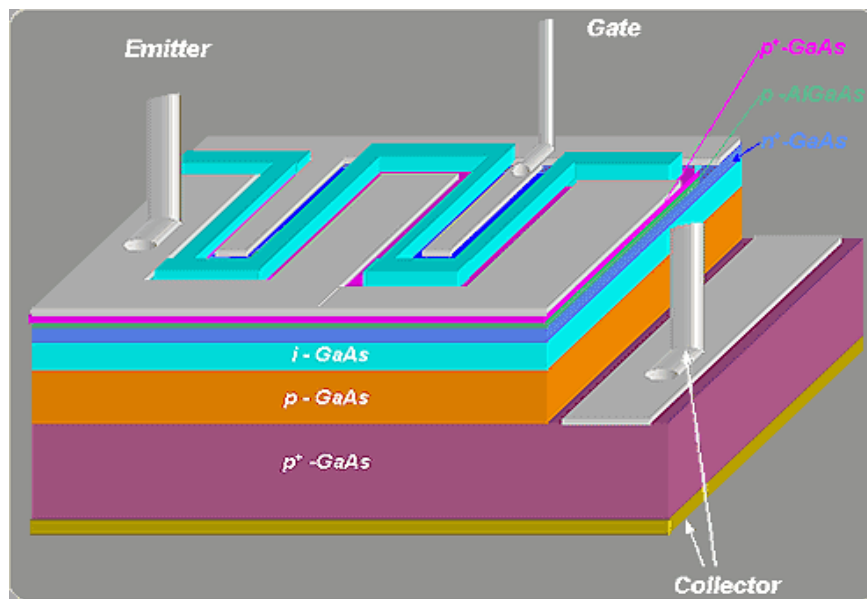


# TECNICAS COM SISTEMAS DIGITAIS

© Prof. Engº Luiz Antonio Vargas Pinto



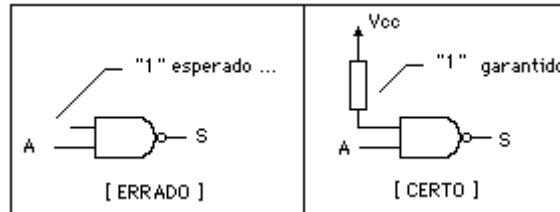


# Índice

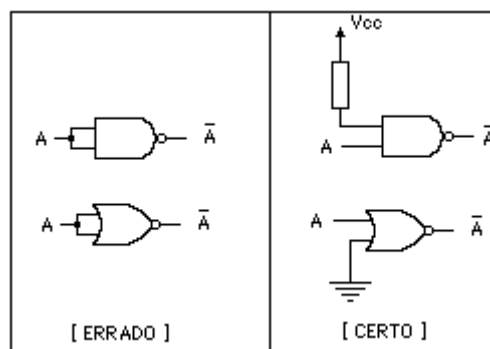
<a href="#">Índice.....</a>	<a href="#">2</a>
<a href="#">Introdução.....</a>	<a href="#">3</a>
<a href="#">Ruído (Bounce).....</a>	<a href="#">3</a>
<a href="#">Transistor como chave digital.....</a>	<a href="#">4</a>
<a href="#">Circuito Tri-State.....</a>	<a href="#">5</a>
<a href="#">Oscilador astável a transistor.....</a>	<a href="#">5</a>
<a href="#">Oscilador astável a CI.....</a>	<a href="#">5</a>
<a href="#">Flip Flop tipo RS.....</a>	<a href="#">6</a>
<a href="#">Flip Flop tipo JK.....</a>	<a href="#">6</a>
<a href="#">Configurações Tipo D e Tipo T.....</a>	<a href="#">7</a>
<a href="#">Aplicação especial de Flip-Flop.....</a>	<a href="#">7</a>
<a href="#">Demultiplexador.....</a>	<a href="#">7</a>
<a href="#">Latch.....</a>	<a href="#">8</a>
<a href="#">Display 7 segmentos.....</a>	<a href="#">8</a>
<a href="#">Osciladores com o CI555.....</a>	<a href="#">9</a>
<a href="#">Década contadora.....</a>	<a href="#">9</a>
<a href="#">Counter.....</a>	<a href="#">10</a>
<a href="#">Tecnologia dos contadores síncronos.....</a>	<a href="#">10</a>
<a href="#">Introdução.....</a>	<a href="#">10</a>
<a href="#">Flip Flops.....</a>	<a href="#">10</a>
<a href="#">Contadores.....</a>	<a href="#">11</a>
<a href="#">Análise de cada JK (Mapa de Karnaught).....</a>	<a href="#">12</a>
<a href="#">Tecnologia dos contadores Assíncronos.....</a>	<a href="#">14</a>
<a href="#">Circuitos Aritméticos.....</a>	<a href="#">14</a>
<a href="#">Generalidades.....</a>	<a href="#">15</a>

## Introdução

1º: TTL não reconhece seguramente o nível "1" quando seus terminais estiverem aberto, sendo necessário o uso de um resistor Pull Up para garantir o nível. Quando isto é para nível "0", caso de lógica inversa, usamos resistor ao terra com Pull Down.



