



FEDERAL UNIVERSITY  
OF SANTA CATARINA

## EEL5105 – Circuitos e Técnicas Digitais

### Aula 6

---

Prof. Héctor Pettenghi

[hector@eel.ufsc.br](mailto:hector@eel.ufsc.br)

<http://hectorpettenghi.paginas.ufsc.br>

## **6. Lógica Sequencial**

---

### **6.1. Sequencial x Combinacional**

### **6.2. Latches**

### **6.3. Sinal de Clock**

### **6.4. Flip-flop Tipo D**

### **6.5. Flip-flop T**

### **6.6. Flip-flop JK**

Nesta aula falaremos sobre lógica sequencial. Qual a diferença da sequencial para a combinacional, tipos de latches, sinal de clock e diferentes tipos de flip-flops.

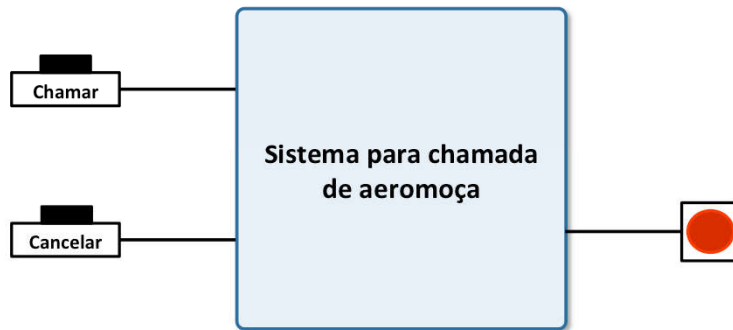
## 6.1. Sequential x Combinacional

---

- 6.2. Latches
- 6.3. Sinal de Clock
- 6.4. Flip-flop Tipo D
- 6.5. Flip-flop T
- 6.6. Flip-flop JK

## 6.1. Sequencial x Combinacional

- Considere que deseja-se projetar o seguinte sistema:

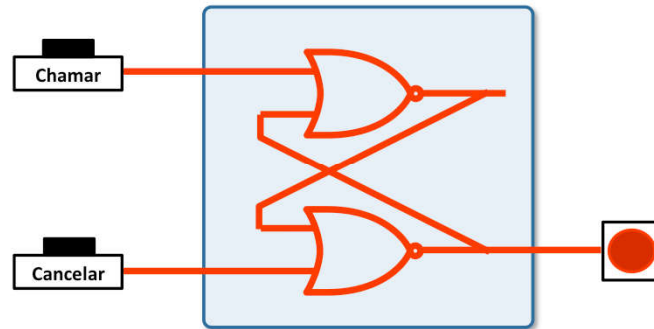


- Um toque no botão “chamar” acende a lâmpada vermelha e um toque no botão “cancelar” apaga a lâmpada.

Para iniciarmos o assunto na diferença entre lógicas sequencial e combinacional, considere que deseja-se projetar o sistema acima. Um toque no botão "chamar" acende a lâmpada vermelha e um toque no botão "cancelar" apaga a lâmpada.

## 6.1. Sequencial x Combinacional

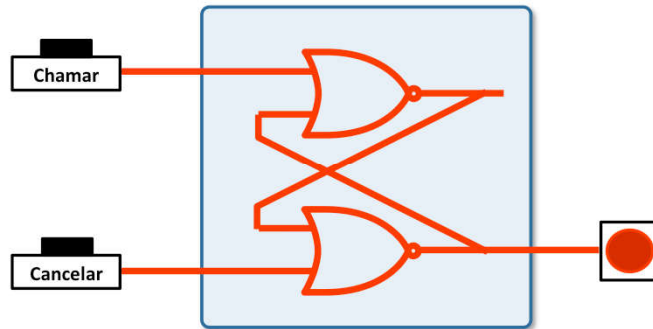
- Uma possível solução:



Aqui, apresentamos uma possível solução.

## 6.1. Sequencial x Combinacional

- Uma possível solução:



- Esse é um circuito que tem **memória**!
  - A saída **não depende somente das entradas atuais**, mas também do que aconteceu no **passado**.

Esse circuito possui memória, pois a saída não depende somente das entradas atuais, mas também do que aconteceu no passado. Temos duas portas NOR. Uma recebendo o estado do botão "chamar" e o estado atual da outra porta NOR, dependente do botão "cancelar". A outra porta, é o contrário. Recebe o estado do botão "cancelar" e o estado atual da outra porta NOR.

Assim, se "chamar"='1', a luz acende, se "cancelar"='1' a luz apaga, e se nenhum dos botões for clicado, a luz permanece acesa ou apagada, de acordo com o último botão que foi clicado.

## 6.1. Sequencial x Combinacional

---

- **Sequencial x Combinacional**
  - Circuito **combinacional**: saída depende apenas dos valores atuais das entradas
  - Circuito **sequencial**: saída depende do que aconteceu com o circuito ao longo do tempo, ou seja, um circuito sequencial tem memória
- Elementos básicos de memória: **Latches** e **Flip-flops**
  - Têm capacidade de armazenar informação
  - Unidade elementar de memória => **1 bit**
  - **Latch** – assíncrono
  - **Flip-flop** – versão síncrona do **latch**

A diferença entre circuito combinacional e sequencial: em um circuito combinacional, a saída depende apenas dos valores atuais das entradas. Já num circuito sequencial, a saída depende do que aconteceu com o circuito ao longo do tempo, ou seja, tem memória. Temos dois elementos básicos de memória: latches and flip-flops. Estes tem capacidade de armazenar informação. Sua unidade elementar de memória é 1 bit. A diferença entre latch e flip-flop é que o flip-flop é síncrono (mudança de estado cadenciada por borda de clock).

