

# Circuitos Lógicos

## Aula 23

Rodrigo R. Paim

ECI, LAND - UFRJ

09/06/2011



# Circuitos Lógicos

## Aula 23

### Aula Passada

- Aritmética Binária
- Representação binária com sinal
- Complemento a 2
- Adição e Subtração
- Multiplicação

### Aula de Hoje

- Revisão de Flip-Flops
- Máquinas de Estado
- Contadores Módulo N
- Contadores Up/Down

# Flip-Flops

- Elementos de Memória: Armazenam estado
- Dois tipos de transição:
  - Síncrona: com o clock
  - Assíncrona: entradas Set e Reset
- Saída: Estado Atual + Entrada
- Duas Saídas:
  - Q: normal
  - $\bar{Q}$ : invertida
- Entradas: De acordo com o tipo do FF

# Tabela de Excitação

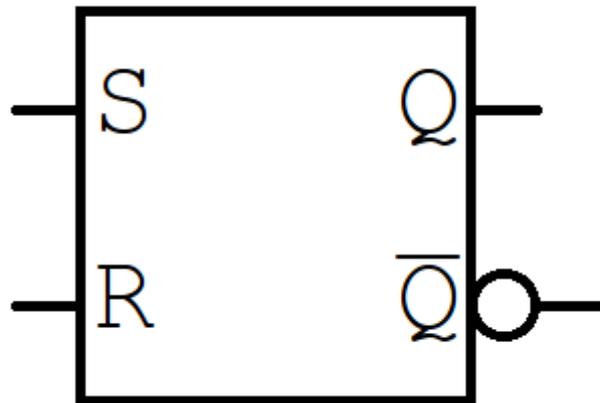
- Facilitam a visualização do funcionamento
- Baseado no estado atual ( $Q$ ) e próximo ( $Q^+$ )

<b>Q</b>	<b>Q<sup>+</sup></b>	<b>Transição (<math>\Delta</math>)</b>
0	0	0
0	1	$\alpha$
1	0	$\beta$
1	1	1

# Tipos de Flip-Flops

## ■ Flip-Flop SR

- Entradas Set (S) e Reset (R) Síncronas (ou não)
- $S = R = 0$ : mantém estado
- $S = R = 1$ : estado proibido

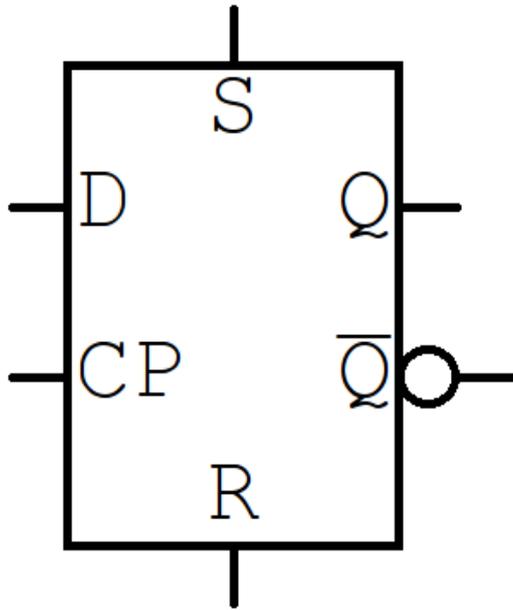


$\Delta$	S	R
0	0	X
$\alpha$	1	0
$\beta$	0	1
1	X	0

# Tipos de Flip-Flops

## ■ Flip-Flop D

- Entrada D repetida em  $Q^+$
- Entradas Assíncronas S, R: desativadas



$\Delta$	D
0	0
$\alpha$	1
$\beta$	0
1	1

# Tipos de Flip-Flops

- Flip-Flop T (Toggle)

- Entrada T controla o funcionamento:

- $T = 0$ : mantém o estado atual

- $T = 1$ : muda o estado

- Implementação:

- Flip-Flop D

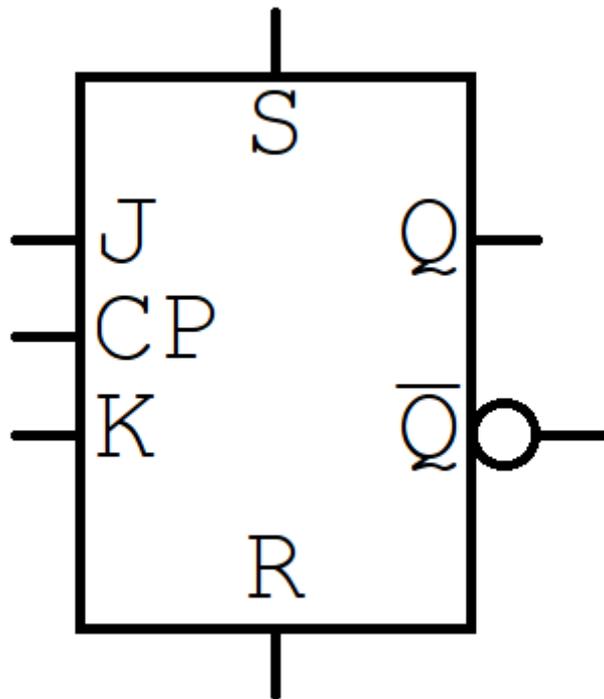
- Flip-Flop JK

$\Delta$	T
0	0
$\alpha$	1
$\beta$	1
1	0

# Tipos de Flip-Flops

- Flip-Flop JK

- $Q^+$ : função de  $Q$ ,  $J$ ,  $K$



$\Delta$	J	K
0	0	X
$\alpha$	1	X
$\beta$	X	1
1	X	0

**Mais “don't care”!**

# Máquinas de Estado

- Modelagem de sistemas
  - Estados
  - Transições
- Estado
  - Representa evento *qualquer* do sistema
  - Dependente do passado
- Transição
  - Evolução do sistema

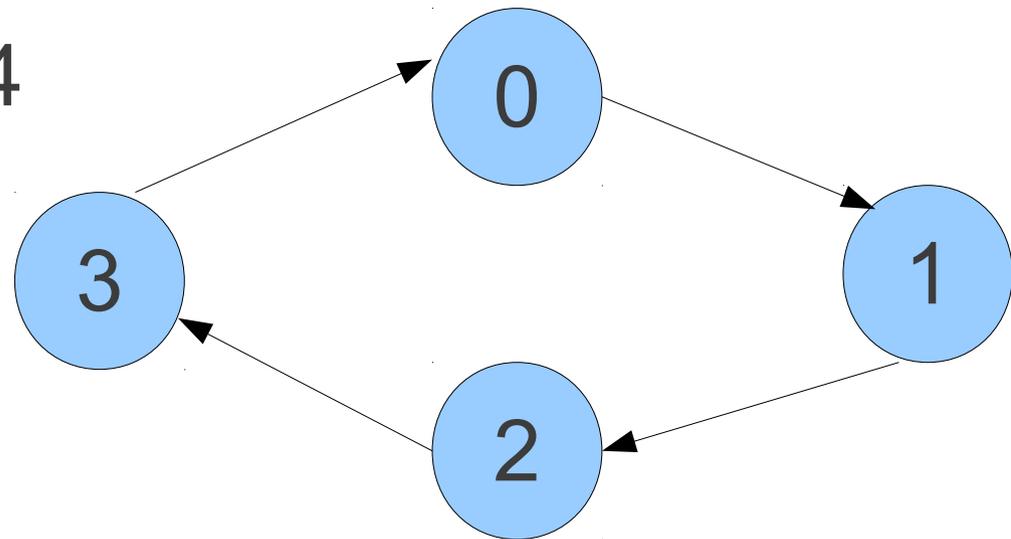
# Máquinas de Estado

- Entrada (\*)
- Estado Atual:
  - Entrada
  - Estado Anterior
- Saída (\*)
  - Transição: Miley
  - Estado: Moore



# Máquinas de Estado

- Exemplo mais comum: contadores
- Contador Módulo N
  - Conta de 0 até N-1 e volta para 0
  - Exemplo: Relógio
- Contador módulo 4



