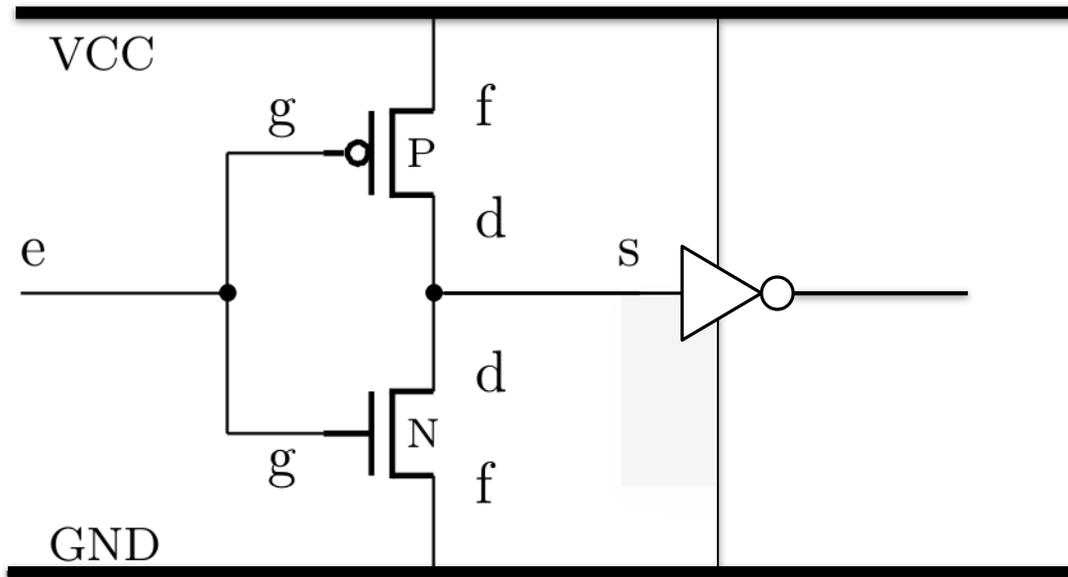




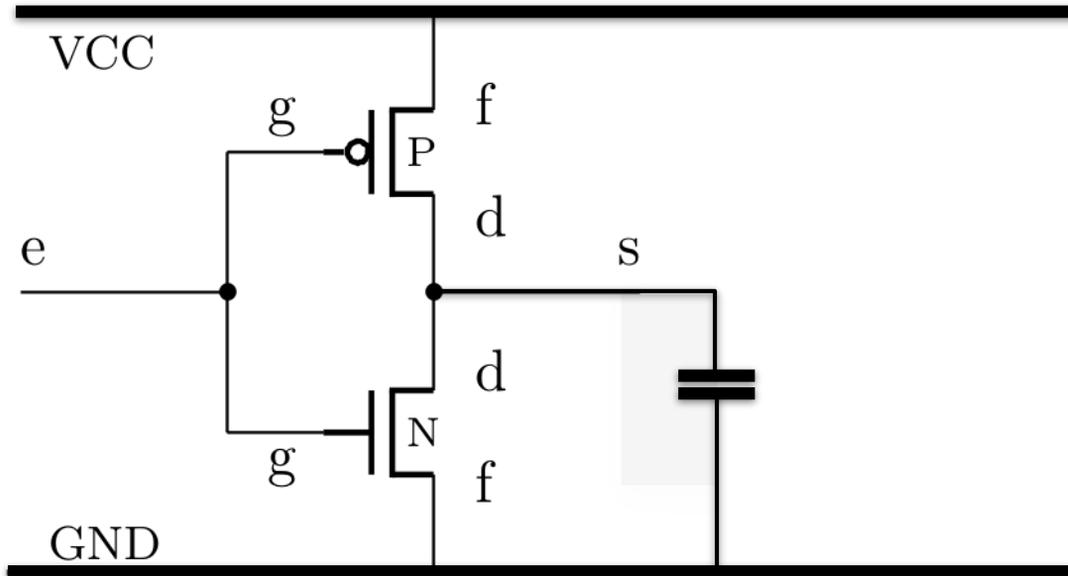
PROJETOS DIGITAIS E MICROPROCESSADORES CIRCUITOS SEQUENCIAIS

Marco A. Zanata Alves

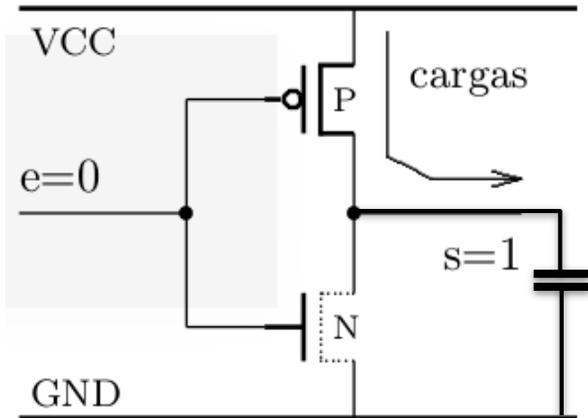
INVERSORES CMOS



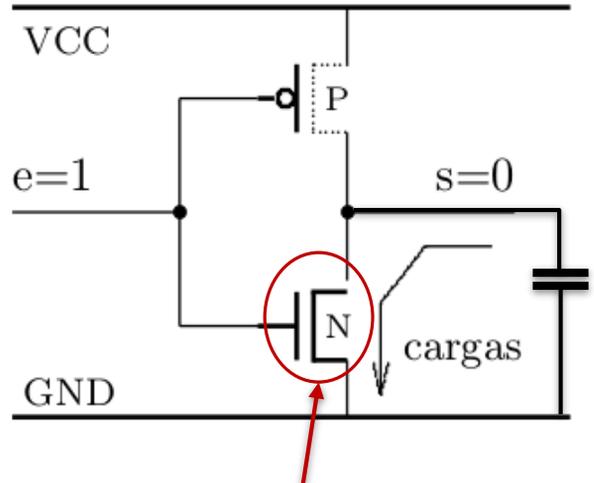
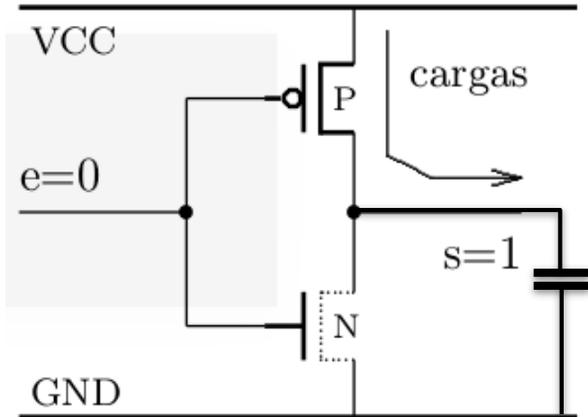
INVERSORES CMOS



INVERSORES CMOS

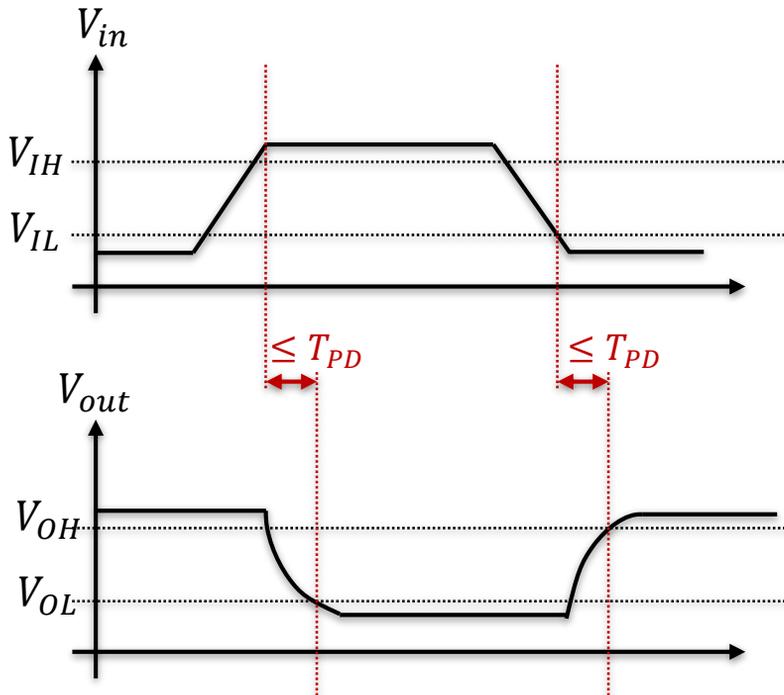


INVERSORES CMOS



Resistência

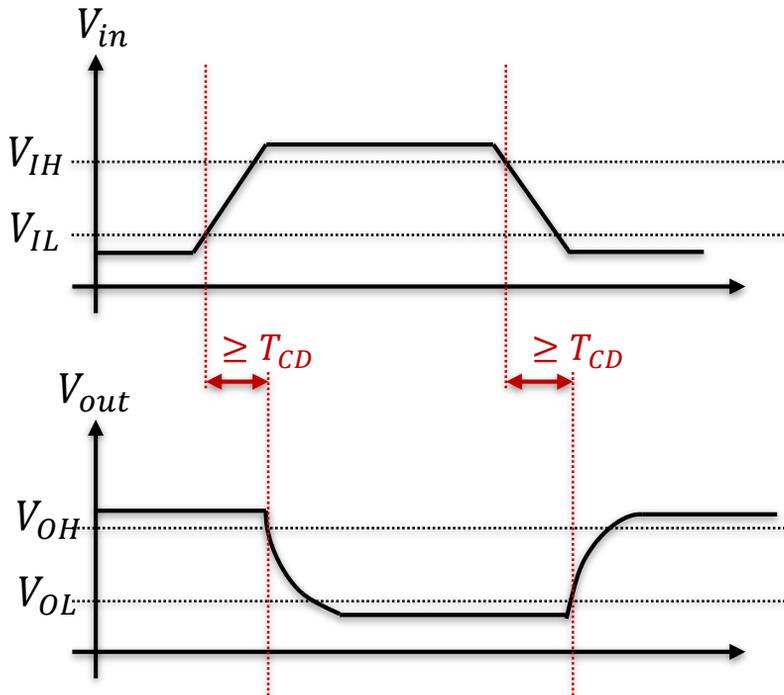
ATRASO DE PROPAGAÇÃO (PROPAGATION DELAY TIME)



