

Desenvolvimento de Circuitos Digitais com VHDL e FPGA

Prof. Mário Véstias



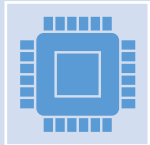
Índice

- Fluxo de desenvolvimento de circuitos digitais
- Linguagens de Descrição de Hardware
- Conceitos Básicos de VHDL
- “Entity/Architecture”
- Descrição de circuitos digitais em VHDL
- Introdução ao Ambiente de Projeto - Quartus

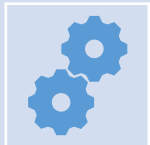
Fluxo de Desenvolvimento de Circuitos Digitais

- Descrição do circuito – Descrição informal do circuito a projetar;
- Especificação formal do circuito – Estruturação do problema e geração das funções lógicas dos subcircuitos (tabelas de verdades, etc.);
- Derivação das funções lógicas – Obtenção das funções lógicas e simplificação;
- Elaboração do circuito lógico – descrição das funções lógicas com portas lógicas. Obtenção do logigrama completo do circuito;
- Descrição do circuito lógico em VHDL – a descrição em VHDL permite utilizar ferramentas automáticas de síntese de implementação na tecnologia FPGA;
- Simulação do circuito – Simulação do circuito para verificar se a descrição lógica está de acordo com a funcionalidade pretendida;
- Implementação e Teste – Implementação do circuito na tecnologia escolhida e teste em um dispositivo FPGA.

Linguagens de Descrição de Hardware



A complexidade da maioria dos circuitos digitais requer melhores métodos de descrição do hardware e ferramentas CAD (*Computer Added Design*);



As linguagens de descrição de hardware são utilizadas para descrever o circuito e são interpretadas pelas ferramentas CAD;



Linguagens mais utilizadas:

VHDL – VHSIC (Very High Speed Integrated Circuit) Hardware
Description Language

Verilog

Implementação do Circuito - Tecnologia

- Componentes discretos (e.g., TTL) – raramente utilizado;
- Circuito integrado não programável (ASIC – *Application Specific Integrated Circuit*);
- Circuito integrado programável (PAL, CPLD, FPGA, etc.);
- Em laboratório serão testados circuitos TTL;
- Componente laboratorial utiliza FPGA.

Procedimentos no Desenvolvimento de Circuitos Digitais

- Todos os circuitos digitais terão de ser inicialmente descritos com um logigrama e simplificados;
- A linguagem VHDL é utilizada para descrever o circuito digital de modo a ser interpretado pelas ferramentas de CAD;
 - Apenas será utilizado um subconjunto do VHDL que permite a descrição direta de funções lógicas;
- A linguagem VHDL é também utilizada para descrever um ficheiro de simulação. Não tem associado um esquema lógico;
- Todos os circuitos devem ser testados em FPGA.

Descrição de Circuitos em VHDL

Entidade/Arquitetura

- A descrição de um circuito digital em VHDL é composta por dois módulos principais:
 - **Entidade** (*entity*) – Identificação das entradas e das saídas do circuito;
 - **Arquitetura** (*architecture*) – Descrição das funções lógicas do circuito.

Declaração da Entidade/Arquitetura

```
LIBRARY IEEE;
use IEEE.std_logic_1164.all;

entity nome_circuito is
    port (
        ...
    );
end nome_circuito;

architecture nome_arquitetura of nome_circuito is
-- declaração de sinais

begin
-- descrição do circuito digital
end nome_arquitetura;
```

Entidade – Declaração de Portos

- A declaração dos portos estabelece a ligação do circuito digital com o exterior;
- A declaração de cada porto tem três partes;
 - Nome – identificador do porto;
 - Modo – *in, out*;
 - Tipo de sinal – tipo predefinido.

```
entity nome_circuito is
    port (
        nome1 : modo tipo;
        ...
        nome2 : modo tipo;
    );
end nome_circuito;
```

Tipo de Sinais

- Um sinal em VHDL corresponde a um fio;
- Um sinal digital basta ter os valores '0', '1' e 'Z';
- Tipo **std_logic** é mais extenso e é útil na simulação. Pode tomar os valores seguintes:
 - '0' e '1'
 - '-' – indiferença
 - 'Z' – alta impedância
 - 'U' – não definido (simulação)
 - 'X' – não conhecido (simulação)
- Tipo **std_logic_vector** é utilizado para declarar palavras de bits.

EX:

```
signal a, b : std_logic;  
signal c : std_logic_vector(3 downto 0);
```

O sinal 'c' é formado pelos fios b(3), b(2), b(1), b(0)

Descrição de Funções Lógicas

- As funções lógicas são descritas com os seguintes operadores:
 - NOT, AND, OR, NAND, NOR, XOR, XNOR
- A atribuição de sinais é feita do seguinte modo:

`senal_destino <= função_lógica;`

EX:

`a <= b and c;`

Exemplos de Atribuição a Sinais

```
signal a, b, c, d : std_logic;
signal v1, v2    : std_logic_vector(2 downto 0);

begin

a <= '0';      -- sinal 'a' fica com o valor lógico '0'
b <= '1';      -- sinal 'b' fica com o valor lógico '1'
c <= b ;       -- sinal 'c' fica com o valor lógico de 'b'

v1 <= "111";   -- palavra 'v1' fica com o valor binário "111"

d <= b and c;  -- sinal 'a' fica com o produto lógico entre 'b' e 'c'

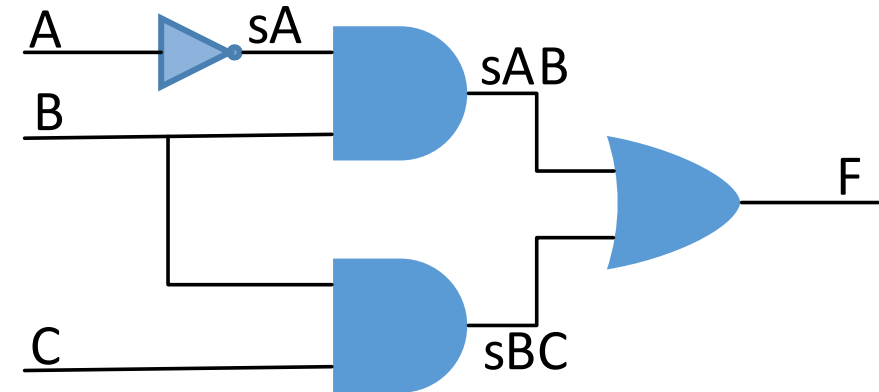
...
```

Exemplo de Descrição

```
LIBRARY IEEE;
use IEEE.std_logic_1164.all;

entity logic is
    port (
        A, B, C : in std_logic;
        F : out std_logic;
    );
end logic;

architecture teste of logic is
    signal sA, sAB, sBC : std_logic;
begin
    sA <= not A;
    sAB <= sA and B;
    sBC <= B and C;
    F <= sAB or sBC;
end nome_arquitetura;
```



Introdução ao Ambiente de Projeto QUARTUS

Ambiente de Projeto

- Ambiente de projeto para implementação e simulação:
 - Quartus Prime Lite Edition (no license);
 - Last version 20.1.1;