# Máquina de Estados

- Como representar máquinas de estados?
- Ideias?
- Flip-Flops!
  - Estado: configuração momentânea dos FF
  - Transição: Mudanças de Clock
    - Depende das entradas
    - Estado Anterior + Lógica Combinacional
  - Tipos de clock:
    - Assíncrono: transições em tempos diferentes nos FF
    - Síncrono (\*): transições simultâneas em todos os FF

# Máquina de Estados

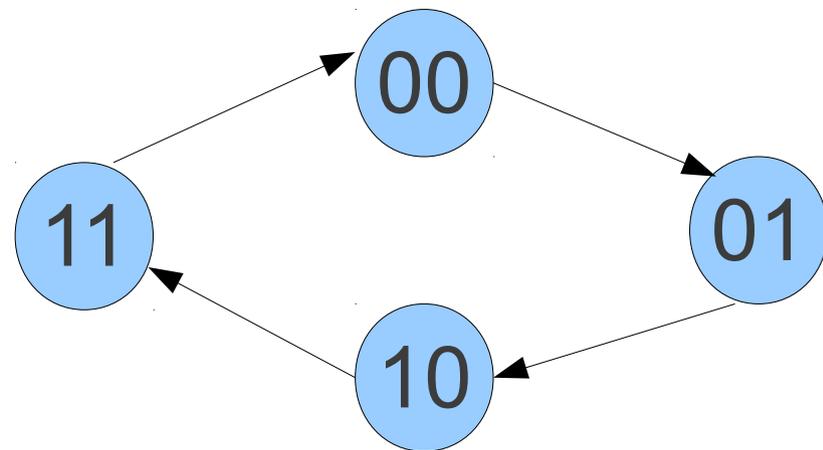
- Qual tipo de FF usar?
  - Qualquer um! Só muda a lógica combinacional!
  - Geralmente FF D e FF JK são mais usados
- Quantos FF usar?
  - 1 FF representa até 2 estados
  - 2 FF representam até 4 estados
  - ...
  - N FF representam até  $2^N$  estados
- Como usar os FF?
  - Analisar a evolução dos estados

# Contador Módulo 4

- Mapear sequência em estados dos FF
- Usar 2 FF quaisquer: X, Y
- Analisar evolução dos FF

X	Y
0	0
0	1
1	0
1	1

X <sup>+</sup>	Y <sup>+</sup>
0	1
1	0
1	1
0	0



# Contador Módulo 4

- Descobrir entradas dos FF
  - Lógica Combinacional: Mapa de Karnaugh
- Flip-Flops Tipo D

