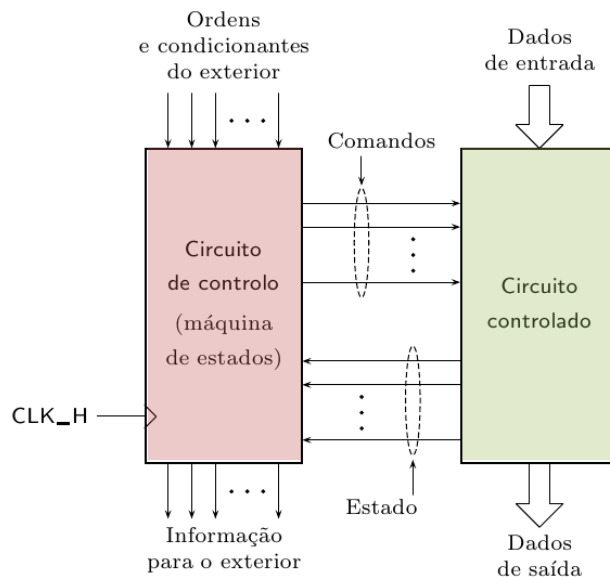


Sistemas Digitais (SD)

Máquinas de Estado Microprogramadas: Circuito de Dados e Circuito de Controlo





■ Na aula anterior:

▶ Memórias:

- Circuitos e tecnologias de memória:
 - RAM:
 - Estática
 - Dinâmica
 - ROM
- Planos de memória
- Mapa de memória
- Hierarquia de memória



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:

- ▶ Projecto de máquinas de estados microprogramadas:
 - Circuito de dados
 - Circuito de controlo
- ▶ Implementação com ROMs
- ▶ Exemplos

□ Bibliografia:

- **M. Mano, C. Kime:** Secção 7.13
- **G. Arroz, J. Monteiro, A. Oliveira:** Secção 7.5
- **G. Arroz, C. Sêro, "Sistemas Digitais: Apontamentos das Aulas Teóricas", IST, 2005:** Secções 19.1 a 19.3 (disponível no [Fenix](#))

■ Circuitos Sequenciais (Revisão):

- ▶ A maioria dos circuitos de controlo apresenta nas saídas um valor que depende das entradas em instantes de tempo anteriores



- ▶ Este comportamento é conseguido usando elementos de **memória**

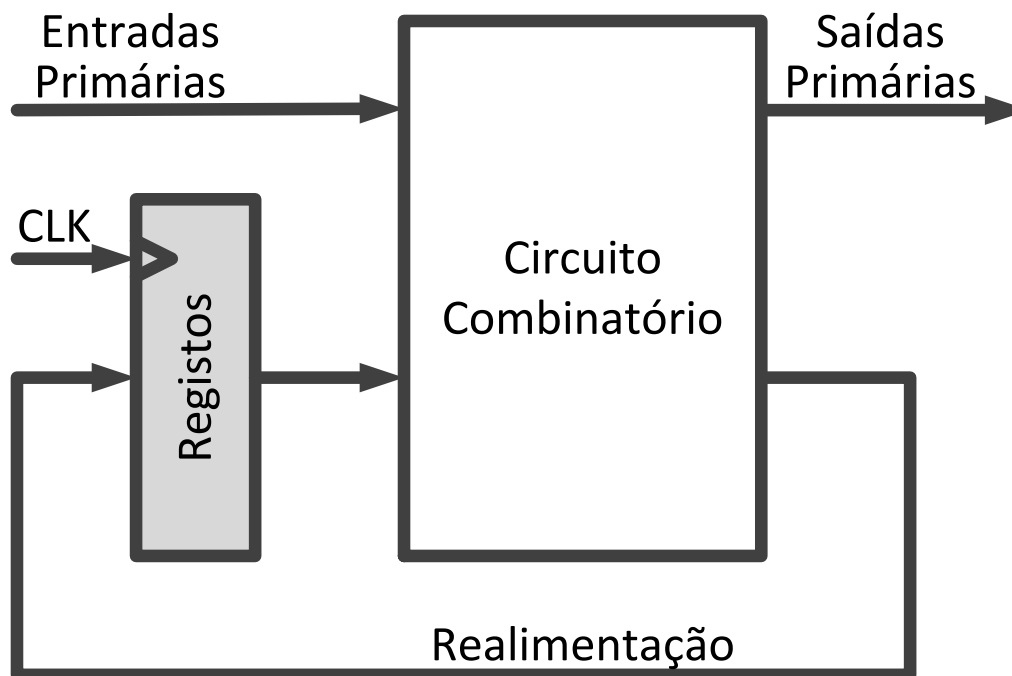


Circuitos Sequenciais

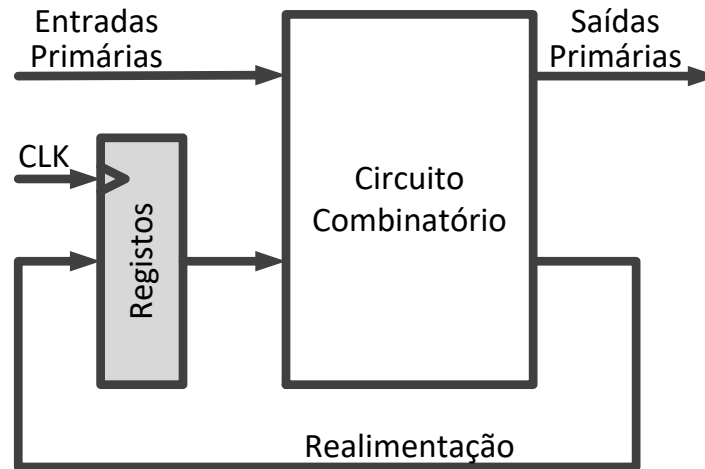
- ▶ Sistemas complexos, como computadores, tablets e telemóveis são também circuitos sequenciais

Porquê?

- Circuito Sequencial Síncrono



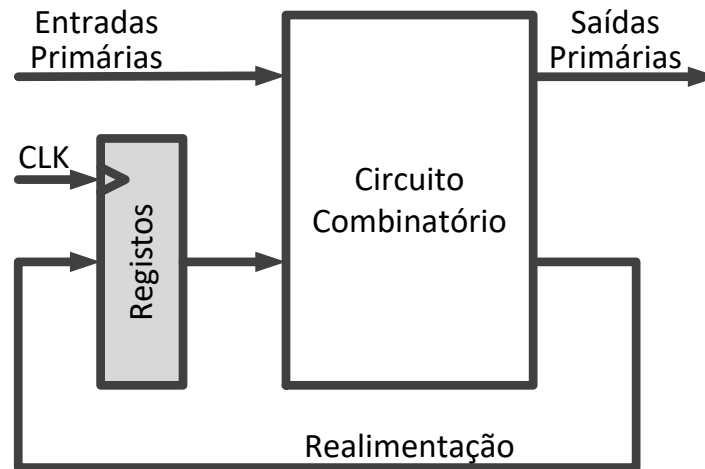
■ Circuito Sequencial Síncrono



- ▶ A mudança de estado ocorre quando existe uma transição do signal de relógio (CLK);
- ▶ O circuito combinatório define o comportamento pretendido (transição entre estados);
- ▶ Os valores das saídas realimentadas no circuito apenas são assumidos como estado na altura em que o relógio muda de nível.