6.1. Sequencial x Combinacional

## 6.2. Latches

---

6.3. Sinal de Clock

6.4. Flip-flop Tipo D

6.5. Flip-flop T

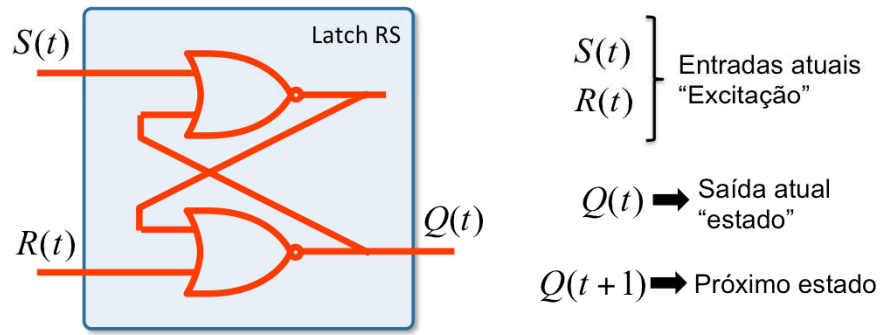
6.6. Flip-flop JK

Agora falaremos sobre Latches.



## 6.2. Latches

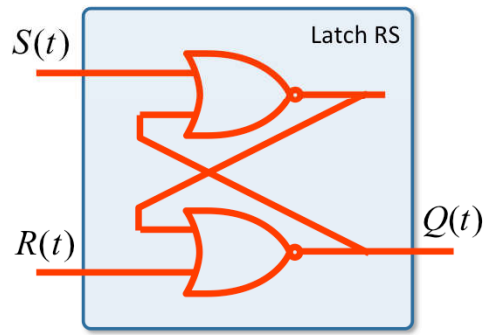
- **Latch RS (Reset-Set) com portas NOR:**



No latch RS, usado no exemplo anterior, temos duas entradas: set e reset. A saída  $Q(t+1)$  depende não só das entradas, mas também da saída atual  $Q(t)$ .

## 6.2. Latches

- **Latch RS (Reset-Set) com portas NOR:**



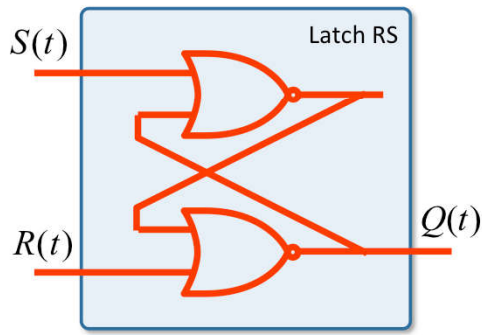
A) Tabela de Transição:

$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A tabela de transição é esta. As entradas  $S(t)$ ,  $R(t)$  e  $Q(t)$  determinam a saída  $Q(t+1)$ .

## 6.2. Latches

- **Latch RS (Reset-Set) com portas NOR:**



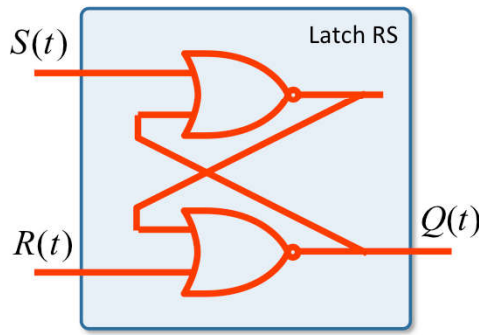
A) Tabela de Transição:

$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$	
0	0	0	0	} Mantém valor armazenado
0	0	1	1	
0	1	0	0	} Iguala a 0 (Reset)
0	1	1	0	
1	0	0	1	} Iguala a 1 (Set)
1	0	1	1	
1	1	0	0	
1	1	1	0	

Fazendo as combinações, temos as seguintes saídas e o que cada uma representa. Observe que quando  $R(t) = 1$ , a saída é igualada a 0. E quando temos  $S(t)=1$ , a saída é igualada a 1. Entretanto, quando  $R(t)=S(t)=0$ , a saída passa a depender somente do estado do valor atual da saída  $Q(t)$ , de forma a manter o valor armazenado.

## 6.2. Latches

- **Latch RS (Reset-Set) com portas NOR:**



A) Tabela de Transição:

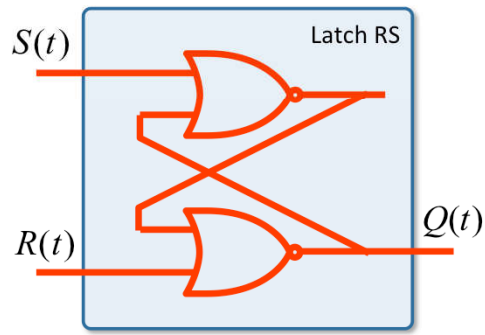
$S(t)$	$R(t)$	$Q(t)$	$Q(t+1)$	
0	0	0	0	} Mantém valor armazenado
0	0	1	1	
0	1	0	0	} Iguala a 0 (Reset)
0	1	1	0	
1	0	0	1	} Iguala a 1 (Set)
1	0	1	1	
1	1	0	0	} Proibido!
1	1	1	0	

Não é possível saber o que exatamente ficará armazenado quando  $S(t)$  e  $R(t)$  se tornarem novamente iguais a 0.

Observe, porém, que não podemos ter uma situação em que ambos valores  $S(t)$  e  $R(t)$  são iguais a 1, pois não é possível saber o que deverá ficar armazenado quando ambos voltarem a 0.

## 6.2. Latches

- Latch RS com portas NOR:



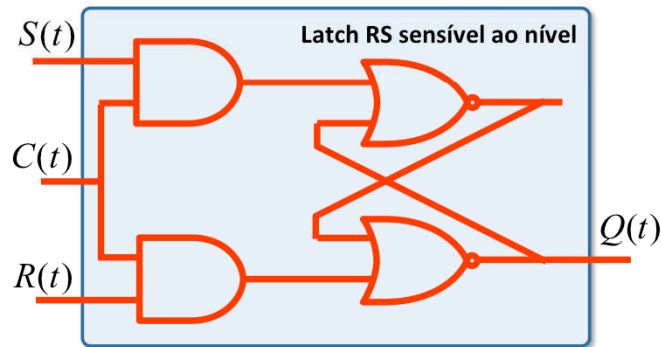
B) Tabela de Função:

$S(t)$	$R(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0 (Reset)
1	0	1 (Set)
1	1	Proibido

Baseado na tabela de transição, temos a tabela de função. Ela simplesmente mostra como o Latch RS se comporta baseado nas entradas.

