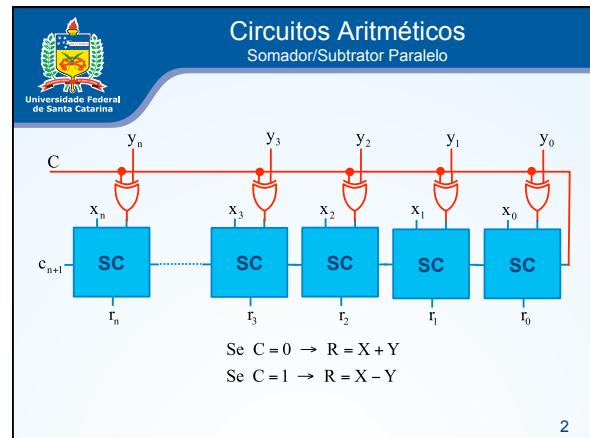



Universidade Federal
de Santa Catarina

EEL7020 – Sistemas Digitais
Aula 7: Circuitos sequenciais -
Flip-flops

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

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the author E. Batista, F. Vahid.




Universidade Federal
de Santa Catarina

Plano de Aula

- Flip-flops



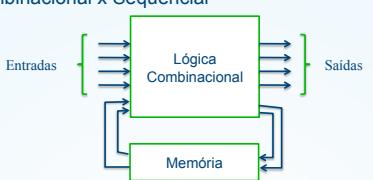
3




Universidade Federal
de Santa Catarina

Lógica Seqüencial

- Combinacional x Seqüencial



© E. Batista – Adapted by D. Lettnin

5



Example Needing Bit Storage

Universidade Federal de Santa Catarina

- Flight attendant call button
 - Press call: light turns on
 - Stays on** after button released
 - Press cancel: light turns off
 - Logic gate circuit to implement this?

Call → NOR gate → Q
Cancel → NOR gate → Q

Doesn't work. Q=1 when Call=1, but doesn't stay 1 when Call returns to 0
Need some form of "feedback" in the circuit

7

© F. Vahid – Adapted by D. Lettmann

First attempt at Bit Storage

Universidade Federal de Santa Catarina

- We need some sort of feedback
 - Does circuit on the right do what we want?
 - No: Once Q becomes 1 (when S=1), Q stays 1 forever – no value of S can bring Q back to 0

S 1 0 1 0 1 0 1 0
t 1 0 1 0 1 0 1 0
Q 1 1 1 1 1 1 1 1

8

© F. Vahid – Adapted by D. Lettmann

Bit Storage Using an SR Latch

Universidade Federal de Santa Catarina

- Does the circuit to the right, with cross-coupled NOR gates, do what we want?
 - Yes! How did someone come up with that circuit? Maybe just trial and error, a bit of insight...

S 1 0 1 0 1 0 1 0
R 1 0 1 0 1 0 1 0
t 1 0 1 0 1 0 1 0
Q 1 1 1 1 1 1 1 1

9

© F. Vahid – Adapted by D. Lettmann

Example Using SR Latch for Bit Storage

Universidade Federal de Santa Catarina

- SR latch can serve as bit storage in previous example of flight-attendant call button
 - Call=1 : sets Q to 1
 - Q stays 1 even after Call=0
 - Cancel=1 : resets Q to 0
- But, there's a problem...

Call button → S
Cancel button → R
Bit Storage → Blue light

10

© F. Vahid – Adapted by D. Lettmann

Problem with SR Latch

Universidade Federal de Santa Catarina

- Problem
 - If S=1 and R=1 simultaneously, we don't know what value Q will take

S 1 1 1 1 1 1 1 1
R 1 1 1 1 1 1 1 1
t 1 0 1 0 1 0 1 0
Q 1 0 1 0 1 0 1 0

Q may oscillate. Then, because one path will be slightly longer than the other, Q will eventually settle to 1 or 0 – but we don't know which.

11

© F. Vahid – Adapted by D. Lettmann

Problem with SR Latch

Universidade Federal de Santa Catarina

- Problem not just one of a user pressing two buttons at same time
- Can also occur even if SR inputs come from a circuit that supposedly never sets S=1 and R=1 at same time
 - But does, due to different delays of different paths

X 1 0 1 0 1 0 1 0
Y 1 0 1 0 1 0 1 0
S 1 0 1 0 1 0 1 0
R 1 0 1 0 1 0 1 0
SR = 11

The longer path from X to R than to S causes SR=11 for short time – could be long enough to cause oscillation

12

© F. Vahid – Adapted by D. Lettmann

Solution: Level-Sensitive SR Latch

Universidade Federal de Santa Catarina

- Add enable input "C" as shown
 - Only let S and R change when C=0
 - Ensure circuit in front of SR never sets SR=11, except briefly due to path delays
 - Change C to 1 only after sufficient time for S and R to be stable
 - When C becomes 1, the stable S and R value passes through the two AND gates to the SR latch's S1 R1 inputs.

