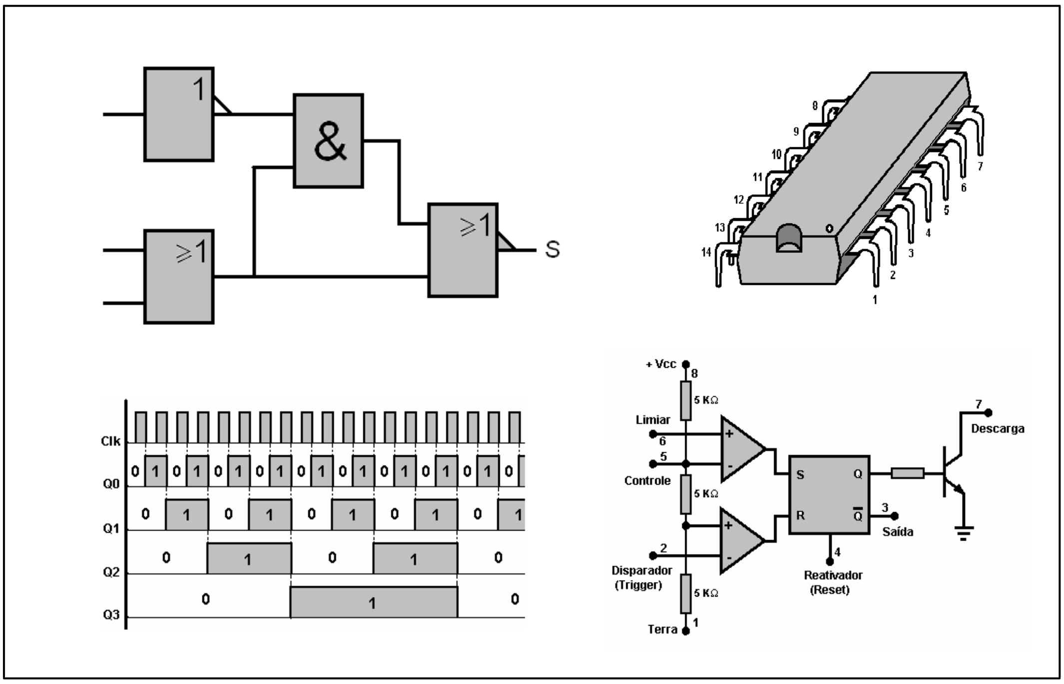


Eletrônica Digital

Prof. Valdir Dugo Zaragoza

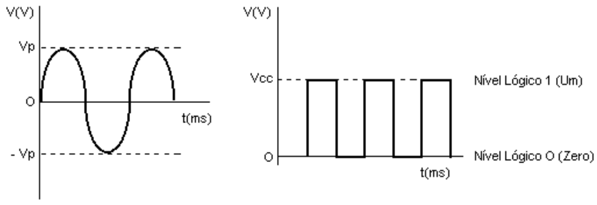
Circuitos Analógicos e Digitais

Os Sinais Elétricos trabalhados nos circuitos eletrônicos podem ser divididos em dois grandes grupos :

- Sinais Analógicos : São aqueles em que a tensão pode assumir qualquer valor contido entre dois limites, como no caso dos sinais senoidais, triangulares, sinais de áudio, vídeo, etc.

- Sinais Digitais : São aqueles em que a tensão pode assumir apenas dois valores, sendo o valor mínimo igual a zero e o valor máximo igual à tensão de alimentação do circuito.

A figura abaixo mostra exemplos gráficos dos dois sinais :



Sinal Analógico Sinal Digital

Os circuitos analógicos são aqueles que trabalham com sinais analógicos, como os amplificadores, aparelhos de rádio e televisão, etc. Os circuitos digitais trabalham com sinais que possuem apenas dois níveis de tensão, como os computadores, calculadoras, circuitos digitais, relógios, etc.

Para facilitar o estudo, quando um ponto de um circuito digital esta com a tensão igual a zero, dizemos que ele está em Nível Lógico 0 (Zero) . Caso sua tensão seja igual à tensão de alimentação (Vcc), dizemos que este ponto esta em Nível Lógico 1 (Um). Então :

V = 0 => Nível Lógico 0 (Zero)

V = Vcc => Nível Lógico 1 (Um)

Sistemas de Numeração

A história da humanidade comprova que quase todos os povos e tradições, muitos deles milenares, criaram sistemas e símbolos para representar quantidades e efetuar cálculos. Coincidentemente, todos os sistemas criados nos mais diversos continentes possuem 10 (dez) símbolos diferentes para representação numérica. Esta coincidência ocorreu porque o ser humano possui dez dedos em suas mãos. O mesmo número de símbolos e de dedos facilita a percepção e nos dá melhor noção da quantidade representada.

Além desse sistema (chamado Sistema Decimal), foram criados sistemas com quantidades diferentes de símbolos para representar quantidades. São eles : Sistema Hexadecimal, Sistema Octal e Sistema Binário.

Sistema Decimal de Numeração :

Possui dez símbolos para representação numérica. No nosso alfabeto são 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9
A aritmética usada para cálculo nesse sistema é conhecida por todos nos.

Sistema Hexadecimal de Numeração :

Possui dezesseis símbolos para representação numérica. São eles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F. É muito utilizado em computação para representar endereços de memória.

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

Sistema Octal de Numeração :

Possui oito símbolos para representação numérica. São eles : 0, 1, 2, 3, 4, 5, 6 e 7.

Sistema Binário de Numeração :

Possui apenas dois símbolos para representação numérica. São eles : 0 e 1. É o sistema mais utilizado em eletrônica digital, pois o número de símbolos é igual ao número de estados possíveis para um ponto do circuito. Os cálculos no sistema Binário de Numeração devem ser feitos com uma técnica diferente, chamada de Álgebra Booleana (criada por Boole).

Conversão entre Sistemas de Numeração

É comum ser necessária a conversão de uma quantidade representada em um sistema de numeração para outro. É importante observar que a mudança de um sistema para outro altera apenas a forma de representação da mesma quantidade. As técnicas para efetuar a conversão entre sistemas serão estudadas agora :

Conversão do Sistema Decimal Para Binário :

Para converter um número representado no Sistema Decimal para o sistema Binário, basta seguir os passos descritos abaixo :

- Dividir o número Decimal no conjunto dos números inteiros sucessivamente por dois, até que o resto seja zero ou um.

- Dispor a partir do último quociente os restos das operações de divisão (de baixo para cima).

Exemplo : Converter 9_d (nove decimal) em binário.

```
9 |2
1 4 |2
  0 2 |2
    0 1
```

Então $9_d = 1001_b$ (nove decimal = 1001 binário)

Converter 11_d em binário :

```
11 |2
 1 5 |2
   1 2 |2
     0 1
```

Então $11_d = 1011_b$ (onze decimal = 1011 binário)

Conversão do Sistema Decimal Para Octal :

O processo é similar ao descrito anteriormente, bastando alterar a operação para divisão por oito.

Exemplo : Converter 133_d para Octal :

133 | 8
5 16 | 8
0 **2**

Então 133_d = 205_o (133 decimal = 205 Octal)

208 | 8
0 26 | 8
2 **3**

Então 208_d = 320_o (208 decimal = 320 Octal)

Conversão do Sistema Decimal Para Hexadecimal :

Assim como a conversão de decimal para binário ou Octal, o processo é baseado na divisão, bastando alterar a operação para divisão por dezesseis.

Exemplo : Converter 133_d para hexadecimal :

133 | 16
5 **8**

Então 133_d = 85_h (133 decimal = 85 hexadecimal)

208 | 16
0 **13** => D

Como 13_h = D,

Então 208_d = $D0_h$ (208 decimal = D0 hexadecimal)

Conversão para o sistema Decimal

Conforme foi visto anteriormente, o processo de conversão do sistema Decimal para outros sistemas é baseado na operação de divisão. Estudaremos agora o processo inverso, ou seja, converter de qualquer sistema para Decimal.

Esta conversão é baseada na somatória das potências da base de cada sistema, ou seja, é uma operação baseada na potenciação.

Conversão de Binário para Decimal :

A base do sistema binário é igual a 2, portanto faremos uma tabela com as potências de 2 crescente da direita para a esquerda :

2^4	2^3	2^2	2^1	2^0

Basta escrever o número cuja conversão é desejada na tabela. Por exemplo : Queremos saber quanto é 10011b no sistema decimal :

Primeiro escrevemos o número na tabela :

2^4	2^3	2^2	2^1	2^0
1	0	0	1	1

Basta agora somar as potências de base 2 das colunas onde o dígito binário for igual a 1 :

Então $10011_b = 2^0 + 2^1 + 2^4 = 1 + 2 + 16 = 19$ => Então **10011_b = 19_d**

Outro Exemplo : Desejamos converter 11011_b em decimal :

2^4	2^3	2^2	2^1	2^0
1	1	0	1	1

$11011_b = 2^0 + 2^1 + 2^3 + 2^4 = 27$ => Então **11011_b = 27_d**

Conversão de Hexadecimal para Decimal :

O sistema Hexadecimal de Numeração possui 16 símbolos para representação numérica, sendo portanto a sua Base igual a 16. A conversão do sistema hexadecimal para decimal é similar à descrita anteriormente, bastando alterar a base de potência para 16.

Montamos então uma tabela com as potências de 16 :

16^4	16^3	16^2	16^1	16^0

Por exemplo, queremos saber quanto é 1FCA 7_h no sistema decimal :

Da mesma forma escreveremos o número a converter na tabela :

16^4	16^3	16^2	16^1	16^0
1	F	C	A	7

Basta agora multiplicar as potências pelo digito correspondente na tabela :

1FCA 7_h = $7.16^0 + A.16^1 + C.16^2 + F.16^3 + 1.16^4$

Como no sistema hexadecimal A = 10 ; B = 11 ; C = 12 ; D = 13 ; E = 14 e F = 15 :

1FCA 7_h = $7.16^0 + 10.16^1 + 12.16^2 + 15.16^3 + 1.16^4$ = $7.1 + 10.16 + 12.256 + 15.4096 + 1.65536$

Então 1FCA 7_h = 7 + 160 + 3072 + 61440 + 65536 => **1FCA 7_h = 130215_d**

Outro exemplo : Queremos saber quanto é $1B03_h$ no sistema decimal :

16^4	16^3	16^2	16^1	16^0
0	1	B	0	3

$1B03_h = 3.16^0 + 0.16^1 + B.16^2 + 1.16^3 = 3.1 + 0.16 + 11.256 + 1.4096$

Então **$1B03_h$ = 6915_d**

Conversão de Octal para Decimal :

O sistema Octal de Numeração possui 8 símbolos para representação numérica, sendo portanto a sua Base igual a 8. Basta utilizar o método da somatória das potências de 8.

Montamos então uma tabela :

8^4	8^3	8^2	8^1	8^0

Exemplo : Converter 1325_o em decimal

8^4	8^3	8^2	8^1	8^0
0	1	3	2	5

Então $1325_o = 5.8^0 + 2.8^1 + 3.8^2 + 1.8^3 + 0.8^4 = 5.1 + 2.8 + 3.64 + 1.512 + 0.4096 = 725$

Então 1325_o **=** $\mathbf{725_d}$

Exercícios Propostos :

1) Converter os seguintes números decimais : 256, 128, 182, 36 e 99

a) Para o sistema Binário
b) Para o sistema Hexadecimal
c) Para o sistema Octal

2) Converter para o sistema Decimal os seguintes números :

a) 13638_o
b) 689_o
c) 2834_o

d) $FFFF_h$
e) 0256_h
f) $4A3D_h$

g) 11111_b
h) 01101_b
i) 10101_b

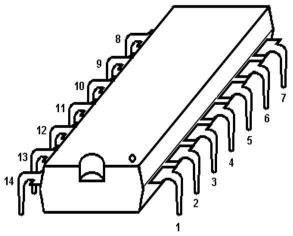
Funções e Portas Lógicas

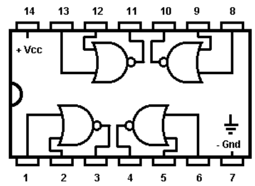
Como vimos na lição anterior, os circuitos digitais trabalham com sinais digitais, ou seja, sinais que podem assumir apenas dois valores diferentes de tensão. Caso a tensão seja igual a zero, o circuito está em nível lógico zero, enquanto que se a tensão for igual a Vcc, dizemos que o circuito está em nível lógico um.

Por esse motivo, os componentes utilizados em circuitos digitais são diferentes dos utilizados em circuitos analógicos.

Os principais componentes utilizados em eletrônica digital são chamados de Portas Lógicas. Uma porta lógica é um circuito construído com transistores, em uma única pastilha de material semicondutor (silício) e encapsuladas em circuitos integrados (CI). O número de portas lógicas em um circuito integrado depende do número de terminais ocupados por ela. Quanto mais entradas uma porta tiver, menor será o número de portas contidas em um circuito integrado (CI).

A figura abaixo mostra um exemplo de circuito integrado digital :





CI 4001 - Família CMOS - 4 Portas NOU de 2 entradas

Encapsulamento DIL ou DIP

(Dual In Line)

Função e Porta E ou AND

O circuito analógico mostrado abaixo funciona segundo a função E ou AND :

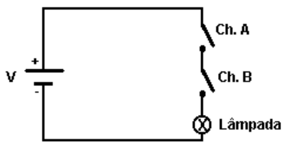
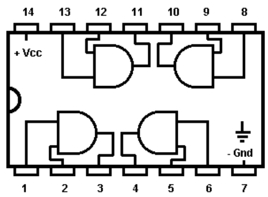
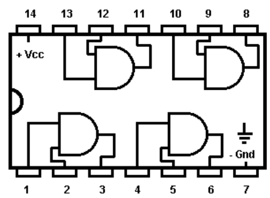


Tabela Verdade

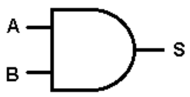
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1



CI 4081 - Família CMOS - 4 Portas E de 2 entradas

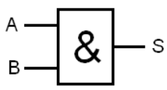


CI 7408 - Família TTL - 4 Portas E de 2 entradas



Porta E ou AND

$\boxed{S = A.B}$



Simbologia ASA

Simbologia ABNT

Quando qualquer uma das chaves (Ch.A e Ch.B) estiverem abertas (nível lógico zero), não haverá circulação de corrente elétrica pela lâmpada, permanecendo a mesma apagada. A lâmpada somente estará acesa (nível lógico 1 (um)) quando ambas as chaves (Ch. A **E** Ch. B) estiverem fechadas (nível lógico 1 (um)).

