

EEL7020 – Sistemas Digitais
Aula 9: Circuitos sequenciais
Máquinas de Moore e de Mealy

Prof. Djones Vinicius Lettnin
lettnin@eel.ufsc.br
<http://dlettnin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the author J. Guentzel.

Controller Design 3.4

- Five step controller design process

Step	Description
Step 1 <i>Capture the FSM</i>	Create an FSM that describes the desired behavior of the controller.
Step 2 <i>Create the architecture</i>	Create the standard architecture by using a state register of appropriate width, and combinational logic with inputs being the state register bits and the FSM inputs and outputs being the next state bits and the FSM outputs.
Step 3 <i>Encode the states</i>	Assign a unique binary number to each state. Each binary number representing a state is known as an <i>encoding</i> . Any encoding will do as long as each state has a unique encoding.
Step 4 <i>Create the state table</i>	Create a truth table for the combinational logic such that the logic will generate the correct FSM outputs and next state signals. Ordering the inputs with state bits first makes this truth table describe the state behavior, so the table is a state table.
Step 5 <i>Implement the combinational logic</i>	Implement the combinational logic using any method.

© F. Vahid – Adapted by D. Lettnin

Controller Example: Sequence Generator

- Want generate sequence 0001, 0011, 1100, 1000, (repeat)
 - Each value for one clock cycle
 - Common, e.g., to create pattern in 4 lights, or control magnets of a "stepper motor"

Inputs: none; Outputs: w,x,y,z
 $wxyz=0001$ $wxyz=1000$

Step 1: Create FSM

Step 2: Create architecture

Step 3: Encode states

Step 4: Create state table

Inputs	Outputs
s1 s0	w x y z n1 n0
A 0 0	0 0 0 0 1 0
B 0 1	0 0 1 1 1 0
C 1 0	1 1 0 0 0 1
D 1 1	1 0 0 0 0 0

Step 5: Create combinational circuit

© F. Vahid – Adapted by D. Lettnin

Example: Seq. Circuit to FSM (Reverse Engineering)

What does this circuit do?

Inputs: none; Outputs: w,x,y,z
 $y=s1'$
 $z=s1s0'$
 $n1=(s1 \text{ xor } s0)x$
 $n0=(s1 \text{ xor } s0)'x$

Step 1: Identify states and transitions

Step 2: Create state table

Inputs	Outputs
s1 s0 x	n1 n0 y z
A 0 0 0	0 0 0 1 0
B 0 0 1	0 1 0 1 0
C 1 0 0	0 0 0 0 1
D 1 0 1	1 0 0 0 1

Step 3: Create combinational circuit

Work backwards

Pick any state names you want

© F. Vahid – Adapted by D. Lettnin

Plano de Aula

Universidade Federal de Santa Catarina

- Registradores (Características temporais)
- Máquinas de Moore
- Máquinas de Mealy

5

D Flip-Flop

• **Flip-flop:** Bit storage that stores on clock edge, not level

• One design -- master-servant

- Two latches, output of first goes to input of second, master latch has inverted clock signal
- So master loaded when C=0, then servant when C=1
- When C changes from 0 to 1, master disabled, servant loaded with value that was at D just before C changed -- i.e., value at D during rising edge of C

Note: Hundreds of different flip-flop designs exist!

rising edges

Diagram showing the internal structure of a D flip-flop using two D-latches. The master D-latch has its clock input connected to the clock signal C through an inverter. Its output Qm is connected to the D input of the servant D-latch. The servant D-latch has its clock input connected to the clock signal C. Its output Qs is the final output of the flip-flop. The diagram also shows the waveforms for the clock C, the master output Qm, the servant output Qs, and the D input.

© F. Vahid – Adapted by D. Lettnin

6

Registradores

Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida

tsu = Tempo de Preparação (setup time)
“Tempo antes da borda ativa de ck (subida, neste caso) durante o qual a entrada D já deve estar em seu valor estável.”

© J. Guentzel – Adapted by D. Lettin

Registradores

Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida

th = Tempo de Manutenção (hold time)
“Tempo, a partir da borda ativa de ck (subida, neste caso), durante o qual a entrada D deve permanecer estável.”

© J. Guentzel – Adapted by D. Lettin

Registradores

Características Temporais de Flip-flops

Supor um Flip-flop D disparado pela Borda de Subida

tco (ou td) = Time from clock to output (ou Tempo ou Atraso de Carga)
“Atraso, em relação à borda ativa de ck (subida, neste caso), para o valor amostrado a partir da entrada D aparecer nas saídas Q e Q.”

