

PCS 3115

Sistemas Digitais I

Aula 18 – Síntese de Circuitos Sequenciais

Preparado por Prof. Marcos A. Simplicio Jr.

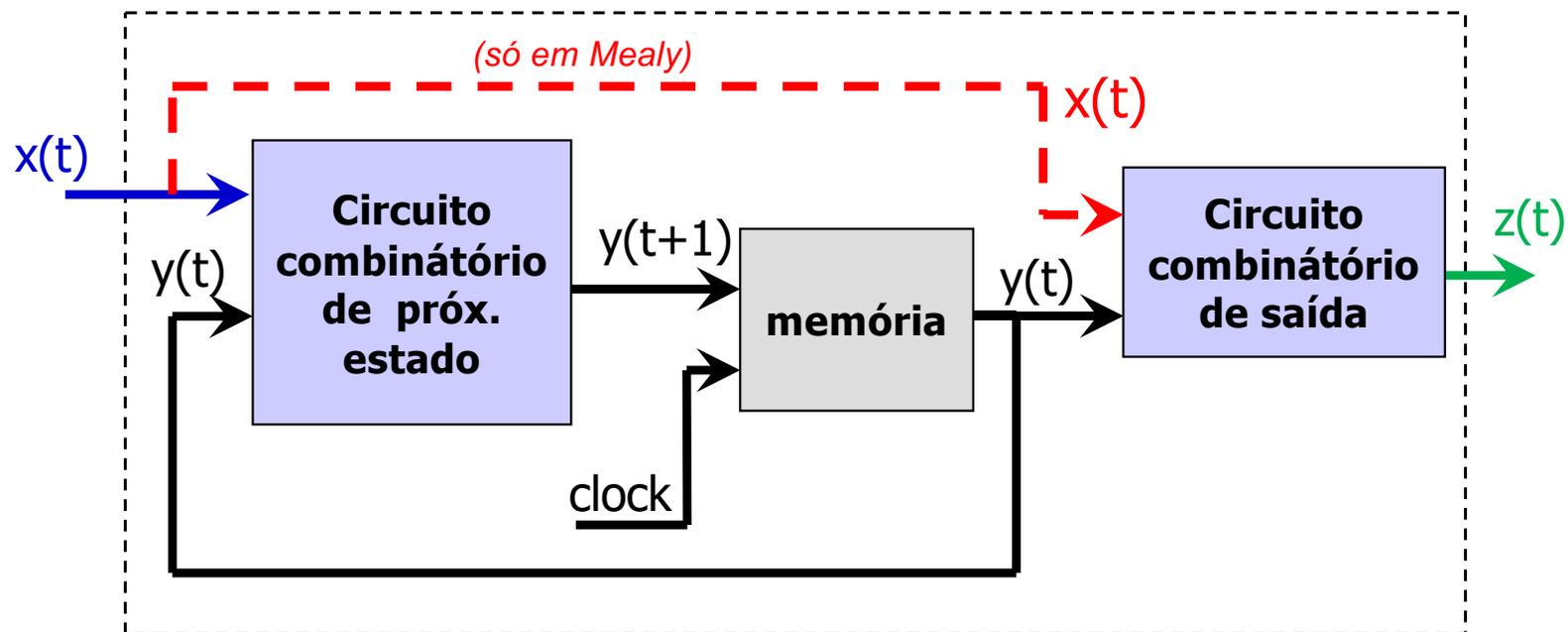
Ministrado por Prof. Edson S. Gomi

Maio de 2020

Versão 1

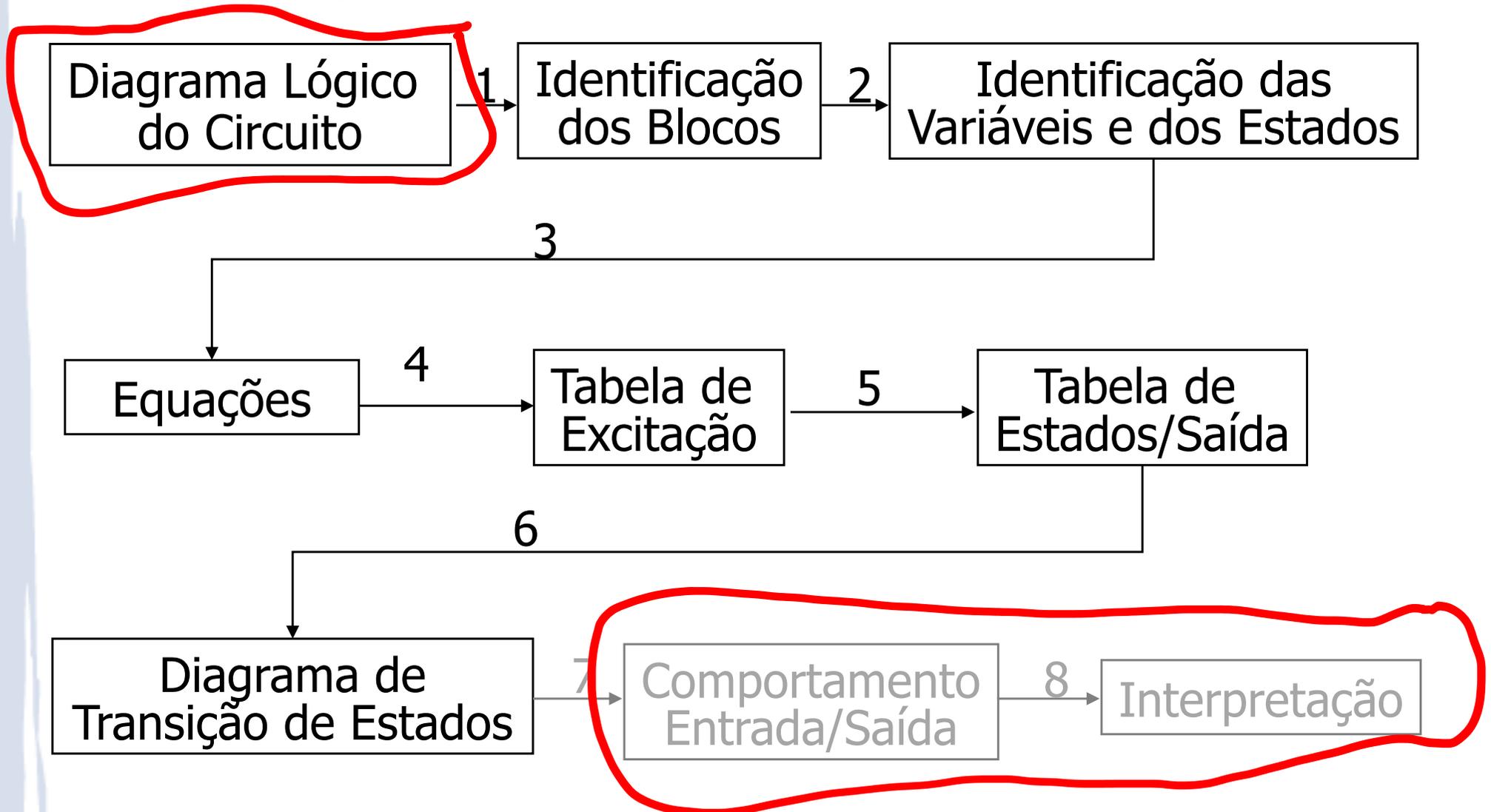
Método de Síntese

- Método aqui adotado segue praticamente os mesmos passos da metodologia de análise, com os passos sendo adotados em ordem inversa.
 - Para projetar circuitos mais complexos, como CPUs de computadores, existem métodos mais eficientes.
- Pode-se escolher modelo de Mealy ou de Moore.

