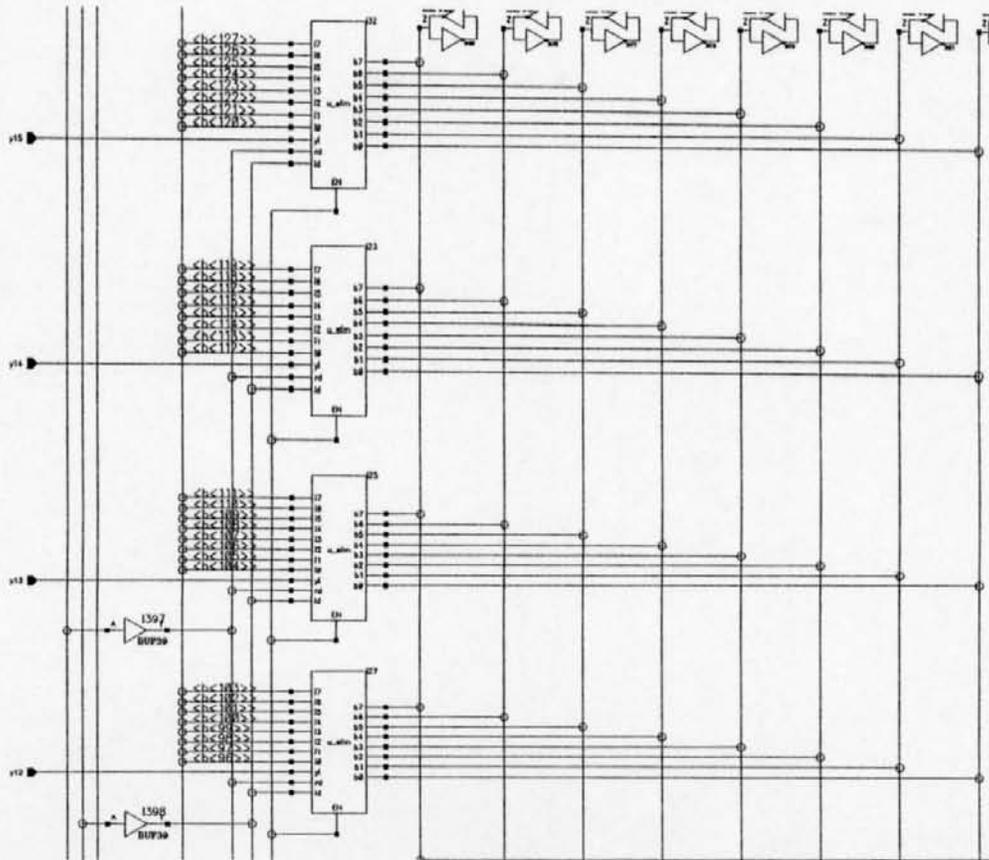


FIGURA A-8.7 – Módulo sfm (5/8).

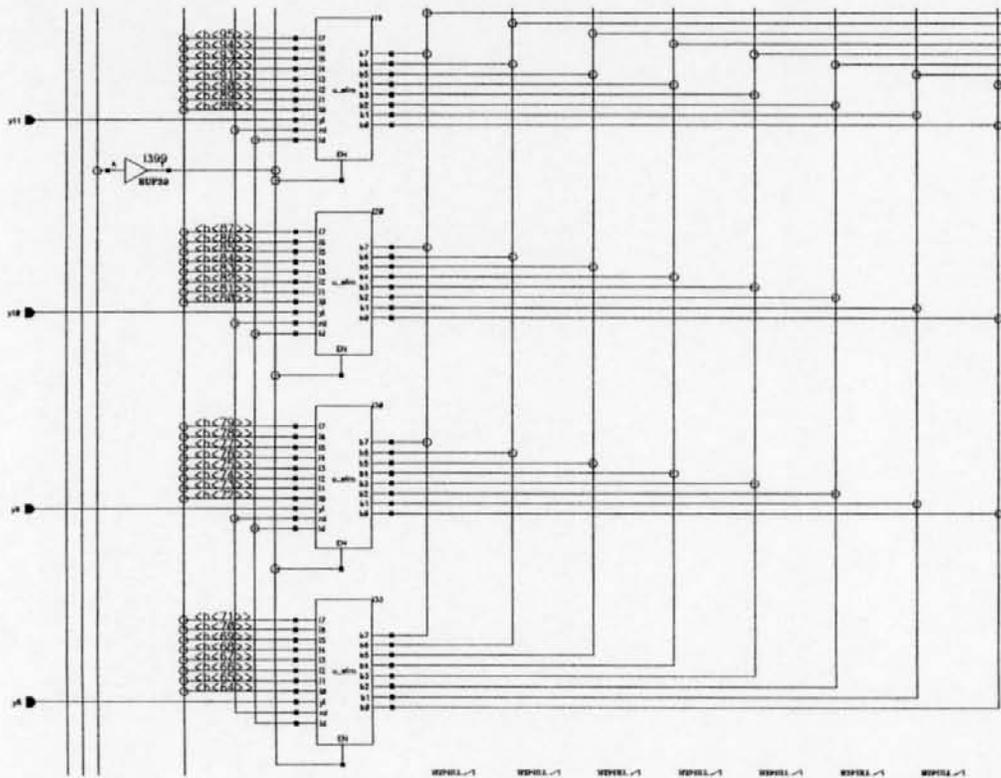


FIGURA A-8.8 – Módulo sfm (6/8).

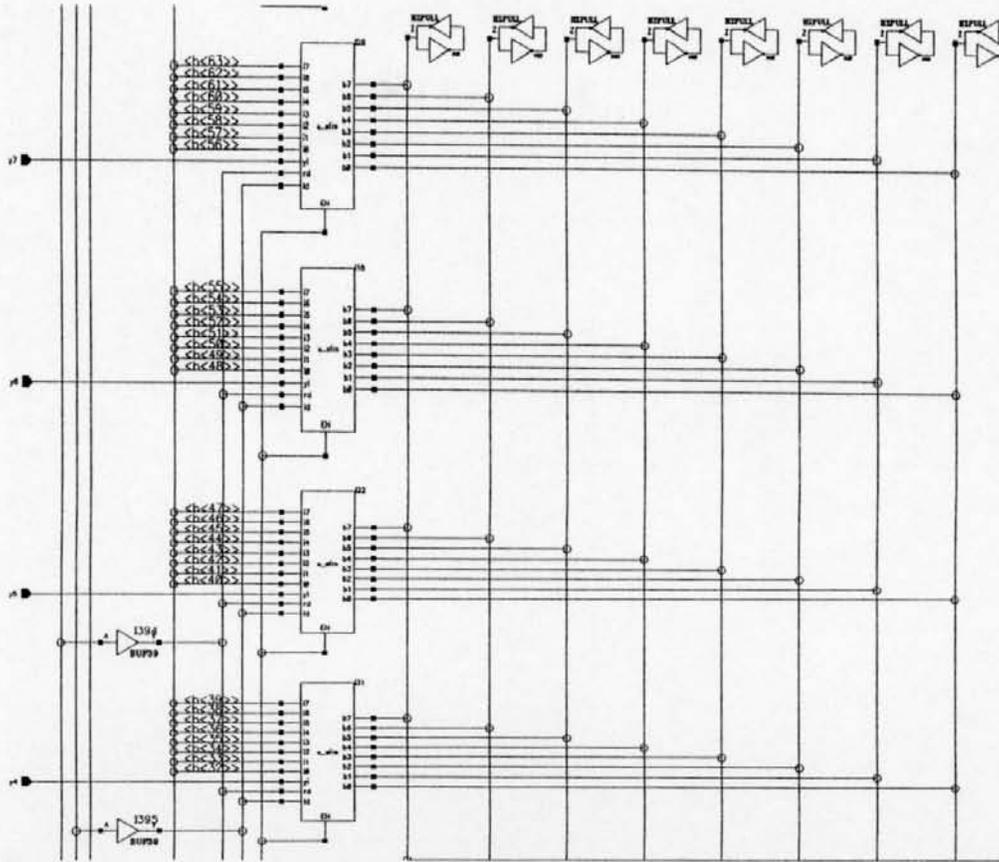


FIGURA A-8.9 – Módulo sfm (7/8).

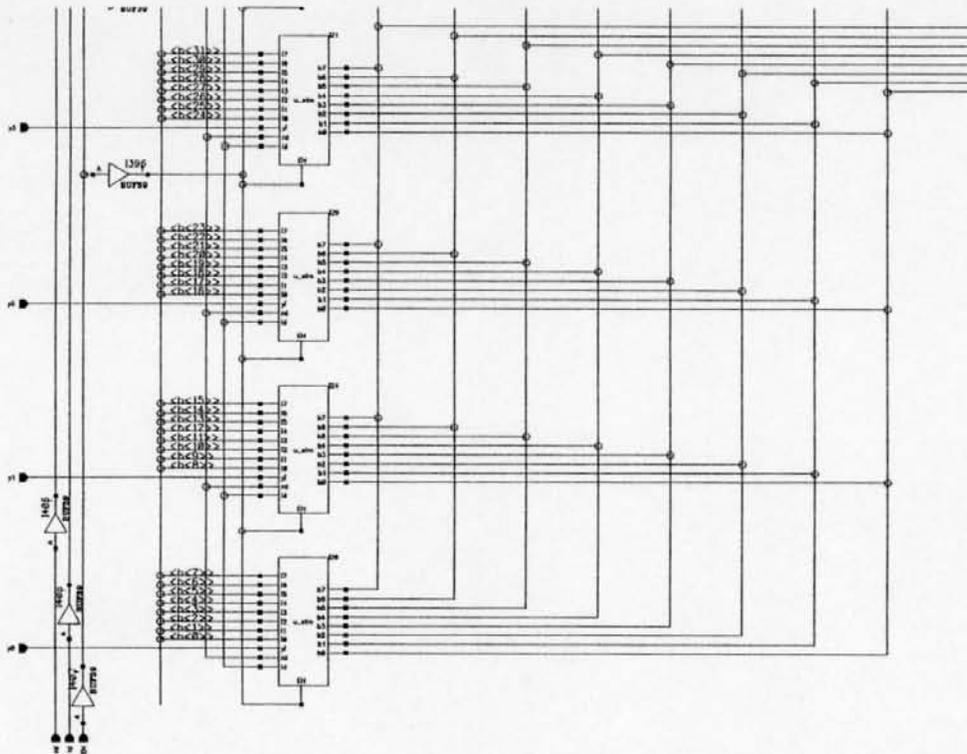
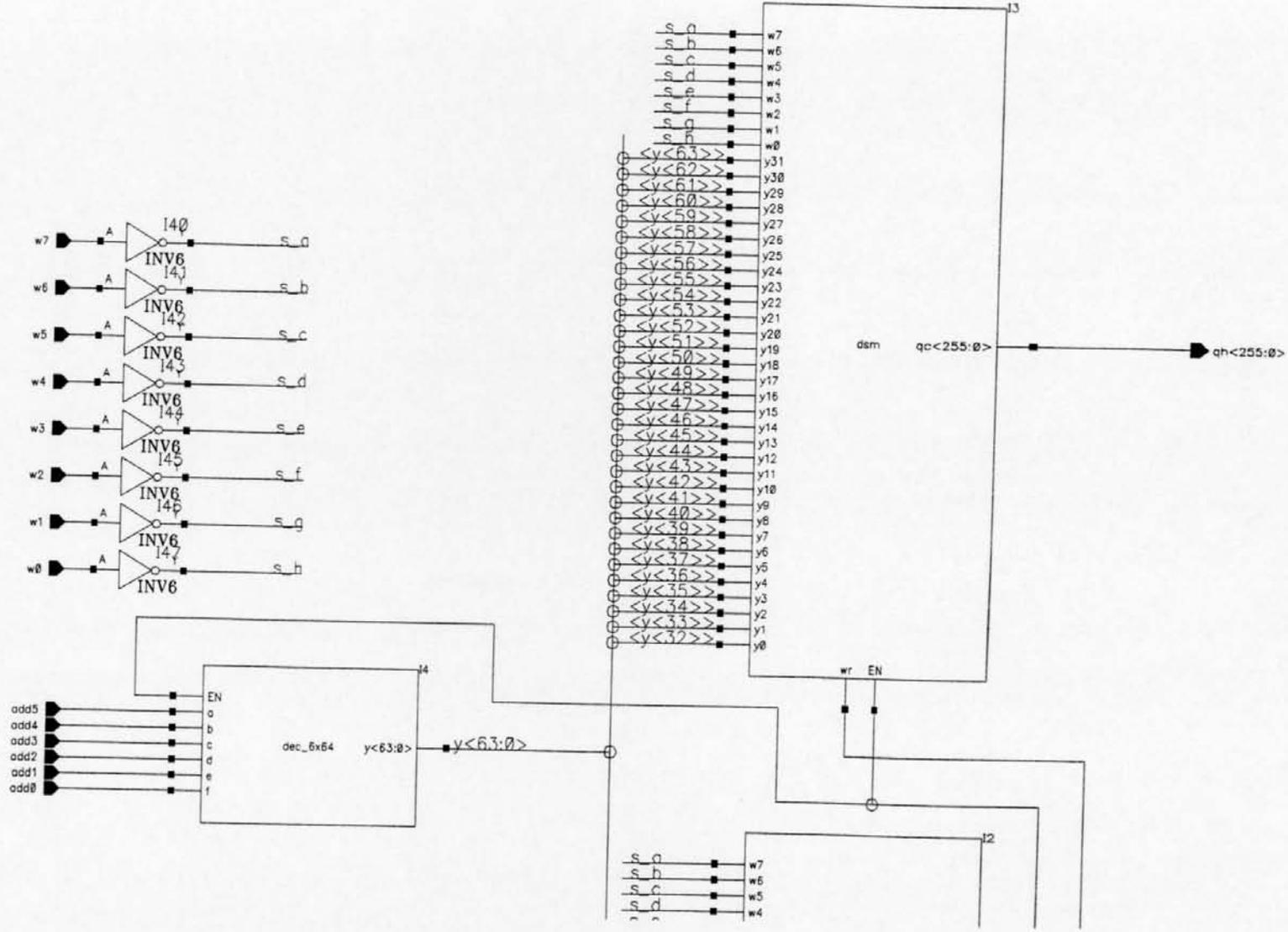


FIGURA A-8.10 – Módulo sfm (8/8).