Atraso de **propagação** (T_{PD}):
O **limite superior** no atraso
entre as **entradas válidas** até
as **saídas válidas**

Tempo para as saídas
ficarem prontas

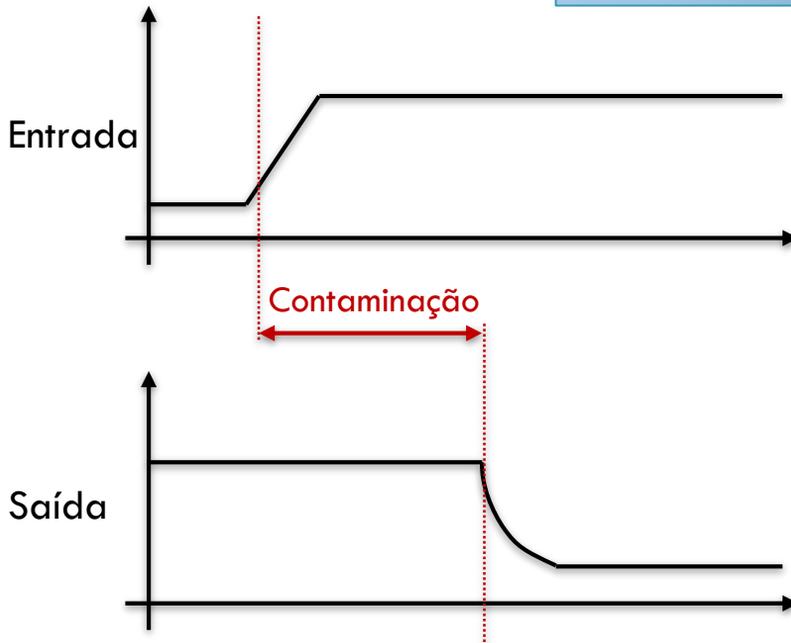
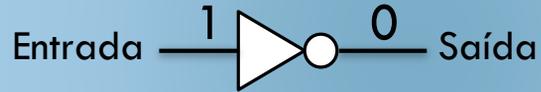
ATRASO DE CONTAMINAÇÃO (CONTAMINATION DELAY TIME)



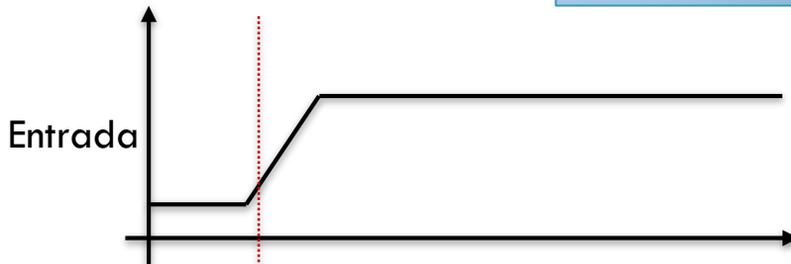
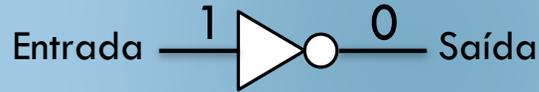
Atraso de **contaminação** (T_{CD}):
O **limite inferior** no atraso
entre as **entradas inválidas**
até as **saídas inválidas**

Tempo para as saídas
ficarem inválidas

CIRCUITOS COM MEMÓRIAS

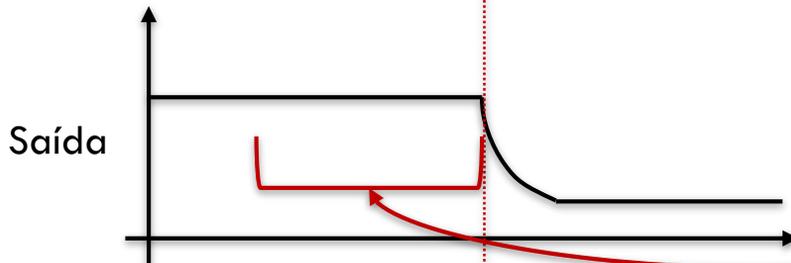


CIRCUITOS COM MEMÓRIAS



Contaminação

Durante um curto período (contaminação) nós temos um efeito de “memória”

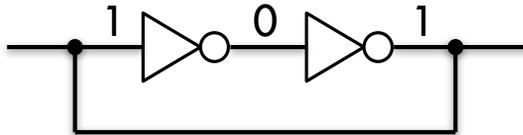


Como fazer esse efeito durar?

CIRCUITOS COM MEMÓRIAS

Estamos interessados agora em formas de armazenar informações em circuitos

O circuito com dois inversores em série, com realimentação é capaz de manter o valor em sua saída enquanto a fonte de energia estiver ligada



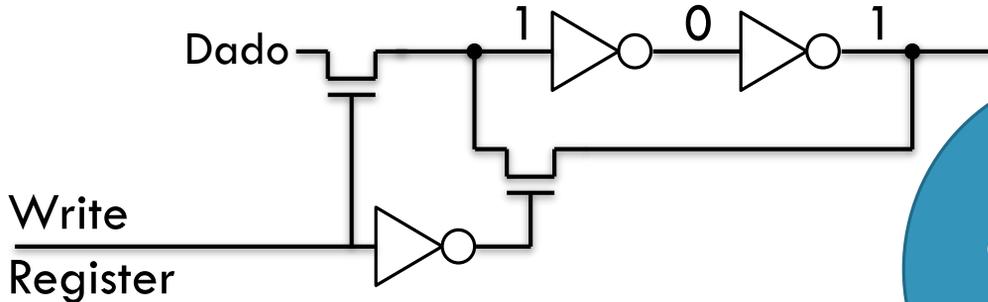
Esse comportamento é função do atraso (tempo de propagação) das portas lógicas

Com o tempo fixo de propagação, os inversores são capazes de manter os sinais estáveis na saída

Porém, esse circuito (da maneira apresentada) não permite a regravação do valor armazenado

CIRCUITOS COM MEMÓRIAS

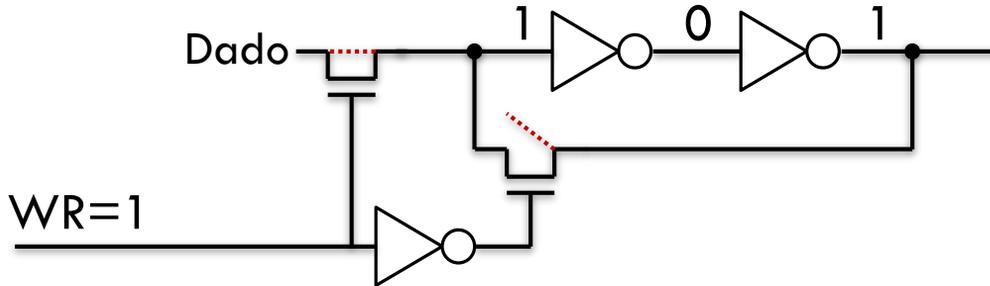
Dois interruptores (**chaves de transmissão**) são necessários, para que possamos modificar o valor armazenado

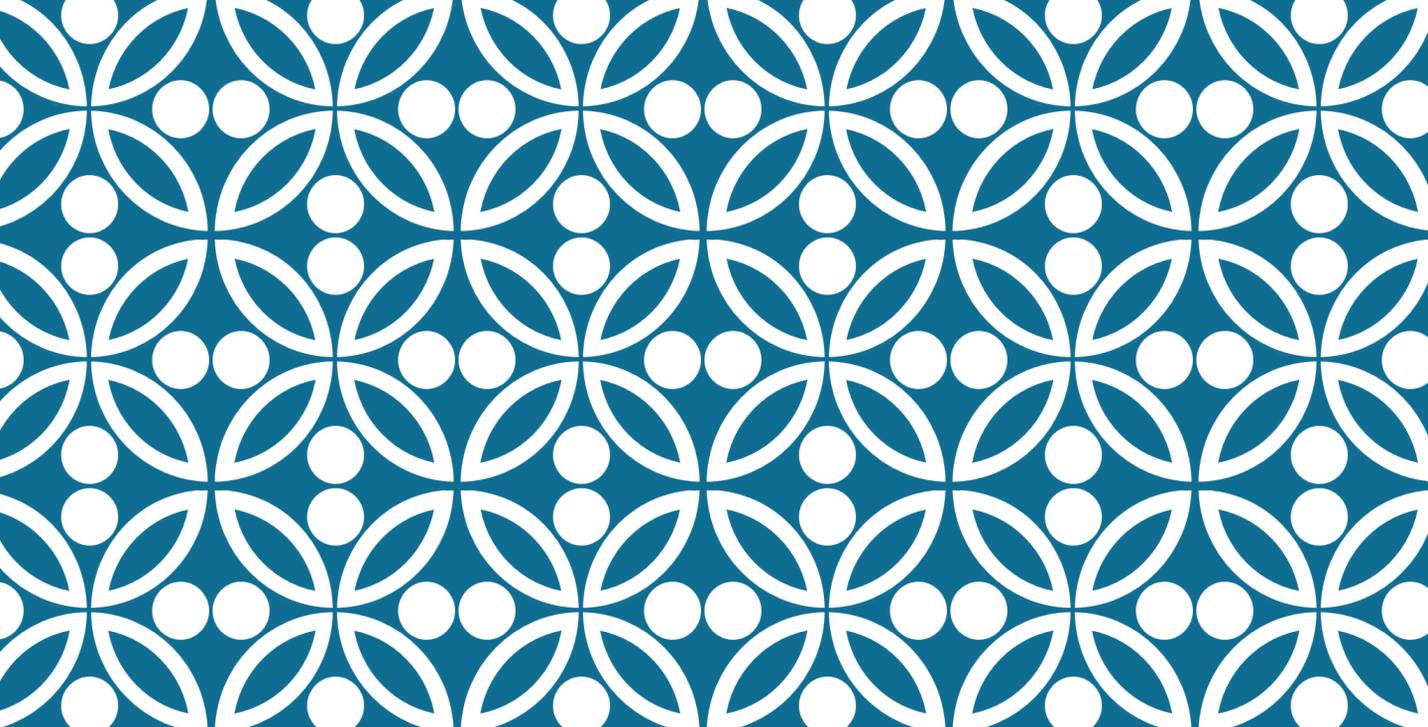


Por que devemos usar portas de transmissão?

