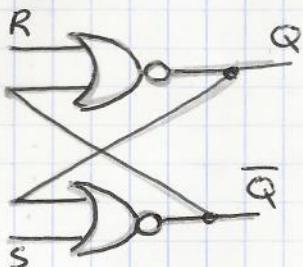


## Aula II

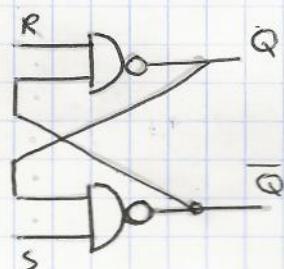
CSB

Latches e Flip-Flops

### Latch RS = Reset / Set

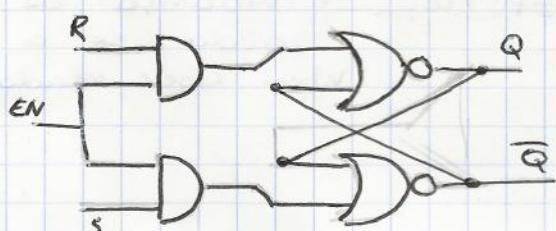


S	R	$Q_{m+1}$	$\bar{Q}_{m+1}$	HOLD
0	0	$Q_m$	$\bar{Q}_m$	RESET
0	1	0	1	—
1	0	1	0	SET
1	1	U	U	—

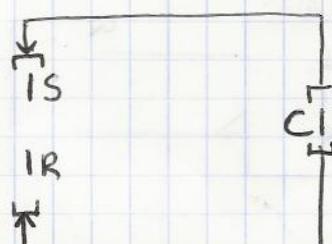
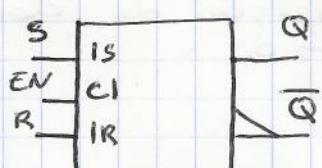


S	R	$Q_{m+1}$	$\bar{Q}_{m+1}$	SET
0	0	U	U	—
0	1	—	0	RESET
1	0	0	1	—
1	1	$Q_m$	$\bar{Q}_m$	HOLD

### Latch RS sincronizado



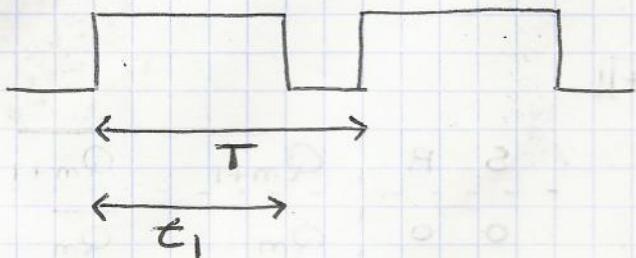
EN	R	S	$Q_{m+1}$	HOLD
0	0	0	$Q_m$	HOLD
1	0	0	$Q_m$	HOLD
1	1	—	0	SET
1	—	0	1	RESET



Set e Reset  
dependentes  
do valor da  
clock

## Circuitos Síncronos

Output com periodos HIGH e LOW bem definidos



$T$  - período

$\epsilon_1$  - Duty Cycle

## Flip-Flops vs. Latches

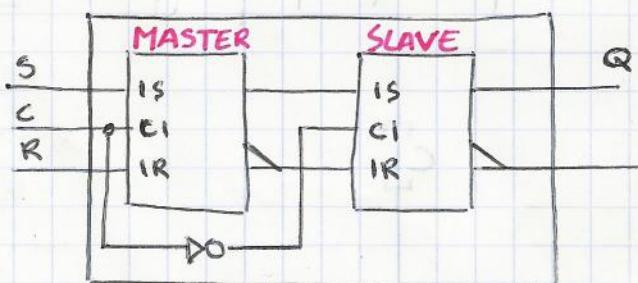
### Flip-Flop Master-Slave



Latch → Mudança de saída se clock = 1



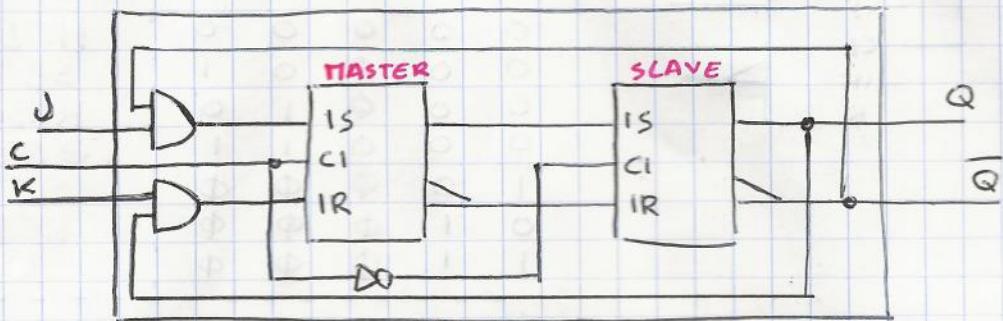
Flip-Flop → Mudança de saída se o v. bool. clock mudar



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 o  $C=0$

## Flip-Flop JK Master-Slave



J	K	Q <sub>m+1</sub>
0	0	Q <sub>m</sub>
0	1	0
1	0	1
1	1	Q <sub>m</sub>

HOLD  
RESET  
SET  
TOGGLE

Resolução do estado indeterminado

## Flip-Flop Master-Slave Pulse-Triggered

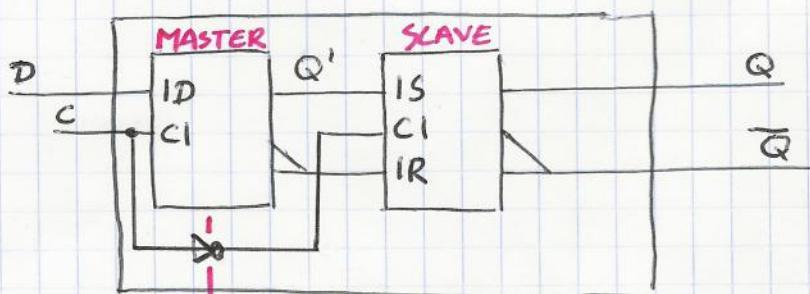
VS:

## Flip-Flop Master-Slave Edge-Triggered

- Pulse Triggered → Exemplos anteriores  
→ Sensíveis a qualquer pulse change, mesmo que acidental

- Edge Triggered

Rising Edge Delay

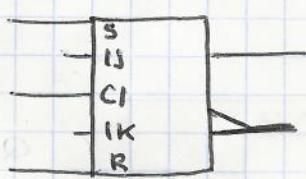


D	CLK	Q <sub>+1</sub>
0	↑	0
1	↓	1

MASTER TIPO D } O estado passado ao SLAVE RS é definido pelas transições de clock )

+ ET - sensível à transição 0 → 1  
- ET - sensível à transição 1 → 0

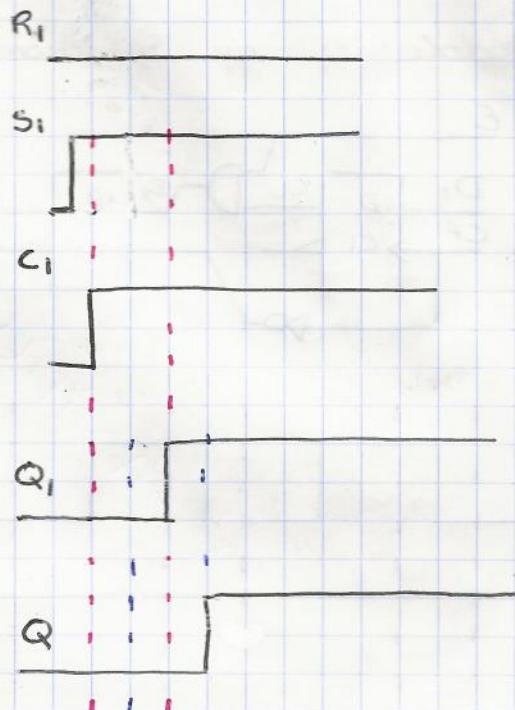
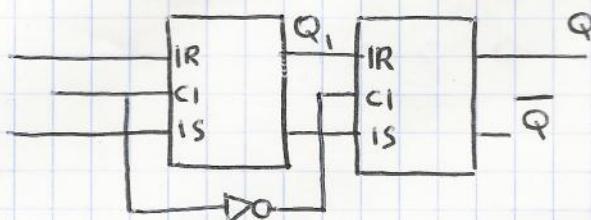
## Entradas Assimétricas



S	R	C	J	K	Q <sub>out</sub>
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



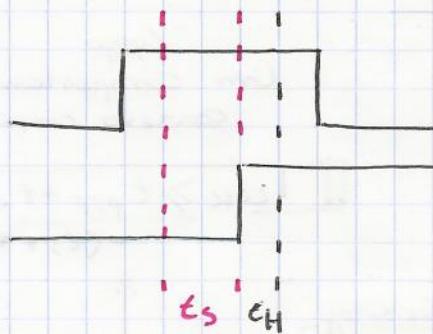
O tempo de propagação é então o desfasamento temporal ( $\Delta t$ ) entre a transição do clock e a mudança do valor do output.