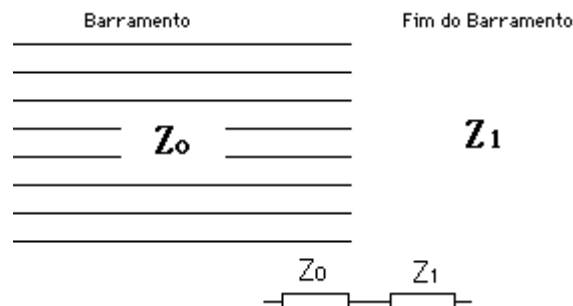
2º: Para o uso de portas inversoras devemos proceder com o uso de resistor Pull Up e Pull Down para garantir os níveis.



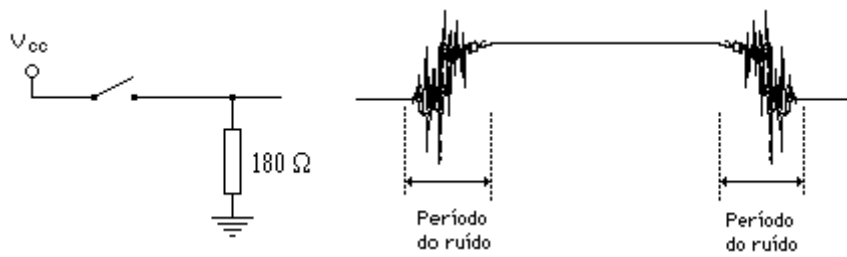
## Ruído (Bounce)

A transição de nível entre "0" e "1", que é um fato comum em um barramento de dados digital causa um efeito de transitório normalmente indesejável. A oscilação causada pela instabilidade de nível, aliada ao fato de que componentes reais não trabalham com 0 e 5 Volt de fato mas com faixas de tensão conforme visto na família TTL ou ainda CMOS fazem com que um determinado nível lógico leve um determinado tempo para ser de fato reconhecido.

Conforme a velocidade de execução de uma determinada porta lógica faz com que muitos sinais fiquem irreconhecíveis. Este exemplo é muito mais claro quando citamos o exemplo a barra de resistores existentes no barramento de uma CPU em um sistema microprocessado. Qual sua função? Justamente tentar evitar ao máximo a oscilação do barramento proporcionando o perfeito casamento de impedância existente entre o final do barramento e o próximo meio: o ar, onde é claro que as impedâncias são diferentes.



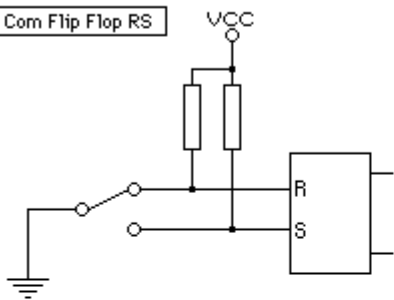
Durante o período de transição esse sinal, chamado de **BOUNCE** deve ser eliminado:



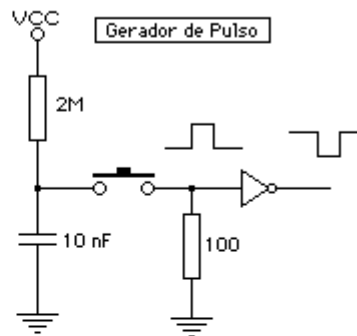
E esse **BOUNCE** pode e deve ser eliminado. Para isto lançamos mão do uso de chaves que, atrasando o sinal de saída compensam o ruído. São estas:

