

# VHDL

## Síncronos

Rafael Corsi Ferrão - IMT



`rafael.corsi@maua.br`  
`http://www.maua.br`

6 de outubro de 2014

Circuitos  
Síncronos  
Flip-Flops e  
Latches

Declaração  
Latch

Declaração  
Flip-Flop

Exemplos  
Contador

## 1 Circuitos Síncronos

- Flip-Flops e Latches

## 2 Declaração Latch

## 3 Declaração Flip-Flop

## 4 Exemplos

- Contador

## Circuitos Síncronos

Flip-Flops e Latches

Declaração Latch

Declaração Flip-Flop

Exemplos Contador

### 1 Circuitos Síncronos

- Flip-Flops e Latches

### 2 Declaração Latch

### 3 Declaração Flip-Flop

### 4 Exemplos

- Contador

## Definição

Circuitos síncronos são aqueles que utilizam um sinal de sincronismo para trigar sua execução, esse sinais são conhecidos como *clock*

Em quase toda sua totalidade, projetos digitais são do tipo síncronos. Alguns exemplos:

- ▶ Relógio
- ▶ filtros digitais
- ▶ televisão
- ▶ computador
- ▶ microprocessador
- ▶ ...

### Circuitos Síncronos

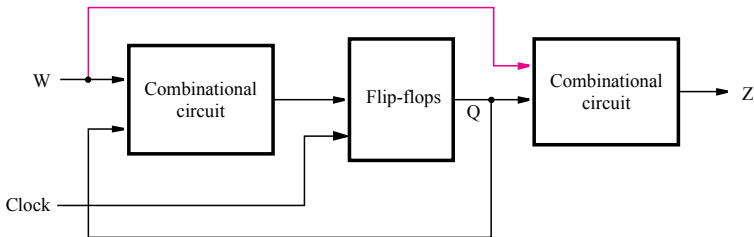
Flip-Flops e Latches

Declaração Latch

Declaração Flip-Flop

Exemplos Contador

- ▶ Para a síntetização de circuitos síncronos, a ferramenta irá utilizar de elementos de memória como Flip-Flop, memórias RAM e ROM.
- ▶ Porém temos que mostrar para o sintetizador que a região descrita deve ser entendida como síncrona.



Principais diferenças entre Flip-Flop e Latches são:

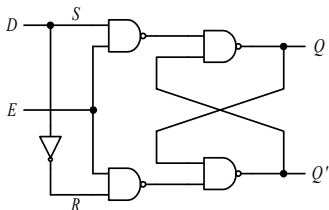
- ▶ Flip-flops são sensíveis a transições, latches são sensíveis a níveis
- ▶ Use preferencialmente em projetos com FPGA os flip-flops;
- ▶ Latches apresentam problema com temporização (veremos mais tarde)

Circuitos  
Síncronos  
Flip-Flops e  
Latches

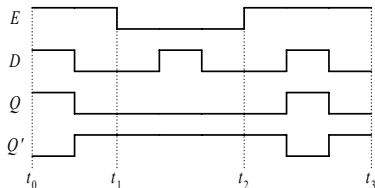
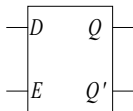
Declaração  
Latch

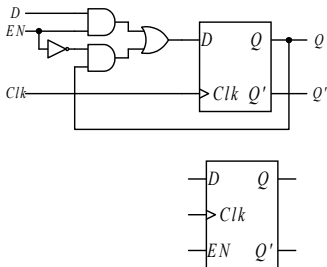
Declaração  
Flip-Flop

Exemplos  
Contador



$E$	$D$	$Q$	$Q_{next}$	$Q_{next}'$
0		0	0	1
0		1	1	0
1	0		0	1
1	1		1	0





$Clk$	$EN$	$D$	$Q$	$Q_{next}$	$Q_{next}'$
0			0	0	1
0			1	1	0
1			0	0	1
1			1	1	0
$\uparrow$	0		0	0	1
$\uparrow$	0		1	1	0
$\uparrow$	1	0		0	1
$\uparrow$	1	1		1	0