- Um circuito sequencial síncrono corresponde a uma **máquina de estados** definida pelos seguintes componentes:
 - ▶ As possíveis combinações de entrada que controlam a máquina de estados;
 - ▶ As possíveis combinações de saída que são geradas pela máquina de estados;
 - ▶ O conjunto de entradas da máquina;
 - ▶ A **função de transição de estados** - determina de que modo a máquina evolui entre estados, de acordo com o estado presente e a combinação presente nas entradas;
 - ▶ A **função de saída** - determina qual a saída gerada pela máquina para um dado estado e uma dada combinação de entradas;
 - ▶ O estado no qual a máquina de estados deve iniciar o seu funcionamento.

■ Circuito Sequencial Síncrono

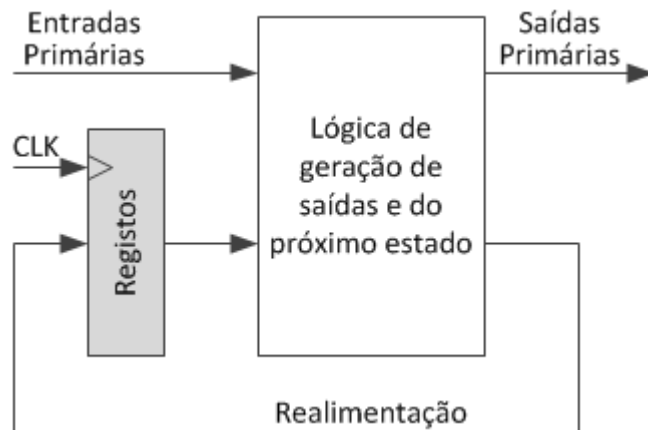


► Na prática...

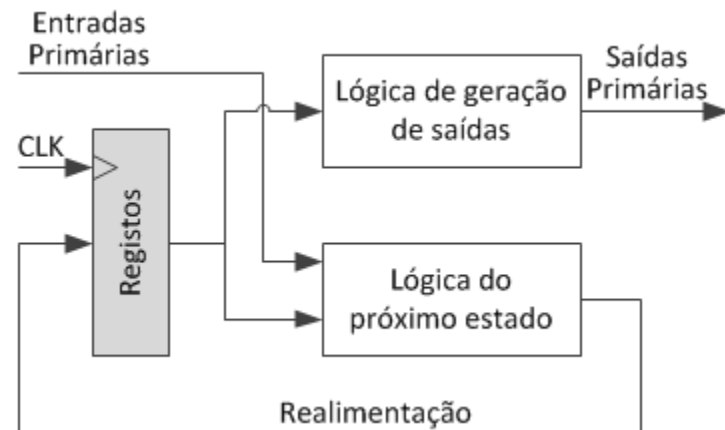
- **Valor dos registos** - define o estado da máquina;
- **Circuito combinatório** - define a função de transição entre estados e a função de saída.

■ Máquinas de Mealy vs Máquinas de Moore

- ▶ Distinguem-se apenas no modo como as saídas são geradas:

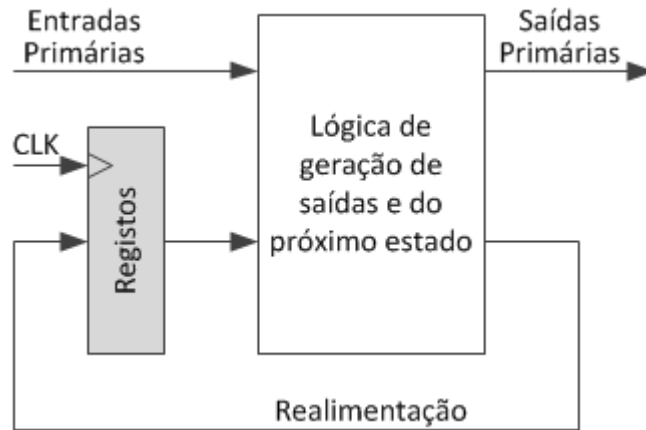


Máquina de Mealy

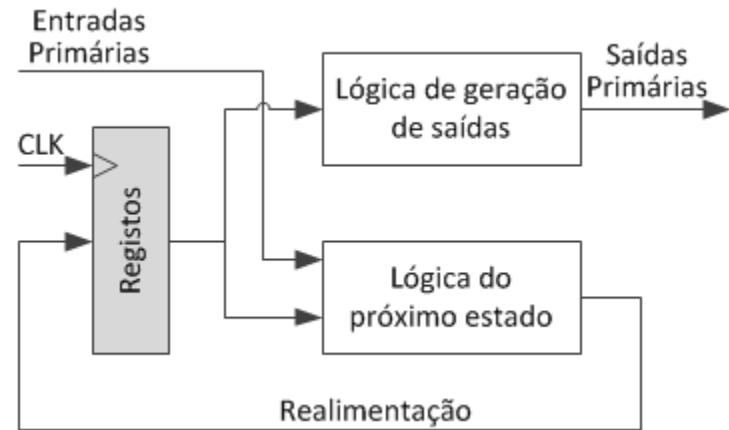


Máquina de Moore

■ Máquinas de Mealy vs Máquinas de Moore



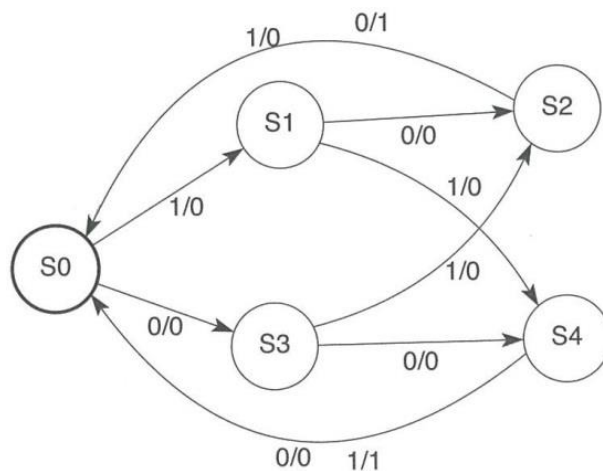
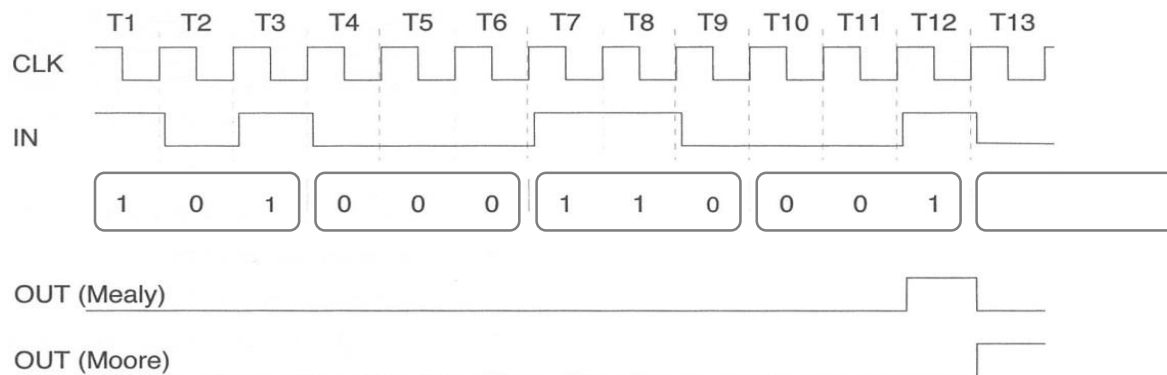
Máquina de Mealy