**ANEXO A-9    SUBCIRCUITOS DA  
MEMÓRIA DSM**

FIGURA A-9.1 – Módulo b-dsm (1/2).



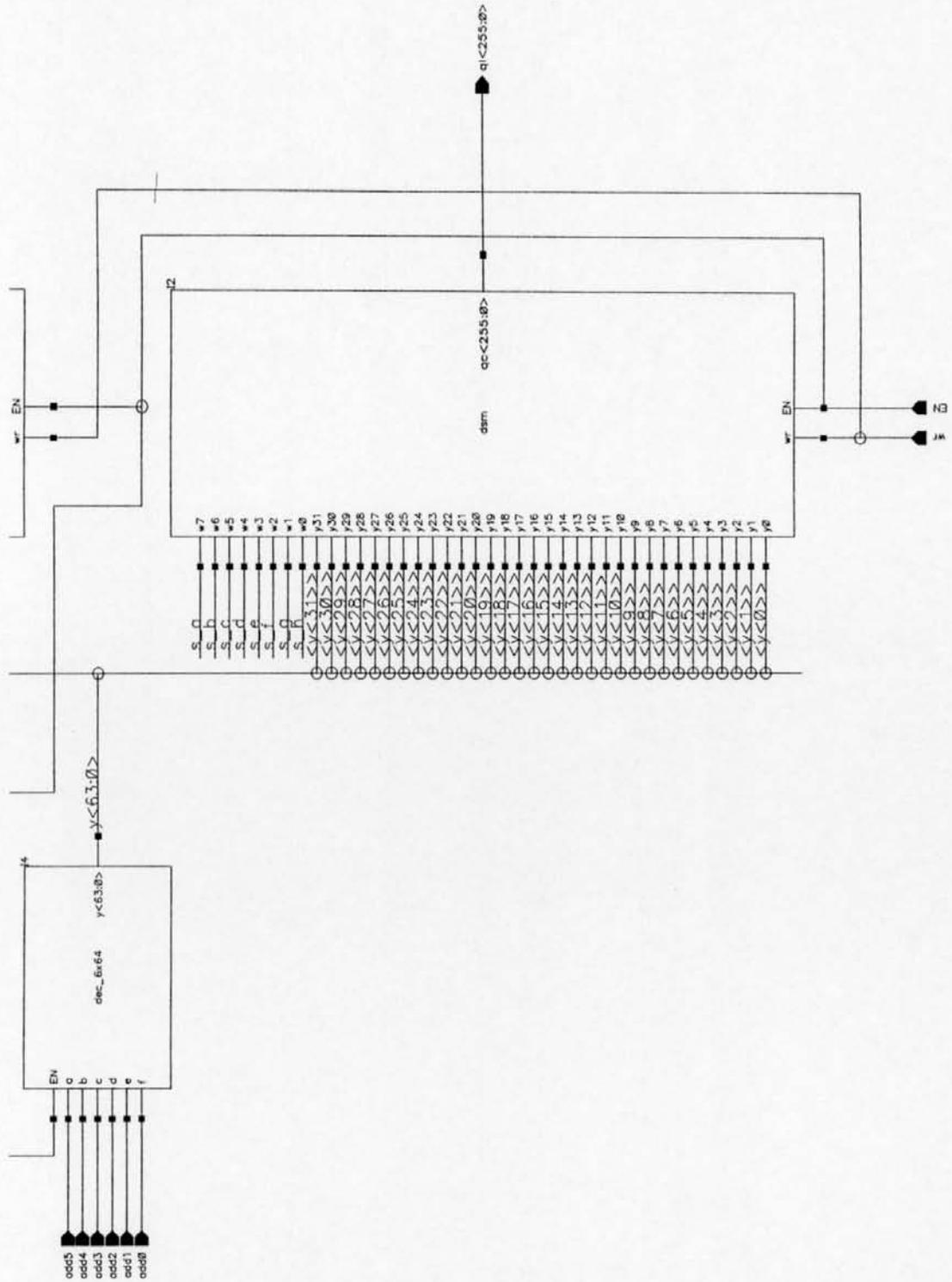


FIGURA A-9.2 – Módulo b-dsm (2/2).

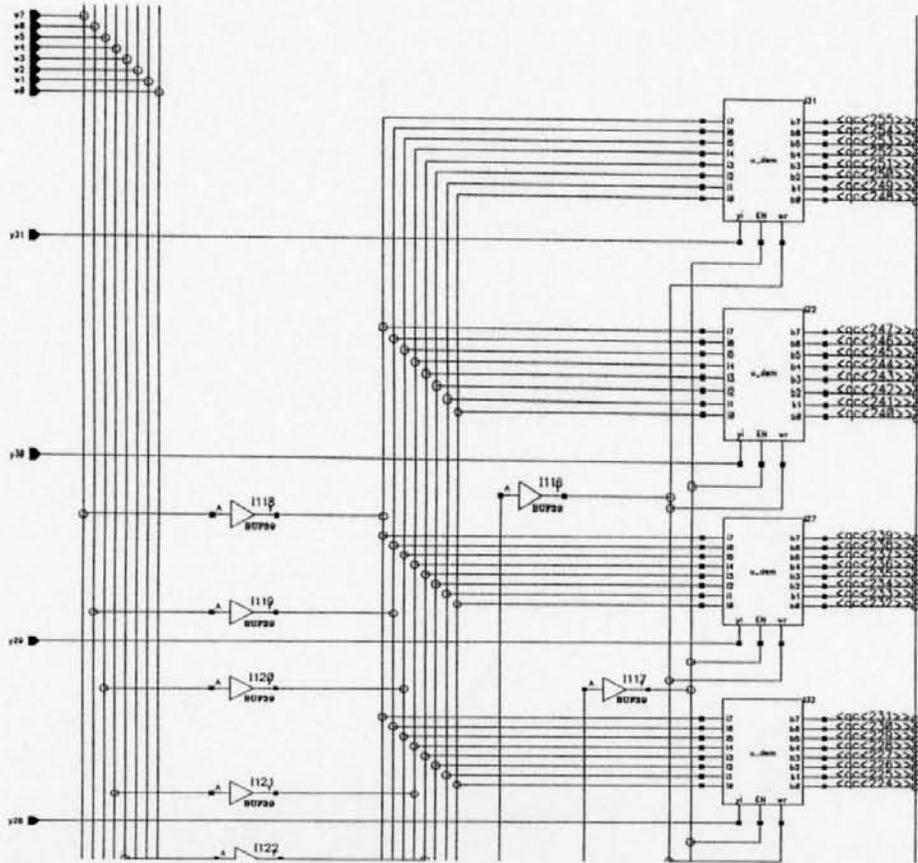


FIGURA A-9.3 – Módulo dsm (1/8).

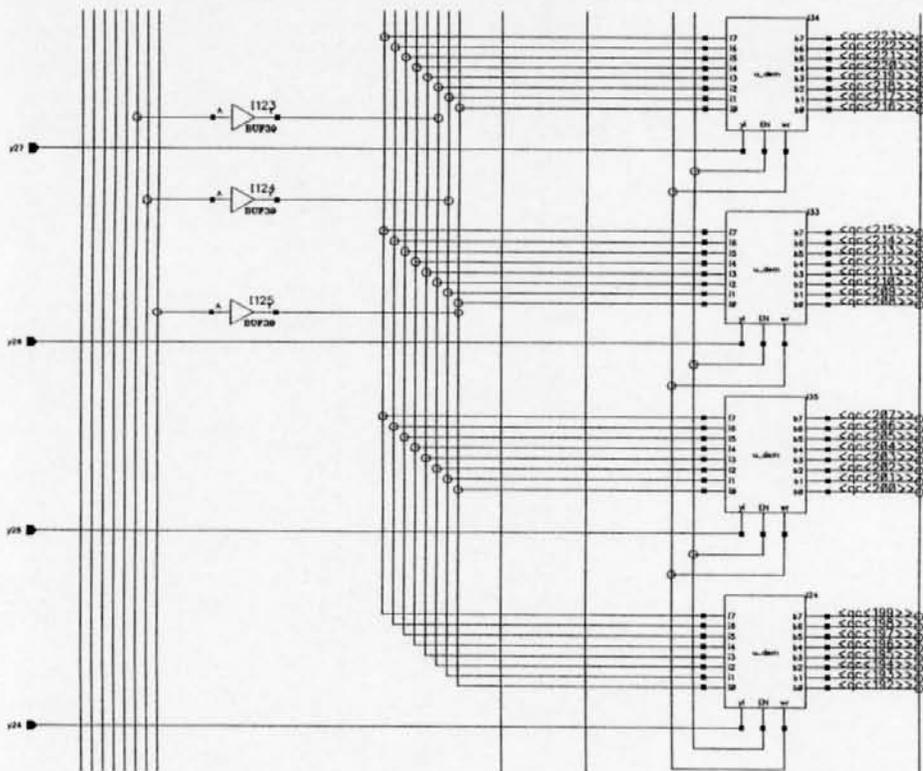


FIGURA A-9.4 – Módulo dsm (2/8).

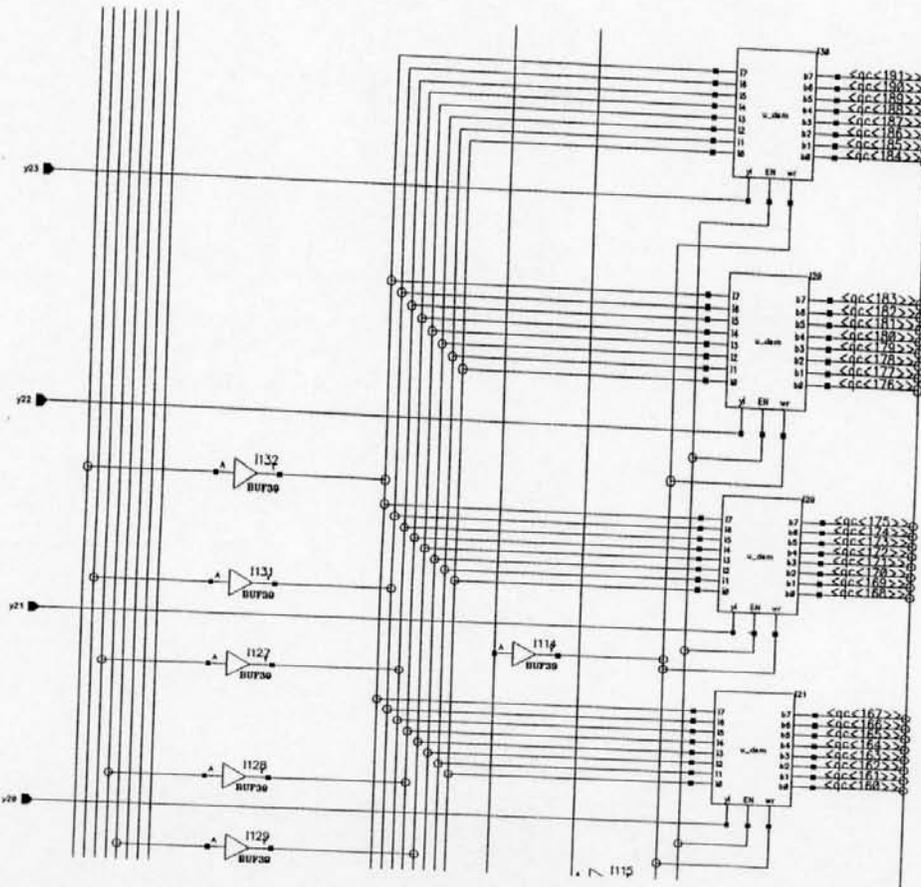


FIGURA A-9.5 – Módulo dsm (3/8).

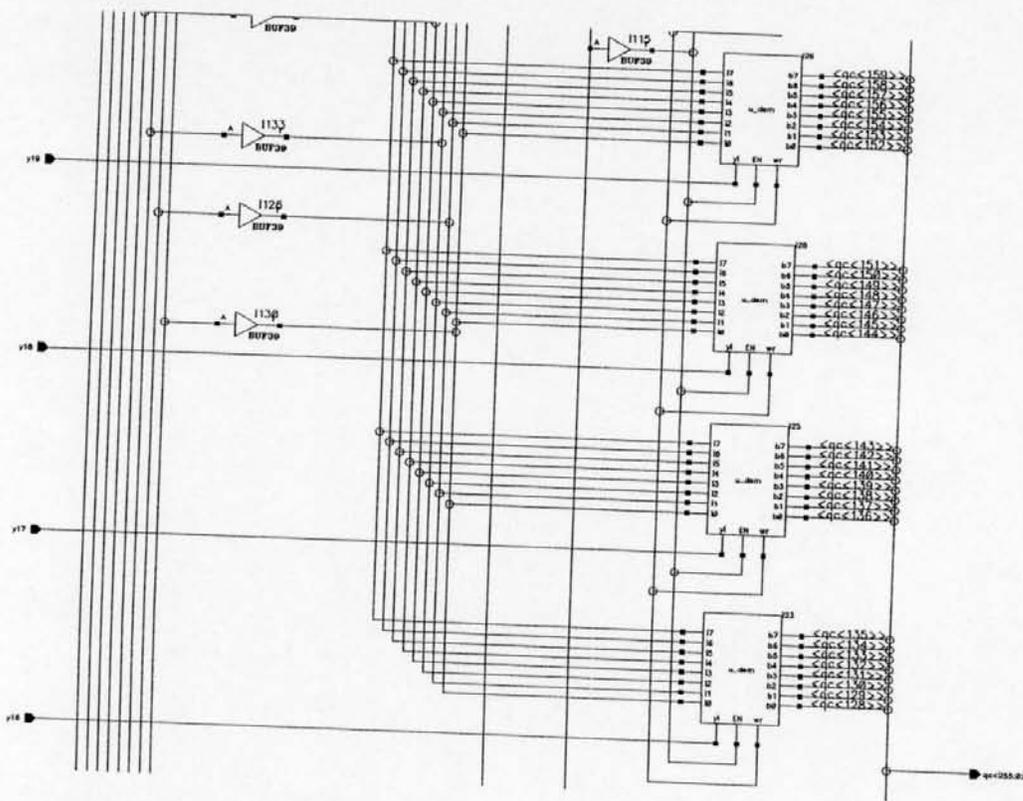


FIGURA A-9.6 – Módulo dsm (4/8).

