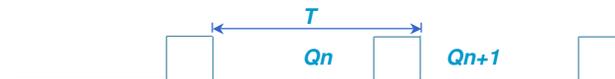


# Circuitos Digitais

## Aula 10

## Circuitos Seqüenciais

- Muitos sistemas digitais são pulsados ou clocked. Isto é, eles operam em sincronismo com um trem de pulsos de período  $T$  chamado relógio (clock).

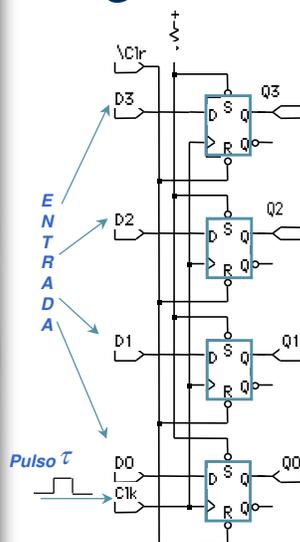


- A cada transição de clock dizemos que a máquina passa para um próximo estado.
- Se considerarmos um certo  $Q_n$  a saída em um dado ponto do sistema no intervalo precedendo o  $n_{th}$  pulso de clock, então  $Q_{n+1}$  é a saída correspondente no intervalo imediatamente depois do  $n_{th}$  pulso.
- Tal sistema onde os valores  $Q_1, Q_2, Q_3, \dots, Q_n$  são obtidos em seqüência no tempo em intervalos  $T$  é chamado um sistema lógico seqüencial.

# Registadores

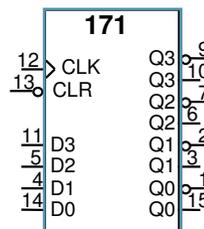
- São dispositivos que armazenam vetores de diferentes dimensões. Estas informações são armazenadas no registrador mediante um sinal de carga externo de carga ( $\tau$ ), e podem ser usadas por outras partes do circuito.
- Os elementos ou células básicas que compõem os registradores são Latches ou Flip-Flops.
- Os registradores podem armazenar dados e também deslocá-los para a direita ou para a esquerda. **Registadores que permitem deslocamento de seus dados são denominados Registradores de Deslocamento (Shift-Register).**
  - Desloca para a direita
  - Desloca para a esquerda

## Registadores - (transferência paralela)



-Quando o pulso é aplicado a entrada Clk (clock) a entrada é carregada no registrador.  
 -O sinal de entrada D0-D3 deve permanecer inalterado durante a transação.

### Componente comercial



TTL 74171

# Eventos

$\tau_i$ : <expressão de transferência>

O evento descrito pela expressão de transferência ocorre no instante  $\tau_i$

Muitas vezes o processamento de uma informação exige uma seqüência de transferência até que a tarefa seja terminada.

**Exemplo: Adição de três valores** ( X e Y são dois registradores)

**tempo do evento 1** - Faça X igual a zero e leia o primeiro

$\tau_1$ : X <- [0]      valor de Y. Acontece então a primeira soma parcial.

Y <- [A]

**tempo do evento 2** - Carregue valor em X, leia o

$\tau_2$ : X <- X+Y      segunda valor de Y e então acontece a segunda soma parcial.

Y <- [A]

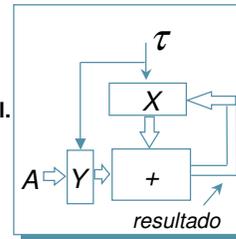
**tempo do evento 3** - Carregue o valor em X, leia o

$\tau_3$ : X <- X+Y      terceiro valor de Y e então acontece a terceira soma.

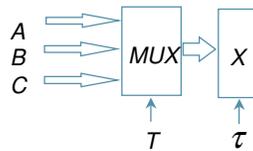
Y <- [A]

**tempo do evento 4** - Carregue o resultado da soma em X e Y mantêm-se inalterado.

$\tau_4$ : X <- X+Y      Y <- [A]



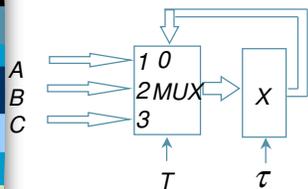
# Entradas Multiplexadas



$\tau$ : X <- (A  $\wedge$  (T=[1])  $\vee$  B  $\wedge$  (T=[2])  $\vee$  C  $\wedge$  (T=[3]))

O que acontece com T = [0] ? ou qualquer outro valor não especificado na equação?

