

IAN R. SINCLAIR

INTRODUÇÃO À ELECTRÓNICA DIGITAL

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

TEMPOS
LIVRES

CULTURA E TEMPOS LIVRES

1. ABC do Xadrez, *Petar Trifunovitch e Sava Vukovitch*
4. ABC do Bridge, *Pierre Jais e H. Lahana*
5. Guia Prático de Fotografia, *W. D. Emanuel*
6. ABC do Judo, *E. J. Harrison*
7. Como Fazer Cinema, *Paul Petzold*
8. Bridge Moderno *Pierre Jais e H. Lahana*
9. Fotografia — Técnicas e Truques, *Edwin Smith*
10. ABC dos Estilos — da Arquitectura ao Mobiliário, *A. Aussel*
11. Fotografia — Técnicas e Truques, *Edwin Smith*
12. A Pesca Submarina, *António Ribera*
13. Teoria dos Finais de Partida, *Yuri Averbach*
14. Aprenda Rádio, *B. Fighiera*
15. Guia do Cão, *Louise Laliberté-Robert e Jean-Pierre Robert*
16. ABC do Aquário, *Anthony Evans*
17. Iniciação à Electricidade e Electrónica, *Fernand Huré*
18. Os Transístores, *Fernand Huré*
19. Karaté, I, *Albrecht Pflüger*
20. Iniciação ao Radiocomando dos Modelos Reduzidos, *C. Péricon*
21. Construa o seu Receptor, *B. Fighiera*
22. Montagens Electrónicas, *B. Fighiera*
23. O Berbequim Eléctrico, *Villy Dreier*
24. Cactos, *J. Nilaus Jensen*
25. Iniciação à Alta Fidelidade, *Peter Turner*
26. O Aquário de Água Doce, *Paulo de Oliveira*
27. ABC do Ténis, *Fonseca Vaz*
28. Karaté, II, *Albrecht Pflüger*
29. ABC da Criação de Canários, *Curt Af Enehjelm*
30. Ginástica Feminina, *Sonja Helmer Jensen*
31. Cartomancia, *Rhea Koch*
32. Calculadoras Electrónicas de Bolso, *E. Dam Ravn*
33. O Pastor Alemão, *Gilles Legrand*
34. Xadrez — Teoria do Meio Jogo I, *Bondarevsky*
35. Manual do Super 8, I, *Myron A. Matzkin*
36. ABC da Criação de Periquitos, *Cyril H. Rogers*
37. O Livro dos Gatos, *Barbel Gerber e Horst Bielfeld*
38. Manual do Super 8, II, *Myron A. Matzkin*
39. ABC do Mergulho Desportivo, *Walter Mattes*
40. Circuitos Integrados/Aplicações Práticas, *F. Bergtold*
41. A Apicultura, *H. R. C. Riches*
42. ABC do Cultivo das Plantas, *H. G. Witham Fogg*
43. ABC da Criação de Pombos, *Kai R. Dahl*
44. Construção de Caixas Acústicas de Alta Fidelidade, *R. Brault*
45. Raças de Canários, *Klaus Speicher*
46. Jogos de Cartas, *Graciano Dolma*
47. Cocker Spaniels, *H. S. Lloyd*
48. ABC da Caça, *Fabian Abril*
49. Aprenda Televisão, *Gordon J. King*
50. Iniciação à Pesca, *Juan Nadal*
51. Basquetebol, *Marius Norregard*
52. Cães de Caça, *Santiago Pons*
53. Aprenda Electrónica, *T. L. Squires e C. M. Deason*
54. A Avicultura, *Jim Worthington*
55. A Produção de Coelho, *P. Surdeau e R. Henaff*
56. ABC dos Computadores, *T. F. Fry*
57. Natação para Crianças, *John Idorn*
58. O Boxer, *Anni Mortensen*
59. Voleibol, *Ole Hansen e Per-Göran Persson*
60. Iniciação à Vela, *Donald Law*
61. ABC da Filatelia, *Jacqueline Caurat*
62. A Pesca à Beira-Mar, *J.-M. Bøelle e B. Doyen*
63. Enxerto de Árvores de Fruto, *Alejo Rigau*
64. A Cultura do Morangueiro, *L. A. Grau*
65. Emissores-Receptores (Walkies-Talkies), *P. Duranton*
66. Iniciação à Fotoelectrónica, *Heinz Richter*
67. Doces e Conservas de Frutas, *Robin Howe*

68. A Criação de Hamsters, *C. F. Snow*
69. A Criação de Porcos, *Roy Genders*
70. Calendário do Horticultor, *Luis Alsina Grau*
71. Jogos Electrónicos, *F. Rayer*
72. Cultivo de Cogumelos e Trufas, *A. Rigau*
73. Aprenda Televisão a Cores, *G. J. King*
74. Gravação em Fita Magnética, *Ian R. Sinclair*
75. Pode de Árvores e Arbustos, *R. Genders*
76. Como Treinar o Seu Cão, *E. Fitch Daghish*
77. Instrumentos de Medida e Verificação, *Heinrich Stöckle*
78. A Criação de Caracóis, *Matias Josa*
79. Rádio - Fundamentos e Técnicas, *Gordon J. King*
80. Como Fazer Gelados, *Sylvie Thiébaud*
81. Iniciação à Jardinagem, *Noel Clarasó*
82. A Congelação dos Alimentos, *S. Lapointe*
83. Windsurf - Prancha à Vela, *E. Prade*
84. Raças de Cães, *O. Hasselfeldt*
85. Rummy e Canasta, *Claus D. Grupp*
86. A Encadernação, *Annie Persuy*
87. Aprenda Electricidade, *Heinz Richter*
88. Taxidermia - Embalsamamento de Pássaros e Mamíferos, *Harry Hjortaa*
89. Jogging - Correr para Manter a Forma, *Werner Sonntag*
90. ABC da Cozinha Chinesa, *S. Richmond*
91. Jogos T.V., *C. Tavernier*
92. Amplificadores de Som, *Richard Zierl*
93. O Livro do Poker, *Claus D. Grupp*
94. Aprenda a Desenhar, *Rose-Marie Prémont e Nicole Philippi*
95. O Minitrampolim na Escola, *Sonja Helmer Jensen e Klaus Dano*
96. Jogos de Luzes e Efeitos Sonoros para Guitarras, *B. Fighiera*
97. O Cultivo do Tomate, *Louis N. Flawn*
98. Pilhas Solares, *F. Juster*
99. Criação Doméstica de Coelho, *C. F. Snow*
100. Iniciação ao Futebol, *W. Männle e H. Arnold*
101. Horóscopos Chineses, *G. Haddenbach*
102. Guia Prático de Marcenaria, *C. H. Hayward*
103. Andebol, *Fritz e Peter Hattig*
104. Dispositivos Anti-Roubo, *H. Schreiber*
105. Perus, Pintadas e Codornizes, *J. Sauze*
106. Crepes, Doces e Salgados, *F. Arzel*
107. Aperitivos e Entradas, *Myrette Tiano*
108. Ténis de Mesa, *Leslie Woollard*
109. Aprenda Surf, *Rick Abbot e Mike Baker*
110. Futebol - Técnica e Tática, *Kurt Laval*
111. A Vaca Leiteira, *C. T. Whittemore*
112. O Cubo Mágico, *Josef Trajber*
113. O Perdigueiro Português, *José M. Correia*
114. Pizzas e Massas à Italiana, *Marie A. Ränk*
115. O Cubo para quem já o Faz, *Josef Trajber*
116. A Pirâmide Mágica, a Torre e o Barril do Diabo, *M. Mrowka e W. J. Weber*
117. Gansos e Patos, *Marie Mourthe*
118. Iniciação ao Kung-Fu, *A. P. Harrington*
119. Electrónica e Fotografia, *Hanns-Peter Siebert*
120. O Livro da Fortuna, *Douglas Hill*
121. Construção de um Alimentador de Corrente, *Waldemar Baitinger*
122. Hóquei em Patins, *Francisco Velasco*
123. Técnicas de Tiro, *Anton Kovacic*
124. Aprenda a Tricotar, *Uta Mix*
125. ABC da Patinagem, *Christa-Maria e Richard Kerler*
126. A Pesca e os seus Segredos, *Armand Deschamps*
127. O Osciloscópio, *R. Rateau*
128. Guia Prático da Banda do Cidadão, *J. M. Normand*
129. Sumos e Batidos, *Manfred Donderski*
130. Introdução à Programação de Microcomputadores, *Peter C. Sanderson*
131. Aprenda Croché, *Uta Mix*
132. ABC do Microprocessador, *P. Mélusson*
133. Guia Prático de Basic, *Roger Hunt*
134. Introdução à Electrónica Digital, *Ian Sinclair*
135. ABC do Vídeo, *David K. Mathewson*

136. Fotografia em Movimento, *Don Morley*
137. Guia Prático de Cobol, *Ray Welland*
138. Fotografia a Pequena Distância, *Sidney F. Ray*
139. Guia Moderno da Canaricultura, *Manuel Gonçalves*
140. Minieletrónica para Amadores, *Heinz Richter*
141. ABC da Programação de Computadores, *John Shelley*
142. Tarot — O Futuro Pelas Cartas, *Edwin J. Nigg*
143. ABC da Equitação, *Dorothy Johnson*
144. Como Programar o seu ZX 81, *Patrick Gueulle*
145. 100 Avarias TV e a Maneira Prática de as Detectar, *P. Duranton*
146. ABC da Horticultura, *Louis Giordano*
147. Basic Para Microcomputadores, *A. P. Stephenson*
148. Como Programar o Seu ZX Spectrum, *Tim Hartnell e Dilwyn Jones*
149. Iniciação aos Motores Diesel, *David S. Maclean*
150. 60 jogos para o ZX Spectrum, *David Harwod*
151. As Linhas da Mão, *Rose Hubert*
152. Cozinha Italiana, *Rotrand Degner*
153. Manual do ZX Spectrum, *R. J. Simpson e T. J. Terrel*
154. Z80 Assembler para o ZX Spectrum — Iniciação ao Código de Máquina, *João Paulo Fragoso*
155. Aeróbica, *H. Schulz*
156. ABC do Atletismo, *Denis Watts*
157. 26 Programas Basic para Microcomputadores, *Derrick Daines*
158. Aprenda Pascal no seu Microcomputador, *Jeremy Ruston*
159. Guia Moderno da Suinicultura, *Colin Whittemore*
160. O Bar em Sua Casa — 888 Cocktails, *Aladar von Wesendonk*
161. Código de Máquina Para Principiantes, *James Walsh*
162. Código de Máquina Para Programadores Avançados, *Paul Holmes*
163. ABC da Fruticultura, *Henri Gosselin*
164. ABC da Canoagem, *Alan Bye*
165. Guia de Fortran, *Philip Ridler*
166. Manual da Secretária, *Philippa Ramage*
167. ABC das Antenas, *Gordon J. King*
168. Programar Aventuras no Seu Computador, *Andrew Nelson*
169. Guia do Sinclair QL, *Boris Allan*
170. Novas Aventuras no Seu ZX Spectrum, *Peter Shaw e James Mortleman*
171. O Computador no Escritório, *John Shelley*
172. Sobremesas, *Fred Timber*
173. Rádio — Do Circuito Oscilante ao Receptor de Ondas Curtas, *Richard Zierl*
174. Xadrez: ABC das Aberturas, *V. N. Panov*
175. A Construção de Pequenos Transformadores, *M. Dourian e F. Juster*
176. Guia de Pascal, *David Watt*
177. O ZX Spectrum na Educação, *Tim Hartnell, C. Johnson e D. Valentine*
178. Iniciação ao Snooker, *John Pulman*
179. Circuitos com Diacs, Tiristores e Triacs, *Fritz Bergtold*
180. O Jogo do Mah-Jong, *Ursula Eschenbach*
181. Cultura Hidropónica, Culturas sem solo, *Laura Fronty*
182. 49 Jogos Explosivos para o ZX Spectrum, *Tim Hartnell*

IAN R. SINCLAIR

INTRODUÇÃO À ELECTRÓNICA DIGITAL

2.^a edição

EDITORIAL  PRESENÇA

Título Original:

BEGINNER'S GUIDE TO DIGITAL ELECTRONICS

Autor: Ian R. Sinclair

© *Copyright by* Butterworth & Co (Publishers) Ltd. London,
1980

Tradução de Conceição Jardim e Eduardo Nogueira

Capa de Rui Ligeiro

Reservados todos os direitos

para a língua portuguesa à

EDITORIAL PRESENÇA, LDA.

Rua Augusto Gil, 35-A — 1000 LISBOA

SINAIS, COMUTAÇÃO E COMPONENTES

Um sinal electrónico consiste numa tensão variável que pode ser usada para representar alguma qualidade não-electrónica como uma onda sinusoidal, uma imagem, um número, uma velocidade, etc. A forma da onda pode ser indicada num gráfico de tensões em função de tempos; na figura 1.1 mostram-se algumas ondas típicas. Quando uma onda contém uma dada forma sistematicamente repetida, como acontece

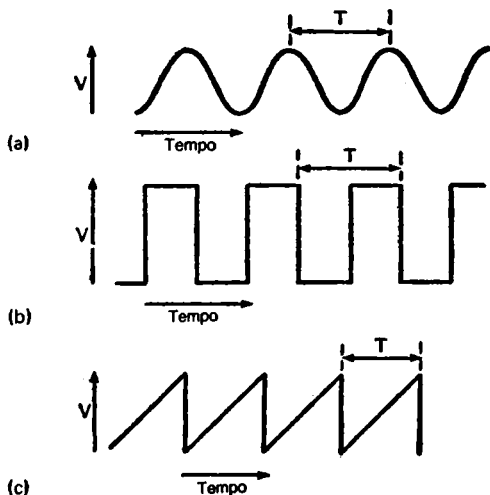


Figura 1.1 — Algumas ondas típicas: (a) sinusoidal; (b) rectangular, ou quadrada; (c) dente de serra. O período T é medido entre dois picos consecutivos da onda.

em todas as ondas desta figura, o tempo necessário para descrever cada uma das variações completas é designado por *período* (T), e o inverso desta quantidade ($1/T$) é designado por *frequência*.

A *amplitude* da onda é a sua medida em volts, e a medição mais útil deste tipo é a amplitude *pico-a-pico*, indicada na figura 1.2. A medição da amplitude de pico e da amplitude r.m.s. tem pouco interesse em electrónica digital.



Figura 1.2 — Amplitude pico-a-pico. Este é o tipo de medição mais útil no caso dos sinais digitais, e deve ser realizada recorrendo a um osciloscópio.

As ondas sinusoidais e as imagens televisionadas podem ser transformadas em ondas cuja forma é muito importante, tão importante como a respectiva frequência ou amplitude. Os sinais deste tipo são designados *analógicos*; a palavra analógico implica a existência de uma correspondência estrita entre a dimensão (amplitude) da onda sonora, por exemplo, e a onda eléctrica por ela produzida. Qualquer operação realizada sobre um sinal analógico que altere a forma da onda está portanto a causar uma distorsão, produzindo a perda de parte da informação veiculada pelo sinal.

Os sinais digitais transportam informações recorrendo a princípios completamente diferentes. Como o nome sugere, os sinais digitais representam números, e constituem uma maneira codificada de transportar informação, tal como o código Morse é um método codificado de comunicar palavras. O sinal digital consiste normalmente num impulso de flancos verticais (figura 1.3), mas a sua forma exacta não é muito importante. As únicas características importantes de uma onda digital são os dois níveis entre os quais varia a tensão. Pode-



Figura 1.3 — Um sinal digital, mostrando os níveis de tensão.

-se-lhes chamar «alto» e «baixo», «ligado» e «desligado», «em condução» ou «cortado», mas são vulgarmente referidos apenas como «1» e «0».

Dado que a forma de um sinal digital não é importante, não são necessários amplificadores lineares para os sinais digitais, e as técnicas de realimentação negativa tão usadas no caso dos amplificadores lineares tornam-se aqui irrelevantes. A amplitude rigorosa do sinal não é importante, desde que os dois níveis de tensão (0 e 1) sejam bastante diferentes — podemos montar os circuitos de tal modo que uma tensão próxima de zero corresponda ao zero e uma tensão próxima do nível 1 seja considerada 1.

Existem porém duas porções do sinal digital que devem possuir uma forma bem determinada: os flancos ascendente e descendente. A onda digital ideal deve ser rectangular, com uma tensão que aumenta instantaneamente de 0 a 1 e diminui de 1 para 0 com a mesma rapidez. Na prática, as transições instantâneas são impossíveis, mas os tempos de crescimento e queda medidos em dezenas de nanosegundos (1 nanosegundo é igual a 10^{-9} segundos, ou seja, um milésimo milionésimo de segundo) já são óptimos para uso prático. Estes crescimentos e quedas rápidas (o tempo de crescimento do sinal é normalmente mais importante) são desejáveis por duas razões, sendo a primeira o facto de os sinais digitais serem muitas vezes usados para medição rigorosa do tempo, e de uma tensão em aumento rápido ser um bom ponto de partida para esta medição; uma tensão em aumento lento introduziria uma incerteza considerável (figura 1.4) numa leitura de tempos. A segunda razão do uso de impulsos de crescimento e queda rápidos consiste no facto de muitos circuitos digitais conterem amplificadores de elevado ganho, que são cortados

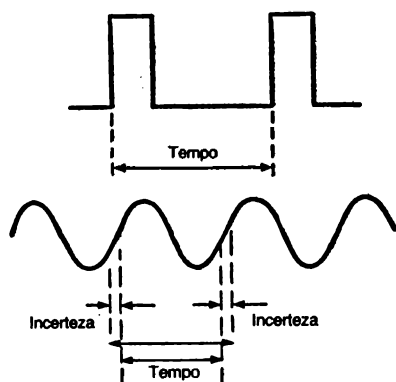


Figura 1.4 — Razão do uso de impulsos de flanco vertical — um intervalo de tempo pode ser medido rigorosamente entre impulsos deste tipo, mas existe uma grande incerteza nesta medição quando se utilizam ondas de variação lenta.

ou colocados em condução total, e que podem oscilar se forem deixados durante algum tempo significativo polarizados entre uma e outra situação, como aconteceria a meio de uma onda de variação lenta (figura 1.5). No que se refere a esta oscila-

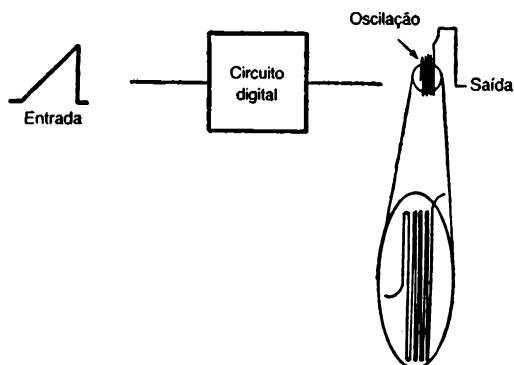


Figura 1.5 — Uma onda de variação lenta aplicada a um circuito integrado pode provocar uma oscilação do sinal de saída.

ção, considera-se um tempo significativo aproximadamente um microsegundo.

Lógica e Contagem

Os circuitos digitais são usados para muitas aplicações, mas os tipos de aplicação classificados como *lógicos* ou de *contagem* são de longe os mais importantes. Os circuitos lógicos utilizam os dois níveis 1 e 0 para indicarem respectivamente «sim» e «não» em resposta a uma pergunta que só pode ter duas respostas possíveis. Qualquer processo que possa ser subdividido numa série de fases, todas as quais possam ser respondidas sim ou não, pode ser realizado pela lógica digital. Por exemplo, consideremos o mecanismo de selecção de moedas numa máquina do tipo «juke-box». O primeiro ensaio será o do diâmetro da moeda — «é ou não demasiado grande?». Se a resposta for o «não», o segundo ensaio pode ser «demasiado pequena ou não?». Dividindo assim este processo em duas fases torna-se possível aplicar uma lógica digital. É do mesmo modo possível realizar ensaios de peso, tipo de metal, tipo de rebordo da moeda, etc.

Qualquer processo lógico pode ser subdividido numa série de fases simples, cada uma das quais terá um resultado «sim» ou «não», representado por um valor 1 ou 0 do sinal de saída. Veremos no capítulo seguinte a maneira como estes sistemas podem ser concebidos e analisados.

Ainda mais importante do que os circuitos lógicos é a aplicação dos métodos digitais de contagem. O uso de dois níveis de tensão significa que apenas é possível representar os números 0 e 1, pelo que se torna necessário utilizar uma escala que apenas contenha estes algarismos; chama-se-lhe *escala binária*.

A escala binária representa números usando apenas os algarismos 0 e 1, da mesma maneira que a escala decimal normal, que utiliza os algarismos 0 a 9. Quando escrevemos um número, por exemplo 684, interpretamo-lo como significando seis centenas mais oito dezenas mais 4 unidades, e a separação das colunas das unidades, dezenas e centenas simplifica as operações de soma, subtracção, multiplicação e

divisão. Tente-se no entanto somar os números romanos CLXII e CCXXIV sem os converter primeiro para a nossa escala decimal...

Na escala decimal cada coluna representa uma potência de 10, pela qual é necessário multiplicar o algarismo existente nessa coluna. A potência é o número que indica o número de vezes em que é necessário multiplicar 10 por si próprio para obter o valor correspondente à coluna. Por exemplo, quando a potência de 10 é 2, a quantidade escrita sob a forma 10^2 é equivalente a 100 (10×10), e a coluna é evidentemente a das centenas. Do mesmo modo, a terceira potência de 10 dá 10^3 ou 1000, e por convenção $10^1 = 10$ e $10^0 = 1$. Podemos agora numerar as colunas em potências de 10 como se faz na figura 1.6, usando 0 para a coluna das unidades, 1 para a das dezenas, 2 para a das centenas, etc. O algarismo existente na coluna 10^0 (unidades) é designado *dígito menos significativo*; o existente na coluna correspondente à maior potência de 10 é designado *dígito mais significativo*. A razão destes nomes é óbvia; um erro de um ou mais dígitos é bastante mais grave nas colunas mais potenciadas do que nas outras.

Número decimal				Número binário				
10^3	10^2	10^1	10^0	2^4	2^3	2^2	2^1	2^0
1000	100	10	unidades	16	8	4	2	1
1	7	4	2	1	0	1	1	0
um milhar				1	dezassexis			
sete centenas				0	oito			
quatro dezenas				1	quatro			
duas unidades				1	dois			
				0	um			

Figura 1.6 — Números decimais («base» 10) e binários («base» 2). Cada algarismo mais à esquerda representa uma maior potência do número de base. A base dos números decimais é 10, e a base dos números binários é 2.

Pode-se conceber uma escala binária exactamente da mesma maneira, usando colunas que representam 2^0 , 2^1 , 2^2 , 2^3 , 2^4 , etc. O algarismo colocado em cada coluna deve ser 0 ou 1, dado que estes são os únicos algarismos disponíveis.

a conversão de um número binário em número decimal é simples: ver a figura 1.7 (a).

Por exemplo, 101101 possui 1 em cada uma das colunas 2^0 , 2^2 , 2^3 e 2^5 ; o valor de 101101 na base decimal deverá portanto ser $2^5 + 2^3 + 2^2 + 2^0$. Ora $2^5 = 32$, $2^3 = 8$, $2^2 = 4$, $2^0 = 1$, pelo que 101101 é equivalente a $32 + 8 + 4 + 1 = 45$ em decimal. A tabela 1.1 mostra os valores das potências de dois até 2^{23} . A conversão de números decimais em binários é feita com grande facilidade utilizando o método indicado na figura 1. = 1.7(b). A figura 1.8 ilustra a maneira de estender estes métodos ao caso das fracções, usando potências negativas para representar quantidades como 0,1 (10^{-1}), 0,01 (10^{-2}) e os seus equivalentes binários, 0,5 (2^{-1}), 0,25 (2^{-2}), etc.

Tabela 1.1 — POTÊNCIAS DE DOIS

2^0	1	2^8	256	2^{16}	65536
2^1	2	2^9	512	2^{17}	131072
2^2	4	2^{10}	1024	2^{18}	262144
2^3	8	2^{11}	2048	2^{19}	524288
2^4	16	2^{12}	4096	2^{20}	1048576
2^5	32	2^{13}	8192	2^{21}	2097152
2^6	64	2^{14}	16384	2^{22}	4194304
2^7	128	2^{15}	32768	2^{23}	8388608

Nota: em linguagem técnica de computadores, 1024 ($=2^{10}$) é conhecido por 1K, tal como o número 1000 é indicado por 1 k em electrónica. 8K é portanto igual a 8192 e não a 8000, por exemplo.

Quando um número é escrito em escala binária, cada dígito é chamado *bit* (de *binary digit*, dígito binário). Os grupos de oito bits são muito usados em contagem, particularmente no caso dos microprocessadores, e são, portanto, designados por uma expressão específica: *byte*.

O bit que representa 2^0 (0 ou 1) é designado *bit menos significativo* (em inglês, LSB), e o que representa 2^7 (128) é chamado *bit mais significativo* (MSB), em cada byte.

O código binário simples não constitui a única maneira possível de utilizar os dígitos 0 e 1 para representar números. O código Gray é também usado, em dispositivos que convertam quantidades analógicas em sinais digitais porque está mais livre de erros. Numa contagem feita de acordo com este método, só se altera um dígito quando a contagem é aumentada de 1 (tabela 1.2), mas como a aritmética praticada no código Gray é difícil, os números usados neste código são sempre convertidos para a base binária para efeitos de processamento.

Fracções

Fracção decimal 0,1111:

$$\begin{array}{cccc} 10^{-1} & 10^{-2} & 10^{-3} & 10^{-4} \\ 1 & 1 & 1 & 1 \end{array}$$

- = uma dezena
- + uma centena
- + um milhar
- + uma dezena de milhar

Fracção binária 0,1011:

$$\begin{array}{cccc} 0,5 & 0,25 & 0,125 & 0,0625 \\ 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \end{array}$$

- = $1 \times 0,5$
- + $0 \times 0,25$
- + $1 \times 0,125$
- + $1 \times 0,0625$

$$= 0.6875 \text{ (em decimal)}$$

Conversão de fracções decimais em binárias

Regras: Multiplicar por dois. Contar 1 no lado esquerdo da vírgula decimal como 1 binário, contar um 0 ou um 2 como um 0 binário. Continuar enquanto for necessário —

muitas fracções binárias nunca dão um valor rigoroso. Ler o número binário de cima para baixo.

Exemplo		0.624	Decimal
		$\times 2$	
1	←	1.248	
		248	
		$\times 2$	
0	←	0.496	
		$\times 2$	
0	←	0.992	
		$\times 2$	
1	←	1.984	
		.984	
		$\times 2$	
1	←	1.968	
		.968	
		$\times 2$	
1	←	1.936	

A fracção binária é 0,100111 com seis casas (de facto equivalente a 0,596 em decimal).

Figura 1.8. — Fracções — uma fracção binária utiliza potências negativas de 2 a seguir à vírgula, tal como a fracção decimal utiliza potências negativas de 10. A conversão de fracções decimais em binárias raramente é exacta.

Os sistemas decimais codificados em binário (em inglês, BCD), utilizam quatro dígitos binários para representarem cada dígito de um número decimal. Isto pode ser feito utilizando o código 8-4-2-1, normalmente designado BCD, no qual cada dígito decimal é convertido no seu equivalente binário de quatro dígitos, como acontece na figura 1.9; um outro sistema por vezes usado é o *Excess-3*, no qual se soma 3 a cada dígito antes da conversão, como se mostra na figura 1.10. A vantagem do código *Excess-3* relativamente ao sistema

8-4-2-1 consiste em tornar possível realizar com maior facilidade as operações de adição e subtração.

Table 1.2 — CÓDIGO GRAY

<i>Decimal</i>	<i>Código Gray</i>	<i>Decimal</i>	<i>Código Gray</i>
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100		

NOTA: A vantagem do código Gray consiste em que apenas se altera um dígito para cada aumento de uma unidade de contagem. Isto evita erros quando são executados certos tipos de conversões analógico-digital. O código Gray deve ser convertido em binário para realização da operação aritmética, existindo circuitos integrados capazes de realizarem esta conversão.

Número decimal: 167

Em binário, $\left. \begin{array}{l} 1 \text{ é } 0001 \\ 6 \text{ é } 0110 \\ 7 \text{ é } 0111 \end{array} \right\}$ usando 4 bits por algarismo

167 é 000101100111 em BCD

Nota: Em binário, 167 é 10100111

Figura 1.9. — O código BCD 8-4-2-1 utiliza quatro bits para representar cada algarismo de um número decimal. Isto é particularmente útil para apresentação de números decimais em displays (ver capítulo 6).

Número decimal: 257

Acrescentar 3 a cada:	5	8	10
Passar a binário de 4 dígitos:	0101	1000	1010

O número em excesso-3 é. 010110001010

Figura 1.10. — O código Excess-3.

ARITMÉTICA BINÁRIA SIMPLES

As operações de aritmética binária podem ser realizadas, no papel, exactamente da mesma maneira que as da aritmética decimal. A adição e a subtração são ilustradas na figura 1.11, usando o transporte quando a soma excede 1 ou quando se subtrai 1 a zero. A multiplicação e a divisão são realizadas da maneira indicada na figura 1.12. Estes métodos não são particularmente fáceis de aplicar em circuitos digitais. Os circuitos que realizam a adição serão tratados mais adiante, e uma simples modificação aritmética torna a subtração possível sem utilizar um circuito diferente. O método usado é designado *subtração em complemento para 2* e consiste no processo seguinte:

- (1) — O complemento para um do número que deve ser subtraído é obtido trocando os uns pelos zeros e os zeros pelos uns (figura 1.13).
- (2) — Soma-se outro 1 ao bit menos significativo.
- (3) — Soma-se agora o complemento para 2 ao outro número. Utilizando este método, o mesmo circuito que realiza a adição realiza simultaneamente a subtração, desde que seja possível obter o complemento para 2 do número que se deseja subtrair.

Adição

$$\begin{array}{r}
 111111 \leftarrow \text{transporte} \\
 1001101 \\
 0111011 \\
 \hline
 = 10001000
 \end{array}$$

Regras

$$\begin{aligned}
 0 + 0 &= 0 \\
 1 + 0 &= 1 \\
 1 + 1 &= 0 \text{ transporte } 1 \\
 \text{Transporte } 1 + 1 + 1 &= \\
 &= 1, \text{ transporte } 1
 \end{aligned}$$

Subtração

$$\begin{array}{r}
 110011010 \\
 - 011010101 \\
 \hline
 = 011000101
 \end{array}$$

Regras

$$\begin{aligned}
 0 - 0 &= 0 \\
 1 - 0 &= 1 \\
 1 - 1 &= 0 \\
 0 - 1 &= 1 \text{ transporte } 1 \\
 1 - 1 - \text{transporte } 1 &= \\
 &= 1 \text{ transporte } 1
 \end{aligned}$$

Figura 1.11. — Adição e subtração de números binários.

Multiplicação

$$\begin{array}{r}
 \text{(a)} \quad 110011 \\
 \text{(b)} \quad \times 101 \\
 \hline
 110011 \\
 000000 \\
 110011 \\
 \hline
 = 1111111
 \end{array}$$

Escrever a linha (a) para cada 1 na linha (b), mas deslocado do mesmo número de casas, e em seguida adicionar as linhas

Divisão

$$\begin{array}{r}
 \underline{1001} \\
 110)110110 \\
 \underline{110} \\
 \dots \\
 \underline{110} \\
 \underline{110}
 \end{array}$$

Utilizar o mesmo método da divisão decimal

Figura 1.12. — Multiplicação e divisão de números binários.

Número:	1 0 0 1 0 1 1 1
Complemento para 1	0 1 1 0 1 0 0 0
Complemento para 2	$\begin{array}{r} \\ + 1 \\ \hline 0 1 1 0 1 0 0 1 \end{array}$

Subtração usando
o complemento para 2

$$\begin{array}{r} 1\ 1\ 0\ 0\ 1\ 0\ 0\ (a) \\ -\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ (b) \end{array}$$

Nota:

obter o complemento
para 2 de (b)

Para	1 1 0 0 0 1 1 0 1
—	1 0 1 1 1

$$-\ 1\ 0\ 0\ 1\ 1\ 0\ 1$$

escrever como	1 1 0 0 0 1 1 0 1
—	0 0 0 0 1 0 1 1 1

e acrescentar a (a)

que é	1 1 0 0 0 1 1 0 1
	+ 1 1 1 1 0 1 0 0 1
	(1) <u>1 0 1 1 1 0 1 1 0</u>

$$\begin{array}{r} 1\ 1\ 0\ 0\ 1\ 0\ 0 \\ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1 \end{array}$$

Resposta:	<u>1 0 1 1 1 0 1 1 0</u>
-----------	--------------------------

eliminar o transporte final
obtendo

$$\underline{0\ 1\ 1\ 0\ 0\ 0\ 1}$$

Figura 1.13. — Aritmética de complemento para dois. O número a subtrair é «complementado», sendo o complemento somado em seguida ao outro número. Ambos os números devem conter o mesmo número de dígitos (acrescentar zeros antes do primeiro algarismo de um número em caso de necessidade).

Circuitos de comutação digitais

O circuito de comutação é aquele cuja tensão de saída varia rapidamente de um extremo ao outro quando é aplicado um sinal digital a uma entrada. Os circuitos digitais utilizam a comutação da maneira que os circuitos lineares utilizam a amplificação, com a importante diferença de as amplitudes do

sinal de entrada e de saída de um circuito de comutação serem muitas vezes idênticas. Uma outra diferença consiste em o circuito de comutação ao actuar de modo a «definir» melhor os bordos dianteiro e traseiro de um impulso que foi amortecido (integrado) num cabo (figura 1.14).



Figura 1.14. — Uso de um circuito de comutação para dar maior «definição» a um impulso.

Na figura 1.15 mostra-se um circuito de comutação de tensão bastante simples. O transistor é de tipo bipolar (neste caso, NPN), e utiliza uma resistência de carga de 330 ohms no seu circuito de colector. O circuito de base não é polarizado, encontrando-se ligado à entrada através de uma resistência de 4,7 kohms, R1, que limita a quantidade de corrente que pode passar no circuito de base. Considera-se uma alimentação de 5,0 V, dado que é esta a tensão normal em muitos circuitos digitais.

No caso de uma tensão de entrada inferior a cerca de 0,5 V, o transistor mantém-se cortado, pelo que a tensão de colector, que é a tensão de saída do circuito, se mantém elevada. Qualquer tensão entre 0 e 0,5 V contará portanto

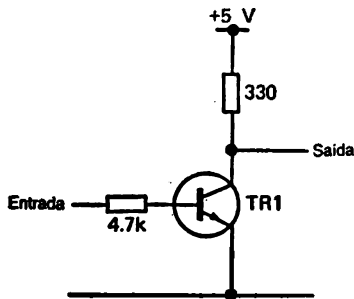


Figura 1.15. — Um circuito de comutação simples

como uma entrada 0: Quando a tensão de base excede os 0,5 V, o transistor é passado à condução; mas o sinal digital aplicado à entrada não se manterá a 0,5 V, antes passando de maneira abrupta de cerca de 0 V para aproximadamente + 5,0 V. Com + 5,0 V na entrada, a resistência de entrada de 4,7 kohms, juntamente com a resistência de entrada da base do transistor (talvez de algumas centenas de ohms) permitirá a passagem de cerca de 1 mA para a base (utilizando a lei de Ohm e considerando uma resistência total de 5,0 kohms).

Ora a quantidade máxima de corrente que pode passar no circuito de colector é limitada pela tensão de alimentação e pela resistência de colector. Se considerarmos que a tensão do colector do transistor pode diminuir para 0 V, então a corrente máxima de colector será de $5,0/0,33 = 15,15$ mA. Para produzir esta quantidade de corrente no circuito de colector com 1 mA passando no circuito de base é necessário um transistor com um ganho de corrente de pelo menos 15, um valor que facilmente é excedido. Uma entrada normal ao nível 1 assegura portanto a saturação do transistor — que deixa passar tanta corrente quanto pode no seu circuito de colector, pelo que a tensão de colector diminui para um valor bastante reduzido, cerca de 0,2 V. A corrente em excesso que passa no circuito de base assegura que o transistor satura mesmo que a tensão correspondentemente ao nível 1 seja inferior aos 5,0 V anteriormente considerados.

A comutação pode também ser realizada por um MOSFET, como se indica na figura 1.16. Não é necessária

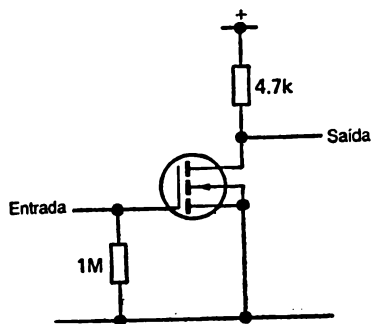


Figura 1.16. — Um circuito de comutação MOS simples.

neste caso qualquer resistência de limitação, porque não passa corrente no circuito de porta, mas deve-se ligar uma resistência ou diodo entre a gate e a massa de modo a impedir a gate de ser afectada por tensões electrostáticas parasitas. A quantidade de corrente que passa no circuito drain-fonte é limitada uma vez mais pela resistência de carga, que é normalmente de vários kohms. Uma pequena variação da tensão de entrada levará novamente a saída a variar entre 1 e 0. O circuito é concebido de tal modo que qualquer tensão inferior a + 1,0 V conte como um zero e qualquer tensão superior a 10,5 V (considerando uma alimentação de 12,0 V) como um 1.