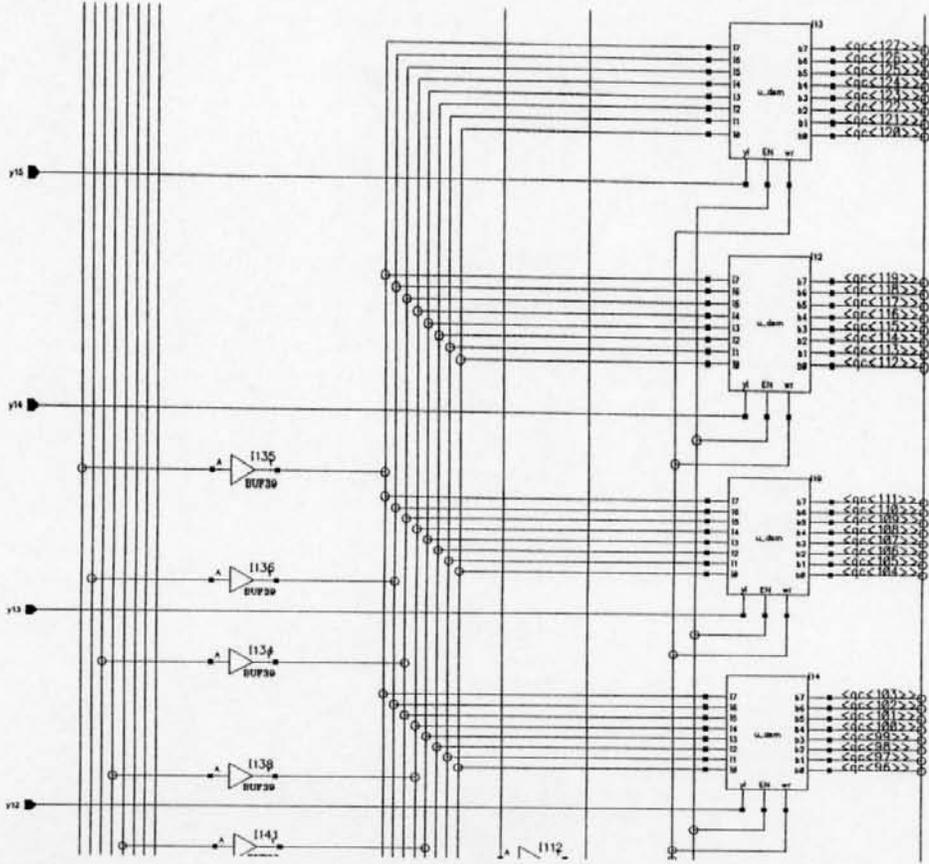


FIGURA A-9.7 – Módulo dsm (5/8).

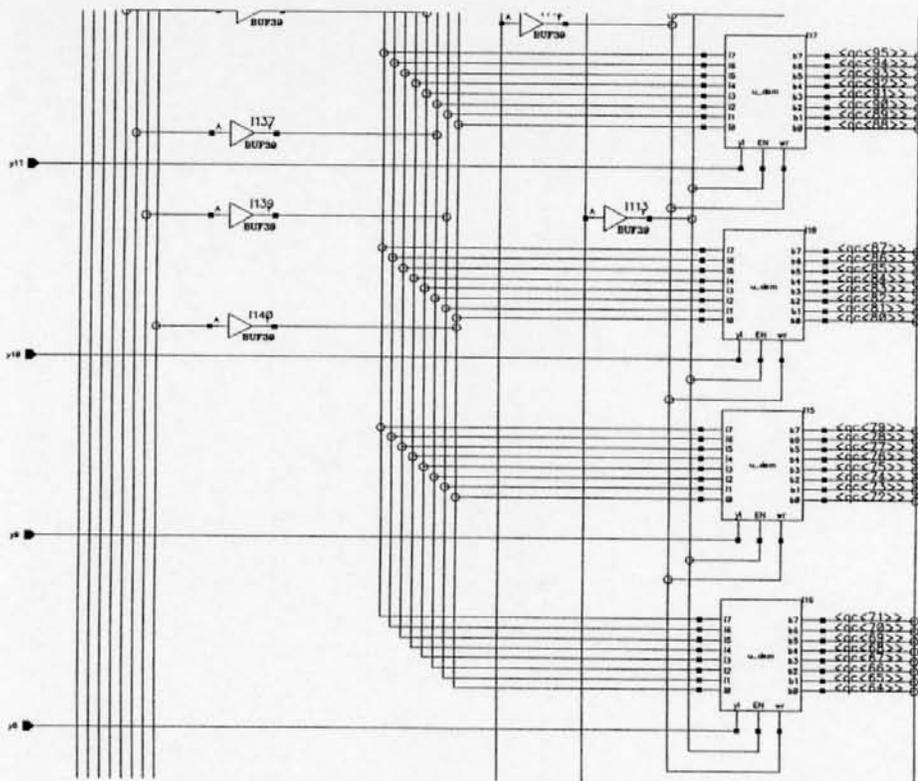


FIGURA A-9.8 – Módulo dsm (6/8).

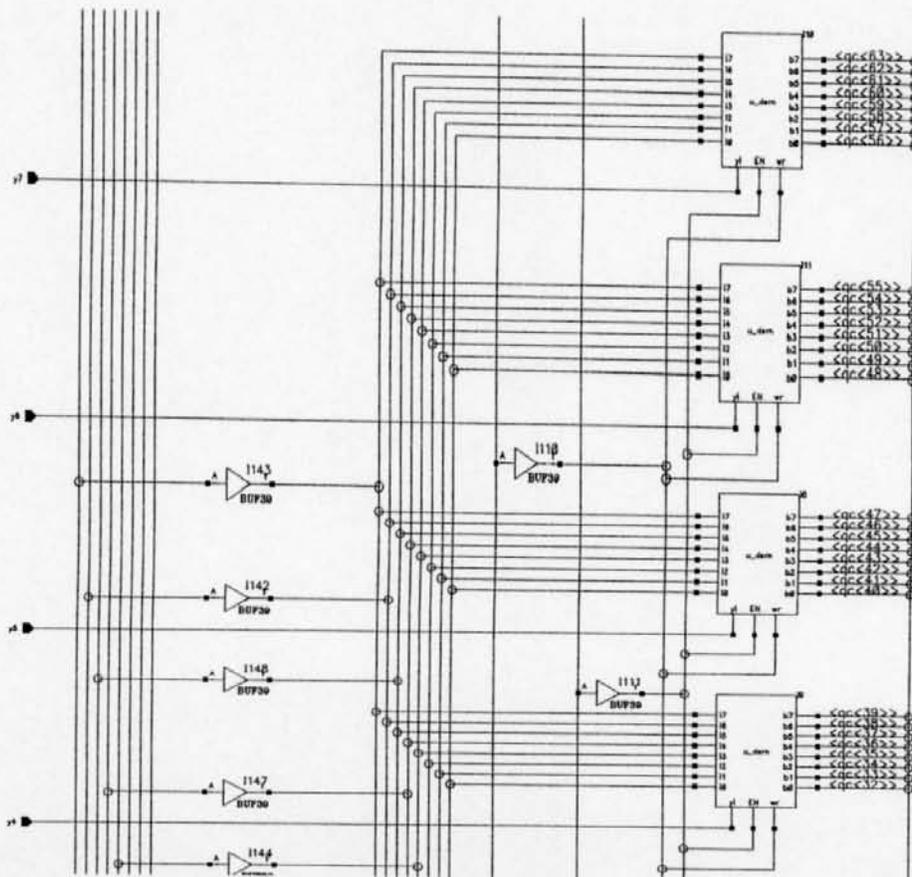


FIGURA A-9.9 - Módulo dsm (7/8).

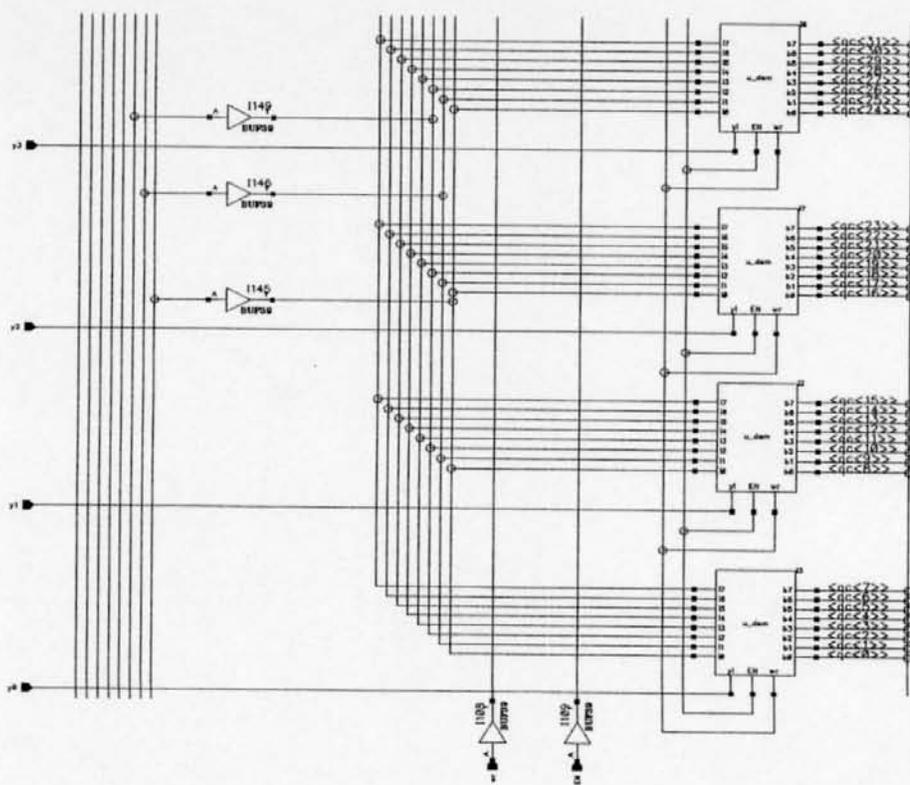


FIGURA A-9.10 - Módulo dsm (8/8).

**ANEXO A-10    SUBCIRCUITOS  
AUXILIARES II**

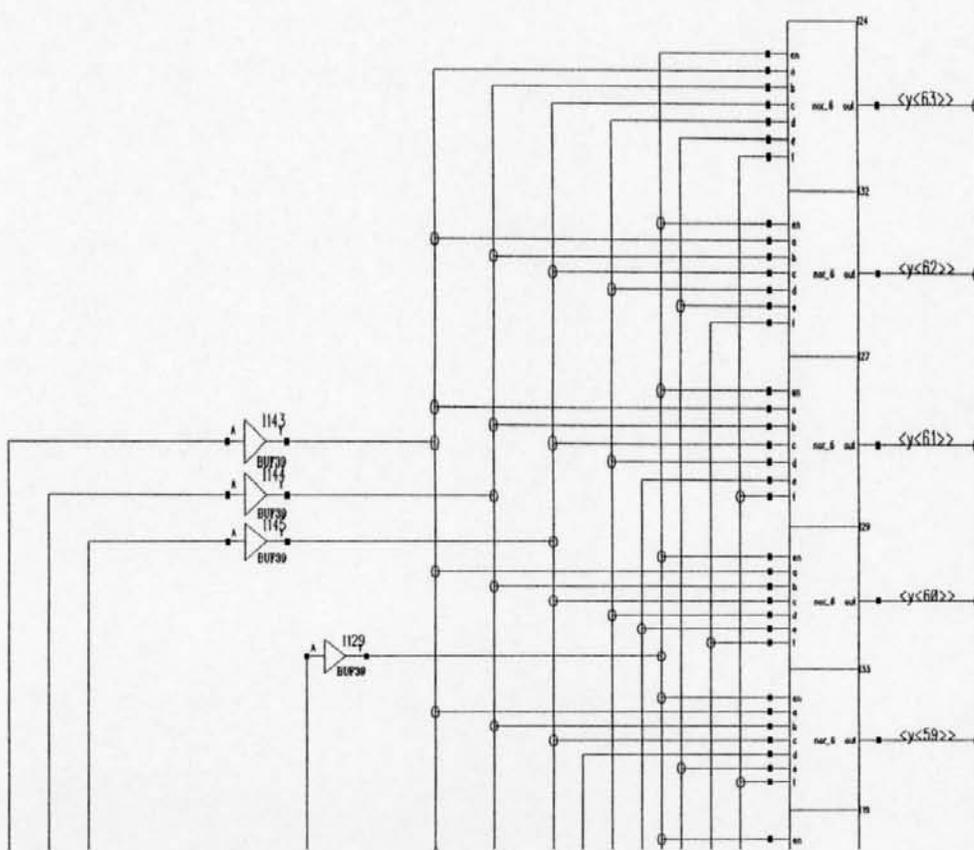


FIGURA A-10.1 – Módulo dec-6x64 (1/17).