CIRCUITOS COM MEMÓRIAS

Modo escrita habilitado

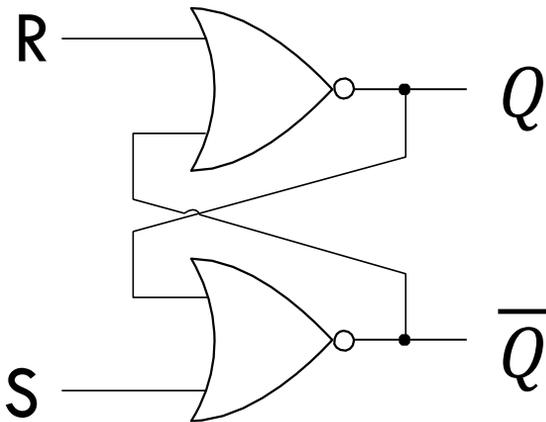




LATCHES (BÁSCULAS)

LATCH DO TIPO R-S (RESET-SET)

Circuitos sequenciais, com capacidade de memória podem ser implementados usando portas lógicas, invés de inversores e portas de transmissão



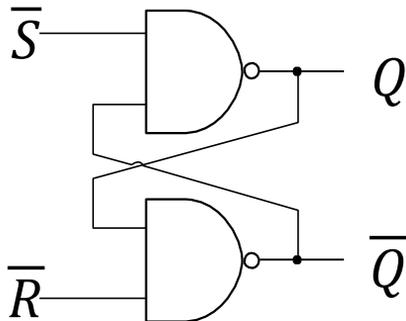
R	S	Q_i	\overline{Q}_i	
1	0	0	1	(reset Q)
0	1	1	0	(set Q)
0	0	Q_{i-1}	\overline{Q}_{i-1}	(mantém Q)
1	1	X	X	Estado Proibido

Se
implementarmos
com NAND, como
fica a tabela?

LATCH DO TIPO \overline{S} – \overline{R} COM PORTAS NAND

É possível construir um latch similar com portas NAND, mas as entradas se tornam ativas em nível baixo.

Quando o estado proibido causa problemas?

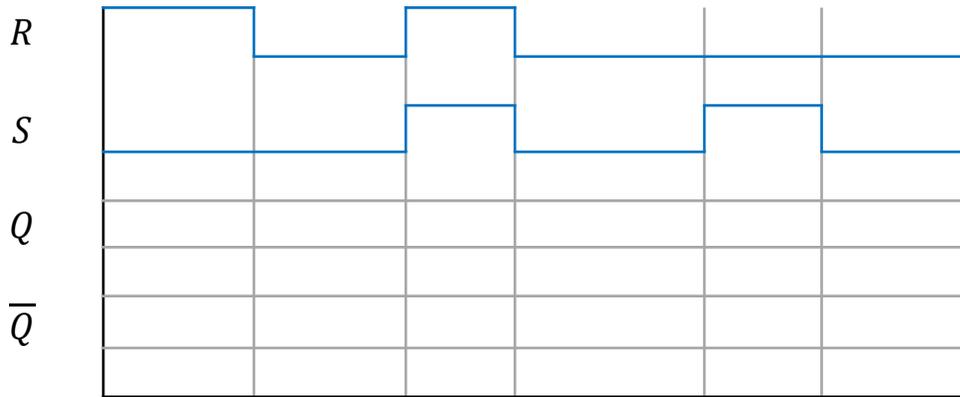


\overline{S}	\overline{R}	Q_i	\overline{Q}_i	
0	1	0	1	(reset Q)
1	0	1	0	(set Q)
0	0	Q_{i-1}	$\overline{Q_{i-1}}$	(mantém Q)
1	1	X	X	Estado Proibido

LATCH DO TIPO R-S (RESET-SET)

E o estado $R = 1, S = 1$?

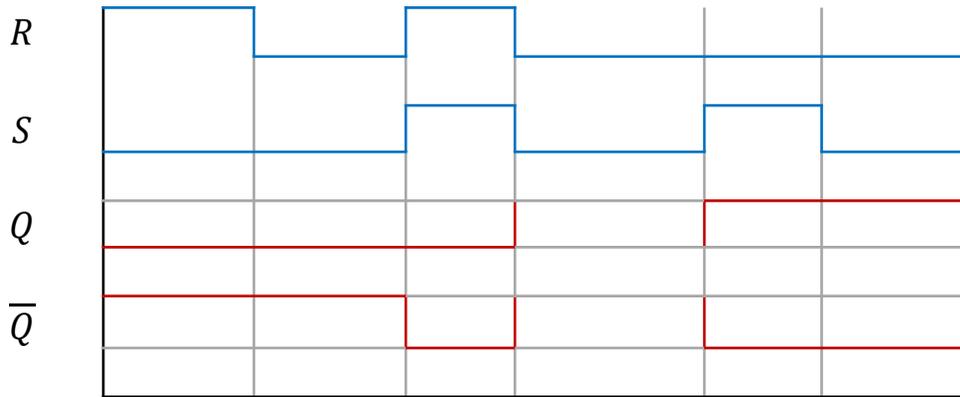
Esboce os diagramas de forma de onda para Q e \overline{Q} , considerando R e S conforme o diagrama abaixo.



LATCH DO TIPO R-S (RESET-SET)

E o estado $R = 1, S = 1$?

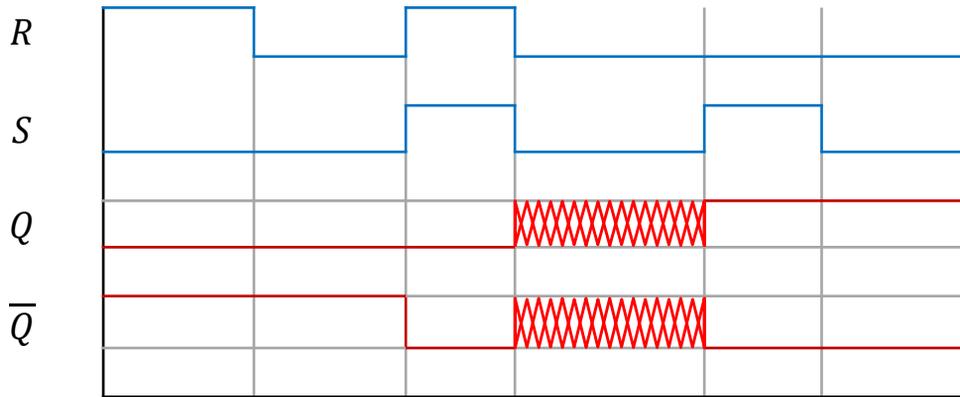
Esboce os diagramas de forma de onda para Q e \overline{Q} , considerando R e S conforme o diagrama abaixo.



LATCH DO TIPO R-S (RESET-SET)

E o estado $R = 1, S = 1$?

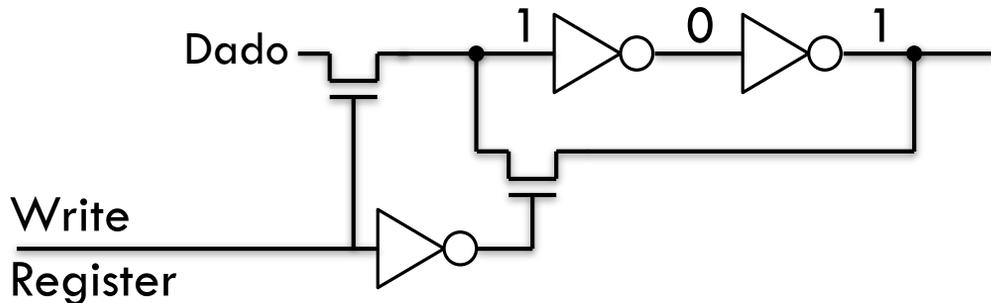
Esboce os diagramas de forma de onda para Q e \bar{Q} , considerando R e S conforme o diagrama abaixo.



Após a uma transição $R = 1, S = 1$ para $R = 0, S = 0$ as saídas ficam instáveis, só voltando ao normal após o próximo reset ou set.

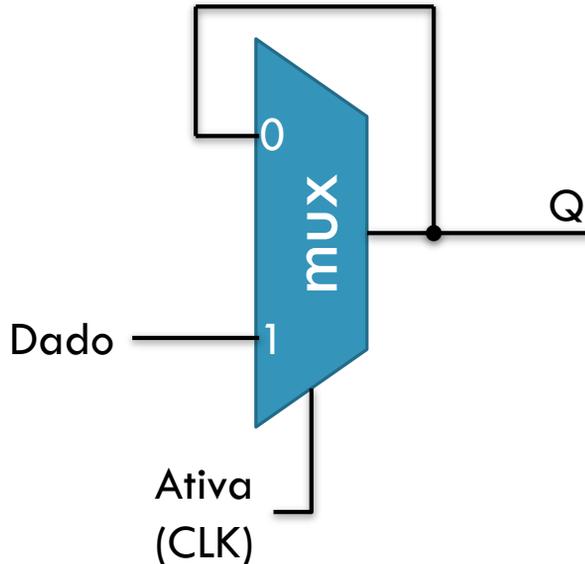
LATCHES BASEADOS EM MUX

Será que podemos implementar um latch utilizando um mux?



