

**Ministério da Educação e do Desporto  
Centro Federal de Educação Tecnológica do Paraná  
Departamento Acadêmico de Eletrônica**

# **ELETRÔNICA DIGITAL I**

**Disciplina: Eletrônica Digital I**

## 1 - SISTEMAS DE NUMERAÇÃO

- 1.1 - Introdução
- 1.2 - Notação Posicional - Transformação Binário-Decimal
- 1.3 - Transformação Decimal-Binário
- 1.4 - Sistemas Octal e Hexadecimal
- 1.5 - Códigos Binários
- 1.6 - Exercícios resolvidos
- 1.7 - Exercícios propostos

### 1.1 - INTRODUÇÃO

A representação escrita de quantidades numéricas se faz através dos algarismos e de um conjunto de normas que constituem o sistema de numeração. Para nós, o sistema de base dez, ou decimal, é o utilizado desde a infância, e nos acostumamos a ele. O fato de termos nascido com dez dedos foi, certamente, o fator que determinou a sua escolha.

Há quem diga que se tivéssemos adotado o número doze como base de nosso sistema, teríamos mais flexibilidade nas operações pois ele possui maior quantidade de submúltiplos que o dez. Quando vamos comprar ovos, por exemplo, fazemos uso dele, assim como utilizamos o sistema sexagesimal nas medidas de tempo.

Quando tratamos com circuitos eletrônicos digitais, um outro sistema é utilizado: o sistema binário. Acontece que o sistema decimal de numeração, com seus dez algarismos, encontraria dificuldades técnicas na representação de seus elementos. No mínimo precisamos de dez níveis distintos de tensão em um circuito, dando margem a interferências e confusões incompatíveis com a precisão requerida nos instrumentos de cálculo, onde a eletrônica digital é largamente utilizada. A simplicidade do sistema binário, exigindo apenas dois níveis distintos de tensão para a representação de seus dígitos, os "bits" (binary digits) 0 e 1, vem a calhar.

### 1.2 - NOTAÇÃO POSICIONAL - TRANSFORMAÇÃO BINÁRIO-DECIMAL

A base de um sistema é o número que serve de referência para a escrita de qualquer outro, estando intimamente ligado à posição que um algarismo ocupa em relação aos demais.

Por exemplo, o número 1979 é decomposto em função da base dez, podendo ser escrito sob forma de soma:

$$1.10^3 + 9.10^2 + 7.10^1 + 9.10^0$$

Nota-se acima que os algarismos que compõem 1979 recebem pesos diferentes de acordo com a sua posição no número. Esse peso nada mais é que uma potência da base, no caso, de dez. A essa forma de representação de quantidades por um conjunto de algarismos devidamente posicionados dá-se o nome de "Notação Posicional".

O sistema binário, que utiliza apenas dois algarismos, também obedece a norma da notação posicional, mudando somente a base do sistema que passa a ser o número 2. Se desejarmos saber quanto vale um número em qualquer sistema, recorreremos à decomposição do número como foi feito anteriormente e, tendo prévio conhecimento das potências de dois, as dificuldades desaparecem.

Seja, por exemplo, converter o número  $101\frac{1}{2}$  (o índice indica a base do sistema) a uma forma mais familiar. Aplicando-se a regra obtemos:

$$1.2^2 + 0.2^1 + 1.2^0 = 5$$

Portanto,  $101$  equivale ao nosso conhecido 5.

Os números que possuem parte fracionária também não devem apresentar dificuldades. Lembremo-nos que, à direita da vírgula, os expoentes da base tornam-se negativos em decorrência de seu decréscimo observado:

$$0,35 = 3.10^{-1} + 5.10^{-2} = 0,3 + 0,05$$

No sistema binário, novamente a regra se mantém.

$$0,11_2 = 1.2^{-1} + 1.2^{-2} = 0,75$$

Com o exposto acima, as transformações do sistema binário ao decimal deixam de ser segredo. Passemos, portanto, às transformações inversas.

### 1.3 - TRANSFORMAÇÃO DECIMAL-BINÁRIO

Aqui, devido às dificuldades que temos em trabalhar com o sistema binário, essas transformações não são tão óbvias. Assim, recorremos a uma regra prática para a sua efetuação. Consiste ela em se determinar quantas vezes o 2 "cabe" em um número. Hei-la:

"Divide-se o número por 2 e o mesmo se faz com o quociente obtido, procedendo-se assim sucessivamente até que resulta a unidade como último quociente. Tomam-se então, os restos obtidos em cada divisão, dispondo-os ao lado do quociente unitário na ordem inversa à que aparecem para se obter o número binário correspondente".

O exemplo abaixo é mais esclarecedor:

$$\begin{array}{r}
 23 \quad | \quad 2 \\
 1 \quad 11 \quad | \quad 2 \\
 \quad 1 \quad 5 \quad | \quad 2 \\
 \quad \quad 1 \quad 2 \quad | \quad 2 \\
 \quad \quad \quad 0 \quad 1
 \end{array}$$

$$23 = 10111_2$$

Se existem frações o processo se modifica:

"Multiplica-se a parte fracionária por 2 e separa-se a parte inteira do produto. Repete-se sucessivamente a operação até que resulte 0 ou que se chegue a uma aproximação satisfatória, pois as partes inteiras que surgem, colocadas sempre a direita das demais, compõem o binário correspondente".

Assim, 0,8 pode ser representado pela forma aproximada

$$\begin{array}{r}
 0,8 \\
 \times 2 \\
 \hline
 1 \quad ,6 \\
 \times 2 \\
 \hline
 1 \quad ,2 \\
 \times 2 \\
 \hline
 0 \quad ,4 \\
 \times 2 \\
 \hline
 0 \quad ,8 \\
 \times 2 \\
 \hline
 1 \quad ,6
 \end{array}
 \quad 0,8 = 0,11001$$

Pode-se notar que, por mais que o processo se repita, nunca encontraremos o resultado nulo; apenas um maior número de casas à direita da vírgula com o correspondente aumento de precisão no resultado.

### 1.4 - SISTEMA OCTAL E HEXADECIMAL

Embora não empregados diretamente pelos circuitos eletrônicos digitais, os sistemas octal e hexadecimal são de grande valia na representação de números binários, que muitas vezes são extensos, devido a seus poucos dígitos. Sendo o 8 e o 16 potências de 2, as transformações entre estes sistemas são imediatas, resultando daí vantagens em seu uso.

O Sistema Octal utiliza apenas os 8 primeiros algarismos decimais (0 a 7) pois a base de um sistema coincide com a quantidade de símbolos que este dispõe para representar qualquer número.

Em relação às transformações que possam ocorrer entre esse e o sistema decimal, cumpre dizer que os processos não se alteram, apenas a base é que se modifica.

Os exemplos confirmam:

1º) Transformar para o sistema decimal:

$$\begin{array}{l}
 21,4_8 \\
 21,4_8 = 2 \cdot 8^1 + 1 \cdot 8^0 + 4 \cdot 8^{-1} = 17,5
 \end{array}$$

Obs: A ausência de índice no número indica que o sistema empregado é o decimal.

2º) Transformar para o sistema octal:

$$\begin{array}{r}
 93,8 \\
 93 \quad | \quad 8 \\
 5 \quad 11 \quad | \quad 8 \\
 \quad 3 \quad 1
 \end{array}
 \quad \begin{array}{r}
 0,8 \\
 \times 8 \\
 \hline
 6 \quad ,4 \\
 \times 8 \\
 \hline
 3 \quad ,2
 \end{array}$$

$$93,8 = 135,63_8$$

Já as transformações entre os sistemas octal e binário são imediatas. Basta corresponder a cada algarismo octal o conjunto de 3 bits de mesmo valor e realizar substituições como feito abaixo:

19) Transformar  $100.101\ 1,01_2$

001.001.011.010  
1 1 3 2  $113,2_8$

29) Transformar para o binário:  $32,6_8$

3 2 ,6  
011 010 110  $32,6_8 = 11010,110_2$

Para maior facilidade nas transformações, representamos a seguir as correspondências entre o octal e o binário.

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

O sistema hexadecimal apresenta uma pequena novidade. Como os dez algarismos do sistema decimal não são suficientes para formar o conjunto necessário à apresentação de números nesse sistema, faz-se uso de 6 letras maiúsculas como complemento.

DECIMAL	HEXADECIMAL	BINÁRIO
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Como também aqui não ocorre mudanças nas regras de transformação, vamos aos exemplos.

19) Transformar para o hexadecimal o número 621.

621  $\overline{16}$   
141 38  $\overline{16}$   
13 6 2

$621 = 26D_{16}$

(D = 13)

29) Transformar do hexadecimal para o decimal:

$3F, C_{16}$

$$3F, C_{16} = 3 \cdot 16^1 + 15 \cdot 16^0 + 12 \cdot 16^{-1} =$$

$$48 + 15 + 0,75 = 63,75$$

39) Transformar do binário para o hexadecimal:

1.0110.1011  
1 6 B =  $16B_{16}$

49) Transformar do hexadecimal para o binário:

$A5, D_{16}$

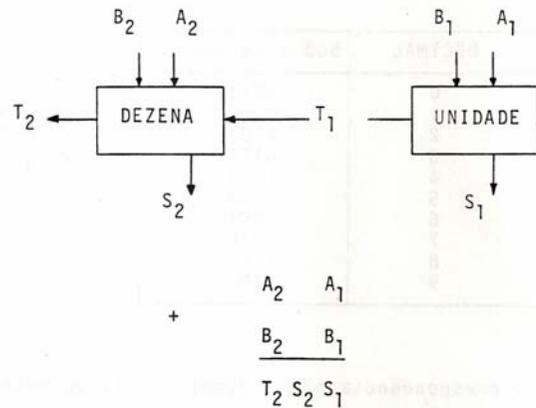
A 5 ,D  
1010 0101 1101

$$A5, D_{16} = 1010\ 0101, 1101_2$$

### 1.5 - CÓDIGOS BINÁRIOS

Os circuitos digitais conseguem unir as vantagens dos sistemas binário e decimal através da utilização de códigos especiais que permitem a representação dos nossos familiares algarismos decimais como combinações de bits.

Essa hibridação de sistemas pode ser entendida ao se observar os blocos somadores de um estágio eletrônico digital.



As referências a UNIDADE e DEZENA, indicam que os números somados o fazem segundo as regras do sistema decimal, sendo  $A_0A_1$ ,  $B_0$  e  $B_1$  algarismos desse mesmo sistema. Mas, o circuito eletrônico que realiza a soma tem natureza binária, estando, os citados algarismos, representados, cada um, por um conjunto de bits previamente codificados. Esses conjuntos de bits recebem o nome de "palavras" ou "bytes".

Alguns dos sistemas de codificação são analisados a seguir:

#### 19) Decimal Codificado em Binário (BCD) natural.

DECIMAL	BCD NATURAL
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Neste código, cada algarismo decimal é representado pelo seu correspondente binário, escrito sob forma de byte de 4 bits. É um código simples em que os bits recebem pesos de acordo com sua posição; da esquerda para a direita, são eles: 8, 4, 2 e 1.

#### 29) BCD 3 em excesso

DECIMAL	BCD 3 em excesso
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Aqui a correspondência não é direta, pois se nota que os bytes representam sempre três unidades a mais que os algarismos que lhes correspondem. Essa é uma vantagem, relativa ao BCD natural, pois facilita os ajustes decimais que devem ocorrer depois dos cálculos realizados pelos blocos somadores.

#### 39) Código Biquinário

É um código em que os bytes são formados por 7 bits, divididos em grupos de dois e cinco. Uma norma é estabelecida para a representação dos algarismos decimais que determina a existência de apenas 2 bits "1" por byte, estando cada um em um dos grupos.

DECIMAL	BIQUINÁRIO
	PESOS 50 43210
0	01 00001
1	01 00010
2	01 00100
3	01 01000
4	01 10000
5	10 00001
6	10 00010
7	10 00100
8	10 01000
9	10 10000

Embora haja um excesso de bits, existe uma vantagem que é a constância no número de bits 1. Isso permite a detecção de erros quando da transmissão de bytes por um meio qualquer. Se aparecer apenas 1, ou mais de dois, bits em um byte, sabe-se que houve perda de informação ou interferência externa na transmissão. A mensagem então é



## 1.7 - EXERCÍCIOS PROPOSTOS

19) converter para o sistema decimal:

$$\begin{aligned} 1011_2 &= \\ 10111_2 &= \\ 1010111_2 &= \\ 36_8 &= \\ 325_8 &= \\ 432_8 &= \\ 4D_{16} &= \\ 59_{16} &= \\ ABC_{16} &= \end{aligned}$$

29) Converter para o sistema decimal:

$$\begin{aligned} 0,001_2 &= \\ 0,1101_2 &= \\ 0,001101_2 &= \\ 0,3_8 &= \\ 0,46_8 &= \\ 0,326_8 &= \\ 0,5_{16} &= \\ 0,3A_{16} &= \\ 0,ABC_{16} &= \end{aligned}$$

39) Transformar para o sistema binário:

$$\begin{aligned} 7 &= \\ 13 &= \\ 162 &= \\ 0,5 &= \\ 0,15 &= \\ 0,032 &= \end{aligned}$$

49) Transformar para os sistema octal e hexadecimal o número:

ro:

97,3

59) Converter diretamente ao sistema binário

$$\begin{aligned} 16_8 &= \\ 46_8 &= \\ 27,3_8 &= \\ AB_{16} &= \\ 736_{16} &= \\ 4B,7C_{16} &= \end{aligned}$$

69) Converter para os sistemas octal e hexadecimal:

$$\begin{aligned} 1011 &= \\ 10\ 10111 &= \\ 110111,101 &= \end{aligned}$$

79) Representar nos códigos BCD natural, no 3 em excesso e no biquinário o número 491.

89) Citar pelo menos uma vantagem relativa de cada um dos seguintes sistemas de codificação binário-decimal:

BCD natural  
BCD 3 em excesso  
Biquinário  
Gray

99) Converter o número 0101 1000 0111 escrito em BCD 3 em excesso para:

a) Sistema decimal  
b) Sistema binário  
c) Sistema octal  
d) Sistema hexadecimal

109) Supondo  $\alpha$  e  $\beta$  os símbolos complementares para representação de algarismos no sistema duodecimal (base 12), converter o decimal 131,5 para esse sistema.

## 2 - ÁLGEBRA BOOLEANA

- 2.1 - Introdução
- 2.2 - Funções Básicas: E, OU e NÃO.
- 2.3 - Funções NE, NOU e EXC. OU
- 2.4 - Propriedades
- 2.5 - Teoremas
- 2.6 - Problemas Resolvidos
- 2.7 - Problemas Propostos

### 2.1 - INTRODUÇÃO

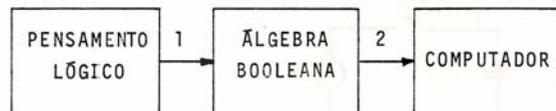
Muito antes do surgimento do primeiro computador eletrônico, o ENIAC, construído em 1946, o matemático George Boole desenvolveu em 1854, quase um século antes, a Álgebra Booleana. A sua finalidade era o estudo da lógica, que vinha sendo feita já há muitos séculos porém carecia de um formalismo matemático.

E a grande aplicação dessa Álgebra veio com o advento dos computadores, que empregam a lógica nos seus processamentos e devem ter precisão matemática.

O "raciocínio" de um computador é realizado pelos circuitos lógicos, como se fossem neurônios, que são combinados segundo as regras da Álgebra Booleana. Esta é, portanto, o elo de ligação entre o cérebro humano e a máquina. A realização de um circuito capaz de desenvolver certas formas de raciocínio é basicamente constituída por 2 processos.

1º) A formulação booleana do raciocínio que a máquina deve possuir e,

2º) A elaboração do circuito eletrônico correspondente ao resultado obtido da álgebra.



A primeira etapa é conseguida inicialmente pela formulação do raciocínio de forma comum, por exemplo:

"Irei à praia se sair o sol"

Matematicamente, a proposição acima pode ser expressa, desde que algumas considerações sejam feitas.

Em primeiro lugar passamos a representar as possíveis variáveis do raciocínio por letras maiúsculas. A ida à praia, fazemos corresponder a letra P e a saída do sol, a letra S. Essas letras maiúsculas são as variáveis booleanas, sujeitas a assumir qualquer um dos seguintes valores:

- (1) se representar verdade
- (0) se representar falsidade

Podemos escrever expressões matemáticas em substituição às ideias.

- P = 1 (irei à praia)
- P = 0 (não irei à praia)
- S = 1 (sairá sol)
- S = 0 (não sairá sol)

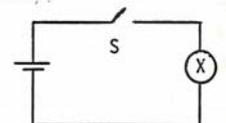
A relação lógica entre essas duas variáveis é uma relação de igualdade, ou seja:

$$P = S,$$

que corresponde à expressão original.

Como vimos, a álgebra booleana trabalha com 2 valores (0 e 1) o que a torna possível de ser empregada em circuitos digitais, vistos que o sistema binário utilizado por tais elementos também os emprega.

Falta agora elaborarmos o circuito que desempenha a função  $P = S$ . Aqui, passamos a outras convenções, agora com chaves e lâmpadas. Vejamos a figura seguinte:



Associando-se possíveis estados da chave S com os bits 0 e 1, temos:

S = 0 - chave aberta  
S = 1 - chave fechada

Da mesma forma:

P = 0 - lâmpada apagada  
P = 1 - lâmpada acesa

Como a lâmpada só acenderá quando a chave for fechada, mantendo-se apagada se isso não ocorrer, pode-se escrever também:

$P = S$

Como vimos, um circuito elétrico é capaz de representar, através dos estados de seus elementos, as diversas condições que decorrem de um raciocínio lógico.

## 2.2 - FUNÇÕES BÁSICAS: E, OU e NÃO

Assim como a matemática com a qual estamos acostumados a trabalhar possui operações fundamentais de adição, subtração, multiplicação e divisão, a álgebra booleana também se apóia em determinadas funções básicas que são conhecidas por: E, OU e NÃO.

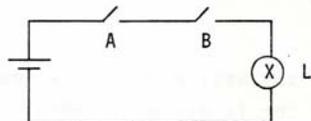
A função E relaciona 3 ou mais variáveis de tal forma que uma delas, chamada "variável dependente", só assumirá o valor "1" se todas as demais variáveis independentes também se tornam iguais a 1.

Sejam A, B e L três variáveis booleanas tal que L depende de A e B através da função E. Matematicamente podemos escrever:

$$L = A \cdot B$$

onde (.) representa a função E que relaciona A e B.

O circuito elétrico correspondente é o dado abaixo, conservando-se as convenções adotadas para chaves e lâmpadas.



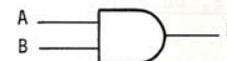
A lâmpada acenderá se as chaves A e B estiverem fechadas.

Como as duas chaves podem assumir diversos estados combinados é interessante expressarmos esses estados em forma de tabela. Obtemos então a Tabela da Verdade da função E, que sintetiza o seu significado:

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

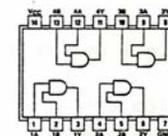
L será 1 somente quando A e B o forem.

Eletronicamente, a função E é representada pelo símbolo abaixo:

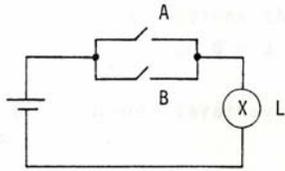


Os níveis de tensão aplicados aos terminais de entrada de terminam o nível de saída, que será alto somente quando ambos os de entrada o forem.

Sob a forma de circuito integrado, a função E é obtida pela utilização, por exemplo, do SN7408 que se constitui de 4 portas E de 2 entradas cada uma e dois terminais para alimentação (5V e massa). O diagrama interno desse CI é apresentado a seguir:



A função OU, por sua vez, apresenta a seguinte característica: L será 1 se A "ou" B for 1. Assim, um modelo simples de circuito com chaves, tem o aspecto apresentado a seguir.



As chaves em paralelo possibilitam a lâmpada acender desde que a chave A ou a chave B esteja acesa, ou ambas. A álgebra booleana reserva para símbolo o sinal (+), como segue:

$$L = A + B$$

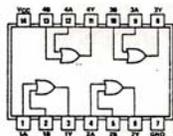
Facilmente se obtêm a Tabela da Verdade

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

Convencionou-se adotar como símbolo para tal função o apresentado a seguir; observando-se que o número de entradas não se limita a duas como em nossos exemplos, podendo variar muito com a aplicação que se quer dar ao circuito.



Da mesma série 74, o representante típico do circuito porta OU é o SN7432:

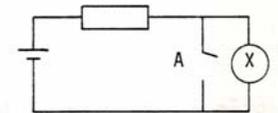


A terceira função lógica básica é desempenhada pelo circuito porta NÃO ou INVERSOR que tem a propriedade de proporcionar a saída o complemento do estado de entrada. Assim, sendo A a entrada e L a saída,  $L = 1$  para  $A = 0$  e  $L = 0$  para  $A = 1$ .

Uma barra sobre a variável independente possibilita a representação:

$$L = \bar{A}$$

para o circuito com chave e lâmpada:



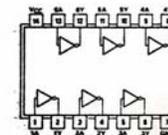
O fechamento da chave implica em lâmpada apagada pois a queda de tensão sobre a lâmpada se anula. Portanto, tem-se:

A	L
0	1
1	0

O símbolo adotado para a porta NÃO é o seguinte:



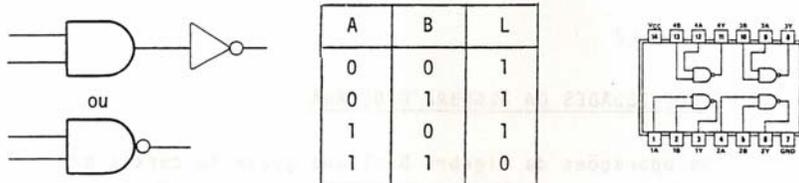
SN7404 é um exemplo típico de circuito digital inversor, apresentando um total de 6 portas em um só integrado.



Essas três funções são básicas e todas as outras decorrem de simples associação destas, como as estudadas a seguir.

## 2.3 - FUNÇÕES NE, NOU e EXCLUSIVAMENTE OU

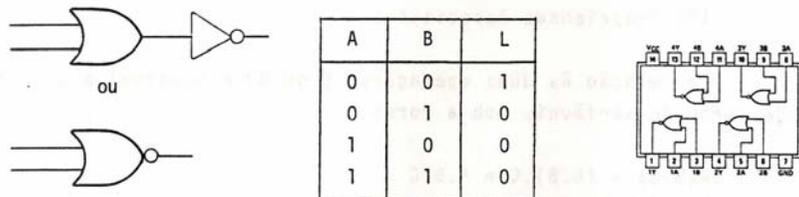
Se acoplarmos a saída de uma porta E outra porta que realize a função NÃO, obtemos uma das funções mais utilizadas em circuitos lógicos, A porta NE (NÃO + E) como é mostrada a seguir, juntamente com sua tabela da verdade e o SN7400, com suas 4 portas NE de 2 entradas.



A	B	L
0	0	1
0	1	1
1	0	1
1	1	0

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

Função similar pode ser conseguida com as portas OU e NÃO dando origem à função NOU que pode ser desempenhada pelo SN7402.



A	B	L
0	0	1
0	1	0
1	0	0
1	1	0

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

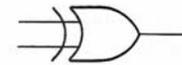
Outra função que desempenha relevante papel entre os circuitos lógicos é a EXCLUSIVAMENTE OU que é definida como segue:

$L = 1$  se A e B forem diferentes e  $L = 0$  se A e B forem iguais.

Essa característica permite que se utilize a EXCLUSIVAMENTE OU como elemento comparador de bits. Em muitos sistemas as mesmas operações são realizadas mais de uma vez e os resultados obtidos são comparados, bit a bit, com a finalidade de se detectar erro de cálculo. Se o resultado da comparação for positivo, pode-se acreditar que há grande probabilidade de ser correto o resultado da operação.

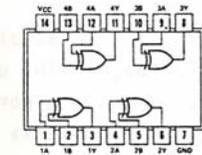
A função EXCLUSIVAMENTE OU pode ser eletronicamente realizada pelo CI SN7486.

## SÍMBOLO



## TV

A	B	L
0	0	0
0	1	1
1	0	1
1	1	0



$$L = A \oplus B$$

## 2.4 - PROPRIEDADES DA ÁLGEBRA BOOLEANA

As operações da Álgebra Booleana gozam de certas propriedades que possibilitam a simplificação de equações lógicas e, em consequência, minimizam os custos dos correspondentes circuitos. O domínio dessas propriedades e das relações lógicas que os seguem são valiosas para quem pretende tirar o máximo proveito em economia no projeto de circuitos básicos combinacionais.

## 1ª) Propriedade Associativa

Em relação às duas operações, E ou OU é possível a associação de parte de variáveis sob a forma:

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

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

## 2ª) Propriedade Comutativa

Ainda em relação às duas operações anteriores, é possível comutar a ordem das variáveis como segue:

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

$$A + B = B + A$$

## 3ª) Propriedade Distributiva

Em relação à soma.

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

Em relação ao produto.

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

Neste ponto é interessante se fazer uma observação. Em primeiro lugar, a notação utilizada é semelhante àquela que se refere às operações de adição e multiplicação, quanto à prioridade na realização de operações onde não haja parênteses ou qualquer outro sinal de mesma espécie: a função E deve ser operada antes da função OU. Em segundo lugar, embora os sinais (.) e (+) não tenham nada a haver com multiplicação e adição, a não ser a semelhança, costuma-se chamar aos resultados das operações lógicas E ou OU, de produto e soma, respectivamente.

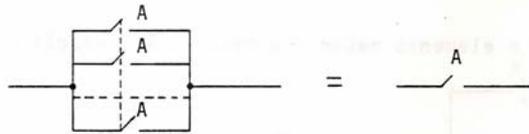
## 2.5 - RELAÇÕES LÓGICAS

$$1) A = A$$

Como uma variável booleana só pode assumir dois valores que são complementares, a dupla complementação implica na volta ao valor original.

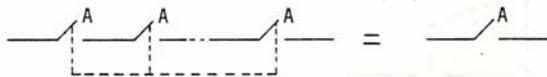
$$2) A + A + \dots + A = A$$

Já é de nosso conhecimento que a operação OU entre duas variáveis pode ser realizado por um sistema de chaves paralelas. Ora, a expressão acima significa a associação de um número qualquer de chaves que assumem, simultaneamente, o mesmo valor A, isto é, ou estão todas fechadas ou todas abertas ao mesmo tempo. Decorre que é válida a substituição desse sistema por outro constituído por uma simples chave.



$$3) A.A \dots A = A$$

Análoga à anterior, pois a operação E é realizada por associação em série de chaves:



$$4) A.0 = 0$$

Uma chave aberta em série com qualquer coisa equivale a uma chave aberta:



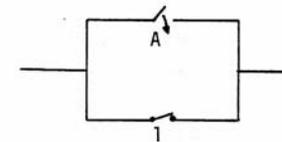
$$5) A.1 = A$$

0 1 é o elemento neutro em relação à operação E.



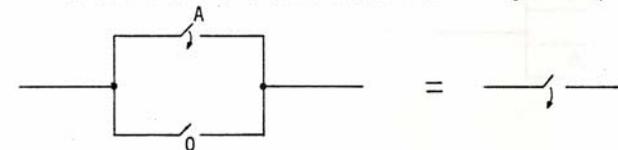
$$6) A + 1 = 1$$

Aqui a associação é em paralelo, logo a chave fechada curto-circuita a outra.



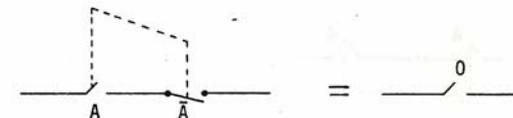
$$7) A + 0 = A$$

0 bit 0 é o elemento neutro em relação à operação OU:



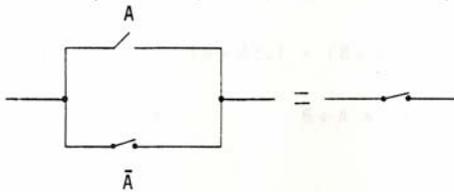
$$8) A.\bar{A} = 0$$

Duas chaves em série onde sempre vai existir uma em aberto:



9)  $A + \bar{A} = 1$

Duas chaves em paralelo, onde uma delas sempre estará fechada:



10)  $A.(A+B) = A$

Aqui, o raciocínio com circuitos elétricos equivalentes pode ser substituído pela análise baseada na Tabela da Verdade da função. Basta desenvolver os dois membros em termos de todas as possibilidades de combinação entre os estados das variáveis em jogo e compará-los no final:

A	B	A + B	A.(A + B)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

11)  $A + AB = A$

Da mesma forma:

A	B	AB	A + AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

12)  $A + \bar{A}B = A + B$

Podemos ainda utilizar o método de desenvolvimento da tabela da verdade ou demonstrar a relação pelo método algébrico, utilizando

do as propriedades e relações já definidas.

$$A + \bar{A}B = (A + \bar{A}).(A + B) \quad \text{propriedade distributiva do produto}$$

$$(A + \bar{A}).(A + B) = 1.(A + B) \quad \text{relação nº 9}$$

$$1.(A + B) = A + B \quad \text{relação nº 5} \quad \text{CQD}$$

13)  $A.(\bar{A} + B) = A.B$

Utilizando novamente o método algébrico:

$$A.(\bar{A} + B) = A.\bar{A} + AB \quad \text{propriedade distributiva da soma}$$

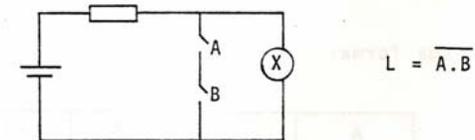
$$A.\bar{A} + AB = 0 + AB \quad \text{relação nº 8}$$

$$0 + AB = AB \quad \text{relação nº 7} \quad \text{CQD}$$

## 2.6 - PROBLEMAS RESOLVIDOS

2.6.1 - Elaborar um circuito elétrico utilizando chaves e lâmpadas que desempenhe a função NE:

Solução:



Consultando a tabela da verdade da função, nota-se que L somente será 0 (lâmpada apagada) quando ambas as chaves estiverem em 1 (posição fechada). No circuito acima, ocorre exatamente isso.

