# VHDL

Circuitos Síncronos



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

13 de outubro de 2014



### Conteúdo

Conceitos

Síncronos Flip-Flops e Lat chs Declaração Lat ch

Descrição síncrona

- 1 Conceitos
- 2 Circuitos Síncronos
  - Flip-Flops e Latchs
- 3 Declaração Latch
- 4 Descrição síncrona
- 5 exemplos



### Conteúdo

#### Conceitos

Circuitos Síncronos Flip-Flops e Lat chs Declaração

Latch

Descrição
síncrona

- 1 Conceitos
- 2 Circuitos Síncrono
  - Flip-Flops e Latchs
- 3 Declaração Latel
- 4 Descrição síncron
- 5 exemplo



#### Conteúdo

Conceitos

#### Circuitos Síncronos

Flip-Flops e Lat chs Declaração Lat ch

Descrição síncrona

- 1 Conceitos
- 2 Circuitos Síncronos
  - Flip-Flops e Latchs
- 3 Declaração Latch
- 4 Descrição síncrona
- 5 exemplos



#### Circuitos Síncronos

Flip-Flops e Lat chs

Descrição síncrona

Latch

# Definição

Circuitos síncronos são aqueles que utilizam um sinal de sincronismos 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 Lat chs Declaração

Latch

Descrição
síncrona

ex e<mark>m plo</mark>s

- ► Circuitos síncronos possuem necessariamente em sua formação elementos de memória (flip-flop)
- ▶ o clock é o sinal responsável pela sincronização do circuito
- ► as mudanças dos estados (variáveis e saídas) ocorrem na transição do *clock* borda de subida **ou** borda de descida.



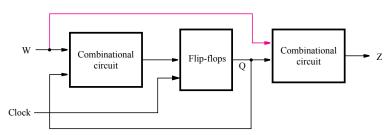
#### Circuitos Síncronos

Flip-Flops e Lat chs Declaração

Latch

Descrição síncrona

- Para a sintetização 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.





### Latch vs Flip-Flop

Conceitos

Circuitos Síncronos Flip-Flops e Lat chs Declaração

Latch

Descrição
síncrona

ex em plos

Principais diferenças entre Flip-Flop e Latchs são:

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



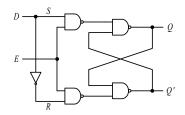
Circuitos Síncronos

Flip-Flops e Lat chs

Declaração Latch

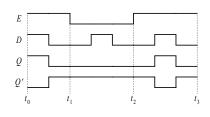
Descrição síncrona

 ${\sf ex}\, {\sf emplos}$ 



| Е | D | Q | $Q_{next}$ | $Q_{next}'$ |
|---|---|---|------------|-------------|
| 0 |   | 0 | 0          | 1           |
| 0 |   | 1 | 1          | 0           |
| 1 | 0 |   | 0          | 1           |
| 1 | 1 |   | 1          | 0           |





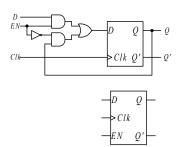


Circuitos Síncronos

Flip-Flops e Lat chs

Declaração Latch

Descrição síncrona



| Clk | EN | D | Q | $Q_{next}$ | $Q_{next}'$ |
|-----|----|---|---|------------|-------------|
| 0   |    |   | 0 | 0          | 1           |
| 0   |    |   | 1 | 1          | 0           |
| 1   |    |   | 0 | 0          | 1           |
| 1   |    |   | 1 | 1          | 0           |
| ₹   | 0  |   | 0 | 0          | 1           |
| ₹   | 0  |   | 1 | 1          | 0           |
| _₹  | 1  | 0 |   | 0          | 1           |
| F   | 1  | 1 |   | 1          | 0           |
|     |    |   |   |            |             |



