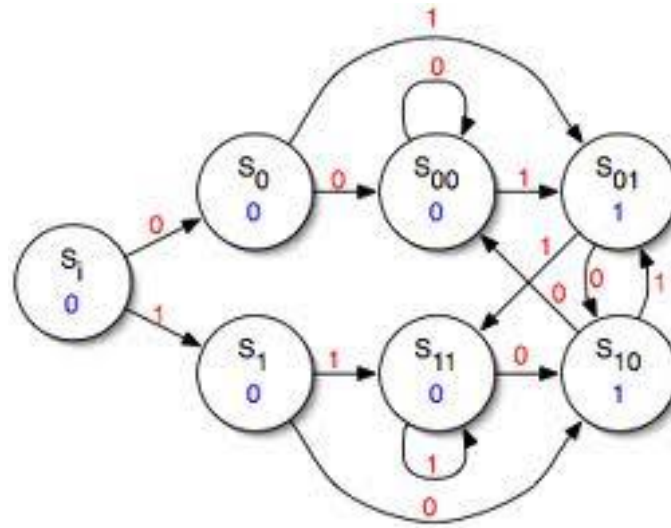


Sistemas Digitais (SD)

Síntese de Circuitos Sequenciais: Definições



■ Na aula anterior:

▶ Contadores síncronos

- Contadores de módulo 2^n
- Projecto de contadores
- Frequência máxima de funcionamento
- Situação de “lock-out”
- Simbologia
- Contador em anel
- Contador Johnson
- Linear feedback shift-register

▶ Contadores assíncronos

- Contadores por pulsação
- Contadores assíncronos vs. síncronos



SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
17/Fev a 21/Fev	Introdução	Sistemas de Numeração	
24/Fev a 28/Fev	CARNAVAL	Álgebra de Boole	P0
02/Mar a 06/Mar	Elementos de Tecnologia	Funções Lógicas	VHDL
9/Mar a 13/Mar	Minimização de Funções	Minimização de Funções	L0
16/Mar a 20/Mar	Def. Circuito Combinatório; Análise Temporal	Circuitos Combinatórios	P1
23/Mar a 27/Mar	Circuitos Combinatórios	Circuitos Combinatórios	L1
30/Mar a 03/Abr	Circuitos Sequenciais: Latches	Circuitos Sequenciais: Flip-Flops	P2
06/Abr a 10/Abr	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA
13/Abr a 17/Abr	Caracterização Temporal	Registos	L2
20/Abr a 24/Abr	Contadores	Circuitos Sequenciais Síncronos	P3
27/Abr a 01/Mai	Síntese de Circuitos Sequenciais Síncronos	Síntese de Circuitos Sequenciais Síncronos	L3
04/Mai a 08/Mai	Exercícios	Memórias	P4
11/Mai a 15/Mai	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Microprograma	L4
18/Mai a 22/Mai	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	P5
25/Mai a 29/Mai	P6	P6	L5

Teste 1

■ Tema da aula de hoje:

- ▶ Definição de circuito sequencial síncrono
- ▶ Máquinas de Mealy e de Moore
- ▶ Especificação de circuitos sequenciais síncronos:
 - Diagrama de estados
- ▶ Projecto de circuitos sequenciais síncronos:
 - Codificação dos estados
 - Tabela de transição de estados
 - Determinação das funções lógicas de saída e estado seguinte

□ Bibliografia:

- **M. Mano, C. Kime:** Secções 5.4 a 5.7
- **G. Arroz, J. Monteiro, A. Oliveira:** Secção 7.1 a 7.4

■ Dicotomia: circuito combinatório vs circuito sequencial

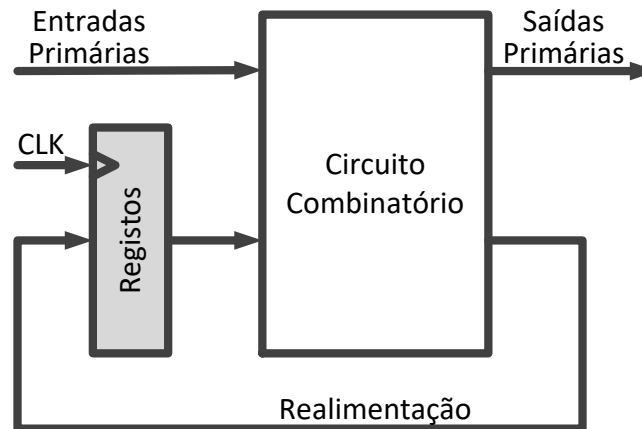
▶ Circuito Combinatório

- O valor da saída depende apenas do valor nas entradas nesse instante

▶ Circuito Sequencial

- O valor da saída depende do valor actual nas entradas, bem como da história anterior dos estados do circuito
 - Como? → através de elementos de memória (ex: latches e flip-flops)
- Podem ser divididos em:
 - **Síncronos**: o sinal de relógio sincroniza toda a actividade do circuito
 - **Assíncronos**: não usam sinal de relógio – as transições de estado ocorrem sempre que há uma alteração nas entradas do circuito