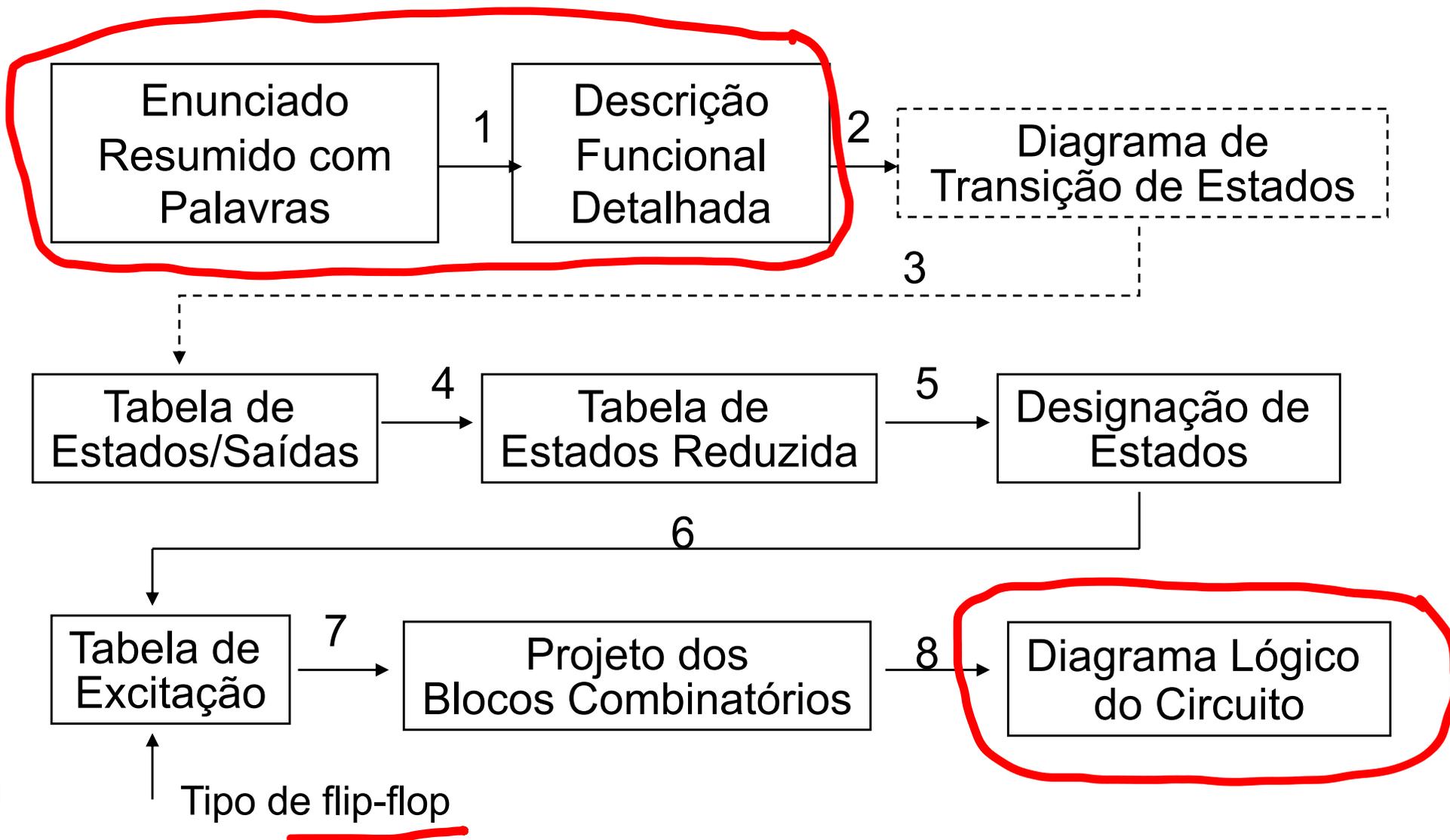


Análise de circuitos sequenciais

- Passo a passo



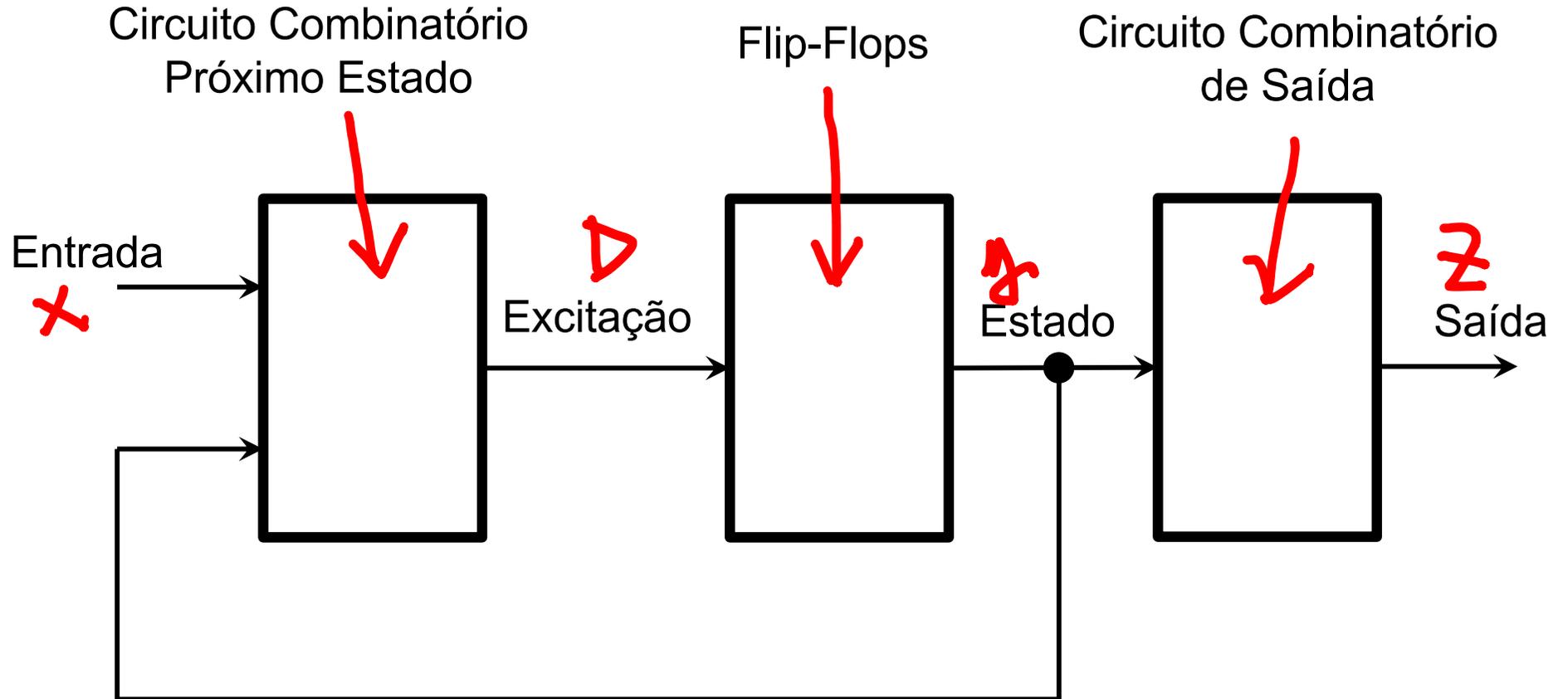
Método de Síntese



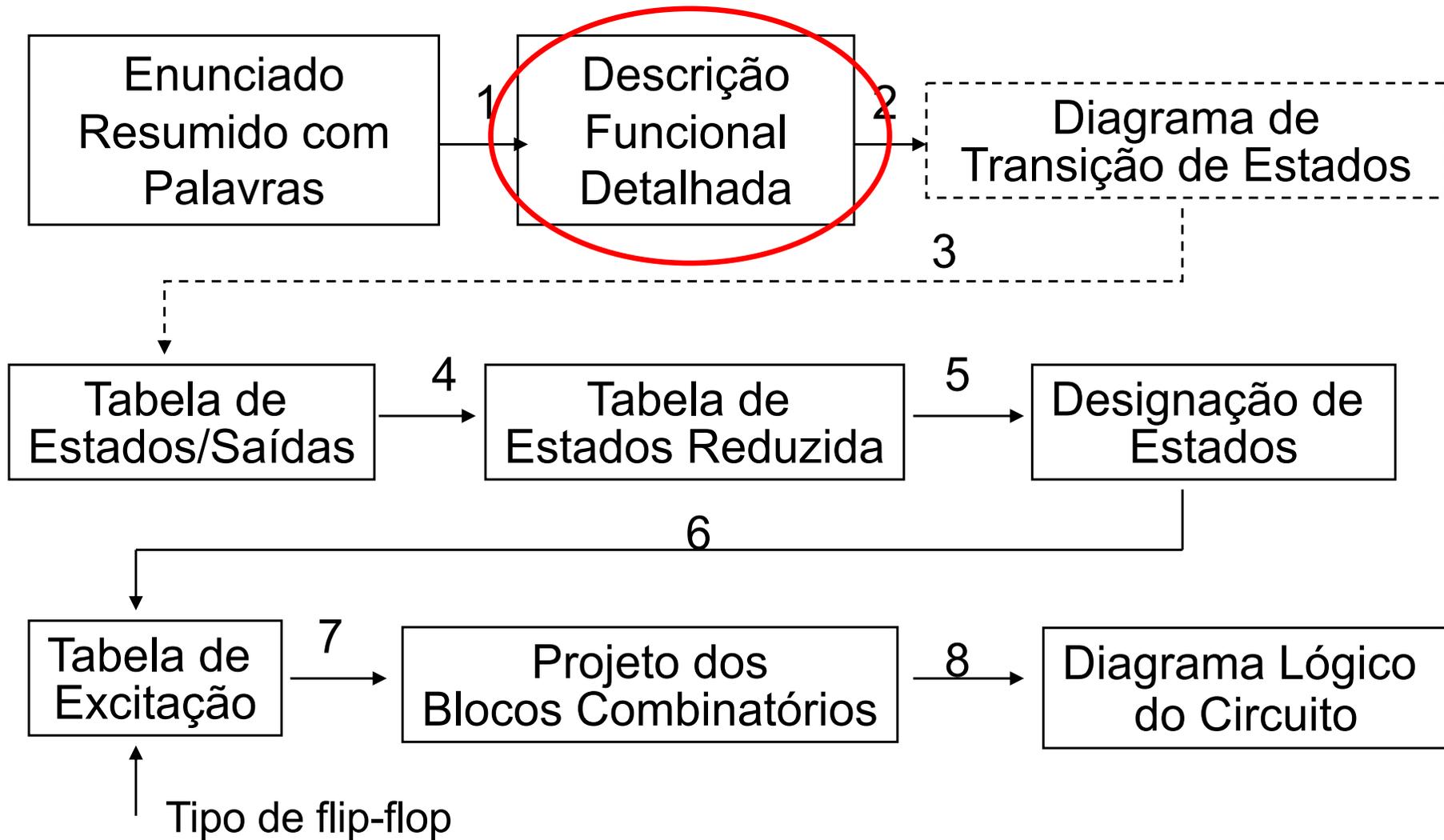
Exemplo de Síntese - Moore

- **Enunciado:** Projetar um circuito sequencial síncrono que reconhece o primeiro ZERO após a ocorrência de dois ou mais UNS consecutivos. Adotar uma solução do tipo Moore e flip-flops tipo D.

Modelo de Moore



Exemplo de Síntese - Moore

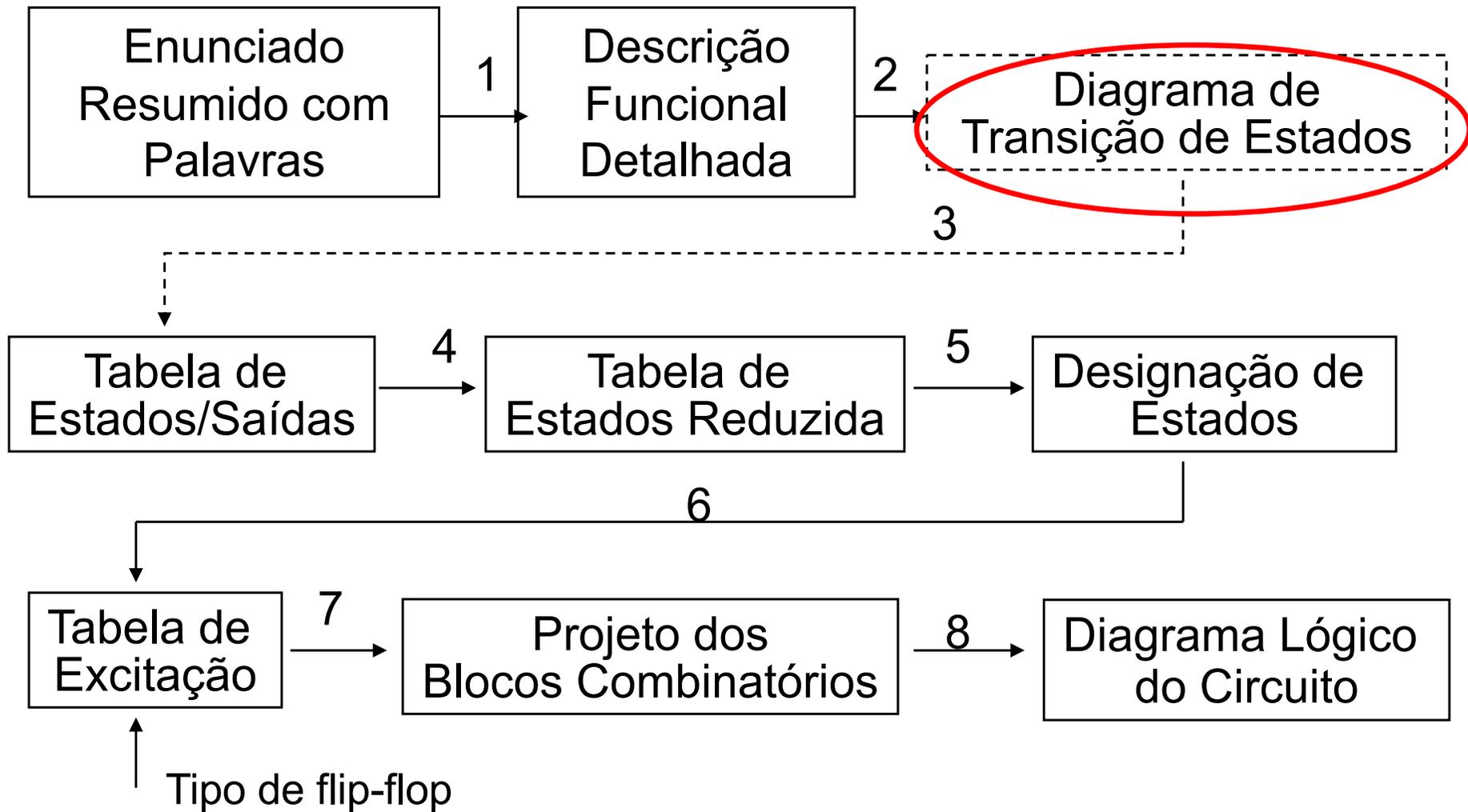


Exemplo de Síntese - Moore

- **Enunciado:** Projetar um circuito sequencial síncrono que reconhece o primeiro ZERO após a ocorrência de dois ou mais UNS consecutivos. Adotar uma solução do tipo Moore e flip-flops tipo D.

Entrada	0	1	0	1	1	0	0	1	1	1	0	1	0	0	0
Saída	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Exemplo de Síntese - Moore



Exemplo de Síntese - Moore

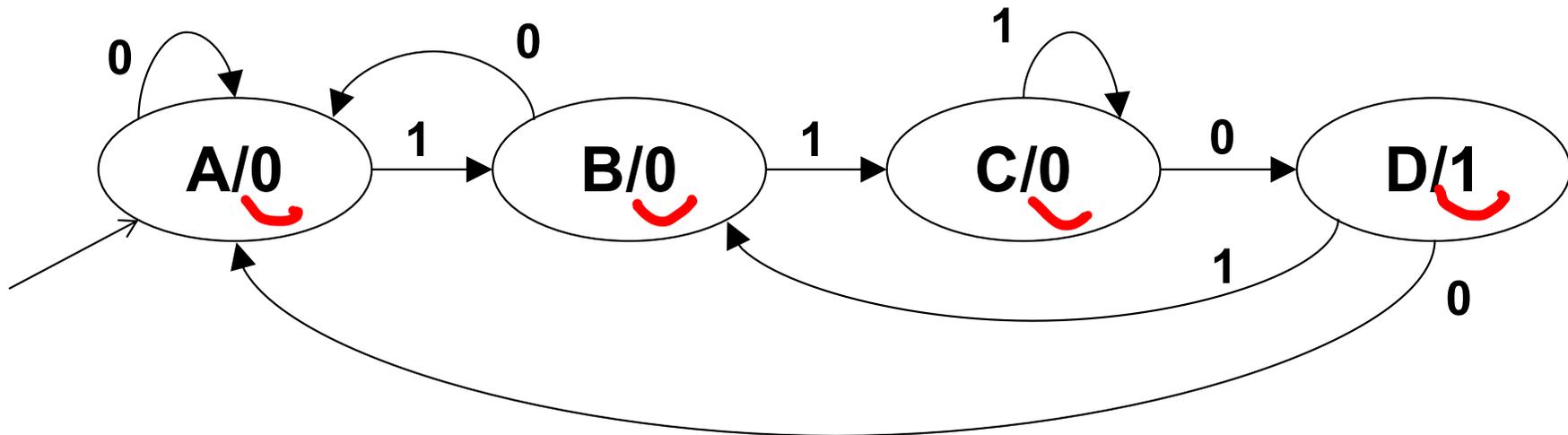
Etapa 2: Diagrama de Transição de Estados

Serão necessários pelo menos 4 estados:

- A: estado inicial
- B: armazena a ocorrência do primeiro UM
- C: armazena a ocorrência do segundo e demais UNS consecutivos
- D: armazena o primeiro ZERO após dois ou mais UNS consecutivos

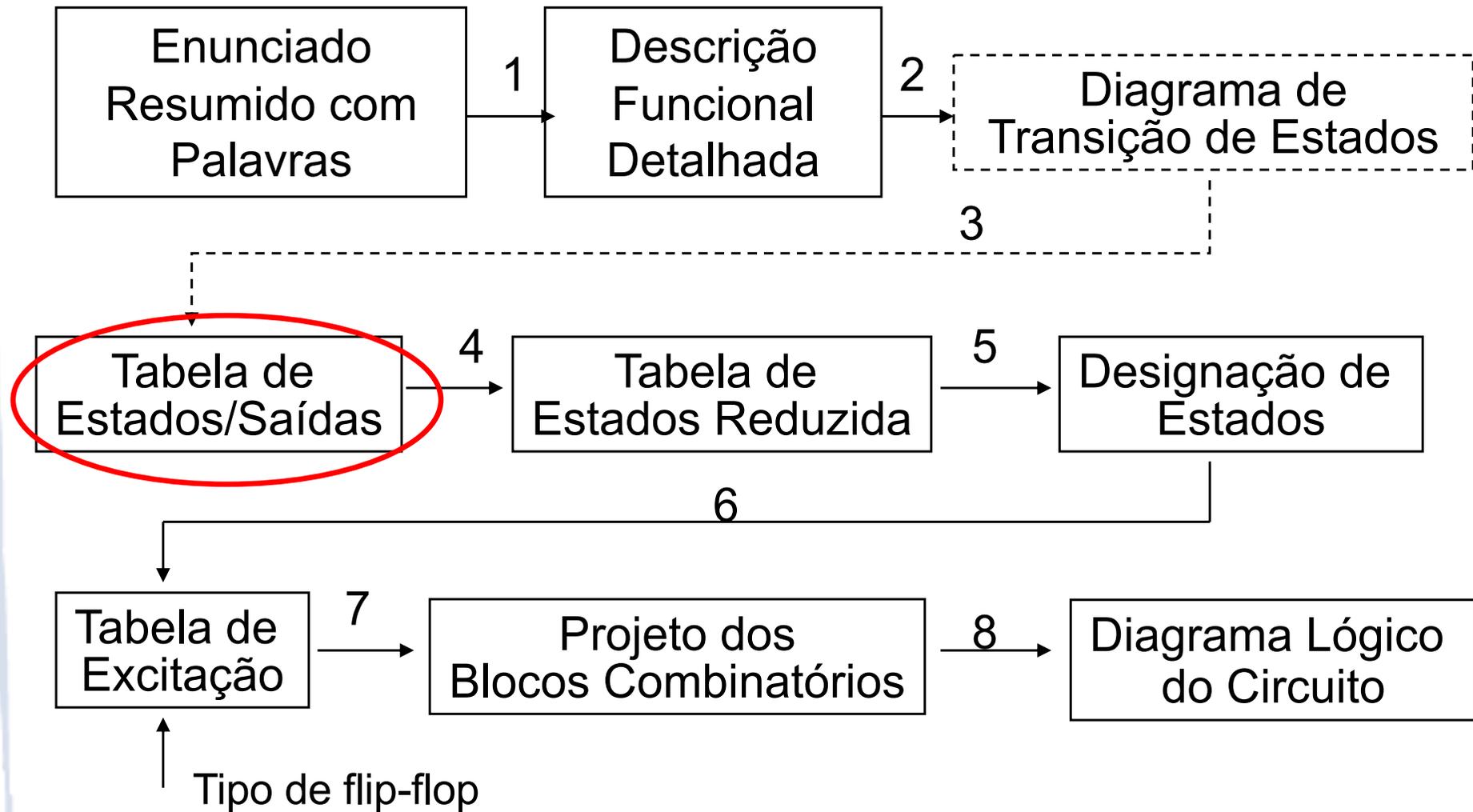
Exemplo de Síntese - Moore

Etapa 2: Diagrama de Transição de Estados



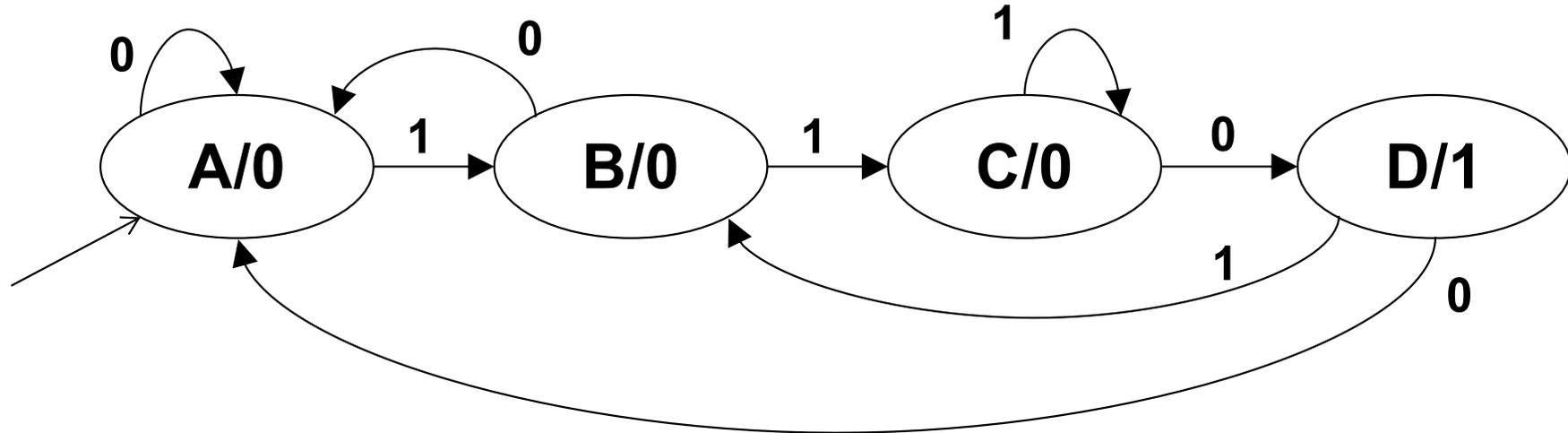
A	Estado inicial: ZEROS
B	Ocorrência de 1 UM
C	Ocorrência de 2+ UNS
D	Ocorrência do 1o ZERO após 2+ UNS

