

Lógica e Sistemas Digitais - 9

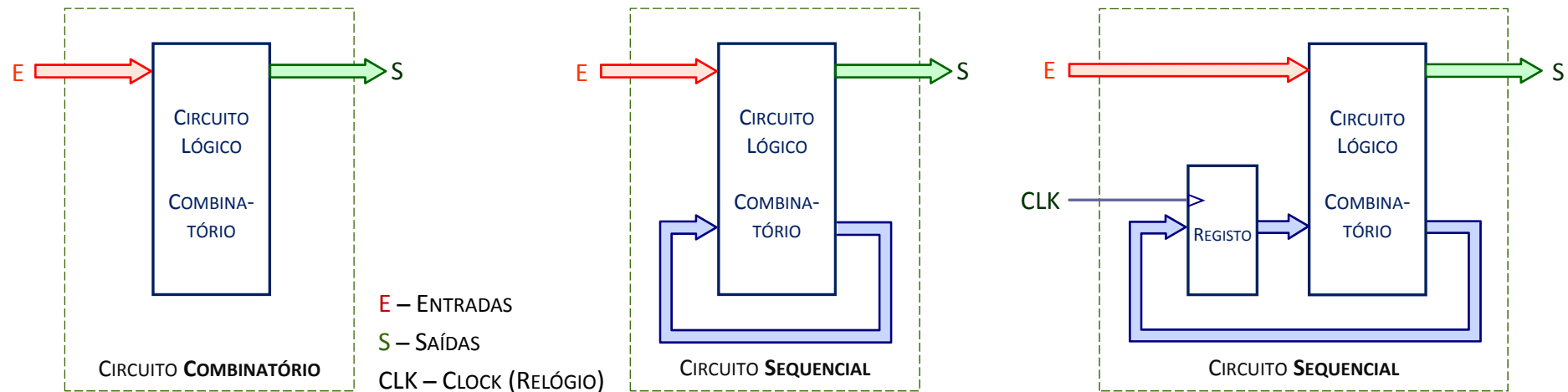
Máquinas de Estado Algorítmicas

ISEL

Departamento de Engenharia de Electrónica
e Telecomunicações e de Computadores
Lisboa

Mário Araújo

2016-1

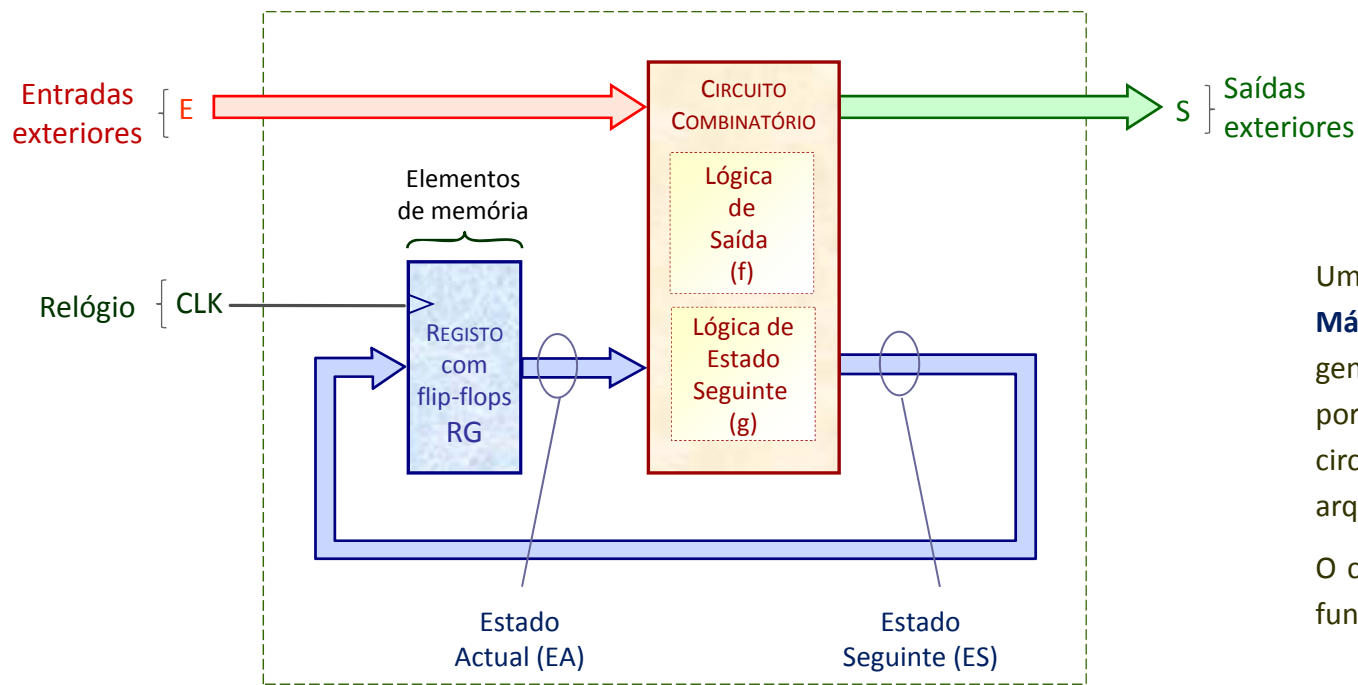


Circuito **COMBINATÓRIO**.

Circuito SEQUENCIAL **ASSÍNCRONO**.

Circuito SEQUENCIAL **SÍNCRONO**.

- Os circuitos lógicos são de dois tipos: **COMBINATÓRIOS** e **SEQUENCIAIS**, e os circuitos sequenciais podem ainda ser **SÍNCRONOS** ou **ASSÍNCRONOS**.
- Ambos possuem entradas por onde os dados são introduzidos, e saídas onde são reflectidos os resultados do processamento realizado.
- Nos circuitos **COMBINATÓRIOS** – Fig. da esquerda – o valor lógico presente nas saídas é unicamente determinado pelos valores lógicos presentes nas entradas em cada momento. Estes circuitos foram estudados em módulos anteriores.
- Nos circuitos **SEQUENCIAIS** o valor lógico presente nas saídas em determinado instante não é unicamente determinado pelos valores lógicos presentes nas entradas nesse instante, depende também da sequência desses valores ao longo do tempo.
- Este comportamento implica a existência de **MEMÓRIA** – o circuito guarda informação de acontecimentos passados.
- Um circuito sequencial **ASSÍNCRONO** – Fig. do meio – não tem entrada de relógio, e faz uso de LATCHES.
- Um circuito sequencial **SÍNCRONO** – Fig. da direita – tem todas as entradas de relógio dos FLIP-FLOPS ligadas a um relógio comum.
- Nem todos os circuitos com flip-flops são síncronos – caso dos contadores assíncronos estudados no módulo anterior.



Modelo geral de um **Circuito Sequencial Síncrono**.

Lógica de Saída (f)

O bloco designado Lógica de Saída (**f**) representa um circuito combinatório que determina as Saídas exteriores.

Lógica de Estado Seguinte (g)

O bloco designado Lógica de Estado Seguinte (**g**) representa um circuito combinatório que determina o Estado Seguinte.

Um **circuito sequencial síncrono** (ou **Máquina de Estados** síncrona como é genericamente designado) é constituído por elementos de memória e por um circuito combinatório de acordo com a arquitectura do desenho ao lado.

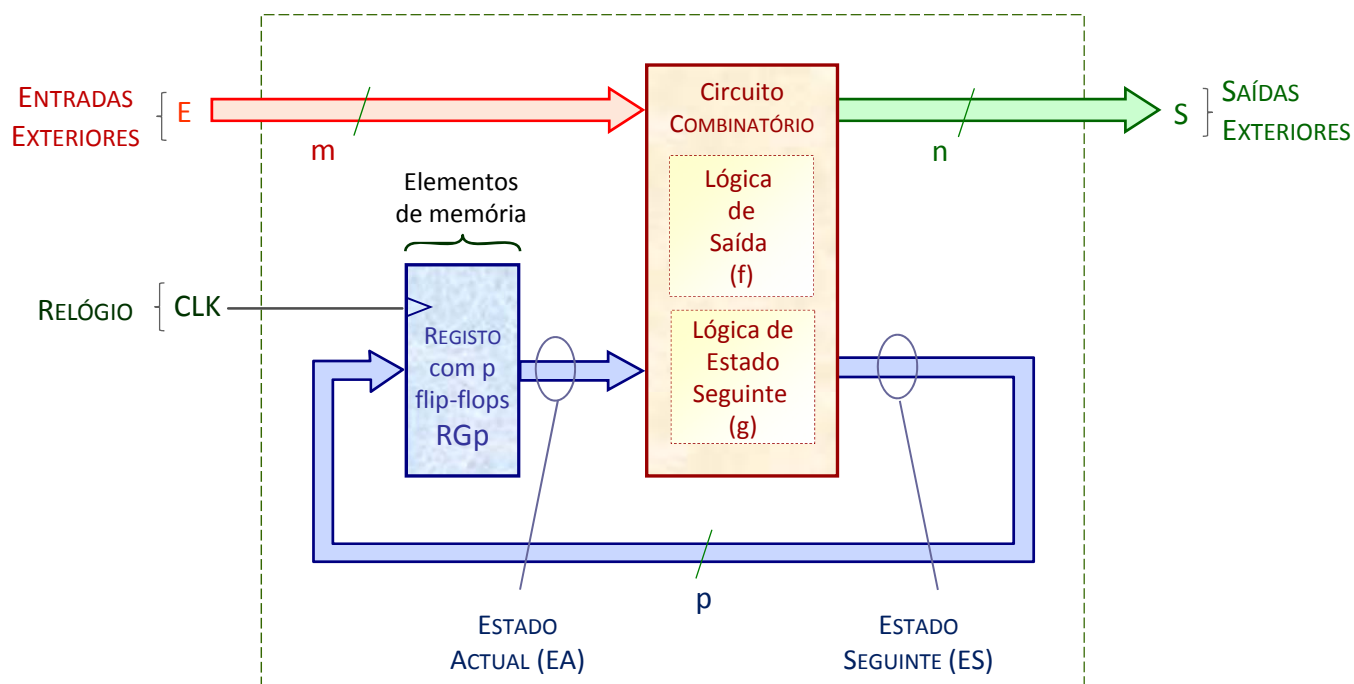
O circuito combinatório desempenha duas funções designadas por f e g:

f – função geradora das **Saídas** do sistema.

g – função geradora do **Estado Seguinte** do sistema.

$$\text{SAÍDAS} = f (\text{Estado actual}, \text{Entradas})$$

$$\text{ESTADO SEGUINTE} = g (\text{Estado actual}, \text{Entradas})$$



Modelo geral de um **Circuito Sequencial Síncrono**.

$$S_1 = f_1(E_1..E_m, Q_1..Q_p)$$

... ..

$$S_n = f_n(E_1..E_m, Q_1..Q_p)$$

$$Q^+_1 = g_1(E_1..E_m, Q_1..Q_p)$$

... ..

$$Q^+_p = g_p(E_1..E_m, Q_1..Q_p)$$

As p saídas $Q_{1..p}$ de cada um dos flip-flops constituintes do registo denominam-se **variáveis de estado** e correspondem ao estado actual.

As funções **f** e **g** podem ser descritas por uma tabela de estados e de saídas ou por equações lógicas retiradas dos diagramas de estados.

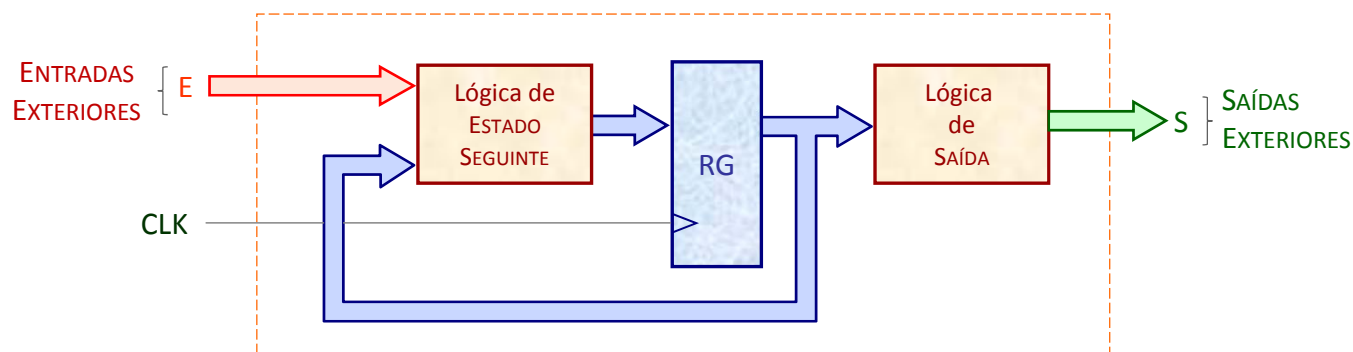
As n Saídas $S_{1..n}$ do sistema e as p saídas $Q^+_{1..p}$ do Estado Seguinte dependem do valor lógico do Estado Actual e das m Entradas como indicado.

Redesenhando e tomando em conta uma divisão possível entre os blocos geradores obtém-se o circuito indicado na figura: trata-se do modelo apresentado por Moore e, por isso, conhecido pela designação de **modelo de Moore**.

Edward F. Moore



1925-2003
(USA)



Modelo de **Moore** de um Circuito Sequencial Síncrono.

O nome advém de **Edward F. Moore** (1925-2003), um pioneiro e investigador da Teoria da Comutação e Lógica que escreveu em 1956 o artigo seminal **Gedanken-experiments on Sequential Machines** (*Gedanken* de pensamento).

Trabalhou entre 1950-1951 no projecto do computador electrónico ILIAC-I (Illinois Automatic Computer) na Universidade de Illinois em Urbana – o ILIAC-I comportava 2800 válvulas, tinha 3 m de largura, 0,6 m de profundidade e 2,6 m de altura, e pesava 4,5 toneladas.

Doutorado pela Brown University em 1950, trabalho nos Bell Labs de 1951 a 1956, foi professor de Matemática e Ciência da Computação no MIT e na Universidade de Harvard de 1961 a 1962, e na Universidade de Wisconsin em Madison de 1966 até 1985.

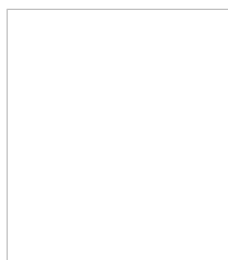
No modelo de **Moore** uma entrada define um caminho para um estado e só no estado a saída sofre alteração - as saídas num dado instante são função directa do estado, e não são sensíveis aos valores lógicos presentes nas entradas externas nesse instante.

O circuito na saída reage apenas ao seu passado e não ao seu presente.

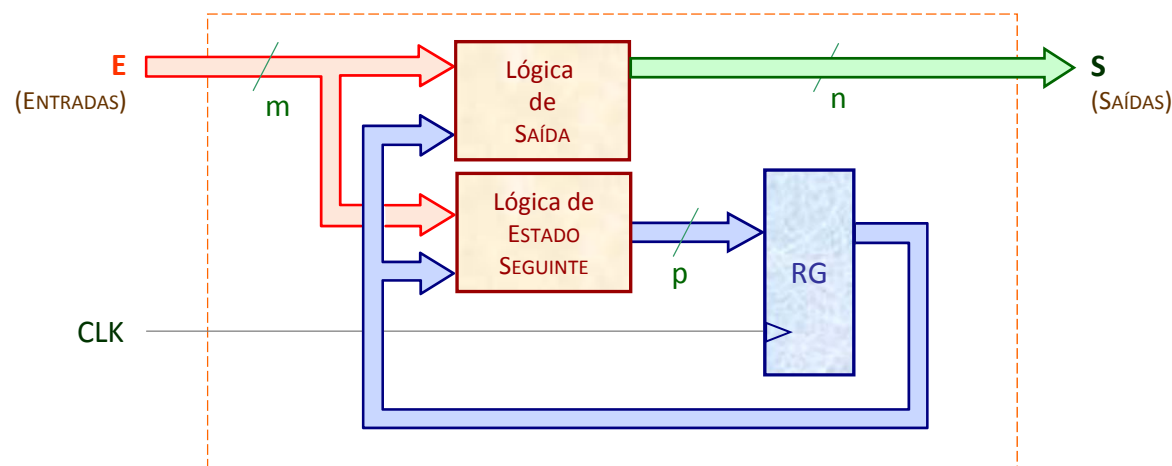
Os circuitos de Moore apresentam maior simplicidade na geração das saídas.

Há outra forma de estruturar circuitos sequenciais síncronos como indicado na figura: trata-se do modelo apresentado por Mealy e, por isso, conhecido por **modelo de Mealy**. A diferença com o esquema anterior (modelo de Moore) consiste na lógica de saída que é função dos estados do circuito e das entradas externas.

George H. Mealy



1927-2010
(USA)



Modelo de **Mealy** de um Circuito Sequencial Síncrono.

A designação de Máquina de Mealey advém do nome de **George H. Mealy**, (1927-2010), outro pioneiro da teoria das máquinas de estado que publicou em 1955 um artigo intitulado **A Method for Synthesizing Sequential Circuits**.

Foi professor em Harvard nos anos 70 e desenhou nos Bell Labs nos anos 50 o primeiro sistema operativo para o computador IBM 704.

Os circuitos de Mealy conduzem a um menor número de estados e consequentemente à redução do número de flip-flops necessários.

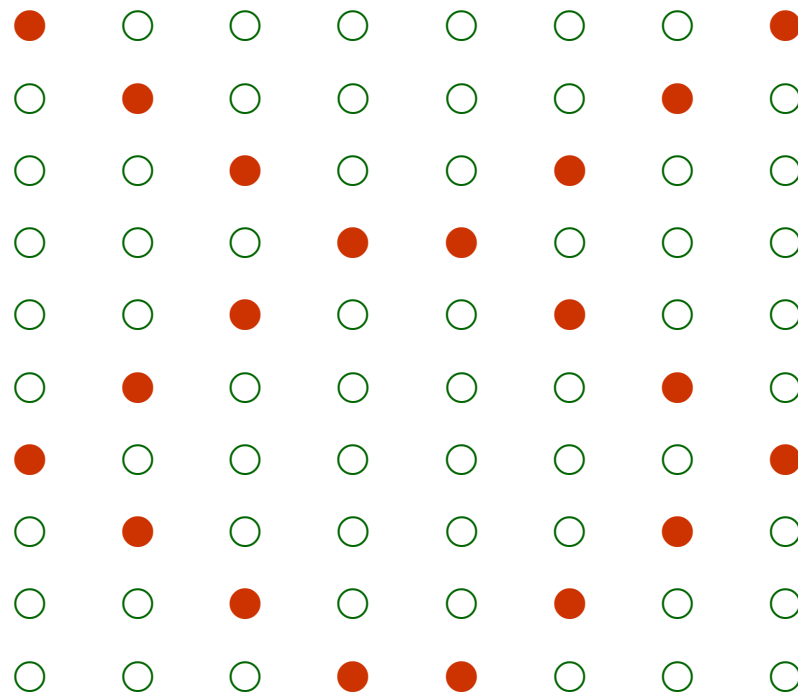
No modelo de **Mealy** a saída é alterada logo que a entrada varia - as saídas são função do estado e das entradas.

A única diferença entre uma Máquina de Moore e uma Máquina de Mealey reside no modo como as saídas são geradas.

Exemplo 9-1

OBJECTIVO

Pretende-se projectar com flip-flops T um circuito sequencial que, ao ritmo dos impulsos de Clock, acenda um conjunto de 8 LEDs segundo os padrões mostrados em cada linha da figura:



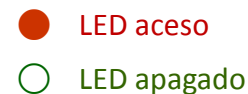
Padrão de LEDs a acender.

- Verifica-se que não é possível realizar o circuito com apenas 4 estados .

Por exemplo, da configuração da 3ª linha o circuito transita para a configuração da 4ª linha nuns casos, e para a configuração da 6ª linha noutros.

É necessário evocar 6 estados.

- Verifica-se ainda pela simetria da figura que o circuito necessitará de apenas 4 saídas distintas (a designar por **a**, **b**, **c**, **d**).



Legenda.

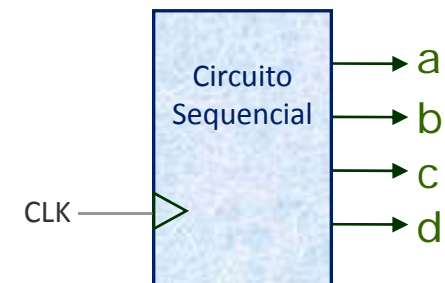


Diagrama de blocos do circuito.

DIAGRAMA DE ESTADOS DO CIRCUITO A SINTETIZAR (EX. 9-1-2)

9-8

Exemplo 9-1

a	b	c	d	d	c	b	a	Estado
●	○	○	○	○	○	○	○	0
○	●	○	○	○	○	○	○	1
○	○	●	○	○	○	○	○	2
○	○	○	●	●	○	○	○	3
○	○	●	○	○	○	○	○	4
○	●	○	○	○	○	○	○	5
●	○	○	○	○	○	○	○	0
○	●	○	○	○	○	○	○	1
○	○	●	○	○	○	○	○	2
○	○	○	●	●	○	○	○	3

Geometria dos padrões a gerar com atribuição de uma saída a cada LED.

O processo de síntese do circuito sequencial inicia-se pela descrição do comportamento do sistema sequencial em linguagem natural. Sendo o sistema dinâmico, o seu comportamento será melhor descrito através da elaboração de um **DIAGRAMA DE ESTADOS**. O Diagrama de Estados pode ser representado por um grafo orientado com **círculos** representativos dos **estados** da máquina (como se viu no módulo anterior).

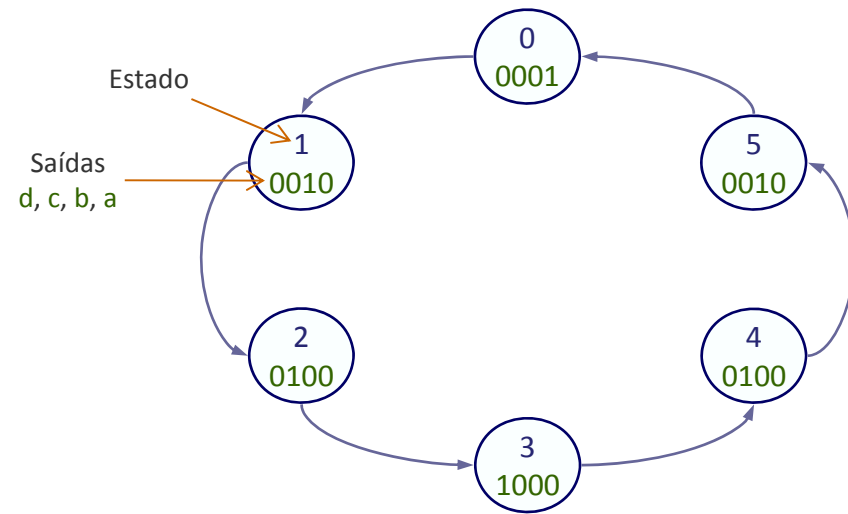
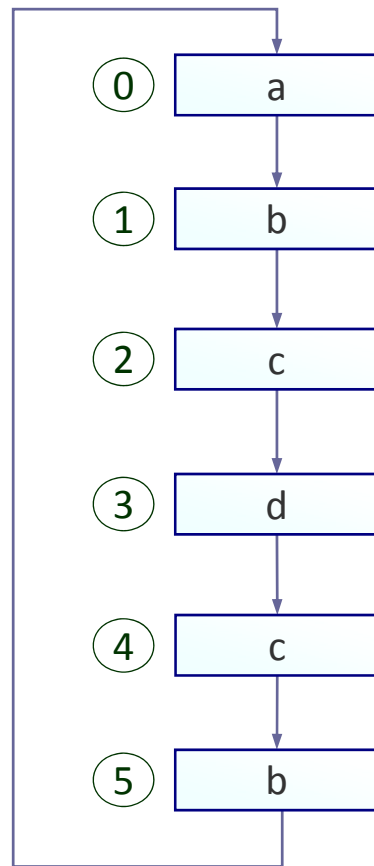


Diagrama de Estados concebido a partir da descrição do funcionamento do circuito sequencial em linguagem natural.

Cada círculo representativo de um estado conterá a referência única desse estado. As transições entre estados são representadas por vectores (arcos) que vão dos estados actuais para os estados seguintes. O circuito assumirá a natureza de um **contador módulo-6** com as saídas **a, b, c e d** representadas em cada estado do modo especificado no Diagrama de Estados – as saídas activas figuram a 1, as outras a 0.

Exemplo 9-1



Fluxograma-ASM
(ASM Chart).

Em vez do Diagrama de Estados convencional, pode ser utilizado para a representação da evolução dos estados do sistema ao longo dos sucessivos impulsos de relógio, em muitos casos com grande vantagem, um fluxograma que se designa **Fluxograma-ASM** ou **ASM Chart**. O método ASM foi preconizado por Tom Osborne e Chris Clare. A sigla **ASM** significa **ALGORITHMIC STATE MACHINE**. Uma Máquina de Estados Algorítmica é uma abstracção de um circuito sequencial síncrono.

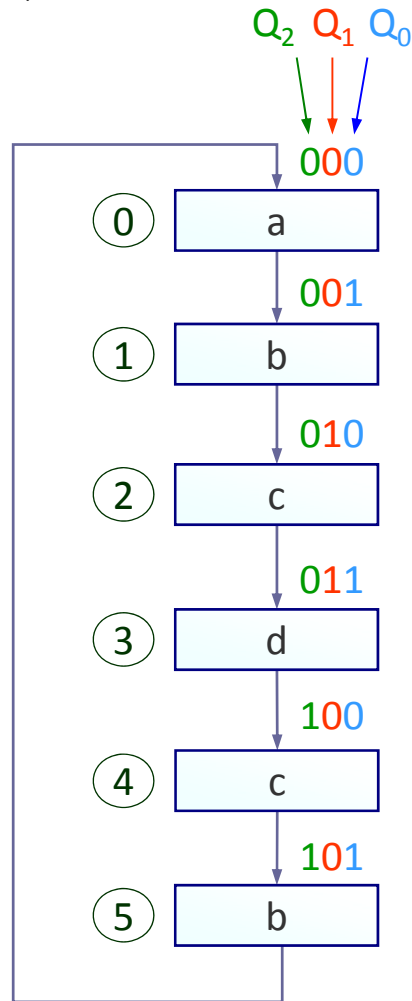
O método ASM permite determinar de forma simples o comportamento do circuito em função do estado actual e das entradas actuais. Tem duas simplificações que permitem melhorar consideravelmente a leitura do fluxograma:

- Em cada estado não são tomadas em consideração todas as variáveis de entrada do circuito, mas apenas aquelas que permitem decidir qual o estado seguinte.
- Em cada estado apenas se indicam as saídas que estão activas.

O Fluxograma-ASM para o circuito que se pretende sintetizar será o indicado ao lado. Cada **ESTADO** é representado por um rectângulo sendo o seu nome inscrito num círculo exterior como indicado.

No que se segue aplicar-se-á o método ASM apenas a circuitos sequenciais síncronos utilizando flip-flops edge-triggered.

Exemplo 9-1



Fluxograma-ASM
(ASM Chart).

A interpretação do Fluxograma-ASM subentende as seguintes considerações:

- A avaliação das variáveis ou expressão booleana de entrada é efectuada na transição ascendente de clock, o que implica que o circuito só pode controlar acontecimentos com duração superior a um período de clock.
- O tempo mínimo que o circuito permanece num estado é um período de clock, pelo que qualquer saída do tipo Moore – função apenas do estado – tem a duração mínima de um período de clock.

Para a síntese do circuito existem diferentes métodos. O método normalmente utilizado prossegue com a codificação dos estados, que pode ser arbitrária, ou imposta pela sequência de contagem pretendida, para diminuir o número de flip-flops. Adiante será exemplificado um método que utiliza um flip-flop por estado, designado de **one-hot**.

No circuito a sintetizar, a existência de 6 estados de contagem impõe a utilização de 3 flip-flops cujas saídas são designadas por Q_0 , Q_1 e Q_2 .

A síntese prossegue com o desenho do **Mapa de Atribuição de Estados (MAE)** e os dois conjuntos de mapas de Karnaugh que descrevem:

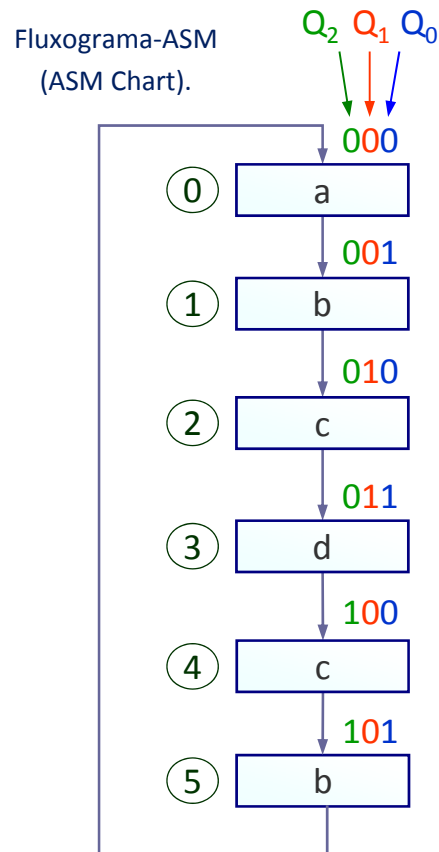
- o primeiro mapa as funções ‘g’ geradoras do Estado Seguinte – as entradas a aplicar aos flip-flops para os forçar às transições indicadas no Fluxograma-ASM;
- o segundo mapa as funções ‘f’ geradoras das Saídas exteriores.

O Mapa de Atribuição de Estados ajuda a situar o quadrado correspondente aos vários estados em todos os mapas de Karnaugh. O preenchimento destes mapas é feito a partir da informação contida no Fluxograma-ASM e na tabela de transição de estados dos flip-flops como ilustrado no slide seguinte.

TABELAS E MAPAS DE KARNAUGH INTERVENIENTES NO PROCESSO DE SÍNTESE (Ex. 9-1-5)

9-11

Exemplo 9-1



Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

Tabela de excitação do flip-flop T.

Estado Actual				Estado Seguinte				Saídas Actuais			
Nome	Q ₂	Q ₁	Q ₀	Nome	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺	d	c	b	a
0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	2	0	1	0	0	0	1	0
2	0	1	0	3	0	1	1	0	1	0	0
3	0	1	1	4	1	0	0	1	0	0	0
4	1	0	0	5	1	0	1	0	1	0	0
5	1	0	1	0	0	0	0	0	0	1	0

Tabela de transição de estados e de saídas.

O preenchimento dos mapas de Karnaugh faz-se da seguinte maneira: no Estado Actual 0, o flip-flop Q₀ passa do estado 0 a 1. Inscreve-se 1 no quadrado do canto superior esquerdo no mapa de Q₀⁺ que representa o valor de cada Estado Seguinte, para o flip-flop Q₀ (mapas no lado direito – para os outros flip-flops tudo se passa de modo idêntico).

Da tabela de Transição de Estados do flip-flop T (também designada de Excitação) infere-se que a entrada T₀ deste flip-flop terá que assumir o valor 1 para que a transição descrita seja efectuada (do estado 0 para o estado 1). Inscreve-se este 1 no quadrado do canto superior esquerdo no mapa de T₀. Idem para a saída actual 'a', que é 1 neste estado. O processo é repetido para Q₁ e Q₂ como patente no slide seguinte, omitindo-se o mapa de Q₀⁺ por ser redundante para o desenho do circuito (são suficientes os mapas de T₀, T₁ e T₂).

MAE

	Q ₀ (1)			
Q ₂ (4)	0	2	3	1
	4	–	–	5

Q₀⁺

	Q ₁ (2)			
Q ₂	1	1	0	0
	1	–	–	0

Mapas MAE e do Estado Seguinte para Q₀.

T₀

	Q ₀			
Q ₂	1	1	1	1
	1	–	–	1

T₀ = 1

a

	Q ₀			
Q ₂	1	0	0	0
	0	–	–	0

a = Q₀' Q₁' Q₂'

Mapas da excitação T₀ e da saída 'a'.



Exemplo 9-1

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

MAE

		Q ₀ (1)	
		0	2
		3	1
Q ₂ (4)		4	5
		-	-
		6	7

Q₁(2)

T₀

		Q ₀	
		1	1
		1	1
Q ₂		1	1
		-	-

Q₁

T₀ = 1

T₁

		Q ₀	
		0	0
		1	1
Q ₂		0	0
		-	-

Q₁

T₁ = Q₀ Q₂'

T₂

		Q ₀	
		0	0
		1	0
Q ₂		0	1
		-	-

Q₁

T₂ = Q₀ Q₂ + Q₀ Q₁ =
= Q₀ (Q₂ + Q₁)

Equações booleanas das entradas de excitação dos flip-flops T deduzidas por mapas de Karnaugh.

Tabela de excitação do flip-flop T – cada linha indica o valor de T requerido para a transição indicada.

a

		Q ₀	
		1	0
		0	0
Q ₂		0	0
		-	-

Q₁

a = Q₀' Q₁' Q₂'

b

		Q ₀	
		0	0
		0	1
Q ₂		0	1
		-	-

Q₁

b = Q₀ Q₁'

c

		Q ₀	
		0	1
		0	0
Q ₂		1	0
		-	-

Q₁

c = Q₀' (Q₁ + Q₂)

d

		Q ₀	
		0	0
		1	0
Q ₂		0	0
		-	-

Q₁

d = Q₀ Q₁

Equações booleanas das saídas a, b, c e d deduzidas por mapas de Karnaugh.



