

Registadores

Circuitos Lógicos

DCC-IM/UFRJ

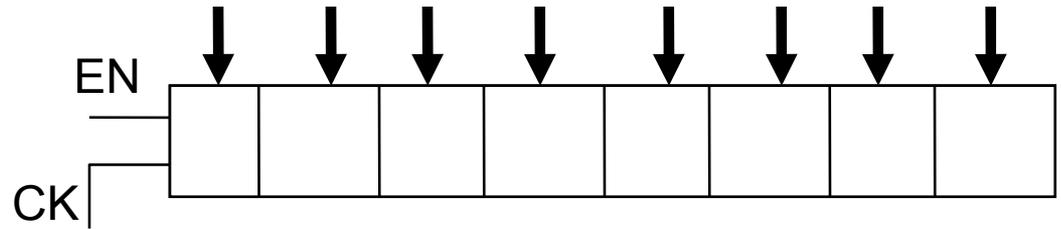
Prof. Gabriel P. Silva

Registradores

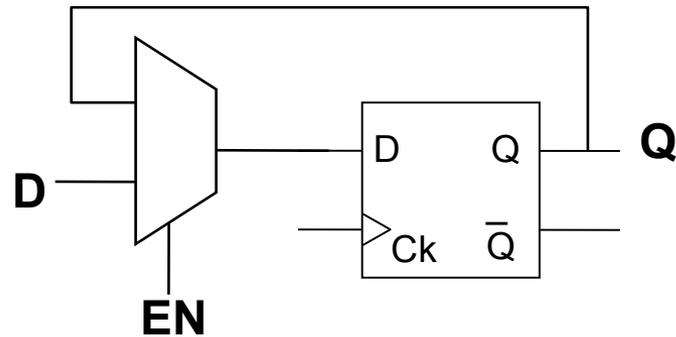
- **Conjunto de elementos de memória (flip-flops ou latches) utilizados para armazenar n bits.**
- **Utilizam um único sinal de clock ou um único sinal de habilitação.**
- **São elementos de memória básicos utilizados quando se deseja armazenar pequenas quantidades de informação, normalmente em circuitos de interface ou controle de máquinas de estado.**

Registadores

Registrador de carga paralela



Carga do registrador



(74LS377)

Registrador

```
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;
```

```
ENTITY Reg IS  
  PORT(Data_in : IN  STD_LOGIC_VECTOR;  
        Data_out: OUT STD_LOGIC_VECTOR;  
        Wr      : IN  STD_LOGIC ;  
        Reset   : IN  STD_LOGIC ;  
        Clk    : IN  STD_LOGIC);  
END Reg;
```

```
ARCHITECTURE behavioral OF Reg IS  
BEGIN  
  PROCESS(Wr,Reset,Clk)  
    CONSTANT Reg_delay: TIME := 2 ns;  
    VARIABLE BVZero:  
STD_LOGIC_VECTOR(Data_in'RANGE):= (OTHERS => '0');
```

Registrador

```
BEGIN  
  IF (Reset = '1') THEN  
    Data_out <= BVZero AFTER Reg_delay;  
  END IF;  
  
  IF (Clk'EVENT AND Clk = '1' AND Wr = '1') THEN  
    Data_out <= Data_in AFTER Reg_delay;  
  END IF;  
END PROCESS;  
END behavioral;
```

Deslocador

- **O deslocador serve para mover um conjunto de bits de uma ou mais posições para a esquerda ou direita.**
- **Dependendo do tipo de deslocamento, podem ser inseridos '0's para as posições que ficam vagas à medida que os bits correspondentes vão sendo deslocados.**
- **Para os números cuja representação coloca o bit de sinal no bit mais à esquerda, normalmente esse bit é replicado quando os bits são deslocados para a direita.**

Deslocador

- **Deslocamento para a direita de 2 bits:**

00001111 --> 00000011

- **Deslocamento para esquerda de 3 bits**

00001111 --> 01111000

- **Deslocamento para a direita de 2 bits de valor negativo em complemento a dois:**

10001111 --> 11100011

Deslocador

- Note que para cada bit deslocado para a direita, corresponde a uma divisão inteira por 2:

00011100 --> 00000111
28 --> 7

- E para cada bit deslocado para a esquerda, corresponde a uma multiplicação por 2:

00001111 --> 01111000
15 --> 120

Deslocador

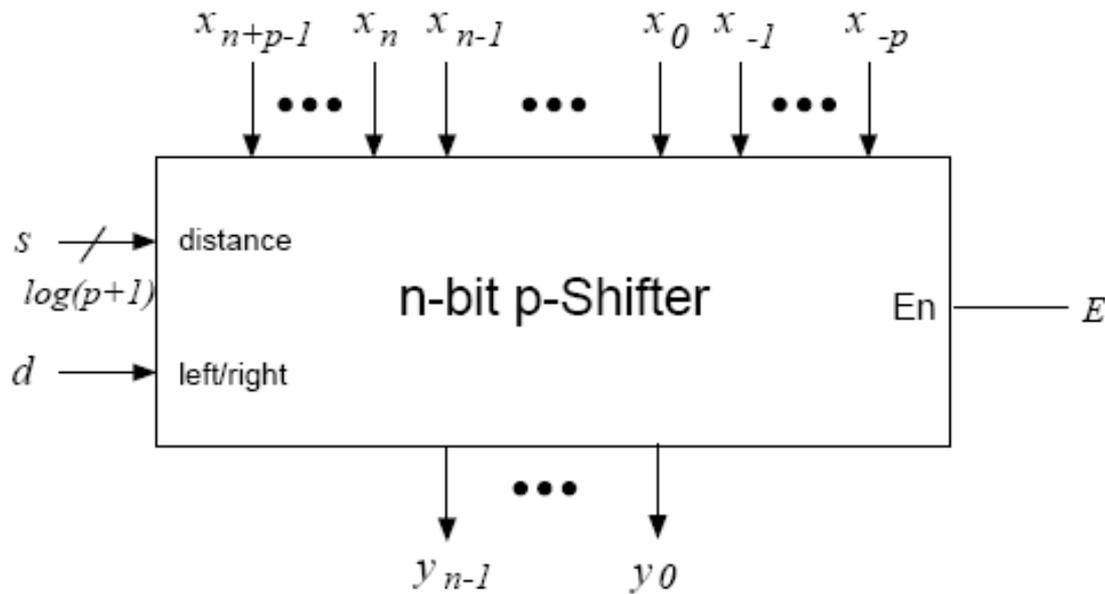
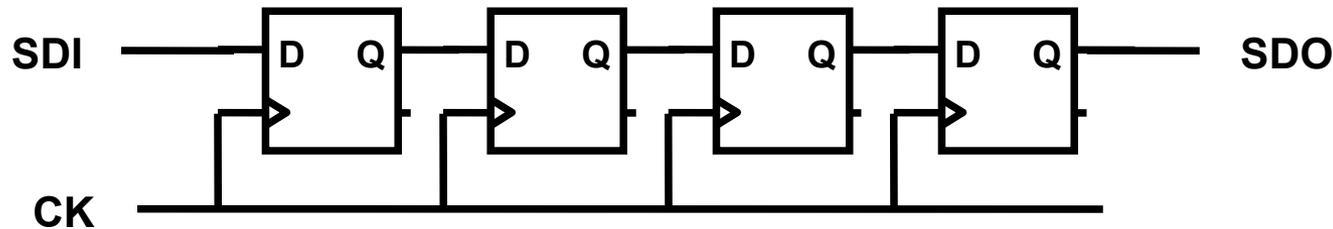


