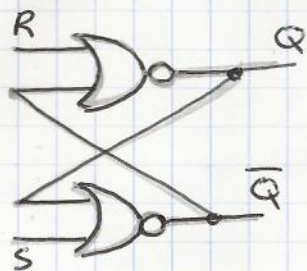


Aula 11

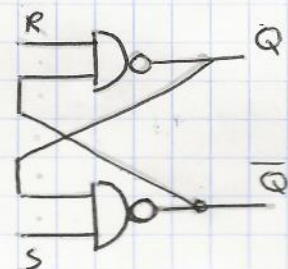
CSB

Latches e Flip-Flops

Latch RS - Reset / Set

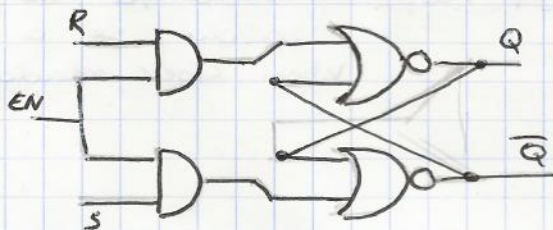


S	R	Q_{m+1}	$\overline{Q_{m+1}}$	
0	0	Q_m	$\overline{Q_m}$	HOLD
0	1	0	1	RESET
1	0	1	0	SET
1	1	U	U	—

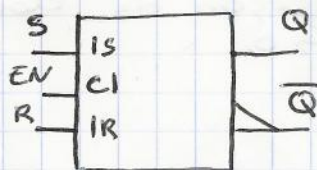


S	R	Q_{m+1}	$\overline{Q_{m+1}}$	
0	0	U	U	—
0	1	0	1	RESET
1	0	1	0	SET
1	1	Q_m	$\overline{Q_m}$	HOLD

Latch RS sincronizado



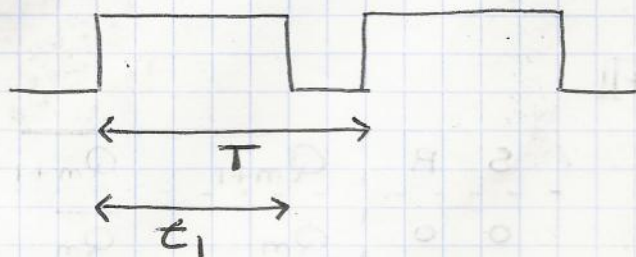
EN	R	S	Q_{m+1}	
0	Φ	Φ	Q_m	HOLD
1	0	0	Q_m	HOLD
1	0	1	0	SET
1	1	0	1	RESET
1	1	1	U	—



Set e Reset dependentes do valor da clock

Circuitos Síncronos

- Output com períodos HIGH e LOW bem definidos

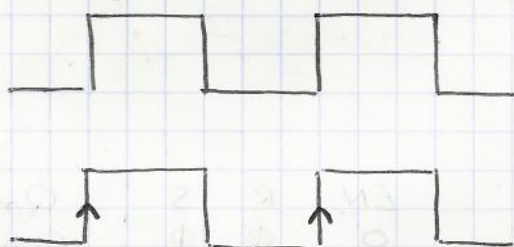


T - período

t_1 - Duty Cycle

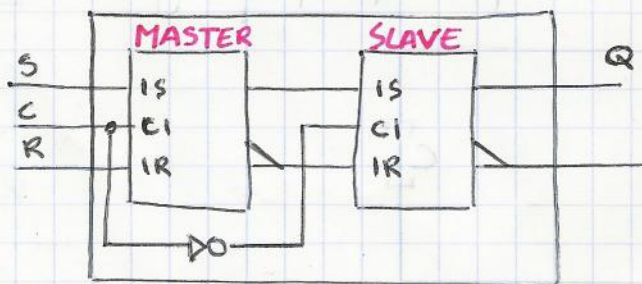
Flip-Flops vs. Latches

Flip-Flop Master-Slave



Latch → Mudança de saída se clock = 1

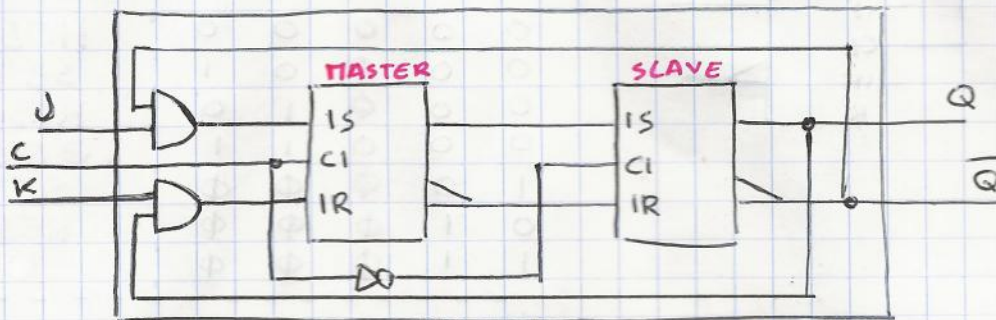
Flip-Flop → Mudança de saída se o nível clock mudar



O output só muda na transição de clock 1 → 0

O Latch Master, quando $C=1$, aceita ordens SET e RESET, mas só são transmitidas ao Latch Slave quando $C=0$

Flip-Flop JK Master-Slave



J	K	Q_{m+1}
0	0	Q_m
0	1	0
1	0	1
1	1	$\bar{Q}_m$

HOLD
RESET
SET
TOGGLE ↗

Resolução do estado indefinido

Flip-Flop Master-Slave Pulse-Triggered

Vs:

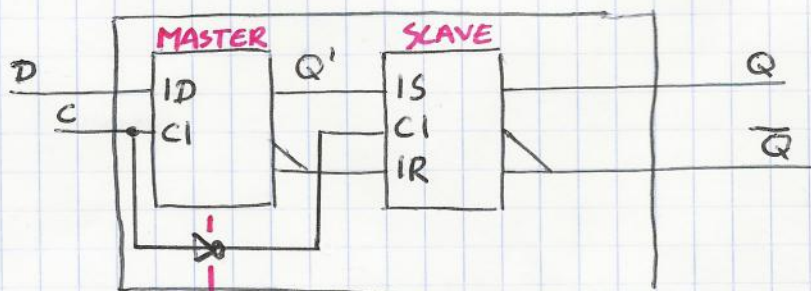
Flip-Flop Master-Slave Edge-Triggered

• Pulse Triggered → Exemplos anteriores

→ Sensíveis a qualquer pulse change, mesmo que accidental

• Edge Triggered

Rising Edge Delay



D	CLK	Q'_{+1}
0	↑	0
1	↑	1

MASTER TIPO D

O estado passado ao **SLAVE RS** é definido pelas transições de clock

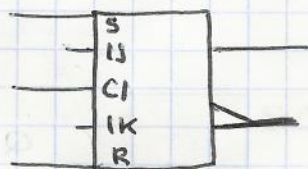
+ ET

- ET

sensível à transição $0 \rightarrow 1$

sensível à transição $1 \rightarrow 0$

Entradas Assíncronas

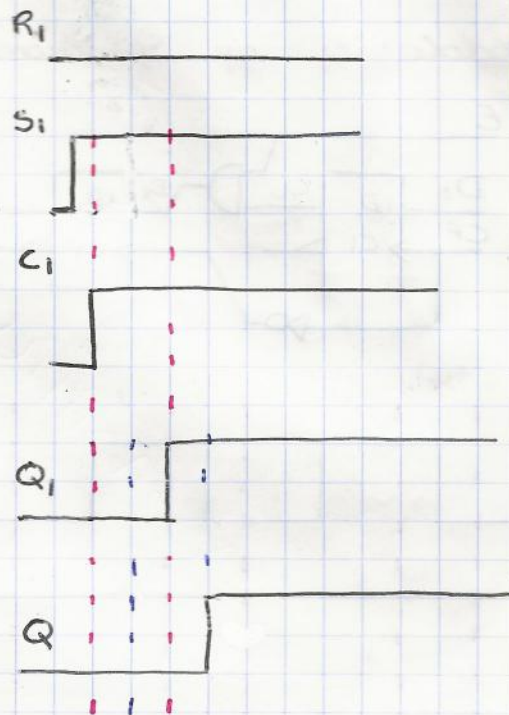
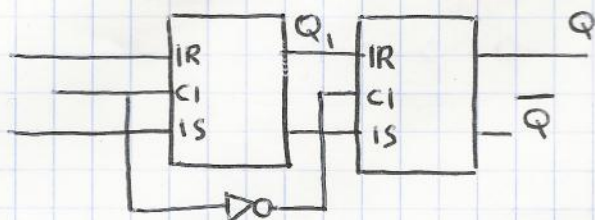