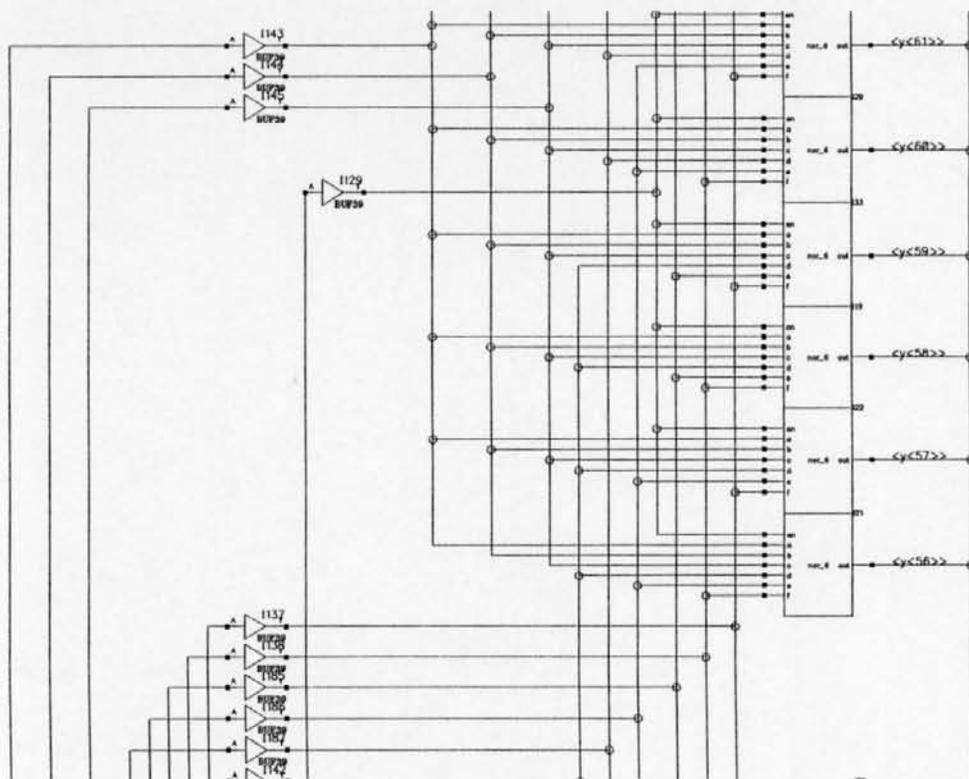


FIGURA A-10.2 – Módulo dec-6x64 (2/17).

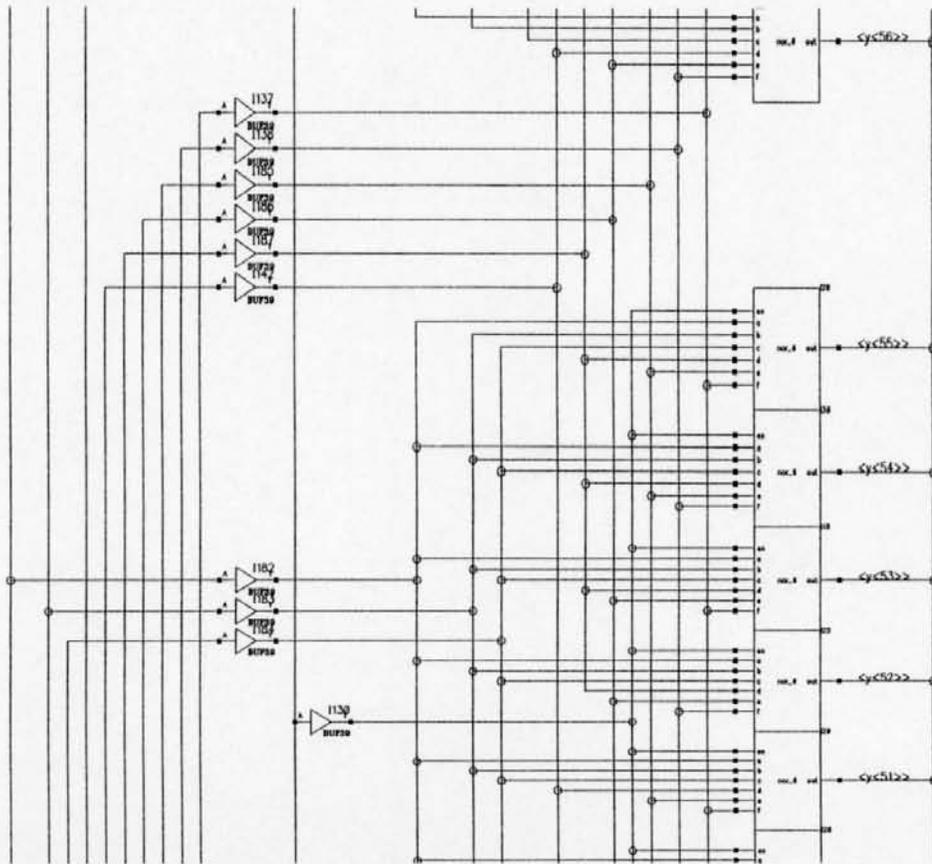


FIGURA A-10.3 – Módulo dec-6x64 (3/17).

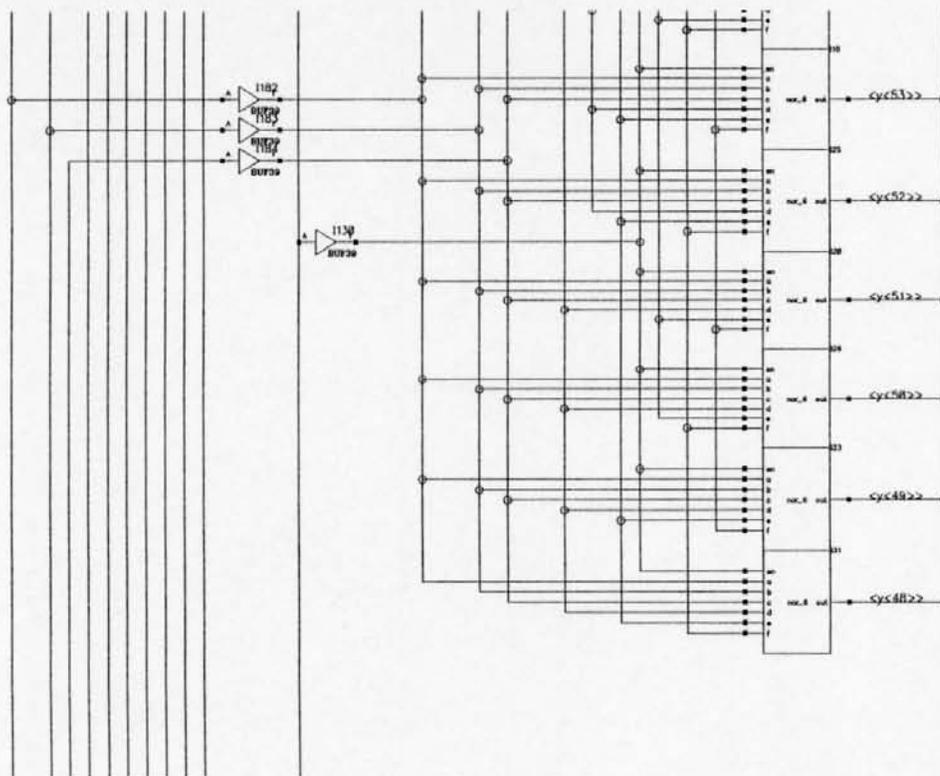


FIGURA A-10.4 – Módulo dec-6x64 (4/17).

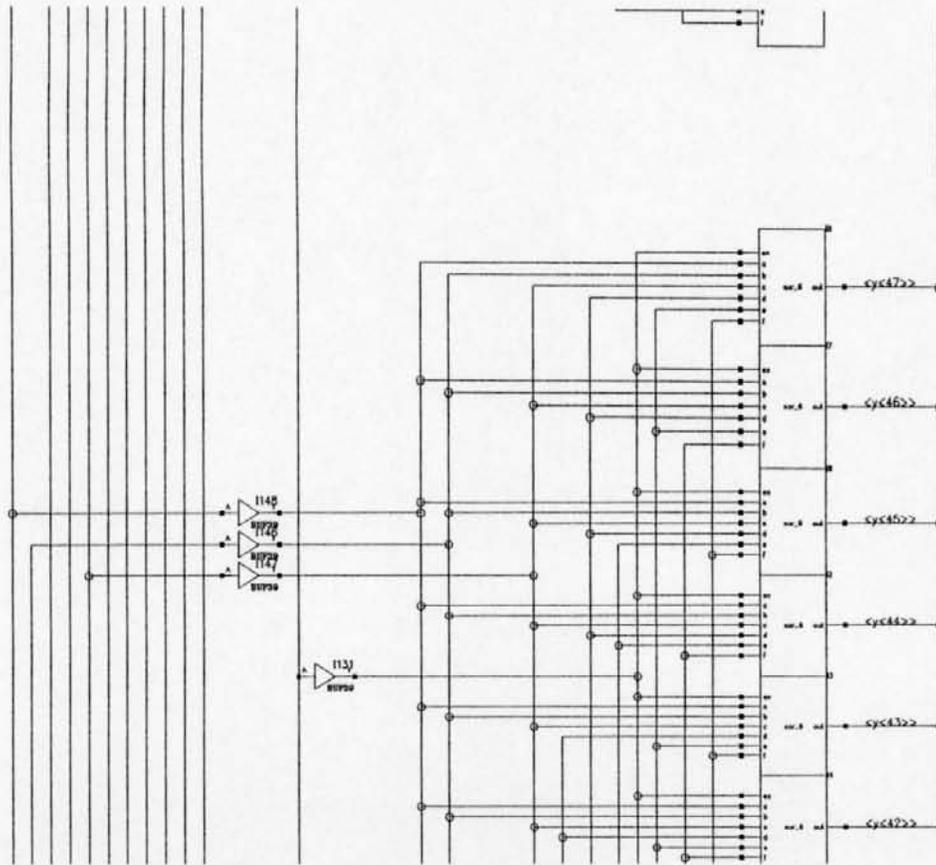


FIGURA A-10.5 – Módulo dec-6x64 (5/17).

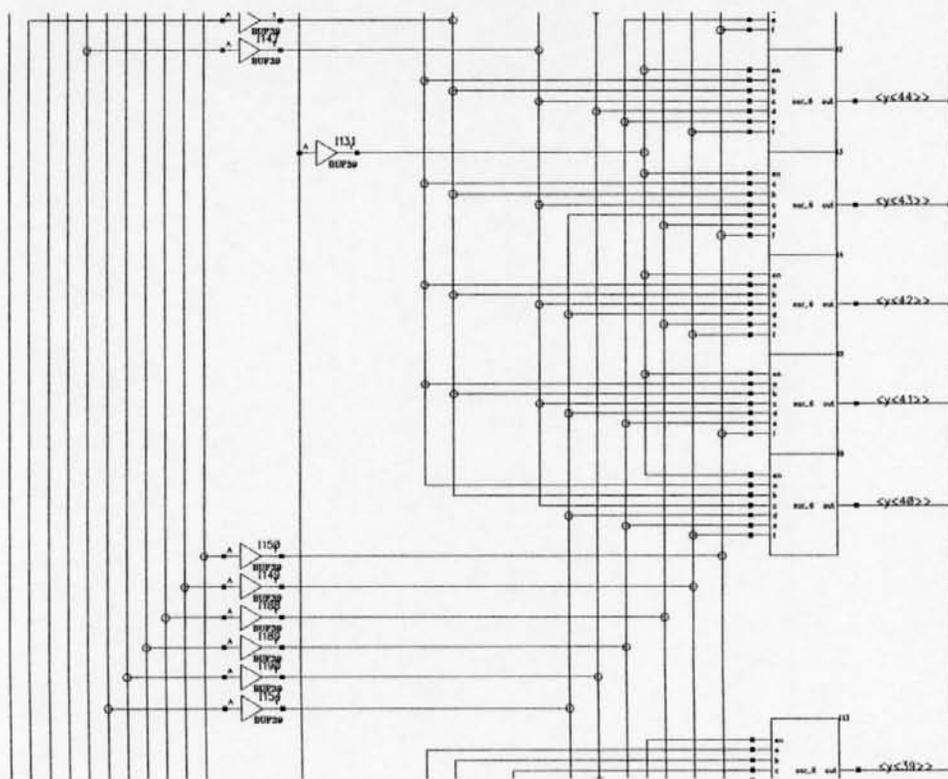


FIGURA A-10.6 – Módulo dec-6x64 (6/17).



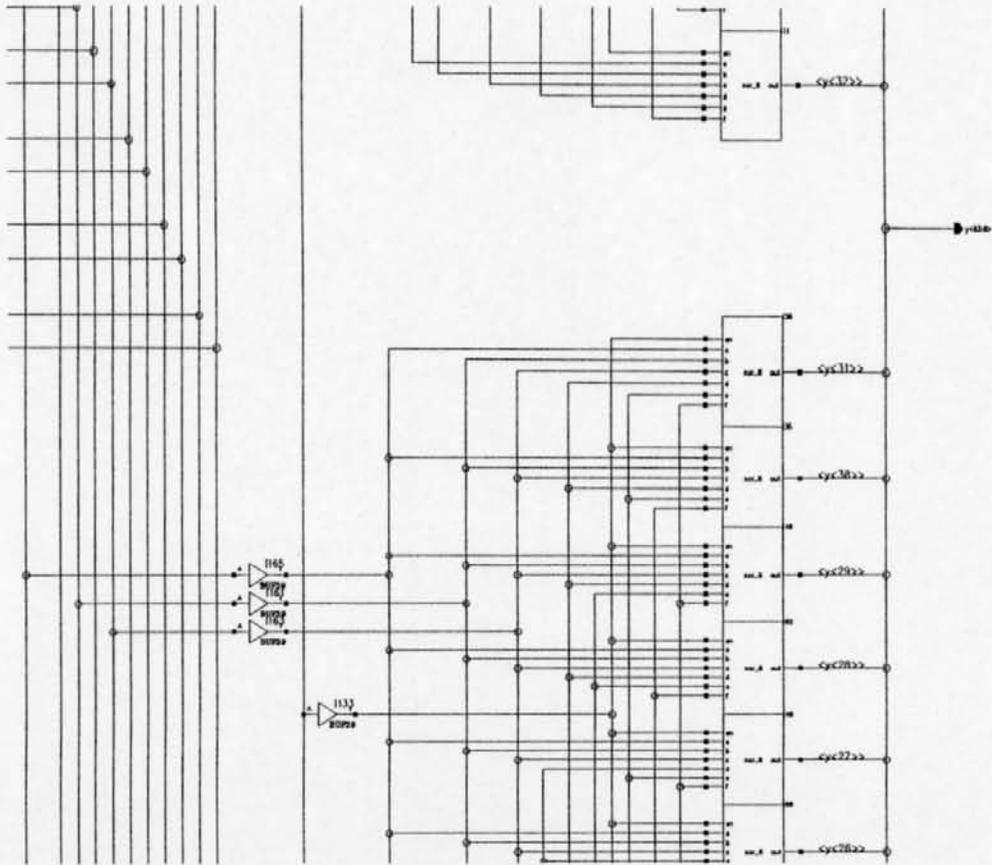


FIGURA A-10.9 – Módulo dec-6x64 (9/17).

