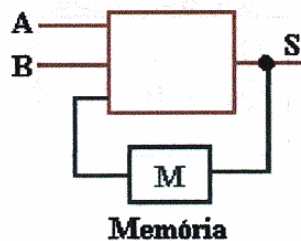


## 6. CIRCUITOS BIESTÁVEIS - LATCHES E FLIP-FLOPS

### 6.1 INTRODUÇÃO

Os circuitos sequenciais são circuitos cujas saídas num determinado instante dependem tanto dos valores das entradas nesse instante quanto dos valores anteriores das saídas.

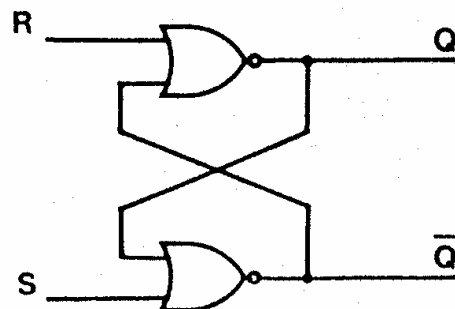
Os circuitos sequenciais mais importantes são os **biestáveis**. São circuitos constituídos por portas lógicas e capazes de armazenar um bit de informação. São por isso vistos como elementos de memória.



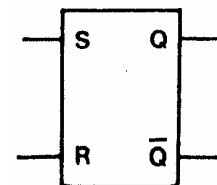
**Biestáveis** { **Síncronos** (Flip-Flops): respondem em determinados instantes, quando se dá a variação de um sinal de relógio ou *clock*  
**Assíncronos** (Latches): as saídas mudam em qualquer instante de tempo, Existindo ou não uma variável de controlo.

### 6.2 LATCH S-R (Set-Reset)

Esquema do circuito :  
(implementação com portas NOR)



Símbolo:



Entradas: S – SET e R – RESET

Saídas: Q e  $\bar{Q}$

Estados de um latch: **SET** ( $Q = 1$  e  $\bar{Q} = 0$ ) e **RESET** ( $Q = 0$  e  $\bar{Q} = 1$ )

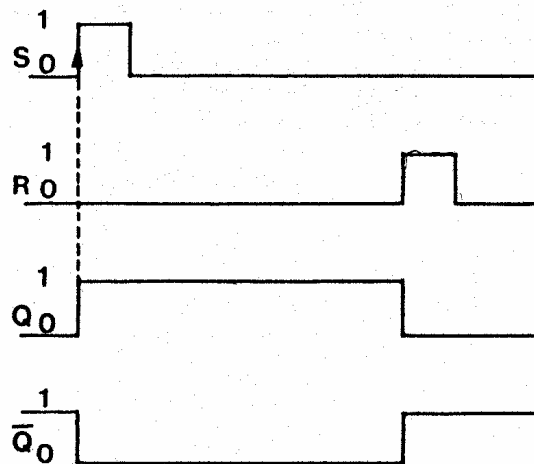
**Princípio de funcionamento (tabela de verdade):**

Estado inicial: n		Entradas		Saídas (Estado final : n+1)			
R	$Q_n$	$\overline{Q}_n$	S	R	$Q_{n+1}$	$\overline{Q}_{n+1}$	Estado
R E S E T	0	1	0	0	0	1	RESET
	0	1	0	1	0	1	RESET
	0	1	1	0	1	0	SET
	0	1	1	1	0	0	INDET.
S E T	1	0	0	0	1	0	SET
	1	0	0	1	0	1	RESET
	1	0	1	0	1	0	SET
	1	0	1	1	0	0	INDET.

**Tabela abreviada:**

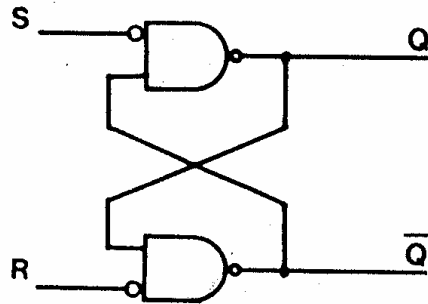
S	R	$Q_{n+1}$	$\overline{Q}_{n+1}$
0	0	$Q_n$	$\overline{Q}_n$
0	1	0	1
1	0	1	0
1	1	0	0

**Diagrama de sinais:**

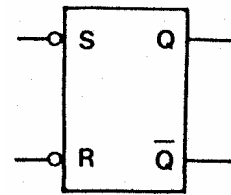


A mudança de estado faz-se no flanco de subida (positivo) das variações das entradas (“leading edge triggered”)

### Implementação do circuito com portas NAND:



### Símbolo:

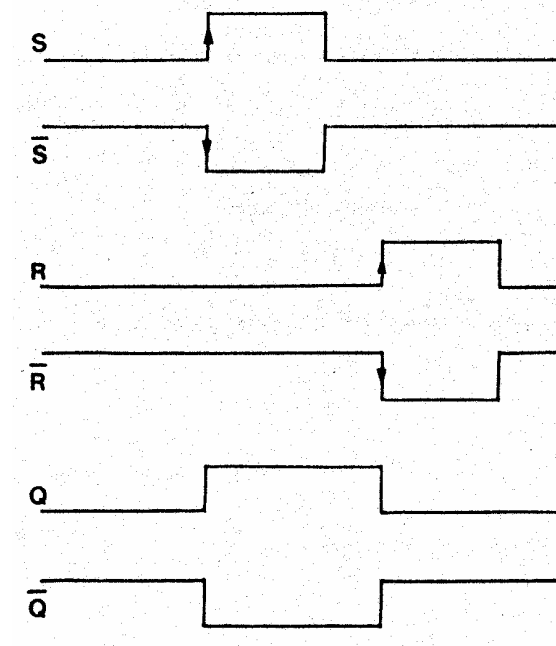


### Tabela de verdade abreviada:

S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$
0	0	$Q_n$	$\bar{Q}_n$
0	1	0	1
1	0	1	0
1	1	1	1

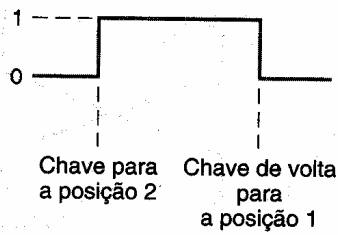
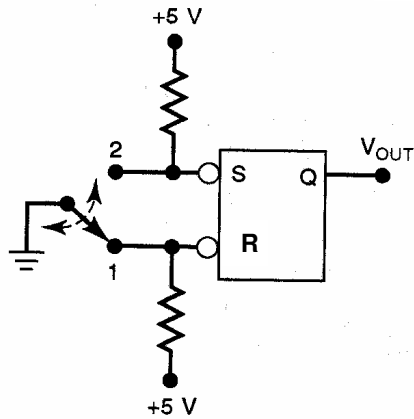
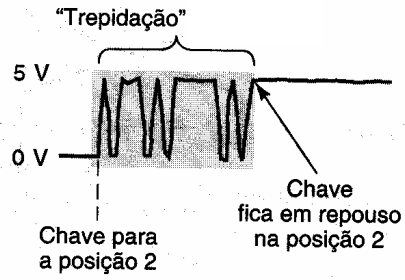
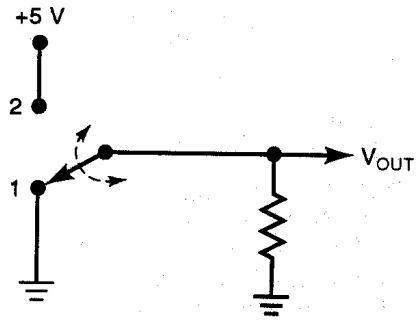
### Diagrama de sinais:

A mudança de estado faz-se no flanco negativo (“trailing edge triggered”) das variações de  $\bar{R}$  e  $\bar{S}$ .