S	R	C	J	K	Q _{n+1}
0	0	1	0	0	HOLD
0	0	1	0	1	SET
0	0	1	1	0	RESET
0	0	1	1	1	TOGGLE
1	0	Φ	Φ	Φ	SET
0	1	Φ	Φ	Φ	RESET
1	1	Φ	Φ	Φ	U

Aula 12

CSB - Caracterização Temporal

Tempo de propagação



O tempo de propagação é então o deslocamento temporal (Δt) entre a transição do clock e a mudança do valor do output.

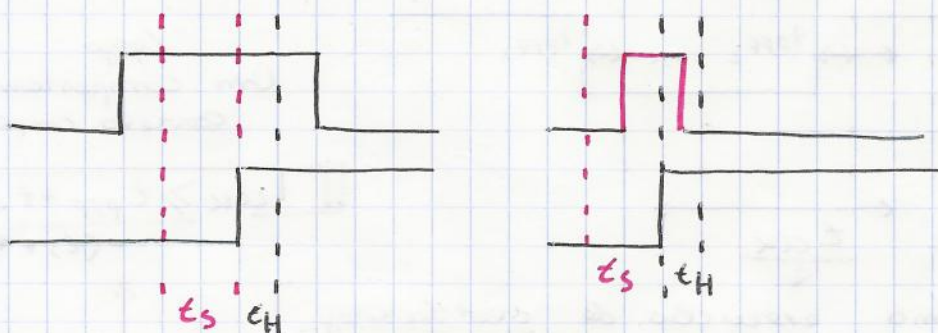
Desfasamento entre transições

Tempos de preparação e manutenção

Definem o tempo mínimo de espera, durante o qual as entradas não podem variar, em função do clock

• Tempo **SETUP** → antes da transição do clock

• Tempo **HOLD** → depois da transição do clock

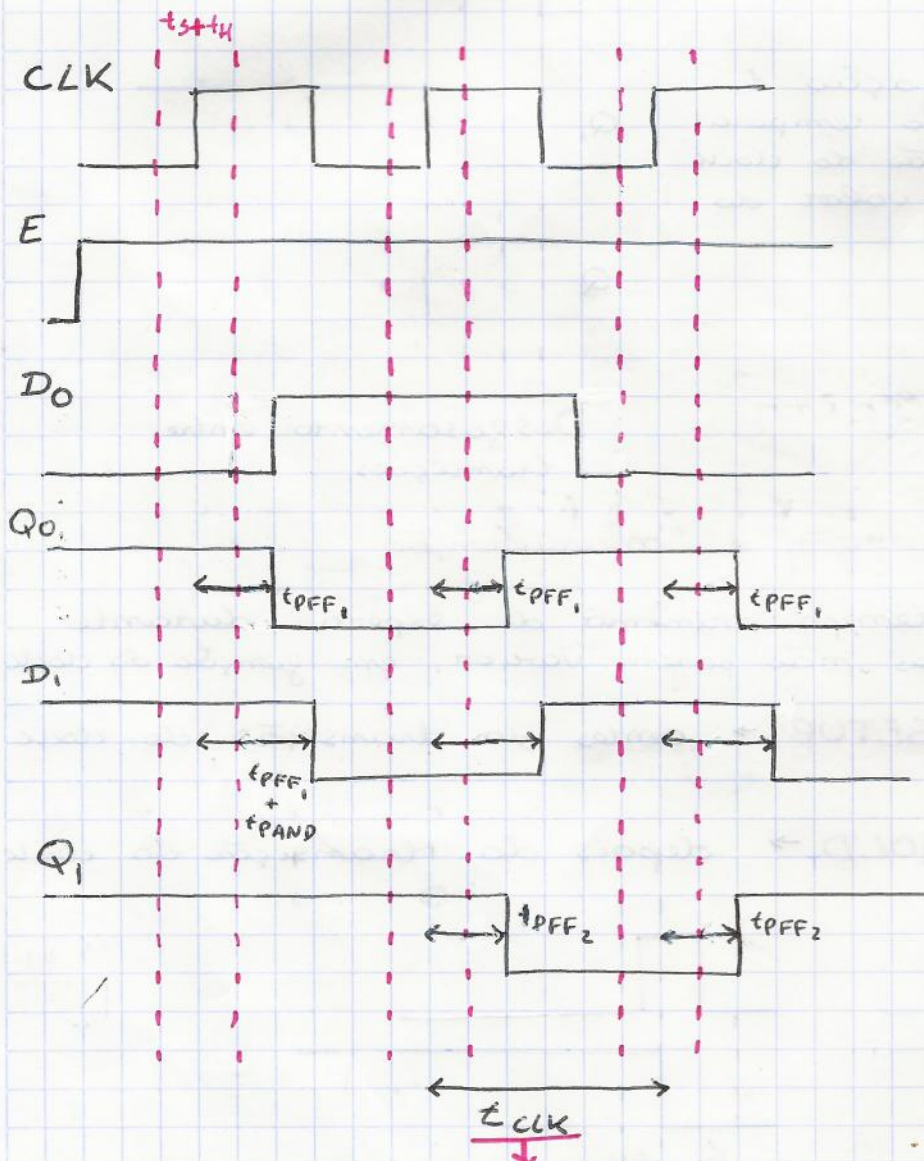
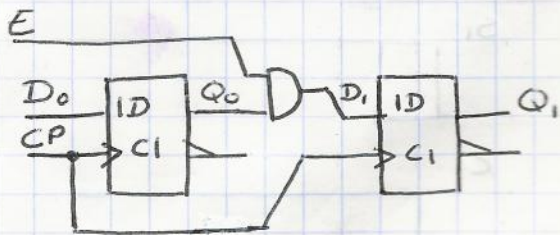


Garantia de funcionamento
correto

Violação dos tempos de
setup e de hold

⚠ - Nos flip-flop pulse triggered os tempos de espera tem que ser inviolados em todo o domínio, contudo, como é expectável, nos flip-flop edge triggered os tempos apenas tem que ser inviolados no fianco da sensibilidade

Metodologia de Sincronização Temporal



Logo
Um comportamento
correto implica

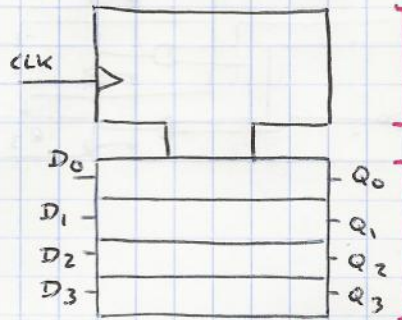
$$T_{CLK} \geq t_{PFF} + t_{\text{c\u00f3gica}} - t_s$$