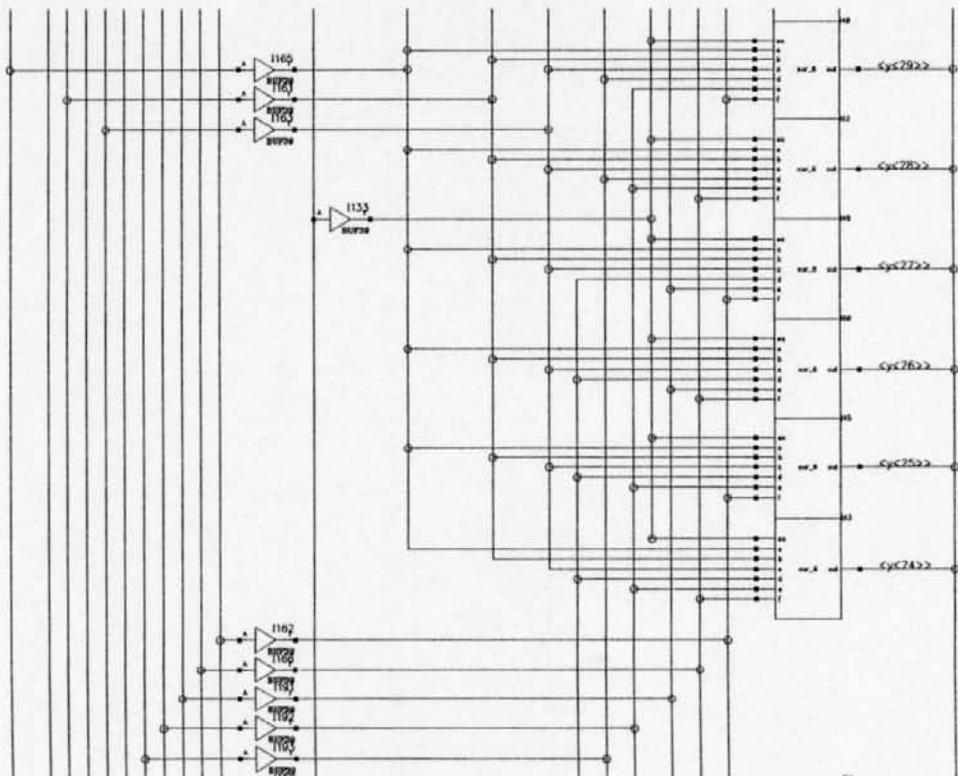


FIGURA A-10.10 – Módulo dec-6x64 (10/17).

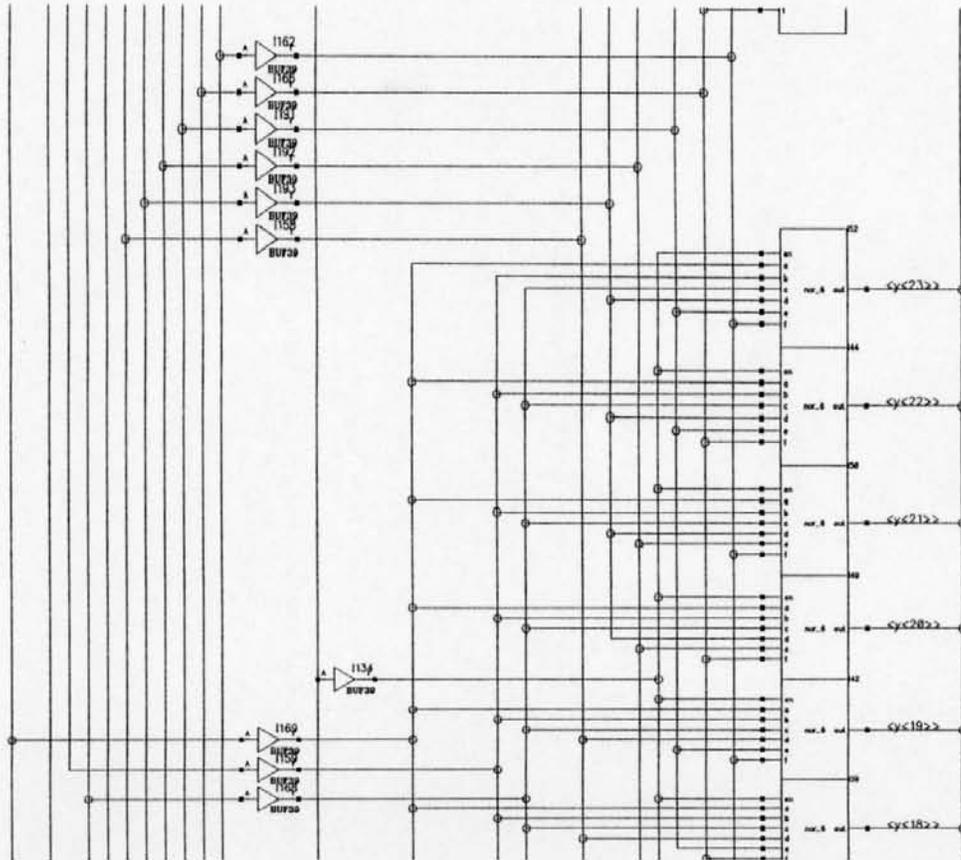


FIGURA A-10.11 - Módulo dec-6x64 (11/17).

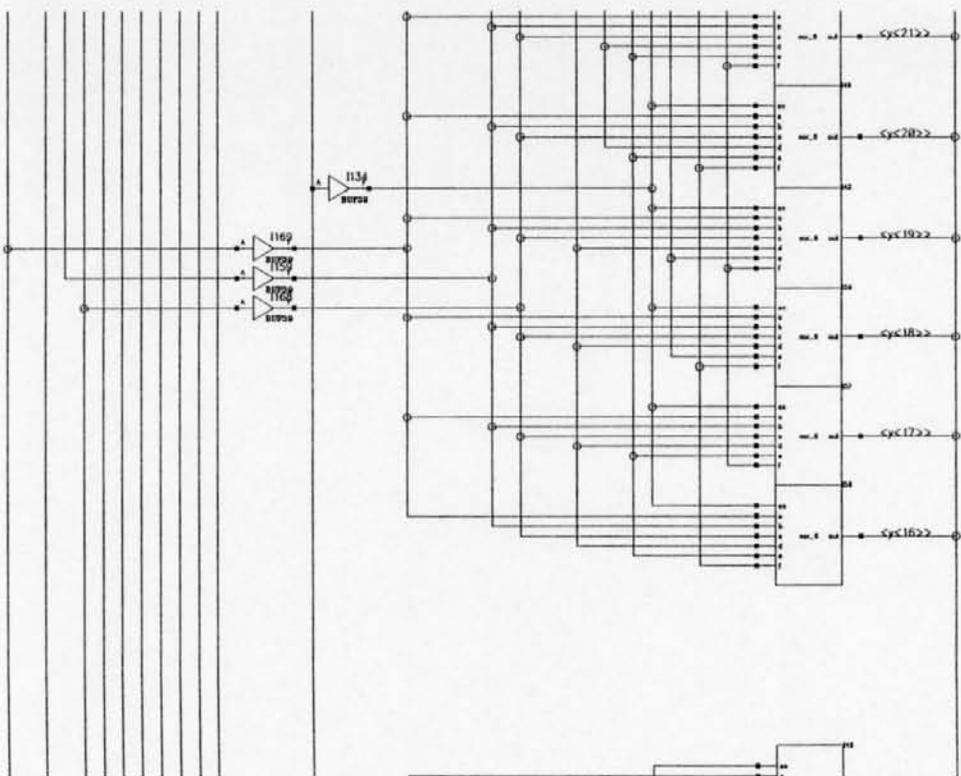


FIGURA A-10.12 - Módulo dec-6x64 (12/17).

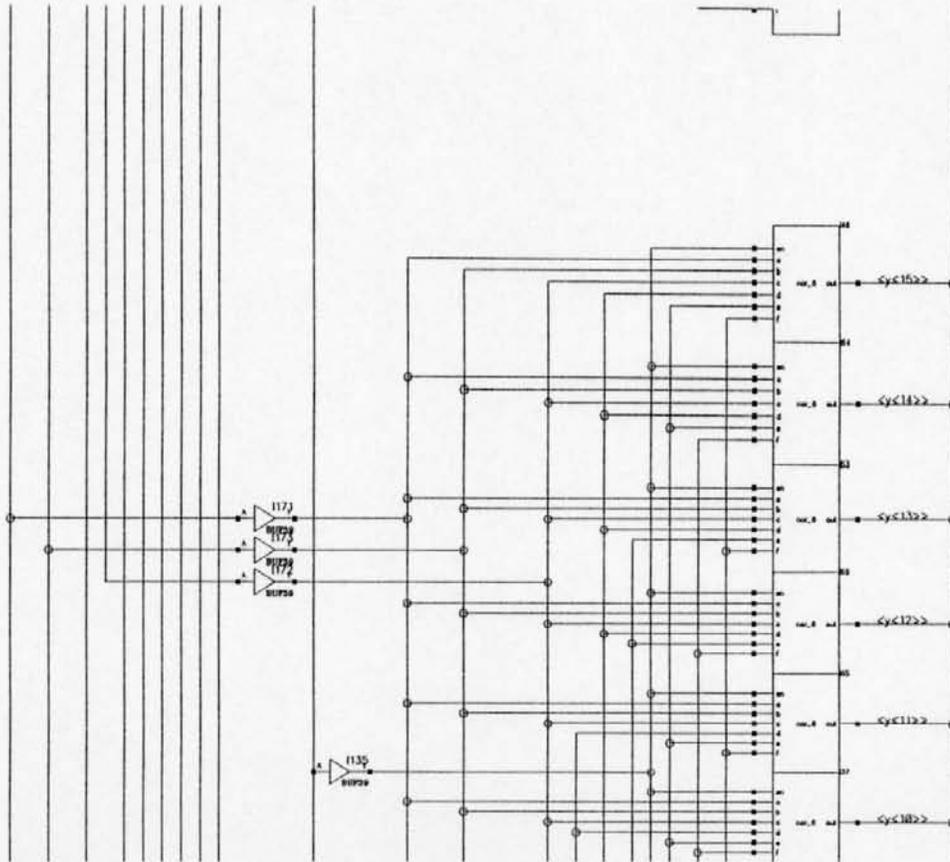


FIGURA A-10.13 - Módulo dec-6x64 (13/17).

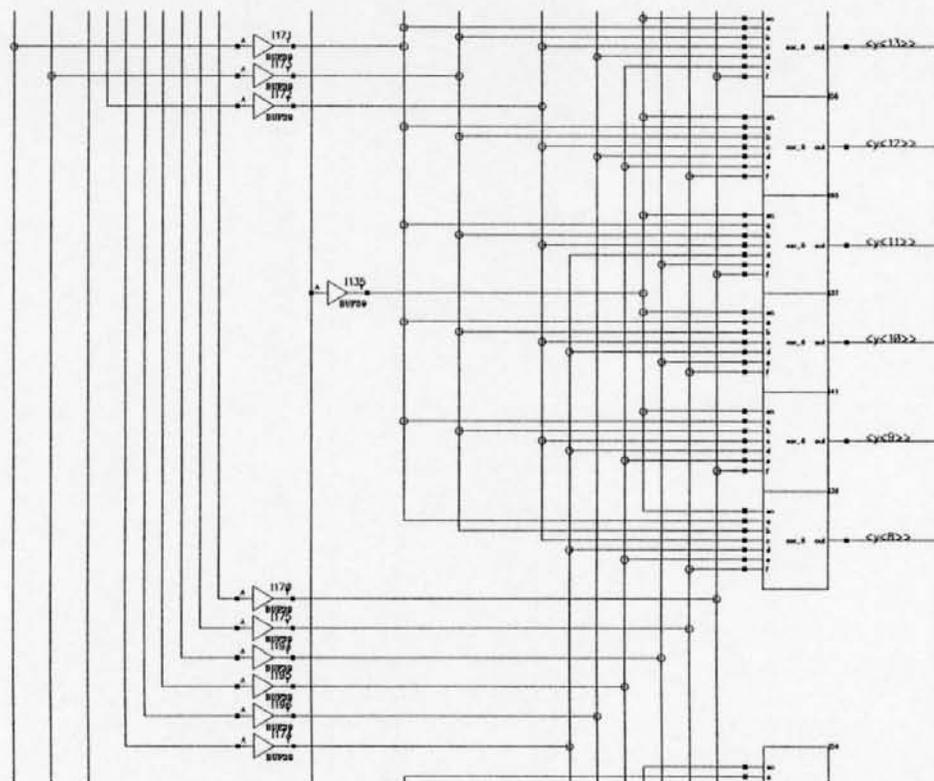


FIGURA A-10.14 - Módulo dec-6x64 (14/17).

