

Introdução à linguagem VHDL

DCA0202 - Circuitos Digitais

Prof. Carlos M. D. Viegas

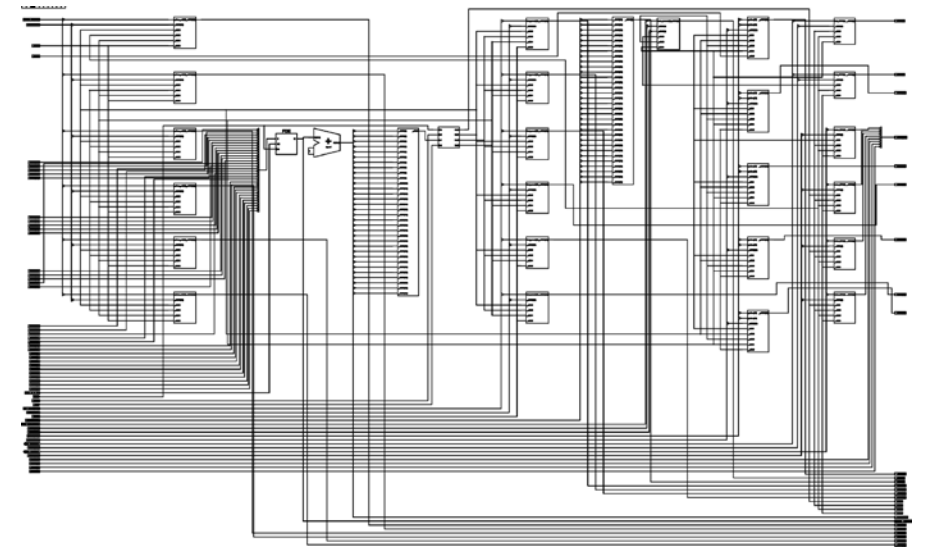
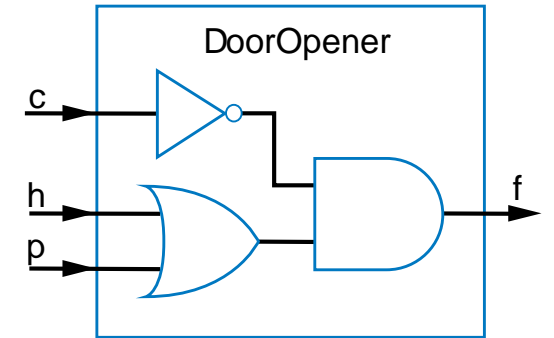


Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte



Introdução

- Em um projeto, o modo esquemático contém informações em formato gráfico de um determinado circuito
 - É fácil identificar as entradas, saídas e portas lógicas
 - Mas para circuitos muito grandes, esta tarefa se torna complexa
- Então, foram criadas linguagens textuais com o intuito de facilitar o projeto de circuitos muito grandes
 - Linguagens HDL
 - HDL = Hardware Description Language
 - Criadas para descrever circuitos de forma textual
 - Exemplos: VHDL, Verilog, SystemC



Introdução à Linguagem VHDL

- VHDL é uma linguagem HDL
 - VHSIC HDL ou Very-High-Speed Integrated Circuit HDL
 - Criada em 1980 a pedido do exército norte-americano
 - Baseado na linguagem ADA
 - De propósito geral

```
with Ada.Text_IO; use Ada.Text_IO;  
procedure Hello is begin  
    Put_Line ("Hello, world!");  
end Hello;
```