A porta lógica “ E ” , cujo símbolo é mostrado na figura, funciona de forma análoga ao circuito acima, ou seja, somente haverá nível lógico 1 na saída “ S “ quando as duas entradas A e B estiverem em nível lógico 1.

Esta função pode ser representada matematicamente através da sua Expressão Lógica :

$\boxed{S = A \cdot B}$

Nas três primeiras linhas da tabela verdade temos o produto A . B = 0

Apenas na Quarta e última linha temos A = 1 e B = 1 .

Então o produto A B = 1 apenas na última situação, pois ambas as entradas estão em nível lógico 1.

Função e Porta OU ou OR

A figura abaixo mostra um circuito analógico que representa a função OU ou OR :

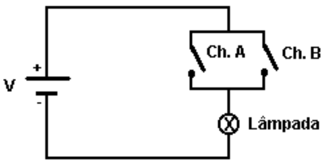
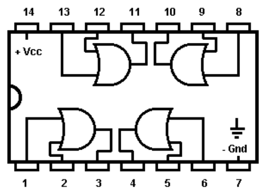
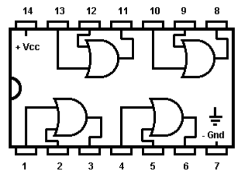


Tabela Verdade

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



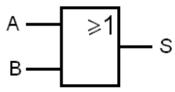
CI 4071 - Família CMOS - 4 Portas OU de 2 entradas



CI 7432 - Família TTL - 4 Portas OU de 2 entradas

$\boxed{S = A+B}$

Porta OU ou OR



Simbologia ASA **Simbologia ABNT**

Para que a lâmpada acenda (nível lógico um), basta que qualquer uma das chaves seja fechada, ou seja , basta fechar a chave A **OU** a chave B.

A lâmpada permanecerá apagada (nível lógico zero) apenas na situação mostrada na primeira linha da tabela verdade, pois neste caso A = 0 e B = 0. Como a expressão lógica desta função é S = A + B, com as duas chaves abertas teremos S = 0 + 0 => S = 0.

Nas demais situações, pelo menos uma das chaves estará fechada (nível lógico 1), garantindo a circulação de corrente pela lâmpada.

Função e Porta NÃO ou NOT
Inversor .

A figura abaixo mostra o circuito analógico correspondente à função NÃO ou NOT. Quando a chave A estiver em nível lógico 0 (aberta) a corrente elétrica poderá passar através da lâmpada, mantendo-a acesa. Se fecharmos a chave (nível lógico 1), a corrente será desviada pela mesma, apagando a lâmpada (nível lógico 0).

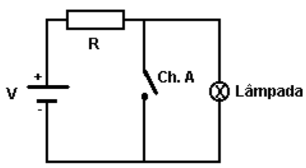
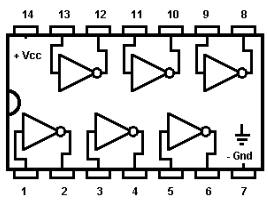
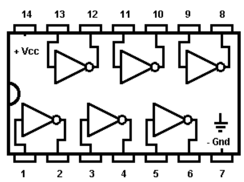


Tabela Verdade

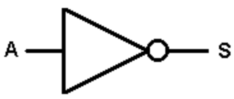
A	S
0	1
1	0



CI 4069 - Família CMOS - 6 Inversores

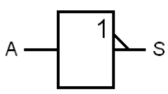


CI 7404 - Família TTL - 6 Inversores



$\boxed{S = \overline{A}}$

Porta NÃO ou NOT - Inversor



Simbologia ASA **Simbologia ABNT**

Sendo assim, no caso de uma porta NÃO, a saída estará sempre em um nível lógico inverso ao da entrada, ou seja, A saída **S** é igual ao **complemento** da entrada **A** , ou **S e A** são variáveis **complementares**. Este é o motivo pelo qual este componente é mais conhecido como **INVERSOR**.

A expressão lógica deste componente é : $S = \overline{A}$ (dizemos S é igual a A "barrado")

A figura mostra também exemplos de circuitos integrados (CI's) comerciais desses componentes. São eles o CI 7404 da família TTL e o CI 4069 da família CMOS.

Portas formadas através de combinação

Utilizando portas lógicas juntamente com inversores, é possível obter portas lógicas cujo funcionamento seja a combinação entre as tabelas verdade dos componentes utilizados.

Estudaremos agora duas destas combinações :

Função e Porta NE ou NAND

A porta lógica NE ou NAND é a combinação entre a porta E, mais um inversor.O circuito analógico mostrado abaixo funciona segundo a função NE ou NAND :

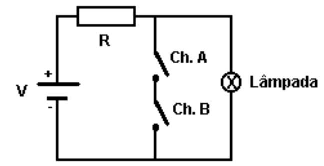
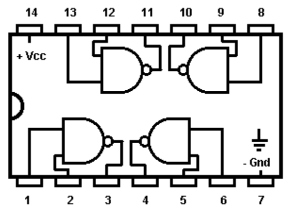
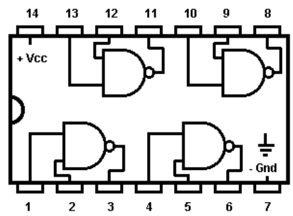


Tabela Verdade

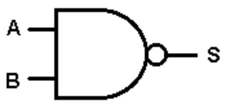
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



CI 4011 - Família CMOS - 4 Portas NE de 2 entradas

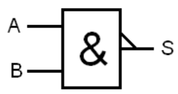


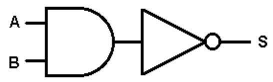
CI 7400 - Família TTL - 4 Portas NE de 2 entradas



Porta NE ou NAND

$S = \overline{A.B}$





Simbologia ASA **Simbologia ABNT** **equivalência**

Enquanto uma das duas chaves A ou B estiver aberta (nível lógico 0), a corrente elétrica passará através da lâmpada, mantendo-a acesa (nível lógico 1). Apenas com ambas as chaves fechadas (nível 1) é que a corrente será desviada, apagando a lâmpada (nível 0)

Comparando as tabelas verdade da porta E e da porta NE, podemos observar que são exatamente contrárias, ou seja, numa situação em que a porta E estiver com a saída em nível 1, a porta NE estará em nível 0 e vice-versa. Isto ocorre pela presença de um inversor na saída da porta NE. Podemos então comparar a porta NE com uma porta E mais um inversor na saída

A expressão lógica da porta NE é : $S = \overline{(A\ .\ B)}$, ou seja, a saída S é igual ao complemento do produto (A **.**B). Dizemos então que S é igual ao produto A vezes B " Barrado" .

A figura mostra dois exemplos de circuitos integrados que contém portas NE. São eles o CI 4011 da família CMOS de circuitos lógicos e o CI 7400 da família TTL.

Função e Porta NOU ou NOR

Da mesma forma, também podemos combinar uma porta OU com um inversor. Esta combinação forma a porta NOU ou NOR. A figura abaixo mostra o circuito analógico que funciona segundo a função NOU , sua tabela verdade, expressão lógica e exemplos de circuitos integrados comerciais. São eles o CI 4001 da família CMOS de circuitos lógicos e o CI 7402 da família TTL.

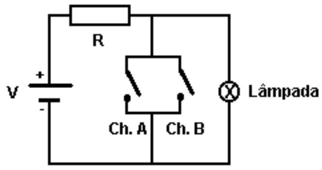
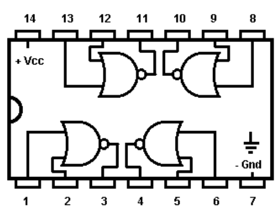
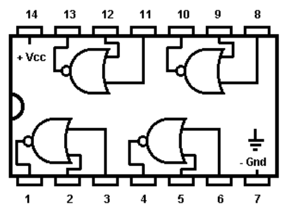


Tabela Verdade

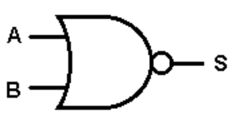
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



CI 4001 - Família CMOS - 4 Portas NOU de 2 entradas

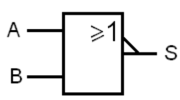


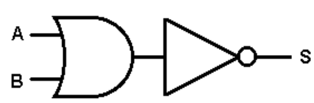
CI 7402 - Família TTL - 4 Portas NOU de 2 entradas



Porta NOU ou NOR

$S = \overline{A+B}$

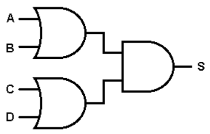




Simbologia ASA **Simbologia ABNT** **equivalência**

Exercícios Resolvidos :

1) Determine a expressão lógica dos circuitos abaixo :

a) 

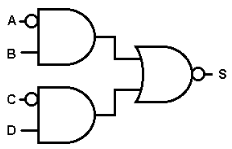
Neste circuito são introduzidas quatro variáveis Binárias, A e B na primeira porta OU e C e D na segunda porta OU. Como a porta OU executa a operação lógica de Adição,

teremos na saída da primeira porta **S = (A + B)** .

Na saída da segunda porta de entrada teremos **S = (C + D)** .

Os níveis lógicos das saídas das duas portas OU de entrada são introduzidos nas entradas da porta E de saída. Como a porta E executa a operação lógica da multiplicação, teremos :

$$\mathbf{S = (A + B) . (C + D)}$$

b) 

Neste circuito também são introduzidas quatro variáveis Binárias, A e B na primeira porta E e C e D na segunda porta E. Como a porta E executa a operação lógica de multiplicação,

teremos na saída da primeira porta $\mathbf{S = (\overline{A} . B)}$.

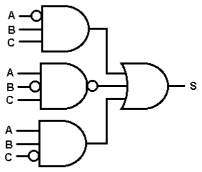
Na saída da segunda porta de entrada teremos $\mathbf{S = (\overline{C} . D)}$.

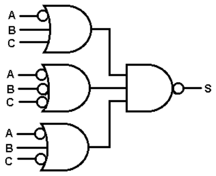
Os níveis lógicos das saídas das duas portas E de entrada são introduzidos nas entradas da porta OU de saída. Como a porta OU executa a operação lógica de Adição, teremos :

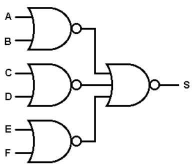
$$\mathbf{S = \overline{(\overline{A} . B) + (\overline{C} . D)}}$$

Exercícios Propostos :

1) Determine a expressão lógica dos circuitos abaixo :

a) 

b) 

c) 

Obtendo Circuitos Lógicos através da Tabela Verdade

Existem situações em que desejamos construir um circuito lógico que funcione segundo uma tabela verdade conhecida. Nesses casos basta utilizar a técnica descrita abaixo :

- Marcar todas as linhas da tabela nas quais a saída for igual a 1 (um).
- Escrever as variáveis de entrada de cada linha na forma de produto.
- Marcar como variável complementar (barrada) aquelas que forem iguais a 0 (zero).
- Efetuar a somatória de todos os produtos.

Exemplo : Desejamos construir um circuito digital que funcione conforme a tabela verdade abaixo :

A	B	C	S
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

Primeiramente marcamos as linhas nas quais a saída seja igual a 1.

O segundo passo é escrever as variáveis de cada linha na forma de produto, marcando as iguais a zero como complementares (barradas).

Temos na primeira linha A = 0 ; B = 0 e C = 0

Então o produto da primeira linha será $(\overline{A}\ .\ \overline{B}\ .\ \overline{C})$

Na segunda linha marcada temos A = 0 ; B = 1 e C = 0

Então o produto da segunda linha será $(\overline{A}\ .\ B\ .\ \overline{C})$

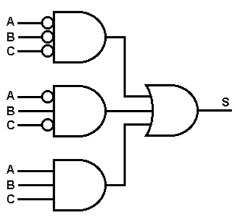
Na terceira linha marcada temos A = 1 ; B = 1 e C = 1

Então o produto da terceira linha será $(A\ .\ B\ .\ C)$

Basta agora efetuar a somatória dos produtos obtidos. A expressão lógica do circuito desejado é :

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

O circuito que executa esta expressão lógica é mostrado na figura :



Outro Exemplo : Determinar o circuito digital que execute a tabela verdade mostrada abaixo :

A	B	C	S
0	0	0	**1**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Primeiramente marcamos as linhas em que a saída é igual a 1

O segundo passo é escrever as variáveis de cada linha na forma de produto, marcando as iguais a zero como complementares (barradas).

Temos na primeira linha A = 0 ; B = 0 e C = 0

Então o produto da primeira linha será $(\overline{A} \cdot \overline{B} \cdot \overline{C})$

Na segunda linha marcada temos A = 1 ; B = 0 e C = 1

Então o produto da segunda linha será $(A \cdot \overline{B} \cdot C)$

Na terceira linha marcada temos A = 1 ; B = 1 e C = 0

Então o produto da terceira linha será $(A \cdot B \cdot \overline{C})$

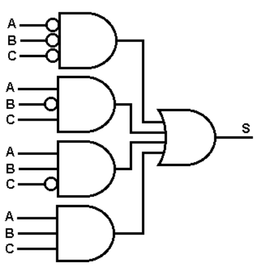
Na terceira linha marcada temos A = 1 ; B = 1 e C = 1

Então o produto da terceira linha será $(A \cdot B \cdot C)$

Basta agora efetuar a somatória dos produtos obtidos. A expressão lógica do circuito desejado é :

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

O circuito que executa esta expressão lógica é mostrado na figura :



Exercícios Propostos :

1) Determinar o circuito digital que execute a tabela verdade mostrada abaixo :

A	**B**	**C**	**S**
0	0	0	**1**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

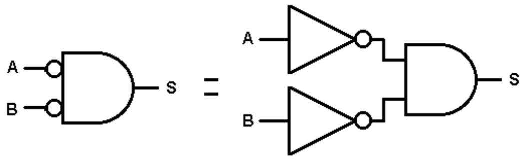
2) Determinar o circuito digital que execute a tabela verdade mostrada abaixo :

A	**B**	**C**	**D**	**S**
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**0**
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**1**
0	1	1	1	**0**
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**0**
1	1	0	0	**0**
1	1	0	1	**0**
1	1	1	0	**0**
1	1	1	1	**1**

Equivalência entre Blocos Lógicos

Conforme foi visto na lição anterior, durante a obtenção de circuitos lógicos a partir da sua tabela verdade, é comum que seja necessário utilizar portas lógicas com inversores nas entradas, porém, não são fabricados circuitos integrados com essas características. Nesses casos é possível substituir um bloco lógico pelo seu bloco equivalente, a fim de reduzir o número de componentes do circuito.