Os circuitos das figuras 1.15 e 1.16 mostram, por uma questão de simplicidade, um circuito de comutação com um único andar. Os circuitos digitais, no entanto, utilizam geralmente circuitos de comutação com vários andares, e possuem um ganho bastante superior ao de um único andar. As razões para uso de vários andares são as seguintes:

- (1) — Um ganho elevado assegurará uma comutação rápida;
- (2) — Um ganho elevado assegurará que a comutação é completa para apenas uma pequena variação de tensão na entrada.
- (3) — O andar de saída do circuito de comutação pode ser montado de modo a fornecer uma potência suficiente para pilotar outros circuitos de comutação.

Considerando estas razões mais pormenorizadamente, observa-se na figura 1.17 um gráfico entrada/saída típico para um circuito de comutação utilizando uma alimentação de 5 V.

Imagine-se um sinal de entrada que consista numa tensão aumentando de 0 V para + 5 V em 1 μ s, como se indica. O tempo de estabelecimento desta tensão é definido como o tempo necessário para que a tensão passe de 10 % a 90 % do seu valor final (neste exemplo, 5,0 V). Este tempo é por-

tanto o necessário para passar de 0,5 V para 4,5 V, ou seja, aumentar 4 V, e é de $4/5 \mu\text{s}$, ou seja, $0,8 \mu\text{s}$. Ora o gráfico entrada/saída mostra que a saída varia de 4,5 V para 0,5 V para uma variação de entrada de 0,6 a 0,7 V. Quanto tempo demora? Se a entrada varia de 5 V em $1 \mu\text{s}$, variará portanto de 0,1 V ($0,7 - 0,6$ V) em $(1,0 \times 0,1)/5,0 \mu\text{s}$, ou seja, $0,02 \mu\text{s}$. É de esperar que a variação na saída de 4,5 V para 0,5 V demore também este tempo, pelo que o tempo de queda na saída é muito mais rápido do que o tempo de crescimento na entrada.

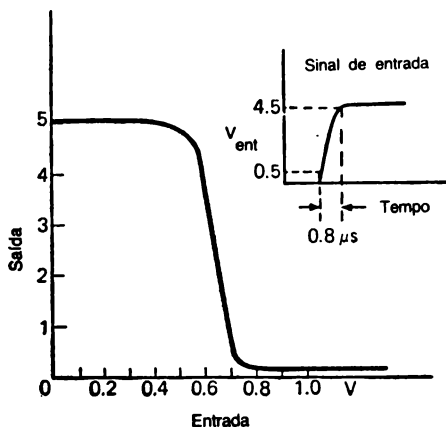


Figura 1.17. — Gráfico da saída em função da entrada num circuito de comutação, e maneira como este produz uma maior definição do sinal.

Isto só é no entanto verdade se a saída do circuito de comutação pode carregar ou descarregar a esta velocidade as capacitâncias parasitas existentes no circuito. Por outro lado, este circuito de comutação pode ser obrigado a comutar outros circuitos de comutação, cada um dos quais terá a sua capacitância parasita e necessitará também de alguma corrente de entrada.

Em tempos, os circuitos de comutação para sistemas digitais utilizavam transistores ligados separadamente, uma montagem designada por *circuito de componentes discretos*. No entanto, à medida que os circuitos digitais demonstraram a sua utilidade e começaram a ser usados em número cada vez maior, o número de transistores, e pior ainda o de juntas soldadas existentes num circuito digital, tornou-se excessivo. O uso de circuitos integrados constitui uma resposta para ambos os problemas. Um *circuito integrado* é um circuito completo, com transistores e resistências, fabricado numa pequena pastilha de silício, usando os mesmos métodos utilizados no fabrico de transistores discretos. Os primeiros circuitos integrados continham apenas alguns transistores e resistências, sendo conhecidos por circuitos SSI (Small Scale Integration, integração em pequena escala); mas à medida que a tecnologia se desenvolveu, depressa se tornou possível construir circuitos integrados contendo o equivalente a 50 ou mais transistores. Estes circuitos integrados tornaram-se conhecidos sob a designação MSI (Medium Scale Integration, integração em média escala). Mais tarde descobriu-se que os circuitos MOSFET podiam também ser fabricados como circuitos integrados, sendo possível concentrar muitos FETs numa única pastilha. Estes circuitos LSI (Large Scale Integration, integração em larga escala) podem conter milhares de transistores e resistências montados em circuitos cuja reprodução seria demasiado cara e complexa utilizando os métodos de construção convencionais.

A existência de circuitos integrados conduziu a um uso muito maior de circuitos digitais em aplicações que anteriormente utilizavam circuitos lineares ou métodos não electrónicos. Como o circuito integrado é produzido recorrendo a uma sequência de operações semelhantes à usada no fabrico de transistores, torna-se tão simples e fácil fabricar o circuito, mesmo no caso da integração LSI, como produzir um transistor. O trabalho de concepção do circuito integrado é evidentemente muito mais caro, mas este maior custo pode ser contrabalançado se forem fabricadas e vendidas grandes quantidades de circuitos. A fiabilidade do circuito integrado pode ser verificada, e o seu uso em vez dos componentes dis-

cretos que substitui permite a realização de um menor número de ligações. Mas mais importante ainda, as dimensões de um circuito podem assim ser bastante reduzidas, de tal modo que é hoje possível conceber equipamento digital que pode ser montado no interior de outros componentes. Por exemplo, os circuitos necessários para converter um receptor de televisão num computador miniatura podem ser facilmente incorporados no interior da sua caixa. Antes da integração LSI, os circuitos necessários para o computador ocupariam grande parte de uma sala de estar.

Não considerando algumas excepções, os circuitos integrados para uso em circuitos digitais são de dois tipos: TTL e MOS. Os tipos TTL utilizam transistores convencionais (bipolares) em forma integrada, normalmente com modelos de canal P e canal N em cada pastilha semicondutora. Se bem que ambas as «famílias» contêm funções de circuito semelhantes, as diferenças entre os dois tipos são importantes pelo que convirá analisá-las mais detalhadamente.

Circuitos integrados digitais TTL

O nome «TTL» significa Lógica transistor-transistor, em inglês; este esquema substituiu a anterior lógica Díodo-Transistor (DTL) e a lógica RTL (resistência-transistor). Os circuitos TTL, cujo melhor exemplo é a série 74 fabricada pela Texas Instruments, são circuitos MSI que utilizam estruturas de transistor NPN. A tensão de funcionamento é 5.0 V, e a concepção dos circuitos é tal que esta tensão não pode ser excedida em caso algum.

Na figura 1.18 mostra-se um típico andar de entrada TTL. O transistor de entrada possui a sua base ligada a uma resistência que por sua vez está ligada à alimentação de +5,0 V. A entrada do circuito TTL é feita ao emissor deste transistor, não à base comum. Quando a tensão de entrada no emissor é elevada, entre 4,5 e 5,0 V, o primeiro transistor deixa de conduzir porque a tensão entre a base e o emissor não é suficientemente elevada. Não passará qualquer corrente entre o colector e o emissor.

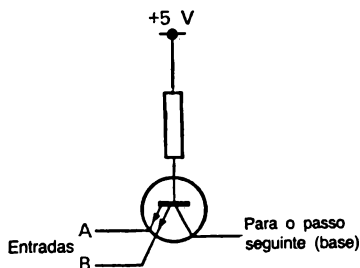


Figura 1.18 — A entrada de um circuito TTL típico segue para o emissor de um transistor, cuja base é devolvida através de uma resistência, para a alimentação de voltagem positiva.

Quando a tensão no emissor do primeiro transistor é baixa, perto do 0 volts, a corrente passará entre a base e o emissor. Na série «standard» de circuitos TTL, é determinado um valor de 1,6 mA para esta corrente pela resistência que liga a base à linha de +5,0 V. Esta corrente é suficiente para saturar o primeiro transistor, o que significa que o trajecto colector-emissor apresenta uma baixa resistência, tão baixa quanto é possível obter-se. Dado que o emissor deste transistor se encontra a 0 V, portanto, a tensão de colector também será reduzida, não mais do que 0,2 V acima da tensão de emissor.

Devido à sua construção, exemplificada no capítulo 2, a entrada de qualquer circuito TTL deve ser pilotada por uma fonte de baixa impedância, capaz de passar 1,6 mA a baixa tensão. Imaginemos por exemplo um circuito TTL pilotado por um seguidor de emissor (figura 1.19). Com o seguidor de emissor polarizado para o seu estado condutor, a entrada TTL estará ao nível lógico 1, mas o corte do seguidor de emissor não cortará necessariamente o circuito TTL. A razão disto é que a resistência R_k pode ter um valor demasiado grande para permitir a passagem de 1,6 mA. Por exemplo, com $R_k = 1 \text{ kohm}$, 1,6 mA provocarão uma queda de 1,6 V, o que pode não ser suficientemente reduzido para deixar o circuito TTL comutar. As entradas TTL devem portanto ser

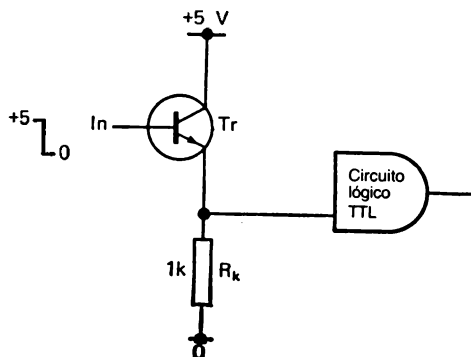


Figura 1.19 — Pilotagem de um circuito TTL a partir de um seguidor de emissor — a menos que R_k seja pequena o circuito TTL não será cortado na entrada. São necessários valores de 470 ohms ou menos para R_k

pilotadas por circuitos que deixem passar correntes de pelo menos 1,6 mA para a massa com uma queda de tensão bastante reduzida. Um circuito apropriado será, por exemplo, o circuito em emissor comum da figura 1.20. Quando a tensão

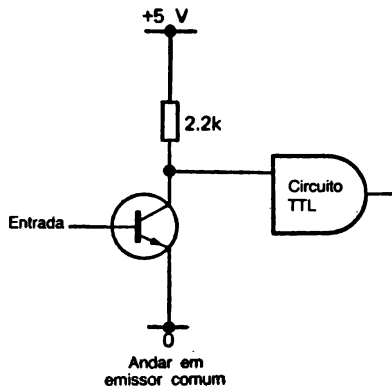


Figura 1.20 — Direcçãoamento de um circuito TTL a partir de um amplificador emissor comum — modo mais satisfatório de utilizar um transistor para direccionar circuitos TTL.

de base do transistor é elevada para cerca de 0,6 V, o transistor satura, pelo que a resistência entre o colector e o emissor é bastante reduzida, e a entrada TTL é mantida a cerca de +0,2 V. Este valor é de facto suficientemente reduzido para assegurar a comutação da entrada TTL para 0. Note-se a propósito que no caso de uma entrada TTL não estar ligada, «flutuará» para um sinal de alta tensão permanente, ou seja, de nível 1. Não se deve porém confiar nesse método de obtenção do nível lógico um na entrada a menos que os sinais digitais sejam de reduzida frequência (100 Hz ou menos). A altas frequências o acoplamento capacitivo entre uma entrada e as outras pode ser suficiente para provocar sinais indesejados de entrada a um perne não ligado, pelo que as entradas não utilizadas devem ser ligadas através de uma resistência de 1 kohm a +5 V, ou directamente à massa se é necessária uma entrada 0.

O tipo de andar de saída utilizado nos circuitos TTL é ilustrado na figura 1.21. Dois transístores NPN encontram-se ligados em série e o terminal de saída está ligado ao ponto

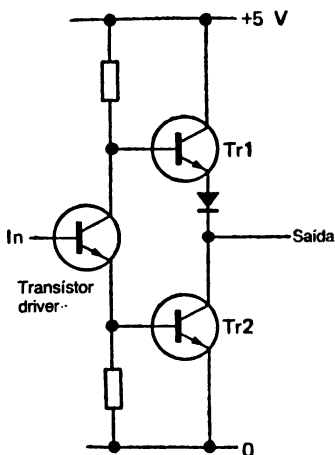


Figura 1.21 — Um circuito de saída TTL típico. Alguns circuitos TTL omitem Tr1, constituindo aquilo que se designa andar de saída em «colector aberto» (ver o capítulo 6).

onde o emissor de um transistor se junta ao colector do outro. Para comutar esta saída para o nível lógico 1, a tensão de base do transistor Tr1 é elevada, +5,0 V, e a tensão de base do transistor Tr2 é simultaneamente reduzida, ligada a 0 V. O terminal de saída possui agora uma baixa resistência à linha de alimentação de +5,0 V, pelo que pode passar corrente através de uma carga ligada entre o terminal de saída e a massa. Nestas condições, o terminal de saída pode actuar como uma fonte de corrente.

Quando o circuito comuta, Tr1 é cortado, ficando com uma tensão de base reduzida, e Tr2 é saturado, permanecendo a sua base a uma tensão de cerca de 0,6 V. Neste estado o terminal de saída encontra-se a baixa tensão — valor lógico 0 — podendo passar corrente para o terminal de saída a partir de uma carga ligada à linha de +5,0 V. Nestas condições, portanto, o terminal de saída pode actuar com um dissipador de corrente.

Em qualquer dos estados lógicos, uma corrente que entre ou saia no terminal de saída provoca uma variação bastante pequena da tensão neste terminal. No caso de grande parte dos circuitos TTL, a corrente máxima que sai ou entra no terminal de saída é de exactamente 16mA. Em linguagem digital, o andar de saída pode alimentar ou dissipar um máximo de 16mA. A importância disto reside em que a baixa impedância e a capacidade de tratamento de corrente deste andar de saída nos permite utilizar a saída TTL para pilotar outros circuitos. Os relés de baixa corrente, por exemplo, podem ser pilotados directamente por uma saída TTL (sujeita à habitual protecção por diodo contra um pico de tensão que possa ocorrer quando é cortada a passagem de corrente por um relé), sendo também possível aplicar esta pilotagem a cargas mais reduzidas, como os indicadores LED («diodo emissor de luz»).

Mais importante ainda, a capacidade de dissipação de corrente manifestada por um andar de saída permite que a saída TTL pilote até 10 entradas TTL, cada uma das quais necessita de 1,6 mA para manter a tensão ao nível lógico 0. D'z-se então que o andar de saída TTL possui um «fanout» (em tradução livre, «leque de distribuição») de 10.

Os circuitos TTL são usados em computadores grandes devido a uma sua importante vantagem — velocidades de funcionamento muito grandes. O tempo necessário para um circuito TTL comutar de 0 para 1 ou de 1 para 0 é de aproximadamente 10ns, conforme o tipo de circuito, e é possível obter velocidades ainda superiores a esta usando a série 74H de circuitos TTL. O preço a pagar por este funcionamento a alta velocidade é uma considerável dissipação de energia, pelo que os circuitos TTL necessitam de alimentações de 5,0 V que passem bastante corrente. O uso de uma concepção modificada (Schottky) da lógica TTL, no entanto, permite o uso de menores correntes de funcionamento, e a série de 74LS de circuitos TTL (LS significa Schottky de baixa potência) pode também conseguir altas velocidades de funcionamento porque não se deixa os transístores saturarem. O que permite isto é a construção numa forma integrada de um díodo de barreira Schottky, colocado entre a base e o colector de cada transístor. Um díodo Schottky utiliza uma junção alumínio-silício, e possui uma tensão directa muito reduzida quando está em con-

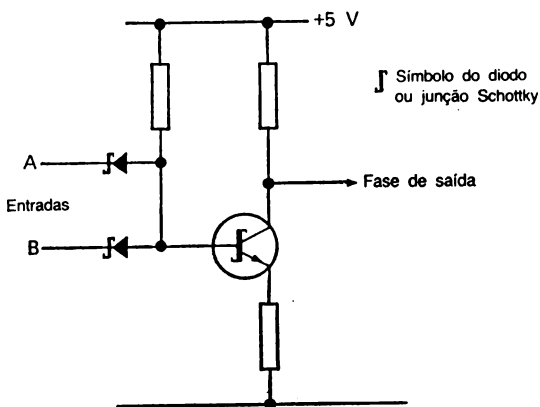


Figura 1.22 — Um andar típico de entrada de baixa potência Schottky. A comutação lógica é feita pelos díodos, e o transístor, que também utiliza funções Schottky, produz a amplificação de potência. A velocidade de comutação pode ser bastante elevada, com uma passagem de corrente bastante reduzida.

dução, da ordem dos 0,3 V, pelo que pode ser ligado entre a base e o colector de um transistor impedindo a sua tensão de colector de diminuir para menos de 0,3 V relativamente à tensão de base. Deste modo, quando o transistor é colocado em condução, passa uma corrente excessiva, através do díodo Schottky, da base para o colector (e portanto para a massa), em vez de ser produzida a saturação. Os circuitos integrados TTL Schottky de baixa potência dissipam vulgarmente apenas um quinto de potência requerida pelos circuitos TTL normais; alternativamente, o uso de circuitos Schottky funcionando a maiores níveis de potência permite tempos de comutação de cerca de 3 ns.

Circuitos CMOS

A integração em larga escala requer circuitos que dissipem quantidades de energia muito reduzidas, dado que é necessário acomodar milhares de transistores numa mesma pastilha semicondutora. O uso de um fabrico MOS permite a produção de circuitos comutadores utilizando MOSFETs de canal P e N muito fracos, conhecidos pelo nome de MOS complementares (CMOS) e concebidos pela RCA, que deram origem a uma nova família de circuitos lógicos.

Na figura 1.23 mostra-se o circuito típico de um circuito integrado digital MOS. Na entrada encontram-se ligados dois FETs (transistores de efeito de campo) em série, um de canal P e outro de canal N. As portas estão ligadas entre si, de tal modo que é aplicado um mesmo sinal a ambas, existindo uma estrutura de díodos e resistências que protege o delgado isolamento das gates contra uma tensão excessiva de entrada. A acção do circuito é tal que uma tensão positiva na entrada permitirá a passagem de corrente no EFT de canal N, realizando assim uma ligação de baixa resistência entre o terminal de saída e a massa. Pelo contrário, uma tensão 0 na entrada permitirá a passagem de corrente no FET de canal P, realizando assim uma ligação de baixa resistência entre o terminal de saída e a linha de alimentação positiva. O terminal de saída é ligado a ambos os drains dos FETs pelo que estes actuam

como amplificadores de tensão em cada sentido. Diversos circuitos utilizam apenas um conjunto de FETs sem saturação nos andares de entrada e saída, de tal modo que os circuitos internos são mais simples do que os dos circuitos TTL. O tempo de operação é superior ao TTL, normalmente de 35 a 100 ns, mas a dissipação de potência é negligenciável.

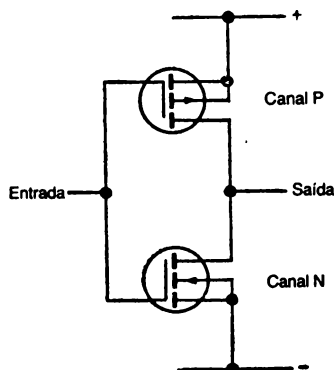


Figura 1.23 — Um circuito inversor CMOS típico, omitindo os diodos de protecção de gate.

Além disto, a entrada de gate não recebe qualquer corrente mensurável tanto à tensão de nível lógico 1 como à de nível lógico 0, pelo que o «fanout» que pode ser obtido é apenas limitado pela capacitância parasita do número de andares ligados. Muitos circuitos CMOS têm um «fanout» de cerca de 50. Por outro lado, a capacidade de dissipação de corrente do andar de saída é limitada a cerca de 0,5 mA, pelo que se tornam necessários circuitos de interface se se deseja pilotar quaisquer cargas que não sejam apenas circuitos CMOS. Mesmo um indicador LED, por exemplo, pode ser pilotado por um seguidor de emissor cuja base seja pilotada pelo circuito CMOS.

Os circuitos digitais CMOS podem ser utilizados com alimentações desde +3,0 V até +12,0 V, pelo que os 5,0 V dos circuitos TTL são compatíveis com a sua utilização. Além

disso, o baixo consumo dos circuitos CMOS permitem um funcionamento com pilhas, tornando possível o uso de pilhas de 6,0 V ou 9,0 V, tão utilizadas em rádio.

Para a maior parte das utilizações, incluindo computadores grandes, os circuitos CMOS são suficientemente rápidos e a sua tolerância de diferentes tensões de alimentação torna-os úteis para uma grande variedade de circuitos. Para uso experimental, no entanto, os circuitos CMOS sofrem da desvantagem de serem facilmente deteriorados por tensões parasitas devido à elevada resistência dos terminais de gate. A tensão electrostática nos isoladores (incluindo os painéis de circuito impresso) ou das mãos humanas pode ser suficiente para deteriorar as gates CMOS, e os circuitos de protecção só são eficazes depois de o circuito integrado ter sido ligado em circuito. As precauções mais importantes serão:

- 1) — Montar os circuitos integrados apenas depois de todos os outros componentes terem sido montados. Todos os pernes devem ser ligados.
- 2) — Utilizar suportes de circuito integrado sempre que possível.
- 3) — Utilizar ferros de soldar cuja estrutura metálica tenha sido ligada à terra, e soldar primeiramente os pernes de massa e alimentação do circuito integrado.
- 4) — Manter os circuitos integrados na sua cápsula protectora até estarem montados.
- 5) — Evitar tocar com as mãos nos pernes do circuito integrado.
- 6) — Tratar um painel de circuito impresso tão cuidadosamente como o próprio circuito integrado até à linha de massa se encontrar definitivamente ligada.

Depois de montado, um circuito integrado CMOS é electricamente tão robusto como qualquer outro circuito integrado; o único período vulnerável é aquele entre a remoção do circuito integrado da sua cápsula protectora em plástico e a sua ligação ao circuito. Por esta razão, no entanto, os circuitos

integrados TTL são preferíveis para painéis de ligações em que os pernos podem ficar em circuito aberto durante as mudanças de ligação.

Funcionamento prático

Como em qualquer outro ramo da electrónica, o ramo digital obriga a alguma experiência prática para se possuir uma compreensão exacta do que se passa. Os circuitos digitais ilustrados neste livro podem ser verificados na prática utilizando circuitos integrados TTL ou CMOS, mas o uso dos TTL será mais conveniente na maior parte dos casos desde que se disponha de uma fonte de alimentação de 5,0 V estabilizada. Se só se dispõe de alimentação de 9,0 V, por exemplo de pilhas, podem-se usar circuitos CMOS, mas sujeitos às precauções de manuseamento já citadas.

Uma vantagem considerável do trabalho com circuitos digitais consiste na possibilidade de funcionamento a velocidade muito baixa, de tal modo que os níveis de sinal podem ser estudados sem recorrer ao auxílio de um cinescópio. Dado que os níveis de sinal são 1 ou 0, o instrumento de diagnóstico mais usado é simplesmente um LED, que acende quando ligado a 1 e não acende quando ligado a 0. Para evitar uma corrente excessiva, é necessário ligar uma resistência em série quando o LED é usado em circuitos TTL (figura 1.24), usan-

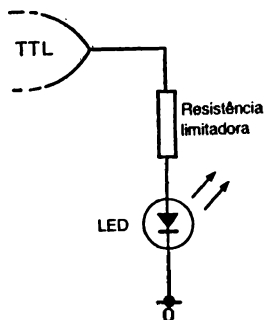


Figura 1.24 — Um indicador LED utilizado com circuitos TTL.

do-se uma combinação de transistor e resistência no caso dos circuitos CMOS (figura 1.25).

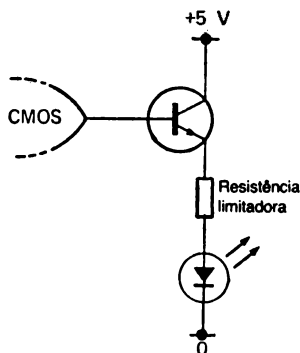


Figura 1.25 — Um indicador LED utilizado num circuito CMOS.

A fim de evitar soldaduras constantes, recomenda-se uma construção que permita evitá-las. Existem no comércio vários tipos de painéis que utilizam formas de montagem diferentes, não recorrendo à soldadura, sendo o mais barato o «Euro-breadboard» ilustrado na figura 1.26. — Possui quatro colunas de contactos, cada uma delas consistindo em 25 linhas de cinco contactos cada. Todos os contactos de uma linha se encontram de tal modo que tornam possível a montagem de qualquer cápsula de circuito integrado digital. A forma compacta do painel elimina a necessidade de longos cabos de ligação quando se ligam dois circuitos integrados. Por uma questão de facilidade de construção as colunas e linhas são numeradas e letradas, pelo que todas as ligações podem ser feitas por referência a este sistema de identificação antes de os circuitos integrados serem montados. Torna-se assim possível o uso de circuitos integrados CMOS.

A melhor técnica consiste em escrever os números (correspondentes às filas) e as letras (correspondentes às colunas)

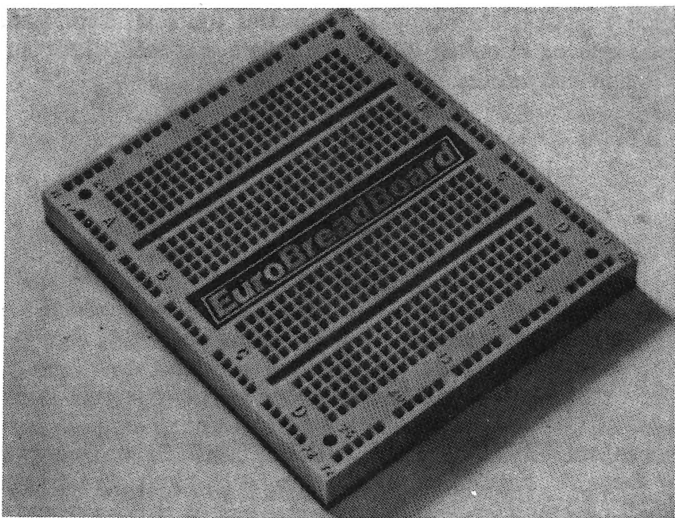


Figura 1.26 — O «Eurobreadboard». Constitui um painel de montagem ideal para experimentação de circuitos integrados digitais dado que torna a montagem muito rápida e todos os componentes podem ser facilmente montados e removidos.

no diagrama do circuito (como se mostra na figura 1.27), de tal modo que se torne possível montar todo o circuito por referência a estas marcações.

Se se utilizarem circuitos integrados da mesma maneira, por exemplo um circuito integrado com cápsula de 16 pernas montando o perne 1 no contacto A1, a construção pode tornar-se muito rápida, tal como de resto as modificações que se deseje fazer ao circuito. A propósito, pode ser muito útil a seguinte sugestão: ligue resistências com o valor de 1 Mohm entre todos os circuitos CMOS e a massa. Se estas resistências não forem afectadas, é possível introduzir modificações ao circuito sem quaisquer riscos para o circuito integrado CMOS e sem necessidade de o remover quando se fazem essas modificações. Note-se ainda que, quando se usam circuitos integrados do tipo TTL, as entradas que se devem manter ao nível

lógico 1 podem ser mantidas flutuantes (ou seja, não ligadas), desde que os circuitos integrados sejam accionados por sinais de baixa frequência.

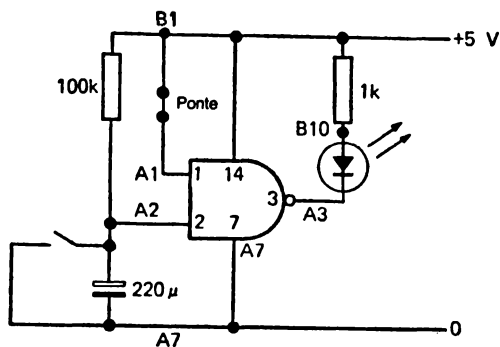


Figura 1.27 — Um circuito típico utilizando o sistema de indicação por números e letras do Eurobreadboard para maior facilidade de montagem. As referências em números e letras permitem verificar os pontos onde devem ser ligados os terminais dos componentes ou pernes dos circuitos integrados. Pode-se usar para montagem qualquer dos cinco pontos existentes em cada linha correspondente a um contacto numerado.

PORTAS LÓGICAS

Uma *porta lógica* é um circuito que utiliza sinais digitais nas suas entradas e saídas. O que transforma um circuito numa «porta» é o facto de cada saída depender *inteiramente* dos sinais aplicados às entradas. Se estes sinais de entrada se alteram, o sinal de saída pode também modificar-se. Os circuitos digitais que utilizam portas lógicas são normalmente montados de tal modo que surge um nível lógico 1 numa saída apenas quando ocorre uma determinada combinação de sinais de entrada — e por esta razão estes circuitos são por vezes designados *circuitos de lógica combinatória*. Em teoria, seria possível fabricar circuitos integrados para cada combinação possível de sinais de entrada de modo a produzir uma saída 1, mas isto obrigaria a desperdiçar recursos. Na prática, prefere-se fabricar circuitos integrados que realizem algumas operações lógicas «standard». A partir destes circuitos integrados de lógica «normalizada» é possível constituir qualquer circuito lógico combinatório. O microprocessador é aliás uma extensão desta ideia — um circuito que pode realizar virtualmente qualquer função lógica.

O funcionamento de um circuito de lógica combinatória ou de qualquer circuito constituído por estas unidades, pode ser descrito de duas maneiras. Uma consiste em utilizar uma *tabela de verdade*. Uma tabela de verdade mostra a saída que é obtida a partir de qualquer combinação possível de entradas, tornando assim possível a verificação do funcionamento do circuito. Um outro método de descrever o funcionamento de um circuito consiste em recorrer à *álgebra de Boole*. Este método é muito mais conciso, mas menos fácil de utilizar

pelo principiante, pelo que ao longo deste livro iremos utilizar ambos os métodos.

A álgebra de Boole, a propósito, foi inventada muito tempo antes dos modernos computadores. O seu nome deve-se a George Boole (1815-1864), que a concebeu como método de transformar afirmações lógicas em expressões algébricas. A sua obra foi pouco utilizada até Shannon verificar em 1938 que a álgebra booleana podia ser usada para analisar circuitos de relés capazes de realizarem operações de comutação que hoje são realizadas pelas chamadas portas «AND» e «OR».

Porta AND

A figura 2.1 mostra o símbolo, a tabela de verdade e a expressão booleana de uma porta AND, uma das portas lógicas «standard». Como a tabela de verdade nos mostra, a saída encontra-se ao nível lógico 0 em todos os casos, excepto

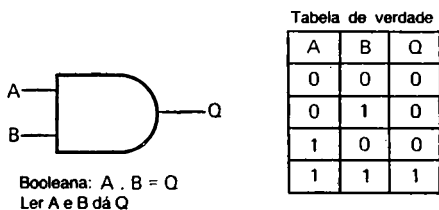
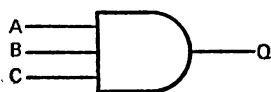


Figura 2.1 — Símbolo de uma porta AND de duas entradas, juntamente com a sua tabela de verdade e expressão booleana correspondente. A saída só é 1 quando todas as entradas têm esse mesmo valor.

quando *ambas* as entradas da porta se encontram ao nível lógico 1. No caso de uma porta AND de três entradas, a saída encontra-se ao nível lógico 1 apenas quando todas as entradas se encontram a esse mesmo nível lógico. Como se mostra na figura 2.2, a tabela de verdade de uma porta com três entradas ocupa muito mais espaço do que a necessária para uma porta de duas entradas, e é excessivamente comprida quando



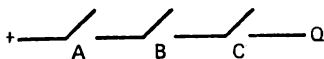
Booleana: $A \cdot B \cdot C = Q$

Tabela de verdade

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Figura 2.2 — Uma porta AND de três entradas. O número de linhas da tabela de verdade é igual a 2^n , sendo n o número de entradas da porta. A saída só é 1 quando todas as entradas se encontram ao nível lógico 1, qualquer que seja o seu número.

se utilizam quatro ou mais entradas. A expressão booleana utiliza o ponto para indicar uma função AND (em português, função «e» ou «produto lógico»), pelo que para uma porta AND de quatro entradas $A \cdot B \cdot C \cdot D = 1$. Por palavras, isto significa que a saída só é 1 quando as entradas A AND B AND C AND D (A e B e C e D) se encontram todas ao nível 1. Note-se que o funcionamento da porta AND pode também ser ilustrado por comutadores montados em série, como se ilustra na figura 2.3



Para cada comutador, aberto=0, fechado=1

Figura 2.3 — Um circuito comutador com a mesma tabela de verdade da porta AND de três entradas.

As portas AND são úteis para verificação da coincidência de uns binários, e no caso de circuitos em que uma saída é controlada por vários sectores. Por exemplo, uma caldeira de aquecimento a gás pode ser ligada (1) quando o

termostato da sala, o termostato da água da caldeira, o detector da temperatura exterior fornecerem todos uma saída lógica 1.

Porta OR

O símbolo, a tabela de verdade e a expressão booleana da porta OR («ou» em português — «soma lógica») são fornecidos na figura 2.4. Este tipo de porta produz uma saída ao nível lógico 1 quando qualquer das suas entradas, ou todas

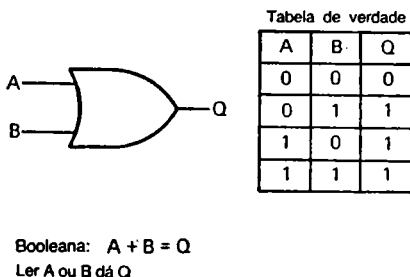


Figura 2.4 — Símbolo, tabela de verdade e expressão booleana da porta OR de duas entradas. A saída é 1 se qualquer das entradas for 1.

elas, se encontram a esse nível lógico. Se todas as entradas estiverem a zero, e só neste caso, a saída será também 0. Tal como no caso anterior, a tabela de verdade de uma porta OR com mais de três entradas torna-se muito volumosa, pelo que se prefere então indicar a acção OR recorrendo à sua expressão booleana: $A + B + C + D + \dots = 1$

A escolha do sinal + pode parecer infeliz à primeira vista porque é mais natural associar este sinal à operação AND, mas as leis da álgebra booleana tornam automática a escolha deste sinal para esta função. A palavra «mais» não deve ser usada quando se lê esta expressão; onde está + deve-se ler sempre OU. A expressão booleana $A + B + C = 1$

é portanto lida A ou B ou C dá uma saída lógica ao nível 1. Note-se que o funcionamento da porta OR também pode ser ilustrada por um circuito de comutadores montados em paralelo, como se indica na figura 2.5. Uma porta OR é usada

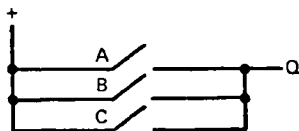


Figura 2.5—Circuito interruptor que tem a mesma tabela de verdade que um porta-OR de três entradas.

quando pode ser usada mais de uma entrada para fornecer uma dada saída. Num sistema de aquecimento central, por exemplo, pode-se pretender uma saída 1 (ligação da caldeira) quando a bomba de circulação arranca «ou» quando é consumida água quente.

Função lógica NOT

A porta NOT («não» em português, ou função «inversora»), ou circuito de complementação, ilustrada na figura 2.6, consiste simplesmente num comutador inversor cuja saída é o

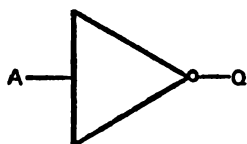


Tabela de verdade

A	Q
0	1
1	0

Booleana: $A = \bar{Q}$

Ler: o inverso (ou complemento) de A é Q,
ou então $Q = \text{NOT } A$:

Figura 2.6—A porta NOT, inversora ou de complementação. Esta porta inverte muito simplesmente o sinal que é apresentado na sua entrada. O pequeno círculo na saída indica inversão—sem este círculo o símbolo é o de um amplificador não inversor, também chamado «buffer».

valor lógico oposto ao da entrada. Em símbolos booleanos, a acção em causa é representada por uma barra colocada sobre a letra que representa a entrada. Por exemplo, um sinal A alimentado a um inversor transforma-se na saída deste em \bar{A} , o que significa que quando $A=0$ $\bar{A}=1$, e quando $A=1$ $\bar{A}=0$.

Podem ser necessárias uma porta inversora num circuito, quando as portas anteriores produziram uma saída com uma polaridade incorrecta, ou se uma dada entrada possui esta polaridade incorrecta. Por exemplo, se quisermos uma saída 1 numa porta AND quando as entradas desta são $A=1$ e $B=0$, uma solução consistirá em utilizar o inversor para converter $B=0$ em $\bar{B}=1$, juntando em seguida a porta AND (figura 2.7).

