

Lógica e Sistemas Digitais - 8

# Circuitos Sequenciais

ISEL

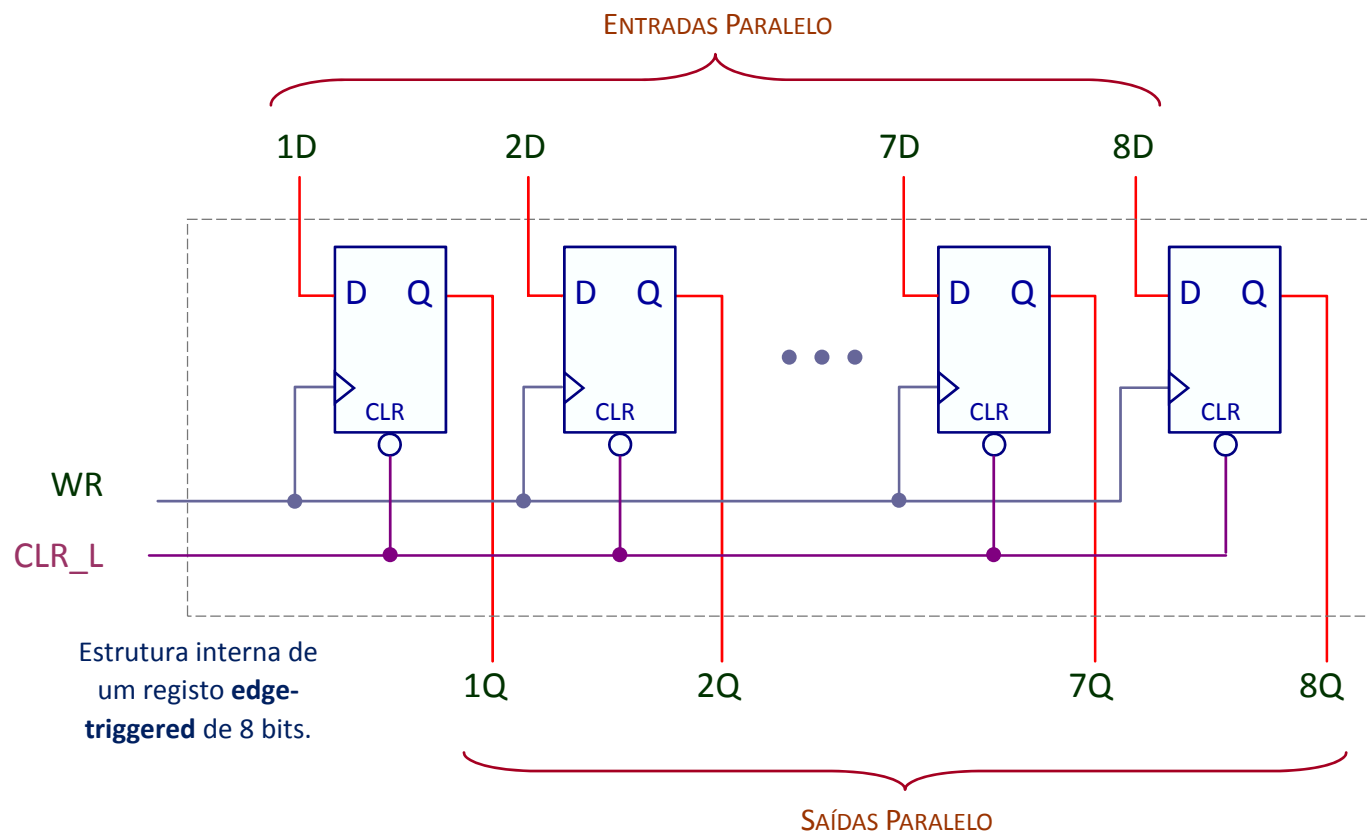
Departamento de Engenharia de Electrónica  
e Telecomunicações e de Computadores  
Lisboa

Mário Araújo

2016-1

## REGISTO SIMPLES (**REGISTER**):

Colecção de 2 ou mais Flip-flops (ou Latches) com sinais de relógio CLK (ou de Enable) e de CLR\_L (Clear) comuns que permitem armazenar **n** bits de informação.



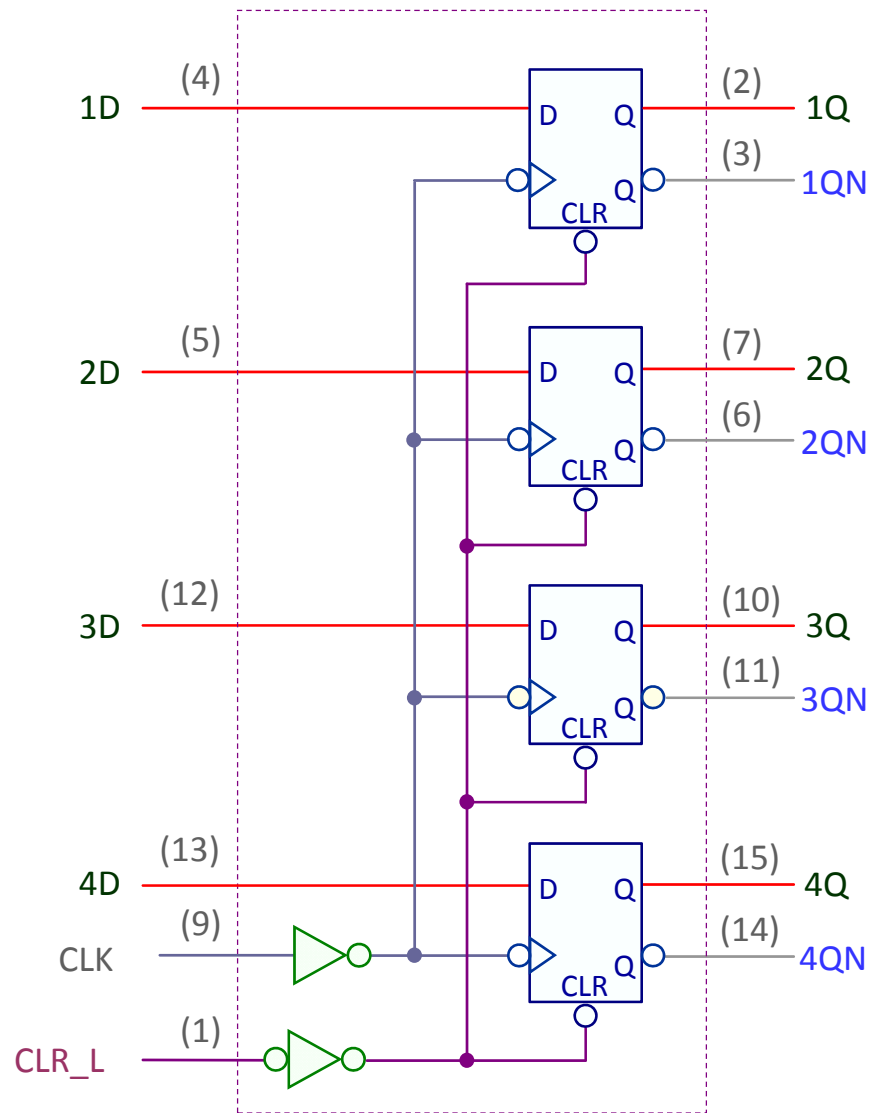
## Registo com 8 flip-flops D:

- Entradas e saídas paralelo
- Controlo de escrita **WR** (**WRITE**)
- Controlo de apagamento **CLR\_L** (**CLEAR**).

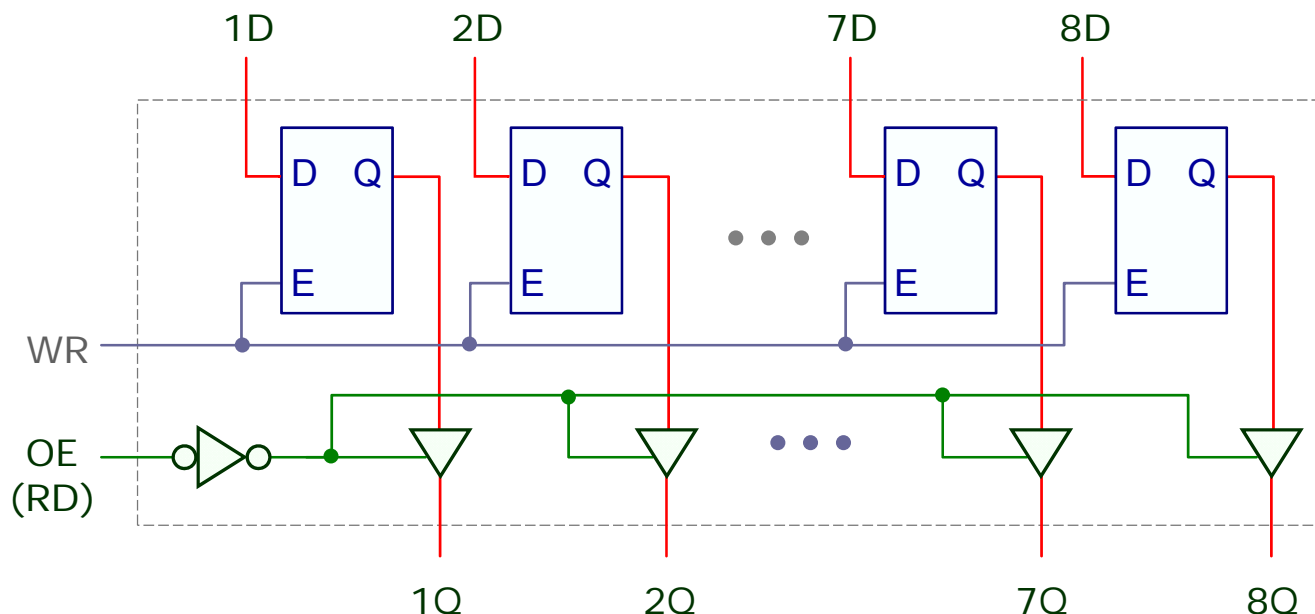
## Funções:

- Carregamento paralelo (**PARALLEL LOAD** ou **WRITE**) – armazenamento de informação paralelo.
- Controlo de apagamento da informação registada (**CLEAR** ou **RESET** assíncrono).

Existem também registos formados por **Latches**. Nos registos de deslocamento (adiante) o uso de **Latches** D é proibido devido às suas características de transparência.

Estrutura interna de um registo **edge-triggered** de 4 bits 74x175.

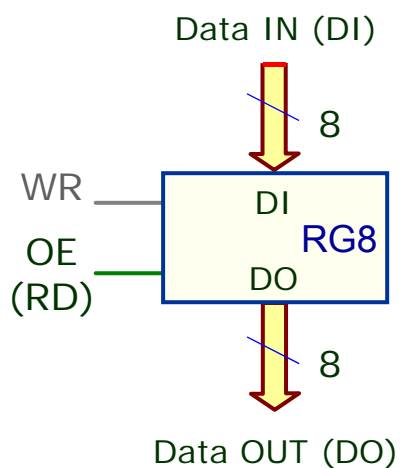
- Os flip-flops D são **negative-edge triggered** mas existe um buffer-inversor interno que os torna **positive-edge triggered** relativamente ao CLK externo.
- Os sinais CLK e CLR\_L possuem **buffers** internos (a verde) para que um dispositivo externo alimentador destas entradas veja apenas **uma carga unitária** em vez de 4 para não diminuir significativamente o seu **fan-out**.
- O controlo de apagamento CLR\_L (**Clear** ou **Reset**) da informação registada faz uso das linhas de CLR\_L **assíncronas** de cada flip-flop D.



Estrutura interna de um registro **latch** de 8 bits com saídas tri-state.

Os registos podem ser construídos com flip-flops **edge-triggered** – caso do slide anterior – ou com dispositivos do tipo **latch**, caso deste registo.

A utilização de **buffers tri-state** nas saídas permite que vários registos possam partilhar o mesmo barramento (**bus**) de saída.



Símbolo lógico do registo **latch** de 8 bits com saídas tri-state.

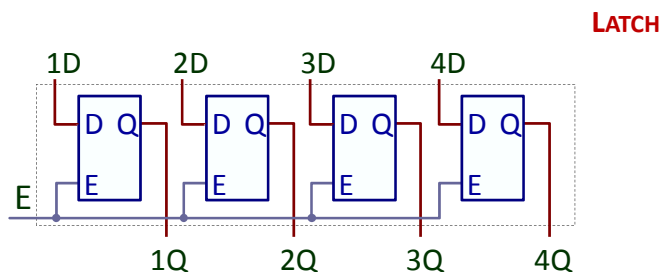
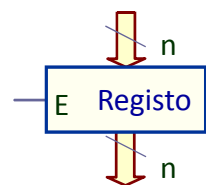
O módulo registo aqui evocado dispõe de sinais de controlo **OE** (**OUTPUT ENABLE** ou **RD** - **READ**) e **WR** (**WRITE**).

**WR:** Quando activo leva os latches a tomar o estado correspondente aos dados presentes nas entradas D. Quando inactivo os latches mantêm o estado anterior.

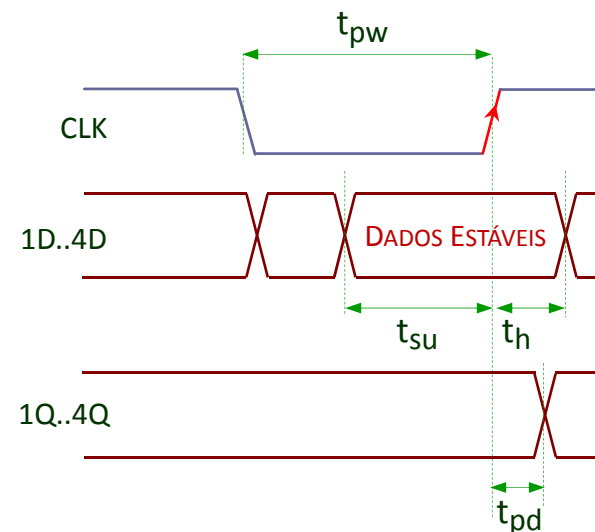
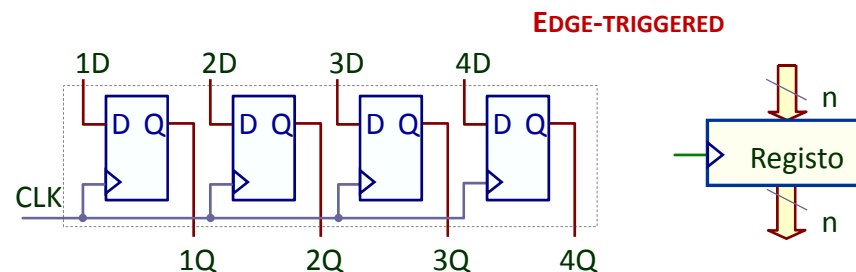
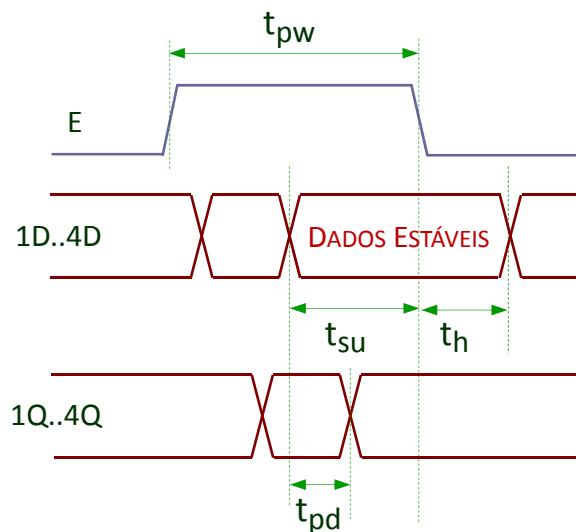
**OE** (ou **RD**): Quando activo coloca em baixa impedância sobre a saída do registo o estado de cada latch constituinte. Quando inactivo mantém em **alta impedância** as saídas do módulo.

# DIAGRAMAS TEMPORAIS PARA A OPERAÇÃO DE ESCRITA EM LATCH E EM FLIP-FLOP

8-5



Símbolos lógicos, estrutura interna e formas de onda relativas a uma acção de memorização num **registo latch** de 4 bits – lado esquerdo – e **registo edge-triggered** de 4 bits – lado direito.



$t_{pw}$  (**PULSE WIDTH**):

duração mínima do sinal de escrita Enable (E) ou Clock (CLK).

$t_{su}$  (**DATA SETUP**):

intervalo de tempo mínimo a observar entre o estabelecimento de informação estável na entrada de dados e o momento da memorização.

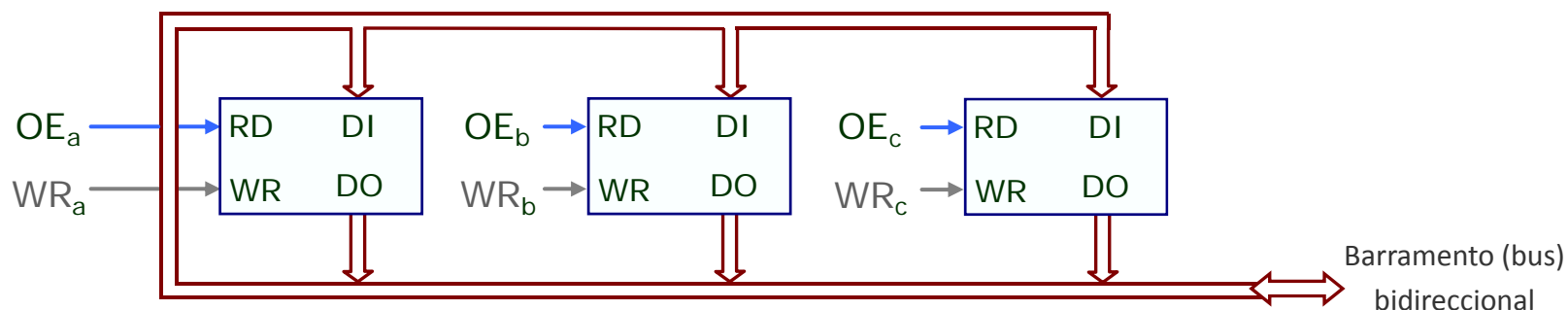
$t_h$  (**DATA HOLD**):

intervalo mínimo de tempo durante o qual ainda se torna necessário manter a informação estável na entrada de dados após o desaparecimento do sinal.

$t_{pd}$  (**PROPAGATION DELAY**):

tempo máximo de propagação entre um sinal na entrada de dados e o seu reflexo na saída do circuito.





Os registos estão ligados por um **barramento (bus)** de dados **bidireccional** - a entrada DI e saída DO dos dados são feitas pelos mesmos pinos como mostrado.

O interesse em usar os mesmos pinos para os dados a ler e a escrever é o de reduzir o número de pinos, o que é útil em memórias de grande capacidade.

Nas aplicações de ligação de memórias a microprocessadores o barramento de dados é normalmente bidireccional.

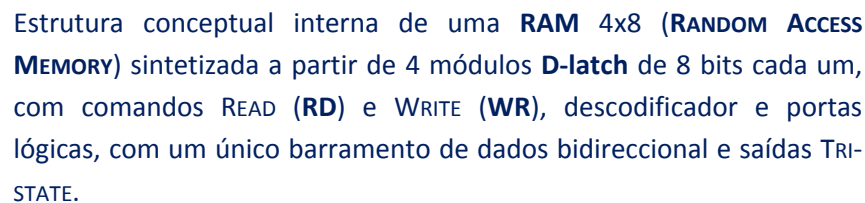
A interligação mostrada permite a transferência de dados entre quaisquer dois registos.

A acção **READ** (**OE<sub>i</sub>** activo) de cada registo i deverá ter a natureza **latch** – os dados têm que se manter presentes em baixa impedância no barramento o tempo suficiente para se estabilizarem.

A acção **WRITE** (**WR<sub>i</sub>** activo) de cada registo i pode tomar a natureza **latch** ou **edge-triggered**.

Existem memórias RAM de várias tecnologias: as mais comuns são as RAMs dinâmicas (DRAM) que baseiam o seu funcionamento na carga acumulada por elementos capacitivos. As RAM estáticas (SRAM) são constituídas por registos – slide seguinte.

As localizações de uma memória RAM estática comportam-se como latches D, e não como flip-flops D. Cada vez que o sinal WR é activado, os dados de entrada fluem através de cada latch, e o valor armazenado é o valor presente quando o latch fecha.



O decodificador 2-to-4 activa um dos 4 sinais de saída de modo a habilitar para leitura ou escrita cada célula (registo) da linha correspondente ao endereço  $A_0$ - $A_1$  aplicado.

