

# Sistemas Digitais

ET46B

---

Prof. Eduardo Vinicius Kuhn

*[kuhn@utfpr.edu.br](mailto:kuhn@utfpr.edu.br)*

Curso de Engenharia Eletrônica

Universidade Tecnológica Federal do Paraná



# Capítulo 7

## Contadores e Registradores

- 7.1 Contadores assíncronos
- 7.2 Atraso de propagação em contadores assíncronos
- 7.3 Contadores síncronos (paralelos)
- 7.4 Contadores de módulo  $< 2^N$
- 7.5 Contadores síncronos crescentes/decrescentes
- 7.6 Contadores com carga paralela
- 7.7 Circuitos integrados de contadores síncronos
- 7.8 Decodificando um contador
- 7.9 Análise de contadores síncronos
- 7.10 Projeto de contadores síncronos
- 7.14 Máquinas de estados finitos
- 7.15 Transferência de dados em registradores
- 7.17 Contadores com registradores de deslocamento

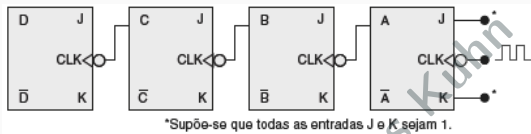
# Objetivos

- Revisitar a operação de contadores assíncronos e **introduzir o conceito de contadores síncronos**.
- Implementar contadores com **módulo  $\leq 2^N$** , **crescentes e decrescentes**, e com **sequências arbitrárias** de contagem.
- Conectar contadores em cascata a fim de produzir faixas de contagens e fatores de divisão de frequência maiores.
- Descrever formas para decodificar contadores.
- Apresentar um **procedimento geral de projeto de circuitos sequenciais**, usando tanto FFs JK quanto FFs D.
- Explicar a forma de operação de diversos tipos de registradores (**e.g., PIPO, SISO, PISO e SIPO**).

Contadores e registradores, com características específicas, podem ser construídos a partir de FFs e portas lógicas.

## PARTE 1 - Contadores

# Contadores assíncronos

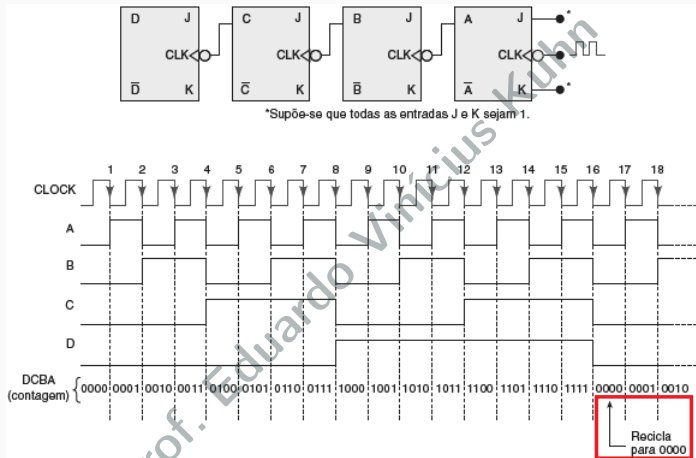


Com respeito à operação de um contador (**assíncrono**) de módulo  $2^N$ , é importante observar que:

- Os pulsos são aplicados apenas na entrada *CLK* do FF *A*.
- A saída do FF *A* funciona como *clock* para o FF *B* e assim por diante.
- Na 16ª borda de descida do *clock*, o contador é “reciclado” e começará um novo ciclo de contagem.

Portanto, em um contador assíncrono (ou contador ondulante), os FFs não mudam de estado com o mesmo pulso de *clock* (i.e., em sincronismo).

# Contadores assíncronos



As saídas dos FFs *D*, *C*, *B* e *A* representam um número binário de 4 bits, em que *D* denota o MSB e *A* o LSB.

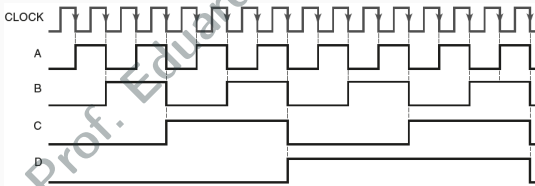


# Divisão de frequência

**Exemplo:** Considere que um contador assíncrono de módulo 16 opera com *clock* de entrada de 16 kHz.



A partir disso, determine a frequência da forma de onda observada nas saídas *A*, *B*, *C* e *D*.

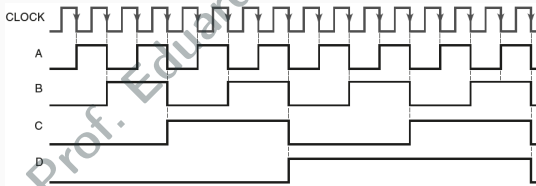


# Divisão de frequência

**Exemplo:** Considere que um contador assíncrono de módulo 16 opera com *clock* de entrada de 16 kHz.



A partir disso, determine a frequência da forma de onda observada nas saídas *A*, *B*, *C* e *D*.



**R:** A forma de onda na saída *A* tem frequência de 8 kHz, na saída *B*, de 4 kHz, na saída *C*, de 2 kHz, e na saída *D*, de 1 kHz.

Em um contador, o sinal de saída do último FF (MSB) tem frequência igual à do *clock* de entrada dividida pelo módulo do contador.

# Contadores assíncronos

**Exemplo:** Um contador é necessário para contar o número de itens que passam por uma esteira de transporte. Uma fotocélula combinada a uma fonte de luz é usada para gerar um único pulso cada vez que um item passa pelo feixe de luz. O contador deve ser capaz de contar 1000 itens. Quantos FFs são necessários?

**Exemplo:** O primeiro passo envolvido na construção de um relógio digital é obter um sinal de 60 Hz para gerar uma forma de onda de 1 Hz. Essa forma de onda entra em uma série de contadores, que contam os segundos, minutos e horas. Quantos FFs são necessários para implementar um contador de módulo 60?

# Contadores assíncronos

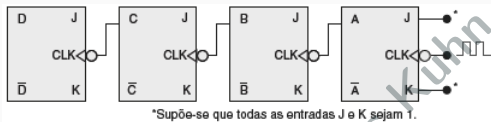
**Exemplo:** Um contador é necessário para contar o número de itens que passam por uma esteira de transporte. Uma fotocélula combinada a uma fonte de luz é usada para gerar um único pulso cada vez que um item passa pelo feixe de luz. O contador deve ser capaz de contar 1000 itens. Quantos FFs são necessários?