- A expressão a esquerda será nula, logo:  
X <- 0

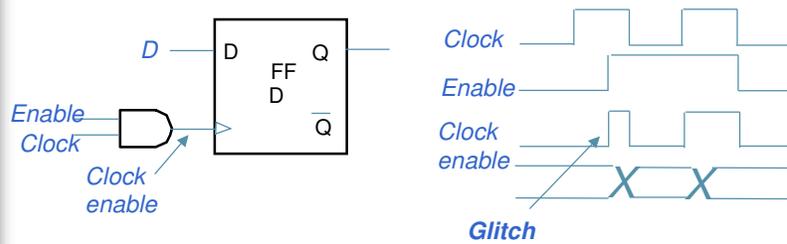


$\tau$ : X <- (X  $\wedge$  (T=[0])  $\vee$  A  $\wedge$  (T=[1])  $\vee$  B  $\wedge$  (T=[2])  $\vee$  C  $\wedge$  (T=[3]))

Obs: Nesse caso se T=[0] o conteúdo de X permanece inalterado

## Projetos de circuitos síncronos

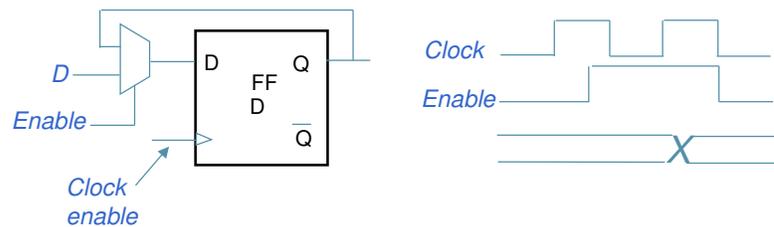
- Clock controlado por gates para controle de dados em registradores (não recomendado)



Obs: Flip-Flop carrega valor na subida do clock

## Projetos de circuitos síncronos

- Controle de enable direto no mux interno do registrador.
- O clock é livre e atua direto no circuito do Flip-flop. (opção recomendada)



Obs: Flip-Flop carrega valor na subida do clock

## Registrador Controlado

- O circuito pode realizar diferentes funções dependendo do sinal de controle.

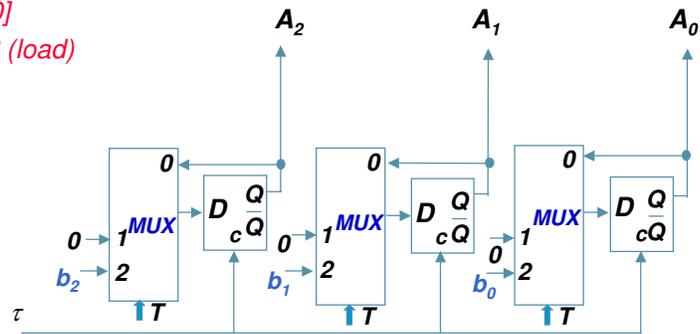
$$\tau : A \leftarrow (T=[0]) \wedge A \vee (T=[1] \wedge SR(x,A)) \vee (T=[2]) \wedge SL(A,x)$$

