

Capítulo 3–Circuitos Sequenciais

Projeto de Máquinas Mealy e Moore
Profa. Eliete Caldeira

Máquinas Mealy e Moore

▶ Máquina Mealy

- Mais geral
- As saídas dependem do estado atual e das entradas da máquina
- As saídas são representadas nos arcos de transição, separadas das entradas por uma barra

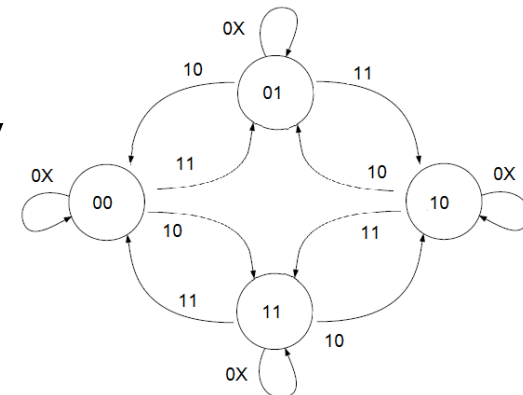
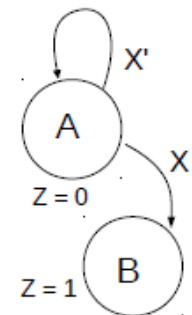
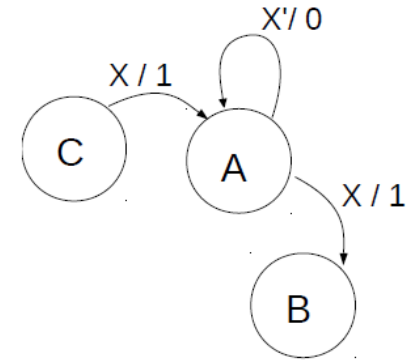
▶ Máquina Moore

- As saídas dependem apenas do estado atual
- São um caso especial de Máquina Mealy
- As saídas são representadas junto dos estados

▶ Máquina Medvedev:

- A saída é tomada diretamente de um flip-flop, ou seja, a saída é o próprio estado dos flip-flops. Um contador é um exemplo.

- ▶ As saídas de uma máquina Moore serão completamente síncronas em relação ao clock do circuito, enquanto saídas de máquinas Mealy podem mudar assincronamente.



Máquinas Mealy e Moore

- Arquitetura padrão

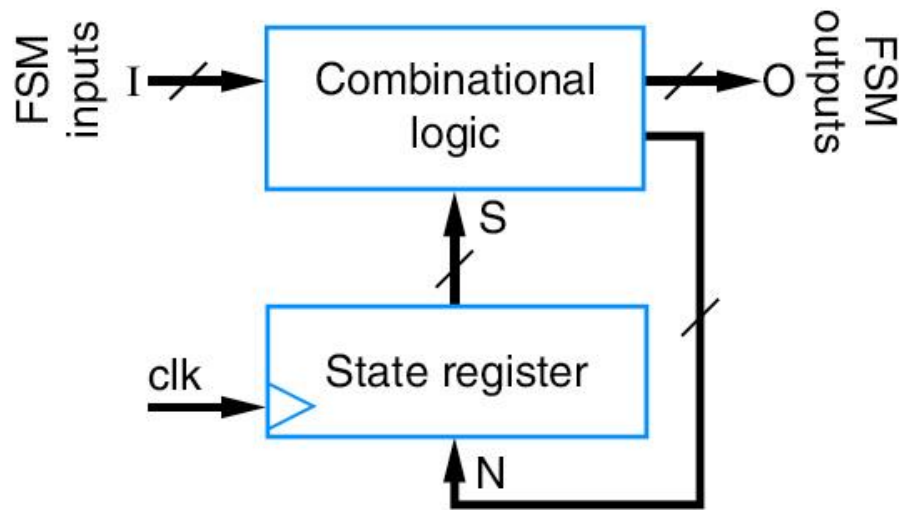


Figure 6.49 Standard controller architecture—general view.

Máquinas Mealy e Moore

► Arquiteturas

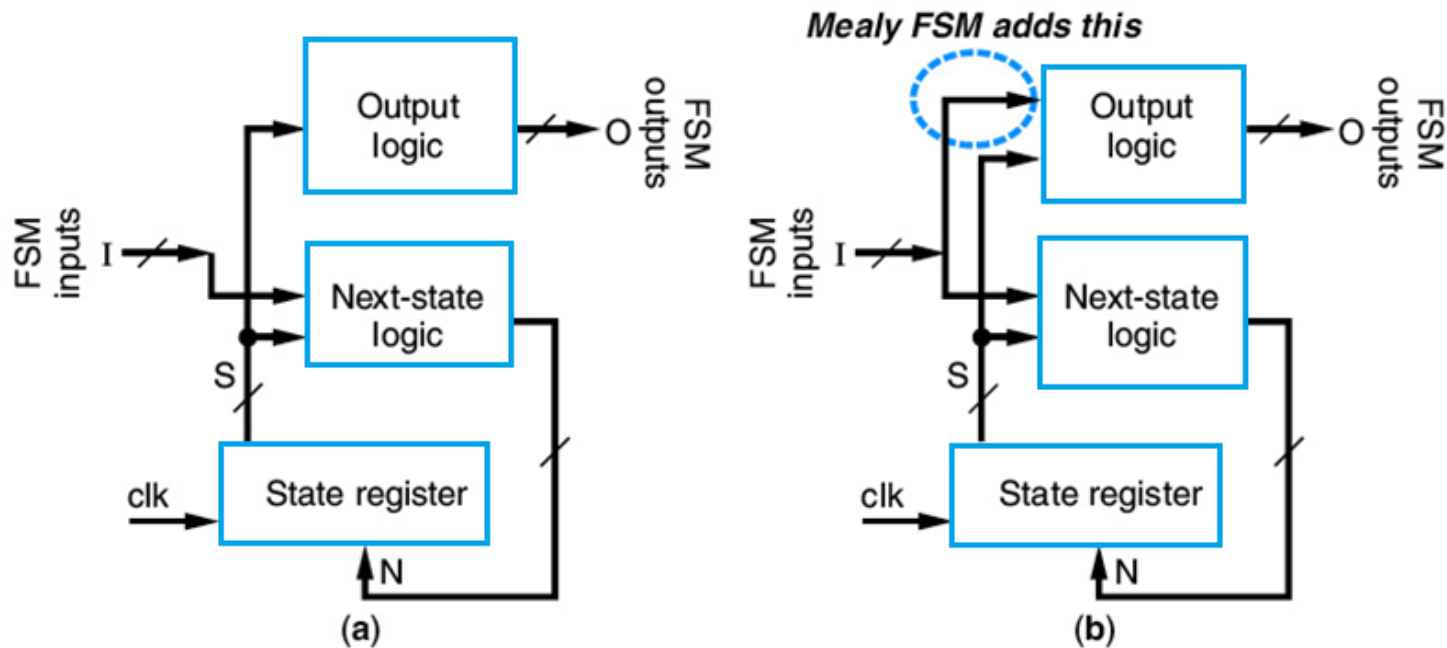


Figure 6.50 Controller architectures for: (a) a Moore FSM, (b) a Mealy FSM.