<b>X</b>	<b>Y</b>
0	0
0	1
1	0
1	1

<b>X<sup>+</sup></b>	<b>Y<sup>+</sup></b>
0	1
1	0
1	1
0	0

<b>D<sub>X</sub></b>	<b>D<sub>Y</sub></b>
0	1
1	0
1	1
0	0

# Contador Módulo 4

## ■ Mapas de Karnaugh

$X \setminus Y$	0	1
0	0	1
1	1	0

$$D_X = X \text{ xor } Y$$

$X \setminus Y$	0	1
0	1	0
1	1	0

$$D_Y = \overline{Y}$$

# Contador Módulo 4

- Repetir Procedimento
- Flip-Flops Tipo JK

<b>X</b>	<b>Y</b>
0	0
0	1
1	0
1	1

<b>X<sup>+</sup></b>	<b>Y<sup>+</sup></b>
0	1
1	0
1	1
0	0

<b>J<sub>x</sub></b>	<b>K<sub>x</sub></b>
0	X
1	X
X	0
X	1

<b>J<sub>Y</sub></b>	<b>K<sub>Y</sub></b>
1	X
X	1
1	X
X	1

# Contador Módulo 4

$X \setminus Y$	0	1
0	0	1
1	X	X

$$J_X = Y$$

$X \setminus Y$	0	1
0	X	X
1	0	1

$$K_X = Y$$

$X \setminus Y$	0	1
0	1	X
1	1	X

$$J_Y = 1$$

$X \setminus Y$	0	1
0	X	1
1	X	1

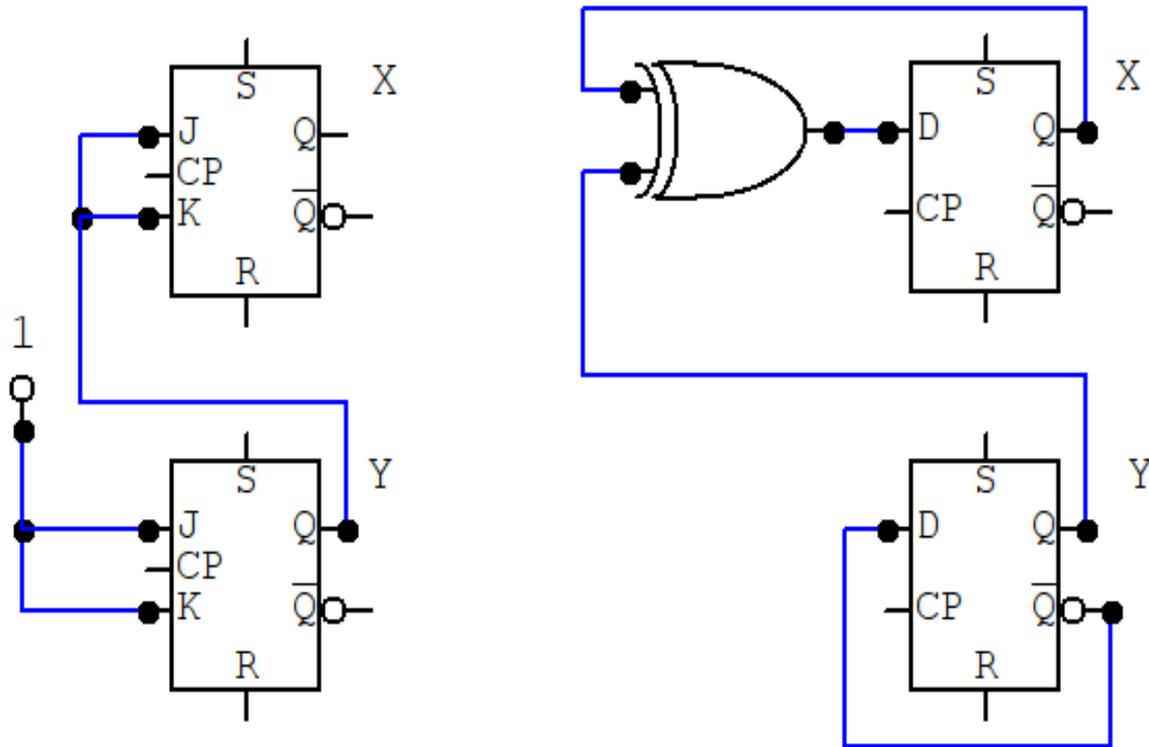
$$K_Y = 1$$

# Contador Módulo 4

- Observações:
- Flip-Flop D:
  - Menos mapas de Karnaugh
  - Circuitos combinacionais maiores
    - Maior custo de projeto
- Flip-Flop JK
  - Mais “don't care”
  - Lógica combinacional mais simples!
    - Menor custo de projeto

