

# CIRCUITOS LÓGICOS LATCHES

Marco A. Zanata Alves

# CIRCUITOS DIGITAIS COMBINACIONAIS

Até agora, todos os circuitos digitais que estudamos possuem uma propriedade em comum:

O estado das saídas depende única e exclusivamente do estado atual das entradas.

Tais circuitos são classificados como circuitos digitais combinacionais.

... **circuitos combinacionais** não guardam nenhuma informação sobre estados anteriores (ausência de memória)

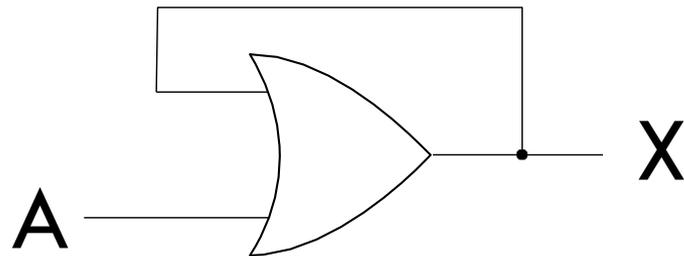
# CIRCUITOS DIGITAIS SEQUENCIAIS

Alguns circuitos digitais, ao contrário, podem guardar informação sobre estados anteriores. Tais circuitos são chamados de circuitos digitais sequenciais.

... **em circuitos sequenciais**, o estado das saídas depende não apenas do estado atual das entradas, mas também de estados anteriores das entradas e/ou saídas (presença de memória).

# CIRCUITOS DIGITAIS SEQUENCIAIS

Exemplo de um circuito digital sequencial:

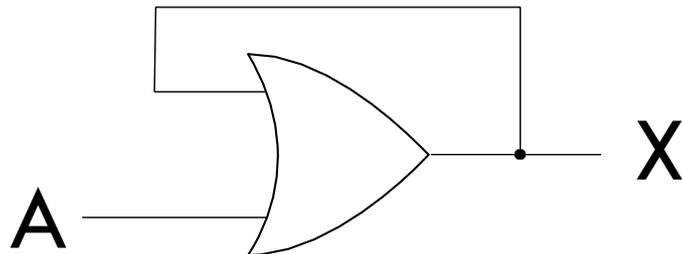


# CIRCUITOS DIGITAIS SEQUENCIAIS

Exemplo de um circuito digital sequencial:

A tabela verdade de um circuito digital sequencial depende de estados anteriores.

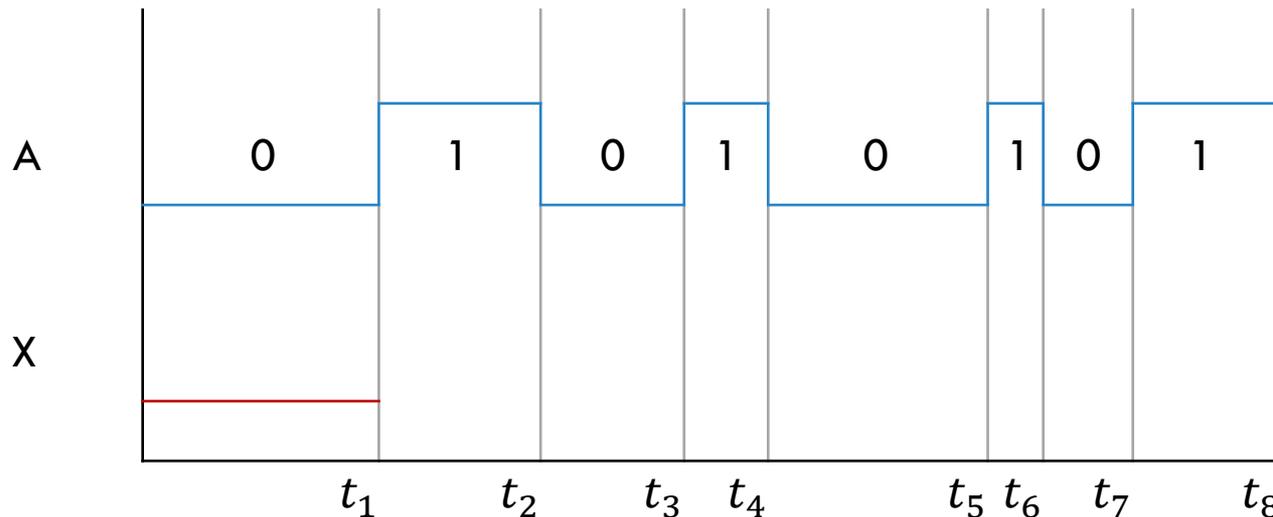
Usaremos  $X_i$  para denotar o estado atual da saída e  $X_{i-1}$  para denotar o estado anterior.