Exemplo

- ▶ Projete uma Máquina Moore que gere uma saída $Z = 1$ quando o padrão 10110 for detectado na entrada X
- ▶ Entrada serial X de um bit (0 ou 1)
- ▶ Saída Z :
 - $Z = 1$ se $x(t-4..t) = 10110$
 - $Z = 0$ caso contrário

Exemplo

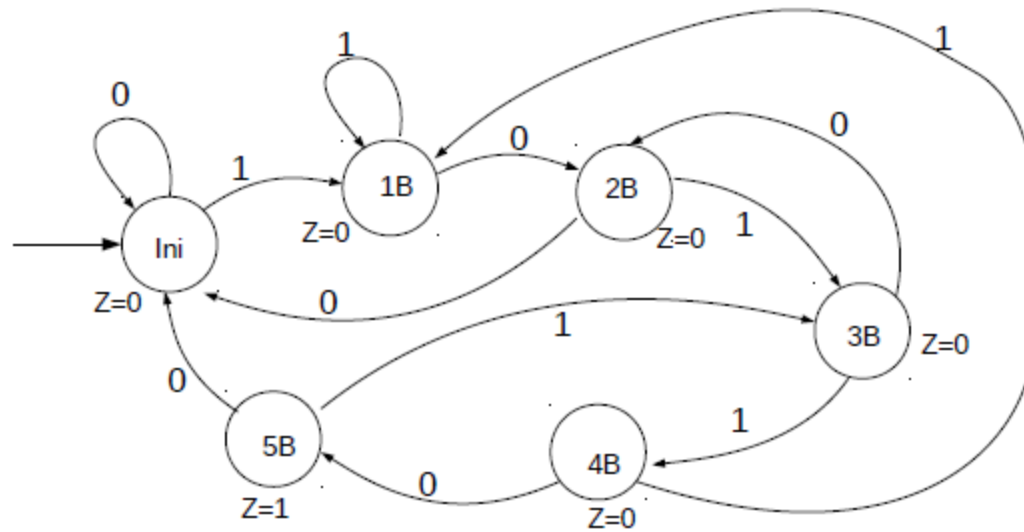
- ▶ Projete uma Máquina Moore que gere uma saída $Z = 1$ quando o padrão 10110 for detectado na entrada X
- ▶ Entrada serial X de um bit (0 ou 1)
- ▶ Saída Z :
 - $Z = 1$ se $x(t-4..t) = 10110$
 - $Z = 0$ caso contrário
- ▶ Quantos estados deve ter o sistema?
- ▶ O que é preciso memorizar?

Exemplo

- ▶ Detecção do padrão 10110
- ▶ Estados:
 - Ini – início e/ou nenhuma parte da sequência foi detectada ($Z = 0$)
 - 1B – bit 1 da sequência foi detectado ($Z = 0$)
 - 2B – bits 10 da sequência foi detectada ($Z = 0$)
 - 3B – bits 101 da sequência foi detectada ($Z = 0$)
 - 4B – bits 1011 da sequência foi detectada ($Z = 0$)
 - 5B – toda a sequência 10110 foi detectada ($Z = 1$)
- ▶ Procedimento:
 - Crie uma bolha no grafo para cada estado e coloque o nome (Ini, 1B, 2B, ...) e a saída Z correspondentes.
 - Construa primeiro o caminho que leva à detecção completa da sequência marcando os ramos do caminho com a condição de X que leva ao próximo estado.
 - Retorne em cada estado analisando quais os próximos estados se a entrada não pertence à sequência, ou seja, não é do valor esperado. Cada nó deve ter um ramo saindo para $X = 0$ e outro para $X = 1$.

Exemplo

- Detecção do padrão 10110



Exemplo

- ▶ Detecção do padrão 10110
- ▶ Fazer o circuito seguindo o procedimento apresentado

Projeto do bloco de controle

▶ Passos:

1. Capture a FSM → Diagrama de bolhas ou de estados
2. Crie a arquitetura → Defina entradas, saídas e o número de flip-flops para representar estados
3. Codifique os estados → um código por estado
4. Crie a tabela de estados
5. Implemente a lógica combinacional

Exemplo

- ▶ Projete uma Máquina Mealy que gere uma saída $Z = 1$ quando o padrão 10110 for detectado na entrada X
- ▶ Entrada serial X de um bit (0 ou 1)
- ▶ Saída Z :
 - $Z = 1$ se $x(t-4..t) = 10110$
 - $Z = 0$ caso contrário

Exemplo

- ▶ Projete uma Máquina Mealy que gere uma saída $Z = 1$ quando o padrão 10110 for detectado na entrada X
- ▶ Entrada serial X de um bit (0 ou 1)
- ▶ Saída Z :
 - $Z = 1$ se $x(t-4..t) = 10110$
 - $Z = 0$ caso contrário
- ▶ Quantos estados deve ter o sistema?
- ▶ O que é preciso memorizar?