LATCHES BASEADOS EM MUX (LATCH NEGATIVO)

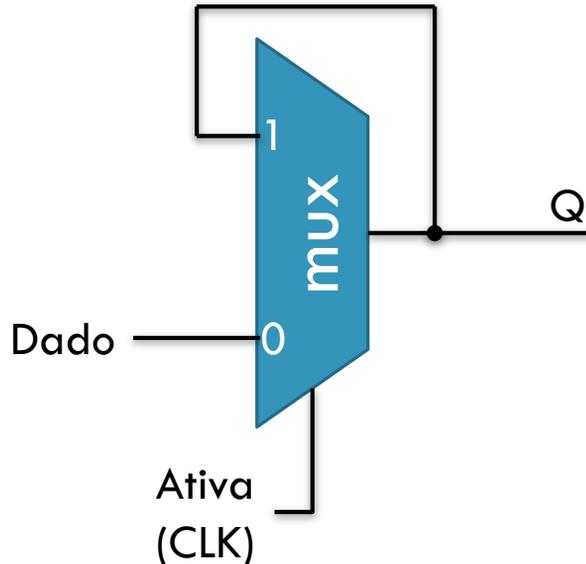
$$Q = \overline{clk} \cdot Q + clk \cdot Dado$$



Como fazer
um latch
positivo?

LATCHES BASEADOS EM MUX (LATCH POSITIVO)

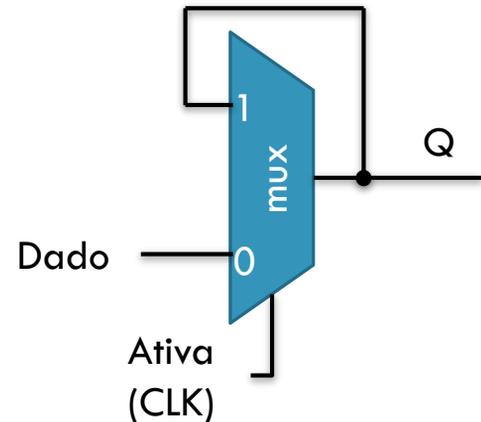
$$Q = clk \cdot Q + \overline{clk} \cdot Dado$$



LATCHES BASEADOS EM MUX

Note que o circuito viola a proibição de laços em circuitos combinacionais!

Precisamos usar um multiplexador bem comportado



LATCHES BASEADOS EM MUX

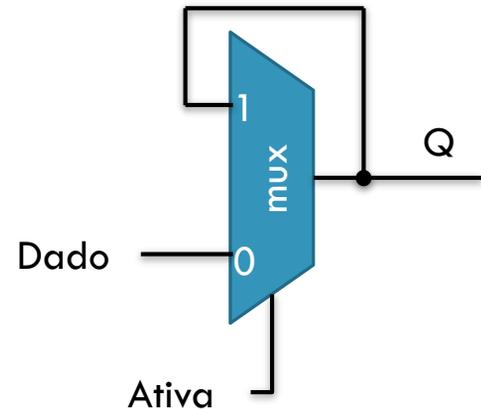
O MUX não deve capturar pulsos espúrios (delay de propagação) do Dado até a saída do MUX

Por isso o sinal para Ativar deve ser mantido ligado por tempo maior que o Tempo de Propagação do Sinal

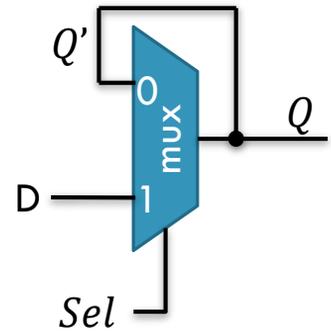
Dessa forma, a entrada Dado e a Saída Q terão valores idênticos

Logo, podemos desligar o sinal Ativa, e o valor estará armazenado

Ou seja, precisamos garantir o tempo mínimo de sinal de ativação.



LATCHES BASEADOS EM MUX



Para garantir um comportamento confiável do latch baseado em mux, devemos garantir algumas condições durante a escrita:

1. $Sel = 1, D = \mu, \mu \in \mathcal{B}$: Devemos manter este estado por tempo maior que T_p , até a saída $Q = \mu$
2. $D = Q' = \mu$: Mantidos válidos por tempo maior que T_p : as duas entradas do mux serão idênticas e portanto o valor do Sel não importa
3. $Sel = 0, D = Q' = \mu$: Mantidos válidos por tempo maior que T_p : $Q = \mu$ e o valor de D não importa. Q' é escolhido como a entrada do MUX. Mantém o valor em Q enquanto $Sel = 0$ independente do que ocorra em D.

LATCHES

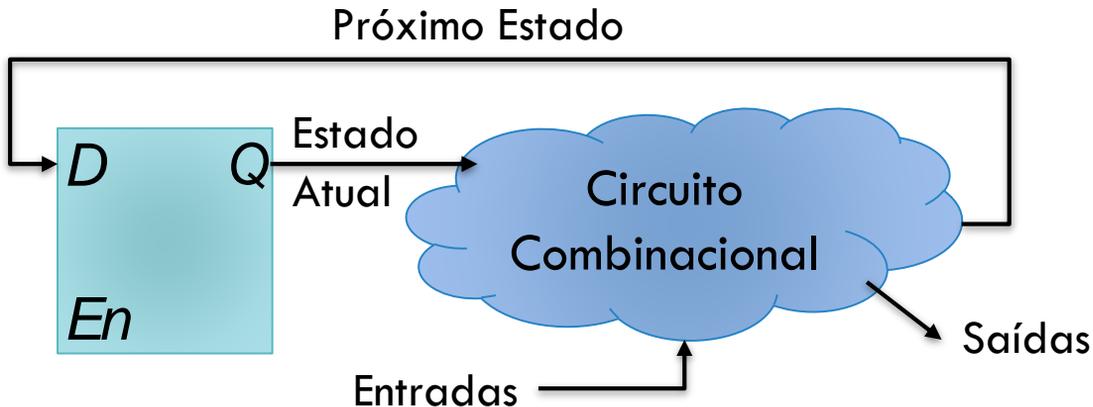
Latches são formas bastante simples de armazenar dados utilizando apenas duas portas lógicas (pequenos e rápidos)

O problema é que os latches aceitam escrita a qualquer momento (não apenas nas bordas do relógio)

Isso faz com que nosso circuito seja mais propenso a erros

LATCHES

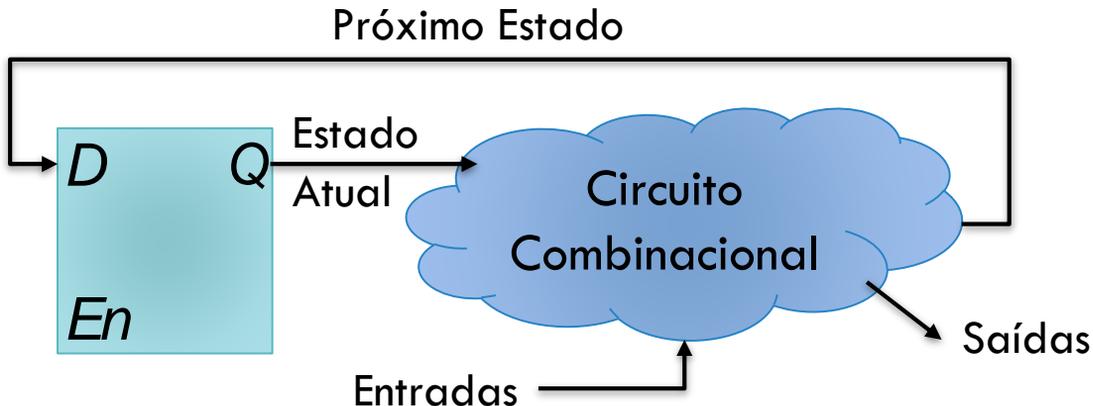
Quando $En=1$, alguns bits do Próximo Estado atravessam o latch...
que atravessam o circuito combinacional... que atravessam o latch...

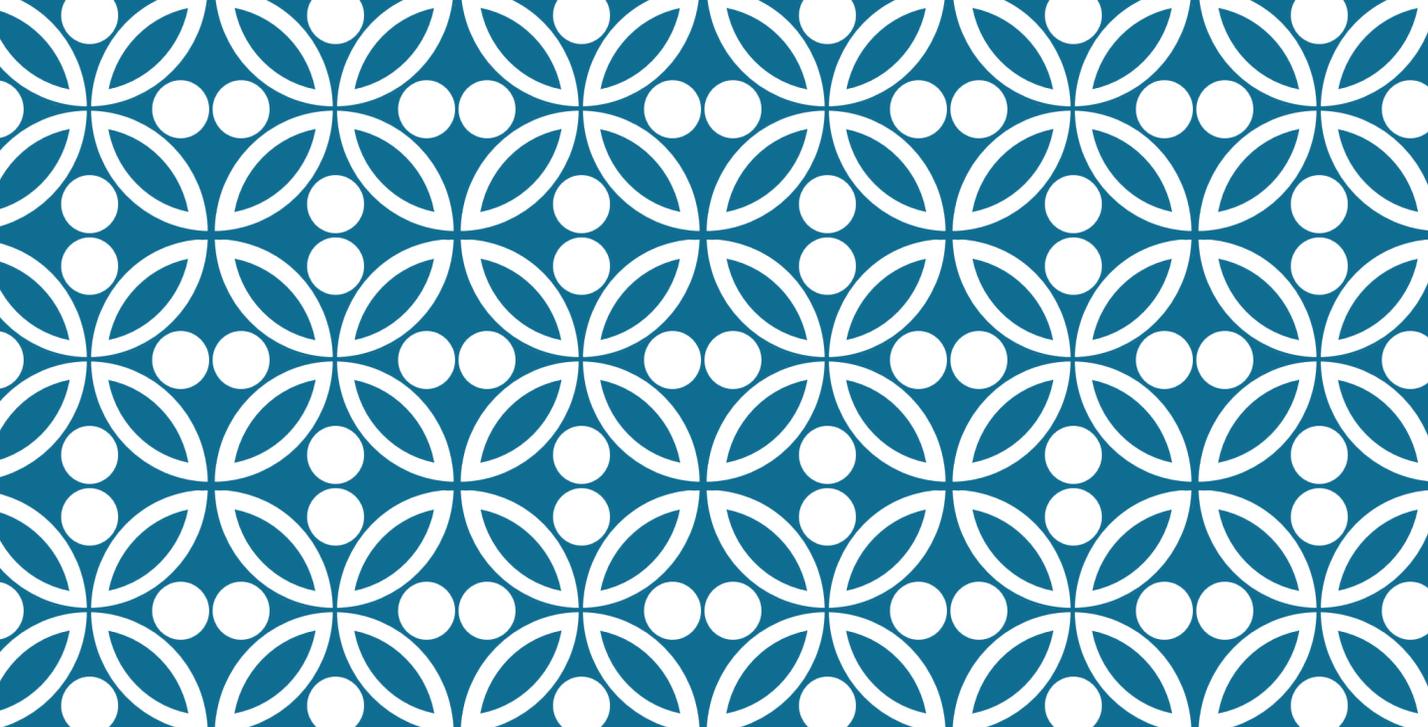


LATCHES

Quando $En=1$, alguns bits do Próximo Estado atravessam o latch...
que atravessam o circuito combinacional... que atravessam o latch...

