

["Ao contrário", continuou Tweedledee, "se foi assim, poderia ser; e se fosse assim, seria; mas como não é, então não é. Isso é Lógico."](Lewis Carroll, Through the Looking-glass: And what Alice Found There, 1875).

# Latches

Paulo Ricardo Lisboa de Almeida

# Relembrando

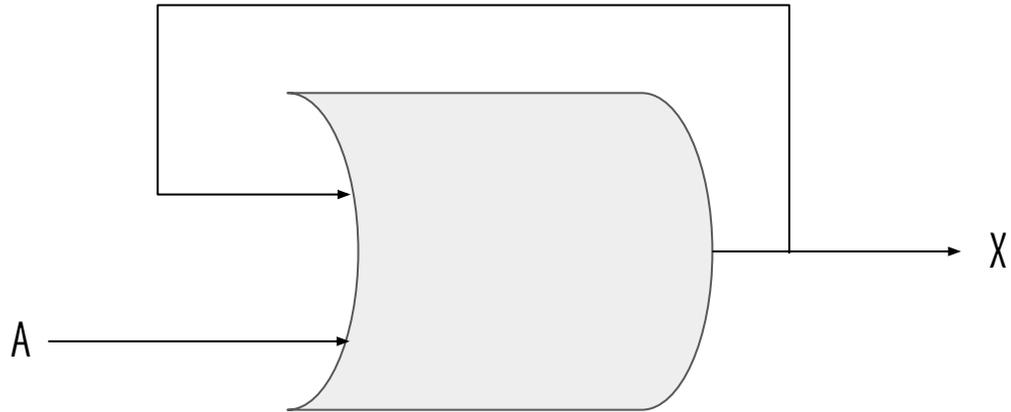
**Circuito combinacional:** não possui “memória”. A saída depende exclusivamente da entrada

# Combinacional versus Sequencial

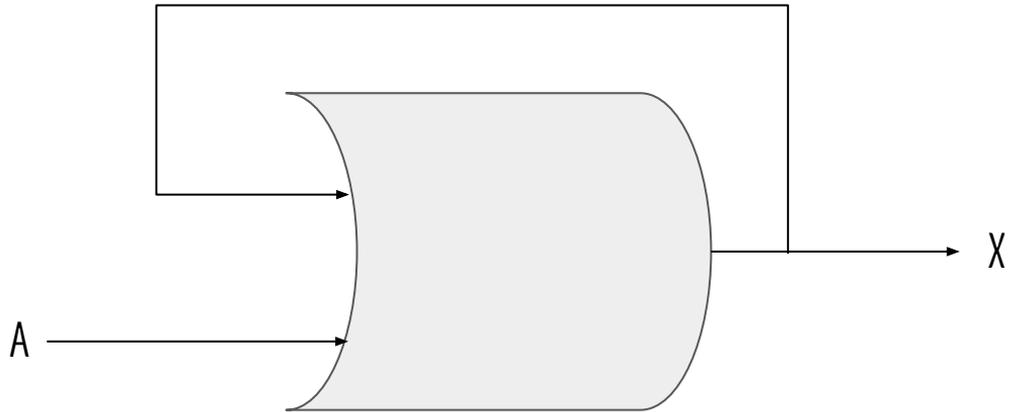
**Circuito combinacional:** não possui “memória”. A saída depende exclusivamente da entrada

**Circuito sequencial:** possuem uma memória interna. A saída depende da entrada e do estado da memória.

# Exemplo

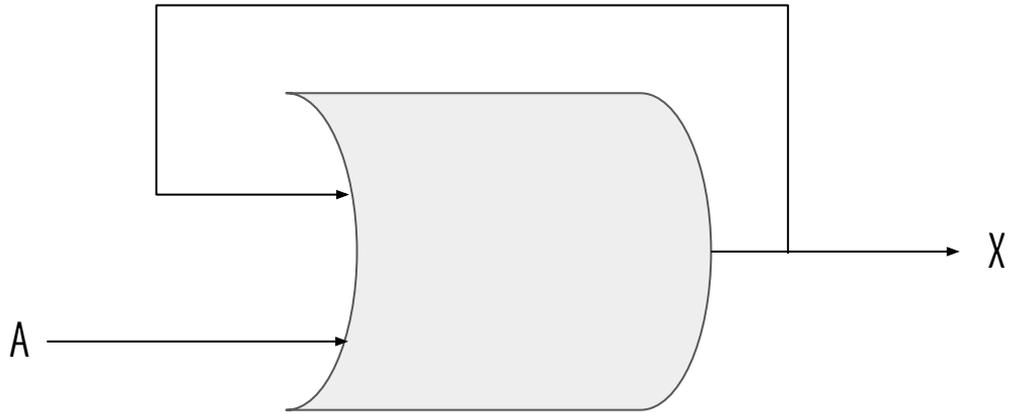


# Exemplo



| $A$ | $X_{t-1}$ | $X_t$ |
|-----|-----------|-------|
| 0   | 0         |       |
| 0   | 1         |       |
| 1   | 0         |       |
| 1   | 1         |       |