Exemplo

▶ Detecção do padrão 10110

▶ Estados:

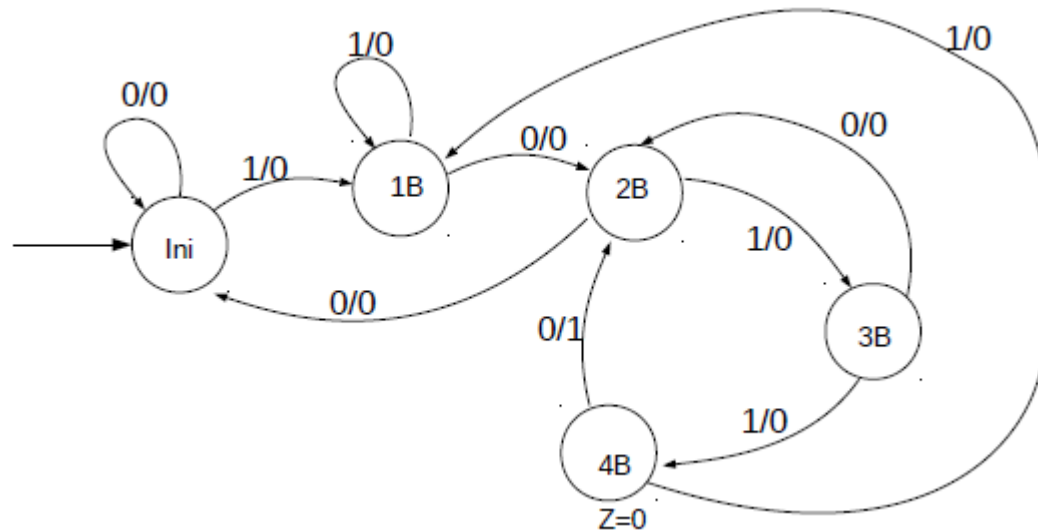
- Ini – início e/ou nenhuma parte da sequência foi detectada ($Z = 0$ para $x=0$ e para $x=1$)
- 1B – primeiro bit 1 da sequência foi detectado ($Z = 0$ para $x=0$ e para $x=1$)
- 2B – bits 10 da sequência foi detectada ($Z = 0$ para $x=0$ e para $x=1$)
- 3B – bits 101 da sequência foi detectada ($Z = 0$ para $x=0$ e para $x=1$)
- 4B – bits 1011 da sequência foi detectada ($Z = 1$ para $x=0$ e para $Z=0$ para $x=1$)

▶ Procedimento:

- Crie uma bolha no grafo para cada estado e coloque o nome (Ini, 1B, 2B...).
- Construa primeiro o caminho que leva à detecção completa da sequência, marcando os ramos do caminho com a condição de X que leva ao próximo estado e a saída correspondente.
- Retorne em cada estado analisando quais os próximos estados se a entrada não pertence à sequência, ou seja, não é do valor esperado, marcando os novos ramos do caminho com a condição de X que leva ao próximo estado e a saída correspondente. Cada nó deve ter um ramo saindo para $X = 0$ e outro para $X = 1$, com as respectivas saídas.

Exemplo

- Detecção do padrão 10110



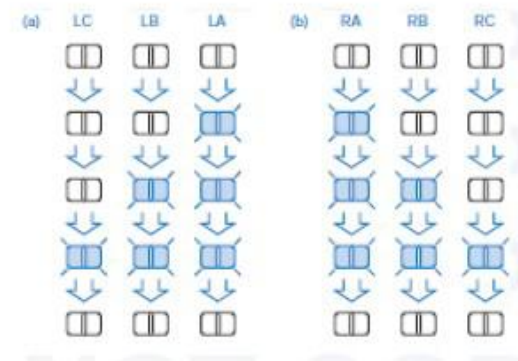
Exemplo

- ▶ Detecção do padrão 10110
- ▶ Fazer o circuito seguindo o procedimento apresentado

Projeto do bloco de controle

▶ Passos:

1. Capture a FSM → Diagrama de bolhas ou de estados
2. Crie a arquitetura → Defina entradas, saídas e o número de flip-flops para representar estados
3. Codifique os estados → um código por estado
4. Crie a tabela de estados
5. Implemente a lógica combinacional



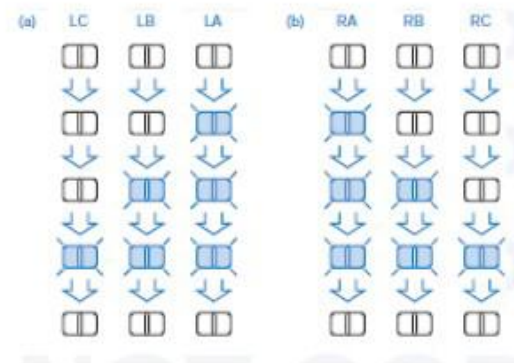
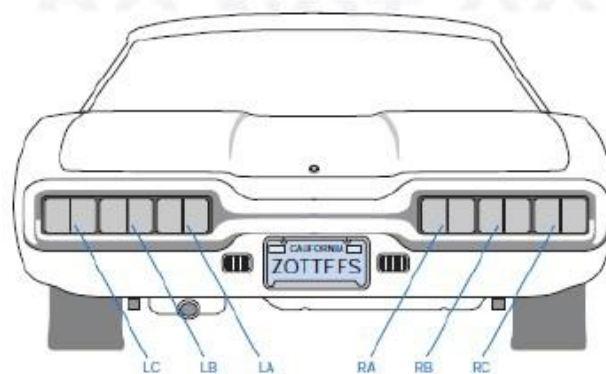
Exemplo

▶ Entradas:

- LEFT – o motorista quer virar à esquerda
- RIGHT – o motorista quer virar à direita
- HAZ – o motorista ligou o pisca alerta (*hazard mode*)

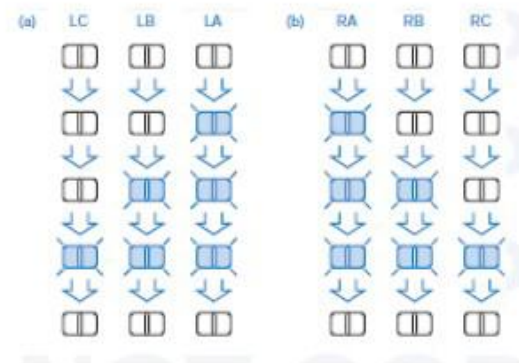
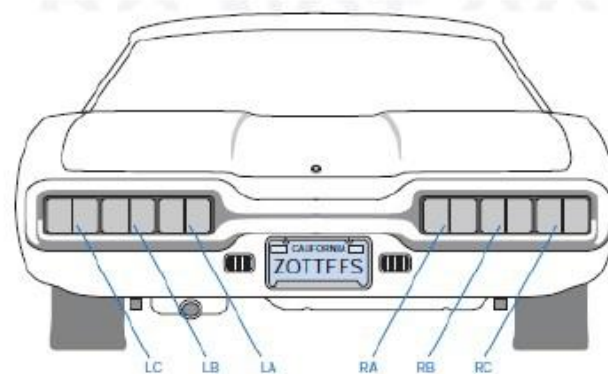
▶ Saídas:

- LC, LB e LA – Lâmpadas da esquerda
- RA, RB e RC – lâmpadas da direita



Exemplo

- ▶ Deve haver um estado para cada situação da tabela:
 - IDLE – 000 000 – Lâmpadas apagadas
 - L1 – 001 000 – 1 lâmpada acesa – esquerda
 - L2 – 011 000 – 2 lâmpadas acesas – esquerda
 - L3 – 111 000 – 3 lâmpadas acesas – esquerda
 - R1 – 000 100 – 1 lâmpada acesa – direita
 - R2 – 000 110 – 2 lâmpadas acesas – direita
 - R3 – 000 111 – 3 lâmpadas acesas – direita
 - LR3 – 111 111 – Todas as lâmpadas acesas



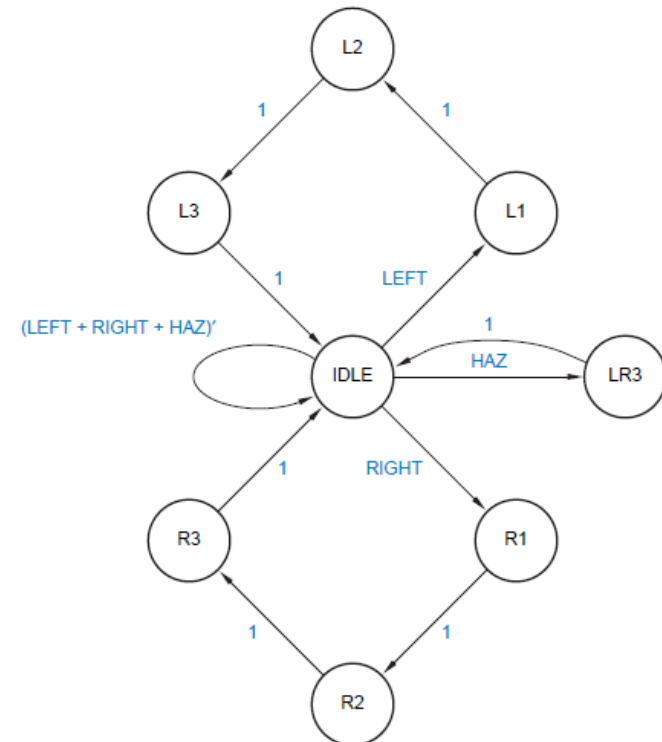
Exemplo

► Primeira versão

- O valor 1 no ramo indica que a transição ocorre independente das entradas

Figure 7-62
Initial state diagram
and output table for
T-bird tail lights.

Output Table						
State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



Exemplo

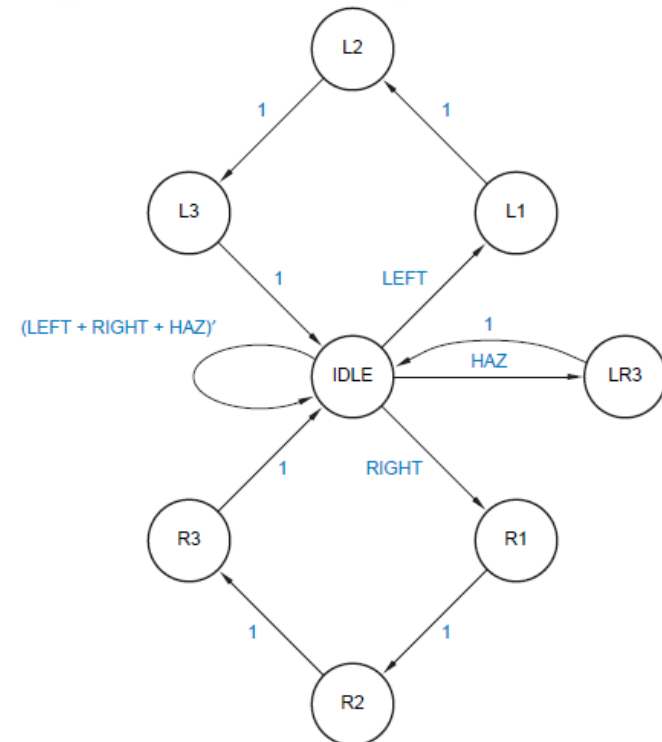
- ▶ Problemas da primeira versão:
 - E se mais de uma entrada for “setada” simultaneamente?

Ex: Pisca alerta e seta para esquerda?
A máquina indica dois estados de destino! Impossível!

- ▶ Resolvendo...

Figure 7-62
Initial state diagram
and output table for
T-bird tail lights.

Output Table						
State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1

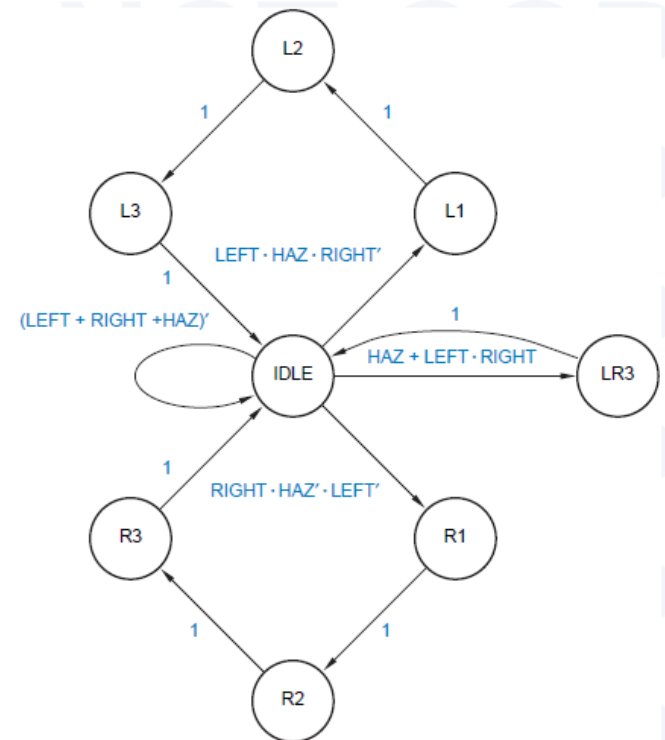


Exemplo

- ▶ Problemas da segunda versão
 - Se alguém apertar o pisca alerta enquanto uma seta estiver ligada, a máquina irá primeiro completar a sequência de lâmpadas para depois entrar no modo de pisca-alerta.

Figure 7-63
Corrected state
diagram for T-bird
tail lights.