Máquina de Moore

- ▶ **Máquina de Mealy** - o valor das saídas é função dos valores das variáveis de estado (registos) e dos valores das entradas;
- ▶ **Máquina de Moore** - o valor das saídas é *unicamente* função dos valores das variáveis de estado (registos)

■ Exemplo – Detector de Paridade



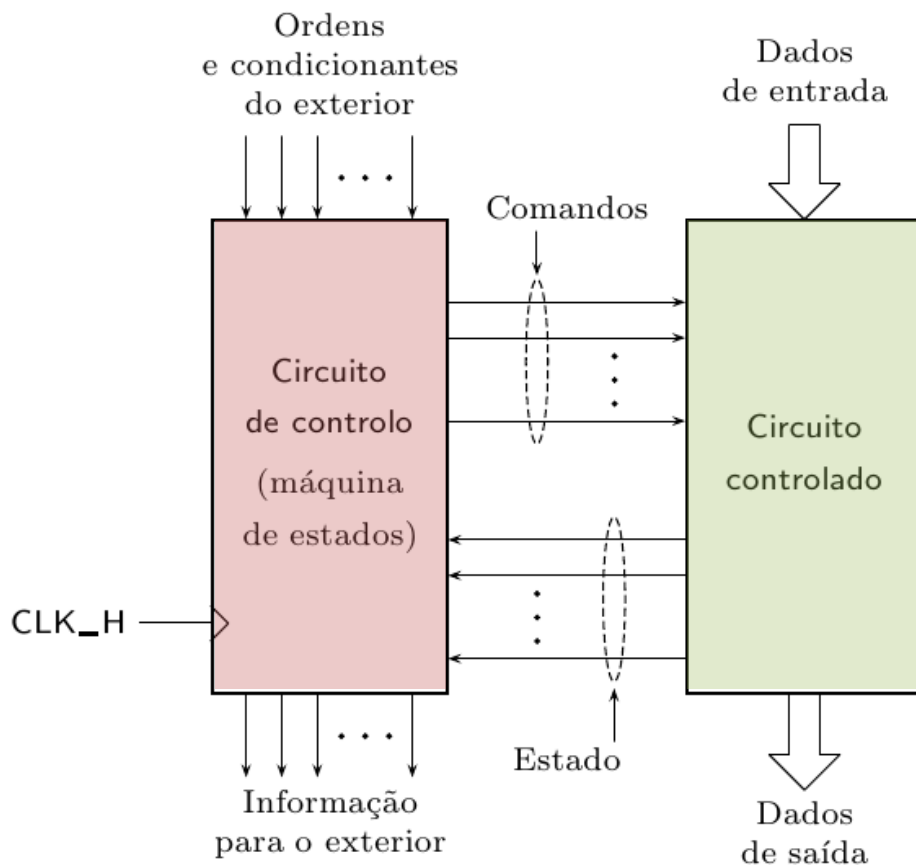
■ Circuito de Dados e Circuito de Controlo

- ▶ Os sistemas digitais com alguma complexidade tornam-se difíceis de ser projectados como vulgares máquinas sequenciais síncronas, porque:
 - Diagramas de estados / tabela de estados de grande dimensão
 - Elevado número de:
 - Entradas
 - Saídas
 - Estados.

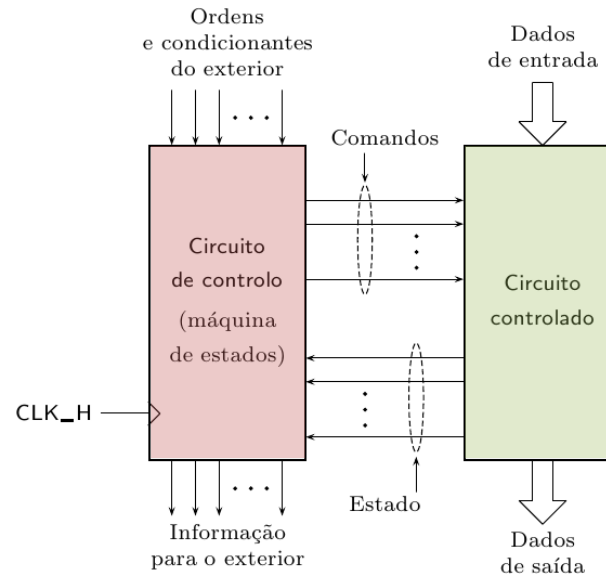
Solução: organizar esses sistemas hierarquicamente, estabelecendo uma divisão clara entre:

- **circuito de dados** - dá suporte ao fluxo e à manipulação de dados;
- **circuito de controlo** - controla o circuito de dados.

■ Circuito de Dados e Circuito de Controlo



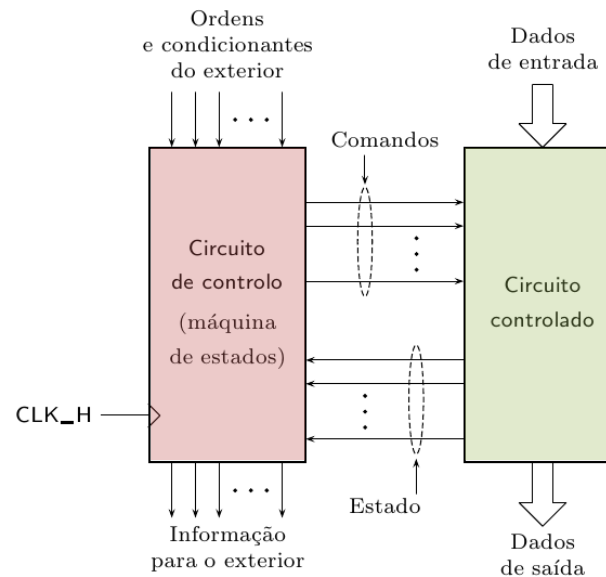
■ Circuito de Dados e Circuito de Controlo



► Em geral:

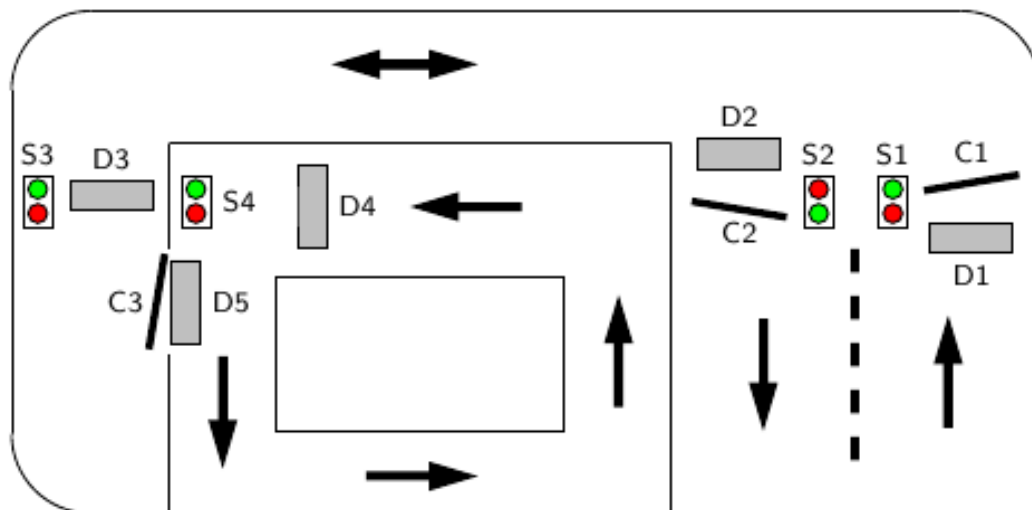
- O **circuito de dados** (controlado) é formado por um conjunto de módulos simples, tais como contadores, registos, multiplexeres, somadores, comparadores, memórias, algumas portas lógicas, etc, podendo ser combinatório ou sequencial.
- O **circuito de controlo** é sempre um circuito sequencial síncrono.