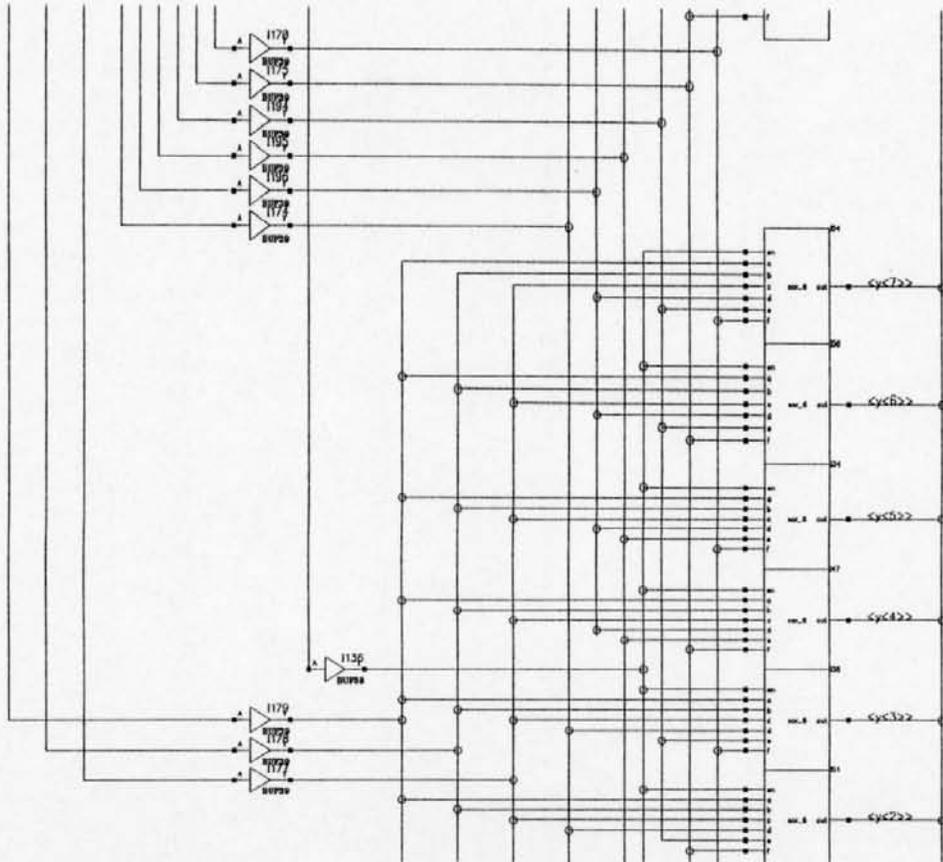


FIGURA A-10.15 – Módulo dec-6x64 (15/17).

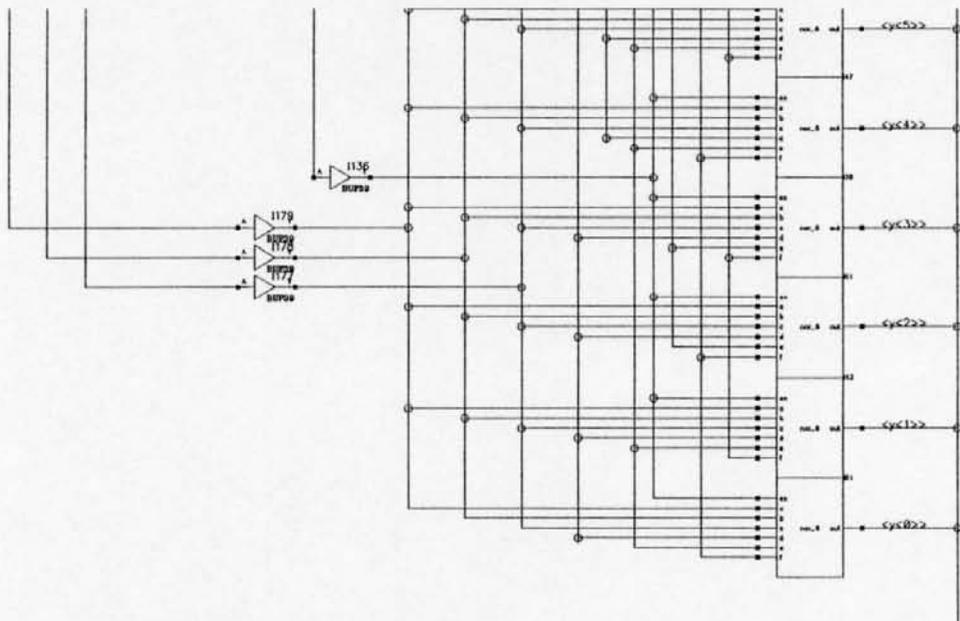


FIGURA A-10.16 – Módulo dec-6x64 (16/17).

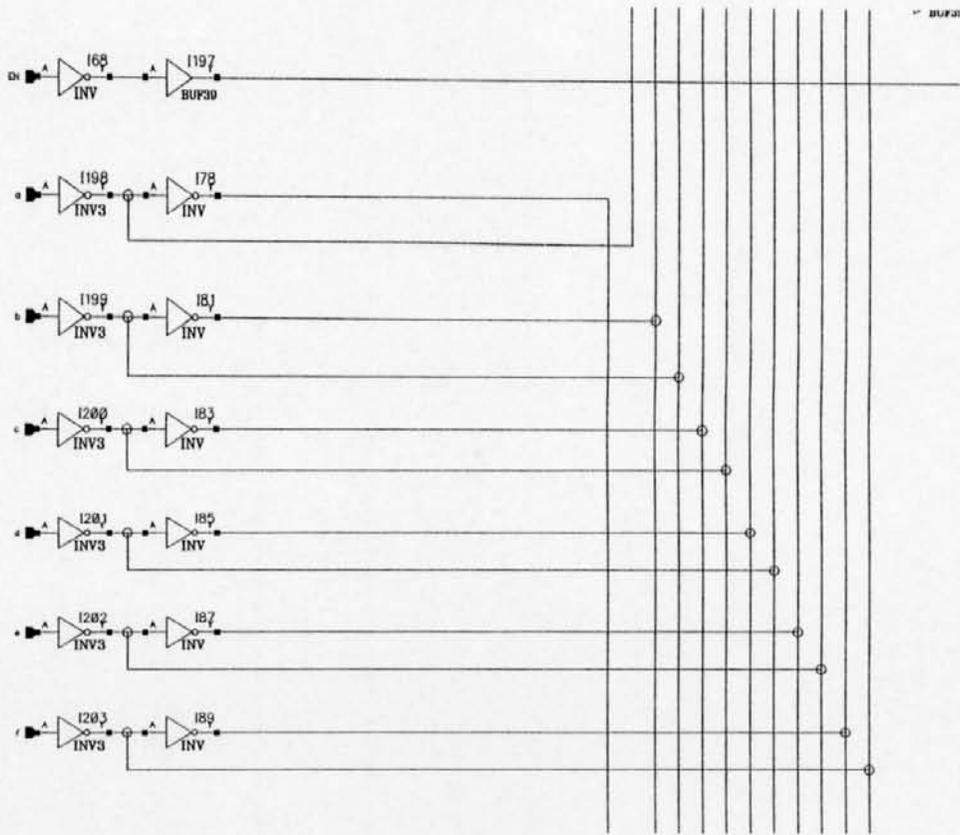


FIGURA A-10.17 – Módulo dec-6x64 (17/17).

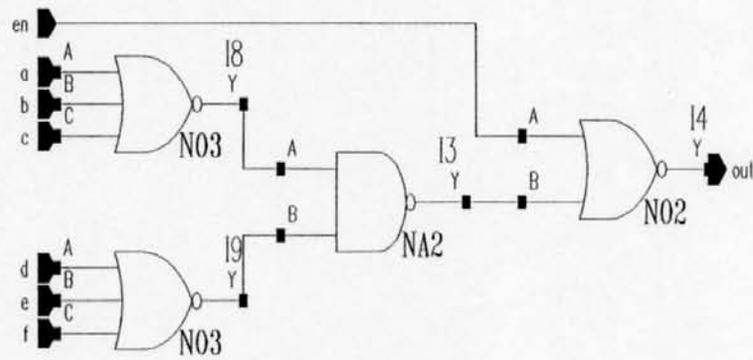


FIGURA A-10.18 – Módulo nor-6.

**ANEXO A-11    SUBCIRCUITOS DO *BUFFER*  
DE ALINHAMENTO**

FIGURA A-11.1 - Módulo 256.celda (1/2).

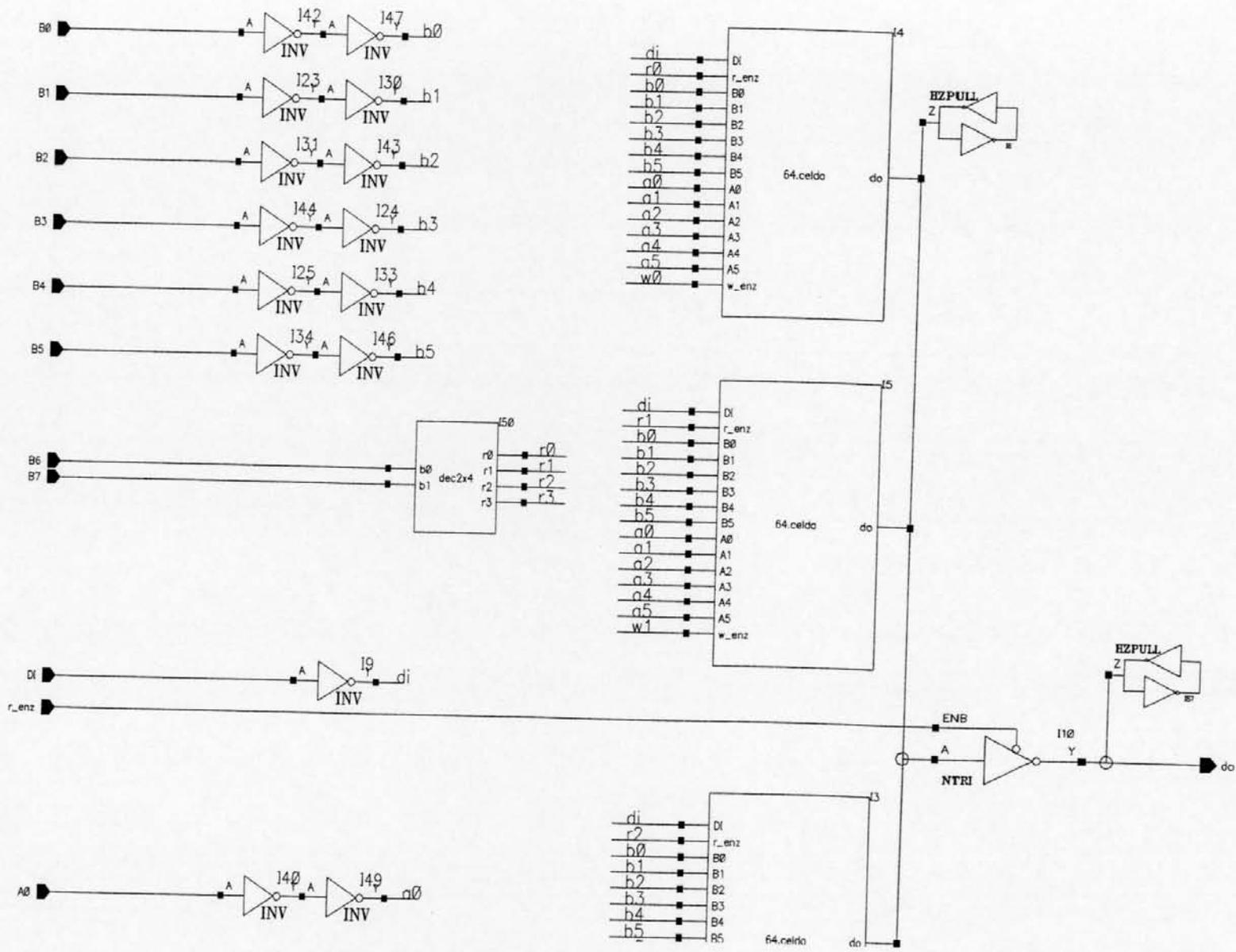
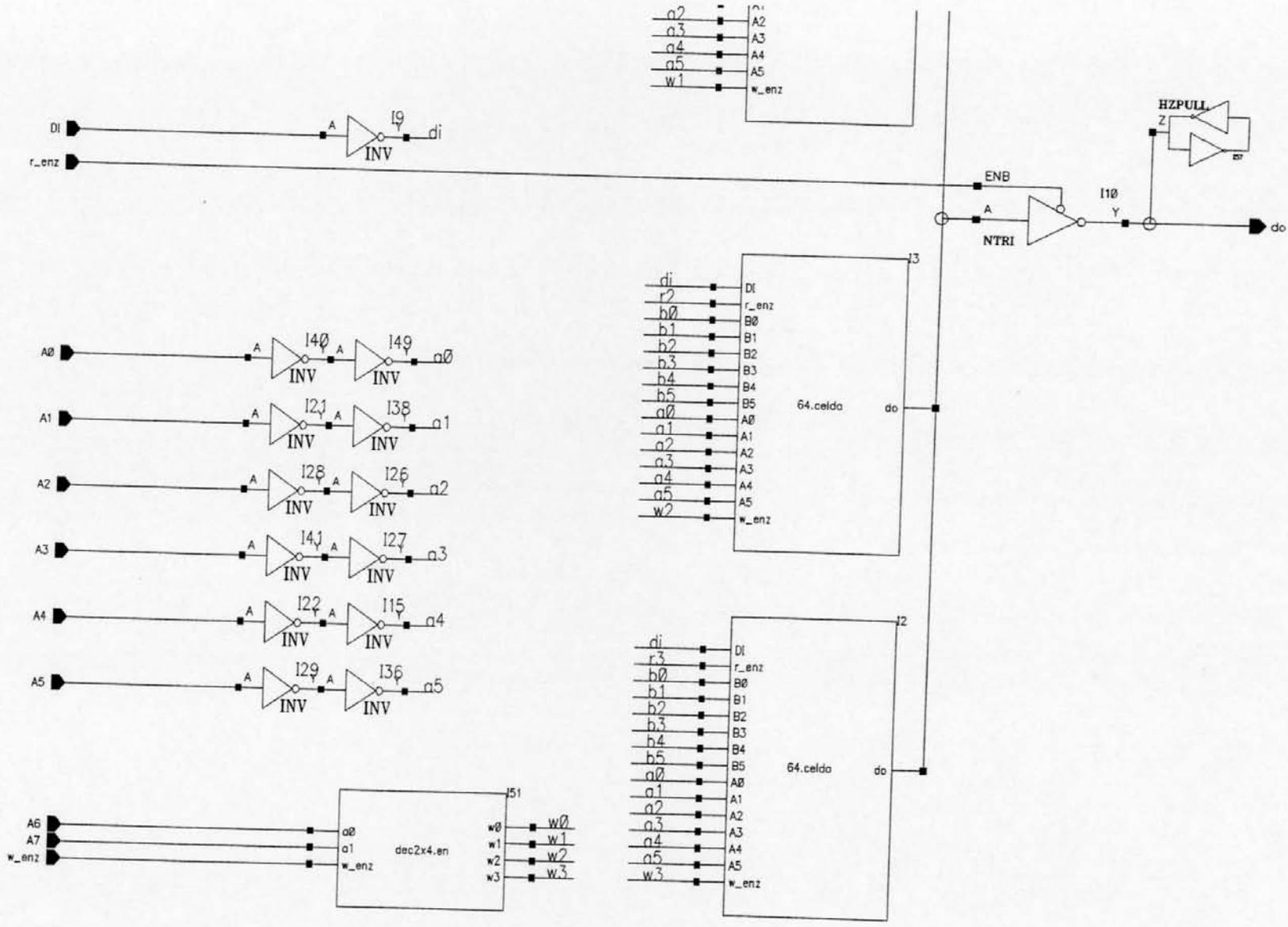


FIGURA A-11.2 - Módulo 256.celda (2/2).



