

TÉCNICAS DIGITAIS II

- *Código:* **FEN 538.5**
- *Carga Horária:* 75 horas
- *Número de Créditos:* 4
- *Professora:* Nadia Nedjah

<i>Tempo</i>		<i>Segunda-Feira</i>
15:10-16:00	T4	<u>Teoria</u> Turmas 5 e 6
16:10-17:00	T5	
17:00-17:50	T6	

EMENTA E BIBLIOGRAFIA

Ementa

- *Introdução*
- *Multi-Vibradores*
- *Máquinas de Estados*
- *Memórias*
- *Controladores Micro-Programados*

Bibliografia

- *John P. Uyemura, Sistemas Digitais: Uma Abordagem Integrada, Thomson Pioneira, 2000.*
- *Milos Ercegovic, Tomas Lang, Jaime H. Moreno, Introdução aos Sistemas Digitais, Bookman, 1999.*

TRABALHOS PRÁTICOS

Projetos

- a) Latch*
- b) Gerador de onda quadrada;*
- c) Contador up/down;*
- d) Contador de módulo programável;*
- e) Dado eletrônico;*
- f) Conversor paralelo/série com registros de deslocamento.*

AVALIAÇÃO

Avaliação

1. *Provas parciais escritas* (2)
2. *Frequência* (75%)

3. *Média parcial computada:*

$$M_p = (2(N_1 + N_2) + (L_1 + L_2 + L_3 + L_4)/4)/5$$

4. *Se for necessário, prova final escrita* (1)

5. *Média final computada:*

$$M_p \geq 7.0 \Rightarrow M_f = M_p$$

$$4.0 \leq M_p < 7.0 \Rightarrow M_f = (M_p + N_f)/2$$

INTRODUÇÃO

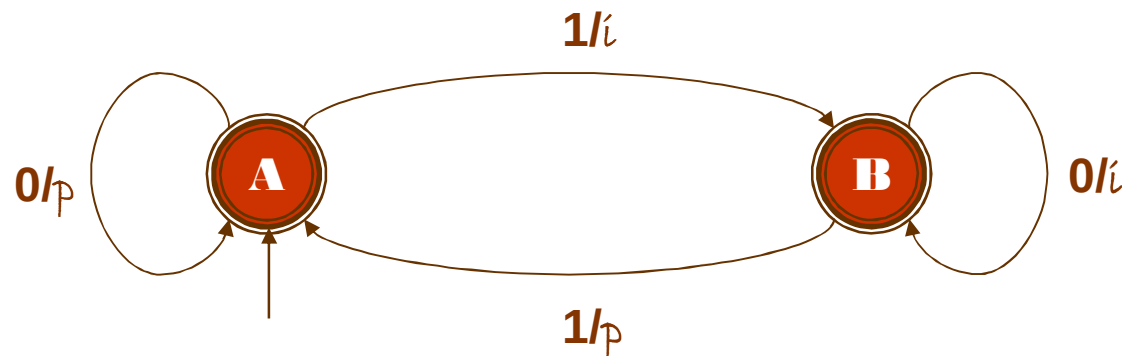
Circuitos Combinacionais

- *Os valores dos sinais de saída do circuito dependem somente dos valores dos sinais de entrada deste.*

Circuitos Sequenciais

- *Além dos valores dos sinais de entrada do circuito, os valores dos sinais de saída dependem também do estado do circuito.*

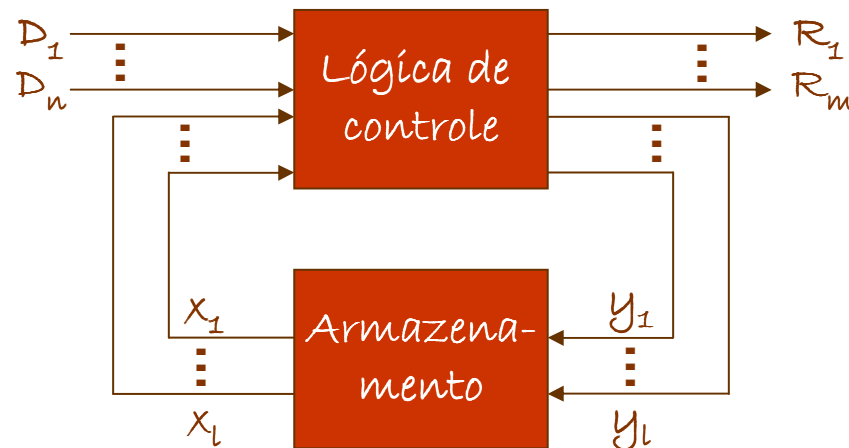
Exemplo



INTRODUÇÃO

Arquitetura Geral de um Sistema Seqüencial

- Em geral, um sistema seqüencial é constituído de 2 partes principais: uma lógica de controle e outra de armazenamento.
- Os sinais de entrada da lógica de controle são os sinais de entrada do sistema e a informação sobre o estado atual deste.
- Os sinais de saída da lógica de controle são os sinais de saída do sistema e a informação sobre próximo estado deste.



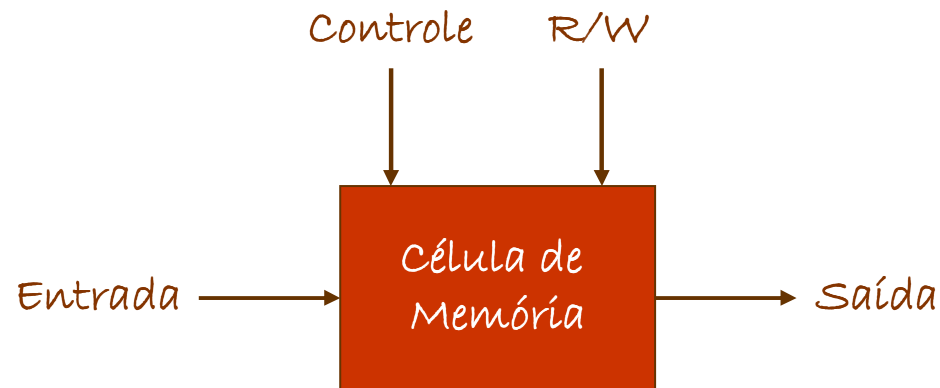
INTRODUÇÃO

Memória de Circuitos Sequenciais

- *Uma célula de memória permite “lembrar” o valor de um bit. Isso é implementado através de três operações fundamentais:*
 1. *Escrita: consiste em transportar o valor na entrada para dentro do elemento de memória;*
 2. *Armazenamento: permite manter o valor na entrada na última ocasião em que houve uma operação de escrita, mesmo se houver mudanças na entrada após a operação de escrita;*
 3. *Leitura: consiste em fornecer o valor armazenado na saída.*

INTRODUÇÃO

- O comportamento de uma célula de memória é:
 1. Se $\text{Controle} = 0$ $\Rightarrow$ Armazenamento
 2. Se $\text{Controle} = 1$ e $R/W = 0$ $\Rightarrow$ Escrita
 3. Se $\text{Controle} = 1$ e $R/W = 1$ $\Rightarrow$ Leitura



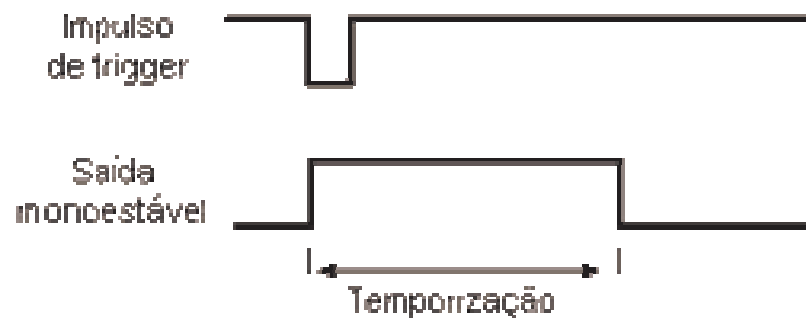
MULTI-VIBRADORES

Definição

- Um multi-vibrador é um circuito que pode mudar de estado ($0 \rightarrow 1$ e $1 \rightarrow 0$) indefinidamente. Existem três diferentes tipos de multi-vibradores: mono-estáveis, astáveis e biestáveis.

Mono-estáveis

- Num circuito mono-estável, a saída produz um impulso quando estimulado. Assim a saída só é estável em um estado.

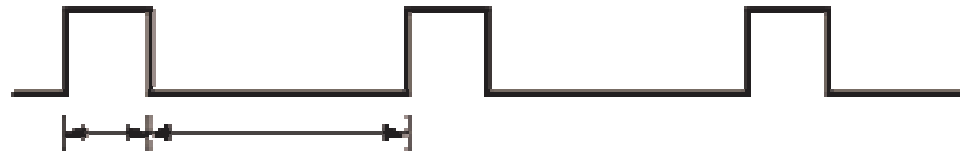


- Circuitos mono-estáveis são geralmente usados como temporizadores.

MULTI-VIBRADORES

Astáveis

- *Num circuito astável, a saída nunca fica estável em nenhum dos dois estados possíveis. Logo produz um trem de pulsos com uma determinada frequência.*

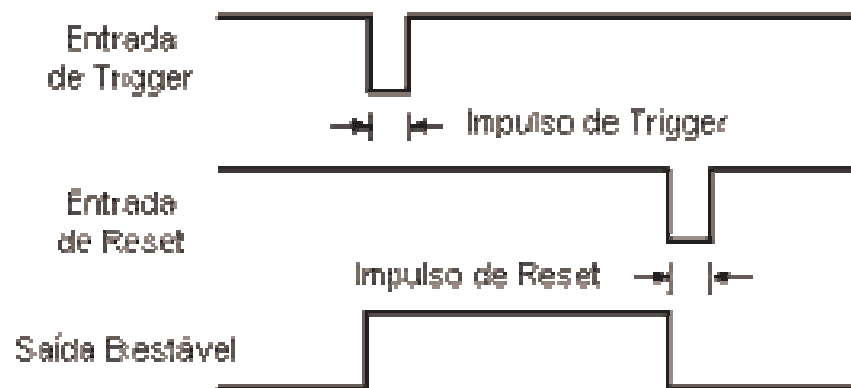


- *Circuitos astáveis são geralmente usados como osciladores*

MULTI-VIBRADORES

Biestáveis

- Num circuito biestável, a saída fica estável num dos dois estados possíveis. A mudança de estado ocorre quando o circuito é estimulado.*



- Circuitos biestáveis são geralmente usados como comutadores.*

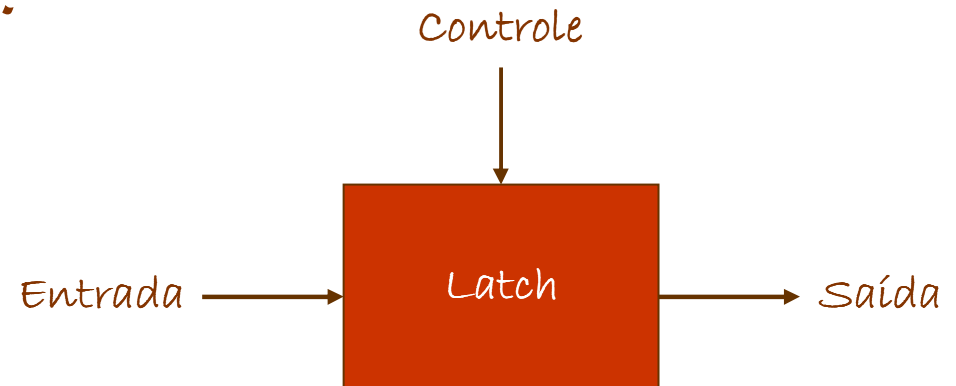
MULTI-VIBRADORES

Latches ou Trincos

- Um latch é um elemento de memória simples e se comporta da seguinte forma:

$$\text{Saída} = \begin{cases} \text{Entrada} & \text{Se Controle} = 1 \\ \text{Saída}^* & \text{Se Controle} = 0 \end{cases}$$

onde Saída* representa o valor do sinal Entrada na última ocasião em que Controle = 1.

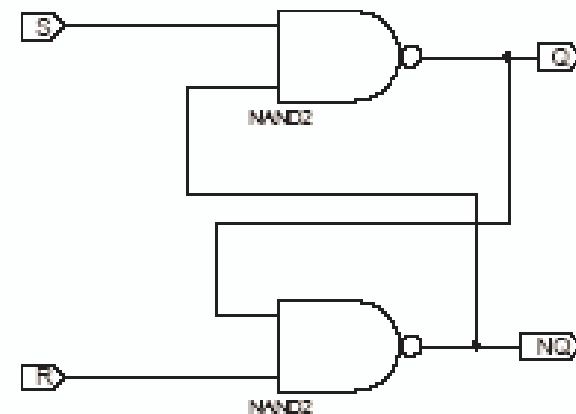


MULTI-VIBRADORES

Latch SR-NAND Básico

- Os sinais de entrada do circuito de latch SR-NAND são geralmente S (set) e R (reset), e os de saída Q e seu complemento $\overline{Q}$.
- O latch SR-NAND tem o seguinte comportamento (tabela):

S	R	Q	$\overline{Q}$	operação
0	0	1	1	Não utilizada
	1	1	0	Set
1		0	1	Reset
1	1	Q	$\overline{Q}$	Armazenamento

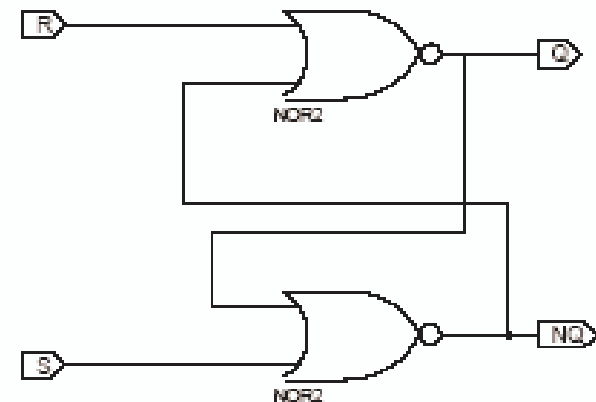


MULTI-VIBRADORES

Latch SR-NOR Básico

- Como no caso do latch SR-NAND, os sinais de entrada do circuito latch SR-NOR são geralmente S (set) e R (reset) e os de saída Q e seu complemento $\overline{Q}$.
- O latch SR-NOR tem o seguinte comportamento (tabela):

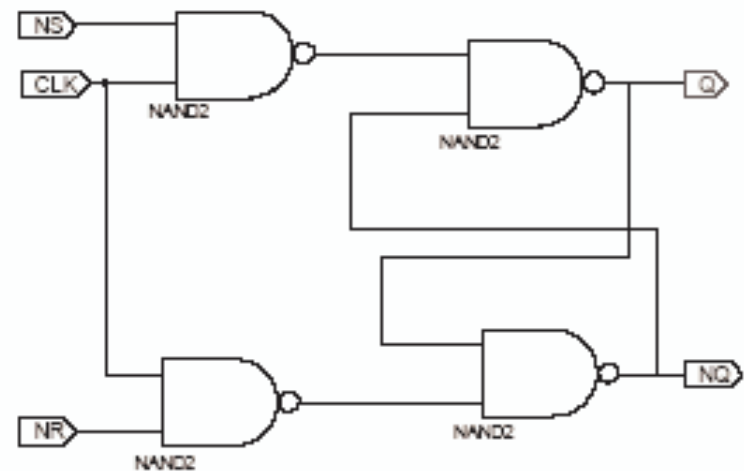
S	R	Q	$\overline{Q}$	operação
0	0	Q	$\overline{Q}$	Armazenamento
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Não utilizada



MULTI-VIBRADORES

Latch SR-NAND com Clock

- Além das entradas S e R , um latch SR-NAND com clock tem uma entrada de controle geralmente chamada CLOCK (CLK).
- A operação de armazenamento ocorre quando $CLK = 0$.
- As entradas S e R de um latch SR-NAND com clock são ativas somente no caso $CLK = 1$, funcionando como num latch SR-NAND básico.
- Observe que o armazenamento ocorre também quando $(CLK=1)$ e $(S=1)$ e $(R=1)$
- É usado para controlar o fluxo de dados, sincronizando as mudanças de estados de vários circuitos de latch.