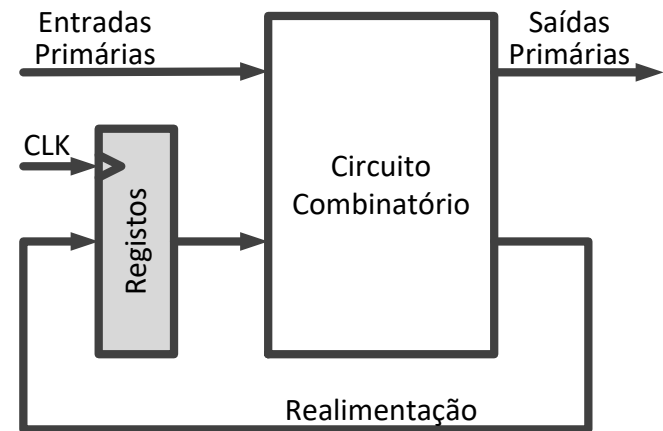
■ Circuito Sequencial Síncrono



► Duas componentes:

- Bloco de lógica puramente combinatória
 - Implementa as funções necessárias para que o circuito tenha a transição entre estados pretendida
- Elementos de memória, controlados por um sinal de relógio
 - Mantém o estado do circuito ao longo do tempo

■ Máquina de Estados

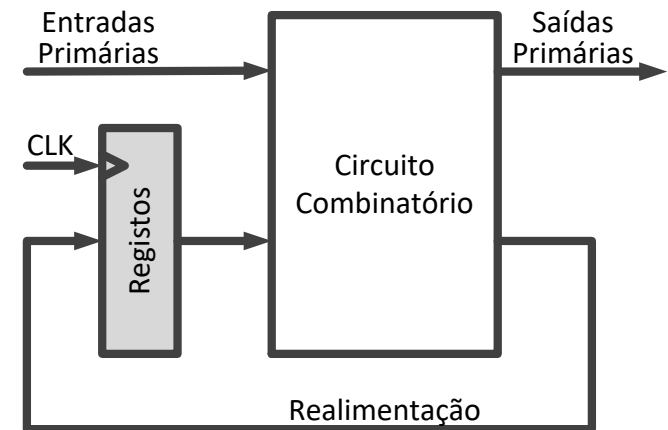


► Definida por:

- As possíveis combinações de entradas que controlam a máquina
- As possíveis combinações de saídas que são geradas pela máquina
- O conjunto de estados da máquina
- A função de transição entre estados
- A função de saída
- O estado inicial

■ Máquina de Estados

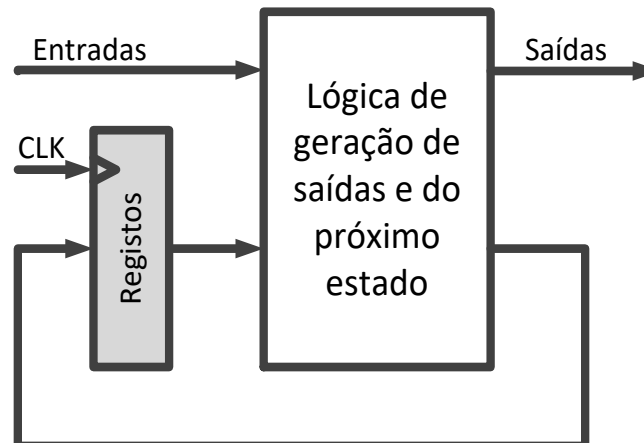
- ▶ O valor dos elementos de memória (FFs) define o estado da máquina em cada instante
- ▶ O circuito combinatório define:
 - a função de transição entre estados
 - a função de saída
- ▶ Inicialização:
 - Entrada de controlo (síncrona ou assíncrona) que conduz todos os FFs para o estado inicial.



■ Máquinas de Moore vs. Máquinas de Mealy

► As máquinas de estado síncronas podem ser divididas em:

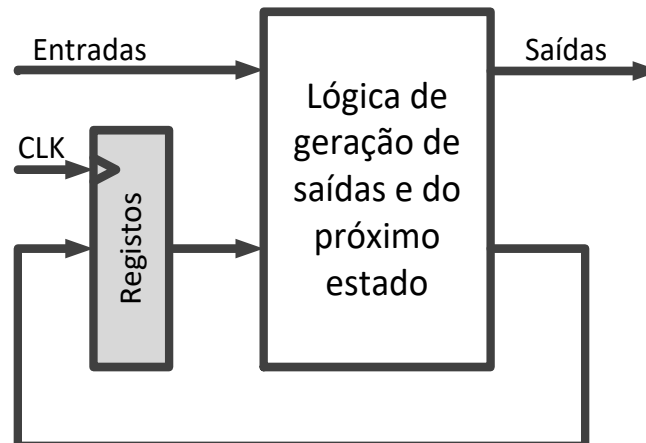
- **Máquinas de Moore:** a saída depende *apenas* das variáveis de estado actuais;
- **Máquinas de Mealy:** a saída é função das variáveis de estado actuais e do valor das entradas presentes no circuito