Exemplo de Síntese - Moore



Exemplo de Síntese - Moore

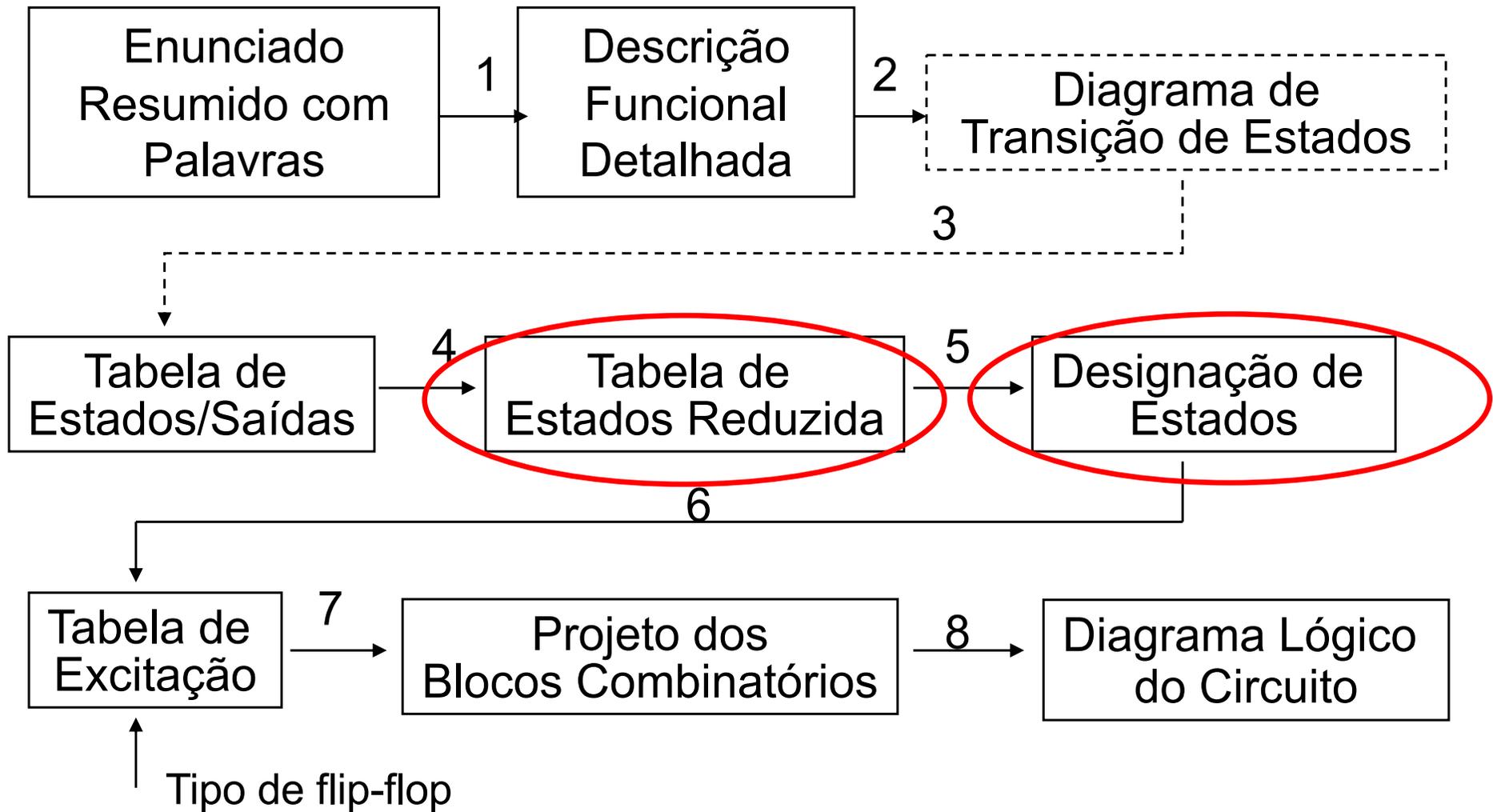
Etapa 3: Tabela de Estados/Saída



<i>Estado</i>	$x = 0$	$x = 1$	z^t
A	A	B	0
B	A	C	0
C	D	C	0
D	A	B	1

Handwritten red annotations:
A bracket on the left groups states D and E. Below the table, the letters A, B, and 1 are written in red under the columns for $x=0$, $x=1$, and z^t respectively.

Exemplo de Síntese - Moore



Exemplo de Síntese - Moore

Etapa 4: Tabela de Estados Reduzida

Não é possível simplificar a tabela anterior

Etapa 5: Designação dos Estados

Como são necessários 4 estados, $n = 2$ e utilizaremos a seguinte designação (arbitrária):

s	y1	y0
A	0	0
B	0	1
C	1	1
D	1	0

g, y0
00
01
10
11

Exemplo de Síntese - Moore

Etapa 5: Designação dos Estados

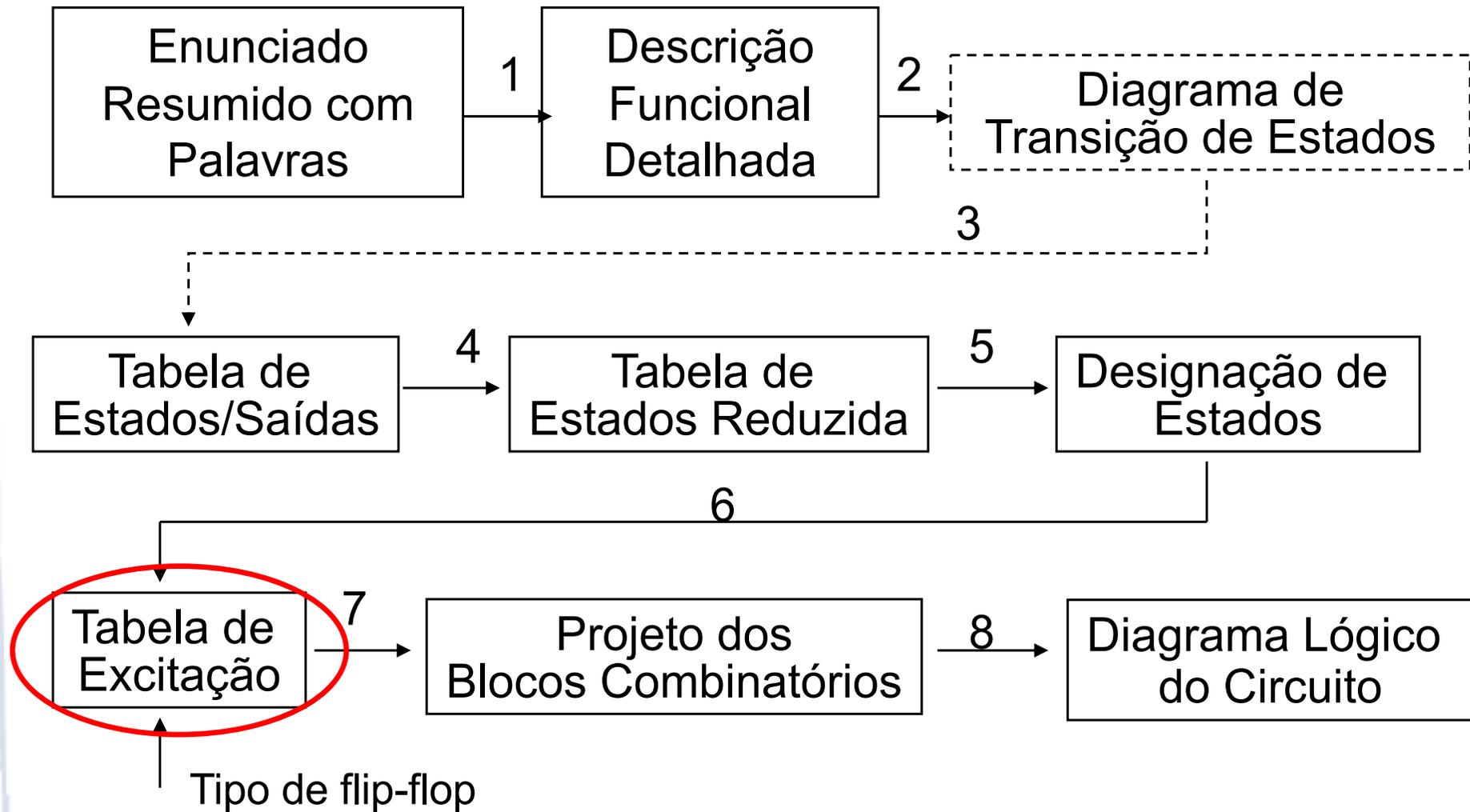
$s^t \backslash x^t$	$x = 0$	$x = 1$	z^t
A	A	B	0
B	A	C	0
C	D	C	0
D	A	B	1



$y1^t y0^t \backslash x^t$	$x = 0$	$x = 1$	z^t
0 0	0 0	0 1	0
0 1	0 0	1 1	0
1 1	1 0	1 1	0
1 0	0 0	0 1	1

$y1^{t+1}y0^{t+1}$

Exemplo de Síntese - Moore



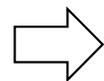
Exemplo de Síntese - Moore

Etapa 6: Tabela de Excitação

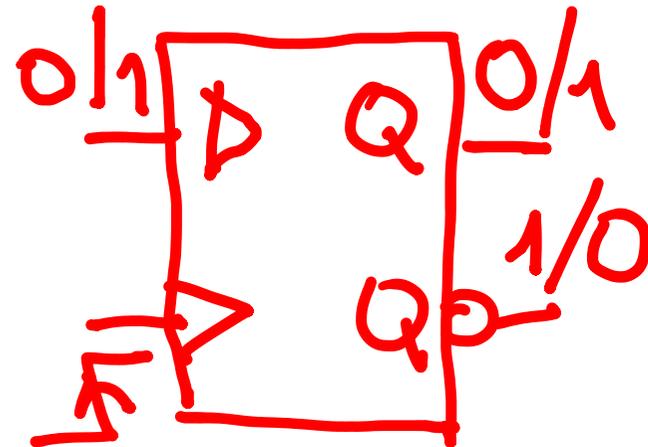
→ Vamos usar Flip-Flops tipo D

$y_1^t y_0^t \backslash x^t$	$x = 0$	$x = 1$	z^t
A 0 0	A 0 0	B 0 1	0
B 0 1	0 0	1 1	0
C 1 1	1 0	1 1	0
D 1 0	0 0	0 1	1

