

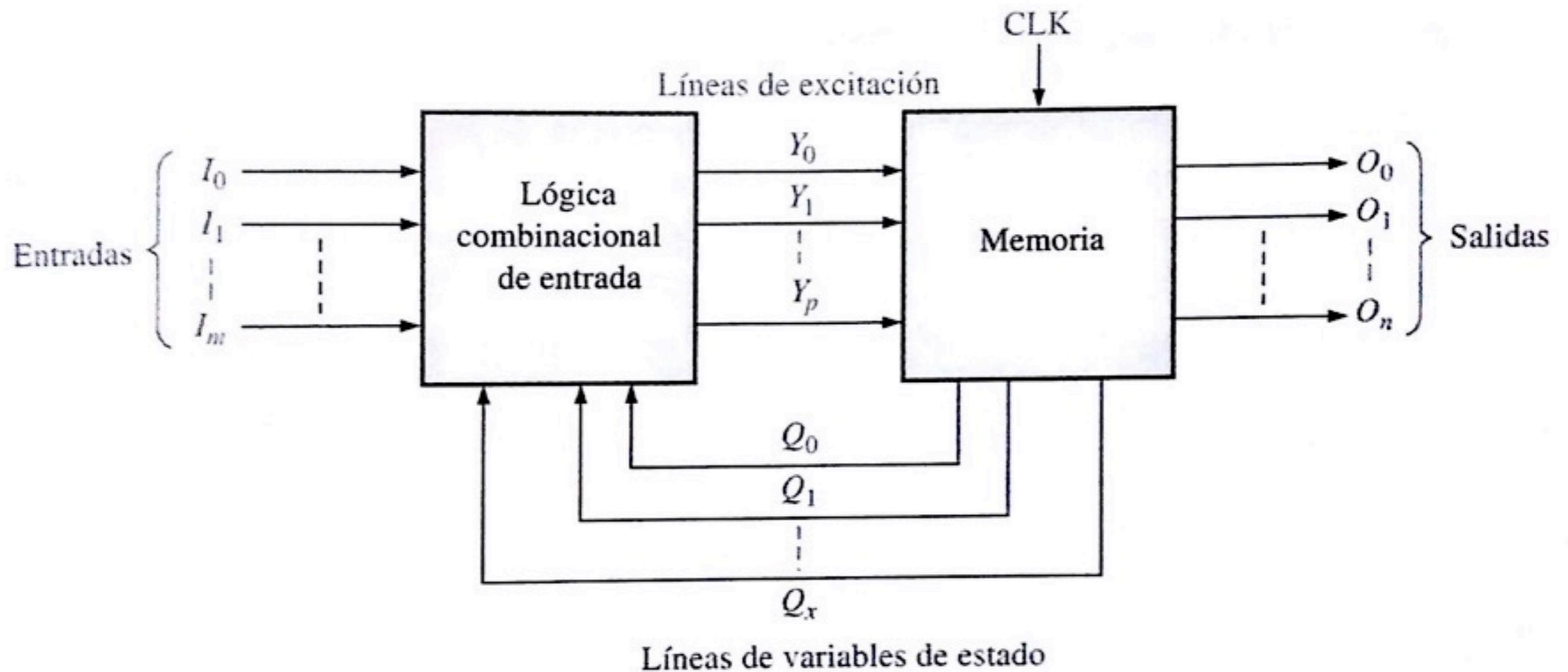
# Máquinas (de Estado) Síncronas (infinitas e finitas)

Circuitos Digitais II  
Prof. Fernando Passold

*Última atualização: 03 Nov 2017*

# Máquina Secuencial

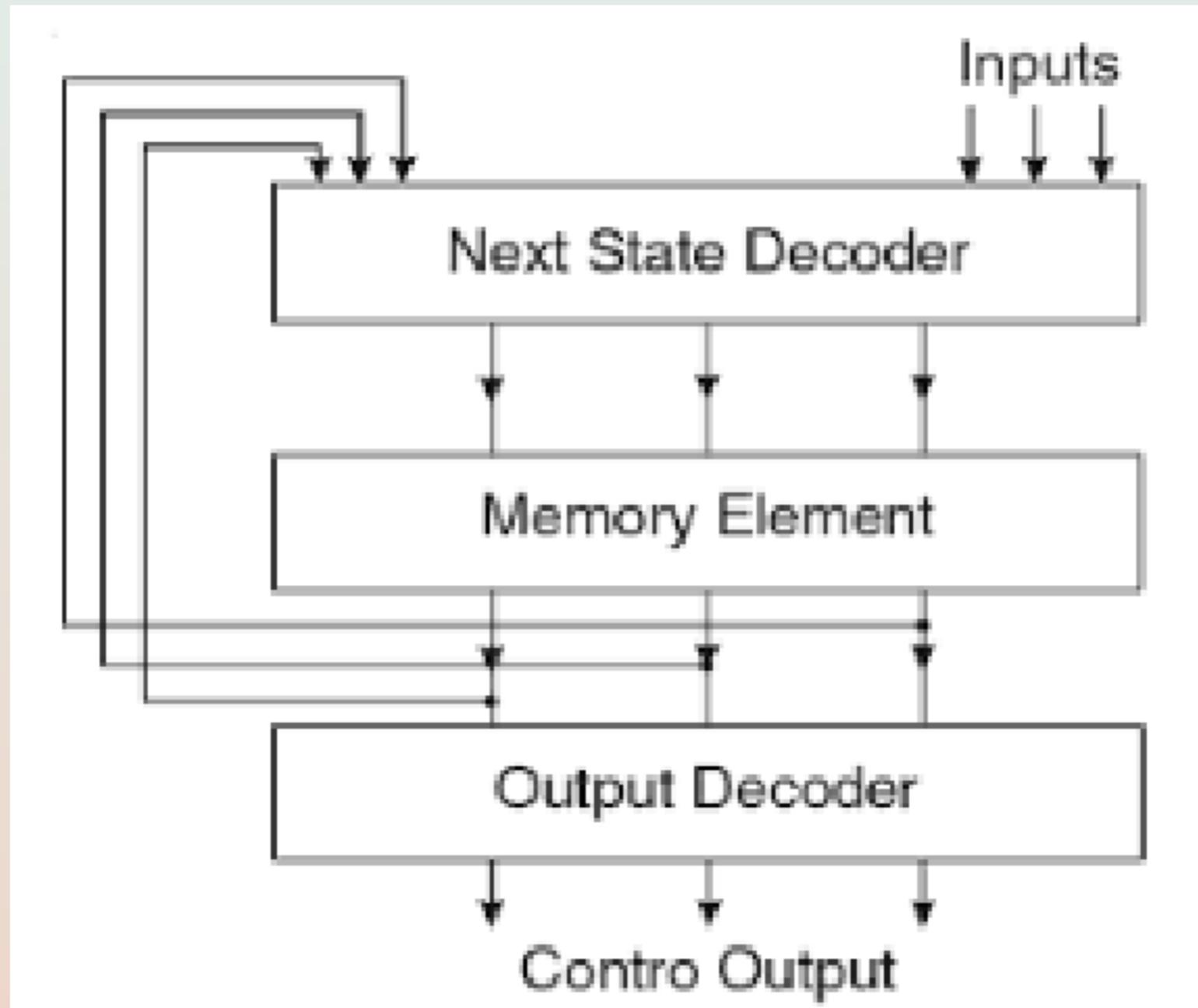
## I) Diagrama de Blocos genérico:



**FIGURA 8.27** Circuito secuencial general sincronizado.

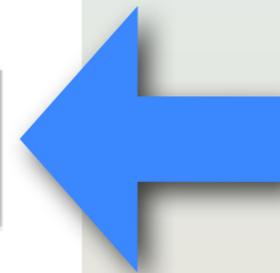
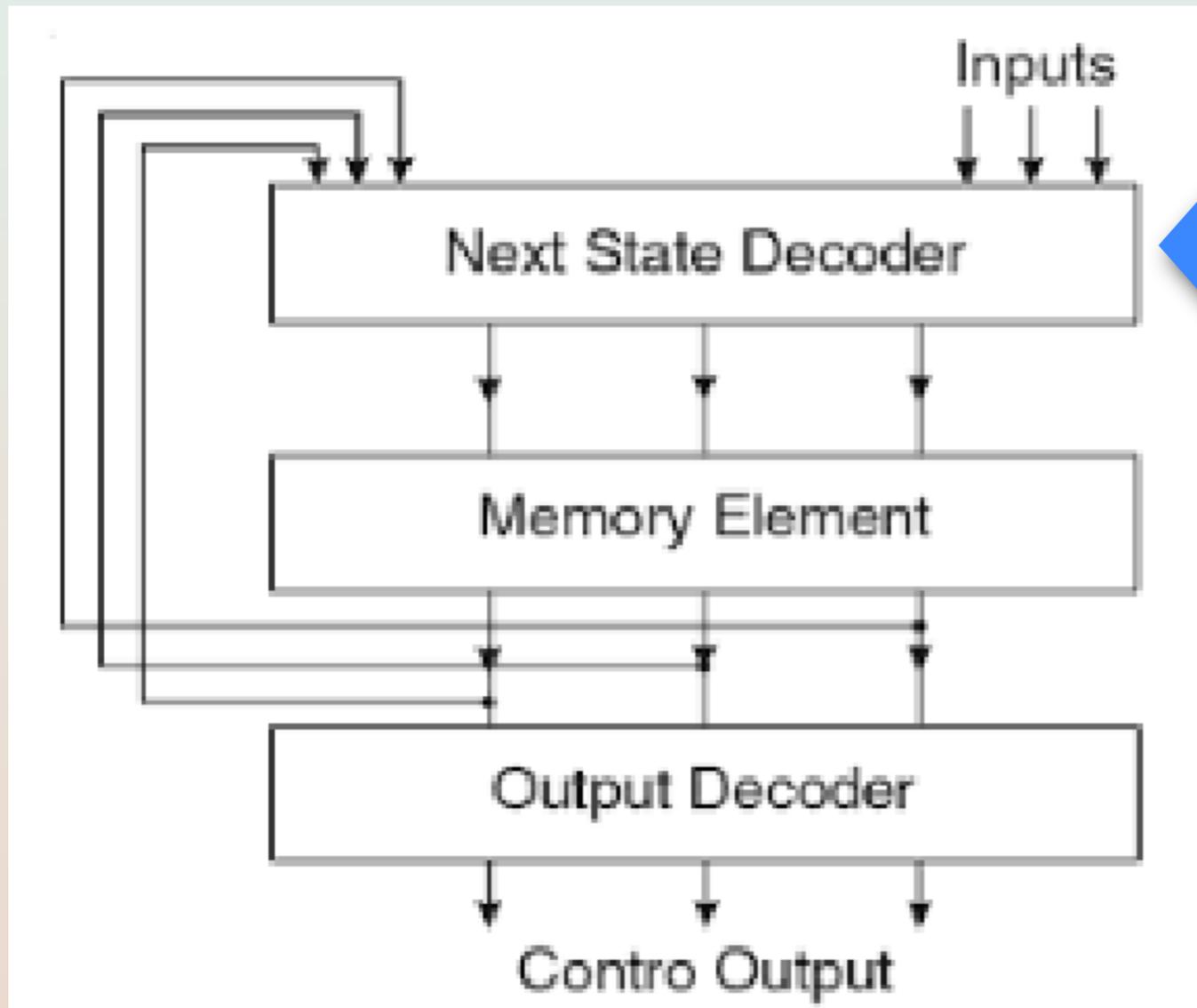
# Máquina Sequencial

I) Diagrama de Blocos genérico (variação):



# Máquina Sequencial

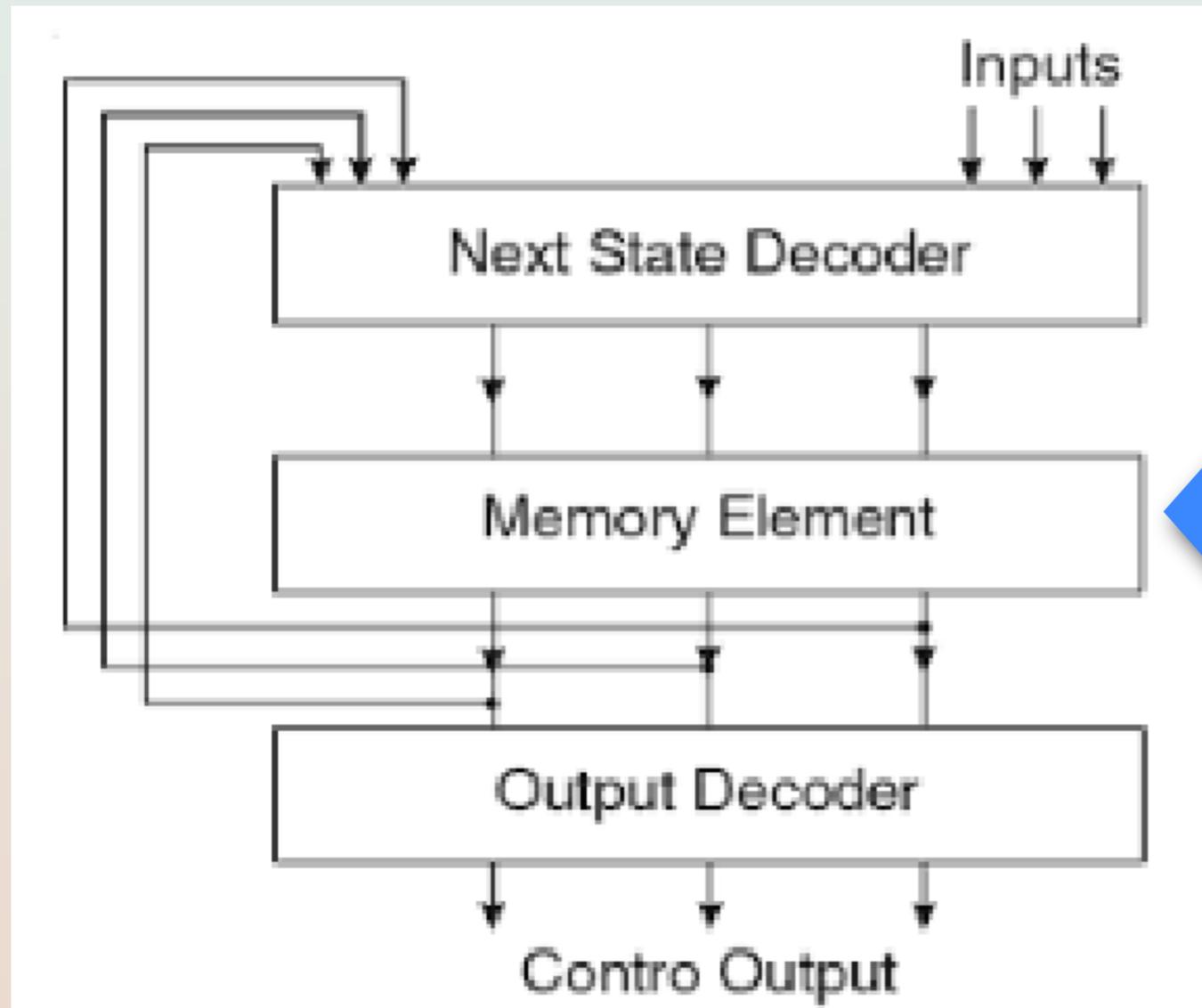
## I) Diagrama de Blocos genérico:



Define como o circuito deve “contar” (define a sequencia binária gerada)

# Máquina Sequencial

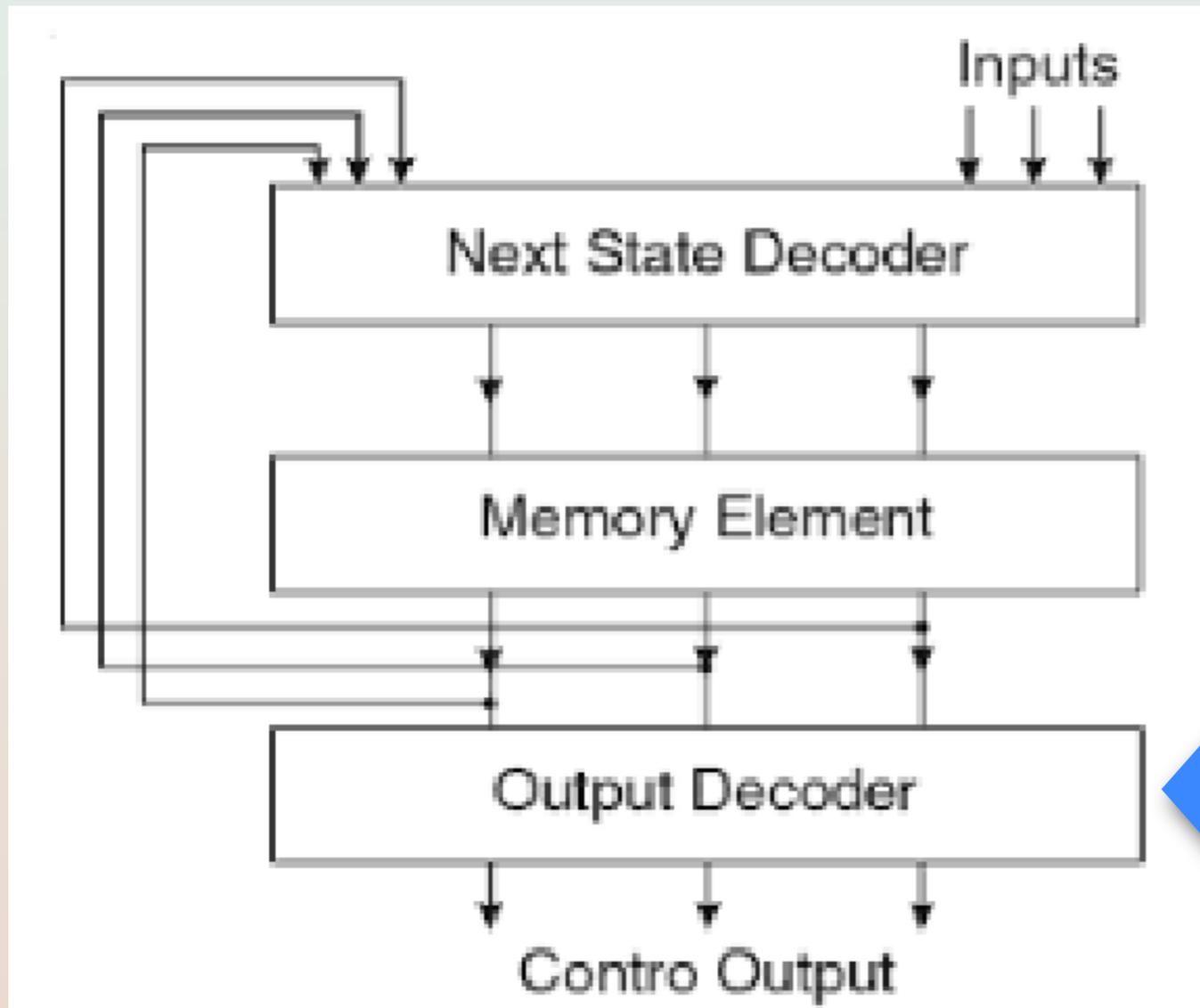
## I) Diagrama de Blocos genérico:



Tantos FF's ou Latches quantos diferentes estados devem ser transitados pelo circuito.

# Máquina Sequencial

## I) Diagrama de Blocos genérico:



Forma a lógica de saída necessária para criar o efeito de saída (interface de saída) desejada.

# Máquinas Sequenciais

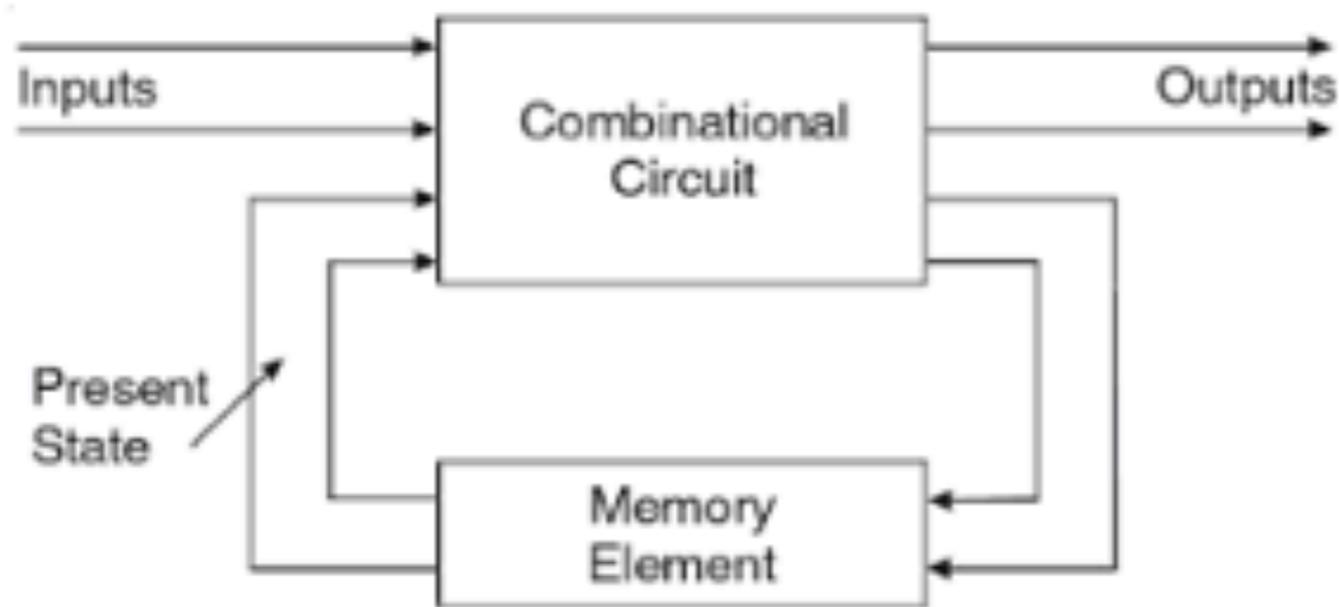


Fig. 10.6. A Asynchronous sequential circuit

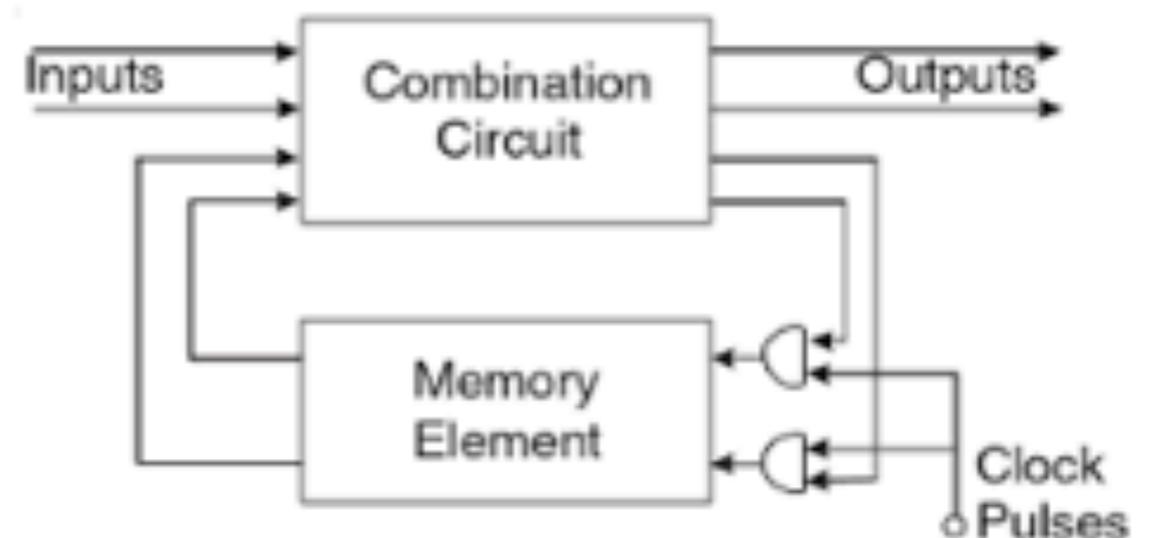


Fig. 10.7. Synchronous sequential circuit

## Circuitos **A**síncronos:

Não possuem NENHUM circuito temporizado associado com o mesmo e uma operação dispara outras operações tão logo ocorra (sem maior controle, ou “sincronização”).

Internamente existe um feed-back (laço de realimentação) para fornecer as informações relacionados com estados internos importantes (significativas) para gerar com sucesso (sem erros) os próximos estados. O laço de realimentação é algumas chamado de “memória” (Flip-Flops ou Latches).

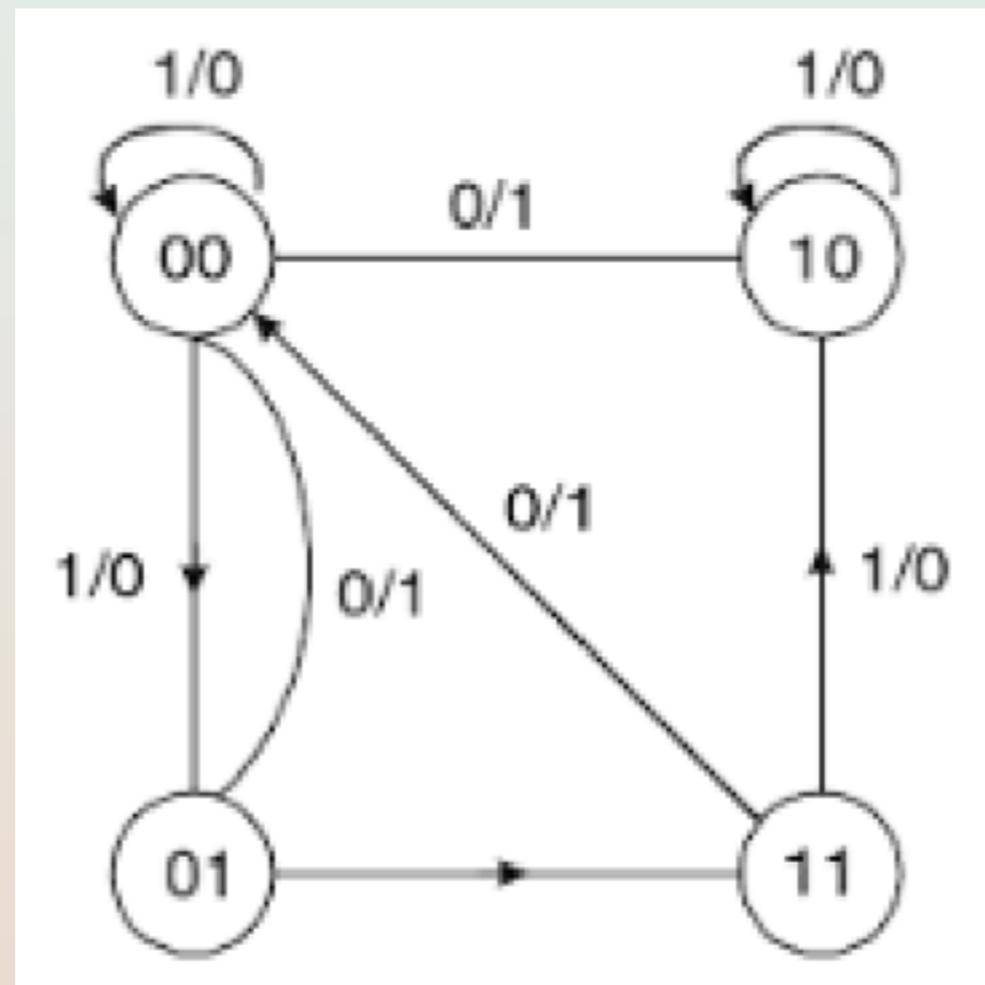
## Circuitos **S**íncronos:

Mais fácil de perceber a diferença com um circuito síncrono, comparando as figuras 10.6 e 10.7.

No circuito síncrono, o MESMO pulso de clock é usado para gatilhar (de forma controlada) a memória do circuito e uma lógica combinacional associada com estes pulsos de clock é que determina como o circuito evolui de um estado para outro.

# Máquina Síncrona

2) Diagrama de estados (exemplo):



# Continuação...

Circuito:

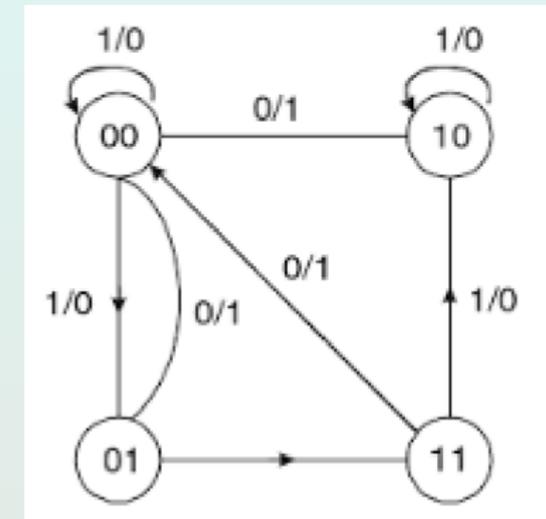
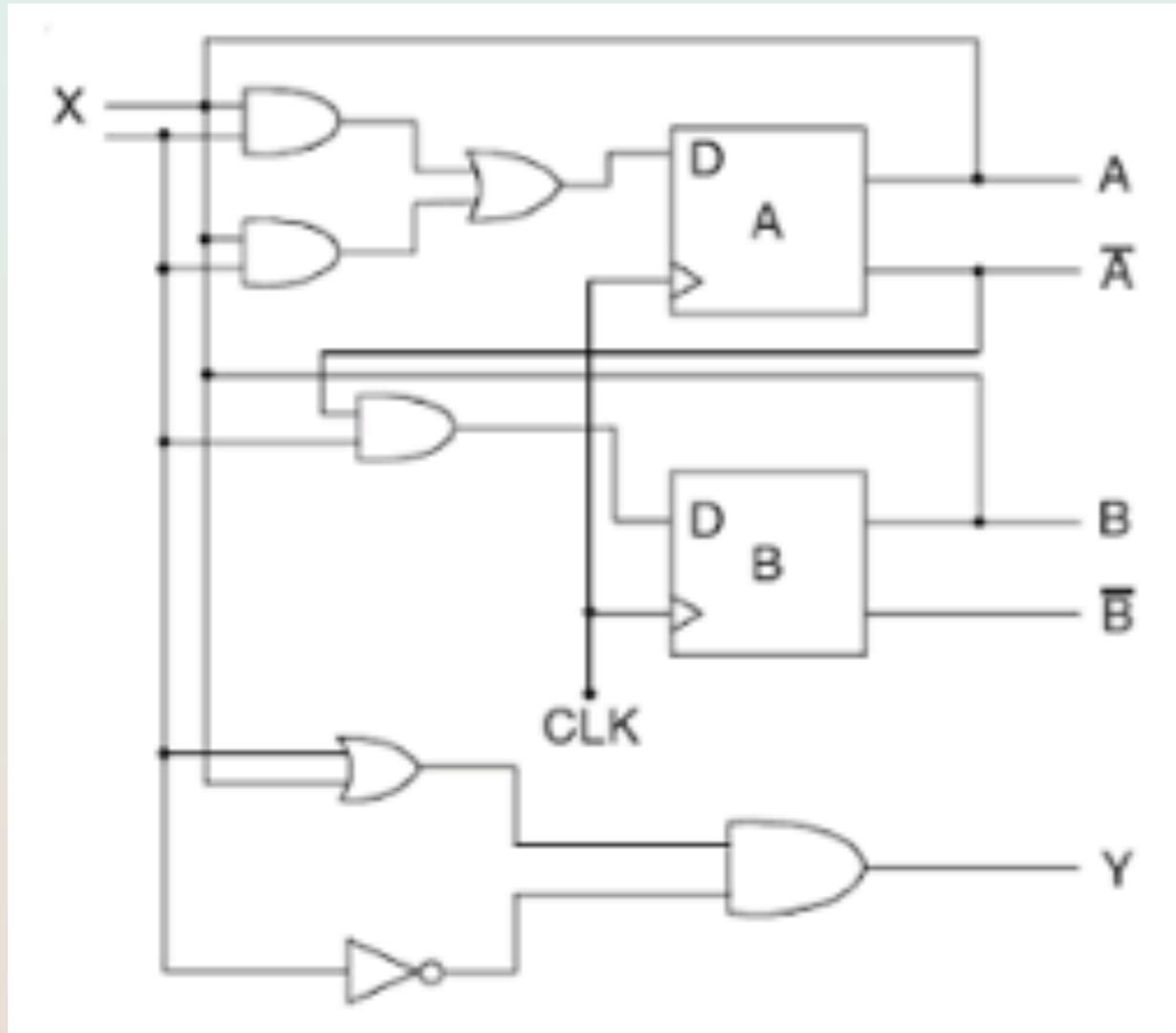


Tabela de transição de estados:

**Table 10.2**

Present state		I/P (X)	Next State		O/P (z)
			A	B	
A	B	X	A	B	(z)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Analisando o circuito notamos que:

$$A(t + 1) = A(t) \cdot X + B(t) \cdot X$$

$$B(t + 1) = \overline{A(t)} \cdot X$$

$$y = [A(t) + B(t)] \overline{X}$$

# Projeto de partes de circuitos síncronos:

- Requer tabelas de transição associadas com os FFs escolhidos para memorizar os diferentes estados necessários para o circuito. A partir destas tabelas fica possível projetar a parte do circuito combinacional (mapas de Karnaugh) usadas para definir o comportamento desejado para o circuito completo (sequencial e síncrono).

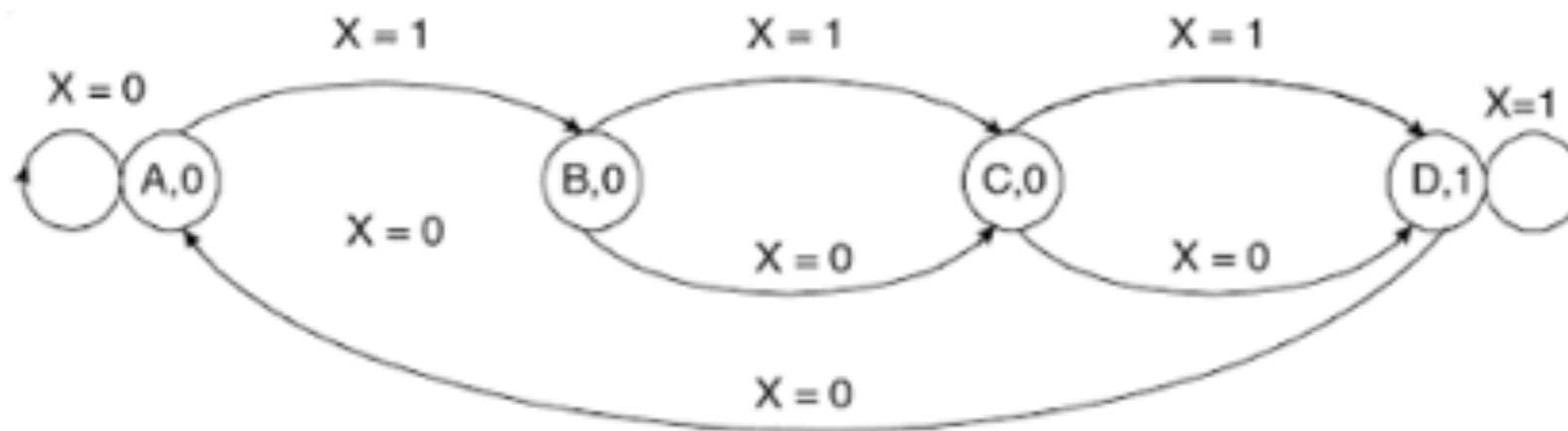


Fig. 10.23. State diagram of the sequential circuit

Table 10.10. (a) State table for the Example 10.6

Present State $Q_n$	Output $Y$	Next State $Q_{n+1}$ for Input $X$	
		0	1
A	0	A	B
B	0	C	C
C	0	D	D
D	1	A	D

Table 10.10. (b) State table rewritten with binary values

Present State $Q_n$		Output $Y$	Next State $Q_{n+1}$ for Input $X$			
			0		1	
$X_1$	$X_2$		$D_1$	$D_2$	$D_1$	$D_2$
0	0	0	0	0	0	1
0	1	0	1	0	1	0
1	0	0	1	1	1	1
1	1	1	0	0	1	1

← Note:

São 4 estados: A, B, C e D. Mas apenas 2 FF's-D são usados para resolver este problema!  
 $2^2 = 4$ .

# Continuação...

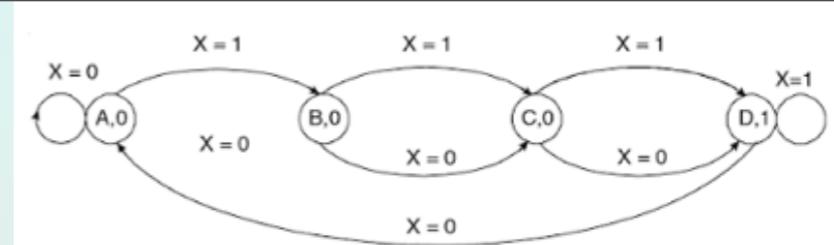


Fig. 10.23. State diagram of the sequential circuit

Table 10.10. (a) State table for the Example 10.6

Present State $Q_n$	Output $Y$	Next State $Q_{n+1}$ for Input $X$	
		0	1
A	0	A	B
B	0	C	C
C	0	D	D
D	1	A	D

Table 10.10. (b) State table rewritten with binary values

Present State $Q_n$	Output $Y$	Next State $Q_{n+1}$ for Input $X$			
		0		1	
		$D_1$	$D_2$	$D_1$	$D_2$
$X_1$ $X_2$					
0 0	0	0	0	0	1
0 1	0	1	0	1	0
1 0	0	1	1	1	1
1 1	1	0	0	1	1

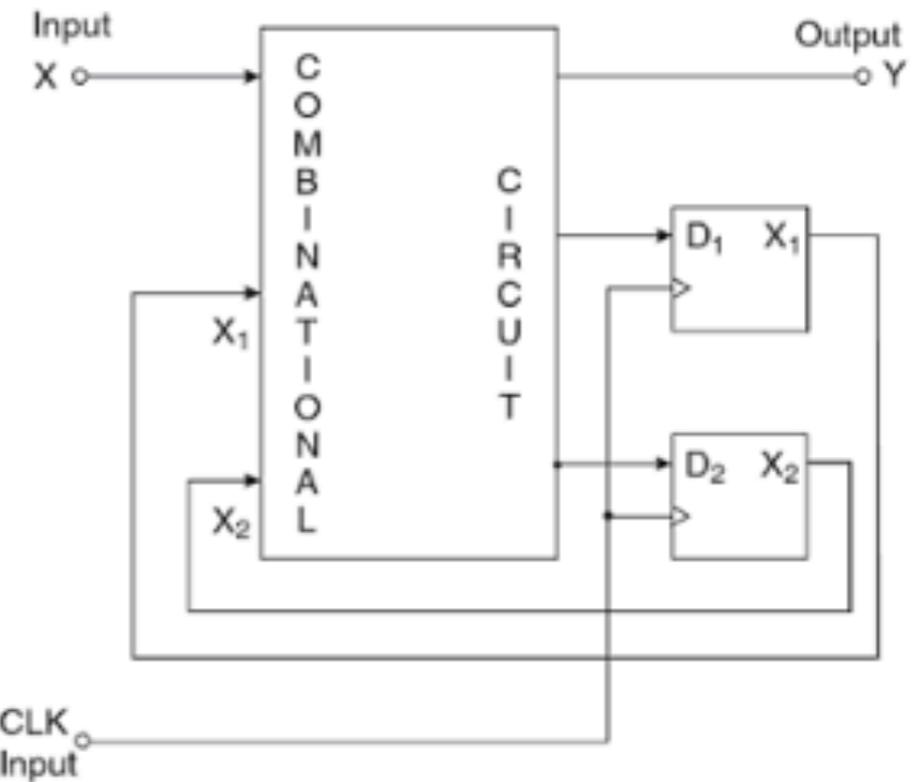


Fig. 10.24. Block diagram of the sequential detector

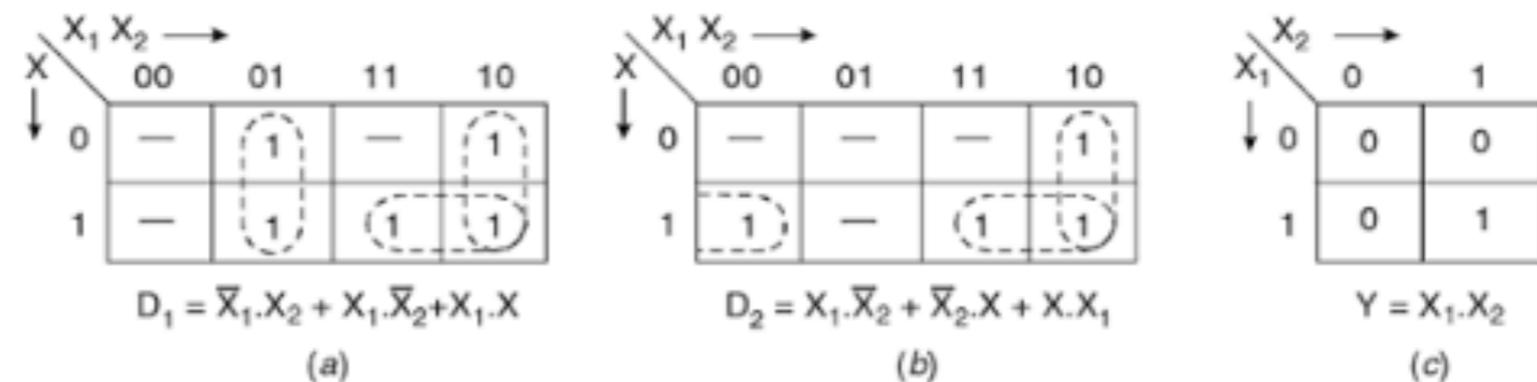


Fig. 10.25. K-maps for (a)  $D_1$  (b)  $D_2$  and (c)  $Y$

From these K-maps, logic expressions for  $D_1$ ,  $D_2$  and  $Y$  are given as:

$$D_1 = X_1 \cdot \bar{X}_2 + \bar{X}_1 \cdot X_2 + X_1 \cdot X$$

$$D_2 = X_1 \cdot \bar{X}_2 + \bar{X}_2 \cdot X + X_1 \cdot X$$

$$Y = X_1 \cdot X_2$$

Combinational circuit is now designed to satisfy these equations and is given in Fig. 10.26.

# ● Continuação:

Table 10.10. (b) State table rewritten with binary values

Present State $Q_n$	Output $Y$	Next State $Q_{n+1}$ for Input $X$			
		0		1	
		$D_1$	$D_2$	$D_1$	$D_2$
$X_1$ $X_2$					
0 0	0	0	0	0	1
0 1	0	1	0	1	0
1 0	0	1	1	1	1
1 1	1	0	0	1	1

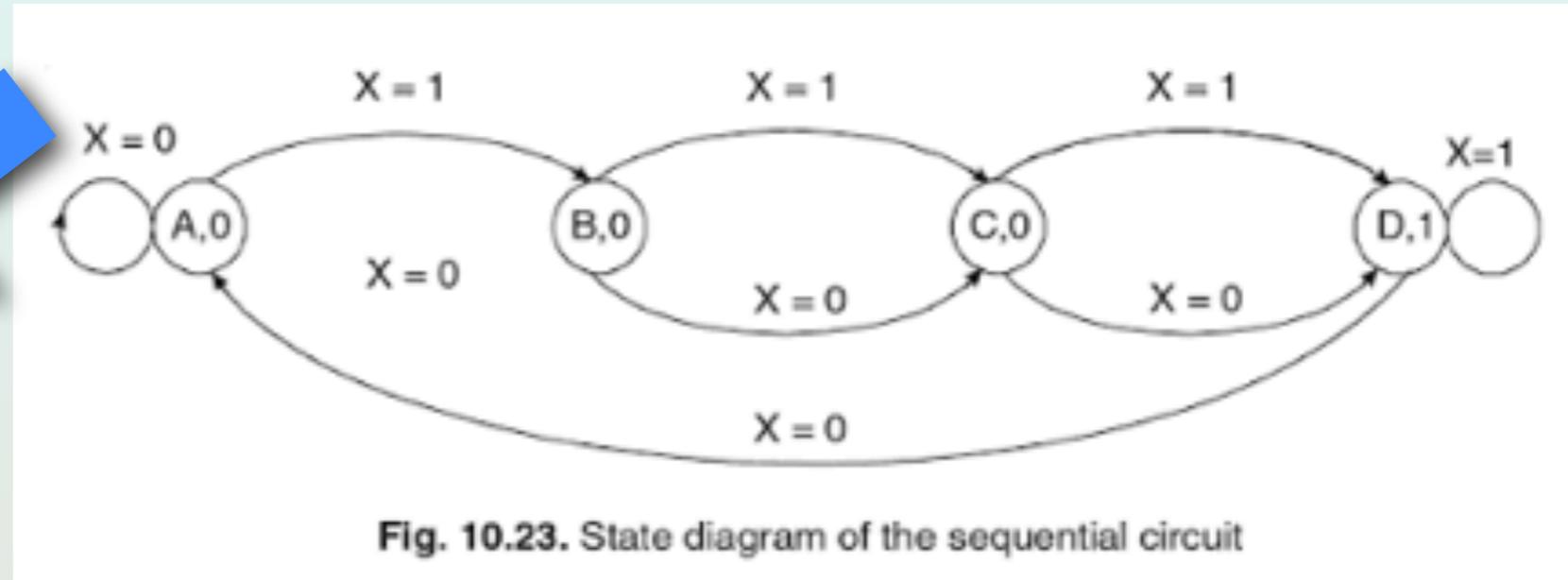


Fig. 10.23. State diagram of the sequential circuit

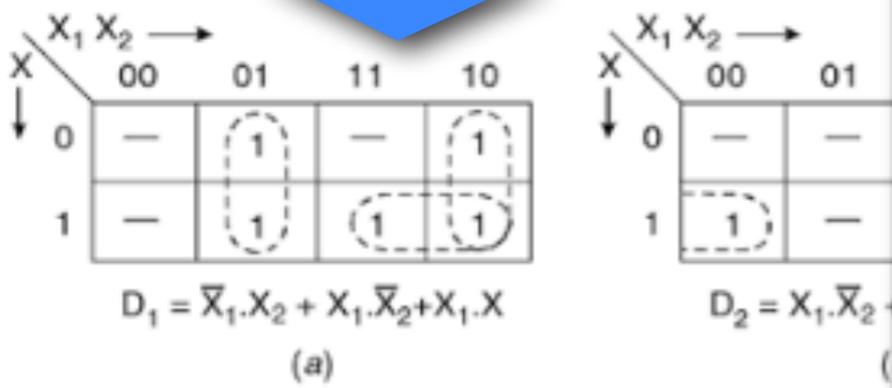


Fig. 10.25. K-maps for (a)  $D_1$  (b)

From these K-maps, logic expressions for  $D_1$

$$D_1 = X_1 \cdot \bar{X}_2 + \bar{X}_1 \cdot X_2 + X_1 \cdot X$$

$$D_2 = X_1 \cdot \bar{X}_2 + \bar{X}_1 \cdot X_2$$

$$Y = X_1 \cdot X_2$$

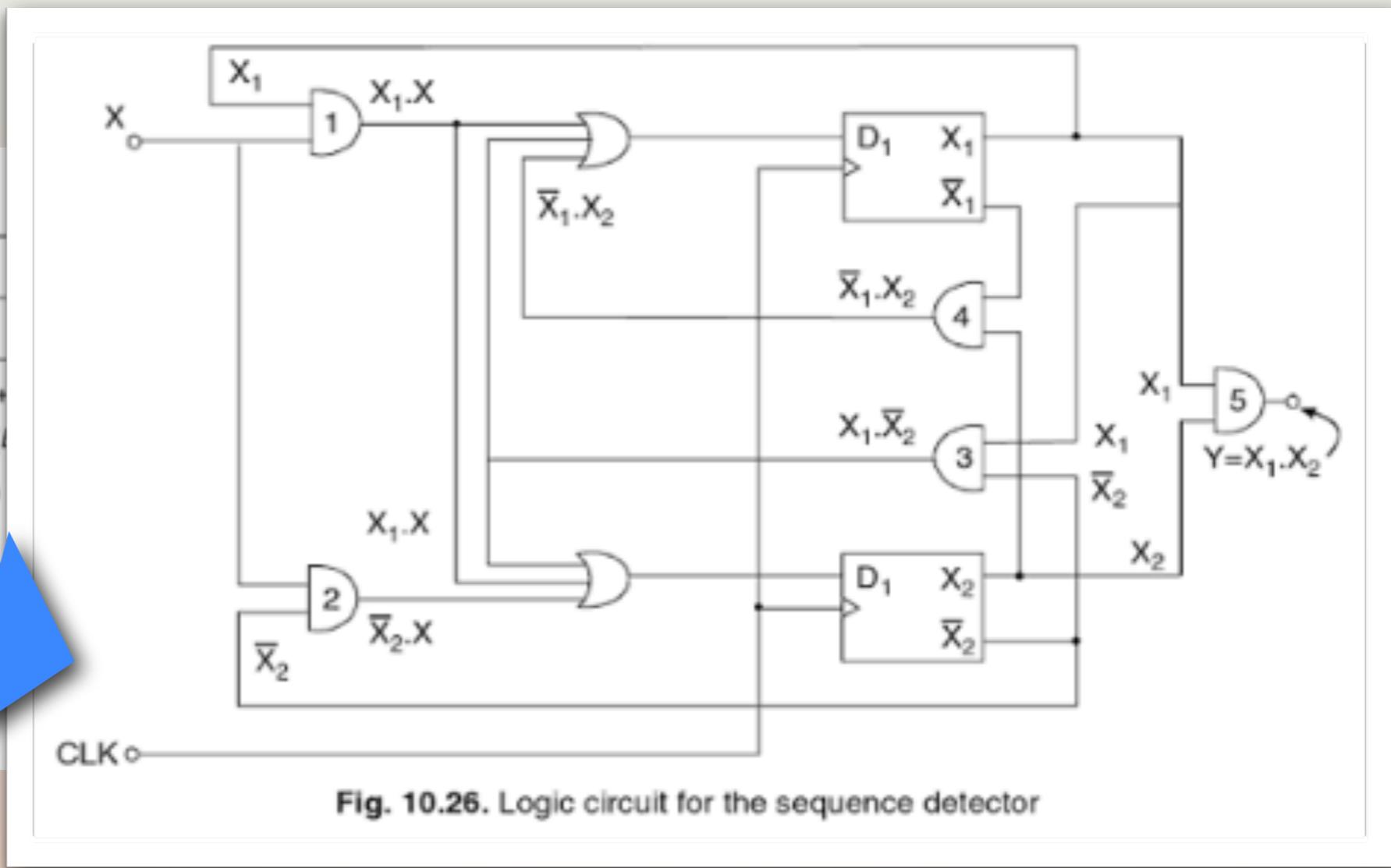


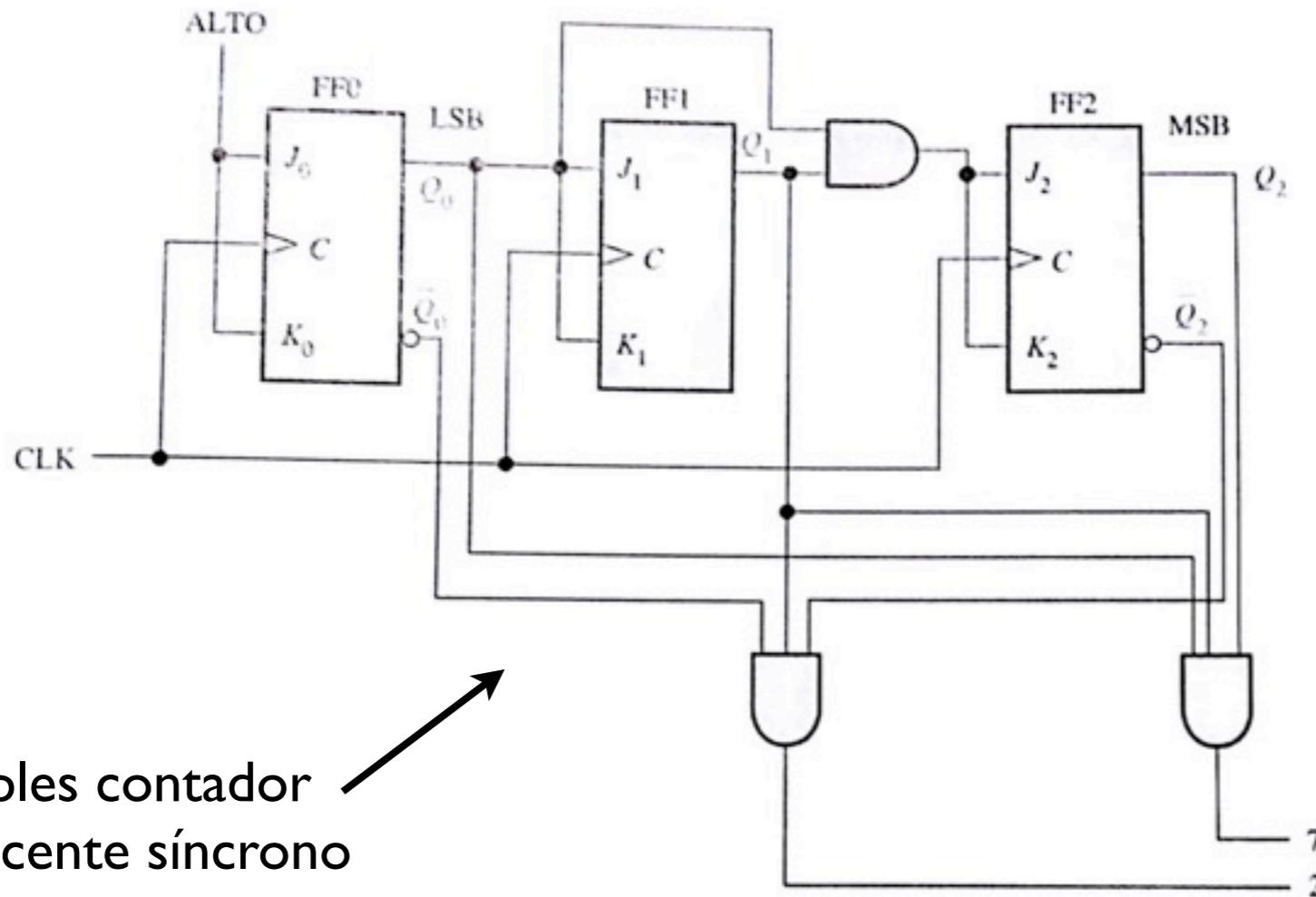
Fig. 10.26. Logic circuit for the sequence detector

# Decodificar saídas de Contador

← Exemplo:

Neste caso, os instantes “2” e “7” podem estar liberando\* certas operações num circuito combinacional (ou sequencial) muito maior.

\* gerando sinais de “Enable” para outras partes de um circuito maior.



Simple contador crescente síncrono

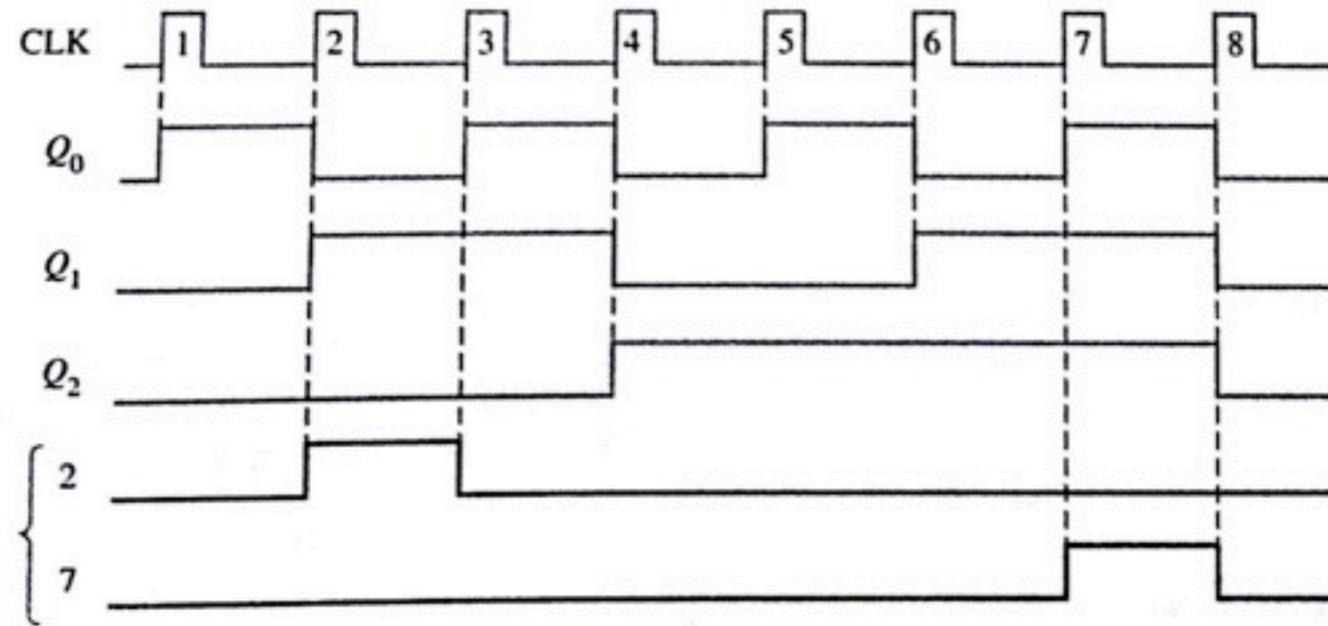
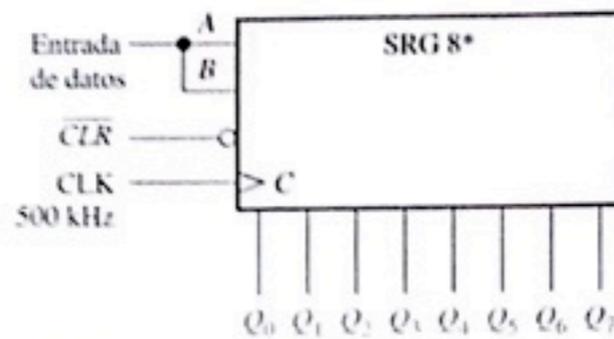


FIGURA 8.46 Contador de 3 bits con decodificación activa a nivel ALTO de los estados 2 y 7.

# Temporizar sinais



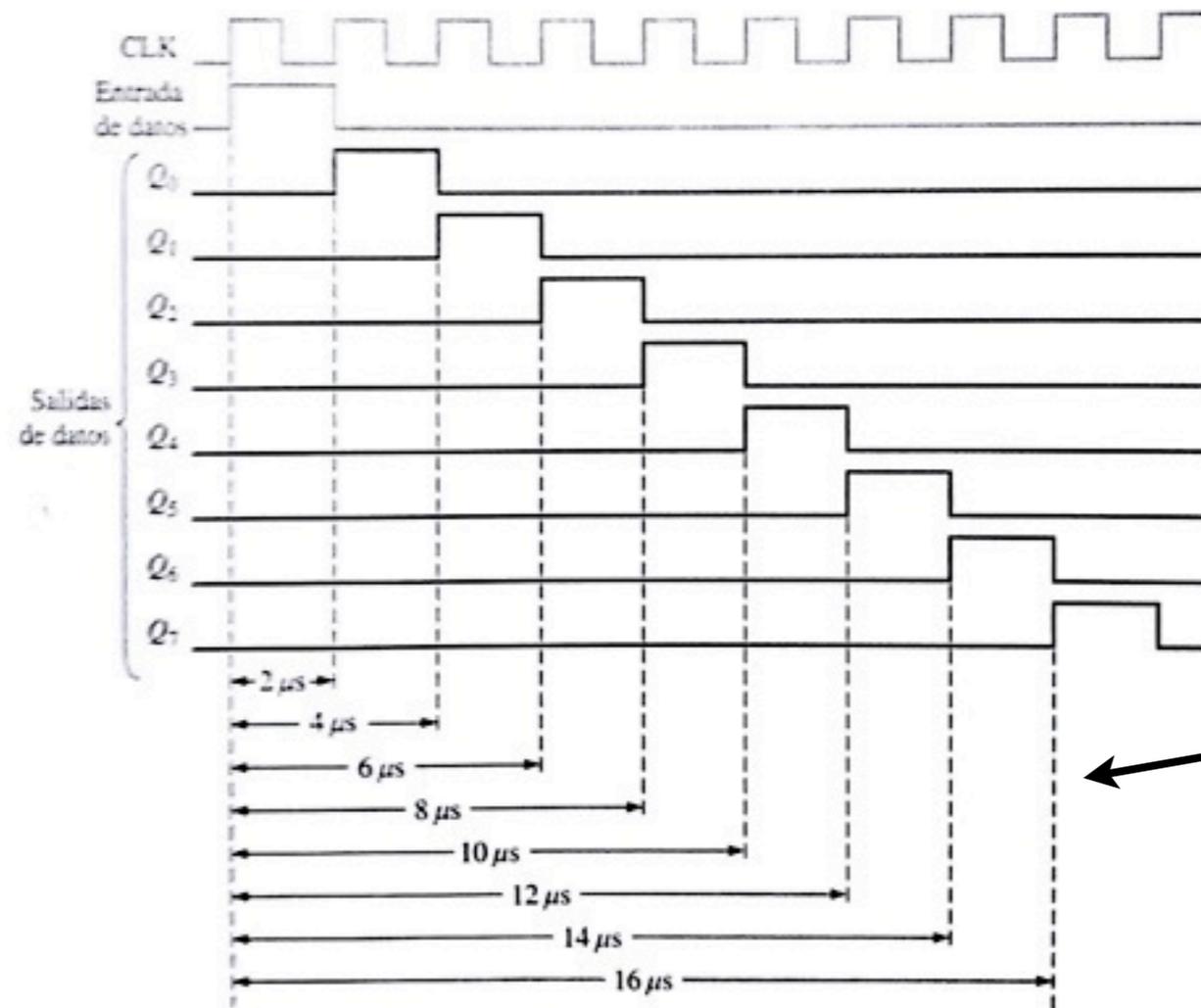
\* Los datos se desplazan de  $Q_0$  a  $Q_7$ .

FIGURA 9.29

Solución

El periodo de reloj es  $2 \mu\text{s}$ . Luego el retardo de tiempo puede incrementarse o decrementarse de dos en dos  $\mu\text{s}$ , desde un mínimo de  $2 \mu\text{s}$  hasta un máximo de  $16 \mu\text{s}$ , como ilustra la Figura 9.30.

← Ejemplo:



Neste caso, nas saídas do registrador é possível se obter pulsos que são gerados em diferentes instantes de tempo, conforme a saída selecionada do registrador.

FIGURA 9.30 Diagrama de tiempos que muestra los retardos para el registro de la Figura 9.29.

# Exemplo/Problema:

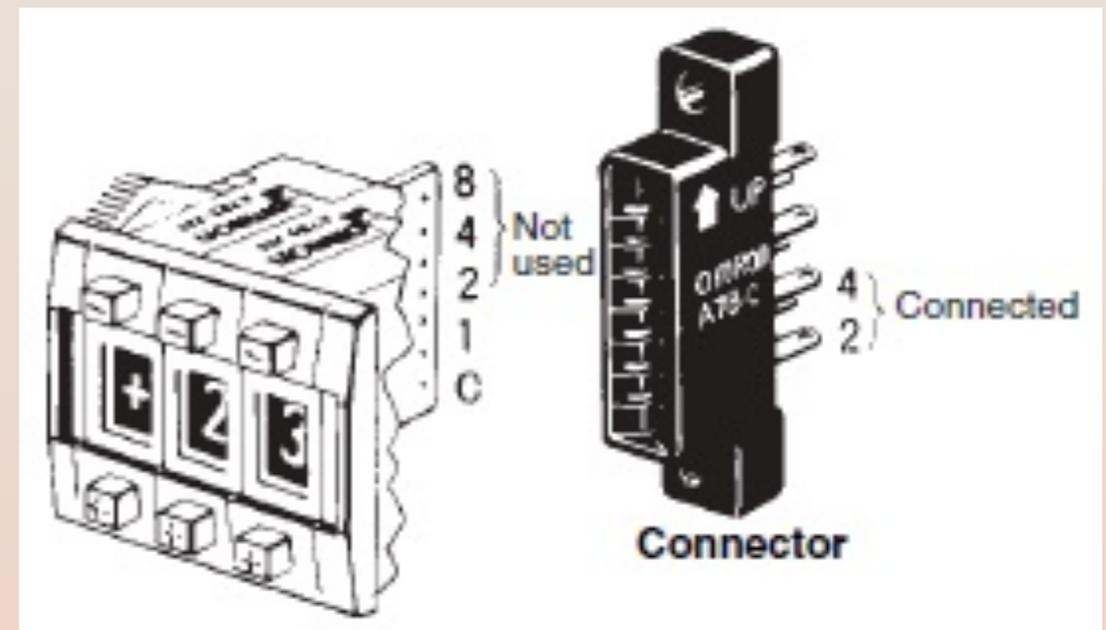
Criar um cronômetro programável capaz de contar de 99 à 0 segundos (decrecente), onde:

- o usuário pode programar (“setar”) o valor inicial;
- usuário usa um botão para dar partida no cronômetro (“Start”);
- este mesmo botão permite pausar a contagem (“Pause”) e
- onde existe um outro botão para reinicializar todo o sistema (“Restart”).