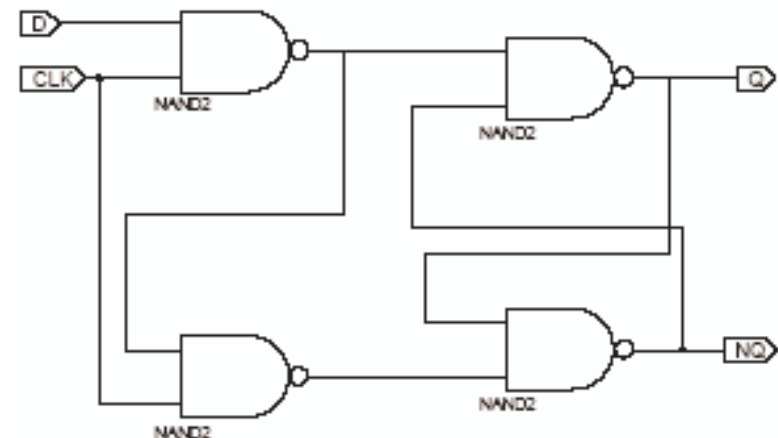


MULTI-VIBRADORES

Latch D

- As situações ilegais nos latches SR-NAND and SR-NOR com ou sem clock podem ser eliminadas utilizando um latch D (Data) com clock.
- O latch D é um latch SR-NAND com a entrada $R = D$ e a entrada $S = \overline{D}$. Assim eliminamos a possibilidade de ter $R = S$.
- A entrada CLK permite a operação de armazenamento.
- O latch D tem o seguinte comportamento (tabela):

CLK	D	Q	$\overline{Q}$	operação
0	X	Q	$\overline{Q}$	Armazenamento
1	0	0	1	Reset
1	1	1	0	Set



MULTI-VIBRADORES

Latches vs. Flip-Flops

- *Um circuito de latch é geralmente transparente no sentido de que as saídas acompanham as mudanças que ocorrem nas entradas na maioria do tempo.*
- *Um circuito de flip-flop não é transparente. As saídas não acompanham as mudanças nas entradas.*
- *Circuitos de latch são sensíveis ao nível dos sinais de controle enquanto flip-flops são sensíveis à transição de um estado ao outro dos sinais de controle.*

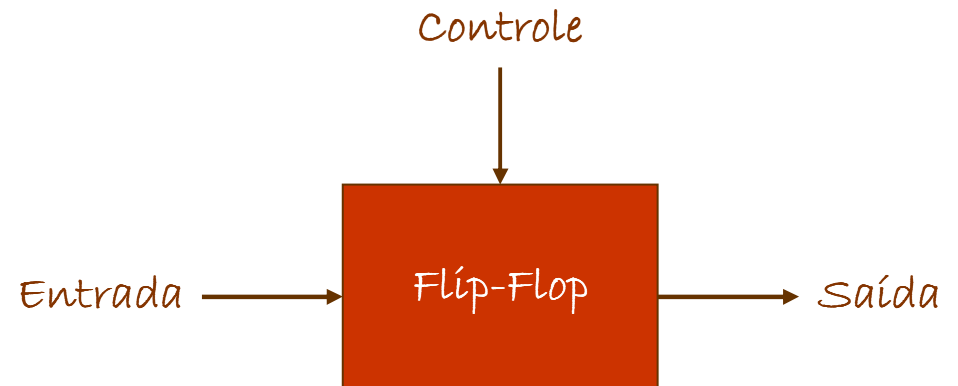
MULTI-VIBRADORES

Funcionamento de um Flip-Flop

- *Um flip-flop se comporta da seguinte forma:*

$$\text{Saída} = \begin{cases} \text{Entrada} & \text{Se Controle} = \text{ } \downarrow \text{ (} \uparrow \text{)} \\ \text{Saída}^* & \text{Senão} \end{cases}$$

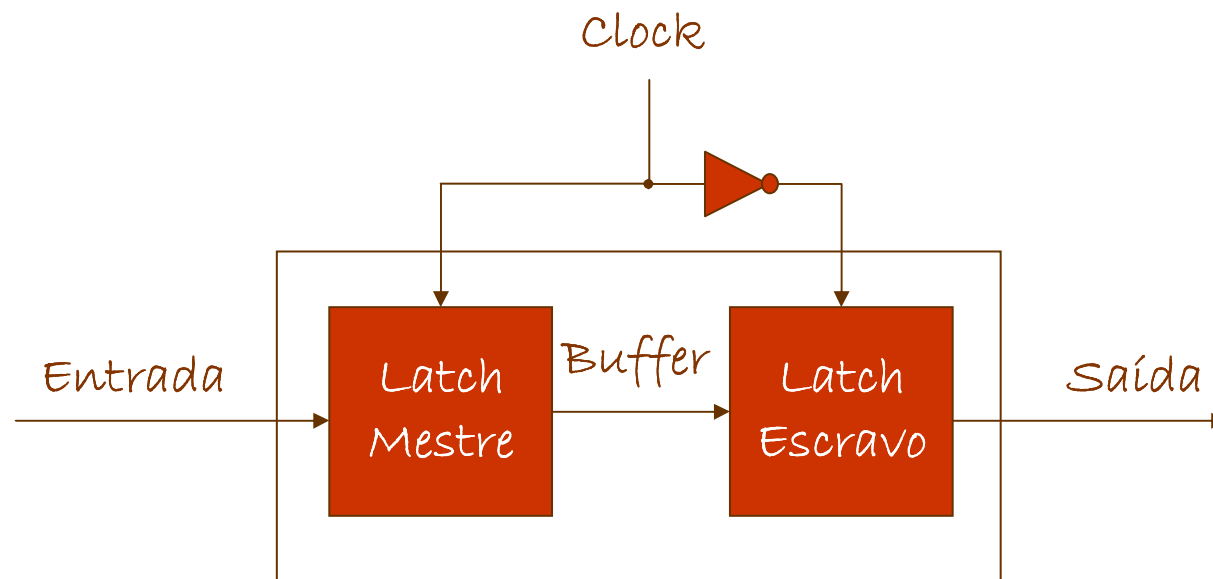
onde Saída representa o valor do sinal Entrada no momento da transição de estado do sinal Controle.*



MULTI-VIBRADORES

Estrutura Mestre-Escravo

- Para que um flip-flop possa mudar de estado, somente na transição do clock, é necessário empregar dois circuitos de latch em operação inversa e ligados em cascata.*



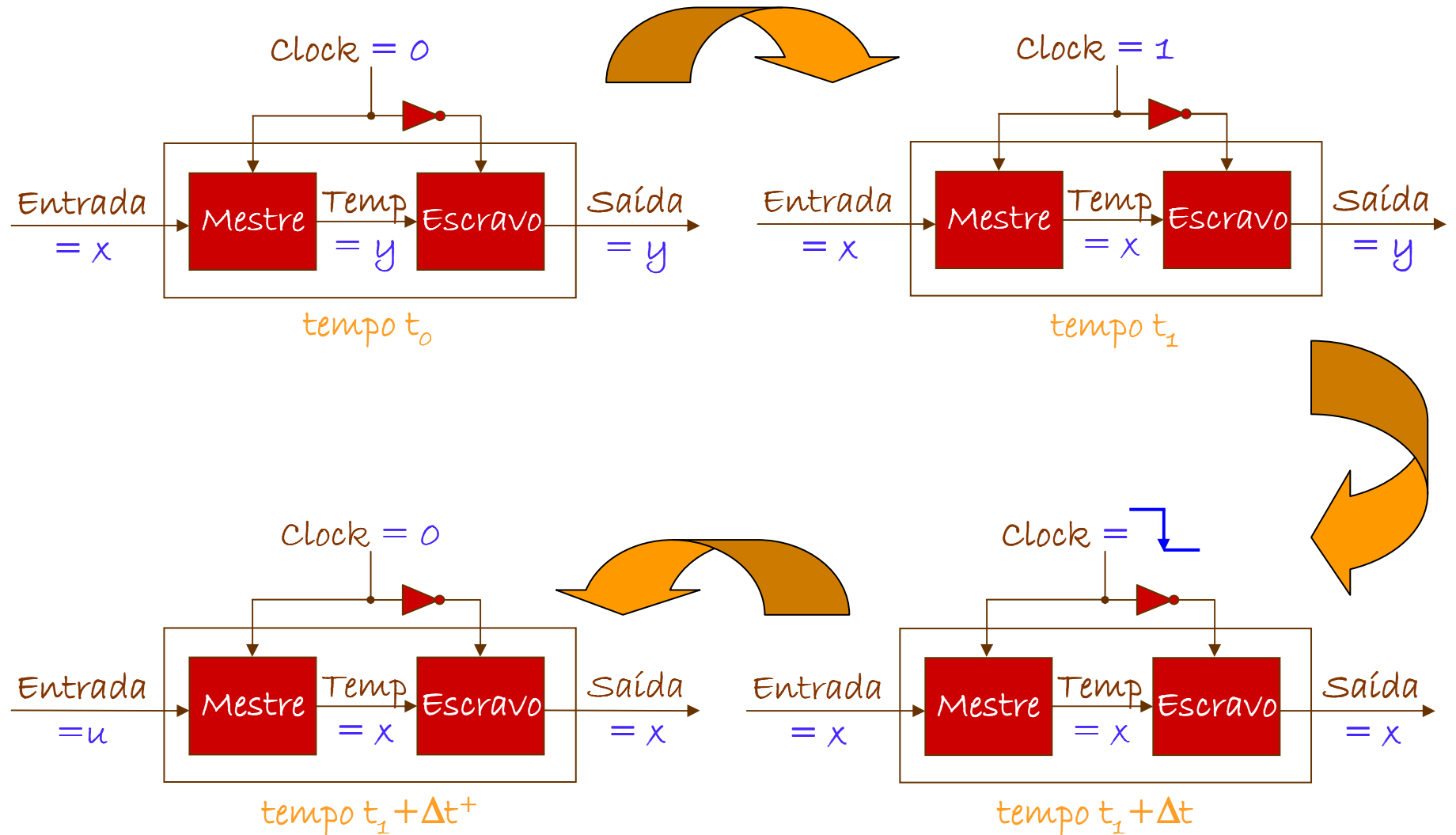
MULTI-VIBRADORES

Comportamento da Estrutura Mestre-Escravo

- Quando o primeiro latch estiver ativo ($\text{clock} = 1$) e, por conseqüência, transportar a Entrada para o Buffer, o segundo latch estará inativo ($\overline{\text{clock}} = 0$) ou em situação de armazenamento.
- Quando o primeiro latch estiver inativo ($\text{clock} = 0$), e então em situação de armazenamento, o segundo latch estará ativo ($\overline{\text{clock}} = 1$) e, por conseqüência, transportará o conteúdo do Buffer para a Saída.
- Somente na mudança do clock do estado 1 para o estado 0, ocorrerá a mudança do conteúdo do sinal saída.

MULTI-VIBRADORES

Estrutura Mestre-Escravo



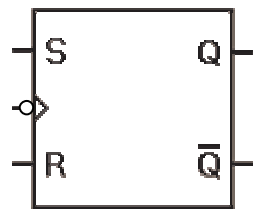
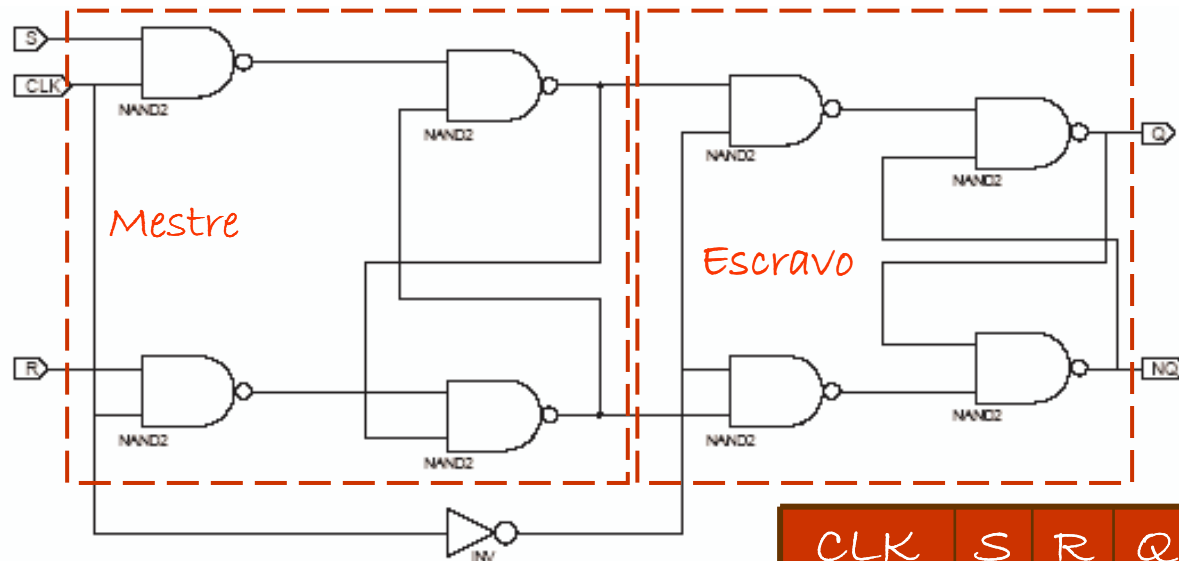
MULTI-VIBRADORES

Flip-Flop tipo SR-NAND

- *É composto de dois circuitos de latch SR-NAND com sinais de clock invertidos;*
- *Independente do nível do clock, o flip-flop está em situação de armazenamento;*
- *Quando S e R estão no estado 0 no momento da transição negativa do clock, o flip-flop estará também em situação de armazenamento;*
- *Quando S e R estão no estado 1 no momento da transição negativa do clock, o flip-flop estará em situação de conflito e, por tanto esta combinação não será utilizada.*

MULTI-VIBRADORES

Comportamento Flip-Flop Tipo SR-NAND



*Símbolo do flip-flop
(transição negativa)*

CLK	S	R	Q	$\overline{Q}$	operação
0	x	x	Q	$\overline{Q}$	Armazenamento
1	x	x	Q	$\overline{Q}$	Armazenamento
$\downarrow$	0	0	Q	$\overline{Q}$	Armazenamento
$\downarrow$	0	1	0	1	Reset
$\downarrow$	1	0	1	0	Set
$\downarrow$	1	1	1	1	Não utilizada

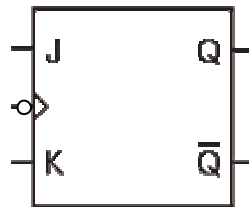
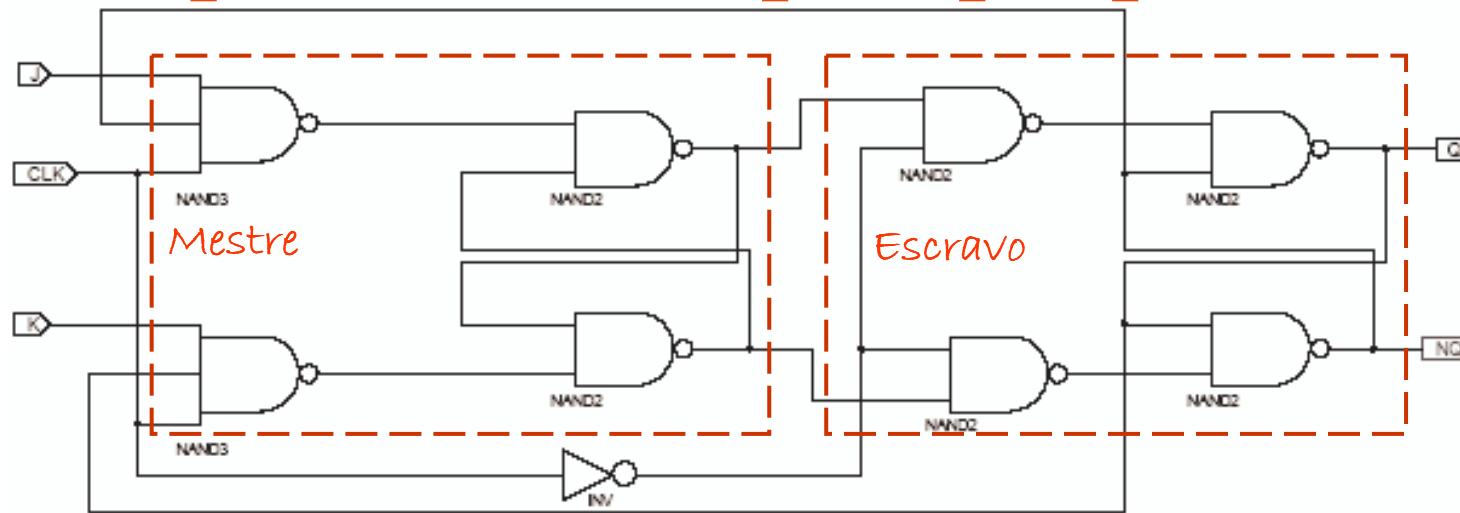
MULTI-VIBRADORES

Flip-Flop Tipo JK-NAND

- *É um flip-flop SR com realimentação das saídas Q e $\overline{Q}$ como entradas para as portas NAND do primeiro nível; As entradas S e R são chamadas J e K respectivamente;*
- *Isto permite que a entrada J seja sem efeito quando o flip-flop está no estado 1 e que entrada K seja sem efeito quando o flip-flop está no estado 0;*
- *Tem o mesmo comportamento que um flip-flop SR, salvo a situação de conflito;*
- *Diferentemente de SR, quando as entradas J e K estão no estado 1 no momento da transição negativa do clock, o flip-flop JK muda de estado ou “toggles”. Isto é, se $Q = 0$ e $\overline{Q} = 1$ então o flip-flop muda para $Q = 1$ e $\overline{Q} = 0$ e vice-versa.*

MULTI-VIBRADORES

Comportamento Flip-Flop Tipo JK



*Símbolo do flip-flop
(transição negativa)*

CLK	J	K	Q	$\bar{Q}$	operação
0	x	x	Q	$\bar{Q}$	Armazenamento
1	x	x	Q	$\bar{Q}$	Armazenamento
$\downarrow$	0	0	Q	$\bar{Q}$	Armazenamento
$\downarrow$	0	1	0	1	Reset
$\downarrow$	1	0	1	0	Set
$\downarrow$	1	1	$\bar{Q}$	Q	toggle

