

Circuitos Sequenciais

Prof. Gustavo Girão
girao@imd.ufrn.br

Roteiro

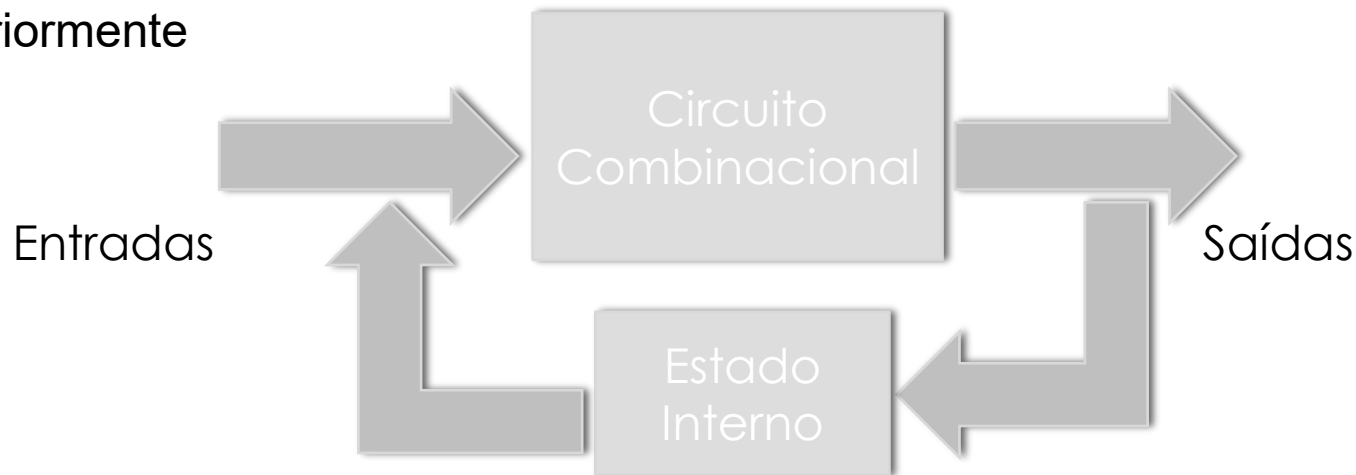
- Apresentação
- Objetivos
- Circuitos Sequenciais
 - Latches
 - Flip-flops
 - Registradores
- Projetando um circuito
- Resumo
- Bibliografia

Objetivos

- Explicar o que são circuitos sequenciais
- Apresentar circuitos sequenciais básicos
 - Latches
 - Flip-flops
 - Registradores
- Apresentar um exemplo de sistema digital composto por circuitos combinacionais e sequenciais

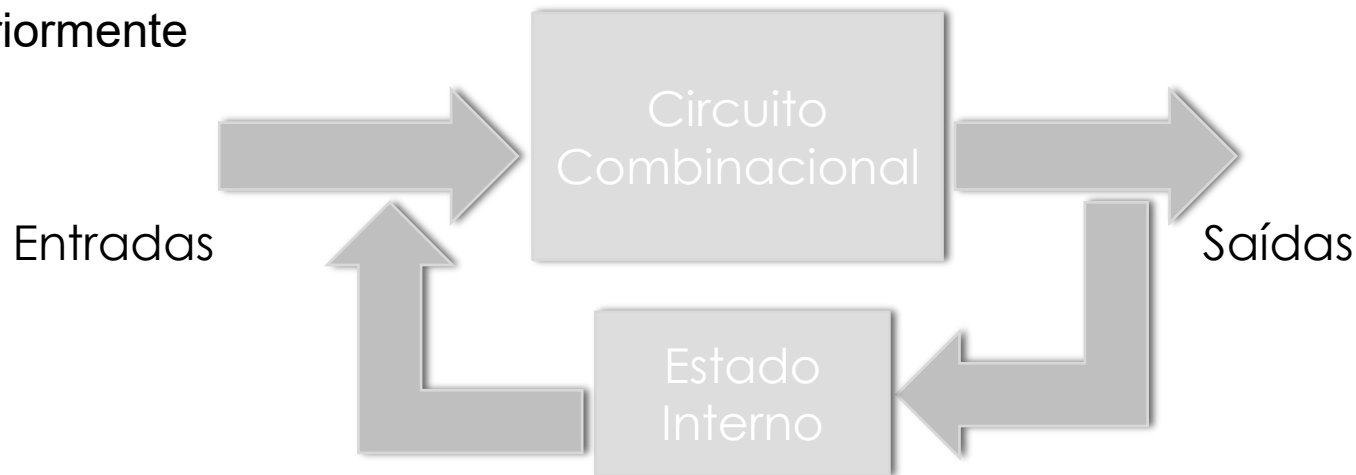
Circuitos Sequenciais

- Um circuito sequencial possui uma realimentação da saída para a entrada, denominada estado interno, cuja principal característica é fazer com que as saídas sejam dependentes das entradas atuais e de estados ocorridos anteriormente



Circuitos Sequenciais

- Um circuito sequencial possui uma realimentação da saída para a entrada, denominada estado interno, cuja principal característica é fazer com que as saídas sejam dependentes das entradas atuais e de estados ocorridos anteriormente



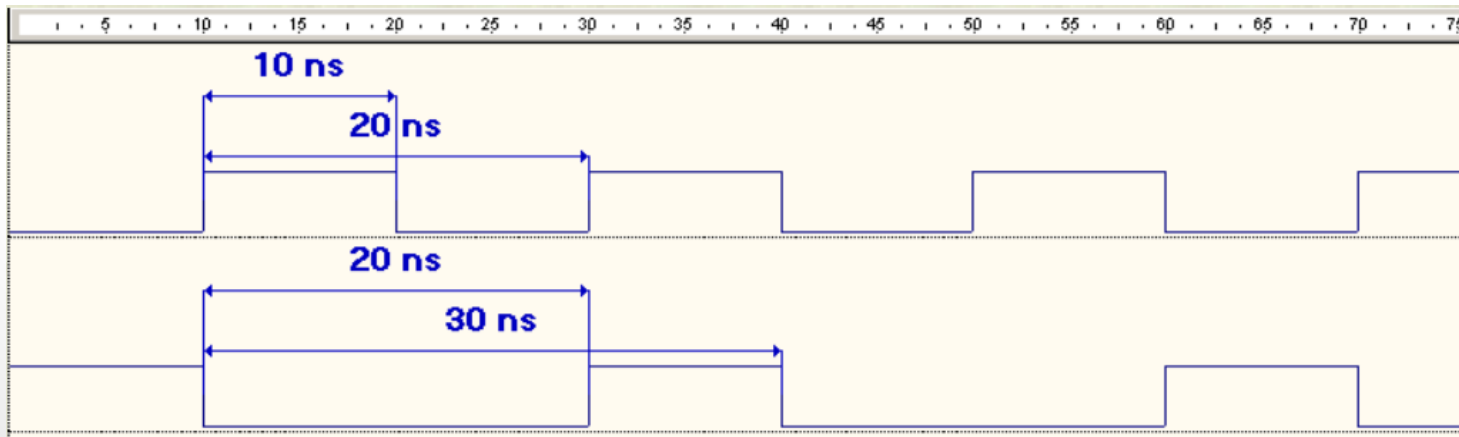
- O estado interno funciona como uma memória que armazena informações de eventos passados exigidos para o funcionamento apropriado do circuito. Os circuitos sequenciais dividem-se em síncronos e assíncronos.
- As funções lógicas, tabelas verdade e Mapas de Karnaugh são utilizadas também no estudo destes circuitos.

Circuitos Sequenciais

- **Assíncronos** - as saídas podem mudar de estado a **qualquer momento** em que uma ou mais entradas também mudarem (projeto e análise de defeito mais difíceis)
- **Síncronos** - as mudanças na saída são determinados por um sinal denominado **clock (ou relógio)**.

Relógio

- Relógio é uma onda periódica com frequência, fase e amplitude. O objetivo deste sinal é determinar os instantes de tempo em que o circuito sequencial deve avaliar as suas entradas
- Exemplo:
 - Abaixo seguem dois relógios CK1 e CK2 com períodos 20ns e 30ns, respectivamente.
 - Consequentemente com frequências 50MHz e 33,33MHz
 - 1 Hz : 1 evento por segundo
 - 1 segundo == 1.000 ms (1×10^3) == 1.000.000 us (1×10^6) == 1.000.000.000 ns (1×10^9)
 - $1.000.000.000 \text{ ns} / 20 \text{ ns} = 50.000.000 = 50\text{M}$ eventos por segundo (MHz)
 - $1.000.000.000 \text{ ns} / 30 \text{ ns} = 33.333.333,333 = 33,33\text{M}$ evento por segundo (MHz)

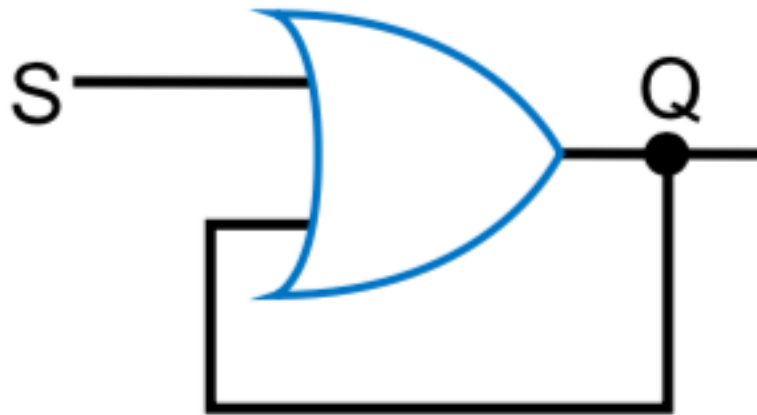


Projeto de um Circuito Sequencial

- Temos que ter em mente que um circuito sequencial tem as suas saídas (*output*) dependentes tanto do estado atual quanto da entrada (*input*)
- Uma vez que este estado é alterado torna-se necessário de alguma forma armazenar essa informação até que futuramente novas combinações nas entradas voltem a ocorrer e mudem este estado
- Portanto, para esse armazenamento são utilizadas células de memória, no caso latches ou flip-flops
- Latches e flip-flops são a base para a criação de registradores, que são os elementos de memória de um sistema de computação