Numa execução de overclocking, o per\u00edodo do rel\u00f3gio \u00e9 diminuido o que pode levar a altera\u00e7\u00f5es do valor das entradas na banda do t_s levando a poss\u00edveis comportamentos err\u00e1ticos do c\u00edrcuito

Aula 13 CSB Registros

Registros

Agregados
de Flip-Flops

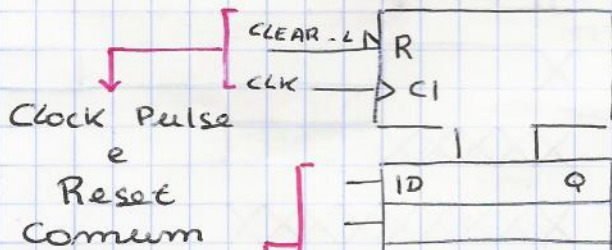


Comum a
todo o registros

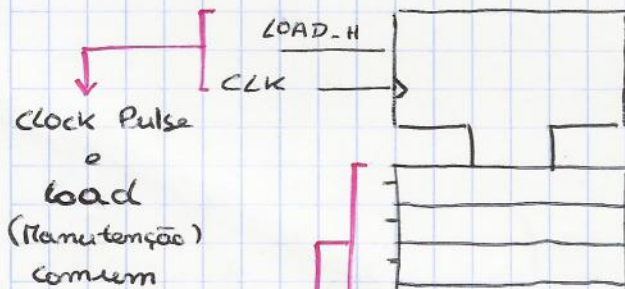
Flip-Flops
Constituintes

Registros de carregamento paralelo

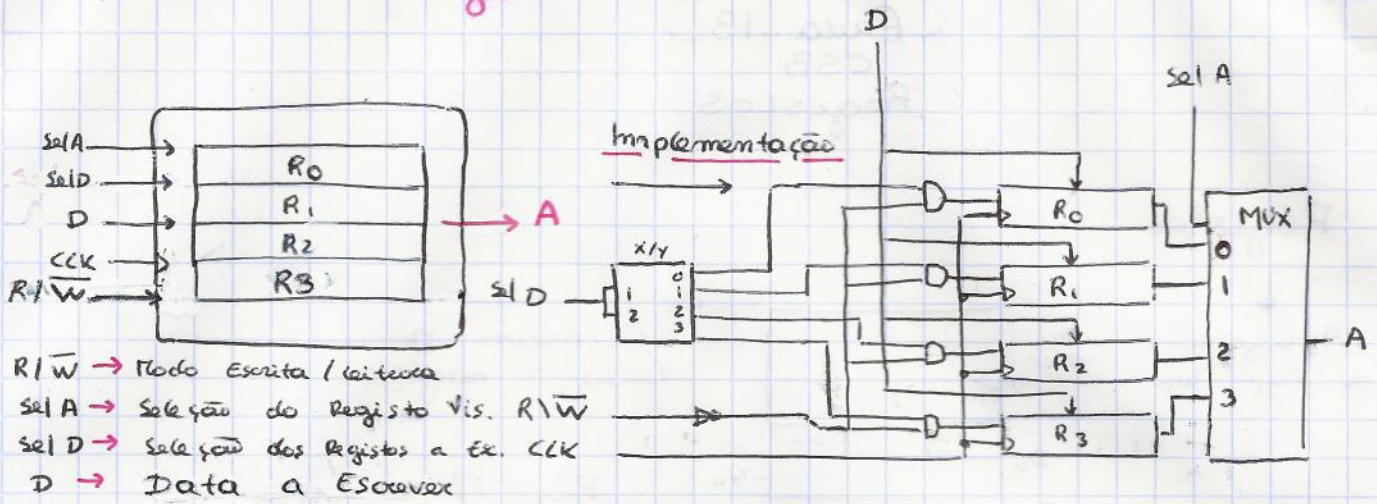
→ Exemplo base do que
é um registro



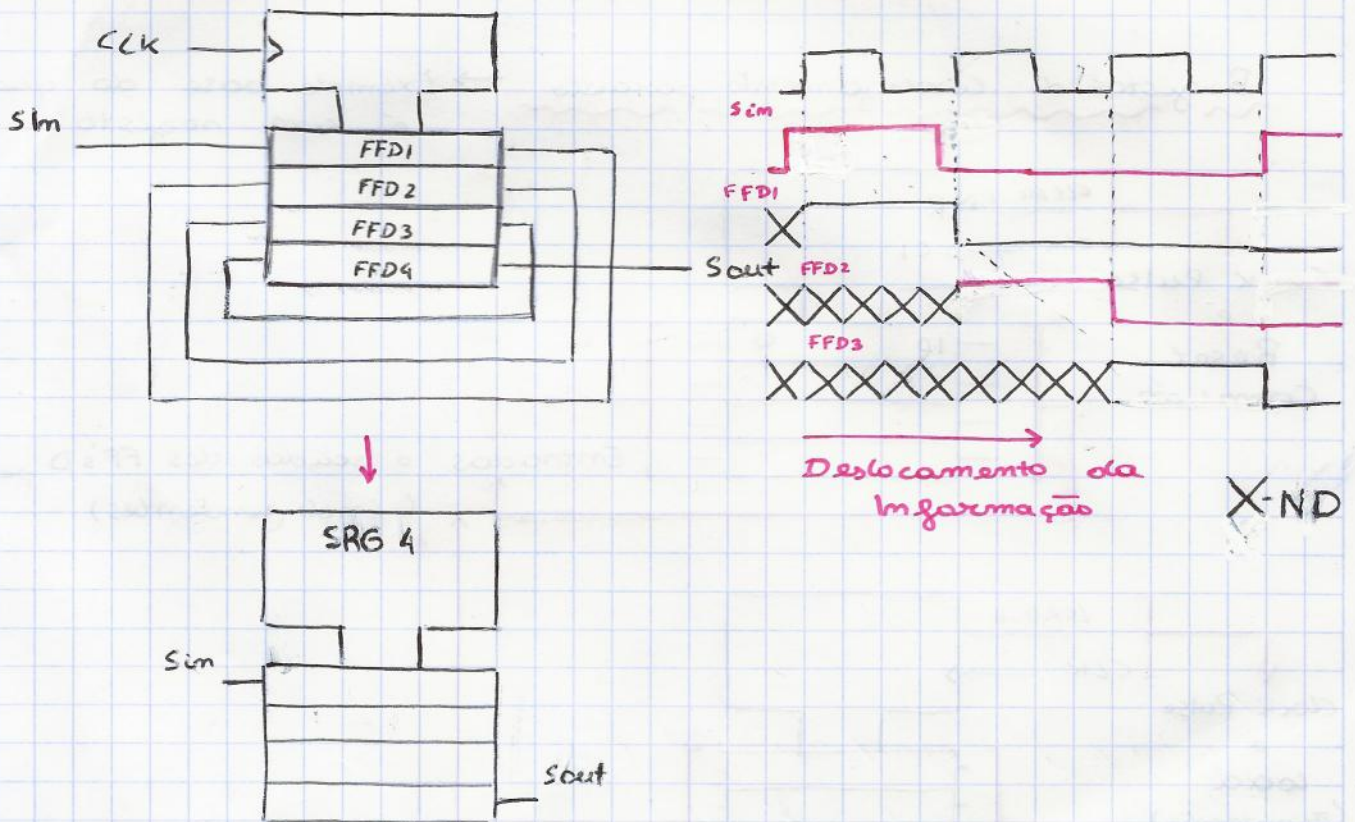
Entradas e saídas dos FF's D paralelas
(independentes)