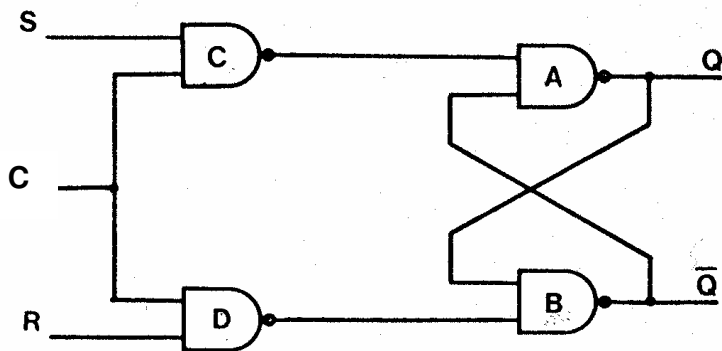
Os circuitos integrados TTL ref. 74LS279 e CMOS ref. 4043 possuem 4 latches S-R

**Aplicação do LATCH S-R:** Eliminação da variação de tensão por trepidação de contacto (“Contact Bounce Eliminator”)



**6.3 LATCH S-R COM CONTROLO**

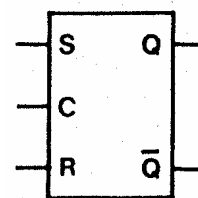
**Esquema do circuito:**



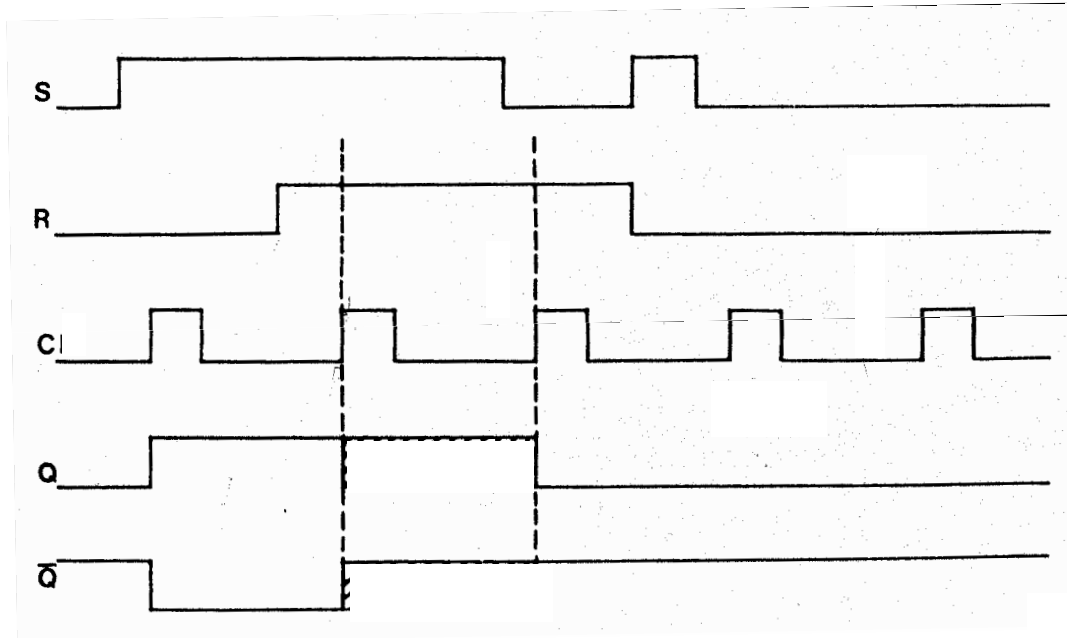
**Tabela de verdade abreviada:**

C	S	R	$Q_{n+1}$	$\bar{Q}_{n+1}$
0	X	X	$Q_n$	$\bar{Q}_n$
1	0	0	$Q_n$	$\bar{Q}_n$
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

**Símbolo:**



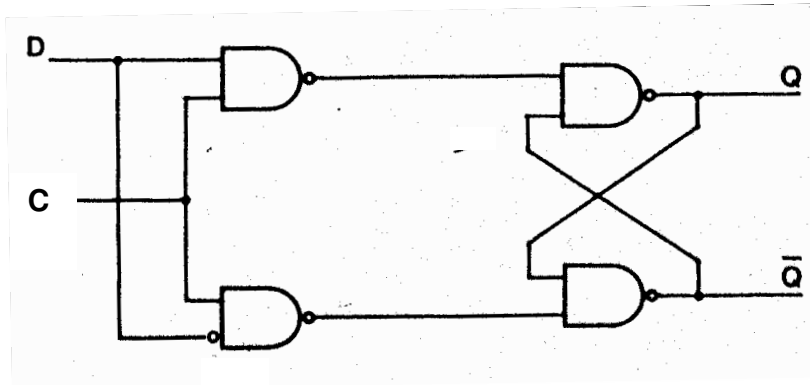
## Diagrama temporal:



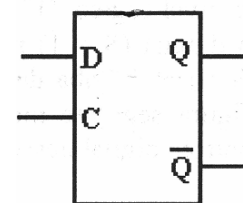
## 6.4 LATCH D

Este *latch* tem a vantagem, em relação aos anteriores, de eliminar o estado indeterminado.

### Esquema do circuito:



### Símbolo:



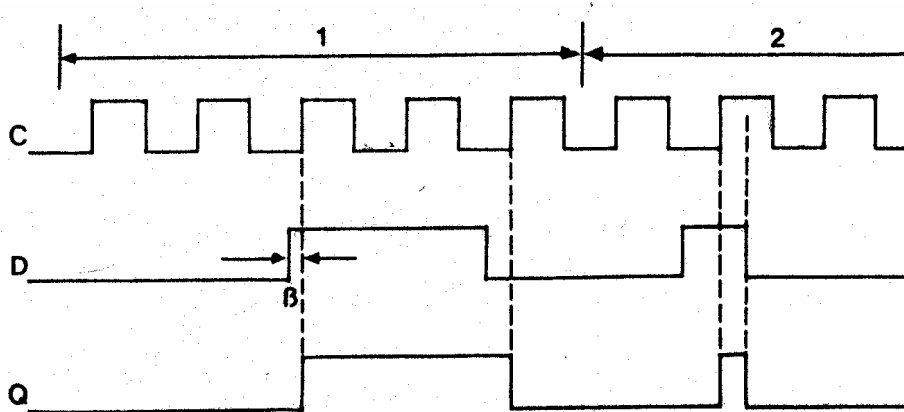
Note-se que este latch se obtém do latch S-R fazendo  $S = D$  e  $R = \bar{D}$

Os circuitos integrados TTL ref. 74LS75 e CMOS ref. 4042 possuem 4 latches tipo D.

### Tabela de verdade:

D	C	$Q_{n+1}$	$\overline{Q}_{n+1}$
0	0	$Q_n$	$\overline{Q}_n$
0	1	0	1
1	0	$Q_n$	$\overline{Q}_n$
1	1	1	0

### Diagrama temporal:



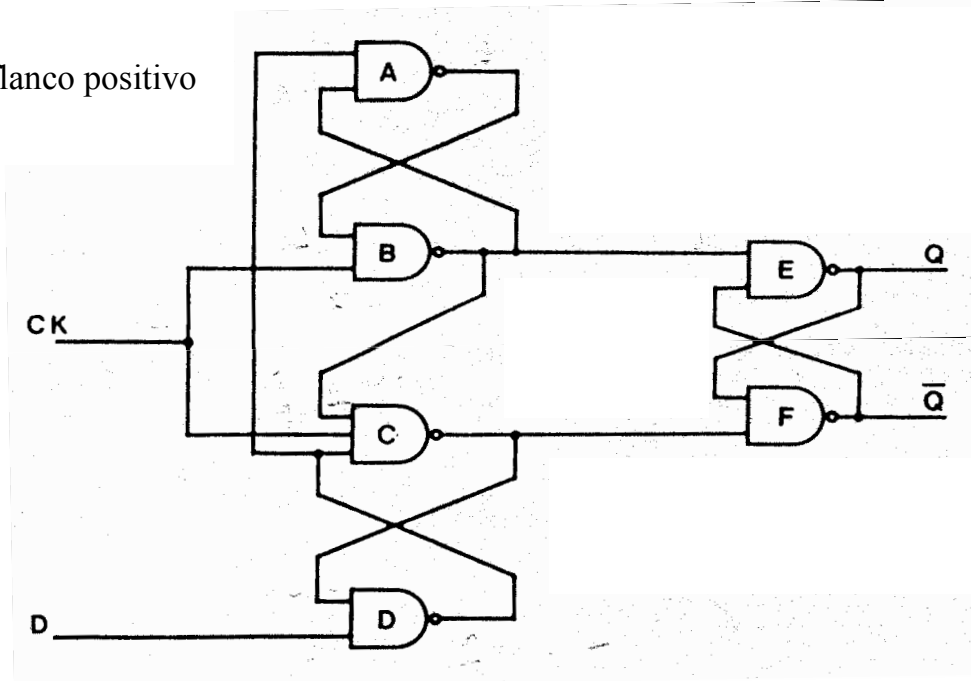
Na zona 2 do gráfico está representada uma situação inconveniente para determinadas aplicações, isto é, durante todo o patamar do impulso de clock, a saída pode variar desde que se varie a entrada.