© J. Guentzel – Adapted by D. Lettin

Registradores

Registrador com Carga Paralela

- Existe um FF para cada bit a ser armazenado
- Todos os FFs “obedecem” a um mesmo sinal de controle
- O termo “carga paralela” refere-se ao fato de existir uma entrada para cada bit, de modo que é possível carregar simultaneamente todos os bits do dado

© J. Guentzel – Adapted by D. Lettin

Registradores

Registrador com Carga Paralela

Exemplo de funcionamento (assumindo que tsu, th e tco são satisfeitos)

A cada borda de subida de “controle” um novo dado é amostrado e fica armazenado no registrador (até a próxima borda de subida de “controle”)

© J. Guentzel – Adapted by D. Lettin

Registradores

Cadenciamento de Sistemas Digitais

Registradores são usados para criar “barreiras temporais” que isolam os circuitos combinacionais

* (obs) :

1. Supondo que R1 e R2 sejam disparados pela borda de subida de ck.
2. Aproximação grotesca; falta considerar as características temporais dos registradores R1 e R2.

© J. Guentzel – Adapted by D. Lettin

Registradores

Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

13

Registradores

Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

14

Registradores

Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

15

Registradores

Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

16

Registradores

Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

17

Registradores

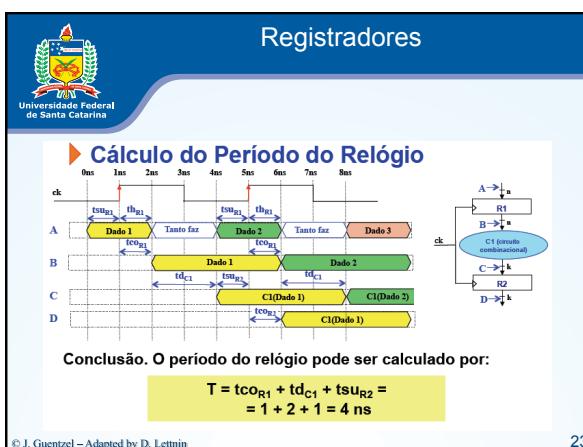
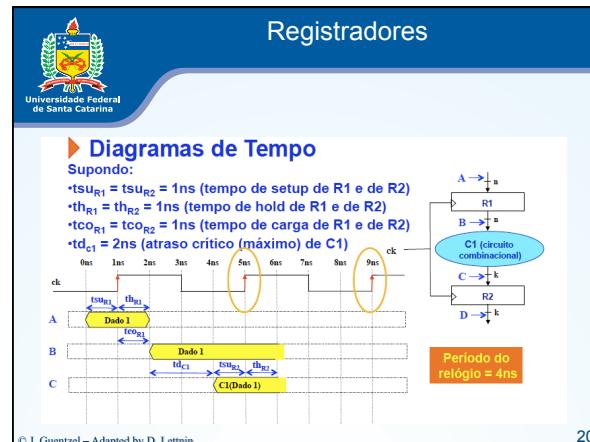
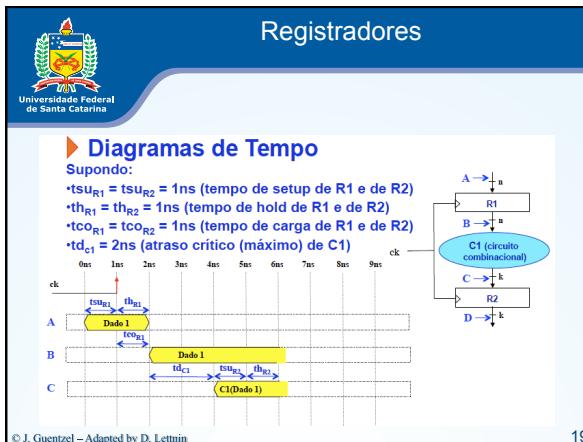
Diagramas de Tempo

Supondo:

- $tsu_{R1} = tsu_{R2} = 1\text{ns}$ (tempo de setup de R1 e de R2)
- $th_{R1} = th_{R2} = 1\text{ns}$ (tempo de hold de R1 e de R2)
- $tco_{R1} = tco_{R2} = 1\text{ns}$ (tempo de carga de R1 e de R2)
- $td_{c1} = 2\text{ns}$ (atraso crítico (máximo) de C1)

© J. Guentzel – Adapted by D. Lettin

18



Máquinas Sequenciais Síncronas

Modelo de Moore

Característica principal: as saídas dependem apenas do estado atual.

© J. Guentzel – Adapted by D. Lettmn

25

Máquinas Sequenciais Síncronas

Modelo de Mealy

Característica principal: as saídas dependem do estado atual e de entrada(s) primária(s).