Projeto de um Circuito Sequencial

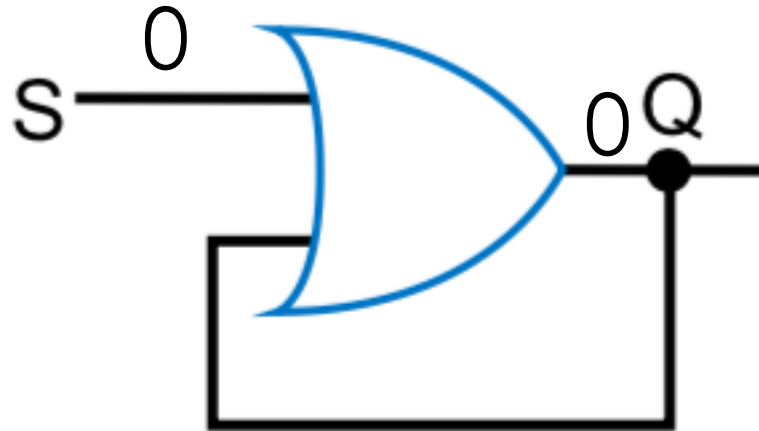
- Para criar circuitos sequenciais, é necessário combinar portas lógicas e realimentação do circuito.
- Exemplo de circuito para armazenamento de um bit



Projeto de um Circuito Sequencial

- Analisando o circuito

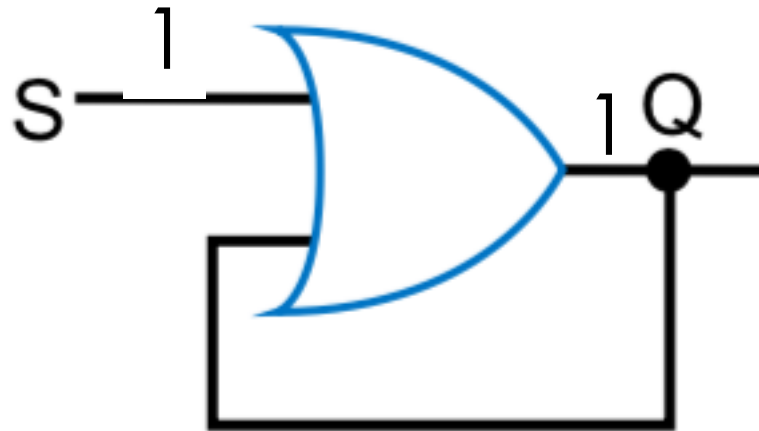
Entrada 0, o circuito deve se manter em 0



Projeto de um Circuito Sequencial

- Analisando o circuito

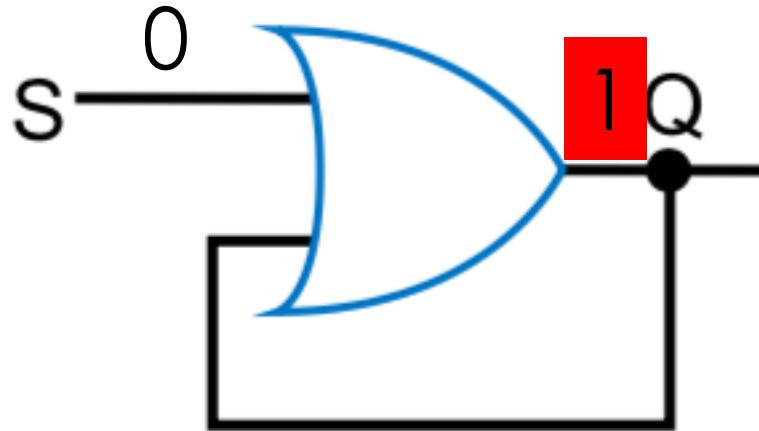
Entrada 1, o circuito deve se manter em 1



Projeto de um Circuito Sequencial

- Analisando o circuito

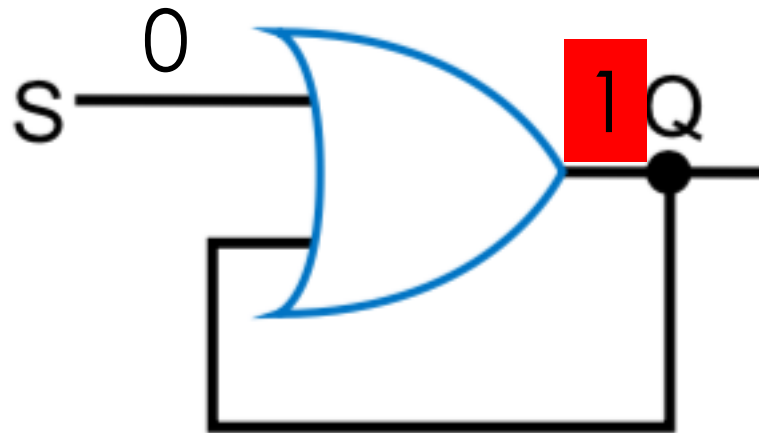
Entrada 0, o circuito deve se manter em 0



Projeto de um Circuito Sequencial

- Analisando o circuito

Entrada 0, o circuito deve se manter em 0



Após S ser configurado para 1, o circuito irá permanecer em 1, mesmo quando S voltar para 0

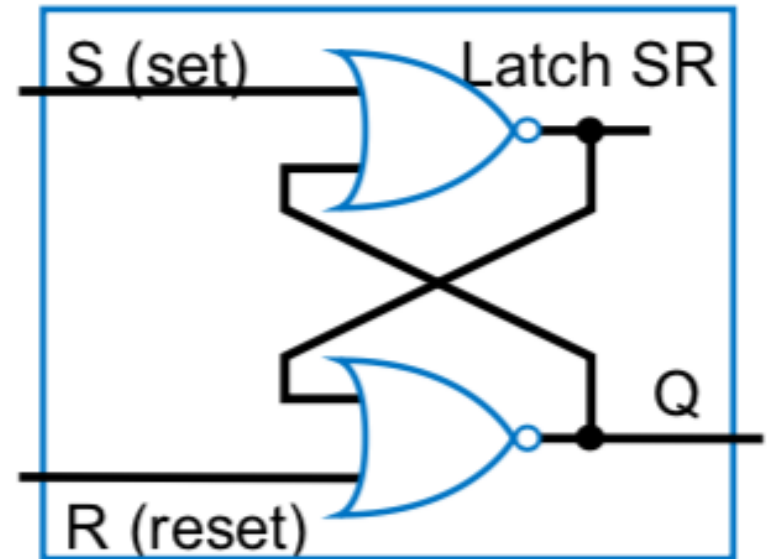
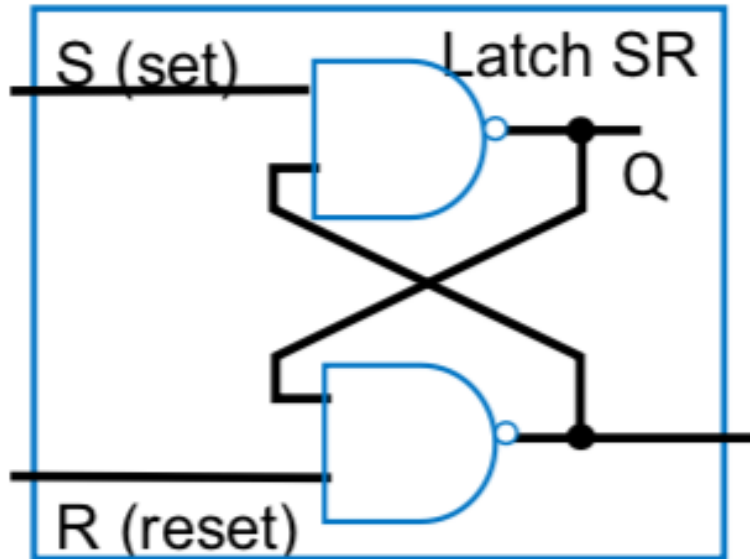
Projeto de um Circuito Sequencial

Latch SR

- O maior problema do circuito apresentado é que uma vez a entrada S muda de estado para 1 a saída Q permanecerá sempre em 1
- Isto ocorre devido à realimentação
- Uma solução para resolver o problema descrito anteriormente foi o desenvolvimento do **Latch SR**: uma combinação entre duas portas NOR (ou NAND) inter-alimentadas
- De forma simples podemos pensar que um latch SR funciona da seguinte forma: a porta S (SET) obriga o latch a ser um, enquanto a porta R (RESET) obriga o latch a ser zero.

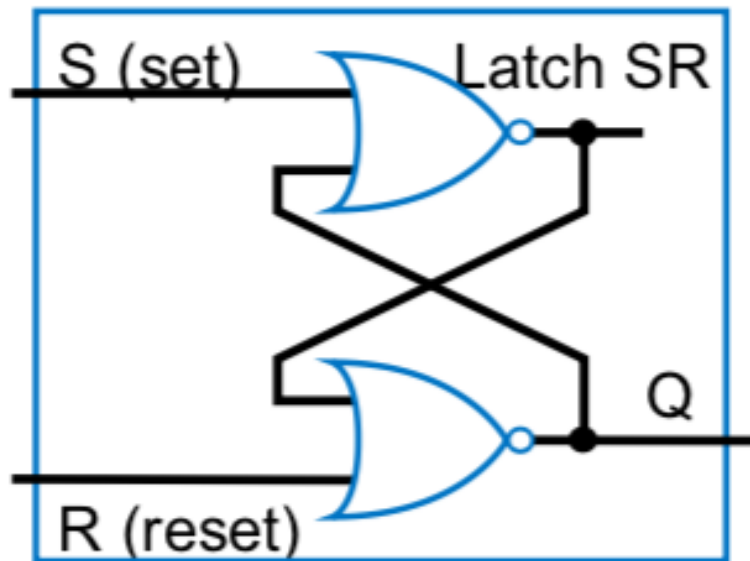
Projeto de um Circuito Sequencial

Latch SR

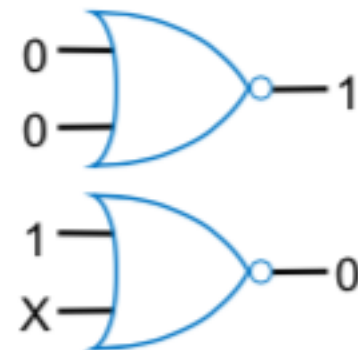


Projeto de um Circuito Sequencial

Latch SR

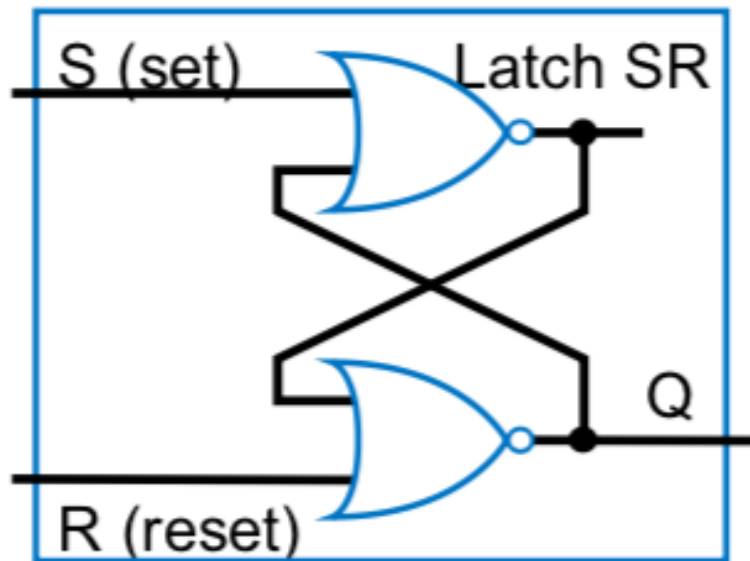


Lembrando...