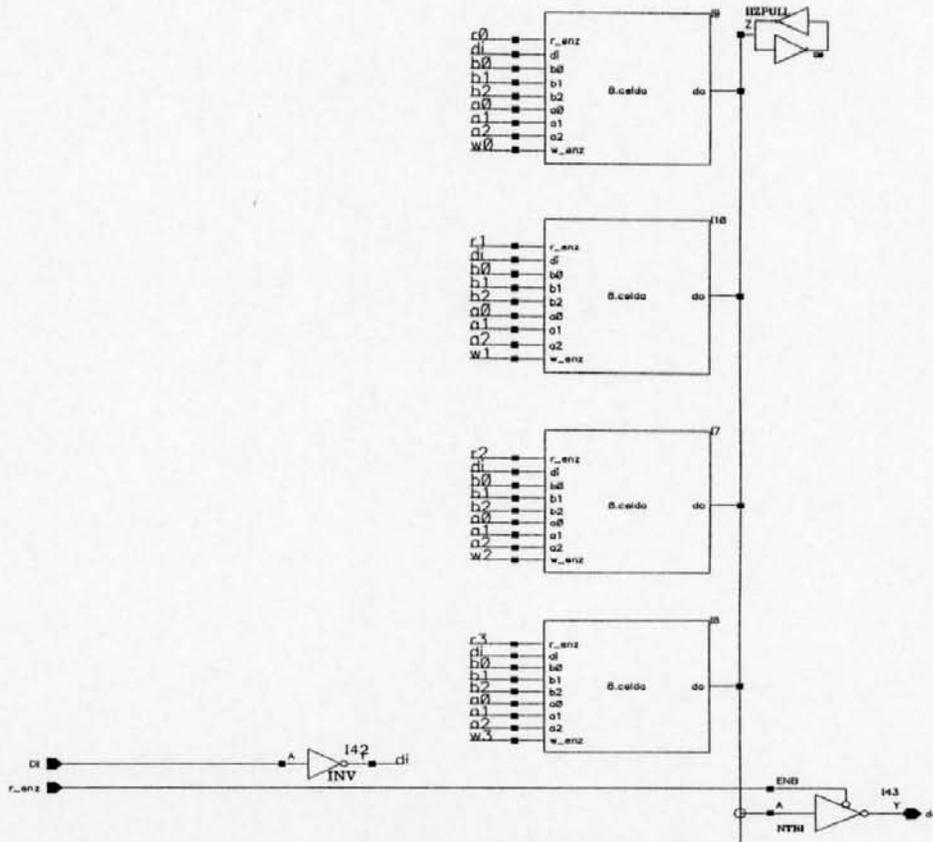


FIGURA A-11.3 - Módulo 64.celda (1/2).

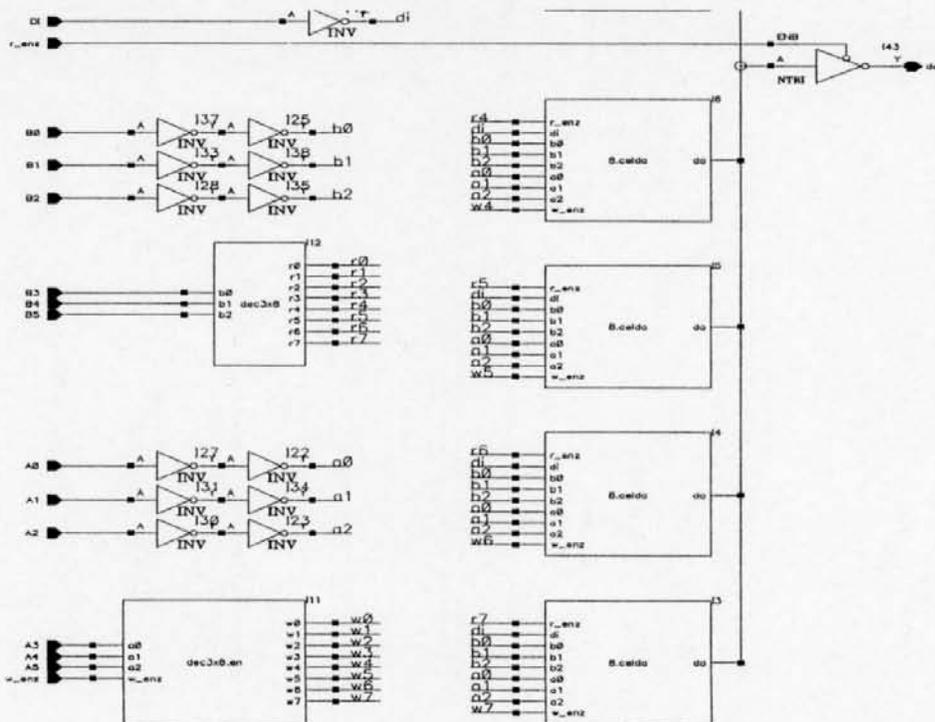


FIGURA A-11.4 - Módulo 64.celda (2/2).

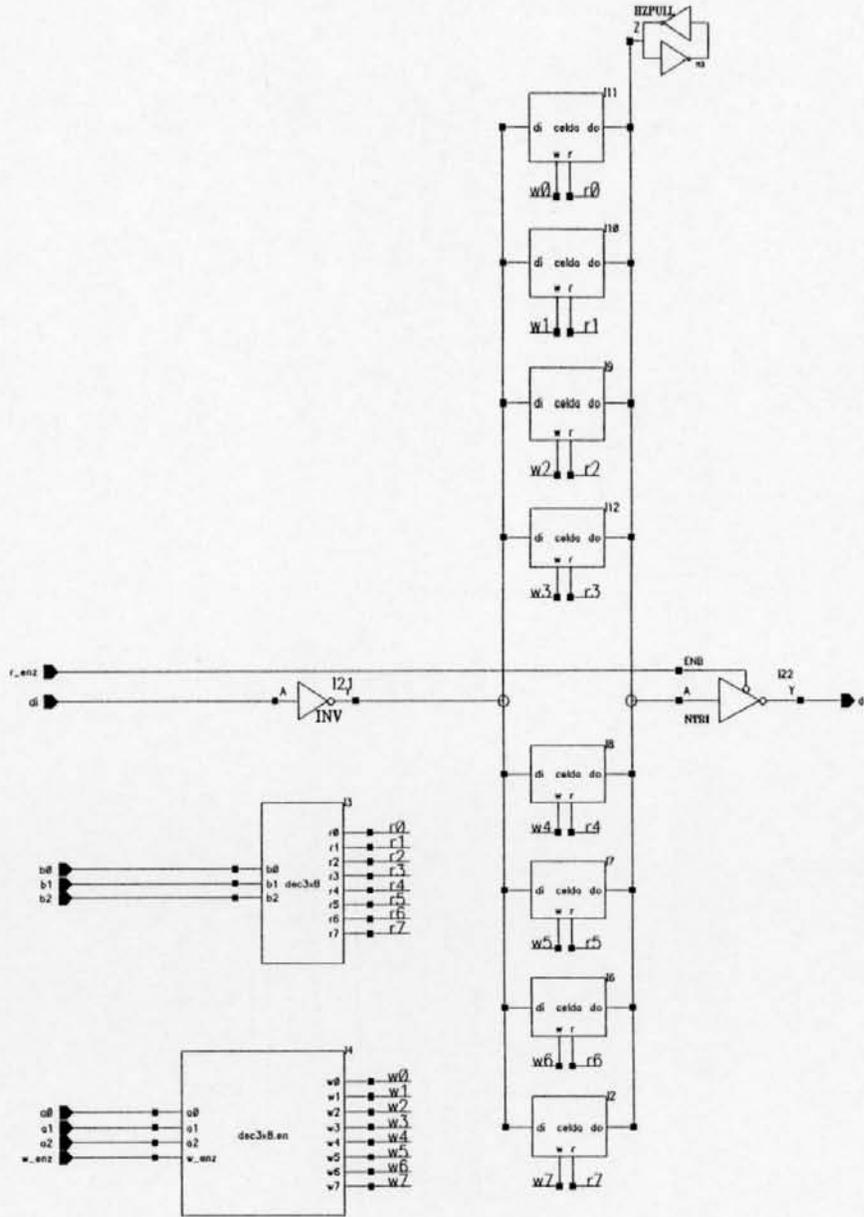


FIGURA A-11.5 – Módulo 8.celda.

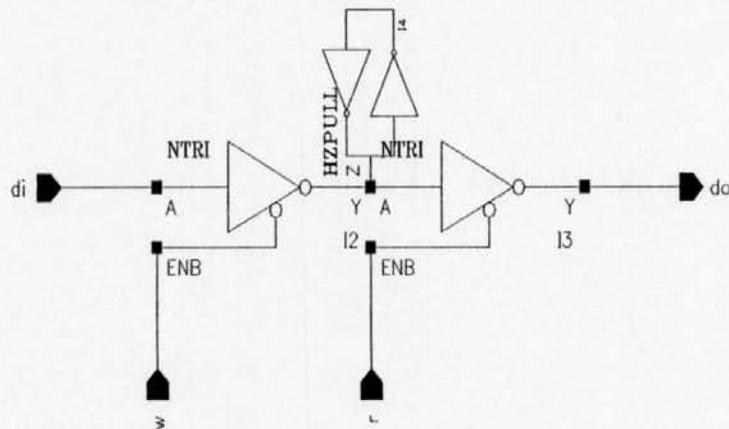


FIGURA A-11.6 – Módulo celda.

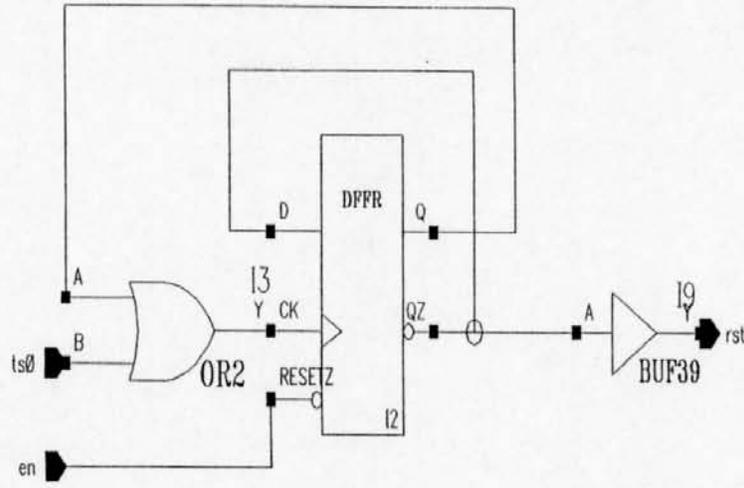


FIGURA A-11.7 – Módulo add.gen.strt.

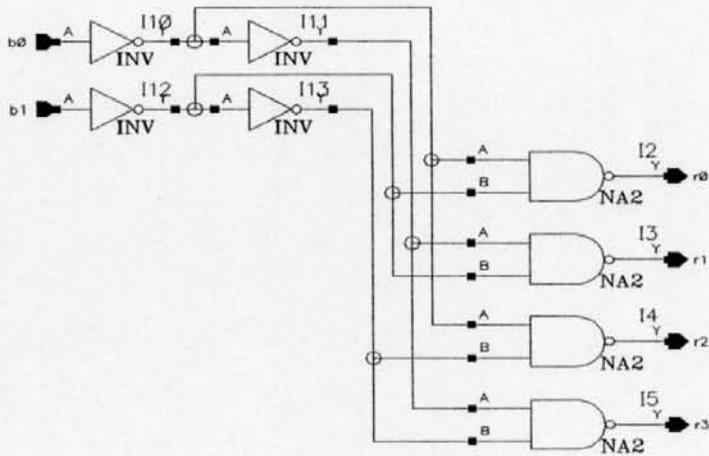


FIGURA A-11.8 – Módulo dec2x4.

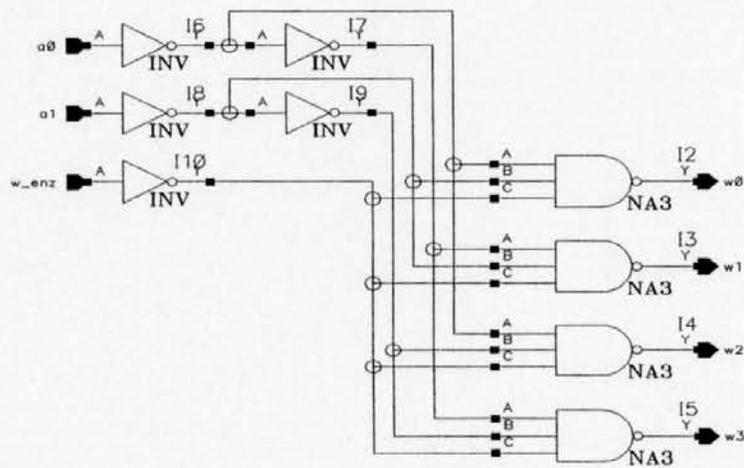


FIGURA A-11.9 – Módulo dec2x4.en.

