

Circuitos Lógicos



Módulo # 6

Projeto de Circuitos Sequenciais Síncronos



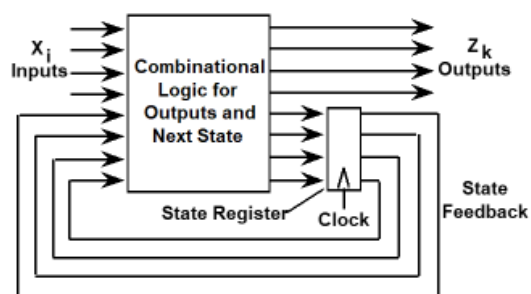
Projeto de Circuitos Sequenciais Síncronos



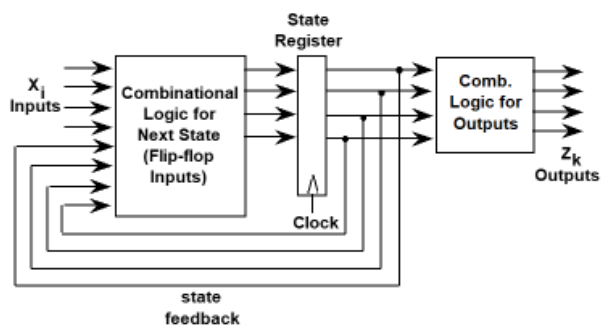
Algoritmo básico

- 1 - Entendimento do problema
- 2 - Análise das informações
- 3 - Diagrama de estados
 - 3.1 - Minimização de estados
 - 3.2 - Identificação de estados
 - 3.3 - Escolha do elemento de memória
- 4 - Implementação da lógica combinacional
 - 4.1 - Minimização da função
 - 4.2 - Redução do circuito





Máquina de Mealy



Máquina de Moore



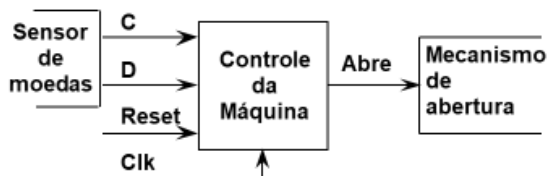
Projeto de Circuitos Sequenciais



Funcionamento geral

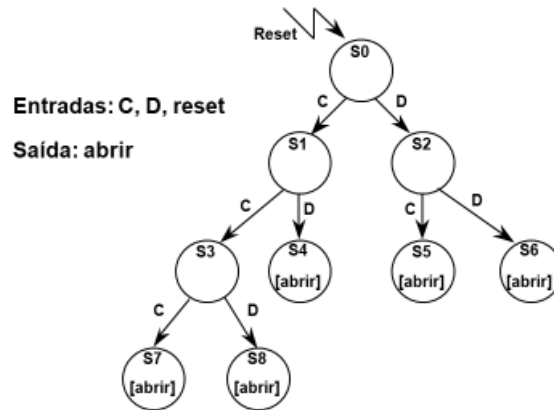
- entrega o chiclete para cada 15 centavos
- entrada de moedas de 10 (D) e 5 (C) centavos
- não há troco

Passo 1. *Entendimento do problema ...*
desenhe um diagrama de blocos



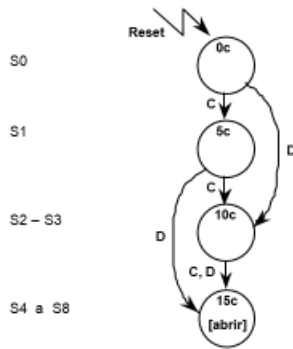
**Passo 2. Análise das informações ...
verificar as possíveis seqüências de entrada
e respectivas saídas**

Possibilidades :
 C + C + C
 C + D
 D + C
 D + D
 C + C + D



Handwritten signature

Passo 3.1 - Minimização e identificação de estados



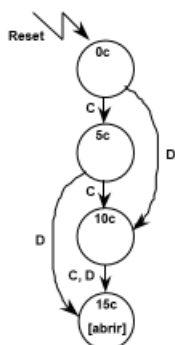
Estado Atual	Entradas		Próximo Estado	Saída abrir
	D	C		
0c	0	0	0c	0
	0	1	5c	0
	1	0	10c	0
5c	1	1	X	X
	0	0	5c	0
	0	1	10c	0
10c	1	0	15c	0
	1	1	X	X
	0	0	10c	0
15c	0	1	15c	0
	1	0	15c	0
	1	1	X	X
15c	X	X	15c	1



Handwritten signature



Passos 3.2 e 3.3 - Identificação de estados ...
Escolha do flip-flop ...