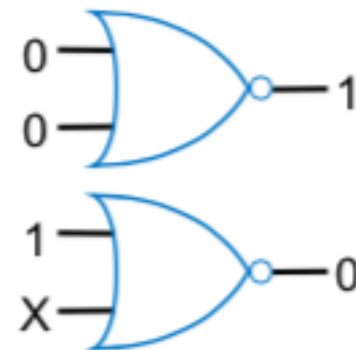


Projeto de um Circuito Sequencial

Latch SR



Lembrando...



Quando ambas as entradas S e R ficam em 1 ao mesmo tempo pode ocorrer um problema de indefinição do estado, ou seja, a saída Q do latch SR vai começar a oscilar quando S e R ambas voltarem a ser zero. Isso para um circuito digital não é aceitável.

Projeto de um Circuito Sequencial

Latch SR

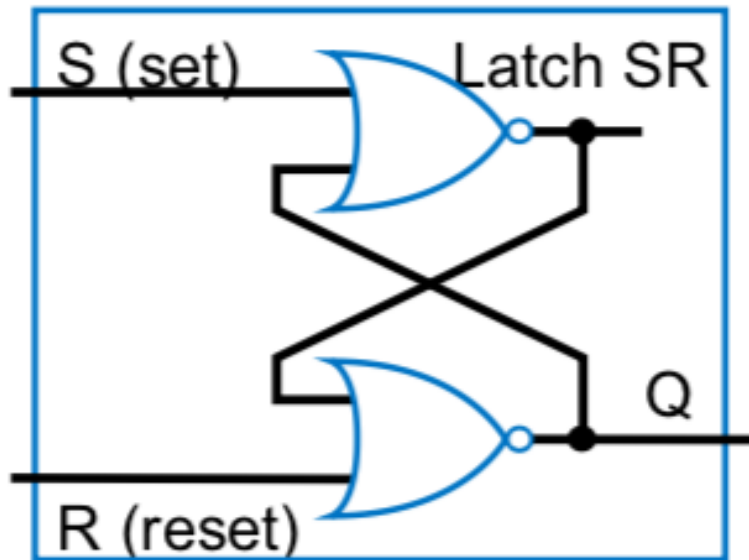
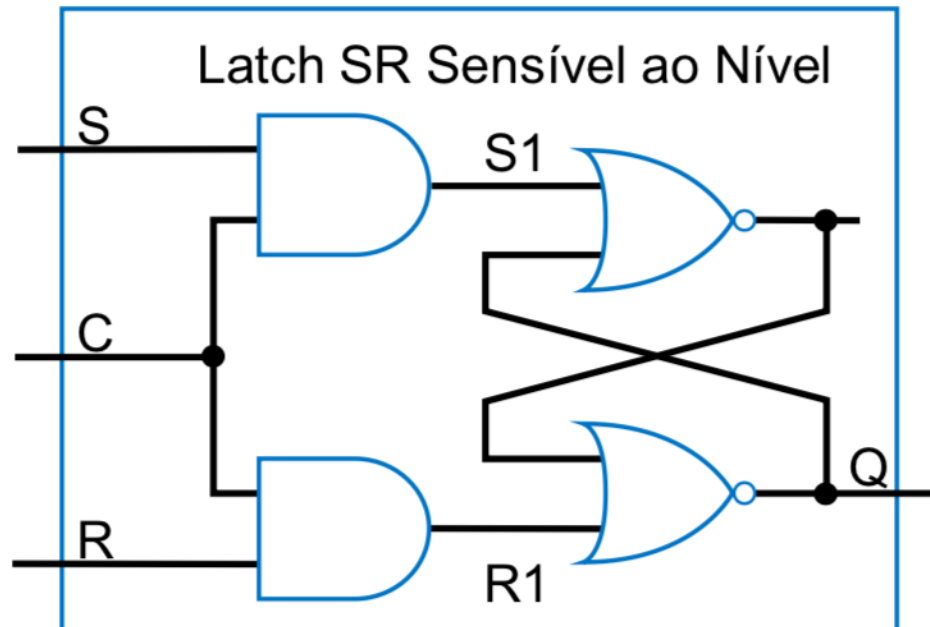


Tabela Verdade		
S	R	Q
0	0	Mantém o estado anterior
0	1	0
1	0	1
1	1	Inválido

Projeto de um Circuito Sequencial

Latch SR sensível ao Nível

- A solução é incluir um controle que só permite a entrada de dados diferentes de 1 quando as entradas estão corretas e estáveis.
- Assim, nunca serão habilitadas as entradas $S=1$ e $R=1$



Projeto de um Circuito Sequencial Latch D

- A diferença entre o SR e o D é que no D, a entrada R é o inverso de S.
- Assim, garante-se que S e R nunca serão iguais

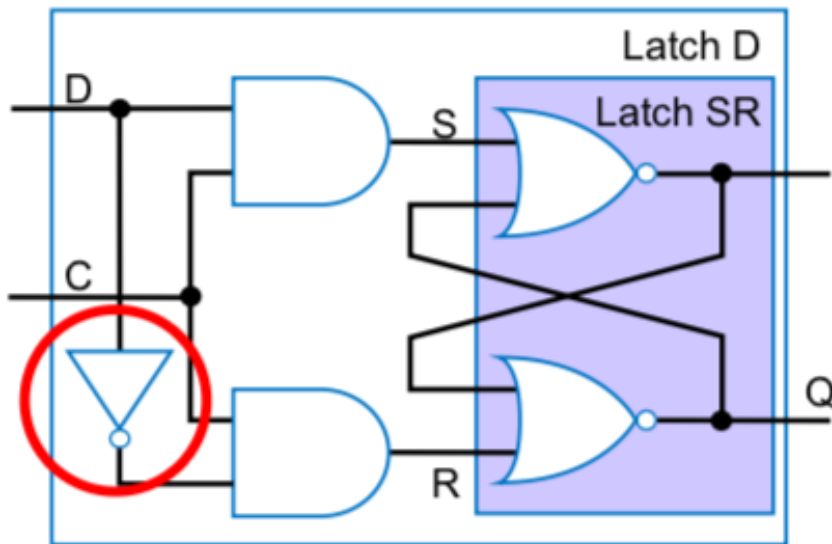
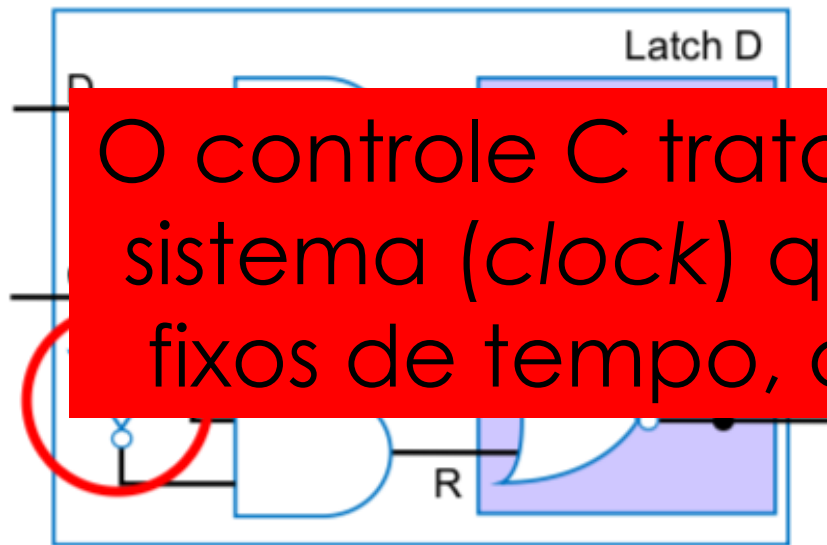


Tabela Verdade		
C	D	Q
0	X	Mantém o estado anterior
1	0	0
1	1	1

Projeto de um Circuito Sequencial Latch D

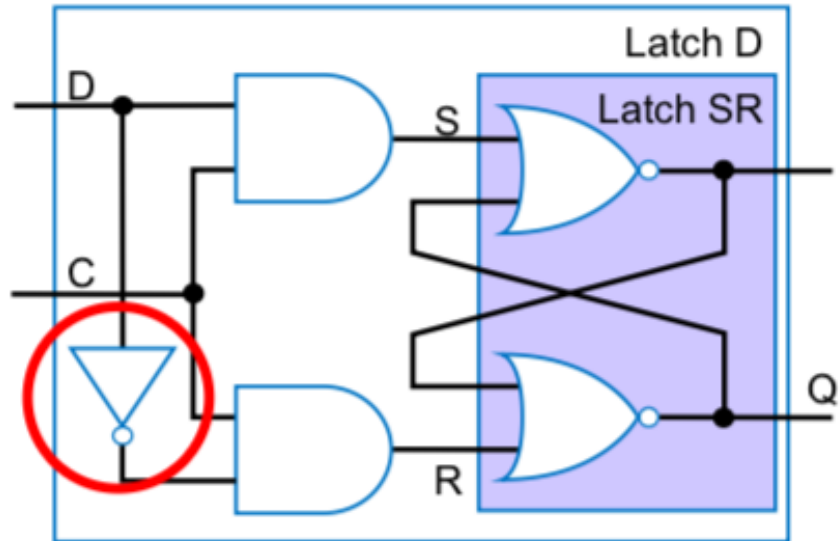
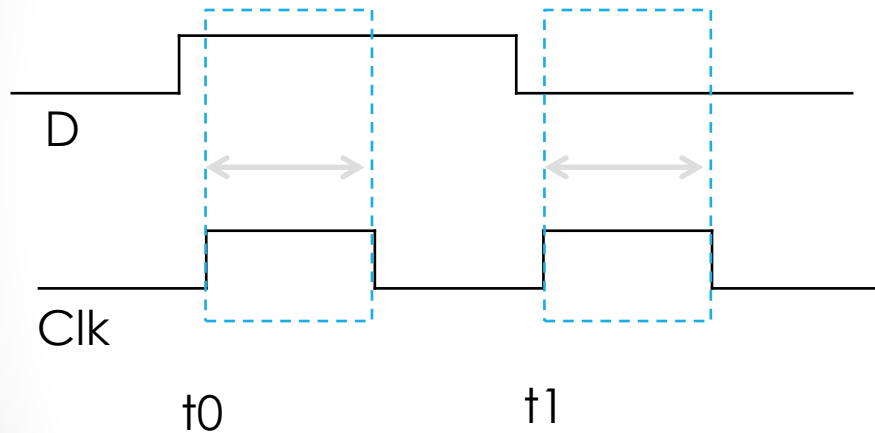
- A diferença entre o SR e o D é que no D, a entrada R é o inverso de S.
- Assim garante-se que S e R nunca serão iguais



