

*UNIOESTE*  
*Ciência da Computação*

*Sistemas Digitais*  
*Circuitos Sequenciais*

*Prof. Jorge Habib El Khouiri*  
*Prof. Antonio Marcos Hachisuca*

2020/2021

## Referências Bibliográficas

1. *Digital Fundamentals*, Thomas L. Floyd; Editora: Pearson; Edição: 11; Ano: 2015;
2. *Sistemas Digitais Princípios e Aplicações*, Ronald J. Tocci; Editora: Pearson; Edição: 11; Ano: 2011;
3. *Computer Organization and Design*, David A. Patterson; Editora: Elsevier; Edição: 1; Ano: 2017
4. *Digital Design: Principles and Practices*, John F. Wakerly; Editora: Pearson; Edição: 5; Ano: 2018;
5. *Guide to Assembly Language Programming in Linux*, Sivarama P. Dandamudi; Editora: Springer; Edição: 1; Ano: 2005.
6. *Fundamentals of Logic Design*, Roth Jr, Charles H; Kinney, Larry L; Seventh Edition. Editora: Cengage Learning, Ano: 2013.

## Sumário

1. Revisão – Sistemas de Numeração
2. Revisão – Representação de Dados
3. Revisão – Operações com Binários
4. Álgebra Booleana
5. Simplificação de Expressões
6. Mapa de Karnaugh
7. Elementos Lógicos Universais
8. Circuitos Combinacionais
9. Circuitos Sequenciais

1. Latches
2. Flip-Flop
3. Registradores
4. Contadores
5. Máquina de Estados
6. Memória RAM

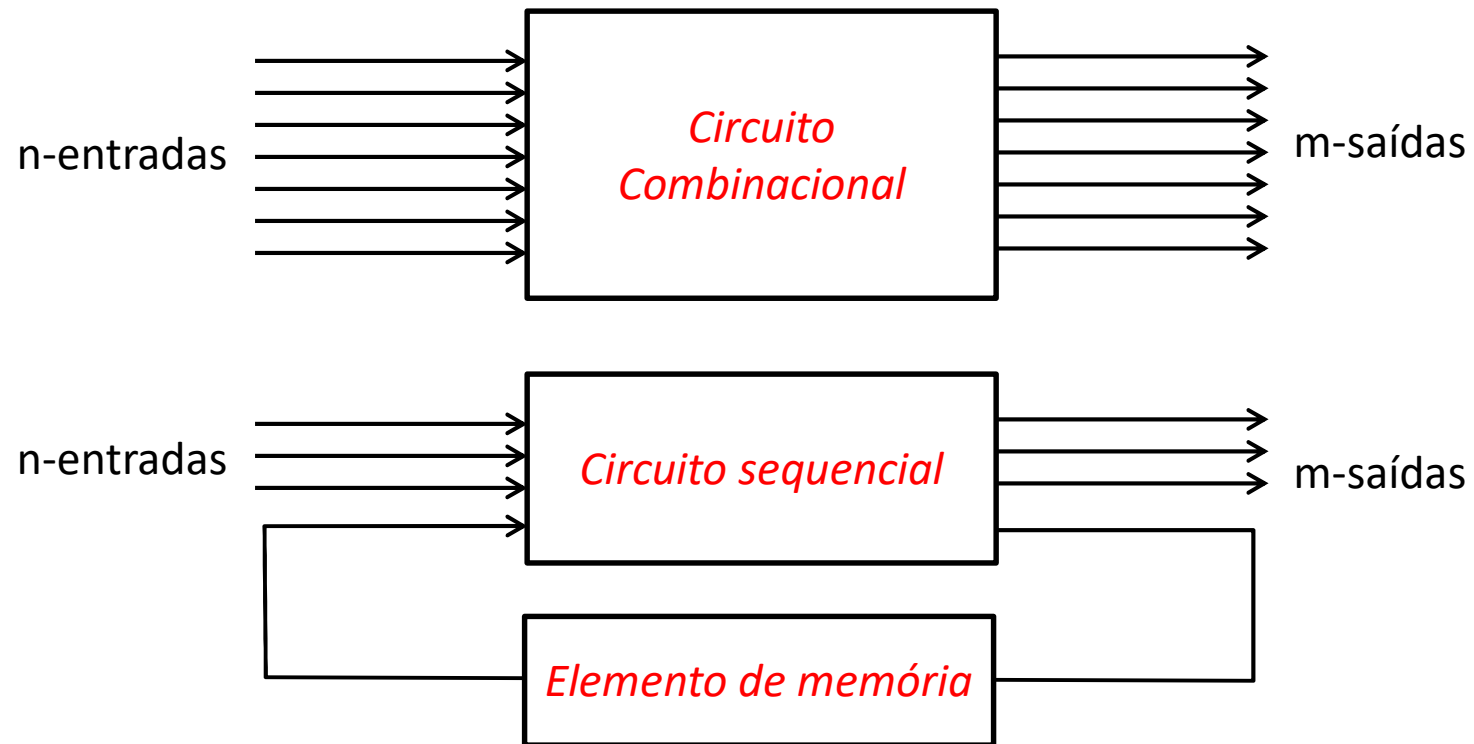
## Circuitos Combinacionais

- Um circuito combinacional consiste em portas lógicas cujas saídas, em qualquer momento, são determinadas pela combinação dos valores das entradas
- Para  $n$  variáveis de entrada, existem  $2^n$  combinações de entrada binária possíveis
- Para cada combinação binária das variáveis de entrada, existe uma saída possível

## Circuitos Combinacionais vs circuitos sequenciais

- Os circuitos combinacionais não possuem memória interna
  - O valor de saída depende apenas dos valores atuais de entrada
- Os circuitos sequenciais contêm lógica combinacional, e elementos de memória (usados para armazenar estados de circuito)
  - As saídas dependem dos valores de entrada atuais e dos valores de entrada anteriores (mantidos nos elementos de memória)

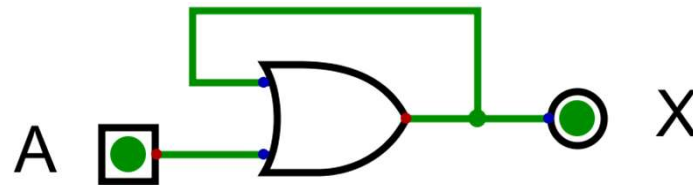
## Circuitos Combinacionais vs circuitos sequenciais



## Circuitos Sequenciais

### Latch

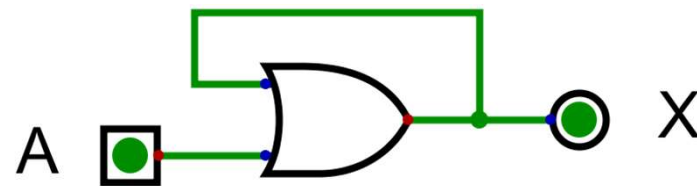
- Exemplo de um circuito sequencial:



- Observe que há um *feedback* (retro alimentação ou realimentação) da saída  $X$  para uma das entradas da porta  $OR$ ;
- Neste caso faremos a distinção de  $X$  em dois momentos, sendo  $X_i$  o estado atual e  $X_{i-1}$  o estado anterior;

## Circuitos Sequenciais

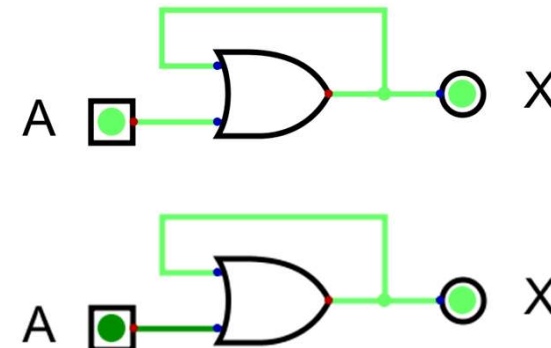
### Latch



- Inicialmente a entrada  $A$  e a saída  $X_i$  estão com o valor 0;
- Logo, podemos montar a equação e a tabela verdade para este circuito:

$$X_i = A + X_{i-1}$$

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

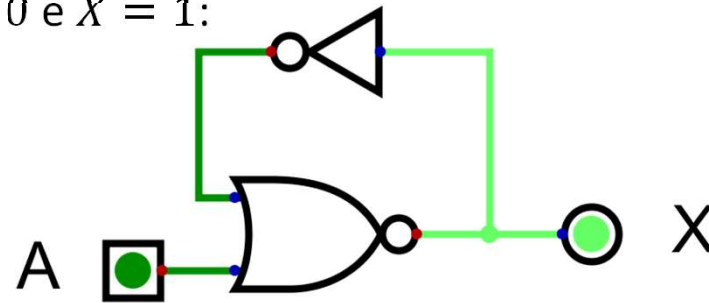




## Circuitos Sequenciais

### Latch

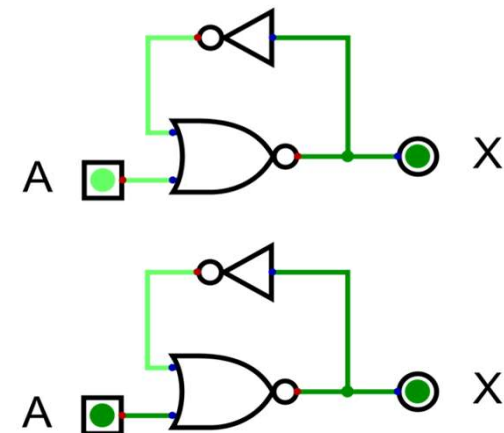
- Determine a equação e a tabela verdade para o seguinte circuito, considerando inicialmente  $A = 0$  e  $X = 1$ :



$$X_i = \overline{A + \bar{X}_{i-1}}$$

$$X_i = \bar{A} \cdot X_{i-1}$$

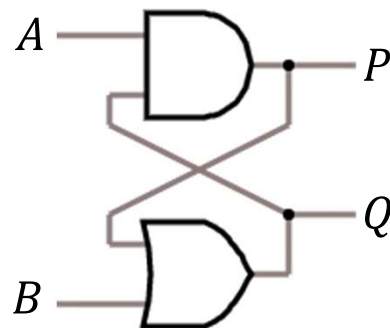
A	$\bar{X}_{i-1}$	$X_i$
0	0	1
1	0	0
0	1	0
1	1	0



## Circuitos Sequenciais

### Latch

- Determine a equação e a tabela verdade para o seguinte circuito, considerando inicialmente  $A = B = P = Q = 0$ :



$$P_i = A \cdot Q_{i-1}$$

$$Q_i = B + P_{i-1}$$

A	B	$P_i$	$Q_i$
0	0	0	$P_{i-1} = 0$
0	1	0	1
1	1	$Q_{i-1} = 1$	1
1	0	$Q_{i-1}$	$P_{i-1}$

Q segue B

P segue A

## Circuitos Sequenciais

### Latch

- Construa a tabela verdade do circuito abaixo para as saídas  $Q$  e  $P$ , considerando a seguinte sequência de estados para as entradas:

✓  $R = 1, S = 0;$

✓  $R = 0, S = 0;$

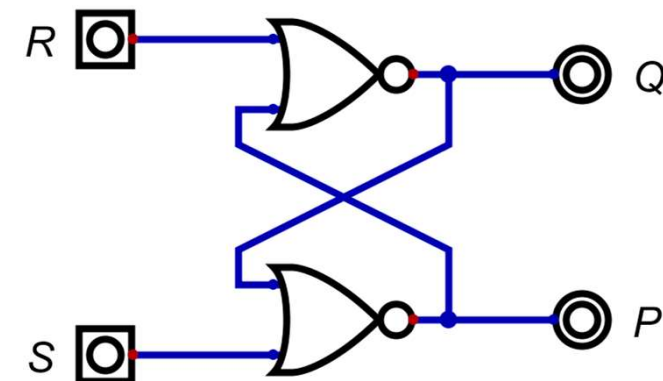
✓  $R = 0, S = 1;$

✓  $R = 0, S = 0$

- Desconsidere o estado  $R = 1, S = 1$ .

$$Q_i = \overline{R + P_{i-1}} = \bar{R} \cdot \bar{P}_{i-1}$$

$$P_i = \overline{S + Q_{i-1}} = \bar{S} \cdot \bar{Q}_{i-1}$$



## Circuitos Sequenciais

### Latch

- Construa a tabela verdade do circuito abaixo para as saídas  $Q$  e  $P$ , considerando a seguinte sequência de estados para as entradas:

✓  $R = 1, S = 0;$

✓  $R = 0, S = 0;$

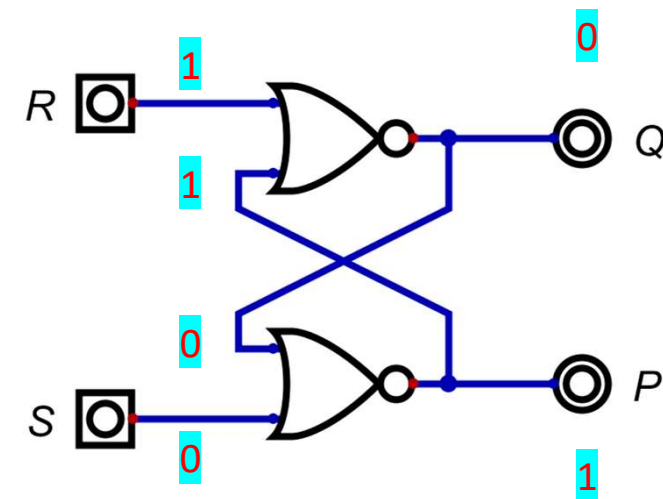
✓  $R = 0, S = 1;$

✓  $R = 0, S = 0$

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = 0$$

$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = \bar{Q}_{i-1} = 1$$

$R$	$S$	$Q_i$	$P_i$
1	0	0	1



## Circuitos Sequenciais

### Latch

- Construa a tabela verdade do circuito abaixo para as saídas  $Q$  e  $P$ , considerando a seguinte sequência de estados para as entradas:

✓  $R = 1, S = 0;$

✓  $R = 0, S = 0;$

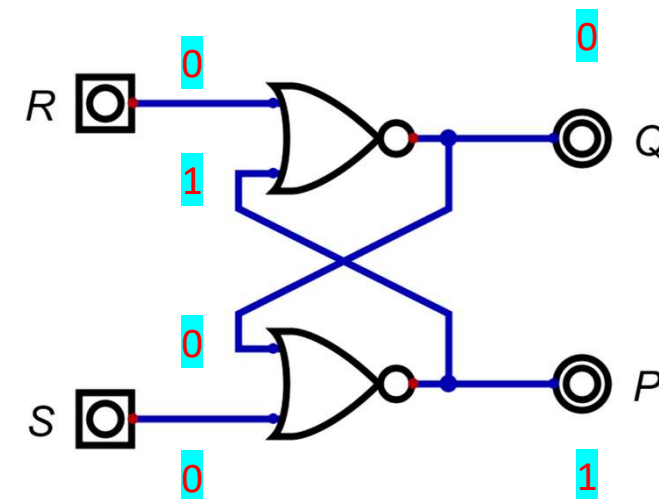
✓  $R = 0, S = 1;$

✓  $R = 0, S = 0$

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = \bar{P}_{i-1} = 0$$

$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = \bar{Q}_{i-1} = 1$$

$R$	$S$	$Q_i$	$P_i$
1	0	0	1
0	0	0	1



# Circuitos Sequenciais

## Latch

- Construa a tabela verdade do circuito abaixo para as saídas  $Q$  e  $P$ , considerando a seguinte sequência de estados para as entradas:

✓  $R = 1, S = 0;$

✓  $R = 0, S = 0;$

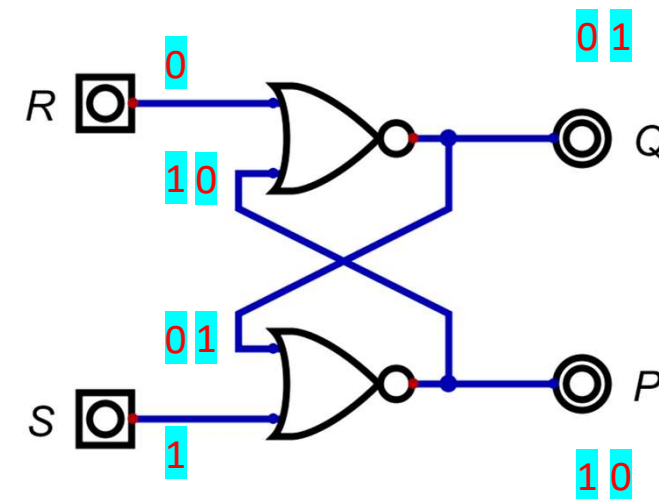
✓  $R = 0, S = 1;$

✓  $R = 0, S = 0$

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = \bar{P}_{i-1} = 1$$

$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = 0$$

$R$	$S$	$Q_i$	$P_i$
1	0	0	1
0	0	0	1
0	1	1	0



## Circuitos Sequenciais

### Latch

- Construa a tabela verdade do circuito abaixo para as saídas  $Q$  e  $P$ , considerando a seguinte sequência de estados para as entradas:

✓  $R = 1, S = 0;$

✓  $R = 0, S = 0;$

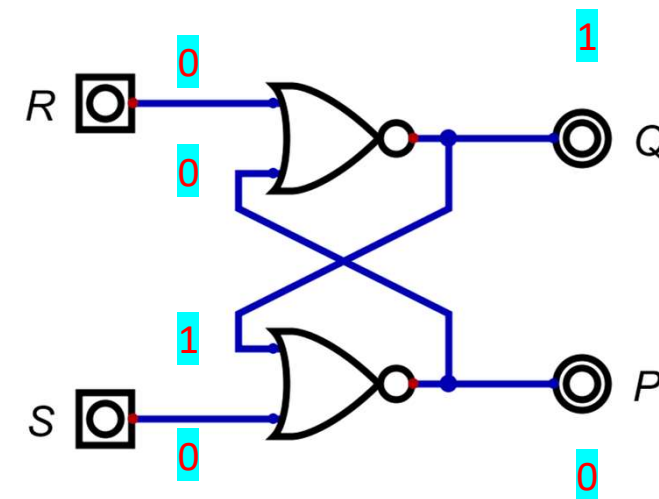
✓  $R = 0, S = 1;$

✓  $R = 0, S = 0$

$R$	$S$	$Q_i$	$P_i$
1	0	0	1
0	0	0	1
0	1	1	0
0	0	1	0

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = \bar{P}_{i-1} = 1$$

$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = \bar{Q}_{i-1} = 0$$



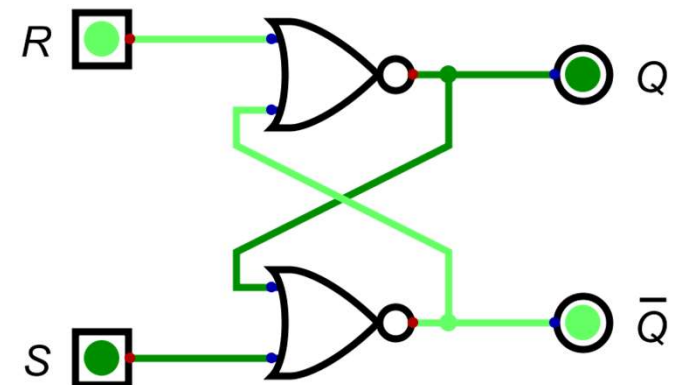
## Circuitos Sequenciais

### Latch

- A saída  $P$  é sempre a negação da saída  $Q$  ( $P = \bar{Q}$ );
- De forma geral a tabela verdade ficará:

$R$	$S$	$Q_i$	$\bar{Q}_i$
1	0	0	1
0	1	1	0
0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$

- Este circuito é denominado *latch R-S*;





## Circuitos Sequenciais

### Latch

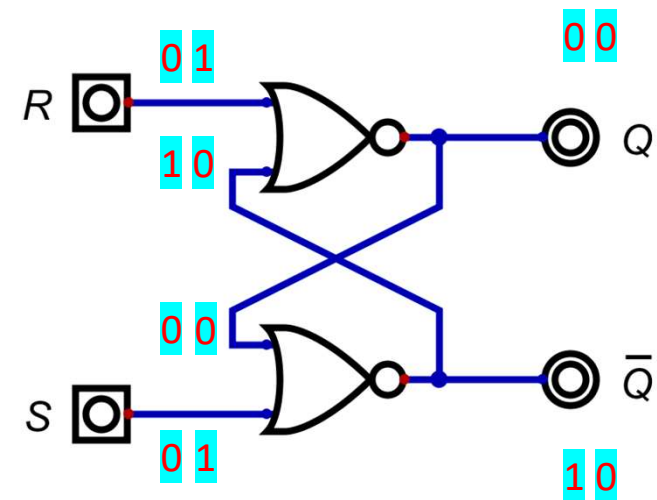
- Qual o comportamento para a seguinte entrada?

✓  $R = 1, S = 1$

$R$	$S$	$Q_i$	$\overline{Q}_i$
1	0	0	1
0	0	0	1
1	1	0	0

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = 0$$

$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = 0$$



## Circuitos Sequenciais

### Latch

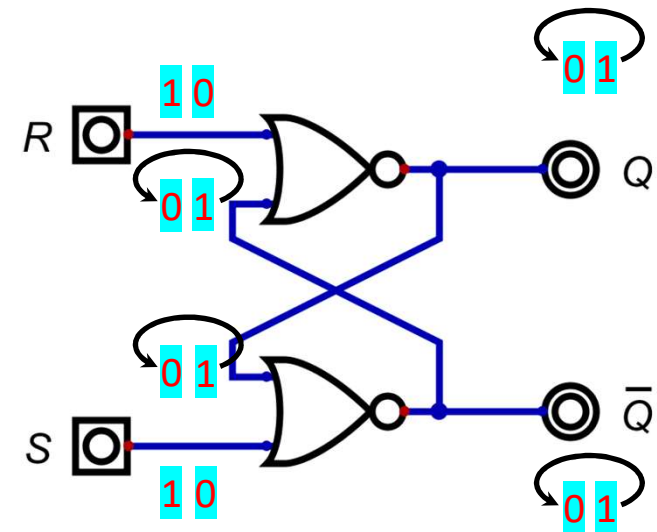
- Qual o comportamento para a seguinte entrada?

✓  $R = 0, S = 0$

$R$	$S$	$Q_i$	$\bar{Q}_i$
1	0	0	1
0	0	0	1
1	1	0	0
0	0	?	?

$$Q_i = \bar{R} \cdot \bar{P}_{i-1} = \bar{P}_{i-1} = 1 = 0 = 1$$

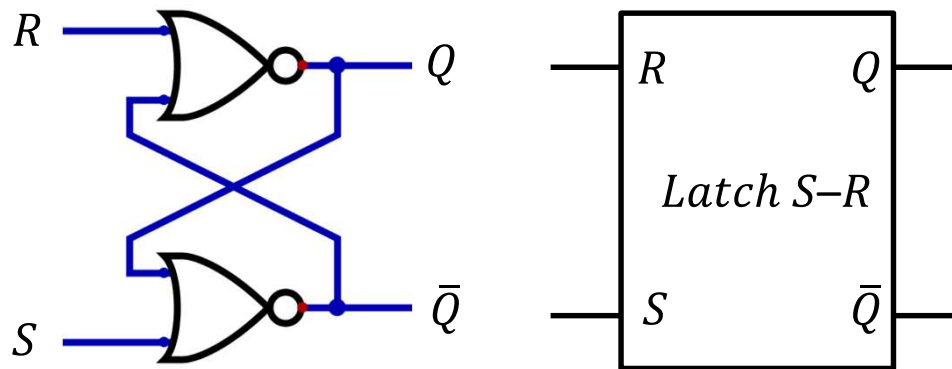
$$P_i = \bar{S} \cdot \bar{Q}_{i-1} = \bar{Q}_{i-1} = 1 = 0 = 1$$



## Circuitos Sequenciais

### Latch

- Este circuito é denominado de *Latch*;
- *Latch* é um tipo de dispositivo de armazenamento temporário que possui dois estados estáveis (biestável);

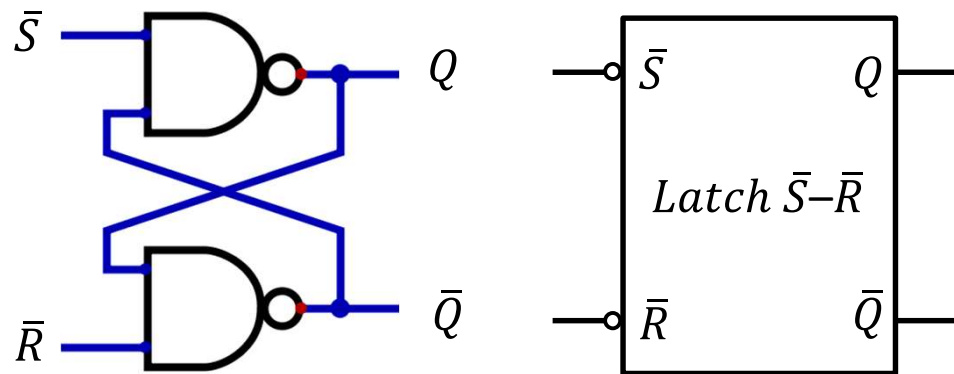


$R$	$S$	$Q_i$	$\bar{Q}_i$	Função
1	0	0	1	Reset $Q$
0	1	1	0	Set $Q$
0	0	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém $Q$
1	1	?	?	Estado proibido

## Circuitos Sequenciais

### Latch

- É possível construir um circuito *Latch* utilizando portas *NAND*;
- Neste caso as entradas serão ativas em *LOW*;



$\bar{S}$	$\bar{R}$	$Q_i$	$\bar{Q}_i$	Função
1	0	0	1	Reset $Q$
0	1	1	0	Set $Q$
1	1	$Q_{i-1}$	$\bar{Q}_{i-1}$	Mantém $Q$
0	0	?	?	Estado proibido

## Circuitos Sequenciais

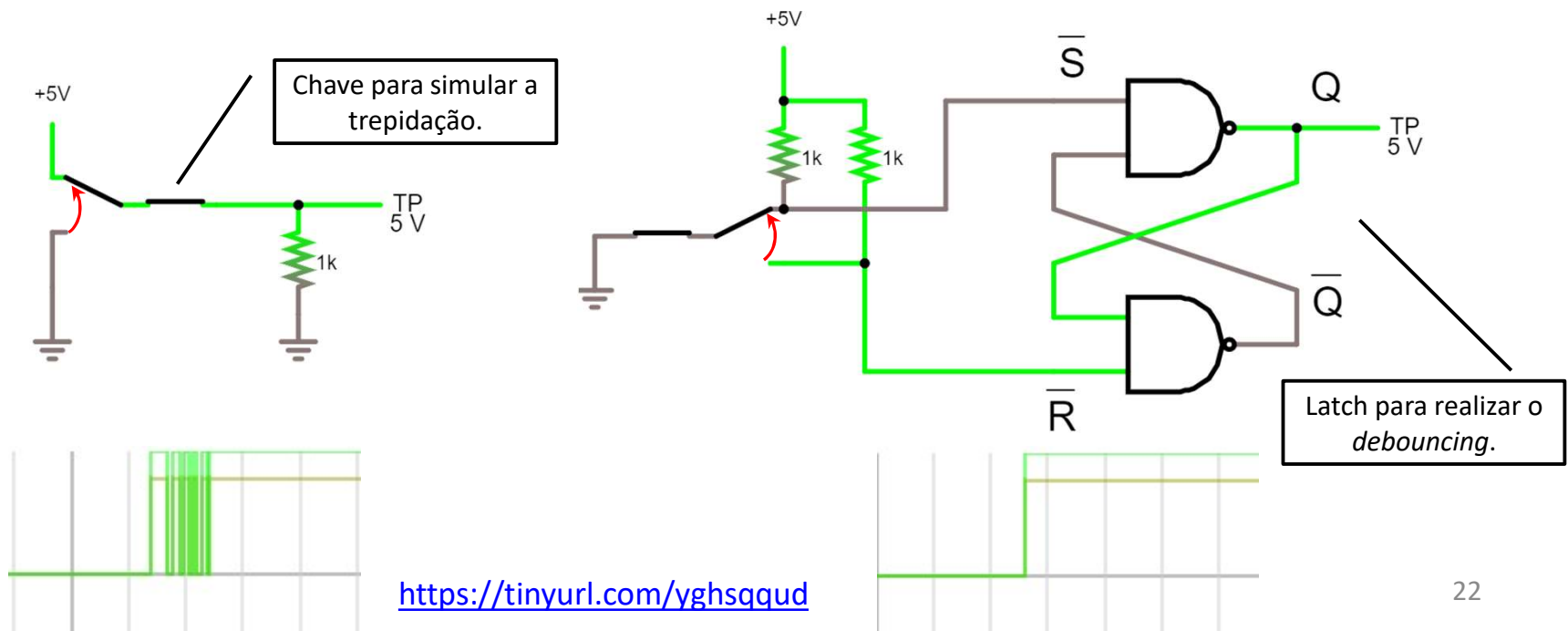
### Latch

- Uma aplicação típica de um *Latch*  $\bar{S}\text{--}\bar{R}$  é na eliminação de ruídos decorrentes do *bouncing* (*trepidação*) no momento que se comuta o contato de chaves mecânicas;
- Imaginem uma situação semelhante, em uma automação residencial, onde as portas possuem sensores que indicam se está aberta ou fechada;
- Normalmente são contatos que se encostam quando a porta é fechada, fazendo com que uma corrente elétrica seja conduzida indicando o estado atual;
- Ou em uma lógica negativa quando um circuito é interrompido;
- O que ocorre na prática é uma trepidação no momento do contato, fazendo com que uma sequência de sinais espúrios seja enviada ao sistema de monitoramento (uma avalanche de abre-fecha até estabilizar);

# Circuitos Sequenciais

## Latch

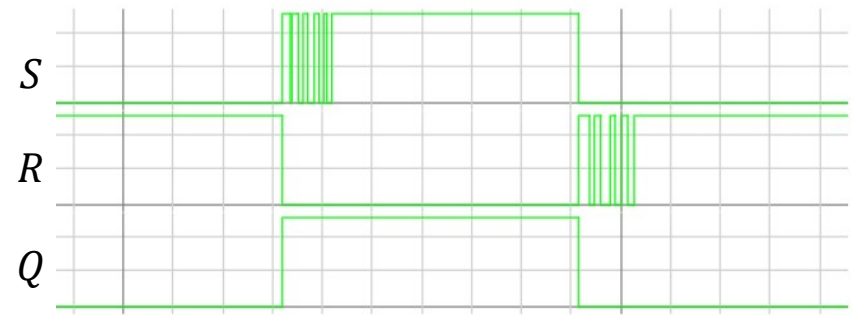
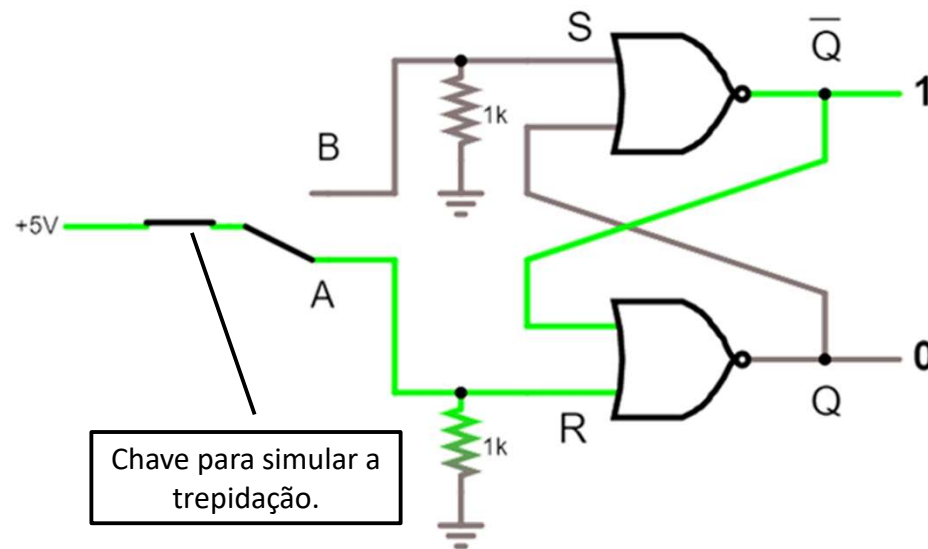
- Uma aplicação típica de um *Latch*  $\bar{S}$ – $\bar{R}$  é na eliminação de ruídos decorrentes do *bouncing* (*trepidação*) no momento que se comuta o contato de chaves mecânicas;