Desfasamento entre transições

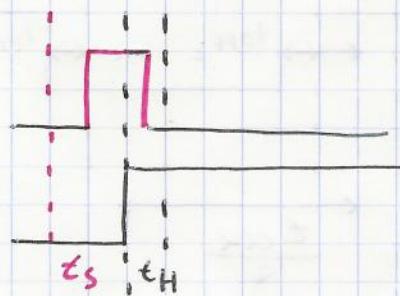
Tempos de preparação e manutenção

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



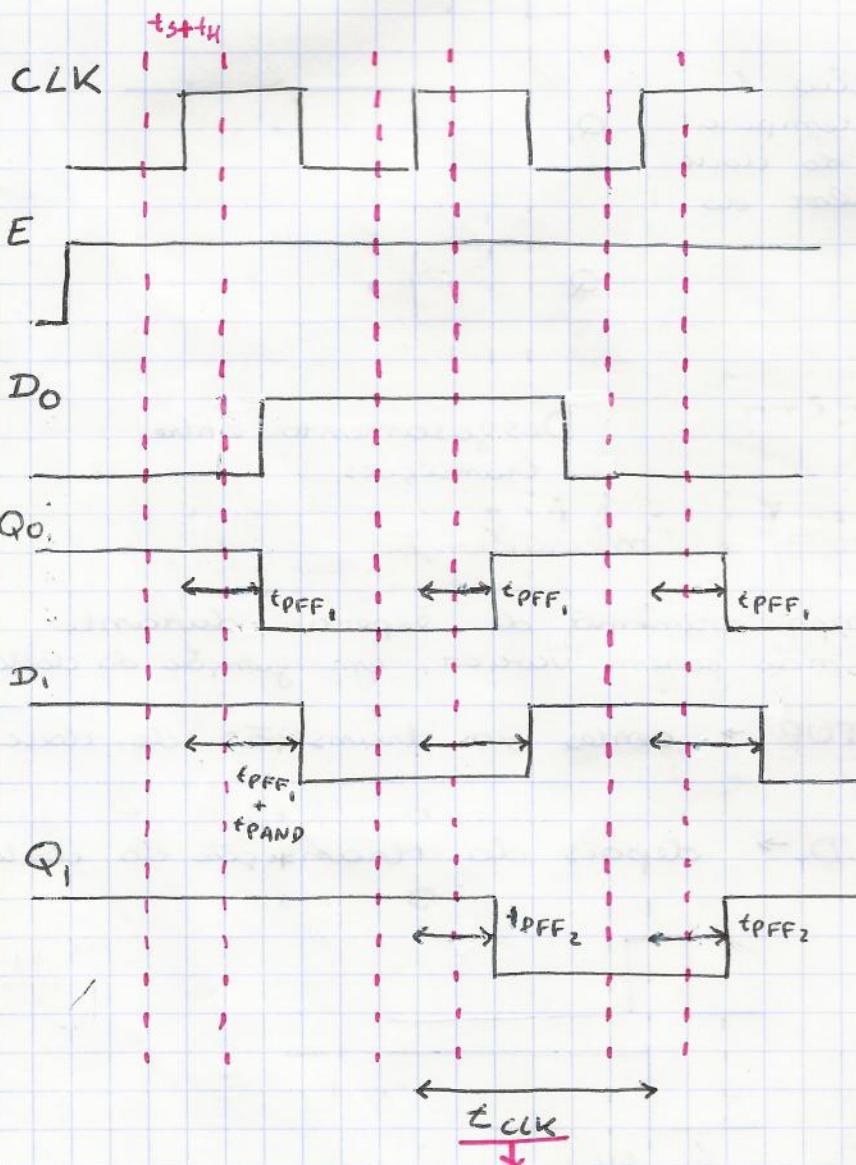
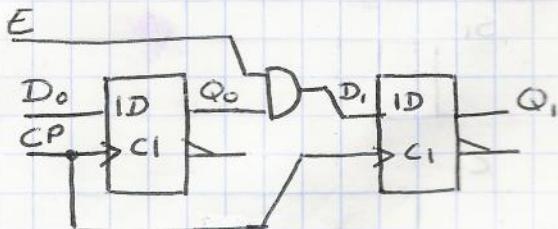
Garantia de funcionamento correto



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 flanco da sensibilidade.

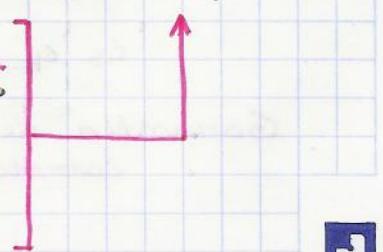
## Metodologia de Sincronização Temporal



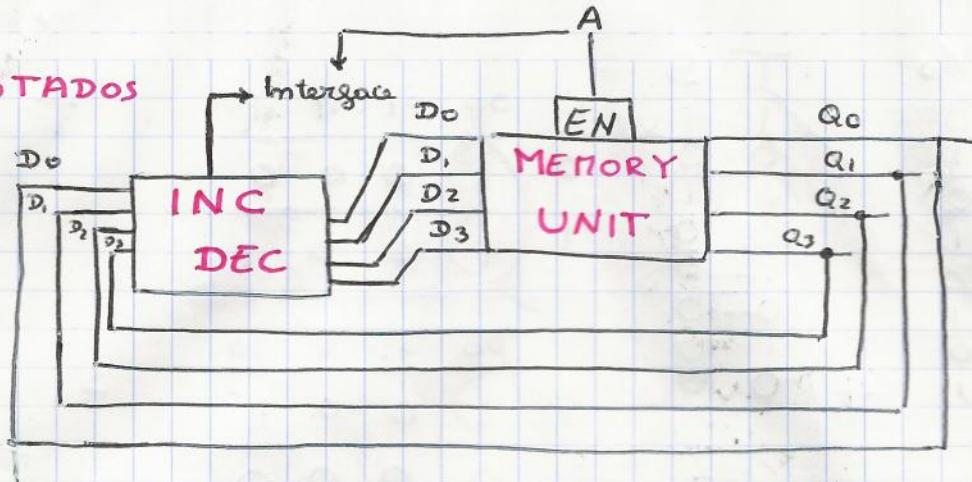
Numa execução de overclocking, o período do relógio é diminuído o que pode causar a alterações do valor das entradas na banda do  $t_s$  levando a possíveis comportamentos erráticos do circuito.

Logo  
Um comportamento correto implica

$$T_{CLK} \geq t_{PFF} + t_{esgica} - t_s$$



ESTADOS



ESTADO  
PRESENTE

ESTADO  
SEGUINTE

, A, INC/DEC, Q<sub>3(m-1)</sub>, Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub>, Q<sub>3(m)</sub>, Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub>  
(...)