■ Circuito de Dados e Circuito de Controlo



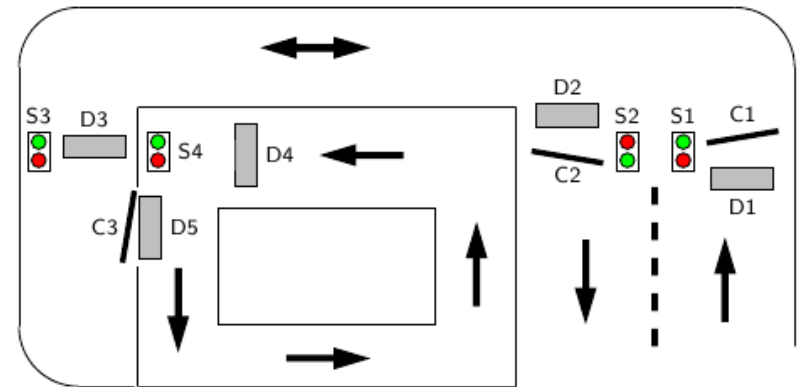
- ▶ O circuito de controlo de um sistema complexo designa-se abstractamente por **máquina de estados** ou **máquina algorítmica**, mais simplesmente pela sigla **ASM**, que significa “*Algorithmic State Machine*”.

■ Exemplo: acesso a um parque de estacionamento



- ▶ O acesso faz-se por uma via de sentido único, controlada na entrada e na saída pelas cancelas C1 a C3, pelos semáforos S1 a S4, e pelos sensores D1 a D5.
- ▶ O controlador contém um contador ascendente/descendente, que guarda a informação sobre o número de carros estacionados no parque.

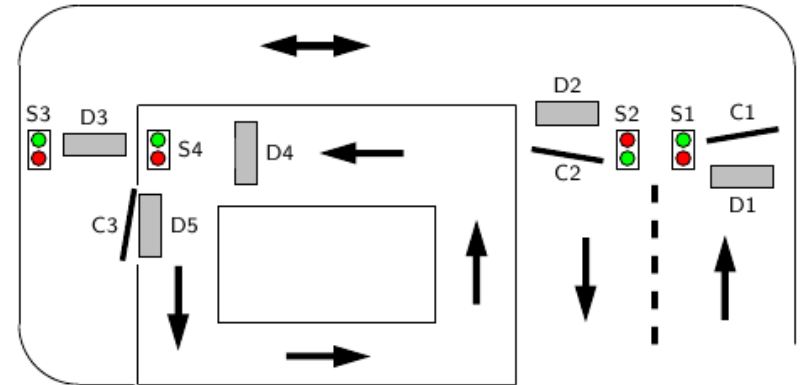
■ Exemplo: acesso a um parque de estacionamento



► Funcionamento:

- Quando o parque está cheio, só podem sair carros;
- Quando não está cheio, podem entrar ou sair;
- Porque a rua de acesso é estreita, só pode passar um carro de cada vez.

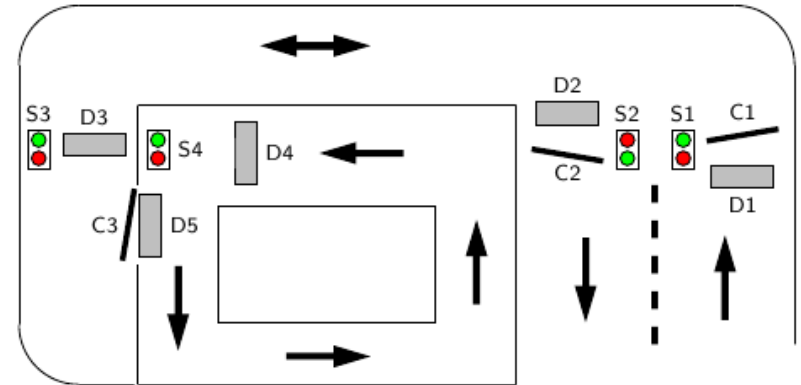
■ Exemplo: acesso a um parque de estacionamento



► Funcionamento:

- A **saída** é detectada pela presença de um carro que pisa D4. Se não há entrada em curso, o semáforo S4 fica verde e a cancela C3 abre. Em seguida, espera-se que o carro pise D5 e saia, para se fechar a cancela e colocar o semáforo S4 em vermelho. Entretanto, coloca-se o semáforo S2 a verde. Quando o carro pisa D2, abre-se C2, que se mantém aberta enquanto a viatura estiver a pisar D2. Quando o carro deixar de pisar D2, o semáforo S2 passa a vermelho e C2 fecha. Nessa altura desconta-se uma unidade no **contador** de lugares ocupados no parque.

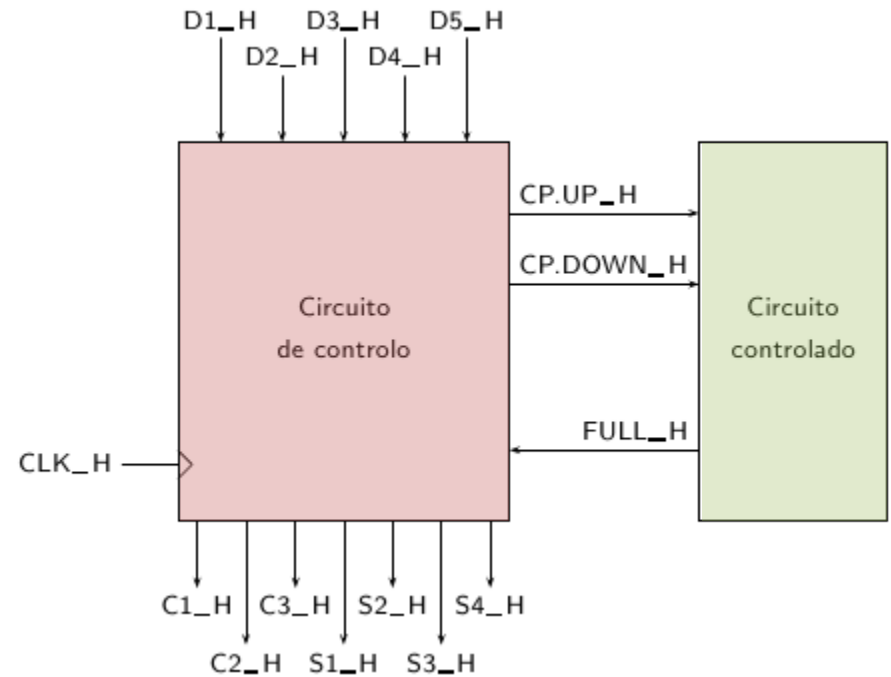
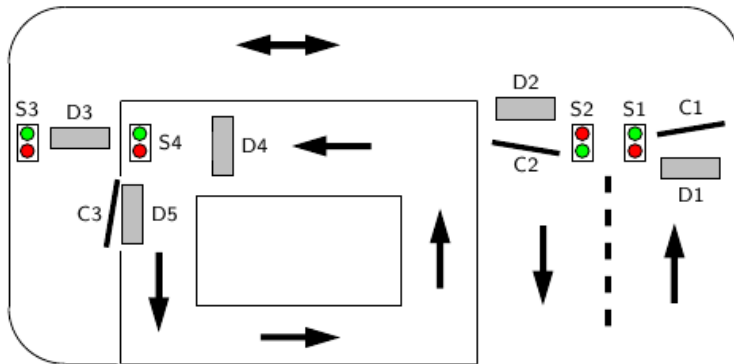
■ Exemplo: acesso a um parque de estacionamento