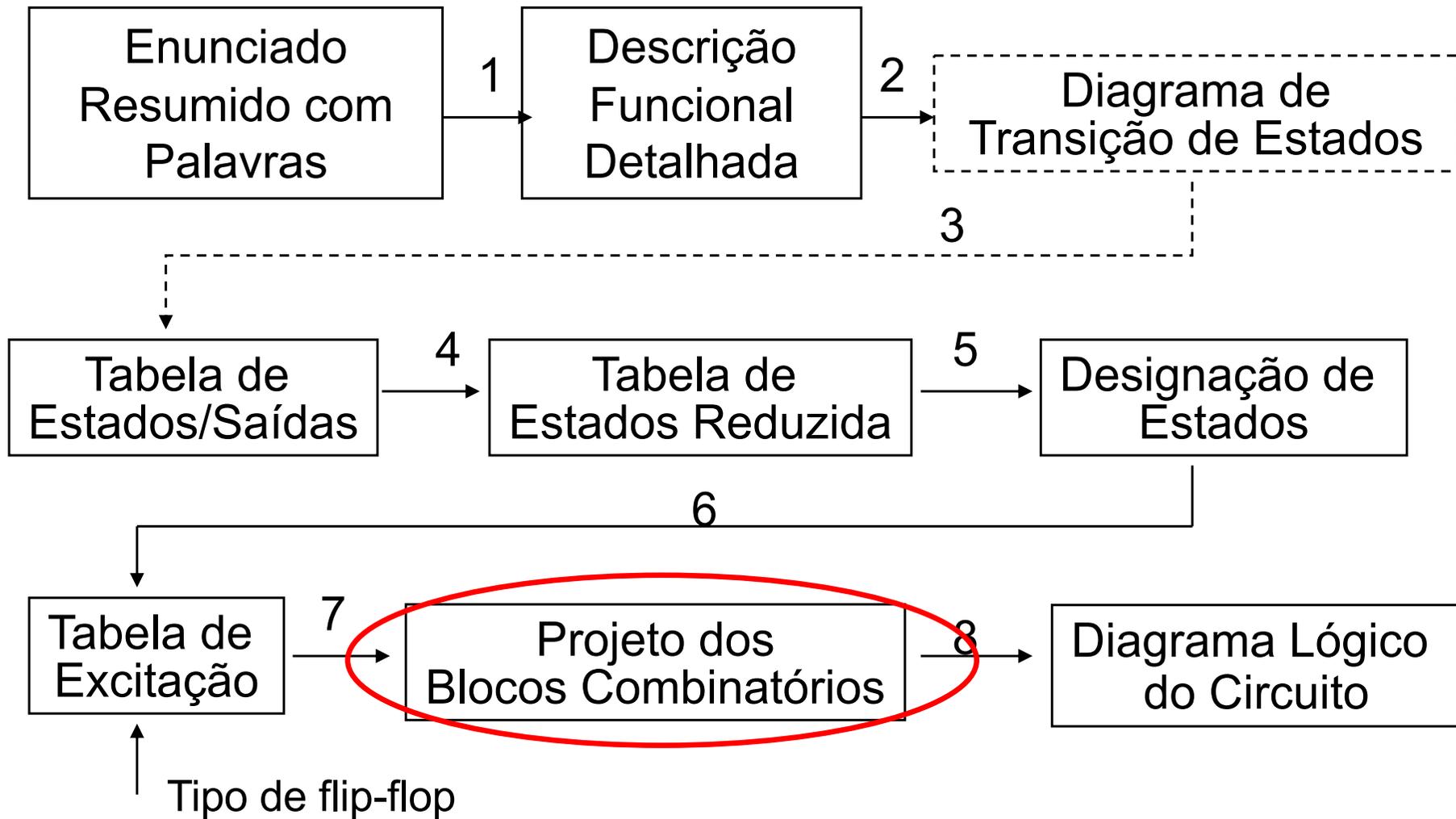
$y_1^{t+1} y_0^{t+1}$



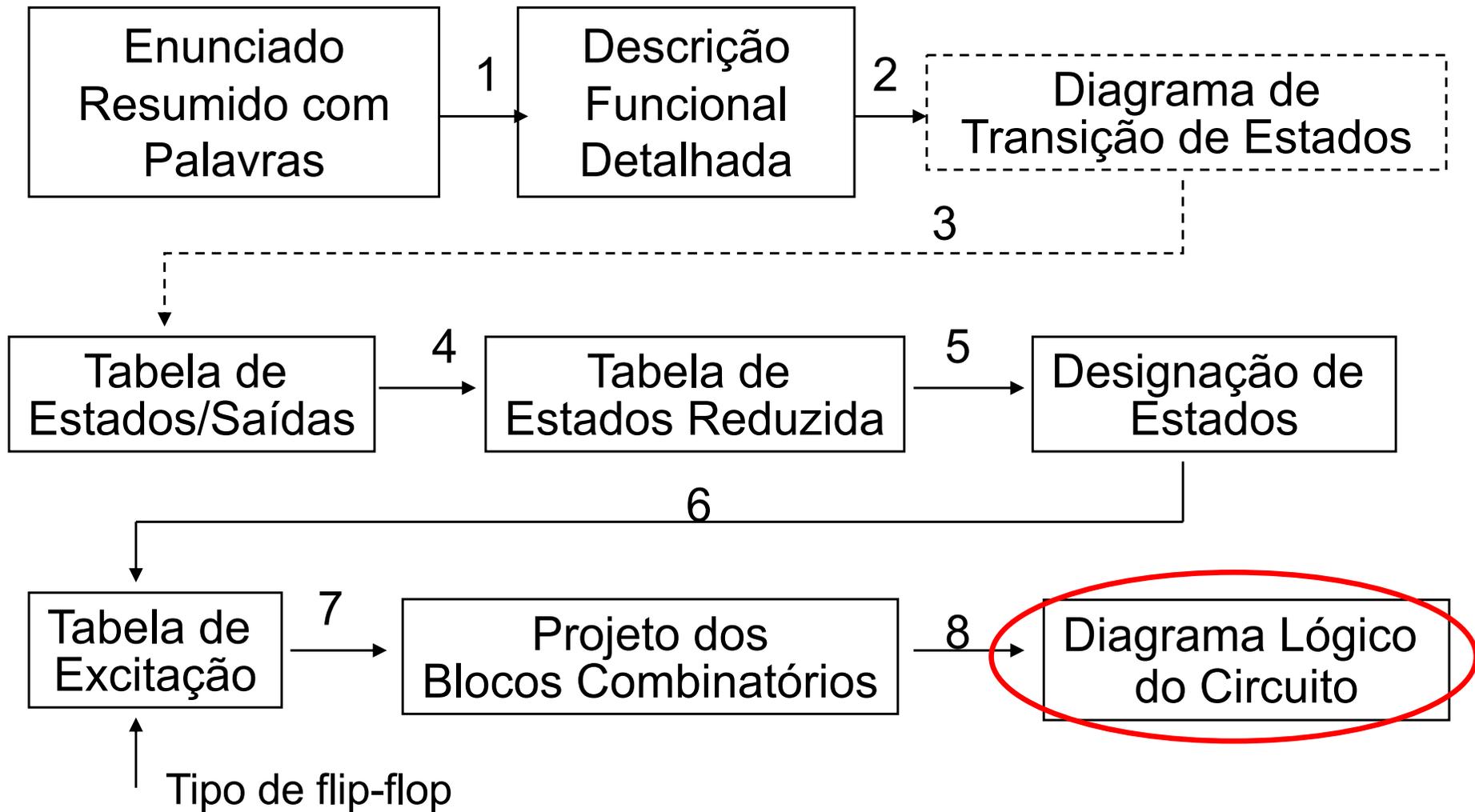
$D_1^t D_0^t$



Exemplo de Síntese - Moore

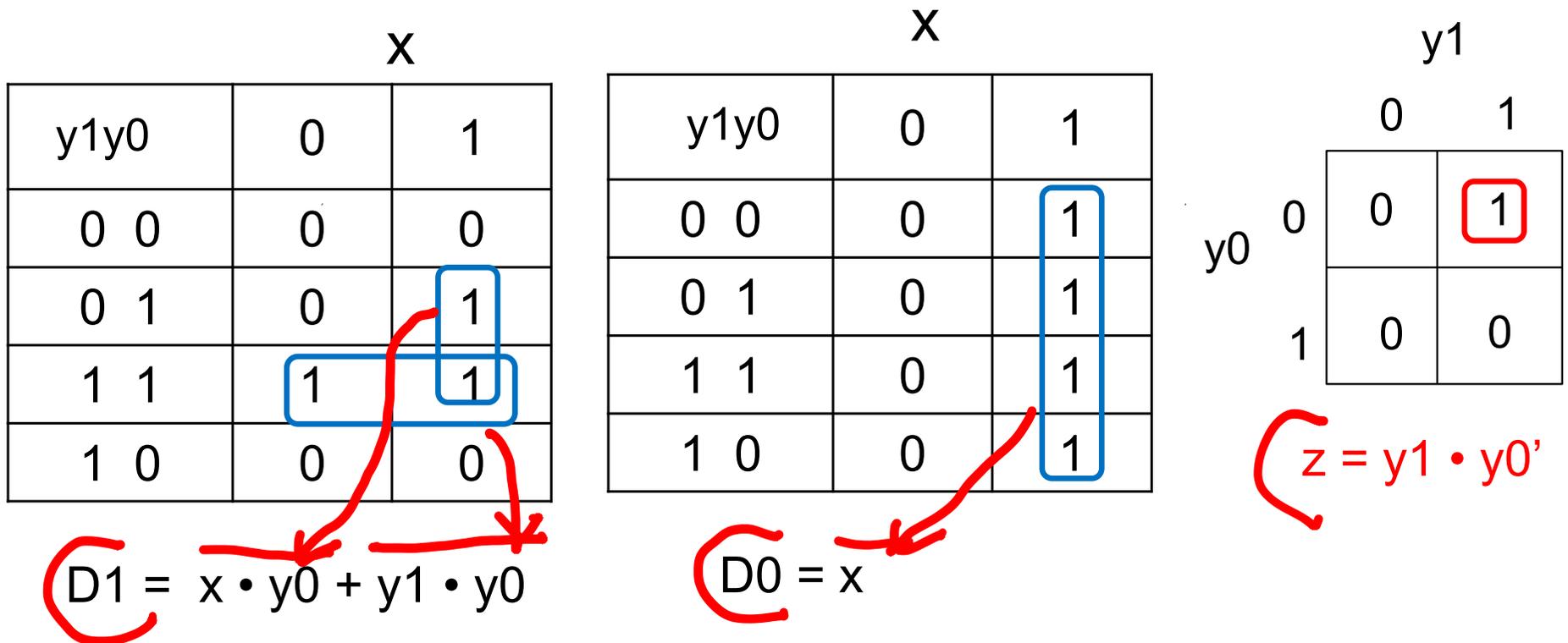


Exemplo de Síntese - Moore



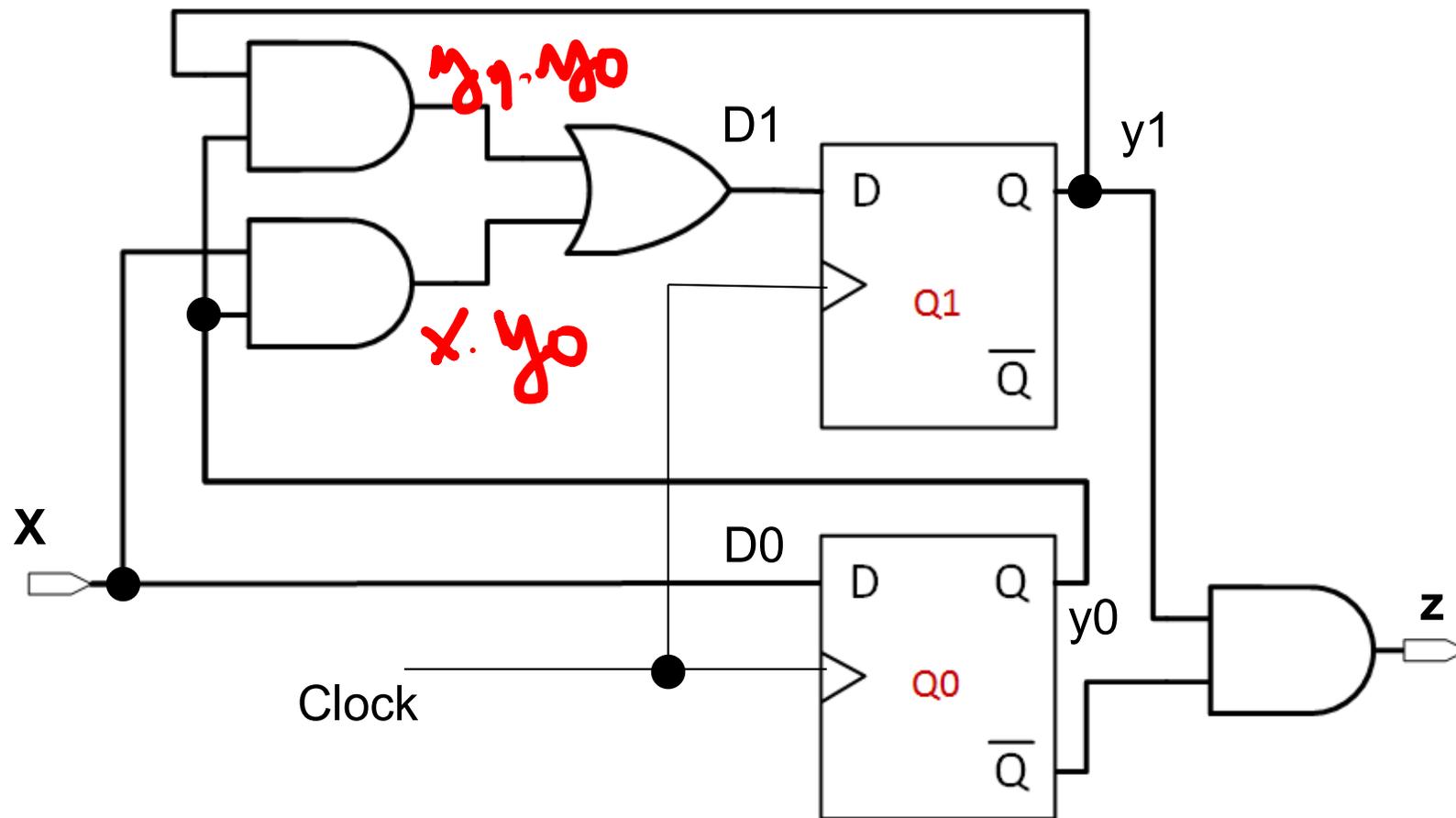
Exemplo de Síntese - Moore

Etapa 7: Projeto dos Blocos Combinatórios



Exemplo de Síntese - Moore

Etapa 8: Diagrama Lógico do Circuito



$$D1 = x \cdot y0 + y1 \cdot y0$$

$$D0 = x$$

$$z = y1 \cdot y0'$$

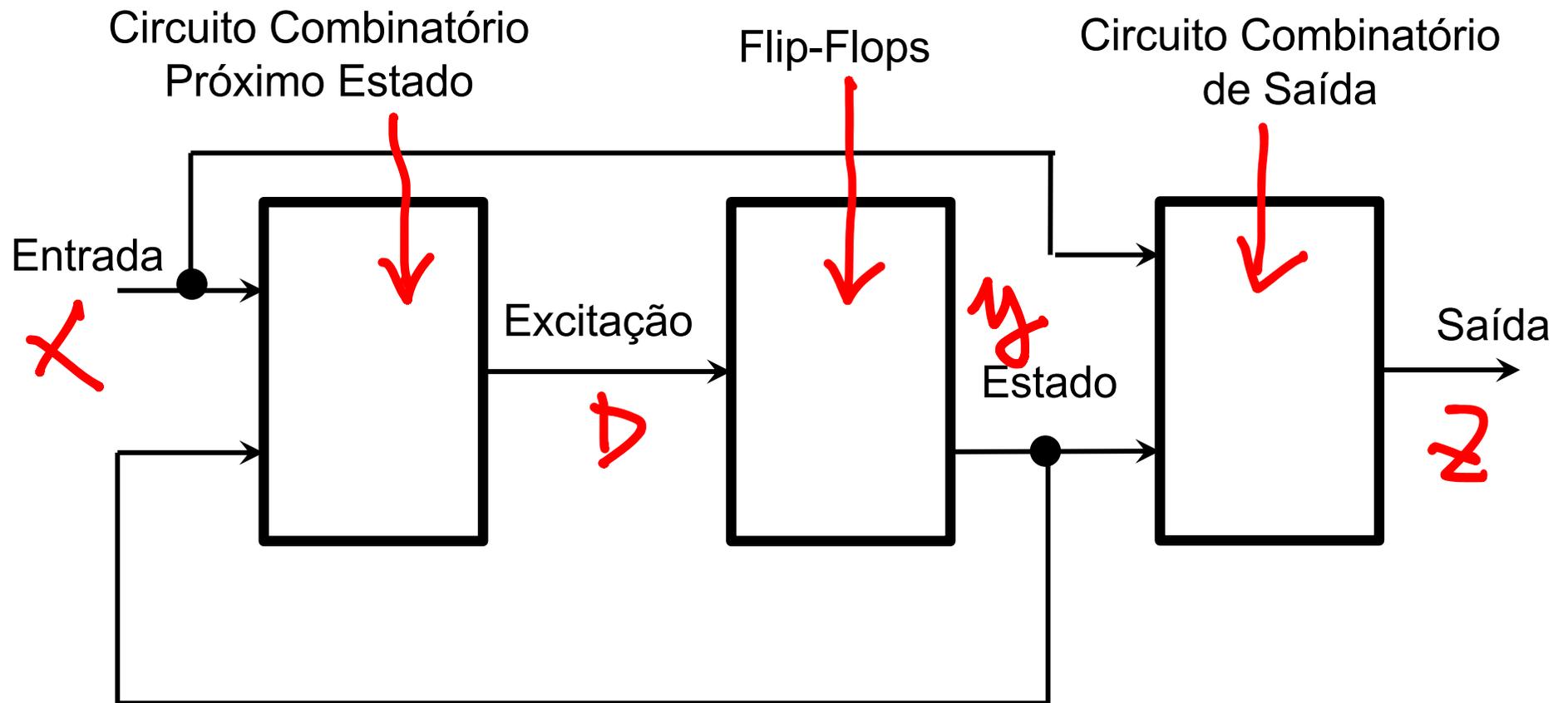
Simulação do detector (Moore)



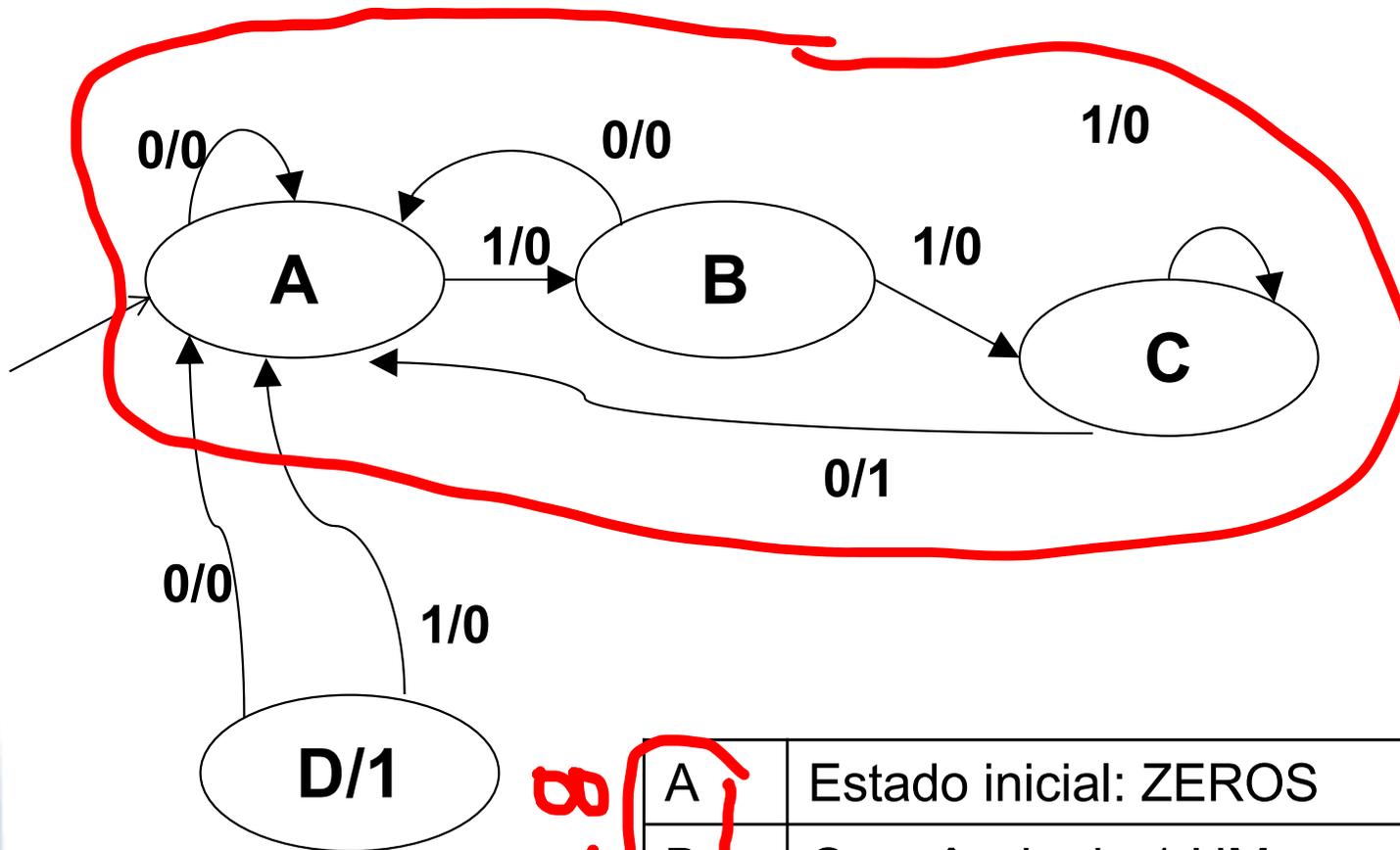
Exemplo de Síntese - Mealy

- **Enunciado:** Projetar um circuito sequencial síncrono que reconhece o primeiro ZERO após a ocorrência de dois ou mais UNS consecutivos. Adotar uma solução do tipo Mealy e flip-flops tipo D.

