



# CIRCUITOS DIGITAIS MÁQUINA DE ESTADOS

Marco A. Zanata Alves

# CIRCUITOS COMBINACIONAIS VS. CIRCUITOS SEQUENCIAIS

**Circuitos Combinacionais:** os valores das saídas  $S_i$  no instante  $t$  depende apenas da combinação dos valores das entradas  $e_i$  neste mesmo instante. Os estados anteriores não interessam.

**Circuitos Sequenciais:** os valores das saídas  $S_i$  no instante  $t$  não depende apenas dos valores das entradas  $e_i$  neste instante, mas também da **sequência das entradas anteriores**.

# CIRCUITOS COMBINACIONAIS VS. SEQUENCIAIS

Nem todos os projetos em sistemas digitais conseguem ser resolvidos utilizando circuitos combinacionais.

Algumas vezes é necessário o conhecimento de um ou mais estados anteriores e também da sequência anterior para se calcular a saída do circuito.

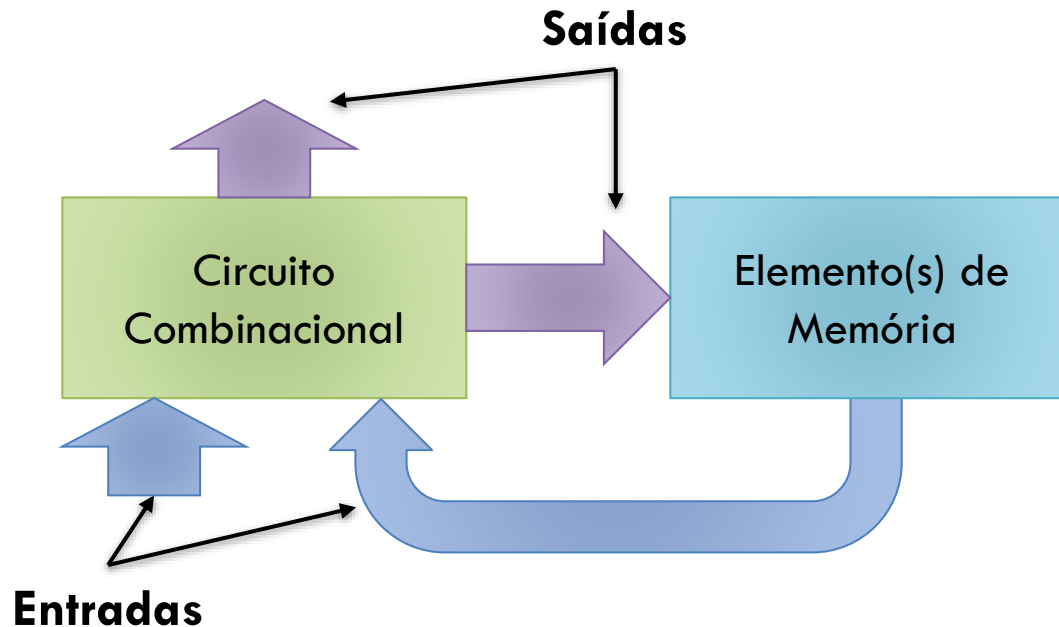
**Exemplo:** Contadores

# CIRCUITOS SEQUENCIAIS

Circuito Sequencial = Circuito Combinacional + Elemento de Memória

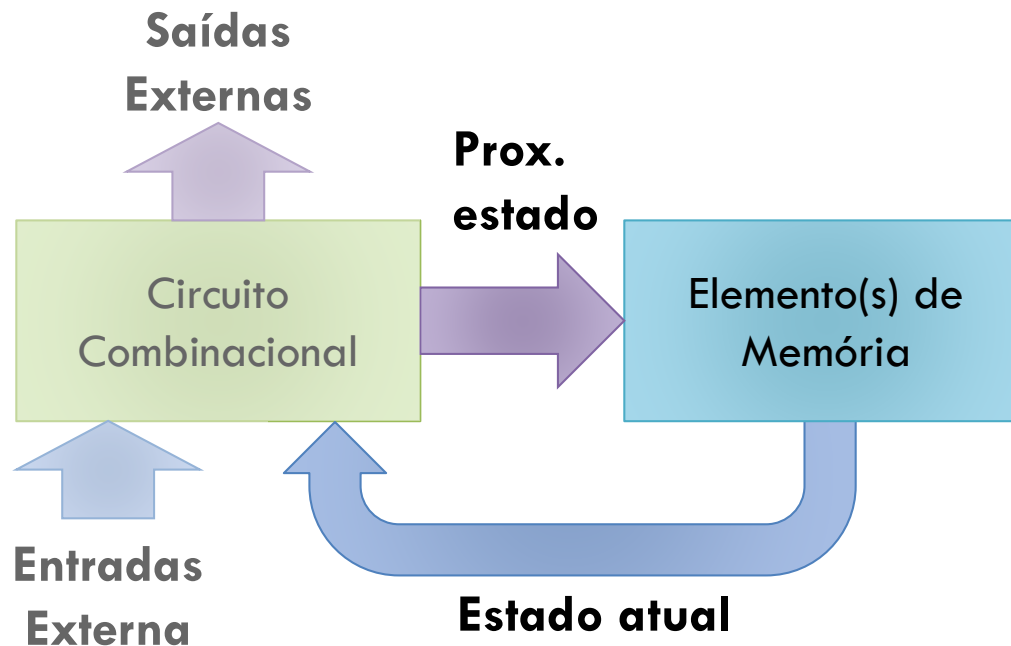
Existe uma realimentação (através da memória);

Dependem da “história” das entradas passadas.



# COMPOSIÇÃO DO CIRCUITO SEQUENCIAL

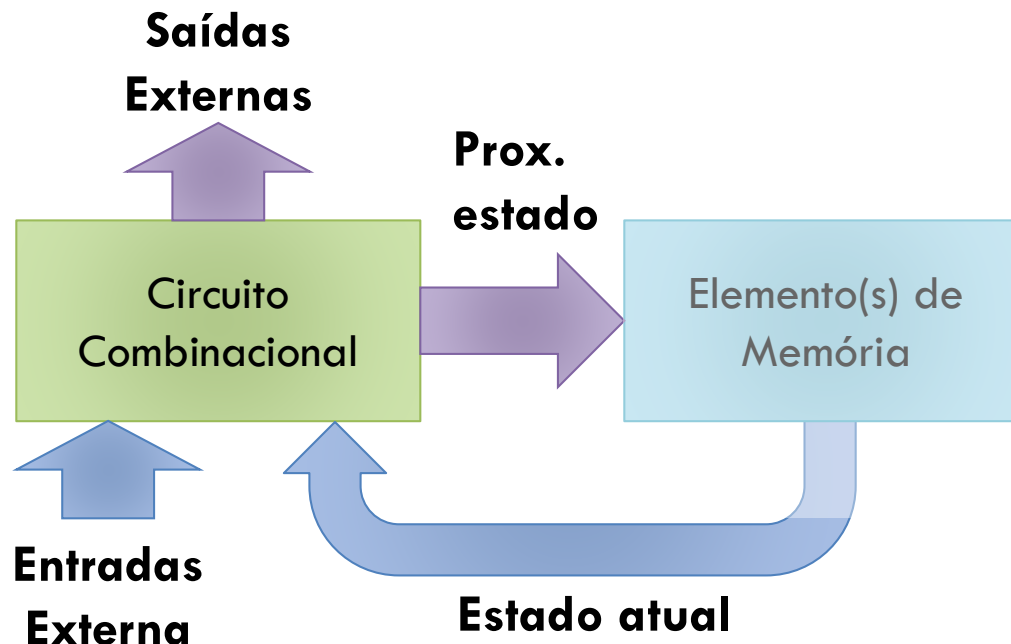
**Bloco de memória** → armazena informações anteriores para definir o **estado atual**. Tem como entrada o **próximo estado**.



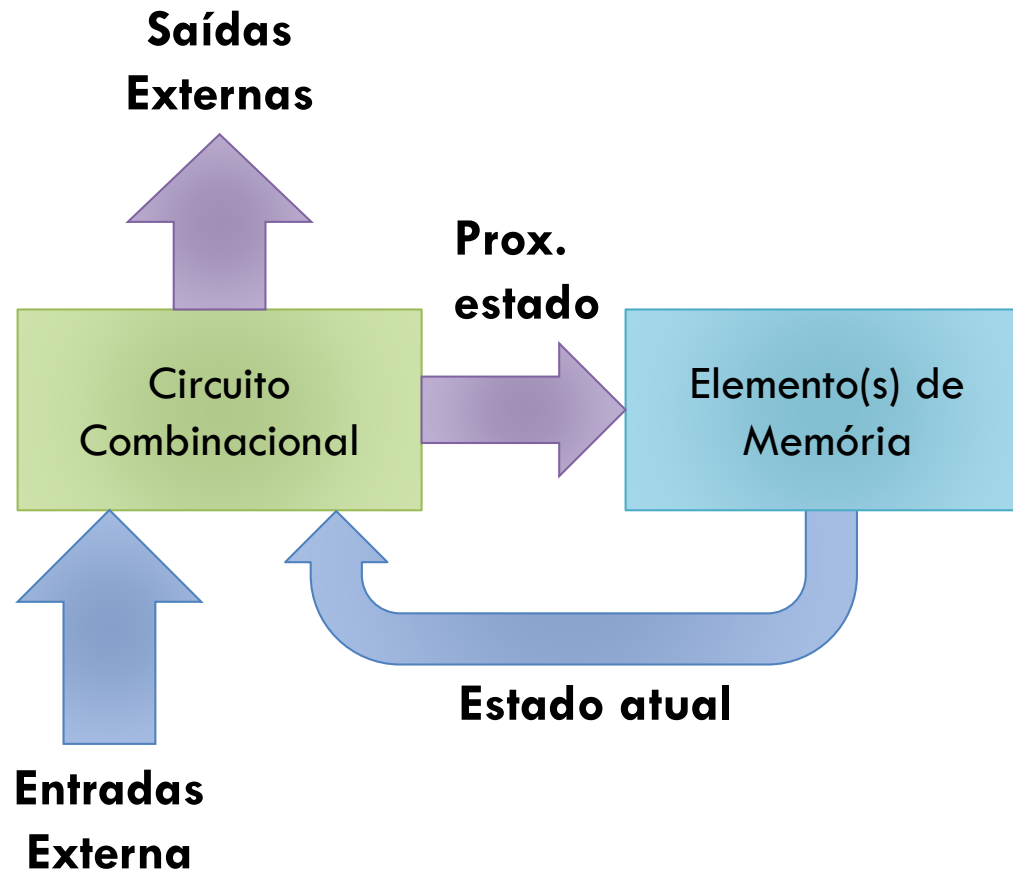
# COMPOSIÇÃO DO CIRCUITO SEQUENCIAL

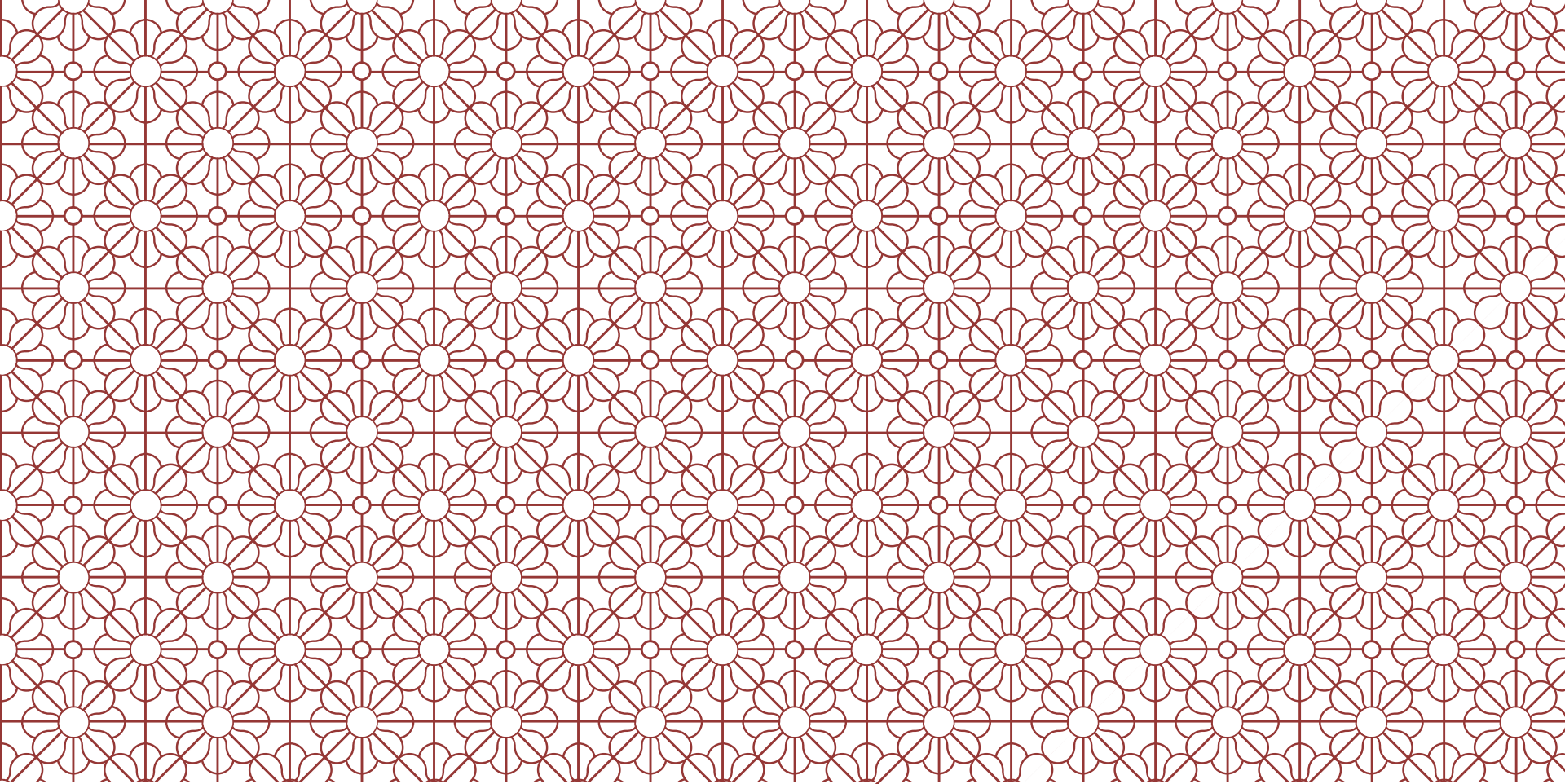
**Bloco de memória** → armazena informações anteriores para definir o **estado atual**. Tem como entrada o **próximo estado**.