# Circuitos Sequenciais

## Latch

- Uma aplicação típica de um *Latch*  $\bar{S}$ – $\bar{R}$  é na eliminação de ruídos decorrentes do *bouncing* (trepidação) no momento que se comuta o contato de chaves mecânicas;

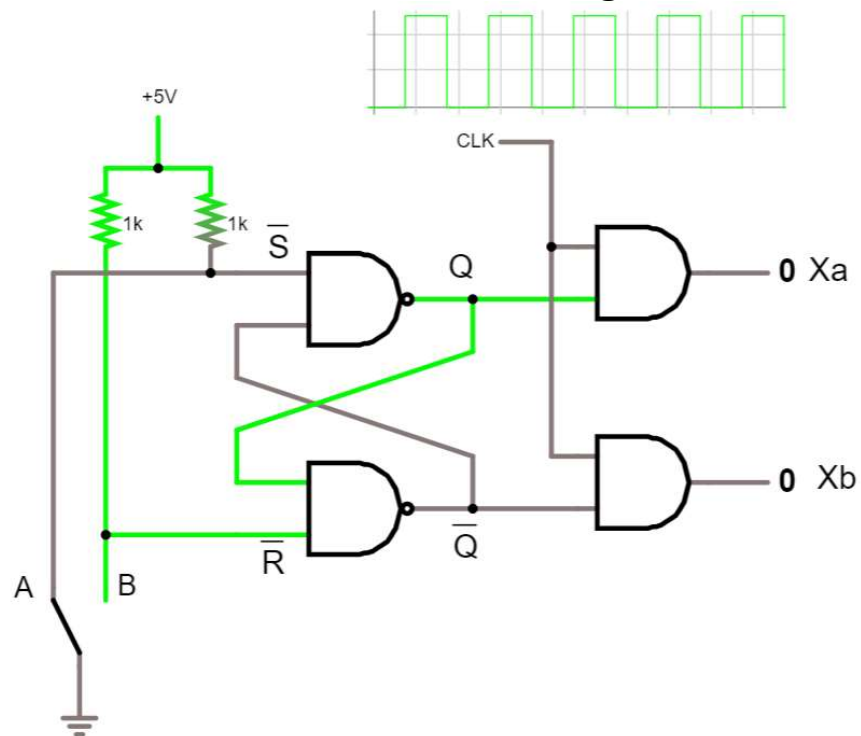


<https://tinyurl.com/yf3rnamo>

# Circuitos Sequenciais

## Latch

- Descreva o funcionamento do seguinte circuito:



Posição da Chave	$X_a$	$X_b$
<i>A</i>		
<i>B</i>		



## Circuitos Sequenciais

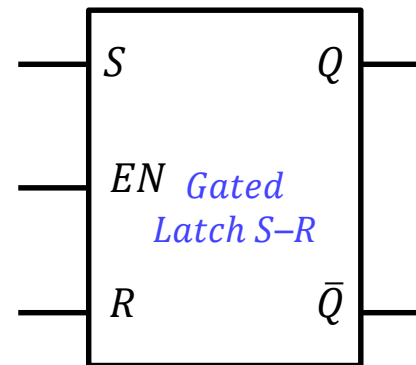
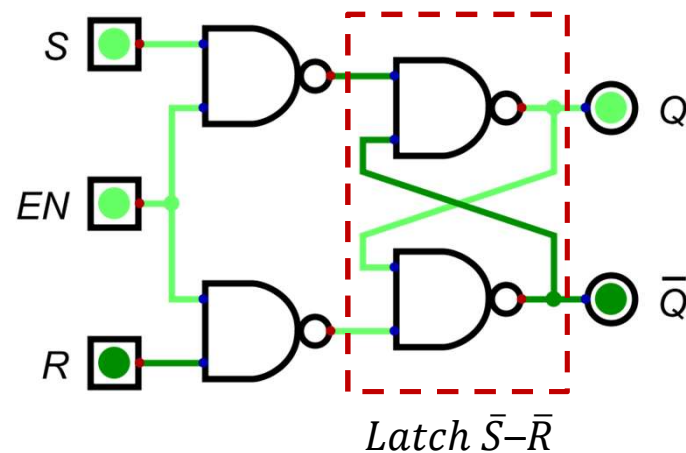
### Latch

- A família dos *Latches* inclui diversos tipos:
  - ✓ Basic *Latch S–R*;
  - ✓ *Gated Latch S–R* ou *Latch S–R com Enable* ou *Latch S–R Sensível a Nível*;
  - ✓ *Gated D Latch*;

# Circuitos Sequenciais

## Latch

- *Gated Latch S-R* ou *Latch S-R com Enable* ou *Latch S-R Sensível a Nível*:
  - ✓ Um *Gated Latch* requer uma entrada de habilitação, *EN* (*G* também é usado para designar uma entrada de habilitação);

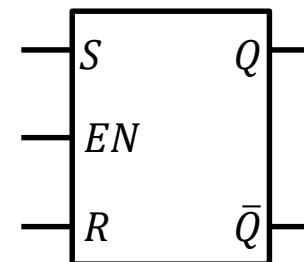


$EN$	$S$	$R$	$Q_i$	<i>Função</i>
1	0	1	0	Reset $Q$
1	1	0	1	Set $Q$
1	0	0	$Q_{i-1}$	Mantém $Q$
0	?	?	$Q_{i-1}$	Mantém $Q$

## Circuitos Sequenciais

### Latch

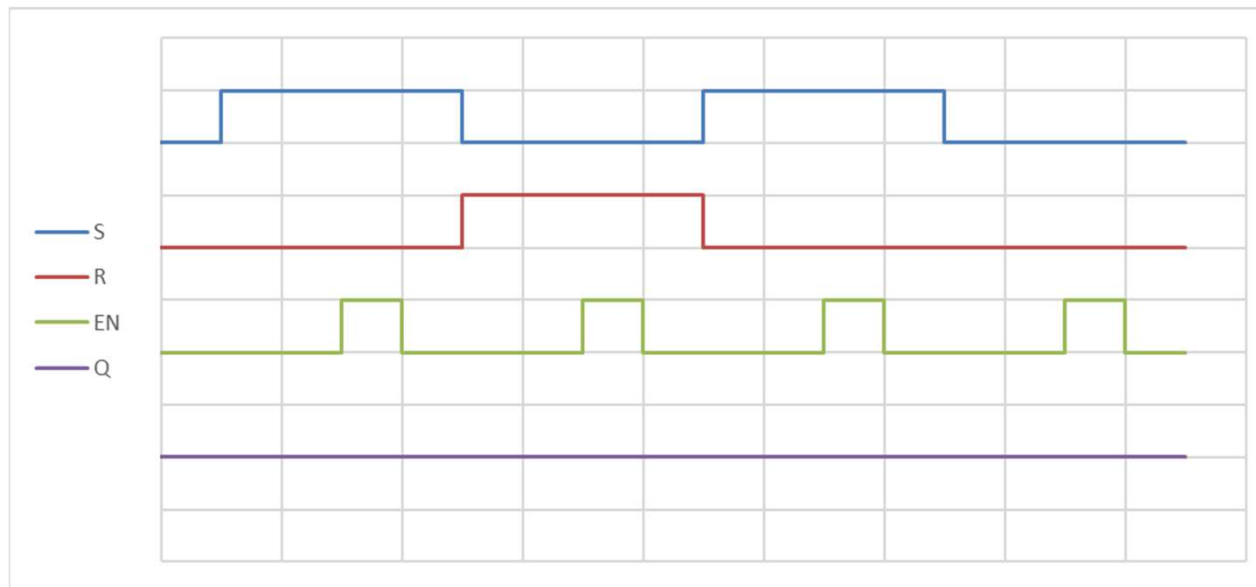
- *Gated Latch S–R* ou *Latch S–R com Enable* ou *Latch S–R Sensível a Nível*:
  - ✓ As entradas *S* e *R* controlam o estado desejado para a *Latch* quando um nível *HIGH* for aplicado à entrada *EN*.
  - ✓ O estado da *Latch* será modificado apenas quando *EN* estiver em *ALTO*;
  - ✓ Enquanto permanecer em *ALTO*, a saída será controlada pelas entradas *S* e *R*;
  - ✓ O estado *inválido* ocorre quando, estando *EN* em *HIGH*, *S* e *R* também estão em *HIGH*.



## Circuitos Sequenciais

### Latch

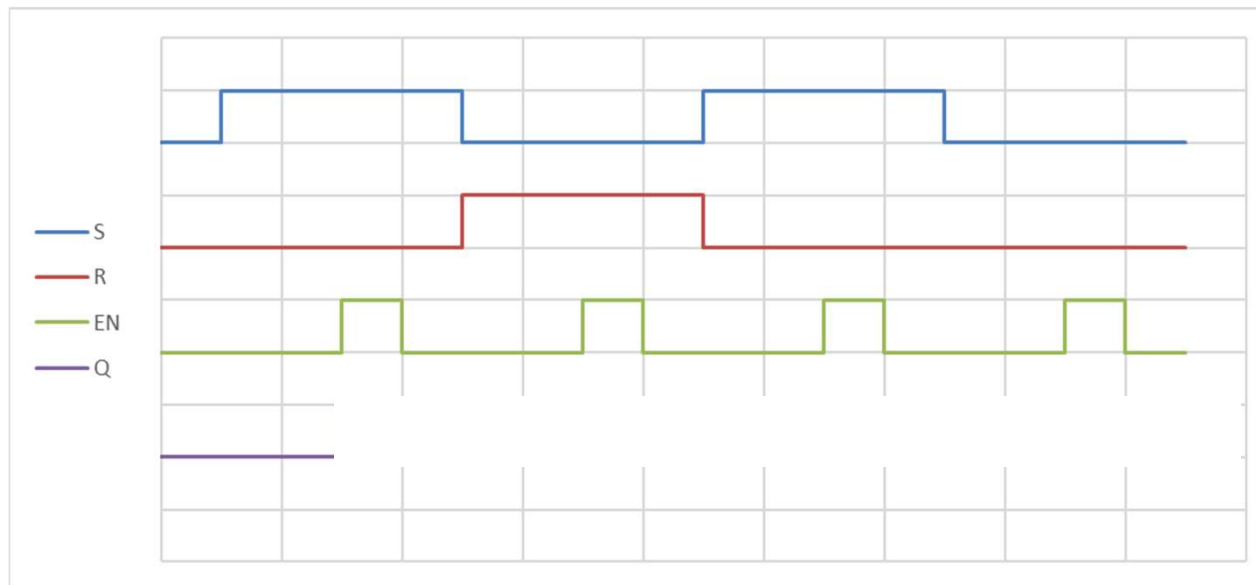
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

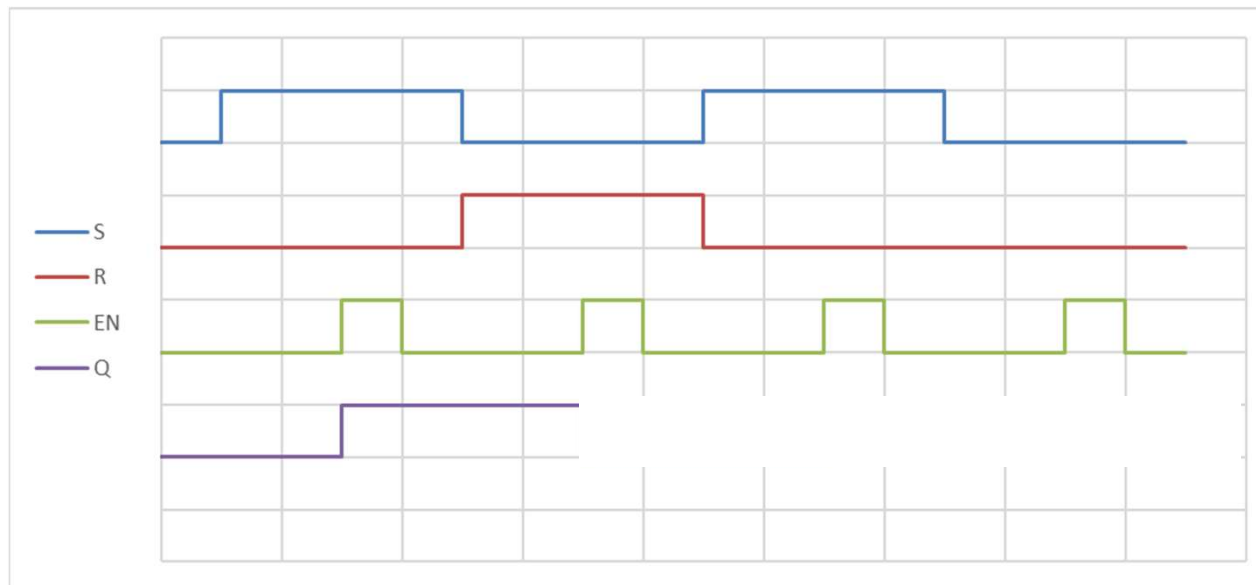
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

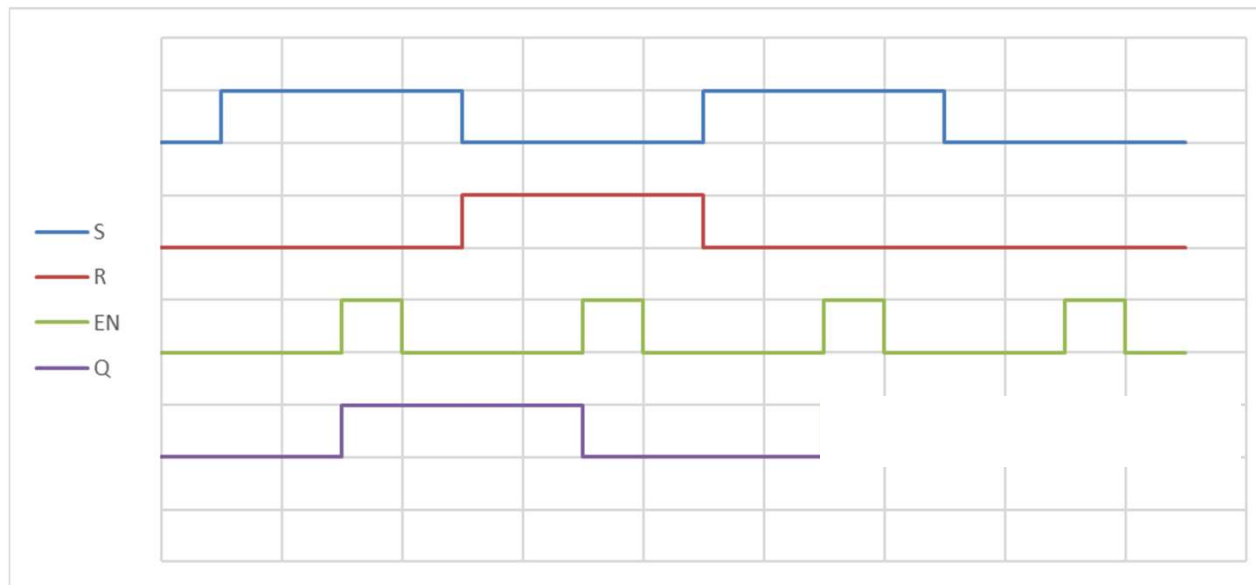
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