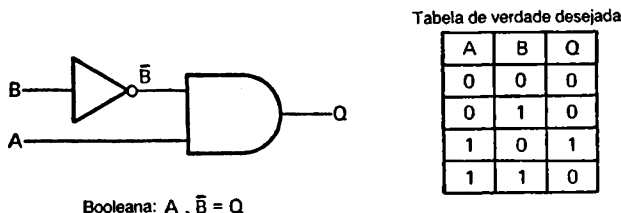


Figura 2.7 — Uso de um inversor para obter uma tabela de verdade que não é fornecida directamente por qualquer das portas AND ou OR.

Portas NAND e NOR

A combinação da porta NOT com qualquer das portas AND ou OR produz circuitos designados por «NAND» e «NOR», cujos símbolos, tabelas de verdade e expressões booleanas são ilustrados na figura 2.8. Estas portas podem parecer não terem grande interesse, mas são mais simples de construir em forma integrada e apresentam a peculiar vantagem de poderem ser convertidas com maior facilidade nas outras portas. Por exemplo, duas portas NAND ligadas da maneira indicada na figura 2.9 formam uma porta AND; e não é possível combinar duas portas AND para produzir uma porta NAND. Nestas condições, as portas NAND/NOR são muito

mais vulgares e fáceis de produzir do que os outros tipos. No final deste capítulo serão fornecidos mais exemplos da utilidade destas portas na formação de outros tipos de portas.

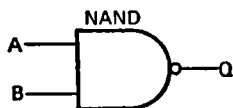


Tabela de verdade

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

$\overline{A \cdot B} = Q$
 Ler NOT (A e B) dá Q

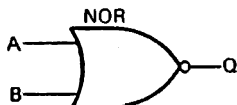


Tabela de verdade

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

$\overline{A + B} = Q$
 Ler NOT (A ou B) dá Q

Figura 2.8 — Portas NAND e NOR, com tabelas de verdade, símbolos e expressões booleanas respectivos.

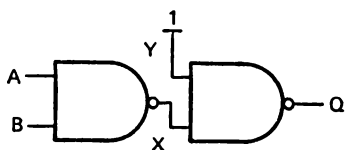


Tabela de verdade

A	B	X	Y	Q
0	0	1	1	0
0	1	1	1	0
1	0	1	1	0
1	1	0	1	1

Figura 2.9 — Duas portas NAND utilizadas para simular a porta AND. A tabela de verdade mostra a maneira como isto é feito — a segunda porta NAND actua como um inversor porque uma das entradas é mantida ao nível lógico 1.

Porta OR — exclusiva

A porta OR simples possui uma saída lógica 1 para qualquer entrada ou combinação de entradas ao nível lógico 1. A porta OR exclusiva (X-OR) tem a sua saída ao nível 1 para

qualquer entrada que se encontre a esse mesmo valor, mas não para uma combinação de entradas 1. A tabela de verdade e o símbolo desta função são apresentados na figura 2.10, juntamente com a sua expressão booleana que, neste caso, é bas-

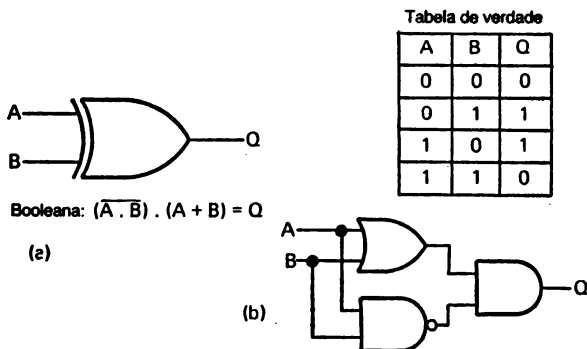


Figura 2.10 — Uma porta X-OR de duas entradas (a). Esta possui uma tabela de verdade semelhante à de uma porta OR, excepto no caso de ambas as entradas estarem ao nível 1. A expressão booleana indica a exclusão utilizando o termo NOT (A e B). (b), uma montagem de portas dando um funcionamento X-OR.

tante mais complicada. No entanto, apesar de existirem circuitos integrados com portas X-OR, a própria expressão booleana contém um método de formação de portas X-OR a partir de outros circuitos (ver mais adiante, neste capítulo).

Como funcionam as portas

A utilidade das funções das portas consiste no facto de ser possível obter qualquer lógica recorrendo a uma combinação adequada destes circuitos. A concepção de um circuito lógico não é muito simples, mas aplicando as regras a seguir

indicadas torna-se relativamente fácil compreender a maneira como o circuito funciona.:

- (1) — Indicar com uma letra cada entrada de todas as portas existentes no circuito, indicando também do mesmo modo a saída (Q).
- (2) — Desenhar uma coluna para cada entrada e saída, deixando as entradas primárias (as das primeiras portas do circuito) à parte das outras.
- (3) — Escrever os valores lógicos 0 e 1 das entradas primárias. Cada fileira deve consistir num número binário, começando por 000... (tantos zeros quantas as entradas primárias) e aumentando um 1 em cada fileira. O número total de fileiras deve portanto ser 2^N , sendo N o número de entradas primárias. Por exemplo, se $N=3$ (três entradas primárias), $n^N=8$, pelo que a tabela de verdade deve possuir oito fileiras.
- (4) — Escreve-se em seguida a saída de cada porta primária, utilizando as tabelas de verdade das portas AND, OR, NAND, NOR ou X-OR, conforme for necessário. Se estas saídas constituírem depois as entradas a outras portas, completam-se as tabelas de verdade destas, até ficar terminada a coluna de saídas. O resultado é a tabela de verdade completa do sistema de portas considerado.

A figura 2.11 mostra um exemplo do uso desta técnica num sistema com três entradas primárias e uma saída. Quando existe porém um grande número de entradas primárias, a tabela de verdade não é prática, sendo então melhor, como habitual, recorrer aos símbolos booleanos. Cada entrada primária é designada A, B, C, etc., como anteriormente, mas as saídas das portas primárias são então designadas pelas expressões booleanas que descrevem o funcionamento da porta. Por exemplo, se uma porta NAND possui duas entradas A, B, a saída é então designada do mesmo modo, utilizando as entradas que já foram determinadas.

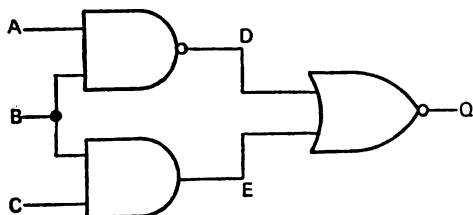


Tabela de verdade

A	B	C	D	E	Q
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	0	1	0

Booleana: $(\overline{A \cdot B}) + (B \cdot C) = Q$
 o que se reduz a $A \cdot B \cdot C = Q$, pelo que
 o sistema de portas pode ser substituído por:

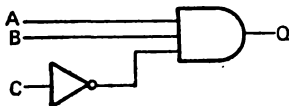


Figura 2.11 — Definição da tabela de verdade de um sistema de portas. Faz-se uma lista das entradas, anotam-se os sinais C e D intermédios partindo das tabelas de verdade das portas AND e NAND. A saída final Q é escrita conhecendo os estados C e D e a tabela de verdade da porta NOR.

A expressão final representa o funcionamento da porta, podendo no entanto necessitar de uma simplificação para remover os termos que não são necessários. As leis da álgebra booleana apresentadas na tabela 2.1 podem ser usadas para realizar esta simplificação, sendo dado um exemplo disto na figura 2.12. Em geral, no entanto, estes métodos são apenas

necessários ao técnico profissional. As leis são aqui indicadas, apenas para referência do leitor, de tal modo que os mais interessados em matemática delas se possam utilizar.

O processo oposto pode ser normalmente realizado de maneira mais simples recorrendo à álgebra booleana. Depois de se ter descoberto a expressão booleana que indica o funcionamento de um dado circuito, este pode ser desenhado mas descobrir esta expressão pode ser bastante problemático. A fim de ilustrar a maneira de construir o circuito usando expressões booleanas, consideremos a expressão da porta X-OR da figura 2.10:

$$\overline{(A \cdot B)} \cdot (A+B) = Q$$

onde $(A+B)$ indica a necessidade de uma porta OR, com entradas A e B. $(A \cdot B)$ indica a necessidade de uma porta NAND, também com as entradas A e B. Finalmente, as saídas destas portas devem ser combinadas numa porta AND, como se indica através do ponto AND existente entre aqueles dois termos. O circuito a usar é indicado na figura 2.10 (b).

Tabela 2.1.

Leis das portas:

AND	OR	NOT
0 . 0 = 0	0 + 0 = 0	$\overline{1} = 0$
0 . 1 = 0	0 + 1 = 1	$\overline{0} = 1$
1 . 0 = 0	1 + 0 = 1	
1 . 1 = 1	1 + 1 = 1	

Leis da álgebra booleana:

A, B são quantidades lógicas que podem ser 1 ou 0.

$$\begin{aligned} A + 0 &= A \\ A + 1 &= 1 \\ A \cdot 0 &= 0 \\ A \cdot 1 &= A \end{aligned}$$

Identidade:

$$A + A = A$$

$$A \cdot A = A$$

Propriedade Comutativa:

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Propriedade Associativa:

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

Propriedade Distributiva:

$$(A \cdot B) + (A \cdot C) = A \cdot (B + C)$$

$$(A + B) \cdot (A + C) = A + (B \cdot C)$$

Complementos:

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

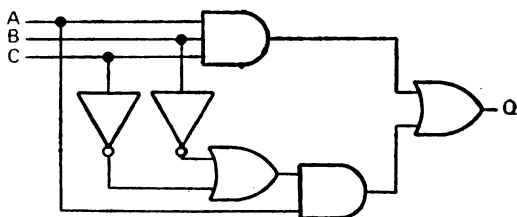
Dupla negação:

$$\overline{\overline{A}} = A$$

Leis de Morgan:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$



Booleana: $(A \cdot B \cdot C) + A \cdot (\bar{B} + \bar{C}) = Q$

Usando as leis de Morgan:

$(A \cdot B \cdot C) + A \cdot (\bar{B} \cdot \bar{C}) = Q$

Pondo A em evidência *

$A \cdot (B \cdot C + \bar{B} \cdot \bar{C}) = Q$

e como $B \cdot C + \bar{B} \cdot \bar{C} = 1$:

$A \cdot 1 = Q$

pelo que o circuito pode ser:

$A \longrightarrow Q$

ou seja, não são necessárias portas

Figura 2.12 — Maneira como se pode usar a álgebra booleana para simplificar uma expressão lógica e eliminar portas desnecessárias.

Este exemplo relativamente simples poderia também ser discutido usando uma tabela de verdade. A tabela mostra que deve ser obtida uma saída 1 quando as entradas sofrem uma lógica OR mas não quando têm ambas o valor 1. Uma porta NAND produz um zero na saída quando as entradas são todas iguais a um, pelo que uma ligação AND de uma porta NAND e outra OR deverá dar o resultado desejado. Depois de se ter concebido deste modo a montagem a efectuar, deve-se verificá-la cuidadosamente construindo uma tabela de verdade da maneira já indicada, a menos que o número de entradas seja extremamente grande.

Pode no entanto ser necessário construir um sistema de portas partindo apenas de informações verbais. Por exemplo, a especificação de que se parte pode ser a seguinte: é necessário um 1 na saída quando qualquer das entradas A e B for 1, mas sendo as entradas C e D iguais a 0. Nenhuma outra combinação deve dar uma saída 1. Pode-se apresentar

estas condições sob a forma de uma tabela de verdade, mas para quatro entradas serão necessárias 16 linhas (2^4). A alternativa consiste em construir uma expressão booleana directamente a partir da especificação em causa. É óbvio que o 1 na saída a partir de entradas A ou B pode ser obtido utilizando uma porta OR, mas não esqueçamos que a saída desta porta só deve produzir uma saída final se nem C nem D produzirem um 1. Estas duas entradas podem então ser alimentadas a uma porta NOR, cuja saída só será 1 quando C e D são ambas zero. Combinando a saída desta porta NOR com a saída da porta OR, utilizando desta vez uma porta AND, é possível obter a função correcta. Em termos booleanos pode-se escrever a expressão do seguinte modo:

$$Q = (A + B) \cdot \bar{C} \cdot \bar{D}$$

Recorrendo às leis de Morgan (tabela 2.1):

$\bar{C} \cdot \bar{D}$ é convertido em $\overline{(C + D)}$, pelo que a expressão se transforma na seguinte:

$$Q = (A + B) \cdot \overline{(C + D)},$$

que corresponde à combinação AND de portas OR e NOR já indicada.

Um dos aspectos fascinantes da concepção de circuitos lógicos consiste em esta não poder ser completamente reduzida a regras; a intuição e a experiência desempenham ainda um papel importante nela. Muitas vezes um técnico experiente será capaz de evitar dificuldades que não são de modo algum óbvias para outro técnico; que se perde no meio de enormes complicações desnecessárias.

Lógica da alta velocidade

Os sistemas lógicos que utilizam sinais que só variam a intervalos bastante prolongados raramente são afectados por problemas devidos a tempos de atraso em cada porta. Alguns

outros sistemas, no entanto, são obrigados a tratarem sinais que se encontram ao nível 0 ou 1 durante apenas curtos períodos, da ordem de um micro-segundo ou menos. Esta lógica a alta velocidade apresenta dificuldades devido aos tempos de atraso dos diferentes circuitos.

Imaginemos por exemplo uma porta NAND que produz um impulso negativo quando ambas as suas entradas se encontram ao nível lógico 1. A duração deste impulso negativo corresponderá ao tempo durante o qual ambas as entradas são positivas. Imaginemos agora que os impulsos presentes em ambas as entradas possuem apenas a duração de 1 γ s. Se coincidirem exactamente, o impulso de saída terá também essa duração, mas se um dos impulsos de entrada tiver um atraso de 300 ns ($1000 \text{ ns} = 1 \mu\text{s}$), a duração do impulso de saída só será logicamente de 700 ns. Se um dos impulsos de entrada estiver atrasado de 1 μs ou mais, não se verificará até qualquer alteração na saída da porta. A concepção do circuito de modo a eliminar este tipo de problemas não é uma tarefa para um principiante, e a principal razão de mencionarmos aqui este problema consiste em mostrar porquê certos circuitos funcionam perfeitamente a baixa velocidade mas falham de maneiras curiosas quando se utilizam impulsos de alta velocidade.

Circuitos aritméticos

Como o nome sugere, os circuitos aritméticos são concebidos para realizarem as funções de soma, subtracção, multiplicação e divisão binárias. Graças ao uso da complementação para 2, a subtracção pode ser realizada utilizando uma porta NOT e um circuito somador, e as funções de multiplicação e divisão podem também ser executadas recorrendo apenas ao circuito somador combinado com registos de deslocamento (capítulo 7).

A adição de dois números binários no papel é fácil, mas a montagem de portas lógicas que realizem esta mesma acção já o é menos. Os bits menos significativos são os mais fáceis

de somar porque não há necessidade de contar com o transporte de uma unidade da ordem anterior. A soma de 1 a 0 produzirá uma saída 1, e a soma de um 1 a outro 1 produzirá um 0 na saída principal; neste caso, porém, terá sido produzido também um valor de transporte, 1, que deverá ser somado ao par seguinte de bits. A tabela de verdade desta secção, designada *semi-somadora*, é apresentada na figura 2.13. A saída na coluna das somas é a mesma que se deveria

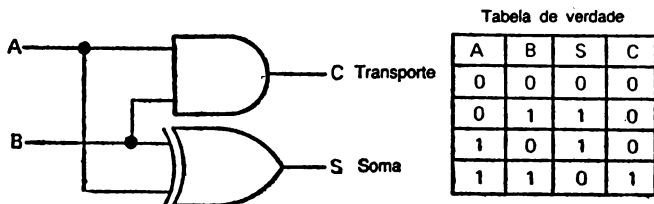


Figura 2.13 — Um circuito semi-somador e a sua tabela de verdade.

esperar de uma porta X-OR, enquanto a saída de transporte é o resultado de uma operação AND executada sobre as entradas. É portanto possível construir um semi-somador simples utilizando o circuito de porta apresentado na figura 2.13. Os semi-somadores podem também, como é óbvio, ser obtidos sob a forma de circuitos integrados MSI, ou como uma reduzida parte de circuitos LSI.

Para somar todos os outros pares de bits já é necessário um circuito mais elaborado, o *somador* propriamente dito. A razão disto consiste em que agora são necessárias três entradas, duas para os bits que vão ser somados e outra para o bit de transporte, proveniente da soma anterior. A tabela de verdade é agora a da figura 2.14; se o leitor não compreender bem o que acabamos de dizer pode realizar uma adição como exemplo.

O somador pode ser obtido sob a forma de um circuito integrado MSI, mas pode também ser constituído recorrendo

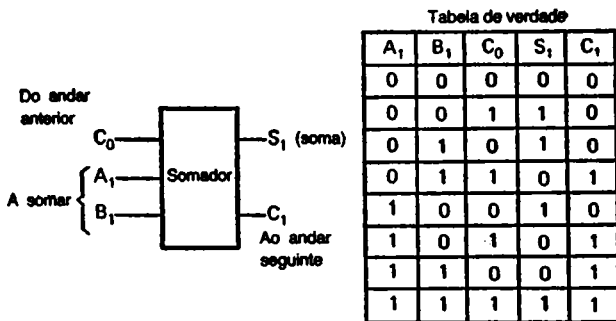


Figura 2.14. — Tabela de verdade e símbolo de um circuito semi-somador.

a portas «standard», da maneira indicada na figura 2.15. O leitor deverá verificar, utilizando o procedimento já indicado, se este conjunto de portas produz ou não a tabela de verdade correcta.

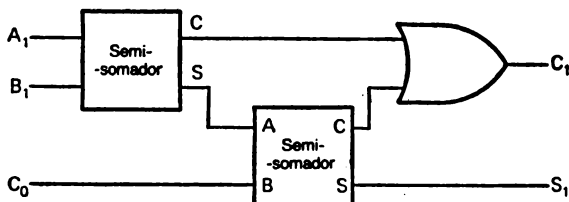


Figura 2.15. — Ligação usada para um circuito somador.

Sinais negativos

Ao discutirmos estes problemas de aritmética temos considerado sempre que os números são positivos. Em muitas aplicações digitais, no entanto, utilizaram-se números negativos, tornando-se portanto necessário algum método de indicação do sinal do número. Um método óbvio de escrever um número negativo consiste em utilizar o complemento para 2

de um número positivo de tal modo que, por exemplo, o número 100101 (37 na base dez) seja convertido em 011011 (27 na base dez). Somando este a um número binário com seis bits, obtém-se o mesmo resultado que subtraindo de 37; por exemplo, a soma de 111001 (57 na base dez) produz 010100 (20 na base decimal) quando o sétimo bit é eliminado, como se vê na figura 2.16.

$$\begin{array}{r}
 100101 \longrightarrow 011011 \quad \text{Complemento para 2} \\
 \text{Decimal 37} \\
 \quad \quad \quad + 111001 \\
 \quad \quad \quad \hline
 \quad \quad \quad (1)010100 \\
 \text{Resposta: } \underline{010100} \quad \text{Depois de eliminar o transporte}
 \end{array}$$

Figura 2.16. — Aritmética do complemento para 2.

O que acabamos de dizer é suficientemente «lógico», mas o problema agora consiste em que um circuito digital não sabe distinguir entre 011011 designando o número decimal 27 e 011011 designando o número decimal — 37; na saída de um computador, não seria possível distinguir o número positivo do negativo. Isto pode no entanto ser resolvido utilizando números em forma de bytes de oito bits, e reservando o bit mais significativo para a indicação do sinal. Os outros sete bits são então usados para representar o número, pelo que quando se utiliza este método de indicação do sinal só se dispõe de sete bits em cada byte. Se desejamos utilizar números maiores, podemos representá-los por bytes duplos, com 16 bits, sendo o mais significativo (ó 16.º) usado para representar o sinal. Restam assim 15 bits para representar o número. Do mesmo modo, é possível utilizar números de três ou quatro bytes, continuando o bit mais significativo a ser usado para indicação do sinal.

A convenção sobre qual o valor do bit mais significativo que deve representar o sinal do número é fácil de compreender. Quando, no nosso exemplo anterior, o número decimal 37 foi escrito sob a forma de um byte de oito bits, transformou-se em 00100101. Convertendo isto em — 37 e tirando

o complemento para 2 obtém-se 11011011, de tal modo que o bit mais significativo é 1, indicando um valor negativo. O número 01011011 será um número positivo (decimal +91). Um 0 como bit mais significativo indica portanto um número positivo, e um 1 indica um número negativo quando esta convenção é usada.

A vantagem deste esquema é que as operações aritméticas podem ser realizadas com estes bytes sem ter em conta o facto de os números binários representarem valores positivos ou negativos, sendo apenas quando se obtém um resultado final que o 1 existente como bit mais significativo deve ser interpretado como 256 (se apenas forem utilizados números positivos) ou como um sinal de «menos» quando são usados números negativos.

Se estiverem a ser usados números binários com sinal, um 0 no algarismo mais significativo indica que os dígitos seguintes representam um número positivo, e a conversão binário para decimal é realizada da maneira habitual. Se o bit mais significativo for 1, no entanto, é escrito um sinal negativo, e os bits seguintes são considerados como o número complementar do verdadeiro (complemento para 2). Para determinar o equivalente decimal, portanto, subtrai-se um do bit menos significativo e complementa-se o número. Por exemplo, o número 11011011 é lido nas seguintes fases:

- (a) o bit mais significativo indica um sinal negativo;
- (b) subtrai-se 1 a 1011011, dando 1011010;
- (c) este é complementado, dando 0100101;
- (d) o equivalente decimal de 0100101 é 37, pelo que o número representado por 11011011 é — 37.

Circuitos sequenciais

Um circuito de lógica combinatória produz uma saída que é completamente determinada pelas entradas, de tal modo que uma determinada combinação de entradas produz sempre a mesma saída. Os circuitos de portas lógicas podem também

ser combinados de modo a produzirem circuitos, designados por *circuitos de lógica sequencial*, que se comportam de maneira bastante diferente:

O circuito da figura 2.17 constitui um exemplo destes circuitos de lógica sequencial. O circuito utiliza duas portas NAND, acopladas em cruz como se mostra, com duas entradas

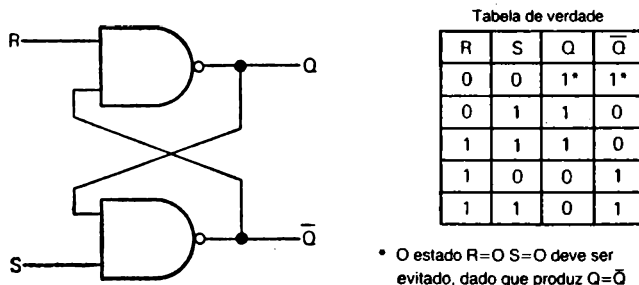


Figura 2.17. — O flip-flop R-S.

e duas saídas. À primeira vista, a tabela de verdade parece bastante normal, mas repetindo parte das entradas verifica-se que a saída depende da sequência de entradas e não apenas da sua combinação.

Por exemplo, com entradas $S = 0$, $R = 1$, a saída é $Q = 0$, $\bar{Q} = 1$; a modificação das entradas para $S = 1$ e $R = 1$ não provoca qualquer alteração. No caso das entradas $S = 1$, $R = 0$, a saída é $Q = 1$, $\bar{Q} = 0$, e mudando $R = 0$ para $R = 1$, de tal modo que $S = 1$ e $R = 1$, não se provoca qualquer alteração. Neste circuito, portanto, as entradas $S = 1$, $R = 1$ podem produzir um 1 ou um 0 numa saída, conforme as entradas que previamente existiam.

O circuito da figura 2.17 é designado *flip-flop R — S*; Estes circuitos são concebidos de modo a manterem o estado da saída mesmo quando as entradas estão a mudar, e são tratados com mais pormenor no capítulo 4. O flip-flop R — S simples tem poucas aplicações.

Utilização prática

A utilização prática de circuitos de lógica combinatória pode ser experimentada recorrendo à placa de montagem já referida, e a alguns circuitos integrados comuns. Na gama 74 TTL, a porta NAND quádrupla 7400 é particularmente útil; na figura 2.18 indica-se a distribuição dos seus pernes.

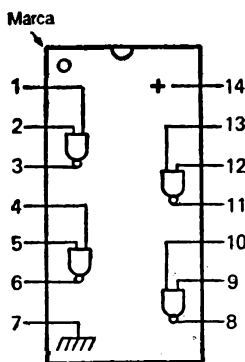


Figura 2.18. — Diagrama dos pernes da porta NAND quádrupla SN7400N.

Os circuitos lógicos podem ser ligados utilizando comutadores como entradas, e recordando que uma entrada em circuito aberto a uma pastilha TTL encontrar-se-á ao nível lógico 1. Os andares de saída podem ser verificados utilizando indicadores LED, com as respectivas resistências de limitação de corrente. Se se utiliza o equivalente CMOS (CD4011A), não se deve esquecer que cada entrada deve possuir uma resistência de 1Mohm permanentemente ligada à massa, e que a saída deve ser ligada a uma combinação transistor/LED, como se indica na figura 1.25.

Os arranjos de circuito da figura 2.19 simulam ações vulgares das portas para as quais é possível determinar a tabela de verdade por experimentação prática ou recorrendo aos métodos de análise atrás mencionados.

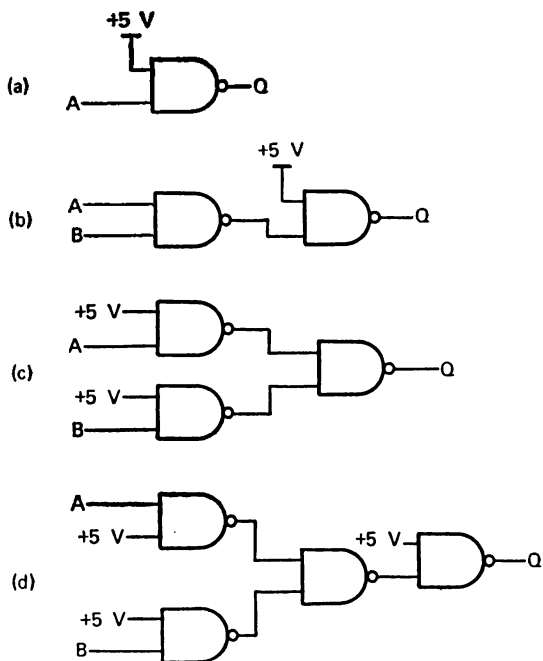
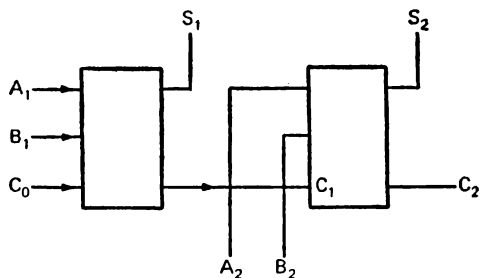


Figura 2.19. — Um exercício prático — ligar as portas de um 7400 de modo a simular as acções de outras portas.

A figura 2.20 mostra o circuito interno do somador de dois bits 7482, com a correspondente tabela de verdade. O bit de transporte entre andares é ligado internamente e não está disponível nos pinos. A distribuição dos pinos do circuito integrado (figura 2.21) mostra quais as ligações que é necessário utilizar aos comutadores a fim de produzir entradas binárias apropriadas, usando A_1 e A_2 como um par de bits e B_1 , B_2 como ordem superior.

Dado que o circuito é um somador, pode-se somar um valor de transporte aos bits menos significativos (C_0), sendo igualmente possível detectar a saída de transporte (C_2) dos bits mais significativos.



Transporte C_0 somado a A_1 somado a B_1 produz S_1 e transporte C_1
 Transporte C_1 somado a A_2 somado a B_2 produz S_2 e transporte C_2

Figura 2.20. — Montagem dos somadores no interior do 7482.

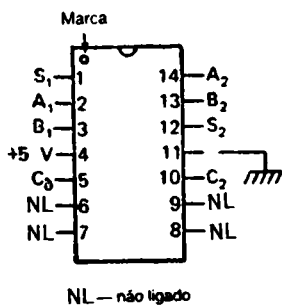


Figura 2.21. — Diagrama da distribuição dos pernes do somador duplo 7482.

ENTRADAS DE RELÓGIO E OUTRAS

Os circuitos combinatórios ou sequenciais simples respondem imediatamente às entradas, perdendo apenas o pequeno tempo de atraso provocado por cada porta. Um circuito de relógio introduz uma ideia completamente nova nos circuitos lógicos — a ideia de que todas as operações se realizam durante o curto tempo de aplicação de um impulso de tensão designado por *impulso de relógio*. O uso destes impulsos é essencial no caso dos circuitos de lógica sequencial porque são eles que determinam o tempo entre as várias fases da sequência.

O impulso de relógio típico apresenta a forma ilustrada na figura 3.1. Entre os impulsos de relógio o nível lógico

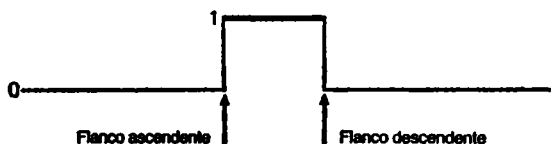


Figura 3.1. — Um impulso de relógio típico.

de tensão é zero, e no flanco ascendente do impulso de relógio a tensão aumenta repentinamente para o valor lógico um. O tempo necessário para este aumento de tensão é designado *tempo de crescimento*, e durará alguns nanosegundos. A tensão mantém-se ao nível lógico 1 durante um tempo designado por *comprimento do impulso*, voltando em seguida ao nível lógico

zero num tempo designado por *tempo de queda*. Os circuitos que utilizam impulsos de relógio funcionarão tanto no flanco ascendente como no flanco descendente do impulso de relógio.

A figura 3.2 mostra um flip-flop R-S. A tabela de verdade deste circuito mostra os valores de saída antes e depois

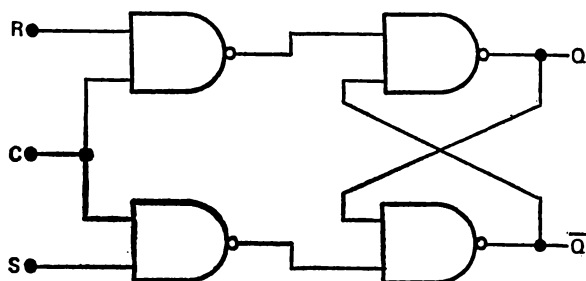


Tabela de verdade

Q_{n-1} = saída antes do impulso de relógio

Q_n = saída depois do impulso de relógio

R	S	Q_{n-1}	Q_n
1	1	não deve ser usado	
0	1	X	0
0	0	0	0
1	0	X	1
0	0	1	1

Figura 3.2. — Um flip-flop R-S pilotado por impulso de relógio. Enquanto a entrada de relógio se encontra a baixa tensão, as entradas às portas flip-flop são ambas elevadas, mantendo as saídas no estado em que se encontravam anteriormente. As alterações nos entradas R e S não têm qualquer efeito até a entrada de relógio adquirir um valor mais elevado. Na tabela de verdade, X significa «variável» — a entrada pode ser 1 ou 0.

de aplicação do impulso de relógio. A diferença importante entre este flip-flop e o seu tipo mais simples consiste no facto de a saída não se alterar até chegar o impulso de relógio, mantendo-se daí em diante novamente fixa até ao impulso seguinte, admitindo que a entrada não se altere entretanto.

Um flip-flop controlado por relógio como este actuará como uma «memória» temporária de um dígito binário, tornando-se particularmente útil quando é necessário que uma saída de variação rápida, como a de uma unidade de cálculo, forneça informações a um dispositivo de funcionamento lento, como uma saída por impressão. Utilizando o dispositivo lento para produzir um impulso de relógio sempre que está disponível para mais informações, pode-se usar o flip-flop para guardar a informação até ser necessária. Nos circuitos de microprocessamento são muitas vezes usados flip-flops de oito bits para este fim.

A pilotagem do relógio é essencial no caso dos circuitos LSI porque os sinais que atingem um dado ponto do circuito podem ter passado por um número muito diferente de portas, trazendo portanto atrasos diferentes. Travando o funcionamento até chegar um impulso de relógio, e verificando se o tempo entre os impulsos de relógio é sempre superior ao maior atraso que o sinal pode sofrer, todos os problemas provocados pelos atrasos do sinal desaparecem automaticamente.

A forma dos impulsos de relógio não é importante. Os circuitos a que é possível aplicar impulsos de relógio incluem amplificadores de elevado ganho que só funcionam como amplificadores durante o tempo em que estão a comutar do nível lógico 0 para o nível lógico 1. Se este tempo não for muito curto (cerca de 35 ns), estes circuitos amplificadores podem oscilar durante a comutação, produzindo assim alguns impulsos de relógio próprios. Estes impulsos falsos, por sua vez, podem ser passados ao longo do circuito produzindo um funcionamento incorrecto.

Muito circuitos utilizando estes impulsos especificam tempos de crescimento e de queda bem definidos para os seus impulsos de relógio, e só é então possível garantir um funcionamento correcto se estes tempos não forem excedidos. O tempo especificado corresponde ao necessário para que a tensão se eleve ou diminua entre 10 % e 90 % do seu valor máximo. No caso de um sistema que utilize um sinal de 5 V, por exemplo, o tempo de crescimento (ou «de estabelecimento») será o tempo necessário para que a tensão passe de 10 % de

5,0 V (ou seja, 0,5 V) para 90 % do mesmo total (ou seja, 4,5 V), e o tempo de queda será evidentemente o necessário para que a tensão passe de 4,5 V para 0,5 V.

Uma outra fonte de impulsos falsos é o comutador mecânico. Quando se utilizam comutadores deste tipo para verificar o funcionamento de portas lógicas, como fizemos no capítulo anterior, não surgem quaisquer problemas porque o resultado final é observado nos LEDs. Porém, quando se usa um comutador para produzir impulsos de relógio, ou quaisquer impulsos que devam ser contados, surge o problema do «ressalto» dos contactos.

O comutador mecânico utiliza molas, pelo que os contactos oscilarão algumas vezes quando fechados por um comutador mecânico, produzindo-se assim um certo número de impulsos, em cada «ressalto»; se o comutador for utilizado para produção de impulsos de relógio cada um destes ressaltos será entendido pelo circuito como um deles.

Nestas condições, sempre que é usado um comutador mecânico este deve ser ligado a um circuito especial que remove os impulsos adicionais produzidos pelo ressalto dos contactos. Na figura 3.3 mostra-se um destes circuitos que utiliza um comutador de duas vias de polo único juntamente com duas portas NAND montadas em circuito flip-flop R-S. O funcionamento é o seguinte:

Com o comutador na posição 1, a entrada R do flip-flop R-S está ligada ao nível lógico 0, e a entrada S ao nível

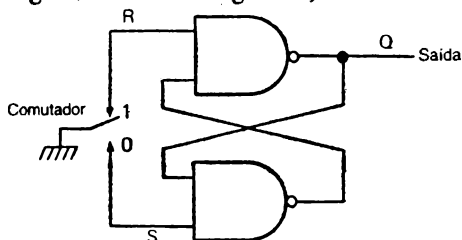


Figura 3.3.— Uso de um circuito flip-flop R-S para «eliminar» o ressalto de um comutador. Nos circuitos de microprocessadores este efeito é conseguido definindo um período de tempo antes do qual não pode ser lido o valor apresentado pelo comutador.

lógico 1. Produz-se assim uma saída 1 em Q. Quando o comutador muda de posição, a primeira parte da sua acção produz o corte de ambos os contactos R e S, pelo que ambas as entradas passarão ao nível lógico 1. Na tabela de verdade do flip-flop R-S verifica-se que isto não afecta a saída, o mesmo acontecendo no caso de o contacto do comutador ressaltar sobre o contacto 1. Quando o contacto toca pela primeira vez na ligação 0, no entanto, liga a entrada S a este nível lógico. Isto comuta imediatamente a saída do flip-flop para 0, porque a entrada R se encontra ao nível lógico 1. Se o contacto do comutador ressaltar, o efeito disto consistirá apenas em levar ambas as entradas R e S ao nível lógico 1, mantendo a saída sem alteração.

Na figura 3.4 mostra-se um outro tipo de circuito eliminador deste «ressalto». O segundo método a que ele corresponde recorre a um *circuito de entrada Schmitt*, que existe no mercado, associado a portas NAND (7413, 74132) ou a inversores (7414). O circuito Schmitt apresenta a característica de poder comutar em ambos os sentidos, mas não à mesma tensão. A entrada Schmitt pode por exemplo comutar num dado sentido a + 4,0 V, mas só voltará à posição anterior quando atingir o valor de + 1,0 V. A velocidade da comutação é determinada inteiramente pelos circuitos existentes no interior do dispositivo, pelo que será produzido um crescimento ou queda da tensão muito rápidos independentemente da possível lentidão da tensão de entrada.