Modelo de Mealy



Exemplo de Síntese – Mealy Diagrama de Transição de Estados



00	A	Estado inicial: ZEROS
01	B	Ocorrência de 1 UM
11	C	Ocorrência de 2+ UNS
10	D	Estado inatingível

Exemplo de Síntese – Mealy

Tabela de Excitação

Estado $y_1^t y_0^t$	$x = 0$	$x = 1$
A 00	00 / 0	01 / 0
B 01	00 / 0	11 / 0
C 11	00 / 1	11 / 0
D 10	00 / 0	00 / 0

$y_1^{t+1} y_0^{t+1} / z^t$

Exemplo de Síntese – Mealy

Projeto dos circuitos combinatórios

		X	
y ₁ y ₀	0	1	
0 0	0	0	
0 1	0	1	
1 1	0	1	
1 0	0	0	

		X	
y ₁ y ₀	0	1	
0 0	0	1	
0 1	0	1	
1 1	0	1	
1 0	0	0	

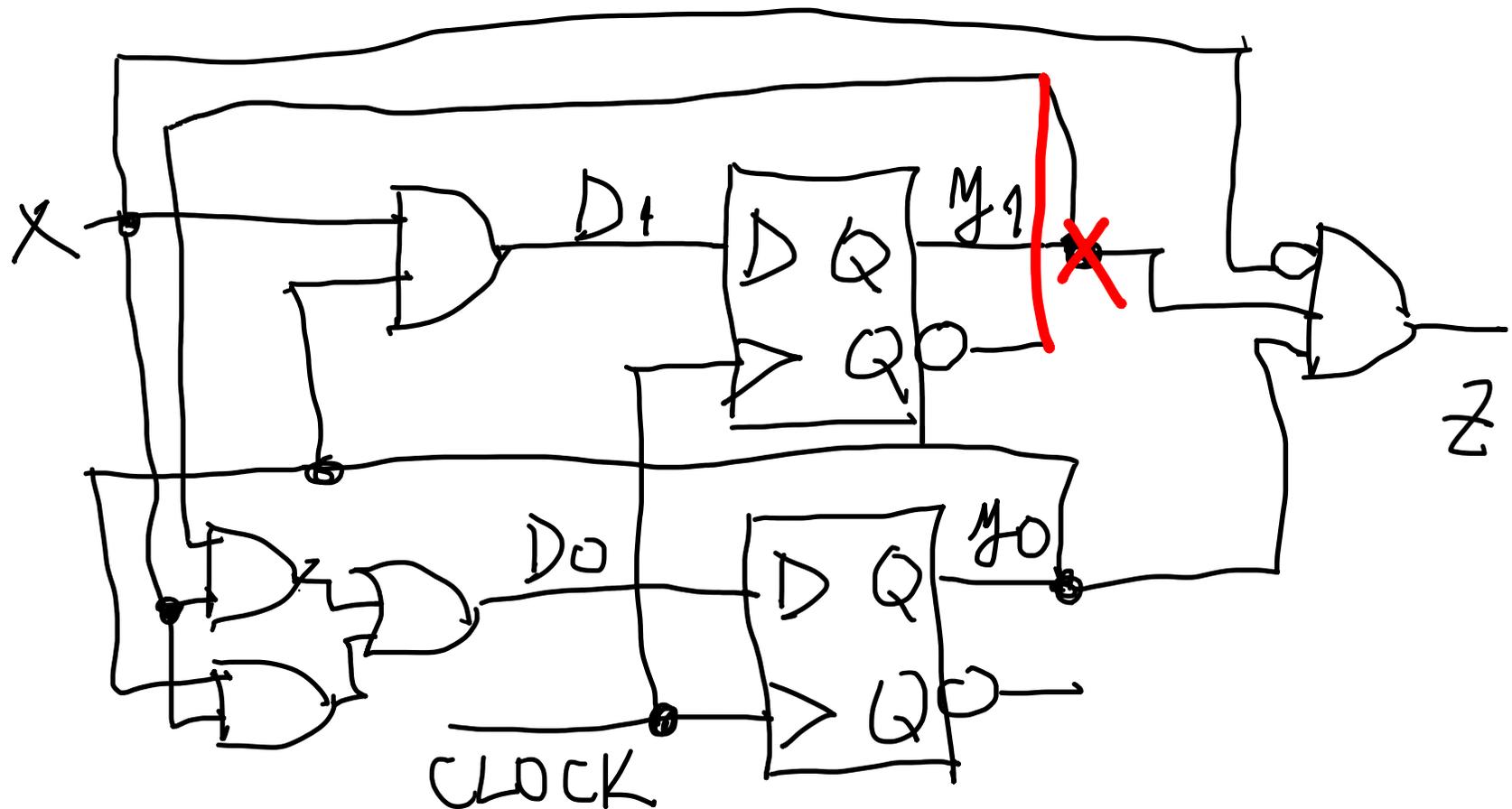
		X	
y ₁ y ₀	0	1	
0 0	0	0	
0 1	0	0	
1 1	1	0	
1 0	0	0	

$$D1 = x \cdot y0$$

$$D0 = x \cdot y1 + x \cdot y0$$

$$z = y1 \cdot y0 \cdot x'$$

Exemplo de Síntese – Mealy Diagrama Lógico do Circuito

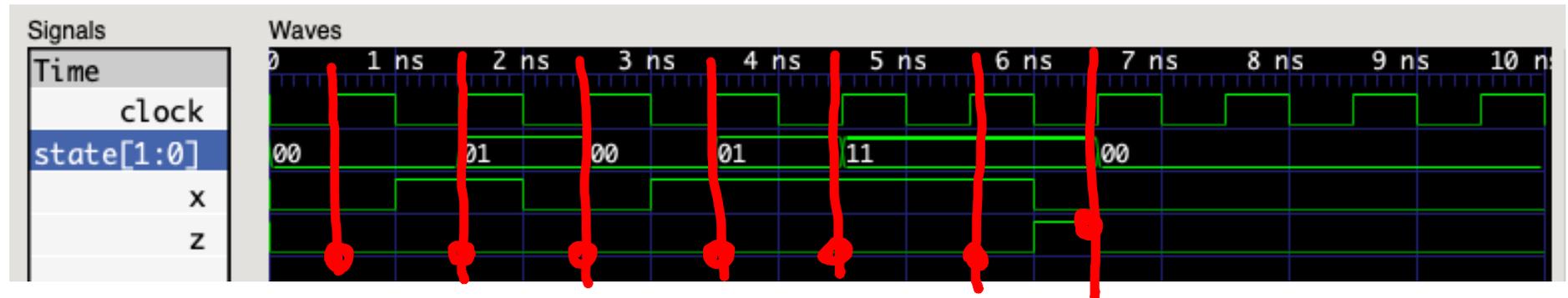


$$D1 = x \cdot y0$$

$$D0 = x \cdot y1' + x \cdot y0$$

$$z = y1 \cdot y0 \cdot x'$$

Simulação do Circuito - Mealy

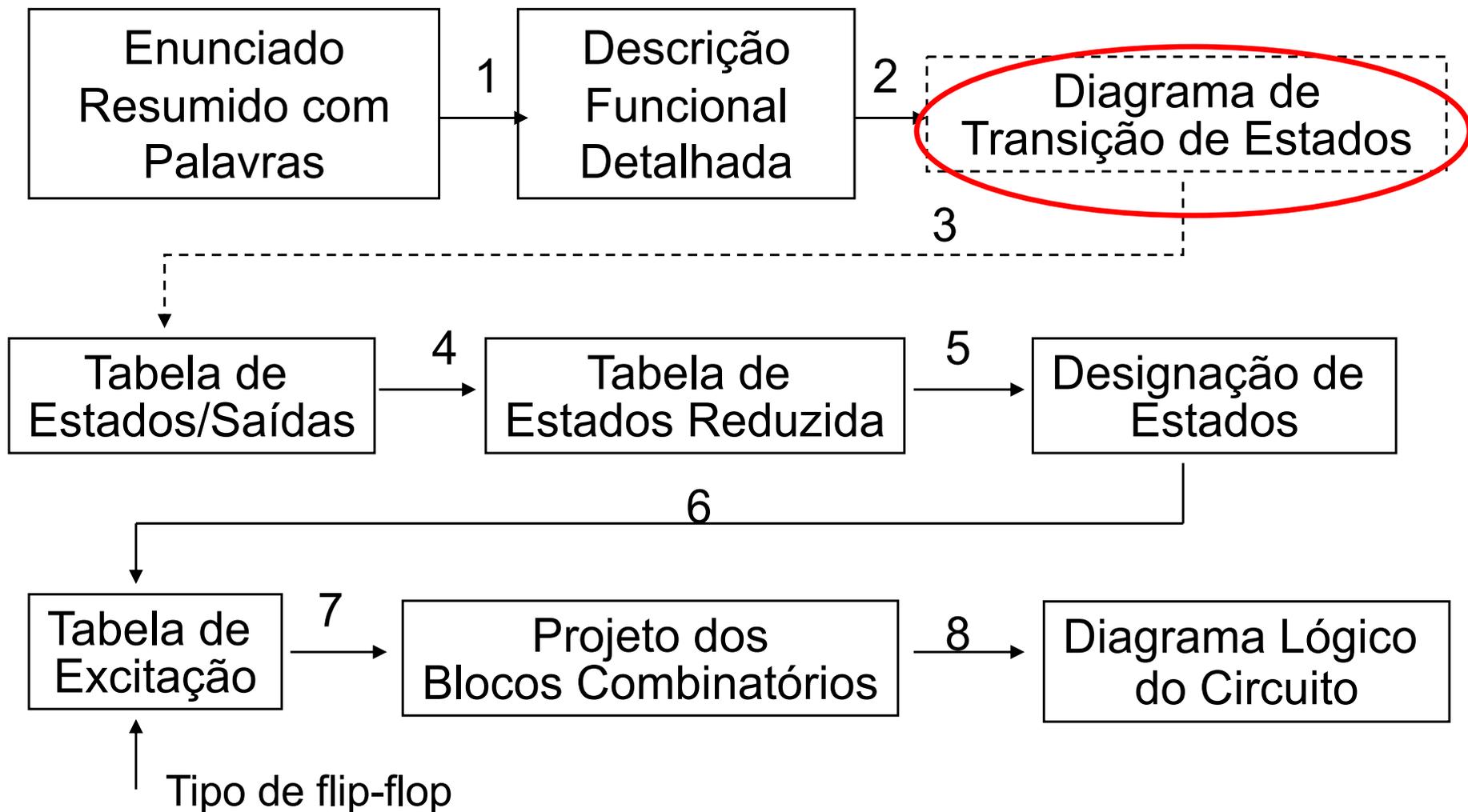


Exemplo de Síntese – Moore (2)

- **Enunciado:** Projetar um circuito sequencial síncrono que reconhece o primeiro ZERO após a ocorrência de três ou mais UNS consecutivos. Adotar uma solução do tipo Moore e flip-flops tipo D.

Entrada	1	1	0	1	0	1	1	1	1	0	0	1	1	1	0
Saída	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Exemplo de Síntese – Moore (2)



Exemplo de Síntese – Moore (2)

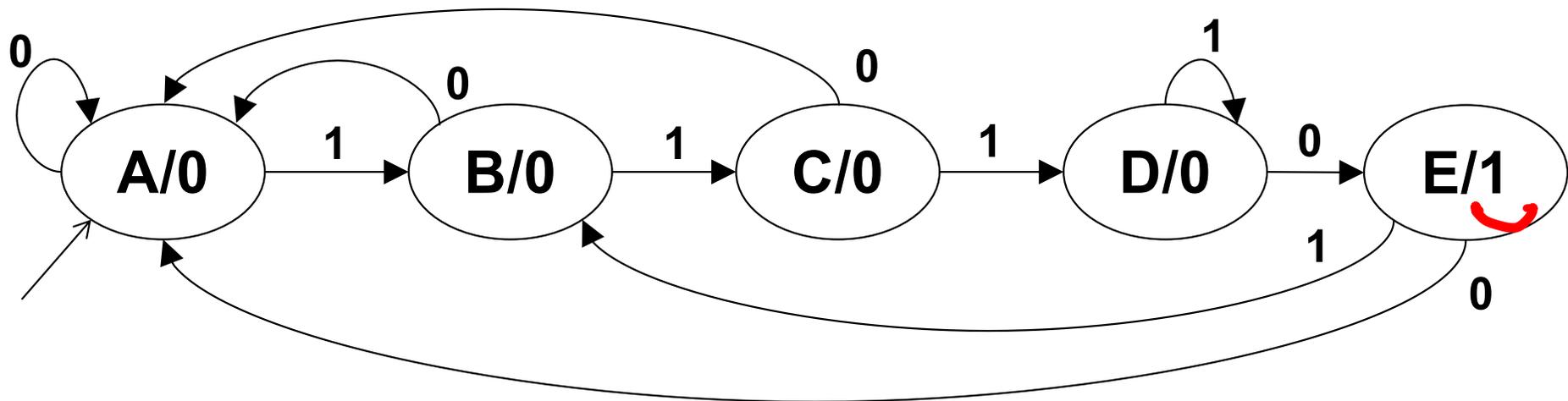
Etapa 2: Diagrama de Transição de Estados

Serão necessários pelo menos 5 estados:

- A: estado inicial
- B: armazena a ocorrência do primeiro UM
- C: armazena a ocorrência de dois UNS consecutivos
- D: armazena a ocorrência de três ou mais UNS consecutivos
- E: armazena o primeiro ZERO após três ou mais UNS consecutivos