2.6.2 - Provar que a função EXCLUSIVAMENTE OU pode ser expressa de forma:

$$L = A \oplus B = \bar{A}.B + A\bar{B}$$

Solução:

Por desenvolvimento da tabela da verdade do último membro da expressão, chega-se a:

A	B	$\bar{A}$	$\bar{A}.B$	$\bar{B}$	$A\bar{B}$	$AB + A\bar{B}$
0	0	1	0	1	0	0
0	1	1	1	0	0	1
1	0	0	0	1	1	1
1	1	0	0	0	0	0

que corresponde à tabela fornecida no item 3 para a função EXCLUSIVAMENTE OU.

2.6.3 - Demonstrar a validade da propriedade distributiva da soma por desenvolvimento da tabela verdade.

Solução:

$$A.(B + C) = AB + AC$$

A	B	C	B + C	A.(B+C)	AB	AC	AB + AC
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

2.6.4 - Provar pelo método algébrico a relação lógica de nº 11:

$$A + A.B = A$$

$$A + A.B = A.1 + A.B \quad \text{relação nº 5}$$

$$A.1 + A.B = A(1 + B) \quad \text{Propriedade distributiva}$$

$$A.(1 + B) = A.1 \quad \text{relação nº 6}$$

$$A.1 = A \quad \text{relação nº 5}$$

2.6.5 - Provar que  $A \oplus 0 = A$

Solução:

Desenvolvendo  $\oplus$  como em 2.6.2, fica:

$$A \oplus 0 = \bar{A}.0 + \bar{0}.A$$

$$\bar{A}.0 + \bar{0}.A = \bar{A}.0 + 1.A$$

$$\bar{A}.0 + 1.A = 0 + A \quad \text{relações nºs 4 e 5}$$

$$0 + A = A \quad \text{relação nº 7}$$

## 2.7 - PROBLEMAS PROPOSTOS

2.7.1 - Elaborar um circuito com chaves que desempenhe a função NOU.

2.7.2 - Elaborar outro circuito que desempenhe a função EXCLUSIVAMENTE OU.

2.7.3 - Provar, pelo método algébrico a relação nº 10:  
 $A.(A + B) = A$

2.7.4 - Por desenvolvimento da tabela da verdade, verificar a validade da propriedade distributiva do produto:

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

2.7.5 - Demonstre as seguintes identidades:

$$A \oplus 1 = \bar{A}$$

$$A \oplus A = 0$$

$$A \oplus \bar{A} = 1$$

$$A \oplus AB = A\bar{B}$$

$$A \oplus (A + B) = \bar{A}.B$$

### 3 - CIRCUITOS LÓGICOS COMBINACIONAIS

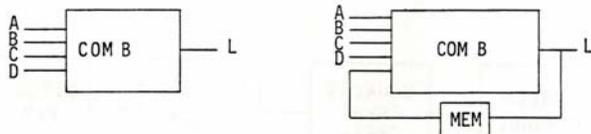
- 3.1 - Introdução
- 3.2 - Obtenção das Equações
- 3.3 - Técnicas de Simplificação
- 3.4 - Realização de Circuitos
- 3.5 - Problemas Resolvidos
- 3.6 - Problemas Propostos

#### 3.1 - INTRODUÇÃO

Os circuitos lógicos, de um modo geral, dividem-se em dois grandes grupos: os combinacionais e os seqüenciais.

Os combinacionais caracterizam-se pelo fato das saídas resultarem de simples combinações das variáveis de entrada, sem ocorrência de realimentações. Já os seqüenciais possuem dispositivos de armazenamento que reinjetam os sinais de saída nos circuitos, que se combinam com novas entradas, determinando novos estados de saída.

A figura da esquerda representa um circuito combinacional ao passo que o da direita corresponde ao seqüencial:



Exemplos típicos de circuitos combinacionais são os decodificadores e somadores; os seqüenciais se manifestam sob a forma de registros e contadores.

Os circuitos combinacionais são aqui tratados, ao passo que os seqüenciais aparecerão em capítulos posteriores.

Inicialmente, vejamos o processo que permite a realização

de um circuito lógico a partir das condições estabelecidas pelo projetista.

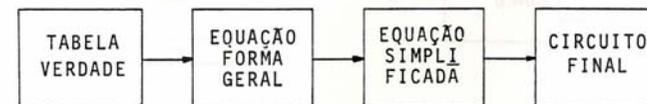
As condições lógicas que um determinado circuito deve obedecer são explicitadas na sua tabela da verdade, que relaciona o estado lógico das saídas com os das entradas. Essa tabela é, portanto, o passo inicial para a realização do circuito, pois sintetiza as suas funções.

De posse da tabela da verdade, podemos montar uma equação lógica que dê a função entre as variáveis de entrada e de saída na sua forma mais geral. Este é o segundo passo, o equacionamento da nossa ideia inicial.

A simplificação da equação obtida, sempre que possível, deve ser feita para que possamos economizar ao máximo as portas lógicas que constituirão o circuito final. Vários métodos de simplificação se impõem; veremos alguns deles. Esse passo exige conhecimento das relações lógicas válidas da álgebra booleana para que os resultados sejam seguros.

Finalmente, o circuito lógico pode ser esquematizado em função da equação obtida e, se todo o processo se desenvolveu corretamente, ele fornecerá estados de saída de acordo com a tabela da verdade.

As etapas do processo são abaixo representadas:



#### 3.2 - OBTENÇÃO DAS EQUAÇÕES

Suponhamos que um circuito combinacional de três entradas A, B e C deva proporcionar na saída L, os estados lógicos dados na tabela abaixo. Não questionaremos as razões que a fizeram assim, apenas vamos equacioná-la...

nº	A	B	C	L
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

O método conhecido por "soma de produtos" nos dá imediatamente a equação de L na sua forma geral ou canônica.

Ei-lo:

1º) Para cada condição em que L for igual a 1, faz-se o produto das variáveis de entrada, que devem ser negadas sempre que responderem ao estado 0.

2º) Somam-se os produtos assim obtidos e iguala-se tudo a L.

Voltando ao nosso exemplo verificamos que L toma o valor lógico 1 para 4 condições diferentes de entrada, as de números 0, 1, 3 e 7. Os produtos a que se referem o item 1º do método da soma de produtos serão os indicados abaixo, correspondentes às condições cujos números se lhes antepõem.

nºs	produtos
0	$\bar{A}.\bar{B}.\bar{C}$
1	$\bar{A}.\bar{B}.C$
3	$\bar{A}.B.C$
7	$A.B.C$

A soma desses produtos nos dá, de imediato:

$$L = \bar{A}.\bar{B}.\bar{C} + \bar{A}.\bar{B}.C + \bar{A}.B.C + A.B.C$$

que é a equação procurada.

É interessante tentarmos justificar o método, para o que recorreremos a equação obtida acima.

Sempre que uma das quatro condições consideradas surja na entrada do circuito, o produto que lhes corresponde toma o valor 1. Por exemplo, seja a condição nº 3 a aparecer (0, 1, 1). O produto  $\bar{A}.B.C$  resultará, obrigatoriamente, 1 pois sendo A igual a 0,  $\bar{A}$  passa a valer 1, e sendo as demais variáveis iguais a 1, ocorre a afirmação. Acontece, porém, que L será igual a 1 para qualquer das condições em jogo, o que é possibilitado pela soma dos produtos. Ocorrendo ao menos um dos produtos com valor 1, o resultado da soma também o será, tendo em vista a definição da função OU.

Outro método que também pode ser empregado, principalmente quando existem muitos 1's na coluna de saída de uma dada tabela é o do produto de somas:

1º) Somam-se as variáveis de entrada, negando as que responderem a 1, em todas as condições em que L for 0.

2º) Realiza-se o produto dessas somas para a obtenção final do resultado.

O nosso mesmo exemplo daria como resultado:

$$L = (A + \bar{B} + C).(\bar{A} + B + C).(\bar{A} + B + \bar{C}).(\bar{A} + \bar{B} + C)$$

Note-se que aqui as condições consideradas são as de números 2, 4, 5 e 6. As somas correspondentes se anulam quando ocorre uma dessas condições na entrada, anulando toda a expressão.

Ambas as equações obtidas anteriormente podem ser realizadas na prática, dando resultados fiéis à tabela proposta, porém, é importante simplificá-la antes, para poupar gastos inúteis.

### 3.3 - TÉCNICAS DE SIMPLIFICAÇÃO

As expressões escritas sob a forma de soma de produtos, são simplificadas a partir de certos procedimentos já consagrados que se constitui na associação de produtos, levando-nos diretamente ao objetivo.

Os produtos que devem ser associados, são escolhidos de tal forma que a evidenciação de algumas variáveis comuns conduza a fatos da forma  $(A + \bar{A})$  como ocorre duas vezes na nossa expressão exemplo:

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$$

$$L = \bar{A}\bar{B}(\bar{C} + C) + BC(\bar{A} + A)$$

As quantidades entre parênteses valem 1 segundo a relação lógica vista anteriormente. Tem-se portanto:

$$L = \bar{A}\bar{B}.1 + BC.1$$

Entretanto o valor 1 é elemento neutro na operação E, o que permite escrever:

$$L = \bar{A}.\bar{B} + BC$$

como equação final.

De maneira análoga, a expressão escrita sob a forma de produto de somas, reduz-se a sua forma simplificada.

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

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

$$L = [(B + C) + (A.\bar{A})][(\bar{A} + B) + (C.\bar{C})]$$

$$L = [(B + C) + 0][(\bar{A} + B) + 0]$$

$$L = (B + C).(\bar{A} + B)$$

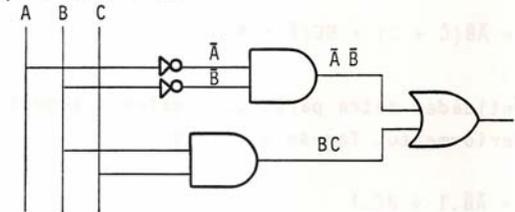
A expressão acima deve ser igual a obtida anteriormente para L. Realmente se substituirmos as variáveis A, B e C pelos seus valores numéricos, considerando qualquer combinação entre eles, os resultados serão iguais para ambas as expressões.

É óbvio que as técnicas de simplificação não se restringem às expostas acima. Num dos itens posteriores, analisaremos diversas expressões e as relações lógicas empregadas nas simplificações.

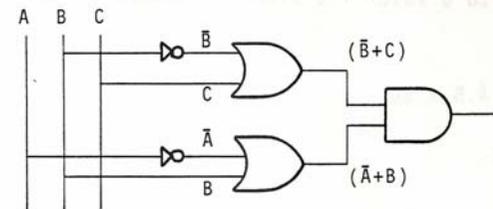
### 3.4 - REALIZAÇÃO DE CIRCUITOS

As expressões simplificadas conseguidas no item 3.3 podem ser realizadas com portas lógicas como mostram os esquemas simbólicos abaixo:

1)  $L = \bar{A}\bar{B} + BC$



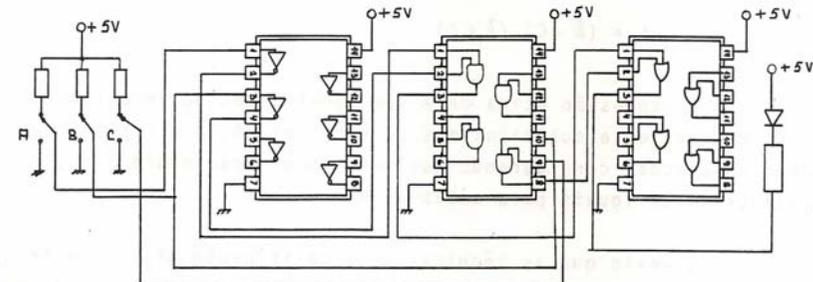
2)  $L = (\bar{B} + C)(\bar{A} + B)$



Os circuitos acima são equivalentes.

Com os circuitos integrados 7404, 7408 e 7432, podemos realizar praticamente os testes de ambas as equações. Conectando as entradas A, B, C, as chaves que possibilitem fazê-las iguais a 0 ou 1 (5V), um LED colocado à saída L acenderia quando a combinação de entrada fizesse corresponder, de acordo com a tabela da verdade,  $L = 0$ .

Para a expressão  $L = \bar{A}\bar{B} + BC$ , o circuito prático deveria ser assim montado:



## 3.5 - PROBLEMAS RESOLVIDOS

Neste item são apresentados as resoluções de uma série de equações lógicas, desde a tabela da verdade até o diagrama esquemático, em grau crescente de dificuldade. Ao lado dos desenvolvimentos, indicam-se as propriedades ou relações lógicas empregadas para tal, na sua forma simples. O estudo de cada um dos problemas resolvidos, dando especial atenção a cada passo, é essencial para que se possa, seguramente, adquirir prática em simplificações de equações lógicas.

1) Elaborar o circuito correspondente à tabela:

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Escrevendo a expressão na forma de soma de produtos:

$$L = \bar{A}.\bar{B}.C + \bar{A}.B.C + A.\bar{B}.\bar{C} + A.\bar{B}.C$$

Associando os termos: 1º e 2º; 3º e 4º

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

Utilizando-se a relação lógica,  $A + \bar{A} = 1$  fica:

$$L = \bar{A}C.1 + A\bar{B}.1$$

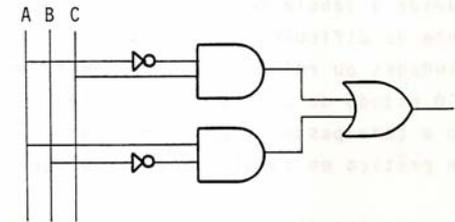
Sendo o 1 elemento neutro no produto, tal que:

$$A.1 = A ; \text{tem-se}$$

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

que é a forma final da equação.

O circuito correspondente é o apresentado a seguir:



2) Esquematizar o circuito correspondente a:

A	B	C	L
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A expressão escrita como soma de produtos resulta:

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

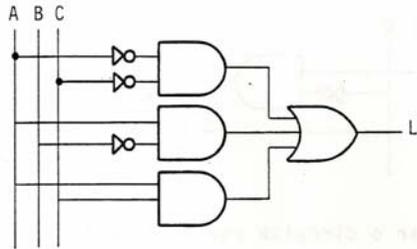
As associações se fazem com os termos: 1º e 2º; 3º e 4º; e 4º e 5º. Observe-se que, embora o 4º termo já fosse associado ao 3º, assim mesmo participa de outra associação pois permite maior simplificação da equação. Essa dupla participação é possível baseada na relação que permite escrever  $A = A + A$ , isto é, o acréscimo de um termo igual, quando somado não altera a expressão. Ora, se repetíssemos o 4º termo na expressão, esta não se alteraria e o termo considerado participaria de duas associações, como, aliás, o fez:

$$L = \bar{A}\bar{C}(\bar{B} + B) + A\bar{B}(\bar{C} + C) + AC(\bar{B} + B)$$

Considerando que  $A + \bar{A} = 1$ , obtemos a equação final:

$$L = \bar{A}\bar{C} + A\bar{B} + AC$$

O circuito simbólico toma o seguinte aspecto:



3) Elaborar o circuito equivalente a:

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Sob a forma de soma de produtos obtemos:

$$L = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

Associando-se os termos: 1º e 2º; 3º e 4º e 4º e 5º, vem:

$$L = \bar{A}B(\bar{C} + C) + A\bar{C}(\bar{B} + B) + AB(\bar{C} + C)$$

que se reduz inicialmente a:

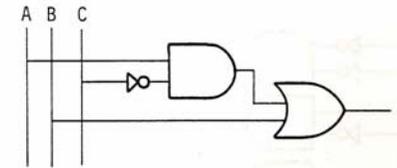
$$L = \bar{A}B + A\bar{C} + AB$$

Neste ponto observamos que a associação do 1º e do 3º termos é possibilitada pela evidenciação da variável B, produzindo o valor simplificável  $A + \bar{A}$ . Temos pois:

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

que se reduz, enfim, a:  $L = B + A\bar{C}$

O circuito com portas lógicas é dado abaixo



4) Elaborar o circuito equivalente a:

A	B	C	L
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Partindo da equação:

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

Chegamos a:

$$L = \bar{A}\bar{C}(\bar{B} + B) + A\bar{B}(\bar{C} + C) + AB(\bar{C} + C)$$

pela associação de termos: 1º e 2º; 3º e 4º e 5º e 6º.

Procedendo-se à simplificação dos termos entre parênteses:

$$L = \bar{A}\bar{C} + A\bar{B} + AB$$

Observa-se que os dois últimos termos da soma podem ser associados:

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

ou  $L = \bar{A}\bar{C} + A$

A expressão acima pode ser ainda simplificada se aplicarmos diretamente a relação já estudada:  $(A + \bar{A}B = A + B)$  ou se preferirmos, aplicando a propriedade distributiva do produto:

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

$$L = (A + \bar{A})(A + \bar{C})$$

Decorre então:

$$L = A + \bar{C}$$

Apenas uma porta lógica é suficiente para realizar a função;



5) Determinar a equação simplificada correspondente à tabela do problema anterior pelo método do produto de somas.

O elevado número de 1's que apareciam na coluna L da tabela anterior, já sugeriam a utilização do método do produto de somas.

Empregue-mo-lo então:

A equação inicial toma a forma:

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

Evidenciando-se o termo  $A + \bar{C}$ , tudo se torna muito fácil:

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

O segundo parênteses subentende uma simplificação da forma  $(A \cdot \bar{A}) = 0$ , resultando:

$$L = (A + \bar{C}) + 0$$

que, segundo  $A + 0 = A$ , toma a expressão final idêntica à do item 4.

$$L = A + \bar{C}$$

6) Elaborar um circuito capaz de detetar combinações ABC lidadas como números binários, que sejam maiores ou iguais a três.

A tabela neste caso deve ser elaborada em função do enunciado do problema. Fazemos  $L = 1$  para as combinações maiores ou iguais a 011 (3).

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A equação, pela soma de produtos, toma a forma:

$$L = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

Procedendo-se à simplificação:

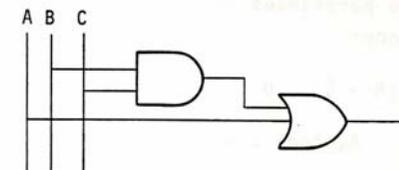
$$L = BC(\bar{A} + A) + A\bar{B}(\bar{C} + C) + AB(\bar{C} + C)$$

$$L = BC + A\bar{B} + AB$$

$$L = BC + A(\bar{B} + B)$$

$$L = BC + A$$

Portanto; o circuito abaixo é o nosso detetor de grandezas binárias maiores ou iguais a três:



7) Em um teste, a questão A tem peso 4,; a questão B, peso 3 e a questão C, peso 3. Elaborar um circuito que indique se o aluno atingiu ou não o objetivo sendo o rendimento m̄nimo aceit̄avel de 60%.

Construimos a tabela da verdade:

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

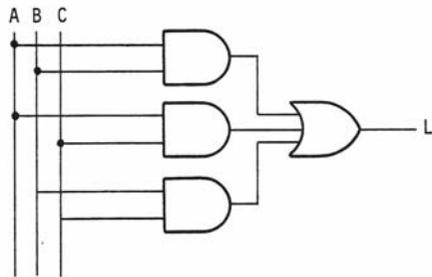
A, B ou C iguais a 1 significa que a quest̄o foi feita cor<sub>re</sub>tamente. L igual a 1 indica objetivo atingido.

$$L = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$L = BC(\bar{A} + A) + AC(\bar{B} + B) + AB(\bar{C} + C)$$

$$L = BC + AC + AB$$

Portanto, o circuito indicado para tal  $\bar{e}$ :



### 3.6 - PROBLEMAS PROPOSTOS

1) Esquematizar os circuitos l̄gicos correspondentes  $\bar{a}$ s e<sub>l</sub> qua<sub>ç</sub>ões: de  $L_1$  a  $L_8$ .

A	B	C	$L_1$	$L_2$	$L_3$	$L_4$	$L_5$	$L_6$	$L_7$	$L_8$
0	0	0	1	0	1	0	1	1	0	0
0	0	1	1	1	1	1	0	1	1	0
0	1	0	0	0	1	0	1	1	0	1
0	1	1	0	1	1	1	0	0	1	1
1	0	0	1	0	0	0	0	1	1	1
1	0	1	1	0	0	1	0	1	0	1
1	1	0	0	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	0	0	1

2) Esquematizar um circuito l̄gico em que  $L = 1$  para combi<sub>na</sub>ões contendo 2 ou 3 bits iguais a 1.

3) Elaborar o circuito em que  $L = 1$  para n̄meros ABC meno<sub>res</sub> ou iguais a 5 (101).

4) Repetir os problemas dos testes, supondo os pesos de A, B e C respectivamente: 5, 3 e 2 e supondo o rendimento m̄nimo de 50%.

5) Elaborar um circuito l̄gico em que  $L = 1$  para as combi<sub>na</sub>ões onde o n̄mero de bits 1 seja  $\bar{i}$ mpar.

#### 4 - TEOREMAS DE DE MORGAN

- 4.1 - Introdução
- 4.2 - Teoremas - Demonstrações
- 4.3 - Teoria dos Conjuntos
- 4.4 - Equacionamento de TV
- 4.5 - Expressões com Negação
- 4.6 - Problemas Resolvidos
- 4.7 - Problemas Propostos

##### 4.1 - INTRODUÇÃO

Muitas vezes nós nos deparamos com expressões aparentemente sem simplificação imediata, ou mesmo sem possibilidades de simplificação se não fizermos uso dos teoremas de De Morgan.

Outras vezes, necessitamos transformar uma função em outra, para aproveitar da melhor maneira os circuitos integrados de que dispomos. Pois bem, tanto em um como no outro caso, os teoremas de De Morgan podem ser utilizados com vantagens perceptíveis.

A propriedade distributiva da soma  $(A + BC) = (A + B).(A + C)$  vista em capítulo anterior foi aceita sem demonstração rigorosa, isto é, sem demonstração algébrica. A aplicação dos teoremas de De Morgan permite que a propriedade seja demonstrada de tal forma. Assim, indiretamente, esses teoremas permitem uma série de simplificações que utilizam a propriedade acima.

Por outro lado, muitas vezes é vantagem realizarmos um circuito combinacional com apenas um tipo de porta ou função. Isso se torna evidente quando consultamos uma tabela de preços qualquer. Os teoremas de De Morgan permitem a transformação de um tipo de função em outro, cujas portas lógicas sejam economicamente mais viáveis.

Além destas, os teoremas de De Morgan permitem uma série de simplificações e seu perfeito domínio torna-se, portanto, indispensável.

Neste capítulo apresentamos aplicações referentes ao primeiro assunto sendo o segundo deixado para o capítulo posterior.

Além disso, apresentaremos a teoria dos conjuntos como forma geral da Álgebra Booleana, fazendo analogias entre as operações lógicas e operações envolvendo conjuntos. Esse estudo servirá para a demonstração dos teoremas de De Morgan, assim como para qualquer outra das relações lógicas já estudadas.

##### 4.2 - TEOREMAS DE DE MORGAN

São dois os teoremas de De Morgan.

O primeiro transforma uma soma em produto e o outro faz o inverso. Matematicamente eles são assim enunciados:

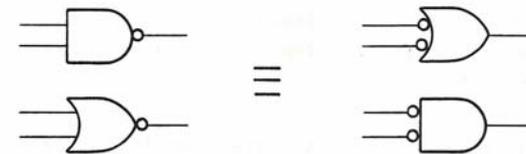
1) "A soma negada de N variáveis é igual ao produto das mesmas variáveis individualmente negadas".

$$\overline{A + B + \dots + N} = \bar{A} \cdot \bar{B} \dots \bar{N}$$

2) "O produto negado de N variáveis, é igual à soma das mesmas individualmente negadas".

$$\overline{A \cdot B \dots N} = \bar{A} + \bar{B} + \dots \bar{N}$$

Esses teoremas admitem, então, as equivalências entre os circuitos abaixo ilustrados, sendo possível a substituição de um pelo outro.



É importante notar que  $\bar{A} \cdot \bar{B}$  é diferente de  $\overline{A \cdot B}$ , o que se prova facilmente substituindo as variáveis A e B por 0 e 1, respectivamente. O mesmo se diz a respeito de  $\bar{A} + \bar{B}$  e  $\overline{A + B}$ . É freqüente a ocorrência de tais abusos na resolução de expressões lógicas.

O primeiro teorema será demonstrado pelo desenvolvimento das tabelas da verdade das expressões dos dois membros. Comparando os resultados finais comprova-se o teorema. Com intuito de simplificar a demonstração, vamos considerar o envolvimento de apenas duas

variáveis: A e B.

Desenvolvendo o primeiro membro:

A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Quanto ao segundo:

A	B	$\overline{A}$	$\overline{B}$	$\overline{A \cdot B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

As últimas colunas, como se verifica, são iguais.

A demonstração anterior não possui um rigor matemático satisfatório. A dedução algébrica é melhor aceita, motivo pelo qual a empregamos na demonstração do segundo teorema. Novamente aqui, consideramos apenas duas variáveis.

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

Para que a expressão acima se verifique, devemos ter:

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

o que implica nas duas condições seguintes:

- a)  $(\overline{A} + \overline{B}) \cdot A \cdot B = 0$  e  
 b)  $(\overline{A} + \overline{B}) + A \cdot B = 1$

Se provarmos a validade destas últimas, o teorema fica demonstrado.

A relação (a) é desenvolvida como segue:

$$(\overline{A} + \overline{B}) \cdot A \cdot B = \overline{A} \cdot A \cdot B + \overline{B} \cdot A \cdot B = 0 \cdot B + 0 \cdot A = 0 + 0 = 0$$

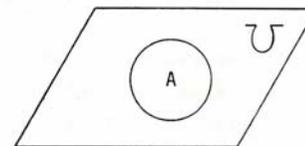
A relação (b) se verifica de forma análoga.

$$\begin{aligned} (\overline{A} + \overline{B}) + A \cdot B &= \overline{A} + (\overline{B} + AB) = \overline{A} + [(\overline{B} + A) (\overline{B} + B)] = \overline{A} + [(\overline{B} + A) \cdot 1] \\ &= \overline{A} + (\overline{B} + A) = \overline{B} + (A + \overline{A}) = \overline{B} + 1 = 1 \end{aligned}$$

#### 4.3 - TEORIA DOS CONJUNTOS

Um outro tipo de demonstração para os teoremas de De Morgan, aliás, para todas as relações já estudadas, reside na aplicação da teoria dos conjuntos em analogia à Álgebra Booleana. Isso é permitido visto que esta álgebra decorre da teoria dos conjuntos.

Inicialmente suponhamos a existência do conjunto universo e de um subconjunto A pertencente ao primeiro, tal como nos mostra o diagrama abaixo:



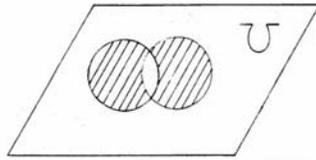
Os formalismos da álgebra booleana e da teoria dos conjuntos permite analogar:

- a) a variável booleana A com o subconjunto A.
- b) O subconjunto complementar de A, notado  $C_A$  (espaço dentro do paralelogramo porém fora do círculo A) como o complemento  $\overline{A}$  da variável A.
- c) O conjunto universo U com o bit 1.
- d) O conjunto vazio com o bit 0.

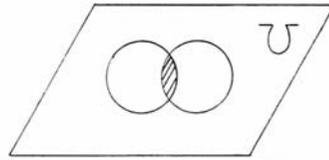
As operações realizáveis entre dois conjuntos, conhecidos por "união" e "intersecção" encontram analogia com as operações  $\cup$  e  $\cap$  da álgebra booleana.