# Flip-Flop

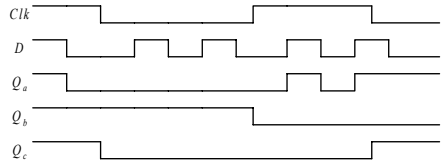
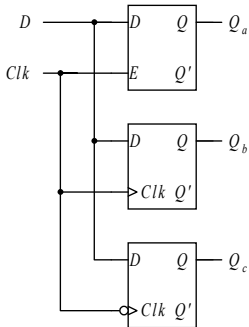
## Carta de tempo

Circuitos  
Síncronos  
**Flip-Flops e  
Latches**

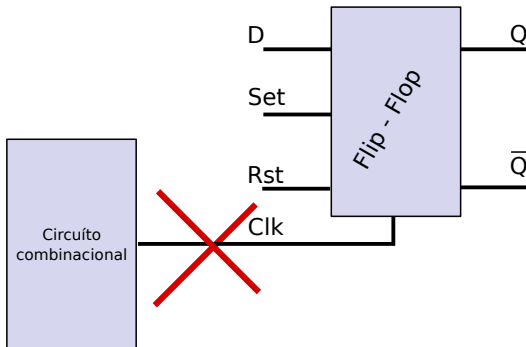
Declaração  
Latch

Declaração  
Flip-Flop

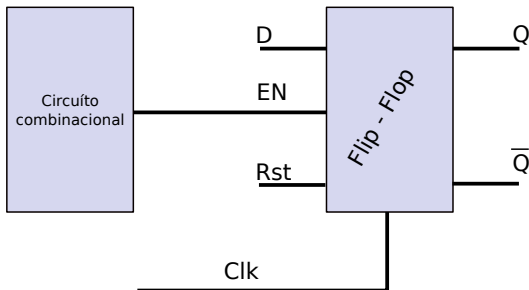
Exemplos  
Contador



Nunca deve-se usar uma lógica para gerar o sinal para trigar um clock:



Porém, deve-se utilizar o circuito combinacional como enable ou rst para o flip-flop



Circuitos  
Síncronos

Flip-Flops e  
Latches

**Declaração  
Latch**

Declaração  
Flip-Flop

Exemplos  
Contador

## 1 Circuitos Síncronos

- Flip-Flops e Latches

## 2 Declaração Latch

## 3 Declaração Flip-Flop

## 4 Exemplos

- Contador

Circuitos  
Síncronos  
Flip-Flops e  
Latches

Declaração  
Latch

Declaração  
Flip-Flop

Exemplos  
Contador

```
ENTITY latch IS
    PORT(
        rst : IN  STD_LOGIC; -- entradas
        set  : IN  STD_LOGIC;
        d    : IN  STD_LOGIC;
        q    : OUT STD_LOGIC
    );
END latch;

ARCHITECTURE bhv OF latch IS
BEGIN

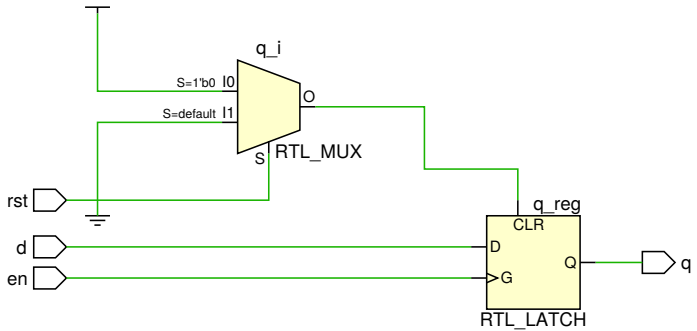
    PROCESS(rst, en, d)
    BEGIN
        IF rst = '0' THEN
            q <= '0';
        ELSIF set = '1' THEN
            q <= d;
        END IF;
    END PROCESS;
END bhv;
```