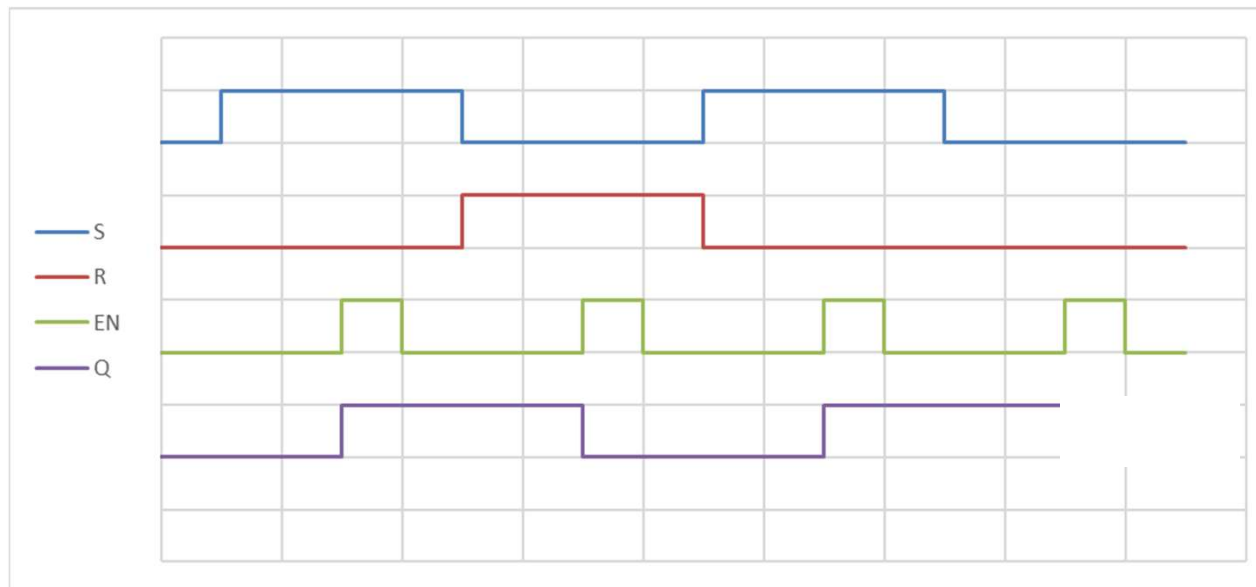
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:

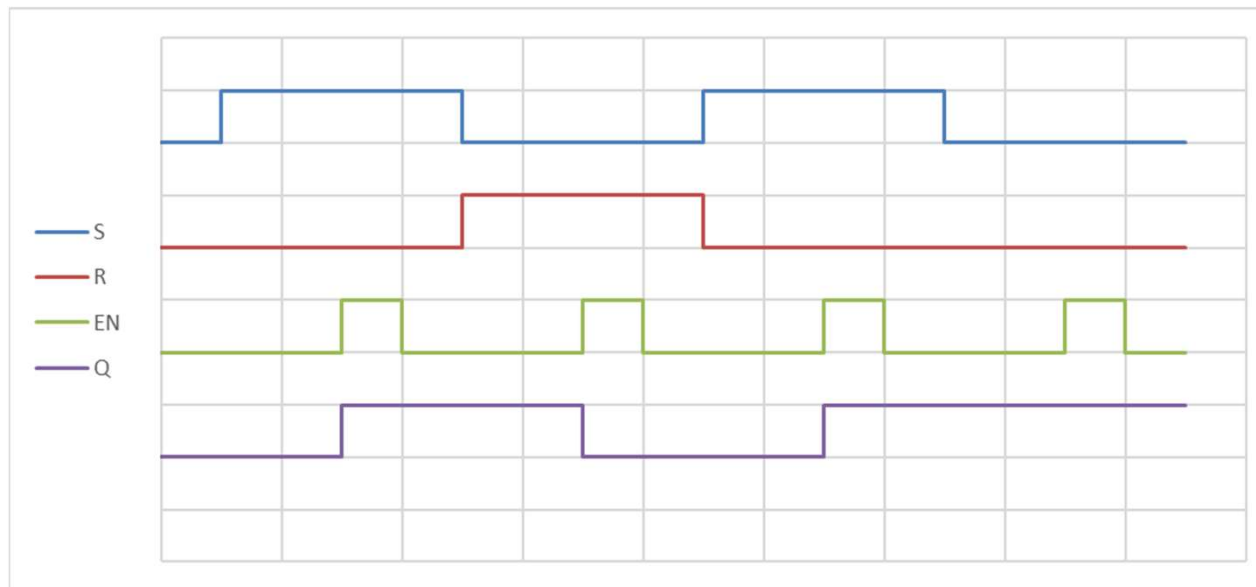




## Circuitos Sequenciais

### Latch

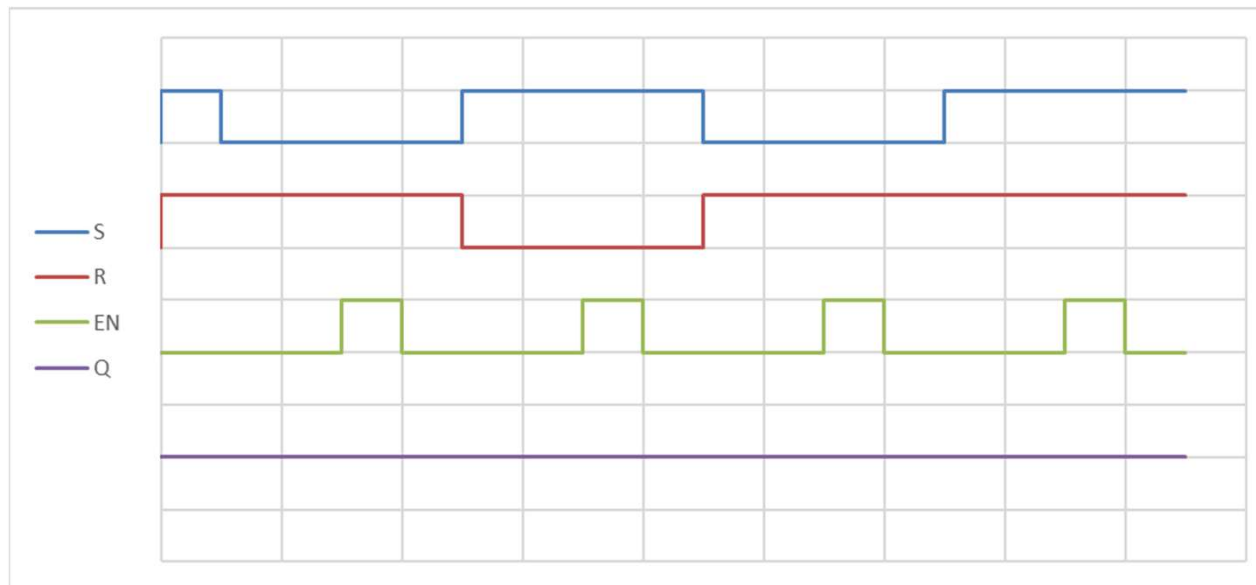
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

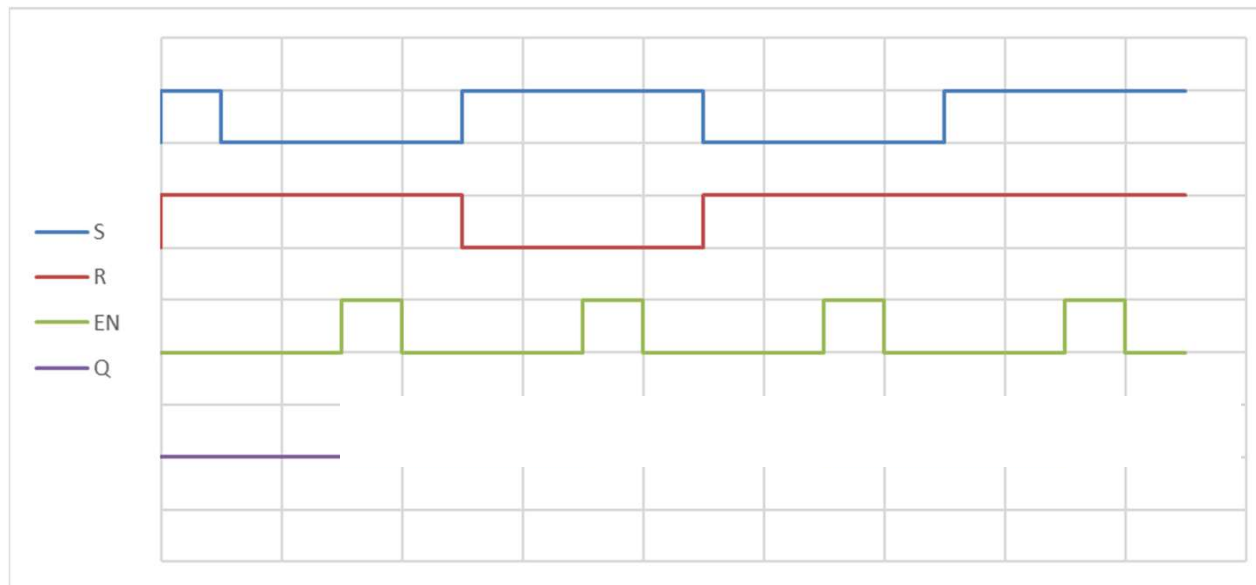
- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

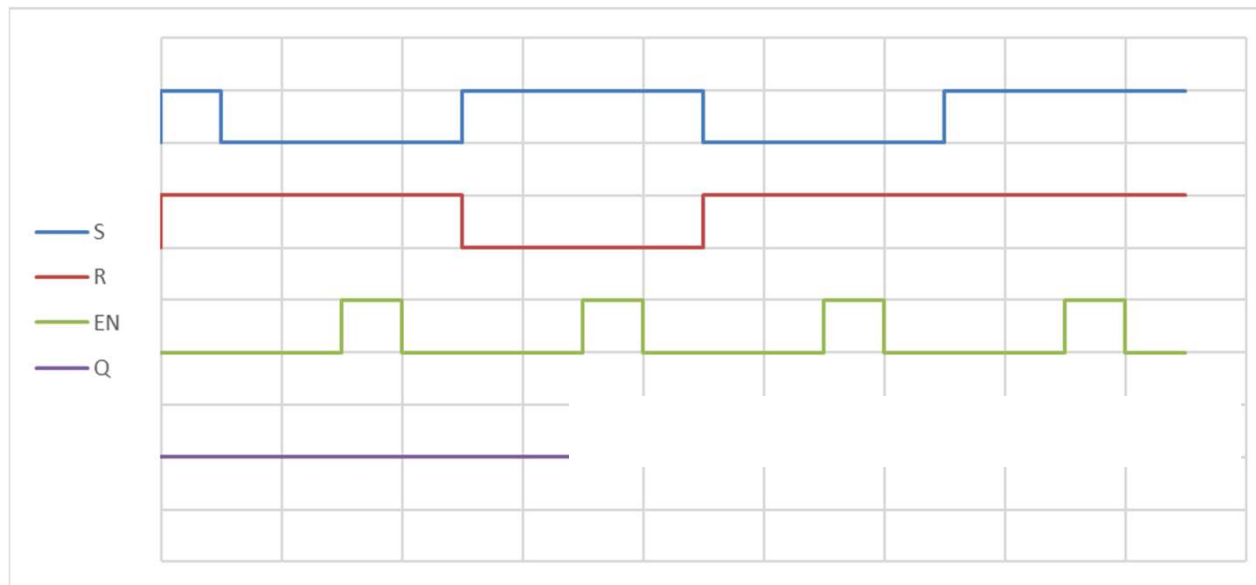
- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

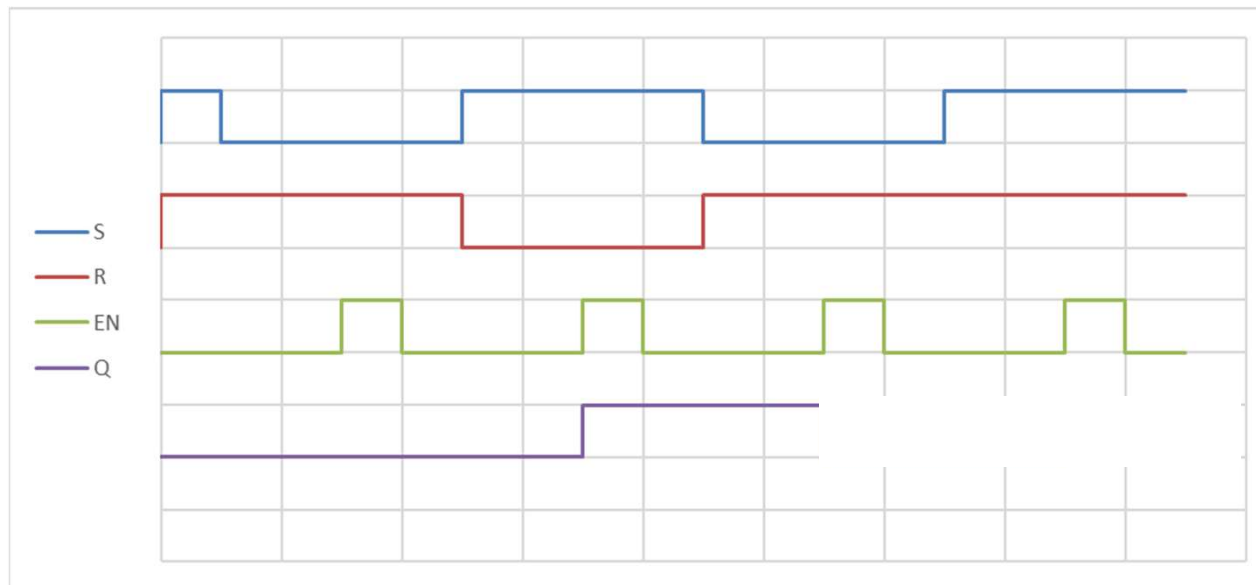
- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

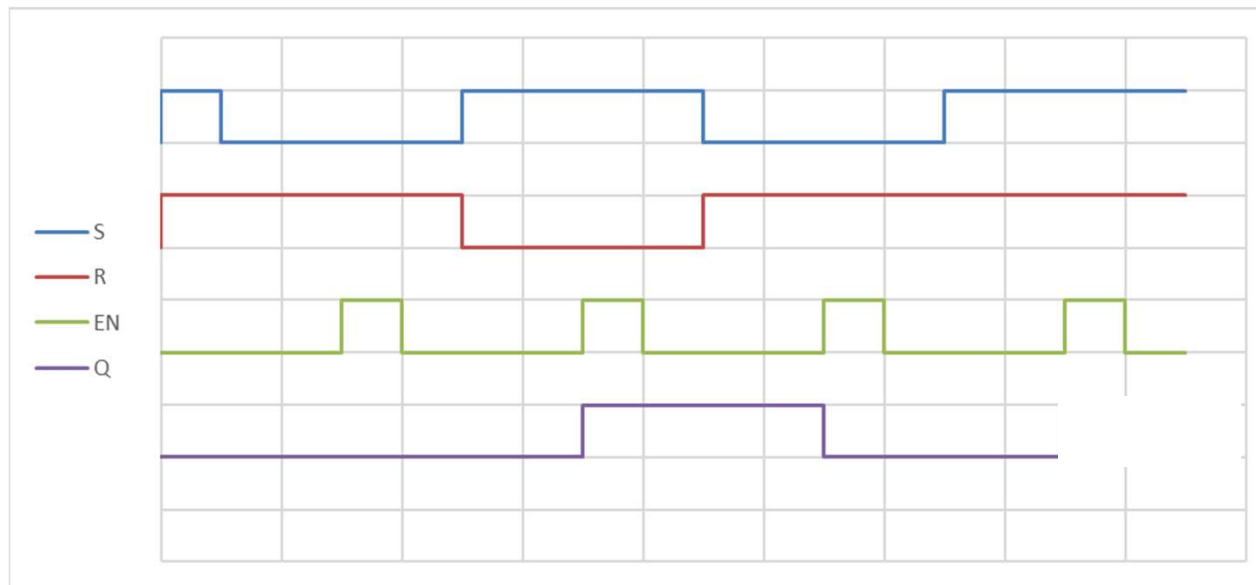
- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

