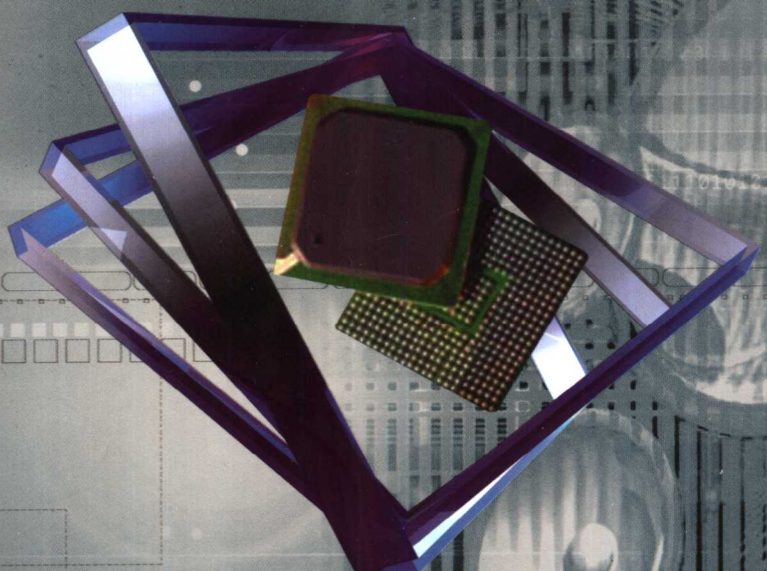
电子电气设计
与自动控制系列

VHDL应用开发技术与工程实践

求是科技　编著

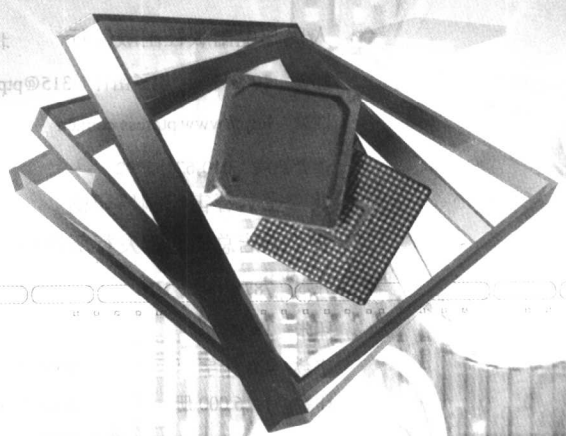


人民邮电出版社
POSTS & TELECOM PRESS

电子电气设计
与自动控制系列

VHDL应用开发技术与工程实践

求是科技　编著



人民邮电出版社

图书在版编目（CIP）数据

VHDL 应用开发技术与工程实践 / 求是科技编著. —北京：人民邮电出版社，2005.1
ISBN 7-115-12718-2

Ⅰ. V... Ⅱ. 求... Ⅲ. 硬件描述语言，VHDL Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2004）第 139949 号

内容提要

本书以 VHDL 程序设计基础与工程实践为内容，全面介绍了 VHDL 程序设计的基础知识和基本技术，并结合工程实例讲解电路设计的基本流程和 VHDL 技术的应用。本书基本涵盖了 VHDL 程序设计的主要技术要点，包括 VHDL 的数据类型和操作符、VHDL 电路描述方法、VHDL 的编译与仿真以及芯片的选择。本书选取的工程实例有数据总线控制器的设计、图像快速傅立叶变换芯片的设计、数值控制振荡器的设计、基于 6502 框架的 8 位微处理器芯片设计以及高精度数字信号处理芯片的设计。

通过本书的学习，读者除了可以掌握 VHDL 程序设计的基础知识外，还可以了解到如何应用 VHDL 技术具体设计一个电路芯片。本书专业性和实用性较强。

本书适合初、中级读者，电路设计人员和广大培训人员阅读和参考。

电子电气设计与自动控制系列

VHDL 应用开发技术与工程实践

◆ 编　　著　求是科技
责任编辑　张立科

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
读者热线　010-67132692
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：18.25
字数：441 千字　　2005 年 1 月第 1 版
印数：1 – 6 000 册　　2005 年 1 月北京第 1 次印刷