Output Table						
State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



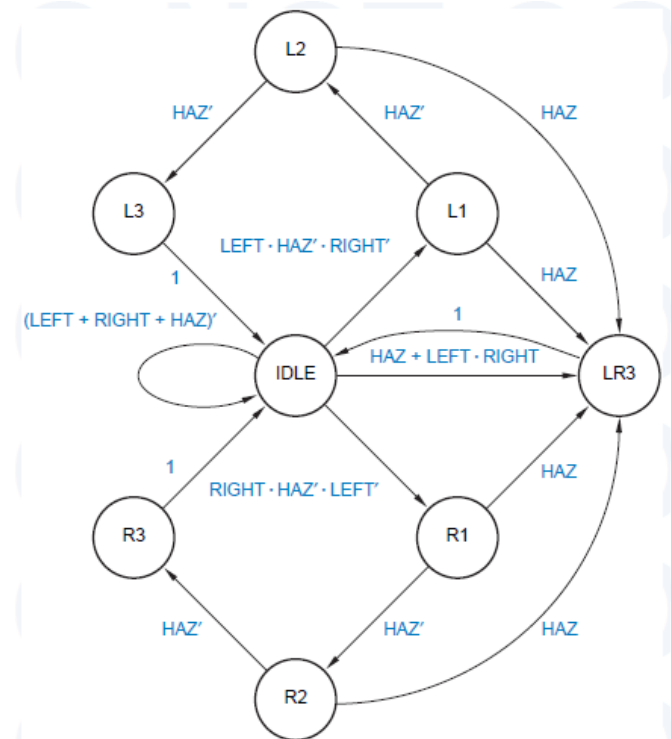
Exemplo

- ▶ Terceira versão, dando prioridade ao pisca alerta

Mais seguro pois permite ligar pisca-alerta mesmo durante a sequência de lâmpadas das setas. Tem-se a opção de ir direto para o pisca-alerta, a partir de qualquer estado da máquina.

Figure 7-64
Enhanced state
diagram for T-bird
tail lights.

Output Table						
State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



Exemplo

► Projeto Thunderbird

Table 7-16
State assignment
for T-bird tail lights
state machine.

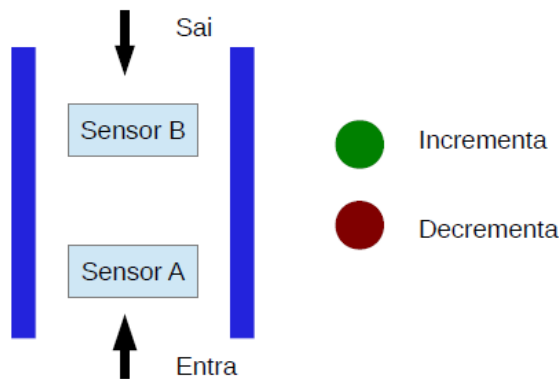
State	Q2	Q1	Q0
IDLE	0	0	0
L1	0	0	1
L2	0	1	1
L3	0	1	0
R1	1	0	1
R2	1	1	1
R3	1	1	0
LR3	1	0	0

S	Q2	Q1	Q0	Transition expression	S*	Q2*	Q1*	Q0*
IDLE	0	0	0	$(\text{LEFT} + \text{RIGHT} + \text{HAZ})'$	IDLE	0	0	0
IDLE	0	0	0	$\text{LEFT} \cdot \text{HAZ}' \cdot \text{RIGHT}'$	L1	0	0	1
IDLE	0	0	0	$\text{HAZ} + \text{LEFT} \cdot \text{RIGHT}$	LR3	1	0	0
IDLE	0	0	0	$\text{RIGHT} \cdot \text{HAZ}' \cdot \text{LEFT}'$	R1	1	0	1
L1	0	0	1	HAZ'	L2	0	1	1
L1	0	0	1	HAZ	LR3	1	0	0
L2	0	1	1	HAZ'	L3	0	1	0
L2	0	1	1	HAZ	LR3	1	0	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	HAZ'	R2	1	1	1
R1	1	0	1	HAZ	LR3	1	0	0
R2	1	1	1	HAZ'	R3	1	1	0
R2	1	1	1	HAZ	LR3	1	0	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0

Table 7-17
Transition list for
T-bird tail lights
state machine.

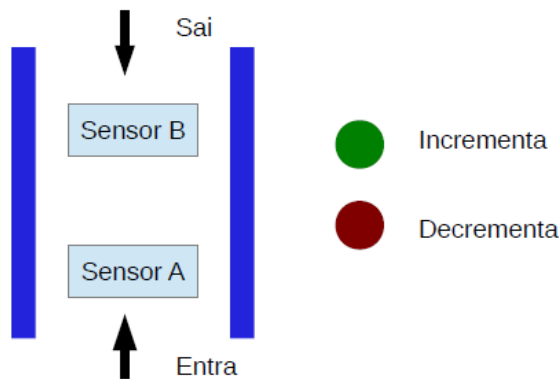
Exemplo

- ▶ Projete um sistema que controle o número de carros em um estacionamento com entrada e saída única. O sistema tem dois sensores A e B posicionados como na figura. Considere que qualquer veículo cobre os dois em algum momento. O sistema deve detectar se um carro está entrando ou saindo e, respectivamente, incrementar ou decrementar um contador. O acumulado não deve ter erros se carro desistir no meio do caminho. Considere que o clock do sistema é muito rápido em relação à velocidade do carro.