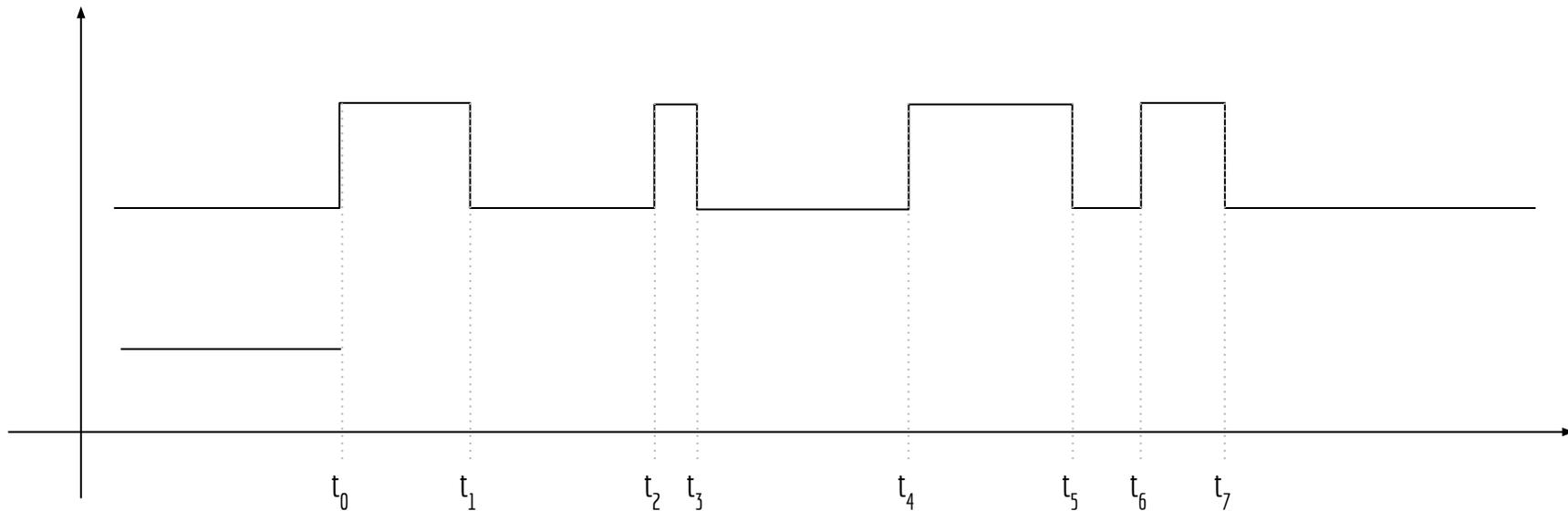
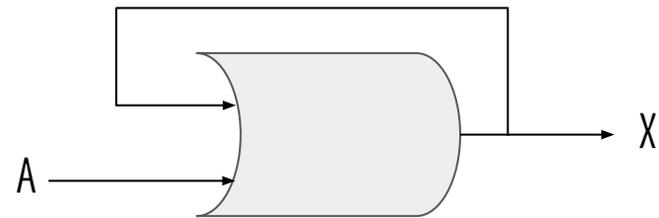
# Exemplo



| A | $X_{t-i}$ | $X_t$ |
|---|-----------|-------|
| 0 | 0         | 0     |
| 0 | 1         | 1     |
| 1 | 0         | 1     |
| 1 | 1         | 1     |