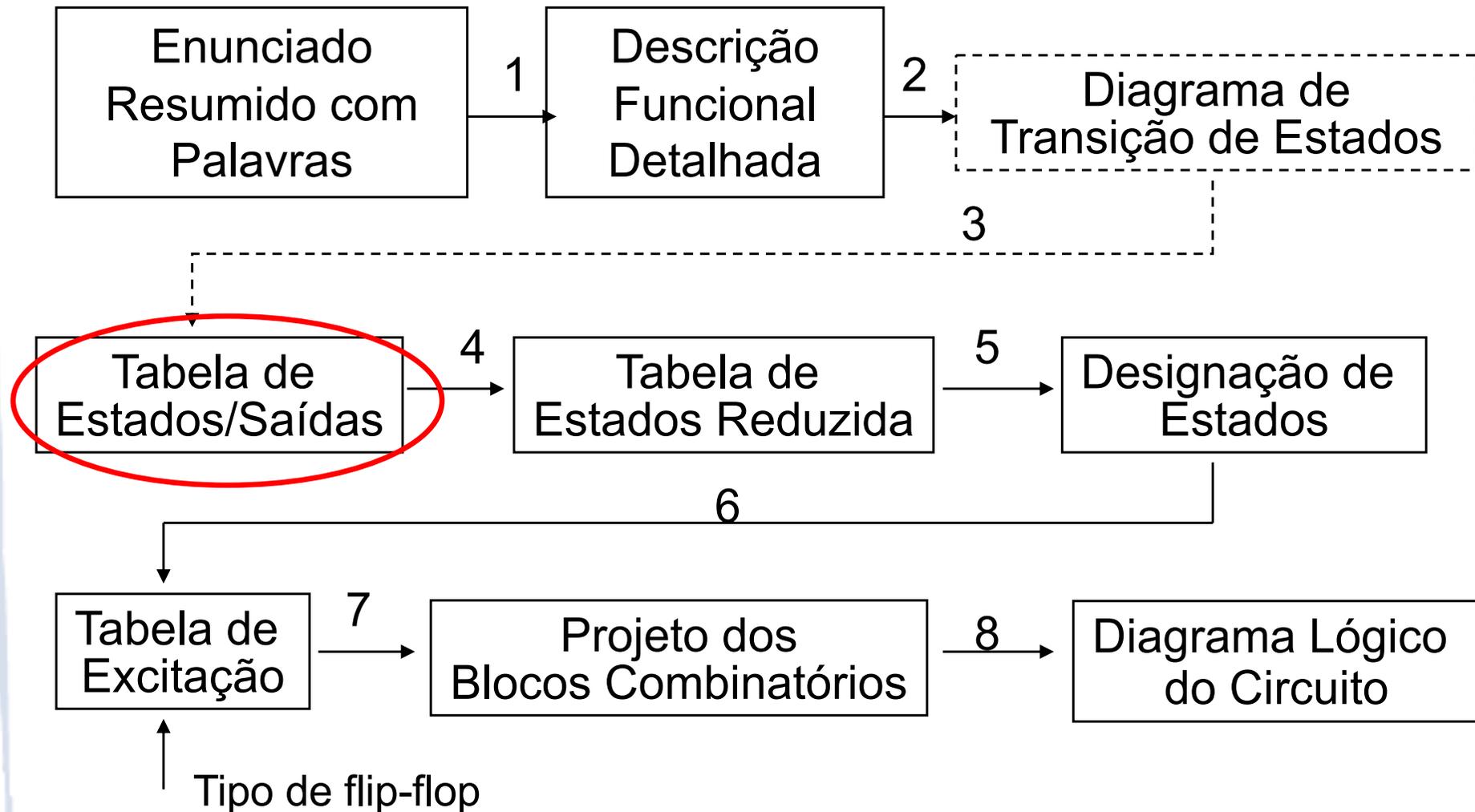
Exemplo de Síntese – Moore (2)

Etapa 2: Diagrama de Transição de Estados



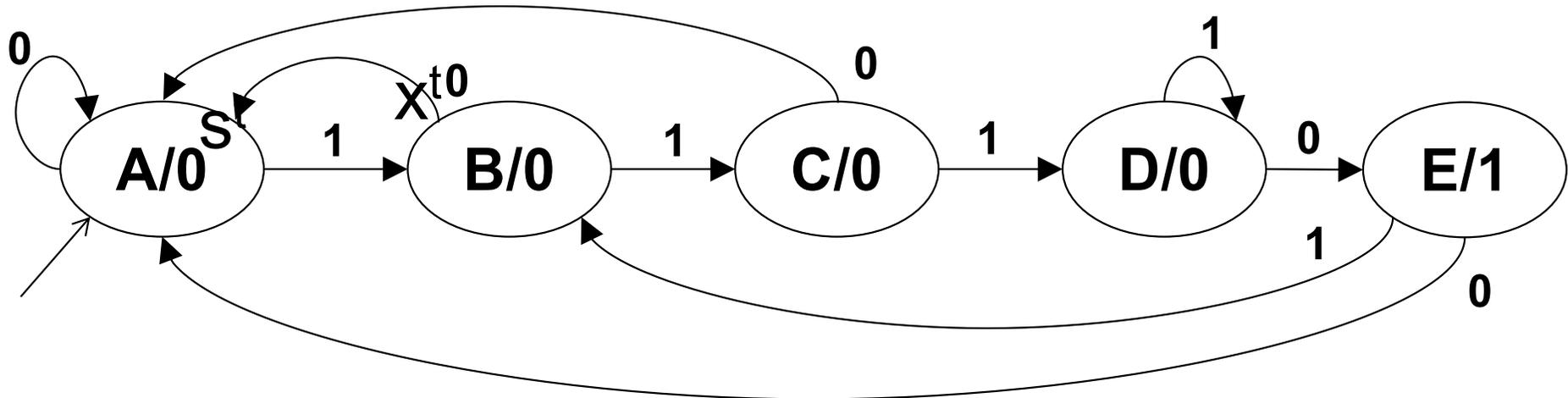
A	Estado inicial: ZEROS
B	Ocorrência de 1 UM
C	Ocorrência de 2 UNS
D	Ocorrência de 3 UNS
E	Ocorrência do 1o ZERO após 3+ UNS

Exemplo de Síntese – Moore (2)



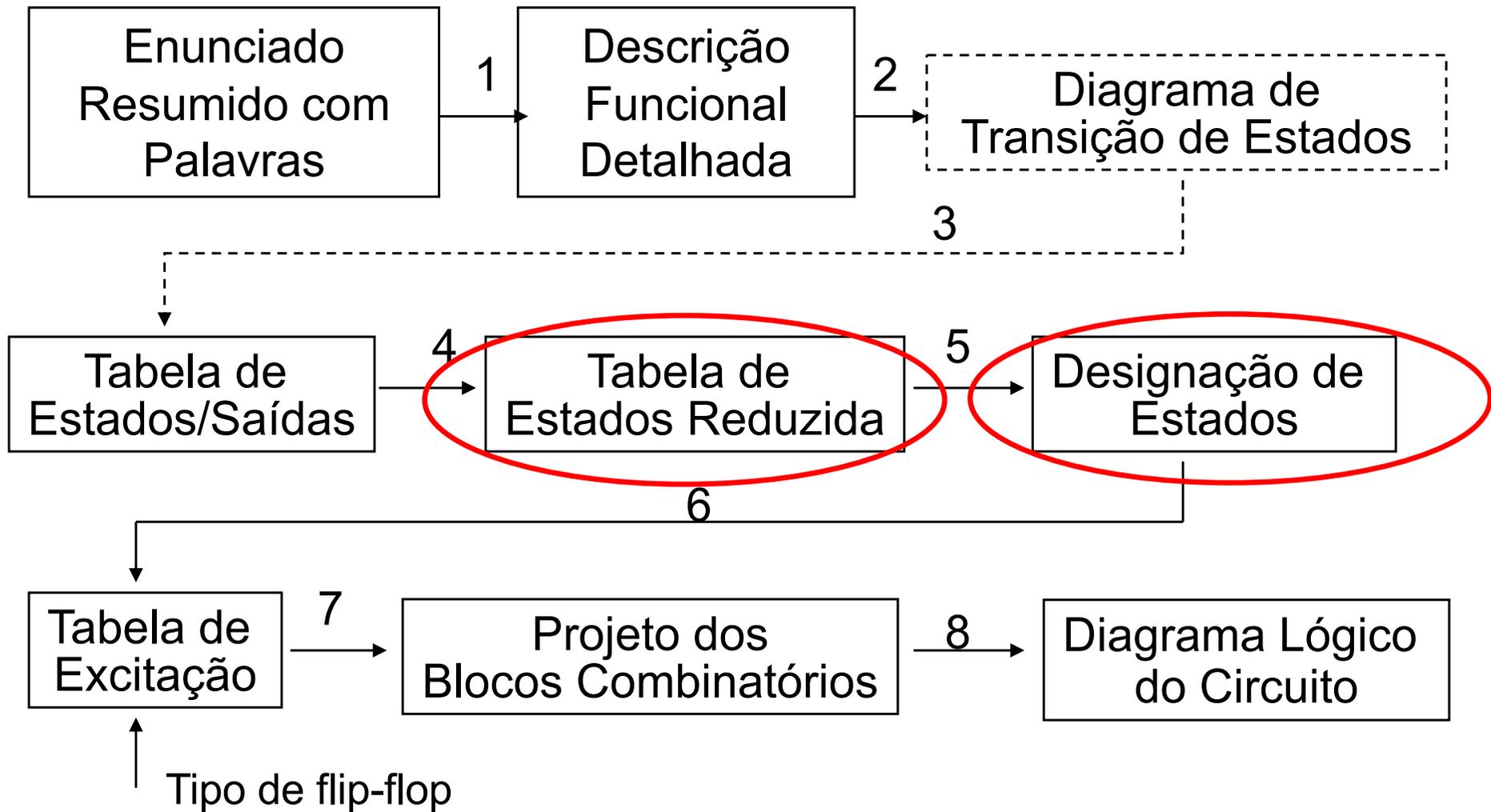
Exemplo de Síntese – Moore (2)

Etapa 3: Tabela de Entrada/Saída



	$x = 0$	$x = 1$	z^t
A	A	B	0
B	A	C	0
C	A	D	0
D	E	D	0
E	A	B	1

Exemplo de Síntese – Moore (2)



Exemplo de Síntese – Moore (2)

Etapa 4: Tabela de Estados Reduzida

Não é possível simplificar a tabela anterior

Etapa 5: Designação dos Estados

Como são necessários 5 estados, $n = 3$ e utilizaremos a seguinte designação (arbitrária):

s	y1	y2	y3	s	y1	y2	y3
A	0	0	0	E	0	0	1
B	0	1	0	F	0	1	1
C	1	1	0	G	1	1	1
D	1	0	0	H	1	0	1

Exemplo de Síntese – Moore (2)

Etapa 5: Designação dos Estados

Duas alternativas para tratar casos não especificados:

Alternativa 1: Não especificar → don't care

$s^t \backslash x^t$	$x = 0$	$x = 1$	z^t
F	*	*	*
G	*	*	*
H	*	*	*

Permite otimizações adicionais

Exemplo de Síntese – Moore (2)

Etapa 5: Designação dos Estados

Duas alternativas para tratar casos não especificados:

Alternativa 2: Impor que o estado seguinte seja o estado inicial e a saída seja 0

$s^t \backslash x^t$	$x = 0$	$x = 1$	z^t
F	A	A	0
G	A	A	0
H	A	A	0

Exemplo de Síntese – Moore (2)

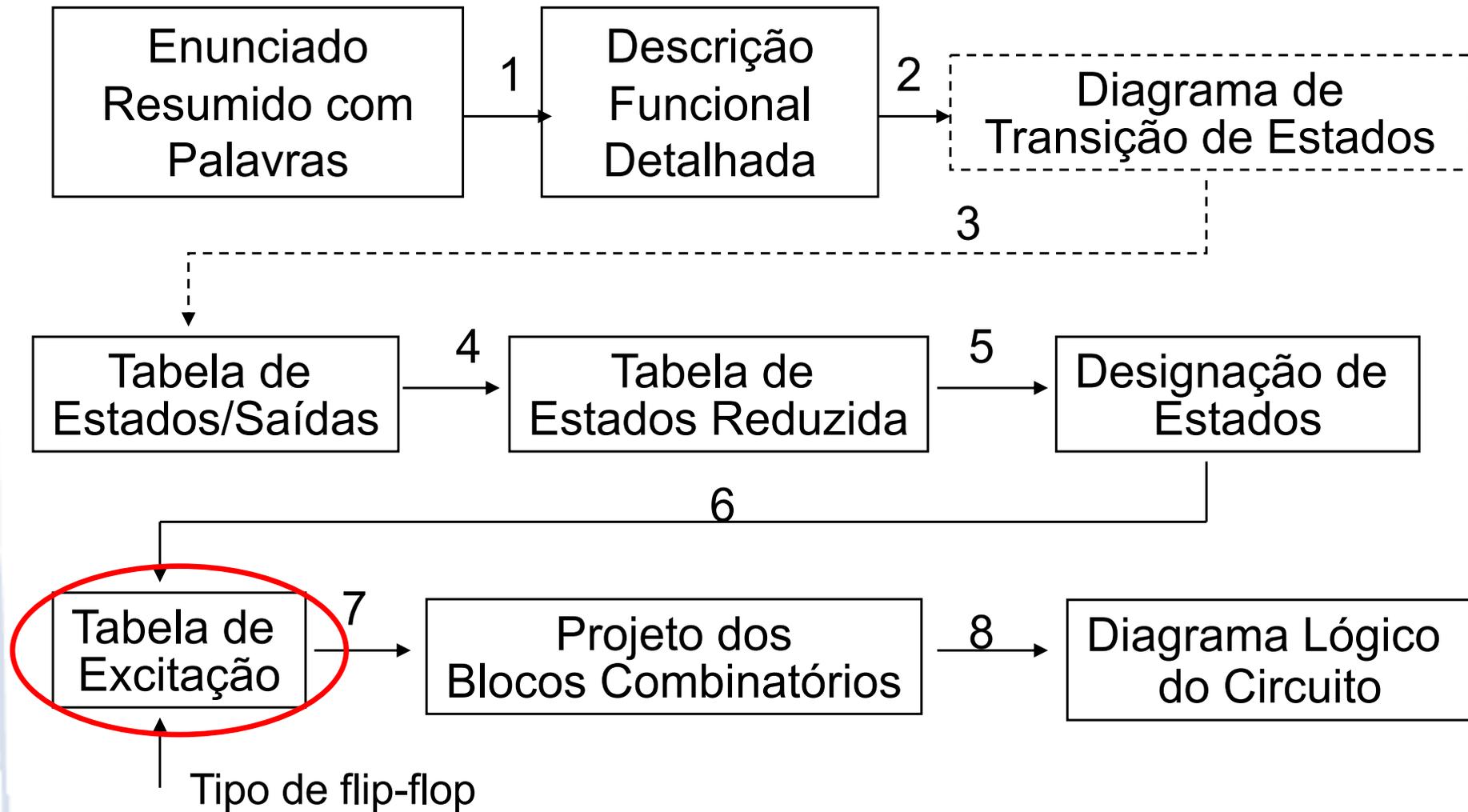
Etapa 5: Designação dos Estados

Adotando a alternativa 2: maior confiabilidade

s^t	y1 y2 y3	X = 0	X = 1	z^t
A	0 0 0	0 0 0	0 1 0	0
B	0 1 0	0 0 0	1 1 0	0
C	1 1 0	0 0 0	1 0 0	0
D	1 0 0	0 0 1	1 0 0	0
E	0 0 1	0 0 0	0 1 0	1
F	0 1 1	0 0 0	0 0 0	0
G	1 1 1	0 0 0	0 0 0	0
H	1 0 1	0 0 0	0 0 0	0

s^{t+1}

Exemplo de Síntese – Moore (2)



Exemplo de Síntese – Moore (2)