## 6.2. Latches

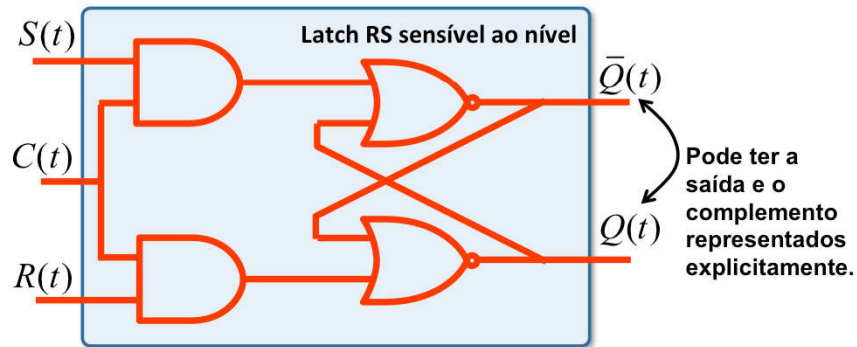
- Outro tipo: **Latch RS sensível ao nível**
  - Latch RS com entrada de habilitação **C** que precisa ser ativada para que ocorra alguma mudança de estado.



Um outro Latch RS existente é o Latch RS sensível ao nível. Este latch depende de uma entrada C que precisa ser ativada para que ocorra alguma mudança de estado.

## 6.2. Latches

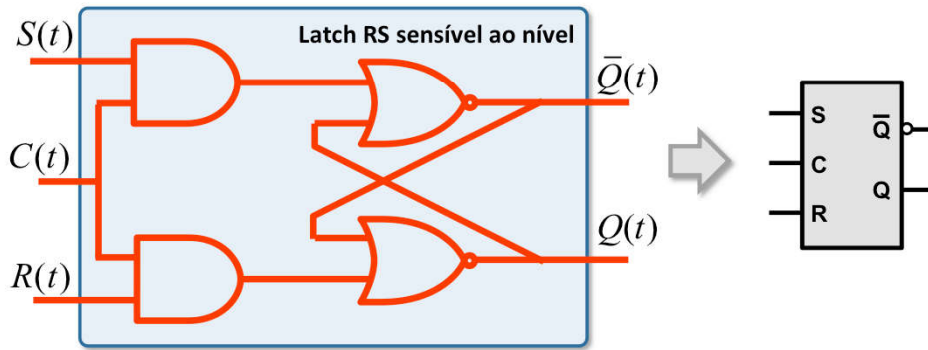
- Outro tipo: **Latch RS sensível ao nível**
  - Latch RS com entrada de habilitação **C** que precisa ser ativada para que ocorra alguma mudança de estado.



Este tipo de latch ainda pode ter a saída e o complemento representados explicitamente.

## 6.2. Latches

- Outro tipo: **Latch RS sensível ao nível**
  - Latch RS com entrada de habilitação **C** que precisa ser ativada para que ocorra alguma mudança de estado.

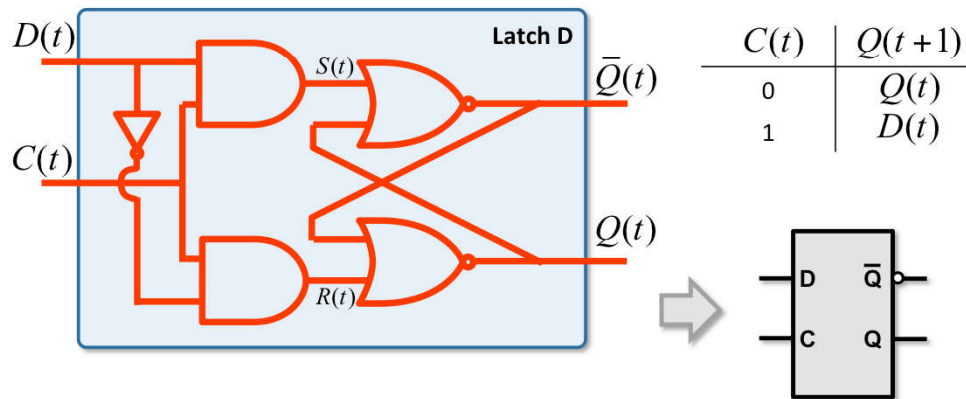


Esta é a representação do latch RS com todos os opcionais. Observe a porta negada representando o complementar da saída  $Q(t)$ .



## 6.2. Latches

- O **Latch D** (Data ou Delay)
  - Mais simples que o **RS** (do ponto de vista de funcionamento) e faz a mesma função (armazenar o valor de um bit)

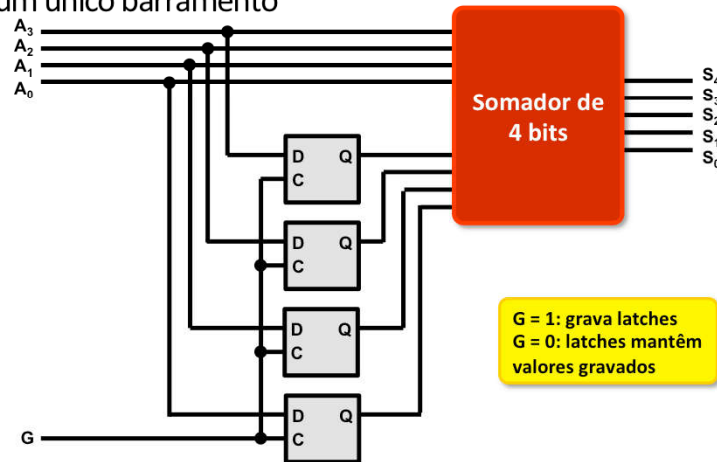


O segundo Latch que veremos é o Latch D. Ele é mais simples que o RS e faz a mesma função. Sempre que o  $C(t)$  estiver alto, a saída será a própria entrada  $D(t)$ . Quando  $C(t)$  estiver baixo, a saída ficará armazenada em forma de 1 bit.

## 6.2. Latches

- **O Latch D**

- Aplicação: somando números disponibilizados a partir de um único barramento



Aqui temos uma possível aplicação somando números disponibilizados a partir de um único barramento. Utilizando 4 latches D, podemos armazenar valores alterando  $G$ , que poderia ser um botão "guardar", por exemplo.