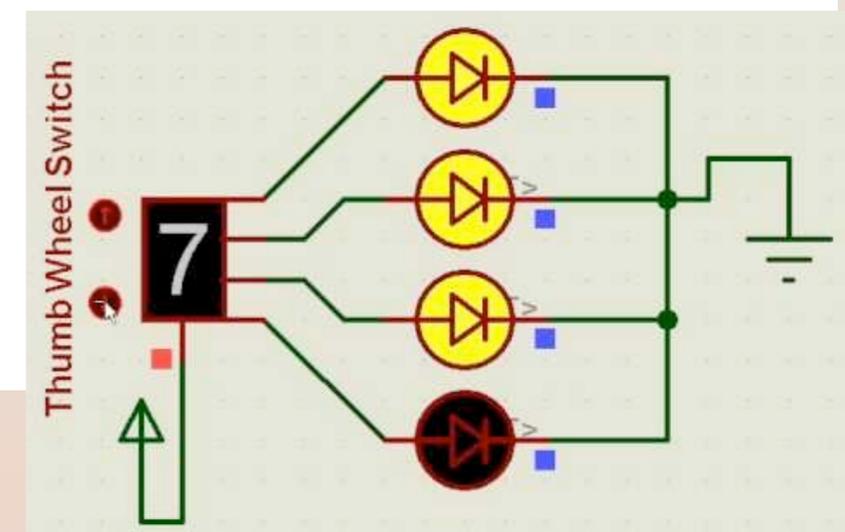
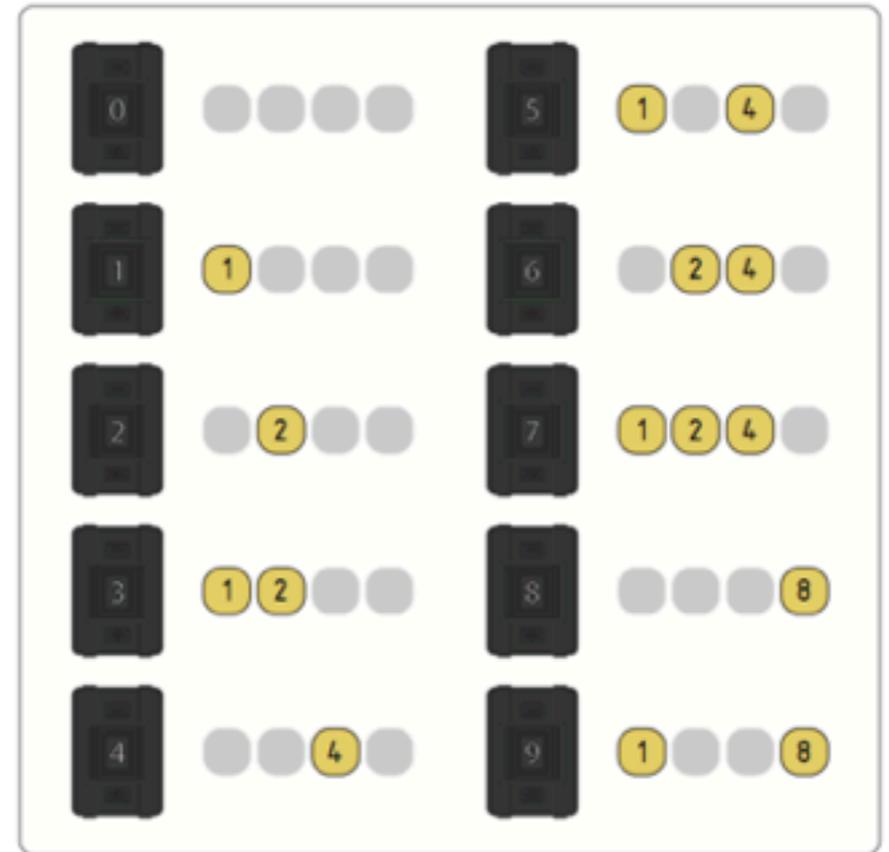
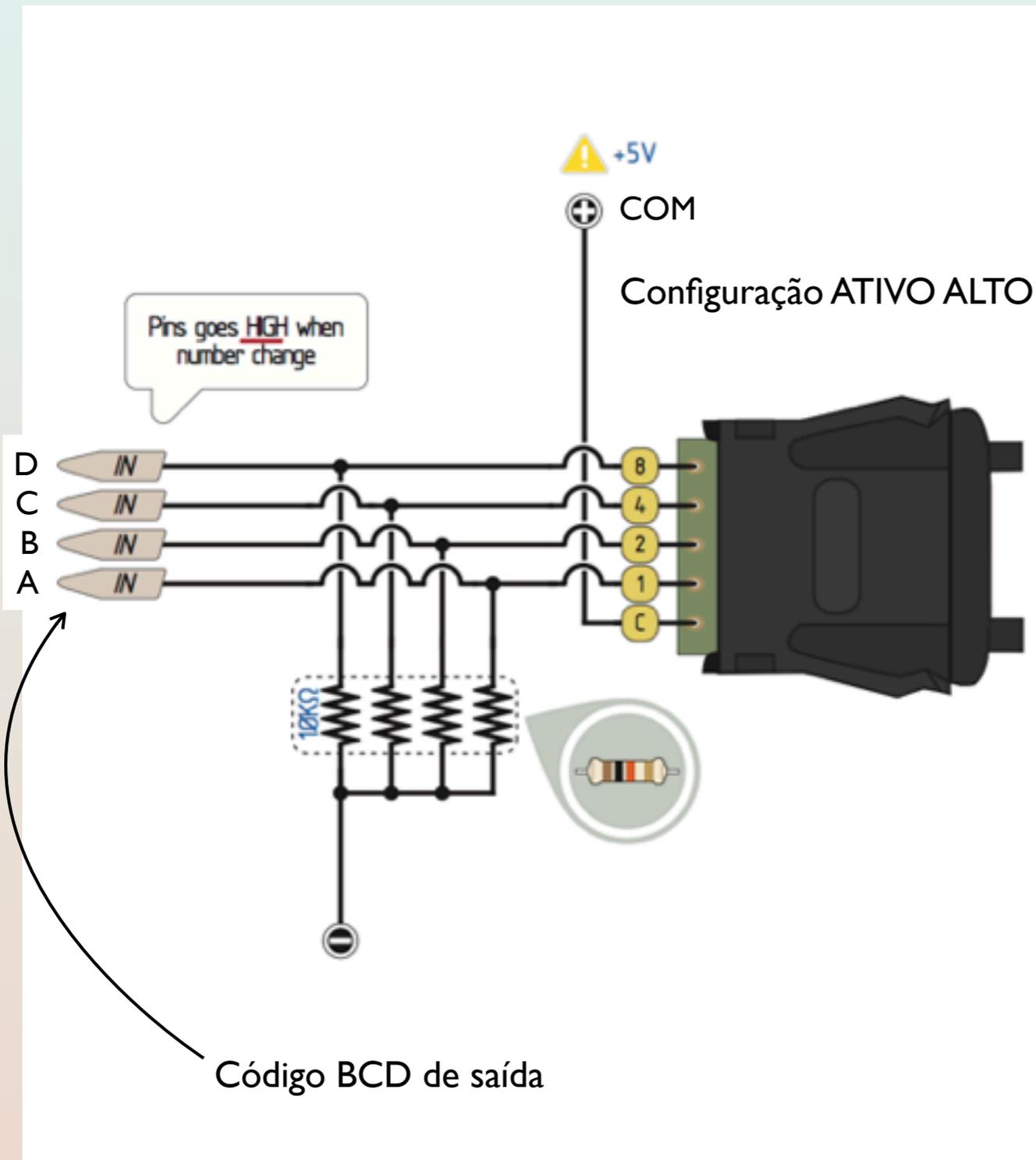


Soluções/Opções iniciais:

- Uso de CIs contadores do tipo 74190 (contadores síncronos up/down, programáveis);
- Uso de “thumb-switches” (SWITCH THUMB BCD) que possibilitam que usuário programe o valor de início da contagem:



# Thumbwheel switch - conexão...



<https://www.youtube.com/watch?v=YW6X6a643fk>

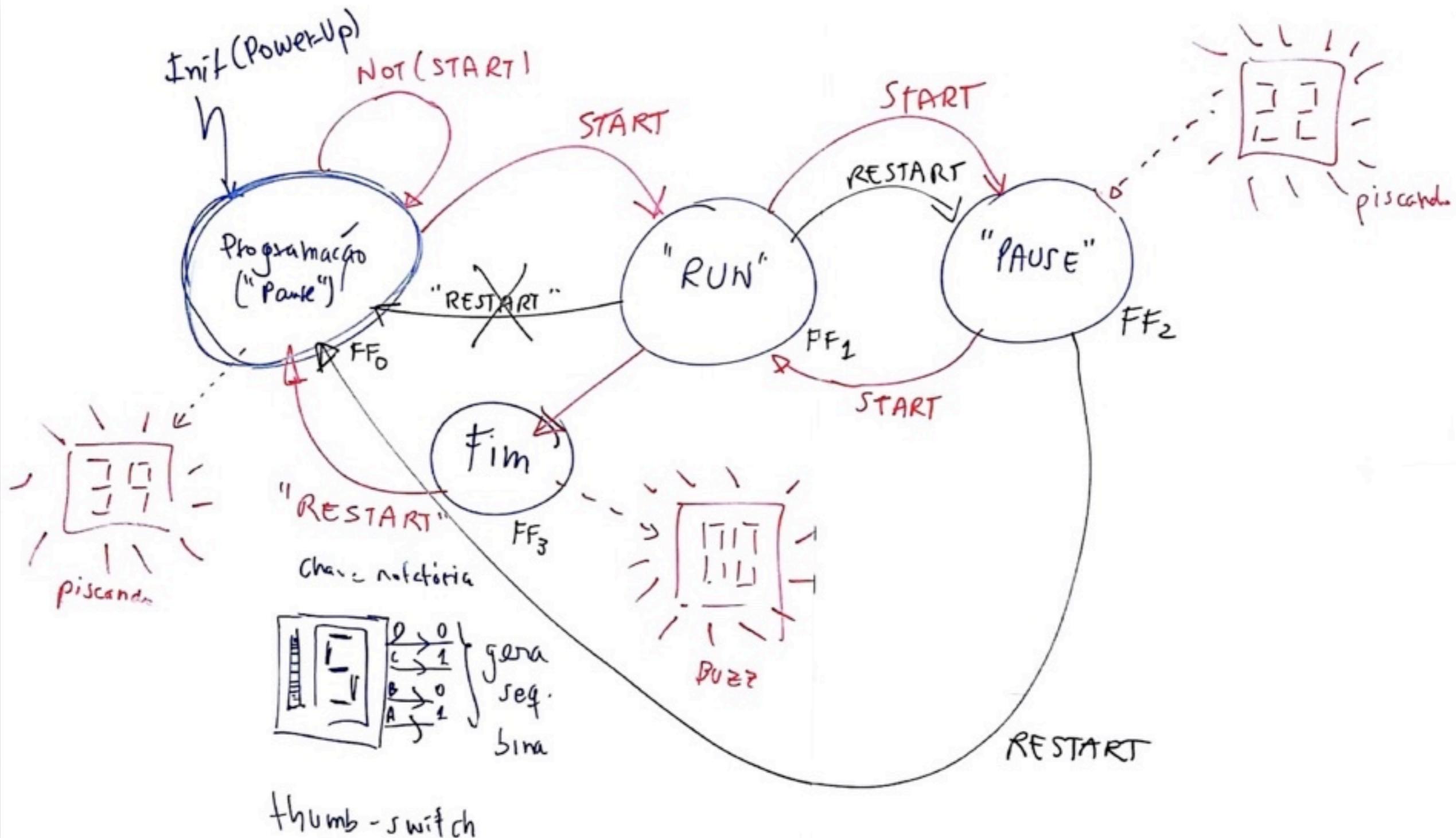
# Exemplo/Problema:

Criar um cronômetro programável capaz de contar de 99 à 0 segundos (decrecente), onde:

- a) o usuário pode programar ("setar") o valor inicial;
- b) usuário usa um botão para dar partida no cronômetro ("Start");
- c) este mesmo botão permite pausar a contagem ("Pause") e
- d) onde existe um outro botão para reinicializar todo o sistema ("Restart").