► Funcionamento:

- A **entrada** começa com um carro a pisar D1. Se não há saída em curso, o semáforo S1 fica verde e a cancela C1 abre, ficando aberta enquanto o carro é detectado por D1. Quando o carro deixa D1, S1 fica a verde, e quando chega a D3 a cancela C3 é aberta e o carro entra, passando S3 a vermelho e ficando o circuito à espera que D5 seja pisado. Só depois de D5 deixar de ser pisado é que C3 fecha. Nessa altura, o contador é incrementado.

■ Modelo do circuito de controlo

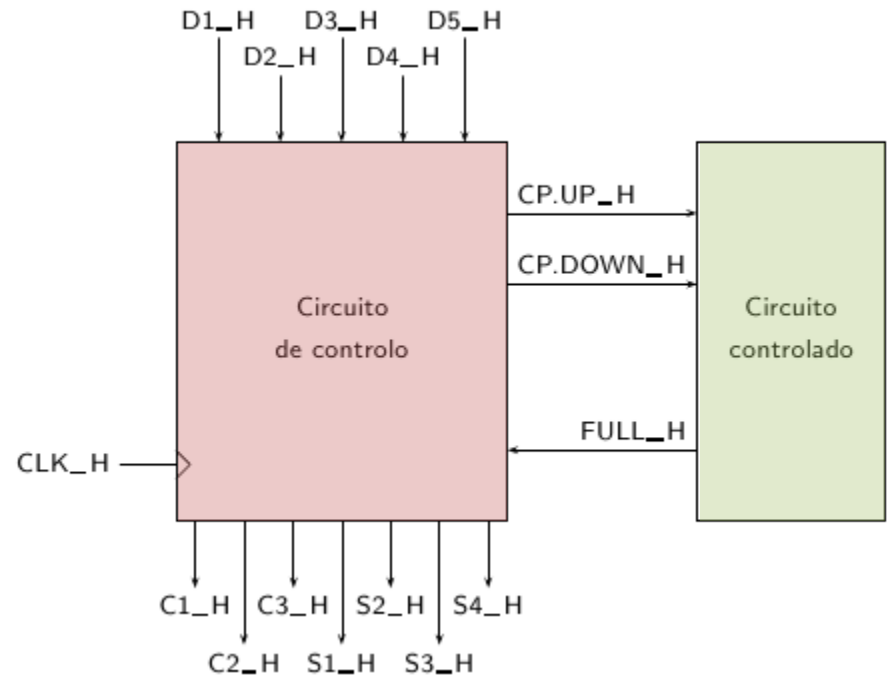


■ Modelo do circuito de controlo

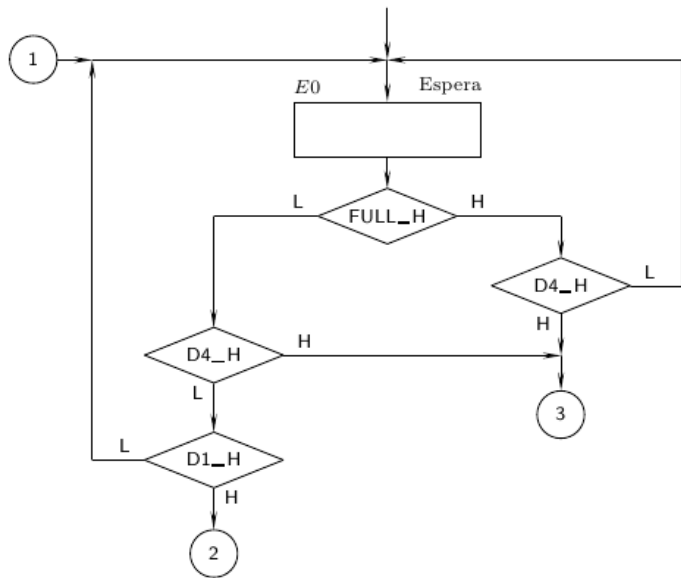
- ▶ 6 entradas
- ▶ 9 saídas
- ▶ Quantos estados???



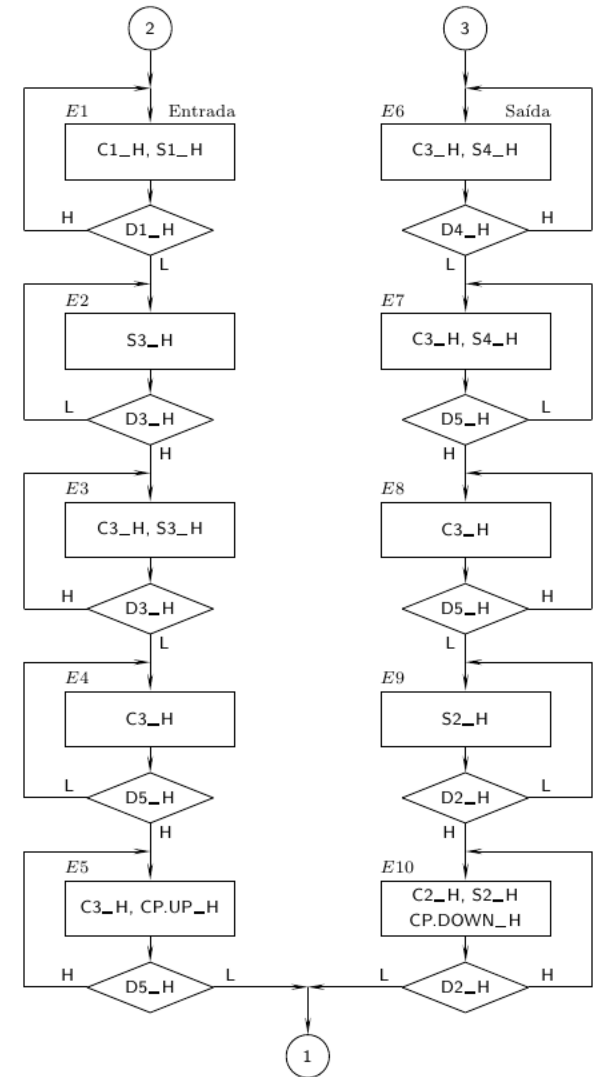
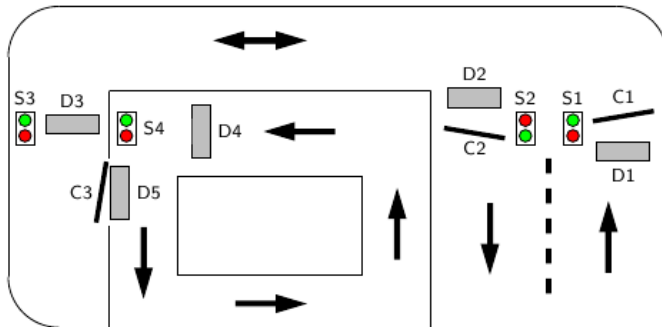
Fluxograma



