



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico – CTC**  
**Departamento de Engenharia Elétrica**



# **“EEL7020 – Sistemas Digitais”**

**Prof. Eduardo Augusto Bezerra**

**Eduardo.Bezerra@eel.ufsc.br**

**Florianópolis, agosto de 2011.**

# Plano de Aula



## **“Projeto de Sistemas Digitais com VHDL”**

- **Objetivos:**
  - Apresentar uma visão geral de VHDL
  - Exemplo de descrição VHDL
  - Introdução ao Quartus II – ferramentas de desenvolvimento
  - Estudo de caso / exercício

# VHDL - Visão Geral

---

- VHDL - linguagem para descrição de hardware (*hardware description language*)
- **VHDL** = **V**HSIC **H**ardware **D**escription **L**anguage
- VHSIC = Very High Speed Integrated Circuits. Programa do governo dos USA do início dos anos 80.
- No final da década de 80, VHDL se tornou um padrão IEEE (Institute of Electrical and Electronic Engineers).
- Existem diversas ferramentas para simular e sintetizar (gerar hardware) circuitos descritos em VHDL.
- Outras linguagens de descrição de hardware: Verilog, SystemC, AHDL, Handel-C, System Verilog, Abel, Ruby, ...

# VHDL - Visão Geral

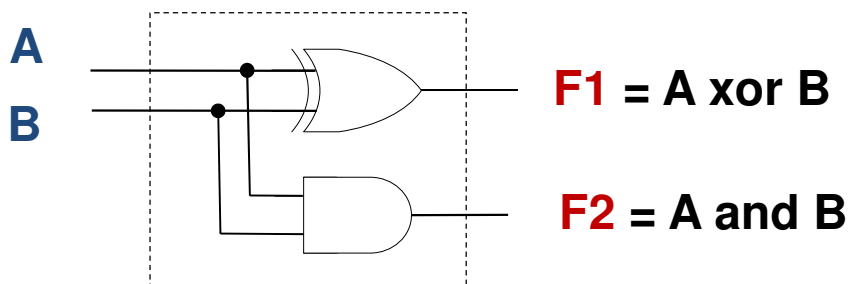
---

- O projeto de um circuito digital pode ser descrito em VHDL em diversos níveis de abstração (estrutural, comportamental).
- Descrições no nível de transferência entre registradores (RTL, Register Transfer Level) são bastante utilizadas.
- VHDL **NÃO** é uma **linguagem de programação**, e as ferramentas de síntese (não são de “compilação”) não geram códigos executáveis a partir de uma descrição VHDL.
- Descrições em VHDL podem ser simuladas (executadas em um simulador).
- Descrições em VHDL podem ser utilizadas para gerar um hardware (arquivo para configuração de um FPGA, por exemplo).
- A geração de estímulos para simulação VHDL é realizada por intermédio de testbenches.
- Um testbench define os estímulos externos a serem utilizados como entrada para o circuito (definição do comportamento externo ao circuito sob teste).
- O testbench pode ser escrito em VHDL ou em diversas outras linguagens (ex. C, C++, ...).

# Descrição de circuito digital em VHDL

## ENTITY

A	B	F1	F2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

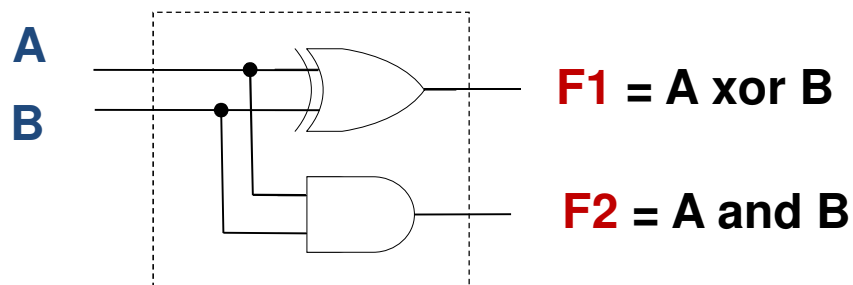


```
entity halfadd is
port (A: in std_logic;
      B: in std_logic;
      F1: out std_logic;
      F2: out std_logic
);
end halfadd;
```

**ENTITY** – define os “pinos” do circuito digital (sinais), ou seja, a **interface** entre a lógica implementada e o mundo externo.

# Descrição de circuito digital em VHDL

## ARCHITECTURE



```
architecture circuito_logico of halfadd is
begin
    F1 <= A xor B;
    F2 <= A and B;
end circuito_logico;
```

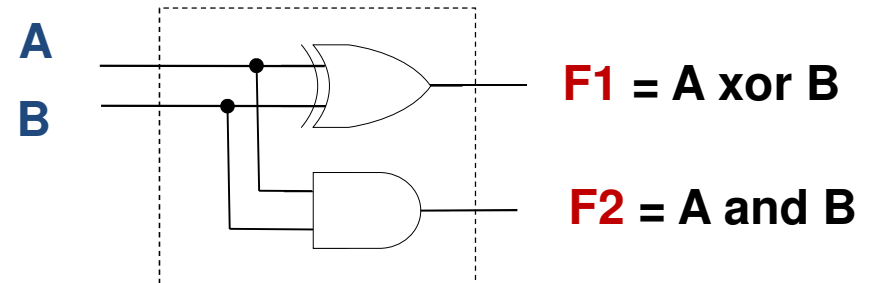
**ARCHITECTURE** – define a funcionalidade do circuito digital, utilizando os “pinos” de entrada e saída listados na ENTITY em questão. Uma ENTITY pode possuir diversas implementações diferentes (diversas ARCHITECTURES).

# Descrição completa do circuito em VHDL (Entity e Architecture)

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

```
entity halfadd is  
  port (A: in std_logic;  
        B: in std_logic;  
        F1: out std_logic;  
        F2: out std_logic  
        );  
end halfadd;
```

```
architecture circuito_logico of halfadd is  
begin  
  F1 <= A xor B;  
  F2 <= A and B;  
end circuito_logico;
```



*Para utilizar o tipo `std_logic` é necessário incluir um pacote da biblioteca IEEE.*

## **Tarefa a ser realizada na aula prática**



# Tarefa a ser realizada na aula prática

---

- Utilizando a ferramenta Quartus II da Altera, criar um projeto VHDL que implemente o circuito apresentado no slide 7 (*and* e *xor*).
- Realizar a simulação do circuito (VHDL) no Quartus II, e obter a tabela verdade.
- O objetivo principal dessa aula prática é possibilitar que o aluno tenha um primeiro contato com as ferramentas de desenvolvimento a serem utilizadas durante o semestre.
- Seguir o tutorial descrito no arquivo: *tut\_quartus\_intro\_vhdl.pdf*
- Após a simulação do circuito no Quartus II, testar o funcionamento no FPGA da plataforma de desenvolvimento.
- No slide a seguir são apresentadas algumas dicas para o desenvolvimento da tarefa dessa aula de laboratório.

# Fluxo de desenvolvimento VHDL no Quartus II

---

1. File -> New Project Wizard
2. No “project wizard”, seguir exatamente os mesmos passos do tutorial da última aula
3. **File -> New -> VHDL File**
4. Copiar o fonte VHDL do slide anterior para esse novo arquivo, e salvar.
5. Assignments -> Assignment Editor -> associar os sinais A, B, F1 e F2 aos pinos existentes na placa (ver manual da placa DE2 - DE2\_UserManual.pdf)
6. Compilar o VHDL (síntese)
7. **ATENÇÃO!!!** Verificar se o nome da entity é o mesmo nome do projeto, para evitar erros na síntese.

# Resumo do tutorial *tut\_quartus\_intro\_schem.pdf*

---

8. **Simulação Funcional** – Circuito sintetizado é testado para verificação de sua funcionalidade -> não considera informação de temporização.
  - *File – New – Vector Waveform File*
  - *Edit – End Time – 200 ns*
  - *View – Fit in Window (^W)*
  - *Edit – Insert Node or Bus – Node Finder*
  - *Assignments – Settings – Simulator Settings – Functional*
  - *Processing – Generate Functional Simulation Netlist*
  - *Processing – Start Simulation*
9. **Análise Temporal** – Análise dos atrasos de propagação de sinais no circuito após a síntese física – resulta em relatório constando o desempenho esperado.
10. **Simulação Temporal** – teste para verificação com atrasos.
11. **Programação** – FPGA é carregado com circuito, configurando fisicamente elementos de processamento e roteamento.

***Tools – Programmer. Hardware Setup – USB-Blaster. Start!***