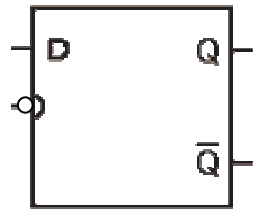
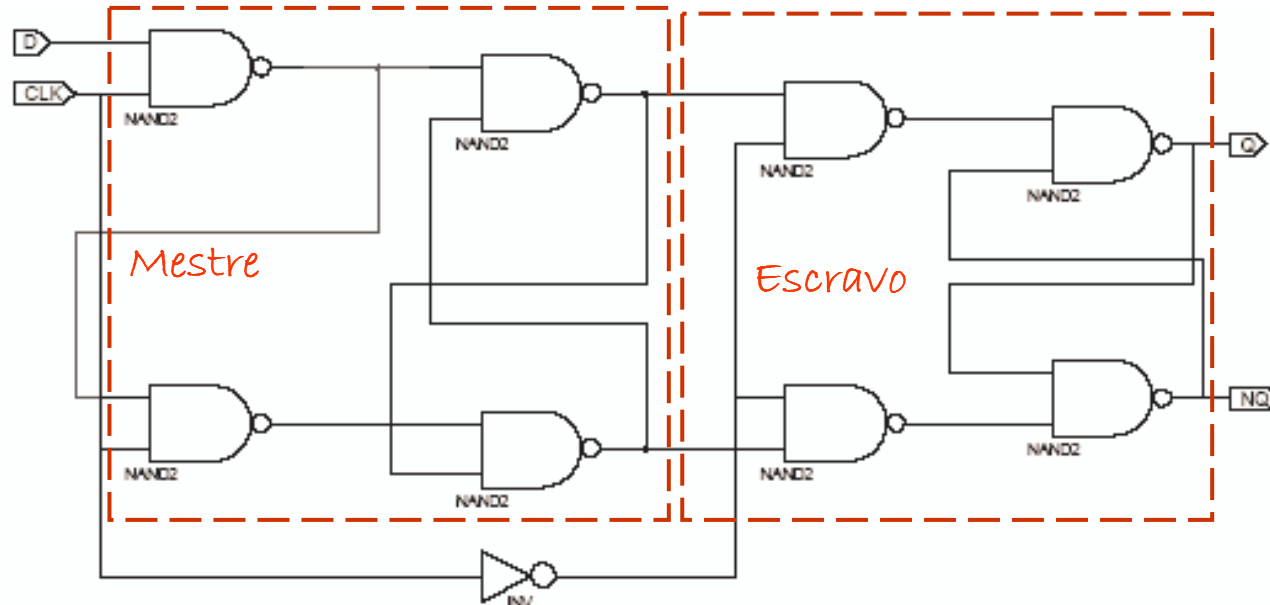
MULTI-VIBRADORES

Flip-Flop Tipo D

- *É composto de um circuito de latch tipo D ligado em cascata com um circuito de latch tipo SR com clockes invertidos;*
- *Tem o mesmo comportamento que um flip-flop da tipo JK, exceto pela situação de toggle já que a combinação de dados de entrada que gerava a situação de conflito não pode acontecer.*

MULTI-VIBRADORES

Comportamento do Flip-Flop do tipo D



Símbolo do flip-flop

CLK	D	Q	$\overline{Q}$	operação
0	x	Q	$\overline{Q}$	Armazenamento
1	1	Q	$\overline{Q}$	Armazenamento
	0	0	1	Reset
	1	1	0	Set

MULTI-VIBRADORES

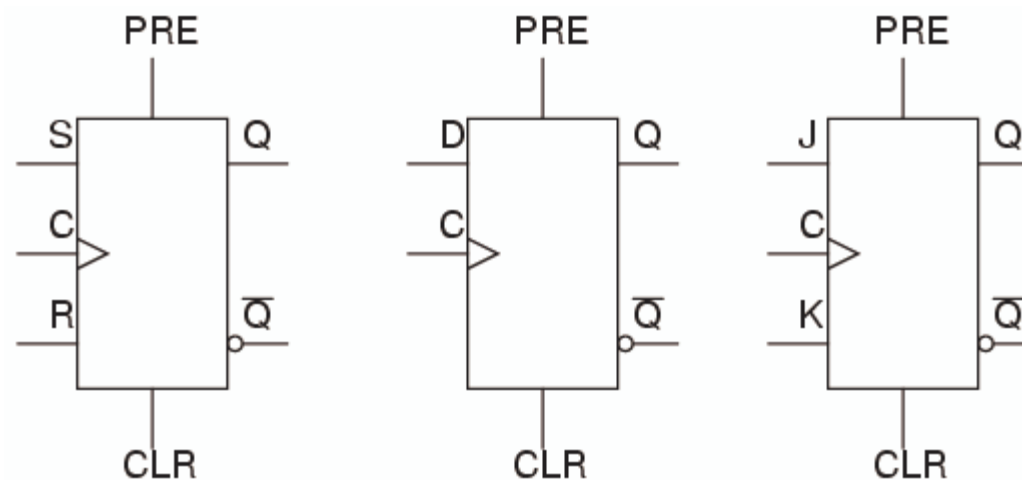
Entradas Assíncronas de Flip-Flop

- *As entradas de flip-flop, tais como D, J, K, R e S são chamadas entradas síncronas porque estas tem efeito nas saídas (Q e $\overline{Q}$) somente numa das transições (negativa ou positiva) do sinal de clock.*
- *Existem duas entradas de flip-flops que são assíncronas. Estas podem mudar as saídas do flip-flop (Set e Reset) independentemente do estado do sinal de clock.*
- *Geralmente, as entradas assíncronas do flip-flop são chamadas de Preset (PRE) e Clear (CLR).*

MULTI-VIBRADORES

Uso de Entradas Assíncronas

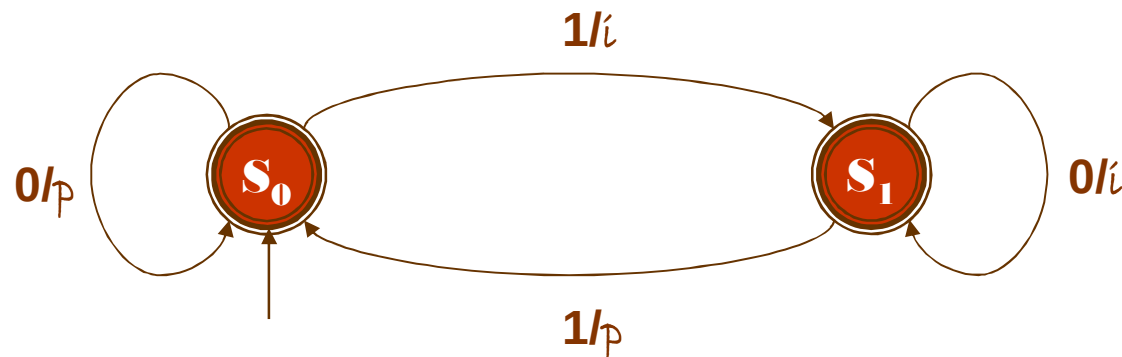
- Quando Preset é ativado, a saída Q vai para o estado 1 e $\overline{Q}$ para 0;
- Quando Clear é ativado, a saída Q vai para o estado 0 e $\overline{Q}$ para 1;
- Quando os dois sinais, Preset e Clear, são ativados no mesmo tempo, o resultado é indefinido.



MÁQUINAS DE ESTADOS

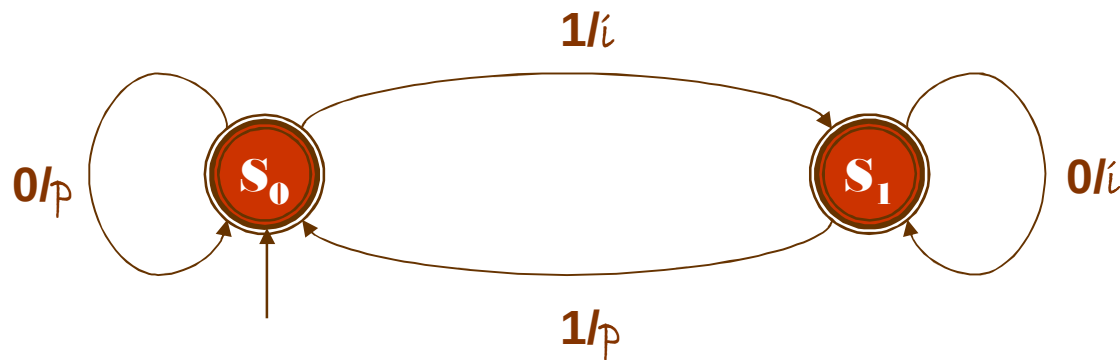
Definição

- *Uma máquina de estados é definida por 6 elementos:*
 - *Estado inicial* $\{S_0\}$
 - *Alfabeto de entrada* $\{0, 1\}$
 - *Alfabeto de saída* $\{p, i\}$
 - *Conjunto de estados* $\{S_0, S_1\}$
 - *A função de transição de estados* *diagrama*
 - *Estados finais* $\{S_0, S_1\}$



MÁQUINAS DE ESTADOS

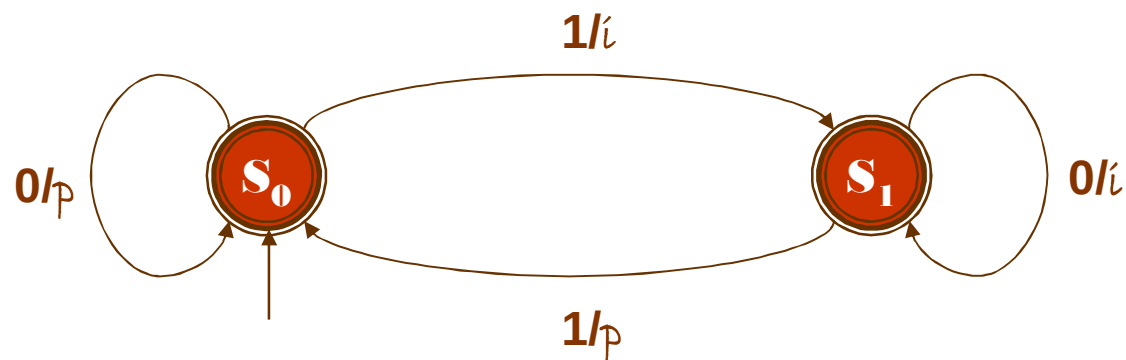
- Num dado instante t , o estado em que a máquina se encontra é chamado estado atual;
- No instante $t + 1$, o estado em que a máquina se encontrará é chamado próximo estado;
- estado atual da máquina junto com as entradas definem o estado total da máquina de estados.
- O estado atual da máquina junto com as entradas determinam o próximo estado e as saídas;



MÁQUINAS DE ESTADOS

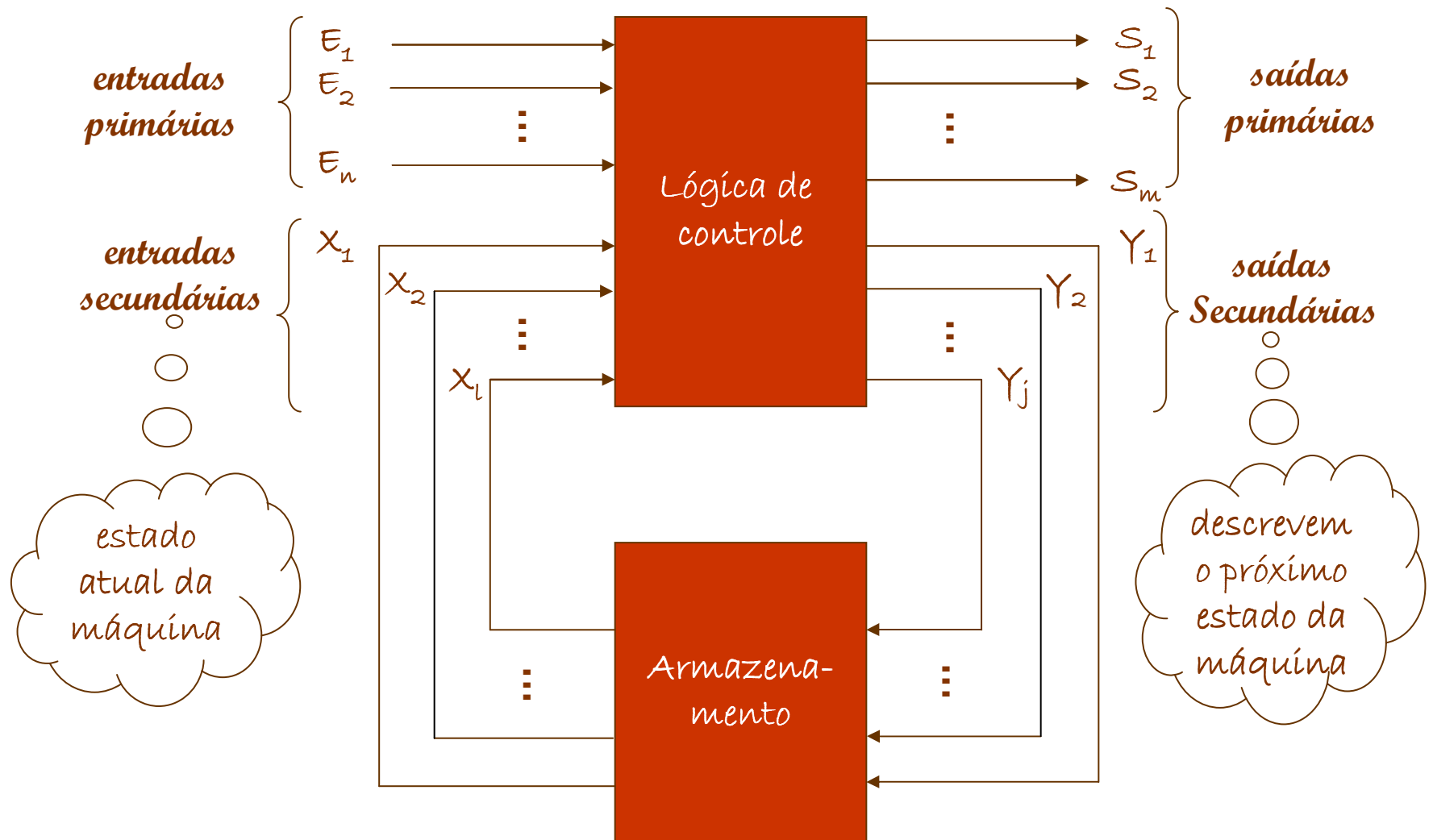
- A função de transição de estados pode ser apresentada numa forma tabular.*

estado atual	próximo estado		saída	
	entrada = 0	entrada = 1	entrada = 0	entrada = 1
s_0	s_0	s_1	p	i
s_1	s_1	s_0	i	p