DIAGRAMA LÓGICO DO CIRCUITO (Ex. 9-1-7)

9-13

Exemplo 9-1

Pág. 176

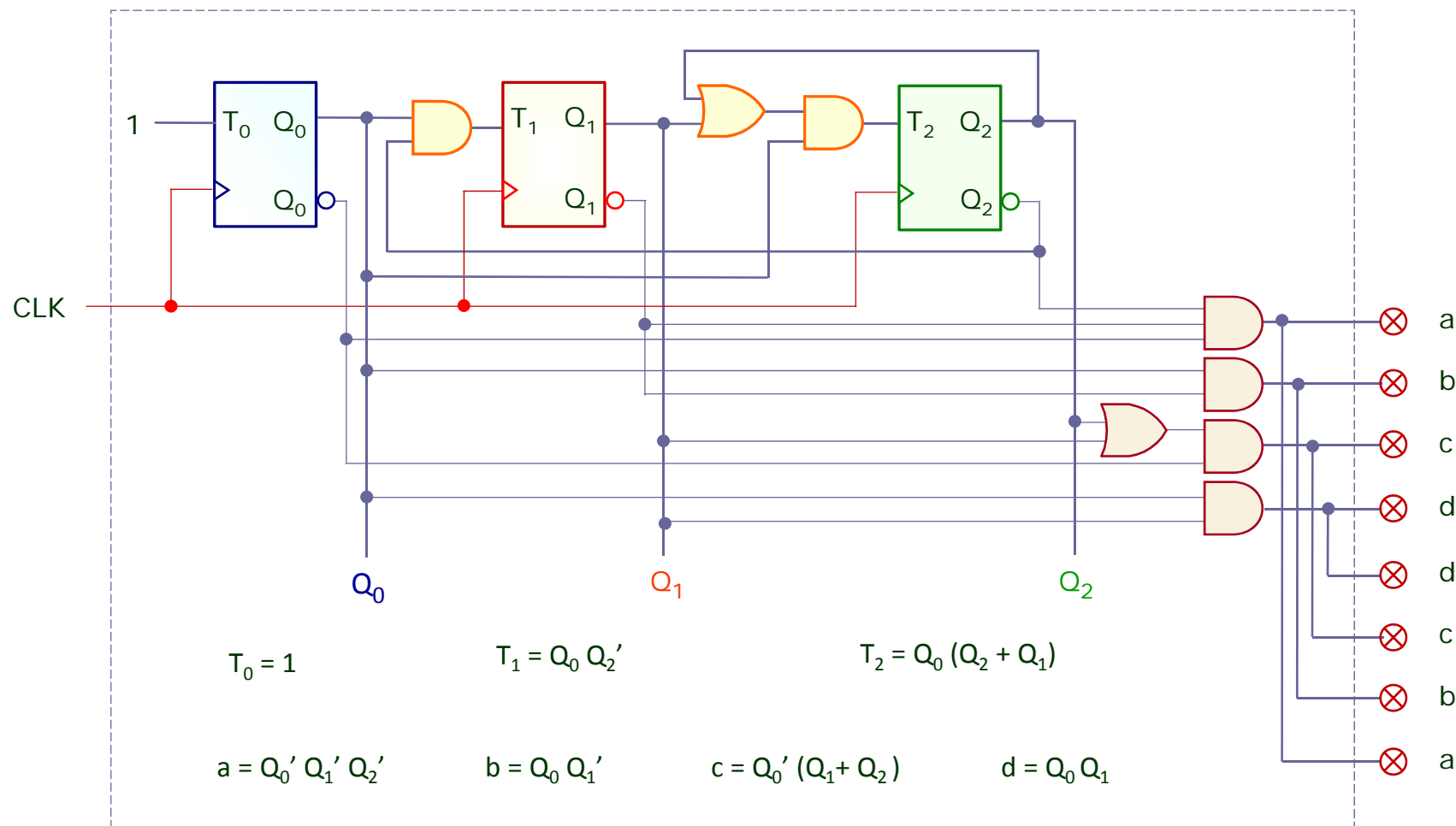


Diagrama lógico do circuito contador módulo-6 que gera o padrão de LEDs pretendido.

ESTADOS NÃO PERMITIDOS (Ex. 9-1-8)

9-14

Exemplo 9-1

MAE	Q ₀			
	0	2	3	1
Q ₂	4	6	7	5
	Q ₁			

Mapa de Atribuição de Estados.

	Q ₂	Q ₁	Q ₀
6	1	1	0
	T ₂ =0	T ₁ =0	T ₀ =1
7	1	1	1
	T ₂ =1	T ₁ =0	T ₀ =1
2	0	1	0

Tabela explicitando as entradas de excitação T dos flip-flops nos estados 6 e 7 não visitados, que decorrem das indiferenças utilizadas e decidem o estado seguinte, ES, da máquina.

Estados não permitidos (nunca visitados em condições normais)

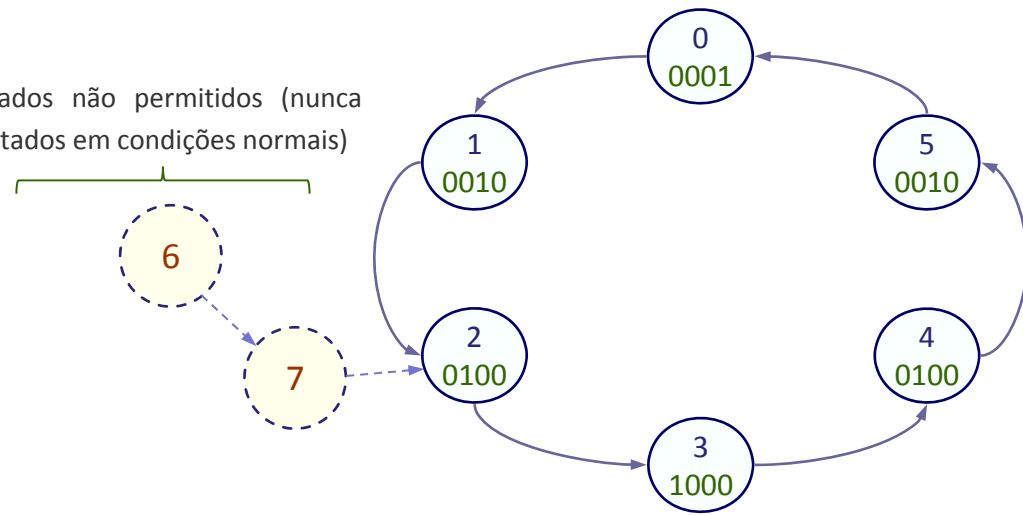


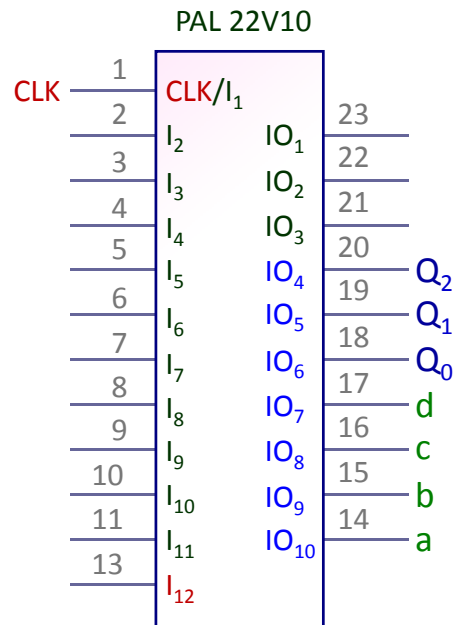
Diagrama de estados assinalando os **estados não permitidos** (6 e 7) no ciclo de contagem, e a sua convergência para a sequência normal ao fim de um máximo de 2 impulsos de relógio decorrente do uso das condições **don't care**.

As configurações 110 e 111 (estados 6 e 7) não são permissíveis no circuito. Constituem condições **don't care** aproveitadas para a simplificação.

Se ocorrerem indevidamente verifica-se pela tabela ao lado a convergência destas configurações para o ciclo normal ao fim de dois impulsos de relógio.



Exemplo 9-1



Símbolo lógico da PAL 22V10 assinalando os pinos de saída utilizados.

Troço do Código CUPL da PAL sequencial correspondente ao contador módulo-6 que gera o padrão de LEDs pretendido, usando a estrutura SEQUENCE.

```

/***** OUTPUT PINS *****/
PIN 14 = a ;
PIN 15 = b ;
PIN 16 = c ;
PIN 17 = d ;
PIN [18..20] = [Q0..2] ;

/***** BODY *****/
[Q0..2].SP='b'0;
[Q0..2].AR='b'0;

SEQUENCE [Q2..0]{
    PRESENT 0
        OUT a ;
        NEXT 1 ;
    PRESENT 1
        OUT b ;
        NEXT 2 ;
    PRESENT 2
        OUT c ;
        NEXT 3 ;
    PRESENT 3
        OUT d ;
        NEXT 4 ;
    PRESENT 4
        OUT c ;
        NEXT 5 ;
    PRESENT 5
        OUT b ;
        NEXT 0 ;
}
    
```

A estrutura SEQUENCE gera as equações de excitação para cada tipo de flip-flop definido. Essas equações ficam visíveis no ficheiro 'doc'.

Sendo o dispositivo alvo a PAL 22V10 há que tomar em consideração que apenas contém um tipo de flip-flop (D). Se o dispositivo suportar mais de um tipo de flip-flop torna-se necessário especificar que tipo de flip-flop se pretende utilizar. Existem as palavras seguintes associadas aos flip-flops seguintes:

- SEQUENCED - flip-flop tipo D
- SEQUENCET - flip-flop tipo T
- SEQUENCEJK - flip-flop tipo JK
- SQUENCERS - flip-flop tipo SR

Por defeito, a palavra SEQUENCE do CUPL assume o flip-flop JK (para os dispositivos alvo com vários tipos de flip-flop).

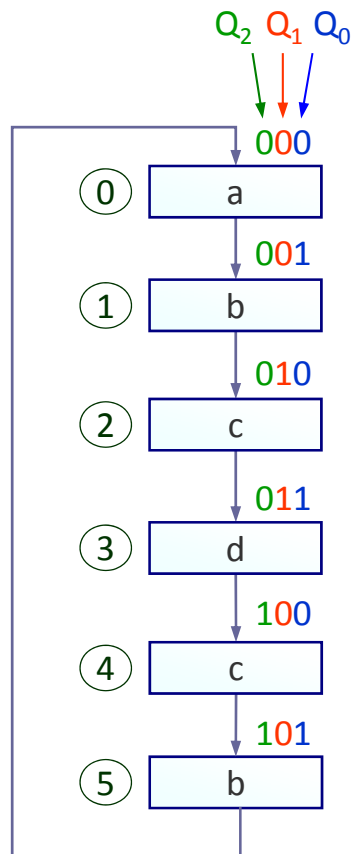
Em vez da utilização da estrutura SEQUENCE podem ser declaradas diretamente as funções associadas às entradas dos flip-flops que compõem cada circuito sequencial.

O slide seguinte refaz o circuito sequencial deste exemplo com flip-flops tipo D para obtenção das equações de excitação a utilizar na descrição de um segmento de código CUPL alternativo à estrutura SEQUENCE.

IMPLEMENTAÇÃO DO CIRCUITO SEQUENCIAL EM PAL SEM ESTRUTURA SEQUENCE (EX. 9-1-10)

9-16

Exemplo 9-1



Fluxograma-ASM
(ASM Chart).

		Q_0			
		1	1	0	0
Q_2	D_0	1	1	0	0
		1	-	-	0

$$D_0 = Q_0'$$

		Q_0			
		0	1	0	1
Q_2	D_1	0	1	0	1
		0	-	-	0

$$D_1 = Q_0' Q_1 + Q_0 Q_1' Q_2'$$

		Q_0			
		0	0	1	0
Q_2	D_2	0	0	1	0
		1	-	-	0

$$D_2 = Q_0 Q_1 + Q_0' Q_2$$

Equações booleanas das entradas de excitação dos flip-flops **D** deduzidas por mapas de Karnaugh.

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

Tabela de excitação do flip-flop D – cada linha indica o valor de D requerido para a transição indicada.

Foram introduzidas duas entradas adicionais no circuito:

- um RESET assíncrono no pino 2;
- um PRESET síncrono no pino 3.

Troço do Código CUPL da PAL sequencial correspondente ao contador módulo-6 que gera o padrão de LEDs pretendido, fazendo uso da declaração directa das funções associadas às entradas dos flip-flops.

```

/***** INPUT PINS *****/
PIN 2 = reset;
PIN 3 = preset;

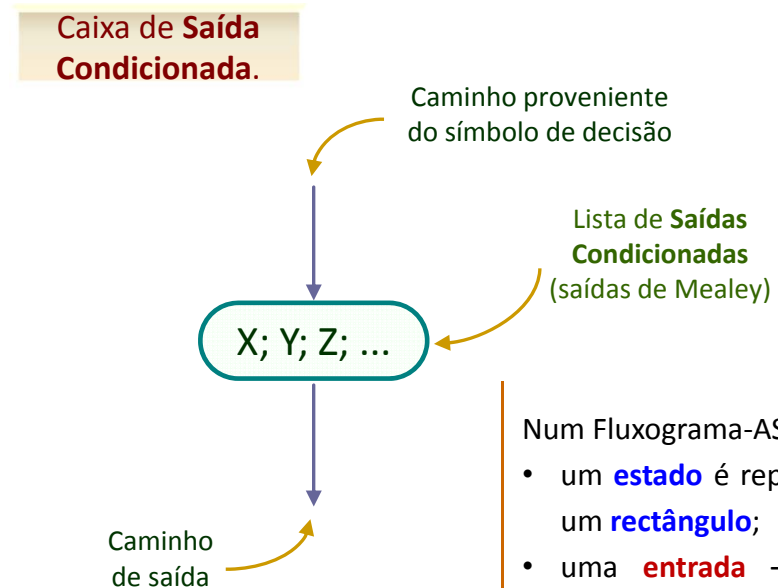
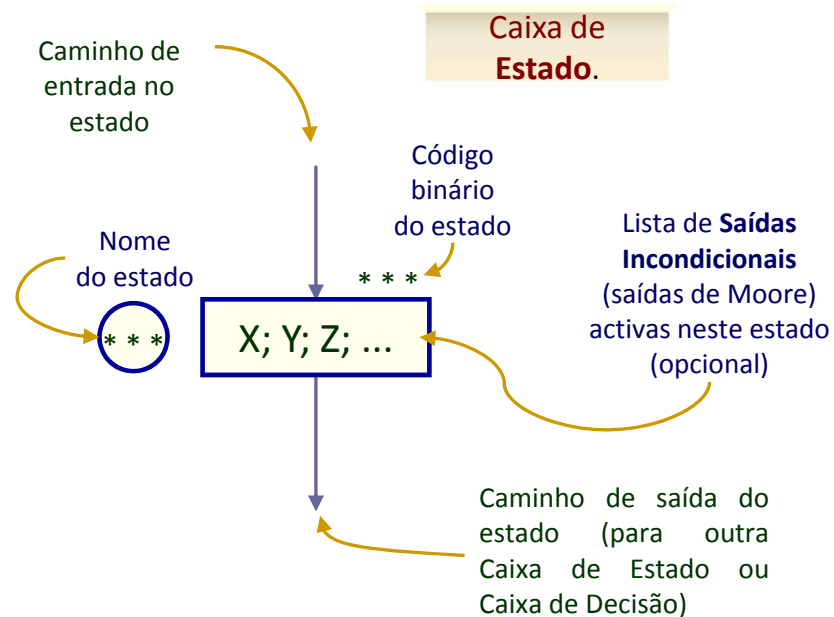
/***** OUTPUT PINS *****/
PIN 14 = a ;
PIN 15 = b ;
PIN 16 = c ;
PIN 17 = d ;
PIN [18..20] = [Q0..2] ;

/***** BODY *****/
[Q0..2].SP = preset;
[Q0..2].AR = reset;

Q0.D = !Q0;
Q1.D = !Q0 & Q1 # Q0 & !Q1 & !Q2;
Q2.D = Q0 & Q1 # !Q0 & Q2;

a = !Q0 & !Q1 & !Q2;
b = Q0 & !Q1;
c = !Q0 & (Q1 # Q2);
d = Q0 & Q1;
    
```

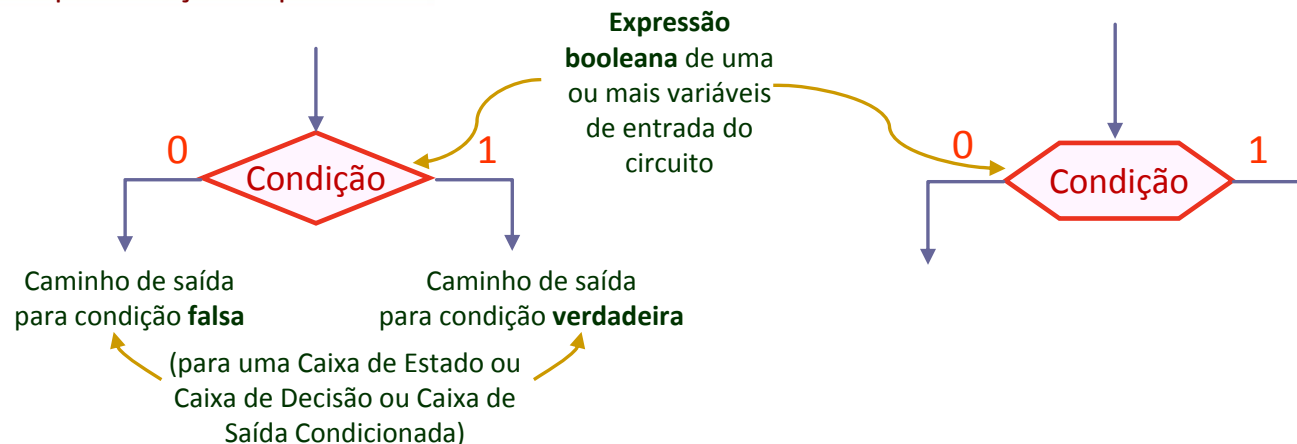


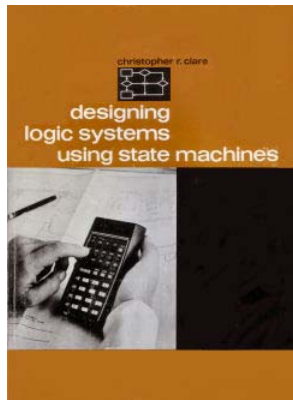


Num Fluxograma-ASM:

- um **estado** é representado por um **retângulo**;
- uma **entrada** – variável ou expressão booleana – é representada por um **losango** – símbolo de decisão;
- uma **saída** é representada dentro do rectângulo de estado (saídas **incondicionais** ou de **Moore**), ou dentro de um símbolo ovalado formado por um rectângulo com os lados substituídos por arcos de circunferência (saídas **condicionadas** ou de **Mealey**).

Caixa de Decisão – duas representações equivalentes.





Chris Clare e o livro que publicou em 1973.



Tom Osborne (ao lado) desenvolveu a técnica de projecto de circuitos sequenciais que designou **Algorithmic State Machine (ASM)**. Em 1965, utilizou-a para criar o protótipo que deu origem à primeira calculadora de mesa electrónica, a HP 9100, apresentada ao mundo em 1969.



Cortesia da Hewlett Packard.

Bill Hewlett em 1970 (à esquerda) oferecendo um HP 9100A a Arthur Clark (que a alcunhou de HAL – o **HAL** 9000 era o **Heuristically** programmed **Algorithmic** computer do filme de 1968 2001: A Space Odyssey)

A notação ASM foi criada por Thomas Osborne pela experiência adquirida no projecto de equipamento electrónico na Universidade de Berkeley. Foi formalizada e divulgada por Christopher R. Clare da Hewlett Packard através da sua obra de 1973 designada *Designing Logic Systems Using State Machines*.

Num Fluxograma-ASM pode observar-se de forma simples o comportamento de um sistema função do estado presente e das entradas. Em cada estado não são tomadas em consideração todas as variáveis de entrada do circuito, mas apenas as relevantes para determinar o próximo estado. E quanto às saídas, só são evocadas as que naquele estado estão activas.

No prefácio da edição original Thomas Osborne escreve:

...You will find that the material in this book is considerably different from material in previous books discussing logic design. ...A dozen years have passed since I discovered the basic design process. ...

*The ASM chart separates the conceptual phase of a design from the actual circuit implementation. It gives the designer the freedom of expressing very complex algorithms without worrying about the circuit implementation. ...Initially you may be bewildered when you see that **an ASM looks like a conventional flow chart but is used differently.***

*In an ASM chart the operations described within a box and all branching associated with the box are **executed simultaneously**, not sequentially as in a flow chart. The reasons for this convention will become evident as you read...*

Exemplo 9-2

OBJECTIVO

Pretende-se projectar com flip-flops T um contador que, por acção de um comutador E, passe de módulo-8 para módulo-5 de acordo com as condições seguintes:

- E=0 módulo-8
- E=1 módulo-5.

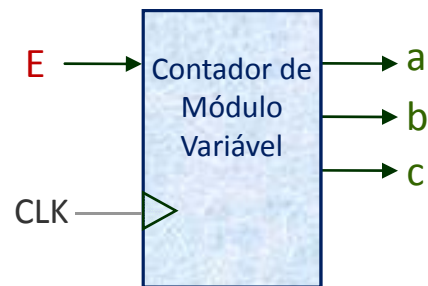
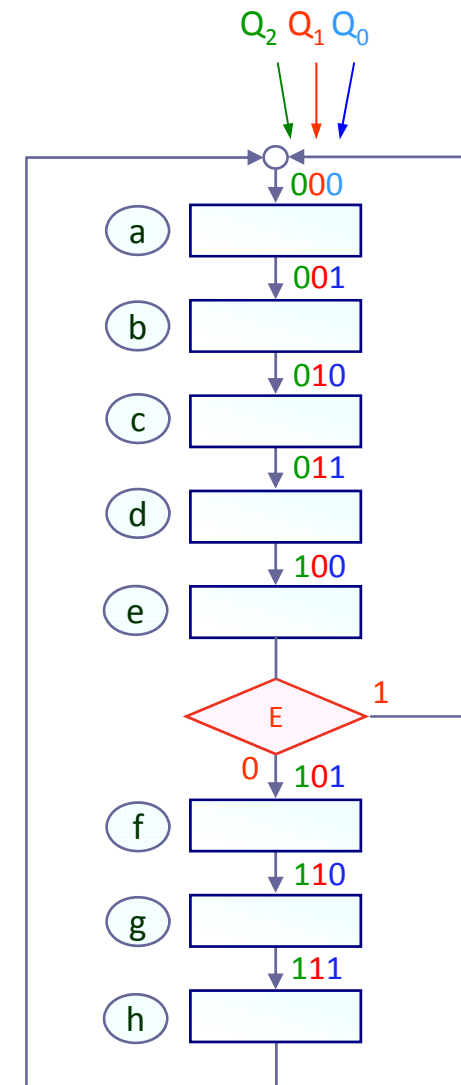


Diagrama de blocos do circuito sequencial.

O símbolo de losango representa um teste sobre a variável **E**, do qual depende a evolução do estado do circuito.

A diferença principal entre a Caixa de Estado e o Círculo de um Diagrama de Estados reside no facto de uma Caixa de Estado ter só um caminho de saída, que vai ao encontro de uma Caixa de Estado ou de uma Caixa de Decisão.

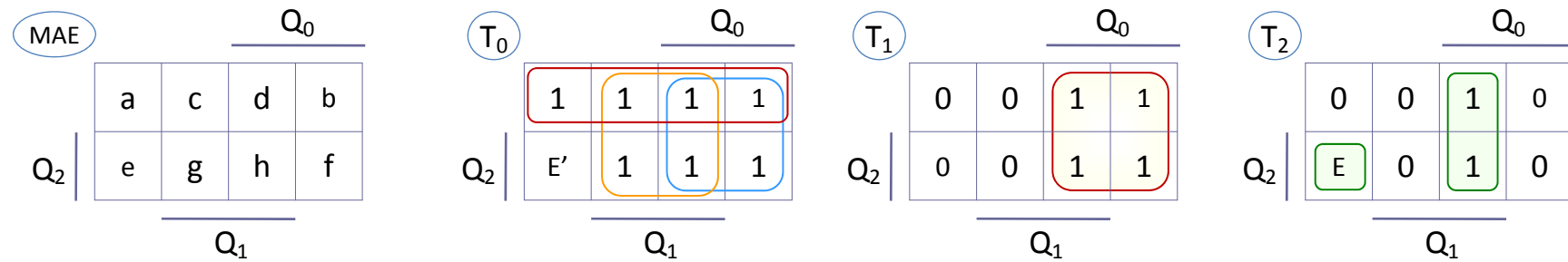


Fluxograma-ASM do contador.

CONTADOR DE MÓDULO VARIÁVEL (EX. 9-2-2)

9-20

Exemplo 9-2



Nos mapas de Karnaugh com variáveis inseridas há que:

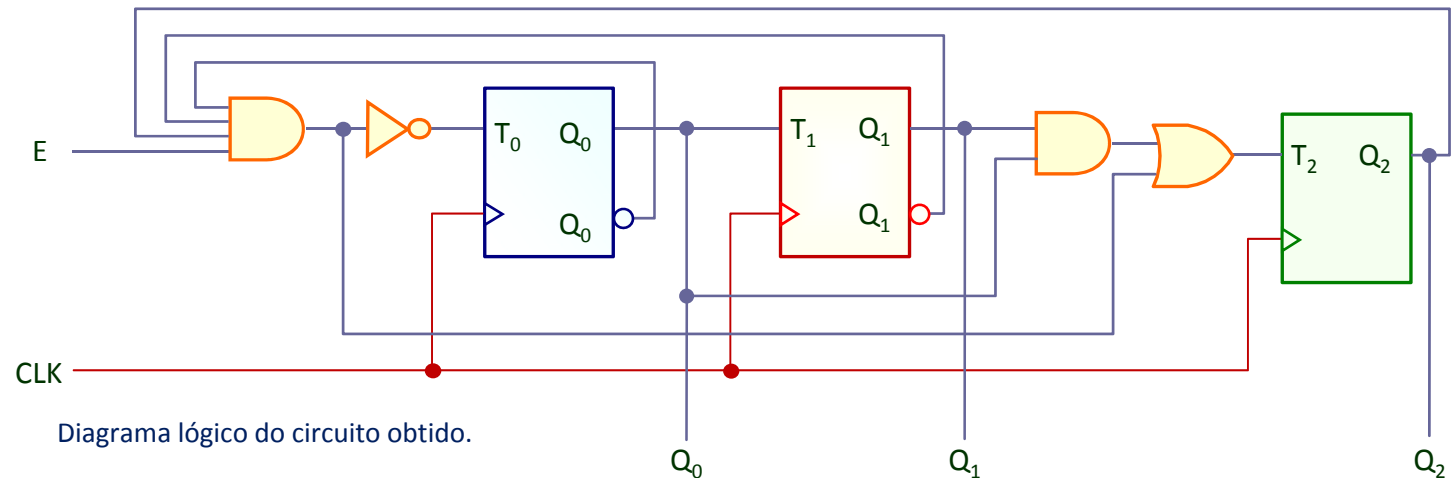
- Extrair os 1s presentes no mapa, considerando como 0 as variáveis e aproveitando as eventuais indiferenças para simplificação;
- Extrair as variáveis inseridas, considerando os 1s como indiferenças.

$$T_0 = (Q_0 + Q_1 + Q_2') + E' = (Q_0'Q_1'Q_2)' + E' = (Q_0'Q_1'Q_2 \cdot E)'$$

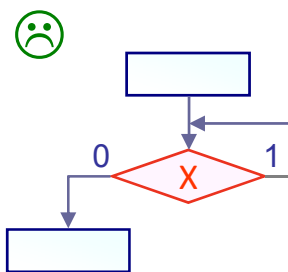
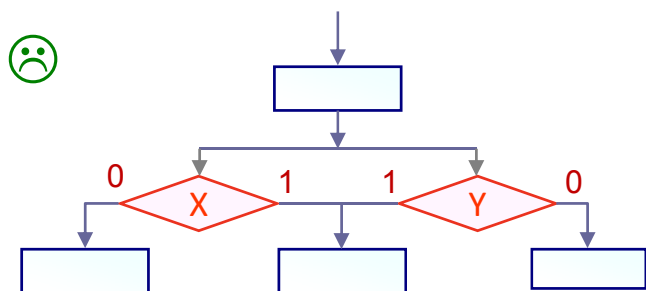
$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1 + Q_0'Q_1'Q_2 E$$

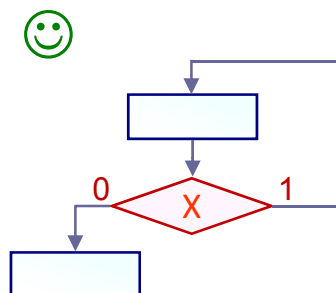
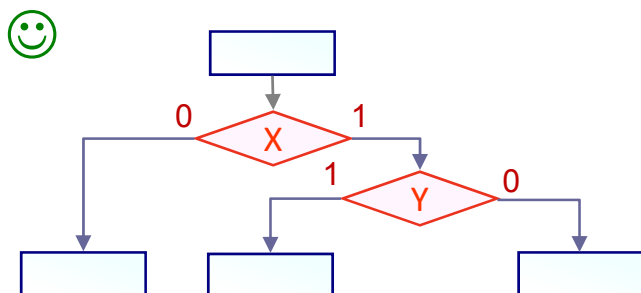
Mapa de atribuição de estados (MAE) e equações booleanas T_0 , T_1 e T_2 .



Incorrectos

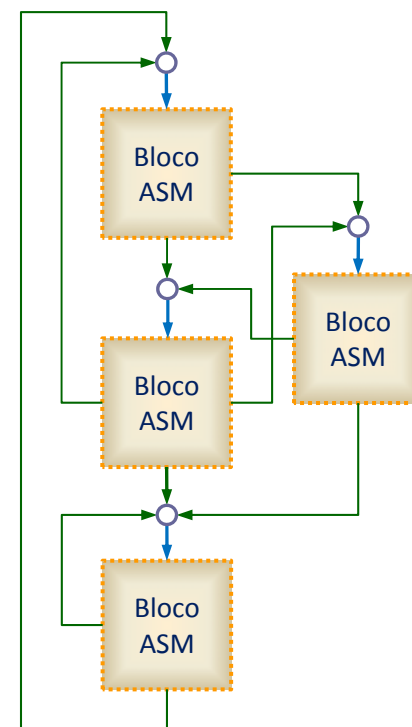


Correctos

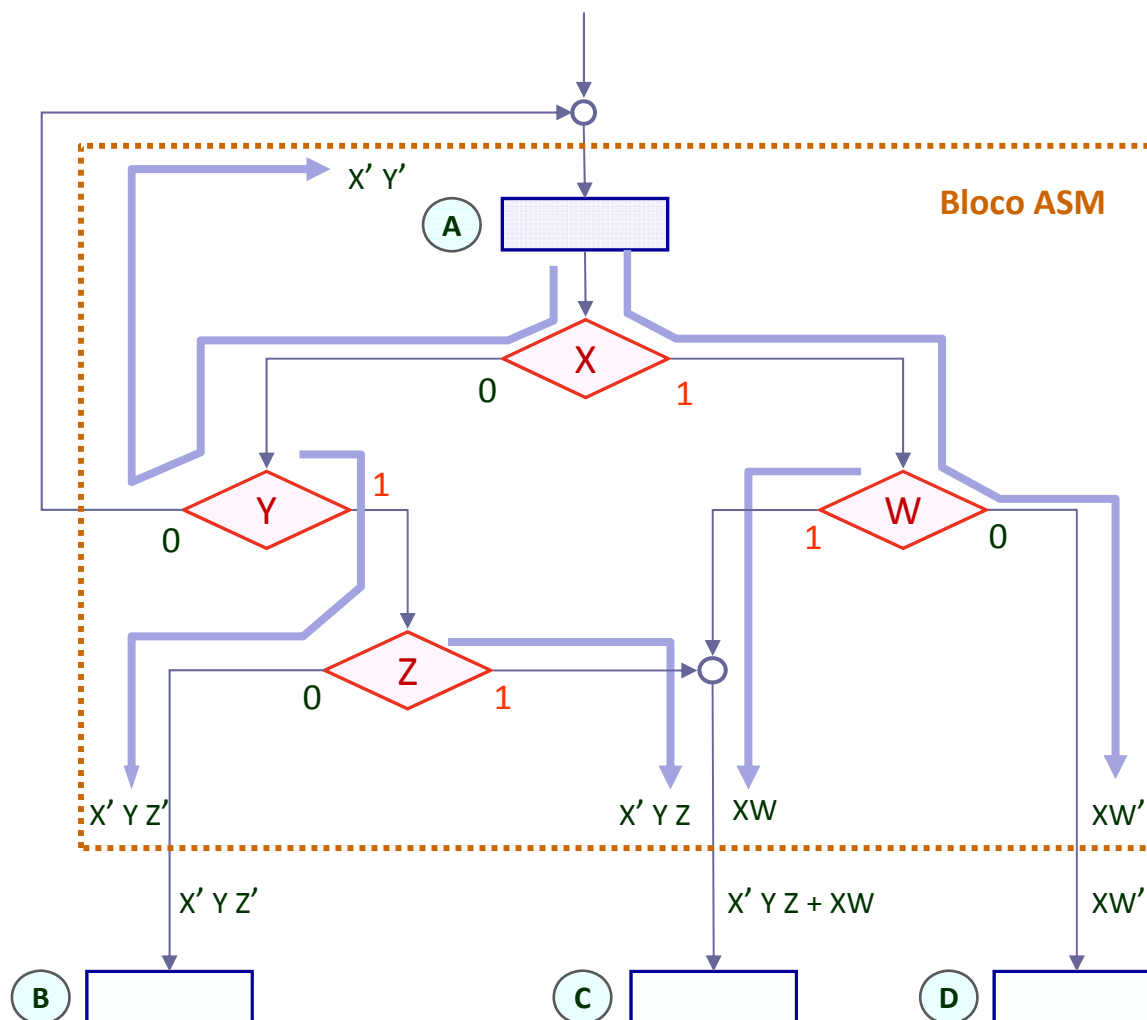


No desenho de cima, com as Caixas de Decisão **em paralelo**, há ambiguidade na definição das transições (por exemplo a condição $X=Y=0$ origina dois Estados Seguintes). Em baixo, as caixas de Decisão em **série** desfazem a ambiguidade.

Cada 'loop' deve conter pelo menos uma caixa de Estado.