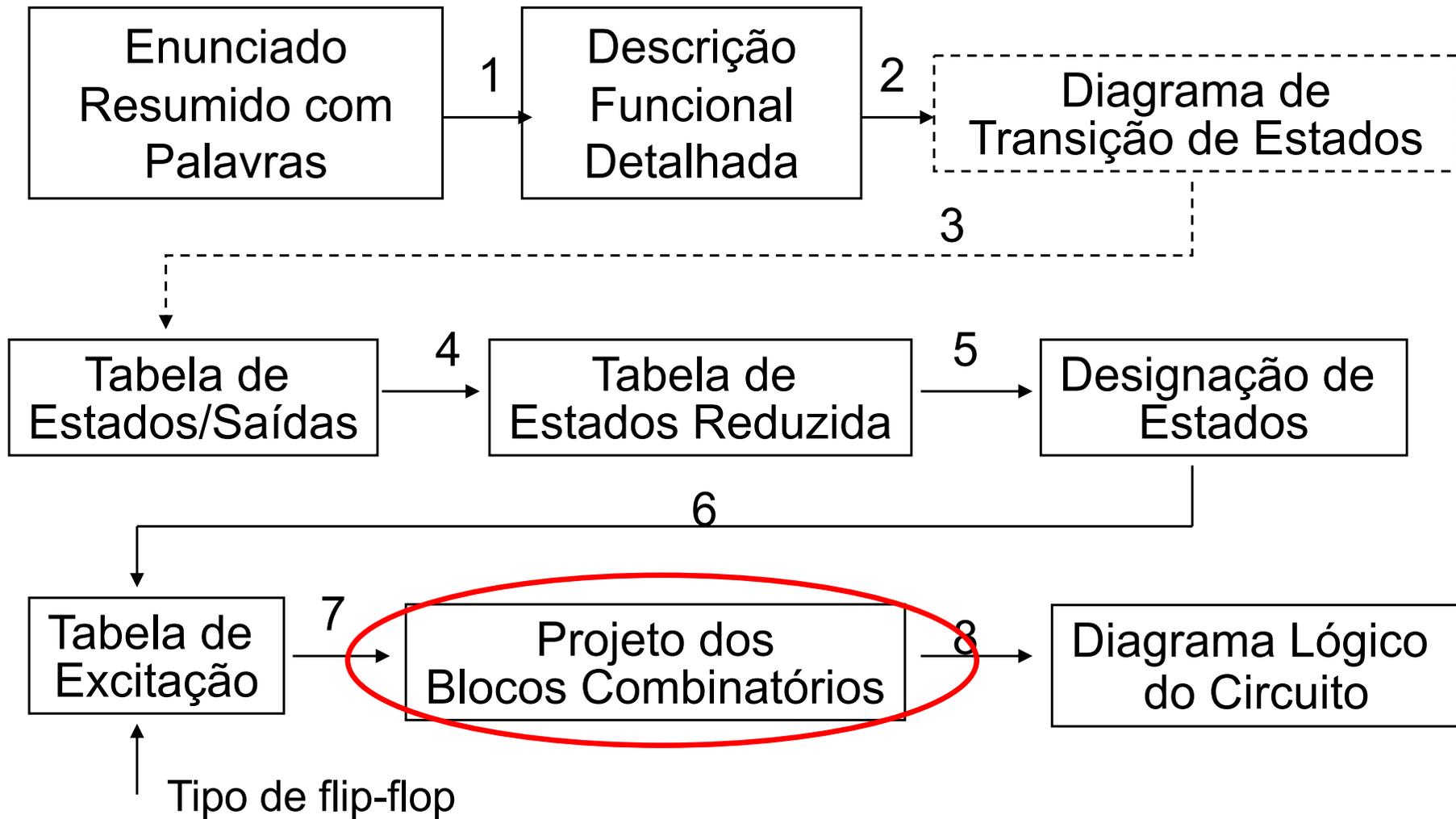
Etapa 6: Tabela de Excitação

→ Vamos usar Flip-Flops tipo D

s^t	y1	y2	y3	X = 0			X = 1			z^t	
A	0	0	0	A	0	0	B	0	1	0	0
B	0	1	0	0	0	0	1	1	0	0	0
C	1	1	0	0	0	0	1	0	0	0	0
D	1	0	0	0	0	1	1	0	0	0	0
E	0	0	1	0	0	0	0	1	0	1	1
F	0	1	1	0	0	0	0	0	0	0	0
G	1	1	1	0	0	0	0	0	0	0	0
H	1	0	1	0	0	0	0	0	0	0	0

⇒ $D1^t D2^t D3^t$

Exemplo de Síntese – Moore (2)



Exemplo de Síntese – Moore (2)

Etapa 7: Projeto dos Blocos Combinatórios

y ₁ y ₂ / xy ₃	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	0	0	1
10	0	0	0	1

D1^t

y ₁ y ₂ / xy ₃	00	01	11	10
00	0	0	1	1
01	0	0	0	1
11	0	0	0	0
10	0	0	0	0

D2^t

y ₁ y ₂ / xy ₃	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

D3^t

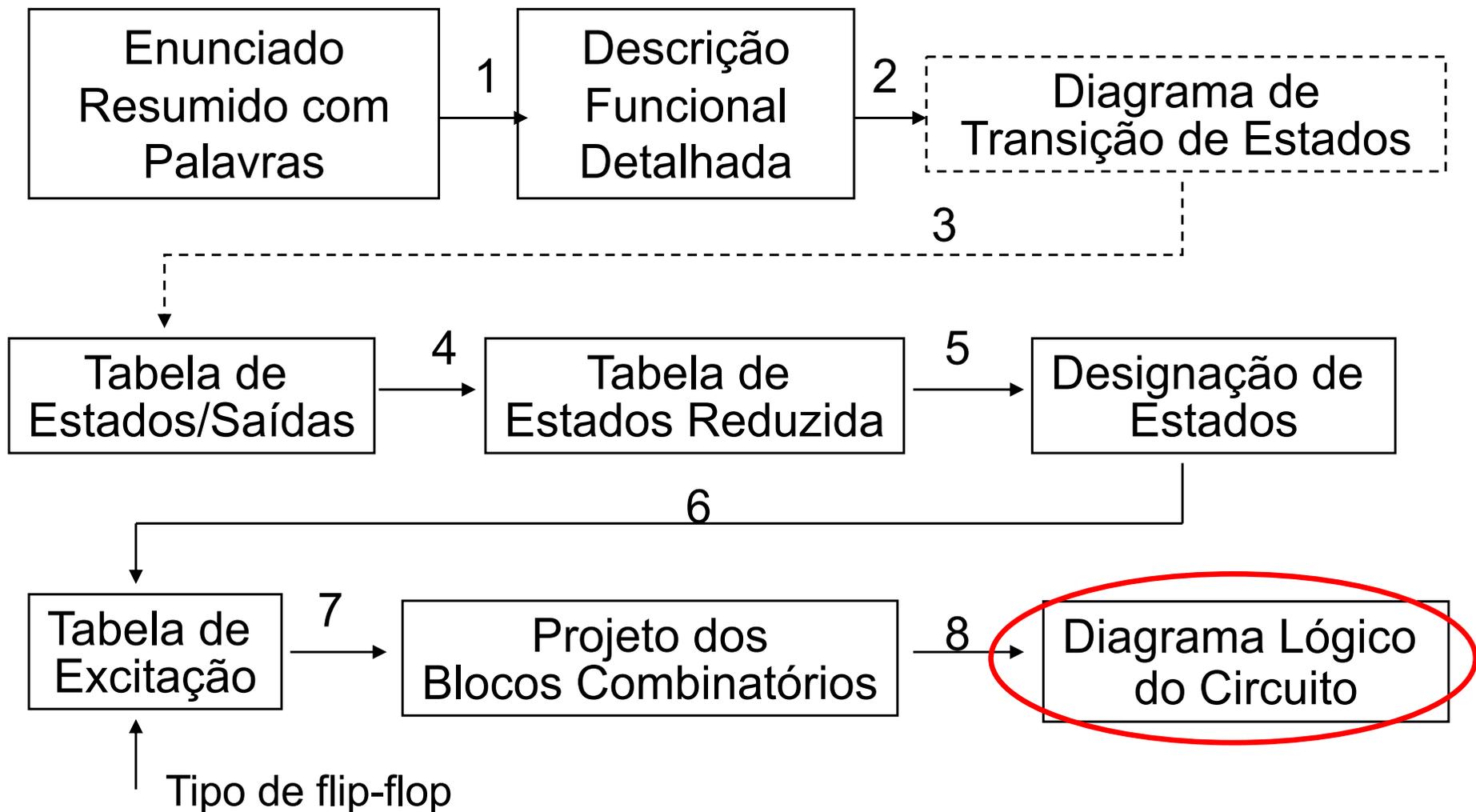
$$D1 = xy_2y_3' + x y_1y_3'$$

$$D2 = xy_1'y_2' + x y_1'y_3'$$

$$D3 = x'y_1y_2'y_3'$$

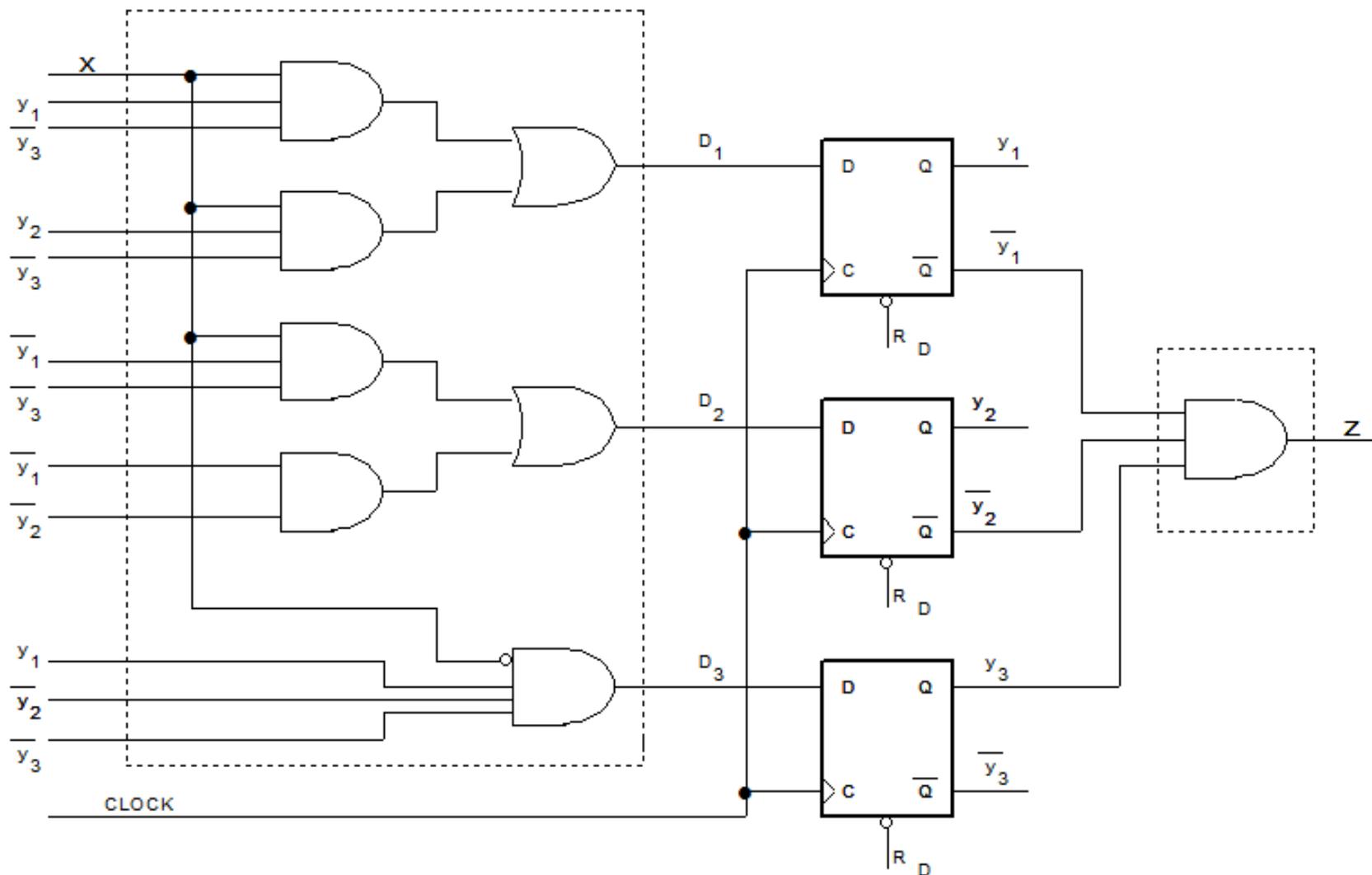
$$z = y_1'y_2'y_3$$

Exemplo de Síntese – Moore (2)



Exemplo de Síntese – Moore (2)

Etapa 8: Diagrama Lógico do Circuito



Exercício (Prova 2016)

- Projetar módulo “turbo” p/ botão B de um controle de video-game, i.e., o comando relativo a B é enviado ao console repetidamente se B for mantido pressionado.
- O circuito digital sequencial síncrono a ser projetado deve, então, ter o seguinte comportamento:
 1. Se o botão não estiver pressionado (i.e., $B = 0$), a saída enviada para o console é $Z=0$;
 2. Se B for pressionado (i.e., $B = 1$), tem-se $Z = 1$ como saída
 3. Para que o modo turbo não fique muito sensível, na segunda borda de subida do clock após o B ser pressionado a saída deve ser $Z=0$, mesmo se ele for mantido pressionado.
 4. Se, após ser pressionado, o botão for mantido pressionado por 3 ou mais bordas de subida do clock, a saída passa a ser 1, mantendo-se assim até o botão ser solto.

Exercício (Prova 2016)

a) Desenhe o diagrama de transição de estados, na forma de uma máquina de estados finitos no modelo de Moore.

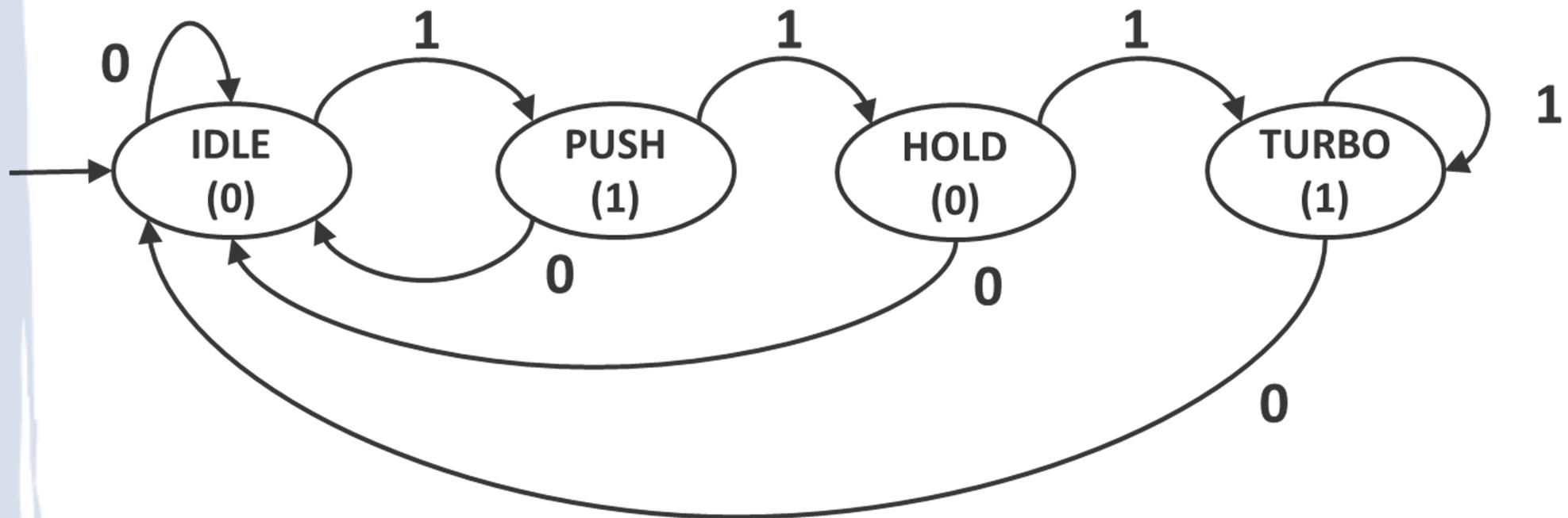
(Entrada) B: 01011011**1**011**11**011**1111**01
(Saída) Z: 01010010**1**010**11**010**1111**01

Exercício (Prova 2016)

a) Desenhe o diagrama de transição de estados, na forma de uma máquina de estados finitos no modelo de Moore.

(Entrada) B: 01011011**1**011**1**1011**1****1****1**101

(Saída) Z: 01010010**1**010**1**1010**1****1****1**101



Exercício (Prova 2016)

b) Construa a tabela com estados e saídas para o item anterior, atribuindo código binário para cada estado que o circuito pode assumir.

- Use no máximo 4 estados
- Caso o número de estados na sua solução não seja uma potência de 2, represente os estados não-atingíveis e adote a estratégia "minimalista": use "don't care" para as transições e saídas desses estados.

Exercício (Prova 2016)

b) Construa a tabela com estados e saídas para o item anterior, atribuindo código binário para cada estado que o circuito pode assumir.