**Bloco combinacional** → define o **próximo estado** e a **saída externa**. Tem como entradas o **estado atual** e as **entradas externas**.



# COMPOSIÇÃO DO CIRCUITO SEQUENCIAL





# MÁQUINA DE ESTADOS FINITOS



# MÁQUINA DE ESTADOS FINITOS

Conhecido como autômato finito ou Finite State Machine (FSM).

Trata-se de uma máquina abstrata que deve estar em um de seus finitos estados e cada momento.

A máquina está em apenas um estado por vez, este estado é chamado de estado atual.

Indiretamente, um estado armazena informações sobre o passado.

Uma transição indica uma mudança de estado e é descrita por uma condição que precisa ocorrer para que a transição ocorra.

Uma ação ou saída, é a descrição de uma atividade que deve ser realizada num determinado momento.

# ESTADOS

Cada estágio através do qual o circuito sequencial avança;

Em cada estado, o circuito armazena uma “recordação” de sua história passada, para saber o que fazer a seguir (instante  $t + 1$ );

Nem toda informação anterior é relevante. Ou seja, nem todo estado precisa ser armazenado.

O próximo estado pode depender:

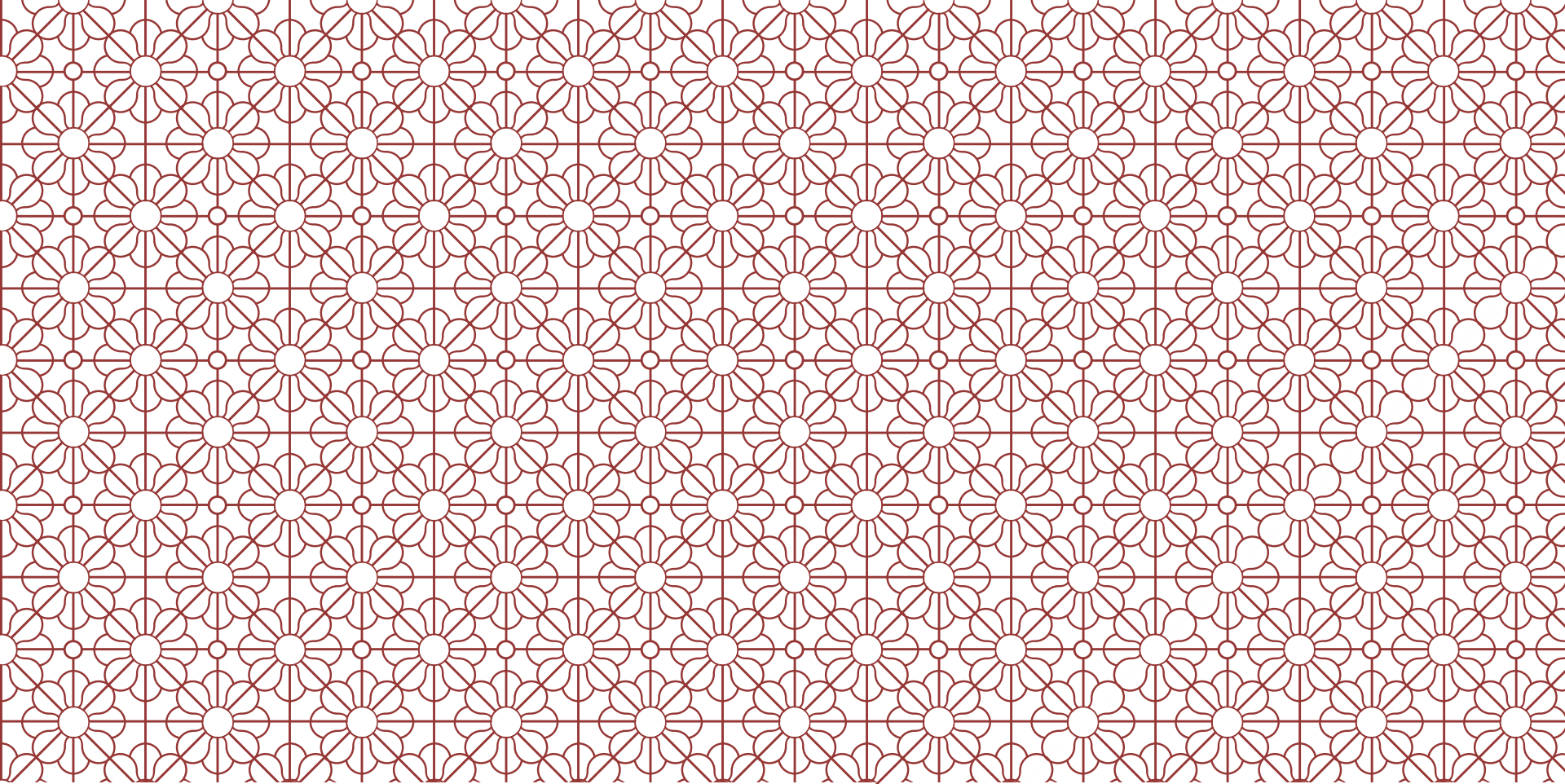
- Apenas do estado anterior;
- Apenas das entradas atuais;
- Uma combinação das entradas atuais com o estado anterior.

# SAÍDAS

As saídas do circuito podem ser geradas de duas formas:

- Apenas o estado atual é utilizado para a geração das saídas;
- O estado atual mais as entradas atuais são utilizadas na geração das saídas;

Dependendo de qual maneira as saídas são geradas damos o nome de máquina de estados de Moore ou Mealy.

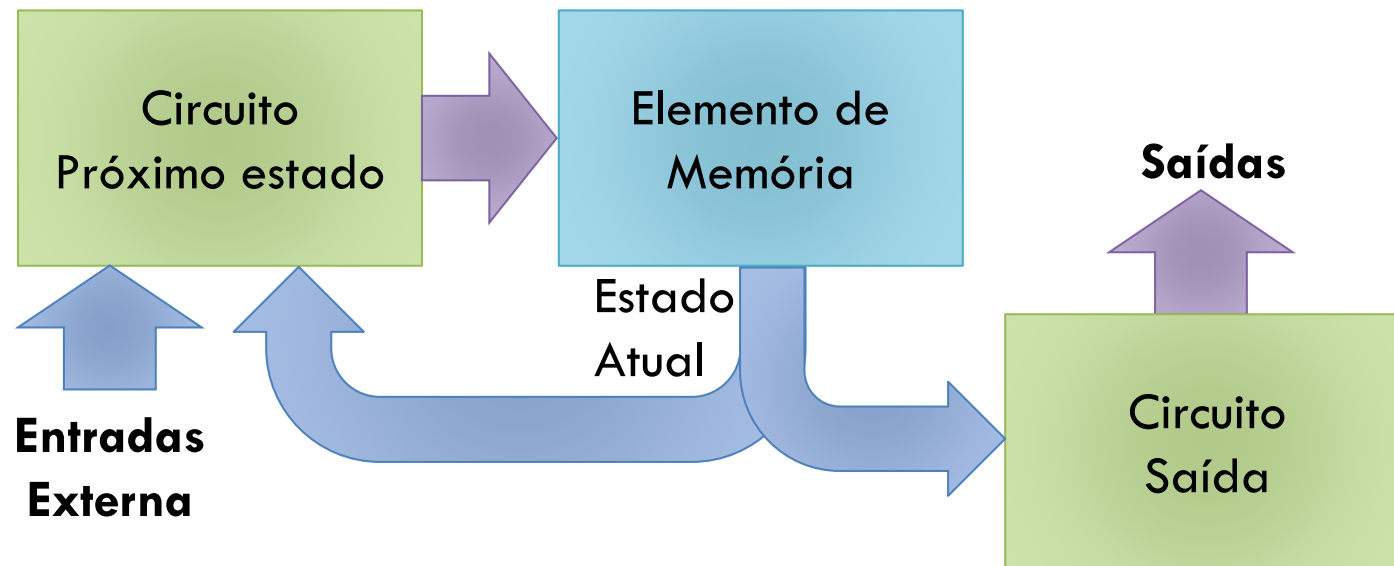


# TIPOS DE MÁQUINAS DE ESTADOS

# MÁQUINA DE MOORE

As entradas não interferem diretamente na saída, somente nos estados futuros;

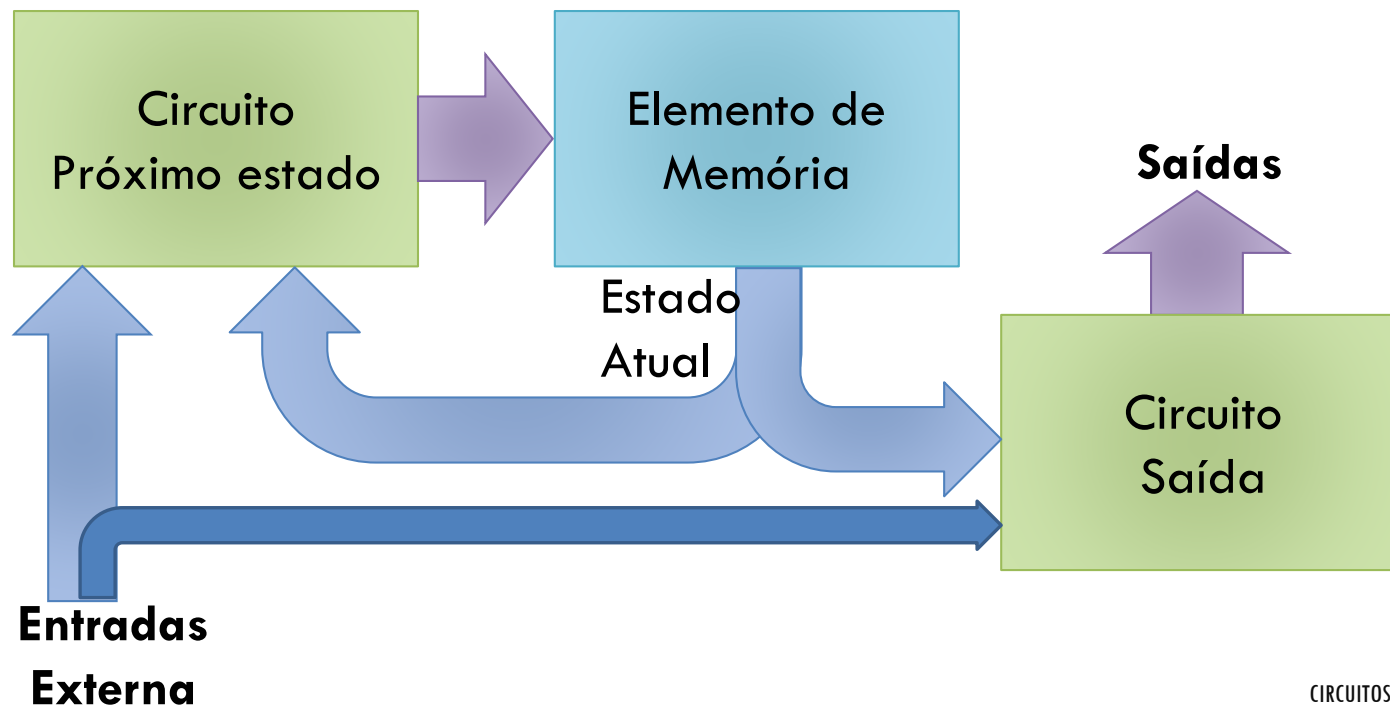
**As saídas dependem apenas do estado atual**



# MÁQUINA DE MEALY

As entradas interferem nos estados futuros e também na saída;