$A$	$X_{i-1}$	$X_i$
0	0	0
1	0	1
0	1	1
1	1	1

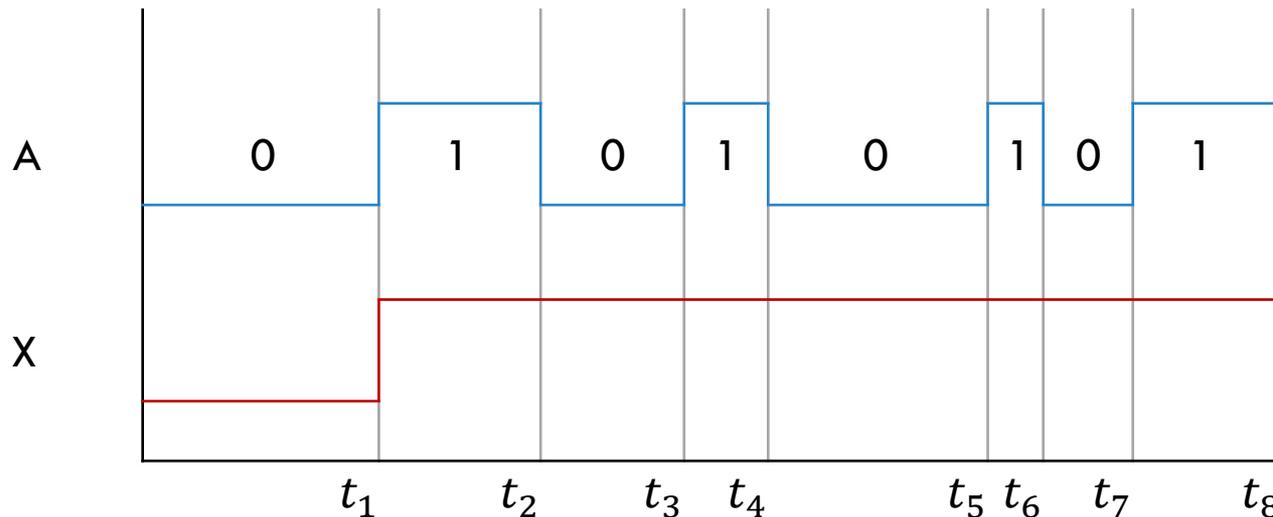
# CIRCUITOS DIGITAIS SEQUENCIAIS

Esboce o diagrama de forma de onda para a saída  $X$ , considerando o diagrama de forma de onda para a entrada  $A$ , e que até o instante  $t_1$  o estado de  $X$  é 0.



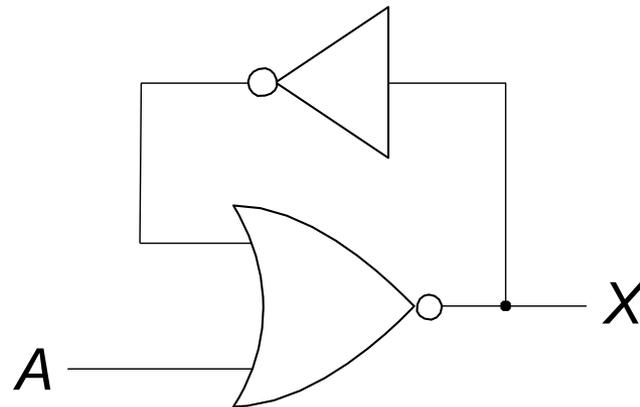
# CIRCUITOS DIGITAIS SEQUENCIAIS

Esboce o diagrama de forma de onda para a saída  $X$ , considerando o diagrama de forma de onda para a entrada  $A$ , e que até o instante  $t_1$  o estado de  $X$  é 0.



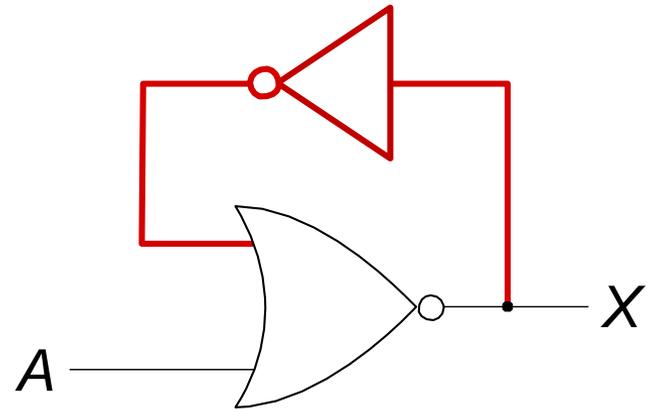
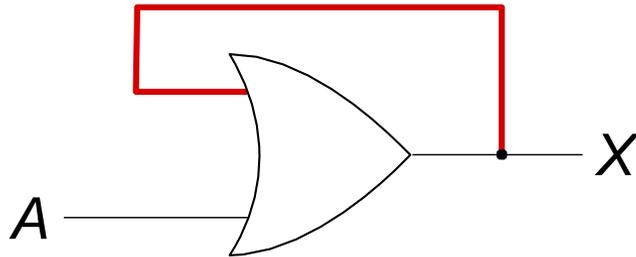
# CIRCUITOS DIGITAIS SEQUENCIAIS

Outro exemplo: construa a tabela verdade e esboce o diagrama de forma de onda para a saída  $X$  abaixo, considerando que o estado inicial de  $A$  é 0 e de  $X$  é 1.



# FEEDBACK

Característica comum aos circuitos digitais sequenciais: presença de feedback (realimentação)



# LATCH DO TIPO R-S (RESET-SET)

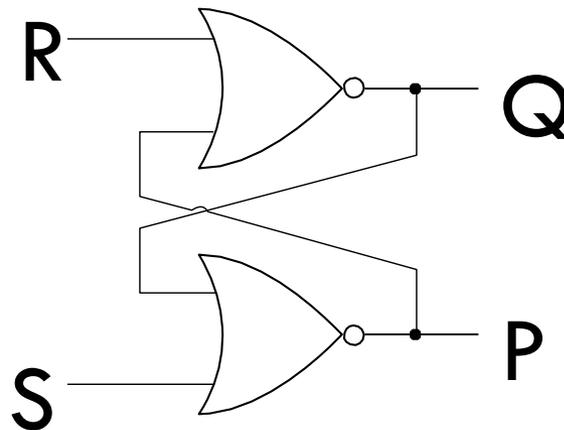
Faça a tabela verdade do circuito abaixo e esboce o diagrama de forma de onda para as saídas Q e P, considerando a seguinte sequência de estados para as entradas:

$R = 1, S = 0;$

$R = 0, S = 0;$

$R = 0, S = 1;$

$R = 0, S = 0.$



Estamos sempre interessados no sinal final (estável)