■ Máquinas de Moore vs. Máquinas de Mealy

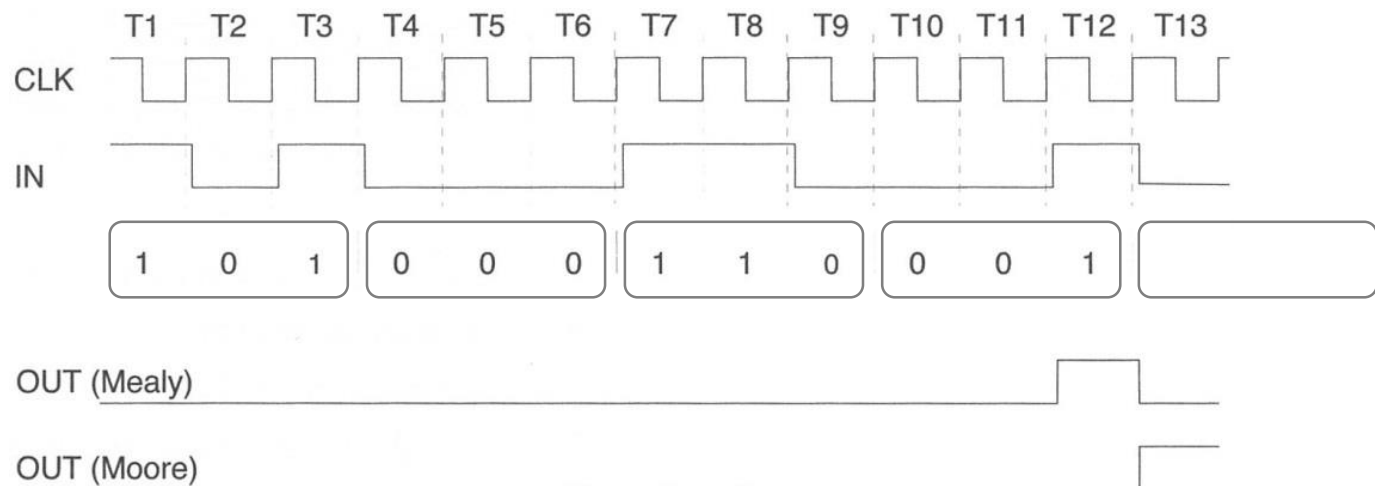
► Do ponto de vista da saída...

- **Máquinas de Moore:** é necessário esperar pelo próximo flanco de relógio para alterar o valor da saída em resposta a uma alteração da entrada;
- **Máquinas de Mealy:** as saídas mudam imediatamente em resposta a uma alteração das entradas.



■ Exemplo – Detector de Paridade (1)

- ▶ Pretende-se enviar dados por uma linha, em grupos de 3 bits. A linha está sujeita a ruído, pelo que se implementou um protocolo de detecção de erros que garante que cada grupo de 3 bits tem um número par de bits a 1.
- ▶ O circuito sequencial pretendido deverá assinalar na sua saída sempre que ocorrer um erro de transmissão, identificado por um número ímpar de bits com valor lógico '1' num grupo de 3 bits



■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

■ Especificação formal:

- Diagrama de estados
- Fluxograma

■ Simplificação da especificação

■ Projecto:

1. Codificação dos estados
2. Tabelas de transição de estados
3. Determinação das funções lógicas de saída e estado seguinte

■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

■ Especificação formal:

- Diagrama de estados
- Fluxograma

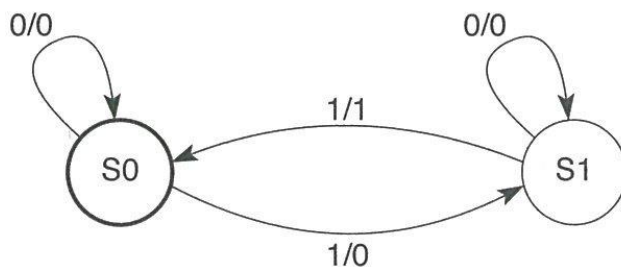
■ Simplificação da especificação

■ Projecto:

1. Codificação dos estados
2. Tabelas de transição de estados
3. Determinação das funções lógicas de saída e estado seguinte

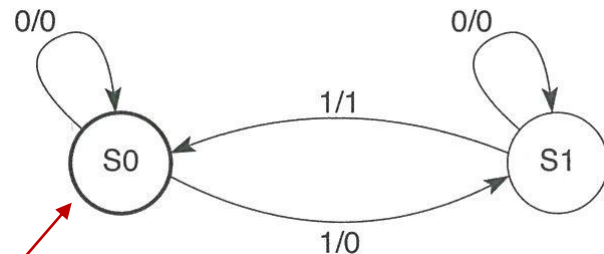
■ Diagrama de Estados

- Representa, de forma gráfica, cada estado da máquina (círculo)

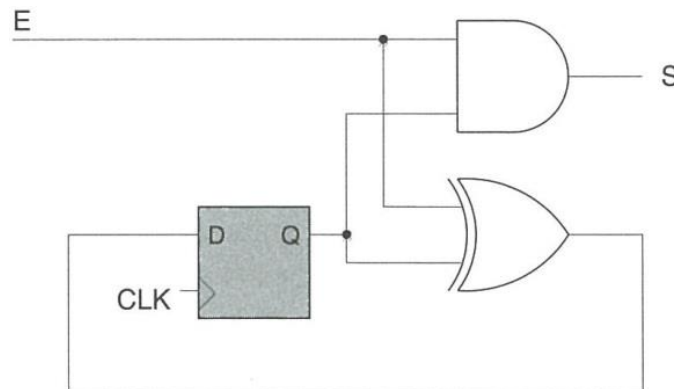
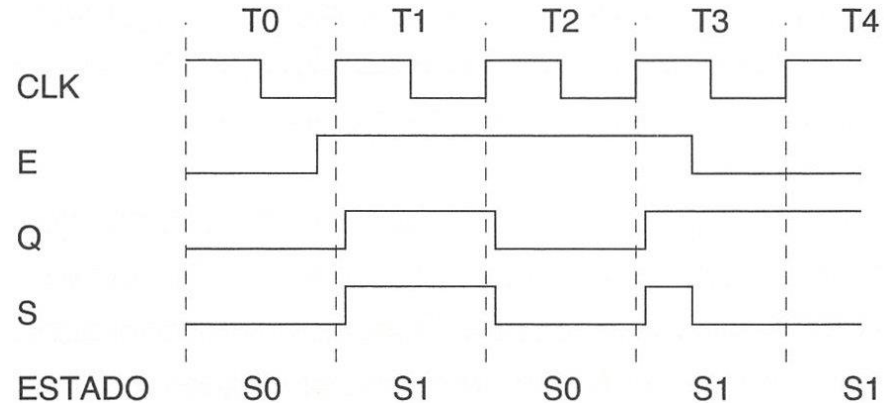