### Tipos de Chaves com Debounce

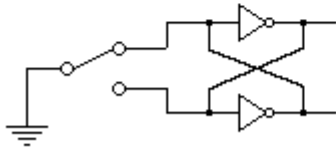
Com Flip Flop RS



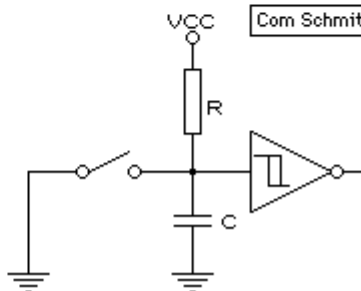
Gerador de Pulso



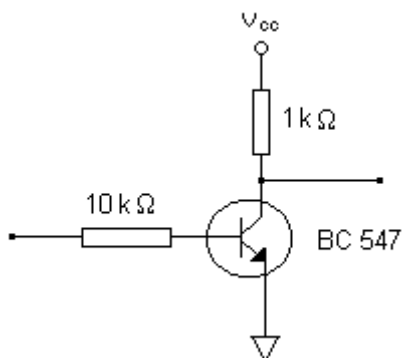
Circuito Simples



Com Schmitt Trigger

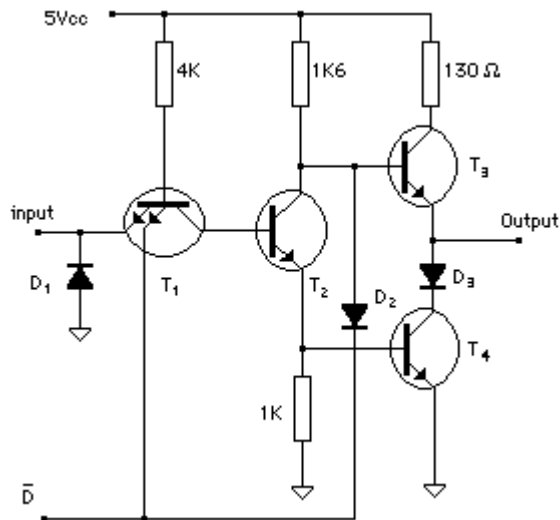


## Transistor como chave digital



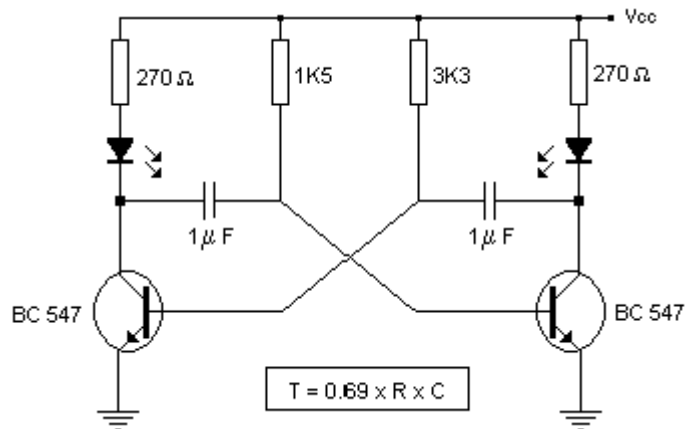
# Circuito Tri-State

Estágio de saída com **Tri-state**

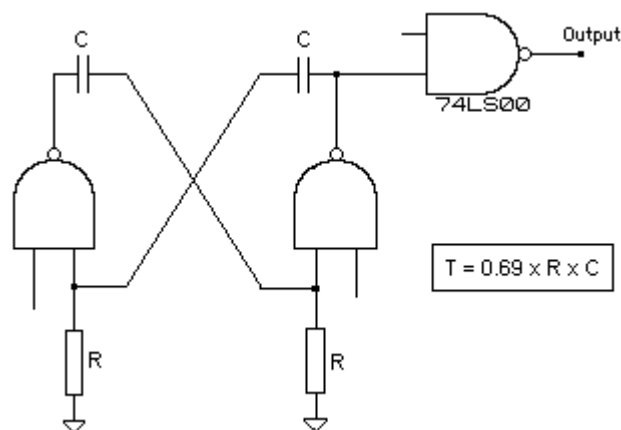


Quando o sinal D é colocado em zero, a corrente elétrica flui por D<sub>2</sub> colocando T<sub>3</sub> e T<sub>4</sub> em corte. Quando isto ocorre a impedância vista em Output é tão alta que somente não vai a infinito porquê existem fugas de corrente e efeito capacitivo.

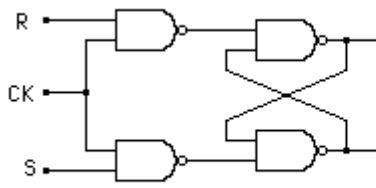
# Oscilador astável a transistor



# Oscilador astável a CI

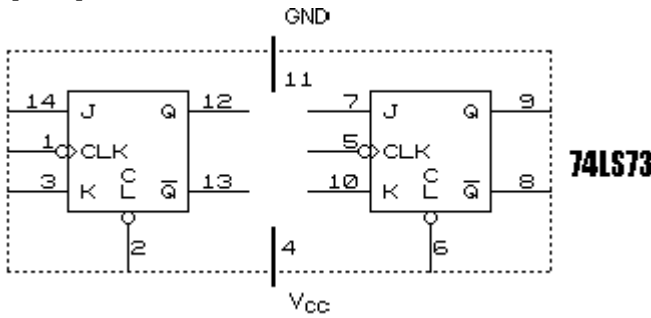


## Flip Flop tipo RS



74LS00				
CK	R	S	Q	$\bar{Q}$
1	0	0	Congela	
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

## Flip Flop tipo JK

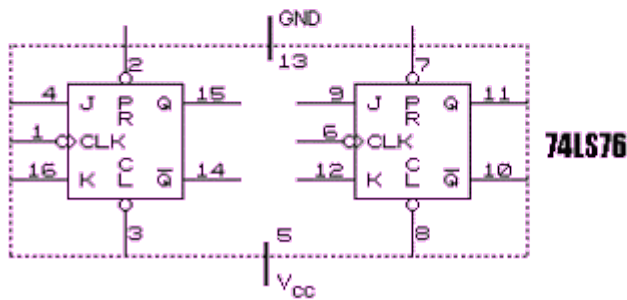


74LS73A					
CL	CK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	+	L	L	não varia	
H	+	H	L	L	H
H	+	L	H	H	L
H	+	H	H	inverte	
H	H	X	X	não varia	

74L73					
CL	CK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	+	L	L	não varia	
H	+	H	L	L	H
H	+	L	H	H	L
H	+	H	H	inverte	