...S1R1 never = 11

© F. Vahid – Adapted by D. Lettmann

Clock Signals for a Latch

Universidade Federal de Santa Catarina

- How do we know when it's safe to set C=1?
 - Most common solution – make C pulse up/down
 - C=0: Safe to change X, Y
 - C=1: Must not change X, Y
 - We'll see how to ensure that later
 - Clock** signal – Pulsing signal used to enable latches
 - Because it ticks like a clock
 - Sequential circuit whose storage components all use clock signals: **synchronous** circuit
 - Most common type
 - Asynchronous circuits – important topic, but left for advanced course

© F. Vahid – Adapted by D. Lettmann

Clocks

Universidade Federal de Santa Catarina

Freq	Period
100 GHz	0.01 ns
10 GHz	0.1 ns
1 GHz	1 ns
100 MHz	10 ns
10 MHz	100 ns

- Clock period:** time interval between pulses
 - Above signal: period = 20 ns
- Clock cycle:** one such time interval
 - Above signal shows 3.5 clock cycles
- Clock frequency:** 1/period
 - Above signal: frequency = 1 / 20 ns = 50 MHz
 - 1 Hz = 1/s

© F. Vahid – Adapted by D. Lettmann

Level-Sensitive D Latch

Universidade Federal de Santa Catarina

- SR latch requires careful design to ensure SR=11 never occurs
- D latch relieves designer of that burden
 - Inserted inverter ensures R always opposite of S

© F. Vahid – Adapted by D. Lettmann

Problem with Level-Sensitive D Latch

Universidade Federal de Santa Catarina

- D latch still has problem (as does SR latch)
 - When C=1, through how many latches will a signal travel?
 - Depends on for how long C=1
 - Clk_A – signal may travel through multiple latches
 - Clk_B – signal may travel through fewer latches
 - Hard to pick C that is just the right length
 - Can we design bit storage that only stores a value on the rising edge of a clock signal?

© F. Vahid – Adapted by D. Lettmann

D Flip-Flop

Universidade Federal de Santa Catarina

- 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

© F. Vahid – Adapted by D. Lettmann

D Flip-Flop

The triangle means clock input, edge triggered

Symbol for rising-edge triggered D flip-flop

rising edges
clk

Symbol for falling-edge triggered D flip-flop

falling edges
clk

Internal design: Just invert servant clock rather than master

© F. Vahid – Adapted by D. Lettmann

19

D Flip-Flop

- Solves problem of not knowing through how many latches a signal travels when C=1
 - In figure below, signal travels through exactly one flip-flop, for Clk_A or Clk_B
 - Why? Because on rising edge of Clk, all four flip-flops are loaded simultaneously -- then all four no longer pay attention to their input, until the next rising edge. Doesn't matter how long Clk is 1.

Two latches inside each flip-flop

Y-D1-Q1-D2-Q2-D3-Q3-D4-Q4

Clk Clk_A Clk_B

© F. Vahid – Adapted by D. Lettmann

20

D Latch vs. D Flip-Flop

- Latch is level-sensitive: Stores D when C=1
- Flip-flop is edge triggered: Stores D when C changes from 0 to 1
 - Saying "level-sensitive latch," or "edge-triggered flip-flop," is redundant
 - Two types of flip-flops -- rising or falling edge triggered.
- Comparing behavior of latch and flip-flop:

© F. Vahid – Adapted by D. Lettmann

21

Bit Storage Summary

Feature: S+1 sets Q to 1, R+1 resets Q to 0. Problem: SR=11 undefined Q.	Feature: S and R only have effect when C=1. We can have a race condition so SR=11 never happens when C=1. Problem: avoiding SR=11 can be a burden.	D is stable before and after C=1, and can be 11 for one bit glitch even if C changes while C=1. Problem: C=1 too long propagates new values through too many latches too short may not enable a store.	Feature: SR can't be 11 if present at rising clock edge, so value can propagate to other flip-flops during same clock cycle. Tradeoff: uses more gates internally than D latch, and requires more external gates than SR – but gate count is less of an issue today.