- Para cada combinação de estado e valor nas entradas está associada uma transição (arco) que aponta para o estado para onde a máquina transita na presença dessa combinação.
- Cada transição apresenta:
 - a combinação de entradas que lhe corresponde
 - o valor da saída (apenas as máquinas de Mealy)

■ Exemplo – máquina de Mealy:

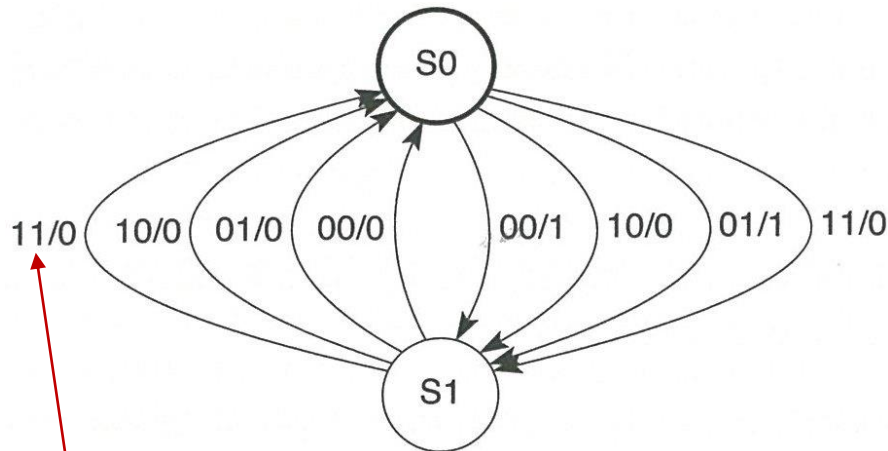


Estado inicial

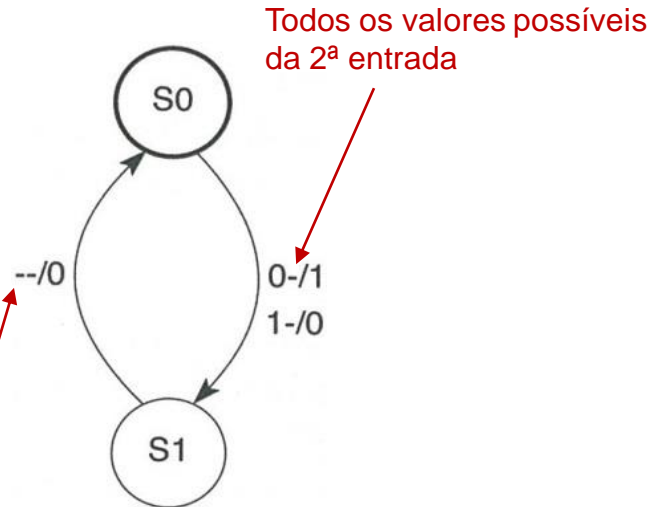


Veremos como sintetizar
este circuito depois...

■ Diagrama de Estados – outros exemplos (Mealy)



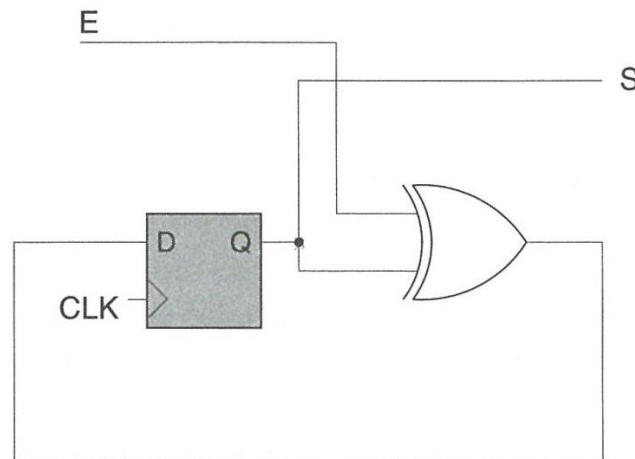
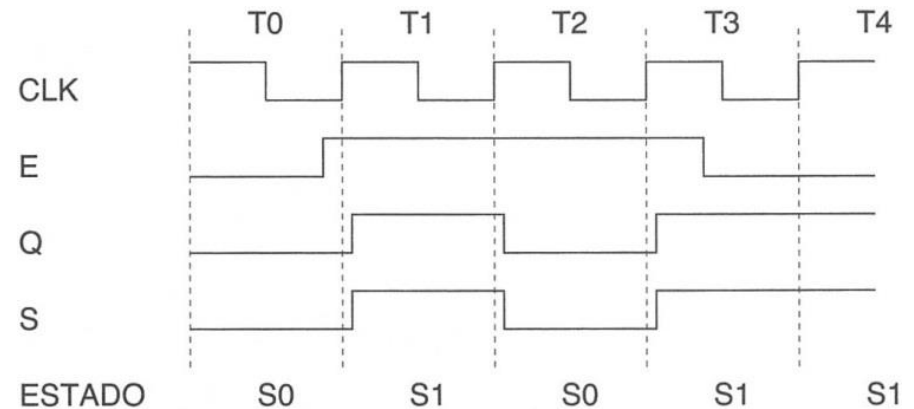
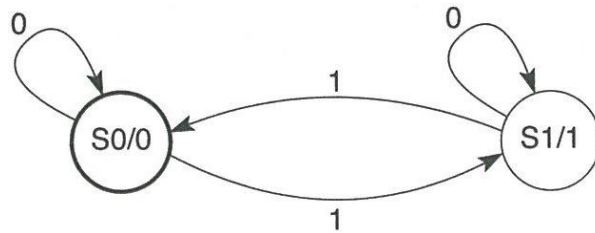
Duas entradas / Uma saída



