

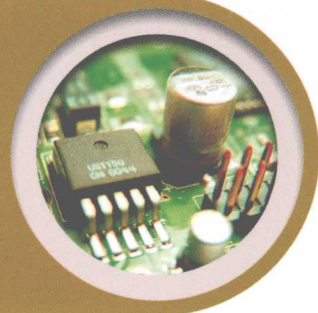


Altium Designer
电路设计全攻略

Technology
实用技术

FPGA

设计



张义和◎编著

苏蕾◎编修



NLIC2970906721



科学出版社

Altium Designer 电路设计全攻略

FPGA 设计

张义和 编著
苏 蕾 编修



NLIC2970906721

科学出版社
北京

图字: 01-2012-6573 号

内 容 简 介

Altium Designer 所提供的电路原理图绘图功能(简称电路绘图),一直都是领先群雄,它提供各种电路图结构的设计,包括单张式电路图、平坦式电路图、阶层式电路图,以及高效能的重复阶层式电路图,等等。

本书的主要目的是探讨 FPGA 设计,包括以 VHDL 为主的数字逻辑基础能力训练与应用技巧,并搭配 NanoBoard 3000 进行系统设计。

本书内容丰富、结构合理、图文并茂、语言清晰。适合各大中型院校电工、电子、自动化及相关专业师生参考阅读,同时适合作为电路设计工程师的参考用书。

原中文繁體字版《電路設計實習——FPAG 篇》(張義和編著)由新文京開發出版股份有限公司於 2010 年出版,本中文簡體字版經該公司授權獨家在大陸地區出版、發行、銷售。

图书在版编目(CIP)数据

FPGA 设计/张义和编著;苏蕾编修. —北京:科学出版社,2013. 7

(Altium Designer 电路设计全攻略)

ISBN 978-7-03-037603-9

I. F… II. ①张…②苏… III. 印制电路-计算机辅助设计-应用软件
IV. TN410. 2

中国版本图书馆 CIP 数据核字(2013)第 115237 号

责任编辑:孙力维 杨 凯 / 责任制作:魏 谨

责任印制:赵德静 / 封面设计:画道设计

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京佳艺恒彩印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2013 年 7 月第 一 版 开本: B5(720×1000)

2013 年 7 月第一次印刷 印张: 18 1/2

印数: 1—3 000 字数: 360 000

定 价: 52.00 元

(如有印装质量问题,我社负责调换)



前言

“电子电路设计自动化”简称 EDA,主要是利用计算机辅助电路设计(Computer-Aided Design,CAD),而计算机辅助电路设计已从电路板设计,延伸到以 FPGA 为主的零件设计。而零件毕竟是零件,到底还是要建立在电路板上,所以,电路板设计是不可或缺的! Altium Designer 将 FPGA 与电路板设计集成在一起,不管是设计零件,还是设计电路板,都在同一个环境下,使用类似的工具与方法设计,称得上一绝!

认识 Altium Designer

Altium Designer 的前身是大家熟识的 Protel,这些年来,Altium Designer 不断地蜕变、成长,早已跃升为电路设计的一线品牌!更是经济衰退以来,节省成本、提升效率的不二选择。Altium Designer 是一套全功能的电路设计软件,而其各部分的特色简介如下:

● 电路原理图绘图

Altium Designer 所提供的电路原理图绘图功能(后简称电路绘图),一直都是领先群雄,即便如此,也没有减缓其前进的脚步。基本上,Altium Designer 提供各种电路图结构的设计,如单张式电路图、平坦式电路图、阶层式电路图,以及高效能的重复阶层式电路图(多通道设计),此外还积极推展 OpenBus 系统的电路绘图设计。而电路绘图的目的,除了可平顺地将电路绘图延伸到电路板设计、电路仿真外,还可作为 FPGA(Field Programmable Gate Array)设计的工具等,可说是无所不能!

