

应用电子教育系列

ABEL 硬件程序设计

刘毅坚 编著

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

EDA 技术是建立在电子电路自动化设计基础上综合多学科知识的新型应用技术。本书以加强大学生素质教育为宗旨,结合大量实例和系列实验,阐述 ABEL 硬件描述语言,指导学生进行电路设计,详细介绍 EDA 技术的设计原理、过程和方法。

本书共分 10 章,前 4 章是 ABEL 语言程序设计基础知识;第 5~7 章介绍 ABEL 语言在 EDA 技术开发工具中的使用方法及优化设计的有关知识;第 8 章介绍硬件程序设计的系统知识和实例;第 9 章介绍 ABEL 语言在换代产品中的应用;第 10 章为不同层次的读者提供基础类和开发类实验,分别培养学生掌握常规实验的基本技能,启发学生完成实用电路设计。

本书可作为高等院校计算机应用、电子电器及自动化控制等本、专科专业的教材,也可作为相关工程技术人员从事应用电路设计的技术参考资料。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ABEL 硬件程序设计/刘毅坚编著. —北京:电子工业出版社,2004.3

(应用电子教育系列)

ISBN 7-5053-9684-6

. A... . 刘... . 电子电路—电路设计:计算机辅助设计 . TN702

中国版本图书馆 CIP 数据核字(2004)第 012494 号

责任编辑:邓小瑜 特约编辑:吴 华

印 刷:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:15.75 字数:403.2 千字

印 次:2004 年 3 月第 1 次印刷

印 数:5 000 册 定价:24.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077。质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

EDA (Electronics Design Automation) 即电子设计自动化。EDA 技术是以计算机硬件和系统软件为基本工作平台, 综合多学科的最新科研成果研制成商品化的新型应用技术。EDA 技术是电子 CAD 的高级阶段, 利用电子设计自动化技术可直接设计出系统所需的集成电路。在逻辑芯片和数字电路设计上, 传统的“固定功能模块+连线”的设计方法已逐步退出历史舞台。在 EDA 软件工具的支持下, 设计者只需完成对系统功能的描述, 就可以由计算机软件帮助完成电路的功能设计、逻辑设计、性能分析、时序测试, 乃至印刷电路板 (PCB) 的自动设计工作, 并得到预知的结果。基于芯片设计的技术正在成为从事电子系统设计方法的主流。

目前, EDA 技术应用于电子产品开发的全过程, 在电子产品生产的过程中都期望由计算机借助软件设计技术来完成各种设计, 即所谓“硬件软化”工作。EDA 技术实验已被应用于大专院校电子类专业的课程设计、毕业设计各类电子设备的制作, 也是教师从事教学和科学研究的理想开发工具。

现场可编程门阵列 FPGA (Field Programmable Gates Array) 和复杂可编程逻辑器件 CPLD (Complex Programmable Logic Device) 芯片等, 具有体积小、容量大、I/O 口丰富、价格低、功耗小、编程容易等优点, 深受数字电路设计者青睐。这类产品的出现, 使得传统的数字电路设计方法和过程产生革命性的飞跃。EDA 走进普通实验室, 实现了基于芯片的设计方法, 可利用 EDA 技术把从前需要几个机箱安装的设备集成在一块电路板上。目前, 大专院校、机电一体化等研究机构和工厂, 都将 EDA 技术的研究开发作为市场竞争的对象, 应用于家电、智能玩具和现场控制设备等广阔的发展空间。

可编程器件的设计离不开 EDA 软件。Lattice 公司的 ISP 技术比较成熟, ispLSI 器件采用 UltraMOS 技术的 E²PROM 存储单元, 能重复编程 10000 次以上, ispLSI 系列器件可以简单地与 PC 并口用编程电缆相连进行编程和下载, 并受众多的开发工具软件的支持。ISP Synario 是其中的一种软件包, 它包括 Data I/O 和 Lattice pDS+Fitter 适配器软件。设计的输入和仿真由 Synario 软件提供, 设计工具由 pDS+Fitter 适配器提供。用户可以通过设计适配控制参数对编译过程实行控制。Synario 软件有两种基本的输入手段: 电路原理图和 ABEL 硬件描述语言。它还有一个功能仿真器, 可以用报告形式和波形输出观察仿真结果。专业版 Synario 还支持 VHDL 行为描述语言和 Verilog 的输入和仿真。

硬件描述语言 (HDL) 是一种用形式化方法来描述数字电路和数字逻辑设计的语言。它支持各种行为输入方式, 包括方程、真值表及状态图。用户能用 ABEL 进行设计, 验证实验结果, 而无需考虑较少涉及的目标器件的结构。