Todos os valores possíveis das 2 entradas

Todos os valores possíveis da 2ª entrada

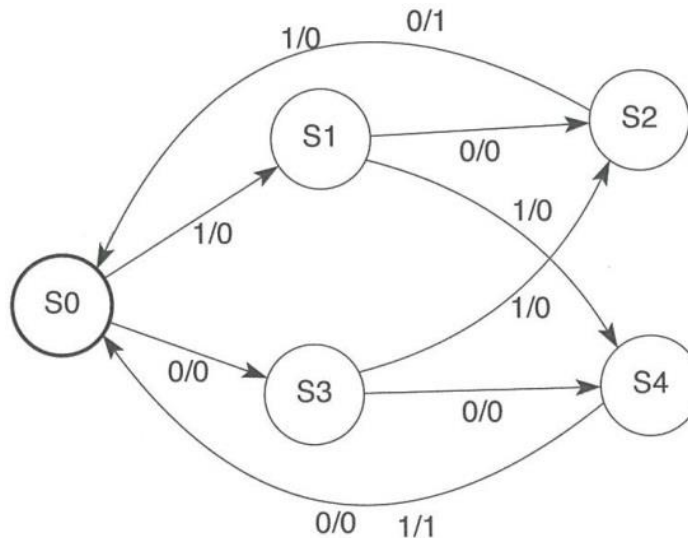
■ Exemplo – máquina de Moore:



Veremos como sintetizar
este circuito depois...

■ Exemplo – Detector de Paridade (2)

- ▶ Pretende-se enviar dados por uma linha, em grupos de 3 bits. A linha está sujeita a ruído, pelo que se implementou um protocolo de detecção de erros que garante que cada grupo de 3 bits tem um número par de bits a 1.
- ▶ O circuito sequencial pretendido deverá assinalar na sua saída sempre que ocorrer um erro de transmissão, identificado por um número ímpar de bits com valor lógico '1' num grupo de 3 bits



Máquina Mealy
ou
Máquina Moore ???

■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

■ Especificação formal:

- Diagrama de estados
- Fluxograma

← **Vemos depois...**

■ Simplificação da especificação

■ Projecto:

1. Codificação dos estados
2. Tabelas de transição de estados
3. Determinação das funções lógicas de saída e estado seguinte

■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

■ Especificação formal:

- Diagrama de estados
- Fluxograma

■ **Simplificação da especificação**

← **Vemos depois...**

■ Projecto:

1. Codificação dos estados
2. Tabelas de transição de estados
3. Determinação das funções lógicas de saída e estado seguinte

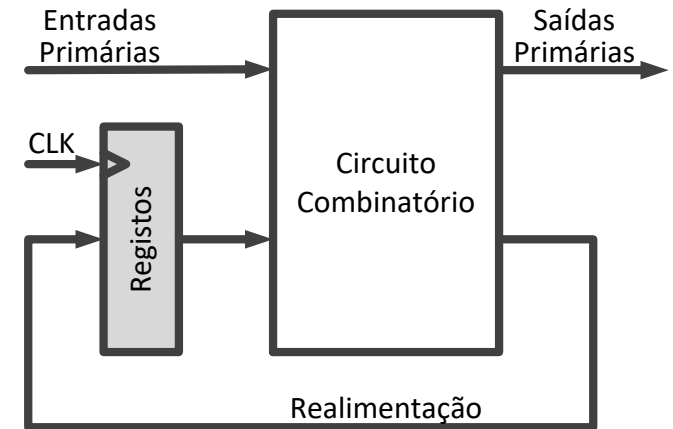
■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

- Especificação formal:
 - Diagrama de estados
 - Fluxograma
- Simplificação da especificação
- Projecto:
 1. **Codificação dos estados**
 2. Tabelas de transição de estados
 3. Determinação das funções lógicas de saída e estado seguinte

■ Codificação dos estados

- ▶ Os estados de um circuito sequencial são guardados nos elementos de memória (Flip-Flops)
- ▶ A este conjunto de Flip-Flops também se pode chamar “Registo de Estado”
- ▶ Cada estado corresponde a uma determinada configuração dos valores das saídas dos Flip-Flops
- ▶ Torna-se necessário definir a codificação a atribuir a cada estado:
 - Várias codificações possíveis
 - O código de cada estado é único.