**R:** Como  $2^{10} = 1024_{10}$ , 10 FFs são necessários.

**Exemplo:** O primeiro passo envolvido na construção de um relógio digital é obter um sinal de 60 Hz para gerar uma forma de onda de 1 Hz. Essa forma de onda entra em uma série de contadores, que contam os segundos, minutos e horas. Quantos FFs são necessários para implementar um contador de módulo 60?

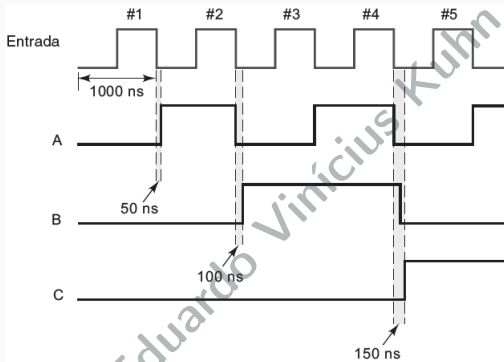
**R:** Não há potência inteira de 2 que seja igual à 60; logo, 6 FFs são usados para produzir um contador de módulo 64.

# Atraso de propagação em contadores assíncronos



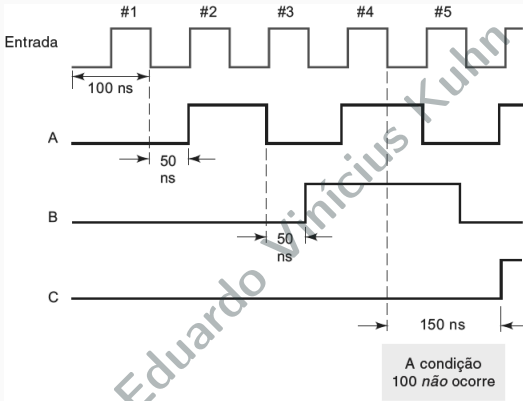
- Contadores assíncronos têm uma grande desvantagem decorrente do fato que **cada FF subsequente é disparado pela transição de saída do FF precedente.**
- O **atraso entre as respostas de FFs sucessivos** (e.g.,  $t_{pd}$  de 5 a 20 ns por FF) **pode ser problemático.**
- Os atrasos de propagação se acumulam tal que **o  $N$ -ésimo FF não muda de estado por um intervalo  $N \times t_{pd}$** , após a transição do *clock* de entrada.

# Atraso de propagação em contadores assíncronos



Note que a saída do FF *A* comuta 50 ns, após a borda de descida do clock de cada pulso de entrada; apesar disso, o contador opera “adequadamente” dado que  $T_{\text{clock}} = 1000 \text{ ns}$ .

# Atraso de propagação em contadores assíncronos



**Caso  $T_{\text{clock}} = 100 \text{ ns}$ , a condição de contagem  $CBA = 100_2$  nunca ocorrerá**, uma vez que a frequência do *clock* de entrada é muito alta.



# Atraso de propagação em contadores assíncronos

Devido ao acúmulo de atrasos de propagação dos FFs, é possível mostrar que, **para uma operação adequada,**

$$T_{\text{clock}} \geq N \times t_{\text{pd}}$$

ou, **em termos de frequência máxima de operação,**

$$f_{\text{max}} \leq \frac{1}{N \times t_{\text{pd}}}.$$

Note que, à medida que o número de FFs  $N$  aumenta, o atraso de propagação total aumenta e  $f_{\text{max}}$  diminui.

Portanto, **a aplicabilidade prática de contadores assíncronos torna-se limitada;** especialmente, em sistemas digitais de alta velocidade e/ou com grande número de bits.

# Atraso de propagação em contadores assíncronos

**Exemplo:** Explique como a frequência máxima dos contadores ondulantes diminui à medida que aumenta o número de FFs?

$$f_{\max} \leq \frac{1}{N \times t_{pd}}$$

**Exemplo:** Determinado FF JK tem um  $t_{pd} = 12$  ns. Qual é o contador de maior módulo que pode ser construído a partir desses FFs que seja capaz operar em uma frequência de até 10 MHz?

# Atraso de propagação em contadores assíncronos

**Exemplo:** Explique como a frequência máxima dos contadores ondulantes diminui à medida que aumenta o número de FFs?

$$f_{\max} \leq \frac{1}{N \times t_{\text{pd}}}$$

**R:** Conforme  $N \rightarrow \infty$ , verifica-se que  $f_{\max} \rightarrow 0$ .

**Exemplo:** Determinado FF JK tem um  $t_{\text{pd}} = 12$  ns. Qual é o contador de maior módulo que pode ser construído a partir desses FFs que seja capaz operar em uma frequência de até 10 MHz?

**R:** Como

$$\begin{aligned} N &\leq \frac{1}{f_{\max} \times t_{\text{pd}}} \\ &\leq 8,333 \end{aligned}$$

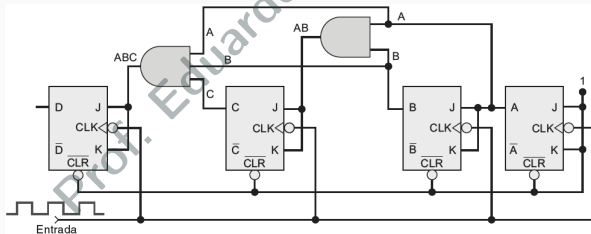
verifica-se que o contador pode ter módulo  $2^8 = 256$ .

“Restrições de tempo” estão se tornando cada vez mais críticas em sistemas digitais de alta velocidade, inviabilizando assim o uso de determinadas abordagens/soluções.

# Contadores síncronos (paralelos)

Em contadores síncronos,

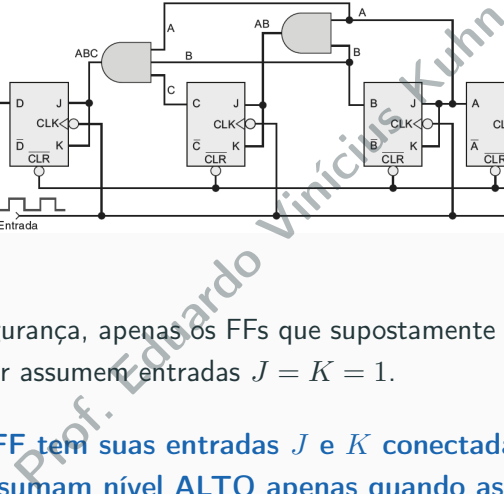
- os FFs são disparados simultaneamente pelo *clock* de entrada.
- Somente o FF *A* (LSB) tem entradas  $J = K = 1$  (constante).
- As entradas  $J$  e  $K$  dos FFs subsequentes são acionadas com base nas saídas dos FFs precedentes.