Altium Designer 向来以功能多、工具多著称!例如,原本就非常强大的搜索/浏览功能,现在更进一步推展到零件信息管理系统(Component Information System,

CIS), 让我们的设计能连接到供货厂商, 以了解市场状况, 迅速掌握最新零件信息, 不但可摆脱不切实际的设计, 还能让设计、制造与备料一次完成。

● 电路板设计

在电路板设计方面, Altium Designer 所提供的交互式布线、总线布线、差分线对布线, 足以满足各种电路板设计, 包括高速板。再通过其 CAM 的功能, 产生电路板制造所需的底片文档、钻孔文档等, 还能连接到其下的 CAMtastic, 以进行底片文档的编辑/排版等。而其搜索/浏览功能, 早已跳脱传统, 可由 3D 功能, 达到互动展示与搜索的效果。特别值得一提的是, Altium Designer 在 3D 显示上, 有重大突破, 可平顺地与机械结构设计结合, 从而成为机电合一的高效能设计软件。

● FPGA 设计

Altium Designer 一直在 FPGA 设计领域默默耕耘, 可通过其超强的电路绘图功能与集成式设计环境(Integrated Development Environment, IDE), 应用电路图、硬件描述语言(Hardware Description Language, HDL)、C 语言、OpenBus 等方式设计 FPGA。另外, 也可在 FPGA 设计中放入 CPU, 并进行嵌入式(Embedded)设计, 包括组合语言与 C 语言。

Altium Designer 还提供相当不错的 FPGA 仿真功能, 除了可进行 HDL 的软件仿真(除硬件)外, 还可通过其自家的 NanoBoardII、NB3000 及其他厂商所提供的硬件或自行开发的 FPGA 电路板, 进行实际仿真, 以验证 FPGA 的功能。比较特殊的是, Altium Designer 应用其电路板设计优势, 让 FPGA 设计与电路板设计完美结合, 使零件设计(FPGA)、电路板设计一次完成。若有变更设计, 可以直接由 FPGA 反应到电路板, 或由电路板反应到 FPGA。整个设计流程简化且更有保证。

本书架构

本书主要目的是探讨 FPGA 设计, 包括以 VHDL 为主的数学逻辑基础能力训练与应用技巧, 并搭配 NanoBoard 3000 进行系统设计。

● 快速逻辑门实验

本章的目的是让第一次接触 Altium Designer 中 FPGA 设计的人, 能从创建 FPGA 工程开始, 然后以简单的 VHDL 描述各式逻辑门, 并能通过测试平台, 快速仿真电路的波形反应。通过本章的学习, 读者除了能逐渐熟悉 Altium Designer 的 FPGA 设计模式外, 还能认识 VHDL 基本架构, 以及基本的语法。

● 加/减法器实验

本章将介绍如何使用 VHDL 描述加/减法器, 并介绍并行执行与顺序执行的区别, 还有 Process 结构与 For 语句的应用。在设计环境方面, 将进一步介绍测试平台

与波形视窗。

● 组合逻辑电路设计

本章从认识常用的 VHDL 语句与语法开始,再将它们分别应用于编码器、解码器、多路分解器与多路复用器等的设计,并进行仿真与验证。

● 时序逻辑电路设计

本章从认识顺序执行的相关语句,如常用的 if 语句及其应用开始。在此还探讨 Component-Map 语句、Function 与 Procedure 等子程序结构。而在实例应用上,将介绍如何设计三态组件、触发器及计数器等,特别在计数器方面,着墨颇深。

● 串/并转换电路设计实例

从本章起,即进入实例演练的阶段,而在此将探讨串/并转换电路,每个范例都详述其工作原理与设计概念。完成电路设计后,随即进行仿真验证,并分析其波形。

● PWM 电路设计实例

本章将从认识 PWM 开始,再用 VHDL 描述一个 PWM 电路,并进行仿真与信号分析,最后,还将介绍如何从触发方式上提升电路的效能。

● 键盘扫描电路设计实例

本章将探讨键盘组的工作原理,再用 VHDL 描述键盘组的扫描电路,并进行仿真与验证。此外,还将针对按钮开关所产生的回跳问题,提出改善方法。

● 七节显示器扫描电路设计实例

本章从七节显示器模块的结构与工作原理开始介绍,然后针对其工作原理,用 VHDL 描述四位数七节显示器模块的驱动电路。最后再仿真其动作,并分析验证其正确性。

● LED 驱动电路设计

本章将进入系统设计阶段,并配合 NanoBoard3000,让我们设计的 RGB 型 LED 驱动电路,快速展现在实体电路上。

● 人机界面与 Script

本章将延续第 9 章的设计,并加入人机界面设计,让我们的设计具有较佳的操作接口。而在人机接口的设计上,将引用 Altium Designer 所提供的 Script 设计工程,在此工程里,将提供市面上常用的 Script 语言,如 Delphi、VB、Java 等,我们可选用熟悉的 Script 语言来撰写人机接口。