■ Codificação dos estados

► Método 1: codificação usando o código binário

- Método mais eficiente, pois usa o menor número de Flip-Flops possível
- Menor número de funções para definir o próximo estado → menos portas lógicas (em geral...)
- Considerando a existência de n estados ($E_0, E_1, E_2, \dots, E_{n-1}$), a codificação usando código binário natural irá usar k Flip-Flops, em que k é o menor inteiro igual ou superior a $\log_2(n)$

• Exemplo:

- 6 estados ($E_0, E_1, E_2, E_3, E_4, E_5$)
- $k = \lceil \log_2(6) \rceil = \lceil 2.584 \rceil = 3$ Flip-Flops

Estado	Codificação		
	Q_2	Q_1	Q_0
E_0	0	0	0
E_1	0	0	1
E_2	0	1	0
E_3	0	1	1
E_4	1	0	0
E_5	1	0	1

Existem várias
codificações
possíveis!!!

■ Codificação dos estados

► Método 2: codificação usando um Flip-Flop por estado

- Usa tantos Flip-Flops quanto o número de estados
 - maior número de funções combinatórias a sintetizar!
 - mas cada uma destas funções é, em geral, mais simples...
- Apenas um Flip-Flop tem a saída a '1' em cada instante
- Exemplo:
 - 6 estados ($E_0, E_1, E_2, E_3, E_4, E_5$)
 - 6 Flip-Flops

Estado	Codificação					
	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
E_0	0	0	0	0	0	1
E_1	0	0	0	0	1	0
E_2	0	0	0	1	0	0
E_3	0	0	1	0	0	0
E_4	0	1	0	0	0	0
E_5	1	0	0	0	0	0

■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

- Especificação formal:
 - Diagrama de estados
 - Fluxograma
- Simplificação da especificação
- Projecto:
 1. Codificação dos estados
 - 2. Tabelas de transição de estados**
 3. Determinação das funções lógicas de saída e estado seguinte

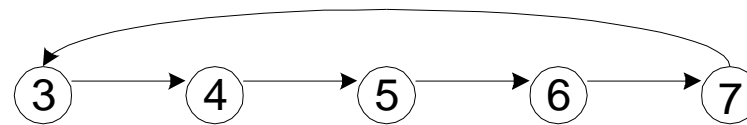
■ Tabela de Transição de Estados

- ▶ Elaborada com base:
 - Análise do diagrama de estados / fluxograma / etc...
 - Codificação atribuída a cada estado
- ▶ Contém uma linha para cada possível combinação do conjunto de bits que inclui: entradas primárias + variáveis de estado
- ▶ Cada linha define:
 - valores das entradas primárias
 - variáveis de estado $Q(n-1)$
 - valores desejados para as variáveis de estado $Q(n)$ no próximo ciclo de relógio
 - valores que as saídas devem tomar

Entradas da Tabela			Saídas da Tabela		
Entrada	Estado Presente		Estado Seguinte		Saída
A	$Q_1(n-1)$	$Q_0(n-1)$	$Q_1(n)$	$Q_0(n)$	B
0	0	0	0	0	0
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	0	0
1	1	0	1	1	0
0	1	1	1	1	1
1	1	1	1	1	1

■ Exemplo: Contador Síncrono de Módulo 5 (PM5)

Diagrama de Estados de Contagem



A existência de 5 estados de contagem impõe, pelo menos, a utilização de 3 FFs

Tabela de Codificação de Estados

Estado	Q2	Q1	Q0
S ₀	0	0	0
S ₁	0	0	1
S ₂	0	1	0
S ₃	0	1	1
S ₄	1	0	0
S ₅	1	0	1
S ₆	1	1	0
S ₇	1	1	1

Tabela de Transição de Estados

Estado Presente (n)			Estado Seguinte (n+1)		
Q2	Q1	Q0	Q2	Q1	Q0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1

■ Projecto de Circuitos Sequenciais Síncronos

► Procedimento:

- Especificação formal:

- Diagrama de estados
- Fluxograma

- Simplificação da especificação

- Projecto:

1. Codificação dos estados
2. Tabelas de transição de estados
- 3. Determinação das funções lógicas de saída e estado seguinte**

■ Determinação das funções lógicas

- Determinação das funções lógicas que definem as **entradas dos flip-flops** (entradas D, J, K, ...) em função das entradas primárias e do estado presente (variáveis de estado)

Exemplo (considerando uma entrada e 2 flip-flops tipo D):

- $D_k(n) = f (A(n-1) , Q_1(n-1) , Q_0(n-1))$ com $k \in \{0,1\}$

Entradas da Tabela			Saídas da Tabela		
Entrada $A(n-1)$	Estado Presente $Q_1(n-1) \quad Q_0(n-1)$		Estado Seguinte $Q_1(n) \quad Q_0(n)$		Saída $B(n-1)$
0	0	0	0	0	0
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	0	0
1	1	0	1	1	0
0	1	1	1	1	1
1	1	1	1	1	1

■ Determinação das funções lógicas

► Determinação das funções lógicas que definem as **saídas** do circuito em função de:

- Estado presente (variáveis de estado) e das entradas primárias → **Máquina de Mealy**
- Estado presente (variáveis de estado) → **Máquina de Moore**

Exemplo (considerando uma entrada e 2 flip-flops tipo D):

- $B(n-1) = f(A(n-1), Q_1(n-1), Q_0(n-1)) \rightarrow$ **Máquina de Mealy**
- $B(n-1) = f(Q_1(n-1), Q_0(n-1)) \rightarrow$ **Máquina de Moore**