Esta situação é eliminada com o FLIP-FLOP D EDGE TRIGGERED, isto é, FF D disparando unicamente no flanco (positivo ou negativo) do impulso de *clock*.

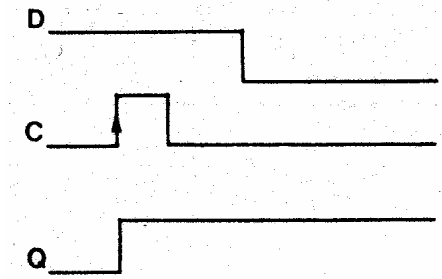
## 6.5 FLIP-FLOP TIPO D EDGE TRIGGERED

### Circuito:

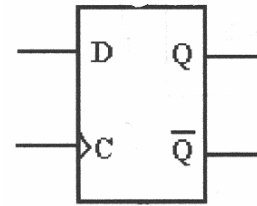
Disparo no flanco positivo



### Disparo no flanco positivo:



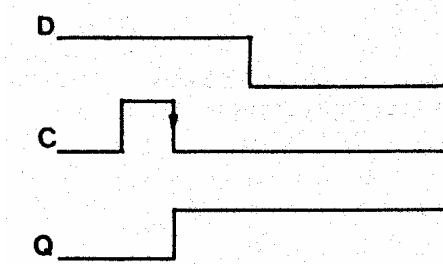
### Símbolo:



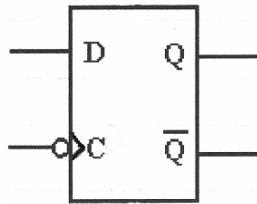
### Tabela de verdade (disparo no flanco positivo):

D	C	$Q_{n+1}$	$\overline{Q}_{n+1}$
X	X	$Q_n$	$\overline{Q}_n$
0	↑	0	1
1	↑	1	0

### Disparo no flanco negativo:



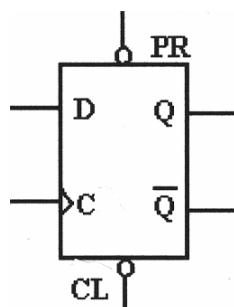
### Símbolo:



## 6.6 FLIP-FLOP D PRESET-CLEAR

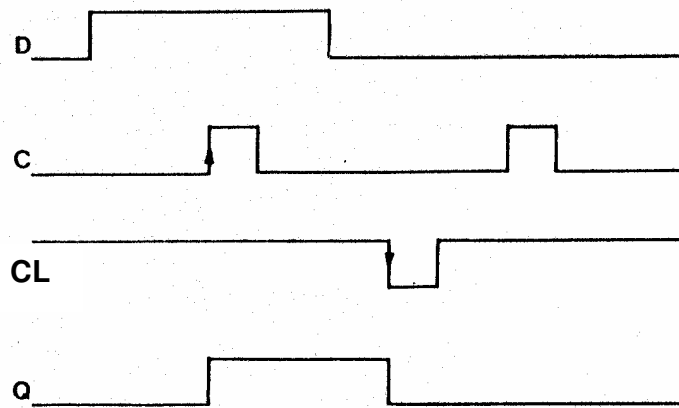
Este flip-flop é idêntico ao anterior, excepto que tem duas entradas assíncronas activadas com nível baixo chamadas Preset (PR) e Clear (CL). Normalmente essas entradas estão no nível lógico 1, para evitar influenciarem o funcionamento normal do FF. Quando uma delas passa momentaneamente ao nível 0 leva o flip-flop, respectivamente, ao estado SET ou RESET. Se o nível se mantiver no nível zero o FF fica bloqueado nesse estado.

### Símbolo:



O circuito integrado 74LS74 possui 2 flip-flops tipo D actuando no flanco de subida

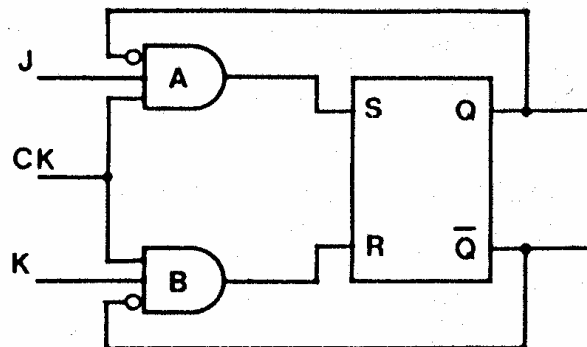
### Diagrama temporal:



### 6.7 LATCH J-K

#### Circuito

Este circuito é constituído por um *latch* R-S e duas portas AND, havendo realimentação das saídas Q e  $\bar{Q}$ , como se apresenta:



#### Tabela de verdade:

C	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$
X	X	X	$Q_n$	$\bar{Q}_n$
1	0	0	$Q_n$	$\bar{Q}_n$
1	0	1	0	1
1	1	0	1	0
1	1	1	$\bar{Q}_n$	$Q_n$

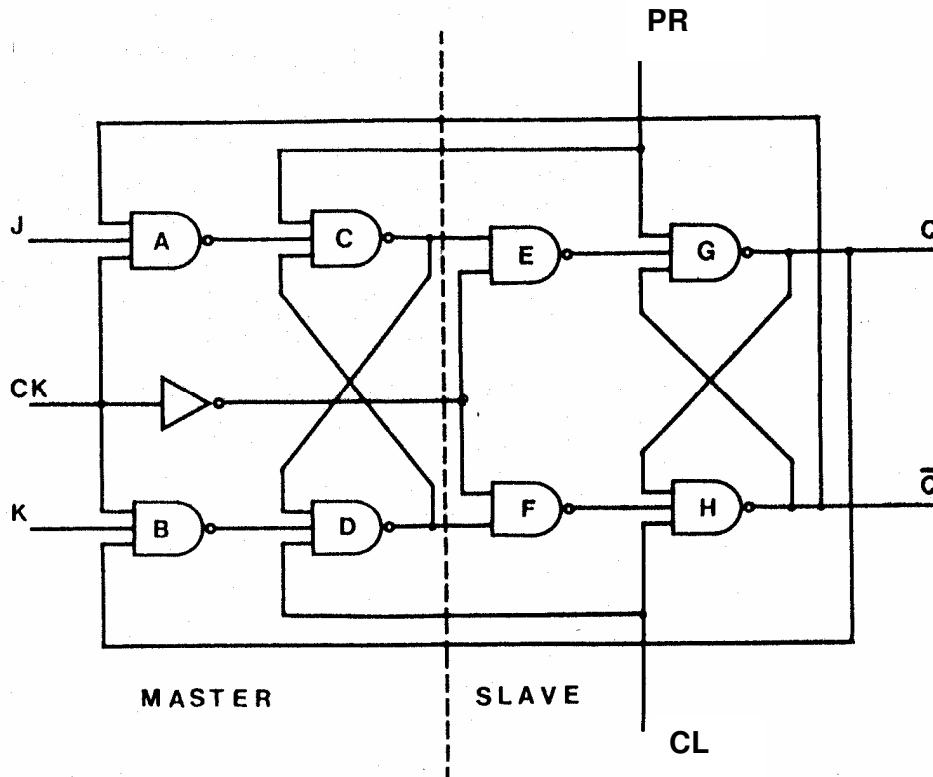
A tabela de verdade mostra que este circuito funciona como um *latch* S-R, sendo o estado indeterminado eliminado. No entanto, se o *clock* for “1” o *latch* J-K ficará oscilando.