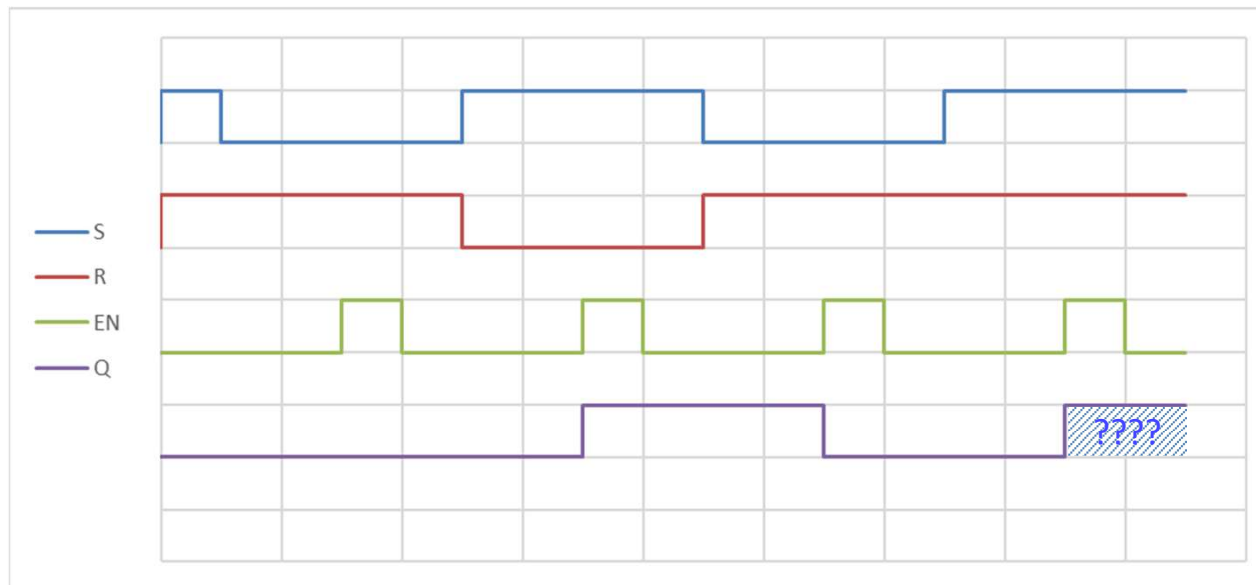
- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

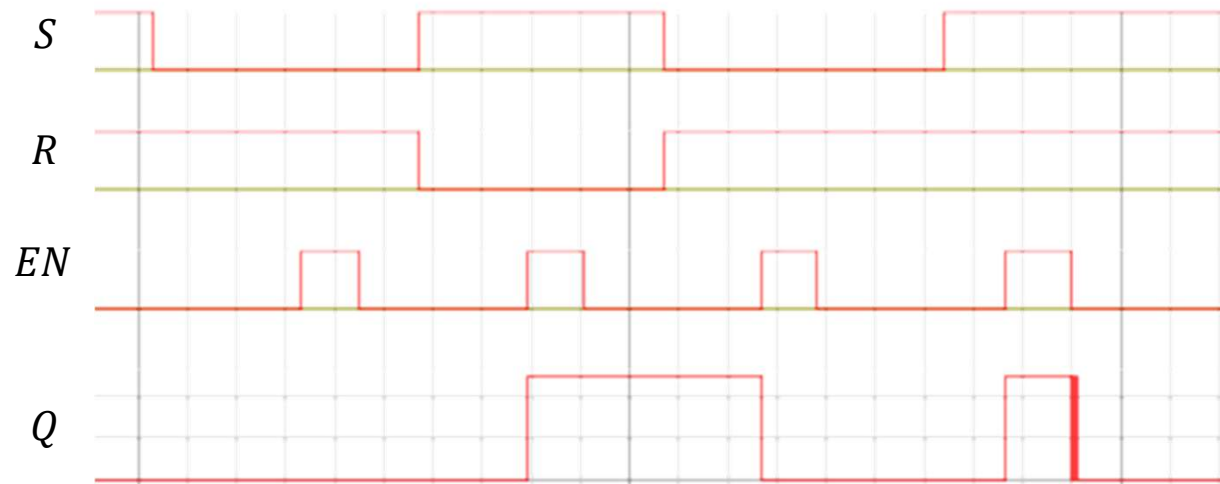
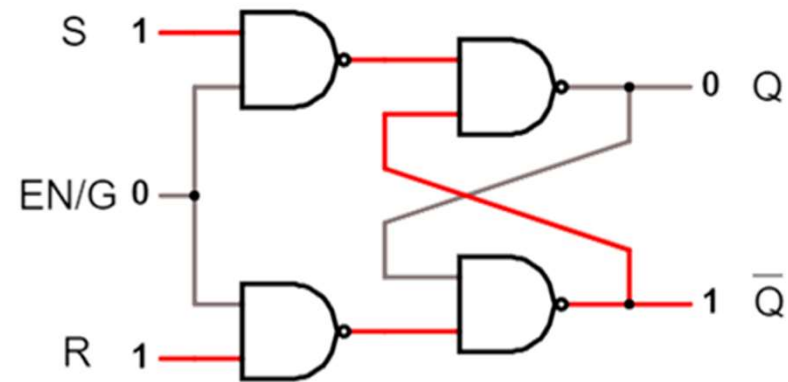
### Latch

- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $S$ ,  $R$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch



<https://tinyurl.com/yj7ncry3>



## Circuitos Sequenciais

### Latch

iii. Considere o seguinte circuito lógico:



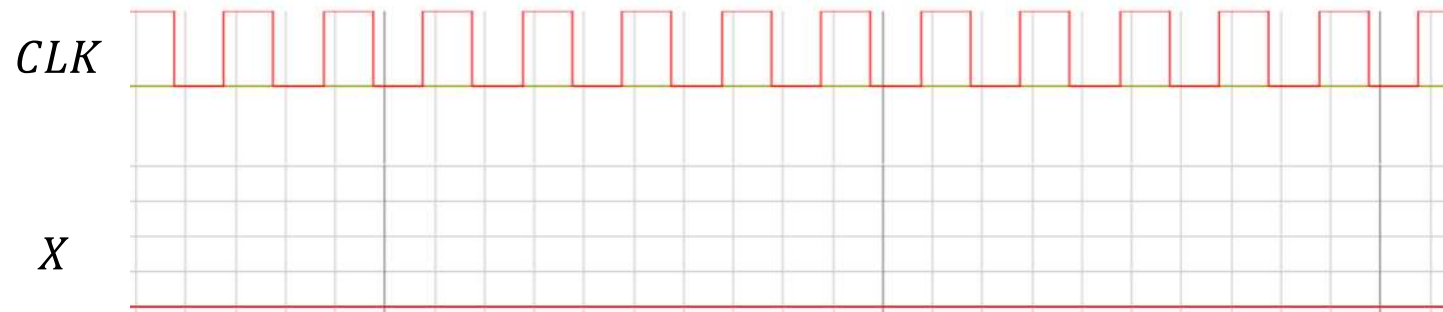
Pede-se:

- ✓ Implementar no circuitjs;
- ✓ Obter o gráfico da entrada do clock ( $CLK$ ) e da saída  $X$ ;
- ✓ Explicar o comportamento;
- ✓ Implementar em outra ferramenta de simulação (digital, logisim,...) e comparar o comportamento;

## Circuitos Sequenciais

### Latch

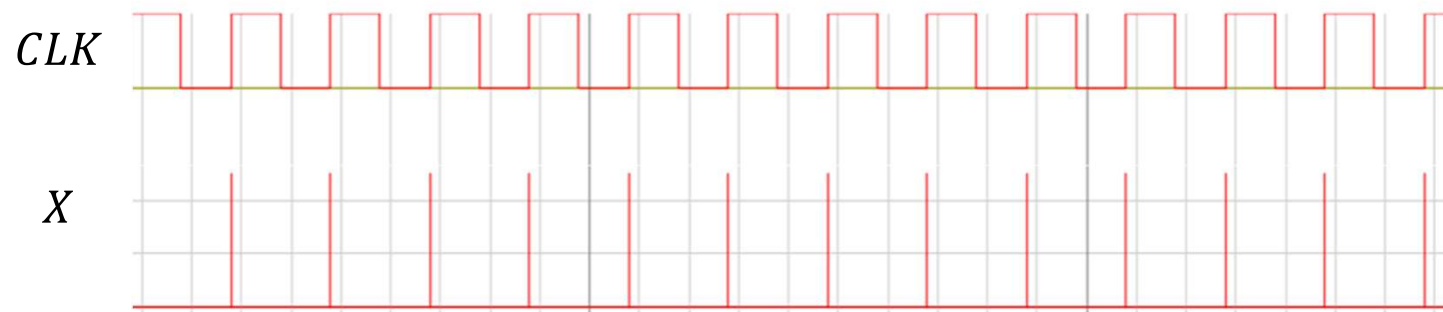
iii. Considere o seguinte circuito lógico:



## Circuitos Sequenciais

### Latch

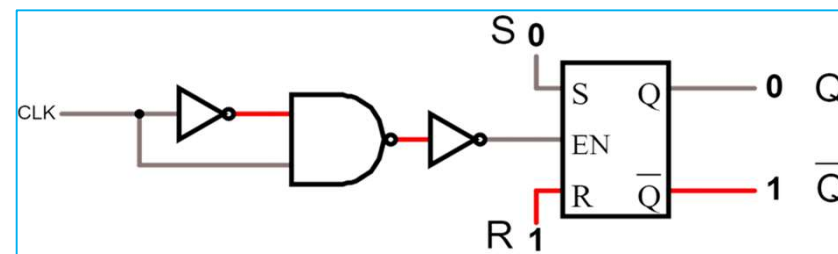
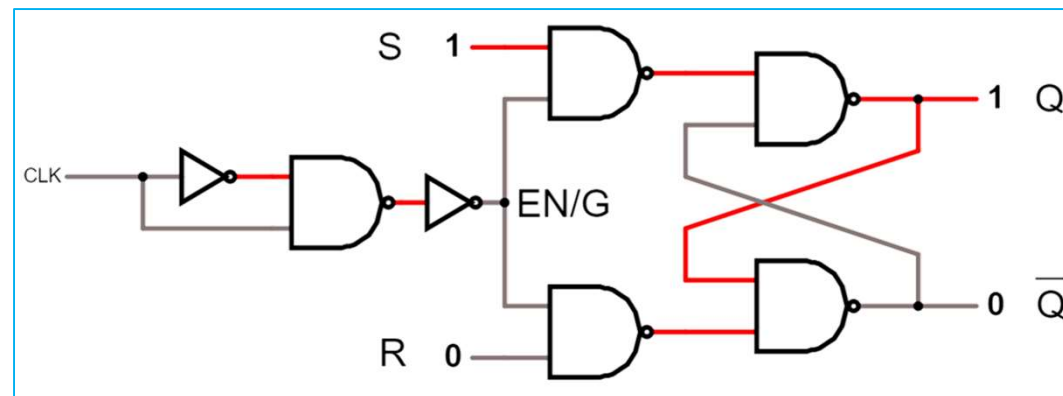
iii. Considere o seguinte circuito lógico:



## Circuitos Sequenciais

### Latch

- iv. Considerar o seguinte circuito que combina o circuito anterior com o *Gated Latch S-R*:

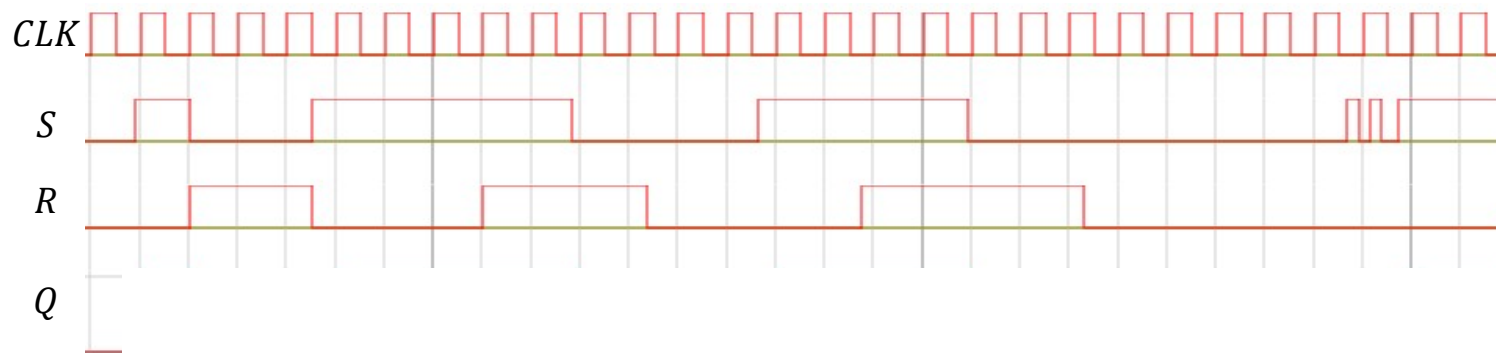


# Circuitos Sequenciais

## Latch

Pede-se:

- ✓ Implementar no circuitjs;
- ✓ Obter o gráfico da entrada do clock ( $CLK$ ) e dos pontos  $EN$ ,  $S$ ,  $R$  e  $Q$ ;
- ✓ Efetuar alguns comandos em  $S$  e  $R$  (exemplo abaixo) e observar a resposta em  $Q$ ;
- ✓ Explicar o comportamento;
- ✓ Implementar em outra ferramenta de simulação (digital, logisim,...) e comparar o comportamento;

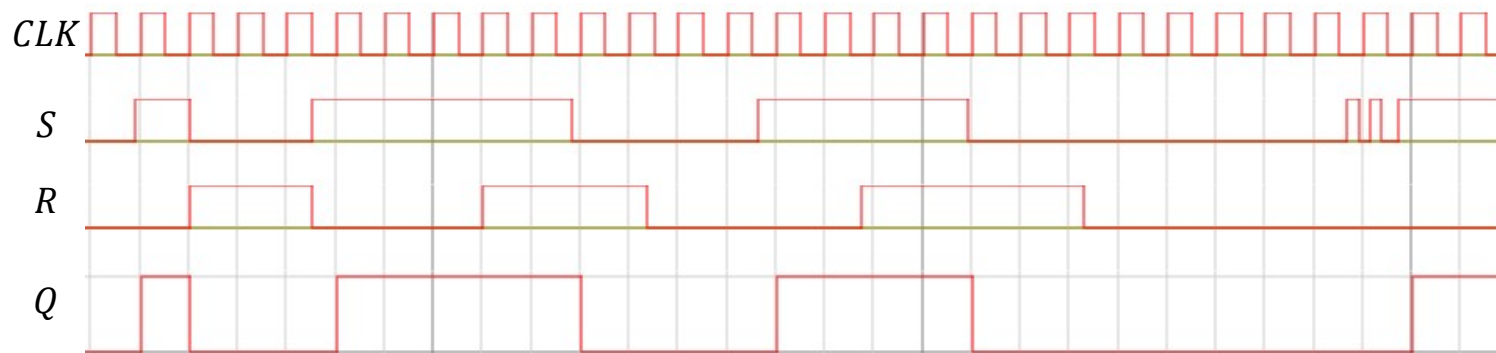


# Circuitos Sequenciais

## Latch

Pede-se:

- ✓ Implementar no circuitjs;
- ✓ Obter o gráfico da entrada do clock ( $CLK$ ) e dos pontos  $EN$ ,  $S$ ,  $R$  e  $Q$ ;
- ✓ Efetuar alguns comandos em  $S$  e  $R$  (exemplo abaixo) e observar a resposta em  $Q$ ;
- ✓ Explicar o comportamento;
- ✓ Implementar em outra ferramenta de simulação (digital, logisim,...) e comparar o comportamento;

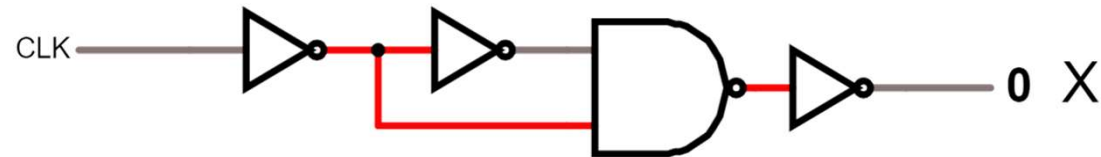


<https://tinyurl.com/ygdr3ezy>

## Circuitos Sequenciais

### Latch

iv. Considere o seguinte circuito lógico:



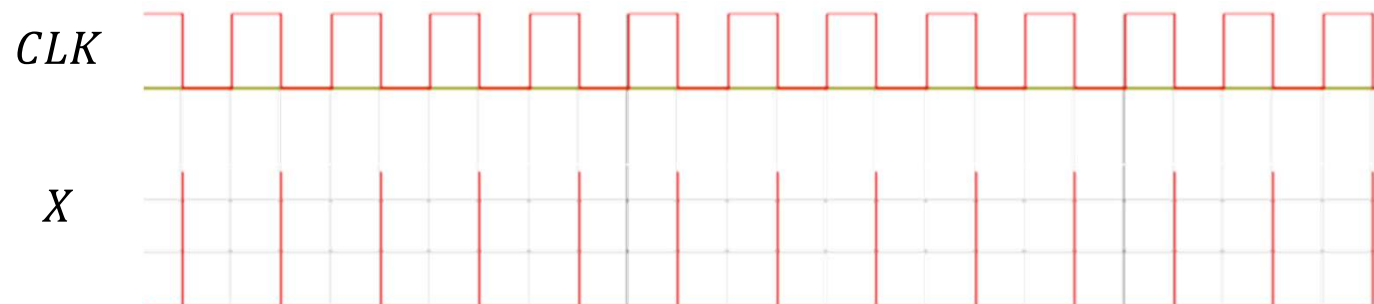
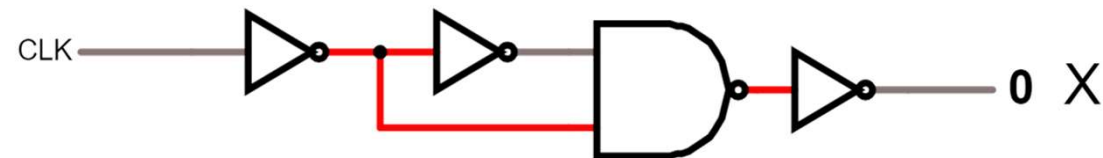
Pede-se:

- ✓ Implementar no circuitjs;
- ✓ Obter o gráfico da entrada do clock (*CLK*) e da saída *X*;
- ✓ Explicar o comportamento;
- ✓ Implementar em outra ferramenta de simulação (digital, logisim,...) e comparar o comportamento;

# Circuitos Sequenciais

## Latch

iv. Considere o seguinte circuito lógico:





## Circuitos Sequenciais

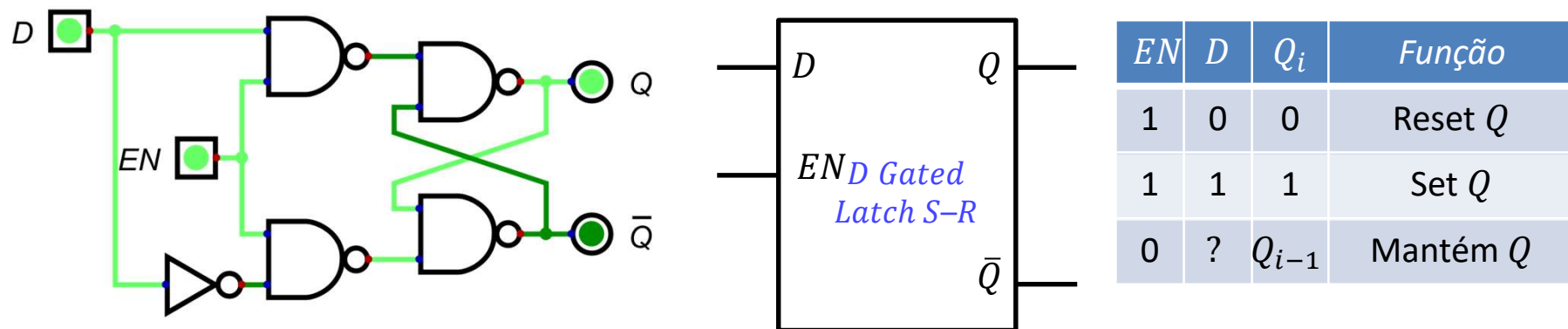
### Latch

- A família dos *Latches* inclui diversos tipos:
  - ✓ Basic *Latch S–R*;
  - ✓ *Gated Latch S–R* ou *Latch S–R com Enable* ou *Latch S–R* Sensível a Nível;
  - ✓ *Gated D Latch*;

## Circuitos Sequenciais

### Latch

- *Gated D Latch* :
  - ✓ A principal diferença com o *Latch S–R* está na entrada única além do *EN*;



## Circuitos Sequenciais

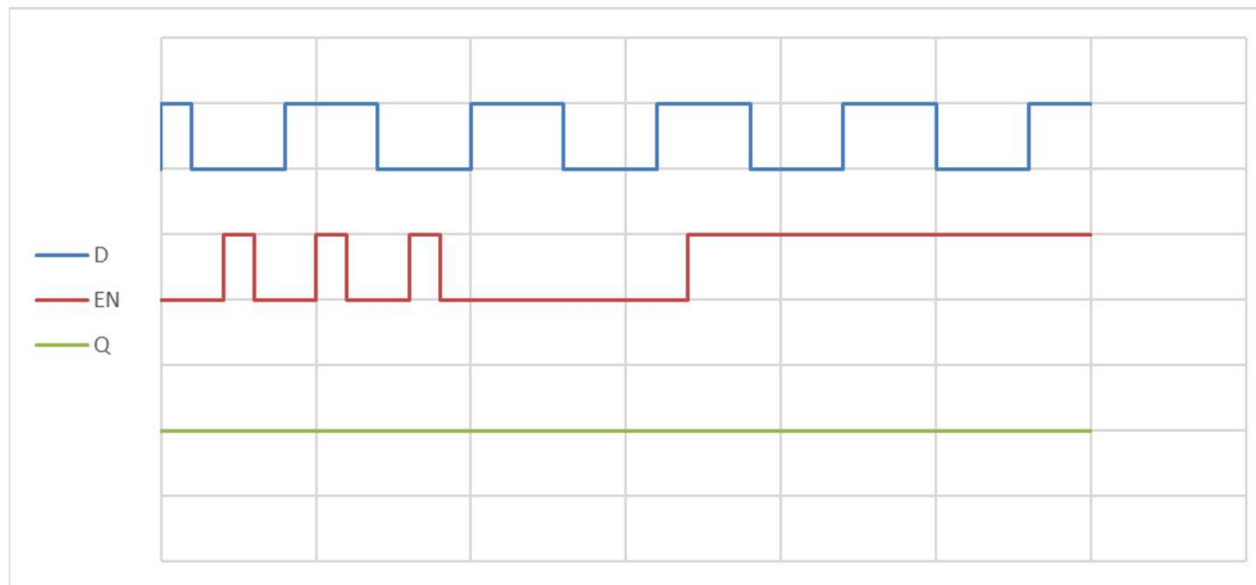
### Latch

- *Gated D Latch* :
  - ✓ A entrada é chamada de entrada *D* (dados);
  - ✓ Quando a entrada *D* está em *HIGH* e a entrada *EN* em *HIGH*, o *Latch* será ativado;
  - ✓ Quando a entrada *D* estiver em *LOW* e *EN* em *HIGH*, o *Latch* será reinicializado - *Reset* (a saída *Q* segue a entrada *D* quando *EN* está em *HIGH*);
  - ✓ Não há a preocupação de evitar o estado  $S = 1$  e  $R = 1$ ;

## Circuitos Sequenciais

### Latch

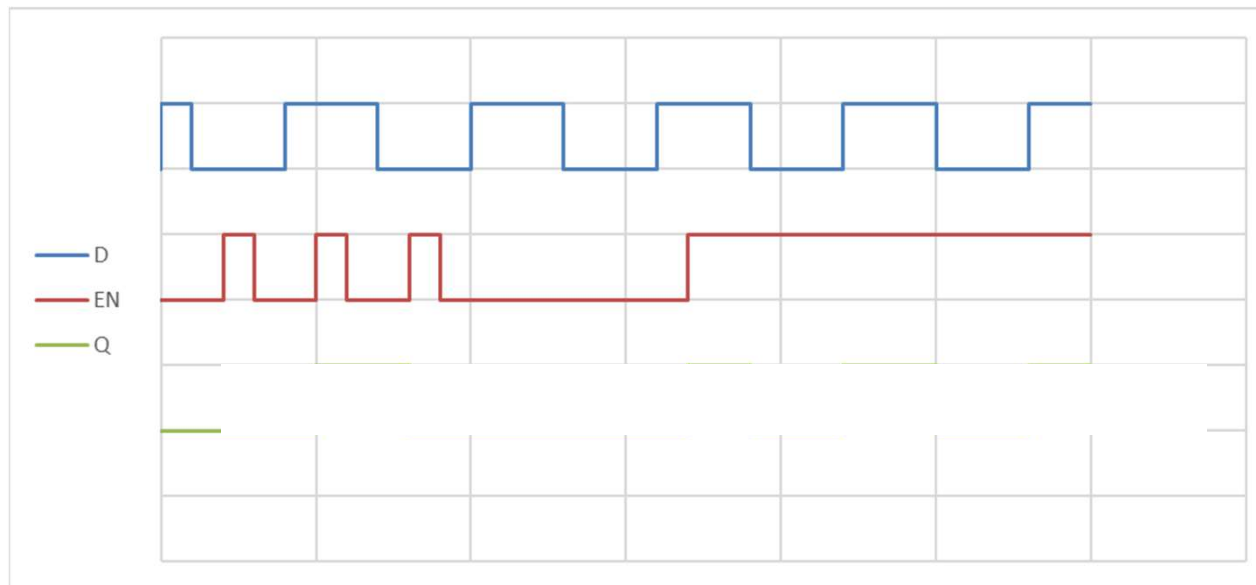
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

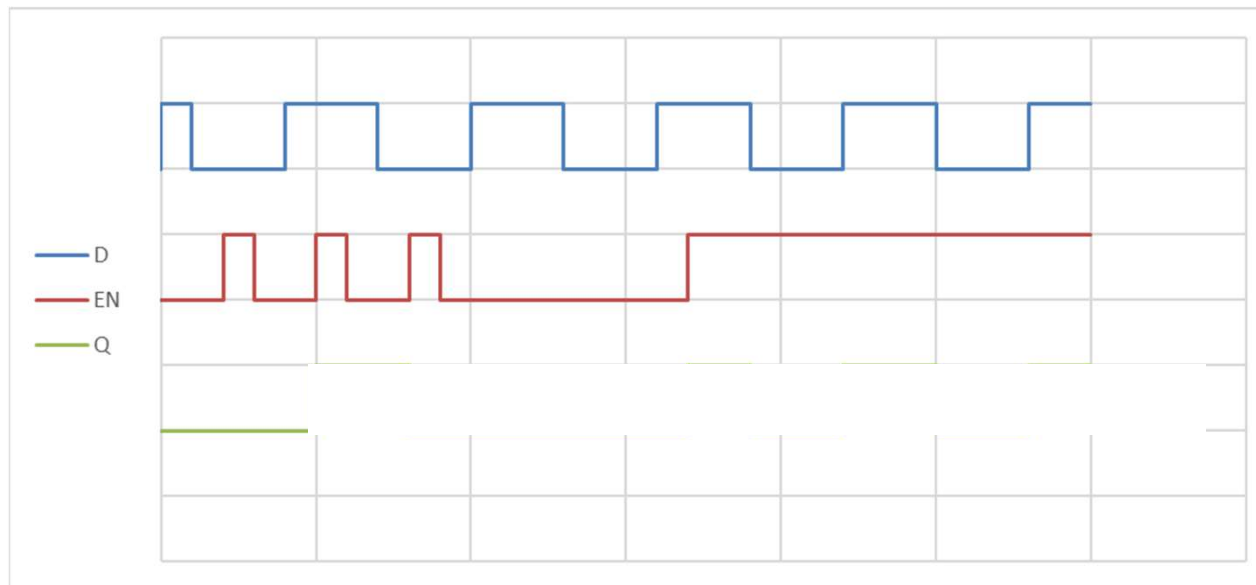
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

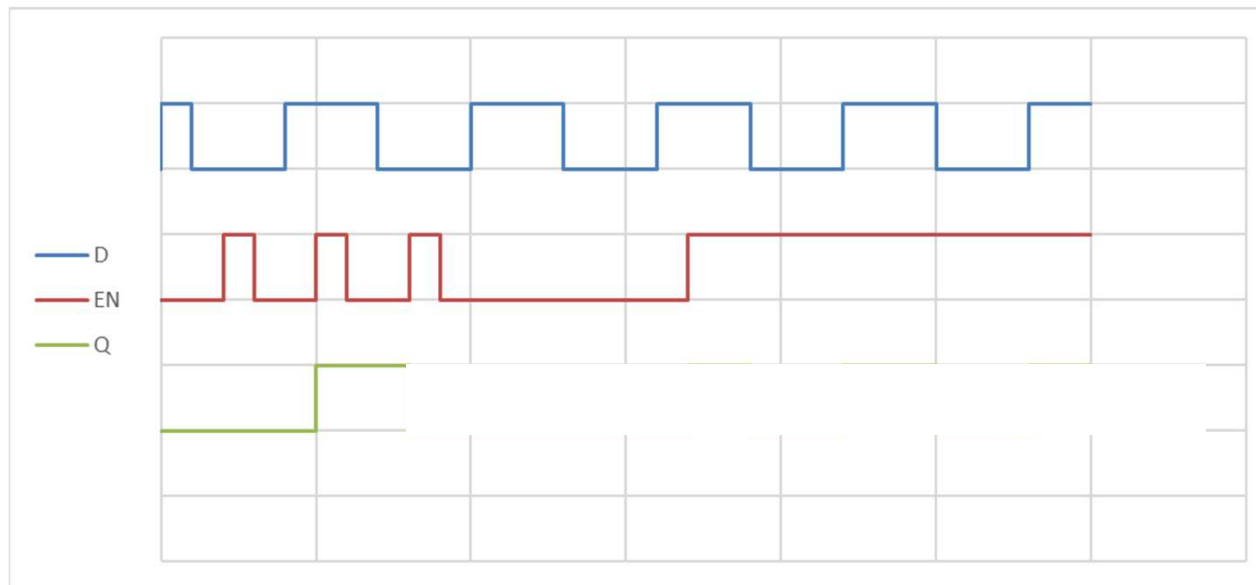
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

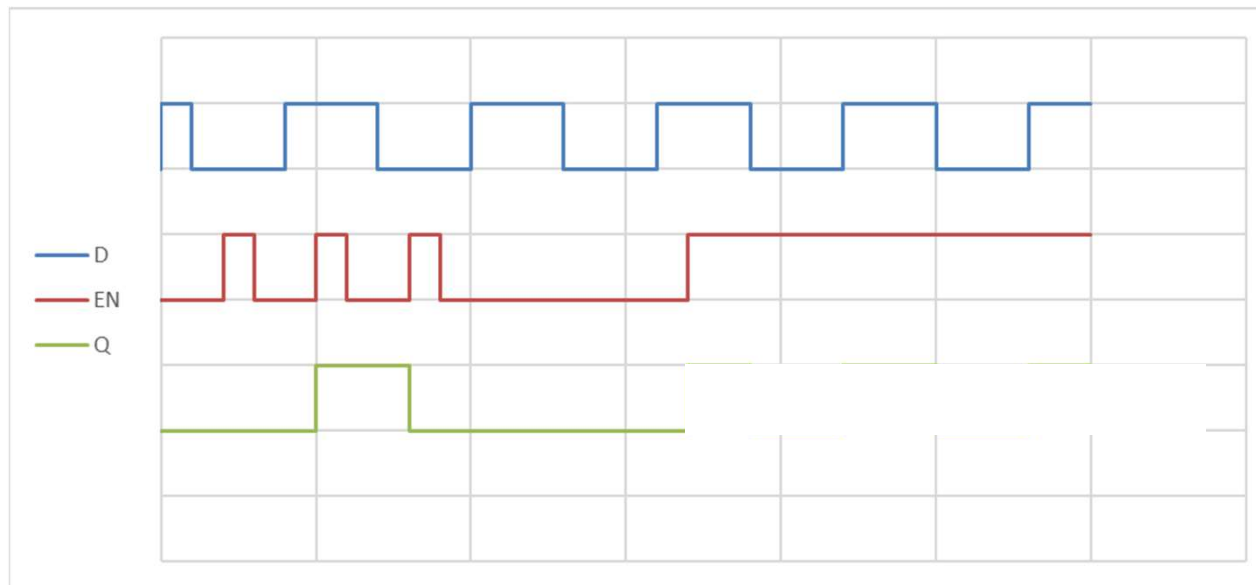
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:

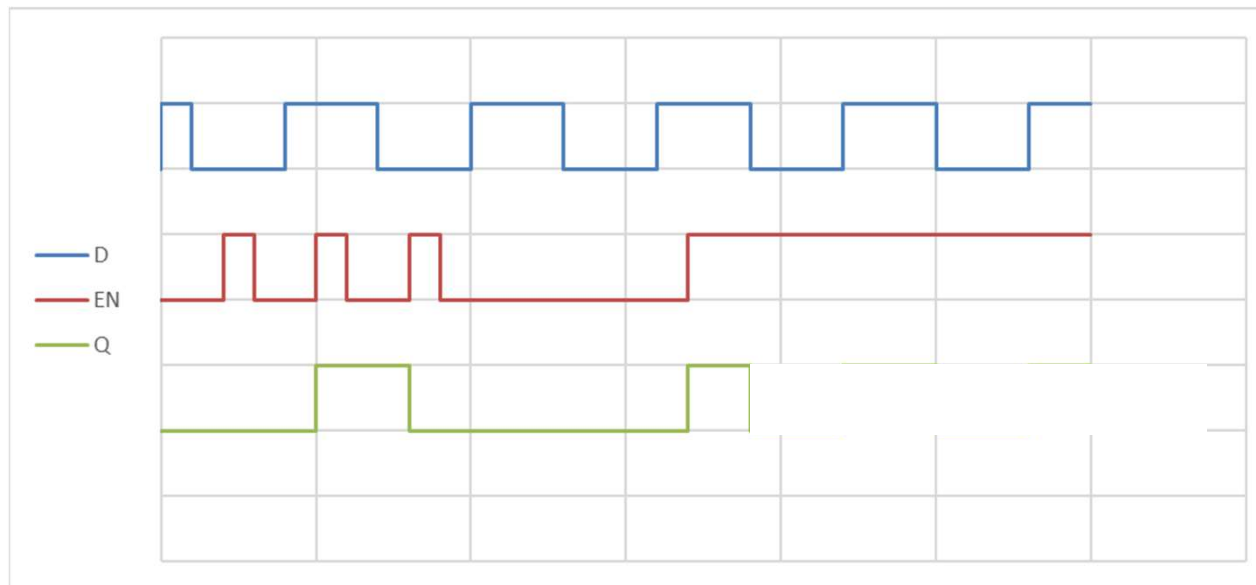




## Circuitos Sequenciais

### Latch

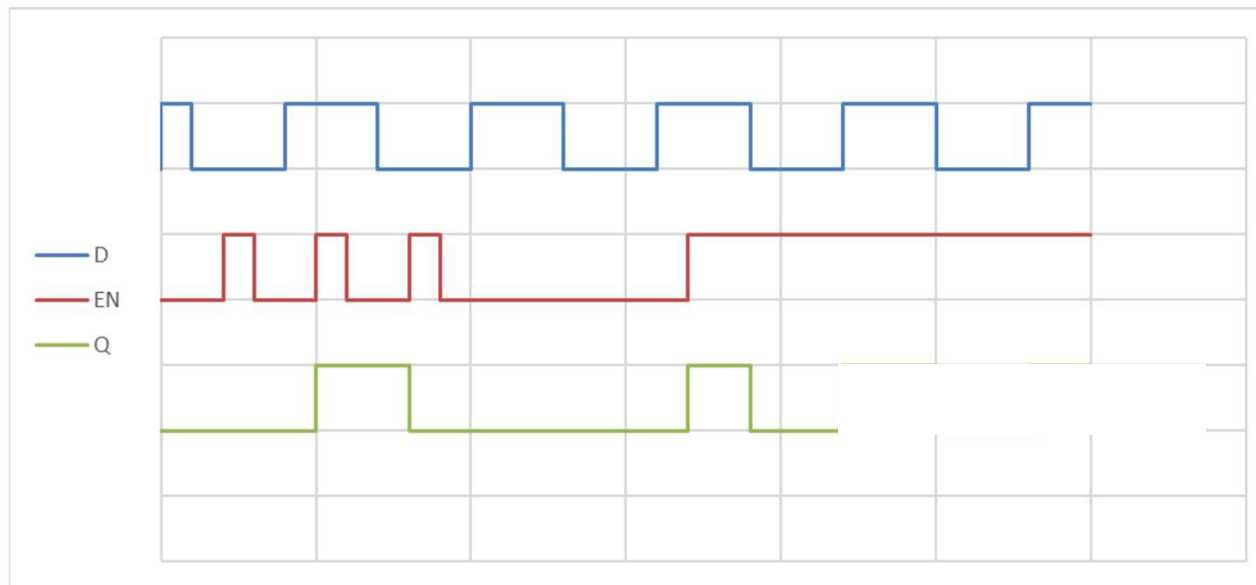
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

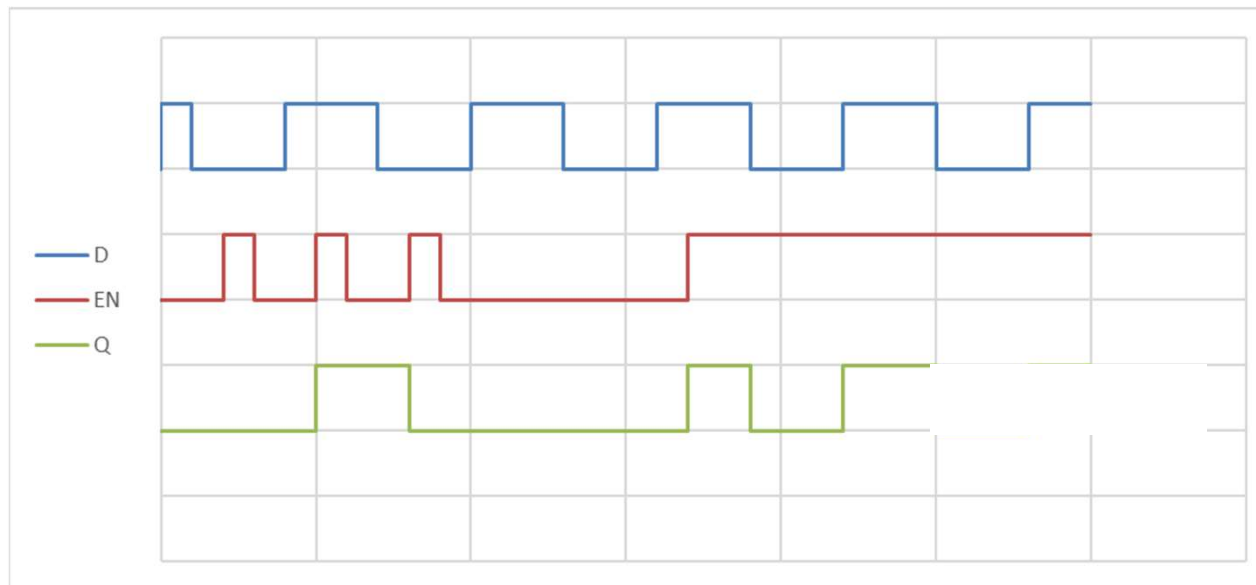
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

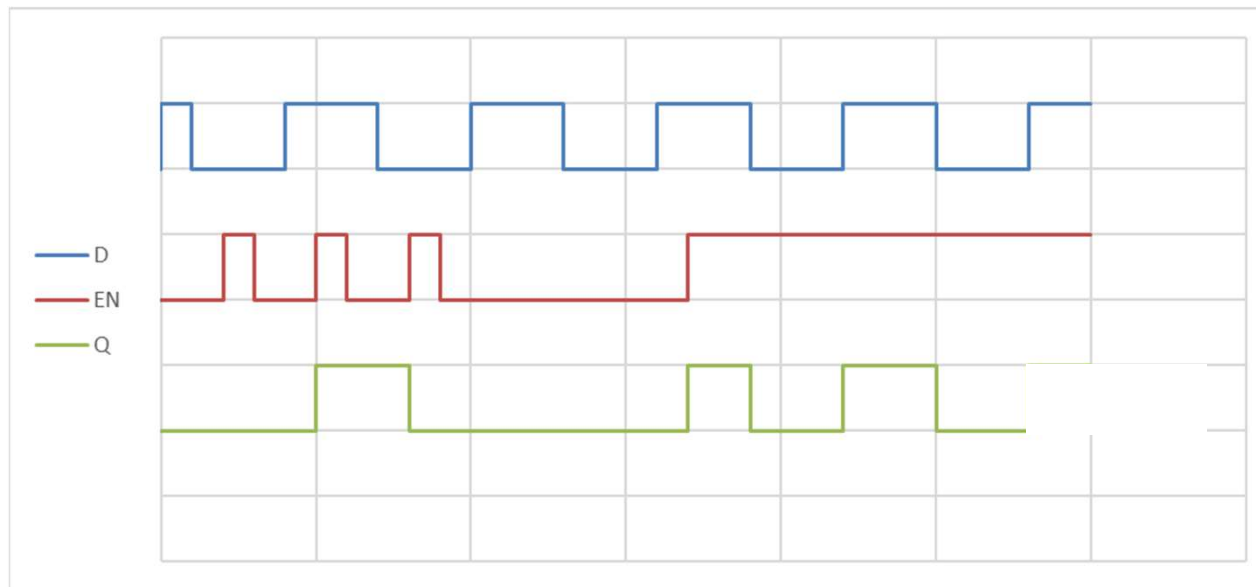
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

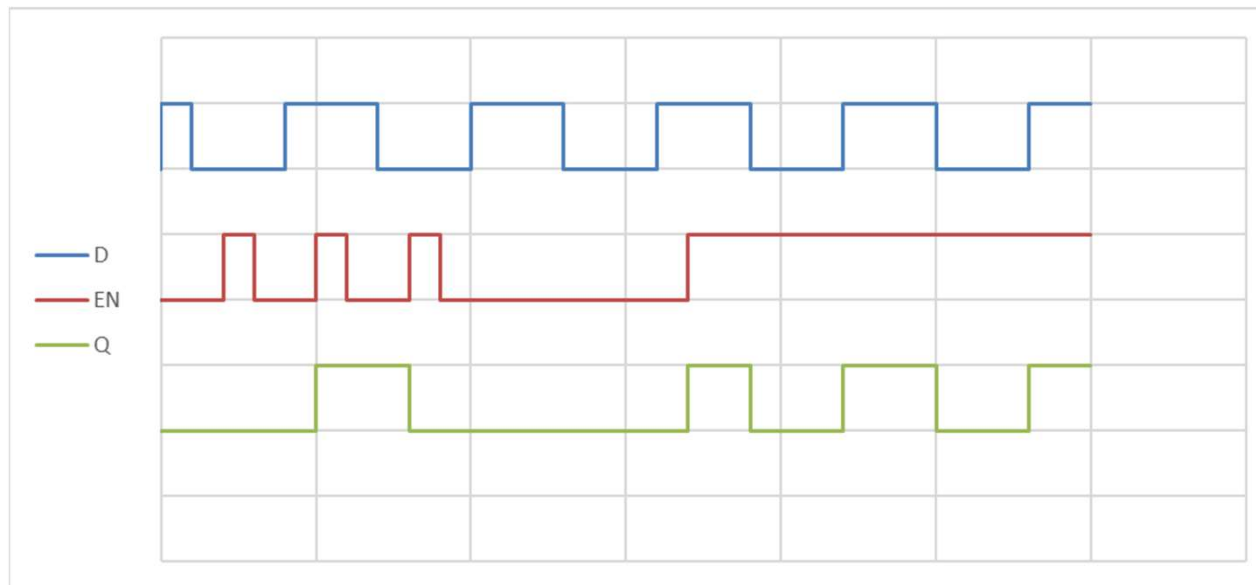
- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:



## Circuitos Sequenciais

### Latch

- i. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:

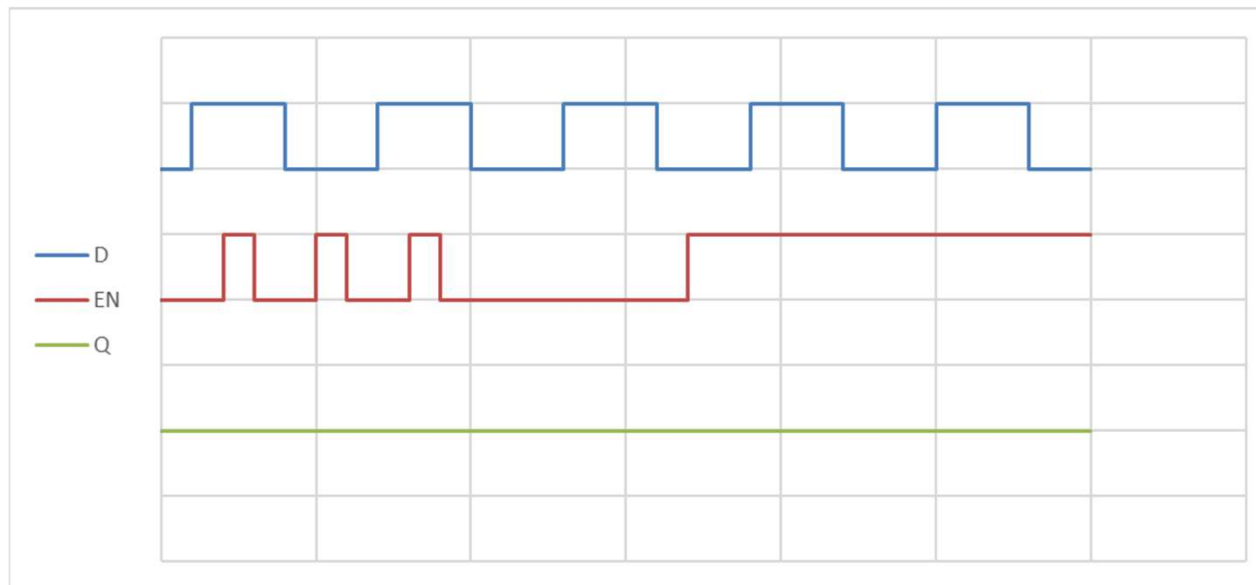


<https://tinyurl.com/ydtg7vza>

## Circuitos Sequenciais

### Latch

- ii. Determine a forma de onda para a saída  $Q$ , considerando as entradas  $D$  e  $EN$  indicadas no gráfico:

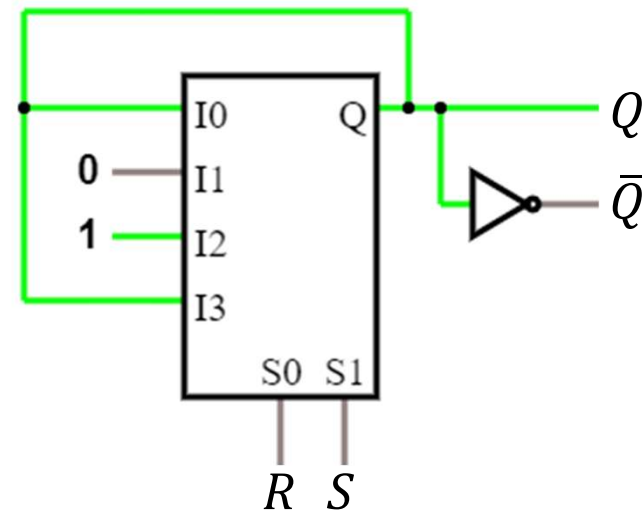
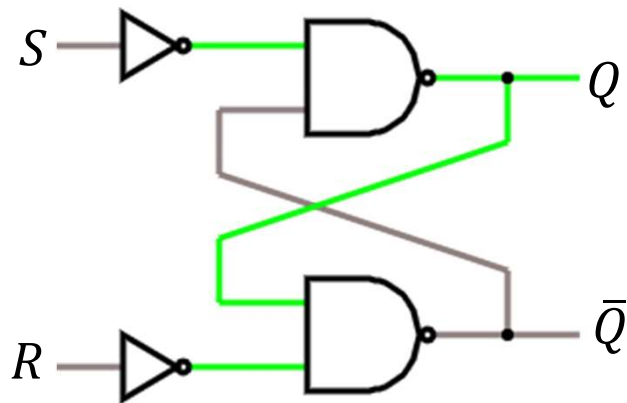