Desconsidere, por enquanto, o estado  $R = 1, S = 1.$

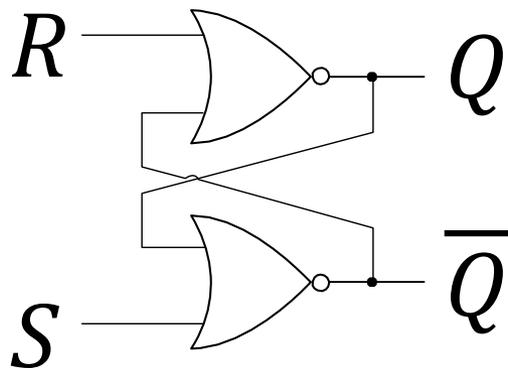
# LATCH DO TIPO R-S (RESET-SET)

A saída  $P$  é sempre o inverso de  $Q$ . Passaremos a chamar a saída  $P$  de  $\overline{Q}$ .

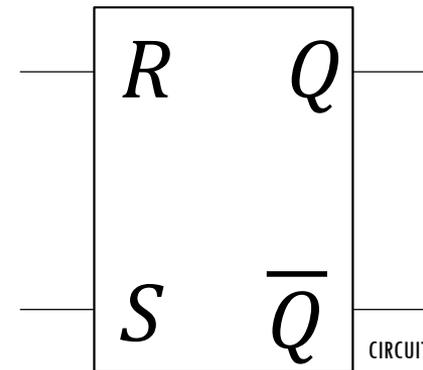
A tabela verdade desse circuito pode ser escrita como:

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)