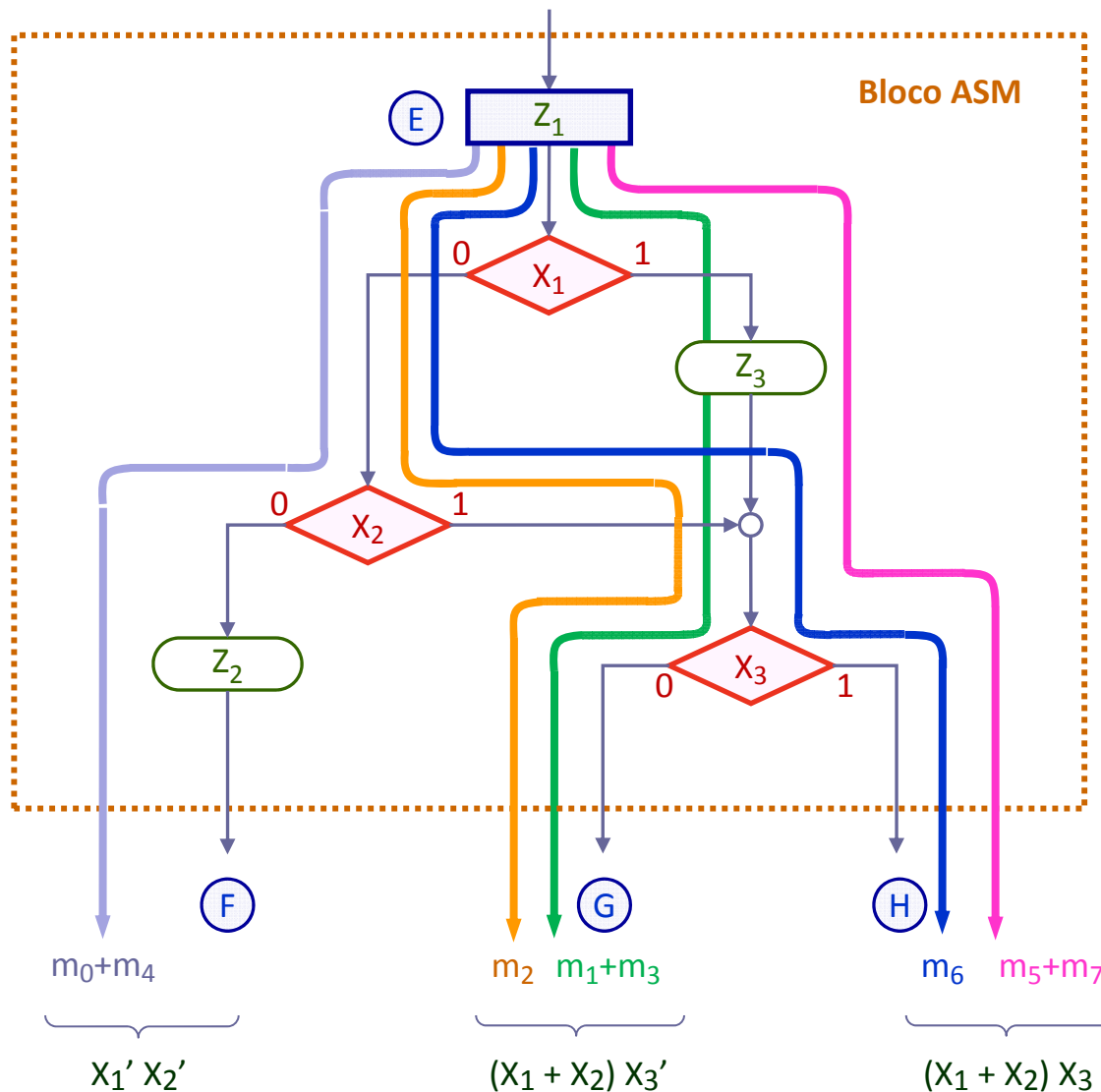


Cada Bloco ASM contém uma Caixa de Estado que define um estado, e a lógica das funções 'g' geradoras do Estado Seguinte, e 'f' geradoras das Saídas nesse estado. Cada Bloco ASM só pode conter um caminho de entrada (a azul), e vários caminhos de saída para o mesmo ou outros blocos (a verde).



Um bloco ASM contém uma caixa de estado e todas os símbolos de decisão, e de saídas condicionadas, ligados entre a caixa de estado e os caminhos de saída.

Bloco ASM com quatro caminhos de saída do estado – as transições condicionadas do **Estado Actual A** conduzem a um dos quatro **Estados Seguintes**: A, B, C ou D.



Mintermos	X_3	X_2	X_1
m_0	0	0	0
m_1	0	0	1
m_2	0	1	0
m_3	0	1	1
m_4	1	0	0
m_5	1	0	1
m_6	1	1	0
m_7	1	1	1

Tabela de mintermos listando as 8 combinações das 3 entradas X_3, X_2, X_1 .

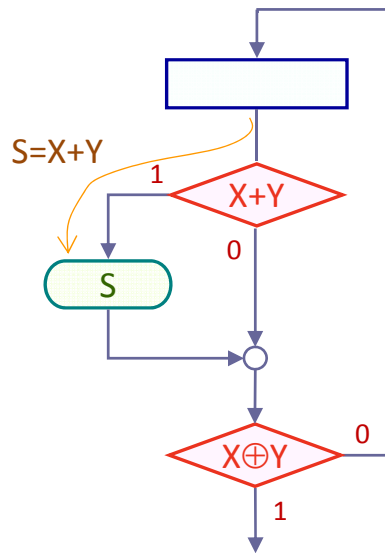
As setas da linha a cinzento fina indicam os 3 caminhos de saída do Estado Actual **E** (representado neste Bloco-ASM) para os Estados Seguintes **F**, **G** e **H**.

Havendo 3 variáveis de entrada X_3, X_2, X_1 , haverá 2^3 transições possíveis a partir deste estado.

Os mintermos (da tabela em cima) correspondentes a cada uma dessas transições têm de estar incluídos, em exclusão mútua, num dos 3 caminhos de saída – a correspondência é a indicada. A expressão de transição para o Estado Seguinte **F** envolve 2 mintermos (m_0 e m_4), enquanto que cada uma das outras duas, para **G** e **H**, envolve 3 mintermos (respectivamente m_1, m_2, m_3 e m_5, m_6, m_7).

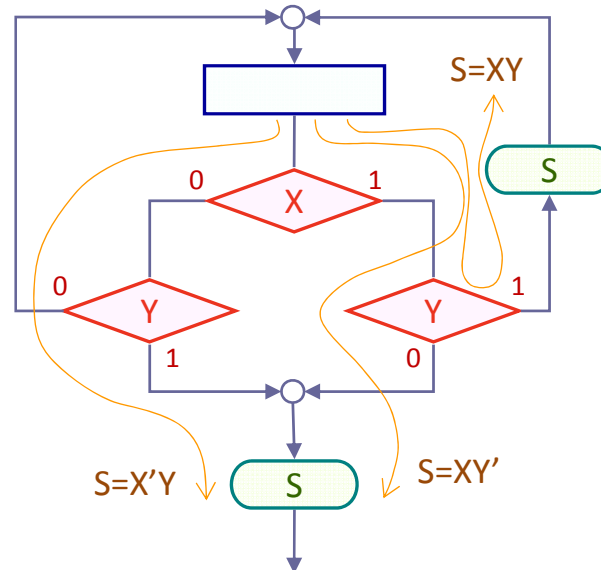
Bloco ASM com três caminhos de saída do estado – as transições condicionadas do Estado Actual **E** conduzem a três Blocos ASM **F**, **G** e **H**.

Saída S de Mealey condicionada por 2 variáveis de entradas **X** e **Y** em 3 grafismos diferentes mas equivalentes com duas e três Caixas de Decisão: as expressões obtidas para **S** são as mesmas após simplificação nos 3 blocos ASM.



		X	
Y	1	1	
	0	1	1

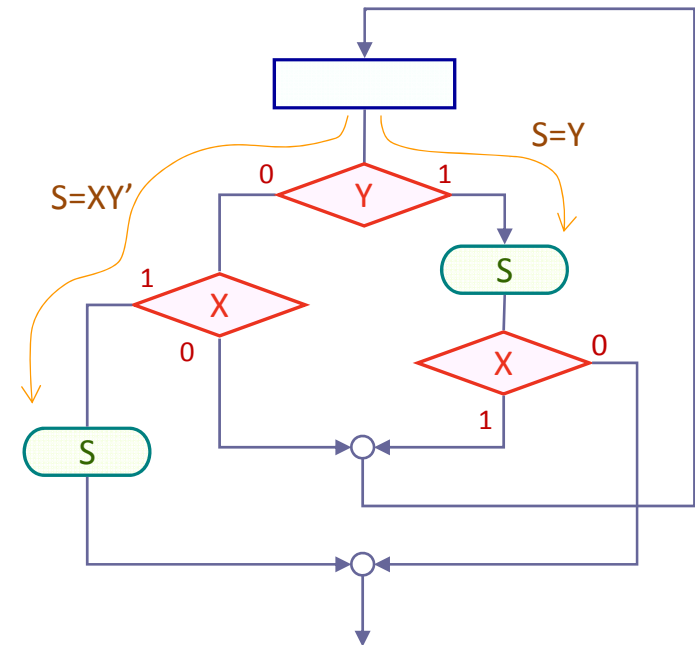
$$S = X + Y$$



		X	
Y	1	1	
	0	1	

		X	
Y	1		1
	0		

$$S = (X'Y) + (XY') + (XY) = X + Y$$



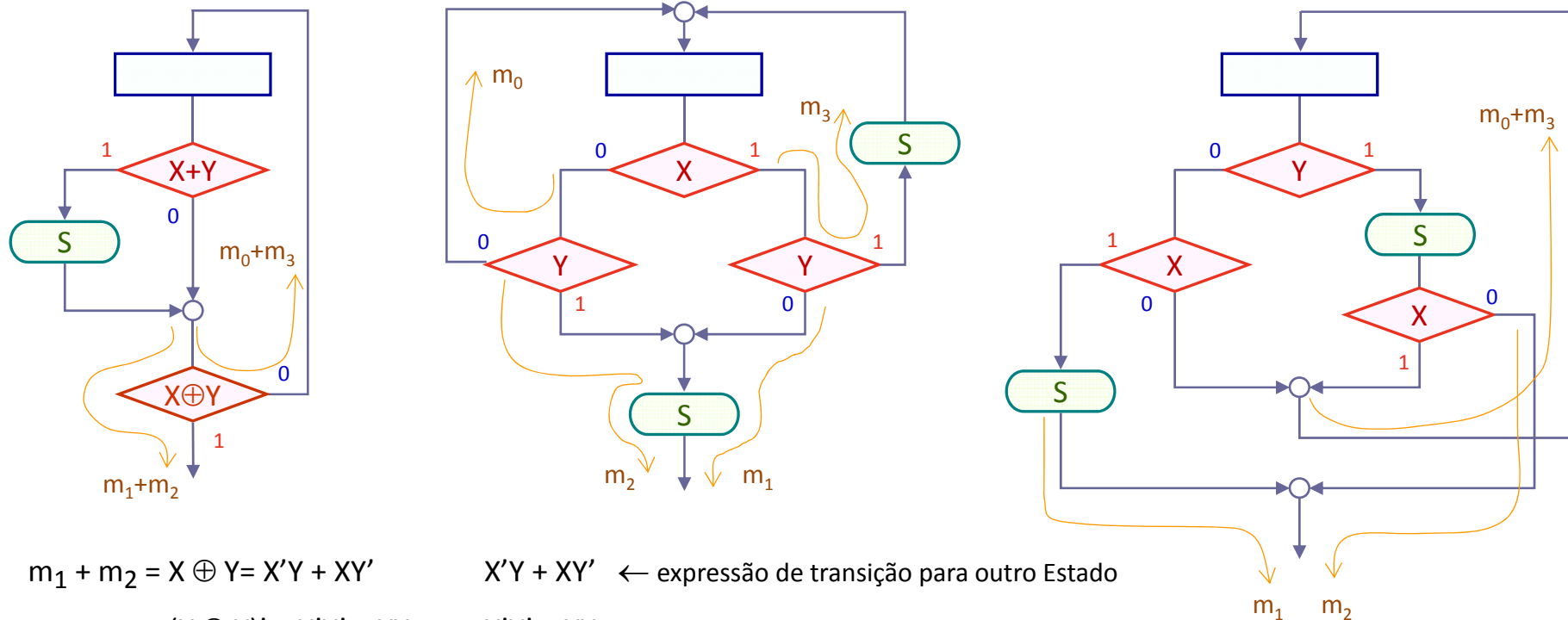
		X	
Y	1	1	
	0		

		X	
Y	1	1	1
	0		

$$S = (XY') + Y = X + Y$$



O mesmo Bloco-ASM do slide anterior é representado em 3 grafismos diferentes com duas e três Caixas de Decisão e duas variáveis de entrada X e Y: as expressões de transição de estado são as mesmas após simplificação.



$$m_1 + m_2 = X \oplus Y = X'Y + XY'$$

$X'Y + XY'$ ← expressão de transição para outro Estado

$$m_0 + m_3 = (X \oplus Y)' = X'Y' + XY$$

$X'Y' + XY$ ← expressão de transição para o mesmo Estado

Dois **Blocos-ASM** serão portanto **equivalentes** se:

- contiverem as mesmas **Saídas** incondicionais na **Caixa de Estado** (saídas de Moore);
- as mesmas **Saídas** condicionadas (saídas de Mealey) estiverem listadas em caminhos com a mesma expressão de transição;
- expressões de transição idênticas conduzirem ao mesmo **Estado Seguinte**.

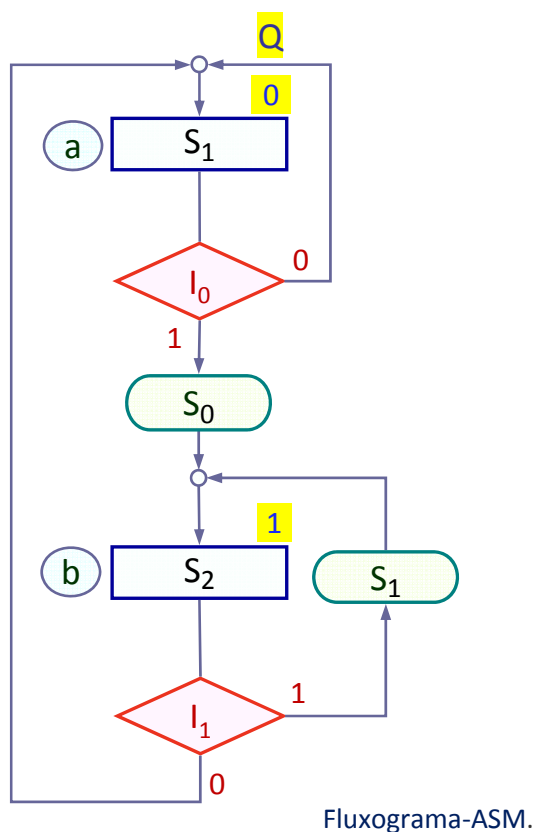
Mintermos	Y	X
m_0	0	0
m_1	0	1
m_2	1	0
m_3	1	1



Exemplo 9-3

OBJECTIVO

Para a máquina sequencial com o Fluxograma-ASM da figura em baixo calcular as expressões lógicas necessárias e desenhar o circuito que tenha o comportamento nela descrito tendo por base flip-flops tipo-D.



Fluxograma-ASM.

Antes de iniciar a síntese desenham-se o diagrama de blocos e a estrutura interna correspondente da forma indicada:

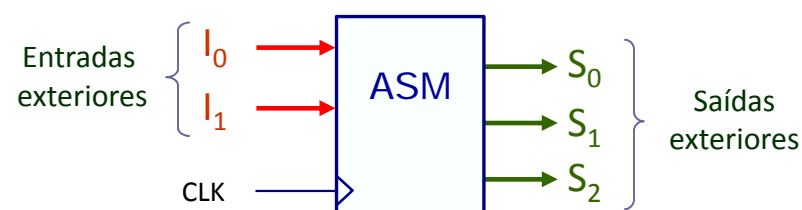
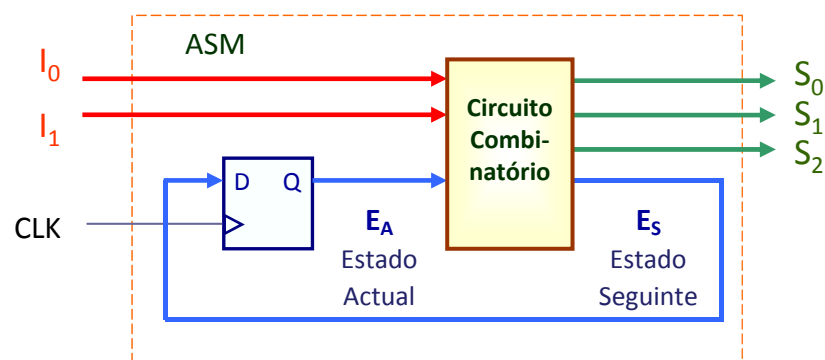


Diagrama de Blocos.



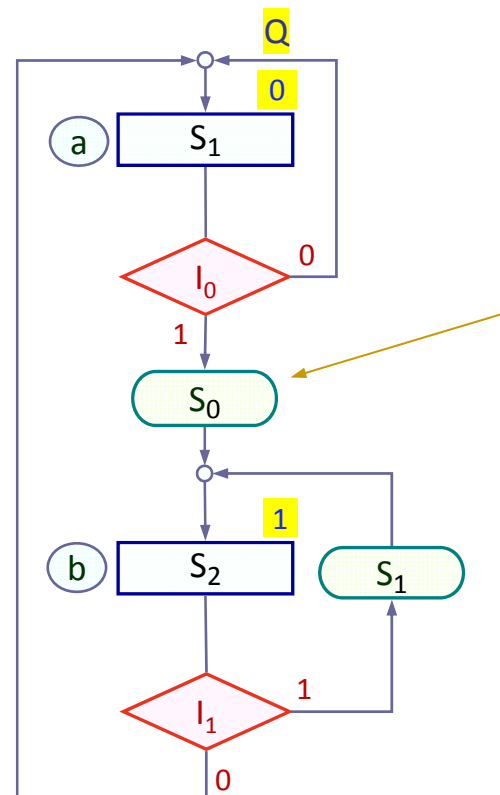
Estrutura interna do circuito que se pretende sintetizar, evidenciando a azul a função geradora de estado.

Exemplo 9-3

Mais uma vez se salienta que no Fluxograma-ASM, e ao contrário da tabela de transição de estados e saídas, apenas se indicam para cada estado :

- As **saídas activas** nesse estado, que dependem ou não das entradas (neste caso I_0 ou I_1).
- As **transições** para os estados seguintes, dependentes ou não de entradas com significado nesse estado.

Esta simplificação constituiu uma das inovações da notação ASM, quando surgiu, como já mencionado.



A saída S_0 listada é gerada **continuamente** quando a máquina ASM está no estado **a** e a entrada I_0 está activa ($I_0=1$), e não apenas durante a transição para o estado seguinte como a notação do Fluxograma-ASM poderá sugerir.

Fluxograma-ASM (ponto de partida para a síntese do circuito).

Exemplo 9-3

A síntese prossegue de modo semelhante ao do projecto do circuito contador do Exemplo 9-1. Começa por construir-se a tabela de transição de estados e de saídas, usando a informação do ASM, como indicado:

Estado Actual	Entradas Actuais		Estado Seguinte	Saídas Actuais		
Q	I ₁	I ₀	Q ⁺	S ₂	S ₁	S ₀
a (0)	–	0	a (0)	0	1	0
	–	1	b (1)	0	1	1
b (1)	0	–	a (0)	1	0	0
	1	–	b (1)	1	1	0

Tabela de transição de estados e de saídas.

A síntese envolve a determinação das **equações de transição**, correspondentes às **funções geradoras do Estado Seguinte**, que definem cada variável do Estado Seguinte em função das variáveis do Estado Actual e das Entradas:

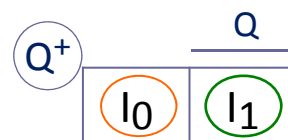
$$Q_k^+ = \text{expressão} = g_k(EA, E)$$

A partir destas equações há que determinar as **equações de excitação** que determinam as entradas de cada flip-flop. Se a máquina de estados utilizar como elementos de memória flip-flops D, a equação de excitação para cada entrada D_k utiliza a mesma expressão que a equação de transição:

$$Q_k^+ = D_k = \text{expressão}$$

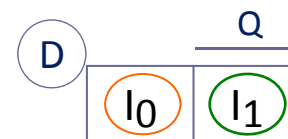
Para outros tipos de flip-flops a passagem das equações de transição para as equações de excitação faz-se através do uso da equação característica do flip-flop, ou da sua tabela de excitação usando mapas de Karnaugh.

Da tabela de transição de estado e de saídas constrói-se o m-K da equação de transição, $Q^+ = g(Q, I_0, I_1)$, que coincide para o flip-flop D com o m-K da equação de excitação:



$$Q^+ = Q' I_0 + Q I_1$$

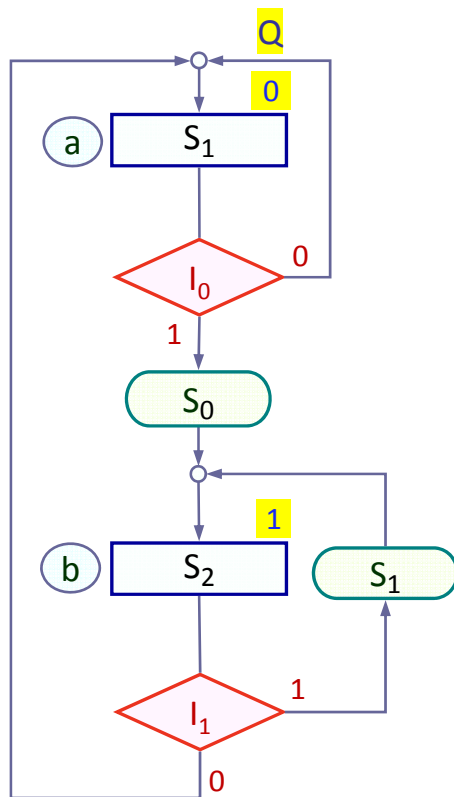
M-K e equação da função geradora do Estado Seguinte (equação de transição).



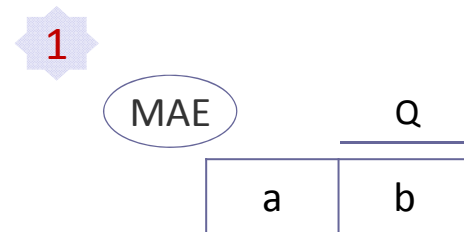
$$Q^+ = D = Q' I_0 + Q I_1$$

M-K e equação de excitação do flip-flop D (idêntica à anterior).

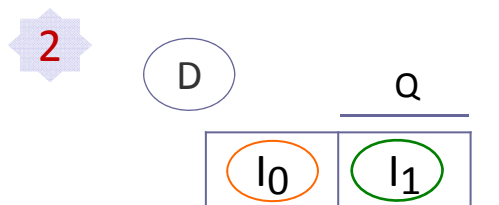
Exemplo 9-3



Fluxograma-ASM.

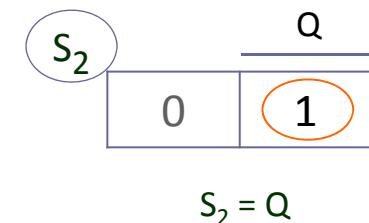
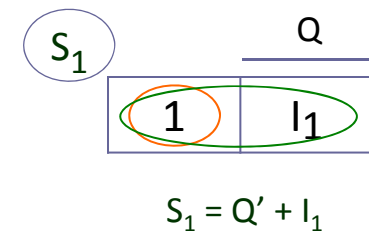
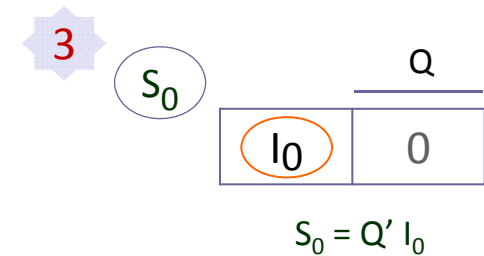


Mapa de Atribuição de Estados.



$$Q^+ = D = Q' I_0 + Q I_1$$

Mapa da equação de excitação.

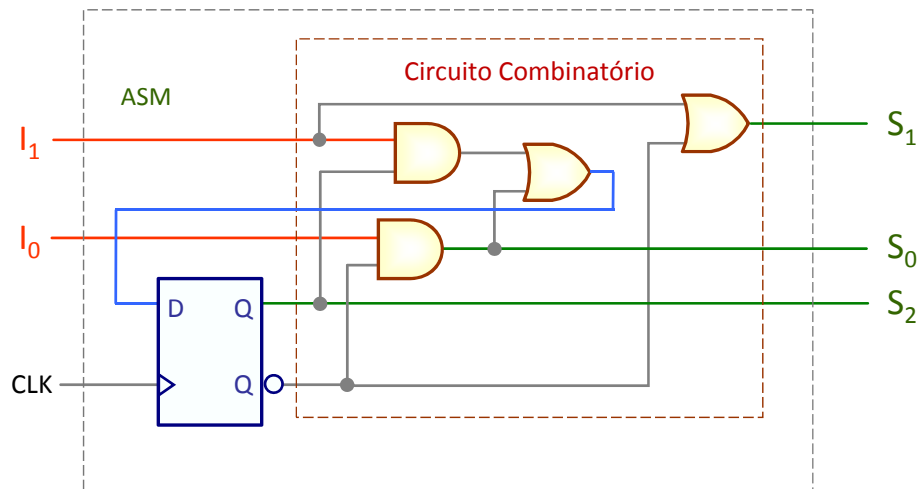


Mapa das funções geradoras das **Saídas** S_0, S_1 e S_2 .

A construção da tabela de transição de estados e de saídas não se revela um passo essencial para a síntese: é possível passar directamente do Fluxograma-ASM para os mapas de Karnaugh de onde se retiram as equações de excitação dos flip-flops, e das saídas, como se exemplifica. Resumindo a ilustração dos passos necessários para a síntese: em **(1)** configura-se o MAE, em **(2)** determinam-se as **equações de excitação** dos flip-flops por inferência directa do Fluxograma-ASM, usando mapas de Karnaugh, em **(3)** as **equações das saídas**, e em **(4)** desenha-se o diagrama lógico.

Exemplo 9-3

4



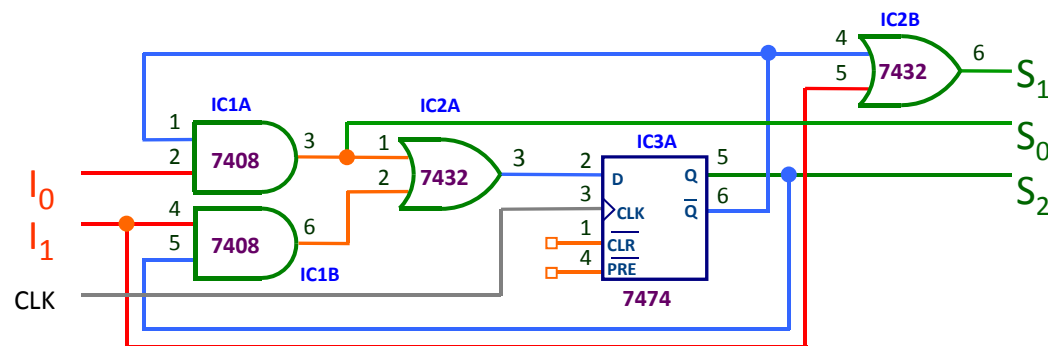
$$D = Q' I_0 + Q I_1$$

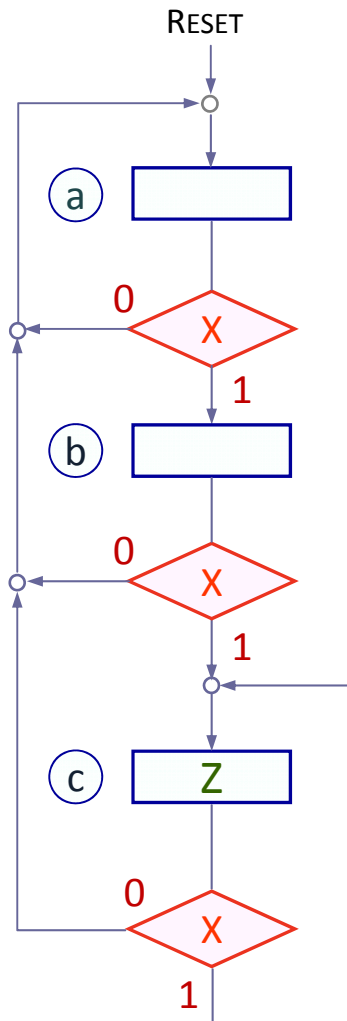
$$S_0 = Q' I_0$$

$$S_1 = Q' + I_1$$

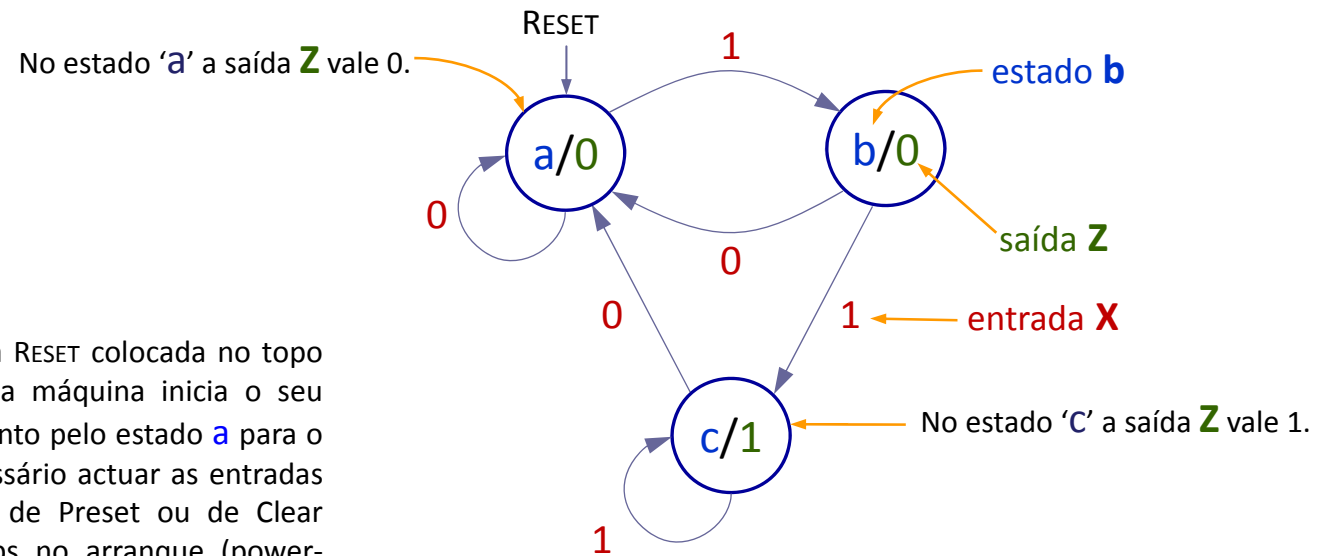
$$S_2 = Q$$

Equações booleanas da entrada de excitação **D** do flip-flop e das saídas **S₀**, **S₁** e **S₂** deduzidas por mapas de Karnaugh no slide anterior.





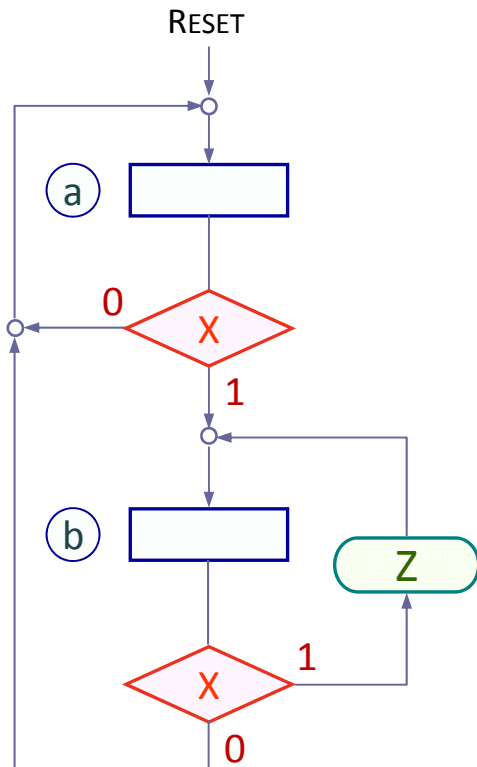
Para uma máquina de Moore Junto a cada arco coloca-se uma indicação do valor lógico das **entradas** externas (a vermelho nas figuras) que ocasionam essa transição. Como as **saídas** são sempre incondicionais (Moore) o seu valor é indicado dentro de cada círculo do estado, separado da referência de estado por uma barra (/).



A referência RESET colocada no topo indica que a máquina inicia o seu funcionamento pelo estado **a** para o que é necessário actuar as entradas assíncronas de Preset ou de Clear dos flip-flops no arranque (power-on) de acordo com a configuração pretendida para o estado **a**.

Grafo de Estados ou Diagrama de Estados da máquina de Moore de acordo com as convenções utilizadas – neste caso só existe uma **entrada X** e uma **saída Z** a representar nas figuras, mas é possível representar várias entradas e saídas.

Fluxograma-ASM de uma máquina de Moore só com saídas dependentes apenas do estado.



Fluxograma-ASM de uma máquina de Mealey (tem uma saída **Z** de Mealey, dependente do estado **b** e da entrada **X** valer 1).

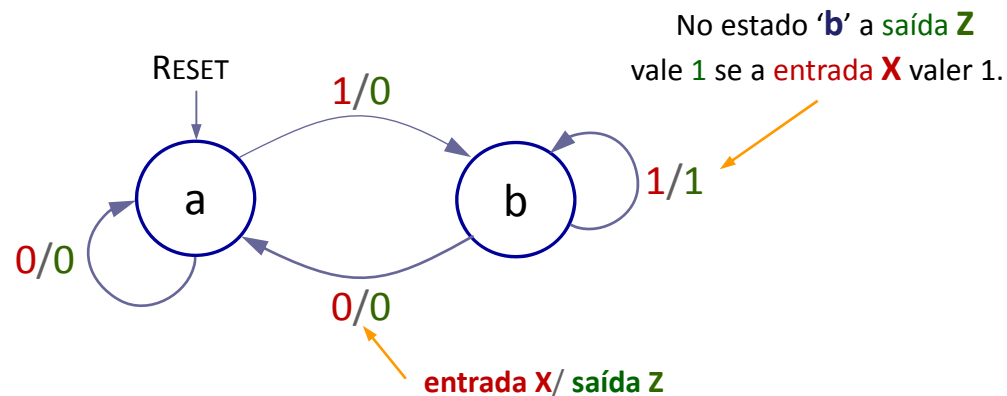
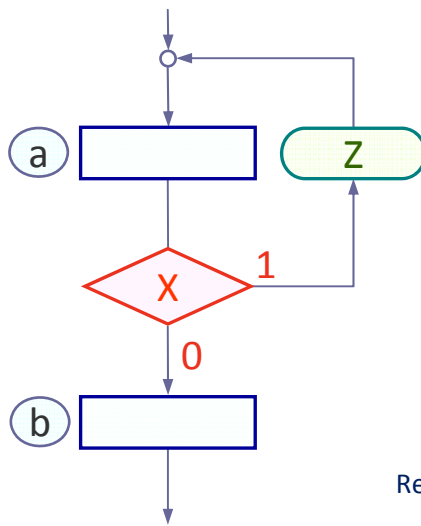


Diagrama de Estados da máquina de Mealey de acordo com as convenções utilizadas - contém a mesma informação que o Fluxograma-ASM.