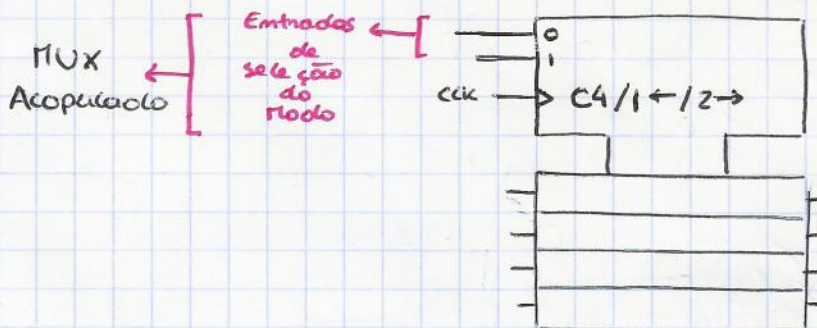
Banco de Registos



Registos de deslocamento



Registos de deslocamento Multimodo



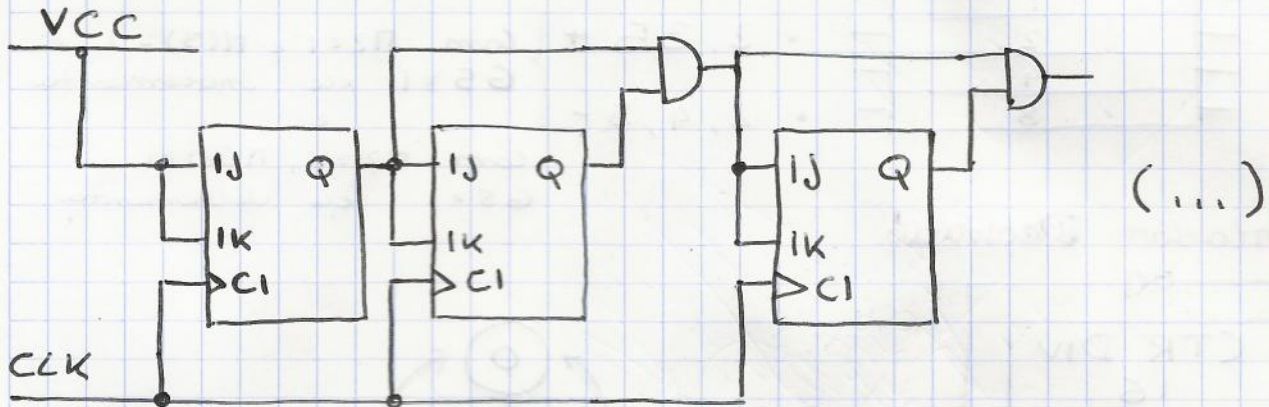
S_1	S_0	Modo
0	0	HOLD
0	1	Shift-left
1	0	Shift-Right
1	1	LOAD

Aula 14

CSB - Contadores

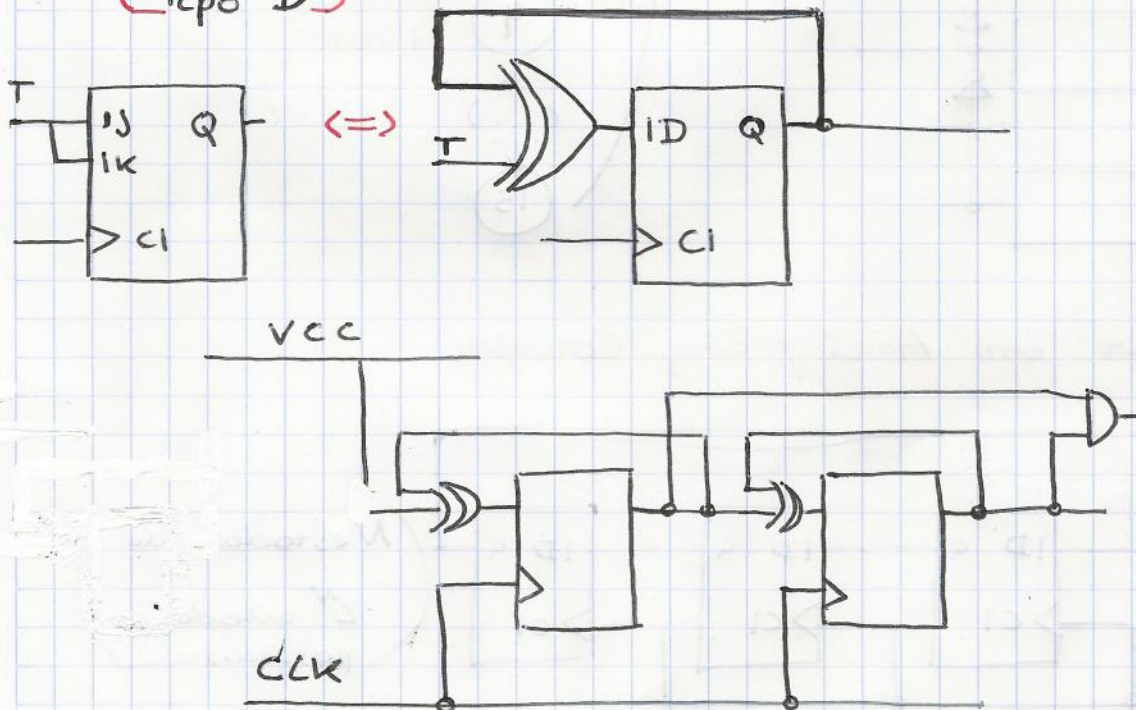
Contador Binário

Tipo JK

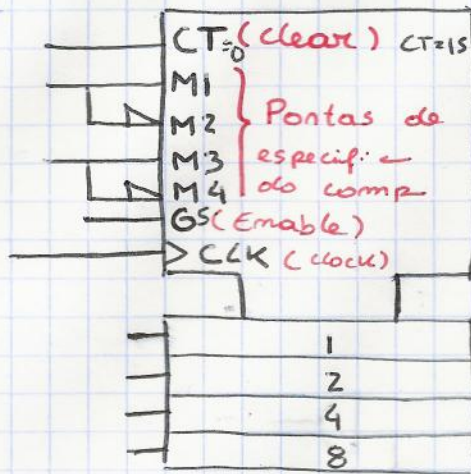


$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{t_{pFF} + (n-2)t_{pAND} + t_{sFF}}$$