		0	1	0	0	0	1	1	1
		0	1	0	0	0	1	0	0
		1	0	0	0	0	0	0	1
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0

+1  
-1  
No FORWARD

1)

TESTE

K <sub>2</sub>	K <sub>1</sub>	K <sub>0</sub>	Z	Y	X	A	B	C	T	S
0	0	0	0	0	0	1	0	0	1	1
0	0	1	0	1	1	0	1	0	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0
1	0	1	0	0	1	0	1	0	0	0
1	1	0	0	0	1	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0

10:38

b)

K<sub>1</sub> K<sub>0</sub>

K <sub>2</sub>	00	01	11	10
0	0	1	1	1
1	0	1	1	0

$$K_1 + K_0 \cdot K_0 + \bar{K}_2 = 2$$

$$= \frac{K_1 + K_0}{K_1 + K_0 + K_0 + \bar{K}_2}$$

K<sub>0</sub>

K<sub>1</sub>

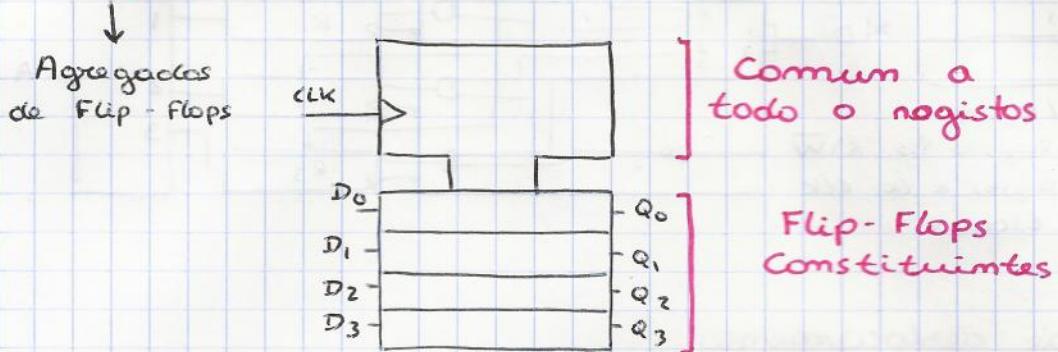
K<sub>2</sub>



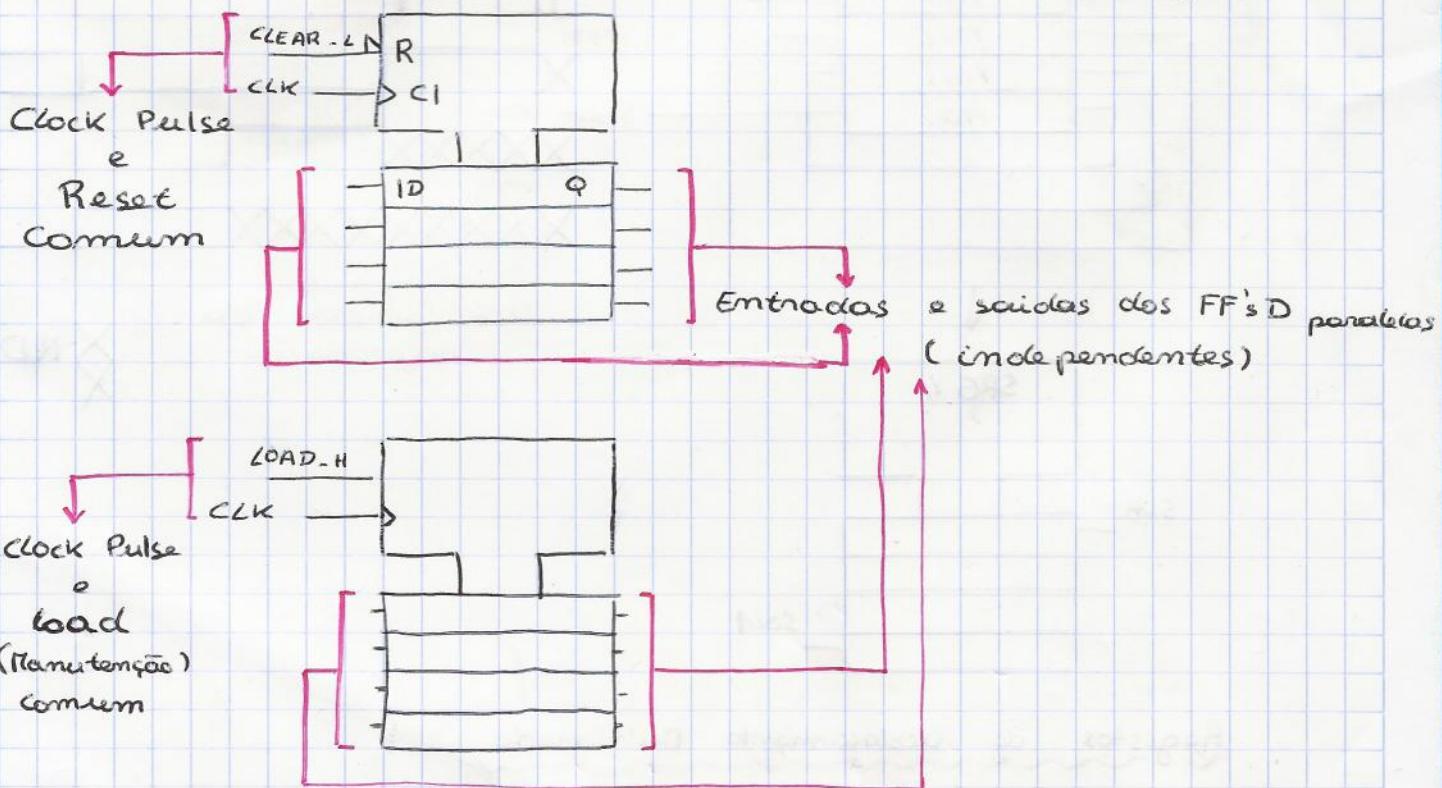
10:38

Aula - 13  
CSB  
Registros

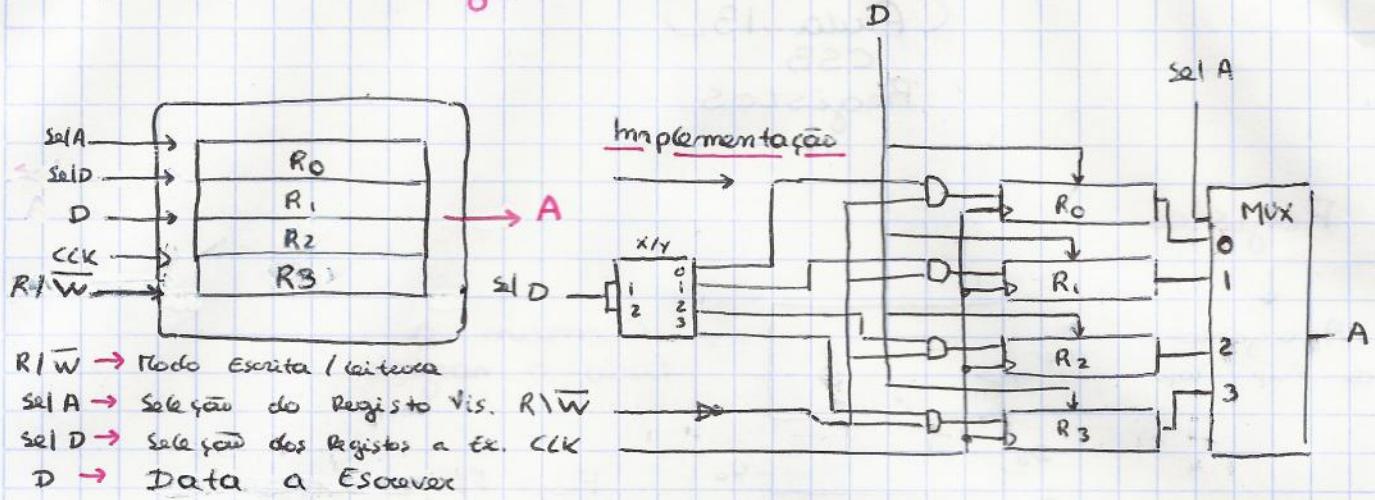
## Registros



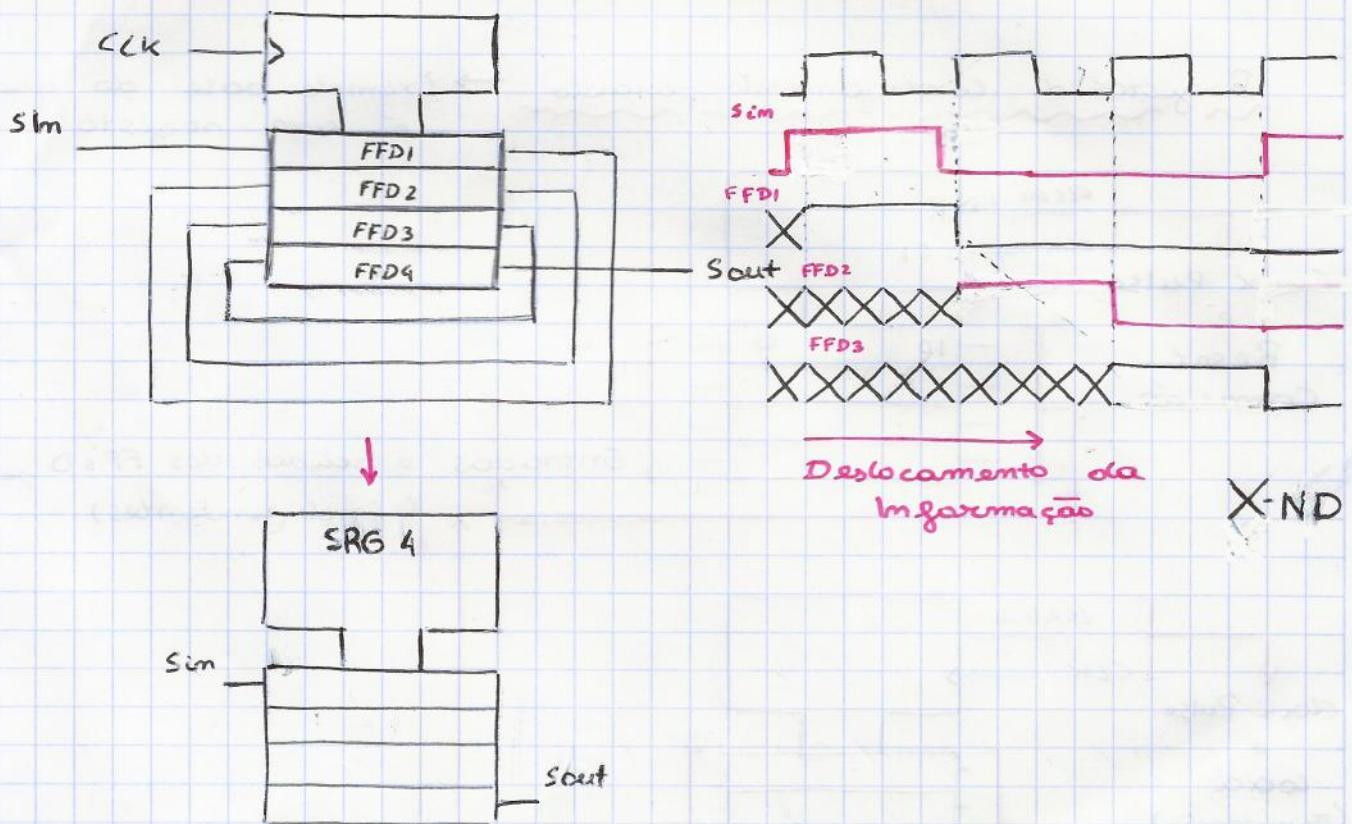
Registros de carregamento paralelo → Exemplo base do que é um registro



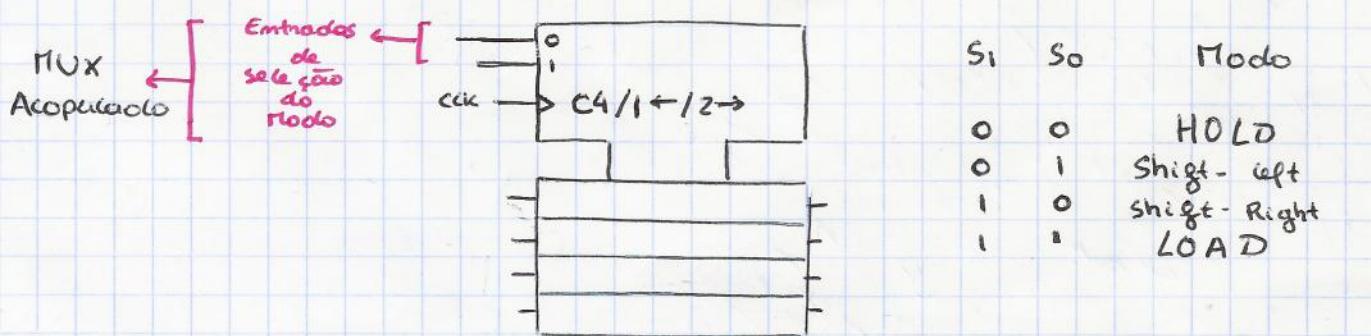
## Banco de Registros



## Registros de deslocamento



## Registros de deslocamento Multimodo

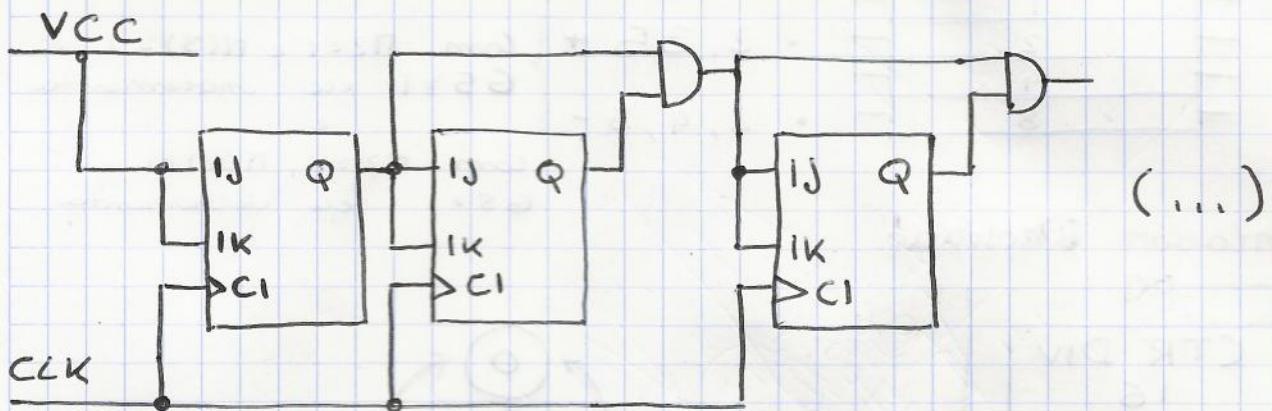


## (Aula - 14)

### CSB - Contadores

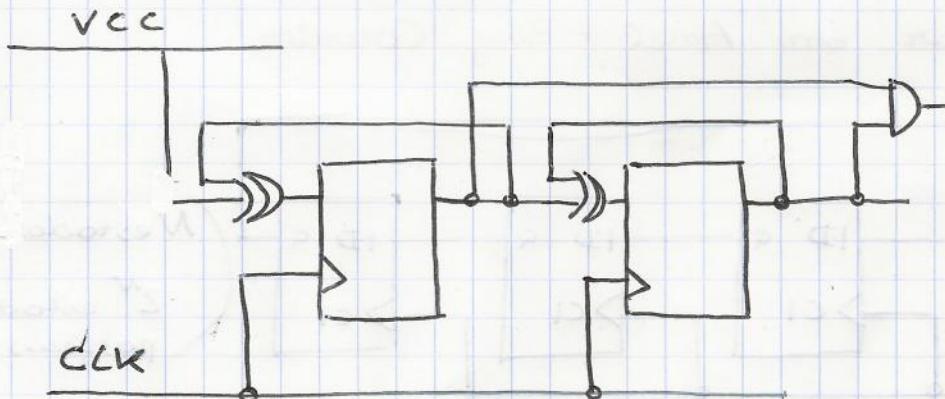
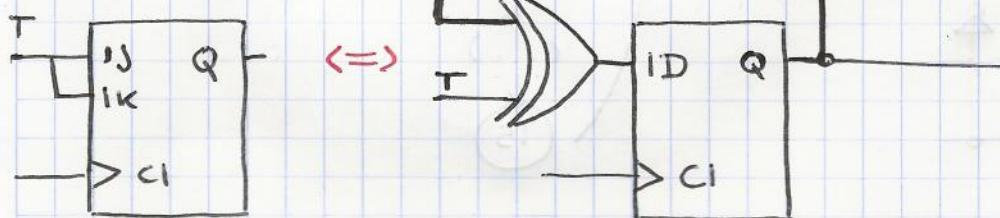
#### Contador Binário