Estado atual Q ₁ Q ₀	Entradas D C	Prox. estado D ₁ D ₀	Saída abrir
0 0	0 0	0 0	0
	0 1	0 1	0
	1 0	1 0	0
	1 1	X X	X
0 1	0 0	0 1	0
	0 1	1 0	0
	1 0	1 1	0
	1 1	X X	X
1 0	0 0	1 0	0
	0 1	1 1	0
	1 0	1 1	0
	1 1	X X	X
1 1	0 0	1 1	1
	0 1	1 1	1
	1 0	1 1	1
	1 1	X X	X

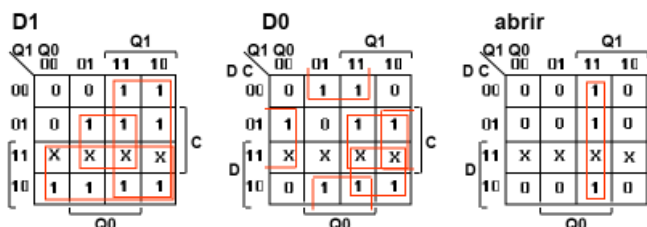


[Handwritten signature]



Passos 4.1 Minimização da função ...

Estado atual Q ₁ Q ₀	Entradas D C	Prox. estado D ₁ D ₀	Saída abrir
0 0	0 0	0 0	0
	0 1	0 1	0
	1 0	1 0	0
	1 1	X X	X
0 1	0 0	0 1	0
	0 1	1 0	0
	1 0	1 1	0
	1 1	X X	X
1 0	0 0	1 0	0
	0 1	1 1	0
	1 0	1 1	0
	1 1	X X	X
1 1	0 0	1 1	1
	0 1	1 1	1
	1 0	1 1	1
	1 1	X X	X



$$D1 = Q1 + D + Q0 C$$

$$D0 = C'Q0 + Q0'C + Q1 C + Q1 D$$

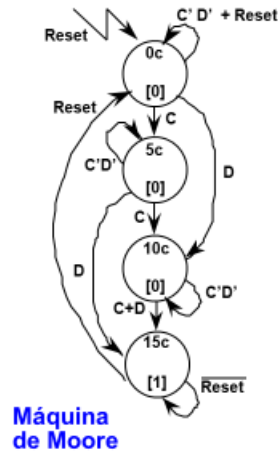
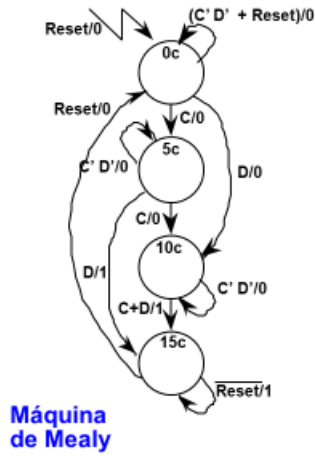
$$\text{abrir} = Q1 Q0$$

... como seria para FFP-JK ?



[Handwritten signature]

Projeto de Circuitos Sequenciais



Handwritten signature

Exercício: Identificação de padrões em seqüências finitas
 Seja um identificador de entrada X e saída Z.
 $Z=1$ sempre que a seqüência ...010... for observada,
 enquanto não aparecer a seqüência ...100....



passo 1: entendimento do problema

X: 00101010010...
 Z: 00010101000...



X: 11011010010...
 Z: 0000001000...

passo 2: fazer o diagrama de estados para as seqüências que devem ser reconhecidas, isto é, 010 and 100.

obs.: reutilizar estados sempre que possível.