● 霹雳灯电路

本章将利用 Altium Designer 所提供的 PWM 电路(免费 IP),快速构建霹雳灯的

驱动电路,而在驱动信号方面,则用 Script 设计工程产生扫描信号,驱动 8 组 PWM 电路,达到 256 阶的 LED 控制。

● 再谈霹雳灯电路

本章还是以霹雳灯电路为例,利用 SOC 方式,将 32 位 CPU 置入 FPGA 芯片里,再撰写系统软件,以实现电路功能。在此,我们将应用 OpenBus 系统设计方式,构建系统硬件。再以嵌入式设计工程,构建系统软件,并利用软件平台(Software Platform),快速加载驱动程序,整个设计过程非常精彩。

张义和 谨识

● 随书 DVD 内容简介

在随书 DVD 光盘中,包含每章的教学辅助用 PPT 文件,其内容精美实用。光盘中还包含每章章末即时练习所用到的参考文档资料和选择题答案。



目 录

第 1 章 快速逻辑门实验	1
1.1 VHDL 的设计与仿真	1
1.2 工程与文档管理	2
1.3 Altium Designer 的 VHDL 编辑环境	4
1.4 VHDL 的基本认识	5
1.4.1 VHDL 的结构	5
1.4.2 信号类型	9
1.4.3 数据类型	10
1.4.4 VHDL 的运算	11
1.5 电路设计与仿真	14
1.6 即时练习	19
第 2 章 加/减法器实验	23
2.1 认识二进制加/减法器	23
2.1.1 以逻辑运算描述加/减法电路	24
2.1.2 以算术运算描述加/减法电路	30
2.2 并行执行与顺序执行	36
2.3 Process 与 For 循环的应用	37
2.4 测试平台再研究	40
2.5 波形窗口的操作	42
2.6 即时练习	46

第 3 章 组合逻辑电路设计	49
3.1 认识常用的 VHDL 语句	49
3.1.1 When-Else 语句	49
3.1.2 With-Select 语句	50
3.1.3 Case-When 语句	51
3.2 编码器的原理与实现	52
3.2.1 编码器的实现	52
3.2.2 优先编码器的实现	54
3.3 解码器的原理与实现	56
3.3.1 解码器的实现	56
3.3.2 七节显示码解码器的实现	58
3.4 多路分解器的原理与实现	60
3.5 多路复用器的原理与实现	62
3.6 全加器的实现	64
3.7 两位数二进制乘法器的实现	67
3.8 即时练习	70
第 4 章 时序逻辑电路设计	73
4.1 认识常用的 VHDL 语句	73
4.1.1 If-Then-Else 语句	73
4.1.2 时钟信号的描述	75
4.1.3 Generate 语句	76
4.1.4 Loop 语句	77
4.1.5 Block 语句	78
4.2 认识 Component 与 Port-Map 语句	79
4.3 认识子程序	82
4.3.1 认识 Procedure	82
4.3.2 认识 Function	82
4.4 三态、栓锁器与触发器	84
4.4.1 三态控制	84
4.4.2 栓锁功能	85
4.4.3 触发器	86
4.5 逻辑方式设计计数器	88
4.5.1 递减计数器	88
4.5.2 递增计数器	94