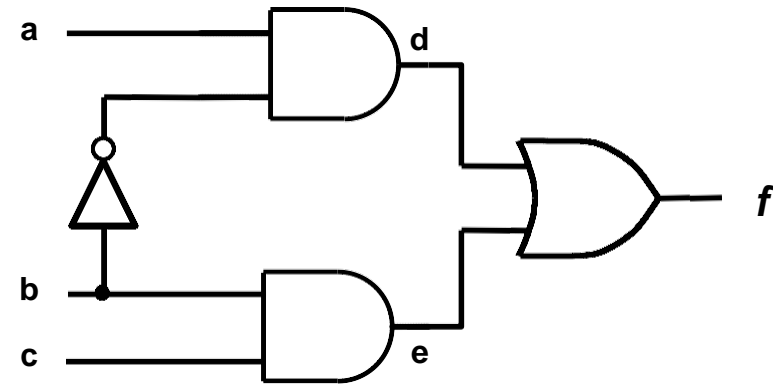
Introdução à Linguagem VHDL

- Exemplo de circuito usando em VHDL

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity exemplo IS  
Port (a, b, c : IN std_logic;  
      f : OUT std_logic);  
End exemplo;
```

```
Architecture estrutural OF exemplo IS  
signal d, e : std_logic;  
Begin  
    f <= d or e;  
    d <= a and not(b);  
    e <= b and c;  
End estrutural;
```



Introdução à Linguagem VHDL

- Principais blocos

```
Library IEEE;  
use IEEE.std_logic_1164.all;
```

```
Entity exemplo IS  
Port (a, b, c : IN std_logic;  
      f : OUT std_logic);  
End exemplo;
```

```
Architecture estrutural OF exemplo IS  
signal d, e : std_logic;  
Begin  
  f <= d or e;  
  d <= a and not(b);  
  e <= b and c;  
End estrutural;
```



Cabeçalho:

- Bibliotecas a serem utilizadas



Entity:

- Define o nome
- Define as interfaces
- Ports Inputs/Outputs
- Tipos de sinal



Architecture:

- Descreve conteúdo funcional do componente
- Podem existir várias
- Definição de sinais internos
- Atribuição de sinais
- Ordem de declaração não é importante

Introdução à Linguagem VHDL

- Como declarar as bibliotecas, entidades e arquiteturas?

- Biblioteca

```
Library nome_da_biblioteca ;  
use nome_do_pacote.all ;
```

- Entidade

```
Entity nome_da_entidade IS  
    Port (  
        nome_da_porta : modo_da_porta tipo_de_sinal ;  
        nome_da_porta : modo_da_porta tipo_de_sinal  
    );  
End nome_da_entidade ;
```

- Arquitetura

```
Architecture nome_da_arquitetura OF nome_da_entidade IS  
    [parte declarativa]  
Begin  
    [código lógico]  
End nome_da_arquitetura ;
```

Introdução à Linguagem VHDL

- Bibliotecas

- Algumas bibliotecas já são automaticamente incorporadas, incluindo os tipos básicos de dados como bit, boolean, etc.
- A biblioteca std_logic_1164 pertence ao IEEE e define o tipo lógico std_logic
 - O tipo std_logic inclui
 - Alta impedância ('Z')
 - Don't care ('-')

Introdução à Linguagem VHDL

- Entidade

```
Entity nome_da_entidade IS
```

```
Port (
```

```
    nome_da_porta : modo_da_porta tipo_de_sinal ;
```

```
    nome_da_porta : modo_da_porta tipo_de_sinal
```

```
);
```

```
End nome_da_entidade ;
```

← A última declaração não tem ';'

- Modo da porta:
 - IN : entrada
 - OUT : saída
- Tipo de sinal:
 - BIT: 0 ou 1
 - BIT_VECTOR: vetor de bits 0 ou 1
 - STD_LOGIC_VECTOR: vetor de 9 possíveis valores
 - INTEGER: valores inteiros

Nove valores para STD_LOGIC

- 'U': Não inicializado
- 'X': Desconhecido
- '0': Nível baixo
- '1': Nível alto
- 'Z': Alta impedância
- 'W': Desconhecido fraco
- 'L': Nível baixo fraco
- 'H': Nível alto fraco
- '-': "Don't care"

Introdução à Linguagem VHDL

- Arquiteturas

- É onde fica o código propriamente dito