书中针对 ABEL 语言、涉及 Synario 编译控制及部分设计实例进行了相关的描述, 借助 GYH— 型电子实验仪, 结合一些具体实验内容进行教学, 帮助读者学习有关电路原理图、真值表、状态图设计, 以及仿真测试、适配下载等方面的知识。硬件程序设计语言是一门实践性很强的技术课程, 编写此书的目的是介绍通过程序设计的方法, 达到高效地完成硬件设计过程的计算机技术, 引导学生完成一些传统的电子设计技术 (包括单片机) 不能实现的内容, 从而突出这一现代电子设计技术的优势。

全书由作者统一编写完成，刘玥负责校对。本书全稿由湖南大学邱光谊教授、国防科技大学邹蓬兴教授亲自审定，王惠刚教授、彭民德教授等也为本书的编排及出版方面提供了宝贵意见，在此一并致谢！

由于书中部分内容由作者直接翻译和整理，虽已应用于部分大学的教学实践，限于编著者水平，书中错误在所难免，恳请广大读者批评指正！

目 录

第 1 章 ABEL 语言基础	(1)
1.1 硬件程序设计环境	(1)
1.1.1 基于芯片的开发过程	(1)
1.1.2 可编程系统简介	(2)
1.1.3 源文件编辑窗口	(3)
1.1.4 Synario 文本编辑工具	(3)
1.1.5 文件属性	(4)
1.1.6 使用 ABEL 语言的其他环境	(5)
1.2 ABEL 语言基本概念	(7)
1.2.1 ABEL 语言的基本内容	(7)
1.2.2 语言描述	(7)
1.2.3 ABEL 语言的特点	(8)
1.3 ABEL 基本组成元素	(8)
1.3.1 语法基础	(8)
1.3.2 运算符	(11)
1.3.3 表达式与方程	(13)
1.4 ABEL 其他组成元素	(14)
1.4.1 集合 (数组)	(14)
1.4.2 块	(15)
1.4.3 变量及变量代换	(16)
1.5 ABEL 基本语法结构	(16)
1.5.1 源文件模块结构	(16)
1.5.2 基本语法规则	(19)
1.5.3 模块语句和标题语句	(19)
1.5.4 声明 (DECLARATIONS)	(20)
1.5.5 模块结束	(23)
习题	(23)
第 2 章 ABEL 的特殊组成	(25)
2.1 宏定义与模块使用	(25)
2.1.1 MACRO 宏定义	(25)
2.1.2 FUNCTIONAL_BLOCK 功能块	(26)
2.1.3 INTERFACE 接口	(26)
2.2 逻辑功能描述	(27)
2.2.1 EQUATIONS 方程	(27)
2.2.2 真值表定义	(28)

2.2.3	真值表声明	(29)
2.2.4	真值表结构	(29)
2.2.5	七段 LED 译码器设计实例	(29)
2.3	状态图和状态机	(31)
2.3.1	状态图的认识	(31)
2.3.2	STATE_DIAGRAM 状态图语句	(32)
2.3.3	STATE_REGISTER 状态机语句	(32)
2.3.4	状态转移	(33)
2.3.5	状态图设计实例	(36)
2.4	异或因子和测试向量	(37)
2.4.1	XOR_FACTORS 异或因子	(37)
2.4.2	库文件和熔丝文件	(38)
2.4.3	测试和跟踪	(39)
2.4.4	测试向量表实例	(40)
	习题	(42)
第 3 章	指示字和点扩展	(44)
3.1	指示字的使用	(44)
3.1.1	常用指示字	(44)
3.1.2	特别指示字	(46)
3.2	触发器和寄存器	(49)
3.2.1	触发器	(49)
3.2.2	寄存器	(50)
3.2.3	内部节点信号	(50)
3.3	ABEL 点扩展方法	(51)
3.3.1	管脚属性定义	(51)
3.3.2	点扩展语法	(51)
3.3.3	广义点扩展和狭义点扩展	(52)
3.3.4	点扩展的典型应用	(53)
	习题	(56)
第 4 章	ABEL 程序设计基础	(57)
4.1	原理图的层次描述	(57)
4.1.1	项目管理器	(57)
4.2	ABEL 语言的层次描述	(60)
4.2.1	一位全加器电路的模块结构	(61)
4.2.2	一位加法器逻辑	(61)
4.2.3	语言模块层次描述	(62)
4.3	三位乘法器电路	(65)
4.3.1	三位乘法器电路设计基本原理	(65)
4.3.2	程序模块设计	(65)
4.3.3	ABEL 程序模块分段说明	(65)