HOLD  
RESET  
SET  
TOOGL

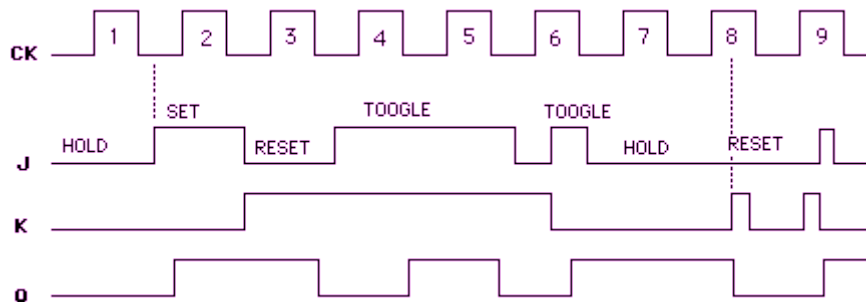
74LS73A					
CL	CK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	+	L	L	não varia	
H	+	H	L	L	H
H	+	L	H	H	L
H	+	H	H	inverte	
H	H	X	X	não varia	



74LS76A						
Pr	CL	CK	J	K	Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H <sup>+</sup>	H <sup>+</sup>
H	H	+	L	L	não varia	
H	H	+	H	L	L	L
H	H	+	L	H	L	H
H	H	+	H	H	inverte	
H	H	H	X	X	não varia	

+ instável

E uma aplicação de Flip Flops JK é a configuração Master Slave:

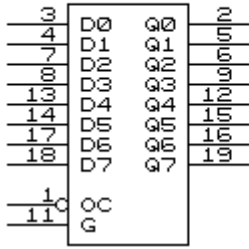






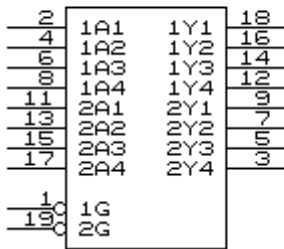
# Latch

*CI 74LS373*



Output Control	Enable		Output
	G	D	
0	1	1	1
0	1	0	0
0	0	x	Q <sub>0</sub>
1	x	x	Z

*CI 74LS244*

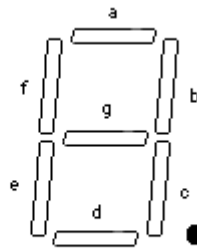
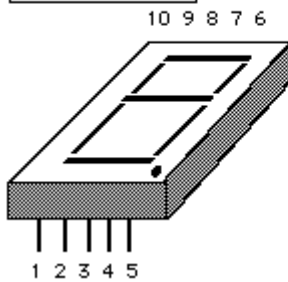