Tipo D



Componente



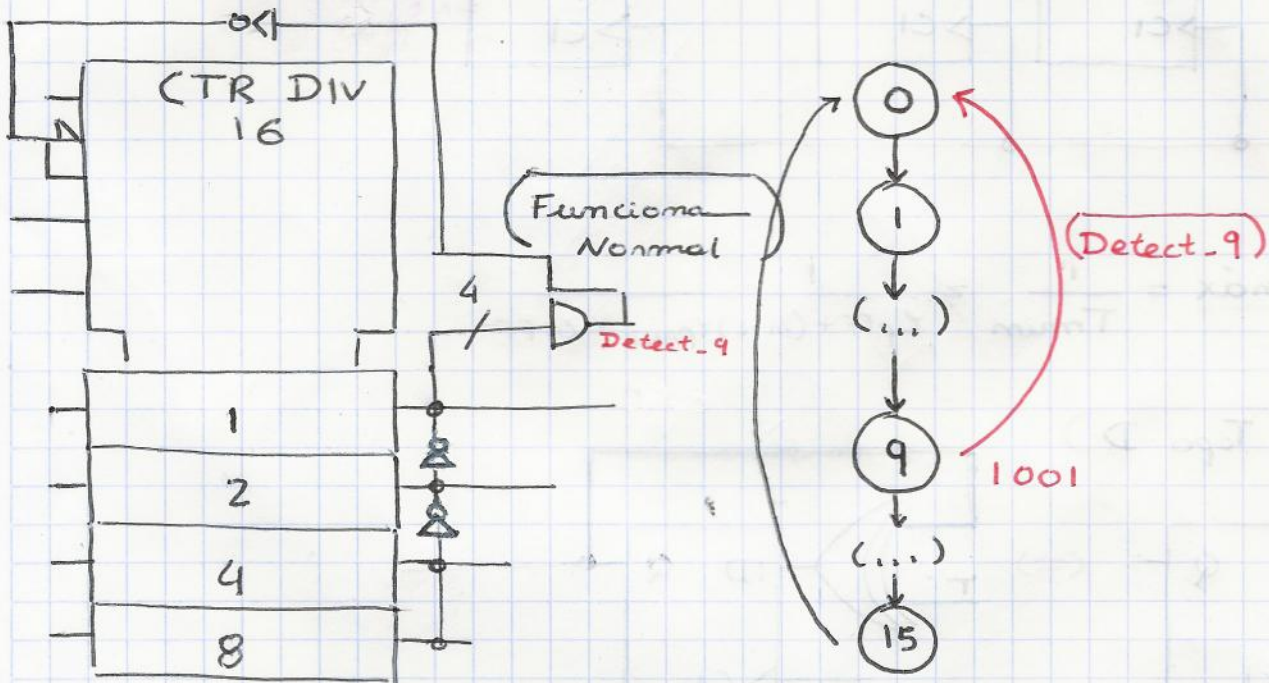
Se o Componente é do tipo $\rightarrow$ Aparece à frente do CLK

• 2, 5 + $\rightarrow$ Com M_2 (count) ≥ 1 e $GS \geq 1$ ele incrementa

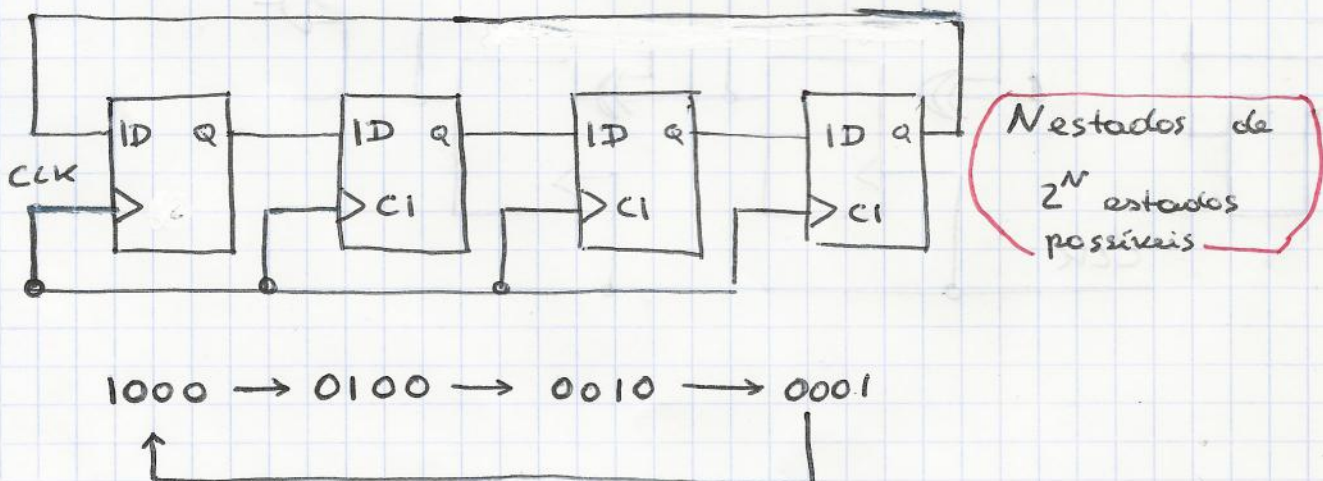
• 2, 3, 5 + } Com $M_2 \geq 1$ e $M_3 \geq 1$ e $GS \geq 1$ ele incrementa e

• 2, 4, 5 - } Com $M_2 \geq 1$, $M_4 \geq 1$ e $GS \geq 1$ ele decrementa

Contador Decimal

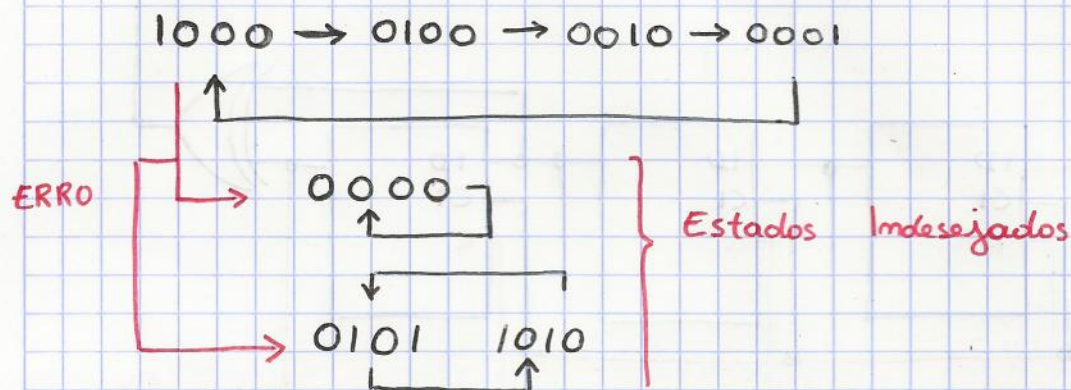


Contadores em Anel - Ring Counter

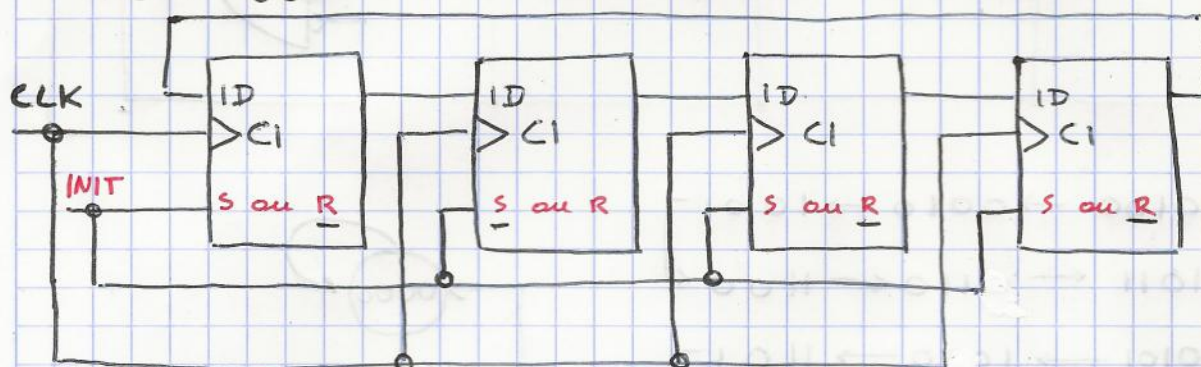


Nestados de 2^N estados possíveis

Contadores "LOCK-OUT"