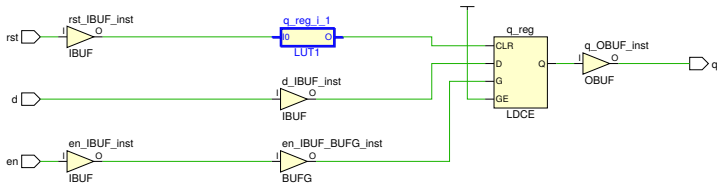
Circuitos  
Síncronos  
Flip-Flops e  
Latches

**Declaração  
Latch**

Declaração  
Flip-Flop

Exemplos  
Contador





LDCE : Transparent Data Latch with Asynchronous Clear and Gate Enable

Circuitos  
Síncronos

Flip-Flops e  
Latches

Declaração  
Latch

**Declaração  
Flip-Flop**

Exemplos  
Contador

## 1 Circuitos Síncronos

- Flip-Flops e Latches

## 2 Declaração Latch

## 3 Declaração Flip-Flop

## 4 Exemplos

- Contador



Circuitos  
Síncronos

Flip-Flops e  
Latches

Declaração  
Latch

**Declaração  
Flip-Flop**

Exemplos  
Contador

- ▶ Sensível a borda
  - ▶ subida OU decida
- ▶ Reset síncrono ou assíncrono

Sensível a borda de subida, reset síncrono:

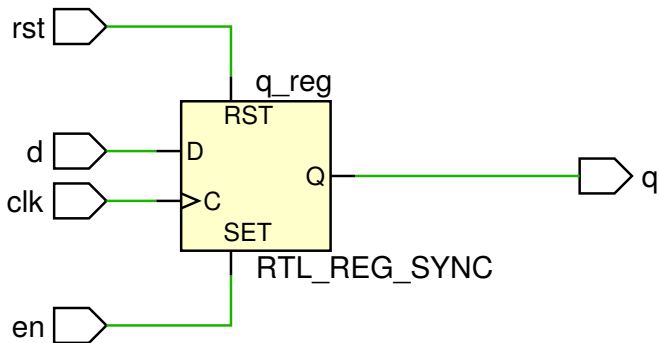
```
PROCESS (clk)
BEGIN
    IF clk'EVENT AND clk='1' THEN
        IF rst='1' THEN
            q <= '0';
        ELSIF set = '1' THEN
            q <= '1';
        ELSE
            q <= d;
        END IF;
    END IF;
END PROCESS;
```

Sensível a borda de descida:

```
IF clk'EVENT AND clk='0' THEN
```

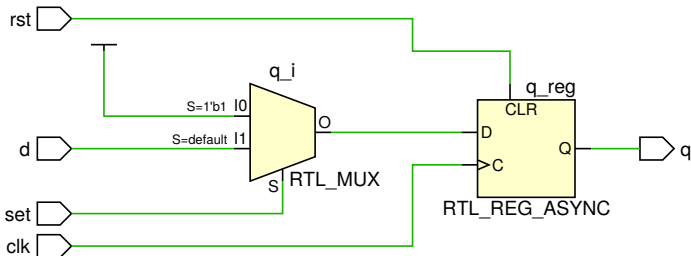
Pode-se utilizar também, em substituição ao 'EVENT:

```
IF RISING_EDGE(clk) THEN = IF clk'EVENT AND clk='1' THEN
IF FALLING_EDGE(clk) THEN = IF clk'EVENT AND clk='0' THEN
```



Sensível a borda de subida, reset assíncrono:

```
PROCESS (clk)
BEGIN
    IF rst = '1' THEN
        q <= '0';
    ELSIF RISING_EDGE(clk) THEN
        IF set = '1' THEN
            q <= '1';
        ELSE
            q <= d;
        END IF;
    END IF;
END PROCESS;
```



Circuitos  
Síncronos

Flip-Flops e  
Latches

Declaração  
Latch

Declaração  
Flip-Flop

Exemplos  
Contador

## 1 Circuitos Síncronos

- Flip-Flops e Latches

## 2 Declaração Latch

## 3 Declaração Flip-Flop

## 4 Exemplos

- Contador

Circuitos  
Síncronos

Flip-Flops e  
Latches

Declaração  
Latch

Declaração  
Flip-Flop

Exemplos  
**Contador**