As saídas dependem da **entrada** e do estado atual



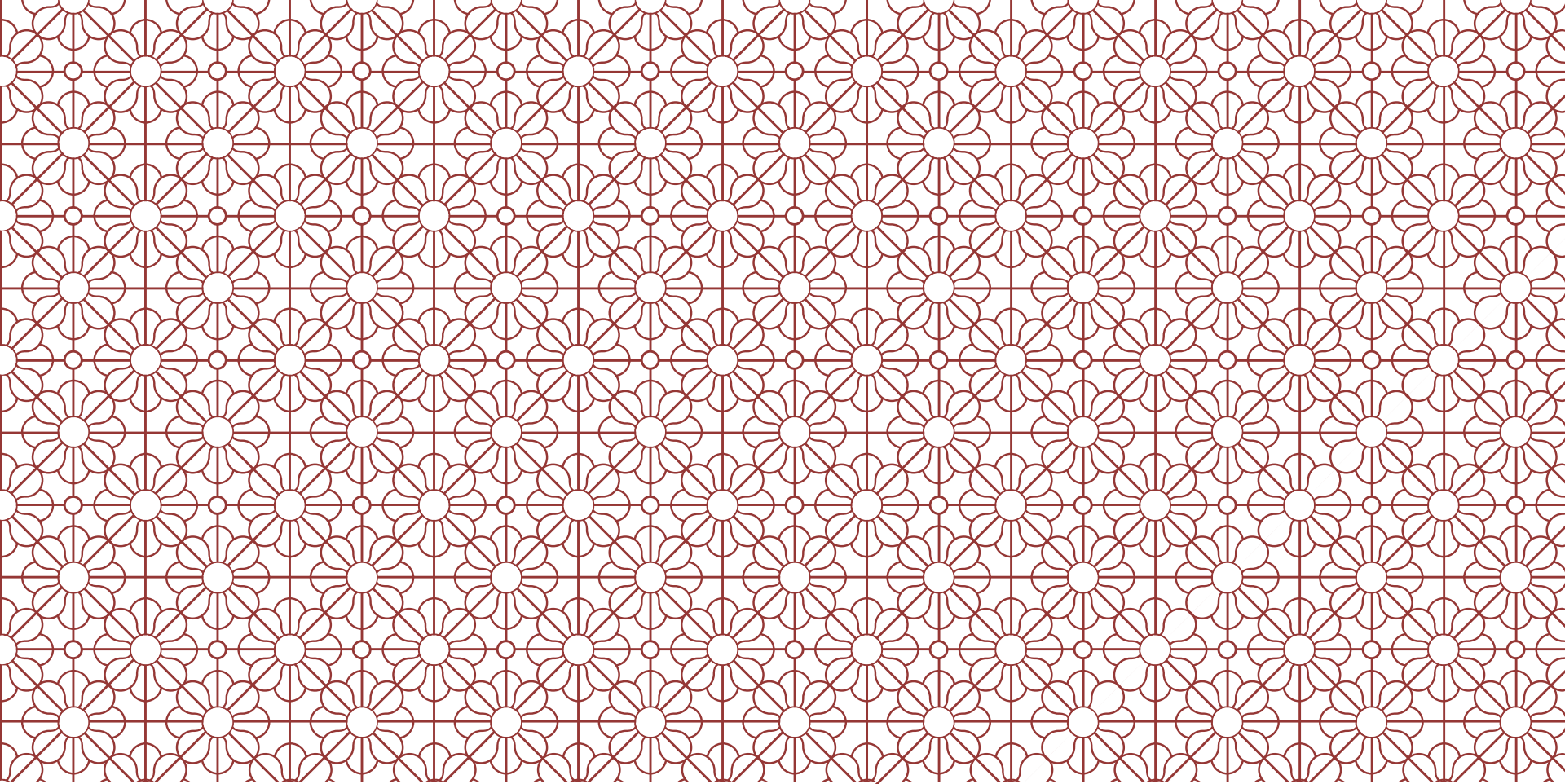
# MOORE E MEALY

## **Máquinas de Moore:**

- As saídas são função apenas do estado presente (não das entradas);
- As entradas só interferem no próximo estado;
- As saídas variam sincronamente;
- Resposta mais lenta ou inexistente à variações na entrada.

## **Máquinas de Mealy:**

- As saídas são função do estado presente e das entradas atuais;
- As entradas interferem no próximo estado e também na saída;
- As saídas variam assincronamente com as entradas;
- Resposta mais rápida à variações na entrada.



# DIAGRAMAS DE ESTADOS



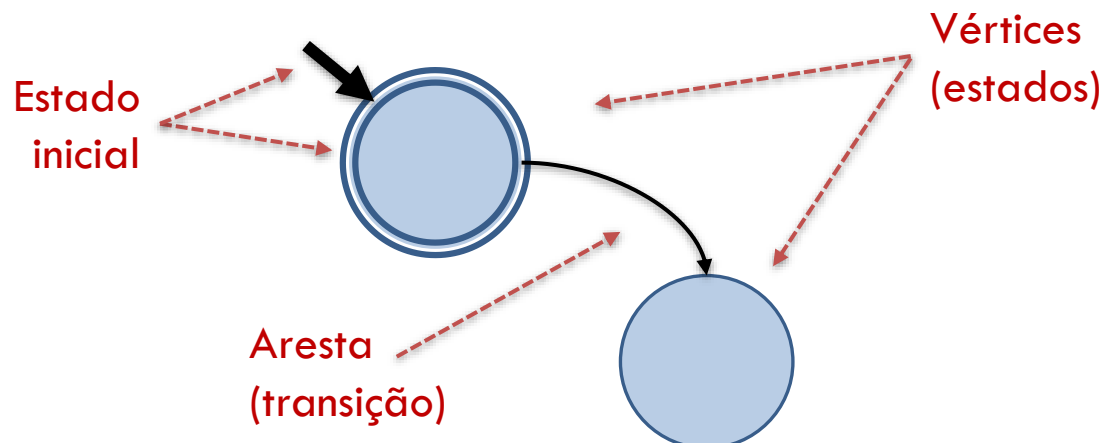
# DIAGRAMA DE ESTADOS

O Diagrama de Estado ou Diagrama de Fluxo de Estado é um **grafo**

- Cada nó (vértice) representa um estado
- Cada arco (aresta) representa uma transição de estados (fluxo);

A cada pulso de clock, o fluxo avança um estado;

Estado com borda dupla indica estado inicial, o estado inicial (estado de entrada) também pode ser indicado por uma flecha;



# DIAGRAMA DE ESTADOS

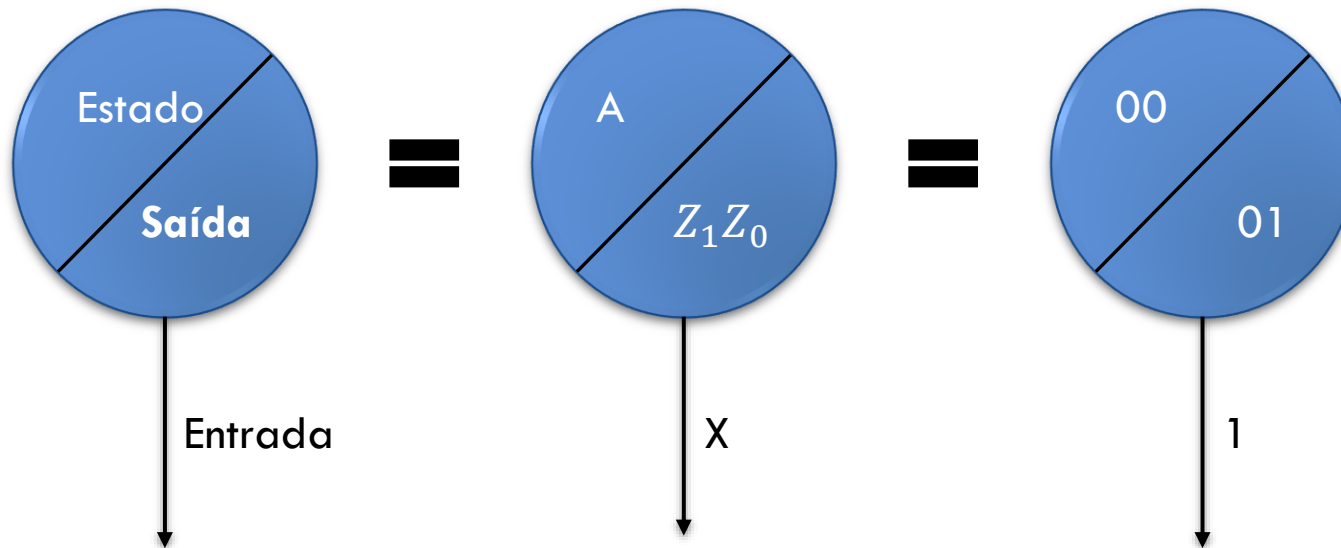
O diagrama de estados tem formatos diferentes para cada um dos modelos:

- Máquina de Moore
- Máquina de Mealy

# MÁQUINA DE MOORE

A saída depende exclusivamente do estado;

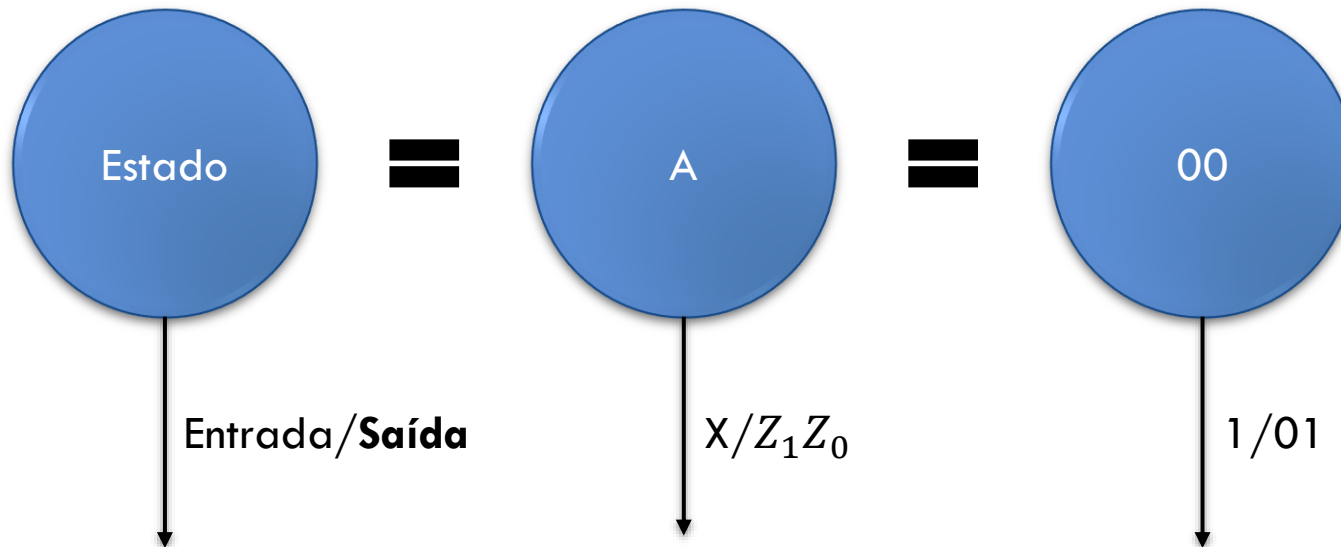
A entrada só interfere no próximo estado.



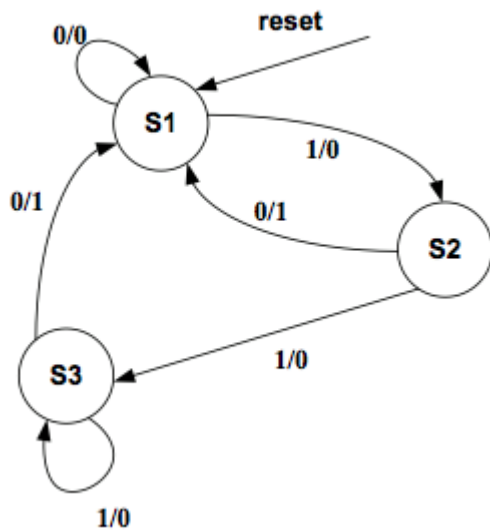
# MÁQUINA DE MEALY

A saída depende do estado presente e da entrada;

A entrada interfere no próximo estado e na saída.