1G	1Ax	1Yx
0	0	0
0	1	1
1	x	Z

2G	2Ax	1Yx
0	0	0
0	1	1
1	x	Z

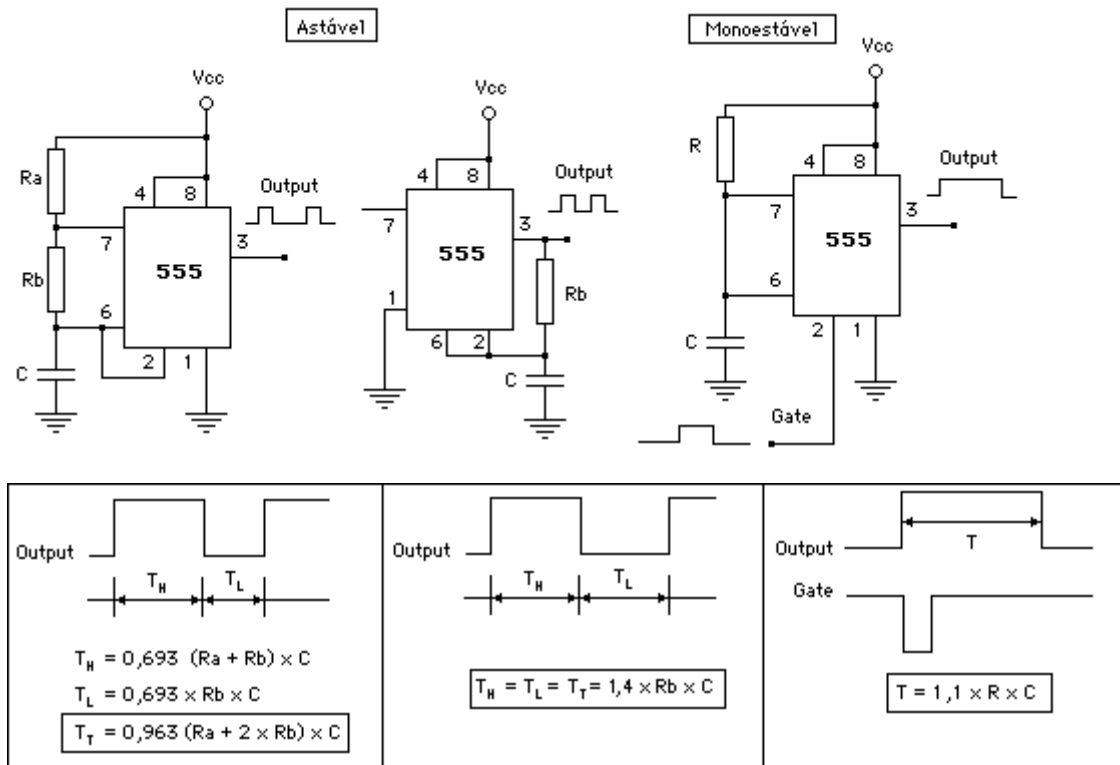
# Display 7 segmentos

*FND 507*

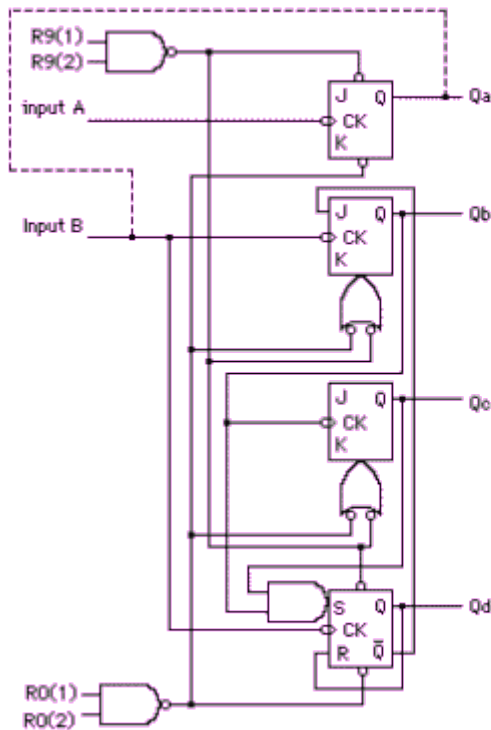


a=7
b=6
c=4
d=2
e=1
f=9
g=10
+ = 3,8
Pt = 5

# Osciladores com o CI555



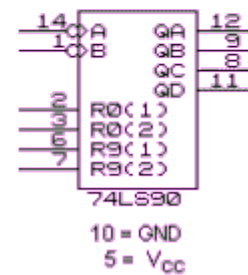
# Década contadora



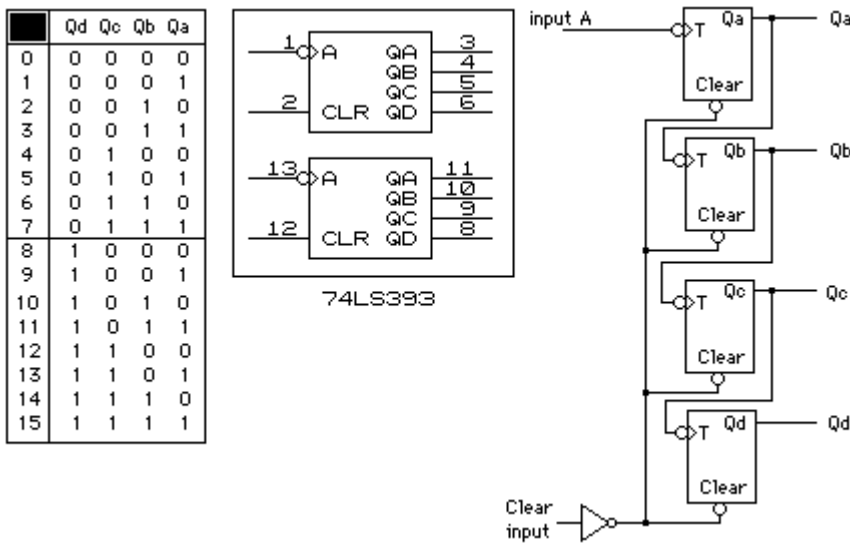
74LS90							
RO(1)	RO(2)	R9(1)	R9(2)	Qd	Qc	Qb	Qa
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			

Para década contadora use conectar o Qa em Input B.  
A contagem é de 0 a 9

	Qd	Qc	Qb	Qa
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



# Counter



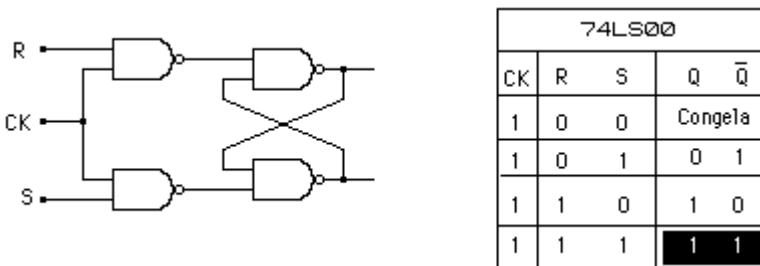
## Tecnologia dos contadores síncronos

### Introdução

Contadores síncronos são geradores de seqüências que, partindo de um valor pré determinado evoluem gerando conjuntos binários em ordem crescente conforme desejado. Os pinos de Clock são ligados em comum e á cada pulso nele os flip-flops JK empregados assumem características programadas previamente e que fazem com que o sinal gerado seja uma seqüência binária, tal como o código Gray, por exemplo.

