



# LABORATÓRIO DE CIRCUITOS DIGITAIS

## SIMULAÇÃO E SÍNTESE VHDL

TIAGO BARROS

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN

CENTRO DE TECNOLOGIA – CT

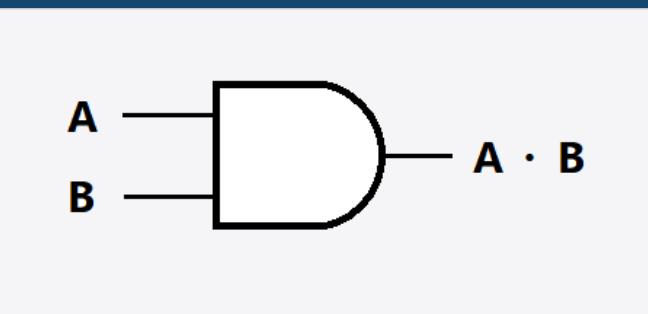
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO – DCA

# BIBLIOGRAFIA

- D'AMORE, Roberto; VHDL – Descrição e Síntese de Circuitos Digitais. LTC, 2005.
  - Capítulos 1 e 2

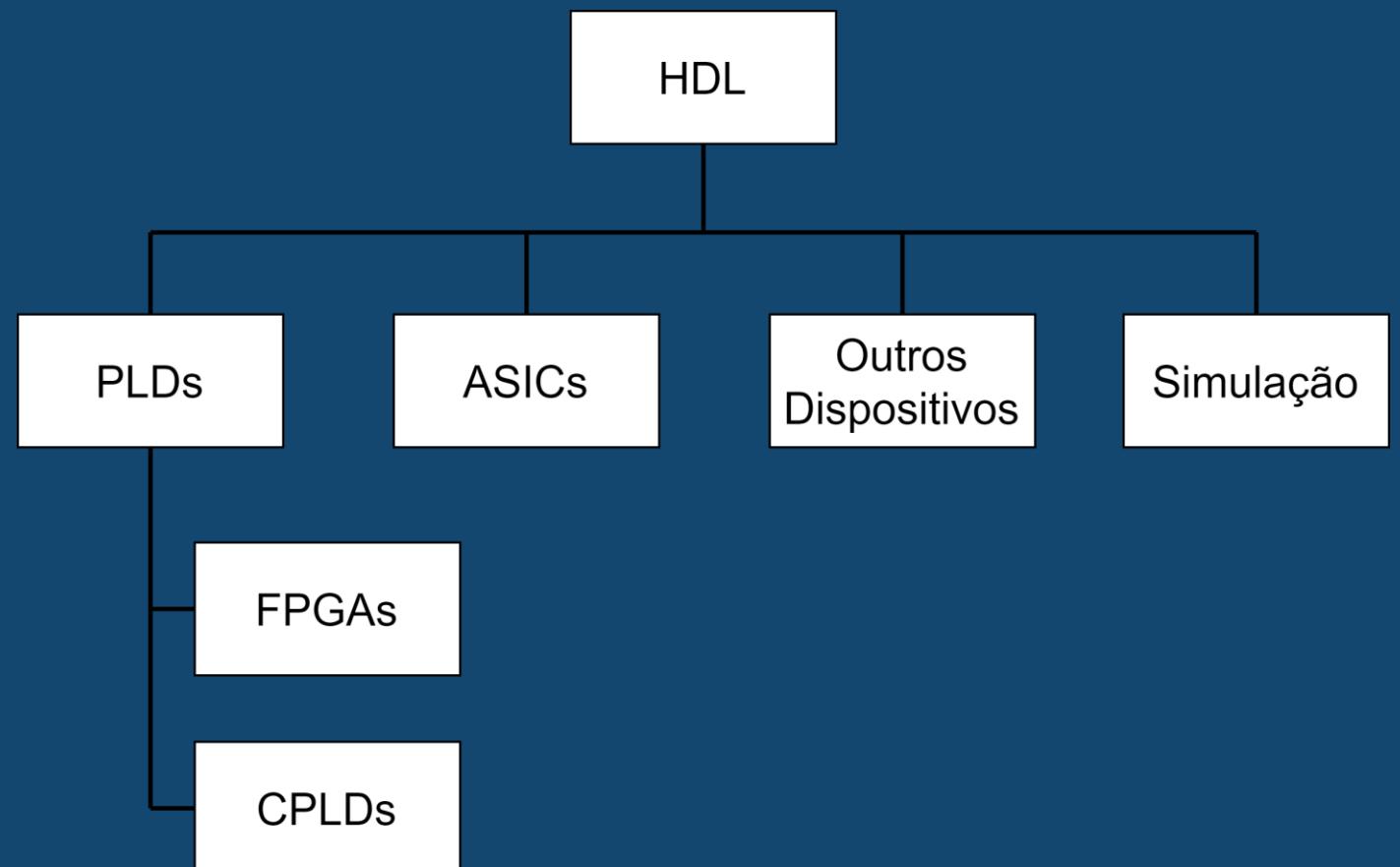
# INTRODUÇÃO

# HDL – HARDWARE DESCRIPTION LANGUAGE



```
1  entity and_gate is
2    port(
3      a, b :  in bit;
4      x     :  out bit
5    );
6  end and_gate;
7
8  architecture main of and_gate is
9
10 begin
11   x <= a and b;
12
13 end architecture main;
```