Este circuito é chamado **latch R-S**



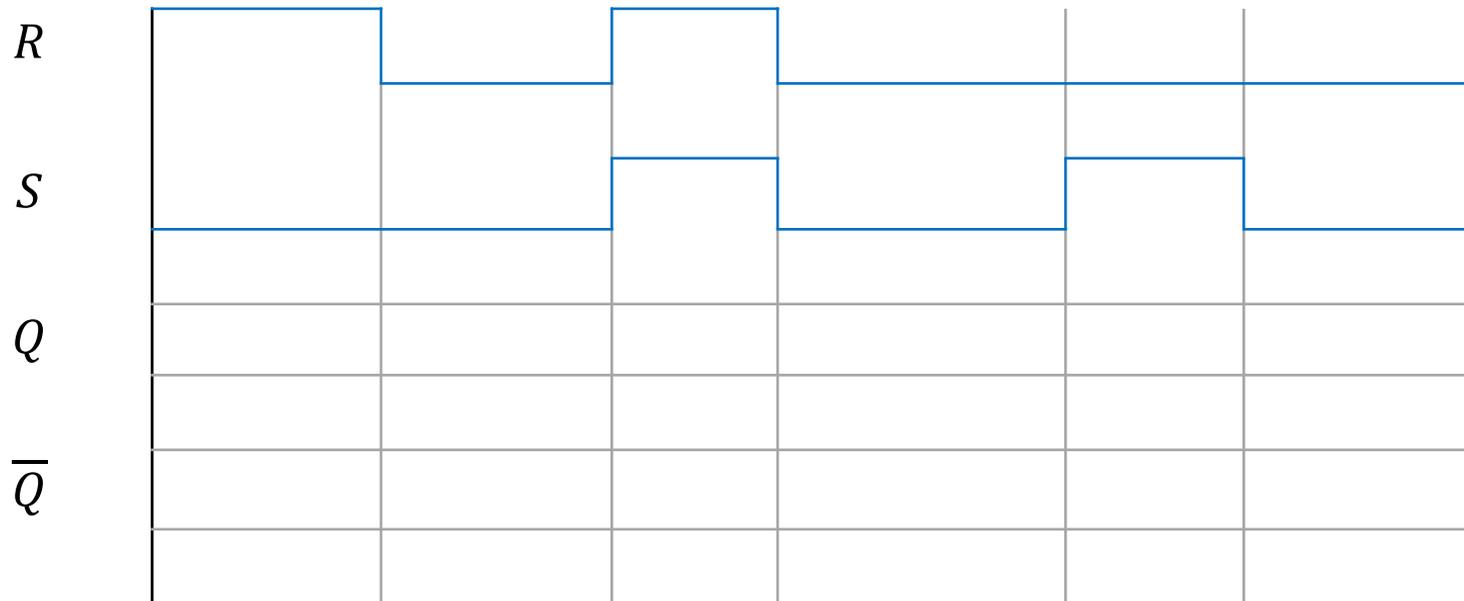
=



# LATCH DO TIPO R-S (RESET-SET)

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

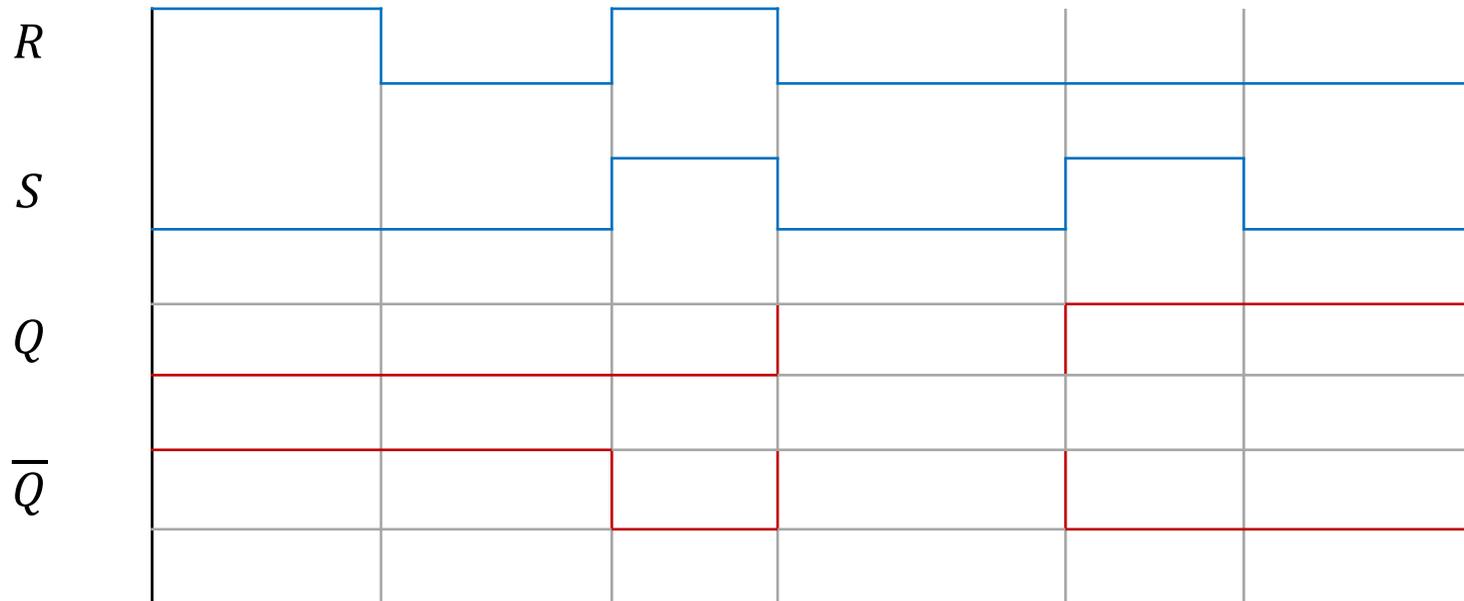
Esboce os diagramas de forma de onda para  $Q$  e  $\overline{Q}$ , considerando  $R$  e  $S$  conforme o diagrama abaixo.



# LATCH DO TIPO R-S (RESET-SET)

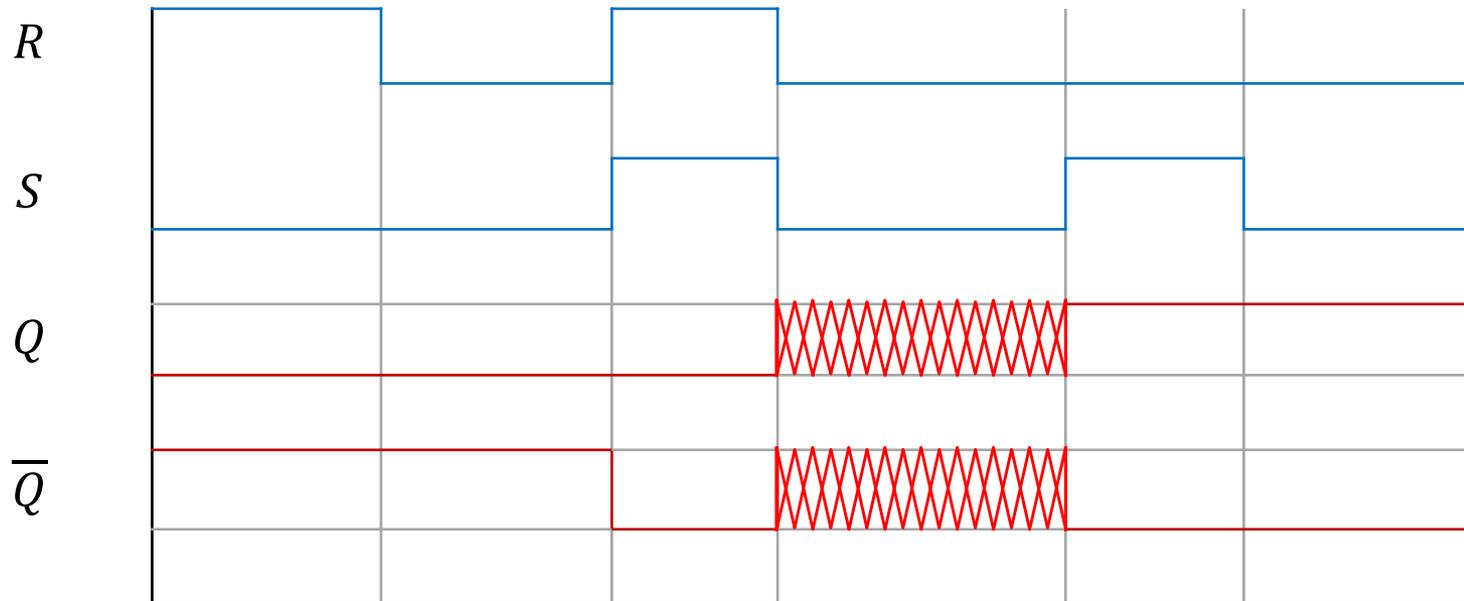
E o estado  $R = 1, S = 1$ ?