ISBN 7-115-12718-2/TP • 4270

定价：32.00 元

前　言

VHDL 语言是目前超大规模集成电路设计中不可缺少的工具之一，它在集成电路设计领域中有着广泛的应用。随着我国芯片制造业的壮大和芯片设计能力的提高，诸如 VHDL 这类电路描述语言的应用会越来越广泛。

本书共分 12 章，前 6 章讲解了 VHDL 语言的基础知识和芯片设计的常用技术，后 6 章讲解了利用 VHDL 设计电路芯片的工程实践案例。

第 1 章向读者介绍了数字系统设计的基本准则。本章是集成电路设计、数字设计的基础，重点介绍了电路对数值的表示方法以及电路设计方法。

第 2 章向读者介绍了 VHDL 基础知识。本章主要向读者讲解了什么是 VHDL，以及它的主要特性和基本概念。重点讲解了利用 VHDL 进行电路建模、描述的方法，VHDL 逻辑综合和常见的 VHDL 设计单元。

第 3 章向读者介绍了 VHDL 的数据类型与运算操作符。全章为 VHDL 描述的客体、VHDL 的数据类型和 VHDL 运算操作符 3 个部分，它们是 VHDL 描述电路的基本组成，同时完成了电路基本功能的描述。

第 4 章向读者介绍了 VHDL 电路描述的常用方法。本章在第 3 章的基础上，向读者全面介绍了信号的使用、进程结构描述的方法、VHDL 变量的使用、电路描述控制语句的使用方法、VHDL 描述的风格及规范，最后笔者结合自己的理解浅谈了对 VHDL 的使用经验。

第 5 章向读者介绍了如何利用 MAX+plus II 编译 VHDL 电路描述文件，并利用它提供的工具对电路描述进行仿真以验证设计是否正确。本章介绍了 MAX+plus II 编译、仿真的流程，并详细讲解了它的编译和仿真的配置，最后简要介绍了其他的编译、仿真工具。

第 6 章向读者介绍了如何选用芯片并最终实现电路设计。本章介绍了如何将描述程序烧制到实际芯片中，并比较了 CPLD 和 FPGA 的异同。

第 7 章向读者介绍了基本逻辑电路的设计。本章包括基本组合逻辑电路的设计、基本时序逻辑电路的设计以及组合与时序逻辑的综合设计。所有设计示例均可在 MAX+plus II 上完成编译并实现仿真，也可作为实际开发中的参考。

第 8 章向读者介绍了数据控制器的设计。本章包含了 8 位数据总线控制器的设计、USB 2.0 接口控制器的设计和 8 位存储器控制器的设计。数据控制器是实际工程开发中最常用的模块，本章详细地讲解了这 3 种数据控制器的设计流程，并通过 MAX+plus II 编译、仿真直观地显示了数据控制器的基本功能。

第 9 章向读者介绍了图像快速傅立叶变换（FFT）芯片的设计。FFT 是数字图像处理芯片中常用的模块之一，本章全面介绍了该芯片的设计流程以及编译、仿真步骤。

第 10 章向读者介绍了数值控制振荡器（NCO）芯片的设计。NCO 是工业控制和数字信号处理中最常用的波形发生器，其输出波可用作载波、调制波和音频的基波。本章介绍了可输出正弦、余弦以及锯齿波的振荡器，并讨论了不同精度条件下的波形情况。

第 11 章向读者介绍了基于 6502 框架的 8 位微处理器芯片的设计。微处理器芯片的设计

具有一定的难度和广泛大的使用群体，本章介绍了在6502框架下的8位微处理器的设计流程，并讨论了它的编译和仿真过程。

第12章向读者介绍了高精度数字/模拟转换器、PWM电路设计及扩展应用。本章切合工程实际应用，详细介绍了DAC和PWM的具体实现过程，并讨论了一阶Delta-Sigma转换器与PWM的扩展应用。