Figure 9.30: n-BIT p -SHIFTER.

Registrador de Deslocamento



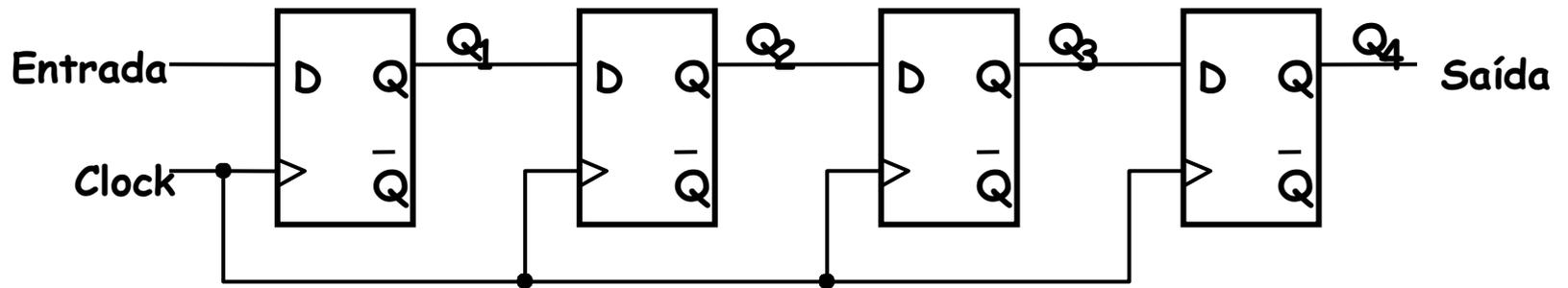
- Composto por uma série de Latches ou Flip-Flops onde a saída Q de um está acoplada a entrada D do seguinte.
- Aplicações:
 - As UALs dos processadores têm registradores de deslocamento (esquerda, direita e em anel).
 - Conversão série para paralelo e vice-versa.
- No exemplo acima os últimos 4 valores da seqüência são armazenados do registrador de deslocamento.

Registrador de Deslocamento

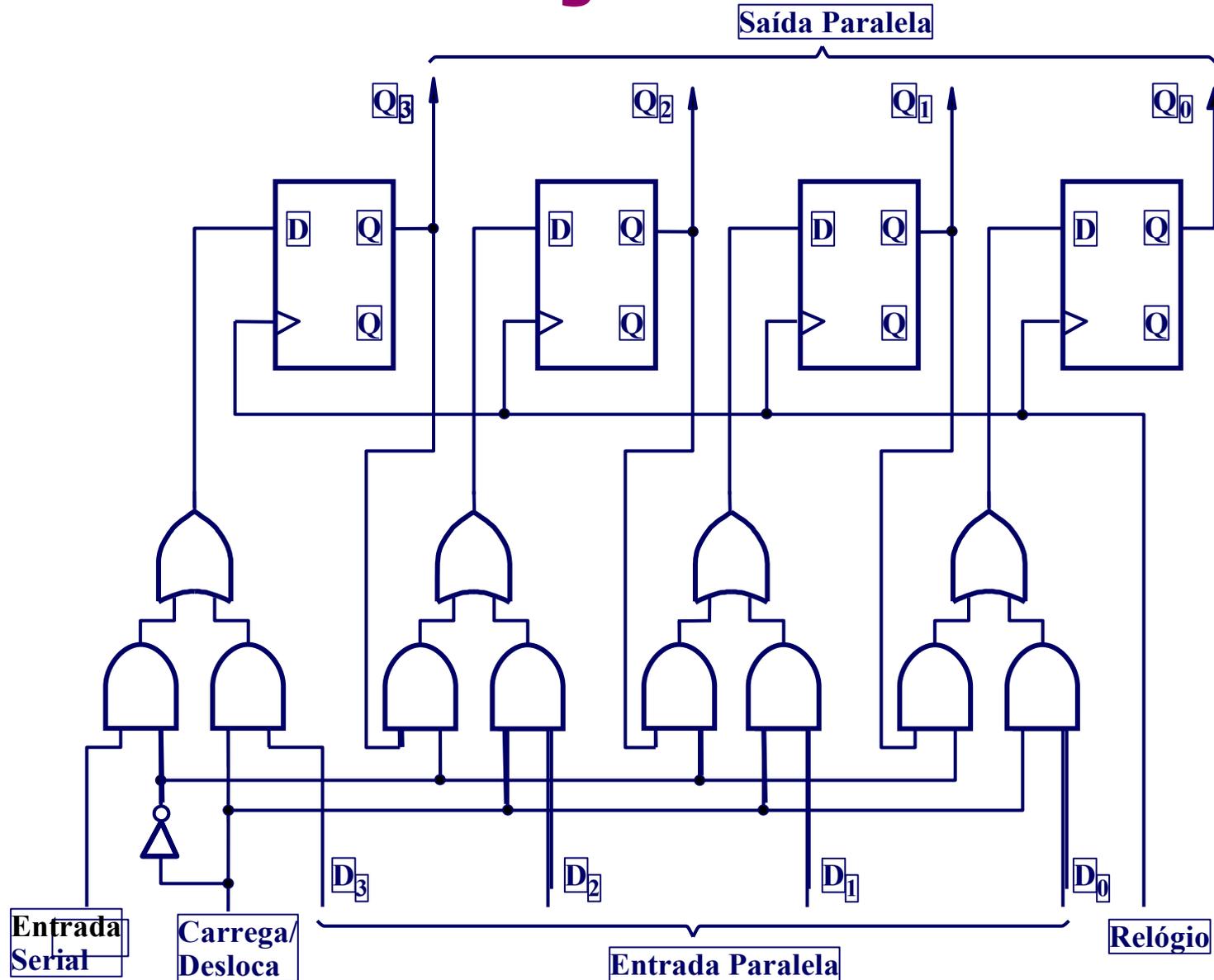
Apresenta o seguinte comportamento:

Clock	Entrada	Q ₁	Q ₂	Q ₃	Q ₄ = Saída
t ₀	1	0	0	0	0
t ₁	0	1	0	0	0
t ₂	1	0	1	0	0
t ₃	1	1	0	1	0
t ₄	1	1	1	0	1
t ₅	0	1	1	1	0
t ₆	0	0	1	1	1
t ₇	0	0	0	1	1

Registrador de Deslocamento



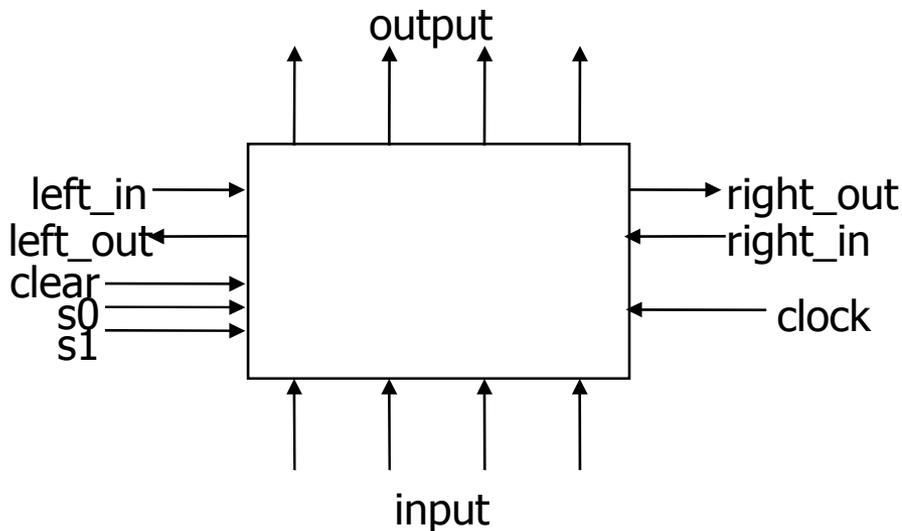
Registrador de Deslocamento com Carga Paralela



Registrador de Deslocamento Universal

□ Armazena 4 valores

- Entradas seriais ou paralelas
- Saídas seriais ou paralelas
- Permite o deslocamento à esquerda ou à direita
- Desloca novos valores à esquerda ou à direita



clear estabelece o conteúdo do registrador e da saída em 0

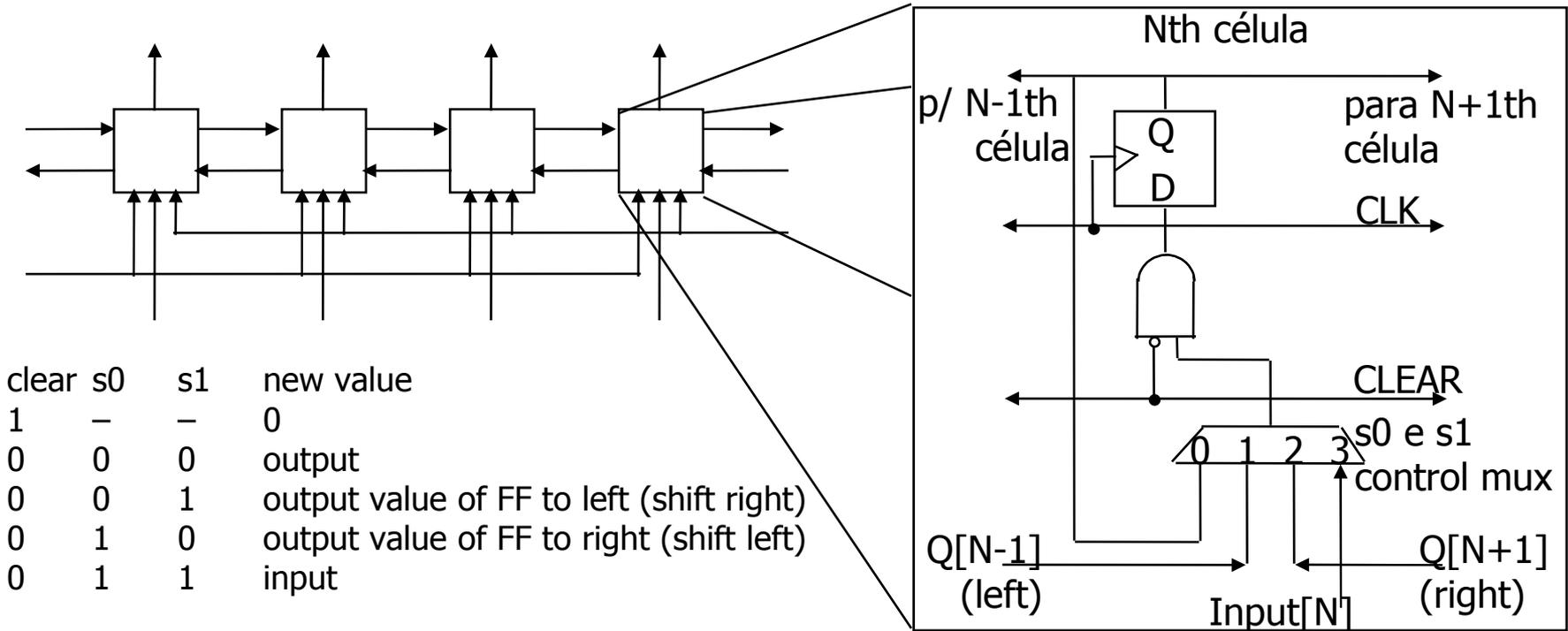
s1 e s0 determinam o tipo de deslocamento

s0	s1	função
0	0	mantém estado
0	1	deslocamento à direita
1	0	deslocamento à esquerda
1	1	carregue nova entrada