*T Operação*

*[0] A ← A*

*[1] A ← [0]*

*[2] A ← B (load)*



## Transferência de informação

*Transferência em paralelo*

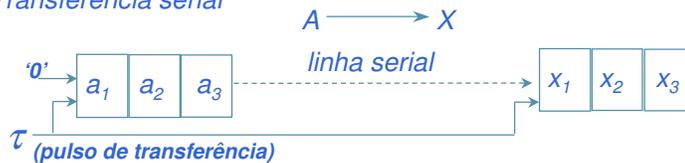


*Transferência serial*



# Transferência de informação

Transferência serial



tempo do evento	transferência		A e X após da transferência	
	A	X	A	X
$\tau_0$ :	$A \leftarrow [1,0,1]$	$X \leftarrow [0,0,0]$	$[1,0,1]$	$[0,0,0]$
$\tau_1$ :	$A \leftarrow SR([1,0,1])$	$X \leftarrow SR([1],[0,0,0])$	$[0,1,0]$	$[1,0,0]$
$\tau_2$ :	$A \leftarrow SR([0,1,0])$	$X \leftarrow SR([0],[1,0,0])$	$[0,0,1]$	$[0,1,0]$
$\tau_3$ :	$A \leftarrow SR([0,0,1])$	$X \leftarrow SR([1],[0,1,0])$	$[0,0,0]$	$[1,0,1]$

Após a transferência os valores nos registradores A e X são:

$A := [0,0,0]$

$X := [1,0,1]$

## Registrador de deslocamento - Shift Register

- Registradores de deslocamento são dispositivos adequados para conversão Serial-Paralela-Serial normalmente utilizados em sistemas de comunicação em terminais de computadores.
- Exemplo: Registrador de deslocamento (direita/esquerda)
  - **Para a direita**
    - Desloca uma posição em todos os elementos do registrador da esquerda para a direita.
    - O bit menos significativo (LSB-bit mais a direita) realimenta o bit mais significativo (MSB - bit mais a esquerda) do registrador ou um novo bit é adicionado ao deslocamento.
  - **Para a esquerda**
    - Desloca uma posição em todos os elementos do registrador da direita para a esquerda.
    - O bit Mais significativo (MSB-bit mais a esquerda) realimenta o bit menos significativo (LSB - bit mais a direita) do registrador ou um novo bit é adicionado ao deslocamento.

# Registrador de Deslocamento

- O circuito pode realizar diferentes funções dependendo do sinal de controle.

$$\tau : A \leftarrow (T=[0]) \wedge A \vee (T=[1]) \wedge SR(x,A) \vee (T=[2]) \wedge SL(A,x)$$

*T* Operação

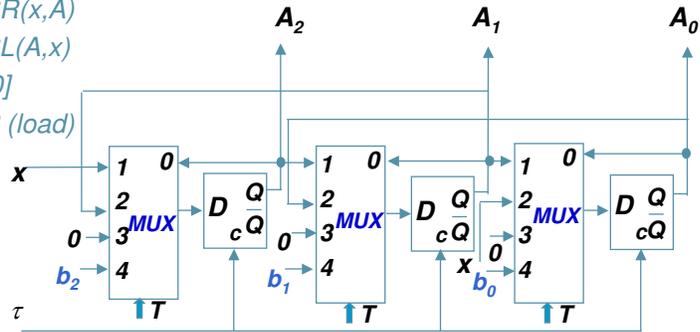
[0]  $A \leftarrow A$

[1]  $A \leftarrow SR(x,A)$

[2]  $A \leftarrow SL(A,x)$

[3]  $A \leftarrow [0]$

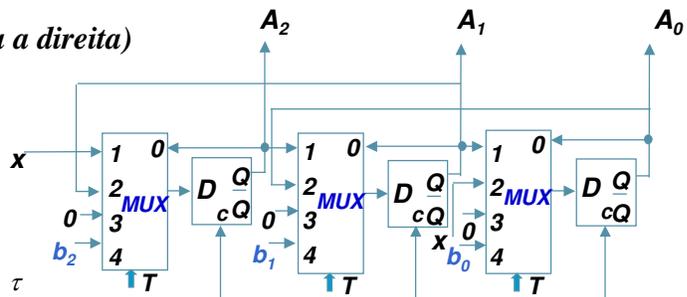
[4]  $A \leftarrow B$  (load)



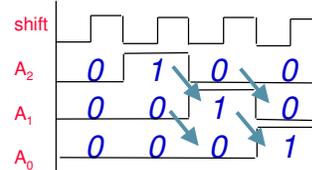
## Registrador de deslocamento

( Shift Register )