Observe a figura abaixo. Uma porta E com inversores nas suas entradas não existe comercialmente. Seria então necessário utilizar três componentes, ou seja, uma Porta E mais dois



inversores :

Para evitar o uso de três componentes, podemos substituir a porta E com dois inversores nas entradas pelo seu bloco equivalente (porta NOU) conforme mostra a figura :

Para comprovar a equivalência entre esses dois blocos, basta confrontar as suas respectivas tabelas verdade.

Porta E c/ Inversores

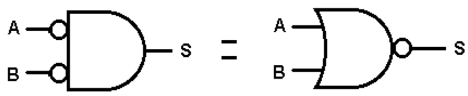
A	B	$\overline{A}$	$\overline{B}$	S
0	0	1	1	**1**
0	1	1	0	**0**
1	0	0	1	**0**
1	1	0	0	**0**

$S = \overline{A} \, . \, \overline{B}$

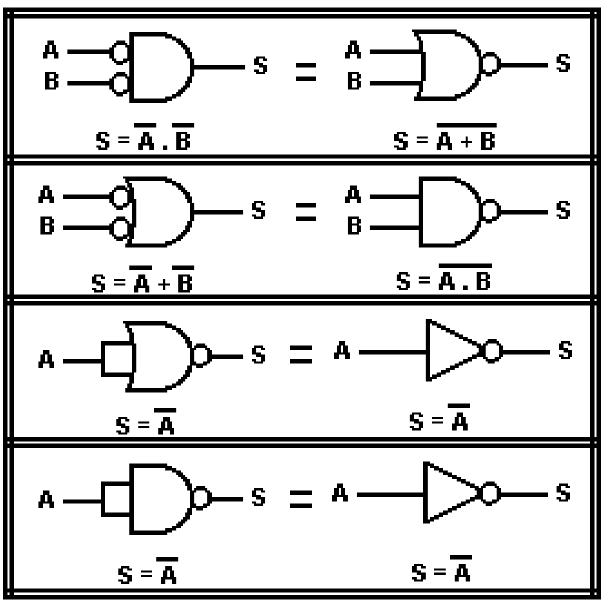
Porta NOU

A	B	$\overline{S}$
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

$S = \overline{A + B}$



A tabela abaixo mostra as equivalências entre blocos lógicos mais utilizadas :



Circuitos Combinacionais

Circuito OU Exclusivo :

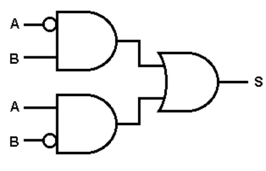
O circuito combinacional OU Exclusivo possui a seguinte característica : quando suas entradas estiverem em um mesmo nível lógico, a saída estará baixa (nível lógico zero) e quando as entradas estiverem em níveis lógicos diferentes, a saída estará alta (nível lógico um).

A tabela verdade do circuito OU Exclusivo é mostrada abaixo :

OU Exclusivo		
A	**B**	**S**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Utilizando a técnica de obtenção do circuito através da tabela verdade, chegamos ao seguinte circuito lógico :

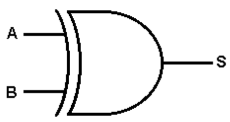
$S = (\overline{A} . B) + (A . \overline{B})$



Circuito OU Exclusivo

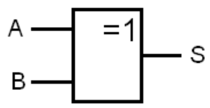
Porta OU Exclusivo ou EXOR

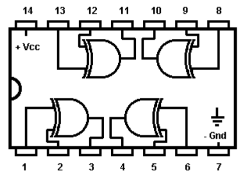
São fabricados circuitos integrados contendo portas EXOR. A figura abaixo mostra dois exemplos de CI's disponíveis no comércio de componentes eletrônicos :



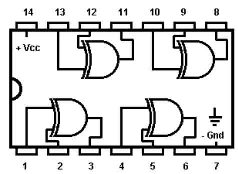
Porta EXOR

S = A ⊕ B





CI 4030 - Família CMOS - 4 Portas EXOR de 2 entradas



CI 7486 - Família TTL - 4 Portas EXOR de 2 entradas

Simbologia ASA **Simbologia ABNT**

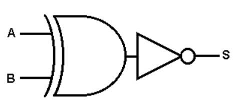
A expressão lógica executada pela porta EXOR é $S = A \oplus B$ (lê-se A "OU Exclusivo" B)

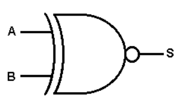
Porta EXNOR ou circuito Coincidência

O circuito Coincidência ou EXNOR funciona exatamente da forma contrária ao circuito OU Exclusivo, ou seja, a saída estará em nível lógico um apenas quando os níveis lógicos das entradas for igual (coincidirem). Podemos então comparar o circuito coincidência com um circuito OU Exclusivo mais um inversor.

Coincidência		
A	**B**	**S**
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

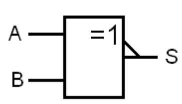
A tabela verdade e o símbolo do componente são mostrados na figura :



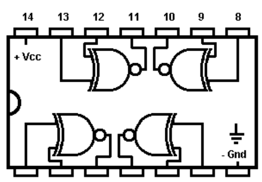


Porta EXNOR

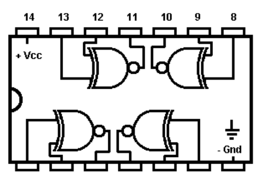
S = A ⊙ B



equivalência **Simbologia ASA** **Simbologia ABNT**



CI 4077 - Família CMOS - 4 portas EXNOR de 2 entradas

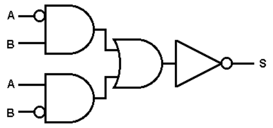


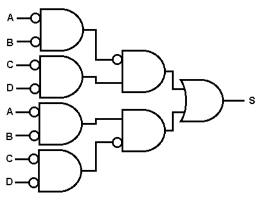
CI74266 - Família CMOS - 4 portas EXNOR de 2 entradas

A expressão lógica do circuito coincidência é $S = A \odot B$ (lê-se A "coincidência" B)

Exercícios Propostos :

Utilizando a equivalência entre blocos lógicos, simplifique os circuitos abaixo :

1) 

2) 

Simplificação de Circuitos com Álgebra Booleana

Nas lições anteriores foram descritas as técnicas de obtenção de circuitos lógicos através de sua tabela verdade, porém, os circuitos obtidos com esta técnica podem ser simplificados, de forma que o circuito final utilize um número menor de componentes. As técnicas de simplificação ou minimização, aplicadas a circuitos lógicos, permitem que os mesmos fiquem menores, mais baratos, mais fáceis de construir e ocupem menos espaço.

A Álgebra Booleana é um conjunto de Postulados, Propriedades, Teoremas e Identidades para simplificação matemática de expressões Booleanas. As variáveis utilizadas nessas expressões são variáveis binárias, ou seja, podem assumir apenas dois valores numéricos distintos (zero ou um).

As regras que compõem a Álgebra Booleana são apresentadas no quadro abaixo :

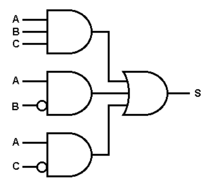
Postulados		
Complementação	**Adição**	**Multiplicação**
$A = 0 \quad \overline{A} = 1$ $A = 1 \quad \overline{A} = 0$	$0 + 0 = 0$ $0 + 1 = 1$ $1 + 0 = 1$ $1 + 1 = 1$	$0 . 0 = 0$ $0 . 1 = 0$ $1 . 0 = 0$ $1 . 1 = 1$
Identidades		
Complementação	**Adição**	**Multiplicação**
$\overline{\overline{A}} = A$	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \overline{A} = 1$	$A . 0 = 0$ $A . 1 = A$ $A . A = A$ $A . \overline{A} = 0$

Propriedades	
Comutativa	$A + B = B + A$ $A . B = B . A$
Associativa	$A + (B + C) = (A + B) + C = A + B + C$ $A . (B . C) = (A . B) . C = A . B . C$
Distributiva	$A . (B + C) = A.B + A.C$
Teoremas de Morgan	$\overline{(A + B)} = \overline{A} . \overline{B}$ $\overline{(A . B)} = \overline{A} + \overline{B}$
Outras Identidades	$A + A.B = A$ $A + \overline{A}.B = A + B$ $(A + B).(A + C) = A + B.C$

Exemplo de Aplicação :

Desejamos construir um circuito lógico através da sua Tabela Verdade mostrada abaixo :

A	B	C	S
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**



Circuito antes da simplificação

Sua expressão lógica $S = A \cdot B \cdot C + A \cdot \overline{C} + A \cdot \overline{B}$

Primeiramente colocamos a variável A em evidência :

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

Aplicando o Teorema de Morgan :

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

Observe que entre parênteses existe a soma de uma variável mais o seu complemento :

Fazendo $B.C = X$, teremos : $S = A \cdot (X + \overline{X})$

Aplicando a identidade da Adição, temos que : $(X + \overline{X}) = 1$. Então

$\underline{S = A}$

Portanto, o circuito todo pode ser substituído por um fio conduzindo a variável A.

Circuito após a simplificação

Exercícios Propostos :

1) Simplificar a Expressão Booleana gerada pela Tabela Verdade mostrada abaixo :

a)

A	B	C	S
0	0	0	**1**
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	**0**

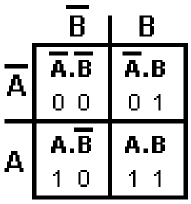
b) Simplifique a Expressão Lógica : $S = (A + B + C) . (\overline{A} + \overline{B} + C)$

Simplificação de Expressões Booleanas pelo Diagrama de Veitch-Karnaugh

A simplificação de expressões lógicas pela Álgebra Booleana é eficiente, porém, é uma técnica mais difícil e demorada. Uma outra forma de realizar essa simplificação Diagrama de Veitch-Karnaugh.

É um processo gráfico relacionado com a tabela verdade do circuito, de forma que os agrupamentos de variáveis permitem a simplificação.

A figura abaixo mostra o Diagrama de Veitch-Karnaugh para duas variáveis :



Em uma Tabela Verdade envolvendo duas variáveis existem quatro combinações possíveis para A e B : São elas - 00 , 01 , 10 e 11

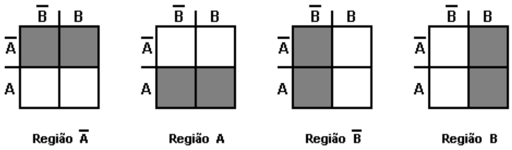
Para saber o número de combinações possíveis basta utilizar a fórmula :

$\mathbf{N = 2^n}$, onde N = Número de Combinações
n = Número de Variáveis

então $N = 2^2 = 4$ combinações possíveis

Cada quadrado do diagrama está relacionado com uma das combinações da Tabela Verdade do circuito. Para sua utilização basta preencher o diagrama com os valores das saídas nos seus respectivos lugares.

A figura abaixo mostra como são divididas as regiões no diagrama :

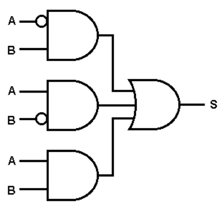


Exemplo de Aplicação :

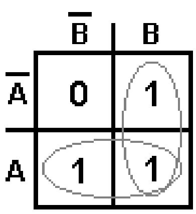
Desejamos obter a simplificação do circuito cuja Tabela Verdade é mostrada abaixo :

A	B	S
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Pela técnica estudada anteriormente chegamos ao seguinte circuito :



Para simplificar a expressão utilizando o diagrama de Veitch-Karnaugh, basta colocar os valores das saídas nos seus respectivos campos, observando a formação de pares na horizontal ou na vertical :



Observe que, após o preenchimento do diagrama, podemos agrupar dois pares de variáveis iguais a 1, um na horizontal – Região A e outro na vertical – Região B.

O resultado da simplificação será :

$S = A + B$ gerando o seguinte circuito simplificado :

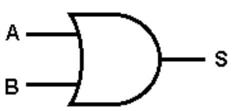
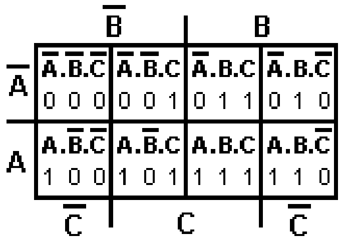


Diagrama de Veitch-Karnaugh para expressões com três variáveis

A utilização do diagrama em expressões de três variáveis é semelhante à descrita na lição anterior, porém, com 3 variáveis de entrada, serão oito combinações diferentes.

O diagrama para 3 variáveis é mostrado na figura abaixo :

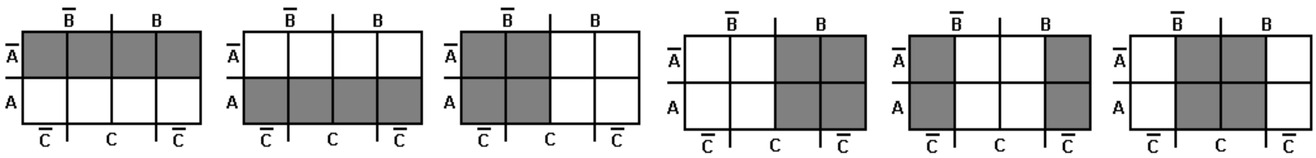


Assim como no diagrama para duas variáveis, os valores da saída na tabela verdade deverão ser colocados no seu campo respectivo .

A diferença principal deste diagrama é que poderemos agrupar as variáveis iguais a 1 em quadras e em pares. Cada quadra envolve apenas uma região, enquanto que um par estará contido em duas regiões diferentes.

Quanto maior for o agrupamento, mais simples será o circuito final obtido.