(Tipo - JK)



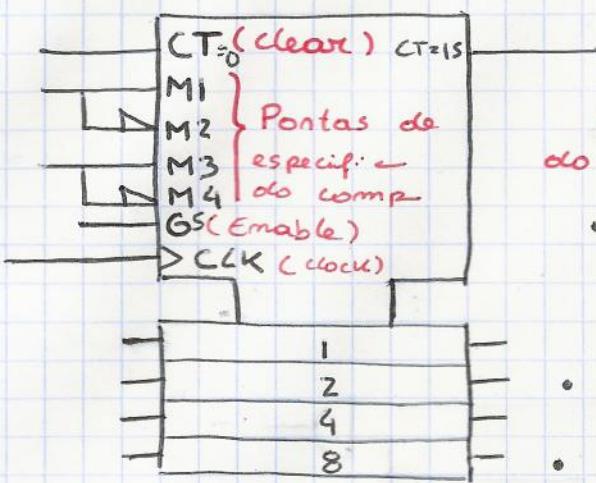
$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{t_{pFF} + (m-2)t_{PAND} + t_{SFF}}$$

(Tipo D)



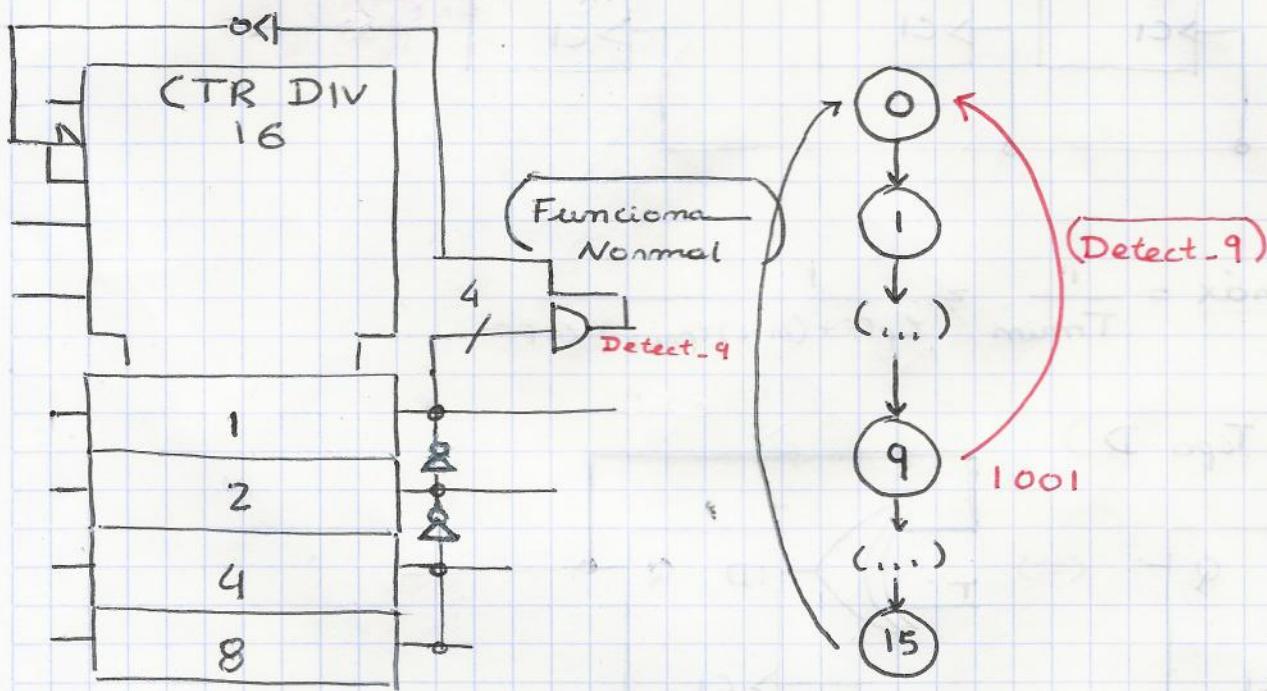
1000 ← 0100 ← 0010 ← 0001

## Componente

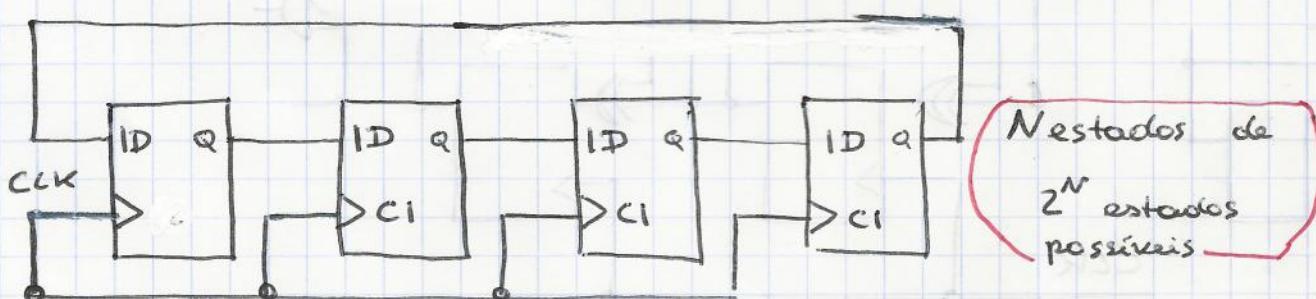


- Se o Componente é do tipo → Aparece à frente do  $CLK$
- $2, 5 + \rightarrow$  Com  $M_2$  (count) = 1 e  $GS = 1$  ele incrementa
  - $2, 3, 5 + \rightarrow$  Com  $M_2 = 1$  e  $M_3 = 1$  e  $GS = 1$  ele incrementa e com  $M_2 = 1$ ,  $M_4 = 1$  e  $GS = 1$  ele decrementa
  - $2, 4, 5 - \rightarrow$  Com  $M_2 = 1$ ,  $M_4 = 1$  e  $GS = 1$  ele decrementa