(deslocamento para a direita)

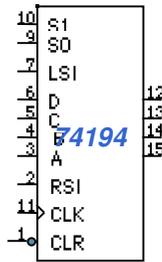


Controle  $\rightarrow$  T=3 T=4 T=1 T=1



# Registrador de deslocamento

Entrada Serial vs. Paralela  
 Saída Serial vs. Paralela  
 Direção de deslocamento: Esquerda vs. Direita



74194 4-bit Universal Shift Register

Entradas Seriais LSI (Left Serial Input)  
 RSI (Right Serial Input)

Entradas Paralelas: D, C, B, A  
 Saídas Paralelas: QD, QC, QB, QA

Sinal de Clear  
 Dispositivo trigerado na subida do relógio

*S1, S0 determina a função de deslocamento*

S1 = 1, S0 = 1: Carrega na subida do relógio  
 - Carga síncrona

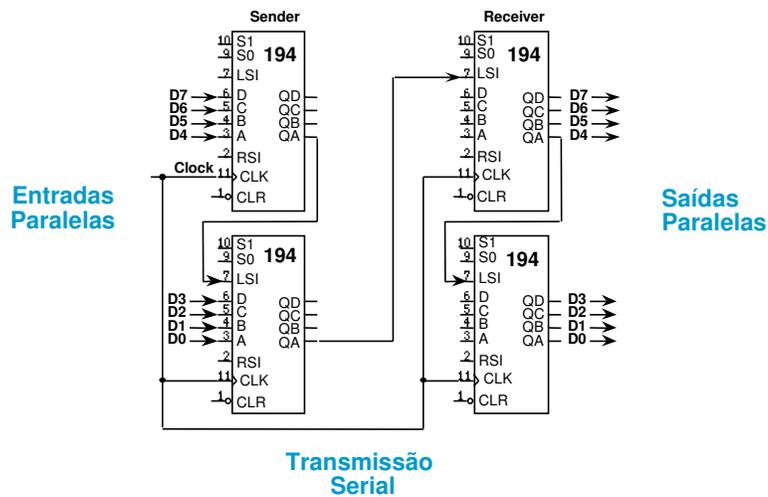
S1 = 1, S0 = 0: Desloca para Esquerda na subida do relógio  
 - LSI substitui o elemento D

S1 = 0, S0 = 1: Desloca para Direita na subida do relógio  
 - RSI substitui o elemento A

S1 = 0, S0 = 0: Preserva os valores

# Registrador de Deslocamento

Conversão Paralelo <=> Serial

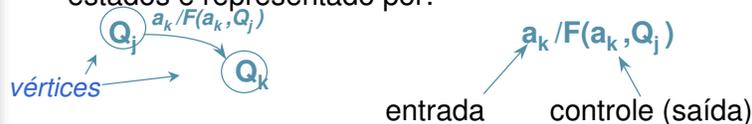


## Diagrama de estados

- Diagrama de estados é uma técnica de projeto que permite ao projetista representar graficamente o funcionamento comportamental de uma máquina de estados finita (fsm).
- Cada diagrama de estados consiste de um conjunto identificado por vértices que correspondem aos estados do circuito e por arcos que indicam a transição entre estados.
- Cada par de vértices (não necessariamente distintos) são conectados por um arco direcionado indicando a transição de um estado para outro. Este arco existe apenas se um determinado sinal de entrada força a mudança de estado da máquina de um estado  $Q_j$  para o estado  $Q_k$ .