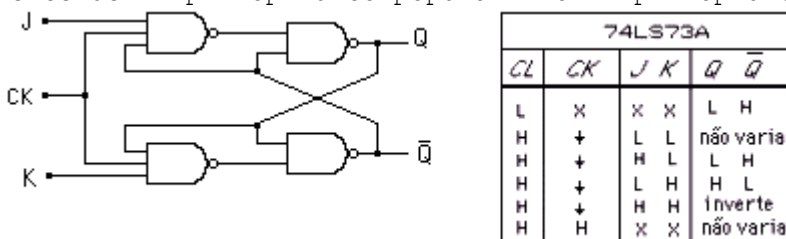
### Flip Flops

Em sua característica genérica trata-se apenas de um circuito capaz de criar situações que denominamos de Estado e que são diferentes do convencional, veja o exemplo a seguir:

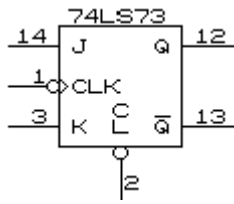


Trata-se de um flip flop do tipo RS. Veja que é apenas uma associação de portas lógicas do tipo NAND, porém, nesta associação elas se comportam como um circuito oscilador e as duas saídas são complementares e se alternam conforme a entrada.

Fazendo-se uma pequena modificação nesse esquema chegamos á um tipo diferente de Flip Flop muito popular - o Flip Flop JK:



Esquemáticamente ele é representado por:



Em termos de tabela de estado, tratamos os JK da seguinte forma:

Q <sub>ant</sub>	Q <sub>final</sub>	J	K
0	0	0	φ
0	1	1	φ
1	0	φ	1
1	1	φ	0

### Contadores

Com a apresentação dos JK propomos a discussão sobre uma associação bastante empregada em projetos que são os Contadores. Sua base é a associação de Flip Flops tipo JK, conforme apresentaremos.

O projeto é relativamente simples e consiste em conhecer os estados desejados e o comportamento dos JK.

Considere um exemplo onde desejamos um contador de 0 á 7 e que depois recomeça em 0 e assim por diante. Indefinidamente.

Para isto precisamos de três Flip Flops JK, pois precisamos de três dígitos binários:

Clock	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
1º	0	0	0	0	φ	0	φ	1	φ
2º	0	0	1	0	φ	1	φ	φ	1
3º	0	1	0	0	φ	φ	0	1	φ
4º	0	1	1	1	φ	φ	1	φ	1
5º	1	0	0	φ	0	0	φ	1	φ
6º	1	0	1	φ	0	1	φ	φ	1
7º	1	1	0	φ	0	φ	0	1	φ
8º	1	1	1	φ	1	φ	1	φ	1

Tabela 1 (Completa)

Considere possível a partida em 000 mas não acrescida neste projeto.

#### 1º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	0	0	⇒	0	φ
Q <sub>1</sub>	0	0	⇒	0	φ
Q <sub>0</sub>	0	1	⇒	1	φ

Consultando a tabela de estados, preencheremos J e K caso após caso, em vermelho.

#### 2º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	0	0	⇒	0	φ
Q <sub>1</sub>	0	1	⇒	1	φ
Q <sub>0</sub>	1	0	⇒	φ	1

### 3º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	0	0	⇒	0	φ
Q <sub>1</sub>	1	1	⇒	φ	0
Q <sub>0</sub>	0	1	⇒	1	φ

### 4º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	0	1	⇒	1	φ
Q <sub>1</sub>	1	0	⇒	φ	1
Q <sub>0</sub>	1	0	⇒	φ	1

### 5º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	1	1	⇒	φ	0
Q <sub>1</sub>	0	0	⇒	0	φ
Q <sub>0</sub>	0	1	⇒	1	φ

### 6º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	1	1	⇒	φ	0
Q <sub>1</sub>	0	1	⇒	1	φ
Q <sub>0</sub>	1	0	⇒	φ	1

### 7º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	1	1	⇒	φ	0
Q <sub>1</sub>	1	1	⇒	φ	0
Q <sub>0</sub>	0	1	⇒	1	φ

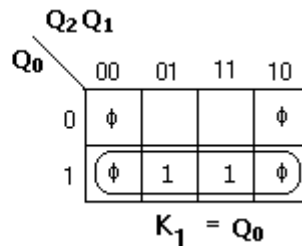
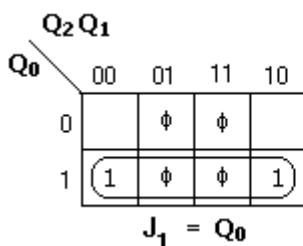
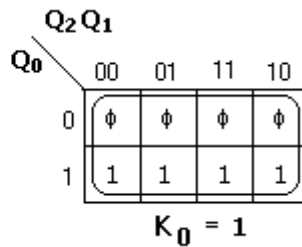
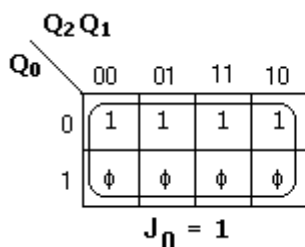
### 8º Clock

	Antes	Depois		J	K
Q <sub>2</sub>	1	0	⇒	φ	1
Q <sub>1</sub>	1	0	⇒	φ	1
Q <sub>0</sub>	1	0	⇒	φ	1

Note que após esta análise, pelo proposto a seqüência repete sendo portanto desnecessário prosseguir. Agora transferimos para a tabela 1 em azul, a condição de cada um dos J e cada um dos K, lembrando que cada um dos Flip Flop possui seu próprio JK.