全书主要由刘健编写，其他参加编写工作以及提供帮助的人员有张宏林、David、周志喜、朱静等。本书代码可登到http://www.cs-book.com下载。

由于作者水平有限，书中难免有不足和疏漏之处，恳请读者朋友和各位同仁批评指正，不胜感激！

编者

2005年1月

目 录

第 1 章　数字系统设计的基本准则

1.1　数字电路与模拟电路

1．模拟电路简介

在工程应用中，为了测量、传递和处理某些物理量，常通过传感器反这些物理量转成与之成比例的电压（或电流）值，这些值在一定范围内是连续变化的。这些电信号表示或模拟了实际的物理量，故称之为模拟信号。处理模拟信号的电路称为模拟电路（Analog Circuit），其准确的定义是实现模拟信号放大、变换、处理和产生等功能的电路总称。如图 1-1 所示的是一个简单的滤波放大电路，它包含了三极管 Q1、电阻 R1 和 R2、电容 C1 和 C2、电感 L1，其中 Vin 为电路的交流输入，Vout 为电路的交流输出。

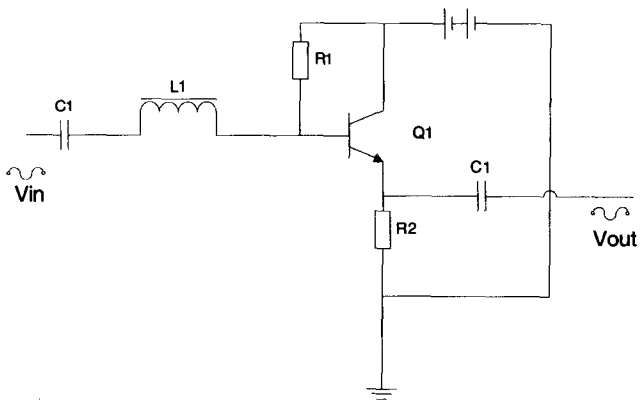


图 1-1　模拟电路

高性能的模拟电路元器件及电路参数对外界条件都有比较严格的要求，例如在图 1-1 中的三极管 Q1，电路设计者通常需要对它进行调节，使其工作在线性区，起到放大电流的作用，保证该放大电路的正常工作。放大电路是模拟电路中最基本的单元电路。放大电路一般包含具有非线性特性的三极管或集成放大器件，它们需要直流提供静态工作点，而被放大的是交流信号。所以，在模拟电路工作时，既有直流信号又有交流信号，既有线性元件工作又有非线性元件工作，既需要静态分析又需要动态分析。

2．数字电路简介

数字量是离散的，只能按有限的或可数的量化单位取值。与数字量相对应的电信号称为数字信号，处理数字信号的电路称为数字电路（Digital Circuit）。数字电路最基本的电路单元

是门电路（Gate），这些基本门电路包含有与门、或门、非门以及它们的有限组合形成的门电路。如图 1-2 所示为与门、或门、与非门以及或非门的电路符号。

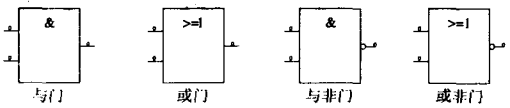


图 1-2　门电路符号

数字电路包括信号的传送、控制、记忆、计数、产生、整形等内容，它在结构、分析方法、功能、特点等方面均不同于模拟电路。数字电路比较简单、抗干扰性强、精度高、便于集成，本书将围绕如何利用 VHDL 设计并描述数字电路系统这一主题，向读者系统介绍其原理、步骤、方法和调试经验。

1.2　数值的表达

数字系统和计算机常常采用二进制数，编写程序主要使用十六进制操作数，而在日常生活中大量使用的是十进制数。所以在数字系统设计和编辑程序中，经常需要实现不同进制操作数之间的相互转换。不管操作数是二进制、十进制还是十六进制，都可以利用等式（1.1）来表示。

$$\begin{aligned}(N)_R &= (b_{n-1}b_{n-2}\cdots b_1 b_0 . b_{-1}b_{-2}\cdots b_{-m})_R \\ &= b_{n-1}\times R^{n-1} + b_{n-2}\times R^{n-2} + \cdots + b_1 \times R^1 + b_0 \times R^0 \\ &+ b_{-1}\times R^{-1} + b_{-2}\times R^{-2} + \cdots + b_{-m}\times R^{-m} \\ &= \sum_{i=-m}^{n} b_i R^i\end{aligned} \tag{1.1}$$

该表达式采用了位置记数法，其中 N 表示操作数在当前进制下的数值，R 表示进制数，b 表示在当前进制下操作数各位的数值（0～n-1 为整数位；-1～-m 为小数位）。