4.4	汽车尾灯控制电路	(67)
4.4.1	汽车尾灯控制电路设计原理	(67)
4.4.2	汽车尾灯控制电路顶层原理图	(68)
4.4.3	模块描述源程序	(69)
	习题	(69)
第5章	Synario 开发软件	(71)
5.1	Synario 系统简介	(71)
5.1.1	项目管理器	(71)
5.1.2	十进制计数器	(72)
5.2	创建新项目	(73)
5.2.1	选择主菜单选项	(73)
5.2.2	选择目录	(74)
5.2.3	选择项目和器件	(74)
5.3	原理图设计	(75)
5.3.1	相关的知识	(75)
5.3.2	原理图编辑工具	(75)
5.3.3	选择新原理图	(76)
5.3.4	添加新器件	(77)
5.4	基本逻辑器件设定	(77)
5.4.1	选择基本逻辑器件	(77)
5.4.2	信号命名	(77)
5.4.3	输入、输出信号标记	(78)
5.4.4	管脚锁定文件的使用	(80)
5.4.5	增加文字说明	(80)
5.5	逻辑功能描述	(81)
5.5.1	选择文本编辑对话框	(81)
5.5.2	描述逻辑功能	(81)
5.5.3	修改	(82)
5.5.4	错误信息	(82)
5.6	测试向量文件	(83)
5.6.1	建立仿真测试向量 (Simulation Test Vectors)	(83)
5.6.2	测试向量文件书写要点	(83)
	习题	(84)
第6章	属性控制与编译	(85)
6.1	参数文件控制方式	(85)
6.1.1	参数文件定义	(85)
6.1.2	参数说明	(85)
6.1.3	实例	(86)
6.1.4	参数文件输入	(87)
6.2	命令控制方式	(87)

6.2.1	设置节点相关属性	(88)
6.2.2	设置系统相关属性	(90)
6.2.3	设置输入输出属性	(93)
6.3	菜单控制方式	(94)
6.3.1	Synario 原理图逻辑设计	(94)
6.3.2	Synario 逻辑适配属性	(96)
6.4	电路优化设计	(97)
6.4.1	逻辑设计密度的优化	(97)
6.4.2	逻辑资源利用率的优化	(97)
6.4.3	速度优化	(98)
6.4.4	逻辑设计注意事项	(98)
6.5	编译与仿真	(99)
6.5.1	编译过程	(99)
6.5.2	波形仿真	(100)
	习题	(102)
第7章	适配下载	(103)
7.1	逻辑适配	(103)
7.1.1	基本适配参数设定	(103)
7.1.2	适配属性参数设定	(104)
7.1.3	适配过程	(104)
7.1.4	查看设计报告	(105)
7.2	可编程开关	(106)
7.2.1	熔丝型开关	(106)
7.2.2	反熔丝型开关	(106)
7.2.3	浮栅编程元器件	(107)
7.3	编程下载	(108)
7.3.1	选择编程下载项目	(108)
7.3.2	读板上芯片的电子标签或写电子标签	(109)
7.3.3	执行下载操作	(110)
7.4	下载接口	(110)
7.4.1	ISP 编程接口	(111)
7.4.2	器件编程组态	(111)
	习题	(113)
第8章	硬件程序设计方法	(115)
8.1	数字系统设计方法	(115)
8.1.1	数字系统的特点	(115)
8.1.2	数字系统的基本模型	(115)
8.1.3	数字系统设计方法	(116)
8.1.4	设计验证	(116)
8.2	ABEL 程序设计技巧	(117)