### Análise de cada JK (Mapa de Karnaught)

Agora é necessário analisar cada caso no mapa de Karnaught:



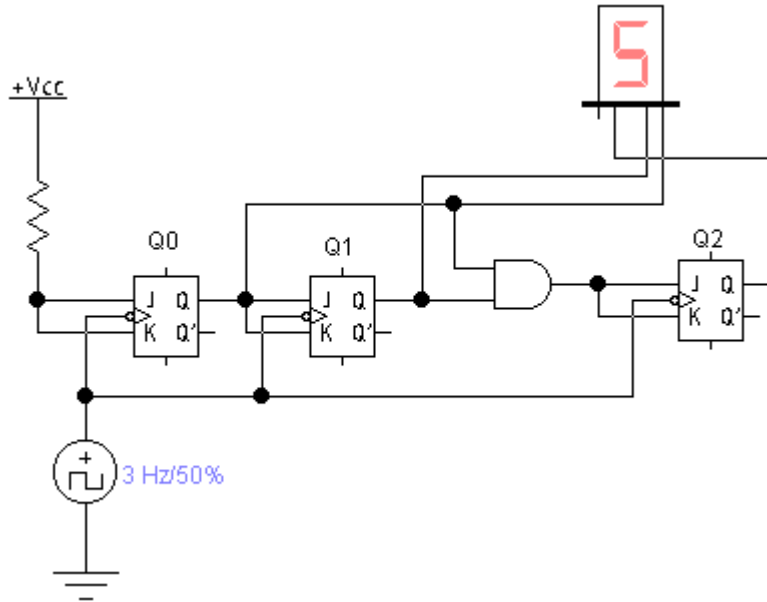
		Q <sub>2</sub> Q <sub>1</sub>			
Q <sub>0</sub>		00	01	11	10
Q <sub>0</sub>	0			φ	φ
	1		1	φ	φ

$J_2 = Q_1 Q_0$

		Q <sub>2</sub> Q <sub>1</sub>			
Q <sub>0</sub>		00	01	11	10
Q <sub>0</sub>	0	φ	φ		
	1	φ	φ	1	

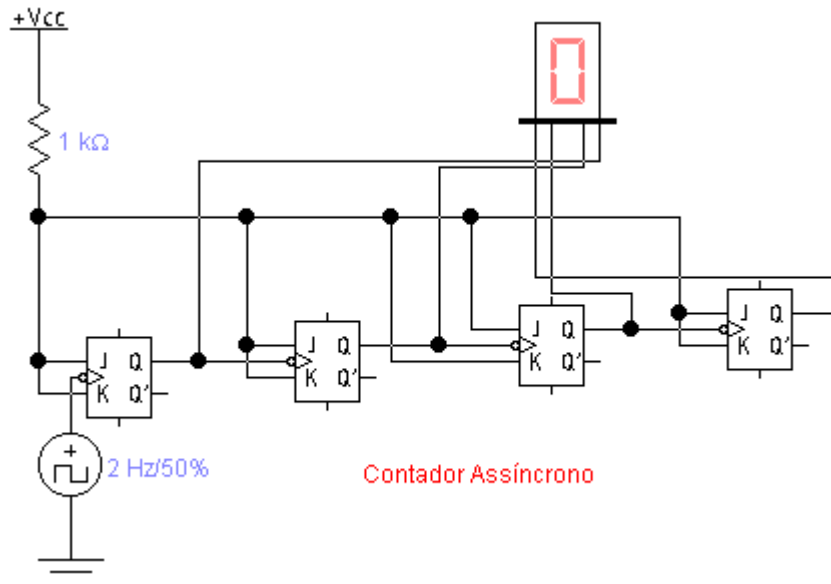
$K_2 = Q_1 Q_0$

E o circuito resultante é:



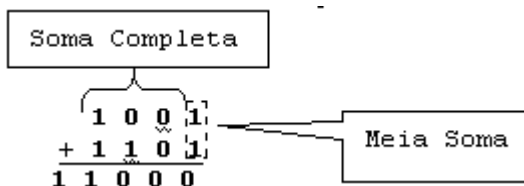
## Tecnologia dos contadores Assíncronos

Contadores Assíncronos tem como característica o fato do terminal **Q** ser o sinal de clock do JK seguinte tal como uma espécie de cascata, assim, isto faz com que as saídas dos JK tenham um sinal de clock que não ocorrem simultaneamente.