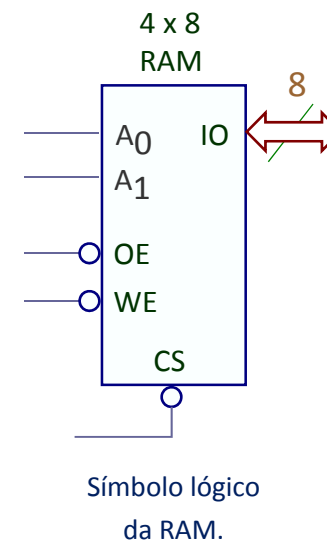
**READ** (leitura)

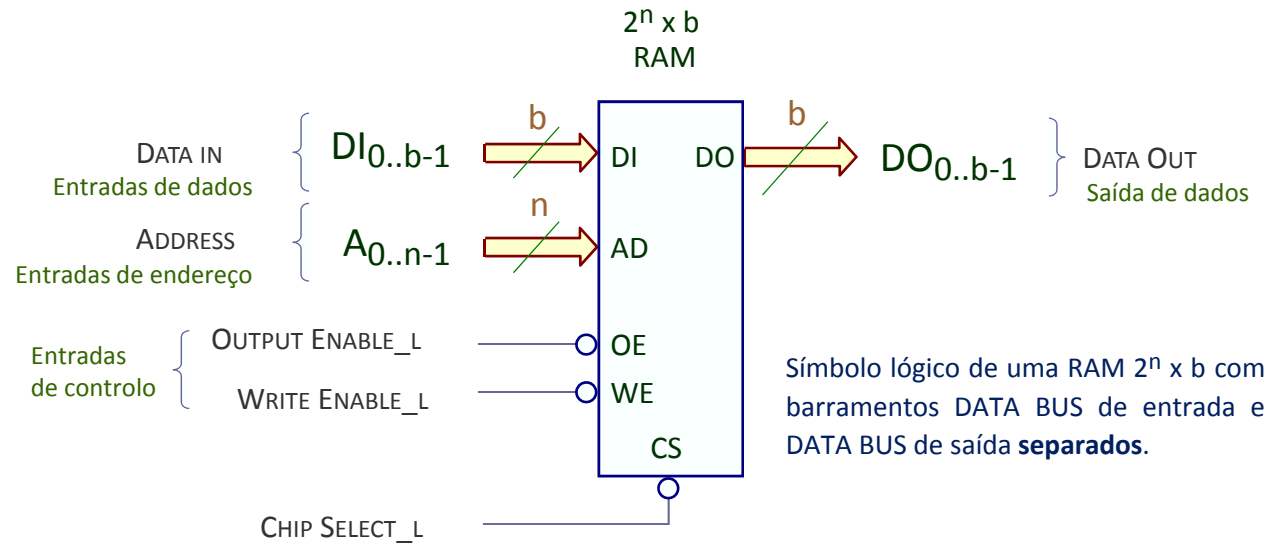
- CS\_L e OE\_L activos a 0, WE\_L inibido a 1;
- endereço colocado em A<sub>0</sub>-A<sub>1</sub>;
- saídas presentes em D0.

**WRITE** (escrita)

- CS\_L e WE\_L activados, OE\_L inibido a 1;
- endereço colocado em A<sub>0</sub>-A<sub>1</sub>;
- palavra de n bits em DI armazenada no endereço seleccionado (Latch Enabled).

Para permitir a expansão da memória de um sistema, é usual as RAMs (tal como as ROMs) disponibilizarem um sinal de entrada **CS (Chip Select)**, para inibição/desinibição: quando desactivo torna a RAM insensível aos sinais de escrita e leitura e coloca o barramento de dados em alta impedância.



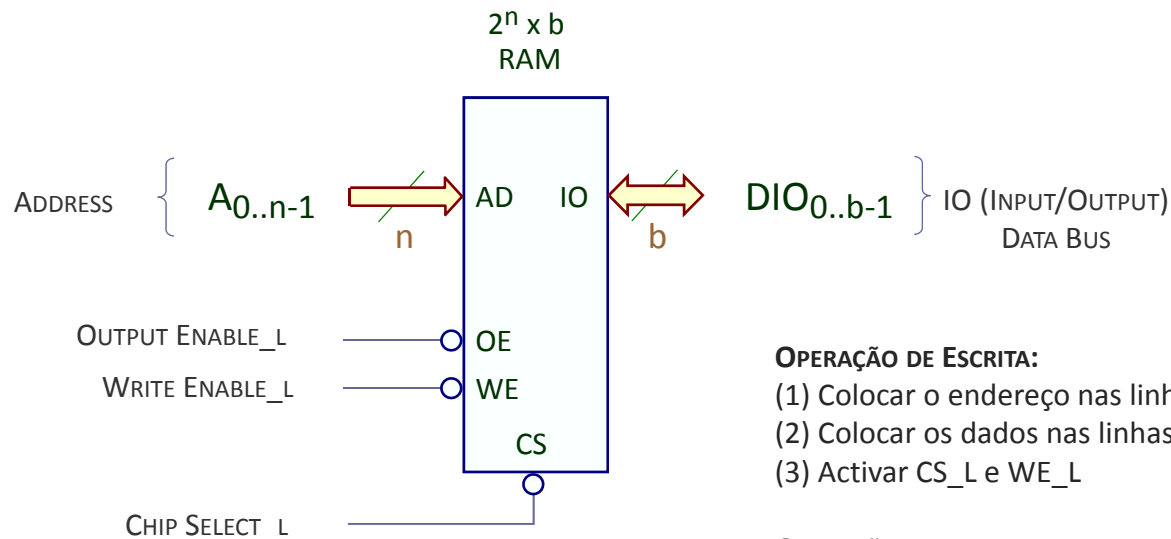


A memória RAM é constituída por  $2^n$  registos de  $b$  bits cada. A cada registo corresponde um endereço próprio que permite a sua invocação para escrita ou leitura.

A RAM tem um barramento de endereço (**ADDRESS BUS**) **unidireccional**, constituído por  $n$  bits que codificam em binário o endereço.

O barramento de dados (**DATA BUS**) pode ser constituído por dois barramentos unidirecionais de  $b$  bits, sendo um para escrita e o outro para leitura (como em cima), ou um único barramento bidireccional para leitura e escrita (como em baixo).

O número de linhas  $b$  que constituem o barramento de dados é igual ao número de bits de cada registo.



Símbolo lógico de uma RAM  $2^n \times b$  com barramento DATA BUS comum **bidireccional**.

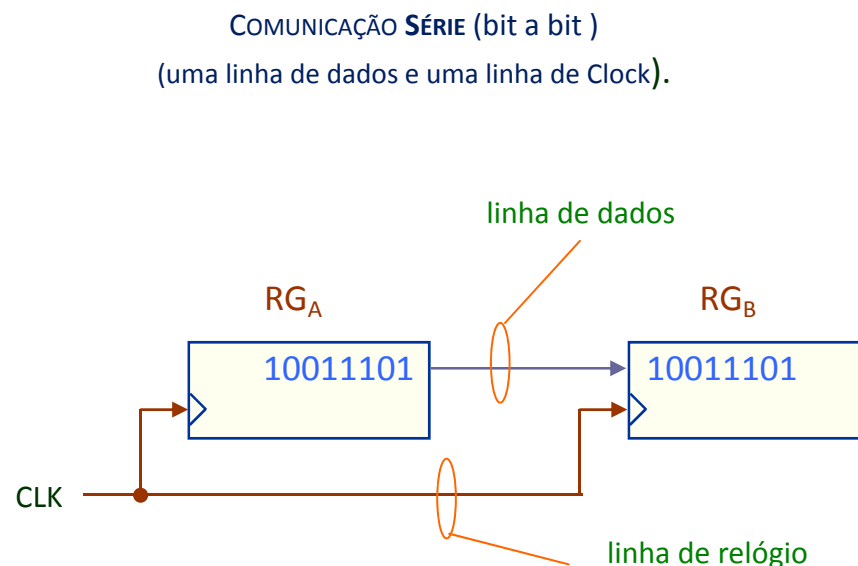
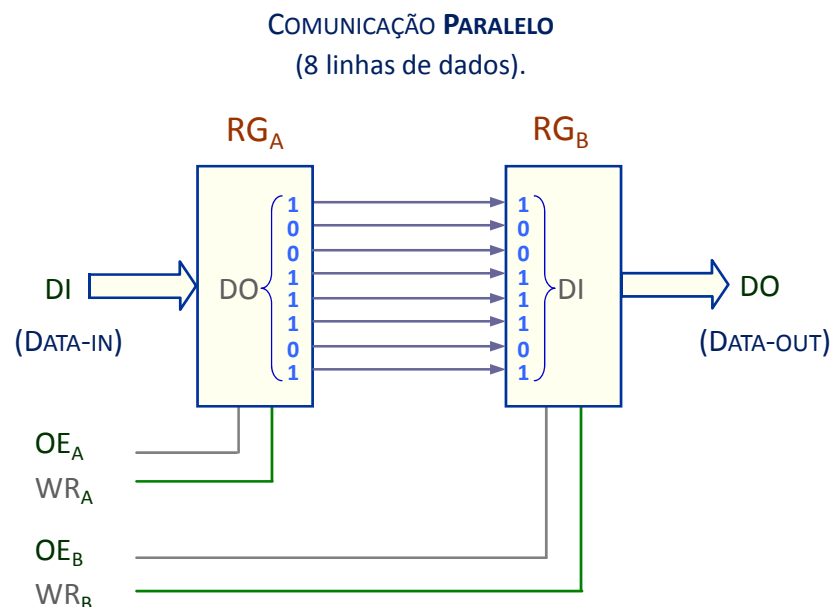
## OPERAÇÃO DE ESCRITA:

- (1) Colocar o endereço nas linhas de endereço  $A_{0..n-1}$ ;
- (2) Colocar os dados nas linhas de entrada de dados ( $DI$  ou  $DIO$ );
- (3) Activar  $CS\_L$  e  $WE\_L$

## OPERAÇÃO DE LEITURA:

- (1) Colocar o endereço nas linhas de endereço  $A_{0..n-1}$ ;
- (2) Activar  $CS\_L$  e  $OE\_L$ .



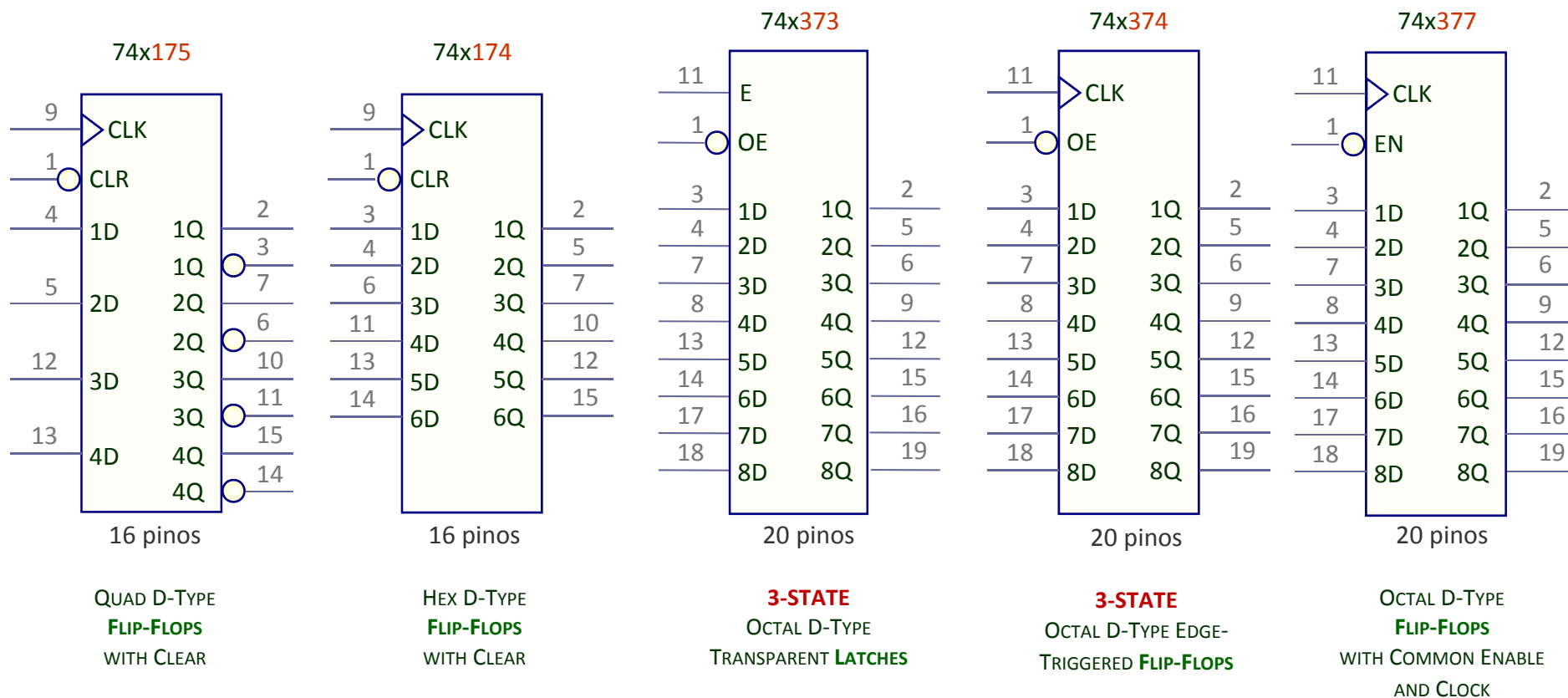


Na transferência de dados **EM PARALELO** entre um **registo-fonte**  $RG_A$  e um **registo-destino**  $RG_B$  os dados são movidos de uma só vez através de linhas múltiplas (8 na figura). É necessário activar simultaneamente :

- O sinal  $OE_A$  (**READ**) no registo-fonte  $RG_A$ .
- O sinal  $WR_B$  (**WRITE**) no registo-destino  $RG_B$ . Este sinal só deverá ser desactivado decorrido o tempo suficiente para estabilização de dados no barramento. Só posteriormente deverá ser desactivado o sinal  $OE_A$  (**READ**) no registo-fonte para garantir que os dados sejam adequadamente recolhidos pelo registo-destino.

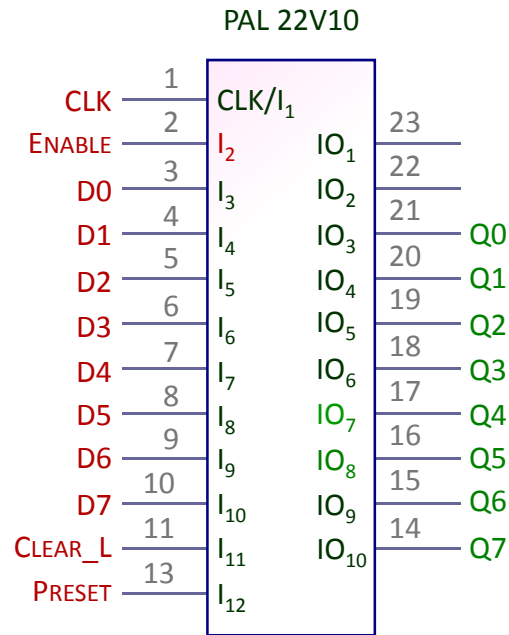
Na transferência de dados **EM SÉRIE** entre um registo-fonte  $RG_A$  e um registo-destino  $RG_B$  os dados são movidos sequencialmente, um bit de cada vez por cada impulso de relógio, através de uma linha única de dados.

Para os 8 bits da figura serão necessários 8 impulsos de relógio. Os registos são do tipo **REGISTO DE DESLOCAMENTO (SHIFT REGISTER)**.



- O circuito 74x174 é semelhante ao 74x175 com mais 2 FF-D (6-bits com eliminação das saídas QN).
- O circuito 74x373 usa D-Latches com Enable (E) e não FF-Ds. A saída de cada um é intersectada por um **Tri-State Buffer**.
- No circuito 74x374 de 8-bits a saída de cada FF-D é também intersectada por um **Tri-State Buffer**. Quando OE (Output Enable) está activo o estado dos flip-flops está presente nas saídas (que permanecem em alta impedância caso contrário).
- O circuito 74x377 necessita de Enable\_L activo (0) e de uma transição de CLK para carregar novos dados.





Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

Considerou-se um registo de 8 bits com ENABLE. A descrição da funcionalidade é feita pela declaração das funções associadas às entradas dos flip-flops do registo. O ENABLE funciona como OUTPUT ENABLE.

```
Name      Register ;
Device    p22v10 ;
```

```
/* ***** INPUT PINS ***** */
PIN 1 = Clk ;
PIN 2 = enable ; /* Output Enable */
PIN [3..10] = [D0..7] /* Data Inputs */
PIN 11 = clear ;
PIN 13 = preset ;

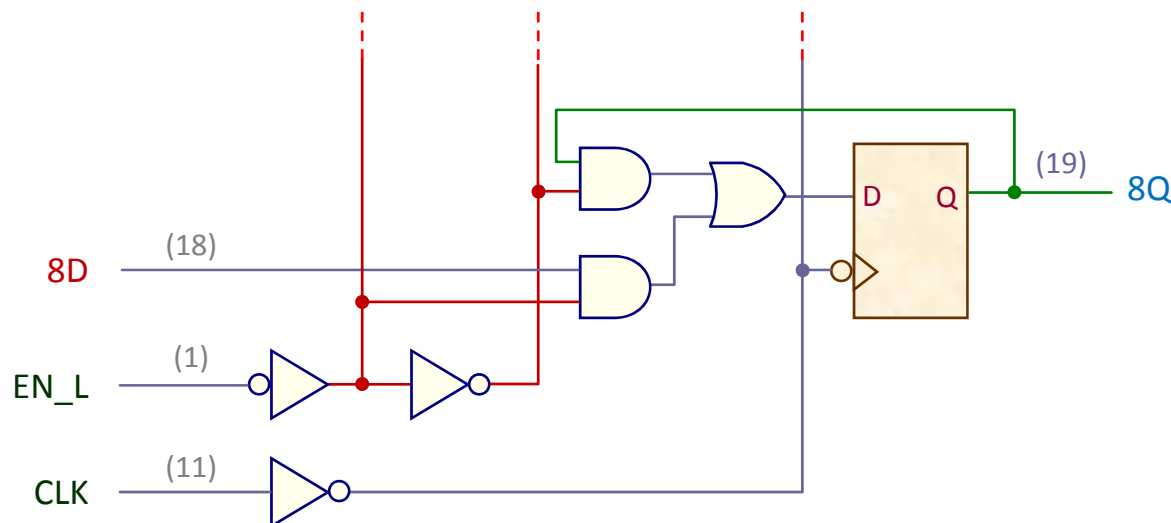
/* ***** OUTPUT PINS ***** */
PIN [21..14] = [Q0..7] ; /*
FIELD output = [Q0..7];
FIELD data = [D0..7];

/* ***** BODY ***** */
output.sp = preset;
output.ar = !clear;
output.OE = enable;
output.d = data;
```

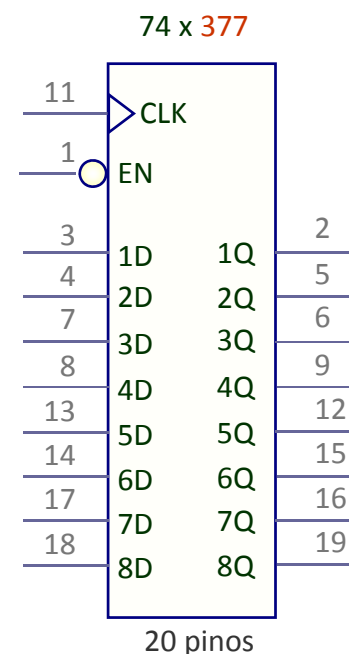
Troço de código CUPL para um REGISTO de 8 bits.

Se o comando output.d fosse descrito do modo alternativo em baixo o ENABLE funcionaria também como CLOCK ENABLE:

```
output.d = data & enable # output & !enable
```



Representação da lógica interna para a oitava célula do registo 74 x 377.



OCTAL D-TYPE

**FLIP-FLOPS**

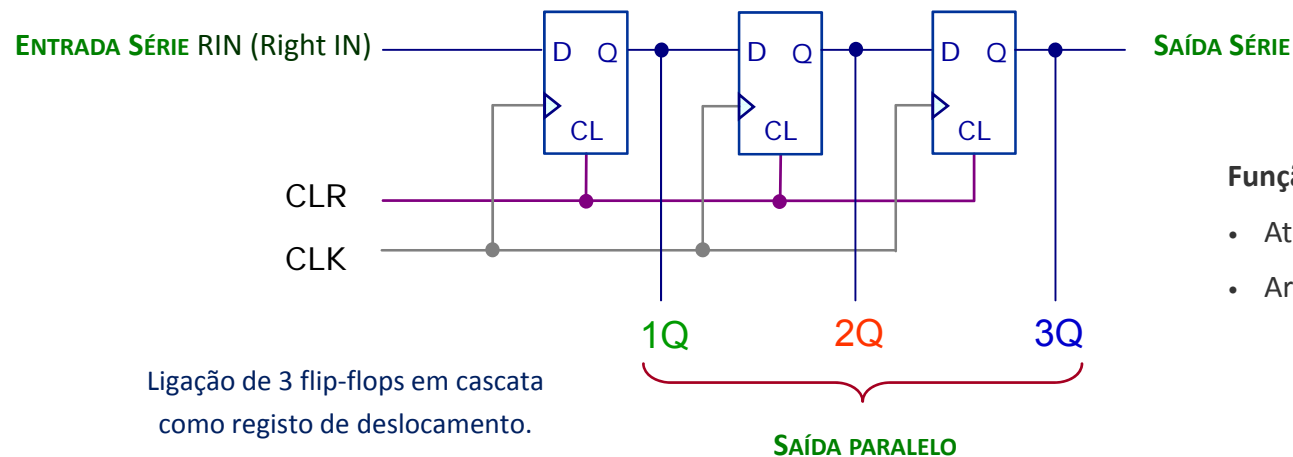
WITH COMMON ENABLE

AND CLOCK

O circuito 74 x 377 é um registo edge-triggered como o seu par 74 x 374, mas não possui saídas tri-state. Em vez dessa característica possui no pino (1) um sinal de **ENABLE ACTIVE-LOW** (EN\_L ou G\_L).