- We considered increasingly better bit storage until we arrived at the robust D flip-flop bit storage

© F. Vahid – Adapted by D. Lettmann

Lógica Seqüencial
Latches e Flip-flops

- Circuitos seqüenciais elementares
- Têm capacidade de armazenar informação
- Unidade elementar de memória => 1 bit
- Latch** – assíncrono
- Flip-flop** – versão síncrona do **latch**

© E. Batista – Adapted by D. Lettmann

23

Lógica Seqüencial
Latch RS (Reset-Set)

- Latch com portas NAND:**

S(t) Q(t)
R(t) Q-bar(t)

© E. Batista – Adapted by D. Lettmann

24

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

$Q(t)$ Complementares
 $S(t)$ Entradas atuais "Excitação"
 $R(t)$ "Exclusão"
 $Q(t) \rightarrow$ Saída atual "estado"
 $Q(t+1) \rightarrow$ Próximo estado

25

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

A) Tabela de Transição:

$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$	$\bar{Q}(t+1)$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

26

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

A) Tabela de Transição:

$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$	$\bar{Q}(t+1)$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

27

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

A) Tabela de Transição:

$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$	$\bar{Q}(t+1)$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

28

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

B) Tabela de Função:

$S(t)$	$R(t)$	$Q(t+1)$
0	0	
0	1	
1	0	
1	1	

29

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:**

$S(t)$ $Q(t)$
 $R(t)$ $\bar{Q}(t)$

B) Tabela de Função:

$S(t)$	$R(t)$	$Q(t+1)$
0	0	Proibido
0	1	1 (Set)
1	0	0 (Reset)
1	1	

30

© E. Batista – Adapted by D. Lettmann

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:

B) Tabela de Função:

S(t)	R(t)	Q(t+1)
0	0	Proibido
0	1	1 (Set)
1	0	0 (Reset)
1	1	Q(t)

© E. Batista – Adapted by D. Lettmann

31

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:

C) Tabela de Excitação:

Q(t) → Q(t+1)	S(t)	R(t)
0	0	X
0	1	1
1	0	1
1	1	1

© E. Batista – Adapted by D. Lettmann

32

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NAND:

C) Tabela de Excitação:

Q(t) → Q(t+1)	S(t)	R(t)
0	1	X
0	0	1
1	0	0
1	X	1

© E. Batista – Adapted by D. Lettmann

33

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NOR:

© E. Batista – Adapted by D. Lettmann

34

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NOR:

Table 1:

S(t)	R(t)	Q(t+1)
0	0	
0	1	
1	0	
1	1	

© E. Batista – Adapted by D. Lettmann

35

Lógica Seqüencial
Latch RS (Reset-Set)

Universidade Federal de Santa Catarina

- Latch com portas NOR:

Table 2:

S(t)	R(t)	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Proibido

© E. Batista – Adapted by D. Lettmann

36

Lógica Seqüencial
Clock e Preset/Clear

37

Lógica Seqüencial
Clock e Preset/Clear

- Preset/Clear:**

© E. Batista – Adapted by D. Lettmin

38

Lógica Seqüencial
Clock e Preset/Clear

- Preset/Clear:**

© E. Batista – Adapted by D. Lettmin

39

Lógica Seqüencial
Clock e Preset/Clear

- Preset/Clear:**

PRESET	CLEAR	$Q(t+1)$
0	0	Proibido
0	1	1
1	0	0
1	1	$Q(t)$

© E. Batista – Adapted by D. Lettmin

40

Lógica Seqüencial
Clock e Preset/Clear

- Clock:**

41

Lógica Seqüencial
Clock e Preset/Clear

- Clock:**

© E. Batista – Adapted by D. Lettmin

42

Lógica Seqüencial
Clock e Preset/Clear

• Clock:
S(t)

CLOCK Detec. Borda

R(t)

CLEAR

PRESET

Q(t) $\bar{Q}(t)$

– Com isso, o circuito passa a responder de forma síncrona (na cadência de um relógio): **Flip-Flop**

© E. Batista – Adapted by D. Lettmann

43

Lógica Seqüencial
Clock e Preset/Clear

• Clock:

CLK S(t) R(t) Q(t+1)

CLK	S(t)	R(t)	Q(t+1)
↑	0	0	Q(t)
↑	0	1	0
↑	1	0	1
↑	1	1	Proibido

© E. Batista – Adapted by D. Lettmann

44

Lógica Seqüencial
Flip-Flop D (Data/Delay)

© E. Batista – Adapted by D. Lettmann

45

Lógica Seqüencial
Flip-Flop D (Data/Delay)

D FF-D Q

CLK

© E. Batista – Adapted by D. Lettmann

46

Lógica Seqüencial
Flip-Flop D (Data/Delay)