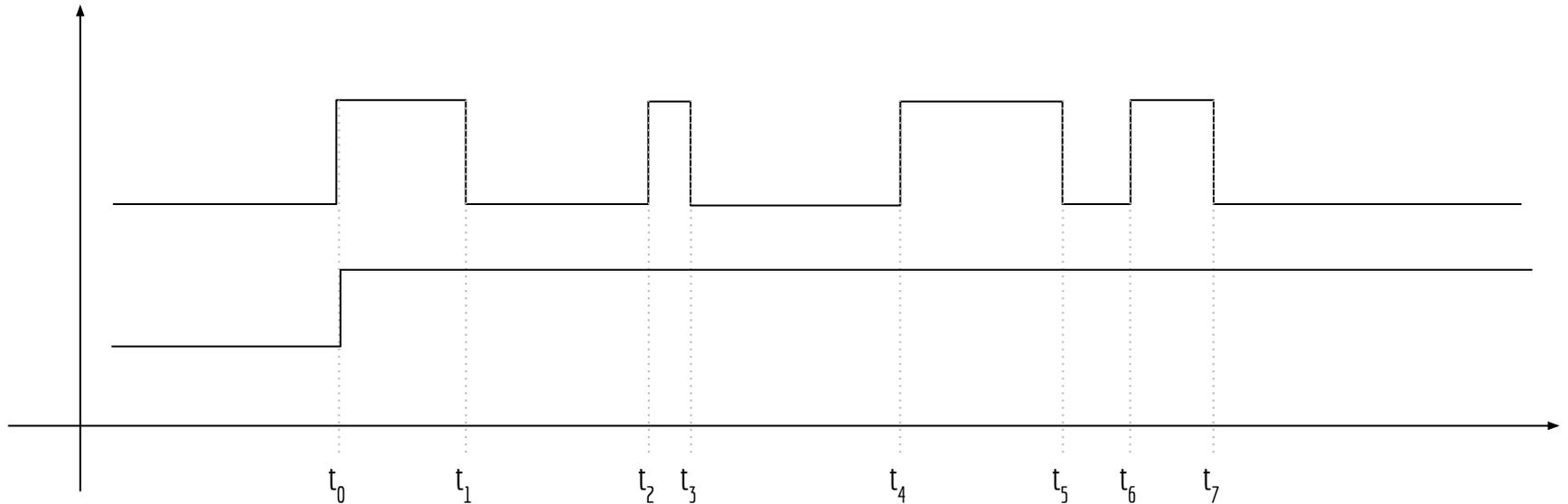
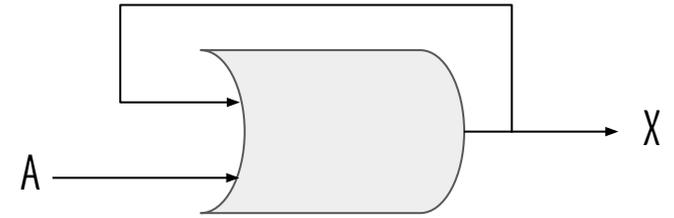
# Faça você mesmo

Faça o diagrama de temporização para a saída X



# Faça você mesmo

Faça o diagrama de temporização para a saída X



# Feedback

Circuitos sequenciais geralmente apresentam algum tipo de feedback (retroalimentação)



# Latches

**Latch:** Circuito básico de memória.

Circuito sequencial

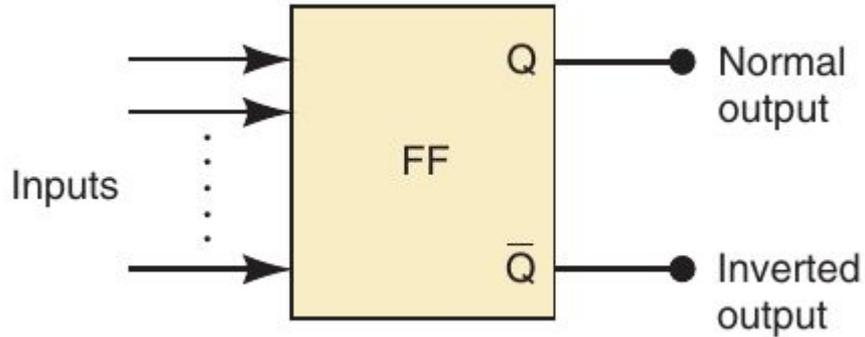
Tipo de **Multivibrador biestável**

Possuem dois estados estáveis

0 ou 1

# Latch

Diagrama geral de um Latch



# Latch

## Duas saídas

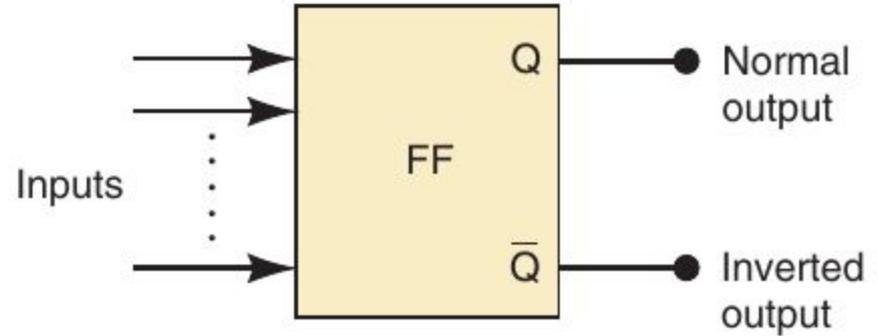
Saída normal  $Q$ , e a saída invertida  $\bar{Q}$

Quando  $Q=1$  e  $\bar{Q}=0$  o latch está em **set**.

Ao inserir entradas que fazem com que  $Q=1$  estamos **setando** o latch.

Quando  $Q=0$  e  $\bar{Q}=1$  o latch está em **reset**.

Ao inserir entradas que fazem com que  $Q=0$  estamos **resetando** o latch.