Fluxograma do circuito de controlo



11 Estados!!!



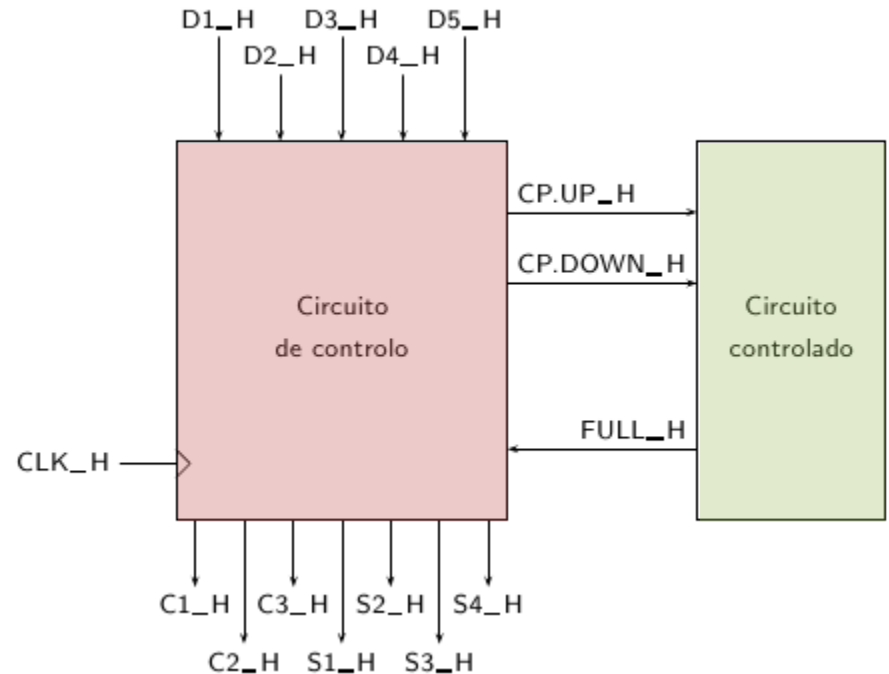
■ Modelo do circuito de controlo

- ▶ 6 entradas
- ▶ 9 saídas
- ▶ 11 estados

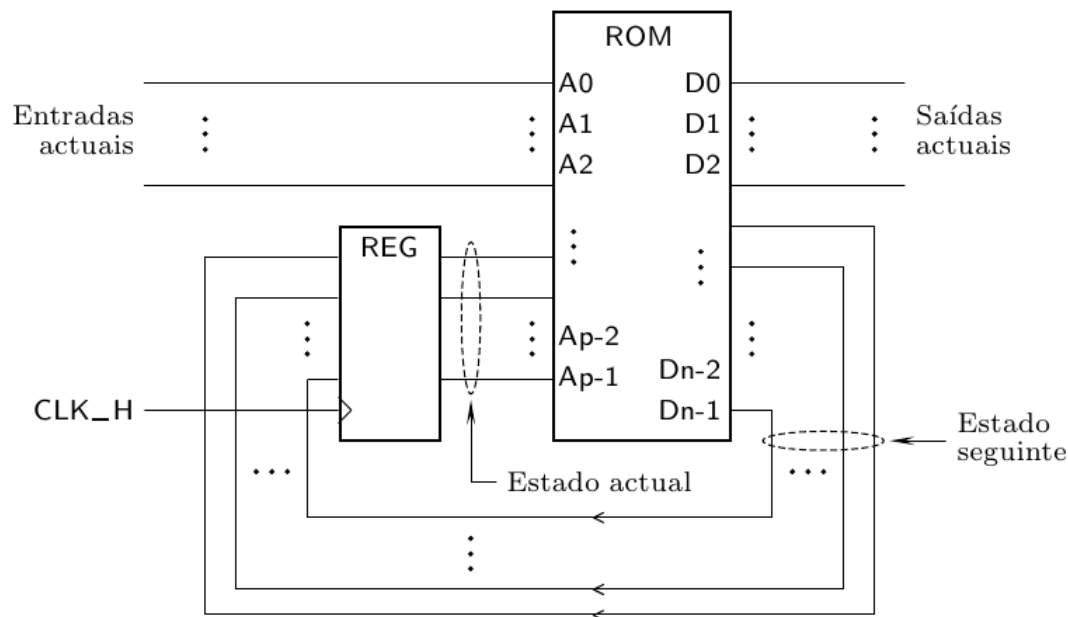


- ▶ Grande quantidade de lógica para excitar os flip-flops e as saídas do circuito
- ▶ Fluxograma/diagrama de estados complexo

SOLUÇÃO: implementação com **ROMs**



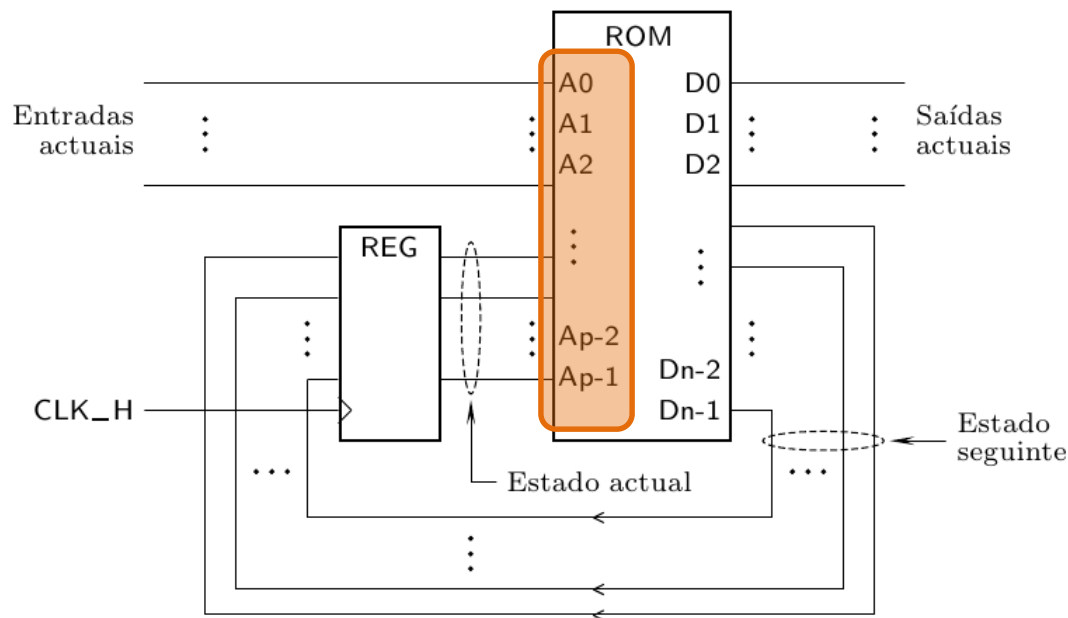
■ Estrutura básica de controlo por ROM



► A memória **ROM** substitui a lógica combinatória para gerar:

- Estado seguinte
- Saída do circuito

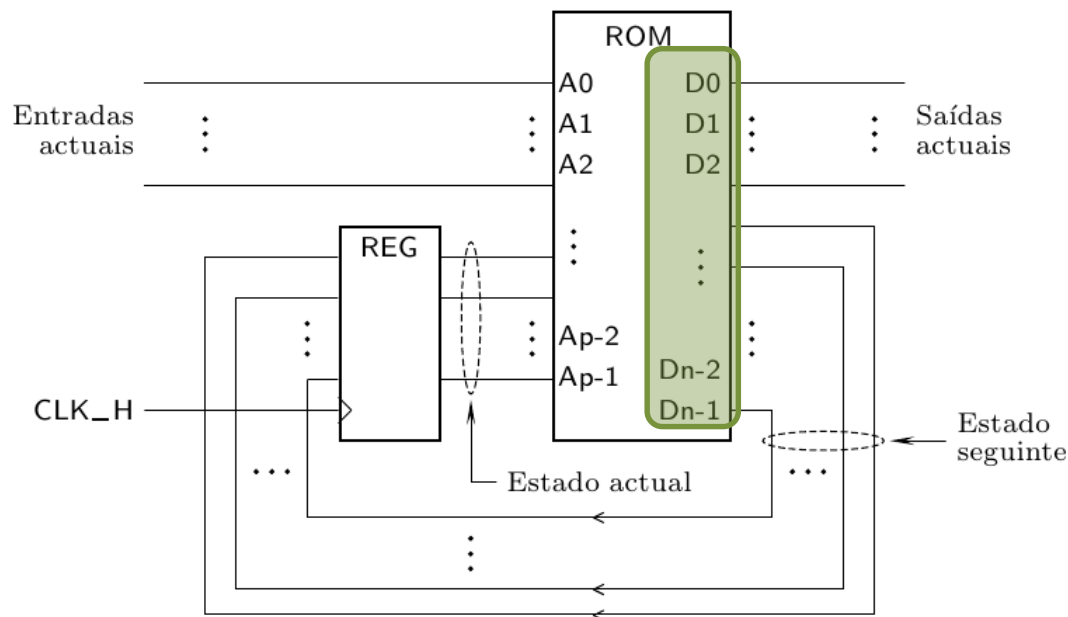
■ Estrutura básica de controlo por ROM



► Entradas da ROM (**barramento de endereços**):

- Entradas externas da máquina de estados
- Estado actual

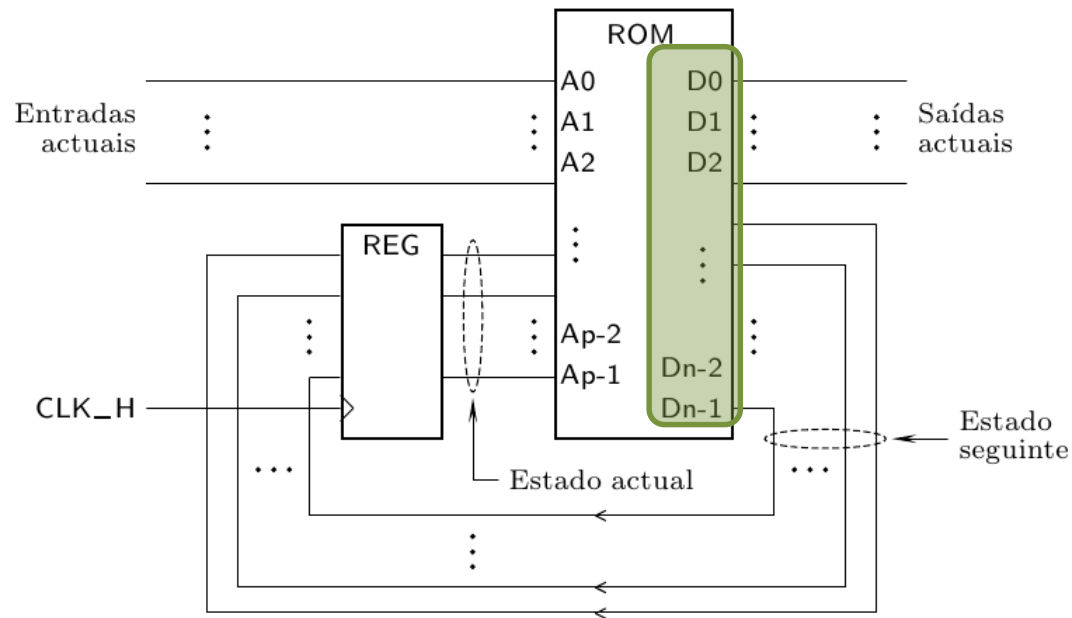
■ Estrutura básica de controlo por ROM



► Saídas da ROM (**barramento de dados**):

- Saídas para o exterior da máquina de estados
- Saídas (comandos) internas + **estado seguinte**

■ Estrutura básica de controlo por ROM



- O conteúdo de cada endereço na ROM é constituído por dois campos:

Saídas Actuais: $Out(t)$	Estado Seguinte: $Q(t+1)$
--------------------------	---------------------------

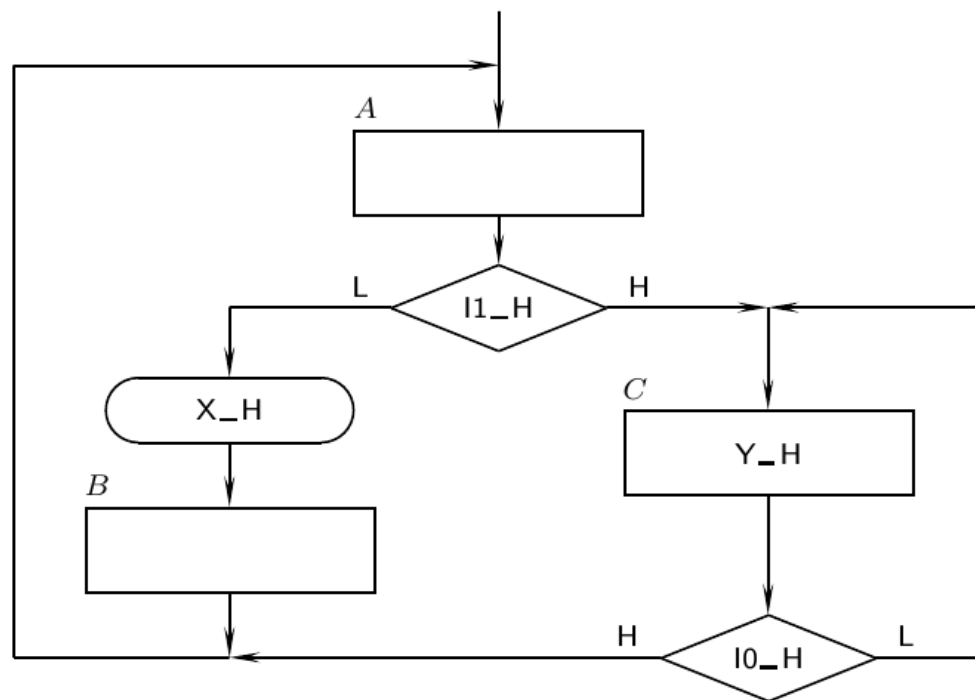
Exemplo (simples)