Máquina de Mealy

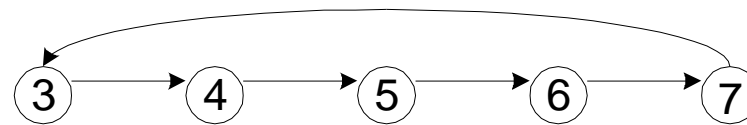
Entradas da Tabela			Saídas da Tabela		
Entrada	Estado Presente		Estado Seguinte		Saída
A(n-1)	Q ₁ (n-1)	Q ₀ (n-1)	Q ₁ (n)	Q ₀ (n)	B(n-1)
0	0	0	0	0	0
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	0	1
1	1	0	1	1	0
0	1	1	1	1	1
1	1	1	1	1	1

Máquina de Moore

Entradas da Tabela			Saídas da Tabela		
Entrada	Estado Presente		Estado Seguinte		Saída
A(n-1)	Q ₁ (n-1)	Q ₀ (n-1)	Q ₁ (n)	Q ₀ (n)	B(n-1)
0	0	0	0	0	0
1	0	0	0	1	0
0	0	1	0	0	0
1	0	1	1	0	0
0	1	0	0	0	0
1	1	0	1	1	0
0	1	1	1	1	1
1	1	1	1	1	1

■ Exemplo: Contador Síncrono de Módulo 5 (PM5)

Diagrama de Estados de Contagem



A existência de 5 estados de contagem impõe, pelo menos, a utilização de 3 FFs

Tabela de Codificação de Estados

Estado	Q2	Q1	Q0
S ₀	0	0	0
S ₁	0	0	1
S ₂	0	1	0
S ₃	0	1	1
S ₄	1	0	0
S ₅	1	0	1
S ₆	1	1	0
S ₇	1	1	1

Tabela de Transição de Estados

Estado Presente (n)			Estado Seguinte (n+1)		
Q2	Q1	Q0	Q2	Q1	Q0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1

■ Exemplo: implementação com FFs tipo D

Tabela de Excitação do FF D

Q(n+1)	D(n)
0	0
1	1

Tabela de Transição de Estados

Estado Presente (n)			Estado Seguinte (n+1)		
Q2	Q1	Q0	Q2(n+1)	Q1(n+1)	Q0(n+1)
			D2(n)	D1(n)	D0(n)
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1

Mapas de Karnaugh por Entrada de FFs

$$D_2 = \overline{Q_2} + \overline{Q_1} + \overline{Q_0}$$

Q ₂	Q ₁ Q ₀			
	00	01	11	10
0	X	X	1	X
1	1	1	0	1

$$D_1 = \overline{Q_1} Q_0 + Q_2 Q_1$$

Q ₂	Q ₁ Q ₀			
	00	01	11	10
0	X	X	0	X
1	0	1	1	1

$$D_0 = \overline{Q_0} + Q_2 Q_1$$

Q ₂	Q ₁ Q ₀			
	00	01	11	10
0	X	X	0	X
1	1	0	1	1

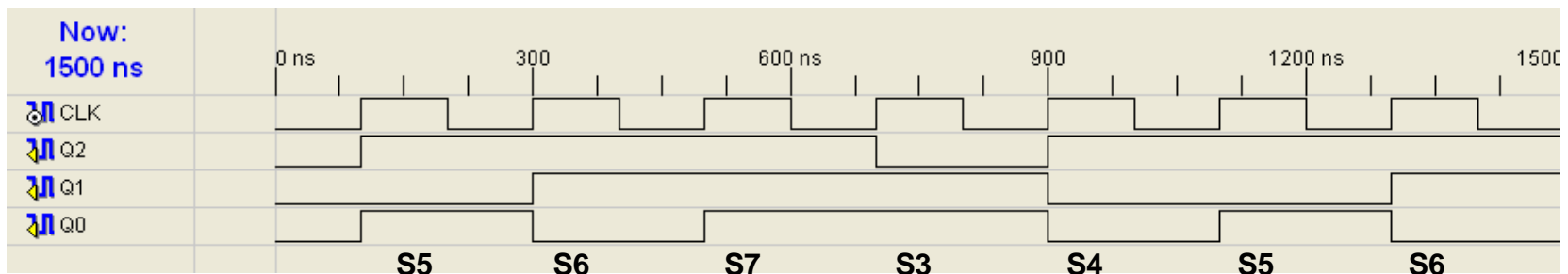
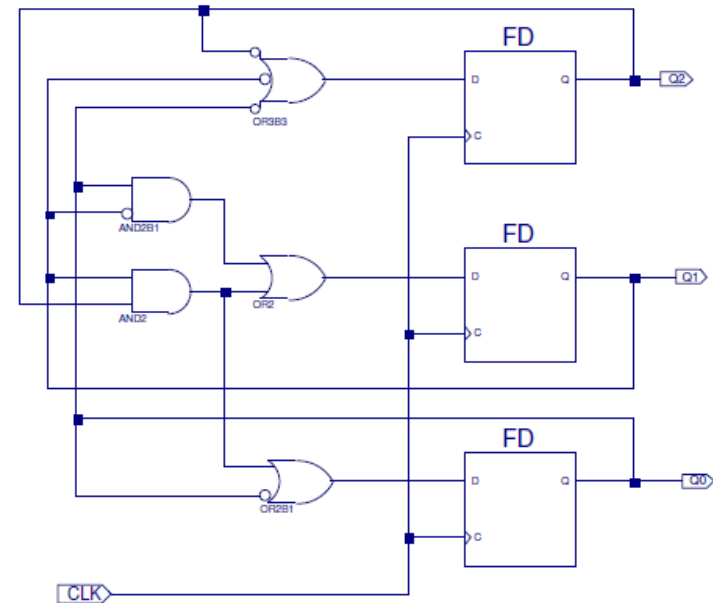
■ Exemplo: implementação com FFs tipo D