Precisamos fazer uma janela em que o latch está transparente, o
mais estreita possível (para quebrar esse ciclo)





FLIP-FLOPS

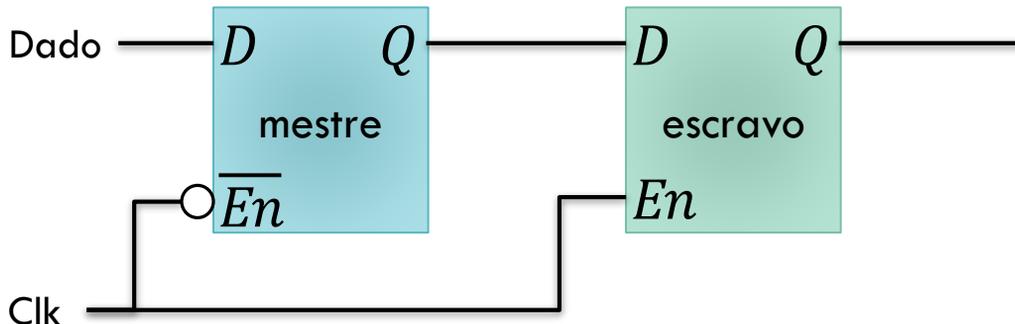
FLIP-FLOPS

Composto de dois latches, um mestre e o escravo

Quando o mestre está transparente o escravo mantém o valor antigo

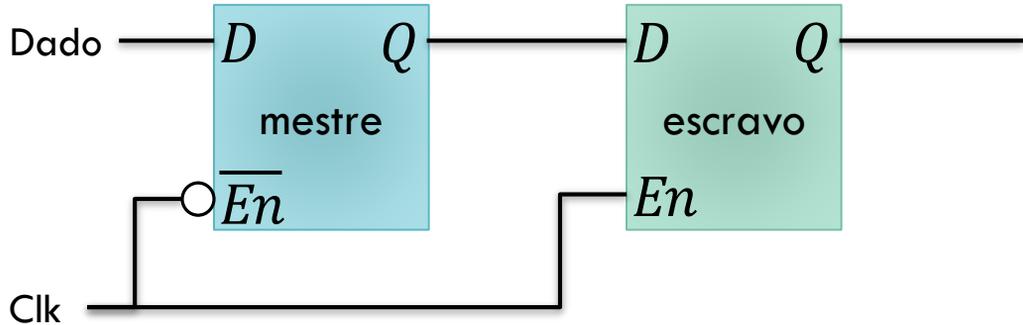
Quando o mestre mantém, o escravo fica transparente e atualiza a saída

A transição $0 \rightarrow 1$ em CLK captura o valor para armazenar



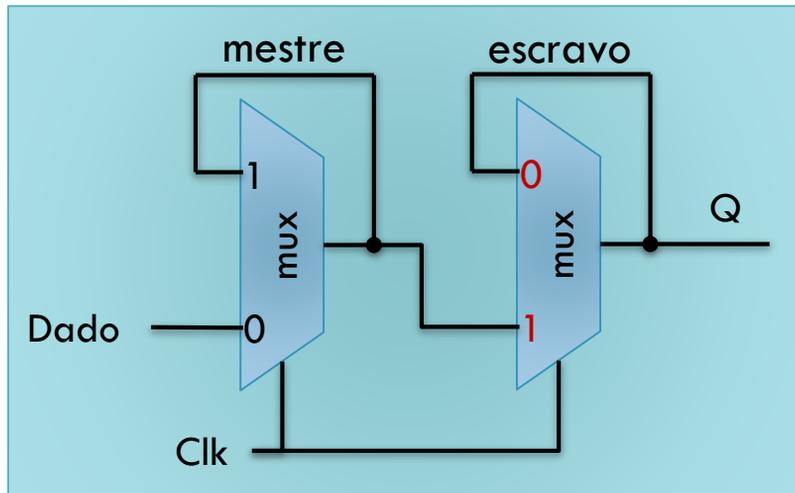
FLIP-FLOPS COM MUXES

Como criar flip-flops com muxes?



FLIP-FLOPS COM MUXES

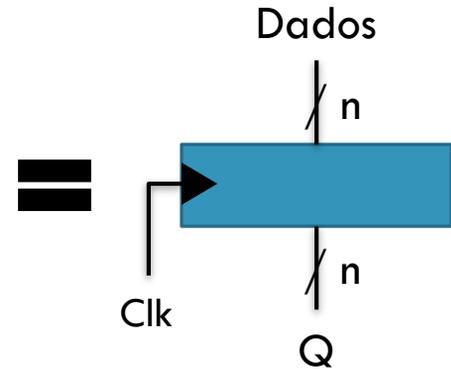
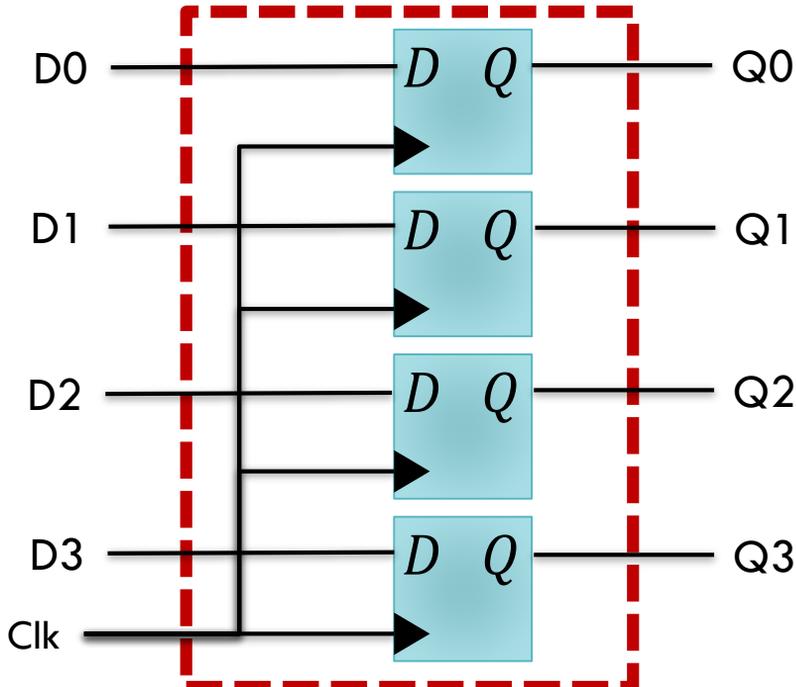
Como criar flip-flops com muxes?



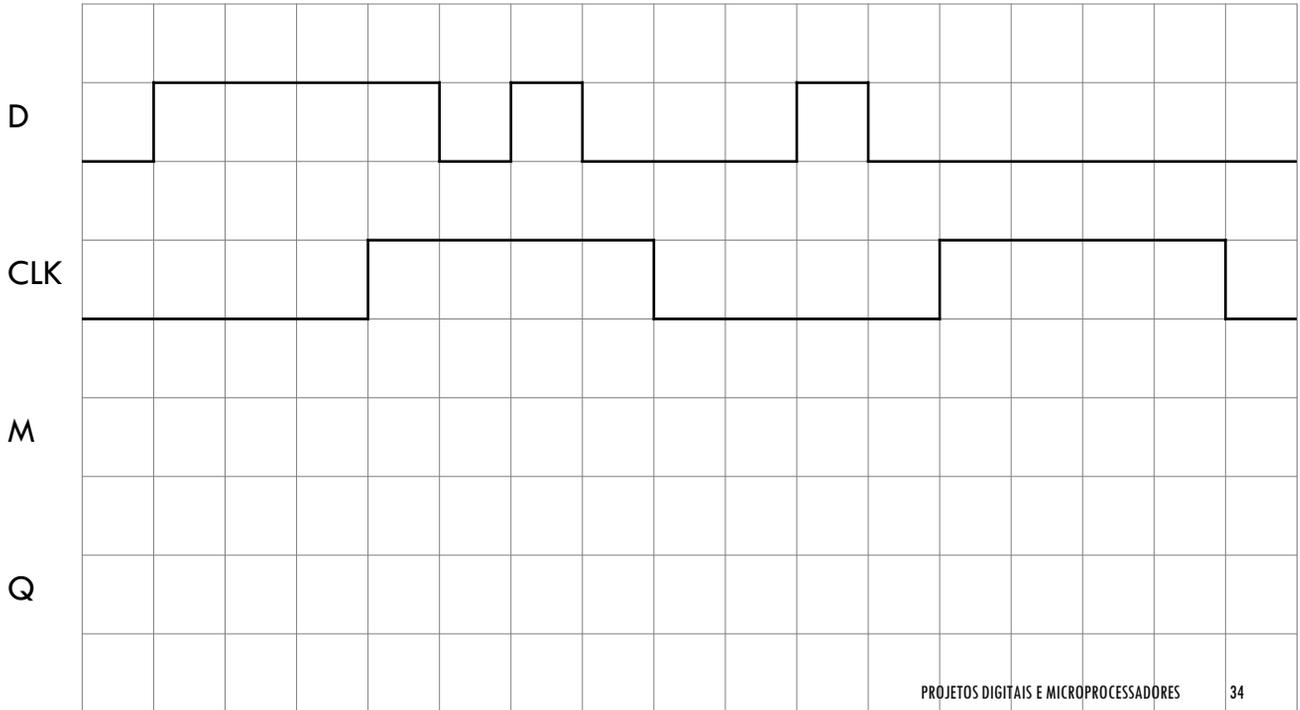
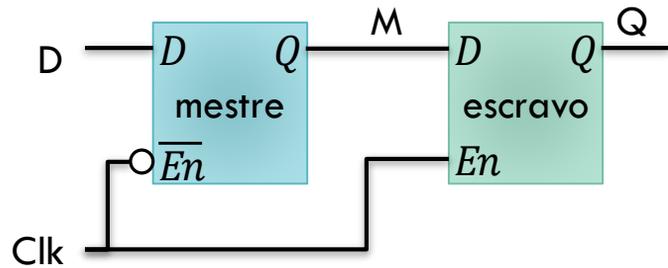
REGISTRADORES