A figura abaixo mostra as possíveis regiões de agrupamento no diagrama :



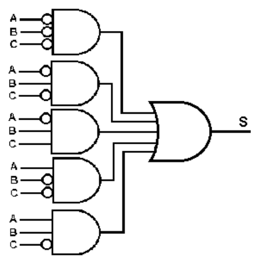
Região $\overline{A}$ Região A Região $\overline{B}$ Região B Região C Região $\overline{C}$

Exemplo de Aplicação :

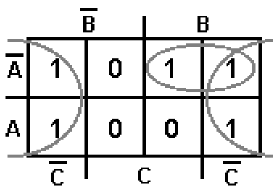
Simplificar o circuito gerado pela Tabela Verdade Abaixo :

A	B	C	S
0	0	0	**1**
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	**0**

O circuito obtido pela tabela é mostrado na figura :



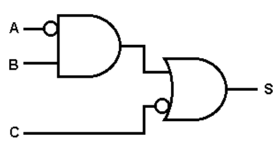
Para obter o circuito simplificado basta utilizar o diagrama de Veitch-Karnaugh :



Observe que existe uma quadra na região $\overline{C}$ e um par nas regiões $\overline{A}$. B

A saída será : <u>S = $\overline{A}$.B + $\overline{C}$</u>

O circuito simplificado é mostrado na figura abaixo :



Exercícios Propostos :

Simplifique pelos diagramas de Veitch-Karnaugh os circuitos gerados pelas Tabelas mostradas abaixo :

a)

A	B	S
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

b)

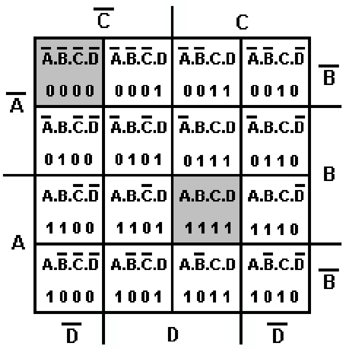
A	B	C	S
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

c)

A	B	C	S
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

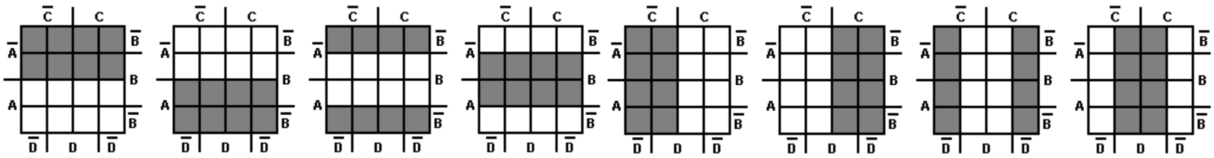
Diagrama de Veitch-Karnaugh para expressões com quatro variáveis

Apesar de mais complexo, a forma de utilização do diagrama para quatro variáveis é a mesma dos apresentados nas lições anteriores. Abaixo é mostrado o diagrama e as regiões envolvidas na simplificação :



É importante observar a forma correta de preenchimento do diagrama, para evitar a obtenção de resultados incorretos. Na prática, os valores da tabela verdade devem ser colocados sempre “de fora para dentro”, ou seja, o primeiro campo a ser preenchido é o superior à esquerda, e o último é o terceiro campo da terceira linha.

A principal diferença deste diagrama é que podemos obter grupos de oito (oitavas), quatro (quadras) e dois (pares). As oitavas envolvem apenas uma região ou variável, enquanto que as quadras envolvem duas regiões e os pares três regiões. Quanto maiores os agrupamentos obtidos, mais simples serão os circuitos finais obtidos.



Região $\bar{A}$ Região A Região $\bar{B}$ Região B Região $\bar{C}$ Região C Região $\bar{D}$ Região D

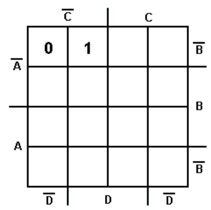
Exemplo de Aplicação :

Vamos obter o circuito que funcione segundo a Tabela Verdade mostrada abaixo :

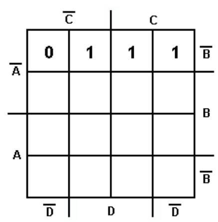
A	B	C	D	S
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**1**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**1**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**1**
1	1	0	1	**1**
1	1	1	0	**0**
1	1	1	1	**1**

Observe a seqüência correta para preenchimento do diagrama.

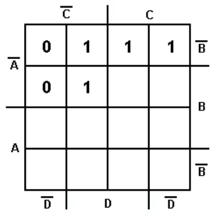
Iniciamos com o campo superior esquerdo, seguindo na mesma linha até o centro do diagrama :



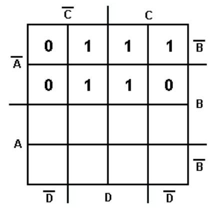
Depois da direita para a esquerda, completando a primeira linha :



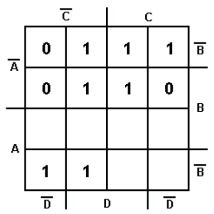
Depois na segunda linha, da esquerda para a direita, até o centro :



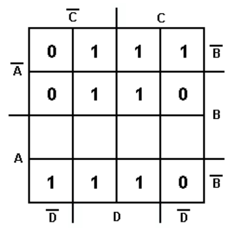
Completamos a segunda linha da direita para a esquerda, sempre de fora para dentro :



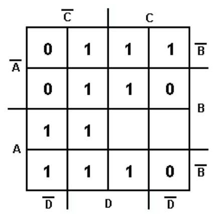
Para manter a seqüência, iniciamos agora na Quarta linha, até o centro :



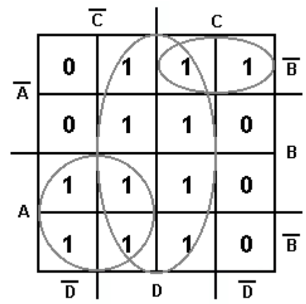
Agora da direita para a esquerda :



Os mesmos passos devem ser seguidos para a terceira e última linha :

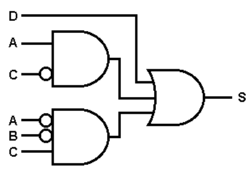


Até obtermos o diagrama completo :



Observe que podemos agrupar uma oitava na região D, uma quadra nas regiões $\bar{A}.C$ e um par nas regiões $\bar{A}.\bar{B}.C$, gerando a seguinte expressão : $\underline{S = D + \bar{A} . C + \bar{A}. \bar{B} .C}$

O circuito final obtido após a simplificação é mostrado na figura abaixo :



Exercícios Propostos :

1) Simplifique os circuitos gerados pelas Tabelas abaixo :

a)

A	B	C	D	S
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**1**
0	0	1	1	**0**
0	1	0	0	**1**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**0**
1	0	1	0	**1**
1	0	1	1	**0**
1	1	0	0	**1**
1	1	0	1	**0**
1	1	1	0	**1**
1	1	1	1	**1**

b)

A	B	C	D	S
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	**1**
1	1	0	0	**1**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**1**

Código BCD 8421

O código BCD 8421 é muito utilizado em circuitos contadores. Nada mais é do que a relação entre os números Decimais e Binários, podendo também haver a relação com os números Hexadecimais.

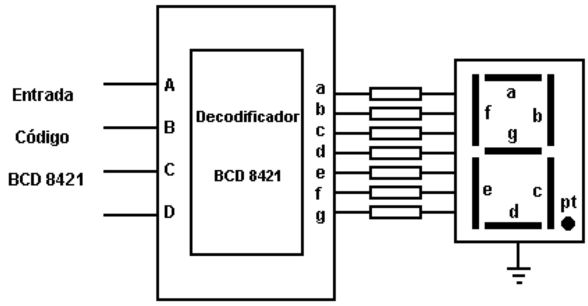
Na tabela abaixo é mostrada esta relação :

Código BCD 8421	**Decimal**	**Hexadecimal**
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Decodificadores

Decodificadores são circuitos lógicos construídos com a finalidade específica de converter um determinado código ou sistema de numeração em outro.

Os decodificadores mais utilizados em eletrônica digital são os decodificadores para "Display" ou indicador de sete segmentos. Através desse componente podemos converter para a forma visual uma informação do código BCD 8421 em decimal. A figura abaixo mostra um exemplo de utilização :



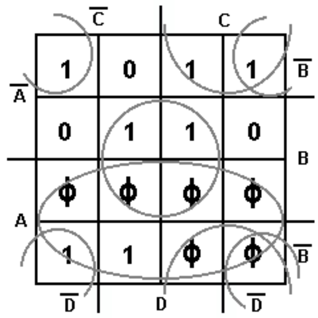
O funcionamento do circuito é bastante simples. Por exemplo, caso os níveis lógicos ABCD na entrada do decodificador sejam 0101 respectivamente (equivale a 5 decimal), o circuito fará com que os LED's a, c, d, f, g acendam, desenhando no indicador o número 5. Os resistores ligados entre o decodificador e o display servem para limitar a corrente nos LED's impedindo que sejam danificados.

Como o decodificador possui sete saídas, internamente existem também sete circuitos lógicos independentes, sendo um para cada saída.

Podemos obter esses circuitos através do diagrama de Veitch-Karnaugh como mostra o exemplo abaixo :

Para determinar o circuito lógico que comanda o LED a . A partir da tabela verdade podemos utilizar o diagrama de Veitch-Karnaugh :

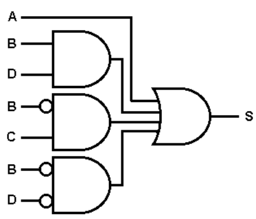
ABCD	*ex.*	
0000	0	**1**
0001	1	**0**
0010	2	**1**
0011	3	**1**
0100	4	**0**
0101	5	**1**
0110	6	**0**
0111	7	**1**
1000	8	**1**
1001	9	**1**
1010	A	ϕ
1011	B	ϕ
1100	C	ϕ
1101	D	ϕ



$S = A + BD + \bar{B}C + \bar{B}\bar{D}$

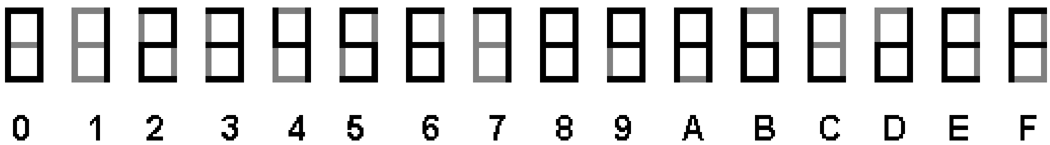
Observe que para representar de A a F em hexadecimal foi adotada a condição irrelevante, representada com o símbolo ϕ**.** Isto significa que qualquer resultado apresentado na saída será indiferente para nós, pois o circuito está sendo construído para representar apenas os valores decimais de zero a nove.

O circuito obtido é representado na figura abaixo :

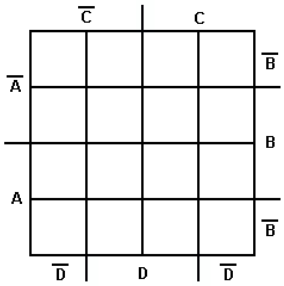


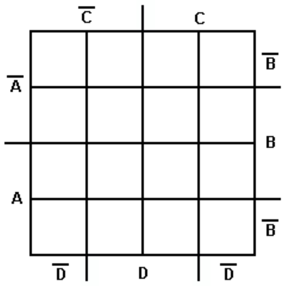
Exercícios Propostos :

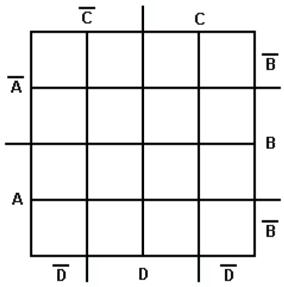
1) Determine os circuitos lógicos usados em um decodificador BCD 8421 para display de sete segmentos que represente os símbolos da forma mostrada abaixo :

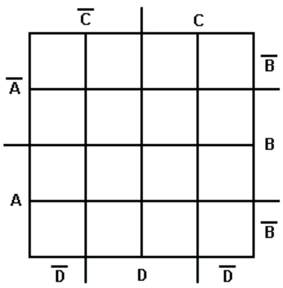


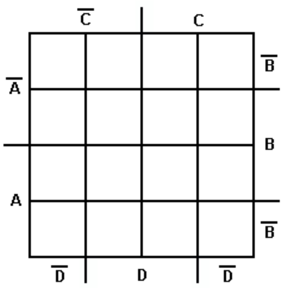
Código BCD 8421	Sa	Sb	Sc	Sd	Se	Sf	Sg
0000							
0001							
0010							
0011							
0100							
0101							
0110							
0111							
1000							
1001							
1010							
1011							
1100							
1101							
1110							
1111							

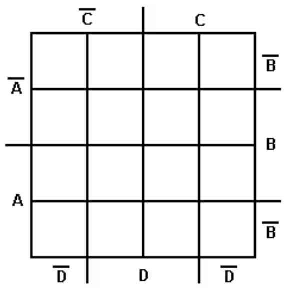


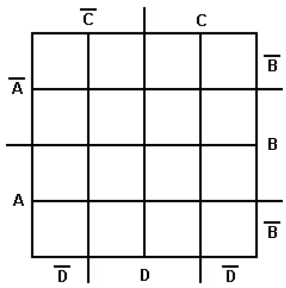












Circuitos Aritméticos

Vimos nas lições anteriores que os circuitos e portas lógicas executam funções matemáticas. Os circuitos que estudaremos agora fazem parte de um grupo muito importante, utilizado como base na construção da ULA (Unidade Lógica e Aritmética) dos computadores.

A ULA é um circuito lógico contido nos microprocessadores e microcontroladores, responsável pela execução das operações lógicas (E. OU, NÃO, NE, NOU, EXOR e EXNOR) e aritméticas durante o processamento de dados.

Circuito meio Somador :

Para construir um circuito capaz de executar a soma de dois números binários de um dígito (1 Bit) podemos utilizar o diagrama de Veitch-Karnaugh. Observe a tabela verdade desta operação mostrada abaixo :

A B	Soma (S)	Transporte (Ts)
0 0	0	**0**
0 1	1	**0**
1 0	1	**0**
1 1	0	**1**

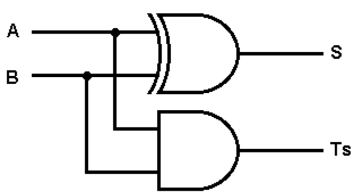
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 e “ vai um “

Fazendo a simplificação para as duas saídas do circuito, chegamos ao seguinte resultado :

A Soma $S = A \oplus B$ (A “OU Exclusivo” B)

O Transporte Ts (Também chamado de **Carry** ou “ Vai Um”) $Ts = A \cdot B$

O circuito do Meio Somador é mostrado na figura abaixo :



Circuito Meio Somador

Circuito Somador Completo :

Para executar a soma de números binários de mais de um dígito (Bit), é necessário que o circuito possa receber o transporte (Estouro ou Carry) do dígito anterior, caso o mesmo ocorra. Observe o Exemplo abaixo :

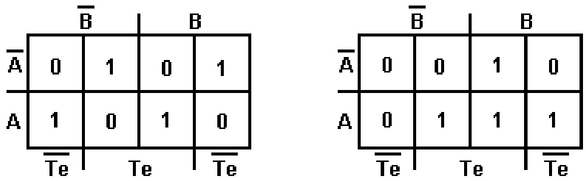
```
  1 1
0 0 1 1
0 0 0 1 +
-------
0 1 0 0
```

Ao realizarmos a soma do primeiro dígito da direita, teremos a soma 1 + 1 = 0 , com ocorrência de Estouro ou transporte de uma unidade para o segundo dígito. O mesmo ocorre na soma do segundo dígito

Podemos então montar a tabela verdade para este circuito, sendo que serão necessárias três entradas, as variáveis A e B que serão somadas e a entrada do transporte do dígito anterior Te. Como saídas teremos o resultado da soma S = A + B e o transporte ou Carry para o próximo dígito Ts.