D FF-D Q

CLK

Função

D(t)	Q(t+1)
0	0
1	1

CLK	D(t)	Q(t+1)
0	X	Q(t)
↑	D(t)	D(t)
1	X	Q(t)

© E. Batista – Adapted by D. Lettmann

47

Lógica Seqüencial
Flip-Flop D (Data/Delay)

D FF-D Q

CLK

Tabela de Excitação:

Q(t) \rightarrow Q(t+1)	D(t)
0	0
0	1
1	0
1	1

© E. Batista – Adapted by D. Lettmann

48

Lógica Seqüencial
Flip-Flop D (Data/Delay)

Universidade Federal de Santa Catarina

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$D(t)$
0	0
0	1
1	0
1	1

© E. Batista – Adapted by D. Lettmann

49

Lógica Seqüencial
Flip-Flop D (Data/Delay)

Universidade Federal de Santa Catarina

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$D(t)$
0	0
0	1
1	0
1	1

Eq. de Transição:

$$Q(t+1) = D(t)$$

© E. Batista – Adapted by D. Lettmann

50

5. Lógica Seqüencial
5.1.3. Flip-Flop D (Data/Delay)

Universidade Federal de Santa Catarina

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$D(t)$
0	0
0	1
1	0
1	1

Eq. de Transição:

$$Q(t+1) = D(t)$$

© E. Batista – Adapted by D. Lettmann

51

Lógica Seqüencial
Flip-Flop T (Toggle – Comutação)

Universidade Federal de Santa Catarina

© E. Batista – Adapted by D. Lettmann

52

Lógica Seqüencial
Flip-Flop T (Toggle – Comutação)

Universidade Federal de Santa Catarina

© E. Batista – Adapted by D. Lettmann

53

Lógica Seqüencial
Flip-Flop T (Toggle – Comutação)

Universidade Federal de Santa Catarina

Função

$T(t)$	$Q(t+1)$
0	$\underline{Q}(t)$
1	$\bar{Q}(t)$

CLK	$T(t)$	$Q(t+1)$
0	X	$\underline{Q}(t)$
↑	0	$\bar{Q}(t)$
↑	1	$\underline{Q}(t)$

© E. Batista – Adapted by D. Lettmann

54

Lógica Seqüencial
Flip-Flop T (Toggle – Comutação)

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$T(t)$
0	0
0	1
1	0
1	1

© E. Batista – Adapted by D. Lettmann

55

Lógica Seqüencial
Flip-Flop T (Toggle – Comutação)

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$T(t)$	
0	0	0
0	1	1
1	0	1
1	1	0

© E. Batista – Adapted by D. Lettmann

56

Lógica Seqüencial
Flip-Flop T (Toggle)

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$T(t)$	
0	0	0
0	1	1
1	0	1
1	1	0

Eq. de Transição:

$$Q(t+1) = \bar{T}(t)Q(t) + T(t)\bar{Q}(t)$$

© E. Batista – Adapted by D. Lettmann

57

Lógica Seqüencial
Flip-Flop T (Toggle)

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$T(t)$	
0	0	0
0	1	1
1	0	1
1	1	0

Eq. de Transição:

$$Q(t+1) = \bar{T}(t)Q(t) + T(t)\bar{Q}(t)$$

© E. Batista – Adapted by D. Lettmann

58

Lógica Seqüencial
Flip-Flop JK

© E. Batista – Adapted by D. Lettmann

59

Lógica Seqüencial
Flip-Flop JK

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

© E. Batista – Adapted by D. Lettmann

60

**Lógica Seqüencial
Flip-Flop JK**

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

Eq. de transição:

$$Q(t+1) = \bar{Q}(t)J(t) + Q(t)\bar{K}(t)$$

© E. Batista – Adapted by D. Lettnin

61

**Lógica Seqüencial
Flip-Flop JK**

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

Eq. de transição:

$$Q(t+1) = \bar{Q}(t)J(t) + Q(t)\bar{K}(t)$$

© E. Batista – Adapted by D. Lettnin

62

**Lógica Seqüencial
Flip-Flop JK**

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$J(t)$	$K(t)$
0	0	
0	1	
1	0	
1	1	

© E. Batista – Adapted by D. Lettnin

63

**Lógica Seqüencial
Flip-Flop JK**

$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

Tabela de Excitação:

$Q(t) \rightarrow Q(t+1)$	$J(t)$	$K(t)$
0	0	0
0	1	1
1	0	X
1	1	0

© E. Batista – Adapted by D. Lettnin

64

**EEL7020 – Sistemas Digitais
Aula 7: Circuitos sequenciais -
Flip-flops**

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 authors E. Batista, F. Vahid.