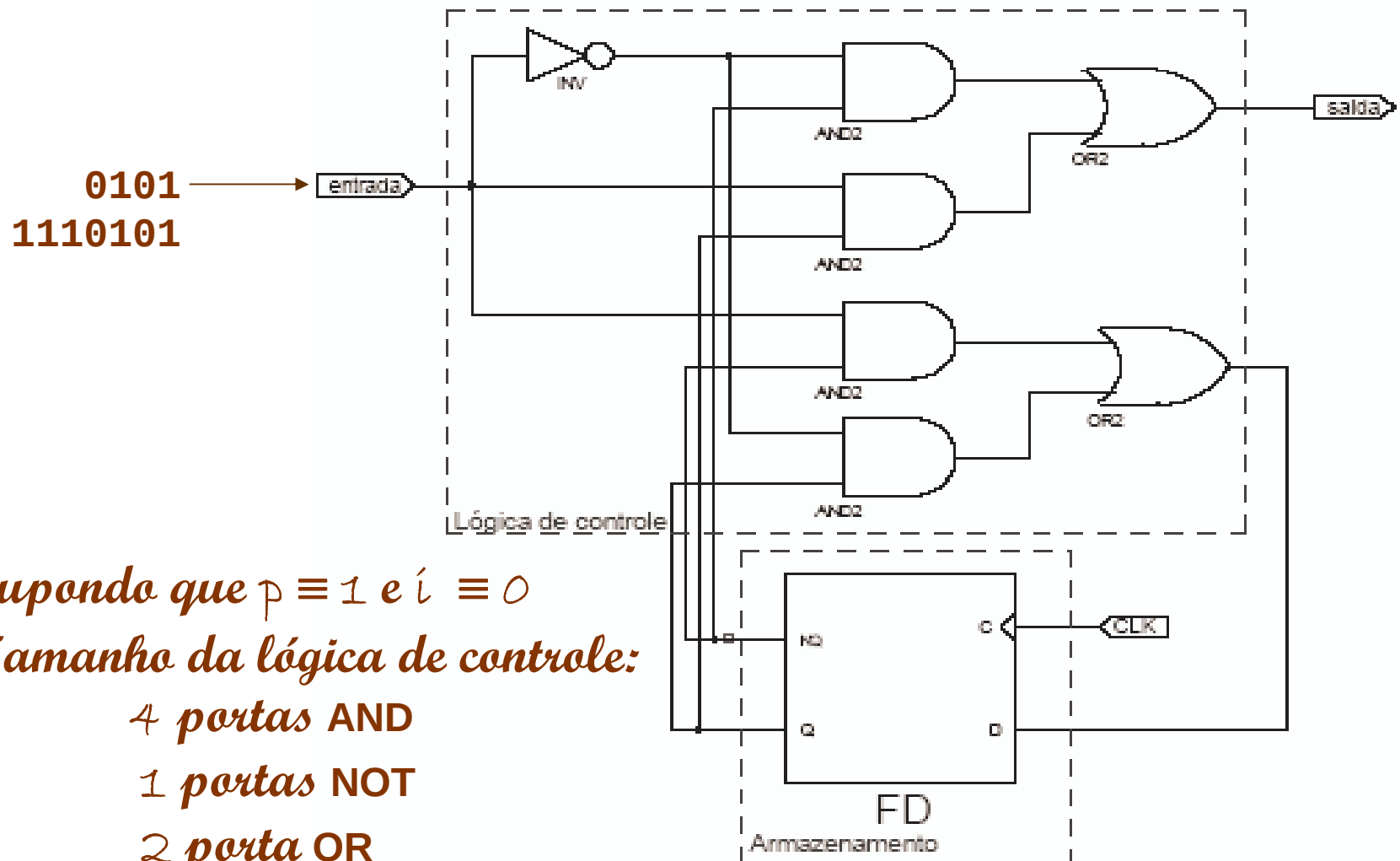
MÁQUINAS DE ESTADOS

Arquitetura de uma Máquina de Estados Genérica



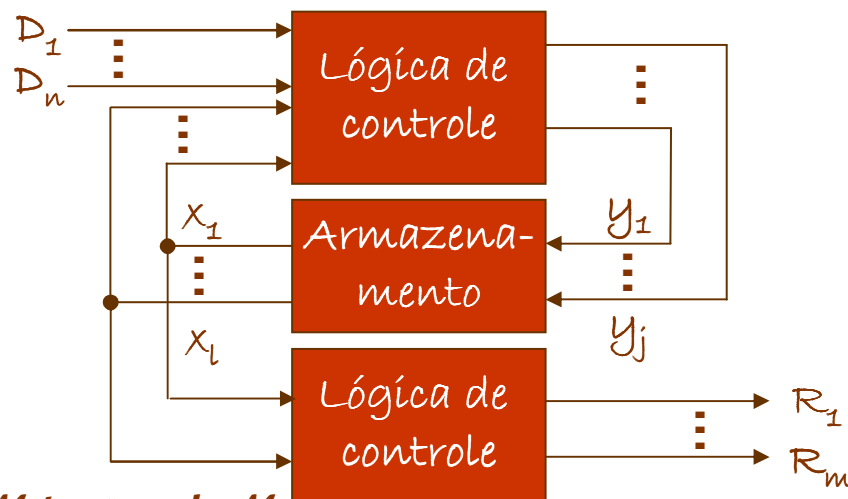
MÁQUINAS DE ESTADOS

Exemplo de Implementação de Máquina de Estado

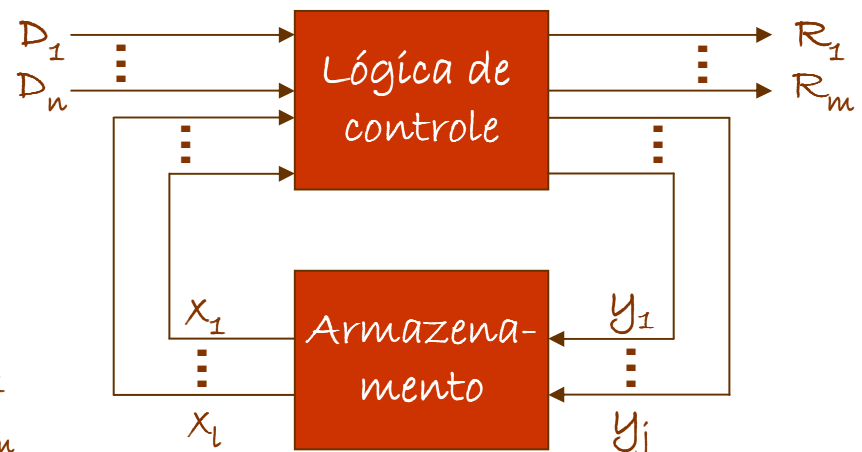


MÁQUINAS DE ESTADOS

- *As máquinas de estados são classificadas de acordo com a função de saída, da seguinte maneira:*
 - *A máquina de Mealy: a saída no tempo t depende do estado e da entrada no tempo t .*
 - *A máquina de Moore: a saída no tempo t depende somente do estado no tempo t .*
- *Note que a saída de uma máquina de Moore não é independente da seqüência de entrada.*



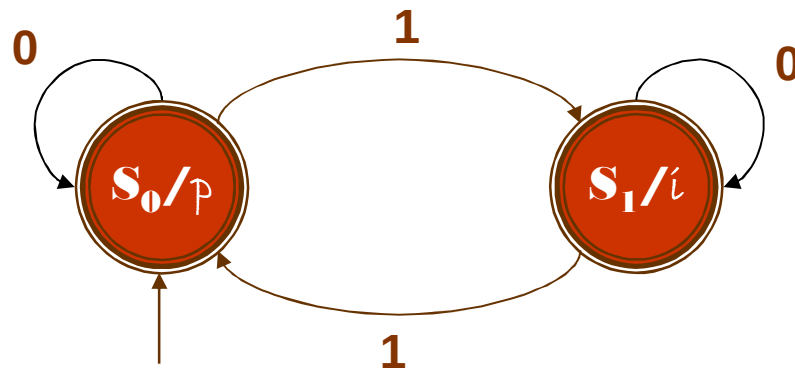
Máquina de Moore



Máquina de Mealy

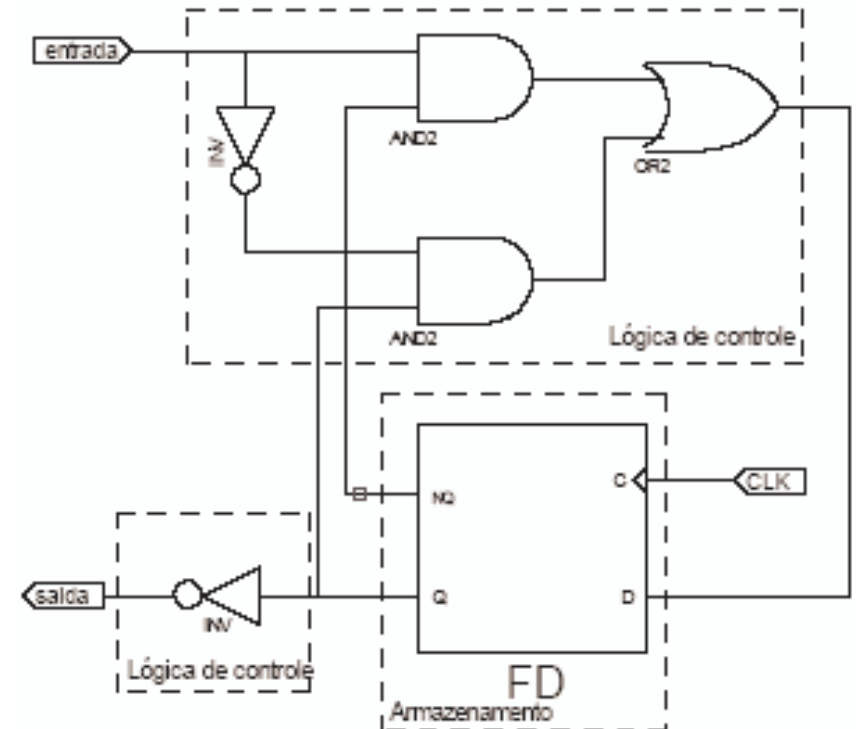
MÁQUINA DE ESTADOS

Exemplo de Implementação de Máquina de Moore



Supondo que $p \equiv 1$ e $i \equiv 0$

estado atual	próximo estado		saída
	entrada = 0	entrada = 1	
S_0	S_0	S_1	p
S_1	S_1	S_0	i



*Somente 2 portas AND
2 portas NOT
1 porta OR*

MÁQUINA DE ESTADOS

Projetos de Sistemas Sequenciais

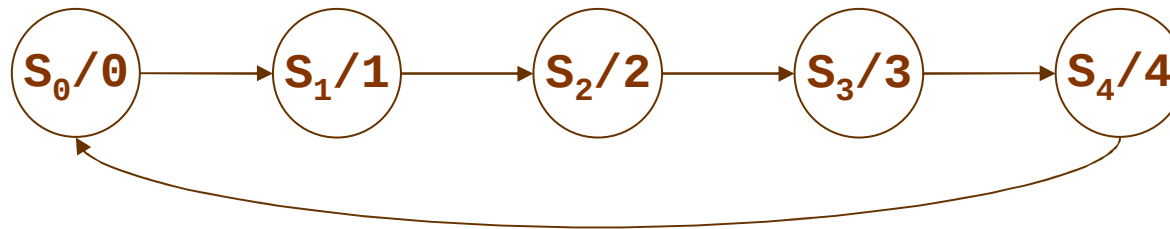


MÁQUINA DE ESTADOS

- O comportamento de um flip-flop de um período de clock para o seguinte pode ser um dos seguintes:
 1. Se o flip-flop é no estado 0 e permanece no estado 0, então seu comportamento é estático e pode ser representado por 0.
 2. Se o flip-flop é no estado 0 e muda para o estado 1, então seu comportamento é dinâmico e pode ser representado por α .
 3. Se o flip-flop é no estado 1 e muda para o estado 0, então seu comportamento é dinâmico e pode ser representado por β .
 4. Se o flip-flop é no estado 1 e permanece no estado 1, então seu comportamento é estático e pode ser representado por 1.
 5. Se mudança do estado do flip-flop não importa, então seu comportamento pode ser representado por $\times$ (don't care).

MÁQUINA DE ESTADOS

Tabela de Comportamento de uma Máquina



<i>Estado atual (Q)</i>	<i>Próximo estado (Q^+)</i>	<i>Símbolo no mapa</i>	<i>comportamento</i>
0	0	0	<i>estático</i>
0	1	α	<i>dinâmico</i>
1	0	β	<i>dinâmico</i>
1	1	1	<i>estático</i>
0	X	X	—
1	X	X	—

<i>atual</i>			<i>estado</i>			<i>comportamento</i>		
A	B	C	A ⁺	B ⁺	C ⁺	A	B	C
0	0	0	0	0	1	0	0	α
0	0	1	0	1	0	0	α	β
0	1	0	0	1	1	0	1	α
0	1	1	1	0	0	α	β	β
1	0	0	0	0	0	β	0	0
1	0	1	X	X	X	X	X	X
1	1	0	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X

MÁQUINA DE ESTADOS

Mapa de Transições e Tabelas de Excitação

A

		B	
	0	0	α
A	β	X	X

C

B

		B	
	0	α	β
A	0	X	X

C

C

		B	
	α	β	β
A	0	X	X

C

Comportamento	S	R
0	0	X
1	X	0
α	1	0
β	0	1
X	X	X

Tabela de excitação para FF-SR

Comportamento	J	K
0	0	X
1	X	0
α	1	X
β	X	1
X	X	X

Comportamento	D
0	0
1	1
α	1
β	0
X	X

Tabelas de transições

Tabela de excitação para FF-JK

Tabela de excitação para FF-D

MÁQUINA DE ESTADOS

Mapeamento Síncrono Indireto das excitações

- Este mapeamento permite a transformação do comportamento sequencial dos flip-flops em uma forma combinacional.

A

	0	0	α	0
A	β	X	X	X

$\underbrace{\hspace{10em}}_C$

S_A

	0	0	1	0
A	0	X	X	X

$\underbrace{\hspace{10em}}_C$

$$\Rightarrow S_A = B.C$$

B

	0	α	β	1
A	0	X	X	X

$\underbrace{\hspace{10em}}_C$

J_B

	0	1	X	X
A	0	X	X	X

$\underbrace{\hspace{10em}}_C$

$$\Rightarrow J_B = C$$

C

	α	β	β	α
A	0	X	X	X

$\underbrace{\hspace{10em}}_C$

D_C

	1	0	0	1
A	0	X	X	X

$\underbrace{\hspace{10em}}_C$

$$\Rightarrow D_C = \overline{A.C}$$

MÁQUINA DE ESTADOS

Mapeamento Síncrono Direto das Excitações

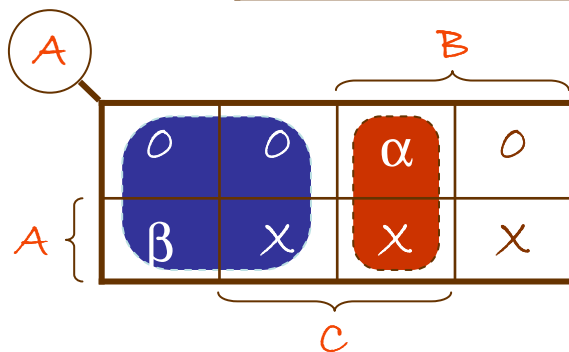
- *Este mapeamento permite a obtenção das equações de controle dos flip-flops diretamente a partir do comportamento seqüencial destes.*
- *Este método considere que o mapa de transições dos flip-flops contem três tipos de entradas:*
 - *Deve ser incluída: é aquela entrada com símbolo de transição que se tornaria num 1 na tabela de excitações;*
 - *Deve ser ignorada: é aquela entrada com símbolo de transição que se tornaria num 0 na tabela de excitações;*
 - *Redundância: Toda entrada num mapa de transições que não deve ser incluída nem ignorada.*

MÁQUINA DE ESTADOS

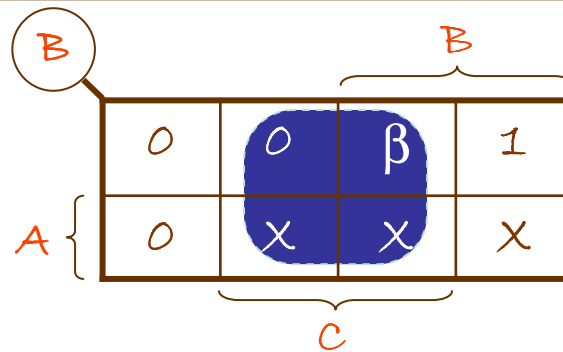
Mapeamento Síncrono Direto das Excitações

- A classificação das entradas do mapa de transições depende do tipo de flip-flop utilizado.

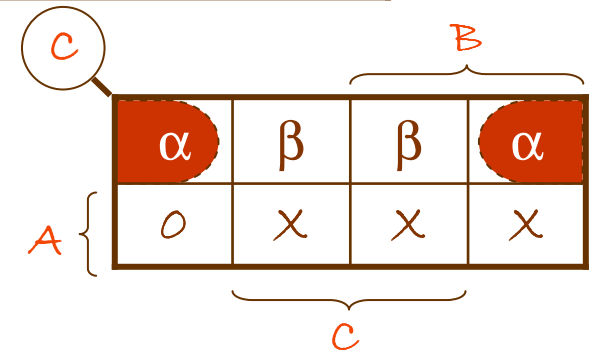
Tipo de FF	Controle	Incluída	Ignorada	Redundância
SR	S	α	$0, \beta$	$1, X$
	R	β	$1, \alpha$	$0, X$
JK	J	α	0	$1, \beta, X$
	K	β	1	$0, \alpha, X$
D	D	$1, \alpha$	$0, \beta$	X