Verifica-se pela lógica que:  $EN\_L = 0$ :  $8D \Rightarrow 8Q$  : o flip-flop regista o valor da entrada na próxima transição ascendente de Clock.

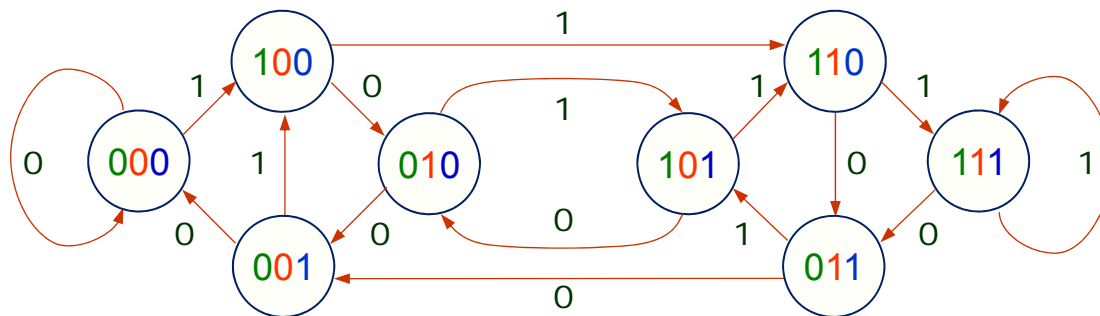
$EN\_L = 1$ :  $8Q \Rightarrow 8Q$  : o flip-flop mantém o valor presente na saída na próxima transição ascendente de Clock.



## Função DESLOCAMENTO (SHIFT):

- Atrasa a informação na saída série;
- Armazena informação na saída paralelo.

$$Q_i^+ \leftarrow Q_{(i-1)}$$

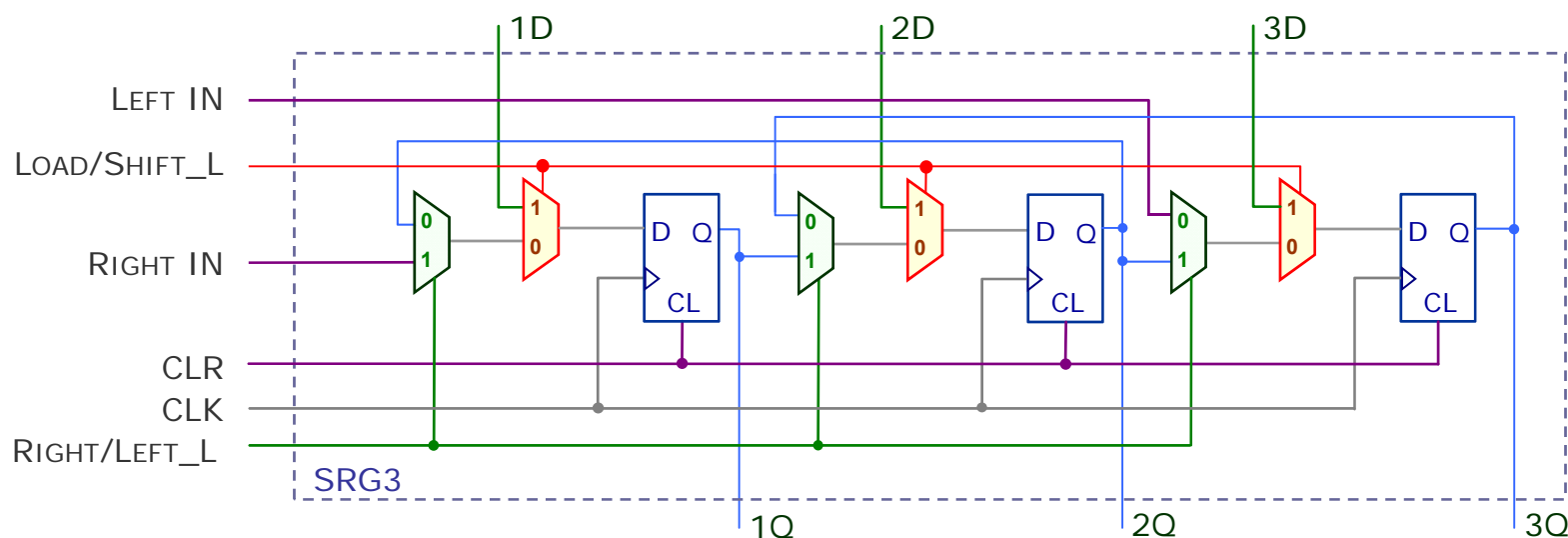


Sequências de estados possíveis.

- O registo evolui segundo uma das sequências de 8 estados da figura.
- O valor de cada estado está representado no interior de cada círculo (nó).
- A existência de 8 estados de contagem é devida à utilização de 3 flip-flops.
- O valor na entrada série RIN é mostrado em cada arco de transição entre estados.

## MODOS DE FUNCIONAMENTO

- **CLEAR:** Iniciação **assíncrona** a 0 de todos os flip-flops (enquanto CLR estiver a 1);
- **SHIFT RIGHT:** Deslocamento à direita (quando Right/Left\_L estiver a 1);
- **SHIFT LEFT:** Deslocamento à esquerda (quando Right/Left\_L estiver a 0);
- **PARALLEL LOAD:** Transferência em paralelo, **síncrona** com CLK, dos valores presentes nas entradas de dados para as saídas.



Estrutura interna típica de um registo de deslocamento bidireccional de 3 bits.

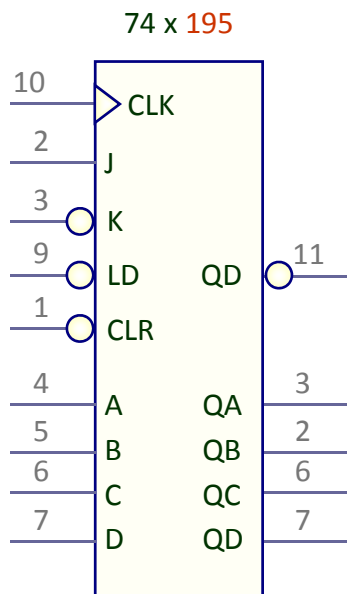
**SRG3** – registo de deslocamento de 3 bits.

## ENTRADAS

- **CLK:** Clock, impulso de relógio.
- **LOAD/SHIFT\_L:** Entrada de controle que distingue entre LOAD (1), e SHIFT (0).
- **RIGHT/LEFT\_L:** Entrada de controle que distingue entre RIGHT (1), e LEFT (0).
- **LEFT IN, RIGHT IN:** Entradas de dados série.
- **CLR:** Clear, entrada de controle assíncrona.
- **1D, 2D, 3D:** Entradas de dados paralelo.

## SAÍDAS

- **1Q, 2Q, 3Q:** saídas de dados em paralelo.



Configuração dos pinos.

As entradas J-K' estão disponíveis exteriormente, para permitir maior versatilidade de utilização.

Só há dois modos de operação, Load e Shift Right (convenção: D está à direita).

Para o Shift Left há que interligar as saídas com as entradas do modo indicado ao lado.

## ENTRADAS

- **CLK:** Clock (impulso de relógio).
- **J e K':** Entradas de dados série (**SERIAL INPUTS**).
- **LD\_L:** Load **síncrono**.
- **CLR\_L:** Clear ou Master Reset **assíncrono**.
- **A, B, C, D:** Entrada de dados paralelo (**PARALLEL INPUTS**).

## SAÍDAS

- **QA, QB, QC, QD:** Saída de dados (**PARALLEL OUTPUTS**).
- **QD':** QD negado.

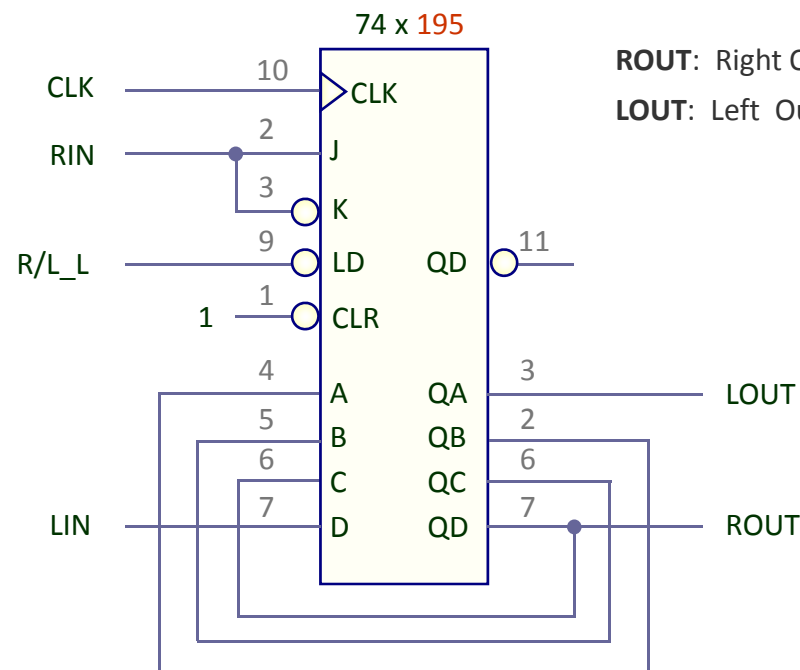
**RIN:** Right In

**LIN:** Left In

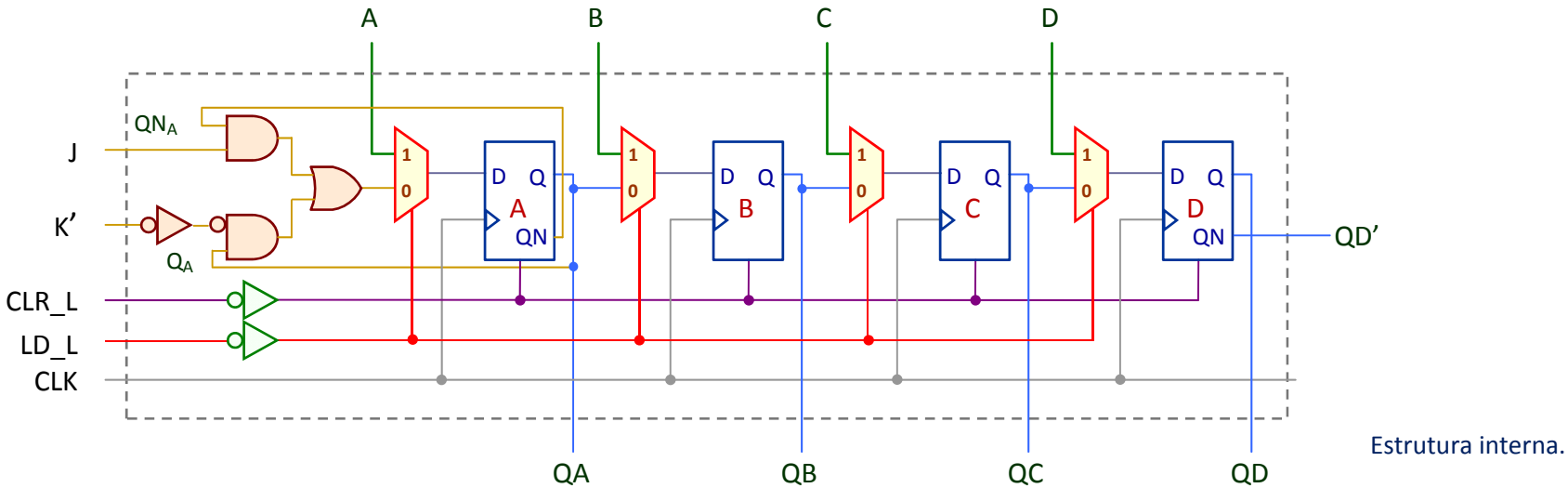
**R/L\_L:** Right/Left

**ROUT:** Right Out

**LOUT:** Left Out



Interligação dos pinos do circuito 74 x 195 para permitir acções **shift-right** e **shift-left**.



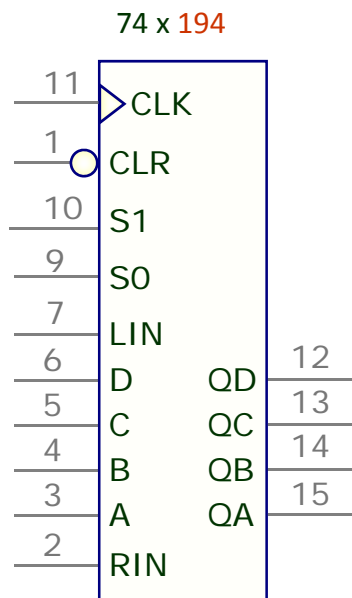
Estrutura interna.

74 x 195 MODO DE OPERAÇÃO	ENTRADAS										SAÍDAS				
	CONTROLE			SÉRIE		PARALELO									
	CLK	CLR_L	LD_L	J	K'	A	B	C	D		QA <sup>+</sup>	QB <sup>+</sup>	QC <sup>+</sup>	QD <sup>+</sup>	(QD <sup>+</sup> )'
RESET Assíncrono	-	0	-	-	-	-	-	-	-		0	0	0	0	1
PARALLEL LOAD Síncrono	↑	1	0	-	-	a	b	c	d		a	b	c	d	d'
SHIFT RIGHT, Reset 1.º flip-flop	↑	1	1	0	0	-	-	-	-		0	QA	QB	QC	QC'
SHIFT RIGHT, Mantém 1.º flip-flop	↑	1	1	0	1	-	-	-	-		QA	QA	QB	QC	QC'
SHIFT RIGHT, Inverte 1.º flip-flop	↑	1	1	1	0	-	-	-	-		QA'	QA	QB	QC	QC'
SHIFT RIGHT, Preset 1.º flip-flop	↑	1	1	1	1	-	-	-	-		1	QA	QB	QC	QC'

Tabela funcional.







Configuração dos pinos.

**S0** e **S1** são entradas de controle que seleccionam a função do SHIFT-REGISTER. Convenção: **D está à direita**.

**UNIVERSAL** (ou MULTIMODO) significa um conjunto de atributos como entrada e saída paralelo (PARALLEL-IN, PARALLEL-OUT), entradas série (LIN e RIN) e saídas série nos flip-flops dos extremos, e funções de manutenção (HOLD), carregamento (LOAD) e deslocamento (SHIFT) com bidireccionalidade (SHIFT-RIGHT e SHIFT-LEFT).

### ENTRADAS

- **CLK**: Clock
- **LIN (LEFT IN)**: Entrada de dados série (**SERIAL INPUT LEFT**).
- **RIN (RIGHT IN)**: Entrada de dados série (**SERIAL INPUT RIGHT**).
- **S0** e **S1**: Pinos de selecção do modo de funcionamento.
- **CLR\_L**: **CLEAR** ou **MASTER RESET assíncrono** (iniciação a 0).
- **A, B, C, D**: Entrada de dados paralelo (**PARALLEL INPUTS**).

### SAÍDAS

- **QA, QB, QC, QD**: Saída de dados (**PARALLEL OUTPUTS**).

FUNÇÃO	ENTRADAS		PRÓXIMO ESTADO			
	S1	S0	QA <sup>+</sup>	QB <sup>+</sup>	QC <sup>+</sup>	QD <sup>+</sup>
HOLD	0	0	QA	QB	QC	QD
SHIFT RIGHT	0	1	RIN	QA	QB	QC
SHIFT LEFT	1	0	QB	QC	QD	LIN
LOAD (Sync)	1	1	A	B	C	D

Tabela funcional para o circuito 74 x 194.

## SHIFT REGISTER DE 8 BITS EM PAL 22V10

8-18

```

Name      ShiftRegister ;
Device    p22v10 ;

/* ***** Entradas ***** */
PIN 1 = clock;      /* Clock */
PIN[6..4] = [S2..0]; /* Número de Shifts a efectuar */
PIN 2 = enable;     /* Output Enable das Saídas */
PIN 3 = sin;        /* Serial-in Left */
PIN 8 = clr;        /* Clear Síncrono */

/* ***** Saídas ***** */
PIN [22..15]=[Q7..0]; /* Saídas do Registo */

/****** Declarações e Variáveis Intermédias ******/
field shift = [S2..0]; /* Número de Shifts */
field output = [Q7..0]; /* Saídas */

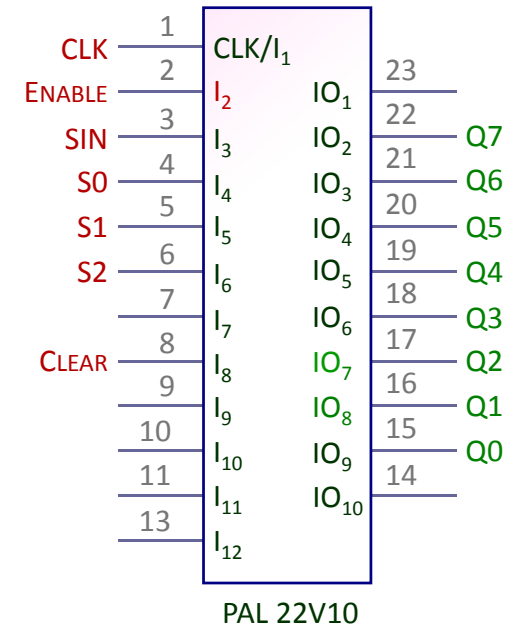
output.sp = 'h'00;
output.ar = 'h'00;
output.oe = enable; /* Controle tri-state */

/* ***** Equações Lógicas ***** */

output.d = !clr&([Q7, Q6, Q5, Q4, Q3, Q2, Q1, Q0] & shift:0
# [Q0, Q7, Q6, Q5, Q4, Q3, Q2, Q1] & shift:1
# [Q1, Q0, Q7, Q6, Q5, Q4, Q3, Q2] & shift:2
# [Q2, Q1, Q0, Q7, Q6, Q5, Q4, Q3] & shift:3
# [Q3, Q2, Q1, Q0, Q7, Q6, Q5, Q4] & shift:4
# [Q4, Q3, Q2, Q1, Q0, Q7, Q6, Q5] & shift:5
# [Q5, Q4, Q3, Q2, Q1, Q0, Q7, Q6] & shift:6
# [Q6, Q5, Q4, Q3, Q2, Q1, Q0, Q7] & shift:7);

append output.d=!clr&sin&'b'10000000;
    
```

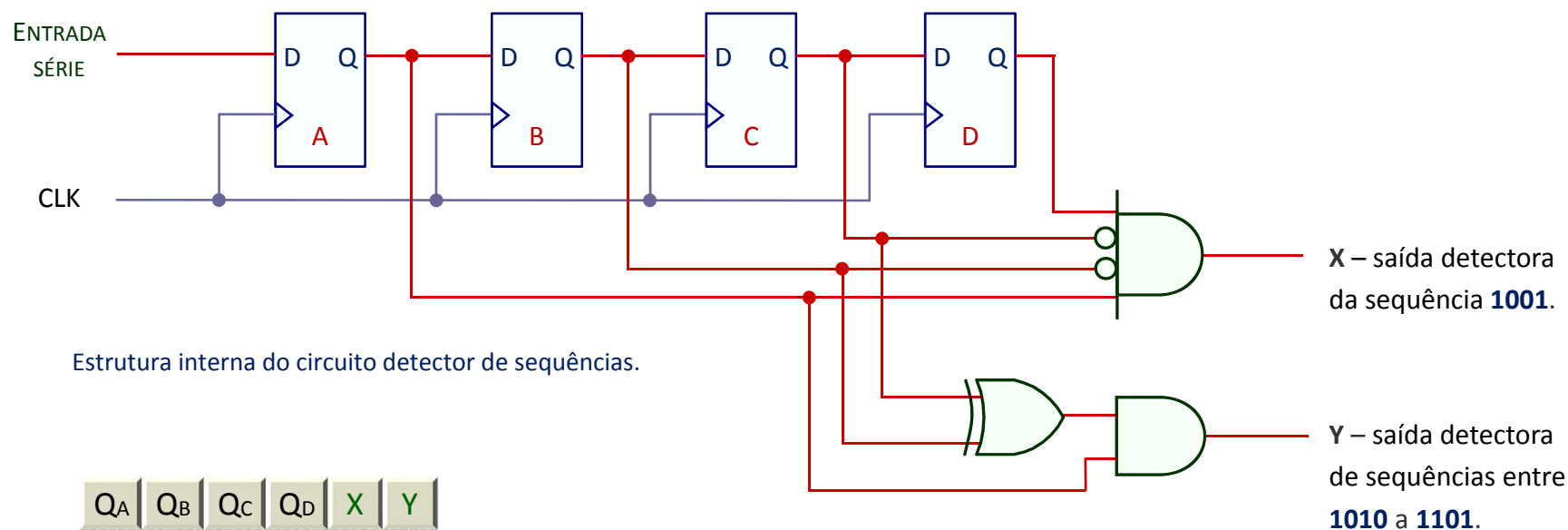
Troço de código  
CUPL para um SHIFT  
REGISTER de 8 bits  
com CLEAR síncrono.



Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

O deslocamento é realizado na direcção Q7 → Q0 e o número de de posições a deslocar de cada vez é indicado pelo valor binário de S0..S1.