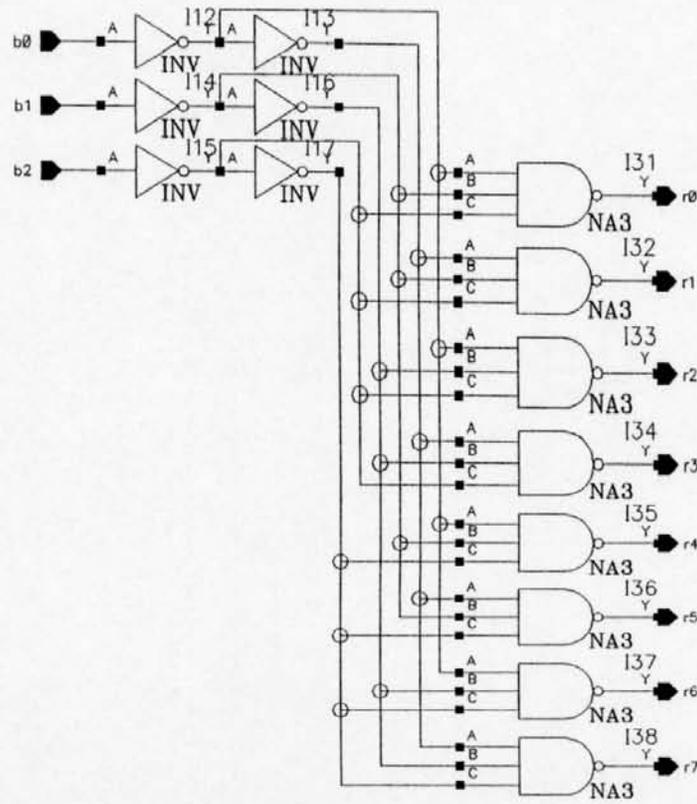


FIGURA A-11.10 - Módulo dec3x8.

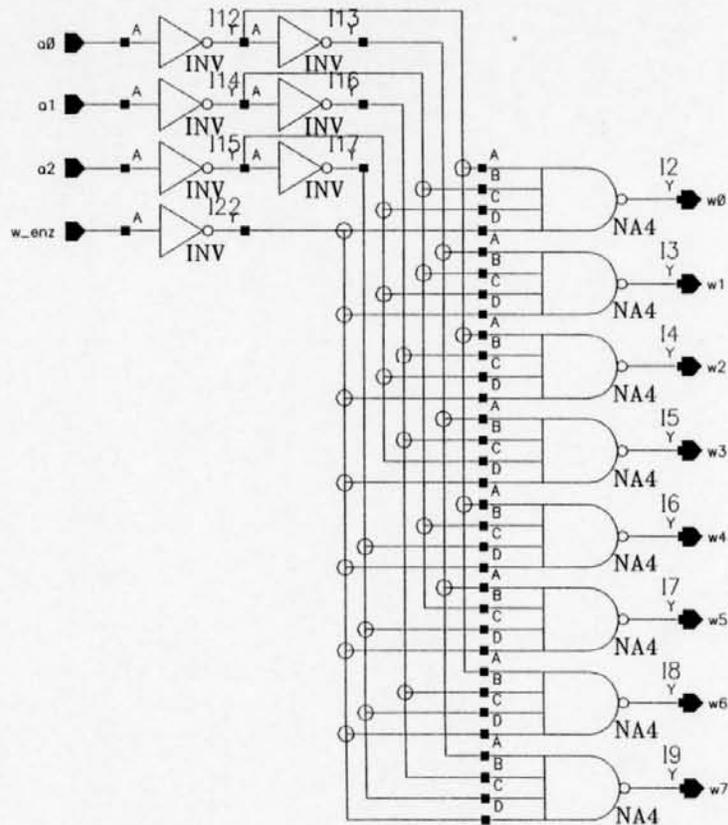


FIGURA A-11.11 - Módulo dec3x8.en.

## BIBLIOGRAFIA

- [ACK 94] ACKLAND, B. The Role of VLSI in Multimedia. **IEEE Journal of Solid-State Circuits**, New York, v.29, p.381-388, Apr. 1994.
- [AGU 95] AGURTO, O. P. Um Estudo Comparativo das Técnicas de Comutação de Circuitos em Redes de Comunicação de Dados. Porto Alegre: CPGCC da UFRGS, Jan. 1995. 50p. Trabalho Individual.
- [AGU 96] AGURTO, O. P.; WAGNER, T.V.; ROCHOL, J. Digital Switch for E1 End-Mux: A Microelectronics Design. In: SBCCI, 9., 1996, Recife, PE. **Proceedings...** Recife: UFPE, 1996. p.189-199.
- [BAT 92] BATES, B. **Introduction to T1/T3 Networking**. Boston: Artech House, 1992. 192p.
- [BAX 82] BAXTER, L. A.; BAUGH, C.R. A Comparison of Architectural Alternatives for Local Voice/Data Communications. **IEEE Communications Magazine**, New York, v.20, p.44-51, Jan. 1982.
- [BEL 90] BELL COMMUNICATION RESEARCH TECHNICAL PERSONNEL. **Telecommunications Transmission Engineering vol.1**. USA: Bellcore, 1990. 787p.
- [BEL 90a] BELL COMMUNICATION RESEARCH TECHNICAL PERSONNEL. **Telecommunications Transmission Engineering vol.2**. USA: Bellcore, 1990. 732p.
- [BEJ 91] BELLAMY, J. **Digital Telephony**. New York: John Willey & Sons, 1991. 572p.
- [BUD 84] BUDRIKIS, Z. L.; NETRAVALI, A. N. A Packet/Circuit Switch. **AT&T Bell Laboratories Technical Journal**, New York, v.63, p.1499-1520, Oct. 1984.

- [BUR 93] BURSON, A.; BAKER, A. Optimizing Communications Solutions. **IEEE Communications Magazine**, New York, v.31, p.15-19, Jan. 1993.
- [CAD 88] CADENCE SYSTEMS INC. **Design Framework Manual**. USA: SDA Systems, 1988.
- [CHA 79] CHARRANSOL, P. et al. Development of a Time Division Switching Network Usable in a Very Large Range of Capacities. **IEEE Transactions on Communications**, New York, v.27, p.982-987, July 1979.
- [CHE 88] CHEN, T. M.; MESSERSCHMITT, D.G. Integrated Voice/Data Switching. **IEEE Communications Magazine**, New York, v.26, p.16-26, June 1988.
- [COO 79] COOPER, C. A. Synchronization for Telecommunications in a Switched Digital Network. **IEEE Transactions on Communications**, New York, v.27, p.1028-1033, July 1979.
- [COR 93] CORREIA, M.; BERNAL, P. **Rede Corporativa Integrada**. São Paulo: Érica, 1993. 150p.
- [DOP 94] DOPKE, R.; NOGUEIRA, A.; DE ROCHA, D. **Comunicação de Dados - Rede de Serviços Digitais E1**. Curitiba: Embratel, 1994. 106p. Apostilha de Treinamento.
- [EUR 89] EUROPEAN SILICON STRUCTURES. **Solo 2000 Standard Cells Family Libraries Manual**. France: European Silicon Structures, 1989. 207p.
- [GIO 86] GIOZZA, W.; ARAÚJO, J.; SAUVÉ, J. **Redes Locais de Computadores**. São Paulo: McGraw Hill, 1986. 399p.

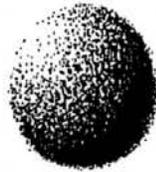
- [HAR 78] HARRINGTON, E. A. Synchronization Techniques for Various Switching Network Topologies. **IEEE Transactions On Communications**, New York, v.26, p.925-932, June 1978.
- [INF 80] INFOTECH STATE-OF-THE-ART REPORT. **Data Communications Analysis**. England: Infotech, 1980.221p.
- [INT 86] INTEL. **Microcontroller Handbook**. USA: Intel Corporation, 1986.864p.
- [INT 87] INTEL. **Microcommunications Handbook**. USA: Intel Corporation, 1987.1604p.
- [ITU 88] INTERNATIONAL TELECOMMUNICATION UNION. **General Aspects of Digital Transmission Systems: Terminal Equipments**, Recomendations G.700-G.795. Melbourne, 1988.
- [JOE 79] JOEL JR, A. E. Digital Switching - How It has Developed. **IEEE Transactions On Communications**, New York, v.27, p.948-959, July 1979.
- [JOE 83] JOEL JR, A. E. Public Digital Switching Systems. **IEEE Communications Magazine**, New York, v.21, p.6-10, May 1983.
- [KAI 94] KAISER, G.W. **Multiplexação de Dados em Alta Velocidade**. Porto Alegre: CPGCC da UFRGS, Fev.1994. 47p. Trabalho Individual.
- [KAI 95] KAISER, G.W.; WAGNER, T.V.; ROCHOL, J. Time Alignment of Plesiochronous and Mesochronous Signals in Digital Data Communication Systems. In: SBMICRO, 10., 1995, Canela, RS. **Proceedings...** Porto Alegre: Palotti, 1995. v.1, p.223-231.
- [KAI 95a] KAISER, G.W. **Funções de Multiplexação E1**. Porto Alegre: CPGCC da UFRGS, Set.1995. 200p. Dissertação de Mestrado.

- [LEE 90] LEE, E.A.; MESSERSCHIMMITT, D.G. **Digital Communication**. Boston: Kluwer Academic, 1990. 713p.
- [LIS 83] LISSACK, T.; MAGLARIS, B.; FRISCH, I.T. Digital Switching in Local Area Networks. **IEEE Communications Magazine**, New York, v.21, p.26-37, May 1983.
- [MAL 88] MALEK, M. Integrated Voice and Data Communications Overview. **IEEE Communications Magazine**, New York, v.26, p.5-15, June 1988.
- [MEL 78] MELVIN, D.K. Microcomputer Applications in Telephony. **Proceedings of the IEEE**, New York, v.66, p.5-15, Feb. 1978.
- [MIT 86] MITEL. **Components Handbook**. USA: Mitel Corporation, 1986.
- [MON 90] MONTORO, F. **Transmissão de Dados e Modem**. São Paulo: Érica, 1990. 265p.
- [MUK 86] MUKHERJEE, A. **Introduction to nMOS and CMOS VLSI Systems Design**. New Jersey: Prentice Hall, 1986. 370p.
- [OWE 82] OWEN, F.F.E. **PCM and Digital Transmission Systems**. New York: McGraw Hill, 1982. 295p.
- [PER 93] PERSONICK, S. D. The Evolving Role of Telecommunications Switching. **IEEE Communications Magazine**, New York, v.31, p.20-24., Jan. 1993.
- [PEN 88] PENNA, R.M. **Teleprocessamento: Conceitos, Aplicações e o Protocolo BSC3**. São Paulo: Érica, 1988. 222p.
- [PIT 79] PITRODA, S.G. Telephones Go Digital. **IEEE Spectrum**, New York, v.16, p.51-60, Oct. 1979.
- [ROC 93] ROCHOL, J. **Comunicação Digital**. Pelotas: CTMR, Dez. 1993. 162p. Apostilha de Treinamento.

- [ROR 85] RORABAUGH, B. **Data Communications and Local Area Networking Handbook**. New York: Tab Books, 1985. 231p.
- [SIL 88] SIMUCAD INC. **SILOS User's guide**. California: Simucad, 1988.
- [SCH 82] SCHRAMMEL, F. J. Trends in Digital Switching and ISDN. **Telecommunication Journal**, New York, v.49, p.421-429, July 1982.
- [SKA 79] SKAPERDA, N. J. Some Architectural Alternatives in the Design of a Digital Switch. **IEEE Transactions On Communications**, New York, v.27, p.961-972, July 1979.
- [SOA 95] SOARES, L.F.; LEMOS, G.; COLCHER, S. **Redes de Computadores: Das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 1995. 576p.
- [SPO 93] SPOHN, D.L. **Data Network Design**. New York: McGraw-Hill, 1993. 778p.
- [SRI 94] SRISKANTHAN, N. et al. An Adaptive Switching Architecture for Multiprocessor Networks. **Microprocessors and Microsystems**, New York, v.18, p.307-314, July-Aug. 1994.
- [STA 90] STALLINGS, W. Digital Signaling: Which techniques are best and why it matters to you. **Basic Guide to Data Communications**. London: McGraw Hill, 1990. p.236-241.
- [STA 92] STALLINGS, W. **ISDN and Broadband ISDN**. New York: MacMillan, 1992. 633p.
- [STE 93] STEWART, J. W. **The 8051 Microcontroller: Hardware, Software and Interfacing**. New Jersey: Prentice Hall, 1993. 273p.
- [TAN 88] TANENBAUM, A. **Computer Networks**. New Jersey: Prentice Hall, 1988. 658p.

- [TAR 84] TAROUCO, L. M. **Redes de Comunicação de Dados**. Rio de Janeiro: Livros Técnicos e Científicos, 1984. 218p.
- [WES 93] WESTE, N.H.E.; ESHRAGHIAN, K. **Principles of CMOS VLSI Design**. New York: Addison-Wesley, 1993. 713p.
- [YAM 86] YAMAMOTO, Y.; MIYANAGA, H. et al. A Novel Concept for High-Speed Time Switch Approaching Memory Read Cycle Limit. **IEEE Transactions On Communications**, New York, v.34, p.953-955, Sept. 1986.

**Informática**



**UFRGS**

**CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

*Comutador de Dados Digitais para TDM Determinístico EI, Visando uma  
Implementação em Microeletrônica.*

por

Oscar Pedro Agurto Hoyos

Dissertação apresentada aos Senhores:

Prof. Dr. Carlos Eduardo Pereira (DELET/UFRGS)

Dr. Fabian Vargas

Dr. Fernando Gehm Moraes

Prof. Juergen Rochol

Prof. Dr. Ricardo Pezzuol Jacobi

Vista e permitida a impressão.

Porto Alegre, 10 / 05 / 96.

Prof. Tiaraju Vasconcellos Wagner,  
Orientador.

**Prof. Flávio Rech Wagner**  
Coordenador do Curso de Pós-Graduação  
em Ciência da Computação - CPGCC  
Instituto de Informática - UFRGS