# QUAL MÁQUINA É ESSA?

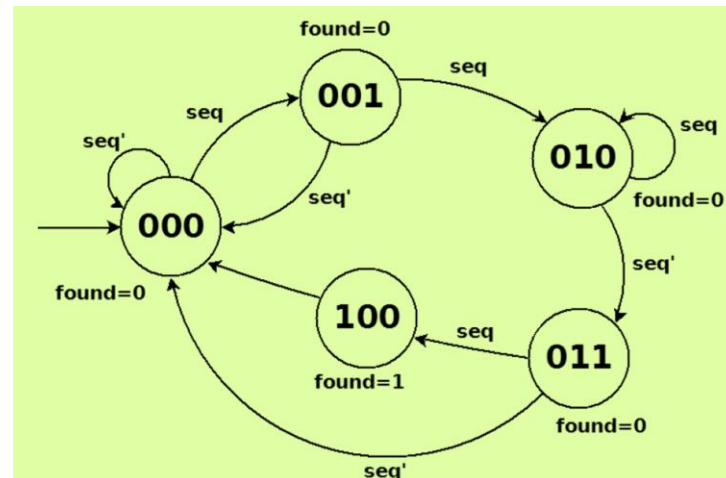
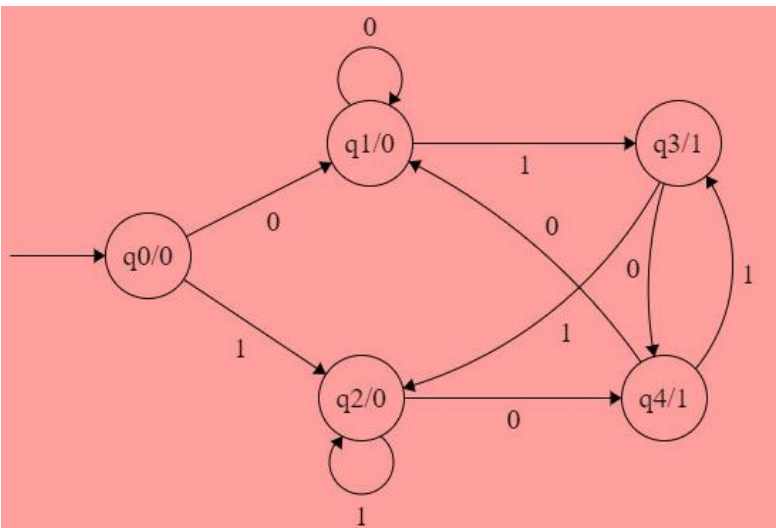
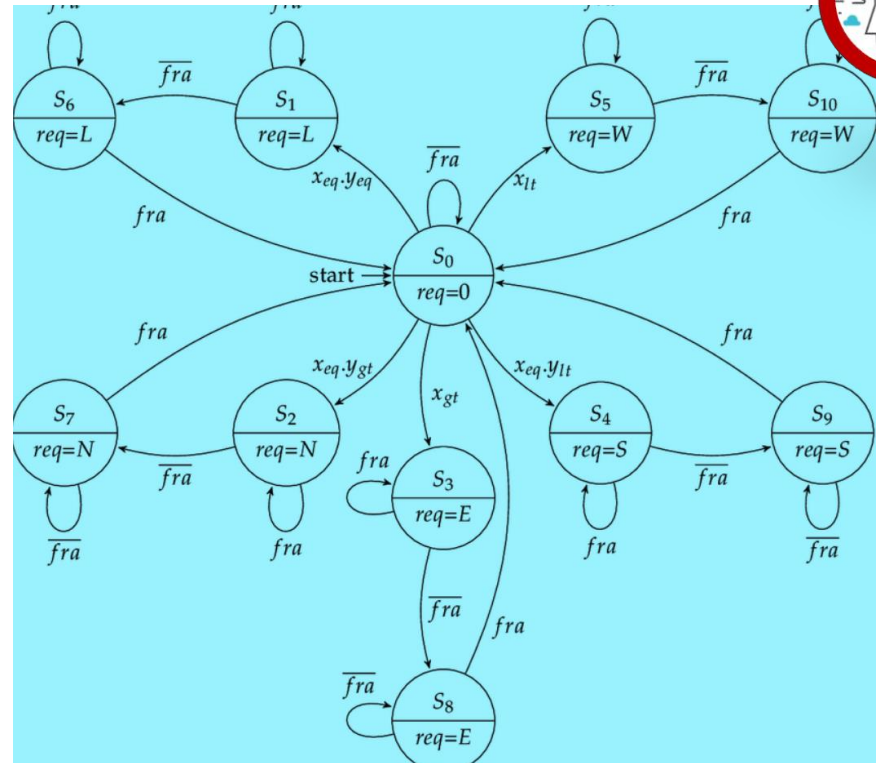
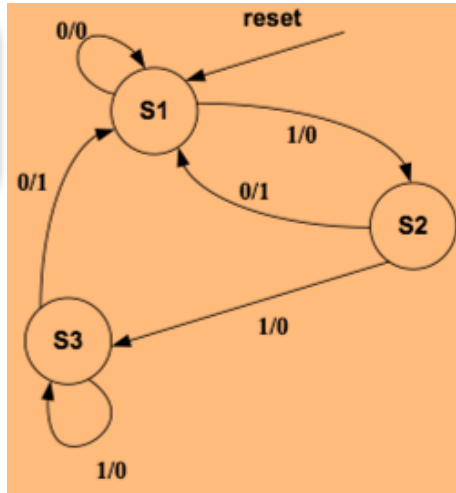


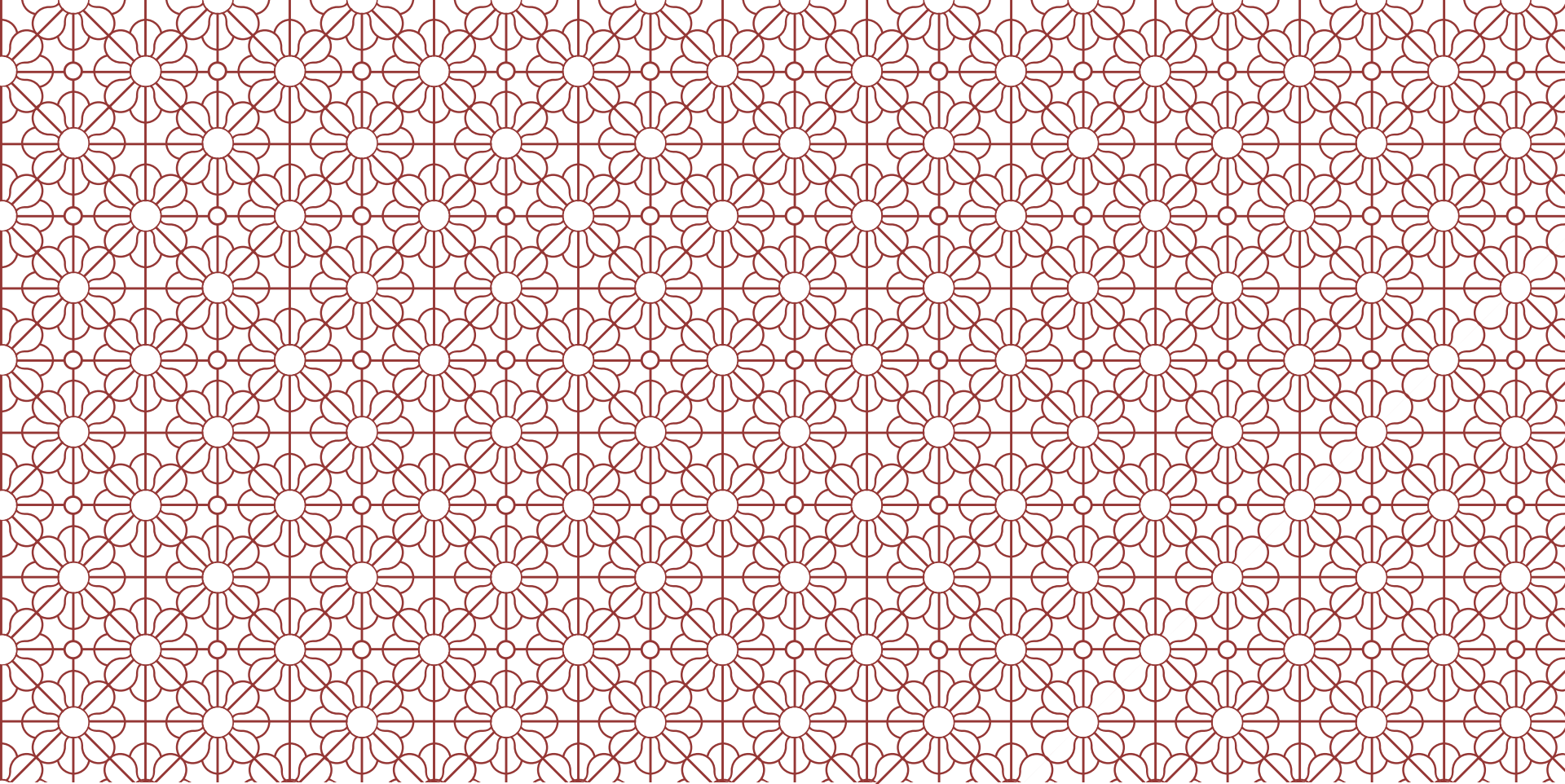


# EXEMPLO DE MÁQUINAS

Quais os tipos?

Qual o estado inicial?





# EXEMPLO

## CONTADOR CRESCENTE 0~3

# EXEMPLO 1: CONTADOR CRESCENTE 0~3

Pensar no diagrama de estados.

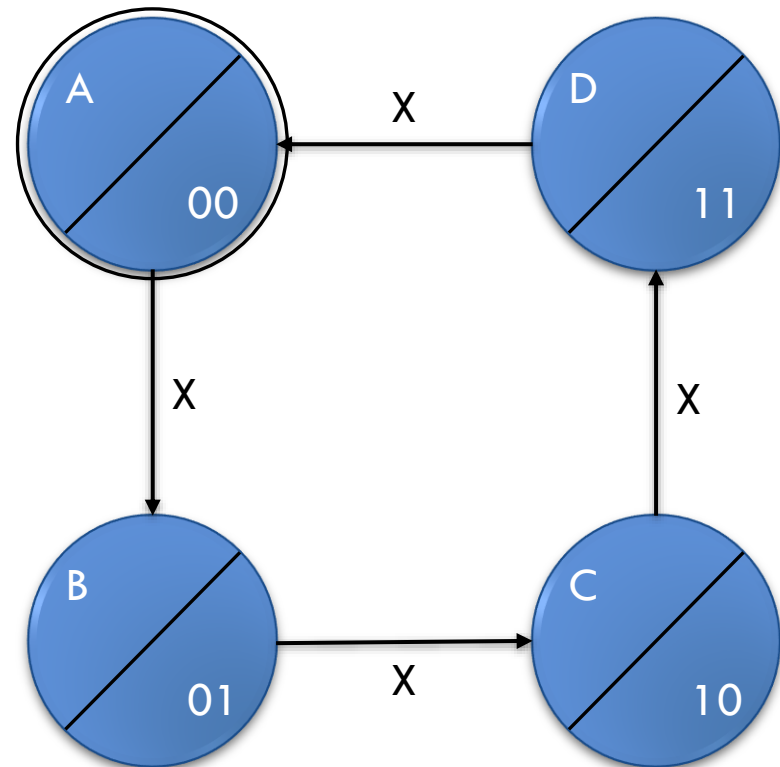


# EXEMPLO 1: CONTADOR CRESCENTE 0~3

O que acontece quando atingimos o contador máximo?

Como podemos fazer uma tabela de transições?