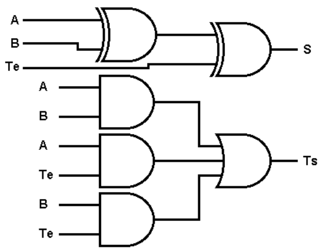
	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Aplicando o diagrama de Veitch-Karnaugh para as saídas S e Ts, teremos :

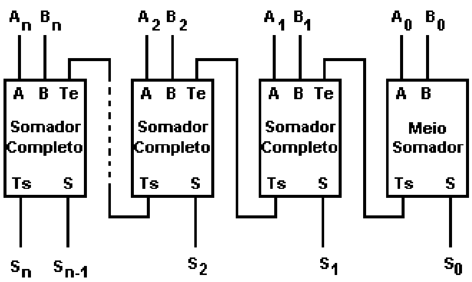


$S = A \oplus B \oplus Te$ $\quad$ $Ts = B.Te + A.Te + AB$

O circuito Somador Completo obtido é mostrado na figura abaixo :



Utilizando os blocos <u>Meio Somador</u> e <u>Somador Completo</u> é possível construir circuitos Somadores de tantos dígitos ou Bit's quantos forem necessários. A figura abaixo mostra um exemplo de um Somador de **n** Bit's :



Circuito Meio Subtrator

Assim como no caso da soma, é possível construir um circuito capaz de efetuar a subtração entre dois números binários. Caso estes números sejam de apenas um **bit** (dígito) utilizaremos o circuito Meio Subtrator.

A operação de subtração entre números binários é mostrada no quadro abaixo :

A B	Subtração (S)	Transporte (Ts)
0 0	0	**0**
0 1	1	**0**
1 0	1	**0**
1 1	0	**1**

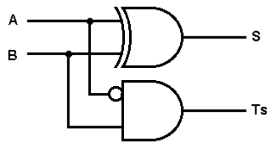
0 - 0 = 0
0 - 1 = 1 e “ empresta um “
1 - 0 = 1
1 - 1 = 0

Utilizando o diagrama de Veitch-Karnaugh para as duas saídas, chegamos ao seguinte resultado :

A Subtração $S = A \oplus B$ (A “OU Exclusivo” B)

O Transporte Ts (“ Empresta Um”) $Ts = \overline{A} . B$

O circuito do Meio Subtrator é mostrado na figura abaixo :



Circuito Meio Subtrator

Circuito Subtrator Completo :

É utilizado para efetuar a subtração entre dois números binários quando for preciso levar em consideração a ocorrência de empréstimo na subtração do dígito anterior. Será necessária a sua utilização na subtração entre dois números que possuam mais do que um Bit.

Para obter este circuito basta efetuar a simplificação pelo diagrama de Veitch-Karnaugh como foi feito no caso da Soma.

	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

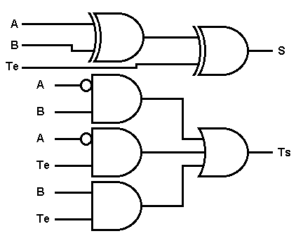
Aplicando o diagrama de Veitch-Karnaugh para as saídas S e Ts, teremos :

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

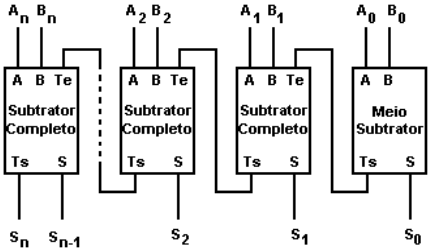
	$\overline{B}$	$\overline{B}$	B	B
$\overline{A}$	0	1	1	1
A	0	0	1	0
	$\overline{Te}$	Te	Te	$\overline{Te}$

$S = A \oplus B \oplus Te$ $Ts = B.Te + \overline{A}.Te + \overline{A}B$

O circuito Somador Completo obtido é mostrado na figura abaixo :



Da mesma forma como foi feito na operação de Soma, podemos agrupar os blocos Meio Subtrator e Subtrator Completo para efetuar a Subtração de números binários de tantos Bit´s quantos forem necessários. A figura abaixo mostra como isso pode ser feito:



Observe que o primeiro bloco utilizado é um circuito Meio Subtrator, pois na operação do primeiro dígito não há empréstimo. Do segundo até o último dígito é necessário utilizar o Subtrator Completo, pois é preciso levar em consideração a ocorrência de empréstimo da subtração do dígito anterior.

Exercícios Propostos :

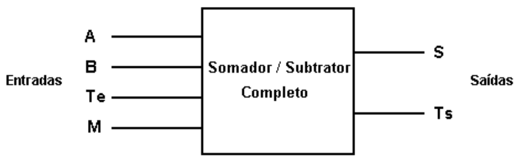
Desejamos construir um circuito que seja capaz de efetuar tanto a Soma como a Subtração entre dois números binários de **n** Bit´s. Para isso é necessário que o circuito possua uma entrada adicional para definir qual será a operação a ser executada. Chamaremos esta entrada de **M**. Complete a tabela verdade deste circuito (Somador/Subtrator) e determine a sua configuração utilizando o diagrama de Veitch-Karnaugh.

Tabela Verdade do circuito Somador / Subtrator

Soma
(M = 0)

Subtração
(M = 1)

M	A	B	Te	S	Ts
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		



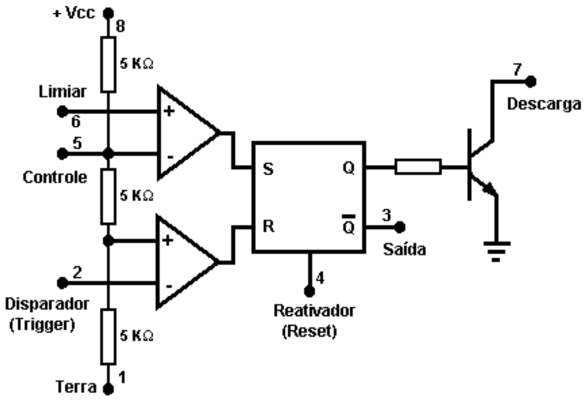
Circuito Integrado Temporizador (Timer) **555**

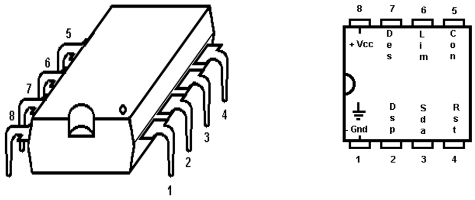
Apesar de não se tratar de um componente digital, faremos agora um estudo do Timer **555**. Este circuito Integrado é largamente utilizado como base de marcação de tempo em circuitos analógicos e também em circuitos digitais.

Devido à sua grande versatilidade, o Timer **555** tornou-se um padrão Industrial, podendo ser utilizado de inúmeras formas e configurações diferentes.

Seu código comercial pode mudar conforme o fabricante, porém o número 555 é comum a todos eles. Podemos citar como exemplo o LM 555, o NE 555, o µA 555, etc.

A figura abaixo mostra o diagrama esquemático simplificado do CI 555 :





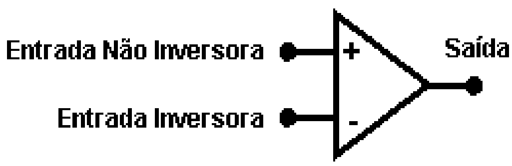
Encapsulamento DIL (Dual in Line)

Para que possamos compreender melhor o funcionamento do circuito, faremos um estudo de cada uma das partes que o formam. Observando a figura podemos identificar os blocos utilizados na construção do Circuito Integrado :

- Dois Amplificadores operacionais funcionando como Comparadores de Tensão
- Um Flip-Flop tipo RS
- Um Divisor de Tensão formado por três Resistores de 5 K Ω
- Um transistor utilizado como Chave

- Comparadores de Tensão :

O funcionamento dos Comparadores de Tensão neste circuito é bastante simples. Observe a figura



:

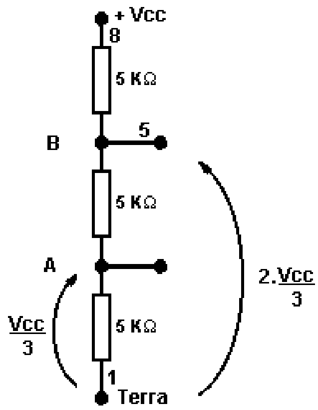
O componente possui duas entradas (Inversora '-' e Não Inversora '+') e uma saída. O nível de tensão na saída poderá ser alto (1) ou baixo (0), dependendo do nível de tensão nas entradas.

A saída será alta (1 ou +Vcc) sempre que a tensão na entrada Inversora '+" for mais alta do que a tensão na entrada Não Inversora "-".

No CI 555 são utilizadas duas tensões de referência nos comparadores : $\frac{Vcc}{3}$ e $2.\frac{Vcc}{3}$ conforme veremos adiante.

Divisor de Tensão :

É formador por três Resistores de 5 KΩ . Sua função é fornecer as tensões de referência para os Comparadores de Tensão. Sobre cada um dos resistores será aplicada uma tensão igual a um terço da tensão de alimentação Vcc.

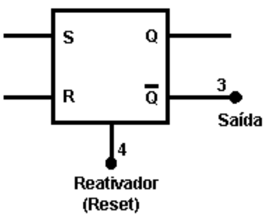


Sendo assim, entre o terminal do Terra e o primeiro resistor (ponto A) teremos uma tensão igual a um terço de Vcc (Vcc/3) aplicada à entrada Não Inversora do primeiro comparador (Disparador).

Entre o terminal do Terra e o segundo resistor (ponto B) teremos uma tensão igual a dois terços da tensão de alimentação (2.Vcc/3) que é aplicada ao segundo comparador (Controle).

Flip-Flop Tipo RS :

Este componente será estudado com maiores detalhes posteriormente, porém, para que possamos compreender o funcionamento do CI 555 faremos uma breve análise do seu funcionamento.

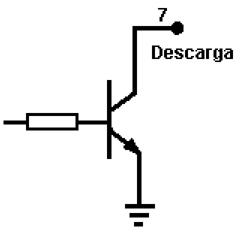


Trata-se de um componente Digital biestável, ou seja, suas saídas podem assumir apenas dois níveis de tensão ou níveis lógicos (zero ou um). As saídas são complementares. Enquanto Q estiver em nível lógico zero, a saída complementar $\overline{Q}$ estará em nível lógico um e vice-versa.

Quando a entrada S (Set) receber um nível lógico igual a um, a saída Q será levada para nível lógico um, ou seja, a operação Set leva a saída Q para o nível de tensão igual a Vcc. Quando a entrada R (Reset) receber um nível lógico igual a um, a saída Q será levada para nível lógico zero, ou seja, a operação Reset leva a saída Q para o nível de tensão igual zero volts.

Transistor como Chave :

Esta é uma configuração simples muito utilizada em circuitos eletrônicos. Observe que o resistor de base do transistor está conectado à saída Q do Flip-Flop.



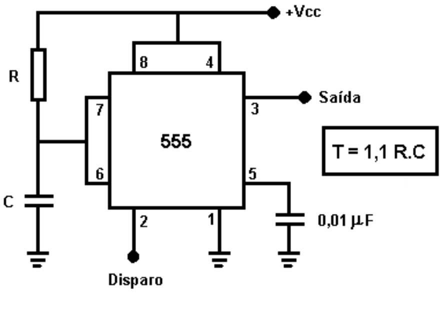
Quando a saída Q estiver em nível lógico um, fará com que a corrente de base leve o transistor à saturação, ou seja, a chave entre coletor e emissor será fechada, aterrando o terminal de coletor do transistor.

Este transistor é utilizado para descarregar capacitores externos utilizados como base de tempo.

Multivibrador Monoestável com CI 555 :

Nesta configuração o 555 é utilizado como temporizador simples, ou seja, uma vez disparado o pino 2, a saída permanecerá em nível lógico um (alta) até que transcorra o tempo determinado pelo resistor e capacitor externos.

O disparo do Monoestável ocorrerá toda vez que o pino 2 do CI for levado a zero volts. A figura abaixo mostra o diagrama esquemático desta configuração :



555 como Monoestável

Enquanto o circuito não for disparado, a saída Q do Flip-Flop interno em nível lógico um fará com que o transistor de descarga permaneça saturado, impedindo que o capacitor externo C seja carregado através do resistor externo R. A saída permanecerá em zero volts.

Quando ocorrer o disparo, a tensão no pino 2 será inferior a um terço da alimentação Vcc, ativando o comparador interno de disparo e provocando a operação de Reset no Flip-Flop interno.

Nesse momento a saída (Pino 3) será levada a nível lógico um (Vcc) e o transistor de descarga entrará na região de corte, permitindo que o capacitor C possa ser carregado pelo resistor R. Esta situação será mantida até que o capacitor atinja uma carga ligeiramente maior do que dois terços da tensão de alimentação, o que ativa o comparador de controle e provoca a operação de Set no Flip-Flop interno.

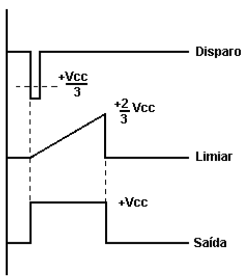
A saída então retorna ao nível lógico zero e o transistor interno satura, descarregando o capacitor C através do pino 7 do CI. Resumindo, a saída é ativada no momento do disparo, permanecendo assim durante o tempo de carga do capacitor, retornando então ao estado inicial. É importante observar que o pino 4 deve ser mantido em nível lógico alto (+Vcc) para permitir o funcionamento do circuito.

O período de tempo do circuito pode ser calculado pela expressão :

$$\boxed{T = 1{,}1\ R.C}$$

Onde : T = Período de tempo em segundos
R = Valor do Resistor externo em Ohms
C = Valor do Capacitor externo em Farad

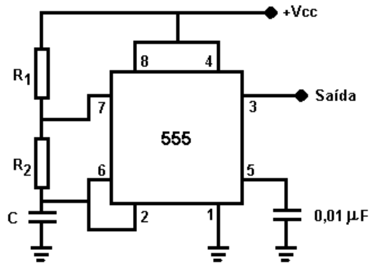
A figura abaixo mostra as formas de onda do Multivibrador Monoestável :



Multivibrador Astável com CI 555 :

Esta configuração é utilizada com muita freqüência como gerador de pulsos (Clock) para circuitos Digitais. Diferente do Monoestável, este circuito não necessita de disparo, gerando uma seqüência de pulsos retangulares contínua. Para interromper a seqüência de pulsos, basta levar o pino 4 do CI (Reset) a nível lógico zero (zero volts) .