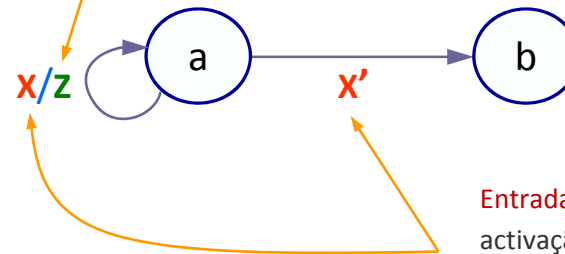
No Diagrama de Estados de uma máquina de Mealey junto a cada arco coloca-se uma indicação do valor lógico (ou expressão lógica) das **entradas** externas (representadas a **vermelho**) que condicionam essa transição, separada por uma barra (/) das **saídas** (representadas a **verde**) que lhe estão associadas ficando visível a influência das entradas na geração dessas saídas.

Ao contrário da representação convencionada para a máquina de Moore em que a indicação das saídas, sempre incondicionais, é feita dentro de cada círculo representativo do estado, as saídas são agora representadas junto a cada arco de transição como indicado. No caso das saídas incondicionais existentes numa máquina de Mealey há que repetir a indicação dessa saída em cada vector de saída do estado – slide seguinte.

Se a transição for incondicional a partir de um estado omite-se a referência da entrada junto ao vector de saída desse estado.

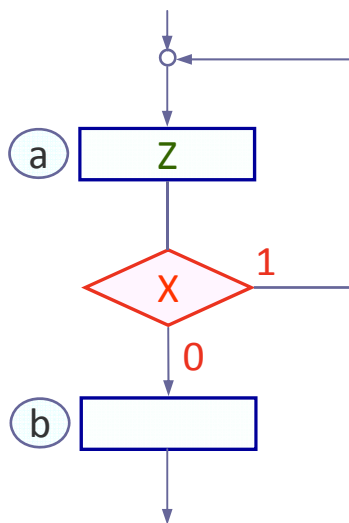


Activação da Saída Actual **Z** no Estado Actual **a** **condicionada** pela condição $X=1$.

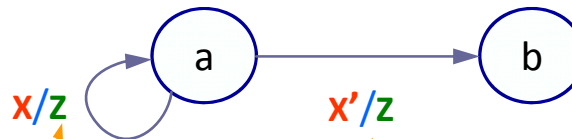


Entrada **X** no Estado Actual **a** a condicionar a activação da Saída Actual **Z** e a transição para o Estado Seguinte (que pode ser **a** ou **b**).

Representação em Fluxograma-ASM e em Diagrama de Estados de uma saída **Z** **condicionada**.



Activação da Saída Actual **Z** **incondicional** no Estado Actual **a** – esta saída é sempre activada independentemente do valor de **X** e por isso é representada a **verde** em todos os arcos de transição que partem do Estado **a**.



Na representação mais comum coloca-se junto a cada seta que parte de um Estado uma expressão que representa a combinação dos valores lógicos das Entradas que produzem essa transição. E são indicadas apenas as Saídas activas nesse estado, incondicionais e condicionadas, separadas das entradas através do símbolo '/', e usando o critério definido.

Representação em Fluxograma-ASM e em Diagrama de Estados de uma saída **Z** **incondicional**.

Exemplo 9-4

OBJECTIVO

Projectar a máquina de estados descrita pelo Fluxograma-ASM da figura ao lado, tendo Q_1 por base um flip-flop T e Q_0 um flip-flop D.

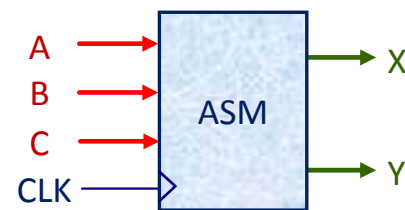


Diagrama de Blocos.

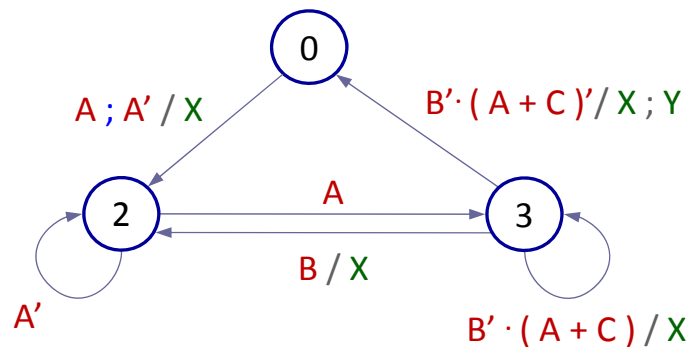
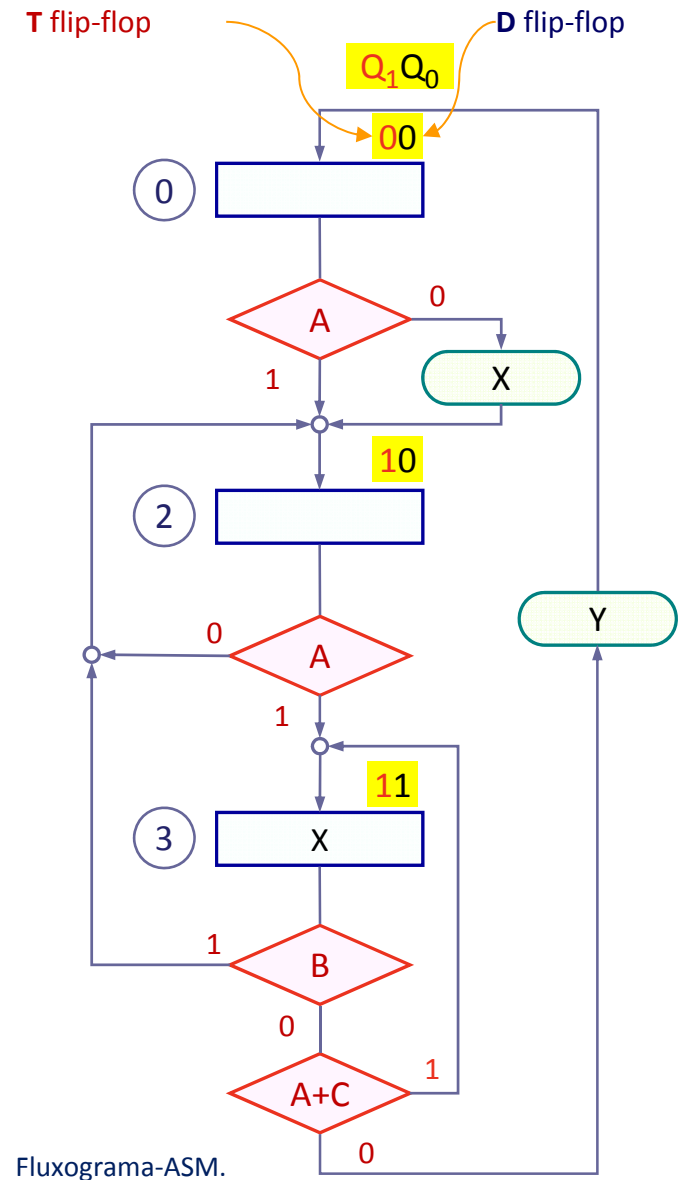


Diagrama de Estados equivalente ao Fluxograma-ASM, sendo menos nítida a visualização das Saídas.

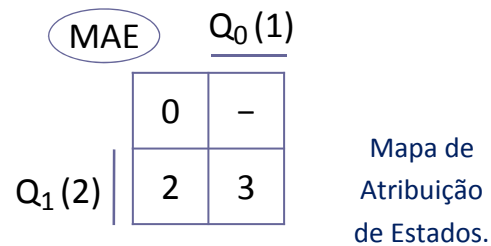


Fluxograma-ASM.

SÍNTESE COM FLIP-FLOP D E FLIP-FLOP T (EX. 9-4-2)

9-35

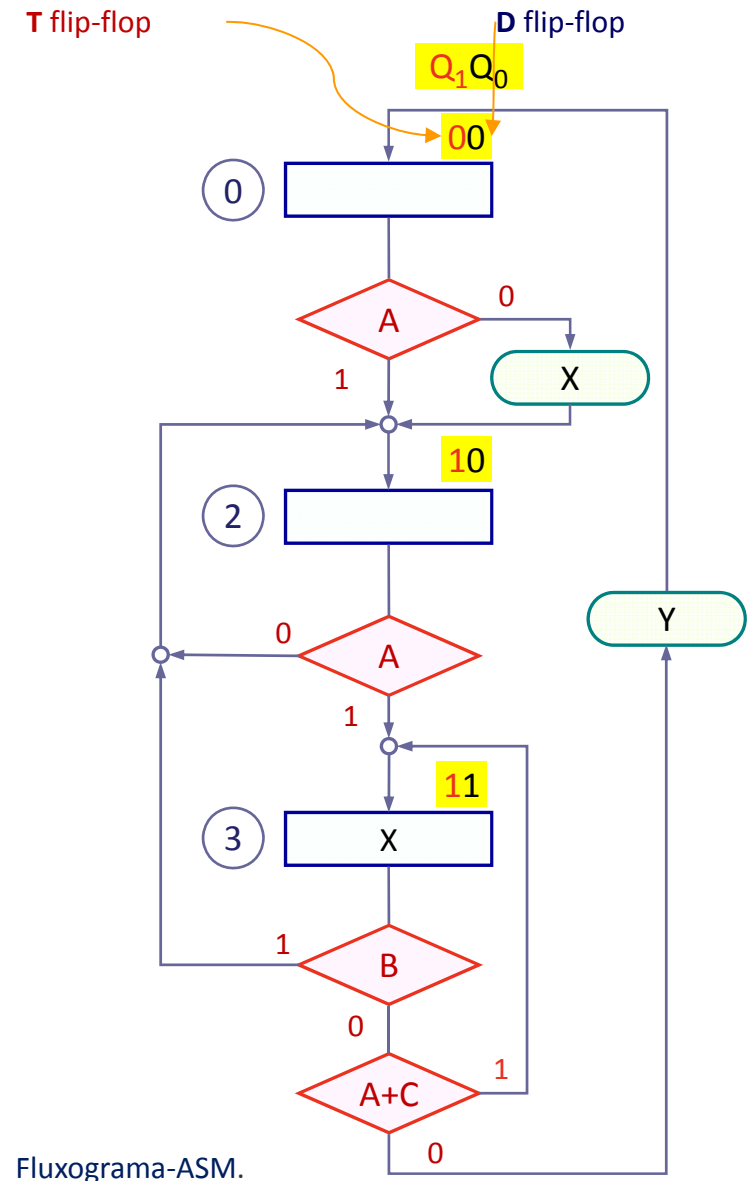
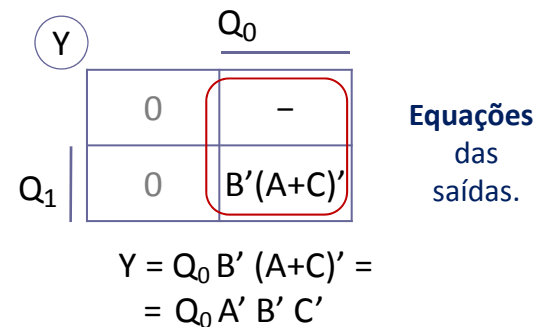
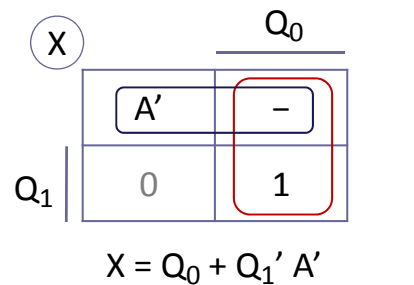
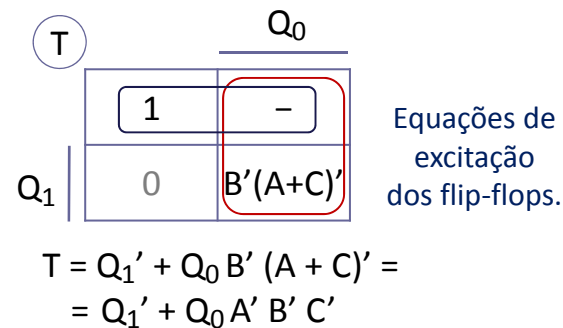
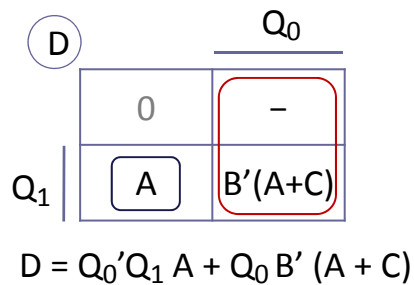
Exemplo 9-4



Q	Q ⁺	D	T
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

Tabela de excitação dos flip-flops D e T.

{ em 0: D=0; T = 1 }
 { em 2: D=A; T = 0 }
 { em 3: D=B'(A+C); T = B'(A+C)' }



Exemplo 9-4

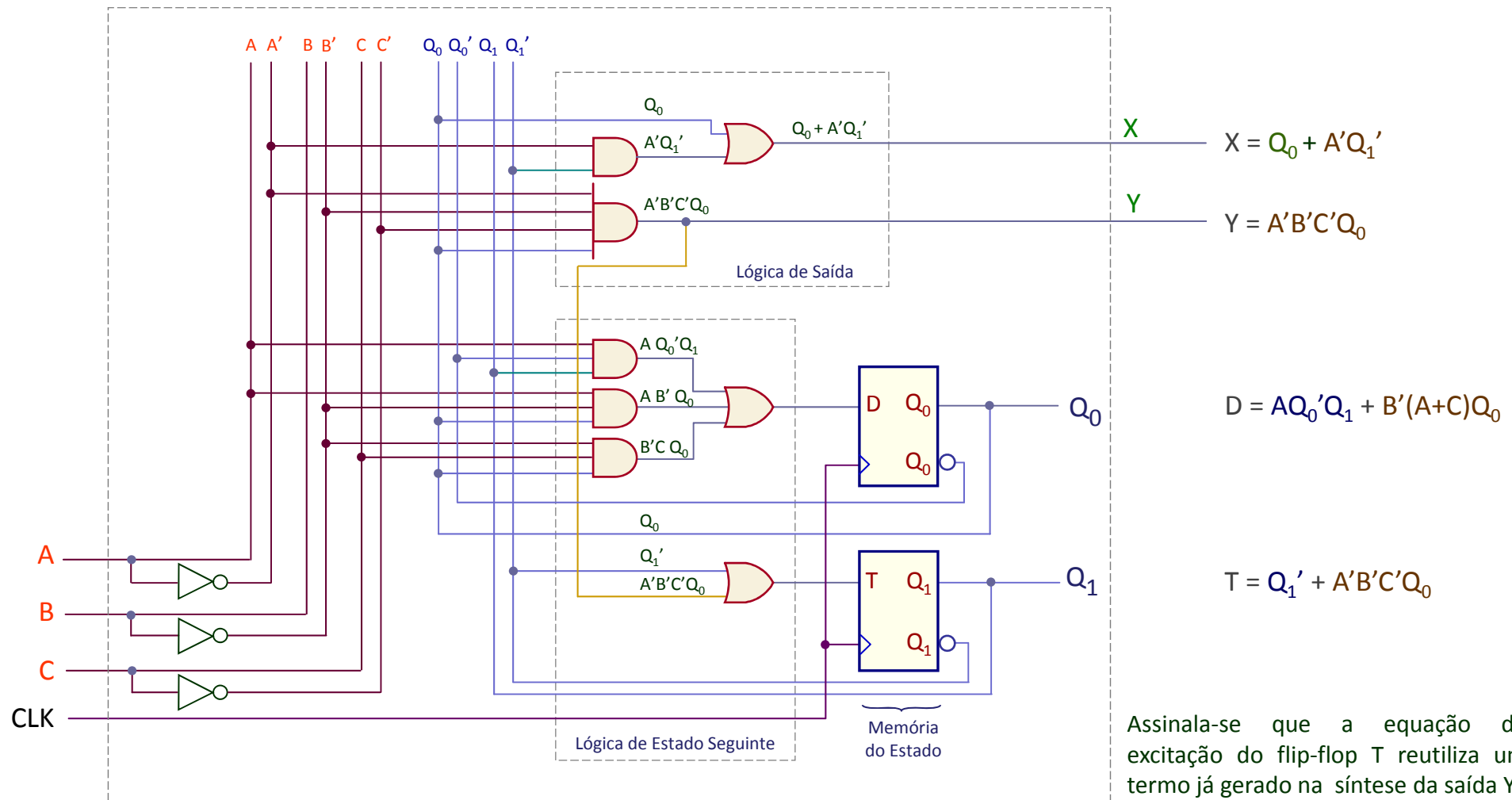
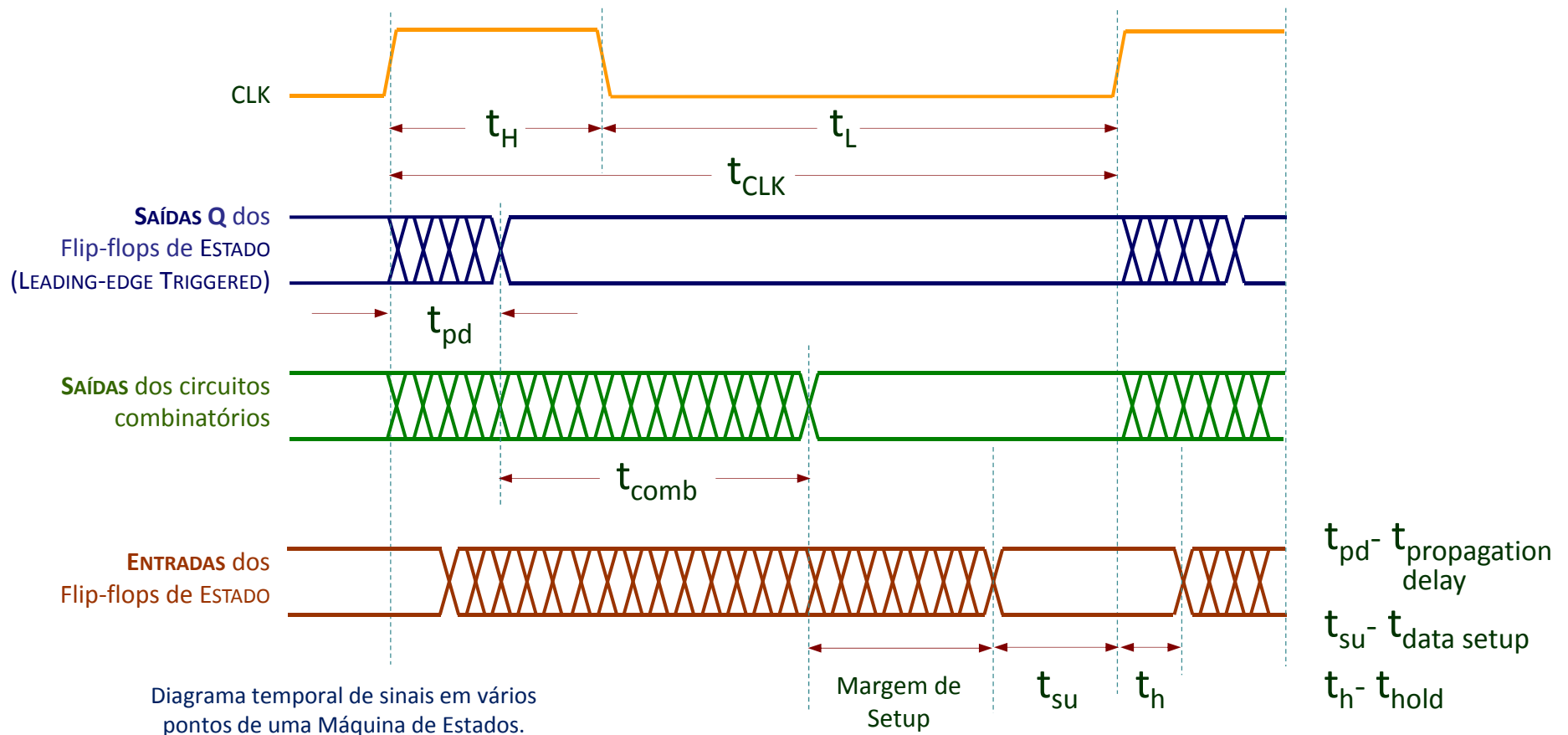


Diagrama lógico do circuito sintetizado.



EQUAÇÃO A OBSERVAR:

$$t_{CLK} \geq t_{pd} + t_{comb} + t_{ds}$$

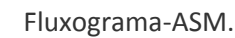
$$\text{MARGEM DE SETUP} = t_{CLK} - t_{pd} - t_{comb} - t_{su}$$

$$t_{pd} + t_{comb} \geq t_{hold}$$

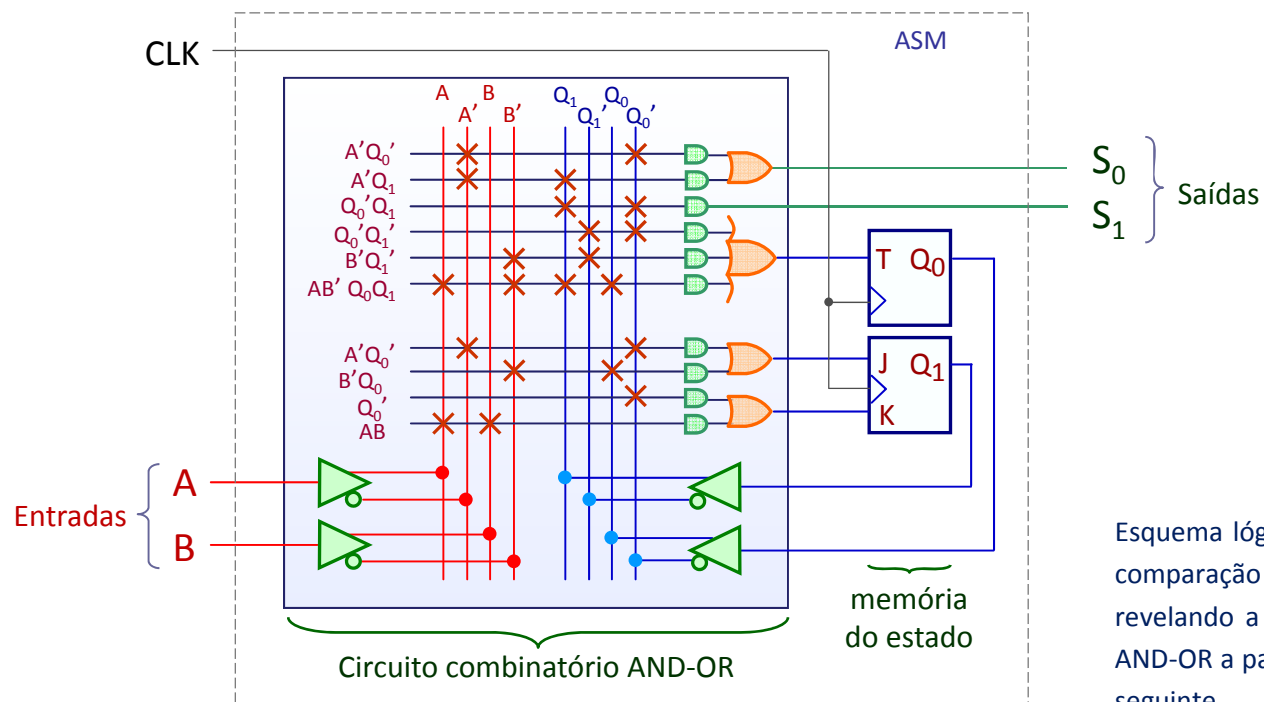
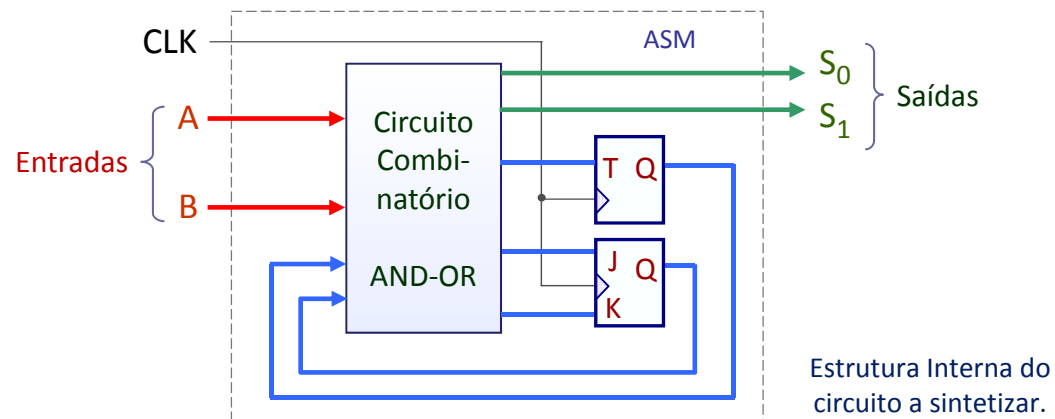
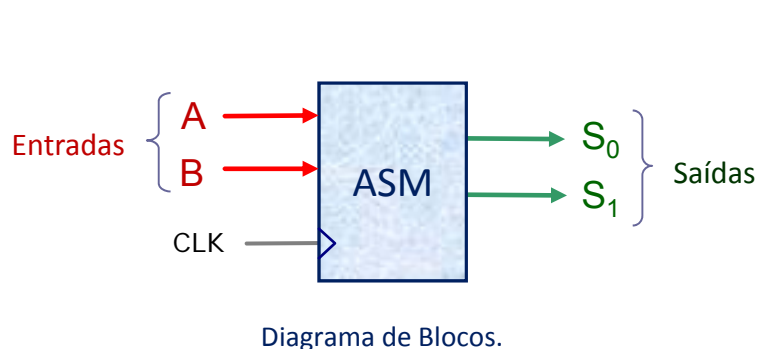
A margem de setup pode ser forçada a um valor positivo mantendo t_{CLK} a um valor suficientemente elevado – um relógio mais lento. t_{hold} é independente do relógio do sistema.

OBJETIVO

- Calcular as expressões necessárias e desenhar o diagrama lógico do circuito.
- Descrever a máquina em código CUPL recorrendo à estrutura SEQUENCE, e explicitar os pinos necessários, admitindo que se utiliza uma PAL 22V10.



Exemplo 9-5



SÍNTESE COM FLIP-FLOP J-K E FLIP-FLOP T (EX. 9-5-3)

9-40

Exemplo 9-5

MAE	Q ₀	
	0	1
Q ₁	2	3

Mapa de Atribuição de Estados.

S ₀	Q ₀	
	A'	0
Q ₁	A'	A'

$$S_0 = A'(Q_0' + Q_1)$$

$$S_1 = Q_0' Q_1$$

Lógica das funções geradoras das Saídas.

J	Q ₀	
	A'	B'
Q ₁	-	-

$$J = A'Q_0' + B'Q_0$$

K	Q ₀	
	-	-
Q ₁	1	AB

$$K = Q_0' + AB$$

T	Q ₀	
	1	B'
Q ₁	0	AB'

$$T = Q_0'Q_1' + B'Q_1' + AB'Q_0Q_1$$

Lógica das funções geradoras do Estado Seguinte.

Q	Q ⁺	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

(em 0: J = A' e K = -)

(em 1: J = B' e K = -)

(em 2: J = - e K = 1)

(em 3: J = - e K = AB)

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

(em 0: T = 1)

(em 1: T = B')

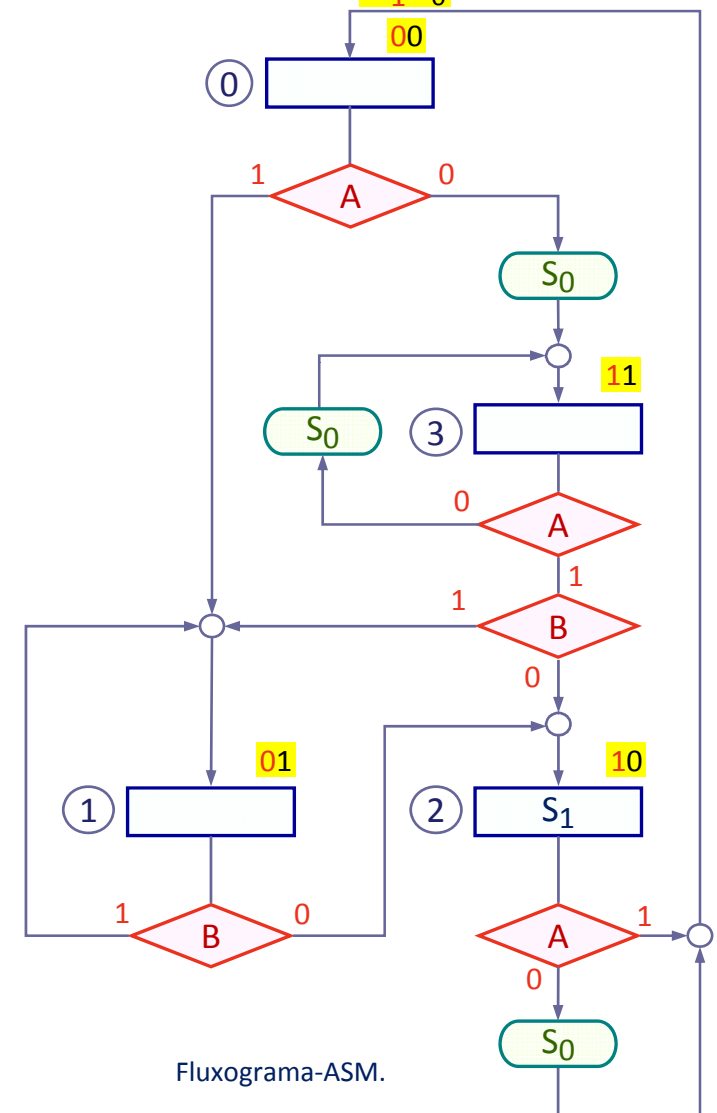
(em 2: T = 0)

(em 3: T = AB')

Tabelas de Excitação dos flip-flops JK e T.

- S₀ é uma saída condicionada – depende do Estado (pode ocorrer nos estados 0, 2 e 3) mas também da entrada A.
- S₁ é uma saída incondicional - só depende do Estado (ocorre sempre no Estado 2).

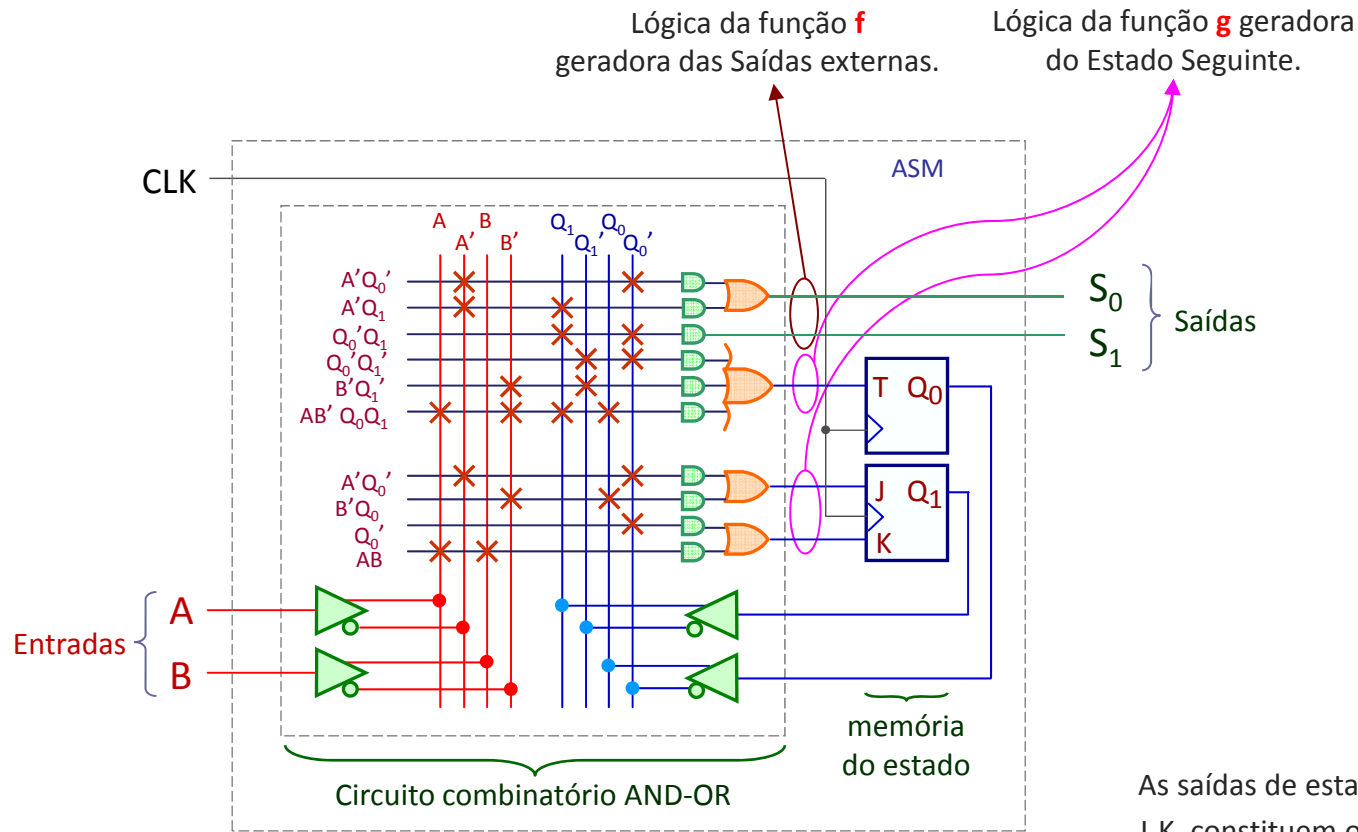
JK flip-flop → Q₁Q₀ ← T flip-flop



Fluxograma-ASM.



