



# LABORATÓRIO DE CIRCUITOS DIGITAIS

## LABORATÓRIO 5 – REGISTRADORES

TIAGO BARROS

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN

CENTRO DE TECNOLOGIA – CT

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO – DCA

# SUMÁRIO

1. Circuitos Sequenciais x Circuitos Combinacionais
2. PROCESS
  - Sintaxe
3. IF-ELSE-ELSIF
  - Sintaxe
4. Latch
  - Latches SR
5. Flip-Flops
6. Registradores
7. Tarefas

# CIRCUITOS SEQUENCIAIS X CIRCUITOS COMBINACIONAIS

- Um circuito combinacional não tem memória;
- Um circuito combinacional é um circuito que depende apenas da entrada atual;
- Um circuito sequencial é um circuito cujas saídas dependem não somente das entradas atuais, mas também do seu **estado** atual.

# CIRCUITOS SEQUENCIAIS

- Em circuitos sequenciais, a ordem de execução importa!
- O VHDL é uma linguagem não-sequencial, então é necessário uma estrutura que o torne sequencial.
- A estrutura que permite isto é o PROCESS.

# PROCESS

- Sintaxe:

```
1  PROCESS (A,B)
2  BEGIN
3  -- Comandos
4  END PROCESS ;
```

- Conceito importante : O trecho entre **BEGIN** e **END** é executado sequencialmente (a ordem em que os códigos estão apresentados, importa);
- O processo é executado concorrentemente com as demais declarações;
- O processo é invocado, quando há uma mudança em algum sinal na lista de sensibilidade.

# IF-ELSE-ELSIF

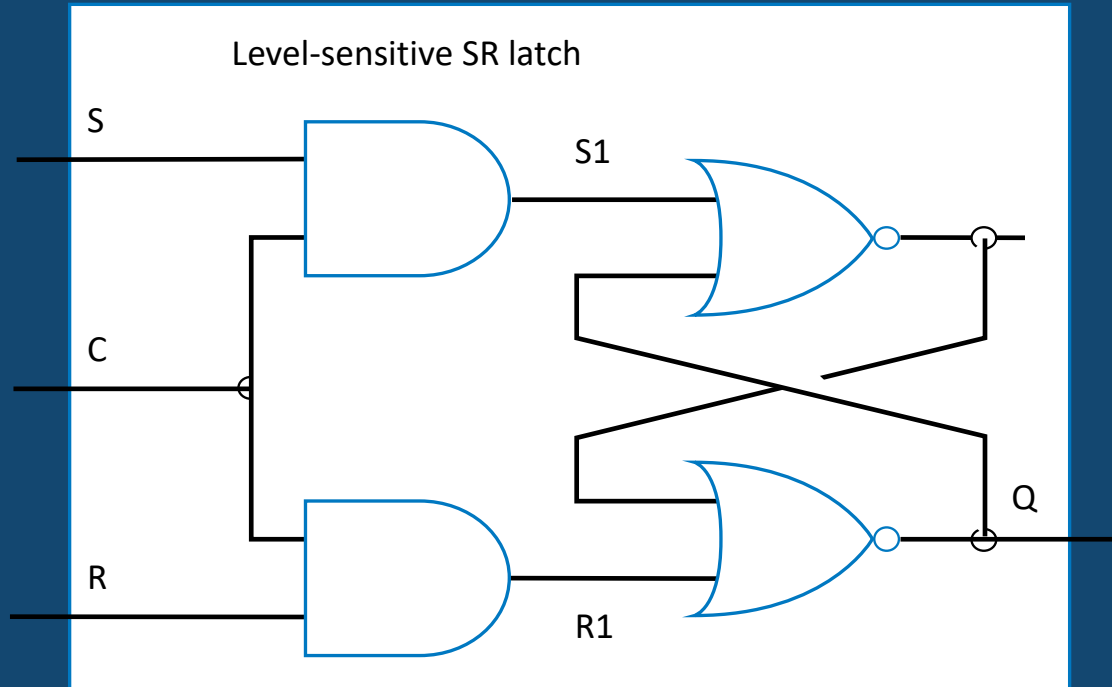
- A estrutura IF-ELSE-ELSIF trata-se de um comando sequencial: só pode ser utilizada em regiões de código sequencial;
- Funciona de forma semelhante à estrutura padrão de IF da linguagem de alto nível, obedecendo a seguinte sintaxe:

```
1  IF( CONDICA0 ) THEN
2  -- Comandos caso condicao for verdadeira
3  ELSIF ( CONDICA02 )
4  -- Comandos caso condicao2 for verdadeira
5  ELSE
6  -- Comandos caso nenhuma das condicoes for verdade
7  END IF;
```

# LATCH

- Um latch é um dispositivo de armazenamento temporário que tem dois estados estáveis;
- Os latches são similares aos flip-flops por serem dispositivos biestáveis, ou seja, podem permanecer em um de dois estados estáveis usando uma configuração de realimentação, na qual as saídas são ligadas nas entradas opostas.
- A principal diferença entre os latches e os flip-flops é o método usado para a mudança de estado.