# Latch S R

Latch **S**et **R**eset

Tipo de latch com duas entradas

Uma **S** e uma **R**

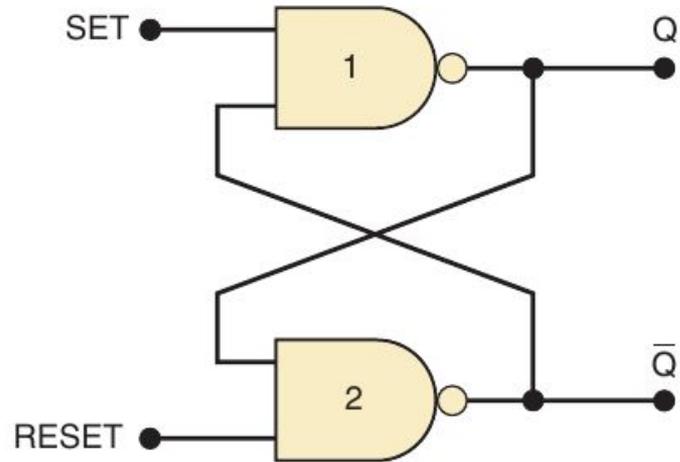
Construído com NANDs ou NORs

# Latch com NANDs

Construído com duas portas NAND ligadas com feedback

Set e reset normalmente estão em **nível lógico alto (1)**

**Exceto** quando desejamos trocar o estado do latch



# Setando o Latch

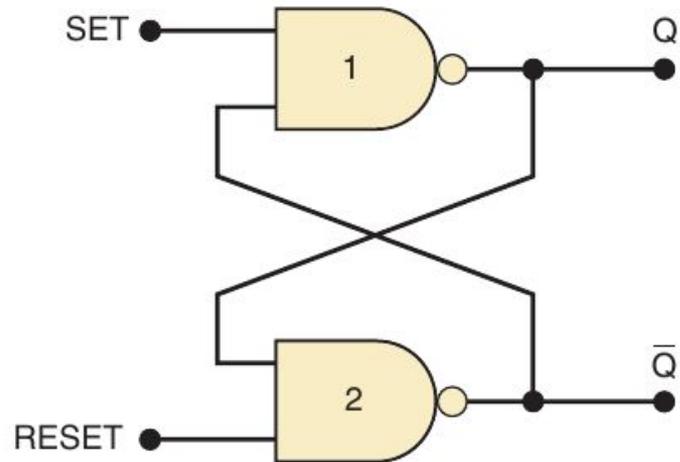
O set é feito através de um **pulso de nível lógico baixo (0)** na entrada Set

Reset é mantido em alto

Temos dois comportamentos

Um caso  $Q$  fosse 0 antes do pulso

Um caso  $Q$  fosse 1 antes do pulso



# Caso $Q$ fosse 0 antes do pulso

Entre  $t_0$  e  $t_1$  enviamos um pulso de nível baixo em set

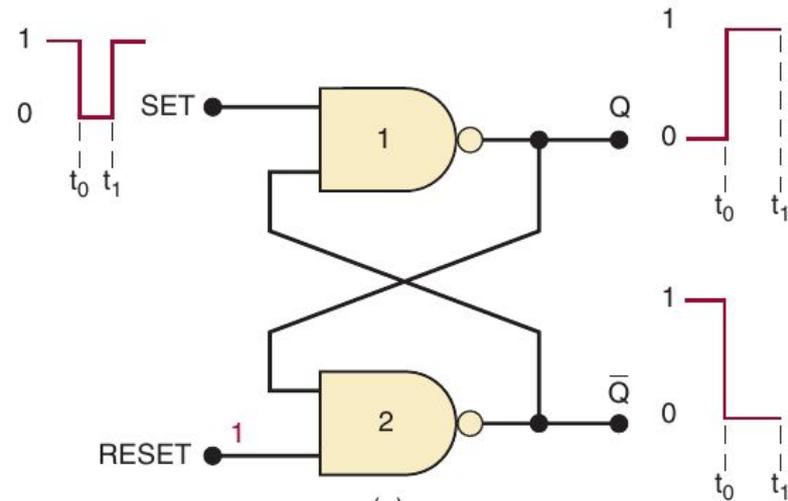
O NAND-1 recebe 0 do set, e 1 de  $\bar{Q}$ , logo  $Q=1$

O NAND-2 recebe 1 do RESET e 1 de  $Q$ , logo  $\bar{Q}=0$

Após  $t_1$  o nível em set volta para alto

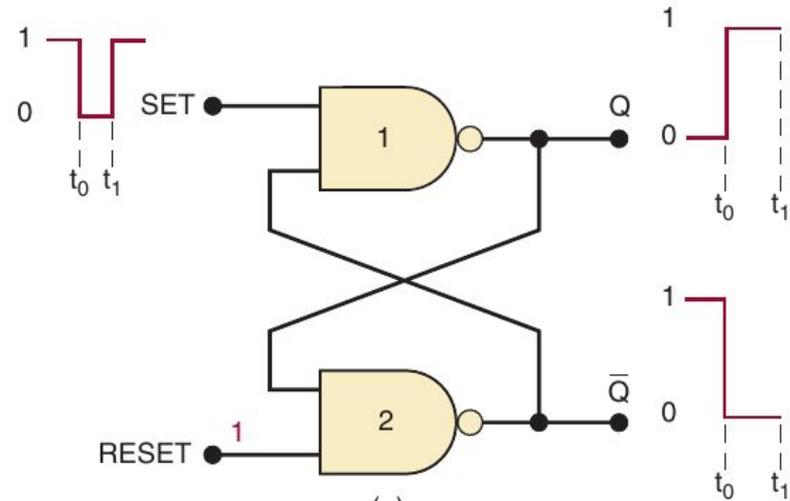
O NAND-1 recebe 1 do set, e 0 de  $\bar{Q}$ , logo  $Q=1$