Vários FF's que compartilham o mesmo sinal de relógio

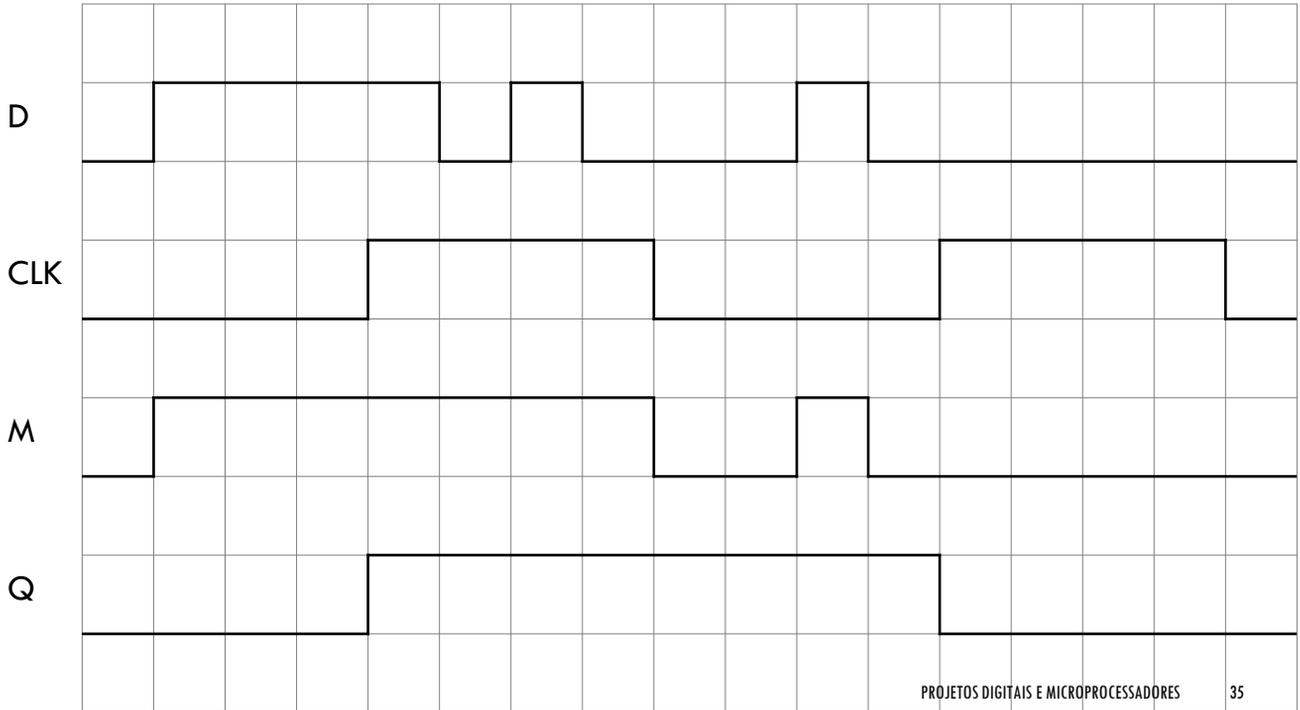
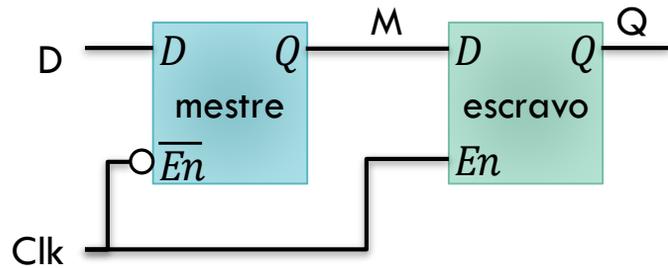
Registrador



FF TIPO D SENSÍVEL A BORDA DE SUBIDA

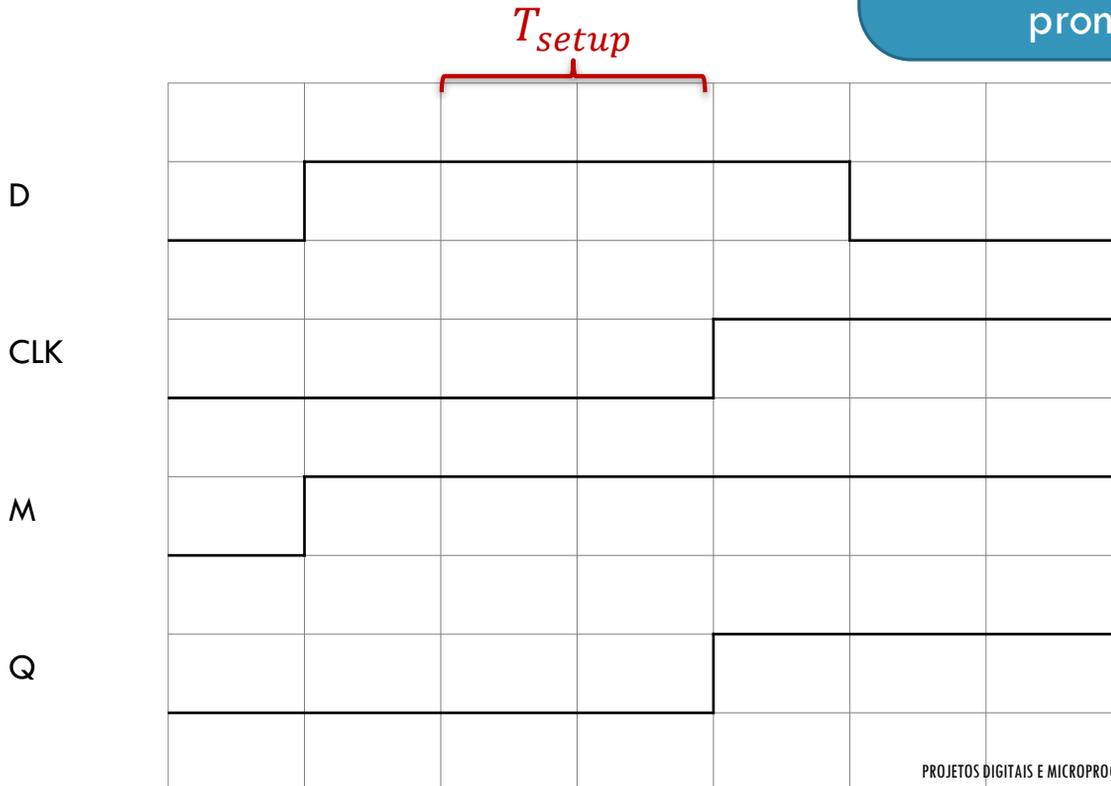


FF TIPO D SENSÍVEL A BORDA DE SUBIDA



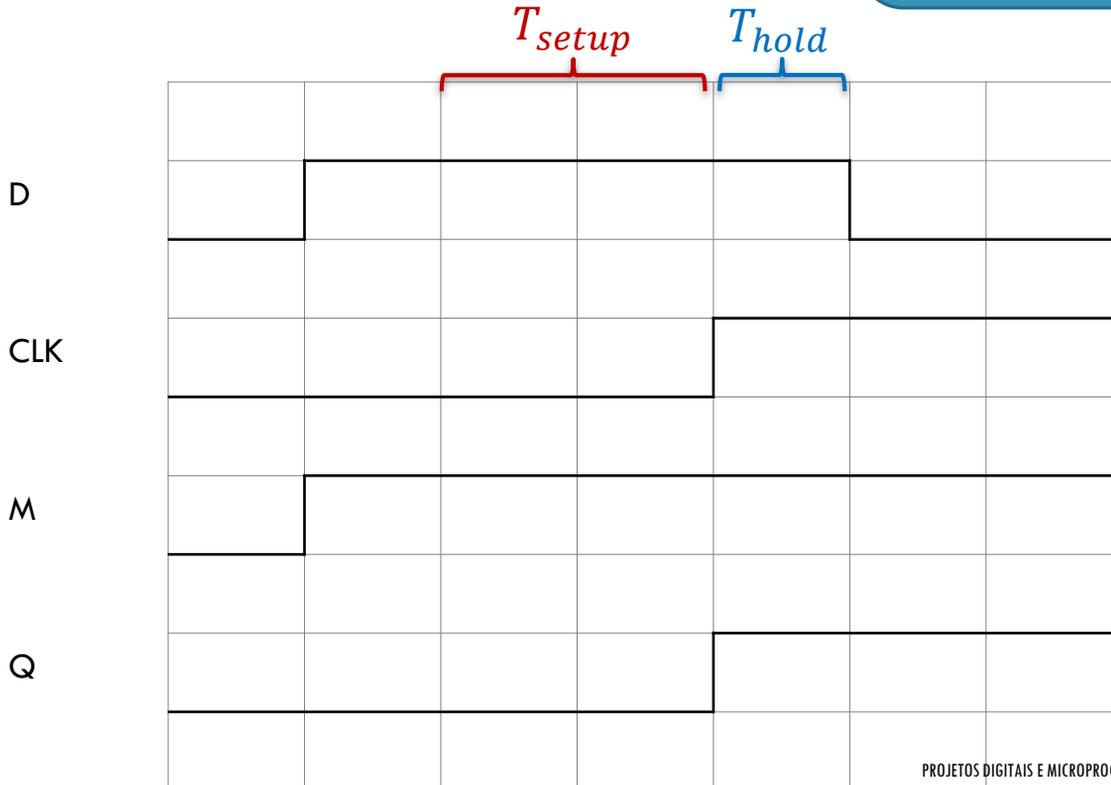
TEMPO DE SETUP E HOLD

Quanto tempo antes da borda do relógio as entradas devem estar com seus valores prontos?