Diagrama de Estados



# EXEMPLO 1:

## CONTADOR CRESCENTE 0~3

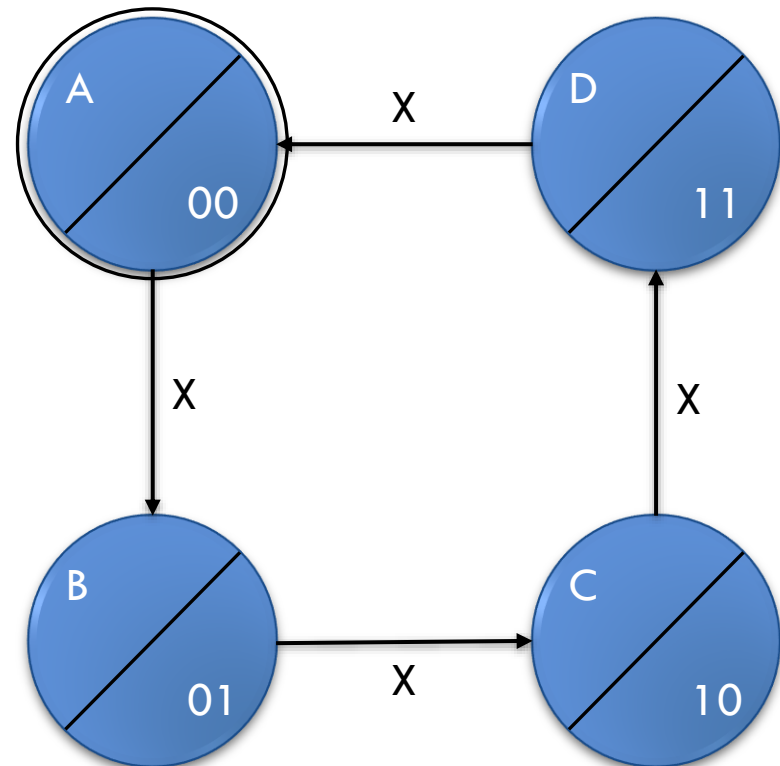
Tabela de Transição de Estados

Estado atual	Saída $Z_1Z_0$	Próximo estado
A	00	B
B	01	C
C	10	D
D	11	A

### Máquina de Moore

Saída depende apenas do estado

Diagrama de Estados



# EXEMPLO 1:

## CONTADOR CRESCENTE 0~3

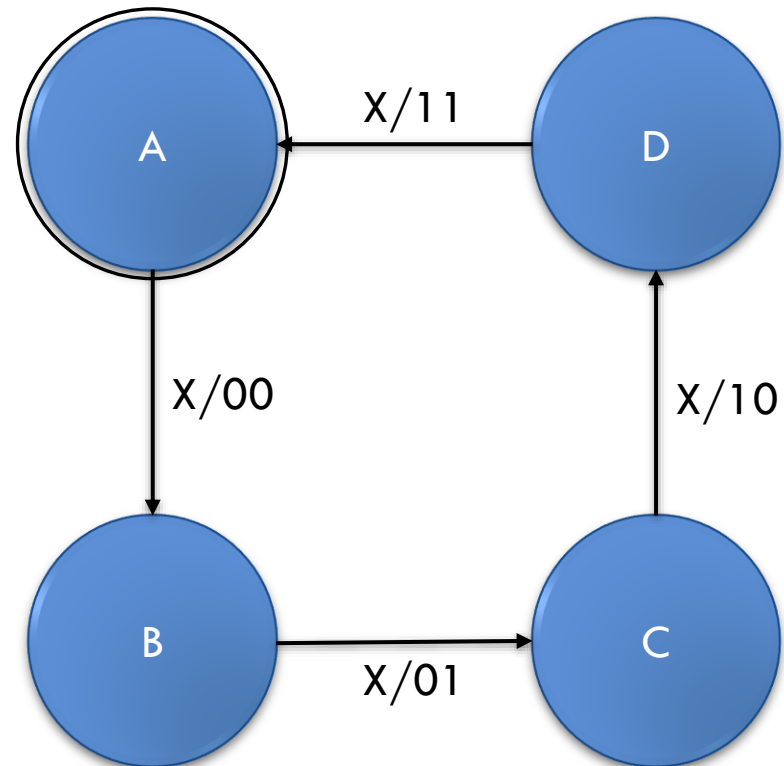
Tabela de Transição de Estados

Estado atual	Saída $Z_1Z_0$	Próximo estado
A	00	B
B	01	C
C	10	D
D	11	A

### Máquina de Mealy

Saída depende da transição

Diagrama de Estados

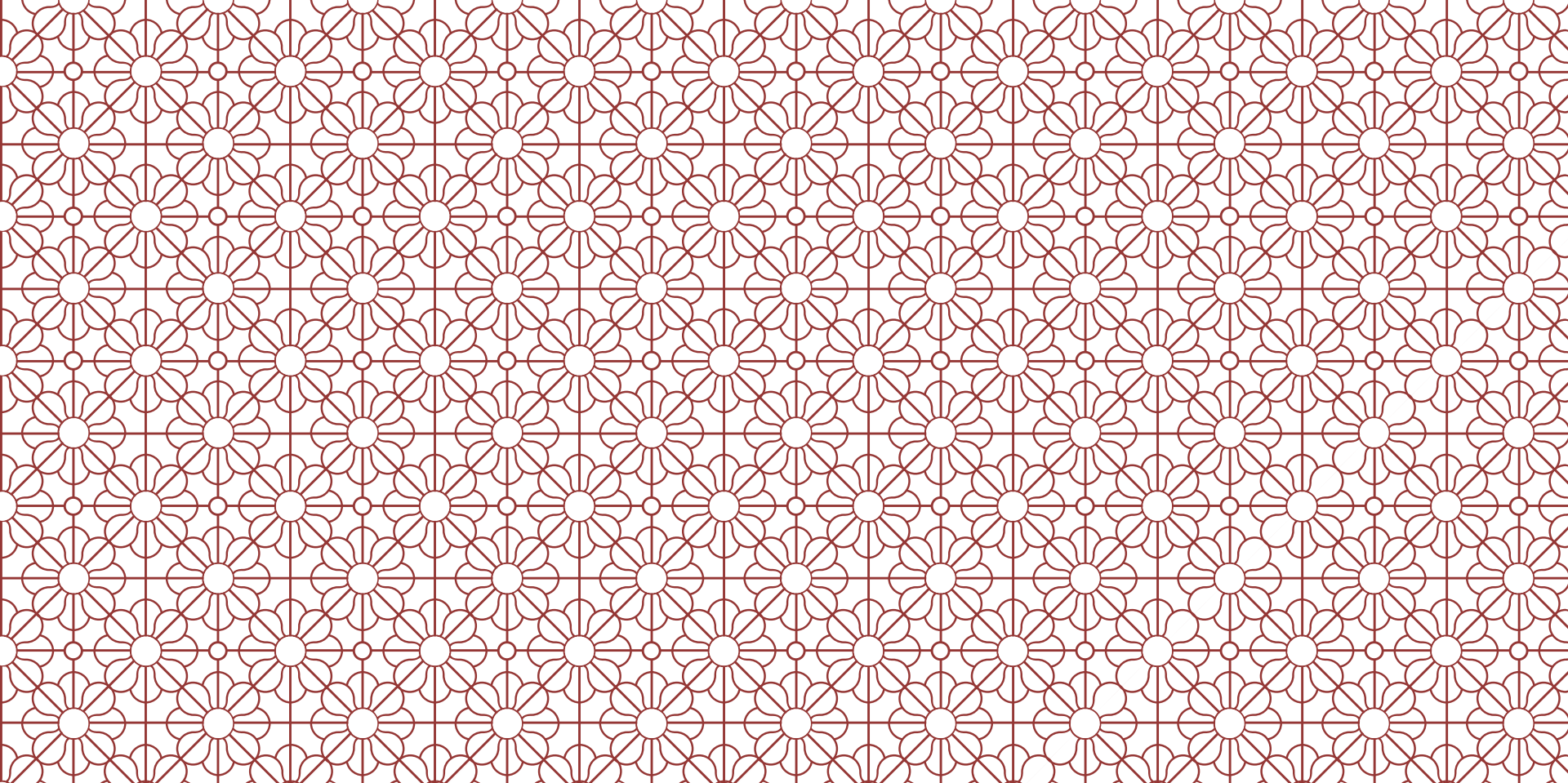


# MOORE OU MEALY?

Em geral, a versão Mealy de um circuito sequencial será mais econômica de componentes físicos (hardware) do que a versão Moore;

Como a saída depende da entrada, valores incorretos na entrada durante o ciclo de “clock” podem afetar a saída;

Isso pode não ocorrer na versão Moore, pois alterações na saída e no estado só ocorrem na transição do “clock” (melhor sincronismo)



# EXEMPLO DE PROJETO DE CIRCUITO SEQUENCIAL

# EXEMPLO: PROBLEMA

Observar uma fileira de 3 lâmpadas;

As lâmpadas só acendem uma de cada vez;

Se as lâmpadas acenderem na sequência 1–2–3, deve-se soar um alarme.

# EXEMPLO: PROBLEMA

A sequência deve ser analisada.

Se a condição 1-2-3 não for observada, despreza-se até a lâmpada 1 acender novamente;

Exemplo de sequência: 1 2 2 1 3 2 1 2 3 1 2 2 2 3 3

Quantas entradas?

Quantas saídas?

# EXEMPLO: ANÁLISE

## Quantas Entradas? 4

- 00 – nenhuma lâmpada acendeu
- 01 – lâmpada 1 acendeu
- 10 – lâmpada 2 acendeu
- 11 – lâmpada 3 acendeu

## Quantas Saídas? 2

- 0 –alarme não toca
- 1 – alarme toca





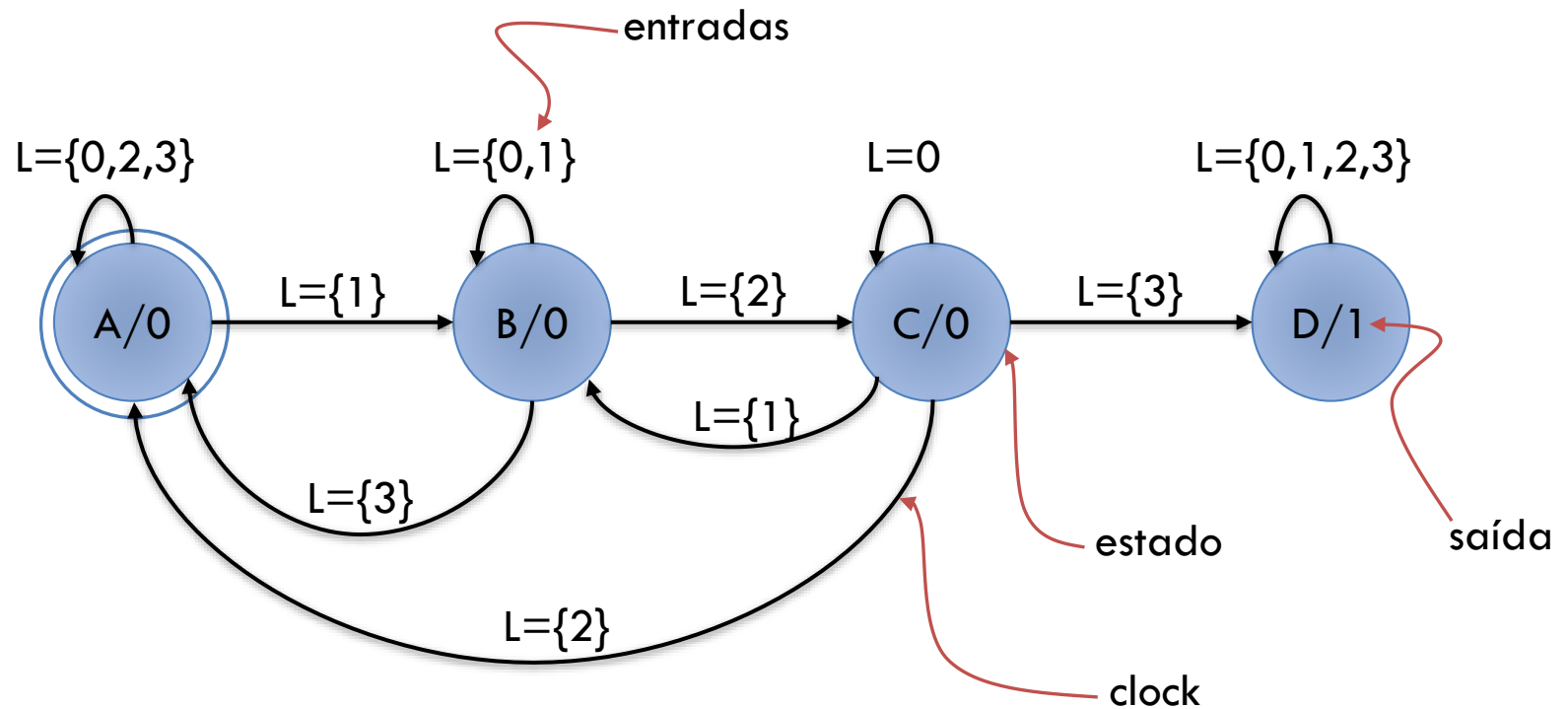
# EXEMPLO: PROBLEMA

Uma fileira de 3 lâmpadas, que só acendem uma de cada vez;

Se as acenderem na sequência 1–2–3, deve-se soar um alarme.

Exemplo de sequência: 1 2 2 1 3 2 1 2 3 1 2 2 2 3 3

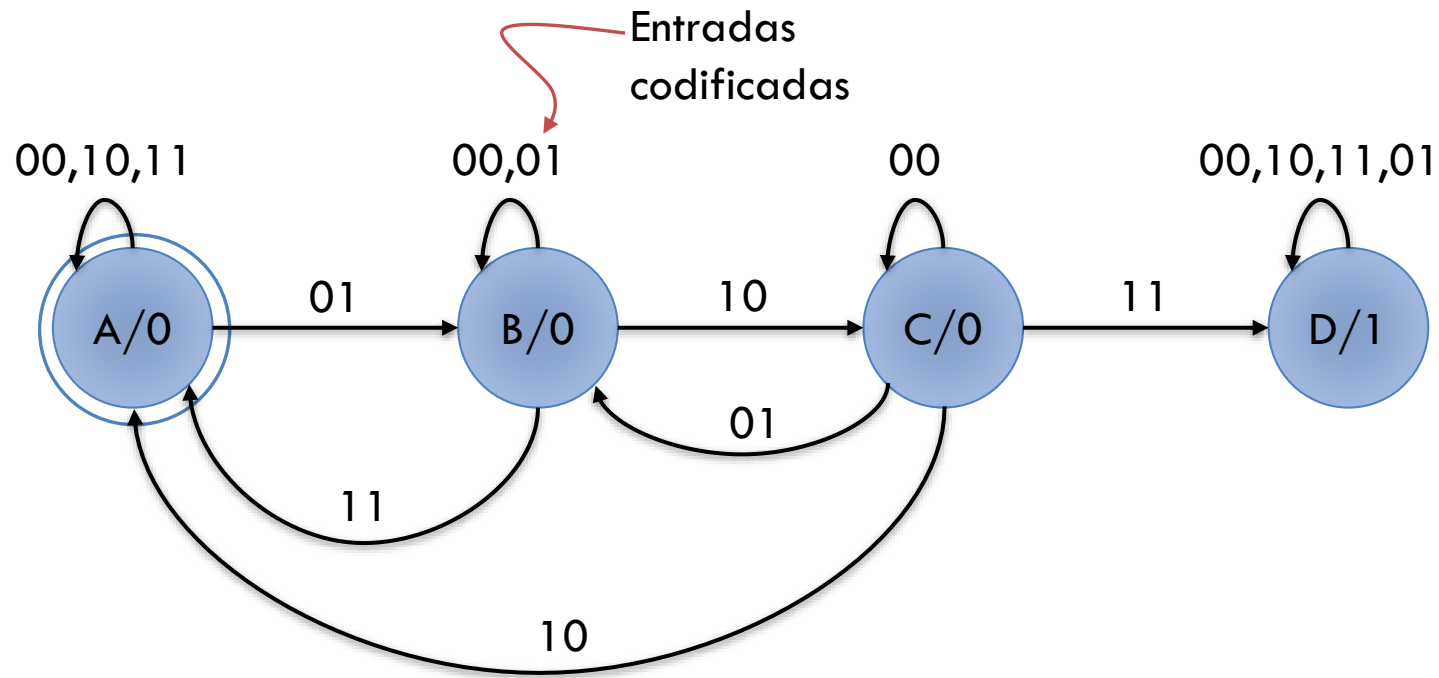
# EXEMPLO: DIAGRAMA DE ESTADOS - MOORE



Saídas:

- 0 – alarme desligado
- 1 – alarme ligado

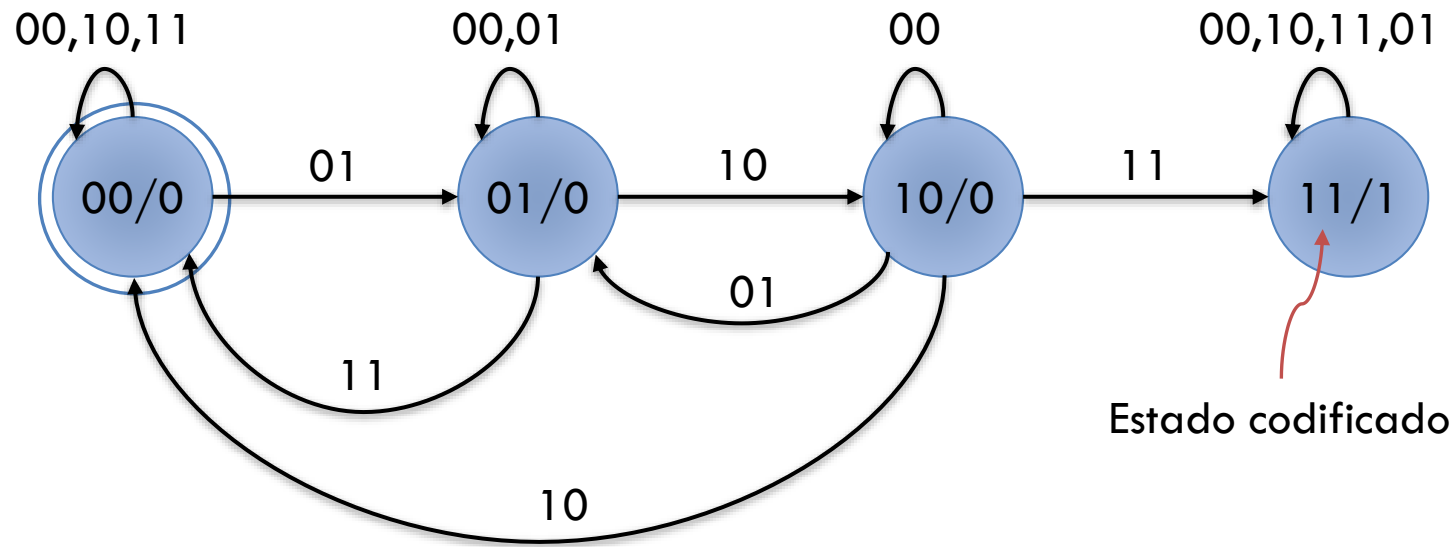
# EXEMPLO: DIAGRAMA DE ESTADOS - MOORE



## Entradas:

- 00 – nenhuma lâmpada acende
- 01 – lâmpada 1 acende
- 10 – lâmpada 2 acende
- 11 – lâmpada 3 acende

# EXEMPLO: DIAGRAMA DE ESTADOS - MOORE



Estados:

- A = 00
- B = 01
- C = 10
- D = 11

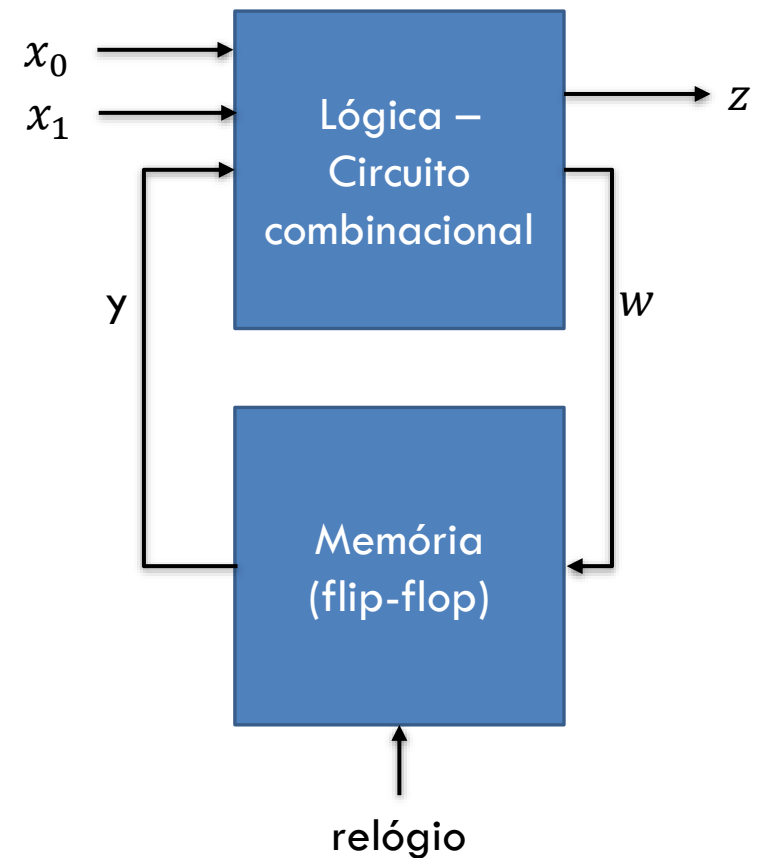
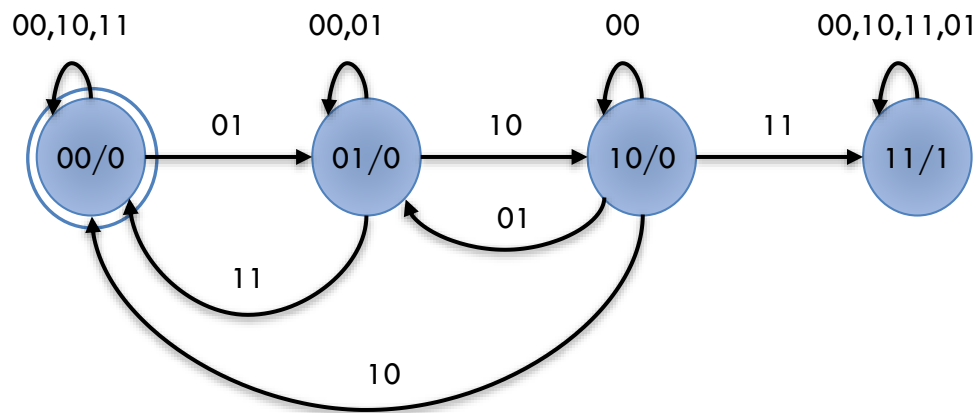
# EXEMPLO: PROJETO DE CIRCUITO SEQUENCIAL - MOORE

$X_0$  e  $X_1$  são as entradas atuais

$Z$  é a saída atual

$W$  indica o próximo estado

$Y$  indica o estado atual

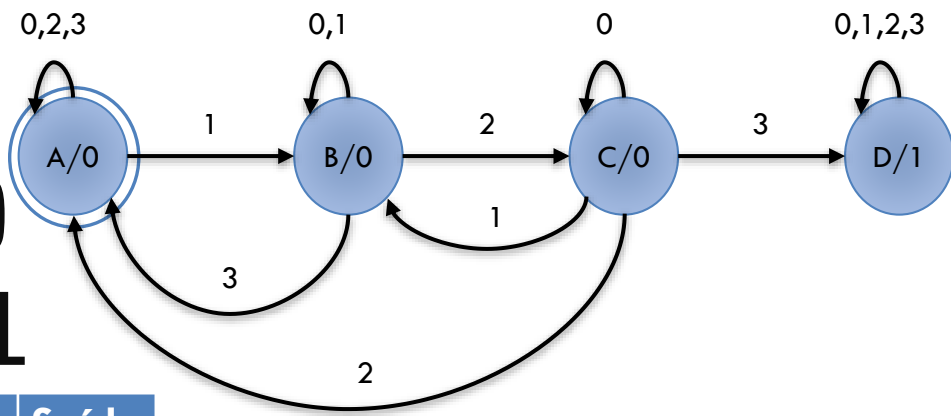


# EXEMPLO: SÍNTESE DO CIRCUITO SEQUENCIAL

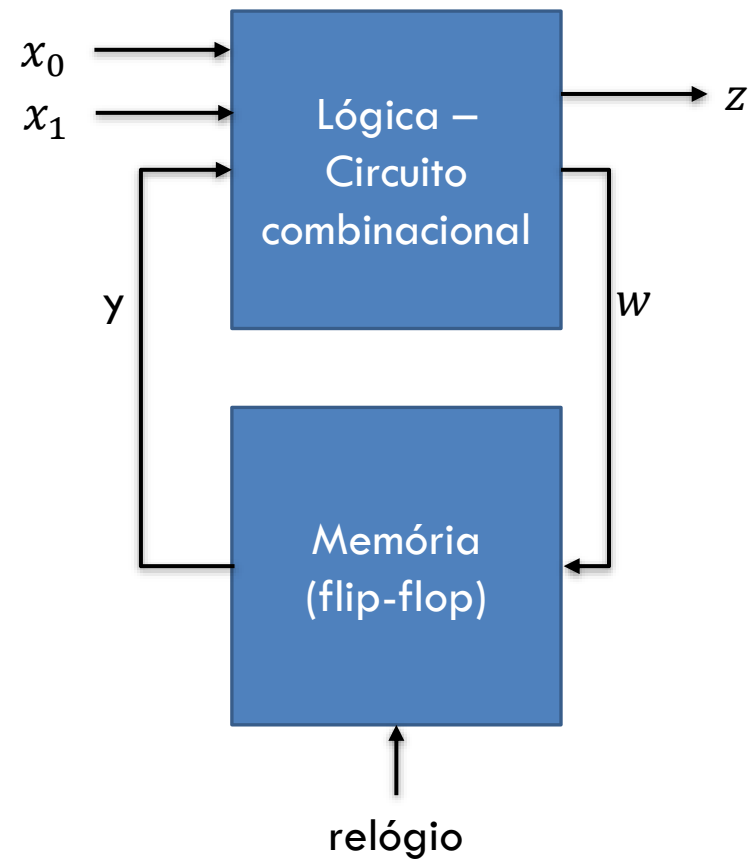
A partir do diagrama de estados, escreve-se a Tabela de Transição de estados e a Tabela de Saída.

- A partir dessa Tabela, projeta-se o circuito sequencial escolhendo qual o tipo de FF que será utilizado (RS, JK, D ou T)
- Circuito combinatório: portas lógicas;
- Circuito de memória: Flip-Flops;

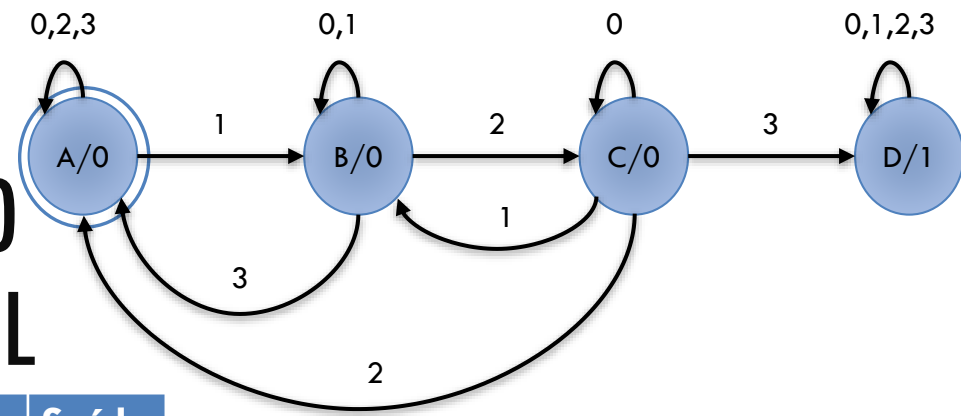
# EXEMPLO: SÍNTESE DO CIRCUITO SEQUENCIAL



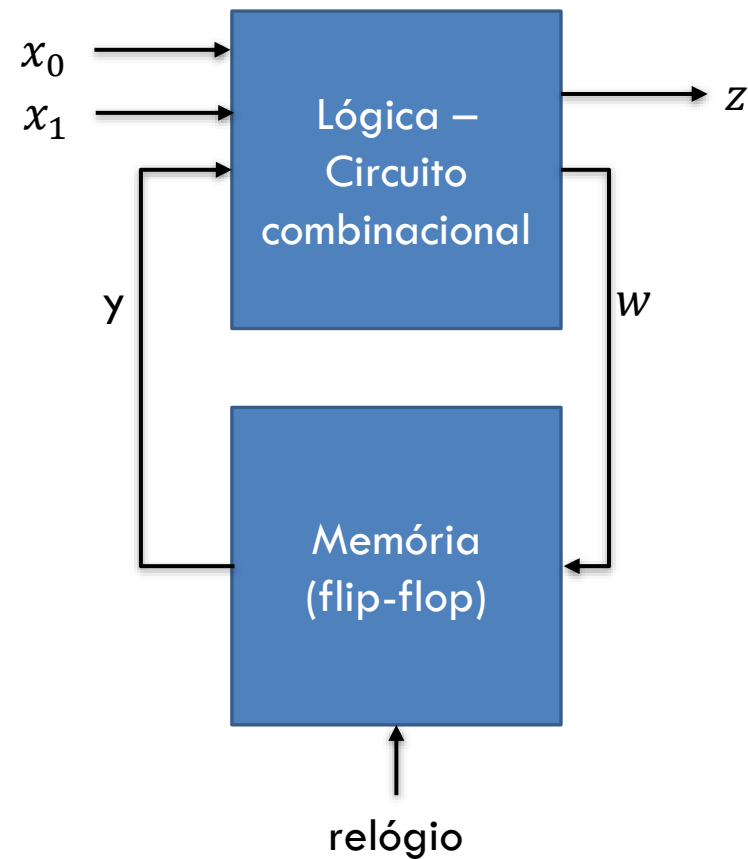
Entrada		Estado Atual		Próx. Estado		Saída
x1	x0	y1	y0	w1	w0	z
0		A				
1		A				
2		A				
3		A				
0		B				
1		B				
2		B				
3		B				
0		C				
1		C				
2		C				
3		C				
0		D				
1		D				
2		D				
3		D				