Estrutura interna do circuito detetor de sequências.

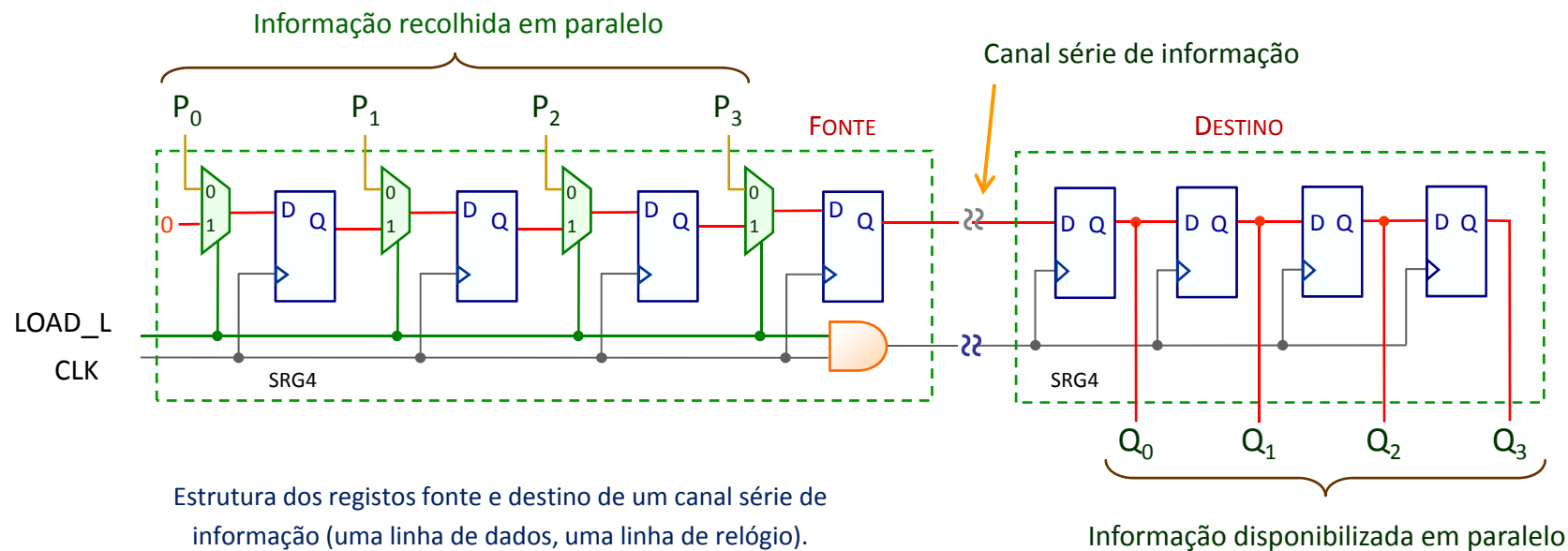
Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>	X	Y
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	0	0

Tabela parcial evidenciando as sequências a ser detectadas.

O circuito efectua constantemente a leitura de palavras com 4 bits e torna activas:

- a saída X quando a palavra lida corresponde à sequência **1001**;
- a saída Y quando as palavras lidas pertencem ao intervalo [**1010** a **1101**].

O bit menos significativo é primeiro a ser lido e a saída é válida apenas em cada quatro impulsos de relógio.

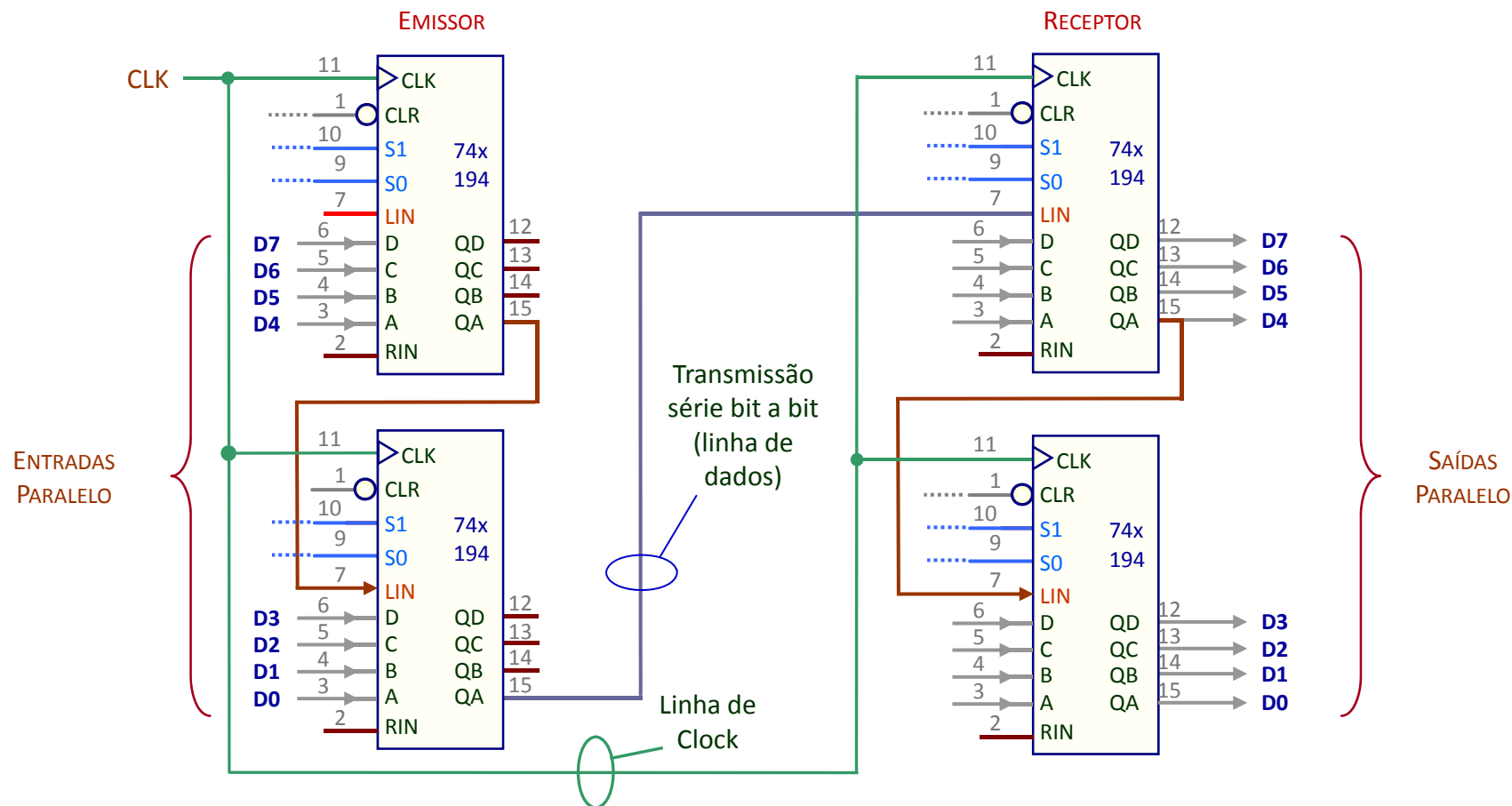


Canal de comunicação série com conversão **paralelo-série** na fonte e **série-paralelo** no destino:

- A informação é recolhida em paralelo das entradas  $P_i$  para as saídas  $Q_i$  do registo-fonte com **LOAD\_L** activo (0).
- A informação é **transferida em série** para o local de recepção em 4 impulsos de clock com **LOAD\_L** inactivo (1).
- A informação é recolhida em paralelo e fica disponível nas saídas  $Q_i$  do registo de destino ao fim de 4 impulsos de relógio.

FONTE				DESTINO			
1	1	0	1	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	1	0	1	0	0
0	0	0	1	1	0	1	0
0	0	0	0	1	1	0	1

Exemplo para uma sequência de estados dos flip-flops ao longo dos 4 impulsos de relógio que comandam a comunicação.



Transmissão bit a bit de uma sequência de 8 bits de informação ao ritmo dos impulsos de relógio com conversão paralelo-série no emissor e série-paralelo no receptor através dos circuitos 74x194.

A ligação de  $n$  flip-flops em cascata, dos quais um está a 1 e os outros a 0, como registo de deslocamento, pode também ser usada como um contador simples, usando um mínimo de hardware. O contador é designado **CONTADOR EM ANEL ( RING COUNTER )**.

Evolui segundo a sequência de 4 estados na tabela, repetindo-a.

O contador é muito rápido (não existem portas lógicas entre flip flops), mas é ineficiente em termos do número total de estados de contagem disponíveis (só usa  $n$  estados dos  $2^n$  estados disponíveis).

É necessário inicializá-lo num dos  $n$  estados de contagem como mostra o diagrama em baixo.

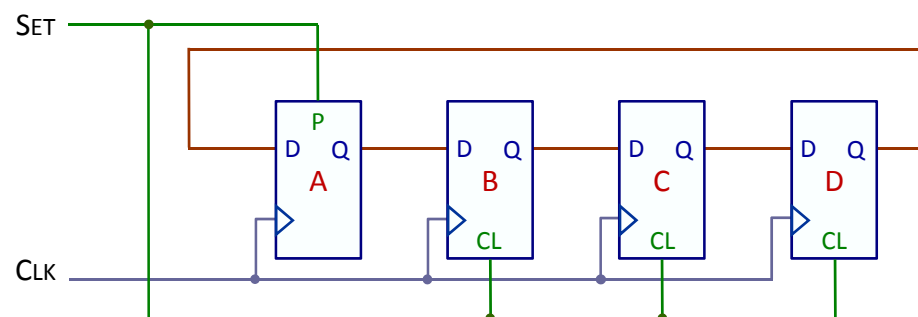


Diagrama lógico de um contador em anel com um único 1 circulante realizado com flip-flop tipo D.

	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
1	1	0	0	0

Tabela de sequência de estados.

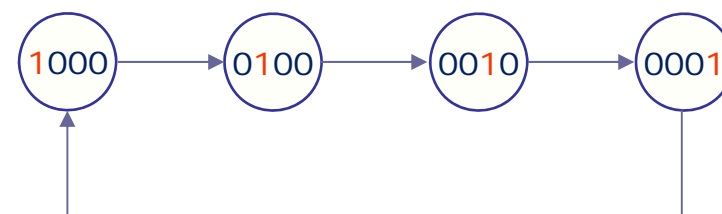


Diagrama de evolução de estados.

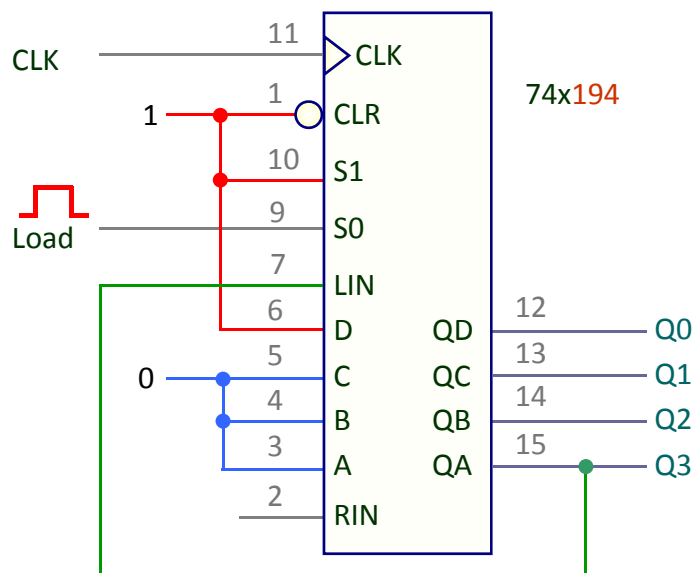


Diagrama lógico de um contador em anel com um único 1 circulante realizado com o circuito 74 x 194.

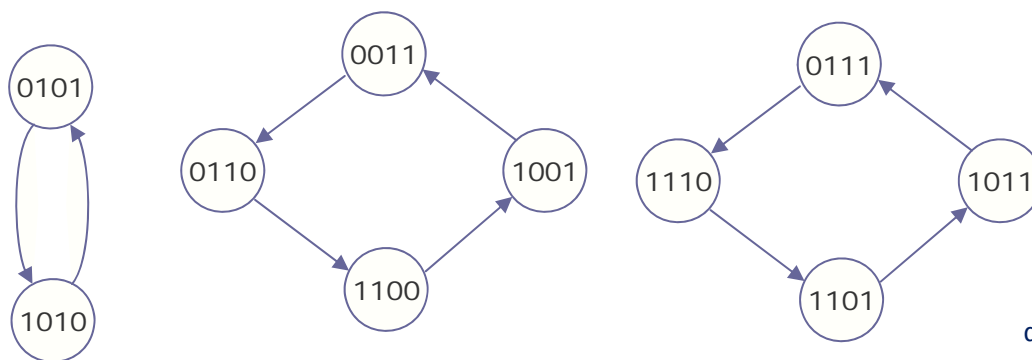
## ACÇÕES:

**LOAD** de 0001 com  $Q0 = QD = D = 1$ , todos os outros bits a 0;  $S0 = S1 = 1$ ;

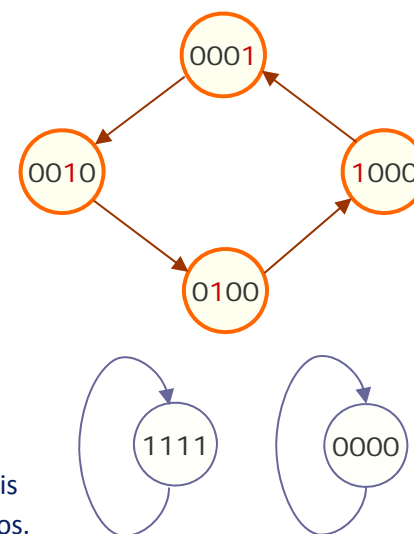
**SHIFT LEFT** direcção  $QD \rightarrow QA$  na sequência de 4 estados possíveis: 0001, 0010, 0100, 1000, 0001, ... ;  $S0=0$  e  $S1=1$ .

O contador não é robusto. Uma falha no circuito (por captação de ruído excessivo, por exemplo) pode:

- Levar a **única** saída activa a 1 a tomar o valor 0. O circuito passa ao estado 0000 e fica nesse estado por tempo indeterminado; ou
- Inserir um 1 extra criando por exemplo o estado 0101. O circuito seguirá um ciclo de estados incorrectos e permanecerá nesse ciclo por tempo indeterminado deixando de funcionar como contador em anel.



Sequência normal de 4 estados.



Sequências possíveis de estados incorrectos.





O CONTADOR DE JOHNSON é semelhante ao CONTADOR EM ANEL, mas a realimentação é feita através da negação do último bit (em vez da ligação directa ao último bit , como no contador em anel). Com  $n$  flip-flops o contador JOHNSON usa apenas  $2n$  dos  $2^n$  estados disponíveis.

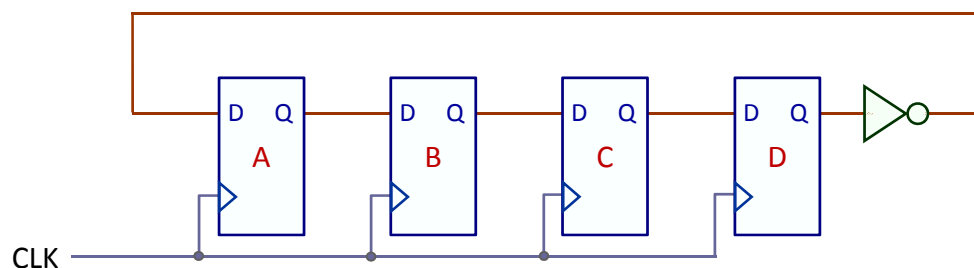


Diagrama lógico de um contador **Johnson** de 4 bits implementado à custa de flip-flops D.

	A	B	C	D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
0	0	0	0	0

Tabela de sequência de estados.

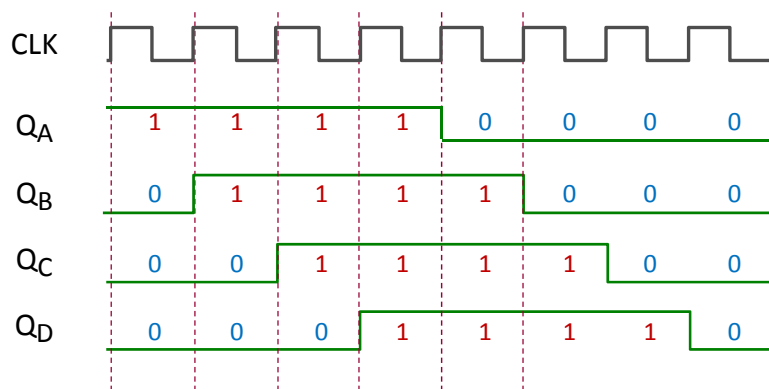


Diagrama temporal de formas de onda nas saídas  $Q_A$  a  $Q_D$ .

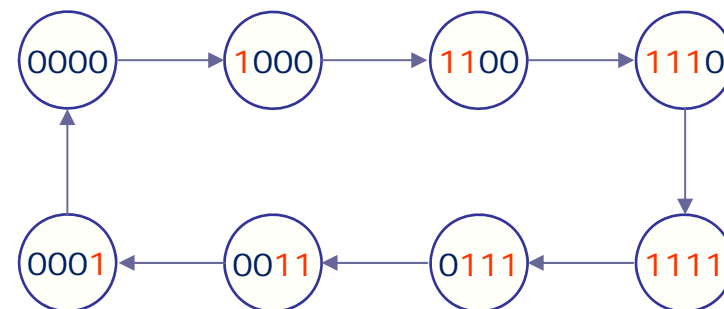


Diagrama de evolução de estados.

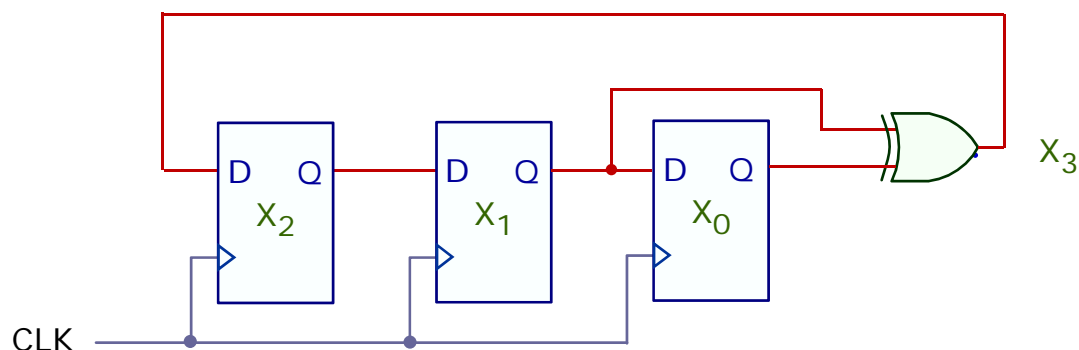
Um contador **Linear Feedback Shift Register (LFSR)** é constituído por um registo de deslocamento de **n** bits e por uma **função de realimentação**, estruturada a partir de portas XOR.

Um LFSR produz sequências cíclicas com quase o máximo número de estados :  $2^n - 1$  dos  $2^n$  estados possíveis – a configuração zero é excluída.

Os contadores LFSR pertencem a uma classe importante de geradores de códigos cíclicos que recebe também a designação de **MAXIMUM LENGTH GENERATOR (MLG)** ou de **GERADOR DE RUÍDO PSEUDO-ALEATÓRIO**.

Têm particular interesse nas técnicas de teste e de caracterização de sistemas, de codificação de canal e de criptografia.

Por convenção, as saídas  **$x_0$**  a  **$x_n$**  são sempre numeradas na direcção e do modo indicado na figura - **n** é o número de flip-flops.



Realização de um LFSR com 3 flip-flops.

$$X_3 = X_0 \oplus X_1$$

Equação de realimentação de um LFSR de 3 bits.

Sequência máxima de estados de comprimento  $2^3 - 1 = 7$  para um contador LFSR de 3 flip-flops (configuração 0 excluída).

	$X_2$	$X_1$	$X_0$
1	0	0	1
4	1	0	0
2	0	1	0
5	1	0	1
6	1	1	0
7	1	1	1
3	0	1	1
1	0	0	1

Por exemplo, a cifra utilizada no Standard Europeu GSM de comunicações móveis utiliza três LFSRs (de 19, 22 e 23 bits).

A construção de contadores LFSR é baseado na teoria dos Campos de Galois (do nome do matemático francês Evariste Galois) – o funcionamento de um LFSR corresponde a operações num campo de Galois com  $2^n$  elementos.