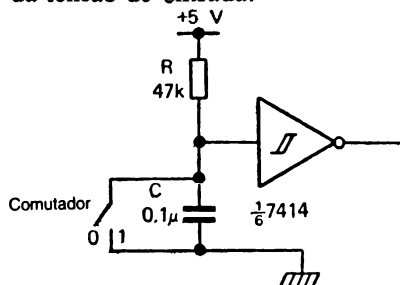


Figura 3.4. — Outro tipo de circuito utilizando uma porta Schmitt, um dos seis inversores Schmitt existentes no circuito integrado 7414. O símbolo apresentado no interior do símbolo inversor indica um funcionamento Schmitt.

No circuito da figura 3.4, o comutador é de tipo unipolar e encontra-se ligado a um condensador e a uma resistência, como se mostra; com o comutador aberto, a ligação de R a + 5,0 V assegura que a entrada do inversor ou porta se encontra ao nível lógico 1, pelo que a saída está ao nível lógico 0. Quando o comutador fecha, o primeiro toque entre os contactos descarrega C, pelo que a tensão de entrada diminui para o nível lógico 0, levando a saída do inversor a comutar novamente para o nível lógico 1. Se os contactos ressaltam, a entrada do inversor não pode aumentar *imediatamente* para uma tensão que provoque a comutação da saída, devido ao tempo necessário para carregar o condensador C. Os valores de R e C são escolhidos de tal modo que o condensador não possa carregar para uma tensão capaz de comutar o inversor no tempo de um ressalto.

Os circuitos de entrada Schmitt, que existem em tipos TTL e CMOS, são extremamente úteis quando se obtêm impulsos de relógio a partir de transístores osciladores ou de outros circuitos que não utilizem circuitos digitais. Qualquer forma de onda, por exemplo, cuja amplitude seja limitada a 5,0 V, comutará um circuito Schmitt produzindo impulsos de relógio cujos tempos de crescimento e queda são óptimos para pilotagem de outros circuitos (TTL e CMOS). Os rápidos tempos de crescimento não são tão importantes para um funcionamento correcto dos circuitos CMOS como o são no caso dos circuitos TTL, mas a existência de impulsos de relógio com uma forma mais perfeita é sempre vantajosa. As entradas podem ser constituídas por corrente alternada rectificadas, ondas sinusoidais, ou qualquer outro tipo de onda desde que a sua amplitude não exceda a amplitude máxima de entrada permitida neste circuito.

Se os sinais usados para converter em impulsos de relógio são obtidos da mesma tensão de alimentação já usada para os circuitos digitais, não surgirá qualquer problema de limitação da amplitude. Os sinais de entrada de diferentes tipos, no entanto, provêm muitas vezes de circuitos muito diferentes, funcionando a tensões diversas e podem deteriorar as entradas dos circuitos digitais se forem aplicadas directamente. Estas entradas não serão necessariamente de relógio —

qualquer entrada de um circuito digital deve ser limitada a fim de não exceder os limites normais 0 e 1 para o tipo de circuito usado.

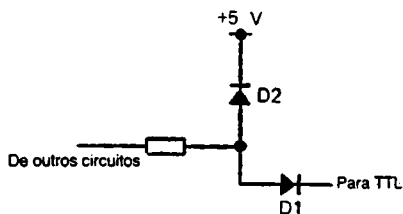


Figura 3.5. — Um circuito de interface que limita a amplitude de um impulso obtido de outros circuitos aos 5,0 V necessários para a lógica TTL.

Na figura 3.5 mostra-se um método de limitação da amplitude usando díodos. O diodo D1 conduzirá apenas quando a tensão de sinal é superior a 0,5 V, de tal modo que a entrada do circuito digital se encontra protegida contra entradas de sinal negativas. O diodo D2 conduzirá no caso de a tensão de sinal exceder 5,5 V, pelo que a tensão aplicada ao diodo D1 não pode exceder 5,5 V. Isto assegura por sua vez que a entrada ao circuito digital não pode exceder 5 V, dada a queda de 0,5 V no diodo condutor.

Um outro método recorre a um transistor para a limitação de tensão. Sendo o colector do transistor alimentado a partir de uma fonte a + 5,0 V, a tensão no colector deve oscilar entre + 5,0 V (bloqueio do transistor) e 0,2 V (condução do transistor). Uma resistência montada em série com a entrada de base impede correntes de pilotagem excessivas, e uma pequena capacidade em paralelo com esta resistência de entrada impede os flancos dos impulsos de serem excessivamente arredondados pela capacitância existente entre a base e o emissor. O circuito pode ser usado em duas formas: na forma NPN é produzido um flanco dianteiro negativo quase vertical; a forma PNP produz um flanco dianteiro positivo também quase vertical.

Produção de impulsos de relógio

Os impulsos de relógio para os circuitos que são comutados a alta velocidade têm de ser normalmente produzidos, a menos que seja possível utilizar impulsos provenientes de outras fontes (como seja a rede de alimentação). Um circuito de produção de impulsos de relógio pode utilizar transistores ou circuitos integrados digitais ou lineares, desde que possam ser usadas as mesmas alimentações e o impulso produzido possua tempos de crescimento e queda apropriados. O circuito integrado 555 é um tipo muito usado juntamente com circuitos digitais como fonte de impulsos de relógio. A figura 3.7 mostra um circuito temporizador 555, capaz de produzir impulsos de relógio cuja frequência pode ser alterada regulando VR1. Os impulsos de relógio produzidos por este circuito são suficientemente definidos para poderem ser usados com circuitos digitais CMOS, mas é necessário um circuito «buffer» TTL no caso de se desejar pilotar circuitos TTL. Usando um «buffer» Schmitt, por exemplo uma parte do Inversor 7414,

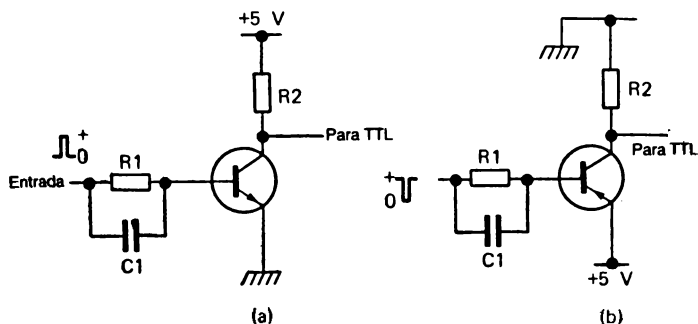


Figura 3.6. — Um interface com transistor (a) NPN, (b), PNP.

assegurar-se-á que os impulsos de saída possuam tempos de crescimento e queda curtos e sejam capazes de pilotarem até 10 entradas de circuitos.

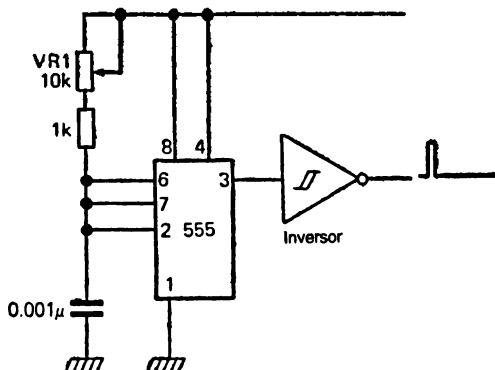


Figura 3.7.—Um gerador de impulsos de relógio utilizando o temporizador 555. A saída do circuito integrado em causa é invertida de tal modo que o curto impulso de retorno do temporizador seja usado como impulso de relógio.

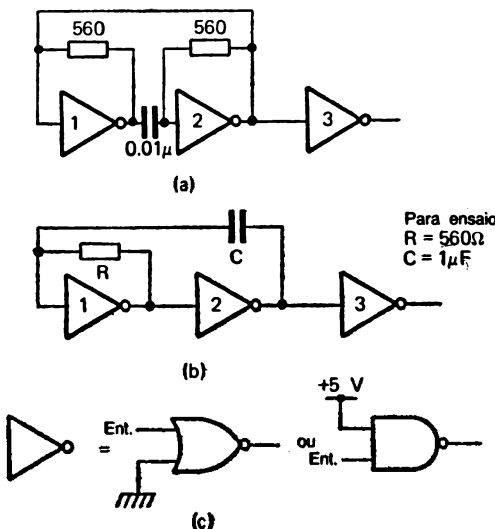


Figura 3.8.—Circuitos osciladores TTL. Os circuitos integrados, dados como inversores, podem também ser portas NOR ou NAND ligadas da maneira indicada. O circuito apresentado em (b) é muito mais vulgar.

Um método alternativo de obter impulsos de relógio consiste em usar um circuito oscilador digital. Podem ser produzidos osciladores utilizando portas NAND ou NOR, usando circuitos como os apresentados na figura 3.8. Note-se que os circuitos TTL necessitam de reduzidos valores de resistência, sendo 1 kohm o valor máximo permissível.

Isto deve-se ao facto de a entrada a um circuito TTL ser feita ao emissor de um transistor cuja base se encontra ligada a $+ 5,0 \text{ V}$. Uma elevada resistência de entrada impedirá este emissor de atingir a tensão lógica 0, muito fraca, impossibilitando portanto uma comutação normal. A corrente utilizada ao nível lógico 0 é $1,6 \text{ mA}$, pelo que a resistência de 1 kohm montada em série com uma entrada TTL produz o crescimento da tensão na entrada para $1,6 \text{ V}$, demasiado elevada para poder funcionar com segurança como nível lógico 0.

No circuito da figura 3.8 (b) a ligação da resistência R entre a saída e a entrada da porta 1 produz uma realimentação negativa, o que reduz o ganho da porta e produz uma gama mais vasta de funcionamento linear. Imaginemos que a tensão na entrada da porta 1 é reduzida. A saída da porta 1 será elevada, e como a porta 2 é também um inversor, a sua saída será baixa. Dadas as ligações de R , o condensador E começará a carregar. Uma placa é mantida a baixa tensão pela saída da porta 2, mas a outra pode ser carregada pela corrente que passa através de R , à medida que a tensão na entrada da porta 1 aumenta, a acção inversora levará a tensão de saída da porta 1 a diminuir, comutando a porta 2 de modo a produzir nela uma saída elevada. O impulso de tensão que passa através do condensador levará a tensão de entrada da porta 1 a aumentar rapidamente para o nível lógico 1, e o condensador descarregará agora através de R porque a saída da porta 1 é agora 0. Os tempos durante os quais as tensões de porta são estáveis não são iguais porque quando a entrada da porta 1 se encontra ao nível lógico 1 a única resistência em circuito é R . Quando a entrada da porta 1 se encontra no entanto ao nível lógico 0, a resistência de entrada da porta (que varia de um circuito integrado para outro) encontra-se em paralelo com R . A terceira

porta em circuito diminui os tempos de crescimento e queda da onda. Note-se que quando se utilizam portas NAND, as entradas não utilizadas devem ser ligadas ao nível lógico 1; quando se usam portas NOR, estas entradas devem pelo contrário ser ligadas ao nível lógico 0.

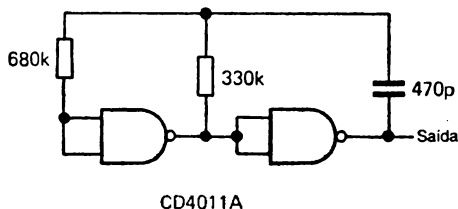


Figura 3.9.— Um oscilador CMOS típico — mostram-se portas NAND, mas é possível utilizar igualmente portas NOR. O inversor CD4069 também pode ser usado.

Os circuitos CMOS podem usar valores de resistência muito mais elevados. Os osciladores TTL têm a reputação de nem sempre arrancarem quando o circuito é ligado, e os osciladores baseados nas portas digitais Schmitt são preferíveis, além de necessitarem de menos secções de um circuito integrado. Na figura 3.10 é ilustrado um oscilador Schmitt típico. Imaginemos que a tensão de entrada é baixa. Como a porta é um amplificador inversor, a saída será elevada, ao nível lógico 1. A corrente que passa através de R carregará C, pelo que a tensão em C aumentará, seguindo a habitual curva de carga do condensador com uma constante de tempo RC. Quando a tensão de entrada atinge a tensão de disparo superior do circuito Schmitt, a porta comuta e a tensão de saída passa para o nível lógico 0. O condensador descarrega agora através de R, mas a diminuição de tensão na entrada não afecta a saída da porta até ser atingida a menor tensão de disparo. Neste ponto, a porta comuta novamente e a acção repete-se. Os tempos de crescimento e queda são muito curtos porque são controlados pelos circuitos internos da porta Schmitt e os impulsos de relógio são de excelente qualidade.

A saída do circuito da figura 3.10 é uma onda quadrada com uma razão marca/espaco de aproximadamente 1/1. Se se desejam impulsos de relógio de duração muito curta, pode-

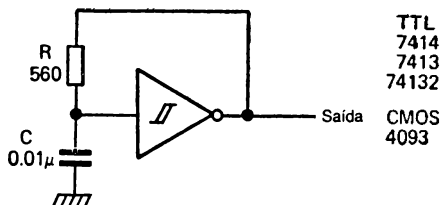
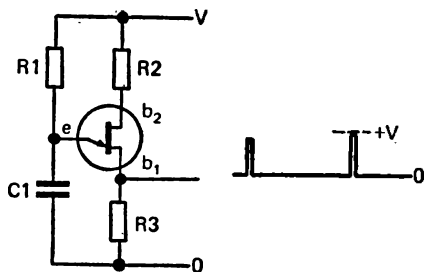


Figura 3.10. — Um circuito oscilador Schmitt. Os circuitos TTL só podem usar reduzidos valores de resistência, inferiores a 1 kohm, neste circuito, mas os circuitos integrados CMOS permitem o uso de valores bastante superiores de resistência.

-se utilizar como fonte o chamado *oscilador unijunção*. Este circuito é mostrado na figura 3.11. O oscilador unijunção é um dispositivo de disparo que possui duas ligações a uma banda semicondutora, e uma a uma junção emissora. Quando é aplicada uma tensão entre os contactos de base só passa uma corrente bastante reduzida até a tensão da junção ser levada a um nível que corresponde a uma fracção constante da tensão entre as ligações da base.

Quando esta tensão «de disparo» é atingida, o oscilador unijunção conduz livremente do emissor para b_1 e de b_2 para b_1 , pelo que podem passar fortes correntes transitórias. Devem ser ligadas resistências de limitação a fim de assegurar que estas correntes não sejam tão fortes que possam deteriorar o componente unijunção.

No circuito da figura 3.11, com o condensador C1 descarregado, a tensão de emissor é baixa, pelo que a unijunção não está em condução. À medida que C1 carrega através de R1, a tensão de emissor aumenta até atingir o nível de disparo. A esta tensão, o unijunção torna-se bom condutor e C1 descarrega para a massa através da junção emissor-base e de R3, passando também corrente através de R2. A descarga é rápida devido ao reduzido valor de R3, sendo produzido um impulso de curta duração. A descarga do condensador tem



Valores típicos
 $R1 = 100k$ (deve ser elevado)
 $C1 = 10n$
 $R2 = R3 = 100$

Figura 3.111.— Um oscilador unijunção. A saída é constituída por impulsos muito curtos com tempos de estabelecimento e queda muito curtos.

ainda o efeito de comutar o unijunção novamente para o seu estado de não condução, pelo que todo este processo se inicia novamente. O impulso produzido por um unijunção é suficientemente bem definido para poder ser utilizado pelos circuitos integrados CMOS, podendo-o ser também, normalmente, para pilotagem de circuitos TTL; no entanto, para poder pilotar um certo número de circuitos TTL é preferível um «buffer», e o inversor Schmitt do tipo 7414 produz simultaneamente a acção «buffer» e de definição da onda que possa ser necessária. A frequência do impulso é controlada pela constante de tempo $C1 \times R1$.

Entradas de teclado

Uma série de impulsos aplicados a um circuito constitui um tipo de sinal de entrada designado por *entrada serial*. O outro tipo de entrada é a *paralela* que consiste em sinais lógicos (impulsos ou tensões estáveis) apresentados num certo número de linhas de entrada. As entradas paralelas são muitas

vezes utilizadas juntamente com teclados e o exemplo mais claro disto é o teclado decimal usado numa calculadora.

Dado que os circuitos digitais trabalham apenas com entradas de números binários, o número decimal correspondente à tecla premida deve ser convertido num número binário. A saída de um teclado que realize esta função é designada BCD («decimal codificada em binário») porque cada algarismo decimal é separadamente transformado em número binário.

Isto é, portanto, diferente de transformar *todo* o número decimal num número binário equivalente. Por exemplo, se estivermos a codificar o número decimal 35, ao carregarmos na tecla 3 esta fornece o binário 0011, e ao carregarmos na tecla 5 esta fornece o binário 0101; em BCD, o decimal 35 equivale, portanto, a 0011 0101, onde cada grupo de quatro bits representa um algarismo do número decimal original. Ora isto é muito diferente do valor obtido na codificação binária normal, na qual 35 é representado por 10011, um número de seis bits. Para efeitos de cálculo se desejamos utilizar a aritmética binária relativamente simples, torna-se necessário converter os sinais BCD em binários — o que pode ser feito usando um circuito integrado conversor, como o 74184.

No teclado propriamente dito, é habitual que cada tecla ligue uma linha comum a uma linha de saída. A codificação para BCD pode ser realizada muito simplesmente utilizando um circuito designado por matriz de díodos, representado na figura 3.12. O princípio de funcionamento desta matriz é detalhado em seguida.

Cada linha binária é ligada através de uma resistência de reduzido valor ao nível lógico 0, de tal modo que quando não se está a carregar em quaisquer teclas todas as linhas se encontram ao nível lógico 0. Quando a tecla corresponde ao decimal 1 é premida, o díodo D1 é ligado entre a linha de alimentação positiva e a linha binária de menor ordem (2^0). Do mesmo, quando se carrega na tecla 2 liga-se a linha positiva do díodo D2, levando a tensão da linha binária 2^1 a aumentar para o nível lógico 1. A tecla 3 liga dois díodos, D3 e D4, a ambas as linhas binárias 2^0 e 2^1 dando, portanto, a saída binária correcta 0011. Do mesmo modo, qualquer outra

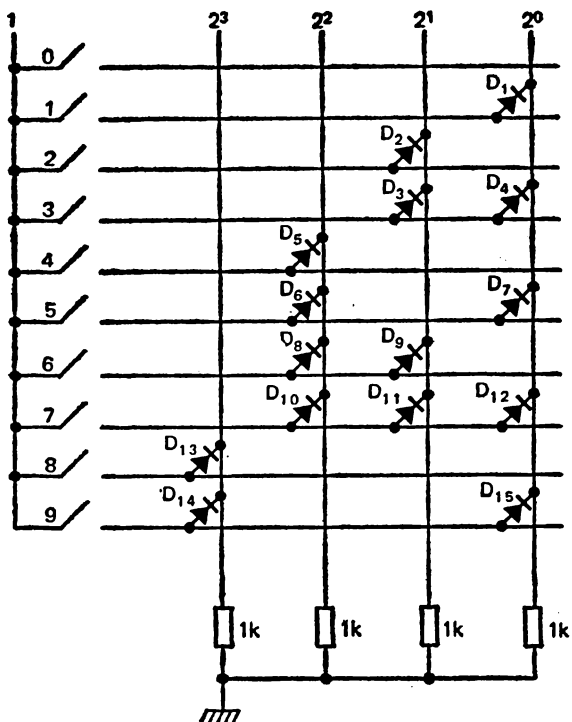


Figura 3.12 — Uma matriz de diodos apropriada às entradas CMOS.

tecla serve para aumentar as tensões das linhas binárias apropriadas através de uma condução por diodos.

Este tipo particular de matriz de diodos é simples e utiliza apenas quinze diodos, mas gasta uma grande quantidade de corrente quando usada com circuitos TTL devido à necessidade de dispor de pequenas resistências em cada linha binária. As entradas CMOS podem utilizar resistências de 100 kohms ou mais, permitindo portanto um funcionamento a baixa corrente. A matriz de diodos da figura 3.13 é melhor para uso

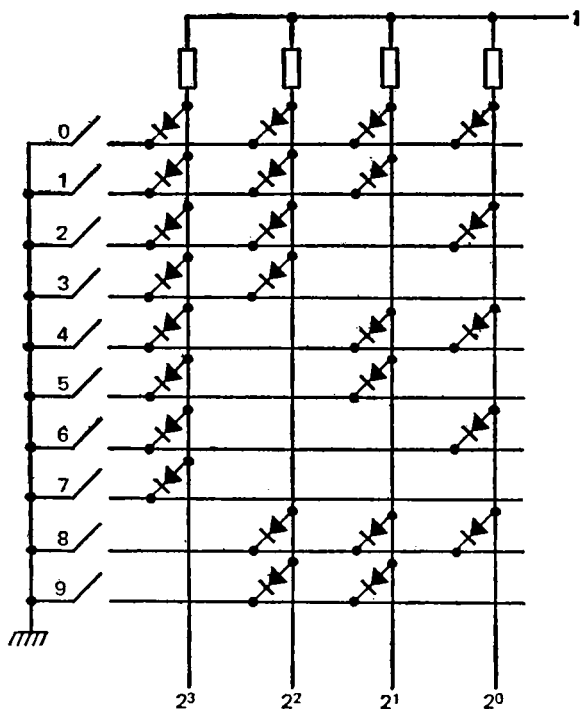


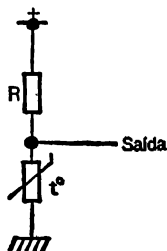
Figura 3.13— Uma matriz de diodos apropriada às entradas TTL. Vários teclados modernos não podem ser usados deste modo, devendo pelo contrário ser ligados a um circuito integrado que realiza uma codificação do teclado.

em circuitos TTL. Neste tipo de matriz, que utiliza 25 díodos, cada comutador decimal produz a ligação de díodos ao nível lógico 0 de tal modo que passe através deles a corrente de entrada TTL. Neste tipo de circuito, no entanto, a tensão de saída é 1111 quando não são fechados quaisquer contactos de comutação, pelo que os circuitos digitais devem ser capazes de identificarem este número (15 decimal) como o sinal de

ausência de entrada, não o passando a quaisquer circuitos de contagem existentes.

Conversão analógico-digital

Imaginemos um detector de temperatura simples usando termistâncias (figura 3.14). A saída deste circuito é uma tensão que se altera em função das variações de temperatura. Este tipo de sinal pode ser detectado muito simplesmente por um instrumento de medida, mas se desejamos obter uma leitura digital somos forçados a converter o sinal de tensão variável (que é um sinal analógico) num conjunto de impulsos que possam ser contados (um sinal digital). Isto requer um circuito chamado conversor analógico-digital (A-D), cuja entrada será o sinal analógico e cuja saída será uma série de impulsos repetidos a intervalos. O conversor A-D necessita, além do sinal de entrada analógico, de uma entrada de relógio de funcionamento contínuo que forneça os impulsos que constituem a saída e de um sinal «reset» que inicia a conversão periodicamente.



(Figura 3.14 — Um simples detector de temperatura utilizando uma termistância para dar um sinal analógico.

Uma forma bastante vulgar deste tipo de conversor é a apresentada na figura 3.15. Um impulso de comando corta um sinal linear em dente de serra, ou sinal «em declive», e dispara igualmente o flip-flop R-S devido à ligação ao terminal S, pelo que a saída Q do flip-flop é elevada. O nível lógico 1

na saída Q do flip-flop é aplicado à porta AND, de tal modo que são alimentados impulsos de relógio ao contador, que foi levado ao nível 0 pelo impulso de comando. Enquanto o nível da tensão em declive é inferior ao da tensão da entrada analógica, a contagem continua; mas quando as duas tensões se tornam iguais o comparador dispara novamente o flip-flop R-S, pelo que a saída Q deste volta novamente ao nível lógico 0. Isto corta a porta AND, terminando portanto a contagem. O número contado é agora proporcional à tensão analógica. O impulso de comando seguinte coloca o contador na posição inicial e inicia o desenvolvimento de outra tensão em declive. Os impulsos de comando podem ser obtidos a partir dos impulsos de relógio ou através de um circuito de atraso accionado pela saída do flip-flop.

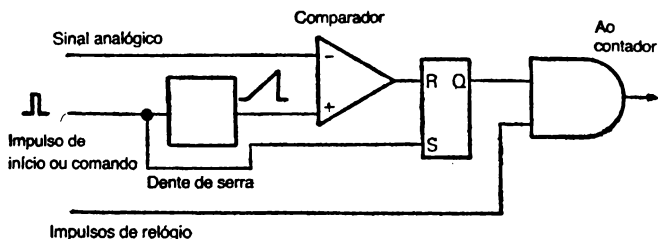


Figura 3.15 — Diagrama de blocos de um conversor analógico-digital. O impulso de início, ou comando, está a uma frequência muito mais reduzida do que os impulsos de relógio, porque a razão frequência do impulso de relógio/frequência do impulso de comando é igual ao número de contagem mais elevado que é possível obter. Note-se que pode ser usado um flip-flop R-S com $R=0$ e $S=0$, dado que só é utilizada a saída Q.

É igualmente necessário o processo inverso ao descrito. Um conversor digital-analógico (D-A) converterá uma série de impulsos em níveis de tensão contando os impulsos e usando o número destes para controlar a tensão de saída. Para darmos um exemplo da utilidade deste circuito, citemos o facto de ser possível sintetizar uma onda de qualquer forma recorrendo a um circuito digital, usando a saída digital para pilotar um conversor D-A. A vantagem deste tipo de produção de sinal relativamente ao procedimento analógico convencional

reside no facto de qualquer onda poder ser produzida com igual simplicidade. Se o circuito digital contém um microprocessador, por exemplo, a onda de saída pode ser controlada pelo programa de instruções do microprocessador, sem necessidade de realizar quaisquer alterações nas ligações internas do circuito.

O princípio de montagem da maior parte dos conversores D-A é a *malha de resistências*, de que é possível observar um exemplo típico na figura 3.16. Cada contador possui uma resistência na saída, estando ligado através dela a uma cadeia ou malha em série, como se vê. Os valores das resistências não são particularmente importantes desde que as razões entre elas sejam correctas, pelo que as resistências são indicadas na figura 3.16 pelas designações R ou 2R — um dos conjuntos de resistências deve possuir exactamente o dobro do valor

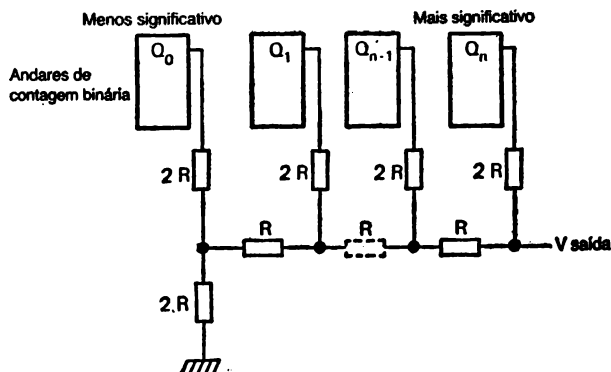


Figura 3.16. — Um conversor D-A com uma malha muito simples.

do outro conjunto. Estes valores das resistências serão geralmente da ordem de 1 kohm, 2 kohms, no caso dos circuitos TTL, e de 10 kohms, 20 kohms nos circuitos CMOS. O bit menos significativo do contador é ligado à extremidade da malha de resistências que se encontra mais longe da saída e o bit mais significativo à extremidade da malha mais próxima da saída. Com esta montagem, a tensão de saída será auto-

maticamente proporcional ao número binário de saída dos andares de contagem.

As razões do uso desta malha consistem em serem apenas necessários dois valores de resistência e na resistência à massa em cada ponto de ligação de um contador ser constante quando todos os contadores estão ao nível lógico 0, a saber, $2R$.

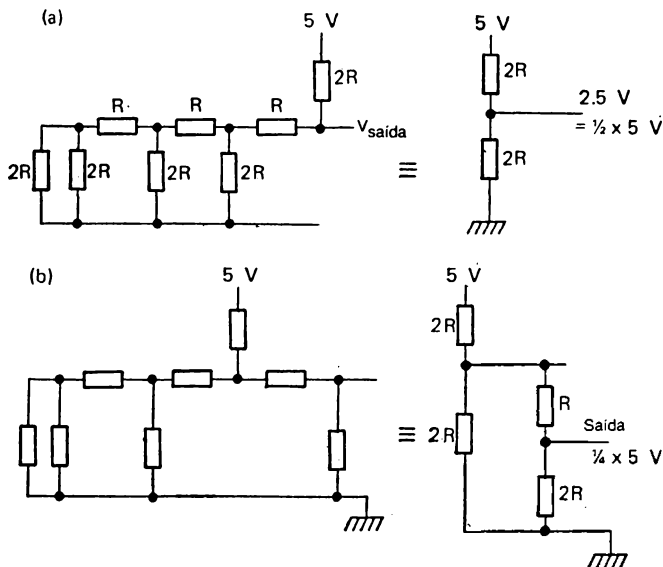


Figura 3.17 — Exemplos de malhas de resistências: (a) dígito mais significativo 1, os outros 0; (b), segundo dígito mais significativo 1, os outros 0.

Se, por exemplo, a saída do dígito mais significativo é 1 e todas as outras são 0, a tensão de saída deve ser exactamente metade da tensão do nível lógico 1, dado que o contador alimenta uma resistência de carga $2R$ através de outra resistência $2R$. Só quando o segundo bit mais significativo é 1 a saída é um quarto da tensão lógica 1, como se pode ver na montagem de resistências apresentada na figura 3.17. Do mesmo

modo, um 1 no dígito seguinte produz $1/8$ da tensão lógica 1, etc. Qualquer número binário nas saídas do contador produzirá portanto uma tensão de saída na malha de resistências proporcional ao número contado.

Funcionamento prático

O efeito do «ressalto» de comutação é difícil de demonstrar por agora, mas pode-se construir um circuito R-S de eliminação de ressalto (figura 3.3). Este pode ser ensaiado utilizando um comutador grosseiro fabricado com três pedaços de fio. O efeito do deslocamento do fio central de um dos fios extremos para o outro, «ressaltando» em seguida, deve ser notado — e se o circuito estiver a funcionar convenientemente este ressalto não deve ter qualquer efeito na saída.

Sem utilizar um osciloscópio não é possível examinar as ondas produzidas por osciladores digitais e outros geradores de impulsos de relógio, mas os sinais de saída podem ser usados para pilotar um amplificador mostrando que o circuito está a oscilar. O leitor deverá construir os circuitos osciladores da figura 3.8, usando os valores indicados. Estes valores produzirão sinais à frequência audio, pelo que a combinação de um amplificador audio e de um alto-falante produzirá um som quando alimentada com estes sinais.

FLIP-FLOPS E CONTADORES

O flip-flop R-S é um tipo simples de circuito sequencial, mas possui pouca utilidade prática. Um outro tipo muito mais útil é o flip-flop D, representado nos diagramas pelo símbolo apresentado na figura 4.1. O circuito interno consiste em portas acopladas entre si, mas só nos interessa tratar aqui o funcionamento global deste circuito integrado, dado que o flip-flop de tipo D é demasiado complexo para valer a pena construí-lo com componentes discretos (separados).

O circuito integrado possui duas entradas principais, D e relógio, e duas saídas, Q e \bar{Q} . Estas saídas são sempre inversas uma da outra nunca podendo ser iguais, a menos que o circuito esteja deteriorado. A entrada D (de Dados) consiste num sinal lógico, 1 ou 0, e a entrada de relógio é o habitual impulso de relógio descrito no capítulo anterior. O funcionamento normal do flip-flop de tipo D é tal que a tensão do sinal lógico presente na entrada D é passada para a saída Q pelo *flanco ascendente* do impulso de relógio. Se, por exemplo, a entrada D se encontra ao nível lógico 1, antes de chegar o impulso de relógio, e a saída Q ao nível lógico 0, esta passará para o nível lógico 1 assim que é recebido o flanco ascendente do impulso de relógio. Os flip-flops de tipo D «prendem» normalmente a informação durante a maior parte do impulso de relógio: significa isto que a comutação está terminada já no momento em que o impulso de relógio atinge o seu nível 1 e qualquer variação da tensão presente na entrada D deixa de ter efeito na saída Q durante o resto do ciclo, ou seja, a maior parte do impulso de relógio, o seu flanco descendente e o tempo entre os impulsos.

O flip-flop de tipo D é muito útil porque torna possível «capturar» sinais em variação muito rápida através da aplicação de impulsos de relógio. Uma saída digital, por exemplo, não se pode alterar rapidamente, senão impossibilita a sua leitura; é portanto necessário algum circuito que mantenha os números estáveis enquanto se está a realizar a leitura.

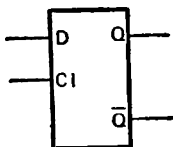


Figura 4.1 — Símbolo de um flip-flop do tipo D. Nesta figura e nas seguintes, «Cl» significa entrada ou impulso de relógio.

Estes circuitos permitem armazenar um bit de informação durante um período de tempo igual ao que medeia entre dois impulsos de relógio, quaisquer que sejam as variações do sinal de entrada verificadas entretanto.

— Pode-se obter uma acção muito importante ligando a saída Q à entrada D, como se mostra na figura 4.2 e usando apenas a entrada de relógio. Recordemos que o sinal de saída \bar{Q} será sempre o inverso do sinal Q; se a saída Q estiver portanto ao nível lógico 0, \bar{Q} estará ao nível 1. Isto conduz a entrada D ao nível 1, pelo que o flanco ascendente do impulso do seguinte comutará o flip-flop, levando a saída Q a mudar para 1. Isto não acontece porém instantaneamente, verificando-se um atraso de cerca de 40 ns; trata-se de um período de tempo bastante curto mas suficientemente longo para assegurar que o sinal de relógio atinge completamente o nível lógico 1

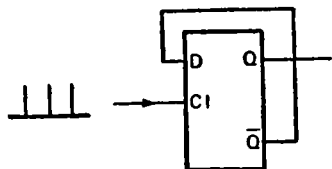


Figura 4.2 — Ligação que transforma um flip-flop de tipo D num contador binário.

antes de se alterar a tensão nas saídas. No momento em que a saída Q muda para 1 e \bar{Q} para 0, o impulso de relógio já atingiu o valor 1, sendo a informação «guardada». O sinal lógico 0 em \bar{Q} não tem, portanto, qualquer efeito na entrada D até chegar o flanco ascendente do impulso de relógio seguinte. Quando tal acontecer, o 0 na entrada D levará a saída Q a comutar para 0, e \bar{Q} para 1, mas de novo com um atraso suficiente para assegurar que a variação de \bar{Q} não tenha qualquer efeito na entrada D.

Na figura 4.3 são mostrados a entrada de relógio e os sinais da saída Q para um flip-flop D ligado deste modo. Dois impulsos completos na entrada de relógio produzem um impulso completo na saída Q, pelo que este circuito é também designado *divisor por dois* ou *contador binário*.

Relógio	Q_{n-1}	Q_n	D_n	\bar{Q}_n
1	0	1	0	0
2	1	0	1	1
3	0	1	0	0

Figura 4.3. — Tabela de verdade do flip-flop de tipo D da figura anterior. A saída Q comuta em cada impulso de relógio.

O flip-flop de tipo D torna-se mais útil no caso de dispor de mais duas entradas. Uma destas é designada «set» ou «preset», e é usada para forçar a saída Q a passar ao nível lógico 1 qualquer que seja o estado do impulso de relógio. A outra entrada é designada «reset» ou «clear», e a sua acção é tal que força a saída Q a passar ao nível lógico 0. Estas entradas são por vezes designadas *entradas assíncronas*, porque qualquer sinal a uma delas não necessita de ser sincronizado com o impulso de relógio. É ainda necessário tomar as precauções necessárias para assegurar que não são accionadas simultaneamente as duas entradas. A maior parte dos flip-flops TTL necessitam de um impulso negativo para accionar qualquer destas entradas.

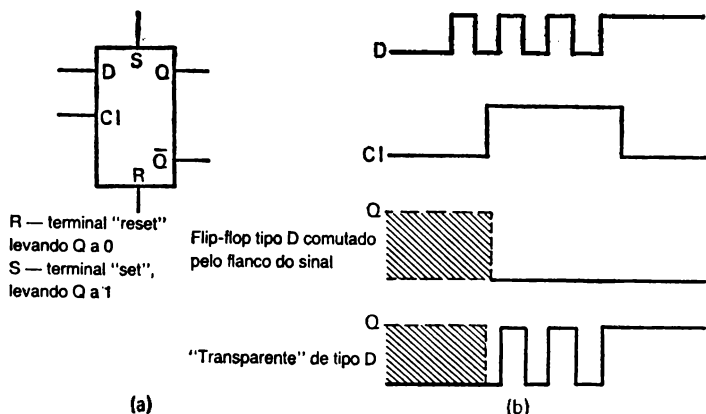


Figura 4.4. — Variantes do tipo D: (a) uso dos terminais reset (clear) e set (preset) para fixar as saídas Q independentemente do impulso do relógio; (b) comparação entre o tipo D disparado pelo flanco ascendente do impulso do relógio e o flip-flop tipo D «transparente» referido no texto. A saída deste corresponderá ao estado da entrada D durante todo o tempo em que o impulso de relógio está ao nível 1. A zona tracejada indica a região em que a saída Q pode ser 1 ou 0 conforme o estado a que a saída foi colocada anteriormente.