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

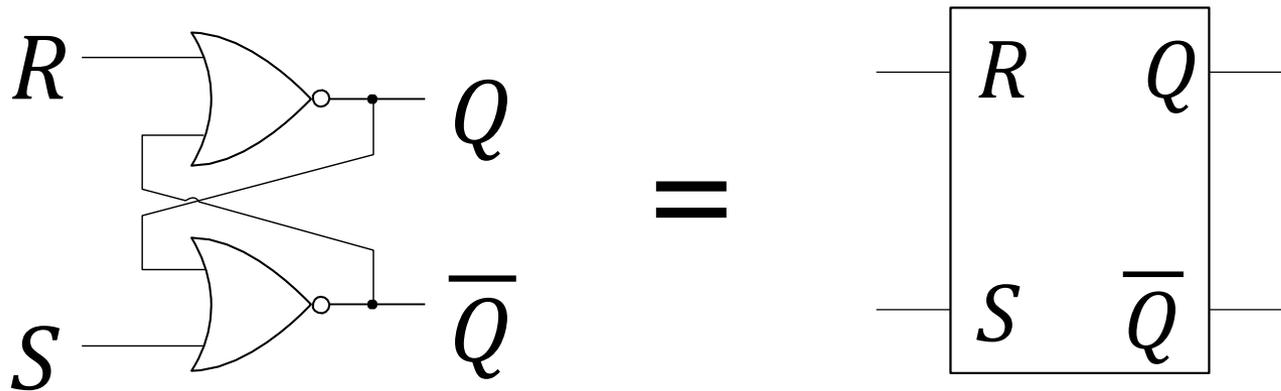


# LATCH DO TIPO R-S (RESET-SET)

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.



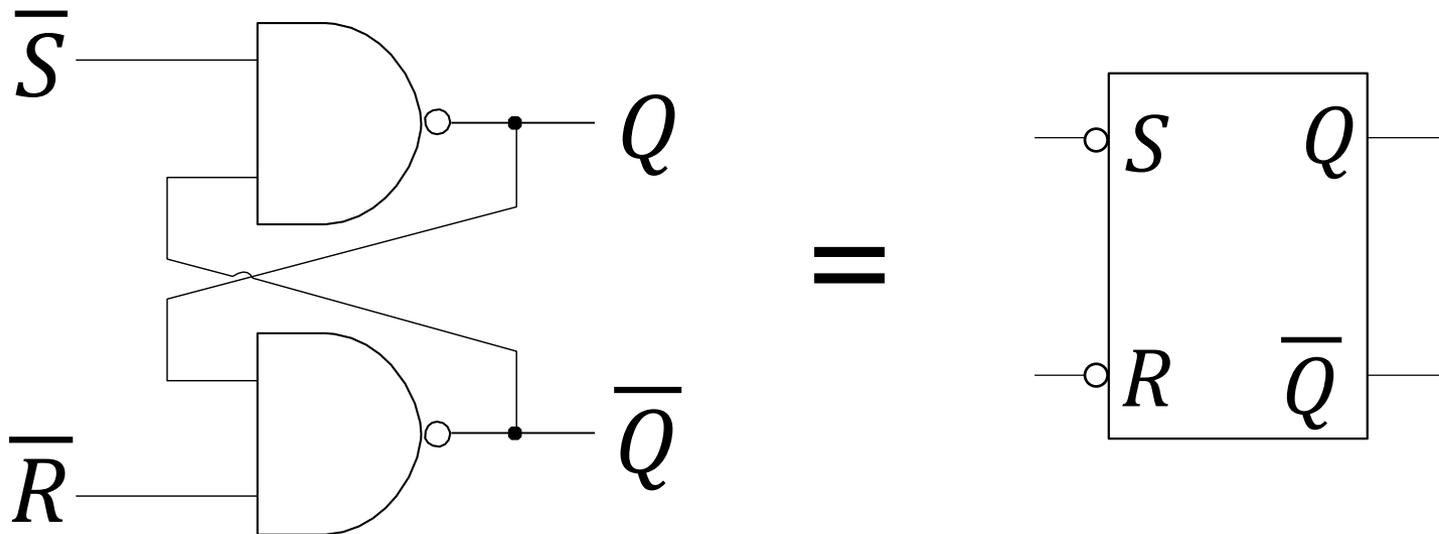
# LATCH DO TIPO R-S (RESET-SET)



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

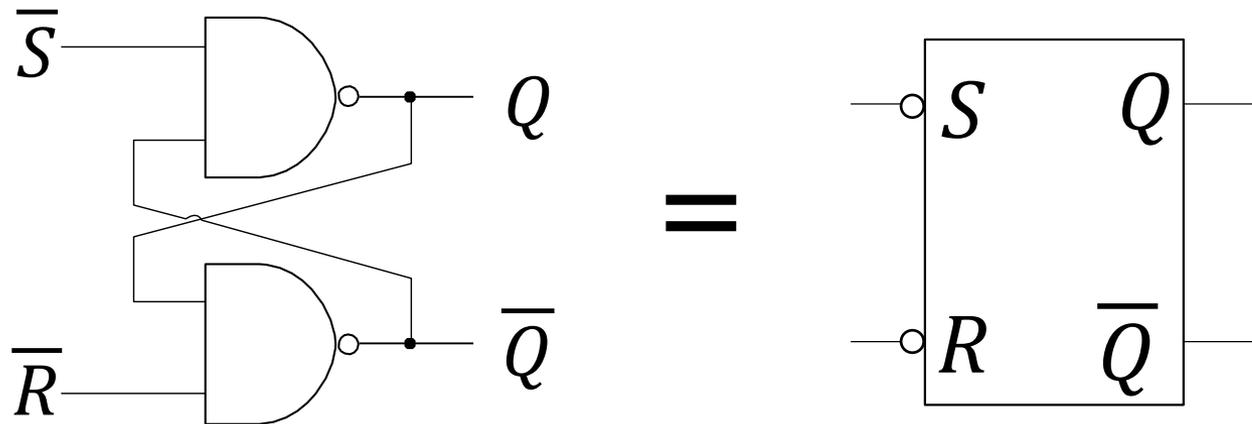
# LATCH DO TIPO $\overline{S} - \overline{R}$ COM PORTAS NAND