Exemplo

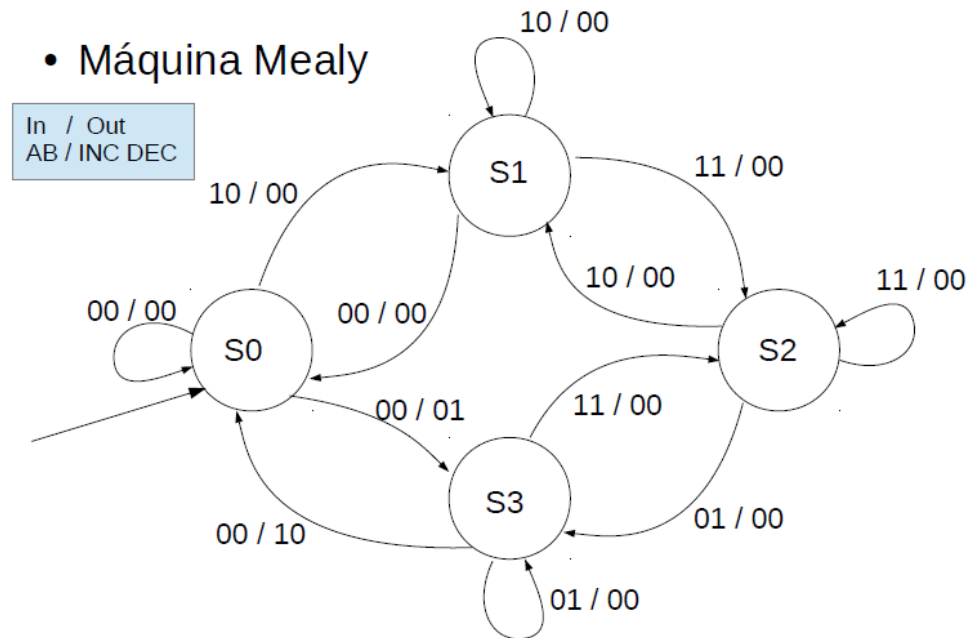
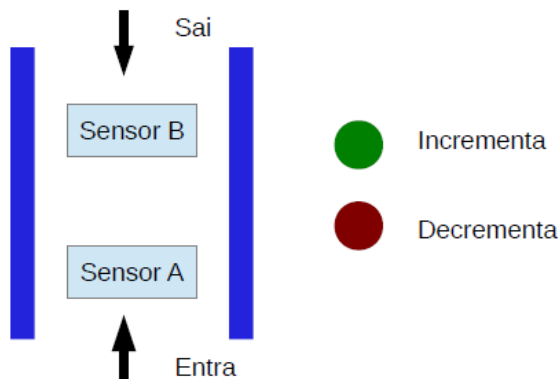
- ▶ Projete um sistema que controle o número de carros em um estacionamento com entrada e saída única. O sistema tem dois sensores A e B posicionados como na figura. Considere que qualquer veículo cobre os dois em algum momento. O sistema deve detectar se um carro está entrando ou saindo e, respectivamente, incrementar ou decrementar um contador. O acumulado não deve ter erros se carro desistir no meio do caminho. Considere que o clock do sistema é muito rápido em relação à velocidade do carro.



- ▶ Estados:
 - S1 – Nenhum sensor ativado
 - S2 – Sensor A ativado
 - S3 – Sensores A e B ativados
 - S4 – Sensor B ativado

Exemplo

- ▶ Máquina Mealy do sistema considerando duas entradas e duas saídas AB/INC DEC



Exemplo

- ▶ Fazer o circuito seguindo o procedimento

Projeto do bloco de controle

▶ Passos:

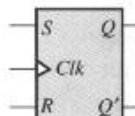
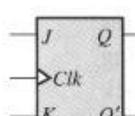
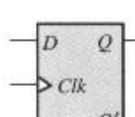
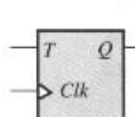
1. Capture a FSM → Diagrama de bolhas ou de estados
2. Crie a arquitetura → Defina entradas, saídas e o número de flip-flops para representar estados
3. Codifique os estados → um código por estado
4. Crie a tabela de estados
5. Implemente a lógica combinacional

Projetos com flip-flops diferentes

► Tipos de flip-flops e tabelas de excitação

TABLE 6.1

Flip-Flop Types

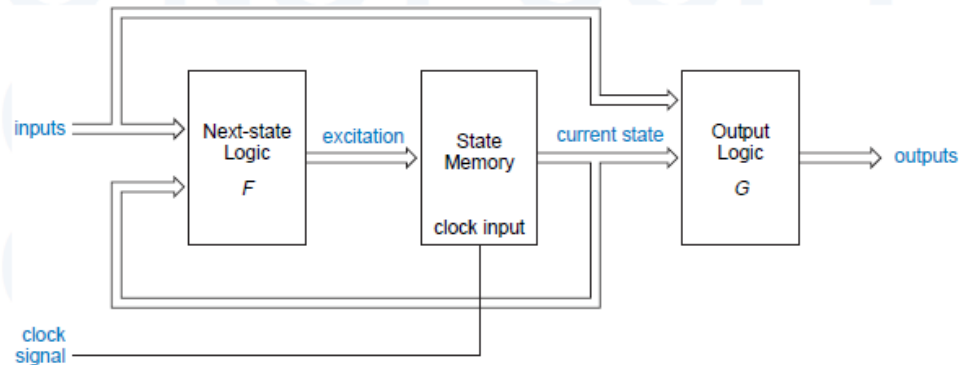
FLIP-FLOP NAME	FLIP-FLOP SYMBOL	CHARACTERISTIC TABLE	CHARACTERISTIC EQUATION	EXCITATION TABLE																																			
SR		<table><tr><th>S</th><th>R</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>NA</td></tr></table>	S	R	Q(next)	0	0	Q	0	1	0	1	0	1	1	1	NA	$Q(next) = S + R'Q$ $SR = 0$	<table><tr><th>Q</th><th>Q(next)</th><th>S</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	Q(next)	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				
JK		<table><tr><th>J</th><th>K</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>Q'</td></tr></table>	J	K	Q(next)	0	0	Q	0	1	0	1	0	1	1	1	Q'	$Q(next) = JQ' + K'Q$	<table><tr><th>Q</th><th>Q(next)</th><th>J</th><th>K</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td></tr><tr><td>1</td><td>0</td><td>X</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q(next)	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				
D		<table><tr><th>D</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q(next)	0	0	1	1	$Q(next) = D$	<table><tr><th>Q</th><th>Q(next)</th><th>D</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Q	Q(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q(next)																																						
0	0																																						
1	1																																						
Q	Q(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table><tr><th>T</th><th>Q(next)</th></tr><tr><td>0</td><td>Q</td></tr><tr><td>1</td><td>Q'</td></tr></table>	T	Q(next)	0	Q	1	Q'	$Q(next) = TQ' + T'Q$	<table><tr><th>Q</th><th>Q(next)</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Q	Q(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q(next)																																						
0	Q																																						
1	Q'																																						
Q	Q(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

Projetos com flip-flops diferentes

- ▶ Arquiteturas de Máquinas Mealy e Moore para outros flip-flops

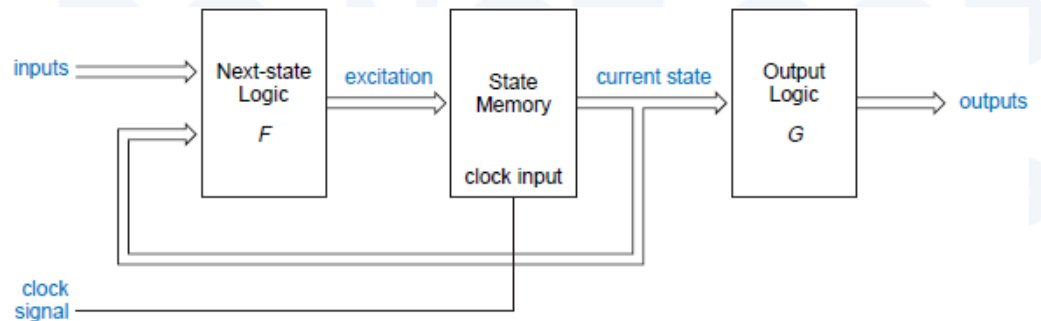
Neste caso, a memória não é um registrador de N bits, mas tem n flip-flops do tipo escolhido

Figure 7-35 Clocked synchronous state-machine structure (Mealy machine).



Copyright © 1999 by John F. Wakerly

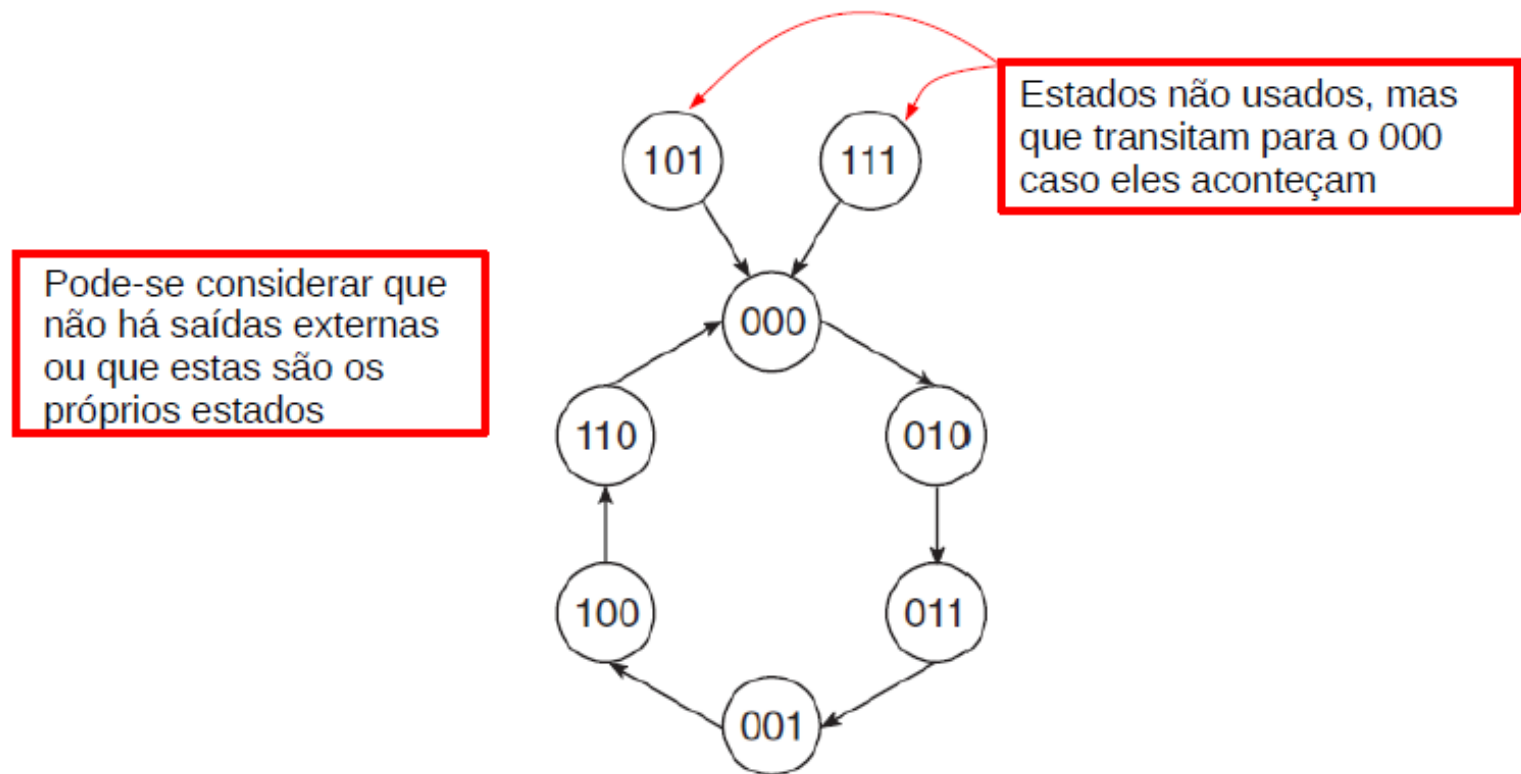
Figure 7-36 Clocked synchronous state-machine structure (Moore machine).



Copyright © 1999 by John F. Wakerly

Exemplo com flip-flop JK

- ▶ Projete o circuito da figura usando flip-flops JK



Exemplo com flip-flop JK

- Tabela de excitação do flip-flop e tabela de transição de estados

Tabela de excitação do JK

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Present state			Next state			Inputs					
<i>C</i>	<i>B</i>	<i>A</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>J_C</i>	<i>K_C</i>	<i>J_B</i>	<i>K_B</i>	<i>J_A</i>	<i>K_A</i>
0	0	0	0	1	0	0	X	1	X	0	X
0	0	1	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	0	0	0	X	1	0	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	X	1	X	1	X	1

