

# PCS 3115 – Sistemas Digitais I

## Circuitos Sequenciais: Registradores e Contadores

### EAD – Ensino A Distância

#### Parte IV: Contadores Síncronos – Exercícios.

Aula: 22 – Data: 08/06 (S)

*Prof. Dr. Marco Túlio Carvalho de Andrade*

*versão: 2.0 (Maio/2020)*

# Contadores

- **Conceito** – Grupos de Flip-Flops acionados por sinal comum de *clock*, que permanecem mudando de estado de acordo com uma sequência pré-estabelecida:
  - Dispensa, às vezes, outros tipos de sinais de entrada – Máquina de *Moore* (só o sinal de clock como uma pseudo-entrada)
  - Basicamente – Circuito Sequencial síncrono que possui uma sequência principal de **transição de estados cíclica, pré-definida.**

# Contadores

## Funções & Usos

- Contagens diversas;
- Geração de sequência de sinais de controle;
- **Divisão de frequência;**
- Medição de frequência;
- **Geração de formas de onda específicas (quadrada, por exemplo);**
- Conversão analógico-digital.

# Contadores

- **Síncronos** – Para todos os Flip-Flops:
  - ✓ O *clock* é o mesmo sinal físico;
  - ✓ As saídas (estados) são atualizadas no mesmo instante (mesma borda de subida ou descida).

## Contadores Síncronos

- Divisor da frequência do *clock* pelo módulo ( $M =$  número de Estados) de sua sequência de contagem – Sinal disponível na saída mais significativa do contador  $Q_n$ :

$$f_{SAÍDA} = (f_{CLOCK-ENTRADA})/M$$

- Se a sequência for contínua:

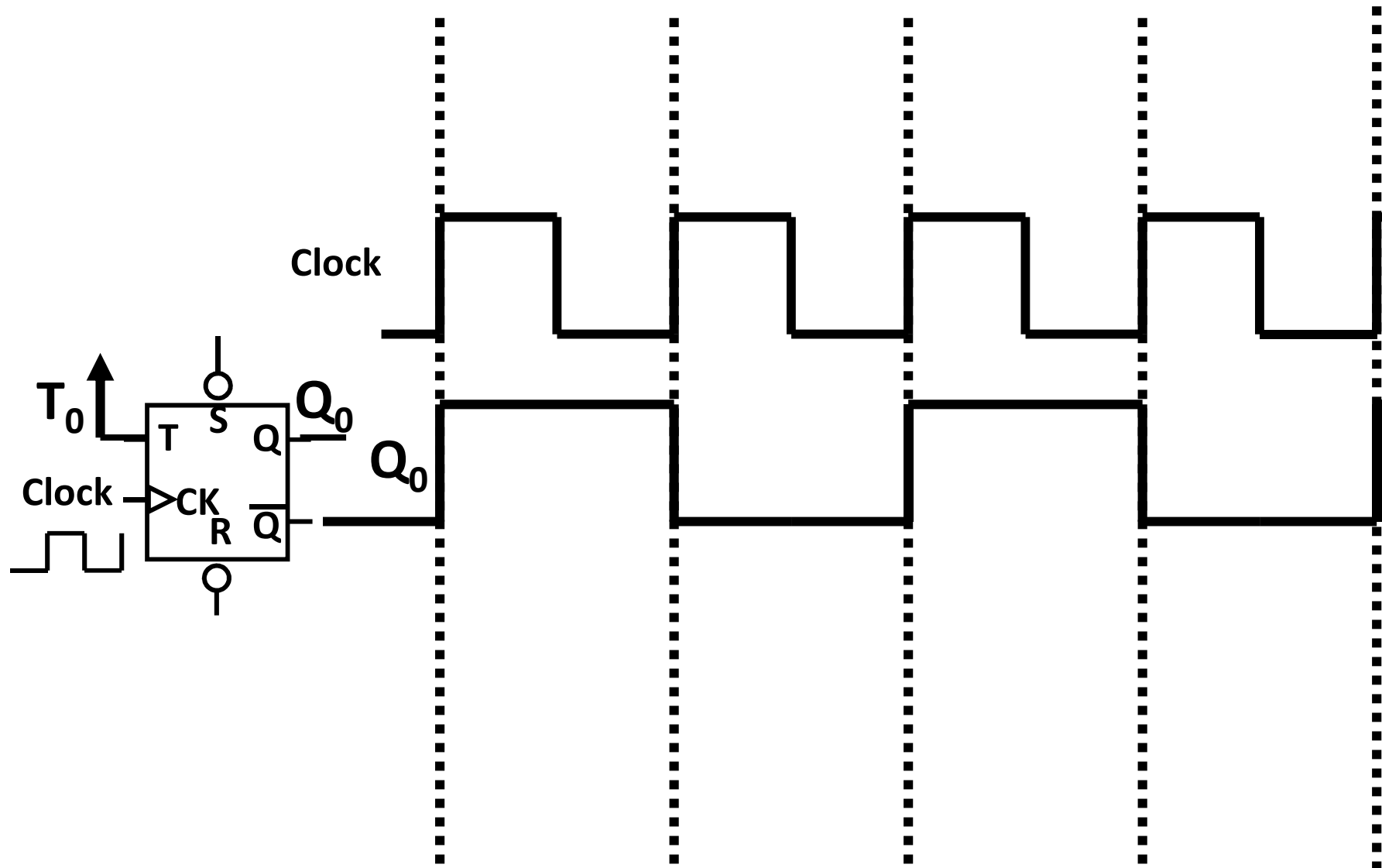
$$M = \text{Estado}_{FINAL} - \text{Estado}_{INICIAL} + 1$$

## Contadores Síncronos

- Contadores Síncronos – Alternativas de síntese.
- Alternativa 1 – Inspirado na síntese de contadores assíncronos.