$$S_A = B.C \text{ e } R_A = \bar{B}$$

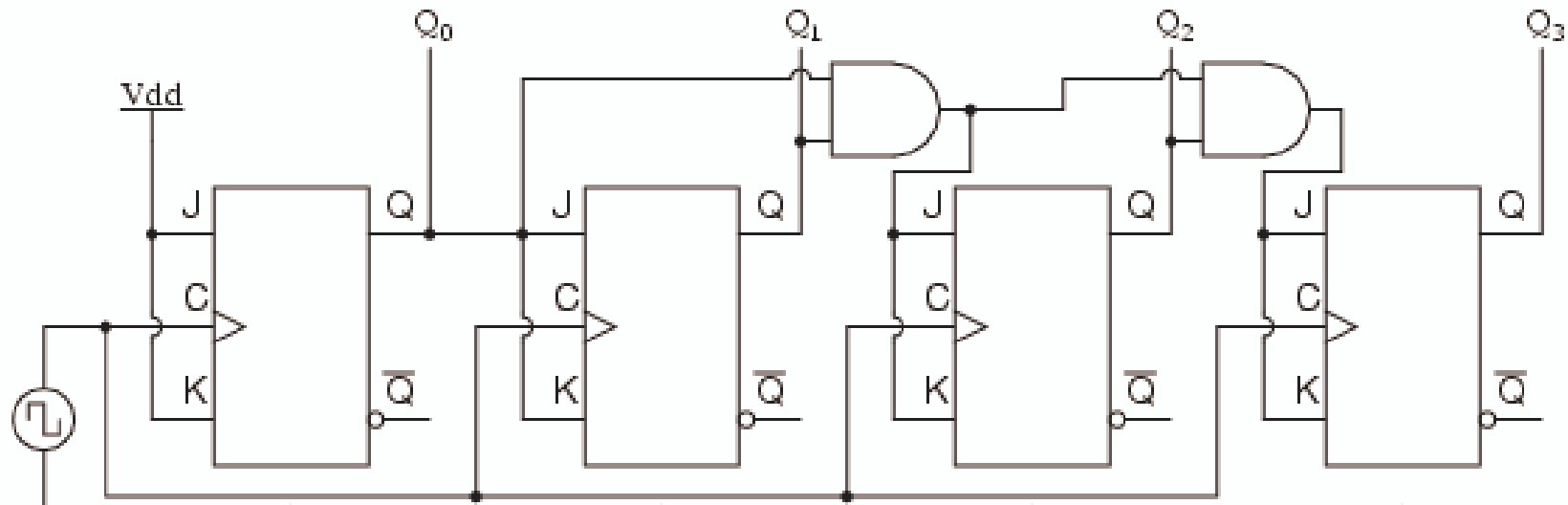


$$J_B = 1 \text{ e } K_B = C$$



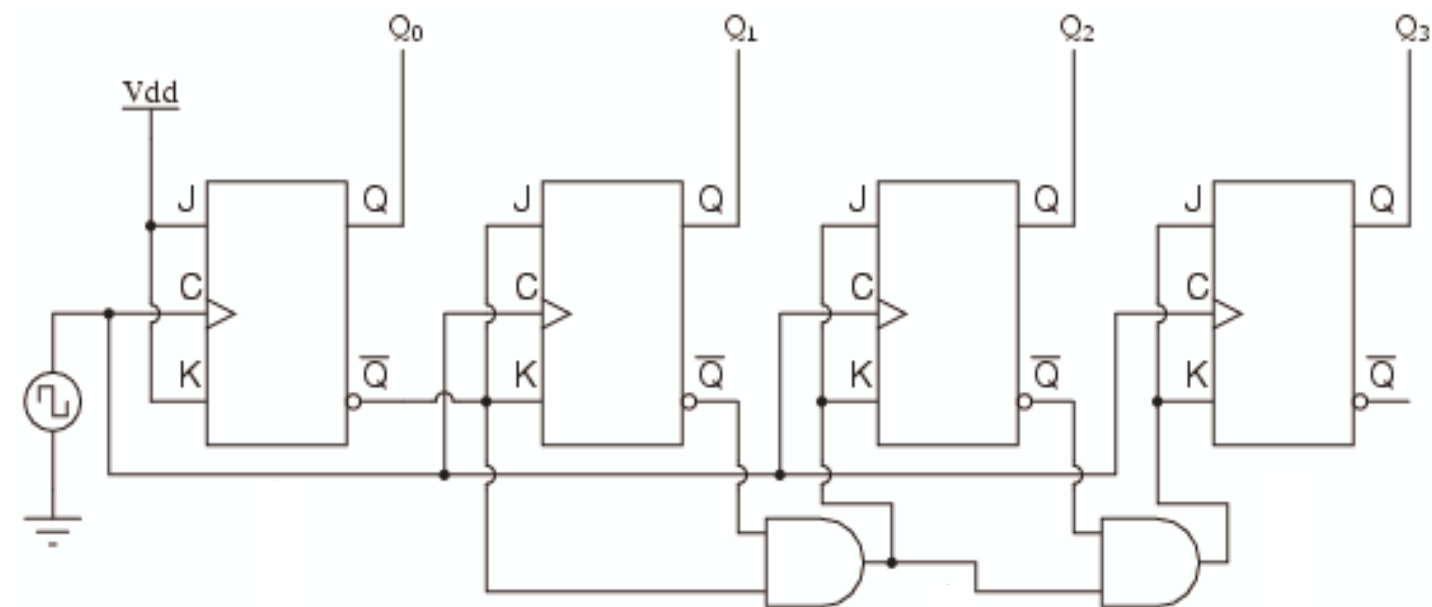
$$D_A = \bar{A}.\bar{C}$$

Contador Síncrono Crescente



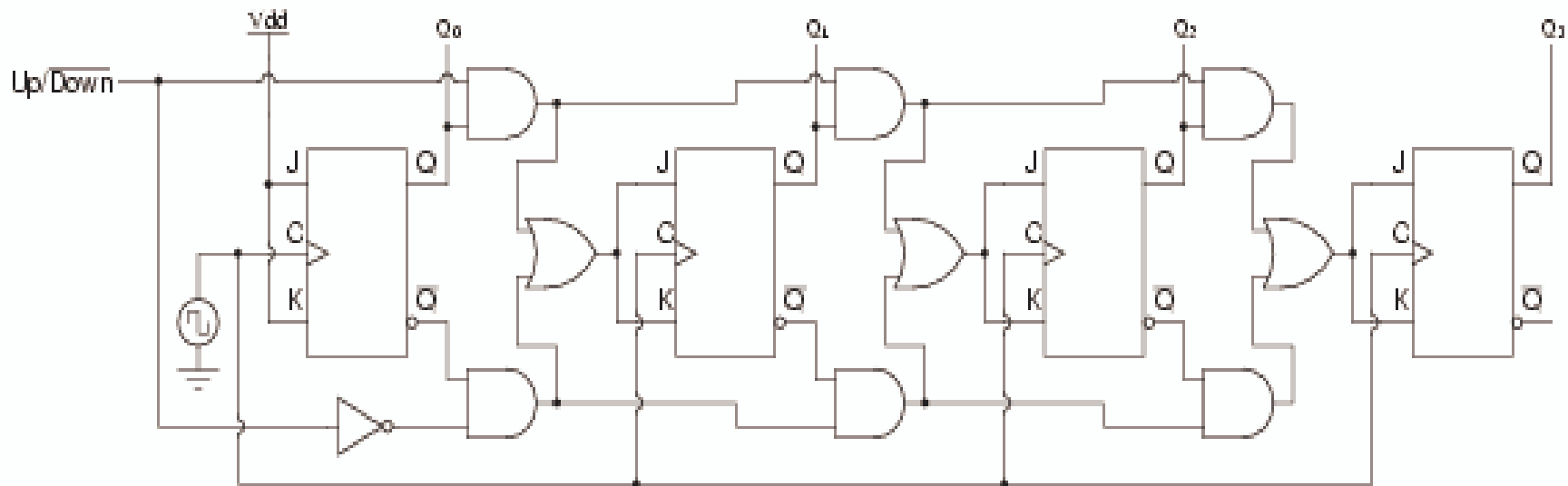
MÁQUINAS DE ESTADOS

Contador Síncrono Decrescente



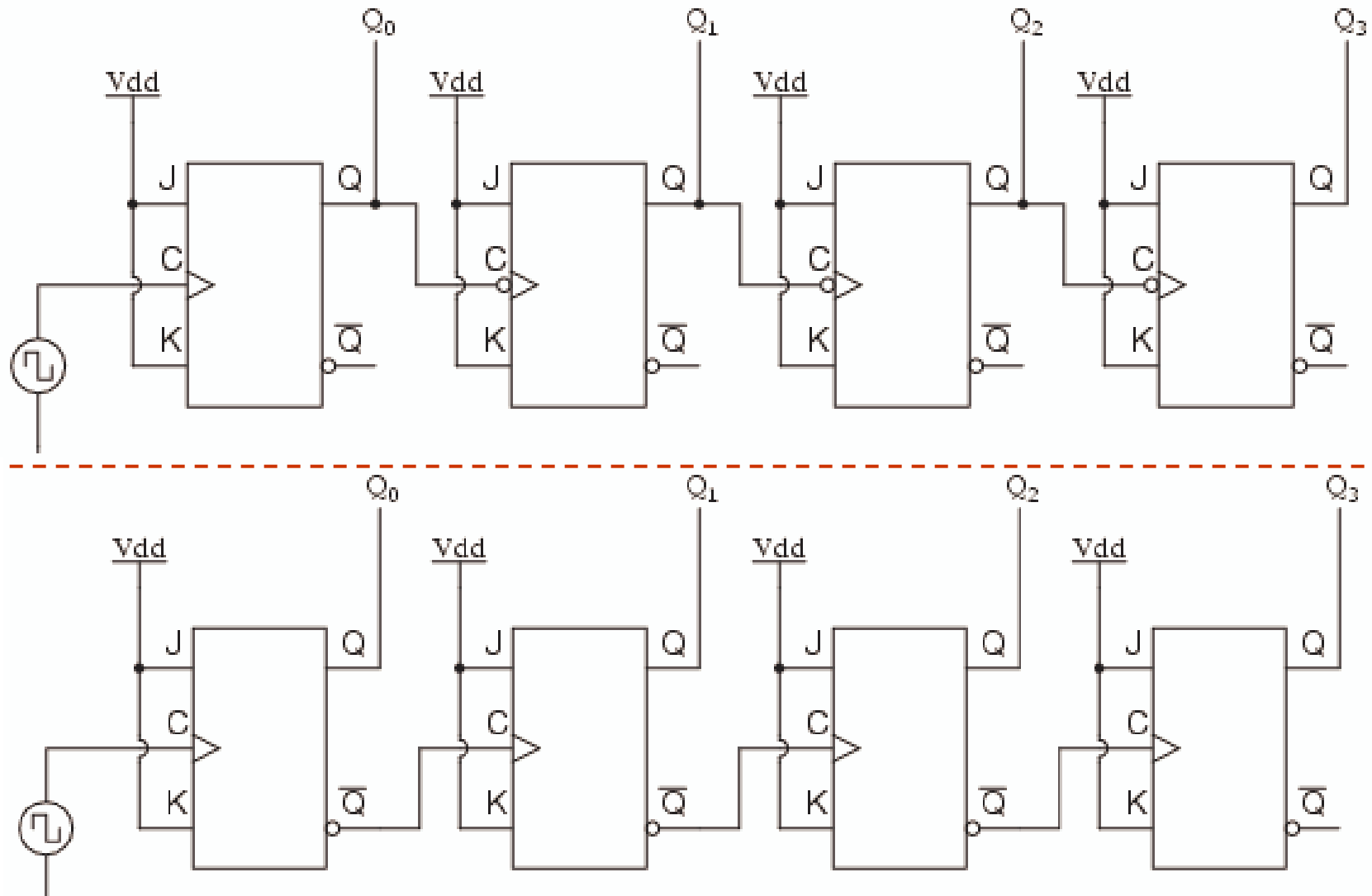
MÁQUINAS DE ESTADOS

Contador Síncrono Crescente/Decrescente



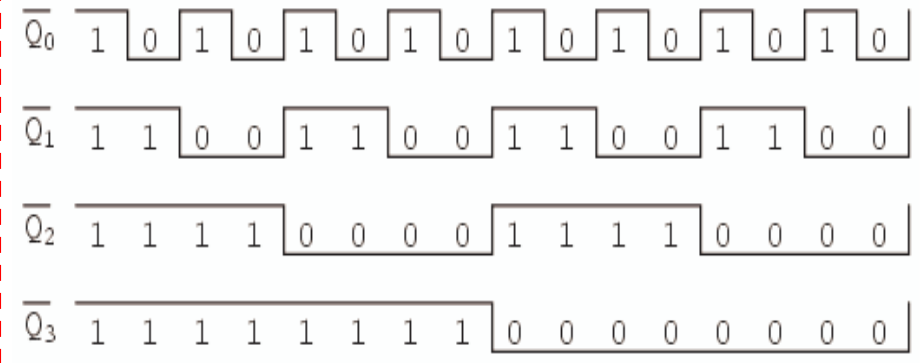
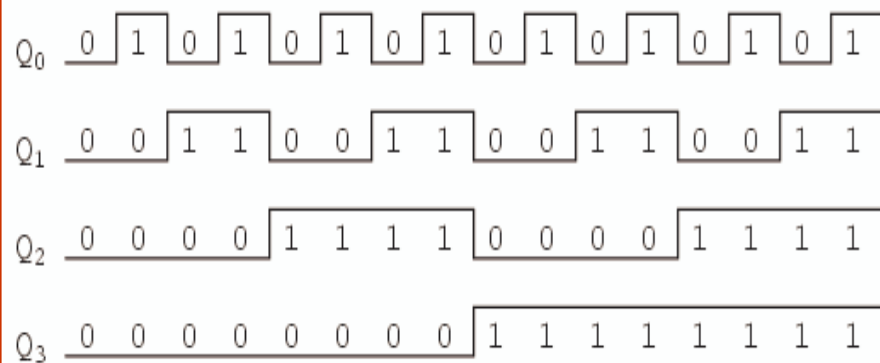
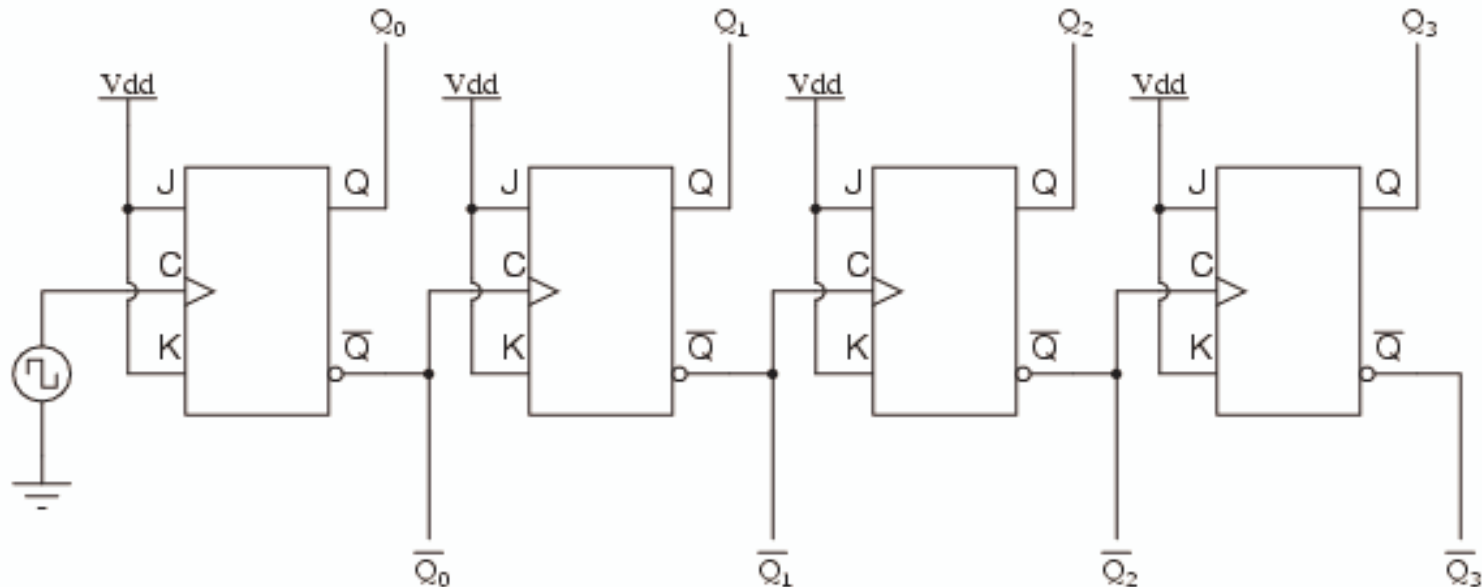
MÁQUINAS DE ESTADOS

Contadores Assíncronos Crescentes



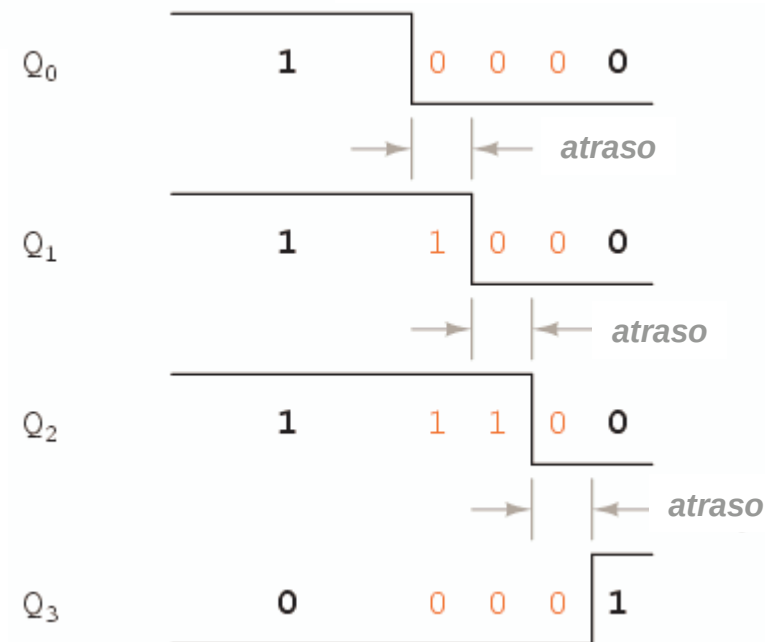
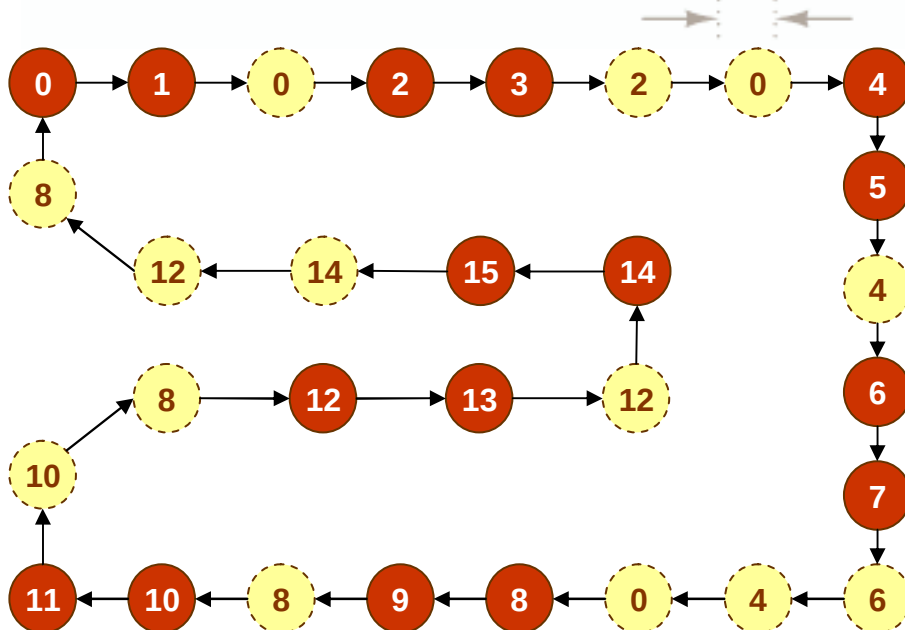
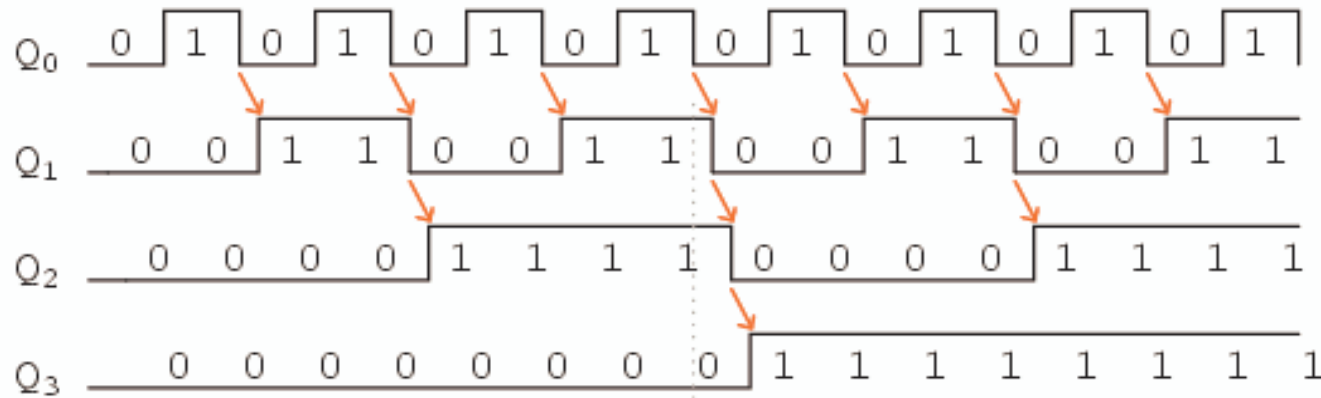
MÁQUINAS DE ESTADOS