Exemplo com flip-flop JK

- Minimização das funções de entradas dos flip-flops

J_A

	\bar{A}	A
$\bar{B}\bar{C}$		X
$\bar{B}C$		X
BC		X
$B\bar{C}$	1	X

K_A

	\bar{A}	A
$\bar{B}\bar{C}$	X	1
$\bar{B}C$	X	1
BC	X	1
$B\bar{C}$	X	

J_C

	\bar{A}	A
$\bar{B}\bar{C}$		1
$\bar{B}C$	X	X
BC	X	X
$B\bar{C}$		

K_C

	\bar{A}	A
$\bar{B}\bar{C}$	X	X
$\bar{B}C$		1
BC	1	1
$B\bar{C}$	X	X

J_B

	\bar{A}	A
$\bar{B}\bar{C}$	1	
$\bar{B}C$	1	
BC	X	X
$B\bar{C}$	X	X

K_B

	\bar{A}	A
$\bar{B}\bar{C}$	X	X
$\bar{B}C$	X	X
BC	1	1
$B\bar{C}$		1

$$J_A = B.\bar{C}$$

$$K_A = \bar{B} + C$$

$$J_B = \bar{A}$$

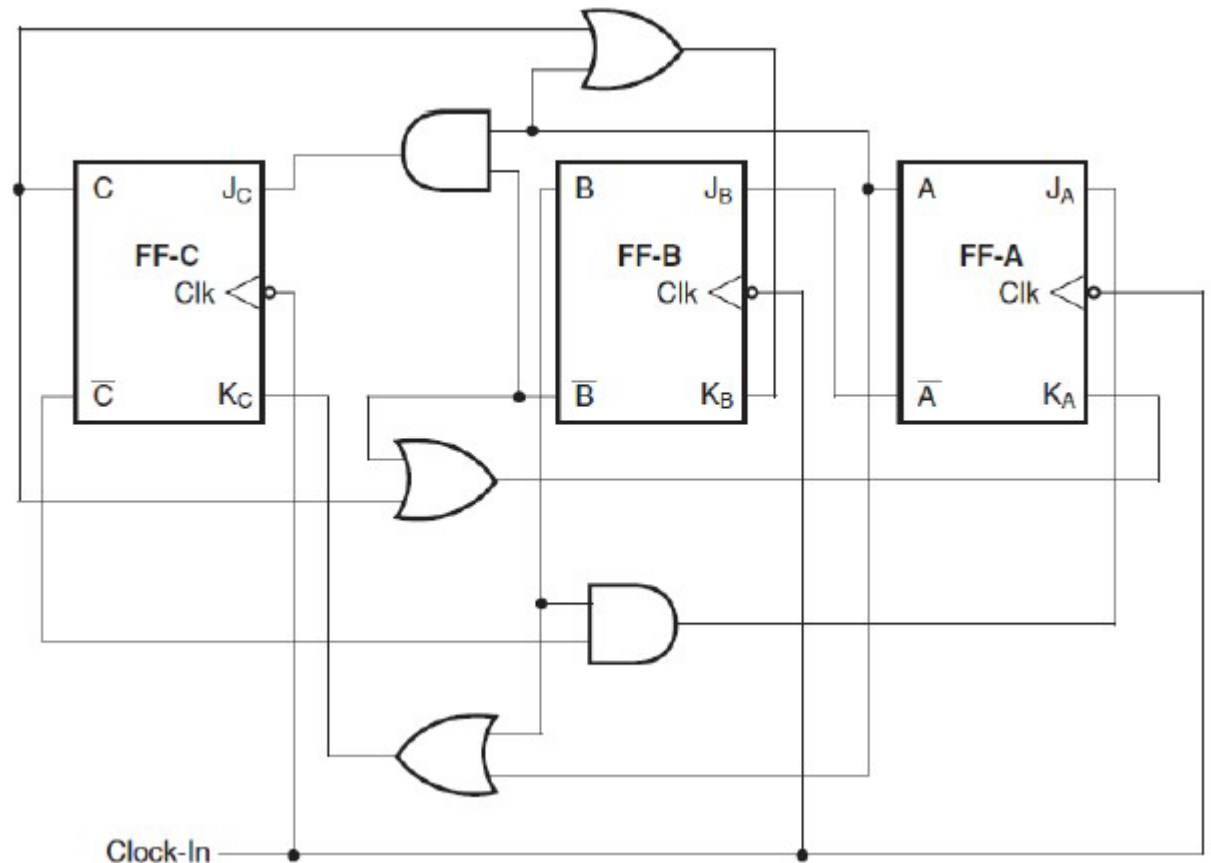
$$K_B = A + C$$

$$J_C = A.\bar{B}$$

$$K_C = A + B$$

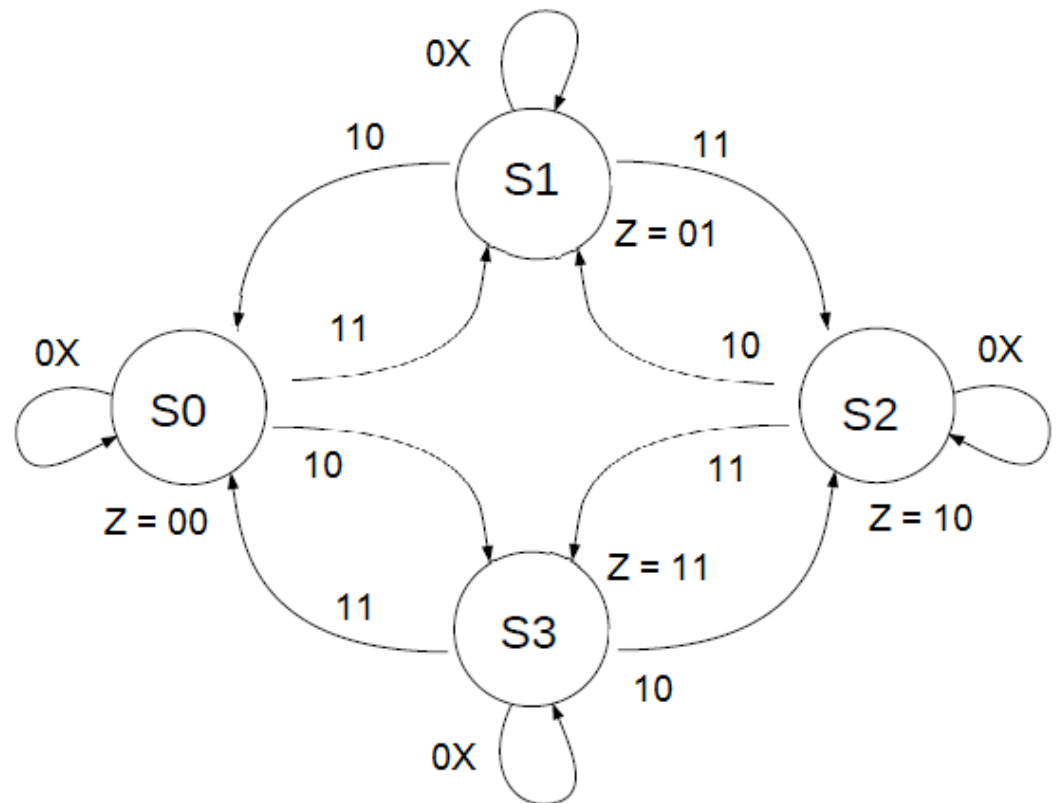
Exemplo com Flip-flop JK

► Circuito final



Exemplo com flip-flop T

- ▶ Projete de um contador binário de 2 bits com habilita de contagem e bit de direção, usando flip-flop tipo T



Para ser continuado...