1.2.1　二进制基础

二进制数制系统是数字电路设计的基础，这是因为二进制数的基数为 2，只有 0 和 1 两个数字，运算规则简单，便于电路实现。本节我们来讨论有关二进制数的属性、用法及一些基本概念。

二进制数采用位置记数法，每位的权是 2 的幂（如表 1-1 所示），利用等式（1.1）得到等式（1.2），即二进制数的一般表达形式。

$$\begin{aligned}(N)_2 &= (b_{n-1}b_{n-2}\cdots b_1 b_0 . b_{-1}b_{-2}\cdots b_{-m})_2 \\ &= (b_{n-1}\times 2^{n-1} + b_{n-2}\times 2^{n-2} + \cdots + b_1 \times 2^1 + b_0 \times 2^0 \\ &+ b_{-1}\times 2^{-1} + b_{-2}\times 2^{-2} + \cdots + b_{-m}\times 2^{-m})_{10}\end{aligned} \tag{1.2}$$

表 1-1　　二进制各位的权

二进制整数位位数	权值	十进制表示值	二进制小数位位数	权值	十进制表示值
8	2^7	128	1	2^{-1}	0.5
7	2^6	64	2	2^{-2}	0.25
6	2^5	32	3	2^{-3}	0.125
5	2^4	16	4	2^{-4}	0.0625
4	2^3	8	5	2^{-5}	0.03125
3	2^2	4	6	2^{-6}	0.015625
2	2^1	2			
1	2^0	1			

例如利用等式（1.2）可将二进制数$(11010.11)_2$表示为十进制数。

$$\begin{aligned}(11010.11)_2 &= (1\times2^4+1\times2^3+1\times2^2+1\times2^1+1\times2^0\\&\quad+1\times2^{-1}+1\times2^{-2})_{10}\\&=(26.75)_{10}\end{aligned}$$

1.2.2　十六进制基础

二进制数的表达数值所用的位数较多，特别是那些较大的数值。为解决二进制数表达上的缺点，常采用以 2 的幂为基数的八进制数或十六进制数来表达数值。八进制有 8 个数：0、1、2、3、4、5、6、7；十六进制有 16 个数：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F（其中 A 为 10，B 为 11，C 为 12，D 为 13，E 为 14，F 为 15）。

十六进制数的一般表达形式为：

$$\begin{aligned}(N)_{16} &= (b_{n-1}b_{n-2}\cdots b_1b_0\bullet b_{-1}b_{-2}\cdots b_{-m})_{16}\\&=(b_{n-1}\times16^{n-1}+b_{n-2}\times16^{n-2}+\cdots+b_1\times16^1+b_0\times16^0\\&+b_{-1}\times16^{-1}+b_{-2}\times16^{-2}+\cdots+b_{-m}\times16^{-m})_{10}\end{aligned}$$

1.2.3　二进制和十六进制编码

1. 有符号数编码

正负数可采用多种编码方式表示，连同符号位在一起作为一个数，称为机器数；而它对应的数值为机器数的真值。常用的编码方法有原码、反码和补码，其中以补码形式进行编码的数最为常用。

(1) 原码

正数的符号位用 0 表示，负数的符号位用 1 表示，这种表示方法为原码，如：

X= +100　　$[X]_{原}$=01100100B=64H

X= - 100　　$[X]_{原}$=11100100B=E4H

原码的最高位为符号位，去除符号位，余下的位表示数的绝对值。所以正数的原码，只需将正数用二进制表示，高位用 0 补齐。例如 100 用原码表示为 64H，若用 16 位表示则为 0064H。对于负数的原码，因为仅和对应的正数的原码最高符号位不同，因而将对应的正数