No diagrama a seguir representa-se hachurado o conjunto AUB

à esquerda e  $A \cap B$  à direita, que correspondem aos resultados das operações  $A+B$  e  $A \cdot B$ .



$$A \cup B = A + B$$

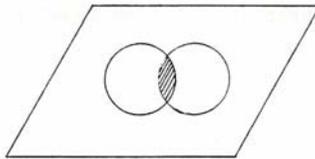


$$A \cap B = A \cdot B$$

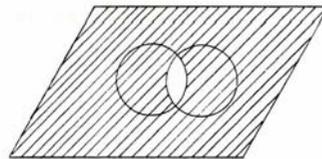
Os teoremas de De Morgan encontram demonstração nessa teoria pela comparação entre os conjuntos resultantes das operações indicadas nos dois membros das expressões. Em primeiro lugar vamos provar que:

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

O termo  $\overline{A \cdot B}$  é representado no diagrama por  $C(A \cap B)$ , como aparece abaixo:

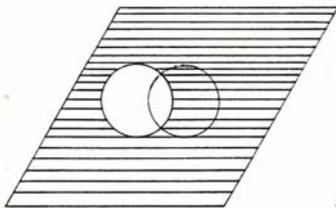


$$A \cap B$$

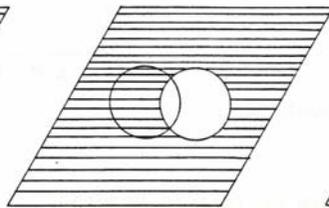


$$C(A \cap B)$$

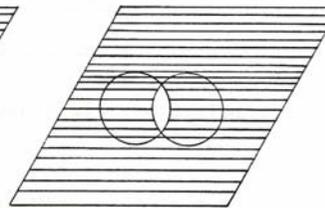
O conjunto resultante da operação  $C(A) \cup C(B)$  corresponde ao segundo membro  $\overline{A} + \overline{B}$  e é idêntico ao obtido acima.



$$C(A)$$

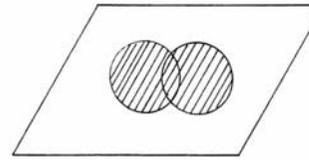


$$C(B)$$

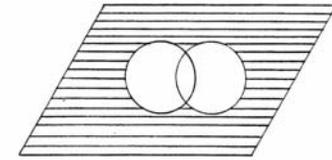


$$C(A) \cup C(B)$$

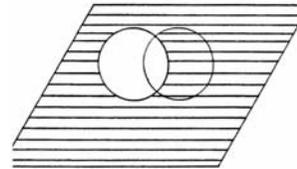
O segundo teorema  $\overline{A+B} = \overline{A} \cdot \overline{B}$  é demonstrado a seguir, ao se comparar os conjuntos resultantes das duas operações indicadas:  $C(A \cup B)$  e  $C(A) \cap C(B)$ .



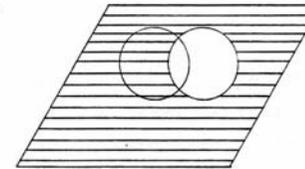
$$A \cup B$$



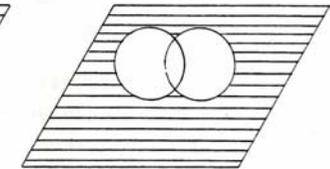
$$C(A \cup B)$$



$$C(A)$$



$$C(B)$$



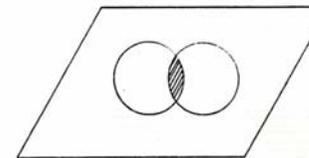
$$C(A) \cap C(B)$$

Como foram representados e demonstrados os teoremas de De Morgan, também outras relações lógicas podem ser comprovadas. Seja, por exemplo, provar que:  $A + AB = A$ , relação já conhecida, pelo método exposto.

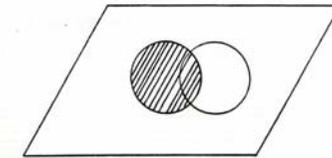
Transformando para notações próprias,  $A + AB$  tomaria a forma:

$$A \cup (A \cap B)$$

que teria como conjunto resposta:



$$A \cap B$$



$$A \cup (A \cap B)$$

ou seja, o próprio conjunto A.

#### 4.4 - EQUACIONAMENTO DE TABELAS VERDADE

Uma opção diferente para a resolução do problema que envolve o equacionamento de uma tabela da verdade e a realização do circuito correspondente é obter  $\bar{L}$  ao invés de L. Se houver necessidade, podemos expressar L como soma de produtos fazendo uso dos teoremas

de De Morgan. Vejamos o exemplo seguinte:

A	B	C	L
0	0	0	0
0	0	1	0
0	0	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A equação correspondente, aplicando o novo método resulta:

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC$$

A simplificação é imediata e nos dá:

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

Para se explicitar L, basta negarmos uma vez ambos os membros da expressão:

$$L = \overline{\bar{A}\bar{B} + \bar{A}C}$$

E, se quisermos obter o resultado sob forma de soma de produtos, aplicamos De Morgan para eliminar o sinal de negação:

$$L = \overline{\bar{A}\bar{B}} \cdot \overline{\bar{A}C}$$

Como vimos, a soma indicada foi transformada em produto e as parcelas negadas individualmente. Faz-se necessária a nova aplicação de De Morgan nos dois fatores restantes, independentemente:

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

A expressão acima é finalmente desenvolvida, encontrando-se então a forma final simplificada da equação.

$$L = A.A + A.\bar{C} + A.B + B.\bar{C}$$

$$\begin{aligned} L &= A + A.\bar{C} + A.B + B\bar{C} \\ L &= A.(1 + \bar{C} + B) + B\bar{C} \\ L &= A.1 + B\bar{C} \\ L &= A + B\bar{C} \end{aligned}$$

Então pode parecer um método muito demorado, à primeira vista, mas, dependendo das circunstâncias, a sua aplicação nos leva ao resultado muito rapidamente.

#### 4.5 - EXPRESSÕES COM NEGAÇÃO

Quando a tarefa é simplificar uma expressão que envolva negações de operação, a saída mais rápida é a aplicação dos teoremas de De Morgan.

Sempre que formos aplicá-los, devemos fazê-lo com segurança, evitando as interpretações falsas. Devemos ter em mente qual é a operação e qual o sinal de negação que serão modificados. Podemos começar pelos sinais de ordem superior, aplicando os teoremas enquanto ocorrerem negações em operação. Nunca devemos esquecer dos parênteses onde forem imprescindíveis.

O exemplo comentado elucidado.

$$L = \overline{\bar{A}\bar{B}\bar{C}} + \bar{A}\bar{B}C + A\bar{B}\bar{C}$$

Transformamos a soma sob o maior sinal de negação:

$$L = \overline{\bar{A}\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{B}C} + A\bar{B}\bar{C}$$

Eliminamos as duas negações sob o primeiro termo, pois elas se cancelam.

$$L = ABC \cdot \overline{\bar{A}\bar{B}C} + A\bar{B}\bar{C}$$

Aplicamos De Morgan no termo do meio, transformando o produto  $\overline{\bar{A}\bar{B}C}$  por  $(\bar{A} + \bar{B} + C)$  em soma. Note o aparecimento dos Parênteses para conservar a ordem de execução das operações.

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

Eliminamos a dupla negação sobre AB

$$L = ABC \cdot (AB + \bar{C}) + A\bar{B}\bar{C}$$

Aplicamos novamente um dos teoremas, agora no último termo:

$$L = ABC \cdot (AB + \bar{C}) + A(\bar{B} + \bar{C})$$

A simplificação decorre de maneira simples:

$$L = ABC \cdot AB + AB\bar{C}C + A\bar{B} + A\bar{C}$$

$$L = ABC + 0 + A\bar{B} + A\bar{C}$$

$$L = A(BC + \bar{B} + \bar{C})$$

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

$$L = A \cdot 1$$

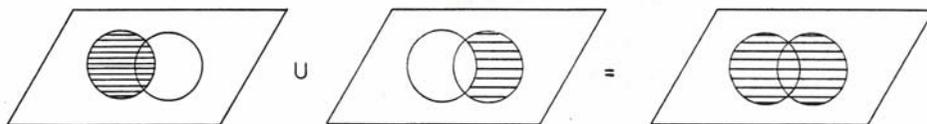
$$L = A$$

#### 4.6 - PROBLEMAS RESOLVIDOS

4.6.1 - Demonstrar, usando a teoria dos conjuntos, a relação lógica:

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

Solução: A representação do primeiro membro resulta:



que é exatamente o conjunto representativo de  $A \cup B$ , ou seja,  $A + B$ .

4.6.2 - Determinar L sob a forma de soma de produtos.

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Solução:

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

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

$$L = \overline{\bar{B}\bar{C} + A\bar{C}}$$

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

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

$$L = \bar{A}B + BC + \bar{A}C + C$$

$$L = \bar{A}B + C(B + \bar{A} + 1)$$

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

4.6.3 - Dada a função:

$$L = \overline{A + B} + \bar{A}BC + \overline{A(B + C)}$$

- Obter a expressão simplificada.
- Representar a tabela da verdade da mesma.

Solução:

$$a) L = \bar{A}\bar{B} + \bar{A}BC + \bar{A} + \overline{(B + C)}$$

$$L = \bar{A}\bar{B} + \bar{A}BC + \bar{A} + \bar{B}\bar{C}$$

$$L = \bar{A}(\bar{B} + BC + 1) + \bar{B}\bar{C}$$

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

b) A tabela se obtém substituindo as 8 combinações de valores na expressão obtida.

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

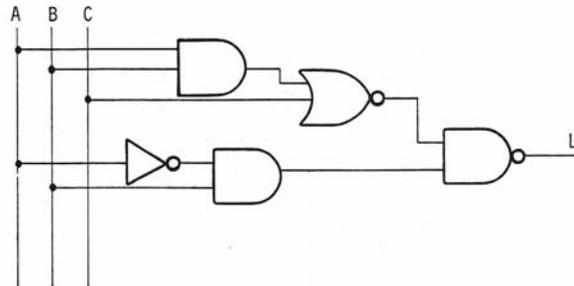
## 4.6.4 - Simplificar as expressões:

a)  $\overline{A + \overline{AB}} = \overline{A} \cdot AB = 0$

b)  $\overline{AB} + \overline{AB} = \overline{A} + \overline{B} + \overline{AB} = \overline{A} + \overline{B}$

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

## 4.6.5 - Simplificar o circuito:



Em primeiro lugar retiramos a equação do circuito:

$$L = \overline{AB + C} \cdot \overline{AB}$$

$$L = (\overline{AB} + \overline{C}) \cdot \overline{AB}$$

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

$$L = \overline{AB} + \overline{C} + \overline{A} + \overline{B}$$

$$L = \overline{A}(\overline{B} + 1) + \overline{C} + \overline{B}$$

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

$$L = \overline{A} \cdot \overline{C} \cdot \overline{B}$$

4.6.6 - Provar que:  $\overline{AB} + C = (\overline{A} + C)(\overline{B} + C)$ 

Solução: Aplicando-se De Morgan no primeiro termo chega-se ao segundo:

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

## 4.7 - PROBLEMAS PROPOSTOS

## 4.7.1 - Provar pela representação gráfica de conjuntos, que:

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

4.7.2 - Determinar  $L_1$ ,  $L_2$  e  $L_3$  pelo método exposto, dando o resultado sob forma de soma de produtos:

A	B	C	$L_1$	$L_2$	$L_3$
0	0	0	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

## 4.7.4 - Simplificar:

a)  $L = A + \overline{A} + B$

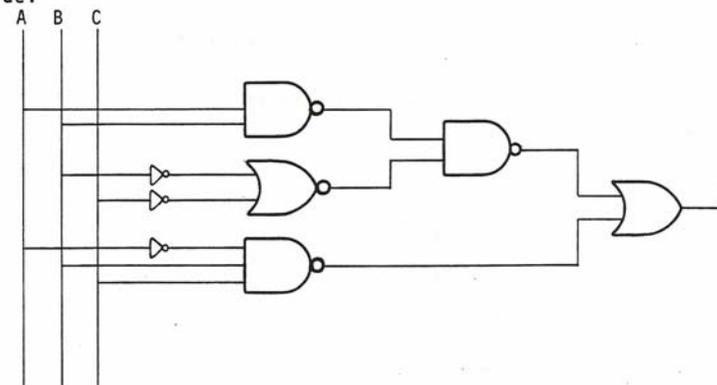
b)  $L = \overline{ABC} + \overline{ABC}$

c)  $L = \overline{\overline{ABC}} + \overline{\overline{ABC}} + \overline{\overline{ABC}}$

d)  $L = \overline{\overline{ABC}} + \overline{\overline{ABC}} + \overline{\overline{ABC}}$

## 4.7.5 - Simplificar o circuito e elaborar a tabela da ver-

dade:



## 5 - REALIZAÇÃO DE CIRCUITOS UTILIZANDO PORTAS NE ou NOU

- 5.1 - INTRODUÇÃO
- 5.2 - A PORTA NE
- 5.3 - REALIZAÇÃO COM PORTAS NE
- 5.4 - REALIZAÇÃO COM PORTAS NOU
- 5.5 - EXERCÍCIOS RESOLVIDOS
- 5.6 - EXERCÍCIOS PROPOSTOS

### 5.1 - INTRODUÇÃO

Os Teoremas de De Morgan, como mencionado anteriormente, possibilitam a transformação de uma função lógica e, se usadas convenientemente, essa transformação implica em um circuito mais econômico em termos financeiros.

É o caso, por exemplo, de se modificar uma função de forma tal que ela se torne realizável com a utilização de portas NE, apenas. Dessa forma o circuito final pode custar menos, já que os CI's constituídos por portas NE são geralmente mais baratos.

Além do mais, a realização de uma função composta de operações diversas implica em gastos desnecessários devido às "sobras" de CI's que ocorrem. Essas sobras aparecem porque o fabricante não fornece as portas individualmente e sim em conjunto, agrupados num mesmo CI. Um exemplo é o 7408 que se compõe de 4 portas E de 2 entradas cada. Assim, se apenas 2 portas E são necessárias, como no caso da expressão  $L = A\bar{B} + B\bar{C}$ , metade do CI ficará sobrando, o mesmo ocorrendo com aquele que permite a operação OU, sobrando um pouco mais do CI 7404. (6 portas NÃO).

Essa é a conveniência da utilização de portas NE na realização de funções lógicas. Neste capítulo, exporemos o método que permite tais transformações, complementando com o estudo da realização de circuitos mediante o uso exclusivo de portas NOU.

### 5.2 - A PORTA NE

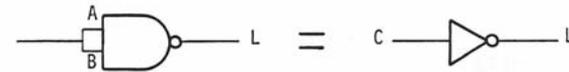
É a mais versátil, pois permite realizar qualquer outra operação básica, o que lhe deu o privilégio de ser fabricada em maior

quantidade e em menor preço.

#### 5.2.1 - Operação NÃO

A porta NE transforma-se em porta inversora (NÃO) mediante dois dos processos seguintes.

1º Curto-circuitando-se as entradas.



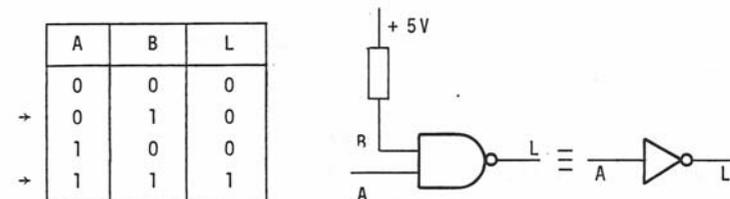
A tabela da verdade do circuito reduz-se a:

A	B	L
0	0	1
1	1	0

Como  $A = B = C$ , resulta a tabela característica da operação NÃO:  $C = \bar{L}$

C	L
0	1
1	0

2º Forçando o nível lógico 1 em todas as entradas não utilizadas.



Por exemplo, fazendo  $B = 1$ , apenas as condições assinaladas serão aceitas, reduzindo a tabela para

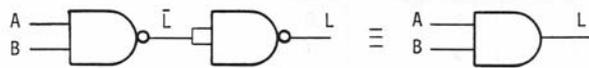
A	L
0	1
1	0

## 5.2.2 - Operação E

Por definição, a operação NE surge da negação do resultado da operação E. Se novamente negada, a operação NE volta à sua forma original, ou seja, à operação E:

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

Então, basta a colocação de um inversor na saída de uma porta NE para convertê-la em porta E:



Obviamente, o inversor também é constituído por uma porta NE.

## 5.2.3 - Operação OU

Aqui, faz-se necessária a aplicação de um dos teoremas de De Morgan para se obter o resultado desejado.

Seja a operação OU entre as variáveis A e B.

$$L = A + B$$

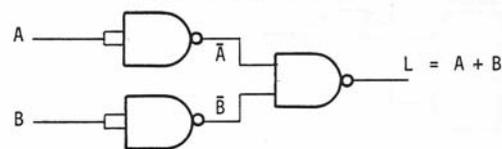
Negando-se duplamente a operação, ela não se altera:

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

Agora, conservando-se a negação superior, aplicamos um dos teoremas na expressão  $\overline{\overline{A + B}}$ :

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

Esta função é então realizada na forma NE:



Evidentemente, não existe vantagem alguma em se realizar somente uma operação no modo NE, como nos exemplos acima. Todas as portas resultantes sairiam mais caras. Torna-se vantagem sua utilização quando uma função envolvendo diversas operações é realizada, como nos exemplos dados a seguir.

## 5.3 - REALIZAÇÃO COM PORTAS NE

Seja a tabela da verdade:

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

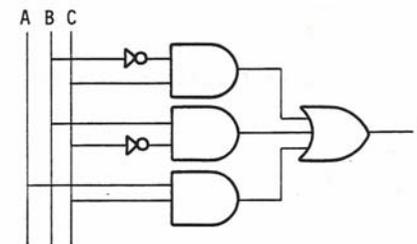
Pede-se determinar a equação simplificada e a montagem do circuito, primeiro, da forma conhecida, depois, através de portas NE.

A equação do circuito com sua devida simplificação resulta em:

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

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

O circuito simbólico da forma como nós concebemos, pode ser realizado assim:



Vamos agora transformar a expressão à forma NE.

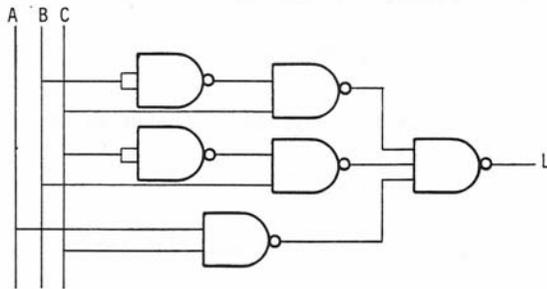
1º) Nega-se duplamente a expressão, o que não lhe altera o resultado:

$$L = \overline{\overline{BC} + \overline{B\overline{C}} + AC}$$

2º) Aplica-se "De Morgan", conservando o sinal superior de negação:

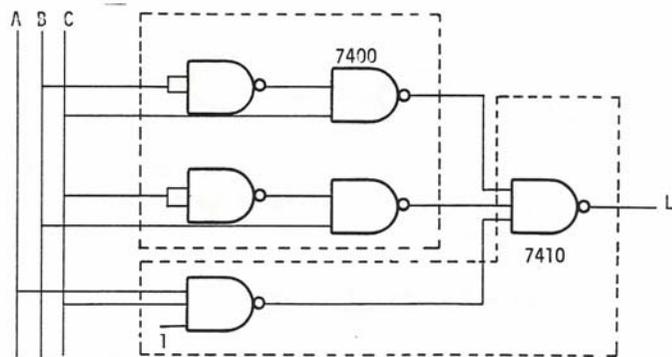
$$L = \overline{\overline{BC} \cdot \overline{B\overline{C}} \cdot AC}$$

Esta expressão pode então ser realizada com portas NE:



O circuito acima é realizado com apenas 2 CI's "NE": 7400 e 7410. O primeiro, como já foi dito, possui 4 portas NE de 2 entradas. O segundo, apresenta 3 portas NE de 3 entradas.

Faz-se necessário, aqui, converter 2 portas NE de duas entradas em duas inversões e uma porta NE de 3 entradas em outra de apenas duas. A figura abaixo demonstra o que se afirmou:



O terminal alto em um dos terminais de entrada de uma porta NE, inutiliza-o, por assim dizer.

Nota-se que o preço do circuito final diminui consideravelmente.

Outro fato importante foi que a única sobra ocorrida constituiu-se de uma porta NE de 3 entradas, bem menos significativa que a sobra anterior.

As vezes pode ocorrer o seguinte problema: Dispondo-se de portas NE de 2 entradas, precisamos de outra do mesmo tipo porém com 4 entradas. O que fazer então?

Escrevemos a função de 4 entradas, ou variáveis.

$$L = \overline{A \cdot B \cdot C \cdot D}$$

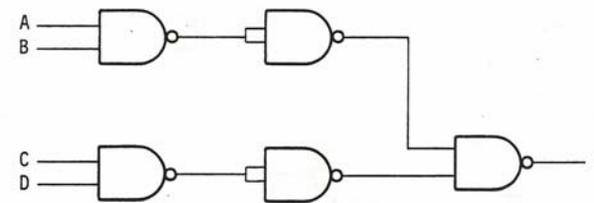
A explicação de um dos Teoremas de Morgan leva a:

$$L = \overline{A \cdot B} + \overline{C \cdot D}$$

E, aplicando-se mais uma vez, obtemos:

$$L = \overline{\overline{\overline{A \cdot B}} \cdot \overline{\overline{C \cdot D}}}$$

que pode ser realizada assim:



Cada caso particular requer uma solução diferente, porém, como ficou claro, há sempre aplicação de "De Morgan" que resolvem as diferentes situações.

#### 5.4 - REALIZAÇÃO COM PORTAS NOU

De maneira análoga, as portas NOU podem ser utilizadas na implementação de circuitos.

Seja a tabela:

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

O método do produto de somas nos dá:

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

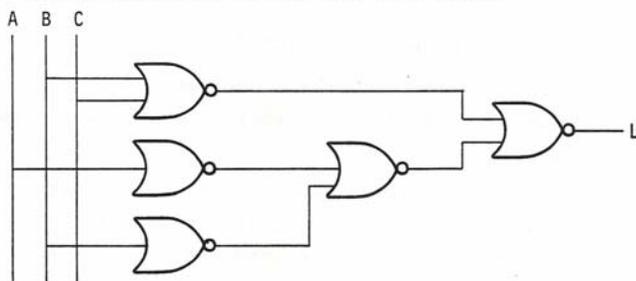
$$L = (B + C) \cdot (\bar{A} + \bar{B})$$

A redução para a forma NOU é análoga ao caso da operação NE:

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

$$L = \overline{(B + C)} + \overline{(\bar{A} + \bar{B})}$$

Simbolicamente, o circuito fica assim:



Se a obtenção de L se desse por meio do método da soma de produtos, teríamos então:

$$L = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$

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

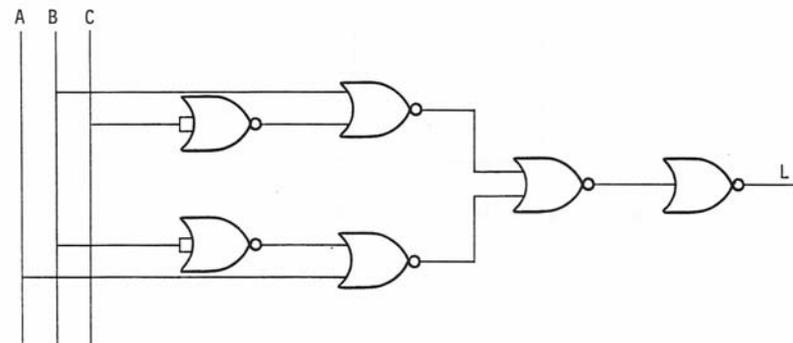
Transformando à forma NOU:

$$L = \overline{\overline{\bar{B}C} + \overline{\bar{A}B}}$$

$$L = \overline{\bar{B}C} + \overline{\bar{A}B}$$

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

Obtemos o circuito:



que utiliza uma porta a mais.

O método de produto de somas seria preferido neste caso.

Como norma geral, sempre que for preciso realizar funções com portas NE (NOU) devemos aplicar as seguintes regras:

- Nega-se duplamente a função
- Se a operação externa é uma soma (produto) aplica-se De Morgan à uma das inversões e se é produto (soma) não se opera.
- Se no interior da expressão existem somas (produtos) aplica-se a cada uma das operações duas inversões e se opera uma delas convertendo em inverso de produto (soma).
- Continua-se assim até que todas as somas (produtos) se convertam em produtos (somadas).

## 5.5 - EXERCÍCIOS RESOLVIDOS

5.5.1 - Realizar o circuito correspondente à tabela apresentada com portas NE.

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Solução:

$$L = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$

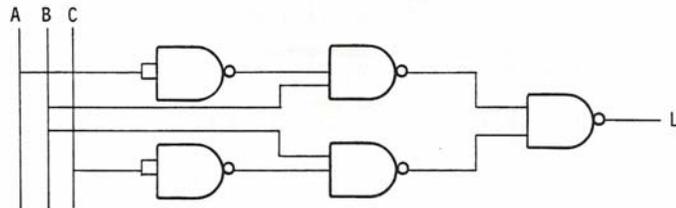
$$L = \bar{A}B(\bar{C} + C) + B\bar{C}(\bar{A} + A)$$

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

$$L = \overline{\bar{A}B + B\bar{C}}$$

$$L = \overline{\bar{A}B} \cdot \overline{B\bar{C}}$$

O circuito simbólico é o seguinte:



5.5.2 - Realizar a operação NOU com portas NE.

Solução:

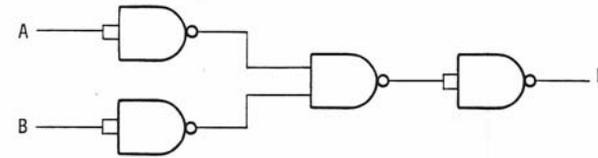
A operação NOU é definida:

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

Aplicando-se "De Morgan"

$$L = \bar{A} \cdot \bar{B} = \overline{\bar{A} \cdot \bar{B}}$$

Facilmente realizável.



5.5.3 - Realizar com portas NOU:

A	B	C	L
0	0	0	0
0	0	1	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Solução:

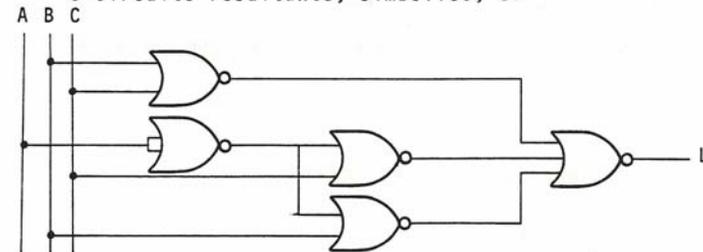
$$L = (A+B+C) (\bar{A}+B+C) (\bar{A}+B+\bar{C}) (\bar{A}+\bar{B}+C)$$

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

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

$$L = \overline{(B+\bar{C}) + (\bar{A}+\bar{C}) + (\bar{A}+B)}$$

O circuito resultante, simbólico, é:



5.5.4 - Realizar a operação NOU de 3 variáveis com portas NOU de 2 entradas.

Solução:

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

A aplicação sucessiva de De Morgan conduz a:

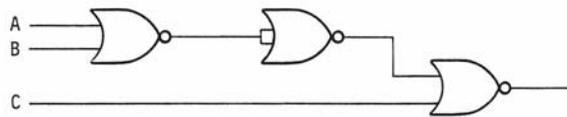
$$L = \bar{A} \cdot \bar{B} \cdot \bar{C}$$

$$L = \overline{A + C} \cdot \bar{C}$$

$$L = \overline{\overline{A + C} + C}$$

que também seria obtida pela dupla inversão diretamente sobre a operação  $A + B$ .

O circuito final:



## 5.6 - EXERCÍCIOS PROPOSTOS

5.6.1 - Realizar  $L_1$ ,  $L_2$  e  $L_3$  com portas NE e NOU. Esquematizar os circuitos simbólicos e práticos.

A	B	C	$L_1$	$L_2$	$L_3$
0	0	0	1	1	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	0	1	1

5.6.2 - Obter  $A \oplus B$  com portas NE.

5.6.3 - O que se deve fazer para "inutilizar" um terminal de uma porta NOU de 3 entradas para reduzi-la a uma NOU de 2 entradas?

5.6.4 - Reduza as funções E, OU, NÃO e NE mediante o uso

de portas NOU.

5.6.4 - Obter uma função NE de 5 entradas mediante portas NE de 2 entradas.

## 6 - DIAGRAMA DE VEITCH - KARNAUGH

- 6.1 - Introdução
- 6.2 - Diagrama para 2 variáveis
- 6.3 - Diagrama para 3 variáveis
- 6.4 - Diagrama para 4 variáveis
- 6.5 - Diagrama para 5 variáveis
- 6.6 - Exercícios Resolvidos
- 6.7 - Exercícios Propostos

### 6.1 - INTRODUÇÃO

O método algébrico para simplificações de equações lógicas, como foi visto até aqui, além de ser trabalhoso quando o aplicamos em equações de mais de 4 variáveis, às vezes não fornece a forma mínima simplificada, como o exemplo adiante.

Seja a tabela da verdade apresentada a seguir:

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A equação obtida pela soma de produtos é:

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

que pode ser simplificada da seguinte forma:

$$L = \bar{A}\bar{B}(\bar{C} + C) + \bar{B}C(\bar{A} + A) + B\bar{C}(\bar{A} + A) + AB(\bar{C} + C)$$

$$L = \bar{A}\bar{B} + \bar{B}C + B\bar{C} + AB$$

O resultado final não mais pode ser reduzido, porém, não é

essa a forma mais "econômica" para a função L.

Se a simplificação tomar outro rumo, obteremos então:

$$L = \bar{A}\bar{B}(\bar{C} + C) + B\bar{C}(\bar{A} + A) + AC(\bar{B} + B)$$

$$L = \bar{A}\bar{B} + B\bar{C} + AC$$

Nota-se então, que L, escrito nesta última forma, pode ser realizado com menor número de portas sendo a equação bem mais vantajosa.

Com o exemplo acima notam-se as falhas que podem ocorrer pela simples aplicação do método, sem malícia.

A única forma de se saber quando é que a equação está simplificada ao máximo é através de uma visualização gráfica do processo de simplificação. É o Diagrama de Veitch - Karnaugh, que permite tal visualização.