4.5.3 递增/递减计数器	96
4.6 算术方式设计计数器	98
4.6.1 递增计数器	100
4.6.2 递增/递减计数器	102
4.6.3 除 N 计数器	103
4.6.4 可预置计数器	104
4.6.5 BCD 计数器	105
4.7 即时练习	107
第 5 章 串/并转换电路设计实例	111
5.1 串/并转换电路设计	112
5.1.1 设计目标	112
5.1.2 电路设计	115
5.1.3 仿真与验证	118
5.2 并/串转换电路设计	121
5.2.1 设计目标	121
5.2.2 电路设计	124
5.2.3 仿真与验证	127
5.3 I ² C 串行通信电路设计	130
5.3.1 传输电路设计与仿真	133
5.3.2 接收电路设计与仿真	141
第 6 章 PWM 电路设计实例	151
6.1 认识 PWM	153
6.2 PWM 电路设计	156
6.3 仿真与验证	159
6.4 电路效能升级	163
第 7 章 键盘扫描电路设计实例	165
7.1 认识键盘组与扫描原理	166
7.2 防回跳设计	173
7.3 键盘扫描电路设计	175
7.4 仿真与验证	179
7.5 “放开后动作”的仿真与验证	181

第 8 章 七节显示器扫描电路设计实例	185
8.1 认识七节显示器模块	186
8.2 七节显示器扫描电路设计	192
8.3 仿真与验证	195
8.4 另类设计	198
第 9 章 LED 驱动电路设计	201
9.1 设计新概念与 NB3000	202
9.2 快速构建 GBB LED 控制电路	203
9.3 实现电路功能	212
9.4 外接 LED 与后记	216
第 10 章 人机界面与 Script	219
10.1 认识仪控面板	219
10.2 快速 Script 工程设计	220
10.3 实现电路功能	232
第 11 章 霹雳灯电路	237
11.1 认识霹雳灯电路	237
11.2 构建霹雳灯电路	239
11.3 人机界面与信号配置	248
11.4 描述驱动信号	254
11.5 实现电路功能	260
第 12 章 再谈霹雳灯电路	263
12.1 认识嵌入式系统与 SOC	263
12.2 构建系统硬件	265
12.2.1 认识 OpenBus 编辑环境	266
12.2.2 OpenBus 系统设计	267
12.2.3 系统电路设计	271
12.2.4 连接 NB3000	277
12.3 构建系统软件	278



第1章 快速逻辑门实验

1.1 VHDL 的设计与仿真

数字电路的设计已从传统电路设计方式,走向以 HDL(硬件描述语言,Hardware Description Language)为核心的 EDA(电子电路设计自动化,Electronic Design Automation)时代。在工具软件与计算机的辅助下进行设计,即所谓的 CAD(Computer Aided Design),让电路设计更有效率,还能进一步产生电路制造/组装所需的各项数据,即所谓的 CAM(Computer Aided Manufacturing),以及测试电路所需的各项数据,即 CAT(Computer Aided Testing),CAM 与 CAT 合称为 CAM&T。

在 EDA 时代,计算机当然是最基本的工具,此外,还需要高度集成的工具软件与适用的设计方式。基本上,设计方式可以采用图式(电路图方式)来设计,也可以采用 HDL 的方式,至于什么是“适用的设计方式”?见仁见智。笔者个人认为,若要设计较大系统,则采用图式设计较为方便;若要进行细部设计,则采用 HDL 方式较为方便;若是数字逻辑的教学,则 HDL 方式效率较高,因为 HDL 的设计方式,就像在撰写程序,再配合功能性仿真,即可快速连接 HDL 与电路的关系。而 HDL 的种类也不少,在此将它区分为两大类。

① 标准型号的 HDL:此类 HDL 是指通过 IEEE(电机电子工程师协会,Institute of Electrical and Electronic Engineers)认证的 HDL,如 VHDL(高速集成电路硬件描述语言,Very High Speed Integrated Circuit Hardware Description Language)、Verlog HDL 等,大部分 EDA 软件都支持这类 HDL,且没有半导体厂商的限制。

② 特定厂商的 HDL:此类 HDL 是由半导体厂商所开发的硬件描述语言,通常

是针对该厂商所生产的芯片而设计,至于其他厂商的芯片是否完全适用,就不得而知了,如 Altera 公司(半导体厂商)的 AHDL(Altera HDL)等。另外,通用型 EDA 软件也可能不能接受/支持这类 HDL,而只有该半导体厂商所提供的 EDA 软件才能使用这类 HDL。若要使用 AHDL,则必须在 Altera 的 MAX+plus II 或 Quartus II 里使用。