## Contador Decimal

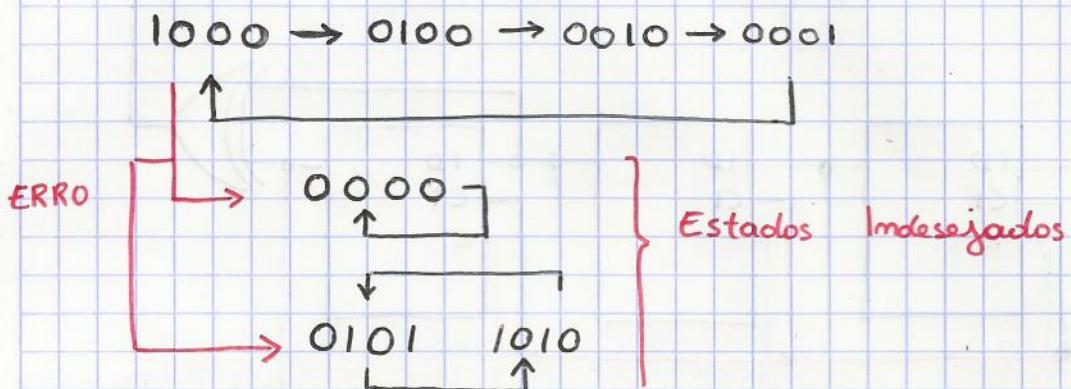


## Contador em Anel - Ring Counter

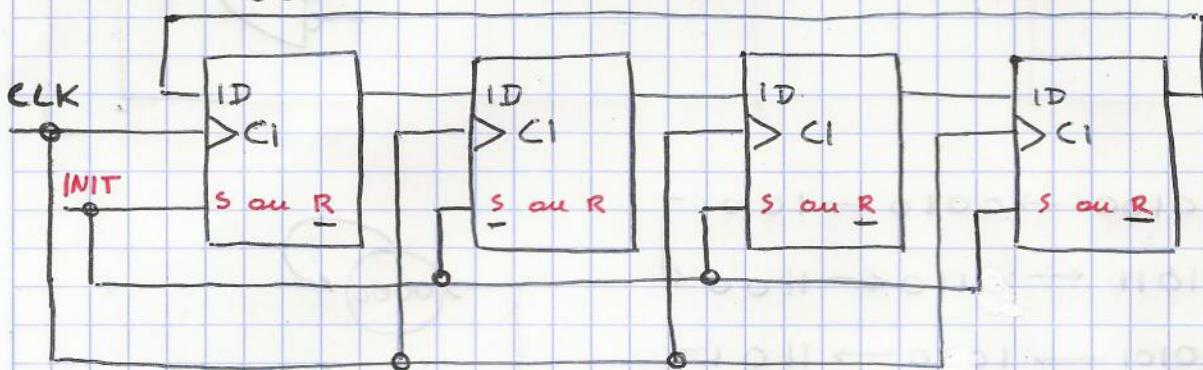


$1000 \rightarrow 0100 \rightarrow 0010 \rightarrow 0001$

## Contadores "LOCK-OUT"



SOLUÇÃO: → CIRCUITO WATCHDOG

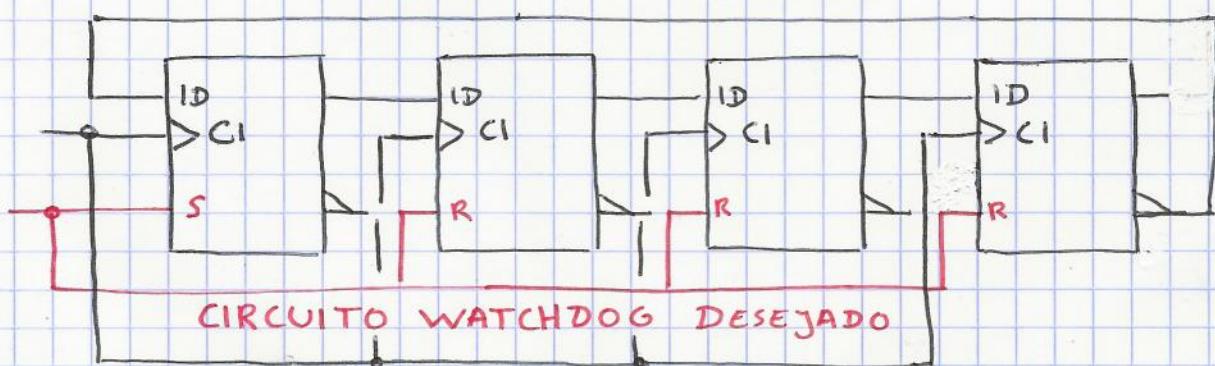


INIT → Detecta estados indesejados

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

## Contador Johnson

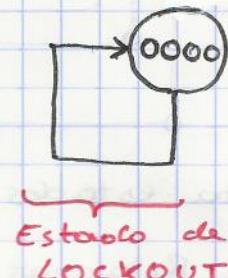
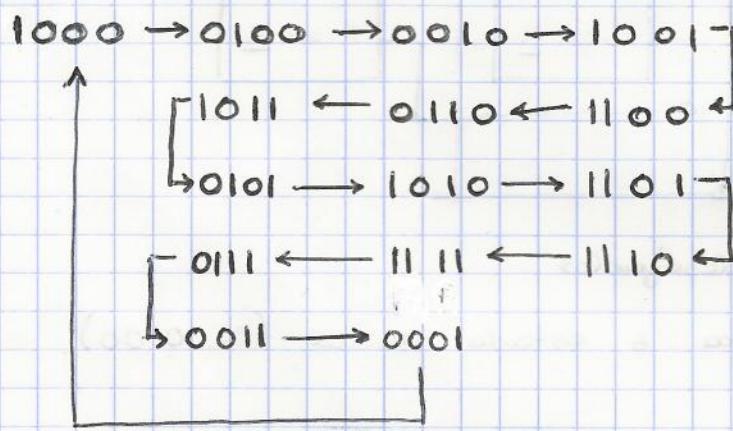
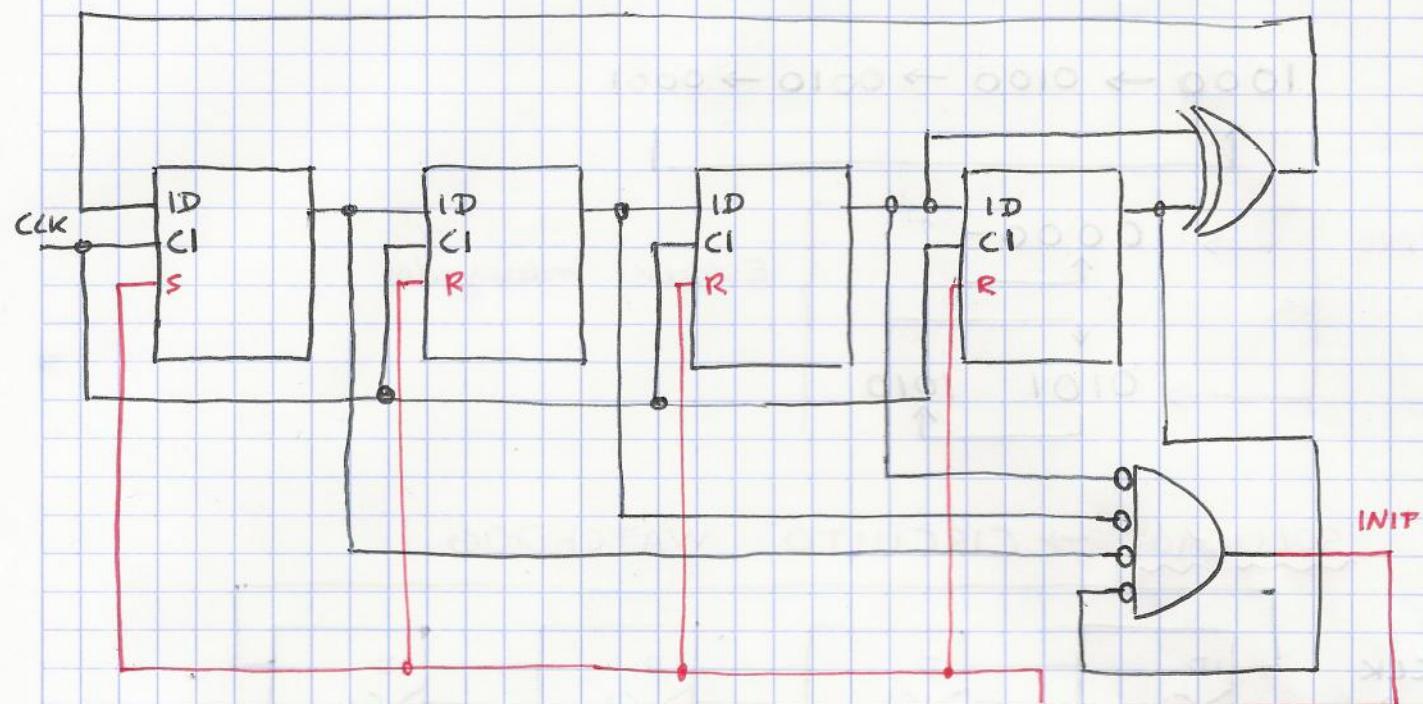
$2^N$  estados de  $2^N$  estados possíveis