## Circuitos Aritméticos

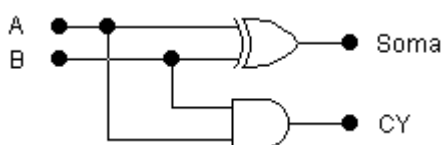
Estes circuitos são muito empregados em **ULAs** de CPUs. Fazem a parte básica da aritmética binária. Existem basicamente dois elementos: Os **somadores** e os **subtratores**. Dentro de ambos os elementos podemos destacar as **Meia Soma** e **Soma Completa** e o **Meio Subtrator** e o **Subtrator Completo**. Faremos uma breve exposição sobre os somadores pela sua maior simplicidade. Considere o exemplo:



A meia soma trata da soma dos bit menos significativos, e os demais são a soma completa. Observe a tabela seguinte:

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

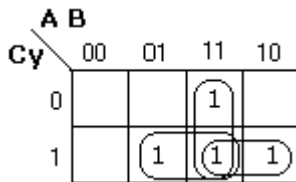
O que se pode observar é que a saída **S** que é a soma, corresponde á uma porta **XOR** e o **CY** pode ser claramente considerado como um **AND**. Desta forma podemos obter o circuito da **meia soma**:



Pelo mesmo princípio, por analogia podemos obter:

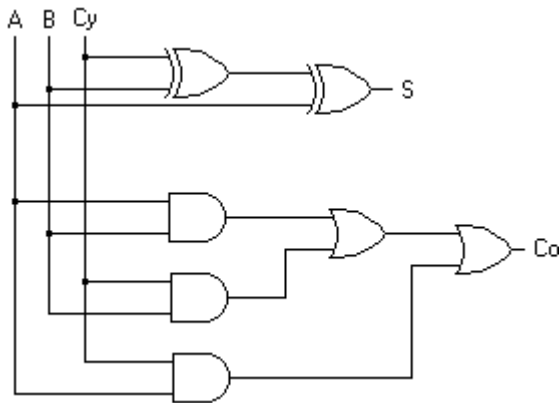
A	B	Cy	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Que aplicada ao mapa de Karnaught, resulta em:



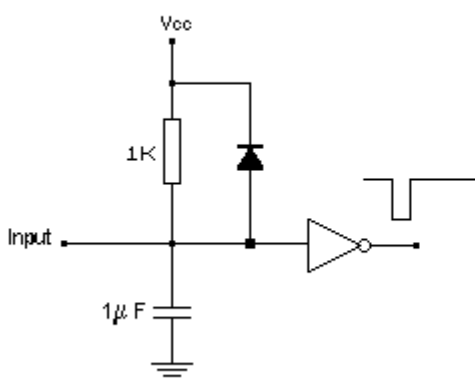
de onde:  $Co = AB + BCy + ACy$

Que resulta no circuito da soma completa:

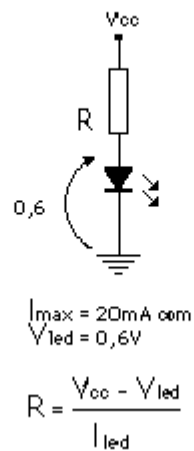


## Generalidades

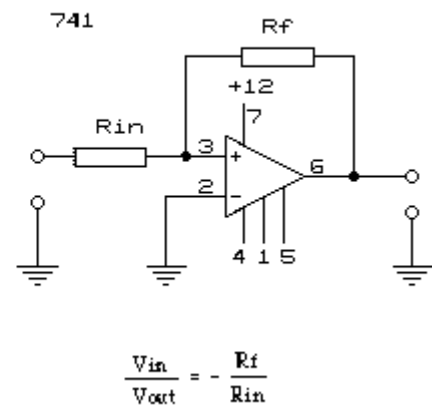
### Circuito de Power-On-Reset



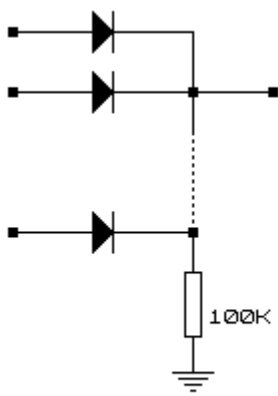
### Dimensionando LED



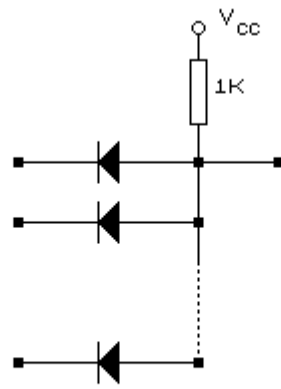
### Amplificador Inversor



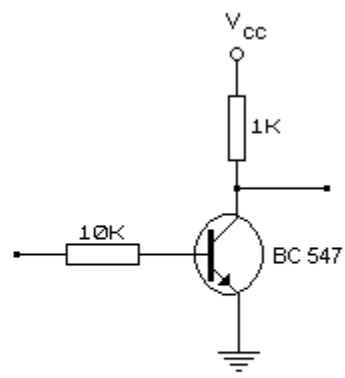




OR



AND



INVERSOR