6.1. Sequencial x Combinacional

6.2. Latches

## 6.3. Sinal de Clock

---

6.4. Flip-flop Tipo D

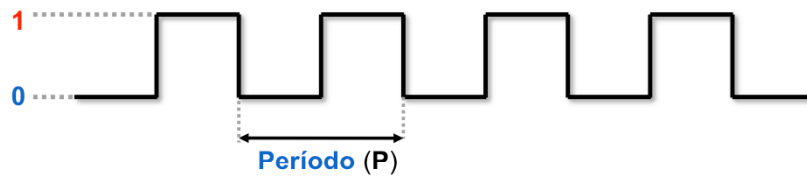
6.5. Flip-flop T

6.6. Flip-flop JK

Agora veremos sobre o sinal de clock.

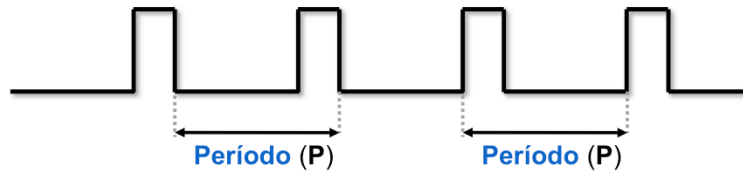
### 6.3. Sinal de Clock

- Sinal de **clock** típico:



$$\text{Frequência (f)} = 1 / \text{Período (P)}$$
$$f = 1 / P$$

- Em certos casos pode não ser simétrico:

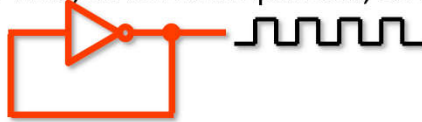


Aqui temos um exemplo típico de um sinal de clock. Temos o estado baixo (0) e alto (1). O tempo entre duas bordas de clock é chamado Período (P), a partir do qual obtemos a frequência (f), em Hertz (Hz) através da fórmula  $f=1/P$ . Nem sempre o período será simétrico.

### 6.3. Sinal de Clock

---

- Como gerar um sinal de **clock**:
  - Normalmente um **oscilador** é usado
  - **Oscilador** pode ser visto, de forma simplificada, como:



- Na prática, osciladores de quartzo são comumente utilizados:



Como gerar um sinal de clock? Podemos usar um oscilador. Ele pode ser visto de forma simplificada e com relação ao seu funcionamento como mostrado no exemplo, e os mais comuns são os osciladores de quartzo.

### 6.3. Sinal de Clock

---

- Funções do **clock** em um circuito digital:
  - **Coordenar** as operações/ações do circuito
  - **Permitir sequenciamento de operações**
- **Clock** é usado em conjunto com **latches** e **flip-flops** para criar barreiras de tempo em um circuito

As funções do clock em um circuito digital são: coordenar as operações/ações do circuito e permitir sequenciamento de operações. O clock é usado em conjunto com latches e flip-flops para criar barreiras de tempo em um circuito.

6.1. Sequencial x Combinacional

6.2. Latches

6.3. Sinal de Clock

## 6.4. Flip-flop Tipo D

---

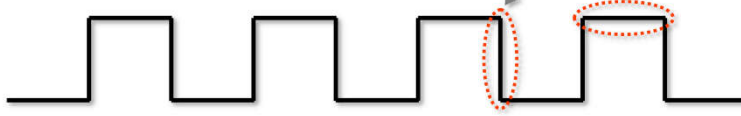
6.5. Flip-flop T

6.6. Flip-flop JK

A partir de agora, veremos os flip-flops, iniciando pelo Flip-flop tipo D.

## 6.4. Flip-flop D

- **Latches** não geram boas barreiras temporais:
- Melhor opção: **Flip-flop**
- Diferença entre **latch** e **flip-flop**
  - **Latch**: pode ser sensível ao nível
  - **Flip-flop**: sensível à borda do relógio

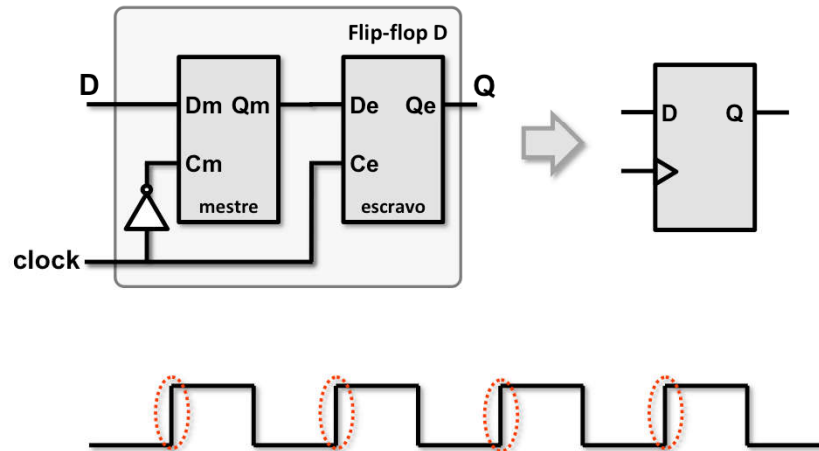


Latches não geram boas barreiras temporais, então a melhor opção para implementar em um circuito sequencial é o flip-flop. A diferença entre os dois é que o latch pode ser sensível ao nível, mas o flip-flop é sensível à borda do relógio, criando uma barreira temporal mais bem definida.