Exemplo 9-5



Circuito lógico em **topologia PAL** da máquina de estados implementada com flip-flops JK e T.

$$S_0 = A'(Q_0' + Q_1)$$

$$S_1 = Q_0' Q_1$$

$$J = A'Q_0' + B'Q_0 \quad K = Q_0' + AB$$

$$T = Q_0'Q_1' + B'Q_1' + AB'Q_0Q_1$$

As saídas de estado Q0 do flip-flop T, e Q1 e do flip-flop J-K, constituem o estado do circuito mas não são saídas externas da máquina ASM (essas são S0 e S1). Para a implementação na PAL 22V10 é no entanto necessário declará-las como saídas do circuito. Na PAL22V10 os flip-flops T e J-K serão substituídos pelos flip-flops D internos (esta PAL não possui flip-flops J-K).

Exemplo 9-5

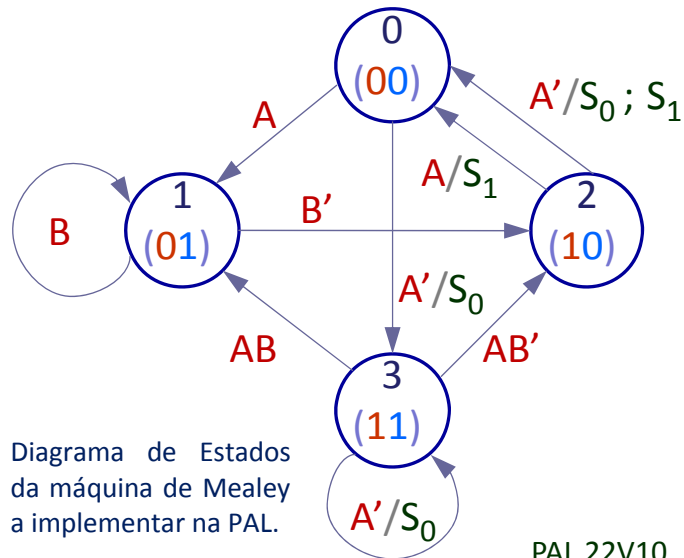
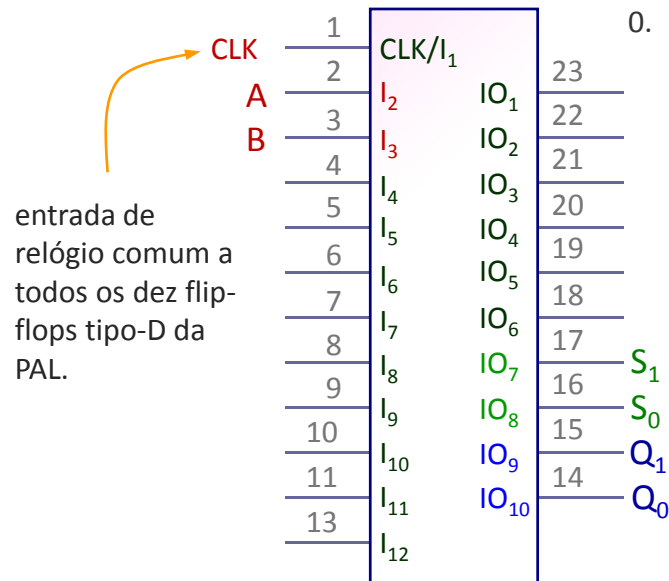


Diagrama de Estados da máquina de Mealey a implementar na PAL.

PAL 22V10



Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

A transição do estado 2 para o estado 0 é incondicional (não depende dos valores lógicos na entrada A).

No entanto a saída S_0 depende de A no estado 2 como assinalado, enquanto que a saída S_1 é incondicional do estado 2 e por isso aparece repetida nas setas de transição do estado 2 para o estado 0.

```

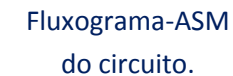
/* ***** INPUT PINS ***** */
PIN 1 = CLK      ; /* Clock do Circuito */
PIN 2 = A        ; /* Entrada A */
PIN 3 = B        ; /* Entrada B */
/* ***** OUTPUT PINS ***** */
PIN 14 = Q0      ; /* Saída Q0 do FF-T */
PIN 15 = Q1      ; /* Saída Q1 do FF-JK */
PIN 16 = S0      ; /* Saída S0 */
PIN 17 = S1      ; /* Saída S1 */
/* ***** BODY ***** */
[Q0..1].AR='b'0; /* Desactivação do Reset */
[Q0..1].SP='b'0; /* Desactivação do Preset */

SEQUENCE [Q1, Q0] {
  PRESENT 0
    IF !A OUT S0 ;
    IF !A NEXT 3;
    DEFAULT NEXT 1;
  PRESENT 1
    IF !B NEXT 2;
    DEFAULT NEXT 1;
  PRESENT 2
    OUT S1;
    IF !A OUT S0;
    NEXT 0;
  PRESENT 3
    IF !A OUT S0;
    IF !A NEXT 3;
    IF A&B NEXT 1;
    DEFAULT NEXT 2;
}
    
```

Troço do código CUPL da PAL.

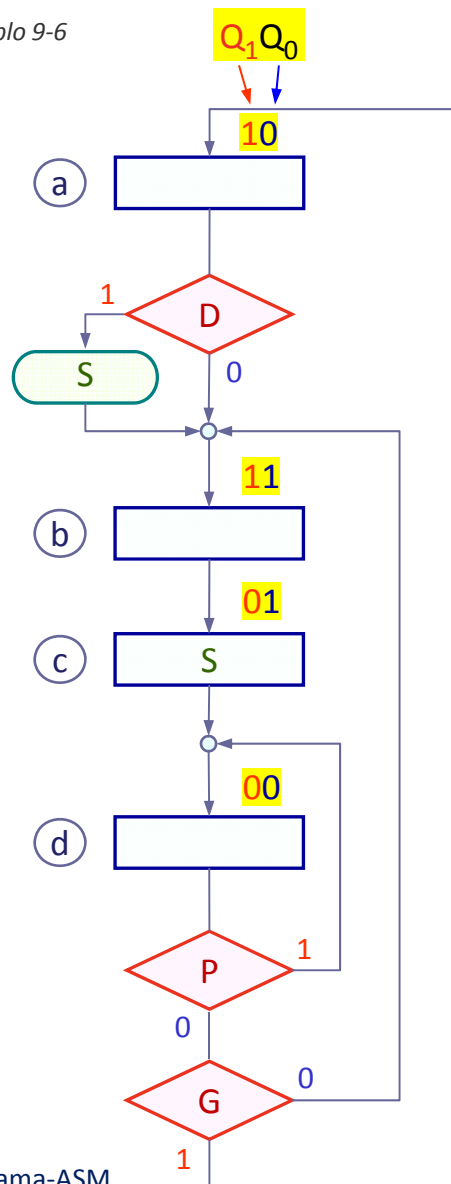
OBJETIVO

- a) Calcular as expressões necessárias das funções geradoras do Estado Seguinte e Saída, tendo por base flip-flops D.
- b) Realizar o circuito recorrendo a uma PAL22V10 e a flip-flops D externos. A PAL será apenas utilizada para a implementação da lógica combinatória. Apresentar o código CUPL e o diagrama lógico do circuito.
- c) Descrever a máquina de estados em CUPL, recorrendo à estrutura SEQUENCE, e explicitar os pinos necessários, admitindo que esta implementação recorre aos flip-flops existentes na PAL22V10.

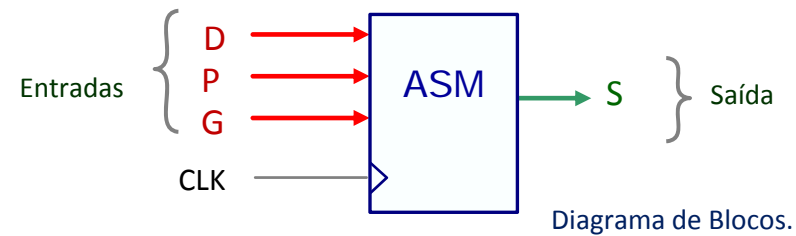


Exemplo 9-6

a)



Fluxograma-ASM do circuito.



MAE

	Q_0	
	d	c
Q_1	a	b

Mapa de Atribuição de Estados.

S

	Q_0	
	0	1
Q_1	D	0

$$S = Q_0 Q_1' + Q_0' Q_1 D$$

Lógica da função geradora da Saída.

D₀

	Q_0	
	P'G'	0
Q_1	1	1

$$D_0 = P'G' Q_0' + Q_1$$

D₁

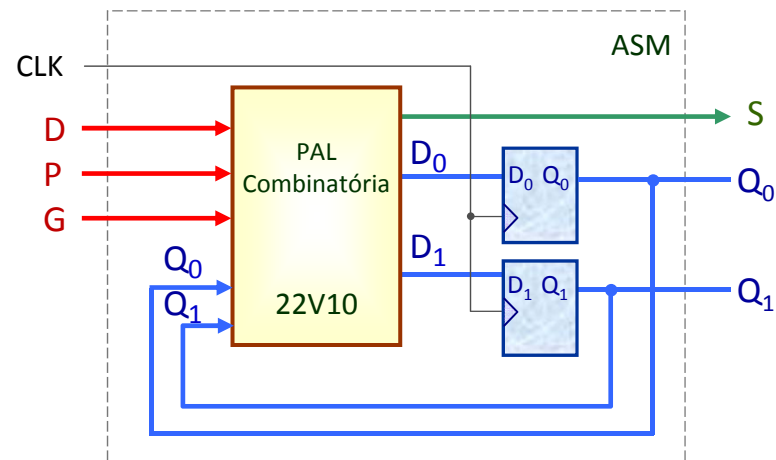
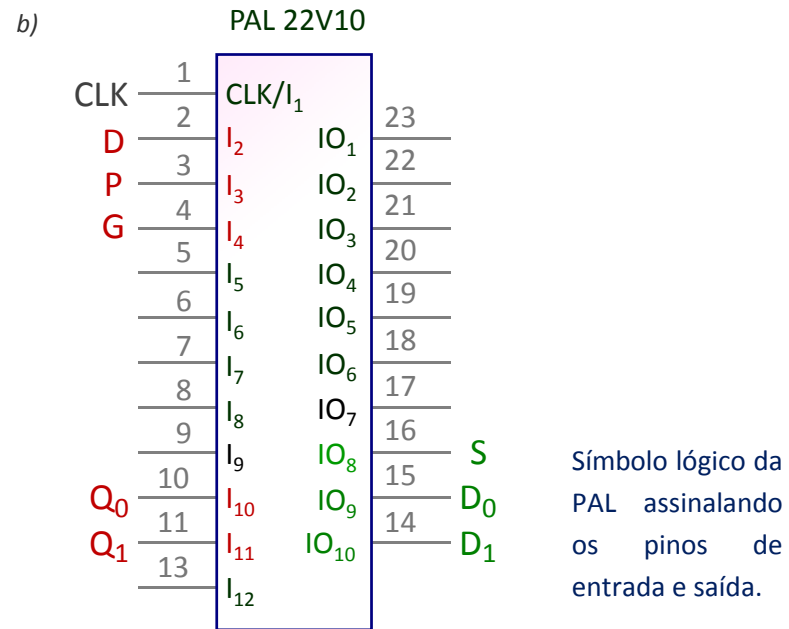
	Q_0	
	P'	0
Q_1	1	0

$$D_1 = Q_0' Q_1 + Q_0' P' = Q_0' (Q_1 + P')$$

Lógica das funções geradoras do Estado Seguinte.



Exemplo 9-6



```

/* ***** INPUT PINS ***** */
PIN 2 = D           ; /* Entrada D      */
PIN 3 = P           ; /* Entrada P      */
PIN 4 = G           ; /* Entrada G      */
PIN [10..11] = [Q0..1] ; /* Saídas dos FFs */

/* ***** OUTPUT PINS ***** */
PIN 16 = S          ; /* Saída S        */
PIN [15..14] = [D0..1] ; /* Entradas dos FFs */

/* ***** BODY ***** */
[Q0..1].AR='b'0;
[Q0..1].SP='b'0;

/* Expressões de D0 e D1 */
D0 = !P & !G & !Q0 # Q1;
D1 = !Q0 & Q1 # !Q0 & !P;

/* Expressão da Saída S */
S = Q0 & !Q1 # !Q0 & Q1 & D;
    
```

Troço do código CUPL da PAL combinatória.

Diagrama lógico do circuito utilizando a PAL 22V10 apenas na parte combinatória – os flip-flops D utilizados são externos à PAL.

Exemplo 9-6

c)

Símbolo lógico da PAL assinalando os pinos de entrada e saída.

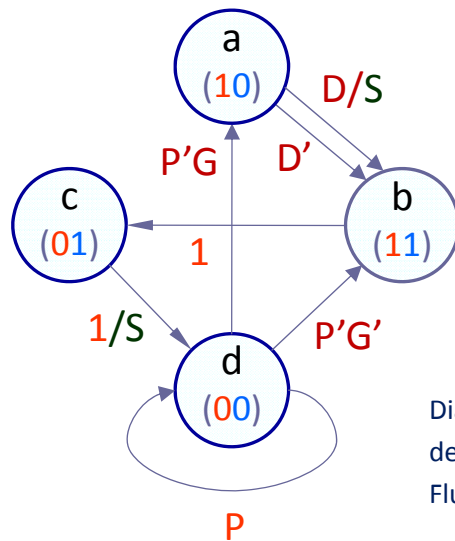
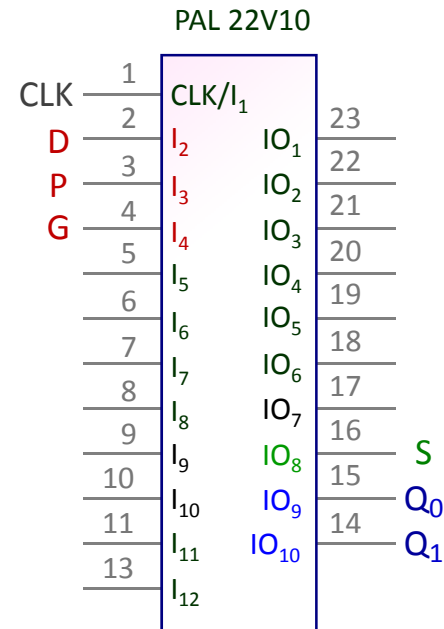


Diagrama de Estados da máquina de Mealey equivalente ao Fluxograma ASM.

```

/* ***** INPUT PINS ***** */
PIN 2 = D ;
PIN 3 = P ;
PIN 4 = G ;

/* ***** OUTPUT PINS ***** */
PIN 16 = S ;
PIN [15..14] = [Q0..1] ;

/* ***** BODY ***** */
[Q0..1].SP = 'b'0;
[Q0..1].AR = 'b'0;

SEQUENCE [Q1,Q0]{
    PRESENT 0
        IF P NEXT 0;
        IF !P & !G NEXT 3;
        DEFAULT NEXT 2;
    PRESENT 1
        OUT S;
        NEXT 0;
    PRESENT 2
        IF D OUT S;
        NEXT 3;
    PRESENT 3
        NEXT 1;
}
    
```

Troço do Código CUPL da PAL sequencial.

OBJETIVO

- Calcular as expressões necessárias e desenhar o diagrama lógico correspondente, tendo por base Flip-flops D. Garantir que se o estado 3 ocorrer indevidamente o circuito converge para o ciclo normal entrando no estado 0 ao fim de um impulso de relógio.
- Descrever a máquina de estados em CUPL, recorrendo à ESTRUTURA SEQUENCE, e explicitar os pinos necessários, admitindo que é utilizada uma PAL 22v10.
- Comparar o conteúdo do ficheiro **.doc** gerado com as equações deduzidas em a).



SÍNTESE COM PORTAS LÓGICAS E FLIP-FLOPS D, E COM PAL SEQUENCIAL (Ex. 9-7-2)

9-48

Exemplo 9-7

a)

MAE	Q_0	
	0	1
Q_1	2	3

O facto de o circuito ser auto-corrector pode limitar a exploração das **indiferenças** (don't cares). É o que acontece neste caso: nos quadrados correspondentes ao Estado 3 terá de figurar o valor **0** nos vários mapas.

Mapa de Atribuição de Estados.

(X)

	Q_0	
	0	A
Q_1	1	0

$$X = Q_0'Q_1 + AQ_0Q_1'$$

(Y)

	Q_0	
	0	1
Q_1	0	0

$$Y = Q_0Q_1'$$

(Z)

	Q_0	
	AB'	1
Q_1	0	0

$$Z = Q_0Q_1' + AB'Q_1'$$

Lógica das funções geradoras das Saídas.

(D₀)

	Q_0	
	$A'+B'$	0
Q_1	A	0

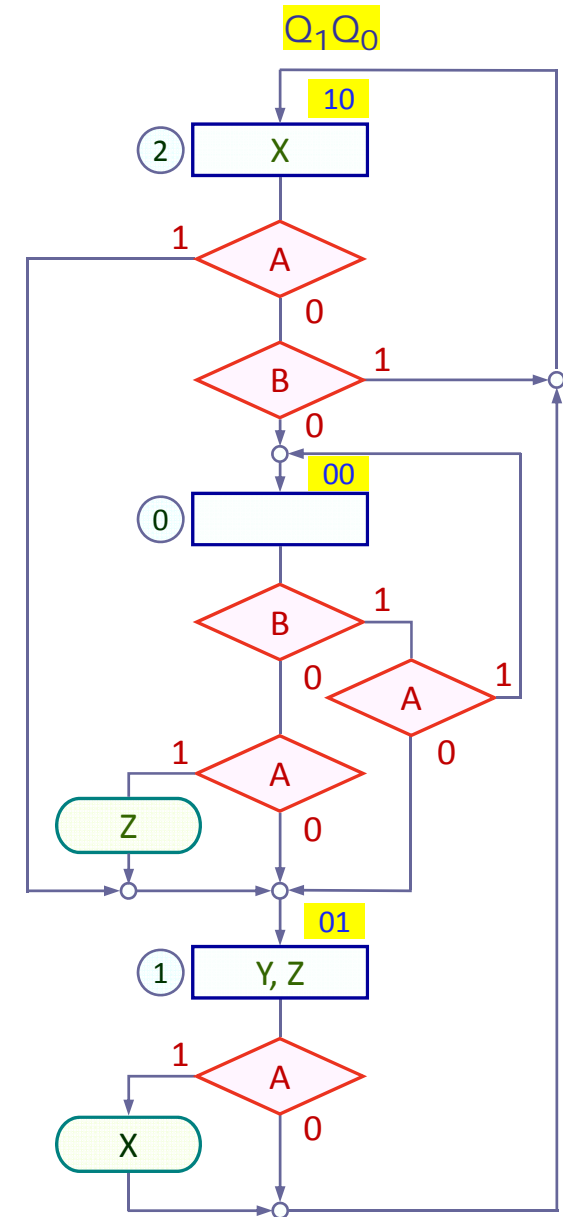
$$D_0 = Q_0'Q_1'(A'+B') + Q_0'Q_1A$$

(D₁)

	Q_0	
	0	1
Q_1	$A'B$	0

$$D_1 = Q_0Q_1' + Q_0'Q_1A'B$$

Lógica das funções geradoras do Estado Seguinte.



Fluxograma-ASM do circuito.



Exemplo 9-7

Estado Actual	Entradas		Estado Seguinte	Saídas
	B	A		
0	0	0	1	-
	0	1	1	Z
	1	0	1	-
	1	1	0	-
1	-	0	2	Y Z
	-	1	2	X Y Z
2	0	0	0	X
	0	1	1	X
	1	0	2	X
	1	1	1	X

Tabela de transição de estados e de saídas.

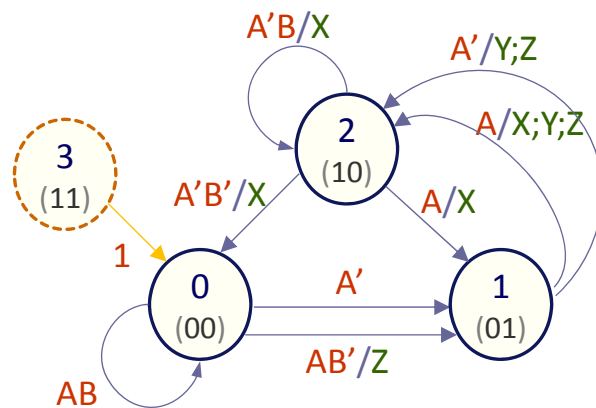


Diagrama de estados.

b)

```

Device p22v10;

/* ***** INPUT PINS *****/
PIN 2 = A ;
PIN 3 = B ;

/* ***** OUTPUT PINS *****/
PIN 18 = Z ;
PIN 19 = X ;
PIN 20 = Y ;
PIN 17 = Q1 ;
PIN 16 = Q0 ;

/* ***** BODY *****/

[Q1..0].AR = 'b'0;
[Q1..0].SP = 'b'0;

SEQUENCE [Q1, Q0] {
    PRESENT 2
        OUT X;
        IF A NEXT 1;
        IF !A & !B NEXT 0;
        DEFAULT NEXT 2;
    PRESENT 1
        OUT Y, Z;
        IF A OUT X;
        NEXT 2;
    PRESENT 0
        IF A & !B OUT Z;
        IF A & B NEXT 0;
        DEFAULT NEXT 1;
}
    
```

Troço do Código CUPL da PAL sequencial.

c)

Q0.d =>

A & !Q0 & Q1
 # !A & !Q0 & !Q1
 # !B & !Q0 & !Q1

Q0.ar => 0

Q0.sp => 0

Q1.d => Q0 & !Q1

!A & B & !Q0 & Q1

Q1.ar => 0

Q1.sp => 0

X => !Q0 & Q1

A & Q0 & !Q1

Y => Q0 & !Q1

Z => Q0 & !Q1

A & !B & !Q0 & !Q1

Extracto do ficheiro de extensão .doc da PAL com as expressões das funções geradoras das Saídas e dos Estados Seguintes, coincidentes com as do slide anterior excepto no caso da saída Z (a PAL não tirou partido da indiferença).

Exemplo 9-8

OBJECTIVO

Projectar um circuito para controlo de abertura e de fecho de uma porta. O movimento da porta é realizado por um motor controlado por dois sinais:

ON para ligar o motor,

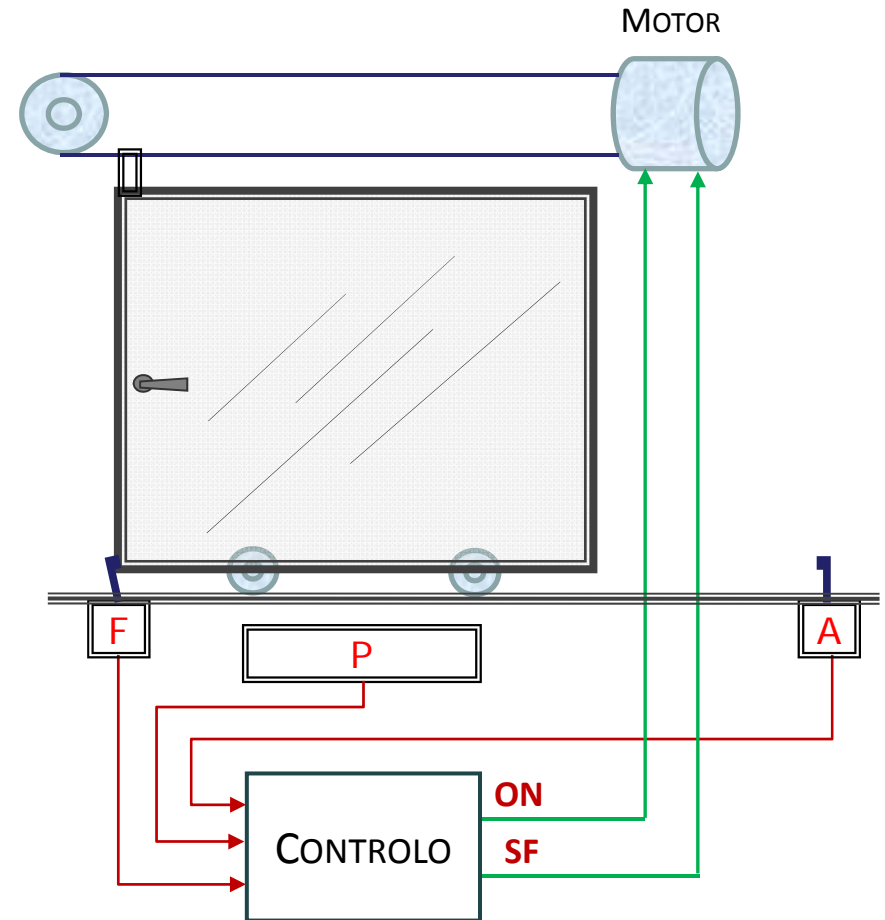
SF para determinar o sentido de rotação do motor sendo que o sentido de fecho é com $SF = 1$.

Existem dois sensores **A** e **F** que ficam activos quando a porta se encontra completamente aberta ($A=1$) ou completamente fechada ($F=1$).

Existe um sensor de presença **P**.

A abertura e o fecho devem obedecer ao seguinte critério:

- Quando a porta se encontra fechada, inicia a abertura se o sensor P ficar activo.
- Iniciada a abertura, esta só termina quando a porta se encontrar aberta.
- A porta depois de totalmente aberta inicia o fecho caso o sensor P esteja desactivo, de contrário permanece aberta.
- Iniciado o fecho, ele só termina quando a porta se encontrar totalmente fechada.
- Durante o fecho, se o sensor P ficar activo, o sistema reinicia a abertura de imediato.



Esboço da montagem e diagrama de blocos.

CONTROLO DE ABERTURA E FECHO DE PORTA (EX. 9-8-2)

9-51

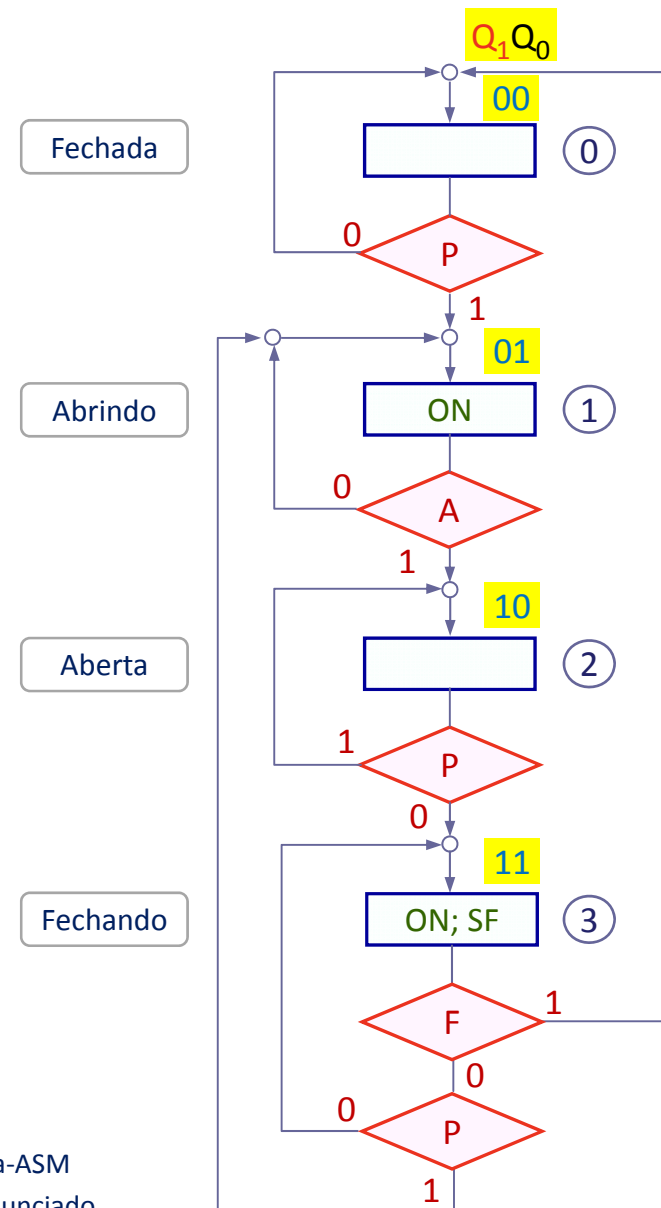
Exemplo 9-8

- Quando a porta se encontra fechada, inicia a abertura se o sensor **P** ficar activo.
- Iniciada a abertura, esta só termina quando a porta se encontrar completamente aberta (**A=1**).
- A porta depois de totalmente aberta inicia o fecho caso o sensor **P** esteja desactivo, de contrário permanece aberta.
- Iniciado o fecho, ele só termina quando a porta se encontrar totalmente fechada (**F=1**).
- Durante o fecho, se o sensor **P** ficar activo, o sistema reinicia a abertura de imediato.

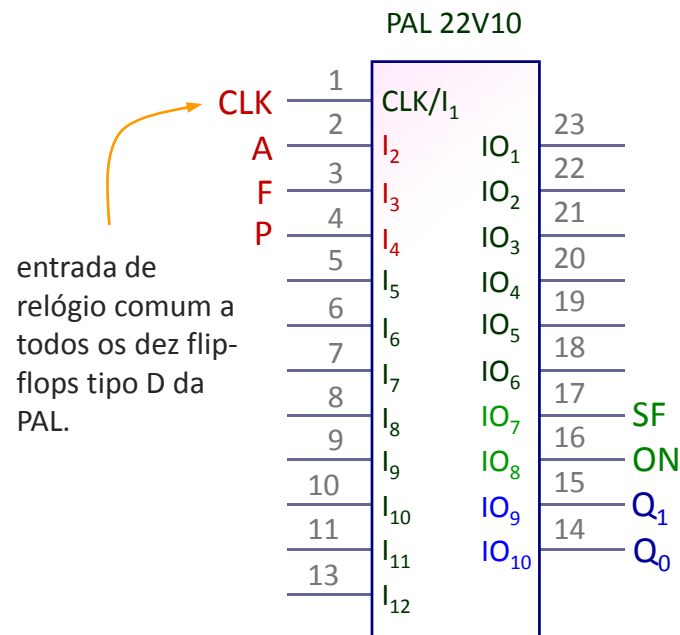
A codificação dos estados é, em princípio, arbitrária, ao contrário do que sucedia na síntese dos contadores síncronos, em que a codificação dos estados era imposta pela sequência de contagem pretendida. Por razões de natureza prática podem impor-se determinadas codificações.

Uma vez desenhado o Fluxograma-ASM e feita a codificação de estados, a parte criativa do projecto do circuito fica em grande parte concluída. O procedimento de síntese prossegue com a implementação em lógica discreta como se verificou nos slides anteriores, ou em PAL como descrito no slide seguinte.

Fluxograma-ASM
inferido do enunciado.



Exemplo 9-8



Símbolo lógico da PAL assinalando os pinos de entrada e saída.

```
Device p22v10;
....

/* ***** INPUT PINS *****/
PIN 2 = A ;
PIN 3 = F ;
PIN 4 = P ;

/* ***** OUTPUT PINS *****/
PIN 16 = ON ;
PIN 17 = SF ;
PIN [14..15] = [Q0..1];

/* ***** BODY *****/

SEQUENCE [Q1, Q0] {
    PRESENT 0
        IF !P NEXT 0;
        DEFAULT NEXT 1;
    PRESENT 1
        OUT ON;
        IF !A NEXT 1;
        DEFAULT NEXT 2;
    PRESENT 2
        IF P NEXT 2;
        DEFAULT NEXT 3;
    PRESENT 3
        OUT ON, SF;
        IF F NEXT 0;
        IF !F & !P NEXT 3;
        DEFAULT NEXT 1;
}
```

Troço do Código CUPL da PAL sequencial.

Exemplo 9-9

Este circuito constitui uma variante do anterior.

OBJECTIVO

Projectar um circuito para controlo de abertura e de fecho de um portão. O movimento do portão é realizado por um motor controlado por dois sinais:

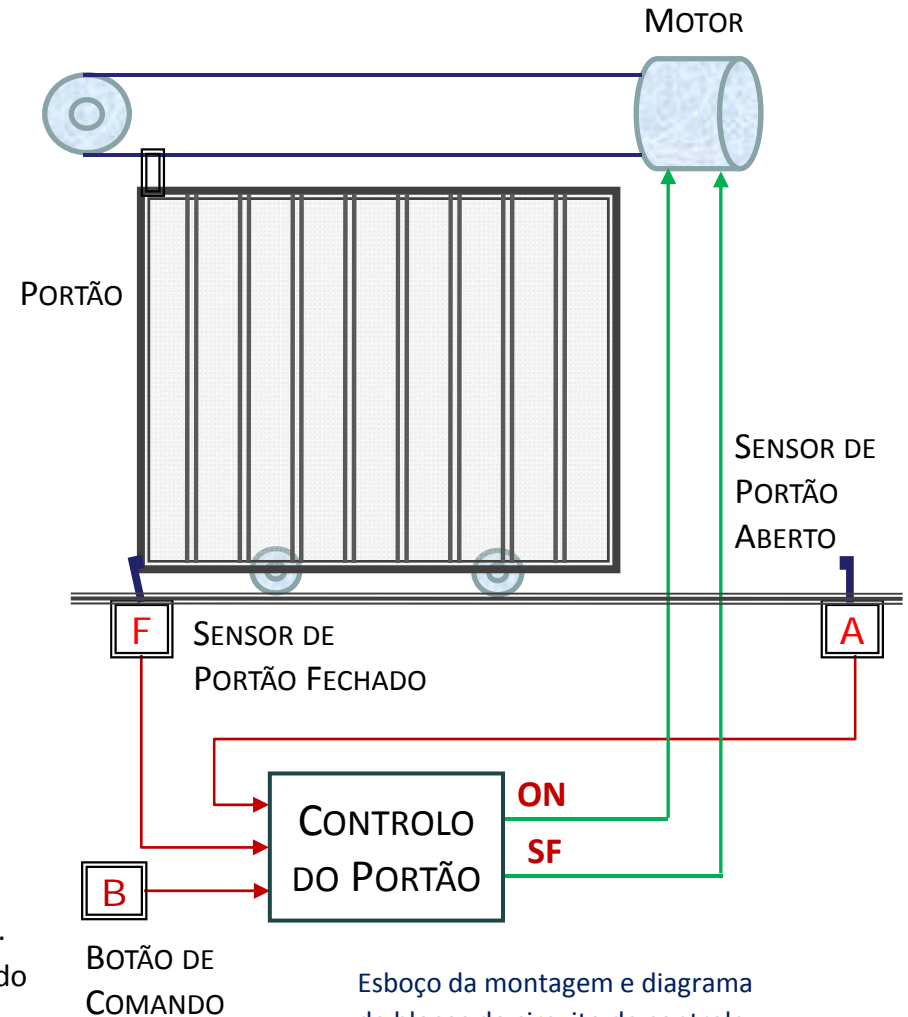
ON para ligar o motor,

SF para determinar o sentido de rotação do motor sendo que o sentido de fecho é com $SF = 1$.