SOLUÇÃO: → CIRCUITO WATCHDOG

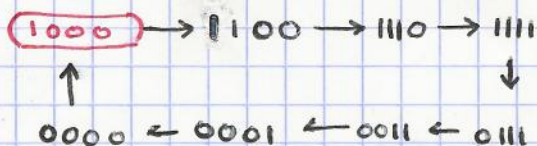
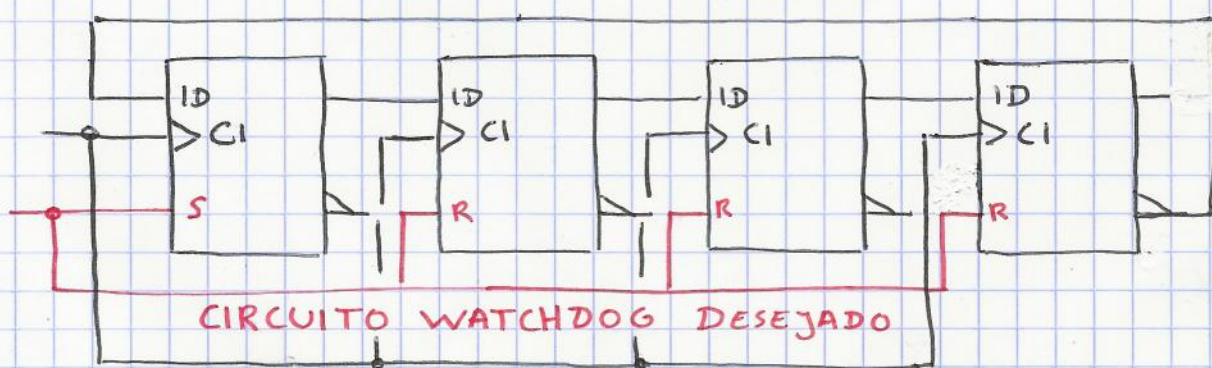


INIT → Detecta estados indesejados

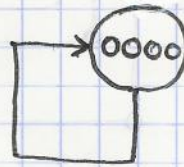
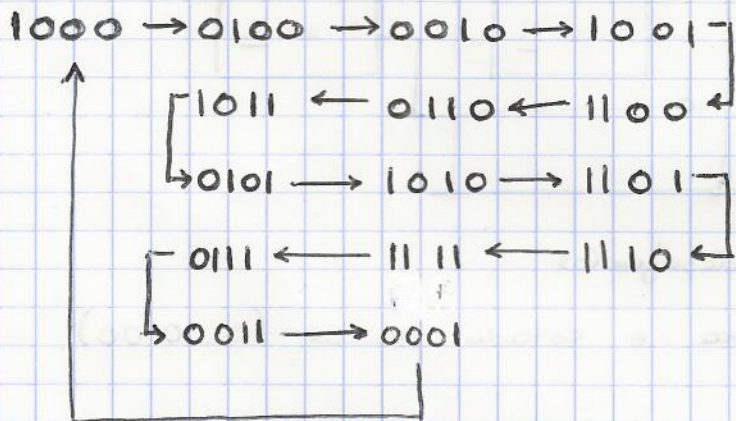
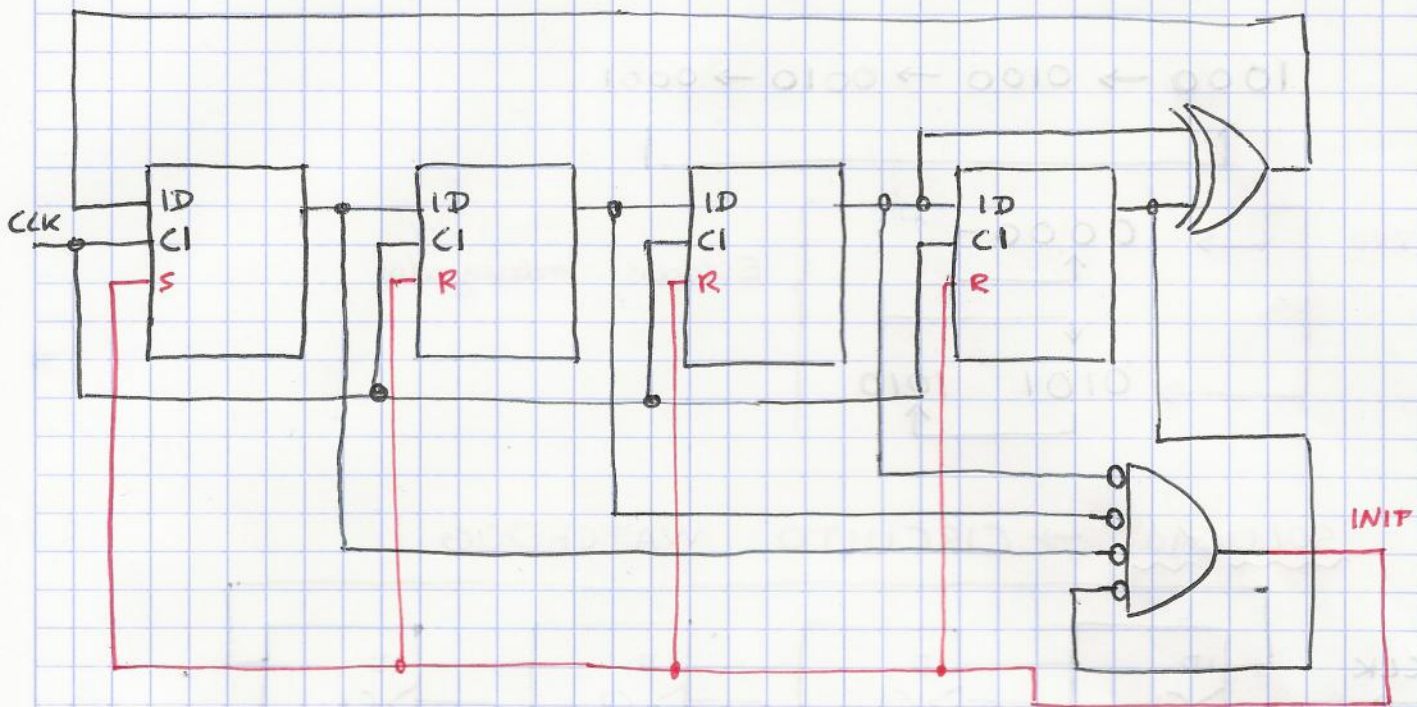
↳ Permite Retornar para o estado inicial (ex. 0100)

Contador Johnson

$2N$ estados
de 2^N estados
possíveis



Contactor LFSR (Linear Feedback Shift Register)



Estorilo de LOCKOUT

15 Estados de possíveis

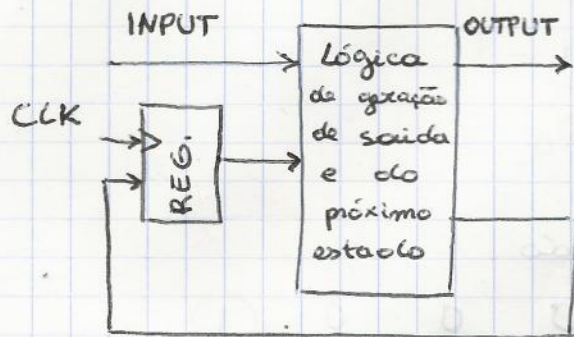
Aula 15

SCS - Definições

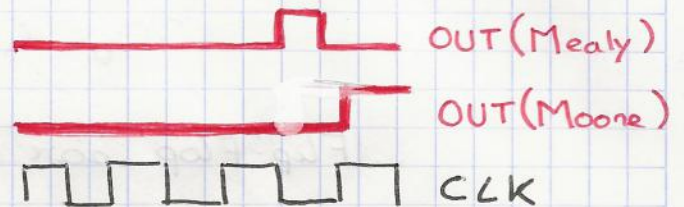
Máquinas de Moore vs. Máquinas de Mealy

A saída depende das Variáveis de estado actuais

A saída é função das variáveis de estado actuais e do valor das entradas presentes no circuito.