```
Architecture nome_da_arquitetura OF nome_da_entidade IS  
[parte declarativa]  
Begin  
    [código lógico]  
End nome_da_arquitetura ;
```

- Parte declarativa: declaração de objetos do tipo:
 - Type
 - Signal
 - Constant
 - Component
 - Function
- Código lógico: contém o código VHDL propriamente dito

Introdução à Linguagem VHDL

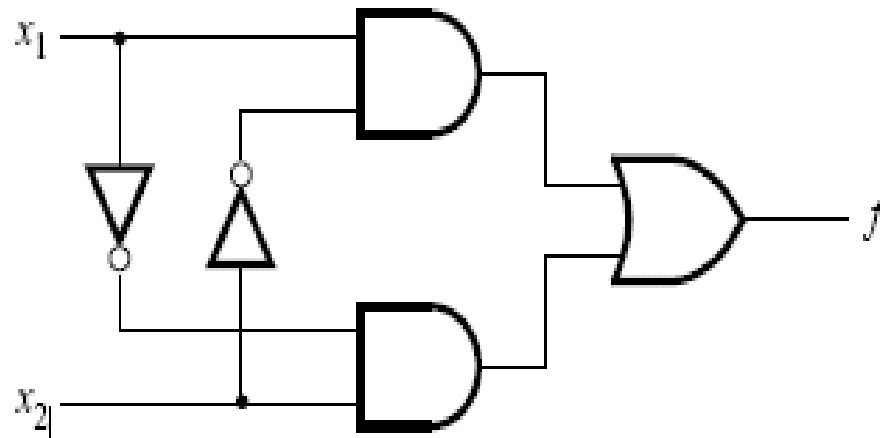
- Arquiteturas
 - Permite utilizar diversos tipos de operadores
 - Lógicos:
 - NOT, AND, OR, NAND, NOR, XOR, XNOR
 - Aritméticos:
 - +, -, *, /, **
 - De comparação
 - =, /=, >, <, >=, <=
 - De concatenação
 - &
 - De atribuição
 - <= (para atribuição de sinais)
 - := (para atribuição de valores)

Introdução à Linguagem VHDL

- Criação de um novo projeto
 - Abrir o Quartus II
 - File > New Project Wizard
 - Definir diretório onde o projeto será armazenado: **tut_vhdl**
 - Escolher nome do projeto (2 próximos campos): **light_vhdl**
 - Next
 - Next (mecanismo para adicionar arquivos)
 - Family device settings:
 - Escolher Cyclone II **EP2C35F672C6**
 - Next (other EDA tools)
 - Finish

Introdução à Linguagem VHDL

- Criação de um novo projeto
 - Circuito de controle de luz



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Introdução à Linguagem VHDL

- Criação de um novo projeto
 - File > New > VHDL File
 - File > Save as > light_vhdl > OK
 - Atenção: marcar checkbox “Add file to current project”
 - Copiar e colar texto abaixo e salvar

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY light_vhdl IS
PORT (
    x1, x2 : IN STD_LOGIC ;
    f : OUT STD_LOGIC
);
END light_vhdl ;

ARCHITECTURE LogicFunction OF light_vhdl IS
BEGIN
    f <= (x1 AND NOT x2) OR (NOT x1 AND x2);
END LogicFunction ;
```

Introdução à Linguagem VHDL

- Criação de um novo projeto
 - Os passos seguintes são semelhantes ao diagrama esquemática
 - Compilação
 - Atribuição dos pinos
 - Criação da waveform (opcional)
 - Simulação (opcional)
 - Programação/gravação na placa
 - Teste

Introdução à Linguagem VHDL

- Tarefa 1:
 - Crie um circuito utilizando a linguagem VHDL que simule uma porta NOR de 3 entradas
 - Faça a tabela verdade (para confirmar os valores)
 - Escolha três switches na placa para representar as entradas
 - Programe a saída para acender um led apenas quando todas as entradas forem 0
- Tarefa 2:
 - Crie um circuito utilizando a linguagem VHDL com 3 entradas, utilizando duas portas AND de 2 entradas cada
 - Faça a tabela verdade (para confirmar os valores)
 - Escolha três switches na placa para representar as entradas
 - Programe a saída para acender um led apenas quando as entradas apresentarem 1 0 1