Existem dois sensores **A** e **F** que ficam activos quando o portão se encontra completamente aberto ($A=1$), ou completamente fechado ($F=1$).

Existe um botão **B** que, quando premido, comanda a abertura e o fecho segundo o critério:

- Inicia a abertura se o portão se encontra fechado.
- Inicia o fecho se o portão se encontra aberto.
- Pára a abertura ou o fecho quando o portão se encontra em movimento.
- Inverte o sentido do movimento quando o portão se encontra parado numa posição intermédia.



CONTROLE DE ABERTURA E FECHO DE PORTÃO (Ex. 9-9-2)

9-54

Exemplo 9-9

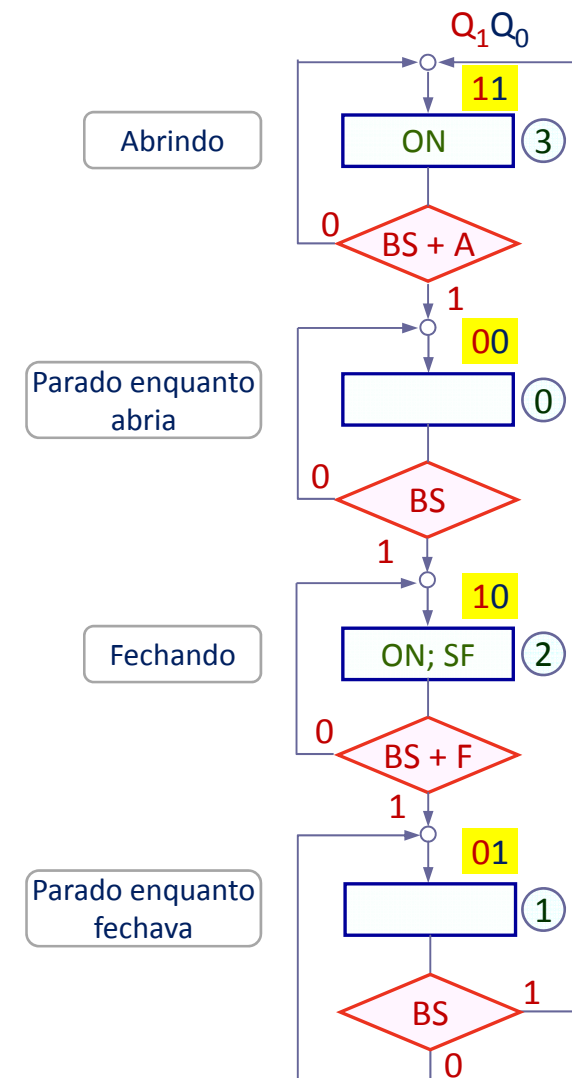
A PAL implementa dois circuitos ASM com o mesmo clock: um que corresponde à estrutura SEQUENCE principal ('Main'), e um outro dedicado à estrutura SEQUENCE de actuação do botão B que obriga a um ciclo completo da sua activação e desactivação, gerando uma saída BS síncrona com o clock.

É esta saída BS que leva o circuito ASM principal a evoluir de estado como mostrado no Fluxograma-ASM ao lado (a não desactivação do botão B depois da sua activação impediria o circuito de funcionar como pretendido).

Para a codificação utilizada é pressuposto que o portão se encontra aberto quando o circuito arranca a primeira vez, para que logo na primeira actuação do botão se inicie o fecho (de contrário seriam necessárias duas actuações seguidas do botão para colocar o portão em funcionamento).

```
SEQUENCE MAIN {
  PRESENT S0
  IF BS NEXT S2;
  DEFAULT NEXT S0;
  PRESENT S1
  IF BS NEXT S3;
  DEFAULT NEXT S1;
  PRESENT S2
  OUT ON, SF;
  IF (BS # F) NEXT S1;
  DEFAULT NEXT S2;
  PRESENT S3
  OUT ON
  IF (BS # A) NEXT S0;
  DEFAULT NEXT S3;
}
```

Troço do programa CUPL correspondente à estrutura Sequence do circuito de controle.



Fluxograma-ASM do circuito de controle.

CONTROLE DE ABERTURA E FECHO DE PORTÃO (Ex. 9-9-3)

9-55

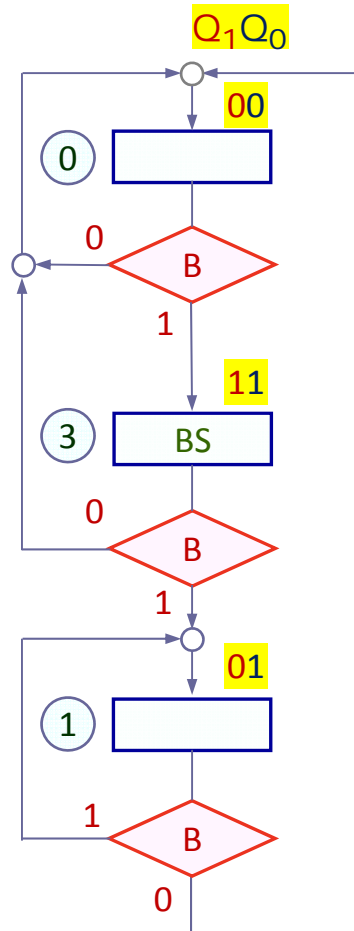
Exemplo 9-9

```
...
Device p22v10;
/***** INPUT PINS *****/
PIN 1 = CLK;
PIN 2 = B;

/***** OUTPUT PINS *****/
PIN 15 = Q0;
PIN 16 = Q1;
PIN 19 = BS;

/**** BOTÃO Edge-triggered ****/
Q0.AR = 'B'0;
Q0.SP = 'B'0;
Q1.AR = 'B'0;
Q1.SP = 'B'0;

SEQUENCE [Q1..Q0]{
    PRESENT 0
        IF B NEXT 3;
        DEFAULT NEXT 0;
    PRESENT 3
        OUT BS;
        IF B NEXT 1;
        DEFAULT NEXT 0;
    PRESENT 1
        IF B NEXT 1;
        DEFAULT NEXT 0;
}
```



Fluxograma-ASM de 3 estados para a geração da saída BS incondicional.

Troço do código CUPL.

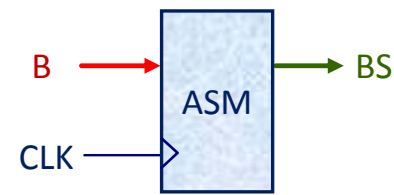
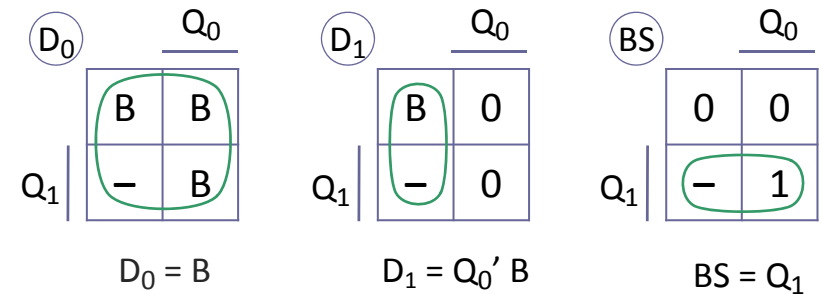


Diagrama de Blocos do circuito gerador da saída síncrona BS.

MAE

		Q ₀	
		0	1
Q ₁	0	-	3
	1	-	3

Mapa de Atribuição de Estados.



Lógica das funções geradoras do Próximo Estado e da Saída.

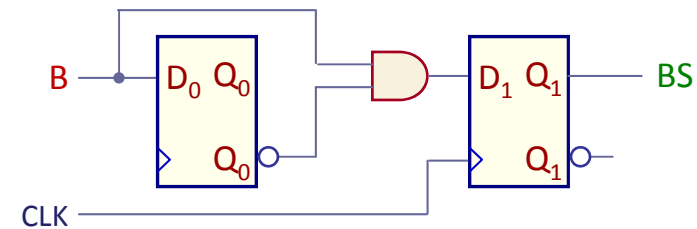


Diagrama lógico do circuito gerador da saída síncrona BS correspondente ao Fluxograma-ASM de 3 estados implementado com flip-flops tipo D.



CONTROLE DE ABERTURA E FECHO DE PORTÃO (Ex. 9-9-4)

9-56

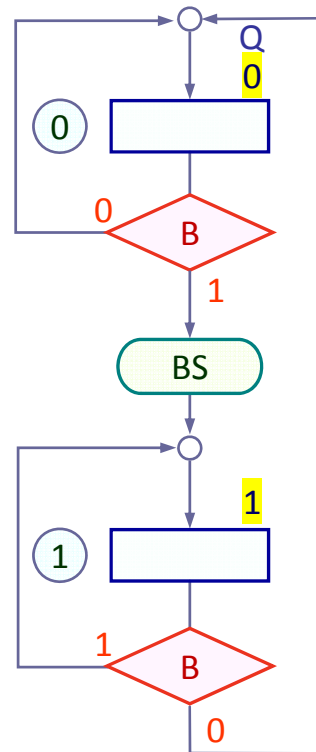
Exemplo 9-9

```
...
Device p22v10;
/***** INPUT PINS *****/
PIN 1 = CLK;
PIN 4 = B;

/***** OUTPUT PINS *****/
PIN 18 = BS;
PIN 19 = BT;

/* BOTÃO Edge-triggered */
BT.AR = 'B'0;
BT.SP = 'B'0;

SEQUENCE BT{
  PRESENT 0
    IF B OUT BS;
    IF B NEXT 1;
    DEFAULT NEXT 0;
  PRESENT 1
    IF B NEXT 1;
    DEFAULT NEXT 0;
}
```



Saída **BS** condicionada assíncrona (NON-REGISTERED OUTPUT): trecho do programa em CUPL e Fluxograma-ASM.

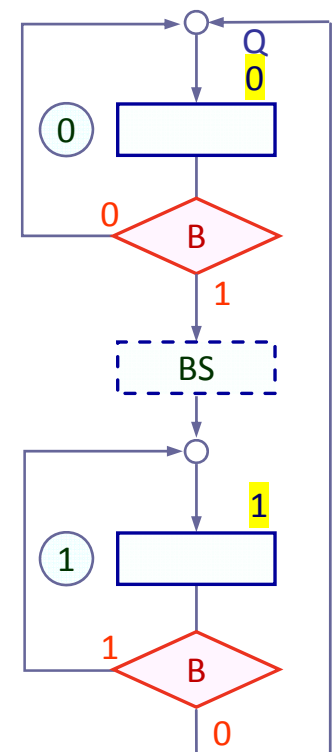
A definição de saídas é feita com a palavra OUT. O formato de descrição de SAÍDAS CONDICIONADAS **ASSÍNCRONAS** (NON-REGISTERED OUTPUT) é o da Fig. da esquerda, e o da descrição de SAÍDAS CONDICIONADAS **SÍNCRONAS** (REGISTERED OUTPUT) o da Fig. da direita.

```
...
Device p22v10;
/***** INPUT PINS *****/
PIN 1 = CLK;
PIN 4 = B;

/***** OUTPUT PINS *****/
PIN 18 = BS;
PIN 19 = BT;

/* BOTÃO Edge-triggered */
BT.AR = 'B'0;
BT.SP = 'B'0;

SEQUENCE BT{
  PRESENT 0
    IF B NEXT 1 OUT BS;
    DEFAULT NEXT 0;
  PRESENT 1
    IF B NEXT 1;
    DEFAULT NEXT 0;
}
```



Idem para uma saída **BS** condicionada síncrona (REGISTERED OUTPUT).

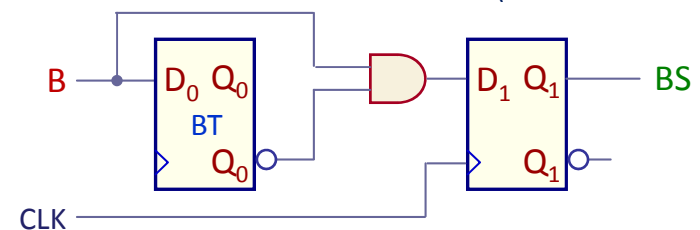


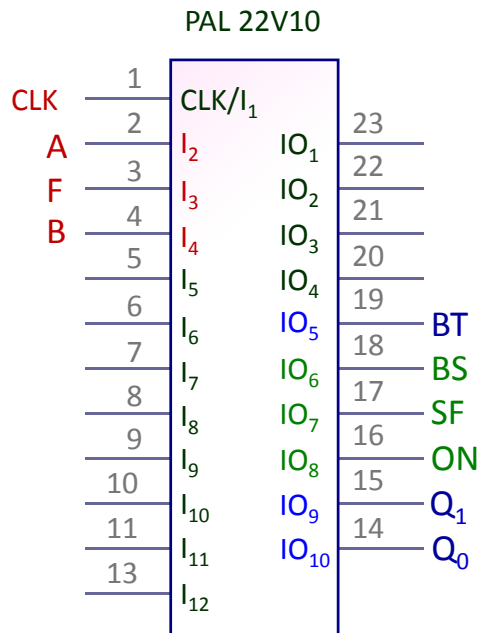
Diagrama lógico do circuito interno da PAL gerador da saída síncrona BS (REGISTERED OUTPUT) correspondente ao Fluxograma-ASM em cima.



CONTROLE DE ABERTURA E FECHO DE PORTÃO (Ex. 9-9-5)

9-57

Exemplo 9-9



Símbolo lógico da PAL assinalando os pinos de entrada e saída.

```
...
Device p22v10;

/***** INPUT PINS *****/

PIN 2 = A;
PIN 3 = F;
PIN 4 = B;

/***** OUTPUT PINS *****/

PIN 14 = Q0;
PIN 15 = Q1;
PIN 16 = ON;
PIN 17 = SF;
PIN 18 = BS;
PIN 19 = BT;

/***** Main *****/

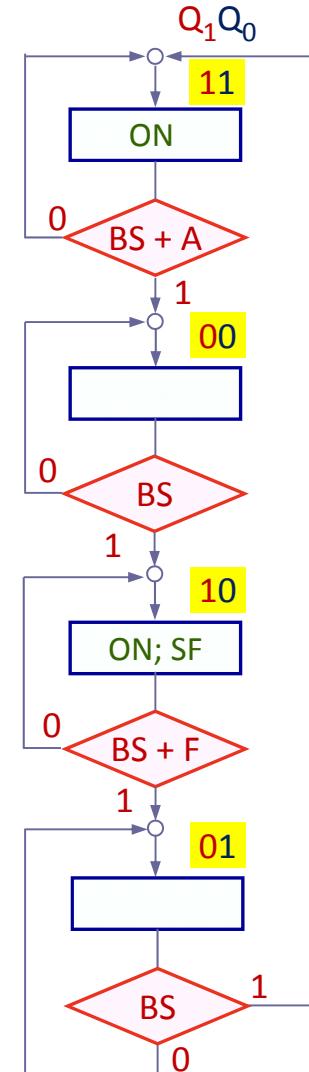
Q0.AR = 'b'0;
Q0.SP = 'b'0;
Q1.AR = 'b'0;
Q1.SP = 'b'0;
BT.AR = 'b'0;
BT.SP = 'b'0;
BS.AR = 'b'0;
BS.SP = 'b'0;

FIELD MAIN = [Q1..Q0];
$DEFINE S0 'b'00
$DEFINE S1 'b'01
$DEFINE S2 'b'10
$DEFINE S3 'b'11
```

```
SEQUENCE MAIN {
    PRESENT S0
        IF BS NEXT S2;
        DEFAULT NEXT S0;
    PRESENT S1
        IF BS NEXT S3;
        DEFAULT NEXT S1;
    PRESENT S2
        OUT ON, SF;
        IF (BS # F) NEXT S1;
        DEFAULT NEXT S2;
    PRESENT S3
        OUT ON
        IF (BS # A) NEXT S0;
        DEFAULT NEXT S3;
}

/***** Botão *****/

SEQUENCE BT {
    PRESENT 0
        IF B NEXT 1 OUT BS;
        DEFAULT NEXT 0;
    PRESENT 1
        IF B NEXT 1;
        DEFAULT NEXT 0;
}
```



Fluxograma-ASM do circuito de controle.

Programa CUPL do circuito de controle de abertura e fecho do portão.



Exemplo 9-10

OBJECTIVO

Desenhe o ASM-chart correspondente à máquina de estados descrita em CUPL na estrutura SEQUENCE ao lado.

TESTE			
x---	1	24	---x Vcc
V x---	2	23	---x
W x---	3	22	---x
T x---	4	21	---x
x---	5	20	---x
x---	6	19	---x
x---	7	18	---x Y
x---	8	17	---x U
x---	9	16	---x S
x---	10	15	---x Q1
x---	11	14	---x Q0
GND x---	12	13	---x

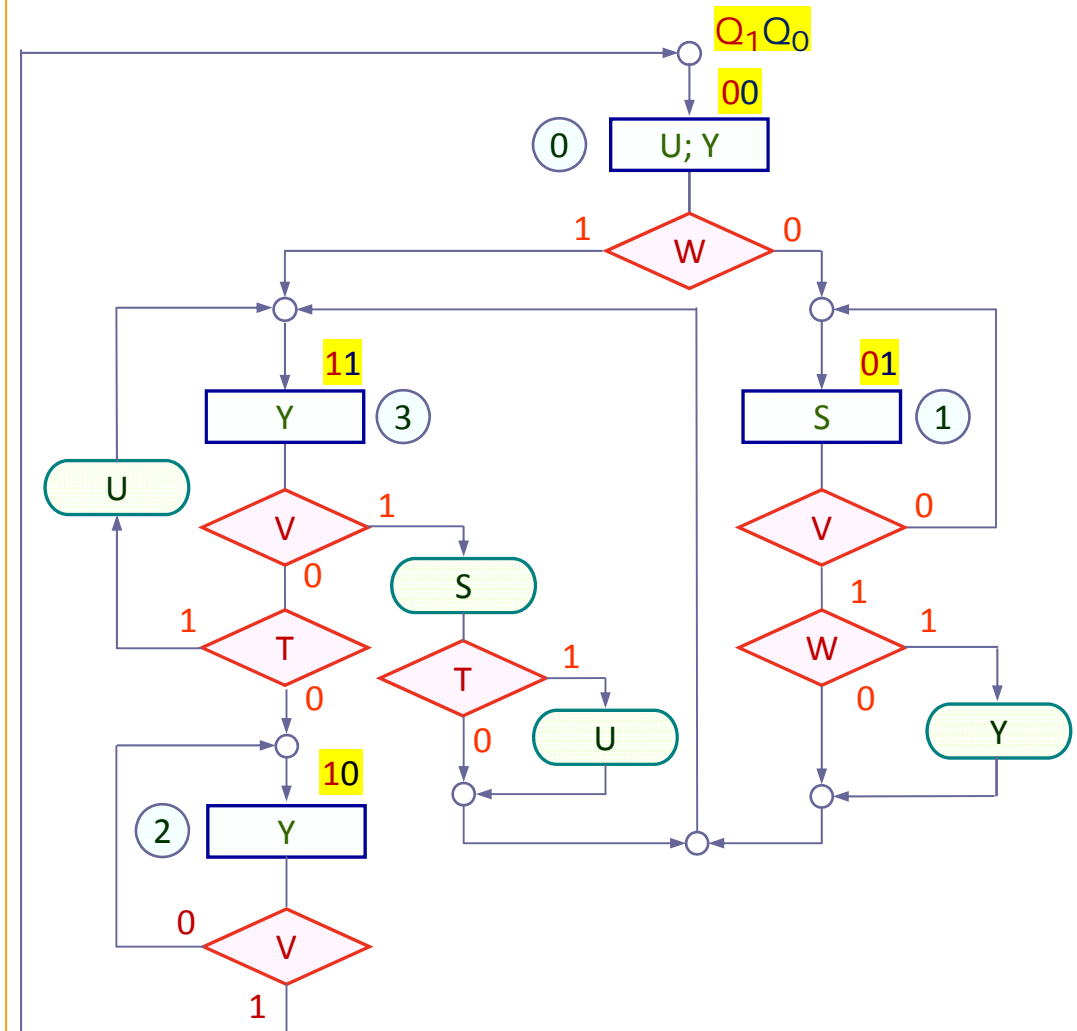
Símbolo lógico da PAL assinalando os pinos de entrada e saída.

Troço do código CUPL dado.

```

/* ***** INPUT PINS ***** */
PIN 2 = V ;
PIN 3 = W ;
PIN 4 = T ;
/* ***** OUTPUT PINS ***** */
PIN 14 = Q0 ;
PIN 15 = Q1 ;
PIN 16 = S ;
PIN 17 = U ;
PIN 18 = Y ;
/* ***** BODY ***** */
[Q0..1].SP = 'b' 0;
[Q0..1].AR = 'b' 0;

SEQUENCE [Q1,Q0] {
    PRESENT 0
        OUT Y, U ;
        IF !W NEXT 1 ;
        DEFAULT NEXT 3 ;
    PRESENT 1
        IF !(W#V) OUT Y ;
        IF !V NEXT 1 ;
        OUT S ;
        DEFAULT NEXT 3 ;
    PRESENT 3
        OUT Y ;
        IF !(T#V) NEXT 2 ;
        IF T OUT U ;
        IF V OUT S ;
        DEFAULT NEXT 3 ;
    PRESENT 2
        IF !V NEXT 2 ;
        OUT Y ;
        DEFAULT NEXT 0 ;
}
    
```



Fluxograma-ASM do circuito correspondente ao código ao lado.

OBJECTIVO

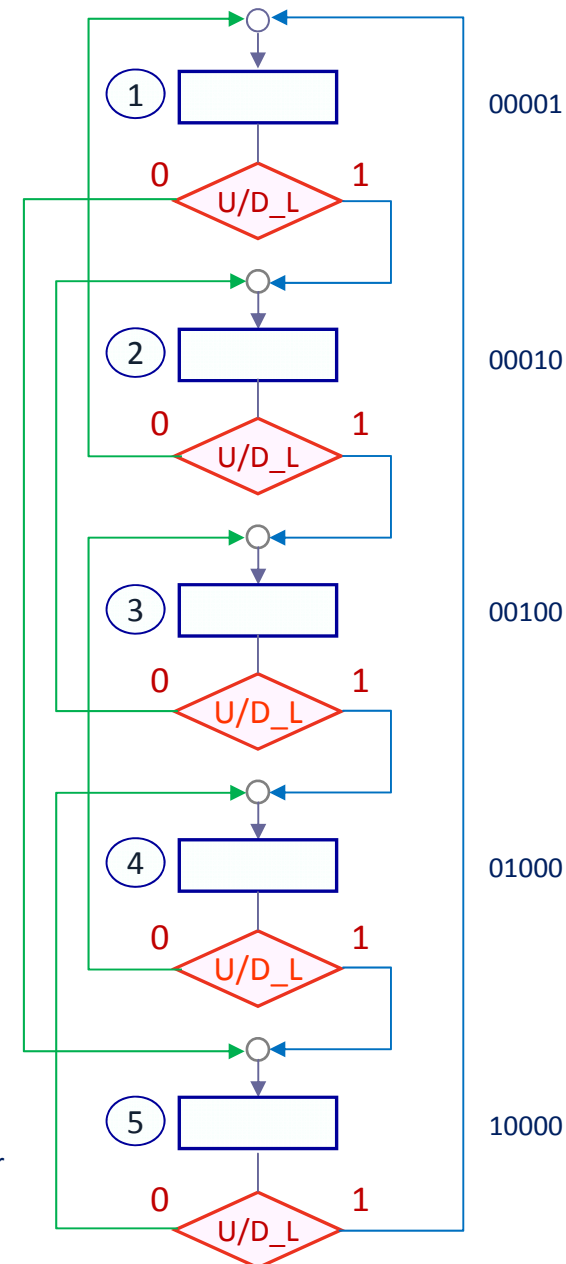
Projectar, usando o método **One-hot** de um flip-flop por estado, um contador **Up-Down Módulo-5** que, por acção de um comutador U/P_L executa as seguintes acções:

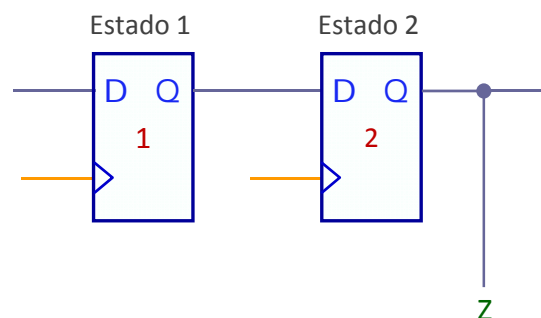
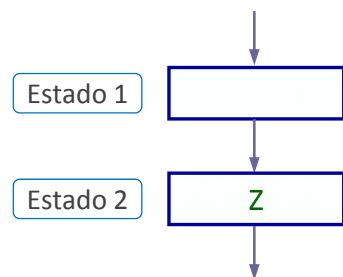
- $U/D_L = 1$ Contador crescente (UP)
- $U/D_L = 0$ Contador decrescente (DOWN)

Este método alternativo utiliza um número de flip-flops igual ao número de estados, e codifica os estados de modo a existir 1 e 1 só flip-flop activo por estado. É denominado **one-hot** porque, em cada momento, só um dos flip-flops tem a sua saída Q activada - esse flip-flop determina o estado em que o circuito se encontra.

A existência de um flip-flop por cada estado do circuito conduz a mais flip-flops do que os necessários com a metodologia anterior, mas acarreta uma simplificação da lógica da função geradora do **Estado Seguinte** a colocar à entrada de cada flip-flop, dispensando a metodologia das tabelas e mapas de Karnaugh se forem utilizados flip-flops tipo-D.

Fluxograma-ASM do contador
Up-Down Módulo-5.

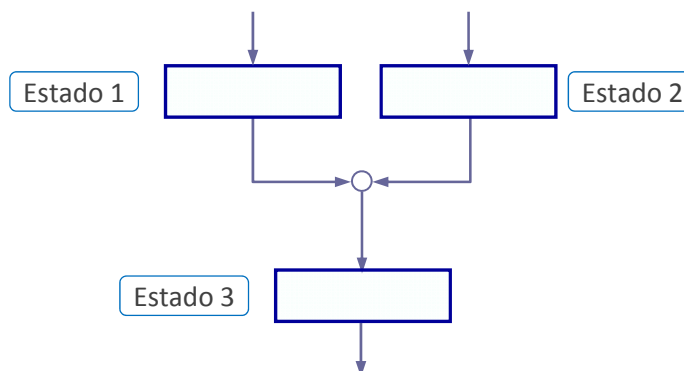




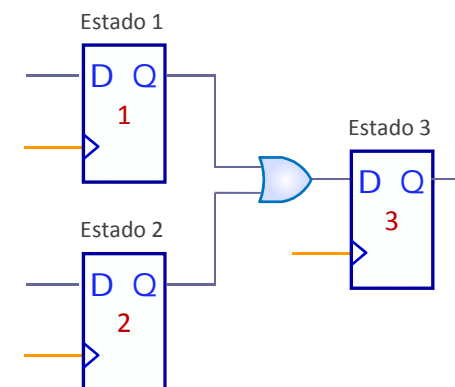
$$D_2 = Q_1$$

$$Z = Q_2$$

Transição incondicional entre estados mostrando-se as caixas de estados, a ligação em série entre os flip-flops correspondentes, e as equações aplicáveis.

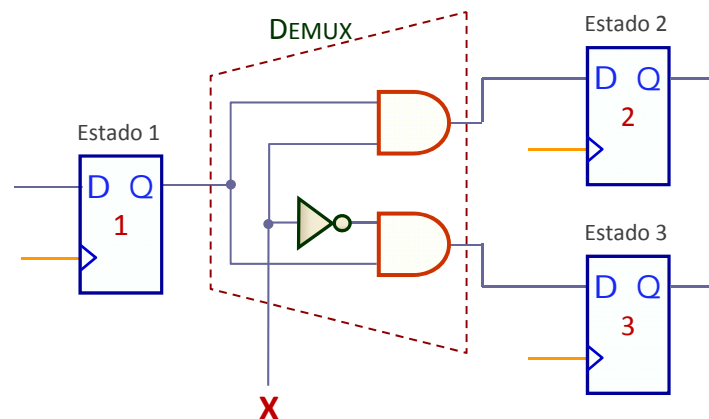
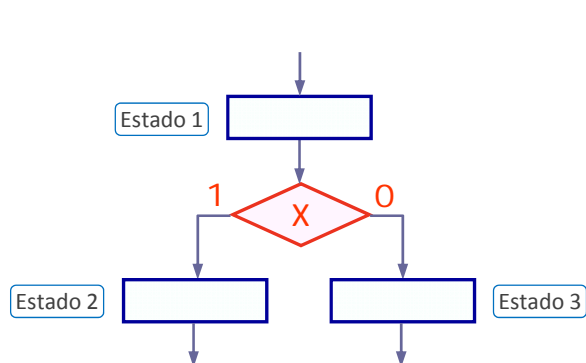


Convergência de ramos (Junção ou Join) entre estados mostrando as caixas de estados, a ligação OR entre os flip-flops correspondentes, e as equações aplicáveis.



Há uma correspondência biunívoca entre os vários elementos da notação ASM e o diagrama lógico do circuito como indicado neste slide e no seguinte:

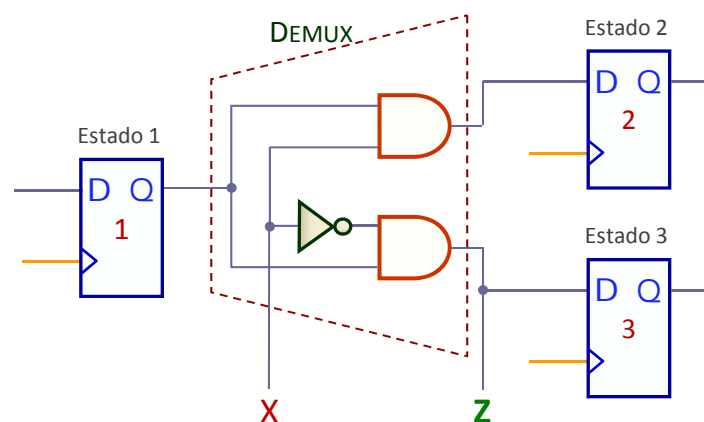
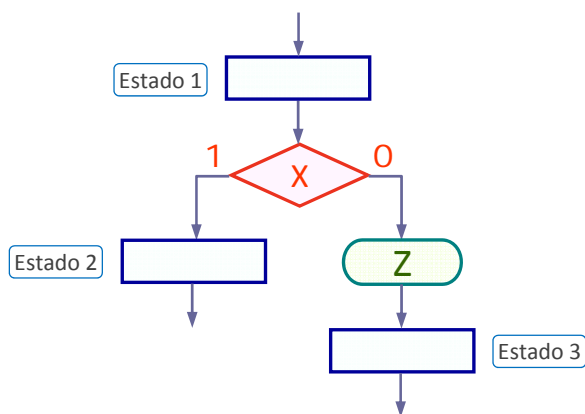
- Caixa de Estado → Flip-flop D
- Caixa de Decisão → Demultiplexers
- Junção → Porta OR
- Saída condicional → Porta AND.



$$D_2 = X Q_1$$

$$D_3 = X' Q_1$$

Transição condicionada (ou Fork) entre estados mostrando-se a caixa de decisão, o demultiplexer correspondente (realizado com portas lógicas), e as equações aplicáveis – o exemplo considera apenas uma variável X a condicionar a transição.



$$Z = X' Q_1$$

Saída condicionada realizada através do AND e equação aplicável – o exemplo considera apenas uma saída Z.

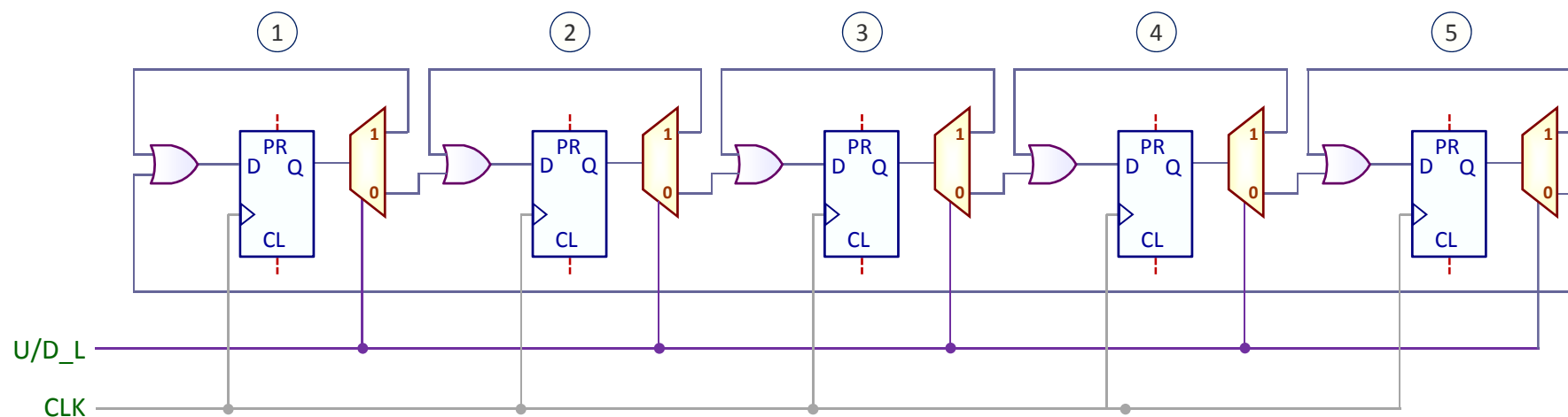


Diagrama lógico do contador Up-Down Módulo-5 sintetizado a partir do método One-hot.

A inicialização terá de ser feita com a activação do flip-flop correspondente ao estado inicial – colocado a 1 por via de um **PRESET ASSÍNCRONO** – e a desactivação de todos os outros – colocados a 0 por via de um **CLEAR ASSÍNCRONO**.

O acréscimo de flip-flops inerente à metodologia One-hot é compensado pela simplicidade de síntese e redução da lógica combinatória.

No sentido dos ponteiros de relógio:
contagem crescente correspondente a $U/D_L=1$;

No sentido contrário:
contagem decrescente correspondente a $U/D_L=0$.

Cada estado é directamente identificado pela existência de um '1' no flip-flop respectivo.