## 6.4. Flip-flop D

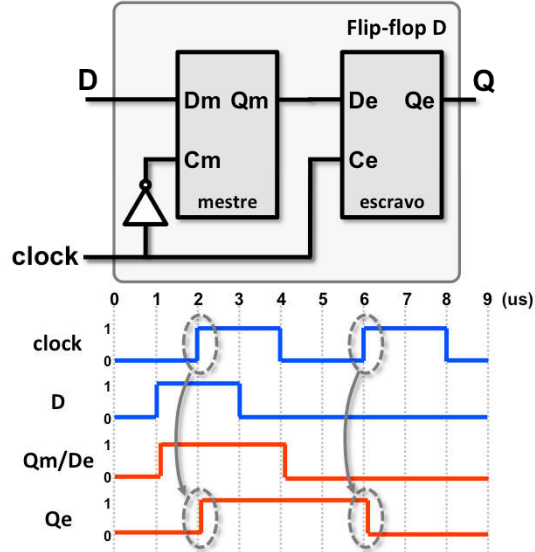
- **Flip-flop D** sensível à borda usando uma estrutura mestre-escravo



O flip-flop D utiliza uma estrutura mestre-escravo em sua construção. Neste exemplo, depende da borda de subida do clock, com a representação gráfica apresentada no slide.

## 6.4. Flip-flop D

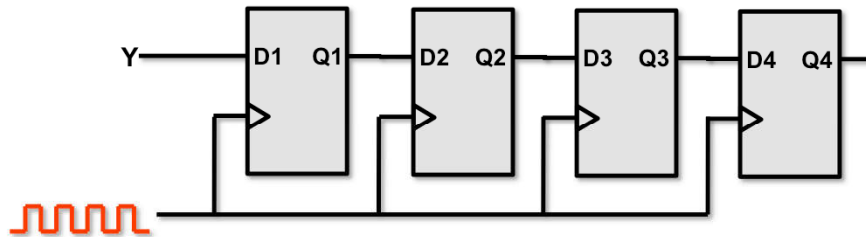
- **Flip-flop D** sensível à borda usando uma estrutura mestre-escravo



O flip-flop tipo D é chamado assim pois a saída Q sempre será igual a entrada D. Porém, o valor da saída Q só será atualizado quando houver uma borda de clock alta, como pode ser visto no diagrama de tempo.

## 6.4. Flip-flop D

- O **Flip-flop D** sensível a borda

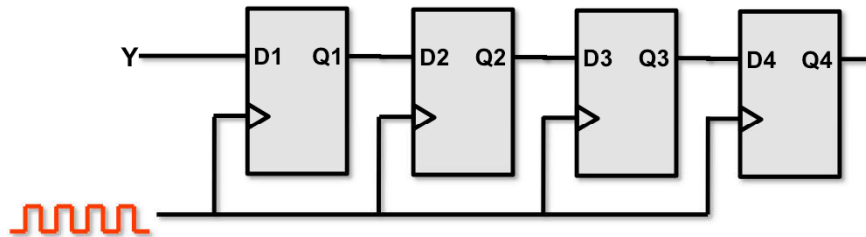


- Através de quantos **flip-flops D** a entrada **Y** irá se propagar após **a transição positiva do clock**?

Considerando este esquema sensível a borda de subida, através de quantos flip-flops D a entrada Y irá se propagar após a transição positiva do clock?

## 6.4. Flip-flop D

- O **Flip-flop D** sensível a borda



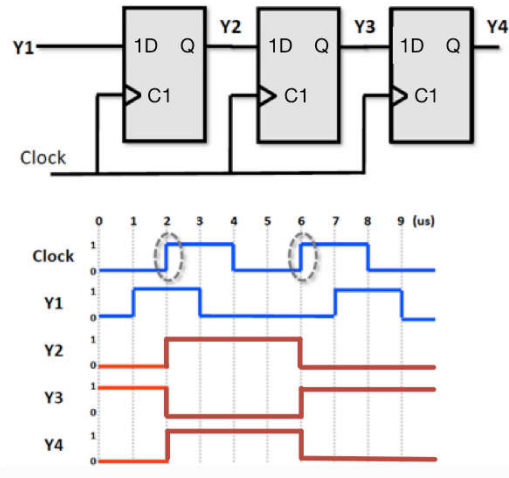
- Através de quantos **flip-flops D** a entrada **Y** irá se propagar após **a transição positiva do clock**?
  - Apenas **1**!

A entrada Y irá se propagar por apenas 1 flip-flop após uma transição positiva do clock.

## PROBLEMAS

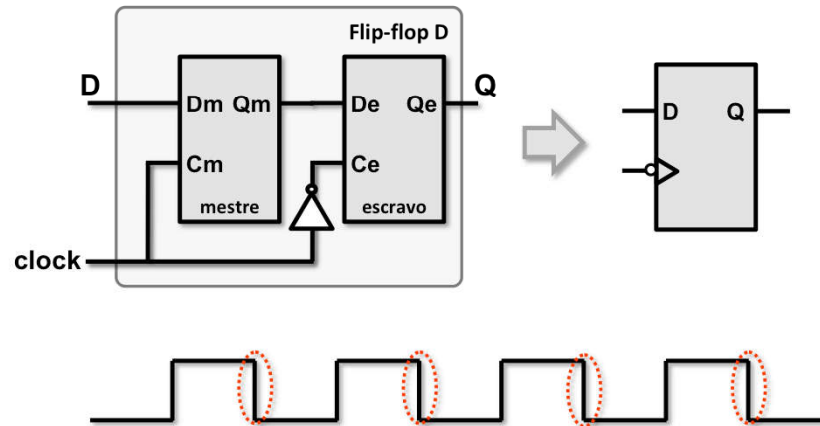
**Problema 6.1.** Complete o diagrama de tempo apresentado na Figura, o qual é relativo ao funcionamento do circuito apresentado.

**Solução 6.1:**



## 6.4. Flip-flop D

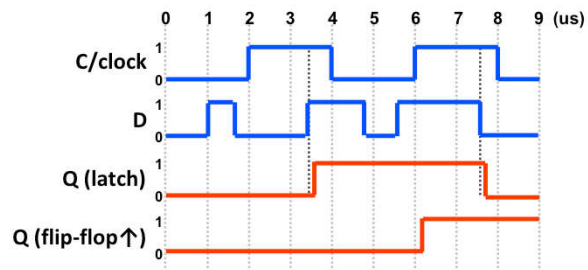
- O **Flip-flop D** sensível a borda
  - Outra forma: sensível à **transição negativa do clock**



Aqui temos um outro exemplo de flip-flop D sensível a borda, porém neste caso, sensível à transição negativa do clock. Repare que a única diferença é a mudança na entrada de clock negada. Observe a diferença na representação gráfica para este flip-flop.