# APLICAÇÃO



# VHDL É UMA HDL

**VHSIC**

- Very High Speed Integrated Circuit

**Hardware**

**Description**

**Language**

# VERSÕES

IEEE 1076-1987

IEEE 1076-1993

IEEE 1076 2000 Edition

IEEE 1076-2002

IEEE 1076-2008

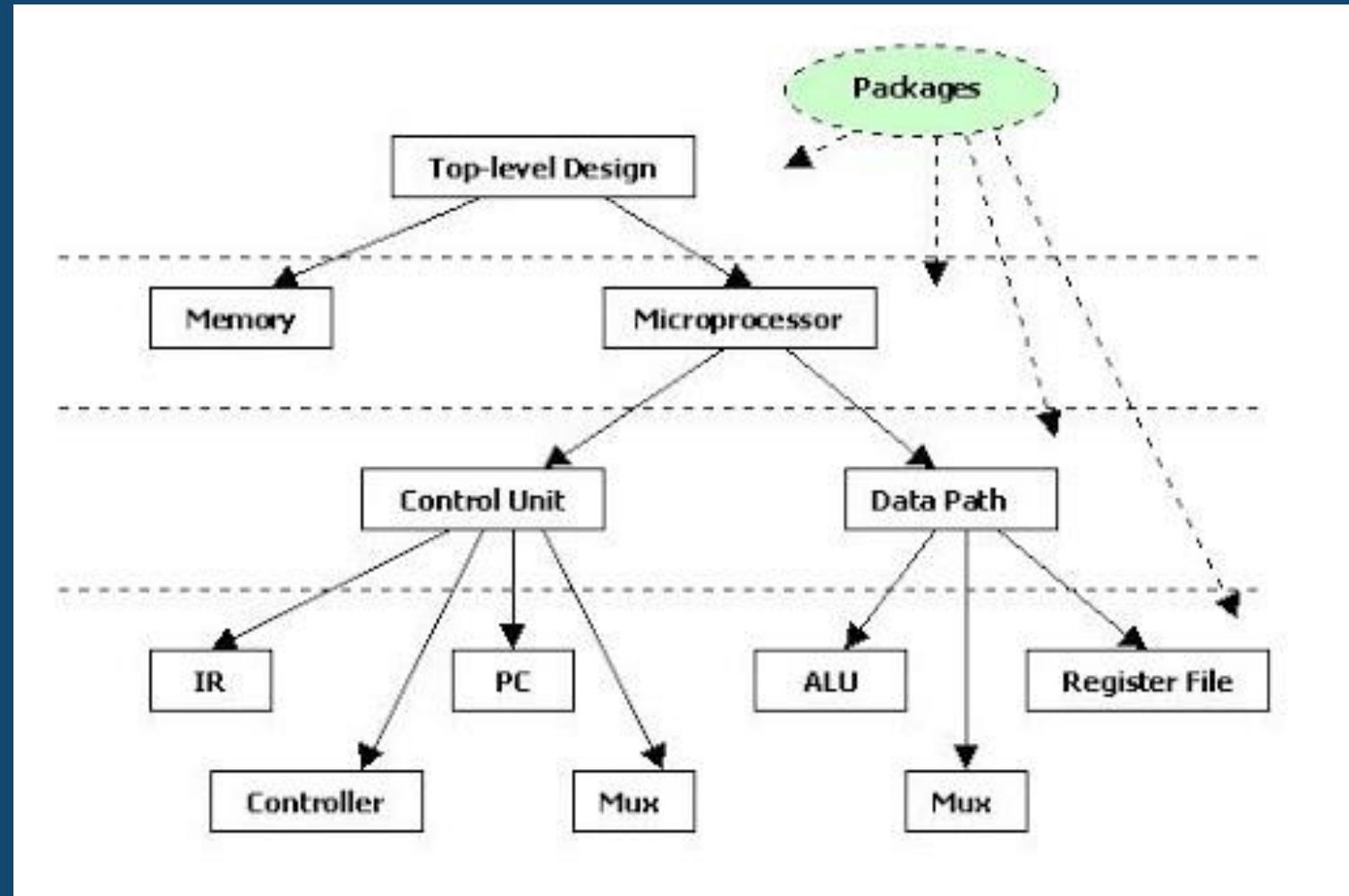
# PACOTES

- IEEE 1164
  - std\_logic\_1164
  - std\_logic\_vector
- IEEE 1076.3
  - numeric\_bit
  - numeric\_std

# ASPECTOS GERAIS DA LINGUAGEM

1. MÚLTIPLOS NÍVEIS DE  
HIERARQUIA:

TOP-DOWN DESIGN



# ASPECTOS GERAIS DA LINGUAGEM

## 2. ESTILO DE UMA DESCRIÇÃO

### COMPORTAMENTAL

```
1  if a = b then
2  |  equals <= '1';
3  else
4  |  equals <= '0';
5  end if;
```

### EXPRESSÕES LÓGICAS

```
1  equals <= '1' when (a = b) else '0';
```

### REDE DE LIGAÇÕES (ESTRUTURAL)

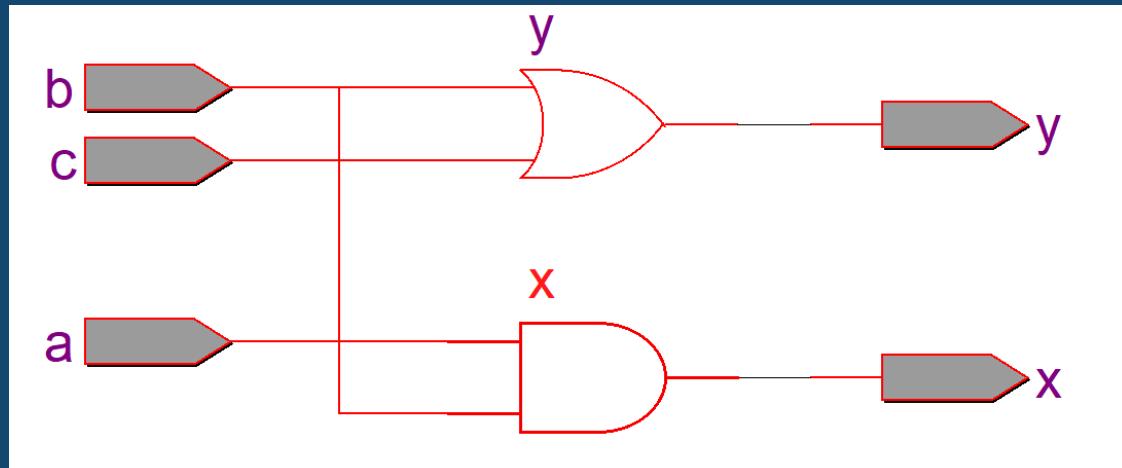
```
1  U0: xnor port map (a(0), b(0), x(0));
2  U1: xnor port map (a(1), b(1), x(1));
3  U2: xnor port map (a(2), b(2), x(2));
4  U3: xnor port map (a(3), b(3), x(3));
5  U4: and4 port map (x(0), x(1), x(2), x(3), equals);
```