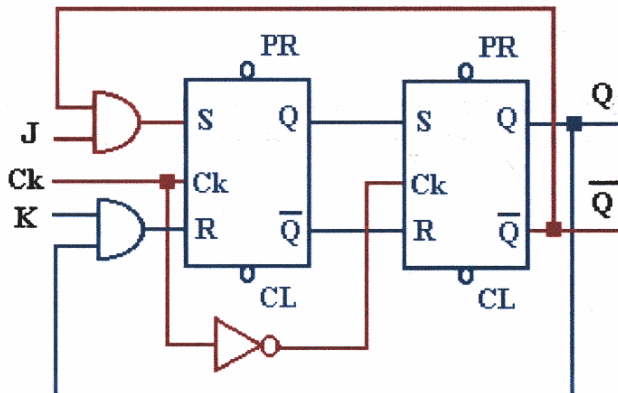
## 6.8 FLIP-FLOP J-K MESTRE ESCRAVO (MS)

Este FF é constituído por dois latches S-R com controlo, no primeiro dos quais o clock actua no flanco de subida dos impulsos (no “leading edge”) e no segundo, no flanco de descida (no “trailing edge”). Portanto a saída do primeiro FF (Mestre - **Master**) vai ser transmitida ao segundo FF (Escravo - **Slave**) no flanco de descida do impulso de clock, o que implica dizer haver um atraso igual à duração do clock. Deste modo a oscilação indesejada do latch J-K quando  $J = 1$  e  $K = 1$  é eliminada.

### Circuito:



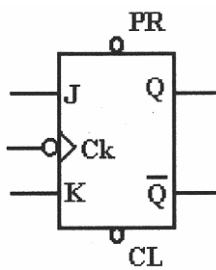
### Esquema:



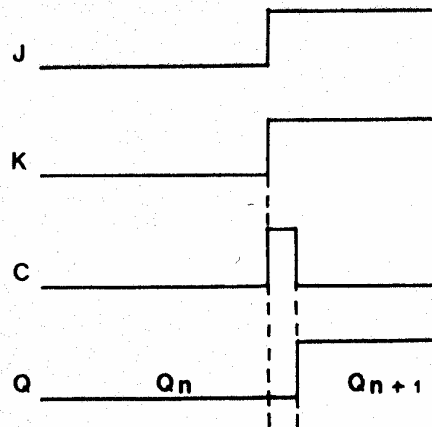
### Tabela de verdade:

C	J	K	$Q_{n+1}$	$\overline{Q}_{n+1}$
X	X	X	$Q_n$	$\overline{Q}_n$
↓	0	0	$Q_n$	$\overline{Q}_n$
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	$\overline{Q}_n$	$Q_n$

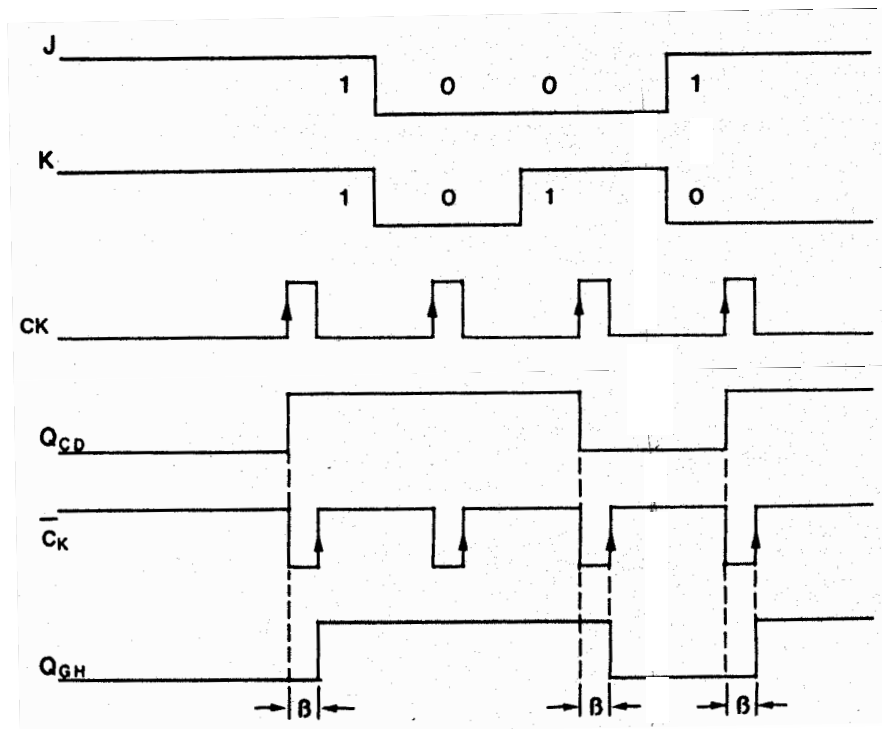
**Símbolo:**



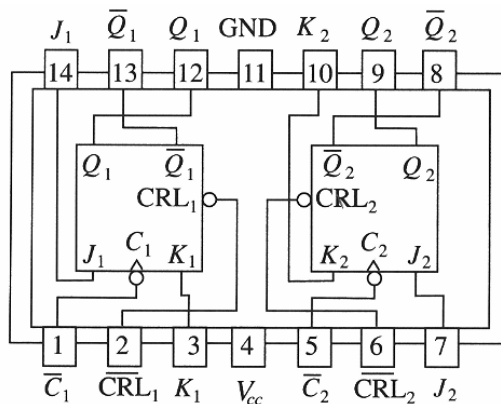
**Mudança de estado:**



**Exemplo de diagrama de sinais do FF J-K MESTRE-ESCRAVO:**



**CI 7473**



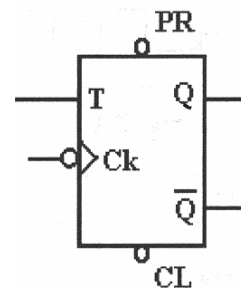
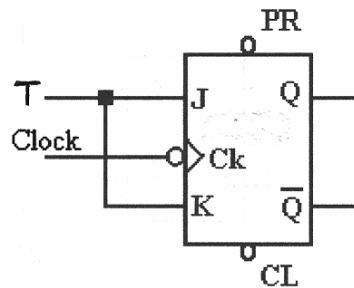
Entradas				Saídas	
CLR	C	J	K	Q	Q̄
L	X	X	X	L	H
H		L	L	Q <sub>0</sub>	Q̄ <sub>0</sub>
H		H	L	H	L
H		L	H	L	H
H		H	H	Toggle	

## 6.9 FLIP-FLOP T OU TOGGLE

Este é um FF que muda de estado no flanco de subida ou de descida do impulso de clock e quando a entrada (T) está a um nível alto. É construído com um flip-flop J-K MS em que as entradas J e Q são curto-circuitadas. Também pode ser construído com um flip-flop tipo D e uma porta XOR como se mostra a seguir.

### Flip-Flop T a partir de um flip-flop J-K MS

### Símbolo:

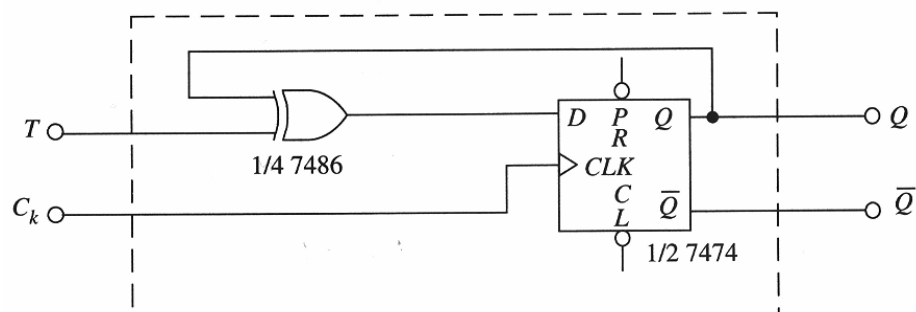


### Tabela de verdade:

C	T	$Q_{n+1}$	$\bar{Q}_{n+1}$
X	X	$Q_n$	$\bar{Q}_n$
↓	0	$Q_n$	$\bar{Q}_n$
↓	1	$\bar{Q}_n$	$Q_n$

### Flip-Flop T a partir de um flip-flop tipo D

(mudança de estado no flanco de subida do sinal de clock)



## 6.10 CONTADORES

**Contadores** - São circuitos sequenciais que fazem uma contagem do número de impulsos aplicados à sua entrada. Esta contagem é representada nas suas saídas, num determinado código.