Circuitos Síncronos

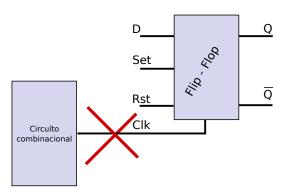
Flip-Flops e Lat chs

Declaração Latch

Descrição síncrona

ex emplos

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





Lat chs

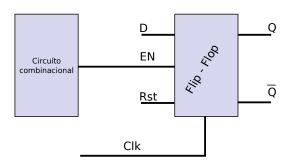
Declaração
Lat ch

Descrição

sín crona

ex emplos

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





### Conteúdo

Conceitos

Circuitos Síncronos Flip-Flops e Latchs

Declaração Latch

Descrição síncrona

- 1 Conceitos
- 2 Circuitos Síncronos
  - Flip-Flops e Latchs
- 3 Declaração Latch
- 4 Descrição síncron
- 5 exemplos



#### Latch

ENTITY latch IS

```
Conceitos
Circuitos
Síncronos
```

```
Flip-Flops e
Lat chs
Declaração
Lat ch
```

sín crona

```
PORT(
      rst : IN STD_LOGIC; -- entradas
      set : IN STD_LOGIC;
      d
          : IN STD_LOGIC;
          : OUT STD_LOGIC
      q
   );
END latch;
ARCHITECTURE bby OF latch IS
BEGIN
   PROCESS(rst, en, d)
   BEGIN
       IF rst = '0' THEN
        q <= '0';
       ELSIF set = '1' THEN
        q \ll d;
       END IF:
   END PROCESS:
END bhv;
```

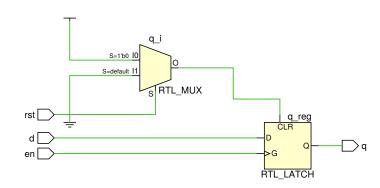


Circuitos Síncronos

Flip-Flops e Lat chs

Declaração Latch

Descrição síncrona





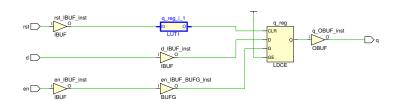
Circuitos Síncronos

Flip-Flops e Lat chs

Declaração Latch

Descrição síncrona

ex emplos



LDCE : Transparent Data Latch with Asynchronous Clear and Gate Enable



### Conteúdo

Conceitos Circuitos

Síncronos Flip-Flops e Lat chs Declaração Lat ch

Descrição síncrona

- 1 Conceitos
- 2 Circuitos Síncronos
  - Flip-Flops e Latchs
- 3 Declaração Latch
- 4 Descrição síncrona
- 5 exemplos



Síncronos Flip-Flops e Lat chs Declaração

Latch

Descrição
síncrona

xemplos

Circuitos síncronos em VHDL são implementados em regiões de códigos sequenciais (**process**).

Uma transição em VHDL pode ser detectada de duas maneiras diferentes :

```
(ck'EVENT and CK = '1') -- Borda de subida
(ck'EVENT and CK = '0') -- Borda de descida
(rising_edge(ck)) -- Borda de subida
(falling_edge(ck)) -- Borda de descida
```

Onde ck é o um sinal/entrada que propaga o clock .



```
entity sincrono is
                    Port ( clk : in std_logic;
Conceitos
                            rst : in std_logic;
Síncronos
                               : in std_logic;
Flip-Flops e
                                 : out std_logic
                            q
Latchs
                           );
Declaração
Latch
                end sincrono;
Descrição
síncrona
                architecture Behavioral of sincrono is
                begin
                    process(clk)
                    begin
                         if(rising_edge(clk)) then
                              if(rst = '1') then
                                  q <= '0';
                             else
                                  q \le d;
                             end if;
                         end if;
                    end process;
     MAUÁ
                end Behavioral;
                Rafael Corsi Ferrão - IMT
                                                      VHDL
                                                                                19/28
```

Registrador sensível a borda

rst síncrono

#### RTL

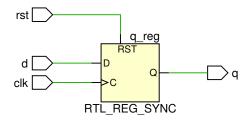
Registrador sensível a borda de subida - rst síncrono