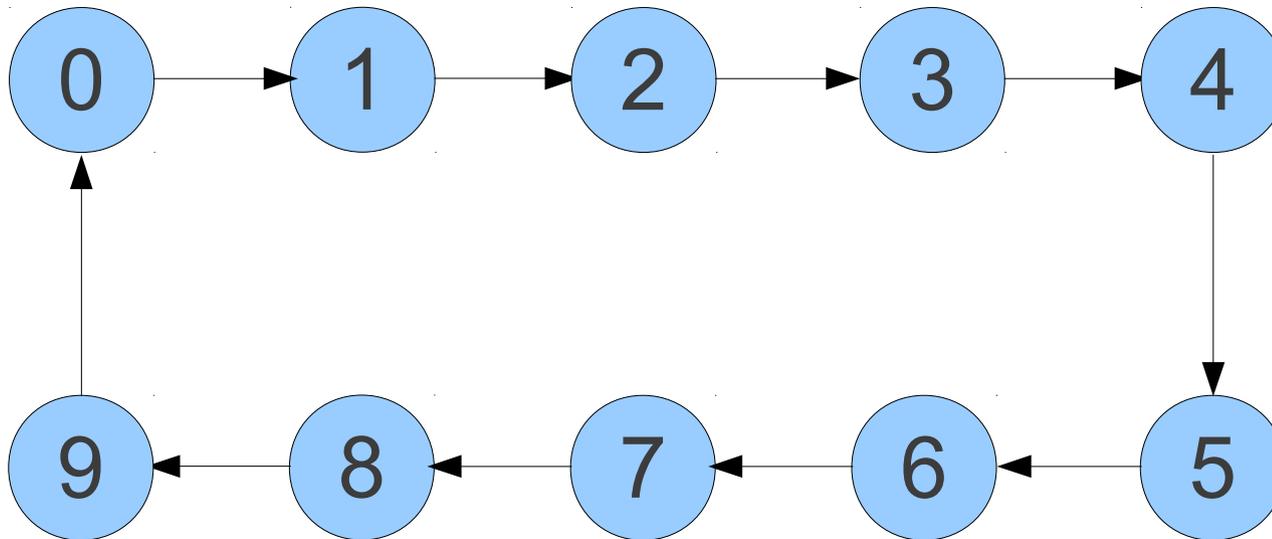
# Contador Módulo 4

## Projeto do Contador:



# Contador Módulo 10

- Conhecido como “década”
- Conta de 0 a 9 e volta para o 0



# Contador Módulo 10

## ■ Problema

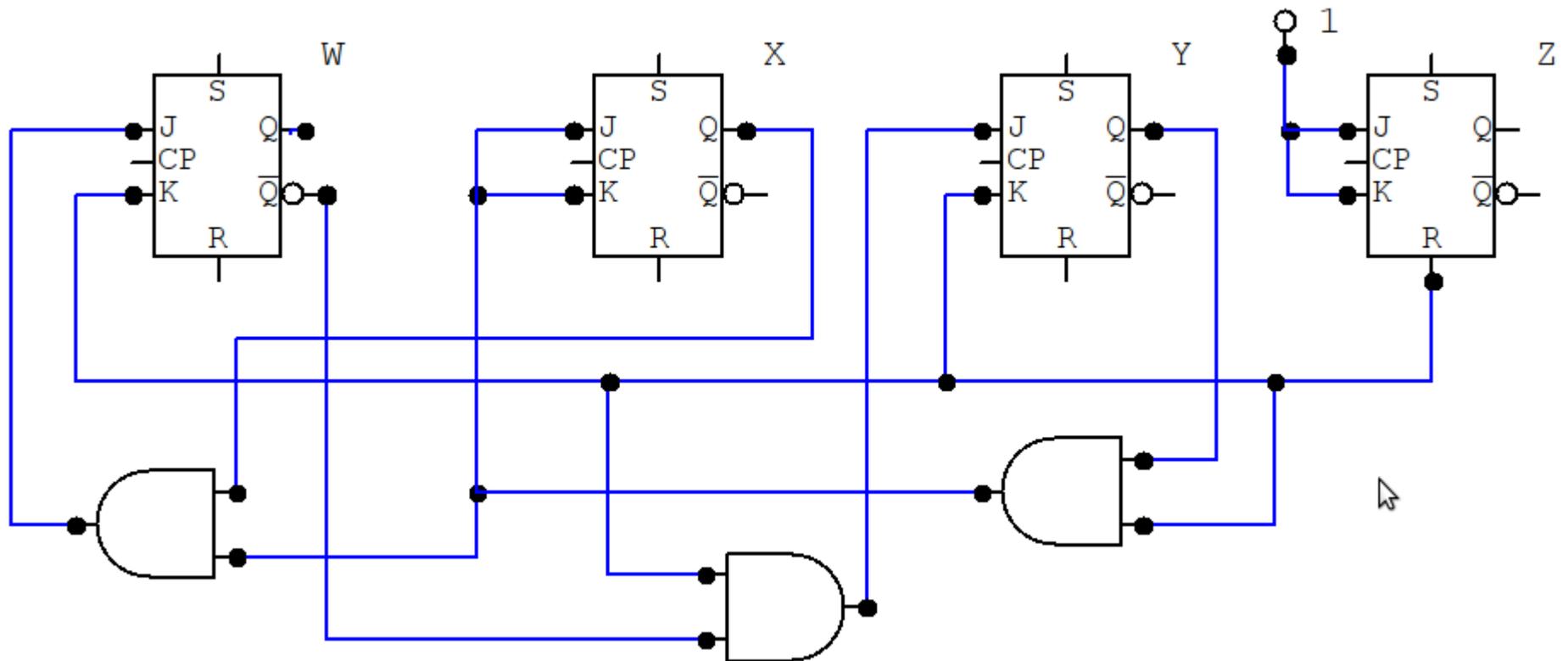
- Década possui 10 estados
- Número de FF: 4
- Estados mapeados: 16
- E os 6 estados restantes?
- Tanto faz! “Don't care”
- Nunca ficar preso em estado “proibido”