Registrador de Deslocamento Universal

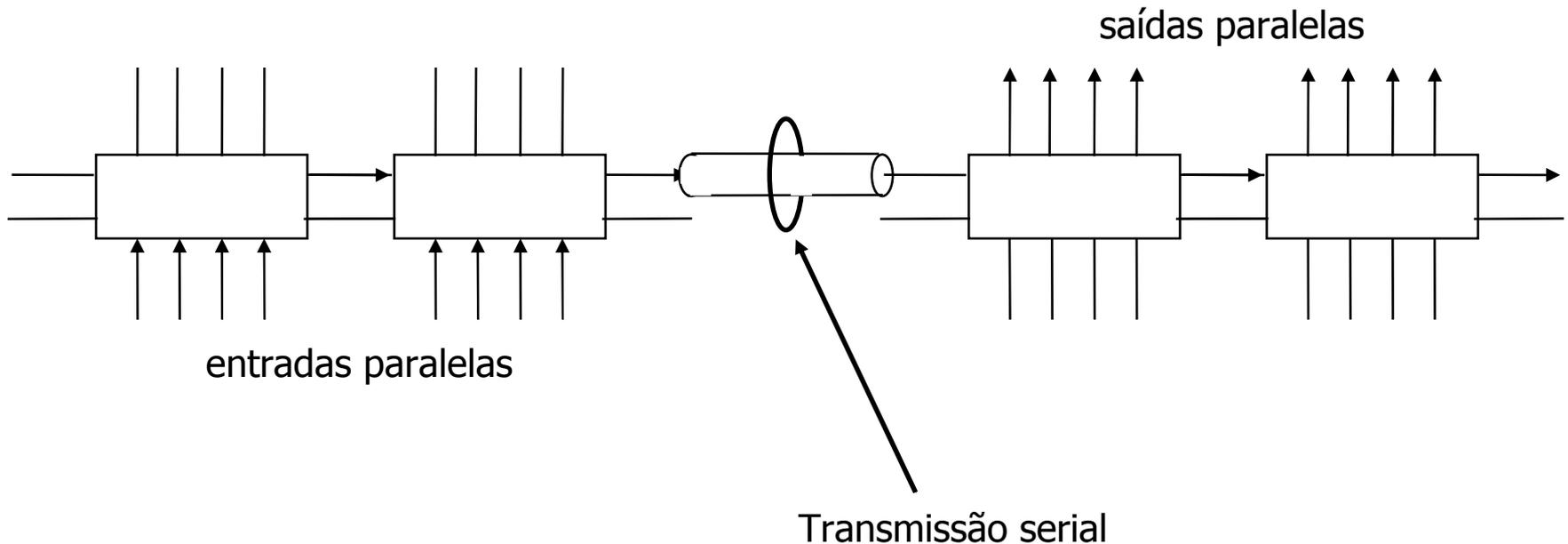
□ Considere um dos quatro flip-flops

- Próximo valor no próximo ciclo de clock:



Aplicação de Registradores de Deslocamento

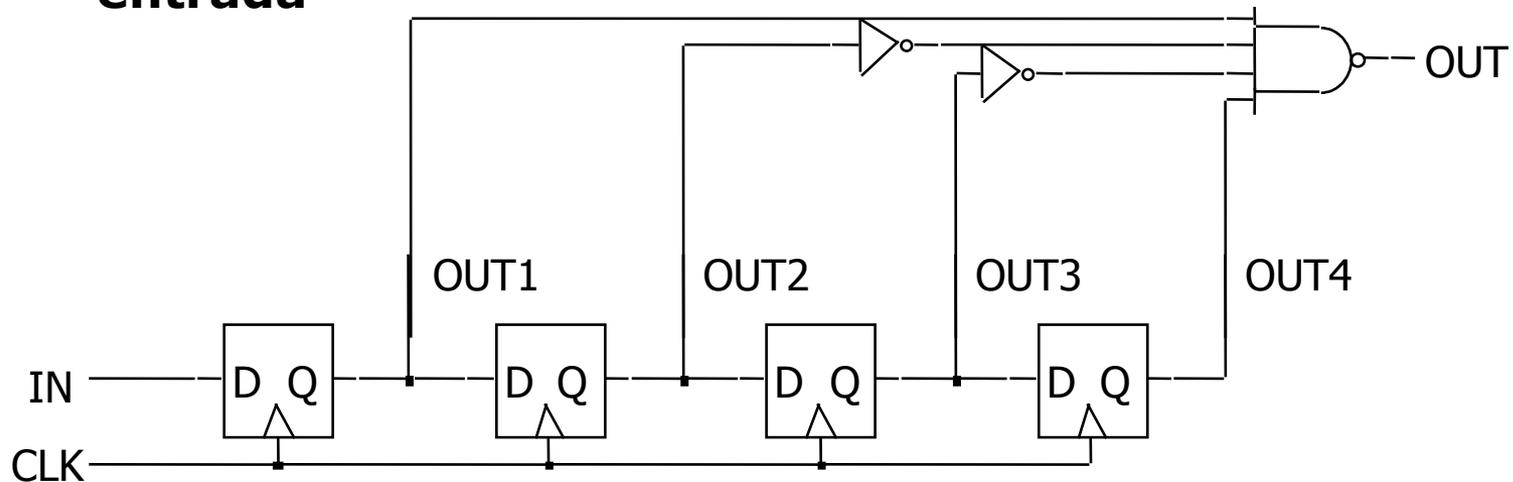
❑ Conversão paralelo-serial para transmissão serial



Reconhecedor de Padrões

❑ Função combinacional de amostras da entrada

- ex: reconhecendo o padrão 1001 em um único sinal de entrada



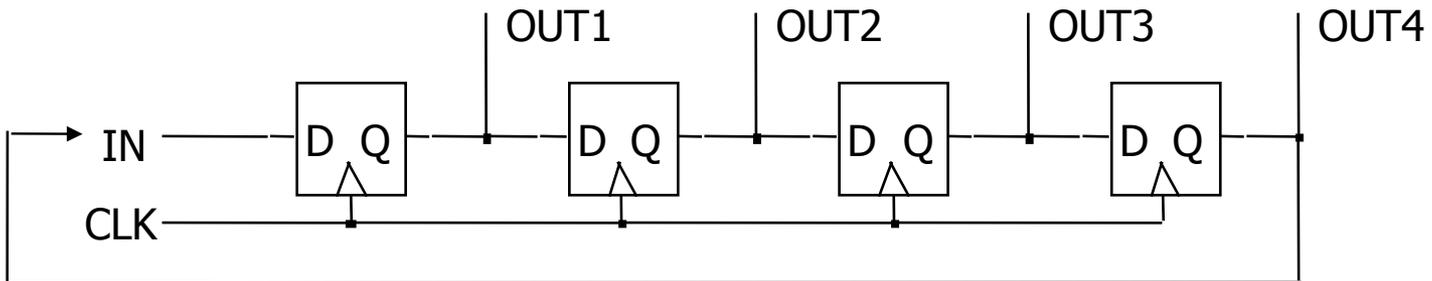
Contadores

- **Podem ser:**
 - **Assíncronos**
 - **São mais lentos pois a saída de um flip-flop dispara uma mudança no flip-flop seguinte**
 - **Síncronos**
 - **O estado de todos os flip-flops são alterados simultaneamente**