Contador Assíncrono Crescente/Decrescente



MÁQUINAS DE ESTADOS

Atraso de Propagação num Contador Assíncrono



MÁQUINAS DE ESTADOS

Redução de Estados

- *Dois estados são equivalentes se as operações realizadas durante esses estados podem ser realizadas simultaneamente e os eventos que seguem os intervalos de controle desses estados são os mesmos.*
- *Dois estados são equivalentes se e somente se seus próximos estados e os valores de seus sinais saídas são respectivamente os mesmos para todas as combinações da entradas.*
- *As técnicas de redução de estados consistem em duas etapas:*
 1. *Identificação dos estados equivalentes;*
 2. *Substituição de todos os estados equivalentes por um único estado representativo.*

MÁQUINAS DE ESTADOS

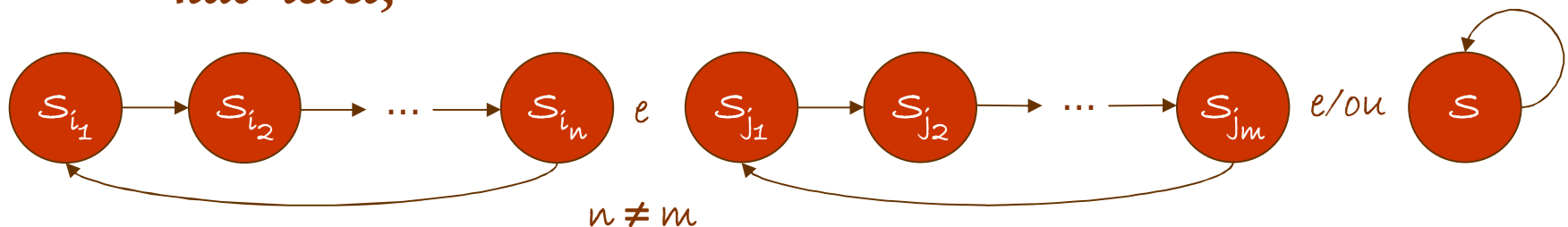
Redução de Estados

- *Os sistemas seqüenciais podem ser classificados em dois tipos:*
 - *Sistemas Reset: São sistemas nos quais o controle sempre retorna para um único estado depois de passar por um número fixo de outros estados;*
 - *Sistemas Não-Reset: São sistemas nos quais o controle não retorna necessariamente para um único estado.*
- *A complexidade de identificação de estados equivalentes num sistema seqüencial depende do seu tipo: reset ou não-reset.*
- *Em sistemas reset, os estados equivalentes são identificados na base do comportamento do próximo estado e do sinais de saída.*
- *Em sistemas não-reset, os estados equivalentes são identificados utilizando o método de redução de Moore.*

MÁQUINAS DE ESTADOS

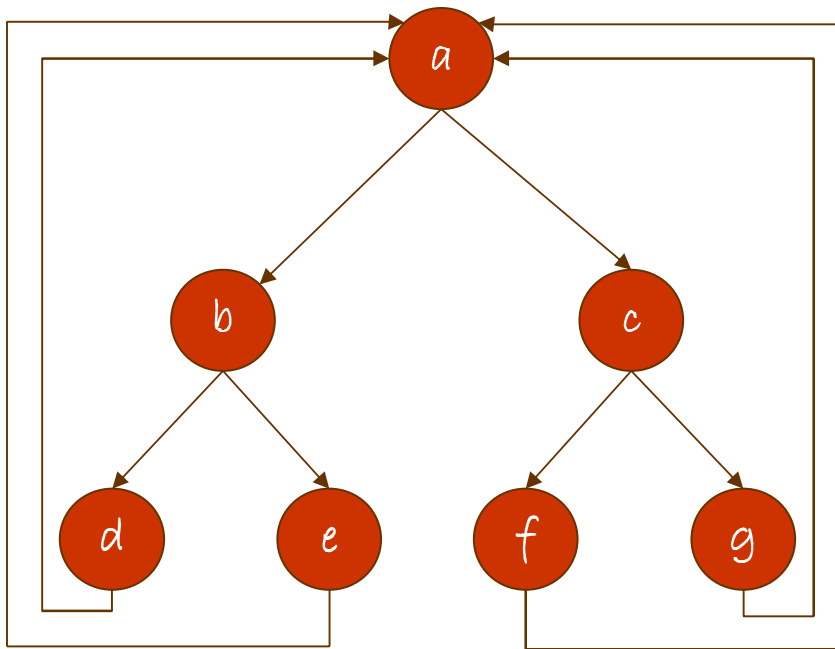
Classificação dos Sistemas Seqüenciais

- A propriedade de reset num sistema seqüencial não sempre simples para identificar.
- A propriedade de não-reset num sistema seqüencial pode ser aparente.
 - Se existir no diagrama de estados do sistema pelo menos duas seqüências fechadas de diferentes comprimentos, então o mesmo é não-reset;
 - Se existir no diagrama de estados do sistema pelo menos um estado cujo próximo estado é ele mesmo, então o mesmo é não-reset;

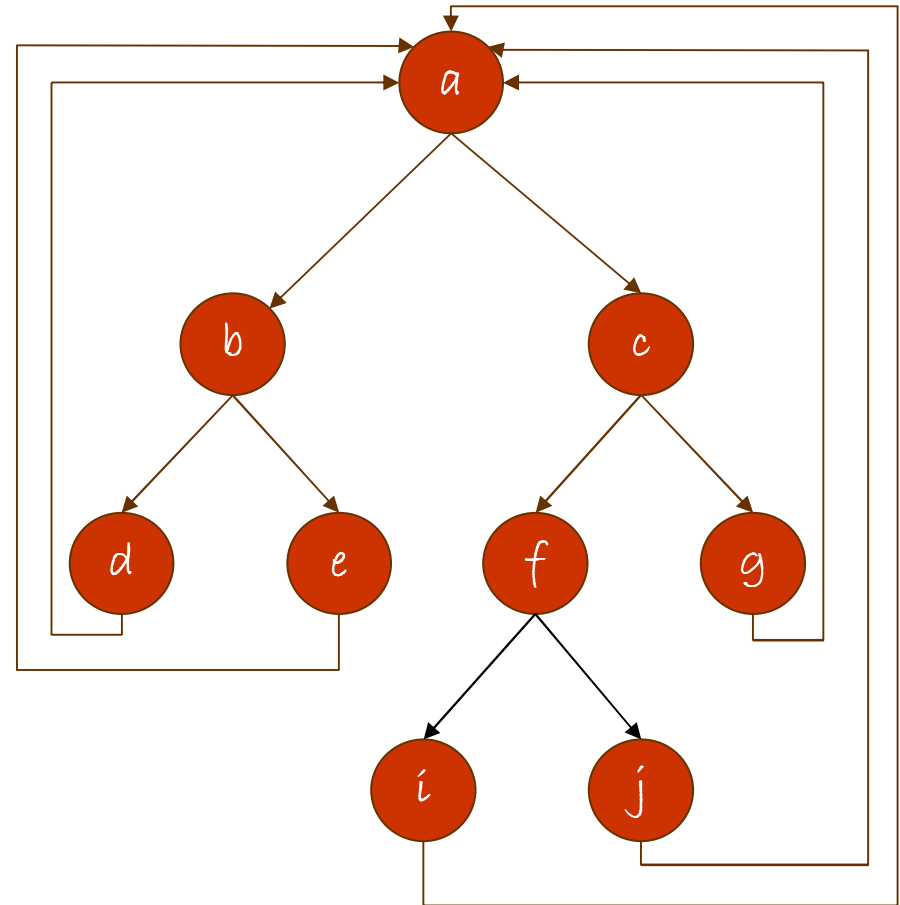


MÁQUINAS DE ESTADOS

Exemplos de Sistemas Reset e Não-Reset



Sistema reset



Sistema não-reset

MÁQUINAS DE ESTADOS

Exemplo de Redução de Sistemas Reset

atual	próximo		saída	
	$E=0$	$E=1$	$E=0$	$E=1$
a	b	c	0	0
b	d	e	0	0
c	f	g	0	0
d	a	a	0	0
e	a	a	0	1
f	a	a	0	0
g	a	a	0	1

atual	próximo		saída	
	$E=0$	$E=1$	$E=0$	$E=1$
a	b	c	0	0
b	d	e	0	0
c	d	e	0	0
d	a	a	0	0
e	a	a	0	1

- Como $\text{próximo}(d) = \text{próximo}(f)$ e $\text{saída}(d) = \text{saída}(f) \Rightarrow d \equiv f$;
- Como $\text{próximo}(e) = \text{próximo}(g)$ e $\text{saída}(e) = \text{saída}(g) \Rightarrow e \equiv g$;
- Substituir os estados d e f por d e os estados e e g por e.

MÁQUINAS DE ESTADOS

Exemplo de Redução de Sistemas Não-Reset

atual	próximo		saída	
	$E = 0$	$E = 1$	$E = 0$	$E = 1$
a	e	b	0	0
b	a	d	0	1
c	f	d	0	0
d	c	b	0	1
e	c	a	0	0
f	a	c	0	0

classe atual	estado	Próxima classe	
		$E = 0$	$E = 1$
C_1	a	C_1	C_2
	c	C_1	C_2
	e	C_1	C_1
	f	C_1	C_1
C_2	b	C_1	C_2
	d	C_1	C_2

- Os estados a, c, e e f caíam na classe de saída (0,0), chamada aqui de C_1 ;
- Os estados b e d caíam na classe de saída (0,1), chamada aqui de C_2 .

MÁQUINAS DE ESTADOS

Exemplo de Redução de Sistemas Não-Reset

classe atual	estado	Próxima classe	
		$E = 0$	$E = 1$
C_1	a	C_1	C_2
	c	C_1	C_2
	e	C_1	C_1
	f	C_1	C_1
C_2	b	C_1	C_2
	d	C_1	C_2

classe atual	estado	Próxima classe	
		$E = 0$	$E = 1$
C_1	a	C_3	C_2
	c	C_3	C_2
C_2	b	C_1	C_2
	d	C_1	C_2
C_3	e	C_1	C_1
	f	C_1	C_1

- Os estados a e c caíam na classe de saída (0,0) com próxima classe (C_1, C_2), chamada aqui de C_1 ;
- Os estados e e f caíam na classe de saída (0,0) com próxima classe (C_1, C_1), chamada aqui de C_3 .

MÁQUINAS DE ESTADOS

Exemplo de Redução de Sistemas Não-Reset

classe atual	estado	Próxima classe	
		$E = 0$	$E = 1$
C_1	a	C_3	C_2
	c	C_3	C_2
C_2	b	C_1	C_2
	d	C_1	C_2
C_3	e	C_1	C_1
	f	C_1	C_1

estado atual	Próximo estado		saída	
	$E = 0$	$E = 1$	$E = 0$	$E = 1$
a	e	b	0	0
b	a	b	0	1
c	a	a	0	0

- $C_1 = \{a, c\} \Rightarrow a \equiv c$; a classe C_1 é representada pelo estado a ;
- $C_2 = \{b, d\} \Rightarrow b \equiv d$; a classe C_2 é representada pelo estado b ;
- $C_3 = \{e, f\} \Rightarrow e \equiv f$; a classe C_3 é representada pelo estado e ;

MÁQUINAS DE ESTADOS

Designação de Estados

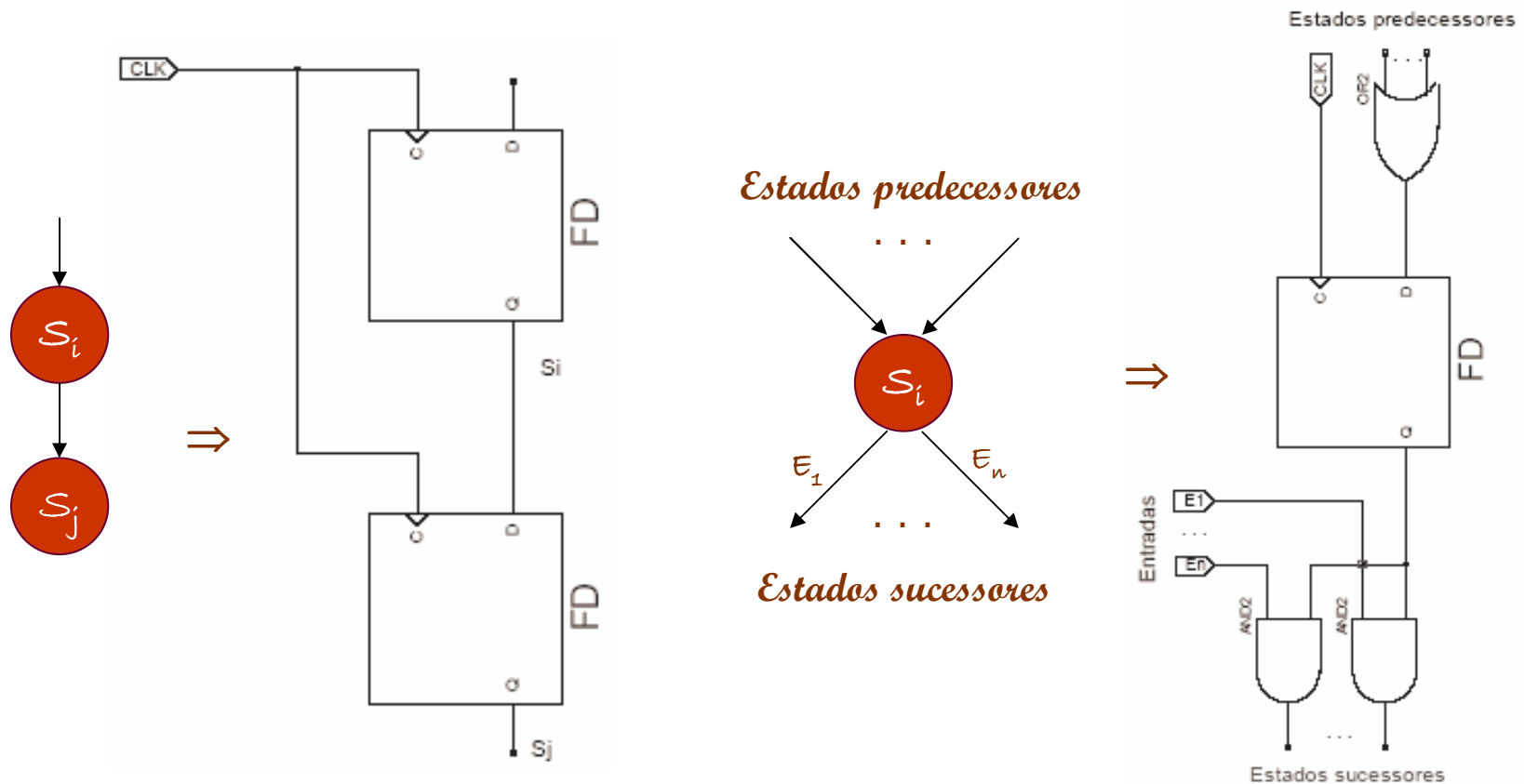
- *A codificação de estados afeta a complexidade e o custo da lógica de controle da máquina de estados.*
- *Exemplo: Considere as atribuições*
 $(a = 0, b = 2, c = 1, d = 3) \Rightarrow 5 \text{ portas AND} + 2 \text{ portas OR}$
 $(a = 0, b = 3, c = 1, d = 2) \Rightarrow 4 \text{ portas AND} + 3 \text{ portas OR}$

estado atual	Próximo estado		Saída, S	
	E = 0	E = 1	E = 0	E = 1
a	a	b	0	0
b	c	b	0	1
c	a	d	1	0
d	c	b	1	1