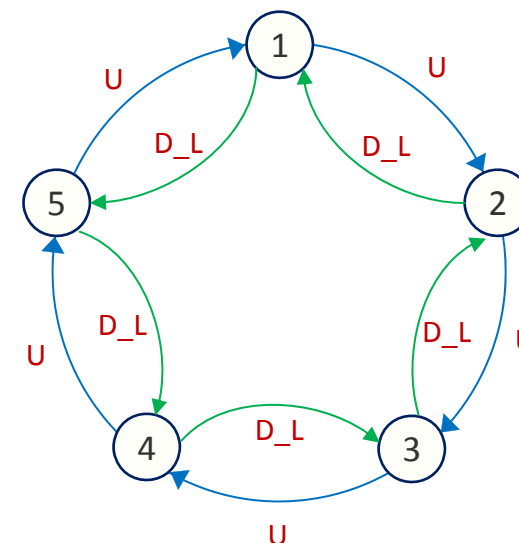


Diagrama de estados do contador Up-Down Módulo-5.

Exemplo 9-11

OBJECTIVO

Para a máquina sequencial síncrona da Fig. baseada num contador de 2 bits (CTR2) com Parallel Load (PL) **síncrono** e lógica auxiliar:

- Desenhar o Fluxograma-ASM correspondente.
- Obter as expressões das funções geradoras do Estado Seguinte e das Saídas.
- Desenhar uma implementação do circuito com flip-flop D e portas lógicas.

O processo de análise de um circuito sequencial síncrono parte do seu diagrama lógico (ou esquema eléctrico) e chega a um Fluxograma-ASM ou Diagrama de Estados que descreve o seu comportamento temporal.

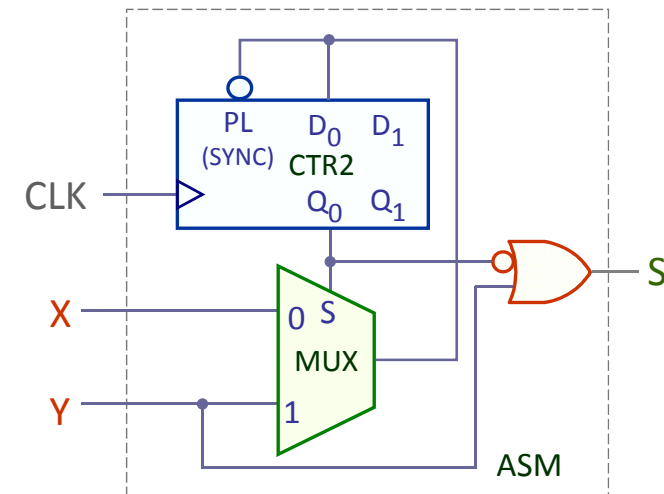
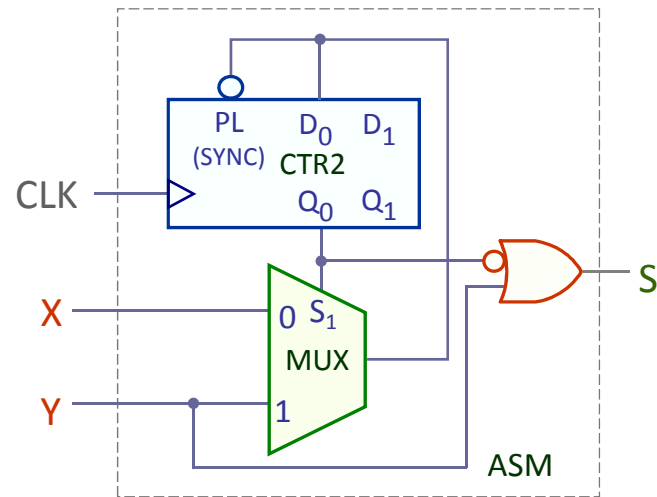
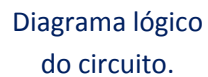


Diagrama lógico do circuito.

A tabela de **transição de estados e saídas** é um passo auxiliar para a análise do circuito (dispensável para os mais experientes). Identifica, para cada Estado Actual e combinação das Entradas Actuais, o Estado Seguinte e as Saídas Actuais do circuito, através da listagem de outras variáveis internas que auxiliam a percepção do comportamento do circuito.

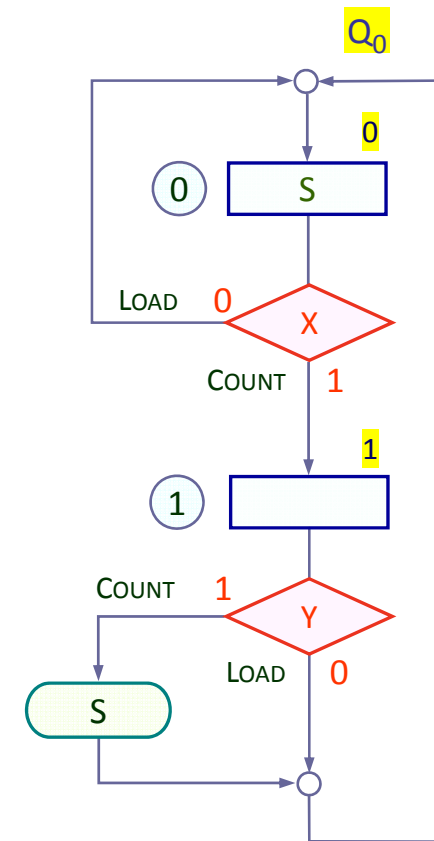
Um dos bits (Q_1) do contador deste ASM não é utilizado.

a)



Estado Actual	Entradas Externas	Variáveis intermédias			Estado Seguinte	Saída Actual
Q_0	X, Y	PL_L	D ₀	Modo do CTR2	Q_0^+	S
0	X=0	0	0	LOAD	0	1
	X=1	1	1	COUNT	1	
1	Y=0	0	0	LOAD	0	0
	Y=1	1	1	COUNT	0	1

Tabela de transição de estados e de saídas.



Fluxograma-ASM pretendido.

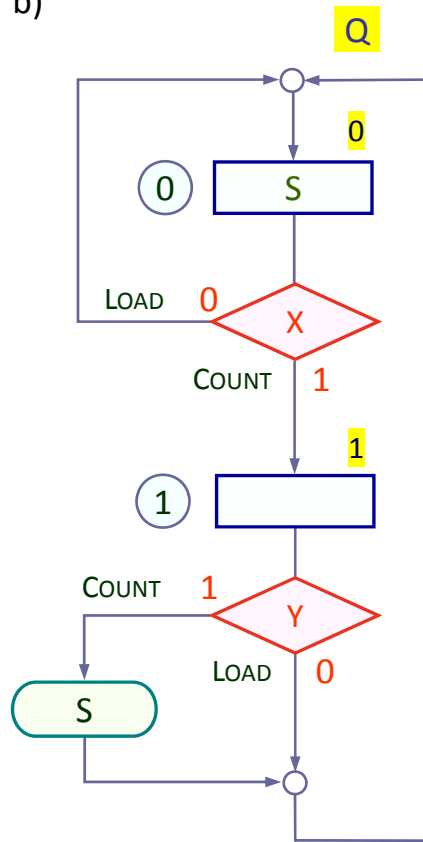
O Fluxograma-ASM é directamente inferível da tabela de transição de estados e saídas.

ANÁLISE DE UM CIRCUITO SEQUENCIAL SÍNCRONO (Ex. 9-11-4)

9-66

Exemplo 9-11

b)



Fluxograma-ASM.

MAE	Q
0	1

Mapa de Atribuição de Estados.

Q^+	Q
X	0

$$Q^+ = D = X Q'$$

Equação de transição e de excitação.

Como a equação característica para o flip-flop D é $Q^+ = D$, vem $D = X Q'$.

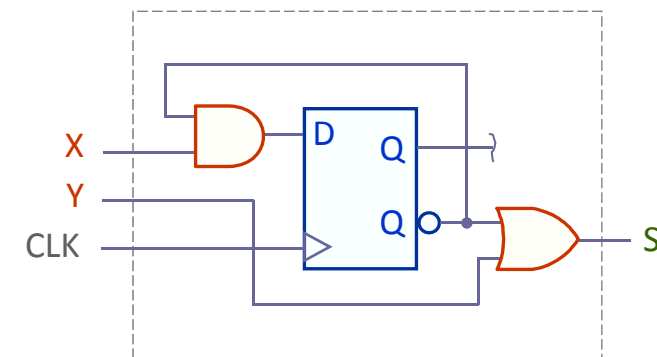
S	Q
1	Y

$$S = Q' + Y$$

Equação da saída S.

Os mapas de Karnaugh podem ser construídos directamente a partir da tabela de transição de estados e de saídas sem passar pelo Fluxograma-ASM.

c)



Implementação do circuito com flip-flop D e portas lógicas.



Exemplo 9-12

OBJECTIVO

Desenhar o Fluxograma-ASM correspondente à máquina sequencial síncrona da figura, baseada num contador de 2 bits (CTR2) com Parallel Load síncrono (PL Sync) e lógica auxiliar.

Verificar o que acontece se o circuito aleatoriamente se encontrar no estado 3 ($Q_1=Q_0=1$).

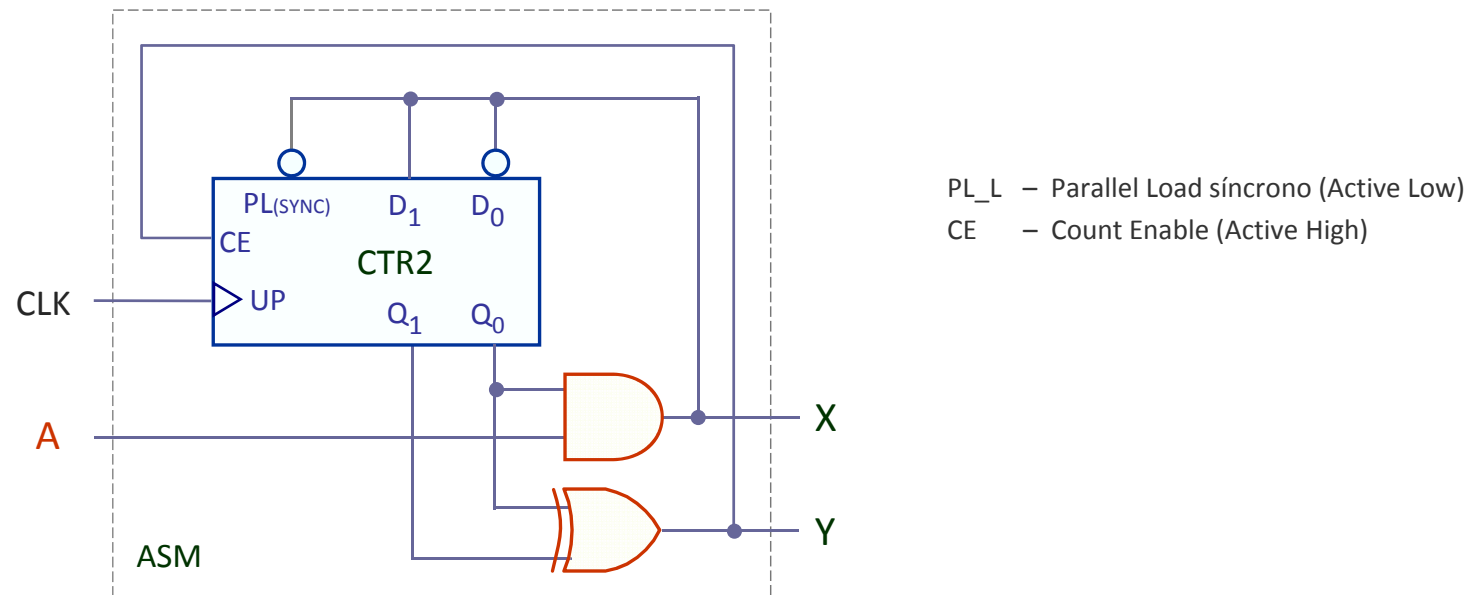


Diagrama lógico do circuito.

Exemplo 9-12

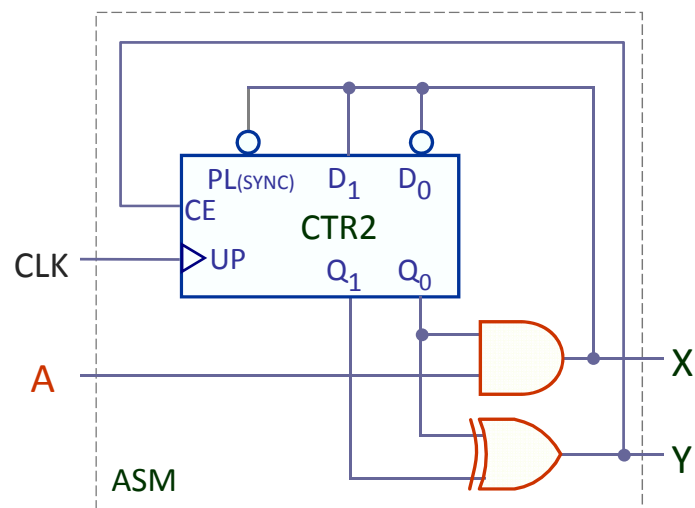


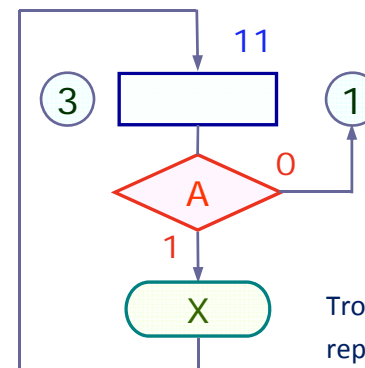
Diagrama lógico do circuito.

Estado Actual	Entrada Actual	Variáveis intermédias					Estado Seguinte	Saídas Actuais	
		PL_L	CE	D ₁	D ₀	Modo do CTR2		X	Y
Q ₁ , Q ₀	A						Q ₁ ⁺ , Q ₀ ⁺		
0 (0, 0)	–	0	–	0	1	LOAD	1 (0, 1)	0	0
1 (0, 1)	0	0	–	0	1	LOAD	1 (0, 1)	0	1
	1	1	1	–	–	COUNT UP	2 (1, 0)	1	1
2 (1, 0)	–	0	–	0	1	LOAD	1 (0, 1)	0	1
3 (1, 1)	0	0	–	0	1	LOAD	1 (0, 1)	0	0
	1	1	0	–	–	–	3 (1, 1)	1	0

Tabela de transição de estados representando o estado 3, as transição e as saídas.

Quando o circuito for colocado aleatoriamente no estado 3 instável:

- Se A=0, entrará no ciclo de contagem normal, através de uma acção LOAD que o levará ao estado 1.
- Se A=1 não sairá do estado 3 enquanto se mantiver esta condição porque estão inibidas as acções LOAD e COUNT (CE=0) – o contador mantém-se inactivo (em IDLE).



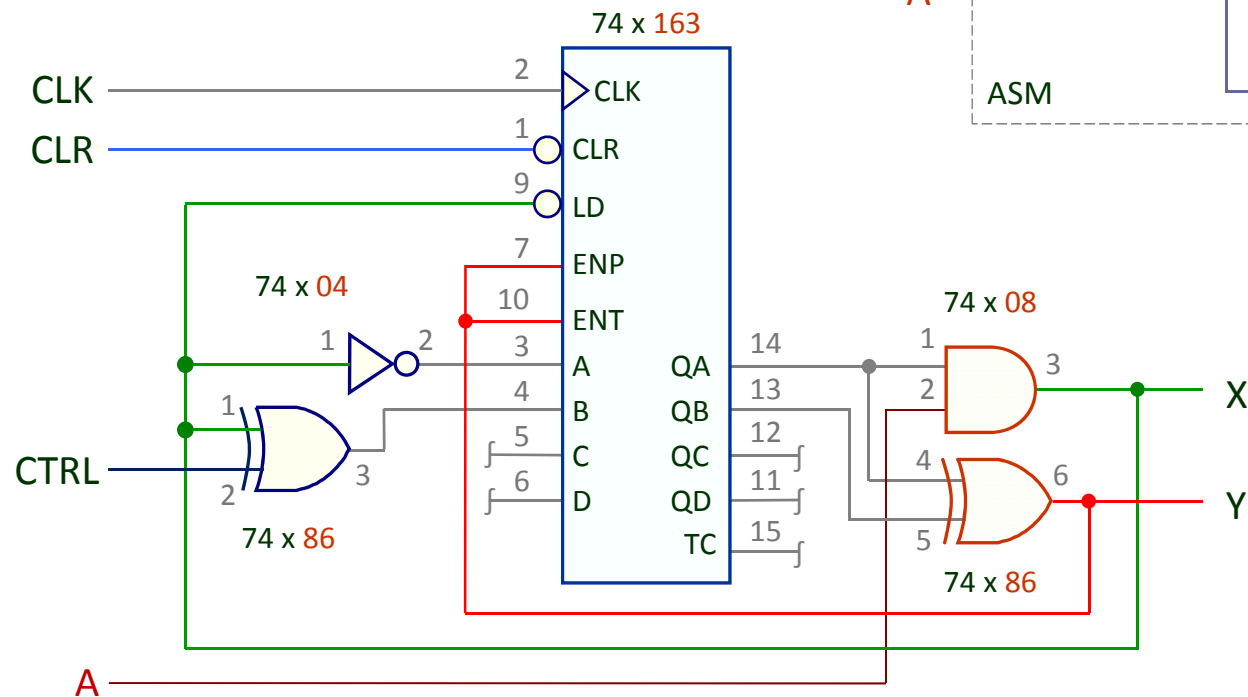
Troço do Fluxograma ASM representando o estado 3 e a transição condicionada para o estado 1.

ANÁLISE DE UM CIRCUITO SEQUENCIAL SÍNCRONO (Ex. 9-12-5)

9-70

Exemplo 9-12

Uma implementação real do circuito, que permite a verificação do Fluxograma-ASM num simulador ou num BREADBOARD, pode ser feita através dos componentes da série TTL indicadas em baixo. Para além das 4 entradas e 2 saídas presentes no diagrama de blocos usou-se uma entrada adicional CTRL que permite forçar o estado 3 e flexibilizar a verificação.



Implementação do circuito original com circuitos da série 74 x.

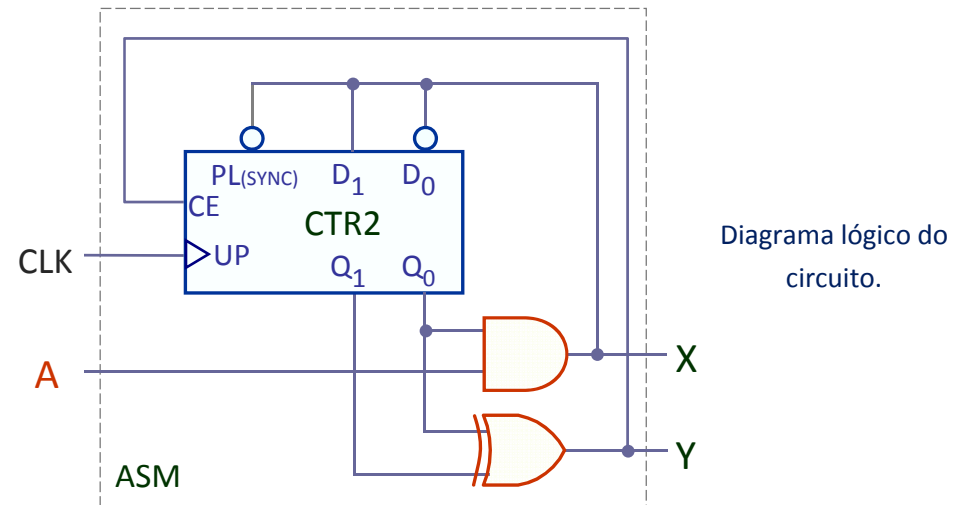
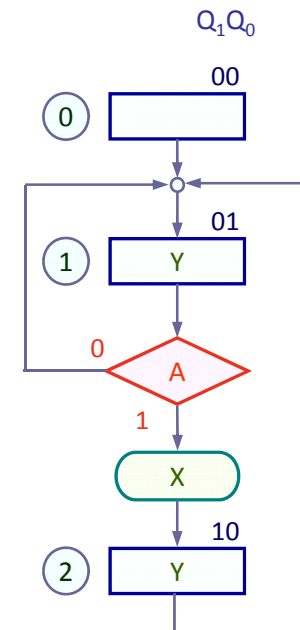


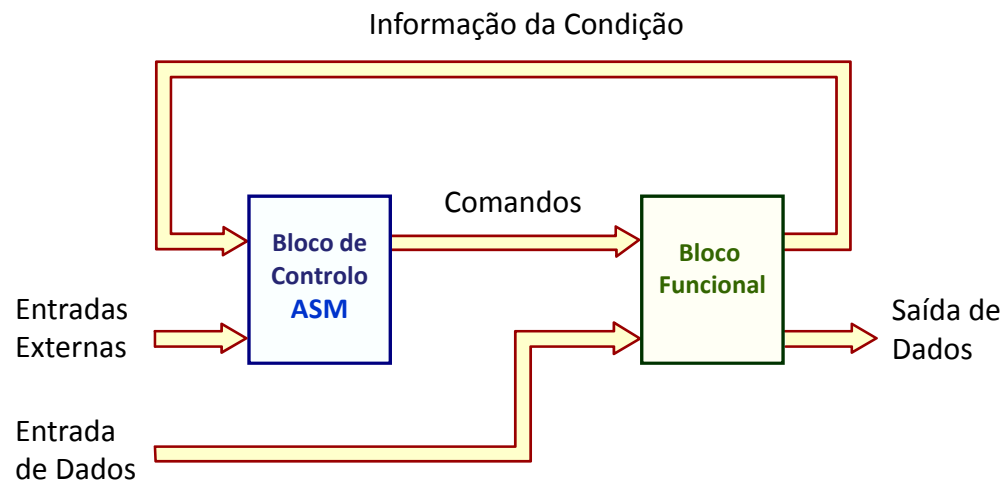
Diagrama lógico do circuito.



Fluxograma-ASM.



Tal como a construção de um Fluxograma-ASM tem similitudes com a escrita de um programa de computador, os procedimentos de síntese dos circuitos têm afinidades com os procedimentos de compilação de programas – e de conversão da descrição em linguagem de alto nível na sua realização em baixo nível. Ver-se-á agora como distribuir a funcionalidade de sistemas complexos em partições.



Partição de um sistema complexo em Bloco Funcional e Bloco de Controlo.

O Projecto Modular envolve a partição entre:

- **Blocos Funcionais** (acções primitivas desempenhadas por módulos já existentes - bibliotecas).
- **Blocos de Controlo**.

Envolve ainda a construção:

- do **Diagrama de Blocos**, que reflecte os subsistemas constituintes, e o percurso de dados entre eles. As entradas e as saídas devem ser caracterizadas por mnemónicas.
- do **Algoritmo** a que obedece o Bloco de Controlo – um Fluxograma ASM ou Diagrama de Estados (ou outro mais apropriado) que estabeleça a estratégia de conjugação para se conseguir o objectivo desejado.

Exemplo 9-13

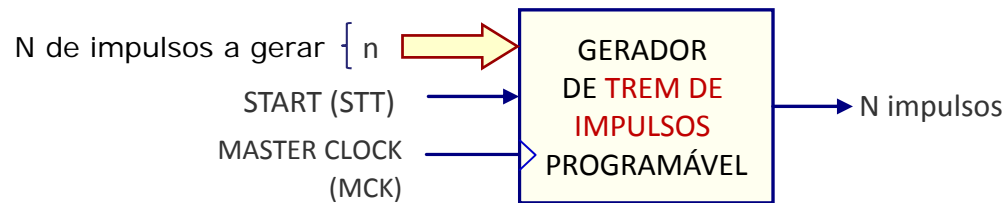
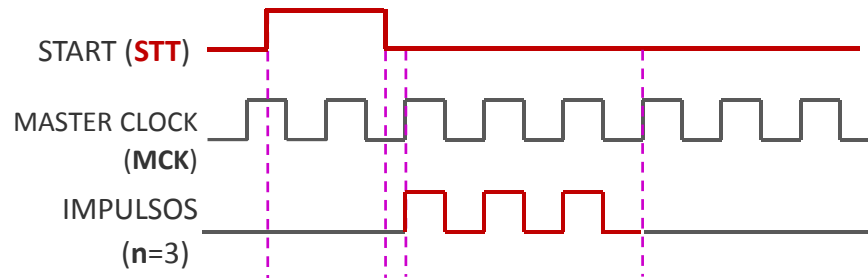


Diagrama de blocos: caixa-preta representativa da síntese do sistema a programar.



Timing a que o sistema a programar deve obedecer.

OBJECTIVO

Pretende-se projectar um sistema que pode ser programado com um número n , e, por acção de um ciclo completo de activação e desactivação de um interruptor de pressão $START$ (STT), é compelido a enviar para o exterior um trem de n impulsos ao ritmo de um Master Clock (MCK).

O trem de impulsos só começa a ser gerado a partir da primeira transição ascendente do MCK, após a transição descendente de STT, que terá, pelo menos, a duração de um impulso MCK.

O sinal STT é assíncrono relativamente a MCK.

Este é um exemplo de um projecto que vai ser desenvolvido em termos **modulares**, envolvendo um **Bloco Funcional** – que desempenha acções standard disponíveis em circuitos já realizados (neste caso um Contador Decrescente Programável) – e um **Bloco de Controlo**, projectado como máquina algorítmica a partir de um Fluxograma-ASM.

Exemplo 9-13

O sistema é pois constituído por dois blocos:

- Um **Bloco Funcional** – contador programável CTR.
- Um **Bloco de Controlo** – ASM.

O **bloco funcional** deve referir a acção que o módulo desempenha (neste caso um Contador Decrescente Programável CTR), os sinais de controlo a que obedece (**PLL**), e os sinais de controlo que põe disponíveis (**BRW**), e a natureza ACTIVE LOW ou ACTIVE HIGH desses sinais.

O **bloco de controlo ASM** toma como entradas sinais de controlo exteriores ao sistema (**STT**), ou oriundos dos módulos funcionais (**BRW**). Apresenta como saída sinais de controlo (**PLL** e **CKE**) que irão comandar os módulos funcionais, e eventualmente saídas para o exterior.

O sinal **CLOCK ENABLE (CKE)** gerado pelo bloco de controlo intercepta, através de uma porta AND, o **MASTER CLOCK (MCK)**, tanto para a saída **n** do sistema como para a entrada **COUNT DOWN (CDN)** do contador.

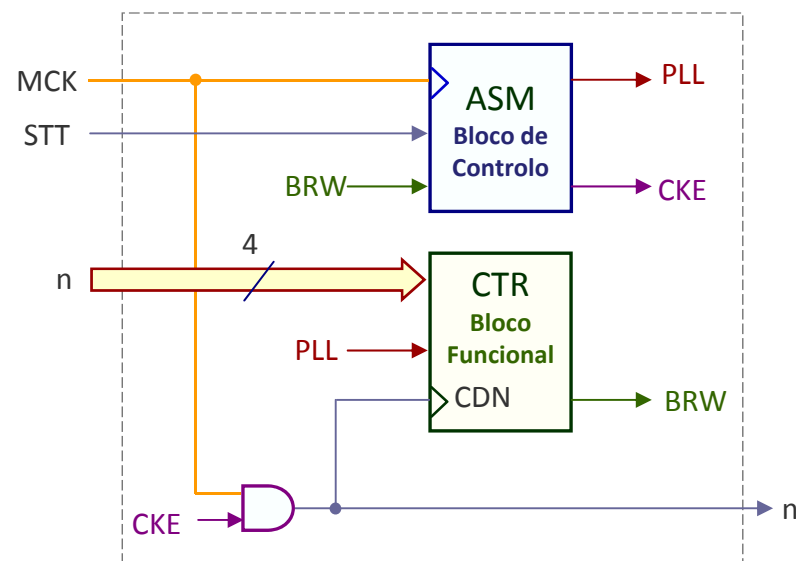


Diagrama de blocos do sistema a projectar.

ASM – Módulo de Controle
CTR – Contador Programável
CDN – COUNT DOWN
MCK – MASTER CLOCK

STT – START
PLL – PARALLEL LOAD
CKE – CLOCK ENABLE
BRW – BORROW

Mnemónicas dos sinais utilizados.

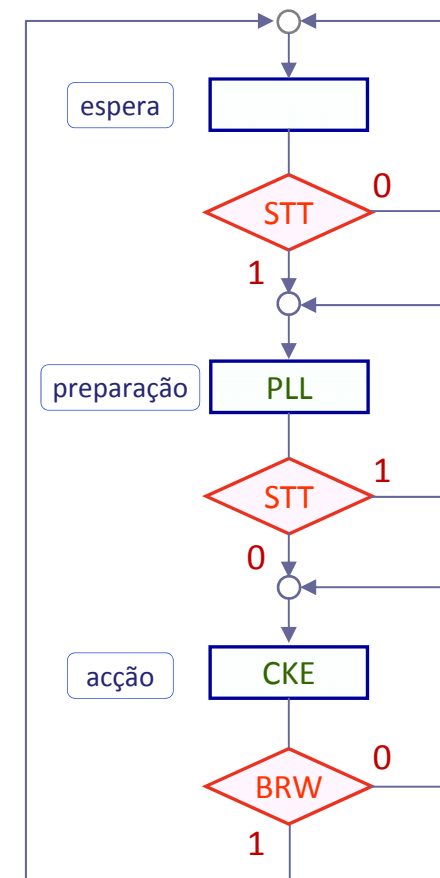
Exemplo 9-13

A primeira aproximação ao Fluxograma-ASM é o de uma máquina de Moore que possui **três estados**, e em que todas as saídas são incondicionais de estado. Os dois primeiros estados têm uma acção edge-triggered relativamente à acção STT (START).

A máquina só passa do **estado de espera** para o **estado de acção** após um ciclo completo de actuação do botão STT (activação e desactivação). Ao ser premido, este botão, leva o sistema do **estado de espera** para um **estado de preparação**.

O **estado de preparação** intermédio promove o PARALEL LOAD (PLL) do contador CTR (assíncrono para o contador 74x193), carregando-o com o código binário de **n** – número de impulsos a gerar. Ao deixar de ser premido o botão STT é que se permite que a máquina passe ao **estado de acção**.

O **estado de acção** mantém o CLOCK ENABLE activo (CKE=1), permitindo a passagem dos impulsos de CLK para a saída do sistema e para o contador. Este estado é mantido enquanto não ocorrer o sinal BRW indicativo de que o contador decrescente chegou a 0. Quando tal ocorrer a máquina abandona, na próxima transição de MCK, o **estado de acção**, regressando ao **estado de espera**.



Fluxograma-ASM pretendido na forma de máquina de Moore.

Exemplo 9-13

A implementação do Fluxograma-ASM anterior, com **três estados**, implica a utilização de **dois** flip-flops. O Fluxograma-ASM pode ser simplificado de modo a envolver só **dois estados**, para o que basta a utilização de **um** único flip-flop.

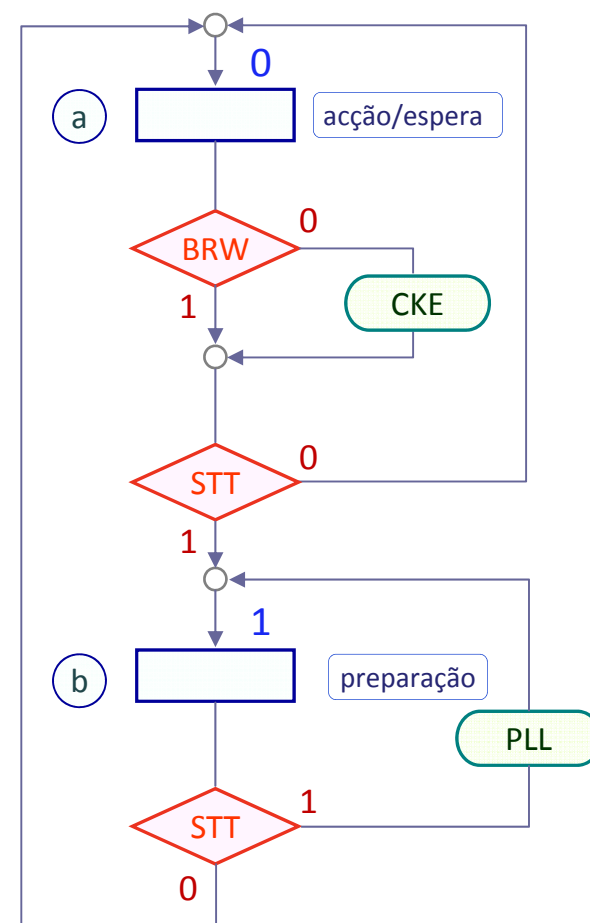
Para isto acontecer impõe-se que a saída CKE correspondente à acção de CLOCK ENABLE seja **condicionada** por estado e por entradas.

O primeiro estado toma o carácter de **ACÇÃO/ESPERA** – a máquina permanece neste estado enquanto o botão **STT** estiver desactivado. A saída **CKE** está activa enquanto não ocorrer BRW.

Logo que ocorrer BRW, sinal de que o contador decrescente chegou ao fim da contagem, CKE deixa de estar activo, inibindo a passagem dos impulsos MCK para a saída.

O sinal **BORROW** (**BRW**) gerado pelo contador programável é indicativo de que o contador está no estado 0, com a entrada de clock COUNT DOWN (CDN) também a 0 (requisito adicional para o contador 74LS193 utilizado). O sinal é acolhido pelo bloco de controlo como entrada.

BRW permanecerá activo e o estado toma o carácter de **ESPERA**, até que nova acção STT retire a máquina deste estado e a leve novamente para o estado de **PREPARAÇÃO**.



Fluxograma-ASM pretendido na forma de máquina de Mealey.