### 6.2 - DIAGRAMA PARA 2 VARIÁVEIS

O diagrama de Veitch para 2 variáveis se compõe de 4 quadros dispostos a formar um quadrado, e correspondentes, cada um, a uma combinação de estados das variáveis.

Sabe-se que 2 variáveis A e B admitem 4 combinações de estados, ou seja:

A	B
0	0
0	1
1	0
1	1

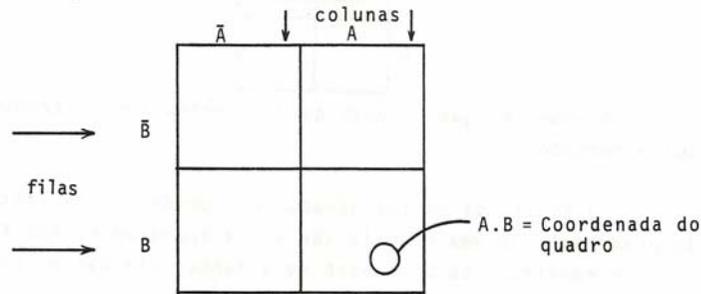
Logo, o diagrama correspondente tem a seguinte forma:

00	10
01	11

Costuma-se representar a combinação 0,0 por  $\bar{A}\bar{B}$ , como ocorre no método da soma de produtos. Procedimento análogo inclui as outras combinações. O diagrama, então, fica da forma:

$\bar{A}\bar{B}$	$A\bar{B}$
$\bar{A}B$	$AB$

Notamos que as colunas e filas possuem variáveis comuns (por exemplo B na fila superior). Isso permite retirar as variáveis de dentro do quadro, representando-as então por suas "coordenadas de quadro", termo que utilizaremos daqui em diante.



Uma peculiaridade do diagrama é que quadros adjacentes se diferenciam por apenas uma coordenada.

Vamos agora à simplificação. Por exemplo, considere-se a tabela:

A	B	L
0	0	1
0	1	0
1	0	1
1	1	0

O método algébrico da soma de produtos leva a:

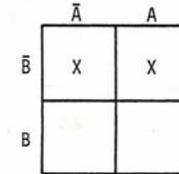
$$L = \bar{A}\bar{B} + A\bar{B}$$

$$L = \bar{B}(\bar{A} + A)$$

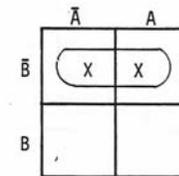
$$L = \bar{B}$$

O método de Veitch é semelhante:

1) Assinalamos os quadros correspondentes às combinações entre A e B onde  $L = 1$

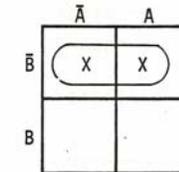


2) Unimos os quadros simétricos à qualquer linha interna.



Dizemos aí que os dois quadros primários se transformaram, em outro secundário.

3) Retiramos as coordenadas dos quadros resultantes (na forma de produtos). Se uma ou mais variáveis aparecem na sua forma normal e complementada, em um quadro resultante, ela não é considerada como coordenada. (não entra no produto).



No nosso exemplo, somente a variável B determina a posição do quadro, visto que A e  $\bar{A}$  aparecem, ambas, nesse quadro.

4) A soma das coordenadas obtidas é o resultado final do problema.

$$L = \bar{B}$$

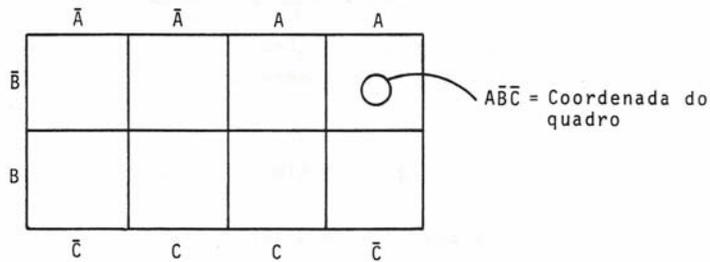
Neste caso, apenas B apareceu como coordenada final de quadro.

O exemplo que acabamos de estudar evidencia o parentesco do

método da soma de produtos com o do Diagrama de Veitch. A união de quadros, com consequente supressão de uma das coordenadas equivale à simplificação de  $(\bar{A} + A)$  na equação algébrica.

### 6.3 - DIAGRAMA PARA 3 VARIÁVEIS

Obedecendo os mesmos critérios anteriores, montamos o Diagrama de Veitch para 3 variáveis.



O processo de simplificação é o mesmo, observando-se o seguinte:

Qualquer quadro pode participar de quantas uniões forem precisas, contanto que ela seja permitida (quadros simétricos a uma linha) e isso contribua para a minimização da equação final.

Analisemos o exemplo seguinte:

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Façamos então o que nos diz o método.

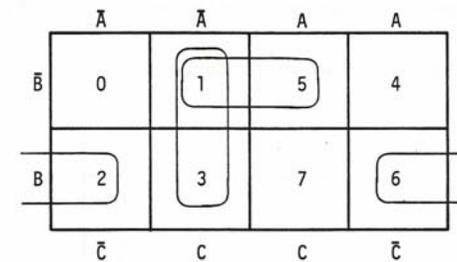
1) Lançar no diagrama, assinalando os quadros correspondentes, todas as combinações em que  $L = 1$ .

	$\bar{A}$	$\bar{A}$	A	A
$\bar{B}$	0	$X_1$	$X_5$	4
B	$X_2$	$X_3$	7	$X_6$
	$\bar{C}$	C	C	$\bar{C}$

OBS: Associamos, para facilitar, cada quadro primário do diagrama a um número decimal equivalente ao binário ABC.

2) Unimos os quadros simétricos a uma linha.

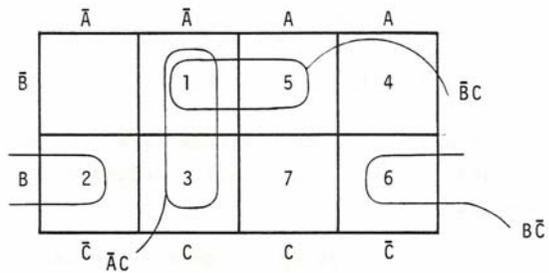
- Os quadros 1 e 3 são simétricos em relação à linha vertical do meio.
- Os quadros 1 e 3 são simétricos em relação à linha horizontal.
- Os quadros 2 e 6 são simétricos em relação à linha vertical intermediária.



Resultam então, 3 quadros ditos secundários.

Cabe aqui a seguinte observação: O quadro 3 poderia ser unido com o quadro 2 ao invés de com o 1, como foi feito. É uma questão de opção; ambos os procedimentos levam a resultados corretos. Porém, nunca poderia ser unido aos dois, pois prejudicaria a minimização da equação, acrescentando mais um termo à ela.

3) Retiramos as coordenadas dos quadros resultantes:

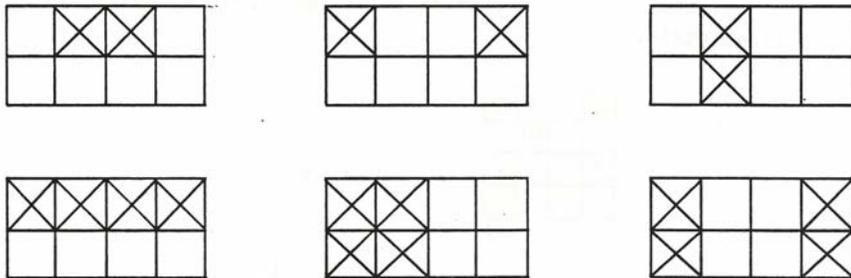


4) Montamos a equação final somando as coordenadas

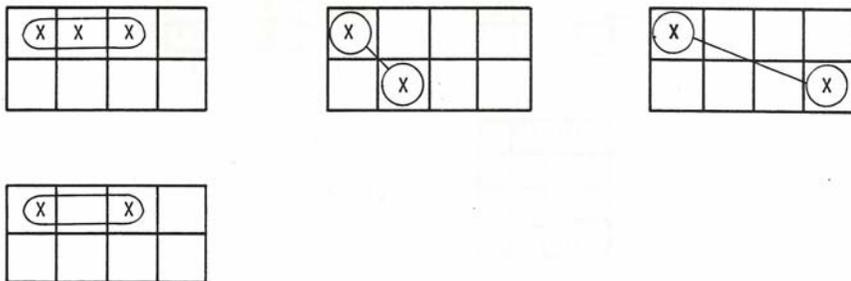
$$L = \bar{B}C + B\bar{C} + AC$$

Essa é a equação minimizada.

Na figura abaixo, apresentam-se todas as possibilidades de união de quadros, incluindo as uniões de 2 quadros secundários, que, sendo simétricos à uma linha, formam um outro terciário, de coordenada única.



São consideradas uniões proibidas:



É interessante notar que os quadros resultantes sempre possuirão 1, 2, 4 ou 8 quadros primários.

#### 6.4 - DIAGRAMA PARA 4 VARIÁVEIS

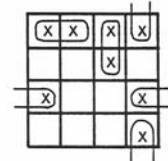
A utilização do Diagrama para 4 ou mais variáveis é mais compensadora, visto que as simplificações algébricas começariam a ficar complicadas.

O Diagrama de Veitch para 4 variáveis pode tomar a forma:

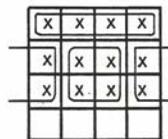
	$\bar{A}$	$A$	$\bar{A}$	$A$	
$\bar{B}$	0	1	9	8	$\bar{C}$
$B$	2	3	11	10	$C$
$\bar{B}$	6	7	15	14	$C$
$B$	4	5	13	12	$\bar{C}$
	$\bar{D}$	$D$	$\bar{D}$	$D$	

O método de simplificação é o mesmo.

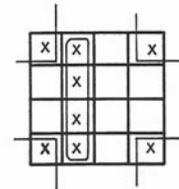
Quanto às uniões permitidas, por simetria, estão representadas a seguir:

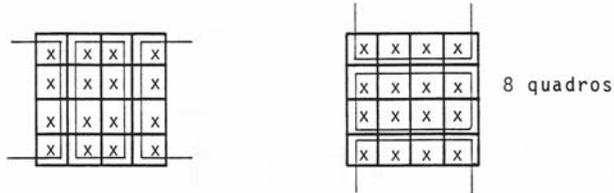


2 quadros

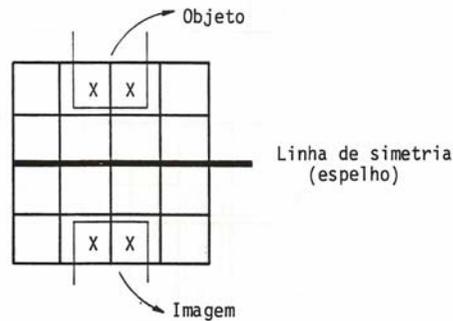


4 quadros





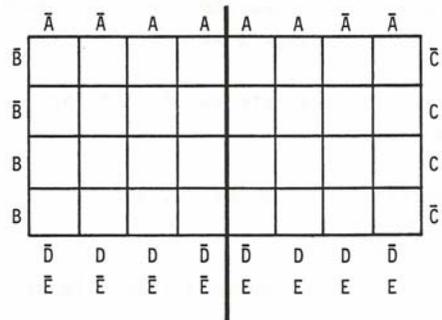
As combinações anteriores se justificam pela existência de linhas de simetria em todas elas, isto é, sempre existe uma linha, como se fosse um espelho, que reparte o quadro em dois, sendo que metade representa o objeto e a outra metade, a sua imagem: EX:



No item 6 dos problemas resolvidos encontram-se dois exemplos de simplificação de equações com 4 variáveis.

6.5 - DIAGRAMA PARA 5 VARIÁVEIS

É obtido pela justaposição de 2 diagramas para 4 variáveis:

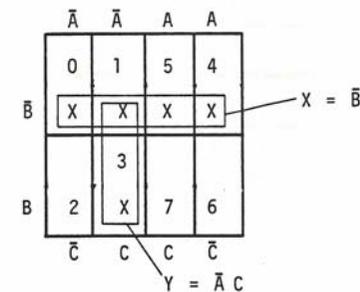


6.6 - EXERCÍCIOS RESOLVIDOS

6.6.1 - Determinar a equação simplificada correspondente à tabela da verdade:

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Solução:



O quadro X resultou do seguinte processo:

Em primeiro lugar, os quadros primários 0 e 1 foram unidos, resultando o secundário 0, 1. Como os quadros 4 e 5 uniram-se para formar outro secundário 4, 5, observou-se a possibilidade de unir esses dois quadros secundários, que são simétricos em relação à linha vertical intermediária, formando o terciário 0, 1, 4 e 5.

Os quadros 1 e 3, simétricos à linha horizontal, resultam no secundário 1, 3, quadro Y.

A equação final, minimizada é, portanto:

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

6.6.2 - Determinar a equação simplificada, na forma NE, correspondente à tabela:

A	B	C	L
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Solução:

	$\bar{A}$	$\bar{A}$	A	A	
$\bar{B}$	0	1	5	X	Y = $\bar{C}$
B	X	X	7	X	
	2	3		6	
	C	C	C	$\bar{C}$	

$X = \bar{A} B$

Os quadros secundários 0,2 e 4,6 são simétricos à linha vertical intermediária, resultando daí a sua união. O terciário formado tem coordenada única  $\bar{C}$ , pois dentre os 4 quadros que o compõem, 0 e 4 possuem  $\bar{B}$ , e 2 e 6 possuem B como coordenadas, cancelando então, essa variável; os quadros 0 e 2 possuem  $\bar{A}$ , e 4 e 6 possuem A, que também a cancelam.

Os quadros 2 e 3 possibilitam mais uma união, simplificando a variável C.

(2 com  $\bar{C}$ , 3 com C).

A equação final é:

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

que sofrendo transformações permitidas pelos teoremas de De Morgan (obj. 5) se torna:

$$L = \overline{AB \cdot C}$$

6.6.3 - Determinar a equação simplificada:

A	B	C	D	L
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Solução:

	$\bar{A}$	$\bar{A}$	A	A	
$\bar{B}$	X <sub>0</sub>	X <sub>1</sub>	X <sub>9</sub>	X <sub>8</sub>	X = $\bar{B} \bar{C}$
B	2	3	11	10	
	6	7	X	X	Y = AB
B	4	X <sub>5</sub>	X	X	
	$\bar{D}$	D	D	$\bar{D}$	

$Z = \bar{C} D$

A fila superior 0,1,8,9 formam um quadro terciário, pelos motivos já vistos.

Já os quadros 5 e 13 formam um secundário simétrico ao 1,9, resultando no terciário Z.

O quadro 14,15 se une ao 12,13 formando Y.

Como já existem os quadros 12,13,14,15 e 0,1,8,9 não há necessidade de formar 8,9,12,13 pois este último somente adicionaria mais um termo à equação final.

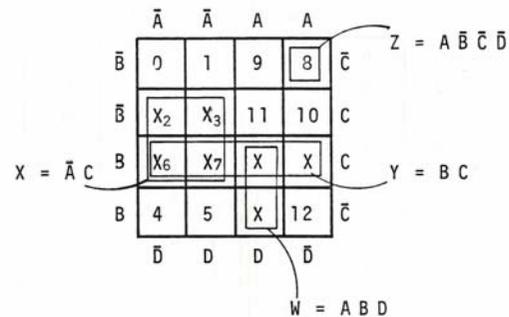
$$L = AB + \bar{B}\bar{C} + \bar{C}D$$

6.6.4 - Simplificar.

$$L = \Sigma (2,3,6,7,8,13,14,15)$$

Esta é uma outra maneira de representar uma soma de produtos. Os números entre os parênteses representam os produtos binários de variáveis A, B, C e D correspondentes.

Solução:



O quadro 8 não possui simétrico, portanto, a sua coordenada é a original, isto é:  $A\bar{B}\bar{C}\bar{D}$ .

A equação resultante é:

$$L = \bar{A}C + BC + ABC + A\bar{B}\bar{C}\bar{D}$$

### 6.7 - PROBLEMAS PROPOSTOS

6.7.1 - Através do diagrama de Veitch para 3 variáveis, encontrar a forma simplificada mínima da equação apresentada na introdução.

6.7.2 - Determinar as equações simplificadas correspondentes às saídas  $L_0$ ,  $L_1$  e  $L_2$ .

A	B	C	$L_0$	$L_1$	$L_2$
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	1	0

7.7.3 - Idem para  $L_3$ ,  $L_4$  e  $L_5$ :

A	B	C	D	$L_3$	$L_4$	$L_5$
0	0	0	0	1	1	1
0	0	0	1	0	1	0
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	1	1	0
1	0	0	1	0	1	1
1	0	1	0	1	1	1
1	0	1	1	1	0	1
1	1	0	0	1	1	1
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	0	0	1

7.7.4 - Esquematizar, com portas NE, um circuito capaz de detectar números binário de 4 bits, que sejam maiores do que 4 e menores do que 14.

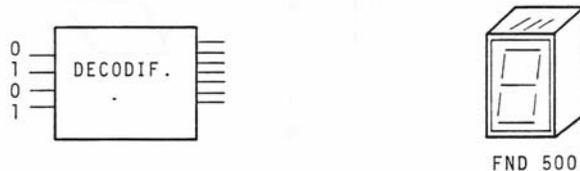
## 7 - DECODIFICADORES E MULTIPLEXADORES

- 7.1 - Introdução
- 7.2 - Decodificadores de 7 segmentos
- 7.3 - Decodificador BCD - decimal
- 7.4 - Multiplexador
- 7.5 - Demultiplexador
- 7.6 - Problemas resolvidos
- 7.7 - Problemas propostos

### 7.1 - INTRODUÇÃO

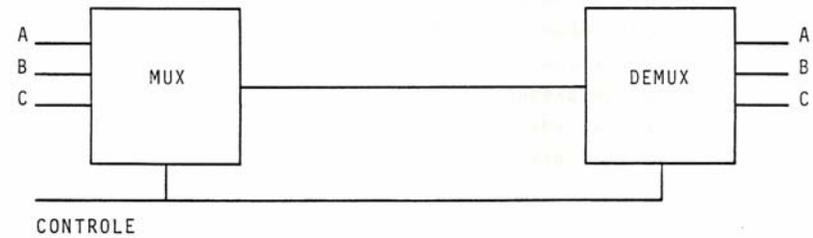
Os circuitos combinacionais até aqui estudados têm aplicações às mais variadas na Eletrônica Digital, mas duas aplicações das mais difundidas, são os circuitos codificadores e multiplexadores.

Um decodificador é qualquer circuito capaz de transformar uma informação escrita em um código binário, em outro de mais fácil acesso. Por exemplo, o decodificador de sete segmentos analisado no item 2 possibilita a transformação de números binários sob forma de níveis de tensão alto e baixo em um número decimal mostrado em um "display" de 7 segmentos. Este mostrador tem emprego largamente difundido hoje em dia e consiste basicamente de 7 "leds" (diodos emissores de luz) que acendem quando excitados pelas saídas do decodificador, formando um carácter decimal.



Os multiplexadores têm funções diferentes. Eles permitem a emissão de dados binários de um a outro local por um número de "linhas" inferior ao número de informações. Isso é possível, pois cada linha é ocupada por diversas informações, que são emitidas sequencialmente, no tempo; o multiplexador deve possuir um circuito de

controle que coloque na linha as informações, cada uma no seu devido tempo de transmissão. Um demultiplexador, no outro extremo da linha, compõe as informações iniciais.

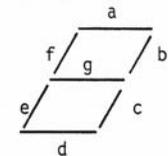


### 7.2 - DECODIFICADOR DE 7 SEGMENTOS

Seja, por exemplo, efetuar a transformação do código BCD natural, ao decimal, indicado num dispositivo de 7 segmentos. Como aparece na figura 1, as quatro entradas do decodificador são carregadas com os números codificados em BCD e as 7 saídas devem fornecer níveis de tensão próprias para os segmentos devidos.

Observe a tabela abaixo.

Nº	BCD	DISPLAY
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9



Nota-se aí que o segmento a, tomado como exemplo, deve acender para as combinações do número 0,2,3,5,6,7,8 e 9, logo, a saída do decodificador que deve excitar esse segmento, a qual chamaremos de La, precisa apresentar nível lógico 1 nessas condições e 0 nas restantes, como na tabela.

É interessante notar que não aparecem na tabela as combinações de 1010 a 1111 pois elas não pertencem ao BCD natural. Surge então a seguinte questão: como deve se comportar  $L_a$  nessas condições? Obviamente, isso fica a critério do projetista do decodificador que terá várias opções. Trataremos aqui do caso em que o estado de  $L_a$  pode ser qualquer, isto é, 0 ou 1, indiferentemente. Essa opção é perfeitamente justificável pois se não existem as combinações citadas anteriormente no código BCD, então não importa quais seriam os correspondentes níveis de saída do decodificador, Representemos por X as condições irrelevantes que  $L_a$  assume:

A	B	C	D	$L_a$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	x
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

O equacionamento de  $L_a$  através do diagrama de veitch é conveniente, pois permite visualizar a equação mais simplificada. Inicialmente, lançamos no diagrama as condições onde  $L_a=1$ . Depois, marcamos também os estados irrelevantes desde que contribuam efetivamente para maior simplificação da equação. Neste caso, notamos que todas as condições irrelevantes devem ser assinaladas, logo,  $L_a=1$  é preferível para todas as combinações de 1010 a 1111.

	$\bar{A}$	$A$	$\bar{A}$	$A$	
$\bar{B}$	$X_0$	1	$X_9$	$X_8$	$\bar{C}$
$\bar{B}$	$X_2$	$X_3$	X	X	C
B	$X_6$	$X_7$	X	X	C
B	4	$X_5$	X	X	$\bar{C}$
	$\bar{D}$	D	D	$\bar{D}$	

A equação final fica, na forma NE:

$$L_a = \bar{A} \cdot \bar{C} \cdot \bar{B}D \cdot \bar{B}D$$

É evidente que as demais saídas, de  $L_a$  e  $L_9$ , devem ser equacionadas de maneira semelhante, mas não necessariamente as combinações de 10 a 15 devem ser assinaladas em sua totalidade.

### 7.3 - DECODIFICADOR BCD - DECIMAL

Por esse nome identifica-se o elemento que transforma o código BCD em um outro de 10 dígitos em que apenas um é excitado para cada combinação binária.

A	B	C	D	$L_0$	$L_1$	$L_2$	...	$L_8$	$L_9$
0	0	0	0	1	0	0	...	0	0
0	0	0	1	0	1	0	...	0	0
0	0	1	0	0	0	1	...	0	0
0	0	1	1	0	0	0	...	0	0
0	1	0	0	0	0	0	...	0	0
0	1	0	1	0	0	0	...	0	0
0	1	1	0	0	0	0	...	0	0
0	1	1	1	0	0	0	...	0	0
1	0	0	0	0	0	0	...	1	0
1	0	0	1	0	0	0	...	0	1
1	0	1	0	0	0	0	...	0	0
1	0	1	1	0	0	0	...	0	0
1	1	0	0	0	0	0	...	0	0
1	1	0	1	0	0	0	...	0	0
1	1	1	0	0	0	0	...	0	0
1	1	1	1	0	0	0	...	0	0



A implementação desse circuito é bem mais fácil, pois não exige métodos complicados para obtenção das equações de saída. Por exemplo:

$$L_0 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$L_1 = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$$

$$L_2 = \bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$$

Muitos outros tipos de decodificadores existem, assim como também existem os circuitos codificadores, com funções inversas aos estudados. Um decodificador decimal terá como função apresentar em correspondência a uma entrada excitada, entre dez, uma combinação binária típica BCD.

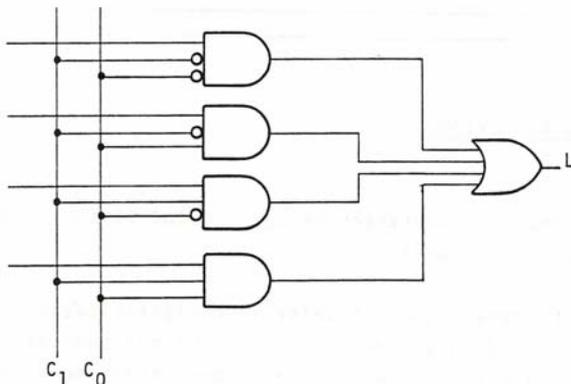
#### 7.4 - MULTIPLEXADORES

Seja, por exemplo, convergir para uma linha de transmissão, uma de 4 informações, selecionada através de terminais de controle.

Os terminais de controle  $C_0$  e  $C_1$ , podem estar em um, de quatro estados binários, 0,0; 0,1; 1,0 e 1,1. Para cada uma dessas combinações, L deve assumir o estado de uma das 4 entradas A,B,C e D



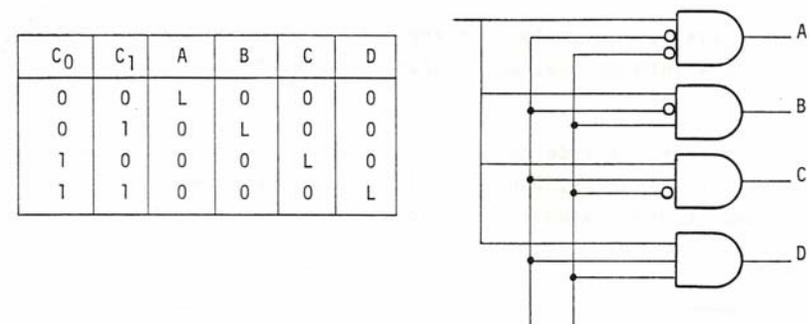
O circuito capaz de produzir esse efeito é o apresentado abaixo. As combinações entre  $C_0$  e  $C_1$  abrem, cada uma, a porta correspondente a cada entrada A,B,C e D, transferindo essa informação para a saída:



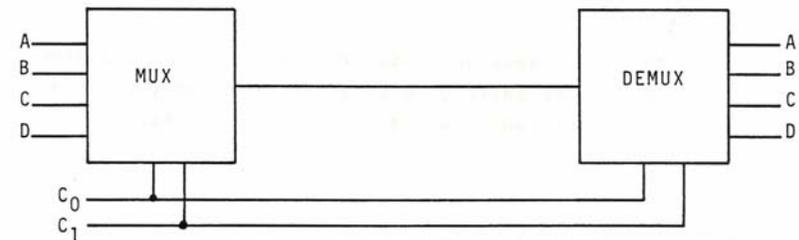
#### 7.5 - DEMULTIPLEXADOR

O circuito que tem função inversa ao multiplexador é o de multiplexador. Ele leva a uma saída, selecionada pelo circuito de controle, a única entrada que possui.

A tabela da verdade desse circuito é sintetizada a seguir:



Fazendo os dois circuitos anteriores trabalharem em sincronismo, é possível montar o sistema de transmissão de 4 informações multiplexadas no tempo.



#### 7.6 - PROBLEMAS RESOLVIDOS

7.6.1 - Deduzir a expressão de  $L_e$  do decodificador de 7 segmentos a partir do BCD natural

Solução:  $L_e$  deve assumir o valor 1, obrigatoriamente, nas condições 0,2,6 e 8. No diagrama de Veitch, se observa que, das condições irrelevantes, apenas a 10 e a 14 são convenientes de serem assinadas. Assim, ficamos com:

	$\bar{A}$	$A$		
$\bar{B}$	$X_0$	1	9	$X_8$
$B$	$X_2$	3	11	$X_{10}$
$\bar{B}$	$X_6$	7	15	$X_{14}$
$B$	4	5	13	12
	$\bar{D}$	$D$	$\bar{D}$	$D$

$$L_e = \bar{B}\bar{D} + C\bar{D} = \overline{\bar{B}\bar{D} \cdot C\bar{D}}$$

7.6.2 - Realizar um codificador decimal - BCD

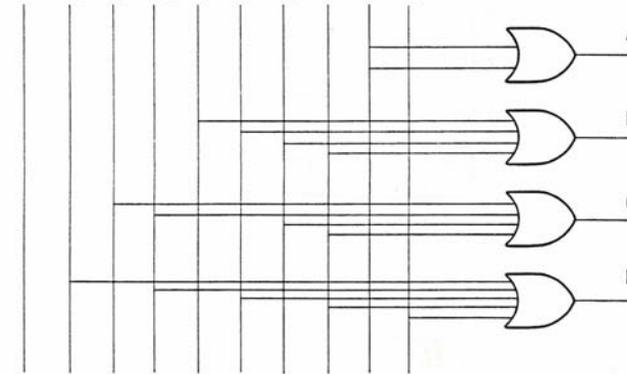
Solução: A tabela da verdade do codificador é a representada a seguir:

$L_0$	$L_1$	$L_2$	$L_3$	$L_4$	$L_5$	$L_6$	$L_7$	$L_8$	$L_9$	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

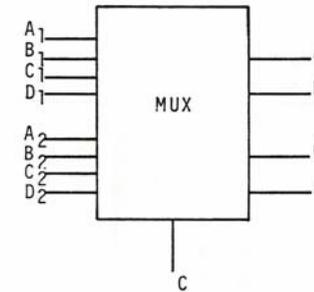
O equacionamento de A,B,C e D se faz por inspeção direta da tabela.

$$\begin{aligned} A &= L_8 + L_9 \\ B &= L_4 + L_5 + L_6 + L_7 \\ C &= L_2 + L_3 + L_6 + L_7 \\ D &= L_1 + L_3 + L_5 + L_7 + L_9 \end{aligned}$$

O circuito do codificador apresenta-se sob o seguinte aspecto:

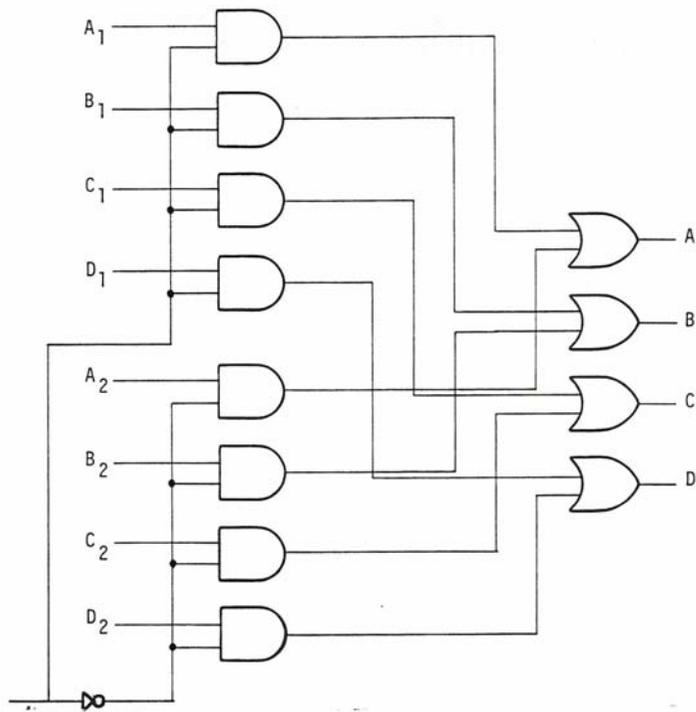


7.6.3 - Realizar um multiplexador "duas linhas para uma linha" onde cada linha consiste de 4 BITS em paralelo.