8.2.1	自顶向下的设计方法	(117)
8.2.2	源程序中的下层程序设计	(117)
8.2.3	上层模块中的声明	(118)
8.2.4	中间变量和状态变换	(118)
8.2.5	运用异或功能简化逻辑	(120)
8.2.6	输出控制	(120)
8.3	移动数字编码器	(121)
8.3.1	移动数字编码器的组成	(121)
8.3.2	下层程序模块	(123)
8.3.3	测试向量程序模块	(128)
8.4	数字钟电路设计	(130)
8.4.1	系统分析设计	(130)
8.4.2	Top-Down 模块设计	(131)
8.4.3	编译与适配	(136)
8.4.4	编程下载	(136)
	习题	(137)
第 9 章	Expert 软件的典型应用	(138)
9.1	Expert 项目管理器	(138)
9.1.1	项目管理器	(138)
9.1.2	项目管理器接口	(139)
9.1.3	进程窗口	(139)
9.1.4	翻译 EDIF 属性	(139)
9.1.5	Expert 资源类型	(139)
9.2	数字脉冲到模拟信号转换的实例	(140)
9.2.1	顶层模块	(140)
9.2.2	下层类模块	(141)
9.2.3	仿真操作	(142)
9.3	Expert 颜色仿真实例	(142)
9.3.1	选择资源项目	(142)
9.3.2	仿真程序	(142)
	习题	(146)
第 10 章	实验指导	(147)
10.1	基础类实验	(147)
10.1.1	实验 1——原理图设计	(147)
10.1.2	实验 2——3-8 译码器	(149)
10.1.3	实验 3——4-16 译码器	(150)
10.1.4	实验 4——四位加法器	(152)
10.1.5	实验 5——四位比较器	(153)
10.1.6	实验 6——十进制计数器	(155)
10.1.7	实验 7——格雷码计数器	(157)

10.1.8	实验 8——交通灯	(159)
10.1.9	实验 9——动态扫描显示	(162)
10.1.10	实验 10——可控移位寄存器	(164)
10.2	开发类实验	(166)
10.2.1	实验 1——单片机通用接口芯片和程序监视器	(166)
10.2.2	实验 2——PWM 脉宽调制的 A/D 转换器	(168)
10.2.3	实验 3——ADC0809 A/D 转换及模拟量显示	(172)
10.2.4	实验 4——PWM 脉宽调制的 D/A 转换器	(176)
10.2.5	实验 5——DAC0832 D/A 转换器及波形发生器	(177)
10.2.6	实验 6——序列脉冲同步码检测	(181)
10.2.7	实验 7——PCM 采编器	(183)
10.2.8	实验 8——利用单片机的波形发生器	(186)
10.2.9	实验 9——利用单片机的 A/D 转换	(189)
10.2.10	实验 10——GYH— 型仪器综合设计实验	(190)
附录 A	Synario 部分错误信息表	(193)
附录 B	CPLD 器件功能简介	(205)
附录 C	常用 ispLSI 芯片简介	(211)
附录 D	GYH— 型电子系统综合实验仪	(216)
附录 E	GYH— 型综合实验仪	(226)
附录 F	部分目标板管脚对照	(231)
附录 G	MAX+PLUSII 软件简介	(234)
附录 H	技术支持	(239)
	参考文献	(240)

第 1 章 ABEL 语言基础

ISP Synario System 是一套完整的逻辑设计系统。采用“在系统可编程”技术可使硬件设计和研制变得像软件那样灵活而易于修改,正所谓达到“硬件软化”的目的。正确地输入设计文件是开发 ispLSI 的关键,由于 pDS 和 ISP Synario System 软件包使用的硬件描述语言都源于 ABEL,所以本章以 ABEL 为核心,结合硬件程序设计环境对逻辑电路的语言描述规则和使用方法作一些简单的介绍。

1.1 硬件程序设计环境

EDA (Electronic Design Automation) 电子设计自动化技术,是一种正在兴起的基于芯片的设计方法。可编程逻辑器件 PLD (Programmable Logic Device) 可直接使用软件平台上的设计结果,从根本上打破软、硬件间的屏障,完成设计文件到硬件电路的直接转换。随着 CPLD/FPLD (复杂/现场可编程逻辑器件) 的出现,并加上 EDA 设计开发工具就构成了一个完整的基于芯片设计的开发系统。

1.1.1 基于芯片的开发过程

目前世界上一些大型的 EDA 软件公司已开发了一些著名的软件,如 orCAD, Cadence, PSPICE 和 Viewlogic 等。大型半导体器件公司典型的开发软件有如:Lattice 公司的 Synario, ALTERA 公司的 MaxplusII 和 Xilinx 公司的 Foundation 等产品。当然,这些公司的开发软件也在随着新的器件和新工艺的出现不断地更新或升级,但较为全面地学习使用一些成熟的开发软件和硬件描述语言,对于正确地认识硬件程序设计的整个过程是十分必要的。对于目前世界上市场占有率较大的 Lattice 公司来说,从器件性能价格比及软件的可靠性等方面,更具有其特别的竞争优势。