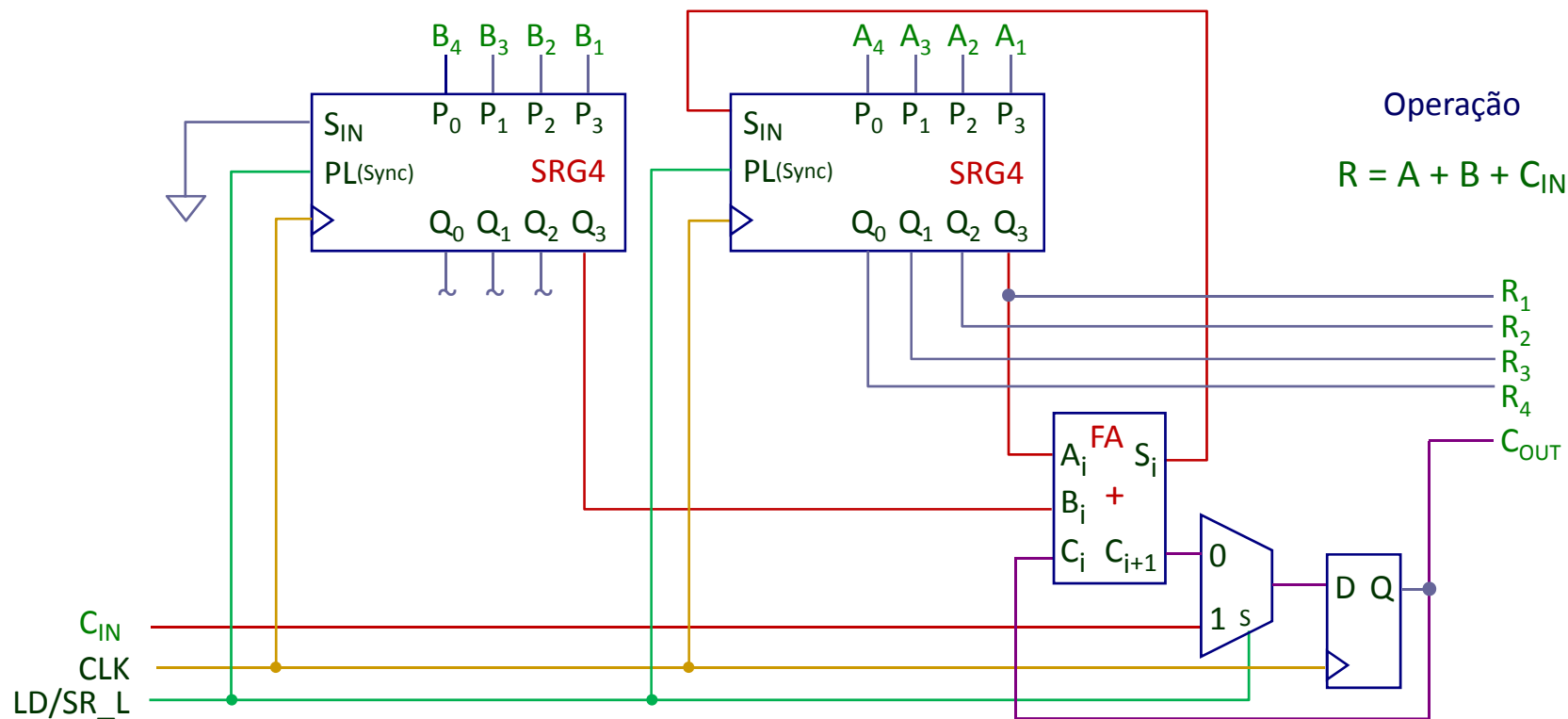


Évariste Galois  
1811 - 1832

Pela teoria dos Campos de Galois pode demonstrar-se que para qualquer valor de  $n$  há pelo menos uma equação de realimentação que faz o contador LFSR evoluir através de todos os  $2^n - 1$  estados antes da repetição. A sequência assim obtida é uma sequência de comprimento máximo (MAXIMUM LENGTH).

$n$ (nº de FFs)	Expressão de $X_n$
2	$X_1 \oplus X_0$
3	$X_1 \oplus X_0$
4	$X_1 \oplus X_0$
5	$X_2 \oplus X_0$
6	$X_1 \oplus X_0$
7	$X_3 \oplus X_0$
8	$X_4 \oplus X_3 \oplus X_2 \oplus X_0$
9	$X_4 \oplus X_0$
10	$X_3 \oplus X_0$

Equação de Realimentação  
para vários valores de  $n$ .



O primeiro impulso de CLK provoca:

- a inicialização dos SHIFT REGISTER de 4 bits (SRG4) com  $LD/SR\_L = 1$ , carregando neles os operandos A e B e
- o carregamento de  $C_{IN}$  no flip-flop D.

De seguida, com  $LD/SR\_L = 0$ , por cada CLK é registada no bit  $Q_0$  do SHIFT REGISTER de 4 bits da direita, a soma  $S_i$  de um bit  $A_i$  do operando A ( $A_{1..4}$ ), com um bit  $B_i$  do operando B ( $B_{1..4}$ ), e um bit CARRY-IN ( $C_i$ ) – proveniente da propagação do CARRY-OUT de peso anterior, ou do  $C_{IN}$  inicial. O somador FA (FULL ADDER) realiza a operação de soma.

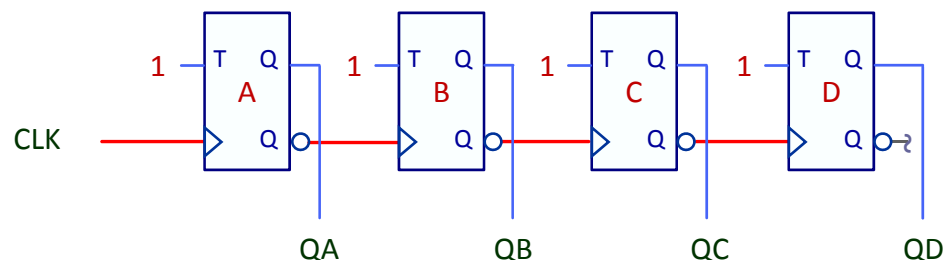
O CARRY-OUT ( $C_{i+1}$ ) proveniente de FA é registado na saída Q do flip-flop D para ser adicionado aos bits de peso seguinte dos operandos A e B.

Ao fim de 4 impulsos de CLK o resultado R ( $R_{1..4}$ ) da adição estará presente nas saídas do SHIFT REGISTER da direita ( $Q_{3..0}$ ), e o  $C_{OUT}$  final na saída do flip-flop D.

## CONTADOR ASSÍNCRONO POR PULSAÇÃO (RIPPLE COUNTER)

8-29

CONTADORES são registos com funções adicionais. O **módulo** do contador é o número de estados no ciclo de contagem. Um contador com m-estados é designado **contador módulo-m**, ou **divisor por m**.



Contador binário **assíncrono** de 4 bits sintetizado a partir de flip-flops T.

- Para a construção do contador assíncrono não são necessários componentes adicionais para além dos flip-flops.
- Cada entrada de CLK está ligada à saída QN do flip-flop anterior. O contador diz-se **assíncrono** porque os flip-flops que o constituem não transitam em simultâneo.
- Cada bit do contador muda se o bit Q precedente mudar de 1 para 0 – e QN mudar de 0 para 1 .
- Há um tempo de trânsito cumulativo ao longo dos sucessivos flip-flops – as inversões não são simultâneas, são decaladas, e o período do sinal de CLK não pode ser inferior ao tempo cumulativo de propagação dos vários flip-flops.
- Este contador tem a desvantagem da limitação da frequência do CLK: para uma frequência de relógio demasiado elevada, o circuito deixa de funcionar correctamente por dependência dos **atrasos de propagação** do sinal.

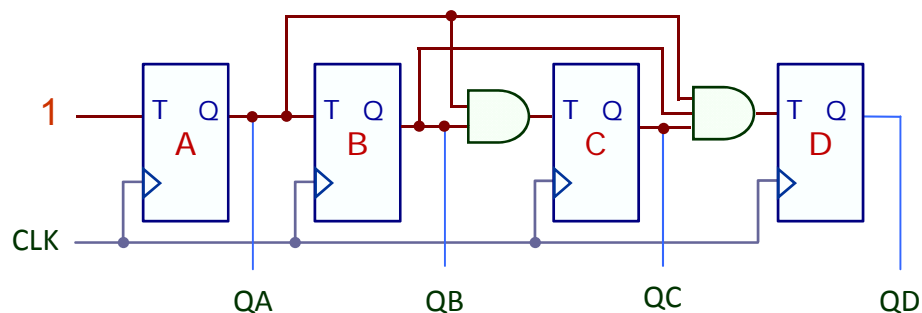
	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1 → 1	0	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1 → 1	1 → 1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

Tabela de evolução de estados em contagem crescente.

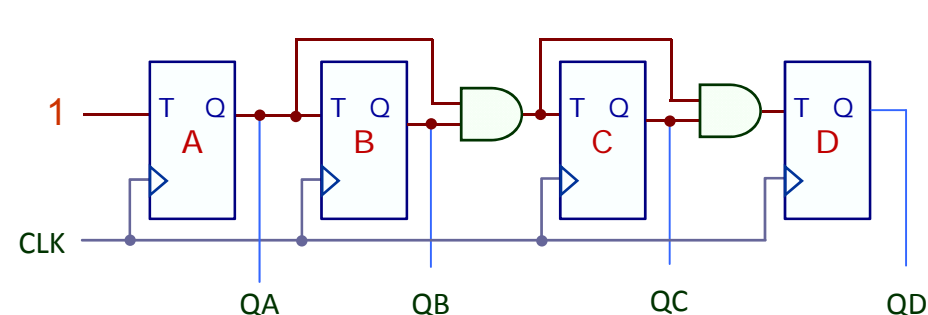
O termo **ripple** é empregue no sentido de propagação: a informação de **carry** 'propaga-se' do LSB para o MSB, um bit de cada vez.



O Contador diz-se **síncrono** porque todos os flip-flops que o constituem (no exemplo flip-flops T) estão ligados ao mesmo sinal CLK e transitam em simultâneo. Cada flip-flop T comuta se todos os bits Q de **ordem inferior** estiverem a 1. É possível construir este contador nas duas versões apresentadas, paralelo e série:



Contador binário **síncrono paralelo** sintetizado a partir de flip-flops T.



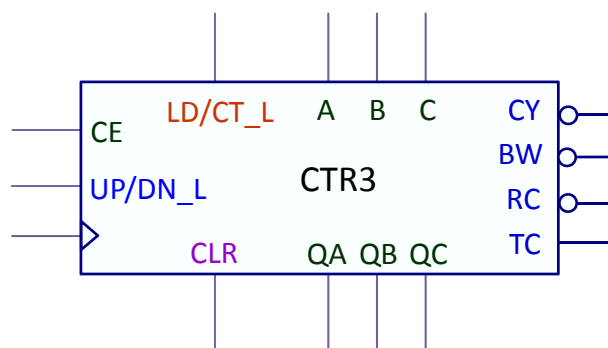
Contador binário **síncrono série** sintetizado a partir de flip-flops T.

- O flip-flop A inverte sempre.
- O flip-flop B inverte de 2 em 2 impulsos de CLK - quando o FF anterior se encontra no estado 1.
- O flip-flop C inverte de 4 em 4 - quando os 2 anteriores se encontrarem no estado 1.
- O flip-flop D inverte de 8 em 8 - quando os 3 anteriores se encontrarem no estado 1 – e assim sucessivamente.
- O comportamento descrito para 4 bits é garantido pela existência de portas AND (diferindo na forma como são concatenadas).
- A estrutura do contador de  $n=4$  bits é facilmente generalizável para contadores módulo  $2^n$ .

Na versão à esquerda o contador diz-se **síncrono paralelo** – permite frequências mais elevadas e menor tempo de trânsito porque utiliza um único nível de portas AND: cada entrada T é gerada por uma porta AND própria sem recurso a portas de ordem inferior. A limitação advém do facto de o **fan-in** das portas AND ir aumentando sucessivamente até à última porta, que terá  $n-1$  entradas se o contador tiver  $n$  bits.

Na versão à direita o contador diz-se **síncrono série** – aproveitando os produtos parciais já realizados, é possível modificar a estrutura do contador anterior para serem usadas apenas portas AND de 2 entradas, mantendo a funcionalidade.

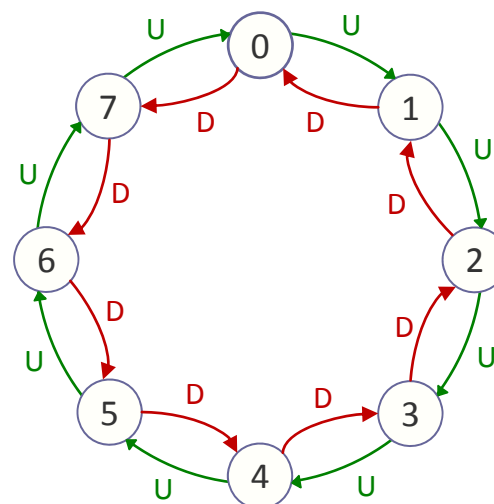
Pretende-se realizar um contador binário síncrono de 3 bits, com contagem crescente e decrescente (**UP/DOWN**), com inibição de contagem através de um sinal **CE (COUNT ENABLE)**, suportando as acções **LOAD assíncrono (LD)** e **CLEAR assíncrono (CLR)**. O contador é baseado num registo formado por 3 flip-flops T (em vez de 4 como no slide anterior) e possui as entradas e saídas descritas no símbolo lógico. A sua construção será feita por fases para melhor compreensão da estrutura interna.



Símbolo lógico do contador.

## ENTRADAS

- **CLK:** Clock
- **UP/DN\_L:** COUNT-UP com 1, COUNT-DOWN com 0.
- **LD/CT\_L:** LOAD (**assíncrono**) com 1, COUNT com 0.
- **CE:** COUNT ENABLE, torna efectivo ou inibe o CLK.
- **CLR:** CLEAR (ou MASTER RESET **assíncrono**).
- **A, B, C:** Entradas de informação paralelo.



Estrutura genérica do diagrama de estados de cada ciclo.

U=UP  
D=DOWN.

## SAÍDAS

- **CY\_L:** CARRY, associado a contagem crescente, assume o valor 0 na contagem 7 se em UP e CLK=0.
- **BW\_L:** BORROW, associado a contagem decrescente, assume o valor 0 na contagem 0 se em DN e CLK=0.
- **RC\_L:** RIPPLE Carry, assume o valor na contagem 7 ou na contagem 0, se CLK=0 e se o contador estiver Enabled (CE=1).
- **TC:** TERMINAL COUNT, também designado de **MAX/MIN**, activo na contagem 7 se em UP, ou na contagem 0 se em DOWN, e se CE=1.
- **QA, QB, QC:** saídas de informação paralelo correspondentes ao estado do contador.



## 1.ª FASE

Contador **UP** paralelo com inibição de contagem através de CE (Count Enable).

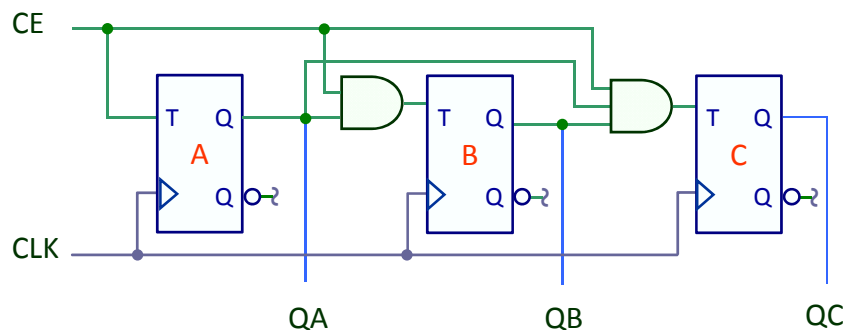


Diagrama lógico para o contador de contagem crescente.

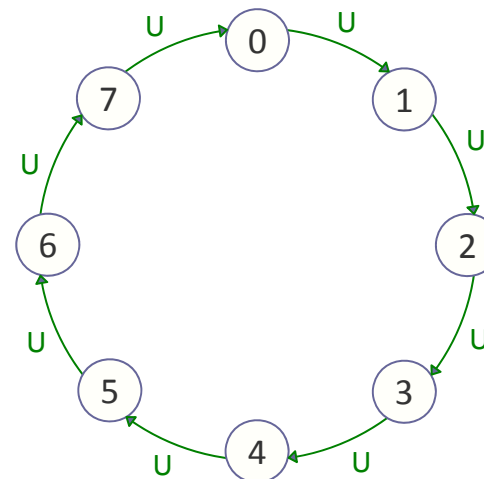


Diagrama de estados para o contador crescente.

## 2.ª FASE

Contador **DOWN** paralelo com inibição de contagem através de CE (Count Enable).

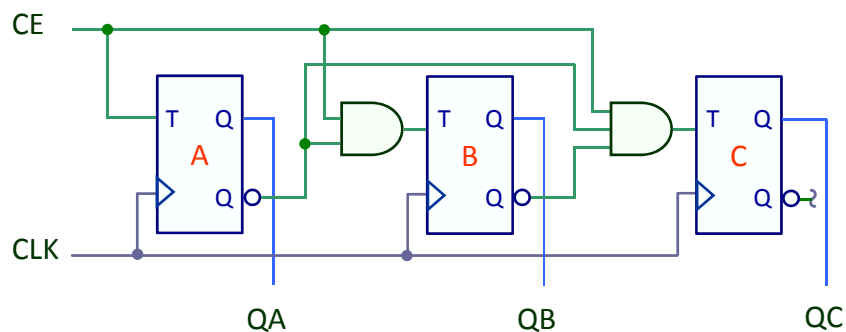


Diagrama lógico para o contador de contagem decrescente.

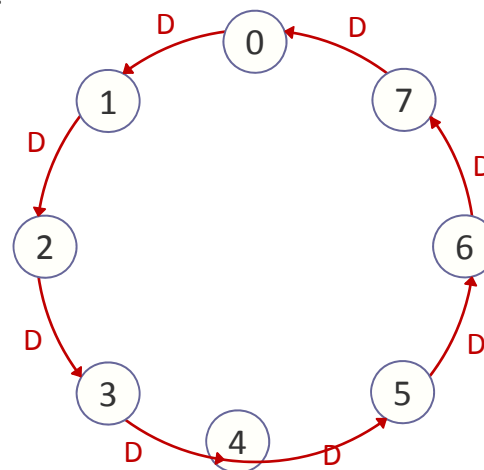
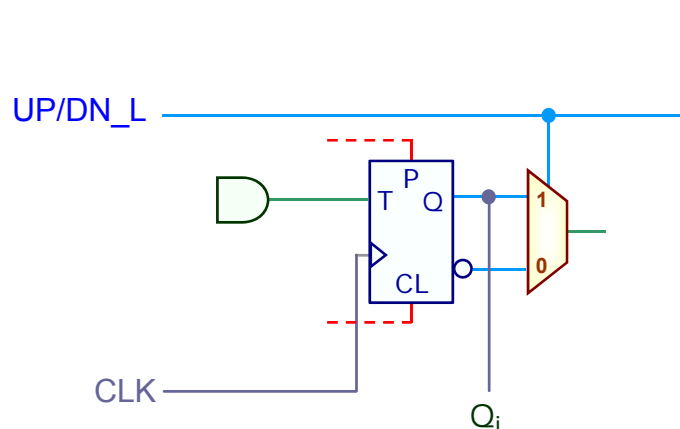


Diagrama de estados para o contador decrescente.

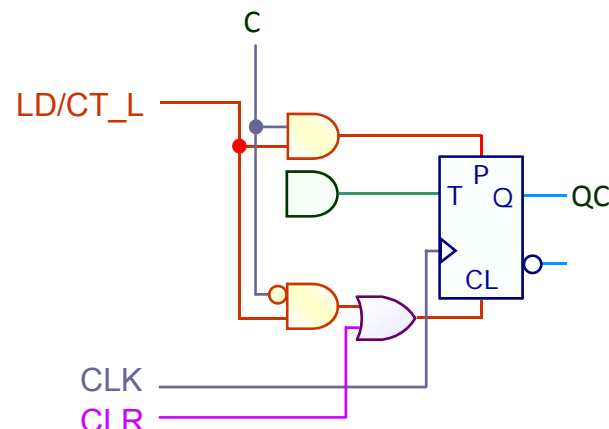


## 3.ª FASE (Fig. da esquerda)

Adição da acção **UP/DOWN** conseguida à custa de um MUX 2x1 comandado por um bit de controlo **UP/DN\_L** que escolhe as saídas Q (UP) ou QN (DOWN) de cada FF-T do seguinte modo:



Adição de um Multiplexer para a conjugação dos modos de contagem UP e DOWN.



Adição de lógica adicional para as acções de LOAD e CLEAR.

**LOAD** (de C em QD):

Se  $CLR = 0$  e  $LD/CT\_L = 1$ :

$P = C$ ,  $CL = C'$  logo  $QC = C$ .