Funções de Excitação dos FFs

$$D_2 = \overline{Q_2} + \overline{Q_1} + \overline{Q_0}$$

$$D_1 = \overline{Q_1} Q_0 + Q_2 Q_1$$

$$D_0 = \overline{Q_0} + Q_2 Q_1$$



■ Exemplo: implementação com FFs tipo JK

Tabela de Excitação do FF JK

$Q_n \rightarrow Q_{n+1}$	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

Tabela de Transição de Estados

Estado Actual			Estado Seguinte			Entradas dos FFs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	X	1	X	0	X	0

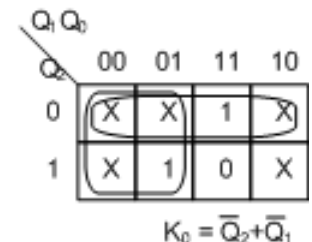
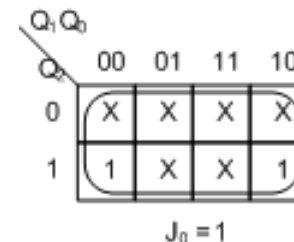
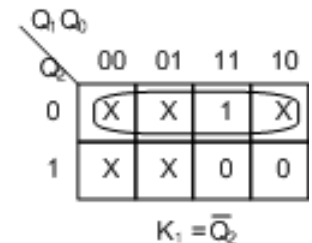
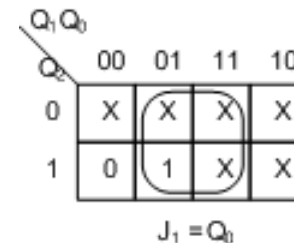
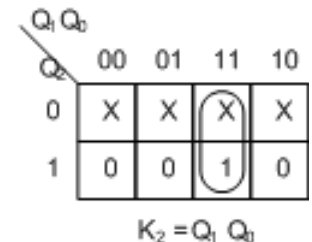
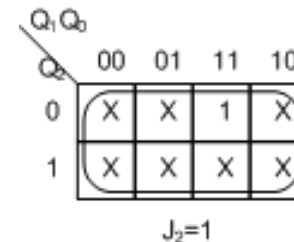
A lógica combinatória do circuito é projectada de modo a forçar nas entradas JK dos FFs os valores que impõem as transições de estado especificadas na tabela de estados.

■ Exemplo: implementação com FFs tipo JK

Tabela de Transição de Estados

Estado Actual			Estado Seguinte			Entradas dos FFs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	X	1	X	0	X	0

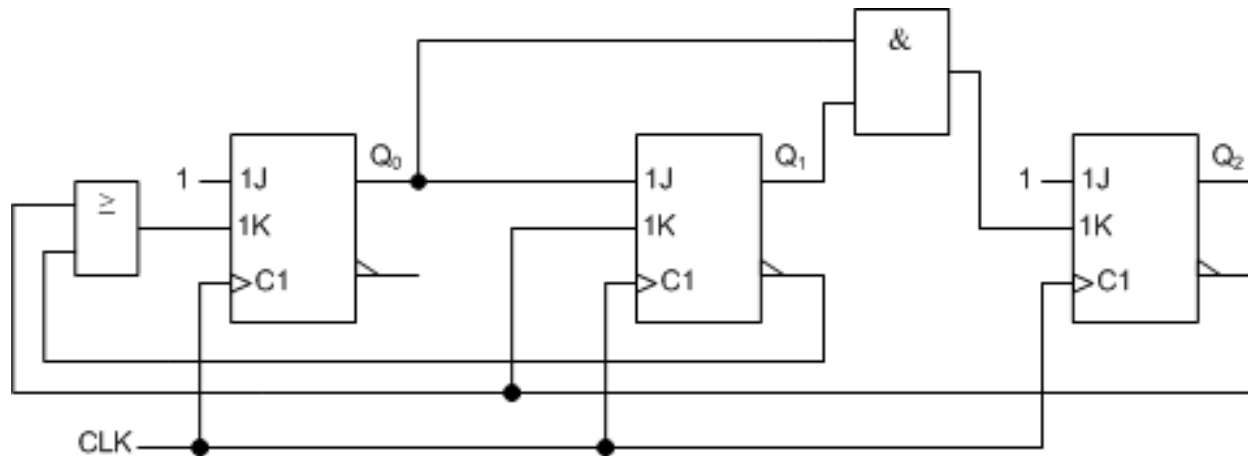
A lógica combinatória força nas entradas JK dos FF os valores que concretizam as transições especificadas, em função do **estado actual**.



■ Exemplo: implementação com FFs tipo JK

Funções de Excitação dos FFs:

$$\begin{array}{lll} J_0 = 1 & J_1 = \underline{Q_0} & J_2 = 1 \\ K_0 = \underline{Q_2} + \underline{Q_1} & K_1 = \underline{Q_2} & K_2 = Q_1 Q_0 \end{array}$$



■ Tema da Próxima Aula:

- ▶ Especificação e projecto de circuitos sequenciais síncronos:
 - Minimização do número de estados
- ▶ Exemplo (Mealy)

Agradecimentos

Algumas páginas desta apresentação resultam da compilação de várias contribuições produzidas por:

- Guilherme Arroz
- Horácio Neto
- Nuno Horta
- Pedro Tomás