《VHDL语言》

[《VHDL语言》 1](#_Toc184978606)

[1认识vhd语言 3](#_Toc184978607)

[1.1RTL HDL语言特点 3](#_Toc184978608)

[1.2 VHDL语言文件组织形式 4](#_Toc184978609)

[1.2.1库 4](#_Toc184978610)

[1.2.2包 4](#_Toc184978611)

[1.2.3实体 5](#_Toc184978612)

[1.2.4结构体 5](#_Toc184978613)

[2VHDL语言基本语法 6](#_Toc184978614)

[2.1数的表示 6](#_Toc184978615)

[2.2位的表示 7](#_Toc184978616)

[2.3沿的表示 7](#_Toc184978617)

[2.4操作符 7](#_Toc184978618)

[2.5数据类型 8](#_Toc184978619)

[2.6量 9](#_Toc184978620)

[2.7赋值 9](#_Toc184978621)

# 1认识vhd语言

作为数字电路设计工具之一HDL，rtl级描述语言目前主要有三种verilog,ahdl，vhdl三种，在国内以verilog为主，在一些研究所军工类企业也有要求用VHDL语言的，至于AHDL语言国内使用非常少。

## 1.1RTL HDL语言特点

**Verilog,ahdl,vhdl都属于RTL级建模**语言，对于系统算法级建模语言为system verilog和system c使用率较高。

|  |  |  |
| --- | --- | --- |
| VHDL/ verilog | 特点 | 特点 |
| 描述层次 | RTL | RTL |
| 语法结构 | 严谨 | 灵活 |
| 文件格式 | 包括库，包，实体，结构等 | 包括头文件，模块 |
| 数据类型 | 强数据类型（2个不同数据类型的信号是不可以赋值）  数据类型较为丰富 | 弱数据类型  数据类型少 |
| 量 | 常量  信号  变量（局部信号） | 常量  信号 |
| 进程 | Process | allways  initial |
| 赋值 | :=（初始化，变量赋值）  <=（信号赋值） | =(阻塞赋值)  <=（非阻塞赋值） |
| 建模 | 流建模（wen,with）  结构建模(定义元件)  行为建模（pross） | 流建模(assign )  结构建模（直接例化，不需要定义元件）  行为建模(always) |

## 1.2 VHDL语言文件组织形式

任何语言都是通过文件直接表示出来，verilogt通过模块文件表示，VDHL通过包，库，结构体，实体组织语言。

### 1.2.1库

库是存放vhdl设计的存储空间，每个存储空间都以物理映射，映射名字就是库名，任何语言描述的设计文件都必须保存在库中，库不能由用户定义，只能是语言设计者或者授权单位或者个人定义。

**Library 库名;**

**Use 库名.报名，项名；**

|  |  |
| --- | --- |
| **库名** | **说明** |
| **IEEE** | **标准库，语言自带的库** |
| **STD** | **IEEE库扩展标准库** |
| **VITAL** | **针对ASIC设计提供的高精度仿真模型** |
| **Work** | **临时库，用来存放所有用户设计有关的文件** |

### 1.2.2包

包用来定义VHDL语言的公共属性（常量,函数），即为头文件，用与其他设计定义的常量或者文件。

**Package** 包名 **is**

定义函数；

定义常量；

**End**  包名；

注意:为了方便函数的引用，vhdl语言引入了包体的理念

**Package body 包名 is**

**函数实现；**

**End** **包名；**

### 1.2.3实体

实体用来定义模块的外部属性，外部数据包括端口定义，模块参数

Entity 模块名 is

Port

(

a ： std\_logic\_vertor (7 downto 0);

…..

);

**Generic**

(

Constant COL\_MAX : std\_logic\_vertor (7 downto 0) := 600

);

End 模块名;

### 1.2.4结构体

结构体用来描述模块的内部行为，即电路功能描述，通过流建模，行为建模，结构建模方式描述电路行为。

**Architecture**  结构体名 **of** 模块名 **is**

*Compent* 元件名 **is**

End Compent;

Begin

#流建模

When …

With …

直接赋值；

#结构建模

Prot map

…

#行为建模

Process ()

Begin

End process;

**End** 结构体名;

# 2VHDL语言基本语法

## 2.1数的表示

数学类型数表示

10#180#：表示10进制数180 如：a := 10#180#;

10#180#e2：表示10进制数180\*102 如：a := 10#180#e2;

2#10101#：表示20进制数10101 如：a := 2#10101#;

2#10101#e2：表示2进制数10101\*22 如：a := 2#10101#e2;

16#AB#：表示16进制数AB 如：a := 16#AB#;

16#AB#E3：表示16进制数AB\*163 如：a := 16#AB#E3;

逻辑类型数的表示

B“10101” :表示二进制数10101

O“25” ：表示8进制数25

X”AB” :表示16进制数AB，注意信号的位宽必须是4的正数倍，否则无法赋值

## 2.2位的表示

[7:0] ： （7 **downto** 0）

[0:7] : (0 to 7)

## 2.3沿的表示

上升沿

clk**’event** and clk=‘1’

rising\_edge(clk)

下降沿

clk`event and clk=’0’

falling\_edge(clk)

## 2.4操作符

|  |  |  |
| --- | --- | --- |
| 运算符 | 说明 | 例子 |
| 基本运算符 | |  |
| + | 加法 | C<=a+b; |
| - | 减法 | C<=a-b; |
| \* | 乘法 | C<=a\*b; |
| / | 除法 | C<=a/b; |
| Rem | 取余 | C<=rem a； |
| 符号运符 | | |
| + | 正数 | C<=+a; |
| - | 负数 | C<=-a; |
| 拼接运算符 | | |
| & | 位拼接 | C<=a(7 downto 0) & b(6 downto 0) |
| 关系运算符 | | |
| = | 等于 | a=b |
| \= | 不等于 | a\=b |
| < | 小于 | A<b |
| <= | 小于等于 | A<=b |
| > | 大于 | a>b |
| >= | 大于等于 | a>=b |
| 逻辑运算符（不支持位运算） | | |
| And | 与 | A and b |
| Or | 或 | A or b |
| nand | 与非 | A nand b |
| nor | 或非 | A nor b |
| xor | 异或 | A xor b |
| Nxor | 同或 | A nxor b |
| 关联运算符 | | |
| => | 关联 | 需在case pot map下使用 |

## 2.5数据类型

标准数据类型

* 数学类型

Integer :整数型

Real:实数类型

Boolean：布尔量

Character:字符类型

StringL：字符串类型

Natural:自然数类型

Positive:正整数类型

Time:时间类型

Severity leve:错误等级类型

* 逻辑类型

bit:单比特逻辑类型

bit\_vector:多比特逻辑类型

std\_logic: 单比特逻辑类型

std\_logic\_vector: 多比特逻辑类型

自定义数据类型

特殊类型

**Type** mem **is array** (7 downto 0) **of**  std\_logic\_vector (255 downto 0);

Type SATE\_CODE **IS**(S0,S1,S2,S3);

子类型(支持重载)

**Subtype** my\_integer **is** integer **range** 0:255；

## 2.6量

常量

Constant a: std\_logic\_vector (7 downto 0) := 10;

信号

**Signal** a: std\_logic\_vector (7 downto 0) := 10;

变量（局部信号，只能在进程语句中申明）

**varibal** a: std\_logic\_vector (7 downto 0) := 10;

## 2.7赋值

* 初值

所有的量（常量，变量，信号）赋初值用:=

* 信号赋值

信号赋值用<=

* 变量赋值

变量赋值用:=

# 3练习