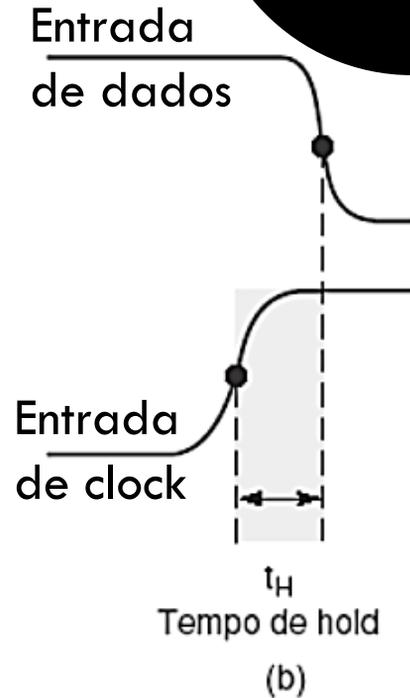
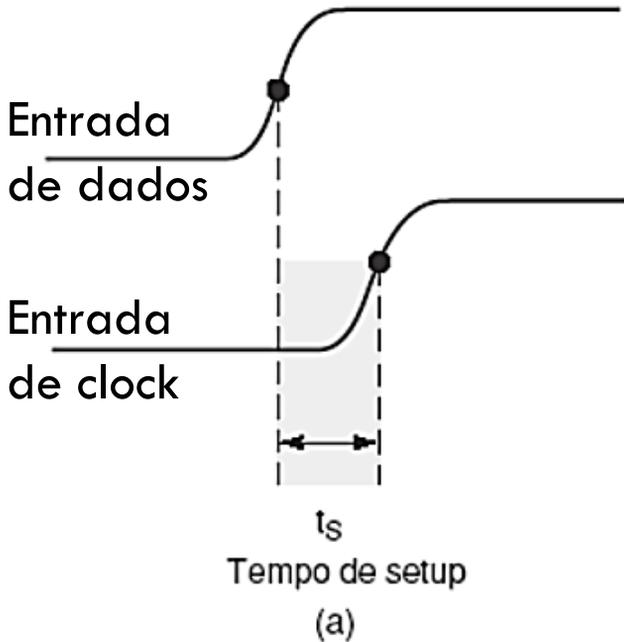


TEMPO DE SETUP E HOLD

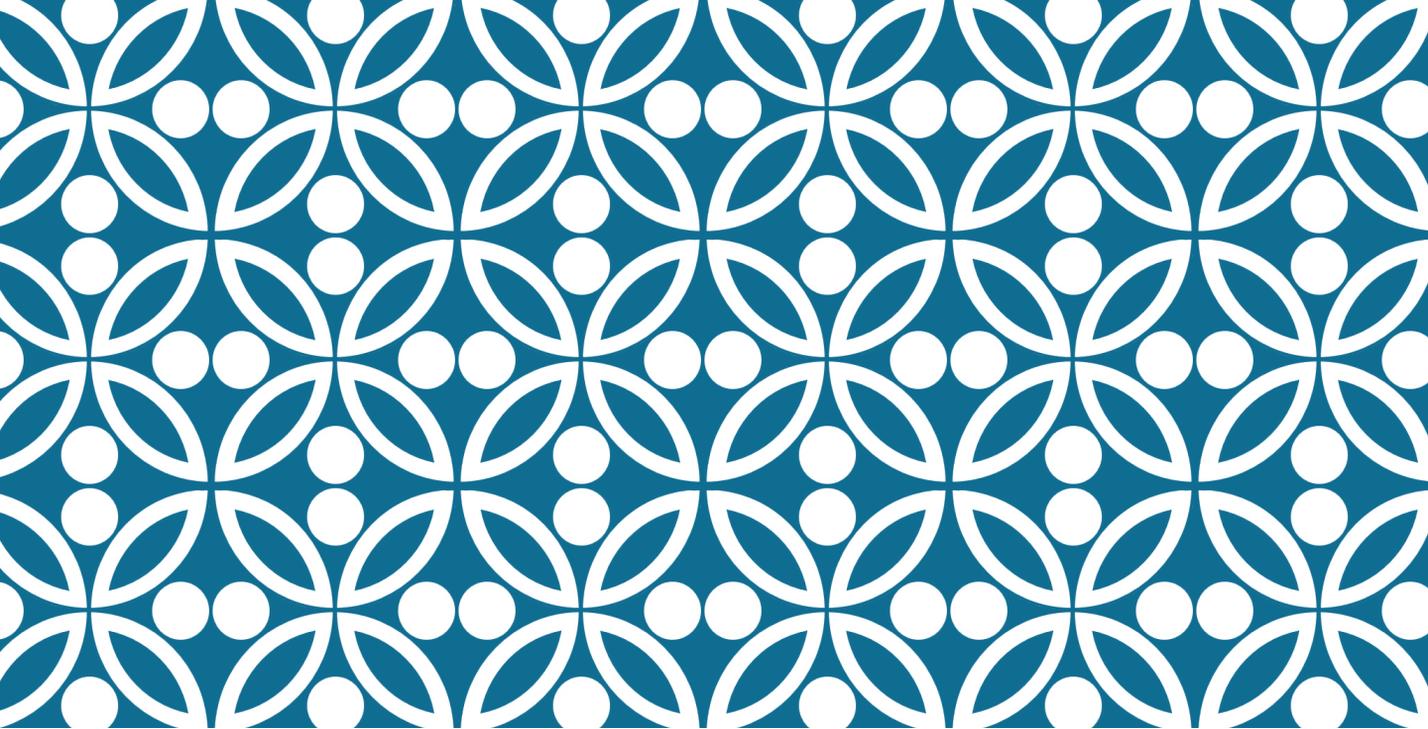
Quanto após a borda do relógio as entradas podem mudar seus valores?



TEMPO DE SETUP E HOLD



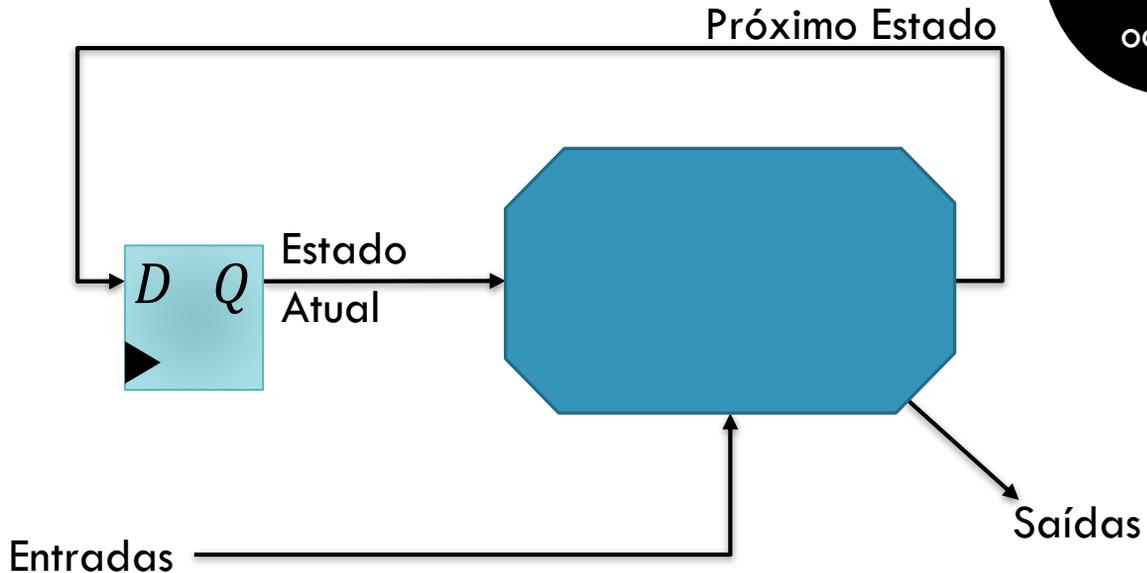
Lembre-se que as transições não acontecem de forma imediata



CIRCUITOS SEQUENCIAIS

CIRCUITOS SEQUENCIAIS SÍNCRONOS

Novos
atrasos
vão
ocorrer



ATRASOS

Circuitos Combinacionais

T_{PC} Tempo de propagação do circuito combinacional

T_{CC} Tempo de contaminação do circuito combinacional

ATRASOS

Circuitos Combinacionais

T_{PC} Tempo de propagação do circuito combinacional

T_{CC} Tempo de contaminação do circuito combinacional

Circuitos Sequenciais

T_{PS} Tempo de propagação (**programação**) do circuito sequencial (registrador)

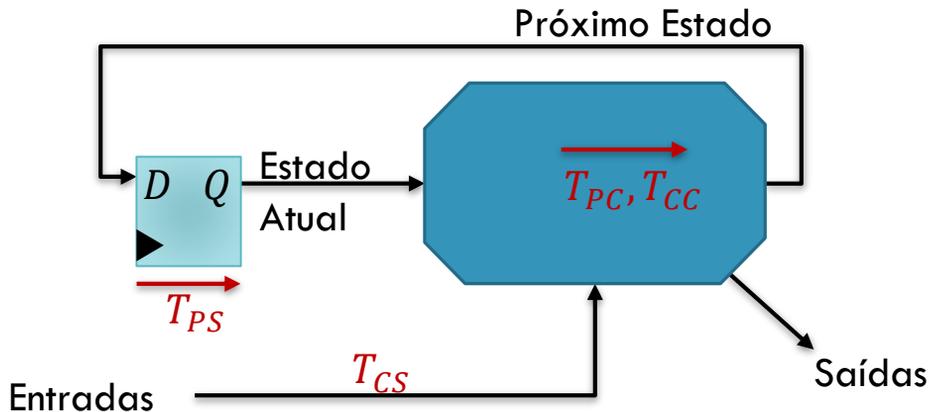
T_{CS} Tempo de **contaminação** do circuito sequencial (registrador)