(1000) → 1100 → 1110 → 1111

↑  
0000 ← 0001 ← 0011 ← 0111

## Contador LFSR (Linear Feedback Shift Register)



15 Estados de possíveis

16

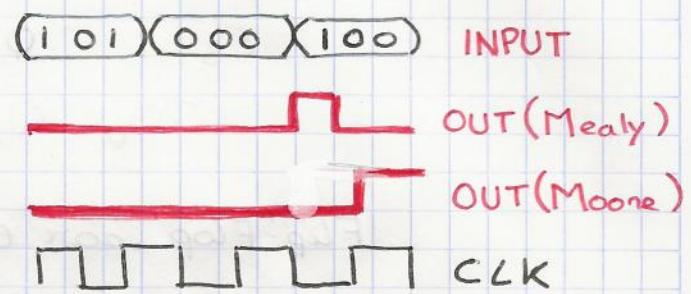
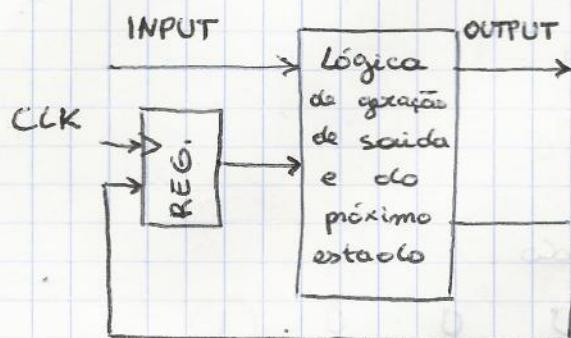
## Aula 15

### SCS - Definições

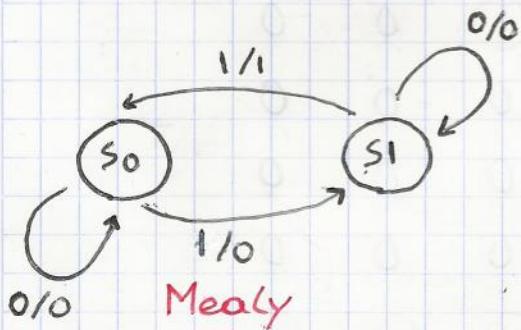
#### Máquinas de Moore vs. Máquinas de Mealy

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.



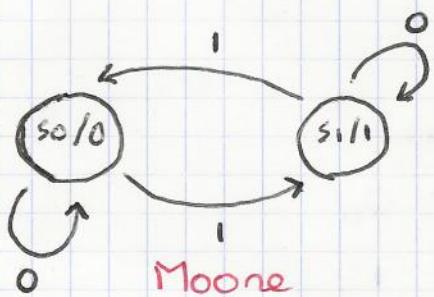
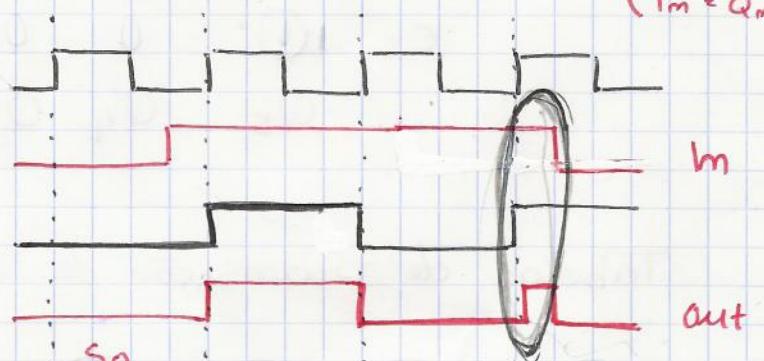
#### Diagrama de Estados



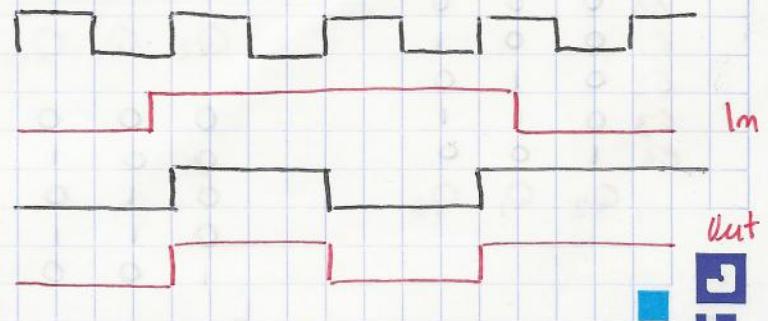
In	Out	Tm+1
0 → 0	0	Tm
0 → 1	1	S1
1 → 0	0	S0
1 → 1	1	

Máquina de Mealy  
(a saída passa a 0)

$$(T_m = Q_m)$$



m	out	Tm+1
0 → X	X	Tm
1 → 0	0	S1
1 → 1	1	S0



## Codificação de Estados

### Codificação - Binária

- Mais eficiente

$\uparrow$   
Menos Flip-Flops utilizados

$E_0 \quad 0 \quad 0 \quad 0$

$E_1 \quad 0 \quad 0 \quad 1$

$E_2 \quad 0 \quad 1 \quad 0$

$E_3 \quad 0 \quad 1 \quad 1$

$E_4 \quad 1 \quad 0 \quad 0$

$E_5 \quad 1 \quad 0 \quad 1$

$Q_2 \quad Q_1 \quad Q_0$

Não é necessário que  $E_m$  corresponda a  $n$  em binário na codificação

### Flip-Flop para Estado

- Funções mais simples mas em maior número

$E_0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1$

$E_1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0$

$E_2 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0$

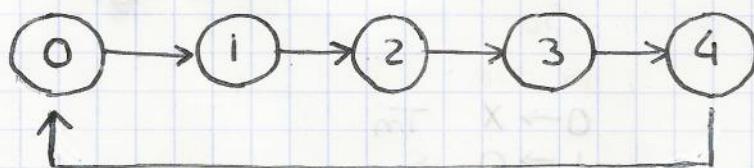
$E_3 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0$

$E_4 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0$

$E_5 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$

$Q_5 \quad Q_4 \quad Q_3 \quad Q_2 \quad Q_1 \quad Q_0$

## Tabelas de transição de estados



$E_m \quad E_{m+1}$

$E_0 \quad 0 \quad 0 \quad 0$

$E_1 \quad 0 \quad 0 \quad 1$

$E_2 \quad 0 \quad 1 \quad 0$

$E_3 \quad 0 \quad 1 \quad 1$

$E_4 \quad 1 \quad 0 \quad 0$

$Q_2 \quad Q_1 \quad Q_0$

$Q_2 \quad Q_1 \quad Q_0$

$Q_2 \quad Q_1 \quad Q_0$

$0 \quad 0 \quad 0$

$0 \quad 0 \quad 1$

$0 \quad 1 \quad 0$

$0 \quad 1 \quad 1$

$1 \quad 0 \quad 0$

$0 \quad 0 \quad 0$

## 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  $Q_2(m)$ ,  $Q_1(m)$ ,  $Q_0(m)$

$D_0$

		$Q_1, Q_0$		00	01	11	10
		$Q_2$	0	1	0	0	1
		0	0	X	X	X	X
		1	0	X	X	X	X

$D_1$

		$Q_1, Q_0$		00	01	11	10
		$Q_2$	0	0	1	0	1
		0	0	X	X	X	X
		1	0	X	X	X	X

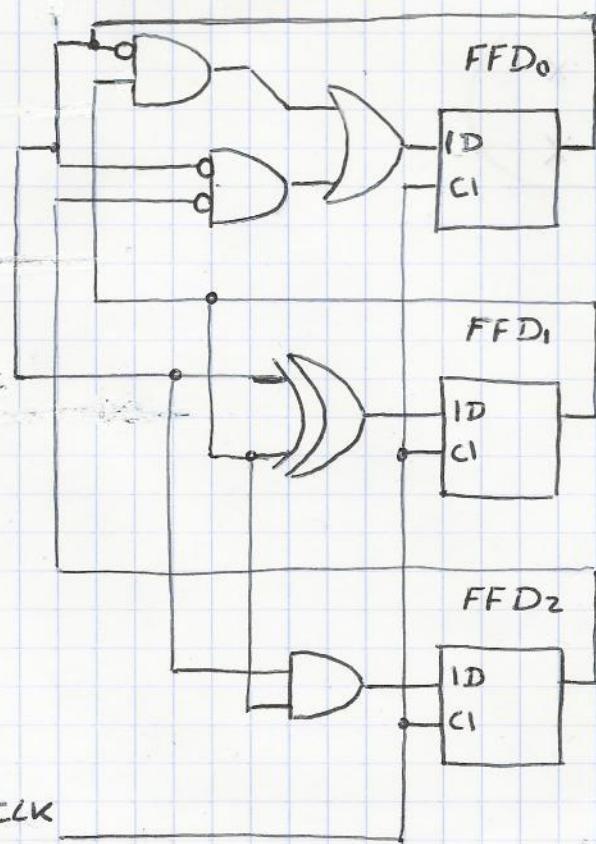
$D_2$

		$Q_1, Q_0$		00	01	11	10
		$Q_2$	0	0	0	1	0
		0	0	X	X	X	X
		1	0	X	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 → 0	0	X
0 → 1	1	X
1 → 0	X	1
1 → 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