ISP Synario System 的设计套件,是美国 Lattice 半导体公司向市场推出的一套基于 Synario 软件环境的可编程系统。它支持 Lattice ispLSI 器件、pLSI 器件、isp GAL 器件及全系列的 GAL 器件的逻辑设计、编译和逻辑仿真的功能,完成适配、下载直到形成产品。Synario 软件包的版本种类很多,常见的有 ISP Synario V5.0, ISP EXPERT/A, ISP Synario V5.1/B, ISP Synario V8.0, ISP EXPERT/B, ISP VHDL Series 7.X 和 ispLEVER 等。尽管不同的软件包在功能上有一些差异,但对 ISP 在系统可编程器件来说,其开发过程并无本质差别,只要精通一种版本,使用其他版本并不困难。目前可结合 ISP Synario V5.1/B 等普通版本的学习,进行较为严格的训练,可为今后进一步掌握 EDA 技术打下一个良好的基础。

使用较为普遍的 Lattice 器件,采用了 ISP Synario System, ISP EXPERT 及 ISP Design EXPERT 及 ispLEVER 等软件,其首选的逻辑描述语言是 Data I/O 公司早已成熟的 ABEL 语言,它与 AHDL 语言有比较接近的描述方法和语言格式。ispLSI 器件受到 IEEE 标准的 VHDL 的支持,因此 Synario 系统是 EDA 技术学习和应用的有效助手。利用可编程器件进行硬件程序设计的流程图如图 1.1 所示,图中简要地介绍了硬件程序的设计过程。以后各章节将结合硬件

程序设计语言的使用特点逐步进行介绍。



图 1.1 硬件程序设计流程图

1.1.2 可编程系统简介

为使读者能在较短时间内掌握硬件程序开发的最新技术，本书以 ISP Synario System 为基础进行重点介绍。ISP Synario System 是强有力的可编程系统，其 Project Navigator（以下称项目管理器）是一种面向对象的管理器。在硬件程序设计中，它将各种源文件联系在一起并能跟踪软件的工作流程，该管理器无论在哪一个设计层面都有十分重要的作用。图 1.2 是项目管理器的工作界面。

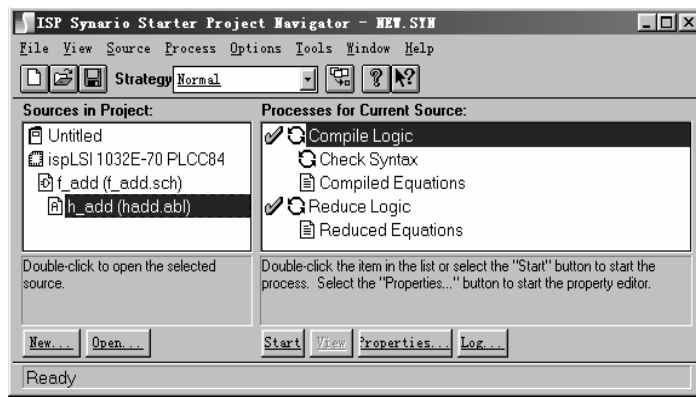


图 1.2 Synario 项目管理器工作界面

项目管理器设置了各类窗口，在项目菜单中安排有各类操作的子命令，帮助用户实现硬件程序的设计、调试和编辑等功能。表 1.1 列出该设计软件的操作命令。

表 1.1 Synario 常用操作命令表

主 菜 单	下 拉 菜 单	主要功能及说明
File	New Project	建立新的设计项目
	Open Project	打开已有的设计项目
	Open Example	打开已有的设计例子
	Close Project	关闭当前的设计项目
	Save	保存当前的设计项目文件
	Save as	用另外的名字保存设计项目文件
	Clean	清除当前的所有设计项目的中间文件
	Clean up All	清除当前的所有设计项目的中间文件和结果文件

主 菜 单	下 拉 菜 单	主要功能及说明
Source	New	建立新的设计源文件 (*.ABL,*.ABV,*.SCH)
	Import	装入已有的设计源文件 (*.ABL,*.ABV,*.SCH)
	Open	打开已有的设计源文件 (*.ABL,*.ABV,*.SCH)
	Remove	从当前设计项目中清除不需要的源文件
Process	Start	启动已知的操作过程
	Force	强制执行指定操作过程
	Force One Level	强制执行指定的单步操作过程
	Log	查阅运行流程记录文件 (AUTOMAKE.LOG)
	Properties	设定或修改适配器控制参数
Windows	Text Editor	打开文本编辑器
	Schematic Editor	打开原理图编辑器
	Report Viewer	打开设计报告文件观察器
	Symbol Editor	打开电路符号编辑器
	Minimize All	已打开的所有窗口最小化

1.1.3 源文件编辑窗口

在使用 ISP Synario System 设计软件时,允许程序 ISP Synario 方便地得到相关窗口的界面。ISP Synario 软件提供以下几类窗口:

文本浏览器 (Synario Report Viewer) 用来浏览报告文件、列表文件和记录文件。

文本编辑器 (Text Editor) 用来编辑 ABEL-HDL 设计文件、测试向量文件和其他文本文件。

原理图编辑器 (Schematic Editor) 用来编辑电路原理图。

符号编辑器 (Symbol Editor) 用来编辑电路信号图形。

波形观察器 (Wave Form Viewer) 用来观察仿真波形。

层次管理器 (Hierarchy Navigator) 用于层次化设计导向。

Synario 文本编辑器可使编辑的文件分别显示在各自的窗口中运行,当激活文件运行时,系统能同时记住大部分菜单项,并使用可视化窗口管理菜单操作。

1.1.4 Synario 文本编辑工具

利用 ABEL 语言进行程序设计,首先接触到的是对源程序的编辑,使用文本编辑器可编辑多种不同类型文件。当一个项目建立以后(以后章节有较为详细的说明),可用文本编辑器对 ABEL 源文件等进行编辑。文本编辑器如图 1.3 所示:

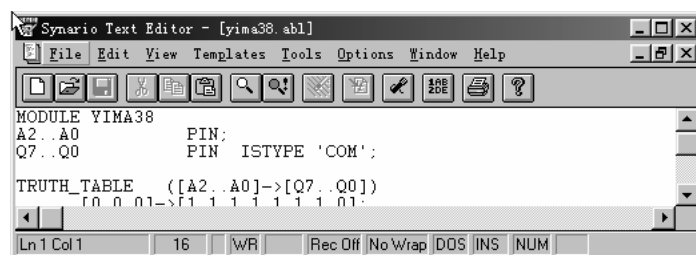


图 1.3 Synario 文本编辑器

Synario 文本编辑器有菜单条、编辑图标工具及文本编辑窗口、提示行等几个区域，下面针对这些工具的内容分别进行简单介绍。

1. 菜单

菜单是实现各种操作的清单。文本编辑器中的菜单，也包括水平菜单（如主菜单）及下拉菜单。主菜单条包括：文件菜单（File）、编辑菜单（Edit）、视图菜单（View）、模板菜单（Templates）、工具菜单（Tools）、选项菜单（Options）、窗口菜单（Window）和帮助菜单（Help）。下面主要介绍文件菜单，一些常用菜单在以后的章节中涉及时逐步介绍，其他未使用的菜单的说明请参考相关的资料及帮助菜单。

文件菜单（File）中包括下列菜单命令，利用这些命令可以对多种类型的文件进行各种不同的操作。其为：

New：打开一个空文件，可用来编辑新文件。

Open：打开一个已存在的文件（如：xxx.ABL 等）。

Close：关闭已激活的文件。

Close All：关闭在文本编辑器中已打开的所有文件。

Save：保存当前文件。

Save as...：用新的文件名保存文件。

Save All：保存所有已打开的文件。

Insert：在光标处插入一个新文件到激活的文件中。

Write：写已激活的文件到另一个文件中。

Name：重新命名已激活的文件名，可将新文件名保存。

Print：打印已激活的文件。

Print Setup：打开打印安装对话框，改变打印的文件名。

File Listing：列出最近打开的大部分文件。

Exit：退出文本编辑。



图 1.4 图标按钮

2. 图标

图标是实现某些特点操作的图形符号。图标经常放在工具栏上，如直接单击图标可以实现对文本编辑的快捷操作。与文件操作菜单相关的图标有：打开文件、保存文件、剪切、复制、粘贴及打印等，Synario 中文件操作图标的功能与 Windows 操作系统常用文本编辑相同，在此不再赘述。图 1.4 示出项目管理器文件操作中涉及的图标按钮。

1.1.5 文件属性

利用项目管理器（Project Navigator）进行编译或其他处理时将产生一系列文件，不同类型的文件有不同的属性和相应的管理方法，文件扩展名用以标识文件属性。在 Synario 中经常使用的文件扩展名有：

.ABL	ABEL 源文件
.ABV	ABEL 测试向量源文件
.SCH	原理图源文件
.SYM	原理图信号文件
.SYN	Synario/ABEL 设计项目文件
.STY	Synario/ABEL 设计项目文件类型文件
.WAV	仿真波形文件
.JHD	层次设计方案文件
.DOC	适配器文档文件
.JED	JEDEC 文件
.RPT	适配器报告文件
.LST	列表文件
.LOG	运行流程记录文件
.OUT	输出报告文件
.ERR	错误报告文件