© J. Guentzel – Adapted by D. Lettmn

26

Máquinas Sequenciais Síncronas

Síntese de Circuitos Sequenciais

Roteiro para a Síntese (=Projeto)

- Determinar quantos estados são necessários (e o nº de variáveis de estado)
- Construir o diagrama de estados, observando com cuidado o comportamento solicitado para a FSM e adotando um modelo de FSM (Moore ou Mealy):
 - Determinar as transições entre estados necessárias
 - Selecionar um estado para servir como estado inicial
- Construir a tabela de próximo estado e a tabela das saídas
- Sintetizar (projetar) os circuitos combinacionais: lógica de próximo estado e lógica de saída.

Vejamos um exemplo...

© J. Guentzel – Adapted by D. Lettmn

27

Máquinas de Moore

Síntese de Circuitos Sequenciais

Exemplo 3:

Projete um circuito que satisfaça às seguintes especificações:
 1.O circuito possui uma entrada, w , e uma saída, z .
 2.Todas as mudanças de valores no circuito ocorrem na borda de subida do sinal de relógio.
 3.Quando o circuito detectar que a entrada w vale "0", a saída z deve valer "0" no ciclo de relógio seguinte. Porém, quando o circuito detectar que a entrada w vale "1" durante duas bordas de relógio consecutivas, a saída z deve passar a valer "1" no ciclo de relógio seguinte à segunda ocorrência do valor "1". As mudanças de z estão sincronizadas com a borda de relógio ativa.

Exemplo de funcionamento:

© J. Guentzel – Adapted by D. Lettmn

28

Máquinas de Moore

Síntese de Circuitos Sequenciais

Exemplo 3: Diagrama de Estados

© J. Guentzel – Adapted by D. Lettmn

29

Máquinas de Moore

Síntese de Circuitos Sequenciais

Exemplo 3: Tabelas de (Transição de) Estados e de Saída

Tabela de (transição de) estados

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	C
C	0	A
C	1	C

Tabela de saída

Estado	z
A	0
B	0
C	1

© J. Guentzel – Adapted by D. Lettmn

30

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: Diagrama de Blocos

Sinais de entrada: w
Sinais de saída: z
Número de estados: 3
Logo, são necessários 2 flip-flops para armazenar as variáveis de estado

© J. Guentzel – Adapted by D. Letmin

31

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: Codificação de Estados

Supondo a seguinte codificação: A=00, B=01, C=10

Estado atual	w	Próximo estado
A	00	A
A	01	B
B	0	A
B	1	C
C	0	A
C	1	C
-	-	-
-	-	-

Estado atual	y ₁ y ₀	w	Próximo estado
A	00	0	00
A	01	1	01
B	01	0	00
B	01	1	10
C	10	0	00
C	10	1	10
-	11	0	XX
-	11	1	XX

© J. Guentzel – Adapted by D. Letmin

32

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: Projetando a lógica de próximo estado

y ₁ y ₀	w	Y ₁ Y ₀
00	0	00
00	1	01
01	0	00
01	1	10
10	0	00
10	1	10
11	0	XX
11	1	XX

Y₁

Y₀

© J. Guentzel – Adapted by D. Letmin

33

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: Projetando a lógica de saída

Estado	y ₁ y ₀	z
A	00	0
B	01	0
C	10	1
-	11	X

z

© J. Guentzel – Adapted by D. Letmin

34

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: circuito final

Y1 = w*y1 + w*y0 = w*(y1+y0)
Y0 = w*y1*y0
z = y1

© J. Guentzel – Adapted by D. Letmin

35

Máquinas de Moore

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3: Funcionamento dinâmico

© J. Guentzel – Adapted by D. Letmin

36

Plano de Aula

Universidade Federal de Santa Catarina

- Registradores
- Máquinas de Moore
- Máquinas de Mealy

37

Máquinas de Mealy

Universidade Federal de Santa Catarina

► **Modelo de Mealy**

Característica principal: as saídas dependem do estado atual e de entrada(s) primária(s)

© J. Guentzel – Adapted by D. Lettin

38

Máquinas de Mealy

Universidade Federal de Santa Catarina

► **Síntese de Circuitos Sequenciais: Modelo de Mealy**

Exemplo 3:

Projete um circuito que satisfaça às seguintes especificações:
 1.O circuito possui uma entrada, **w**, e uma saída, **z**.
 2.Todas as mudanças de valores no circuito ocorrem na borda de subida do sinal de relógio.
 3.Quando o circuito detectar que a entrada **w** vale “0”, a saída **z** deve valer “0” no ciclo de relógio seguinte. Porém, quando o circuito detectar que a entrada **w** vale “1” durante duas bordas de relógio consecutivas, a saída **z** deve passar a valer “1” no ciclo de relógio seguinte à segunda ocorrência do valor “1”. As mudanças de **z** estão sincronizadas com a borda de relógio ativa.