# LATCH SR (SENSÍVEL AO NÍVEL)



C	S	R	Q
0	0	0	Q
0	0	1	Q
0	1	0	Q
0	1	1	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	X

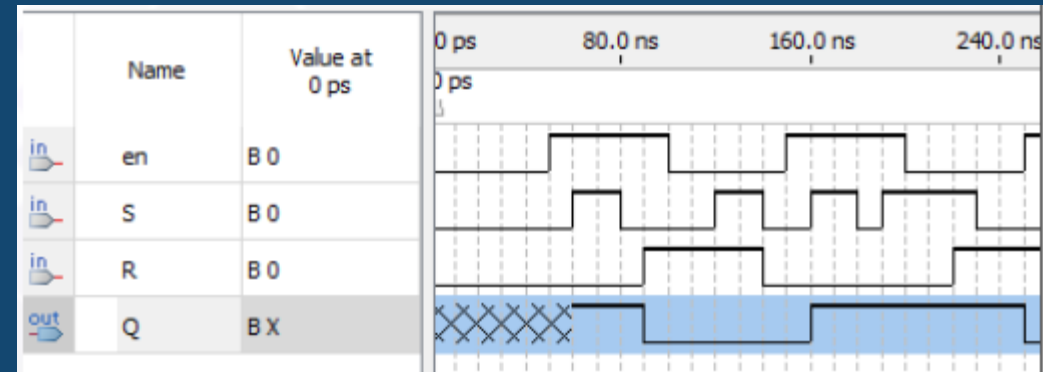


# CÓDIGO EM VHDL (COMPORTAMENTAL)

```
1  LIBRARY ieee;
2  USE ieee . std_logic_1164 . all;
3  ENTITY latchSR IS
4  PORT (S,R,c : IN BIT;
5        |      |      Q : OUT BIT );
6  END ;
7  ARCHITECTURE behav OF LatchSR IS
8  BEGIN
9  PROCESS (S,R,c)
10 BEGIN
11     IF(c = '1' AND S = '1') THEN
12         Q <= '1';
13     elsifIF (c = '1' AND R = '1') THEN
14         Q <= '0';
15     END IF;
16 END PROCESS;
17 END;
```

# DESCRIÇÃO E SIMULAÇÃO

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  ENTITY latchSR IS
4  PORT (S,R,en : IN BIT;
5        Q : OUT BIT);
6  END;
7  ARCHITECTURE behav OF LatchSR IS
8  BEGIN
9  PROCESS (S,R,EN)
10 BEGIN
11
12     IF (EN = '1' AND S = '1') THEN
13         Q <= '1';
14     elsif (EN = '1' AND R = '1') THEN
15         Q <= '0';
16     END IF;
17
18 END PROCESS;
19 END;
```



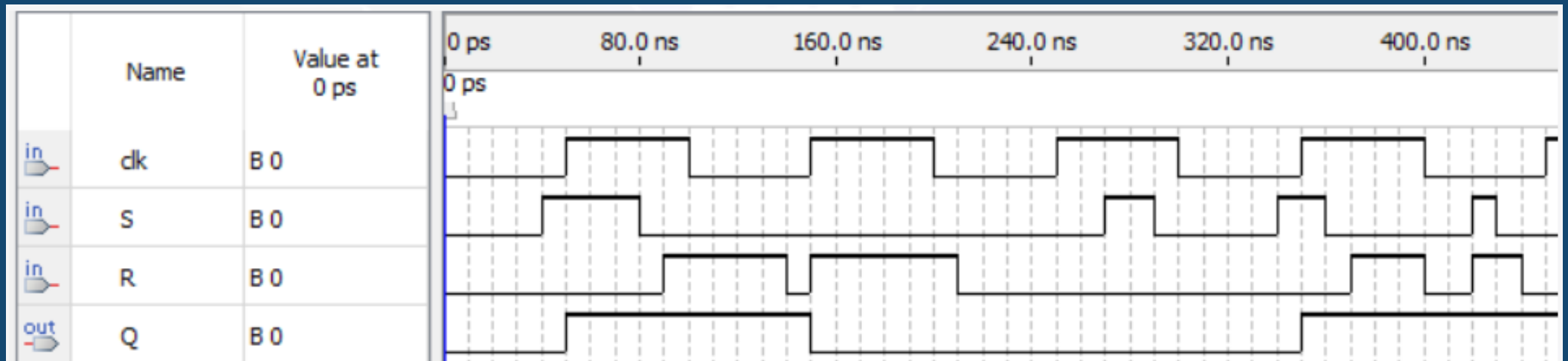
# FLIP-FLOPS

- Um problema dos latches é que se o valor da entrada muda enquanto o sinal **C** está ativado, ele ainda é levado em consideração;
- **Meta** : Estabelecer um instante preciso para que a informação possa ser armazenada.
- Flip-Flops só registram o bit quando há bordas de subidas;

# CÓDIGO EM VHDL (COMPORTAMENTAL)

```
1  LIBRARY ieee;
2  USE ieee . std_logic_1164 . all;
3  ENTITY flipflopSR IS
4  PORT (S,R, clk : IN BIT ;
5        |   |   | Q : OUT BIT );
6  END ;
7  ARCHITECTURE behav OF flipflopSR IS
8  BEGIN
9  PROCESS (S,R, clk)
10 BEGIN
11     IF(clk ' EVENT AND clk = '1' AND S = '1') THEN
12         Q <= '1';
13     ELSIF (clk ' EVENT AND clk = '1' AND R = '1') THEN
14         Q <= '0';
15     END IF;
16 END PROCESS;
17 END;
```

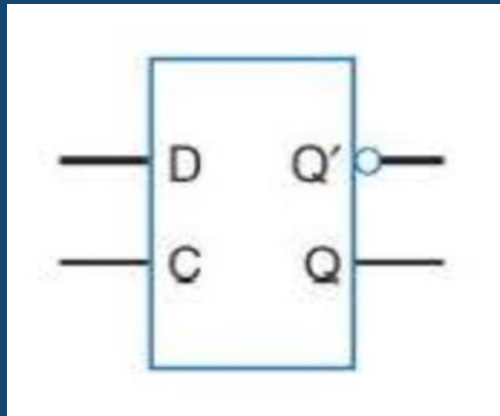
# DESCRIÇÃO E SIMULAÇÃO



# REGISTRADORES

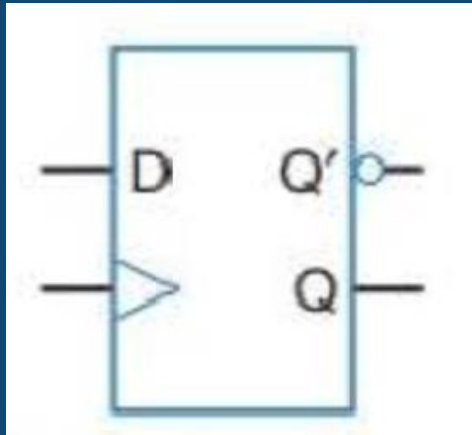
- Registradores são uma sequencia de latches ou flip-flops;
- Por possuírem um estado em que a saída é indesejável (estado em que  $SR = 11$ ), latches e flip-flops SR não são usados para criar registradores.
- Em geral, são usados latches ou flip-flops D;

# LATCH D

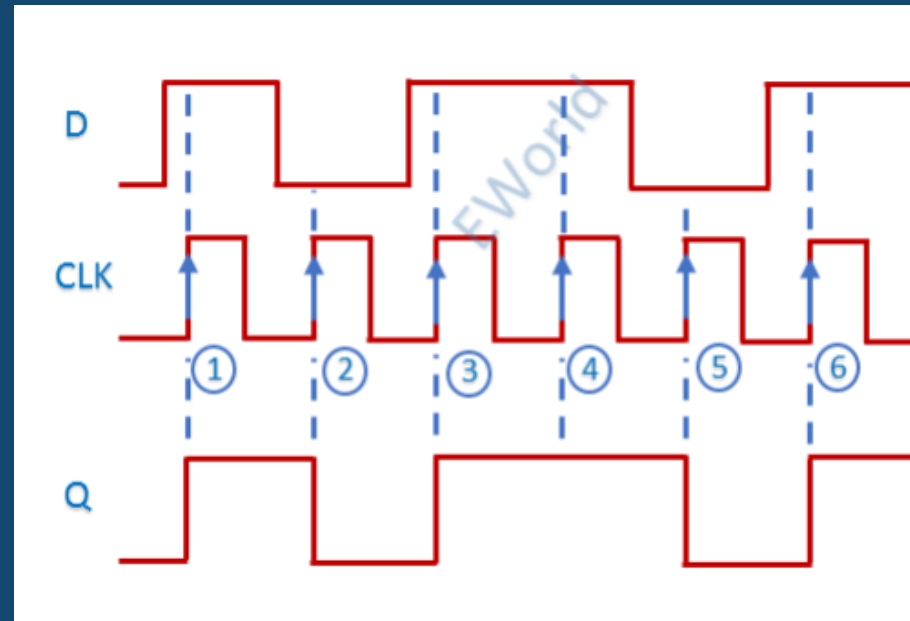


clk	D	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

# FLIP-FLOP D

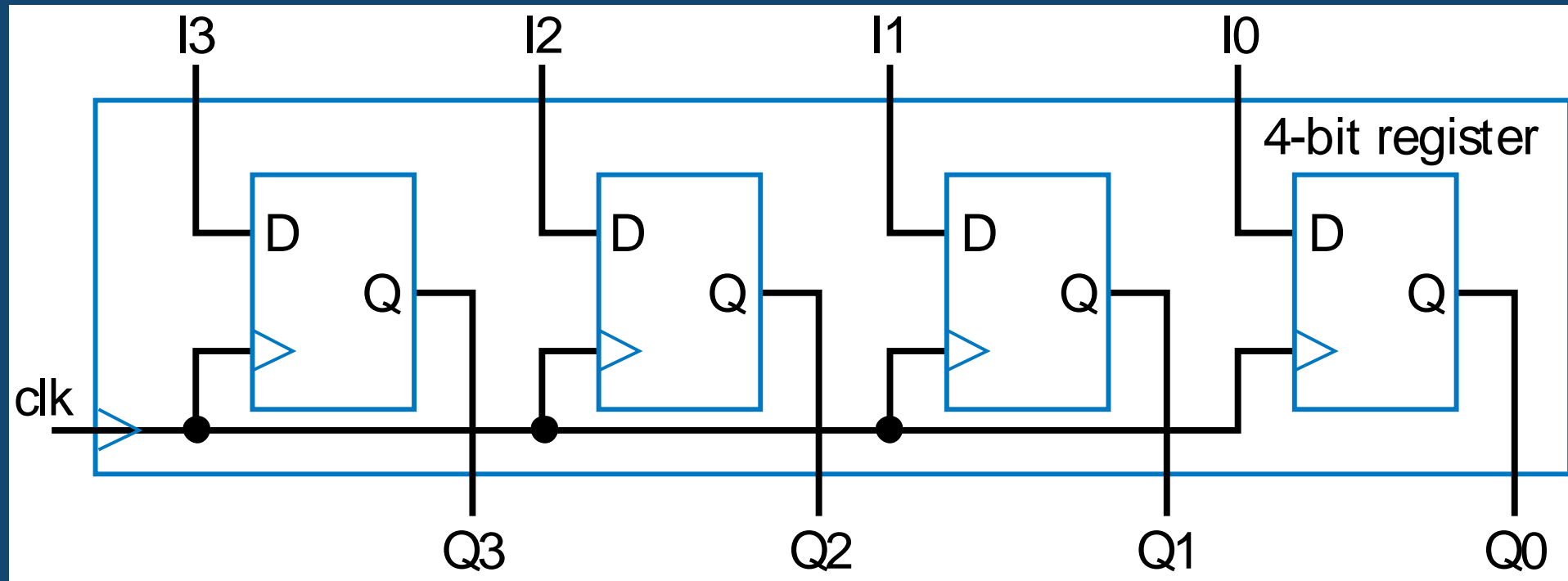


D	CLK	SAÍDA
0	↑	$Q = 0$
1	↑	$Q = 1$





# REGISTRADORES: FUNCIONAMENTO



# TAREFAS: LATCHES E FLIP-FLOPS

1. Utilizando a linguagem VHDL, crie um latch D para 1 Bit.
2. Utilizando a linguagem VHDL, desenvolva um flip-flop D para 1 bit.
3. Utilizando a linguagem VHDL, desenvolva um registrador de quatro bits, baseado em latches D.
4. Utilizando a linguagem VHDL, desenvolva um registrador de quatro bits, baseado em flip-flops D.
5. Elabore e entregue um relatório com a descrição do experimento, códigos desenvolvidos e resultados de simulações que comprovem o funcionamento dos componentes desenvolvidos nos itens 1 a 4.

ATENÇÃO: Utilize como base a figura apresentada no slide 17 para a criação do registradores.  
NÃO UTILIZEM BIT\_VECTOR.