Handwritten signature



Projeto de Circuitos Sequenciais

- Algoritmo básico
- 1 - Entendimento do problema
 - 2 - Análise das informações
 - 3 - Diagrama de estados
 - 3.1 - Minimização de estados
 - 3.2 - Identificação de estados
 - 3.3 - Escolha do elemento de memória
 - 4 - Implementação da lógica combinacional
 - 4.1 - Minimização da função
 - 4.2 - Redução do circuito



Redução de estados

Dois estados são equivalentes se, partindo-se destes, obtivermos as mesmas sequências de saída. Podemos, assim, fazer uma prévia redução de estados por simples inspeção visual.

Algoritmo de redução por classes

- 1 - fazer a tabela de estados inicial ;
- 2 - dividir os estados em classes, de acordo com as saídas apresentadas, isto é, estados que têm a mesma saída estarão na mesma grupo (classe) ;
- 3 - estabelecer, baseado nos próximos estados, as próximas classes de cada estado ;
- 4 - reagrupar em novas classes, de acordo com as próximas classes apresentadas ;
- 5 - repetir os itens 3 e 4 até que, em cada classe, todos os estados tenham as mesmas próximas classes ; estes estados são equivalentes.



Projeto de Circuitos Digitais

Combinacionais Sequenciais



Algoritmo
básico

- 1 - Entendimento do problema
- 2 - Análise das informações
- 3 – Tabela verdade 3 - Diagrama de estados
 - 3.1 - Minimização de estados
 - 3.2 - Identificação de estados
 - 3.3 - Escolha do elemento de memória
- 4 - Implementação da lógica combinacional
 - 4.1 - Minimização da função
 - 4.2 - Redução do circuito



Identificação de estados

Algoritmo para a determinação de adjacências

R1 - dois ou mais estados que têm os mesmos próximos estados devem ter identificações adjacentes ;

R2 - dois ou mais estados que são os próximos estados de um mesmo estado devem ter identificações adjacentes.

Do exemplo anterior :

	x=0	x=1		anterior	atual	próximo
a	b/0	c/0		e	a	b, c
b	c/0	c/0		a	b	c
c	c/0	e/1		a, b, c, e	c	c, e
e	c/0	a/1		c	e	a, c
			R1			R2

R1 : (ab), (ac), (ae), (bc), (be), (ce)

R2 : (bc), (ce), (ac)

	0	1
0	a	c
1	e	b




Projeto de Circuitos Digitais

Combinacionais Sequenciais

Algoritmo
básico

- 1 - Entendimento do problema
- 2 - Análise das informações
- 3 – Tabela verdade 3 - Diagrama de estados
 - 3.1 - Minimização de estados
 - 3.2 - Identificação de estados
 - 3.3 - Escolha do elemento de memória
- 4 - Implementação da lógica combinacional
 - 4.1 - Minimização da função
 - 4.2 - Redução do circuito



Algoritmo de redução por classes

- 1 - fazer a tabela de estados inicial ;
- 2 - dividir os estados em classes, de acordo com as saídas apresentadas, isto é, estados que têm a mesma saída estarão na mesma grupo (classe) ;
- 3 - estabelecer, baseado nos próximos estados, as próximas classes de cada estado ;
- 4 - reagrupar em novas classes, de acordo com as próximas classes apresentadas ;
- 5 - repetir os itens 3 e 4 até que, em cada classe, todos os estados tenham as mesmas próximas classes ; estes estados são equivalentes.

Identificação de estados

Algoritmo para a determinação de adjacências

- R1 - dois ou mais estados que têm os mesmos próximos estados devem ter identificações adjacentes ;
- R2 - dois ou mais estados que são os próximos estados de um mesmo estado devem ter identificações adjacentes.



Projeto de Circuitos Digitais

Combinacionais Sequenciais

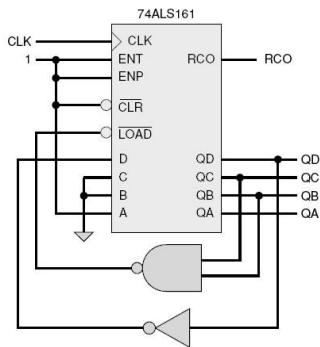


Algoritmo básico

- 1 - Entendimento do problema
- 2 - Análise das informações
- 3 - Tabela verdade 3 - Diagrama de estados
 - 3.1 - Minimização de estados
 - 3.2 - Identificação de estados
 - 3.3 - Escolha do elemento de memória
- 4 - Implementação da lógica combinacional
 - 4.1 - Minimização da função
 - 4.2 - Redução do circuito



Exercícios



Consulte as características do chip 74ALS161 (74161) e apresente :

- a) o diagrama de estados desse circuito, considerando as saídas $Q_D Q_C Q_B Q_A$
- b) qual é a relação da frequência de saída do MSB (Q_D) com a frequência de entrada do clock (CLK).