<https://tinyurl.com/ydtg7vza>

## Circuitos Sequenciais

### Latch

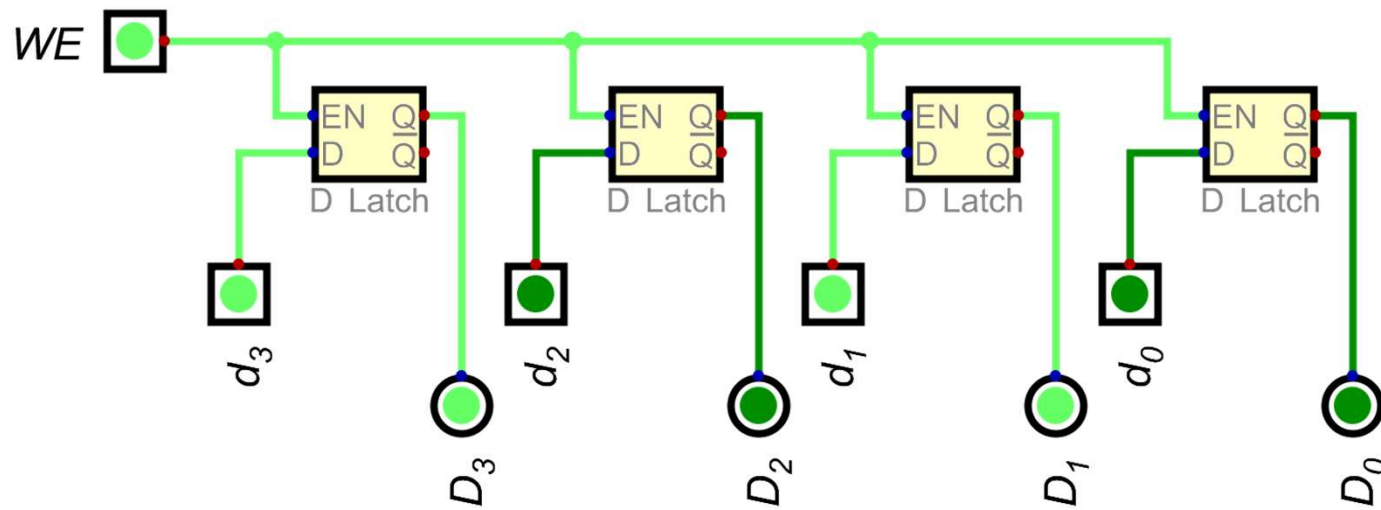
- iii. Utilize tabela verdade para o funcionamento desses circuitos. Qual a relação com um *Latch S-R*? O que acontece para cada circuito quando  $S = R = 1$ ?



## Circuitos Sequenciais

### Latch

- iv. Implementar um Registrador de armazenamento de 4 bits utilizando *D Latch*:

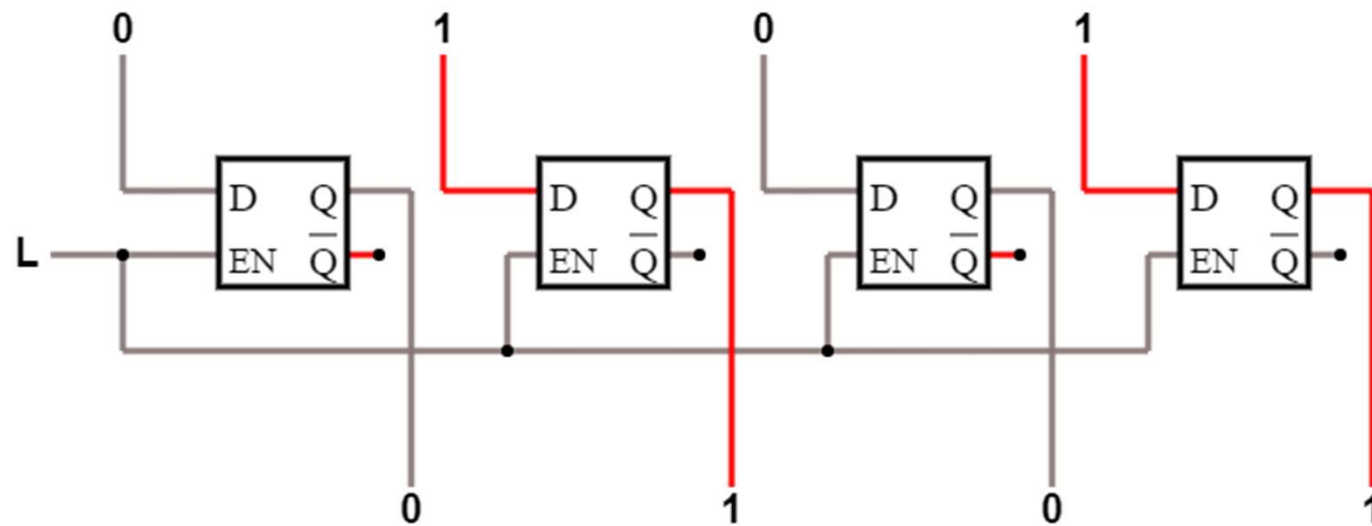




## Circuitos Sequenciais

### Latch

iv. Implementar um Registrador de armazenamento de 4 bits utilizando *D Latch*:

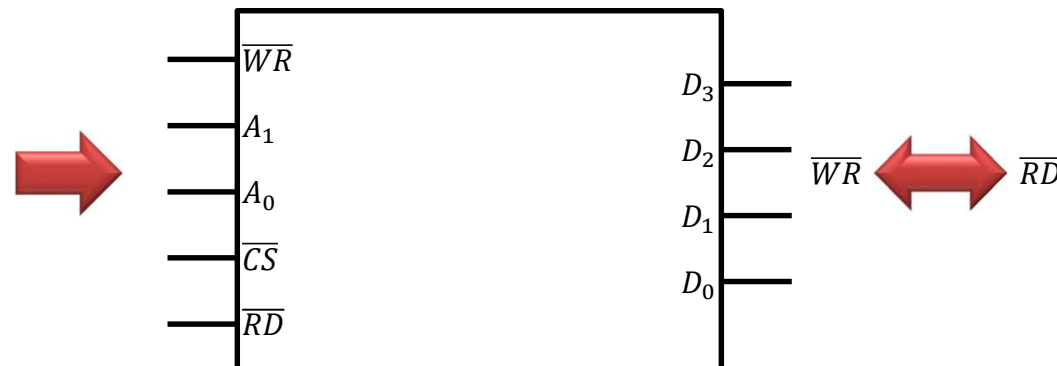


## Circuitos Sequenciais

### Latch

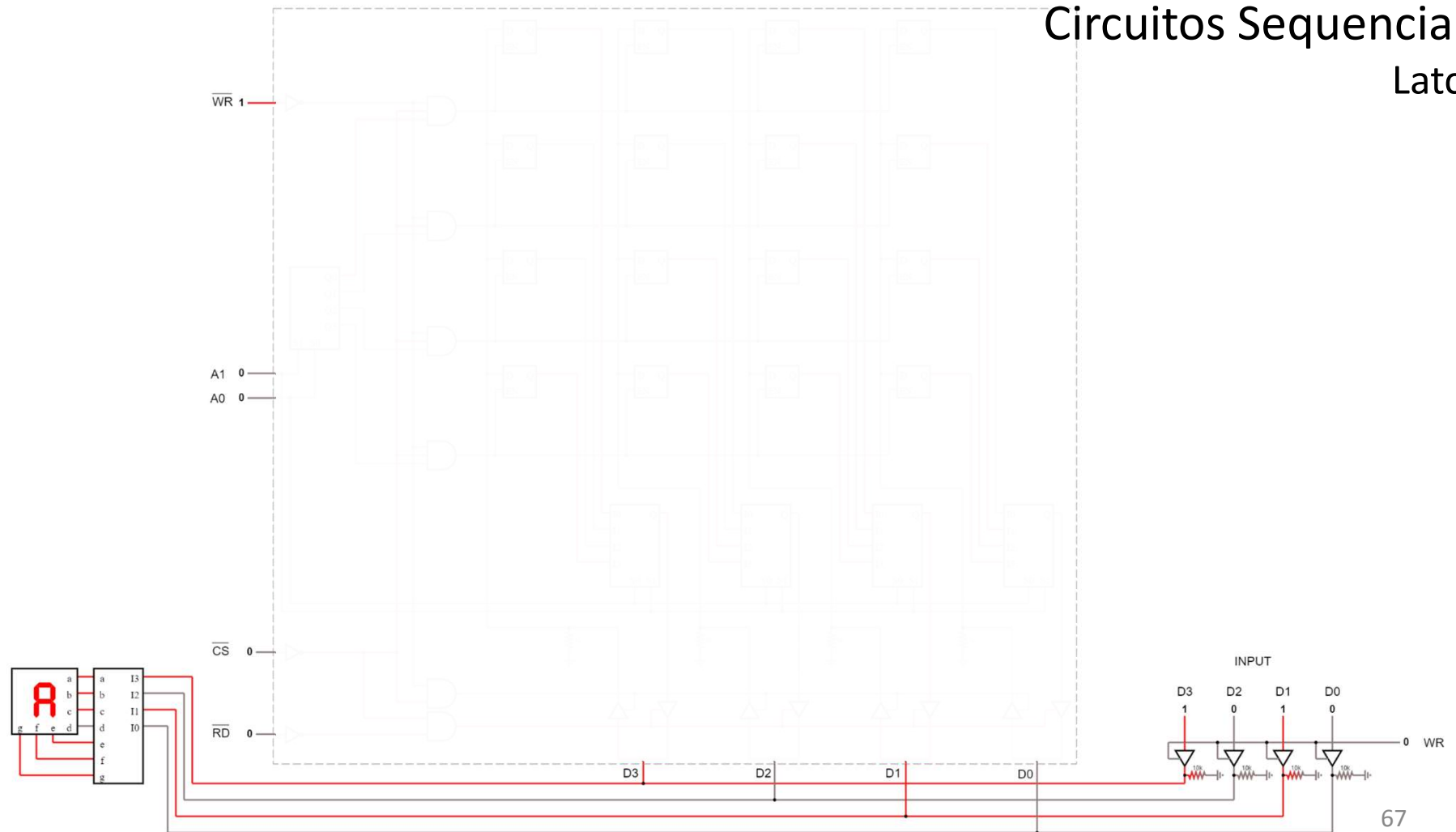
v. Implementar uma Memória  $4 \times 4 \text{ bits}$  utilizando *D Latch*:

- ✓ A memória deverá fazer interface com um barramento;
- ✓ A entrada e saída de dados é feita através do barramento;
- ✓ Uma entrada  $\overline{RD}/WR$  (integrada ou separada) definirá se uma informação será transferida de memória para a saída ou da entrada para a memória;
- ✓ Uma entrada  $\overline{CS}$  serve para habilitar/desabilitar o *Chip*;



# Circuitos Sequenciais

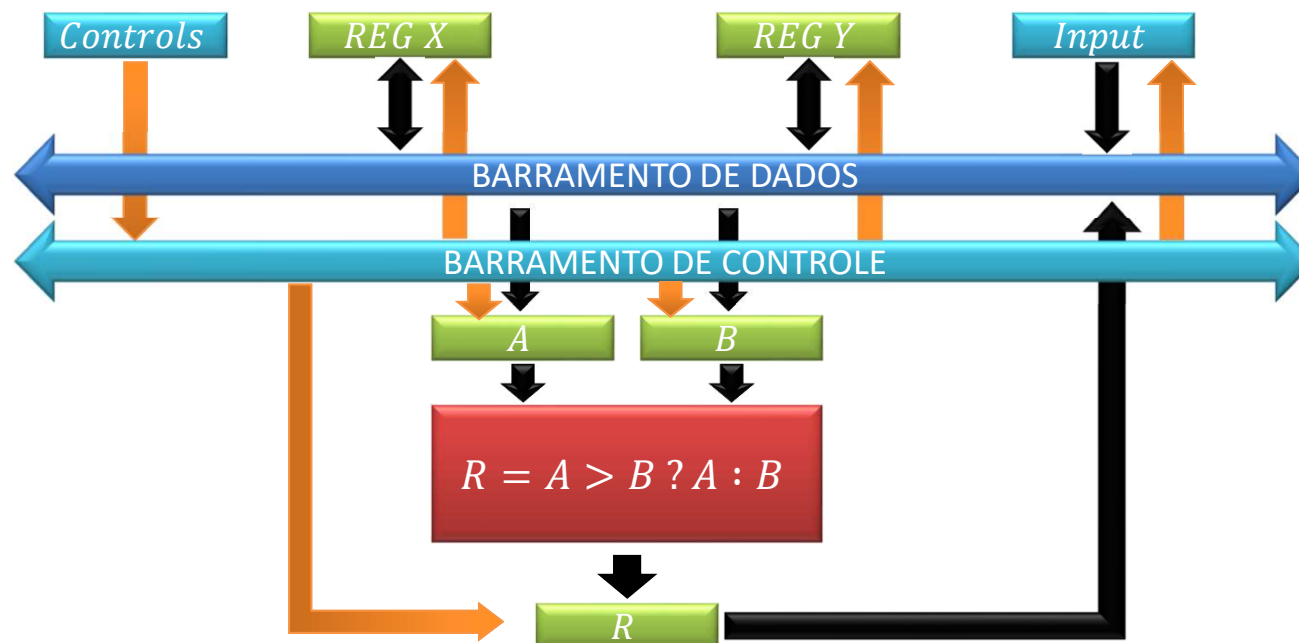
## Latch



## Circuitos Sequenciais

### Latch

- vi. Implementar um circuito de 4 *bits* que fornece o maior entre dois inteiros. Segue um esboço da arquitetura:

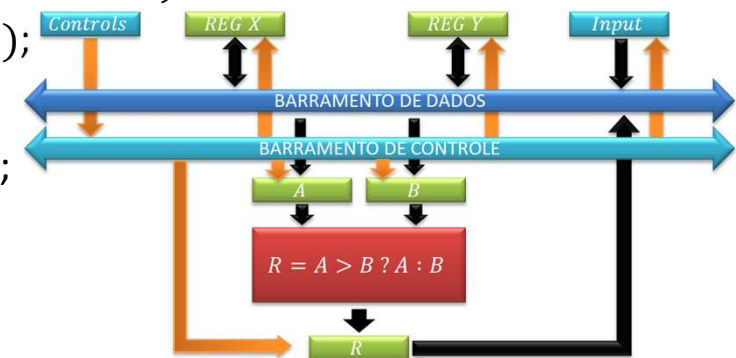


# Circuitos Sequenciais

## Latch

Considerar as seguintes especificações:

- ✓ Os Registradores e Buffers utilizam *Latch D*;
- ✓ A interface entre os componentes é feita através do barramento;
- ✓ A interface *Inputs* aceita um inteiro de 4 *bits*;
- ✓ A interface *Controls* aceita códigos binários que definem as configurações:
  - Selecionar *SOURCE* (*INPUT* | *R* | *X* | *Y* = 00, 01, 10, 11);
  - Selecionar *DEST* (*A* | *B* | *X* | *Y* = 00, 01, 10, 11);
  - $OP = \overline{DISABLE}/ENABLE = PULSE$ ;
- ✓ Uma lógica de controle fará a coordenação dos sinais;



## Circuitos Sequenciais

### Latch

- Selecionar *SOURCE* ( $INPUT \mid R \mid X \mid Y = 00, 01, 10, 11$ );
- Selecionar *DEST* ( $A \mid B \mid X \mid Y = 00, 01, 10, 11$ );
- $OP = \overline{(DISABLE)} / ENABLE = PULSE$ ;

$X = X > Y ? X : Y$	CONTROLES
$X = Input$	$INPUT = X_0; SRC = 00; DST = 10; OP = ENABLE-DISABLE;$
$Y = Input$	$INPUT = Y_0; SRC = 00; DST = 11; OP = ENABLE-DISABLE;$
$A = X$	$SRC = 10; DST = 00; OP = ENABLE-DISABLE;$
$B = Y$	$SRC = 11; DST = 01; OP = ENABLE-DISABLE;$
$X = R$	$SRC = 01; DST = 10; OP = ENABLE-DISABLE;$

- Certificar que o circuito opera corretamente quando  $DST == SRC$ .

# Circuitos Sequenciais

## Latch