- Use no máximo 4 estados
- Caso o número de estados na sua solução não seja uma potência de 2, represente os estados não-atingíveis e adote a estratégia "minimalista": use "don't care" para as transições e saídas desses estados.

		Entrada: B		Saída
Estado	Q1 Q0	0	1	Z
IDLE	0 0	0 0	0 1	0
PUSH	0 1	0 0	1 1	1
HOLD	1 1	0 0	1 0	0
TURBO	1 0	0 0	1 0	1

Exercício (Prova 2016)

- b) Monte o circuito usando Flip-Flops tipo D
- Minimize a lógica combinatória utilizada.
 - Não é preciso desenhar o circuito: apenas mostre as equações de excitação para cada Flip-Flop e saída.

Exercício (Prova 2016)

b) Monte o circuito usando Flip-Flops tipo D

- Minimize a lógica combinatória utilizada.
- Não é preciso desenhar o circuito: apenas mostre as equações de excitação para cada Flip-Flop e saída.

		Entrada: B		Saída
Estado	Q1 Q0	0	1	Z
IDLE	0 0	0 0	0 1	0
PUSH	0 1	0 0	1 1	1
HOLD	1 1	0 0	1 0	0
TURBO	1 0	0 0	1 0	1

$$D1 = B \cdot Q0 + B \cdot Q1 = B \cdot (Q0 + Q1)$$

$$D0 = B \cdot Q1'$$

$$Z = Q1' \cdot Q0 + Q1 \cdot Q0' = Q1 \oplus Q0$$

Exercício (Prova 2016-2)

- Projetar módulo de controle de reação química, para evitar temperaturas excessivas. Especificação:
 - Entrada (bit): X. 1 se temperatura excessiva for detectada.
 - Saída (bits): T e R, comandos para ativar, respectivamente, redução de temperatura e adição de catalisador à reação.
 - Operação: T deve ser ativado após dois 1s consecutivos na entrada X, voltando ao estado inicial sem ativar R caso isso seja suficiente para obter $X = 0$. Caso contrário, entra-se em um modo de "emergência" em que T e R mantêm-se ambos em 1 enquanto X também for 1. Caso o sistema chegue nesse estado, ele só voltará à normalidade após três leituras consecutivas de $X = 0$. Enquanto isso não ocorrer, qualquer leitura de $X = 0$ faz com que R volte a 0, mas não com que T volte a 0; além disso, qualquer leitura de $X=1$ nesse período também reativa R além de manter ativado T.

Exercício (Prova 2016-2)

- Exemplo:
 - T ativado após dois 1s consecutivos na entrada X
 - Volta a estado inicial sem ativar R caso isso leve a $X = 0$; senão, entra-se em um modo de "emergência", em que T e R mantêm-se em 1 enquanto X também for 1.
 - Se em estado de emergência, volta à normalidade após três leituras consecutivas de $X = 0$. Enquanto isso não ocorrer, qualquer leitura de $X = 0$ faz com que R volte a 0, mas não com que T volte a 0; além disso, qualquer leitura de $X=1$ nesse período também reativa R além de manter ativado T.
 - Notação: sublinhado = "emergência"; senão = "modo normal":

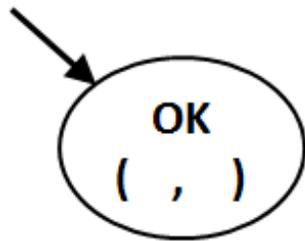
X: 0010110101111101001010001011011111101100010

T: 0000010000111111111111000010011111111111000

R: 0000000000011010010100000000000111101100000

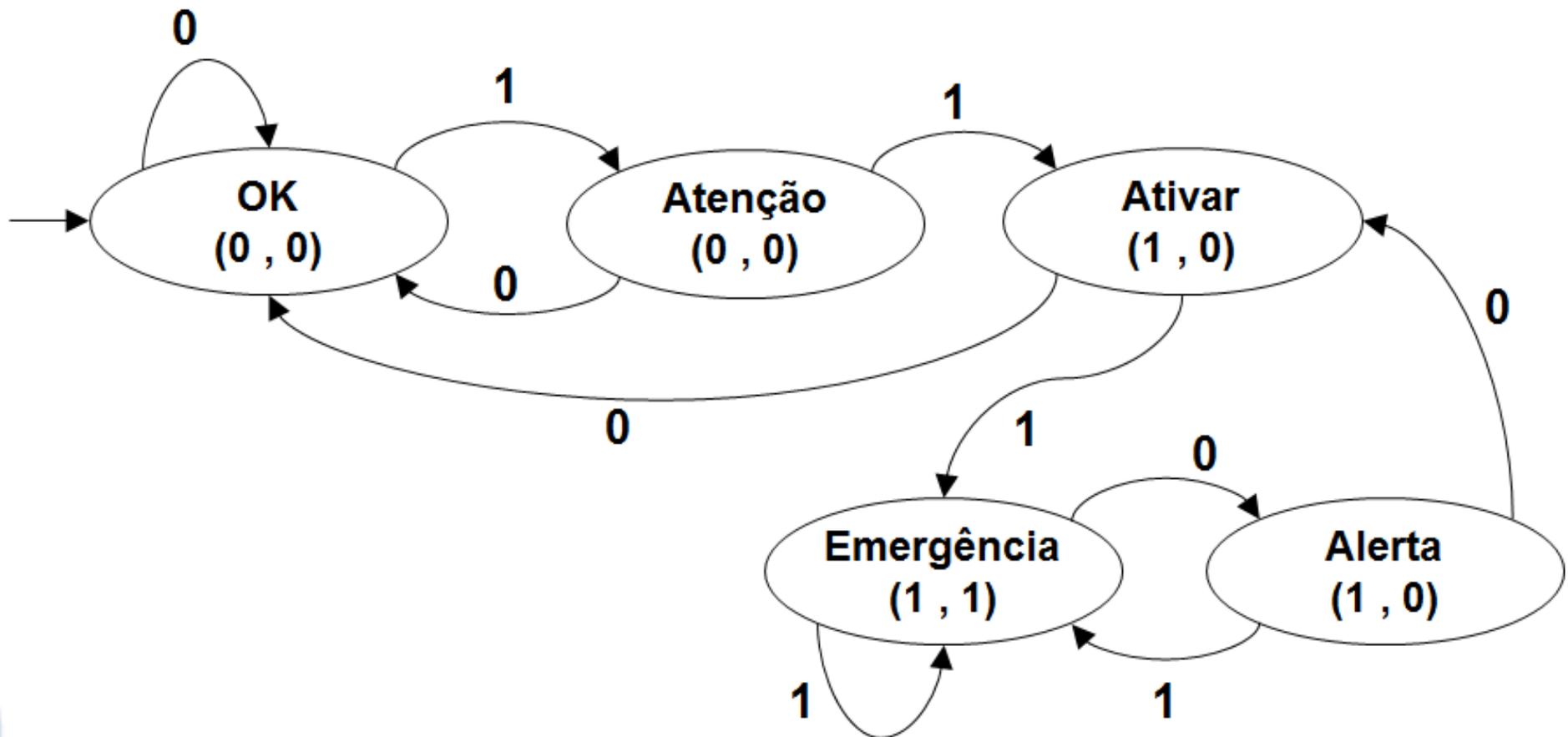
Exercício (Prova 2016-2)

a) Desenhe o diagrama de transição de estados (máquina de estados finitos) que apresente o comportamento pedido, usando o **modelo de Moore**.



Exercício (Prova 2016-2)

a) Desenhe o diagrama de transição de estados (máquina de estados finitos) que apresente o comportamento pedido, usando o **modelo de Moore**.



Exercício (Prova 2016-2)

b) Construa tabela com os estados e saídas referentes ao diagrama de transição apresentado no item a. Atribua um código binário a cada estado usando no máximo 3 bits, de forma que tenha-se como lógica de saída **$T=Q1$ e $R=Q0$** , ignorando-se estados não-atingíveis (caso haja).

Estado		Próximo Estado		Saídas	
Nome	Q2Q1Q0	X = 0	X = 1	T	R
OK	000				

Exercício (Prova 2016-2)

b) Construa tabela com os estados e saídas referentes ao diagrama de transição apresentado no item a. Atribua um código binário a cada estado usando no máximo 3 bits, de forma que tenha-se como lógica de saída **$T=Q1$ e $R=Q0$** , ignorando-se estados não-atingíveis (caso haja).

Estado		Próximo Estado		Saídas	
Nome	Q2Q1Q0	X = 0	X = 1	T	R
OK	000	000	100	0	0
Atenção	100	000	010	0	0
Ativar	010	000	011	1	0
Emergência	011	110	011	1	1
Alerta	110	010	011	1	0
Erro1	001	***	***	*	*
Erro2	101	***	***	*	*
Erro3	111	***	***	*	*

Exercício (Prova 2016-2)

c) Especifique o circuito usando Flip-Flops tipo D. Caso o número de estados na sua solução não seja uma potência de 2, utilize a estratégia "otimizada" para os estados não-atingíveis do sistema, i.e., usando a estratégia de "don't care" para transições e saídas. **Não desenhe o circuito**, mas apresente as equações de excitação dos Flip-Flops.

Q0 X Q2Q1	00	01	11	10
0 0				
0 1				
1 1				
1 0				

Exercício (Prova 2016-2)

c) Especifique o circuito usando Flip-Flops tipo D. Caso o número de estados na sua solução não seja uma potência de 2, utilize a estratégia "otimizada" para os estados não-atingíveis do sistema.

Q0 X Q2Q1	00	01	11	10
0 0	000	100	***	***
0 1	000	011	011	110
1 1	010	011	***	***
1 0	000	010	***	***

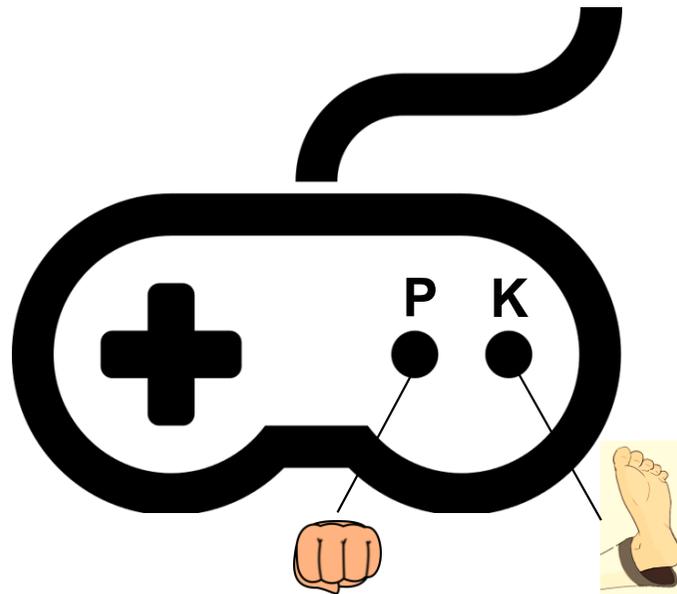
$$D2 = Q2' \cdot Q1' \cdot X + Q0 \cdot X'$$

$$D1 = Q2 \cdot Q1 + Q0 + Q1 \cdot X + Q2 \cdot X$$

$$D0 = Q1 \cdot X$$

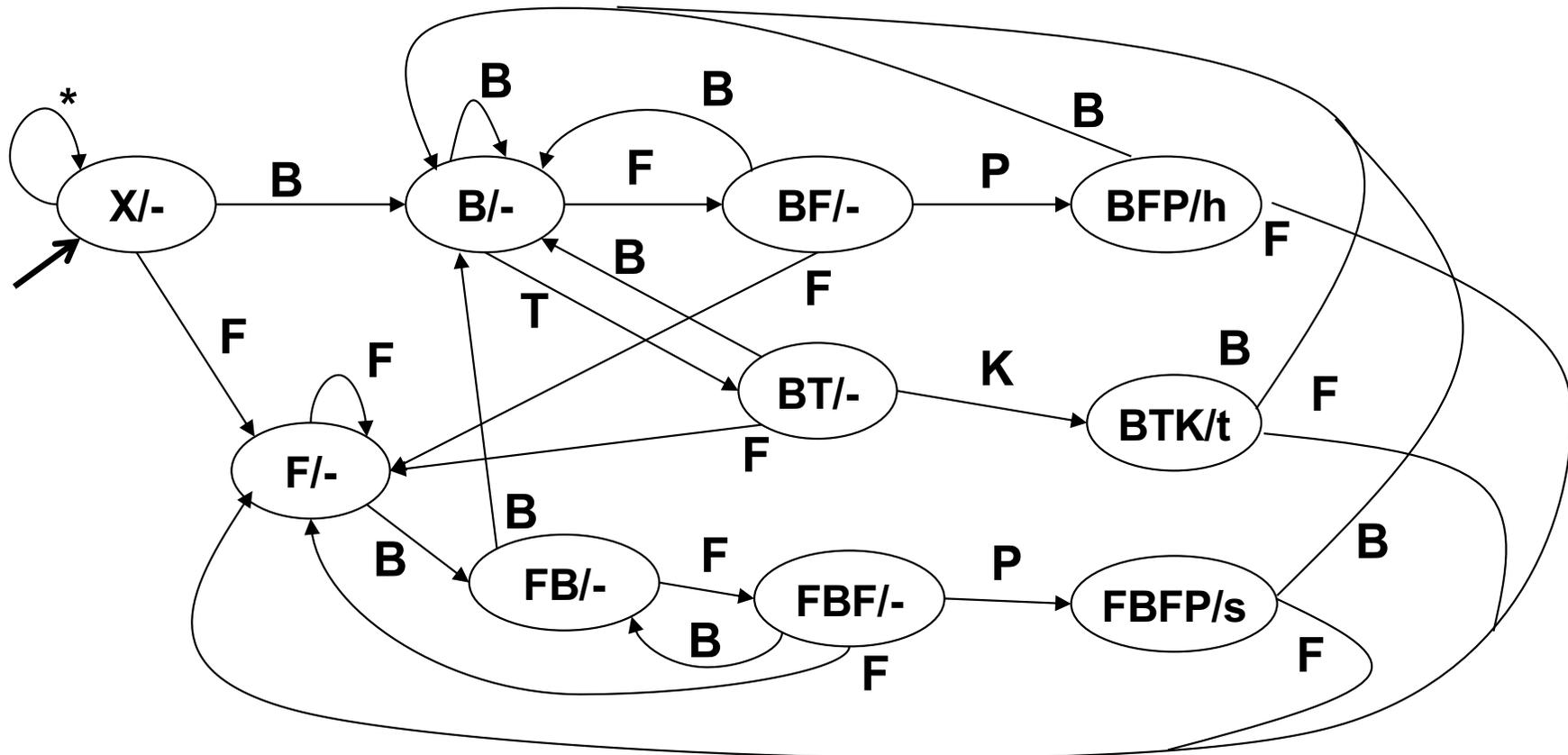
Exercício

- Detector de golpes do Street Fighter – Ryu
- Escolha 1 dos seguintes (ou tente fazer todos)
 - Hadouken: ↓ → P
 - Shoryuken: → ↓ → P
 - Tatsumaki Senpukyaku: ↓ ← K



Exercício – uma tentativa

- (obs.: não foi testado... pode haver erros!)
 - Entradas direcionais: \downarrow : B; \rightarrow : F; \leftarrow : T; \uparrow : C
 - Saídas: **h**: Hadouken; **s**: Shoryuken; **t**: Tatsumaki Senpukyaku
 - Transições não mostradas: levam a estado inicial



Exercício: reflexão

- Alguns projetos podem ficar muito complexos!
- No exercício:
 - 10 estados → 4 bits
 - 6 botões → 3 bits (se apenas 1 pressionado por vez)
 - 6 bits se qualquer combinação de botões for aceita
- Projeto do bloco de próximo estado usando abordagem tradicional requer minimização com 7 a 10 bits...
 - Não é muito viável...
- Como gerenciar complexidade?
 - Modelagem modularizada (e.g., 3 máquinas distintas)
 - Uso de linguagem de descrição de hardware: **HDL!**