O NAND-2 recebe 1 do RESET e 1 de  $Q$ , logo  $\bar{Q}=0$



# Faça você mesmo

Faça a mesma análise anterior, mas agora considere que o estado do latch era  $Q = 1$  antes do pulso



# Caso $Q$ fosse 1 antes do pulso

Entre  $t_0$  e  $t_1$  enviamos um pulso de nível baixo em set

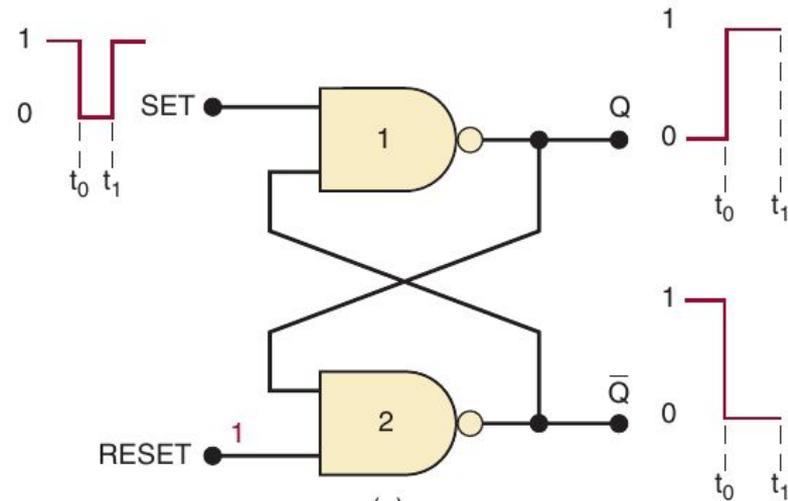
O NAND-1 recebe 0 do set, e 1 de  $\bar{Q}$ , logo  $Q$  permanece em 1

O NAND-2 recebe 1 do RESET e 1 de  $Q$ , logo  $\bar{Q}$  permanece em 0

Após  $t_1$  o nível em set volta para alto

O NAND-1 recebe 1 do set, e 0 de  $\bar{Q}$ , logo  $Q=1$

O NAND-2 recebe 1 do RESET e 1 de  $Q$ , logo  $\bar{Q}=0$



# Resetando o Latch

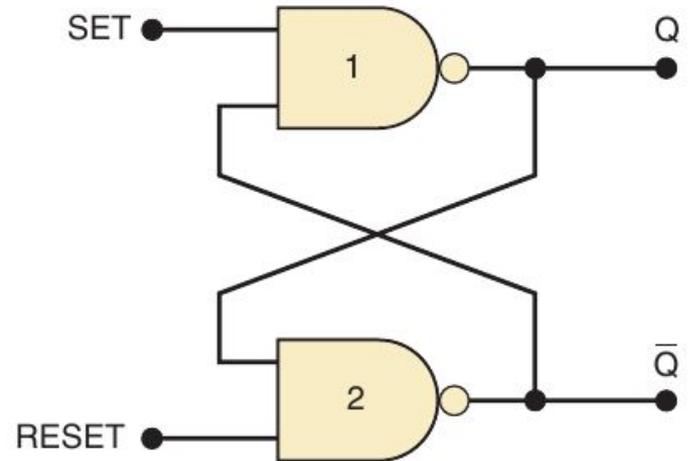
O set é feito através de um **pulso de nível lógico baixo (0)** na entrada **Reset**

Set é mantido em alto

Temos dois comportamentos

Um caso  $Q$  fosse 0 antes do pulso

Um caso  $Q$  fosse 1 antes do pulso



# Caso $Q$ fosse 0 antes do pulso

Entre  $t_0$  e  $t_1$  enviamos um pulso de nível baixo em **Reset**

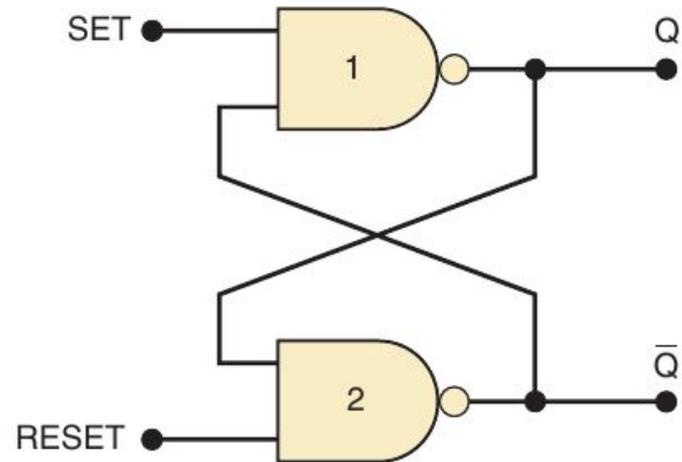
O NAND-2 recebe 0 do RESET e 0 de  $Q$ , logo  $\bar{Q}$  permanece em 1