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



# LATCH DO TIPO $\bar{S} - \bar{R}$ COM PORTAS NAND

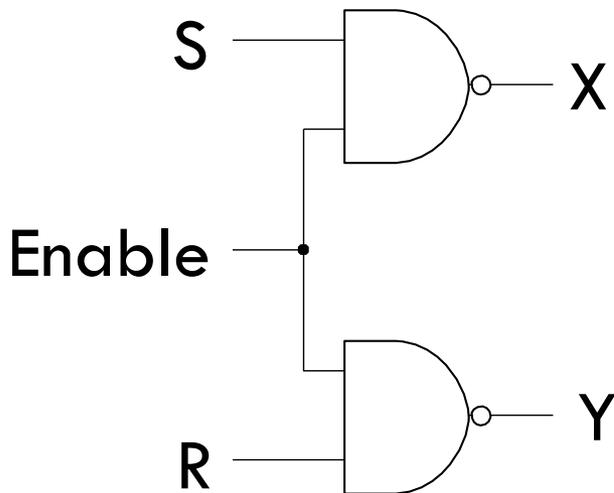
É possível construir um latch similar com portas NAND, mas as entradas se tornam ativas em nível baixo.



$\bar{S}$	$\bar{R}$	$Q_i$	$\bar{Q}_i$	
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

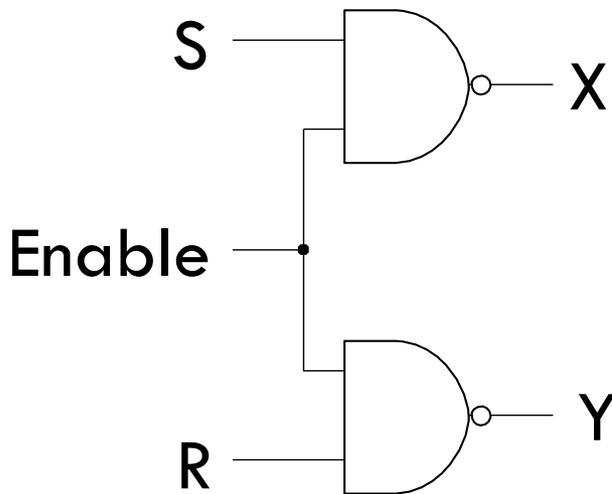
# CIRCUITO DE HABILITAÇÃO (ENABLE)

Problema 1: considere o circuito abaixo. Qual é o estado de cada saída X e Y quando  $En = 0$  e quando  $En = 1$ ?



# CIRCUITO DE HABILITAÇÃO (ENABLE)

Problema 1: considere o circuito abaixo. Qual é o estado de cada saída X e Y quando  $En = 0$  e quando  $En = 1$ ?



Enable	X	Y
0	1	1
1	$\bar{S}$	$\bar{R}$

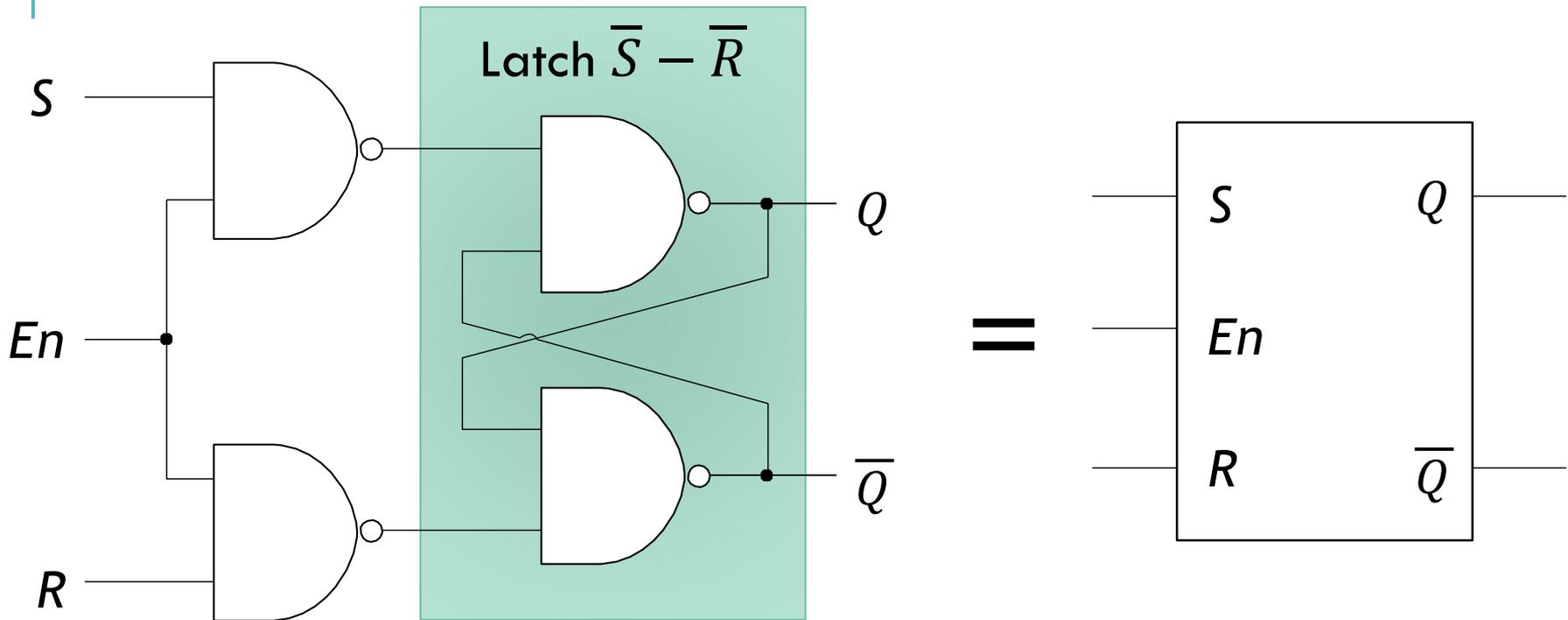
Circuito de habilitação com portas NAND torna as entradas S e R:

Se  $En = 1$ : ativas em nível baixo, ( $\bar{S}$  e  $\bar{R}$ )

Se  $En = 0$ , desabilitadas

A entrada **Enable** é chamada entrada de habilitação (enable input)

# LATCH DO TIPO S-R COM ENABLE



En	S	R	$Q_i$	
1	0	1	0	(reseta Q)
1	1	0	1	(seta Q)
1	0	0	$Q_{i-1}$	(mantém Q)
0	?	?	$Q_{i-1}$	(mantém Q, não importa R nem S)

# LATCH DO TIPO D (DATA)

A inclusão da entrada  $En$  é uma conveniência a mais no latch S-R, e permite uma outra forma de manter o estado do latch.