MÁQUINAS DE ESTADOS

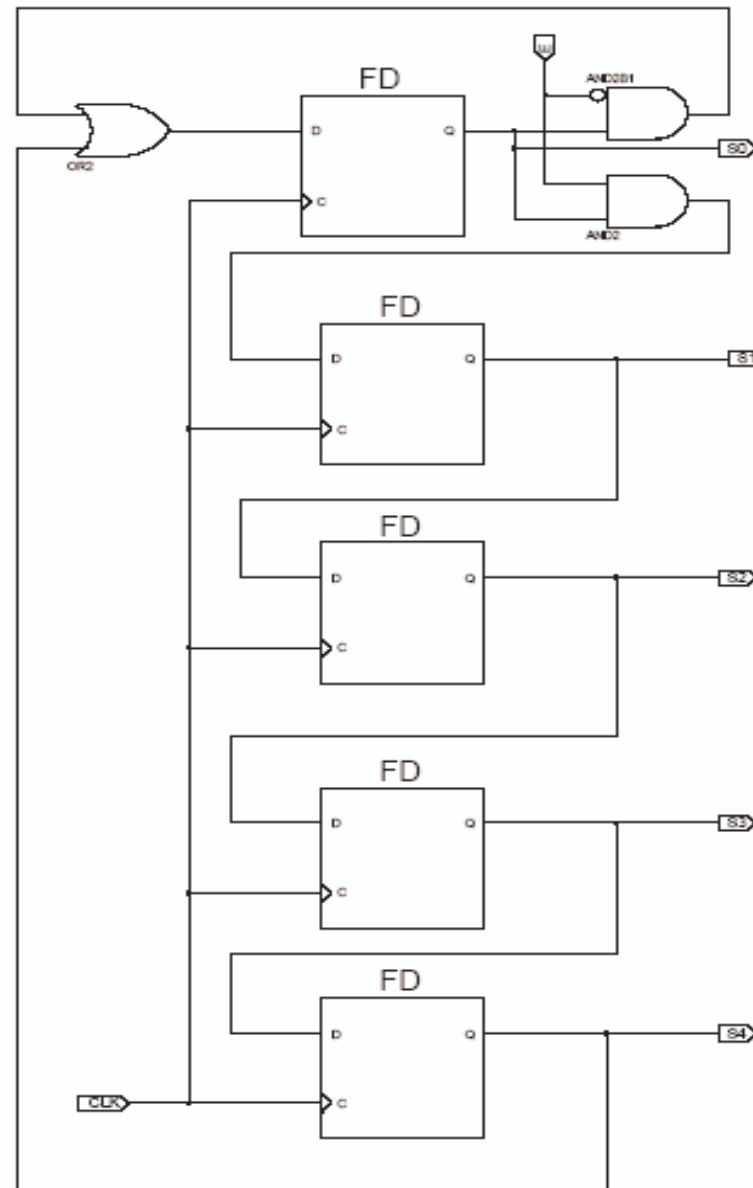
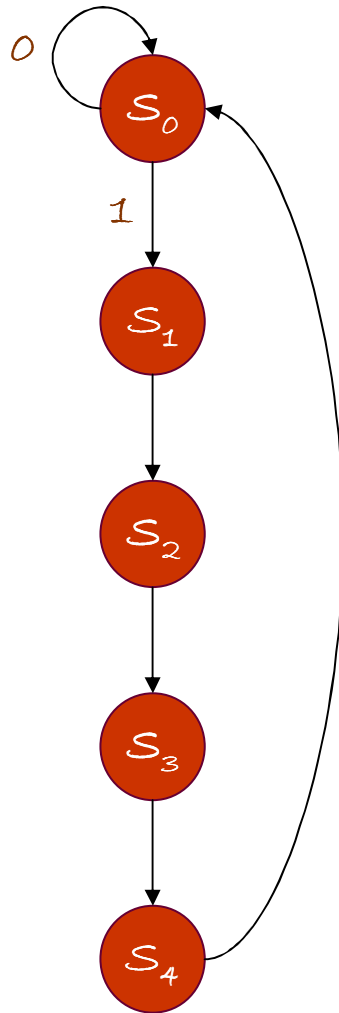
Designação One-Hot Bit

- *Esta designação associa um flip-flop distinto a cada estado.*
- *É interessante somente quando o número de estados é pequeno*



MÁQUINAS DE ESTADOS

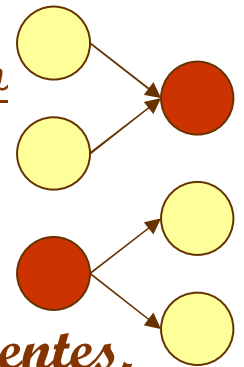
Exemplo



MÁQUINAS DE ESTADOS

As Regras de Armstrong e Humphrey

- A identificação da melhor atribuição de estados é um problema NP-Completo.
- Uma “boa” atribuição respeitaria as regras de Armstrong e Humphrey.
- Regra 1: Dois ou mais estados que têm o mesmo próximo estado devem receber códigos adjacentes;
- Regra 2: Dois ou mais estados que são os próximos estados de um mesmo estado devem receber códigos adjacentes.
- Códigos adjacentes são diferentes em somente uma posição.
- Por exemplo, os códigos 01010 e 11010 são adjacentes mas os códigos 01010 e 11110 não são adjacentes.



MÁQUINAS DE ESTADOS

Exemplo

estado atual	Próximo estado		Saída, S	
	E = 0	E = 1	E = 0	E = 1
a	a	b	0	0
b	c	b	0	1
c	a	d	1	0
d	c	b	1	1

estado atual	predecessores
a	a, c
b	a, b, d
c	b, d
d	d

- Regra 1:
 - $\{(a, c)\}$, $\{(a, b), (a, d), (b, d)\}$ e $\{(b, d)\}$ *devem ter códigos adjacentes.*
 - *Note que (b, d) aparece duas vezes;*
 - *Como três estados não podem ter códigos adjacentes dois a dois, a regra pode ser respeitada somente duas duplas de $\{(a, b), (a, d), (b, d)\}$.*

MÁQUINAS DE ESTADOS

Exemplo

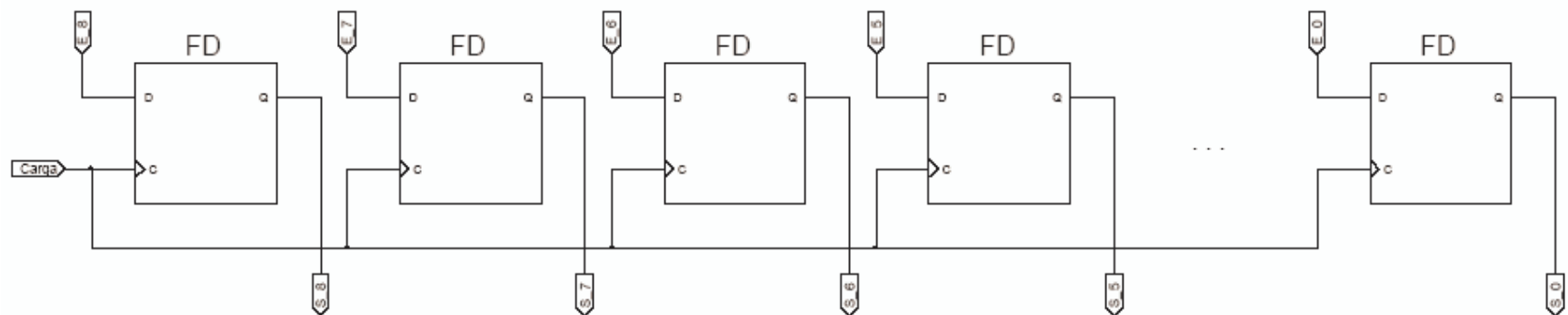
- Regra 2:
 - $\{(a, b)\}, \{(c, b)\}, \{(a, d)\}$ e $\{(c, b)\}$ devem ter códigos adjacentes; Note que (c, b) aparece duas vezes;
- Uma sugestão de “boa” atribuição seria obtida satisfazendo o máximo de duplas.
- Por exemplo, a atribuição $a \equiv 00$, $b \equiv 11$, $c \equiv 01$ e $d \equiv 10$ satisfaz todas as duplas exceto (a, b) .
- É recomendável satisfazer as duplas com maior número de ocorrência como as duplas (b, d) e (c, b) .

estado atual	Próximo estado		Saída, S	
	E = 0	E = 1	E = 0	E = 1
a	a	b	0	0
b	c	b	0	1
c	a	d	1	0
d	c	b	1	1

MEMÓRIAS

Registradores

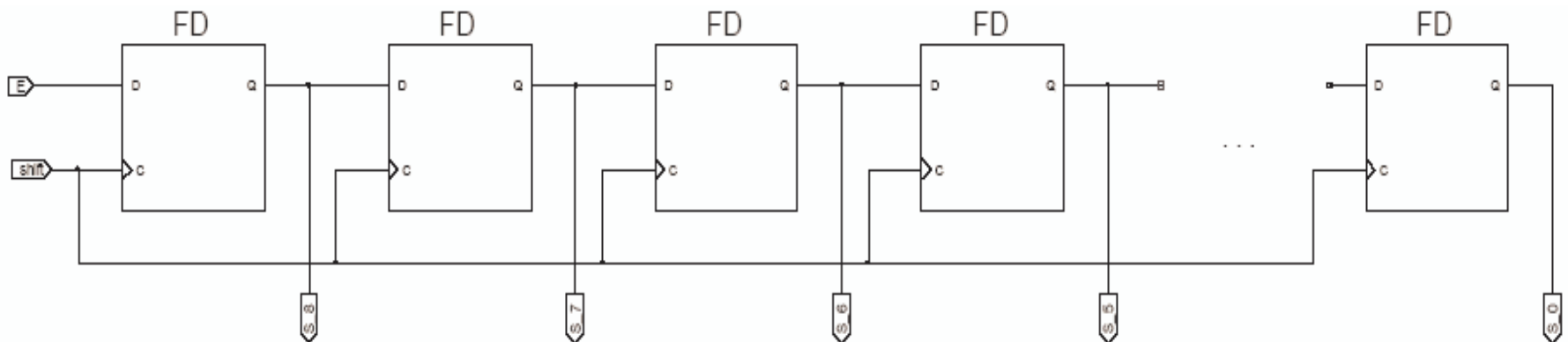
- *Um registrador é geralmente usado para armazenar uma palavra de n bits.*
- *Praticamente, existem inúmeros tipos de registradores, com diferentes aplicações.*
- *Basicamente, um registrador é formado por um conjunto de flip-flops em cascata, possibilitando a leitura e escrita de todas as células em série ou em paralelo.*
- *Um registrador paralelo-paralelo é muito simples:*



MEMÓRIAS

Registradores de Deslocamento

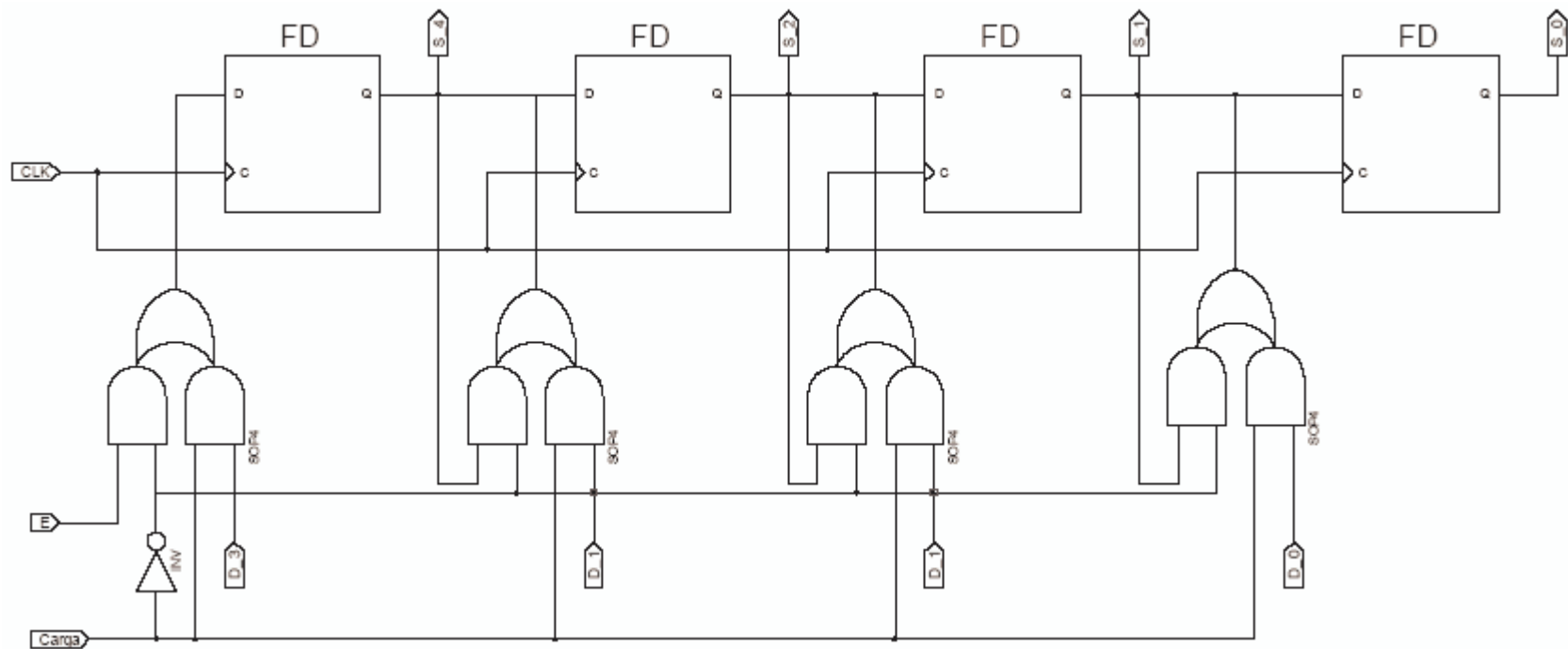
- *Existem dois tipos de registradores de deslocamento:*
 - *Entrada em série e saída em paralelo ou shift-in;*
 - *Entrada em paralelo e saída em série ou shift-out.*
- *Num registrador shift-in de n bits, um bit E é deslocado para o flip-flop seguinte a cada transição de clock.*
- *Só depois de n períodos de clock que o dado estará armazenado no registrador.*



MEMÓRIAS

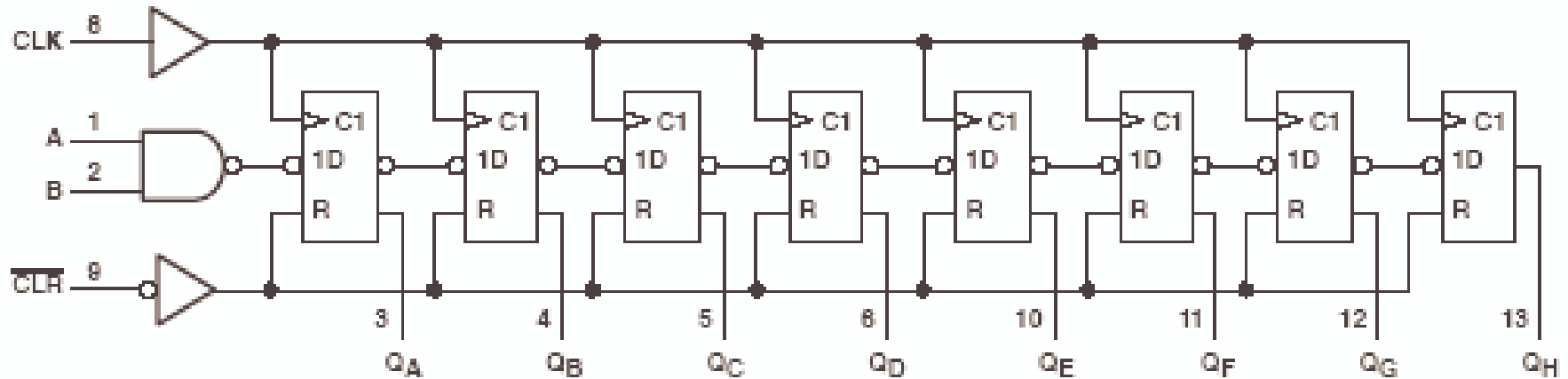
Registadores de Deslocamento

- O registrador shift-out deve permitir primeiro a carga do dado (modo de carga);
- Num registrador shift-out em modo de shift, o conteúdo de um flip-flop é deslocado para o seguinte à cada transição de clock.