Solução: O controle  $C_1$  podendo ser 0 ou 1, leva a saída as entradas 1 ou 2, em cada caso. O circuito deve ser tal que:

$$\begin{aligned} A &= A1 \cdot C + A2 \cdot \bar{C} \\ B &= B1 \cdot C + B2 \cdot \bar{C} \\ C &= C1 \cdot C + C2 \cdot \bar{C} \\ D &= D1 \cdot C + D2 \cdot \bar{C} \end{aligned}$$



### 7.7 - PROBLEMAS PROPOSTOS

7.7.1 - Realizar um decodificador 3 em excesso - 7 segmentos de tal forma que as condições irrelevantes sejam utilizadas para máxima simplificação.

7.7.2 - Realizar o decodificador BCD - 7 segmentos do item 2, de forma tal que as condições não existentes (1010 a 1111) induzam o aparecimento do caracter E no display.

7.7.3 - Esquematizar um demultiplexador para o circuito do item 7.6.3.

7.7.4 - Elaborar um sistema que permita mostrar em 2 displays de 7 segmentos, as informações decodificadas provenientes de 2 conjuntos de 4 chaves A1, B1; C1, D1 e A2, B2, C2, D2, multiplexando

as informações de tal forma que sejam transmitidas por apenas 4 canais A, B, C e D até os decodificadores. Utilizar blocos para representar os circuitos decodificadores e multiplexadores, e displays FND 500.

## 8 - FLIP - FLOPS

- 8.1 - Introdução
- 8.2 - FF - RS (NE)
- 8.3 - FF - RS (NOU)
- 8.4 - FF - RS com controle
- 8.5 - FF - D com controle
- 8.6 - Exemplos Práticos
- 8.7 - Problemas Resolvidos
- 8.8 - Problemas Propostos

## 8.1 - INTRODUÇÃO

Dentro da lógica seqüencial, os Flip-Flops são os elementos básicos, permitindo o armazenamento temporário de dados e se constituindo nas peças mais importantes na realização de contadores binários e registros de dados.

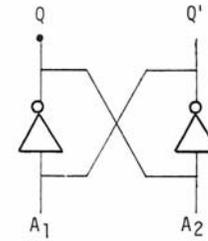
O significado do termo Flip-Flop é um tanto curioso. Não nos atrevemos a traduzir o termo; usamos o original inglês, abreviadamente FF, em nossos textos. A consulta a um dicionário nos levaria às seguintes definições:

FLIP: jogar para cima com o polegar  
 FLOP: cair pesadamente

Isso se relaciona com o fato de as saídas dos FF's comutarem de 1 para 0 ou 0 para 1 em função da mudança de estados nas entradas.

Como dissemos, os FF's são freqüentemente utilizados no armazenamento de dados. Essa é a aplicação maior desses elementos, sendo inclusive encontrados em memórias semi-condutoras.

Para essa aplicação, exige-se que o circuito disponha de dois estados de saída, 0 e 1, que sejam estáveis no tempo. Assim, um exemplo dos mais comuns de FF é o multivibrador biestável. Ele pode ser realizado pelo acoplamento cruzado entre dois inversores como na figura a seguir.



A <sub>1</sub>	Q	Q'
0	1	0
1	0	1

Nota-se aí que, se A<sub>1</sub> assume o nível 0, decorre Q = 1. Esse nível alto em Q é realimentado para A<sub>2</sub> forçando Q'<sub>2</sub> a adquirir nível lógico baixo, ou 0. Como Q' é ligado diretamente em A<sub>1</sub>, há um "retorno" do nível 0 a esse ponto, o que implica na estabilidade do sistema.

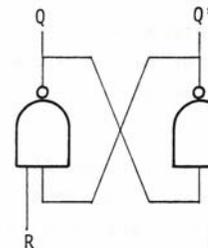
Por raciocínio análogo, a condição em que A<sub>1</sub> = 1 =  $\bar{A}_2$  e Q = 0 =  $\bar{Q}'$  também resulta estável. Assim, se tomarmos a saída Q como referência podemos observar que os estados estáveis Q = 1 ou Q = 0 existem em função dos níveis de entrada aplicados em A<sub>1</sub> ou A<sub>2</sub>. Se forçamos A<sub>1</sub> para nível 0 o FF armazenará em Q o nível 1. Se A<sub>1</sub> for induzido ao nível 1, Q armazenará o nível 0.

Os FF's mais comuns encontrados na prática são os RS, o D e o JK.

Neste capítulo analisaremos os dois primeiros.

## 8.2 - FLIP - FLOP RS (NE)

O acoplamento de duas portas NE conforme a figura abaixo resulta no FF RS (NE), que passaremos a analisar. RS significa reset-set, ou seja, possui dois terminais de entrada que têm por função ativar (set) e desativar (reset) o FF, induzindo a 1 ou 0 a saída referencial Q, respectivamente.



R	S	Q	Q'
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q <sub>0</sub>	Q' <sub>0</sub>

Passemos então à análise do circuito:

a)  $R = 0, S = 0$ : sendo 0 uma das entradas de uma porta NE, obrigatoriamente teremos em sua saída o nível 1.

Então:  $Q = 1$  e  $Q' = 1$

b)  $R = 0, S = 1$ : o nível baixo em R induz  $Q = 1$  e, consequentemente,  $Q' = 0$ .

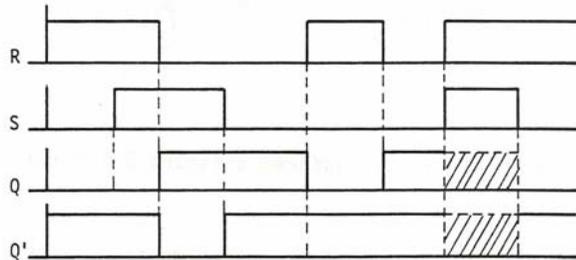
c)  $R = 1, S = 0$ : por simetria do circuito, concluímos que:  $Q = 0$  e  $Q' = 1$ .

d)  $R = 1$  e  $S = 1$ : essa condição das entradas não permite deduzir os estados de Q e Q' de imediato. Mas, verifica-se que são estáveis os estados  $Q = \bar{Q}' = 1$  e  $Q = \bar{Q}' = 0$ , isto é, qualquer um dos dois pode existir.

Concluímos pois, que se  $Q = \bar{Q}'$  na condição anterior a  $R = S = 1$ , o circuito não muda de estado. Porém se  $Q = Q'$  antes de aparecer  $R = S = 1$ , o estado seguinte será indeterminado.

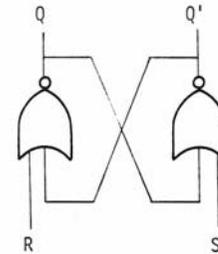
A tabela da verdade ao lado do circuito sintetiza o que foi exposto.  $Q_0$  e  $Q'_0$  representam a condição anterior ao aparecimento da condição de entrada.

Para melhor exemplificar o funcionamento do FF RS (NE), apresentamos na figura seguinte as formas de onda de saída Q e Q' a partir das formas de R e S, arbitrárias.



### 8.3 - FF RS (NOU)

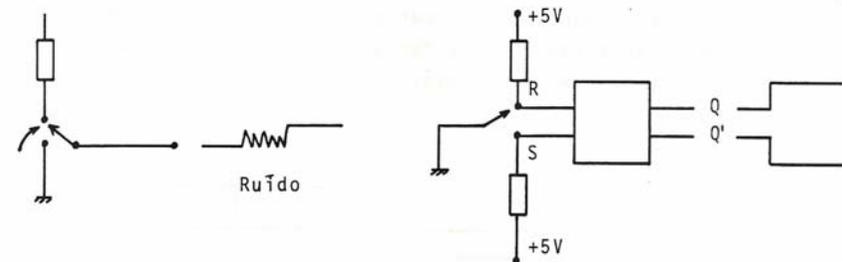
Com portas NOU também é possível realizar um FF RS, que adquire a forma apresentada na figura e possui a tabela da verdade ao lado.



R	S	Q	Q'
0	0	$Q_0$	$Q'_0$
0	1	1	0
1	0	0	1
1	1	0	0

Uma importante aplicação do FF RS (NE) é no chaveamento digital anti-ruído.

As chaves comuns, quando comutam de 0 para 1, ou vice-versa, geram um ruído, devido ao repique dos contatos, prejudicial aos circuitos lógicos. Isso é ilustrado na figura abaixo, à esquerda.



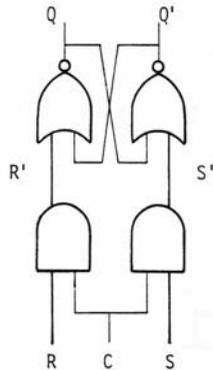
O FF RS ligado conforme a figura à direita evita esse problema.

Estando a chave em S, inicialmente, o FF se encontra desativado ( $R = 1$ ), logo,  $Q = 0$ . Quando a chave atinge o ponto R, Q comuta para 1 e aí permanece, pois, o "repique" pode apenas produzir  $R = S = 1$ , ou seja,  $Q = Q_0 = Q' = Q'_0$  o que significa a permanência das saídas no estado anteriormente adquirido.

### 8.4 - FLIP-FLOPS RS (NOU) COM CONTROLE

O terminal C adicionado ao RS permite um controle sobre a comutação desse elemento pois somente se  $C = 1$  as entradas R e S surtirão efeito.

No caso em que C tem nível baixo, 0, as estradas R e S não "passam" pela porta de controle e o FF não consegue mudar de estado, pois  $R' = S' = 0$  induzem Q e Q' à permanência no estado anterior.



R	S	C	Q	Q'
0	0	0	$Q_0$	$Q'_0$
0	1	0	$Q_0$	$Q'_0$
1	0	0	$Q_0$	$Q'_0$
1	1	0	$Q_0$	$Q'_0$
0	0	1	$Q_0$	$Q'_0$
0	1	1	1	0
1	0	1	0	1
1	1	1	0	0

Dizemos então que o terminal C tem ação de controle sobre o FF.

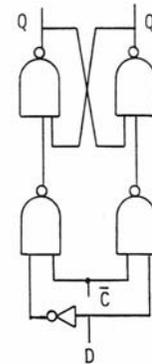
$C = 0$ : o FF não comuta

$C = 1$ : o FF funciona normalmente

Dizemos também que esse FF é sensível ao nível 1.

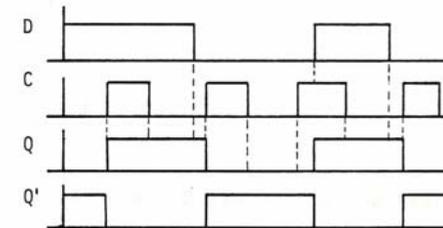
### 8.5 - FLIP-FLOP D COM CONTROLE

O FF D ou armazenador de dados, também conhecido por LATCH, possibilita o armazenamento do bit introduzido no terminal D no momento em que o FF é sensibilizado. A figura apresenta um FF D sensível a nível 0, que não é nada mais do que um FF RS (NE) modificado, onde se faz  $D = S = \bar{R}$ . A tabela ao lado sintetiza o funcionamento do mesmo.



D	C	Q	Q'
0	0	0	1
0	1	$Q_0$	$Q'_0$
1	0	1	0
1	1	$Q_0$	$Q'_0$

Como exemplo de aplicação, apresentamos as formas de onda Q e Q' em função de C e D. Supomos condição inicial  $Q = 0$  e  $Q' = 1$ . Observe-se que o estado de Q é o mesmo que o estado de D no instante do último pulso de controle.



### 3.6 - EXEMPLOS PRÁTICOS

Podemos buscar na família TTL (transistor-transistor-logic) alguns exemplos de circuitos lógicos integrados que desempenham funções de Flip-Flops:

O 7475 é constituído por 4 FF's tipo D com controle sensível a nível alto.

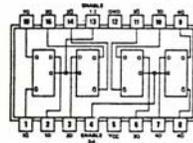
A tabela da verdade apresentada ao lado do "lay-out" do CI, reproduzida na sua forma original é interpretada da seguinte maneira:

Se a entrada G (de controle) estiver em 1 (H) as saídas Q e  $\bar{Q}$  adquirem estados tais que  $Q = \bar{Q} = D$ .

Se a entrada do controle G for levada a nível 0 (L), não importa o estado que D assuma (X) pois as saídas Q e  $\bar{Q}$  permanecem no estado anterior, quando C era igual a 1. ( $Q_0, \bar{Q}_0$ )

FUNCTION TABLE  
(Each Latch)

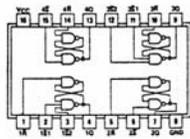
INPUTS		OUTPUTS	
D	G	Q	$\bar{Q}$
L	H	L	H
H	H	H	L
X	L	$Q_0$	$\bar{Q}_0$



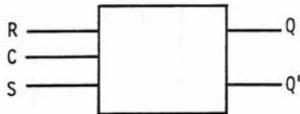
Quatro FF's  $\bar{R} \bar{S}$  são encontrados no 74279. Observa-se aí que em dois exemplares a entrada  $\bar{S}$  é subdividida em duas devido à porta NE de 3 entradas: é necessário que as duas entradas estejam em nível alto para que tenhamos um S também alto.

FUNCTION TABLE

INPUTS		OUTPUT
$\bar{S}'$	$\bar{R}$	Q
H	H	$Q_0$
L	H	H
H	L	L
L	L	H*



Com relação à simbologia adotada, podemos representar os FF através de um retângulo indicando as entradas de controle com ou sem o inversor, pequena circunferência entre o terminal e o bloco.



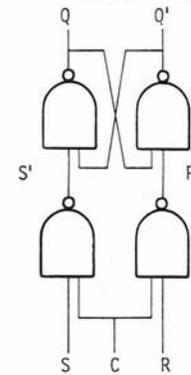
FF RS sensível a "1"



FF D sensível a "0"

8.7 - PROBLEMAS RESOLVIDOS

8.7.1 - Determinar a tabela da verdade do FF apresentado.



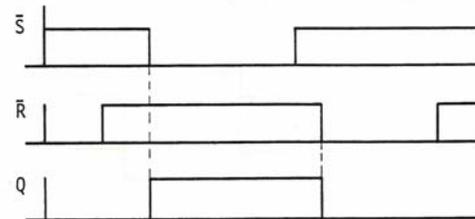
Solução:

Sempre que o terminal de controle estiver em nível 0, as saídas das portas NE R' e S' serão altas, do que decorre (item 2) a permanência dos estados anteriores nas saídas. Com C = 1, as entradas R e S têm acesso aos pontos R' e S', resultando então:

S	R	S'	R'	Q	Q'
0	0	1	1	$Q_0$	$Q'_0$
0	1	0	1	0	1
1	0	1	0	1	0
1	1	0	0	1	1

8.7.2 - Determinar a forma de onda de saída do FF RS do 74279 a partir das de entrada,  $\bar{S}$  e  $\bar{R}$ .

Solução:



$\bar{S}$	$\bar{R}$	Q
H	H	$Q_0$
L	H	H
H	L	L
L	L	H

8.9 - PROBLEMAS PROPOSTOS

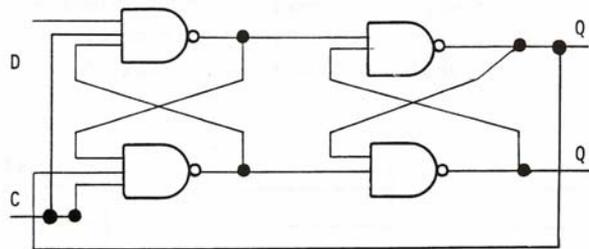
8.9.1 - Determinar as formas de onda de saída do FF do item 3 se lhes aplicam nas entradas R e S as mesmas formas de onda do item 2.

8.8.2 - Desenhar os símbolos dos Flip-Flops

- a) RS
- b) D
- c) D sensível a nível 1
- d) RS sensível a nível 0

8.8.3 - É possível realizar uma chave anti-ruído com o FF RS (NOU)? Em caso afirmativo, qual é a configuração do circuito?

8.8.4 - Determinar a tabela da verdade do FF da figura.

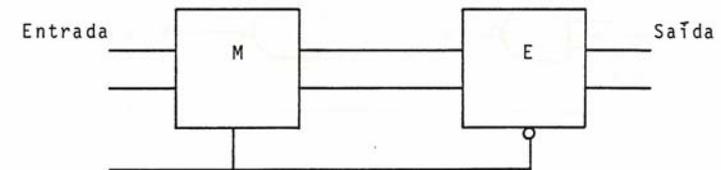
9 - FLIP FLOPS SENSÍVEIS À BORDA

- 9.1 - Introdução
- 9.2 - FF D
- 9.3 - FF J K
- 9.4 - Divisores de Frequência
- 9.5 - Problemas Resolvidos
- 9.6 - Problemas Propostos

9.1 - INTRODUÇÃO

Os flip-flops sensíveis à nível apresentam certas desvantagens em relação ao tempo de sensibilidade de suas entradas. Sendo necessária a presença de um pulso completo na entrada de controle para que o FF possa ser influenciado pelos níveis de entrada, é bem possível que ocorra mudanças de estado lógico nessas entradas, enquanto dura o pulso de controle, que por sua vez provocarão na saída outras mudanças muitas vezes indesejáveis. Era, pois, necessário que se criasse um FF que se sensibilizasse durante um tempo curto não permitindo às variações rápidas da entrada interferir na saída de modo inconveniente.

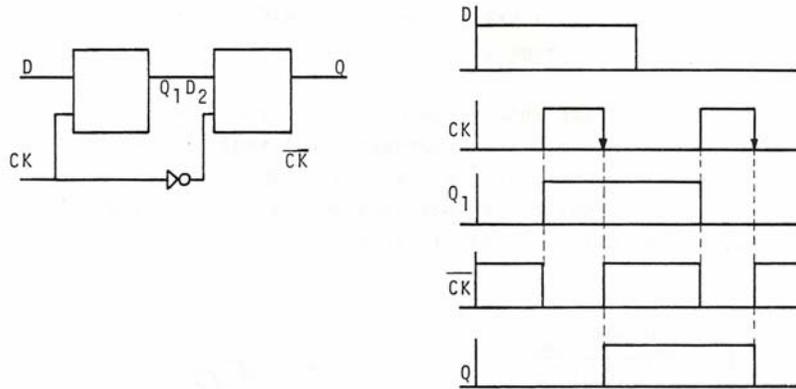
Um tipo de FF que possibilita tal fato é o MASTER-SLAVE que se constitui de 2 FF's sensíveis à nível associados em série. O 1º FF, que recebe a informação de entrada, é o MESTRE (master em inglês), e o segundo, conhecido por ESCRAVO (slave), proporciona a saída do conjunto. A diferença entre os dois está no nível de sensibilidade, que são opostos.



Durante o nível alto do clock, o MESTRE retém a informação de entrada, porém, o ESCRAVO permanece insensível. Podem as estradas variarem seus níveis lógicos que as saídas não comutarão. Quando o clock volta a zero, então é o ESCRAVO que se sensibiliza, proporcionando

nando a informação na saída. Durante o tempo em que o ESCRAVO permanece sensibilizado é a vez do MESTRE se tornar insensível, o que significa estabilidade na saída.

Por exemplo, sejam dois FF's D associados como na figura abaixo: aí podemos observar que durante todo o tempo em que o clock = 1, Q é estável. Somente durante a BORDA de descida do clock é que a saída tem condições de comutar, assumindo o estado de entrada D.

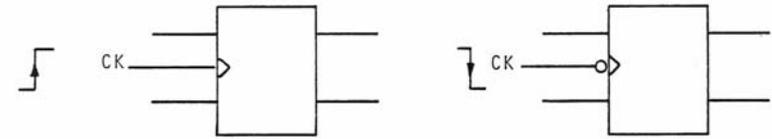


Concluimos, pois, que apenas durante o tempo muito curto, em que dura a transição do clock de 1 para 0, o FF MASTER SLAVE, ou MESTRE ESCRAVO, tem condições de comutar.

Esse FF pertence à classe dos sensíveis à borda do sinal de controle; tanto podem se sensibilizar com a borda de descida como foi o caso do analisado a pouco, como com a borda de subida, exemplificados na continuação do texto.

É bom que se diga que nem todos os sensíveis à borda são MASTER-SLAVE. Existem outros, de implementação mais complexa, que realizam funções diferentes das vistas até aqui, como é o caso do JK.

Os símbolos usados para representar os sensíveis à borda são apresentados abaixo: o da direita é sensível a borda de descida e o da esquerda é sensível à borda de subida.



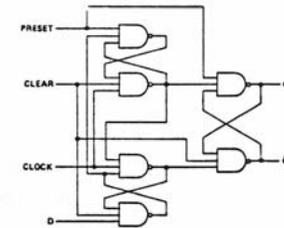
9.2 - FF D

Como exemplo prático de FF D sensível à borda, citamos o 7474, de tecnologia bipolar.

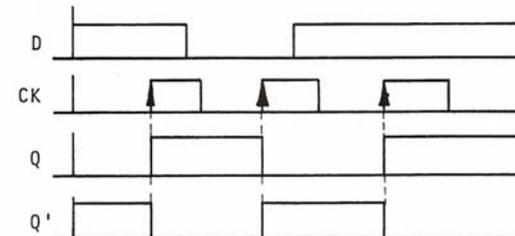
O diagrama interno do CI referente a cada FF, que inclui entradas de PRESET e CLEAR é apresentado a seguir. Como se vê, aumenta a complexidade nos sensíveis à borda. Foge ao escopo da obra a análise deste circuito, interessando-nos mais o comportamento do FF em resposta à aplicação dos sinais de entrada.

FUNCTION TABLE

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	Q'
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	1	H	H	L
H	H	1	L	L	H
H	H	L	X	Q <sub>0</sub>	Q <sub>0</sub>



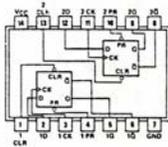
Suponhamos que as formas de onda D e Ck são aplicadas ao FF D cuja tabela da verdade aparece na figura anterior. Cumpre determinarmos as formas de onda de saída Q e Q'.



Analisando os resultados obtidos em Q e Q', que foram arbitrariamente inicializados nos estados 0 e 1, respectivamente, concluímos o seguinte:

Durante as bordas de subida dos pulsos de clock, o FF armazena os estados da entrada D na sua saída Q, oferecendo ainda o estado complementar em Q'.

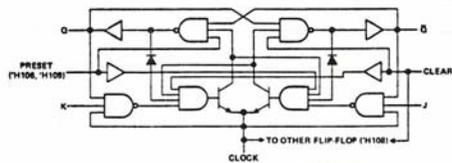
Assim, o símbolo correto para esse FF é o que aparece abaixo, no lay-out do CI.



A maior aplicação dos FF's D, como já foi dito anteriormente, é no armazenamento de dados, em dispositivos conhecidos como registros. São empregados vários FF's D associados em série ou paralelo, cada qual com capacidade de armazenar um bit de informação.

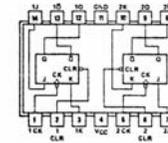
9.3 - FF JK

De todos os FF's estudados nenhum é mais versátil, e portanto, mais importante, que o FF JK. Circuito de relativa complexidade mostrado pela figura abaixo, possibilita uma inovação das mais importantes: ele tem uma capacidade de inverter os estados das saídas mediante determinada combinação de estados nas entradas.



No exemplo escolhido (74103) verificamos pela tabela da verdade que a condição J=K=1 (H) tem a propriedade referida.

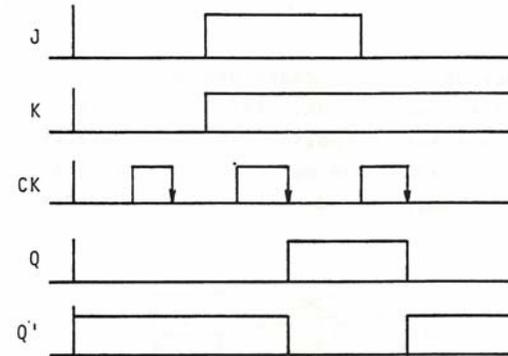
FUNCTION TABLE						
INPUTS				OUTPUTS		
CLEAR	CLOCK	J	K	Q	Q'	
L	X	X	X	L	H	
H	↑	L	L	Q <sub>0</sub>	Q' <sub>0</sub>	
H	↑	H	L	H	L	
H	↑	L	H	L	H	
H	↑	H	H	TOGGLE	TOGGLE	
H	H	X	X	Q <sub>0</sub>	Q' <sub>0</sub>	



Verificamos também que é durante a borda descendente do sinal de clock que ocorre comutação no FF.

Essa condição especial que é a inversão dos estados das saídas, aliada à sensibilidade à borda, é que permitiu a obtenção de contadores digitais dos mais diferentes tipos, e de implementação bem mais complexa sem a sua utilização.

Para que fique mais evidente a função do FF JK sensível à borda de descida tomado como exemplo, apresentamos as formas de onda de saída obtidas a partir das de entrada.



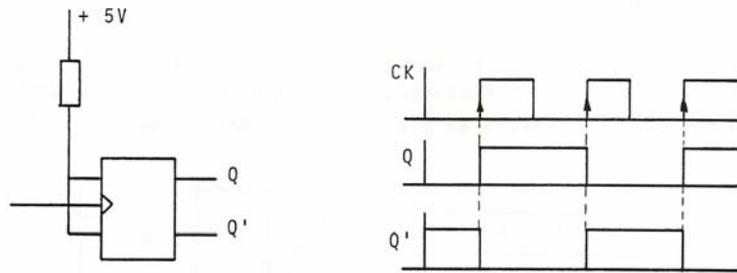
9.4 - DIVISORES DE FREQUENCIA

Os circuitos contadores binários se utilizam grandemente dos FF's, muitas vezes dispostos sob a forma de divisores de frequência.

Um divisor de frequência é um FF que apresenta em sua saída um sinal de frequência igual à metade do sinal de entrada, geralmente aplicado no clock do FF.

O JK, quando as duas entradas são levadas a nível alto, comporta-se como divisor de frequência, pois cada pulso de clock provoca uma comutação na saída, sendo necessários portanto, 2 pulsos na entrada para o aparecimento de apenas 1 pulso na saída.

Trabalhando assim, os FF's recebem, freqüentemente a designação de FF's "T". A figura abaixo exemplifica bem o que foi exposto. Basta comparar as freqüências dos sinais Ck e Q.



Cumpra ainda observar certas limitações existentes no funcionamento dos sensíveis à borda, no que se refere aos tempos de propagação dos sinais pelo interior das portas.

Devido ao fato de a resposta do sistema não ser instantânea, exige-se que os dados de entrada permaneçam por um certo tempo estáveis, dando "chance" ao FF de realizar a comutação.

Os manuais definem 2 intervalos de tempo importantes:

**SET UP TIME:** é o tempo mínimo exigido de permanência de um dado na entrada do FF antes que ocorra a transição ativa no clock. É da ordem de nano segundos.

**HOLD TIME:** é o tempo mínimo em que o dado deve permanecer na entrada depois da transição de clock.

Por exemplo, o 7474 exige que um dado para ser armazenado deva estar presente na entrada 20 ns antes e 5 ns depois da transição de clock para ser efetivamente armazenado.

Portanto,  $t_{set-up} = 20 \text{ ns}$  e  $t_{hold} = 5 \text{ ns}$ .

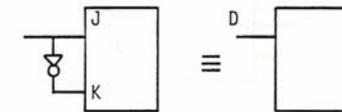
9.5 - PROBLEMAS RESOLVIDOS

9.5.1 - Transformar um flip-flop JK sensível à borda de descida em um FF D sensível à borda de subida.

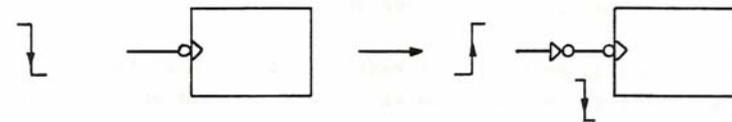
Solução:

a) A transformação de FF JK em D se faz pelo acoplamento entre as entradas J e K através de uma porta NÃO. Assim,  $J = \bar{K}$  elimina as condições de entrada em que  $J=K=1$  ou  $J=K=0$ . Chamamos D à entrada

D	J	K	Q	Q'
0	0	1	0	1
1	1	0	1	0



Quanto à sensibilidade à borda de subida, basta acrescentar outra porta inversora como mostra a figura:

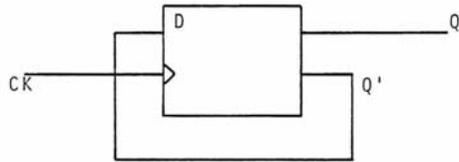


9.5.2 - Transformar em um divisor de frequência o FF D sensível à borda.

Solução:

Para que o FF D mude de estado sempre que ocorra um pulso de clock, é necessário que a entrada de dados inverta seu estado a

cada transição efetuada. Para isso, se aproveita a saída  $Q'$  do FF, que é realimentada à entrada, como mostra a figura:



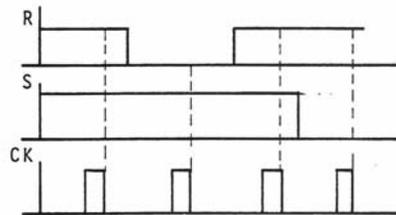
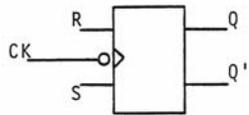
### 9.6 - PROBLEMAS PROPOSTOS

9.6.1 - Determinar a tabela da verdade do FF MASTER SLAVE em que o MASTER é um RS e o SLAVE é um JK.

OBS:

R	S	Q	Q'
0	0	1	1
0	1	1	0
1	0	0	1
1	1	$Q_0$	$Q'_0$

9.6.2 - Determinar as formas de saída  $Q$  e  $Q'$  de um FF RS simbolizado abaixo a partir das formas de onda de entrada.



9.6.3 - Transformar um FF RS, cuja tabela aparece no exercício 9.6.2 em:

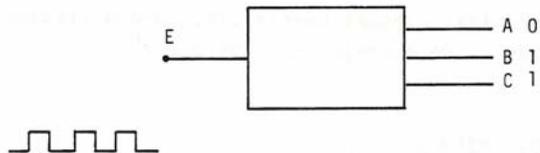
- um divisor de frequência
- um FF D.

## 10 - CONTADORES ASSÍNCRONOS

- 10.1 - Introdução
- 10.2 - Contador Módulo 4 - Análise
- 10.3 - Contadores Progressivos e Regressivos
- 10.4 - Outros Contadores
- 10.5 - Exemplo Prático
- 10.6 - Problemas Resolvidos
- 10.7 - Problemas Propostos

### 10.1 - INTRODUÇÃO

Um contador binário é um sistema digital de uma entrada e  $n$  saídas que apresenta, através de níveis lógicos altos e baixos, combinações binárias que correspondem a um número igual ao número de pulsos que foram aplicados na entrada do contador. Por exemplo, se aplicarmos 3 pulsos na entrada do sistema representado abaixo, as suas saídas A, B e C apresentarão, respectivamente, os valores lógicos 0, 1 e 1, que, lidos como um número ABC, equivalem ao binário  $011_2 = 3$ .



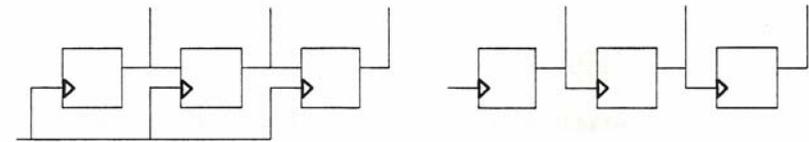
Internamente, os contadores são constituídos por FF's associados em série, correspondendo cada saída dos FF's às saídas do contador. Normalmente, a entrada do contador é a entrada de "clock" de um ou mais FF's da associação. Assim, um pulso aplicado nessa entrada, pode sensibilizar os FF's induzindo-os a comutarem de maneira tal a incrementar o número da saída.

Em relação à sensibilização dos FF's, os contadores binários se classificam em dois grupos: os síncronos e os assíncronos.

Os contadores síncronos têm a característica de todos os FF's, que devem comutar devido à aplicação de um pulso de entrada, o fazerem simultaneamente. Os classificados como assíncronos já se

comportam diferentemente: os pulsos de entrada são aplicados apenas no primeiro FF, na maioria das vezes, e é a saída deste que vai sensibilizar o seguinte, isto é, as comutações se processam em cadeia. Devido aos tempos de atraso inerentes aos FF's, as saídas não comutam ao mesmo tempo, ou, em outras palavras, as comutações são assíncronas.

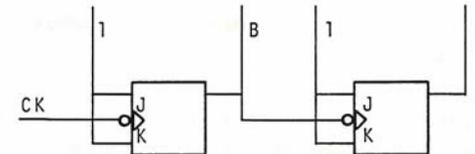
Na figura abaixo representam-se, em blocos, o contador síncrono (esquerda) e assíncrono (direita).



Define-se MÓDULO de um contador, como o número total de combinações binárias que podem aparecer na saída. Assim, se um contador possui  $n$  saídas, o número de combinações é  $2^n$ , o que quer dizer que o módulo máximo que pode ter este contador é  $2^n$ . Isso não significa que todos os contadores de  $n$  saídas têm esse módulo pois em alguns casos são suprimidas algumas combinações, propositadamente, a fim de se obter módulos de contagem menores que  $2^n$ .

### 10.2 - CONTADOR MÓDULO 4 - ANÁLISE

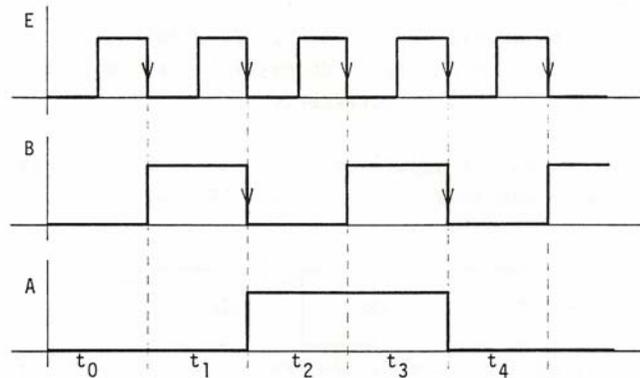
A associação de 2 FF's divisores de frequência, tais como os estudados no capítulo anterior, resulta em um contador módulo 4, já que  $2^2 = 4$ .



No exemplo apresentado, utilizam-se 2 FF's JK onde  $J=K=1$ , o que os torna divisores de frequência.

A entrada do primeiro FF é a entrada do contador, logo, em B aparecerá a frequência de entrada dividida por 2. A seguir, nota-se que a entrada do segundo FF é a saída do primeiro, portanto, aí se obtém outra divisão, resultando em A a frequência de entrada dividida por 4.

As formas de onda apresentadas a seguir demonstram o funcionamento do contador. Observe-se que ambos os FF's são sensíveis à borda de descida, e que as condições iniciais para A e B são arbitrárias como 0.



Antes da aplicação do 1º pulso, ou melhor ainda, do término do 1º pulso, no intervalo de tempo  $t_0$ , temos:

$A=B=0$ , logo:

$n^\circ AB = 00 (0)$

Após o 1º pulso, no intervalo  $t_1$ :

$n^\circ AB = 01 (1)$

Após o 2º pulso, ou  $t_2$ :

$n^\circ AB = 10 (2)$

Após o 3º pulso, em  $t_3$ :

$n^\circ AB = 11 (3)$

Já no intervalo  $t_4$ , após o 4º pulso, a condição de saída retorna a zero reiniciando-se o ciclo.

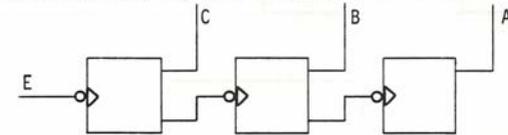
Conclui-se pois, que o contador módulo 4 conta até 3:

00, 01, 10, 11

### 10.3 - CONTADORES PROGRESSIVOS E REGRESSIVOS

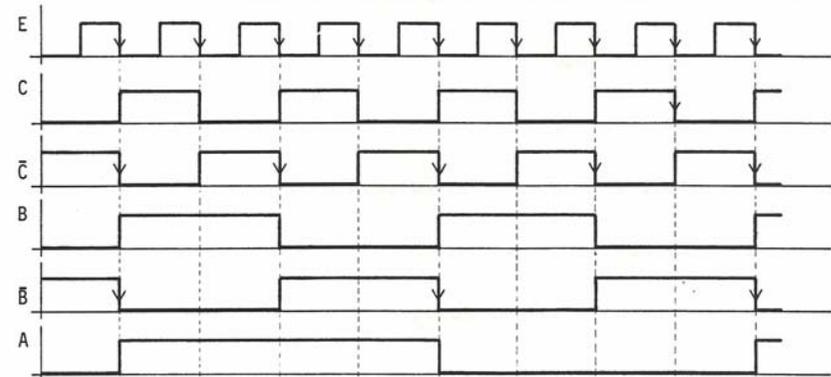
Se os acoplamentos entre os FF's for realizado entre a saída Q' do anterior e o "clock" do seguinte, a ordem da contagem é invertida. Ela passa a ser decrescente.

O exemplo a seguir utiliza esse tipo de acoplamento, no contador binário módulo 8, que utiliza 3 FF's JK.



As entradas J e K, não representadas, são feitas iguais a 1.

Supondo-se condições iniciais iguais a 1 para todas as saídas, são analisadas aqui, através de formas de onda, as combinações de saída obtidas para alguns pulsos de entrada.

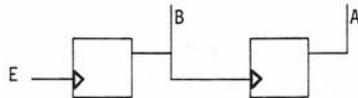


Fazendo-se a leitura dos valores de A, B e C, nessa ordem, obtém-se a seqüência de contagem:

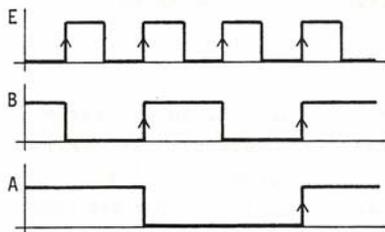
E	A	B	C	nº
pulsos				
0	0	0	0	0
1	1	1	1	7
2	1	1	0	6
3	1	0	1	5
4	1	0	0	4
5	0	1	1	3
6	0	1	0	2
7	0	0	1	1
8	0	0	0	0

Outra forma de se obter contagem regressiva, utilizando acoplamento "Q-clock", é utilizar FF's sensíveis à borda de subida.

Por exemplo, analisemos o contador módulo 4 representado abaixo:



As formas de ondas obtidas a partir dos pulsos de entrada E, são apresentadas a seguir. Note que os FF's comutam durante o crescimento da tensão do clock:

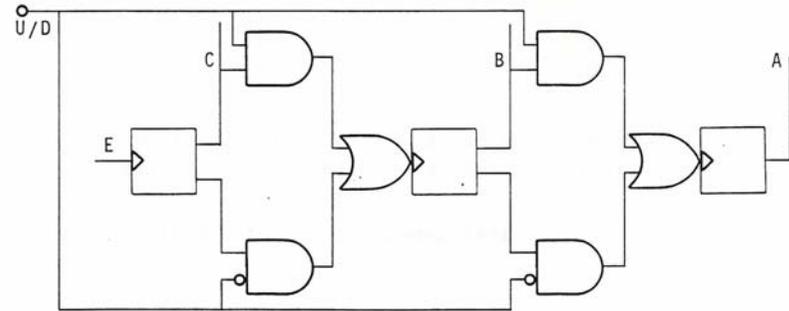


E	A	B	nº
0	1	1	3
1	1	0	2
2	0	1	1
3	0	0	0
4	1	1	3

Se um contador tiver os seus FF's sensíveis ao crescimento de tensão de "clock" e ao mesmo tempo possuir acoplamento do tipo "Q' - clock", então ele será um contador progressivo como se pode deduzir.

10.4 - OUTROS CONTADORES

Já que conhecemos os modos de inverter os sentidos de contagem de um contador binário, é possível a realização de um contador reversível, isto é, que conte tanto progressiva como regressivamente, dependendo de um sinal de comando (U/D).

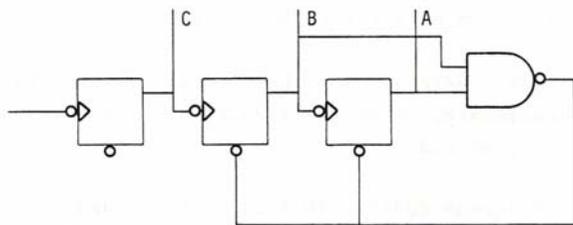


A figura acima apresenta um contador reversível. Se a entrada de controle U/D estiver no nível lógico 1, então as portas E superiores se abrirão deixando passar os sinais de Q para o "clock" seguinte. O contador será regressivo. Se U/D for baixo (0), então serão as portas inferiores que permitirão os acoplamentos do tipo "Q'-clock", e o contador se torna progressivo.

Podemos também conseguir contadores de módulos diferentes de 2<sup>N</sup>, utilizando as entradas de Clear dos FF's. Por exemplo, se já implementar, a partir de um contador módulo 8, progressivo, outro de módulo 6.

É característica do contador módulo 6, apresentar a combinação 000 logo após a combinação 101 (5). O que deve ser feito então é uma realimentação que detecte a combinação 110 (6) na saída e envie, imediatamente, um pulso de clear para os FF's que estiverem no estado 1.

Isso se consegue através da porta NE de realimentação, que produz um sinal lógico baixo na saída sempre que  $B=A=1$ .



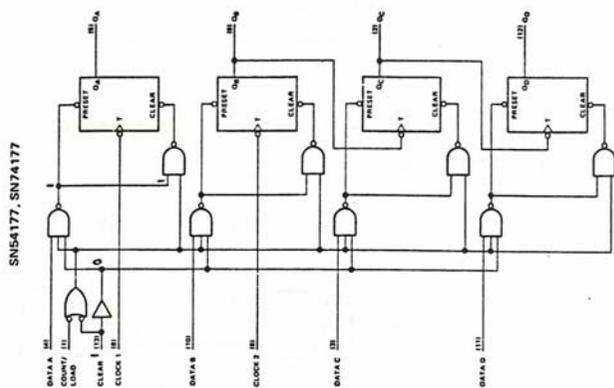
Note-se que a combinação 110 aparece na saída, por um instante muito curto (alguns ns). Se o dispositivo estiver acoplado a apenas um "display" numérico, através de um decodificador, essa rápida passagem pelo nº 6 não será notada, porém em certas aplicações, não são admissíveis essas combinações intermediárias muito comuns em contadores assíncronos.

10.5 - EXEMPLO PRÁTICO

O 74177 é um contador binário assíncrono progressivo módulo 16 "presetável", isto é, é possível inicializá-lo com qualquer combinação.

O circuito e a tabela da verdade são apresentados abaixo, e a análise do mesmo é feita a seguir:

COUNT	OUTPUT			
	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H



1) A saída  $Q_A$  deve ser conectada à entrada do  $CLOCK_2$  para se ter um contador módulo 16. O fato de não existir essa ligação internamente, dá ao usuário a opção de utilizar o sistema como contador módulo 8 e possuir um FF divisor por 2 "de quebra".

2) Quando a entrada  $COUNT/LOAD$  está a nível alto, o contador funciona normalmente, desde que a entrada  $CLEAR$  esteja desativada (1),  $CLOCK_1$  é a entrada.

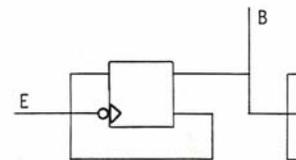
3) Se fizermos  $COUNT/LOAD$  igual a 0, os dados introduzidos em data A, B, C e D aparecerão imediatamente nas saídas  $Q_A$ ,  $Q_B$ ,  $Q_C$  e  $Q_D$ , através da ação dos controles de Clear e Preset nos FF's.

10.6 - PROBLEMAS RESOLVIDOS

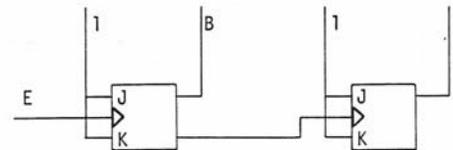
10.6.1 - Realizar um contador módulo 4 progressivo com FF D.

Solução:

Basta fazer o FF D se comportar como divisor de frequência e teremos então o contador, implementado-o conforme a figura abaixo:



10.6.2 - O contador abaixo apresenta contagem progressiva ou regressiva?



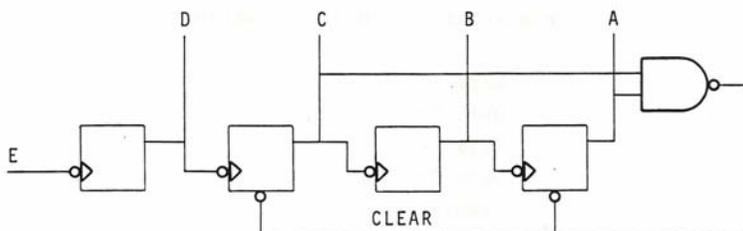
Solução:

Tomando como base o contador de FF's sensíveis à descida e acoplamento do tipo "Q-clock" que é progressivo, nota-se que duas inversões de sentido de contagem foram realizadas, isto é, o contador apresenta sentido progressivo de contagem.

10.6.3 - Elaborar um contador módulo 10 a partir de realimentações externas em um módulo 16. O contador será usado para acionar um dispositivo visual de numeração.

Solução:

Ao atingir os estados de saída 1010, deve ocorrer realimentação própria para o CLEAR do conjunto, então, basta conectar uma porta NE às entradas A e C e com saída conectada nos terminais de CLEAR dos FF's segundo e quarto.



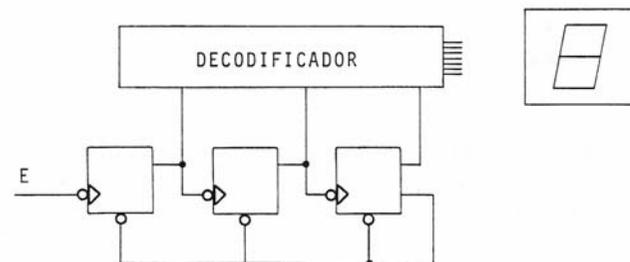
### 10.7 - PROBLEMAS PROPOSTOS

10.7.1 - Implementar um contador regressivo módulo 16 com FF's RS.

10.7.2 - Se as saídas de um contador binário progressivo módulo 4 implementado com FF's JK forem retirados em Q' ao invés de Q, o que ocorre com o sentido de contagem do mesmo?

Sugestão: Analisar o contador através das formas de ondas.

10.7.3 - Quais os números que aparecerão e poderão ser notados no display de 7 segmentos do sistema? (FF's JK).



10.7.4 - Realizar um contador binário progressivo módulo 13 a partir de um módulo 16.

10.7.5 - Verificar através da análise de estados lógicos, o que ocorre com as saídas do 74177 quando:

Data A = 1	Data A = 1
Data B = 1	Data B = 1
Data C = 0	Data C = 0
Data D = 1	Data D = 1
Count/load = 0	Count/load = 0
Clear = 1	Clear = 0

## 11 - CONTADORES SÍNCRONOS

- 11.1 - Introdução
- 11.2 - Contador Progressivo módulo 3 - método de análise
- 11.3 - Outros Módulos de Contagem
- 11.4 - Contadores Comerciais
- 11.5 - Divisores de Frequência
- 11.6 - Problemas Resolvidos
- 11.7 - Problemas Propostos

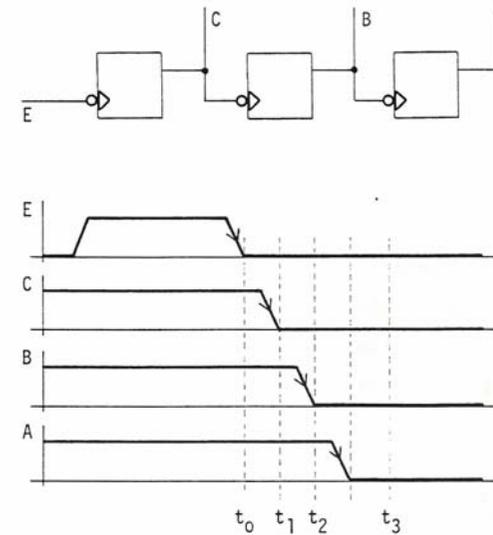
### 11.1 - INTRODUÇÃO

A segunda classe de contadores compreende os síncronos, isto é, aqueles cujos FF's comutam simultaneamente, não ocorrendo combinações intermediárias indesejáveis durante as transições de estado.

É bom que se diga, que na prática ocorrem com muita frequência, como podemos notar neste capítulo, contadores que combinam as modalidades síncrona e assíncrona, com alguns FF's não comutando em sincronismo. São os casos dos CI's 7490 e 7492 analisados aqui.

Outra vantagem bastante importante dos contadores síncronos sobre os assíncronos é quanto à resposta em frequência.

Os contadores assíncronos, comutam seqüencialmente, isto é, o clock, que é aplicado somente ao 1º FF, tem condições de comutar apenas esse; o 2º FF é então comutado pela variação de tensão na saída do primeiro, que se dá com um certo atraso em relação ao clock, devido aos atrasos de propagação dos transistores que compõem o FF. O atraso total será a soma dos atrasos individuais de cada FF. O exemplo abaixo esclarece o fato, ao se analisar a transição de 111 para 000 no contador módulo 8:



Pode-se observar que ocorrem 2 estados intermediários: 110 e 100.

Os contadores síncronos, por definição, são aqueles que têm todas as saídas comutando simultaneamente, isto é, não aparecem os estados intermediários do caso anterior.

### 11.2 - CONTADORES PROGRESSIVO MÓDULO 3 - MÉTODO DE ANÁLISE

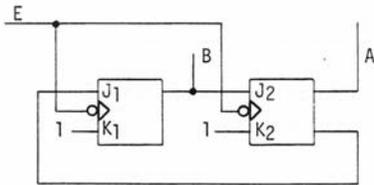
O contador apresentado possui as seguintes peculiaridades:

1) Ambos os FF's que o compõem são sensíveis à borda de descida e estão ligados a um mesmo ponto comum, que é a entrada de contagem. Ambos recebem, simultaneamente, os mesmos pulso de "clock".

2) O primeiro FF (JK) tem a entrada K fixa em nível lógico 1 e a entrada J conectada à saída Q' do segundo FF.

3) Este, por sua vez, acopla a entrada J à saída Q do primeiro, mantendo K fixa em 1.

4) Foram desprezadas as entradas de preset e clear.



Para se realizar a análise de funcionamento do circuito, nunca se deve esquecer que **TODOS** os FF's **COMUTAM SIMULTANEAMENTE**. Portanto, não devemos incorrer no erro de proceder à análise de comutação do primeiro FF e levar em consideração essa comutação na análise do segundo. A comutação é simultânea, portanto, quando a saída do primeiro FF mudar de estado devido à aplicação de um "clock", o segundo FF já estará comutado.

O procedimento para análise é o seguinte:

1) Determinar os estados de todas as entradas de todos os FF's, sem se considerar a aplicação do "clock". Para isso determinamos as equações das entradas:  $J_1 = \bar{A}$ ,  $J_2 = B$ ,  $K_2 = K_1 = 1$ .

2) "Aplicar o "clock" e determinar os estados das saídas em função dos estados de entrada, obtidos no primeiro passo. Para isso, fazemos uso das tabelas da verdade dos FF's envolvidos:

J	K	Q	Q'
0	0	$Q_0$	$Q'_0$
0	1	0	1
1	0	1	0
1	1	$\bar{Q}_0$	$\bar{Q}'_0$

Análise:

	$J_1$	$K_1$	$J_2$	$K_2$	A	B
Condição Inicial					0	0
Entradas antes do 1º clock	1	1	0	1	0	1
Saídas depois do 1º clock					1	0
Entradas antes do 2º clock	1	1	1	1	1	0
Saídas depois do 2º clock					0	1
Entradas antes do 3º clock	0	1	0	1	0	1
Saídas depois do 3º clock					1	0

11.3 - OUTROS MÓDULOS DE CONTAGEM

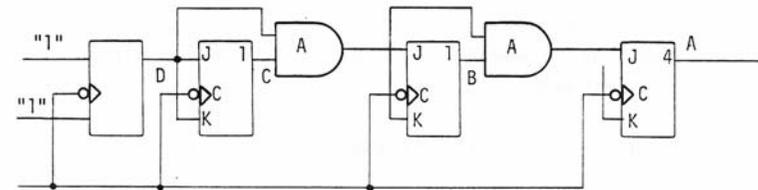
O circuito seguinte é um contador binário, módulo 16, síncrono. Todos os FF's são JK e suas entradas obedecem às equações:

$$J_1 = K_1 = 1$$

$$J_2 = K_2 = D$$

$$J_3 = K_3 = CD$$

$$J_4 = K_4 = BCD$$

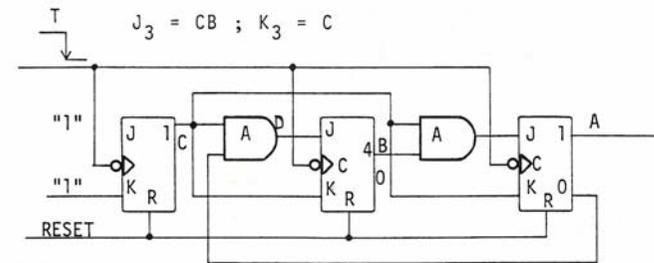


Um contador módulo 6, síncrono, pode ser obtido a partir de 3 FF's JK, onde:

$$J_1 = K_1 = 1$$

$$J_2 = C.\bar{A}; K_2 = C$$

$$J_3 = CB; K_3 = C$$

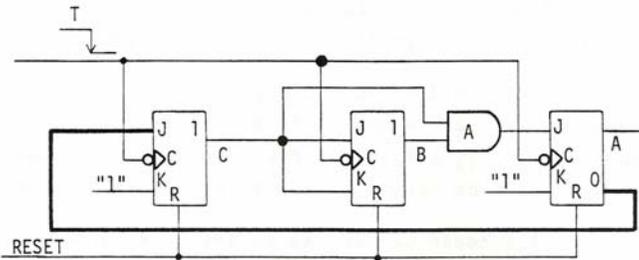


Já um contador módulo 5 implementado com FF's JK é apresentado na figura a seguir onde se observa:

$$J_1 = \bar{A}; K_1 = 1$$

$$J_2 = K_2 = C$$

$$J_3 = B.C; K_3 = 1$$

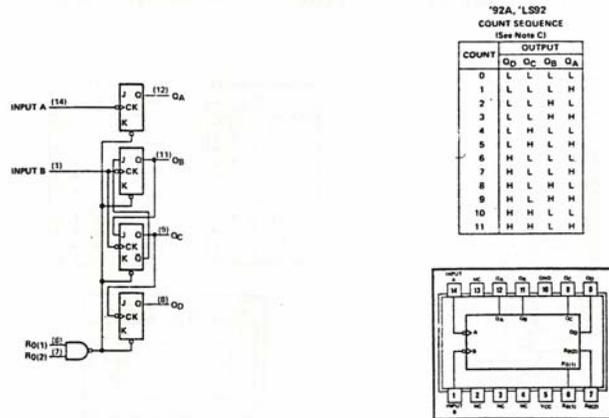


11.4 - CONTADORES COMERCIAIS. TTL

Os CI's 7490, 7492, são excelentes exemplos de contadores TTL (transistor transistor logic). Ambos são elaborados a partir de FF's do tipo master-slave, sensíveis à borda de descida dos pulsos de "clock". Não são contadores puramente síncronos, o que implica em classificá-los no modo assíncrono.

Nos exemplos considerados, dois FF's em cada contador trabalham em sincronismo, isto é, comutam simultaneamente, ao passo que os outros não se comportam assim.

O 7492 é um contador módulo 12, que trabalham no código 6-4-2-1, em que dois FF's (intermediário) são acionados simultaneamente.



Para desempenhar a tabela da verdade apresentada na figura é necessário realizar a ligação entre os pontos Q<sub>A</sub> e IN<sub>B</sub> do contador.

Todos os terminais de CLEAR são ligados em um ponto comum, internamente: a saída de uma porta NE de duas entradas de RESET, que são denominadas R<sub>0</sub>(1) e R<sub>0</sub>(2). É possível zerar o contador mediante aplicação de níveis de tensão convenientes nestas duas entradas.

É preciso tomar cuidado ao se analisar este circuito, pois os 2 métodos de análise anteriormente estudados devem ser aplicados. Em primeiro lugar, verifica-se a influência do pulso de controle sobre o primeiro FF. Depois, é a vez da análise conjunta dos dois FF's seguintes, pelo método síncrono. Finalmente, determina-se o estado de saída do último FF.

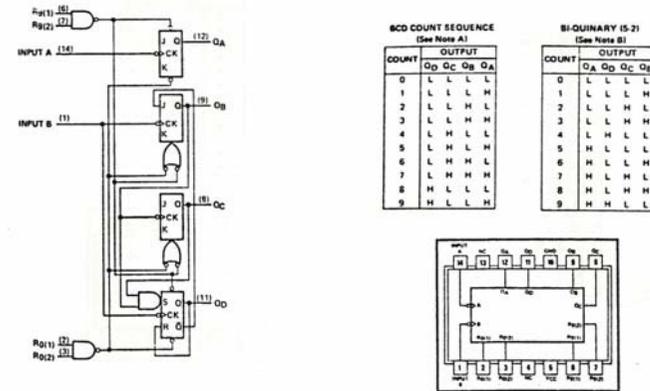
O 7490 utiliza 3 FF's JK e 1 FF RS. Os FF's síncronos são o segundo e o quarto. Portanto, após a análise do primeiro FF se faz a do segundo e quarto concomitantemente. O terceiro FF deve ser deixado por último.