## Diagrama de estados - Mealy

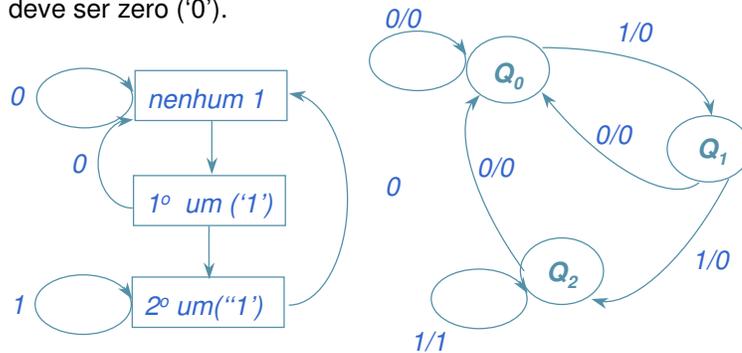
- Se um arco direcionado conecta  $Q_j$  a  $Q_k$  quando a entrada da máquina for  $a_k$ , então o arco entre os dois estados é representado por:



- \* **Máquina Mealy - Observe que a saída é função da entrada e do estado atual**
- Assim vemos que:
  - Vértices correspondem ao estado corrente do circuito.
  - O código descrito no arco representa “entrada corrente” e a “saída corrente”.
  - A indicação da direção do arco indica o próximo estado

## Diagrama de estado - aplicação

- Exemplo:
- Projete um circuito que gere saída '1' quando for observado 3 uns '1's consecutivos na entrada. Nos demais casos a saída deve ser zero ('0').



## Conversão do diagrama de estados

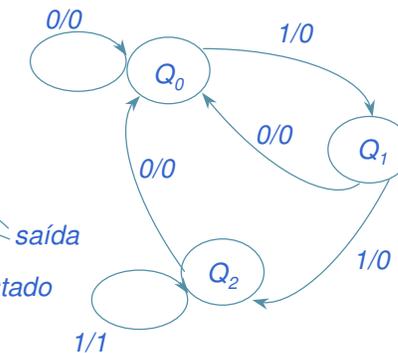
Variáveis de entrada

Tabela de estados

Q \ x	0	1
Q <sub>0</sub>	Q <sub>0</sub> /0	Q <sub>1</sub> /0
Q <sub>1</sub>	Q <sub>0</sub> /0	Q <sub>2</sub> /0
Q <sub>2</sub>	Q <sub>0</sub> /0	Q <sub>2</sub> /1

estado presente

próximo estado



saída

1/1

Codificação de estados:

Q<sub>0</sub> -> 00  
 Q<sub>1</sub> -> 01  
 Q<sub>2</sub> -> 10

Tabela de transição

Q \ x	0	1
00	00/0	01/0
01	00/0	10/0
10	00/0	10/1

## Implementação do circuito

- Tabela verdade

entrada	estado presente		próximo estado		Flip-Flop				saída
<b>x</b>	<b>y<sub>1</sub></b>	<b>y<sub>2</sub></b>	<b>y<sub>1+t</sub></b>	<b>y<sub>2+t</sub></b>	<b>J<sub>1</sub></b>	<b>K<sub>1</sub></b>	<b>J<sub>2</sub></b>	<b>K<sub>2</sub></b>	<b>Z</b>
0	0	0	0	0	0	X	0	X	0
0	0	1	0	0	0	X	X	1	0
0	1	0	0	0	X	1	0	X	0
0	1	1	X	X	X	X	X	X	X
1	0	0	0	1	0	X	1	X	0
1	0	1	1	0	1	X	X	1	0
1	1	0	1	0	X	0	0	X	1
1	1	1	X	X	X	X	X	X	X