Além disso, são utilizados em diversa instrumentação digital, por exemplo em sistemas de medição digital de quantidades como tempo, velocidade, frequência e distância, entre outras.

São constituídos por flip-flops ligados entre si.

### Tipos de contadores

- Ascendentes ou progressivos:** fazem uma contagem de forma crescente
- Descendentes ou regressivos:** fazem uma contagem de forma decrescente
- Reversíveis:** Podem contar na forma crescente ou decrescente – UP/DOWN

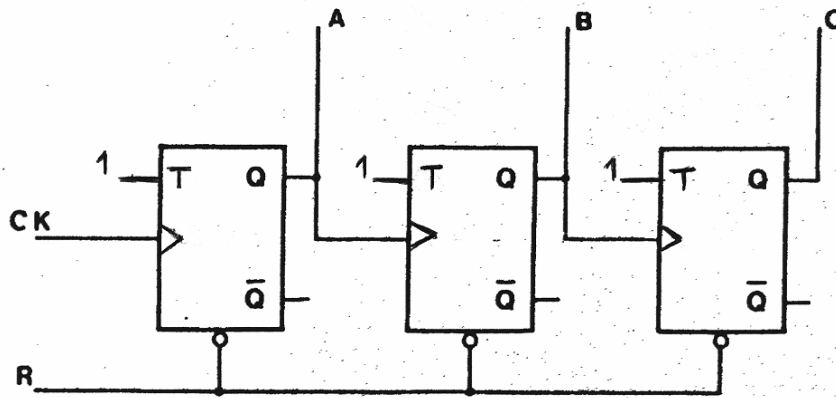
- Síncronos :** o sinal de *clock* é aplicado simultaneamente a todos os flip-flops
- Assíncronos:** o sinal de clock é aplicado apenas ao primeiro flip-flop, a saída deste ao sinal de clock do seguinte e assim sucessivamente.

- Binários:** o número de estados que se podem representar nas saídas é múltiplo de dois.
- Decimais:** o número de estados que se podem representar nas saídas é múltiplo de dez.
- Módulo N:** este termo refere que o número de estados que se podem representar nas saídas é N; utiliza-se geralmente para contadores que não são binários nem decimais; conta de 0 a N-1.

### 6.10.1 Contadores assíncronos binários regressivos

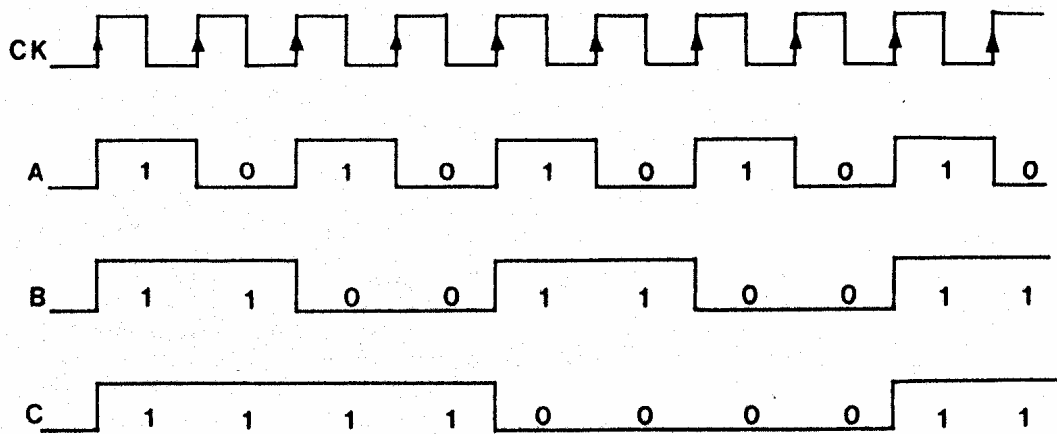
#### Exemplo

Consideremos o seguinte circuito, implementado com flip-flops tipo T que actua no flanco ascendente do sinal de clock:



A saída do primeiro FF liga ao sinal de clock do segundo e assim sucessivamente. Os vários flip-flops não comutam em sincronismo com o clock mas sim com um atraso de um FF para outro. Por este motivo se diz assíncrono, isto é, não síncrono.

O diagrama de sinais deste circuito é o seguinte

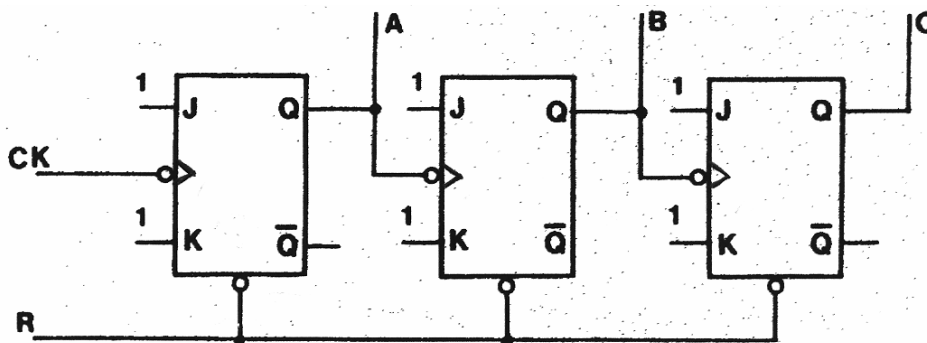


Verificamos que o circuito conta, em binário, os dígitos decimais de 7 a 0, em que o primeiro FF (A) corresponde ao bit menos significativo e o último FF (C) ao bit mais significativo. A contagem é feita no sentido decrescente e por isso o contador diz-se *regressivo*.

### 6.10.2 Contadores assíncronos binários progressivos

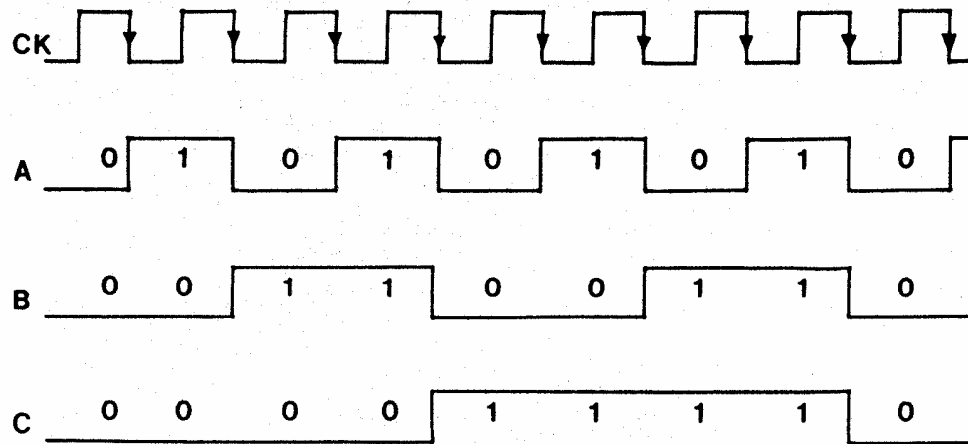
#### Exemplo

Consideremos agora o seguinte circuito, implementado com FF's J-K:



Todas as entradas J e K dos vários flip-flop's estão ligadas ao um lógico (1), o que equivale aos FF's J-K funcionarem como T's.