Existem duas opções de contagem: No BCD ou no Bi-quinário.

A contagem BCD é realizada mediante a conexão de saída Q<sub>A</sub> a IN<sub>B</sub>.

Se Q<sub>D</sub> for conectada externamente a IN<sub>A</sub>, a contagem será bi-quinária.

Existem ainda quatro entradas de RESET. Duas delas permitem retornar o contador a zero; as outras duas levam as saídas a indicar o número 9, usadas em aplicações que envolvem complemento de nove BCD.



### 11.5 - DIVISORES DE FREQUÊNCIA

Formados por FF's, os contadores podem dividir frequência por um fator, no máximo, igual ao módulo de contagem do mesmo.

Por exemplo, o 7490 produz, na saída  $Q_D$ , um pulso de tensão a cada 10 pulsos de controle. Portanto, ele realiza uma divisão por 10. Se a saída considerada for  $Q_B$ , a divisão é por 5, pois surgem 2 pulsos para cada 10 entradas. Tudo isso em contagem BCD. Verificando-se as tabelas da verdade dos diversos contadores, pode-se concluir sobre as divisões em cada caso. Mediante realimentações, que modificam os módulos dos contadores, pode-se obter divisões por qualquer número, até um máximo igual ao módulo de contagem. É interessante observar que as divisões podem ocorrer até por números fracionários, como é o caso do 7490, funcionando em contagem bi-quinária, quando a saída considerada é  $Q_B$ : aparecem 4 pulsos nesse terminal para cada 10 aplicados na entrada, donde se deduz que ocorre divisão por 2,5.

### 11.6 - PROBLEMAS RESOLVIDOS

11.6.1 - Analisar o contador síncrono módulo 5 apresentado no item 3.

Solução: O que precisamos para resolver esse exercício é da tabela da verdade FF JK, pois todos são desse tipo, e das equações de entrada dos terminais J e K no circuito:

$$J_1 = \bar{A} ; K_1 = 1$$

$$J_2 = K_2 = C$$

$$J_3 = B.C ; K_2 = 1$$

Realizamos então a análise segundo as considerações expostas:

	J <sub>1</sub>	K <sub>1</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>3</sub>	K <sub>3</sub>	A	B	C
1) Condição Inicial							0	0	0
2) Entradas antes do 1º clock	1	1	0	0	0	1	0	0	1
3) Saídas após o 1º clock							0	0	1
4) Entradas antes do 2º clock	1	1	1	1	0	1	0	1	0
5) Saídas após o 2º clock							0	1	0
6) Entradas antes do 3º clock	1	1	0	0	0	1	0	1	1
7) Saídas após o 3º clock							0	1	1
8) Entradas antes do 4º clock	1	1	1	1	1	1	1	0	0
9) Saídas após o 4º clock							1	0	0
10) Entradas antes do 5º clock	0	1	0	0	0	1	0	0	0
11) Saídas após o 5º clock							0	0	0

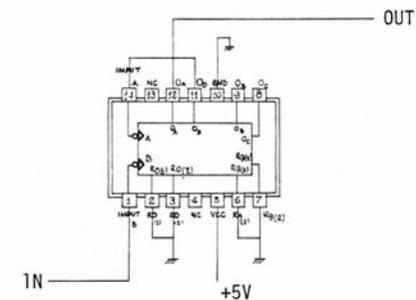
Repetimos aqui a descrição do método utilizado:

- A partir de uma condição das saídas A, B e C determinamos os estados das entradas, com o auxílio das equações de entrada.
- Aplicado o "clock", e consultando-se a tabela da verdade do FF JK, são estabelecidos os novos estados de saída A, B e C.
- E assim por diante.

11.6.2 - Como devem ser ligados os pinos do 7490 para que ele se comporte como um divisor de frequência por 10 e forneça um pulso quadrado na saída?

Resposta:

Consultando-se a tabela da verdade do 7490, observa-se que as saídas capazes de fornecer divisões por 10 são  $Q_D$  e  $Q_C$  em contagem BCD e  $Q_A$  em contagem bi-quinária. Nos dois primeiros casos, as saídas não são "quadradas" pois o sinal permanece a maior parte do tempo em nível lógico baixo. Na opção bi-quinária, a divisão de níveis no tempo é equitativa; metade do tempo em alto e metade em baixo. Portanto, o 7490 deve ser ligado assim:



11.6.3 - Transformar o 7490 em um contador módulo 6 com finalidade de indicação visual, apenas.

Solução:

Basta realimentar os terminais  $Q_B$  e  $Q_C$  para as entradas de RESET  $R_0(1)$  e  $R_0(2)$ , diretamente.

## 11.7 - PROBLEMAS PROPOSTOS

11.7.1 - Analisar o contador módulo 6 do item 3.

11.7.2 - Analisar o funcionamento do 7492, verificando a validade da tabela da verdade apresentada.

11.7.3 - Como se deve ligar o 7492 para que este possa dividir a frequência de dois sinais, independentemente, por 2 e por 6?

11.7.4 - Transformar o 7490 em um divisor por 7.

11.7.5 - Converter o 7492 em um divisor por 8.

## 12 - REGISTROS

12.1 - Introdução

12.2 - Registros em paralelo (LATCHES)

12.3 - Registros de deslocamento série (SHIFT)

12.4 - Registros série-paralelo

12.5 - Problemas Resolvidos

12.6 - Problemas Propostos

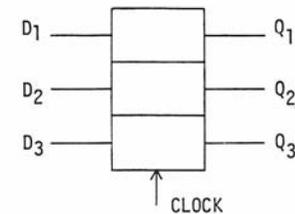
### 12.1 - INTRODUÇÃO

Registros são elementos encontrados onde quer que se queira um armazenamento temporário de bits. Também utilizam os FF's como elementos básicos na sua constituição, pois eles são armazenadores por excelência. Tanto isso é verdade, que o tipo mais simples de registro que pode haver é o constituído de um único FF D.

O bit a ser armazenado é introduzido pela entrada de dados e armazenado mediante a aplicação de um pulso de controle. Aí permanece até que um outro bit venha a ser armazenado, destruindo a informação inicial; ou, sendo a alimentação desligada.

Os registros de mais de um bit dividem-se em duas classes, quanto ao modo de operação: os registros de transferência em paralelo e os registros de deslocamento em série.

Os registros de transferência em paralelo, ou simplesmente registros paralelos, possuem  $n$  terminais de entrada e  $n$  terminais de saída para armazenar  $n$  bits. O armazenamento se dá no momento em que um pulso de clock é aplicado simultaneamente aos FF's.

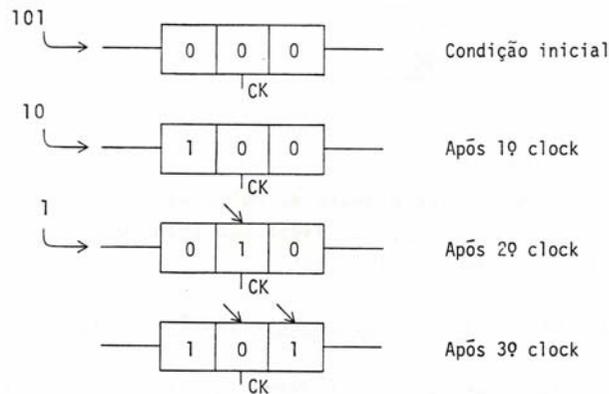


A figura acima representa um registro com capacidade para 3 bits.

Os registros de deslocamento sêrie se comportam daseguinte forma:

Atravês de uma ûnica entrada os bits são introduzidos, um a um, e deslocados atravês do circuito mediante a aplicação dos pulsos de clock.

São necessãrios n pulsos de clock para se armazenar n bits, e outros n-1 pulsos para se realizar a leitura dos bits armazenados. No exemplo abaixo nota-se a açã de armazenamento, num registro de 3 bits, do nûmero binãrio 101.

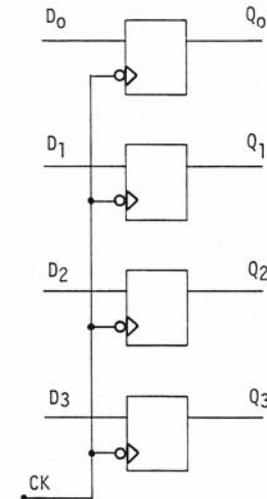


No registro paralelo a vantagem estã na economia de tempo utilizado para o armazenamento: 1 pulso de clock contra 3 pulsos no registro de deslocamento.

A vantagem dos registros de deslocamento estã no nûmero de terminais de entrada e saída, que determina a quantidade de canais para a transmissão dos dados de, e para, o registro. Apenas um canal de entrada, no registro de deslocamento, é mais econômico que os três do registro paralelo.

## 12.2 - REGISTROS EM PARALELO (LATCHS)

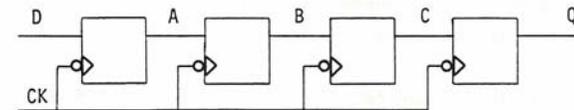
Alguns FF's D associados em paralelo formam o registro de transferência em paralelo, também conhecido por LATCH.



Na borda descendente do pulso de controle (Ck) todos os bits inseridos nas entradas de dados são transportados para as saídas, aĩ permanecendo indefinidamente.

## 12.3 - REGISTROS DE DESLOCAMENTO SÊRIE (SHIFT)

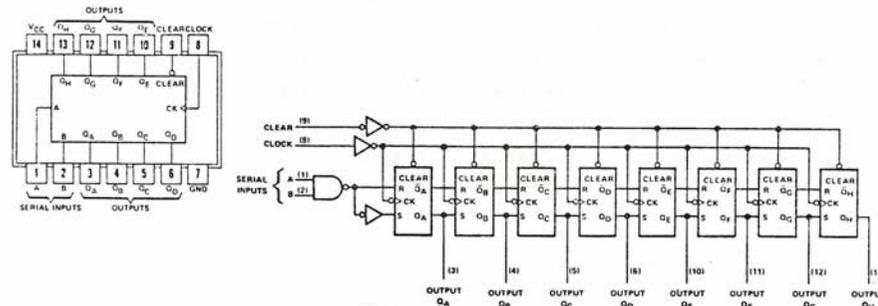
Se os mesmos FF's D forem associados em sêrie, obtemos o registro de deslocamento sêrie, como o apresentado a seguir:



A cada pulso de clock, os estados dos pontos D, A, B e C são transferidos, respectivamente, para A, B, C e Q. Ocorre, portanto, um deslocamento de bits no interior do registro.

O 74164 é um bom exemplo TTL de registro de deslocamento de 8 bits. Nele, dispomos de todos os terminais de saída dos vários FF's RS que o compõe.

O diagrama em blocos do mesmo aparece na figura abaixo:



FUNCTION TABLE		INPUTS		OUTPUTS							
CLEAR	CLOCK	A	B	Q <sub>A</sub>	Q <sub>B</sub>	...	Q <sub>n</sub>	Q <sub>n+1</sub>	Q <sub>n+2</sub>	...	Q <sub>h</sub>
L	X	X	X	L	L	...	L				L
H	L	X	X	Q <sub>A0</sub>	Q <sub>B0</sub>	...	Q <sub>n0</sub>	Q <sub>n1</sub>	Q <sub>n2</sub>	...	Q <sub>h1</sub>
H	T	H	H	H	Q <sub>A1</sub>	...	Q <sub>n1</sub>	Q <sub>n2</sub>	Q <sub>n3</sub>	...	Q <sub>h2</sub>
H	T	L	X	L	Q <sub>A2</sub>	...	Q <sub>n2</sub>	Q <sub>n3</sub>	Q <sub>n4</sub>	...	Q <sub>h3</sub>
H	T	X	L	L	Q <sub>A3</sub>	...	Q <sub>n3</sub>	Q <sub>n4</sub>	Q <sub>n5</sub>	...	Q <sub>h4</sub>

A análise desse sistema se fez a seguir:

A entrada  $\overline{A.B}$  é aplicada ao terminal de Reset do 1º FF e seu complemento  $\overline{AB}$  à entrada Set. Portanto, se ambas as entradas estiverem em nível lógico alto,  $\overline{AB}$  é 1, ou seja, está se introduzindo o bit 1 no registro. Para outras combinações de A e B tudo se passa como se o bit 0 estivesse sendo armazenado. Os bits podem se deslocar no interior do sistema graças às ligações entre as saídas  $\overline{Q_n}$  e  $Q_n$  e as entradas R e S do FF seguinte:

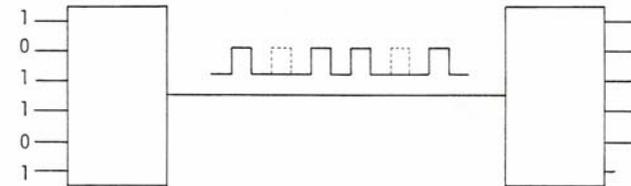
Os circuitos dos FF's são sensíveis à borda de descida dos pulsos de clock, porém, devido à presença do inversor que alimenta esses terminais de controle, é a borda de subida dos pulsos que provoca os deslocamentos.

O terminal CLEAR pode ser acionado por um nível lógico baixo, como se pode deduzir observando o esquema.

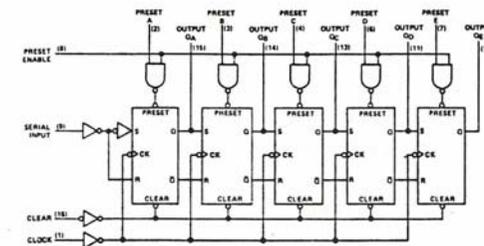
Oito impulsos de controle são suficientes para armazenar oito bits, e se pode dispor de todos os estados de saída, simultaneamente, através das oito saídas do CI correspondentes às saídas individuais dos FF's.

12.4 - REGISTROS SÉRIE-PARALELO

São muito utilizadas em Eletrônica Digital as conversões de dados da forma série a paralela ou paralela a série. Por exemplo: na transmissão de dados entre dois sistemas que trabalham com dados em paralela, é anti-econômico o uso de transmissão paralela pois precisaríamos de n canais de transmissão, sendo n o número de bits que compõem um byte. Torna-se viável, então, transformar os bits da forma paralela em série, transmiti-los um a um, e efetuarmos a recomposição do byte com os bits em paralelo.



O CI 7496 permite ambos os tipos de conversão, trabalhamos com 5 bits.



Façamos a análise deste CI para as 4 modalidades de funcionamento permitidas.

1) Entrada série-saída série:

A utilização do SERIAL INPUT como entrada e  $Q_E$  como saída, possibilita o uso de 7496 funcionando como registro de deslocamento.

2) Entrada paralela-saída paralela:

Para isso deve-se fazer uso das entradas PRESET A, B, C,

D e E e das saídas  $Q_a$ ,  $Q_b$ ,  $Q_c$ ,  $Q_d$  e  $Q_e$  e do terminal PRESET ENABLE (habilitadora de preset) que deve ser ativado (1).

3) Entrada sêrie-saída paralela:

O armazenamento é efetuado no modo deslocamento, pela aplicação de 5 impulsos de clock e o byte pode então ser lido nas saídas de  $Q_a$  a  $Q_e$ .

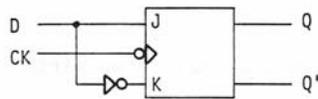
4) Entrada paralela-saída sêrie:

Ativando-se o terminal PRESET ENABLE, os dados são armazenados via PRESET A a E; após, desativando-se PRESET ENABLE, pode-se retirar os dados em sêrie pela saída  $Q_e$  mediante a aplicação de alguns impulsos de "clock".

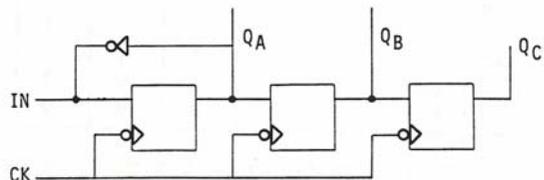
12.5 - PROBLEMAS RESOLVIDOS

12.5.1 - Como devem ser ligados as entradas de um FF JK para que ele possa desempenhar o papel de um armazenador de dados?

Solução: Colocando-se uma porta inversora entre as entradas e utilizando J como entrada de dados.



12.5.2 - Quais serão os estados lógicos das saídas do registro abaixo depois de 4 pulsos de clock?



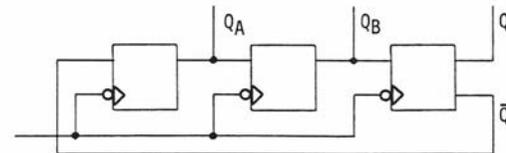
Solução: Supondo condição inicial  $Q_a = Q_b = Q_c = 0$  para saídas e efetuando-se os deslocamentos:

$$\begin{aligned} \bar{Q}_a &\rightarrow Q_a \\ Q_a &\rightarrow Q_b \\ Q_b &\rightarrow Q_c \end{aligned}$$

podemos concluir que os estados assumidos na saída serão os seguintes:

	Q	Q	Q
1. Condição inicial	0	0	0
2. Após o 1º Clock	1	0	0
3. Após o 2º Clock	0	1	0
4. Após o 3º Clock	1	0	1
5. Após o 4º Clock	0	1	0

12.5.3 - O circuito abaixo apresenta um tipo especial de "seqüencial". Qual é ele?



Solução: Fazemos a análise do circuito, utilizando as relações de deslocamento.

$$\begin{aligned} \bar{Q}_c &\rightarrow Q_a \\ Q_a &\rightarrow Q_b \\ Q_b &\rightarrow Q_c \end{aligned}$$

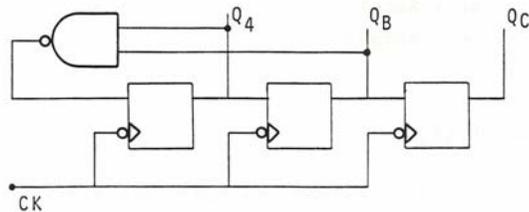
	$Q_a$	$Q_b$	$Q_c$
Condição inicial	0	0	0
1º Clock	1	0	0
2º Clock	1	1	0
3º Clock	1	1	1
4º Clock	0	1	1
5º Clock	0	0	1
6º Clock	0	0	0

A tabela acima evidencia o tipo de seqüencial que se obtêm.

12.6 - PROBLEMAS PROPOSTOS

12.6.1 - Esquematizar um registro de transferência em paralelo, de 3 bits, com FF's JK.

12.6.2 - Determinar as condições de saída do registro de deslocamento abaixo após 5 pulsos de clock.



12.6.3 - Como devem ser conectados os terminais do 7496 para que ele produza um efeito seqüencial em que:

- Apenas uma saída, entre todas, assuma o valor 1, a cada novo estado.
- Esse nível alto percorra ciclicamente as saídas, da forma  $Q_a \rightarrow Q_b \rightarrow Q_c \rightarrow Q_d \rightarrow Q_e \rightarrow Q_a$  etc?

13 - SOMADORES

- 13.1 - Introdução
- 13.2 - Semi Somador
- 13.3 - Somador Completo
- 13.4 - Somador Paralelo
- 13.5 - Somador Série
- 13.6 - Subtrador
- 13.7 - Problemas Resolvidos
- 13.8 - Problemas Propostos

13.1 - INTRODUÇÃO

As operações aritméticas podem ser efetuadas por um sistema digital graças a um elemento conhecido como somador completo.

Não somente a operação de adição pode ser efetuada com esse circuito, mas também, as operações de subtração, multiplicação, divisão, etc.

Isso se torna possível graças ao fato de a subtração ser realizada pela soma de um número com o complemento de outro, como se verá visto mais adiante. A multiplicação não passa de um processo que envolve sucessivas somas e deslocamentos, e a divisão, sucessivas subtrações (somas de complementos) e deslocamentos. Portanto, qualquer operação aritmética recai na adição, tornando-se o somador um elemento de máxima importância.

13.2 - SEMI-SOMADOR

É um elemento que adiciona apenas 2 bits, dando como resultado outros 2: a Soma e a Transposição.

Fazendo-se uso da tabuada da adição binária, podemos elaborar a tabela da verdade do semi-somador, que possibilitará a realização do circuito.

$$\begin{array}{r} 0 + 0 = 00 \\ 0 + 1 = 01 \\ 1 + 1 = 10 \\ 1 + 1 + 1 = 11 \end{array}$$

A operação  $A + B = TS$ , é representada na tabela onde A e B são variáveis de entrada e T e S as de saída. S é o Bit "soma" e T é o bit "transposição".

A
+ B
---
T S

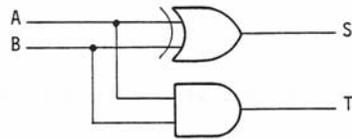
A	B	T	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

As equações correspondentes a S e T ficam, pois:

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$T = A.B$$

É o circuito combinacional que as realiza, ou seja, semi-somador, é representado a seguir.



É importante notar que o semi-somador soma apenas 2 bits, sendo impotente para a realização da soma de 2 números binários de n dígitos. Para tanto, precisamos de somador completo.

### 13.3 - SOMADOR COMPLETO

Suponhamos que se deseje realizar a soma do número A, de n bits, com outro, B, também de n bits:

$$\begin{array}{r}
 \text{n}^\circ \text{ A} \quad \text{T}_{n-1} \quad \text{T}_{i-1} \quad \text{T}_0 \\
 \text{An} \dots \text{Ai} \dots \text{A1} \text{ A}_0 \\
 \text{n}^\circ \text{ B} \quad \text{Bn} \dots \text{Bi} \dots \text{B1} \text{ B}_0 \\
 \hline
 \text{Tn Sn} \quad \text{Si} \quad \text{S1} \text{ S}_0
 \end{array}$$

Como podemos observar, a soma se compõe de sucessivas operações do tipo  $T_{i-1} + A_i + B_i = T_i S_i$ .

O circuito que possibilita tal operação binária é o somador completo. A tabela da verdade de tal circuito é dada abaixo:

T <sub>i-1</sub>
A <sub>i</sub>
B <sub>i</sub>
---
T <sub>i</sub> S <sub>i</sub>

A <sub>i</sub>	B <sub>i</sub>	T <sub>i-1</sub>	T <sub>i</sub>	S <sub>i</sub>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

O equacionamento das variáveis de saída resulta em:

$$T_i = \bar{A}B T_{i-1} + A\bar{B} T_{i-1} + AB \bar{T}_{i-1} + AB T_{i-1}$$

$$T_i = T_{i-1} (\bar{A}B + A\bar{B}) + AB (\bar{T}_{i-1} + T_{i-1})$$

Simplificando e escrevendo em função das variáveis de saída do semi-somador:

$$T_i = T_{i-1} . S + T$$

$$S_i = \bar{A}\bar{B} T_{i-1} + \bar{A}B \bar{T}_{i-1} + A\bar{B} \bar{T}_{i-1} + AB T_{i-1}$$

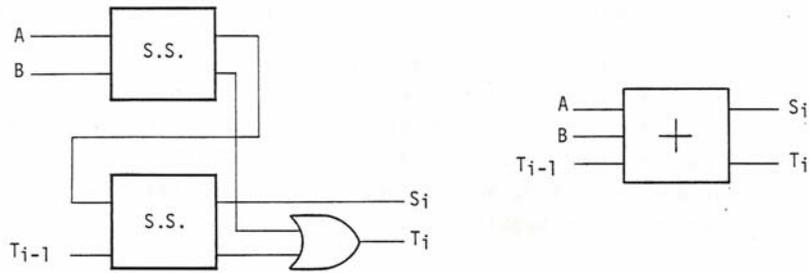
$$S_i = T_{i-1} (\bar{A}\bar{B} + AB) + \bar{T}_{i-1} (\bar{A}B + A\bar{B})$$

Pode-se provar que  $\bar{A}\bar{B} + AB = \bar{S}$  então:

$$S_i = T_{i-1} . \bar{S} + \bar{T}_{i-1} . S$$

$$S_i = T_{i-1} \oplus S$$

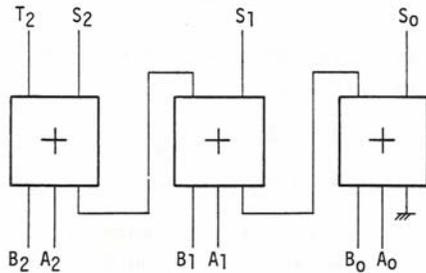
Dois circuitos semi-somadores e mais uma porta são suficientes para compor o somador completo:



Essa unidade, como vimos, tem condições de adicionar apenas dois bits mais uma transposição, não sendo possível ainda a realização da soma dos números A e B como é nosso objetivo. Para tanto, é preciso incluir dispositivos adicionais ao somador completo, dando origem aos sistemas analisados a seguir.

13.4 - SOMADOR PARALELO

Associando-se em paralelo n somadores completos, a soma é imediata, pois as transposições são transferidas quase que instantaneamente.



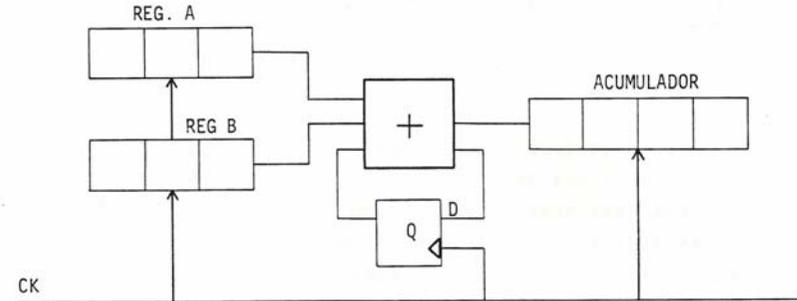
A vantagem deste sistema é a rapidez com que é efetuada a operação. Logo que os bits de entrada são inseridos em seus respectivos terminais, aparecem os resultados nas saídas S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub> e T<sub>2</sub>.

13.5 - SOMADOR SÉRIE

Com apenas um somador completo, utilizado repetidas vezes, também é possível realizar a operação, fazendo-se uso de regis

tros de deslocamento.

A cada "clock" aplicado aos registros, ocorre o deslocamento de dois bits para o interior do somador completo, um do nº A e o outro do nº B, sendo então processados. A soma é deslocada para o interior do registro de deslocamento chamado ACUMULADOR e a transposição é guardada pelo FF D, que se comporta como um retardador, até que chegue a vez dos bits seguintes serem processados.



Comercialmente, entre outros, existe o CI 7480, TTL, que realiza a função de somador completo. O Lay-out e a tabela da verdade de desse circuito são apresentados a seguir.

A terminologia adotada é um pouco diferente da nossa, o que não altera em nada o funcionamento desse componente, como é obvio.

Observe-se também que as entradas A e B são dadas por uma combinação entre outras variáveis.

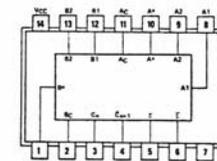
$$A = \bar{A}C + \bar{A}^* + A_1.A_2$$

$$B = \bar{B}C + \bar{B}^* + B_1.B_2$$

Quando A\* é usada como entrada, A<sub>1</sub> ou A<sub>2</sub> deve ser baixa. Quando B\* é usada como entrada, B<sub>1</sub> ou B<sub>2</sub> deve ser baixa.

Quando A<sub>1</sub> e A<sub>2</sub> ou B<sub>1</sub> e B<sub>2</sub> são usadas como entradas, A\* ou B\*, respectivamente, devem permanecer abertas.

INPUTS			OUTPUTS		
C <sub>n</sub>	B	A	C <sub>n+1</sub>	S	T
L	L	L	H	H	L
L	L	H	H	L	H
L	H	L	H	L	H
L	H	H	H	L	L
H	L	L	H	L	H
H	L	H	H	L	L
H	H	L	H	L	L
H	H	H	H	L	L



## 13.6 - SUBTRADOR

Seja subtrair o número binário 00110 (6) de 01110 (14). Uma forma de se realizar tal operação é somando o segundo byte com o complemento do primeiro:

$$\begin{array}{r} 14 \quad 01110 \\ \bar{6} \quad 11001 \\ \hline 100111 \end{array} +$$

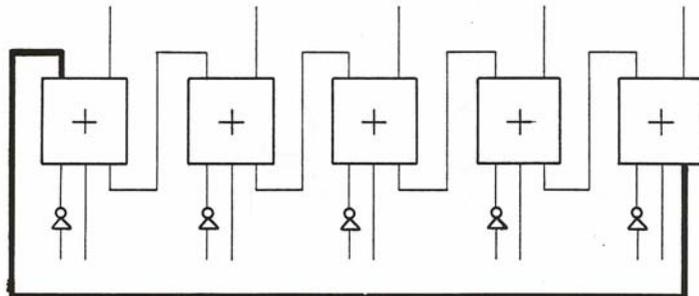
Essa operação pode ser feita facilmente com um sistema como o somador paralelo de 5 bits. Porém, nota-se que o resultado não é correto pois  $14 + 6$  não é  $39$  !!

O ajuste necessário é realizado de forma extremamente simples: soma-se o bit mais significativo do resultado ao próprio, resultado, por um processo de realimentação.

$$\begin{array}{r} \textcircled{1} \quad 00111 \\ \quad \quad \quad 1 \\ \hline 01000 \end{array} +$$

Ignorando-se o bit mais significativo, aquele que foi realimentado, o resultado torna-se o esperado ( $1000 = 8$ ).