▶ 3 estados:

- A ($Q_1Q_0=00$)
- B ($Q_1Q_0=01$)
- C ($Q_1Q_0=10$)

▶ 2 entradas: I0 e I1

▶ 2 saídas: X e Y

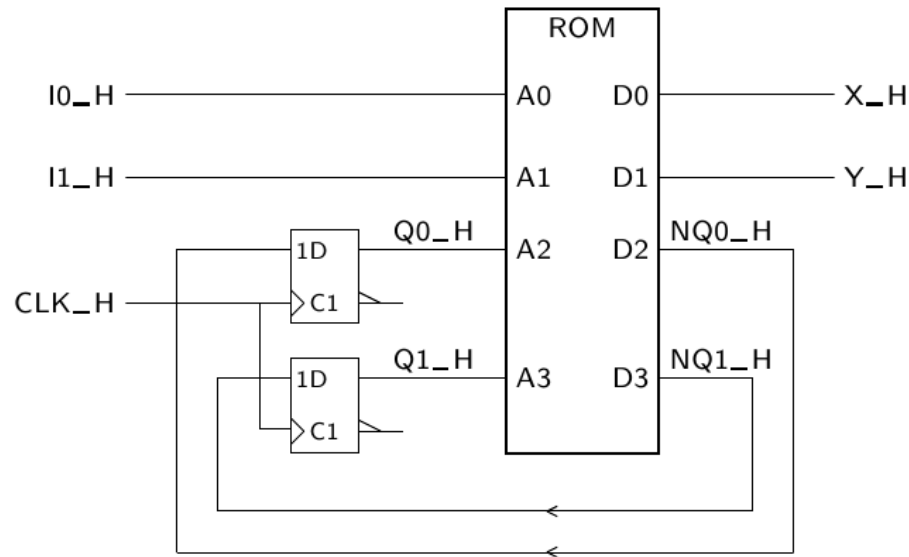


Moore ou Mealy?

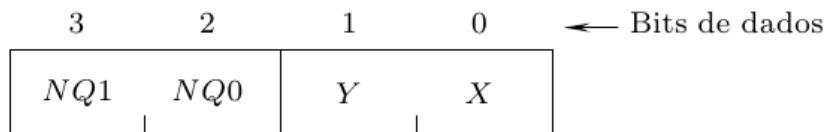
- **Mealy**, pois no estado A a variável X está activa apenas se I1 está inactiva (saída condicional)

■ Exemplo (simples)

- ▶ Diagrama de blocos de um controlador implementado com ROM:



- ▶ Formato de cada palavra da ROM:



■ Exemplo (simples)

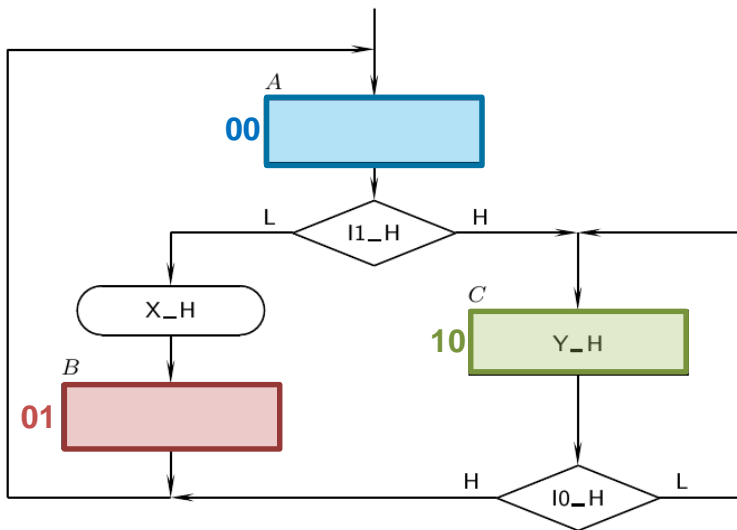


Tabela de transição de estados

Estado	Endereço	Q1	Q0	I1	I0	NQ1	NQ0	Y	X
		A3	A2	A1	A0	D3	D2	D1	D0
A	0	0	0	0	0	0	1	0	1
A	1	0	0	0	1	0	1	0	1
A	2	0	0	1	0	1	0	0	0
A	3	0	0	1	1	1	0	0	0
B	4	0	1	0	0	0	0	0	0
B	5	0	1	0	1	0	0	0	0
B	6	0	1	1	0	0	0	0	0
B	7	0	1	1	1	0	0	0	0
C	8	1	0	0	0	1	0	1	0
C	9	1	0	0	1	0	0	1	0
C	10	1	0	1	0	1	0	1	0
C	11	1	0	1	1	0	0	1	0
	12	1	1	0	0	×	×	×	×
	13	1	1	0	1	×	×	×	×
	14	1	1	1	0	×	×	×	×
	15	1	1	1	1	×	×	×	×

■ Exemplo (simples)

Tabela de transição de estados

Estado	Endereço	Q1	Q0	I1	I0	NQ1	NQ0	Y	X
		A3	A2	A1	A0	D3	D2	D1	D0
A	0	0	0	0	0	0	1	0	1
A	1	0	0	0	1	0	1	0	1
A	2	0	0	1	0	1	0	0	0
A	3	0	0	1	1	1	0	0	0
B	4	0	1	0	0	0	0	0	0
B	5	0	1	0	1	0	0	0	0
B	6	0	1	1	0	0	0	0	0
B	7	0	1	1	1	0	0	0	0
C	8	1	0	0	0	1	0	1	0
C	9	1	0	0	1	0	0	1	0
C	10	1	0	1	0	1	0	1	0
C	11	1	0	1	1	0	0	1	0
	12	1	1	0	0	×	×	×	×
	13	1	1	0	1	×	×	×	×
	14	1	1	1	0	×	×	×	×
	15	1	1	1	1	×	×	×	×

Conteúdo da ROM

Endereço	Dados
0	0101
1	0101
2	1000
3	1000
4	0000
5	0000
6	0000
7	0000
8	1010
9	0010
10	1010
11	0010
12	0000
13	0000
14	0000
15	0000

Don't care substituídos por um valor concreto: 0 ou 1

■ Exemplo (simples)

Conteúdo da ROM

Endereço	Dados
0	0101
1	0101
2	1000
3	1000
4	0000
5	0000
6	0000
7	0000
8	1010
9	0010
10	1010
11	0010
12	0000
13	0000
14	0000
15	0000

► Dimensão da ROM:

- Mínima: 12 endereços de 4 bits = 48 bits
- Normalizada: 16 endereços de 4 bits = 64 bits



Nº de endereços corresponde a uma potência inteira de 2



■ Tema da Próxima Aula:

- ▶ Projecto de máquinas de estados microprogramadas:
 - com endereçamento explícito
 - com endereçamento implícito
- ▶ Exemplos

Agradecimentos

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

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