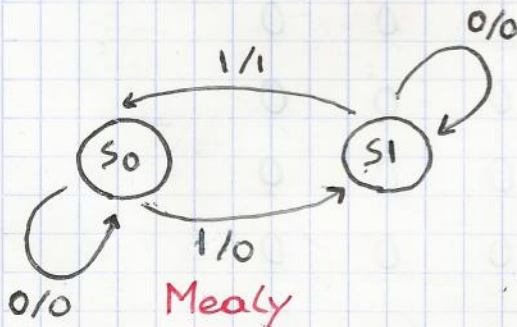


(1 0 1) (0 0 0) (1 0 0) INPUT



Distincão entre Outputs na detecção de pacotes com nº ímpar de bits = 1

Diagrama de Estados

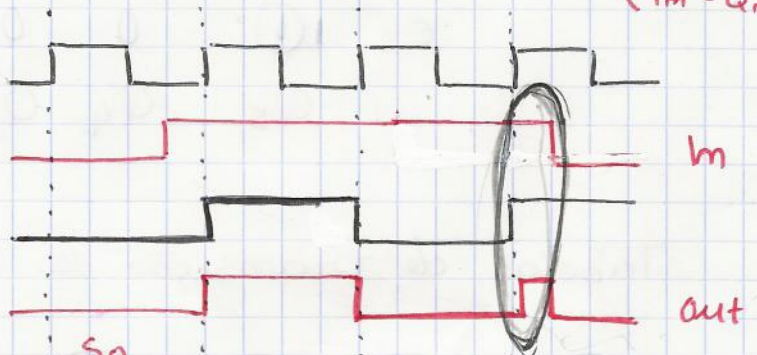


In	Out
0 → 0	
0 → 1	
1 → 0	
1 → 1	

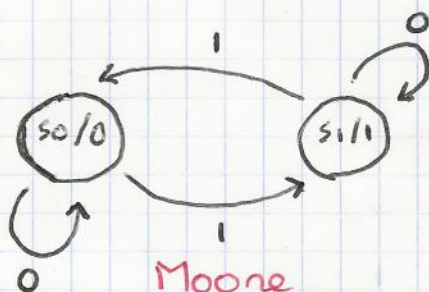
T_{m+1}
 T_m
 U
 S_1
 S_0

Máquina de Mealy (a saída passa a 0)

($T_m = Q_m$)

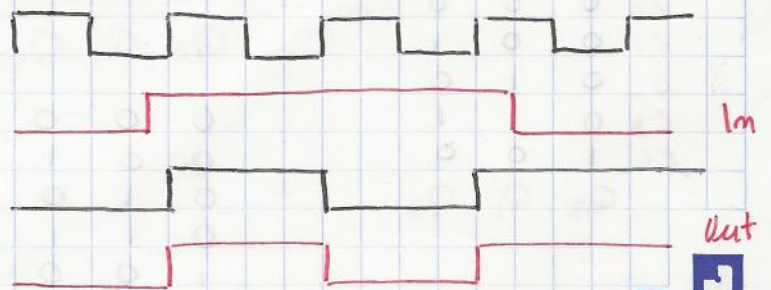


In	Out
0 → X	T_m
1 → 0	S_1
1 → 1	S_0



Moore

In	out+1	T_{m+1}
0	$\frac{out}{out}$	$\frac{T_m}{T_m}$
1		



Codificação de Estados

Codificação Bimária

• Mais eficiente

Menos Flip-Flops utilizados

E_0	0	0	0
E_1	0	0	1
E_2	0	1	0
E_3	0	1	1
E_4	1	0	0
E_5	1	0	1
	Q_2	Q_1	Q_0

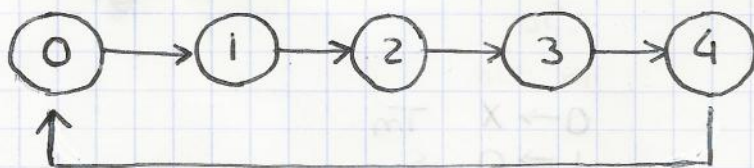
Não é necessário que E_m corresponda a m em binário na codificação

Flip-Flop por Estado

• Funções mais simples mas em maior número

E_0	0	0	0	0	0	1
E_1	0	0	0	0	1	0
E_2	0	0	0	1	0	0
E_3	0	0	1	0	0	0
E_4	0	1	0	0	0	0
E_5	1	0	0	0	0	0
	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0

Tabelas de Transição de Estados



E_0	0	0	0
E_1	0	0	1
E_2	0	1	0
E_3	0	1	1
E_4	1	0	0
	Q_2	Q_1	Q_0

E_m	Q_2	Q_1	Q_0	E_{m+1}	Q_2	Q_1	Q_0
0	0	0	0	1	0	0	1
1	0	0	1	2	0	1	0
2	0	1	0	3	0	1	1
3	0	1	1	4	1	0	0
4	1	0	0	0	0	0	0

Implementação do Circuito anterior com FFD's

$$Q_2(m+1) = D_2$$

$$Q_1(m+1) = D_1$$

$$Q_0(m+1) = D_0$$

Implementação das funções em função de $Q_2(m), Q_1(m), Q_0(m)$

D_0

		Q_1, Q_0			
		0 0	0 1	1 1	1 0
Q_2	0	1	0	0	1
	1	0	X	X	X

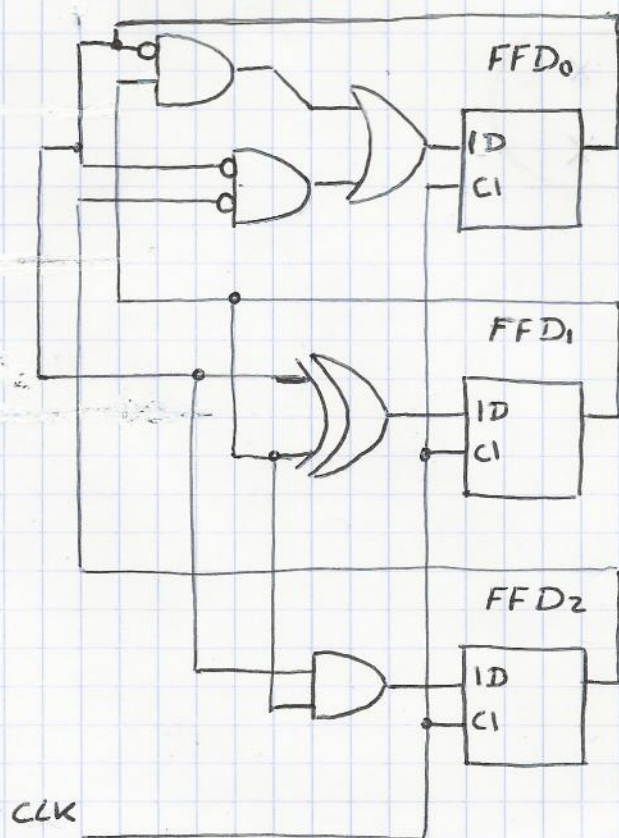
		Q_1, Q_0			
		0 0	0 1	1 1	1 0
Q_2	0	0	1	0	1
	1	0	X	X	X

		Q_1, Q_0			
		0 0	0 1	1 1	1 0
Q_2	0	0	0	1	0
	1	0	X	X	X

$$D_0 = \overline{Q_0} \overline{Q_2} + Q_1 \overline{Q_0}$$

$$D_1 = \overline{Q_1} Q_0 + Q_1 \overline{Q_0} = Q_1 \oplus Q_0$$

$$D_2 = Q_1 Q_0$$



Implementação do circuito anterior com FF's JK

$Q_m \rightarrow Q_{m+1}$	J	K
0 $\rightarrow$ 0	0	X
0 $\rightarrow$ 1	1	X
1 $\rightarrow$ 0	X	1
1 $\rightarrow$ 1	X	0

