

**MÓDULO 8810**

***MANUAL TÉORICO E  
PRÁTICO***





**MÓDULO 8810**

**MANUAL TEÓRICO E PRÁTICO**



**SUMÁRIO**

<b>CAPÍTULO 1 – GATES LÓGICOS BÁSICOS</b>	<b>1</b>
<b>1.1 FUNÇÃO LÓGICA GATES “E” (AND)</b>	<b>1</b>
1.1.1 RESUMO	1
1.1.2 OBJETIVO	1
1.1.3 EXPERIÊNCIA 1	1
1.1.4 EXPERIÊNCIA 2	3
1.1.5 DEMORA DE PROPAGAÇÃO (DELAY TIME)	4
1.1.6 EXERCÍCIOS	6
<b>1.2 FUNÇÃO LÓGICA GATE “OU” (OR)</b>	<b>7</b>
1.2.1 RESUMO	7
1.2.2 OBJETIVO	7
1.2.3 EXPERIÊNCIA 3	7
1.2.4 EXPERIÊNCIA 4	8
1.2.5 EXPERIÊNCIA 5	9
1.2.6 ANÁLISE DE DEMORA	11
1.2.7 EXERCÍCIOS	12
<b>1.3 FUNÇÃO LÓGICA GATE “NÃO” (INVERSOR) (NOT)</b>	<b>13</b>
1.3.1 RESUMO	13
1.3.2 OBJETIVO	13
1.3.3 EXPERIÊNCIA 6	13
1.3.4 EXERCÍCIOS	16
<b>1.4 FUNÇÃO LÓGICA GATE “NÃO-E” (NAND)</b>	<b>17</b>
1.4.1 RESUMO	17
1.4.2 OBJETIVO	17
1.4.3 EXPERIÊNCIA 7	17
1.4.4 EXPERIÊNCIA 8	19
1.4.5 EXPERIÊNCIA 9	20

1.4.6	EXPERIÊNCIA 10	21
1.4.7	EXPERIÊNCIA 11	22
1.4.8	EXERCÍCIOS	23
<b>1.5</b>	<b>FUNÇÃO LÓGICA GATE “NÃO-OU” (NOR)</b>	<b>24</b>
1.5.1	RESUMO	24
1.5.2	OBJETIVO	24
1.5.3	EXPERIÊNCIA 12	24
1.5.4	EXPERIÊNCIA 13	26
1.5.5	EXPERIÊNCIA 14	26
1.5.6	EXPERIÊNCIA 15	28
1.5.7	EXPERIÊNCIA 16	28
1.5.8	EXPERIÊNCIA 17	29
1.5.9	EXERCÍCIOS	31
<b>1.6</b>	<b>FUNÇÃO LÓGICA GATE “OU-EXCLUSIVO” (XOR)</b>	<b>32</b>
1.6.1	RESUMO	32
1.6.2	OBJETIVO	32
1.6.3	EXPERIÊNCIA 18	32
1.6.4	EXPERIÊNCIA 19	33
1.6.5	EXPERIÊNCIA 20	34
1.6.6	EXPERIÊNCIA 21	35
1.6.7	EXERCÍCIOS	37
<b>1.7</b>	<b>TEOREMA DE “DE MORGAN”</b>	<b>38</b>
1.7.1	RESUMO	38
1.7.2	EXPERIÊNCIA 22	38
1.7.3	EXPERIÊNCIA 23	39
1.7.4	EXPERIÊNCIA 24	41
1.7.5	EXERCÍCIOS	44
<b>1.8</b>	<b>GATES DE COLETOR ABERTO</b>	<b>45</b>
1.8.1	RESUMO	45

1.8.2	EXPERIÊNCIA 25	46
1.8.3	EXPERIÊNCIA 26	47
1.8.4	EXERCÍCIOS	49
<b>1.9</b>	<b>GATES TRI-STATE</b>	<b>50</b>
1.9.1	RESUMO	50
1.9.2	EXPERIÊNCIA 27	51
1.9.3	EXPERIÊNCIA 28	52
1.9.4	EXERCÍCIOS	54
<b>CAPÍTULO 2 – FLIP FLOPS</b>		<b>55</b>
<b>2.1</b>	<b>RESUMO</b>	<b>55</b>
<b>2.2</b>	<b>FLIP-FLOP S-R</b>	<b>55</b>
<b>2.3</b>	<b>FLIP-FLOP T</b>	<b>58</b>
<b>2.4</b>	<b>FLIP-FLOP D</b>	<b>58</b>
<b>2.5</b>	<b>FLIP-FLOP JK</b>	<b>59</b>
<b>2.6</b>	<b>TIPOS DE GATILHAMENTO DOS FLIP-FLOPS</b>	<b>59</b>
<b>2.7</b>	<b>FLIP-FLOPS GATILHÁVEIS DO TIPO MESTRE-ESCRAVO</b>	<b>61</b>
2.7.1	FLIP-FLOP MESTRE-ESCRAVO COM DATA LOCK-OUT	63
<b>2.8</b>	<b>PARÂMETROS IMPORTANTES DOS FLIP-FLOPS</b>	<b>63</b>
<b>2.9</b>	<b>EXPERIÊNCIA 1</b>	<b>64</b>
<b>2.10</b>	<b>EXPERIÊNCIA 2</b>	<b>65</b>
<b>2.11</b>	<b>EXPERIÊNCIA 3</b>	<b>67</b>
<b>2.12</b>	<b>EXPERIÊNCIA 4</b>	<b>68</b>
<b>2.13</b>	<b>EXERCÍCIOS</b>	<b>70</b>
<b>CAPÍTULO 3 - CONTADORES</b>		<b>71</b>
<b>3.1</b>	<b>DEFINIÇÃO</b>	<b>71</b>
<b>3.2</b>	<b>CONTADORES RIPPLE (ASSINCRONO)</b>	<b>71</b>
3.2.1	EXPERIÊNCIA 1	71

3.2.2	EXPERIÊNCIA 2	73
<b>3.3</b>	<b>CONTADORES SÍNCRONOS</b>	<b>74</b>
3.3.1.	EXPERIÊNCIA 3	74
<b>3.4</b>	<b>CONTADORES SÍNCRONOS CONTROLADOS</b>	<b>76</b>
<b>3.5</b>	<b>CONTADORES EM ANEL</b>	<b>76</b>
3.5.1	EXPERIÊNCIA 4	76
<b>3.6</b>	<b>CONTADORES NA FORMA DE CI's</b>	<b>78</b>
3.6.1	CONTADOR 74LS90	78
3.6.1.1	EXPERIÊNCIA 5	80
3.6.1.2	EXPERIÊNCIA 6	82
3.6.1.3	LIGAÇÃO DE CONTADORES EM CASCATA	84
3.6.1.4	EXPERIÊNCIA 7	84
3.6.1.5	EXPERIÊNCIA 8	86
3.6.1.6	EXPERIÊNCIA 9	88
3.6.1.7	EXPERIÊNCIA 10	89
3.6.1.8	EXPERIÊNCIA 11	91
<b>CAPÍTULO 4 - REGISTRADORES</b>		<b>95</b>
<b>4.1</b>	<b>RESUMO</b>	<b>95</b>
<b>4.2</b>	<b>REGISTRADORES BUFFER</b>	<b>95</b>
4.2.1	EXPERIÊNCIA 1	95
4.2.2	EXPERIÊNCIA 2	96
<b>4.3</b>	<b>REGISTRADORES DE DESLOCAMENTO</b>	<b>98</b>
4.3.1	EXPERIÊNCIA 3	98
4.3.2	EXPERIÊNCIA 4	99
4.3.3	EXPERIÊNCIA 5	100
<b>4.4</b>	<b>CARREGAMENTO PARALELO</b>	<b>101</b>
4.4.1	EXPERIÊNCIA 6: (OPCIONAL)	101
<b>4.5</b>	<b>REGISTRADORES NA FORMA DE CI's</b>	<b>103</b>

<b>CAPÍTULO 5 - DECODIFICADORES</b>	<b>105</b>
5.1 RESUMO	105
5.2 EXPERIÊNCIA 1	106
5.3 DECODIFICADOR BCD PARA DECIMAL	106
5.3.1 EXPERIÊNCIA 2	107
5.4 DECODIFICADOR DRIVER	108
5.4.1 DECODIFICADOR/DRIVERS BCD PARA 7 SEGMENTOS	109
5.4.1.1 EXPERIÊNCIA 3	110
<b>CAPÍTULO 6 - CODIFICADORES</b>	<b>113</b>
6.1 RESUMO	113
6.2 EXPERIÊNCIA 1	113
6.3 EXPERIÊNCIA 2	114
6.4 CHAVES CODIFICADORAS	115
6.5 CONVERSORES DE CÓDIGOS	116
6.5.1 EXPERIÊNCIA 3	116
<b>CAPÍTULO 7 - MULTIPLEXADORES</b>	<b>119</b>
7.1 RESUMO	119
7.2 MULTIPLEXADOR DE 2 ENTRADAS	119
7.2.1 EXPERIÊNCIA 1	120
7.3 MULTIPLEXADORES NA FORMA DE CI's	121
7.3.1 EXPERIÊNCIA 2	121
7.4 USO DE CI's MULTIPLEXADORES PARA FORMAR MULTIPLEXADORES MAIORES	123
<b>CAPÍTULO 8 - DEMULTIPLEXADORES</b>	<b>125</b>
8.1 RESUMO	125
8.2 DEMULTIPLEXADOR DE 4 SAÍDAS	126
8.2.1 EXPERIÊNCIA 1	126

<b>8.3 DEMULTIPLEXADORES NA FORMA DE CI's</b>	<b>127</b>
8.3.1 DEMULTIPLEXADOR 74LS155	127
8.3.1.1 EXPERIÊNCIA 2	128
8.3.1.2 EXPERIÊNCIA 3	130
 <b>CAPÍTULO 9 - COMPARADORES DIGITAIS</b>	 <b>133</b>
9.1 RESUMO	133
9.2 EXPERIÊNCIA 1	133
9.3 EXPERIÊNCIA 2	134
9.4 EXPERIÊNCIA 3	134
9.5 EXPERIÊNCIA 4	135
9.6 EXPERIÊNCIA 5	136
9.7 EXPERIÊNCIA 6	136
9.8 COMPARADOR DE MAGNITUDE NA FORMA DE CI	137
9.8.1 RESUMO	137
9.8.2 EXPERIÊNCIA 7	138
9.9 COMPARADORES MAIORES COM O 74LS85	140
 <b>CAPÍTULO 10 - GERAÇÃO E CHECK DE PARIDADE</b>	 <b>143</b>
10.1 RESUMO	143
10.2 CHEQUE DE PARIDADE	144
10.2.1 EXPERIÊNCIA 1	145
10.3 GERAÇÃO DE PARIDADE	145
10.3.1 EXPERIÊNCIA 2	146
10.4 CIRCUITOS INTEGRADOS PARA CHEQUE E GERAÇÃO DE PARIDADE	147
10.4.1 EXPERIÊNCIA 3	147
10.5 USO DO 74LS180 PARA CHEQUE DE PARIDADE DE 9 BITS	149
10.5.1 EXPERIÊNCIA 4	149
10.6 USO DO 74LS180 EM CASCATA	150

<b>CAPÍTULO 11 - CIRCUITOS ARITMÉTICOS</b>	<b>151</b>
<b>11.1 RESUMO</b>	<b>151</b>
<b>11.2 MEIO SOMADOR</b>	<b>151</b>
11.2.1 EXPERIÊNCIA 1	152
<b>11.3 MEIO SUBTRATOR</b>	<b>152</b>
11.3.1 EXPERIÊNCIA 2	153
<b>11.4 SOMADOR</b>	<b>154</b>
11.4.1 EXPERIÊNCIA 3	155
<b>11.5 SUBTRATOR</b>	<b>156</b>
11.5.1 EXPERIÊNCIA 4	158
<b>11.6 SUBTRAÇÃO ATRAVÉS DE SOMA</b>	<b>159</b>
11.6.1 COMPLEMENTO DE UM NÚMERO BINÁRIO	159
11.6.2 COMPLEMENTO DE DOIS DE UM NÚMERO BINÁRIO	159
11.6.3 CIRCUITO SUBTRATOR SOMADOR	160
<b>11.7 SOMADORES PARALELOS NA FORMA DE CI's</b>	<b>162</b>
11.7.1 EXPERIÊNCIA 5	162
<b>11.8 LIGAÇÃO DE SOMADORES CI's PARALELOS EM CASCATA</b>	<b>163</b>
11.8.1 EXPERIÊNCIA 6	164
11.8.2 EXPERIÊNCIA 7	165
<b>11.9 SOMADOR SÉRIE</b>	<b>166</b>
11.9.1 FUNCIONAMENTO DO SOMADOR SÉRIE	167
<b>11.10 OVERFLOW E UNDERFLOW NA ARITMÉTICA DO COMPLEMENTO DE 2</b>	<b>167</b>
11.10.1 OVERFLOW E UNDERFLOW NA ADIÇÃO	168
11.10.2 OVERFLOW E UNDERFLOW NA SUBTRAÇÃO	168
11.10.3 DETEÇÃO DE OVERFLOW E UNDERFLOW	169
11.10.4 EXPERIÊNCIA 8	170
<b>11.11 ARITMÉTICA BCD</b>	<b>171</b>

<b>CAPÍTULO 12 - UNIDADES ARITMÉTICAS E LÓGICAS (ALU's)</b>	<b>173</b>
<b>12.1 RESUMO</b>	<b>173</b>
<b>12.2 O 74LS181</b>	<b>173</b>
12.2.1 EXPERIÊNCIA 1	175
<b>12.3 USO DO COMPARADOR DO 74LS181</b>	<b>176</b>
12.3.1 EXPERIÊNCIA 2	177
 <b>CAPÍTULO 13 - MONOESTÁVEIS E ASTÁVEIS</b>	 <b>179</b>
<b>13.1 INTRODUÇÃO</b>	<b>179</b>
<b>13.2 MONOESTÁVEIS COM CI's</b>	<b>180</b>
13.2.1 O 74121	180
13.2.2 O 74221	182
13.2.3 O 74LS122	183
13.2.4 O 74LS123	184
13.2.5 O 555	185
13.2.6 EXPERIÊNCIA 1	189
13.2.7 EXPERIÊNCIA 2	190
13.2.8 EXPERIÊNCIA 3	191
13.2.9 EXPERIÊNCIA 4	193
13.2.10 EXPERIÊNCIA 5	195
13.2.11 EXPERIÊNCIA 6	195
13.2.12 EXPERIÊNCIA 7	196
<b>13.3 EXERCÍCIOS</b>	<b>198</b>
 <b>CAPÍTULO 14 – MEMÓRIAS SEMICONDUTORAS</b>	 <b>201</b>
<b>14.1 RESUMO</b>	<b>201</b>
<b>14.2 MEMÓRIAS RAM's</b>	<b>201</b>
<b>14.3 ORGANIZAÇÃO DE UMA MEMÓRIA RAM ESTÁTICA</b>	<b>202</b>
<b>14.4 TEMPOS IMPORTANTES DE MEMÓRIA</b>	<b>203</b>

<b>14.5 MEMÓRIAS RAM's BIPOLARES E MOS</b>	<b>204</b>
14.5.1 EXPERIÊNCIA 1	205
<b>14.6 CONSTRUÇÃO DE MEMÓRIAS MAIORES</b>	<b>208</b>
14.6.1 EXPERIÊNCIA 2	209
14.6.2 EXPERIÊNCIA 3	212
<b>14.7 MEMÓRIAS ROM</b>	<b>213</b>
14.7.1 EXPERIÊNCIA 4	217
 <b>CAPÍTULO 15 - INTERLIGAÇÕES LÓGICAS OU INTERFACES LÓGICAS</b>	 <b>219</b>
<b>15.1 RESUMO</b>	<b>219</b>
<b>15.2 INTERFACES CMOS – TTL</b>	<b>220</b>
15.2.1 INTERFACES CMOS – TTL DE BAIXA POTÊNCIA (Low-Power)	220
15.2.2 CMOS PARA TTL STANDARD	221
15.2.2.1 EXPERIÊNCIA 1	221
15.2.2.2 EXPERIÊNCIA 2	222
15.2.3 INTERFACE TTL – CMOS	223
15.2.3.1 EXPERIÊNCIA 3	224
<b>15.3 CIRCUITOS INTEGRADOS SERVINDO DE INTERFACES</b>	<b>224</b>

## **ATENÇÃO!**

O Módulo Digital Avançado 8810 possui uma **chave de seleção TTL/CMOS**, que quando posicionada em TTL faz com que as chaves de dados, os led's e a ponta de prova operem com **níveis lógicos compatíveis com circuitos TTL's, os quais são alimentados com +5 Volts.**

Estando esta chave posicionada em CMOS, as chaves de dados, os led's e a ponta de prova passam a operar com **níveis lógicos compatíveis com circuitos CMOS, que são alimentados com tensão de + 15 Volts.**

Deste modo, **em todas as experiências deste manual esta chave de seleção deverá permanecer na posição TTL.** Em caso contrário, as placas de experiências e o próprio módulo serão danificados.

**As condições de garantia do equipamento não cobrem operações indevidas no mesmo.**



## **CAPÍTULO 1 – GATES LÓGICOS BÁSICOS**

### **1.1 FUNÇÃO LÓGICA GATES “E” (AND)**

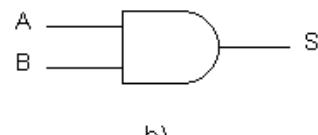
#### **1.1.1 RESUMO**

A função “E” é definida como:

“Produz uma saída 1 somente quando todas as entradas forem 1”. Desta maneira o circuito lógico “E” dará um nível alto (1) na saída quando todas as entradas estiverem em nível lógico alto (1). A figura 1 mostra a tabela verdade, o símbolo e a expressão booleana para o gate “E”.

ENTRADAS		SAÍDAS
A	B	$S = A \cdot B$
0	0	0
0	1	0
1	1	1
1	0	0

a)



$$S = A \cdot B = AB$$

c)

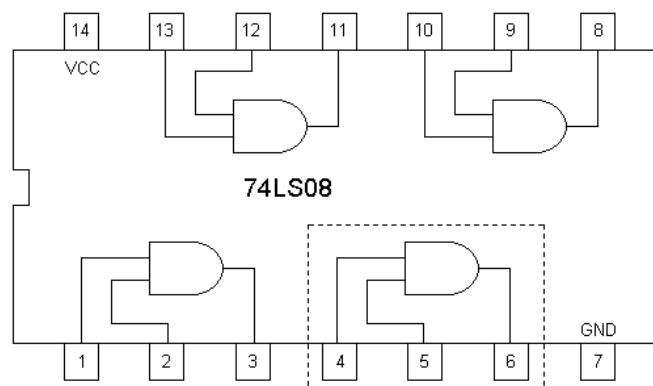
**Figura 1 - Gate E - a) Tabela Verdade - b) Símbolo - c) Expressão Booleana**

#### **1.1.2 OBJETIVO**

A familiarização com os GATES E, tabela verdade, conceito de entradas múltiplas e demora de propagação.

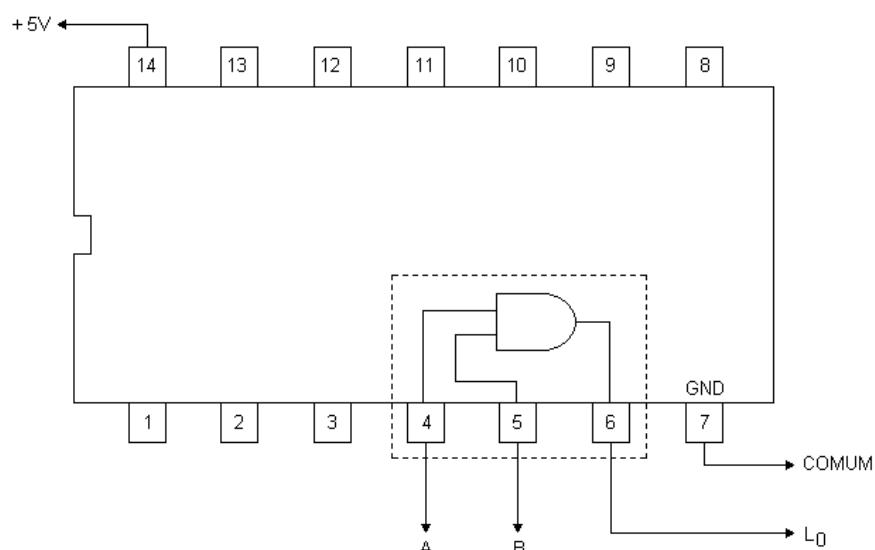
#### **1.1.3 EXPERIÊNCIA 1**

- a)** Material utilizado: 1 x CI 74LS08
- b)** Layout do CI 74LS08



**Figura 2 - Layout do CI 74LS08**

c) Montar o circuito da figura 3.



**Figura 3 - Circuito com o GATE E**

d) Completar a tabela 1.

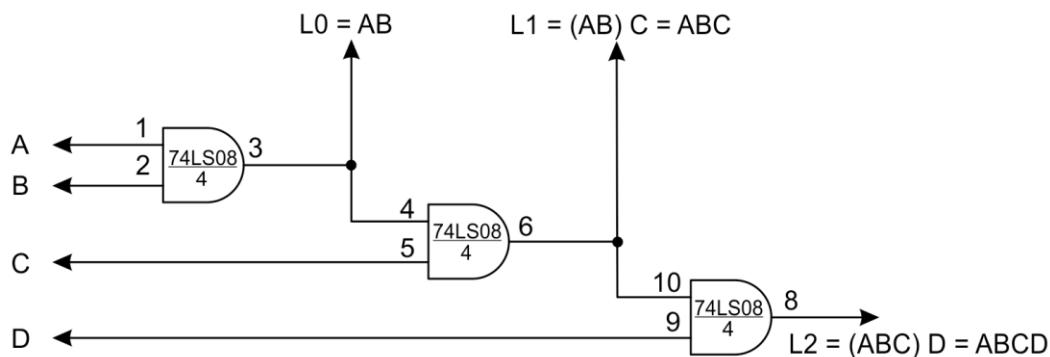
ENTRADAS		SAÍDAS
A	B	$L_0 = A \cdot B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

NOTA: O led aceso  
indica 1, apagado Ø.

**Tabela 1 - Tabela Verdade da Função E de duas entradas.**

### 1.1.4 EXPERIÊNCIA 2

- a) Material utilizado: 1 x CI 74LS08  
 b) Montar o circuito da figura 4, ligando o pino 14 ao +15V e o pino 7 ao comum.



**Figura 4 - Circuito E de 4 entradas.**

- c) Completar a tabela 2.

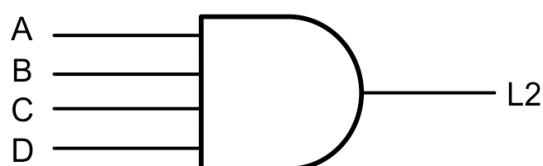
ENTRADAS				SAÍDAS		
A	B	C	D	L0 = AB	L1 = ABC	L2 = ABCD
Ø	Ø	Ø	Ø			
Ø	Ø	Ø	1			
Ø	Ø	1	Ø			
Ø	Ø	1	1			
Ø	1	Ø	Ø			
Ø	1	Ø	1			
Ø	1	1	Ø			
Ø	1	1	1			
1	Ø	Ø	Ø			
1	Ø	Ø	1			
1	Ø	1	Ø			
1	Ø	1	1			
1	1	Ø	Ø			
1	1	Ø	1			
1	1	1	Ø			
1	1	1	1			

**Tabela 2 - Tabela Verdade do GATE E de 4 entradas.**

- d) Observação: No circuito testado foi montado um gate “E” de 4 entradas usando gates “E” de duas entradas. Usando as propriedades da álgebra de Boole foi feito:

$$L2 = ABCD = ((AB) C) D$$

Em termos de blocos lógicos, teremos o mostrado na figura 5.



**Figura 5 - Gate “E” de 4 entradas.**

### 1.1.5 DEMORA DE PROPAGAÇÃO (DELAY TIME)

É o tempo requerido para a saída do gate mudar de estado após as entradas terem mudado.

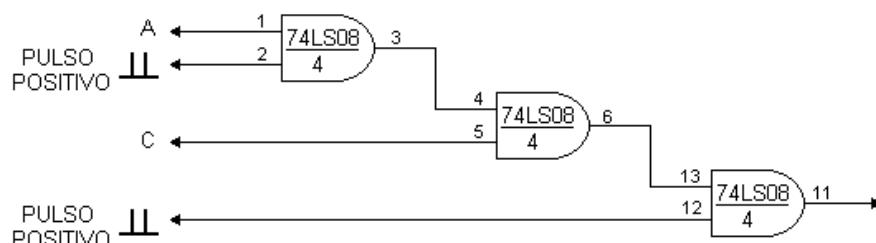
Um gate TTL típico possui uma demora de propagação de 10ns. Esta demora de propagação depende da tensão de alimentação, temperatura ambiente e da carga capacitiva de saída.

O significado de algumas simbologias referentes a tempo e encontradas nas folhas de dados são:

$t_{PLH}$  - Demora de propagação quando a saída está mudando de um nível  $\emptyset$  (baixo), para um nível 1 (alto).

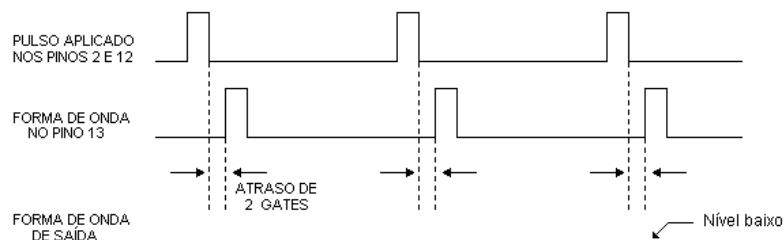
$t_{PHL}$  - Demora de propagação quando a saída está mudando de um nível 1 (alto), para um nível  $\emptyset$  (baixo).

Ambos os parâmetros,  $t_{PHL}$  e  $t_{PLH}$ , são medidas com respeito ao pulso de entrada. Os circuitos das figuras 6 e 8 ilustram a demora de propagação. Devido a estes tempos estarem na ordem de nanosegundos não poderão ser observados a olho nu, e sim com o uso de equipamentos de alta frequência.

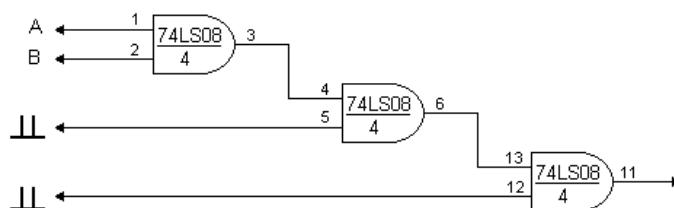


**Figura 6 - Circuito para teste de demora de propagação.**

Na figura 6, se  $A = 1$ ,  $C = 1$  e os pinos 2 e 12 estão ligados a um gerador de pulso ( $\perp\perp$ ) com uma frequência de 1Hz e com largura de pulso menor que a demora de propagação do bloco lógico, então a saída (pino 11) ficaria constantemente em zero, devido aos instantes de ocorrência dos pulsos nos pinos 12 e 13 acontecerem em momentos não coincidentes. A figura 7 ilustra estes atrasos.

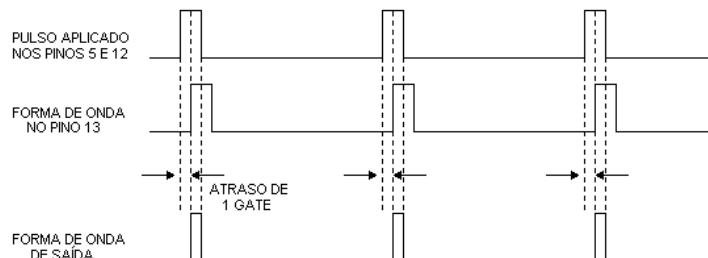


**Figura 7 - Formas de ondas correspondente a figura 6.**



**Figura 8 - Circuito para teste de demora de propagação.**

Na figura 8, se  $A = 1$  e  $B = 1$ , na saída teríamos um pulso a cada segundo, com duração igual ao instante de coincidência dos pulsos. A figura 9 ilustra estes atrasos.



**Figura 9 - Formas de onda correspondente a Figura 8**

Disto conclui-se que devido à Demora de Propagação de cada gate na figura 6, o sinal aplicado no pino 2 vai chegar a entrada do gate de saída após ter terminado o pulso introduzido na outra entrada deste gate; desta maneira a saída permanece em zero. Na figura 8, o atraso não é suficiente, de maneira que os sinais chegam a tempos próximos um do outro no gate de saída, dando assim uma saída adequada, isto, é um pulso.

### 1.1.6 EXERCÍCIOS

- a) Em um *gate* E de 3 entradas A, B e C, para A = 1, B = 1 e C = Ø a saída será:

1. 1
2. Ø
3. Indeterminada

- b) Se ligarmos todas as entradas de um *gate* E a uma só chave, a saída será:

1. Igual a entrada
2. O inverso da entrada
3. A função OU

- c) A saída de um *gate* E de 3 entradas pode ser representada pela expressão Booleana:

1.  $(A \cdot B \cdot C)$
2.  $(A \cdot B) C$
3.  $A (B \cdot C)$
4.  $(B C) A$
5.  $C \cdot A \cdot B$

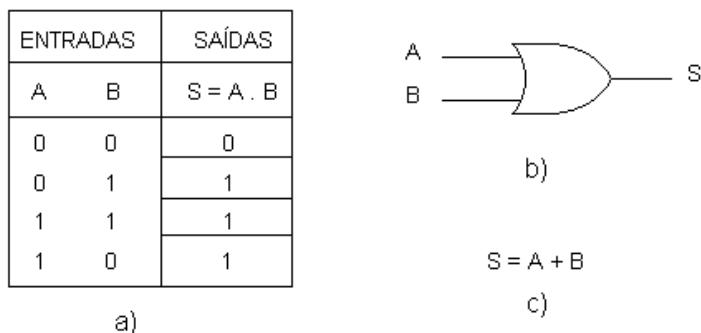
## 1.2 FUNÇÃO LÓGICA GATE “OU” (OR)

### 1.2.1 RESUMO

A função “OU” é definida como:

**“Se uma ou mais entradas forem 1 a saída será 1”.**

Desta maneira, o circuito lógico “OU” dará uma saída alta (1) se pelo menos uma de suas entradas estiver no nível alto (1). A figura 10 mostra a tabela verdade, o símbolo e a expressão booleana para o gate “OU”.



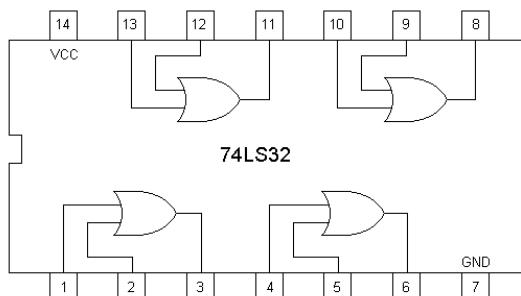
**Figura 10 - Gate OU - a) Tabela Verdade - b) Símbolo Lógico - c) Expressão Booleana**

### 1.2.2 OBJETIVO

Neste ponto será examinada a tabela verdade do gate “OU” e o conceito de múltiplas entradas e entradas flutuando.

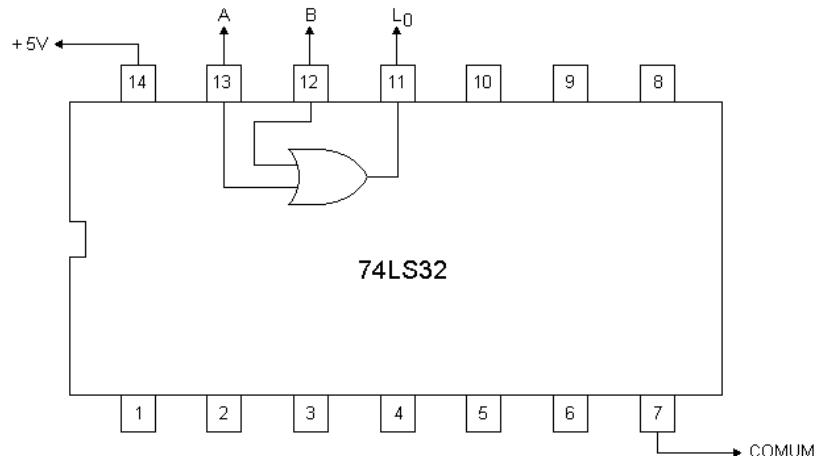
### 1.2.3 EXPERIÊNCIA 3

- a) Material utilizado: 1 x CI 74LS32
- b) Layout do CI 74LS32



**Figura 11 - Layout do CI 74LS32.**

**c)** Montar o circuito da figura 12.



**Figura 12 - Circuito com o Gate OU.**

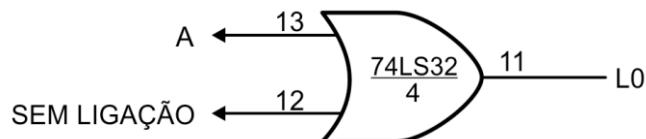
**d)** Completar a tabela 3.

ENTRADAS		SAÍDAS
A	B	$L\bar{0} = A + B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 3 - Tabela Verdade do Gate “OU”.**

#### 1.2.4 EXPERIÊNCIA 4

- a)** Material utilizado: 1 x CI 74LS32  
**b)** Montar o circuito da figura 13



**Figura 13 - Gate “OU” com uma entrada flutuando.**

c) Completar a tabela 4.

ENTRADAS	SAÍDAS
A	LØ
Ø	
1	

**Tabela 4 - Tabela Verdade do Gate OU com uma entrada flutuando.**

#### Observação:

Notar que o circuito é independente de A, isto significa que o pino 12 (que está flutuando) introduz um nível lógico 1 no circuito.

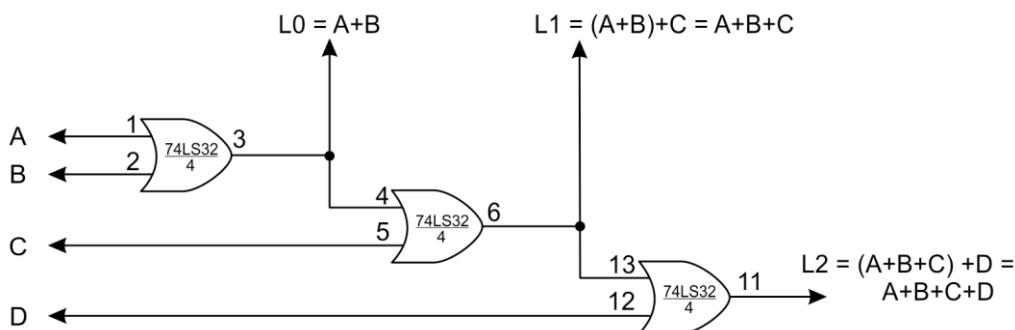
Isto demonstra uma propriedade dos circuitos integrados da série TTL-74.

**“Na tecnologia TTL (série 74) um pino de entrada sem conexão funciona como nível lógico 1”.**

Na prática, entretanto, para montagens definitivas, não se deve deixar pinos de entradas sem conexão, pois os mesmos poderão operar como antenas recebendo ruídos, que alteram a operação do circuito.

#### 1.2.5 EXPERIÊNCIA 5

- a) Material utilizado: 1 x CI 74LS32
- b) Montar o circuito da figura 14, ligando o pino 14 ao +5V e o pino 7 ao comum.



**Figura 14 – Circuito OU de 4 entradas.**

c) Completar a tabela 5.

ENTRADAS				SAÍDAS		
A	B	C	D	L0 = A+B	L1 = A+B+C	L2 = A+B+C+D
Ø	Ø	Ø	Ø			
Ø	Ø	Ø	1	1		
Ø	Ø	1	Ø	Ø		
Ø	Ø	1	1	1		
Ø	1	Ø	Ø	Ø		
Ø	1	Ø	1	1		
Ø	1	1	Ø	Ø		
Ø	1	1	1	1		
1	Ø	Ø	Ø	Ø		
1	Ø	Ø	1	1		
1	Ø	1	Ø	Ø		
1	Ø	1	1	1		
1	1	Ø	Ø	Ø		
1	1	Ø	1	1		
1	1	1	Ø	Ø		
1	1	1	1	1		

**Tabela 5 - Tabela Verdade do GATE E de 4 entradas.**

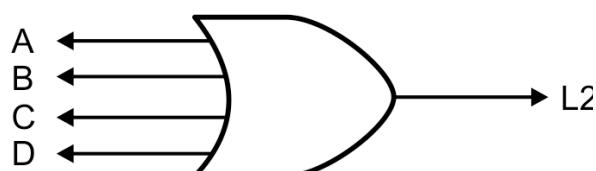
**Observação:**

O circuito montado executou a função de um gate “OU” de 4 entradas, porém usa gates “OU” de duas entradas.

Isto foi conseguido lançando mão de uma das propriedades da álgebra de Boole que diz:

$$L2 = A + B + C + D = ((A + B) + C) + D$$

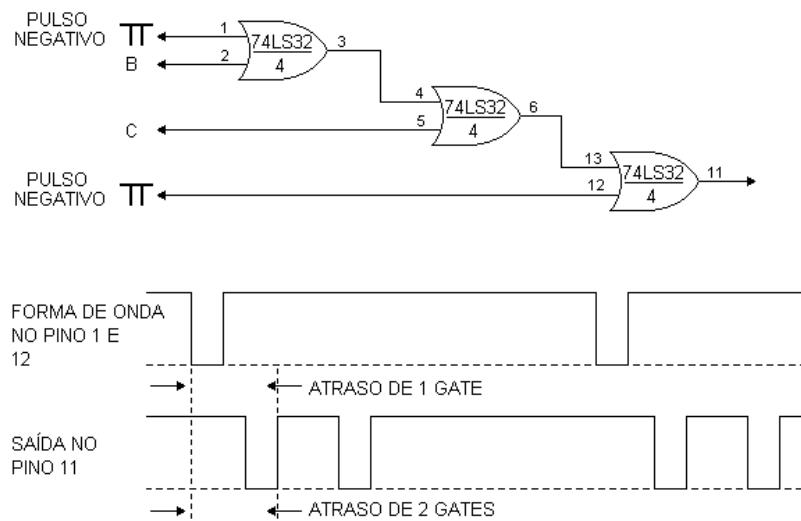
Em termos de blocos lógicos tem-se o mostrado na figura 15.



**Figura 15 – Gates “OU” de 4 entradas.**

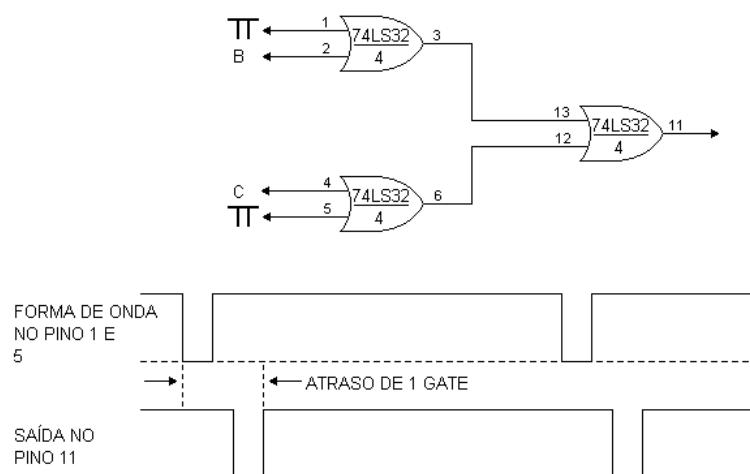
### 1.2.6 ANÁLISE DE DEMORA

O circuito das figuras 16 e 17 ilustram a demora de propagação.



**Figura 16 - Circuito do Gate OU com 4 entradas.**

Na figura 16, se  $B = \emptyset$ ,  $C = \emptyset$  e os pinos 1 e 12 são ligados a um gerador de pulso negativo (  $\Pi$  ) com uma frequência de 1Hz e com largura de pulso menor que a demora de propagação, ter-se-á na saída, pino 11, a forma de onda apresentada na figura 16. Isto porque o pulso negativo de entrada sofre atrasos diferentes, provocando pulsos adicionais na saída.



**Figura 17 - Circuito do Gate “OU” de 4 entradas com demora reduzida.**

Na figura 17, se  $B = \emptyset$ ,  $C = \emptyset$  e os pulsos negativos de 1Hz são aplicados aos pinos 1 e 5, a saída, pino 11, apresentaria a forma de onda da figura 17. Isto porque, neste caso, os atrasos seriam de um único bloco lógico e aplicado ao circuito de saída em instantes coincidentes.

Disto conclui-se que apesar dos circuitos das figuras 16 e 17, terem expressões lógicas equivalentes, o circuito da figura 17 introduz menor demora de propagação e, consequentemente, poderá operar em maiores frequências.

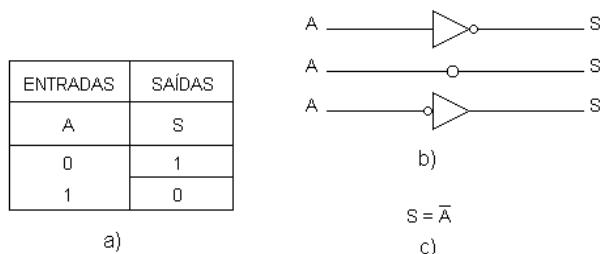
### 1.2.7 EXERCÍCIOS

- a)** Em um bloco “OU” de 4 entradas A, B, C e D, para  $A = 1$ ,  $B = \emptyset$ ,  $C = \emptyset$  e  $D = \emptyset$ , a saída será:
1. Alta (1) ( )
  2. Baixa ( $\emptyset$ ) ( )
  3. Indeterminada ( )
- b)** Se temos disponível gates “OU” de 2 entradas, quantos gates são necessários para se executar uma função OU de 8 entradas?
1. 1 ( )
  2. 2 ( )
  3. 5 ( )
  4. 7 ( )
  5. 9 ( )
  6. 10 ( )
  7. Nenhuma das anteriores ( )
- c)** Tem-se disponível um gate “OU” de 5 entradas e deseja-se executar a expressão  $L = A + B$ . O que deve ser feito com as outras entradas?
1. Ligar ao potencial 1 ( )
  2. Ligar ao potencial  $\emptyset 1$  ( )
  3. Deixar flutuando1 ( )
  4. Ligar a uma entrada usada1 ( )
- d)** Se a saída do gate “OU” é baixa ( $\emptyset$ ) isto indica que:
1. Todas as entradas são altas ( )
  2. Uma entrada é alta (1) ( )
  3. Uma entrada é baixa ( $\emptyset$ ) ( )
  4. Todas as entradas são baixas ( $\emptyset$ ) ( )

### 1.3 FUNÇÃO LÓGICA GATE “NÃO” (INVERSOR) (NOT)

#### 1.3.1 RESUMO

O gate “**NÃO**” providencia na saída um nível lógico que é o inverso do nível lógico da entrada. A tabela da figura 23a. ilustra esta operação.



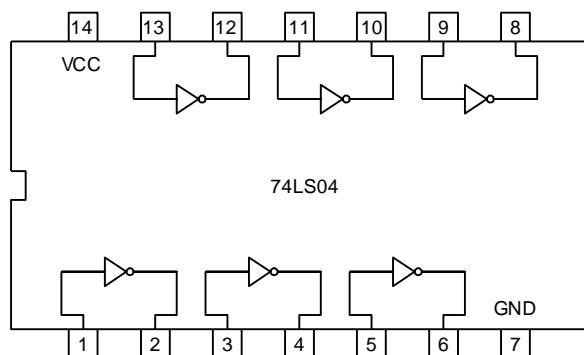
**Figura 18 - Gate “NÃO” - a) Tabela Verdade - b) Símbolos Lógicos - c) Expressão Booleana.**

#### 1.3.2 OBJETIVO

Neste ponto será examinada a tabela verdade do gate “**NÃO**” e algumas aplicações do gate “**NÃO**” como BUFFER e gerador de atraso.

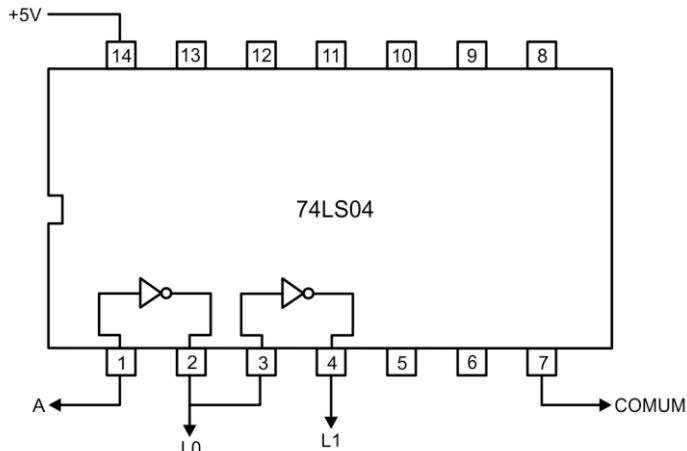
#### 1.3.3 EXPERIÊNCIA 6

- a) Material utilizado: 1 x CI 74LS04
- b) Layout do CI 74LS04



**Figura 19 - Layout do CI 74LS04.**

- c) Montar o circuito da figura 20.



**Figura 20 - Circuito com o gate NÃO.**

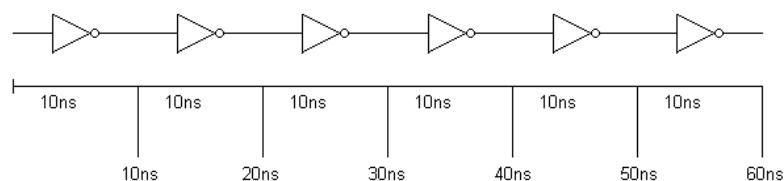
- d) Completar a tabela 6.

ENTRADAS	SAÍDAS	
A	$L\bar{0} = \bar{A}$	$L\bar{0} = \bar{\bar{A}} = a$
$\emptyset$		
1		

**Tabela 6 - Tabela Verdade do Gate NÃO (inversor).**

- e) Demora de Propagação

Os gates “NÃO” podem ser usados para introduzir demora de propagação em uma determinada linha. Usados em cascata, como mostra a figura 21, cada gate “NÃO” introduzem um atraso típico de 10ns, para TTL padrão.

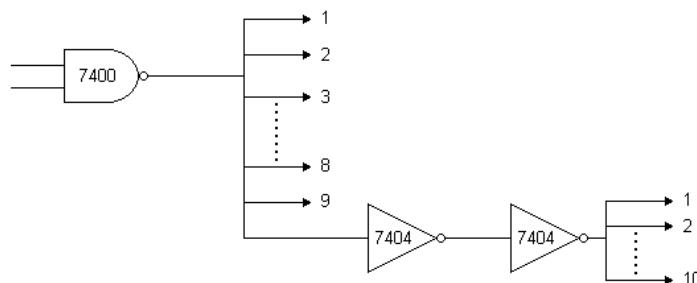


**Figura 21- Gates NÃO funcionando como Delay (atraso).**

**f) Fan-Out (Capacidade de Cargas)**

Um parâmetro importante dos circuitos integrados é a quantidade de outros gates do mesmo tipo, ou cargas, que a saída de um determinado gate poderá alimentar. Esta característica é chamada “FAN-OUT” e nos circuitos TTL tem valor típico de 10.

Se o circuito exige que mais de 10 cargas devam ser controladas por uma determinada saída, podemos usar gates “NÃO” como BUFFER, de maneira a aumentar este número. A figura 22 ilustra este fato.



**Figura 22 - Uso de Gates NÃO como BUFFER.**

O circuito original 74LS00 alimenta diretamente apenas 10 entradas, porém neste caso faz o controle de 19, podendo ser expandido para 100.

### 1.3.4 EXERCÍCIOS

a) O gate “**NÃO**” pode ser usado para inverter dois sinais ao mesmo tempo.

1. Falso
2. Verdadeiro

b) Os gates “**NÃO**” podem ser usados como elemento de demora.

1. Falso
2. Verdadeiro

c) Os gates “**NÃO**” podem ser usados como BUFFER.

1. Falso
2. Verdadeiro

d) Um número par de gates “**NÃO**” ligados em série:

1. Inverte o sinal de entrada
2. Mantém o mesmo sinal de entrada

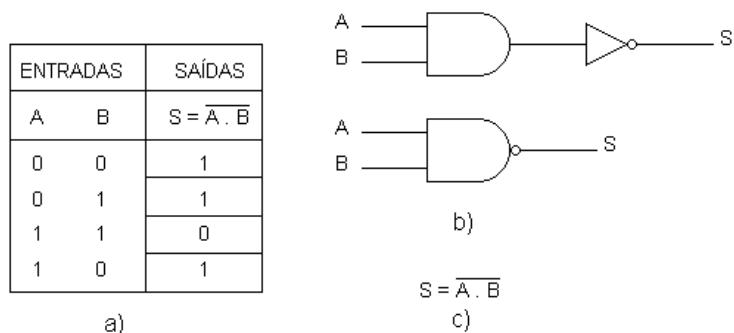
## 1.4 FUNÇÃO LÓGICA GATE “NÃO-E” (NAND)

### 1.4.1 RESUMO

A função “**NÃO-E**” pode ser definida como:

***“Produz uma saída Ø somente quando todas as entradas são 1.”***

Em suma, o gate “**NÃO-E**” é o complemento do gate “**E**”. A figura 23 mostra a tabela verdade, o símbolo e a expressão lógica do gate “**NÃO-E**”.



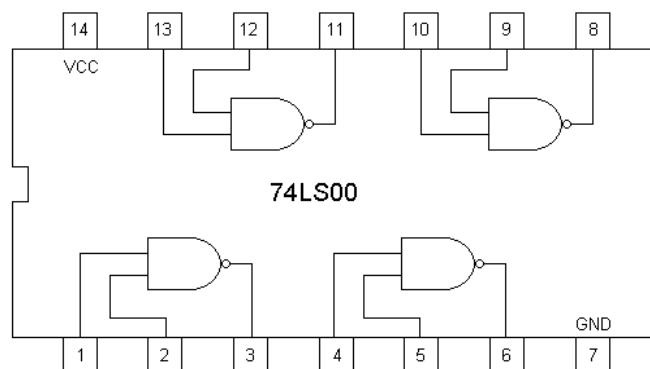
**Figura 23 - Gate “NÃO-E” - a) Tabela Verdade - b) Símbolo - c) Expressão Booleana.**

### 1.4.2 OBJETIVO

Será examinada a tabela verdade do gate “**NÃO-E**”, a maneira de montar o circuito NÃO-E com múltiplas entradas e a implementação de qualquer expressão Booleana com apenas o uso de Gates “**NÃO-E**” (sem a necessidade do uso de outros gates).

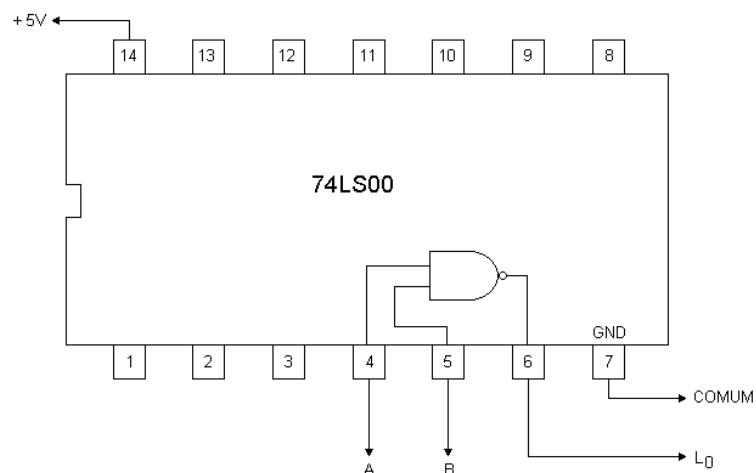
### 1.4.3 EXPERIÊNCIA 7

- a) Material utilizado: 1 x CI 74LS00
- b) Layout do CI 74LS00



**Figura 24 - Lay-out do CI 74LS00.**

c) A figura 25 mostra o circuito a ser utilizado:



**Figura 25 - Circuito com um Gate “NÃO-E”.**

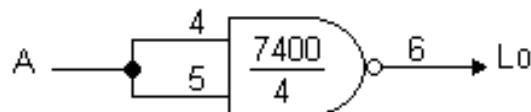
d) Completar a tabela 7.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = \overline{AB}$
∅	∅	
∅	1	
1	1	
1	∅	

**Tabela 7 - Tabela Verdade do Gate “NÃO-E”.**

#### 1.4.4 EXPERIÊNCIA 8

- a) Material utilizado: 1 x CI 74LS00
- b) Montar o circuito da figura 26, não se esquecendo de ligar o pino 14 ao 5V e o pino 7 ao comum.



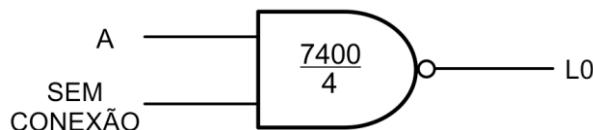
**Figura 26 - Gate “NÃO-E” funcionando como inveror.**

- c) Completar a tabela 8.

ENTRADAS	SAÍDAS
A	$L\emptyset = \bar{A}$
$\emptyset$	
1	

**Tabela 8 - Tabela Verdade do gate “NÃO-E” funcionando como inveror.**

- d) Montar o circuito da figura 27



**Figura 27 – Gate NÃO-E funcionando como inveror.**

- e) Completar a tabela 9

ENTRADAS	SAÍDAS
A	$L\emptyset = \bar{A}$
$\emptyset$	
1	

**Tabela 9 – Tabela Verdade do gate NÃO-E funcionando como inveror.**

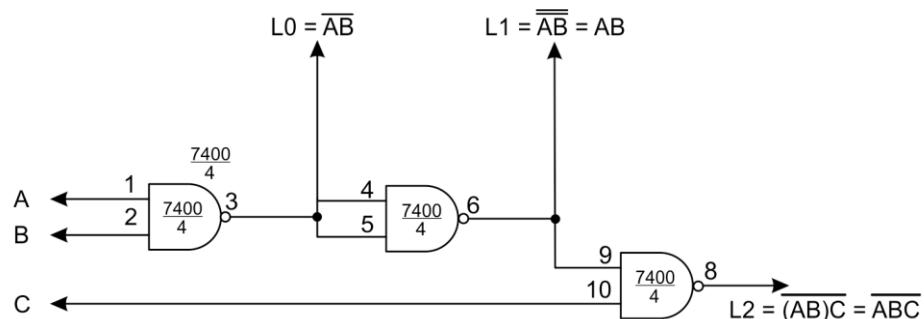
## **CONCLUSÃO:**

O gate “NÃO-E” quando tem as entradas ligadas no mesmo ponto, ou quando é usada apenas uma das entradas deixando as outras flutuando funciona como um gate “NÃO”. No primeiro caso a entrada do circuito equivale a 1 unidade de carga, quando em nível lógico baixo e 2 unidades de carga, quando em nível lógico alto.

No segundo caso, será sempre uma unidade de carga, porém, para altas frequências o pino aberto poderá operar com uma antena, fornecendo ruídos para o circuito. Neste caso é melhor que o pino seja conectado a nível lógico alto, ao invés de deixá-lo aberto.

### **1.4.5 EXPERIÊNCIA 9**

- Material utilizado: 1 x CI 74LS00
- Montar o circuito da figura 28.



**Figura 28 – Circuito do Gate NÃO-E de 3 entradas.**

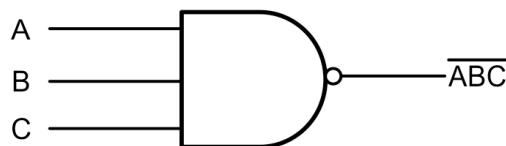
- Completar a tabela 10

ENTRADAS			SAÍDAS		
A	B	C	$L_0 = \overline{AB}$	$L_1 = AB$	$L_2 = \overline{ABC}$
0	0	0	1	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	0	1

**Tabela 10 - Tabela Verdade do gate “NÃO E” de 4 entradas.**

### **CONCLUSÃO:**

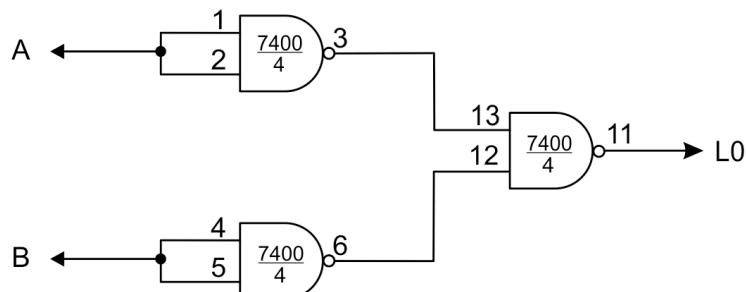
O circuito montado na figura 28 pode ser representado simbolicamente como mostra a figura 29.



**Figura 29 - Gate “NÃO-E” de 3 entradas (símbolo).**

#### **1.4.6 EXPERIÊNCIA 10**

- c) Material utilizado: 1 x CI 74LS00
- d) Montar o circuito da figura 30.



**Figura 30 – Circuito do gate “OU” montado com gates “NÃO-E”.**

- c) Completar a tabela 11

ENTRADAS		SAÍDAS
A	B	$L\bar{0} = \overline{\overline{A}\overline{B}}$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 11 – Tabela Verdade do circuito do gate “OU” montado com gates “NÃO-E”.**

### **CONCLUSÃO:**

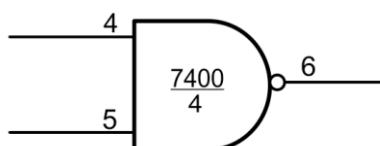
Comparar o resultado da tabela 11 com a tabela do gate “OU” e notar que são idênticas. Isto pode ser explicado através do teorema da álgebra de Boole chamado **Teorema de De Morgan**.

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

Deste modo, se observa que podendo-se implementar o gate “OU” com Gates “NÂO-E”, qualquer expressão Booleana pode ser montado usando apenas Gates “NÂO-E”.

#### **1.4.7 EXPERIÊNCIA 11**

- a) Material utilizado: 1 x CI 74LS00, ponta de prova do equipamento (PP)
- b) Montar o circuito da figura 31.



**Figura 31 – Circuito para teste de Nível Lógico de Entrada.**

- c) Completar a tabela 12, colocando a ponta de prova (PP) nos pinos assinalados e anotando a indicação lógica do display.

PONTA DE PROVA	INDICAÇÃO LÓGICA
Pino 4	
Pino 5	
Pino 6	

**Tabela 12 – Teste de Níveis Lógicos de um gate “NÂO-E”.**

### **CONCLUSÃO:**

Observar que nas entradas desconectada a ponta de prova (PP) indicou nível lógico falso, isto é a tensão deve estar entre 0,7V e 2,1V. Porém na saída à ponta de prova indicou nível lógico “0”. Disto conclui-se que as entradas abertas foram interpretadas pelo gate como nível “1”.

A série 74 foi construída de maneira que as entradas desconectadas sejam entendidas como nível lógico “1”, embora a tensão nelas se apresente em torno de 1,4V, que é um nível lógico falso, ou seja, está na região de transição lógica.

**1.4.8 EXERCÍCIOS**

- a) Em um *gate “NÃO-E”* de 3 entradas, duas entradas são conectadas ao +5v e a terceira é ligada a chave A, a saída será:

1. A
2.  $\bar{A}$
3. 1
4.  $\emptyset$

<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

- b) Com *gates “NÃO-E”* pode-se executar qualquer expressão Booleana.

1. Certo
2. Errado

<input type="checkbox"/>
<input type="checkbox"/>

## 1.5 FUNÇÃO LÓGICA GATE “NÃO-OU” (NOR)

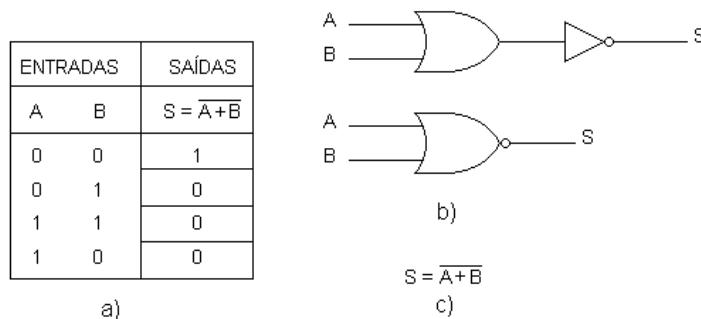
### 1.5.1 RESUMO

A função “NÃO-E” pode ser definida como:

***"Produz uma saída Ø se uma ou mais entradas forem 1."***

Isto que dizer que o gate “NÃO-OU” é complemento do gate “OU”.

A figura 32 mostra a tabela verdade, o símbolo e a expressão Booleana do gate “NÃO-OU”.



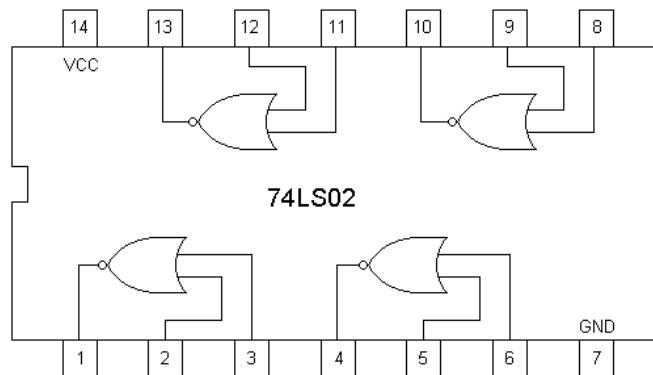
**Figura 32 - Gate “NÃO-OU” - a) Tabela Verdade - b) Símbolo Lógico - c) Expressão Booleana.**

### 1.5.2 OBJETIVO

Será examinada a tabela verdade do gate “NÃO-OU”, seu uso como inversor e implementação de gates “NÃO-OU” com múltiplas entradas, execução de qualquer expressão Booleana usando Gates “NÃO-OU”, ponto “E” (lógica por fios).

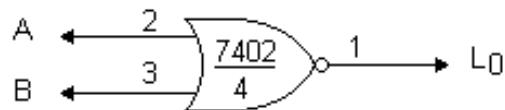
### 1.5.3 EXPERIÊNCIA 12

- a) Material utilizado: 1 x CI 74LS02
- b) Layout do CI 74LS02



**Figura 33 - Layout do CI 74LS02.**

- c) Montar o circuito da figura 34, não se esquecendo de ligar 14 ao +5V e o pino 7 ao comum.



**Figura 34 - Circuito com o gate “NÃO-OU”.**

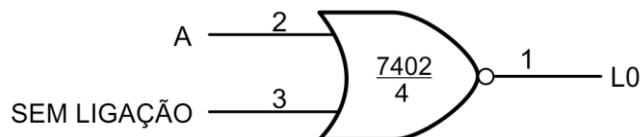
- d) Completar a tabela 13.

ENTRADAS		SAÍDAS
A	B	$L\bar{0} = \overline{A+B}$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Figura 13 - Tabela Verdade do gate NÃO-OU.**

### 1.5.4 EXPERIÊNCIA 13

- a) Material utilizado: 1 x CI 74LS02
- b) Montar o circuito da figura 35.



**Figura 35 - Circuito com o gate “NÃO-OU”.**

- c) Completar a tabela 14.

ENTRADAS	SAÍDAS
A	LØ
Ø	
1	

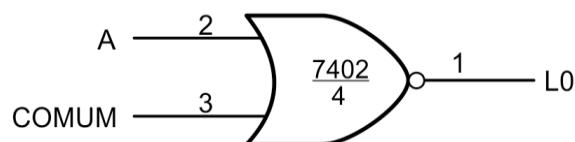
**Tabela 14 - Tabela Verdade do gate “OU” com uma entrada flutuando.**

### CONCLUSÃO:

Da tabela da figura 14, nota-se que o circuito não executa função lógica (L0 sempre Ø). Isto é devido ao pino 3 estar flutuando, o que equivale a introdução de um nível “1” no gate, produzindo assim sempre uma saída “Ø”.

### 1.5.5 EXPERIÊNCIA 14

- a) Material utilizado: 1 x CI 74LS02
- b) Montar o circuito da figura 36.



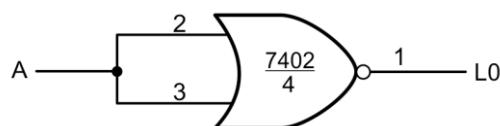
**Figura 36 - Circuito com o gate “NÃO-OU” funcionando como Inversor.**

c) Completar a tabela 15.

ENTRADAS		SAÍDAS
A		LØ
Ø		
1		

**Tabela 15 - Tabela Verdade do gate “OU” com uma entrada ligada ao comum.**

d) Montar o circuito da figura 37.



**Figura 37 – Circuito com gate “NÃO-OU” funcionando como inverSOR.**

e) Completar a tabela 16.

ENTRADAS		SAÍDAS
A		LØ
Ø		
1		

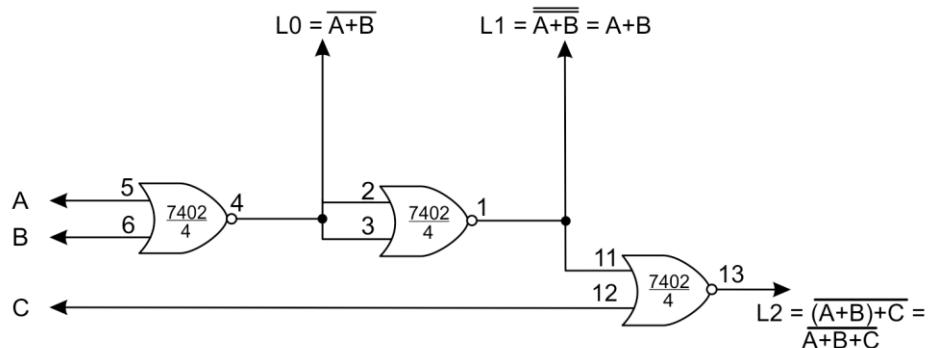
**Tabela 16 - Tabela Verdade do gate “NÃO-OU” com uma entrada ligada ao mesmo ponto.**

### **CONCLUSÃO:**

Nos dois circuitos apresentados o gate “NÃO-OU” está funcionando como inverSOR.

### 1.5.6 EXPERIÊNCIA 15

- a) Material utilizado: 1 x CI 74LS02
- b) Montar o circuito da figura 38.



**Figura 38 – Gate “NÃO-OU” de 3 entradas.**

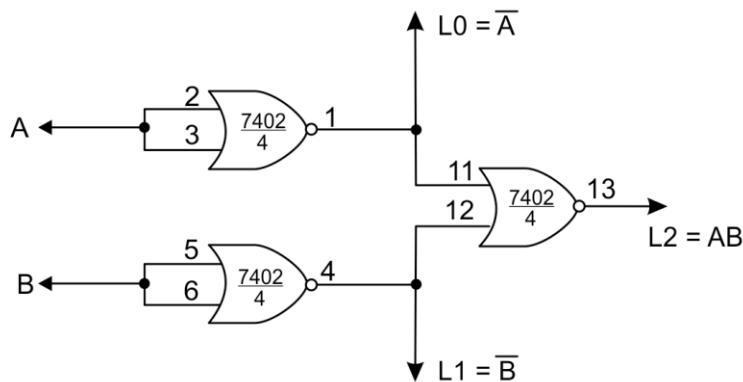
- c) Completar a tabela 17

ENTRADAS			SAÍDAS		
A	B	C	$L_0 = \overline{A+B}$	$L_1 = A+B$	$L_2 = \overline{A+B+C} = \overline{\overline{A+B}+C}$
Ø	Ø	Ø			
Ø	Ø	1			
Ø	1	Ø			
Ø	1	1			
1	Ø	Ø			
1	Ø	1			
1	1	Ø			
1	1	1			

**Tabela 17 - Tabela Verdade do gate “NÃO OU” de 3 entradas.**

### 1.5.7 EXPERIÊNCIA 16

- a) Material utilizado: 1 x CI 74LS02.
- b) Montar o circuito da figura 39.



**Figura 39 – Gate “E” implementado com Gates “NÃO OU”.**

- c) Completar a tabela 18.

ENTRADAS		SAÍDAS		
A	B	L0 = $\bar{A}$	L1 = $\bar{B}$	L2 = AB
Ø	Ø			
Ø	1			
1	Ø			
1	1			

**Figura 18 - Tabela Verdade do circuito da figura 39.**

### CONCLUSÃO:

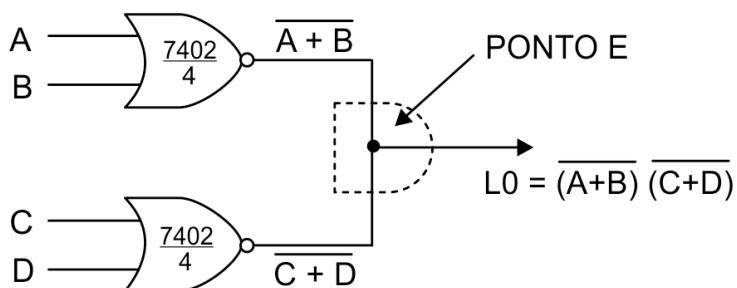
O circuito apresentado representa a função “E”, deste modo podemos deduzir que com o gate “NÃO-OU” é possível montar qualquer expressão Booleana.

O processo de se conseguir isto é através do Teorema de De Morgan:

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

### 1.5.8 EXPERIÊNCIA 17

- a) Material utilizado: 1 x CI 74LS02.
- b) Montar o circuito da figura 40.



**Figura 40 – Circuito que executa lógica por fios.**

#### Observação:

Não é possível visualizar somente  $A+B$  e nem  $C+D$  sem que se desfaça o ponto E.

- c) Completar teoricamente a tabela 19 e comparar com o resultado prático.

ENTRADAS				SAÍDAS
A	B	C	D	$L_0 = \overline{(A+B)} \overline{(C+D)}$
1	1	1	1	
Ø	Ø	1	1	
Ø	Ø	Ø	Ø	
1	1	Ø	Ø	

**Figura 19 - Tabela Verdade do circuito com lógica por fios.**

Como os led's  $L_0$ ,  $L_1$  e  $L_2$  estão conectados ao mesmo ponto, prevalece a lógica "E" por fios, ou seja, o valor de  $L_2$ .

#### CONCLUSÃO:

A lógica por fios, ou ponto "E", é uma condição onde as saídas de dois ou mais Gates são ligadas juntas.

Neste circuito, as saídas de todos os Gates devem ter nível "1" para produzir uma saída "1" no ponto "E". Se uma das saídas dos Gates for "0" esta saída manterá o ponto "E" um nível "0", independente do valor de outra saída.

O circuito montado, usando lógica por fios, economiza um gate "E", porém esta configuração não é recomendada por circuitos TTL standard, devido ao grande consumo de corrente. Algumas famílias lógicas podem produzir o ponto, porém TTL produz o ponto "E", e pode ser ligadas no máximo duas saídas de gates no mesmo ponto.

### 1.5.9 EXERCÍCIOS

- a) Se uma entrada de um gate “**NÃO-OU**” é 1 e as outras são desconhecidas, podendo ser 1 ou  $\emptyset$ , qual será a saída:

1. 1
2.  $\emptyset$
3. Indeterminada

- b) Para implementar a função  $S = A + B + C + D$ , usando gates “**NÃO-OU**” de duas entradas e sem o uso da lógica por fios, quantos gates são necessários?

1. 2
2. 4
3. 6
4. 8

- c) A Lógica por fios

1. É uma boa prática em TTL
2. Reduz o número de gates necessários
3. Pode ser usadas com várias saída de gates ligadas juntas

- d) Com gates “**NÃO-OU**” pode-se executar qualquer expressão Booleana.

1. Errado
2. Certo

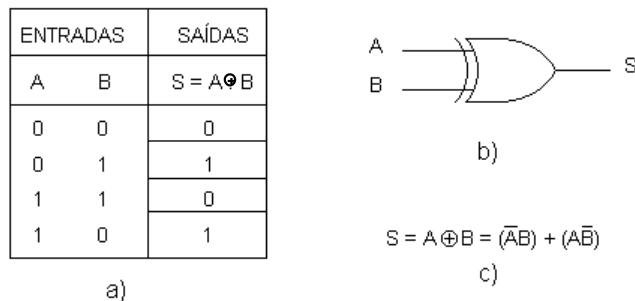
## 1.6 FUNÇÃO LÓGICA GATE “OU-EXCLUSIVO” (XOR)

### 1.6.1 RESUMO

O gate “OU-EXCLUSIVO” é um gate “OU” modificado, no qual uma entrada “1” produz uma saída “1”, porém se mais de uma entrada for “1” a saída será “0”, e também se todas as entradas forem “0” a saída será “0”. Deste modo podemos definir a função “OU-EXCLUSIVO” como:

**“Produz uma saída “1” somente se as entradas forem diferentes.”**

A figura 60 apresenta uma tabela verdade, símbolo e expressão Booleana do gate OU-EXCLUSIVO.



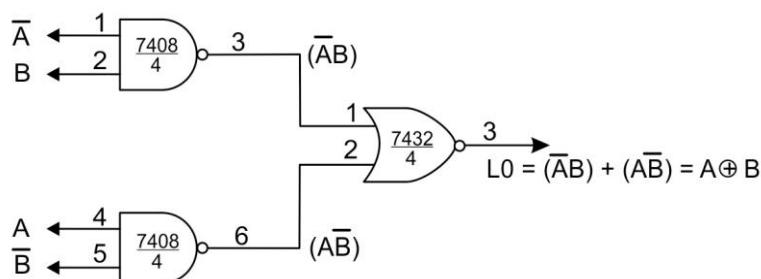
**Figura 41 - Gate “OU-EXCLUSIVO” a) Tabela Verdade - b) Símbolo - c) Expressão Booleana**

### 1.6.2 OBJETIVO

Será examinada a implementação da função “OU-EXCLUSIVO” com Gates “E”, “OU”, “NÃO-E”, “NÃO-OU” e com Gates “OU-EXCLUSIVO” disponível em Cl.

### 1.6.3 EXPERIÊNCIA 18

- a) Material utilizado: 1 x Cl 74LS08 e 1 x Cl 74LS32.
- b) Montar o circuito da figura 42, não se esquecendo de alimentar os Cl's (+5V e comum).



**Figura 42 - Gate para gerar a função “OU-EXCLUSIVO”.**

- c) Completar as tabela 20.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = A \oplus B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 20 - Tabela Verdade da função “OU-EXCLUSIVO”.**

### **CONCLUSÃO:**

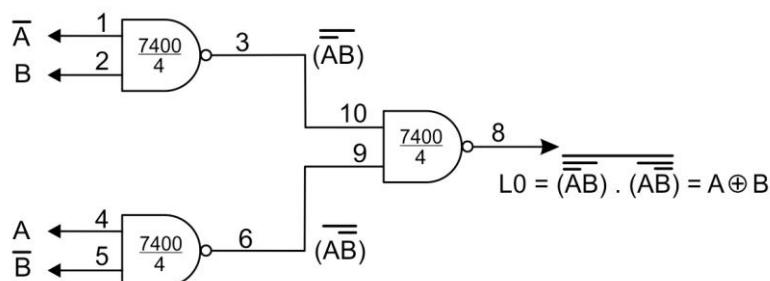
Notar que a função “OU-EXCLUSIVO” é montada a partir da expressão Booleana.

$$A \oplus B = \overline{AB} + A\overline{B}$$

Usando assim dois Gates “E” e um gate “OU”

#### **1.6.4 EXPERIÊNCIA 19**

- a) Material utilizado: 1 x CI 74LS00  
 b) Montar o circuito da figura 43, não se esquecendo de alimentar os CI (+5V e comum).



**Figura 43 – Função “OU-EXCLUSIVO” montada com gates “NÃO-E”.**

- c) Completar as tabelas 21.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = A \oplus B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 21 - Tabela Verdade da função “OU-EXCLUSIVO”, montada com gates “NÃO-E”.**

### **CONCLUSÃO:**

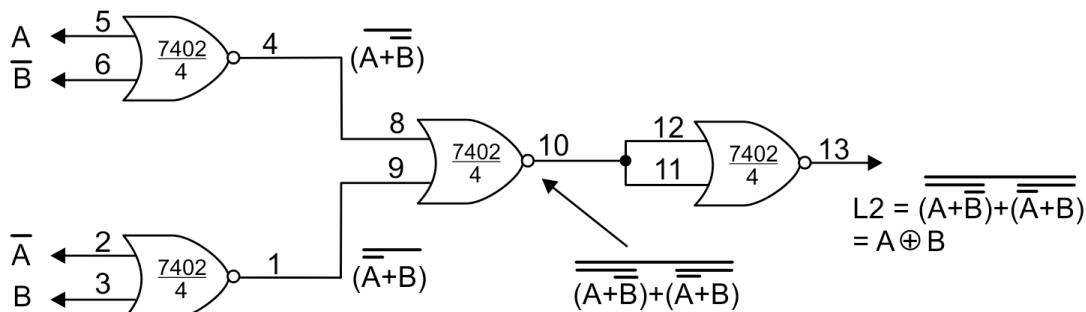
A função “OU-EXCLUSIVO” implementando com Gates “NÃO-E” é conseguida com o uso do teorema de De Morgan como se segue:

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

Comparando os resultados das tabelas Verdades vemos a validade da aplicação do teorema.

### **1.6.5 EXPERIÊNCIA 20**

- a) Material utilizado: 1 x CI 74LS02  
 b) Montar o circuito da figura 44.



**Figura 44 – Função “OU-EXCLUSIVO”, montada com gates “NÃO-OU”.**

- c) Completar as tabelas 22.

ENTRADAS		SAÍDAS
A	B	$L\bar{0} = A \oplus B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 22 - Tabela Verdade da função “OU-EXCLUSIVO”, montada com gates “NÃO-OU”.**

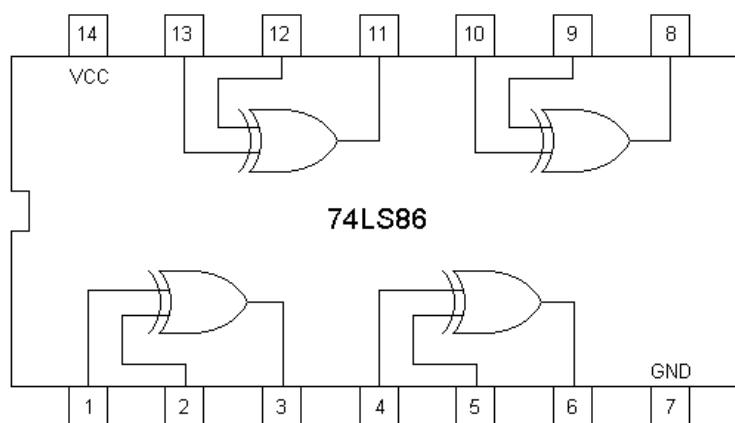
### CONCLUSÃO:

A função “OU-EXCLUSIVO” montada com gates ‘NÃO-OU’, é conseguida com a aplicação do teorema de De Morgan, cujos resultados podem ser conferidos comparando as tabelas verdadeiras.

$$\begin{aligned}
 A \oplus B &= \overline{\overline{A}B} + \overline{A}\overline{B} = \overline{\overline{(AB)}} + \overline{\overline{(AB)}} = \overline{\overline{\overline{(A+B)}}} + \overline{\overline{\overline{(A+B)}}} \\
 A \oplus B &= \overline{\overline{(A+B)}} + \overline{\overline{(A+B)}}
 \end{aligned}$$

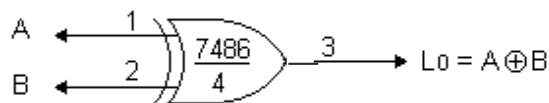
### 1.6.6 EXPERIÊNCIA 21

- a) Material utilizado: 1 x CI 74LS86  
 b) Layout do CI 74LS86.



**Figura 45 - Lay-out do CI 74LS86.**

- c) Mostrar o circuito da figura 46.



- d) Completar as tabelas 23 e 24.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = A \oplus B$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 23 - Tabela Verdade do gate “OU-EXCLUSIVO”.**

ENTRADAS		SAÍDAS
A	B	$L\emptyset$
Ø	Ø	
1	Ø	
Ø	1	
1	1	

**Tabela 24 - Tabela Verdade do gate “OU-EXCLUSIVO” funcionando como um Inversor Programável.**

- e) Observar na tabela 24 que quando  $B = \emptyset$ , a saída do gate “OU-EXCLUSIVO” apresenta o nível lógico idêntico ao da entrada A. Porém, quando  $B = 1$  a saída é o inverso de “A”. Deste modo, o gate “OU-EXCLUSIVO” pode ser usado como um inversor programável; se uma das entradas é  $\emptyset$  não há inversão, porém se for 1 há inversão do nível lógico da outra entrada.

### 1.6.7 EXERCÍCIOS

a) O gate “OU-EXCLUSIVO” produz uma saída 1 se:

1. Ambas as entradas forem  $\emptyset$
2. As entradas forem estados contrários
3. Ambas as entradas forem 1
4. Nenhuma das respostas anteriores

b) O gate “OU-EXCLUSIVO” produz uma saída  $\emptyset$  se as entradas forem as mesmas.

1. Verdade
2. Falso

c) Para usar o gate “OU-EXCLUSIVO” como inversor:

1. As entradas devem ser 1 simultaneamente
2. As entradas devem ser  $\emptyset$  simultaneamente
3. Uma entrada deve ser sempre  $\emptyset$
4. Uma entrada deve ser sempre 1

## 1.7 TEOREMA DE “DE MORGAN”

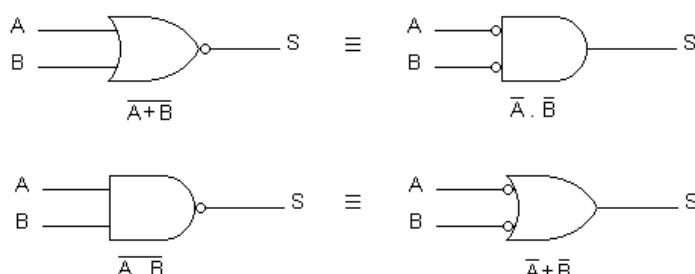
### 1.7.1 RESUMO

Basicamente, o teorema de De Morgan permite a execução de uma expressão Booleana usando determinados tipos de gates, por exemplo “NÃO-E” ou então “NÃO-OU”. O teorema funciona da seguinte maneira:

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

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

Isto está mostrado em símbolos na figura 47.

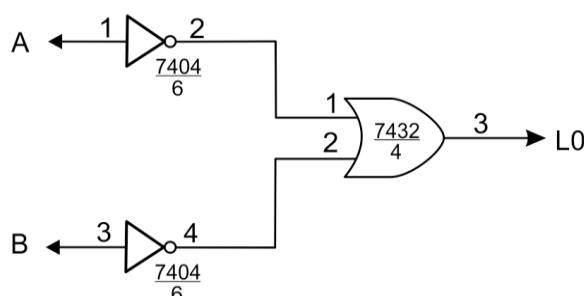


**Figura 47 - Teorema De Morgan.**

Deste modo podemos passar de um tipo de função a outra pela aplicação do Teorema de De Morgan.

### 1.7.2 EXPERIÊNCIA 22

- a) Material utilizado: 1 x CI 74LS04 e 1 x CI 74LS32.
- b) Montar o circuito da figura 48, não se esquecendo de alimentar os CI's (+5V e comum).



**Figura 48 – Implementação do Teorema de De Morgan.**

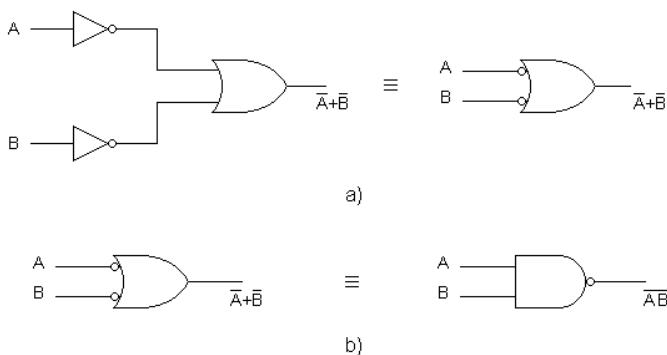
- c) Completar a tabela 25.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = \bar{A} + \bar{B}$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 25 - Tabela Verdade da expressão  $\bar{A} + \bar{B}$**

### CONCLUSÃO:

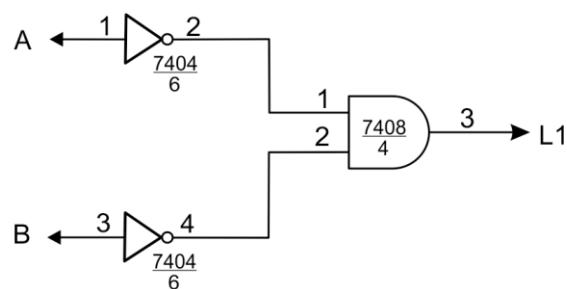
O circuito montado representa a função NÂO-E, e pode ser provado comparando o resultado da tabela verdade com a já visto no gate “NÂO-E”. Deste modo, observamos que a troca da função “OU” para “E”, foi acompanhada de inversão do sinal nas barras de entrada e saída, como mostra a figura 49 em símbolos lógicos.



**Figura 49 - a) Representação Simbólica do circuito montado - b) Identidade pelo Teorema de De Morgan.**

### 1.7.3 EXPERIÊNCIA 23

- a) Material utilizado: 1 x CI 74LS04 e 1 x CI 74LS08.
- b) Montar o circuito da figura 50, não se esquecendo de alimentar os CI's (+5V e comum).



**Figura 50 – Implementação do teorema de De Morgan.**

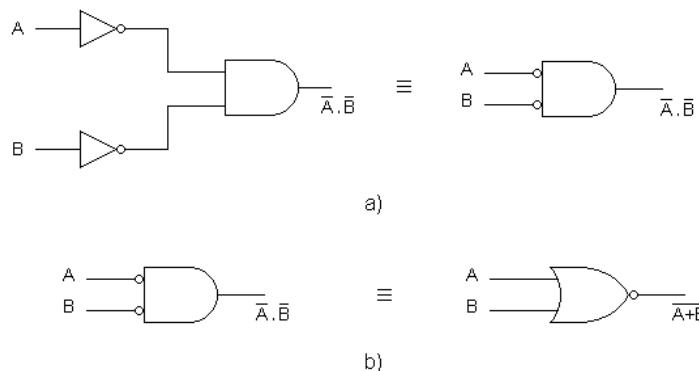
- c) Completar a tabela 26.

ENTRADAS		SAÍDAS
A	B	$L\emptyset = \bar{A} \cdot \bar{B}$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 26 - Tabela Verdade da expressão  $\bar{A} \cdot \bar{B}$ .**

### CONCLUSÃO:

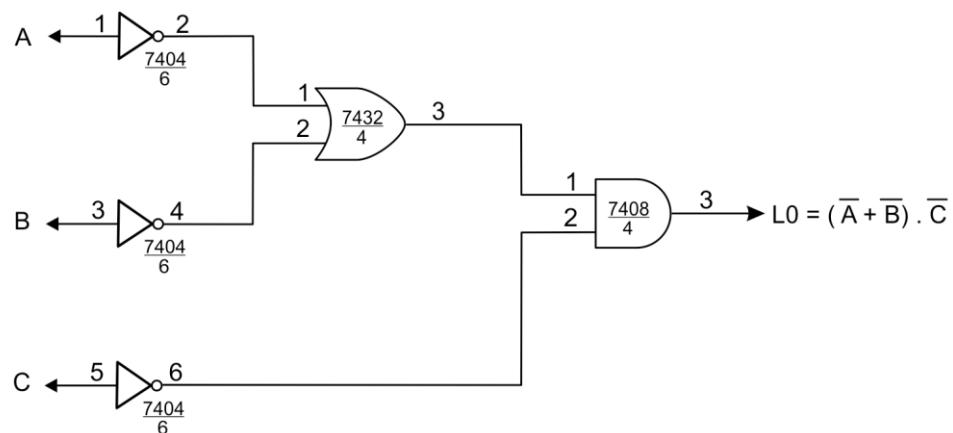
O circuito montado representa a função “NÂO-OU”, o que pode ser comprovado pelas tabelas verdades. Observamos que a troca de função “E” para “OU” foi acompanhada de uma inversão nos sinais de entrada e saída, como podemos ver em símbolos na figura 51.



**Figura 51 - a) Representação Simbólica do circuito montado - b) Identidade pelo Teorema De Morgan.**

### 1.7.4 EXPERIÊNCIA 24

- a) Material utilizado: 1 x CI 74LS04, 1 x CI 74LS08 e CI 74LS32.
- b) Montar o circuito da figura 52, não se esquecendo de alimentar os CI's (+5V e comum).



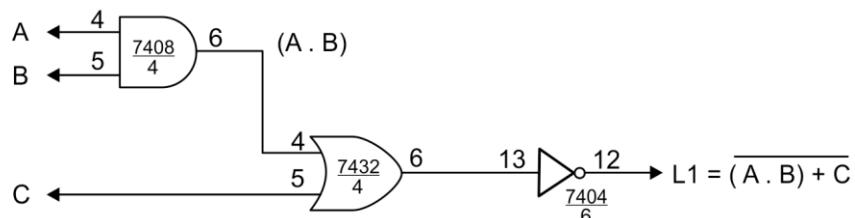
**Figura 52 – Circuito da expressão Booleana  $(\bar{A} + \bar{B}) \cdot \bar{C}$**

- c) Completar a tabela 27.

ENTRADAS			SAÍDAS
A	B	C	$L_0 = (\bar{A} + \bar{B}) \cdot \bar{C}$
Ø	Ø	Ø	
Ø	Ø	1	
Ø	1	Ø	
Ø	1	1	
1	Ø	Ø	
1	Ø	1	
1	1	Ø	
1	1	1	

**Tabela 27 - Tabela Verdade da expressão  $(\bar{A} + \bar{B}) \cdot \bar{C}$ .**

- d) Sem desmontar o circuito anterior, montar o circuito da figura 53.



**Figura 53 – circuito da expressão  $(A \cdot B) + C$**

- e) Completar a tabela 28.

ENTRADAS			SAÍDAS
A	B	C	$L1 = \overline{(A \cdot B)} + C$
Ø	Ø	Ø	
Ø	Ø	1	
Ø	1	Ø	
Ø	1	1	
1	Ø	Ø	
1	Ø	1	
1	1	Ø	
1	1	1	

**Tabela 28 - Tabela Verdade da expressão  $\overline{(A \cdot B)} + C$**

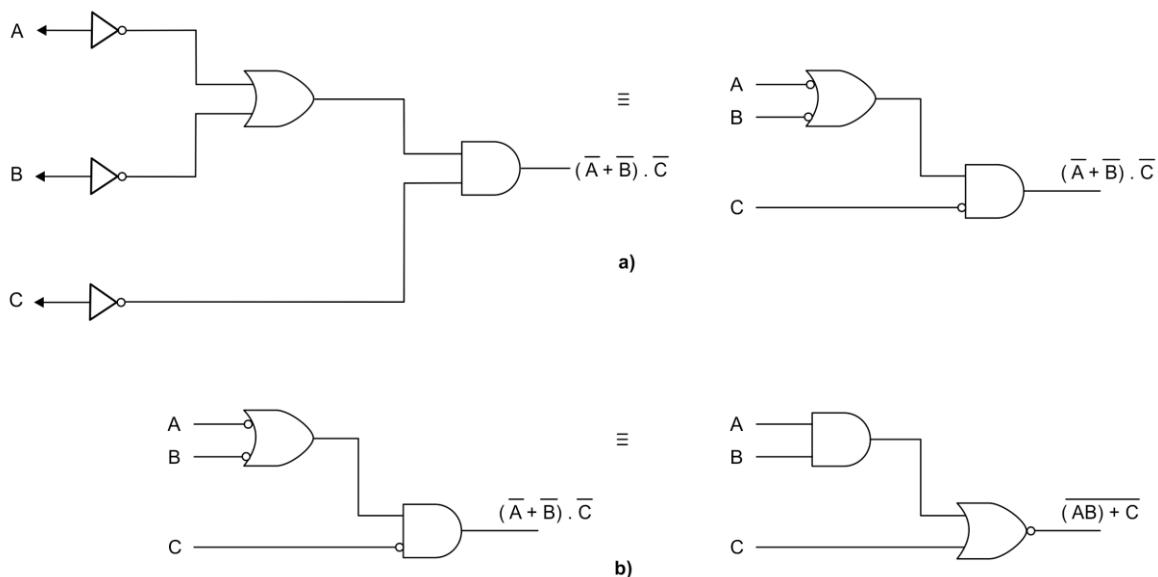
### CONCLUSÃO:

Observar que os resultados das tabelas verdades 27 e 28 são idênticos. Isto nos mostra a aplicação do teorema de **De Morgan** de maneira mais ampla. O realizado nesta experiência, em termos de expressão Booleana, foi:

$$\begin{aligned}
 (\overline{A} + \overline{B}) \cdot \overline{C} &= \overline{\overline{A} + \overline{B}} \cdot \overline{C} = \overline{\overline{A} \cdot \overline{B}} + \overline{C} = \\
 &= (\overline{A} \cdot \overline{B}) + C = \overline{(A \cdot B)} + C
 \end{aligned}$$

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

Em termos de símbolos temos a figura 54.



**Figura 54 - a) Representação simbólica do circuito da figura 78 – b) Identidade pelo Teorema de De Morgan.**

Analizando a representação simbólica dos circuitos, podem-se enunciar algumas regras práticas para que, dispondo de um circuito, se possa fazer aplicação do teorema de De Morgan sem a necessidade de recorrer a expressão Booleana:

***“Dispondo de um circuito, a aplicação do teorema de De Morgan se faz trocando-se todas as funções “E” por “OU” e “OU” por “E”, e fazendo a inversão de todas as entradas e saídas, de modo que duas inversões na mesma linha se anulam.”***

### 1.7.5 EXERCÍCIOS

**a)** A função “E” pode ser implementado com gates:

1. E
2. OU
3. NÃO-E
4. NÃO-OU
5. Nenhum dos itens anteriores

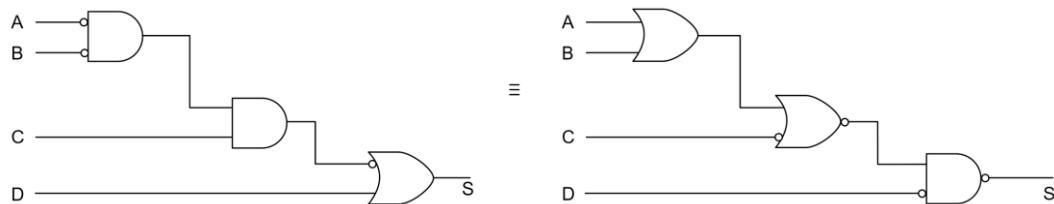
  
  
  
  


**b)** Pelo teorema de De Morgan a expressão  $S = A + B$  é semelhante a:

1.  $\overline{A} \cdot \overline{B}$
2.  $\overline{A} \cdot B$
3.  $\overline{A} + \overline{B}$


**c)** Indicar a validade do circuito da figura 55:



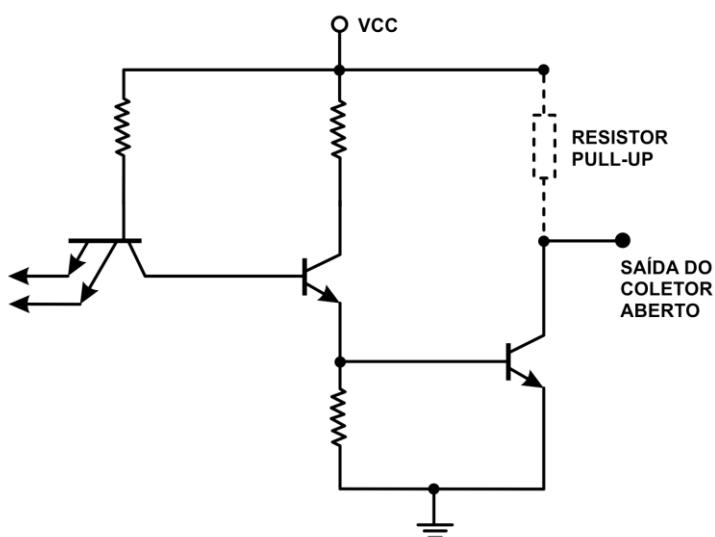
**Figura 55 – Teorema de De Morgan.**

1. Verdadeiro
2. Falso

## 1.8 GATES DE COLETOR ABERTO

### 1.8.1 RESUMO

O gate de Coletor Aberto é um CI que possui a característica especial de possuir o transistor de saída com o coletor disponível em um pino do CI, e desconectado de qualquer ligação interna, conforme mostra a figura 56.



**Figura 56 – Circuito do gate de coleto aberto.**

Para o funcionamento correto do gate há a necessidade da ligação de um resistor, denominado **Resistor Pull-up**, entre o coletor aberto e Vcc (mostrado na figura 56 em tracejado).

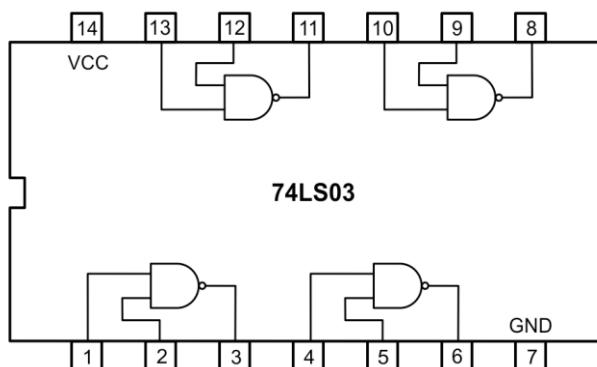
Este CI é apresentado assim a fim de permitir a implementação da Lógica por Fios (E por fios) e, deste modo, o resistor pull-up funciona como um regulador de corrente, sendo que seu valor vai depender do número de gates, que irão participar da conexão “E” por fios, e do número de cargas a serem alimentadas pelo ponto “E”.

Existem também os **Buffers e Drivers** de coletor aberto, que diferem dos gates ordinários de coletor aberto por terem maior capacidade de absorção de corrente e um **Fan-Out** mais elevado: são usados na alimentação de cargas que exigem altas correntes.

Alguns destes circuitos têm a vantagem adicional de permitir a ligação do resistor pull-up em até 30V (como o 74LS06 e o 74LS07), podendo assim fazer interface de TTL para circuitos com tensão mais elevada que a dos gates TTL comuns.

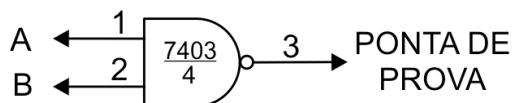
### 1.8.2 EXPERIÊNCIA 25

- a) Material utilizado: 1 x CI 74LS03, 1 resistor de 1KΩ e Ponta de Prova do equipamento.
- b) Layout do CI 74LS03.



**Figura 57 – Layout do CI 74LS03.**

- c) Montar o circuito da figura 58



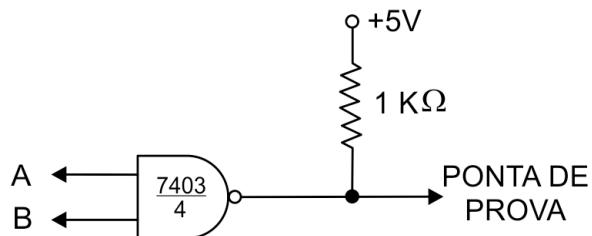
**Figura 58 – Circuito para teste do gate de coletor aberto sem resistor pull-up.**

- d) Completar a tabela 29.

ENTRADAS		SAÍDAS
A	B	Nível Lógico
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 29 – Tabela Verdade do gate de coletor aberto sem resistor pull-up.**

- e)** Montar o circuito da figura 59



**Figura 59 – Gate de coletores abertos com o resistor pull-up.**

- f)** Completar a tabela 30.

ENTRADAS		SAÍDAS
A	B	Nível Lógico
Ø	Ø	
Ø	1	
1	1	
1	Ø	

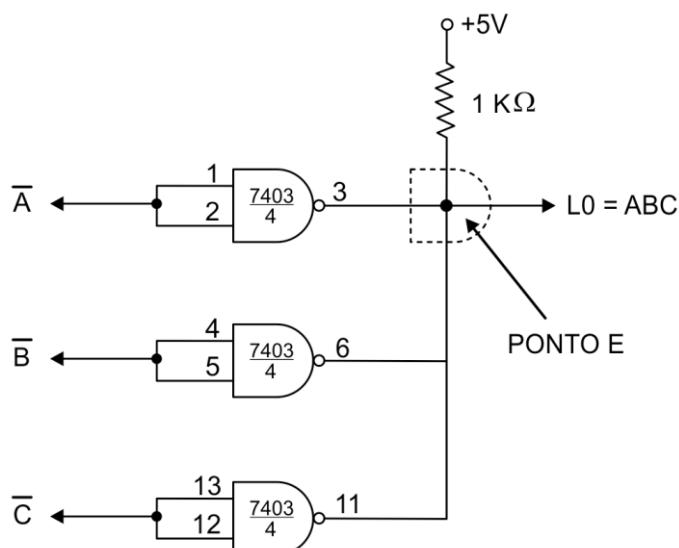
**Tabela 30 – Tabela Verdade do gate de coletores abertos com o resistor pull-up**

### CONCLUSÃO:

Observar que o funcionamento correto do gate se faz quando é feita a ligação do resistor de 1KΩ.

### 1.8.3 EXPERIÊNCIA 26

- a)** Material utilizado: 1 x CI 74LS03, 1 resistor de 1KΩ  
**b)** Montar o circuito da figura 60.



**Figura 60 – Lógica por fios usando Gates de coletor aberto.**

c) Completar a tabela 31

ENTRADAS			SAÍDAS
A	B	C	$L_0 = A \cdot B \cdot C$
Ø	Ø	Ø	
Ø	Ø	1	
Ø	1	Ø	
Ø	1	1	
1	Ø	Ø	
1	Ø	1	
1	1	Ø	
1	1	1	

**Tabela 31 – Tabela Verdade do circuito com Gates de coletor aberto executando lógica por fios.**

### CONCLUSÃO:

No circuito montado o número de Gates envolvidos pode ser aumentado sem que haja dano nos Cl's. Como regra prática, para até 7 gates de entrada com um Fan-out de 7 use-se um resistor de 1KΩ.

#### 1.8.4 EXERCÍCIOS

**a)** O gate de coleto aberto:

1. Permite a lógica por fios
2. Introduz atraso no sistema
3. É usado como Interface

**b)** O resistor pull-up serve:

1. Para quadrar a onda de saída
2. Para evitar ruído
3. Para controle de corrente

## 1.9 GATES TRI-STATE

### 1.9.1 RESUMO

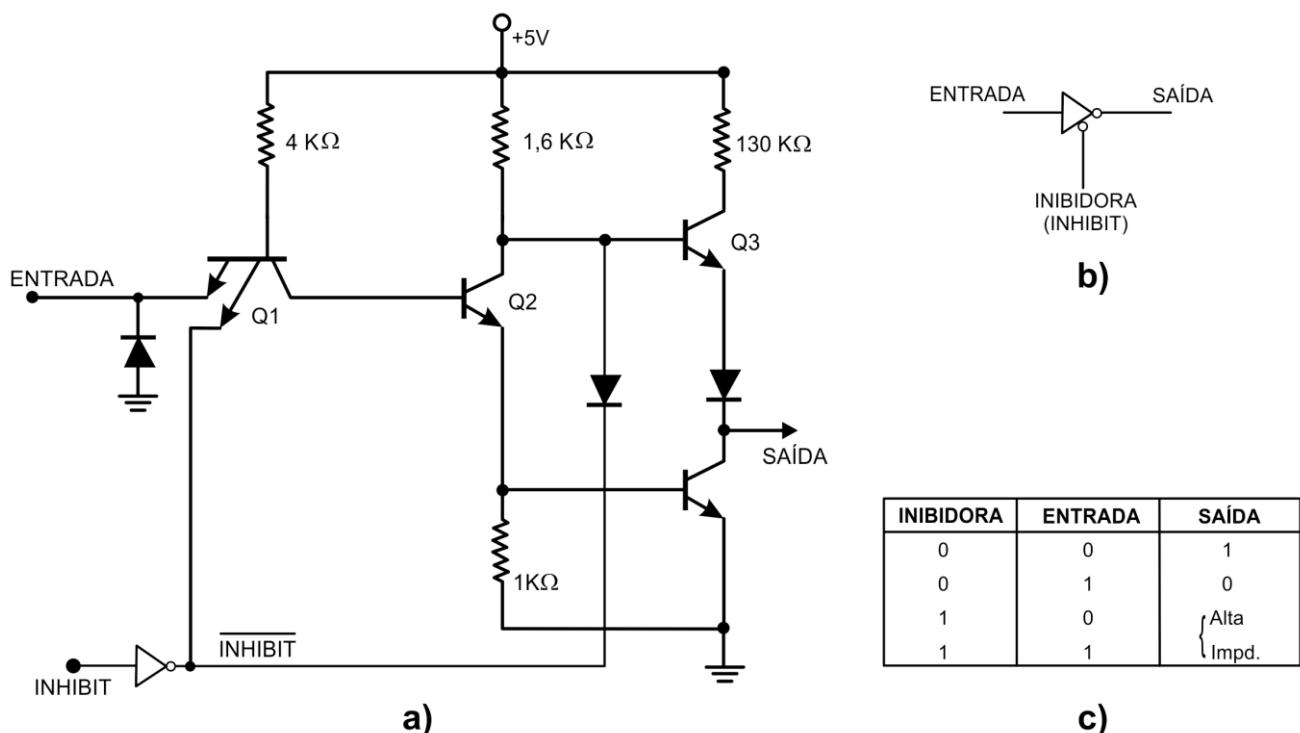
O dispositivo denominado ***Tri-State*** é um circuito que permite três estados:

1. Estado lógico “1” (baixa impedância para VCC);
2. Estado lógico “0” (baixa impedância para terra);
3. Desligado (*disabled*) (apresenta alta impedância tanto para VCC como para terra).

Destes três estados, o que se apresenta como novidade é o terceiro (desligado). Neste estado o circuito se apresenta como se estivesse desconectado do resto do sistema, isto é, não há troca de corrente com os circuitos conectados a esta saída. Assim, quando está neste estado, sua saída não influencia e nem é influenciada pelo sistema a ela conectada.

Os dispositivos *Tri-State* possuem uma entrada habilitadora / desabilitadora além das entradas e saídas normais.

Quando habilitado o gate funciona normalmente e quando desabilitado apresenta uma alta impedância de saída. A figura 61 ilustra o circuito do dispositivo *Tri-State TTL*.



**Figura 61 – a) Circuito do dispositivo *Tri-state* – b) Símbolo - c) Tabela Verdade.**

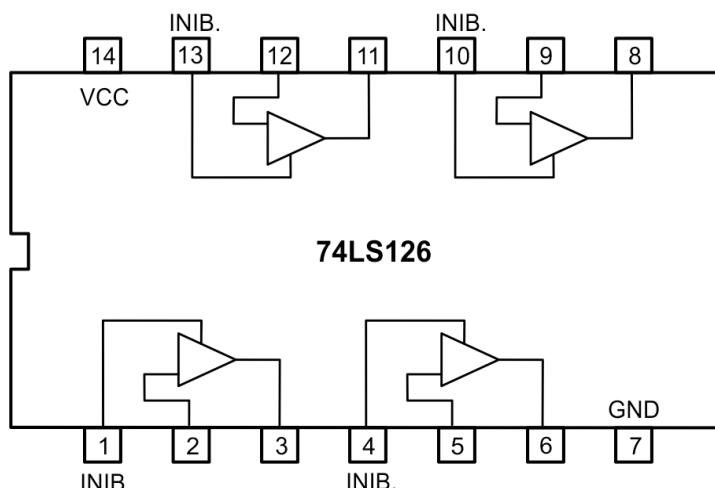
O circuito da figura 61, a) funciona da seguinte maneira: Quando a entrada **Inhibit** está em “0”, faz com que o transistor de entrada (MET) e o diodo da base de **Q3** seja polarizadas reversamente, assim o circuito funciona como um inversor normal.

Porém quando **Inhibit** esta em “1”, o fluxo de corrente na junção base-emissor de **Q1** deixa **Q2** sem corrente de base, levando-o ao corte e, consequentemente, **Q4** também. O diodo na base de **Q3** estará conduzindo e levando **Q3** ao corte.

Assim **Q3** e **Q4** estarão em corte, o que ocasiona uma saída de alta impedância tanto para VCC como para terra.

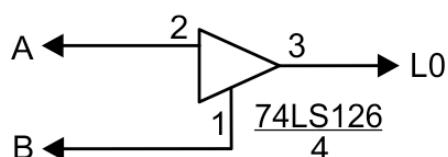
### 1.9.2 EXPERIÊNCIA 27

- a) Material utilizado: 1 x CI 74LS126
- b) Layout do CI 74LS126.



**Figura 62 – Layout do CI 74126.**

- c) Mostrar o circuito da figura 63



**Figura 63 – Gate Tri-State.**

- d) Completar a tabela 32

ENTRADAS		SAÍDAS
DADOS	INIBIDORA	
A	B	L <sub>0</sub>
Ø	Ø	
Ø	1	
1	1	
1	Ø	

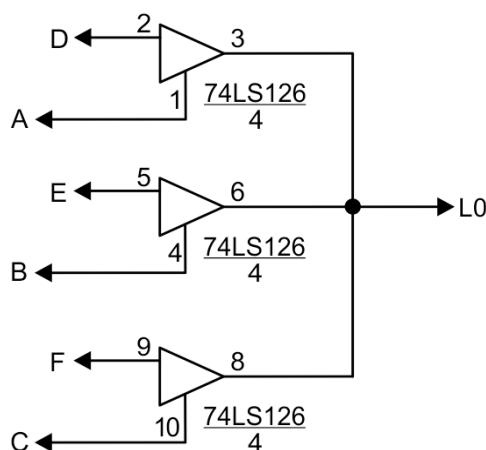
**Tabela 32 – Tabela Verdade do gate Tri-State**

## CONCLUSÃO:

Observar que quando a entrada inibidora está em “0” (B = 0) nada ocorre com a saída L<sub>0</sub>, e para a entrada inibidora em “1” (B = 1) a saída L<sub>0</sub> segue o nível colocado em A. O uso da ponta de prova, ao invés de L<sub>0</sub>, irá mostrar a operação do circuito *Tri-State* mais adequadamente. (A = aberto, quando B = 0).

### 1.9.3 EXPERIÊNCIA 28

- a) Material utilizado: 1 x CI 74LS126  
 b) Montar o circuito da figura 64.



**Figura 64 – Circuito de um barramento com Gates Tri-States.**

**Observação:**

O circuito do 74LS126 pode ser facilmente queimado. Somente uma das entradas inibidoras (ABC) pode estar no nível “1” de cada vez.

- c) Completar a tabela 33.

ENTRADAS INIBIDORAS			ENTRADAS DE DADOS			SAÍDA
A	B	C	D	E	F	L <sub>0</sub>
1	Ø	Ø	Ø	Ø	Ø	
1	Ø	Ø	1	Ø	Ø	
1	Ø	Ø	Ø	1	Ø	
1	Ø	Ø	Ø	Ø	1	
Ø	1	Ø	1	Ø	Ø	
Ø	1	Ø	Ø	1	Ø	
Ø	1	Ø	Ø	Ø	1	
Ø	Ø	1	1	Ø	Ø	
Ø	Ø	1	Ø	1	Ø	
Ø	Ø	1	Ø	Ø	1	
Ø	Ø	1	Ø	Ø	Ø	

**Tabela 33 – Tabela Verdade do circuito com gate Tri-State.**

**CONCLUSÃO:**

Observar na tabela que estando A = 1, B = 0 e C = 0, somente a informação introduzida pela chave “D” altera o estado do led L<sub>0</sub>.

Isto indica que somente o gate comandado pela chave “A” esta habilitado a transferir informação ao led, os outros estão com que desconectados do circuito.

#### 1.9.4 EXERCÍCIOS

a) O gate de coletor aberto:

1. Baixa Impedância
2. Alta Impedância
3. Saída indeterminada


b) No circuito do item 1.9.3 para  $A = 0$ ,  $B = 0$  e  $C = 0$ :

1. A entrada “D” transfere informações
2. A entrada “E” transfere informações
3. Todas as entradas transferem informações
4. Nenhuma entrada transfere informações


## **CAPÍTULO 2 – FLIP FLOPS**

### **2.1 RESUMO**

Os flip-flops são os componentes básicos de vários dispositivos, dentre os quais destacam-se as memórias semicondutoras e os contadores.

Os flip-flops podem ser classificados em dois tipos principais:

- a) Flip-Flops Síncronos ou com "clock"**
- b) Flip-Flops Assíncronos ou sem "clock"**

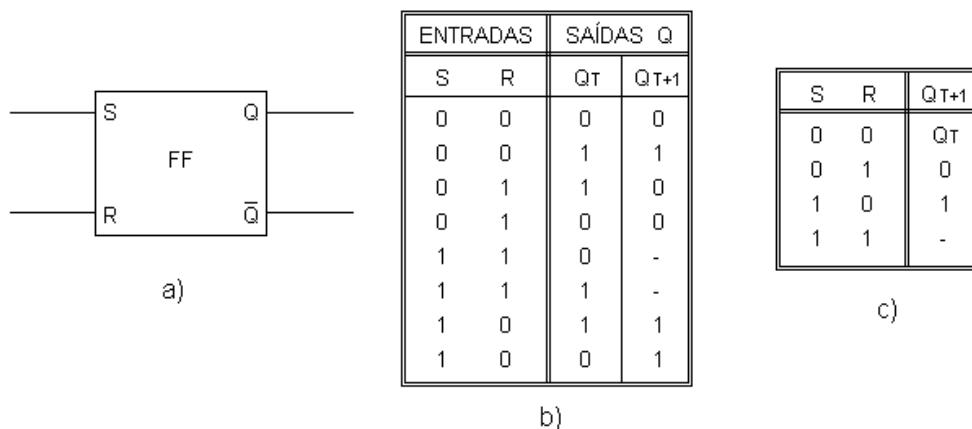
Os flip-flops síncronos exigem, para seu perfeito funcionamento, um pulso de gatilhamento também chamado "clock", além das entradas normais. Deste modo, nos flip-flops síncronos as mudanças só ocorrerão durante o pulso de sincronização, ou *clock*. Já nos flip-flops assíncronos, as transições ocorrem após terem sido completadas mudanças nas entradas normais, sem haver a necessidade de pulsos de sincronização.

De acordo com o funcionamento de suas entradas existem diversos tipos de flip-flops, sendo os mais comuns: **S-R**, **T**, **D** e **JK**, cujas características de funcionamento passamos a descrever.

### **2.2 FLIP-FLOP S-R**

Este é o mais simples dispositivo de memória, também chamado de "**Flip-Flop Set Reset**".

A figura 1 mostra o diagrama em blocos e a tabela verdade do flip-flop S-R.



**Figura 1 - Flip-Flop S-R - a) Símbolo - b) Tabela de Funcionamento - c) Tabela Resumida.**

A leitura da tabela se faz do seguinte modo:

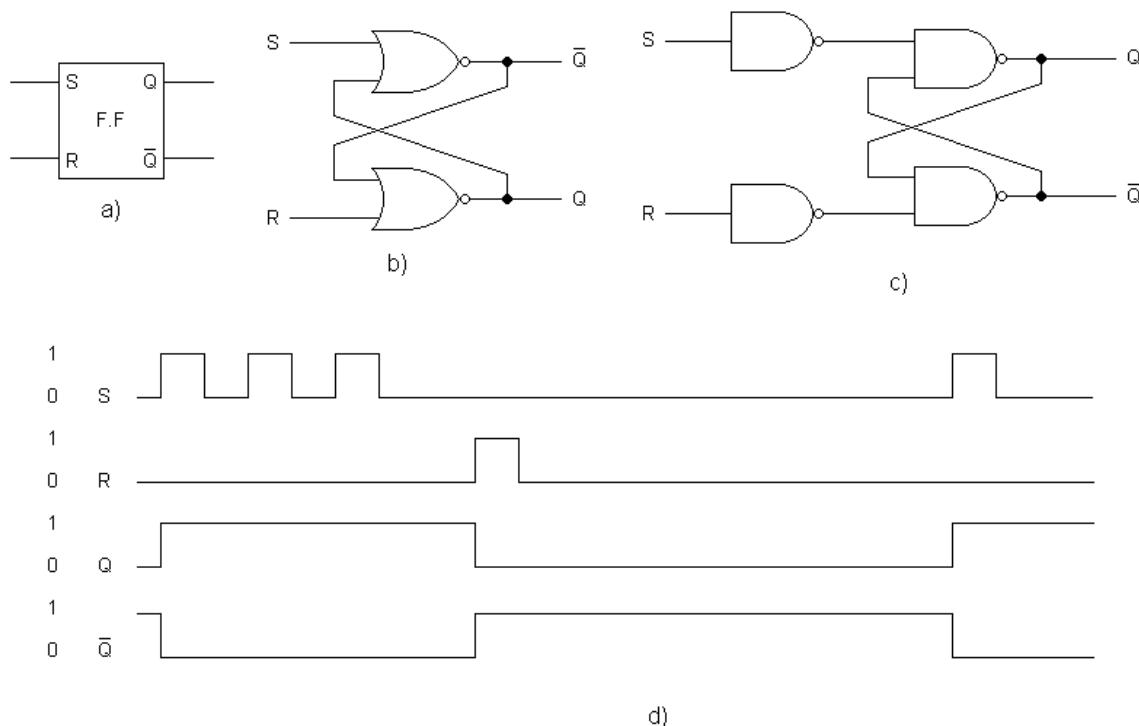
"-" significa não deve ocorrer ou não interessa.

$Q_T$  é a saída Q existente no instante (T) de aplicação das entradas S e R.

$Q_{T+1}$  é a saída Q após a aplicação das entradas S e R.

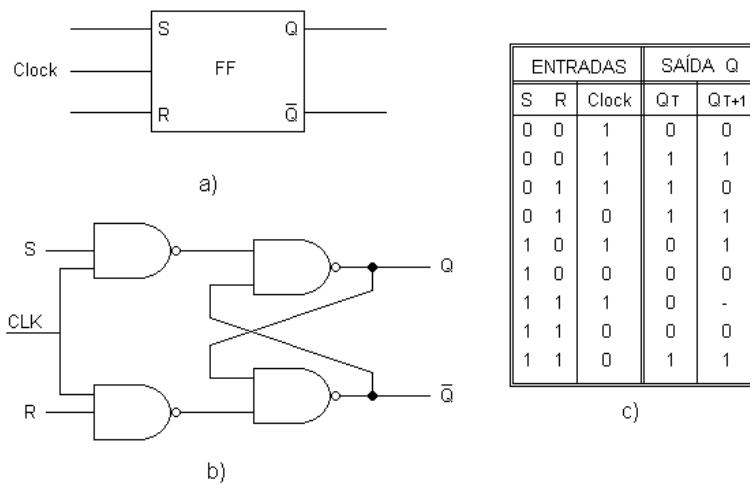
Da tabela da figura 1 pode-se observar que quando  $S = 1$  e  $R = \emptyset$  a saída Q vai para 1; quando  $S = \emptyset$  e  $R = 1$ , a saída Q vai para  $\emptyset$ ; quando  $R = S = \emptyset$ , Q mantém o estado anterior e quando  $S = R = 1$  a saída será indeterminada.

O flip-flop em sua versão mais simples é muitas vezes chamado de **LATCH**, e pode ser implementado com gates NÃO-OU ou com gates NÃO-E, como mostra a figura 2.



**Figura 2 - Flip-Flop S-R - a) Símbolo - b) Implementação com gates “NÃO-OU” - c) Implementação com gates “NÃO-E” - d) Formas de onda.**

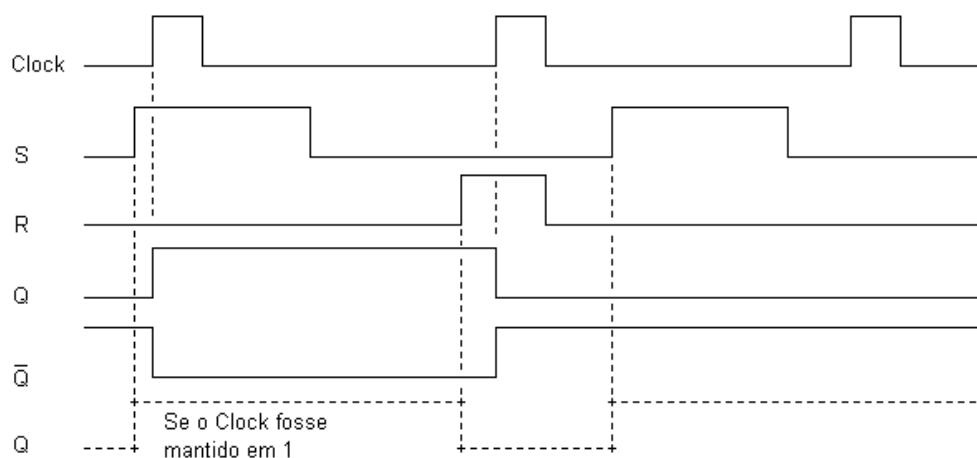
Como pode ser observado, os flip-flops vistos são assíncronos, isto é, são diretamente operados pelas entradas S e R; porém, estes dispositivos podem ser implementados com uma linha de sincronização em *clock*. Na figura 3 é mostrado este circuito.



**Figura 3 - Flip-Flop S-R Síncrono - a) Símbolo - b) Implementação com gates “NÃO-E” - c) Tabela de Funcionamento.**

Na tabela da figura 3c notar que quando o clock está em 1 o flip-flop funciona normalmente, e quando está em  $\emptyset$  não ocorre mudanças na saída.

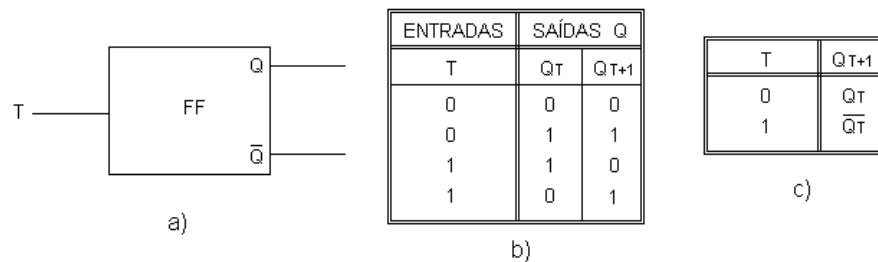
A figura 4 mostra as formas de onda de funcionamento do flip-flop S-R síncrono.



**Figura 4 - Formas de onda de funcionamento de um Flip-Flop S-R Síncrono.**

### 2.3 FLIP-FLOP T

O Flip-Flop T (*Toggle*) ou Complementar tem sua tabela de funcionamento mostrada na figura 5.

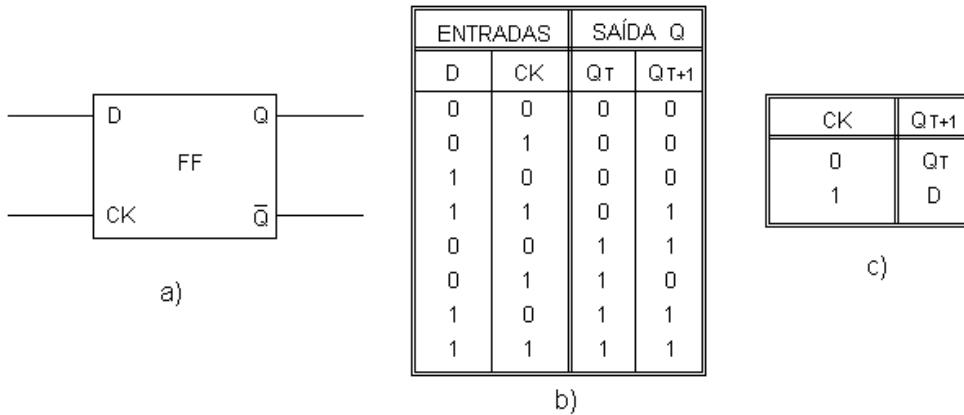


**Figura 5 - Flip-Flop T - a) Símbolo - b) Tabela de Funcionamento - c) Tabela Resumida**

O Flip-Flop T não é disponível comercialmente, podendo, no entanto ser construído a partir de outros.

### 2.4 FLIP-FLOP D

O Flip-Flop D é quase sempre gatilhável (síncrono), isto é, vem sempre associado com uma entrada de clock, às vezes denominada **ENABLE**. A figura 6 mostra a sua tabela de funcionamento.

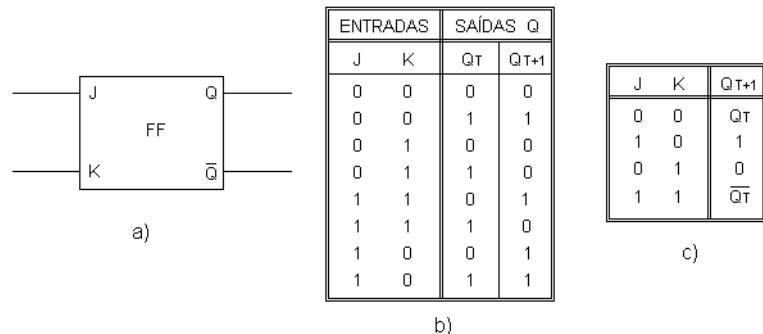


**Figura 6 - Flip-Flop D - a) Saída - b) Tabela de Funcionamento - c) Tabela Resumida.**

Da figura 6b vemos que a saída Q acompanha a entrada D, desde que o clock esteja em 1. Com o clock em  $\emptyset$  não há mudança de saída.

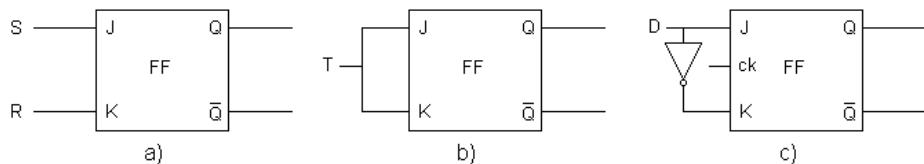
## 2.5 FLIP-FLOP JK

Este é o mais versátil dos flip-flops, não possuindo condições de entradas inválidas (como o S-R para  $S = R = 1$ ).



**Figura 7 - Flip-Flop J-K - a) Símbolo - b) Tabela de Funcionamento - c) Tabela Resumida.**

Com o flip-flop J-K podemos construir outros tipos de flip-flops, como mostra a figura 8.



**Figura 8 - Uso do Flip-Flop J-K na construção de outros - a) Construção do FF S-R - b) Construção do FF T - c) Construção do FF D.**

## 2.6 TIPOS DE GATILHAMENTO DOS FLIP-FLOPS

Quanto ao tipo de gatilhamento, os flip-flops se dividem em:

1. **Gatilhável por nível (Level-Triggered)**
2. **Gatilhável por rampa (Edge-Triggered)**

Os flip-flops gatilháveis por nível executam uma ação se a linha de clock estiver no nível 1 (ou 0 algumas vezes).

Deste modo, "**em um dispositivo gatilhável por nível as entradas de dados não devem ser mudadas, exceto após o término do estado ativo do clock.**"

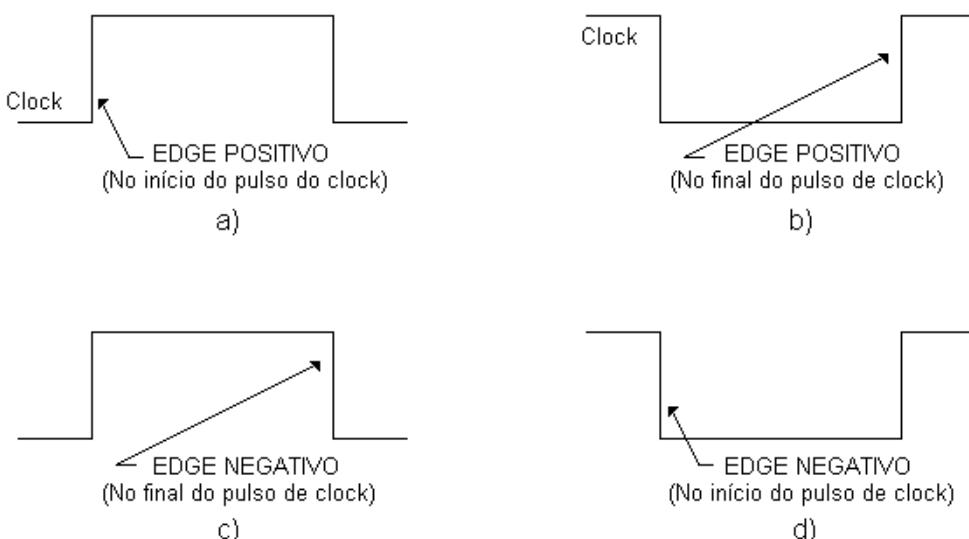
Os flip-flops gatilháveis por edge podem ser de 2 tipos:

- 1. Edge Positivo**
- 2. Edge Negativo**

Os gatilháveis por *Edge Positivo* fazem a transferência de informação, da entrada para a saída, na subida do pulso de *clock* (*edge positivo*).

Os gatilháveis por *Edge Negativo* fazem esta transferência na descida do pulso de *clock* (*edge negativo*).

A figura 9 mostra os tipos de *Edge*:

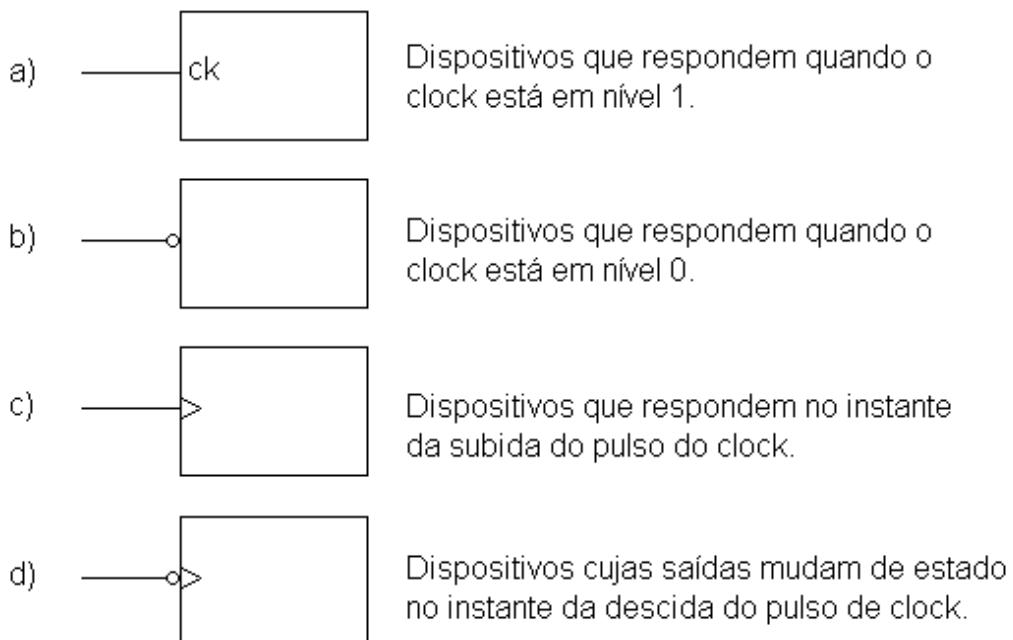


**Figura 9 - Tipos de Edge - a) Edge Positivo no início do clock "Leading Edge" - b) Edge Positivo no fim do clock "Trailing Edge" - c) Edge Negativo no fim do clock "Trailing Edge" - d) Edge Negativo no início do clock "Leading Edge".**

Pode-se estabelecer que:

"Basicamente um dispositivo gatilhável por *Edge* pode ter seus dados de entrada mudados em qualquer instante, exceto durante o *Edge*."

A figura 10 mostra as simbologias mais usuais da conexão do *clock*.



**Figura 10 - Símbolos das conexões do clock.**

## 2.7 FLIP-FLOPS GATILHÁVEIS DO TIPO MESTRE-ESCRAVO

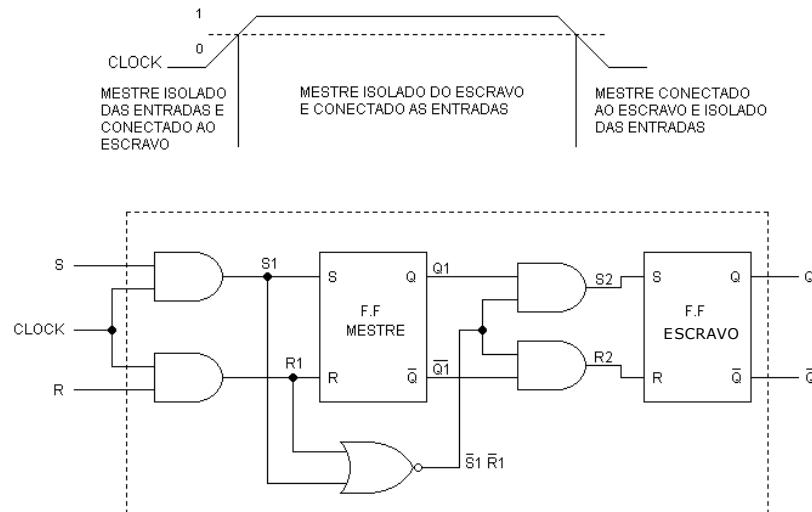
Estes flip-flops são classificados como gatilháveis por nível.

Basicamente, estes dispositivos são compostos de dois flip-flops, sendo um chamado **Mestre** e o outro chamado **Escravo**.

A fim de executar uma transferência correta de informação, da entrada para a saída, este flip-flop exige que o clock execute uma transição completa, isto é, saia do nível  $\emptyset$ , vá para o nível 1, permaneça um certo tempo e volte para o nível  $\emptyset$ .

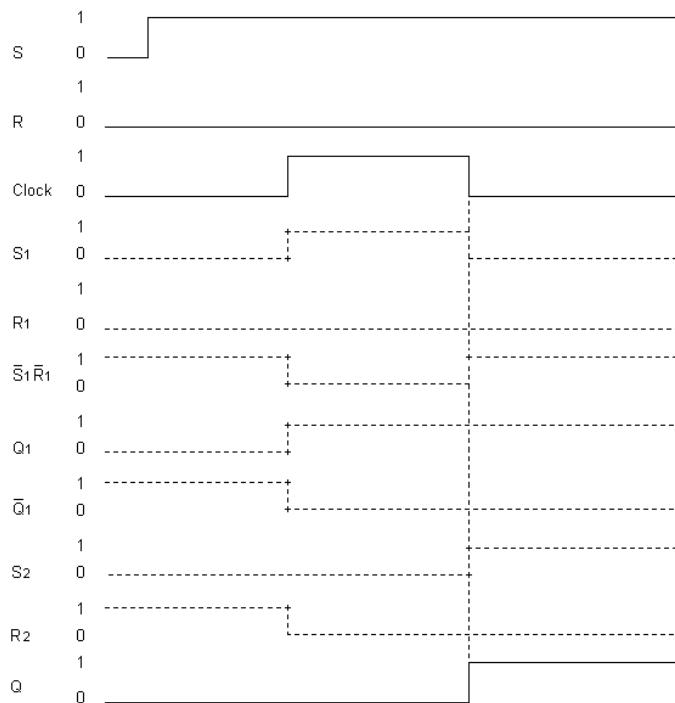
Deste modo, as entradas não devem mudar durante o ciclo de *clock*.

A figura 11 mostra um circuito e as fases de funcionamento do flip-flop Mestre-Escravo.



**Figura 11 - Flip-Flop Mestre-Escravo : Fases do Gatilhamento e Circuito de um FF Mestre-Escravo SR.**

A figura 12 mostra as formas de onda do circuito mostrado na figura 11.

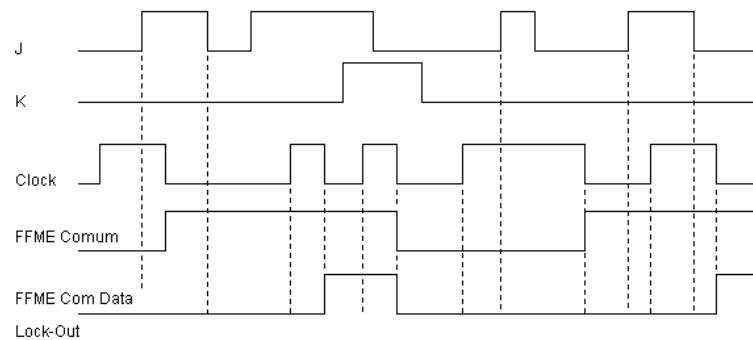


**Figura 12 - Formas de onda do FF Mestre-Escravo SR.**

### 2.7.1 FLIP-FLOP MESTRE-ESCRAVO COM DATA LOCK-OUT

Este flip-flop difere do Mestre-Escravo normal apenas pelo fato dos dados serem guardados no mestre no Edge positivo e transferidos para a saída no Edge negativo do pulso do *clock*. Deste modo, neste flip-flop pode haver mudança nas entradas durante o nível positivo do *clock*, pois a informação da entrada já foi armazenada no mestre no edge positivo do *clock*.

A figura 13 mostra as formas de onda FF Mestre-Escravo normal e do FF Mestre-Escravo com *Data Lock-Out* (74111).

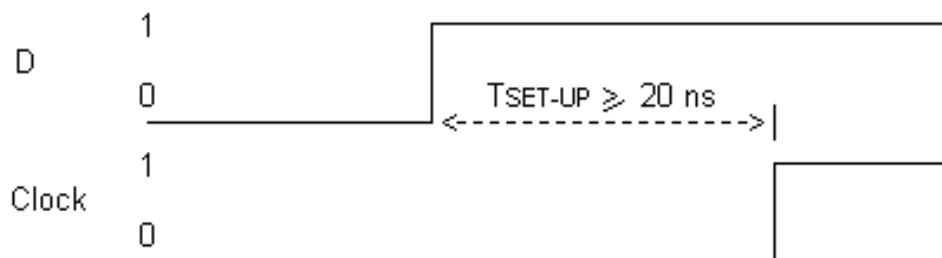


**Figura 13 - Formas de onda de comparação entre os Flip-Flops Mestre-Escravo Normal e com Data Lock-Out.**

## 2.8 PARÂMETROS IMPORTANTES DOS FLIP-FLOPS

### a) TEMPO DE SETAMENTO -tSET-UP

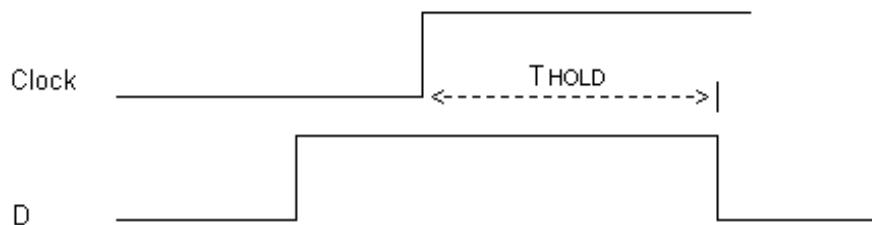
Este é o tempo mínimo que o sinal deve estar presente nas entradas, antes da ocorrência do pulso de gatilhamento. A figura 14 mostra tal fato:



**Figura 14 - TSET-UP (tempo de setamento).**

**b) TEMPO DE MANUTENÇÃO -  $T_{HOLD}$** 

Este é o tempo que o sinal deve permanecer nas entradas após a ocorrência do pulso de gatilhamento. A figura 15 mostra tal fato:



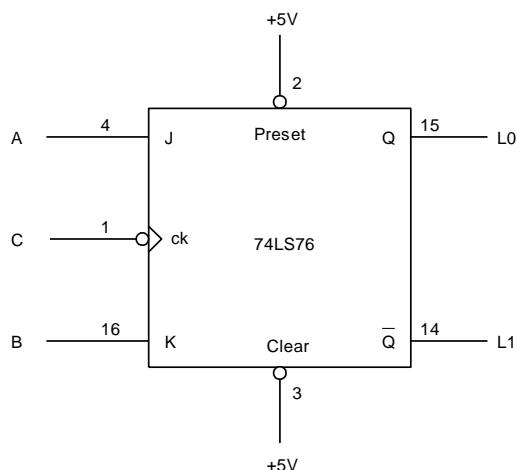
**Figura 15 -  $T_{HOLD}$  (tempo de Manutenção).**

**c) FREQUÊNCIA MÁXIMA –  $f_{máx}$** 

É a maior frequência dos pulsos de clock que pode ser aplicado ao dispositivo, mantendo um chaveamento confiável.

## 2.9 EXPERIÊNCIA 1

- Material utilizado: 1 x CI 74LS76
- Montar o circuito da figura 16. Consultar as folhas de dados para a ligação da alimentação.



**Figura 16 - Flip-Flop JK Mestre-Escravo 74LS76.**

- c)** Completar a tabela 1.

ENTRADAS			SAÍDAS	
J	K	Clock	Q	$\bar{Q}$
A	B	C	L $\emptyset$	L1
Ø	Ø	Ø		
Ø	Ø	1		
Ø	Ø	Ø		
1	Ø	Ø		
1	Ø	1		
1	Ø	Ø		
Ø	1	Ø		
Ø	1	1		
Ø	1	Ø		
1	1	Ø		
1	1	1		
1	1	Ø		

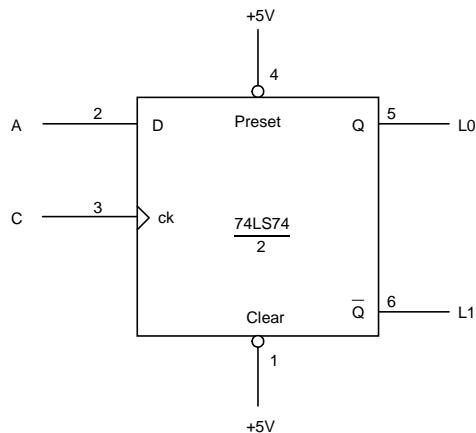
**Tabela 1 - Funcionamento do flip-flop 74LS76.**

### CONCLUSÃO:

Nesta experiência, pode-se observar o funcionamento do flip-flop mestre escravo e a necessidade da aplicação de um pulso completo de *clock* para que o dispositivo atue corretamente.

### 2.10 EXPERIÊNCIA 2

- a)** Material utilizado: 1 x CI 74LS74
- b)** Montar o circuito da figura 17. Consultar as folhas de dados para a ligação da alimentação.



**Figura 17 - Flip-Flop D Edge Triggered 74LS74.**

c) Completar a tabela 2.

ENTRADA		SAÍDAS	
$\emptyset$	Clock	Q	$\bar{Q}$
A	C	L $\emptyset$	L1
$\emptyset$	$\emptyset$		
$\emptyset$	1		
$\emptyset$	$\emptyset$		
1	$\emptyset$		
1	1		
1	$\emptyset$		
1	1		
$\emptyset$	1		
1	1		
$\emptyset$	1		

D =  $\emptyset$        $\left\{ \begin{array}{l} \text{Clock } \emptyset \rightarrow 1 \\ \text{Clock } 1 \end{array} \right.$   
 Clock  $\emptyset \rightarrow 1$   
 D = 1       $\left\{ \begin{array}{l} \text{Clock } \emptyset \rightarrow 1 \\ \text{Clock } 1 \end{array} \right.$   
 Clock  $\emptyset \rightarrow 1$   
 Clock = 1  
 D  $1 \rightarrow \emptyset$        $\left\{ \begin{array}{l} \text{Clock } \emptyset \rightarrow 1 \\ \text{Clock } 1 \end{array} \right.$   
 $\emptyset \rightarrow 1$

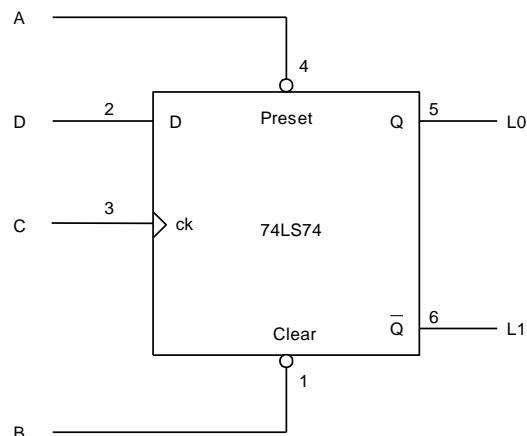
**Tabela 2 - Funcionamento do FF D 74LS74.**

### CONCLUSÃO:

Observar que a transferência de informação da entrada “D” para a saída se faz na subida do *clock*, e que as entradas podem mudar de estado durante a permanência do *clock* em “1” ou “ $\emptyset$ ”, sem alteração da saída.

## 2.11 EXPERIÊNCIA 3

- a)** Material utilizado: 1 x CI 74LS74
- b)** Montar o circuito da figura 18.



**Figura 18 - Circuito para teste do Preset e do Clear.**

- c)** Completar a tabela 3.

ENTRADAS				SAÍDAS	
Prese	Clea	Cloc	D	Q	Q
A	B	C	D	L0	L1
1	1	Ø	Ø		
1	1	1	Ø		
1	1	Ø	Ø		
Atuação do Preset		Ø	1	Ø	Ø
		Ø	1	1	Ø
		1	1	Ø	Ø
		1	Ø	Ø	Ø
Atuação do Clear		1	Ø	Ø	1
		1	Ø	1	1
Atuação do Preset e do Clear simultâneos		Ø	Ø	1	1
		Ø	Ø	Ø	Ø

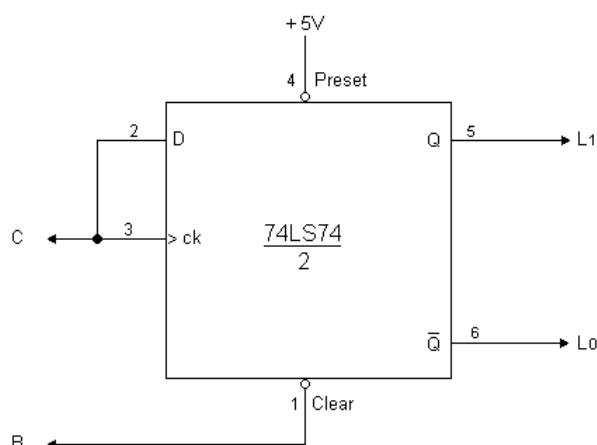
**Tabela 3 - Funcionamento do Preset e do Clear.**

## CONCLUSÃO:

Pode ser notado, pela tabela de funcionamento, que um “ $\emptyset$ ” na linha do **Preset** faz a saída “Q” ir para o nível “1” (“preseta” a saída), e um “ $\emptyset$ ” na linha **clear** faz a saída “Q” ir para “ $\emptyset$ ” (“limpa” a saída). Ainda, deve ser observado que as linhas **Preset** e **Clear** têm prioridade sobre as outras linhas de entrada (no caso, sobre as linhas “D” e **clock**). Pela informação das últimas duas linhas da tabela, vê-se que a atuação de **Preset** e do **Clear**, simultaneamente, faz com que a saída Q e  $\bar{Q}$  fiquem em nível 1, perdendo assim a característica de serem complementares.

### 2.12 EXPERIÊNCIA 4

- a) Material Utilizado 1 x CI 74LS74 e 1 x CI 74LS04
- b) Montar o circuito da figura 19.



**Figura 19- Circuito para teste do tSET-UP**

- c) Completar a tabela 4.

ENTRADAS		SAÍDAS	
D = ck	Clear	Q	$\bar{Q}$
C	B	L1	L $\emptyset$
$\emptyset$	$\emptyset$		
$\emptyset$	1		
1	1		

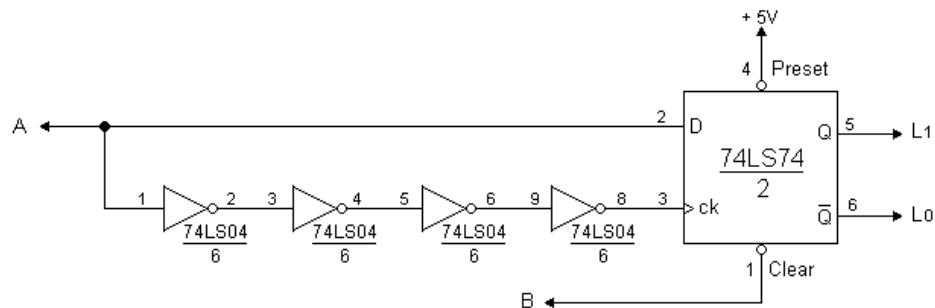
**Tabela 4 - Funcionamento do Circuito da Figura 19.**

### **CONCLUSÃO:**

Quando a chave “C” vai de  $\emptyset$  para 1, a saída deveria mudar de  $\emptyset$  para 1, pois a entrada D está com 1 e houve um edge positivo do clock. Isto não ocorre devido ao set-up-time, pois a informação nas linhas de entrada devem estar disponíveis um tempo antes do aparecimento do edge positivo do clock, tempo este chamado de  $t_{SET-UP}$  (setamento).

- d)** Introduzir, no circuito da figura 19, a modificação mostrada na figura 20.

Este circuito fornece um atraso entre o dado e o clock. Caso o uso de 4 inversores não for suficiente, deve-se aumentar o número de inversores para obter um atraso maior. Este atraso pode variar dependentemente do fabricante.



**Figura 20 - Circuito com Introdução de um atraso na linha do Clock**

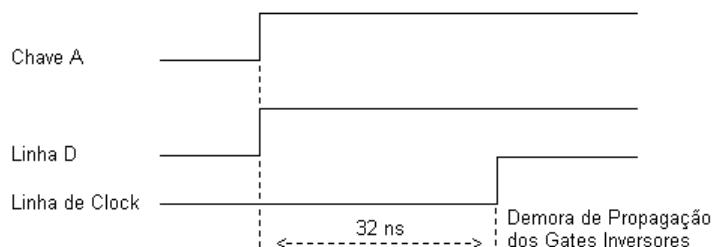
- e)** Completar a tabela 5.

ENTRADAS		SAÍDAS	
D = ck	Clear	Q	$\bar{Q}$
A	B	L1	L $\emptyset$
$\emptyset$	$\emptyset$		
$\emptyset$	1		
1	1		

**Tabela 5 - Funcionamento do Circuito da Figura 20.**

## **CONCLUSÃO:**

Com a modificação introduzida, o atraso de tempo dos gates inversores promovem o  $t_{SET-UP}$ . Deste modo o dispositivo funcionou corretamente. A figura 26 mostra as formas de onda do ocorrido.



## **2.13 EXERCÍCIOS**

- a)** Com um flip-flop J-K pode-se construir um flip-flop D.

1. Certo

2. Errado

- b)** As linhas de Preset e de Clear prevalecem sobre as outras linhas de entrada.

1. Certo

2. Errado

- c)** Em um flip-flop Mestre-Escravo o funcionamento correto exige que o clock:

1. Tenha um Edge negativo

2. Tenha um Edge positivo

3. Saia do nível  $\emptyset$ , vá para 1, permaneça um tempo

mínimo 1, e volta para  $\emptyset$

- d)** O tempo set-up em um flip-flop D exige que:

1. A entrada D esteja presente após o clock

2. A entrada D esteja presente antes do clock

3. A entrada D e o clock devam ocorrer simultaneamente.

## **CAPÍTULO 3 - CONTADORES**

### **3.1 DEFINIÇÃO**

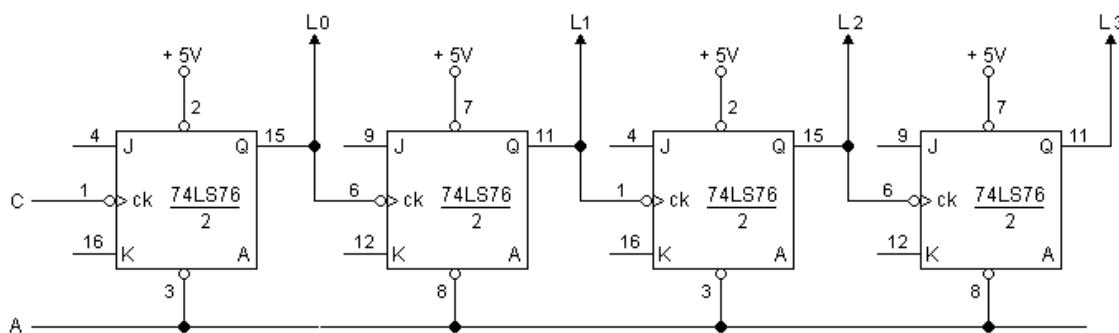
Um contador é um registro que pode somar uma quantidade, geralmente um, ao dado guardado.

### **3.2 CONTADORES RIPPLE (ASSINCRONO)**

Nos contadores ripple a saída de um flip-flop é conectada a entrada do clock do próximo. Tal contador é, às vezes, chamado **contador série**.

#### **3.2.1 EXPERIÊNCIA 1**

- Material Utilizado: 2 x CI's 74LS76.
- Montar o circuito da figura 1 e alimentar os componentes.



*Figura 1 - Contador ripple de 0000 até 1111 com retorno.*

#### **c) OBSERVAÇÃO:**

Na figura 1 lembrar que as entradas J e K abertas funcionam como se estivessem conectadas ao nível lógico 1.

- d) Completar a tabela 1.

CLEAR	CLOCK	SAÍDAS			
		L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
A	C				
Ø	x				
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					
Ø					

①  
②

③

**Tabela 1 - Tabela para circuito da figura 1**

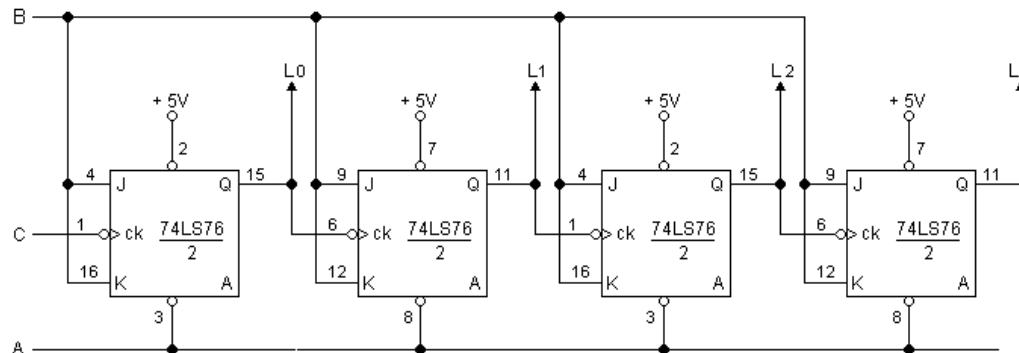
- e) Na tabela 1 observar que em:

- ① e ③ A = Ø, é feito um clear no contador, portanto L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>Ø</sub> = ØØØØ.
- ② o contador conta em binário puro desde ØØØØ (Ø<sub>10</sub>) até 1111 (15<sub>10</sub>), retornando a ØØØØ (Ø<sub>10</sub>) e o ciclo se repetindo.

- f) O contador da figura 1 é chamado **contador ripple**. O flip-flop mais significativo (o que tem saída  $L_3$  na figura 1) não pode mudar de estado durante um tempo correspondente a 4 demoras (uma para cada flip-flop). Se, por exemplo, o  $T_P$  de um flip-flop é 10ns, o contador gasta 40ns após o edge negativo do clock para fornecer a palavra correta. O circuito da figura 1 pode ser ampliado para **contadores ripple maiores**.

### 3.2.2 EXPERIÊNCIA 2

- a) Material Utilizado: 2 x CI's 74LS76.  
b) Modificar o circuito da experiência 1 como mostrado na figura 2.



**Figura 2 - Contador ripple controlado.**

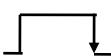
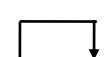
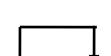
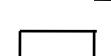
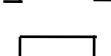
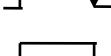
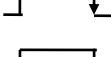
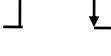
- c) Completar a tabela 2.

CLEAR	CONTROLE	CLOCK	SAÍDAS			
			$L_3$	$L_2$	$L_1$	$L_0$
$\emptyset$	x	x				
1	$\emptyset$					
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					

①

②

③

CLEAR	CONTROLE	CLOCK	SAÍDAS			
			L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					
1	1					

(3)

**Tabela 2 - Tabela para o circuito da figura 2.**

d) Na tabela 2 observar que em:

- ① A = Ø, é dado um clear no contador e L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>Ø</sub> = ØØØØ.
- ② Com o controle B = Ø o contador não conta.
- ③ Com B = 1 o contador conta normalmente.

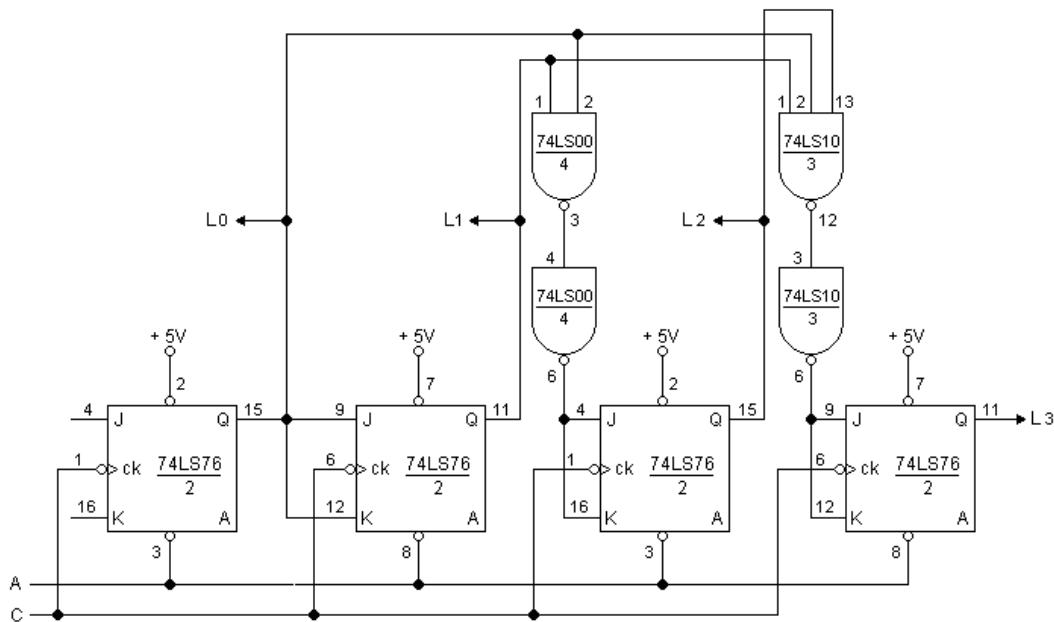
### 3.3 CONTADORES SÍNCRONOS

A fim de contornar o problema da demora de propagação dos contadores ripple, usa-se os contadores síncronos, que tem a característica de possuir as entradas de clock dos flip-flops ligados em paralelo.

#### 3.3.1. EXPERIÊNCIA 3

a) Material Utilizado: 1 x CI 74LS00, 1 x CI 74LS10 e 2 x CI's 74LS76

- b)** Montar o circuito da figura 3 e alimentar os componentes.



**Figura 3 - Contador Síncrono.**

- c)** Completar a tabela 3.

CLEAR	CLOCK	SAÍDAS			
		L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø	X				
1					
1					
1					
1					
1					
1					
1					
1					
1					
1					

①

②

CLEAR	CLOCK	SAÍDAS			
		L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
A	C				
1					
1					
1					
1					
1					
1					
1					

(2)

**Tabela 3 - Tabela para circuito da figura 3.**

- d) Na tabela 3 observar que em:

- ① A = Ø, é dado um clear no contador e L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>Ø</sub> = ØØØØ.
- ② A = 1 e há clock, portanto o contador conta normalmente desde ØØØØ até 1111, retornando a ØØØØ, e repetindo o ciclo.

### 3.4 CONTADORES SÍNCRONOS CONTROLADOS

Com pequenas modificações no circuito da figura 3, pode-se tornar o circuito em um contador síncrono controlado, no qual só haverá contagem quando acionada a linha de contagem.

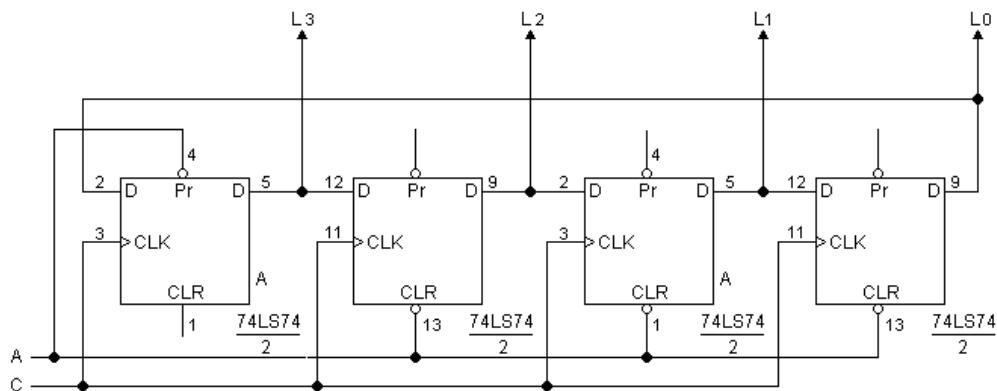
### 3.5 CONTADORES EM ANEL

Em vez de contar com números binários, um contador Anel usa palavras que tem um único Bit alto (nível lógico 1).

#### 3.5.1 EXPERIÊNCIA 4

- a) Material Utilizado: 2 x CI's 74LS74

- b)** Montar o circuito da figura 4 e alimentar os componentes.



**Figura 4 - Contador em anel de 4 bits.**

- c)** Completar a tabela 4.

CLEAR		CLOCK		SAÍDAS			
A	C		X	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø							
1							
1							
1							
1							

①  
②

**Tabela 4 - Tabela para o circuito da figura 4**

- d)** Na tabela 4 observar que em:

- ① A = Ø, é estabelecida a condição inicial do contador (no caso L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>0</sub> = 1ØØØ).
- ② O contador conta de 1ØØØ para Ø1ØØ, deste valor para ØØ1Ø e daí para ØØØ1, retornando a 1ØØØ, onde é reiniciado outro ciclo.

### 3.6 CONTADORES NA FORMA DE CI's

Devido a necessidade geral de contadores, já existem muitos contadores de forma de CI's. Na série TTL 74 os mais simples são o 74LS90, 74LS92 e 74LS93 que são, respectivamente, contadores de décadas, contador que divide por 12 e contador que divide por 16.

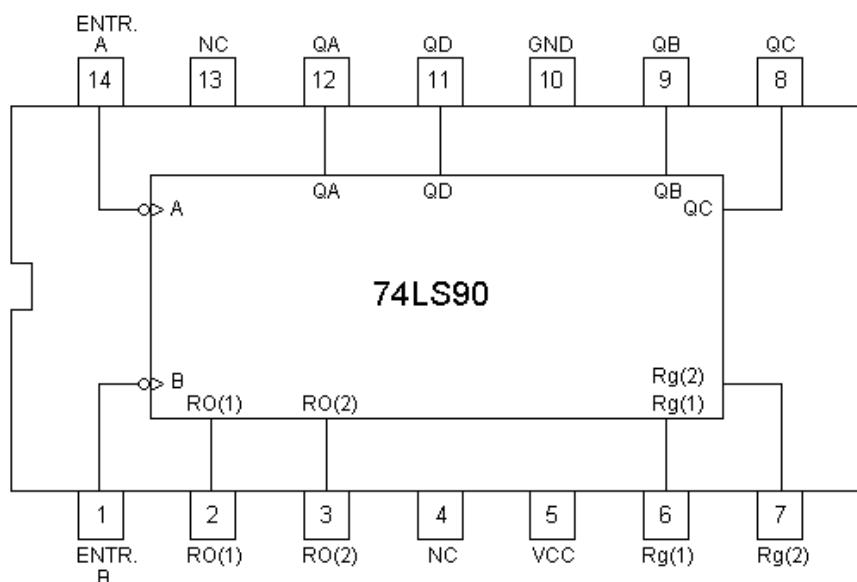
Em alguns casos estes contadores trabalham como contadores ripple e podem produzir **GLITCHES** nas saídas.

O 74LS193 é outro contador da série 74, porém com mais recursos.

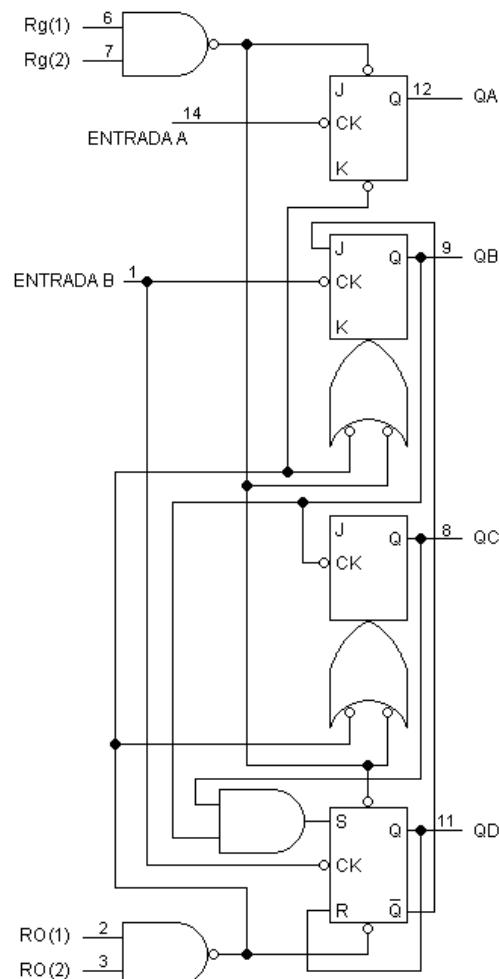
#### 3.6.1 CONTADOR 74LS90

O 74LS90 é um contador de décadas (Módulo 10), que consiste de um flip-flop JK e um circuito que divide por 5. No 74LS90 o flip-flop A funciona como um contador que divide por 2, e os outros três flip-flops funcionam como um contador que divide por 5.

A figura 5 mostra o lay-out do 74LS90; a figura 6 mostra seu diagrama funcional em blocos; a tabela 5 mostra a contagem do CI e a tabela 6, mostra a tabela reset/contagem.



**Figura 5 - Lay-out do CI 74LS90**



**Figura 6 - Diagrama Funcional em blocos do CI 74LS90.**

CONTAGEM	SAÍDAS			
	QD	QC	QB	QA
Ø	Ø	Ø	Ø	Ø
1	Ø	Ø	Ø	1
2	Ø	Ø	1	Ø
3	Ø	Ø	1	1
4	Ø	1	Ø	Ø
5	Ø	1	Ø	1
6	Ø	1	1	Ø
7	Ø	1	1	1
8	1	Ø	Ø	Ø
9	1	Ø	Ø	1

**Tabela 5 - Contagem do CI 74LS90.**

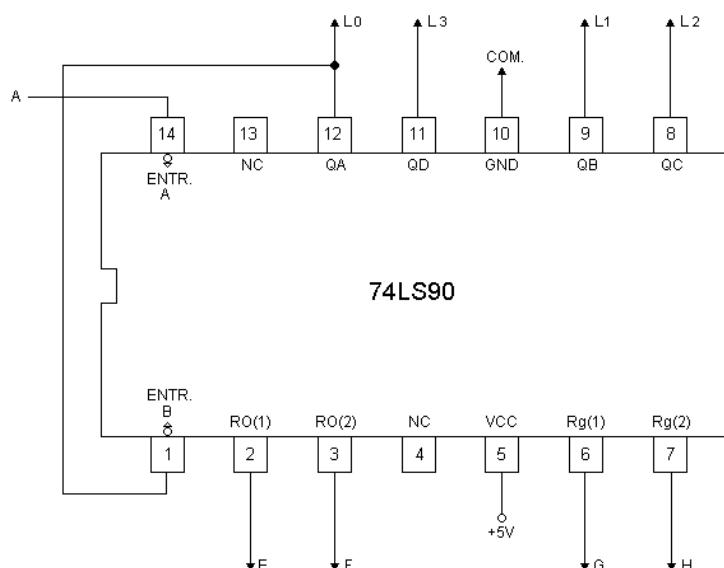
ENTRADAS RESET				SAÍDAS			
RØ(1)	RØ(2)	Rg(1)	Rg(2)	QD	QC	QB	QA
1	1	Ø	X	Ø	Ø	Ø	Ø
1	1	X	Ø	Ø	Ø	Ø	Ø
X	X	1	1	1	Ø	Ø	1
X	Ø	X	Ø				Contagem
Ø	X	Ø	X				Contagem
Ø	X	X	Ø				Contagem
X	Ø	Ø	X				Contagem

**Tabela 6 – Reset / Contagem do CI 74LS90.**

No 74LS90, se o sinal de entrada é conectado a entrada A (pino 14), e a saída QA (pino 12) é conectada a entrada B (pino 1) obtém-se um contador de décadas, e as saídas são as mesmas que as da tabela 5. As entradas Rø(1) e Rø(2) são entradas clear. As saídas QD QC QB QA do 74LS90 são todas baixas ("Øs") sempre que Rø(1) e Rø(2) são altas "1s", desde que uma das entradas Rg seja "Ø", como mostrado na tabela 6. O contador pode ser presetado para uma contagem de  $9_{10}$ , e para tal Rg(1) e Rg(2) devem ser altas (1). Se o contador deve contar normalmente, no mínimo uma das entradas Rø e uma das entradas Rg devem ser "0".

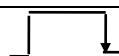
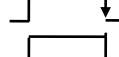
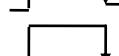
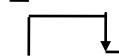
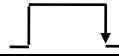
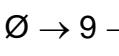
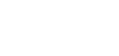
### 3.6.1.1 EXPERIÊNCIA 5

- a) Material Utilizado: 1 x CI 74LS90
- b) Montar o circuito da figura 7.



**Figura 7 - CI 74LS90 como contador de década.**

**c)** Completar a tabela 7.

ENTRADAS RESET/CONTAGEM				ENTRADA	SAÍDAS			
RØ(1)	RØ(2)	Rg(1)	Rg(2)	A	QD	QC	QB	QA
E	F	G	H	A	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
1	1	x	Ø	x				
1	1	Ø	x	x				
x	x	1	1	x				
1	1	x	Ø	x				
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					
Ø	x	Ø	x					

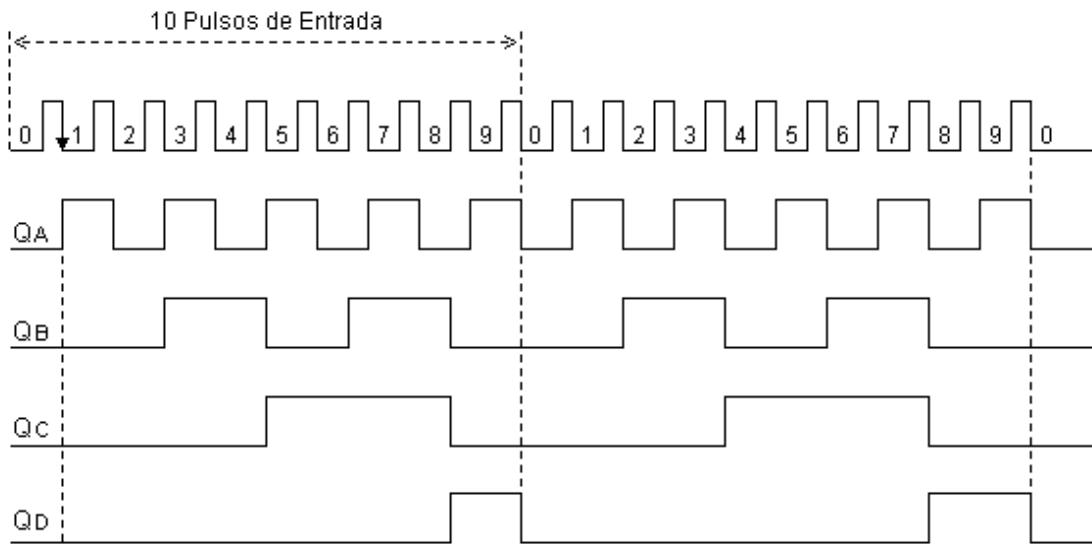
**Tabela 7 - Tabela para o circuito da figura 7.**

**d)** Na tabela 7 observar que em:

- ① é dado o clear nas saídas.
- ② as saídas são presetadas para 1001.
- ③ é efetuada a contagem de Ø → 9 → Ø.

Confira pelas tabelas das 5 e 6.

**e)** A figura 8 mostra as formas de ondas correspondentes às saídas da tabela 7.



**Figura 8 - Gráfico das saídas do circuito da figura 7.**

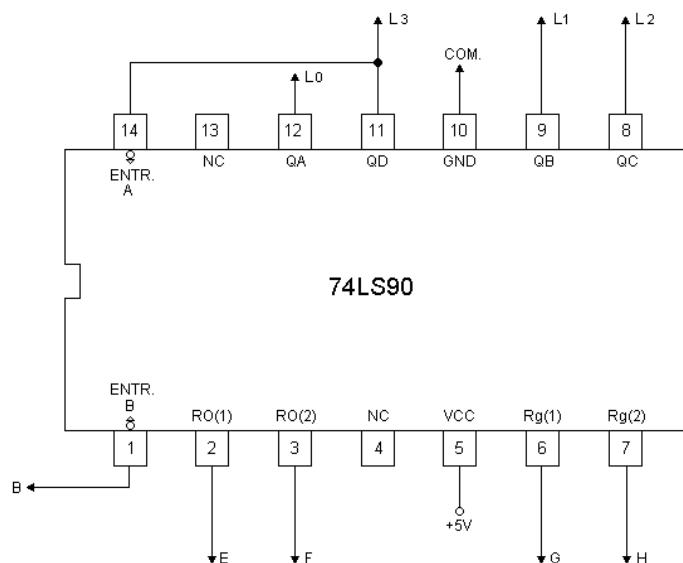
- f) A figura 8 mostra uma sequência de 20 pulsos de entradas, e pode-se notar nesta sequência que, a cada 20 pulsos de entrada, ocorrem dez pulsos na saída Q<sub>A</sub> e 2 pulsos na saída Q<sub>D</sub>. Pode-se observar na figura 8 que o tempo em que a saída QD fica alta é bem menor do que o tempo em que ela fica baixa, portanto, não é uma onda simétrica.

A fim de se conseguir pulsos simétricos com o 74LS90 e com uma frequência de 1/10 da frequência de clock, os fabricantes aconselham conectar a saída QD (pino 11) à entrada A (pino 14) e aplicar a entrada de contagem à entrada B (pino 1), obtendo-se, deste modo, uma onda quadrada com frequência dividida por dez na saída Q<sub>A</sub>.

### 3.6.1.2 EXPERIÊNCIA 6

- Material Utilizado: 1 x CI 74LS90
- Montar o circuito da figura 9.

## CAPÍTULO 3 – CONTADORES

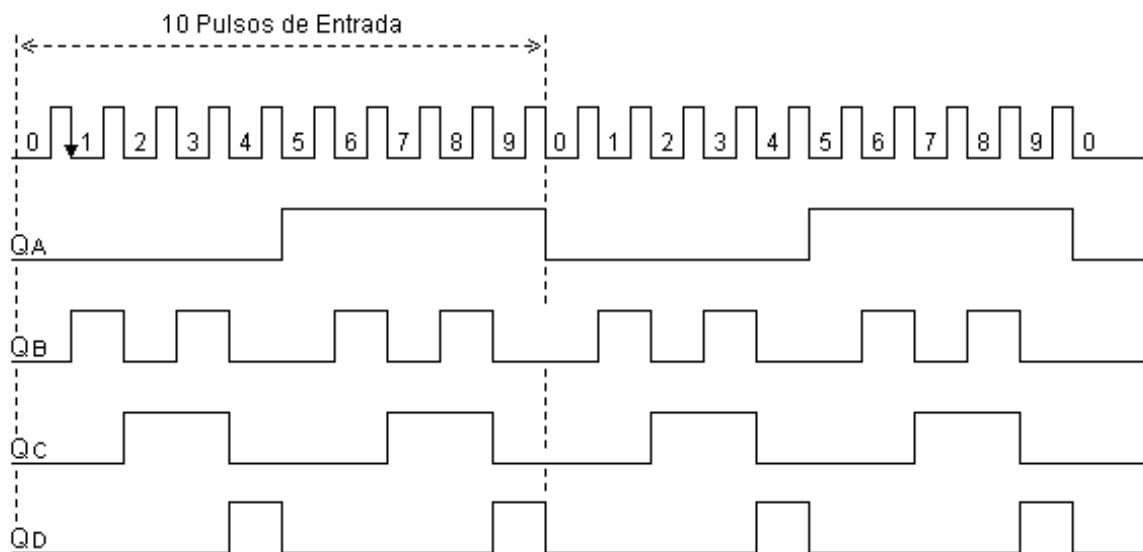


**Figura 9 - CI 74LS90 para fornecer onda simétrica com frequência de 1/10 da de clock**

c) Completar a tabela 8.

**Tabela 8 - Tabela para o circuito da figura 9.**

- d) Na tabela 8 observar que em:
- ① é dado o clear nas saídas.
  - ② é efetuada a contagem.
- e) A figura 10 mostra as formas de ondas correspondentes às saídas da tabela 8.



**Figura 10 - Formas de ondas das saídas do circuito da figura 9.**

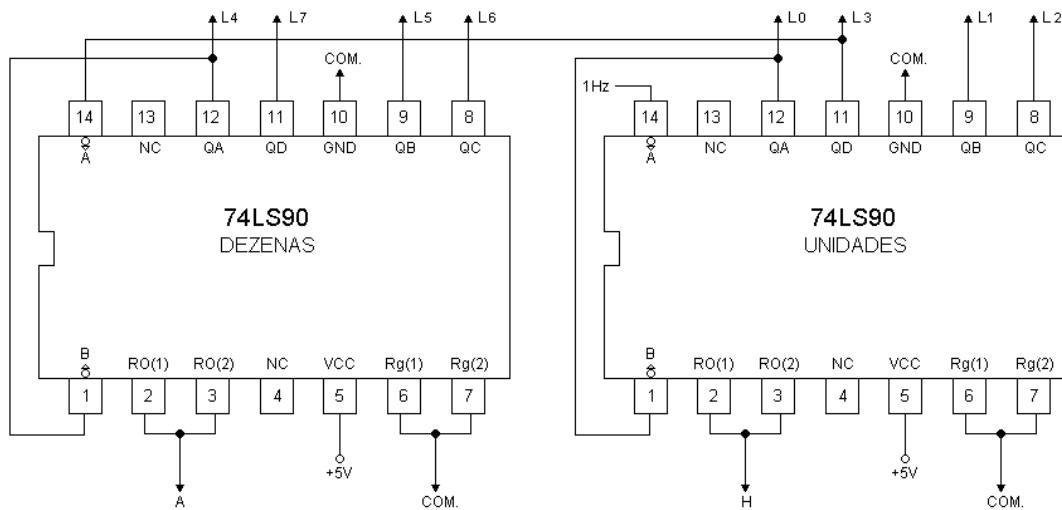
- f) Da figura 10, pode-se observar que a forma de onda da saída QA é quadrada simétrica e que ocorre um pulso QA a cada 10 pulsos de entrada, tendo o pulso em QA uma duração de cinco vezes a duração do pulso de entrada.

### 3.6.1.3 LIGAÇÃO DE CONTADORES EM CASCATA

Pode-se acoplar contadores a fim de se obter um contador de módulo maior. Se dois contadores de módulo  $M_1$  e  $M_2$  forem acoplados corretamente, pode-se obter um contador de módulo  $M = M_1 \times M_2$ .

### 3.6.1.4 EXPERIÊNCIA 7

- a) Material Utilizado: 2 x CI's 74LS90
- b) Montar o circuito da figura 11.



**Figura 11 - CI 74LS90 em Cascata.**

c) Completar a tabela 9.

ENTRADAS RESET		ENTRADA	SAÍDAS								
Dezenas	Unidades	A	Dezenas				Unidades				
R0(1)	R0(2)	R0(1)	R0(2)	QD	QC	QB	QA	QD	QC	QB	QA
A	H	1 Hz	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	①
1	1	X									
Ø	Ø										②
Ø	Ø										
Ø	Ø										
Ø	Ø										
Ø	Ø										
Ø	Ø										
Ø	Ø										
Ø	Ø										
Ø	Ø										

**Tabela 9 - Tabela para o circuito da figura 11.**

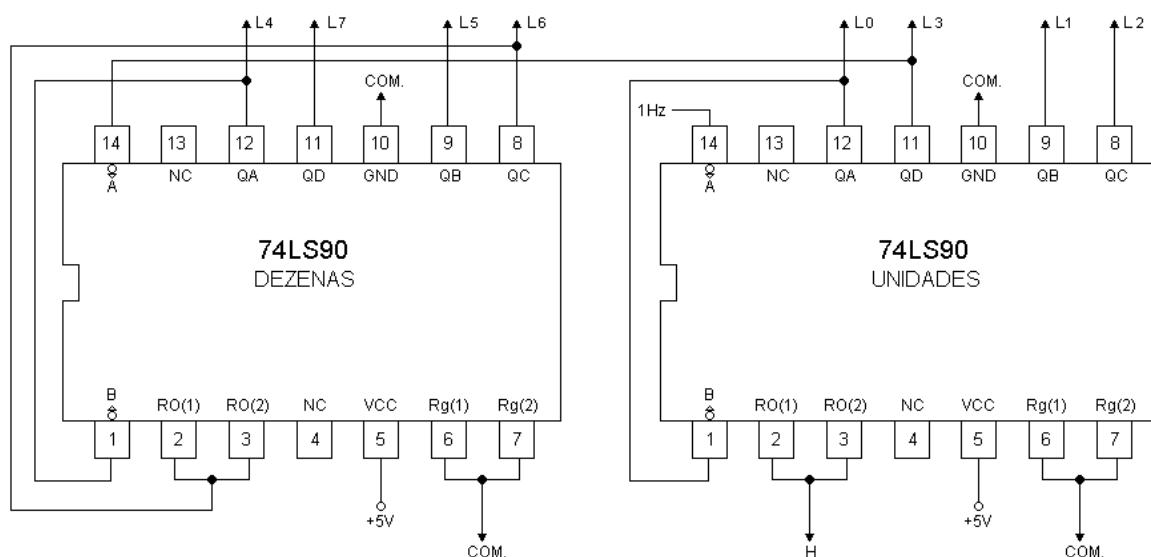
d) Na tabela 9 pode-se observar que em:

- ② as saídas  $L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0$  contam crescentemente em BCD 8421, desde  $0000\ 0000$  até  $0000\ 1001$ , o que corresponde a contagem entre os decimais 00 e 09.
- ③ reinicia-se o ciclo de contagem em  $0001\ 0000$ , isto é decimal 10. Continuando-se a aplicar pulsos de clock, o sistema continuaria contando, sucessivamente, até atingir a contagem binária  $1001\ 1001$ , o que corresponde ao decimal 99; quando, então, o circuito retornará a sua contagem original  $0000\ 0000$ . Então o circuito da figura 11 é um contador de módulo 100.

**Observação:** Não desmontar o circuito.

### 3.6.1.5 EXPERIÊNCIA 8

- a) Material Utilizado: 2 x CI's 74LS90
- b) Modificar o circuito da figura 11 como na figura 12. Para tal, o circuito da figura 11 deve sofrer a seguinte modificação: desligar as entradas  $R\bar{O}(1)$  e  $R\bar{O}(2)$  do contador das dezenas que estão conectadas a chave “A” do equipamento e conectá-las ( $R\bar{O}(1)$   $R\bar{O}(2)$ ) ao pino 8 (saída QC) do contador das dezenas.



**Figura 12 - Contadores com CI 74LS90 para contagem até 39 com retorno.**

**c)** Completar a tabela 10.

ENTRADAS RESET		ENTRADA	SAÍDAS								
RØ(1)	RØ(2)	A	QD	QC	QB	QA	QD	QC	QB	QA	
H		1 Hz	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>	
1		X									
Ø		1º Pulso 									
Ø		Ø									
Ø		39º Pulso 									
Ø		40º Pulso 	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	

①  
②  
③

**Tabela 10 - Tabela para o circuito da figura 12.**

**d)** Na tabela 10 observar que em:

- ① é dado o clear nas saídas, isto é L<sub>7</sub> L<sub>6</sub> L<sub>5</sub> L<sub>4</sub> L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>Ø</sub> = ØØØØ ØØØØ.
- ② o contador conta crescentemente.
- ③ o contador retorna a ØØØØ ØØØØ.

**e)** Pergunta-se qual a maior contagem assinalada pelo contador em ②?

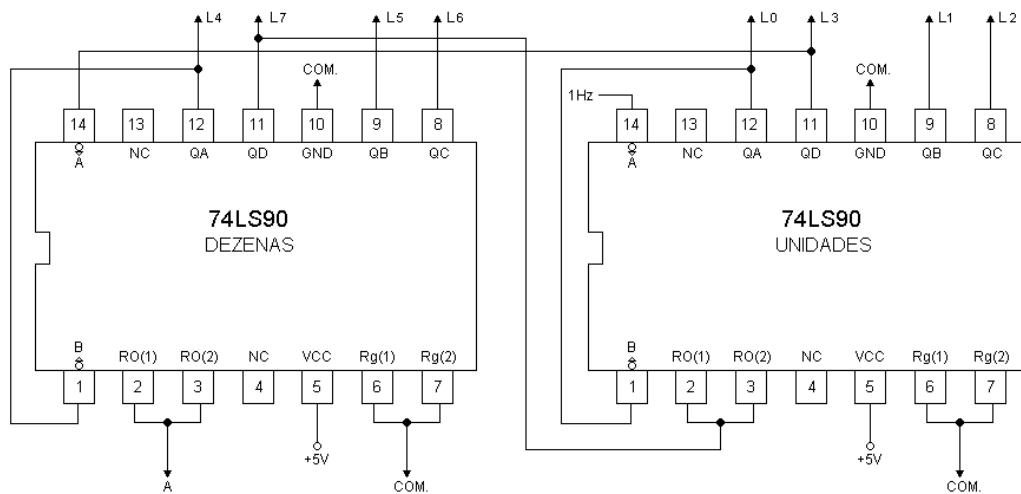
Resposta: \_\_\_\_\_

**f)** Pede-se agora fazer uma pequena modificação no circuito da figura 12, a fim de se obter um contador que conte crescente e ciclicamente, desde Ø até 19 retornando a Ø.

**Observação: Não desfazer a montagem.**

### 3.6.1.6 EXPERIÊNCIA 9

- a) Material Utilizado: 2 x CI's 74LS90
- b) Modificar o circuito da figura 12 de modo a obter o circuito da figura 13. Para isto, retornar as entradas RØ(1) e RØ(2) do contador de unidades a saída QD (pino 11) do contador de dezenas.



**Figura 13 - Contador com CI 74LS90 para contagem até 80, parando no 80.**

- c) Completar a tabela 11.

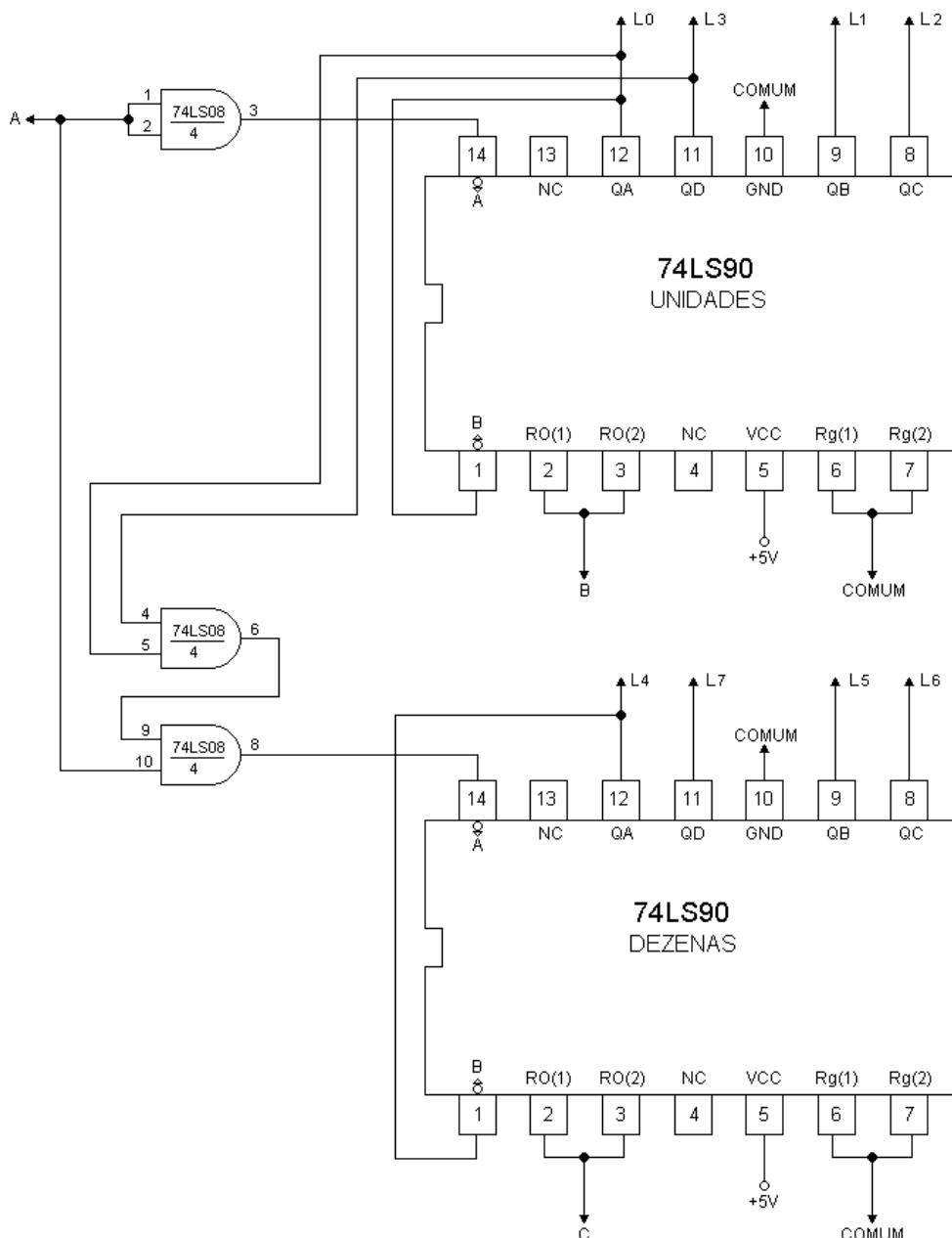
ENTRADAS RESET		ENTRADA	SAÍDAS								
RØ(1)	RØ(2)	A	QD	QC	QB	QA	QD	QC	QB	QA	
		1 Hz	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	
1	X						X	X	X	X	
Ø	Ø	1º Pulso 									
Ø	Ø	80º Pulso 									
Ø	Ø	81º Pulso 									
1											

**Tabela 11 - Tabela para o circuito da figura 13**

- d)** Na tabela da figura 24 observar que em:
- ① e ③ é dado um clear nos 4 bits mais significativos do contador, isto é  $L_7 L_6 L_5 L_4 = \emptyset\emptyset\emptyset\emptyset$ .
  - ② o contador conta, parando em um determinado valor.
- e)** Pergunta-se qual a maior contagem assinalada em ② pelo contador.  
Resposta: \_\_\_\_\_
- f)** Pede-se agora fazer uma pequena modificação no circuito da figura 13 de modo a se obter um contador que conte:
- a) de 0 a 40 parando em 40
  - b) de 0 a 20 parando em 20
- g)** As saídas dos contadores poderiam ser ligadas a dois conjuntos decodificador / display no lugar dos led's.

### 3.6.1.7 EXPERIÊNCIA 10

- a) Material Utilizado: 2 x CI's 74LS90 e 1 x CI 74LS08
- b) Montar o circuito da figura 14.



**Figura 14 - Acoplamento Pseudo-Síncrono de Contadores Síncronos.**

- c) Executar a sequência descrita a seguir:
1. Colocar as chaves: A = Ø, B = 1 e C = 1, isto faz o zeragem dos contadores.
  2. Colocar A = Ø, B = Ø e C = Ø, liberando os contadores para a contagem.
  3. Atuar na chave “A” de Ø para 1 e depois para Ø, notando que o contador das unidades conta um.

4. Repetir o item anterior mais 7 vezes, observar que o contador das unidades parou no número 8.
5. Atuar novamente em A ( $\emptyset \rightarrow 1 \rightarrow \emptyset$ ) e observar que os led's  $L_3$ ,  $L_2$ ,  $L_1$  e  $L_0$  mostram o número nove (1001). Agora o pino 9 do CI 74LS08 (gate E) estará no nível 1.
6. Colocar A = 1; isto faz o pino 8 do CI 74LS08 ir para o nível 1. Assim as entradas do clock de ambos os contadores estarão em 1, prontos para executar a transição na queda do clock.
7. Fazer A =  $\emptyset$ ; neste ponto os dois contadores executaram a transição ao mesmo tempo, como é a característica do acoplamento síncrono. O contador das unidades passa a mostrar no display 0000, e o contador das dezenas 0001.
8. Repetir os itens 3, 4, 5, 6 e 7, e observar a próxima transição em que participam os dois contadores (isto é de 19 para 20).

### **CONCLUSÃO:**

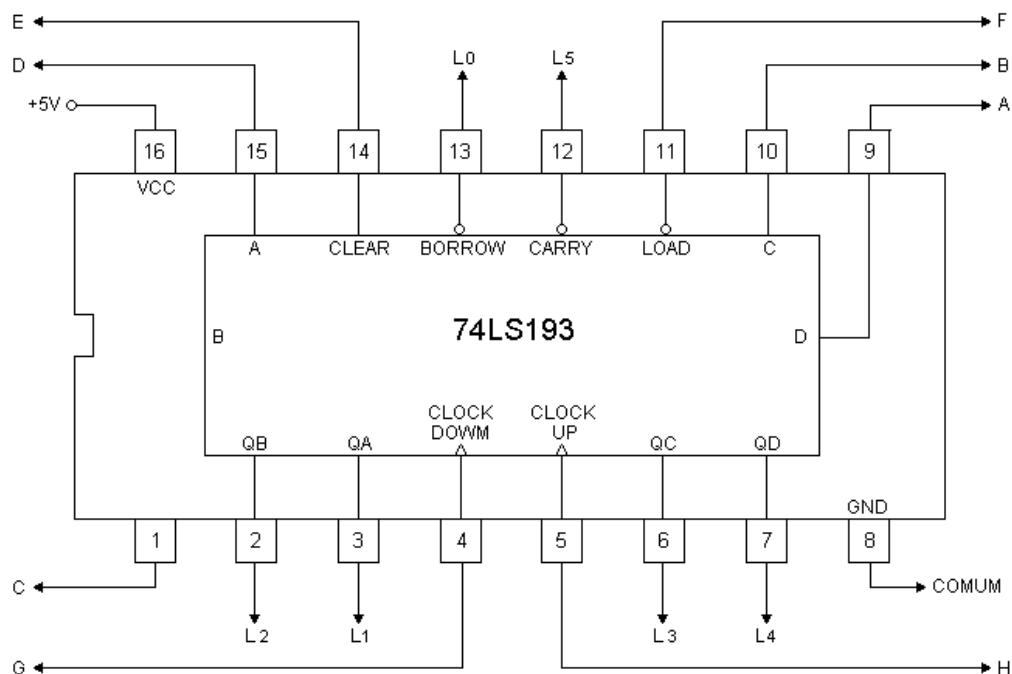
Para os pulsos de clock que não executam transição nas dezenas, o pino 6 do CI 74LS08 se mantém em  $\emptyset$  impedindo, através do gate “E”, que o clock atue no contador das dezenas. Porém, quando a unidade atinge a contagem 9, o pino 6 do CI 74LS08 vai para 1, abrindo assim o gate “E” e dando passagem para o clock atuar no contador das dezenas.

De tal modo que, quando o pulso de clock passar de 1 para  $\emptyset$ , os dois contadores recebem a transição ao mesmo tempo, executando as mudanças simultânea-mente como se fossem síncronos. A grande vantagem deste circuito é a maior precisão e velocidade com que pode ser implementado o circuito contador, usando vários contadores.

O uso da configuração assíncrona introduz a soma do atraso de cada contador na transmissão dos pulsos de clock de um contador para o próximo, o que não ocorre no circuito pseudo-síncrono analisado.

#### **3.6.1.8 EXPERIÊNCIA 11**

- a) Material Utilizado: 1 x CI 74LS193
- b) Montar o circuito da figura 15.



**Figura 15 - Circuito do contador UP/DOWN 74LS193**

c) Complete a tabela 12.

ENTRADA				CLOCK				SAÍDAS							
D	C	B	A	Clear	Load	Down	Up	Borrow	Carry	QD	QC	QB	QA		
D	C	B	A	E	F	G	H	L5	L4	L3	L2	L1	L0		
Ø	Ø	Ø	Ø	Ø	1	Ø	Ø								①
Ø	Ø	Ø	Ø	1	1	Ø	Ø								②
Ø	Ø	Ø	Ø	1	Ø	Ø	Ø								③
Ø	1	1	Ø	Ø	Ø	Ø	Ø								④
Ø	1	1	Ø	Ø	1	Ø	Ø								⑤
1	1	1	1	Ø	Ø	Ø	Ø								
1	1	1	1	Ø	1	Ø	Ø								
X	X	X	X	1	1	Ø	Ø								
X	X	X	X	Ø	1	Ø	Ø								

**Tabela 12 - Tabela para o circuito da figura 15.**

Nesta tabela, mostra-se as operações de Clear e Load do 74LS193:

- Em ①, coloca-se o CI pronto para operar.
- Em ② e ⑤ é dado um clear na saída. A saída borrow (L5) fica em nível “Ø” indicando que a contagem zerou.

- Em ③ e ④ é feito o carregamento de um número, colocando-o disponível na saída do contador através da atuação no pino “Load” (Chave F). Ao se carregar o número 1111 (em ④) o carry (LØ) fica em nível “Ø”, indicando que a contagem atingiu o valor máximo.

**d)** Completar a tabela 13.

ENTRADA				CLOCK		SAÍDAS							
D	C	B	A	Clear	Load	Down	UP	Borrow	Carry	QD	QC	QB	QA
A	B	C	D	E	F	G	H	L5	L4	L3	L2	L1	LØ
X	X	X	X	Ø	1	1	1						
Ø	1	1	Ø	Ø		1	1						
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							
X	X	X	X	Ø	1	1							

**Tabela 13 - Tabela para o circuito da figura 15.**

Nesta tabela, mostra-se a contagem crescente:

- Em ①, carrega-se o número Ø11Ø (6).
- Em ②, faz-se a contagem até atingir 1111.
- Em ③, após a contagem atingir 1111, assim que o *clock up* (H) for para zero, nota-se o aparecimento da indicação de carry (LØ = Ø).
- Em ④, assim que o *clock up* voltar a “1”, a contagem reinicia em ØØØØ.

e) Completar a tabela 14.

ENTRADA				CLOCK		SAÍDAS								
D	C	B	A	Clear	Load	Down	UP	Borrow	Carry	QD	QC	QB	QA	
A	B	C	D	E	F	G	H	L5	L4	L3	L2	L1	LØ	
X	X	X	X	Ø	1	1	1							①
X	X	X	X		1	1	1							②
X	X	X	X	Ø	1	Ø	1							③
X	X	X	X	Ø	1	1								
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							
X	X	X	X	Ø	1		1							

**Tabela 14 - Tabela para o circuito da figura 15.**

Nesta tabela, mostra-se a contagem decrescente:

- Em ①, faz-se o *clear* das saídas.
- Em ②, com a saída em ØØØØ, fazendo-se o *clock down* (G) igual a “Ø”, aparece a indicação de zeragem (L5 = Ø).
- Em ③, com o *clock down* voltando a “1”, todas as saídas vão para “1”, iniciando a contagem decrescente.
- Em ④, faz-se a contagem decrescente.

## **CAPÍTULO 4 - REGISTRADORES**

### **4.1 RESUMO**

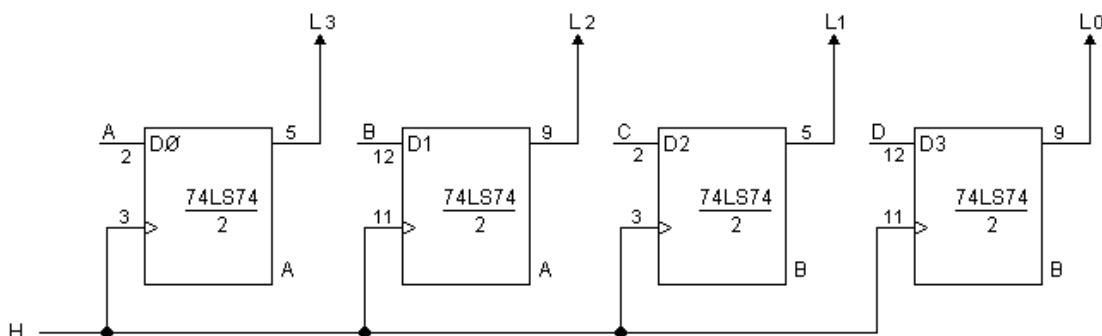
Um registrador (ou registro) é um grupo de elementos de memória que trabalham em conjunto como uma única unidade. Os registradores mais simples guardam palavras binárias, outros modificam a palavra guardada, somando 1 ou deslocando bits para a direita ou para a esquerda ou, ainda, executando outras operações.

### **4.2 REGISTRADORES BUFFER**

O registrador buffer é um dos mais simples; tudo o que faz é simplesmente guardar temporariamente uma palavra binária.

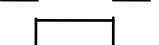
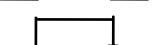
#### **4.2.1 EXPERIÊNCIA 1**

- a) Material Utilizado: 2 x CI's 74LS74
- b) Montar o circuito da figura 1.



**Figura 1 - Registrador de 4 bits.**

- c) Completar a tabela 1, lembrando que o clock deve ser ativado após selecionadas as entradas A, B, C e D.

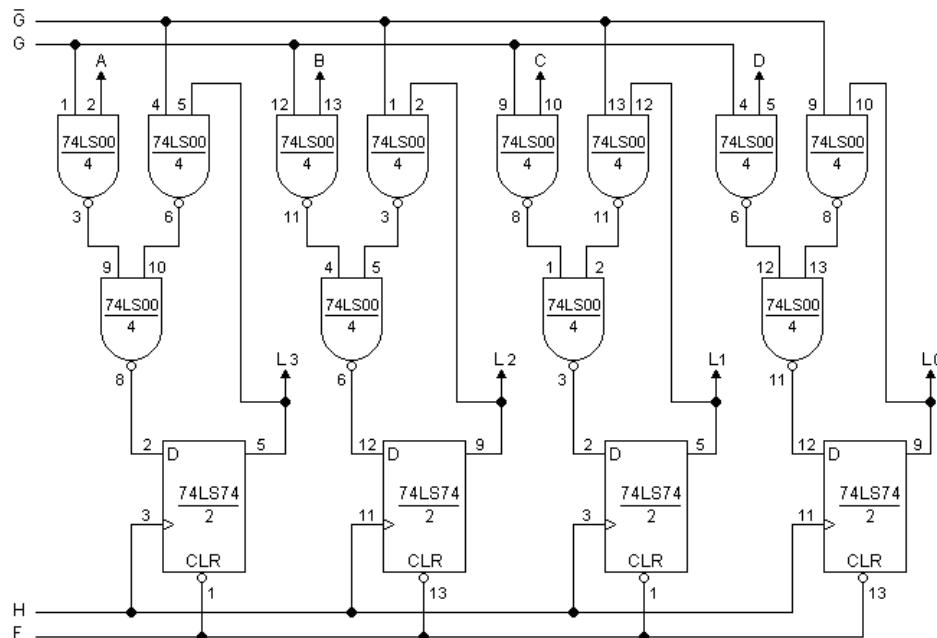
ENTRADAS				CLOCK	SAÍDAS			
A	B	C	D	H	L3	L2	L1	LØ
Ø	Ø	Ø	Ø	Ø				
Ø	Ø	Ø	Ø					
1	Ø	Ø	Ø					
1	1	Ø	Ø					
1	1	Ø	1					
1	1	1	1					
Ø	Ø	Ø	Ø					
Ø	1	1	Ø					
1	1	Ø	1					
Ø	1	Ø	Ø					
Ø	Ø	Ø	1					

**Tabela 1 - Tabela para o circuito da figura 1.**

- d) O circuito da figura 1 se destina a carregar no registrador, após aplicação do clock, a palavra selecionada por A, B, C e D. O circuito é muito primitivo para ser de qualquer uso, porém a seguir é apresentado melhorias neste circuito básico.

#### 4.2.2 EXPERIÊNCIA 2

- a) Material Utilizado: 3 x Cl's 74LS00 e 2 x Cl's 74LS74  
b) Montar o circuito da figura 2.



**Figura 2 - Registrador de 4 bits melhorado.**

c) Completar a tabela 2.

LOAD (CARGA)	ENTRADAS				CLOCK	CLEAR	SAÍDAS			
G	A	B	C	D	H	F	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø	Ø	Ø	Ø	Ø	Ø	Ø				
Ø	Ø	Ø	Ø	Ø		Ø				
Ø	1	Ø	1	Ø			1			
Ø	1	1	1	Ø			1			
1	1	1	1	Ø			1			
Ø	Ø	Ø	Ø	Ø			1			
1	Ø	Ø	1	1			1			
Ø	1	1	1	1			1			

**Tabela 2 - Tabela para o circuito da figura 2.**

d) Na tabela 2, observar que em:

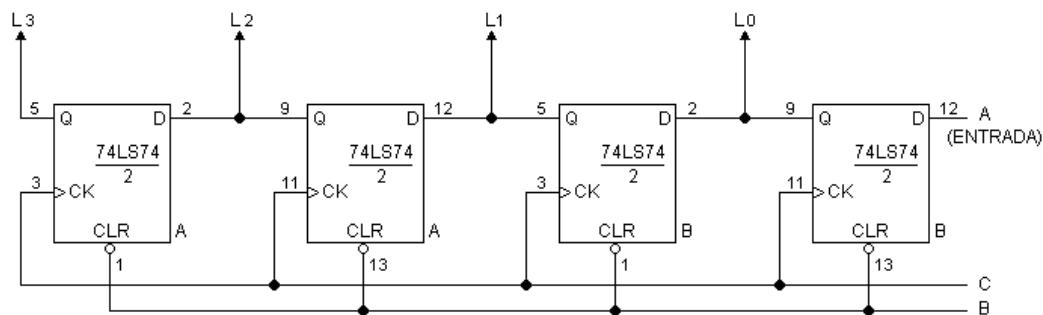
- ① tem-se  $F = \emptyset$ , logo as saídas  $L_3 L_2 L_1 L_0$  devem ser  $\emptyset\emptyset\emptyset\emptyset$ , pois  $F = \emptyset$  significa clear.
- ② com  $G = \emptyset$  não se pode carregar o registrador.
- ③ e ④ com  $G = 1$ , o registrador deve ser carregado com as informações contidas em A B C D, porém com  $G = \emptyset$  a informação anterior é mantida no registrador.

### 4.3 REGISTRADORES DE DESLOCAMENTO

Os registradores de deslocamento ou *shift registers* deslocam os bits guardados para a direita ou para a esquerda, dependendo do seu projeto.

#### 4.3.1 EXPERIÊNCIA 3

- Material Utilizado: 2 x Cl's 74LS74
- Montar o circuito da figura 3.



**Figura 3 - Registrador de deslocamento a esquerda.**

- Completar a tabela 3.

CLEAR	ENTRADA	CLOCK	SAÍDAS			
			L3	L2	L1	LØ
Ø	x	x				
1	1					
1	1					
1	1					
1	1					
1	Ø					
1	Ø					
1	Ø					
1	Ø					

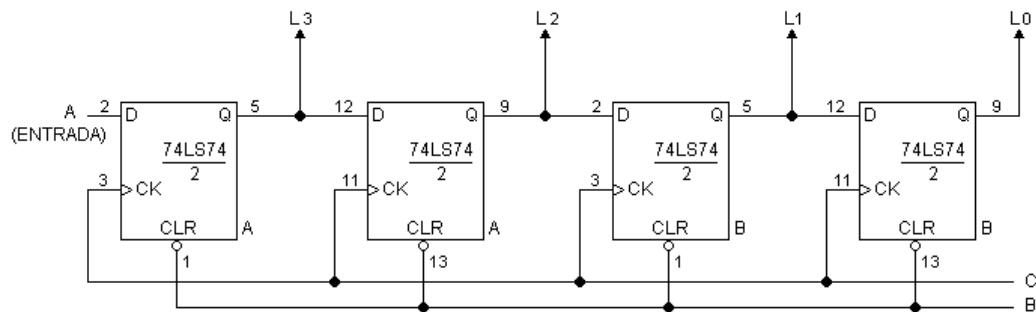
①  
②  
③

**Tabela 3 - Tabela para o circuito da figura 3**

- Na tabela 3 observar que em ① foi feito clear no registrador, e em ② e ③ há o deslocamento para a esquerda de um bit 1 e um bit Ø, respectivamente. Este circuito é então um registrador de deslocamento à esquerda.

### 4.3.2 EXPERIÊNCIA 4

- a) Material Utilizado: 2 x CI's 74LS74
- b) Montar o circuito da figura 4.



**Figura 4 - Registrador de deslocamento à direita.**

- c) Completar a tabela 4.

CLEAR	ENTRADA	CLOCK	SAÍDAS			
			L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
B	A	C				
Ø	x	x				
1	1					
1	1					
1	1					
1	1					
1	Ø					
1	Ø					
1	Ø					
1	Ø					

①

②

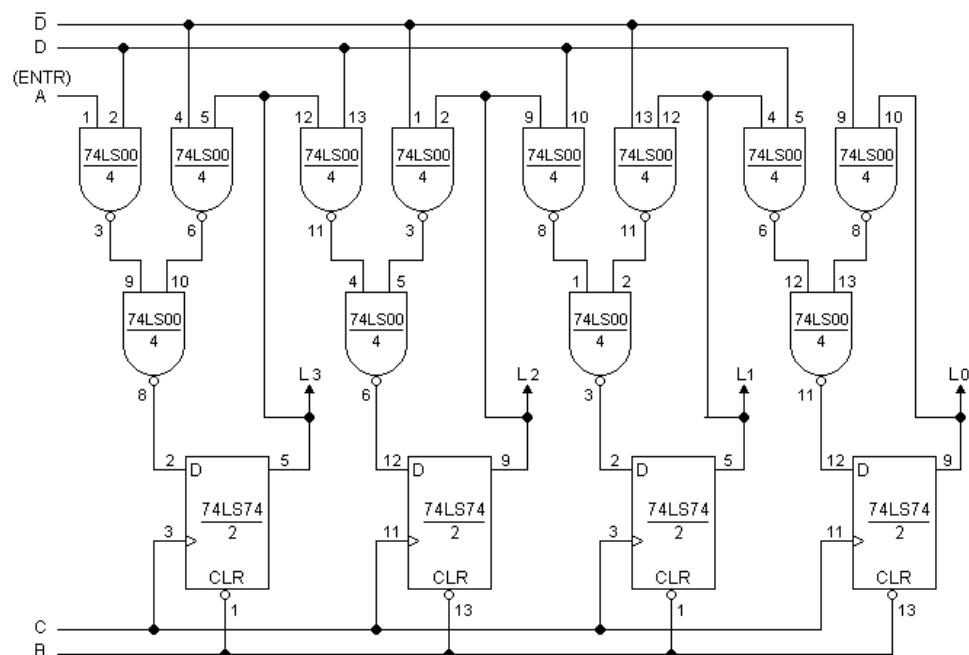
③

**Tabela 4 - Tabela para o circuito da figura 4.**

- d) Na tabela 4 observar que em ① foi feito clear no registrador, e em ② e ③ há o deslocamento para a direita de um bit 1 e um bit Ø, respectivamente. Assim, o circuito da figura 4 é um registrador de deslocamento à direita.

### 4.3.3 EXPERIÊNCIA 5

- a) Material Utilizado: 3 x CI's 74LS00 e 2 x CI's 74LS74
- b) Montar o circuito da figura 5.

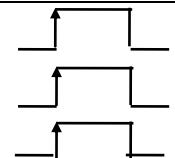


**Figura 5 - Registrador de deslocamento controlado.**

- c) Completar a tabela 5.

CLEAR	DESLOCAMENTO	ENTRADA	CLOCK	SAÍDAS
B	D	A	C	L <sub>3</sub> L <sub>2</sub> L <sub>1</sub> L <sub>Ø</sub>
Ø	x	x	x	
1	Ø	1	[Clock pulse]	
1	1	1	[Clock pulse]	
1	1	1	[Clock pulse]	
1	1	1	[Clock pulse]	
1	1	Ø	[Clock pulse]	
1	Ø	Ø	[Clock pulse]	
1	Ø	Ø	[Clock pulse]	

①  
②  
③  
④

CLEAR	DESLOCAMENTO	ENTRADA	CLOCK	SAÍDAS
B	D	A	C	$L_3$ $L_2$ $L_1$ $L_\emptyset$
1	1	$\emptyset$		
1	1	$\emptyset$		
1	1	$\emptyset$		

**Tabela 5 - Tabela do circuito da figura 5.**

d) Na tabela 5 observar que em:

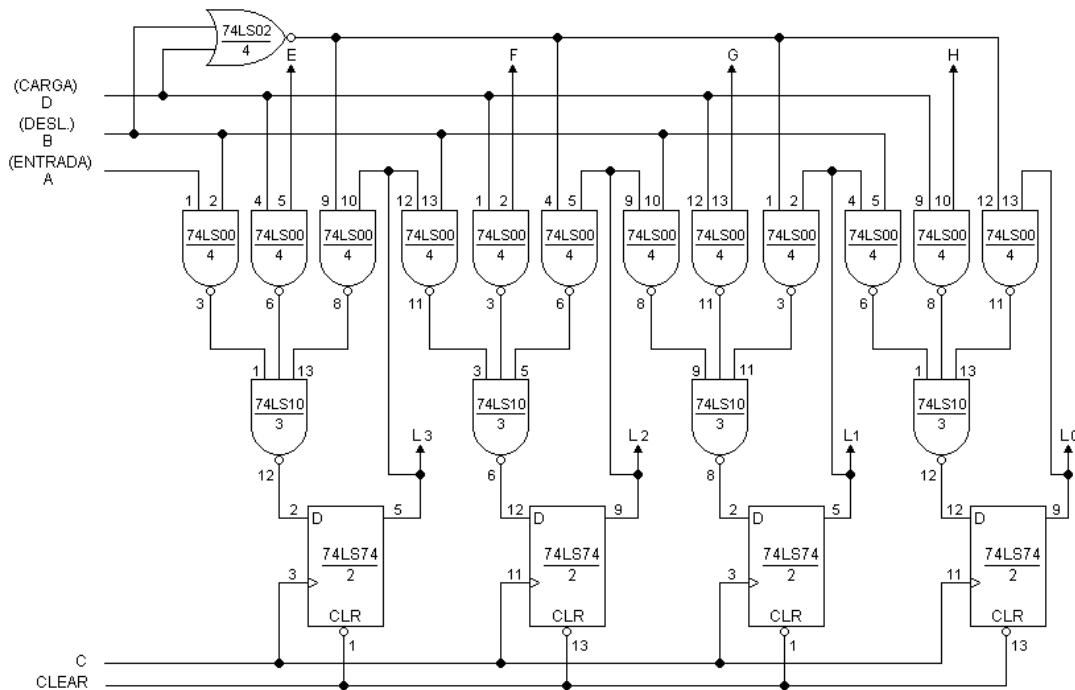
- ①  $B = \emptyset$ , o que significa clear do registrador.
- ②  $D = \emptyset$ , não há deslocamento. No clock os dados do registrador serão mantidos.
- ③  $D = 1$ , há deslocamento. O registrador no 1º clock recebe o bit da entrada “A”, no seguinte clock o bit  $L_3$  é deslocado para  $L_2$  e o segundo bit de “A” entra em  $L_3$ , e assim por diante.
- ④ os dados do registrador são mantidos, pois  $D = \emptyset$ .

#### 4.4 CARREGAMENTO PARALELO

Na experiência seguinte é mostrado outro passo da evolução dos registradores de deslocamento. Além da entrada CLR, pode-se carregar os bits diretamente nos flip-flops, do mesmo modo que num registrador buffer. Esta espécie de entrada de dados é chamada carregamento paralelo.

##### 4.4.1 EXPERIÊNCIA 6: (OPCIONAL)

- a) Material Utilizado: 3 x CI's 74LS00, 1 x CI 74LS02, 2 x CI's 74LS10 e 2 x CI's 74LS74
- b) Montar o circuito da figura 6.



**Figura 6 - Registrador de deslocamento com carregamento paralelo.**

c) Completar a tabela 6.

CLEAR	DESLOCAMENTO	CARGA	ENTRADAS					CLOCK	SAÍDAS			
VCC/COM	B	D	A	E	F	G	H	C	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø	x	x	x	x	x	x	x	x				
1	Ø	Ø	x	x	x	x	x					
1	Ø	1	x	1	Ø	1	Ø					
1	Ø	1	x	1	1	Ø	Ø					
1	Ø	1	x	Ø	Ø	Ø	Ø					
1	1	Ø	1	Ø	Ø	Ø	Ø					
1	1	Ø	Ø	1	Ø	Ø	Ø					
1	1	Ø	Ø	1	Ø	Ø	Ø					
1	1	Ø	Ø	Ø	1	Ø	Ø					
1	1	1	—	—	—	—	—	—	—	—	—	—

**Tabela 6 - Tabela para circuito da figura 6.**

d) Na tabela 6 observar que em:

- ① clear =  $\emptyset$  (comum), é dado um clear nas saídas, portanto  $L_3, L_2, L_1$  e  $L_\emptyset = \emptyset\emptyset\emptyset\emptyset$ .
- ②  $B = \emptyset$  e  $D = \emptyset$ , os dados do registrador são mantidos após o clock (recirculados).
- ③  $B = \emptyset$  e  $D = 1$ , o registrador é carregado com os dados contidos nas entradas EFGH.
- ④  $B = 1$  e  $D = \emptyset$ , o conteúdo do registrador é deslocado a cada clock, com a entrada A aparecendo na saída  $L_3$ .

#### 4.5 REGISTRADORES NA FORMA DE CI's

Na série 74 existem alguns registradores na forma de CI's, entre os quais pode-se citar: o 74164, que é um registrador de deslocamento entrada série, saída paralelo; o 74165, que é um registrador de deslocamento, entrada paralela, saída série; e o 74194, que é um registrador de deslocamento universal, no qual os dados podem entrar em série ou em paralelo e sair em paralelo ou em série.

Em computadores, circuitos baseados em registradores de deslocamento com entrada paralela e saída série, são responsáveis pela transmissão serial de dados, enquanto que circuitos baseados em registradores de deslocamento com entrada série e saída paralela, são responsáveis pela recepção serial de dados.

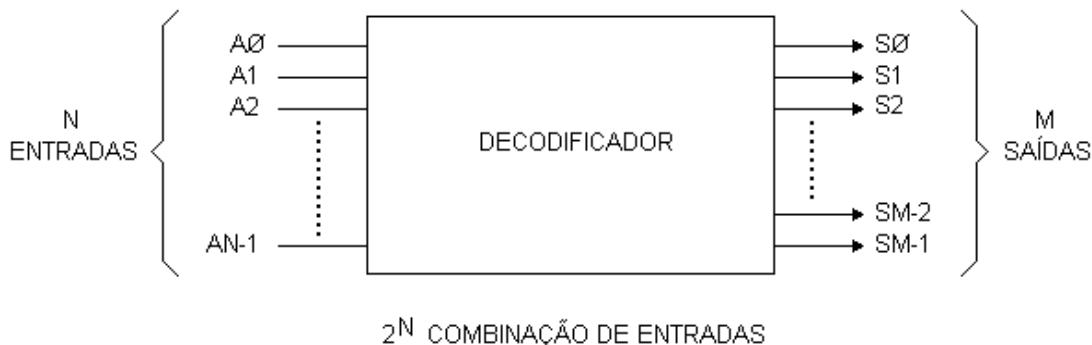


## **CAPÍTULO 5 - DECODIFICADORES**

### **5.1 RESUMO**

Um decodificador é um circuito que tem N entradas binárias e M linhas de saídas (sendo  $M \leq 2^N$ ), de modo que somente uma linha de saída M é ativada por vez, para cada combinação das N entradas.

A figura 1 mostra o diagrama geral de um decodificador com N entradas e M saídas.



**Figura 1 - Decodificador geral com N entradas e M saídas**

Alguns decodificadores não utilizam todas  $2^N$  combinações de entradas de código possíveis, mas apenas algumas destas. Decodificadores deste tipo são frequentemente, projetados de modo que se qualquer dos códigos não usados for aplicado as entradas, nenhuma das saídas será ativada.

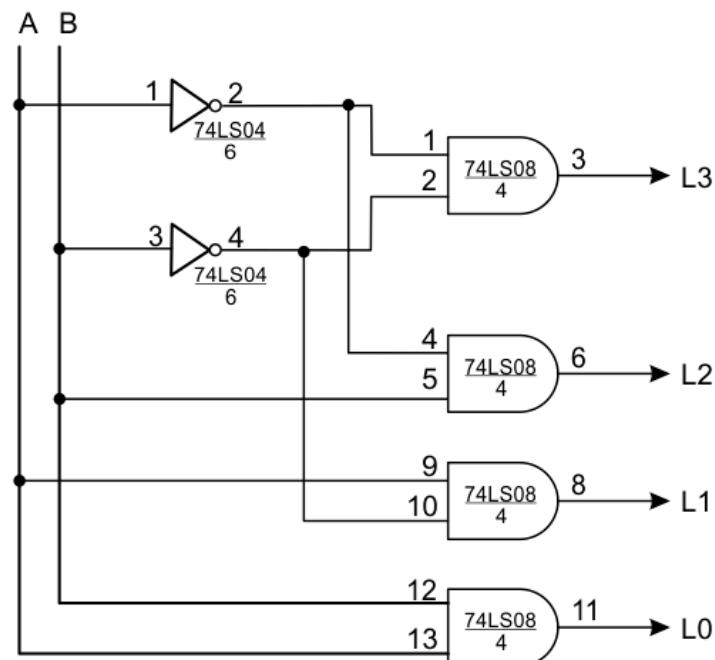
A tabela 1 mostra um decodificador de 2 para 4 linhas.

ENTRADAS		SAÍDAS			
A	B	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø	Ø	1	Ø	Ø	Ø
Ø	1	Ø	1	Ø	Ø
1	Ø	Ø	Ø	1	Ø
1	1	Ø	Ø	Ø	1

**Tabela 1 - Decodificador de 2 entradas e 4 saídas.**

## 5.2 EXPERIÊNCIA 1

- Material Utilizado: 1 x CI 74LS04 e 1 x CI 74LS08
- Montar o circuito da figura 2.



**Figura 2 - Decodificador de 2 entradas e 4 saídas com Gates.**

- Completar a tabela 2.

ENTRADAS		SAÍDAS			
A	B	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
Ø	Ø				
Ø	1				
1	Ø				
1	1				

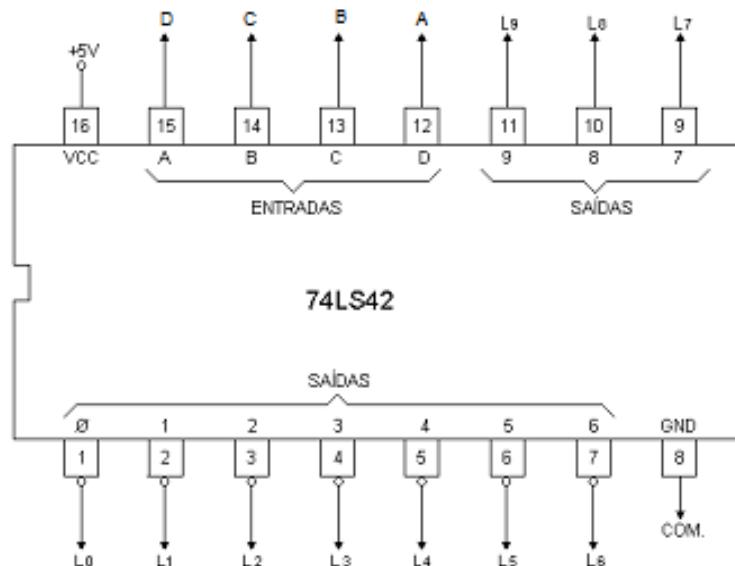
**Tabela 2 - Tabela do circuito da figura 2.**

## 5.3 DECODIFICADOR BCD PARA DECIMAL

Um decodificador que tem 4 entradas no código BCD e produz 10 saídas correspondendo aos dígitos decimais é chamado **Decodificador (ou conversor) BCD para decimal, ou Decodificador 1 de 10**.

### 5.3.1 EXPERIÊNCIA 2

- a)** Material Utilizado: 1 x CI 74LS42  
**b)** Montar o circuito da figura 3.



**Figura 3 - Decodificador 74LS42.**

- c)** Completar a tabela 3.

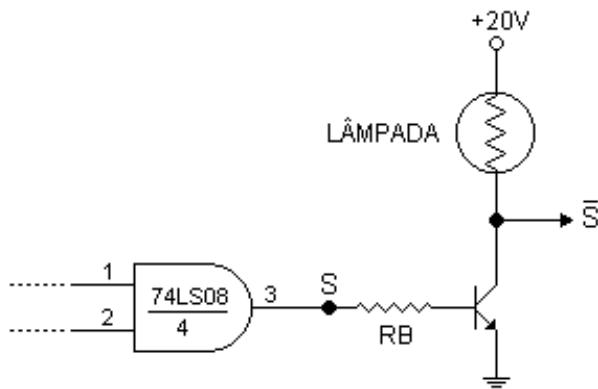
ENTRADAS				SAÍDAS									
D	C	B	A	9	8	7	6	5	4	3	2	1	Ø
CHAVES				L <sub>9</sub>	L <sub>8</sub>	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
Ø	Ø	Ø	Ø										
Ø	Ø	Ø	1										
Ø	Ø	1	Ø										
Ø	Ø	1	1										
Ø	1	Ø	Ø										
Ø	1	Ø	1										
Ø	1	1	Ø										
Ø	1	1	1										
1	Ø	Ø	Ø										
1	Ø	Ø	1										
①	1	Ø	1	Ø									
	1	Ø	1	1									
②	1	1	Ø	Ø									
	1	1	Ø	1									

**Tabela 3 - 74LS42.**

- d) Observar que em ① uma única saída é  $\emptyset$  de cada vez. Na prática, este decodificador pode ser usado para habilitar um circuito externo que requeira uma única entrada para cada saída. As combinações de entradas de ② são combinações que não devem ser usadas e neste caso nenhuma saída é  $\emptyset$ .

#### 5.4 DECODIFICADOR DRIVER

O decodificador da figura 3 pode ser modificado para ter saída com maiores correntes e tensões. Isto está mostrado na figura 4, onde a saída do decodificador atua na base do transistor, o qual tem coletor aberto e é capaz de operar correntes e tensões relativamente altas. A figura 4 mostra uma conexão típica para alimentar uma pequena lâmpada incandescente de 20 volts. Deve-se observar que a saída do transistor fica  $\bar{S}$ , já que este age como um inversor.



**Figura 4 - Uma das saídas do decodificador com transistor driver.**

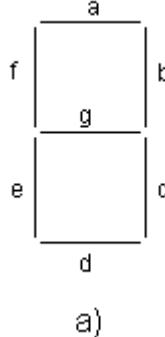
Na família TTL 74 já existem circuitos de coletor aberto que são indicados para se usar na saída  $s$ , em substituição ao transistor. Entre estes, pode-se citar o CI 74LS06, que consta de 6 inversores *Buffers/Drivers* com saída para alta tensão (até 30 volts) e podendo absorver correntes de até 40mA. Outro destes é o CI 7407, que consta de 6 *Buffers/Drivers* com saída possuindo características de tensão e corrente iguais as do 74LS06.

Existem também CI's decodificadores, na família 74, que já vêm com o driver incorporado e, nestes casos, são denominados **Decodificadores/Drivers**. Entre estes pode-se citar o 74141, que é um decodificador/driver BCD para decimal.

#### 5.4.1 DECODIFICADOR/DRIVERS BCD PARA 7 SEGMENTOS

Alguns displays numéricos usam uma configuração de 7 segmentos para produzir um caractere alfanumérico. Cada segmento é composto de um material que emite luz quando percorrido por corrente. Os materiais mais comumente utilizados são diodos de emissão de luz (LED's) e filamentos incandescentes.

Um decodificador/driver BCD para 7 segmentos recebe entradas BCD de 4 bits e fornece as saídas que conduzirão as correntes, através dos segmentos apropriados para mostrar o caractere alfanumérico. A tabela 4 mostra os segmentos acesos com os respectivos dígitos decimais.



a)

SEGMENTOS ACESSOS							DÍGITO DECIMAL
a	b	c	d	e	f	g	
1	1	1	1	1	1	Ø	Ø
Ø	1	1	Ø	Ø	Ø	Ø	1
1	1	Ø	1	1	Ø	1	2
1	1	1	1	Ø	Ø	1	3
Ø	1	1	Ø	Ø	1	1	4
1	Ø	1	1	Ø	1	1	5
Ø	Ø	1	1	1	1	1	6
1	1	1	Ø	Ø	Ø	Ø	7
1	1	1	1	1	1	1	8
1	1	1	Ø	Ø	1	1	9

b)

**Tabela 4 - Display de 7 segmentos.**

O nome decodificador é aplicado para este caso pois, apesar de se ter várias saídas ativas simultaneamente no decodificador, na saída do decodificador/driver-display só se tem um único dígito decimal.

Existem decodificadores/drivers BCD para 7 segmentos na série 74. Entre estes pode-se citar o 7446 e o 7447. Entretanto, um decodificador / driver muito popular é o 9368 e que é utilizado neste equipamento, podendo ser encontrado ou similar (7448)

O módulo 8810 contém 2 decodificadores em EPLD's ligados aos displays.

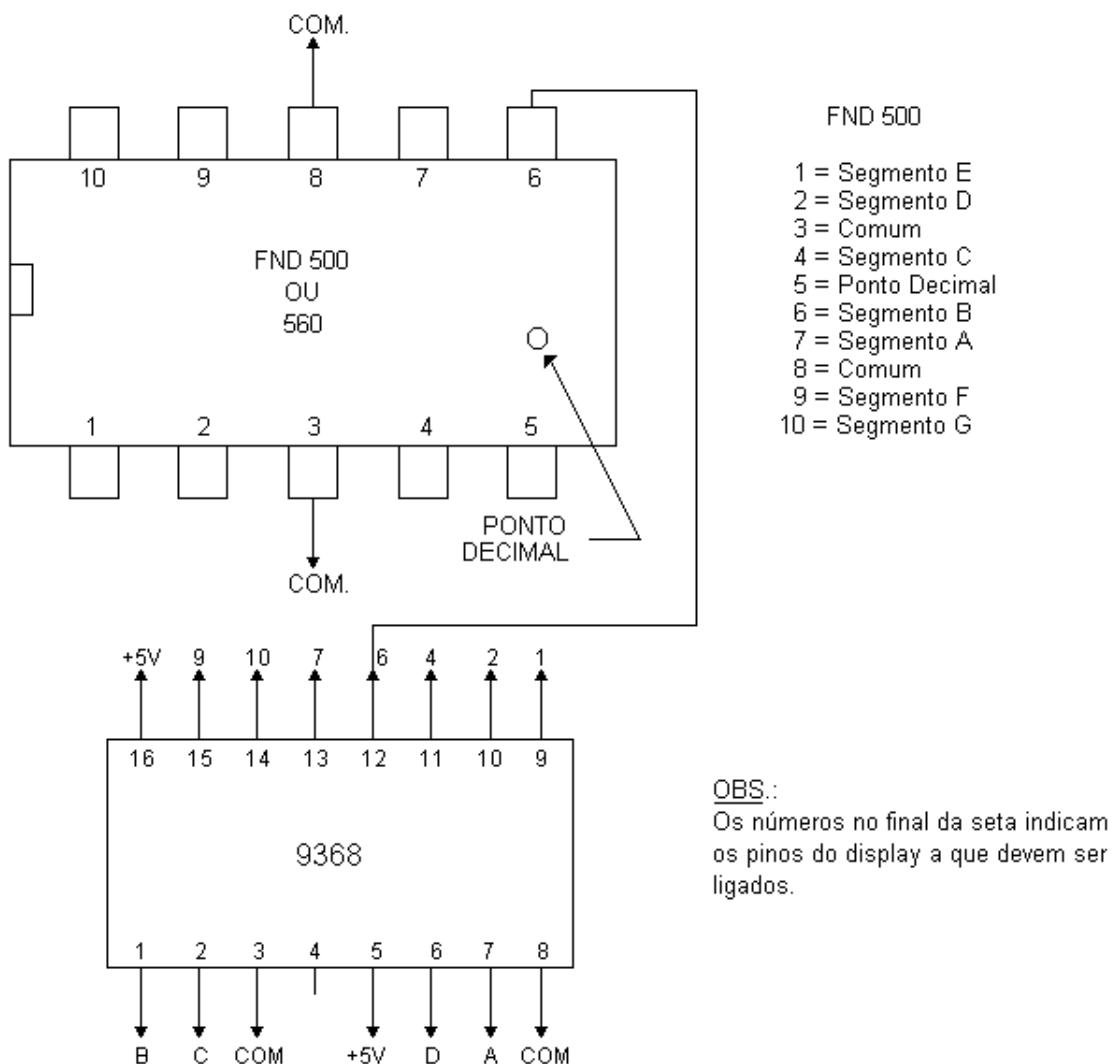
Suas respectivas entradas são:

L<sub>0</sub> a L<sub>3</sub> – dígito correspondentes aos 4 bits menos significativos

L<sub>4</sub> a L<sub>7</sub> – dígitos correspondentes aos 4 bits mais significativos

Existe display que já têm ao seu circuito o decodificador / driver.

A figura 5 mostra como é o circuito discreto equivalente ao do módulo.



**Figura 5 – Circuito decodificador / display do Módulo 8810.**

#### 5.4.1.1 EXPERIÊNCIA 3

- a) Ligar as chaves A, B, C e D em  $L_3$ ,  $L_2$ ,  $L_1$  e  $L_0$ , respectivamente.

Observação: Poderá ligar E, F, G e H em  $L_7$ ,  $L_6$ ,  $L_5$  e  $L_4$ , respectivamente.

- b) Verificar a tabela 5.

ENTRADAS				SAÍDA
A	B	C	D	Caracter Alfanumérico
Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	1	1
Ø	Ø	1	Ø	2
Ø	Ø	1	1	3
Ø	1	Ø	Ø	4
Ø	1	Ø	1	5
Ø	1	1	Ø	6
Ø	1	1	1	7
1	Ø	Ø	Ø	8
1	Ø	Ø	1	9
1	Ø	1	Ø	A
1	Ø	1	1	B
1	1	Ø	Ø	C
1	1	Ø	1	D
1	1	1	Ø	E
1	1	1	1	F

①

②

**Tabela 5 - 9368 e FND 500.**

c) Observar que:

- Em ① são os decimais correspondentes às entradas.
- Em ② as saídas são os hexadecimais correspondentes às entradas.

Disto conclui-se que:

1º - Usando-se entradas desde ØØØØ até 1001 o 9368 funciona como decodificador / driver BCD para decimal.

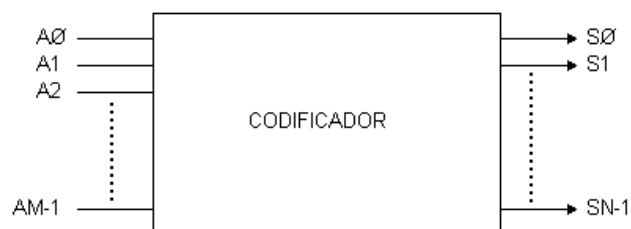
2º - Usando-se entradas desde ØØØØ até 1111 o 9368 funciona como decodificador / driver binário para hexadecimal.



## **CAPÍTULO 6 - CODIFICADORES**

### **6.1 RESUMO**

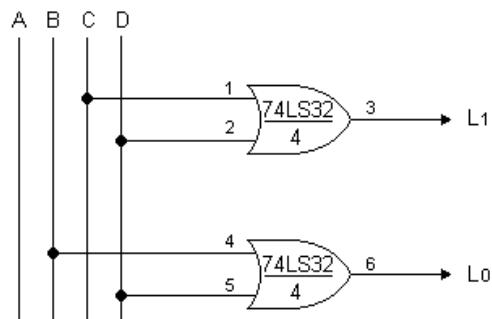
Um codificador executa o processo inverso do decodificador, isto é, um codificador é um circuito que tem um número “M” de linhas de entradas, sendo que somente uma delas é ativada por vez e produz um código de saída de “N” bits. O código de saída depende de que entrada é ativada. A figura 1 é o diagrama geral de um codificador com “M” entradas e “N” saídas.



**Figura 1 - Diagrama geral de um codificador de M entradas e N saídas.**

### **6.2 EXPERIÊNCIA 1**

- Material Utilizado: 1 x CI 74LS32
- Montar o circuito da figura 2, lembrando que a chave “A” só seria usada para circuitos maiores.



**Figura 2 - Codificador de 4 entradas e 2 saídas.**

- c) Completar a tabela 1.

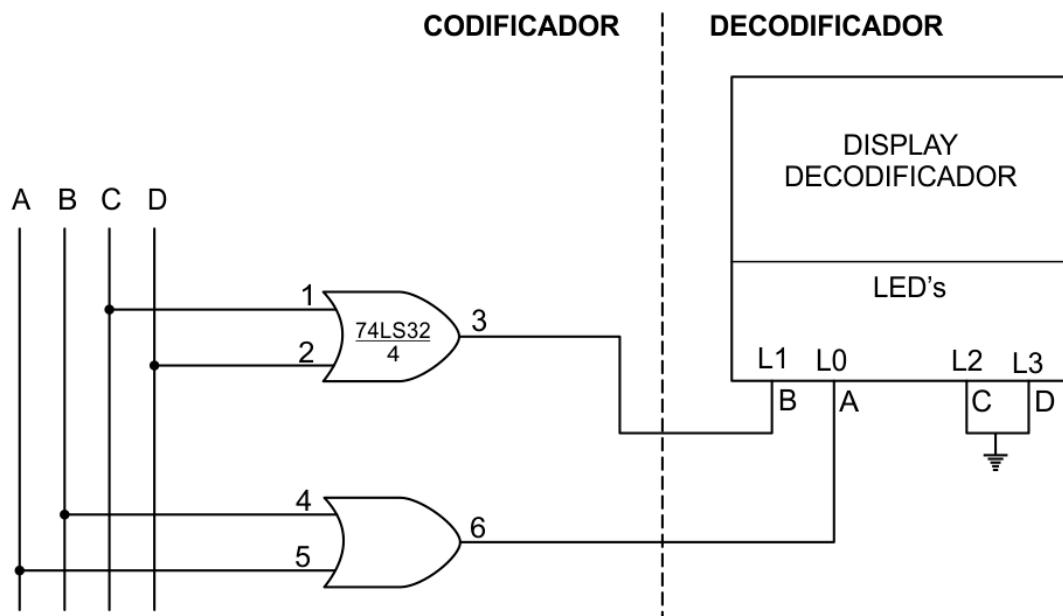
ENTRADAS				SAÍDAS	
A	B	C	D	L <sub>1</sub>	L <sub>0</sub>
1	Ø	Ø	Ø		
Ø	1	Ø	Ø		
Ø	Ø	1	Ø		
Ø	Ø	Ø	1		

**Tabela 1 - Codificador de 4 entradas e 2 saídas.**

- d) Na tabela 1, se mais de uma entrada fosse 1 num dado instante, o resultado da saída seria errado. Se nenhuma entrada for 1, as saídas fornecerão ØØ.

### 6.3 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74LS32
- b) Montar o circuito da figura 3, lembrando que a chave só seria usada para circuitos maiores.



**Figura 3 - Codificador/decodificador de chaves para displays de sete segmentos, com 4 entradas.**

- c) Completar a tabela 2

ENTRADAS				CARACTER GERADO
A	B	C	D	
1	Ø	Ø	Ø	
Ø	1	Ø	Ø	
Ø	Ø	1	Ø	
Ø	Ø	Ø	1	

**Tabela 2 - Tabela do circuito da figura 3.**

- d) Observar que cada chave corresponde a um número decimal.

#### 6.4 CHAVES CODIFICADORAS

Muitos sistemas digitais usam chaves para entrada de dados. Como exemplo tem-se as calculadoras eletrônicas, as quais usam teclado de entrada. Outro exemplo é o teclado de computadores. Nestes e em outros casos similares, a atuação das chaves deve ser codificada no código apropriado e requerido pelo sistema. Num sistema similar ao usado na figura 2, em que no lugar das entradas A B C D tenha-se as chaves do teclado, tais chaves codificadoras podem ser usadas, sempre que dados B C D tenham de entrar manualmente no sistema digital.

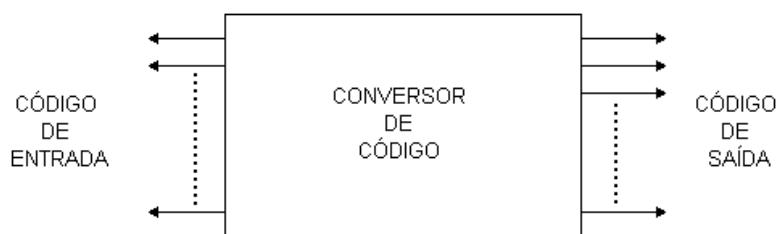
Como exemplo de seu uso pode-se citar a calculadora eletrônica, onde o operador pressiona várias chaves em sequência para entrar dados decimais. Numa calculadora simples, o código B C D para cada dígito decimal entra num registrador de 4 bits. Em outras palavras, quando a primeira chave é pressionada, o código B C D para aquele dígito é guardado num registrador de 4 bits.

Quando a segunda chave é pressionada, o código B C D para este dígito é enviado para outro registrador de 4 bits e assim sucessivamente.

Existem codificadores na forma de CI's e entre estes pode-se citar, na série TTL 74, o 74147, que é um codificador decimal para B C D, o 74148 que é um codificador, com prioridade, para oito chaves.

## 6.5 CONVERSORES DE CÓDIGOS

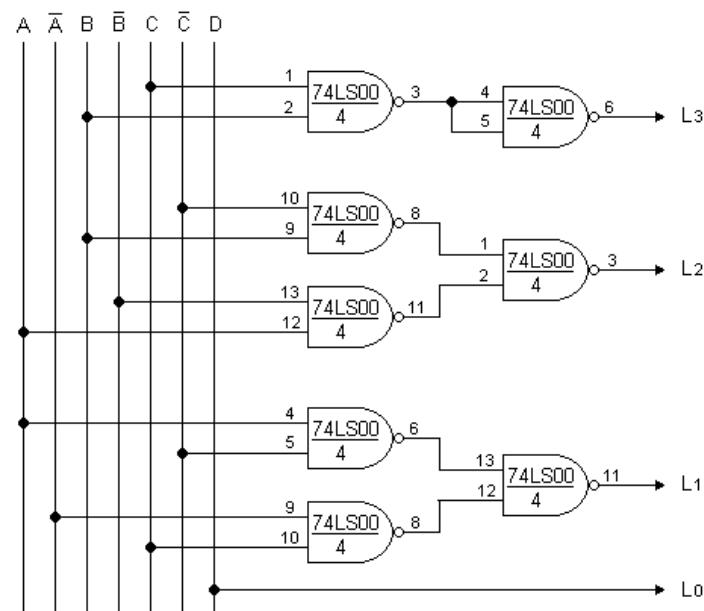
Frequentemente, em sistemas digitais é necessário converter de um código para outro. Um conversor de código (às vezes também chamado de codificador) é um dispositivo para converter um código específico, aplicado à suas entradas, em um outro código, que será fornecido nos seus terminais de saída. A figura 4 mostra o diagrama geral de um conversor de código.



**Figura 4 - Diagrama geral de um conversor de código.**

### 6.5.1 EXPERIÊNCIA 3

- a) Material Utilizado: 2 x CI's 74LS00
- b) Montar o circuito da figura 5.



**Figura 5 - Conversor de código 2421 para 8421.**

- c) Completar a tabela 3.

ENTRADAS				SAÍDAS			
A	B	C	D	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
Ø	Ø	Ø	Ø				
Ø	Ø	Ø	1				
Ø	Ø	1	Ø				
Ø	Ø	1	1				
Ø	1	Ø	Ø				
1	Ø	1	1				
1	1	Ø	Ø				
1	1	Ø	1				
1	1	1	Ø				
1	1	1	1				

**Tabela 3 - Tabela do circuito da figura 5.**

- d) Observar, pela tabela 3 que as entradas A B C D estão no código 2421 e que as saídas L<sub>3</sub>, L<sub>2</sub>, L<sub>1</sub> e L<sub>Ø</sub> estão no código 8421.

**Observação:**

Pode-se fazer outros conversores de códigos para converter de um código qualquer para outro. O procedimento lógico é simples. Contudo, já existem alguns circuitos integrados que fazem algumas das conversões mais usuais. Outras conversões podem ser feitas usando-se memórias EPROM's, ou dispositivos lógicos programáveis, tais como PLA, PAL, EPLD's, etc.

Como exemplos de circuitos integrados conversores de códigos na série 74, tem-se o 74184, que converte de B C D para binário puro, e o 74185 que converte de binário puro para B C D.



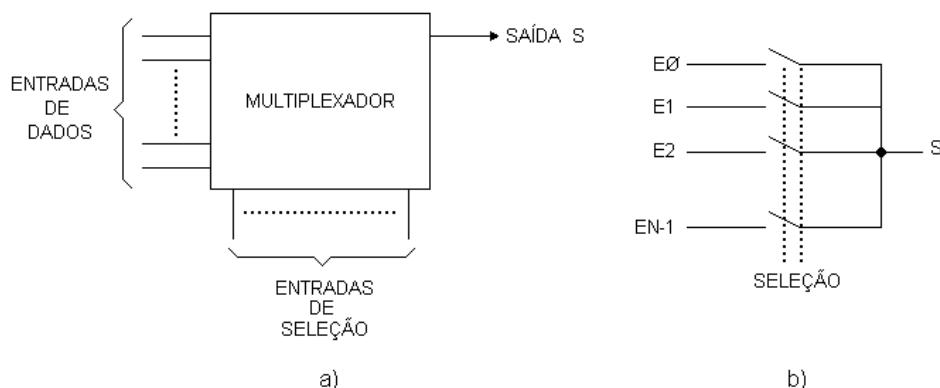
## **CAPÍTULO 7 - MULTIPLEXADORES**

### **7.1 RESUMO**

Um multiplexador ou seletor de dados é um circuito lógico que aceita diversas entradas de dados e permite que somente uma delas atinja a saída por vez.

A rota da entrada de dados desejada para saída é controlada pelas entradas de seleção (Select), algumas vezes denominadas entradas de endereço.

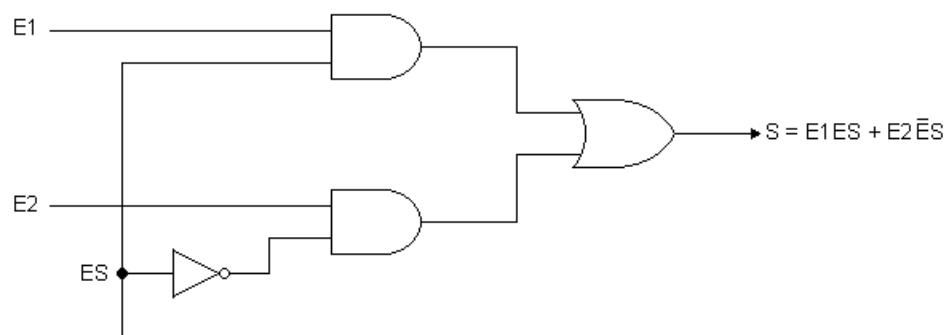
A figura 1 mostra o diagrama em bloco de um multiplexador geral e sua representação conceitual em termos de chaves.



**Figura 1 - a) Diagrama Geral de um Multiplexador digital - b) Representação conceitual do multiplexador.**

### **7.2 MULTIPLEXADOR DE 2 ENTRADAS**

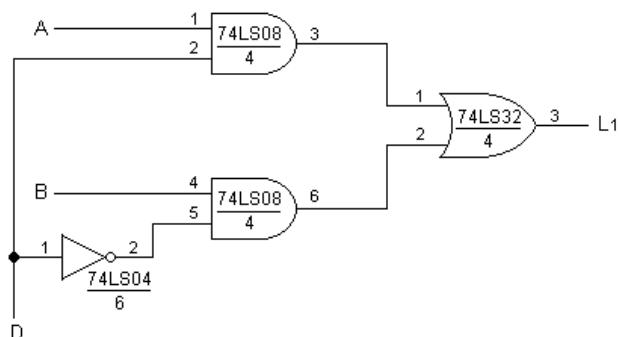
A figura 2 mostra o circuito de um multiplexador de 2 entradas de 1 bit, ou multiplexador de 2 canais de 1 bit.



**Figura 2 - Multiplexador de 2 canais de 1 bit.**

### 7.2.1 EXPERIÊNCIA 1

- a)** Material Utilizado: 1 x CI 74LS04, 1 x CI 74LS08 e 1 x CI 74LS32
- b)** Montar o circuito da figura 3.



**Figura 3 - Multiplexador de 2 canais de 1 bit**

- c)** Completar a tabela 1.

ENTRADAS		SAÍDA
DADOS	SELEÇÃO	
A	B	D
Ø	Ø	Ø
Ø	1	Ø
1	1	Ø
1	Ø	Ø
		L <sub>1</sub>
Ø	Ø	1
Ø	1	1
1	1	1
1	Ø	1

**Tabela 1 - Tabela de 1 multiplexador de 2 entradas de 1 bit.**

- d)** Observar que com  $D = \emptyset$  é selecionada a entrada “B”, e que com  $D = 1$  é selecionada a entrada “A”.

### 7.3 MULTIPLEXADORES NA FORMA DE CI's

Existem vários multiplexadores na forma de CI's, de modo que não há necessidade de construí-los a partir dos blocos básicos.

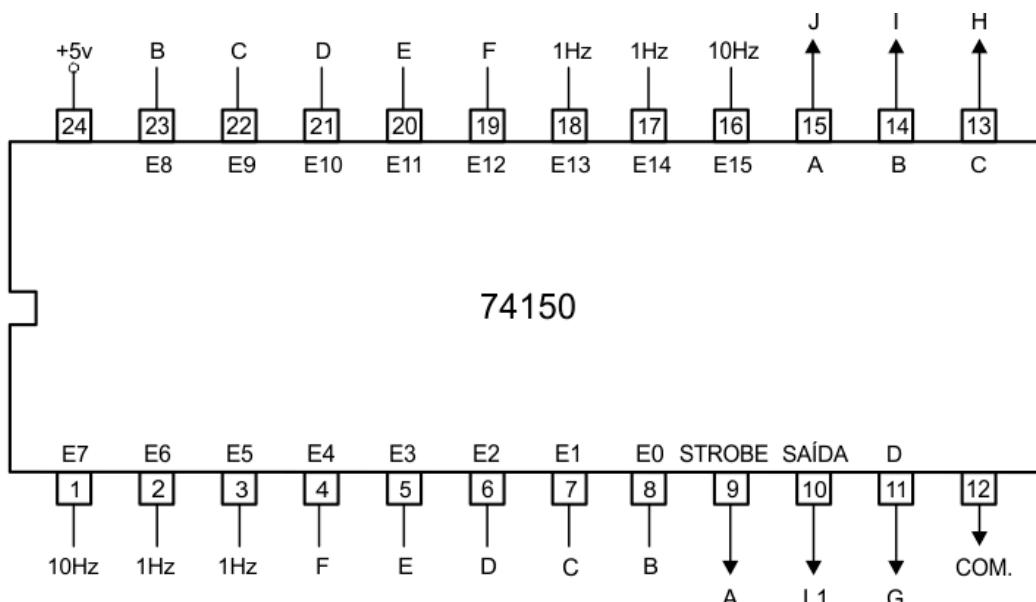
Entre estes multiplexadores, na família TTL 74, pode-se citar os seguintes:

- 74150 - Multiplexador 16 entradas de 1 bit.
- 74151 - Multiplexador 8 entradas de 1 bit.
- 74152 - Multiplexador 8 entradas de 1 bit.
- 74153 - Dois Multiplexadores de 4 entradas de 1 bit.
- 74157 - Quatro Multiplexadores de 2 entradas de 1 bit.

Outros multiplexadores existem na série 74, tais como: 74251, 74253, 74257, 74258, etc.

#### 7.3.1 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74150
- b) Montar o circuito da figura 4.



**Figura 4 - Multiplexador 74150.**

c) Completar a tabela 2.

ENTRADAS DE SELEÇÃO				STROBE	CANAL SELECIONADO	SAÍDA OBSERVADA	
G	H	I	J	A	—	L <sub>1</sub>	1
X	X	X	X	1	—		2
Ø	Ø	Ø	Ø	Ø			3
Ø	Ø	Ø	1	Ø			4
Ø	Ø	1	Ø	Ø			5
Ø	Ø	1	1	Ø			6
Ø	1	Ø	Ø	Ø			7
Ø	1	Ø	1	Ø			8
Ø	1	1	Ø	Ø			9
1	Ø	Ø	Ø	Ø			10
1	Ø	Ø	1	Ø			11
1	Ø	1	Ø	Ø			12
1	Ø	1	1	Ø			13
1	1	Ø	Ø	Ø			14
1	1	Ø	1	Ø			15
1	1	1	Ø	Ø			16
1	1	1	1	Ø			17

**Tabela 2 - Operação do CI 74150.**

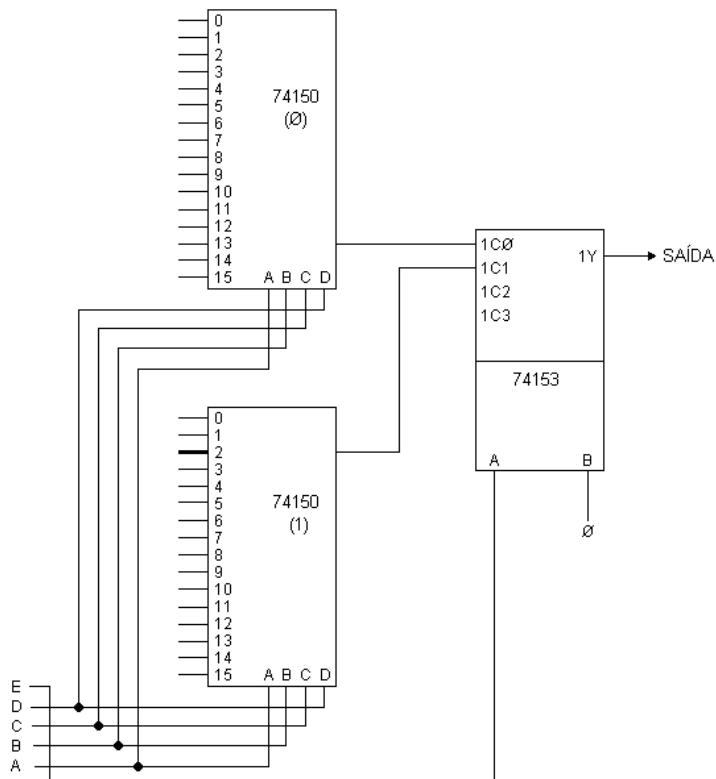
#### Observações:

1. Com STROBE em 1, independente das entradas de seleção, ou da entrada de dados, a saída L<sub>1</sub> é 1.
2. Foi selecionado o canal EØ, portanto, a saída acompanha o valor da chave “B”. Note que a saída é invertida em relação ao sinal de entrada.
3. Foi selecionado o canal E1, portanto, a saída acompanha o valor da chave “C” (invertida).
4. Foi selecionado o canal E2, portanto, a saída acompanha o valor da chave “D” (invertida).
5. Foi selecionado o canal E3, logo L<sub>1</sub> =  $\bar{E}$ .
6. L<sub>1</sub> =  $\bar{F}$
7. L<sub>1</sub> = 1Hz
8. L<sub>1</sub> = 1Hz
9. L<sub>1</sub> = 10Hz

10.  $L_1 = \bar{B}$
11.  $L_1 = \bar{C}$
12.  $L_1 = \bar{D}$
13.  $L_1 = \bar{E}$
14.  $L_1 = \bar{F}$
15.  $L_1 = 1\text{Hz}$
16.  $L_1 = 10\text{Hz}$
17.  $L_1 = 100\text{Hz}$

#### **7.4 USO DE CI's MULTIPLEXADORES PARA FORMAR MULTIPLEXADORES MAIORES**

A figura 5 é o circuito de um multiplexador de 32 canais de 1 bit, formado com multiplexadores menores (de 16 canais e 1 bit). Para se selecionar uma entrada entre as 32 é necessário que  $2^{n^0}$  linhas de seleção = 32; assim 5 linhas de seleção.



**Figura 5 - Multiplexador de 32 canais de 1 bit.**

As linhas de seleção menos significativas (D, C, B, A) são aplicadas aos 2 multiplexadores 74150. As saídas dos 74150 são conectadas a uma seção do 74153. A linha de seleção restante (E) é conectada a entrada menos significativa (A) do 74153, que seleciona a saída de qual multiplexador deverá aparecer na saída do 74153.

No circuito da figura 5, para se selecionar uma entrada entre as 32 procede-se de modo como a seguir: "**Selecionar a entrada 18.**" Transformando-se 18 em binário obtém-se: 10010 para E D C B A, respectivamente. Os 4 bits menos significativos (D C B A = ØØ1Ø) selecionam as entradas 2 dos multiplexadores 74150, e o bit mais significativo (E = 1) seleciona o 74150 nº 1. Então, a linha selecionada é a 3<sup>a</sup> entrada (entrada 18) do multiplexador nº 1. Na figura 5, o caminho do dado selecionado está mostrado por uma linha mais grossa.

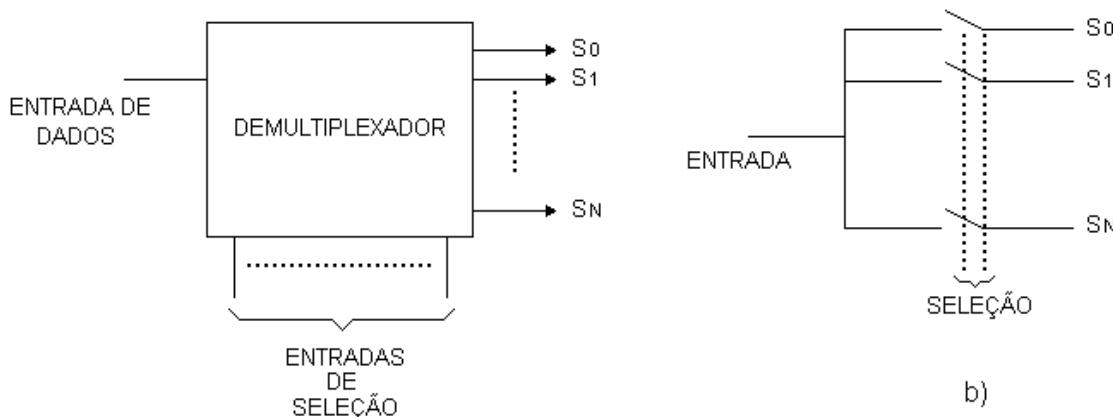
Nos sistemas de telefonia digital, os multiplexadores são usados para conectar um canal telefônico digital na linha de transmissão. Através de multiplexagem no tempo, todos os canais ocuparão um intervalo, na linha de transmissão.

## **CAPÍTULO 8 - DEMULTIPLEXADORES**

### **8.1 RESUMO**

Um demultiplexador executa a operação inversa do multiplexador, isto é, toma uma única entrada e a distribui para uma das diversas saídas.

A figura 1 mostra o diagrama em bloco de um demultiplexador e sua representação conceitual em termos de chaves.



**Figura 1 - Demultiplexador de 1 para 4 linhas.**

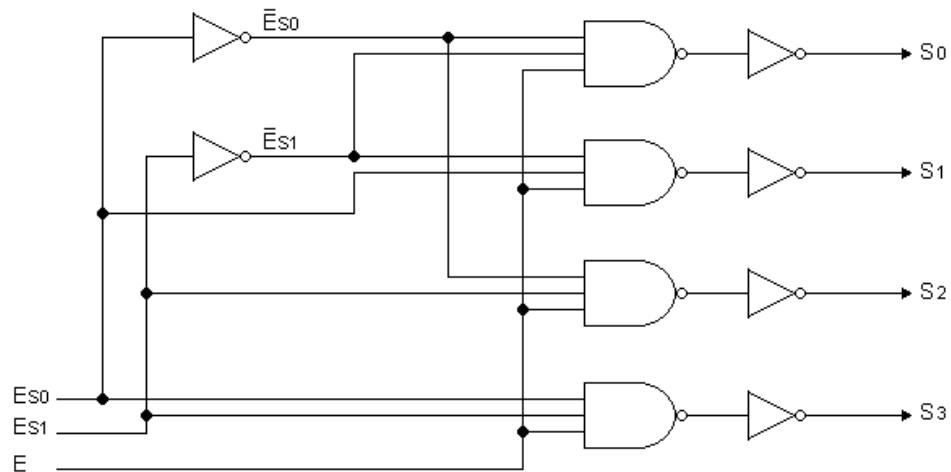
A tabela 1 mostra um demultiplexador que distribui uma linha de entrada para 4 linhas de saídas. Desta tabela, pode-se obter as equações das saídas aplicando-se a conhecida fórmula de interpolação.

ENTRADAS			SAÍDA SELECIONADA			
DADOS	SELEÇÃO		S3	S2	S1	S0
E	Ø	Ø	Ø	Ø	Ø	1
E	Ø	1	Ø	Ø	1	Ø
E	1	Ø	Ø	1	Ø	Ø
E	1	1	1	Ø	Ø	Ø

**Tabela 1 – Demultiplexador de 1 para 4 linhas.**

## 8.2 DEMULTIPLEXADOR DE 4 SAÍDAS

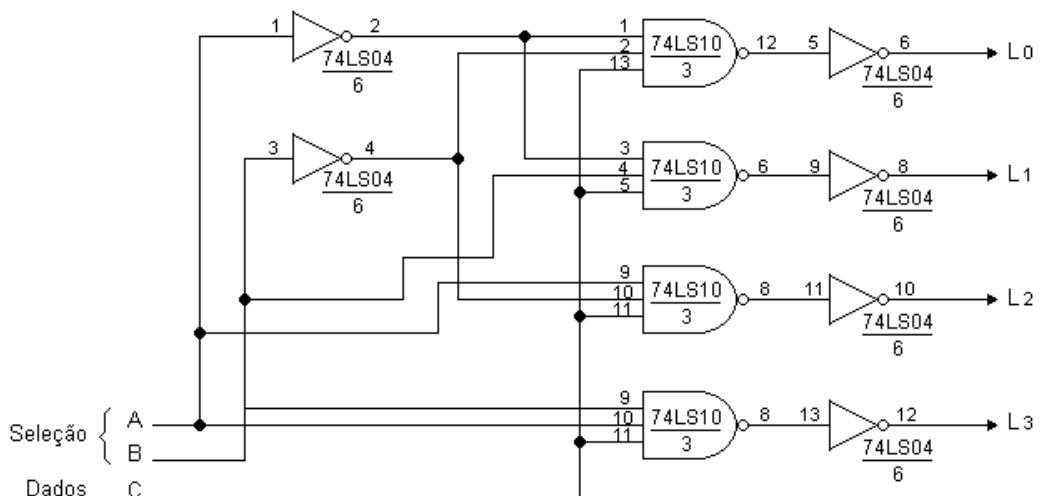
A figura 2 mostra o circuito, em termos de blocos lógicos, do demultiplexador da tabela 1.



**Figura 2 - Demultiplexador de 1 para 4 linhas, em termos de blocos lógicos.**

### 8.2.1 EXPERIÊNCIA 1

- Material Utilizado: 1 x CI 74LS10 e 1 x CI 74LS04
- Montar o circuito da figura 3.



**Figura 3 - Demultiplexador de 4 saídas.**

- Completar a tabela 2.

ENTRADAS			SAÍDA SELECIONADA			
DADOS	SELEÇÃO					
C	A	B	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
Ø	Ø	Ø				
1	Ø	Ø				
Ø	Ø	1				
1	Ø	1				
Ø	1	Ø				
1	1	Ø				
Ø	1	1				
1	1	1				

**Tabela 2 - Demultiplexador de 4 saídas.**

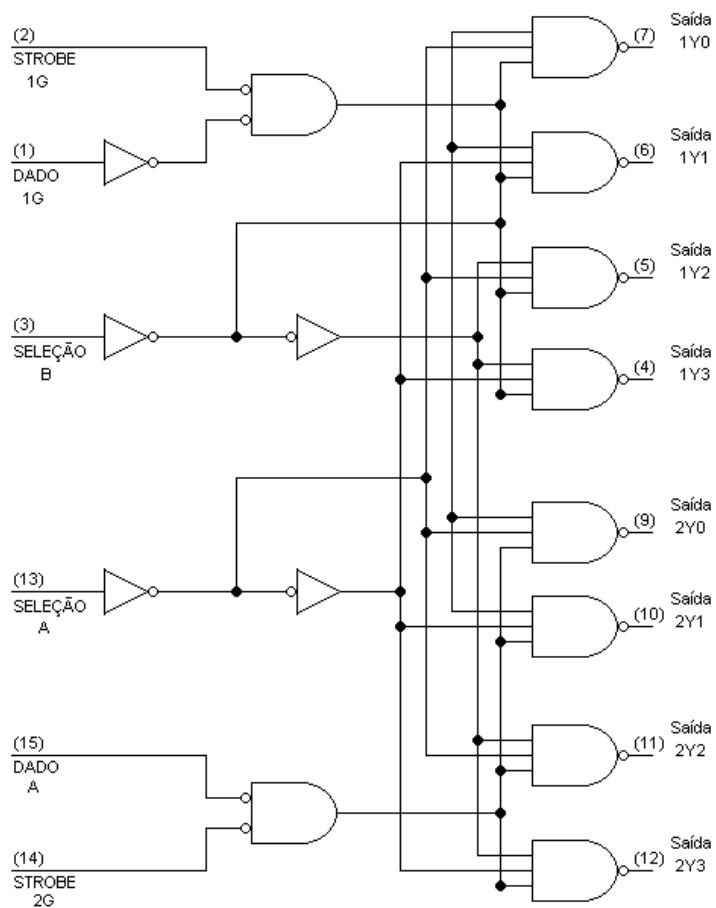
- d) Observar que com  $A = B = \emptyset$  é selecionada a saída de  $L_\emptyset$ , e que com  $A = \emptyset$  e  $B = 1$  é selecionada a saída  $L_1$ , e assim por diante.

### 8.3 DEMULTIPLEXADORES NA FORMA DE CI's

Existem vários demultiplexadores na forma de CI's, daí não haver necessidade de projetá-los. Entre estes demultiplexadores, na família TTL 74, pode-se citar os: 74154, 74155, 74156, 74159.

#### 8.3.1 DEMULTIPLEXADOR 74LS155

Este CI contém um duplo demultiplexador de 1 entrada para 4 saídas. A figura 4 mostra seu circuito em blocos e a tabela de funcionamento.



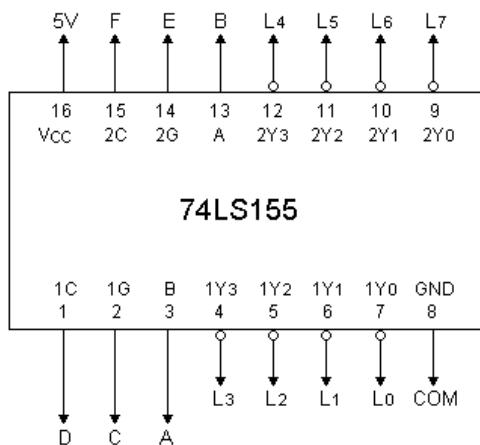
**Figura 4 - CI 74LS155 e sua tabela de funcionamento.**

O nível de saída ativa para o 74LS155 é  $\emptyset$  (baixo). Consequentemente, todas as saídas não selecionadas têm saídas 1. Se a entrada **STROBE** de qualquer seção for 1, desliga a seção, levando todas suas saídas para 1. Então a **entrada STROBE deve ser  $\emptyset$  para o demultiplexador funcionar**.

Na tabela da figura 4, pode ser visto que cada saída selecionada da seção superior do 74LS155 é o complemento dos dados na linha de entrada do CI, isto é  $1Y = 1C$ ; enquanto que, na seção inferior do 74LS155, a saída selecionada  $2Y$  é a mesma que a entrada  $2C$ , isto é  $2Y = 2C$ .

### 8.3.1.1 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74LS155
- b) Montar o circuito da figura 5.



**Figura 5 - Demultiplexador 74LS155.**

c) Completar as tabelas seguintes:

ENTRADAS				SAÍDAS			
SELEÇÃO		DADOS	STROBE	1Y3	1Y2	1Y1	1YØ
A	B	D	C	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
Ø	Ø	Ø	1				
Ø	Ø	1	1				
Ø	1	Ø	1				
Ø	1	1	1				
1	Ø	Ø	1				
1	Ø	1	1				
1	1	Ø	1				
1	1	1	1				
Ø	Ø	Ø	Ø				
Ø	Ø	1	Ø				
Ø	1	1	Ø				
1	Ø	1	Ø				
1	1	1	Ø				

- ①  
②  
③  
④  
⑤

**Tabela 3 - Seção 1 do 74155.**

#### Observações:

- ① - Todas saídas iguais a 1
- ② - Selecionada a saída 1YØ =  $\bar{D}$
- ③ - Selecionada a saída 1Y1 =  $\bar{D}$
- ④ - Selecionada a saída 1Y2 =  $\bar{D}$
- ⑤ - Selecionada a saída 1Y3 =  $\bar{D}$

ENTRADAS				SAÍDAS			
SELEÇÃO		DADOS	STROBE	2Y <sub>3</sub>	2Y <sub>2</sub>	2Y <sub>1</sub>	2Y <sub>Ø</sub>
A	B	D	C	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>
Ø	Ø	Ø	1				
Ø	Ø	1	1				
Ø	1	Ø	1				
Ø	1	1	1				
1	Ø	Ø	1				
1	Ø	1	1				
1	1	Ø	1				
1	1	1	1				
Ø	Ø	Ø	Ø				
Ø	Ø	1	Ø				
Ø	1	1	Ø				
1	Ø	1	Ø				
1	1	1	Ø				

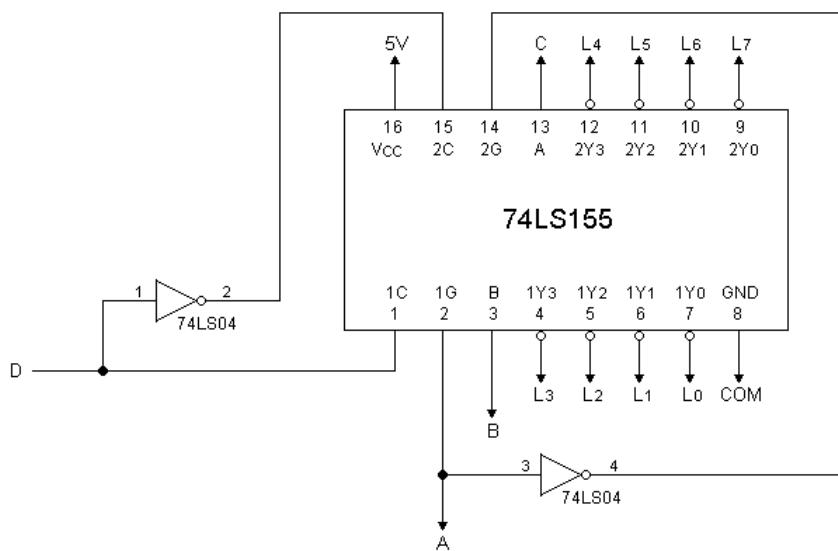
**Tabela 4 - da seção 2 do 74155.**

### Observações:

- ① - Todas saídas iguais a 1
- ② - Selecionada a saída 2Y<sub>Ø</sub> = F
- ③ - Selecionada a saída 2Y<sub>1</sub> = F
- ④ - Selecionada a saída 2Y<sub>2</sub> = F
- ⑤ - Selecionada a saída 2Y<sub>3</sub> = F

#### 8.3.1.2 EXPERIÊNCIA 3

- a) Material Utilizado: 1 x CI 74LS04 e 1 x CI 74LS155
- b) Montar o circuito do demultiplexador de 8 saídas da figura 6.



**Figura 6 - Demultiplexador de 8 saídas com 74155.**

c) Completar a tabela 5.

ENTRADAS			SAÍDAS								
Seleção			Dados	2Y <sub>3</sub>	2Y <sub>2</sub>	2Y <sub>1</sub>	2Y <sub>Ø</sub>	1Y <sub>3</sub>	1Y <sub>2</sub>	1Y <sub>1</sub>	1Y <sub>Ø</sub>
A	B	C	D	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
Ø	Ø	Ø	Ø								
Ø	Ø	Ø	1								
Ø	Ø	1	1								
Ø	1	Ø	1								
Ø	1	1	1								
1	Ø	Ø	1								
1	Ø	1	1								
1	1	Ø	1								
1	1	1	1								

**Tabela 5 - Tabela do circuito da figura 6**

### OBSERVAÇÃO:

Notar que na tabela 5 foi obtido um demultiplexador de 8 saídas.



## **CAPÍTULO 9 - COMPARADORES DIGITAIS**

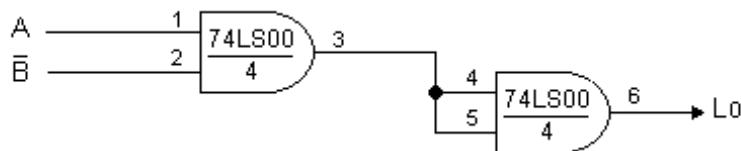
### **9.1 RESUMO**

Uma operação comum em circuitos aritméticos é a comparação de dois números, para decidir se são iguais, ou se não, ou qual o maior. Gates simples podem ser usados para efetuar esta comparação, quando se usa números de 1 único bit. No entanto, números com mais de um bit requerem circuitos mais complexos.

Os gates que executam estas comparações são chamados comparadores.

### **9.2 EXPERIÊNCIA 1**

- a)** Material Utilizado: 1 x CI 74LS00
- b)** Montar o circuito da figura 1.



**Figura 1 - Comparador A > B**

- c)** Completar a tabela 1.

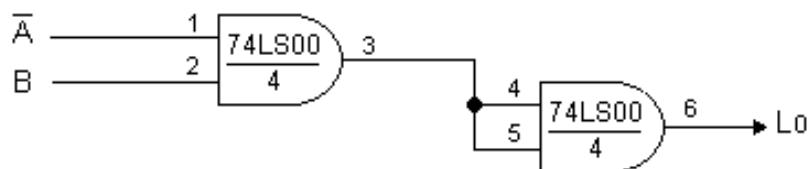
ENTRADAS		SAÍDAS
A	B	$L_0$
$\emptyset$	$\emptyset$	
$\emptyset$	1	
1	1	
1	$\emptyset$	

**Tabela 1 - Comparador A > B.**

- d)** Observar que este é um comparador do tipo A > B, isto é, a saída é 1 somente quando “A” for maior do que B.

### 9.3 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74LS00
- b) Montar o circuito da figura 2.



**Figura 2 - Comparador A < B.**

- c) Completar a tabela 2.

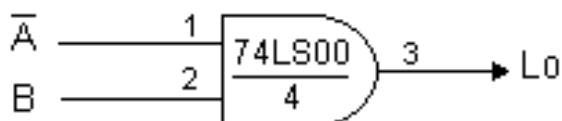
ENTRADAS		SAÍDAS
A	B	$L_0$
$\emptyset$	$\emptyset$	
$\emptyset$	1	
1	1	
1	$\emptyset$	

**Tabela 2 - Comparador A < B.**

- d) Observar que o circuito da figura 2 é um comparador do tipo A < B.

### 9.4 EXPERIÊNCIA 3

- a) Material Utilizado: 1 x CI 74LS00
- b) Montar o circuito da figura 3.



**Figura 3 - Comparador A ≥ B.**

- c) Completar a tabela 3.

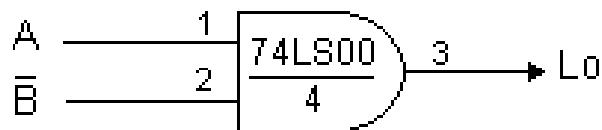
ENTRADAS		SAÍDAS
A	B	$L_\emptyset$
$\emptyset$	$\emptyset$	
$\emptyset$	1	
1	1	
1	$\emptyset$	

**Tabela 3 - Comparador  $A \geq B$ .**

- d) Observar que o circuito da figura 3 é um comparador do tipo  $A \geq B$ .

## 9.5 EXPERIÊNCIA 4

- a) Material Utilizado: 1 x CI 74LS00  
 b) Montar o circuito da figura 4.



**Figura 4 - Comparador  $A \leq B$ .**

- c) Completar a tabela 4.

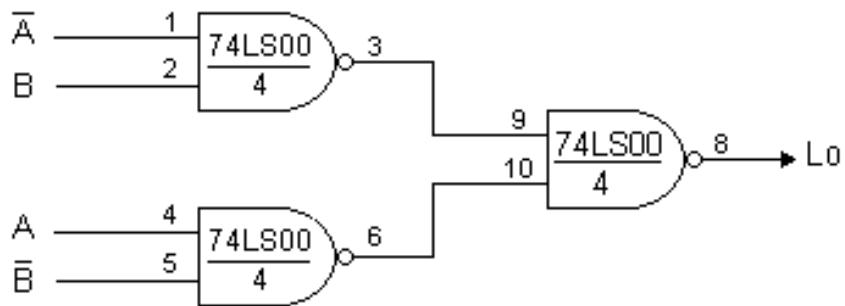
ENTRADAS		SAÍDAS
A	B	$L_\emptyset$
$\emptyset$	$\emptyset$	
$\emptyset$	1	
1	1	
1	$\emptyset$	

**Tabela 4 - Comparador  $A \leq B$ .**

- d) Observar que o circuito da 4 é um comparador do tipo  $A \leq B$ .

## 9.6 EXPERIÊNCIA 5

- a) Material Utilizado: 1 x CI 74LS00
- b) Montar o circuito da figura 5.



**Figura 5 - Comparador  $A \neq B$ .**

- c) Completar a tabela 5.

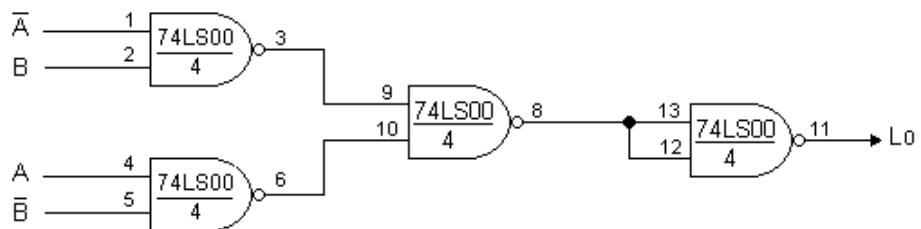
ENTRADAS		SAÍDAS
A	B	$L_\emptyset$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 5 - Comparador  $A \neq B$ .**

- d) Observar que o circuito da figura 5 é um comparador do tipo  $A \neq B$ .

## 9.7 EXPERIÊNCIA 6

- a) Material Utilizado: 1 x CI 74LS00
- b) Montar o circuito da figura 6.



**Figura 6 - Comparador  $A = B$ .**

- c) Completar a tabela 6.

ENTRADAS		SAÍDAS
A	B	$L_\emptyset$
$\emptyset$	$\emptyset$	
$\emptyset$	1	
1	1	
1	$\emptyset$	

**Tabela 6 - Comparador  $A = B$ .**

- d) Observar que o circuito da figura 6 é um comparador do tipo  $A = B$ .

## 9.8 COMPARADOR DE MAGNITUDE NA FORMA DE CI

### 9.8.1 RESUMO

Um comparador de magnitude é um circuito que aceita dois números binários como entradas e determina qual dos dois é maior. O CI 74LS85 é um comparador de magnitude de 4 bits e pode ser ligado em cascata, com outros 74LS85, a fim de efetuar a comparação de números com mais de 4 bits.

O 74LS85 tem 4 entradas A ( $A_3, A_2, A_1$  e  $A_\emptyset$ ), 4 entradas B ( $B_3, B_2, B_1$  e  $B_\emptyset$ ) e 3 entradas para cascamenteamento ( $A > B$ ,  $A < B$  e  $A = B$ ). A função deste CI é comparar dois números de 4 bits: A ( $A_3, A_2, A_1$  e  $A_\emptyset$ ) e B ( $B_3, B_2, B_1$  e  $B_\emptyset$ ). A tabela 7 mostra o funcionamento do 74LS85.

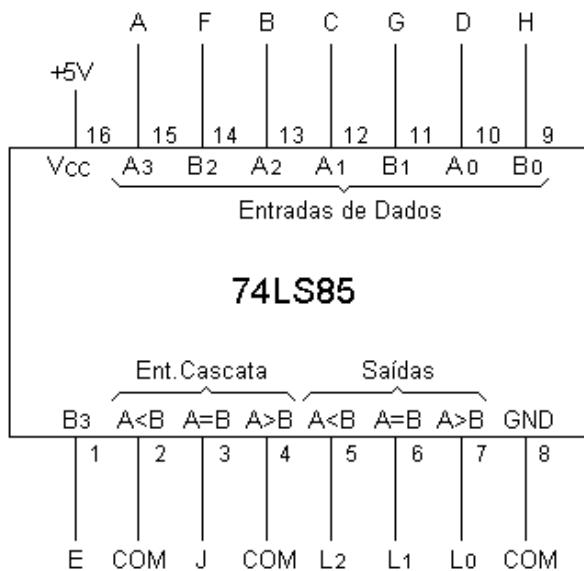
ENTRADAS DE COMPARAÇÃO				ENTRADAS DE CASCATEAMENTO			SAÍDAS		
A3, B3	A2, B2	A1, B1	AØ, BØ	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	x	x	x	x	x	x	H	L	L
A3<B3	x	x	x	x	x	x	L	H	L
A3=B3	A2>B2	x	x	x	x	x	H	L	L
A3=B3	A2<B2	x	x	x	x	x	L	H	L
A3=B3	A2=B2	A1>B1	x	x	x	x	H	L	L
A3=B3	A2=B2	A1<B1	x	x	x	x	L	H	L
A3=B3	A2=B2	A1=B1	AØ>BØ	x	x	x	H	L	L
A3=B3	A2=B2	A1=B1	AØ<BØ	x	x	x	L	H	L
A3=B3	A2=B2	A1=B1	AØ=BØ	H	L	L	H	L	L
A3=B3	A2=B2	A1=B1	AØ=BØ	L	H	L	L	H	L
A3=B3	A2=B2	A1=B1	AØ=BØ	L	L	H	L	L	H
A3=B3	A2=B2	A1=B1	AØ=BØ	x	x	H	L	L	H
A3=B3	A2=B2	A1=B1	AØ=BØ	H	H	L	L	L	L
A3=B3	A2=B2	A1=B1	AØ=BØ	L	L	L	H	H	L

**Tabela 7 - Funcionamento do 74LS85.**

A comparação começa com os bits mais significativos, A<sub>3</sub> e B<sub>3</sub>. Se A<sub>3</sub> e B<sub>3</sub> são desiguais a comparação já está determinada (isto é, se A<sub>3</sub> = 1 e B<sub>3</sub> = Ø, então A>B independente dos valores dos bits menos significativos). Se A<sub>3</sub> e B<sub>3</sub> o resultado da comparação depende de A<sub>2</sub> e B<sub>2</sub>. Se estes também são iguais, a comparação depende de A<sub>1</sub> e B<sub>1</sub> e, finalmente, de A<sub>Ø</sub> e B<sub>Ø</sub>. Se todos 4 bits de A e de B são iguais, os dois números são iguais e a saída A = B deve **Alta (1)**. Isto ocorre se as entradas de cascateamento, A>B e A<B, são **baixas (Ø)** e a entrada A = B é **Alta (1)**, como mostrado pela tabela de funcionamento.

### 9.8.2 EXPERIÊNCIA 7

- a) Material Utilizado: 1 x CI 74LS85
- b) Montar o circuito da figura 7.



**Figura 7 - CI 74LS85.**

c) Completar a tabela 8.

ENTRADAS DE COMPARAÇÃO				ENTRADAS DE CASCATEAMENTO				SAÍDAS					
A3	A2	A1	AØ	B3	B2	B1	BØ	A>B	A<B	A=B	A<B	A=B	A>B
A	B	C	D	E	F	G	H	Com.	Com.	J	L2	L1	LØ
1	1	1	Ø	Ø	Ø	Ø	Ø	Com.	Com.	1			
1	1	1	Ø	Ø	Ø	Ø	1	Com.	Com.	1			
1	1	1	Ø	Ø	Ø	1	Ø	Com.	Com.	1			
1	1	1	Ø	Ø	Ø	1	1	Com.	Com.	1			
1	1	1	Ø	Ø	1	Ø	Ø	Com.	Com.	1			
1	1	1	Ø	Ø	1	Ø	1	Com.	Com.	1			
1	1	1	Ø	Ø	1	1	Ø	Com.	Com.	1			
1	1	1	Ø	1	Ø	Ø	Ø	Com.	Com.	1			
1	1	1	Ø	1	Ø	Ø	1	Com.	Com.	1			
1	1	1	Ø	1	1	1	Ø	Com.	Com.	1			
1	1	1	Ø	1	Ø	Ø	Ø	Com.	Com.	1			
1	1	1	Ø	1	Ø	Ø	1	Com.	Com.	1			
1	1	1	Ø	1	Ø	1	1	Com.	Com.	1			
1	1	1	Ø	1	1	Ø	Ø	Com.	Com.	1			
1	1	1	Ø	1	1	Ø	1	Com.	Com.	1			
1	1	1	Ø	1	1	1	Ø	Com.	Com.	1			
1	1	1	Ø	1	1	1	Ø	Com.	Com.	1			
1	1	1	Ø	1	1	1	Ø	Com.	Com.	Ø			

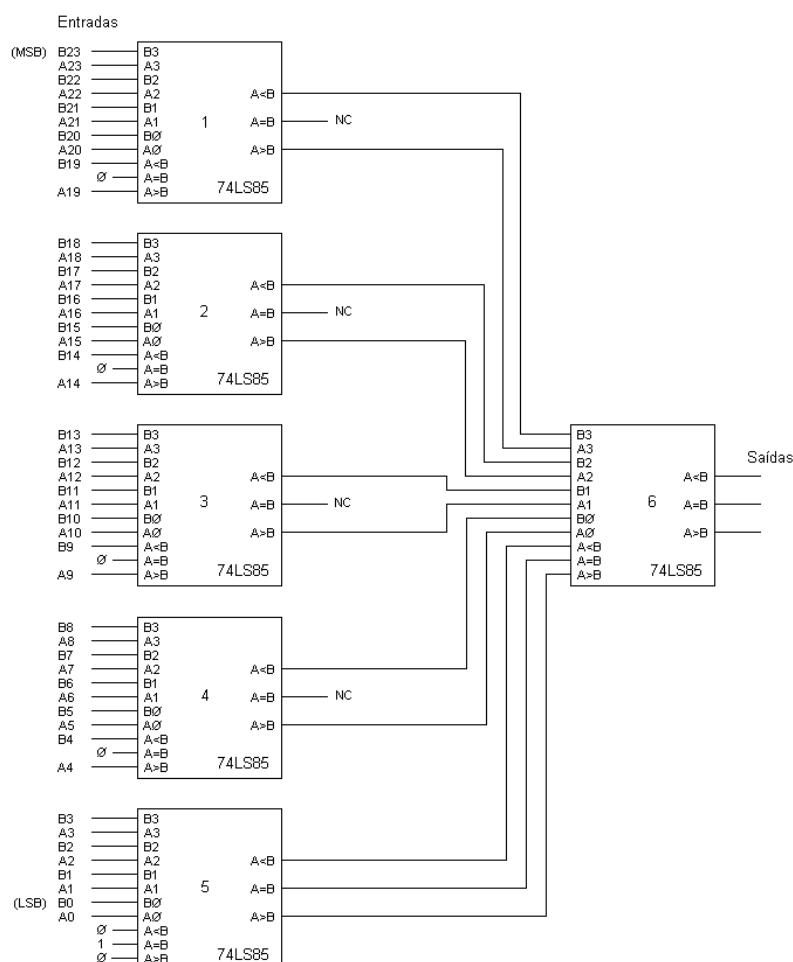
**Tabela 8 - 74LS85.**

d) Observar que em:

- ① tem-se  $A_3, A_2, A_1 \text{ e } A_\emptyset = 111\emptyset$  e  $B_3, B_2, B_1 \text{ e } B_\emptyset$  está variando desde  $\emptyset\emptyset\emptyset\emptyset$  até  $11\emptyset1$ , portanto  $A_3, A_2, A_1 \text{ e } A_\emptyset > B_3, B_2, B_1 \text{ e } B_\emptyset$ .
- ② tem-se  $A_3, A_2, A_1 \text{ e } A_\emptyset = B_3, B_2, B_1 \text{ e } B_\emptyset$ , com as entradas de cascamenteamento ( $A>B = \emptyset$ ,  $(A<B) = \emptyset$ ,  $(A=B) = 1$ ). Então da linha 11 da tabela da figura 13 tem-se as saídas  $(A>B) = \emptyset$ ,  $(A<B) = \emptyset$  e  $(A=B) = 1$ .
- ③ tem-se  $A_3, A_2, A_1 \text{ e } A_\emptyset = B_3, B_2, B_1 \text{ e } B_\emptyset$ , com as entradas de casca-teamento ( $A>B = \emptyset$ ,  $(A<B) = \emptyset$ ,  $(A=B) = \emptyset$ ). Então, de acordo com a linha 14 da tabela da figura 13 tem-se que as saídas  $(A>B) = 1$ ,  $(A<B) = 1$  e  $(A=B) = \emptyset$ .

## 9.9 COMPARADORES MAIORES COM O 74LS85

Os comparadores 74LS85 podem ser ligados em cascata, para efetuarem a comparação de números maiores do que 4 bits. A figura 8 mostra um circuito composto de seis CI's 74LS85, destinado a comparar dois números de 24 bits.



**Figura 8 - Comparador de 24 bits com 74LS85.**

Para compreender o funcionamento do circuito da figura 8, analisemos o caso onde:

$A_{23} A_{22} A_{21} A_{20} A_{19} A_{18} A_{17} A_{16} \dots = 1\emptyset\emptyset1\emptyset\emptyset\emptyset1 \dots$

e

$B_{23} B_{22} B_{21} B_{20} B_{19} B_{18} B_{17} B_{16} \dots = 1\emptyset\emptyset1\emptyset\emptyset11 \dots$

A diferença mais significativa ocorre no bit 17, e B é maior do que A. Para o Cl<sub>1</sub>, A<sub>3</sub> = B<sub>3</sub>, A<sub>2</sub> = B<sub>2</sub>, A<sub>1</sub> = B<sub>1</sub>, A<sub>Ø</sub> = B<sub>Ø</sub> e as entradas de cascamenteamento são todas Ø, pois A<sub>19</sub> = B<sub>19</sub> e a entrada A = B está ligada a Ø. Sob estas condições, as saídas A > B e A < B são 1, e como estas saídas estão ligadas as entradas A<sub>3</sub> e B<sub>3</sub> do Cl<sub>6</sub>, tem-se que para o Cl<sub>6</sub> A<sub>3</sub> = B<sub>3</sub>.

Para o Cl<sub>2</sub>, B<sub>3</sub> = A<sub>3</sub>, mas B<sub>2</sub> > A<sub>2</sub> (B<sub>17</sub> = 1, A<sub>17</sub> = Ø), portanto a saída A < B é 1. Esta saída é ligada a entrada B<sub>2</sub> do Cl<sub>6</sub>, enquanto a entrada A > B, que é Ø, é conectada a entrada A<sub>2</sub>. Consequentemente, a saída A < B do Cl<sub>6</sub> é 1, que é a comparação correta.



## CAPÍTULO 10 - GERAÇÃO E CHECK DE PARIDADE

### 10.1 RESUMO

Pode-se definir paridade como a adição de um bit a uma palavra binária, para se assegurar a integridade dos dados. Há dois tipos de paridade: **par** e **ímpar**. Para se gerar uma paridade **ímpar**, **um bit é adicionado a cada palavra de modo que o número de 1's na palavra seja ímpar**. Por exemplo, se uma palavra representando um caracter, quando fornecida por um computador ou outro dispositivo, consiste de 7 bits, um 8º bit é acrescentado a palavra para tornar o número de 1's no caracter de 8 bits ímpar.

A paridade par é o complemento da paridade ímpar. Se paridade par for usada, o número de 1's em cada caracter tem que se par. Este processo de somar um bit para fazer a paridade par ou ímpar é chamado de **geração de paridade**.

#### Exemplo:

Os caracteres X, Y e Z são recebidos de um computador. Gerar um oitavo bit para manter:

- a)** Paridade par      **b)** Paridade ímpar.

onde:

$$X = \emptyset 1 \emptyset 1 \emptyset 1 1$$

$$Y = \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset$$

$$Z = 1 1 1 1 1 1 1$$

#### Solução:

- a)** Paridade par

$$X = \emptyset 1 \emptyset 1 \emptyset 1 1 \quad \text{o bit de paridade é } \emptyset$$

$$Y = \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset \quad \text{o bit de paridade é } \emptyset$$

$$Z = 1 1 1 1 1 1 1 \quad \text{o bit de paridade é } 1$$

- b)** Paridade ímpar

$$X = \emptyset 1 \emptyset 1 \emptyset 1 1 \quad \text{o bit de paridade é } 1$$

$$Y = \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset \emptyset \quad \text{o bit de paridade é } 1$$

$$Z = 1 1 1 1 1 1 1 \quad \text{o bit de paridade é } \emptyset$$

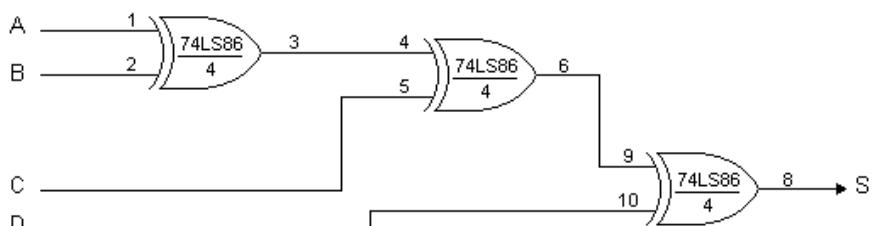
## 10.2 CHEQUE DE PARIDADE

Cheque de paridade é o processo de se examinar todos n bits de uma palavra, para determinar se o número de 1's nos n bits é ímpar ou par, e indicar um erro se a paridade estiver errada. A tabela 1 mostra o cheque de paridade ímpar para 4 entradas.

ENTRADAS				CHEQUE DE PARIDADE ÍMPAR:
A	B	C	D	S
Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	1	1
Ø	Ø	1	1	Ø
Ø	Ø	1	Ø	1
Ø	1	1	Ø	Ø
Ø	1	1	1	1
Ø	1	Ø	1	Ø
Ø	1	Ø	Ø	1
1	1	Ø	Ø	Ø
1	1	Ø	1	1
1	1	1	1	Ø
1	1	1	Ø	1
1	Ø	1	Ø	Ø
1	Ø	1	1	1
1	Ø	Ø	1	Ø
1	Ø	Ø	Ø	1

**Tabela 1 - Cheque de paridade ímpar para 4 entradas.**

Da tabela 1, observa-se que o cheque de paridade é “1” se houver um número ímpar de entradas “1”. Um circuito que satisfaça a tabela 1 pode ser implementado por OU exclusivos, conforme mostra a figura 1.

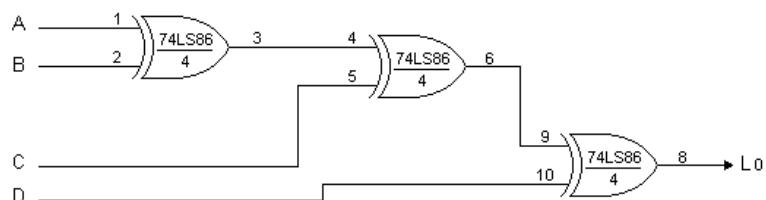


**Figura 1 - Circuito de cheque de paridade ímpar**

Existe uma relação geral entre circuitos de paridade e gates OU Exclusivos, que é a seguinte: **Se todas saídas de um registrador são ligadas a OU exclusivos, a saída do circuito OU exclusivo é 1 somente se o número de 1's da palavra de entrada é ímpar.**

### 10.2.1 EXPERIÊNCIA 1

- a)** Material Utilizado: 1 x CI 74LS86  
**b)** Montar o circuito da figura 2.



**Figura 2 - Circuito para cheque de paridade ímpar**

- c)** Completar a tabela 2.

ENTRADAS				SAÍDA CHEQUE DE PARIDADE
A	B	C	D	LØ
Ø	Ø	Ø	1	
Ø	Ø	1	Ø	
Ø	1	Ø	Ø	
Ø	1	1	1	
1	Ø	Ø	Ø	
1	Ø	1	1	
1	1	Ø	1	
1	1	1	Ø	

**Tabela 2 – Tabela para o circuito da figura 2.**

- d)** Na tabela 2, observar que qualquer saída Ø indica um erro, pois está se checando paridade ímpar.

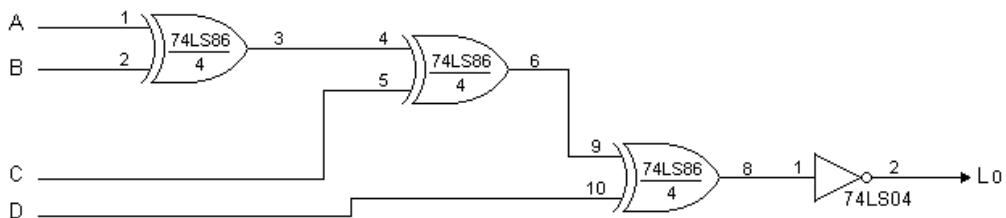
### 10.3 GERAÇÃO DE PARIDADE

A geração de paridade envolve o acréscimo um bit extra a uma palavra de n bits, para se produzir a paridade apropriada na palavra de n + 1 bits. Um circuito para gerar a paridade apropriada pode ser construído usando-se “OU Exclusivos”. Se, por exemplo, é requerido **paridade ímpar**, a palavra de n bits é **checada para paridade ímpar**. Se o número de 1's na palavra de n bits examinada é ímpar, um Ø deve ser escrito no bit de paridade, e o número ímpar de 1's na palavra é assim preservado. Se, o número de 1's for par, um 1 deve ser escrito na palavra de n bits examinada como bit de paridade, de modo que a palavra de n + 1 bits tenha paridade ímpar.

**No circuito de cheque de paridade da figura 2 a geração do bit de paridade é obtida invertendo-se o bit de cheque.**

### 10.3.1 EXPERIÊNCIA 2

- a)** Material Utilizado: 1 x CI 74LS86 e 1 x CI 74LS04  
**b)** Montar o circuito da figura 3.



**Figura 3 - Gerador de paridade ímpar.**

- c)** Completar a tabela 3.

ENTRADAS				SAÍDA Bit de Paridade Ímpar
A	B	C	D	L $\emptyset$
Ø	Ø	Ø	Ø	
Ø	Ø	Ø	1	
Ø	Ø	1	1	
Ø	Ø	1	Ø	
Ø	1	1	Ø	
Ø	1	1	Ø	
Ø	1	1	1	
Ø	1	Ø	1	
Ø	1	Ø	Ø	
1	1	Ø	Ø	
1	1	Ø	1	
1	1	1	1	
1	1	1	Ø	
1	Ø	1	Ø	
1	Ø	1	1	
1	Ø	Ø	1	
1	Ø	Ø	Ø	

**Tabela 3 - Tabela para o circuito da figura 5.**

- d)** Na tabela 3, observar que quando se considera a saída L $\emptyset$  e as entradas A B C D, o número de bits “1” é ímpar.

## 10.4 CIRCUITOS INTEGRADOS PARA CHEQUE E GERAÇÃO DE PARIDADE

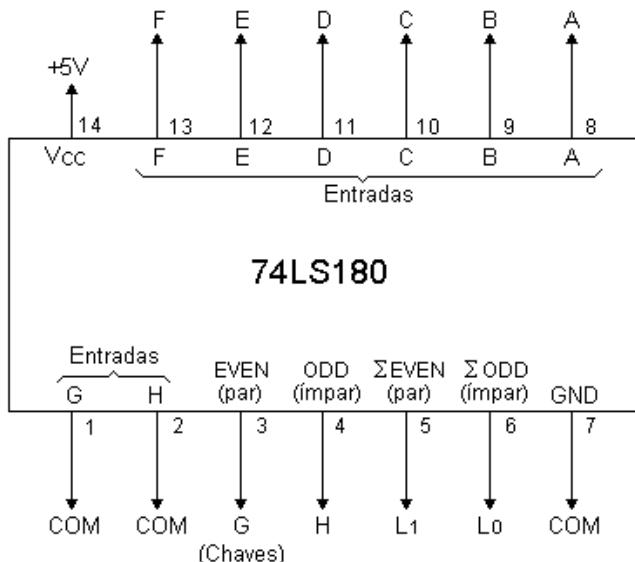
Vamos neste texto abordar o CI TTL 74LS180, que é um **gerador e cheque** de paridade e que pode checar uma entrada de 9 bits, gerando um 10º bit de paridade.

O 74LS180 consta de:

- 8 entradas para os bits cuja paridade deve ser checada (**Entradas A até H**).
- 2 entradas **ímpar (ODD)** e **par (EVEN)**. Tais entradas podem ser usadas para acomodar o 9º bit, ou para permitir a ligação em cascata de 74LS180's para palavras maiores que 9 bits.
- 2 saídas, uma **ímpar** e uma **par**, que indicam a paridade das entradas. Estas saídas são sempre complementares.

### 10.4.1 EXPERIÊNCIA 3

- Material Utilizado: 1 x CI 74LS180
- Montar o circuito da figura 4.



**Figura 4 - O CI 74LS180.**

- c) Observar a tabela 4, obtida do manual do fabricante.

ENTRADAS			SAÍDAS	
$\Sigma$ de 1's nas entradas A até H	PAR	ÍMPAR	$\Sigma$ PAR	$\Sigma$ ÍMPAR
Par	1	$\emptyset$	1	$\emptyset$
Ímpar	1	$\emptyset$	$\emptyset$	1
Par	$\emptyset$	1	$\emptyset$	1
Ímpar	$\emptyset$	1	1	$\emptyset$
x (irrelevante)	1	1	$\emptyset$	$\emptyset$
x (irrelevante)	$\emptyset$	$\emptyset$	1	1

**Tabela 4 - 74LS180.**

- d) Completar a tabela 5.

Nesta experiência usaremos apenas 6 entradas, conectando os pinos 1 e 2 (entradas G e H) do 74LS180 ao COMUM.

ENTRADAS								SAÍDAS	
A	B	C	D	E	F	PAR	ÍMPAR	PAR	ÍMPAR
A	B	C	D	E	F	G	H	L1	L $\emptyset$
$\emptyset$	1	1	$\emptyset$	1	1	1	$\emptyset$		
$\emptyset$	$\emptyset$	1	$\emptyset$	1	1	1	$\emptyset$		
$\emptyset$	1	1	$\emptyset$	1	1	$\emptyset$	1		
$\emptyset$	$\emptyset$	1	$\emptyset$	1	1	$\emptyset$	1		
x	x	x	x	x	x	1	1		
x	x	x	x	x	x	$\emptyset$	$\emptyset$		

①

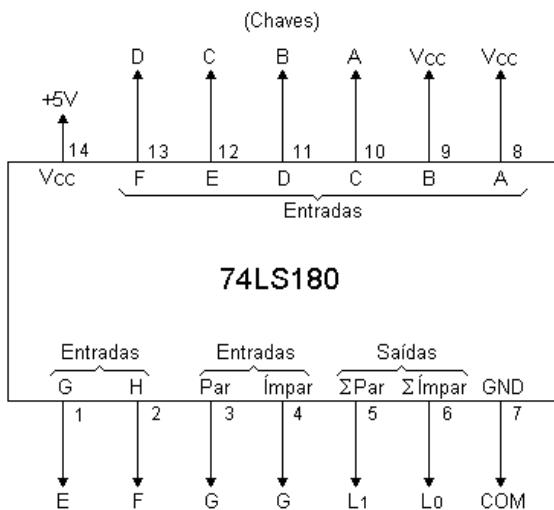
**Tabela 5 - Tabela para o circuito da figura 4.**

- e) Na tabela 5, o item ①, observar que com  $G = 1$ ,  $H = \emptyset$  e de 1's nas entradas A até F = par, as saídas são:  
 $L1$  (par) = 1 e  $L\emptyset$  (ímpar) =  $\emptyset$ , o que confere com a 1<sup>a</sup> fila da tabela 4. As demais filas da tabela 5 conferem com a tabela do fabricante (tabela 4).
- f) Se fosse desejado usar o CI 74LS180 como um gerador de paridade para um número menor de entradas, por exemplo, 5 bastariam ligar a 6<sup>a</sup>, 7<sup>a</sup> e 8<sup>a</sup> entradas (no caso F, G e H) ao nível lógico  $\emptyset$ . Deste modo, a entrada não usada não afeta a paridade. De modo análogo, pode-se obter geradores de paridade para números de entradas menores. Os níveis das saídas pares e ímpares devem ser selecionados pelas entradas pares e ímpares.

## 10.5 USO DO 74LS180 PARA CHEQUE DE PARIDADE DE 9 BITS

### 10.5.1 EXPERIÊNCIA 4

- a) Material Utilizado: 1 x CI 74LS180
- b) Montar o circuito da figura 5.



**Figura 5 - 74LS180 para cheque de paridade de palavra de 9 bits.**

- c) Completar a tabela 6.

Chaves	ENTRADAS									SAÍDAS	
	A	B	C	D	E	F	G	H	PAR	$\Sigma$ PAR	$\Sigma$ ÍMPAR
VCC	VCC	A	B	C	D	E	F	F	G	L1	LØ
1	1	1	1	1	1	Ø	Ø	Ø	Ø		
1	1	1	1	1	1	Ø	Ø	1	Ø		
1	1	1	1	1	1	Ø	Ø	Ø	1		
1	1	1	1	1	1	Ø	Ø	1	1		

Led's  
 ①  
 ②

**Tabela 6 - Tabela para circuito da figura 5.**

- d) Em ①, observar que os primeiros 8 bits são aplicados de A até H, e que o 9º bit é aplicado a entrada par, sendo seu complemento aplicado a entrada ímpar (I). O 9º bit, que é o Ø, faz com que a entrada par seja Ø e a entrada ímpar seja 1. Desde que as entradas A até H contém um número ímpar de 1's (cinco), a linha 4 da tabela de funcionamento (tabela 4) se aplica e  $\Sigma$  par é 1. Portanto, o circuito tem sua saída  $\Sigma$  par = 1 quando a paridade da palavra de entrada for ímpar, e tem  $\Sigma$  par = Ø quando a paridade da palavra de entrada for par (linha 3 da tabela 4).

- e) Em ②, observar que  $G = 1$  (entrada par = 1) e  $\bar{G} = \emptyset$  (entrada ímpar =  $\emptyset$ ). Neste caso, as saídas obedecem as 2 primeiras filas das tabela 4.

## 10.6 USO DO 74LS180 EM CASCATA

Para se checar a paridade de palavras maiores do que 9 bits, pode-se ligar CI's 74LS180 em cascata, ligando-se a saída  $\Sigma$  **EVEN (par)** do primeiro 74LS180 a entrada **EVEN (par)** do segundo 74LS180, e a saída  $\Sigma$  **ODD (ímpar)** do primeiro 74LS180 à entrada **ODD (ímpar)** do segundo 74LS180. O primeiro 74LS180 pode aceitar 9 entradas e o segundo mais 8 entradas. Cada 74LS180 adicionado a cascata aumenta sua capacidade de entradas de 8.

## **CAPÍTULO 11 - CIRCUITOS ARITMÉTICOS**

### **11.1 RESUMO**

A vantagem de sistemas digitais é que todas as quantidades podem ser manipuladas rapidamente usando operações numéricas. As operações de adição, subtração, multiplicação e divisão de números binários são executadas pelos circuitos aritméticos na maioria dos equipamentos digitais. Existem dispositivos mais flexíveis tais como as unidades aritméticas e lógicas "ALU" que são fornecidas na forma de CI's, e que podem executar diversos tipos de operação, as quais são selecionadas por entradas programáveis.

### **11.2 MEIO SOMADOR**

O meio somador é um circuito básico destinado a somar dois bits.

A soma de dois bits A e B obedecem às regras da tabela 1, onde S é a soma e C é o Carry ou "Vai 1".

A	B	S	C
Ø	Ø	Ø	Ø
Ø	1	1	Ø
1	1	Ø	1
1	Ø	1	Ø

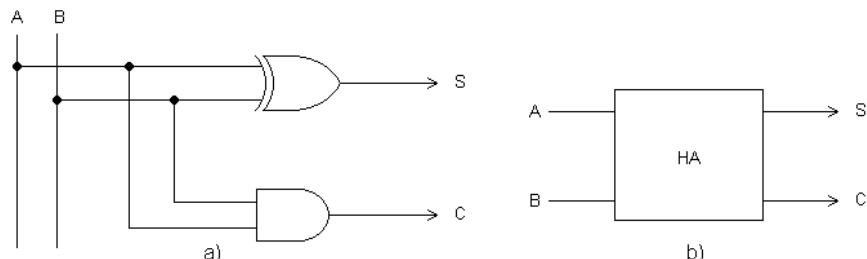
**Tabela 1 - Meio-Somador.**

Da tabela 1, pode-se obter as expressões lógicas de S e C, que são:

$$S = \bar{A}\bar{B} + \bar{A}B + A\bar{B} = A \oplus B \quad \text{e}$$

$$C = AB$$

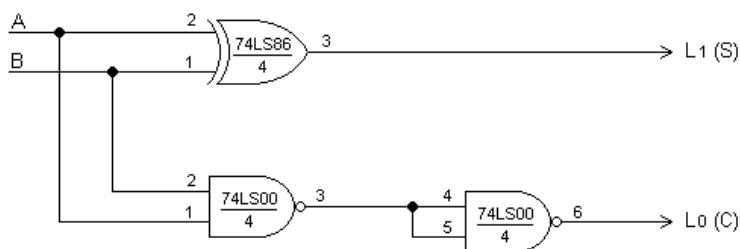
A partir destas expressões obtém-se o circuito de um meio somador como na figura 1a. e sua representação esquemática é mostrada na figura 1b.



**Figura 1 - Circuito e representação esquemática de um Meio-Somador.**

### 11.2.1 EXPERIÊNCIA 1

- a)** Material Utilizado: 1 x CI 74LS00 e 1 x CI 74LS86  
**b)** Montar o circuito da figura 2.



**Figura 2 - Meio Somador com Gates.**

- c)** Completar a tabela 2.

A	B	S      C	
		L <sub>1</sub>	L <sub>0</sub>
Ø	Ø		
Ø	1		
1	1		
1	Ø		

**Tabela 2 – Tabela para o circuito da figura 2.**

### 11.3 MEIO SUBTRATOR

O meio subtrator é um circuito básico destinado a subtrair dois bits.

A subtração de dois bits A e B obedecem às regras da tabela 3, onde “S” é o resultado da subtração e “C” é o carry ou "pede emprestado."

A	B	S	C
Ø	Ø	Ø	Ø
Ø	1	1	1
1	1	Ø	Ø
1	Ø	1	Ø

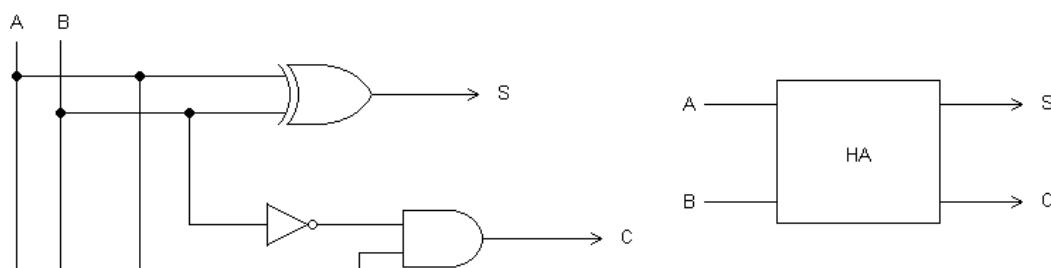
**Tabela 3 - Tabela para Meio-Subtrator.**

Da tabela 3, pode-se obter as expressões lógicas de S e C, que são:

$$S = A\bar{B} + \bar{A}\bar{B} = A \oplus B \quad \text{e}$$

$$C = \bar{A}B$$

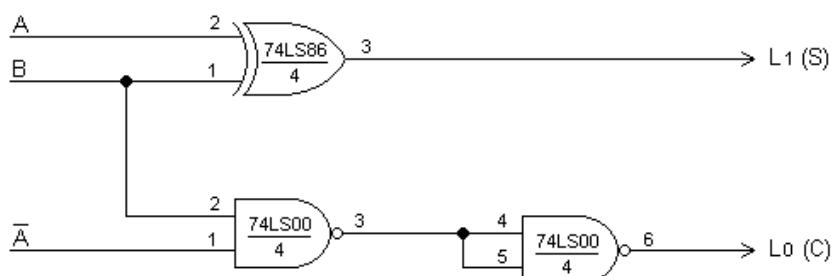
A partir destas expressões obtém-se o circuito de um meio subtrator como na figura 3a. e sua representação esquemática é mostrada na figura 3b.



**Figura 3 - Circuito e representação esquemática de um Meio Subtrator.**

### 11.3.1 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74LS00 e 1 x CI 74LS86
- b) Montar o circuito da figura 4.



**Figura 4 - Meio Subtrator com Gates.**

- c) Completar a tabela 4.

A	B	S      C	
		L <sub>1</sub>	L <sub>0</sub>
Ø	Ø		
Ø	1		
1	1		
1	Ø		

**Tabela 4 - Tabela para o circuito da figura 4.**

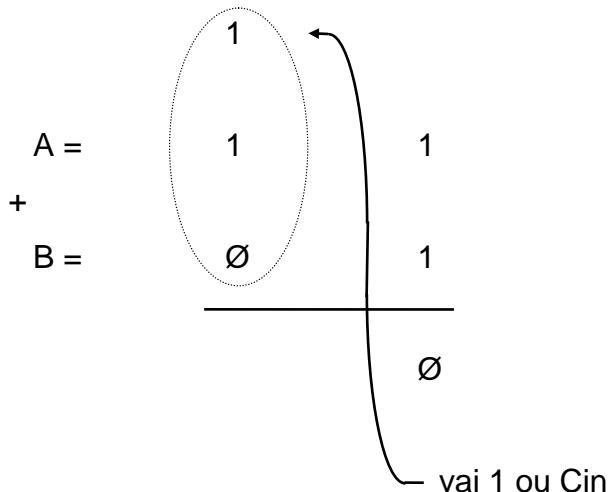
## 11.4 SOMADOR

Quando se deseja somar dois números binários A e B, cada qual contendo mais de um bit, usa-se o somador. Quase sempre há necessidade de se somar 3 bits conforme mostrado no exemplo seguinte.

Exemplo:

Efetuar a soma dos números binários A e B, sendo  $A = 11$  e  $B = 01$ .

Solução:



As regras de soma de 3 bits estão mostradas na tabela 5, onde "S" é o resultado da soma, Cin é o Carry interno ou "vai 1" interno e Cout é o Carry externo ou "vai 1" externo.

A	B	Cin	S	Cout
Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	1	Ø
Ø	1	1	Ø	1
Ø	1	Ø	1	Ø
1	1	Ø	Ø	1
1	1	1	1	1
1	Ø	1	Ø	1
1	Ø	Ø	1	Ø

**Tabela 5 - Soma de três bits.**

Da tabela 5 podem-se obter as expressões lógicas de S e C (Cout), que são:

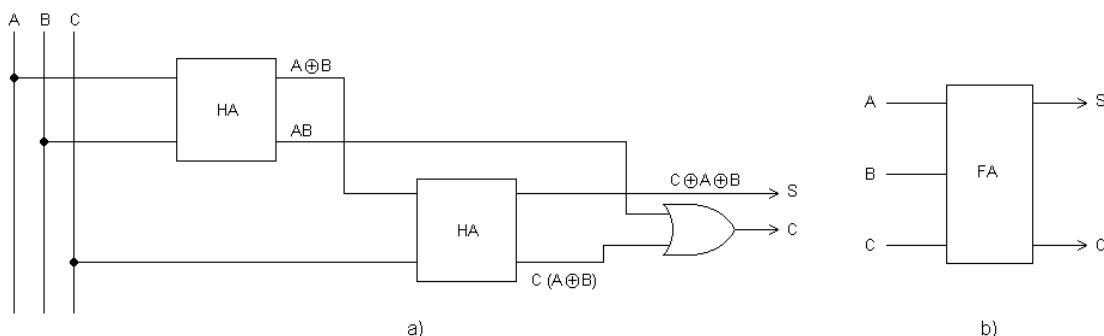
$$S = \overline{A}\overline{B}C + \overline{A}B\overline{C} + ABC + A\overline{B}\overline{C} = C(\overline{A}\overline{B} + AB) + \overline{C}(\overline{A}B + A\overline{B})$$

$$S = C(\overline{A} \oplus B) + \overline{C}(A \oplus B) = C \oplus A \oplus B$$

$$C = \overline{A}BC + A\overline{B}\overline{C} + ABC + A\overline{B}C = C(\overline{A}B + A\overline{B}) + AB$$

$$C = C(A \oplus B) + AB$$

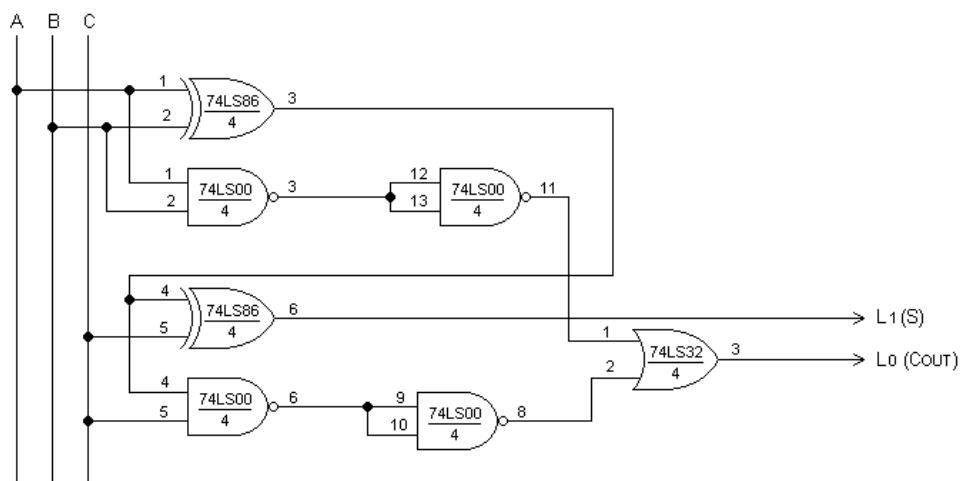
A partir das expressões de “S” e “C”, obtém-se o circuito de um somador utilizando-se meio-somadores como na figura 5a. e sua representação esquemática é mostrado na figura 5b.



**Figura 5 - Circuito e representação esquemática de um Somador.**

#### 11.4.1 EXPERIÊNCIA 3

- a) Material Utilizado: 1 x CI 74LS00, 1 x CI 74LS32 e 1 x CI 74LS86
- b) Montar o circuito da figura 6



**Figura 6 - Somador com Gates.**

c) Completar a tabela 6.

A	B	C	S      C	
			L <sub>1</sub>	L <sub>0</sub>
Ø	Ø	Ø		
Ø	Ø	1		
Ø	1	1		
Ø	1	Ø		
1	1	Ø		
1	1	1		
1	Ø	1		
1	Ø	Ø		

**Tabela 6 - Tabela para circuito da figura 6.**

### 11.5 SUBTRATOR

Tal como já abordado para o somador, quando se deseja subtrair dois números binários A e B, cada qual contendo mais de um bit, quase sempre há necessidade de se efetuar uma subtração envolvendo 3 bits, conforme mostrado no exemplo seguinte. Nestes casos usa-se um subtrator.

Exemplo:

Efetuar a subtração entre os números binários A e B, sendo A = 10 e B = 01.

Solução:

$$\begin{array}{r}
 A = \begin{array}{c} 1 \\ \text{---} \\ \emptyset \end{array} \\
 - \quad \quad \quad \\
 B = \begin{array}{c} \emptyset \\ \text{---} \\ 1 \end{array} \\
 \hline
 \begin{array}{c} 1 \\ \text{---} \\ 1 \end{array} \quad \text{Pede emprestado ou Carry 1} \\
 \hline
 \emptyset
 \end{array}$$

As regras de subtração de 3 bits estão mostradas na tabela 7, onde “S” é o resultado da subtração, Cin é o Carry interno ou "pede emprestado interno", Cout é o Carry externo ou "pede emprestado externo".

A	B	Cin	S	Cout
Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	1	1
Ø	1	1	Ø	1
Ø	1	Ø	1	1
1	1	Ø	Ø	Ø
1	1	1	1	1
1	Ø	1	Ø	Ø
1	Ø	Ø	1	Ø

**Tabela 7 - Subtração de três bits.**

Da tabela 7, pode-se obter as expressões lógicas de S e C (Cout), que são:

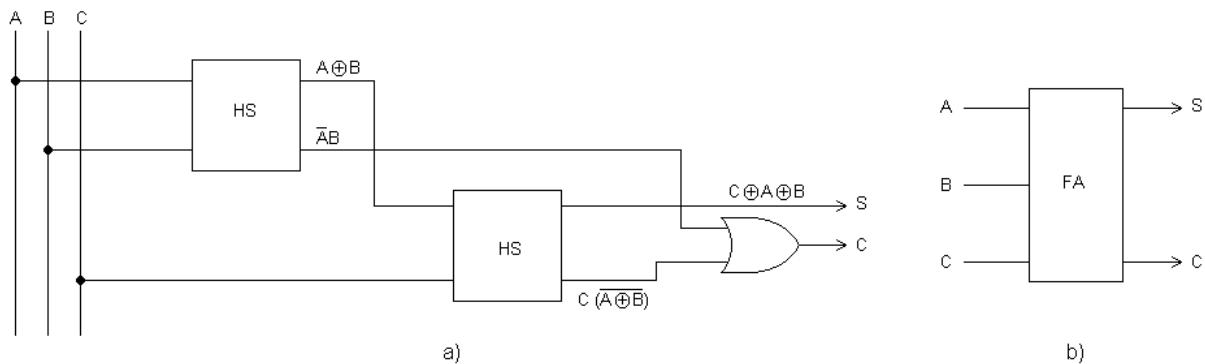
$$S = \overline{A}\overline{B}C + \overline{A}B\overline{C} + ABC + A\overline{B}\overline{C} = C(\overline{A}\overline{B} + AB) + \overline{C}(\overline{A}B + A\overline{B})$$

$$S = C(\overline{A} \oplus B) + \overline{C}(A \oplus B) = C \oplus A \oplus B$$

$$C = \overline{A}\overline{B}C + \overline{A}BC + \overline{ABC} + ABC = C(\overline{A}\overline{B} + AB) + \overline{AB}$$

$$C = C(\overline{A} \oplus B) + \overline{AB}$$

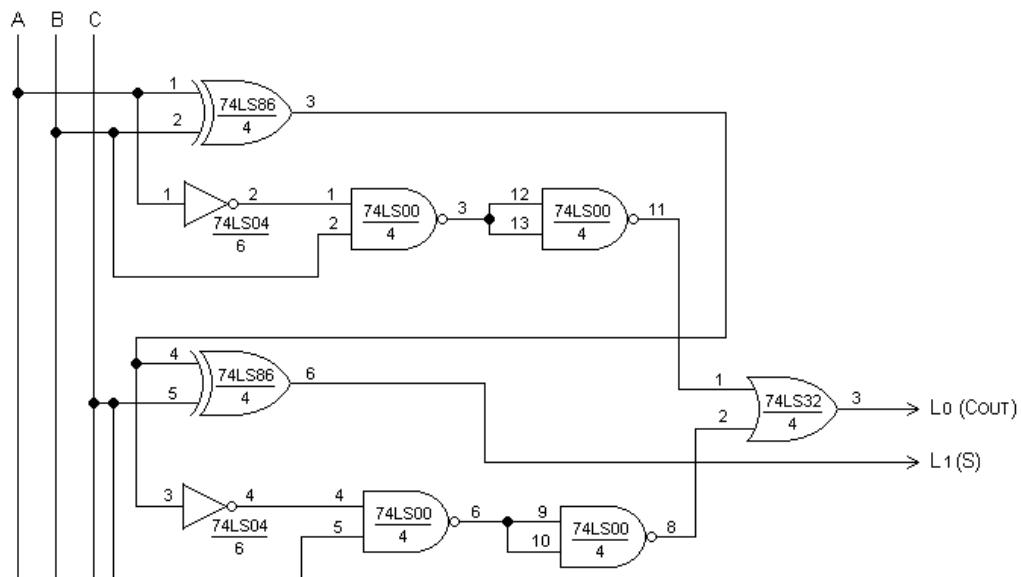
A partir das expressões de S e C, obtém-se o circuito de um subtrator utilizando-se meio-subtratores como na figura 7a. e sua representação esquemática é mostrado na figura 7b.



**Figura 7 - Circuito e representação esquemática de um Subtrator.**

### 11.5.1 EXPERIÊNCIA 4

- a)** Material Utilizado: 1 x CI 74LS00, 1 x CI 74LS04, 1 x CI 74LS32 e 1 x CI 74LS86
- b)** Montar o circuito da figura 8.



**Figura 8 - Subtrator com Gates.**

- c)** Completar a tabela 8.

A	B	C	S      C	
			L <sub>1</sub>	L <sub>0</sub>
Ø	Ø	Ø		
Ø	Ø	1		
Ø	1	1		
Ø	1	Ø		
1	1	Ø		
1	1	1		
1	Ø	1		
1	Ø	Ø		

**Tabela 8 - Tabela para o circuito da figura 8.**

## 11.6 SUBTRAÇÃO ATRAVÉS DE SOMA

Grande parte dos sistemas digitais usa um único circuito para efetuar subtração e soma. Neste circuito a subtração é efetuada por soma, num processo conhecido como complemento de dois.

### 11.6.1 COMPLEMENTO DE UM NÚMERO BINÁRIO

O complemento de um número binário é obtido trocando-se todos "0s" por "1s" e todos "1s" por "0s".

Exemplo:

Obter o complemento do binário 101011

Solução:

O complemento é 010100

### 11.6.2 COMPLEMENTO DE DOIS DE UM NÚMERO BINÁRIO

O complemento de dois de um número binário é obtido tomando-se o complemento do número binário e somando-se 1 a este complemento.

Exemplo 1:

Obter o complemento de dois do binário correspondente ao decimal 3.

Solução:

$3_{10} = 11_2$

O complemento de 11 é 00.

O complemento de dois de 11 é 00

$$\begin{array}{r} + 1 \\ \hline 01 \end{array}$$

Então, o complemento de 2 de 11 é 01.

Exemplo 2:

Obter o complemento de dois do binário 01.

Solução:

O complemento de  $\emptyset 1$  é  $1\emptyset$ .

O complemento de 2 de  $\emptyset 1$  é  $1\emptyset + 1 = 11$

Então, o complemento de 2 do complemento de 2 é o número original.

Exemplo 3:

Efetuar a subtração através da soma

$1\emptyset\emptyset$  (Minuendo)

-  $\emptyset 11$  (Subtraendo)

Solução:

Toma-se o complemento de dois do subtraendo e soma-se este complemento ao minuendo; tem-se então:

$$\begin{array}{r}
 1\emptyset\emptyset \\
 - \underline{\emptyset 11} \\
 \hline
 \end{array}
 \rightarrow
 \begin{array}{r}
 1\emptyset\emptyset \\
 + \quad 1 \\
 \hline
 1\emptyset 1
 \end{array}$$

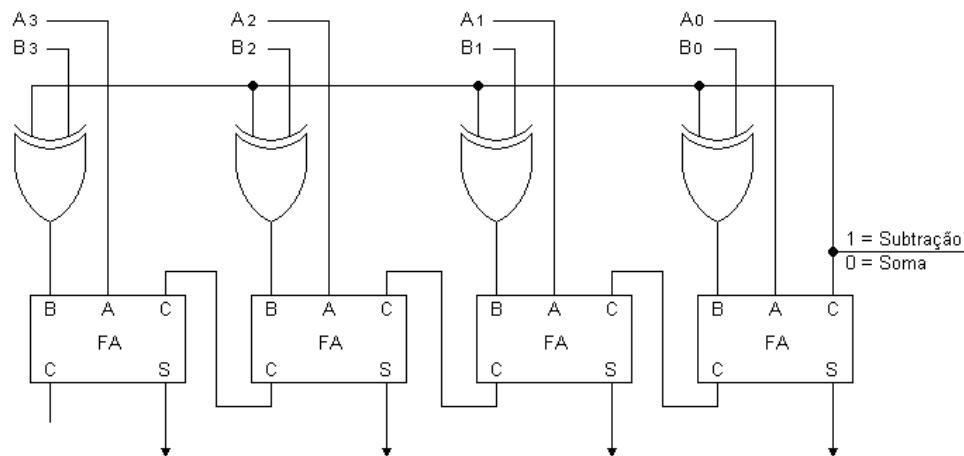
$$\begin{array}{r}
 1\emptyset\emptyset \\
 + \quad 1\emptyset 1 \\
 \hline
 \boxed{\emptyset\emptyset 1}
 \end{array}$$

Abandona-se o overflow

Assim o resultado é  $\emptyset\emptyset 1$ .

### 11.6.3 CIRCUITO SUBTRATOR SOMADOR

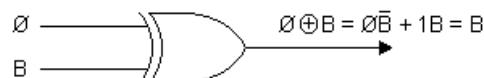
A figura 9 mostra um circuito subtrator somador paralelo de 4 bits usando o processo do complemento de dois.



**Figura 9 - Circuito Subtrator Somador.**

O circuito da figura 9 tem o seguinte funcionamento:

**Quando a entrada de seleção estiver em  $\emptyset$ , a operação executada é a soma dos números  $A = A_3A_2A_1A_\emptyset$  e  $B = B_3B_2B_1B_\emptyset$ . Isto porque, com a entrada de seleção em  $\emptyset$  obtém-se a saída dos "OU EXCLUSIVOS" como ilustra a figura 10.**



**Figura 10 - Ilustração da saída do OU EXCLUSIVO com uma entrada  $\emptyset$ .**

Então, o circuito é um somador paralelo que efetua a soma de

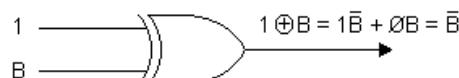
$$A_3A_2A_1A_\emptyset$$

+

$$B_3B_2B_1B_\emptyset$$


---

Por outro lado, quando a entrada de seleção estiver em 1, obtém-se a saída dos "OU EXCLUSIVOS" como ilustra a figura 11.



**Figura 11 - Ilustração da saída do OU EXCLUSIVO com uma entrada 1.**

Então, o circuito somador irá efetuar a soma de  $A_3A_2A_1A_\emptyset$  e  $\bar{B}_3\bar{B}_2\bar{B}_1\bar{B}_\emptyset$  e 1, sendo que o 1 entra no carry menos significativo.

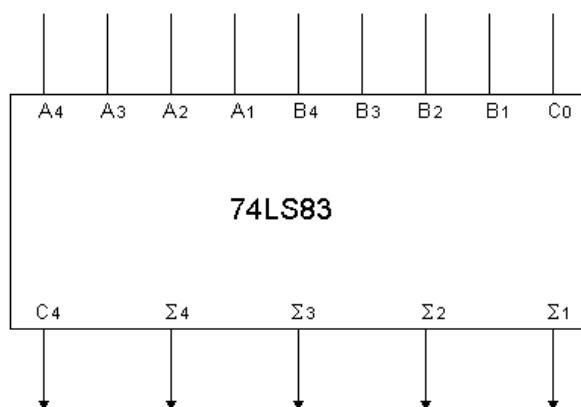
Isto é:

	$A_3A_2A_1A_\emptyset$		
+	$\bar{B}_3\bar{B}_2\bar{B}_1\bar{B}_\emptyset$	porém como	$\bar{B}_3\bar{B}_2\bar{B}_1\bar{B}_\emptyset$
+	<hr style="width: 100px; border: 0; border-top: 1px solid black; margin-bottom: 5px;"/>	1	+ <hr style="width: 100px; border: 0; border-top: 1px solid black; margin-bottom: 5px;"/>
			1

é o complemento de 2 de  $B_3B_2B_1B_\emptyset$ , a operação executada é a soma de "A" com o complemento de dois de "B", o que é a operação A-B.

### 11.7 SOMADORES PARALELOS NA FORMA DE CI's

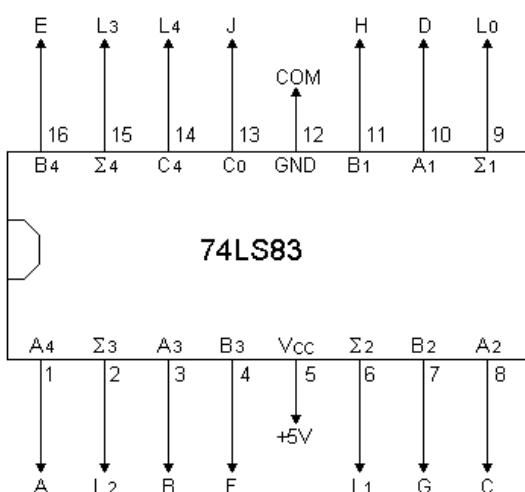
Existem na forma de CI's alguns somadores paralelos. Na linha TTL 74 vamos abordar o 74LS83, que é um somador de 4 bits e que aceita dois números de 4 bits (A e B) e um carry in ( $C_0$ ) como entradas. O 74LS83 produz uma soma na saída de 4 bits ( $\Sigma$ ) e uma saída carry ( $C_4$ ). A figura 12 mostra o diagrama funcional do 74LS83.



*Figura 12 - Diagrama Funcional do 74LS83.*

#### 11.7.1 EXPERIÊNCIA 5

- Material Utilizado: 1 x CI 74LS83
- Montar o circuito da figura 13.



*Figura 13 - Somador 74LS83.*

c) Completar a tabela 9.

A4	A3	A2	A1	B4	B3	B2	B1	C $\emptyset$	C4	$\Sigma$ 4	$\Sigma$ 3	$\Sigma$ 2	$\Sigma$ 1	
A	B	C	D	E	F	G	H	J	L4	L3	L2	L1	L $\emptyset$	
0	0	0	0	0	0	0	0	0						
0	0	0	1	0	0	0	0	0						
0	0	0	1	0	0	0	1	0						
0	0	0	1	0	0	1	0	0						
0	0	1	0	0	0	1	0	0						
0	0	1	0	0	0	1	1	0						①
0	0	1	1	0	0	1	1	0						
0	0	1	1	0	1	0	0	0						
0	1	0	0	1	0	0	0	0						
0	1	1	0	1	0	0	1	0						
1	0	0	0	1	0	0	0	0						②
1	0	0	1	1	0	0	0	0						
1	0	0	1	1	0	0	0	1						③
1	1	1	1	1	1	1	1	1						

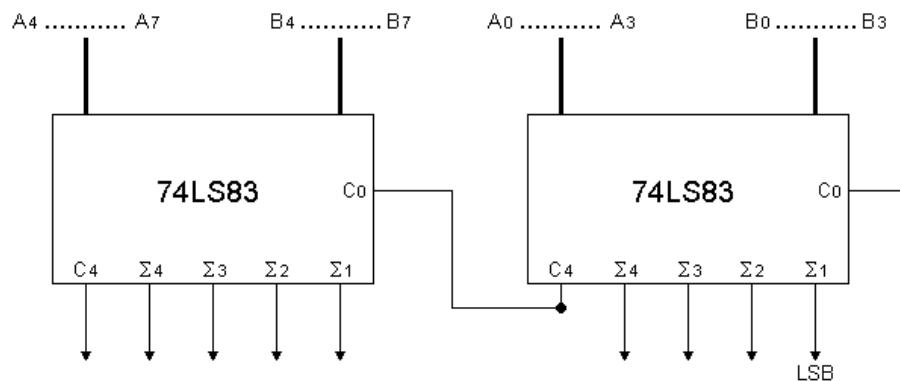
**Tabela 9 - Tabela do circuito da figura 13.**

d) Nesta tabela observar que em:

- ① Ocorre a soma de dois números A e B, tendo resultado menor ou no máximo igual a 1111
- ② Ocorre a soma de dois números A e B, tendo resultado maior do que 1111. Nestes casos  $C4 = L4 = 1$ .
- ③ Ocorre a soma de dois números A e B e mais  $C\emptyset$ , tendo resultado maior do que 1111.

## 11.8 LIGAÇÃO DE SOMADORES CI's PARALELOS EM CASCATA

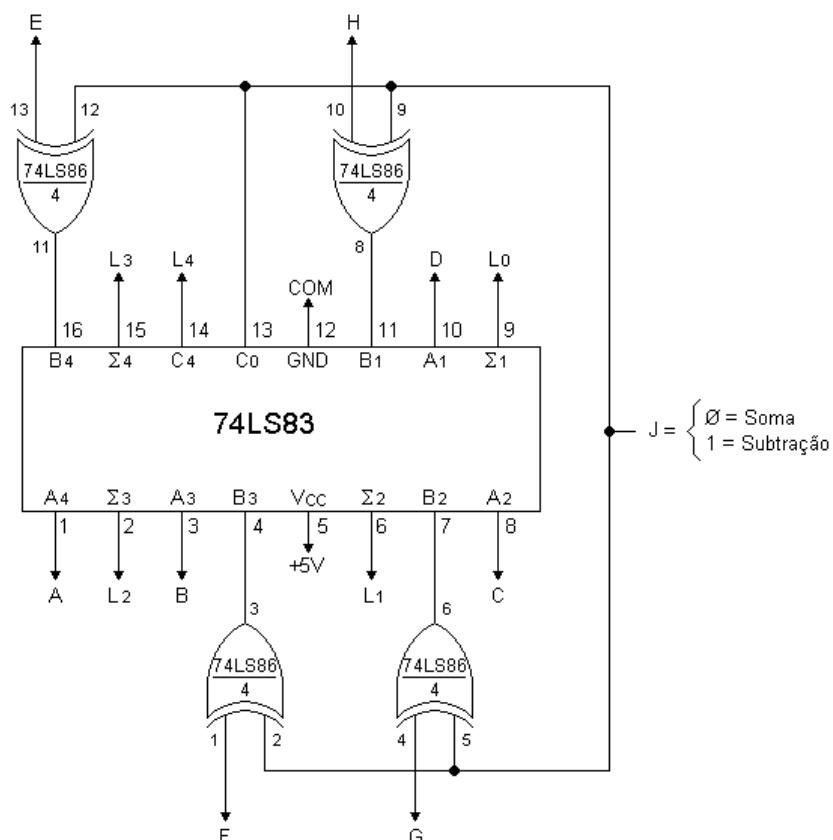
Para se construir um somador para números maiores do que 4 bits pode-se simplesmente ligar vários 74LS83 em cascata. Para isto, conecta-se o  $C4$  de um 74LS83 a entrada  $C\emptyset$  do próximo 74LS83 mais significativo, como mostrado na figura 14.



**Figura 14 - Ligação de CI's 74LS83 em cascata.**

### 11.8.1 EXPERIÊNCIA 6

- Material Utilizado: 1 x CI 74LS83 e 1 x CI 74LS86
- Montar o circuito da figura 15.



**Figura 15 - Subtrator Somador com 74LS83.**

c) Completar a tabela 10.

DEC	BINÁRIO				DEC	BINÁRIO				SELEÇÃO	SAÍDAS					
	A4	A3	A2	A1		B	B4	B3	B2	B1	Σ	C4	Σ4	Σ3	Σ2	Σ1
	A	B	C	D		E	F	G	H	J		L4	L3	L2	L1	LØ
Ø	Ø	Ø	Ø	Ø	2	Ø	Ø	1	Ø	Ø	Ø					
4	Ø	1	Ø	Ø	6	Ø	1	1	Ø	Ø	Ø					
9	1	Ø	Ø	1	7	Ø	1	1	1	Ø	Ø					
11	1	Ø	1	1	6	Ø	1	1	Ø	1						
11	1	Ø	1	1	8	1	Ø	Ø	Ø	1						

**Tabela 10 - Tabela do circuito da figura 15.**

d) Nesta tabela observar que:

- O carry-out (C4) deve ser abandonado.
- Em ①, CØ = Ø seleciona uma soma.
- Em ②, CØ = 1 seleciona uma subtração.

#### Observação:

Nos circuitos subtratores/somadores o primeiro bit é destinado a sinal, assim sendo não se deve invadir o local deste bit com bit de magnitude.

#### 11.8.2 EXPERIÊNCIA 7

A experiência seguinte se destina a soma e subtração de números relativos. O primeiro bit é o bit de sinal, um Ø indica um número positivo e um 1 indica um número negativo na forma de complemento de dois.

Usando o mesmo circuito da experiência 6, vamos procurar fazer a soma e a subtração de dois números. Devido ao uso de somente 3 bits para magnitude, serão usados números adequados, de modo a não haver ultrapassagem do número máximo ou mínimo possível.

#### Exemplo:

Escrever com 3 bits de magnitude os números sinalizados: +5, -5, +2 e -2. Em seguida, completar a tabela 11 e verificar-a pelo circuito da figura 15.

Solução:

$$\begin{array}{r}
 +5 = \emptyset 1 \emptyset 1 \\
 -5 = 1 \emptyset 1 \emptyset \\
 \hline
 1 \\
 1 \emptyset 1 1
 \end{array}
 \quad
 \begin{array}{r}
 +2 = \emptyset \emptyset 1 \emptyset \\
 -2 = 1 1 \emptyset 1 \\
 \hline
 1 \\
 1 1 1 \emptyset
 \end{array}$$

DEC	BINÁRIO				DEC	BINÁRIO				SELEÇÃO	SAÍDAS					
	A4 A3 A2 A1		B4 B3 B2 B1			C0		DEC			BINÁRIO					
	A	B	C	D		E	F	G	H		$\Sigma$	C4	$\Sigma 4$	$\Sigma 3$	$\Sigma 2$	$\Sigma 1$
+5					+2					$\emptyset$						
+5					-2					$\emptyset$						
-5					+2					$\emptyset$						
-5					-2					$\emptyset$						
+5					+2					1						
+5					-2					1						
-5					+2					1						
-5					-2					1						

**Tabela 11 - Operações com +5, -5, +2 e -2.**

Observação:

Na tabela 11, C4 deve ser abandonado e quando se tiver  $\Sigma 4 = \emptyset$  significa um número positivo, porém com  $\Sigma 4 = 1$ , tem-se um número negativo na forma de complemento de dois. Este circuito pode ser estendido para números maiores e é ideal para o uso em computadores onde os números são guardados na memória na forma de complemento de dois e, neste caso, necessitam somente de serem retirados e passados através de um subtrator-somador para se obter os resultados desejados.

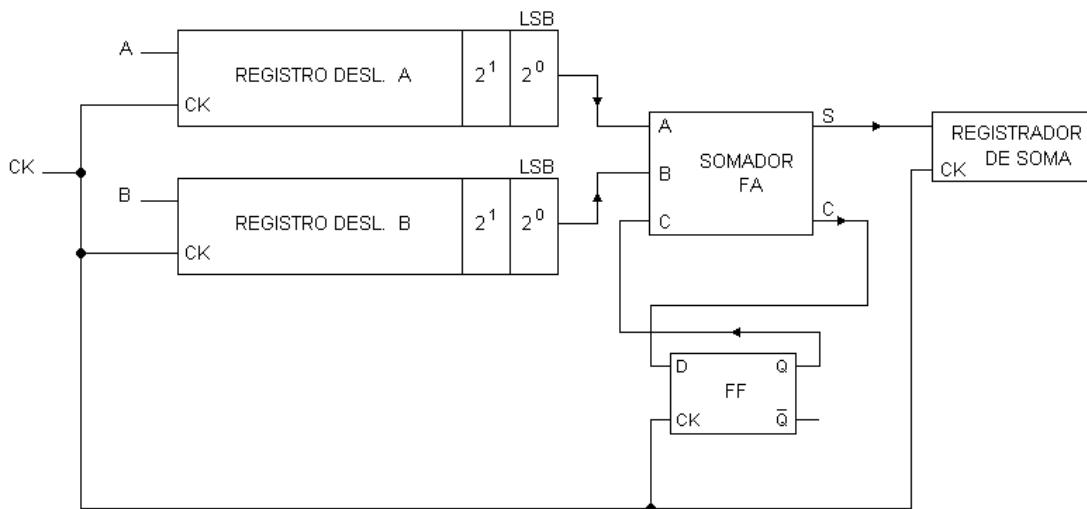
### 11.9 SOMADOR SÉRIE

Outro método de se efetuar a soma de números binários de vários bits é através de um somador série. Os números são somados em série, com os bits menos significativos sendo somados primeiro.

Após cada soma é gerado S e C. O C é guardado num flip-flop até ser executada a próxima adição.

A soma deve ser usada logo que cada bit fica disponível. Um registrador de deslocamento pode ser usado para guardar os bits até que toda a soma tenha sido efetuada.

A figura 16 mostra o circuito de um somador série.



**Figura 16 - Somador série.**

### 11.9.1 FUNCIONAMENTO DO SOMADOR SÉRIE

Os dados entram a partir de uma fonte externa, que pode ser uma memória, nos registradores A e B, sendo que em ambos registradores os bits menos significativos ficam a direita. O bit menos significativo de cada registrador é deslocado para o somador. Então, a soma é deslocada para o registrador de soma e os próximos 2 bits menos significativos dos registradores A e B são deslocados para o somador, no mesmo pulso de clock. Se a primeira adição produzir um carry, este é guardado no flip-flop e será uma entrada para o somador durante a próxima adição.

### 11.10 OVERFLOW E UNDERFLOW NA ARITMÉTICA DO COMPLEMENTO DE 2

Um overflow (ou sobrecarga) ocorre quando o resultado de uma operação aritmética é um número maior do que o registrador pode acomodar. Por outro lado, um underflow ocorre quando o resultado de uma operação aritmética é um número menor do que o registrador pode acomodar.

As limitações dos números que podem ser manuseados por um registrador de n bits são:  $2^{n-1} - 1$  números positivos e  $2^{n-1}$  números negativos. Assim sendo, um registrador de 8 bits pode acomodar números entre +127 e -128.

Para ilustrar o overflow considere a soma de dois números de 8 bits:

$$A = (1 \ 0 \ 1)1\emptyset = \emptyset \ 1 \ 1 \ \emptyset \ \emptyset \ 1 \ \emptyset \ 1 \quad \text{e}$$

$$B = (1 \ \emptyset \ 4)1\emptyset = \emptyset \ 1 \ 1 \ \emptyset \ 1 \ \emptyset \ \emptyset \ \emptyset$$

$$\text{O resultado é } (2\emptyset 5)1\emptyset = 1 \ 1 \ \emptyset \ \emptyset \ 1 \ 1 \ \emptyset \ 1$$

Infelizmente, quando se opera um registrador de 8 bits com complemento de 2 dispõe-se somente de 7 bits de magnitude, desde que o MSB é destinado à sinalização. Assim 205 está além da capacidade do registrador, já que seriam necessários 9 bits (com o MSB de sinal). Usando registrador de 8 bits, o resultado seria 1 1 0 0 1 1 0 1 na forma de complemento de 2. Isto significa -51, que é um resultado errado. Este fato acontece porque 205 está além da faixa de números que podem ser manuseados por um registrador de 8 bits.

### 11.10.1 OVERFLOW E UNDERFLOW NA ADIÇÃO

Se overflow e underflow podem causar problemas, deve-se construir circuitos para detectar esta condição.

As regras para detecção de overflow e underflow na adição são:

- a) Se dois números de sinais diferentes são somados nunca haverá overflow ou underflow. A soma nunca vai além da faixa permitida.
- b) Se dois números positivos são somados e tem-se um resultado negativo (MSB = 1), ocorreu um overflow.
- c) Se dois números negativos são somados e tem-se um resultado positivo (MSB =  $\emptyset$ ), ocorreu um underflow. O número negativo está abaixo da faixa permitida de números.

### 11.10.2 OVERFLOW E UNDERFLOW NA SUBTRAÇÃO

A operação de subtração pode também produzir overflow e underflow, mas somente se os operando são de sinais diferentes. Isto pode ocorrer, por exemplo nos seguintes casos:

- a) Se um número negativo é subtraído de um número positivo ocorre um overflow se o MSB de resultado for 1.
- b) Se um número positivo é subtraído de um número negativo ocorre um underflow se o MSB do resultado é  $\emptyset$ , indicando uma resposta positiva.

### 11.10.3 DETEÇÃO DE OVERFLOW E UNDERFLOW

A deteção de overflow e underflow em sistemas digitais pode ser necessária para avisar os dispositivos da ocorrência de tais condições e impedir que respostas erradas sejam tratadas como dados válidos.

Exemplo:

Projetar um circuito para detectar overflow no somador/subtrator complemento de 2.

Solução:

Um overflow somente ocorre nos seguintes casos:

- a) Na adição, onde o MSB dos operando são ambos  $\emptyset$  e o MSB da soma é 1 (que significaria que a soma de dois números positivos é negativa).
- b) Na subtração de  $(A-B)$  quando o operando A é positivo, o operando B é negativo e o resultado é negativo.

Fazendo-se a tabela destas condições do seguinte modo:

$x$  = BIT mais significativo do operando A

$y$  = BIT mais significativo do operando B

$R$  = BIT mais significativo do resultado

$s$  = seleção de operação

$s = 1$  = subtração  
 $s = \emptyset$  = soma

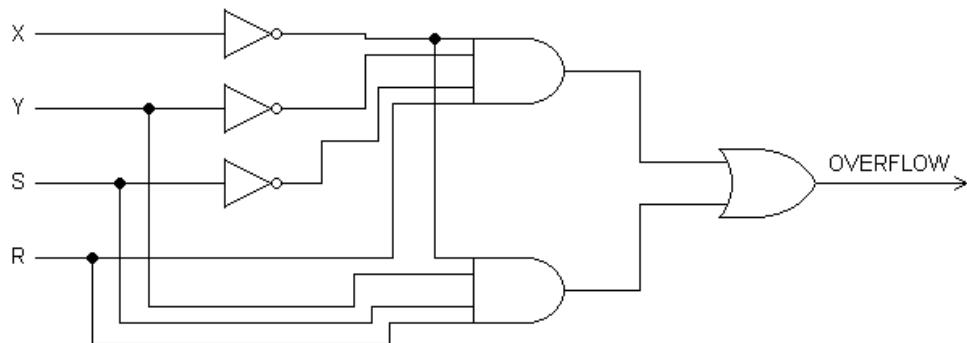
tem-se a tabela 12, da qual obtém-se por interpolação.

<b>x</b>	<b>y</b>	<b>s</b>	<b>R</b>	<b>OVERFLOW</b>
$\emptyset$	$\emptyset$	$\emptyset$	1	1
$\emptyset$	1	1	1	1

**Tabela 12 - Detecção de Overflow e Underflow num somador/subtrator.**

$$\text{Overflow} = \overline{x} \overline{y} \overline{s} R + \overline{x} y s R$$

Com esta expressão, pode ser construído o circuito da figura 17, que pode ser ligado diretamente aos circuitos somadores subtraidores.



**Figura 17 - Circuito detector de overflow em somador/subtrator.**

#### 11.10.4 EXPERIÊNCIA 8

Utilizando o circuito da experiência 6 (figura 15), complete a tabela 13 e observe a ocorrência de overflow e underflow.

DEC	BINÁRIO				DEC	BINÁRIO				SELEÇÃO	SAÍDAS					
	DEC		BINÁRIO			DEC		BINÁRIO			DEC		BINÁRIO			
A	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	B	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	C <sub>Ø</sub>	Σ	C <sub>4</sub>	Σ <sub>4</sub>	Σ <sub>3</sub>	Σ <sub>2</sub>	Σ <sub>1</sub>
	A	B	C	D		E	F	G	H	J		L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>
-5					+5					Ø	↑					
+5					-5					Ø						
+5					+5					Ø						
-5					-5					Ø						
-5					-5					1						
+5					+5					1						
+5					-5					1						
-5					+5					1	↓					

①  
 ②  
 ③  
 ④  
 ⑤  
 ⑥

**Tabela 13 - Exemplos de overflow e underflow.**

- ① – Não ocorre overflow nem underflow, pois soma números de sinais diferentes.
- ② – Ocorre overflow, pois o resultado de 5 +5 é um número positivo e a resposta obtida foi um número negativo (“ $\Sigma_4$ ” = 1).
- ③ – Ocorre underflow, pois o resultado de -5 +5 é um número negativo, e obteve-se um número positivo (“ $\Sigma_4$ ” = Ø).
- ④ – Não ocorre overflow nem underflow, pois se subtrai números de mesmos sinais.
- ⑤ – Ocorre overflow, pois o resultado -5 -5 é um número positivo, e obteve-se um número negativo (“ $\Sigma_4$ ” = 1).
- ⑥ – Ocorre underflow, pois o resultado de -5 -5 é um número negativo, e o resultado obtido foi positivo (“ $\Sigma_4$ ” = Ø).

### 11.11 ARITMÉTICA BCD

Para certas aplicações, tais como calculadoras, é vantajoso executar todas as operações no formato BCD. Isto elimina a necessidade de conversões, porém, infelizmente os circuitos aritméticos obtidos são mais complexos e mais caros do que os circuitos para operações com números binários puros.



## CAPÍTULO 12 - UNIDADES ARITMÉTICAS E LÓGICAS (ALU's)

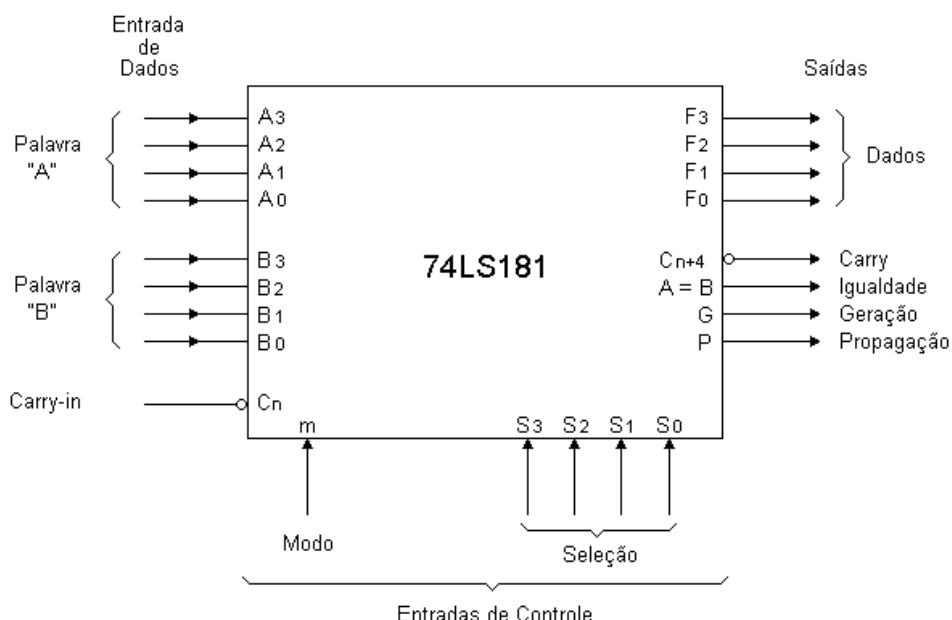
### 12.1 RESUMO

Em circuitos lógicos há necessidade de circuitos que possam executar diversas operações aritméticas tais como: soma, subtração, deslocamento de palavras, comparação de magnitude, geração de funções lógicas e outras. Tais circuitos existem na forma de CI's e são chamados Unidades Aritméticas e Lógicas (ALU's).

### 12.2 O 74LS181

O 74LS181 é a unidade aritmética e lógica (ALU) básica da série 74. A figura 1 mostra o seu lay-out funcional simplificado.

O 74LS181 aceita duas palavras de 4 bits, A e B, como entradas de dados e um **carry-in**, que age como um carry-in invertido durante operações de adição, pois é  $\emptyset$  quando ocorre um carry-in.



*Figura 1 - Lay-out Simplificado do 74LS181.*

Há também cinco entradas de controle que determinam as operações a serem executadas sobre as entradas. A entrada **modo** (*mode*) determina se a saída é uma função aritmética ou lógica das entradas. O **carry-in não afeta as funções lógicas**. As 4 linhas de seleção selecionam 1 das 16 possíveis operações aritméticas e lógicas.

As saídas do 74LS181 são as saídas dos 4 bits do resultado (saídas  $F_3 F_2 F_1 F_0$ ), a saída **carry-out  $C_{n+4}$** , a saída  $A = B$ , e as saídas **Geração e Propagação** (destinadas a conexões *Look AHEAD Carry* entre ALU's). As saídas são determinadas de acordo com a tabela funcional, onde deve-se notar que o sinal **+** significa a função lógica **OU** e a palavra **mais** significa a soma das entradas.

A vantagem do 74LS181 é que pode executar as operações **adição, subtração, deslocamento (uma casa), E, OU, OU Exclusivo, e outras operações sobre as variáveis de entrada**, simplesmente mudando-se as entradas de controle, **modo e seleção**.

Na subtração um **Carry-out  $C_{n+4} = \emptyset$**  indica um resultado **positivo ou zero** e  **$C_{n+4} = 1$**  indica um resultado **negativo ou um empréstimo**.

Se o resultado da operação **MENOS for negativo é apresentado como um número de 4 bits na forma de complemento de 2. Por exemplo, se o resultado for -5, as saídas F acusam  $1\emptyset 11$  e o Carry-out é 1.**

A tabela 1 mostra a funcionalidade de um 74LS181.

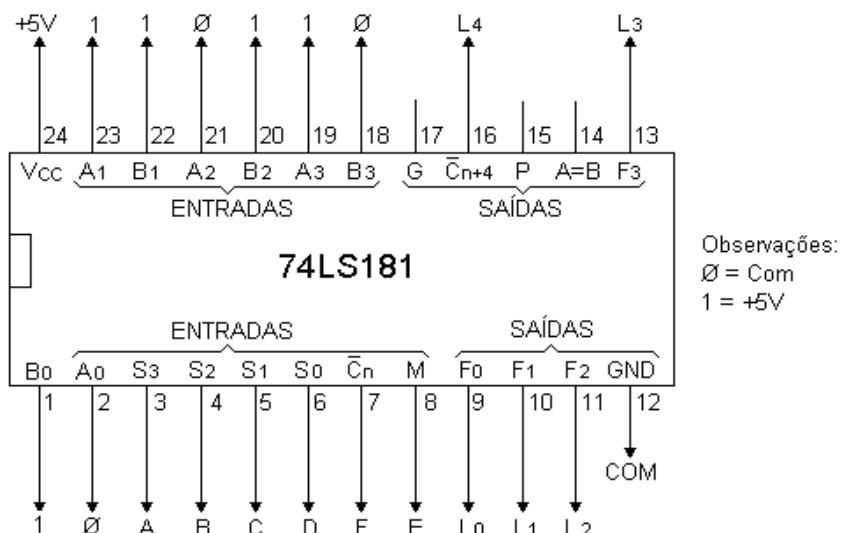
Seleção				ATIVO - DADOS ALTO		
				M = H Funções Lógicas	M = L : Operações Aritméticas	
$\bar{C}_n = H$ (Sem Carry)		$\bar{C}_n = L$ (Com Carry)				
S3	S2	S1	S0			
L	L	L	L	$F = \bar{A}$	$F = A$	$F = A$ mais 1
L	L	L	H	$F = \bar{A} + B$	$F = A + B$	$F = (A+B)$ mais 1
L	L	H	L	$F = \bar{A}\bar{B}$	$F = A + \bar{B}$	$F = (A+\bar{B})$ mais 1
L	L	H	H	$F = \emptyset$	$F = \text{menos } 1 \text{ (comp. 2)}$	$F = \text{zero}$
L	H	L	L	$F = AB$	$F = A$ mais $\bar{A}\bar{B}$	$F = A$ mais $\bar{A}\bar{B}$ mais 1
L	H	L	H	$F = \bar{B}$	$F = (A+B)$ mais $\bar{A}\bar{B}$	$F = (A+B)$ mais $\bar{A}\bar{B}$ mais 1
L	H	H	L	$F = A \oplus B$	$F = A$ menos $B$ menos 1	$F = A$ menos $B$
L	H	H	H	$F = \bar{A} \oplus B$	$F = AB$ menos 1	$F = \bar{A}\bar{B}$
H	L	L	L	$F = \bar{A} + B$	$F = A$ mais $AB$	$F = A$ mais $AB$ mais 1
H	L	L	H	$F = A \oplus B$	$F = A$ mais $B$	$F = A$ mais $B$ mais 1
H	L	H	L	$F = B$	$F = (A+\bar{B})$ mais $AB$	$F = (A+\bar{B})$ mais $AB$ mais 1
H	L	H	H	$F = AB$	$F = AB$ menos 1	$F = AB$
H	H	L	L	$F = 1$	$F = A$ mais $A^*$	$F = A$ mais $A$ mais 1
H	H	L	H	$F = A + \bar{B}$	$F = (A+B)$ mais $A$	$F = (A+B)$ mais $A$ mais 1
H	H	H	L	$F = A + B$	$F = (A+\bar{B})$ mais $A$	$F = (A+\bar{B})$ mais $A$ mais 1
H	H	H	H	$F = A$	$F = A$ menos 1	$F = A$

(\*) Cada bit é deslocado para a próxima posição mais significativa.

Tabela 1 – funcionalidade de um 74LS181.

### 12.2.1 EXPERIÊNCIA 1

- a) Material Utilizado: 1 x CI 74LS181
- b) Montar o circuito da figura 2.



**Figura 2 - CI 74LS181**

- c) No circuito da figura 2 observar que a palavra de dados A ( $A_3 A_2 A_1 A_0$ ) foi fixada no valor de 1 Ø 1 Ø e que a palavra de dados B ( $B_3 B_2 B_1 B_0$ ) foi fixada no valor Ø 1 1 1.
- d) Utilizando a tabela 1, completar a tabela 2 para executar a função indicada. Selecionar as operações e verifique o resultado obtido.

Operação	SELEÇÃO								SAÍDAS			
	S <sub>3</sub> A	S <sub>2</sub> B	S <sub>1</sub> C	S <sub>Ø</sub> D	M E	C <sub>n</sub> F	C <sub>n+4</sub> L <sub>4</sub>	F <sub>3</sub> L <sub>3</sub>	F <sub>2</sub> L <sub>2</sub>	F <sub>1</sub> L <sub>1</sub>	F <sub>Ø</sub> L <sub>Ø</sub>	
F = $\bar{A}$							Ø					
F = $\bar{B}$							Ø					
F = A . B							Ø					
F = A $\oplus$ B							Ø					
F = A + B							Ø					
F = A + B							1					

①

$F = A+B$ mais 1		$\emptyset$			②
$F = A$ mais B		1			
$F = A$ mais B mais 1		$\emptyset$			
$F = A$ menos 1		1			
$F = A$		$\emptyset$			
$F = A$ menos B menos 1		1			
$F = A$ menos B		$\emptyset$			

**Tabela 2 - Tabela do Circuito da figura 2**

- e) Em ① tem-se apenas operações lógicas, e  $M = 1$ .  
Em ② tem-se operações lógicas e aritméticas, e  $M = \emptyset$ .
- f) Com o auxílio da tabela 1, executar outras funções que considerar importante.

### 12.3 USO DO COMPARADOR DO 74LS181

Considere-se o caso da comparação de dois números positivos A e B e quando se deseja determinar qual deles é o maior. Para isto, coloca-se  $M = \emptyset$  e  $S_3, S_2, S_1$  e  $S_0 = \emptyset 1 \emptyset \emptyset$  e  $\bar{C}_n = 1$ . Na tabela de funcionamento do 74LS181 acha-se " $F = A$  menos  $B$  menos 1".

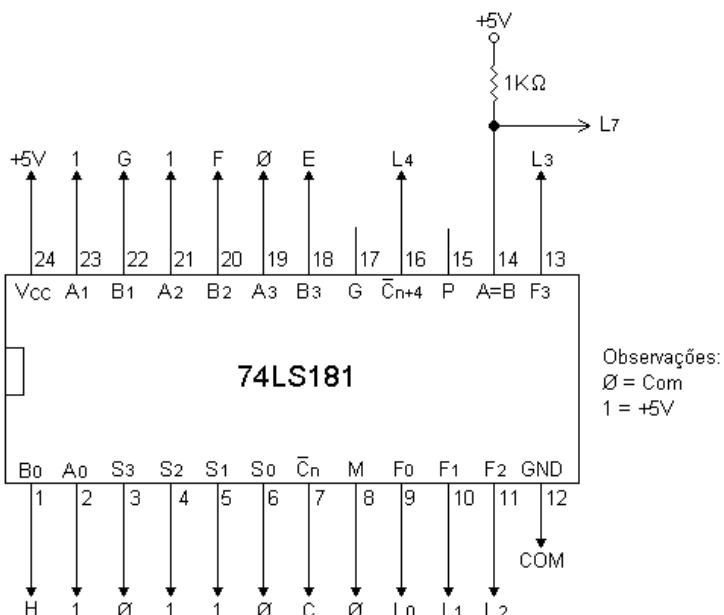
Nestas condições, se  $A > B$  o bit de sinal de  $\bar{C}_{n+4}$  será  $\emptyset$ . No 74LS181 os números negativos são apresentados na forma de complemento de 2; então, se  $A \leq B$ , F será negativo e  $\bar{C}_{n+4} = 1$ . Se as entradas A e B forem iguais, a saída  $A = B$  será 1. A tabela 3 mostra como interpretar as magnitudes relativas, dependendo do valor da saída  $C_{n+4}$ .

ENTRADA $\bar{C}_n$	SAÍDA $\bar{C}_{n+4}$	SIGNIFICADO
1	1	$A \leq B$
1	$\emptyset$	$A > B$
$\emptyset$	1	$A < B$
$\emptyset$	$\emptyset$	$A \geq B$

**Tabela 3 – Comparador.**

### 12.3.1 EXPERIÊNCIA 2

- a) Material Utilizado: 1 x CI 74LS181 e 1 Resistor de  $1\text{ K}\Omega$   
b) Montar o circuito da figura 3.



**Figura 3 - 74LS181 com Comparador.**

- c) Completar a tabela 4, observando que  $S_3, S_2, S_1$  e  $S_\emptyset = \emptyset 11\emptyset$  e  $M = \emptyset$  selecionam a operação do tipo "A menos B" se  $\bar{C}_n = \emptyset$ , e "A menos B menos 1" se  $\bar{C}_n = 1$ . Note também que A ( $A_3, A_2, A_1$  e  $A_\emptyset$ ) foi fixado em  $\emptyset 111 = 7_{10}$ .

ENTRADAS								SAÍDAS															
①	BINÁRIO				DEC	BINÁRIO				DEC	CARRY	$\bar{C}_n$	$\bar{C}_n+4$	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>	DEC	A=B				
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>					C	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>				
①	0	1	1	1	+7	0	1	0	0	+4	1												
②	0	1	1	1	+7	0	1	0	1	+5	1												
③	0	1	1	1	+7	0	1	1	0	+6	1												
①	0	1	1	1	+7	0	1	1	1	+7	1												
②	0	1	1	1	+7	0	1	0	0	+4	0												
③	0	1	1	1	+7	0	1	0	1	+5	0												
①	0	1	1	1	+7	0	1	1	0	+6	0												
②	0	1	1	1	+7	0	1	1	1	+7	0												
③	0	1	1	1	+7	0	1	1	1	+7	0												

**Tabela 4 - 74LS181 como comparador.**

d) Na tabela 4, observar que nos itens ① e ② tem-se  $C_n = 1$ , então a saída  $A = B$  nos dá informação de comparação. Assim sendo, no item ② tem-se  $A = 7$  e  $B = 7$  e a saída  $(A = B) = 1$ .  $\bar{C}_n$  e  $\bar{C}_{n+4}$  fornecem informações relativas a comparação, assim em 1 a operação selecionada é "A menos B menos 1" pois  $\bar{C}_n = 1$ , e como  $\bar{C}_{n+4} = \emptyset$  conclui-se que  $A > B$  (o que está de acordo com a tabela 3).

Em ③  $\bar{C}_n = \emptyset$ , logo a saída  $A = B$  não dá informação de comparação. Tal informação pode ser obtida através dos valores de  $\bar{C}_n$  e  $\bar{C}_{n+4}$ . Neste caso, como  $\bar{C}_n = \emptyset$  e  $\bar{C}_{n+4} = \emptyset$  conclui-se da tabela 3 que  $A \geq B$ .

## **CAPÍTULO 13 - MONOESTÁVEIS E ASTÁVEIS**

### **13.1 INTRODUÇÃO**

Basicamente tem-se dois tipos de multivibradores: os monoestáveis e os astáveis ou osciladores.

Como o próprio nome indica, um monoestável tem somente um estado estável (geralmente  $\emptyset$ ) e, quando sofre um pulso de disparo, sua saída vai para o estado quase-estável (1), permanecendo neste estado por um tempo fixo (tempo este definido, normalmente, por resistores e capacitores).

Os monoestáveis dividem-se em dois tipos: regatilháveis e não regatilháveis. No primeiro, ocorrendo um pulso de disparo enquanto a saída está no estado quase-estável, o tempo sofre um reset e a saída ainda permanece neste estado pelo tempo definido pelo resistor e capacitor. Já para os não regatilháveis isto não acontece, ou seja, aplicando-se um disparo quando o monoestável está no estado quase-estável, este pulso de disparo será ineficaz.

Quando se trabalha com monoestáveis não regatilháveis, deve-se levar em conta o ciclo de trabalho ligado, o qual não pode ser ultrapassado sob pena de ocorrer *Jitter* (oscilação na duração do pulso de saída). Entende-se como ciclo de trabalho ligado a relação:

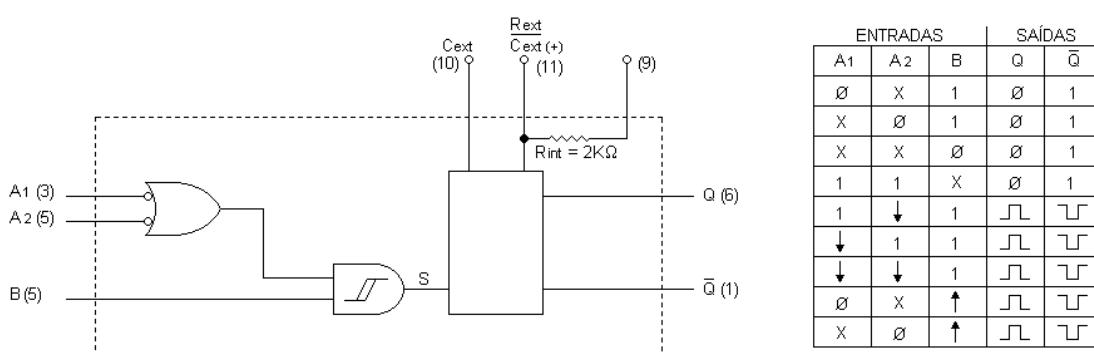
$$\text{Ciclo de traba.lig.} = \frac{\text{Tempo Ligado}}{\text{Tempo ligado} + \text{Tempo desligado}}$$

Os monoestáveis são usados para determinação de tempo e controle de sequência de eventos; os astáveis (osciladores) são utilizados para gerar pulsos de clock, necessários para controlar a temporização de circuitos digitais. Existem comercialmente osciladores de cristal com alta estabilidade, na forma de circuitos integrados e com saída compatível com TTL. Se a tolerância em uma determinada frequência não for extremamente crítica, pode-se construir osciladores usando-se circuitos TTL, circuitos estes que podem ser os monoestáveis.

## 13.2 MONOESTÁVEIS COM CI's

### 13.2.1 O 74121

O 74121 é um monoestável não regatilhável em um dispositivo CI completo, exceto para os componentes de temporização que devem ser conectados externamente. A figura 1 mostra o diagrama e a tabela de funcionamento deste CI.



**Figura 1 - Diagrama e tabela de funcionamento do 74121.**

Pela figura 1, vê-se que para disparar o 74121 deve-se criar um edge positivo no ponto “S”. Assim, o 74121 podem ser disparados de dois modos:

- Uma ou ambas as entradas A ( $A_1$  e  $A_2$ ) estão em  $\emptyset$  e a entrada B recebe um edge positivo;
- A entrada B está em 1 e uma das entradas A recebe um edge negativo enquanto a outra entrada A permanece em 1, ou ambas as entradas A recebem um edge negativo, simultaneamente.

A duração do pulso de saída é determinada pelo resistor e pelo capacitor. Para o resistor, tem-se duas possibilidades:

- O resistor interno (valor nominal  $2\text{ K}\Omega$ ). Para tal conecta-se o pino a VCC;
- Um resistor externo (entre  $1.4\text{ K}\Omega$  e  $40\text{ K}\Omega$ ). Este resistor deve ser conectado entre o pino 11 e VCC, estando o pino 9 flutuando.

O capacitor deve ser colocado entre os pinos 11 e 10 (no caso de capacitor eletrolítico, ligar o terminal positivo ao pino 11) e pode variar desde circuito aberto entre 10 e 11 (capacitância de fuga) até  $1000\text{ }\mu\text{F}$ .

A duração do pulso de saída do 74121 pode ser obtida por

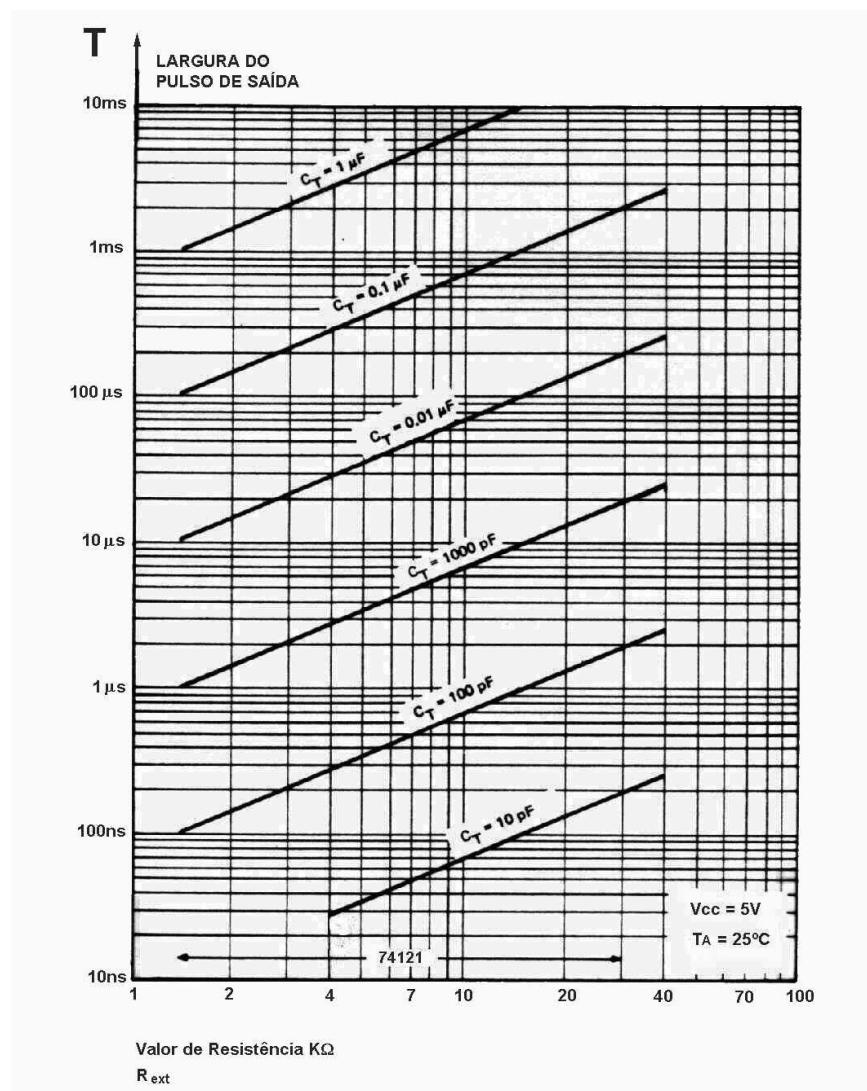
$$T = 0,7 C_T R_T$$

ou pelo gráfico da figura 2, lembrando que na fórmula anterior:

$T$  = largura do pulso de saída [ns]

$C_T$  = capacidade de temporização [pF]

$R_T$  = resistência de temporização [KΩ]



**Figura 2 - Largura dos pulsos de saída do 74121 em função dos componentes de temporização.**

Escolhendo-se valores adequados para os componentes de temporização, pode-se ter a duração do tempo de saída variando entre 30 ns e 28 segundos.

Convém lembrar-se de três fatos:

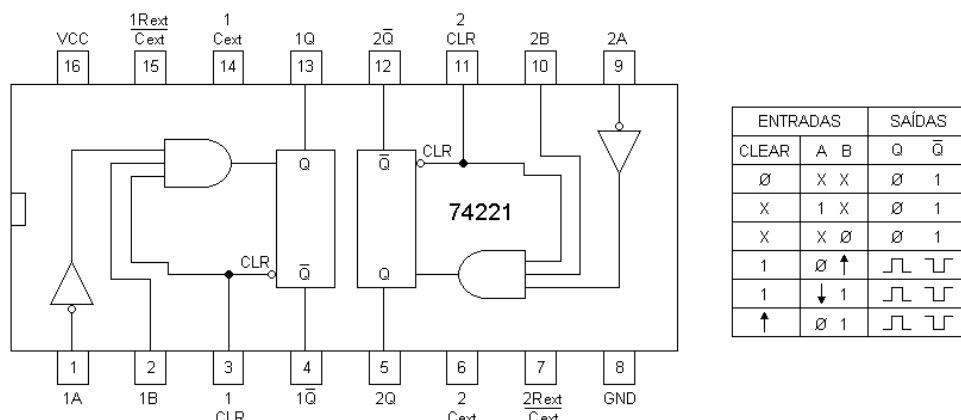
- Devido a variações entre CI's e à tolerância de resistores e capacitores, recomenda-se testar e ajustar o tempo dos pulsos de saída após o cálculo;
- A entrada B do 74121 é a entrada de um *Schmitt Trigger* (responde até 1V/S) e, se formas de ondas lentas são requeridas para disparar um monoestável, elas devem ser aplicadas à entrada B;
- O 74121 sendo não regatilhável têm limitações de ciclo de trabalho, sendo seu ciclo de trabalho máximo de 90% e de 67% para resistores de 40 KΩ e 2 KΩ, respectivamente.

### 13.2.2 O 74221

O CI 74221 é praticamente dois 74121 em um único chip, apresentando apenas como diferenças:

- Não tem resistor interno;
- Possui apenas uma entrada A para cada monoestável;
- Possui uma entrada clear, que permite interromper o pulso de saída quando ela está no estado quase-estável.

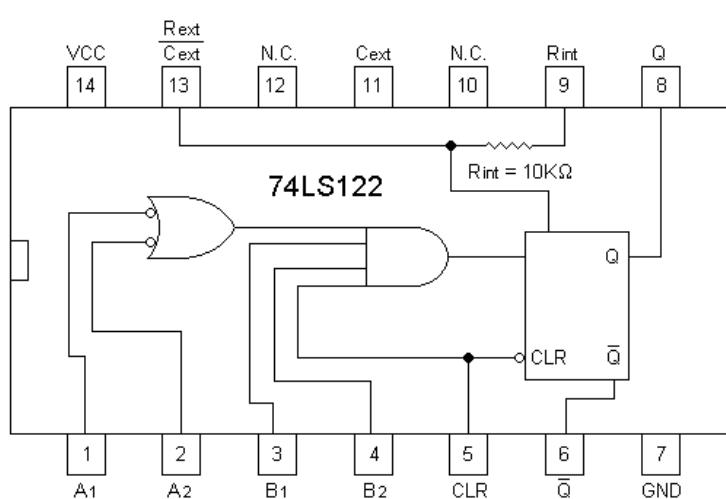
O funcionamento do 74221 é idêntico ao do 74121, tendo a duração do seu pulso de saída as mesmas restrições e a mesma forma de cálculo. O diagrama e a tabela de funcionamento do 74221 estão mostrados na figura 3.



**Figura 3 - Diagrama e tabela de funcionamento do 74221.**

### 13.2.3 O 74LS122

O 74LS122 é um monoestável regatilhável numa embalagem DIP 14 pinos. Seu diagrama e tabela de funcionamento estão mostrados na figura 4. O 74LS122 possui um resistor interno de  $10\text{ k}\Omega$ .



CLEAR	ENTRADAS				SAÍDA
	A1	A2	B1	B2	
Ø	X	X	X	X	Ø
X	1	1	X	X	Ø
X	X	X	Ø	X	Ø
X	X	X	X	Ø	Ø
X	Ø	X	1	1	Ø
1	Ø	X	↑	1	↓
1	Ø	X	1	↑	↓
1	X	Ø	1	1	Ø
1	X	Ø	↑	1	↓
1	X	Ø	1	↑	↓
1	1	↓	1	1	↓
1	↓	↓	1	1	↓
1	↓	1	1	1	↓
↑	Ø	X	1	1	↓
↑	X	Ø	1	1	↓

**Figura 4 - Diagrama e tabela de funcionamento do 74LS122.**

Para se usar o resistor interno deve-se ligar o pino 9 a VCC e para se usar o resistor externo, deve-se ligar este entre os pinos 13 e VCC (este resistor deve estar entre  $5\text{ k}\Omega$  e  $50\text{ k}\Omega$ ).

O capacitor deve ser ligado entre os pinos 11 e 13 e não tem limite. Caso o capacitor seja eletrolítico, o terminal positivo deve ser ligado ao pino 13.

Os pontos importantes da tabela da figura 4 são:

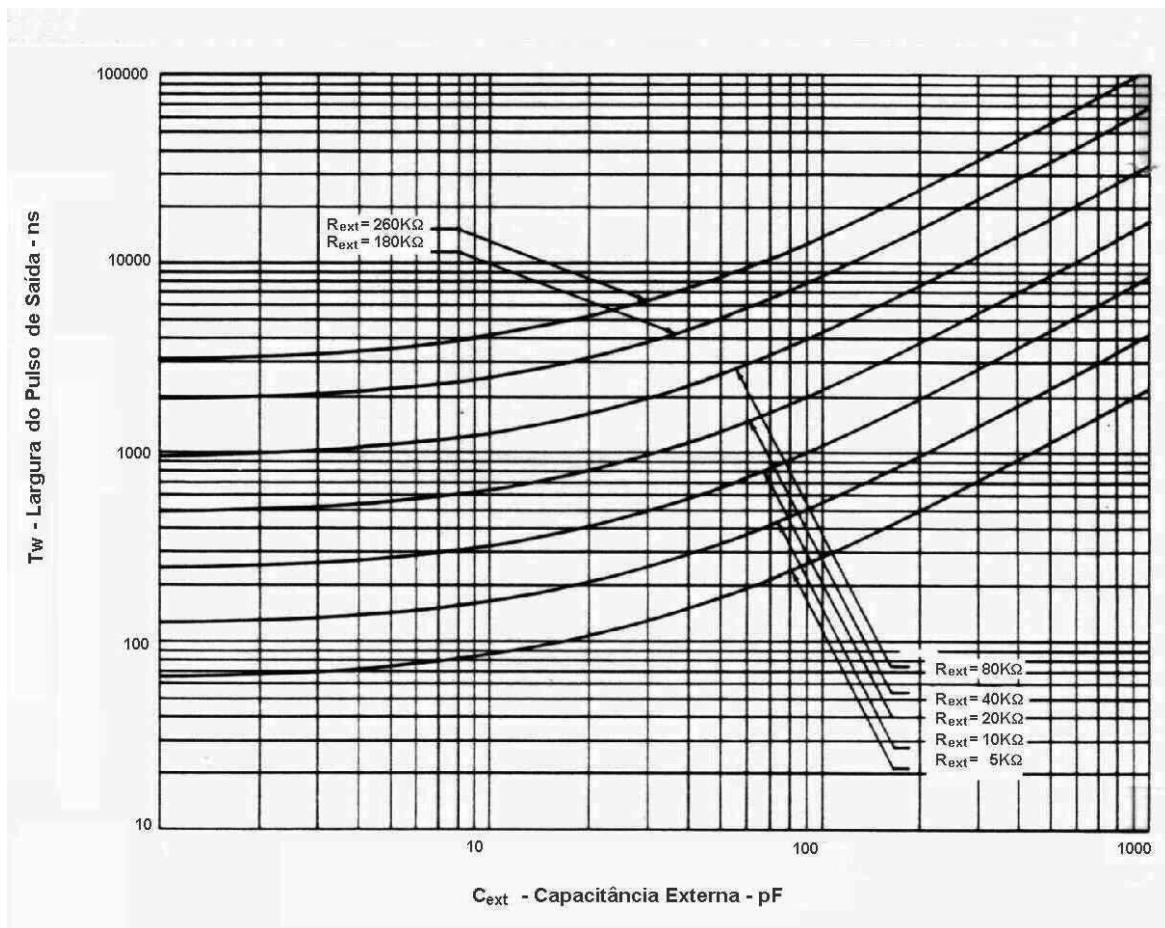
- a) Se a linha clear é Ø, a saída é resetada (Ø);
- b) Um edge positivo na saída do bloco “E” do CI é necessário para disparar o monoestável;
- c) Quaisquer condições de entrada que levem a saída do bloco “E” do CI a ser constante não dispara o monoestável.

Convém lembrar que as entradas “B” do 74LS122 não são entradas de um *Schmitt Trigger* e, caso se deseje disparo com formas de ondas lentas, deve-se usar o 74LS122 que possui entradas “B” respondendo até  $0,1\text{V}/\mu\text{s}$ .

A seleção do pulso de saída do 74LS122 pode ser realizada de duas maneiras:

- Se o capacitor é menor que 1000 pF usa-se o gráfico da figura 5;
- Se o capacitor é maior que 1000 pF usa-se a seguinte fórmula

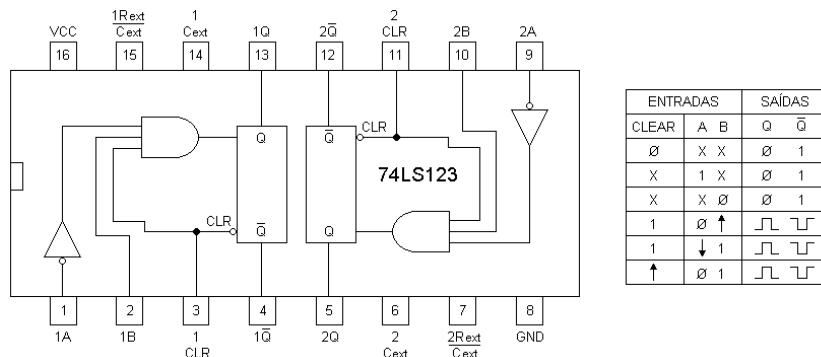
$$T = 0,32 \frac{R_T C_T}{R_T} (1 + 0,7), \text{ com } T [\text{ns}], R_T [\text{K}\Omega] \text{ e } C_T [\text{pF}]$$



**Figura 5 - Largura da saída do 74LS122 e 74LS123 para  $C_T < 1000\text{pF}$ .**

#### 13.2.4 O 74LS123

O 74LS123 é um CI onde tem-se dois monoestáveis regatilháveis e nenhum resistor interno. Cada monoestável funciona de maneira muito próxima ao 74LS122. O diagrama e a tabela de funcionamento estão na figura 6.



**Figura 6 - Diagrama e tabela de funcionamento do 74LS123.**

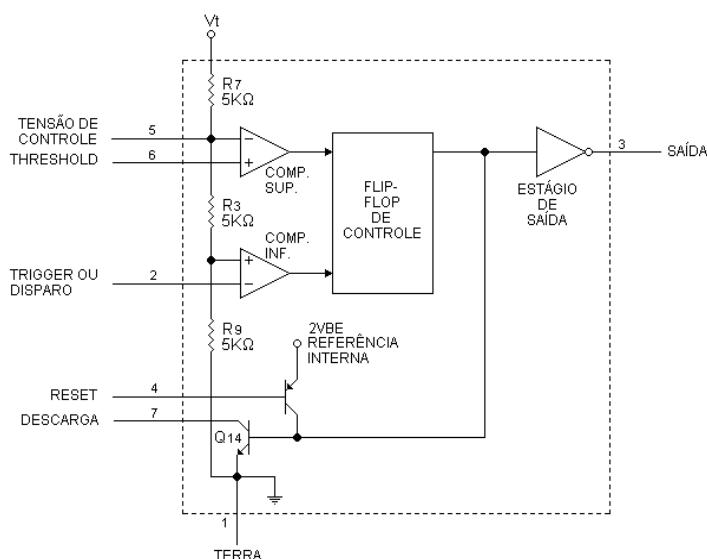
As limitações do resistor externo são as mesmas que do 74LS122 e a duração do pulso de saída é calculada da seguinte forma:

- Se  $C_T > 1000 \text{ pF}$ , usa-se a curva tracejada da figura 5;
- Se  $C_T < 1000 \text{ pF}$ , usa-se a fórmula

$$T = 0,28 \frac{R_T C_T (1 + 0,7)}{R_T}$$

### 13.2.5 O 555

O 555 é um circuito integrado que serve a vários propósitos. Aqui ele será analisado atuando como monoestável e, posteriormente, como astável. Este CI pode ter sua fonte de alimentação variando desde  $\emptyset$  até 15V e quando  $VCC = 5V$ , ele é plenamente compatível com TTL. O diagrama funcional do 555 está mostrado na figura 7.



**Figura 7 - Diagrama funcional do 555.**

O 555 opera do seguinte modo:

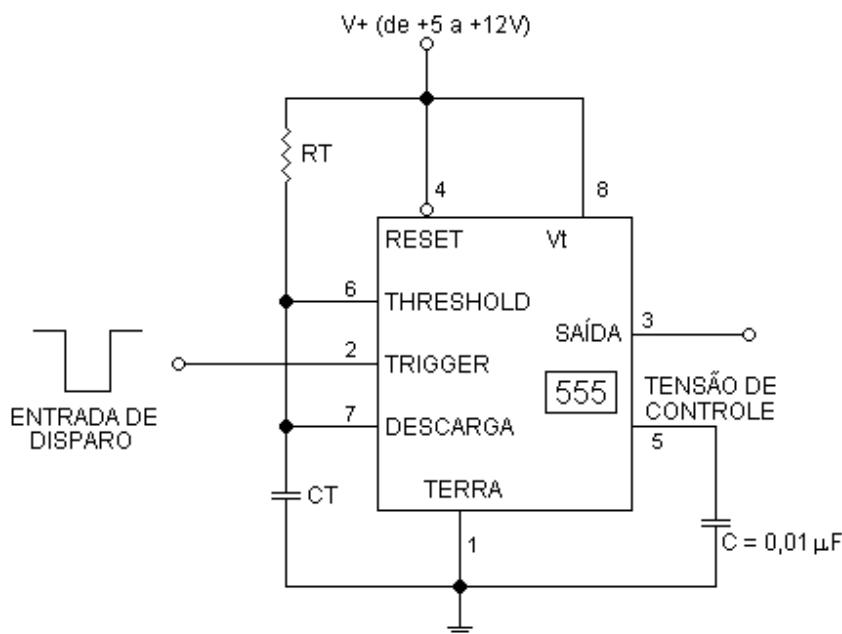
- Toda vez que a tensão no pino 2 (*trigger*) cai abaixo da tensão na entrada positiva do comparador inferior, a saída vai para nível alto (1);
- Toda vez que a tensão do pino 6 (*threshold*) vai a um valor acima da tensão no pino 5 (tensão de controle), a saída vai para nível baixo ( $\emptyset$ );
- A entrada *reset* é ativa quando seu nível é baixo ( $\emptyset$ );
- Quando não se usa o pino 5 (tensão de controle), este deve ser ligado para a terra através de um capacitor de  $0,01 \mu\text{F}$ , para imunidade de ruído;
- O comparador inferior possui um *storage time* da ordem de  $10 \mu\text{s}$ , e isto significa que a largura mínima de pulso será desta ordem ( $10 \mu\text{s}$ ).

A figura 8 ilustra o 555 ligado como monoestável. Deve-se observar que o pulso de disparo deve ser negativo e que também deve ter uma duração menor que o tempo do pulso de saída.

Para a configuração mostrada na figura 8, a duração do tempo de saída é:

$$T = 1,1 R_t C_t$$

Lembrando, que o limite inferior de T é de  $10 \mu\text{s}$ .



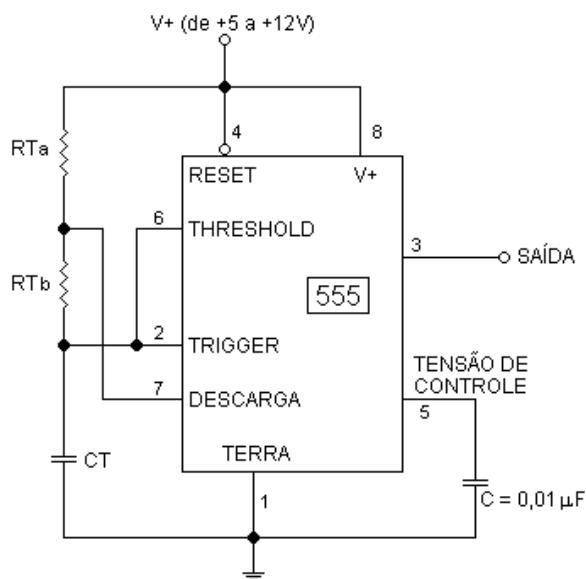
**Figura 8 - O 555 como monoestável.**

Um limite inferior razoável para  $R_T$  é de  $10\text{ K}\Omega$ , embora  $R_T$  possa ser menor que  $10\text{ K}\Omega$  (este limite é aconselhável devido ao aspecto econômico).

Um limite inferior e prático para  $C_T$  é de  $100\text{ pF}$ , pois abaixo disto os efeitos da capacidade de fuga se tornam apreciáveis, limitando a precisão.

O limite superior de  $R_T$  é da ordem de  $13\text{ M}\Omega$ , pois é boa prática selecionar  $R_T$  de modo que uma queda de tensão maior que  $1/3\text{ VCC}$  seja provocada por uma corrente de 10 a 100 vezes o valor da corrente de fuga mais a corrente threshold. Assim, o limite superior de  $C_T$  dependerá de sua corrente de fuga.

Como o 555 pode operar sozinho como astável, esta sua aplicação já será tratada aqui. A figura 9 mostra o 555 ligado como astável ou oscilador.



**Figura 9 - O 555 como astável.**

As restrições de operação do 555 como astável são poucas e similares a operação monoestável. O limite superior da frequência é da ordem de  $100\text{ KHz}$  (devido ao storage time) e o limite inferior é imposto, praticamente, pelas limitações de R e C. Os limites de  $C_T$  são idênticos ao do monoestável. O máximo valor de  $R_{Ta} + R_{Tb}$  é o mesmo que o de  $R_T$  do monoestável, ou seja da ordem de  $13\text{ M}\Omega$ .

As fórmulas para o 555 como multivibrador astável na configuração da figura 9 são:

$$T = 0,693 (R_{Ta} + R_{Tb}) C_T$$

$$T = 0,693 R_{Tb} C_T$$

$$f = \frac{1}{T} = \frac{1}{T_1 + T_2} = \frac{1,44}{(R_{Ta} + 2R_{Tb}) C_T}$$

onde:

T1 é o tempo ligado (saída alta)

T2 é o tempo desligado (saída baixa)

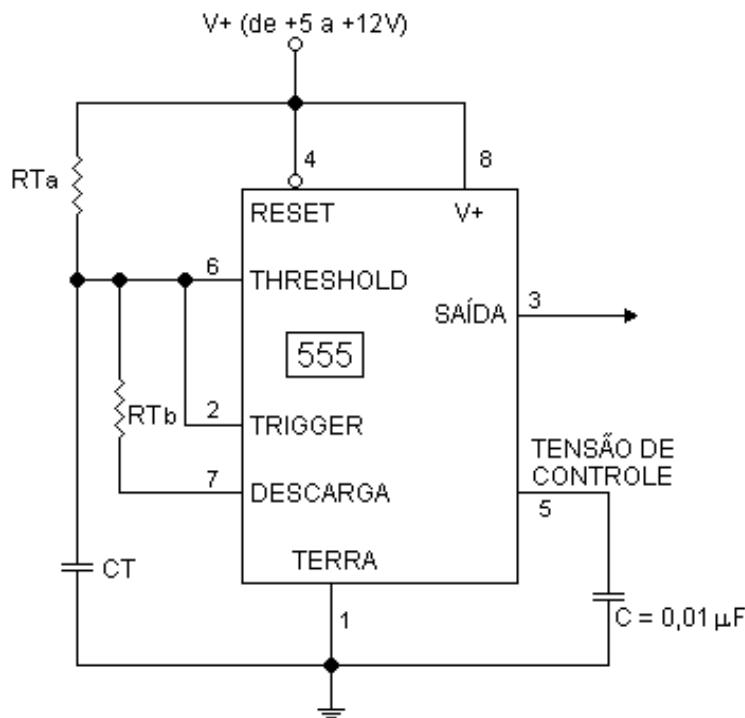
Neste caso, observa-se que quando  $R_{Tb}$  se torna maior do que  $R_{Ta}$ , o ciclo de trabalho se aproxima de 50% (onda quadrada). Por outro lado, quando  $R_{Ta}$  se torna maior que  $R_{Tb}$ , o ciclo de trabalho aumenta, no sentido de 100%.

Outra configuração para o 555 operando como astável é a mostrada na figura 10. Nesta configuração, as fórmulas são:

$$T = 0,693 R_{Ta} C_T$$

$$T = \frac{R_{Ta} \cdot R_{Tb} \cdot C_T \cdot \ln \frac{R_{Tb} - 2R_{Ta}}{2R_{Tb} - R_{Ta}}}{R_{Ta} + R_{Tb}}$$

e  $R_{Tb}$  deve ser menor  $R_{Ta}/2$ , pois caso contrário a tensão no pino 2 permanecerá acima de  $1/3 VCC$ , não disparando o CI.

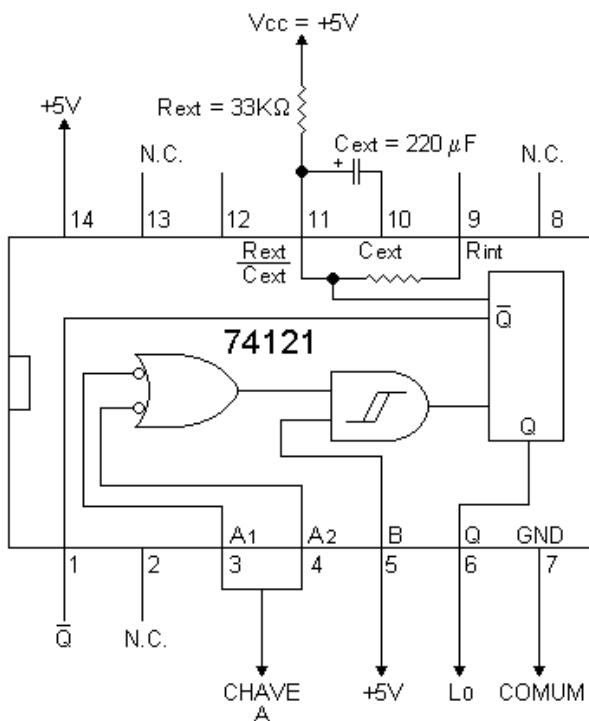


**Figura 10 - Outra possibilidade do 555 como astável.**

Com a configuração mostrada na figura 10, é possível obter-se onda quadrada com o 555 (p.ex.  $R_{Ta} = 51 K\Omega$  e  $R_{Tb} = 22 K\Omega$ ).

### 13.2.6 EXPERIÊNCIA 1

- a)** Material Utilizado: 1 x CI 74121, 1 Resistor de  $33\text{ K}\Omega$  e 1 Capacitor de  $220\text{ }\mu\text{F}$ .
- b)** Montar o circuito da figura 11.

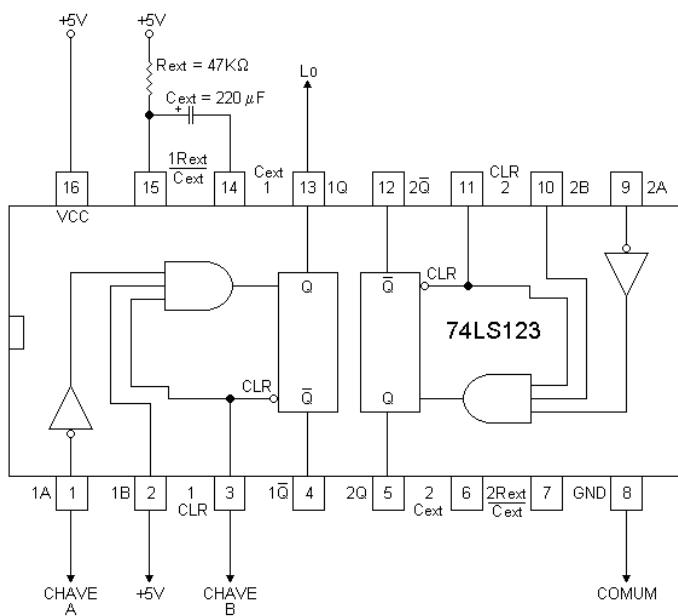


**Figura 11 - Monoestável com o 74121.**

- c)** Estando a chave A em  $\emptyset$ , levá-la para 1 e observar que a saída não mudou. Em seguida, levar a chave "A" para  $\emptyset$  e observar que a saída ficará em 1 por um tempo aproximado de \_\_\_\_\_ (verificar o tempo pela fórmula:  $T = 0,7 R_{ext} C_{ext}$ ).
- d)** Repetir o item "C" e, quando o tempo em que a saída deve estar em 1 estiver se esgotando, repetir a operação e verificar que isto não afeta a duração do tempo de saída alta. Isto se deve ao fato do 74121 não ser regatilhável.
- e)** Desligar as entradas A1 e A2 da chave "A" e ligá-las ao comum. Após, desligar a entrada "B" de +5V e ligá-la a chave "B". Verificar que a chave "B" sendo levada para 1 ocasiona um pulso de saída de mesma duração que nos itens anteriores.

### 13.2.7 EXPERIÊNCIA 2

- a)** Material Utilizado: 1 x CI 74LS123, 1 Resistor de  $47\text{ K}\Omega$  e 1 Capacitor de  $220\text{ }\mu\text{F}$
- b)** Montar o circuito da figura 12.

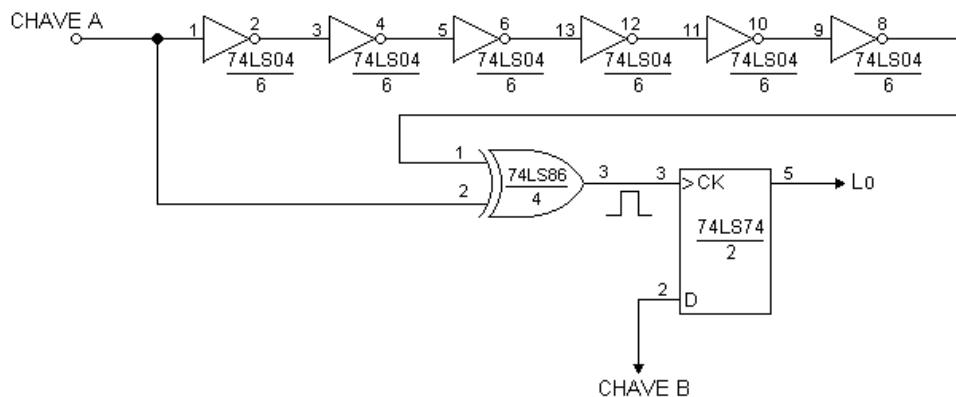


**Figura 12 - Monoestável com o 74LS123.**

- c)** Colocar a chave “A” em 1 e, estando a chave “B” em  $\emptyset$ , levá-la para 1 e observar que a saída não mudou. Em seguida, levar a chave “A” para  $\emptyset$  e observar que a saída ficará em “1” por um tempo aproximado de \_\_\_\_\_ (verificar o tempo pela fórmula ou pelo gráfico).
- d)** Repetir o item “C” e, quando o tempo em que a saída deve estar alta estiver se extinguindo, repetir a operação e verificar que o tempo de duração do pulso de saída foi afetado, isto é, aumentado. Isto se deve ao fato do 74LS123 ser regatilhável.
- e)** Repetir o item “C” e, durante o tempo em que a saída está alta, levar a chave “B” para “ $\emptyset$ ” e observar a influência do clear.

### 13.2.8 EXPERIÊNCIA 3

- a)** Material Utilizado: 1 x CI 74LS04, 1 x CI 74LS86 e 1 x CI 74LS74
- b)** Montar o circuito da figura 13, não se esquecendo de ligar os pinos 14 a +5V e os pinos 7 ao comum.



**Figura 13 - Monoestável com Gates.**

- c)** Toda vez que a chave A mudar de posição ocorrerá um pulso na saída do 74LS86 (pulso de duração igual ao número de inversores vezes o *time delay* de cada inversor). Como este pulso é de muita curta duração, foi usado o flip-flop D 74LS74 para detetá-lo.
- d)** Colocar a chave “B” em 1, mudar a chave “A” de posição e verificar que a saída do 74LS74 será 1. Em seguida, colocar a chave “B” em  $\emptyset$ , mudar novamente a chave “A” de posição e verificar que a saída do 74LS74 será  $\emptyset$ .
- e)** Comprovando o item “D”, estará provado que o circuito da figura 13 gera um pulso toda vez que a chave “A” muda de posição. Isto se explica pelo fato do 74LS74 apresentar em sua saída o dado que estava presente na linha “D” (chave B), toda vez que sofre um pulso de *clock*.
- f)** Como observado, tanto faz a chave “A” ir de  $\emptyset$  para 1 ou de 1 para  $\emptyset$  que ocorre um pulso na saída do 74LS86. Na prática, a maioria dos monoestáveis respondem apenas a um tipo de transição de entrada (ou positiva ou negativa).

**Exercício:**

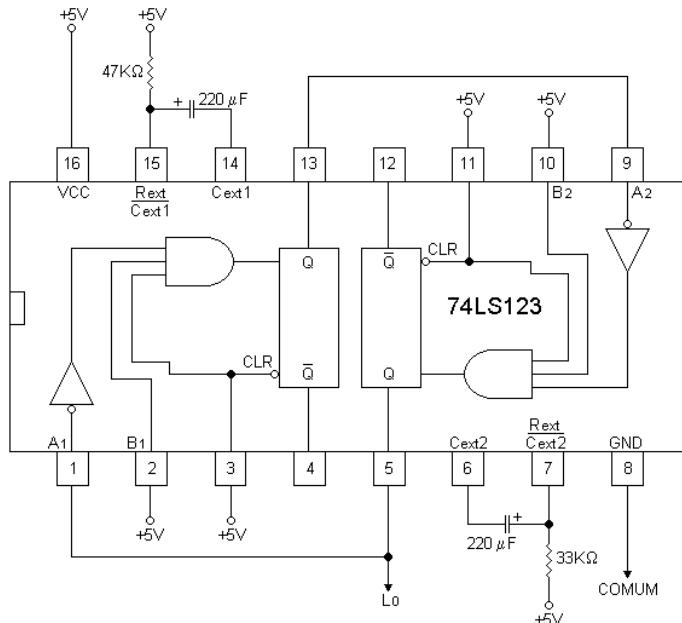
Usando CI's 74LS04 (seis inversores) e 74LS08 (quatro gates "E" de duas entradas), projetar um monoestável que responda somente a transições positivas da entrada (de  $\emptyset$  para 1) e que tenha uma duração do pulso de saída a menor possível.

**Solução:**

### 13.2.9 EXPERIÊNCIA 4

**a)** Material Utilizado: 1 x CI 74LS123, 1 Resistor de 33 K $\Omega$ , 1 Resistor de 47 K $\Omega$  e 2 Capacitores de 220  $\mu$ F

**b)** Montar o circuito da figura 14.

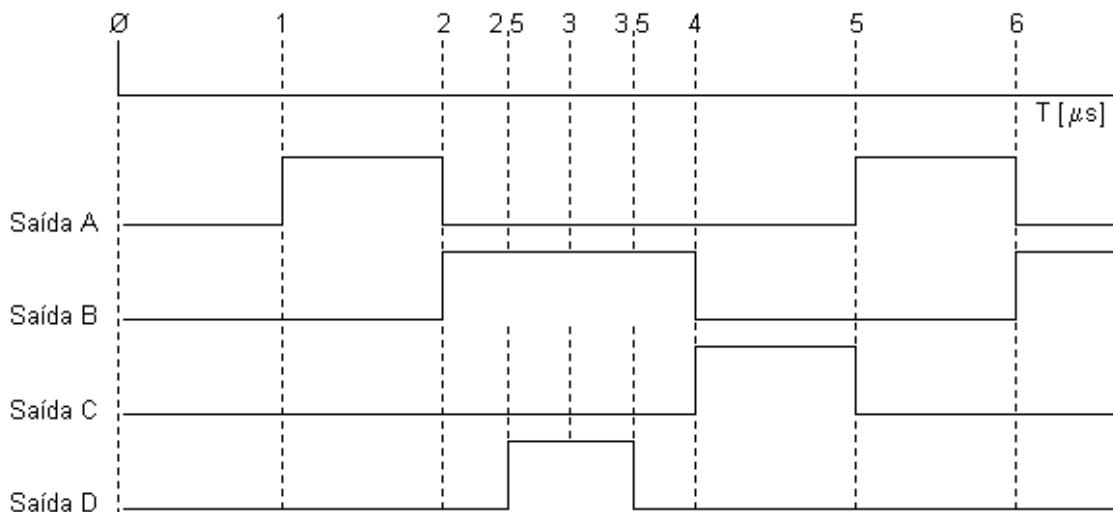


**Figura 14 - O 74LS123 como Astável.**

- c)** Observar que a saída Q2 ( $L\bar{Q}$ ) fica no estado baixo por \_\_\_\_\_ e no estado alto por \_\_\_\_\_, oscilando continuamente. Desta maneira, notar que o circuito montado é um oscilador ou astável construído com monoestável.
- d)** O tempo em que a saída Q2 fica alta ou baixa pode ser modificado, variando-se os resistores e capacitores externos. No caso em que  $R_{ext1} = R_{ext2}$  e  $C_{ext1} = C_{ext2}$ , tem-se que a saída Q2 será uma onda quadrada (tempo ligado = tempo desligado).
- e)** Pode-se construir osciladores com monoestáveis com várias saídas, cada saída tendo uma forma de onda diferente. Para isto, basta ir acoplando os monoestáveis (cada qual com um valor próprio de resistência e capacidade), sendo que a saída do último monoestável alimenta a entrada do primeiro. Estes astáveis também podem ser projetados com flip-flops, usando-se técnicas de projeto de contadores.
- f)** Quando usar monoestáveis não regatilháveis para construir circuitos astáveis, tomar precaução com o ciclo de trabalho.

**Exercício:**

Projetar um circuito usando CI's 74121 para gerar uma série de pulsos, conforme a carta de tempo dada na figura 15. Incluir os modos de operação "único-ciclo" e "contínuo". Incluir também uma chave de partida.

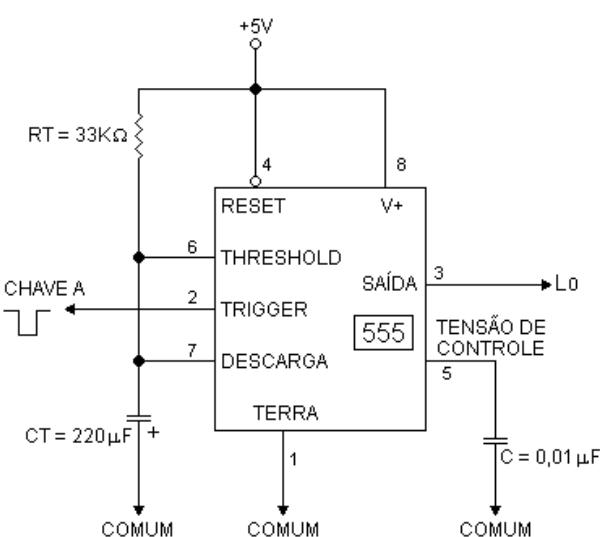


**Figura 15- Carta de tempo para Astável com CI's 74121.**

**Solução:**

### 13.2.10 EXPERIÊNCIA 5

- a)** Material Utilizado: 1 x CI 555, 1 Resistor de 33 K $\Omega$ , 1 Capacitor de 220  $\mu\text{F}$  e 1 Capacitor de 0,01  $\mu\text{F}$
- b)** Montar o circuito da figura 16.



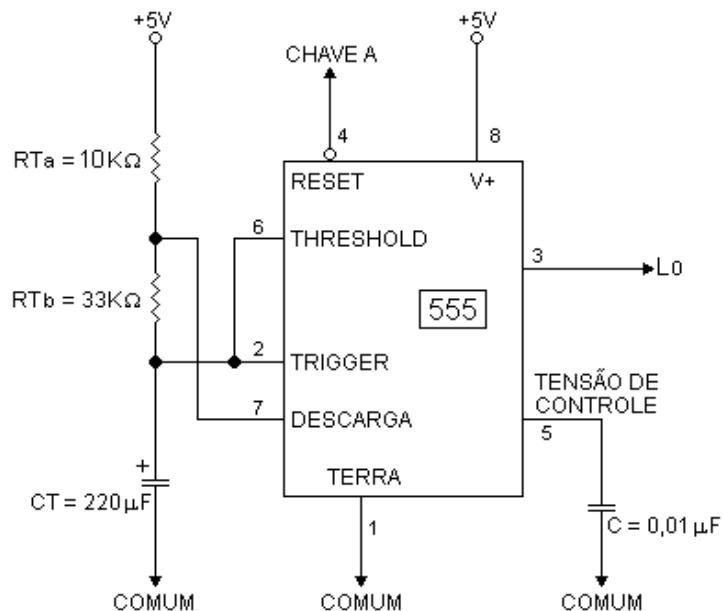
**Figura 16- O 555 como Monoestável.**

- c)** Colocar a chave “A” em 1 trazê-la a Ø e, em seguida, retornar a chave “A” em 1, verificando que na saída ocorre um pulso de duração \_\_\_\_\_ (verificar pela fórmula  $T = 1,1 R_T C_T$ ).

Cuidar para que a chave “A” permaneça em Ø por um tempo menor que T.

### 13.2.11 EXPERIÊNCIA 6

- a)** Material Utilizado: 1 x CI 555, 1 Resistor de 33 K $\Omega$ , 1 Resistor de 10 K $\Omega$ , 1 Capacitor de 220  $\mu\text{F}$  e 1 Capacitor de 0,01  $\mu\text{F}$ .
- b)** Montar o circuito da figura 17.

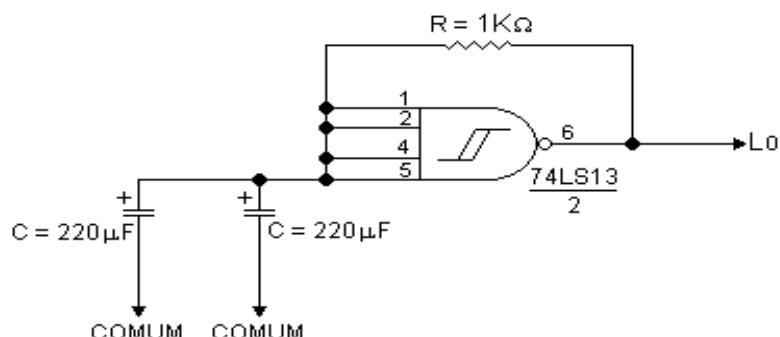


**Figura 17 - O 555 como astável.**

- c) Ligar o sistema, levar a chave “B” para 1 e observar a saída, que deverá estar oscilando. Assim foi construído um oscilador com o 555, o qual pode ter seu ciclo de trabalho alterado por mudanças em  $R_{Ta}$ ,  $R_{Tb}$  e  $C_T$ .
- d) Levar a chave “B” para Ø e observar a ação do reset sobre a saída.

#### 13.2.12 EXPERIÊNCIA 7

- a) Material Utilizado: 1 x CI 74LS13, 1 Resistor de 1 KΩ e 2 Capacitores de 220 μF.
- b) Montar o circuito da figura 18, não esquecendo de ligar os pinos 14 e 7 do 74LS13 a +5V e comum, respectivamente.



**Figura 18 - 74LS13 como Astável.**

- c) Antes de iniciar a experiência será lembrado o que vem a ser o 74LS13.

O 74LS13 é um CI com dois blocos “NÃO E” Schmitt-Trigger com quatro entradas. Um Schmitt-Trigger é um circuito que possui dois valores distintos para a tensão de entrada:

"Tensão **threshold** indo para positivo ( $V_{T+}$ )" e "Tensão **threshold** indo para negativo ( $V_{T-}$ )", sendo  $V_{T+}$  maior que  $V_{T-}$ .

O circuito interpreta como entrada 1 toda tensão acima de  $V_{T-}$ , desde que a tensão ( $V_{T+}$ ) já tenha sido alcançada, e interpretada como entrada Ø toda tensão abaixo de ( $V_{T+}$ ), desde que a entrada já tenha ficado submetida a uma tensão menor que ( $V_{T-}$ ). A diferença entre ( $V_{T+}$ ) e ( $V_{T-}$ ) é chamada *histeresi* e permite que o schmitt-trigger quadre formas de ondas lentas. Convém lembrar que as entradas A do 74121 eram entradas de um schmitt-trigger.

- d) Uma vez entendido o funcionamento do 74LS13, ligar o sistema e observar que a saída é oscilante. Desta maneira, construir um oscilador bem simples, cuja frequência é dada por

$$f = \frac{1}{1,1 RC}$$

e pode variar desde 0,1 Hz a 10 MHz.

### 13.3 EXERCÍCIOS

- a) Um monoestável pode ser usado para aumentar ou diminuir o tempo de duração de um pulso.

Certo  
 Errado

- b) O CI 74121 é um monoestável não regatilhável, e por esta razão não há limite para a frequência de pulsos aplicados as suas entradas.

Certo  
 Errado

- c) Tem-se um oscilador construído com 1 CI 74LS123 trabalhando com onda quadrada e frequência de 1 MHz. Tem-se também disponível vários CI's 74121, 74LS122, 74221, 555 e 556 (dois 555 em um só chip). No entanto, não se tem o CI 74LS123 disponível. No caso de queimado CI do oscilador, desejando-se a solução mais fácil, o 74LS123 poderá ser substituído por:

CI's 74121  
 1 CI 74121  
 2 CI's 74LS122  
 1 CI 74LS122  
 CI's 74221  
 1 CI 74221  
 CI's 555  
 1 CI 555  
 1 CI 556  
 Nenhuma das respostas

- d) Deseja-se construir um oscilador para onda quadrada de 3 MHz e dispõe-se de: vários resistores, vários capacitores, 1 CI 555 e 1 CI 74LS123. Para isto deve-se usar:

- O CI 555
- O CI 74LS123
- O CI 555 e o CI 74LS123
- Apenas com estes CI's não é possível construir este oscilador.
- Tanto faz, pode-se usar o 74LS123 ou o 555, desde que se escolha os resistores e os capacitores adequadamente.

- e) Deseja-se construir um oscilador em que a saída seja 1 por 10  $\mu$ s e 0 por 1  $\mu$ s, e dispõe-se de vários CI's 74121, vários resistores e vários capacitores. Pergunta-se: se for possível construir tal oscilador com CI's 74121, quantos CI's serão necessários?

Caso negativo justificar o porque.

**Obs.:** Considerar o ciclo de trabalho ligado.

Resposta:



## CAPÍTULO 14 – MEMÓRIAS SEMICONDUTORAS

### 14.1 RESUMO

As memórias semicondutoras na forma de CI possibilitaram a substituição das memórias de núcleo, tais como armazenagem em fitas e discos. Estes circuitos podem ser divididos em duas categorias principais: **memórias voláteis e memórias não voláteis**.

Memórias voláteis mantêm seus dados somente enquanto estiverem alimentadas, isto é, a informação é perdida na ausência de alimentação. Entretanto, na grande maioria das aplicações esta limitação não apresenta problemas. O termo genérico para definir este tipo de memória é **RAM (Random Access Memory)**, memória de acesso aleatório, onde os dados podem ser gravados (escritos), lidos e alterados sem problemas.

Em outras situações é imperativo o uso de uma memória não volátil, a qual manterá os seus dados armazenados com ou sem alimentação. Nestes casos, o termo genérico para definir este tipo de memória é **ROM (Ready Only Memory)**, memória somente de leitura. Estas são basicamente usadas nas situações onde as informações armazenadas não são sujeitas a trocas.

### 14.2 MEMÓRIAS RAM's

O mercado de memórias tem proliferado e mais de 3000 circuitos de memória são disponíveis. Consequentemente, em um projeto específico, a escolha de qual componente deve ser usado deve ser feita considerando as características da memória para tal aplicação. Dados relacionados com esta escolha serão: o tamanho da memória, seu consumo de potência, custo, sua velocidade e características de volatilidade.

Porque as *RAM's* perdem seus dados na falta de alimentação, os sistemas necessitarão de baterias de “backup” ou, então, deverão transferir os dados importantes para um meio não volátil, antes de desligar a alimentação.

As memórias semicondutoras são construídas, principalmente, de gates bipolares ou gates MOS. Ambos os tipos de memórias guardam suas informações em flip-flops ou em capacitores dentro do CI.

As memórias RAM's são divididas em duas categorias: **Estáticas e Dinâmicas**.

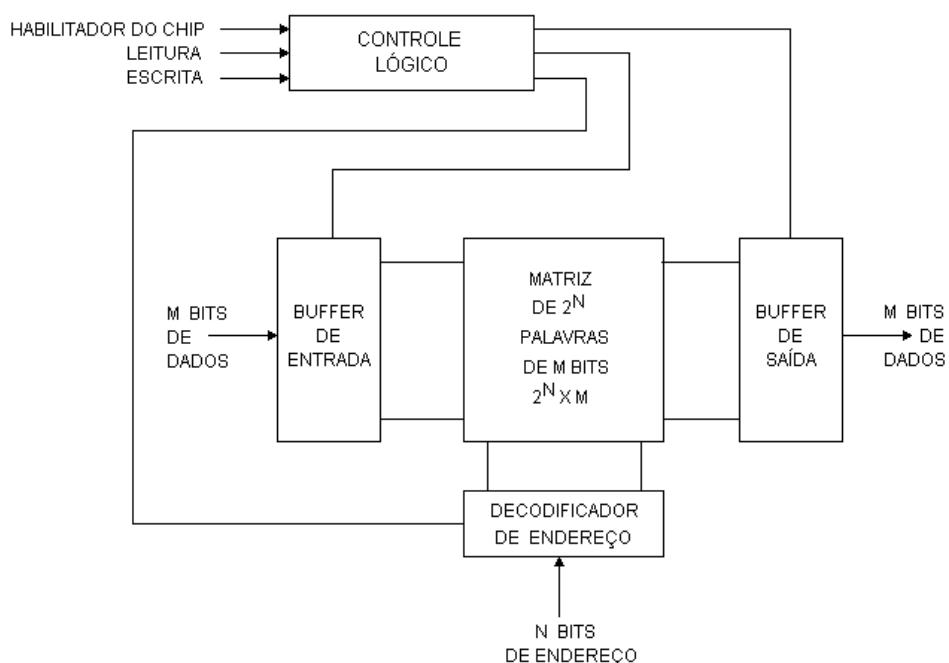
Memórias **Dinâmicas** são dotadas de capacitores internos, para armazenagem dos dados, e necessita de um circuito adicional, denominado circuito de refreshamento (*refreshing*), para recarga periódica dos dados armazenados.

Isto porque existe a tendência natural de descarga dos capacitores com o passar do tempo. Estas memórias são de velocidade moderada, de baixo consumo, de maior

capacidade de armazenagem, requerem maior tempo de projeto e têm menor custo por bit devido a sua maior densidade.

Memórias **Estáticas** guardam suas informações em flip-flops e são de controle mais simplificado, não necessitando de nenhum refrescamento periódico. Neste capítulo serão abordadas as memórias RAM's estáticas.

### 14.3 ORGANIZAÇÃO DE UMA MEMÓRIA RAM ESTÁTICA



**Figura 1 - Organização de uma memória RAM ESTÁTICA.**

Uma memória RAM normalmente tem pinos com as seguintes funções de entradas ou de saídas:

1. M Bits de saída de dados;
2. M bits de entrada de dados;
3. N bits de endereços (para acessar  $2^N$  palavras);
4. Entrada(s) de habilitação para leitura/escrita;
5. Entrada de habilitação ou seleção do *CHIP*.

Devido a limitações no tamanho dos *CHIPS*, memórias de maior capacidade possuem as funções de entrada e saída de dados em um mesmo conjunto de pinos. Quem diz se o dado estará entrando na memória ou saindo da mesma será o comando de leitura/escrita na memória.

A maioria das memórias tem saídas em coletor aberto ou em *tri-state*, para permitir ligação em paralelo das mesmas, a fim de se obter maior capacidade de manuseio de dados. Assim, quando o sinal de habilitação do *CHIP* não estiver atuando, este componente ficará no estado de alta impedância, onde não se pode escrever na memória e nem ler seus conteúdos. Isto significa que a memória estará desconectada dos demais componentes externos.

A operação de gravação ou escrita na memória é feita colocando-se os dados nas linhas de entrada, habilitando o *CHIP*, colocando-se os sinais de endereço da posição desejada e habilitando a escrita na memória.

Deste modo, os dados colocados nas linhas de entrada serão escritos na posição selecionada pelas linhas de endereçamento. Deve-se tomar o cuidado com a seqüência correta de colocação dos sinais, pois uma alteração nos dados ou nos endereços, enquanto o sinal de escrita estiver habilitado, irá alterar o conteúdo do endereço atual selecionado, apagando o conteúdo anterior.

A operação de leitura da memória é feita colocando-se os sinais de endereço da posição que se deseja ler, habilitando a leitura da memória e, finalmente, habilitando o *CHIP*. Deste modo, a posição de memória selecionada libera seu conteúdo para os pinos correspondentes de saída de dados.

A leitura é não destrutiva. Novamente, deve-se tomar o cuidado com o(s) sinal(is) de leitura/escrita para que, durante um processo de leitura da memória, não se faça uma escrita errada na posição selecionada.

Normalmente, o(s) sinal(is) de leitura/escrita deve(m) permanecer selecionando leitura; só selecionarão escritas quando tiver certeza de que os outros sinais estão corretamente colocados.

#### 14.4 TEMPOS IMPORTANTES DE MEMÓRIA

a) Tempo de Acesso

É o tempo requerido para a memória apresentar dados válidos após firmados os sinais de endereço e seleção.

b) Tempo de Ciclo ou Tempo de Escrita

É o tempo requerido para que o endereço e dados sejam mantidos constantes a fim de se gravar na memória.

## 14.5 MEMÓRIAS RAM's BIPOLARES E MOS

A maioria das memórias são construídas com flip-flops bipolares (TTL) ou MOS. As memórias bipolares são todas estáticas. Tais memórias são muito rápidas e são usadas em sistemas de alta velocidade, exigindo somente uma fonte de +5 volts.

Existem na linha TTL 74 várias memórias RAM, entre as quais pode-se citar a 7481 (16 palavras de 1 bit), a 74LS170 (4 palavras de 4 bits), a 7489 (16 palavras de 4 bits), a 74201 (256 palavras de 1 bit), etc.

Estas memórias têm pequena capacidade de armazenagem e, portanto, o custo de produção por bit é alto. Por este motivo, estão deixando de ser produzidas por alguns fabricantes.

O aperfeiçoamento da tecnologia MOS possibilitou a construção de memórias estáticas HMOS e HCMOS que possuem características de alta performance relativos a baixo consumo e altas velocidades.

A seguir será abordada a memória 6116 (2048 x 8 bits).

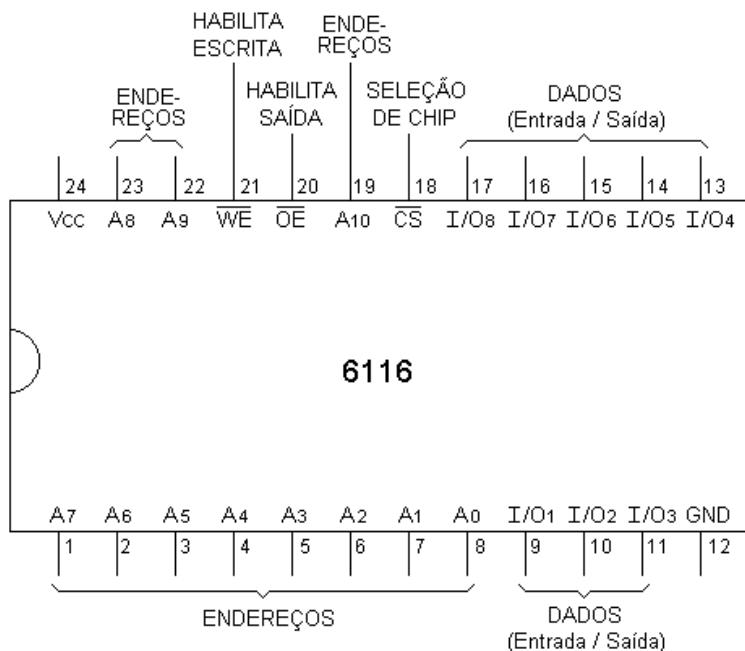
A memória estática 6116 é uma memória na tecnologia CMOS, contendo 2048 palavras de 8 bits e possuindo baixo consumo, o que possibilita operação com baterias de *backup* e não requer nenhum circuito de refrescamento. A retenção de dados é garantida para alimentação de até 2 volts no mínimo.

Para tal capacidade, esta memória tem 8 pinos ( $M = 8$ ) para bits de dados, operando como entrada/saída ( $I/O_1$  até  $I/O_8$ ), 11 pinos ( $N = 11$ ) para endereçamento ( $A_0$  até  $A_{10}$ ) e 3 pinos para operações de controle lógico.

Possui as seguintes características:

1. Tempo de acesso de 100 ns/120 ns/150 ns;
2. Corrente de operação no modo normal 70 mA;
3. Corrente de operação no modo standby 50  $\mu$ A;
4. Retenção de dados para 2 volts;
5. Entradas e saídas diretamente compatíveis com TTL;
6. Única fonte de  $5V \pm 10\%$ ;
7. Saídas *Tri-state*;
8. Barramento de entrada e saída comum;
9. Pinagem compatível com memórias EPROM de 16K, por exemplo 2716.

A figura 2 mostra o lay-out dos pinos da memória 6116.



**Figura 2 - Lay-out da memória 6116.**

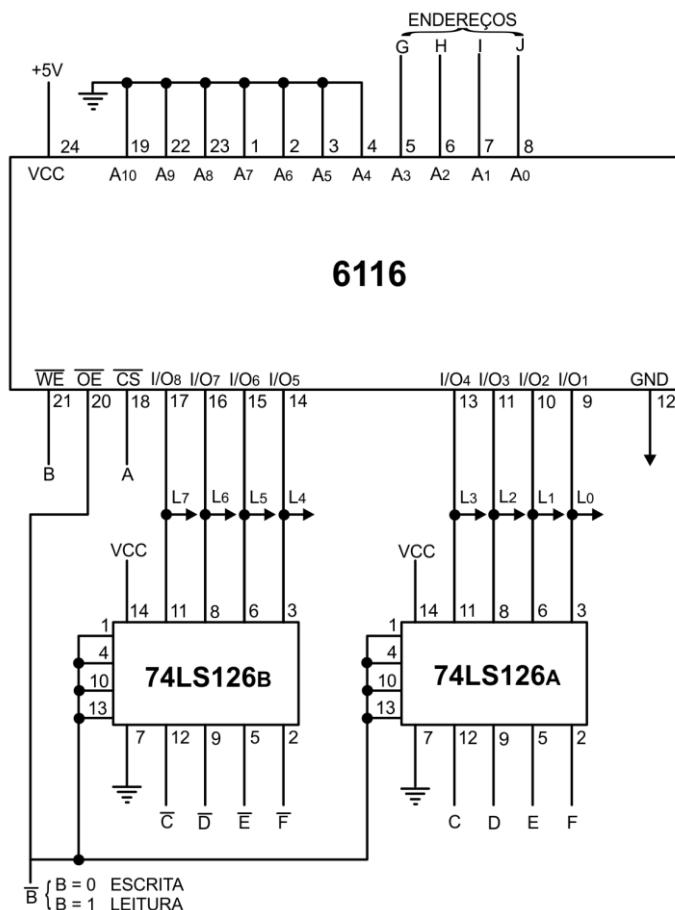
A tabela 1 mostra o funcionamento da memória 6116.

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	I/O1-8	OPERAÇÃO
H	X	X	Alta Impedância	Desabilitado
L	L	H	Saída	Leitura
L	H	L	Entrada	Escrita
L	L	L	Entrada	Escrita

**Tabela 1 - Funcionamento para 6116.**

#### 14.5.1 EXPERIÊNCIA 1

- a) Material Utilizado: 1 x CI 6116 e 2 x CI's 74LS126.
- b) Montar o circuito da figura 3, observando que, com poucas modificações, o mesmo poderá ser adaptado para a experiência 2 (figura 5) não sendo necessário desmontá-lo.



**Figura 3 – Interface com memória 6116.**

- c) No circuito da figura 3, a finalidade dos CI's 74LS126 (buffer tri-state) é não deixar as chaves conectadas diretamente às entradas/saídas da memória 6116, pois quando fosse feita operação de leitura da memória, haveria conflito entre as saídas da memória e os valores nas chaves.

A memória 6116 é de 2048 palavras x 8 bits. Nesta experiência, os endereços acessados são:

	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	HEXADECIMAL
DE	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	= Ø Ø Ø H
ATÉ	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	1	= Ø Ø ØFH

Portanto, a experiência irá gravar nas primeiras 16 posições de endereço da memória, conforme apresentado na tabela 2.

Observe a sequência correta de mudanças nas chaves, durante a escrita na memória.

ENTRADAS					SAÍDAS								OPERAÇÃO					
SELEÇÃO		ENDEREÇOS			DADOS													
CS	WE	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	I/O <sub>4</sub>	I/O <sub>3</sub>	I/O <sub>2</sub>	I/O <sub>1</sub>	I/O <sub>8</sub>	I/O <sub>7</sub>	I/O <sub>6</sub>	I/O <sub>5</sub>	I/O <sub>4</sub>	I/O <sub>3</sub>	I/O <sub>2</sub>	I/O <sub>1</sub>	
A	B	G	H	I	J	C	D	E	F	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>	
1	x	x	x	x	x	x	x	x	x	--	--	--	--	--	--	--	--	①
0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	②
0	0	0	0	0	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE ESCRITA
0	0	0	0	1	0	0	0	1	0	x	x	x	x	x	x	x	x	③
0	0	0	0	1	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	0	1	0	0	0	1	0	0	x	x	x	x	x	x	x	x	④
0	0	0	1	0	1	0	1	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	0	1	1	0	0	1	1	0	x	x	x	x	x	x	x	x	⑤
0	0	0	1	1	1	0	1	1	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	1	0	0	0	1	0	0	0	x	x	x	x	x	x	x	x	⑥
0	0	1	0	0	1	1	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	1	0	1	0	0	1	0	0	x	x	x	x	x	x	x	x	⑦
0	0	1	0	1	1	0	1	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	1	1	0	0	1	1	0	0	x	x	x	x	x	x	x	x	⑧
0	0	1	1	0	1	1	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	0	1	1	1	0	0	1	1	0	x	x	x	x	x	x	x	x	⑨
0	1	0	0	0	0	1	1	1	1	x	x	x	x	x	x	x	x	⑩
0	1	0	0	0	1	0	0	1	0	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	0	0	1	0	0	0	1	0	x	x	x	x	x	x	x	x	⑪
0	1	0	0	1	0	1	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	0	0	1	1	0	0	1	0	x	x	x	x	x	x	x	x	⑫
0	1	0	1	0	0	0	1	0	0	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	0	1	0	1	0	0	1	0	x	x	x	x	x	x	x	x	⑬
0	1	0	1	1	0	0	0	1	0	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	0	1	1	0	1	0	0	1	x	x	x	x	x	x	x	x	⑭
0	1	0	1	1	1	0	0	1	0	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	0	0	0	0	1	0	0	x	x	x	x	x	x	x	x	⑮
0	1	1	0	0	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	0	1	0	0	0	1	0	x	x	x	x	x	x	x	x	⑯
0	1	1	0	1	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	1	0	0	0	1	0	0	x	x	x	x	x	x	x	x	⑰
0	1	1	1	0	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	1	1	0	0	0	1	0	x	x	x	x	x	x	x	x	⑱
0	1	1	1	1	1	0	0	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	1	1	1	1	0	0	1	x	x	x	x	x	x	x	x	⑲
0	1	1	1	1	1	1	1	0	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	1	1	1	1	1	1	0	x	x	x	x	x	x	x	x	⑳
0	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x	OPERAÇÃO DE LER
0	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x	⑳

- ① - Alta impedância

**Tabela 2 - memória 6116.**

d) Na tabela 2, observa-se que em:

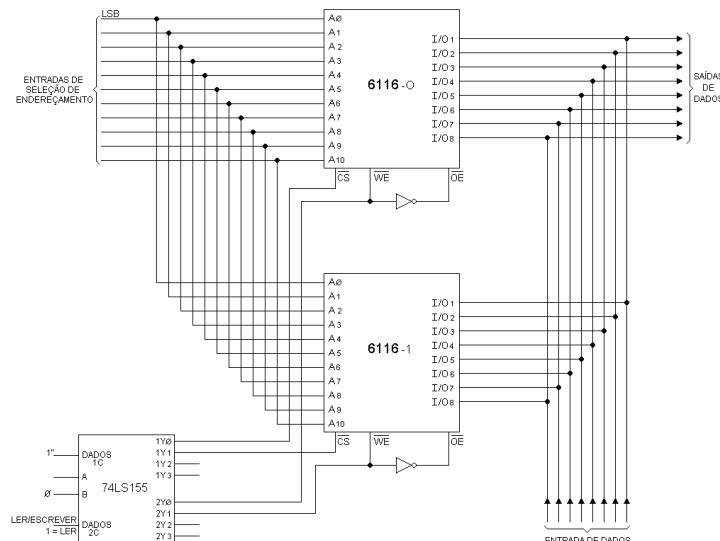
- ① tem-se A = 1, o que coloca a memória no estado de alta impedância e, portanto, desconectada dos demais pinos de entrada/saída. A chave “B” estará controlando os *buffers tri-state*. Com B = Ø, os buffers estarão habitados passando os valores das chaves para os led's, mas não para a memória. Com B = 1, os buffers também estarão em *tri-state*.

- ② tem-se  $A = \emptyset$ , e  $B = 1$ . Isto significa que a memória está selecionada e “B” fará um ciclo de escrita na posição de memória especificada pelas chaves de endereço (GHIJ). Observe que se a memória permanece constantemente habilitada por  $A = \emptyset$ , então “B” deverá permanecer em 1 (leitura) durante as alterações de endereços e dados nas chaves, para não efetuar uma operação de escrita com dados errados em um endereço em um endereço indesejado.
- Para a gravação correta dos dados, nos correspondentes endereços da memória, é necessária que se mantenha a memória desabilitada ( $\overline{WE} = 1$ ), coloque o endereço, coloque o dado, habilite para escrita ( $\overline{WE} = \emptyset$ ). Assim o dado será armazenado no endereço selecionado. Após isto, deve-se novamente desabilitar a memória ( $\overline{WE} = 1$ ) para mudança de endereços e dados e nova sequência de escrita. Este processo de escrita é realizado automaticamente pelos sinais de controle de um microprocessador.
- ③ tem-se  $A = \emptyset$  e  $B = 1$ , que seleciona operação de leitura nos endereços especificados (GHIJ). Lembre-se que a leitura é não destrutiva, o que significa que se o endereço for repetido, deve-se obter o mesmo conteúdo da posição selecionada.

#### 14.6 CONSTRUÇÃO DE MEMÓRIAS MAIORES

As experiências 2 e 3 a seguir abordam o caso de construção de memórias maiores. Na experiência 2, tem-se o caso em que é desejado aumentar o número de palavras das memórias, porém mantendo o mesmo número de bits por palavras na memória maior. Já na experiência 3, tem-se o caso em que é mantido o mesmo número de palavras na memória maior e aumentado o número de bits por palavra.

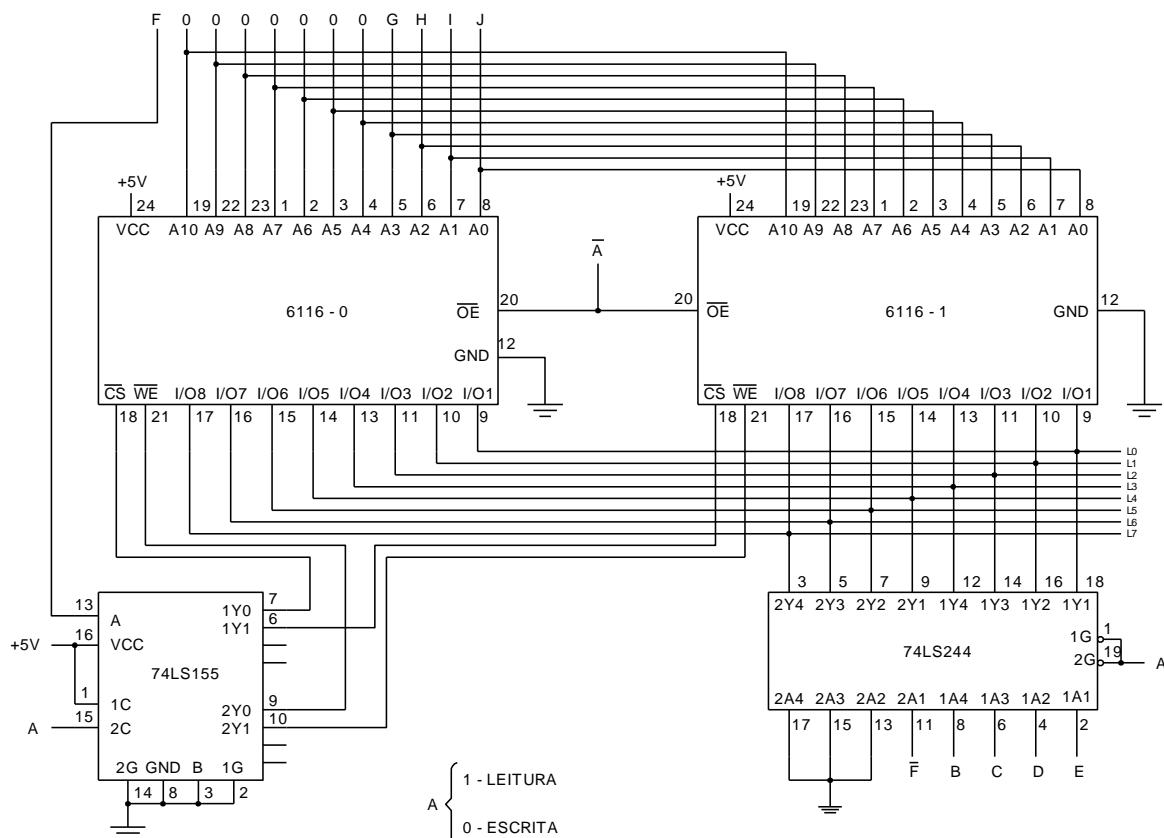
A figura 4 mostra o diagrama em blocos resumido de uma memória de 4096 palavras de 8 bits, usando duas memórias 6116 de 2048 palavras de 8 bits. A finalidade do CI 74LS155 é operar como decodificador de endereços, ou seja, dependendo do endereço especificado pelo bit mais significativo (MSB), este CI selecionará uma das memórias 6116.



**Figura 4 - Memória de 4096 palavras x 8 bits construída com 2 memórias de 2048 palavras x 8 bits**

### 14.6.1 EXPERIÊNCIA 2

- a) Material Utilizado: 2 x CI 6116, 2 x CI's 74LS126 e 1 x 74LS155.
- b) Montar o circuito da figura 5.



**Figura 5 – Acoplamento de memórias 6116 (2048 x 8 bits) para formar uma memória de 4096 palavras de 8 bits.**

- c) No circuito da figura 5, o CI 74LS155 estará habilitando a escrita/leitura nas 16 primeiras posições da memória 6116-Ø, quando  $F = \emptyset$ , ou habilitando a escrita / leitura nas 16 primeiras posições da memória 6116-1, quando  $F = 1$ . A operação de leitura/escrita será selecionada pela chave A ( $\emptyset$  = escrita, 1 = leitura). Se todas as linhas de endereçamento fossem usadas, haveria 2048 palavras de 8 bits endereçadas para cada memória, num total de 4096 palavras de 8 bits.

d) Completar a tabela 3:

ENDEREÇOS						ENTRADAS					SAÍDAS							COMEN-	TÁRIOS
A	F	G	H	I	J	B	C	D	E	L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>Ø</sub>		
	Ø	1	1	1	1	1	1	1	1									①	
	Ø	Ø	1	1	1	Ø	1	1	1										
	Ø	Ø	Ø	1	1	Ø	Ø	1	1										
	Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	1										
	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø										
	1	1	1	1	1	Ø	Ø	Ø	Ø									②	
	1	Ø	1	1	1	1	Ø	Ø	Ø										
	1	Ø	Ø	1	1	1	1	Ø	Ø										
	1	Ø	Ø	Ø	1	1	1	1	Ø										
	1	Ø	Ø	Ø	Ø	Ø	1	1	1										
1	1	Ø	Ø	Ø	Ø	Ø	X	X	X									③	
1	1	Ø	Ø	Ø	Ø	1	X	X	X										
1	1	Ø	Ø	Ø	1	1	X	X	X										
1	1	Ø	Ø	1	1	1	X	X	X										
1	1	1	1	1	1	X	X	X	X										
1	Ø	1	1	1	1	X	X	X	X									④	
1	Ø	Ø	1	1	1	X	X	X	X										
1	Ø	Ø	Ø	1	1	X	X	X	X										
1	Ø	Ø	Ø	Ø	1	X	X	X	X										
1	Ø	Ø	Ø	Ø	Ø	X	X	X	X										

**Tabela 3 - Tabela para o circuito da figura 5.**

#### COMENTÁRIOS:

- 1) Escrita nas 16 palavras iniciais de 6116-Ø
- 2) Escrita nas 16 palavras iniciais de 6116-1
- 3) Leitura das 16 palavras iniciais de 6116-1
- 4) Leitura das 16 palavras iniciais de 6116-Ø

e) Na tabela 3, observar que em:

- ① foram feitas operações de escrita em cinco endereços localizados entre as 16 primeiras palavras, o que corresponde ao uso da 6116- Ø.
- ② foram feitas operações de escrita em cinco endereços localizados entre as 16 primeiras palavras da 6116-1.
- Nos casos ① e ②, na parte mais significativa dos bytes da memória foi gravados o valor da entrada “F”, conforme mostrado na conexão do 74LS126 “B” da figura 5. (ØØØF)
- ③ foram lidos os mesmos endereços usados em ②. As saídas devem ser as mesmas entradas escritas em ②, com partes mais significativas iguais a ØØØ1.
- ④ foram lidos os mesmos endereços usados em ①. As saídas devem ser as mesmas entradas escritas em ①, com partes mais significativas iguais a ØØØØ.

**Observação:** os endereços usados nos itens ① e ② foram escolhidos de forma a evitar corridas. Deve-se obedecer as sequências de mudanças das chaves, conforme experiência anterior.

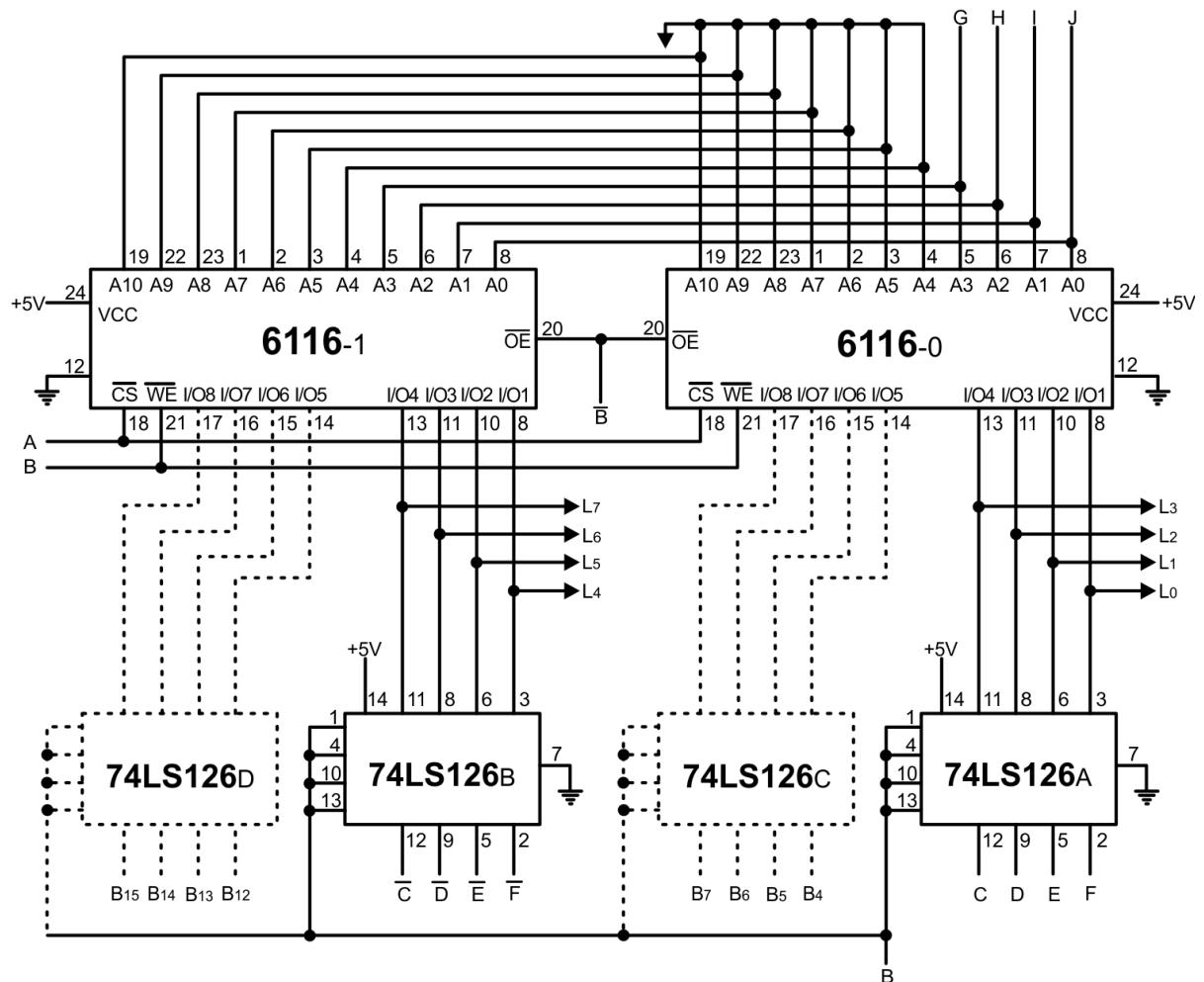
f) O demultiplexador da figura 5 permite a construção de uma memória de 8192 palavras de 8 bits, sendo necessárias mais duas memórias 6116. Isto seria possível pelo fato do demultiplexador 74LS155 ainda ter disponíveis as saídas  $1Y_2$ ,  $1Y_3$ ,  $2Y_2$  e  $2Y_3$ . Neste caso, a entrada “B” do 74LS155 participaria como uma entrada de seleção, ou de endereço. Assim, o mapa de endereçamento deste circuito seria dado pelas linhas  $A_0$  até  $A_{10}$  das memórias 6116 conectadas em paralelo e pelas linhas A e B do 74LS155, que seriam responsáveis pela habilitação de uma das quatro memórias existentes, como esquematizado na tabela 4 a seguir.

LINHAS DE ENDEREÇAMENTO												MEMÓRIA		
	B	A	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	SELECIONADA
De	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	6116-Ø
Até	Ø	Ø	1	1	1	1	1	1	1	1	1	1	1	End.ØØØØH até Ø7FFH
De	Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	6116-1
Até	Ø	1	1	1	1	1	1	1	1	1	1	1	1	End.Ø8ØØH até ØFFFH
De	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	6116-2
Até	1	Ø	1	1	1	1	1	1	1	1	1	1	1	End.1ØØØH até 17FFH
De	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	6116-3
Até	1	1	1	1	1	1	1	1	1	1	1	1	1	End.18ØØH até 1FFFH

**Tabela 4 - Mapa de endereços de uma memória de 8192 palavras de 8 bits, construída com 4 x 6116 (2048 x 8 bits).**

### 14.6.2 EXPERIÊNCIA 3

- a) Material Utilizado: 2 x CI's 6116 e 2 x CI's 74LS126
- b) Montar o circuito da figura 6.



**Figura 6 – Construção de uma memória de 2-48 palavras de 16 bits usando duas 6116 em paralelo.**

- c) No circuito da figura 6, foi construída uma memória de 2048 palavras de 16 bits, usando duas 6116 ligadas em paralelo. Devido às limitações de chaves no módulo, serão manuseados somente os quatro bits menos significativos da cada memória. Para manuseio dos quatro bits mais significativos de cada memória seria necessário o uso de mais dois buffers 74LS126, como mostrado na figura 6.

d) Complete a tabela 5.

		ENTRADAS				SAÍDAS																					
SELEÇÃO	ENDEREÇOS	DADOS				I/O <sub>8</sub>	I/O <sub>7</sub>	I/O <sub>6</sub>	I/O <sub>5</sub>	I/O <sub>4</sub>	I/O <sub>3</sub>	I/O <sub>2</sub>	I/O <sub>1</sub>		I/O <sub>8</sub>	I/O <sub>7</sub>	I/O <sub>6</sub>	I/O <sub>5</sub>	I/O <sub>4</sub>	I/O <sub>3</sub>	I/O <sub>2</sub>	I/O <sub>1</sub>		L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
CS	WE	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	I/O <sub>4</sub>	I/O <sub>3</sub>	I/O <sub>2</sub>	I/O <sub>1</sub>						L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>						L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
A	B	G	H	I	J	C	D	E	F						L <sub>7</sub>	L <sub>6</sub>	L <sub>5</sub>	L <sub>4</sub>						L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
1	X	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x
0	0	0	0	0	1	0	0	1	1	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	0	0	0	1	1	0	1	1	0	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	0	1	1	1	1	1	1	0	0	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	0	1	1	1	1	1	0	0	1	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	1	1	1	1	1	x	x	x	x	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	1	0	1	1	1	x	x	x	x	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	1	0	0	1	1	x	x	x	x	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--
0	1	0	0	0	1	x	x	x	x	--	--	--	--		--	--	--	--	--	--	--	--	--	--	--	--	--

(--) Sinais não conectados

(x ) Não interessa

**Tabela 5 – Tabela para o circuito da figura 6.**

e) Na tabela 5, observar que em:

- ① com A = 1, as duas memórias em paralelo estão no estado de alta impedância.
- ② foi feita uma escrita em quatro endereços das memórias, sendo que cada palavra agora deveria ser composta de 16 bits.
- ③ foi feita a leitura nos quatro endereços de ②, a saída deverá ter os respectivos valores escritos.

## 14.7 MEMÓRIAS ROM

As memórias ROM são pré-gravadas de algum modo ou de forma semi-permanente. Não são gravadas durante a operação normal do dispositivo; o conteúdo de uma ROM não muda e é não volátil, isto é, se faltar energia de alimentação o conteúdo não é destruído. As ROM's são usadas para conversões de códigos, memorização de tabelas e para programas de finalidades especiais de controle em computadores. Usualmente, são as ROM's que rodam o programa num microprocessador.

Este programa é denominado de programa monitor, ou sistema operacional da máquina.

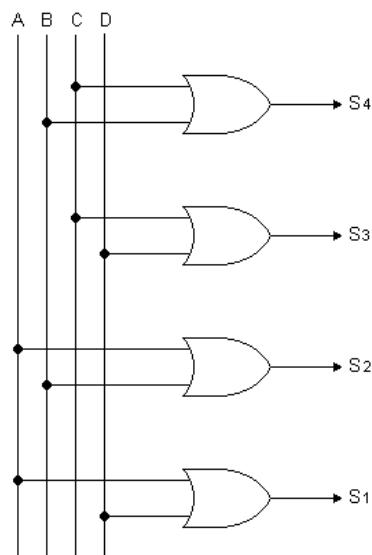
Pode-se classificar as ROM's em:

- 1) ROM propriamente dita;
  - 2) PROM;
  - 3) EPROM;
  - 4) EEPROM.
- a)** Nas ROM's propriamente ditas, as palavras já são gravadas no processo de fabricação. O usuário não pode mudar o conteúdo da ROM.
- b)** Nas PROM's (ROM's programáveis), o usuário consegue gravar na memória uma única vez. Geralmente, isto é feito através da queima de elos fusíveis na matriz, sendo a programação realizada por equipamentos especiais.
- c)** EPROM's são memórias PROM's apagáveis, em que os conteúdos da memória podem ser apagados e novamente gravados com aparelhagem especial. Geralmente, o apagamento é feito com luz ultravioleta.
- d)** EEPROM's são memórias EPROM's apagáveis eletricamente. Pode-se encarar a ROM como um codificador, no sentido de que para uma única entrada ativada de cada vez há um conjunto de saídas correspondente a esta entrada. A figura 12 é a tabela de um codificador de 4 entradas e 4 saídas.

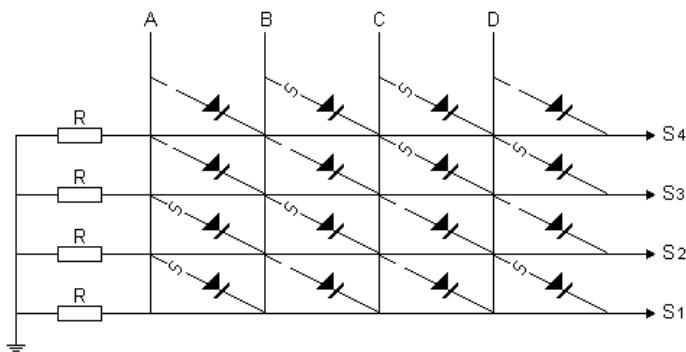
ENTRADAS				SAÍDAS			
A	B	C	D	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>
Ø	Ø	Ø	1	Ø	1	Ø	1
Ø	Ø	1	Ø	1	1	Ø	Ø
Ø	1	Ø	Ø	1	Ø	1	Ø
1	Ø	Ø	Ø	Ø	Ø	1	1

**Tabela 6 - Codificador (ROM) de 4 entradas e 4 saídas.**

A figura 7 mostra o circuito lógico da ROM da tabela 6.



**Figura 7 - Circuito correspondente a ROM da tabela 6.**



**Figura 8 - Memória PROM correspondente a tabela 6.**

No caso da ROM da figura 7, tem-se uma memória de 4 palavras e cada palavra de 4 bits. A figura 8 mostra a execução deste codificador na forma de PROM com matriz de diodos e elos fusíveis.

Se as memórias fossem feitas como a das figuras 7 ou 8, o número de pinos necessários para endereçar a memória teria de ser igual ao número de palavras endereçadas. Então, não seria possível fazer-se estas memórias na forma de circuitos integrados, os quais têm um número limitado de pinos.

A fim de se diminuir o número de entradas de endereçamento, usa-se um circuito chamado decodificador, em que para cada combinação das entradas do decodificador uma única entrada de endereçamento da memória é ativada.

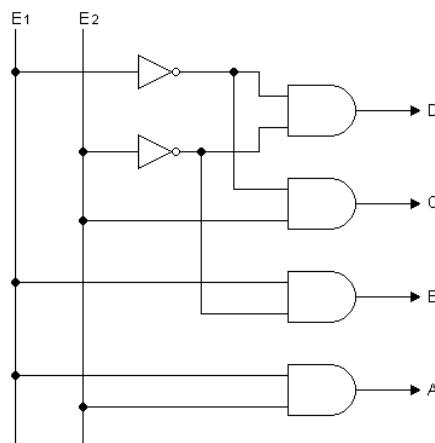
A tabela 7 é a de um decodificador adequado a memória da figura 7. Normalmente, as memórias já vêm com o decodificador incorporado.

ENTRADAS		SAÍDAS			
E1	E2	A	B	C	D
Ø	Ø	Ø	Ø	Ø	1
Ø	1	Ø	Ø	1	Ø
1	Ø	Ø	1	Ø	Ø
1	1	1	Ø	Ø	Ø

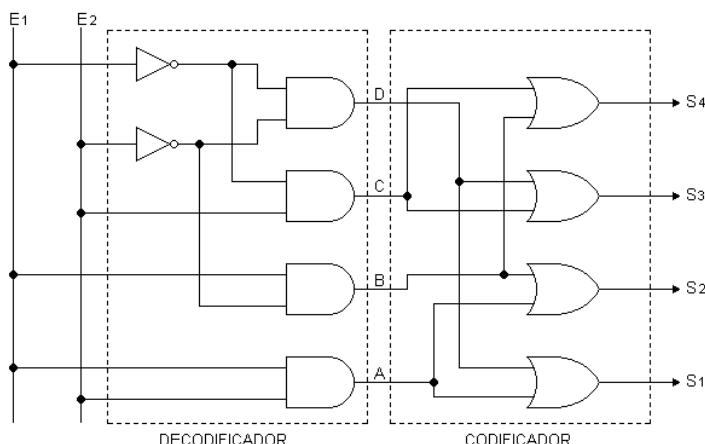
**Tabela 7 - Decodificador de 2 entradas e 4 saídas.**

A figura 9 mostra o circuito do decodificador correspondente a tabela 7.

Finalmente, a figura 10 mostra o circuito da memória ROM correspondente a tabela 6, com o decodificador incorporado.



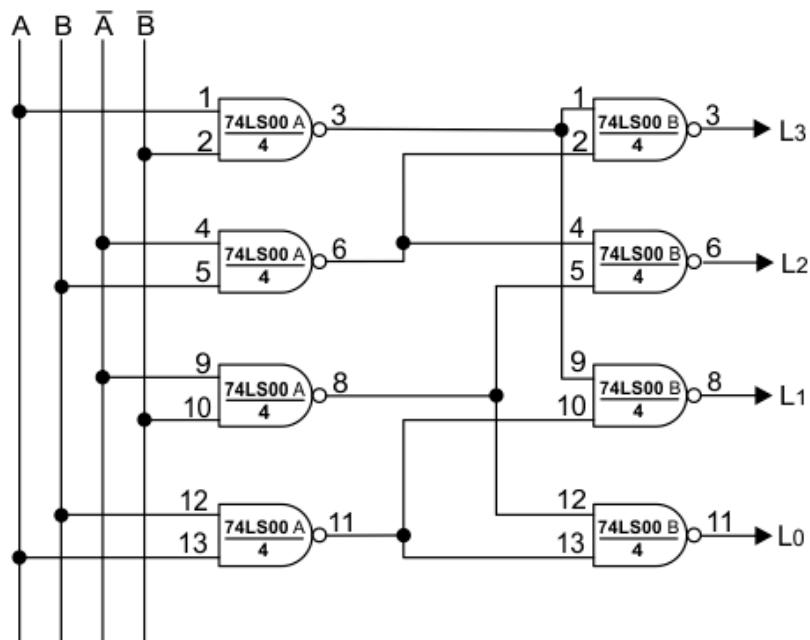
**Figura 9 - Circuito do decodificador correspondente a tabela 7.**



**Figura 10- Circuito da Memória ROM com decodificador incorporado.**

### 14.7.1 EXPERIÊNCIA 4

- a)** Material Utilizado: 2 x CI's 74LS00  
**b)** Montar o circuito da figura 11.



*Figura 11 – Memória ROM de 4 palavras com 4 bits cada.*

ENTRADAS		SAÍDAS			
A	B	L3	L2	L1	L0
Ø	Ø				
Ø	1				
1	Ø				
1	1				

*Tabela 8 – Tabela para o circuito da figura 11.*

- c)** A memória utilizada na figura 11, é uma pequena memória ROM construída com Gates “NÃO E”. Tal memória consta de 4 palavras de 4 bits, num total de 16 bits. Na série TTL existem memórias ROM, como exemplo tem-se o 74187 que é uma memória ROM de 256 palavras de 4 bits, tendo 8 linhas de endereços, para selecionar uma das 256 palavras, e 4 saídas para cada palavra. Enquanto o chip estiver habilitado a saída tem o valor dos 4 bits da palavra endereçada. A 74187 é do tipo bipolar e tem tempo de acesso típico de 40ns.



## CAPÍTULO 15 - INTERLIGAÇÕES LÓGICAS OU INTERFACES LÓGICAS

### 15.1 RESUMO

É possível um grande número de interligações (*interface*) entre os três tipos básicos de MOS (PMOS, NMOS e CMOS), e entre estes e as outras famílias lógicas bipolares.

Em todos os casos a compatibilidade entre vários parâmetros deve ser verificada antes de se efetuar as interligações. Estes parâmetros incluem:

**Níveis de tensão e polaridades, correntes de entradas e saídas, impedância, formas de ondas, tempos de crescimento e queda, proteção contra transitórios destrutivos e exigências de tempo do sistema.**

Algumas interligações entre famílias lógicas são completamente compatíveis, enquanto muitas outras podem ser executadas com relativa facilidade embora, frequentemente, sejam necessários componentes extras, tais como resistores. Outras vezes são necessários circuitos trocadores de níveis. As exigências de entrada e capacidade de saída de algumas famílias lógicas estão mostradas na tabela 1, sendo esta tabela de grande auxílio na realização das várias interligações entre CI's de diversas famílias.

DISPOSITIVOS		ENTRADAS LÓGICAS REQUERIDAS		SAÍDAS LÓGICAS		VOLTS			
		LÓGICA 0 Vmáx./I (Volts) (mA)	LÓGICA 1 Vmín./I (Volts) (μA)	LÓGICA 0 Vmáx./I (Volts) (mA)	LÓGICA 1 Vmín./I (Volts) (μA)	VCC	TERRA	VSS	VDD
TTL	Std..	0,8 / 1,6	2,0 / 40	0,4 / 16	2,4 / 400	5	Ø		
	H	0,8 / 2	2,0 / 50	0,4 / 20	2,4 / 500	5	Ø		
	S	0,8 / 2	2,0 / 50	0,5 / 20	2,4 / 1000	5	Ø		
	LS	0,7 / 0,36	2,0 / 20	0,5 / 8,0	2,4 / 400	5	Ø		
	FAST	0,8 / 0,6	2,0 / 100	0,4 / 2,0	2,5 / 1000	5	Ø		
CMOS <sup>2</sup>	4000	+5V	1,5V	3,5V	0,4 / 1,6	2,5 / 3 mA		Ø	5
		+10V	3,0V	7,0V	0,5 / 3,5	9,5 / 2,3mA		Ø	10
		+15V	4,5V	10,5V	1,5 / 7,6	12,5 / 3000			
	54C/74C	+5V	1,5V	3,5V	0,4 / 0,36	2,5 / 100		Ø	5
		+10V	2,0V	8,0V	10,0 / 10μA	9 / 10		Ø	10
		+15V	4,5V	10,5V	1,5 / 7,8	13,5 / 3500			
	14500	+5V	1,5V	3,5V	0,4 / 0,8	2,5 / 1700		Ø	5
		+10V	3,0V	7,0V	0,5 / 2,0	9,5 / 900		Ø	10
		+15V	4,5V	10,5V	1,5 / 7,8	12,5 / 3500			

**Tabela 1 – Características DC das famílias lógicas de CI's.**

Neste texto serão abordadas em especial as interligações CMOS – TTL e TTL – CMOS.

## 15.2 INTERFACES CMOS – TTL

### 15.2.1 INTERFACES CMOS – TTL DE BAIXA POTÊNCIA (Low-Power)

Na tabela 1, vê-se que os CMOS com alimentação de +5V tem as seguintes tensões e correntes de saídas:

#### a) NÍVEL LÓGICO Ø

##### Tensões:

- Tensão de saída máxima = 0,4 volts para CMOS 7
- Tensão de entrada máxima = 0,7 volts para LSTTL

##### Correntes:

- Corrente de saída mínima, que é a corrente que a saída pode absorver quando está baixa ( $I_{OL}$ ) = 0,3mA para CMOS 74C.
- Corrente de entrada máxima, que é a corrente que o LSTTL pode fornecer quando sua entrada estiver no nível lógico Ø ( $I_{IL}$ ) = 0,36 mA para LSTTL. Então, no nível Ø a saída CMOS pode suportar até 1 carga LSTTL:

$$\frac{I_{OL}}{I_{IL}} = \frac{0,4}{0,36} = 1$$

#### b) NÍVEL LÓGICO 1

##### Tensões:

- Tensão de saída mínima = 2,5 volts para o CMOS 74C
- Tensão de entrada mínima = 2,0 volts para LSTTL

##### Correntes:

- Corrente de saída alta = 100 µA para o CMOS 74C
- Corrente de entrada alta = 20 µA para o LSTTL

Disto pode-se ver que a ligação Ø limita o número de cargas LSTTL na saída do CMOS 74 C a 1 carga.

O manual do fabricante traz indicações para interfaces.

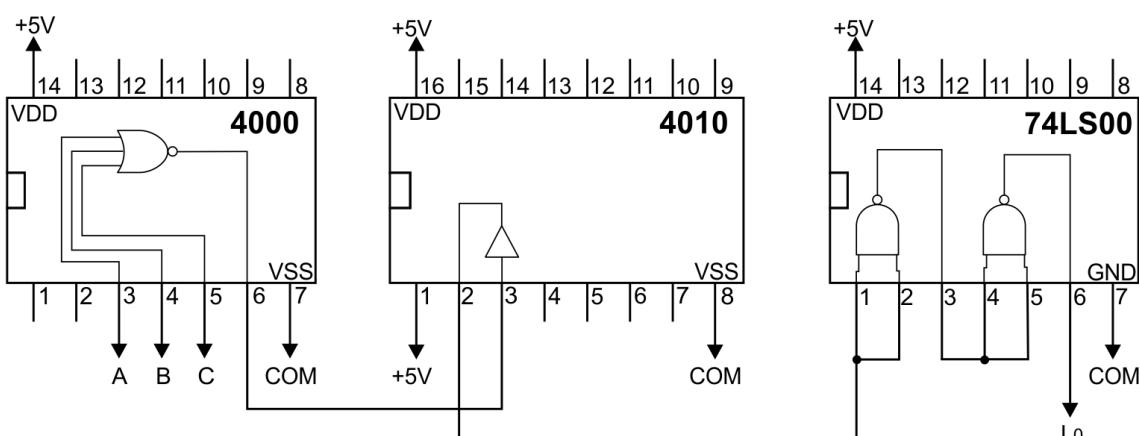
### 15.2.2 CMOS PARA TTL STANDARD

Os gates CMOS com alimentação +5V não suportam uma carga TTL. A explicação é a seguinte: quando a saída CMOS é alta “1” não há problema de tensão com TTL, nem problema de corrente, pois a corrente de entrada máxima do TTL, nem problema de corrente, pois a corrente de entrada máxima do TTL é de  $40\mu A$ , a qual é facilmente fornecida pela saída do CMOS. Entretanto, quando a saída é baixa “0”, ocorre um problema devido ao IIL do TTL. Esta corrente que é tipicamente 1,6 mA tem de fluir através da saída CMOS, em outras palavras a saída CMOS deve absorver esta corrente de 1,6mA através da resistência ligada ( $R_{ligada}$ ) do dispositivo canal N. O valor de  $R_{ligada}$  varia para diversos circuitos CMOS entre  $100\Omega$  a  $52\Omega$ . Em muitos casos, a corrente de 1,6mA através da resistência CMOS produz uma tensão na saída do CMOS que é muito alta para satisfazer a exigência de  $V_{IL}$  do TTL. Recorde que  $V_{IL} \leq 0,8V$  para TTL. Lembre-se também que é necessário limitar a tensão de entrada baixa a 0,4 volts, a fim de manter a margem de ruído de 0,4 volts, características dos circuitos TTL. Algumas saídas CMOS podem preencher esta exigência, enquanto outras não podem. De fato, alguns circuitos CMOS são projetados para absorver até 6mA, podendo suportar uma carga de três ou quatro cargas TTL. Estes circuitos especiais são chamados buffers e podem ser usados entre a saída do CMOS convencional e diversas cargas TTL. Entre estes buffers pode-se citar na série 4000, o 4009 e 4010.

#### 15.2.2.1 EXPERIÊNCIA 1

Nesta experiência os CMOS funcionam com +5 volts.

- a) Material Utilizado: 1 x CI 74LS00, 1 x CI 4000 e 1 x CI 4010
- b) Montar o circuito da figura 1.



**Figura 1 – Interfaces CMOS – TTL usando Buffer, com CMOS trabalhando com +5V.**

- c) Completar a tabela 2.

ENTRADAS CMOS			SAÍDAS TTL
A	B	C	$L\bar{O} = \overline{A+B+C}$
Ø	Ø	Ø	
Ø	Ø	1	
Ø	1	1	
Ø	1	Ø	
1	1	Ø	
1	1	1	
1	Ø	1	
1	Ø	Ø	

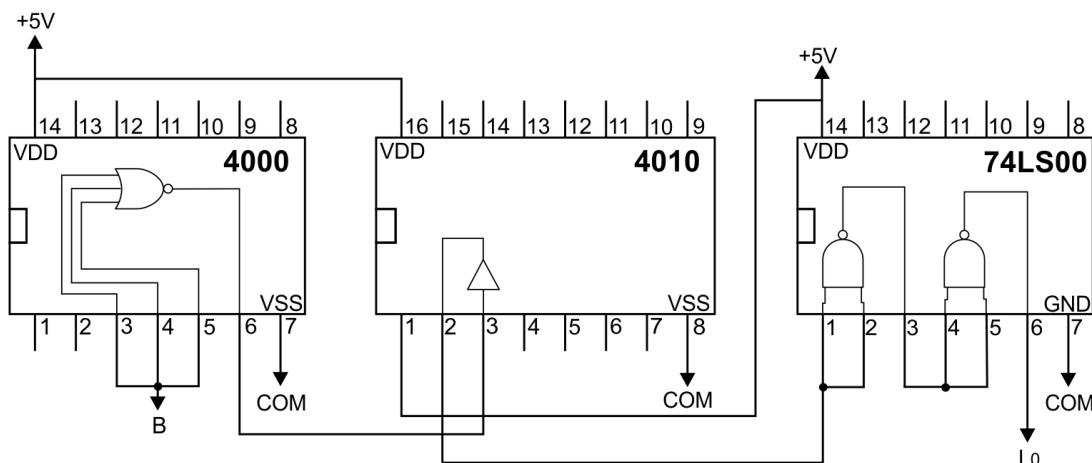
**Tabela 2 – Tabela para o circuito da figura 1.**

- d) Da tabela 2, pode-se observar que a saída  $L\bar{O} = A+B+C$ . Neste caso, foi usado o buffer 4010 para se fazer o interface CMOS 4000 para TTL 74LS00.

### 15.2.2.2 EXPERIÊNCIA 2

Nesta Experiência será usado o buffer CMOS 4010, para fazer a interface entre o CMOS 4000 funcionando com uma fonte VDD = + 15 volts e TTL 74LS00 funcionando com +5 volts.

- a) Material Utilizado: 1 x CI 74LS00, 1 x CI 4000 e 1 x CI 4010  
b) Montar o circuito da figura 2.



**Obs :** Não ligue o terminal T a nenhuma chave, e sim como na tabela 3.

**Figura 2 – Interface CMOS – TTL usando buffer, com CMOS trabalhando com +15V.**

- c) Completar a tabela 3.

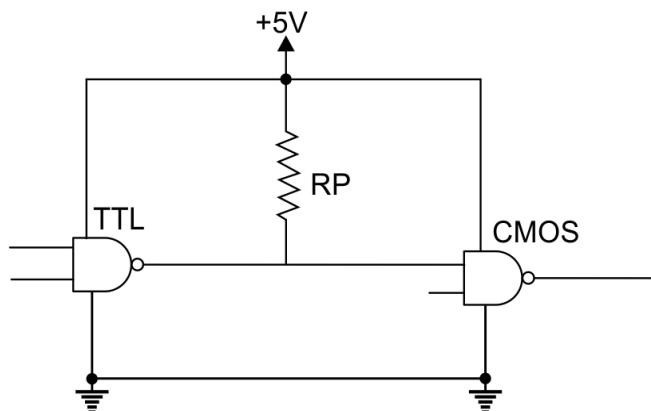
ENTRADA CMOS	SAÍDA TTL
T	$L \emptyset = \overline{T}$
Comum ( $\emptyset$ )	
+15 volts	

**Tabela 3 – Tabela para o circuito da figura 2.**

- d) Da tabela 3, pode-se observar que a saída  $L\emptyset$  confere com expressão lógica  $L\emptyset = \overline{T}$ , pois quando T estiver conectado ao comum (nível lógico  $\emptyset$ ) a saída  $L\emptyset = 1$  e quando T estiver conectada a fonte de +15 volts (nível lógico 1) a saída  $L\emptyset = \emptyset$ .

### 15.2.3 INTERFACE TTL – CMOS

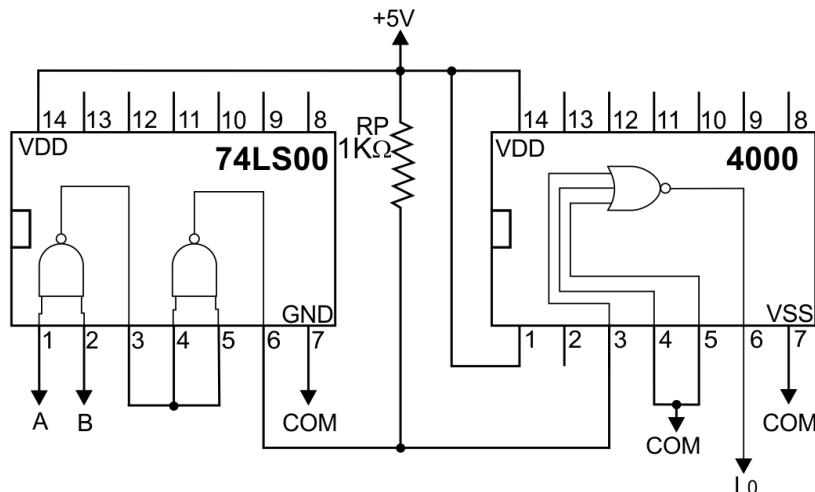
Quando na saída de um gate TTL tem-se uma carga CMOS não há problema no estado baixo “ $\emptyset$ ”, desde que  $V_{\emptyset L(\max)} = 0,4$  volts para a saída TTL e a entrada CMOS aceitará qualquer valor até 1,5 volts para o nível “ $\emptyset$ ”. Contudo, no estado alto pode ocorrer um problema, pois a tensão de saída “1” dos TTL sem carga não é igual a +15V mas sim, tipicamente, em torno de 3,6 volts. Esta tensão de 3,6 volts é realmente suficiente para entrada CMOS que requer uma tensão de entrada superior ou igual a 3,5 volts porém, como 3,6 volts a margem de ruído é de 0,1 volts, o que é desejável. Portanto, é recomendado que um resistor pull-up externo RP seja usado como mostrado na figura 6. O efeito de RP é aumentar  $V_{\emptyset H}$  do circuito TTL para mais próximo de +5V. RP é escolhido do mesmo modo que o resistor RC é escolhido para circuitos de coletor aberto.



**Figura 3 – Interface TTL – CMOS com resistor pull-up.**

### 15.2.3.1 EXPERIÊNCIA 3

- a) Material Utilizado: 1 x CI 74LS00, 1 x CI 4000 e 1 Resistor de 1KΩ  
b) Montar o circuito da figura 4.



**Figura 4 – Interface TTL – CMOS.**

- c) Completar a tabela 4.

ENTRADAS TTL		SAÍDAS CMOS
A	B	$L\bar{O} = \overline{AB}$
Ø	Ø	
Ø	1	
1	1	
1	Ø	

**Tabela 4 – Tabela para o circuito da figura 4.**

- d) Da tabela 4, pode-se observar que a saída  $L\bar{O}$  confere com a expressão lógica  $L\bar{O} = AB$ , pois  $L\bar{O}$  só será 1 quando  $A = B = 0$ .

## 15.3 CIRCUITOS INTEGRADOS SERVINDO DE INTERFACES

Os fabricantes desenvolveram muitos CI's destinados a resolver os problemas de interfaces. Estes CI's existem muito nas famílias MOS como nas famílias bipolares. É aconselhável a consulta do Guia de Seleção de Drivers dos fabricantes, que auxiliam na seleção de circuitos para interfaces.