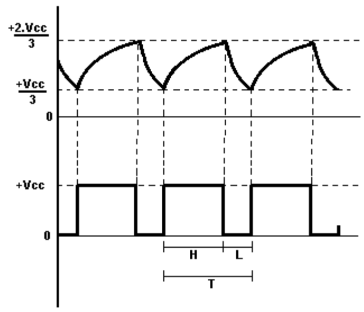
Seu funcionamento não é muito diferente do Monoestável. Quando o capacitor estiver descarregado, a tensão no pino 2 será menor do que um terço de Vcc, provocando o Reset do Flip-Flop interno. Isto leva a saída (pino 3) a nível lógico 1 e corta o transistor interno, permitindo que o capacitor inicie seu ciclo de carga através de R_1 e R_2. Quando a tensão no capacitor ultrapassar dois terços de Vcc, o comparador de controle efetua a operação de Set no Flip-Flop interno, levando a saída a zero e saturando o transistor interno, fazendo com que o capacitor inicie seu ciclo de descarga através de R_2 pelo pino 7 do CI . o capacitor terá sua tensão reduzida até que a mesma seja ligeiramente inferior a um terço de Vcc, o que ativa o comparador de disparo e provoca o Reset do Flip-Flop interno, iniciando outro ciclo.



555 como Astável

$$f = \frac{1,44}{(R1 + 2.R2).C}$$

Observe na figura que o capacitor C é carregado através de R_1 e R_2 , enquanto que a descarga e feita apenas através de R_2 . Isto faz com que o tempo de carga seja maior do que o tempo de descarga, provocando uma assimetria no sinal de saída, ou seja, a saída permanece mais tempo em nível alto do que em nível baixo. Esta assimetria pode ser observada nas formas de onda do circuito mostradas na figura abaixo :



Da forma como o circuito está configurado, a assimetria do sinal pode variar entre 50% e 100%. A relação entre o tempo em que a saída está alta e o período do sinal e´ expresso na equação abaixo :

$$D = \frac{H}{T} .100\%$$

Como a relação de assimetria depende dos valores dos resistores :

$$D = \frac{R1 + R2}{R1 + 2.R2} .100\%$$

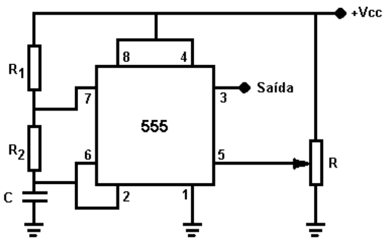
Podemos então determinar os tempos H e L pela equação : $T = H + L$

Como $f = \frac{1,44}{(R_1 + 2.R_2).C}$ então $T = \frac{(R1 + 2.R2).C}{1,44}$

Com essas expressões é possível determinar todos os períodos de tempo dos sinais gerados pelo circuito.

Oscilador Controlado pela Tensão (VCO) :

Podemos controlar a freqüência de oscilação do Multivibrador astável através do pino 5 do Circuito Integrado. Lembre-se que o pino 5 está conectado à Entrada Inversora do Comparador Interno de Controle. Variando a tensão neste terminal, mudamos o valor da tensão de controle, alterando o tempo de carga do capacitor externo. A figura abaixo mostra um exemplo desta aplicação :



Observe que o pino 5 foi conectado ao ponto central de um potenciômetro, o qual funciona como um divisor de tensão.

Alterando a posição do potenciômetro, provocamos uma mudança na tensão fornecida ao terminal de controle do Circuito integrado. Caso essa tensão seja maior, fará com que o capacitor demore mais tempo para atingi-la, aumentando o período e reduzindo a freqüência do sinal gerado pelo oscilador.

A tensão de controle pode ser fornecida por um potenciômetro ou por circuitos ativos mais complexos. A principal aplicação desse tipo de circuito são os Circuito de Fase Locada, também chamados de **PLL** .

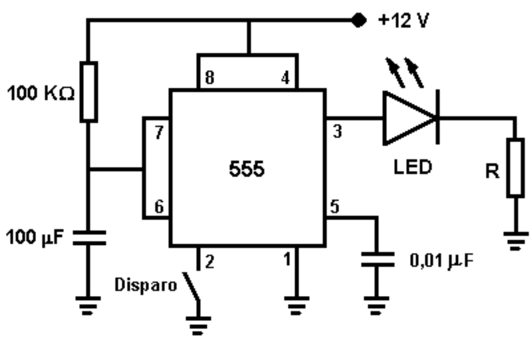
Limitações do 555 :

A desenvolver projetos com o CI 555 é necessário respeitar suas limitações. A seguir são fornecidas as especificações do TLC555M e do TLC555C fabricados pela Texas Instruments :

- R1 + R2 < 3,3 MΩ
- R1 e R2 > 1 KΩ
- C > 500 pF
- Corrente de Alimentação a 15 V = 360 μA
- Tensão de Alimentação - 4,5 V < Vcc < 18 V
- Dissipação Total = 600 mW
- Freqüência Máxima = 2,1 MHz
- Corrente Máxima de Saída – 100 mA

Exercícios Resolvidos :

1) Determine o tempo em que o LED do circuito abaixo permanecerá aceso após o disparo e qual o valor do resistor R para que a corrente no mesmo seja de 20 mA.



O tempo em que o LED permanecerá aceso é determinado pela expressão estudada acima :

T = 1,1 R.C

É importante observar que os valores do resistor e do capacitor devem ser convertidos para Ohms e Farads respectivamente, então :

$T = 1,1 \cdot 100x10^{3}.100x10^{-6}$ ou $T = 1,1 \cdot 100000. 0,0001$

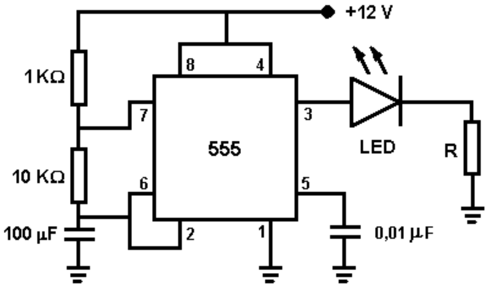
Então <u>T = 11 segundos</u>

O valor do resistor limitador de corrente do LED é calculado pela fórmula abaixo.

Considerando uma queda de tensão no LED de 1,8 V e convertendo a corrente para Ampéres:

$$R = \frac{Vcc - Vd}{IL} = \frac{12 - 1,8}{0,02} = \underline{510\ \Omega}$$

2) Dado o circuito abaixo, desenhe os gráficos das formas de onda das tensões de saída (pino 3) e de carga do capacitor (pinos 2 e 6) :



Para desenhar a forma de onda da tensão de carga do capacitor, precisamos determinar as tensões máxima e mínima atingidas.

$$V_{Max} = \frac{2.Vcc}{3} = \frac{2 . 12}{3} = 8 \text{ v}$$

$$V_{Min} = \frac{Vcc}{3} = \frac{12}{3} = 4 \text{ v}$$

Para o gráfico da tensão de saída é necessário calcular os períodos e a relação de assimetria :

O Período total do sinal é $T = \frac{(R1 + 2R2).C}{1,44} = \frac{(1000+20000).0,0001}{1,44} = \underline{1,458 \text{ s}}$

A relação de assimetria $D = \frac{(R1 + R2)}{(R1 + 2.R2)} .100\% = \frac{11000}{21000} .100\% = 52,4\%$

Podemos então determinar o tempo em que a saída permanece alta :

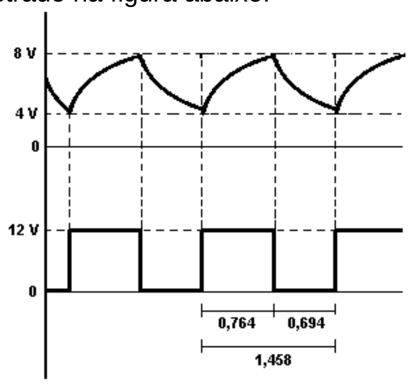
$H = \frac{D . T}{100} = \frac{52,4 . 1,458}{100} = \underline{0,764 \text{ s}}$

O tempo em que a saída permanece baixa : L = T – H

L = 1,458 – 0,764 = $\underline{0,694 \text{ s}}$

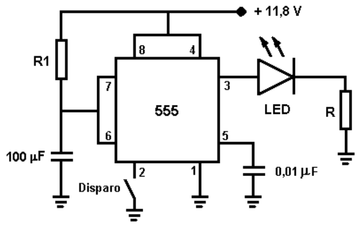
Observe que a assimetria está muito próxima de 50%, pois o valor de R2 é muito maior do que o valor de R1.

O gráfico das tensões é mostrado na figura abaixo:

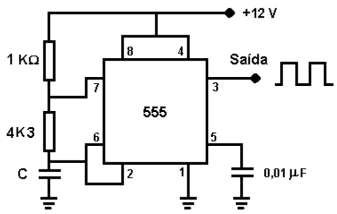


Exercícios Propostos :

1) Determine o valor do resistor R1 no circuito abaixo para que o LED permaneça aceso por 1 minuto e 50 segundos após o pulso de disparo e qual o valor de R para que a corrente no mesmo seja de 10 mA.



1) Determine o valor do capacitor C para que o circuito abaixo oscile em uma freqüência de 1 KHz e desenhe os gráficos das tensões de saída e de carga do capacitor.



Flip-Flop RS :

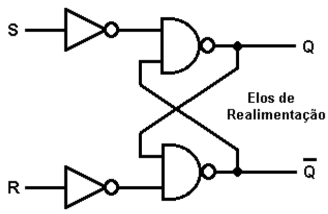
A Eletrônica Digital pode ser dividida para efeito de estudos em duas partes :

- Lógica Combinacional - É baseada em circuitos que possuem apenas uma saída, sendo que a saída depende exclusivamente das variáveis de entrada.

- Lógica Seqüencial - É baseada em circuitos nos quais o estado da saída depende das variáveis de entrada e também do estado anterior da saída, o qual permanece armazenado.

Os Flip-Flop’s são componentes construídos com portas lógicas, sendo a base da lógica seqüencial. Possuem basicamente duas saídas e duas entradas, alem de entradas auxiliares de controle. São chamados de componentes biestáveis, pois as saídas podem assumir dois níveis estáveis. Os níveis lógicos das saídas ficam armazenados enquanto não houver alteração nas entradas e a alimentação elétrica for mantida. Essa capacidade de reter números binários permite que o Flip-Flop seja muito utilizado como célula básica em circuitos de Memória de Computadores.

A figura abaixo mostra a configuração básica do Flip-Flop (FiFo) RS :



Podemos montar a tabela verdade do circuito levando em consideração os estados das variáveis de entrada (R e S) e o estado anterior da saída Q (Qa ou Q anterior) .

Para S e R iguais a zero, a saída Q assumirá o mesmo valor anterior da saída, ou Qa.

Para S = 0 e R = 1, a saída Q assumirá o valor igual a zero, qualquer que seja o seu estado anterior.

Para S = 1 e R = 0, a saída Q assumirá o valor igual a um, qualquer que seja o seu estado anterior.

Um problema ocorrerá quando as entradas S e R forem iguais a um. Observe na figura. R e S iguais a um fazem com que ambas as portas NE do circuito recebam zero nas suas entradas. Como a porta NE efetua a operação de produto, zero multiplicado por qualquer nível lógico recebido pelos elos de realimentação faz com que as saídas de ambas as portas sejam iguais a um. Esta é uma situação de instabilidade, pois as saídas do FiFo $\overline{Q}$ e sua saída complementar Q estarão no mesmo nível lógico. É portanto um estado não permitido.

A tabela verdade do Flip-Flop RS é mostrada abaixo :

S	R	Qf
0	0	Qa
0	1	0
1	0	1
1	1	Não Permitido

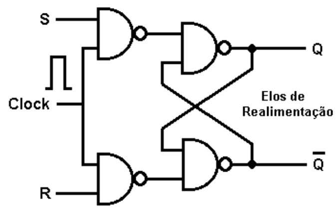
Onde :

Qf = Estado final da saída Q

Qa = Estado anterior da saída Q
R e S = Entradas Set e Reset

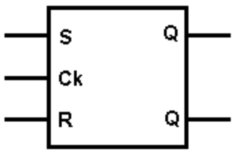
Flip-Flop RS comandado por um pulso de Clock :

O circuito estudado anteriormente possui o inconveniente de alterar o nível lógico das saídas instantaneamente caso haja alteração nas entradas S e R. Isto é possível adicionando uma lógica de controle conforme mostra a figura abaixo :



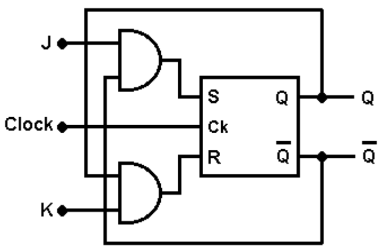
Neste circuito poderão ocorrer alterações nos níveis lógicos das entradas S e R sem que haja alteração na saída. Isto ocorre porque as duas portas NE das entradas impedem que essas alterações sejam transferidas para a parte posterior do circuito. Somente haverá alteração na saída caso seja aplicado um nível de tensão igual a um no terminal de Clock. Isto faz com que os níveis complementares das entradas S e R sejam aplicados à parte posterior do FiFo, provocando a alteração apropriada na saída.

Para tornar a representação mais simples, passaremos a utilizar para o FiFo RS o símbolo mostrado na figura abaixo :



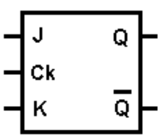
Flip-Flop JK :

O FiFo RS possui um estado não permitido para as entradas, ou seja, vai para uma situação instável quando as entradas R e S forem iguais a um. Este problema é solucionado com o Flip-Flop JK. Isto é conseguido realimentando-se o Flip-Flop RS da maneira mostrada na figura abaixo :



Podemos observar que a realimentação através das portas E fará com que o valor assumido pela saída quando as entradas J e K forem iguais a um será o complemento do nível lógico anterior da saída.

A tabela verdade do FiFo JK e seu símbolo são mostrados abaixo :



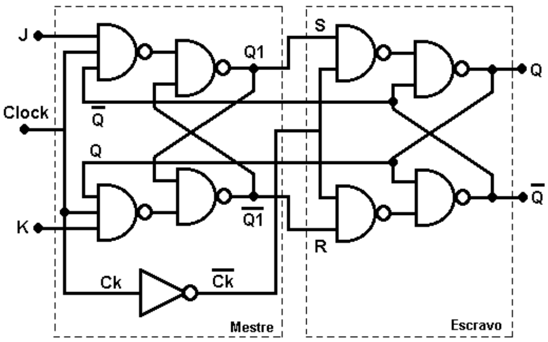
J	K	Qf
0	0	Qa
0	1	0
1	0	1
1	1	$\overline{Qa}$

Flip-Flop JK Mestre - Escravo (Master/Slave)

Conforme estudamos anteriormente, o Flip-Flop JK resolve o problema apresentado no Flip-Flop RS de possuir uma situação de instabilidade. A presença do terminal de Clock também torna o circuito melhor, pois permite que a mudança de estado na saída somente ocorra quando este terminal estiver em nível lógico um.

Depois de todas essas melhorias, ainda pode ocorrer uma situação inconveniente no circuito. Enquanto a entrada de Clock estiver em nível lógico um, mudanças de estado nas entradas, provocadas por ruído ou não, podem alterar o estado da saída.

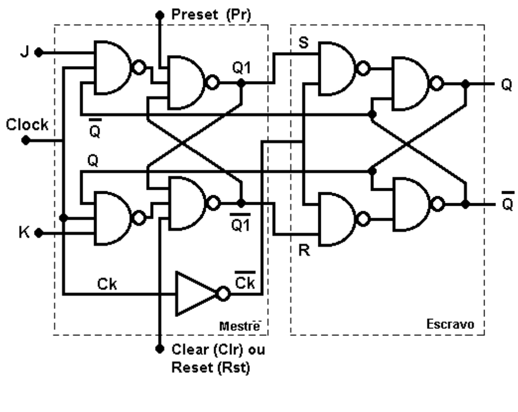
O Flip-Flop JK Mestre Escravo (Master/Slave) tem por finalidade impedir que esse problema ocorra. A figura abaixo mostra a sua configuração básica :

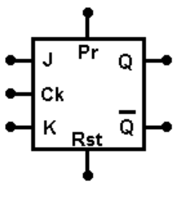


Observe na figura que o circuito é dividido em duas partes : A primeira trata-se de um Flip-Flop JK básico, funcionando como Mestre ou comandando a segunda parte formada por um Flip-Flop RS (Escravo). Entre os terminais de Clock do Mestre e do Escravo encontra-se um inversor . Quando o circuito receber um pulso de Clock, o Mestre estará pronto para receber os níveis lógicos nas entradas J e K, enquanto que o terminal de Clock do Escravo será mantido em nível lógico zero, impedindo que qualquer variação nas entradas seja transferida para a saída. Uma vez posicionadas as entradas, o Clock é levado a nível lógico zero, bloqueando as entradas J e K e mudando o Clock do escravo de zero para um. Nesse momento, os níveis lógicos presentes nas saídas do Circuito Mestre são transferidas para a saída do circuito Escravo sem possíveis perturbações causadas por ruídos.

Flip-Flop JK com entradas Preset e Clear :

Em muitas aplicações é preciso estabelecer uma condição inicial aos Flip-Flop’s. Para isso foram acrescentados ao circuito os terminais de Preset e Clear. Sua função é muito simples. Quando o terminal de Preset for levado a nível lógico zero, fará com que o circuito assuma um valor inicial para a saída Q igual a um. O terminal Clear (também chamado de Reset) funciona de forma exatamente contrária, ou seja, uma vez levado a zero faz com que a saída Q do circuito assuma o nível lógico zero. A figura abaixo mostra como isso é feito :



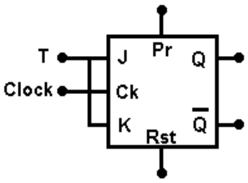


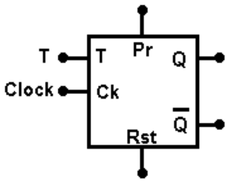
Símbolo do Flip-Flop JK

Flip-Flop Tipo T

O Flip-Flop Tipo T é um caso particular do FiFo JK. É conseguido simplesmente conectando as duas entradas J e K em Curto-circuito. Neste caso teremos apenas duas situações para as entradas, ou seja, ambas iguais a zero ou iguais a um.

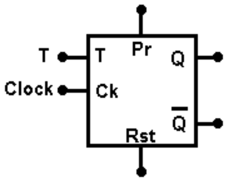
A figura abaixo mostra o FiFo tipo T e sua respectiva Tabela Verdade :





J	K	T	Qf
0	0	0	Qa
0	1	Não Existe	-
1	0	Não Existe	-
1	1	1	$\overline{Qa}$

Resumindo, mostramos abaixo o símbolo e a Tabela Verdade do FiFo Tipo T :

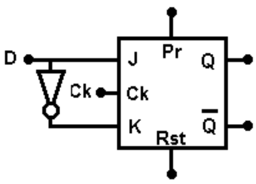


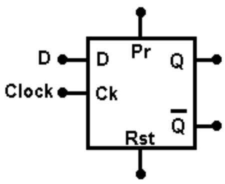
T	Qf
0	Qa
1	$\overline{Qa}$

Flip-Flop Tipo D

O Flip-Flop Tipo D é outro caso particular do FiFo JK. É conseguido simplesmente aplicando níveis lógicos complementares (invertidos) às duas entradas J e K . Neste caso teremos também apenas duas situações para as entradas, ou seja, J = 0 e K = 1 ou J = 1 e K = 0.

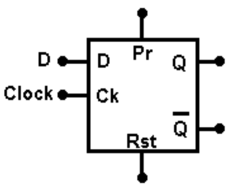
Isso é conseguido interligando as entradas J e K através de um inversor como mostra a figura abaixo :





J	K	T	Qf
0	0	Não Existe	-
0	1	0	0
1	0	1	1
1	1	Não Existe	-

Resumindo, mostramos abaixo o símbolo e a Tabela Verdade do FiFo Tipo D :



D	Qf
0	0
1	1

Exercícios Propostos :

1) Qual a diferença entre Lógica Combinacional e Lógica Sequencial ?

2) O que são Componentes Biestáveis ?

3) Quais as variáveis que determinam o estado da saída de um Flip-Flop?

4) Qual a vantagem de um Flip-Flop RS possuir um terminal de comando por pulso (Clock) ?

5) Explique a diferença entre um Flip-Flop RS e um Flip-Flop JK .

6) Qual a diferença entre um Flip-Flop JK simples e um Flip-Flop JK Mestre-Escravo ?

7) Para que servem as entradas Preset e Clear em um Flip-Flop ?

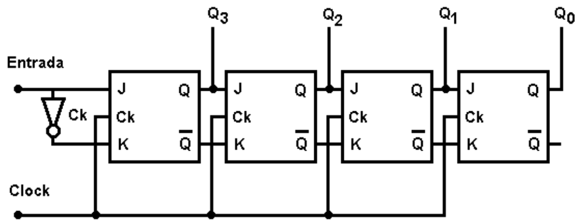
8) Descreva as diferenças entre o Flip-Flop JK , o Flip-Flop Tipo T e o Flip-Flop Tipo D .

Registrador de Deslocamento (Shift Register)

Como vimos na lição anterior, o Flip-Flop pode armazenar uma informação na forma de um número binário de um Bit (dígito). Nos casos em que for necessário armazenar uma informação com um número maior de Bits, o Flip-Flop será insuficiente, sendo necessária a utilização do Shift Register.

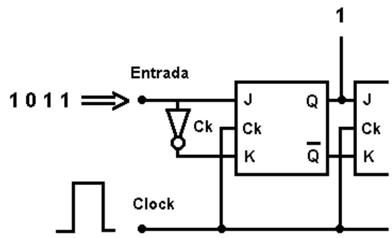
É um circuito bastante simples, sendo construído com Flip-Flop's RS ou JK Master/Slave de forma que as saídas do Flip-Flop anterior alimentem as entradas do Flip-Flop seguinte.

Além de armazenar uma informação binária, o Registrador de Deslocamento é capaz de efetuar a conversão série-paralelo e paralelo-série. A figura abaixo mostra um Shift Register básico de quatro Bits :

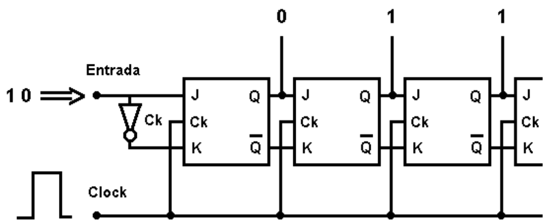


Nesta configuração o circuito é capaz de armazenar uma informação binária de quatro Bits, sendo que a entrada é feita de forma serial, enquanto que a saída é paralela.

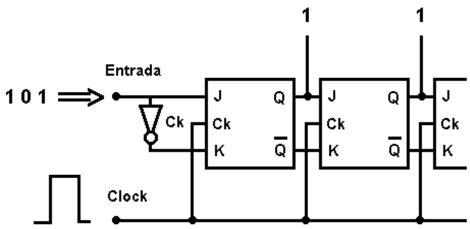
Por exemplo, caso se queira armazenar o número binário 1011. Devemos injetar o Bit menos significativo na entrada do circuito e aplicar um pulso no terminal de Clock. Sendo a entrada J igual a um seu complemento (zero) será aplicado à entrada K pelo inversor. Observe que o primeiro Flip-Flop está configurado como Flip-Flop Tipo D. Ao receber o pulso de Clock, faz com que a saída Q seja igual à entrada J, ou seja, ocorre uma cópia do nível lógico da entrada para a saída. Os demais Flip-Flop's não sofrem modificação nessa primeira etapa.



Na segunda etapa, as saídas do primeiro Flip-Flop alimentam as entradas do segundo. O segundo dígito do número a ser armazenado deve ser aplicado à entrada do circuito. Aplicando-se outro pulso de Clock, o segundo Flip-Flop transfere o valor de sua entrada J para a saída, assim como o primeiro. O resultado é que teremos o primeiro Bit armazenado no segundo Flip-Flop e o segundo dígito armazenado no primeiro. A figura representa esta operação :

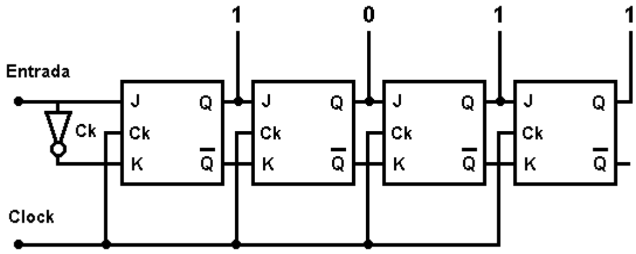


A operação é repetida para o terceiro e para o quarto Bit :



Após o quarto e último pulso de Clock, cada uma dos Flip-Flop's terá armazenado um dos dígitos do número binário. Este número permanecerá armazenado até que o circuito receba um novo número e enquanto a alimentação elétrica for mantida. Este é princípio básico da memória RAM Estática (Cache) utilizada em computadores.

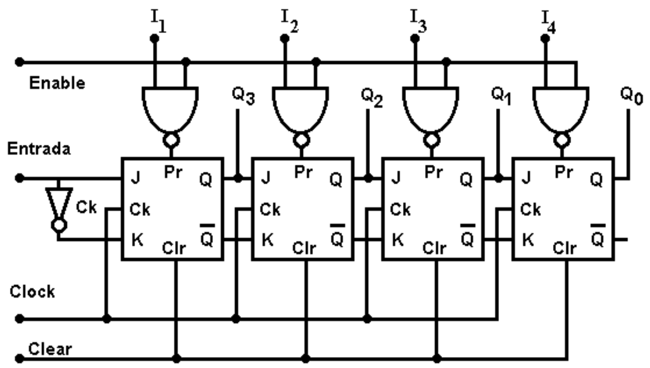
A figura abaixo mostra como fica o circuito após o armazenamento :



Observe que o número está disponível no formato paralelo, ou seja, todos os Bits podem ser transmitidos ao mesmo tempo através das saídas Q3, Q2, Q1 e Q0.

Conversor Paralelo – Série :

É um circuito muito semelhante ao estudado anteriormente, tendo a mesma capacidade de armazenar números binários. A diferença é que a entrada do número a ser armazenado deve ser feita no formato paralelo, ou seja, todos os Bits do número são fornecidos para o circuito ao mesmo tempo. A figura abaixo mostra um circuito para armazenar um número de quatro Bits :



Neste circuito os Bits do número binário são aplicados ao circuito pelos terminais I_1, I_2, I_3 e I_4. Quando a entrada Enable (Habilita) for levada a nível lógico um, fará com que os níveis lógicos presentes nas entradas I_1, I_2, I_3 e I_4 sejam transferidos para as saídas Q_3, Q_2, Q_1 e Q_0. Para obter os dados no formato serial, basta aplicar pulsos de Clock ao circuito, obtendo os Bits um a um na saída Q_0 .

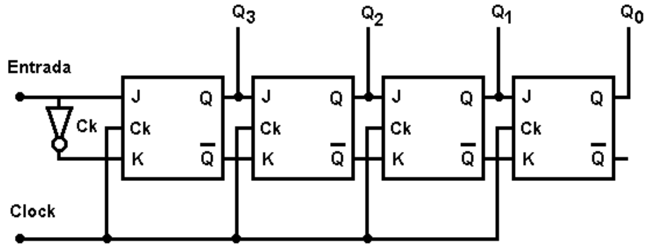
Como este circuito é igual ao estudado anteriormente, poderá ser utilizado tanto para a conversão Paralelo – Série como para a conversão Série – Paralelo, bastando para isso mudar a entrada e a saída de forma adequada para a aplicação desejada.

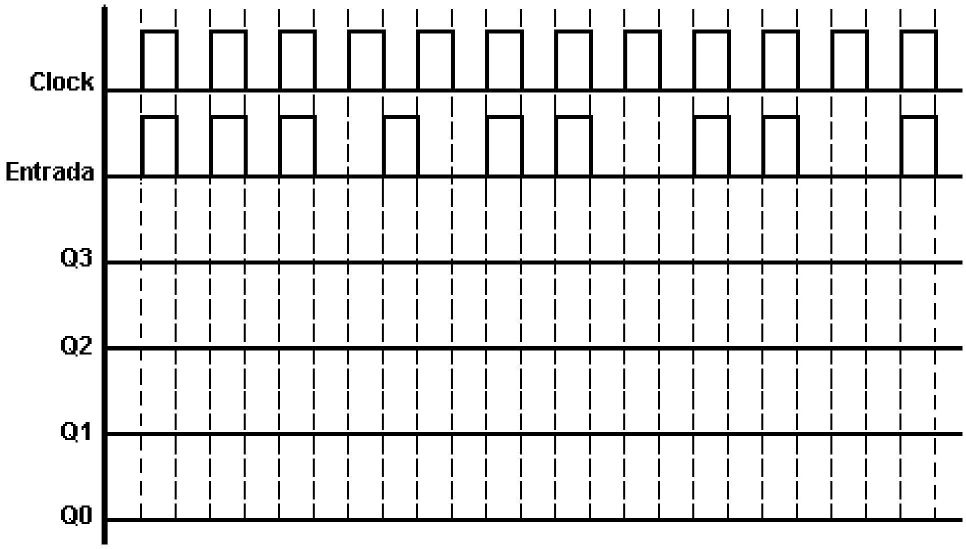
O terminal Clear serve para limpar (Reset) o circuito para que um novo número possa ser armazenado.

Esse tipo de circuito serve para a construção de Interfaces Paralelo – Serial, Serial – Paralelo, Modem de computadores, etc.

Exercícios Propostos :

1) Desenhe os gráficos das formas de onda nas saídas Q_3, Q_2, Q_1 e Q_0 do circuito abaixo :





2) Descreva a diferença entre o circuito conversor Série – Paralelo e o Conversor Paralelo – Série.

Circuitos Contadores

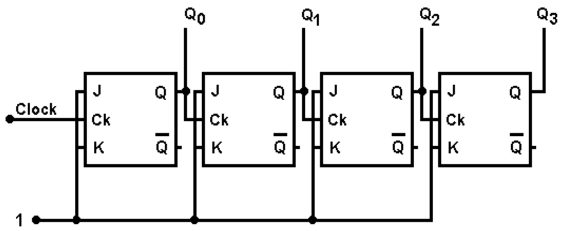
São circuitos Digitais Seqüenciais comandados por pulsos de Clock que tem como componente básico o Flip-Flop. São utilizados para gerar uma seqüência pré – determinada em circuitos divisores, seqüenciamento de operações em máquinas, contagens, etc.
São divididos em dois grupos distintos :

- Contadores Assíncronos : São aqueles em que os componentes de cada etapa são acionados por pulsos de Clock diferentes, sem que haja sincronismo entre eles.

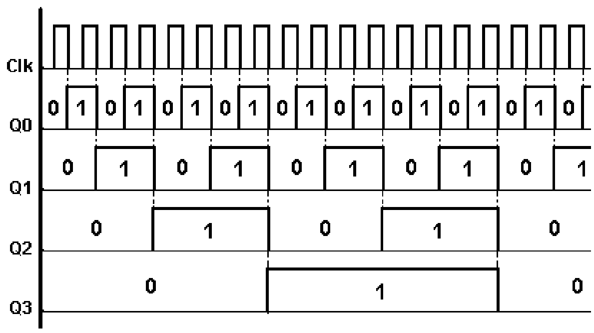
- Contadores Síncronos : São aqueles em que os componentes de cada etapa são acionados pelo mesmo pulso de Clock, ou seja, as entradas de Clock são curto – circuitadas, mantendo-as em sincronismo .

Contador Assíncrono Crescente :

É utilizado para gerar em suas saídas o Código BCD 8421 em seqüência . É construído com Flip-Flop's do Tipo JK com as entradas J e K mantidas em nível lógico um, de forma que a cada pulso de Clock recebido inverte o nível lógico da saída. É necessário utilizar Flip-Flop's JK Master – Slave acionados ou sensíveis a rampa de descida, ou seja, a mudança no estado da saída ocorre quando a entrada de Clock perceber uma transição de um para zero. A figura abaixo mostra um circuito de quatro Bits :



A saída Q do primeiro Flip-Flop alimenta a entrada de Clock do segundo, cuja saída alimenta a entrada de Clock do terceiro, e assim sucessivamente até o último. Cada um dos Flip-Flop's muda o estado de sua saída quando sua entrada de Clock perceber uma transição de um para zero.
Nessas condições podemos observar que o sinal de saída de cada FiFo terá a metade da freqüência do sinal de entrada. Isso permite que o FiFo possa também ser utilizado como **Divisor por dois** quando configurado dessa forma (J = K = 1).
A figura abaixo mostra as formas de onda obtidas nas saídas do contador :



Observe que o sinal da saída Q0 possui o dobro do período ou a metade da freqüência do sinal de Clock, enquanto que a saída Q1 tem a metade da Freqüência de Q0 e assim por diante.

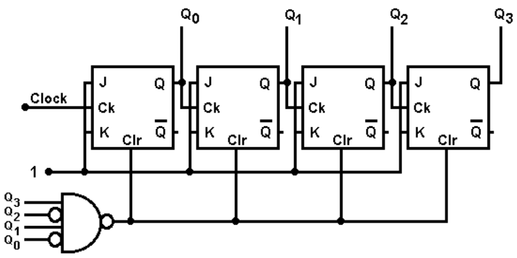
No início, antes do circuito começar a receber os pulsos de Clock, os estados das saídas Q0, Q1, Q2 e Q3 são **0000.** Após o primeiro pulso as saídas vão para **0001** gerando o Código BCD 8421 como mostra a tabela abaixo:

Clock	Q_3	Q_2	Q_1	Q_0	Hexa
1º	0	0	0	0	0
2º	0	0	0	1	1
3º	0	0	1	0	2
4º	0	0	1	1	3
5º	0	1	0	0	4
6º	0	1	0	1	5
7º	0	1	1	0	6
8º	0	1	1	1	7
9º	1	0	0	0	8
10º	1	0	0	1	9
11º	1	0	1	0	A
12º	1	0	1	1	B
13º	1	1	0	0	C
14º	1	1	0	1	D
15º	1	1	1	0	E
16º	1	1	1	1	F
17º	0	0	0	0	0
18º	0	0	0	1	1

Contador de Década Assíncrono

Para construir um circuito capaz de contar de zero a nove (Década), utilizaremos como base o mesmo contador assíncrono utilizado anteriormente, bastando adicionar uma pequena lógica de controle. Como desejamos contar até nove, o circuito deverá sofrer a operação de Reset toda vez que a contagem atingir Dez.

A forma como isso pode ser feito é mostrada na figura abaixo :



O circuito executa a contagem normalmente de zero até nove. Observe que os valores das saídas são injetados na lógica de controle formada pela porta NE e os inversores. Quando o valor das saídas for igual a Dez decimal, "A" hexadecimal ou 1010 binário, fará com que a saída da porta NE vá para nível lógico zero, provocando instantaneamente a operação de Reset nos Flip-Flop's, levando as saídas para 0000.

Sendo assim, o contador efetua a contagem de zero a nove, reiniciando em zero no próximo pulso de Clock.

Os inversores utilizados nas entradas da porta NE podem ser retirados, bastando injetar nessas entradas as saídas complementares de Q_2 e Q_0 obtidas nos Flip-Flop's.

Exercícios Propostos :

1) Explique a diferença entre contadores Assíncronos e Síncronos.

2) Para que servem os Circuitos Contadores ?

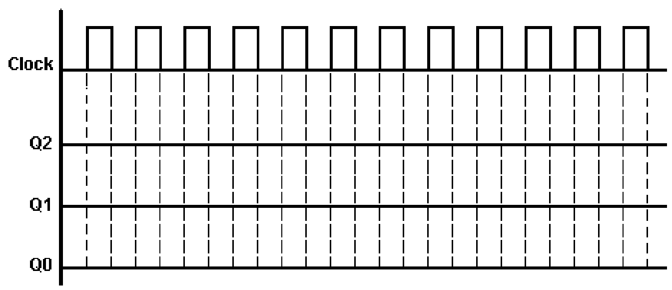
3) Qual o componente básico utilizado na construção dos Circuitos Contadores ?

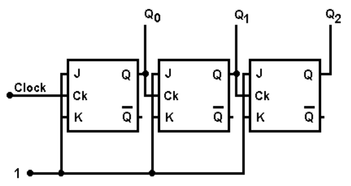
4) Em que condições é possível utilizar o Flip-Flop JK como Divisor por dois ?

5) Podemos substituir o Flip-Flop JK pelo Flip-Flop RS no Circuito Contador estudado ? Por que ?

6) De que forma podemos converter um Contador BCD 8421 de quatro Bits em um Contador de Década ?

7) Desenhe as formas de onda das saídas do circuito abaixo :



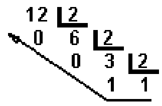


Contador Seqüencial de Zero a **n**

Como vimos anteriormente, é possível construir circuitos contadores para seqüências diferentes do código BCD 8421, bastando para isso adicionar uma lógica de controle para comandar os Flip-Flop's nos momentos apropriados. O primeiro passo na construção desses circuitos é estabelecer o número de componentes necessários. Para isso utilizaremos a técnica de conversão Decimal – Binário estudada no início do curso.

O segundo passo é determinar a lógica de controle para o circuito. Como a contagem desejada é de zero a **n**, é necessário que a lógica de controle atue sobre os Flip-Flop's quando a contagem atingir **n+1**.

Por Exemplo, desejamos construir um contador de zero a 12 decimal (C hexadecimal). O primeiro passo é determinar quantos Flip-Flop's serão necessários. Façamos então a conversão para binário :



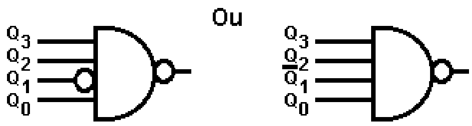
O resultado da conversão para binário é $12_d = 1100_b$. O número binário resultante é de quatro bits, sendo portanto necessário utilizar quatro Flip-Flop's no circuito.

O número imediatamente superior a **n** é 13 decimal. Basta então somar uma unidade a **n** :

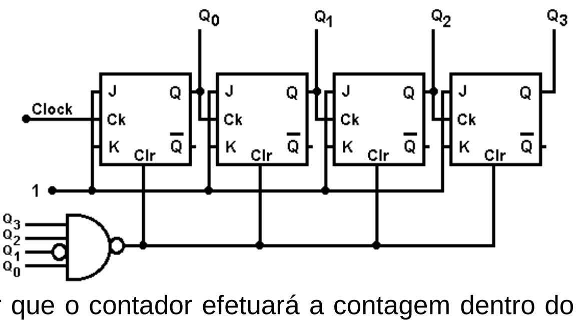
$$(n + 1) = 1100 + 1 = 1101 \text{ (13 decimal ou D hexadecimal)}$$

Para determinar a lógica de controle basta verificar quais dígitos do número binário **(n+1)** são iguais a zero. Para essas variáveis será necessária a utilização de inversores ou da variável complementar (barrada). Este é o caso do segundo dígito do exemplo, relacionado à variável de saída Q_1.

A lógica de controle a ser utilizada no circuito é mostrada abaixo :



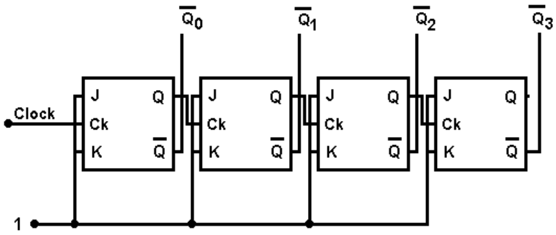
O circuito final é mostrado na figura abaixo :



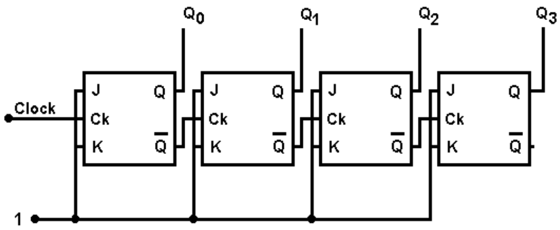
Podemos observar que o contador efetuará a contagem dentro do Código BCD 8421 normalmente até 12 decimal ou 1100 binário. No próximo pulso de Clock recebido, as saídas assumem o valor 1101 (13 decimal), o que leva a porta NE para nível lógico zero, provocando o Reset do circuito e levando instantaneamente as saídas para 0000, reiniciando a contagem.

Contadores Assíncronos Decrescentes

O circuito básico para obtermos uma contagem decrescente é o mesmo, bastando apenas mudar os terminais de saída ou de Clock. Uma contagem crescente é iniciada em 0000. As saídas complementares (barradas) dos Flip-Flop’s estarão em 1111. Então uma das formas de se obter uma contagem decrescente é a retirada da seqüência através das saídas complementares como mostra a figura abaixo :



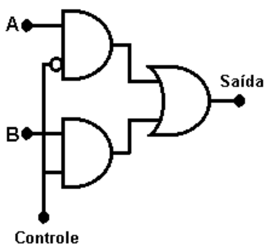
Uma outra forma possível é injetando nas entradas de Clock as saídas complementares. Esta opção será útil quando desejarmos construir um contador Crescente e Decrescente. A figura abaixo mostra essa possibilidade :



Contador Crescente – Decrescente :

Conforme foi visto no item anterior, para fazer um contador funcionar como crescente ou decrescente basta mudar a saída injetada na entrada de Clock do próximo Flip-Flop. Podemos construir uma lógica de controle que permita a um mesmo Circuito Contador funcionar como Crescente ou Decrescente.

Esta lógica é muito simples. Observe a figura abaixo :

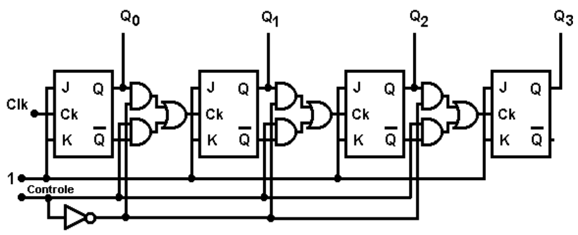


Esta lógica de controle é capaz de selecionar qual das variáveis de entrada será transferida para a saída. É formada por duas portas E que efetuam o produto entre as variáveis de entrada e o nível lógico presente no terminal de controle.

Quando o controle for igual a zero, teremos na porta E superior o produto A . 1 = A e na porta E inferior o produto B . 0 = 0. Somando tudo na porta OU, teremos na saída S = A + 0 = A

Mudando o controle para um teremos exatamente o contrário, ou seja, a saída S = B.

Aplicando esta lógica de controle ao Circuito Contador poderemos selecionar qual saída dos Flip-Flop’s será injetada nas entradas de Clock, alterando o contador para Crescente ou Decrescente conforme o nível lógico presente na entrada de controle conforme mostra a figura abaixo :



Exercícios Propostos :

1) Desenhe o circuito de um Contador Assíncrono Crescente de Zero a Cinco .

2) Desenhe o circuito de um Contador Assíncrono Crescente de Zero a Vinte e Três .