## 4.ª FASE (Fig. da direita exemplificada para a variável de entrada C e saída QC)

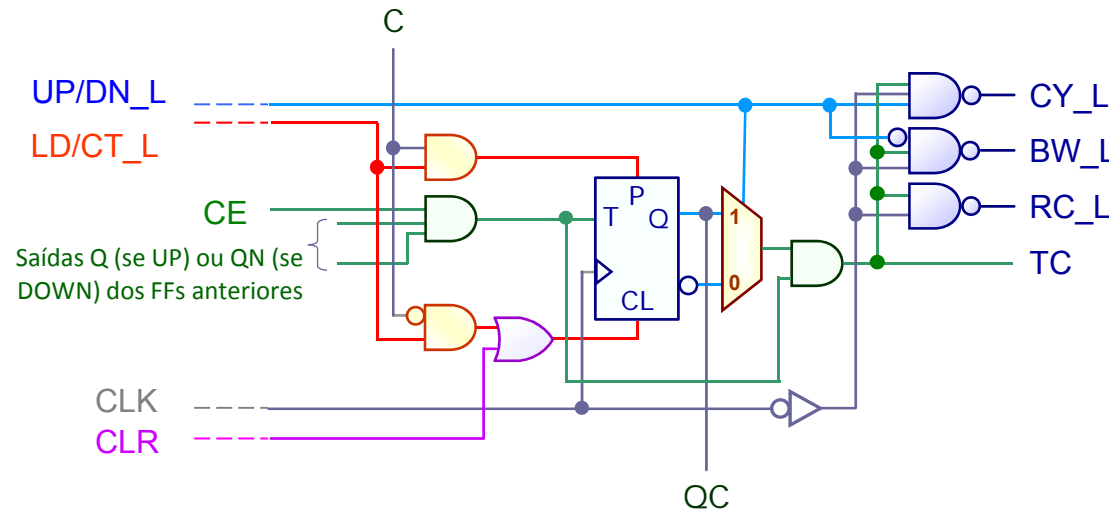
Adição das acções **LOAD** e **CLEAR** (prioritário em relação ao **LOAD**) **assíncronas**, conseguidas através da manipulação das entradas assíncronas Preset (P) e Clear (CL) de cada FF-T por meio dos sinais de controlo **LD/CT\_L** e **CLR** e lógica adicional como indicado.

A quantidade lógica por bit num contador deste tipo é fixa e corresponde a:

1 flip-flop T, 3 portas AND, 1 porta OR, 1 porta NOT e 1 MUX 2x1 - não está contabilizada a lógica geradora das saídas de controle – slide seguinte.

## 4.ª FASE

Adição da lógica geradora das saídas de controle.



Estrutura interna do último andar do contador.

$$CY\_L = \{ QA \cdot QB \cdot QC \cdot CE \cdot CLK' \cdot UP/DN\_L \}'$$

$$BW\_L = \{ QA' \cdot QB' \cdot QC' \cdot CE \cdot CLK' \cdot (UP/DN\_L)' \}'$$

$$TC = [(QA \cdot QB \cdot QC) + (QA' \cdot QB' \cdot QC')] \cdot CE$$

$$RC\_L = \{ TC \cdot CLK' \}'$$

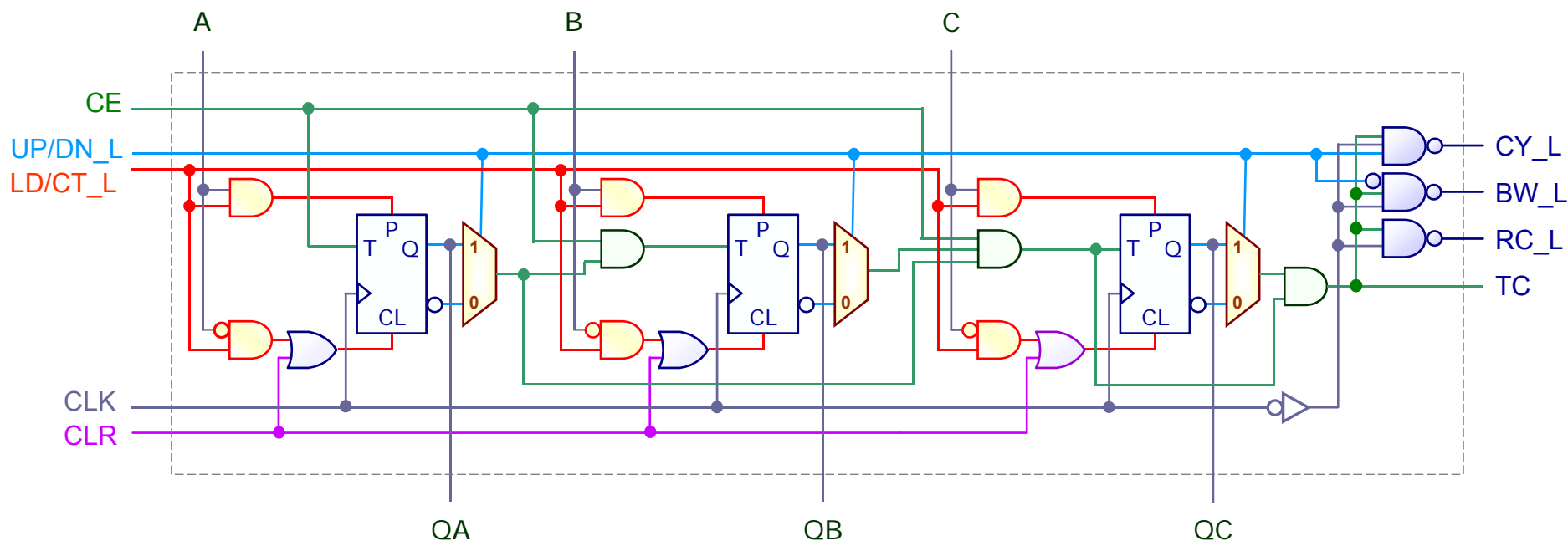
Equações lógicas descritivas das saídas do contador.

A saída **TC** (**TERMINAL COUNT**) também designada de **Max/Min** indica, quando activa, que o contador atingiu um dos extremos de contagem:

- 7 na contagem crescente,
- 0 na contagem decrescente.

É utilizada para ligação à entrada **CE** (**COUNT ENABLE**) de outro módulo para concatenação **síncrona** em cascata.

A saída **RC\_L** (**RIPPLE CARRY**) corresponde à intersecção de TC com CLK'. Utilizada para propagação do CLK ao contador de peso seguinte em concatenação **assíncrona**.



Estrutura interna do contador de 3 bits com as funcionalidades estabelecidas.

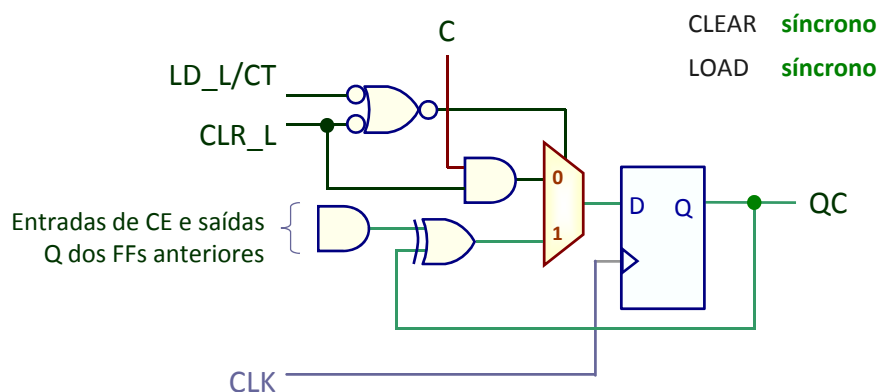
## ENTRADAS

- **CLK:** Clock
- **UP/DN\_L:** COUNT-UP com 1, COUNT-DOWN com 0.
- **LD/CT\_L:** LOAD (**assíncrono**) com 1, COUNT com 0.
- **CE:** COUNT ENABLE, torna efectivo ou inibe o CLK.
- **CLR:** CLEAR (ou MASTER RESET **assíncrono**).
- **A, B, C:** Entradas de informação paralelo.

## SAÍDAS

- **CY\_L:** CARRY, associado a contagem crescente, assume o valor 0 na contagem 7 se em UP e CLK=0.
- **BW\_L:** BORROW, associado a contagem decrescente, assume o valor 0 na contagem 0 se em DN e CLK=0.
- **RC\_L:** RIPPLE Carry, assume o valor na contagem 7 ou na contagem 0, se CLK=0 e se o contador estiver Enabled (CE=1).
- **TC:** TERMINAL COUNT, também designado de **Max/Min**, activo na contagem 7 se em Up, ou na contagem 0 se em Down, e se CE=1.
- **QA, QB, QC:** saídas de informação paralelo correspondentes ao estado do contador.

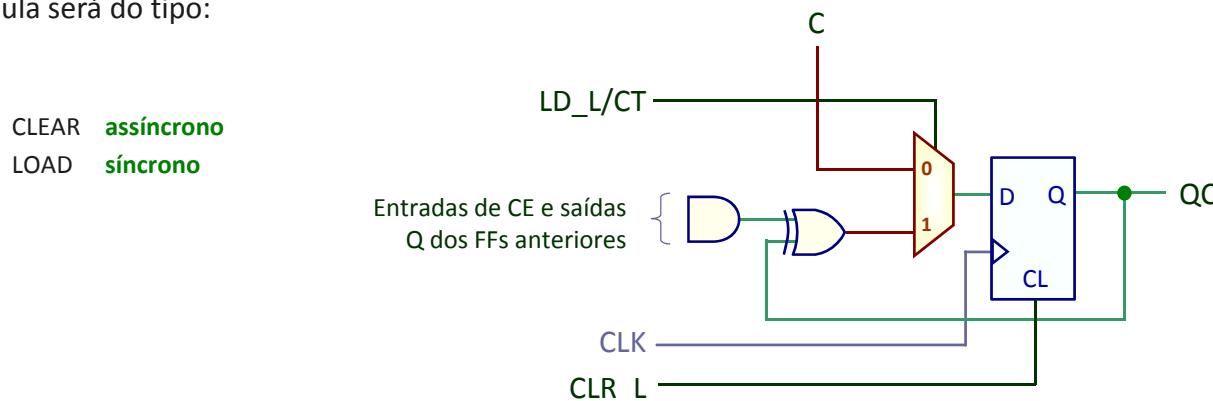
Há módulos MSI, como o contador 74 x 163 (descrito adiante), em que as acções CLEAR e LOAD são ambas **síncronas**. O 74 x 163 utiliza internamente FF D em vez de FF T, para facilitar essas acções. A estrutura típica de uma célula de um bit de um contador semelhante ao 74 x 163 será do tipo da Fig. :



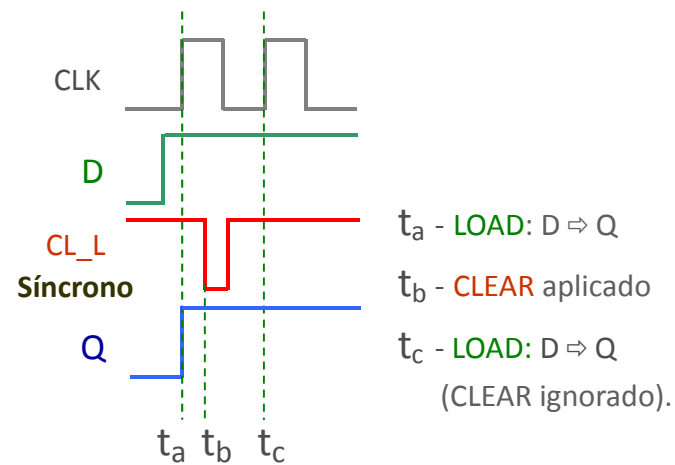
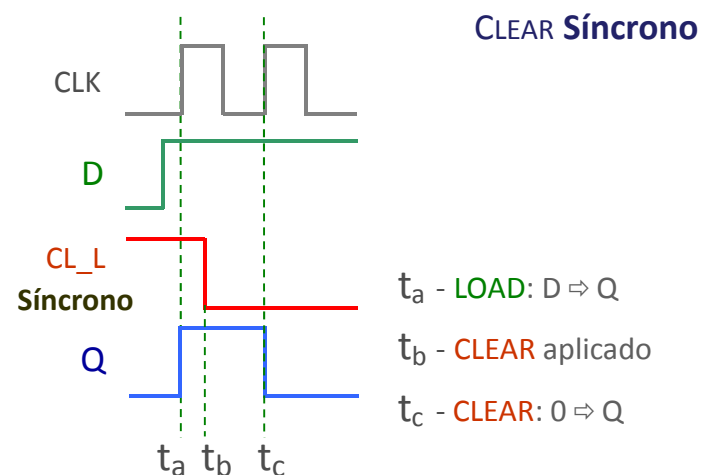
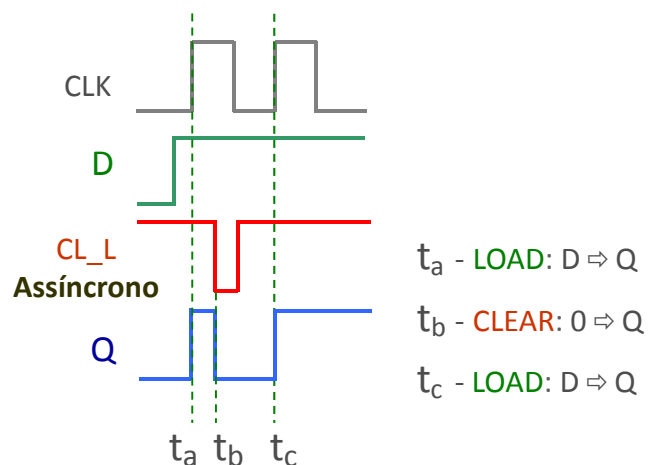
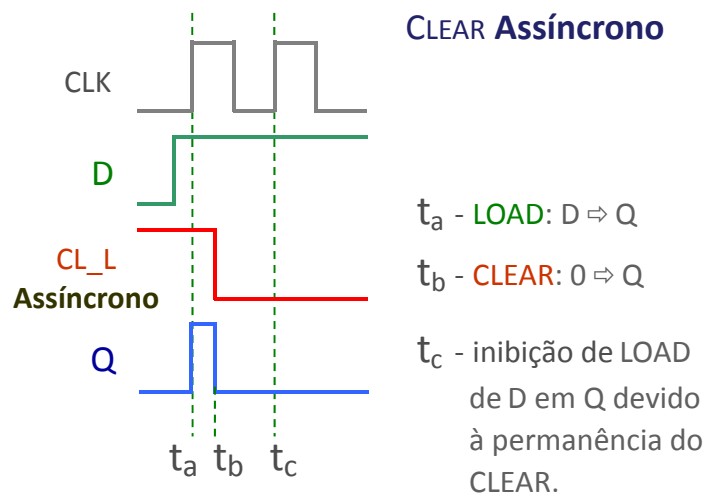
QC é a saída do contador para a célula exemplificada, e C a entrada para LOAD.

A porta XOR, essencial à lógica de concatenação, realiza a função de contagem, uma vez que se recorre a flip-flop D e não a flip-flop T. O contador 74 x 163 só conta em modo UP, pelo que da célula não consta o MUX de saída do contador anterior ligado à selecção UP/DOWN.

Há aplicações que requerem uma entrada de CLEAR **assíncrona**, e contadores MSI que a possuem. O circuito contador 74 x 161 é em tudo semelhante ao circuito 74 x 163, e só difere na acção CLEAR, que é feita em modo assíncrono. Esta acção será suportada pelo próprio flip-flop D e a estrutura da célula será do tipo:



Uma operação CLEAR assíncrona é realizada imediatamente após a aplicação da entrada, é independente do CLK, e a acção é realizada durante todo o tempo em que a entrada se mantiver activa. A operação síncrona só se realiza na transição de CLK se essa entrada estiver activa. Exemplifica-se a diferença:



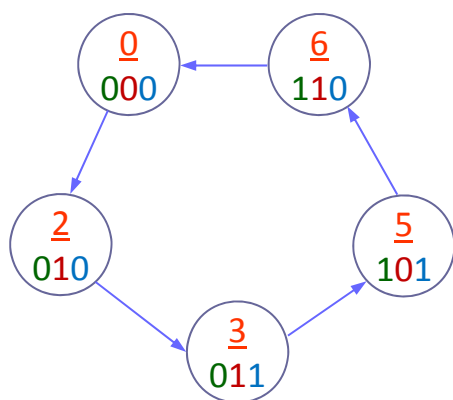


Diagrama de transição de estados.

	ESTADO ACTUAL			ESTADO SEGUINTE		
	QC	QB	QA	QC <sup>+</sup>	QB <sup>+</sup>	QA <sup>+</sup>
0	0	0	0	0	1	0
1	0	0	1	—	—	—
2	0	1	0	0	1	1
3	0	1	1	1	0	1
4	1	0	0	—	—	—
5	1	0	1	1	1	0
6	1	1	0	0	0	0
7	1	1	1	—	—	—

Tabela de transição de estados assinalando os estados inválidos a (cor).

O contador repete **5 estados** numa sequência que não é a do código binário puro.

- A existência de 5 estados de contagem impõe, pelo menos, a utilização de 3 flip-flops.
- O processo de síntese passa pela definição da tabela de transição de estados do circuito, derivada do diagrama de transição de estados, e prossegue com a determinação das entradas dos flip-flops como função do estado anterior, através da construção de mapas de Karnaugh.
- Como não são utilizados todos os estados disponíveis, pode verificar-se a ocorrência de **estados inválidos** - fora da sequência de contagem – que podem acontecer devido a ruído eléctrico no circuito, a valores da fonte de alimentação fora da gama especificada, ou à não imposição de estado inicial no arranque (**power-up**).
- Se o contador se encontrar num estado inválido pode entrar na sequência de contagem pretendida ou ficar indefinidamente fora dela (Locked-out).
- Um contador **AUTO-CORRECTOR (SELF-CORRECTING)** impõe a transição de qualquer estado inválido para um estado dentro da sequência de contagem explorando a existência de **don't cares** (indiferenças).
- Este contador será auto-corrector. A escolha das indiferenças nos mapas de Karnaugh do próximo slide conduzem a uma minimização do circuito. Levam a que a sequência de contagem para cada um dos estados inválidos 1, 4 e 7, até à entrada no ciclo, seja:  

$$1 \rightarrow 6 ; 4 \rightarrow 2 ; 7 \rightarrow 4 \rightarrow 2$$
- Está garantida a entrada na sequência especificada ao fim de um máximo de 2 impulsos de relógio: o caso mais desfavorável é o caso do estado 7, que demora 2 impulsos, enquanto que os restantes demoram 1.

MAE

	$Q_A(1)$			
	0	2	3	-
$Q_C(4)$	-	6	-	5
	$Q_B(2)$			

Mapa de atribuição de estados.

$Q_A^+$

	$Q_A$			
	0	1	1	-
$Q_C$	-	0	-	0
	$Q_B$			

$$Q_A^+ = Q_B Q_C'$$

$Q_B^+$

	$Q_A$			
	1	1	0	-
$Q_C$	-	0	-	1
	$Q_B$			

$$Q_B^+ = Q_B' + Q_A' Q_C'$$

$Q_C^+$

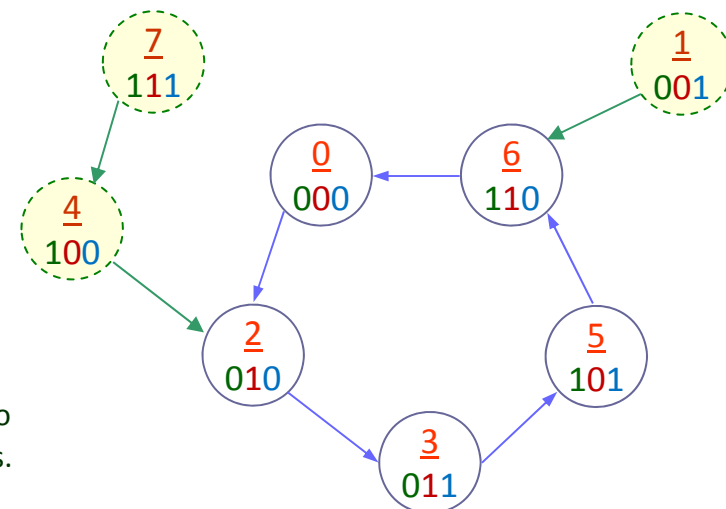
	$Q_A$			
	0	0	1	-
$Q_C$	-	0	-	1
	$Q_B$			

$$Q_C^+ = Q_A$$

Mapas e equações lógicas do ESTADO SEGUINTE de cada flip-flop.

	ESTADO ACTUAL			ESTADO SEGUINTE		
	QC	QB	QA	QC <sup>+</sup>	QB <sup>+</sup>	QA <sup>+</sup>
0	0	0	0	0	1	0
1	0	0	1	1	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	1
4	1	0	0	0	1	0
5	1	0	1	1	1	0
6	1	1	0	0	0	0
7	1	1	1	1	0	0

Tabela de TRANSIÇÃO DE ESTADOS depois da atribuição das indiferenças.



Os estados a amarelo são estados inválidos.

Diagrama de TRANSIÇÃO de ESTADOS mostrando a natureza auto-correctora do circuito.

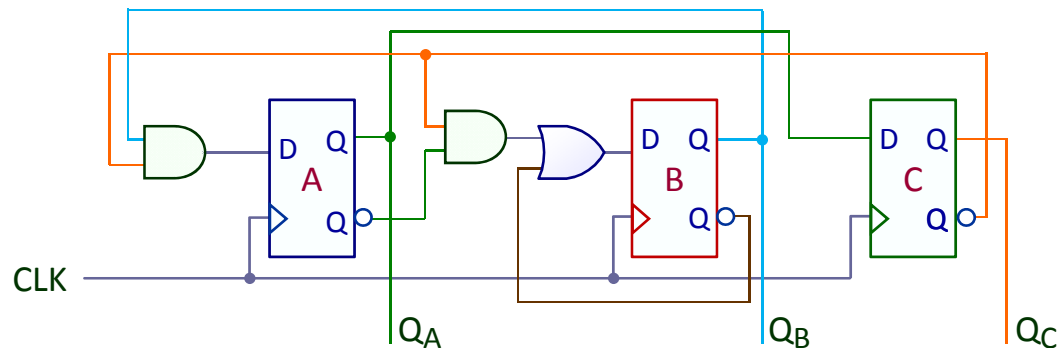


Diagrama lógico do circuito CONTADOR MÓDULO-5 evoluindo na sequência 0-2-3-5-6-0 e implementado com flip-flops D.

$$D_A = Q_A^+ = Q_B Q_C'$$

$$D_B = Q_B^+ = Q_B' + Q_A' Q_C'$$

$$D_C = Q_C^+ = Q_A$$

Equações lógicas do ESTADO SEGUINTE.