O seu diagrama de sinais é o seguinte:



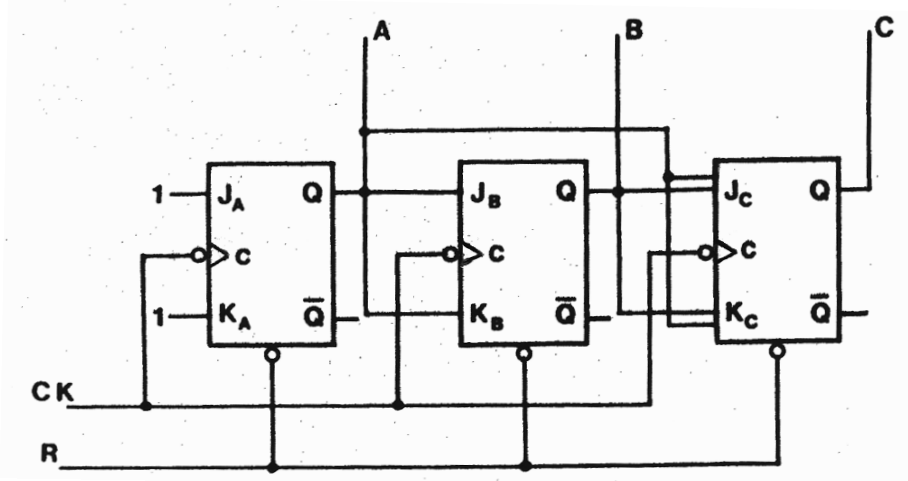
Verificamos que, neste caso, também é feita a contagem dos dígitos decimais de 0 a 7. A contagem é feita no sentido crescente, o contador diz-se *progressivo*.

Cada FF representa um bit da contagem. O limite do número decimal contado por um número  $n$  de FF's é  $2^n - 1$  (0 a  $2^n - 1$ ). O número de estados que se podem representar nas saídas do contador é  $2^n$ .

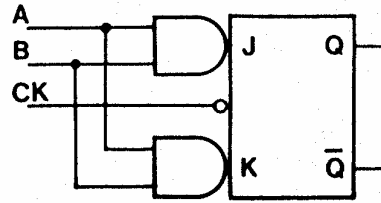
### 6.10.3 Contador síncrono binário progressivo

#### Exemplo

Consideremos o seguinte circuito, em que o impulso de Clock é aplicado simultaneamente às entradas C dos FF's. Por isso este contador se chama síncrono, os seus FF's actuam em sincronismo.



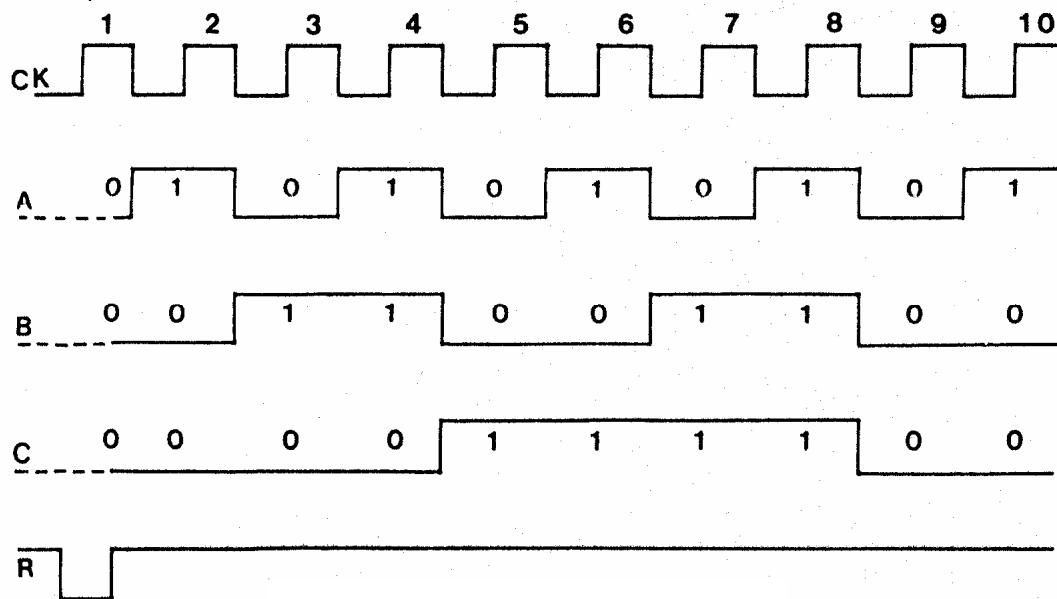
O último FF tem nas suas entradas portas AND, sendo equivalente a:



As expressões das várias entradas são:

$$\begin{array}{lll}
 J_A = 1 & J_B = A & J_C = A.B \\
 K_A = 1 & K_B = A & K_C = A.B
 \end{array}$$

O diagrama de sinais do contador é o seguinte:



Como vemos este circuito conta, em **binário**, os dígitos decimais de 0 a 7 no sentido crescente, sendo por isso um *contador progressivo*. O FF mais à direita(A) dá o dígito menos significativo e o FF mais à esquerda (C) dá o dígito mais significativo.

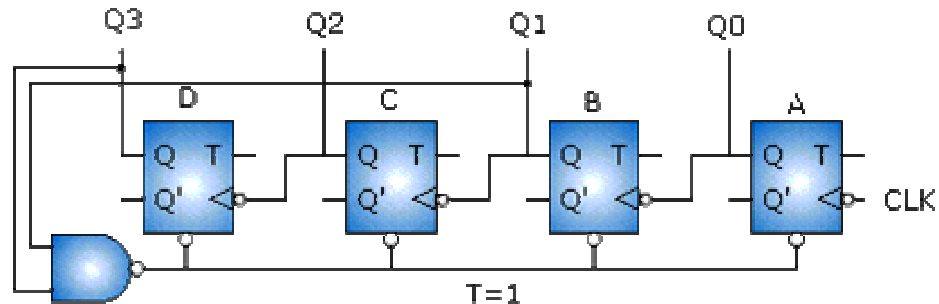




### 6.10.5 Contador decimal ou Módulo 10 (MOD10)

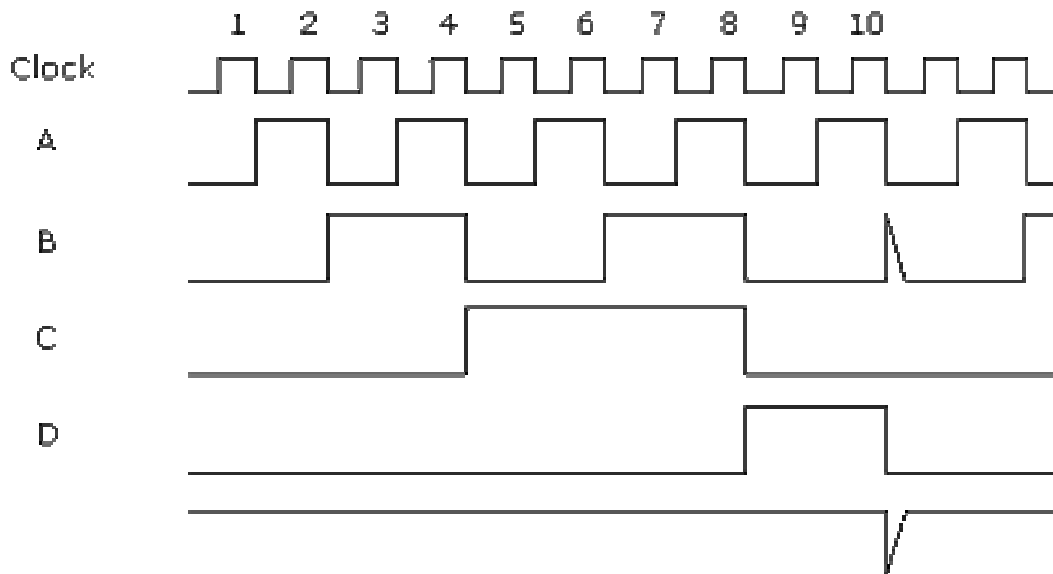
#### Exemplo

Consideremos o seguinte circuito:



Se a porta NAND não estivesse presente, este seria um contador binário com  $2^4 = 16$  estados, ou seja, contaria de 0 a 15. A porta NAND faz com que, quando as saídas  $Q_3Q_2Q_1Q_0$  vão a passar de 1001 (9) para 1010 (10), a entrada  $CLR$  dos flip-flops vá a zero e então todas as saídas ficam também em zero e a contagem recomeça. A contagem faz-se assim de 0 a 9.