Conceitos

Circuitos Síncronos Flip-Flops e Latchs

Declaração Latch Descrição

sín crona ex em plos





```
entity sincrono is
Conceitos
                    Port ( clk : in std_logic;
                           rst : in std_logic;
Síncronos
Flip-Flops e
                              : in std_logic;
Latchs
                              : out std_logic
Declaração
Latch
                           );
               end sincrono:
Descrição
síncrona
ex emplos
               architecture Behavioral of sincrono is
               begin
                    process(clk, rst)
                    begin
                     if(rst = '1') then
                         q <= '0';
                     elsif(clk'EDGE = '1' and clk = '1') then
                         q \le d;
                     end if:
```

end process;
end Behavioral;

Registrador sensível a borda

rst assíncrono

VHDL

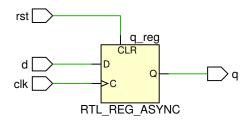
#### RTL

Registrador sensível a borda de subida - rst assíncrono

Conceitos

Circuitos Síncronos Flip-Flops e Lat chs

Declaração Latch Descrição síncrona





```
Port ( clk : in std_logic;
                             rst : in std_logic;
Síncronos
                             set : in std_logic;
Flip-Flops e
Latchs
                               : in std_logic;
                                : out std_logic
Declaração
                             q
Latch
                            ):
                 end sincrono;
Descrição
síncrona
                 architecture Behavioral of sincrono is
ex emplos
                 begin
                     process(clk, rst)
                     begin
                      if(rst = '1') then
                          q <= '0';
                      elsif(set = '1') then
                          q <= '1';
                      elsif(clk'EDGE = '1' and clk = '1') then
                          q \le d;
                      end if;
```

VHDL

23/28

Registrador sensível a borda

rst assincrono e set
entity sincrono is

end process;
end Behavioral;

Rafael Corsi Ferrão - IMT

Conceitos

MAUÁ

# Conteúdo

- Conceitos
- Circuitos Síncronos Flip-Flops e
- Lat chs

  Declaração
- Latch Descrição
- síncrona
- ex emplos

- 1 Conceitos
- 2 Circuitos Síncronos
  - Flip-Flops e Latchs
- 3 Declaração Latel
- 4 Descrição síncron
- 5 exemplos



### Contador de 0 a 100

```
Conceitos
                 entity contador is
                     Port ( clk : in std_logic;
Síncronos
                             rst : in std_logic;
Flip-Flops e
                                : in std_logic;
Lat chs
                                 : out std_logic
                             q
Declaração
Latch
                 end contador:
síncrona
                 architecture Behavioral of contador is
ex emplos
                     signal cnt : integer range 0 to 100;
                 begin
                     process(clk, rst)
                     begin
                          if(rst = '1') then
                              cnt <= 0:</pre>
                          elsif(rising_edge(clk)) then
                              cnt <= cnt + 1;
                          end if:
                     end process;
```



end Behavioral:

# Pisca led a cada 1 segundo com clk de 100Mhz

```
Conceitos
Circuitos
Síncronos
```

Flip-Flops e Lat chs Declaração

Latch Descriçã síncrona

ex emplos

```
Port ( clk : in std_logic; -- 100 Mhz
           rst : in std logic:
           led : out std_logic
end contador led:
architecture Behavioral of contador_led is
    signal cnt : integer range 0 to 100_000_000 := 0;
    signal led_i : std_logic := '0';
begin
    process(clk, rst)
    begin
        if (rst = '1') then
            cnt <= 0:
        elsif(rising_edge(clk)) then
            if (cnt = 100 000 000) then
                 cnt <= 0;
                led_i <= not led_i;</pre>
            else
                 cnt <= cnt + 1;
            end if;
        end if;
    end process;
   led <= led_i;</pre>
end Behavioral:
```

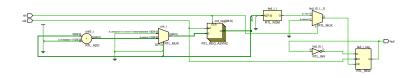
entity contador\_led is



Circuitos Síncronos Flip-Flops e Lat chs

Declaração Latch

Descrição síncrona





### Esquemático Síntese

Conceitos

Circuitos Síncronos Flip-Flops e Lat chs

Latichs Declaração

Latch

Descrição síncrona