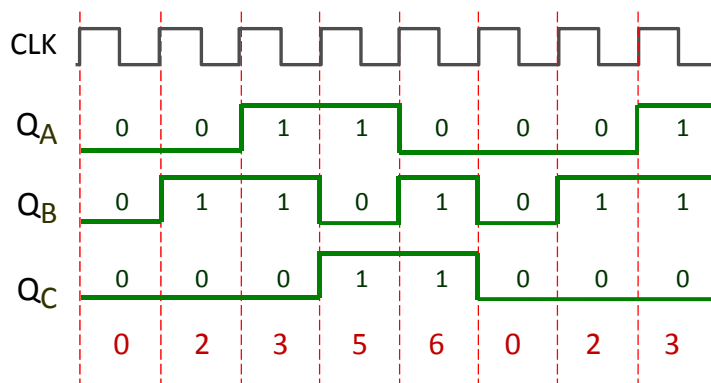


Diagrama temporal de formas de onda nas saídas QA, QB e QC do Contador.

Num projecto com flip-flops do tipo D, as entradas D serão iguais aos valores do ESTADO SEGUINTE  $Q^+$  para cada bit.



## ENTRADAS

- **CU:** **Count-Up**, clock da contagem crescente.
- **CD:** **Count-Down**, clock da contagem decrescente.
- **CLR:** **Clear** ou **Master Reset assíncrono**.
- **LD\_L:** **Load assíncrono**.
- **A, B, C, D:** entradas de dados em paralelo (**Parallel Inputs**).

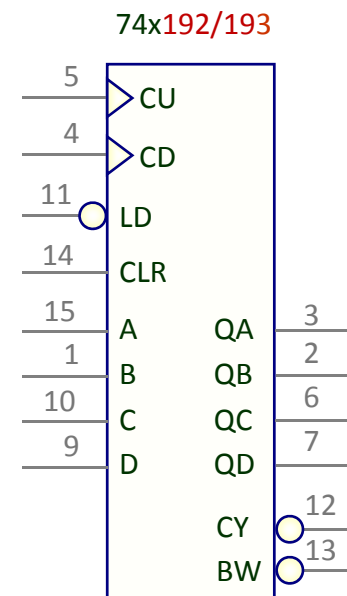
## SAÍDAS

- **CY\_L:** **Carry**, associado a contagem crescente.
- **BW\_L:** **Borrow**, associado a contagem decrescente.
- **Q<sub>A</sub>, Q<sub>B</sub>, Q<sub>C</sub>, Q<sub>D</sub>:** saída de dados em paralelo (**Parallel Outputs**).

$$CY\_L = (QA \cdot QB \cdot QC \cdot QD \cdot CU')'$$

$$BW\_L = (QA' \cdot QB' \cdot QC' \cdot QD' \cdot CD')'$$

Equações lógicas descritivas das  
saídas de controle do contador.



Configuração dos  
pinos dos circuitos  
74x192 e 74x193.

Os contadores 74x193 e 74x192 são pino-a-pino compatíveis e diferem entre si no módulo de contagem – **binário** no contador 193, **BCD** (decade) no 192.

## 74x163 – SYNCHRONOUS PRESETTABLE 4-BIT BINARY COUNTER

8-42

### ENTRADAS

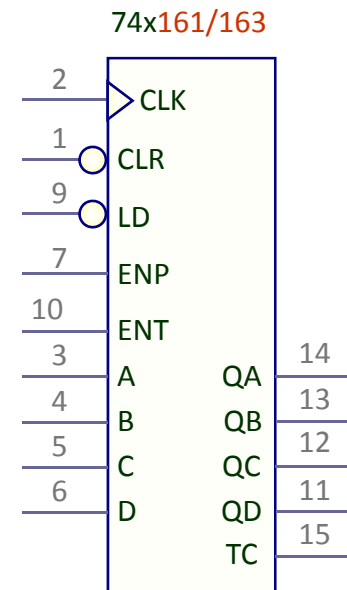
- **CLK:** Clock .
- **ENP:** Enable Count Parallel.
- **ENT:** Enable Count **Trickle**.
- **CLR\_L:** Clear ou Master Reset **síncrono**.
- **LD\_L:** Parallel Load **síncrono**.
- **A, B, C, D:** Entradas de dados em paralelo (**Parallel Inputs**).

### Saídas

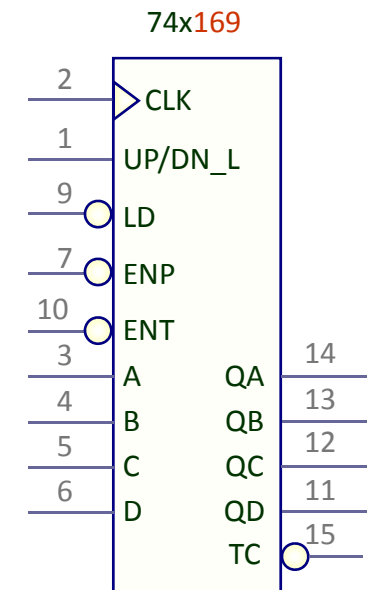
- **TC:** **Terminal Count**, também designado por alguns fabricantes por **RCO – Ripple Carry Output**
- **QA, QB, QC, QD:** Saídas de dados em paralelo (**Parallel Outputs**).

O contador fica **enabled** sse ENP e ENT estiverem ambos a 1. TC fica a 1 quando ENT=1 e o contador atingir o último estado de contagem (1111) – este contador só conta em modo crescente.

Ligar a saída TC de um contador à entrada ENT de um segundo permite incrementar este segundo de uma unidade sempre que o primeiro passe do último para o primeiro estado (roll over).



Configuração dos pinos dos circuitos 74x161 e 74x163.



Configuração dos pinos do circuito 74x169.

Os contadores 74x163 e 74x161 são compatíveis pino-a-pino. Diferem entre si no sincronismo das entradas LD\_L e CLR\_L (**assíncronas** no 74x161, **síncronas** no 74x163).

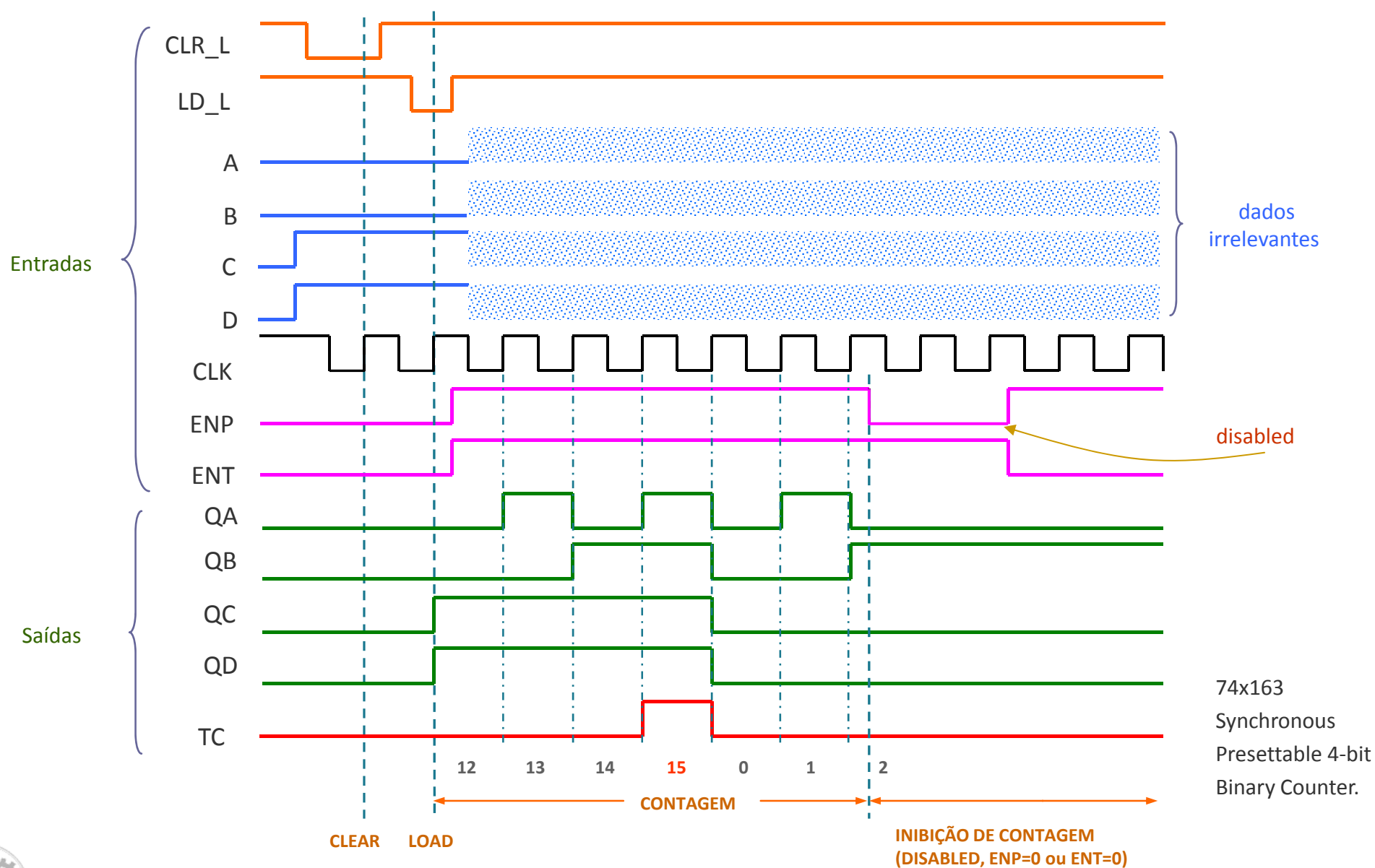
O contador 74x169 possui uma lógica interna de geração do próximo estado mais desenvolvida que lhe permite actuar em modo crescente/decrescente. O pino 1 em vez de CLEAR assume agora a função de controle de direcção Up/Down, e os sinais de controle ENP, ENT e TC (RCO) são **active-low**. A lógica da saída **TC\_L** para o contador x169 será:

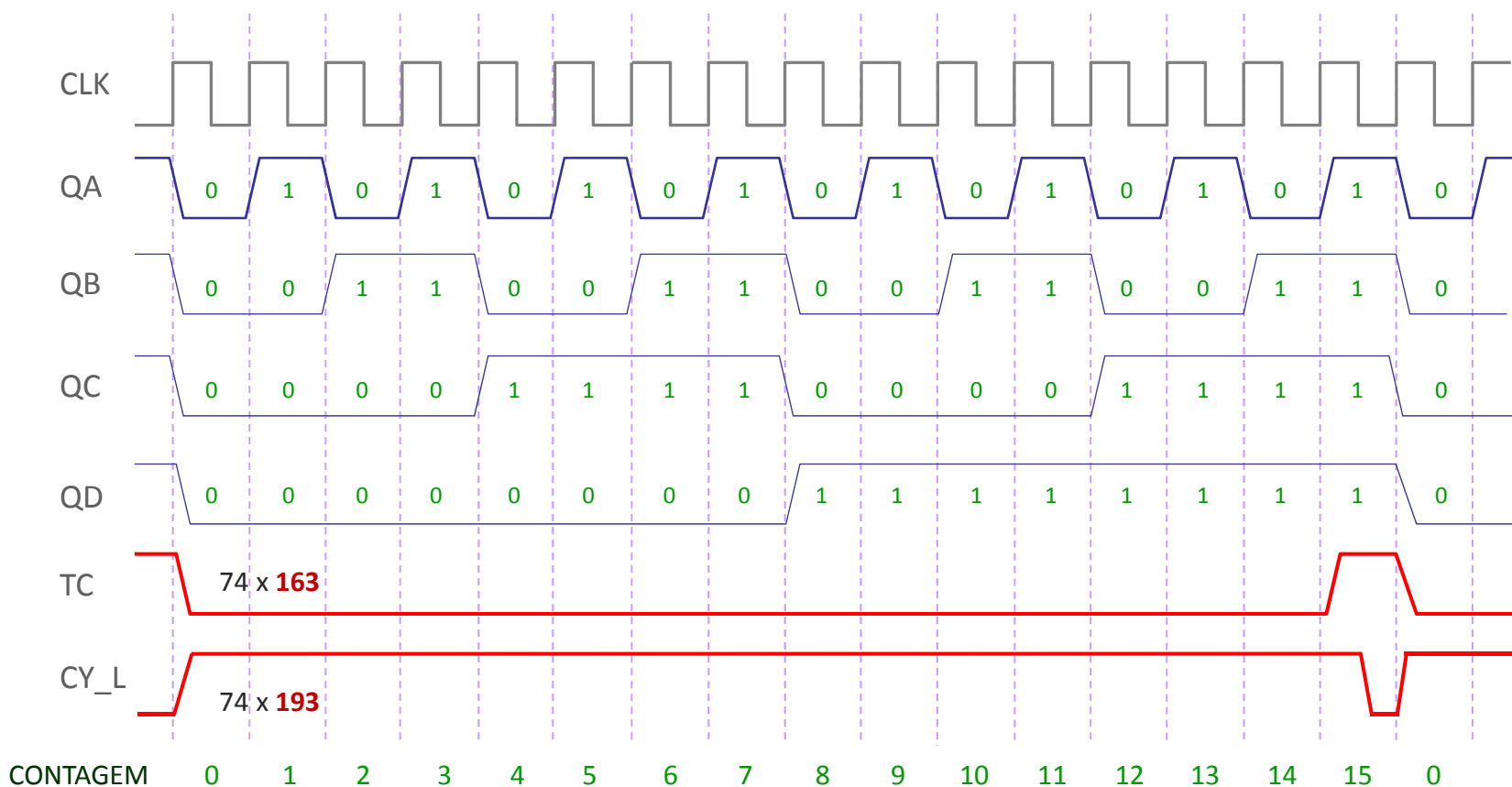
$$TC\_L = (QA \cdot QB \cdot QC \cdot QD \cdot (UP/DN\_L) \cdot ENT\_L' + QA' \cdot QB' \cdot QC' \cdot QD' \cdot (UP/DN\_L)' \cdot ENT\_L')'$$



# DIAGRAMA TEMPORAL OBSERVADO NAS ENTRADAS E SAÍDAS DO CONTADOR 74x163

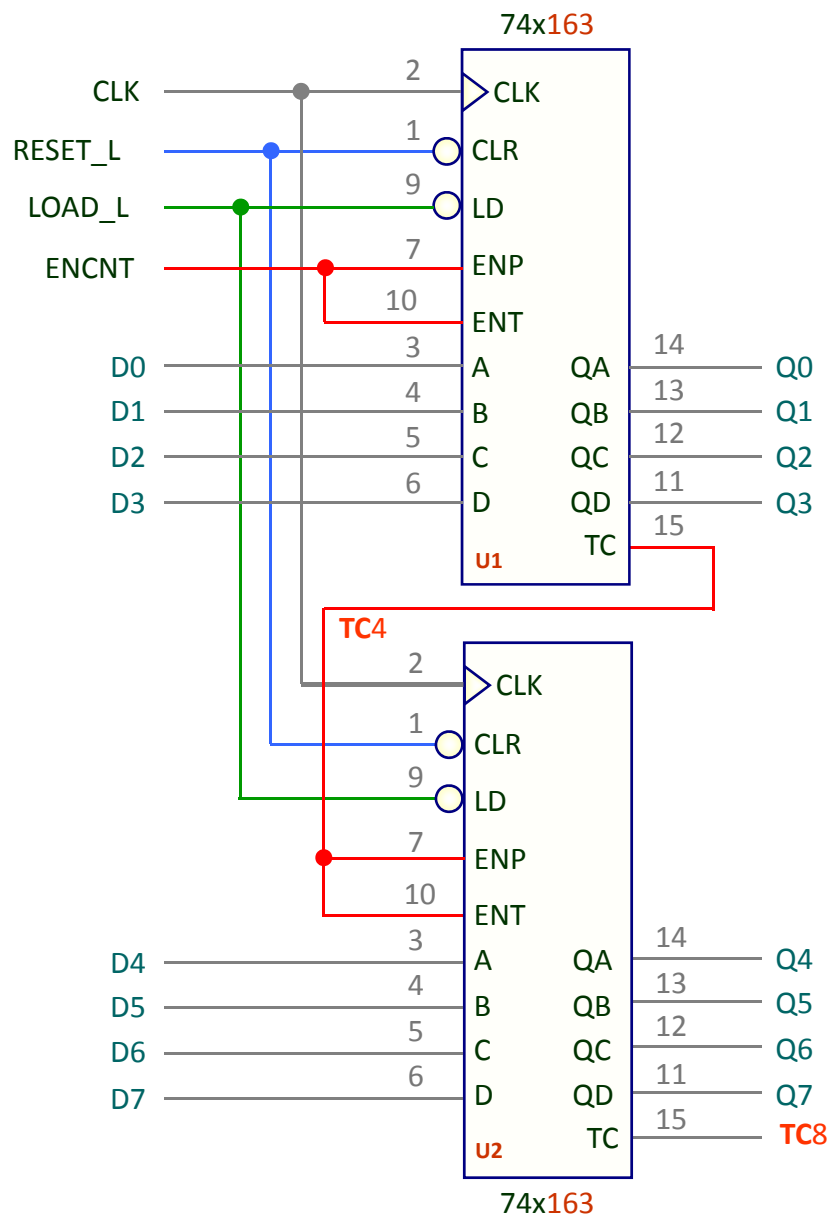
8-43





Formas de onda ilustrativas do comportamento temporal dos contadores 74x163 e 74x193.

Os contadores estão em modo **free-running**. As formas de onda das saídas são representadas com uma inclinação para relembrar que na vida real não ocorrem em tempo zero relativamente ao impulso de relógio CLK. A saída **TC** (**TERMINAL COUNT**) do circuito 74x163 detecta o estado 15, é **active-high** e tem a duração de um período inteiro de relógio, enquanto que a saída **CY\_L** (**CARRY**) do circuito 74x193 detecta também o estado 15, é **active-low** e tem a duração de meio período de relógio.



- O contador mais significativo U2 só é habilitado quando o contador U1 atinge o último estado de contagem (15). Neste estado, quando se verificar a próxima transição de CLK, U2 passará para o estado 1 e U1 passará novamente para o primeiro estado (0).
- Para manter o carácter síncrono da interligação os contadores possuem um relógio comum sendo que o Enable de contagem de U2 (ENP e ENT ambos a 1) depende da sinalização da chegada ao estado 15 de U1 através da activação da sua saída TC.

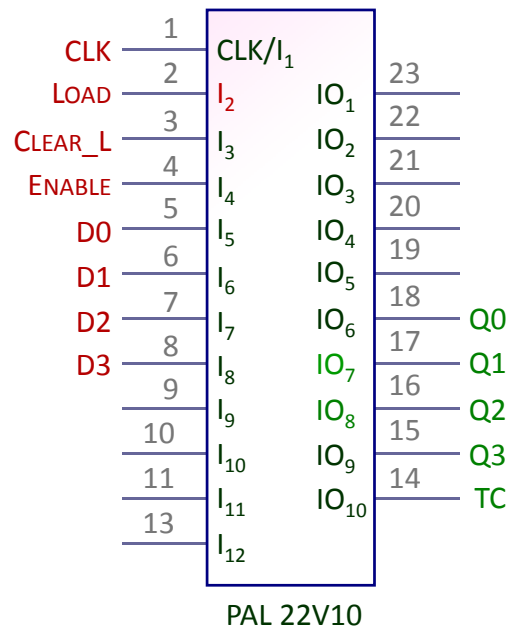
Diagrama lógico de um contador síncrono de **módulo-256** implementado por ligação síncrona de dois contadores 74 x 163 de módulo-16.

## CONTADOR DE 4 BITS EM PAL 22V10

8-46

```
Name      4-Bit Counter ;
Device    p22v10 ;
/* **** INPUT PINS ****/
PIN 1     =   clk       ;
PIN 2     =   load      ;
PIN 3     =   !clear    ;
PIN 4     =   enable    ;
PIN [5..8] = [D0..3]    ;
/* **** OUTPUT PINS *****/
PIN 14    =   tc        ;
PIN [18..15] = [Q0..3] ;
```

Troço de código  
CUPL para um  
CONTADOR  
crescente de 4 bits  
com CLEAR e LOAD  
síncronos.



Símbolo lógico da  
PAL assinalando os  
pinos de entrada e  
saída utilizados.

```
/* ***** BODY ***** */
field in = [D0..3];
field output = [Q0..3];
field mode = [!clear, load, enable];
clr = mode:[0..3];
idle = mode:4;
count = mode:5;
ld = mode:[6..7];
output.AR = 'b'0;
output.SP = 'b'0;
Sequence output{
    $repeat i=[0..15]
        present 'h'{i}
            if count next 'h'{(i+1)%16};
            if clr next 'h'{0};
            if idle next 'h'{i};
        $repend
    }
tc= output:'h'f;
append output.D = in & load ;
```

Tabela funcional e modos de funcionamento  
dependentes da configuração das entradas.

