

# Linguagem de Descrição de Hardware

SICO5A – Sistemas Digitais

Curso: Engenharia Elétrica

Professor: Layhon Santos  
[layhonsantos@utfpr.edu.br](mailto:layhonsantos@utfpr.edu.br)

# Introdução

Inserção de projeto:

- ✓ Em eletrônica, uma linguagem de descrição de hardware ou LDH é qualquer linguagem de uma classe de linguagens de computador, linguagem de especificação ou linguagem de modelagem para uma descrição formal e design de circuitos eletrônicos, e mais comumente, a lógica digital.
- ✓ Pode descrever o funcionamento do circuito, a sua concepção e organização, e ainda testá-lo para verificar seu funcionamento por meio de simulação.
- ✓ LDHs são padrões de expressões baseados em texto, da estrutura espacial, temporal e comportamental dos sistemas eletrônicos. LDHs incluem anotações explícitas para expressar a simultaneidade bem como sintaxe e semântica próprias.
- ✓ LDHs também incluem uma noção implícita de tempo, como um atributo primário de hardware.

# Introdução

A primeira parte da Aula consiste em um resenha sobre o Tema HDL e VHDL.

# Introdução

Inserção de projeto:

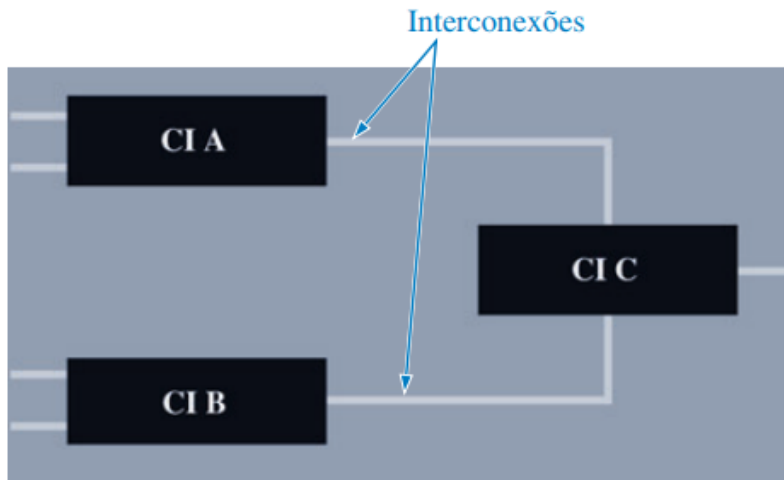
- ✓ Inserção de baseada em texto é realizada com uma linguagem de descrição de hardware (HDL - *hardware description language*), tal como VHDL, tal como VHDL, Verilog, AHDL ou ABEL.
- ✓ inserção gráfica (esquemático) permite que funções lógicas pré-armazenadas sejam selecionadas a partir de um biblioteca, apresentada na tela e então interconectadas para criar o projeto lógico;
- ✓ A inserção por diagrama de estados requer a especificação dos estados pelos quais um circuito lógico sequencial passa e as condições que provocam a mudança de cada estado.
- ✓ **A finalidade da descrição de um circuito lógico usando um programa VHDL é que ele possa ser programado num PLD.**

# Teste

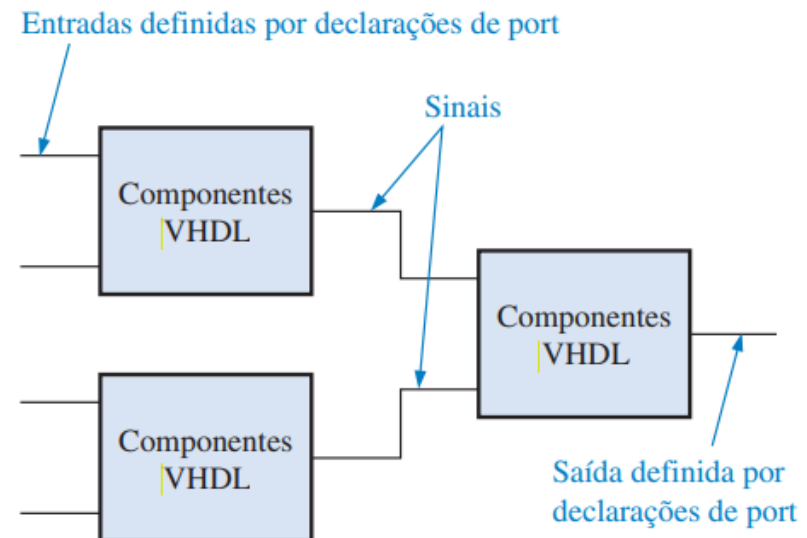
✓ O que é o VHDL?

- (a) dispositivo lógico
- (b) linguagem de programação de PLD
- (c) linguagem de computador
- (d) lógica de densidade muito alta.

# Estrutural Programação VHDL



(a) Implementação de hardware com lógica de funções fixas



(b) Implementação estrutural VHDL

O componente VHDL é uma forma de predefinir uma função lógica para uso repetido num programa ou em outros programas;

O componente pode ser usado para descrever qualquer coisa desde uma simples porta lógica até uma função lógica complexa;

O sinal VHDL pode ser entendido como uma forma de especificar uma conexão de “fio” entre componentes.



# Componentes VHDL

- ✓ Um componente VHDL descreve uma lógica predefinida que pode ser armazenada como um conjunto de declarações numa biblioteca VHDL e utilizada num programa quantas vezes forem necessária
- ✓ Podemos criar um componente VHDL para uma porta AND e então usá-la todas as vezes que desejarmos sem ter que escrever um programa para uma porta AND sempre que precisarmos de uma.
- ✓ Os componentes VHDL são armazenados e disponibilizados para uso quando escrevemos um programa. Isso é similar a termos, por exemplo, uma caixa de CIs disponível quando construímos um circuito.
- ✓ O programa VHDL para qualquer função lógica pode se tornar um componente e ser usado sempre que for necessário num programa maior com o uso de uma declaração de componente.

# Componentes VHDL

```
component name_of_component is
port (port definitions);
end component name_of_component;
```



**Porta AND de 2 entradas**

```
entity AND_gate is
port (A, B: in bit; X: out bit);
end entity AND_gate;

architecture ANDfunction of AND_gate is
begin
X <= A and B;
end architecture ANDfunction;
```



**Porta OR de 2 entradas**

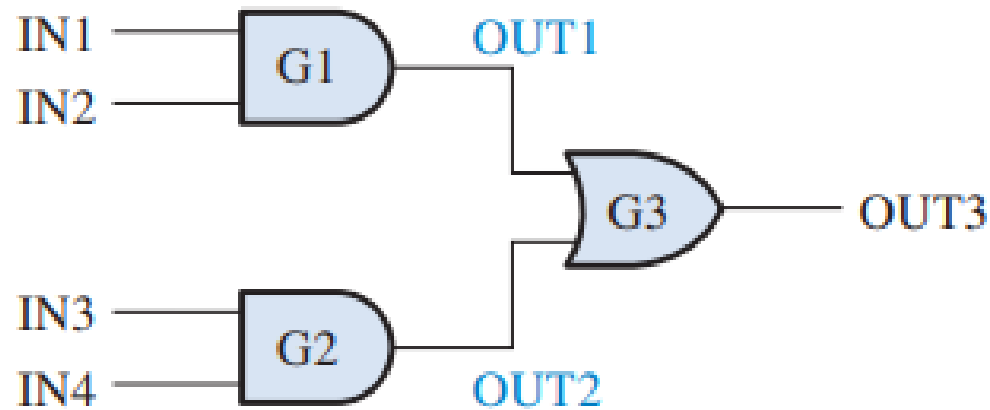
```
entity OR_gate is
port (A, B: in bit; X: out bit);
end entity OR_gate;

architecture ORfunction of OR_gate is
begin
X <= A or B;
end architecture ORfunction;
```



# Componentes em um Programa

Circuito Lógico Soma-de-produtos.



# Sinais

- ✓ Em VHDL, os sinais são equivalentes aos fios que interconectam componentes numa placa de circuito.
- ✓ Os sinais são conexões internas no circuito lógico e são tratados diferentemente das entradas e saídas.
- ✓ As entradas e saídas são declaradas na declaração de entidade usando a declaração de port, os sinais são declarados dentro da arquitetura usando a declaração de sinal.
- ✓ A palavra signal (sinal) é uma palavra reservada em VHDL.

# Programa

```
entity AND_OR_Logic is
    port (IN1, IN2, IN3, IN4: in bit; OUT3: out bit);
end entity AND_OR_Logic;
```

A declaração de arquitetura contém as declarações de componentes para a porta AND e a porta OR, as definições de sinais e os componentes instanciais.

```
architecture LogicOperation of AND_OR_Logic is
```

```
    component AND_gate is
        port (A, B: in bit; X: out bit);
    end component AND_gate;
```

← Declaração de componente para a porta AND

```
    component OR_gate is
        port (A, B: in bit; X: out bit);
    end component OR_gate;
```

← Declaração de componente para a porta OR

```
    signal OUT1, OUT2: bit;
```

← Declaração de sinal

```
begin
```

```
    G1: AND_gate port map (A => IN1, B => IN2, X => OUT1);
```

```
    G2: AND_gate port map (A => IN3, B => IN4, X => OUT2);
```

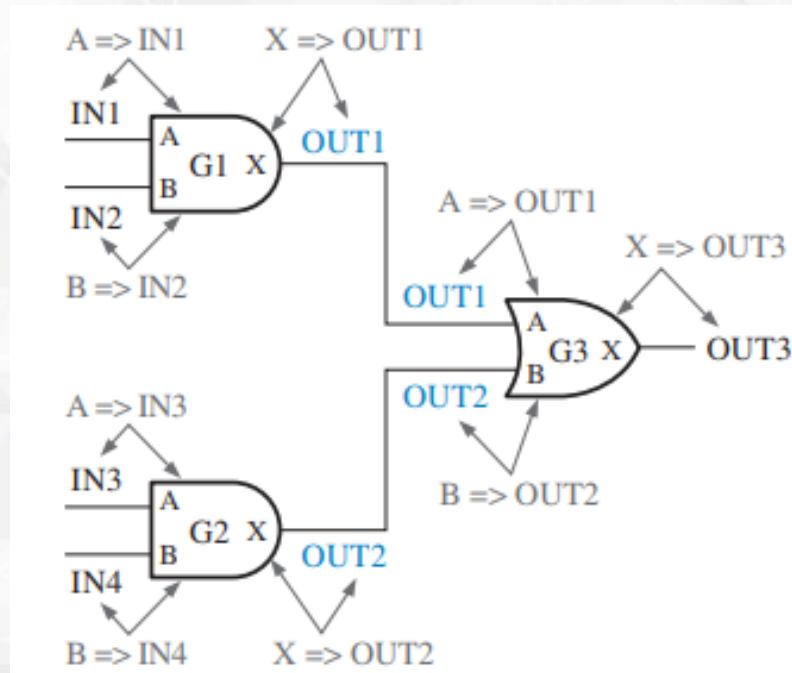
```
    G3: OR_gate port map (A => OUT1, B => OUT2, X => OUT3);
```

← Componentes instanciais

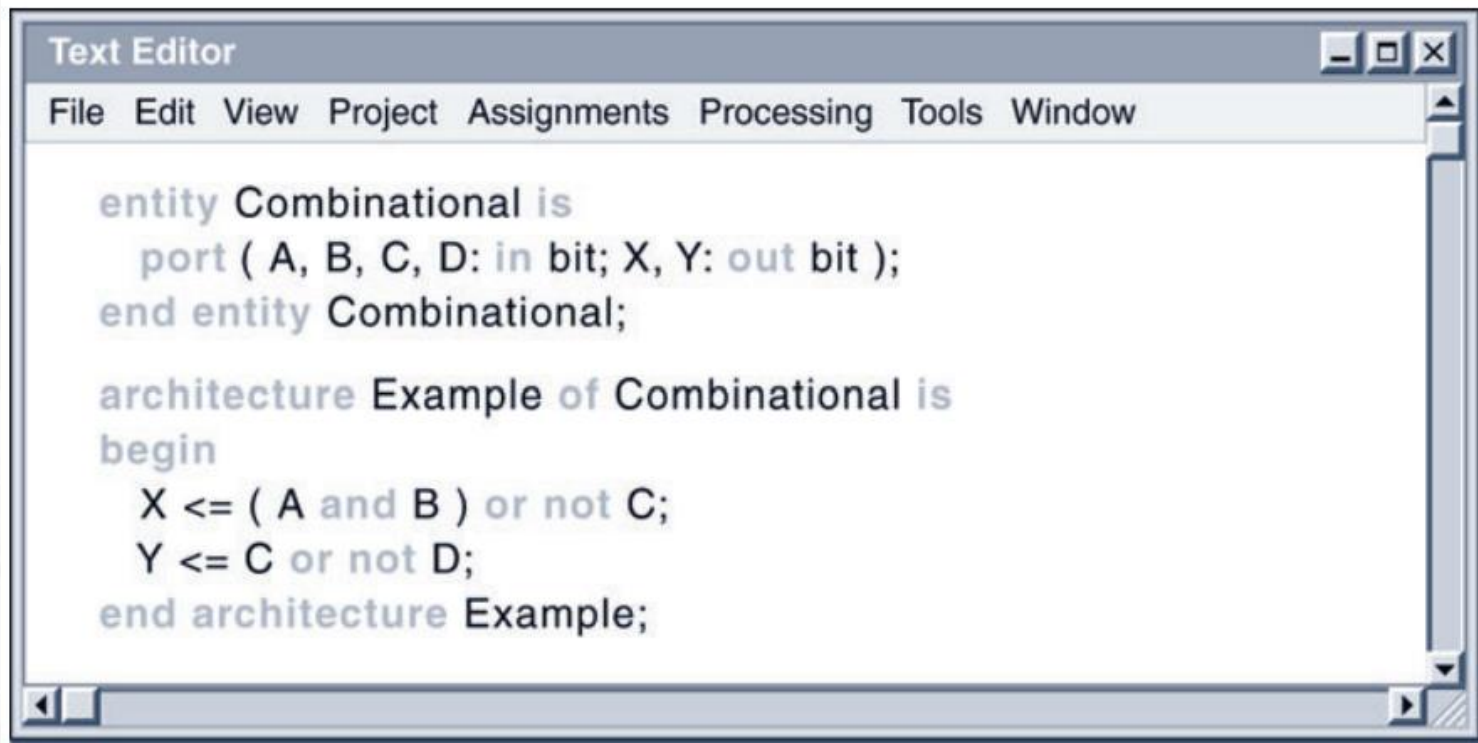
```
end architecture LogicOperation;
```

# Componentes Instanciais

- ✓ Componentes instanciais aparecem entre as palavras reservadas da declaração begin (início) e end (fim).
- ✓ Para cada componente instancial um identificador é definido, tal como G1, G2 e G3 nesse caso.
- ✓ O mapa de port faz essencialmente todas as conexões para a função lógica usando o operador  $\Rightarrow$ . Por exemplo, o primeiro componente instancial.



# Ferramentas de Desenvolvimento



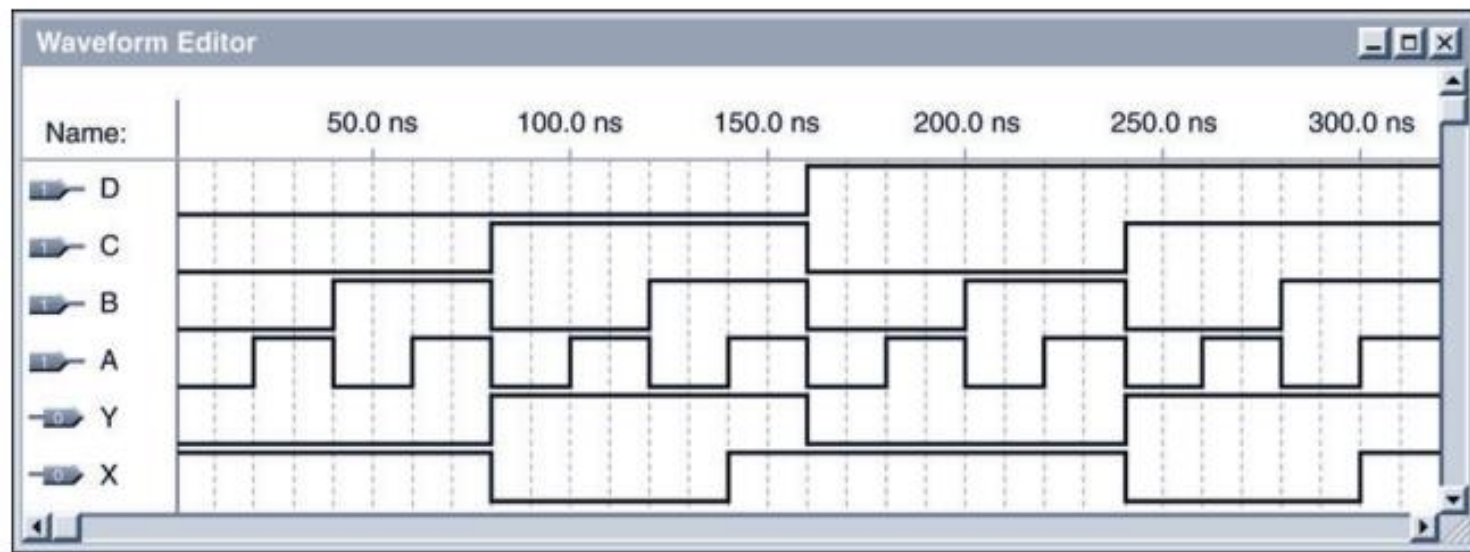
The image shows a screenshot of a 'Text Editor' window. The window has a title bar with the text 'Text Editor' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with the following options: 'File', 'Edit', 'View', 'Project', 'Assignments', 'Processing', 'Tools', and 'Window'. The main text area contains the following VHDL code:

```
entity Combinational is
  port ( A, B, C, D: in bit; X, Y: out bit );
end entity Combinational;

architecture Example of Combinational is
begin
  X <= ( A and B ) or not C;
  Y <= C or not D;
end architecture Example;
```

The code is formatted with indentation for the port declarations and the logic equations. The window also features a vertical scrollbar on the right side and a horizontal scrollbar at the bottom.

# Ferramentas de Desenvolvimento





# Ferramentas de Desenvolvimento

- ✓ O que é um componente VHDL?
- ✓ Qual é a finalidade de um componente instancial na arquitetura de um programa?
- ✓ Como são feitas as interconexões entre os componentes em VHDL?
- ✓ O uso de componentes em programas VHDL representa qual abordagem?