O diagrama de sinais é o seguinte:

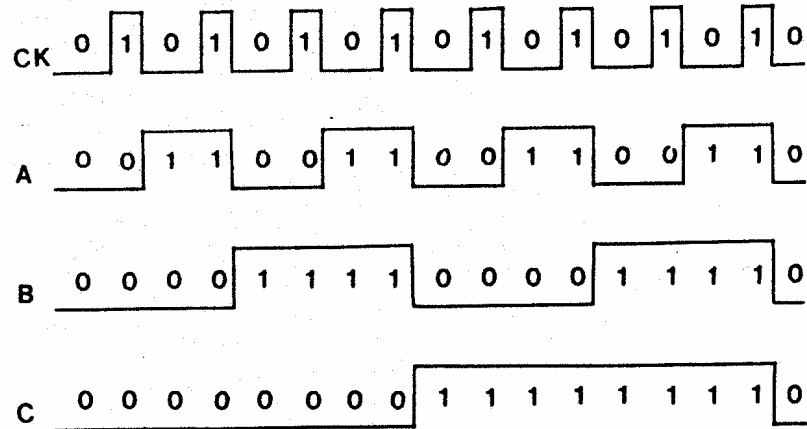


De um modo geral, obtém-se um contador Módulo N ( contagem de 0 a N-1) ligando as saídas de nível alto (1) do contador, correspondente ao decimal N codificado em binário, às entradas  $CLR$  dos flip-flops.

Por exemplo um contador MOD13 obtém-se ligando as saídas  $Q_3$ ,  $Q_2$  e  $Q_1$  às entradas da porta NAND ( $1110_2 = 13_{10}$ )

### 6.10.6 Divisores de frequência

Como podemos verificar pelos diagramas de sinais dos diversos contadores binários estudados, as saídas dos diversos FF's vão-nos dando impulsos com uma frequência submúltipla da frequência do sinal de clock, que vai sendo dividida por dois quando passamos de uma saída para a seguinte.



#### Contadores comerciais

CI 4029 : binário/decimal up/down  
CI 4020 : binário 14 bits  
CI 4040 : binário 12 bits  
CI 7493 : binário assíncrono 4 bits

CI 74160 : decimal 4 bits (MOD10)  
CI 74161-63 : binário 4 bits síncrono  
CI 74190 : decimal 4 bits (MOD10)  
CI 74191 : binário 4 bits reversível síncrono

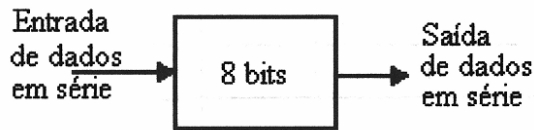
## 6.11 REGISTOS DE DESLOCAMENTO

- São circuitos sequenciais que armazenam informações binárias.
- São construídos por flip-flops ligados em cadeia contendo tantos flip-flops quantos os bits a armazenar.
- Realizam conversão de dados em série para paralelo e vice-versa.
- Realizam operações aritméticas básicas.

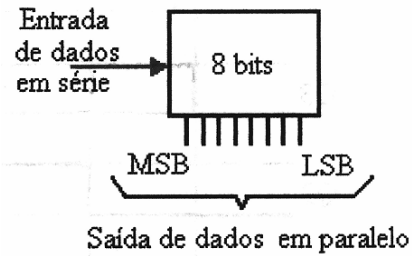
### 6.11.1 Tipos de registos

Os registos podem ser classificados de acordo com o modo como é feita a recepção e a transmissão de dados (série ou paralelo). Temos assim quatro combinações possíveis entre a entrada e a saída dos dados:

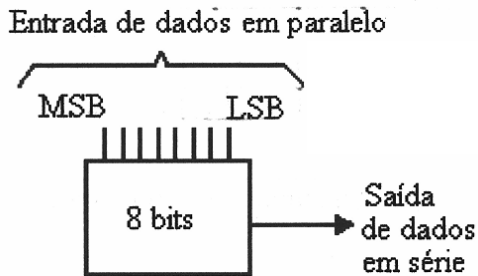
### Série- Série



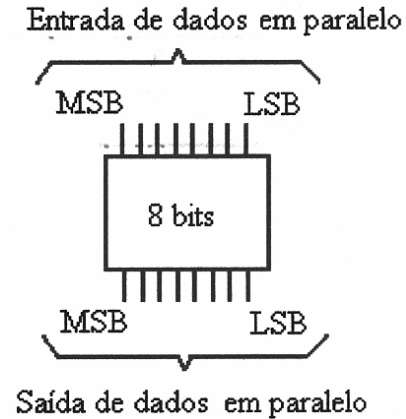
### Série-Paralelo



### Paralelo - Série



### Paralelo - Paralelo

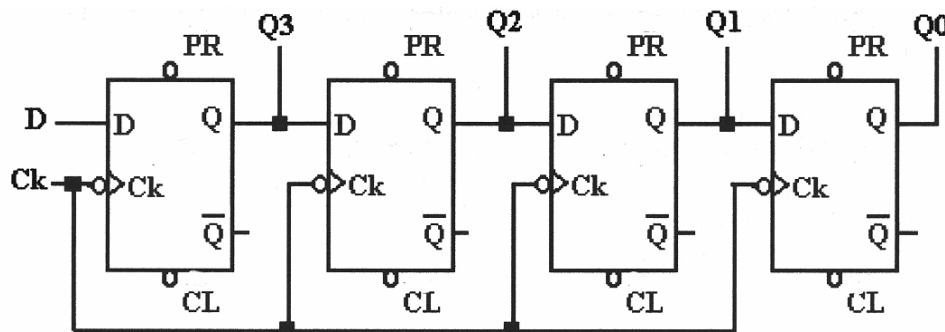


- Circuitos comerciais:** CI 7491 (série - série , 8 bits)  
CI 74164 (série – paralelo, 8 bits)  
CI 74165 (paralelo – série, 8 bits)  
CI 74198 (paralelo – paralelo, 8 bits)  
CI 74194 (todas as formas, 4 bits)

## 6.11.2 Registos de deslocamento com entrada de dados em série

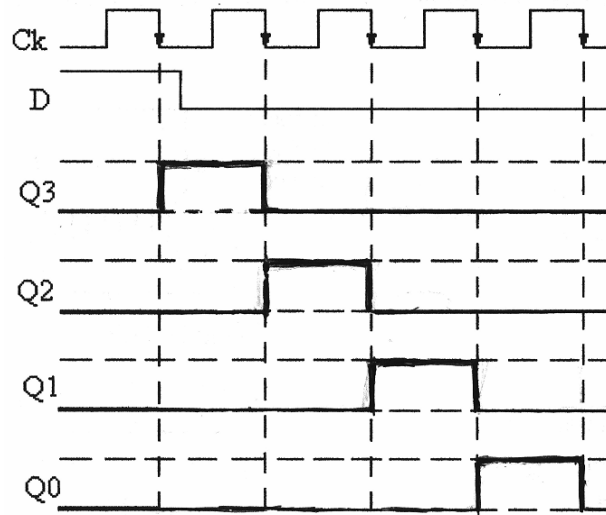
### Deslocamento para a direita

A figura seguinte mostra como construir um registo de deslocamento para a direita utilizando flip-flops tipo D.



Se antes e durante um impulso de clock for aplicado um nível um (1) na entrada D do primeiro FF do registrador, esse nível vai sendo **deslocado para a direita** ao longo deste a cada impulso de clock. Ou seja, o estado de saída de um FF é transmitido ao seguinte a cada impulso de clock.

O diagrama temporal seguinte ilustra este funcionamento.