1.1.6 使用 ABEL 语言的其他环境

目前较为流行的 EDA 开发软件 ISP Design EXPERT 8.10，也是 Lattice 公司研制的一种易于设计、执行、验证并对 Lattice 可编程系列器件进行编程的集成软件。ISP Design EXPERT 与 Synario 的使用环境一样，在一个完整的集成开发系统中，包含形形色色的项目（原理图，ABEL，VHDL 和 Verilog HDL），能完成各类项目设计、编译、适配及优化的操作，以及用做验证报告、逻辑仿真和静态时序分析等工具。进一步说，ISP Design EXPERT 包括了标准的电子设计自动化(EDA)接口。因此，用户可拥有一个集成的 ISP Design EXPERT 个人设计环境。

ISP Design EXPERT 的用法与 Synario 开发工具几乎一样，其项目管理器是 ISP Design EXPERT 的主要接口，是一种提高维护项目元素及进程的集成环境、完善的开发工具。在 ISP Design EXPERT 环境中，同样有原理图编辑(.SCH)、符号编辑器，提供了用于以上编辑的标准信号库；文本编辑器用于编辑基本的文本文件，如 ABEL 文件、测试文件和报告文档文件等。此外，波形编辑器能用鼠标建立波形确立时序关系，波形观察器用来观察仿真结果。Lattice 逻辑仿真器可在进入 Lattice 半导体器件前执行逻辑仿真。

在 ISP Design EXPERT 的项目管理器中，列出了该软件所能支持的 3 种硬件程序描述语言（ABEL，VHDL 和 Verilog HDL）设计环境。ISP Design EXPERT 系统支持混合模式的设计，也就是以最小一个原理图作为顶部项目源，并且一个或更多的项目都是同一种语言资源。几种语言资源互相排斥，在进行项目设计时必须选择其中的一种语言。在第 9 章将通过实例专门介绍这一软件的应用。

此外，ABEL 语言也被其他半导体公司用来为在系统可编程（ISP）器件的设计服务，面向对象的可视化编程向导为用户准备了一个电子电路设计的轻松环境。如 Xilinx 半导体公司包括 CPLD 和 FPGA 的系列产品：XC3000A/L，XC3100A/L，XC4000/E/L/EX/XL/XV/XLA，XC5200，Spartan，Spartan-XL，XC9500，XC9500XL，XC9500XV 等系列器件，普遍采用 Foundation 和 Alliance 开发软件进行编程。Foundation 系列软件支持标准 HDL 设计，也就是包括 ABEL，VHDL 和 Verilog HDL 硬件设计描述语言。从 Xilinx Foundation 软件的项目管理器很容易按照

其列出的设计流程（如图 1.5 所示）进行操作。

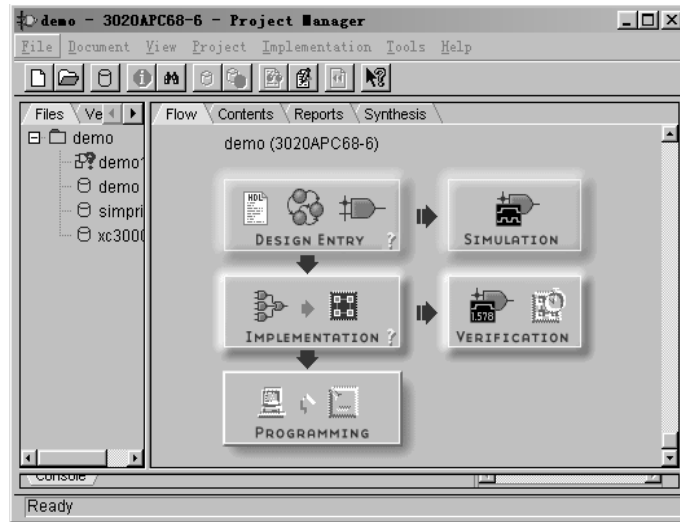


图 1.5 Foundation 软件的项目管理器

从图中可以清晰地看出完成设计任务的全部过程，如在 Xilinx Foundation 的项目设计（Design Entry）组件包括 HDL 语言、状态图和电路原理图的编辑，为硬件设计安排了较为宽松的环境。在该组件中可利用 HDL 语言设计向导设定 ABEL 语言环境，选定以后会出现如图 1.6 所示的对话框。