Note que o circuito de um contador síncrono é “mais complexo”

do que o de um contador assíncrono.

## Universidade Tecnológica Federal do Paraná



-

# Contadores síncronos (paralelos)

Como **vantagem dos contadores síncronos**, tem-se que o tempo total de resposta de um contador síncrono reduz-se à

$$\text{Atraso total} = t_{pd} \text{ do FF} + t_{pd} \text{ da porta AND}$$

Portanto,

**o atraso total não depende do número de FFs do contador**

podendo assim operar com frequência de entrada elevada.

## Contadores síncronos (paralelos)

**Exemplo:** Determine  $f_{\max}$  para o contador síncrono se o  $t_{pd}$  de cada FF for 50 ns e o  $t_{pd}$  de cada porta AND for 20 ns. Compare esses valores com  $f_{\max}$  para um contador assíncrono de módulo 16.

**Exemplo:** O que deve ser feito para mudar o módulo de um contador de 16 para 32?



# Contadores síncronos (paralelos)

**Exemplo:** Determine  $f_{\max}$  para o contador síncrono se o  $t_{pd}$  de cada FF for 50 ns e o  $t_{pd}$  de cada porta AND for 20 ns. Compare esses valores com  $f_{\max}$  para um contador assíncrono de módulo 16.

**R:** Para o contador síncrono (paralelo),

$$f_{\max} \leq \frac{1}{t_{pd} \text{ do FF} + t_{pd} \text{ da porta AND}} = 14,3 \text{ MHz}$$

enquanto, para o contador assíncrono,

$$f_{\max} \leq \frac{1}{N \times t_{pd}} = 5 \text{ MHz.}$$

**Exemplo:** O que deve ser feito para mudar o módulo de um contador de 16 para 32?

**R:** Adicionar mais um FF, tal que  $2^5 = 32$ , e realizar as conexões.

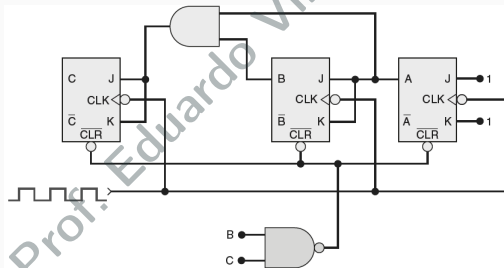
Como modificar o circuito de um contador para que ele reinicie antes do último estado?

Como modificar o circuito de um contador para que ele reinicie antes do último estado?

Basta incluir um circuito que detecte o estado desejado e realize o *reset* dos FFs.

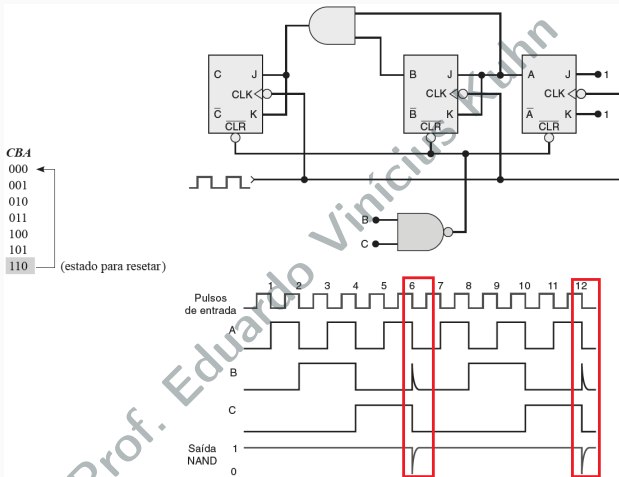
# Contadores de módulo $< 2^N$

- O contador síncrono projetado está limitado ao valor do módulo que é igual a  $2^N$ , em que  $N$  é o número de FFs.
- Contudo, **um contador pode ser modificado para gerar um módulo  $< 2^N$** , fazendo com que o contador pule estados.



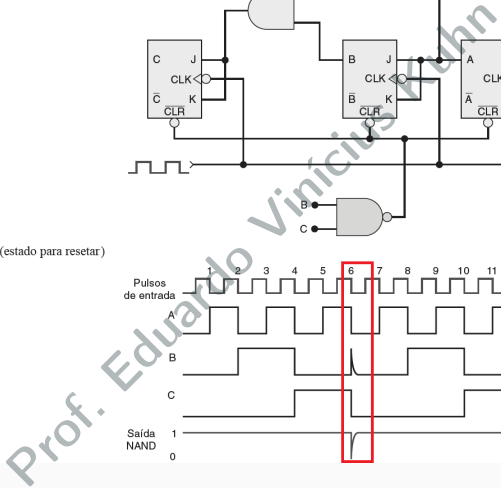
Os FFs  $A$ ,  $B$  e  $C$  mudam de estado à medida que os pulsos são aplicados na entrada de *clock*.

# Contadores de módulo $< 2^N$



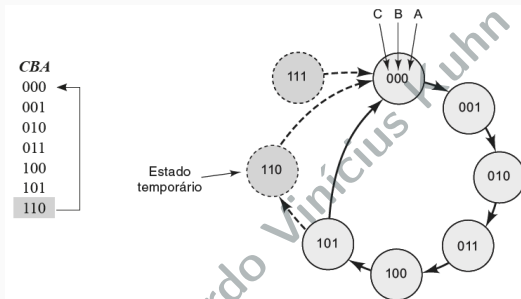
O contador chega ao estado 110, mas se mantém por apenas alguns nanossegundos antes de “reciclar” para 000; logo, o contador vai de 000 (zero) a 101 (cinco) e então “recicla”, pulando os estados 110 e 111.

## Universidade Tecnológica Federal do Paraná



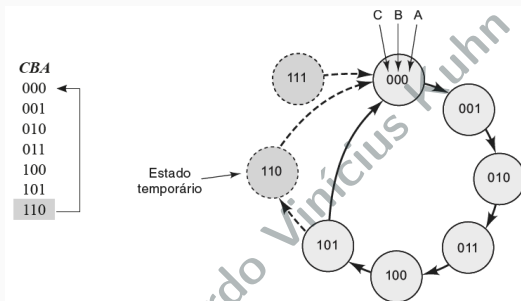
a  $B$  seja usada para acionar outros circuitos externos ao co

# Diagrama de estados de um contador de módulo $< 2^N$



- Cada círculo representa um dos possíveis estados e as setas indicam transições de estado em resposta ao *clock*.
- Passa por 6 estados diferentes; por isso, **trata-se de um contador de módulo 6**.