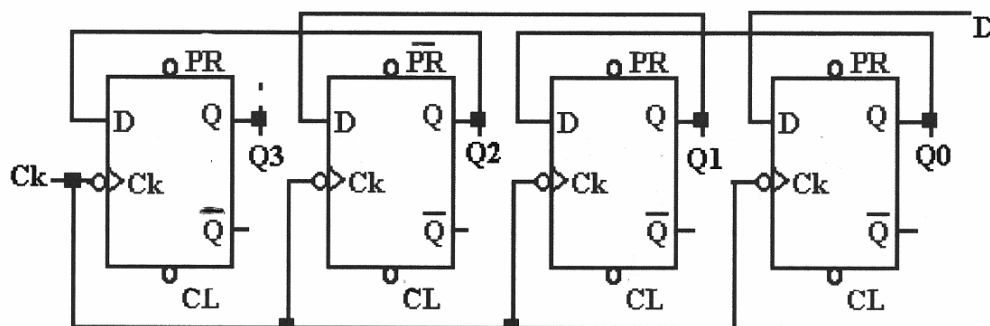


Note-se que a combinação das saídas Q3 Q2 Q1 Q0 representa sucessivamente, a cada impulso de *clock*, os decimais 8, 4, 2 e 1. Portanto, a cada impulso de *clock* o número decimal representado em binário é dividido por dois. O registo efectua uma divisão por dois.

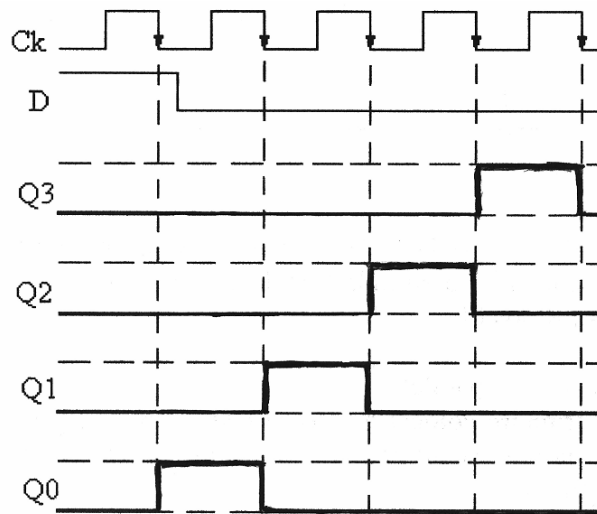
Suponhamos que se apresentava à entrada D do primeiro FF (à esquerda) a sequência de impulsos 1 0 0 1. Ao fim de quatro impulsos de *clock*, esta sequência estaria armazenada no registo, ficando o bit que chegou primeiro armazenado no FF da direita e o último a chegar no FF da esquerda.

### Deslocamento para a esquerda

A figura seguinte mostra como construir um registo de deslocamento para a esquerda utilizando flip-flops tipo D.



Se antes e durante um impulso de clock for aplicado um nível um (1) na entrada D do primeiro FF do registor, esse nível vai sendo **deslocado para a esquerda** ao a cada impulso de clock. O diagrama temporal seguinte ilustra este funcionamento.

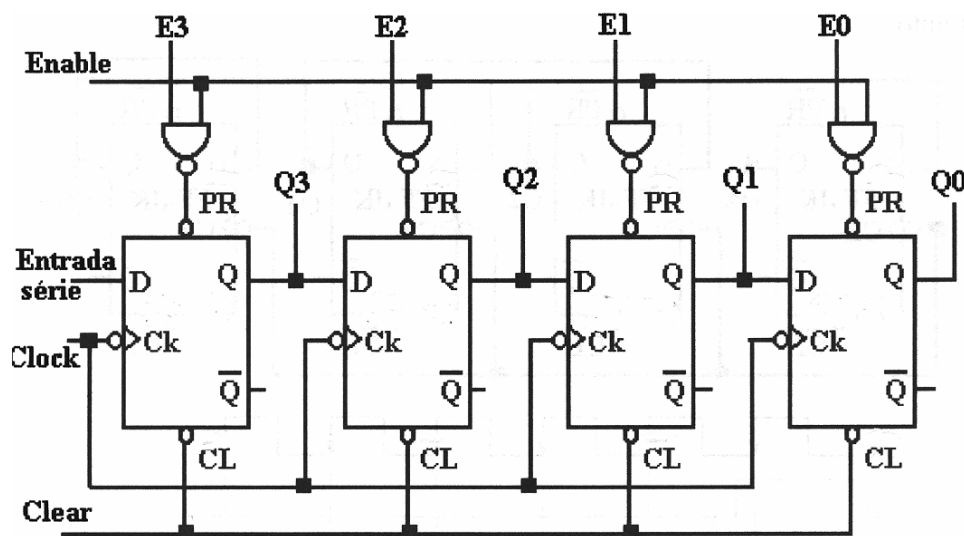


Note-se que a combinação das saídas Q3 Q2 Q1 Q0 representa sucessivamente, a cada impulso de *clock*, os decimais 1, 2, 4 e 8. Portanto, a cada impulso de *clock* o número decimal representado em binário é multiplicado por dois. Este registo efectua uma multiplicação por dois.

Nos registos anteriores, como a entrada de dados se faz um a um na entrada D do primeiro FF estamos na presença de registos com entrada em série. A transmissão de dados pode ser feita na saída do último FF (série) ou nas saídas Q de cada FF (paralelo).

### 6.11.3 Registo de deslocamento genérico

A figura seguinte mostra um registo que pode trabalhar com os dados em série ou em paralelo, na entrada ou na saída:



Para trabalhar com entrada série:

- Dado inserido na entrada D do primeiro FF;
- Entrada Clear em “1” e a entrada Enable em “0” (coloca um “1” nas entradas PR).

Para trabalhar com entrada paralelo:

- 1º - Colocar a entrada *Clear* em “0” e a *Enable* em “0” para gerar um “0” nas saídas Q dos FFs e depois em “1” para permitir o normal funcionamento do registrador;
- 2º - Colocar as informações nas entradas E3, E2, E1 e E0 e depois colocar a entrada *Enable* em “1”, para ocorrer um preset (Q = 1) nos FFs onde a informação for “1”. Depois disso coloca-se a entrada *Enable* em “0” para permitir o normal funcionamento do registrador.

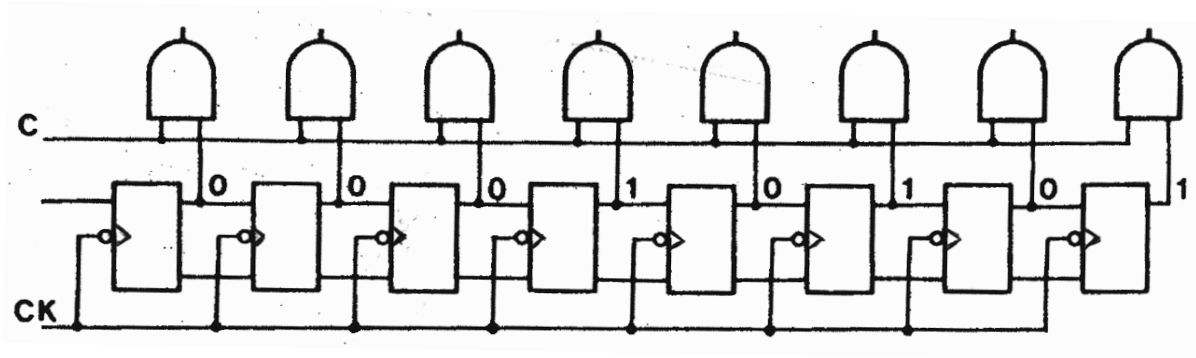
Para trabalhar com saída série:

A cada impulso de *clock* os dados armazenados vão sendo deslocados para a direita e transmitidos um a um pela saída Q0 do último FF.

Para trabalhar com saída paralelo:

Utilizam-se as saídas Q dos FF.

A figura seguinte mostra um **registo que converte dados em série para dados em paralelo**, isto é, todos os bits são transmitidos simultaneamente, cada um na sua linha.



#### 6.11.4 Registro de deslocamento em anel

O registro de deslocamento em anel é um registro que é utilizado para controlar eventos que devem ocorrer numa sequência rigorosa cíclica, por exemplo

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow A \rightarrow \dots$

O registro seguinte e o respectivo diagrama temporal de sinais ilustra este funcionamento.