O NAND-1 recebe 1 do set, e 1 de  $\bar{Q}$ , logo  $Q$  permanece em 0

Após  $t_1$  o nível em reset volta para alto

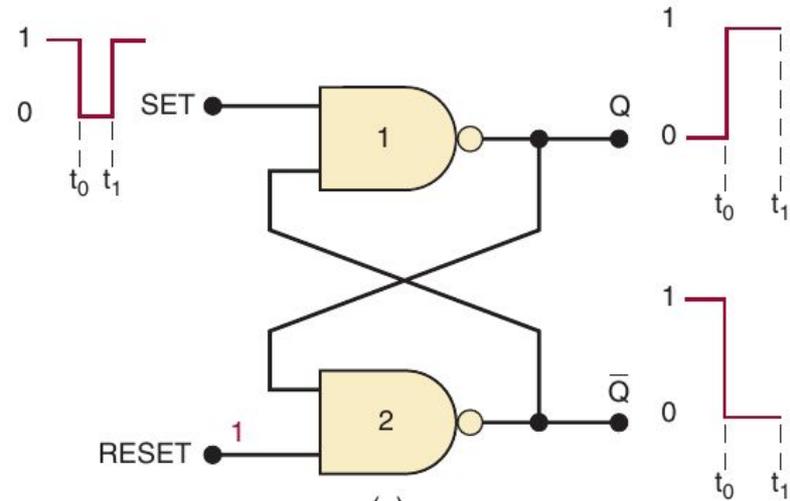
O NAND-2 recebe 1 do RESET e 0 de  $Q$ , logo  $\bar{Q}$  permanece em 1

O NAND-1 recebe 1 do set, e 1 de  $\bar{Q}$ , logo  $Q$  permanece em 0



# Faça você mesmo

Faça a mesma análise anterior, mas agora considere que o estado do latch era  $Q = 1$  antes do pulso



# Caso $Q$ fosse 1 antes do pulso

Entre  $t_0$  e  $t_1$  enviamos um pulso de nível baixo em **Reset**

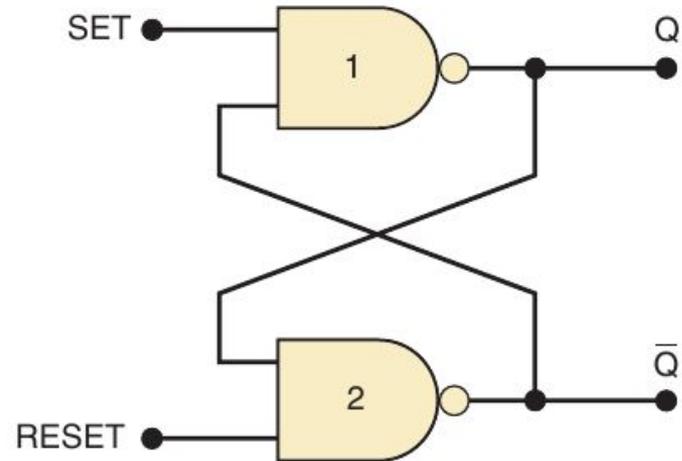
O NAND-2 recebe 0 do RESET e 1 de  $Q$ , logo  $\bar{Q} = 1$

O NAND-1 recebe 1 do set, e 1 de  $\bar{Q}$ , logo  $Q = 0$

Após  $t_1$  o nível em reset volta para alto

O NAND-2 recebe 1 do RESET e 0 de  $Q$ , logo  $\bar{Q}$  permanece em 1

O NAND-1 recebe 1 do set, e 1 de  $\bar{Q}$ , logo  $Q$  permanece em 0



# Set e Reset simultâneos

Durante o pulso,  $Q = \bar{Q} = 1$

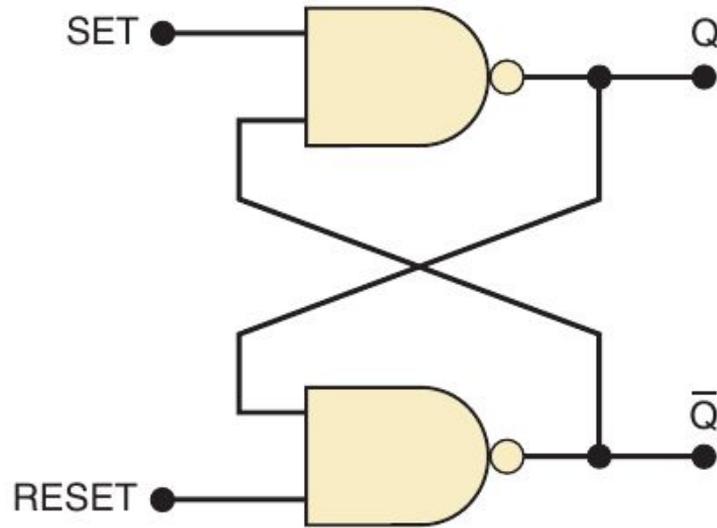
**Uma situação absurda!**

Após o fim do pulso, o resultado vai depender de quem retorna para alto antes (set ou reset)