原码的最高符号位改为 1 即可。

(2) 反码

正数的反码表示与原码相同，这个基本概念常被忽视，如：

$$[+100]_{反}=01100100B=64H$$

而负数的反码则为它对应的正数的反码按位取反（包括符号位），如：

$$[+45]_{反}=00101101B=2DH$$

$$[-45]_{反}=11010010B=D2H$$

(3) 补码

正数的补码表示和原码相同，而负数的补码为对应的正数补码按位取反后在最低位加 1，如：

$$[+115]_{补}=01110011B=73H$$

$$[-115]_{补}=10001101B=8DH$$

若用 16 位二进制表示补码，则：

$$[+115]_{补}=0000000001110011B=0073H$$

$$[-115]_{补}=1111111110001101B=FF8DH$$

从上面的讨论可以看出，反码可理解为对“1”求补，补码则对“2”求补。一字节的正数的反码与它对应的负数反码求和为 FFH，而补码为 100H。正由于补码具有这种特性，在进行减法运算时，可将减数用补码表示，与被减数相加，进位自然丢失，其和为运算的正确结果。为加深对补码的理解，我们举一个浅显的十进制数减法的示例，如：

$$9-8 = 9+(-8) = 9+(10-8) -10$$

$$= 9+2-10 = 1$$

其中对于十进制来说，-8 的补码为+2，9+2 产生的进位不能保留在个位上，与 10 自然抵消，得到正确的计算结果，这就是自然丢失进位的原理。二进制运算也是这样，例如：

$$78H-25H=78H+[-25H]_{补}=78H+DBH=53H$$

其计算过程如下：

```
  0 1 1 1 1 0 0 0
+) 1 1 0 1 1 0 1 1
-----------------
 [1] 0 1 0 1 0 0 1 1
  ↑
  └── 自然丢失
```

若两数相减无借位，则结果可能为负数，如：

$$25H-78H=25H+[-78H]_{补}=25H+88H=ADH$$

其计算过程如下：

```
  0 0 1 0 0 1 0 1
+) 1 0 0 0 1 0 0 0
-----------------
  1 0 1 0 1 1 0 1
```

2. 编码溢出的解决

电路在进行加减运算时，凡是有符号的数一般用补码表示。寄存器的字长一般都有

一定的限制，它存储的有符号数限定在一定范围内，若字长为 8 位，数值范围是+127～-128。若运算结果超过这个表达范围，称为溢出。解决溢出的方法是将参加运算的数进行符号扩展，用多字节来表示。例如两个正数 7AH 和 52H 相加，由于产生溢出，得出的结果为负数。

$$7AH + 52H = CCH$$

为了保证结果的正确性，可以将这两个数表示成 16 位数，即 7AH 表示成 007AH；52H 表示成 0052H，因此：

$$007AH + 0052H = 00CCH$$

得到了正确的结果。

假设两个负数-112（90H）和-30（E2H）相加，其运算表达式为：

$$90H + E2H = 72H$$

由于产生溢出，得出的结果为正数。若将它们分别用 16 位补码表示，则运算表达式为：

$$FF90H + FFE2H = FF72H$$

由此可见，扩展正数（或无符号数）时，扩展位全为 0；扩展负数时，扩展位全为 1。

3．格雷码

格雷码（Gray Code）有许多种，其共同的特点是任意相邻码之间只有一位不同。如表 1-2 所示为典型格雷码的编码顺序，0 和最大数（2^n-1）之间只有一位不同，是一种循环码。格雷码在代码形成和传输时引起的误差较小。在角度-数字转换器中，输入的角度为模拟量，输出为数字量。相邻数字量之间必然存在边界，对应这些边界，某些数位会出现不稳定的跳变状态（也称模糊状态）。例如，7 的二进制码是 0111，8 的二进制码是 1000，在 7 和 8 之间的边界上，四位二进制位均出现模糊状态，使得角度-数字转换器会输出 0000 或 1111，出现较大的误差。若角度-数字转换器改用格雷码方案，7 的格雷码是 0100，8 的格雷码是 1100，在 7 和 8 的边界上，仅在最高位出现模糊状态，误差不大于 1。

表 1-2　　格雷码与二进制码、十六进制码的对照表

十进制数	二进制码	十六进制码	格雷码
	B3 B2 B1 B0	H0	R3 R2 R1 R0
0	0 0 0 0	0	0 0 0 0
1	0 0 0 1	1	0 0 0 1
2	0 0 1 0	2	0 0 1 1
3	0 0 1 1	3	0 0 1 0
4	0 1 0 0	4	0 1 1 0
5	0 1 0 1	5	0 1 1 1
6	0 1 1 0	6	0 1 0 1
7	0 1 1 1	7	0 1 0 0
8	1 0 0 0	8	1 1 0 0
9	1 0 0 1	9	1 1 0 1
10	1 0 1 0	A	1 1 1 1
11	1 0 1 1	B	1 1 1 0
12	1 1 0 0	C	1 0 1 0
13	1 1 0 1	D	1 0 1 1
14	1 1 1 0	E	1 0 0 1
15	1 1 1 1	F	1 0 0 0