Considera a seguinte modificação da especificação acima:
 • O sinal de saída **z** não precisa esperar que um segundo valor igual a “1” seja amostrado da entrada **w**.
 • Porém, se **z** = 1 e **w** muda de “1” para “0”, **z** deve também mudar para “0”, independentemente da borda ativa do relógio

© J. Guentzel – Adapted by D. Lettin

39

Máquinas de Mealy

Universidade Federal de Santa Catarina

► **Síntese de Circuitos Sequenciais: Modelo de Mealy**

Exemplo 3.1: exemplo de funcionamento dinâmico

Especificação modificada:

- O sinal de saída **z** não precisa esperar que um segundo valor igual a “1” seja amostrado da entrada **w**.
- Porém, se **z** = 1 e **w** muda de “1” para “0”, **z** deve também mudar para “0”, independentemente da borda ativa do relógio

© J. Guentzel – Adapted by D. Lettin

40

Máquinas de Mealy

Universidade Federal de Santa Catarina

► **Síntese de Circuitos Sequenciais: Modelo de Mealy**

Exemplo 3.1: Diagrama de Estados

Interpretação do Diagrama de Estados:

- Durante o ciclo de relógio atual, o valor da saída **z** corresponde ao rótulo assinalado em alguma das arestas que partem do estado atual.
- No caso do estado B, por exemplo, **z** pode valer “0” ou valer “1”, conforme for o valor de **w**. Isto implica que **z** pode mudar de valor antes que a máquina de estados mude de estado.

41

Máquinas de Mealy

Universidade Federal de Santa Catarina

► **Síntese de Circuitos Sequenciais: Modelo de Mealy**

Exemplo 3.1: Tabelas de (Transição de) Estados e de Saída

Estado atual	w	Próximo estado
A	0	A
A	1	B
B	0	A
B	1	B

Estado	w	z
A	0	0
A	1	0
B	0	0
B	1	1

© J. Guentzel – Adapted by D. Lettin

42

Máquinas de Mealy

Síntese de Circuitos Sequenciais: Modelo de Mealy

Exemplo 3.1: Projetando a lógica de próximo estado e a lógica de saída

- Como há somente dois estados, iremos utilizar somente uma variável de estado.
- Assumindo o seguinte assinalamento de estados: A=0, B=1

Lógica de Próximo Estado			Lógica de Saída		
y	w	Y	y	w	z
A	0	0	0	A	
A	0	1	1	B	
B	1	0	0	A	
B	1	1	1	B	

$$Y = \bar{y} \cdot w + y \cdot w = w$$

$$z = y \cdot w$$

© J. Guentzel – Adapted by D. Lettmann 43

Máquinas de Mealy

Síntese de Circuitos Sequenciais: Modelo de Mealy

Exemplo 3.1: circuito final

$$Y = w$$

$$z = y \cdot w$$

© J. Guentzel – Adapted by D. Lettmann 44

Máquinas de Mealy

Síntese de Circuitos Sequenciais

Exemplos 3 e 3.1: Comparando Moore e Mealy

$$Y_1 = w \cdot y_0$$

$$Y_0 = w$$

$$z = y_1$$

$$Y = w$$

$$z = y \cdot w$$

© J. Guentzel – Adapted by D. Lettmann 45

Máquinas de Mealy

Síntese de Circuitos Sequenciais: Modelo de Mealy

Exemplo 3.1: exemplo de funcionamento dinâmico

© J. Guentzel – Adapted by D. Lettmann 46

Máquinas de Mealy

Síntese de Circuitos Sequenciais

Exemplos 3 e 3.1:

Mealy

Moore

© J. Guentzel – Adapted by D. Lettmann 47

Máquinas de Mealy

Síntese de Circuitos Sequenciais

Exemplo 3.1:

Porém, se passarmos a saída z por um segundo flip-flop, filtraremos o comportamento assíncrono. De fato, estaremos transformando o circuito para o Modelo de Moore...

© J. Guentzel – Adapted by D. Lettmann 48

Máquinas de Mealy

Universidade Federal de Santa Catarina

► Síntese de Circuitos Sequenciais

Exemplo 3.1:

Mealy

Mealy transform. em Moore

© J. Guentzel – Adapted by D. Lettnin

49

Máquinas de Mealy

Universidade Federal de Santa Catarina

Mealy transform. em Moore

Moore

© J. Guentzel – Adapted by D. Lettnin

50

EEL7020 – Sistemas Digitais

Aula 9: Circuitos sequenciais

Universidade Federal de Santa Catarina

Prof. Djones Vinicius Lettnin
lettnin@eel.ufsc.br
<http://dlettnin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the author J. Guentzel.