# Contador Módulo 10

- Usando Flip-Flop JK:
  - $J_W: X.Y.Z$  e  $K_W: Z$
  - $J_X: Y.Z$  e  $K_X: Y.Z$
  - $J_Y: \overline{W}.Z$  e  $K_Y: Z$
  - $J_Z: 1$  e  $K_Z: 1$
- Pouca lógica combinacional!
  - 3 portas AND: 1 circuito integrado

# Contador Módulo 10

## Projeto do Contador:

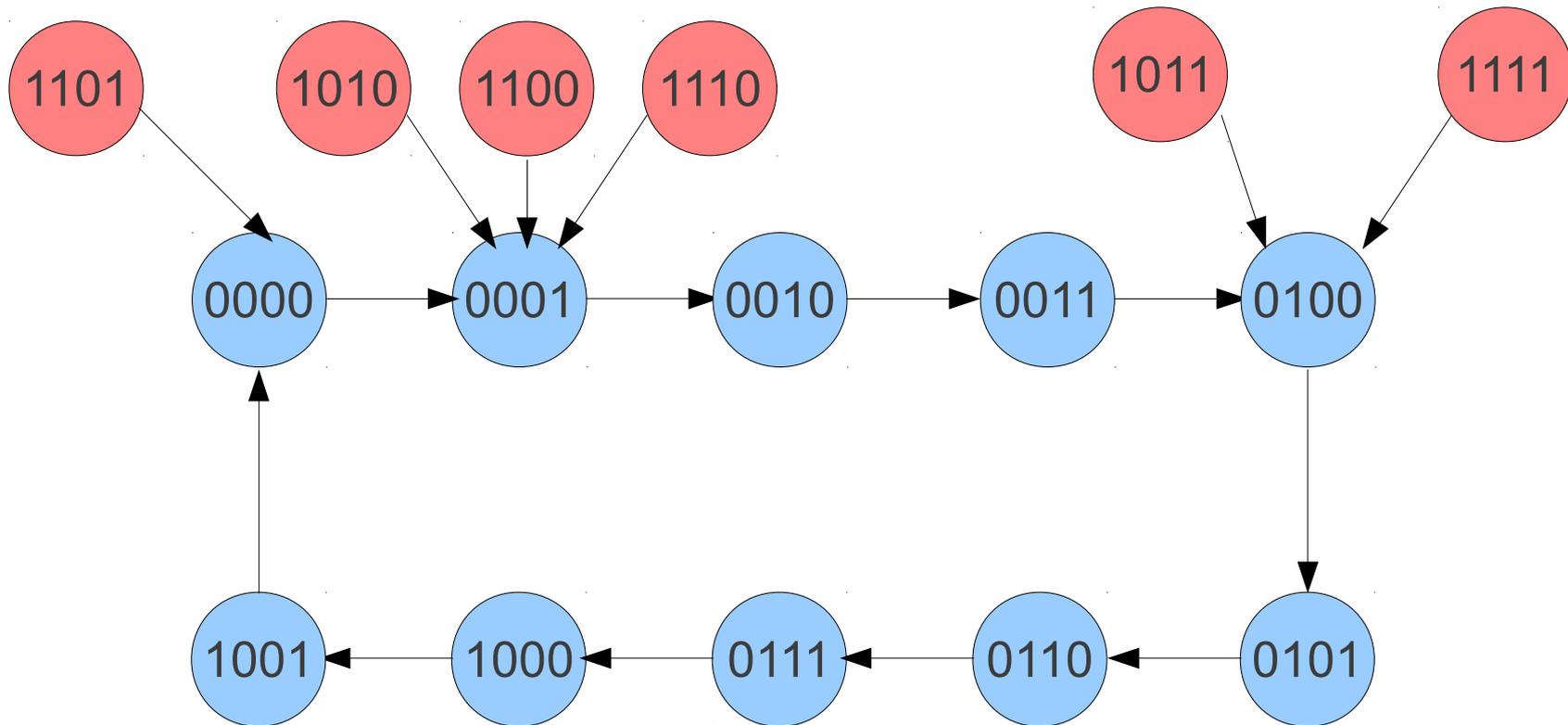


# Contador Módulo 10

- Evolução dos Estados
  - Funciona como uma “prova real”
  - Como os estados “proibidos” se comportam
  - Verificar se existem falhas

# Contador Módulo 10

## ■ Evolução dos Estados

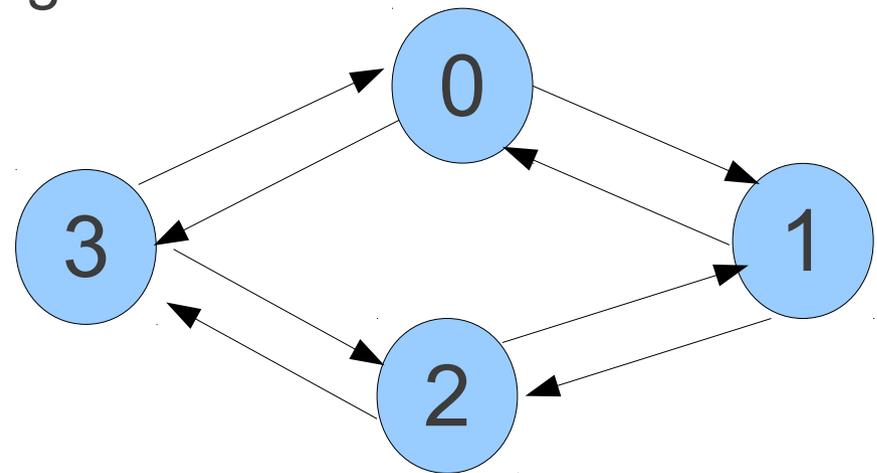


# Contadores

- Exercícios:
  - Implementar contador módulo-8
  - Implementar década usando Flip-Flops D
  - Implementar contador módulo-16

# Contadores

- Contadores Unidirecionais
- Contadores Bidirecionais
  - Conhecidos como Up-Down
  - Entrada (E) da máquina de estados
    - Controla direção de contagem
  - Contador Módulo 4



# Contador Módulo 4 Up-Down

- $E = 0$ : contagem crescente
- $E = 1$ : contagem decrescente
- Tabela-Verdade:
  - 2 estados de  $E$  x 4 números = 8 posições
  - Número de Flip-Flops?
    - Dois: temos apenas 4 estados!
- Entrada  $E$  usada na lógica combinacional!

# Contador Módulo 4 Up-Down

## ■ Evolução dos Flip-Flops:

E	X	Y
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

X <sup>+</sup>	Y <sup>+</sup>
0	1
1	0
1	1
0	0
1	1
0	0
0	1
1	0

# Contador Módulo 4 Up-Down

- Usando Flip-Flop JK:
  - $J_X: E \text{ xor } Y$  e  $K_X: E \text{ xor } Y$
  - $J_Y: 1$  e  $K_Y: 1$
- Pouca lógica combinacional!
  - 1 portas XOR: 1 circuito integrado

# Contadores Up-Down

## ■ Exercícios:

- Implementar contador módulo-8 up-down
- Implementar contador módulo-10 up-down
- Implementar contador módulo-16 up-down

Dúvidas?

Obrigado pela Atenção!