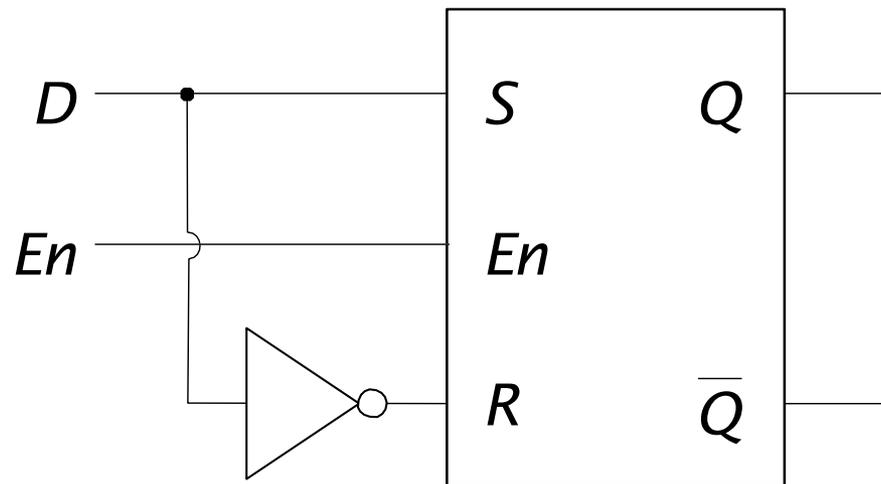
Porém, ainda temos que nos preocupar em nunca fazer  $S = 1$  e  $R = 1$  enquanto o latch estiver habilitado ( $En = 1$ ).

# LATCH DO TIPO D (DATA)

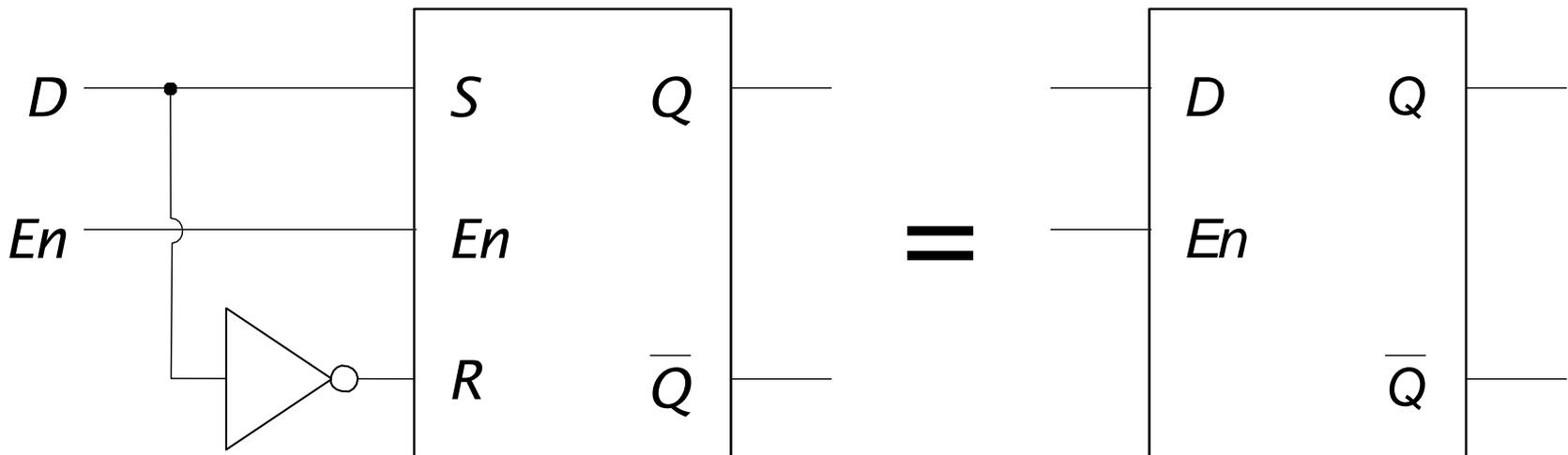
A inclusão da entrada  $En$  é uma conveniência a mais no latch S-R, e permite uma outra forma de manter o estado do latch.

Porém, ainda temos que nos preocupar em nunca fazer  $S = 1$  e  $R = 1$  enquanto o latch estiver habilitado ( $En = 1$ ).