Exercícios

1 - Seja o circuito abaixo, com estados XYZ descritos pela lógica combinacional de controle dos flip-flops que o compõem.

$$Dx = XY' + XZ' + X'YZ \quad Dy = XY' + Y'Z + X'YZ' \quad Dz = X'Y + XZ$$

- Analisar este circuito, comentando o diagrama de estados encontrado ;
- Implementar uma máquina equivalente utilizando FFP-JK.

2 - Deseja-se um circuito sequencial síncrono que detecte, a cada 4 bits, as sequências **1100** ou **0100**.

- desenvolver o diagrama de estados mínimo para este detector, implementando uma Máquina de Moore ;
- desenvolver o diagrama de estados mínimo para este detector, implementando uma Máquina de Mealy .



Exercícios

3 - Projete um circuito que implemente uma máquina equivalente à descrita pela tabela de estados abaixo. Utilize, no projeto, mapas de excitação, além dos demais algoritmos (redução e identificação de estados).

est. atual	prox.est. / saída	
	l = 0	l = 1
A	A/0	B/0
B	B/0	C/1
C	D/0	F/0
D	C/0	F/0
E	G/0	D/1
F	F/1	H/0
G	E/0	C/1
H	B/0	E/0



Exercícios

Projetar e simular circuitos que detectem, numa linha serial, a presença das sequências 110 ou 101 ;

o detetor deve ser implementado utilizando -se, primeiramente, uma máquina de Moore e, em seguida, utilizando uma máquina de Mealy; comparar as duas máquinas e a equivalência dos seus resultados.

Implementar um contador de 3 bits que tenha um controle M e aja da seguinte forma :

M = 0 a contagem é ascendente na sequência binária;

M = 1 a contagem é ascendente em código de Gray (ponderado)



Deseja-se colocar um sinal de trânsito no cruzamento de duas ruas. Sensores C detetam a presença de carros na rua secundária. Se não há carros nesta rua, o sinal permanece verde para a rua principal. Se há carros naquela, o sinal passa de verde para amarelo e para vermelho, permitindo então o sinal verde na rua secundária, por um intervalo TL. Após este intervalo, o sinal passa de verde para amarelo e para vermelho, retornando o sinal da rua principal a verde. Mesmo que existam veículos na rua secundária, a principal deve ter sinal verde pelo menos por TL. Assumir a existência de um temporizador que gere os tempos TL, para os sinais verdes, e TS dos sinais amarelos, ambos em resposta a um set (ST) dos respectivos tempos.



Leitura indicada



Maini, A.K. "Digital Electronics – Principles and Integrated Circuits"

a) Sec. 8.3 – 8.11, pgs. 284 – 311

b) Sec. 9.1 – 9.13, pgs. 317 – 360

8

Flip-Flops and Related Devices

LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Operational basics of bistable, monostable, and astable multivibrators.
- Digital and linear integrated circuits for implementing multivibrator functions.
- Operational basics of *R-S*, *J-K*, toggle, and *D* flip-flops.
- Use of *J-K* flip-flop to construct other flip-flops.
- Timing parameters of flip-flops.
- Applications of flip-flops.
- Application-relevant data on commonly used flip-flops and related devices.

9

Counters and Registers

LEARNING OBJECTIVES

After completing this chapter, you will learn the following:

- Difference between asynchronous (or ripple) and synchronous counters.
- Operational principle of asynchronous and synchronous counters.
- Design methodology for asynchronous and synchronous counters.
- Designing counters with variable modulus.
- Designing counters with arbitrary sequences.
- BCD, decade counters, UP, DOWN, and UP/DOWN counters.
- Counters with PRESET and CLEAR features.
- Cascading counters.
- Decoding counters.
- Operational principle of different types of registers.
- Operational principle and features of register-based counters.
- ANSI representation of counters and registers available in IC form.
- Application-relevant information on counters and registers.