Se ambos voltam para alto “ao mesmo tempo”, não podemos prever o resultado

De forma geral, enviar um nível lógico baixo para set e reset ao mesmo tempo nos levará a **comportamentos indesejados e/ou imprevisíveis!**

# Latch com NAND - Resumo

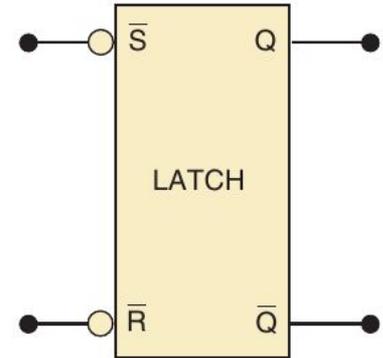
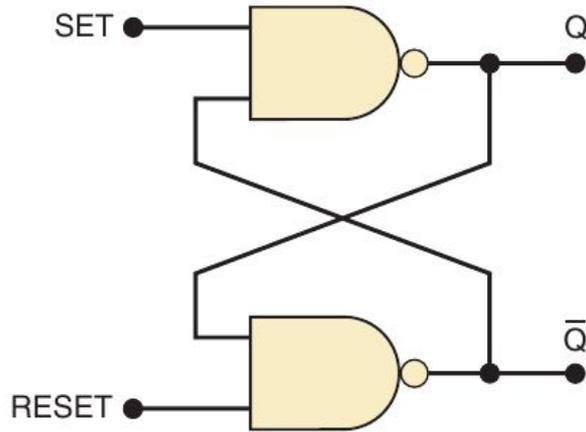


| Set | Reset | Output    |
|-----|-------|-----------|
| 1   | 1     | No change |
| 0   | 1     | $Q = 1$   |
| 1   | 0     | $Q = 0$   |
| 0   | 0     | Invalid*  |

\*Produces  $Q = \bar{Q} = 1$ .

# Latch com NAND

Como **S**et e **R**eset devem estar em 1 para manter o latch estável, o latch NAND geralmente é representado com S e R negados



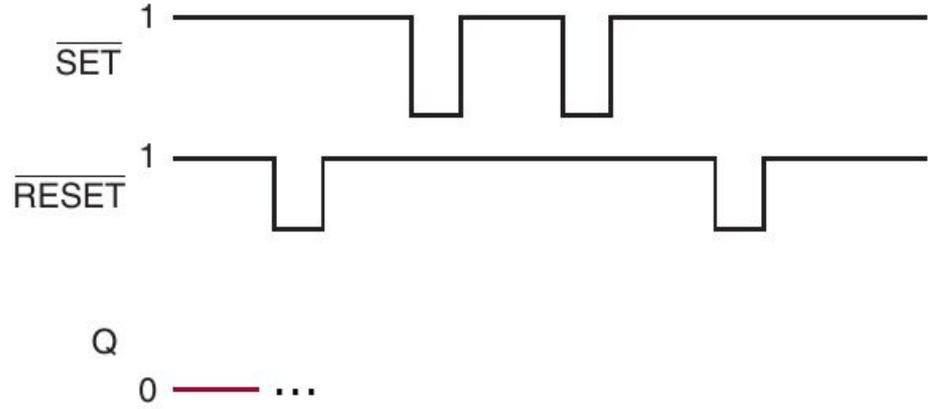
# Faça você mesmo

Faça o diagrama de temporização para o Latch

Considere que  $Q$  está inicialmente em 0

Como será o sinal em  $Q$ ?

Como será o sinal em  $\overline{Q}$ ?



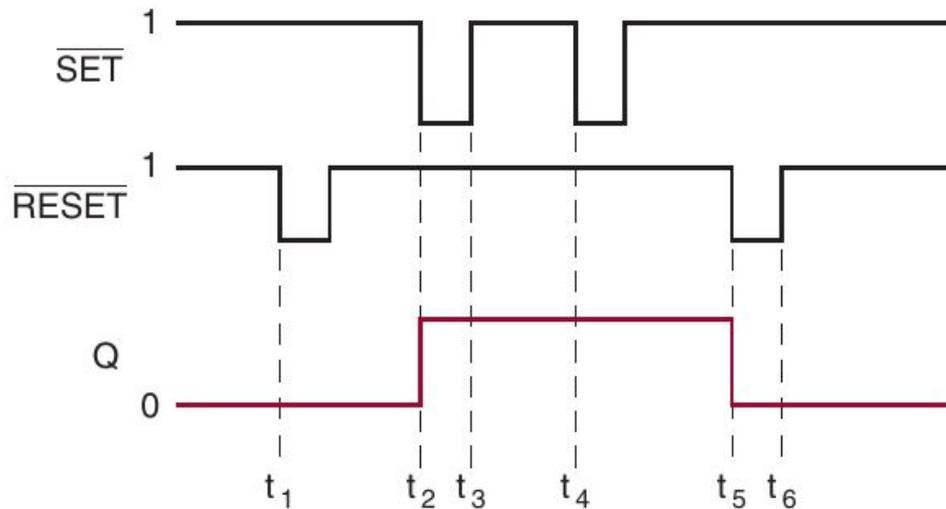
# Faça você mesmo

Faça o diagrama de temporização para o Latch

Considere que  $Q$  está inicialmente em 0

Como será o sinal em  $Q$ ?

Como será o sinal em  $\overline{Q}$ ?



# Latch com NOR

Podemos seguir um raciocínio similar ao utilizado com NAND's para construir um latch com NOR's

Uma das diferenças principais é que set e reset ficam em nível lógico baixo

Um nível lógico alto é enviado somente quando desejamos enviar um **Set/Reset**

# Energizando

Ao energizar um latch, não podemos afirmar se  $Q=0$  ou  $Q=1$

Depende de alguns fatores.

Exemplos:

- Atrasos de propagação do circuito

- Capacitâncias parasitas

Se o latch precisa ser iniciado em determinado estado, como podemos proceder?

# Energizando

Ao energizar um latch, não podemos afirmar se  $Q=0$  ou  $Q=1$

Depende de alguns fatores.

Exemplos:

Atrasos de propagação do circuito

Capacitâncias parasitas

Se o latch precisa ser iniciado em determinado estado, como podemos proceder?

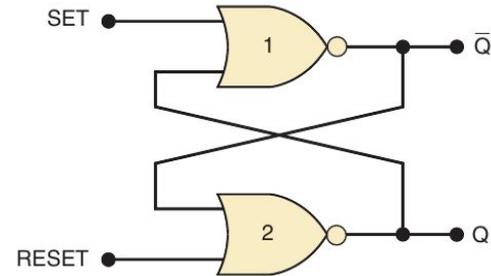
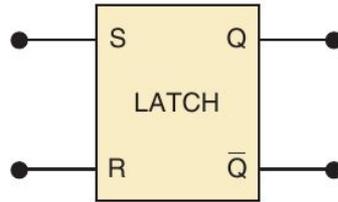
Aciona-se o seu **S**et ou **R**eset ao energizar o latch

Dependendo se desejamos que o estado inicial seja 0 ou 1

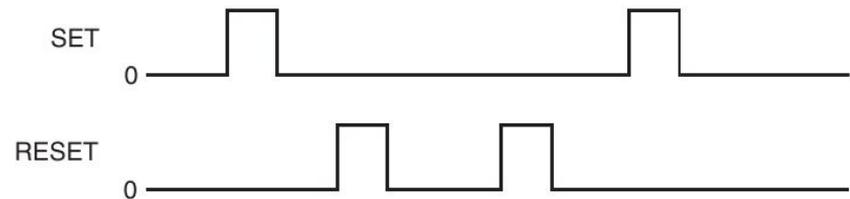
# Exercícios

1. Faça a mesma análise realizada para o latch NAND no latch de NOR's  
Preencha a tabela a seguir com a saída esperada em  $Q$  em caso de **Set/Reset**  
Compare seus resultados com Tocci et al (2017).

| Set | Reset | Output   |
|-----|-------|----------|
| 0   | 0     |          |
| 1   | 0     |          |
| 0   | 1     |          |
| 1   | 1     | Invalid* |

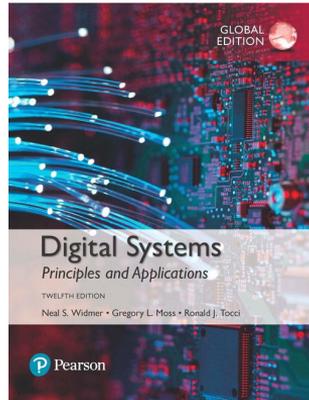


2. Considere o diagrama de temporização a seguir para as entradas **Set** e **Reset** de um Latch NOR. Considerando que  $Q$  está inicialmente em 0, como será o sinal em  $Q$  e  $\bar{Q}$ ?

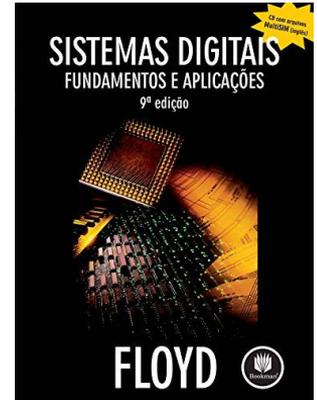


# Referências

Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Thomas Floyd. Sistemas Digitais: Fundamentos e Aplicações. 2009.



# Licença

Este obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