Operacionalmente, a realimentação é simples, como se observa na figura seguinte:



## 13.7 - PROBLEMAS RESOLVIDOS

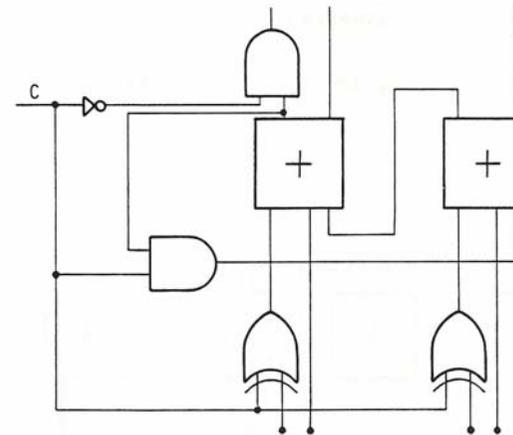
13.7.1 - Modificar o circuito do item anterior de modo a convertê-lo em um subtrador/somador, comandado por um pulso de controle, que determina a sua função.

Solução:

As duas diferenças entre os circuitos somador e subtrador são: 1º) a realimentação da última para a primeira transporção que ocorre no subtrador e, 2º) as entradas invertidas da variável B.

Para resolver o primeiro problema, utilizamos duas portas E, comandadas pelo controle C, que direcionam a última transposição para a saída (somador) ou primeira transposição (subtrador).

O segundo problema é resolvido com o auxílio de portas exclusivamente OU: Conectando uma das entradas ao controle C, a saída indicada o estado invertido ou não invertido da outra entrada, conforme o estado de C. (Para simplificar, considera-se apenas 2 blocos somadores).



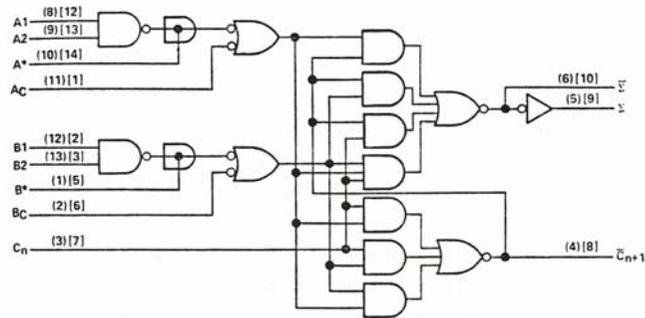
C = 1 subtrador

C = 0 somador

## 13.8 - PROBLEMAS PROPOSTOS

13.8.1 - Transformar um somador sêrie em um subtrador.

13.8.2 - Provar que o 7480 é um somador completo, através da análise lógica dos seus circuitos.



## 14 - MEMÓRIAS MAGNÉTICAS

14.1 - Generalidades

14.2 - Memória de Tambor Magnético

14.3 - Memória de Núcleos

14.4 - Matrizes de Coincidência

14.5 - Memórias de Discos Magnéticos

14.6 - Memória de Fitas Magnéticas

Em muitos sistemas digitais, em especial nos computadores, um estágio de máxima importância é o que tem por finalidade o armazenamento dos dados binários envolvidos no processamento; é a MEMÓRIA do sistema. Vários são os tipos de memória encontradas em sistemas digitais. Por exemplo, existem a memória de fita e cartão perfurados, de núcleos magnéticos e de semicondutores. Dependendo do sistema é mais vantajoso o emprego de um ou outro meio de armazenamento, visto que eles diferem quanto a certas características que passamos a analisar em seguida.

Os conceitos aos quais nos referimos são válidos para todos os tipos de memórias existentes.

O armazenamento é permanente, quando os bits armazenados permanecem na memória, quando o fornecimento de energia elétrica cessa. É volátil, quando a falta de alimentação destrói a informação armazenada. Muitos computadores possuem fonte de alimentação "no break" operando com acumuladores que entram imediatamente em operação no caso de faltar energia da rede, evitando a perda de dados pelas memórias voláteis.

Capacidade de uma memória é a quantidade máxima de bits que ela pode armazenar. Utilizam-se os múltiplos quilobite e megabit para indicar a capacidade de memória. 1 kilobit equivale 1.024 bits, ou  $2^{10}$  bits. Por exemplo, em valor típico para um tambor magnético médio é 500 quilobits.

O tempo de acesso é definido como o intervalo de tempo necessário para se efetuar uma escrita ou leitura na memória. O tempo de acesso para uma memória de fita magnética é consideravelmente maior que uma de núcleos, pois a informação armazenada em uma fita necessita que se percorra a fita até o ponto em que a informação está

armazenada, e isso leva tempo.

Seqüencial é a memória, como a fita magnética, cujo acesso a uma determinada informação se faz "passando por cima" de outras. O acesso aleatório vai diretamente à informação requerida. Para que se possa ler uma informação, é fornecido um endereço à memória e esta procura o bit, ou conjunto de bits (byte), endereçados. A memória seqüencial, obviamente, deve possuir um tempo de acesso maior que a aleatória pois a procura do dado é mais rápida nesta última.

#### 14.2 - MEMÓRIA DE TAMBOR MAGNÉTICO

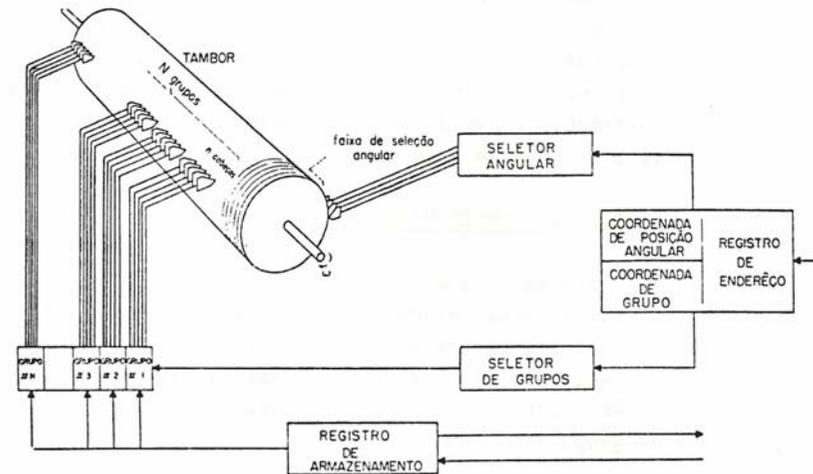
Fixo por um eixo perpendicular às suas faces paralelas, sobre o qual gira, o tambor magnético é um dispositivo de armazenamento revestido por uma película magnética capaz de gravar um bit pelo sentido da imantação produzida em sua superfície. As células de armazenamento são endereçadas através de suas coordenadas: uma angular e outra de grupo.

O tambor gira constantemente com velocidade angular fixa, ficando todas as suas informações disponíveis à grupos de  $n$  cabeças magnéticas gravadoras leitoras, alinhadas ao longo do cilindro. Em uma faixa no extremo do tambor, são gravadas as informações que permitem a seleção angular, constituindo-se de diversas linhas que contêm a gravação, em código, de cada posição angular da memória. Quando se deseja ler ou escrever qualquer informação no tambor, deve-se proporcionar ao sistema a coordenada angular de posição da memória onde vai se efetuar a operação e o seletor angular se responsabiliza por encontrar a referida posição, realizando a leitura das diversas linhas na extremidade do tambor, até que sua informação coincida com a coordenada fornecida.

Além da seleção angular, a memória de tambor dispõe de um seletor de grupo que determina qual o grupo de cabeças magnéticas que deve entrar em operação no processo de escrita/leitura. Isso é conseguido através de um seletor de grupo que age de acordo com a coordenada de grupo fornecida ao sistema.

A leitura é realizada como nas reproduções magnéticas comuns pois, enquanto o tambor gira, as variações de campo magnético induzem em um enrolamento um impulso de corrente, que depois de amplificado e remodelado, é aproveitado no processamento. A escrita

também não tem segredo, constituindo-se na aplicação de um impulso no mesmo enrolamento, que magnetiza a película. A polaridade do impulso determina se o bit a ser gravado é 0 ou 1.



Alguns dados típicos para um tambor magnético são os seguintes:

Tempo de acesso:	5 ms
Dimensões:	25 cm de comprimento 20 cm de diâmetro
Capacidade:	600 quilobits

#### 14.3 - MEMÓRIA DE NÚCLEOS

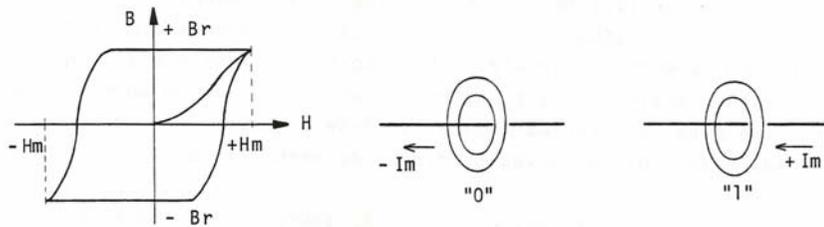
Um condutor percorrido por corrente elétrica gera um campo magnético ao seu redor. Esse é um princípio conhecido em Física. Um material magnético colocado nas proximidades do condutor pode ter suas características modificadas pelo campo gerado, tornando-se um pequeno ímã, cuja polaridade é determinada pelo sentido da corrente que lhe deu origem. Esse pequeno ímã conserva a informação por tempo indefinido, constituindo-se em um armazenador de informações binárias; a memória de núcleos magnéticos.

Os núcleos normalmente aparecem na forma de anéis pelo interior do qual passam fios condutores responsáveis pela escrita e leitura na memória.

Dependendo do sentido da corrente através do núcleo a informação 0 ou 1 é armazenada.

Uma vez cessada a corrente de escrita, o que equivale a se retirar o campo magnético que envolve o núcleo, este adquire uma indução remanescente que a i se conserva até que uma corrente em sentido contrário consiga gerar um campo suficiente forte para inverter a polaridade do imã.

Os materiais indicados são os que apresentam um laço de histerese o mais "quadrado" possível, como o apresentado abaixo.



Supondo o Núcleo inicialmente desmagnetizado, fazemos fluir pelo seu interior a corrente  $+I_m$ , capaz de provocar um campo magnetizante  $+H_m$ . A indução do núcleo atinge o valor de saturação  $B_{sat}$  e quando o campo magnetizante é retirado, como se nota no gráfico, o núcleo conserva o valor  $+B_r$  que é uma indução remanescente e se deve às propriedades do núcleo. Diz-se que o mesmo armazena o bit 1.

Ao se fazer passar uma corrente em sentido contrário de intensidade maior ou igual a  $-I_m$ , que provoque o aparecimento de um campo magnetizante  $-H_m$  satura-se o núcleo em sentido contrário ao inicial e este passa a contar com uma indução remanescente  $-B_r$ , ou seja, armazena o bit 0.

Nota-se que, se o campo magnetizante não atingir o valor mínimo  $+H_m$ , neste último caso, ou  $-H_m$ , no primeiro, o núcleo não tem suas características modificadas.

A leitura de um núcleo magnético é feita através de um outro fio que passa pelo seu interior. Faz-se passar pelo fio de escrita uma corrente capaz de zerar o núcleo. Se o estado anterior era 1, ocorre uma variação no sentido de magnetização do núcleo, ou seja

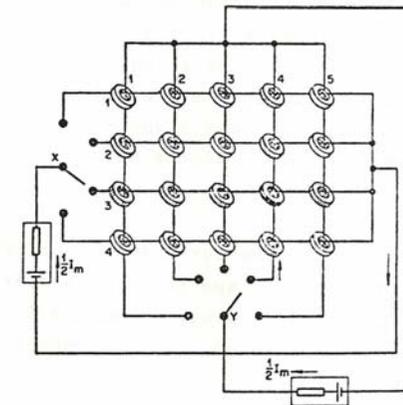
o campo magnético varia nesse instante. A variação de campo induz no fio de leitura um impulso que é então processado por outras unidades. Se o estado do núcleo era zero, não ocorre variação de campo, logo, não aparece impulso algum induzido no fio de leitura.

#### 14.4 - MATRIZES DE COINCIDÊNCIA

O modo de endereçamento é realizado pela passagem de correntes coincidentes endereçados a um determinado núcleo da matriz.

A matriz é um conjunto de núcleos alinhados no sentido horizontal e vertical pelos quais passam fios de escrita/leitura. Os fios são denominados X e Y e servem como coordenadas para identificar o núcleo na matriz. Por eles passam correntes iguais a  $1/2 I_m$ , insuficientes, sozinhas, para magnetizar um núcleo. Porém, um dos núcleos, como se pode observar na figura seguinte, sofre a ação de campo provocado pela soma das meias-correntes de magnetização X e Y.

Várias dessas matrizes são sobrepostas, formando o conjunto de armazenamento completo.



A seguir são enumeradas algumas características de núcleos magnéticos de FEROCUBE 6E1.

Diâmetro interno: 2,2 mm  
 Corrente de excitação: 350 mA  
 Tensão de saída (leitura): 37 mV  
 Tempo de comutação: 12 us.

#### 14.5 - MEMÓRIAS DE DISCOS MAGNÉTICOS

O disco magnético tem forma de um disco comum, sendo re coberto por uma película de material magnético. É uma memória de aces so aleatório pois um braço de leitura provido de uma cabeça magnēti ca procura a pista onde a informação deve ser gravada ou lida, em fun ção de uma coordenada ou endereço fornecido em um tempo relativamen te curto (quando comparado às fitas magnéticas, suas principais concor rentes) da ordem de 6 ms.

O disco gira a uma velocidade constante e seu mecanismo de escrita e leitura é semelhante ao do tambor magnético.

#### 14.6 - MEMÓRIA DE FITA MAGNÉTICA

As fitas magnéticas constituem elementos importantes de memorização, pois tem uma capacidade bastante alta. Ademais, os modernos cassetes são bastante versáteis e podem ser acoplados a um sistema quando se deseja realizar uma escrita ou leitura de maneira bastante fácil. É comum encontrar-se bibliotecas de programas conten do rolos e mais rolos de fitas magnéticas.

A desvantagem de tal dispositivo reside no fato de que são de acesso seqüencial, portanto, lentos. Isso se explica pelo fato de que as informações são enroladas umas sobre as outras o que exige sejam desenroladas para se buscar uma outra informação.

O processo de escrita e leitura em fitas magnéticas difere muito pouco do encontrado em gravadores de áudio convencionais.

A título ilustrativo, fornecemos um quadro comparativo de características entre memórias de fita magnética e disco, na versão moderna conhecida por diskette, pequeno disco magnético flexível:

	diskette	cassete	cartucho
Capacidade (Kb)	800	720	2870
Ritmo de transferência Kb/s	500	24	48
Densidade de gravação b/pol	6400	800	1600
Tempo médio de acesso	236ms	20s	20s

#### 15 - MEMÓRIAS DE SEMICONDUTOR

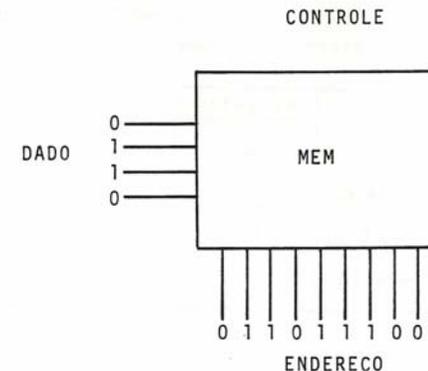
- 15.1 - Introdução
- 15.2 - Técnicas de Implementação
- 15.3 - RAM's
- 15.4 - ROM's e PROM's
- 15.5 - EPROM's
- 15.6 - Um Circuito simples de Escrita/Leitura

##### 15.1 - INTRODUÇÃO

As memórias implementadas com semicondutores, geralmente em LSI (alta escala de integração), se apresentam na forma de CI's como os dos exemplos anteriores. São basicamente constituídas por FF's, cada qual encarregado de armazenar um bit de informação, ou simplesmente por transistores ou diodos que se encontram em um ou outro estado de operação: conduzindo ou cortado.

Para se escrever, ou ler, em uma memória, necessário se faz introduzir um endereço no sistema para se determinar o local onde se realizará a operação. Os endereços são combinações binárias em correspondência biunívoca com as células que armazenam os bits, ou os bytes. A quantidade máxima de bits, ou bytes, que se pode armazenar é a capacidade da memória. Por exemplo, a memória com capacidade igual a  $512 \times 4$  pode, no máximo conter 512 bytes, sendo que cada byte, nesse caso, deve ser composto por 4 bits. Um endereço de qualquer célula dessa memória é uma outra palavra de 9 bits, pois as combinações necessárias para corresponder às 512 células devem ser em igual número, ou seja,  $2^9$ .

O procedimento para escrita é exemplificado na figura seguinte, onde se realiza o armazenamento do byte 0110 no endereço 011011100.



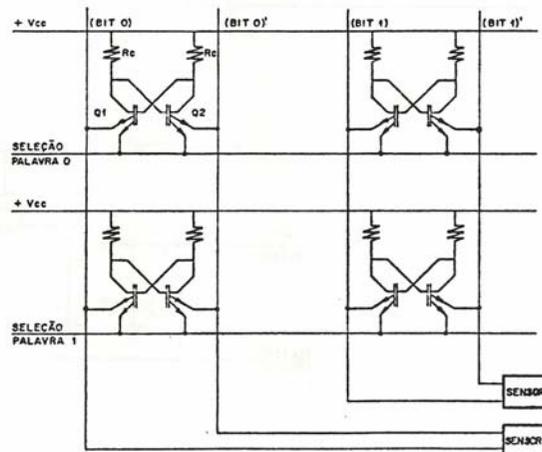
Existem duas grandes classes de memórias de semicondutores: as de ESCRITA E LEITURA e as SOMENTE DE LEITURA. As primeiras são conhecidas como RAM's (random access memory) ou RWM's (read write memory) e são voláteis. Podemos escrever ou ler nesses dispositivos com relativa facilidade, porém, se a alimentação é interrompida, o conteúdo se perde.

As segundas são as ROM's (read only memory) ou memórias somente de leitura. São acessíveis à leitura de seus conteúdos como as RAM's, mas o método de gravação é especial e não são voláteis. Algumas não podem ser apagadas, outras, são alteráveis por processos especiais. Todas elas, no entanto, não são voláteis.

### 15.2 - TÉCNICAS DE IMPLEMENTAÇÃO

As memórias de semicondutores podem ser implementadas com transistores bipolar ou MOS.

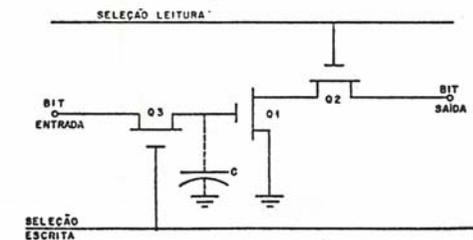
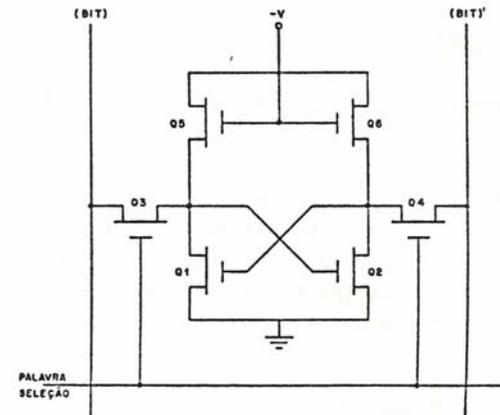
As memórias bipolares constituem-se de células armazenadoras com capacidade para um bit, semelhante aos multivibradores biestáveis, porém mais simples em relação às polarizações dos transistores. Observa-se também a utilização de transistores multiemissor, largamente feita em circuitos integrados. A célula básica aparece à esquerda na figura seguinte:



Um transistor conduz; o outro corta. Esse é o estado estável do multivibrador. Suponhamos que Q<sub>1</sub> conduza, o que se convencionou

estar a célula armazenando o bit 1. A linha "seleção de palavra" é mantida em potencial baixo, permitindo a passagem da corrente por Q<sub>1</sub>. Para se fazer a leitura, enviamos à essa linha um potencial positivo, bloqueando a corrente e forçando-a a passar pela linha S que excita o sensor correspondente indicando que a célula armazenava o "1" lógico. Caso fosse Q<sub>2</sub> o transistor a conduzir, a excitação seria em S' o que evidenciaria o estado inverso da célula.

Com transistores MOS, também sob a forma de biestáveis, o processo de armazenamento é o mesmo: um transistor permanece em condução e o outro em corte.



Forçando-se os transistores Q<sub>3</sub> e Q<sub>4</sub> à condução, através de um potencial positivo aplicado aos "gates" de ambos, as informações guardadas por Q<sub>1</sub> e Q<sub>2</sub> são transferidas para os terminais de leitura B e B'. Nota-se aí que as cargas dos transistores são outros dois transistores Q<sub>5</sub> e Q<sub>6</sub> cujas polarizações fazem com que elas se comportem como resistores.

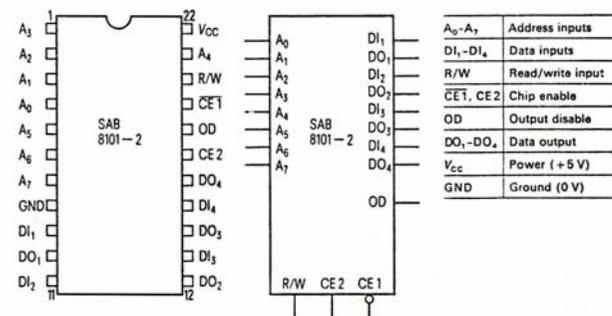
Um modo mais econômico de se implementar memórias com transistores MOS é aproveitando a capacitância "gate-source" inerente a esses tipos de dispositivos. O transistor Q<sub>1</sub> da figura anterior, à direita, é cortado ou saturado em função da carga da citada capacitância. Como a sua descarga poderia provocar a perda de informação, ela é recarregada periodicamente por um processo rápido de leitura/escrita na mesma célula, sendo o sinal de saída amplificado e reentroduzido pela entrada. A leitura é feita pela aplicação de um potencial a Q<sub>2</sub>, que coloca o terminal de saída à massa ou o deixa flutuando em função do estado de Q<sub>1</sub>. Cada um desses estados é interpretado como um bit. A escrita se faz via Q<sub>3</sub>; quando conduz, o potencial aplicado à entrada é utilizado na carga da capacitância "gate-source". O processo de regeneração é realizado em intervalos muito curtos, da ordem de milissegundos. Esse tipo de memória é conhecido como DINÂMICA.

### 15.3 - RAM's

As RAM's, como já vimos, são memórias de escrita e leitura que podem ser requisitadas a qualquer momento para essas operações, porém, são voláteis. As memórias analisadas no item anterior pertencem a esse grupo. A falta de alimentação destrói as informações. São muito úteis no transcorrer de um processamento pois servem para armazenar dados em grande quantidade, resultados parciais e mesmo finais das diversas operações realizadas.

Certos sistemas utilizam fontes de alimentação "no break", que ao menor sinal de falha no fornecimento de energia elétrica, põe em operação acumuladores que sustentam as operações do sistema por um tempo determinado, evitando a "fuga" de informações.

Como exemplo prático desse tipo de memória, faremos referência ao CI SAB8101, uma RAM implementada com tecnologia MOS Estática.



Eis algumas características suas:

É organizada na forma 256 x 4 bits, isto é armazena 256 bytes de 4 bits, necessitando pois de 8 pinos para endereçamento ( $2^8 = 256$ ).

Possui um tempo de acesso de 350 ns.

É alimentada por uma tensão de 5V, sendo portanto compatível com a lógica TTL.

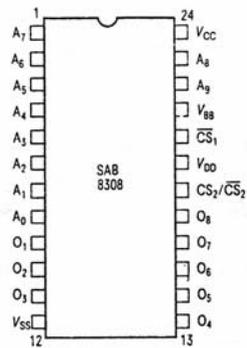
Possui também um terminal que inibe as saídas, e outro que possibilita a seleção ou não do CI, fatos que evidenciam suas importâncias durante as aplicações práticas.

### 15.4 - ROM's e PROM's

As ROM's (memórias somente de leitura) e as PROM's (ROM's programáveis) são, respectivamente, programadas na fábrica ou pelo usuário, porém não podem ser mais apagadas. São memórias fixas.

A utilização de memórias com essas características é feita geralmente no armazenamento de programas para computador, que não devem ser apagados ao se desligar a alimentação.

Um exemplo típico é a SAB 8308 cujas características são aqui expostas:



A <sub>0</sub> -A <sub>7</sub>	Address inputs
O <sub>0</sub> -O <sub>7</sub>	Data outputs
CS <sub>1</sub> , CS <sub>2</sub>	Chip select inputs

#### Memória MOS Estática

Organização: 1024 x 8 bits

Tempo de acesso: 450 ns

Compatível com TTL

Alimentação: +12V, ±5V

A memória descrita é uma ROM cuja programação é feita na fábrica pelo emprego de máscaras especiais durante a difusão das impurezas na pastilha de Si.

As memórias PROM, são programadas pelo usuário que dispõe de um programador de PROM. São memórias constituídas por microdiodos que são "queimados" pela aplicação de correntes maiores que as nominais quando se quer armazenar dados. Os diodos queimados apresentam uma certa informação, e os intactos, outra. A PROM, após programada, não pode ser apagada, pois os diodos queimados não são recuperáveis.

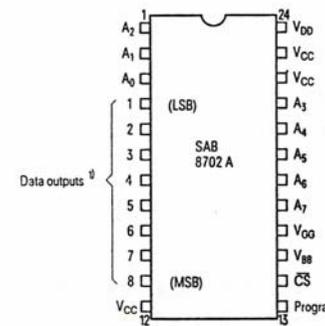
#### 15.5 - EPROM's

Existem memórias fixas, isto é, que não dependem da alimentação para a conservação dos dados mas podem ser reprogramadas. Entre elas destacam-se as EPROM's e as UV EPROM's.

As primeiras podem ser apagadas mediante a aplicação de impulsos de corrente nos transistores (MOS) componentes das células de armazenagem. Esses transistores possuem um "gate" flutuante no seu interior, que pode ser carregado eletricamente, conservando a carga por um tempo muito longo e conseqüentemente mantendo o transis

tor em um estado determinado. A aplicação do referido impulso descarrega esse gate, readquirindo, a célula de memória, seu estado original.

As UV EPROM's têm características semelhantes, porém o processo de apagamento é diferente: por uma janela de quartzo situada na parte superior do CI, por onde pode-se ver a pastilha de semicondutor, faz-se passar raios de luz ultravioleta por alguns segundos até que a carga armazenada no "gate flutuante" se desfaça. O CI SAB 8702 é um exemplo desse tipo de memória:



A <sub>0</sub> -A <sub>7</sub>	Address inputs
DO <sub>0</sub> -DO <sub>7</sub>	Data outputs <sup>1)</sup>
CS	Chip select input

For all other pins see next page.

As características enumeradas a seguir dizem respeito a esse dispositivo:

#### Memória MOS Estática

Tempo de acesso: 1,3 us

Tempo de programação: 2 minutos para todos os 2.048 bits.

Organização: 256 x 8

Compatível com TTL

Vejamos o que diz o manual do fabricante em relação ao apagamento da SAB 8702.

"Os conteúdos armazenados na SAB 8702 podem ser apagados pela exposição à radiação ultra-violeta de alta intensidade e comprimento de onda de 2.537 Å<sup>0</sup>. A dose recomendada integrada (isto é, intensidade de UV x tempo de exposição) é 6 W.seg/cm<sup>2</sup>. Exemplos de fontes de ultravioleta que podem apagar a SAB 8702 em 10 a 20 minutos são as lâmpadas ultravioleta (Osram, tipo HNS).

"As lâmpadas podem ser usadas sem filtros, e a SAB 8702

deve ser colocada a cerca de 2 ou 3 cm da lâmpada".

### 15.6 - UM CIRCUITO SIMPLES DE ESCRITA/LEITURA

Tomemos como exemplo o CI 2102, memória bastante popular. Possui 1.024 células ( $2^{10}$ ) cada qual com capacidade para 1 bit.

A "pinologia" do CI é a seguinte:

- 1) 10 entradas de endereço ( $A_0$  a  $A_9$ ), responsáveis pela seleção dos 1024 bits.
- 2) 2 pinos de alimentação: +5V e massa (GND).
- 3) Um terminal de entrada de dados ( $D_{in}$ ) e outro de saída ( $D_{out}$ ).
- 4) Um terminal de escrita-leitura (R/W) e
- 5) Um terminal de inibição  $\overline{CE}$ .

A escrita de um dado qualquer é realizada inserindo-se nas entradas de endereço a combinação lógica correspondente à célula escolhida para o armazenamento, através das chaves à esquerda do CI.

A entrada  $\overline{CE}$  é aterrada, ativando o CI, preparando-o para a operação.

O dado é colocado em  $D_{in}$  e o terminal R/W (read-write) é aterrado, o que possibilita a escrita. Nesse momento, o dado é armazenado na célula escolhida; no caso, o bit 1 é escrito na posição de endereço 1010111100.

A leitura, por exemplo, do mesmo dado considerado na escrita, é realizada através do terminal  $D_{out}$ , e visualizado pelo LED, quando se conserva o endereço e o estado de  $\overline{CE}$ , pela simples comutação da chave R/W, de 0 para 1. Nesse instante, o LED se acende.