### MISTO (MESCLADO)

# ASPECTOS GERAIS DA LINGUAGEM

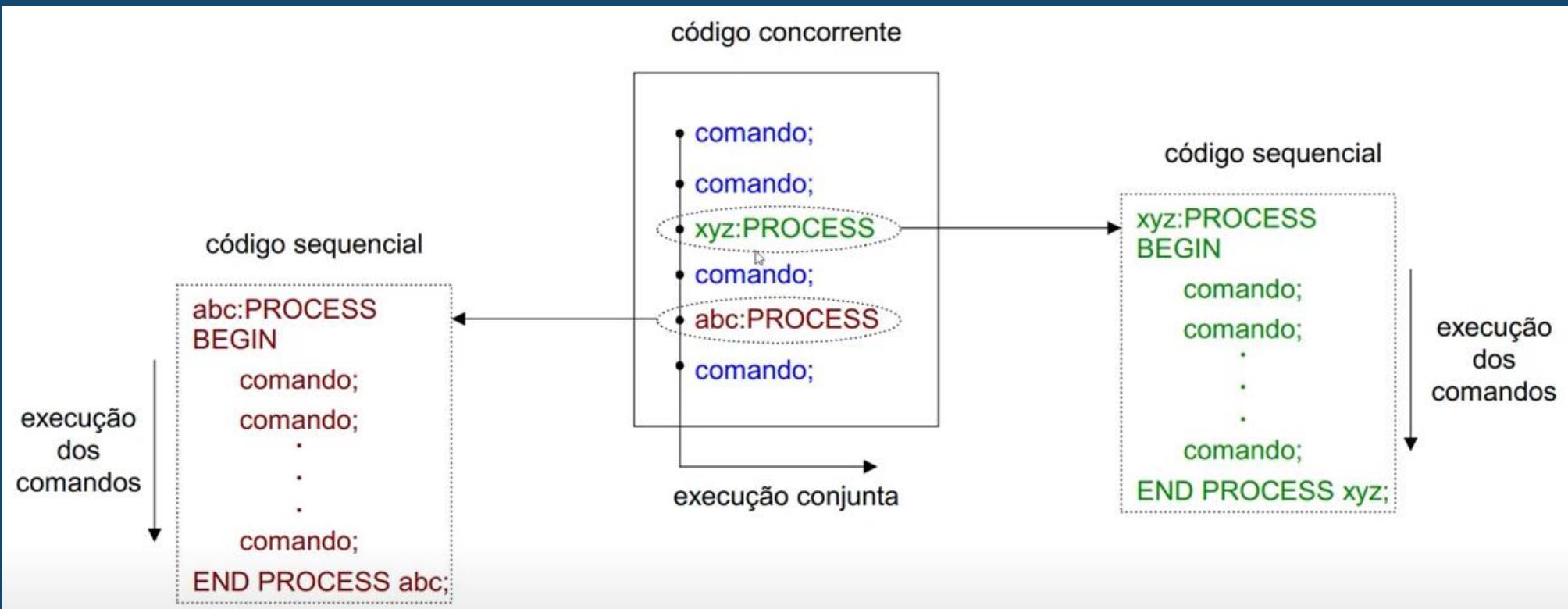
## 3. LINGUAGEM CONCORRENTE

```
1 entity gates is
2     port(
3         a, b, c :  in bit;
4         x, y      :  out bit
5     );
6 end gates;
7
8 architecture main of gates is
9
10 begin
11
12     x <= a and b;
13     y <= c or b;
14
15 end architecture main;
```



# ASPECTOS GERAIS DA LINGUAGEM

## 4. COMANDOS SEQUENCIAIS



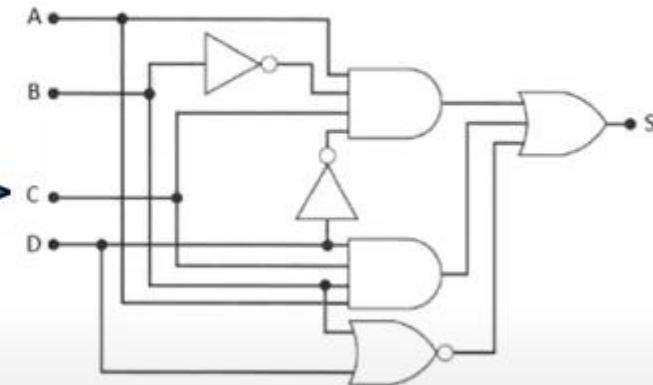
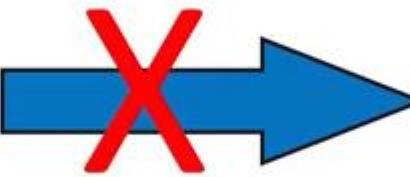
# SÍNTESIS DE CIRCUITOS

# NEM TODOS OS CÓDIGOS VHDL PODEM SER SINTETIZADOS

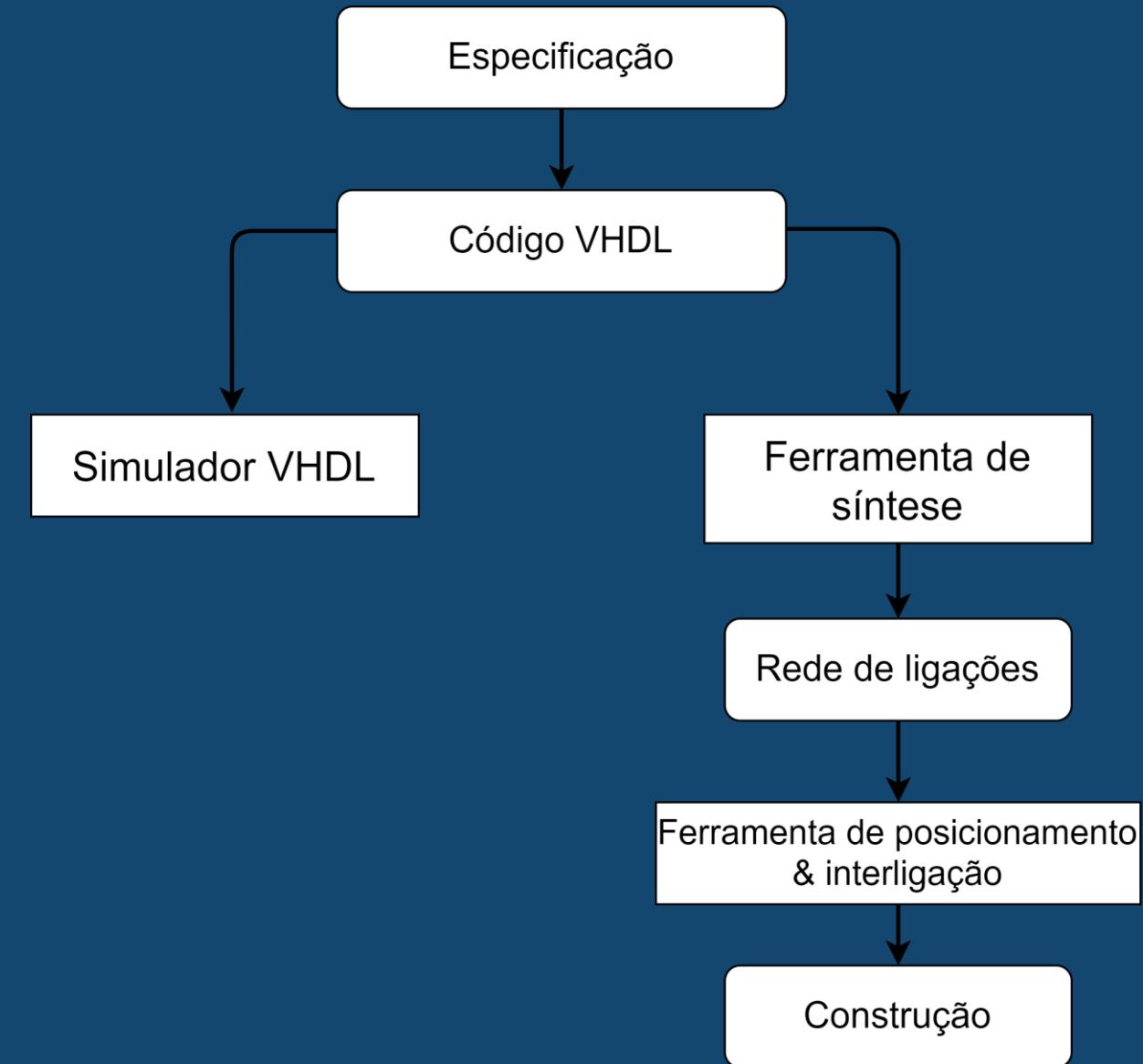
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity periph is

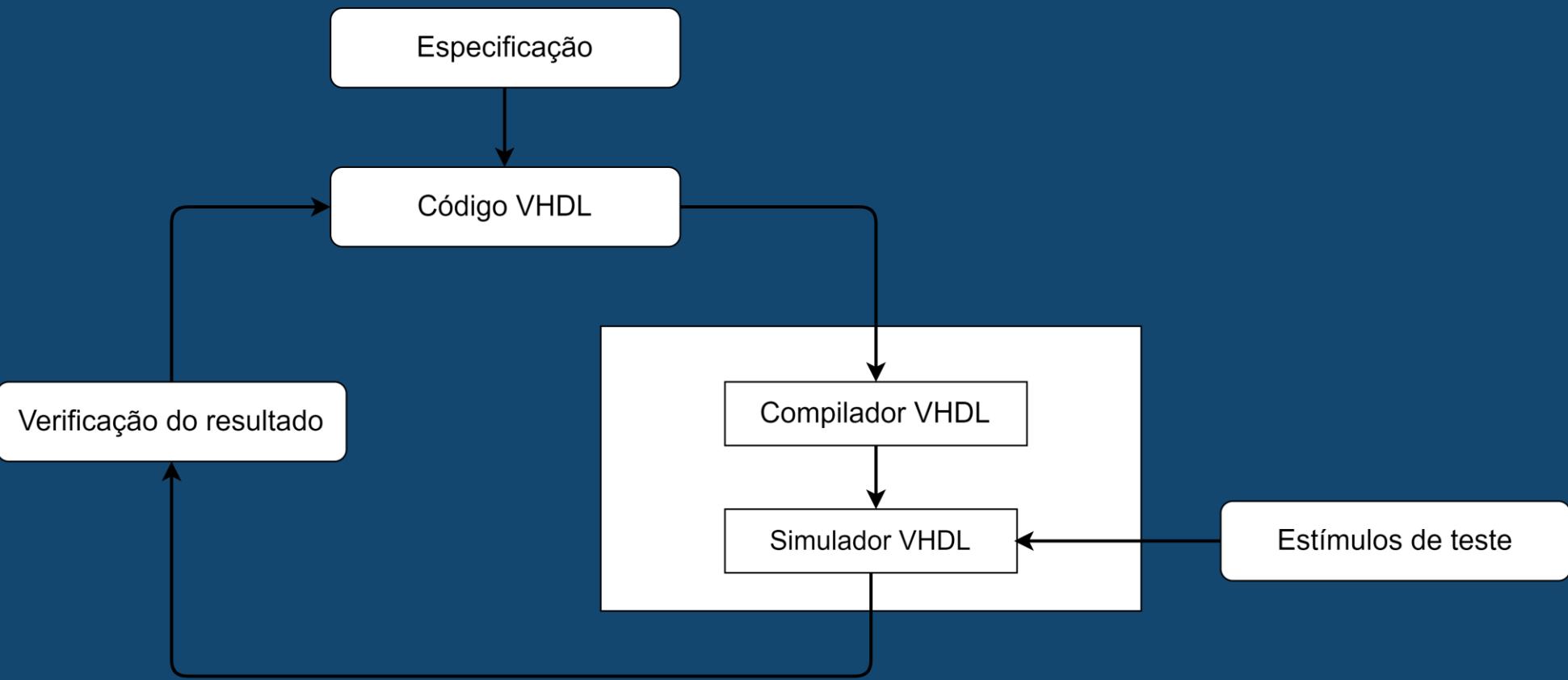
    generic
    (
        DATA_WIDTH : natural := 4;
        ADDR_WIDTH : natural := 3
    );
    port
    (
        addr      : in  std_logic_vector(ADDR_WIDTH-1 downto 0);
        data       : in  std_logic_vector(DATA_WIDTH-1 downto 0);
        we         : in  std_logic;
        q          : out std_logic_vector(DATA_WIDTH-1 downto 0);
        enable     : in  std_logic;
        leds1, leds0 : out std_logic_vector(3 downto 0);
        sw1, sw0   : in  std_logic_vector(3 downto 0)
    );
end entity;
```



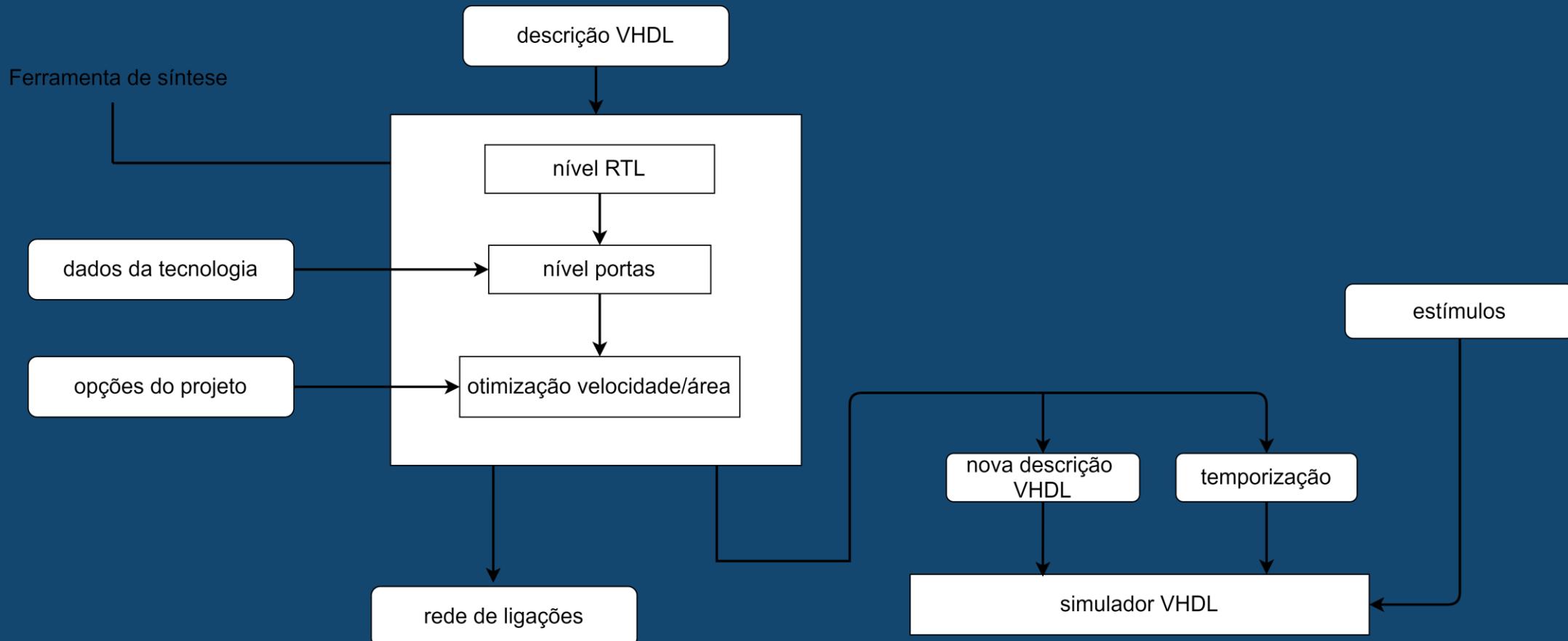
# ETAPAS DE PROJETO



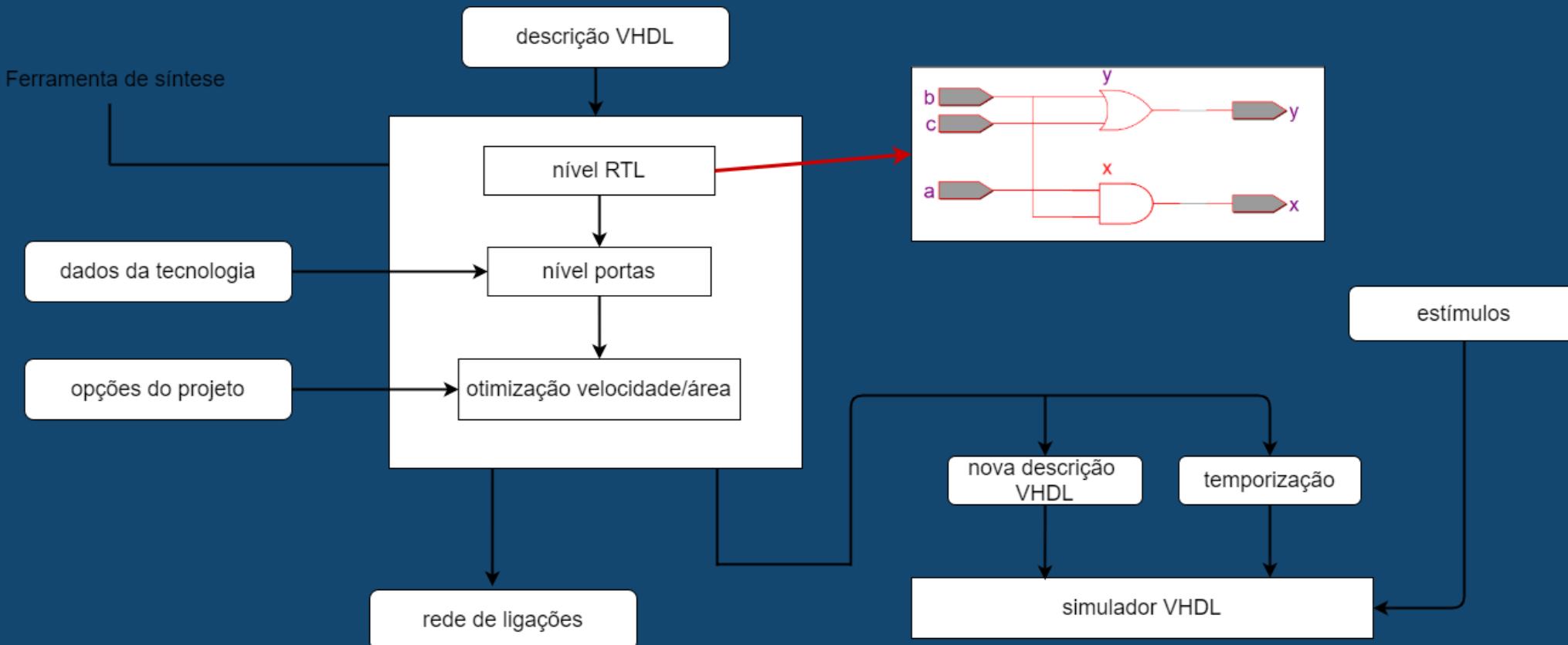
# ELABORAÇÃO DA DESCRIÇÃO VHDL



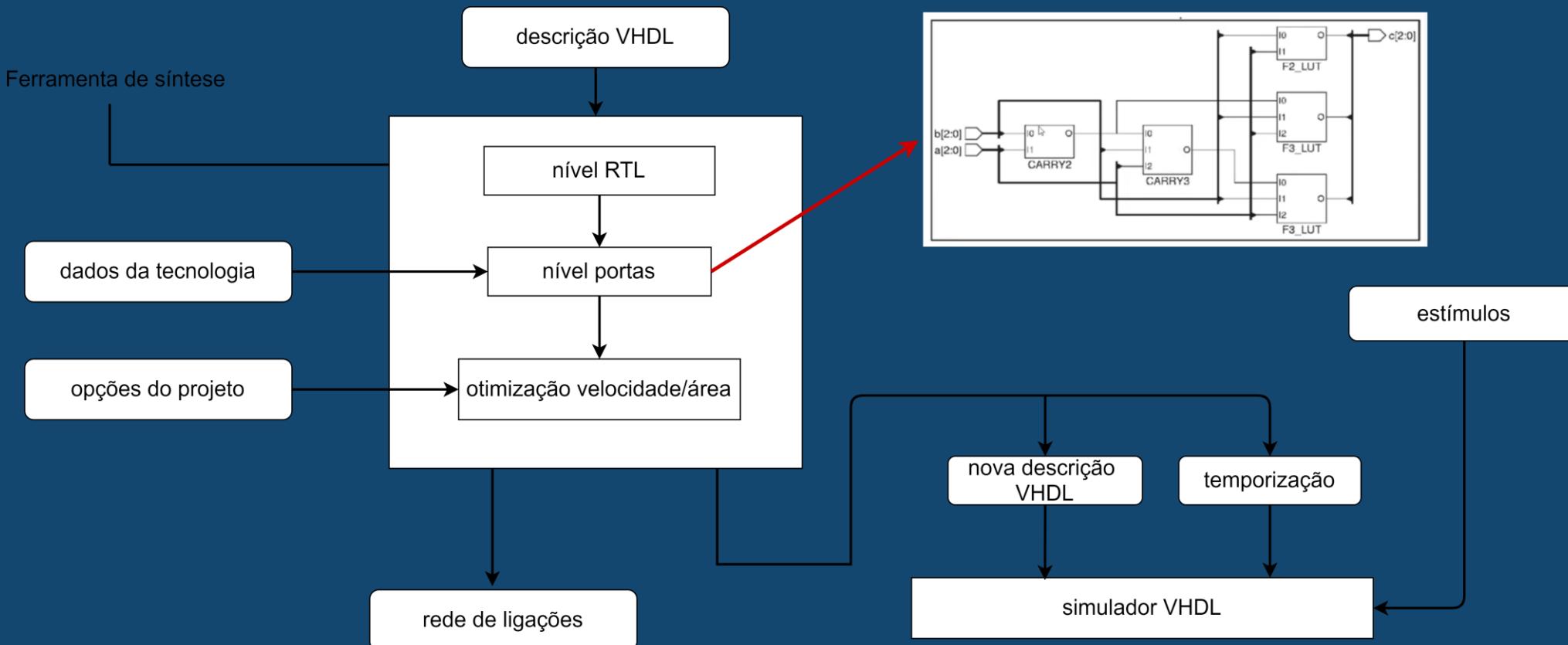
# SÍNTESE DA DESCRIÇÃO VHDL



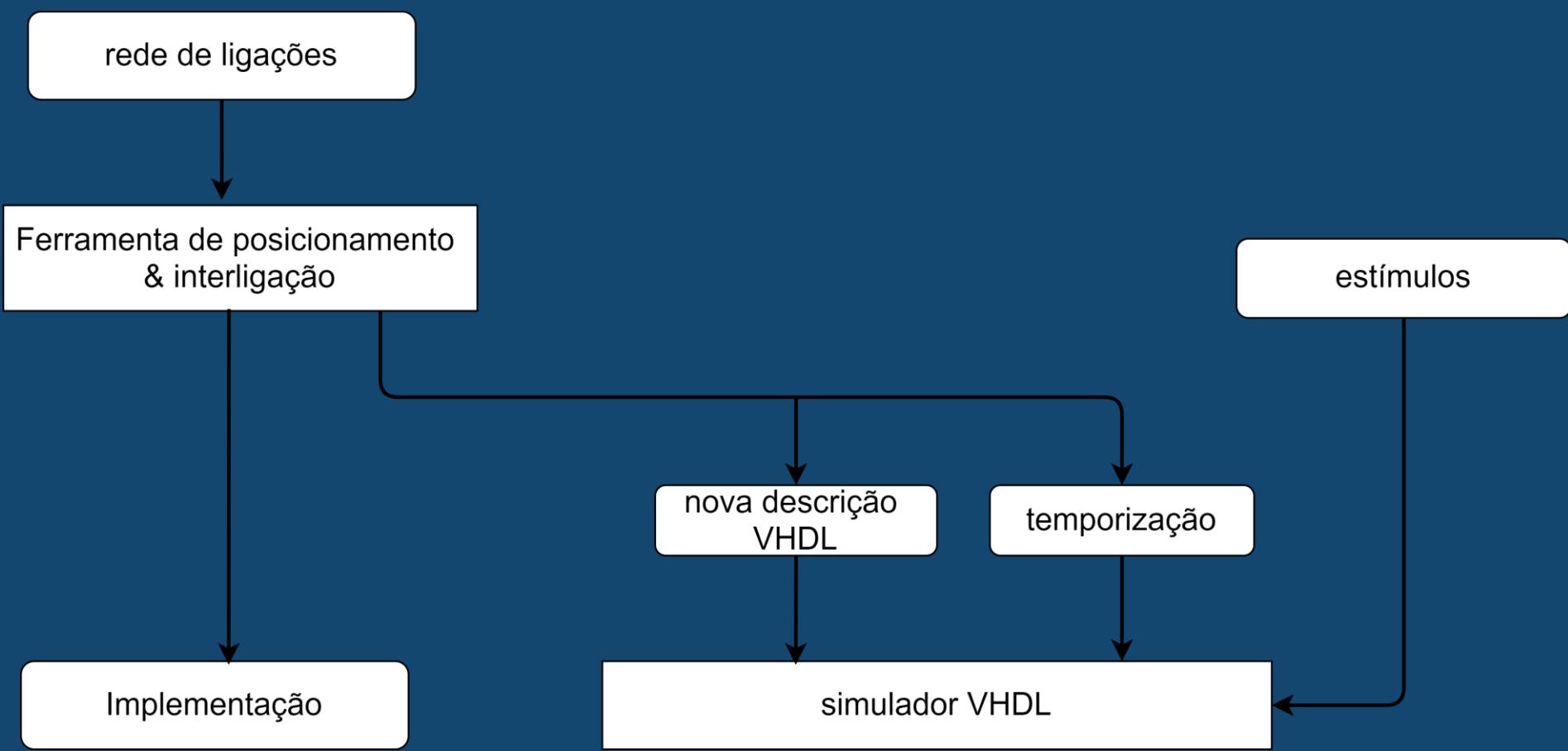
# SÍNTESE DA DESCRIÇÃO VHDL



# SÍNTESE DA DESCRIÇÃO VHDL

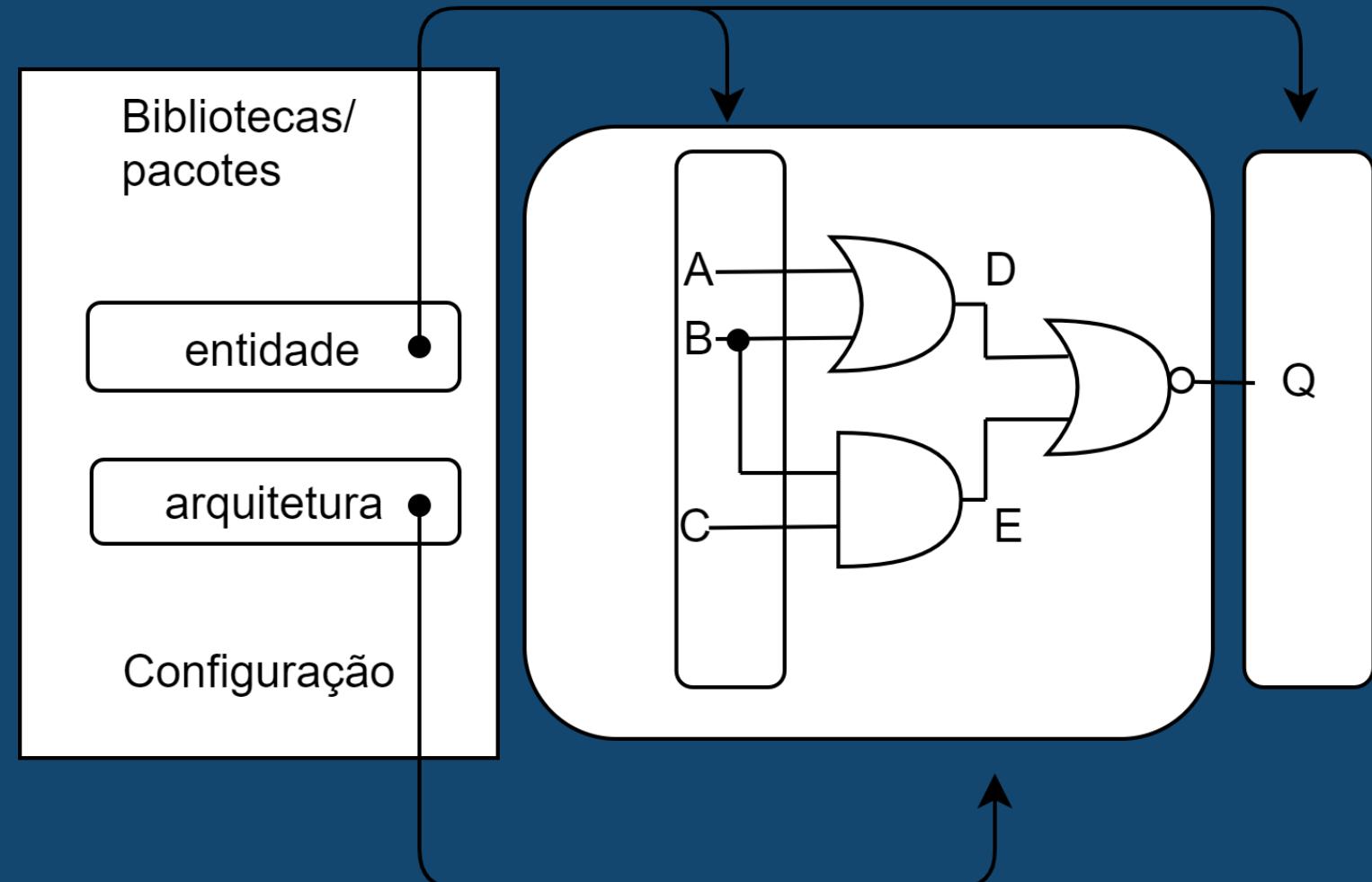


# ETAPA FINAL DO PROJETO (PLACE AND ROUTE)

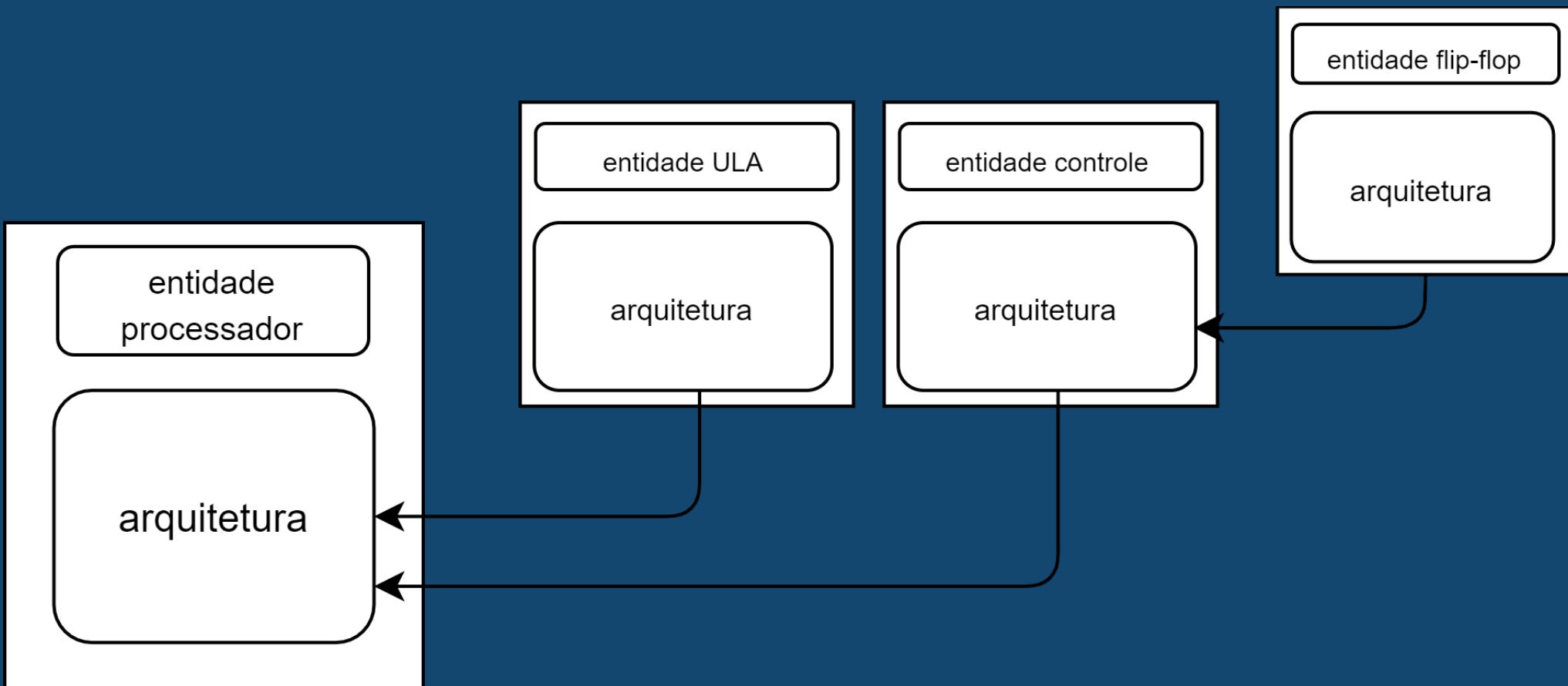


# ESTRUTURA DE CÓDIGO VHDL

# ENTIDADE DE PROJETO



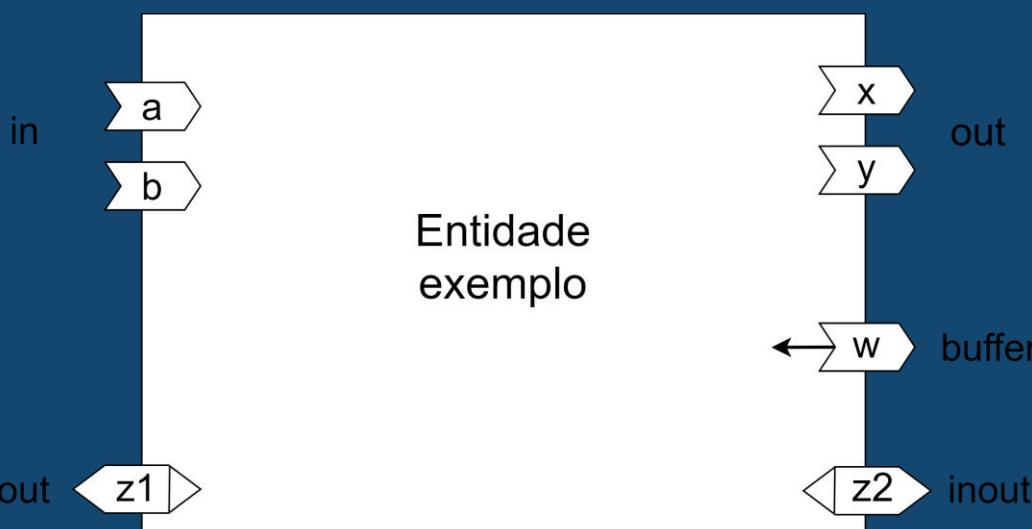
# HIERARQUIA



# EXEMPLOS: DECLARAÇÃO DA ENTIDADE E CORPO DA ARQUITETURA

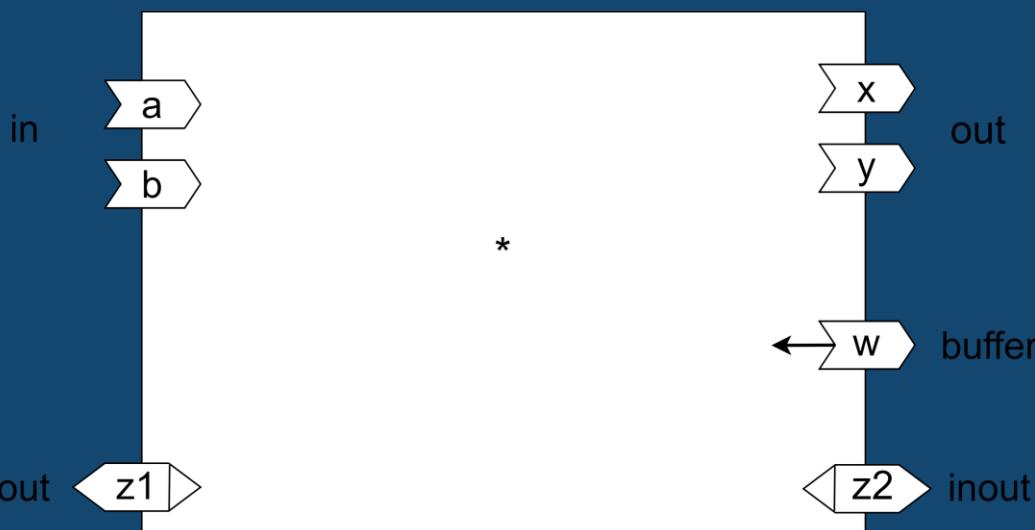
```
1  -- Porta and
2
3  -- Entidade
4  entity and_gate is
5    port(
6      a, b :  in bit;
7      z    :  out bit
8    );
9  end and_gate;
10
11 -- Arquitetura
12 architecture main of and_gate is
13
14 begin
15
16   z <= a and b;
17
18 end architecture main;
```

# ENTIDADE



```
1  entity entidade_exemplo is
2      generic(
3          n : tipo := valor
4      );
5      port(
6          a, b      : in     tipo_1;
7          x, y      : out    tipo_2;
8          w         : buffer tipo_3;
9          z1, z2 : inout   tipo_4
10     );
11 end entidade_exemplo;
```

# ARQUITETURA



```
1  architecture name_arc of entidade_exemplo is
2      -- Declarações:
3          -- Sinais
4          -- Constantes
5          -- Componentes
6          -- Subprogramas
7          -- Novos tipos
8          -- outros
9      begin
10         -- Comandos concorrentes
11     end architecture name_arc; --main
```