1.2.4 进制间的数值转换

1. 十进制数转换为二进制数

整数部分的转化方法可以归纳为“除 2 取余”；而小数的转化方法为“乘 2 取整”。例如十进制数 21，将其转化为二进制数的步骤为：

```
2 | 21
  2 | 10   余 1
    2 | 5    余 0
      2 | 2    余 1
        2 | 1    余 0
          0      余 1
```

在除 2 的过程中，最先得到的余数为二进制的最低位，最后得到的余数为二进制的最高位。因此十进制数 21 转化为二进制数的结果是 $(21)_{10} = (10101)_2$。

再如十进制小数 0.625，依据“乘 2 取整”规则，将最先得到的整数作为小数部分的最高位，最后得到的整数作为小数的最低位。转化步骤为：

```
  0.625
+)    2
------
  1.250
+)    2
------
  0.500
+)    2
------
  1.000
```

注意：乘 2 之后的整数位不再参加运算。

因此 0.625 转化为二进制数的结果是 $(0.625)_{10} = (.101)_2$。

同理可得 $(21.625)_{10} = (10101.101)_2$。

2. 二进制数转换为十六进制数

整数部分的转化规则是，从小数点开始向左数，每 4 位二进制数用对应的十六进制数表示，不足 4 位的补零；小数部分的转化规则是，从小数点开始向右数，每 4 位二进制数利用对应的十六进制数表示，不足 4 位的补零。以二进制数（10101.101）$_2$ 的转化为例，其转化为十六进制数的结果为 $(10101.101)_2 = (15.A)_{16}$。

3. 十进制数转换为十六进制数

这种转化的方法有两种：第 1 种是先将十进制数转换成二进制数，再将二进制数转换成

十六进制数；第 2 种是利用“除 R 取余”（整数部分）和“乘 R 取整”（小数部分）的规则，可以将十进制数化成 R 进制数（本节中 R=16）。例如将十进制数 235 转化为十六进制数，其步骤为：

```
16 | 235
16 |  14      余 B
      0       余 E
```

转化结果为 $(235)_{10}=(EB)_{16}$。

4. VHDL 的数值表达方式

在 VHDL 描述文件中，十六进制数用“X”标记，二进制数用“B”标记，八进制数用“O”标记，如表 1-3 所示。

表 1-3　　VHDL 中数制表示的示例

数制	示例
二进制	B“11000”
八进制	O“456”
十六进制	X“FFA3”
十进制	236
实数	5.4E-4

1.3　电路设计方法

在计算机辅助电子系统设计出现之前，人们采用传统的硬件电路设计方法，即自底向上（Bottom Up）的设计方法。自底向上的电路设计方法是根据系统对硬件的要求，写出详细的技术规格书，并画出系统的控制流程图。根据技术规格书和系统控制流程图，对系统功能进行细化，合理地划分功能模块，画出系统的功能框图。对各个功能模块进行细化和电路设计。各个功能模块电路设计、调试完成后，将各个功能模块的硬件电路连接起来再进行系统地调试，最后完成整个系统的硬件设计。

随着大规模专用集成电路（ASIC）的开发和研制，开发效率不断提高，同时还增加了已有开发成果的可重用性以及缩短开发时间，其中各个 ASIC 研制和生产厂家相继开发出适用于自己的硬件描述语言。其中最具代表性的是美国国防部开发的 VHDL 语言（VHSIC Hardware Description Language），Verilog 公司开发的 Verilog HDL，使利用硬件描述语言（HDL）的电路设计方法成为主流。

1.3.1　自顶向下的设计方法

传统的电路设计方法都是自底向上的，而基于 EDA 技术的自顶向下的(TOP-DOWN)设计方法正好与其相反，其步骤就是采用可完全独立于目标器件芯片物理结构的硬件描述语言，如 VHDL，在系统的基本功能或行为级上对设计的产品进行描述和定义，结合多层次的仿真技术，在确保设计的可行性与正确性的前提下，完成功能确认。然后利用 EDA 工具的逻辑