# EXEMPLO: SÍNTESE DO CIRCUITO SEQUENCIAL

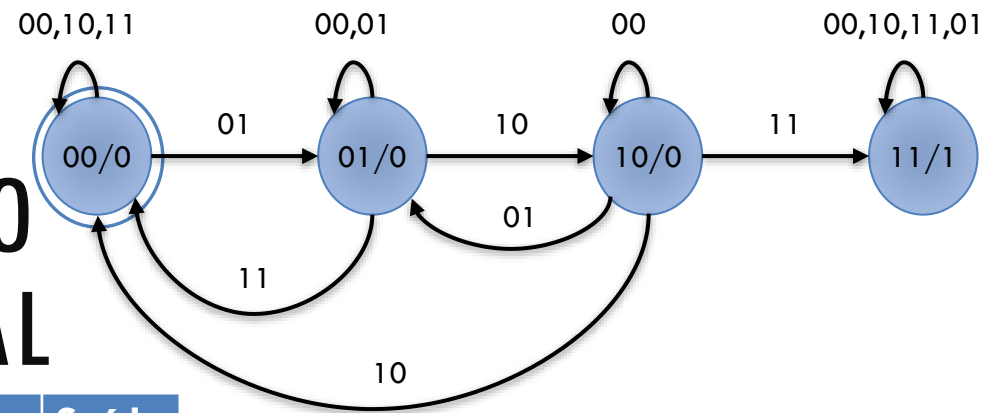


Entrada		Estado Atual		Próx. Estado		Saída
x1	x0	y1	y0	w1	w0	z
0		A		A		0
1		A		B		0
2		A		A		0
3		A		A		0
0		B		B		0
1		B		B		0
2		B		A		0
3		B		A		0
0		C		C		0
1		C		B		0
2		C		A		0
3		C		D		0
0		D		D		1
1		D		D		1
2		D		D		1
3		D		D		1

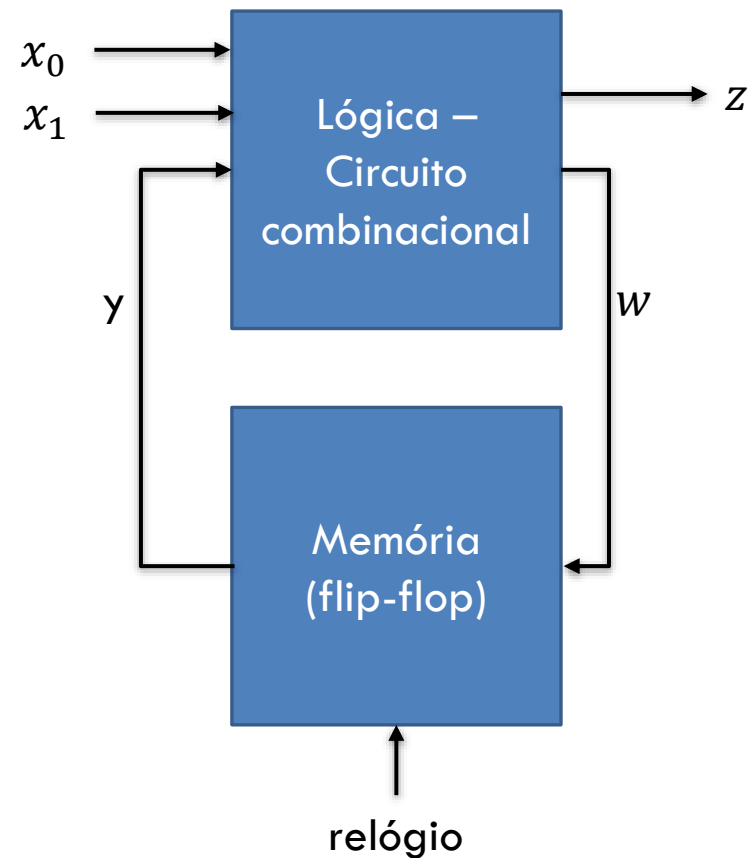




# EXEMPLO: SÍNTESE DO CIRCUITO SEQUENCIAL



Entrada		Estado Atual		Próx. Estado		Saída
x1	x0	y1	y0	w1	w0	z
0	0	0	0	0	0	0
0	1	0	0	0	1	0
1	0	0	0	0	0	0
1	1	0	0	0	0	0
0	0	0	1	0	1	0
0	1	0	1	0	1	0
1	0	0	1	1	0	0
1	1	0	1	0	0	0
0	0	1	0	1	0	0
0	1	1	0	0	1	0
1	0	1	0	0	0	0
1	1	1	0	1	1	0
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	1	1	1	1	1
1	1	1	1	1	1	1



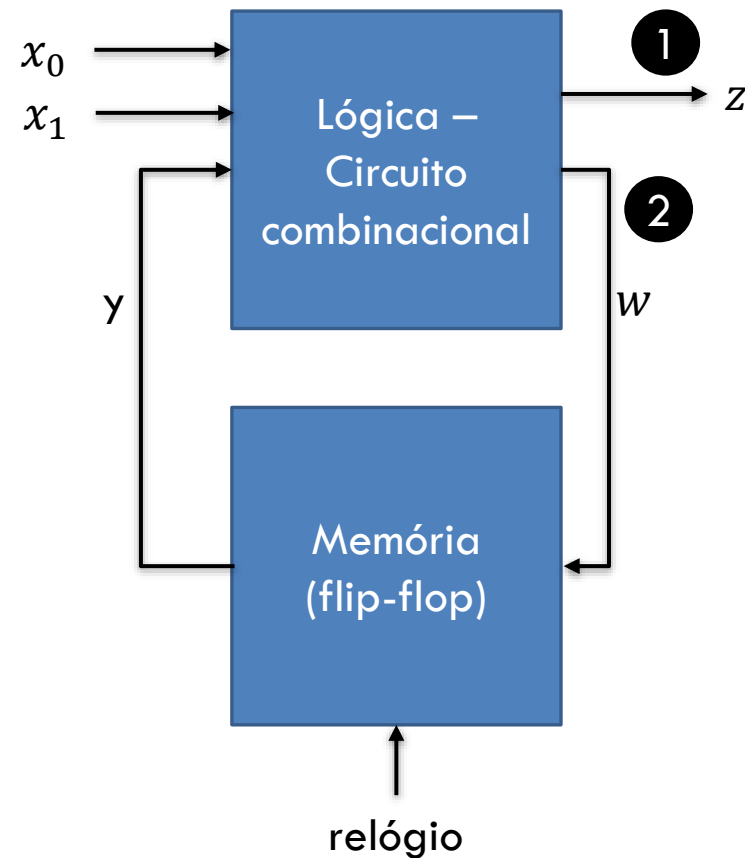
# EXEMPLO: DIVISÃO DA TABELA

Para simplificar a análise, podemos dividir a tabela de estados em duas

1. Tabela de Transição de estados

2. Tabela de Saída

- Máquina de moore: Contém apenas o estado atual para gerar a saída
- Máquina de mealy: Contém os estados atuais e as entradas para gerar a saída



# EXEMPLO: TABELA DE TRANSIÇÃO DE ESTADOS

Entrada		Estado Atual		Próx. Estado		FF-D
x1	x0	y1	y0	w1	w0	d1d0
0	0	0	0	0	0	00
0	1	0	0	0	1	01
1	0	0	0	0	0	00
1	1	0	0	0	0	00
0	0	0	1	0	1	01
0	1	0	1	0	1	01
1	0	0	1	1	0	10
1	1	0	1	0	0	00
0	0	1	0	1	0	10
0	1	1	0	0	1	01
1	0	1	0	0	0	00
1	1	1	0	1	1	11
0	0	1	1	1	1	11
0	1	1	1	1	1	11
1	0	1	1	1	1	11
1	1	1	1	1	1	11

Podemos usar outros tipos de FF, basta pensar no que devemos colocar como entrada para que ele escreva o que queremos

## EXEMPLO: SIMPLIFICANDO COM MAPAS DE KARNAUGH

$$D_1 = Y_1Y_0 + X'_1X'_0Y_1 + X_1X_0Y_1 + X_1X'_0Y_0$$

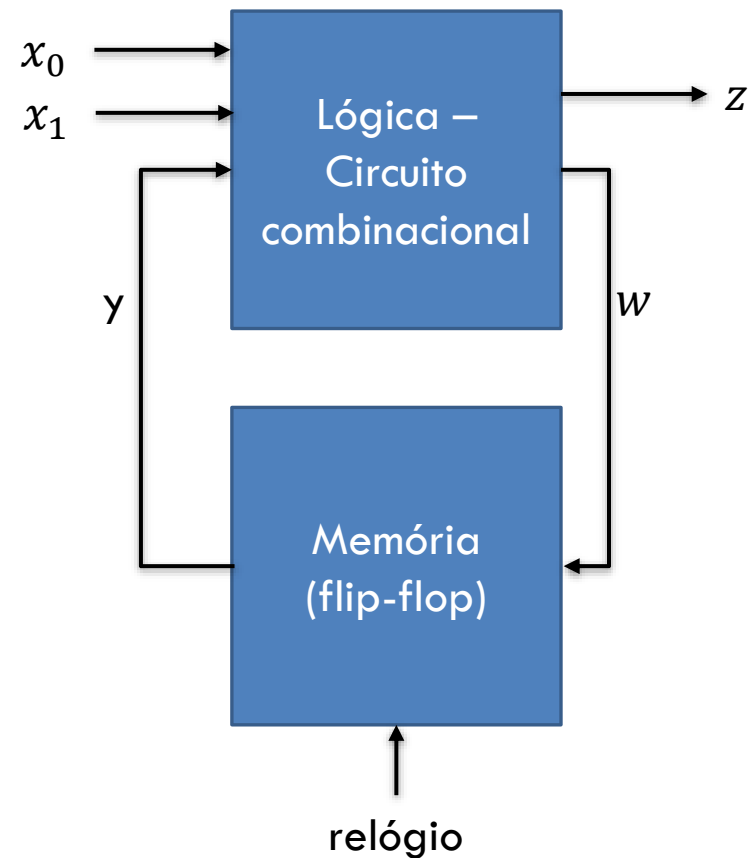
$$D_0 = Y_1Y_0 + X'_1X_0 + X_1X_0Y_1 + X'_1X'_0Y_0$$

# EXEMPLO: TABELA DE SAÍDA

A Saída nunca depende do próximo estado

## Máquina de Moore:

- A saída não depende da entrada
- A Saída só depende do estado atual



# EXEMPLO: TABELA DE SAÍDA

		Estado Atual				Saída
		y1	y0			z
		0	0			0
		0	0			0
		0	0			0
		0	0			0
		0	1			0
		0	1			0
		0	1			0
		0	1			0
		1	0			0
		1	0			0
		1	0			0
		1	0			0
		1	1			1
		1	1			1
		1	1			1
		1	1			1