J_2	K_2	J_1	K_1	J_0	K_0
0	X	0	X	1	X
0	X	1	X	X	1
0	X	X	0	1	X
1	X	X	1	X	1
X	1	0	X	0	X

J_0

1	X	X	1
0	X	X	X

K_0

X	1	1	X
X	X	X	X

$$J_0 = \overline{Q_2}$$

$$K_0 = 1$$

J_1

0	1	X	X
0	X	X	X

K_1

X	X	1	0
X	X	X	X

$$J_1 = Q_0$$

$$K_1 = Q_0$$

J_2

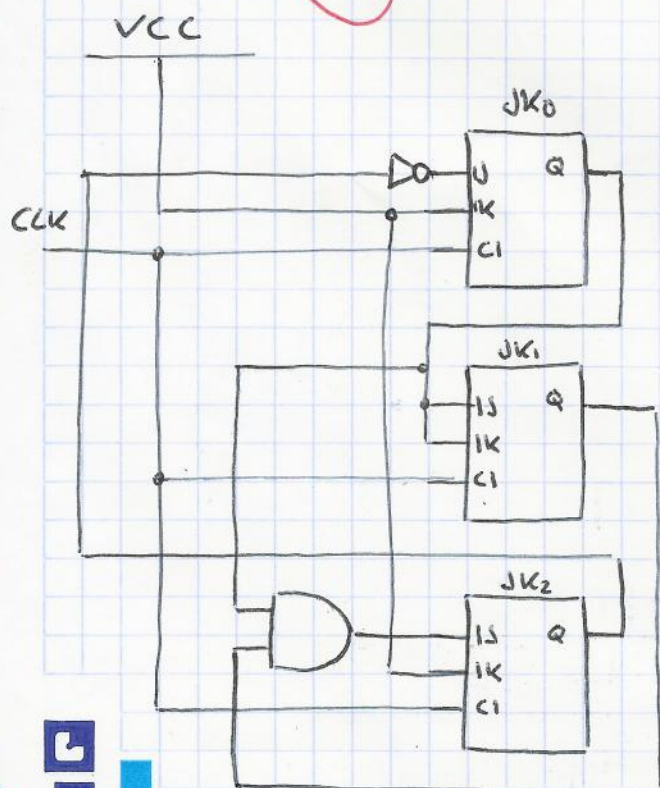
0	0	1	0
X	X	X	X

K_2

X	X	X	X
1	X	X	X

$$J_2 = Q_1 Q_0$$

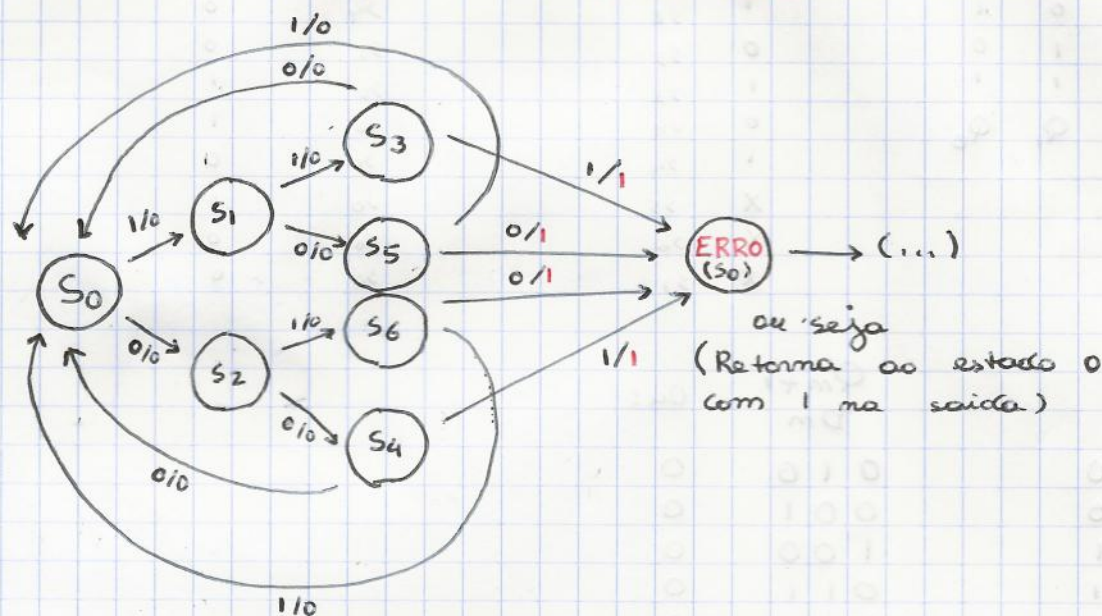
$$K_2 = 1$$



Aula 16

SCS - Minimização do número de estados

Diagrama de estados - Detector de paridade de pacotes de 3 bits



Simplificação de Estados

S0							
S1	X						
S2	X	X					
S3	X	X	X				
S4	X	X	X	✓			
S5	X	X	X	X	X		
S6	X	X	X	X	X	✓	
	S0	S1	S2	S3	S4	S5	S6

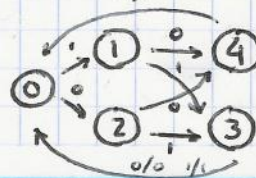
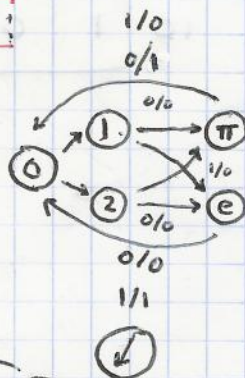
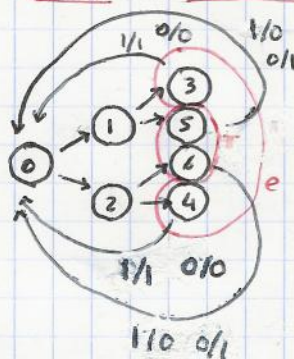
Estados equivalentes implicam:

- Igual por Input / Output
- Igual por Input / Estado seguinte
- Igual por Input / Estados equivalentes seguintes

Igual METAESTADO

S1	S1 ↔ S3 → X
	S2 ↔ S4 → X
X ← S1	S1 ↔ S5 → X
X ← S2	S2 ↔ S6 → X
	S3 ↔ S5 → X
	S4 ↔ S6 → X
	S0 S1

conclui-se então:



Implementação do circuito anterior

s_0	0	0	0
s_1	0	0	1
s_2	0	1	0
$s_e \rightarrow s_3$	0	1	1
$s_r \rightarrow s_4$	1	0	0
s_5	1	0	1
s_6	1	1	0
s_7	1	1	1

Estados
Inválidos

Q_2 Q_1 Q_0

Codificação Bimária
dos Estados

h	S_n	S_{n+1}	Out
0	s_0	s_2	0
1	s_0	s_1	0
0	s_1	s_4	0
1	s_1	s_3	0
0	s_2	s_3	0
1	s_2	s_4	0
0	s_3	s_6	0
1	s_3	s_0	1
0	s_4	s_0	1
1	s_4	s_0	0
X	s_5	s_0	0
X	s_6	s_0	0
X	s_7	s_0	0

h	Q_n	Q_{n+1} D_n	Out
0	000	010	0
1	000	001	0
0	001	100	0
1	001	011	0
0	010	011	0
1	010	100	0
0	011	000	0
1	011	000	1
0	100	000	1
1	100	000	0
X	101	000	0
X	110	000	0
X	111	000	0

h	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

D_{n+1}

(...)