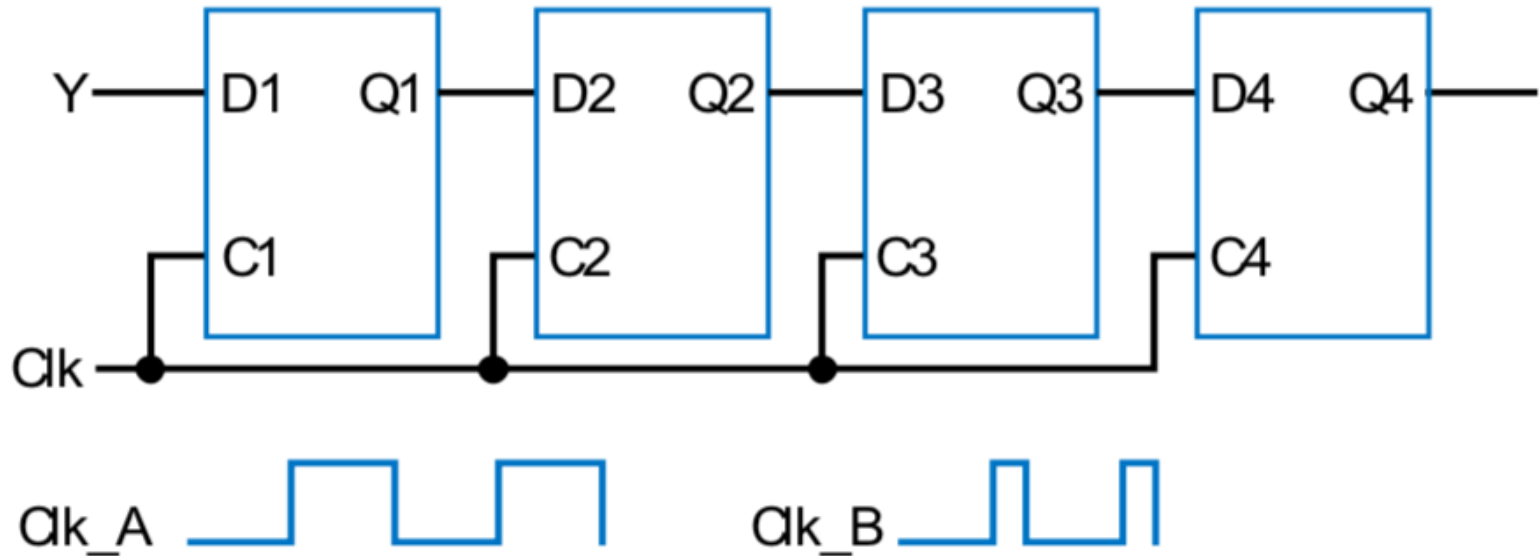
O controle C trata-se do relógio do sistema (*clock*) que, em intervalos fixos de tempo, oscila entre 0 e 1

nterior

Projeto de um Circuito Sequencial Latch D

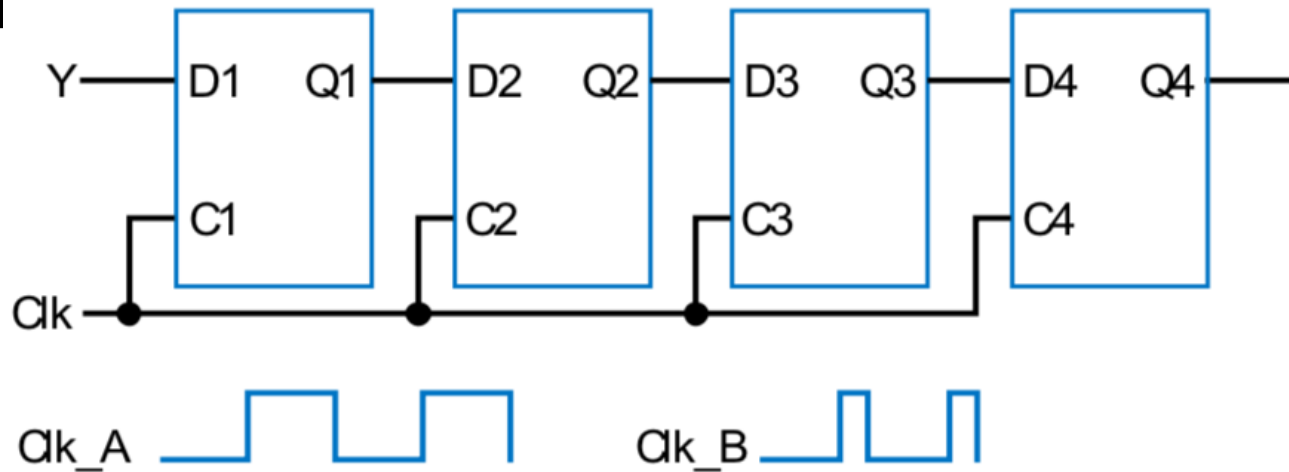


Problema com o Latch D sensível a nível



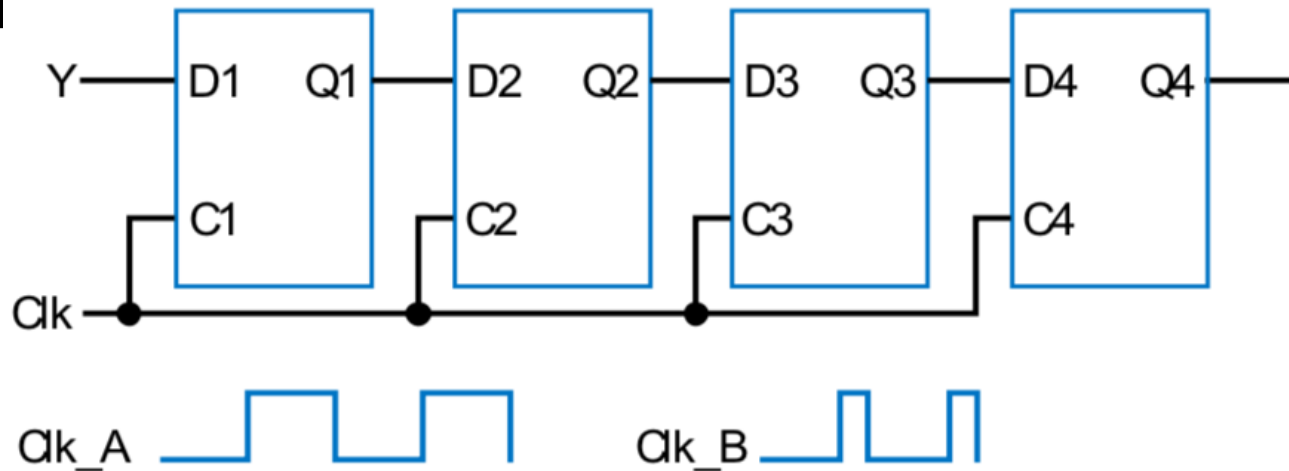
Quando $C=1$, o sinal (Clk) será propagado para quantos latches?

Problema com o Latch D sensível a nível



Quando C=1, o sinal (Clk) será propagado para quantos latches?

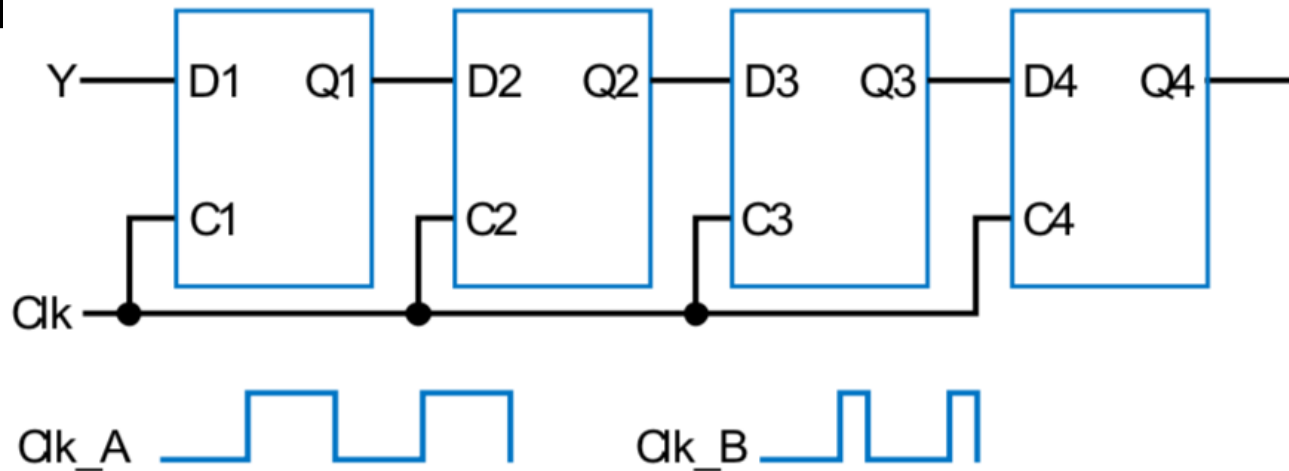
Problema com o Latch D sensível a nível



Quando $C=1$, o sinal (Clk) será propagado para quantos latches?