MODE	CLEAR_L (3)	LOAD (2)	ENABLE (4)	FUNÇÃO
0..3	0	—	—	CLEAR
4	1	0	0	IDLE
5	1	0	1	COUNT
6..7	1	1	—	LOAD



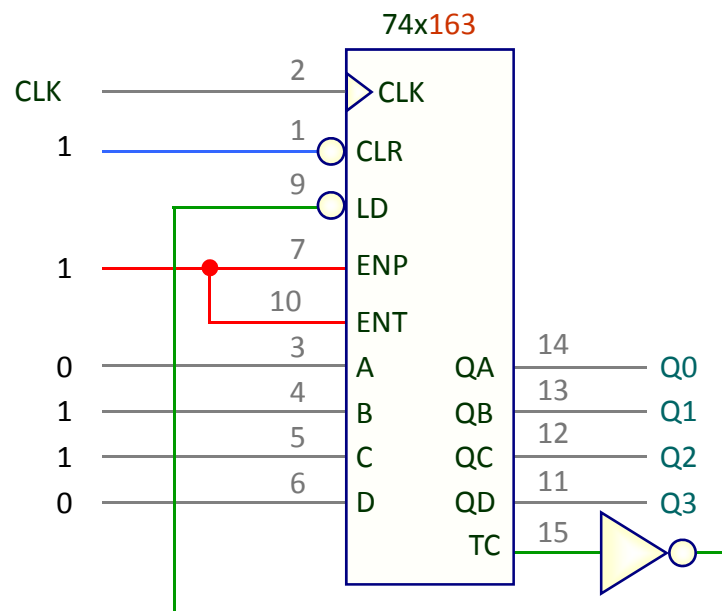


Diagrama lógico do contador módulo-10 e da lógica auxiliar.

A saída **Terminal Count** (TC) detecta o estado 15 e é usada para forçar o próximo estado a 6, recomeçando a contagem módulo-10 até 15 como patente no diagrama temporal.

A forma de onda em TC tem um décimo da frequência do sinal CLK.

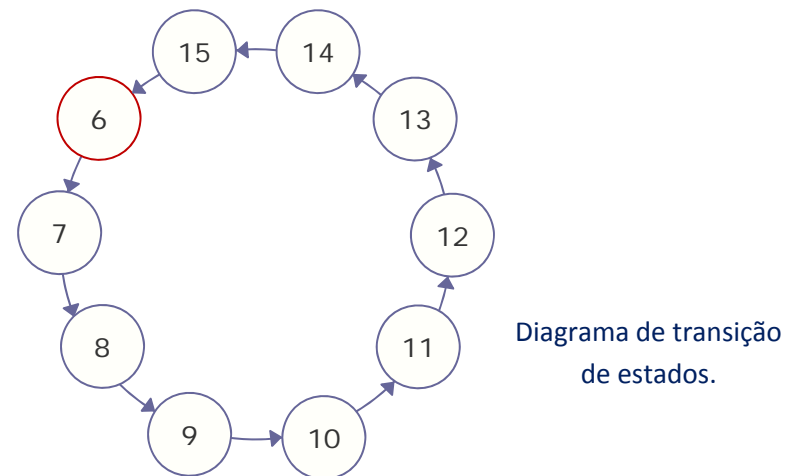


Diagrama de transição de estados.

**LOAD síncrono** – carregamento em paralelo do número 6.

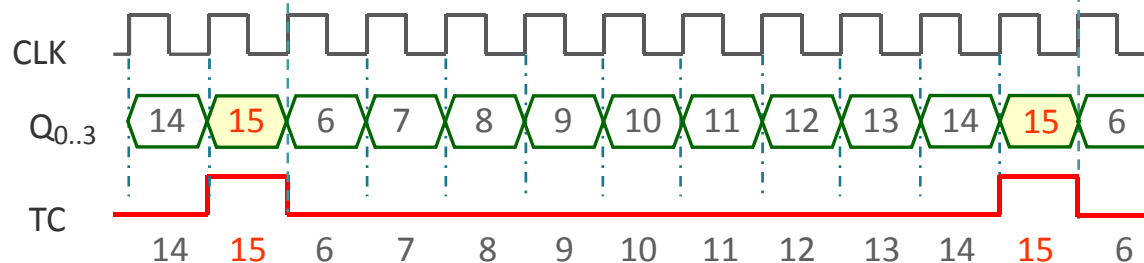


Diagrama temporal da saída TC nos vários estados do contador.

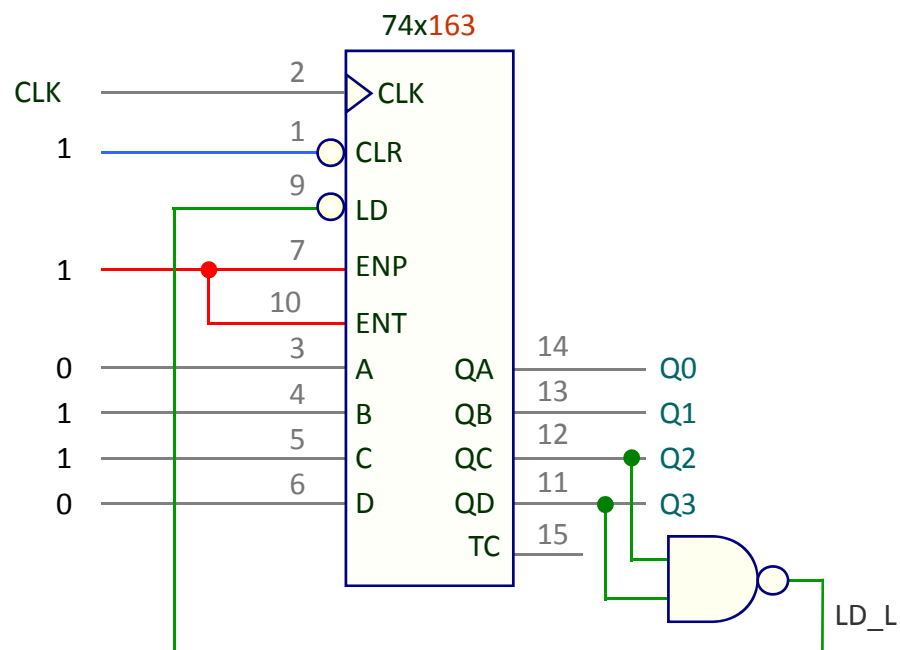


Diagrama lógico do contador e da lógica auxiliar.

A saída **terminal count** (TC) não é utilizada nesta aplicação. A detecção do estado 12 (1100) é feita com uma porta NAND de apenas 2 entradas (em vez das 4 usuais) – genericamente, para se detectar uma contagem **n** num contador binário que conta de **0** a **n** basta utilizar um AND dos bits que são 1 nessa contagem.

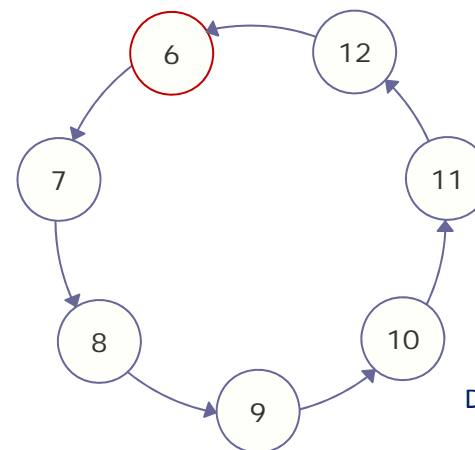


Diagrama de transição de estados.

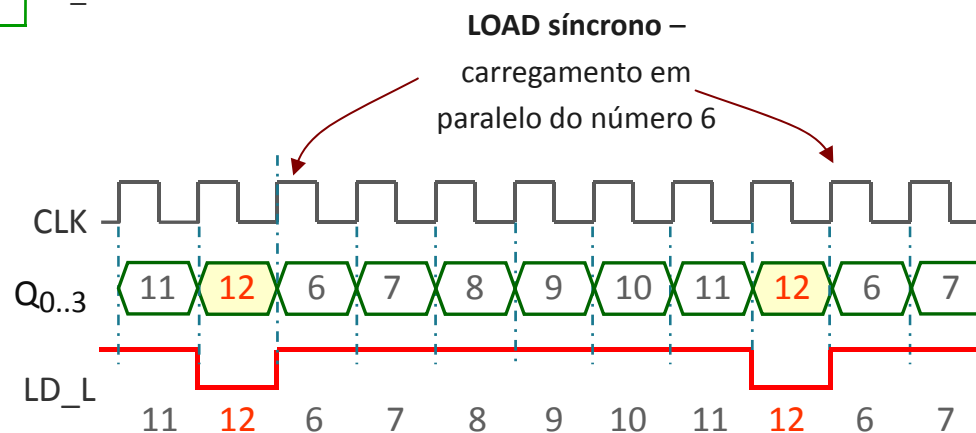


Diagrama temporal da entrada LD\_L nos vários estados do contador.



# CONTADOR COM SEQUÊNCIA DE CONTAGEM 0 A 15 SEM PASSAR PELAS CONTAGENS 4, 8 E 12

8-49

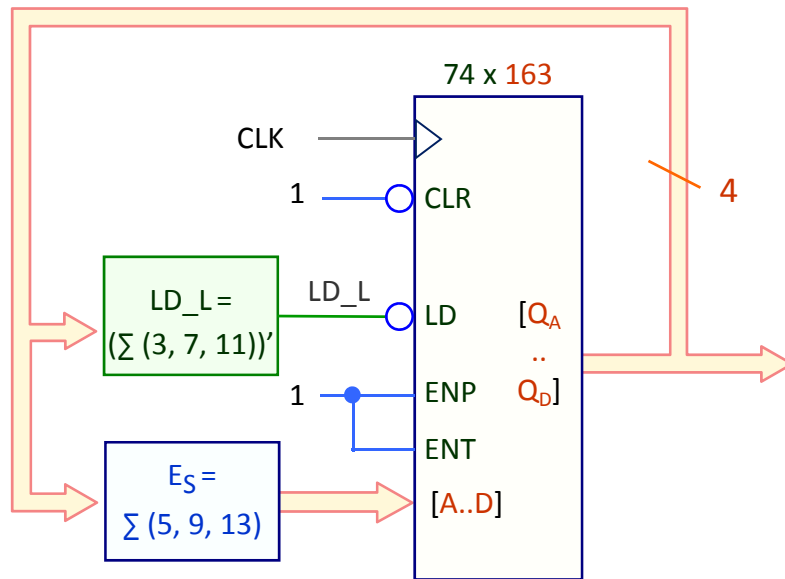


Diagrama de blocos do contador e da lógica auxiliar.

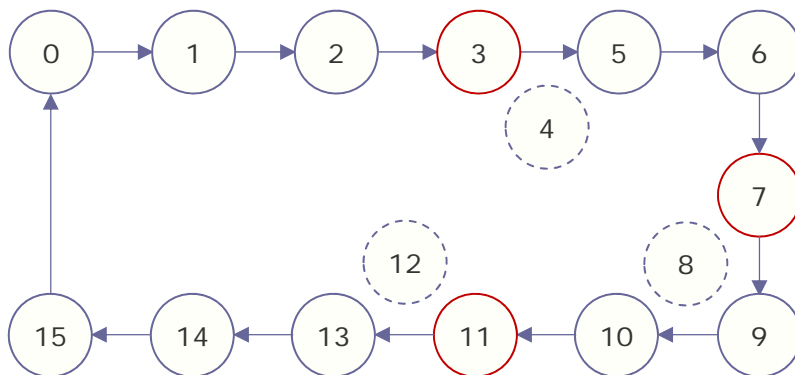


Diagrama de transição de estados evidenciando os estados 3, 7 e 11 em que se fará a acção LOAD, e a traço interrompido os estados inválidos 4, 8 e 12.

O contador está normalmente em modo de **contagem crescente**, com as entradas de Enable (ENP e ENT) activas a 1.

Há que detectar (descodificar) o estado actual ( $E_A$ ) **3, 7 e 11** para se fazer, na próxima transição de CLK, o **carregamento síncrono em paralelo** (LOAD) dos números **5, 9 e 13** respectivamente, correspondentes a cada estado seguinte ( $E_S$ ).

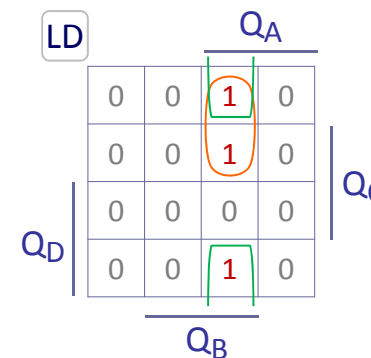
Estados $E_A$ em que LD_L está activo									
$E_A$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$E_S$	D	C	B	A
3	0	0	1	1	5	0	1	0	1
7	0	1	1	1	9	1	0	0	1
11	1	0	1	1	13	1	1	0	1

$E_A$  – Estado actual

$E_S$  – Estado seguinte

Na acção LOAD, A e B tomarão sempre os valores 0 e 1 respectivamente.

Tabela de transição de estados (elaborada para os estados actuais 3, 7, 11).



$$LD = \Sigma(3, 7, 11) = Q_A Q_B (Q_C' + Q_D')$$

Mapa de Karnaugh e equação lógica do sinal LD (LOAD) interpretado como **active-high**.



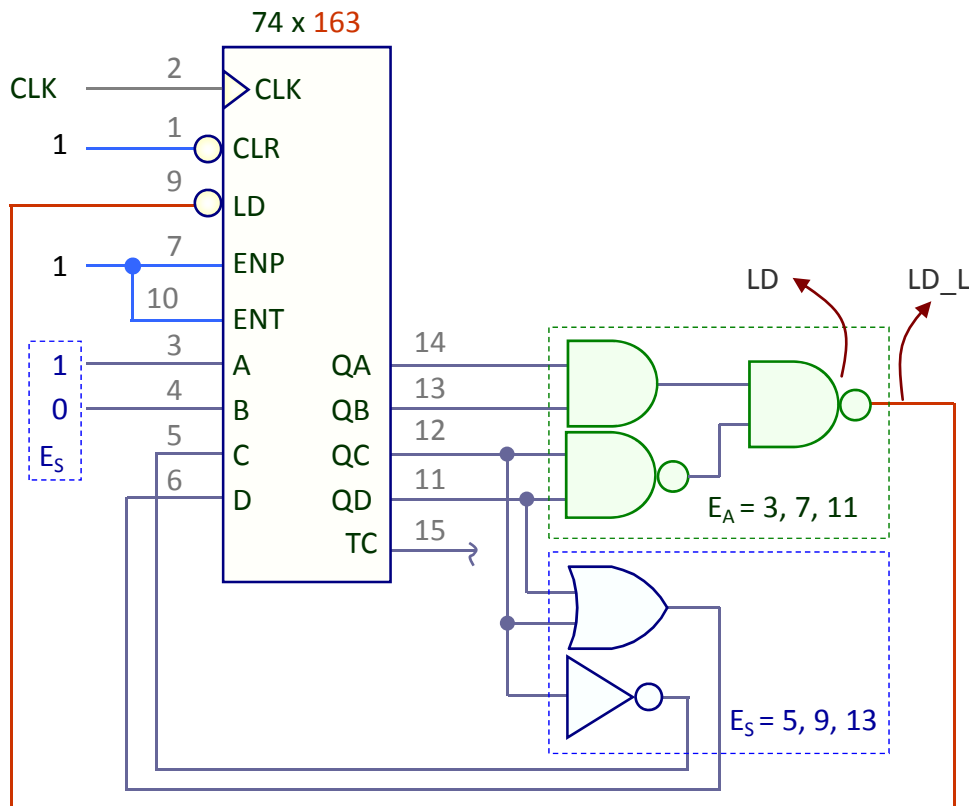
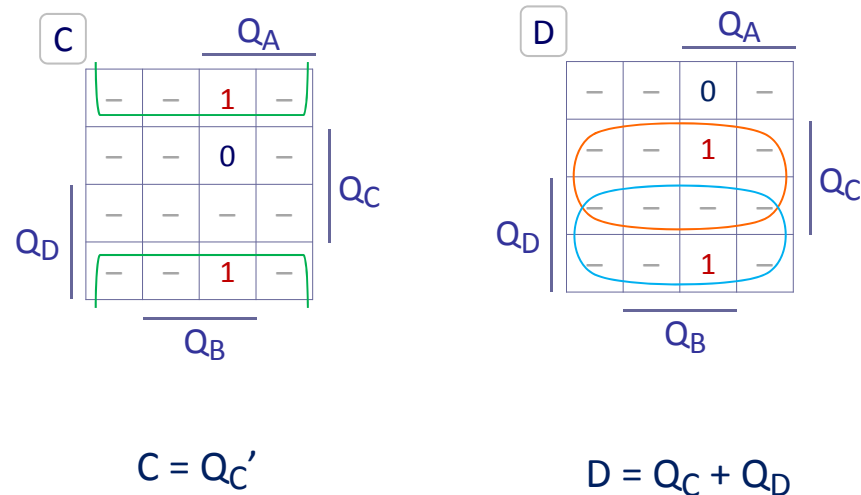


Diagrama lógico do contador evidenciando os circuitos  $E_A$  e  $E_S$  como denotados no diagrama de blocos.



Mapas de Karnaugh e equações das entradas C e D para a acção de LOAD. Os valores das entradas A e B são fixos (1 e 0 respectivamente) como se verifica pela tabela do slide anterior.

$$LD = \Sigma (3, 7, 11) = Q_A Q_B (Q_C' + Q_D')$$

O sinal LD da equação lógica em cima é invertido na implementação final por ser de natureza **active-low** ( $LD\_L$ ).

1. **LSD-8 – CIRCUITOS SEQUENCIAIS**
2. Registo Simples de Carregamento Paralelo
3. 74x175 – Quad D-type Positive Edge-triggered Flip-flop with Clear
4. Registo de Carregamento Paralelo com Saídas Tri-state
5. Diagramas Temporais para a Operação de Escrita em Latch e em Flip-flop
6. Registos com Barramento de Dados Bidireccional
7. RAM Estática – Estrutura Interna
8. RAM Estática com Barramentos de Dados Unidireccional e Bidireccional
9. Comunicação de Dados entre Registos
10. Símbolos Lógicos de Registos MSI Tradicionais
11. Registo de 8 bits em PAL 22V10
12. 74x377 – Octal D-type Positive Edge-triggered Flip-flop with Data Enable
13. Registo de Deslocamento (Shift Register)
14. Registo de Deslocamento Bidireccional
15. 74x195 – 4-Bit Parallel Access Shift Register
16. 74x195 – Estrutura Interna
17. 74x194 – 4-bit Universal Shift Register
18. Shift Register de 8 Bits em PAL 22V10
19. Aplicação de Shift-Registers – detecção de sequências
20. Aplicação de Shift-Registers – Transmissão Série de Informação
21. Transmissão Série de Informação com Conversão Paralelo-série e Série-paralelo
22. Aplicação de Shift-Registers – Contador em Anel
23. Contador em Anel com 74x194
24. Contador em Anel Auto-corrector com 74x194
25. Aplicação de Shift-Registers – Gerador de Código Cíclico (Johnson ou Moebius)



26. Aplicação de Shift-Registers – Contador Linear Feedback Shift Register (LFSR)
27. Equações de Realimentação dos Linear Feedback Shift Register (LFSR)
28. Aplicação de Shift-Registers – Somador Série de 4 bits
29. Contador Assíncrono por Pulsação (Ripple Counter)
30. Contador Síncrono
31. Contador Síncrono Programável de 3 bits
32. Contador Síncrono Programável de 3 bits Up e Down
33. Contador Síncrono Programável de 3 bits com Load e Clear
34. Contador Programável de 3 bits – Último Andar e Saídas de Controle
35. Estrutura Interna de um Contador Síncrono Programável de 3 bits
36. Entrada de Controle Clear em Variantes Assíncrona e Síncrona
37. Efeitos na Temporização de uma Entrada de Controle em Modo Assíncrono e Síncrono
38. Contador Módulo-5 – Diagrama de Estados e Tabela de Transição de Estados
39. Contador Módulo-5 – Estados Inválidos e Programação dos Flip-flops
40. Contador Módulo-5 – Diagramas Lógico e Temporal
41. 74x192/3 – Up/Down 4-bit Decade/Binary Counters
42. 74x163 – Synchronous Presettable 4-bit Binary Counter
43. Diagrama Temporal observado nas Entradas e Saídas do Contador 74x163
44. Contadores 74x163 e 74x193 – Diagrama Temporal dos Sinais TC e CY\_L
45. Concatenação de Módulos Contadores de 4 Bits
46. Contador de 4 bits em PAL 22V10
47. Contador Módulo-10 com Sequência de Contagem Repetitiva 6..15
48. Contador Módulo-7 com Sequência de Contagem Repetitiva 6..12
49. Contador com Sequência de Contagem 0 a 15 sem passar pelas contagens 4, 8 e 12
50. Contador com Sequência de Contagem 0 a 15 sem passar pelas contagens 4, 8 e 12



- 51. LSD – 8 Índice 1
- 52. LSD – 8 Índice 2
- 53. LSD – 8 Índice 3