本书以基本数字逻辑教学为目的,采用 VHDL 方式,在 Altium Designer 里,进行各项数字逻辑的教学与实验。整个流程如图 1.1 所示,其中分为 4 个阶段,说明如下:



图 1.1 VHDL 设计与仿真流程

① 创建新工程与文档:较新的设计方式都是采用工程(Project)管理,在 Altium Designer 里,不管是进行电路板设计、FPGA 设计,还是 VHDL 仿真,也都是采用工程管理。因此,在进行 VHDL 设计与仿真之初,必先创建新工程,然后在该工程里,新建顶层(Top-level)电路图文档,及 VHDL 文档。

② 以 VHDL 设计电路:在 Altium Designer 的 VHDL 编辑环境里,编辑 VHDL,以定义

所要设计的电路,就像在写程序一样。

③ 构建测试平台:测试平台(Test Bench)的功能是为了测试所设计的电路,而定义的激励信号与时序。基本上,测试平台也是由 VHDL 所构建,Altium Designer 提供快速产生测试平台的方法,我们只要加入部分关键词句即可。

④ 仿真与分析:VHDL 仿真是根据测试平台,追踪我们所设计电路的信号,并描绘其波形,即可分析输入输出信号的关系,以验证所设计 VHDL 电路的功能性。

1.2 工程与文档管理

当我们要开启 Altium Designer 时,则按 **开始** 按钮拉出“开始”功能表,再选取“所有程序”→“Altium Designer Summer 09”→“Altium Designer Summer 09”选项,即可开启 Altium Designer。紧接着,按下列步骤操作:

① 启动“File”→“New”→“Project”→“FPGA Project”命令,屏幕左边的 Projects 面板里将出现一个 FPGA_Project1.PrjFpg 项目。

② 启动“File”→“New”→“Schematic”命令,则 Projects 面板里,FPGA_Project1.PrjFpg 项目的下面将出现一个 Sheet1.SchDoc 项目。

③ 启动“File”→“New”→“VHDL Docurement”命令,则 Projects 面板里,FPGA_Project1.PrjFpg 项目的下面将出现一个 VHDL1.Vhd 项目,同时进入 VHDL 编辑环境。





④ 前面三个步骤所产生的工程与文档,尚未保存,并非真实的文档。紧接着,指向 Projects 面板里的 FPGA_Project1.PrjFpg 项目(不要指在左边的图示上),按鼠标右键拉出菜单,再选取“Save”命令,屏幕出现图 1.2 所示保存对话框,其中所要保存的是“电路图”文档。这时,请在对话框上方的“保存在”栏位中,指定所要保存的位置。在此强烈建议,每个工程都保存在专属的文件夹里。按对话框里的  钮,即可新增文件夹。



图 1.2 指定文档名与路径

⑤ 在“文件名”栏位里指定所要保存的电路图文档名(在此指定为 Gates),再按  钮即可将电路图保存;同时切换到下一个保存对话框,其中要求我们保存 VHDL 文档。在“文件名”栏位里指定所要保存的 VHDL 文档名(在此保持为 VHDL1),在此要特别注意,VHDL 的文档名不可与顶层电路图的文档名一样,再按  钮即可将 VHDL 存档。

⑥ 两个文档保存完成后,即切换到工程的保存对话框,在文档名栏位里指定所要保存的工程名称。在此要特别注意,工程的文档名必须与顶层电路图的文档名一样,所以在此指定为 Gates,再按  钮即可将文档存档,而保存结果将反映在 Projects 面板里,如图 1.3 所示。

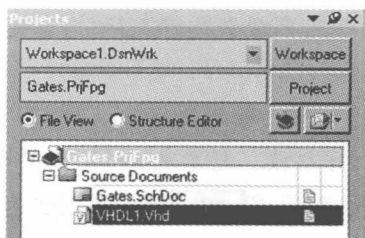


图 1.3 完成工程存档

1.3 Altium Designer 的 VHDL 编辑环境

接续前一节,我们所面对的是最后一次开启的 VHDL 编辑环境,如图 1.4 所示。基本上,VHDL 的编辑环境就是一个文字编辑器,而这个文字编辑器具有 VHDL 关键字的判别能力(以颜色区分),而在大部分文字编辑器里所具有的自动缩排与改变字形等功能,在此不但有,还更好用。