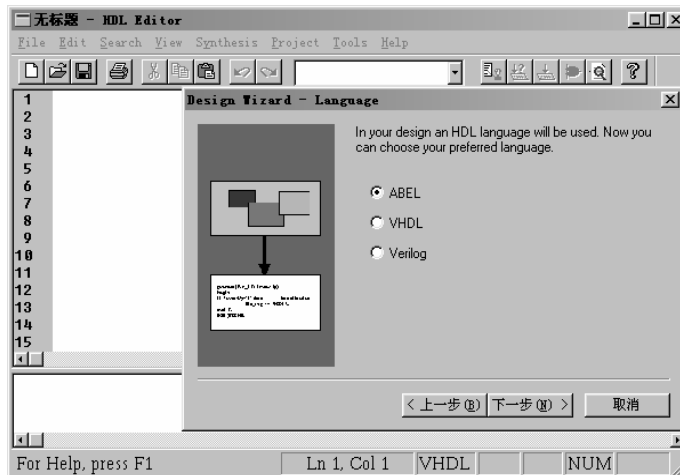


图 1.6 设定 ABEL 语言环境的对话框

由此可以看出，ABEL 语言已在几家著名 ISP 半导体器件公司的 EDA 开发环境中作为首选的硬件设计描述语言。为使读者对 ABEL 语言有一个清楚的认识，下面首先介绍 ABEL 语言的基本概念及组成原理。

1.2 ABEL 语言基本概念

硬件描述语言 (HDL) 的发展至今已有 20 多年的历史, 这类语言已成功地应用于硬件设计的各个阶段。20 世纪 80 年代, 便出现上百种的硬件描述语言, 它们对电子设计自动化起到了促进和推动作用。但是, 这些语言一般各自面向特定的设计领域和层次, 众多的语言使用户无所适从, 因此急需一种面向多领域、多层次, 得到认同的硬件描述语言。通过实践证明, 如: ABEL, AHDL, VHDL 和 Verilog 等语言已处于领先的地位。

Lattice 公司的软件产品 ABEL-HDL 语言和 Xilinx 公司的软件产品 XABEL 及早期 ABEL 语言, 其语法结构基本相同, 在这里简称 ABEL (Advanced Boolean Expression Language)。此类语言都是基于美国 DATA I/O 公司较早开发的一种有完善的逻辑设计功能的硬件描述语言, 它支持 PAL, GAL, CPLD 等各种类型的 PLD 器件。ABEL 是一种支持不同输入形式的结构简单的硬件描述语言。这种语言能直接支持真值表数据输入, 其输入可包括逻辑方程、高级方程、状态方程等多种形式, 容易被电子设计工程师直接用于电子产品的设计。ABEL 的编译器适用于 Synario 等软件开发工具的各版本, 能自动地校验或通过仿真源程序生成不同性质的记录文件。当然, 其设计软件经过标准格式转换文件, 也可转移到其他的设计环境 (如 VHDL 等应用环境)。如果不指定器件去对一个特殊器件作设想是有一定难度的, 一种与结构无关的器件设计 (不包括实际器件和管脚号声明) 比与结构有关的类似器件设计要有更为全面的说明。事实上, 用指定器件进行设计使用起来并不方便。运用 ABEL 语言则可以很方便地描述和完成对上述器件的设计工作。

1.2.1 ABEL 语言的基本内容

ABEL 语言源文件的基本语法包括了下面的一些内容:

支持 ASCII 码字符、标识符、关键字、常量、块、备注、数字、字符串、操作符、表达式和方程等描述方式。能执行的操作有: 逻辑操作、算术操作、关系操作和赋值操作。此外还包括集合及其运算, 变量及其变换的相关内容。

ABEL 语言是一种用形式化方法来描述数字电路和数字逻辑设计的语言, 可对电路进行层次设计, 采用自顶向下的设计方法, 先建立用于设计的行为模型 (行为模型不可以被逻辑综合)。此后, 可以在设计的初期对设计的模型进行行为模拟, 并通过对模拟结果的分析, 逐步调整和修改设计项目。每当设计规模增加时, 设计工作量将按指数级增长, 但自顶向下的设计方法可以很好地解决这个问题。在层次设计中, 上层电路中可以包含下层模块。用 ABEL 语言能对不同的层次结构和行为模型在模块设计中进行具体描述。图 1.7 示出一个一位计数器的设计模块。

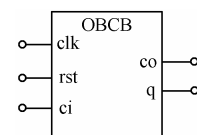


图 1.7 一位计数器的设计模块

1.2.2 语言描述

利用 ABEL 语言可对每个功能模块进行个别的描述。在模块程序文件中, 必须按 ABEL 硬件描述语言规定的格式来书写。例如, 用 ABEL 代码来描述一位计数器逻辑功能的程序模块有如下内容: