



Sistemas Digitais

Aulas práticas de laboratório

Comandos concorrentes do VHDL

Prof. Dr. Eng. Rafael Luiz Cancian





Introdução



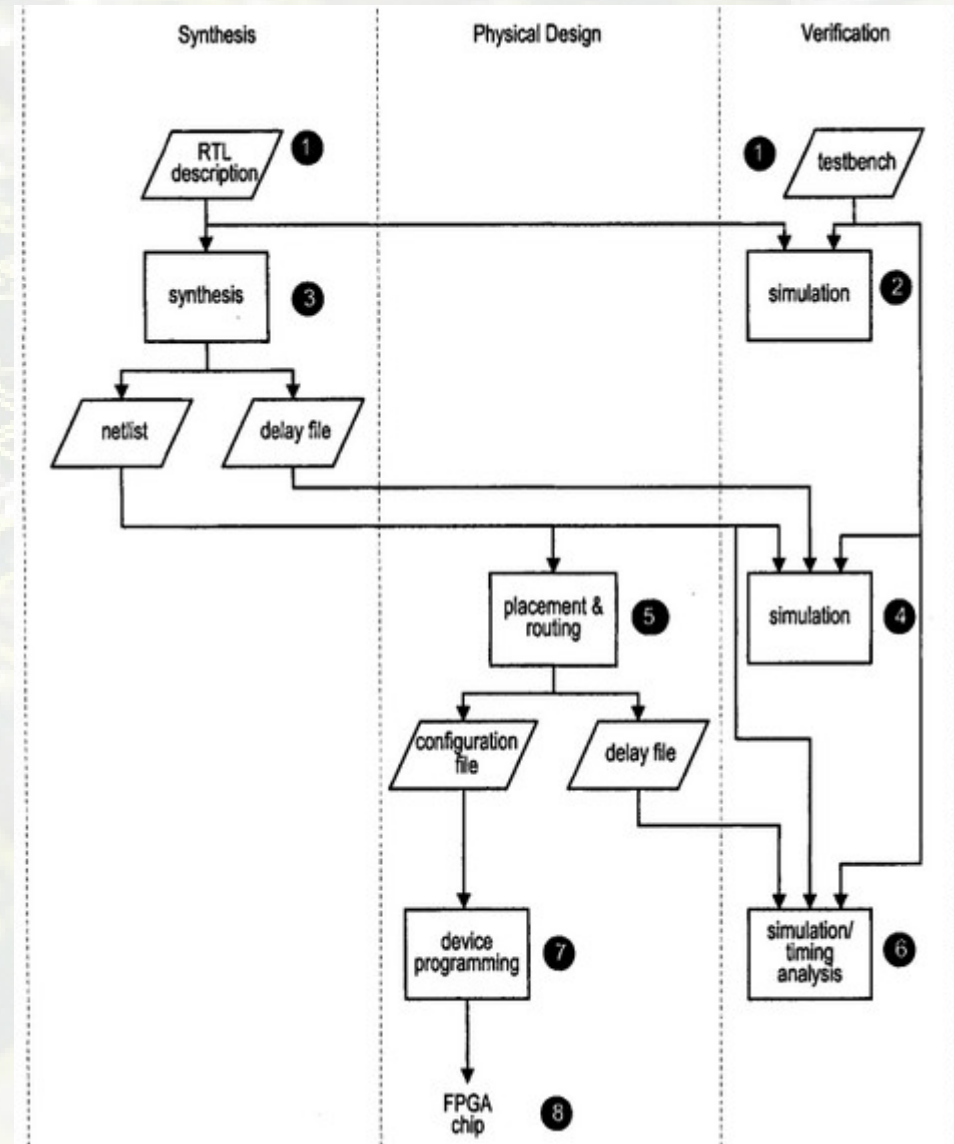
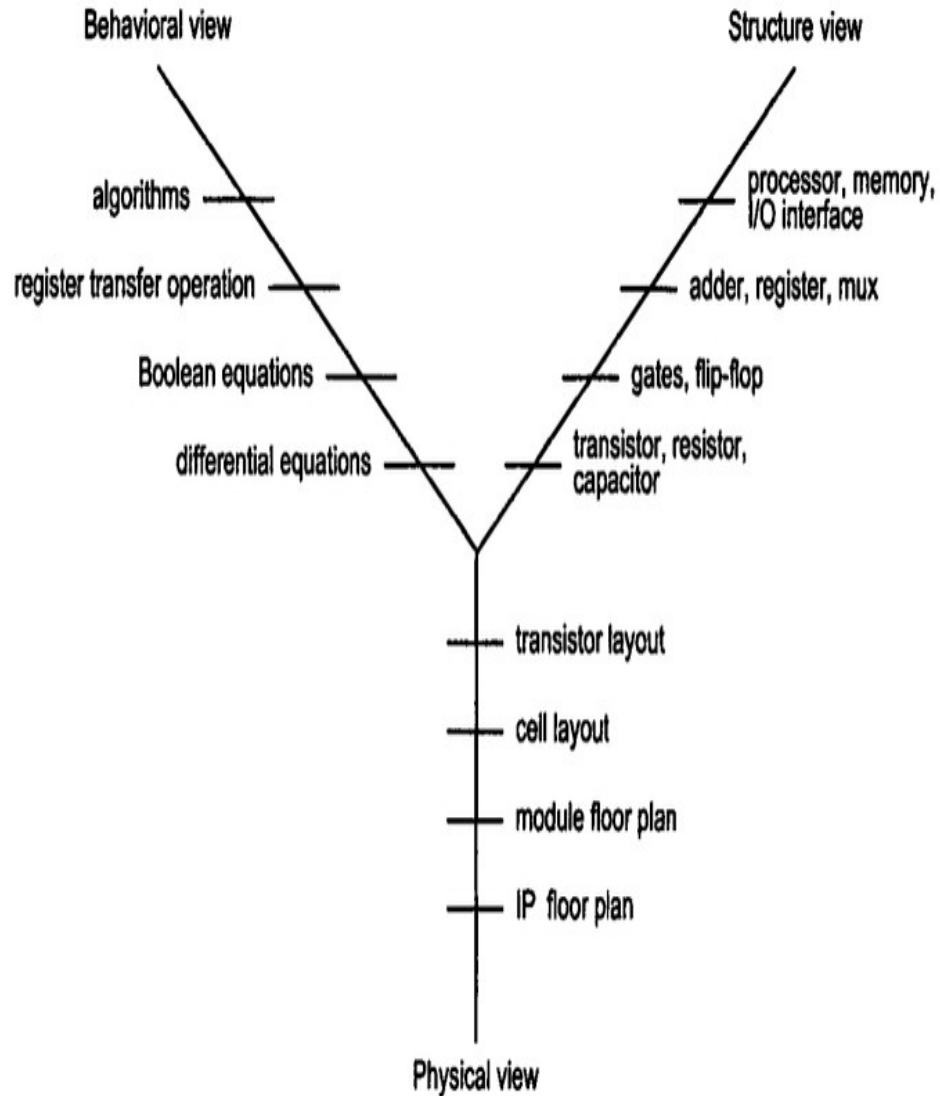


Introdução

- Tecnologia de dispositivos
 - Full-custom ASIC, Standard-cell ASIC, Gate array ASIC, Simple field-programmable logic device, Complex field-programmable logic device, Off-the-shelf small- and medium-scaled IC (SSVMSI) components
- Representação de sistemas
 - Behavioral view, Structural view, Physical view
- Níveis de abstração
 - Transistor level, Gate level, Register transfer (RT) level, Processor level
- Ferramentas de projeto (EDA)
 - Synthesis, Physical design, Verification, Testing
- Fluxo de projeto



Introdução



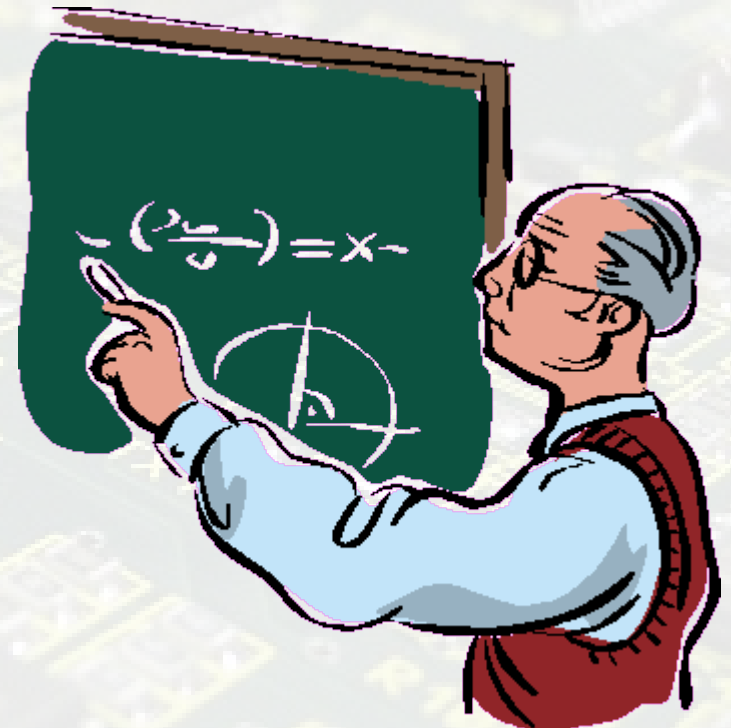


Construções Básicas do VHDL

- Unidades de projeto
 - Entity, Architecture, Configuration, Package, ...
- Elementos lexicos
 - Comments, Identifiers, Reserved words, Numbers, characters and strings
- Objetos
 - Signals, Variables, Constants, Alias,
- Tipos de dados e Operadores
 - Predefined data types in VHDL, Operators precedence, Data types in the IEEE std-logic-1164 package (std-logic and std-logic_vector, ...), Operators over an array data type (Concatenation, Aggregate, ...), Data types in the IEEE numeric_std package (Signed and unsigned, Overloaded Operators, Functions, Type conversion, ...)



Conteúdo





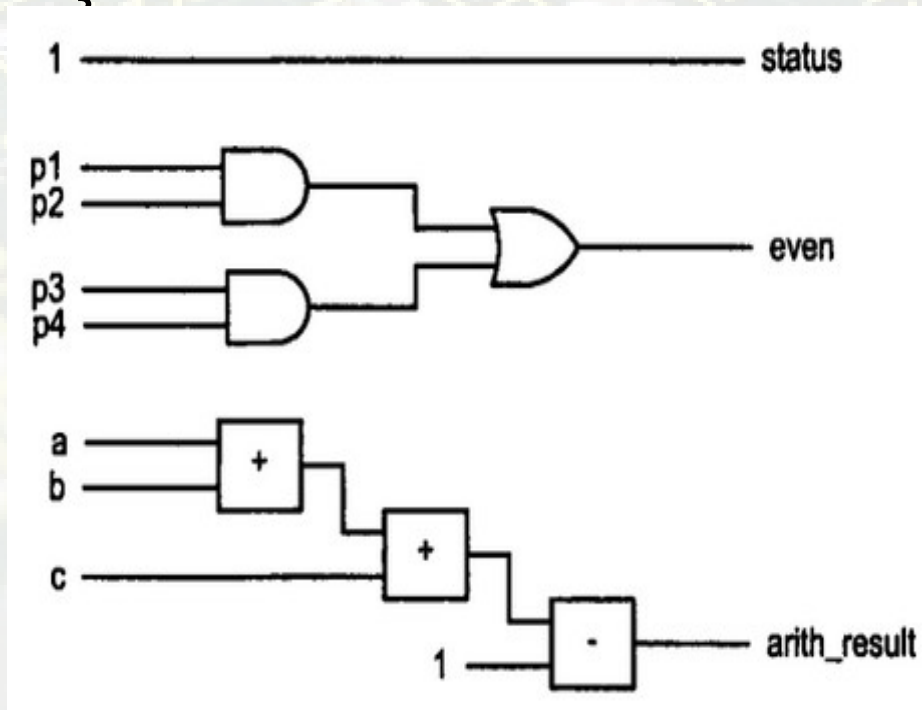
Comandos Concorrentes

- Todos os comandos contidos por “*begin*” e “*end*” da unidade *architecture* são comandos concorrentes.
- Os comandos concorrentes do VHDL são:
 - Atribuição simples;
 - Atribuição condicional;
 - Atribuição com sinal selecionado;
 - Instanciação de componentes;
 - If Generate;
 - For Generate;
 - (Process).
- Qualquer circuito combinacional pode ser especificado apenas com comandos concorrentes.



Atribuição Simples

- Sintaxe: $\langle target \rangle \leq \langle expression_same_type \rangle;$
- Exemplos: $status \leq '1';$
 $even \leq (p1 \text{ and } p2) \text{ or } (p3 \text{ and } p4);$
 $arith_out \leq a + b + c - 1;$
- Implementação conceitual:





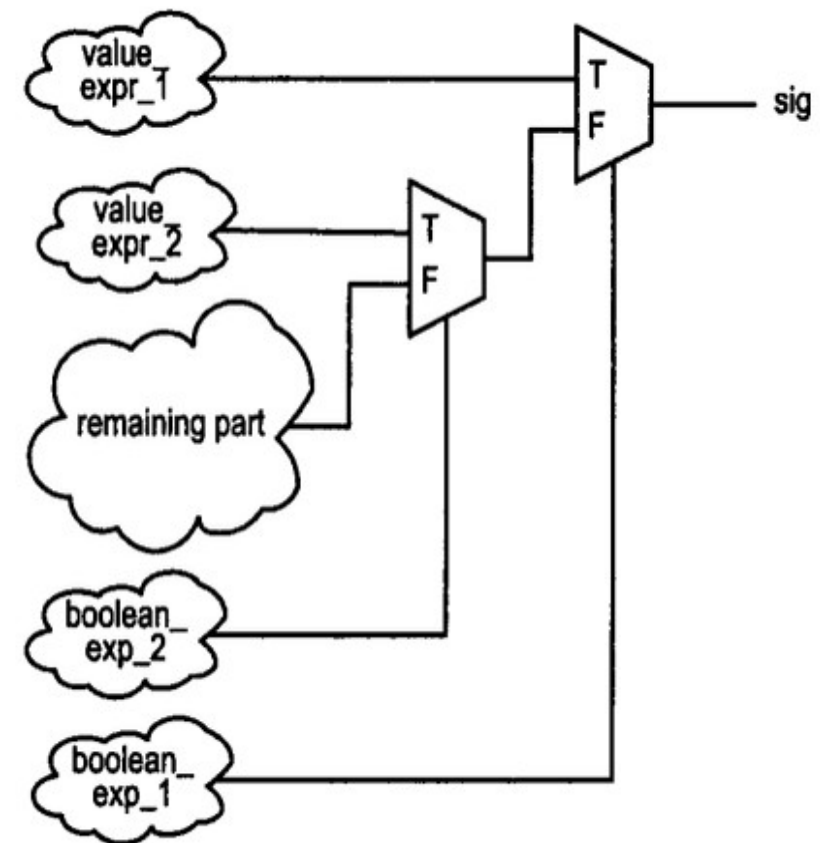
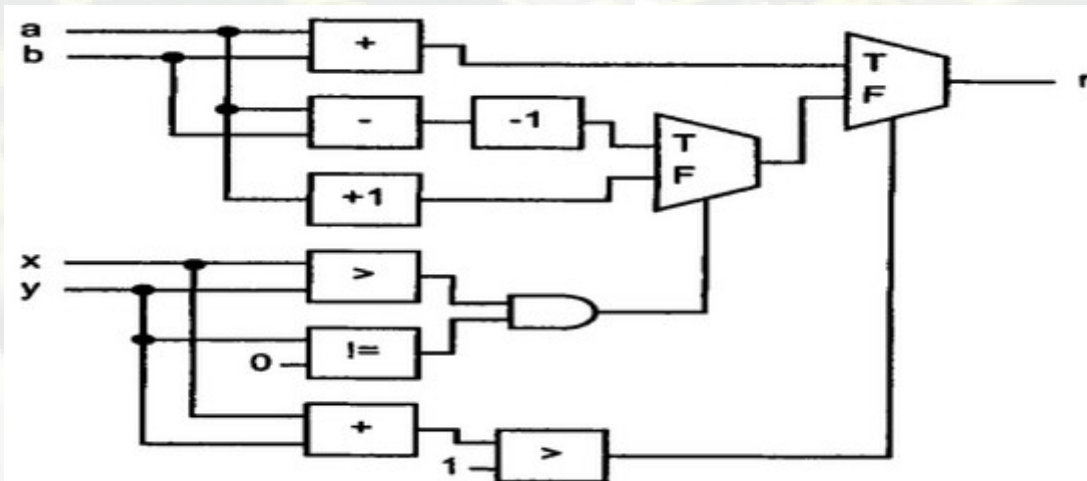


Atribuição Condicional

- **Sintaxe:** $\langle \text{optional_label} \rangle : \langle \text{target} \rangle \leq \langle \text{value} \rangle \text{ when } \langle \text{condition} \rangle \text{ else } \langle \text{value} \rangle \text{ when } \langle \text{condition} \rangle \text{ else } \dots \langle \text{value} \rangle ;$

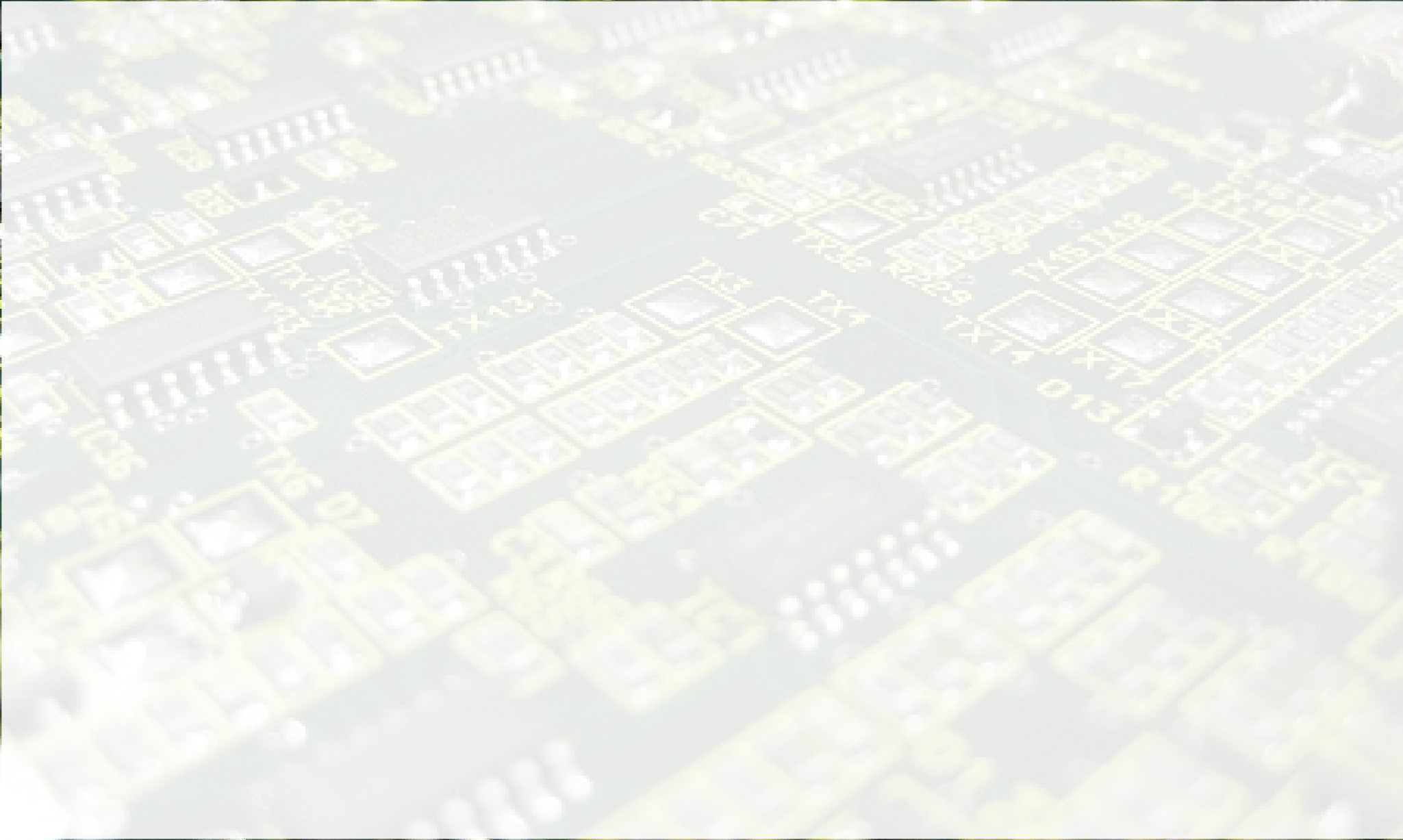
- **Exemplos:** $r \leq a+b \text{ when } x+y>1 \text{ else } a-b-1 \text{ when } x>y \text{ else } a+1;$

- **Implementação conceitual:**





Atribuição Condicional



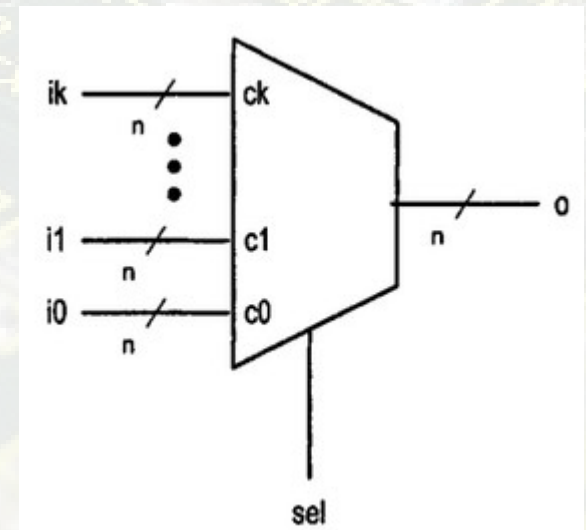
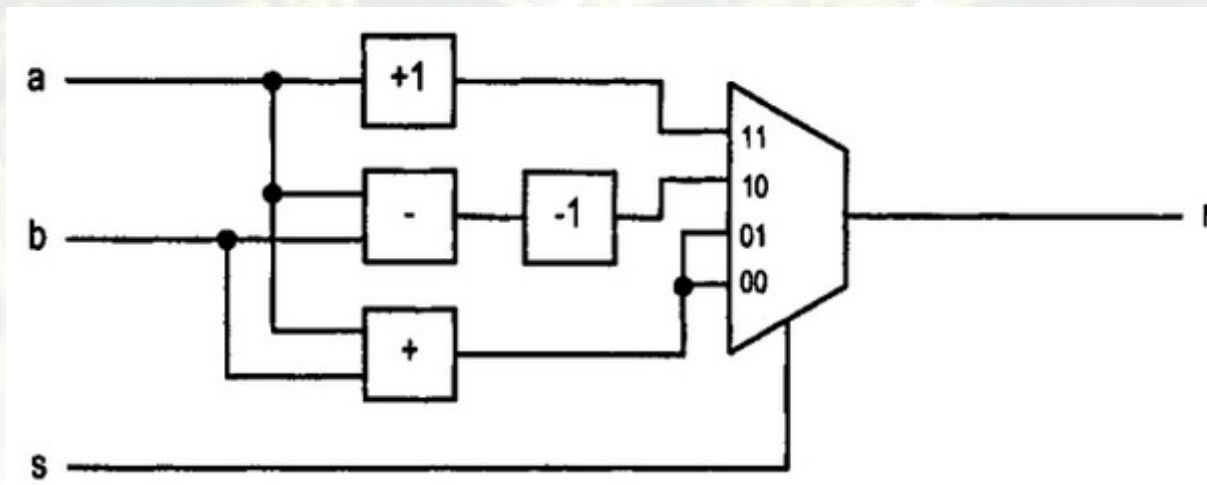


Atribuição com Sinal Selecionado

- Sintaxe: $\langle \text{optional_label} \rangle$: with $\langle \text{expression} \rangle$ select
 $\langle \text{target} \rangle \leftarrow \langle \text{value} \rangle$ when $\langle \text{choices} \rangle$,
 $\langle \text{value} \rangle$ when $\langle \text{choices} \rangle$,
 $\langle \text{value} \rangle$ when $\langle \text{choices} \rangle$,
...
 $\langle \text{value} \rangle$ when others;

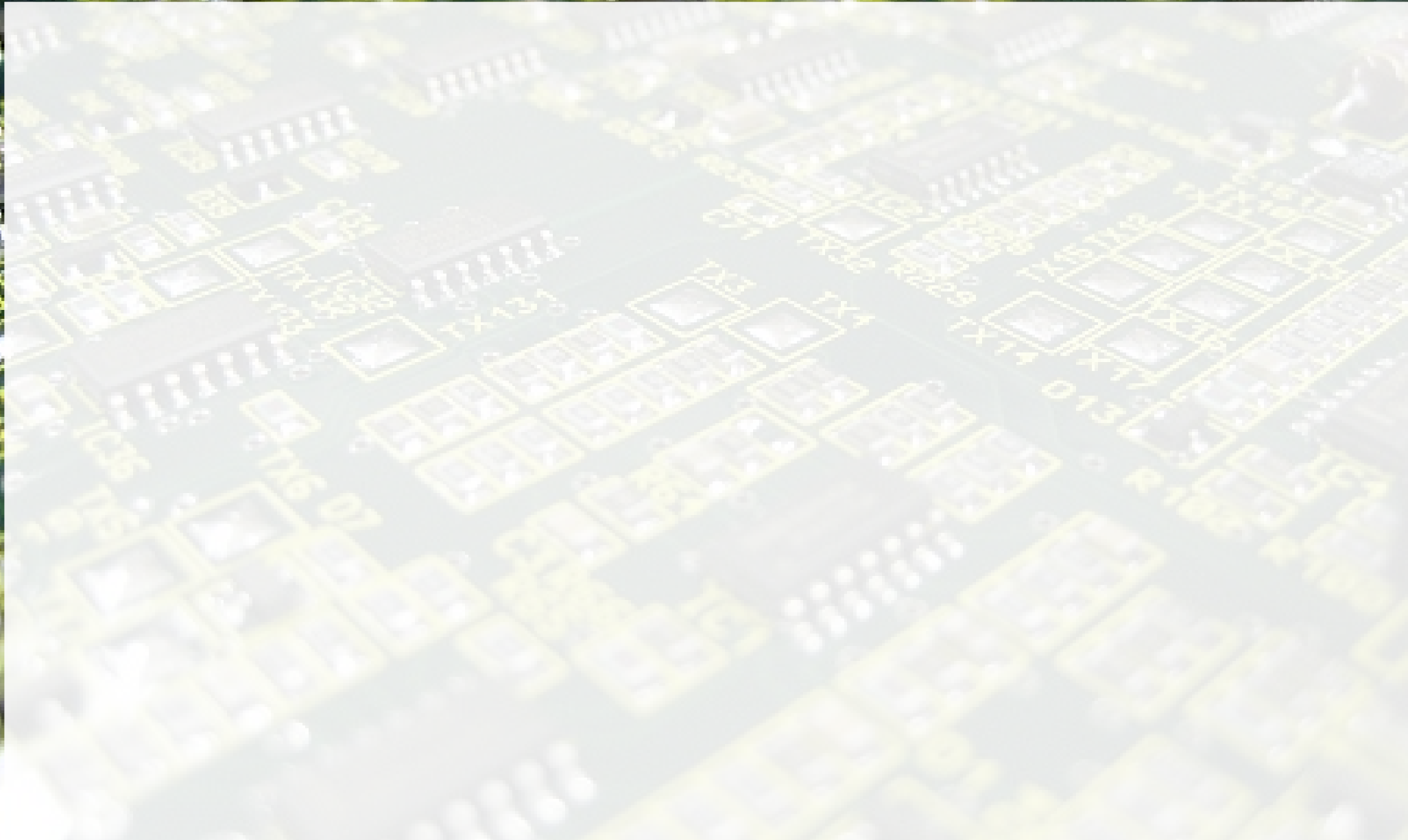
- Exemplos: with s select
 $r \leftarrow a+1$ when "11",
 $a-b-1$ when "10",
 $a+b$ when others;

- Implementação conceitual:





Atribuição com Sinal Selecionado





Comparação

- Atribuição condicional:
 - A ordem das condições é relevante, pois a primeira que for verdadeira especifica a atribuição que será efetuada. Portanto, pode haver mais de uma condição verdadeira;
 - Implementação conceitual é uma cascata de multiplexadores.
- Atribuição com sinal selecionado:
 - A ordem das condições não importa; Apenas uma única condição pode ser verdadeira;
 - Implementação conceitual é um único multiplexador.



Instanciação de Componentes

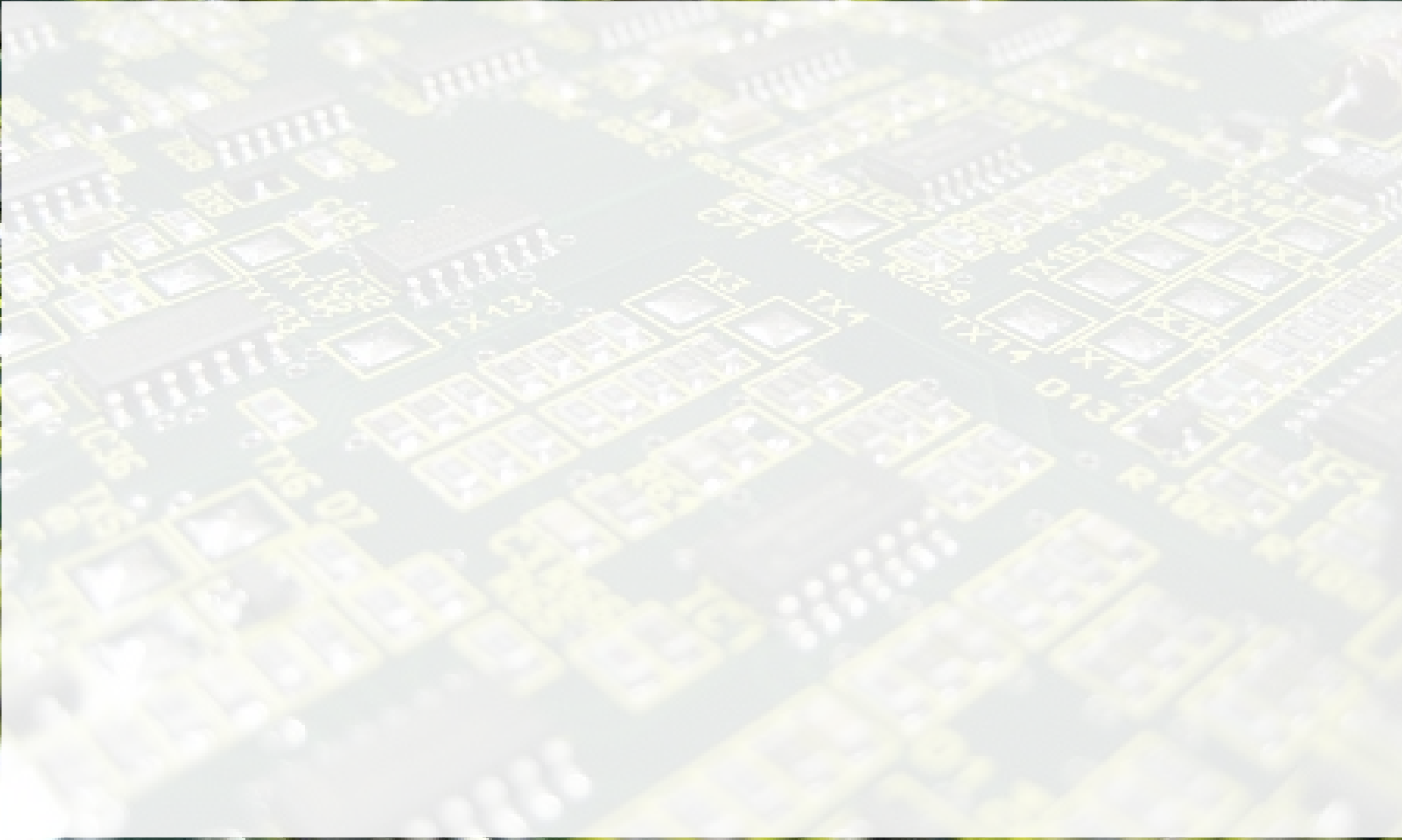
- **Sintaxe:**

```
<instance_name> : <component_name>  
    generic map ( <name> => <value>, ... )  
    port map ( <formal_input> => <expression>, <formal_output>  
=> <signal>, <formal_inout> => <signal>, ... );
```
- **Exemplos:**

```
architecture ...  
    component adderNbits is  
        generic (N: positive);  
        port (cin: in std_logic, a, b: in std_logic_vector(N-1 ....  
        ...  
        signal cinreal: std_logic;  
        signal areal, breal: ...  
  
begin  
  
    adder01: adderNbits generic map (N => 8) port map  
    (cin=>cinreal, a=>areal, b=>breal, cout=>coutreal);  
  
end architecture;
```



Instanciação de Componentes





If Generate

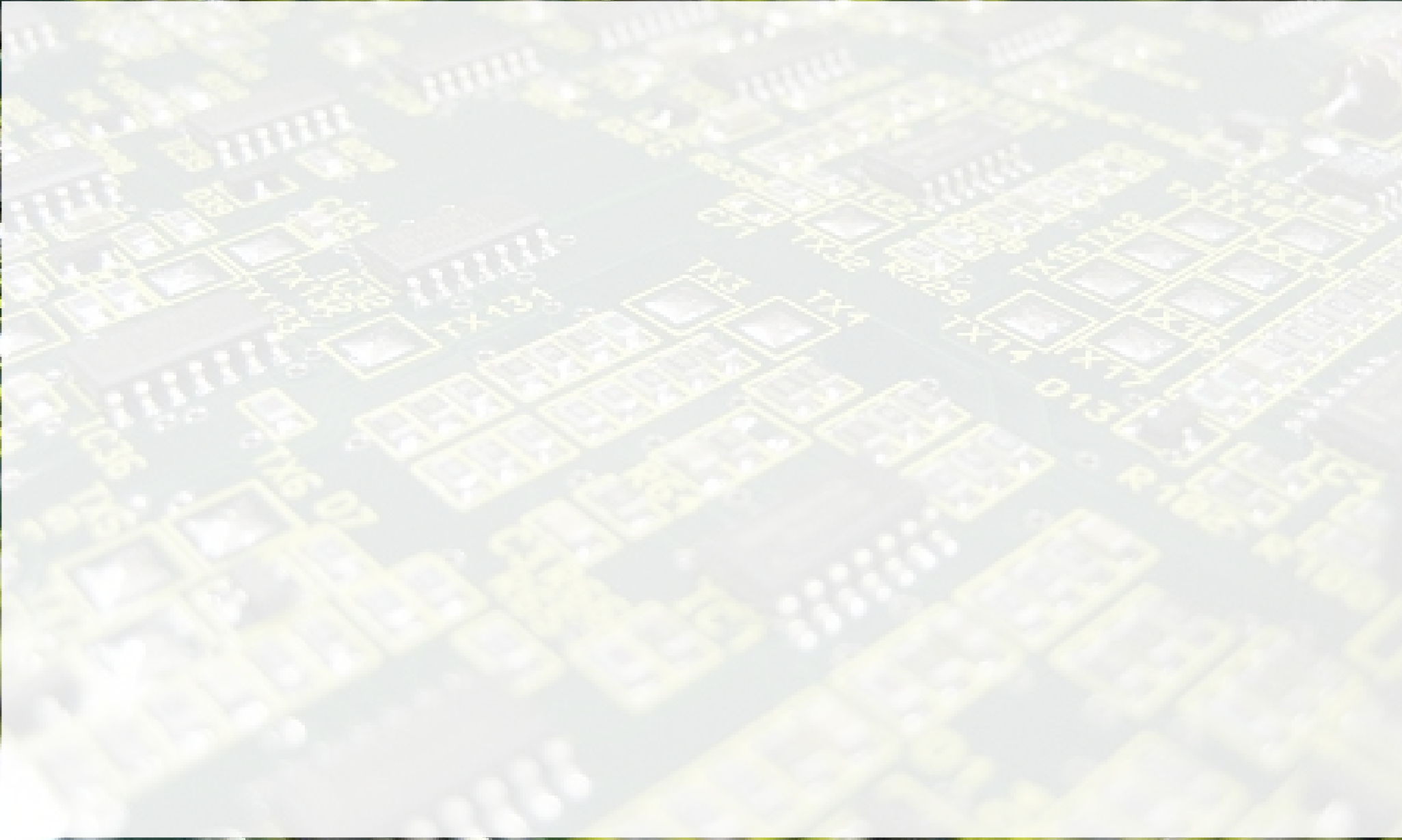
- **Sintaxe:**

```
<generate_label>:  
  if <condition> generate  
    -- Concurrent Statement(s)  
  end generate;
```
- **Exemplos:**

```
entity ...  
  generic(inverted: boolean);  
  ...  
  
architecture ...  
  signal btemp: ....  
begin  
  
  invert_operand_if_set: if inverted generate  
    btemp <= not b;  
  end generate;  
  
  same_operand_if_not_set: if not inverted generate  
    btemp <= b;  
  end generate;
```



If Generate





For Generate

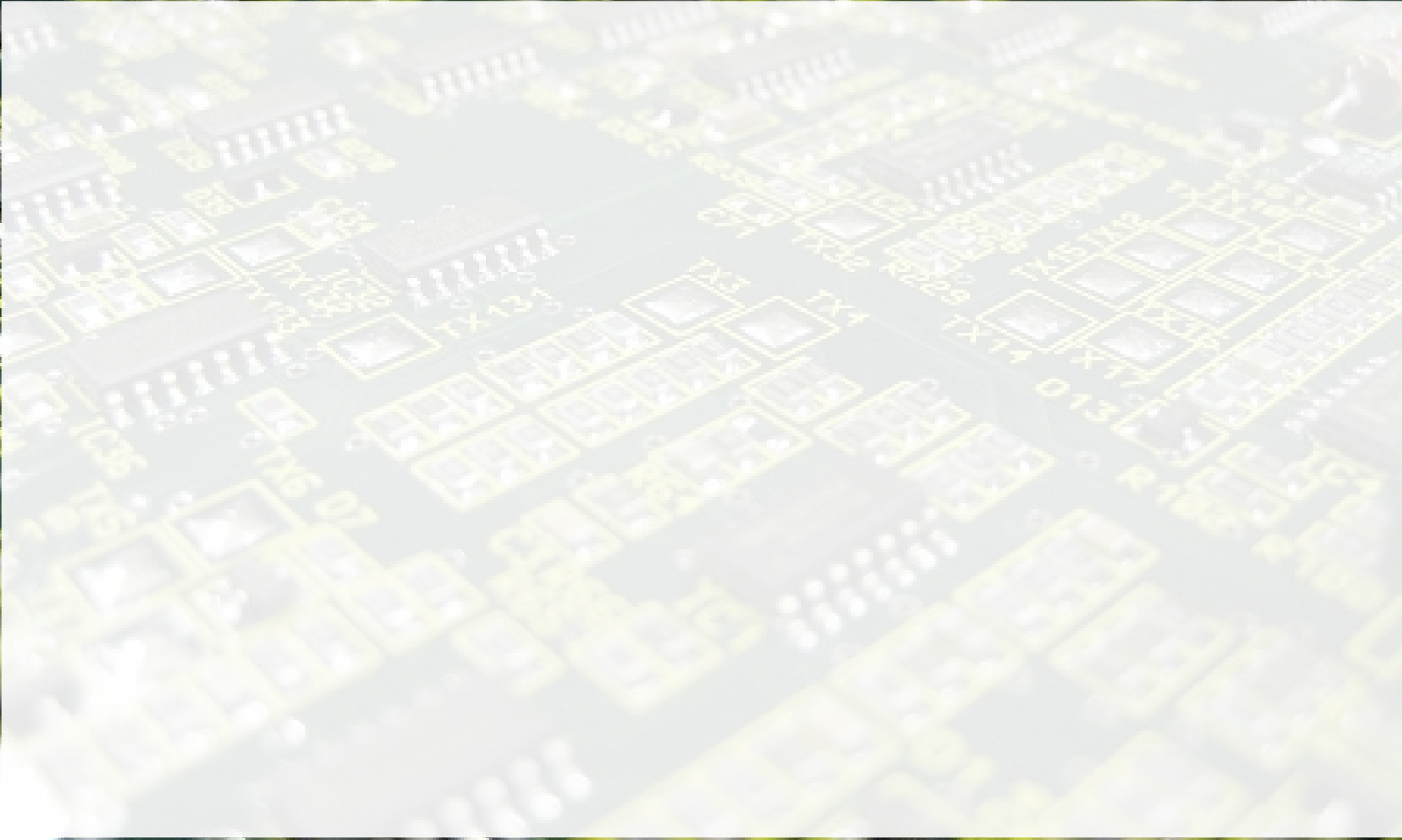
- **Sintaxe:**

```
<generate_label>:  
  for <loop_id> in <range> generate  
    -- Concurrent Statement(s)  
  end generate;
```
- **Exemplos:**

```
generate_adders: for i in 0 to N-1 generate  
  adderi: fulladder  
    port map (cin(i), a(i), b(i), sum(i), cout(i));  
end generate;
```



For Generate





Process

- Sintaxe:

```
<optional_label>:  
  process(<sensitivity_list>) is  
    -- Declaration(s)  
  begin  
    -- Sequential Statement(s)  
  end process;
```

- Exemplos:

```
my_decoder: process(inp) is  
  begin  
    outp <= (others=>'0');  
    for i in inp'range loop  
      if inp(i)='1' then  
        outp <= std_logic_vector(to_unsigned(i,N));  
      end if;  
    end loop;  
  end process;
```



Process



Referências Bibliográficas

- Vahid, Frank. Sistemas Digitais: projeto, otimização e HDLs. Porto Alegre: Bookman, 2008. ISBN 978-85-7780-190-9
- Chu, Pomg P. RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability. Wiley-Interscience, 2006.
- Pedroni, Volnei. Circuit Design with VHDL. The MIT Press, 3th edition, 2020.