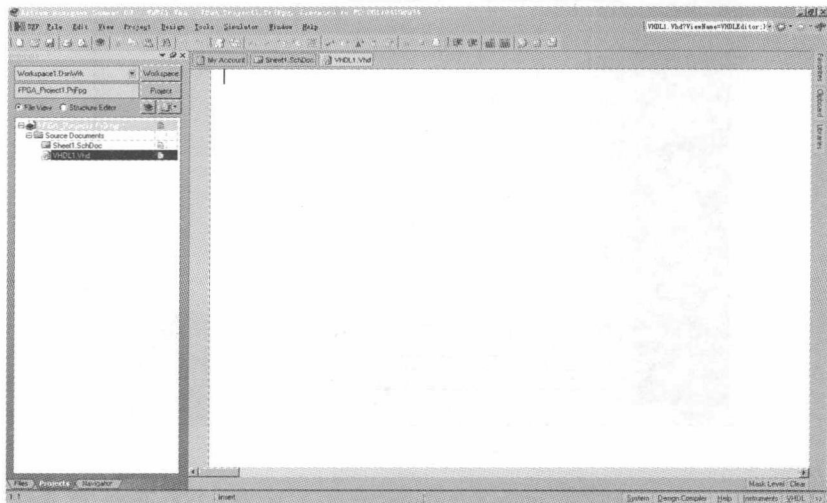


图 1.4 Altium Designer 的 VHDL 编辑环境

如果读者的屏幕解析度较高,在编辑区里所输入的文字较小,可能会造成判读的困扰。这时可改变字形的设定,让其中输入的文字变大。若要改变编辑器的字形设定,则启动“Tools”→“Schematic Preferences...”命令,在随即开启的对话框里,选取左边 Text Editors 下的 Display 项目,切换到 Display 页,如图 1.5 所示。

目前设定的字形是“Curier New,9pt”,稍微小了一点。按 钮开启字形设定对话框,在其中的大小栏位里指定“16”,再按 钮退回前一个对话框。最后按 钮关闭该对话框,完成设定,此后编辑区里所显示的文字,将比较清楚。

除了字体大小的设定外,在设计电路(输入 VHDL)时,最好注意下列事项。

① 善用注解:在 VHDL 里若要放置注解文字(可使用中文),则连续输入两个(或更多)负号,其右边文字将被视为注解文字,而不予编译/合成。

② 多利用定位键():与一般程序语言一样,为了容易解读,VHDL 常会使用缩排以及定位(对齐),让程序易读。基本上,VHDL 可视为设计硬件电路的程序语言,当然也要注意解读的问题,而使用 键可达到快速定位与对齐的效果。