Estado Atual		Saída
y1	y0	z
0	0	0
0	1	0
1	0	0
1	1	1

$$z = Y_1 Y_0$$

# EXEMPLO: CIRCUITO FINAL

Entrada		Estado Atual		Próx. Estado	
x1	x0	y1	y0	w1	w0

$$W_1 = Y_1 Y_0 + X'_1 X'_0 Y_1 + X_1 X_0 Y_1 + X_1 X'_0 Y_0$$

← Mem. 1

$$W_0 = Y_1 Y_0 + X'_1 X_0 + X_1 X_0 Y_1 + X'_1 X'_0 Y_0$$

← Mem. 0

$$Z = Y_1 Y_0 \quad \leftarrow \text{Saída}$$

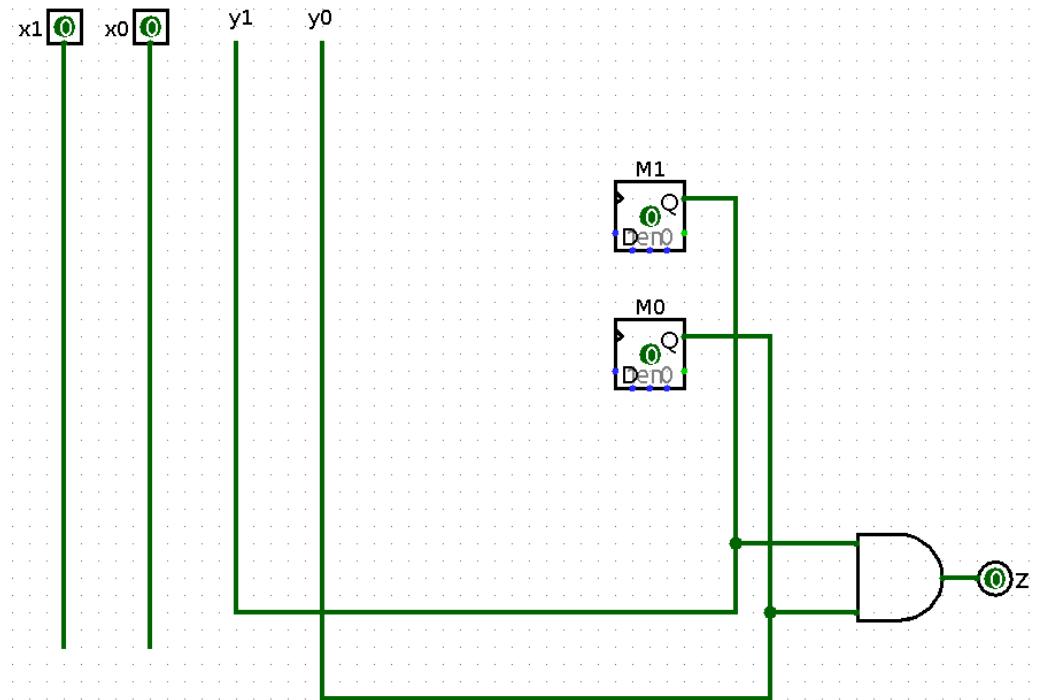
# EXEMPLO: CIRCUITO FINAL

Entrada		Estado Atual		Próx. Estado	
x1	x0	y1	y0	w1	w0

$$W_1 = Y_1Y_0 + X'_1X'_0Y_1 + X_1X_0Y_1 + X_1X'_0Y_0$$

$$W_0 = Y_1Y_0 + X'_1X_0 + X_1X_0Y_1 + X'_1X'_0Y_0$$

$$Z = Y_1Y_0$$





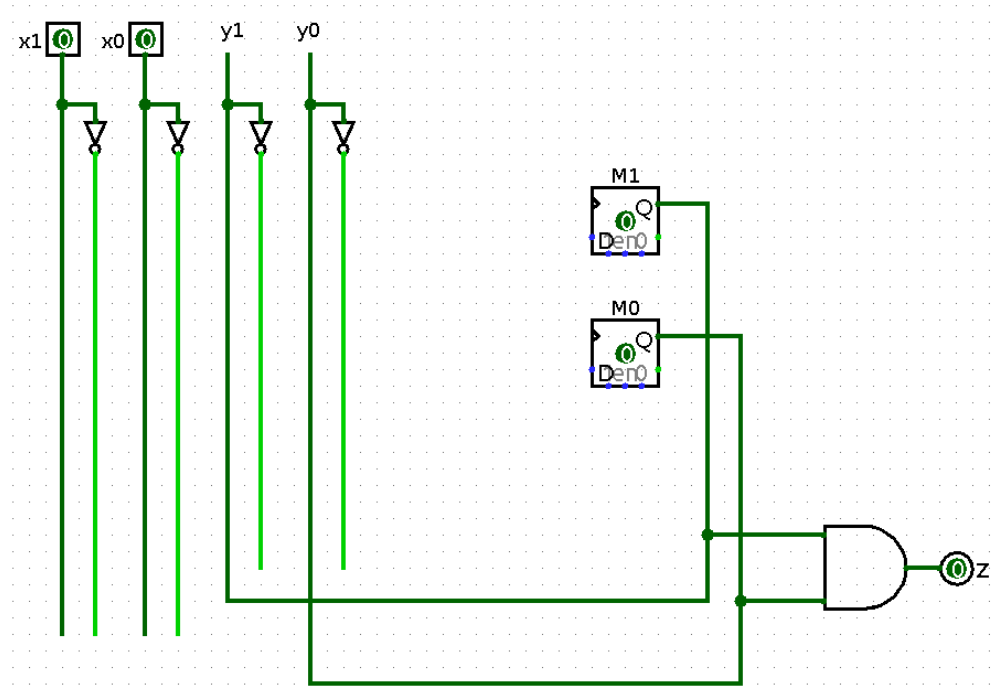
## EXEMPLO: CIRCUITO FINAL

Entrada		Estado Atual		Próx. Estado	
x1	x0	y1	y0	w1	w0

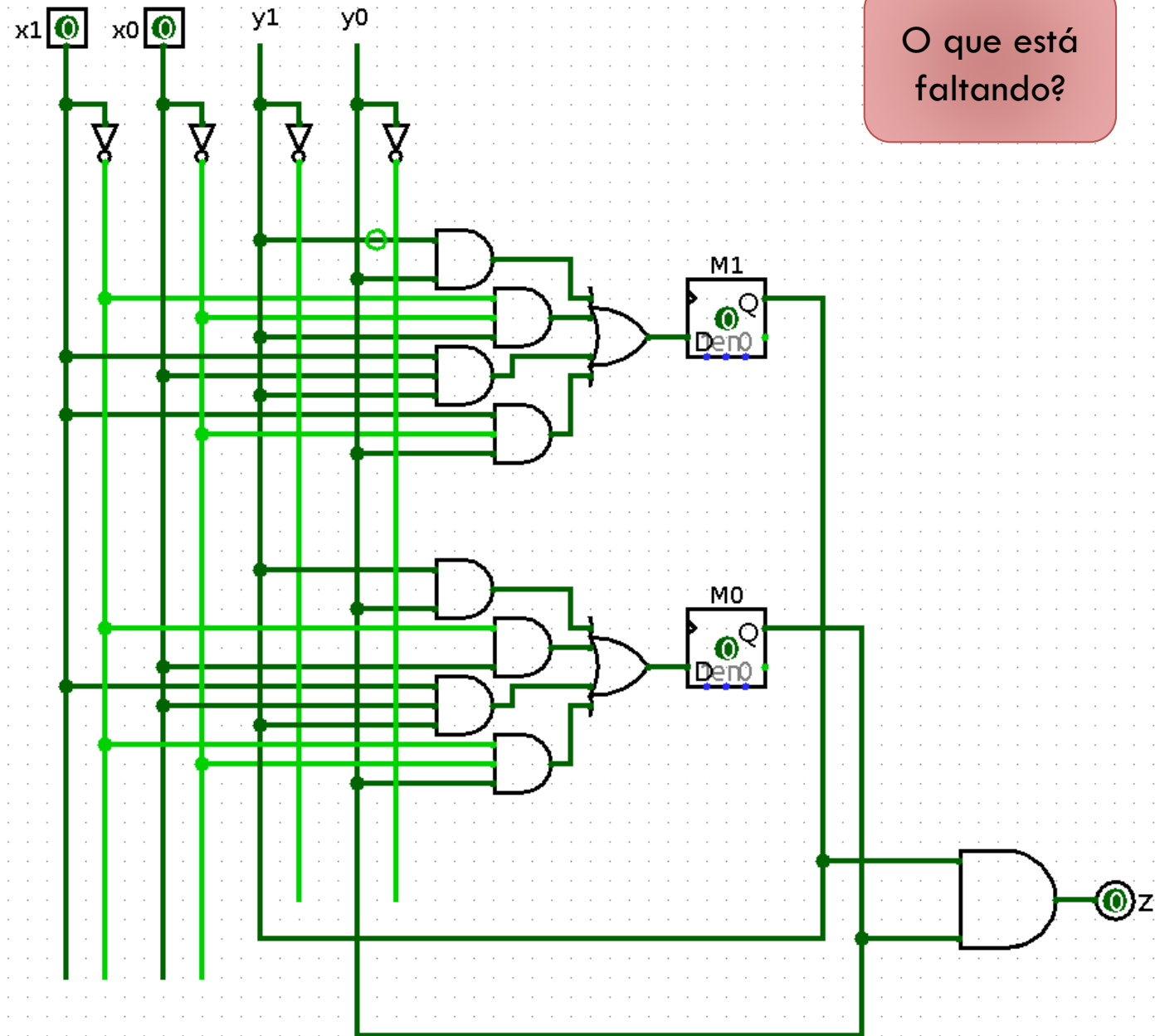
$$W_1 = Y_1 Y_0 + X'_1 X'_0 Y_1 + X_1 X_0 Y_1 + X_1 X'_0 Y_0$$

$$W_0 = Y_1 Y_0 + X_1' X_0 + X_1 X_0 Y_1 + X_1' X_0' Y_0$$

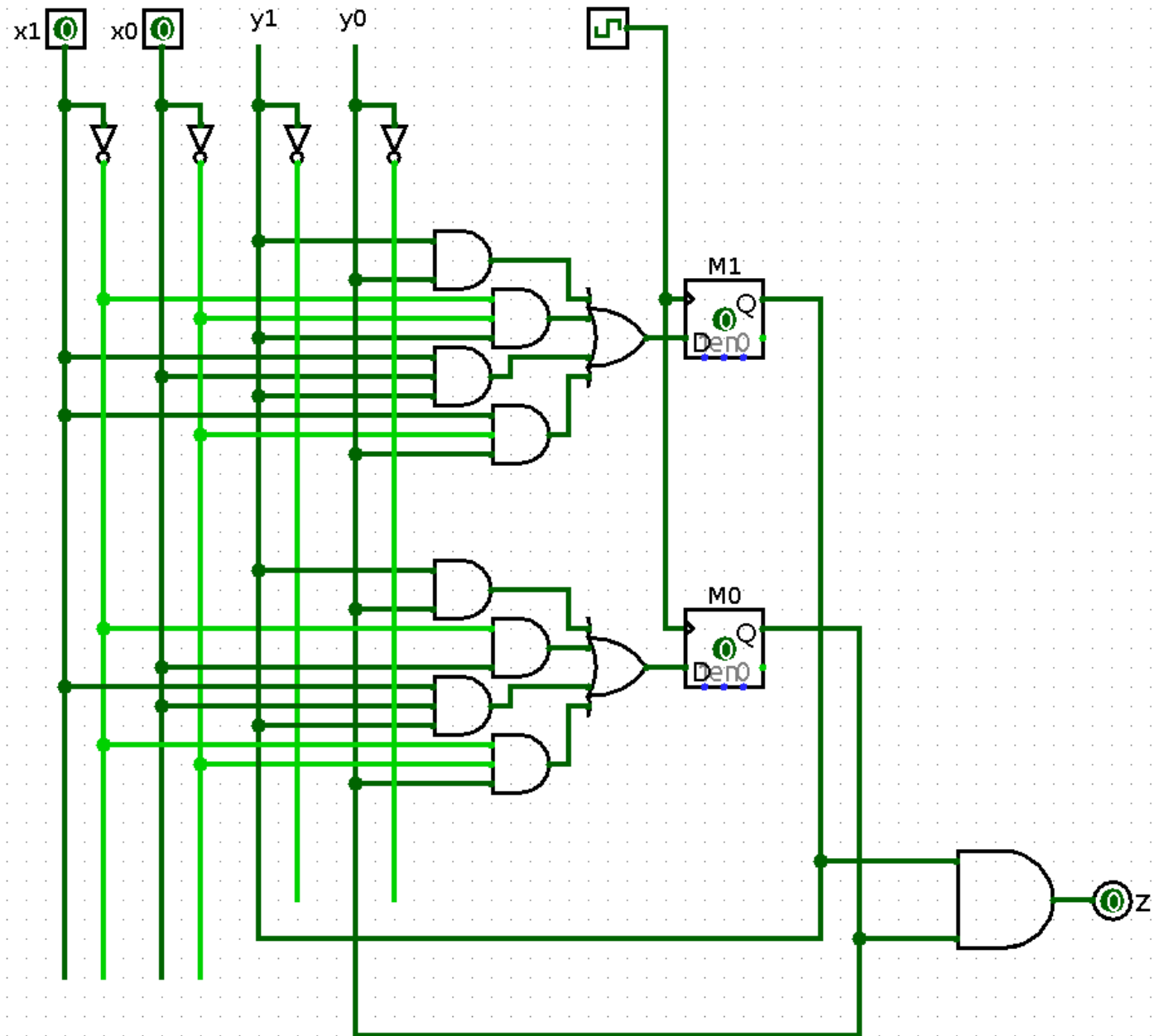
$$Z = Y_1 Y_0$$



# EXEMPLO: CIRCUITO FINAL



100



# ALGORITMO PARA MÁQUINA DE ESTADOS

1. Modelar o diagrama de boletas
2. Codificar os estados
3. Criar a tabela de transições
4. Para cada saída (saída externa / próximo estado) obter a equação booleana simplificada.
5. Fazer o circuito final com memórias para cada bit de estado atual