$J_0$                            $K_0$

1	X	X	1
0	X	X	X

X	1	1	X
X	X	X	X

$$J_0 = \overline{Q_2}$$

$$K_0 = 1$$

$$J_1 = Q_0$$

$$K_1 = Q_0$$

$$J_2 = Q_1, Q_0$$

$$K_2 = 1$$

$J_1$                            $K_1$

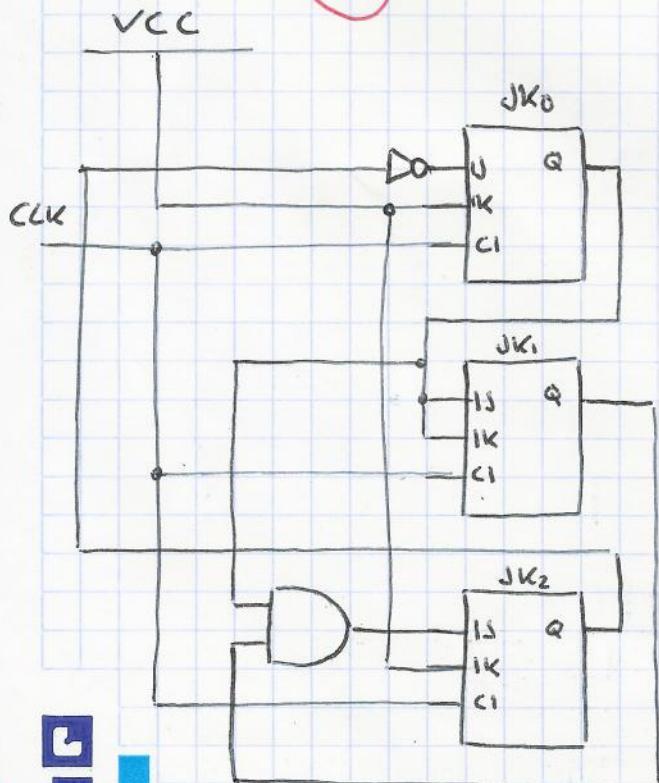
0	1	X	X
0	X	X	X

X	X	1	0
X	X	X	X

$J_2$                            $K_2$

0	0	1	0
X	X	X	X

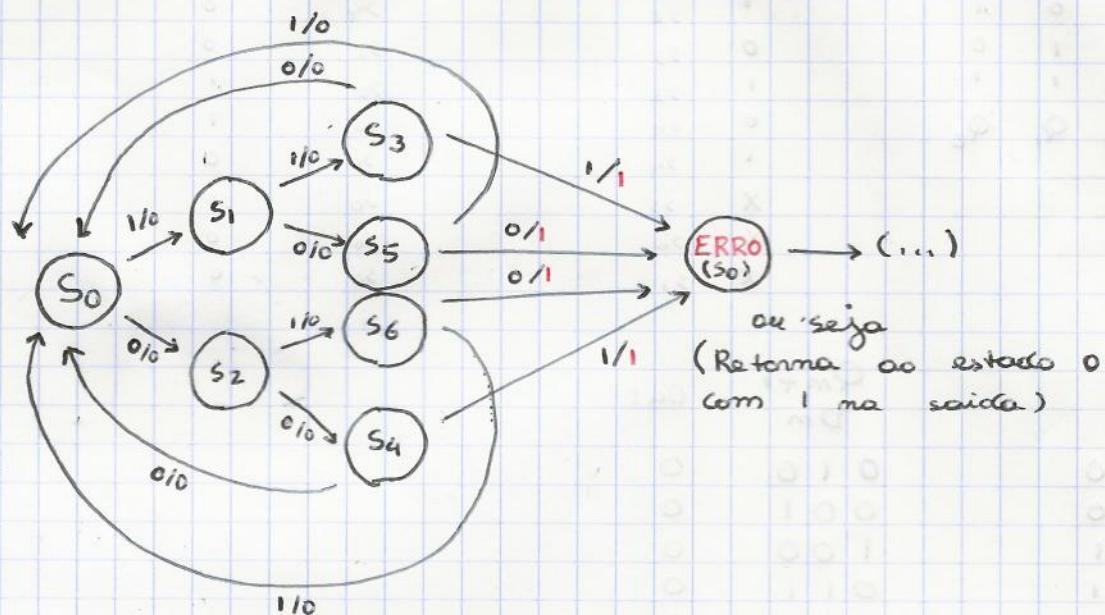
X	X	X	X
1	X	X	X



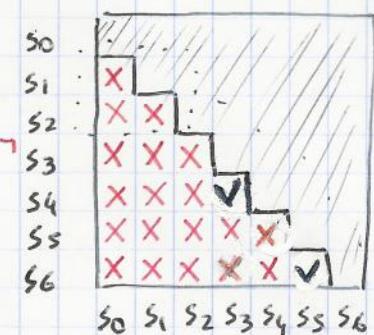
## Aula-16

SCS - Minimização do número de estados

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



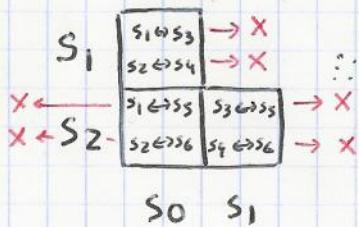
Simplificação de Estados



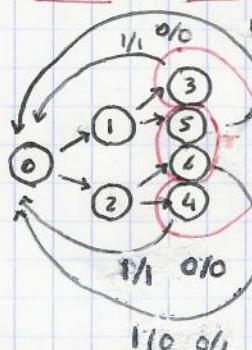
Estados equivalentes implicam:

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

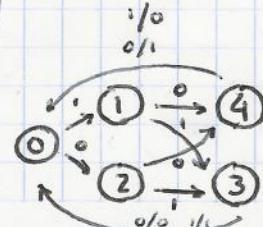
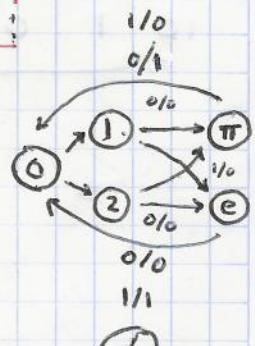
Igual METAESTADO



conclui-se então:



ou seja



## Implementação do circuito anterior

	$I_m$	$S_m$	$S_{m+1}$	Out
$s_0$	0	0	0	$s_2$
$s_1$	0	0	1	$s_1$
$s_2$	0	1	0	$s_4$
$s_e \rightarrow s_3$	0	1	1	$s_3$
$s_\pi \rightarrow s_4$	1	0	0	$s_3$
Estados Inválidos	1	0	1	$s_4$
	1	1	0	$s_0$
	1	1	1	$s_0$
	$Q_2$	$Q_1$	$Q_0$	
<u>Codificação Binária dos Estados</u>				
		X	$s_5$	$s_0$
		X	$s_6$	$s_0$
		X	$s_7$	$s_0$

$I_m$	$Q_m$	$Q_{m+1}$	Out
		$D_m$	
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

$I_m$	00	01	11	10	$D_o$
00	0	0	0	1	
01	0	0	0	0	
11	0	0	0	0	
10	1	1	0	0	

(...)