O nome «tipo D» é também usado para um outro dispositivo em que os dígitos binários presentes na entrada D só são ligados à saída enquanto o impulso de relógio se encontra ao nível lógico 1 e são armazenados enquanto este se encontra ao nível 0. Este dispositivo não pode ser usado como o anterior, comutado pelo flanco ascendente do impulso de relógio. Seria possível construir ainda um flip-flop comutado pelo flanco descendente do impulso de relógio, mas o tipo estudado é o mais usado.

• O flip-flop J — K

O flip-flop de tipo D, apesar de útil, apresenta várias desvantagens que conduziram à concepção de um circuito

mais versátil, o flip-flop J-K «master-slave». As ligações feitas a um dispositivo deste tipo são indicadas na figura 4.5. Além da habitual entrada de relógio, existem duas entradas designadas «J» e «K» que controlam a acção do flip-flop quando chega um impulso de relógio. Duas outras entradas, designadas como acima «set» (ou «preset») e «reset» (ou «clear») controlam a saída do flip-flop quer tenha sido aplicado ou não um impulso de relógio. As saídas, tal como anteriormente, são Q e \bar{Q} .

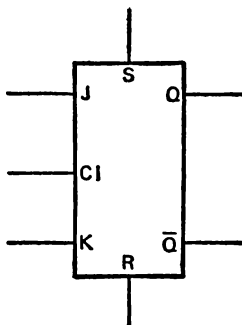


Figura 4.5.— Entradas e saídas da flip-flop J—K. A entrada *set* (S) é por vezes designada P (*preset*), e a entrada R (*reset*) é por vezes designada C (*clear*).

A montagem interna de um flip-flop J—K master-slave é bastante complicada, mas basicamente o circuito consiste em dois flip-flops. O primeiro («master») é accionado pelo flanco ascendente do impulso de relógio e a sua saída é controlada pelos estados das entradas J e K nesse instante. As saídas deste flip-flop «master» são ligadas às entradas do segundo («slave»), que comuta no flanco descendente do impulso de relógio. As saídas do flip-flop «slave» são Q e \bar{Q} , pelo que estas saídas se alteram quando ocorre o flanco descendente do impulso de relógio.

Obtemos assim um sistema que não depende dos atrasos de circuito nem necessita de um impulso de crescimento tão rápido como no caso do flip-flop de tipo D. Os dados exis-

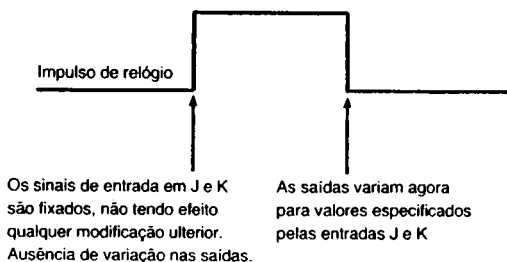


Figura 4.6. — Sequência de acontecimentos num flip-flop J — K master-slave.

tentes nas entradas J e K são transferidos para o «master» no momento em que ocorre o flanco ascendente do impulso de relógio e quaisquer variações das entradas posteriores não têm nenhum efeito. As saídas modificam-se no momento em que ocorre o flanco descendente daquele mesmo impulso, pelo que nenhuma ligação da saída à entrada tem qualquer efeito até chegar o impulso seguinte. A figura 4.6 apresenta um diagrama desta sequência de funcionamento.

A vantagem mais significativa do flip-flop J — K consiste na sua versatilidade devida ao controlo exercido pelas entradas J e K. A figura 4.7 mostra-nos a sua tabela de verdade, referindo-se Q à tensão de saída *antes* de chegar o impulso de relógio, e Q_{n+1} à tensão de saída *depois* do impulso de relógio. Se bem que a tabela de verdade dê uma ideia do que se passa, este funcionamento é tão importante que iremos estudá-lo em maior detalhe. Com $J = 0$, $K = 0$, o sinal lógico na entrada Q não é alterado pelo impulso de relógio, pelo que o flip-flop pode manter a informação durante todo o tempo em $J = 0$ e $K = 0$. Se as entradas forem $J = 1$, $K = 0$, a saída Q comutará para o nível lógico 1 quando for aplicado o impulso de relógio (ou manter-se-á ao nível 1 se era este o seu estado). As entradas $J = 0$, $K = 1$ levarão a saída Q a comutar para o nível 0 quando receber o impulso de relógio ou a manter-se a esse nível se era já esse o seu estado.

J	K	Q	Q _{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	1
		1	0

Figura 4.7. — Tabela de verdade de um flip-flop J — K master-slave.

Quando as entradas J e K são $J = 1$, $K = 1$, a tensão de saída Q inverter-se-á para cada impulso de relógio, produzindo a mesma ação de contagem que o flip-flop de tipo D ligado de maneira inversa na figura 4.2; o flip-flop J — K pode, portanto, ser usado como contador binário sem necessidade de realizar quaisquer ligações de realimentação.

Como se notou anteriormente, as entradas assíncronas designadas «set/preset» e «reset/clear» podem ser usadas para comutar a saída Q para qualquer tensão lógica independentemente da presença de tensões de relógio.

Dispositivos de retenção de informação com três estados

Uma variante do dispositivo descrito é o circuito com três estados, particularmente importante nos circuitos microprocessadores. Um dispositivo de três estados realiza a normal função de retenção do sinal na saída depois das entradas se terem modificado, mas possui ainda um terminal de «isolamento». Quando a tensão a que este terminal se encontra é comutada, as saídas do dispositivo são isoladas, flutuando e podendo assumir qualquer tensão lógica. Este «terceiro estado» é necessário quando os sinais podem passar nas linhas em qualquer dos sentidos e evita os problemas que poderiam surgir no caso de um dispositivo se encontrar ao nível lógico 1 e outro, ligado à mesma linha, ao nível lógico 0. O sinal de

isolamento deve ser produzido em condições tais que o dispositivo «flutue» sempre que as linhas são usadas para outros sinais. Existem no comércio dispositivos de quatro linhas (quádruplos), seis linhas (sêxtuplos) e oito linhas (óctuplos) com esta montagem em três estados, sendo possível escolher ainda entre modelos de dois tipos («high» e «low»). Podemos citar como exemplos o 74125 e o 74126 (quádruplos), o 74365 e o 74367 (sêxtuplos).

Contadores binários

Um contador binário é um circuito cuja entrada é uma série de impulsos e cuja saída é constituída por dígitos binários, com uma linha separada para cada potência de dois, 2^0 , 2^1 , 2^2 , 2^3 , etc. O tipo mais simples de contador binário é uma cadeia de flip-flops ligados entre si da maneira indicada na figura 4.8 (na qual se mostram flip-flops de tipo D). O contador deste tipo é conhecido por vários nomes, principalmente *contador assíncrono*.

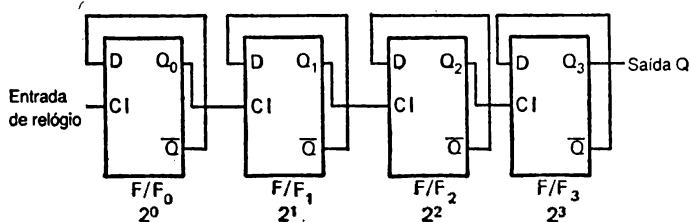


Figura 4.8. — Uma cadeia de flip-flops formando um contador binário. Este tipo de circuito é designado pelo nome de contador assíncrono.

Imaginemos que dispomos de uma cadeia de flip-flops que mudam de estado quando surge o flanco *descendente* do impulso de relógio. Encontra-se ligado um LED a cada saída Q de flip-flop, de tal modo que aquele acenderá quando $Q = 1$. Imaginemos ainda que todos os LEDs estão «apagados» no início da contagem. Quando chega o primeiro impulso, o flip-flop 0 é comutado, pelo que a sua saída Q_0 se

torna elevada (figura 4.9) e o LED correspondente emite luz. Dado que este tipo de flip-flop comuta no flanco descendente do impulso de relógio, o flip-flop 1 não é comutado pelo aumento de Q_0 . Um segundo impulso aplicado ao flip-flop 0, no entanto, obrigará este a comutar novamente, passando a sua tensão de saída de 1 para 0. Este é um flanco descendente pelo que o flip-flop 1 é comutado e a sua saída Q_1 passa para o nível lógico 1, acendendo o segundo LED. Simultaneamente, o primeiro LED deixa de emitir luz devido à diminuição da tensão de Q_0 . Um terceiro impulso levará agora o primeiro LED a acender novamente, não afectando o segundo. Um quarto impulso extinguirá os LEDs 1 e 2, mas ligará o LED 3.

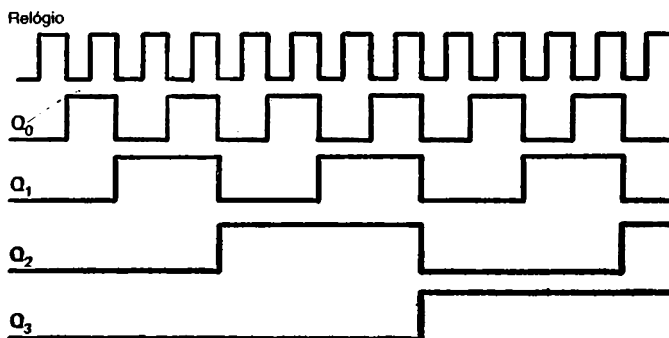


Figura 4.9. — Diagrama da sequência de contagem de um contador binário, considerando que todos os flip-flops estavam «limpos» (*clear*) no início.

A razão de numerarmos os flip-flops 0, 1, 2, etc., em vez de 1, 2, 3, etc., torna-se agora evidente. O flip-flop 0 conta as colunas binárias 2^0 , o flip-flop 1 conta as colunas 2^1 , etc., pelo que os LEDs podem ser lidos como um conjunto de uns binários, cada um deles relacionado com uma determinada potência de 2, como se indica na figura 4.10. Se se acrescenta agora uma porta ao sistema (figura 4.10), o contador fica quase completo. Se todos os flip-flops estão regulados para uma saída 0 (nenhum LED aceso), a porta está aberta

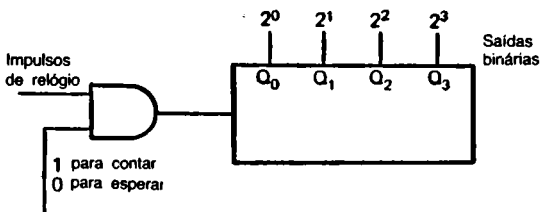


Figura 4.10. — O acrescento de uma porta a um contador binário permite a paragem e o início da contagem recorrendo ao uso de uma tensão de «isolamento».

(isolada). Mais tarde a porta é fechada e o número de impulsos que passaram através dela será registado pelos LEDs do contador. Podemos também considerar esta montagem como um circuito de memória que pode ser regulado para um dado dígito binário (lido a partir do bit mais significativo) que se manterá guardado até todo o contador ser limpo ou serem acrescentados mais impulsos. Uma outra maneira de considerar o circuito é como *somador* em série, que permite somar o número de impulsos na entrada a qualquer número que possa ter sido definido por uma entrada anterior.

O circuito pode ser usado puramente como um contador binário, com qualquer número de flip-flops — sendo vulgar quatro ou oito elementos. Alternativamente, o circuito pode ser modificado a fim de contar de outros modos. Consideremos por exemplo que é necessário um contador BCD. Este contador deve realizar uma contagem binária até nove impulsos, passando novamente a zero quando recebe o décimo impulso — e passando este décimo impulso para o andar BCD seguinte sob a forma de «transporte».

Para uma contagem até nove serão necessários quatro flip-flops, e se forem usados flip-flops J—K master-slave, como o 7476, a passagem a zero pode ser executada recorrendo às entradas de «reset». Como poderemos usar o décimo impulso para accionar esta entrada? A solução mais simples consiste em utilizar um circuito de porta que detectará o decimal 10 e dará uma saída capaz de accionar a entrada «reset». O retorno do flip-flop a zero requer a aplicação de

um impulso negativo naquele terminal, pelo que a porta deve comutar de 1 para 0 quando é contado o decimal 10 para poder produzir a acção desejada.

Na figura 4.11 mostra-se um circuito deste tipo. Como o decimal 10 equivale ao binário 1010, uma porta NAND accionada pelas saídas 2^3 e 2^1 do contador produzirá uma comutação de 1 para 0 que pode ser usada para accionar o retorno a zero. Sempre que se verifica esta acção é necessário restaurar a tensão inicial no perne «reset» a fim de que a contagem possa recommear, pelo que este terminal deve receber apenas um impulso negativo muito breve. Um inversor pode converter este breve impulso negativo num impulso positivo capaz de ser contado pelo andar seguinte de flip-flops BCD.

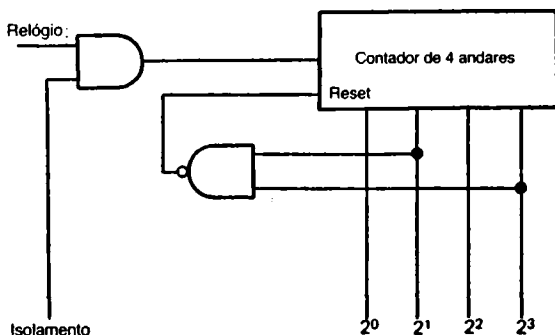


Figura 4.11. — Conversão de um contador de quatro andares num contador binário voltando a zero quando é detectado o decimal 10 (binário 1010) nas saídas.

O uso de portas para controlar as entradas de «reset» de uma cadeia de flip-flops é uma maneira simples e directa de definir o valor máximo da contagem, mas não constitui de facto um método ideal. Quando se pretendem usar muitos andares ou velocidades de contagem muito elevadas, o tempo de atraso do impulso na montagem torna-se excessivo. Consideremos por exemplo um contador com oito flip-flops que já tenha contado até 01111111. O impulso seguinte passará o dígito menos significativo de 1 para 0, o que por sua vez

comutará o seguinte também para zero e assim por diante até o último dígito passar de 0 para 1. Ora existe um atraso notável, de 25 ns ou mais, entre a acção do impulso de relógio e a modificação da saída; estas modificações não são imediatas mas sim sequenciais.

Seria possível montar um circuito de modo a detectar o estado 10000001 ligando as saídas 2^7 e 2^0 à entrada de uma porta NAND que por sua vez accionasse o «reset». Se no entanto a velocidade de contagem for elevada, os primeiros andares do circuito podem já ter contado vários impulsos a mais antes de o flip-flop final comutar, pelo que a contagem aparente não corresponderá à real. O circuito comportar-se-á bem a baixas velocidades devido aos seus atrasos internos. Por estas razões não se usam estes contadores para aplicações em que ocorram altas velocidades de contagem, apesar de apresentarem uma vantagem importante: só o primeiro flip-flop, correspondente ao dígito menos significativo, é obrigado a funcionar à alta velocidade dos impulsos de entrada. O flip-flop (2^1) funciona a metade da velocidade de entrada, etc. Se o contador é usado meramente como um divisor de frequência rigoroso, ou para registar o número de contagens numa saída binária *depois* de os impulsos terem sido cortados, este modelo é perfeitamente apropriado.

O contador não deve necessariamente partir de 0 e continuar até um valor máximo definido anteriormente. Pode construir-se um contador que conte ao contrário, ligando muito simplesmente as saídas Q de cada flip-flop à entrada de relógio do seguinte, como se mostra na figura 4.12.

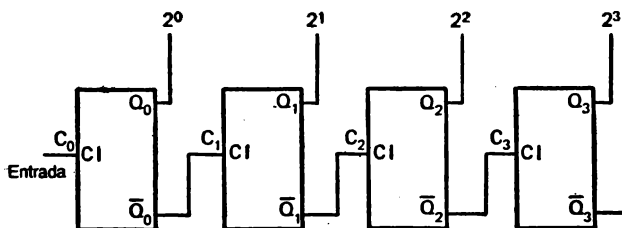


Figura 4.12. — Uma montagem para contagem descendente construída ligando cada saída Q à entrada de relógio seguinte. Só se mostram aqui as ligações Q — relógio.

Imaginemos que as saídas Q se iniciaram no número 1111 (binário), que corresponde ao 15 decimal. O primeiro impulso completo aplicado a C_0 comutará Q_0 de 1 para 0 e Q_0 de 0 para 1. O estado das saídas será agora 1110 (recordemos que C_0 corresponde ao bit menos significativo), que corresponde ao número decimal 14. O impulso completo seguinte comutará novamente Q_0 para o valor elevado, mas como Q_0 passa de 1 a 0, C_1 será pilotado de tal modo que a sua saída passa de 1 para 0. A saída é agora 1101, correspondente ao decimal 13. Esta contagem descendente continuará até as saídas atingirem 0, momento em que o impulso seguinte levará novamente o contador para o estado 1111.

A contagem descendente é muitas vezes um método mais prático de obter uma saída ao fim de um número bem definido de impulsos. A figura 4.13 mostra uma série de flip-flops montados para montagem descendente mas com comu-

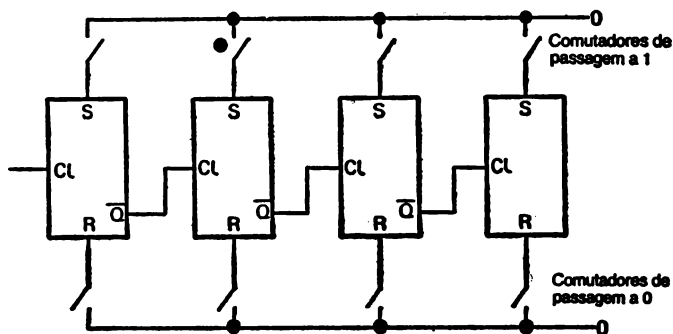


Figura 4.13. — Um contador descendente capaz de ser regulado previamente. A saída Q de cada flip-flop pode ser regulada para 0 ou 1, de modo a que a contagem parta de um determinado número.

tadores ligados aos terminais set/reset. Estes comutadores permitem conduzir cada contador para o estado 1 ou 0, de tal modo que seja possível «carregar» inicialmente um qualquer número binário no contador. O problema resume-se então a arranjar uma maneira de detectar o fim da contagem.

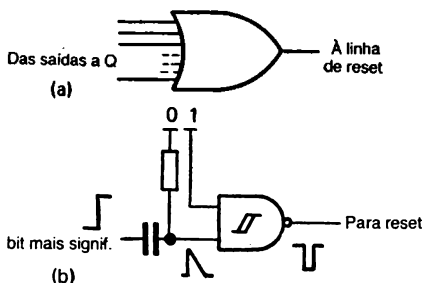


Figura 4.14. — Maneiras de detectar o estado 0000 (a), usando uma porta OR de saídas múltiplas cuja saída só seja zero quando todas as entradas o forem; (b), uso de um circuito diferenciador que detecta o impulso positivo no final de uma contagem.

Um método consiste em usar uma porta para detectar o momento em que todos os flip-flops se encontram ao nível 0. Outro consiste em utilizar um díodo para detectar a variação 0 para 1 que se verifica no bit menos significativo quando o contador passa ao estado 1111 imediatamente depois do estado 0000. Este impulso pode ser definido por uma porta Schmitt e usado para parar a contagem e accionar outros circuitos.

A contagem descendente constitui um sistema muito mais flexível porque o estado que se pretende detectar é sempre o mesmo — o estado 0000 ou a mudança para 1111. O número de contagens é determinado pelo número binário marcado no contador antes de este começar a funcionar. Isto corresponde de facto a uma maneira «primitiva» de programar, em que a contagem é decidida pelo número carregado (*software*) e não pelas ligações existentes no interior do contador (*hardware*).

Uma outra acção possível destes contadores consiste em contarem em qualquer dos sentidos, ascendente ou descendente. Dado que a única diferença entre a contagem ascendente e a descendente reside no uso das saídas Q no primeiro caso e das saídas \bar{Q} no segundo, é possível ligar portas entre

os andares de contagem a fim de seleccionar a saída a utilizar para o impulso seguinte. A figura 4.15 mostra um sistema possível. Cada saída de um flip-flop é ligada à entrada de uma porta AND de duas entradas, cuja outra entrada é ligada a uma linha. As portas AND que possuem entradas dos terminais Q são ligadas à linha ascendente, e as portas AND com entradas dos terminais \bar{Q} são ligadas à linha descendente. As saídas das portas AND são ligadas a uma porta OR que por sua vez pilota o impulso de relógio do flip-flop seguinte.

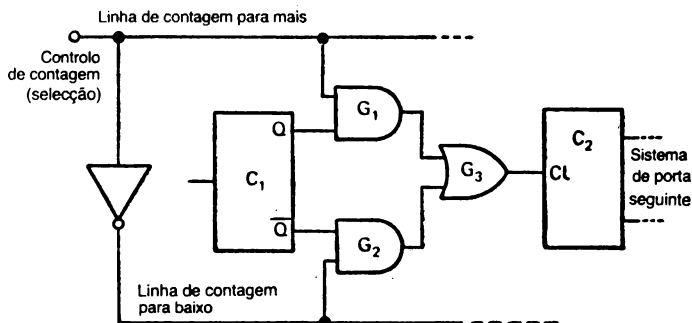


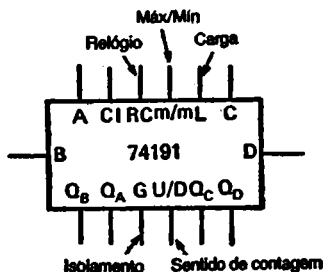
Figura 4.15. — Um sistema de contagem ascendente/descendente, mostrando apenas um par de flip-flops. As duas linhas de controlo são pilotadas por uma única entrada usando um inversor para assegurar que qualquer delas apresenta sempre um valor lógico complementar do da outra.

Quando a linha ascendente é levada (por um comutador de controlo) para o nível lógico 1, mantendo-se a outra linha ao nível 0, um 1 na saída Q produzirá um 1 na saída da porta AND G_1 e, por sua vez, um 1 na saída da sua porta OR, G_3 . Deste modo os impulsos de relógio são ligados entre a saída Q de C_1 e a entrada de relógio de C_2 . Se a linha ascendente for passada ao nível lógico 0, a saída da porta AND G_1 manter-se-á a este nível. Se isto acontecer e simultaneamente a linha descendente estiver ao nível 1, um impulso em \bar{Q} produzirá uma saída 1 em G_2 , e portanto na

saída da porta OR. Nestas condições, os impulsos de relógio presentes em C_2 foram obtidos na saída Q de C_1 , pelo que a contagem efectuada é descendente.

Dado que as linhas ascendente e descendente se devem encontrar sempre em níveis lógicos diferentes, pode ser usado um único controlo, juntamente com um inversor, para assegurar a existência das tensões de linha correctas, como se mostra na figura 4.15. Usando este esquema, um 1 na entrada «selecção» permitirá obter uma contagem descendente, e um 0 na mesma entrada produzirá uma contagem ascendente.

O 74191 é um contador binário integrado de quatro andares ascendente/descendente. O terminal de selecção ascendente/descendente é colocado ao nível lógico 0 para realização de uma contagem ascendente e ao nível 1 para a contagem descendente. Existem possibilidades de carregar cada um dos andares do contador, usando as entradas A, B, C e D. Estas são aceites pelo flip-flop apenas quando o terminal de carga é colocado ao nível lógico 0; em quaisquer outras circunstâncias estas entradas são ignoradas. Um controle adicional, «enable», («isolamento») permite a interrupção da contagem. A contagem realiza-se quando a tensão neste terminal está ao nível 0 e é suspensa quando a tensão neste terminal é 1. O uso deste terminal não conduz o contador à posição inicial; a contagem é apenas interrompida quando este terminal está ao nível 1, e continuará a partir do mesmo ponto quando este passar para o nível zero.



A, B, C, D, são entradas de carga, passando os flip-flops para o estado inicial quando a entrada de carga (L) está ao nível 0

Q_A , Q_B , Q_C , Q_D são as saídas de cada flip-flop

Figura 4.16. — O contador 74191, que permite definir o estado inicial de cada andar, contagem ascendente e descendente, e possui diversas saídas que servem para detectar a progressão da contagem.

Duas saídas muito úteis neste modelo particular são a *máx./mín.* e a de *relógio*. A saída *máx./mín.* fornece um impulso positivo mais longo do que o impulso de relógio em cada comutação do bit mais significativo. Os impulsos desta saída são concebidos de tal modo que um tem um flanco descendente coincidente com o flanco ascendente do impulso Q , e o seguinte tem o seu flanco descendente coincidente com o flanco descendente desse impulso (figura 4.17). Podem ser usados para detectar o final da contagem, quer se trate de um valor máximo ou um valor mínimo. A saída de relógio produz um impulso negativo no final da contagem (com a duração normal dos impulsos de relógio) que pode ser usado como transporte ao andar de contagem seguinte.



Figura 4.17. — Saída máx./mín. do 74191.

Contadores síncronos

As desvantagens do contador que acabamos de descrever já foram indicadas anteriormente, em particular no que se refere ao problema posto pelo tempo necessário para que o impulso passe por todos os andares de um contador. Um método alternativo de contar é oferecido pelo contador síncrono, no qual cada flip-flop é comutado pelo mesmo impulso de entrada, verificando-se todas as alterações ao mesmo tempo. A concepção de um contador síncrono com poucos andares é bastante fácil, mas torna-se mais difícil quando são necessários muitos andares. Felizmente, existem contadores síncronos LSI para os tipos de contagem mais necessários, havendo ainda métodos alternativos para contar números que não sejam decimais, BCD ou potências de dois fáceis de usar.

No contador síncrono, em cada estado de contagem os números binários presentes nas saídas de cada flip-flop são usados para definir as entradas J e K do flip-flop seguinte. O padrão de ligações deve ser correcto para que a contagem continue normalmente.

Para compreendermos como este contador funciona, consideremos o contador muito simples de dois andares da figura 4.18, e recordemos a tabela de verdade do flip-flop J—K (figura 4.7). A tabela de estados ao lado do diagrama da figura 4.18 mostra as saídas Q_0 e Q_1 , assim como as tensões presentes J_1 e K_1 .

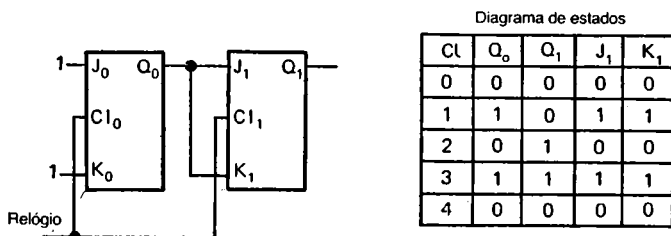


Figura 4.18. — Um contador síncrono de dois andares e a sua tabela de estados. Os impulsos de relógio são alimentados a cada flip-flop, e as ligações J, K determinam a acção em cada impulso de relógio. Na tabela de estados os valores de Q_0 e Q_1 são os que se seguem ao flanco descendente de cada impulso de relógio.

O primeiro impulso de relógio terá como efeito comutar $C1_0$, dado que J_0 e K_0 se encontram permanentemente ligados ao nível lógico 1. Dado que Q_1 (ao nível 0) liga a J_1 e K_1 , no entanto, $C1_1$ não será afectado pelo impulso de relógio. Q_0 não comuta para o nível lógico 1 até chegar o flanco descendente do impulso de relógio, momento em que J_1 e K_1 já terão mudado — estavam em 0 no flanco ascendente do impulso de relógio, pelo que é este o estado que determina a saída de $C1_1$.

No impulso de relógio seguinte, como é óbvio, J_1 e K_1 estão ao nível 1, pelo que a entrada de relógio que leva Q_0

a mudar de 1 para 0 leva também Q_1 a passar de 0 para 1, dando o valor binário 10 (decimal 2). Esta acção deixa J_1 e K_1 ao nível 1, pelo que o impulso de relógio seguinte não afecta $C1_1$, mas altera $C1_0$ de 0 para 1. A contagem binária é agora 11, decimal 3. Com J_1 e K_1 postos em 1 por Q_0 , o impulso de relógio seguinte comutará ambos os flip-flops novamente para 0.

Para aumentar o contador síncrono para mais de dois andares será necessário usar portas além dos flip-flops. Na figura 4.19 mostra-se um contador de três andares, com uma porta AND utilizada para definir as entradas J e K ao terceiro andar. O efeito desta porta consiste em colocar J_2 e K_2 ao nível lógico 1 apenas quando Q_0 e Q_1 estão ao nível 1.

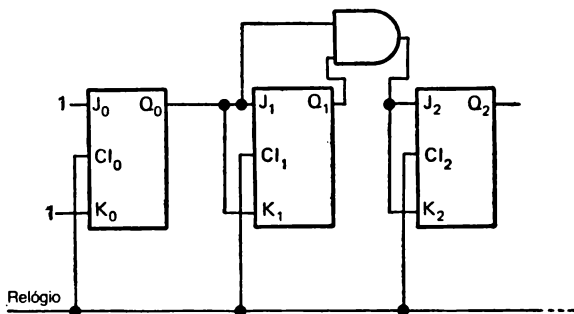


Figura 4.19. — Um contador síncrono de três andares.

A contagem continua portanto da maneira descrita no caso anterior até $Q_0 = 1$ e $Q_1 = 1$. Neste ponto, a acção da porta AND mantém J_2 e K_2 ao nível lógico 1, de tal modo que o impulso de relógio seguinte, que leva novamente Q_0 e Q_1 para o nível 0, comutará Q_2 , dando o número 100 (decimal 4). Os três impulsos seguintes não terão efeito, visto que a saída da porta AND é reduzida, e portanto Q_2 não pode comutar. Ao decimal 7, no entanto, Q_0 e Q_1 encontram-se novamente ao nível 1, pelo que o oitavo impulso comuta novamente $C1_2$.

A figura 4.20 mostra um contador síncrono de quatro andares; convida-se o leitor a verificar a pilotagem de porta ao quarto andar. Os contadores síncronos, tal como os anteriores, podem ser levados a contar em qualquer sentido, e a contagem descendente com detecção do zero é, tal como antes, um método mais simples de contar. Os contadores síncronos podem ser concebidos de modo a voltarem ao estado inicial depois de um dado número, o que se consegue ligando apenas as entradas J e K à porta, mas o método não é de modo algum simples quando os contadores contêm uma grande quantidade de andares. O método de concepção habitual consiste em escrever o estado de cada saída Q para cada número contado, e depois os números J e K do estado seguinte, de tal modo que se possam montar portas que produzam a combinação adequada de estados J e K. Este processo é aborrecido, e a sua versão gráfica, designada *método Karnaugh*, só é de facto útil quando estão a ser concebidos relativamente poucos andares, ou quando se deseja usar códigos diferentes do simples binário.

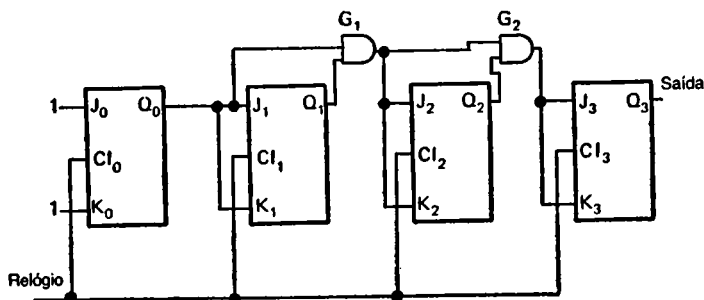


Figura 4.20. — Um contador síncrono de quatro andares.

Os contadores raramente são construídos a partir de flip-flops, visto que existem no mercado contadores em forma integrada. Por outro lado, podem-se usar registos de deslocamento (ver o capítulo 5) como contadores em muitas aplicações. No entanto, a construção de um contador usando flip-

-flops constitui um exercício muito útil porque é possível verificar e seguir a acção — o que raramente é possível quando se utiliza um circuito integrado!

Funcionamento prático

A montagem prática de contadores requer uma fonte de impulsos de relógio. É útil um comutador controlado por botão de pressão quando interessa analisar a acção de cada impulso de relógio; um oscilador de baixa frequência é mais útil quando se necessita da aplicação de impulsos a maior velocidade. Uma das «introduções» práticas mais úteis à técnica aqui usada é o flip-flop J-K duplo 7476 — este circuito integrado é barato e fácil de usar. O diagrama de distribuição dos seus terminais é indicado na figura 4.21. Na maior parte das aplicações a baixas velocidades, é possível deixar os terminais «preset» e «reset» não ligados — se forem ligados a linhas comuns deve-se ter o cuidado de assegurar que não estejam nunca *ambos* a baixo nível.

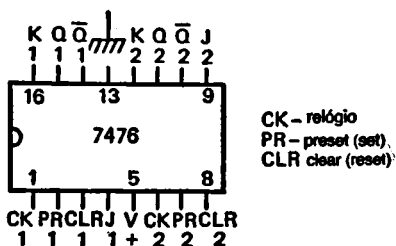
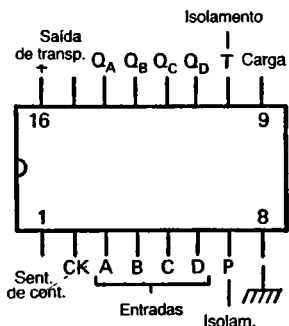


Figura 4.21. — Distribuição dos terminais do flip-flop J-K duplo 7476.

Pode-se construir toda a gama de contadores usando flip-flops J — K 7476; se se pretende investigar o funcionamento de um contador síncrono de quatro bits, o 74168, contador síncrono de contagem ascendente e descendente, é também muito útil. A distribuição dos seus terminais é apresentada

na figura 4.22. O sentido de contagem é ascendente quando o terminal ascendente/descendente está ao nível lógico 1, e descendente no caso contrário. As entradas de «isolamento», designadas «P» e «T», constituem uma porta que impede a contagem quando ambas estão ao nível 1.



As entradas de isolamento P e T devem ser baixas para que a contagem continue

O flanco descendente da saída de transporte coincide com o flanco ascendente e descendente de Q_D

As entradas A, B, C, D permitem ao contador ser colocado em qualquer número de 4 bits quando a entrada de carga está ao nível 0

Figura 4.22. — Distribuição de terminais no contador 74168.

REGISTOS DE DESLOCAMENTO

Um registo de deslocamento é um circuito que utiliza flip-flops ligados em linha de tal modo que cada bit armazenado na saída Q seja deslocado para o flip-flop seguinte sempre que é aplicado um impulso de relógio. Os impulsos de relógio são aplicados a todos os flip-flops do registo, de tal modo que a acção é síncrona. Pode ser usado qualquer tipo de flip-flop cujo funcionamento seja apropriado (disparados pelo flanco ascendente), e a figura 5.1 mostra circuitos típicos utilizando tipos D, tipos S — R controlados por relógio, e flip-flops J — K master-slave.

Para compreender o funcionamento destes circuitos deve-se recordar a tabela de verdade de cada flip-flop, e ter presente que a comutação da saída ocorre demasiado tarde para que possa afectar o flip-flop seguinte, qualquer que seja o tipo usado. Imaginemos que os circuitos da figura 5.1 foram colocados numa posição tal que todas as saídas Q se encontram ao nível 0, e que a entrada de informação do flip-flop 0 se encontra ao nível lógico 1.

O flip-flop de tipo D, com um 1 na sua entrada D, comutará no flanco ascendente do impulso de relógio, mas o atraso assegurará que a sua saída Q não afecta a entrada do flip-flop seguinte e portanto não modificará a saída Q_1 depois do flanco ascendente do impulso de relógio. Do mesmo modo, a entrada $R = 1$, $S = 0$ (devido ao inversor) do dispositivo R — S assegura que $Q_0 = 0$, $\overline{Q}_0 = 1$ pouco depois do flanco ascendente do impulso de relógio, mas que isto não tem qualquer efeito no flip-flop 1. O J — K é afectado de maneira semelhante, com $J_0 = 1$, $K_0 = 0$ no momento do primeiro

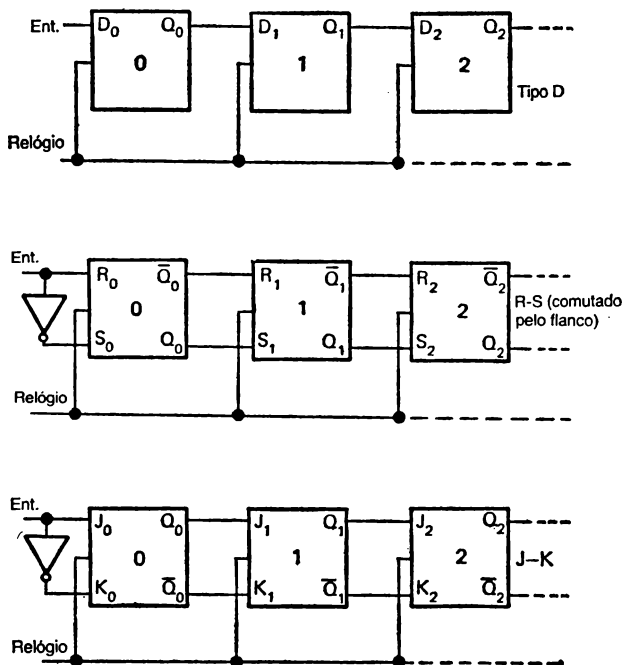


Figura 5.1. — Circuitos de registros de deslocamento, usando três tipos diferentes de flip-flops. Note-se que é a saída \bar{Q} do flip-flop R — S que pilota a entrada R seguinte.

impulso de relógio, $Q_0 = 1$ e $Q_1 = 0$ no flanco descendente do impulso de relógio, preparando o segundo flip-flop J — K para o impulso de relógio seguinte.