\* Implementação da máquina de estados usando Flip-Flop tipo JK

## Implementação do circuito

- Equações booleanas

$$y_1(t+1) = x y_1 + x y_2$$

$$y_2(t+1) = x y_1 + \bar{x} y_2$$

$$Z = x y_1$$

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	0	0	X	0
1	0	0	X	0

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	0	0	X	0
1	1	0	X	1

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	0	0	X	0
1	0	0	X	1

$$J_1 = x y_2$$

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	0	0	X	X
1	0	1	X	X

$$J_2 = \bar{y}_1 x$$

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	0	X	X	0
1	1	X	X	0

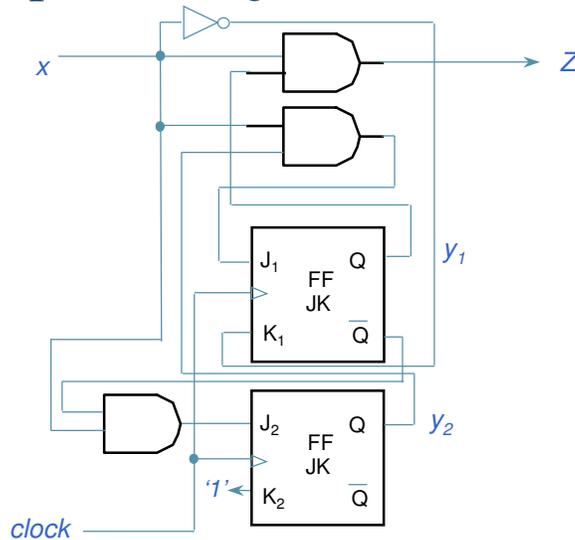
$$K_1 = \bar{x}$$

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	X	X	X	1
1	X	X	X	0

$$K_2 = 1$$

x \ y <sub>1</sub> y <sub>2</sub>	00	01	11	10
0	X	1	X	X
1	X	1	X	X

## Implementação do circuito



## Contadores

- São máquinas seqüenciais (FSMs) que possuem uma seqüência de estados bem definida em resposta a sinal de contagem.
- Os contadores podem contar para frente ou para trás (up/down) e podem ser construídos por diferentes tipos de Flip-Flops.
- A saída de um contador expõe o estado atual da máquina de estados.
- Exemplo:
  - Contador de 2 bits: 00, 01, 10, 11 (contador módulo 4)
  - Contador de 4 bits: 0000, ..., 1111 (Contador binário de 4 bits - módulo 16)

# Contadores - implementação

- ☑ Implementar um contador síncrono octal

Exemplo Contador binário de 3 bits

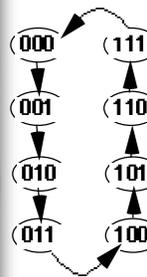


Diagrama de transição de estados

Estado Presente			Próximo Estado			Flip-flops tipo T		
C	B	A	C+	B+	A+	TC	TB	TA
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Tabela de Transição de Estados      Tabela de entrada dos Flip-Flops

Implementação com Flip-Flops tipo T (Toggle)

# Contadores - implementação

- ☑ Mapa K para Flip-Flops tipo T

	CB			
A	00	01	11	10
0				
1				

TA =

	CB			
A	00	01	11	10
0				
1				

TB =

	CB			
A	00	01	11	10
0				
1				

TC =

## Contadores - implementação

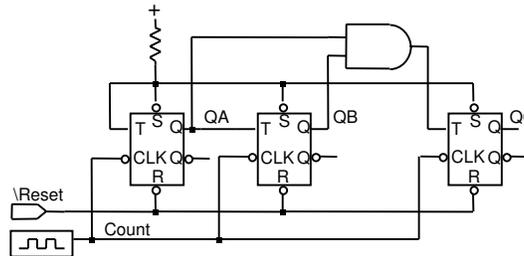
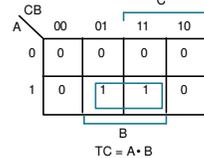
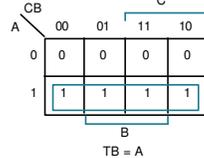
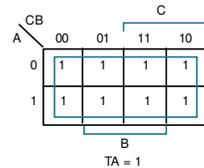
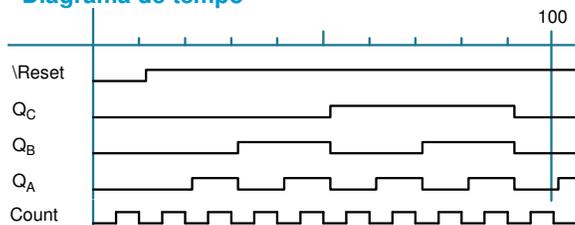
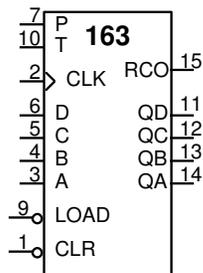