③ 大小写:在 VHDL 里的语句,并不会区分字母的大小写,不过,我们还是要养

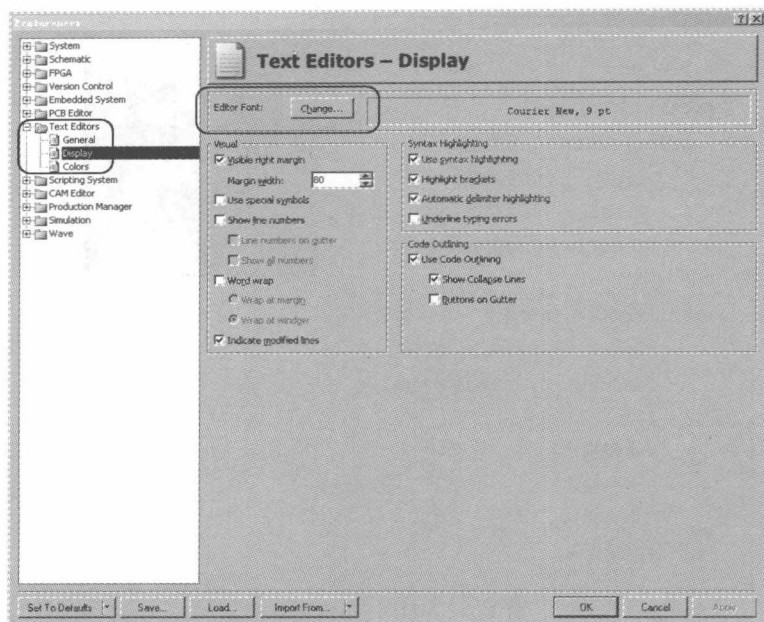


图 1.5 文字编辑器的 Display 页

成大小写一致的习惯,不要让同一个关键字或变量,出现不同的大小写。

④ 插入/覆盖模式:通常在描述电路时,都是以插入模式(Insert)输入,我们可在编辑区下方中间的状态栏位里看到“Insert”,表示目前是在插入模式。当按 **Ins** 键时,可切换插入与覆盖模式。若原本状态栏位显示“Insert”,按 **Ins** 键将切换为“Overwrite”,也就是覆盖模式;若原本状态栏位显示“Overwrite”,按 **Ins** 键将切换为“Insert”,也就是插入模式。

1.4 VHDL 的基本认识

VHDL 是一个完整的描述语言,就像是程序语言。因此,VHDL 具有其特定的结构、数据类型(信号类型)、数据种类、运算与语法(叙述),很难在一个章节中交代清楚。我们将根据需要,分别在各章节中介绍,以减少读者负担。在本节里将探讨 VHDL 的结构,以及较基本的概念,如数据类型、数据种类与运算等。

1.4.1 VHDL 的结构

由于 VHDL 的内容就是要描述一个电路,所以其结构与电路完全对应,如图 1.6 所示。

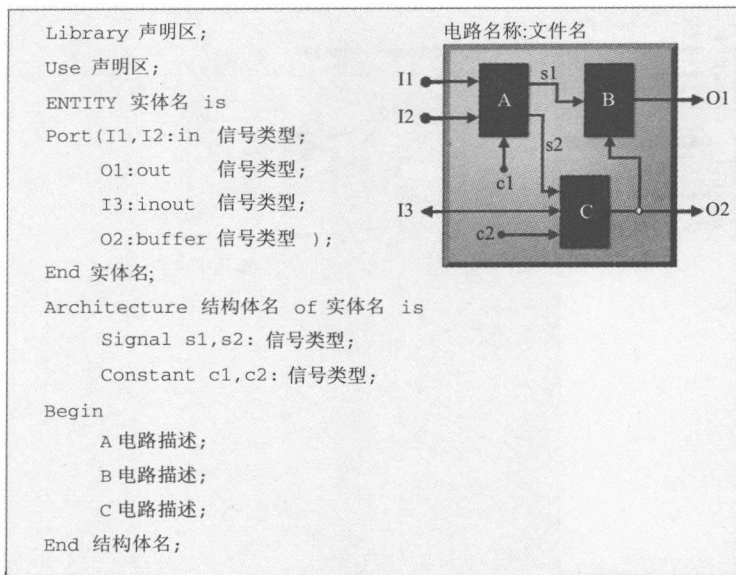


图 1.6 VHDL 结构示意图

1. 零件库的声明

在 VHDL 语句的开始处,以 Library 关键字为首,其右边是所要引用的零件库(文件夹)。例如,Library IEEE,就是声明接下来所要引用的程序包(Package)取自 IEEE 零件库,而其中字母的大小写并不影响。

2. 引用程序包的声明

在 Library 声明之后,可能会有一个或多个 Use 关键字,其右边是所要引用程序包的声明,而程序包包括各种设计实体都能共享的信号类型、常数、运算、函数(Function)及程序(Procedure)等。其声明的格式为

```
Use A. B. C;
```

其中,A 为零件库名称,B 为程序包名称,C 为程序包子项目。例如,

```
Use ieee.std_logic_1164.all;
```

表示将引用 IEEE 零件库里的 std_logic_1164 程序包文档(std_logic_1164.vhd 文档)中的所有程序包(all 代表所有程序包子项目)。在 IEEE 零件库里,提供三个常用的程序包文档,除了 std_logic_1164.vhd 文档外,还有 std_logic_unsigned.vhd 及 std_logic_arith.vhd,每个文档的文档名都很长,不管有没有引用其中的程序包,我们都会一起声明如下: