

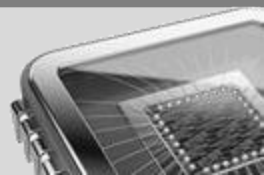
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES II

AULA 11: LINGUAGEM VHDL

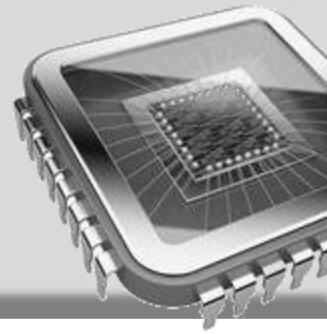
Prof. Max Santana Rolemberg Farias

max.santana@univasf.edu.br

Colegiado de Engenharia de Computação

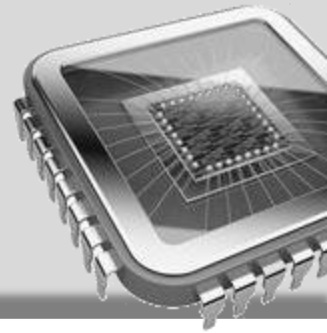


LINGUAGEM VHDL: INTRODUÇÃO



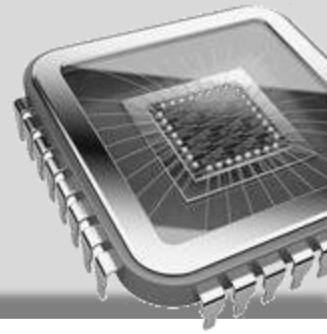
- VHDL é uma linguagem de descrição de hardware que descreve hardware;
 - Originalmente foi projetada para especificar hardware.
 - Hoje já suporta simulação e síntese de hardware.
- Foi padronizada pelo IEEE em 1987.
 - Revisada em 1993, 2000, 2002 e 2004.
- Linguagem utilizada mundialmente por empresas de CAD.
 - Simulação, síntese, propriedade intelectual.

LINGUAGEM VHDL: BENEFÍCIOS

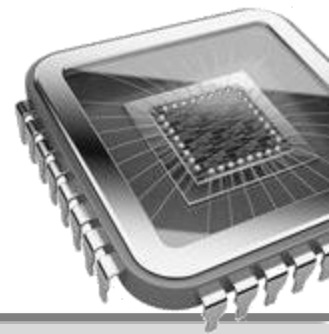


- Especificação do hardware
 - Projetos independentes da tecnologia
 - Ferramentas de CAD compatíveis
 - Flexibilidade (reutilização)
 - Permite explorar, em nível alto de abstração, diferentes alternativas de implementação
 - Permite, através de simulação, verificar o comportamento do hardware
- Nível físico
 - Reduz tempo de projeto (favorece níveis abstratos de projeto)
 - Reduz custo do projeto
 - Elimina erros de baixo nível
 - Reduz o time-to-market

LINGUAGEM VHDL: DESVANTAGEM

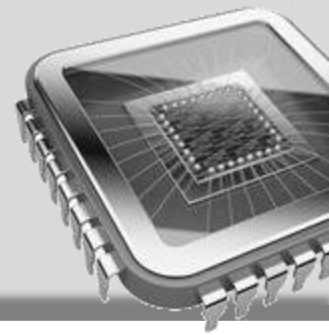


- Desvantagem em relação a diagramas de esquemáticos
 - Hardware gerado pode ser menos otimizado
 - Controlabilidade de projetos reduzidos



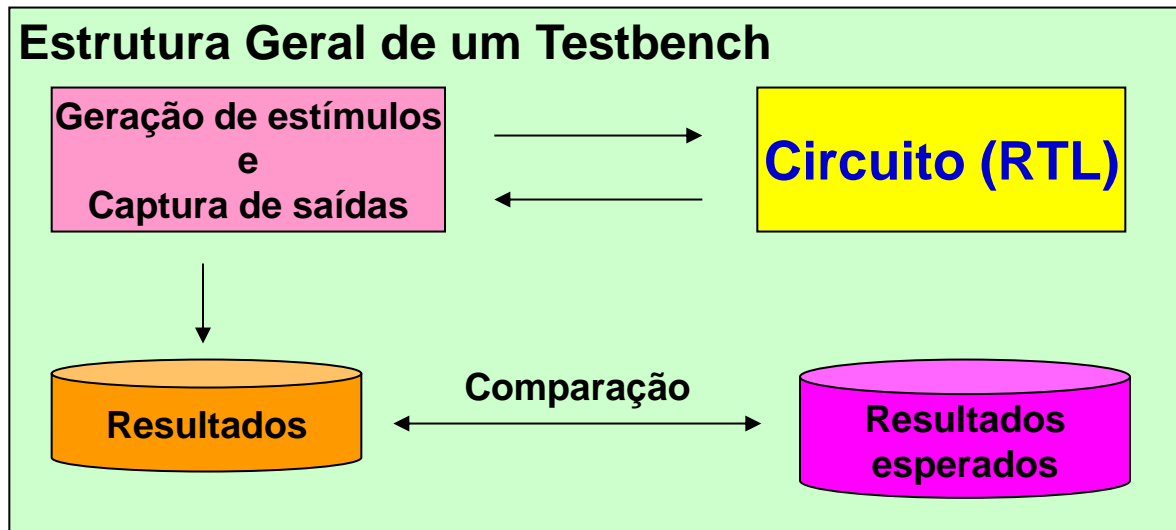
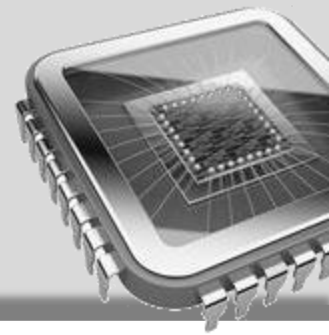
VHDL É UMA LINGUAGEM DE PROGRAMAÇÃO?

LINGUAGEM VHDL

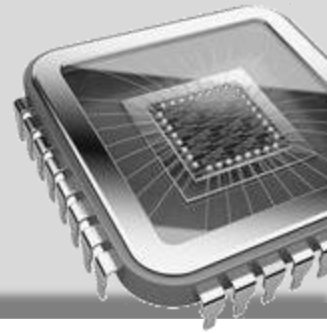


- VHDL **não** é uma linguagem de programação, é uma **linguagem de descrição de hardware!**
 - Código é executado em um **simulador**
 - **Não se enxerga o compilador do VHDL, não há um código executável**
 - Projeto do usuário é especificado no nível RTL, mas não apenas neste nível.
 - Procedimentos de teste do hardware são descritos em VHDL ou outra linguagem (**testbench**)
 - Especificação comportamental do ambiente externo ao projeto (estímulos externos)
 - Interage com o projeto
 - O projeto VHDL pode ser validado em ambiente C/C++

LINGUAGEM VHDL: TESTBENCH



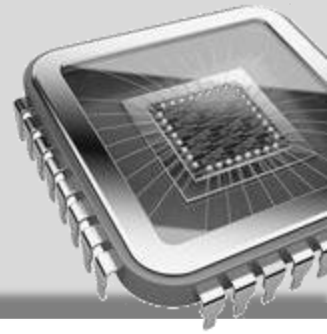
LINGUAGEM VHDL: PACKAGE



- A linguagem VHDL permite utilizar pacotes não definidos nas bibliotecas padrão da linguagem.
 - Dever ser previamente definido antes do entity.
 - O uso é feito por meio da declaração **library** e **use**.
- O mais conhecido e usado é o **STD_LOGIC_1164** da IEEE.
 - Contém a maioria dos comandos adicionais mais usados em VHDL

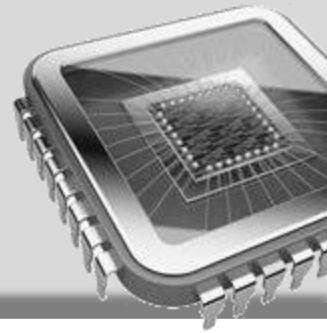
```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

LINGUAGEM VHDL: SIGNAL



- Os sinais são importantes em todos sistemas eletrônicos.
 - Transmitem dados internamente ou externamente ao hardware
 - Os sinais externos são apresentados na entity, pelo comando **port**.
 - Os sinais internos são apresentados na architecture, pelo comando **signal**.
- Os sinais podem ser uma linha (fio) ou um barramento, também chamado de vetor.
 - bit (fio)
 - bit_vector (barramento)

LINGUAGEM VHDL: SIGNAL



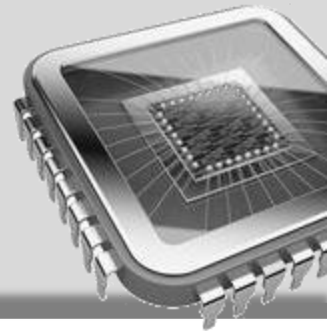
PORT

```
port (<nome_port>: <modo>  
<tipo_port>);
```

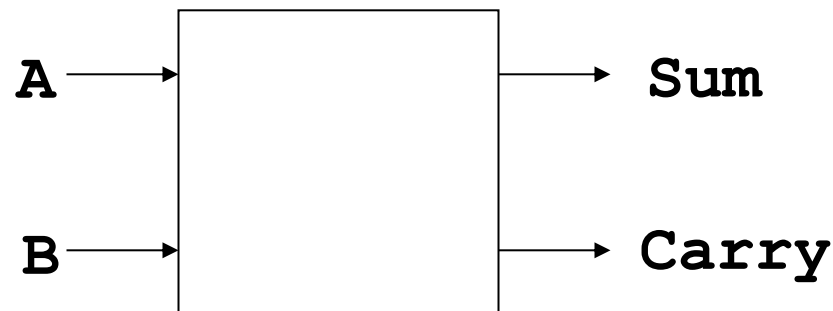
SIGNAL

```
Signal <nome>: <tipo>;
```

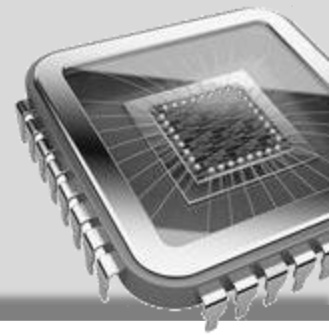
LINGUAGEM VHDL: INTERFACE DO HARDWARE



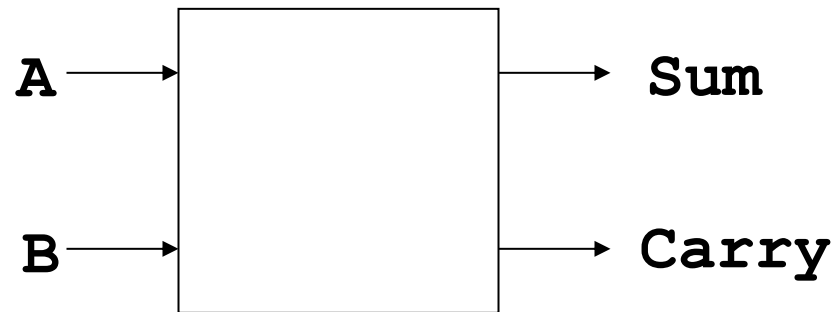
- Todo hardware necessita de uma interface com o ambiente (mundo externo). Em VHDL essa interface é chamada de *entity*.
 - Especifica somente a interface do hardware e o ambiente.
 - Não contém definição do comportamento (arquitetura interna)



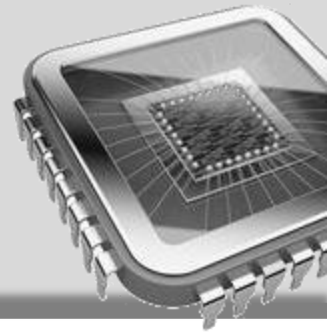
LINGUAGEM VHDL: ENTITY



```
entity halfadd is
port (A, B: in std_logic;
      Sum, Carry: out std_logic);
end halfadd;
```



LINGUAGEM VHDL: COMPORTAMENTO DO HARDWARE



- O comportamento do hardware pode ser descrito:
 - Em termo de funcionalidade, isto é, o que o hardware faz;
 - Especifica as respostas nas saídas .
 - Em termos de estrutura, isto é, como o hardware é composto.
 - Especifica quais componentes devem ser usados e como devem ser ligados
- Em VHDL a especificação do comportamento e/ou estrutura interna da interface de um hardware (**entity**) é chamada de **architecture**.
 - Deve ser **associada a uma entity específica**.
 - Uma entity pode ter várias architectures associadas a ela.
 - Representando diferentes formas de implementar um mesmo módulo

LINGUAGEM VHDL: ARCHITECTURE

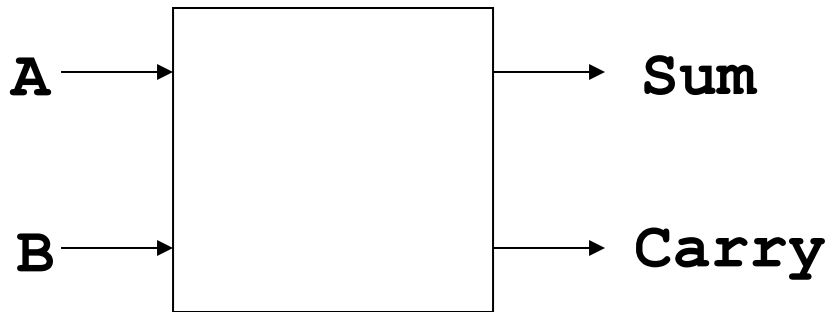
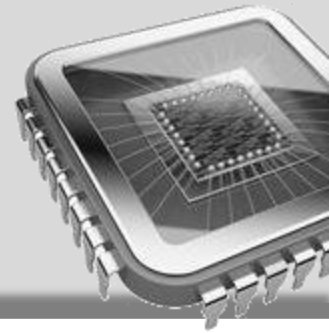


Tabela Verdade

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

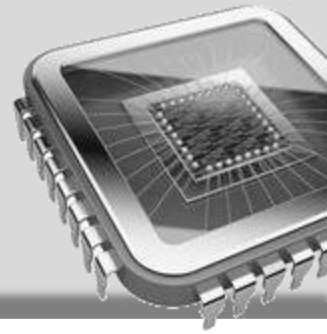
```
architecture comp of halfadd is  
begin
```

```
    Sum <= A xor B;
```

```
    Carry <= A and B;
```

```
end comp;
```

LINGUAGEM VHDL: SOMADOR (HALFADD)



```
library IEEE;  
use IEEE.Std_Logic_1164.all;  
  
entity halfadd is  
port (A, B: in std_logic;  
      Sum, Carry: out std_logic);  
end halfadd;  
  
architecture comp of halfadd is  
begin  
    Sum <= A xor B;  
    Carry <= A and B;  
end comp;
```