O uso de uma única entrada para set/reset evita esse problema:



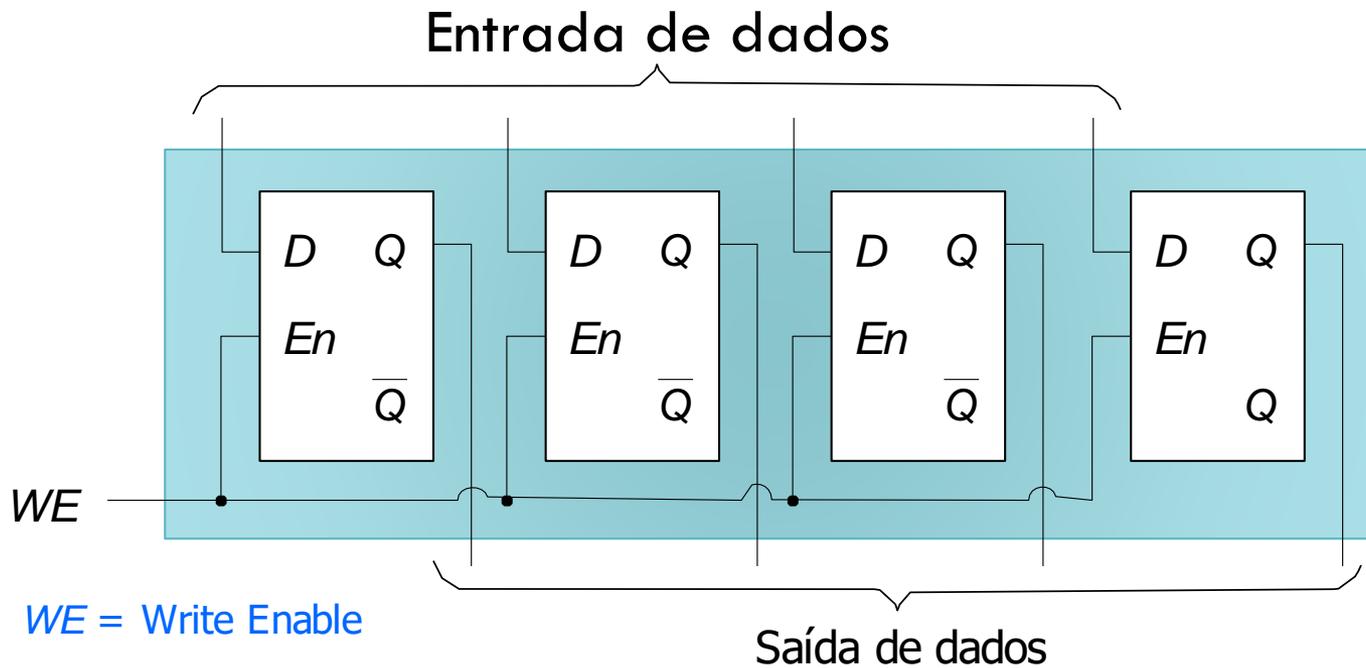
# LATCH DO TIPO D (DATA)



$D$	$En$	$Q_i$	
0	1	0	(reset)
1	1	1	(set)
?	0	$Q_{i-1}$	(mantém, sem importar com D)

# LATCH D: APLICAÇÃO

Registrador de armazenamento: armazena uma palavra de dado

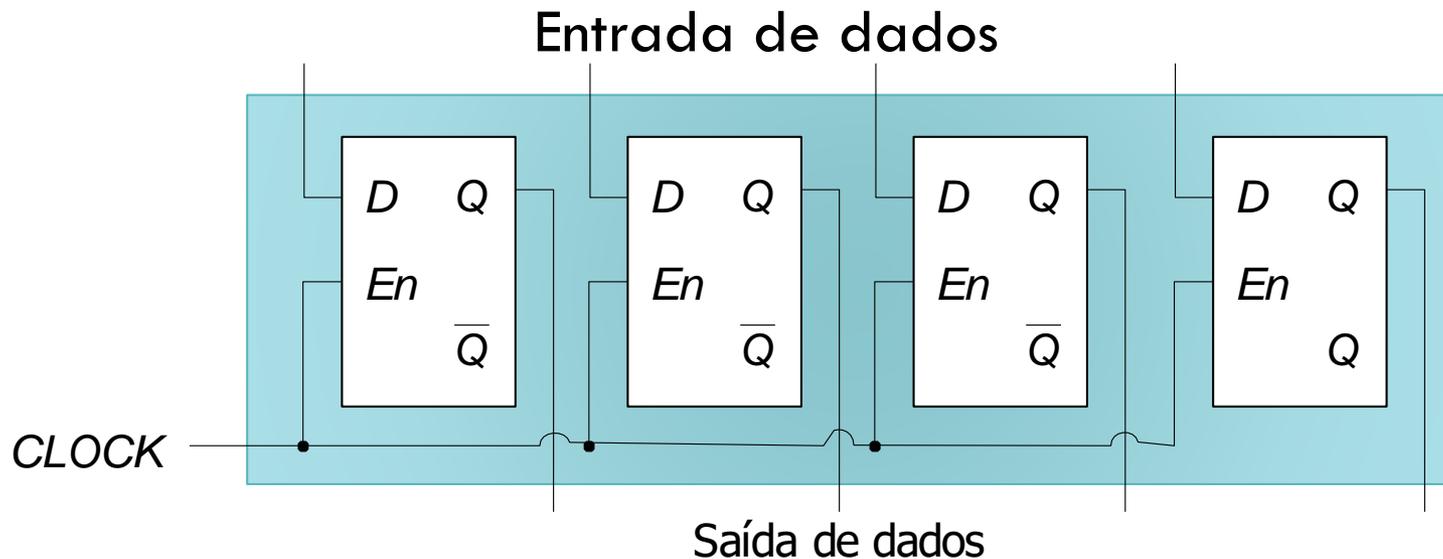


# LATCH D: APLICAÇÃO

Os latches são muito utilizados com Write-Enable ligado ao Clock.

Dessa maneira, os latches ficam **transparentes** (aceitando escritas) apenas durante a “parte alta” do clock (“1” lógico).

Durante a “parte baixa” (“0” lógico) do clock, o latch fica **opaco**.



# LATCH D: APLICAÇÃO

Registrador de armazenamento: armazena uma palavra de dado