Soluções:

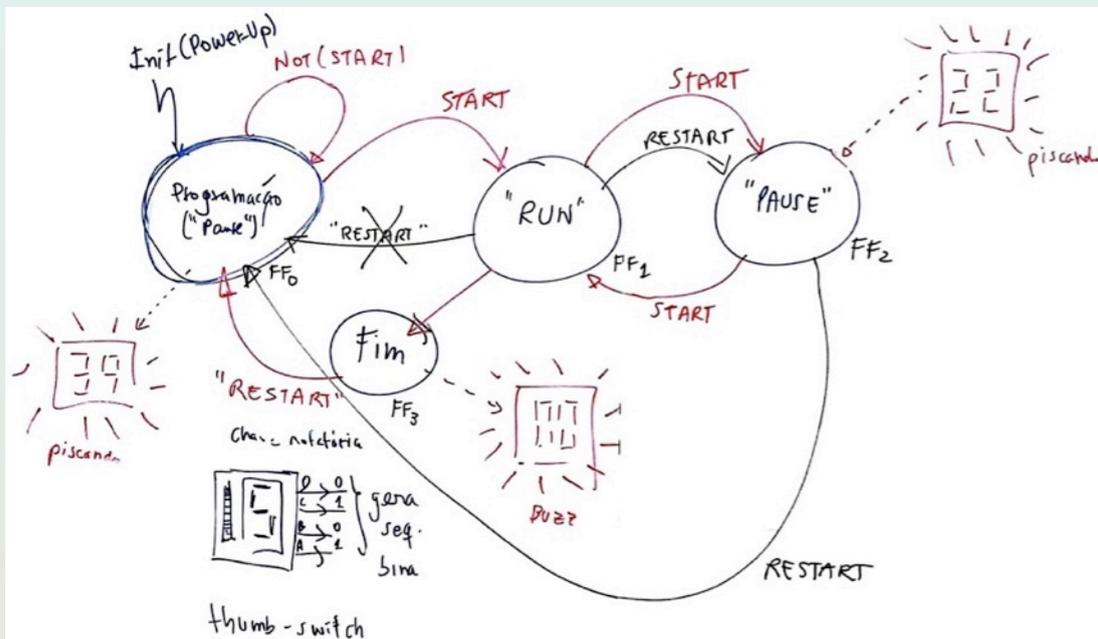
1) Idealizar um diagrama de estados para o circuito:



# Exemplo/Problema:

Criar um cronômetro programável capaz de contar de 99 à 0 segundos (decrecente), onde:

- o usuário pode programar ("setar") o valor inicial;
- usuário usa um botão para dar partida no cronômetro ("Start");
- este mesmo botão permite pausar a contagem ("Pause") e
- onde existe um outro botão para reinicializar todo o sistema ("Restart").



Soluções:

2) Decidir componentes/opções de projeto:

2ª Etapa: Decidir componentes.

- 2 x (5) 74190 (Sync, Bid, PL)

- opções

nº do estados do circuit.

a) Uso 4 FF's  $\Rightarrow$  1 FF para cada estado ou

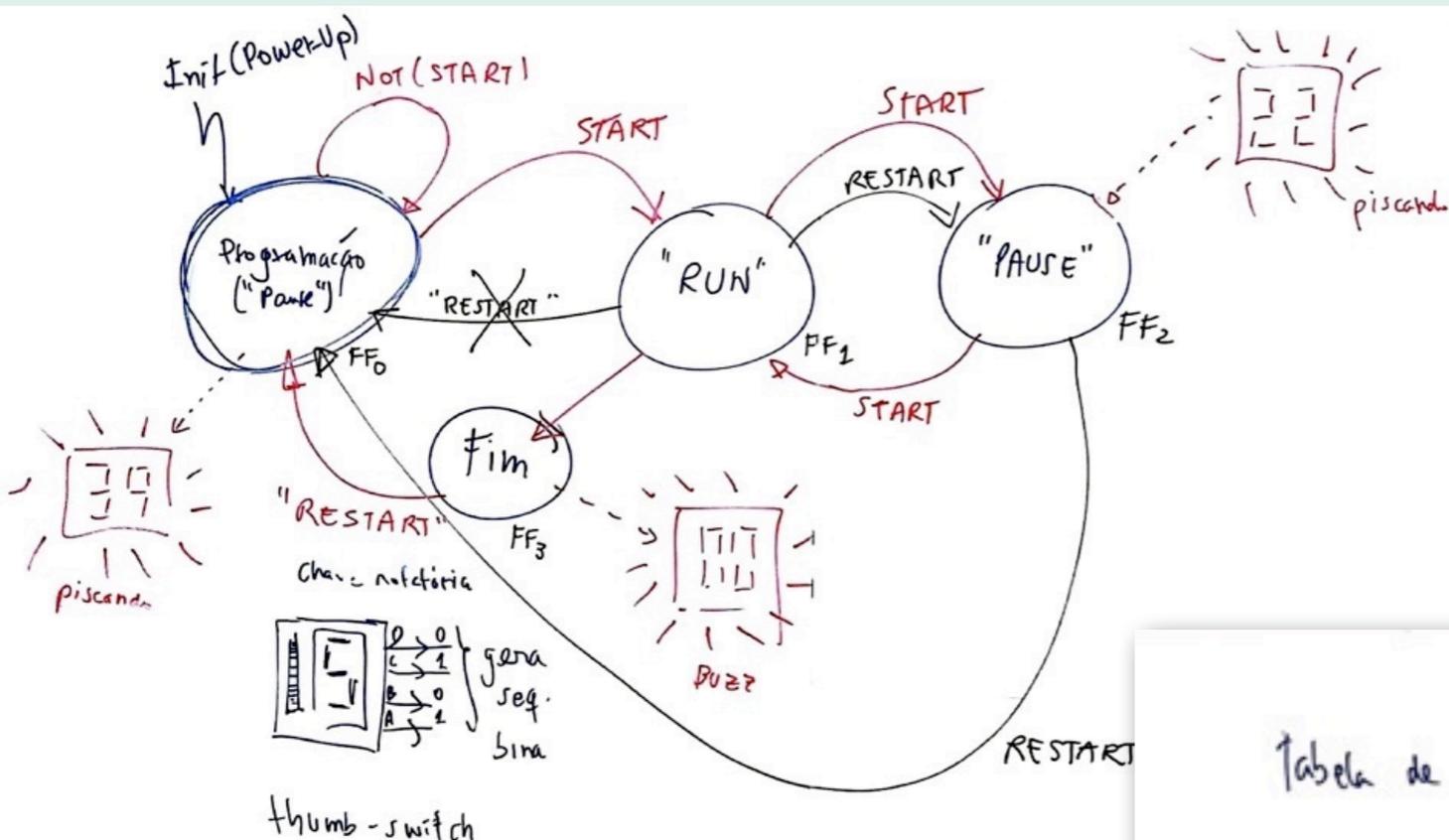
b) Uso 2 FF's  $\Rightarrow$  Contador (sync) contar de 0 à 3 (MOD-4)

$\Rightarrow$  Fica implícito o provável uso do DEC na saída deste contador (para identificação dos estados)

Necessário p/gerar as linhas internas de Controle (lógica combinacional).

# Exemplo/Problema:

Criar um cronômetro programável capaz de contar de 99 à 0 segundos (decrecente), onde:



e programar ("setar") o valor inicial;  
 um botão para dar partida no cronômetro ("Start");  
 um botão permite pausar a contagem ("Pause") e  
 um outro botão para reinicializar todo o sistema ("Restart").

Soluções:

3) Deduzir tabela transição do circuito completo:

Tabela de transição:

Entradas (Ext.) de Controle	Estado atual				Próx. estado				Observação / Comentário
	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	
"Init"	1	0	0	0	1	0	0	0	"Not start" ("Parallel Load = PL")
START=1	1	0	0	0	0	1	0	0	"PL" → "RUN"
START=1	0	1	0	0	0	0	1	0	"RUN" → "PAUSE"
START=1	0	0	1	0	0	1	0	0	"PAUSE" → "RUN"
RESTART=1	0	1	0	0	0	0	1	0	"RUN" → "PAUSE"
RESTART=1	0	0	1	0	1	0	0	0	"PAUSE" → "PL"
"TC" do 74190	0	1	0	0	0	0	0	1	"RUN" → fim
RESTART=1	0	0	0	1	1	0	0	0	fim → "PL"

Terminal Count

# Exemplo/Problema:

Soluções:

4) **Levantar equações** para as entradas de controle dos FF's usados para controlar os estados do circuito:

$$d_0 = q_2 \cdot RESTART + q_3 \cdot RESTART$$

$$d_0 = RESTART \cdot (q_2 + q_3)$$

$$d_1 = q_0 \cdot START + q_2 \cdot START$$

$$d_1 = START \cdot (q_0 + q_2)$$

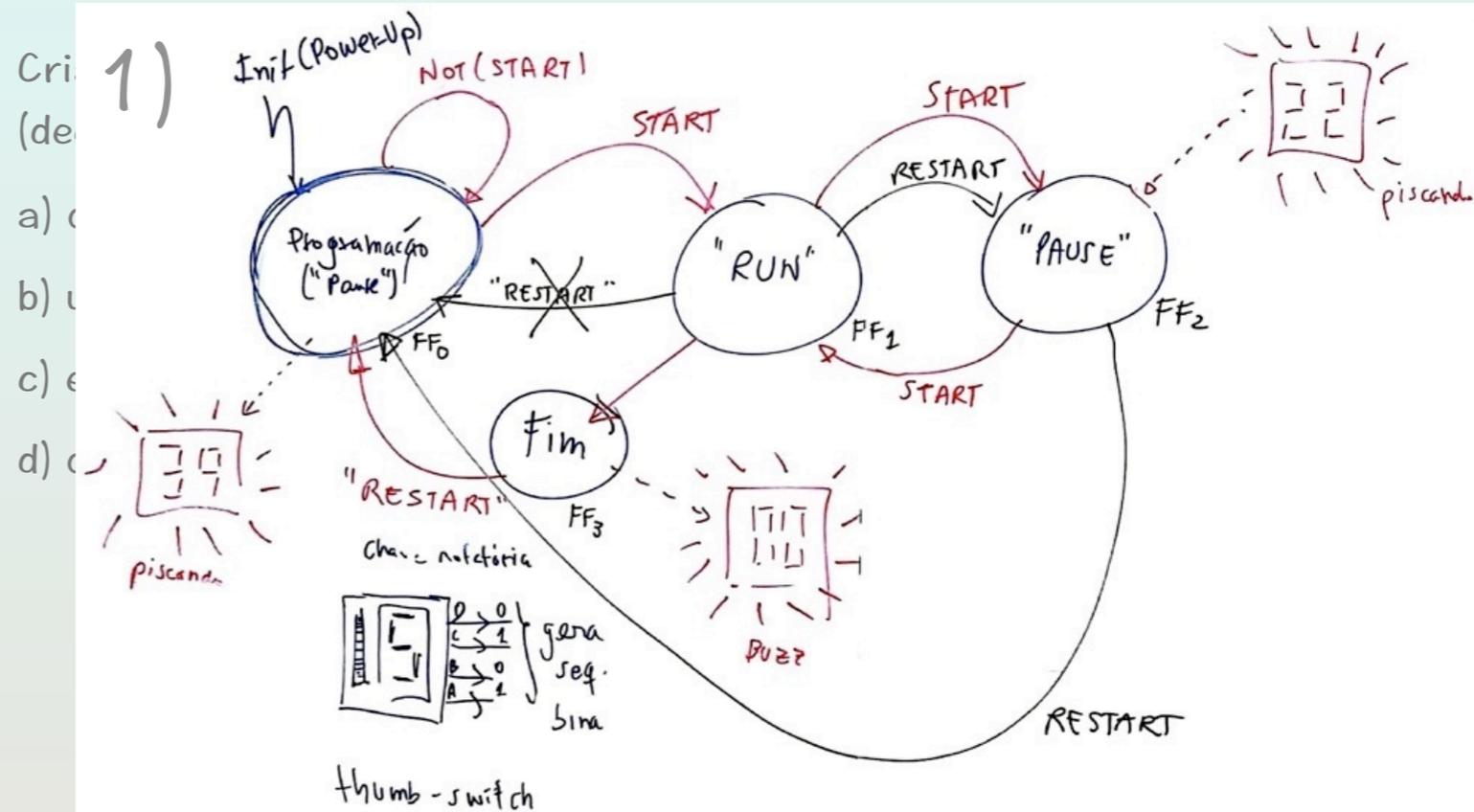
$$d_2 = q_1 \cdot START + q_1 \cdot RESTART$$

$$d_2 = q_1 \cdot (START + RESTART)$$

$$d_3 = (TC_{(CI\ 74190-Dezenas)}) \cdot (q_1 + q_2)$$

Notar que:

$$Init = Power - Up \rightarrow \begin{cases} Q_0 = 1 \\ Q_1 = 0 \\ Q_2 = 0 \\ Q_3 = 0 \end{cases}$$



3) Tabela de transição:

Entradas (Ext.) de Controle	Estado atual				Prox. estado				Observação / Comentar
	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	
"Init"	1	0	0	0	1	0	0	0	"Not start" ("Parallel Load = PL")
START=1	1	0	0	0	0	1	0	0	"PL" → "RUN"
START=1	0	1	0	0	0	0	1	0	"RUN" → START → "PAUSE"
START=1	0	0	1	0	0	1	0	0	"PAUSE" → START → "RUN"
RESTART=1	0	1	0	0	0	0	1	0	"RUN" → RESTART → "PAUSE"
RESTART=1	0	0	1	0	1	0	0	0	"PAUSE" → RESTART → "PL"
TC do 74190	0	1	0	0	0	0	0	1	"RUN" → FIM
RESTART=1	0	0	0	1	1	0	0	0	FIM → RESTART → "PL"

Terminal Count

Notar que as saídas dos FF's geram sinais de controle internos:

$Q_0=1 \Rightarrow$  "Parallel Load (74190's) + piscar display;

$Q_1=1 \Rightarrow$  Count Enable (CE) dos 74190's;

$Q_2=1 \Rightarrow$  Disable Count Enable dos 74190's + piscar display;

$Q_3=1 \Rightarrow$  Disable Count Enable dos 74190's + piscar display;

#### 4) Equações:

$$d_0 = q_2 \cdot RESTART + q_3 \cdot RESTART$$

$$d_0 = RESTART \cdot (q_2 + q_3)$$

$$d_1 = q_0 \cdot START + q_2 \cdot RESTART$$

$$d_1 = START \cdot (q_0 + q_2)$$

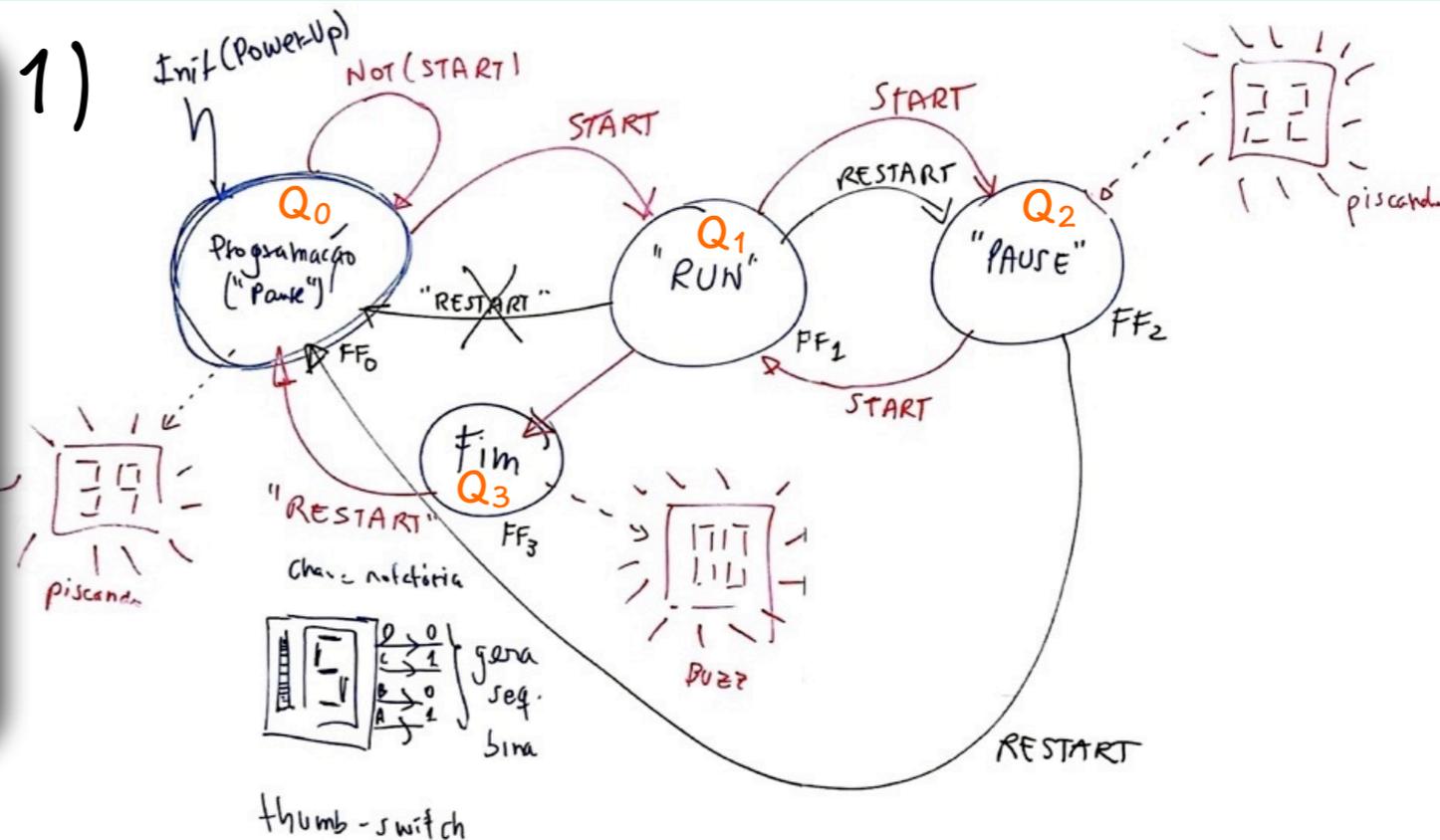
$$d_2 = q_1 \cdot START + q_1 \cdot RESTART$$

$$d_2 = q_1 \cdot (START + RESTART)$$

$$d_3 = (TC_{(CI\ 74190-Dezenas)}) \cdot (q_1 + q_2)$$

Notar que:

$$Init = Power - Up \rightarrow \begin{cases} Q_0 = 1 \\ Q_1 = 0 \\ Q_2 = 0 \\ Q_3 = 0 \end{cases}$$



3) Tabela de transição:

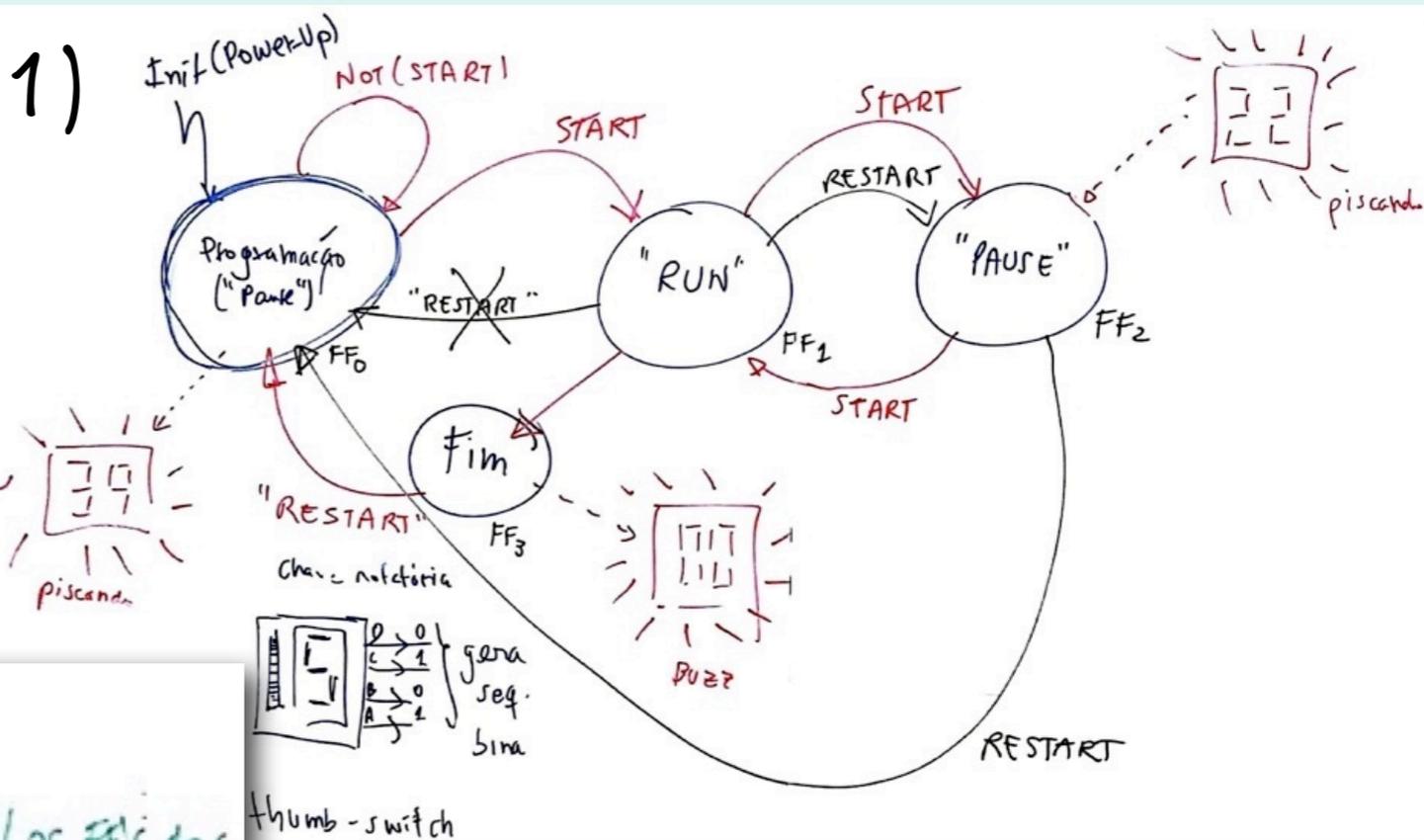
Entradas (Ext.) de Controle	Estado atual				Prox. estado				Observação / Comentar
	$q_0$	$q_1$	$q_2$	$q_3$	$Q_0$	$Q_1$	$Q_2$	$Q_3$	
"Init"	1	0	0	0	1	0	0	0	"Not start" ("Parallel Load = PL")
$START=1$	1	0	0	0	0	1	0	0	"PL" $\rightarrow$ "RUN"
$START=1$	0	1	0	0	0	0	1	0	"RUN" $\xrightarrow{START}$ "PAUSE"
$START=1$	0	0	1	0	0	1	0	0	"PAUSE" $\xrightarrow{START}$ "RUN"
$RESTART=1$	0	1	0	0	0	0	1	0	"RUN" $\xrightarrow{RESTART}$ "PAUSE"
$RESTART=1$	0	0	1	0	1	0	0	0	"PAUSE" $\xrightarrow{RESTART}$ "PL"
TC do 74190	0	1	0	0	0	0	0	1	"RUN" $\rightarrow$ FIM
$RESTART=1$	0	0	0	1	1	0	0	0	FIM $\xrightarrow{RESTART}$ "PL"

Terminal Count



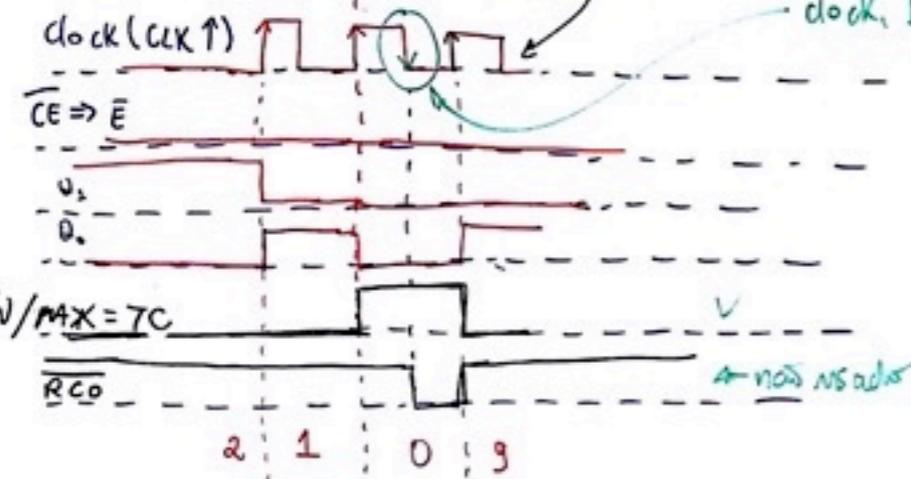
# Exemplo/Problema:

Cri  
(de  
a) c  
b) u  
c) e  
d) c



Detalhes do 74190:

Fim da contagem, modo decrescente.



"TC" do 74190	0100	0001	"RUN" → FIM
RESTART=1	0001	1000	FIM → RESTART → "PL"

Terminal Count

4) Eqs:

$$d_0 = q_2 \cdot RESTART + q_3 \cdot RESTART$$

$$d_0 = RESTART \cdot (q_2 + q_3)$$

$$d_1 = q_0 \cdot START + q_2 \cdot START$$

$$d_1 = START \cdot (q_0 + q_2)$$

$$d_2 = q_1 \cdot START + q_1 \cdot RESTART$$

$$d_2 = q_1 \cdot (START + RESTART)$$

$$d_3 = (TC_{(CI\ 74190 - Dezenas)}) \cdot (q_1 + q_2)$$

$$Init = Power - Up \rightarrow \begin{cases} Q_0 = 1 \\ Q_1 = 0 \\ Q_2 = 0 \\ Q_3 = 0 \end{cases}$$

Parte Final...

6) Diagrama Elétrico...