# Diagrama de estados de um contador de módulo $< 2^N$

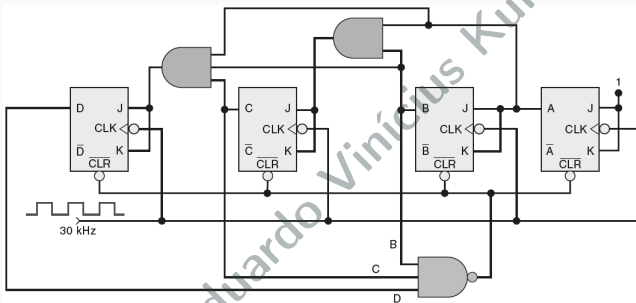


- Não há nenhuma seta entrando no estado 111 porque o contador nunca avançará até esse estado.
- **Caso o estado 111 ocorra indevidamente**, o contador é imediatamente “reciclado” para 000 pela lógica do circuito.



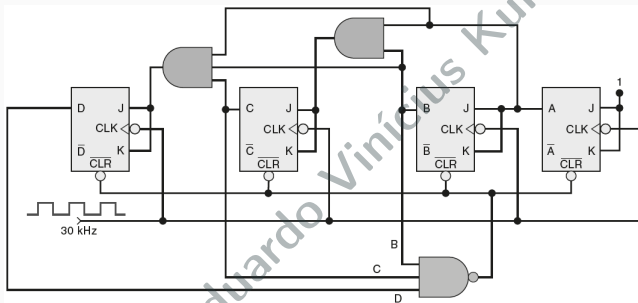
# Contadores de módulo $< 2^N$

**Exemplo:** Qual é o módulo do contador e a frequência na saída  $D$ ?



# Contadores de módulo $< 2^N$

**Exemplo:** Qual é o módulo do contador e a frequência na saída  $D$ ?



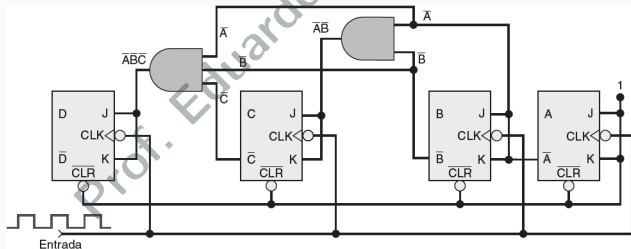
**R:** Um contador de 4 bits tem módulo 16; entretanto, como o *reset* ocorre em 1110, o módulo do contador fica limitado à 14. Por consequência, a frequência de saída  $D$  é

$$\frac{30 \text{ kHz}}{14} = 2,14 \text{ kHz}$$

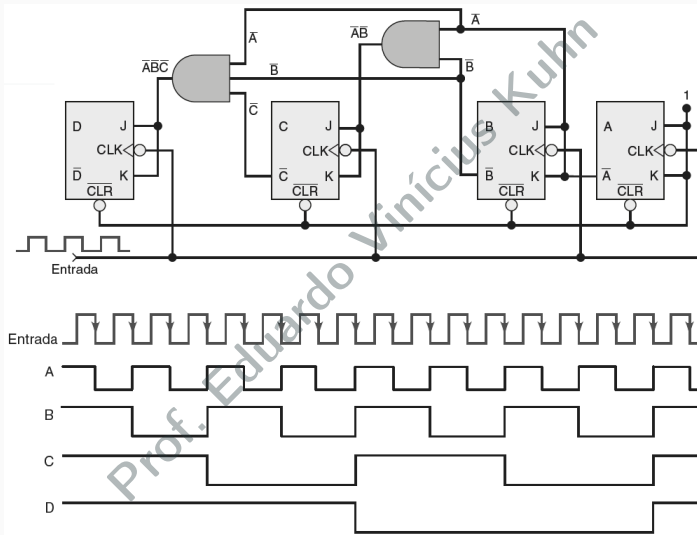
Não usar as entradas assíncronas dos FFs (PRESET e CLEAR) elimina a necessidade de lidar com estados temporários e a ocorrência de possíveis *glitches*.

# Contadores síncronos decrescentes

- Um contador **decrescente** síncrono pode ser criado de maneira semelhante, usando as saídas invertidas para controlar as entradas dos FFs de ordem mais alta.
- Os FFs são “pré-setados”, tal que  $DCBA = 1111$ .



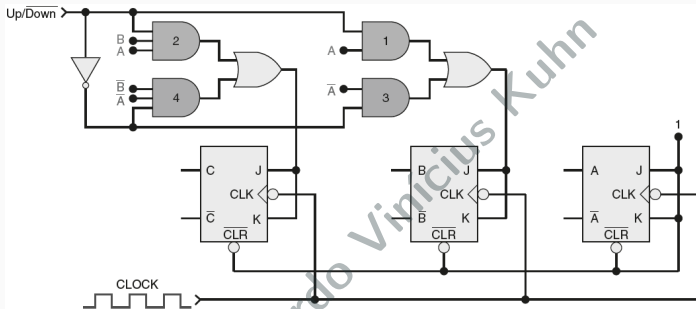
# Contadores síncronos decrescentes



Observe que a sequência de contagem *DCBA* inicia em 1111 e vai até 0000.

Como implementar contadores síncronos  
crescentes/decrescentes no mesmo circuito?

# Contadores síncronos crescentes/decrescentes



A **entrada Up/Down** controla se as entradas  $J$  e  $K$  dos FFs subsequentes são acionadas pelas saídas

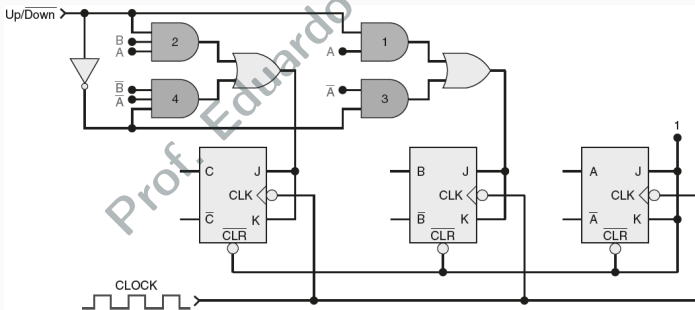
- **normais** (contagem crescente); ou
- **invertidas** (contagem decrescente)

dos FFs precedentes (de menor ordem).

# Contadores síncronos crescentes/decrescentes

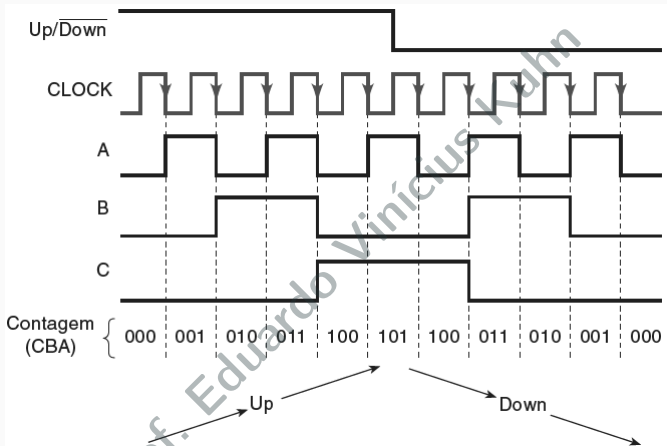
Enquanto a **entrada Up/Down** estiver em nível

- **ALTO**, as portas AND 1 e 2 estarão habilitadas, e as portas 3 e 4 estarão desabilitadas.
- **BAIXO**, as portas AND 3 e 4 estarão habilitadas, e as portas 1 e 2 estarão desabilitadas.



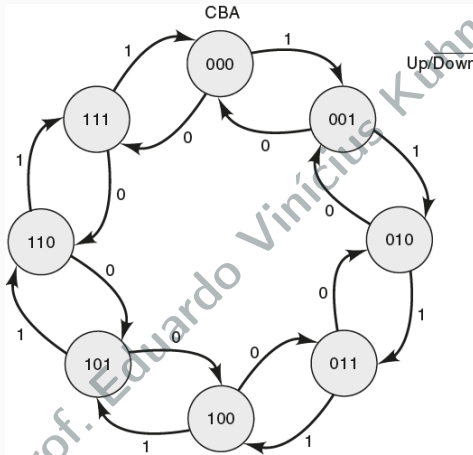


# Contadores síncronos crescentes/decrescentes



Nos primeiros 5 pulsos de *clock*,  $Up/\overline{Down} = 1$  e a contagem é crescente. Por outro lado, para os últimos 5 pulsos,  $Up/\overline{Down} = 0$  e a contagem é decrescente.

# Contadores síncronos crescentes/decrecentes



Existem duas setas deixando cada estado, indicando uma **transição condicional** relacionada a entrada  $Up/\overline{Down}$ .

Como iniciar a contagem em um dado valor?

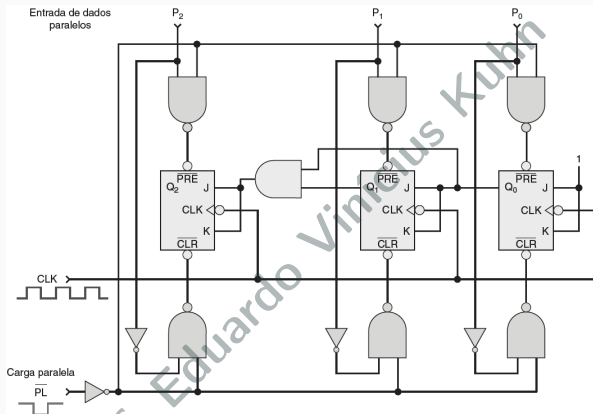
Como iniciar a contagem em um dado valor?

Inserindo um circuito para realizar o  
PRESET/CLEAR dos FFs.

# Contadores com carga paralela

- Alguns contadores podem ser inicializados em um dado valor/estado.
- Esse processo de inicialização, denominado “carga paralela” do contador, pode ser realizado de forma
  - **assíncrona**, usando as entradas de PRESET e CLEAR dos FFs para definir diretamente os estados dos bits;
  - **síncrona**, sendo o valor desejado carregado na transição ativa do *clock*.
- A carga síncrona é preferível em sistemas de alta velocidade, visando evitar problemas com *glitches* bem como garantir sincronismo nas operações.

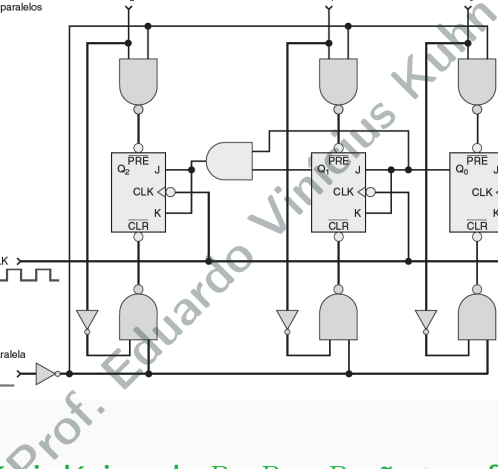
# Contadores com carga paralela



A carga no contador é realizada da seguinte maneira:

- **Aplique a entrada desejada em  $P_2$ ,  $P_1$  e  $P_0$ .**
- **Aplique um pulso na entrada de carga paralela  $\overline{PL}$ .**

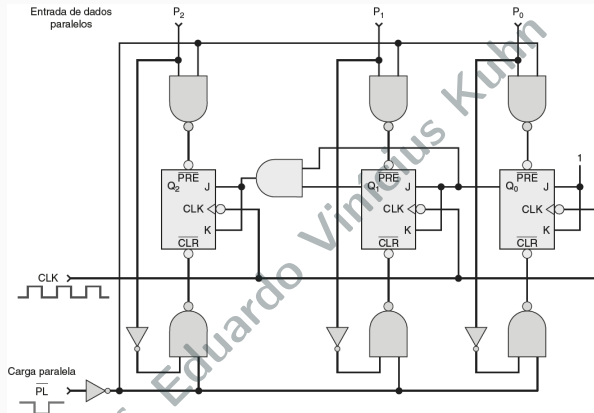
## Universidade Tecnológica Federal do Paraná



Prof.

Krishna Kumar P. D., P. D., Bangalore

# Contadores com carga paralela

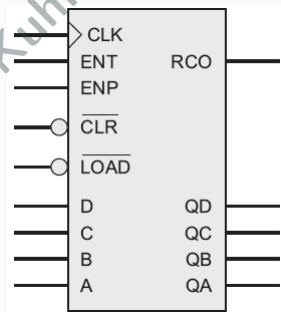


Quando  $\overline{PL}$  retorna ao nível ALTO, os FFs voltam a responder às entradas de *clock*; assim, o contador pode prosseguir com a contagem a partir do valor carregado.



# CI's de contadores síncronos

- Entradas de carga paralela:  $DCBA$
- Saídas:  $Q_D Q_C Q_B Q_A$
- *Clock*: CLK
- Carga paralela:  $\overline{LOAD}$
- *Reset*:  $\overline{CLR}$
- Habilitar contagem: ENT e ENP
- Indica o estado final: RCO



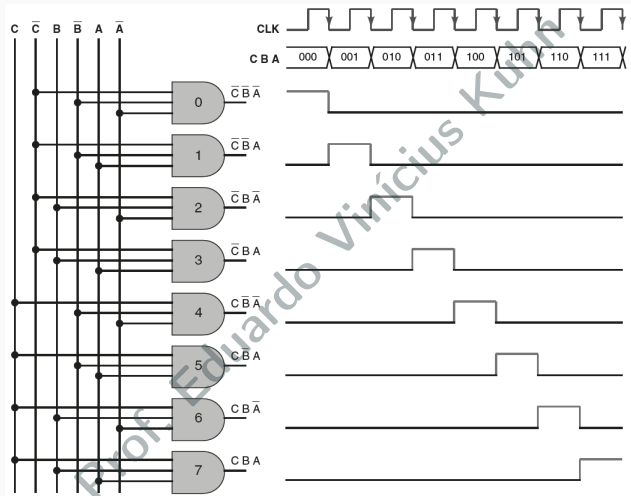
A saída RCO é útil quando dois ou mais CI's de contadores são conectados em um arranjo de múltiplos estágios para criar contadores maiores.

Como o estado atual de um contador pode ser decodificado (identificado)?

# Decodificando um contador

- Decodificar o conteúdo de um contador **significa identificar estados, visando gerar saídas correspondentes.**
- Essa operação é **fundamental em aplicações** onde
  - onde a contagem precisa ser visualizada; ou
  - onde é necessário **controlar a temporização ou o sequenciamento** de operações.
- Um circuito lógico pode ser usado para decodificar (i.e., identificar) um determinado estado de um contador.
- **Uma malha de decodificação é um circuito lógico que gera uma saída diferente para cada estado do contador.**

# Exemplo de uma malha de decodificação



Em uma malha de decodificação, cada estado do contador é decodificado (com uma porta AND) em uma saída diferente (e.g.,  $2^3 = 8$  estados/saídas).

# Decodificando um contador

**Exemplo:** Quantas portas AND são necessárias para decodificar completamente todos os estados de um contador de módulo 32?

**Exemplo:** Quais são as entradas da porta que decodificam a contagem 21 de um contador módulo 32?

**Exemplo:** Quantas portas AND são necessárias para decodificar completamente um contador de 6 bits?

# Decodificando um contador

**Exemplo:** Quantas portas AND são necessárias para decodificar completamente todos os estados de um contador de módulo 32?

**R:** Um contador de módulo 32 tem 32 estados possíveis; logo, 32 portas AND são necessárias.

**Exemplo:** Quais são as entradas da porta que decodificam a contagem 21 de um contador módulo 32?

**R:**  $EDCBA = 10101_2$ ; logo,  $E\overline{D}C\overline{B}A$ .

**Exemplo:** Quantas portas AND são necessárias para decodificar completamente um contador de 6 bits?

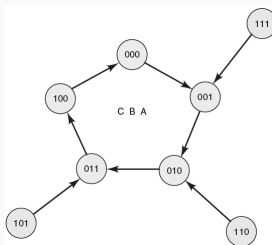
**R:** Como  $2^6 = 64$ , são necessárias 64 portas AND.

Como analisar e projetar contadores  
(síncronos)?

# Análise de contadores síncronos

- A análise de um contador é facilitada por meio de uma **tabela de estados**, a qual relaciona o estado atual ao próximo.
- A partir dessa tabela de estados, **torna-se possível elaborar um diagrama de transição de estados e vice-versa**.
- Ambas as representações descrevem o comportamento de um contador, i.e., o que ocorre a cada pulso de *clock*.

Estado ATUAL			PRÓXIMO estado		
C	B	A	C	B	A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0





# Análise de contadores síncronos

- A partir dessa tabela de estados, é possível **definir os sinais de controle para os FFs** (a depender do tipo de FF usado).
- Dispositivos lógicos programáveis (e.g., FPGAs) **utilizam FFs do tipo D como componentes de memória**.
- Por isso, **é usual projetar circuitos sequenciais utilizando “FFs D”**, já que isso simplifica a implementação em PLDs.
- **O uso de “FFs D” permite obter diretamente as equações de excitação, facilitando a síntese do circuito lógico.**

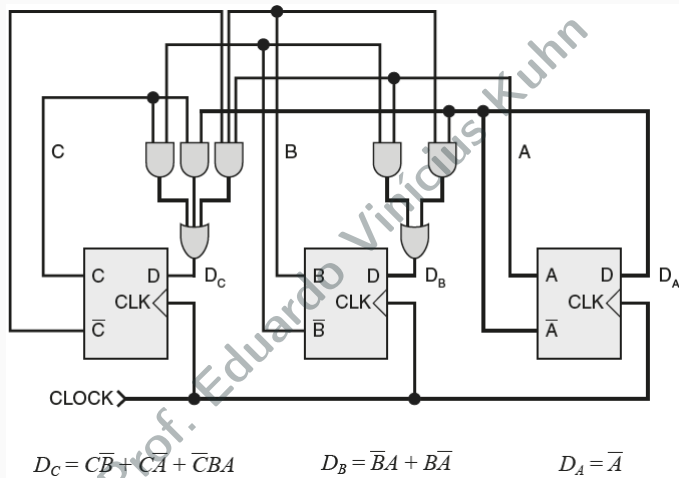
Estado ATUAL			Entradas de controle			PRÓXIMO estado		
C	B	A	$D_C$	$D_B$	$D_A$	C	B	A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

$$D_C = C\bar{B} + C\bar{A} + \bar{C}BA$$

$$D_B = \bar{B}A + B\bar{A}$$

$$D_A = \bar{A}$$

# Análise de contadores síncronos



O circuito de controle de um contador com FFs do tipo D costuma ser mais complexo do que usando FFs JK, mas tem a metade do número de entradas.

# Projeto de contadores síncronos

- Em determinadas situações, um “contador” deve seguir uma **sequência arbitrária de contagem**, e.g.
  - 000, 010, 101, 001, 110, 000...
  - 110, 001, 101, 010, 000, 110...
  - 1010, 1011, 1110, 1010...
- Diferentes **métodos de projeto** de contadores que sigam sequências arbitrárias estão disponíveis, os quais utilizam
  - FFs JK; ou
  - **FFs D (como comumente usado em circuitos sequenciais).**
- **A mesma técnica de projeto de contadores pode ser adotada para outros tipos de circuitos sequenciais.**

# Metodologia de projeto de circuitos sequenciais

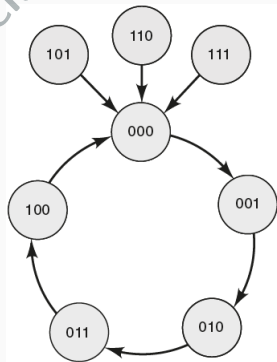
- 1) **Determine o número de bits (i.e., FFs)** necessários para uma dada sequência de contagem.
- 2) **Desenhe o diagrama de transição de estados**, contendo todos os estados (inclusive os que não serão utilizados).
- 3) A partir do diagrama de transição de estados, **monte a tabela de estados**, relacionando os estados atuais aos próximos.
- 4) Escolha o tipo de FF, acrescente colunas na tabela de estados para cada entrada  $J$  e  $K$  ou  $D$  e defina o sinal de cada entrada de controle para produzir a transição ao próximo estado.
- 5) Obtenha as **expressões lógicas descrevendo o sinal de controle de cada entrada**  $J$  e  $K$  ou  $D$ .
- 6) **Implemente os circuitos lógicos a partir das expressões.**

# Metodologia de projeto de circuitos sequenciais

1) **Determine o número de bits (i.e., FFs)** para uma dada sequência de contagem.

C	B	A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
0	0	0
0	0	1
etc.		

2) **Desenhe o diagrama de transição de estados**, contendo todos os estados.



3) A partir do diagrama de transição de estados, **monte a tabela de estados**, relacionando os estados atuais aos próximos.

	Estado ATUAL			PRÓXIMO estado		
	C	B	A	C	B	A
Linha 1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0
6	1	0	1	0	0	0
7	1	1	0	0	0	0
8	1	1	1	0	0	0

# Metodologia de projeto de circuitos sequenciais

- 4) Escolha o tipo de FF, acrescente colunas na tabela de estados para cada entrada  $J$  e  $K$  e defina o sinal de cada entrada de controle para produzir a transição ao próximo estado.

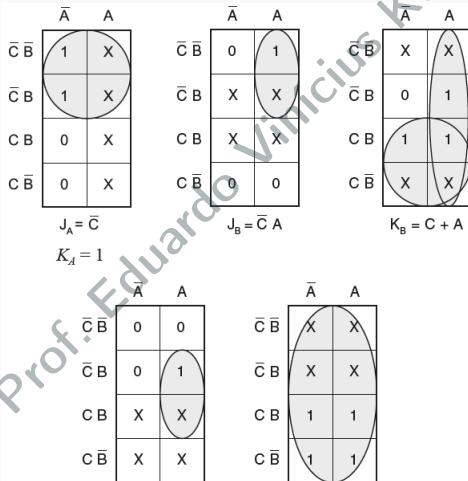
	Estado ATUAL			PRÓXIMO estado			$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
	$C$	$B$	$A$	$C$	$B$	$A$						
Linha 1	0	0	0	0	0	1	0	x	0	x	1	x
2	0	0	1	0	1	0	0	x	1	x	x	1
3	0	1	0	0	1	1	0	x	x	0	1	x
4	0	1	1	1	0	0	1	x	x	1	x	1
5	1	0	0	0	0	0	x	1	0	x	0	x
6	1	0	1	0	0	0	x	1	0	x	x	1
7	1	1	0	0	0	0	x	1	x	1	0	x
8	1	1	1	0	0	0	x	1	x	1	x	1

Tabela de transição de estados de FFs JK e D.

Transição	$Q_n$	$Q_{n+1}$	$J$	$K$	$D$
$0 \rightarrow 0$	0	0	0	x	0
$0 \rightarrow 1$	0	1	1	x	1
$1 \rightarrow 0$	1	0	x	1	0
$1 \rightarrow 1$	1	1	x	0	1

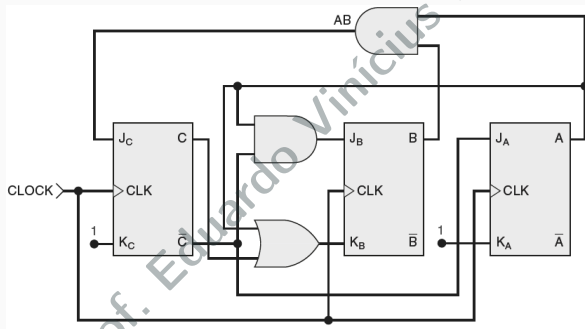
# Metodologia de projeto de circuitos sequenciais

5) Obtenha as **expressões lógicas** descrevendo o sinal de controle de cada entrada  $J$  e  $K$ .





## 6) Implemente os circuitos lógicos a partir das expressões.



E, o que ocorre quando FFs D são utilizados?

# Metodologia de projeto de circuitos sequenciais

- 4) Escolha o tipo de FF, acrescente colunas na tabela de estados para cada entrada  $D$  e defina o sinal de cada entrada de controle para produzir a transição ao próximo estado.

Estado ATUAL			PRÓXIMO estado			Entradas de controle		
C	B	A	C	B	A	$D_C$	$D_B$	$D_A$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Tabela de transição de estados de FFs JK e D.

Transição	$Q_n$	$Q_{n+1}$	$J$	$K$	$D$
$0 \rightarrow 0$	0	0	0	x	0
$0 \rightarrow 1$	0	1	1	x	1
$1 \rightarrow 0$	1	0	x	1	0
$1 \rightarrow 1$	1	1	x	0	1

5) Obtenha as **expressões lógicas** descrevendo o **signal de controle de cada entrada  $D$** .