## 6.4. Flip-flop D

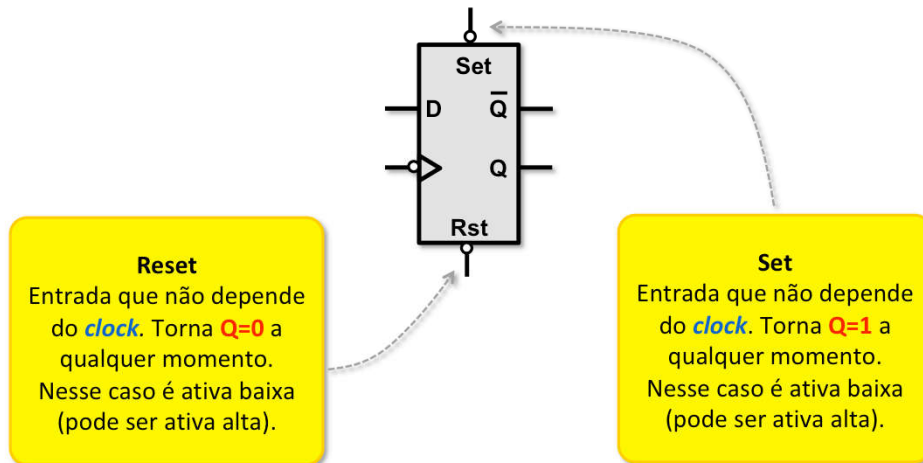
- **Latch D** versus **Flip-flop D**
  - **Latch D** é sensível ao nível (armazena **D** quando **C = 1**)
  - **Flip-flop D** é sensível à borda (armazena **D** quando o **clock** muda de 0 para 1 ou de 1 para 0)
- Comparação no tempo:



Qual a diferença entre o Latch D e o Flip-Flop D? O Latch D é sensível ao nível, ou seja, ele armazena a entrada D quando C=1. Já o Flip-Flop D é sensível à borda, apenas armazena D quando o clock muda de 0 para 1 ou de 1 para 0 (sensível à borda de subida ou de descida, respectivamente).

## 6.4. Flip-flop D

- Um **Flip-flop D** com todos opcionais:

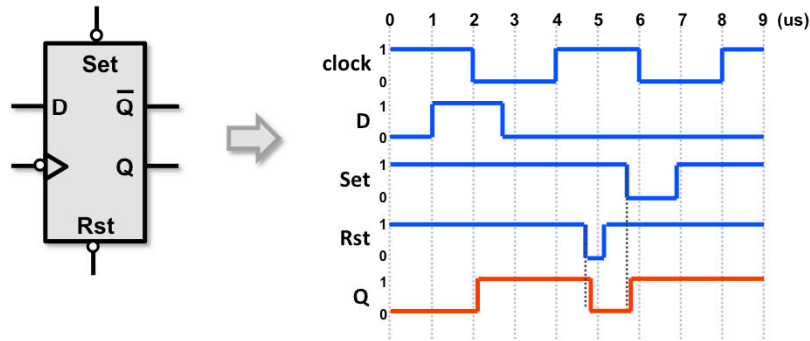


Aqui, temos um flip-flop D com Set e Reset. O set não depende do clock e torna  $Q=1$  a qualquer momento. O reset é o contrário, também não depende do clock e torna  $Q=0$  a qualquer momento.



## 6.4. Flip-flop D

- Um **Flip-flop D** com todos opcionais:

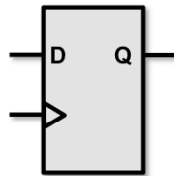


Observe este exemplo através do tempo. Este flip-flop é sensível à borda de descida e possui Set e Reset, e ambos são ativos baixos. Repare que: Em 2us há uma borda de descida do clock. Se olharmos um pouquinho antes desta borda, D=1, logo, Q =1. Próximo a 5us, Reset=0 e isto torna Q=0 imediatamente. Próximo a 6us, Set = 0 e isso torna Q=1.

## 6.4. Flip-flop D

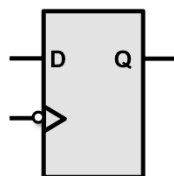
- **Resumo:**

- **Tabelas de função**



$CLK$	$D(t)$	$Q(t+1)$
0	x	$Q(t)$
↑	$D(t)$	$D(t)$
1	x	$Q(t)$

$D(t)$	$Q(t+1)$
0	0
1	1



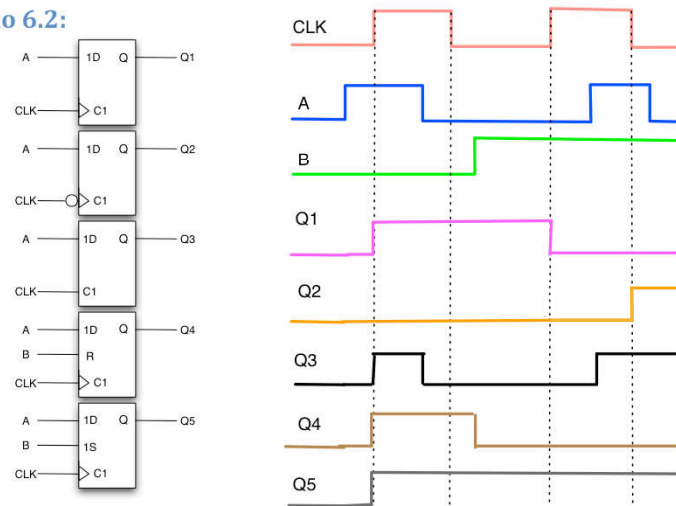
$CLK$	$D(t)$	$Q(t+1)$
0	x	$Q(t)$
↓	$D(t)$	$D(t)$
1	x	$Q(t)$

Aqui temos um resumo do funcionamento dos Flip-Flops D. Para um flip-flop sensível à borda de subida, a saída se atualiza quando há mudança do CLK de 0 para 1. Para um flip-flop sensível à borda de descida, a saída se atualiza quando há mudança do CLK de 1 para 0.

## PROBLEMAS

**Problema 6.2.** Desenhe as formas de onda dos seguintes flip-flops e latches.

**Solução 6.2:**



Para Q1 temos a saída dum flip-flop tipo D sensível a borda de subida.