Qualquer que seja o tipo de circuito flip-flop usado, portanto, a acção do primeiro impulso de relógio no primeiro flip-flop consiste em comutar a saída do seu estado original (nível lógico 0) para o nível 1, a tensão da entrada de informação. A este primeiro impulso de relógio, cada um dos outros flip-flops da cadeia tem na entrada um nível zero produzido pelo andar anterior ($R = 1$, $S = 0$ no tipo R — S),

pelo que o primeiro impulso de relógio deixa todos os flip-flops no mesmo estado (saída zero) excepto o primeiro. Ver a figura 5.2.

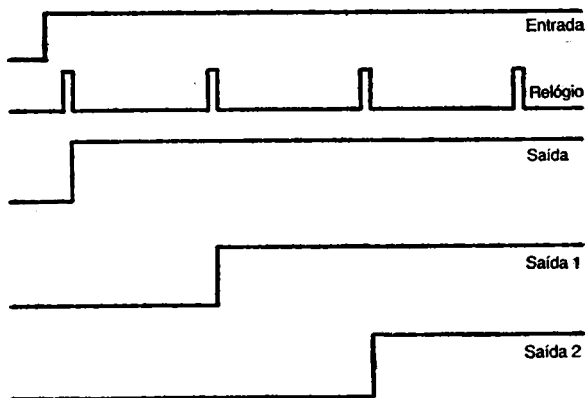


Figura 5.2. — Ilustração da acção de deslocamento do registo.

Quando ocorre o segundo impulso de relógio, no entanto, estando ainda a entrada ao primeiro flip-flop ao nível 1, a entrada a cada um dos segundos flip-flops é também 1 ($R = 0, S = 1$ no tipo R — S). Nestas condições, quando o segundo impulso ocorre, a saída do flip-flop 1 ficará inalterada, mas a do flip-flop 2 comutará para o nível 1. Os flip-flops seguintes não serão afectados porque quando ocorre o flanco ascendente do impulso de relógio têm entradas zero.

Ao terceiro impulso de relógio, e partindo do princípio de que a entrada é ainda mantida ao nível lógico 1, o terceiro flip-flop de cada cadeia comutará, deixando o quarto ao nível 0. Este comutará, por sua vez, no quarto impulso de relógio.

Os registos de deslocamento constituem uma categoria muito importante de componentes em todos os tipos de circuitos digitais. Como a saída do flip-flop só é alterada por um impulso de relógio depois de a entrada o ter sido, a remoção dos impulsos de relógio, deixando apenas presente

a tensão de alimentação, deixa a saída do flip-flop inalterada durante todo o tempo em que as condições se mantiverem. Cada flip-flop pode portanto ser levado a armazenar um dígito binário (ou bit) durante o tempo em que for alimentado e não receber quaisquer impulsos. É possível armazenar um conjunto de dígitos binários num registo que tenha um flip-flop para cada dígito. É portanto normal usar-se registos de deslocamento sempre que se guardam dígitos entre duas operações, sendo possível encontrar vários exemplos da sua utilidade neste capítulo e no capítulo 7, que trata dos microprocessadores.

Antes de examinarmos alguns circuitos de registos de deslocamento, devemos notar que existem quatro tipos básicos de registos. O tipo apresentado na figura 5.1 é um tipo «entrada em série — saída em série» (SISO). A tensão lógica na entrada é alimentada ao registo de deslocamento em cada impulso de relógio, e pode modificar-se no período de tempo entre impulsos. Depois de ser aplicado um número de impulsos igual ao de flip-flop usados no registo, obtém-se o primeiro bit na saída. Um registo SISO utilizado deste modo pode actuar como um dispositivo de atraso — os bits apresentados na saída encontram-se atrasados de um certo número de impulsos de relógio (igual ao número de flip-flops usados) relativamente aos bits apresentados à entrada.

Um outro tipo básico de registo de deslocamento é o de «entrada em paralelo, saída em paralelo» (PIPO), que pode ser visto na figura 5.3. Neste tipo, os bits de informação são carregados em cada flip-flop em paralelo, usando os terminais set/reset em vez dos impulsos de relógio. Se não

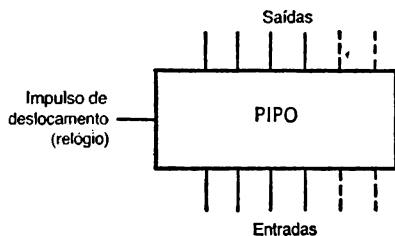


Figura 5.3. — Um registo PIPO.

forem aplicados impulsos, como é óbvio, os bits não serão deslocados, e a leitura nos terminais Q será igual às entradas. Esta maneira de usar o registo é ótima para um armazenamento temporário de alguns bits. Se são aplicados impulsos de relógio, cada bit será deslocado de um flip-flop para cada impulso de relógio, e a leitura feita na saída já não será igual à entrada. Fica-nos agora o problema de saber como se utiliza a entrada e saída em série. Uma possibilidade consiste em ligar a entrada ao nível lógico 0 ou 1, e ignorar a saída. Neste tipo de funcionamento, o registo encher-se-á com zeros e uns depois de um conjunto completo de impulsos de relógio. Outra possibilidade consiste em promover uma rotação, ligando a saída final do registo à entrada em série, de tal modo que a saída seja realimentada à entrada em cada impulso de relógio; neste caso, um conjunto completo de impulsos de relógio conduzirá novamente o registo ao estado inicial.

Os dois tipos restantes de registo de deslocamento combinam métodos em série e em paralelo. O registo PISO (entrada em paralelo, saída em série) (figura 5.4) utiliza os terminais set/reset de um flip-flop J—K para carregar bits em cada flip-flop independentemente e simultaneamente. Os dados são depois deslocados de um bit cada vez quando se aplicam impulsos de relógio. Isto torna possível converter dados que foram apresentados em forma paralela (em diversas linhas simultaneamente) para a forma serial (um bit a seguir

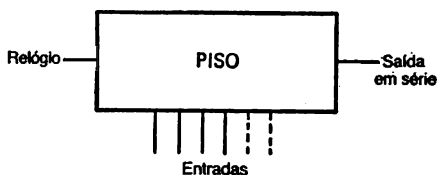


Figura 5.4. — Um registo PISO.

ao outro) e transmiti-los ao longo de uma linha única. A conversão para a forma serial é necessária no caso de os bits deverem ser transmitidos ao longo de linhas telefónicas, gravadas em fita, enviados a um monitor video (ver o capítulo 6) e usados para alimentar uma tele-impressora.

O registo SIPO, de entrada em série e saída em paralelo (figura 5.5) realiza a função contrária. Neste tipo os dados são apresentados sob a forma de um bit de cada vez, e deslocados em cada impulso de relógio. Depois de ser aplicado um conjunto completo de impulsos de relógio o registo encontra-se cheio, podendo o conteúdo ser lido nos terminais Q ou descarregado para um conjunto de linhas paralelas. Neste sentido, «descarregar» significa apenas que os bits podem ser usados para accionar portas ou outros circuitos — os registos não são alterados de qualquer modo por esta acção. Utilizando o registo SIPO, os bits de dados que foram transmitidos em sequência por uma linha única podem ser juntados de modo a constituírem uma «palavra» de vários bits.

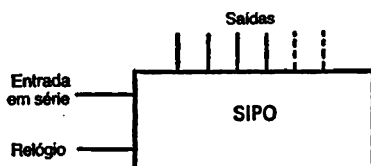


Figura 5.5. — Um registo SIPO.

Um exemplo simples do uso de um registo SIPO é o contador de indicação decimal, utilizando um registo de deslocamento de 10 andares. Carrega-se um nível lógico 1 no registo no último flip-flop, e um nível lógico 0 em todos os outros flip-flops. A saída em série é então realimentada à entrada em série, como se mostra na figura 5.6, e cada saída em paralelo é ligada a um «display» (ver o capítulo 6 quanto a pormenores). Como se mostra, antes de chegar um impulso de relógio, o 1 armazenado no último flip-flop ilumina o 0 do «display». O primeiro impulso de relógio deslocará então a tensão 1 para o flip-flop A, pelo que a saída de flip-flop A é agora 1, iluminando-se o algarismo 1. Do mesmo modo, o segundo impulso de relógio deslocará o bit 1 de modo a ser apresentado na saída o dígito 2, etc. Pode-se conseguir o transporte para um segundo andar de contagem ligando a saída do contador zero à entrada de relógio de outro registo

semelhante. Dado que o registo de deslocamento de 10 andares pode ser construído num único circuito integrado, este método de descodificação de um valor decimal é mais simples do que os métodos binário/BCD que serão descritos adiante, mas só se dispõe de um sistema de «display» apropriado.

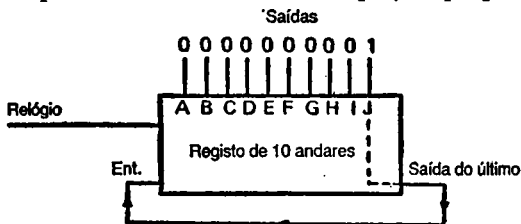


Figura 5.6.—Um contador de décadas (decimal) utilizando um registo de 10 andares — mostra-se o exemplo de um contador em anel.

Contadores Johnson

O contador Johnson é um circuito que utiliza registos de deslocamento para realizar uma contagem síncrona sem problemas de concepção. São possíveis elevadas velocidades de contagem, e a descodificação pode ser relativamente simples, ao contrário do que acontece num contador binário ou BCD. O princípio de funcionamento do contador Johnson consiste em ligar um registo de deslocamento à saída final realimentada à entrada em série, mas com uma inversão (figura 5.7). O contador Johnson simples possui um comprimento de contagem máximo de $2n$, onde n é o número de flip-flops usado no registo de deslocamento. Este valor é evidentemente diferente dos 2^n de um contador binário, e de n de um contador em anel simples.

O funcionamento do contador Johnson, ilustrado na figura 5.8, é a seguinte. Imaginemos que o estado de partida é constituído por tensões zero em todas as saídas. A inversão da saída final colocará portanto um 1 lógico na entrada, pelo que o primeiro impulso de relógio carrega um 1 no flip-flop A. Quando ocorre o segundo impulso de relógio as condições

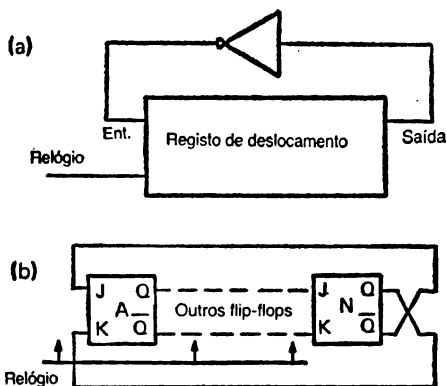
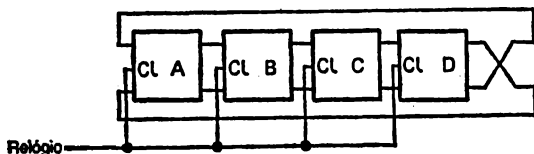


Figura 5.7. — O contador Johnson (a) é também conhecido pelo nome de «contador em anel invertido», devido à inversão (b) que pode ser realizada por cruzamento das ligações Q—J e Q—K entre o último flip-flop e o primeiro. Quando se usa um registo de deslocamento integrado, esta inversão deve ser realizada por um inversor separado.

de entrada do flip-flop A não são alteradas, e o 1 presente neste flip-flop será deslocado para o flip-flop B. Esta acção continua, carregando-se uns no flip-flop A de cada vez, até cinco impulsos de relógio (neste exemplo) conseguirem encher o registo com uns. A inversão da saída final assegura agora a passagem de um zero para a entrada em série, pelo que o impulso de relógio seguinte carregará um 0 no flip-flop A, e cada impulso seguinte deslocará este zero e introduzirá um novo 0 no registo, pelo que este fica novamente preenchido com zeros. A figura 5.8 mostra a tabela de verdade, que indica igualmente as saídas que devem ser ligadas a portas a fim de tornar possível o conhecimento do estado da saída. Note-se que cada número contado pode ser detectado por uma porta de duas entradas.

O contador Johnson simples, devido ao facto de o número de contagens ser igual ao dobro do número de flip-flops usados, pode ser utilizado apenas para contagens pares, pois o dobro de qualquer número é sempre um número par. Para



Número do impulso de relógio	Q_A	Q_B	Q_C	Q_D	Descodificação
0	0	0	0	0	$\bar{A} \bar{D}$
1	1	0	0	0	$A \bar{B}$
2	1	1	0	0	$B \bar{C}$
3	1	1	1	0	$C \bar{D}$
4	1	1	1	1	$A D$
5	0	1	1	1	$\bar{A} B$
6	0	0	1	1	$\bar{B} C$
7	0	0	0	1	$\bar{C} D$

Figura 5.8 — Um contador Johnson de quatro andares e a sua tabela de verdade. A coluna de descodificação mostra quais as quantidades que são tratadas por uma porta AND de modo a formar as oito saídas.

contagens ímpares, é necessário acrescentar circuitos de porta que permitam omitir o estado em que todos os flip-flops têm o valor 1. Isto pode ser feito utilizando a porta para detectar o estado imediatamente anterior a este. No caso de um registo de cinco andares esse estado será o 11110, pelo que são ligados à entrada uma porta AND de quatro entradas e um inversor (que inverte o 0). No exemplo do contador até nove apresentado na figura 5.9 utilizam-se portas NAND, pelo que é necessário inverter novamente a entrada.

Dado que os flip-flops de um registo de deslocamento podem passar para o valor lógico 1 ou 0 quando é aplicada energia, é necessário algum método de regular inicialmente o contador Johnson, antes de a contagem se iniciar. Se o registo de deslocamento for de um tipo que permita a limpeza de cada andar será simples fazê-lo, em particular se a limpeza poder ser realizada automaticamente quando se liga a alimenta-

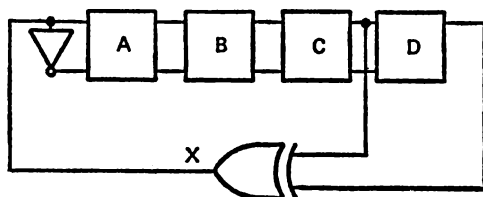


Figura 5.10 — Um contador não binário de comprimento máximo. Os flip-flops do contador fazem normalmente parte de um registo de deslocamento integrado.

de um contador de quatro andares é apresentado na figura 5.10, e a sequência de contagem respectiva na figura 5.11. Note-se que a acção de contagem será parada se o registo ficar cheio de zeros em qualquer momento, pelo que é necessário utilizar uma entrada preset (set) ou um circuito de porta para impedir que isto aconteça quando se liga.

Conta- gem	A	B	C	D	X
1	1	1	1	1	0
2	0	1	1	1	0
3	0	0	1	1	0
4	0	0	0	1	1
5	1	0	0	0	0
6	0	1	0	0	0
7	0	0	1	0	1
8	1	0	0	1	1
9	1	1	0	0	0
10	0	1	1	0	1
11	1	0	1	1	0
12	0	1	0	1	1
13	1	0	1	0	1
14	1	1	0	1	1
15	1	1	1	0	1

Figura 5.11 — Sequência de contagem do circuito da figura 5.10.

Para compreender o funcionamento, imaginemos que o registo se encontra cheio de uns. Dado que as entradas à porta X — OR estão ambas ao nível 1, a saída da porta é 0, e este 0 será deslocado para o primeiro flip-flop do registo. As contagens seguintes terão o mesmo efeito, mas à quarta contagem a acção de deslocamento deixou apenas uma única entrada 1 na porta X — OR. Com esta entrada, a saída da porta X — OR é também 1, pelo que na contagem seguinte é carregado um 1. Entretanto, as entradas à porta X — OR passaram a 0, pelo que o dígito carregado a seguir é também 0. Este ciclo continua então até aquele estado se repetir, depois de uma contagem de 14 (decimal).

A figura 5.12 mostra quais os andares que devem alimentar a porta X — OR para registos de deslocamento de vários andares, até um máximo de 12. Podem-se obter contagens mais curtas utilizando portas adicionais que levam o contador a saltar um ou mais estados. As ligações são feitas de tal modo que é apenas necessário carregar um dígito diferente na entrada. Por exemplo, para encurtar a contagem do circuito da figura 5.10 em três posições, a contagem pode saltar de

N.º de fases	Fases para a alimentação da porta X-OR	Contagem
3	BC	7
4	CD	15
5	CE	31
6	EF	63
7	FG	127
8	DE FH	255
9	EI	511
10	GJ	1023
11	IK	2047
12	FHL	4095

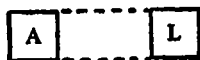


Figura 5.12 — Ligações entre as saídas do registo de deslocamento e a porta X — OR para produzir contagens de diferentes comprimentos.

1000 para 1100 desde que se carregue um 1 extra. O estado 1000 é o único em que as saídas dos flip-flops B, C e D são todos zeros, pelo que pode ser montada numa ligação OR com uma porta X — OR uma porta que detecte este estado de modo a produzir o salto referido. Para contagens mais longas a ligação pode ser muitas vezes simplificada utilizando álgebra booleana para descrever a condição de salto.

Conversão binária, decimal e BCD

Devido à sua simplicidade, a aritmética binária é usada extensivamente nos circuitos digitais, mas os displays são geralmente de números decimais (ver também o capítulo 6). Para apresentar números decimais é necessário algum tipo de conversão dos números binários para BCD. Isto não é difícil de realizar, utilizando um registo de deslocamento, desde que as regras para conversão de binário para BCD sejam conhecidas. As regras são:

- 1) O número binário é deslocado, começando pelo bit mais significativo, para o registo de deslocamento, cujos flip-flops estão agrupados em conjuntos de quatro.
- 2) Cada grupo de flip-flops do registo BCD representa uma potência de 10 num número decimal, com os bits binários deslocados primeiramente para o grupo de ordem inferior.
- 3) Se, antes de um impulso de deslocamento, um grupo de quatro flip-flops contiver um número inferior a cinco (decimal), o deslocamento realiza-se normalmente.
- 4) Se o número armazenado num grupo de quatro flip-flops for entre cinco e nove, inclusivamente, é somado o binário três (0011) ao número armazenado antes de ser deslocado.

A figura 5.13 mostra um exemplo de aplicação destas regras. O número a converter é o binário 11010 (decimal 26), e é fácil compreender a maneira como os bits entram, começando pelo bit mais significativo, num par de registos de deslocamento de quatro bits que guardará os números BCD.

	<i>Registos de deslocamento</i>		<i>Número</i>
	<u>8</u> <u>4</u> <u>2</u> <u>1</u>	<u>8</u> <u>4</u> <u>2</u> <u>1</u>	
			1 1 0 1 0
1.º deslocamento	0 0 0 0	0 0 0 1	1 0 1 0
2.º »	0 0 0 0	0 0 1 1	0 1 0
3.º »	0 0 0 0	0 1 1 0	1 0
Somar 3	0 0 0 0	0 0 1 1	1 0
Total	0 0 0 0	1 0 0 1	1 0
4.º deslocamento	0 0 0 1	0 0 1 1	0
5.º »	0 0 1 0	0 1 1 0	

Figura 5.13. — Conversão de binário para BCD, utilizando o método de «somar 3».

Ao primeiro deslocamento, o bit 1 que entra no registo BCD de ordem inferior produz um total de um neste registo, e o segundo bit deslocado para ele produz uma contagem de três. Isto não requer qualquer correcção, mas o bit que entra em seguida, um 0, produz o total 110 (decimal 6), que requer agora uma correcção sob a forma de uma soma de 011 (decimal 3). Isto torna o total existente no registo igual a 1001. O impulso de deslocamento seguinte movimenta o 1 para o registo BCD seguinte, sem necessidade de qualquer correcção. Penetra igualmente um 1 no registo de menor ordem, ainda sem necessidade de correcção.

O último deslocamento apresenta um 0 em cada registo, e como este é o último impulso (o número de impulsos de relógio é igual ao número de dígitos do número a converter) não é necessária qualquer adição ao número presente no registo de ordem inferior apesar de já ter atingido 0110 (decimal 6). O estado de registo é agora 0010 0110, o equivalente BCD de 26.

Para utilizar estas regras em «hardware», tornam-se necessários três tipos de circuitos. Um é, como seria de esperar, o registo de deslocamento, outro um somador (o funcionamento

de um somador é discutida no capítulo 7), e o terceiro uma porta que detecta o momento em que o conteúdo de um grupo de quatro dígitos BCD é cinco ou mais.

A figura 5.14 mostra o circuito de um conversor binário para BCD de quatro décadas, da Texas Instruments. O registo de deslocamento é um 74199, que é um tipo PIPO de oito bits. O terminal 23 deste circuito particular é uma entrada de deslocamento/carga; um nível lógico 1 neste terminal levará o circuito integrado a deslocar para a direita para cada impulso de relógio, e com uma entrada lógica 0 o registo pode ser carregado em paralelo nas entradas A a H. Para a conversão binário-BCD o terminal 23 é mantido ao nível lógico 0.

O número binário é introduzido na entrada do primeiro 74199, começando pelo bit mais significativo. Isto pode ser feito obtendo os dígitos a partir de outro registo, utilizando uma direcção de deslocamento que assegure a entrada deste bit mais significativo em primeiro lugar. Para cada impulso de relógio entra um novo bit, e os bits restantes são deslocados (para a direita do diagrama da figura 5.14), mas *através do somador*, como se nota em seguida.

As saídas de cada grupo de quatro são ligadas à porta integrada 7452. Esta contém quatro portas AND cujas entradas (A e B) se encontram ligadas a uma porta OR que possui ainda uma entrada de «isolamento». As entradas vindas do registo de deslocamento são ligadas de tal modo que a saída do 7452 seja 1 para $(Q_A \text{ AND } Q_C) \text{ OR } (Q_B \text{ AND } Q_C) \text{ OR } Q_D$ igual a 1. Em álgebra booleana, isto equivale a $Q_A \cdot Q_C + Q_B \cdot Q_C + Q_D$. São estas as condições para que o número armazenado nos quatro flip-flops do registo de deslocamento totalize 5 ou mais, porque se $Q_D = 1$ é armazenado o decimal 8, e se Q_B e Q_C forem 1, o número armazenado é 6. Se Q_A e Q_C forem 1, o número armazenado é 5. Para os números 7 e 9, Q_C , ou Q_B , serão 1, pelo que a condição Q_B e Q_C iguais a 1 se encontra preenchida. Se for armazenado um 9 (1001), $Q_D = 1$ e $Q_A = 1$, pelo que a saída da porta é também 1.

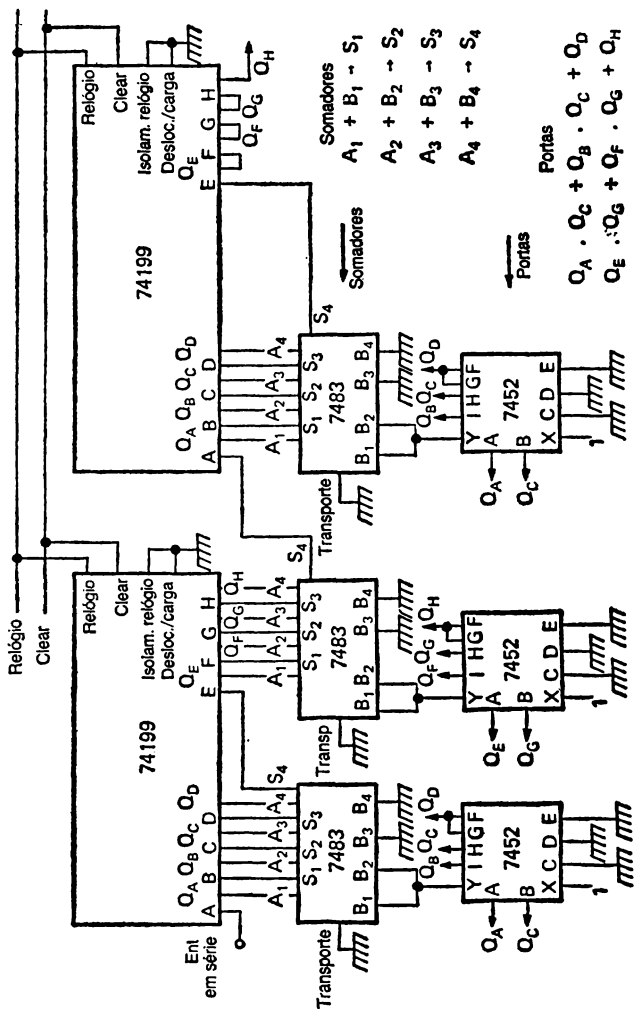


Figura 5.14. — Conversão de binário para BCD utilizando registros de deslocamento, somadores e portas.

O efeito de uma saída 1 do 7452 é aplicado às entradas B_1 e B_2 do somador (7483). Ora estes são o bit menos significativo e o que se lhe segue, que serão somados aos dois bits de ordem inferior do número já guardado no registo de deslocamento, pelo que o número 0011 (decimal 3) é somado ao número existente no registo. A acção do terminal deslocamento/carga, como se recorda, consistiu em permitir apenas um carregamento em paralelo; o registo não desloca para cada impulso de relógio. O que acontece é que os dígitos nas entradas A e B do somador são adicionados, e a soma é apresentada nas saídas S, com um transporte para o dígito seguinte. O deslocamento verifica-se porque a soma de Q_A e da saída da porta 7452 é a entrada ao flip-flop B, a soma de Q_B e da saída da porta é a entrada ao flip-flop C, etc. Quando já foi aplicado um número de impulsos de relógio igual ao número de bits, os registos contêm o número BCD equivalente ao número binário.

Um outro método de conversão binário-BCD utiliza um circuito integrado desenvolvido para o efeito, o 74185. Dado que este circuito integrado é uma memória e não um registo de deslocamento, não trataremos dele aqui.

A conversão de BCD para binário é igualmente importante, porque normalmente será utilizado um teclado decimal para dar uma saída BCD de cada dígito, e esta saída total deve ser convertida em binário puro a menos que se disponha de circuitos capazes de realizarem uma aritmética BCD. Como se poderia esperar da acção necessária para converter binário em BCD, a conversão BCD-binário envolve um deslocamento e uma subtração. Os dígitos do número BCD são, evidentemente, agrupados em conjuntos de quatro e deslocados para a direita em cada impulso de relógio. Quando um 1 lógico passa de um grupo de quatro para o seguinte, é necessário subtrair o decimal 3 (0011). Isto é feito adicionando o complemento para 2 do número em causa, que será 1101 na forma BCD, e omitindo qualquer bit de transporte. As fases de uma conversão normal são indicadas em detalhe na figura 5.15. O número a converter neste exemplo é 0011 0101. A figura 5.15 mostra este número escrito em

	BCD <i>high digit</i>	BCD <i>low digit</i>	<i>Registo da resposta</i>	
	0 0 1 1	0 1 0 1		
1.º deslocamento	0 0 0 1	1 0 1 0	1	
— 3		1 1 1 0	1	
Ign. transporte		0 1 1 1	1	
2.º deslocamento	0 0 0 0	1 0 1 1	1 1	
— 3		1 1 0 1	1 1	
Ign. transporte		1 0 0 0	1 1	
3.º »	0 0 0 0	0 1 0 0	0 1 1	
4.º »		0 0 1 0	0 0 1 1	
5.º »		0 0 0 1	0 0 0 1 1	
6.º »		0 0 0 0	1 0 0 0 1 1	
7.º »		0 0 0 0	0 1 0 0 0 1 1	
8.º »		0 0 0 0	0 0 1 0 0 0 1 1	res. final

Figura 5.15. — Conversão BCD-binário, usando o método «subtrair 3».

dois grupos; na prática, como é óbvio, um registo de deslocamento de oito bits pode guardar todos os dígitos, mas só pode ser usado se for possível realizar a adição de 1101 nos momentos apropriados.

Ao primeiro deslocamento, é movido um 1 do grupo de dígitos de ordem superior para o de ordem inferior, sendo deslocado um outro 1 para fora do registo BCD de ordem inferior. De acordo com as regras de conversão, soma-se agora 1101 ao número BCD do grupo de ordem inferior, sendo ignorado o transporte. Note-se que nada é acrescentado ao dígito que foi deslocado para a saída, pois este já não faz parte do número BCD.

O impulso de relógio seguinte desloca outro 1 para o registo de ordem inferior, sendo mais uma vez necessário somar 1101 a este registo, ignorando o transporte. Todos os uns foram agora deslocados para fora do registo de ordem superior, pelo que resta apenas realizar deslocamentos para a direita até terem sido usados os oito impulsos. O resultado final é 0010 0011, correspondente a 35.

A figura 5.16 mostra um circuito que realiza a conversão a partir de um contador de quatro décadas. O 74199 é um registo de deslocamento PIPO de oito bits com um controlo deslocamento/carga, como se descreveu anteriormente, e o 7483 é um somador binário de quatro bits. A sequência de funcionamento é a seguinte:

Os registos são limpos, usando um 0 na linha «clear» comum, e a entrada deslocamento/carga (terminal 23) de cada registo é colocada ao nível lógico elevado, de modo a que se verifica um deslocamento para a direita em cada impulso de relógio. O número BCD é então apresentado em série à entrada do contador, começando pelo bit mais significativo, utilizando 16 impulsos de relógio. Se o número BCD foi obtido a partir de um teclado, será necessário um par de registos de oito bits para armazenar os algarismos até serem apresentados na entrada.

A entrada de deslocamento/carga é então passada para o nível lógico 0, de tal modo que os registos fiquem prontos para uma entrada em paralelo. O bit mais significativo do número de quatro décadas encontra-se agora em Q_H no registo 2, e o menos significativo em Q_A no registo 1. O primeiro impulso de relógio alimentará o bit menos significativo à saída binária (Q_A registo 1) e cada bit existente nos registos será alimentado ao andar da esquerda (no diagrama) dadas as ligações que existem entre as saídas e as entradas. Por exemplo, o bit mais significativo em Q_H será alimentado à entrada G, o bit em Q_G à entrada F, etc., até ao segundo registo. O bit que passa para o grupo seguinte encontra-se em Q_E , e é alimentado ao somador, usando a entrada A_4 juntamente com as entradas B_1, B_3, B_4 . Se o bit em Q_E for zero, as entradas B ao somador serão todas 0, pelo que nada será acrescentado. Se o bit em Q_E é 1, $B_4 = B_3 = B_1 = 1$, e B_2 é colocado a zero, dando o número 1101 que será acrescentado ao número presente nas entradas A.A. O bit soma de A_4, B_4 encontra-se em S_4 , e é carregado à entrada D; a soma de A_3, B_3 estará em S_3 e é carregada à

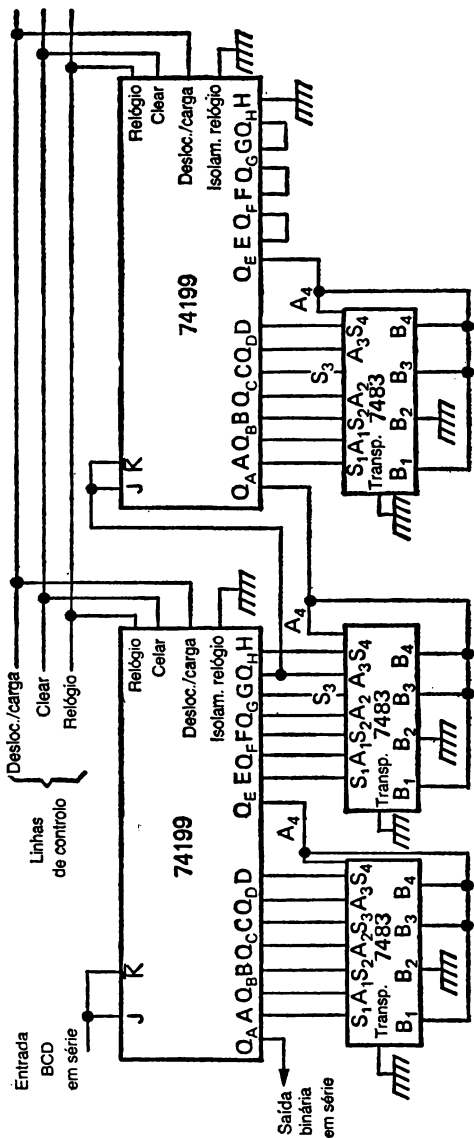


Figura 5.16. — Conversão BCD — binário utilizando registros de deslocamento e somadores.

entrada C, etc. Não há transporte ao somador, ficando o dígito de transporte desligado. A ligação somadora assegura automaticamente o acréscimo do factor corrector 1101 de cada vez que é passado um bit 1 ao longo de um grupo de quatro, pelo que o bit alimentado à saída Q_A no registo 1, em cada impulso de relógio, é um bit de um número binário de 16 bits.

Selectores de dados e descodificadores

Os selectores de dados («multiplexer») e os descodificadores («demultiplexer») não são circuitos de registos de deslocamento, mas são frequentemente usados com estes. Um selector de dados permite a ligação de um bit entre muitos outros a uma saída através da selecção de um código. Um selector com uma única saída pode realizar a conversão paralelo-série que normalmente é efectuada por um registo de deslocamento (SIPO); ao contrário deste, no entanto, o selector não se limita a bits em sequência.

A figura 5.17 mostra a distribuição dos terminais do selector de dados de 16 bits, referência 74150. Existem nele 16 terminais de entrada, a cada um dos quais se pode aplicar um bit. O sinal «strobe» no terminal 0 corta todas as portas

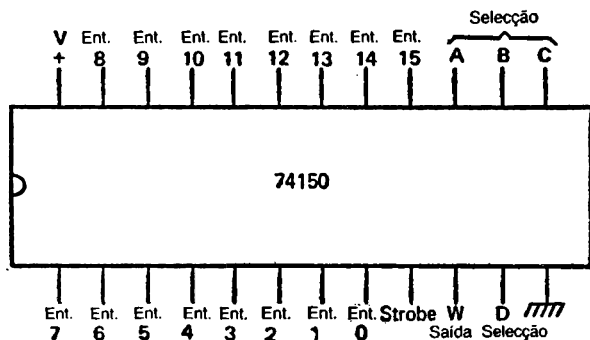


Figura 5.17. — Distribuição dos terminais de um selector de dados 74150 («multiplexer»).

(lógica 1 na entrada de «strobe») ou permite uma saída (lógica 0); os quatro terminais de selecção de dados, 11, 13, 14 e 15 são usados para «endereçar» a entrada que deve ser transmitida ao terminal de saída único. Um 0 em cada terminal de selecção, por exemplo, dará um endereço de 0000, escolhendo o número de entrada 0 (no perne 8) para ligação à saída. A escolha de $A = 1$, $B = 0$, $D = 0$ e $C = 0$ dará o endereço 0001, sendo portanto escolhida a entrada 1. O terminal D de selecção de dados corresponde à maior ordem do endereço, A à menor ordem deste. Podem ser usados vários grupos de selectores de dados a fim de que as respectivas saídas formem uma «palavra». Por exemplo, podem-se usar quatro selectores 74150 de modo a que possa ser escolhido nas saídas um qualquer dígito BCD num total de 16, obtendo-se um bit de cada selector, com as linhas de endereço em paralelo.

Um descodificador («demultiplexer») realiza a função oposta. As linhas de entrada consistem numa ou mais linhas de dados e nas habituais linhas de endereço. Por exemplo (figura 5.18), o descodificador 74154 possui 16 terminais de saída, um terminal de entrada de informação, uma entrada de «isolamento» e quatro entradas de endereço. Quando os terminais de «isolamento» e de entrada de dados se encontram ao nível lógico 0, obter-se-á um zero de saída no terminal cujo número corresponde ao número de endereço formado pelas quatro linhas de endereço. As saídas de todos os outros terminais encontram-se ao nível lógico 1. Tal como anteriormente, o terminal de endereço D constitui a linha mais significativa do endereço de quatro bits, e A a menos significativa. Uma entrada 1 no terminal de «entrada de dados» produz sinais ao nível lógico 1 em todos os terminais de saída, e uma entrada ao nível 1 no terminal de isolamento produz uma saída semelhante, independentemente dos sinais de facto aplicados ao terminal de entrada de dados.

Note-se que o descodificador que estamos a discutir também pode ser usado como descodificador de quatro a dezasseis linhas desde que se liguem os terminais de «isolamento» e de dados ao nível lógico 0. Nestas condições, as

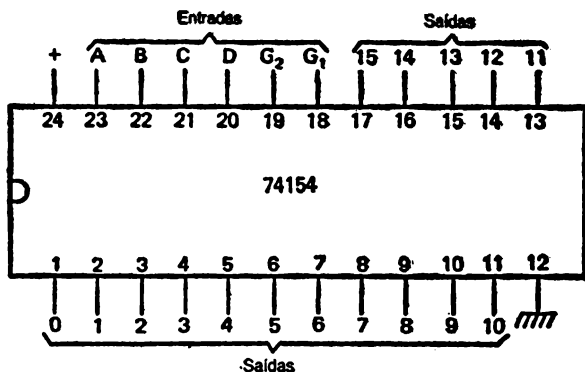


Figura 5.18. — Diagrama da distribuição dos terminais de um decodificador («demultiplexer») de 4 a 16 linhas 74154.

entradas de endereço passam a poder ser usadas como linhas de entrada de dados, de tal modo que quatro linhas de entrada podem produzir um total máximo de 10 saídas. O terminal de saída que nestas condições é endereçado encontrar-se-á ao nível lógico zero, e todos os terminais que não forem endereçados estarão ao nível lógico um. Notemos ainda que ambos estes métodos, a selecção de dados e a descodificação, se tornam extremamente importantes na escolha de memórias e no endereçamento, questão que será tratada com mais pormenor no capítulo 7.

DISPLAYS E PILOTAGENS

Sistemas de contagem

Um circuito digital que é usado para qualquer tipo de contagem necessitará de um «display» que apresente exteriormente o número contado. As formas mais vulgares de display são a binária, octal, decimal e hexadecimal. Os displays binários consistem simplesmente em LEDs ou indicadores semelhantes que mostram o estado de cada flip-flop existente no contador.

Os circuitos que podem ser usados com estes indicadores são mostrados na figura 6.1. Os indicadores LED podem ser pilotados directamente por circuitos de contagem TTL, usando uma resistência de limitação para impedir

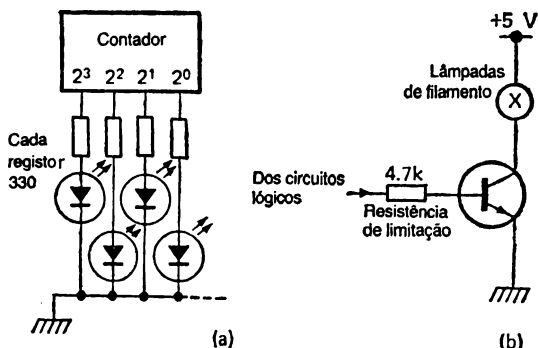


Figura 6.1. — Pilotagem de indicadores binários: (a), pilotagem directa de LEDs, com resistências de limitação; (b), pilotagem de uma lâmpada de filamento, usando uma «interface» transistorizada.

excessos de pilotagem. A tensão num LED condutor é de cerca de 2,0 V (a tensão rigorosa depende do material semiconductor; são usados materiais diferentes para os LEDs, conforme a cor da luz por estes produzida, sendo vulgares correntes de 2 a 10 mA). Para displays maiores utilizando lâmpadas incandescentes, será necessária uma interface transistorizada, dado que os circuitos integrados TTL não conseguem fornecer a corrente necessária para alimentar estas lâmpadas. Pode-se usar uma montagem semelhante quando se utilizem LEDs pilotados por uma lógica CMOS, sendo também possível, em alternativa, construir o andar que pilota os displays recorrendo à série de circuitos integrados Schottky TTL de baixa potência.

Este display binário é muito simples, mas devido ao grande número de dígitos necessários, e à facilidade com que se podem trocar zeros e uns quando se escreve os números, os displays binários simples raramente são práticos.

Os displays octais utilizam uma contagem na base 8, na qual o maior algarismo é o 7; nesta base, 10 equivale ao decimal 8. A conversão de um número decimal para a base oito, no papel, é realizada da mesma maneira já estudada para a base dois, excepto no que se refere ao facto de as divisões sucessivas serem feitas por oito e não por dois (figura 6.2). A conversão de octal para decimal é também realizada no papel da mesma maneira que a conversão binário-decimal, mas utilizando potências de 8 (figura 6.3). O que torna a base octal atraente é o facto de a conversão binário-octal ser extremamente simples. O número binário é agru-

$$\begin{array}{r}
 \text{Restos} \\
 8 \) \ 275 \ (\\
 \quad 34 \ | \ 3 \\
 \quad \quad 4 \ | \ 2 \\
 \quad \quad \quad \dots \ | \ 4 \\
 \quad \quad \quad \quad \uparrow \ \text{Ler de baixo para cima} \\
 \hline
 \text{Octal: } 423
 \end{array}$$

Figura 6.2. — Conversão de decimal para octal. O número é repetidamente dividido por oito, sendo os restos anotados. Estes restos constituem o número octal, da maneira indicada.

Potência de 8	Decimal
0	1
1	8
2	64
3	512
4	4096
5	32768
6	262144

Figura 6.3. — Tabela das potências de oito.

pado em unidades de três dígitos, começando pelo bit menos significativo. Cada número de três bits é então escrito como número decimal; dado que três bits não podem dar um número superior a sete, os números convertidos encontram-se na base oito. Na figura 6.4 apresentam-se exemplos desta conversão em ambos os sentidos. Uma outra característica interessante desta base octal consiste na possibilidade de usar os decodificadores BCD de sete segmentos fáceis de obter no comércio, ligando apenas as três linhas de entrada de menor ordem (figura 6.5).

A conversão octal reduz as possibilidades de erros quando é necessário escrever muitos números binários, como acontece quando se está a programar um microprocessador. A representação hexadecimal, usando a base 16, dá um número ainda menor de dígitos mas apresenta a desvantagem de necessitar de símbolos adicionais para representar os números 10, 11, 12, 13, 14, 15 (decimais). Isto não é no entanto tão grave como parece, porque parte dos sistemas de

1 1 0	0 1 1	0 1 0	Binário
⏟	⏟	⏟	
6	3	2	Octal
4	2	3	Octal
⏟	⏟	⏟	
1 0 0	0 1 0	0 1 1	Binário

Figura 6.4. — Conversão binário-octal e octal-binário — cada dígito octal corresponde a três bits.

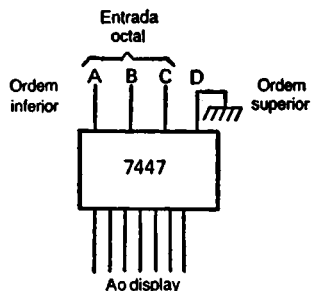


Figura 6.5. — Uso de um decodificador BCD para apresentar números octais.

«display» (ver adiante, neste capítulo) utilizam as letras A, B, C, D, E, F (numa forma estilizada) para representar os dígitos extra necessários. Nesta fase (figura 6.6) 10 representa o número decimal 16, pelo que para evitar confusões este e outros números hexadecimais são escritos sob a forma X'10, X10 ou 10₁₆, nos quais X', X e o número 16 indicam a utilização da base 16. A conversão entre a base hexade-

<u>Decimal</u>	<u>Hexadecimal</u>
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

$$X'1F = 16 + 15 = 31$$

$$X'2C = 2 \times 16 + 12 = 44$$

$$X'FF = 15 \times 16 + 15 = 255$$

1 8 9 7 decimal para hex:

$$\begin{array}{r|l} 16)1897 & \\ 16)118 & 9 \\ 16)7 & 6 \\ & 7 \end{array}$$

$$\underline{\underline{= X'769}}$$

Figura 6.6. — A base hexadecimial, e algumas conversões entre esta base e a decimal.

<i>Lugar do número HEX</i>				<i>Menos significativo</i>			
4		3		2		1	
<i>Hex</i>	<i>Decimal</i>	<i>Hex</i>	<i>Decimal</i>	<i>Hex</i>	<i>Decimal</i>	<i>Hex</i>	<i>Decimal</i>
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

Hex para decimal: veja no quadro o número decimal que corresponde ao número hexadecimal para cada lugar e depois adicione.

Decimal para hex: veja no quadro o número decimal igual ou imediatamente menor ao número dado

Exemplo: 0F2C

Lugar 1 C → 12

Lugar 2 2 → 32

Lugar 3 F → 3840

Lugar 4 0 → 0000

Total 3884

Exemplo: 2215

O mais perto é 2048 = X'800

2215

2048

167

X'A0 = 160

160

7

O n.º hex é 8A7

Verificação: 7 + 160 + 2048 = 2215

Figura 6.7. — Tabela de conversão hexadecimal-decimal para números hexadecimais de quatro dígitos.

decimal e a decimal pode ser realizada (com alguma dificuldade!) pelos mesmos métodos já indicados, mas usando 16 em vez de 8. Como o nosso aprendizado das tabelas de multiplicação (se pertencermos às gerações felizes...) parou na tabela de 10 ou 12 vezes, a base hexadecimal é muito difícil de converter; nestas condições tornam-se bastante úteis tabelas como as da figura 6.7. Na prática a base hexadecimal não é fácil de utilizar nos displays; são poucos os descodificadores existentes para ela, e muitos displays são difíceis de ler, principalmente devido à semelhança dos números 6 e B.

Tal como o código octal, o hexadecimal é particularmente útil devido à facilidade de conversão entre ele e o código binário. O que torna a base 16 ainda mais útil é que um grupo de apenas quatro bits constitui um carácter hexadecimal, de tal modo que o grupo de oito bits vulgarmente usado, ou byte, pode ser representado por um número hexadecimal de apenas dois dígitos. Dado que a tendência dos seres humanos para confundirem os dígitos à medida que o número destes aumenta (como se verificou em certos países quando os telefones passaram a ter indicativos apenas constituídos por números...) a utilização dos números hexadecimais reduz bastante os erros, pelo menos depois de se ganhar alguma experiência.

Na figura 6.8 mostram-se algumas conversões, que exemplificam a maneira de converter com simplicidade grandes números binários; observa-se também a menor confusão produzida pelos números hexadecimais.

Para um uso constante, no entanto é, sempre necessário um display decimal — ninguém estará certamente interessado em usar um voltímetro digital que dê indicações na base dezasseis! Além de alguns sistemas de display decimal directo, a maior parte deles utilizam um sistema BCD 8-4-2-1 intermédio entre as unidades aritméticas binárias e os displays decimais. Os tipos de display que não utilizam BCD são hoje raramente usados, mas serão descritos brevemente.

1 1 0 0	1 1 0 1	0 1 1 1	Binário
⏟	⏟	⏟	
C	D	7	Hex
0	5	F	Hex
⏟	⏟	⏟	⏟
0 0 0 0	0 1 0 1	1 1 1 1	1 1 0 0
			Binário
Binário			Hex
1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 1			DD37
1 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1			DD37

Figura 6.8. — Conversões binário-hexadecimal, com um exemplo da facilidade de distinguir a notação hexadecimal.

Sistemas de display

Os tipos mais antigos de display de contagem utilizavam sistemas que apresentavam um dígito decimal quando a tensão era aplicada a um perne, pelo que era necessário um perne para cada dígito. Os displays incandescentes (de filamento) deste tipo utilizavam pequenos filamentos como os usados nas lâmpadas, montados de forma a produzirem imagens dos números decimais 0 a 9, com uma ligação comum. A selecção dos dígitos era feita aplicando a tensão de trabalho a qualquer das extremidades do filamento escolhido, de tal modo que este emitia luz com a forma do dígito pretendido. Os displays de filamento modernos possuem sete segmentos (ver adiante, neste capítulo), usando 5,0 V a cerca de 8 mA para cada segmento. O display de segmentos moderno possui vantagens consideráveis quando comparado com alguns dos seus competidores, em particular um maior ângulo de visão, um funcionamento a baixa corrente, grande dimensão do dígito, baixo preço e longo tempo de vida.

As lâmpadas cheias a gás (fluorescentes) foram em tempos muito usadas em displays, mas já não o são, pelo menos nas dimensões mais reduzidas. O princípio de funcionamento consiste em que se dois eléctrodos estiverem rodeados

por um gás (como o néon) a baixa pressão, uma voltagem apropriada (45,0 V ou mais) entre eles levará o gás a emitir luz; mas o brilho quase rodeará apenas o eléctrodo (cátodo). Dando portanto ao cátodo a forma de um dígito, o brilho tomará também essa forma. Num display deste tipo, são montados 10 conjuntos de cátodos, com as formas dos dígitos de 0 a 9. Com um ânodo comum ligado a + 50,0 V, bastará passar um dos cátodos para 0 V para que seja iluminado um dos dígitos. Um display deste tipo necessita de um conjunto de 10 entradas (11 no caso de ser incluída uma vírgula decimal), e cada dígito é apresentado levando o cátodo correspondente a uma tensão de 0 V. Para manter um dígito apagado, no entanto, a tensão do cátodo correspondente deve ser relativamente elevada, normalmente de cerca de + 30,0 V. Como esta tensão de trabalho é demasiado elevada para o circuito normal de tipo TTL ou CMOS, é necessário algum tipo de circuito de interface. Este pode assumir a forma de um driver TTL especial, utilizando andares de saída calibrados para 30,0 V, ou uma interface transistorizada para cada cátodo.

Display de sete segmentos e de matriz de pontos

Os dispositivos modernos de «display», qualquer que seja o método por eles usado para apresentar os números, encontram-se organizados na forma «sete segmentos» ou «matriz de pontos». O tipo de sete segmentos, como o nome indica, utiliza um conjunto de sete barras montadas de modo a formarem um algarismo ligeiramente oblíquo (figura 6.9). Por convenção, as letras apresentadas na figura 6.9 correspondem aos segmentos mais próximos delas. Iluminando os segmentos apropriados pode-se apresentar qualquer algarismo, assim como letras em forma estilizada, de A até F. As formas dos dígitos são razoavelmente fáceis de reconhecer, mas é necessário utilizar letras pequenas para evitar (por exemplo) confusões entre «B» e «8» e «D» e «0». Note-se a diferença (figura 6.9) entre «b» e «6»; o dígito 6 utiliza uma barra horizontal na parte superior. A maior parte dos displays de sete segmentos incluirão ainda um oitavo segmento, a vírgula

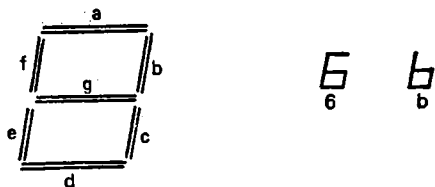


Figura 6.9. — A forma de um display de sete segmentos, com dois exemplos. Os segmentos são designados por letras.

decimal. Esta pode ser colocada antes ou depois dos segmentos (figura 6.10), sendo necessário algum cuidado para escolher o tipo mais correcto para cada apresentação.

Os sinais de um contador BCD não podem ser aplicados directamente a um display de sete segmentos, pelo que neste caso é necessário usar como interface um descodificador BCD para sete segmentos. Este descodificador consiste simplesmente em portas cujas entradas são os dígitos BCD e cujas saídas são as linhas de pilotagem do display de sete segmentos. O descodificador pode incluir ainda saídas de alta tensão ou, no caso de certos tipos de display, saídas c.a. Na figura 6.11 apresenta-se uma tabela de verdade de um descodificador BCD-sete segmentos. Os descodificadores que são concebidos apenas para displays decimais são ligados internamente de tal modo que as entradas correspondentes às quantidades 10 a 15 não dêem qualquer saída ou dêem saídas sem significado. Os caracteres hexadecimais não podem então ser obtidos.



Figura 6.10. — Posições alternativas da vírgula decimal (sendo de origem inglesa ou americana, os displays apresentam normalmente pontos em vez de vírgulas).

A maior parte dos descodificadores BCD-sete segmentos possuem ainda outras capacidades como sejam as de verificação das lâmpadas, «strobing», limpeza dos zeros à esquerda,

<i>Contagem</i>	<i>Segmentos iluminados</i>
0	a b c d e f
1	b c
2	a b g e d
3	a b g c d
4	f g b c
5	a f g c d
6	a f e d c g
7	a b c
8	a b c d e f g
9	a b g f c

Figura 6.11. — Tabela de verdade das saídas do decodificador BCD — sete segmentos. Conforme o tipo de display que se pretende obter, o segmento pode ser iluminado quando recebe uma tensão lógica 0 ou 1.

etc. O perne de verificação das lâmpadas permite acender todos os segmentos, verificando assim se algum deles está deteriorado. «Strobing» permite o isolamento das saídas do decodificador, desligando portanto o display. Esta técnica é bastante útil quando o display é do tipo «multiplex», como se descreve mais adiante, neste mesmo capítulo. A supressão dos zeros à esquerda é usada quando se utiliza um conjunto de vários displays. Quando todas as entradas e saídas de supressão de zeros são ligadas em linha (figura 6.12), não será iluminado qualquer 0 à esquerda do dígito mais significativo do display. Deste modo, os zeros em causa são suprimidos, por exemplo no número 0034, mas já não o são os que se encontram à direita, por exemplo no número 1200.

Existem decodificadores para uso com outros códigos de contagem, como sejam o código Gray e o código excesso-3, mas a gama destes decodificadores é bastante limitada, quando comparada à gama de decodificadores binários e BCD.

Como já se disse anteriormente, neste mesmo capítulo, podem-se obter displays de filamentos incandescentes na forma sete segmentos, os quais apresentam várias vantagens relativamente a quaisquer outros. Podem ser por exemplo pilotados a partir de decodificadores como o 7448. Os display de sete

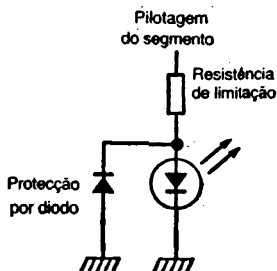


Figura 6.13. — Protecção de um segmento LED contra tensões inversas. Esta protecção torna-se necessária apenas quando são usados circuitos de interface com tensões de alimentação negativas.

decimal) ligados separadamente ao decodificador através de uma resistência limitadora. O decodificador deve ser de um tipo apropriado a estes displays de ânodo comum, com uma saída zero para um segmento iluminado, e uma saída 1 para um segmento «apagado». O display de cátodo comum trabalhará com o terminal comum à massa e os terminais dos segmentos ligados através de resistências de limitação aos terminais do decodificador. O decodificador deverá ser de um tipo apropriado a estes displays, dando um nível lógico 1 na saída aos segmentos iluminados, e um nível lógico 0 na correspondente aos segmentos não iluminados.

Se só se dispõe de um tipo «errado» de decodificador, podem-se usar inversores lógicos, montados entre o decodificador e as resistências de limitação.

O display do tipo «matriz de pontos» (figura 6.14) permite apresentar uma gama completa de números, letras e símbolos, mas à custa de um sistema de decodificação muito mais complexo que utiliza uma memória «read-only». Como o nome sugere, a matriz de pontos consiste num certo número de LEDs com as dimensões de pontos dispostos em filas e colunas, sendo normal encontrar a configuração 5×7 . A matriz 5×7 obriga a controlar 35 LEDs diferentes, existindo diversos tipos de displays deste género com uma decodificação incorporada, reduzindo-se assim o número de terminais de ligação que de outro modo seriam necessários. Podem-

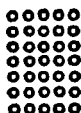


Figura 6.14. — Matriz de pontos 5×7 . Este display pode apresentar algarismos, letras e símbolos.

-se conceber sistemas de descodificação mais simples se se pretende iluminar os pontos em sequência e não simultaneamente.

Os displays LED de qualquer tipo possuem uma grave desvantagem, a de necessitarem de bastante corrente, o que os torna inconvenientes para uso em displays alimentados a pilhas quando devem ser usados durante períodos longos. As máquinas de calcular que utilizam displays LED recorrem de um modo geral a circuitos que cortam automaticamente a alimentação do display (sem perda da energia necessária para os circuitos de cálculo depois de cerca de um minuto); uma alternativa consiste no uso de pilhas recarregáveis, caras e por vezes com um tempo de vida reduzido.

Os displays fluorescentes de baixa tensão constituem um método de reduzir a corrente necessária a um display de vários dígitos. Estes displays, normalmente de cor verde, são indicadores de raios catódicos em miniatura, nos quais os electrões produzidos por um filamento quente são acelerados em direcção a um ânodo fluorescente cuja forma é a de uma barra do tipo dos displays de sete segmentos. É necessária uma tensão de cerca de 18,0 V entre o ânodo e o cátodo para facilidade de leitura, a qual pode ser obtida da maneira indicada na figura 6.15. Dado que não se verifica um brilho apreciável para uma diferença de tensão de 4,5 V, a tensão do cátodo do display pode ser ligada a 4,5 V, usando os circuitos lógicos para fornecer as tensões de 0 V (desligado) e + 13,5 V (ligado). Isto permite que o display fluorescente seja pilotado directamente por um contador descodificador CMOS, como o CD4026A ou o CD4033A. Se a lógica usada com o display fluorescente for TTL, é necessário usar descodificadores que possuam saídas por transístores em colectores

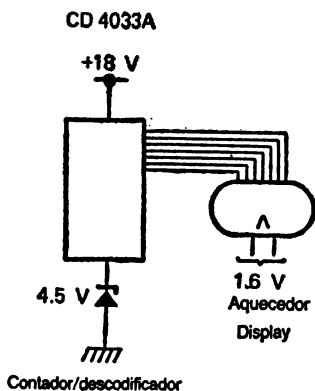


Figura 6.15. — Pilotagem de um display fluorescente a partir de um descodificador CMOS.

aberto (figura 6.16). O transistor de saída do circuito integrado é fabricado de tal modo que suporta até +30,0 V, pelo que é possível pilotar directamente displays de tensão elevada desde que se disponha de uma fonte de alimentação apropriada. O display é a carga das ligações de colector dos transistores de saída.

Um tipo de display que actualmente é muito usado é o chamado «de cristal líquido» (LOD, «liquid-crystal display»). Um cristal líquido é um material cujo comportamento se encontra de certos pontos de vista a meio caminho entre o de um líquido e o de um cristal sólido. A característica mais importante destes materiais do ponto de vista das técnicas de display consiste em que o material é sensível a um campo eléctrico que lhe é aplicado entre duas placas condutoras, como o dieléctrico de um condensador. A montagem normal de um LOD consiste numa placa traseira metálica, normalmente escurecida, e um material transparente, com um revestimento igualmente transparente, funcionando como placa dianteira. A placa frontal pode ser revestida com um condutor com o arranjo habitual em sete segmentos, colocando-se então as duas placas numa célula com o cristal líquido enchendo o

espaço entre elas. Quando não recebe energia o cristal líquido é opaco, pelo que a placa frontal parece reflectora. Quando é aplicada uma tensão, de preferência uma onda quadrada a uma frequência entre 30 e 200 Hz, entre a placa traseira e uma dada barra da placa frontal, o cristal líquido torna-se transparente entre estes pontos. A superfície traseira escurecida surge agora com a forma de um segmento, ao contrário do reflexo que se observa no resto do display. Dado que não é produzida qualquer luz pelo display, a visibilidade só é boa quando existe iluminação exterior. Uma variante deste esquema básico, quando não é conveniente para funcionamento a pilhas, consiste em tornar também transparente a placa traseira, e em utilizar uma luz traseira contínua. Quando é aplicada uma onda quadrada ao cristal líquido, a luz da parte traseira do display pode ser vista da frente.

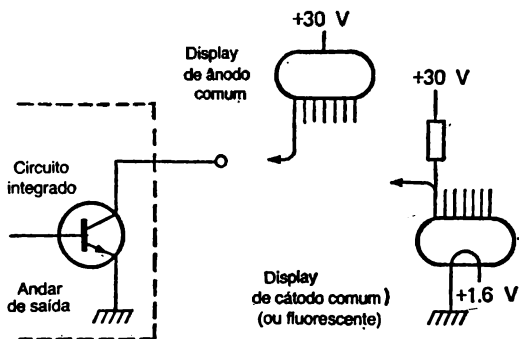


Figura 8.16. — Pilotagem TTL de displays de alta tensão, usando saídas em colector aberto.

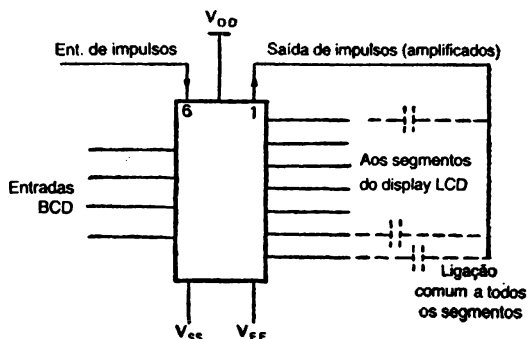
Como a onda quadrada é aplicada àquilo que na prática constitui um condensador de reduzido valor, são necessárias correntes muito fracas, da ordem dos $30\mu\text{A}$, para todos os segmentos do display. Não deve ser aplicada corrente contínua a um LCD, pelo que o tipo de circuito piloto aqui necessário difere consideravelmente do utilizado noutros displays. O esquema habitual consiste em utilizar uma acção descodificadora BCD-sete segmentos, juntamente com um

andar piloto alimentado por uma entrada quadrada. A onda quadrada é amplificada, normalmente para 15,0 V pico-a-pico, e alimentada à placa traseira do display. Quando não está a ser usado um dado segmento, o sinal dele é uma onda quadrada idêntica em fase à da placa traseira, pelo que a diferença de potencial entre ambas é zero. Quando é escolhido um segmento pelos circuitos do descodificador, é aplicada àquele uma onda quadrada inversa à da placa traseira pelo que a onda resultante entre as placas (figura 6.17) é uma onda quadrada cuja amplitude é igual à da tensão de alimentação em cada sentido. Deste modo, uma alimentação de 15,0 V pode fornecer o equivalente a 30,0 V pico-a-pico, se bem que a onda quadrada piloto alimenta ao circuito integrado seja apenas 5,0 V pico-a-pico. O CD4055A é um circuito de pilotagem típico para LCDs.

Displays «multiplex»

Quando um display consiste apenas em alguns algarismos decimais, o método de montagem mais simples consiste em utilizar um contador e descodificador/piloto BCD para cada algarismo do display. Os displays de quatro ou mais algarismos, no entanto, são normalmente accionados numa forma «multiplex», a tal ponto que as ligações dos terminais só permitem geralmente este tipo de funcionamento quando numa mesma unidade existe um conjunto de dígitos

O display «multiplex», qualquer que seja o número de dígitos utilizado, possui apenas um conjunto de terminais dos segmentos, correspondentes às sete barras e à vírgula decimal, ou seja, um total de oito. Deve existir também um conjunto de terminais de ânodo comum (um cátodo comum, conforme a montagem usada), um para cada dígito. Uma saída em cada linha de segmentos provocará apenas a apresentação de um dígito, porque só é escolhido simultaneamente um dado ânodo (ou cátodo). O circuito deve ser montado de tal modo que quando é escolhido o ânodo (ou cátodo) de um determinado dígito os sinais de pilotagem dos segmentos sejam correctos para esse dígito.



Exemplo: $V_{DD} = +5 \text{ V}$, $V_{SS} = 0$, $V_{EE} = -10 \text{ V}$
 Entradas BCD 0 - 5 V

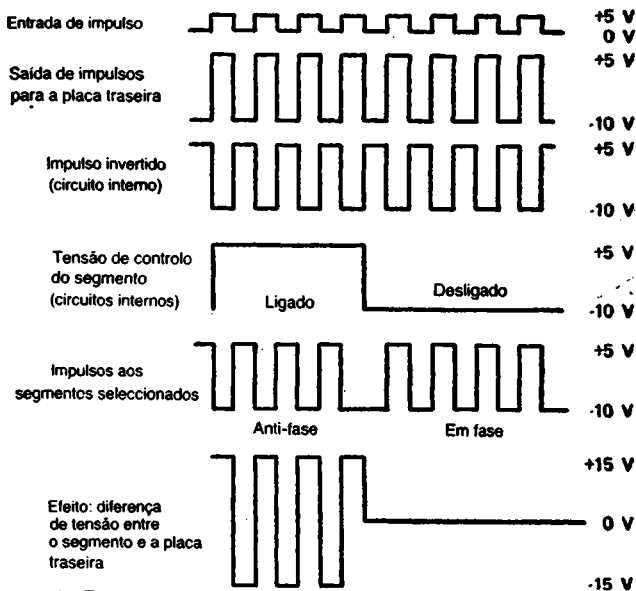


Figura 6.17. — Pilotagem de um display de cristal líquido (LCD). O circuito integrado piloto realiza a amplificação e inversão dos impulsos e assegura que não seja fornecida ao display qualquer corrente contínua.

Uma maneira de utilizar esta montagem não permite poupar muito nas pastilhas de circuitos integrados (a menos que se recorra a um circuito LCI) mas poupa muito em corrente, dado que só se encontra ligado em qualquer momento um único elemento. Este circuito é esboçado na figura 6.18. Cada dígito necessita de um contador/decodificador que pilota os displays através de um circuito de interface. Cada contador/decodificador integrado possui uma entrada de «isolamento» que isola a saída até surgir neste terminal um impulso positivo. Para apresentar o dígito menos significativo, portanto, o impulso de isolamento é aplicado a esta unidade de con-

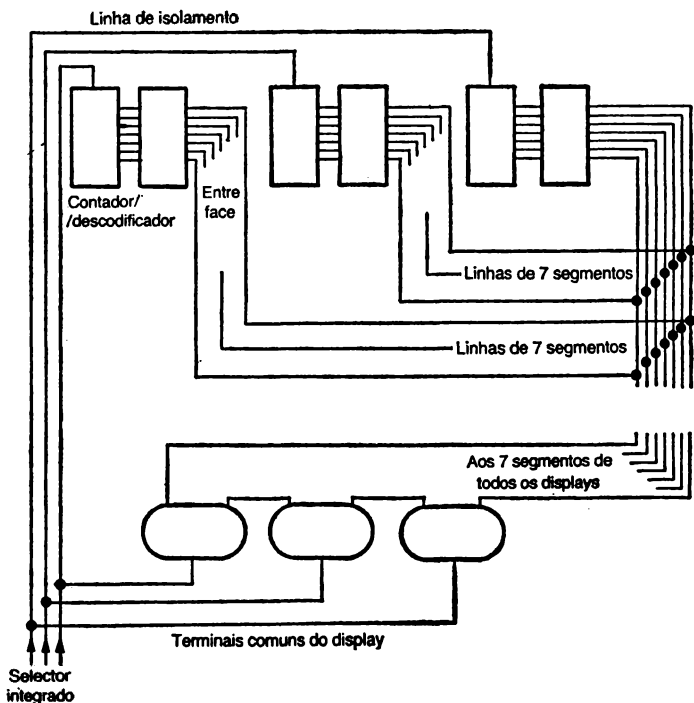


Figura 6.18 — Dispositivo multiplexado, no qual os terminais estão ligados em sequência, utilizando um conjunto comum de sete-linhas segmento. Essencial se o dispositivo for construído em bloco.

tagem, depois do que o contador/descodificador isolado fornecerá sinais às linhas de saída, mostrando o dígito. A comutação dos impulsos de isolamento pode ser realizada por um contador sequencial, como o contador Johnson descodificado, ou um dos muitos circuitos integrados que existem para este fim. Se o descodificador não possui esta entrada de isolamento (aqui designada «strobe»), a mistura pode ser realizada aplicando impulsos de isolamento a uma porta capaz de pilotar o terminal comum do descodificador.

Um esquema «multiplex» mais ambicioso, conveniente para os displays «multi-dígitos», é ilustrado na figura 6.19 e utiliza um circuito integrado como o 74153. Por exemplo, um conjunto de quatro «multiplexers» de um bit pode comutar um sinal de quatro bits (como num conjunto de dígitos BCD) em quatro linhas de saída, a partir de vários conjuntos de linhas de entrada. Usando este esquema só é necessário um descodificador, alimentado pelas quatro linhas de saída do multiplex; este aspecto é muito importante quando se utiliza um descodificador hexadecimal, bastante caro.

A sequência de apresentação é a seguinte. No caso do dígito decimal menos significativo, o contador piloto envia impulsos ao display e escolhe ainda o conjunto adequado de entradas de linhas ao multiplex, assegurando assim que o número BCD enviado ao descodificador é o correspondente àquilo que se deseja apresentar no display. É então isolado o display seguinte, enviando-se o conjunto seguinte de sinais BCD pelas linhas do descodificador; este processo continua a ser repetido até ser apresentada toda a informação.

Quando se usam displays «multiplex», é necessário ter algum cuidado quanto à frequência em que os vários elementos são escolhidos. Num LCD, por exemplo, existirá uma frequência de ondas quadradas para o display propriamente dito, uma frequência de relógio para o circuito lógico, e ainda uma frequência de selecção enviada ao multiplex. Estas frequências devem estar relacionadas e sincronizadas entre si, o que se pode assegurar pilotando-as a partir de uma mesma fonte de frequência, usando andares divisores para obter as diferentes frequências necessárias. Este método evita os efeitos de «batimento» que de outro modo poderiam ser notórios.

Displays video

Os métodos de apresentação de dados descritos até agora são óptimos para máquinas de calcular e para pequenos dis-

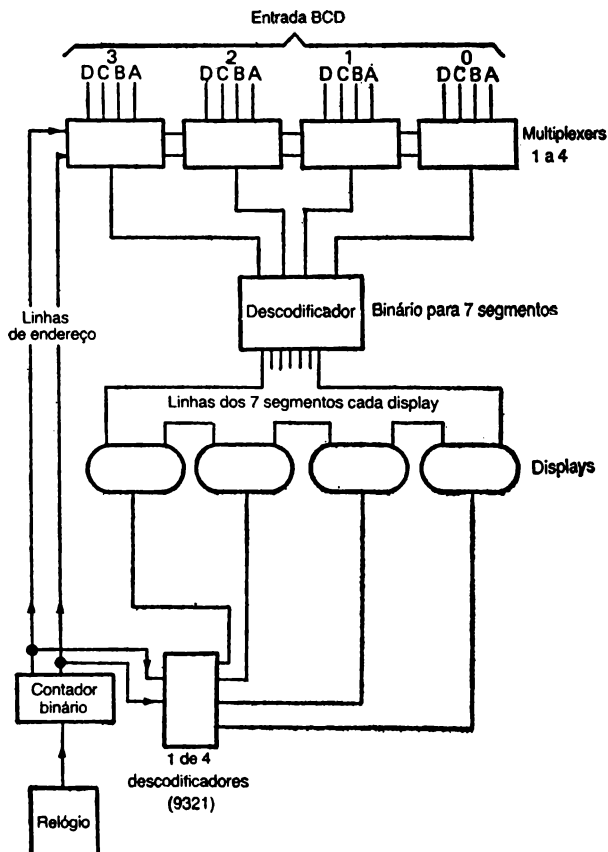


Figura 6.19.—Um outro tipo de display «multiplex». O contador binário activa linhas de endereço que seleccionam o dígito que cada multiplex envia ao descodificador único. Ao mesmo tempo, um de quatro descodificadores selecciona o display que vai ser usado.

plays hexadecimais, mas não para a apresentação de palavras dispostas em frases nem para grandes blocos de algarismos. Para apresentar a saída de computadores é preferível um écran video a qualquer outro tipo de display temporário. Os «displays» permanentes, como por exemplo as tele-impresoras, não serão consideradas aqui.

Um display video utiliza um tubo de raios catódicos semelhante ao dos televisores. Se bem que não possam ser usados os métodos normais de varrimento, a existência no comércio de componentes para este tipo de equipamento a baixo preço leva a maior parte dos displays video a utilizarem o varrimento de linha e campo que é empregue nos receptores de televisão. Quando não é importante obter uma elevada resolução, por exemplo quando são apresentados menos de 40 caracteres por linha, pode-se usar um receptor TV convencional como display video, desde que os sinais do computador (com uma forma apropriada) possam ser modulados numa portadora na gama de frequências UHF. A degradação da resolução produzida pela maior parte dos moduladores e filtros passa-banda de frequência intermédia existentes nos receptores tornam porém este método inaceitável no caso de unidades de alta resolução (60 caracteres por linha ou mais), devendo então ser usado um monitor que aceite sinais video compostos. Um receptor de televisão ligado à rede nunca deve ser usado como monitor, se bem que seja possível adaptar alguns receptores alimentados a pilhas desde que sejam accionados apenas pelas alimentações de baixa tensão do computador.

O problema seguinte diz respeito à obtenção das ondas video necessárias para modular ou alimentar o monitor. Ao contrário de um display de sete segmentos ou de matriz de pontos, o écran video oferece uma escolha bi-dimensional da posição de cada letra, símbolo ou número. Dada a vasta quantidade de símbolos que podem ser apresentados, os códigos binários simples são insuficientes, sendo geralmente usado um código múltiplo designado ASCII (American Standard Code for Information Interchange). O uso do código ASCII não é universal, mas constitui de longe o sistema de codificação mais usado nos pequenos computadores modernos. O código ASCII, ilustrado na figura 6.20, permite usar uma

Carácter =		Carácter =		Carácter =		Carácter =	
NULL	00						
SOH	01	!	21	A	41	a	61
STX	02	"	22	B	42	b	62
ETX	03	☞	23	C	43	c	63
EOT	04	\$	24	D	44	d	64
ENQ	05	%	25	E	45	e	65
ACK	06	&	26	F	46	f	66
BEL	07	'	27	G	47	g	67
BS	08	(28	H	48	h	68
HT	09)	29	I	49	i	69
LF	0A	*	2A	J	4A	j	6A
VT	0B	+	2B	K	4B	k	6B
FF	0C	,	2C	L	4C	l	6C
CR	0D	-	2D	M	4D	m	6D
SO	0E	.	2E	N	4E	n	6E
SI	0F	/	2F	O	4F	o	6F
DLE	10	0	30	P	50	p	70
DC1	11	1	31	Q	51	q	71
DC2	12	2	32	R	52	r	72
DC3	13	3	33	S	53	s	73
DC4	14	4	34	T	54	t	74
NAK	15	5	35	U	55	u	75
SYN	16	6	36	V	56	v	76
ETB	17	7	37	W	57	w	77
CAN	18	8	38	X	58	x	78
EM	19	9	39	Y	59	y	79
SUB	1A	:	3A	Z	5A	z	7A
ESC	1B	;	3B	[5B	{	7B
FS	1C	<	3C	\	5C		7C
GS	1D	"	3D]	5D	}	7D
RS	1E	>	3E	↑	5E	~	7E
US	1F	?	3F	←	5F	Delete	7F
SP	20	@	40	\.	60		

vasta quantidade de instruções e símbolos. Estas instruções são concebidas para resolver os problemas de localização correcta dos caracteres no écran, e assumem a forma de retorno do carro, alimentação de linha, e outras que seriam igualmente usadas numa tele-impressora. O código ASCII permite ainda o uso de um bit de paridade, um bit extra que é acrescentado ao código de sete bits para tornar par o número total de uns. Utilizando este bit adicional, o computador pode detectar ou assinalar um erro num carácter codificado por simples contagem do número de uns existente em cada carácter codificado.

A conversão do código ASCII em sinais viedo requer uma considerável quantidade de circuitos digitais, o que pode obrigar ao uso de um grande número de pastilhas semicondutoras ou de um micro-processador (capítulo seguinte). Os sistemas lógicos deste tipo permitem algum controlo sobre o número de caracteres que são apresentados por linha, enquanto que os sistemas em apenas uma pastilha semicondutora determinam normalmente esta quantidade num valor que é apropriado ao uso em monitores video de qualidade profissional. A expressão «pastilha única» é um pouco enganadora

Instruções

SOH — start heading/message	SI — shift in
STX — start text	DLE — data link escape
ETX — end text	DC1-4 — device control signals
EOT — end transmission	NAK — negative acknowledge
ENQ — enquiry	SYN — synchronous idle
ACK — acknowledge	ETB — end transmission block
BEL — ring bell	CAN — cancel
BS — backspace	EM — end medium
HT — horizontal tabulation	SUB — substitute
LF — line feed (nova linha)	ESC — escape, prefix
VT — vertical tabulation	FS — file separator
FF — form feed (pág. seg.)	GS — group separator
CR — carriage return	RS — record separator
SO — shift out	US — unit separator
	SP — space

Figura 6.20 — O código ASCII. Para poupar espaço apresenta-se o equivalente hexadecimal do código binário de sete bits. Nem todos os códigos de instruções poderão ser necessários num dado teletipo.

dados que são de facto necessárias várias pastilhas para formar um descodificador lógico completo, e que aumenta o custo do descodificador para um preço superior ao da versão anterior. Como seria de esperar, o procedimento de descodificação é bastante complexo e encontra-se para além do objectivo deste livro.

SISTEMAS MICROPROCESSADORES

Qualquer tabela de verdade que possa ser realizada pela combinação habitual de portas lógicas e registos de deslocamento pode também sê-lo por um programa de computador. Um programa deste tipo começa por examinar as entradas *em sequência* e produzindo uma saída que depende das entradas. Enquanto o uso de portas, registos e outros circuitos digitais é considerado uma solução *hardware*, o uso de um programa é considerado uma solução *software* do problema de realização prática da tabela de verdade.

O primeiro microprocessador verdadeiro, o INTEL 4004, surgiu em 1969, e constituiu uma tentativa de produzir um sistema lógico de um computador, com a possibilidade de ser programado, numa única pastilha de circuito integrado. Esta deve conter um número suficiente de portas para produzir as funções de porta básicas, como sejam a AND, a OR, a X-OR, e as funções aritméticas, como a soma e a complementação-e-soma além de registos de deslocamento que retenham números binários em bytes de oito bits, e contadores que controlem a sequência de acontecimentos ao longo do programa. Dado que são estas as funções realizadas pela *unidade de processamento central* de um computador grande, o microprocessador é por vezes designado pelas letras CPU (iniciais daquela expressão em inglês).

No momento em que escrevemos, a maior parte dos fabricantes de semicondutores fabricam pastilhas CPU de concepção própria ou copiando outras. Se bem que estas CPUs variem consideravelmente quanto à maneira como realizam

as instruções, pode-se resumir um certo número de características comuns à actual geração de microprocessadores.

Para começar, todas estas CPUs utilizam bytes de oito bits, o que significa que é possível processar em paralelo oito bits de sinais digitais. No momento em que escrevemos, começam a aparecer no mercado microprocessadores de 16 bits mas a maior parte das aplicações presentes podem ser realizadas confortavelmente com oito bits — a diferença reside apenas no tempo de processamento.

Qualquer pastilha CPU deve portanto permitir a entrada e saída de um byte de oito bits a partir de registos de deslocamento com o mesmo número de bits. O registo principal usado para a recepção e transmissão destes bytes de oito bits é designado *acumulador*, e é ainda nele que são mantidos todos os resultados das operações de processamento. Alguns modelos só utilizam este único registo para carregar bytes, mas outras CPUs podem ser programadas para utilizar vários registos.

A carga de um byte num registo de deslocamento requer oito pernes de entrada (para carga em paralelo), e seria de esperar que a sua descarga obrigaria igualmente ao uso de oito pernes, num total de 16. Dada a necessidade de manter reduzido o número de ligações, é conveniente usar apenas oito terminais tanto para a saída como para a entrada. Um conjunto de portas internas controladas por um sinal interno liga os oito pernes da pastilha à entrada ou à saída do registo, conforme as necessidades. O sinal que controla este uso duplo dos terminais de dados deve ser concebido de tal modo que a escolha das portas de entrada seja automática a menos que as instruções do programa seleccionem uma saída. Alguns modelos utilizam uma entrada especial através da qual é alimentado um sinal que isola a CPU das tensões correspondentes aos dados. Todos eles, no entanto, permitem a saída de um sinal que pode ser usado para controlar outras pastilhas indicando se os dados estão a entrar ou a sair da CPU. Em todos os casos a entrada de dados na CPU é designada por «leitura», e a saída como «escrita».

Antes de podermos continuar a descrição da CPU, convirá observarmos mais atentamente as pastilhas semicondutoras que o rodeiam. Sem estas, a CPU não pode funcionar, pelo que não é possível descrever o funcionamento desta sem descrever igualmente as funções dos circuitos integrados de que depende.

Entre estes outros circuitos integrados, as memórias são sem dúvida os mais importantes. Uma pastilha de memória armazena bits digitais até serem necessários à CPU, e o acto de fornecimento de bits (leitura da memória) não destrói ou modifica os dígitos armazenados. A memória pode ser «volátil» ou «não-volátil». A primeira perde as informações nela guardadas quando a alimentação é cortada. Um registo de deslocamento é uma forma simples de memória volátil, dado que os flip-flops não se manterão comutados na ausência de uma alimentação de energia. Por outro lado, quando a alimentação é novamente ligada, os flip-flops comutarão ao acaso, pelo que são armazenados dígitos sem qualquer significado. As memórias não-voláteis, pelo contrário, podem ser usadas para armazenamento permanente porque os bytes nelas guardados não são apagados nem modificados quando a alimentação é cortada.

Tal como num computador de grandes dimensões, a memória não-volátil de um sistema microprocessador pode ser uma fita, uma cassete ou um disco magnético, mas o microprocessamento, ao contrário daqueles, utiliza também pastilhas de memória não-volátil designadas por ROM, um nome formado pelas iniciais de *Read-Only Memory* (ou «memória apenas de leitura»), porque estes circuitos são concebidos de tal modo que os bytes neles armazenados não podem ser modificados por qualquer acção do microprocessador. Ao contrário da fita ou do disco magnéticos, os dados armazenados numa ROM podem ser obtidos pela CPU ou lidos desta muito rapidamente, normalmente em menos de um microsegundo. A utilidade principal das ROMs num sistema microprocessador consiste em fornecer um programa que permite usar o sistema e programá-lo de acordo com as necessidades. Sem este programa «monitor», torna-se necessária uma quan-

tidade muito maior de circuitos lógicos para permitir ao microprocessador realizar a entrada e a saída de dados.

Uma ROM, como um programa monitor ou um gerador de caracteres (para armazenamento das instruções necessárias para imprimir caracteres num visor de televisão), que é muito usada e constitui de facto uma parte essencial do sistema microprocessador, será programada durante o fabrico. Isto significa que a ROM é produzida a partir do programa, que assume então a forma de ligações feitas no interior do circuito integrado. Este método é proibitivamente caro em pequenas quantidades dado o custo das «grelhas» ou «máscaras» usadas no fabrico. Quando estes custos fixos são divididos por um grande número de pastilhas de circuito integrado, no entanto, estas ROMs podem-se tornar mais baratas do que os outros tipos.

Qualquer que seja a aplicação de um microprocessador, será necessário conceber um programa de utilização e verificá-lo; isto pode ser feito utilizando de início, uma memória volátil. No entanto, se se pretende utilizar o microprocessador num sistema de controlo é necessário que o programa seja produzido como uma ROM. Existem várias maneiras de o fazer. O programa pode ser escrito para um computador grande e em seguida verificado no sistema microprocessador usando uma memória volátil. O programa do computador principal pode então ser impresso e enviado a um fabricante de semicondutores que produzirá máscaras ROM directamente a partir dele. Um outro método, que é necessário no caso de não se dispor do computador grande, consiste em usar um tipo de ROM conhecida pelo nome de PROM, na qual é possível escrever um programa. A letra «P» de PROM significa programável, mas a programação não pode ser realizada pela acção normal dos circuitos do microprocessador. A vantagem do uso de uma PROM consiste em esta permitir a concepção dos protótipos de um circuito de controlo do microprocessador e a sua verificação antes de serem usadas ROMs já programadas. Por outro lado, para pequenas quantidades o uso de PROMs é normalmente mais económico do que o das ROMs.

As PROMs de ligação fusível contêm um conjunto de ligações que, tal como os fusíveis, podem rebentar devido à passagem de uma corrente de cerca de 200 mA a 17,0 V. Esta corrente é fornecida por impulsos, e as tensões existentes nas outras partes dos circuitos PROM são concebidas de tal modo que não é possível verificar-se uma deterioração em qualquer outro ponto. Quando saem da linha de fabrico, as PROMs apresentam ligações não fundidas, pelo que cada bit de memória se encontra ao nível lógico 1. O acto de programação consiste portanto em fundir cada ligação onde seja necessário um nível lógico zero. Depois de fundidas, como é óbvio, as ligações não podem ser restauradas. A corrente necessária para fundir as ligações é bastante maior do que a que o microprocessador pode fornecer, se por acaso a PROM fosse ligada a um terminal ao nível lógico zero, pelo que se tornam necessários circuitos de programação separados. Estes circuitos são concebidos de tal modo que o programa pode ser lido de uma memória volátil ou de uma fita magnética entrando o seu conteúdo na PROM, fazendo-se passar uma corrente suficiente para fundir a ligação correspondente a cada zero.

Um último passo no desenvolvimento das PROMs consistiu na EPROM, ou PROM «apagável». O sistema hoje mais popular utiliza luz ultra-violeta para apagar a PROM, que possui uma «janela» de quartzo cobrindo toda a pastilha. Esta acção tem como consequência o armazenamento de um nível lógico 1 em cada parte da memória, e é realizada deixando a janela da pastilha exposta a uma lâmpada de ultravioletas durante cerca de 30 minutos a uma distância de alguns centímetros. A operação de programação coloca certos bits ao nível 0, o que é feito mantendo os bits escolhidos, em sequência, ao nível lógico 0 durante um milissegundo e passando 125 a 150 vezes por todos os bits de modo a fixar o programa. Durante a programação são usadas tensões de alimentação muito maiores, de tal modo que o funcionamento normal não tem qualquer efeito na PROM daí em diante.

Os circuitos integrados de memória volátil usados com os microprocessadores são conhecidos pelo nome RAM, ou

«memória de acesso aleatório», uma designação antigamente usada para distinguir uma memória deste tipo daquela em que os bits são alimentados sequencialmente. Qualquer destes tipos de memória pode ser fabricado utilizando flip-flops montados como registos, mas como todos os circuitos integrados hoje usados com microprocessadores permitem um acesso aleatório (o que significa que é possível escolher um bit em qualquer parte da memória), quer sejam RAM ou ROM, seria mais apropriado designar aqueles por *memória de leitura-escrita*.

Ao contrário da ROM, a RAM pode ser escrita a partir do microprocessador utilizando-se para tal uma tensão de controlo de «escrita». Um nível de sinal lógico no terminal «escrita» do circuito integrado RAM permitirá ao microprocessador armazenar bytes de informação na memória, e o sinal oposto servirá para comutar a memória para «leitura», sendo os bytes lidos da memória para o microprocessador. O processo de escrita substitui automaticamente o que estava escrito na memória, mas o processo de leitura não destrói a informação nela contida.

Existem RAMs estáticas e dinâmicas. O tipo dinâmico baseia-se na carga de condensadores que, devido à perda de carga produzida por fugas, devem ser recarregadas periodicamente. Isto é feito através de um impulso que é enviado ciclicamente a todos os condensadores que armazenam um nível lógico 1. Qualquer sistema que utilize memórias dinâmicas (que são relativamente baratas e consomem muito pouca corrente) deve portanto fornecer um impulso deste tipo para manter os dados armazenados. Obviamente, este impulso deve ocorrer num momento em que as memórias não estejam a ser escritas ou lidas, pelo que deve ser controlado pelo microprocessador. Na CPU Z80 este esquema é conduzido à sua conclusão lógica, sendo o próprio microprocessador que produz os impulsos necessários às memórias dinâmicas.

As memórias estáticas, usando flip-flops ou circuitos semelhantes para armazenamento, não necessitam de impulsos de recarga, podendo portanto ser usadas em sistemas microprocessadores em que aquele impulso seria indesejável. No

momento em que escrevemos, o custo (e o consumo de energia) de uma memória estática RAM é consideravelmente superior ao da sua equivalente dinâmica.

Qualquer tipo que seja o tipo de memória usado nos circuitos de apoio do microprocessador, é necessário existir algum meio de escolha dos dados a ler da memória, ou de escolha da parte da memória na qual os dados podem ser escritos. A selecção de um bit de informação ou de um dado conjunto de bits é realizada recorrendo a um conjunto de linhas de endereço nas quais se utilizam dígitos para especificar um número de código ou endereço para cada byte da memória. Um decodificador existente na pastilha de memória é activado pelos sinais existentes nas linhas de endereço de modo a seleccionar uma parte da memória, que será ligada aos terminais de saída.

A natureza desta decodificação depende em grande medida da maneira como a pastilha de memória se encontra organizada. Por exemplo, uma pastilha descrita como uma memória 1024×1 será capaz de armazenar um bit único em cada um de 1024 endereços diferentes, de tal modo que cada bit possa ser obtido num único terminal de entrada/saída. Para endereçar um total de 1024 (2^{10}) posições serão usadas dez linhas de endereço. No caso de um microprocessador que utilize um byte de oito bits, serão usadas oito destas pastilhas de memória (figura 7.1), tendo cada uma delas o seu terminal de leitura ligado a um dos terminais de dados da CPU. Os dez terminais de endereço das memórias serão ligados em paralelo, sendo pilotados pelas linhas de endereço do microprocessador. Só 10 das habituais 16 linhas de endereço do microprocessador seriam necessárias neste exemplo.

Outras organizações de memória, como as 512×4 ou 256×8 , utilizam menos linhas de endereço e mais linhas de dados, dependendo o método particular de organização dos circuitos existentes, dos preços relativos, e da necessidade de uma dada quantidade de memórias. Pode ser necessário recorrer a alguns circuitos de porta a fim de endereçar correctamente cada memória.

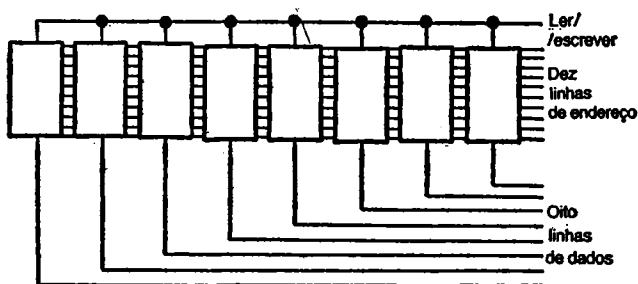


Figura 7.1.—Uso de memórias integradas de um bit. É necessária uma memória integrada para cada um dos oito bits de dados. No exemplo apresentado, que utiliza pastilhas de 1024 bits, são necessárias 10 linhas de endereço.

O endereçamento da memória é realizado pela CPU sob a direcção de um programa ou por contagem de números binários na linha de endereço. O número habitual de linhas de endereço da CPU é 16, como já se disse, o que permite o uso de 2^{16} endereços de memória (65 536). Dado que nenhuma pastilha de memória existente hoje requer 16 linhas de endereço é necessária uma selecção, que é feita por um terminal de selecção existente em cada pastilha de memórias. Imaginemos por exemplo uma pastilha de 128×8 memórias, como a popular 6810. Esta pode ser concebida para fornecer ao microprocessador, nas suas oito linhas de dados, qualquer dos 128 bytes que podem ser endereçados por sete linhas de endereço quando um dos terminais de selecção de pastilha se encontra ao nível lógico um. Quando este terminal está ao nível 0, a pastilha de memória é isolada, ficando flutuante e não sendo afectada por qualquer sinal de endereço ou de dados. O terminal de selecção da pastilha pode ser controlado por uma porta de tal modo que se o número de linhas de endereço for superior a 128, um 1 em qualquer das linhas entre a oitava e a décima sexta indicará que esta pastilha não é escolhida. É então possível ligar outra semelhante ao mesmo conjunto de sete linhas de endereço, mas com o seu terminal de selecção de pastilha controlado

por outra porta, de tal modo que a tensão do terminal é 1 quando a oitava linha de endereço têm presente um 1. Pode-se escolher uma outra pastilha semelhante colocando a nona linha de endereço ao nível 1, etc.

Alguns microprocessadores não utilizam todas as linhas de endereço (16) como saídas da CPU. Um modelo notável o INS8060 da National Semiconductor, utiliza 12 terminais para os 12 bits inferiores de endereço, e mais quatro que podem ser usados pilotando as linhas de dados com um impulso existente noutro terminal. Um outro tipo, a F8 da Fairchild, não possui terminais de endereço na CPU, utilizando uma pastilha de memória separada com endereçamento e contagem incorporados.

O total de 16 terminais de endereço e oito terminais de dados, numerados a partir de 0, requer ligações entre a CPU e todas as pastilhas que a rodeiam. As comunicações são feitas por um *bus*, um conjunto de linhas paralelas num painel de circuito impresso, ou por um conjunto de cabos ligando terminais correspondentes de cada pastilha. O *bus* de dados consiste em oito linhas, numeradas de D0 a D7, que transferem bits de dados do microprocessador para as memórias e vice-versa. O bus de endereços consiste em dezasseis linhas, numeradas A0 a A15, que enviam informação de endereços às memórias e outras pastilhas. Estas ligações correspondem a 24 dos 40 terminais normais das CPUs.

Quanto aos terminais restantes na CPU, parte são usados para alimentação e sinais de relógio. As primeiras CPUs utilizavam várias linhas de alimentação a diferentes tensões; por exemplo, a 8080 usava + 12,0 V, + 5,0 V e - 5,0 V, com uma massa comum. Os modelos posteriores passaram a requerer apenas uma linha de alimentação, normalmente + 5,0 V, pelo que se tornam necessários apenas dois terminais de alimentação, + e massa, o que tem a vantagem de libertar uma maior quantidade de terminais para os sinais. Dado que todas as operações do microprocessador são síncronas, é necessário um impulso de relógio. Alguns microprocessadores possuem circuitos osciladores internos, pelo que é possível produzir impulsos de relógio ligando um circuito sensível à frequência entre dois terminais. Este

circuito pode consistir apenas numa resistência e num condensador no caso de impulsos de relógio que não devam ser mantidos a uma frequência constante, ou um cristal de quartzo trabalhando a 1 MHz ou mais. O uso de um gerador externo de impulsos de relógio é mais vulgar, e requer apenas um terminal de entrada no caso de se usar um relógio de fase única. Alguns modelos necessitam de duas fases, existindo até uma pastilha que necessita de quatro.

Os restantes terminais da CPU são usados para sinais de controlo que são igualmente passados por um bus de ligações. Os sinais neste bus variam bastante quando se consideram diferentes CPUs, mas existem alguns factores necessariamente comuns a todos os tipos. Os sinais que controlam a leitura ou a escrita de memórias devem fazer parte do bus de controlo, sendo designados por *sinais de definição do bus de dados* visto que decidem qual o sentido em que os sinais passam nas linhas do bus de dados. Usam-se também sinais de temporização que permitem à CPU esperar até os dados estarem prontos a serem fornecidos. Uma entrada de sinal «ready», por exemplo, indica ao ser aplicada à CPU que uma determinada memória tem dados prontos a serem lançados no bus. Isto é necessário porque se observa sempre um atraso entre a saída de um endereço nos terminais de endereço da CPU e a saída de dados de outras pastilhas integradas podendo este atraso variar de um tipo de pastilha para outro. Nestas condições, durante um reduzido período de tempo depois de o endereço ter surgido nas linhas, os sinais do bus de dados flutuarão enquanto funcionam os circuitos de porta da memória. Até estas flutuações terem terminado e os sinais estarem estáveis não deve ser lida qualquer informação, sendo o sinal «ready» usado para indicar este estado.

Um sinal «hold» tem como efeito desligar a CPU das linhas de dados e das linhas de endereço, de tal modo que estas linhas possam ser controladas por outros circuitos. Durante o período de espera a CPU não está ligada e apresenta num terminal de saída um sinal que avisa a suspensão do seu funcionamento normal. Os sinais «ready» e «hold» são usados na CPU 8080, mas outras CPUs utilizam sinais

de temporização que recebem nomes diferentes que em certos casos possuem funções bastante diversas. Todas as CPUs possuem, no entanto, um terminal de limpeza que apaga o registo de contagem do programa (ver adiante), permitindo portanto o início de um programa em alguns endereços previamente determinados da memória.

O terceiro conjunto de sinais de controlo tem a ver com as interrupções. Os microprocessadores funcionam em sincronia com um ciclo de relógio bem definido mas podem ser obrigados a ler sinais de outros dispositivos, como sejam conversores analógico-digítals, que não são accionados por impulsos de relógio com a mesma frequência, ou de dispositivos, como as portas lógicas, que não são temporizados de modo algum. O método usado para permitir estas entradas, que podem ocorrer em qualquer ponto durante um programa, é designado por *interrupção*. A maneira de realizar uma interrupção varia conforme a CPU, mas basicamente consiste no envio, pelo dispositivo que deve fornecer um byte, de um sinal lógico chamado *pedido de interrupção* que pode, por exemplo, ser um 0 lógico numa linha que normalmente se encontra ao nível 1 e se encontra ligada a um terminal correspondente da CPU. No final da instrução de programa que a CPU está a realizar, a tensão noutro terminal da CPU, designado por *aceitação da interrupção*, é modificada. Isto constitui um sinal de que o dispositivo que pediu a interrupção pode fornecer os dados que possui. Simultaneamente, a CPU deve comutar para um programa que trate os sinais de interrupção — chama-se a isto uma *rotina de interrupção*.

Se o programador não preparou nenhuma rotina de interrupção o microprocessador deixará de funcionar, dado que não possui quaisquer instruções quanto à maneira de tratar o pedido de interrupção. O início da rotina de interrupção deve incluir instruções que preservem o conteúdo dos registos da CPU que existam no momento da interrupção, tornando assim possível continuar o programa depois desta.

A maior parte dos microprocessadores só permitem interrupções depois de terem sido processadas determinadas partes do programa principal, usando uma instrução designada

«interrupt enable», e podem impedir uma interrupção de ser por sua vez interrompida usando uma instrução chamada «interrupt disable».

Entradas e saídas

Até agora tratámos principalmente de sinais passados entre a CPU e as memórias, mas dado que o objectivo do microprocessador consiste em controlar máquinas através do envio de sinais digitais, é necessário que existam alguns sinais de saída. Do mesmo modo, deve existir alguma maneira de realizar a entrada de programas diferentes do monitor, de dados, e de instruções como «go», «halt», «reset». Estes sinais são tratados por circuitos integrados designados pelo nome PIA ou PIO — Adaptador de Interface Periférica ou Entrada/Saída Periférica.

As técnicas variam consideravelmente de um microprocessador para outro, mas um sistema muito comum consiste em ligar as linhas do bus de dados uma pastilha PIA que é comutada por sinais de controlo ou por uma combinação de sinais de endereço e controlo. Quando o PIA é escolhido, quer por um endereço como se fizesse parte da memória quer por sinais de selecção de memória, realiza ligações de porta às linhas de dados. Além disso, um outro sinal de controlo determinará se o PIA vai actuar como um terminal (ou *port*) de entrada ou saída. Se se escolhe o PIA de modo a funcionar como uma saída, então os sinais presentes nas linhas de dados nesse momento surgirão igualmente nos oito terminais de saída do PIA. Uma pastilha PIA possui normalmente dois conjuntos de terminais de entrada/saída de dados, e como estes podem ser usados como entradas ou saídas em qualquer combinação, um grupo de oito terminais pode controlar ou aceitar sinais de oito dispositivos necessitando apenas de sinais de dois níveis (0 ou 1). A selecção do PIA encontra-se completamente sob controlo do programa CPU, pelo que certas fases elementares como o uso de uma entrada de teclado ou a escrita de dados num conjunto de displays são impossíveis

a menos que exista um programa monitor que instrua convenientemente a CPU. Alguns tipos de CPUs possuem potencialidades PIA incorporadas, mas isto é raro.

Funcionamento da CPU

Dado que uma descrição completa do funcionamento de qualquer tipo de CPU encheria todo um livro igual a este, só tentaremos aqui uma breve descrição. A CPU microprocessadora não é um dispositivo simples, e as explicações simples são normalmente demasiado vagas e enganadoras a menos que o leitor tenha já uma experiência razoável no campo da electrónica digital.

A CPU possui diversos registos do tipo entrada paralela e saída paralela, alguns de oito bits e outros de dezasseis bits. Dois registos particularmente importantes são o acumulador, ao qual são enviados os resultados dos cálculos, e o contador de programa, que controla o desenvolvimento do programa de aplicação, de uma instrução para a seguinte. Todos os registos são, como é óbvio, controlados pelos impulsos de relógio, mas isto não significa que cada passo do programa seja realizada no tempo correspondente a um impulso de relógio. É muito mais habitual que sejam necessários bastantes impulsos de relógio para realizar um único passo do programa, chegando a 88 em algumas operações, dado que se realizam sequencialmente no interior da pastilha várias operações por cada passo do programa, e cada acção necessitará por sua vez de um ou mais impulsos de relógio.

Imaginemos um microprocessador constituído por uma ROM, uma RAM e PIO (figura 7.2). O conteúdo da ROM será o programa monitor, e o endereço do início do programa será o binário 0000000000000001 (os zeros à esquerda serão omitidos daqui em diante), pelo que a CPU partirá sempre deste ponto. Quando se liga a alimentação, os circuitos em torno da CPU accionam o terminal «reset», e a limpeza continua durante vários ciclos de relógio, dando ao relógio tempo a adquirir uma oscilação estável e permitindo simultaneamente a limpeza dos registos. Muitas vezes dá-se a esta

operação um tempo de 500 impulsos de relógio. Isto não bastará necessariamente para limpar a RAM, pelo que cada endereço da RAM conterá um byte de dígitos desconhecidos.

Quando esta acção termina, o registo de contagem do programa passa de 0 a 1 (os zeros à esquerda são omitidos aqui), indicando o endereço de início da pastilha ROM que fornece o programa de operação. Cada instrução do programa de operação requerirá vários impulsos de relógio para ser completada, e pode envolver a transferência de dados entre o registo acumulador e as posições de memória ou os «ports» de entrada/saída. Depois de cada instrução ter sido completada o contador de programa é aumentado de uma unidade.

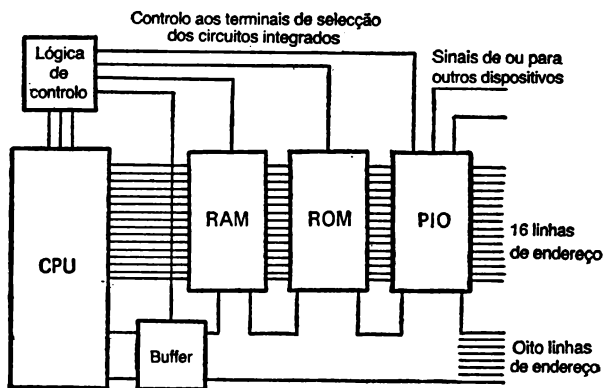


Figura 7.2. — Um sistema microprocessador elementar. A ROM contém o programa, e a RAM é usada para armazenar informação de entrada e resultados temporários. A PIO liga dispositivos externos, como teclados ou displays. É mostrado um buffer integrado, nas linhas de dados, que serve para isolar a CPU das memórias e PIA quando necessário, o que nem sempre acontece.

O programa monitor é normalmente concebido de tal modo que passe ciclicamente na ausência de outras instruções, de tal modo que pode ser usado um teclado ligado a um «port» de entrada para produzir um endereço RAM e colocar um byte de entrada nesse endereço. Dado que a

velocidade de produção de impulsos de relógio pelo microprocessador é tão rápida que é possível completar muitos ciclos de programa durante o tempo em que a tecla de entrada é premida, os teclados encontram-se normalmente ligados de modo a funcionarem por um sistema de interrupções. Em vez de utilizar uma descodificação no teclado (convertendo teclas hexadecimais para o código binário), a descodificação é normalmente realizada pelo microprocessador sob o controlo do programa monitor. Do mesmo modo a apresentação de caracteres hexadecimais em displays de sete segmentos é realizada por programação e não através de pastilhas descodificadoras. Um bom programa monitor permite ao utilizador inspeccionar cada fase de um programa escrito numa RAM, verificando o conteúdo de cada posição de memória RAM, voltando ao ponto inicial de um programa RAM, e modificando qualquer programa da maneira necessária. Além disto, quando é escolhido o ponto inicial de um programa, deve existir uma tecla «go» que leve o microprocessador a passar o programa, e uma tecla «reset» que torne possível voltar às condições de partida se o programa da RAM for defeituoso. Torna-se muito conveniente passar o programa passo a passo em vez de o fazer à velocidade normal de impulsos.

O esquema já descrito é o de um sistema reduzido contendo apenas um número mínimo de componentes capazes de tratar programas simples. A escrita de um programa para um microprocessador é extremamente aborrecida mesmo quando são usados caracteres hexadecimais, e esta escrita não será descrita aqui; no entanto, o acrescento de mais circuitos periféricos pode facilitar consideravelmente a tarefa de programação. Uma pastilha integrada muito útil é o conversor em série que converte o byte de oito bits para uma forma serial, um impulso de cada vez, permitindo assim o registo de informação em fita, o seu envio para uma tele-impressora ou um terminal TV e a sua apresentação ao exterior (se existir uma descodificação apropriada). Utilizando um teclado

letras/dígitos completo que apresente na saída números binários na forma ASCII standard, a tarefa de programação pode ser realizada numa linguagem de nível superior.

Isto implica a possibilidade de apresentar as instruções num teclado sob a forma de palavras e números, segundo um padrão que é mais fácil de reconhecer do que os números binários do código-máquina ou os seus equivalentes hexadecimais. Dado que o microprocessador ainda funciona em código binário, no entanto, é necessário usar um programa que converta cada instrução do teclado no conjunto correcto de algarismos em código-máquina. A fase seguinte ao código-máquina consiste num programa «assembler» que permite a entrada de curtas palavras em código mnemónico, em vez de binário ou hexadecimal. Este programa «assembler» ocupa apenas um espaço relativamente reduzido da ROM, porque a linguagem usada é razoavelmente semelhante ao código-máquina — trata-se apenas de converter uma instrução como LDI («Load immediate») em código-máquina, por exemplo 11000100. A fase seguinte consiste num programa compilador ou intérprete; este permite o uso de uma linguagem de processamento como a BASIC («Beginners' All-purpose Symbolic Instruction Code»), a FORTRAN ou a ALGOL. Um programa compilador requer uma vasta quantidade de espaço de memória, ROM se o programa compilador é parte integrante do sistema ou RAM se é registado em banda e carregado na máquina sempre que é necessário usar o programa. No caso das aplicações de microprocessadores que requerem um accionamento contínuo do teclado (como terminais de tele-impressão, processadores de palavras), estes compiladores são necessários; mas para o desenvolvimento de sistemas simples de controlo da máquina torna-se mais útil um programa assembler, relativamente simples.

Para terminar esta descrição necessariamente breve, convirá mencionar o lado «software» da programação. Até ter sido programado, um sistema microprocessador é completamente inútil. Para cada CPU existe um conjunto de instruções

«software» — uma lista das códigos-máquina que devem ser usados como instruções capazes de serem compreendidas pela máquina e levá-la a processar um dado programa. Depois de o programa ter sido escrito, estes bytes de código serão alimentados ao bus de dados a partir da memória de programa de cada vez que o contador de programa é aumentado de uma unidade, e a acção que é exigida nesse passo de programação será realizada antes de o contador de programa ser novamente incrementado.

Com apenas algumas excepções, os conjuntos de instruções variam consideravelmente, pelo que é praticamente impossível memorizar mais do que um. As excepções notáveis são o 8080 e o Z80, pois este é um desenvolvimento daquele e utiliza todos os conjuntos de instruções do 8080 mais alguns que lhe são próprios. Uma instrução consiste geralmente em um ou mais bytes seguidos por um ou mais bytes de dados. A codificação dos bytes de instruções indica à CPU se se seguem ou não quaisquer bytes de dados ou de novas instruções.

Como exemplo, note-se que a instrução «halt» não requer qualquer byte em seguida, mas uma instrução como LDI que indica à máquina que deve carregar no acumulador o byte contido na posição de memória seguinte, deve ser evidentemente seguida de um byte de dados, senão a instrução não pode ser realizada. O byte que se segue ao código de instrução pode no entanto completar a instrução de outros modos, cada um dos quais é indicado pelo primeiro byte do código de instruções. Por exemplo, o segundo byte pode especificar um certo número de passos de programação para a frente ou para trás em termos de contador de programa, de tal modo que toda a contagem se desloca para um número completamente diferente do que seria obtido por simples incrementação de uma unidade. Uma outra possibilidade consiste em que o byte que se segue ao da instrução indicar um endereço da memória, e ser o número armazenado nesse endereço que constitui de facto a instrução seguinte ou que deve ser usado num cálculo de acordo com a codificação do primeiro byte de instruções.

Para um conhecimento mais detalhado da programação de microprocessadores o leitor deve consultar um manual de programação, de preferência um dos excelentes manuais fornecidos juntamente com unidades de desenvolvimento de microprocessadores como o KIM-1. Infelizmente nem todos os manuais ou mesmo livros de informação dos fabricantes têm esta elevada qualidade, e uma grande quantidade de folhetos não «oficiais» sobre programação não são suficientemente bem impressos ou bem explicados para serem compreendidos por um principiante.

ÍNDICE

Sinais, comutação e componentes	7
Portas lógicas ...	39
Entradas de relógio e outras ...	63
Flip-flops e contadores ...	85
Registos de deslocamento	107
«Displays» e pilotagem	131
Sistemas de microprocessamento	155

Este livro acabou de se imprimir
em 1985
para a
EDITORIAL PRESENÇA, LDA.
na
Empresa Gráfica Feirense, Lda.
Vila da Feira

Escrito por um técnico de renome, esta obra dá-lhe as bases necessárias para a introdução às técnicas digitais. Cada capítulo explica pormenorizadamente os temas a tratar com base em exercícios práticos. Abrange todo um campo que vai desde sinais e comutadores até portas lógicas, sinais temporizados, *flip-flops*, registo de deslocamento, etc., sendo o último capítulo dedicado inteiramente ao sistema de microprocessadores. Mais de 120 ilustrações e tabelas completam um texto perfeitamente compreensível.