	$\bar{A}$	$A$
$\bar{C}\bar{B}$	0	0
$\bar{C}B$	0	1
$CB$	0	0
$C\bar{B}$	0	0

$$D_C = CBA$$

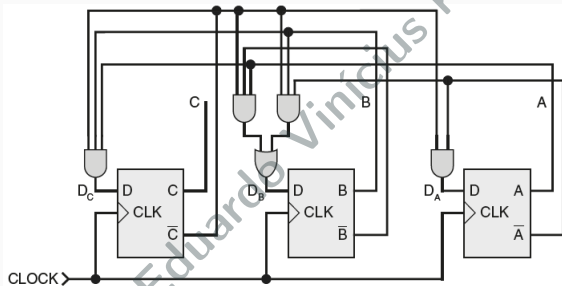
	$\bar{A}$	$A$
$\bar{C}\bar{B}$	0	1
$\bar{C}B$	1	0
$CB$	0	0
$C\bar{B}$	0	0

$$D_B = \bar{C}\bar{B}A + \bar{C}B\bar{A}$$

	$\bar{A}$	$A$
$\bar{C}\bar{B}$	1	0
$\bar{C}B$	1	0
$CB$	0	0
$C\bar{B}$	0	0

$$D_A = \bar{C}\bar{A}$$

## 6) Implemente os circuitos lógicos a partir das expressões.



As expressões lógicas são mais complexas, i.e., requerem mais portas; contudo, o projeto é mais simples já que cada FF D tem apenas uma entrada.

O termo máquina de estado refere-se a um circuito que sequencia um conjunto de estados predeterminados controlados por um *clock* e outros sinais de entrada; logo, contadores são máquinas de estados.

**Sugestão de leitura:** Seção 7.14, a qual trata sobre máquinas de estados finitos (modelos Mealy e Moore).

## PARTE 2 - Registradores



# Transferência de dados em registradores

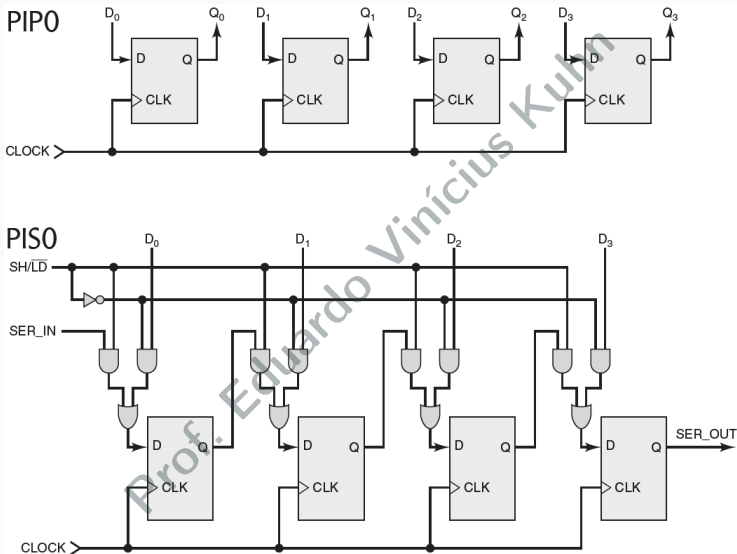
Os diferentes tipos registradores são **classificados de acordo com a maneira pela qual os dados são apresentados ao registrador para armazenamento e pelo modo como saem dele**, i.e.,

- **entrada paralela/saída paralela (PIPO);**
- **entrada serial/saída serial (SISO);**
- **entrada paralela/saída serial (PISO); e**
- **entrada serial/saída paralela (SIPO).**

Vale comentar que:

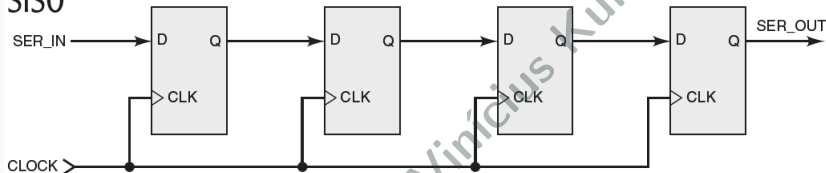
- O fluxo de dados serial por um registrador é, geralmente, chamado de **deslocamento (*shifting*)**.
- A entrada paralela de dados é, usualmente, descrita como **“carga do registrador”**.

# Transferência de dados em registradores

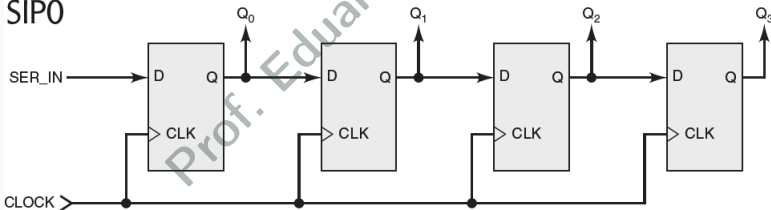


# Transferência de dados em registradores

## SISO



## SIPO



**Sugestão de leitura:** Seção 7.17, a qual trata da implementação de contadores com registradores de deslocamento.

- Em contadores assíncronos, o *clock* é aplicado apenas ao FF LSB sendo os outros disparados pela saída do FF precedente.
- **A frequência máxima  $f_{\max}$  de clock para um contador assíncrono diminui à medida que  $N \rightarrow \infty$ .**
- O módulo de um contador  $2^N$  define o número de estados de contagem possíveis (e o maior fator de divisão de frequência).
- O módulo de um contador pode ser reduzido acrescentando um circuito que faça a “**reciclagem**” **antes do último estado**.
- Os contadores podem ser conectados em cascata para produzir faixas de contagens e fatores de divisão de frequência maiores.
- Em contadores síncronos, todos os FFs são disparados a partir do mesmo *clock* de entrada; logo,  $f_{\max}$  é independente de  $N$ .

- Um contador que possui entrada de dados pode ser carregado com um dado valor inicial de contagem.
- Um contador crescente/decrecente permite tanto contar de forma crescente quanto decrescente.
- Portas lógicas podem ser arranjadas para decodificar (identificar) um determinado estado de um contador.
- A sequência de contagem de um contador pode ser determinada com uma tabela de estados.
- Máquinas de estados podem ser implementadas seguindo o procedimento de projeto apresentado.
- Sistemas digitais podem ser subdivididos em módulos/blocos menores que podem ser interconectados de forma hierárquica.

# Considerações finais

## Exercícios sugeridos:

7.2, 7.3, 7.4, 7.7, 7.12-7.14, 7.16, 7.21, 7.26, 7.31 e 7.43.

de R.J. Tocci, N.S. Widmer, G.L. Moss, *Sistemas digitais: princípios e aplicações*, 12a ed., São Paulo: Pearson, 2019. → (Capítulo 7)

## Para as próximas aulas:

R.J. Tocci, N.S. Widmer, G.L. Moss, *Sistemas digitais: princípios e aplicações*, 12a ed., São Paulo: Pearson, 2019.

Apresentações → (Capítulos 8, 9, 11 e 12)

Até a próxima aula... =)