- O resultado depende de quanto tempo C permanece em 1
- Clk_A – sinal pode ser propagado por múltiplos latches
- Clk_B – sinal pode ser propagado por poucos latches

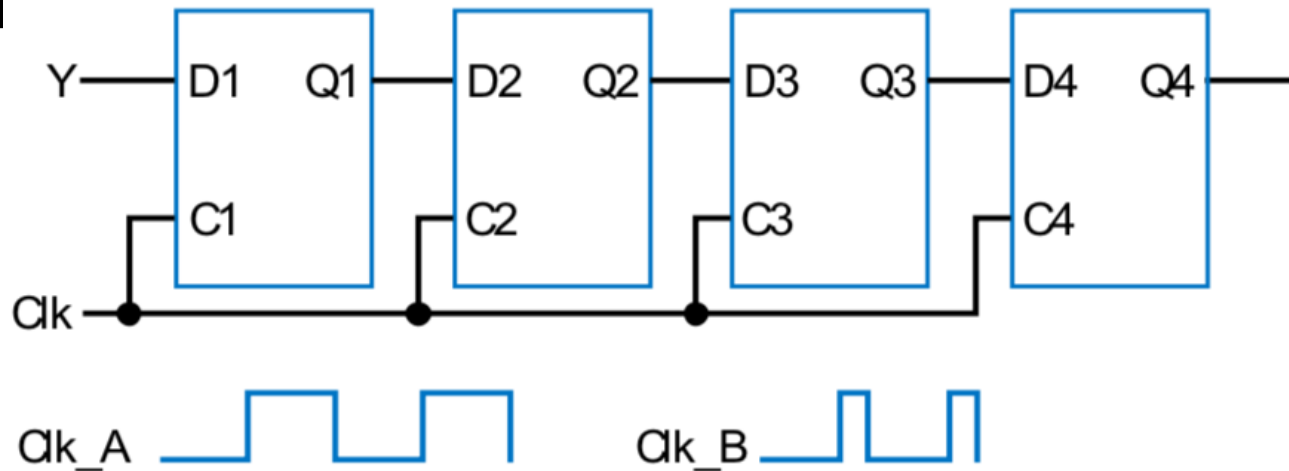
Problema com o Latch D sensível a nível



Quando $C=1$, o sinal (Clk) será propagado para quantos latches?

- O resultado depende de quanto tempo C permanece em 1
- Clk_A – sinal pode ser propagado por múltiplos latches
- Clk_B – sinal pode ser propagado por poucos latches
- Projetar o tempo em que C deverá ficar ativo é complexo!!!

Problema com o Latch D sensível a nível



Solução: utilizar a borda de subida ou descida do clock

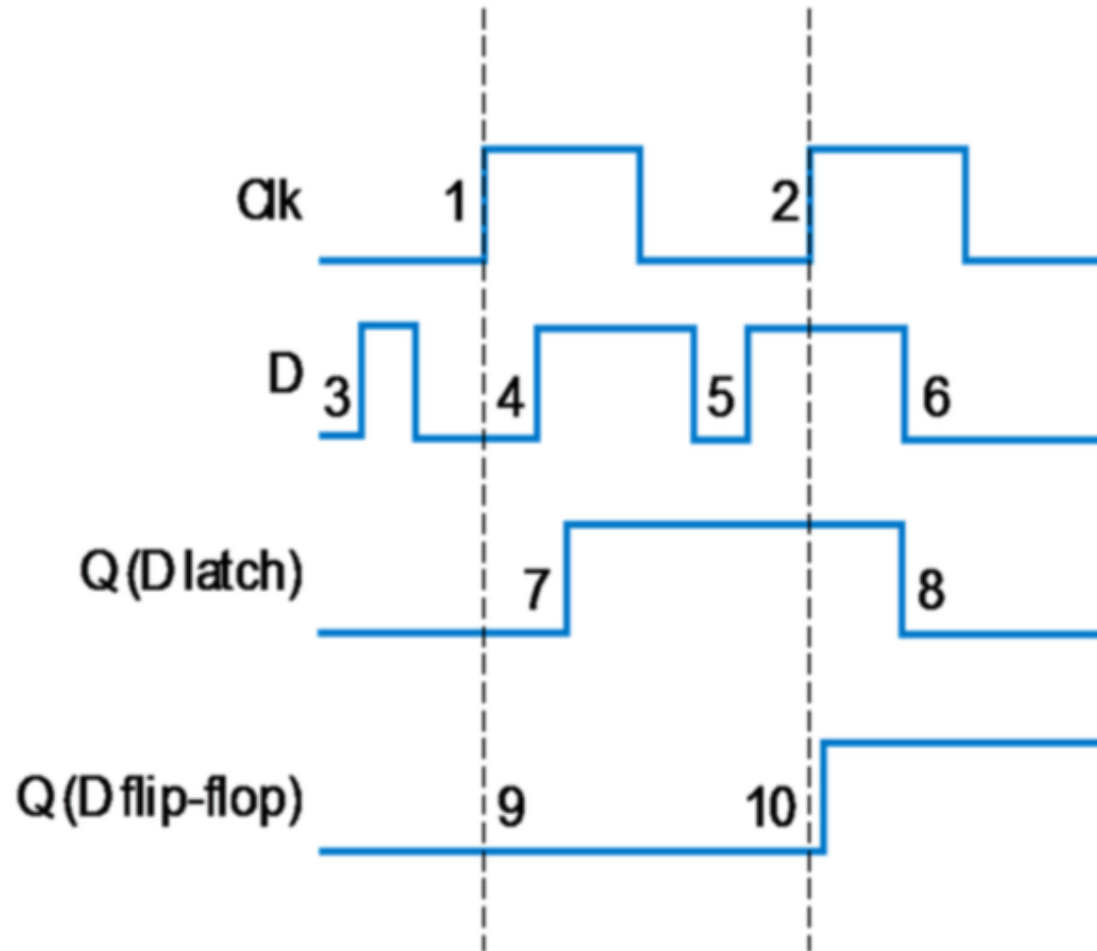
Quando utilizamos esta abordagem, passamos a chamar o latches de.....

- Clk_B – sinal pode ser propagado por poucos latches
- Projetar o tempo em que C deverá ficar ativo é complexo!!!

Flip-Flops

- Os flip-flops são a unidade básica de memória de um sistema de computação
- Formado por latches e sensível a borda do relógio
- Representam a unidade elementar de memória de 1 bit (*binary digit*), ou seja, funcionam como um elemento de memória por armazenar níveis lógicos temporariamente.
- São chamados de biestáveis porque possuem dois estados lógicos estáveis, geralmente representados por “0” e “1”.
- Existem diversos tipos de flip-flops. Estudaremos o flip-flop tipo D (FFD)

Latches x Flip-flops



Flip-Flop D

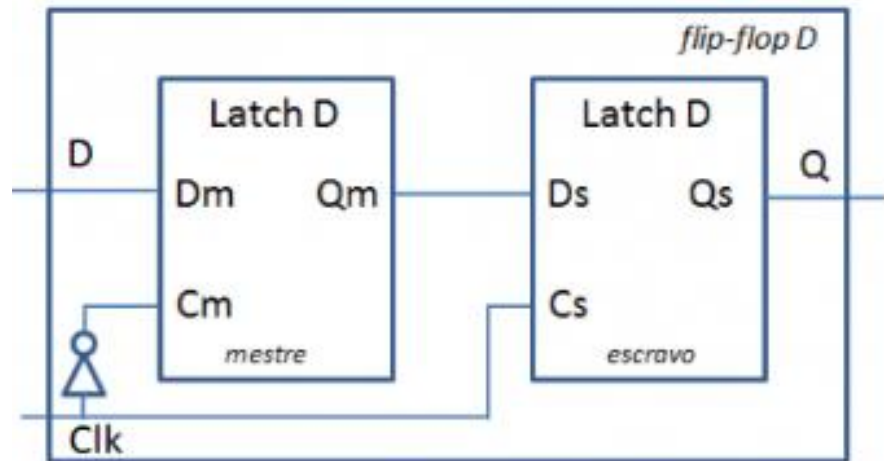
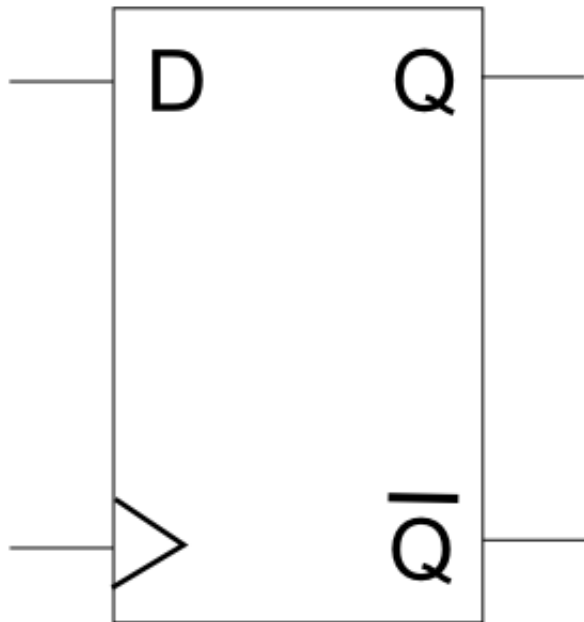
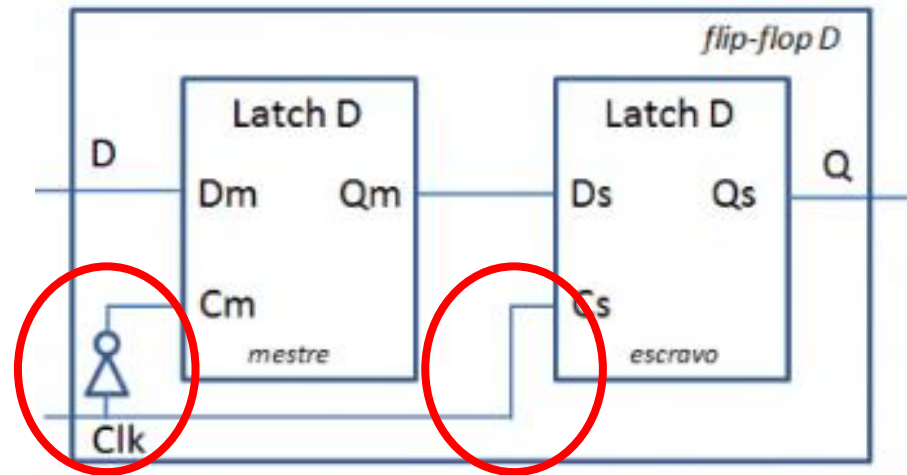
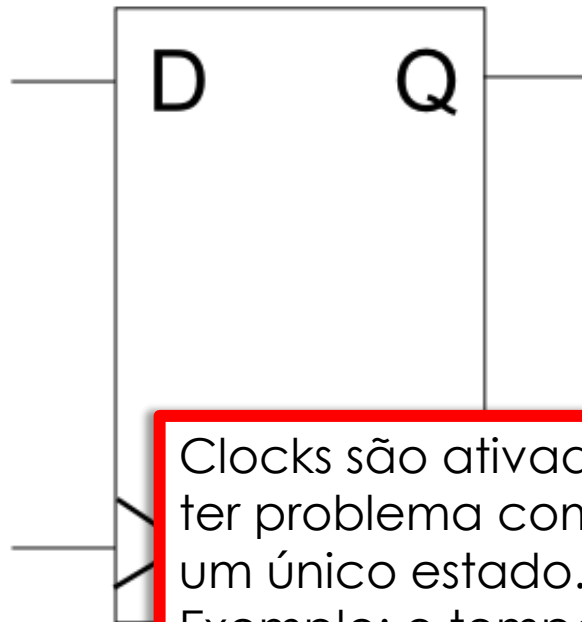


Tabela Verdade

D	CLK	Saída
0	↑ / ↓	0
1	↑ / ↓	1

Flip-Flop D



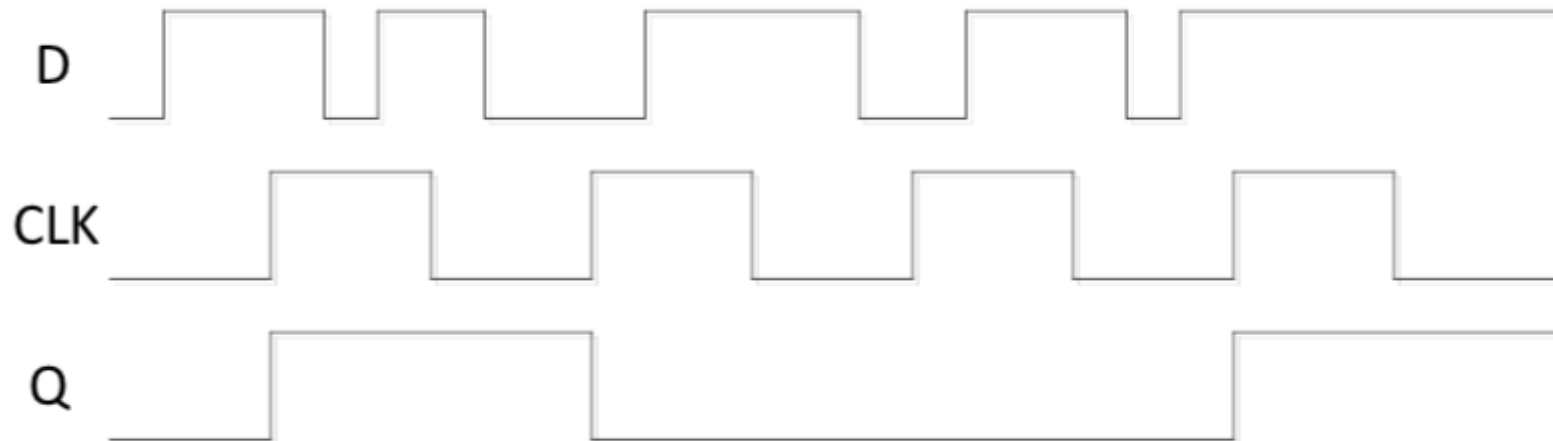
Clocks são ativados alternadamente para não ter problema com o clock por muito tempo em um único estado.

Exemplo: o tempo que demora para o segundo ff receber a entrada no primeiro e habilitar sua saída, o primeiro já pode ter recebido outro valor.

Porém, com bordas invertidas, isso não pode ocorrer.

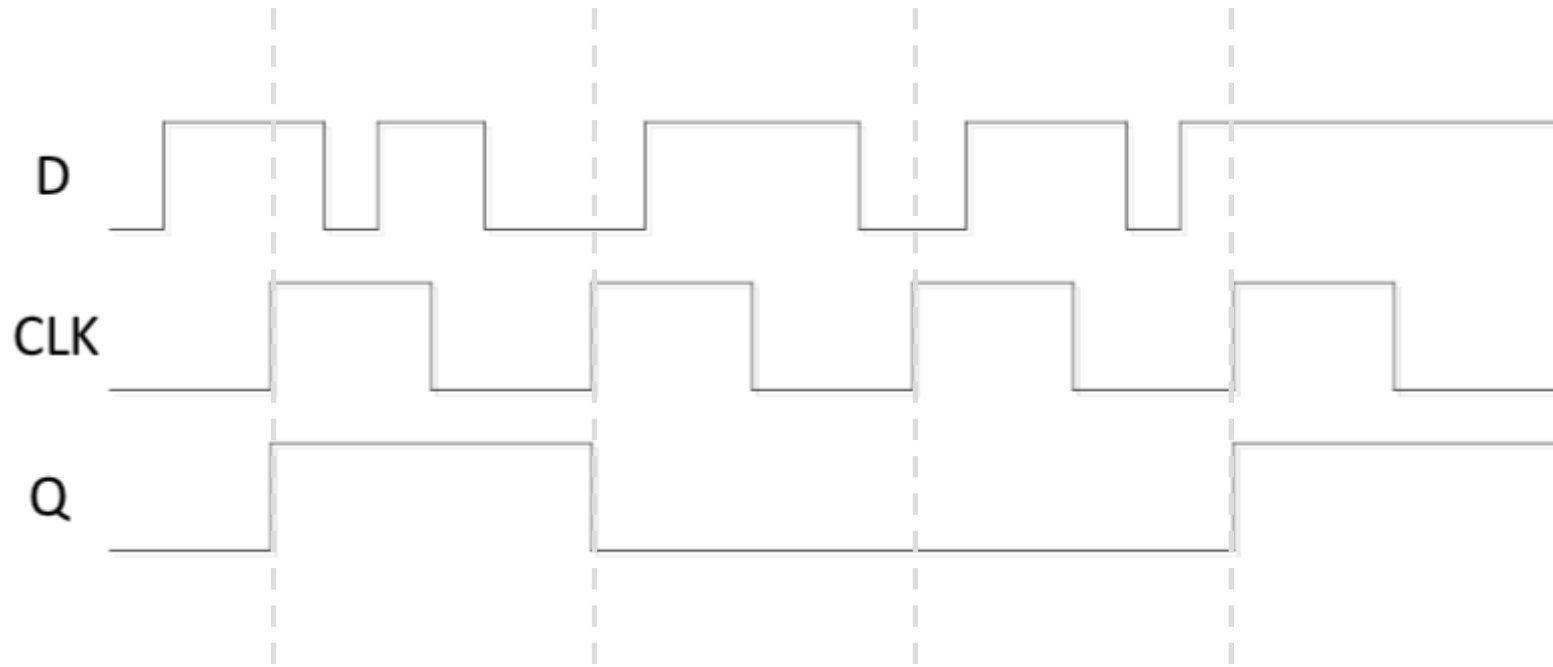
Flip-Flop D

Exemplo



Flip-Flop D

Exemplo (borda de subida)



Outros tipos Flip-Flops

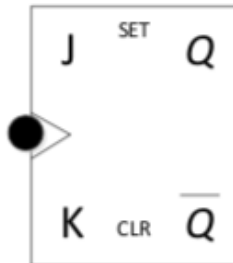
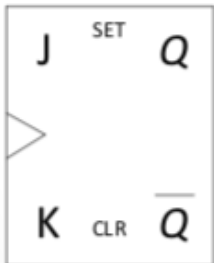
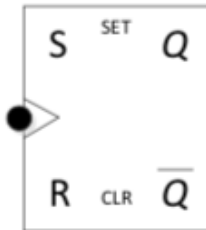
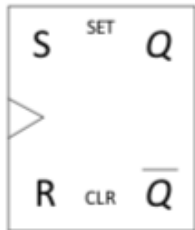


Tabela Verdade

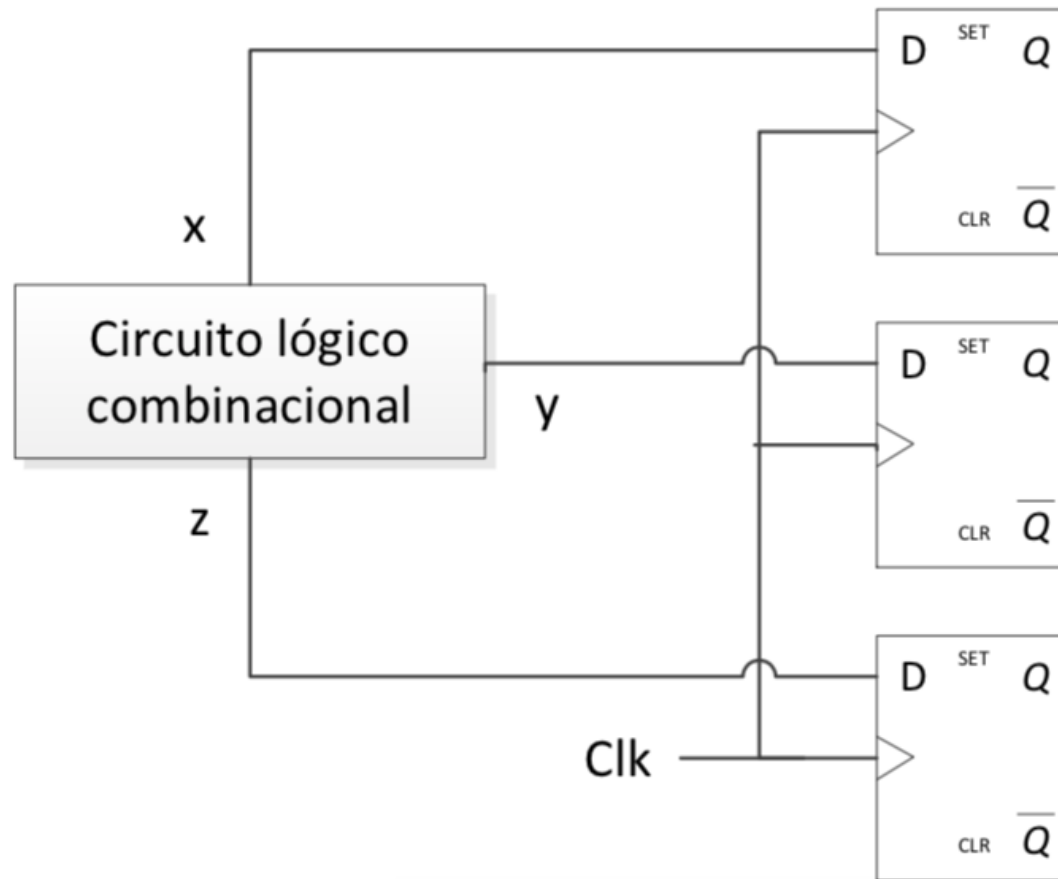
S	R	CLK	Saída
0	0	\uparrow / \downarrow	Q_0 (não muda)
0	1	\uparrow / \downarrow	$Q = 0$
1	0	\uparrow / \downarrow	$Q = 1$
1	1	\uparrow / \downarrow	Inválida

Tabela Verdade

J	K	CLK	Saída
0	0	\uparrow / \downarrow	Q_0 (não muda)
0	1	\uparrow / \downarrow	$Q = 0$
1	0	\uparrow / \downarrow	$Q = 1$
1	1	\uparrow / \downarrow	$\overline{Q_0}$ (comuta)

Flip-Flop D

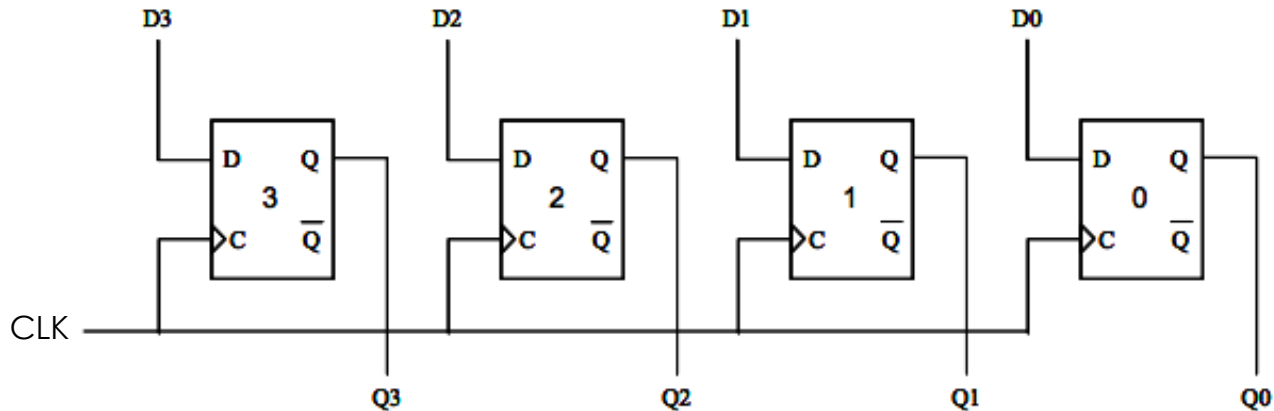
Aplicação – transferência de dados em paralelo



Registrador

- Um registrador é um circuito digital formado por n flip-flops, de modo a poder armazenar simultaneamente (e de maneira independente) n bits.
- Trata-se de um tipo de elemento de armazenamento básico: um processador possui um conjunto de registradores que pode variar até algumas dezenas.

Registrador

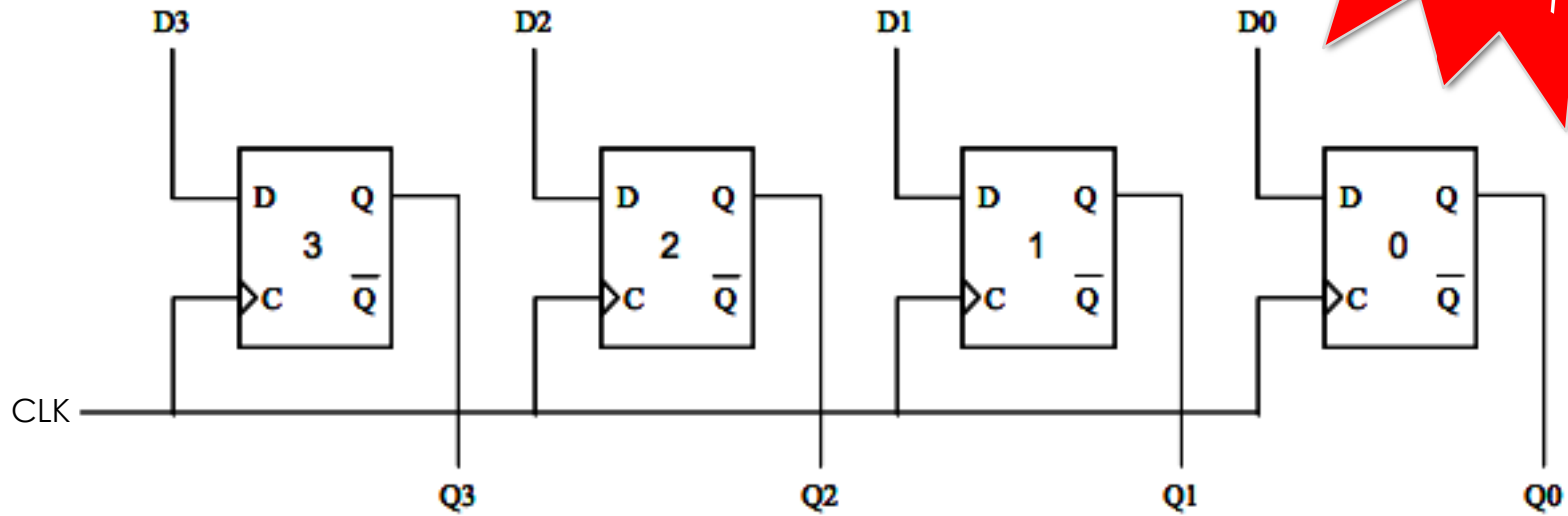


- Note que cada flip-flop é responsável pelo armazenamento de um bit, seguindo a notação posicional, e que cada bit possui um caminho independente dos demais, tanto para entrada como para a saída
- Por isso, o registrador é dito "**de carga paralela**"
- Note também que o flip-flop de índice 0 armazena o bit menos significativo e o flip-flop de índice 3 armazena o bit mais significativo de uma palavra de 4 bits.

Registrador

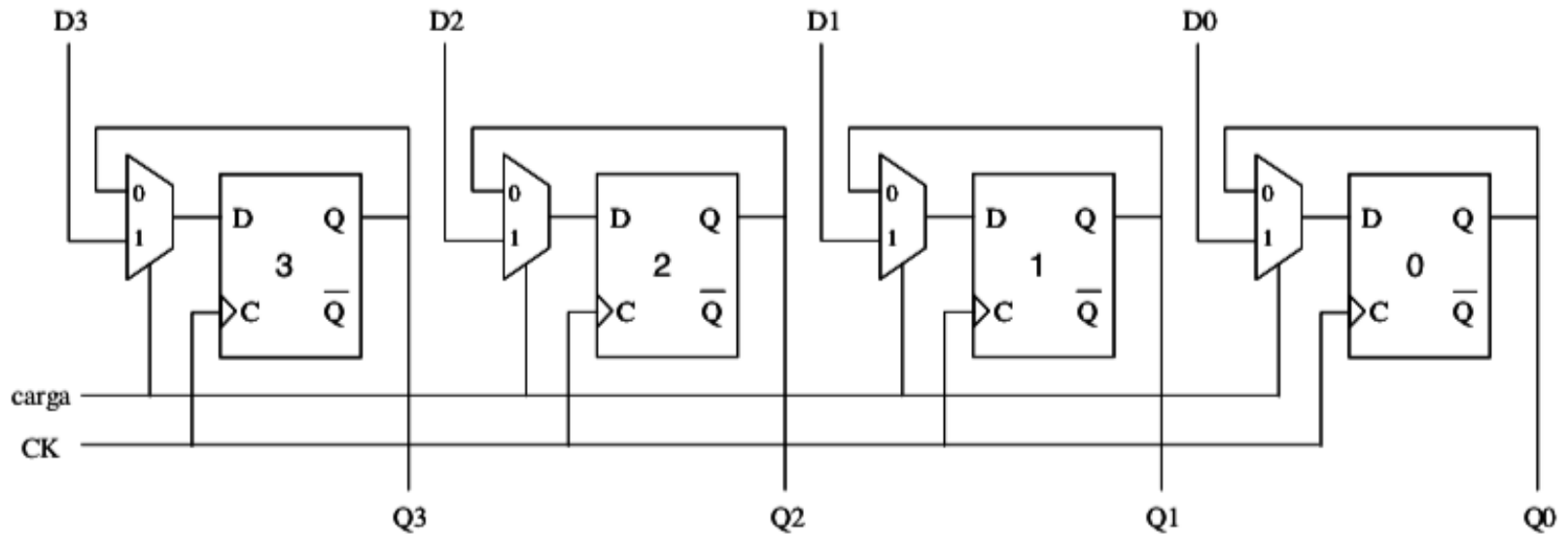
- Um registrador funciona como uma barreira:
 - Os dados disponíveis nas entradas D0, D1, D2 e D3 somente serão copiados quando o sinal de relógio (CLK, no caso) passar por uma borda de subida.
 - Os valores copiados quando da passagem de uma borda de subida permanecerão armazenados pelos flip-flops até a ocorrência da próxima borda de subida.
 - Isto deixa o registrador imune a eventuais mudanças indesejadas dos sinais representados por D0, D1, D2 e D3
 - O valor armazenado num flip-flop qualquer está sempre presente na sua saída Q. Isto quer dizer que o dado armazenado no registrador pode ser consultado por outro recurso de hardware a qualquer tempo
 - O outro recurso pode ser, por exemplo, um somador/subtrator

Registrador



Toda a vez que o sinal de relógio CLK passar por uma borda ascendente, os valores das entradas D0, D1, D2 e D3 serão copiados, mesmo que isso não seja explicitamente desejado.

Registrador



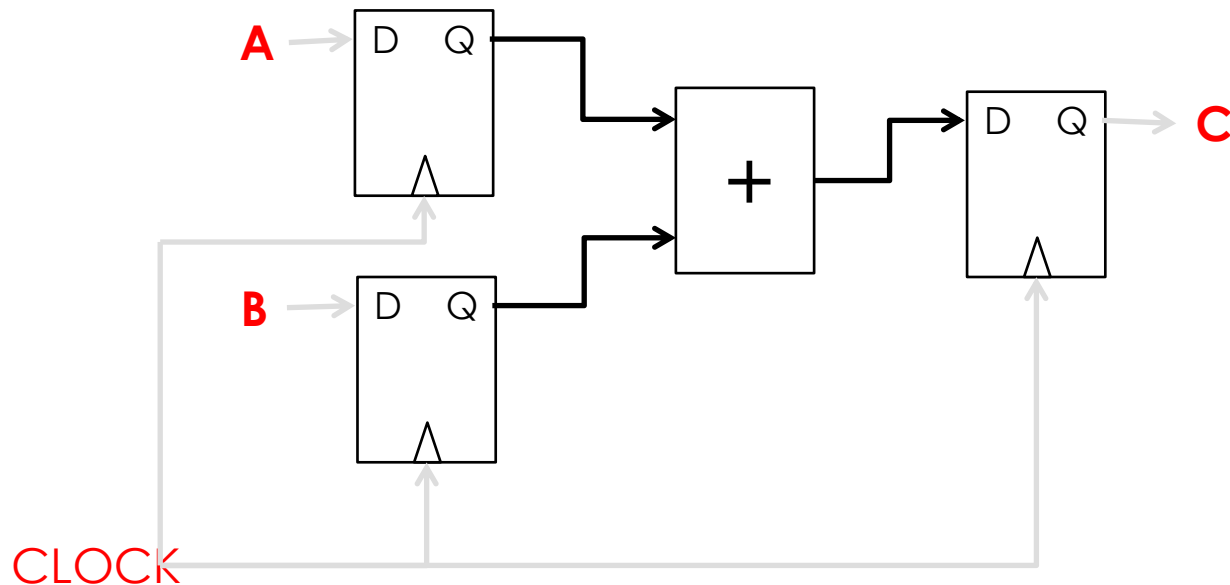
Projetando um Circuito

Projetando um circuito

$$C = A + B$$

Projetando um circuito

$$C = A + B$$



Projetando um circuito

se ($sel = 0$)