Contadores

❑ Seqüências através de um conjunto fixo de padrões

- Neste caso: **1000, 0100, 0010, 0001**
- Um dos padrões é o estado inicial (usar load ou set/reset)

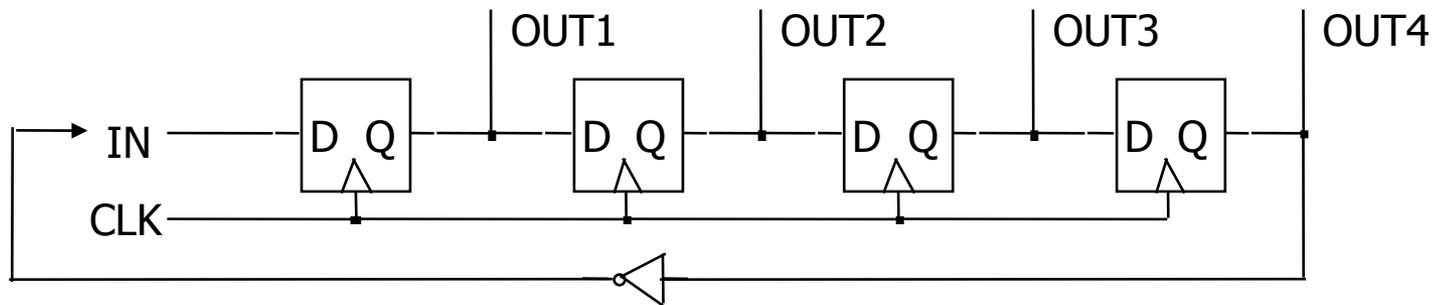


Contadores

□ Contador Mobius (ou Johnson)

- Neste há apenas a mudança de um bit entre dois estados adjacentes:

1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000

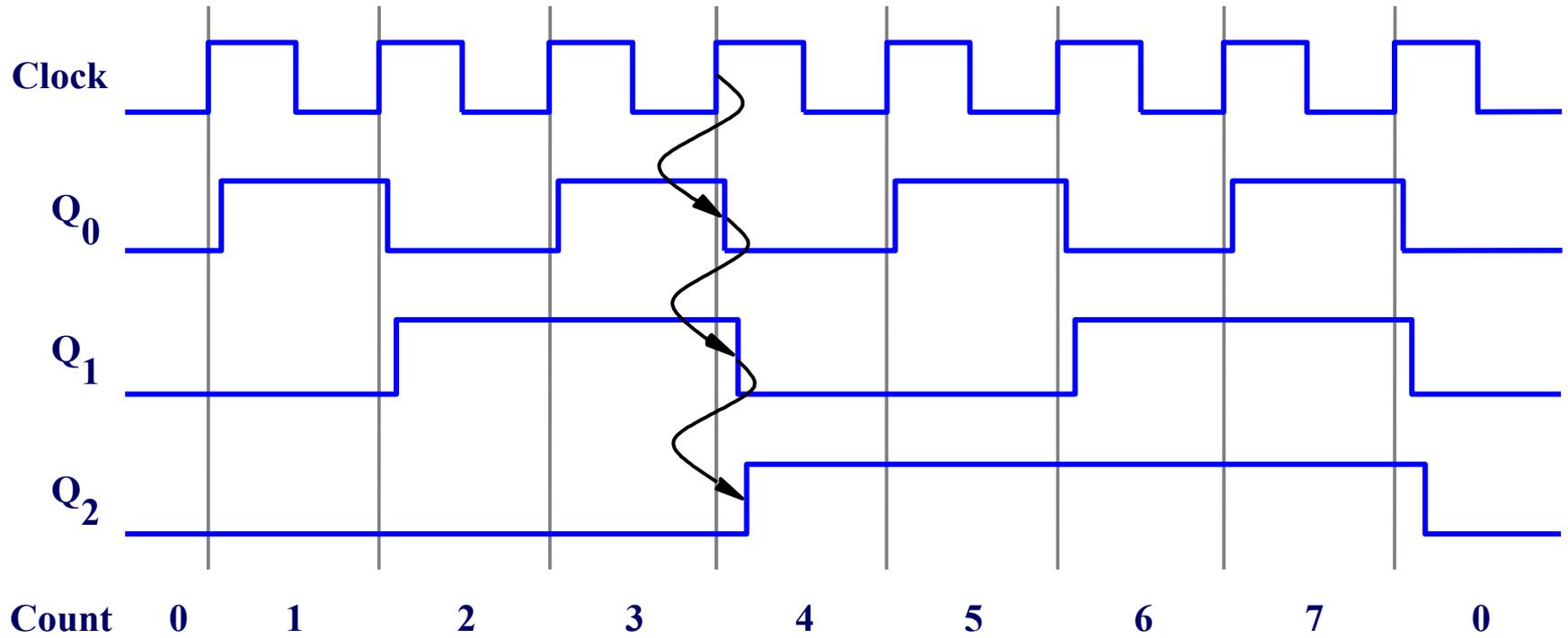


Contadores

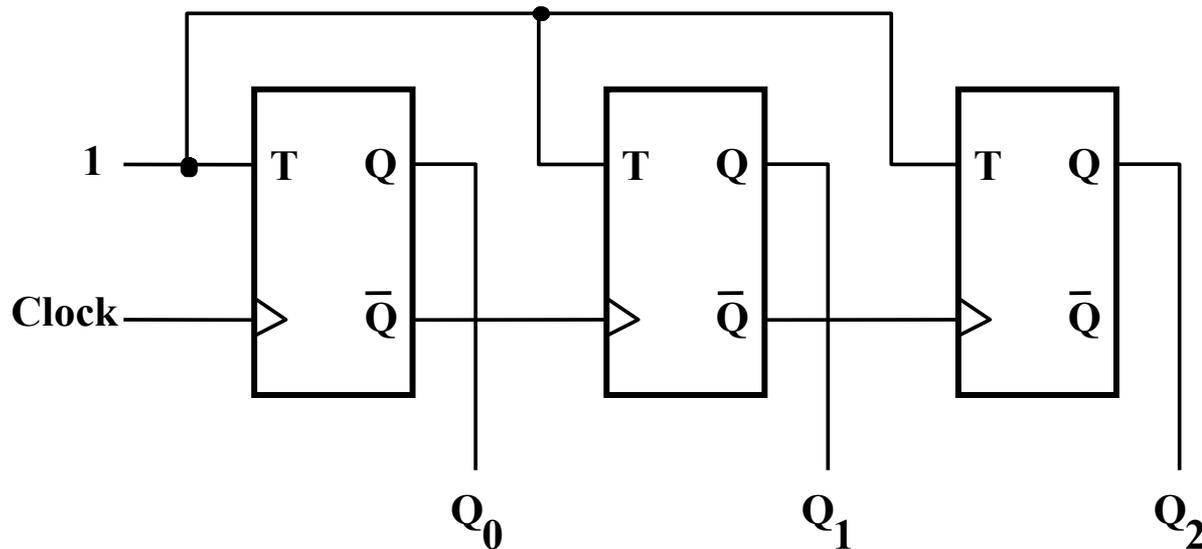
Contador Binário Crescente

	clk	Q ₂	Q ₁	Q ₀
<i>t</i> ₀	↑	0	0	0
<i>t</i> ₁	↑	0	0	1
<i>t</i> ₂	↑	0	1	0
<i>t</i> ₃	↑	0	1	1
<i>t</i> ₄	↑	1	0	0
<i>t</i> ₅	↑	1	0	1
<i>t</i> ₆	↑	1	1	0
<i>t</i> ₇	↑	1	1	1

Contador Binário – FF tipo T

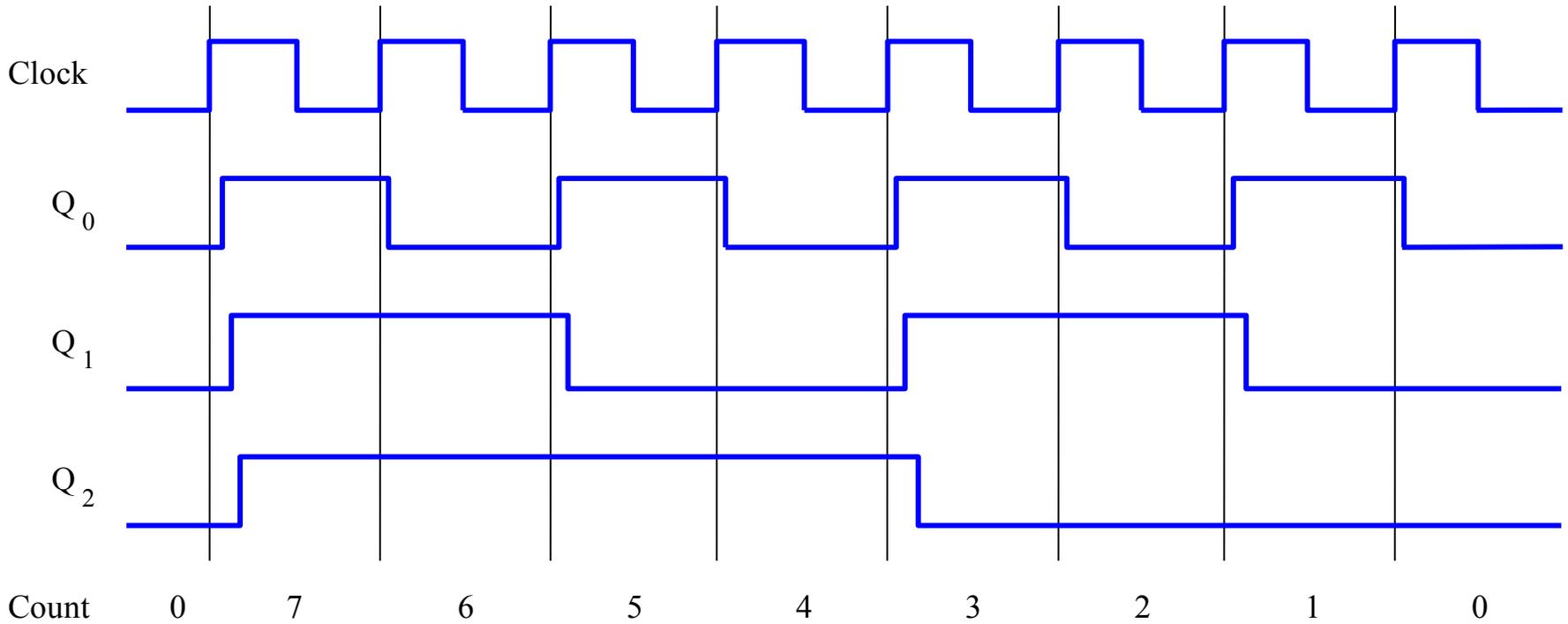


Contador Binário Assíncrono – FF tipo T

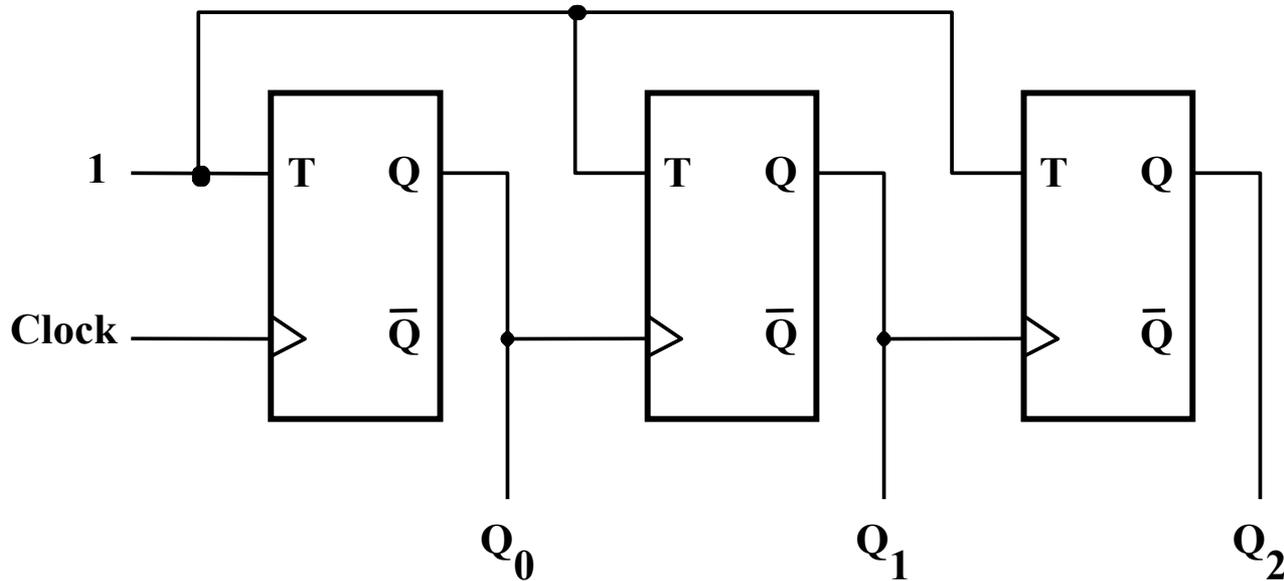


**Contador Binário Assíncrono
(crescente)**

Contador Binário Assíncrono – FF tipo T (decescente)

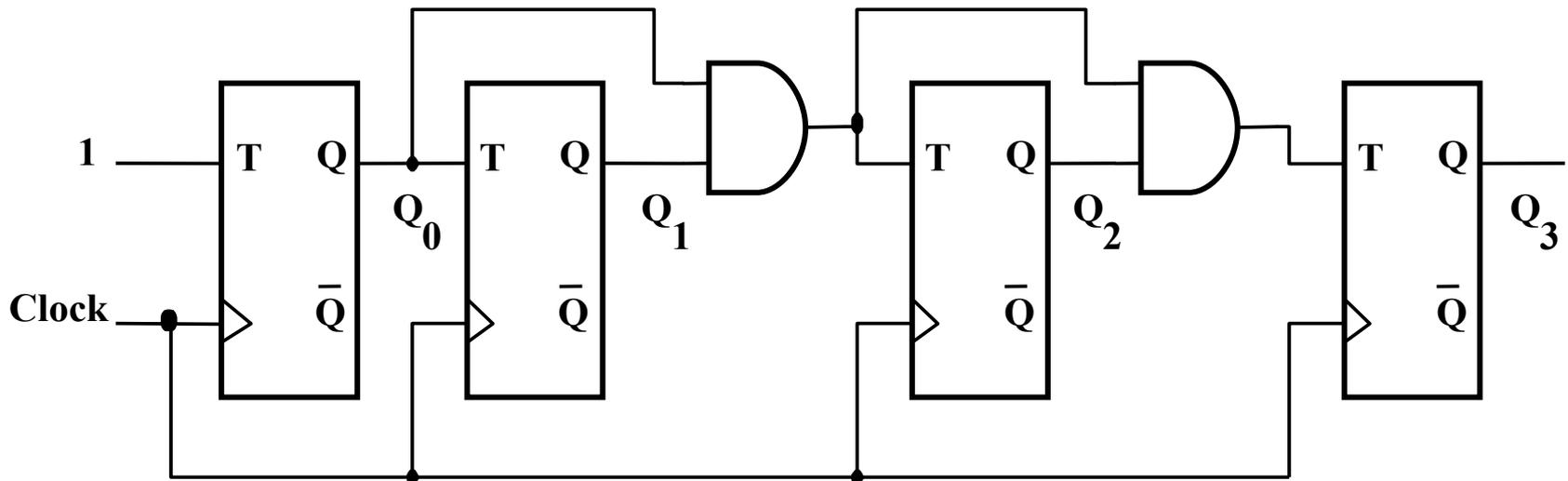


Contador Binário Assíncrono FF tipo T



Contador Binário Assíncrono (decrecente)

Contador Binário Síncrono FF tipo T



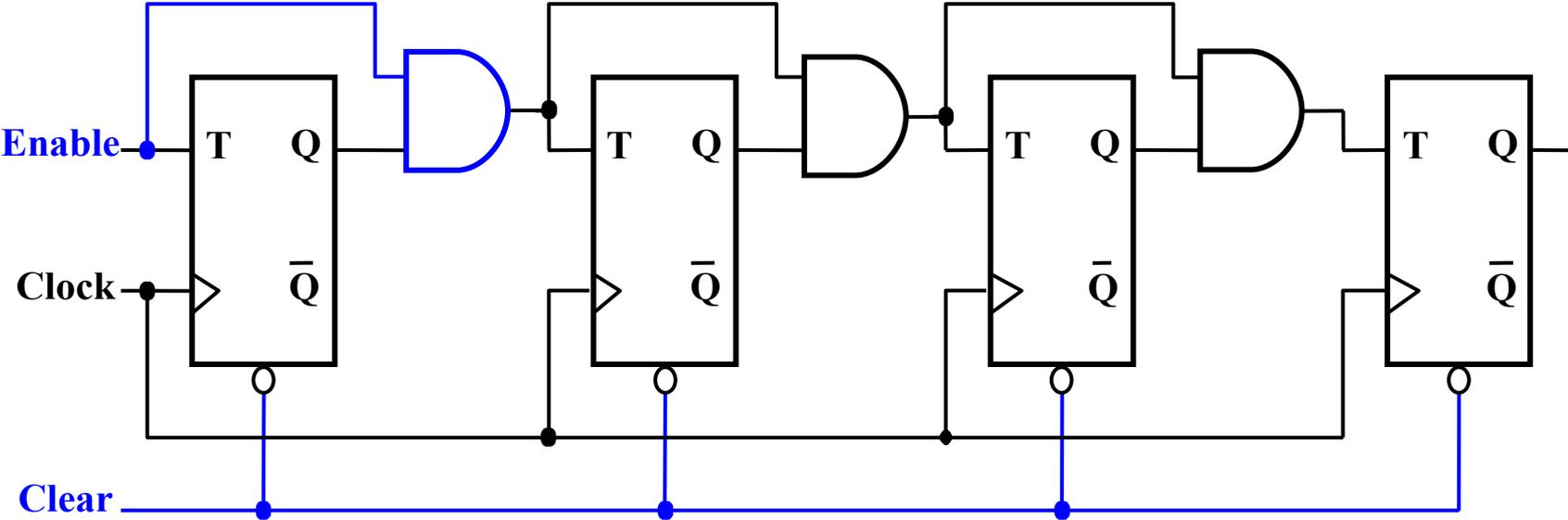
Contador Binário Síncrono (crescente)

Contadores

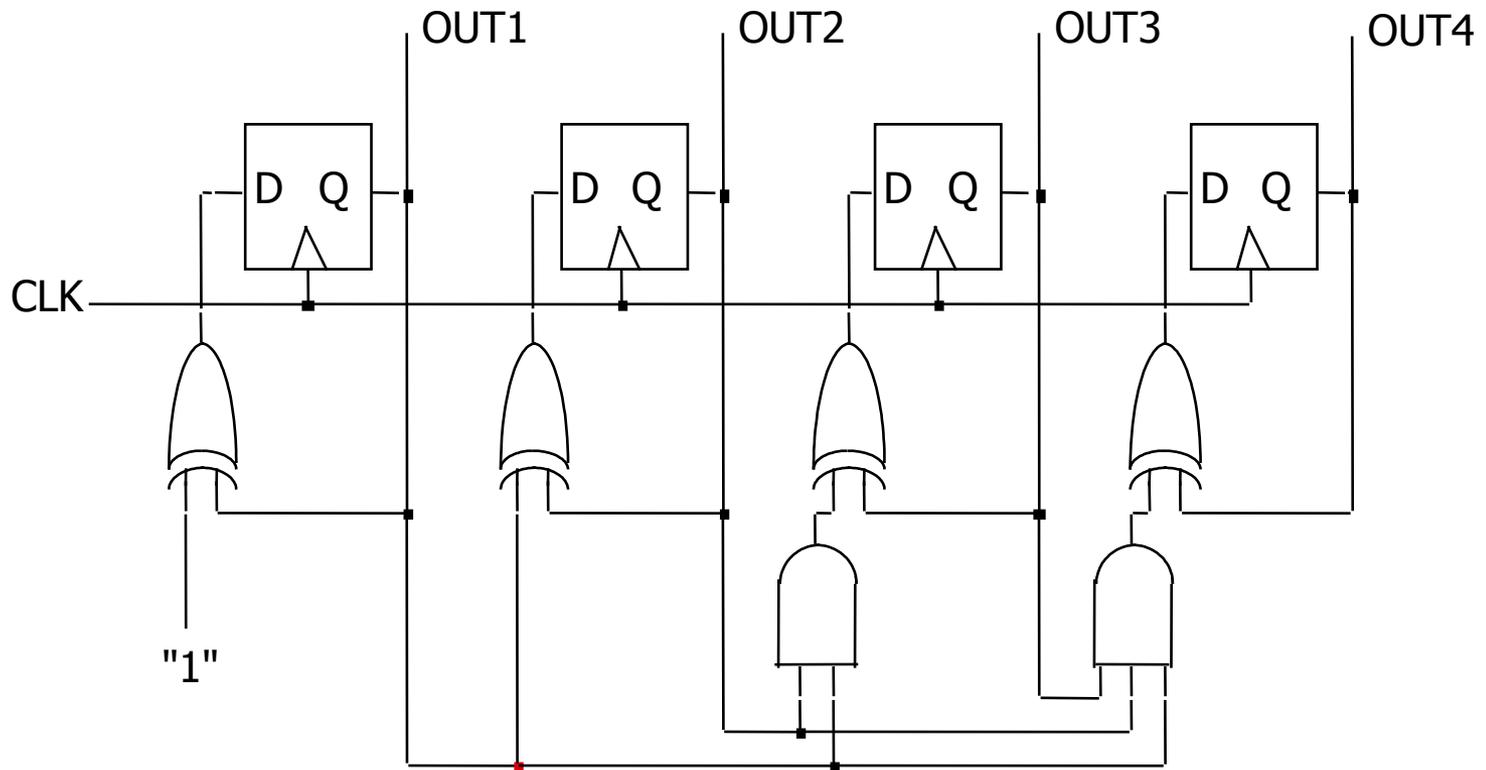
Contador Binário Crescente

	clk	Q ₂	Q ₁	Q ₀
<i>t</i> ₀	↑	0	0	0
<i>t</i> ₁	↑	0	0	1
<i>t</i> ₂	↑	0	1	0
<i>t</i> ₃	↑	0	1	1
<i>t</i> ₄	↑	1	0	0
<i>t</i> ₅	↑	1	0	1
<i>t</i> ₆	↑	1	1	0
<i>t</i> ₇	↑	1	1	1

Contador Binário Síncrono - FF tipo T com enable e clear



Contador Binário - FF Tipo D



**Contador Binário Síncrono
(crescente)**

Contador Binário com FF Tipo D

- ❑ **Lógica entre os registradores (não somente um multiplexer)**
 - **XOR decide quando um determinado bit deve ser “toggled”**
 - **Sempre para o bit de mais baixa ordem; somente quando o primeiro bit é verdade para o segundo bit, e assim por diante**

Contador 4 Bits em VHDL

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_unsigned.all;  
  
ENTITY contador IS  
PORT(CLK, CLR : IN std_logic;  
      Q : OUT std_logic_vector(3 downto 0));  
END contador;
```

Contador 4 Bits em VHDL

```
ARCHITECTURE arquitetura OF contador IS
  SIGNAL tmp: std_logic_vector(3 downto 0);
BEGIN

  PROCESS (CLK, CLR)
  BEGIN
    IF (CLR = '1') THEN
      tmp = "0000";
    ESLIF (CLK'event AND CLK = '1') THEN
      tmp = tmp + 1;
    END IF;

  END PROCESS;

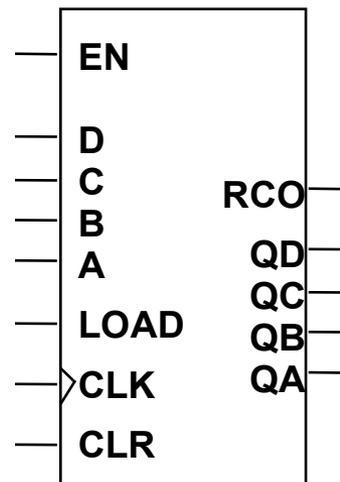
  Q = tmp;

end arquitetura;
```

Contador Binário Crescente Síncrono de 4-bits

□ Componente padrão em muitas aplicações

- FF ativado na transição positiva com entradas de load e clear
- Dado carregado das entradas paralelas D, C, B, A
- Entradas de Enable: precisam estar em 1 para contagem
- RCO: saída de ripple-carry utilizada para cascadeamento
 - '1' quando o contador está em seu estado mais alto (1111)
 - implementado usando uma porta AND



Contador Binário Crescente Síncrono de 4-bits