T_{Setup} – Trata-se do intervalo imediatamente **antes da borda onde a entrada não pode variar**

T_{Hold} – Trata-se do intervalo imediatamente **após a borda onde a entrada não pode variar**

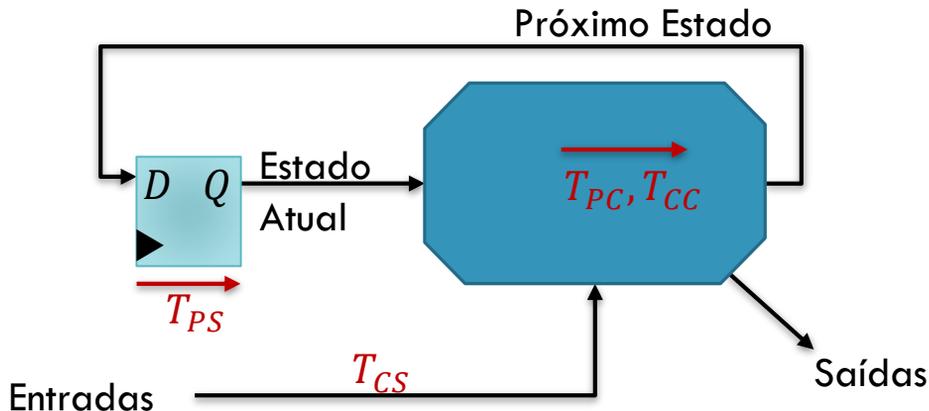
CIRCUITOS SEQUENCIAIS SÍNCRONOS

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$



CIRCUITOS SEQUENCIAIS SÍNCRONOS

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$
$$T_{CS} + T_{CC} \geq T_{Hold} \Rightarrow \text{Evitar ciclo}$$

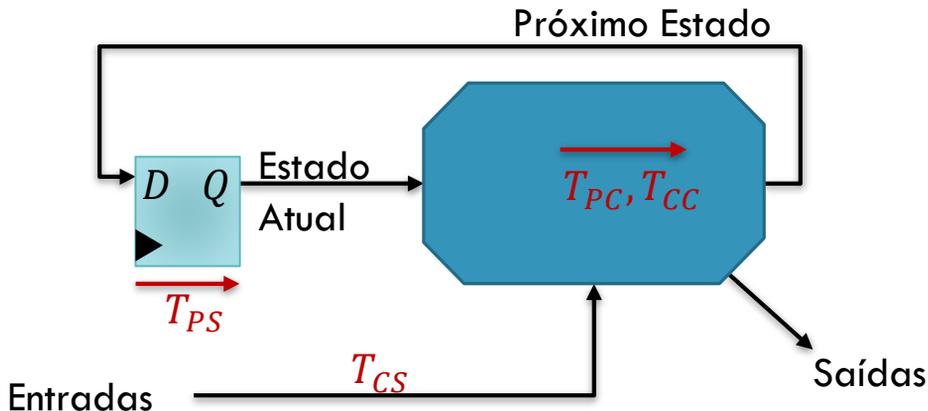


CIRCUITOS SEQUENCIAIS SÍNCRONOS

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$
$$T_{CS} + T_{CC} \geq T_{Hold} \Rightarrow \text{Evitar ciclo}$$

As entradas (externas) devem respeitar os tempo de Setup e Hold

$$T_{SetupEntrada} \geq T_{PS} + T_{Setup}$$
$$T_{HoldEntrada} \geq T_{Hold} - T_{CC}$$



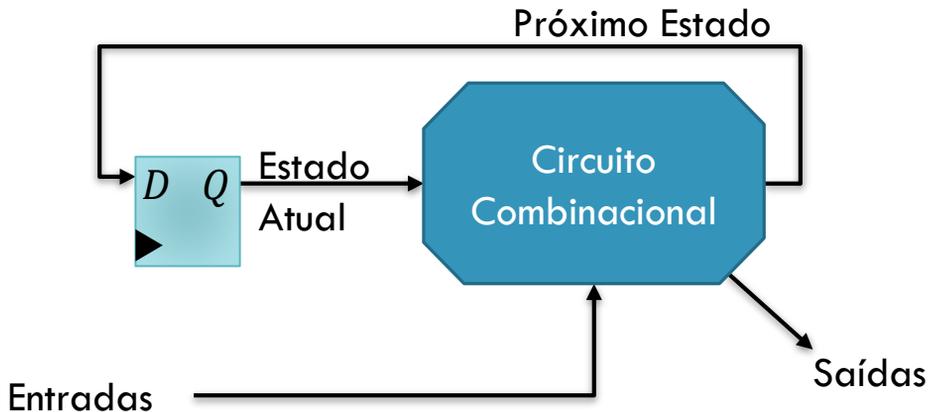
EXERCÍCIO 1

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$
$$T_{CS} + T_{CC} \geq T_{Hold}$$
$$T_{SetupEntrada} \geq T_{PS} + T_{Setup}$$
$$T_{HoldEntrada} \geq T_{Hold} - T_{CC}$$

$$T_{CS} = 1ns$$
$$T_{PS} = 3ns$$
$$T_{Setup} = 2ns$$
$$T_{Hold} = 2ns$$

$$T_{CC} = ?$$
$$T_{PC} = 5ns$$

Qual T_{clock} ?



EXERCÍCIO 1

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$

$$T_{CS} + T_{CC} \geq T_{Hold}$$

$$T_{SetupEntrada} \geq T_{PS} + T_{Setup}$$

$$T_{HoldEntrada} \geq T_{Hold} - T_{CC}$$

$$T_{CS} = 1ns$$

$$T_{PS} = 3ns$$

$$T_{Setup} = 2ns$$

$$T_{Hold} = 2ns$$

$$T_{CC} = ?$$

$$T_{PC} = 5ns$$

Qual T_{clock} ?

$$T_{CS} + T_{CC} \geq T_{Hold} \Rightarrow$$

$$\Rightarrow 1ns + T_{CC} \geq 2ns$$

$$\Rightarrow T_{CC} = 1ns$$

$$T_{Clock} \geq T_{PS} + T_{PC} + T_{Setup} \Rightarrow$$

$$\Rightarrow 3 + 5 + 2 = 10ns$$

$$\Rightarrow Freq \leq 100MHz$$

$$T_{SetupEnt} = T_{PC} + T_{Setup}$$

$$= 5 + 2 = 7ns$$

$$T_{HoldEnt} = T_{Hold} - T_{CC}$$

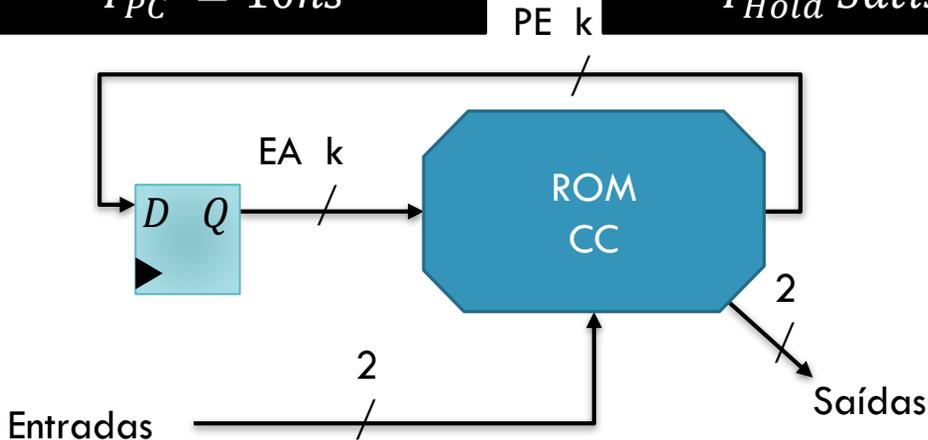
$$= 2 - 1 = 1ns$$

EXERCÍCIO 2

$T_{CS} = 0,5ns$
 $T_{PS} = 2ns$
 $T_{Setup} = 2ns$
 $T_{Hold} = 1ns$
 $T_{CC} = 2ns$
 $T_{PC} = 10ns$

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$
$$T_{CS} + T_{CC} \geq T_{Hold}$$
$$T_{SetupEntrada} \geq T_{PS} + T_{Setup}$$
$$T_{HoldEntrada} \geq T_{Hold} - T_{CC}$$

Qual a *altura da ROM*?
Qual a *largura da ROM*?
Qual T_{clock} ?
Qual $T_{SetupEntrada}$?
Qual $T_{HoldEntrada}$?
 T_{Hold} Satisfeito?



EXERCÍCIO 2

$$T_{CS} = 0,5ns$$

$$T_{PS} = 2ns$$

$$T_{Setup} = 2ns$$

$$T_{Hold} = 1ns$$

$$T_{CC} = 2ns$$

$$T_{PC} = 10ns$$

$$T_{Clock} > T_{PS} + T_{PC} + T_{Setup}$$

$$T_{CS} + T_{CC} \geq T_{Hold}$$

$$T_{SetupEntrada} \geq T_{PS} + T_{Setup}$$

$$T_{HoldEntrada} \geq T_{Hold} - T_{CC}$$

Qual a *altura da ROM*?
Qual a *largura da ROM*?

Qual T_{clock} ?

Qual $T_{SetupEntrada}$?

Qual $T_{HoldEntrada}$?

T_{Hold} Satisfeito?

$$T_{Clock} \geq T_{PS} + T_{PC} + T_{Setup} \geq 2 + 10 + 2 = 14ns$$

$$T_{SetupEnt} \geq T_{PC} + T_{Setup} \geq 10 + 2 = 12ns$$

$$T_{HoldEnt} \geq T_{Hold} - T_{CC} \geq 1 - 2 = -1ns \text{ (sobra 1ns)}$$

$$T_{CS} + T_{CC} \geq T_{Hold} \Rightarrow 0,5 + 2 \geq 1ns$$