$c = a + b$

senão

$c = d + b$

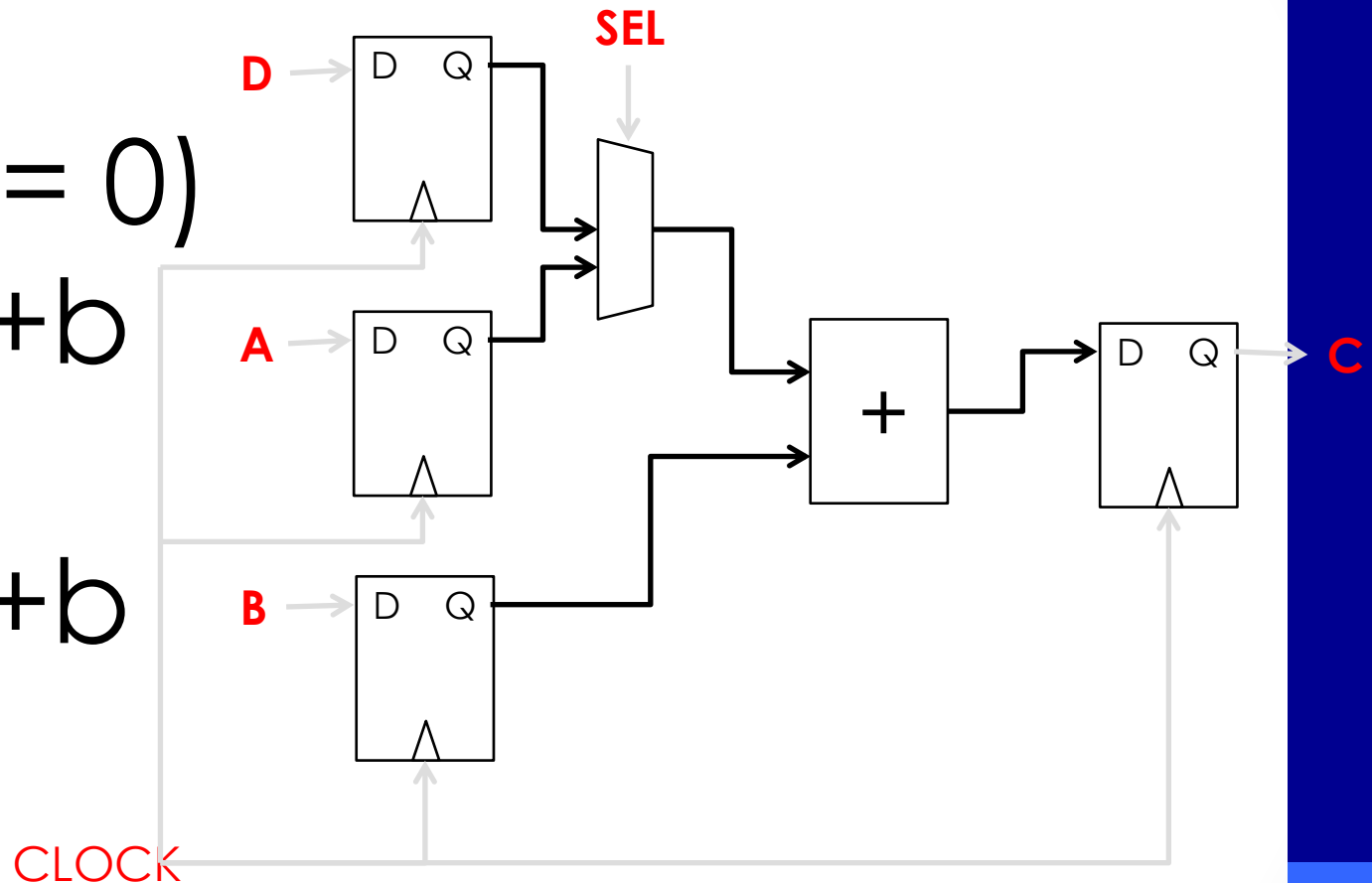
Projetando um circuito

se ($sel = 0$)

$$c = a + b$$

senão

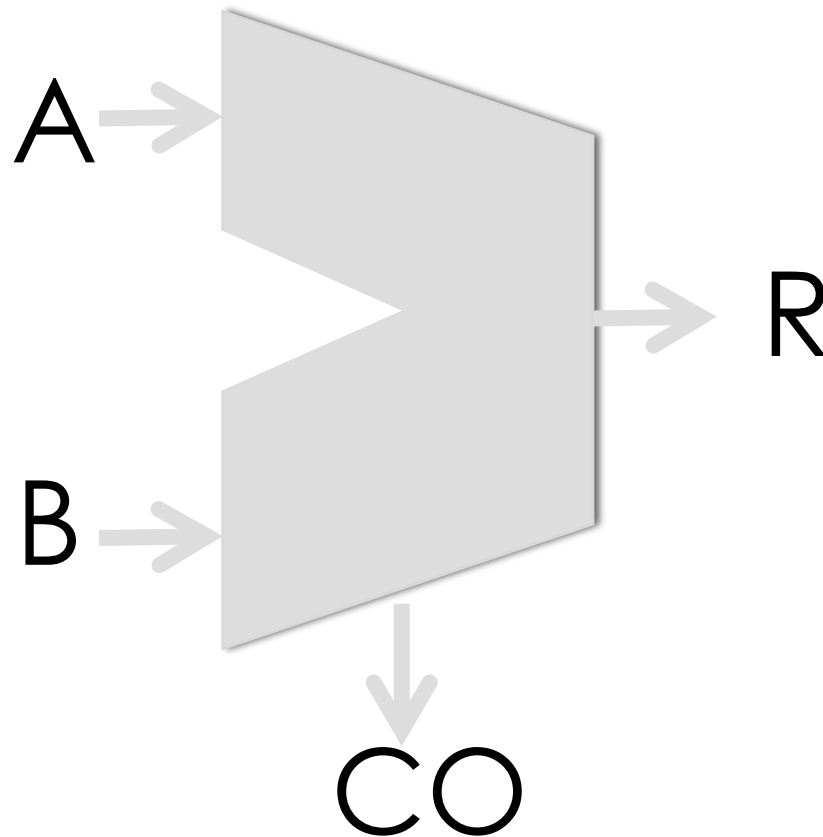
$$c = d + b$$



Projetando um circuito

ULA – Unidade Lógica e Aritmética

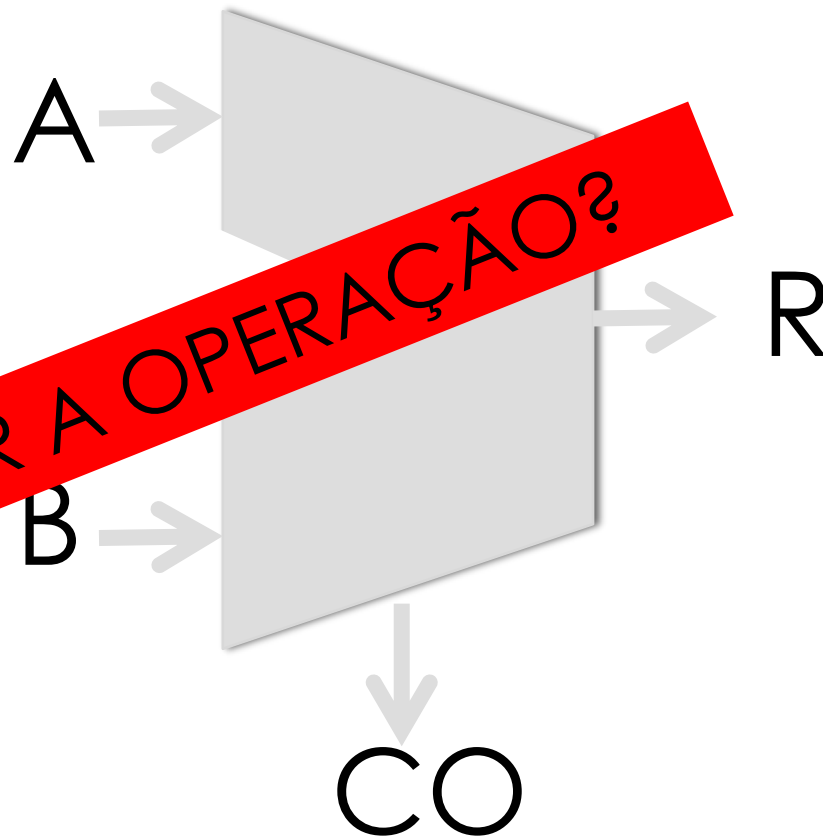
- Soma
- Subtração
- AND
- OR
- NOT
- XOR



Projetando um circuito

ULA – Unidade Lógica e Aritmética

- Soma
- Subtração
- AND
- OR
- NOT
- XOR



COMO ESCOLHER A OPERAÇÃO?

Projetando um circuito

se $(sel = 0)$

$$c = a + b$$

senão

$$c = d - b$$

Resumo

- Diferentemente dos circuitos combinacionais, a saída dos circuitos lógicos sequenciais dependem do estado anterior
- Princípio da realimentação é utilizado para armazenar 1 bit (Flip-Flop)
- Tipos circ. seq.: assíncronos e síncronos
- Latches são os Flip-Flops mais simples
 - Latch SR ($S=R=1$ inválido)
 - Latch D (insere a entrada D na saída)
 - Sensível ao nível
 - Sensível a borda

Resumo

- Flip-Flop são **ativados na borda do relógio**, enquanto latches são **sensíveis ao nível do relógio**
- Tipos de flip-flops
 - SR ($S=R=1$ inválido)
 - JK ($J=K=1$ comuta)
 - D (entrada fica armazenada na saída até a próxima transição)
- Registrador é o elemento básico de memória de um sistema de computação
- Um registrador de n bits é composto por n flip-flops

Referências

- **STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson, 2017. 814 p.**
 - Capítulo 9
- TOCCI, Ronald J; Widmer, Neal S. Sistemas Digitais: princípios e Aplicações. 11. ed. São Paulo SP: Pearson, 2011, 817 p. ISBN 9788576050957
 - Capítulo 5
- PATTERSON, David A; HENNESSY, John L. Organização e projeto de computadores: A interface HARDWARE/SOFTWARE. Rio de Janeiro: Elsevier, 2005, 3ª edição.

Circuitos Sequenciais

Prof. Gustavo Girão
girao@imd.ufrn.br