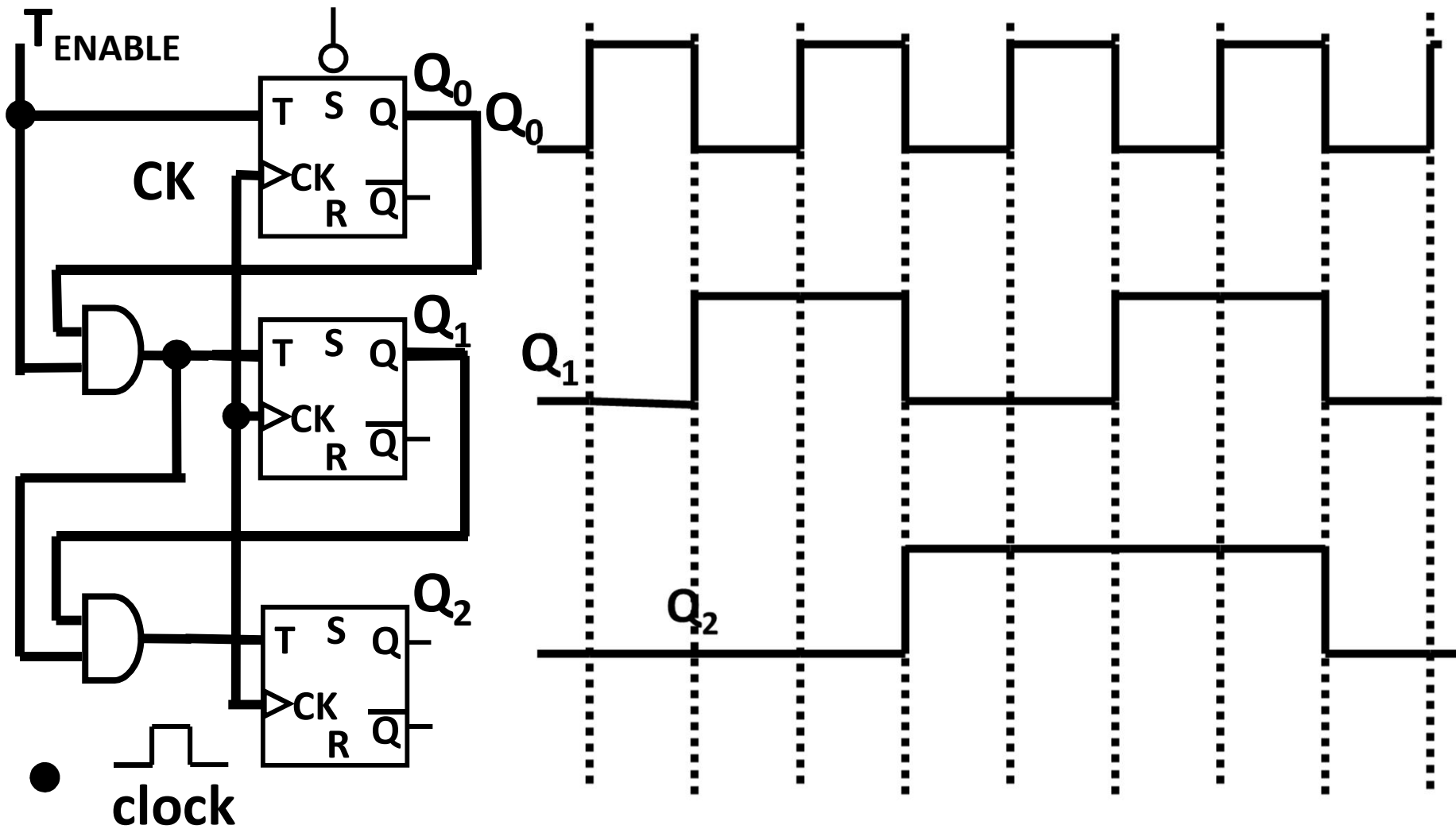
# Contadores Síncronos

## FF tipo T – Configuração divisor por 2 – Onda quadrada



# Contador Síncrono – Divisor por 8 (Saída em $Q_2$ , onda quadrada)

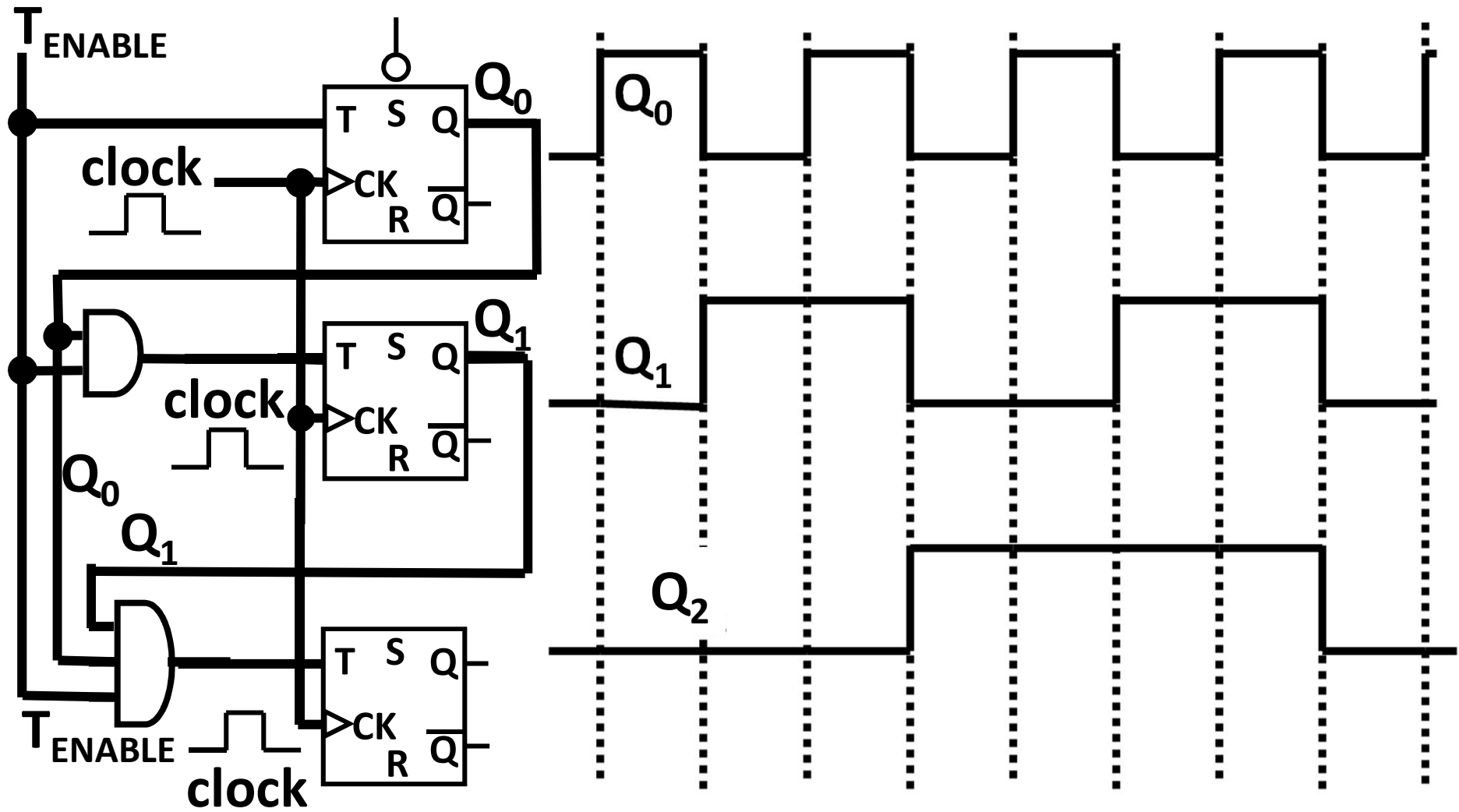
## Lógica de *ENABLE* serial – Mais lento





# Contador Síncrono – Divisor por 8 (Saída em $Q_2$ , onda quadrada)

## Lógica de *ENABLE* paralela – Mais rápido

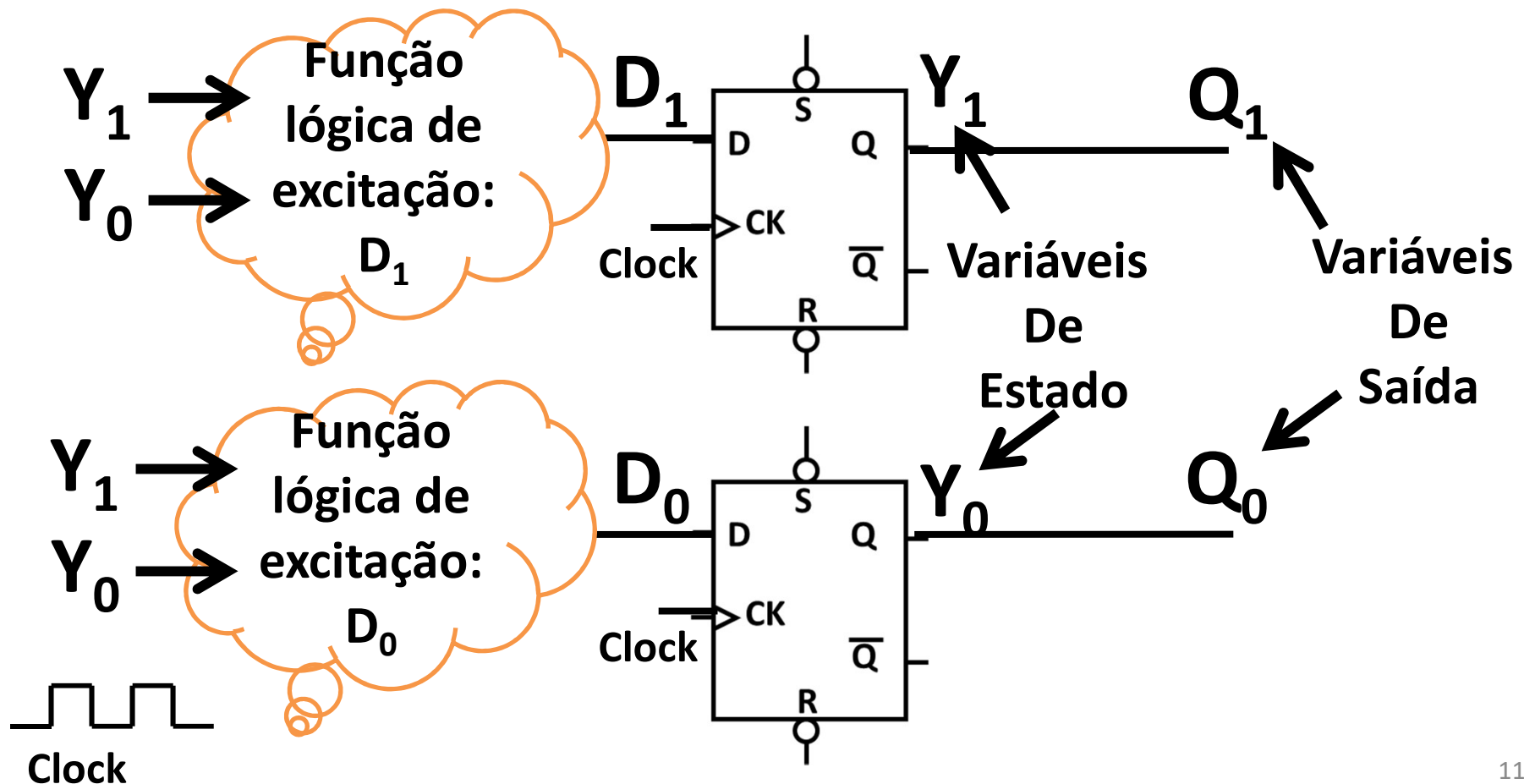


## Contadores Síncronos

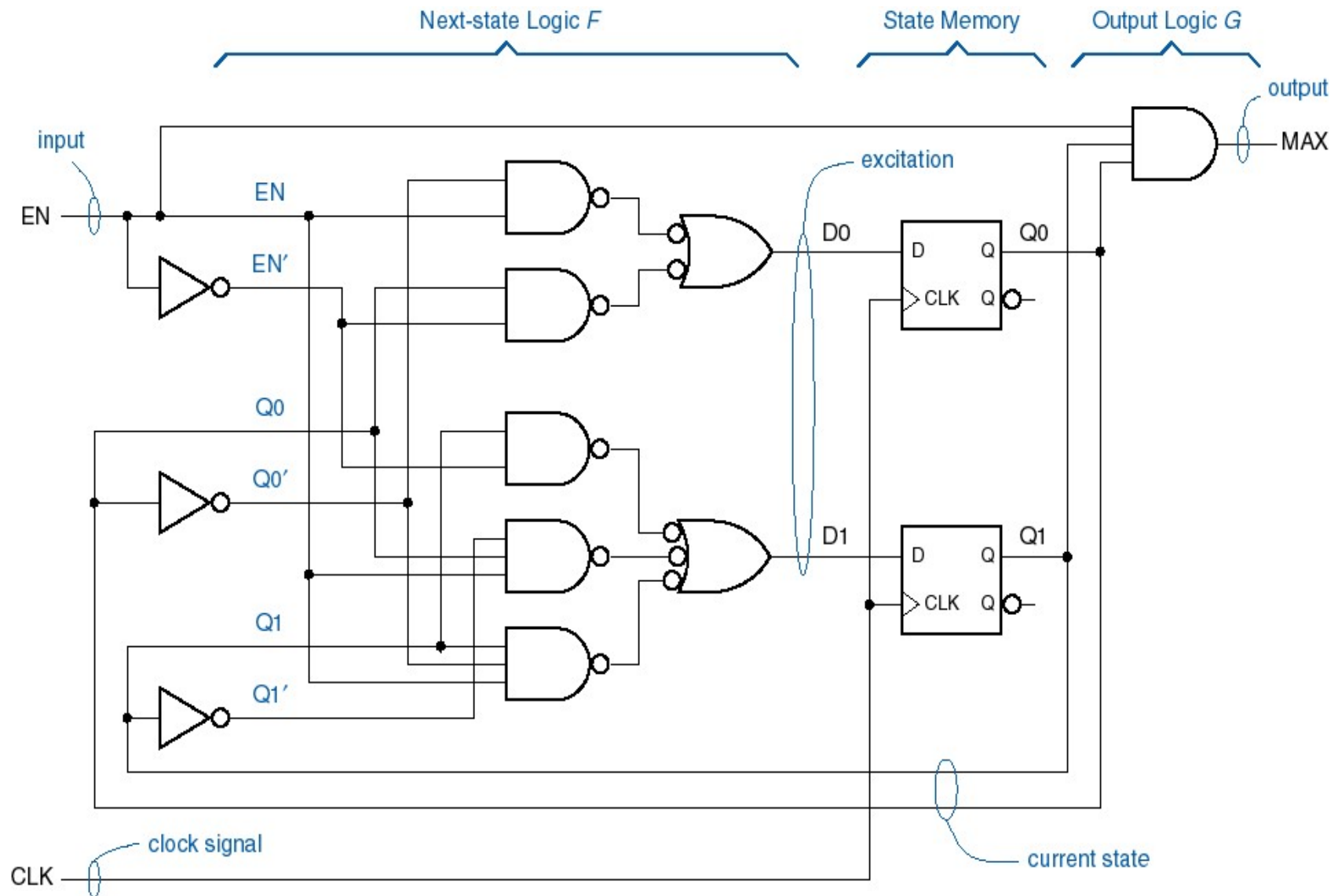
- Contadores Síncronos – Alternativas de síntese.
- Alternativa 2 – Inspirado na síntese de FSMs.

# Contadores Síncronos

- *Finite State Machines* – FSMs – Representação possível.



# 1. Circuitos Lógicos Seqüenciais



## Contadores Síncronos

- Contadores Síncronos – Alternativas de síntese.
- Alternativa 3 – Inspirado na síntese a partir de módulos de contadores síncronos prontos, utilizados como blocos elementares de projeto (como fizemos com blocos de circuitos combinatórios).

## Contadores – Geração de Onda Quadrada

- **Problema** – Gerar onda quadrada com frequência menor do que clock original
- Por que quadrada?
  - Reduz nível DC do sinal
    - Transmissão de sinais – Diminui perdas;
    - Conversão digital-analógico – Previne saturação de capacitores;
    - Aplicações diversas – Qualquer luz piscante em que o tempo total da luz apagada é igual ao tempo total da luz acesa ...

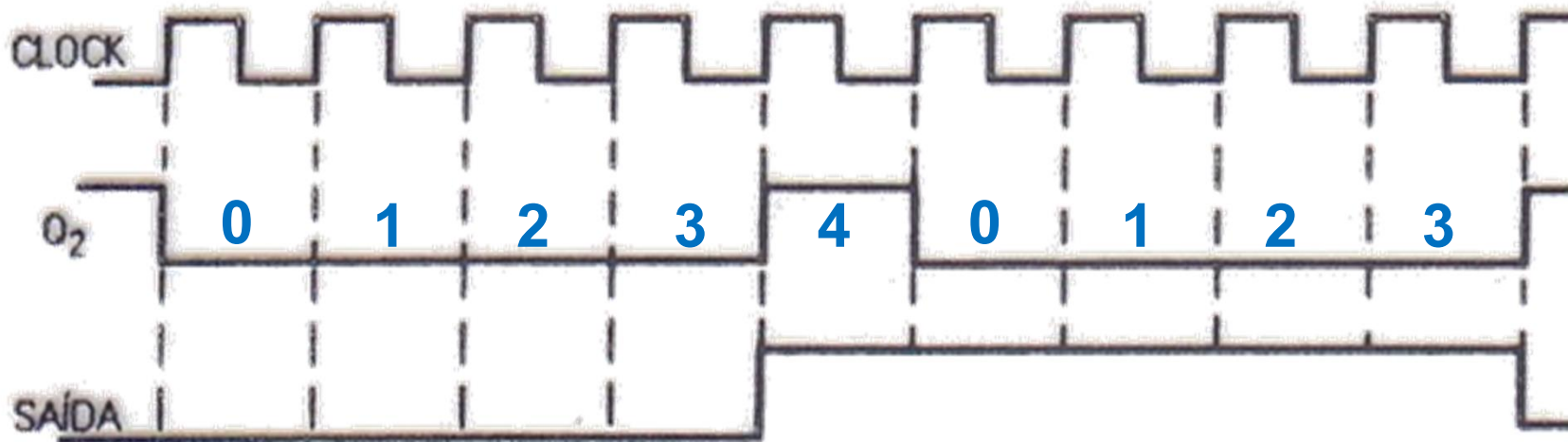
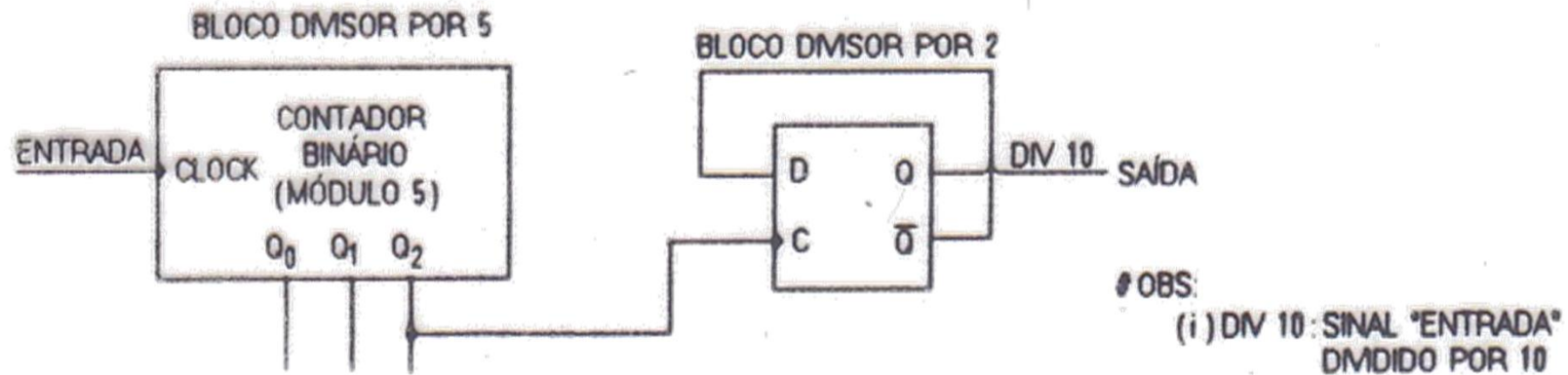
## Contadores – Geração de Onda Quadrada

- **Problema** – Gerar onda quadrada com frequência menor do que clock original.
- **Sub-problema 1** – Divisão por **número par**
- Solução possível – Associação já discutida...
  - Contador módulo  $n/2$ , que não precisa ser gerador de onda quadrada.
  - Saída mais significativa usada como *clock* para Flip-Flop em configuração divisor por 2.
    - Pode ser borda de subida ou descida

$$f_{\text{CK-SAÍDA}} = [f_{\text{CK-ENTRADA}} / (n/2)] \cdot 1/2$$

# Contadores – Geração de Onda Quadrada

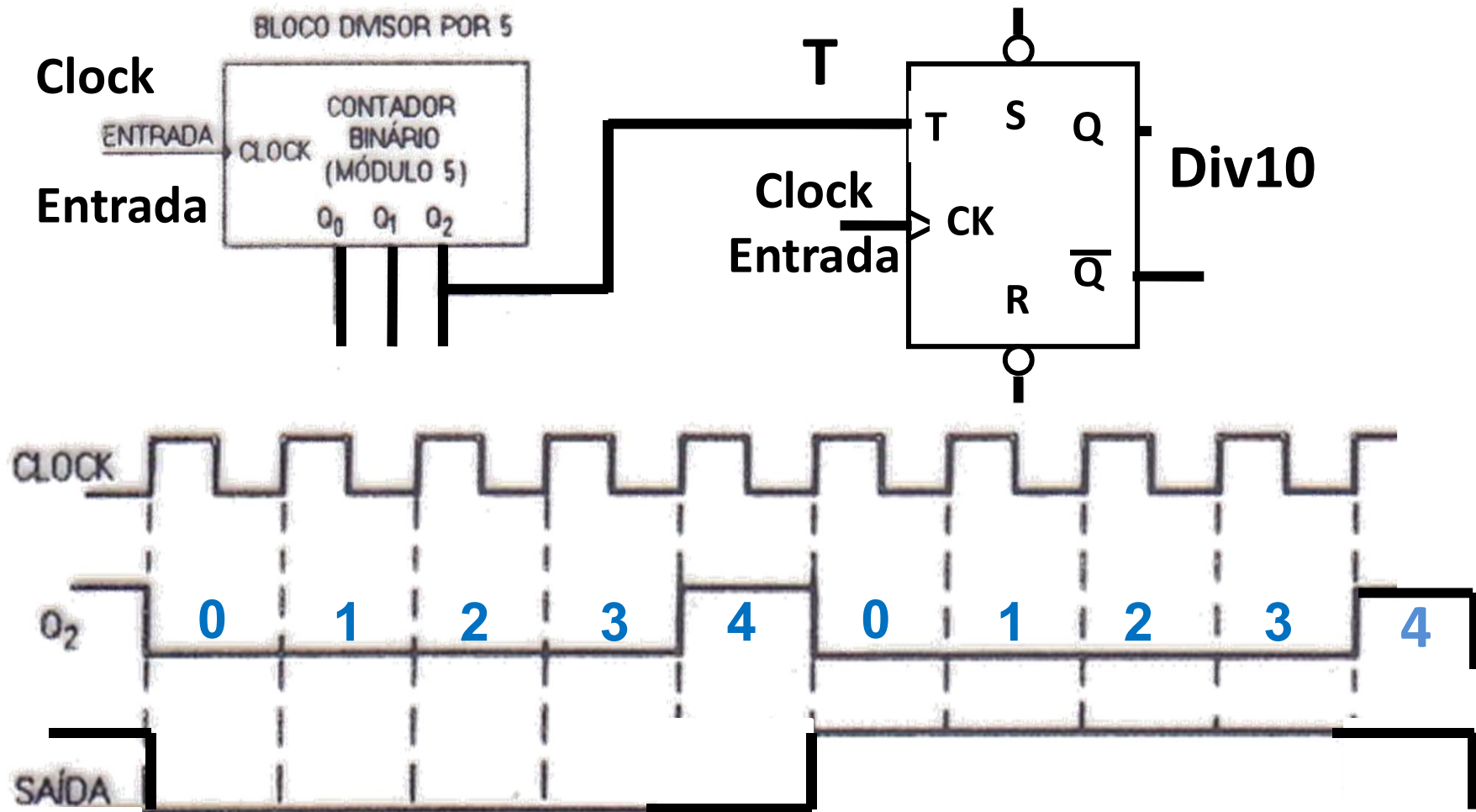
- Para divisão por número par:





# Contadores – Geração de Onda Quadrada

- Divisão por número par – Alternativa com sinal em fase:

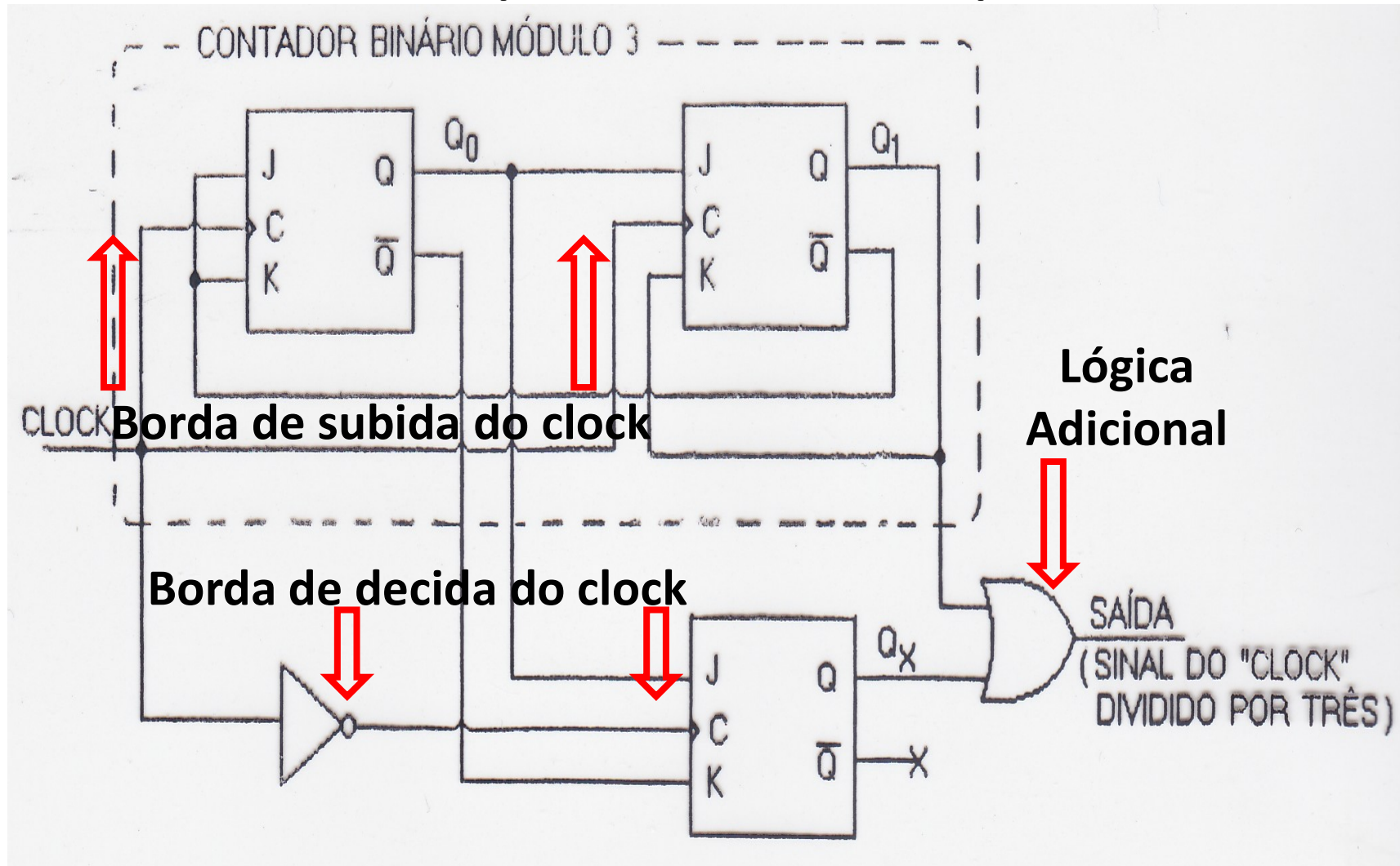


## Contadores – Geração de Onda Quadrada

- Para divisão por número ímpar
  - Sub-problema 2:
- São necessárias mudanças, dentro do contador, tanto na borda de subida como de descida, associadas a lógica adicional, para que se possa gerar a onda quadrada na saída.

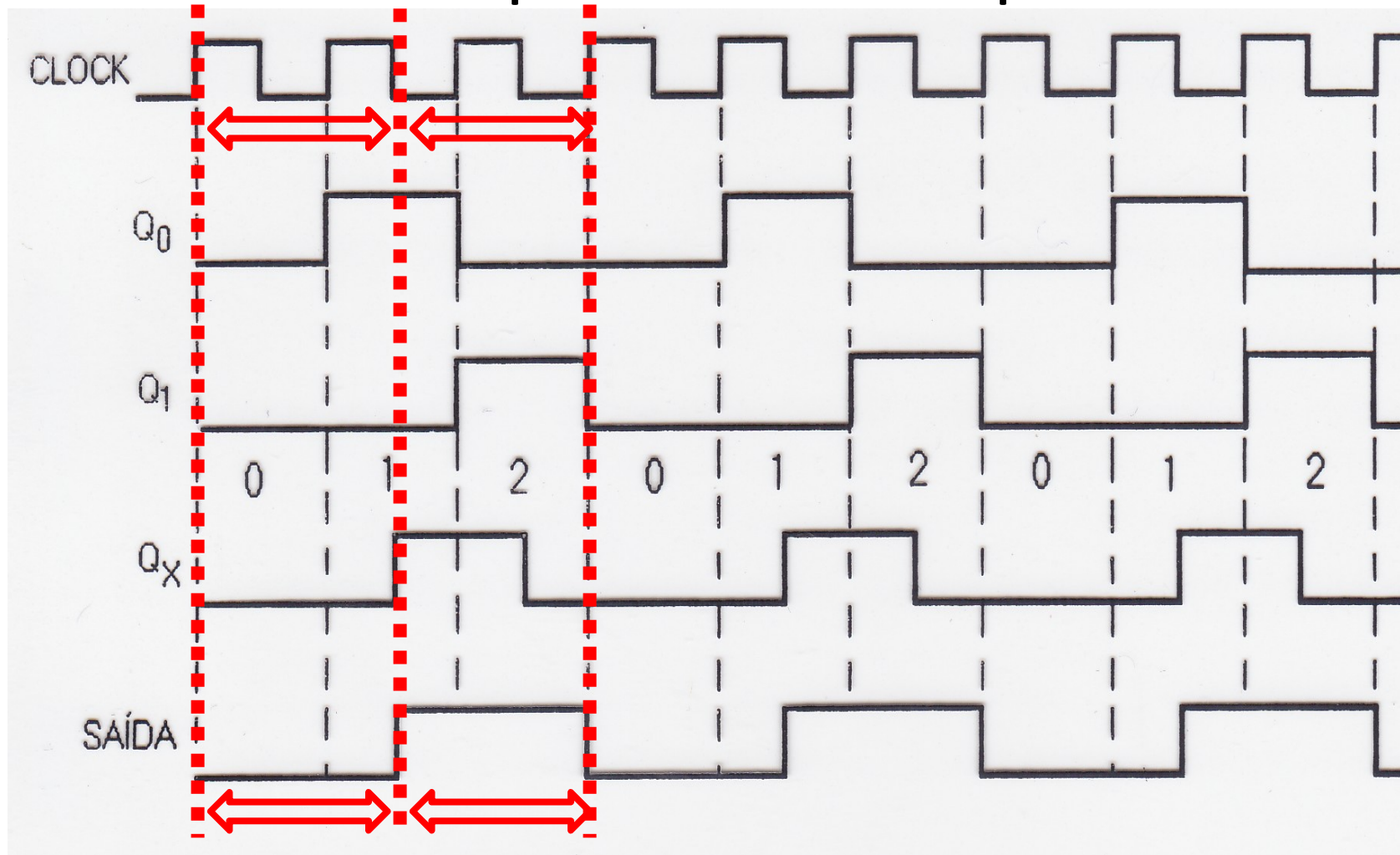
# Contadores – Geração de Onda Quadrada

- Para divisão por número ímpar:



## Contadores – Geração de Onda Quadrada

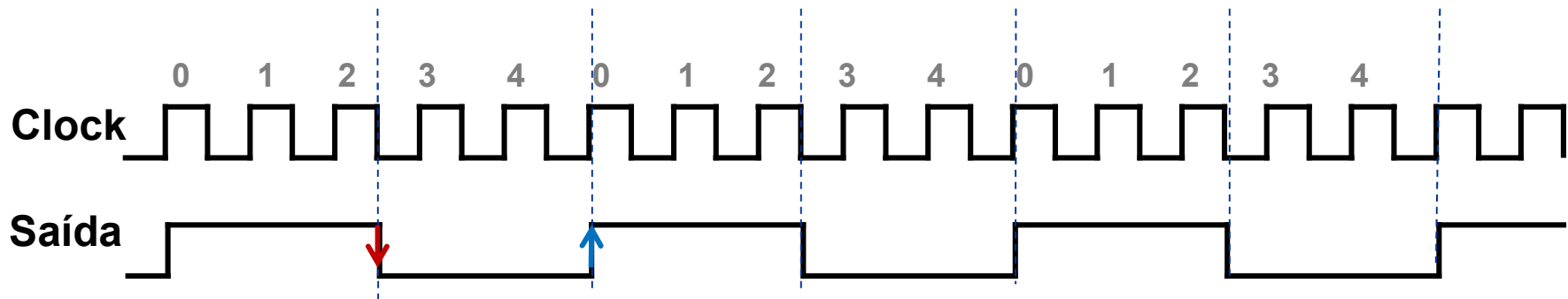
- Para divisão por número ímpar:



$$1 \frac{1}{2} T_{\text{clock}} + 1 \frac{1}{2} T_{\text{clock}} = 3 T_{\text{clock}}$$

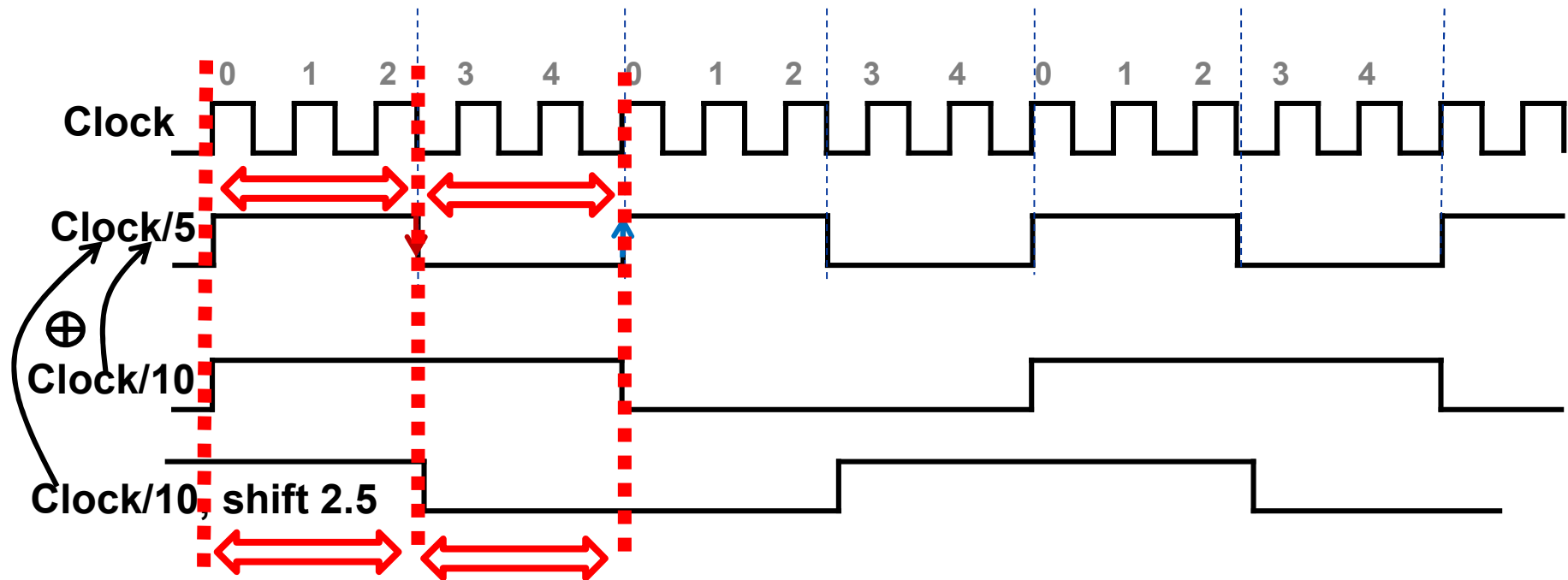
# Contadores – Geração de Onda Quadrada

- **Problema** – Gerar onda quadrada com frequência menor do que clock original
- **Sub-problema 2** – Divisão por **número ímpar**, mais um exemplo (divisão por 5):



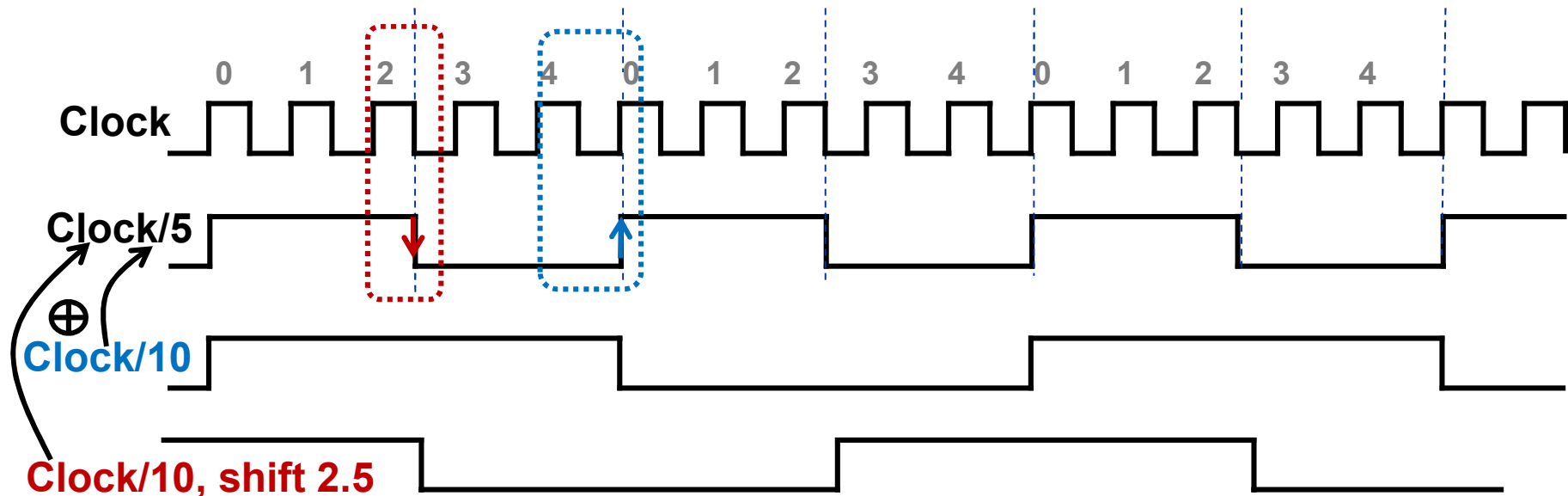
# Contadores – Geração de Onda Quadrada

- **Problema** – Onda quadrada, divisão por  $n = 5$  (ímpar).
- **Ideia** – Sabemos dividir por  $2n = 10$  com onda quadrada ( $5 \cdot 2$ ):
  - Podemos deslocar essa onda para sua transição aparecer após no “ $n/2 = 2.5$  clock”, nas bordas de descida;
  - Agora basta um XOR ou XNOR entre esses dois sinais!



# Contadores – Geração de Onda Quadrada

- **Problema:** onda quadrada, divisão por  $n = 5$  (ímpar)



- Ideia:
  - Sabemos dividir por  $2n=10$  com onda quadrada ( $5*2$ )
    - Basta fazer a inversão ao detectar  $n-1$ , com borda de subida
  - E podemos deslocar essa onda para sua transição aparecer após no “2,5 clock”, nas bordas de descida
    - Inverter Flip-Flop ao detectar  $n/2 = 2$  e for borda de descida

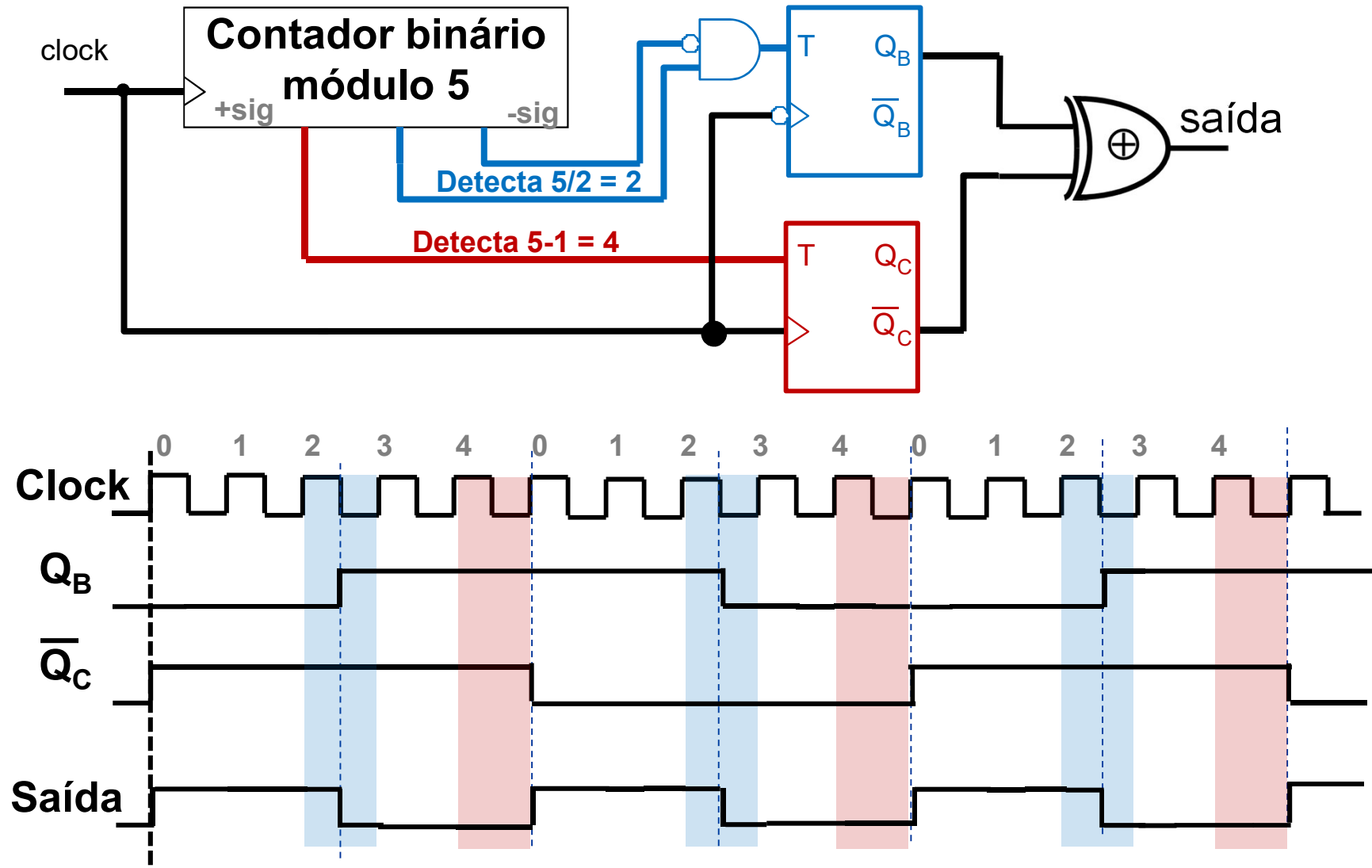
## Contadores – Geração de Onda Quadrada

- Uma estratégia para divisão por  $n$  ímpar:
  - **Bloco A**: um **contador módulo  $n$**  ativo na **borda de subida**
  - **Bloco B**: um **detector** ativado quando a saída do Bloco A for  $\lfloor n/2 \rfloor$ , ligado a um **Flip-Flop tipo T** ativo na **borda de descida**
  - **Bloco C**: um **detector** ativado quando a saída do Bloco A for  $n-1$ , ligado a um **Flip-Flop tipo T** ativo na **borda de subida**
  - **Saída**: operação de “**XOR**” ou “**XNOR**” entre a saída dos blocos **B e C**



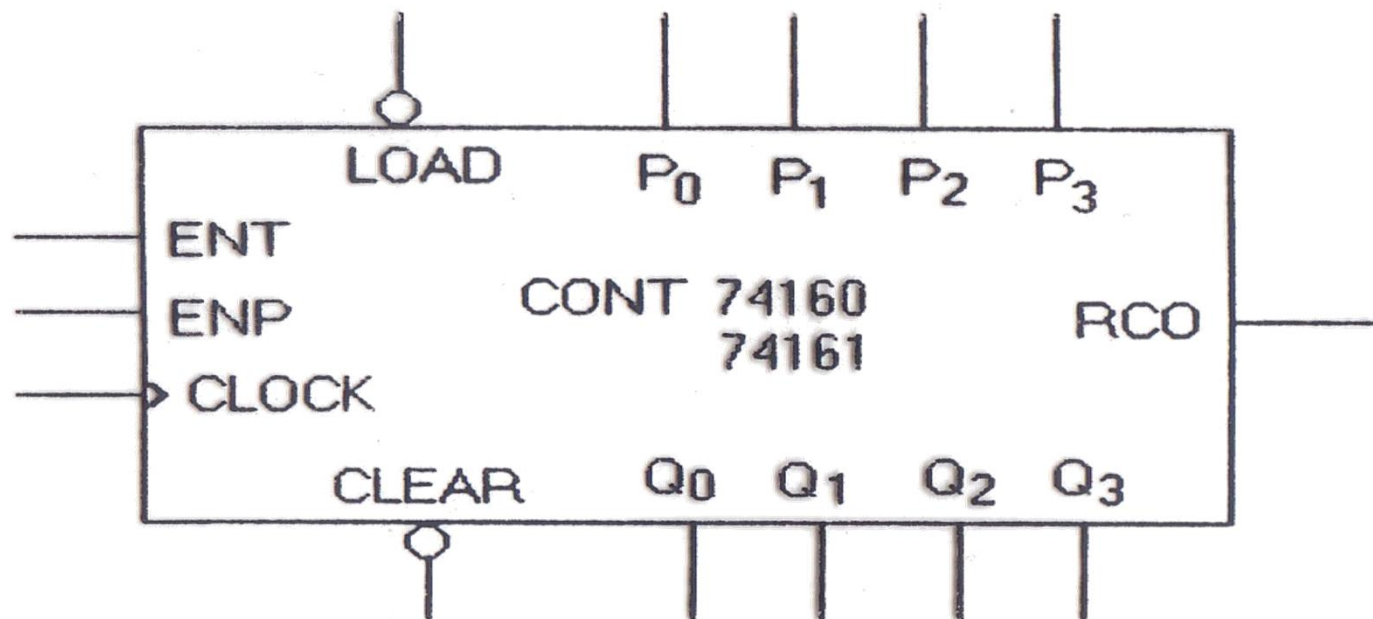
# Contadores – Geração de Onda Quadrada

- Exemplo usando método descrito: divisão por 5



## Contadores Síncronos – Associação em Cascata

- 74160 – Contador módulo 10, sequência BCD.
- 74161 – Contador binário (hexadecimal), módulo 16, *Clear* assíncrono:
  - ENT e ENP – Enable;
  - RCO – *Ripple carry out*.



# Contadores Síncronos – Associação em Cascata

ENTRADAS					FUNÇÃO ASSOCIADA	EFEITO
$\overline{\text{CLEAR}}$	$\overline{\text{LOAD}}$	ENP	ENT	CLOCK		$Q_0 - Q_3$
0	X	X	X	X	ANULA	0 - 0
1	0	X	X	↑	CARREGA	$P_0 - P_3$
1	1	0	X	X	INIBE	$Q_0 - Q_3$
1	1	X	0	X	INIBE	$Q_0 - Q_3$
1	1	1	1	↑	CONTA	$(Q_0 - Q_3)_{n+1}$

# OBS Nº1:

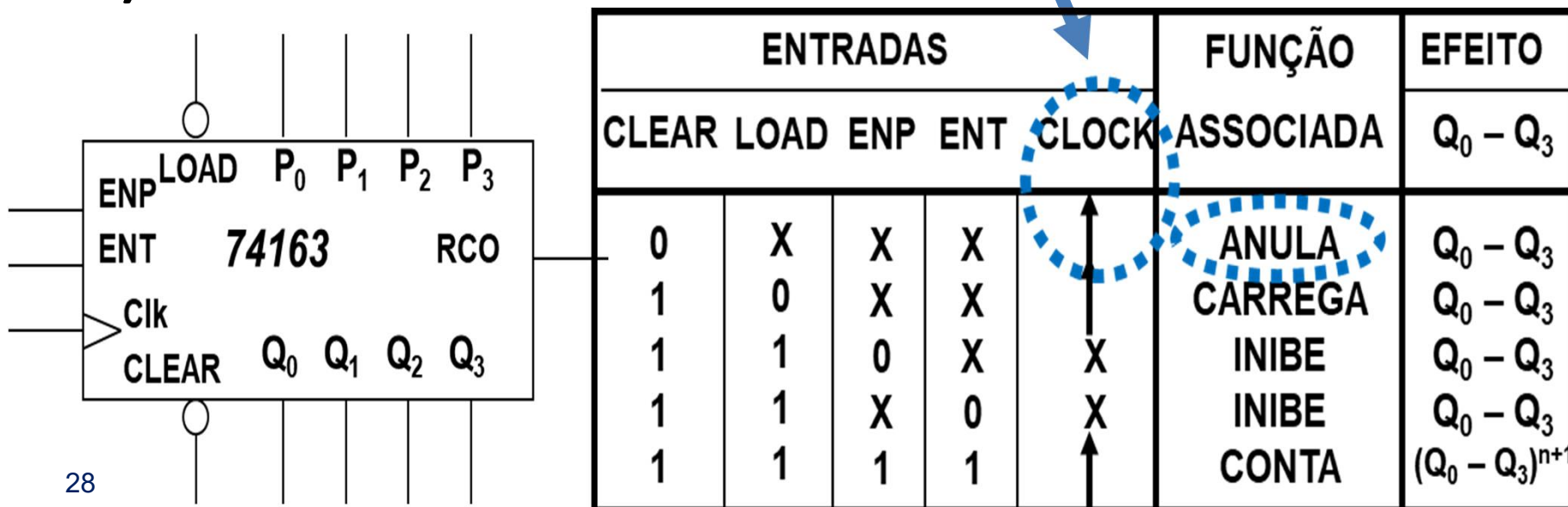
(i) 7416\*: CONTADORES 74160 OU 74161

Decimal: (ii) 74160 :  $\text{RCO} = Q_0 \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot Q_3$  . ENT **Inibe**

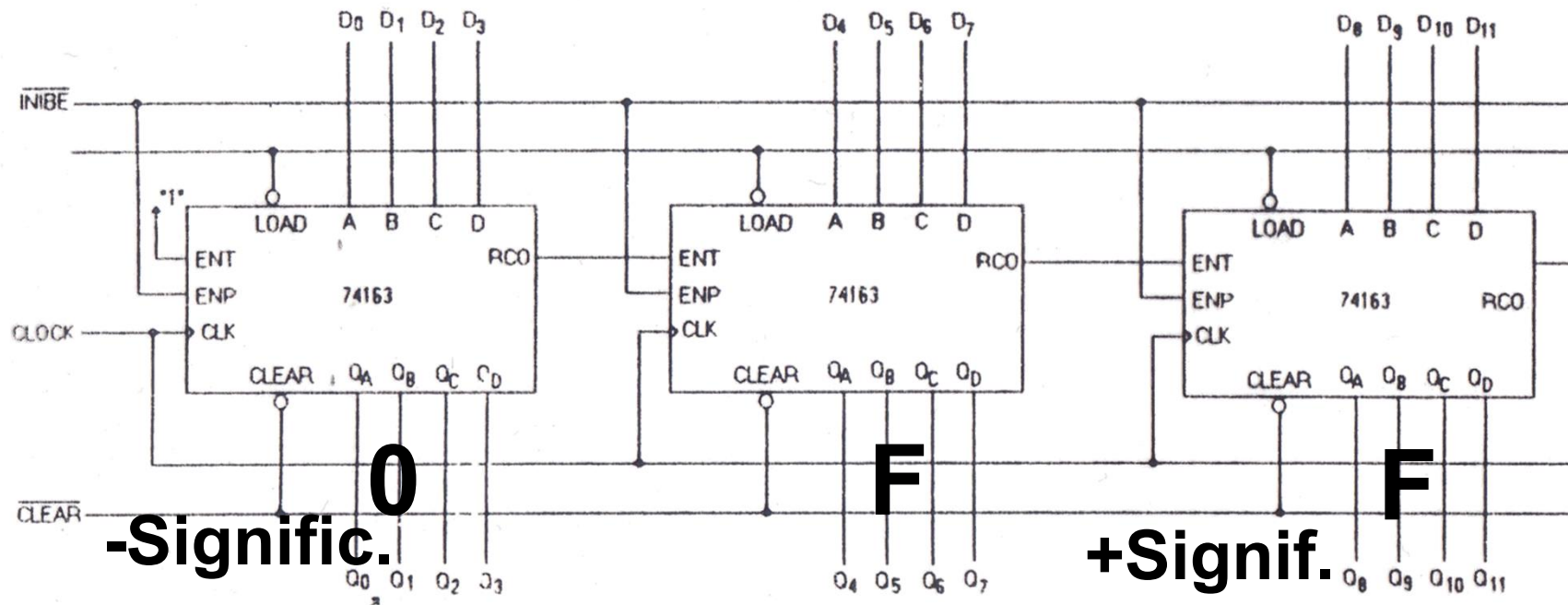
Hexa: (iii) 74161 :  $\text{RCO} = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3$  . ENT **RCO**

# Contadores Síncronos – Exercício

- Projetar **contador módulo 11** (conta dos Estados 0 até 10) utilizando um contador 74163.
- **1) 74163: Contador módulo 16 (hexadecimal), sequência binária, com clear síncrono;**
- **2) RCO: “Ripple Carry Out”;**
- **3) ENT: “Enable Trickle Input”;**
- **4) ENP: “Enable Parallel Input”;**
- **5) RCO = Q0 . Q1 . Q2 . Q3 . ENT**

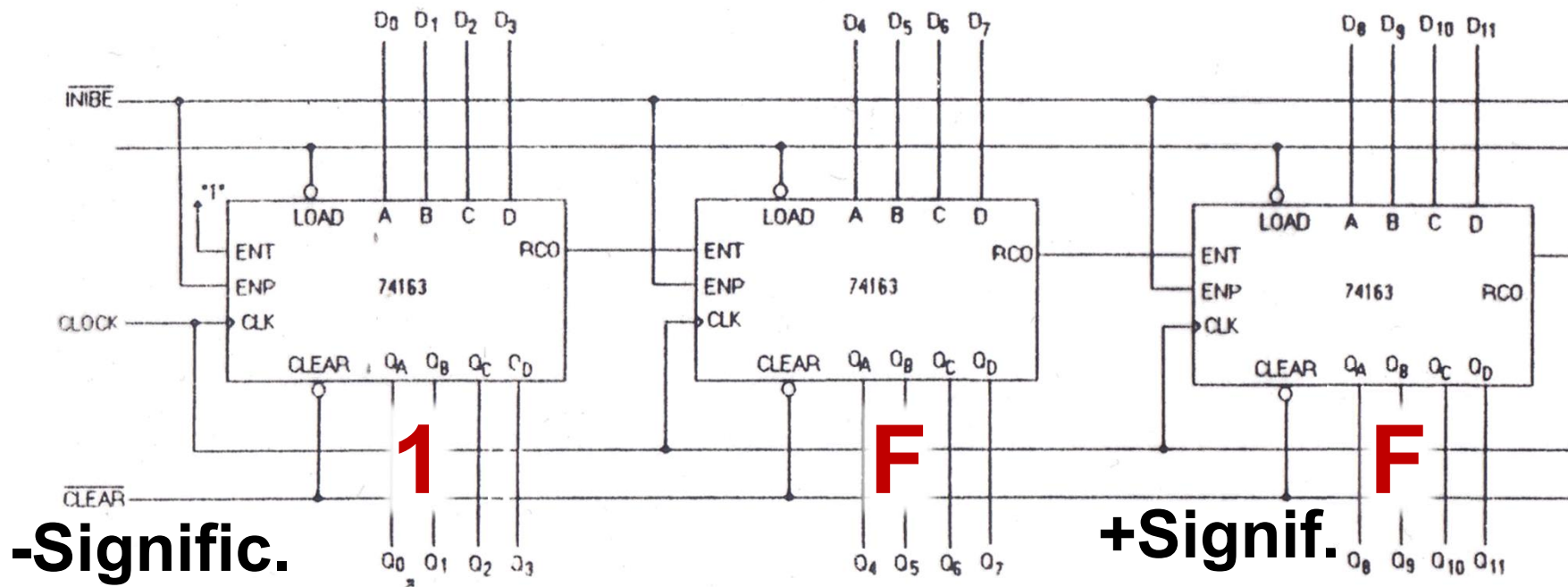


## Contadores Síncronos: Associação em Cascata



- Se o Estado atual dos contadores fosse:  
(-Sig.)**OFF**(+Sig.)
- Qual seria o Estado após a próxima borda de subida do *clock*?

# Contadores Síncronos: Associação em Cascata



- Resposta: Se o sinal ENT do caractere do meio não desabilitasse seu próprio RCO, o Estado após a próxima borda seria:

(-Sig.) **1F0** (+Sig.)

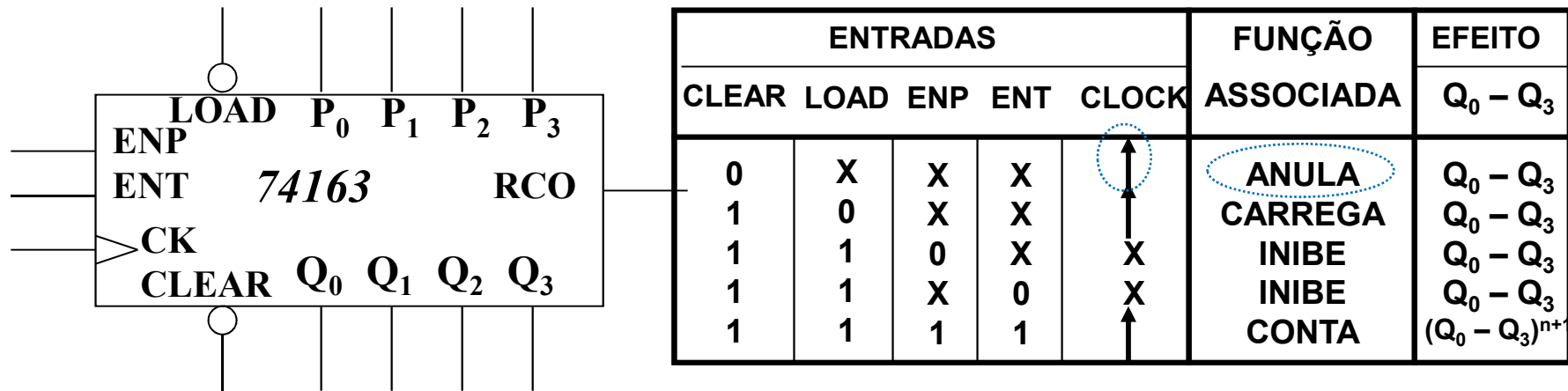
- Obs.: o sinal de enable "ENT" deve ser usado, pois o RCO não depende do enable "ENP"

## Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo $\neq 2^n$

- **Lembrando:** A detecção e **carga assíncrona** de Estado necessitava da detecção do **Estado posterior ao último Estado da sequência desejada**.
- **Síncrono** – Detecta-se o **último Estado da sequência desejada**. O contador permanece um **período de *clock* completo, neste Estado**, e, na **próxima borda do *clock* é *resetado***.

## Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo $\neq 2^n$

- **Síncrono** – Exemplo: Contador módulo 11 (conta dos Estados 0 até 10) com 74163.

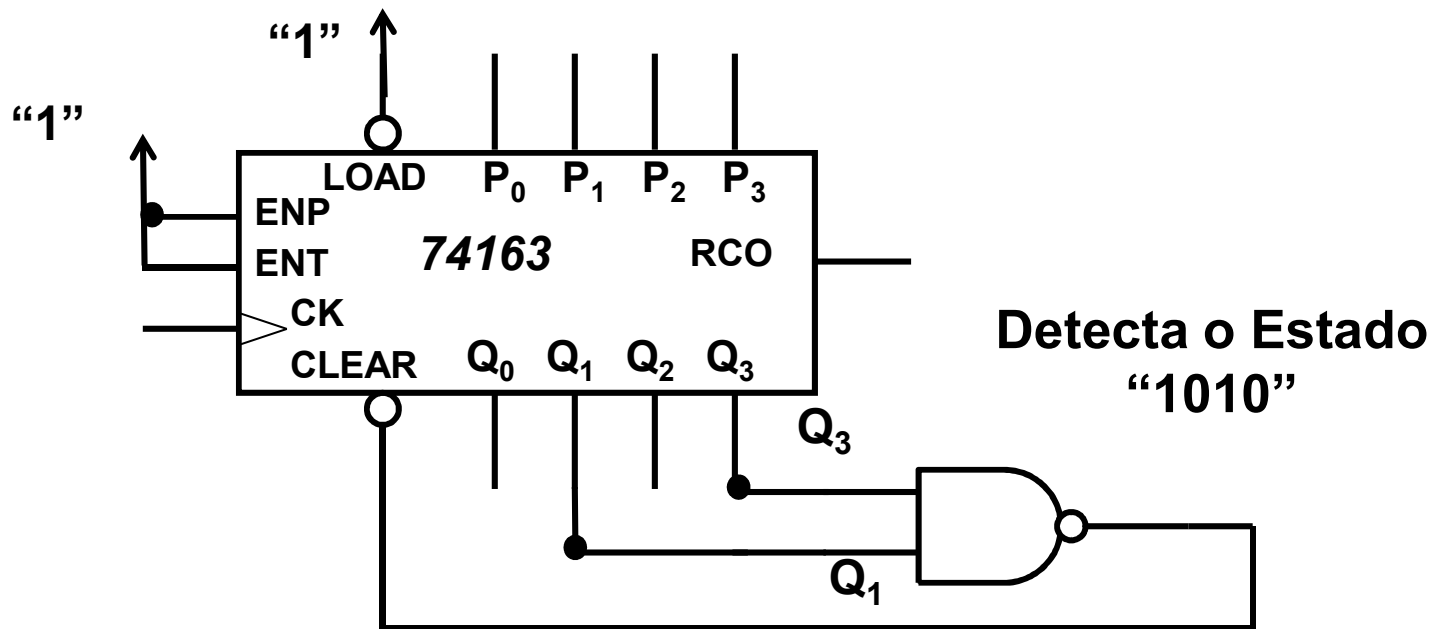


- 1) 74163: Contador módulo 16 (hexadecimal), sequência binária, com [clear síncrono](#)
- 2) RCO: “Ripple Carry Out”;
- 3) ENT: “Enable Trickle Input”;
- 4) ENP: “Enable Parallel Input”;
- 5)  $RCO = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot ENT$



# Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo $\neq 2^n$

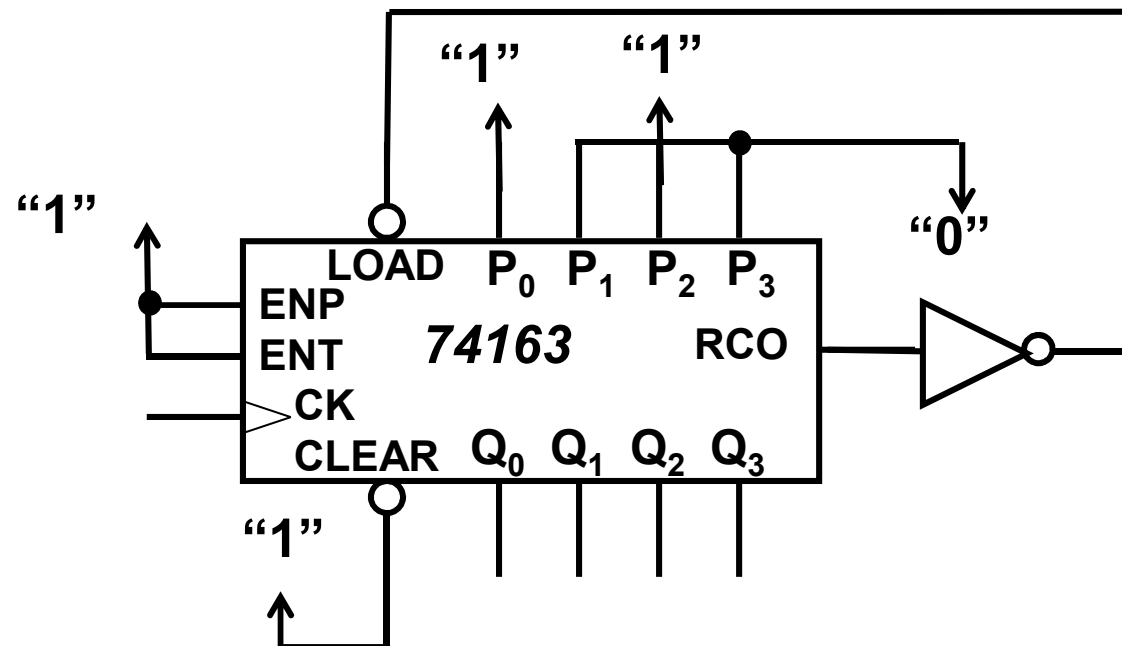
- **Síncrono** – Exemplo: Contador módulo 11 (conta dos Estados 0 até 10) com 74163.



## Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo $\neq 2^n$

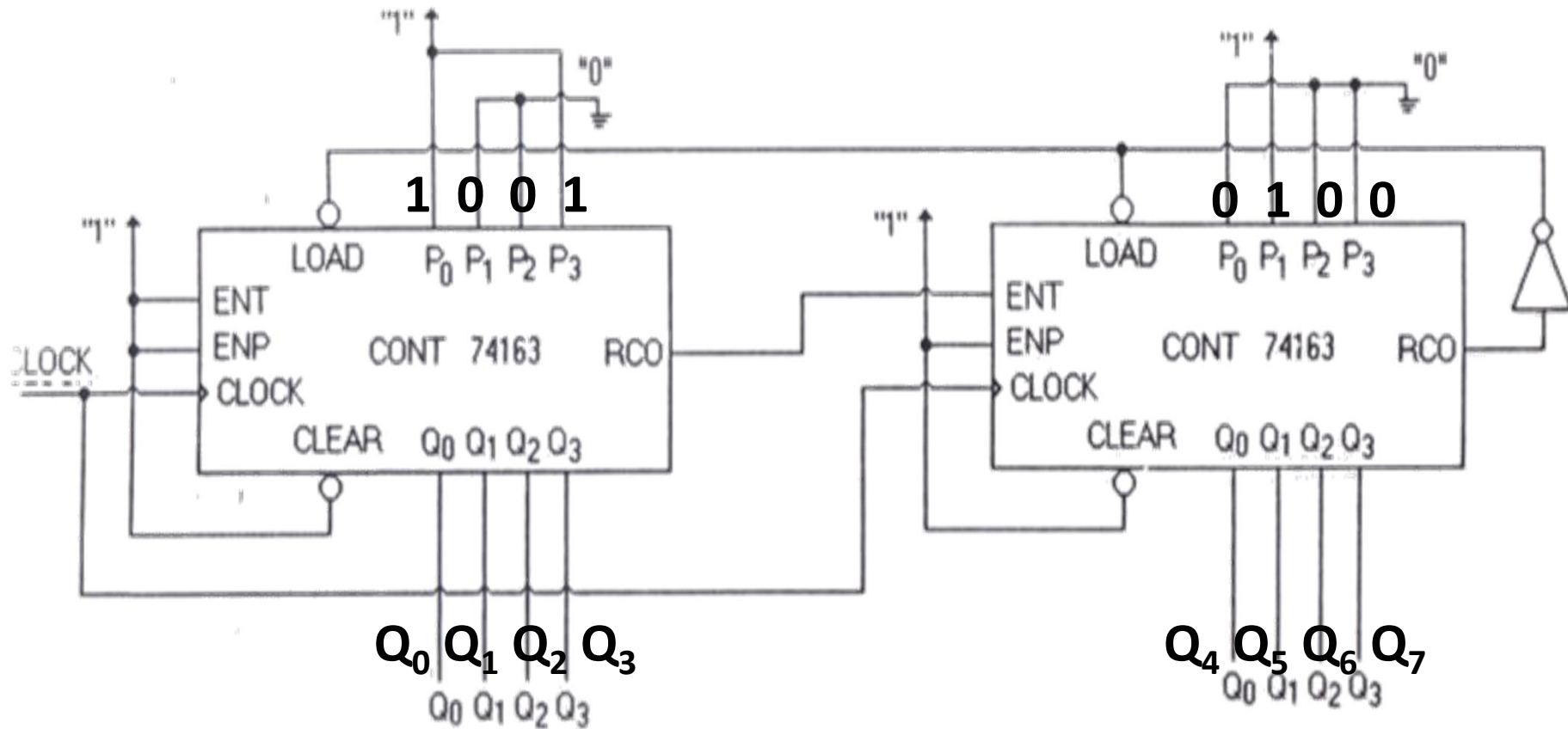
- **Alternativa – Contador módulo 11**

- Conta dos Estados **5 até 15**;
- Usa o sinal de **RCO** para detectar o Estado **15**, ligando-o (invertido) ao **LOAD** para carregar o Estado **5**.



# Detecção/Carga Síncrona de Estado – Sequência de Módulo $\neq 2^n$

- Exercício** – Qual o módulo deste contador?



**Q<sub>7</sub> Q<sub>6</sub> Q<sub>5</sub> Q<sub>4</sub> Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub> Q<sub>0</sub> 0 0 1 0 1 0 0 1**  
**0 0 1 0 1 0 0 1**

**32 + 8 + 1 = 41**

Detecção/Carga Síncrona de Estado – Sequência de Módulo  $\neq 2^n$

- **Exercício** – Qual o módulo deste contador?

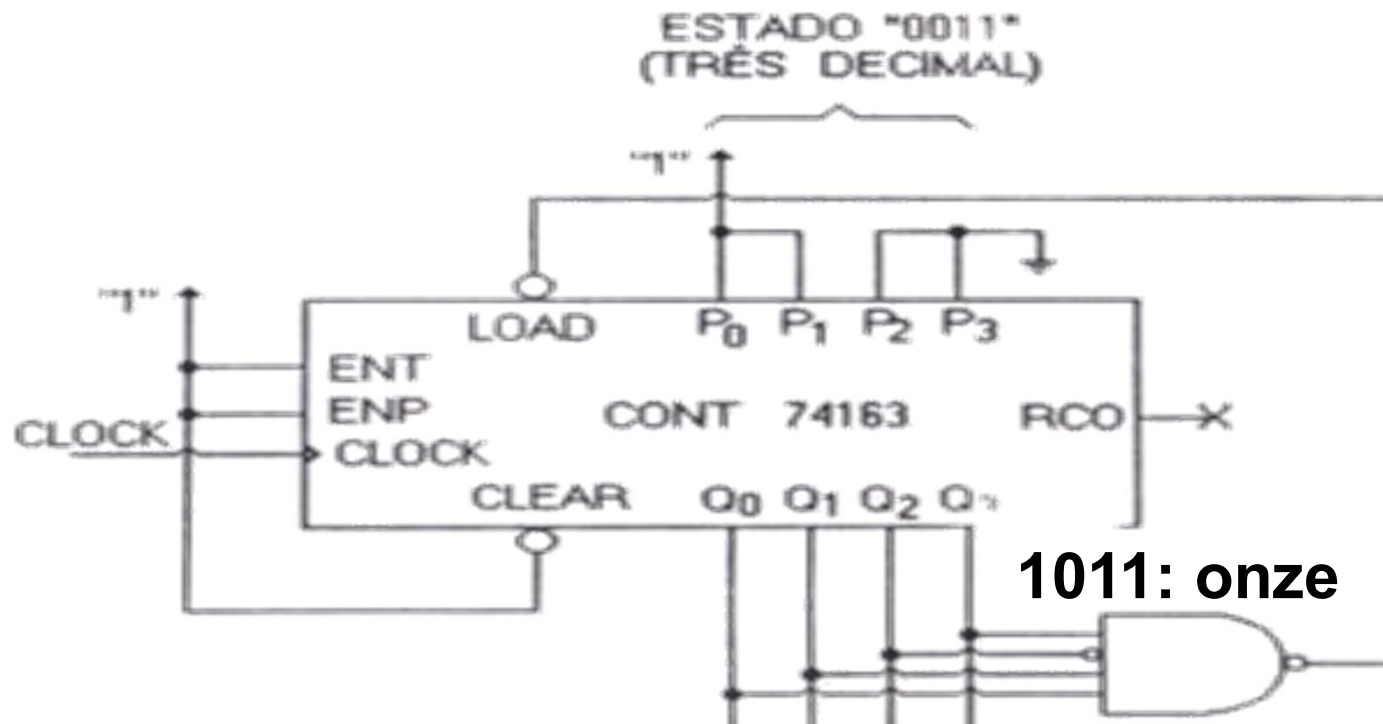
	C1	C2	clocks
<b>Carga Paralela De C1 e C2</b>	$(9)_{10} = (1001)_2$	$(2)_{10} = (0010)_2$	<b>Total:</b>
<b>Conta</b>		de 2 até 15 (F) = 13	<b>7+13*16</b>
<b>Conta</b>	de 9 até 15 (F) = 7 (1a vez)	2	<b>+7</b>
<b>Conta</b>	de 0 até 15 (F) = 16	3	<b>+16</b>
<b>Conta</b>	de 0 até 15 (F) = 16	...	<b>+16</b>
<b>RCO<sub>2</sub> = 1•ENT Carga Paralela</b>	de 0 até 15 (F) = 16	F	<b>+16</b>

*Handwritten annotations:*

- A large curved arrow on the left side points from the bottom row up to the top row.
- A bracket on the right side groups the rows with values +16, +16, and +16, with a multiplier **\*13** next to it.
- Arrows on the right point from the values 2, 3, ..., and F in the C2 column to the corresponding +16 values in the clocks column.

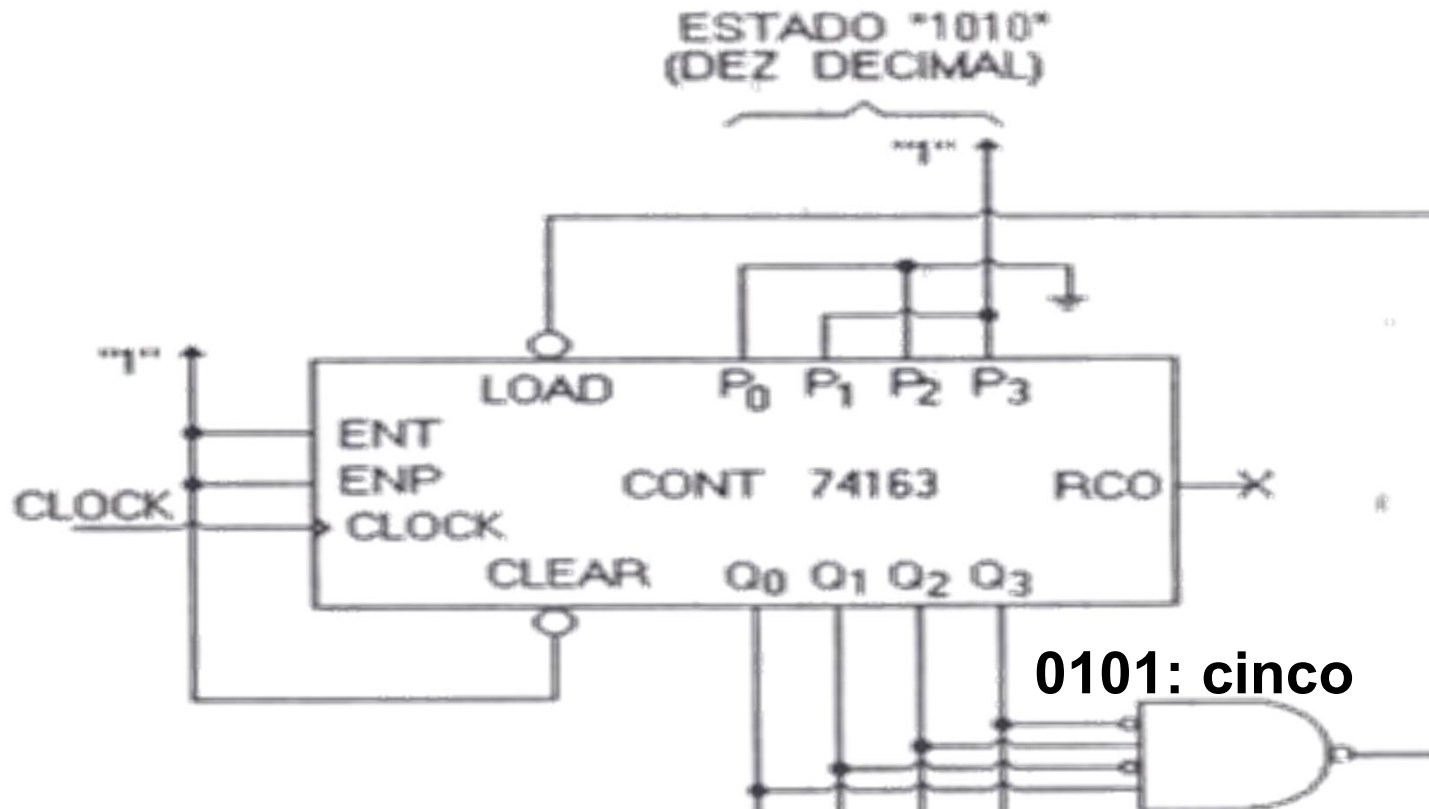
Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo  $\neq 2^n$

- **Exercício** – Qual o módulo deste contador? Qual a sequência de contagem?
  - **Resposta:** intervalo [3 , 11] → módulo 9



Contadores Síncronos – Detecção e Carga Síncrona de Estado para Sequência de Módulo  $\neq 2^n$

- **Exercício** – Qual o módulo deste contador? Qual a sequência de contagem?
  - **Resposta:** intervalo  $[10, 15] + [0, 5] \rightarrow$  módulo 12



## Contadores Síncronos – Geração de Onda Quadrada para Sequências de Módulo $\neq 2^n$

- Divisão de Frequência por  $x = 2^n$ : “pulando” alguns estados escolhidos
  - Exemplo: divisão por 6

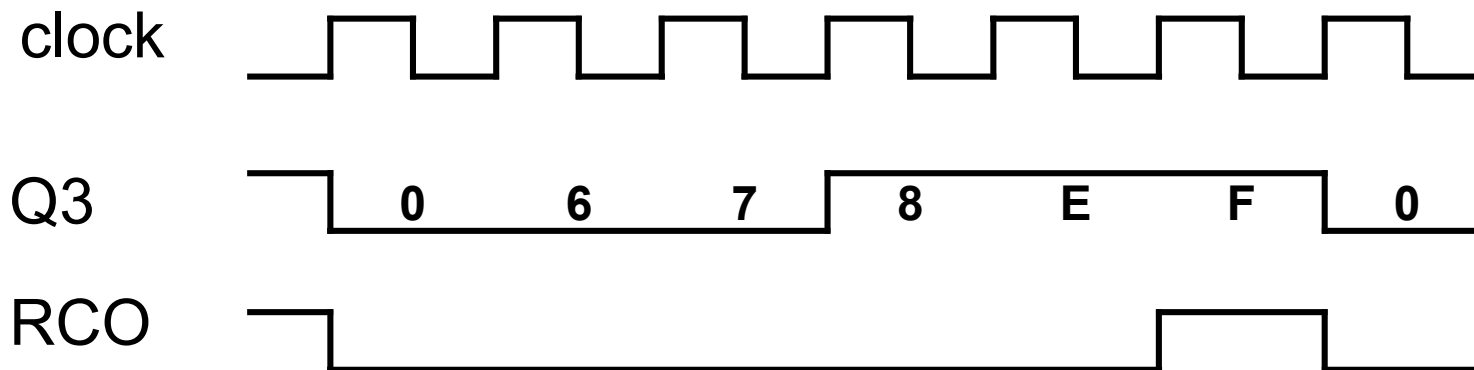
	Q3	Q2	Q1	Q0	Estado	Ação	Pulou
3 zeros	0	0	0	0	0	carrega 6	1,2,3,4,5
	0	1	1	0	6	conta	
	0	1	1	1	7	conta	
3 uns	1	0	0	0	8	carrega 14	9, A, B, C, D
	1	1	1	0	E	conta	
	1	1	1	1	F	conta	

Carga: Q3,1,0,0

# Contadores Síncronos – Geração de Onda Quadrada para Sequências de Módulo $\neq 2^n$

## ○ Divisão de Frequência por 6

Q3	Q2	Q1	Q0	Estado	Ação	Pulou
0	0	0	0	0	carrega 6	1,2,3,4,5
0	1	1	0	6	conta	
0	1	1	1	7	conta	
1	0	0	0	8	carrega 14	9, A, B, C, D
1	1	1	0	E	conta	
1	1	1	1	F	conta	





# Contadores Síncronos – Geração de Onda Quadrada para Sequências de Módulo $\neq 2^n$

## ○ Divisão de Frequência por 6