Diagrama de tempo



## Contadores

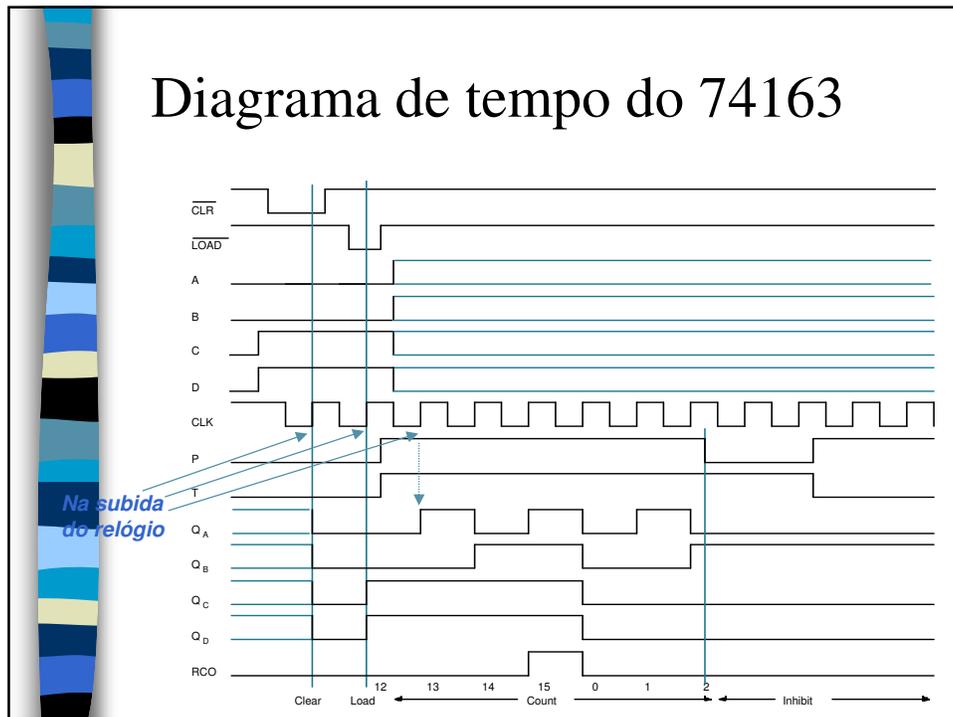
### ☑ Contador Síncrono - 74163



Contador UP síncrono de 4-Bits (74163)

- Sinais de entrada Load and Clear síncronos (sincronizados com o relógio)
- Flip-Flop trigados na subida do relógio
- Load Paralelo a partir dos sinais de entrada D, C, B, A
- P, T Enable Inputs: Ambos os sinais devem ser habilitados para permitir contagem
- RCO ("Ripple Carry Output") acionado quando o contador atinge o último estado "1111". Este sinal é normalmente usado quando queremos cascatear contadores.

## Diagrama de tempo do 74163



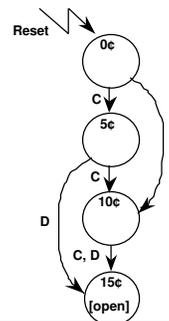
## Projeto - Síntese Lógica

Implementar uma máquina de vender bombom

**•Características:**

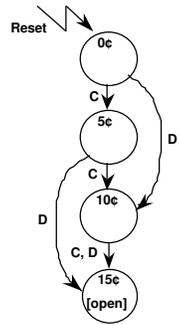
- Recebe moedas de 5 e 10 centavos
- Não dá troco
- Libera bombom quando valor recebido é igual ou superior a 15 centavos

Diagrama de estados



Obs: Implementar projeto com Flip-Flop tipo JK

# Projeto - Síntese Lógica



Present State	INPUT		Next State	Output Open
	D	C		
0c	0	0	0c	0
	0	1	5c	0
	1	0	10c	0
	1	1	X	X
5c	0	0	5c	0
	0	1	10c	0
	1	0	15c	0
10c	1	1	X	X
	0	0	10c	0
	0	1	15c	0
15c	1	0	15c	0
	1	1	X	X
	X	X	15c	1