Para Q2 temos a saída dum flip-flop tipo D sensível a borda de descida.

Para Q3 temos a saída dum latch tipo D sensível a nível alto.

Para Q4 temos a saída dum flip-flop tipo D sensível a borda de subida com reset assíncrono.

Para Q5 temos a saída dum flip-flop tipo D sensível a borda de subida com set síncrono.

6.1. Sequencial x Combinacional

6.2. Latches

6.3. Sinal de Clock

6.4. Flip-flop Tipo D

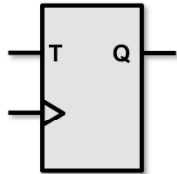
## 6.5. Flip-flop T

---

6.6. Flip-flop JK

Agora veremos o Flip-Flop T.

## 6.5. Flip-flop T



Função

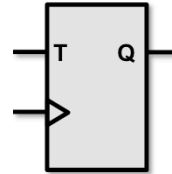
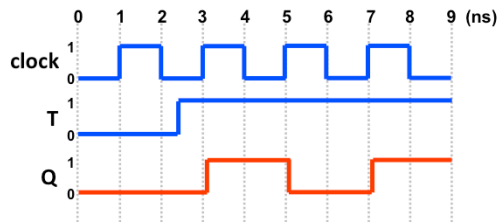
$T(t)$	$Q(t+1)$
0	$Q(t)$
1	$\bar{Q}(t)$

$CLK$	$T(t)$	$Q(t+1)$
0	X	$Q(t)$
↑	0	$Q(t)$
↑	1	$\bar{Q}(t)$

O funcionamento do Flip-Flop T é simples. Temos a entrada T e o clock. Sempre que a entrada  $T=0$ , a saída Q será igual a ela mesma, ou seja, ela é memorizada. Quando  $T=1$ , a saída Q será igual a  $Q'$ , ou seja, o seu valor será invertido. Repare que este Flip-flop é sensível à borda de subida do clock.

## 6.5. Flip-flop T

- Exemplo no tempo:

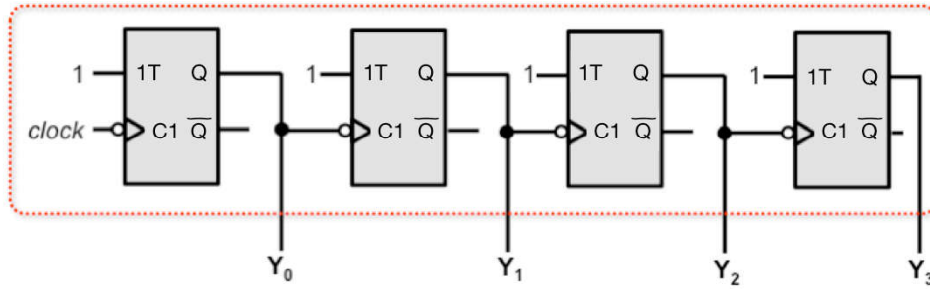


$T(t)$	$Q(t+1)$
0	$\underline{Q}(t)$
1	$\overline{Q}(t)$

Aqui temos um exemplo do seu funcionamento através do tempo. Repare que  $T=1$  entre 2 e 3ns, porém  $Q$  é alterado apenas na borda de subida, quando seu valor se iguala a 1. Em 5ns, temos a mudança de  $Q$  para 0, pois  $T=1$ , e como vimos no slide anterior, sempre que  $T=1$ , a saída  $Q$  é igual a sua inversa. Em 7ns, vemos isso se repetindo quando  $Q$  volta a ser 1.

## PROBLEMAS

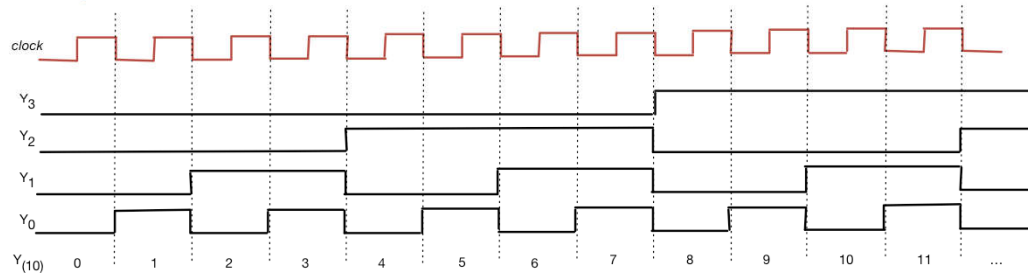
**Problema 6.5.** Identifique a funcionalidade do seguinte circuito.



## PROBLEMAS

**Problema 6.5.** Identifique a funcionalidade do seguinte circuito.

**Solução 6.5:**



A saída Y indica que temos um contador ascendente de 4 bits.



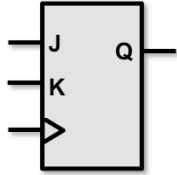
- 6.1. Sequencial x Combinacional
- 6.2. Latches
- 6.3. Sinal de Clock
- 6.4. Flip-flop Tipo D
- 6.5. Flip-flop T

## **6.6. Flip-flop JK**

---

Agora veremos o Flip-flop JK.

## 6.6. Flip-flop JK



$J(t)$	$K(t)$	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

O flip-flop JK elimina o problema que havia no latch RS, com os estados proibidos. Quando J e K igual a 0, a saída será mantida, e quando J e K forem igual a 1, a saída será invertida. Se tivermos apenas K=1, a saída será 0 e quando somente J=1, a saída será 1.

## 6.6. Flip-flop JK

- Usando o FF JK para implementar outros flip-flops

- JK para D:

$J(t)$	$K(t)$	$Q(t+1)$		$D(t)$	$Q(t+1)$
0	0	$Q(t)$		0	0
0	1	0		1	1
1	0	1			
1	1	$\bar{Q}(t)$			

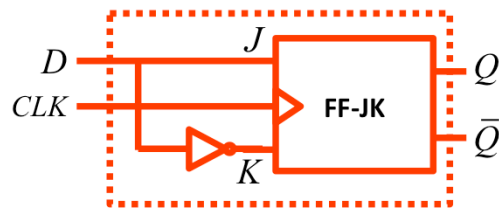
A seguir, veremos como podemos implementar o Flip-Flop tipo D utilizando um Flip-Flop JK.

## 6.6. Flip-flop JK

- Usando o FF JK para implementar outros flip-flops

- JK para D:

$J(t)$	$K(t)$	$Q(t+1)$		$D(t)$	$Q(t+1)$
0	0	$Q(t)$	} →	0	0
0	1	0		1	1
1	0	1			
1	1	$\bar{Q}(t)$			



Lembre que o Flip-Flop D possui apenas uma entrada D, a saída Q deve assumir os valores 0 ou 1, e os estados não se invertem como acontece em um Flip-Flop T. Então, conectando D à J e D' à K, dessa forma, evitamos a situação que ocorre no FF JK, onde J e K possuem valores iguais. Agora, temos um FF-JK funcionando como um FF-D.

## 6.6. Flip-flop JK

- Usando o FF JK para implementar outros flip-flops

- JK para T:

$J(t)$	$K(t)$	$Q(t+1)$	$T(t)$	$Q(t+1)$
0	0	$Q(t)$	0	$Q(t)$
0	1	0	1	$\bar{Q}(t)$
1	0	1		
1	1	$\bar{Q}(t)$		

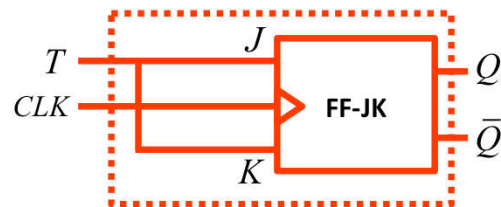
Agora, iremos implementar o Flip-Flop T utilizando o Flip-Flop JK.

## 6.6. Flip-flop JK

- Usando o FF JK para implementar outros flip-flops

- JK para T:

$J(t)$	$K(t)$	$Q(t+1)$		$T(t)$	$Q(t+1)$
0	0	$Q(t)$			
0	1	0	→	0	$\underline{Q(t)}$
1	0	1			
1	1	$\bar{Q}(t)$	→	1	$\bar{Q}(t)$



Lembre-se que um Flip-Flop T inverte a saída atual dependendo do valor de T, que é o que ocorre em um Flip-Flop JK quando J e K possuem valores iguais. Logo, para implementar o FF-T utilizando o FF JK, basta que a entrada T seja conectada a J e a K.



FEDERAL UNIVERSITY  
OF SANTA CATARINA

## EEL5105 – Circuitos e Técnicas Digitais

### Aula 6

---

Prof. Héctor Pettenghi

[hector@eel.ufsc.br](mailto:hector@eel.ufsc.br)

<http://hectorpettenghi.paginas.ufsc.br>

## Exercícios

---

- Exercícios do livro do **Frank Vahid**:
  - **3.1 a 3.4**
  - **3.8 a 3.16**
  - **3.17 a 3.19**



## Exercícios

- Respostas de alguns dos exercícios do livro do **Frank Vahid**:

- 3.1:** a)  $1/50.000 = 20 \text{ us}$ ;  
b)  $1/300.000.000 = 3,33 \text{ ns}$ ;  
c)  $1/(3,4 \times 10^9) = 294 \text{ ps} = 0,294 \text{ ns}$   
d)  $1/(10 \times 10^9) = 100 \text{ ps} = 0,1 \text{ ns}$   
e)  $1 \text{ os}$

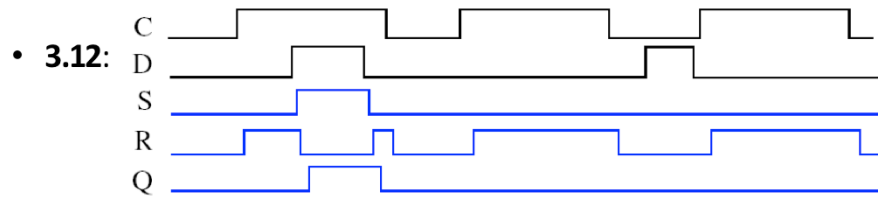
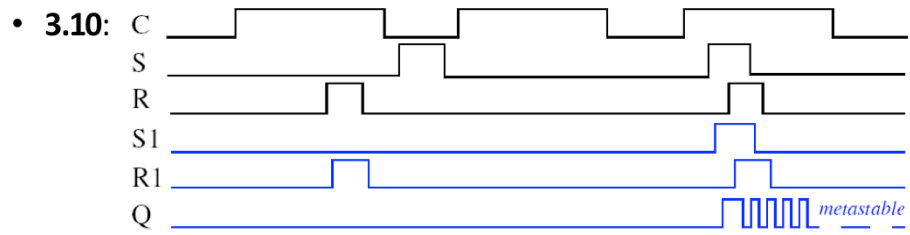
- 3.3:** a)  $1 \text{ Hz}$   
b)  $1/0,001 = 1000 \text{ Hz} = 1 \text{ kHz}$   
c)  $1/(20 \text{ ns}) = 50 \text{ MHz}$   
d)  $1 \text{ GHz}$

- 3.8:**

The diagram shows six digital signals over time. Inputs C, S, and R are black; outputs S1, R1, and Q are blue. C is a periodic square wave. S is a single pulse. R has two pulses. S1 is a single pulse occurring while S is high. R1 has two pulses, each occurring while R is high. Q has a single pulse occurring while both R1 and S1 are high.

## Exercícios

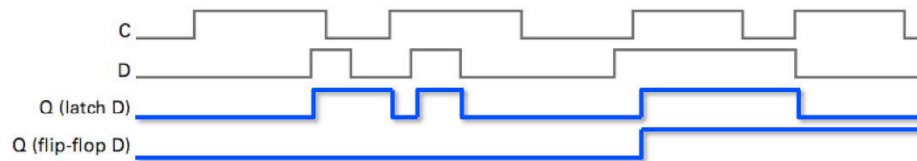
- Respostas de alguns dos exercícios do livro do **Frank Vahid**:



## Exercícios

- Respostas de alguns dos exercícios do livro do **Frank Vahid**:

- 3.16:**



- 3.19:**