MEMÓRIAS

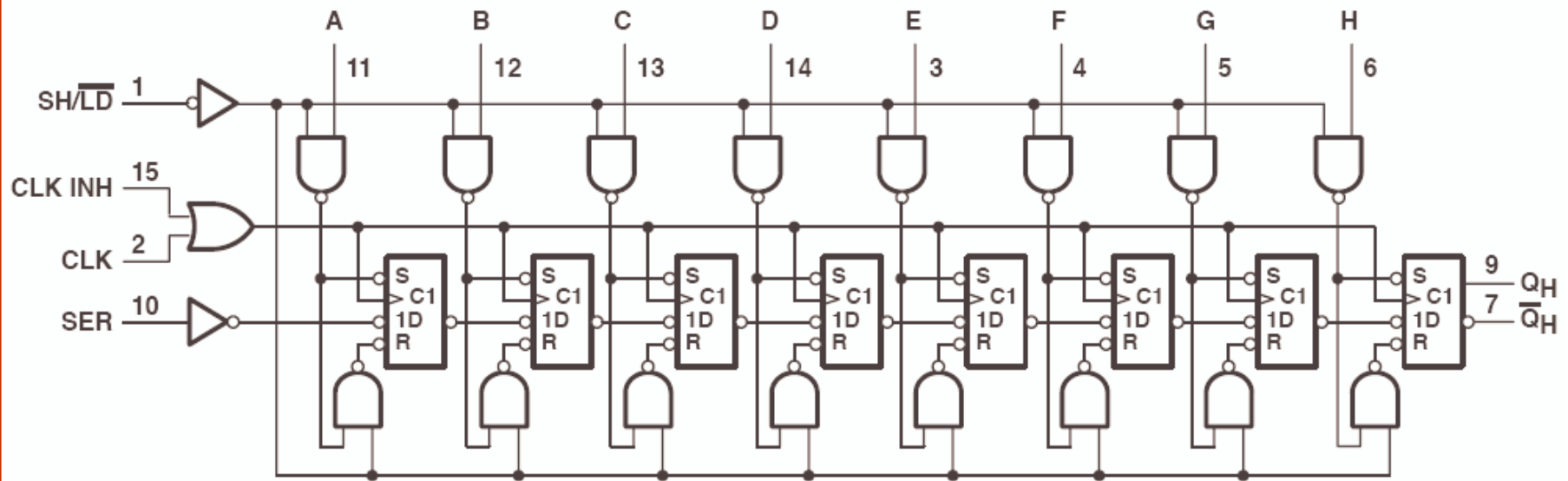
Registradores Comercias



Inputs				Outputs		
$\overline{CLR}$	CLK	A	B	Q_A	Q_B	Q_H
0	X	X	X	0	0	0
1	0	X	X	Q_{A0}	Q_{B0}	Q_{H0}
1	$\uparrow$	1	1	1	Q_{An}	Q_{Gn}
1	$\uparrow$	0	X	0	Q_{An}	Q_{Gn}
1	$\uparrow$	X	0	0	Q_{An}	Q_{Gn}

74-164:
8-Bit Parallel out
Serial Shift Register

MEMÓRIAS



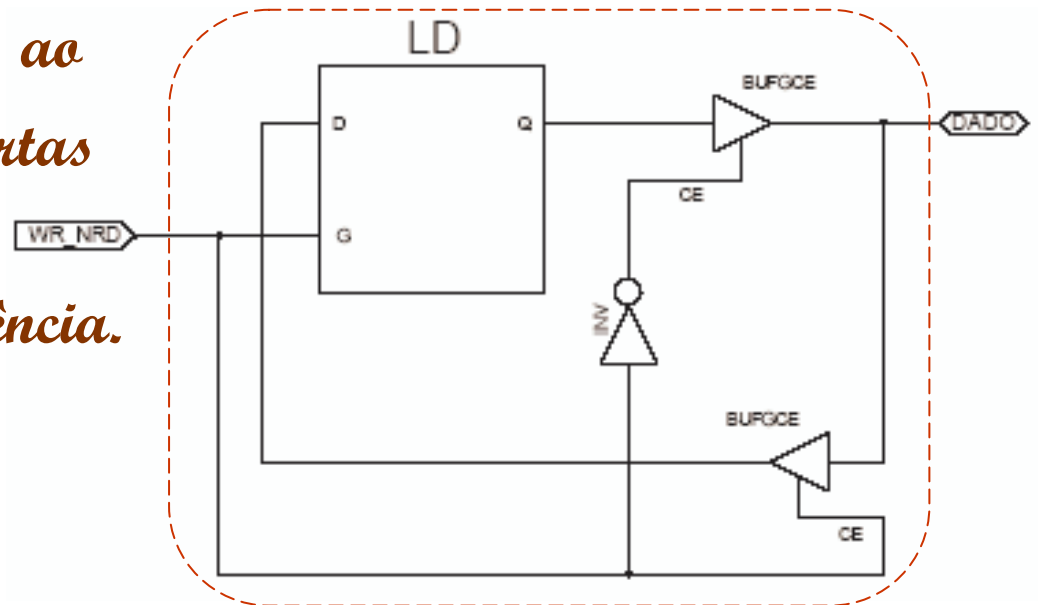
Inputs					Internal outputs		Out Q_H
$\overline{SH/LD}$	CLK INH	CLK	SER	PAR A ... H	Q_A	Q_B	
0	X	X	X	A ... H	A	B	H
1	0	L	X	X	Q_{A0}	Q_{B0}	Q_{H0}
1	0	$\uparrow$	1	X	1	Q_{An}	Q_{Qn}
1	0	$\uparrow$	0	X	0	Q_{An}	Q_{Qn}
1	1	X	X	X	Q_{A0}	Q_{B0}	Q_{H0}

74-165:
8-Bit
Parallel
in
Serial
Shift
Register

MEMÓRIAS

Memória RAM

- *É capaz de manter os dados armazenados apenas enquanto conectada.*
- *Uma célula de RAM armazena um bit e permite tanto a leitura quanto a modificação ou escrita.*
- *Como o barramento bidirecional de dado é geralmente compartilhado por várias células, a entrada e saída do latch D são conectadas ao barramento através de portas tri-state (0, 1 e Z).*
- *Z representa alta-impedência.*



A solid orange square is located in the top-left corner. A horizontal orange line extends from the right side of this square across the top of the page. A vertical orange line extends downwards from the bottom of the square, running along the left margin.

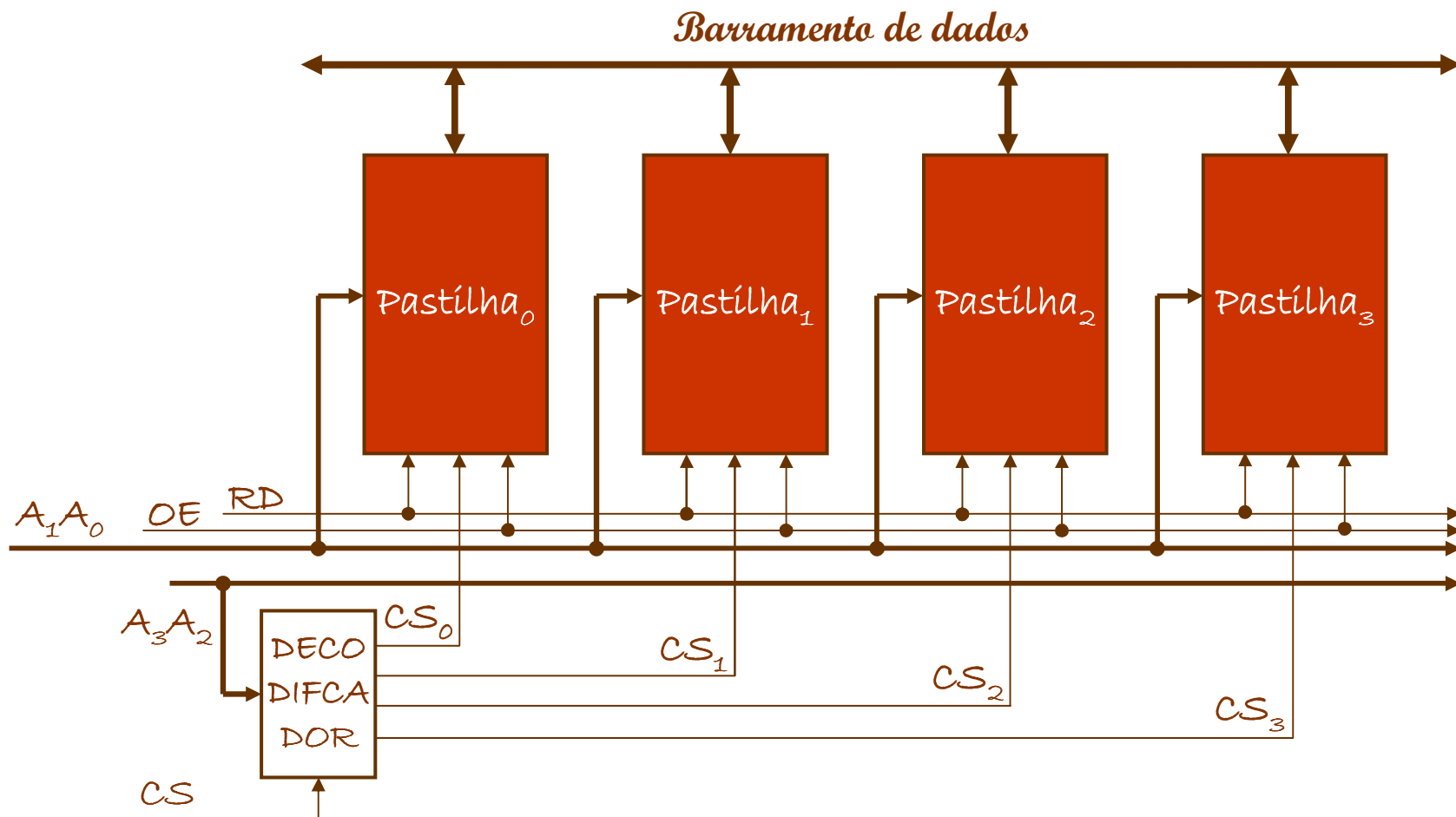
MEMÓRIAS

MEMÓRIAS

- A memória tem oito sinais de entradas: 3 bits de dados para escrita (I_0, I_1, I_2); 2 bits de endereço (A_0, A_1); 3 bits de controle (CS, RD, OE);
- A memória tem três sinais de saída: 3 bits de dados lidos (D_0, D_1, D_2).
- os bits (I_0, I_1, I_2) e (D_0, D_1, D_2) formam o barramento de dados que é bidirecional.
- $CS = 1$ e $RD = 1$ indica uma operação de leitura. Neste caso, se $OE = 1$ permite liberar a saída dos flip-flops da palavra de endereço A_0A_1 no barramento de dados;
- $CS = 1$ e $RD = 0$ indica uma operação de escrita. Neste caso, os dados I_0, I_1, I_2 do barramento são carregados nos Flip-flops da palavra de endereço A_0A_1 .
- $CS = 0$ indica que pastilha de memória não está selecionada nem para leitura nem para escrita.

MEMÓRIAS

Organização de Memória 16×3bits



CONTROLADORES MICRO-PROGRAMADOS

Definição de Controladores Micro-Programados

- São usados para implementar sistemas seqüenciais complexos necessitando muitos estados.
- O conjunto de flip-flops é substituído por uma memória chamada memória de controle;
- A memória de controle armazena uma seqüência de micro-instruções chamada micro-programa de controle;
- Qualquer transição de estado é feita com base nas micro-instrução corrente junto com os dados de entrada do sistema;
- O micro-programa é carregado na memória de controle a partir do endereço 0;
- A cada ciclo, a micro-instrução corrente é lida da memória de controle utilizando o endereço calculado no ciclo anterior e em seguida executada permitindo a transição de estado.

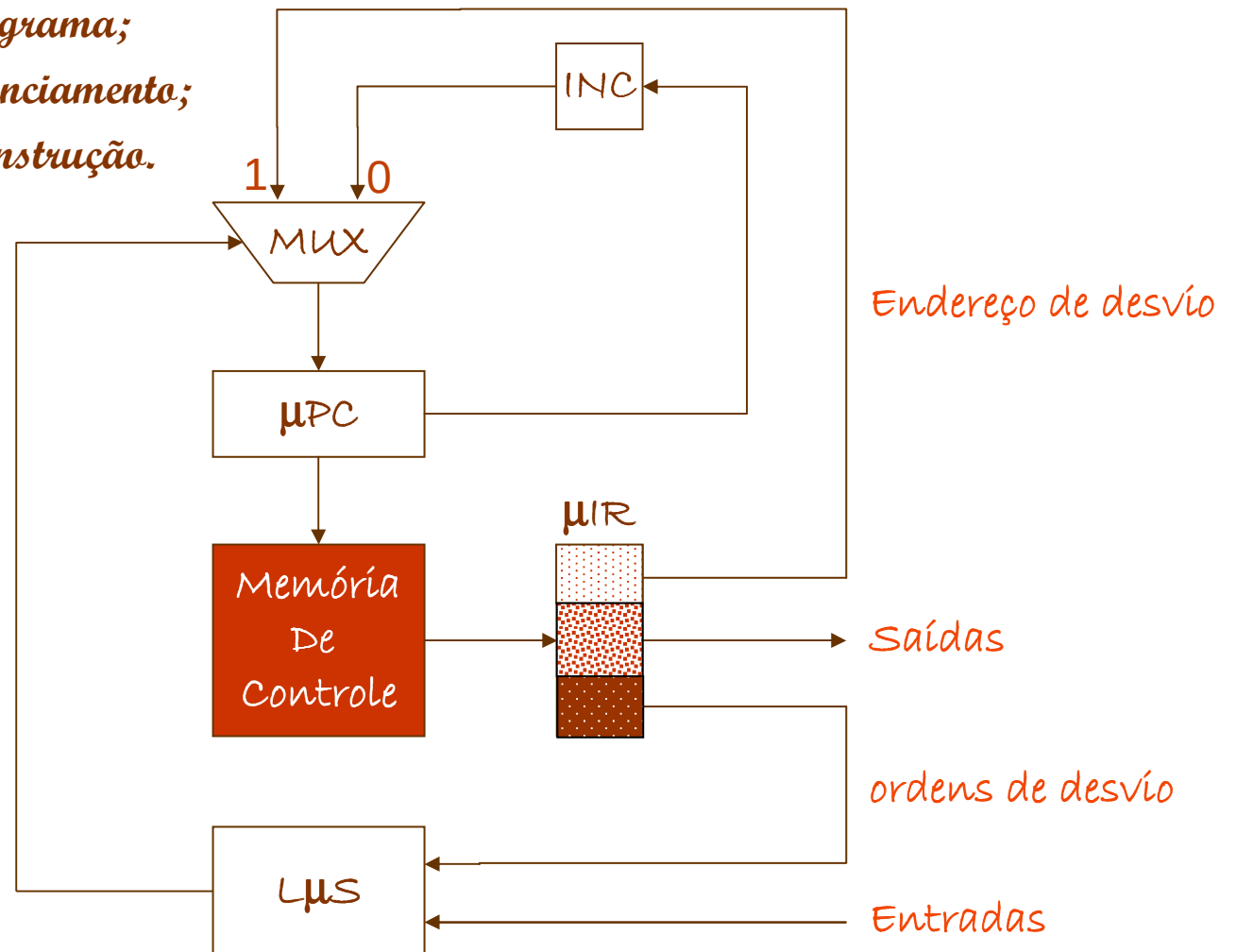
CONTROLADORES MICRO-PROGRAMADOS

Arquitetura de Controlador Micro-Programado

μ PC: Contador de μ -programa;

L μ S: Lógica de μ -seqüenciamento;

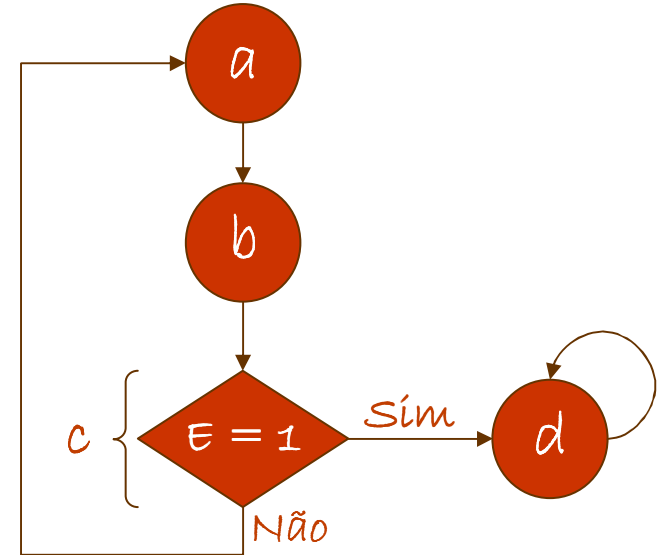
μ IR: Registrador de μ -instrução.



CONTROLADORES MICRO-PROGRAMADOS

Exemplo de Controlador Micro-Programado

estado atual	Próximo estado		Saídas	
	$E = 0$	$E = 1$	S_1	S_2
a	b	b	1	0
b	c	c	0	1
c	a	d	0	0
d	d	d	0	0



Estado	Endereço de mem.	Saídas		Controle de desvio		Endereço de desvio	Mícro-programa
		S_1	S_2	C	u		
a	000	1	0	0	0	000	000: 10000000
b	001	0	1	0	0	000	001: 01000000
c	010	0	0	1	0	100	010: 00101000
	011	0	0	0	1	000	011: 00010000
d	100	0	0	0	1	100	100: 00011000