Exemplo 9-13

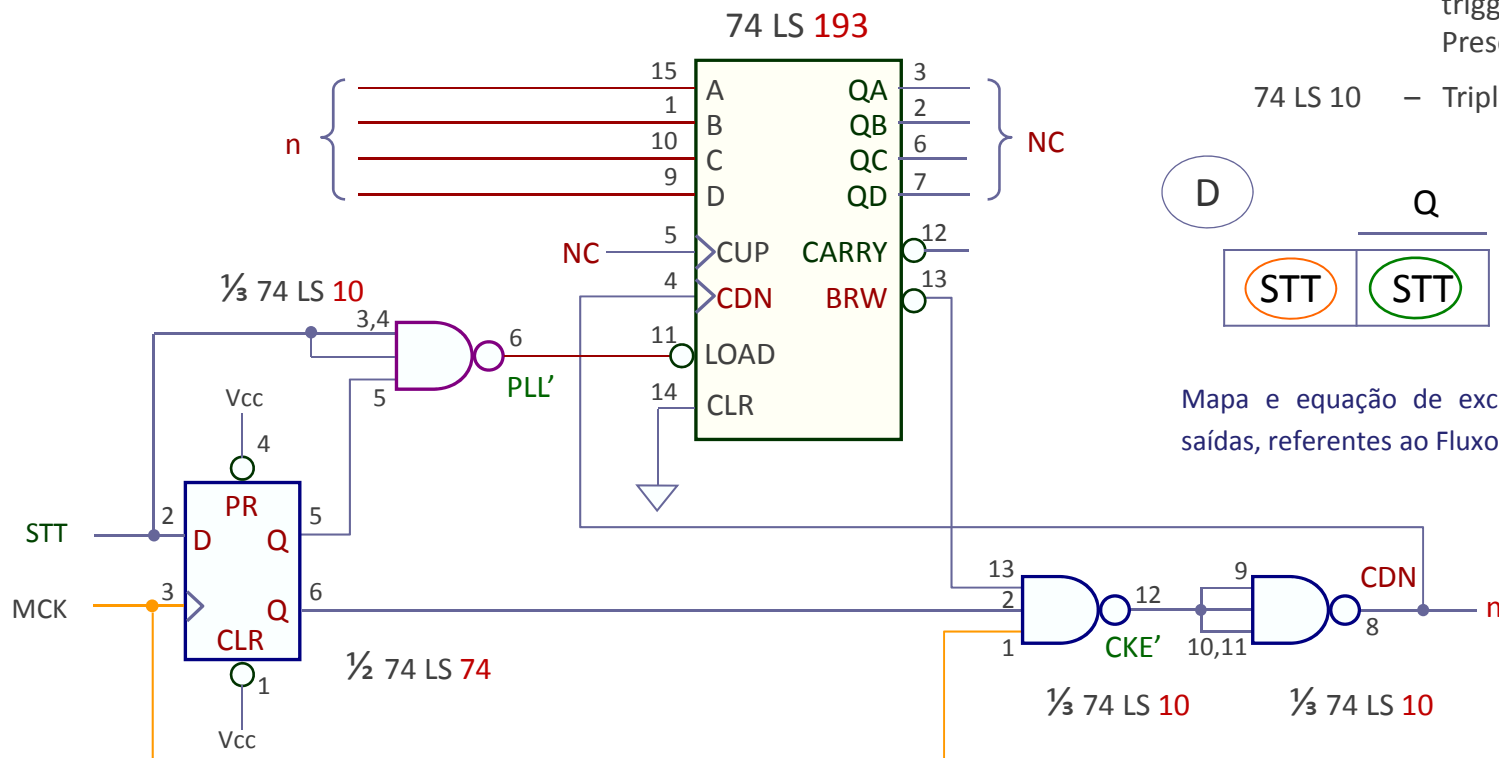
NC = not connected

Designação dos circuitos TTL utilizados.

74 LS 193 – 4-Bit Binary Counter with Dual Clock (with asynchronous Parallel Load)

74 LS 74 – Dual D-type Positive-edge-triggered Flip-flop With Preset and Clear

74 LS 10 – Triple 3-Input NAND Gate



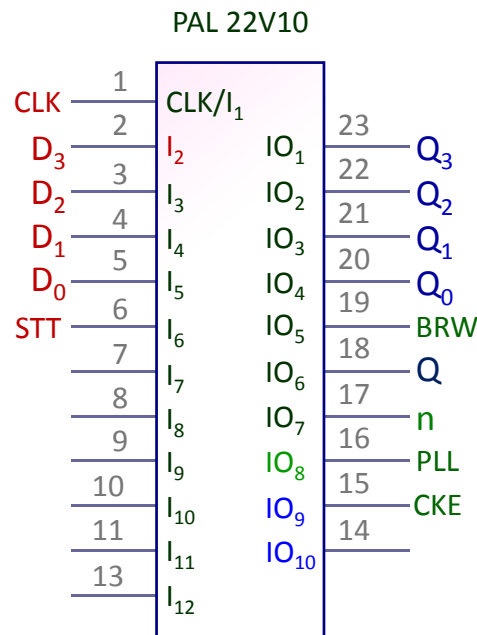
$$\begin{aligned} Q^+ &= D = STT \\ PLL &= Q STT \\ CKE &= Q' BRW' \end{aligned}$$

Mapa e equação de excitação do flip-flop D, e das saídas, referentes ao Fluxograma-ASM do slide anterior.

Esquema lógico do circuito final, envolvendo um módulo contador 74LS193, um flip-flop 74LS74 – só é utilizado um de dois circuitos idênticos disponíveis neste circuito integrado – e 3 portas NAND de 3 entradas, correspondentes ao circuito integrado 74LS10.



Exemplo 9-13



Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

```

...
Device      p22v10;

/* Inputs */
PIN 1       = MCK;
PIN [5..2]  = [D0..3];
PIN 6       = STT;

/* Outputs */
PIN [20..23] = [Q0 ..3];
PIN 19      = BRW;
PIN 18      = Q;
PIN 17      = N;
PIN 16      = PLL;
PIN 15      = CKE;

/* Definicao de Campos */
FIELD COUNT = [Q3..0];
FIELD DATA = [D3..0];

/* Configuracao dos Flip-flops */
COUNT.AR = 'b'0;
COUNT.SP = 'b'0;
Q.AR      = 'b'0;
Q.SP      = 'b'0;

/* Sequence Count */
SEQUENCE COUNT {
    PRESENT 'd'0
    IF !MCK # BRW OUT BRW;
    DEFAULT NEXT 'd'0;

    $REPEAT i = [1..15]
    PRESENT 'd'{i}
    IF !N & !PLL NEXT 'd'{i-1};
    DEFAULT NEXT 'd'{i};
    $REPEND
}
APPEND COUNT.D = PLL & DATA;

/* Sequence Start */
SEQUENCE Q {
    PRESENT 'd'0
    IF !BRW OUT CKE;
    IF !STT NEXT 'd'0;
    DEFAULT NEXT 'd'1;

    PRESENT 'd'1
    IF STT OUT PLL;
    IF STT NEXT 'd'1;
    DEFAULT NEXT 'd'0;
}

N = CKE & MCK;

```

Troço do programa CUPL correspondente à implementação em PAL ao lado.

Nesta implementação coexistem duas máquinas algorítmicas ASM na mesma PAL 22V10 partilhando o mesmo clock: uma realiza o bloco de controlo, a outra implementa o contador decrescente programável com Parallel Load (PLL) síncrono (PLL é síncrono ao contrário da implementação em lógica discreta exemplificada com o contador 74x193, em que o PLL é assíncrono).



Exemplo 9-14

OBJECTIVO

Pretende-se obter um circuito com uma entrada X e uma saída Z que identifique a ocorrência da sequência **101** na sua entrada. Quando isso ocorrer, e só nessas circunstâncias, a sua saída deve exibir o valor 1. As sequências podem ser sobrepostas. Pretende-se obter:

- O diagrama de estados do circuito na forma de Grafo de estados.
- O diagrama de estados do circuito na forma de Fluxograma-ASM correspondente a uma máquina de Mealey.
- A tabela de transição de estados, os mapas de Karnaugh e as equações de excitação com flip-flops D.
- O diagrama lógico do circuito em Máquina de Mealey.
- A transformação da Máquina de Mealey numa Máquina de Moore equivalente evidenciando:
 - As equações de entrada dos flip-flops e da saída Z;
 - O diagrama lógico da Máquina de Moore.
- A configuração de pinos e os troços do programa CUPL correspondentes às implementações em PAL.

RESOLUÇÃO

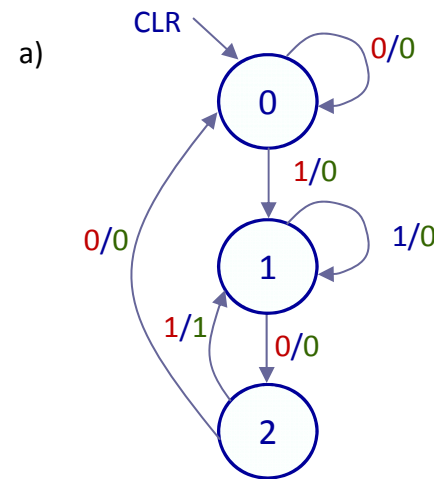


Diagrama de estados do detector da sequência 101, usando um modelo de Mealey.

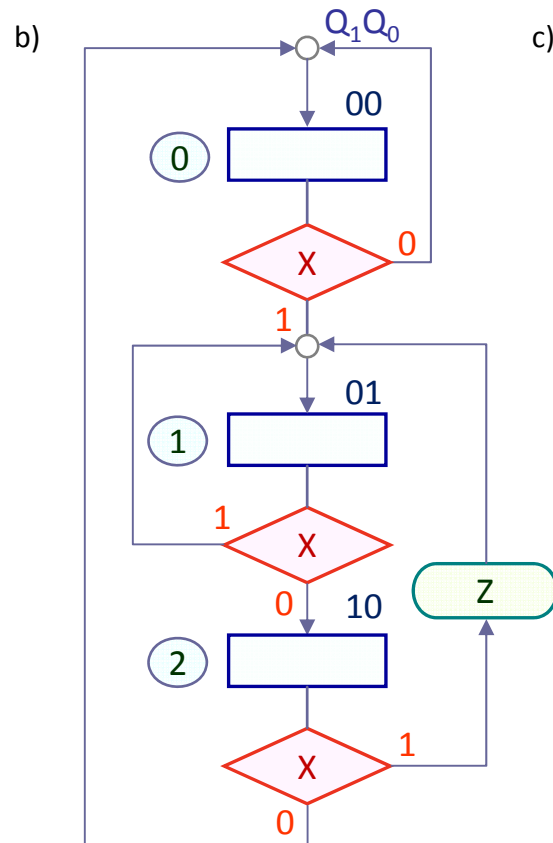
A entrada de CLEAR (C) é assíncrona e sendo actuada no arranque (ligação) do circuito permite garantir um RESET e a inicialização pelo estado **0**.

No estado 0, o circuito pode receber como entrada um 0 ou um 1. O processo de detecção só é iniciado quando recebe um 1, passando a máquina ao estado 1. No estado 1, uma entrada 0 levá-la-á ao estado 2, enquanto que uma entrada 1 mantê-la-á no mesmo estado. No estado 2, se receber uma entrada de 1, completa a sequência e gera a saída Z. Caso contrário, não pode a sequência ser terminada e o circuito volta ao seu estado inicial.

DETECTOR DE SEQUÊNCIA (Ex. 9-14-2)

9-79

Exemplo 9-14



Fluxograma-ASM em Máquina de Mealey.

D ₀		Q ₀
Q ₁	0	1
	X	X

$$D_0 = X$$

MAE		Q ₀
Q ₁	0	1
	0	1

D ₁		Q ₀
Q ₁	0	1
	0	X'

$$D_1 = Q_0 X'$$

c)

Estado Actual	Entrada Actual	Estado Seguinte	Saída Actual
Q ₁ Q ₀	X	Q ₁ ⁺ Q ₀ ⁺	Z
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	1 0	0
0 1	1	0 1	0
1 0	0	0 0	0
1 0	1	0 1	1
1 1	0	1 0	0
1 1	1	0 1	0

Tabela de transição de estados, mapas de Karnaugh e equações lógicas (em baixo).

Z		Q ₀
Q ₁	0	1
	X	

$$Z = Q_0' Q_1 X$$

O estado 3 nunca é visitado em condições normais. As transições a partir deste estado decorrem da utilização das indiferenças nos mapas-K.

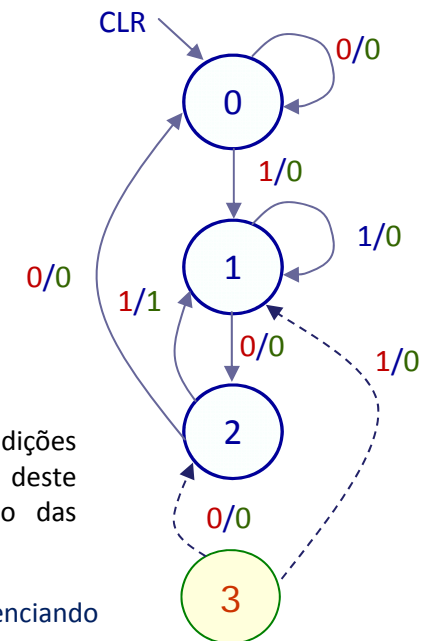
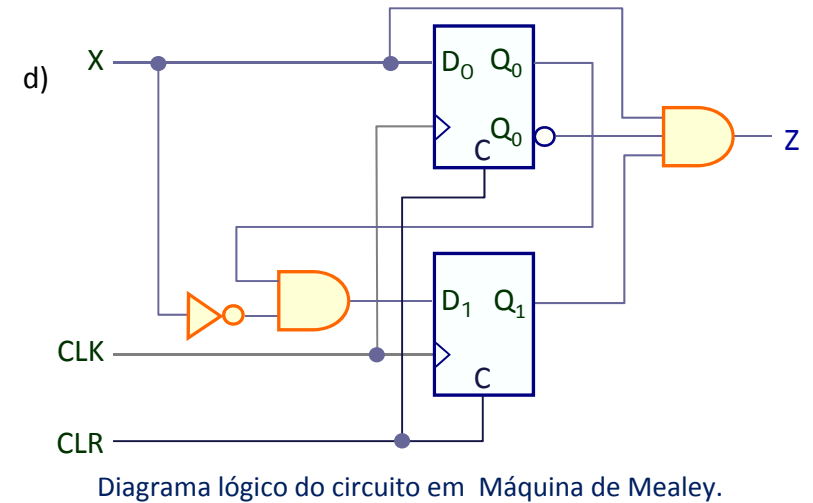


Diagrama de estados evidenciando o estado inexistente.

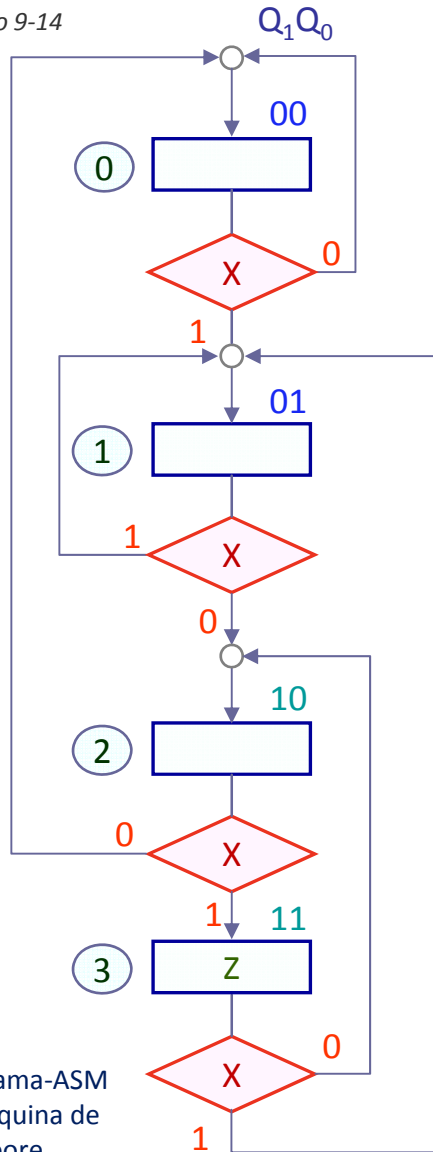


DETECTOR DE SEQUÊNCIA (Ex. 9-14-3)

9-80

Exemplo 9-14

e)



Fluxograma-ASM em Máquina de Moore.

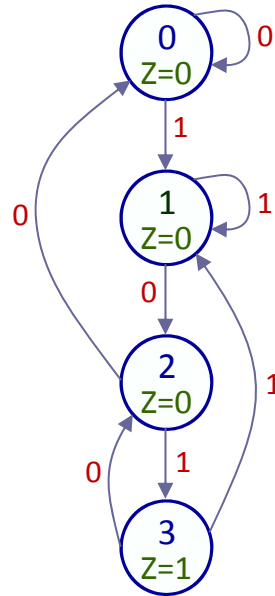


Diagrama de estados usando um modelo de Moore.

MAE

	Q ₀	
	0	1
Q ₁	0	1
	2	3

Z

	Q ₀	
Q ₁		1

$$Z = Q_0 Q_1$$

D₁

	Q ₀	
	0	X'
Q ₁	X	X'

D₀

	Q ₀	
	X	X
Q ₁	X	X

Mapas de Karnaugh e equações lógicas.

$$D_1 = Q_0 X' + Q_0' Q_1 X$$

$$D_0 = X$$

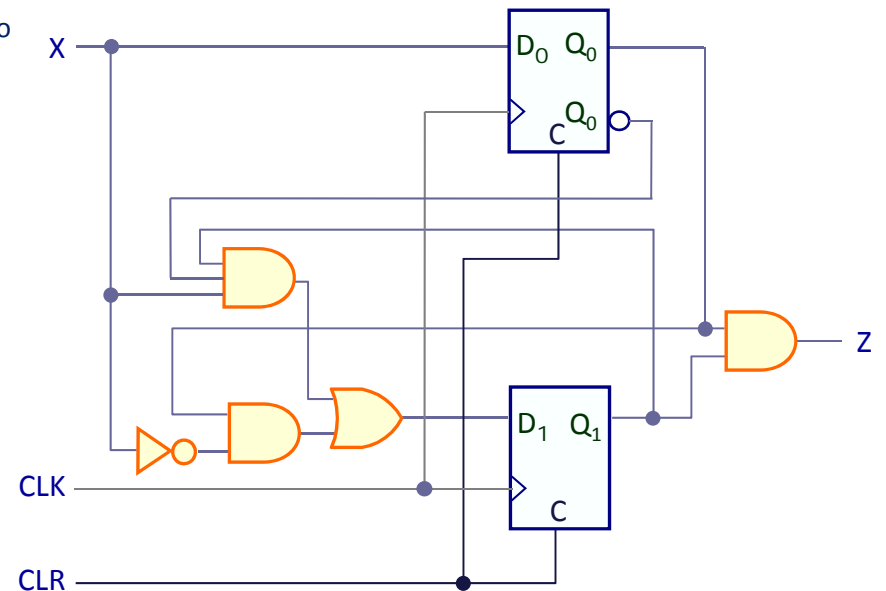


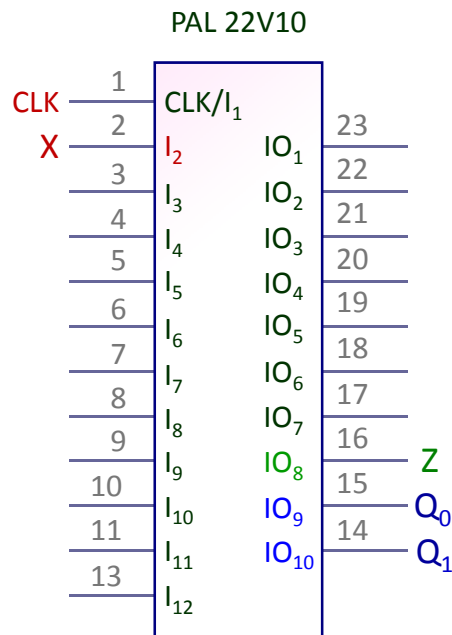
Diagrama lógico do circuito em Máquina de Moore.



DETECTOR DE SEQUÊNCIA EM PAL (Ex. 9-14-4)

9-81

Exemplo 9-14



Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

```

/***** INPUT PINS *****/
PIN 2 = X      ;

/***** OUTPUT PINS *****/
PIN 16 = Z      ;
PIN [15..14] = [Q0..1] ;

/***** BODY *****/
[Q0..1].SP='b'0;
[Q0..1].AR='b'0;

SEQUENCE [Q1,Q0]{
    PRESENT 0
        IF !X NEXT 0;
        DEFAULT NEXT 1;
    PRESENT 1
        IF X NEXT 1;
        DEFAULT NEXT 2;
    PRESENT 2
        IF X OUT Z;
        IF X NEXT 1;
        DEFAULT NEXT 0
}
    
```

Troço do Código CUPL da PAL sequencial correspondente à máquina de Mealey.

```

/***** INPUT PINS *****/
PIN 2 = X      ;

/***** OUTPUT PINS *****/
PIN 16 = Z      ;
PIN [15..14] = [Q0..1] ;

/***** BODY *****/
[Q0..1].SP='b'0;
[Q0..1].AR='b'0;

SEQUENCE [Q1,Q0]{
    PRESENT 0
        IF !X NEXT 0;
        DEFAULT NEXT 1;
    PRESENT 1
        IF X NEXT 1;
        DEFAULT NEXT 2;
    PRESENT 2
        IF X NEXT 3;
        DEFAULT NEXT 0;
    PRESENT 3
        OUT Z;
        IF X NEXT 2;
        DEFAULT NEXT 1;
}
    
```

Troço do Código CUPL da PAL sequencial correspondente à máquina de Moore.

Exemplo 9-15

OBJECTIVO

Realizar o diagrama de estados de um circuito sequencial síncrono do tipo **Máquina de Moore** com uma entrada X e uma saída Z.

A saída deve permanecer 0 até que a máquina tenha detectado a seguinte sequência de bits na entrada:

- dois zeros (0s) e dois uns (1s) consecutivos.

A ordem de ocorrência é irrelevante podendo entre a ocorrência de um grupo de dois zeros (00) e um grupo de dois uns (11) ocorrer outros padrões.

São válidos padrões do tipo: **00**1010**11**;
1100; **11**01**00**.

Quando a ocorrência tiver sido detectada a saída Z ficará 1 e a máquina recomeça o ciclo de detecção no próximo impulso de relógio.

RESOLUÇÃO

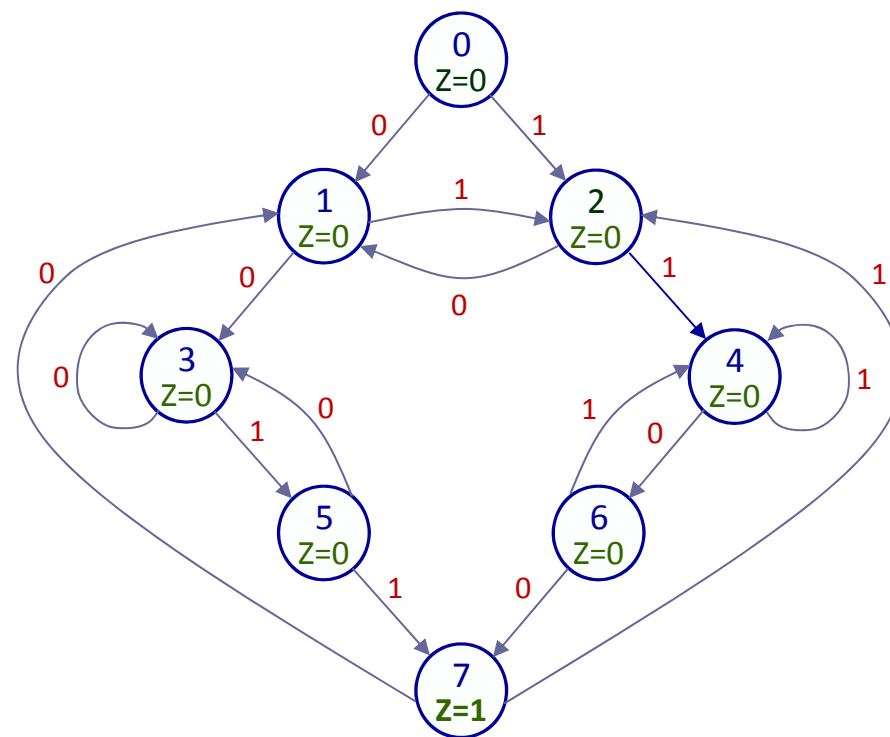


Diagrama de estados do detector da sequência usando um modelo de Moore.

DETECTOR DE SEQUÊNCIA EM MÁQUINA DE MOORE (Ex. 9-15-2)

9-83

Exemplo 9-15

```

/***** INPUT PINS *****/
PIN 2 = X      ;
PIN 13 = re    ;
/***** OUTPUT PINS *****/
PIN 17 = Z     ;
PIN [16..14] = [Q0..2] ;
/***** BODY *****/
[Q0..2].SP='b'0;
[Q0..2].AR='b'0;
SEQUENCE [Q2,Q0]{
    PRESENT 0
        IF !X & !re NEXT 1;
        IF X & !re NEXT 2;
        DEFAULT NEXT 0;
    PRESENT 1
        IF !X & !re NEXT 3;
        IF X & !re NEXT 2;
        DEFAULT NEXT 0;
    PRESENT 2
        IF !X & !re NEXT 1;
        IF X & !re NEXT 4;
        DEFAULT NEXT 0;
    PRESENT 3
        IF !X & !re NEXT 3;
        IF X & !re NEXT 5;
        DEFAULT NEXT 0;
    ...
}

```

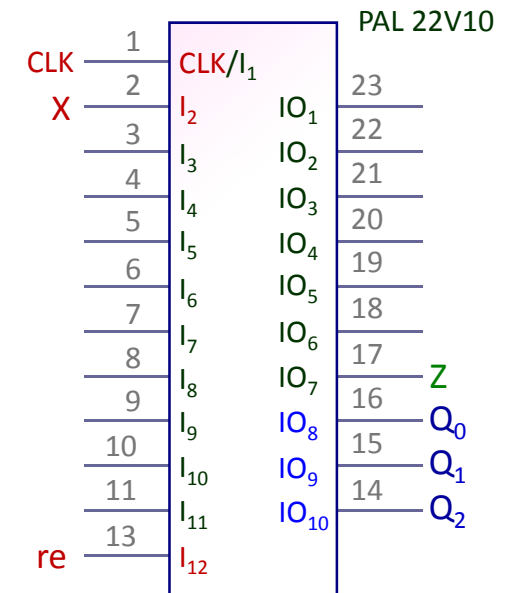
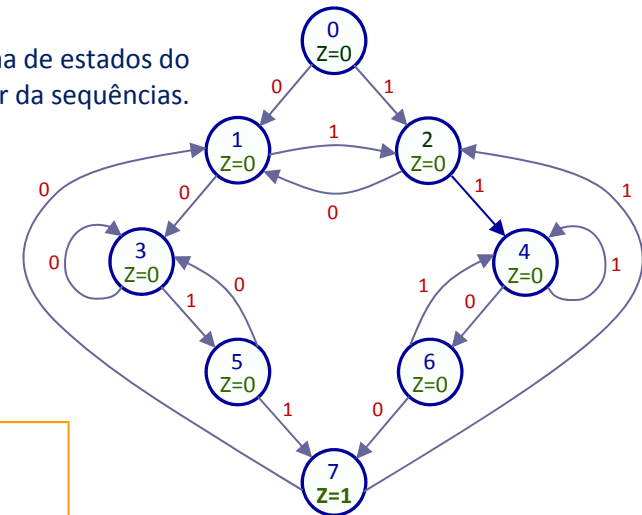
Nota: foi introduzido um Reset síncrono (re) na máquina sequencial através do pino de entrada 13.

```

...
PRESENT 4
    IF !X & !re NEXT 6;
    IF X & !re NEXT 4;
    DEFAULT NEXT 0;
PRESENT 5
    IF !X & !re NEXT 3;
    IF X & !re NEXT 7;
    DEFAULT NEXT 0;
PRESENT 6
    IF !X & !re NEXT 7;
    IF X & !re NEXT 4;
    DEFAULT NEXT 0;
PRESENT 7
    OUT Z
    IF !X & !re NEXT 1;
    IF X & !re NEXT 2;
    DEFAULT NEXT 0;
}

```

Diagrama de estados do detector da sequências.



Símbolo lógico da PAL assinalando os pinos de entrada e saída utilizados.

Troço do Código CUPL da PAL sequencial correspondente à máquina de Moore.



1. **LSD-9 – MÁQUINAS DE ESTADO ALGORÍTMICAS**
2. Taxonomia dos Circuitos Lógicos – Combinatórios, Sequenciais Síncronos e Sequenciais Assíncronos
3. Circuito Sequencial Síncrono – Modelo Geral
4. Circuito Sequencial Síncrono – Modelo Geral
5. Circuitos Sequenciais – Modelo de Moore
6. Circuitos Sequenciais – Modelo de Mealy
7. Síntese de um Circuito Sequencial com Flip-flops T (Ex. 9-1-1)
8. Diagrama de Estados do Circuito a Sintetizar (Ex. 9-1-2)
9. Fluxograma-ASM do Circuito a Sintetizar (Ex. 9-1-3)
10. Fluxograma-ASM com Codificação de Estados (Ex. 9-1-4)
11. Tabelas e Mapas de Karnaugh intervenientes no Processo de Síntese (Ex. 9-1-5)
12. Funções Geradoras das Entradas dos Flip-flops e das Saídas Exteriores (Ex. 9-1-6)
13. Diagrama Lógico do Circuito (Ex. 9-1-7)
14. Estados não Permitidos (Ex. 9-1-8)
15. Implementação do Circuito Sequencial em PAL com a Estrutura Sequence (Ex. 9-1-9)
16. Implementação do Circuito Sequencial em PAL sem Estrutura Sequence (Ex. 9-1-10)
17. Símbolos da Notação ASM (Algorithmic State Machine)
18. Notação ASM (Algorithmic State Machine) – um Pouco de História
19. Síntese com Flip-flops T – Contador de Módulo Variável (Ex. 9-2-1)
20. Contador de Módulo Variável (Ex. 9-2-2)
21. Fluxogramas-ASM Incorrectos
22. Bloco ASM e Caminhos de Saída
23. Bloco ASM e Transições Condicionadas
24. Blocos-ASM Equivalentes – Saídas
25. Blocos-ASM Equivalentes – Transições de Estado



26. Síntese a partir do Fluxograma-ASM prédefinido (Ex. 9-3-1)
27. Síntese a partir do Fluxograma-ASM (Ex. 9-3-2)
28. Síntese a partir do Fluxograma-ASM (Ex. 9-3-4)
29. Síntese a partir do Fluxograma-ASM (Ex. 9-3-5)
30. Síntese a partir do Fluxograma-ASM (Ex. 9-3-6)
31. Fluxograma-ASM e Diagrama de Estados para uma Máquina de Moore
32. Fluxograma-ASM e Diagrama de Estados para uma Máquina de Mealey
33. Máquina de Mealey – representação de Saídas Incondicionais e Condicionadas
34. Síntese com Flip-flop D e Flip-flop T (Ex. 9-4-1)
35. Síntese com Flip-flop D e Flip-flop T (Ex. 9-4-2)
36. Síntese com Flip-flop D e Flip-flop T (Ex. 9-4-3)
37. Temporização dos Sinais numa Máquina de Estados Síncrona
38. Síntese com Flip-flop J-K e Flip-flop T (Ex. 9-5-1)
39. Síntese com Flip-flop J-K e Flip-flop T (Ex. 9-5-2)
40. Síntese com Flip-flop J-K e Flip-flop T (Ex. 9-5-3)
41. Síntese com Flip-flop J-K e Flip-flop T (Ex. 9-5-4)
42. Síntese com PAL22V10 (Ex. 9-5-5)
43. Síntese em Lógica Discreta, em PAL Combinatória (Flip-flops Externos), e em PAL Sequencial (Ex. 9-6-1)
44. Síntese em Lógica Discreta, em PAL Combinatória (Flip-flops Externos), e em PAL Sequencial (Ex. 9-6-2)
45. Síntese em Lógica Discreta, em PAL Combinatória (Flip-flops Externos), e em PAL Sequencial (Ex. 9-6-3)
46. Síntese em Lógica Discreta, em PAL Combinatória (Flip-flops Externos), e em PAL Sequencial (Ex. 9-6-4)
47. Síntese com Portas Lógicas e Flip-flops D, e com PAL Sequencial (Ex. 9-7-1)
48. Síntese com Portas Lógicas e Flip-flops D, e com PAL Sequencial (Ex. 9-7-2)
49. Síntese com Portas Lógicas e Flip-flops D, e com PAL Sequencial (Ex. 9-7-3)
50. Síntese a partir de uma Especificação em Forma Narrativa (Ex. 9-8-1)



51. Controlo de Abertura e Fecho de Porta (Ex. 9-8-2)
52. Controlo de Abertura e Fecho de Porta (Ex. 9-8-3)
53. Síntese a partir de uma Especificação em Forma Narrativa (Ex. 9-9-1)
54. Controlo de Abertura e Fecho de Portão (Ex. 9-9-2)
55. Controlo de Abertura e Fecho de Portão (Ex. 9-9-3)
56. Controlo de Abertura e Fecho de Portão (Ex. 9-9-4)
57. Controlo de Abertura e Fecho de Portão (Ex. 9-9-5)
58. Fluxograma-ASM obtido a Partir de um Troço de Código CUPL (Ex. 9-10-1)
59. Síntese pelo método One-hot – um Flip-flop por Estado
60. Síntese pelo método One-hot – um Flip-flop por Estado
61. Síntese pelo método One-hot – um Flip-flop por Estado
62. Síntese pelo método One-hot – um Flip-flop por Estado
63. Análise de um Circuito Sequencial Síncrono (Ex. 9-11-1)
64. Análise de um Circuito Sequencial Síncrono (Ex. 9-11-2)
65. Análise de um Circuito Sequencial Síncrono (Ex. 9-11-3)
66. Análise de um Circuito Sequencial Síncrono (Ex. 9-11-4)
67. Análise de um Circuito Sequencial Síncrono (Ex. 9-12-1)
68. Análise de um Circuito Sequencial Síncrono (Ex. 9-12-2)
69. Análise de um Circuito Sequencial Síncrono (Ex. 9-12-3)
70. Análise de um Circuito Sequencial Síncrono (Ex. 9-12-5)
71. Síntese Modular
72. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-1)
73. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-2)
74. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-3)
75. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-4)



- 76. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-5)
- 77. Síntese Modular – Gerador de Trem de Impulsos Programável (Ex. 9-13-6)
- 78. Detector de Sequência em Máquina de Mealey e Máquina de Moore (Ex. 9-14-1)
- 79. Detector de Sequência (Ex. 9-14-2)
- 80. Detector de Sequência (Ex. 9-14-3)
- 81. Detector de Sequência em PAL (Ex. 9-14-4)
- 82. Detector de Sequência em Máquina de Moore (Ex. 9-15-1)
- 83. Detector de Sequência em Máquina de Moore (Ex. 9-15-2)
- 84. LSD – 9 Índice 1
- 85. LSD – 9 Índice 2
- 86. LSD – 9 Índice 3
- 87. LSD – 9 Índice 4

