



EDA 应用技术

基于 Quartus Prime 的 FPGA / CPLD 数字系统设计实例 (第4版)

● 周润景 南志贤 张玉光 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

EDA 应用技术

基于 Quartus Prime 的 FPGA/CPLD 数字系统设计实例 (第4版)

周润景 南志贤 张玉光 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 Altera 公司全新推出的 Quartus Prime16.1 为设计平台，结合大量的实例来介绍基于 FPGA/CPLD 数字系统的设计方法。书中的例子包含了简单的数字逻辑电路实例、数字系统设计实例及复杂的数字控制系统设计实例，由浅入深地介绍了采用 Quartus Prime16.1 进行数字系统开发的设计流程、设计思想和设计技巧。

本书适合从事数字系统设计的研发人员阅读，也可以作为高等学校电子、通信、自动化等相关专业的教学用书。

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

基于 Quartus Prime 的 FPGA/CPLD 数字系统设计实例 / 周润景, 南志贤, 张玉光编著. —4 版. —北京: 电子工业出版社, 2018.8

(EDA 应用技术)

ISBN 978-7-121-34919-5

I. ①基… II. ①周… ②南… ③张… III. ①可编程序逻辑阵列—系统设计 IV. ①TP303

中国版本图书馆 CIP 数据核字 (2018) 第 191935 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 夏平飞

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

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

开 本: 787×1092 1/16 印张: 30 字数: 768 千字

版 次: 2010 年 8 月第 1 版

2018 年 8 月第 4 版

印 次: 2018 年 8 月第 1 次印刷

定 价: 99.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

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。

本书咨询联系方式: zhang@phei.com.cn。

前 言

随着电子技术、计算机应用技术和 EDA 技术的不断发展,利用 FPGA/CPLD 进行数字系统的开发已经被广泛应用于通信、航天、医疗电子、工业控制等领域。与传统电路设计方法相比,FPGA/CPLD 具有功能强大、开发过程投资少、周期短、便于修改及开发工具智能化等特点。近年来,FPGA/CPLD 市场发展迅速,并且随着电子工艺的不断改进,低成本、高性能的 FPGA/CPLD 器件推陈出新,从而促使 FPGA/CPLD 成为当今硬件设计的首选方式之一。熟练掌握 FPGA/CPLD 设计技术已经是对电子工程师的基本要求。

VHDL 语言是国际标准的硬件描述语言,本书实例中的文本编辑均采用 VHDL 语言编写,并且书中实例均通过仿真和硬件测试。

本书通过实例介绍 Altera 公司最新推出的以 Quartus Prime16.1 为设计平台的 FPGA/CPLD 数字系统设计,由浅入深地介绍数字系统开发的设计流程、设计思想和设计技巧。

Quartus Prime16.1 是 Altera 公司最新推出的革命性的设计软件,其前端用户界面的使用与以前软件版本相比还是同样方便,而在后端增加了 Spectra-Q 引擎,采用一组更快、更易于扩展的新算法,前所未有地缩短了编译时间,提高了设计性能。这一引擎还具有分层数据库,保留了 IP 模块的布局布线,保证了设计的稳定性,同时避免了不必要的时序收敛投入。Quartus Prime 设计软件还在多个 Arria@10 设计上展示了极高的设计性能和设计人员效能。

本书共分为 12 章。第 1 章介绍了利用 Quartus Prime 进行 FPGA/CPLD 设计的开发流程,包括设计输入、约束输入、综合、布局布线、仿真、编程与配置等。第 2 章介绍了 Quartus Prime 的使用,包括原理图编辑、文本编辑和混合编辑的设计方法。第 3~7 章介绍了 VHDL 硬件描述语言,以及简单的数字电路实例,包括门电路、组合逻辑电路、触发器、时序逻辑电路以及存储器的设计方法。第 8 章介绍了课程设计中涉及的数字系统设计实例,使读者熟练掌握 Quartus Prime 的使用方法和 VHDL 语言。第 9 章介绍了宏模块及 IP 核的使用方法和简单的应用实例。第 10 章介绍了 ModelSim、TimeQuest、SignalTap II 与 Quartus Prime 的结合使用方法及简单的应用实例,VHDL 硬件调试,以及在 VHDL 设计当中使用库模块的方法。第 11 章和第 12 章介绍了两个大型的数字系统的设计实例,使读者能更深入地掌握数字系统的设计方法。

本书由周润景、南志贤和张玉光编著。其中,南志贤编写了第 1 章和第 2 章,张玉光编写了第 3 章和第 4 章,周润景编写了第 5~12 章。全书由周润景负责统稿。参加本书编写的还有邵绪晨、李楠、邵盟、冯震、刘波、李艳、崔婧、陈萌、井探亮、丁岩、李志和任自鑫。

由于作者水平有限,书中难免存在错误和不足之处,敬请读者批评指正。

编 著 者

目 录

第1章	数字集成电路概述与 Altera Quartus Prime 开发流程	1
1.1	数字集成电路概述	1
1.2	Quartus Prime 软件综述	7
1.3	设计输入	13
1.4	约束输入	15
1.5	综合	20
1.6	布局布线	24
1.7	仿真	27
1.8	编程与配置	32
第2章	Altera Quartus Prime 的使用	38
2.1	原理图和图表模块编辑	38
2.2	文本编辑	58
2.3	混合编辑（自底向上）	67
2.4	混合编辑（自顶向下）	72
第3章	VHDL 硬件描述语言	78
3.1	VHDL 语言简介和优点	78
3.2	VHDL 语言设计实体的基本结构	79
3.3	VHDL 语言要素	84
3.4	VHDL 顺序语句	97
3.5	VHDL 并行语句	106
3.6	VHDL 子程序	115
3.7	VHDL 的描述风格	119
3.8	状态机的设计	126
第4章	门电路设计范例	132
4.1	与非门电路	132
4.2	或非门电路	135
4.3	异或门电路	137
4.4	三态门电路	140
4.5	单向总线缓冲器	142
4.6	双向总线缓冲器	144

第 5 章	组合逻辑电路设计范例	147
5.1	编码器.....	147
5.2	译码器.....	153
5.3	数据选择器.....	159
5.4	数据分配器.....	163
5.5	数值比较器.....	164
5.6	加法器.....	166
5.7	减法器.....	172
第 6 章	寄存器、存储器、锁存器和触发器的 VHDL 描述	179
6.1	寄存器.....	179
6.2	移位寄存器.....	182
6.3	只读存储器 (ROM).....	190
6.4	随机存储器 (RAM).....	192
6.5	堆栈.....	194
6.6	FIFO.....	198
6.7	锁存器.....	200
6.8	RS 触发器.....	202
6.9	JK 触发器.....	204
6.10	D 触发器.....	207
6.11	T 触发器.....	209
第 7 章	计数器、信号发生器和分频器的 VHDL 描述	212
7.1	计数器.....	212
7.2	可变模计数器.....	224
7.3	顺序脉冲发生器.....	228
7.4	序列信号发生器.....	230
7.5	分频器.....	231
第 8 章	数字系统设计范例	247
8.1	数字系统的基本结构.....	247
8.2	数字系统的设计方法.....	247
8.3	数字系统设计的一般过程.....	249
8.4	数字系统的设计准则.....	250
8.5	数字系统设计范例.....	252
第 9 章	可参数化宏模块及 IP 核的使用	313
9.1	ROM、RAM、FIFO 的使用.....	313
9.2	乘法器、锁相环的使用.....	320
9.3	正弦波信号发生器.....	323

9.4 NCO IP 核的使用	325
第10章 Quartus Prime 的深度使用	331
10.1 使用 ModelSim 波形编辑器对 VHDL 设计进行仿真	331
10.2 TimeQuest 时序分析仪的用法	341
10.3 SignalTap II 嵌入式逻辑分析仪的使用	349
10.4 VHDL 硬件设计调试	363
10.5 在 VHDL 设计当中使用库模块	381
第11章 基于 FPGA 的射频热疗系统的设计	395
11.1 肿瘤热疗的生物学与物理学技术概论	395
11.2 温度场特性的仿真	398
11.3 射频热疗系统设计	398
11.4 系统硬件电路设计	399
11.5 软件实现	410
11.6 温度场测量与控制的实验	437
第12章 基于 FPGA 的直流电动机伺服系统的设计	442
12.1 电动机控制发展情况	442
12.2 系统控制原理	443
12.3 算法设计	445
12.4 系统硬件设计原理	447
12.5 系统软件设计原理	453
12.6 系统调试及结果分析	464
参考文献	469

第1章 数字集成电路概述与 Altera

Quartus Prime 开发流程

1.1 数字集成电路概述

近年来,电子设计的飞速发展使得数字电路的研究及应用出现了广阔的发展空间。使用现场可编程门阵列(Field Programmable Gate Array, FPGA)或复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD)的设计软件可以将设计好的程序烧写入FPGA器件中,如同自行设计集成电路一样,可节省电路开发的费用与时间。

1.1.1 数字集成电路的分类

尽管本书的学习目标是如何利用可编程逻辑器件实现要求的设计功能,但是考察可供选择的各种器件对数字系统的设计者来说还是有益的,因为它有助于我们更好地理解所有可供选择的方案,同时也可以意识到虽然描述数字系统和数字电路的方法及用来实现这些方法的技术在不断变化,但是基本原理并没有改变。

现代数字系统中所使用的数字电路几乎都是集成电路。使用集成电路实现系统功能比使用分立元件具有电路体积小、可靠性高等优点。从20世纪60年代开始,数字集成电路在集成度方面的发展经历了以下四个阶段:①包含数十到数百个逻辑门的小规模集成电路(Small Scale Integration, SSI);②包含数百到数千个逻辑门的中规模集成电路(Medium Scale Integration, MSI);③包含数千到数万个逻辑门的大规模集成电路(Large Scale Integration, LSI);④包含数万个以上逻辑门的超大规模集成电路(Very Large Scale Integration, VLSI)。在工作原理方面,数字集成电路又可以划分为标准逻辑器件、微处理器和专用集成电路。

1. 标准逻辑器件

标准逻辑器件在集成度方面属于中小规模集成电路。它包括各种逻辑门、触发器、译码器、多路选择器、寄存器和计数器等器件。标准逻辑器件有三种主要类型:TTL、CMOS和ECL。TTL是一种成熟的技术,新的系统设计已经很少采用TTL逻辑器件,但是正在运行的系统中仍然包含这种器件。CMOS器件是当前最流行的标准逻辑器件,它的优点是功耗低。ECL器件主要用于高速系统中。

作为传统数字系统设计中使用的主要器件,标准逻辑器件已经使用了40多年。标准逻辑器

件的产量很大, 生产成本低廉, 价格便宜。如果设计不很复杂, 这些器件仍然是很实用的。标准逻辑器件对于研究数字系统基本构成模块的工作原理具有重要的意义, 它在许多基础的理论 and 实验教学课程中仍然占据重要的位置。

标准逻辑器件由于集成度较低, 采用它们设计数字系统需要较多的器件, 这就使得电路连线复杂, 系统的可靠性降低。由于用户无法修改这类器件的功能, 修改系统设计必须通过对电路重新设计和组装来实现。

2. 微处理器

数字技术已经进入众多的领域, 其中数字计算机是最著名和应用最广泛的产品。尽管计算机影响了人类生活的许多方面, 但是许多人并不完全知道计算机能干些什么。简单地说, 计算机是一个能完成算术运算、逻辑运算、数据处理和做出判断的数字系统。个人计算机(PC)是最常见的计算机, 它由一些数字集成电路芯片组成, 这些芯片包括微处理器芯片、存储器芯片及输入/输出(I/O)接口芯片等。在大多数情况下, 凡是人能做的, 计算机都能做, 而且计算机还能干得更快、更精确。尽管事实上计算机每次只能完成所有计算中的一步, 但是计算机完成每一步的速度非常快, 它的高速度弥补了它的低效率。

计算机依靠所运行的软件(程序)来完成工作。这个软件是人们给计算机的一组完整的指令, 指令告诉计算机其操作的每一步应该干什么。这些指令以二进制代码的形式存储在计算机的存储器中, 计算机从存储器中一次读取一条指令代码, 并完成由指令代码指定的操作。

通过编写软件可以控制计算机完成不同的工作, 这个特点使得设计灵活性得到提高。当修改系统设计时, 设计者只需要改变软件, 不需要或者较少需要修改电路连线。由于计算机一次只能执行一条指令, 因此它的主要局限性是工作速度。采用硬件方案设计的数字系统总是比软件方案的数字系统的工作速度快。

集成电路制造工艺的发展使得在一个芯片上制造大量的数字电路成为可能, 这也促进了计算机技术的发展。把计算机中的微处理器芯片、存储器芯片及输入/输出接口芯片等做在一块芯片上就形成单片机, 有的文献上也将其称为微控制器。这种单芯片的微控制器的性能价格比非常高, 它在工程中应用非常广泛, 例如, 仪表控制、数控机床、自动提款机、复印机、汽车的防抱死制动系统(ABS)、医疗设备等。

3. 专用集成电路

专用集成电路(Application Specific Integrated Circuit, ASIC)的出现在一定程度上克服了上述两种逻辑器件的缺点。专用集成电路是为了满足一种或几种特定功能而专门设计和制作的集成电路芯片, 它的集成度很高。一片专用集成电路芯片甚至可以构成一个完整的数字系统, 因此, 这使得系统的硬件规模进一步降低, 可靠性进一步提高。

专用集成电路可以分为全定制(Full Custom)产品、半定制(Semi-custom)产品和可编程逻辑器件(Programmable Logic Device, PLD)。

1) 全定制产品 全定制产品是指专为特定目的设计、制造的集成电路芯片, 如电视机、电话等设备中大量使用的专用集成电路芯片。这类产品的设计从晶体管的版图尺寸、位置和相互连线开始进行, 其目的是达到半导体芯片面积利用率高、工作速度快、功耗低的优良性能。专用集成电路芯片的制作过程包括电路设计、逻辑模拟、版图设计和集成电路的全

部生产工序。全定制产品的性能优越，但是它的设计制造成本高、周期长，同时还具有较大的风险，因此该产品仅适用于需要进行特大批量生产的情况。

2) 半定制产品 半定制产品内部包含基本逻辑门、触发器和具有特定功能的逻辑块所构成的标准单元。这些标准单元由器件生产厂家预先做好，但是标准单元之间的连线有待按用户要求进行连接。应用半定制产品时，用户需要根据设计要求选择合适的产品，再由产品的结构设计出连线版图，最后交给生产厂家完成各个标准单元之间的连线。

3) 可编程逻辑器件 全定制产品和半定制产品的使用都离不开器件生产厂家的支持，这给用户带来很多麻烦。用户希望自己能设计专用集成电路芯片，并且能立即投入到实际应用之中，而且在使用中也能比较方便地对设计进行修改。可编程逻辑器件就是为满足这一需求而产生的。可编程逻辑器件内的电路和连线都是事先由器件生产厂家做好的，但是其逻辑功能并没有确定。逻辑功能的确定可以由设计者借助于开发工具，通过编写软件的方法来实现。可编程逻辑器件的工作速度与标准逻辑器件工作速度相当，但目前使用它们实现信号处理比使用微处理器要复杂，而且使用成本较高。

1.1.2 标准逻辑器件简介

标准逻辑器件是“数字电路逻辑设计”课程中用来组成数字电路系统的主要器件。“数字电路逻辑设计”课程中介绍的数字电路描述方法不仅适用于由标准逻辑器件组成的电路，而且也适用于在本书将要学习的由可编程逻辑器件组成的电路。

1. 数字电路的描述

数字电路也称为逻辑电路。数字电路的任意一个 I/O 信号仅存在两种可能的状态：高电平或低电平。由于二进制数也只用两个数字（即 0 和 1），因此它适合用来表示数字信号。布尔代数是一种描述逻辑关系的数学工具，利用这种数学工具，数字电路输入信号与输出信号之间的关系（简称 I/O 关系）可以用代数方程（布尔表达式）来描述。布尔代数中的数只有两种可能的取值，与普通代数相比，布尔代数容易计算。布尔代数仅有三种基本运算：与（AND）、或（OR）和非（NOT）。

布尔代数不仅可以作为分析和简化数字电路的工具，而且也可以作为数字电路的设计工具，用来设计满足给定 I/O 关系的逻辑电路。用于数字电路分析与设计的其他方法还包括真值表、电路图、时序图及本书将要讨论的硬件描述语言。如果对这些描述方法进行分类，则可以认为：布尔代数利用数学表达式来描述电路 I/O 关系；真值表利用数字来描述电路 I/O 关系；电路图利用符号来描述电路 I/O 关系；时序图利用信号波形来描述电路 I/O 关系；硬件描述语言利用文本来描述电路 I/O 关系。

布尔表达式可以通过化简获得一个比较简单的形式。简单的形式可用简单的电路来实现，这个简单电路与原电路在功能上等效，但是使用较少的器件，包含较少的连线。进一步来看，这样也提高了电路的可靠性，因为相互之间的连线减少，减少了可能的潜在电路故障。常用的化简方法有代数法和卡诺图法两种。

2. 组合逻辑电路

当一个电路的逻辑功能用布尔表达式给出时，具体的逻辑电路图则可以直接由表达式画

出。适用于简单例子的原理可以推广到复杂的电路。

在组合逻辑电路中,常用的中规模逻辑器件还有编码器、译码器、数据选择器、数据分配器及加法器等。这些中规模逻辑器件也由门电路构成,读者可以参考相关器件的数据手册。

布尔代数仅有三种基本运算,即与(AND)、或(OR)和非(NOT),由这三种布尔代数基本运算的组合就能获得需要的输出。

3. 时序逻辑电路

有些逻辑电路被当作组合电路。组合电路是指在任何时刻,输出状态只取决于该时刻电路输入状态的组合,而与先前电路的状态无关的逻辑电路。无论是控制电路还是数据显示电路,它们的输出只与电路当时的输入有关,与电路先前的状态无关,因此它们都是组合逻辑电路。

但是,有些逻辑电路的输出不仅取决于当前时刻各输入状态的组合,而且还与先前电路的状态有关,即需要记忆功能。这样的逻辑电路称为时序逻辑电路。组合逻辑电路的输出只与电路当前的输入有关;时序逻辑电路的输出则不仅与电路当前的输入有关,而且还与电路上一个状态有关。

最基本的具有记忆功能的电路是触发器,它也由逻辑门组成。逻辑门本身没有记忆能力,但是几个逻辑门组合起来就具有了记忆能力。图 1-1-1 给出边沿触发的 J-K 触发器的内部电路及其电路符号,表 1-1-1 给出功能表。

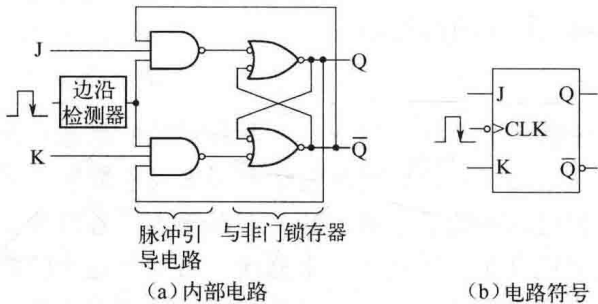


图 1-1-1 边沿触发的 J-K 触发器的内部电路及其电路符号

表 1-1-1 边沿触发的 J-K 触发器的功能表

J	K	CLK	Q
0	0	↑	Q_n (状态不变)
0	1	↑	0
1	0	↑	1
1	1	↑	$\overline{Q_n}$ (状态翻转)

在表 1-1-1 的第 2 行, $J=0, K=0$, 时钟跳变时触发器保持原有状态; 在第 3 行, $J=0, K=1$, 时钟跳变时无论前一个输出状态是什么, 输出状态都清零; 在第 4 行, $J=1, K=0$, 时钟跳变时无论前一个输出状态是什么, 输出状态都置位; 在第 5 行, $J=1, K=1$, 时钟跳变时输出状态发生翻转。

对于时序逻辑电路,常用的中规模逻辑器件还有其他形式的计数器和寄存器等器件。这些中规模逻辑器件也由门电路和触发器构成,读者可以参考相关器件的数据手册。

任意组合逻辑电路都可以由与门、或门和非门这样的逻辑门组成。逻辑门本身没有记忆能力,但是几个逻辑门组合起来可以组成触发器,就具有了记忆能力。把逻辑门和触发器结合起来就可以组成各种时序逻辑电路。综上所述,与门、或门和非门这样的逻辑门是数字电路或数字系统的最基本单元,换句话说,应用与门、或门和非门这样的逻辑门就可以实现所有逻辑功能。

1.1.3 可编程逻辑器件简介

1. 可编程逻辑器件基础

在介绍可编程逻辑器件前，我们再次回顾一下基于标准逻辑器件的数字电路设计过程。首先定义 I/O 变量，即确定设计 I/O 信号，并指定变量名称。然后写出描述 I/O 关系的真值表。真值表是描述电路如何工作的方法之一，描述电路工作原理的另一种方法是布尔表达式。利用布尔表达式就可以用逻辑门符号画出电路图。最后是选择合适的数字集成电路器件组装实际电路。你也许有这样的体会，组装实际电路的过程最麻烦，既费时间又容易出错。如果需要修改电路功能，电路还必须重新组装。

可编程逻辑器件能使组装电路这个烦人的步骤借助计算机和相关的开发软件来完成，因此现在许多数字系统采用可编程逻辑器件实现以提高设计效率。为支持计算机的工作，可编程逻辑器件提供了硬件基础。由数字电路逻辑设计相关的知识，可以得到以下结论：与门、或门和非门这样的基本逻辑门能够组成任何组合逻辑电路；这样的基本逻辑门也能够组成触发器，使电路具有存储能力；组合电路加上存储元件就构成了时序逻辑电路。

可编程逻辑器件的基本原理结构图如图 1-1-2 所示。

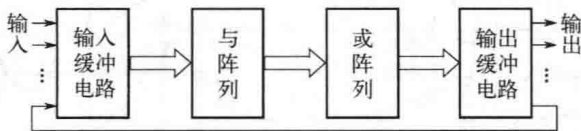


图 1-1-2 可编程逻辑器件的基本原理结构图

输出缓冲电路中通常包括触发器，输出信号可以直接输出，也可以通过触发器输出。触发器的输出信号还可以反馈回来以实现时序电路。

图 1-1-2 也可以看作 CPLD 的原理结构图，这种“与—或”结构组成的可编程逻辑器件的功能比较简单。可编程逻辑器件还有一种基于查找表的逻辑形成方法。查找表的功能就像真值表，对于每一组输入组合存储 0 或 1，从而产生所要求的组合函数。由于使用多个查找表构成一个查找表阵列，因此这种器件称为 FPGA。

2. 可编程逻辑器件内部电路描述

图 1-1-3 给出一个简单的可编程逻辑器件的内部电路图。这个器件具有两个输入端：A 和 B。两个同相缓冲器和两个反相缓冲器构成输入缓冲电路，输入信号 A 和 B 分别经过同相缓冲器和反相缓冲器产生原变量和反变量。这些原变量和反变量连接到与门阵列的输入线。与阵列由 4 个与门组成，它们能够产生两个输入信号的所有组合的与运算，为此每个与门分别连接到两条不同的输入线。与门的输出也称作乘积项线，它们把与运算的结果送到或阵列。

每条乘积项线通过熔丝与每个 4 输入或门的一个输入端相连。1 个 4 输入或门能够实现一个布尔表达式，采用多个 4 输入或门能够实现多个布尔表达式。由于所有熔丝的原始状态是完好的，因此每个或门的输出为 1。以或门 1 为例，未编程前它的输出为

$$O_1 = \overline{A}B + A\overline{B} + \overline{A}\overline{B} + AB = 1$$

烧断熔丝的或门输入端被认为是逻辑 0，通过有选择地烧断熔丝，每一个或门的输出都能产生变量 A 和 B 的任意函数。例如，设计或门 1 的输入端 1 和输入端 4 的熔丝被烧断，则或门 4 的输出为

$$O_4 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B}$$

又如，设计或门 2 的输入端 1、输入端 2 和输入端 3 的熔丝被烧断，则或门 2 的输出为

$$O_2 = 0 + 0 + 0 + AB = AB$$

图 1-1-3 给出的电路仅有两个输入信号，但是电路图已经很复杂了，具有许多连线。实际使用的可编程逻辑器件可能具有几十甚至上百个输入/输出引脚，如果画出其内部电路，将是非常复杂的，以致无法使用。为了简化电路的绘制，描述可编程逻辑器件内部电路时采用了不同于以前的基于标准逻辑器件电路绘制的特殊方法。图 1-1-4 所示为使用这种特殊方法重新绘制的可编程逻辑器件内部电路图。

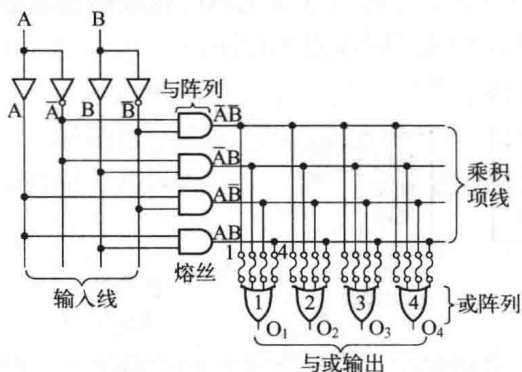


图 1-1-3 可编程逻辑器件内部电路图 (1)

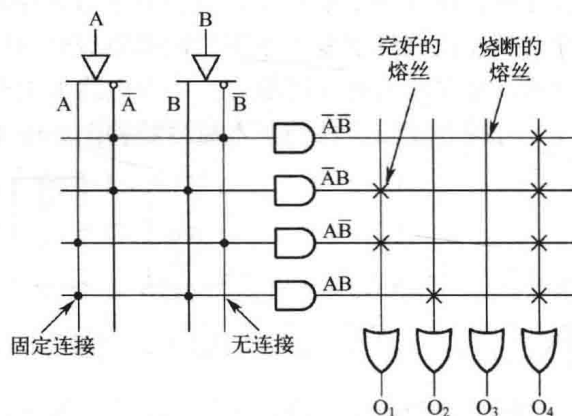


图 1-1-4 可编程逻辑器件内部电路图 (2)

对于图 1-1-4 所示的电路图，与基于标准逻辑器件电路绘制的区别为：一是用具有两个输出端的单个缓冲器代替原来的同相缓冲器和反相缓冲器，新缓冲器的两个输出端，一个表示同相，另一个表示反相；二是与门和或门虽然图上只画了一条输入线，但是它表示该逻辑门具有多个输入信号，行线和列线之间具有多少个交叉点，表示该逻辑门具有多少个输入端。

逻辑门的输入线上的交叉点表示该逻辑门的输入端，但是这些输入端的一些可能与输入信号相连接，另一些可能与输入信号不连接。如果在交叉点上具有“·”符号，则表示这个信号与逻辑门为固定连接；如果在交叉点上具有“×”符号，则表示这个信号与逻辑门为编程连接；如果在交叉点上既没有“·”符号也没有“×”符号，则表示这个信号与逻辑门不连接。以图 1-1-4 所示的电路图为例，它的与门输入端采用固定连接，或门输入端采用编程连接，4 个或门的输出分别为

$$O_1 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B}$$

$$O_2 = 0 + 0 + 0 + AB = AB$$

$$O_3 = 0 + 0 + 0 + 0 = 0$$

$$O_4 = 0 + \bar{A}B + A\bar{B} + 0 = \bar{A}B + A\bar{B}$$

3. 可编程逻辑器件内部电路的分类

可编程逻辑器件一直在发展之中, 这个发展过程经历了可编程只读存储器 (Programmable Read Only Memory, PROM)、可编程逻辑阵列 (Programmable Logic Array, PLA)、通用阵列逻辑 (Generic Array Logic, GAL), 直到复杂可编程逻辑器件 (CPLD) 和现场可编程门阵列 (FPGA)。可编程逻辑器件也可以按器件的编程工艺来划分, 编程工艺包括熔丝型、EPROM 型、E²PROM 型、Flash 型和 SRAM 型。

1.2 Quartus Prime 软件综述

Quartus Prime16.1 是 Altera 公司最新提供的可编程逻辑器件的集成开发软件, 包括了从设计输入和综合直至优化、验证和仿真各个阶段设计 Altera FPGA、SoC 和 CPLD 所需的一切步骤。Quartus Prime 软件提供三种版本: 专业版、标准版和精简版。Quartus Prime 集成开发软件支持可编程逻辑器件开发的整个过程, 它提供一种与器件结构无关的设计环境, 使设计者能方便地进行设计输入、设计处理和器件编程。Quartus Prime 集成开发软件开发流程如图 1-2-1 所示。

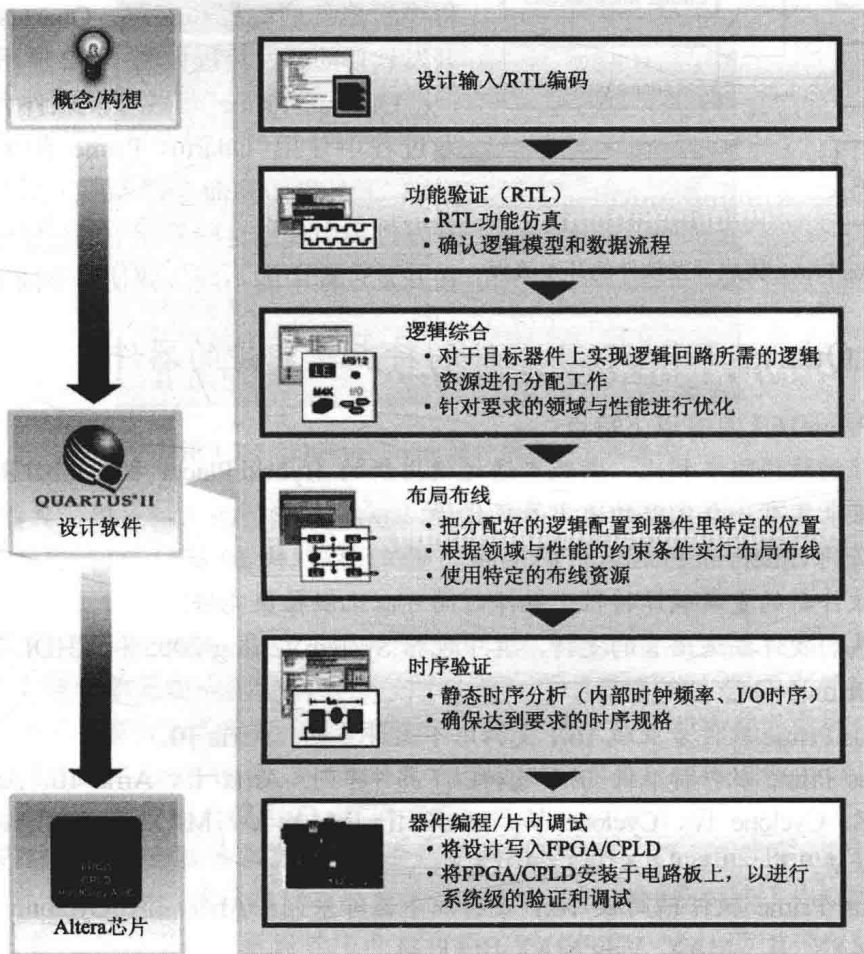


图 1-2-1 Quartus Prime 集成开发软件开发流程

Quartus Prime 16.1 集成开发软件要求的平台工作环境为 Win7sp1、Win8.1、Win10 及部分 Linux 的 64 位操作系统。它支持更多种类的可编程逻辑器件的开发,同时也提供在片可编程系统 (System On a Programmable Chip, SOPC) 设计的综合性环境和基本设计工具。另外,Quartus Prime 集成开发软件也可以利用第三方软件的结果,并支持第三方软件的工作。

为加快应用系统的开发,Quartus Prime 集成开发软件提供更多的知识产权模块 (Intellectual Property, IP)。知识产权模块是一些预先设计好的电路功能模块,在设计中使用这些模块不仅可以加快设计进程,而且还可以提高系统性能。

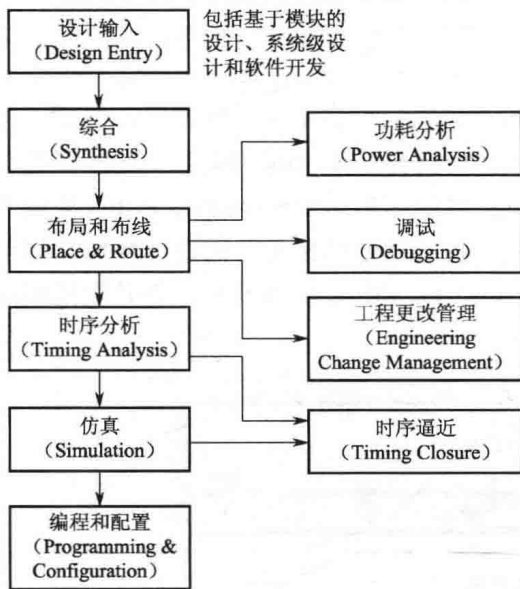


图 1-2-2 Quartus Prime 集成开发软件的开发流程

Quartus Prime 集成开发软件的核心是模块化的编译器。编译器包括的功能模块有分析与综合器 (Analysis & Synthesis)、适配器 (Fitter)、装配器 (Assembler)、时序分析器 (Timing Analyzer)、设计辅助模块 (Design Assistant) 及 EDA 网表文件生成器 (EDA Netlist Writer) 等。可编程逻辑器件的开发流程为:设计输入、综合、布局和布线、时序分析、仿真,以及可编程逻辑器件的编程和配置。Quartus Prime 集成开发软件的开发流程如图 1-2-2 所示。

Quartus Prime 集成开发软件允许用户在设计过程中使用 Quartus Prime 图形用户界面、EDA 工具界面和命令行界面。用户可以在整个开发过程中使用这些界面中的任意一个,也可以在开发过程中的不同步骤使用不同的界面。

1.2.1 Quartus Prime 16.1 软件的特点及支持的器件

Quartus Prime 16.1 具有以下特点:

- 和以往的软件版本相比,新版本通过使用新的 Hybrid Placer 和 Global Router 算法,平均下来具有一个完整的速率等级优势。
- 使用新的 Blueprint Platform Designer,可以比以往快 10 倍。
- 使用软件新的重新编译特性,编译时间可以比以往快 4 倍。
- 对扩展的硬件描述语言的支持,其中包括 SystemVerilog-2005 和 VHDL-2008。

Quartus Prime 16.1 支持的器件:

- Quartus Prime 软件专业版 16.1 支持以下器件系列: Arria 10。
- Quartus Prime 软件标准版 16.1 支持以下器件系列: Arria II、Arria 10、Arria V、Arria V GZ、Cyclone IV、Cyclone V、MAX II、MAX V、MAX 10 FPGA、Stratix IV 和 Stratix V。
- Quartus Prime 软件精简版 16.1 支持以下器件系列: Arria II、Cyclone IV、Cyclone V、MAX II、MAX V 和 MAX 10 FPGA。

1.2.2 Quartus Prime 软件的集成工具及其功能简介

Quartus Prime 软件允许用户在设计流程的每个阶段使用 Quartus Prime 软件图形用户界面、EDA 工具界面或命令行方式，Quartus Prime 软件集成的工具模块如图 1-2-3 所示。



图 1-2-3 Quartus Prime 软件集成的工具模块

1) 设计输入 设计输入即使用 Quartus Prime 软件的模块输入方式、文本输入方式、Core 输入方式和 EDA 设计输入工具等表达用户的电路构思，同时使用分配编辑器 (Assignment Editor) 设定初始约束条件。

2) 综合 综合是将 HDL 语言、原理图等设计输入翻译成由与门、或门、非门、RAM 和触发器等基本逻辑单元组成的逻辑链接 (网络表)，并根据目标与要求 (约束条件) 优化所生成的逻辑链接，输出 edf 或 vqm 等标准格式的网络表文件，供布局布线器实现。除了用 Quartus Prime 软件的 “Analysis & Synthesis” 命令进行综合外，也可以使用第三方综合工具生成与 Quartus Prime 软件配合使用的 edf 网络表文件或 vqm 文件。

3) 布局布线 布局布线输入文件是综合后的网络表文件，Quartus Prime 软件中布局布线包含分析布局布线结果、优化布局布线、增量布局布线和通过反向标注分配等。

4) 时序分析 允许用户分析设计中所有逻辑的时序性能，并协助引导布局布线以满足设计中的时序分析要求。默认情况下，时序分析作为全编译的一部分自动运行，它观察和报

告时序信息,如建立时间、保持时间、时钟至输出延时、最大时钟频率,以及设计的其他时序特性,可以使用时序分析生成的信息分析,调试和验证设计的时序性能。

5) 仿真 仿真分为功能仿真和时序仿真。功能仿真用来验证电路功能是否符合设计要求;时序仿真包含了延时信息,它能较好地反映芯片的工作情况。可以使用 Quartus Prime 集成的仿真工具进行仿真,也可以使用第三方工具对设计进行仿真,如 ModelSim 仿真工具。

6) 编程和配置 在全编译成功后,对 Altera 器件进行编程和配置,包括 Assemble (生成编程文件)、Programmer (建立包含设计所用器件名称和选项的链式文件)和转换编程文件等。

7) 系统级设计 系统级设计包括 SOPC Builder 和 DSP Builder。Quartus Prime 与 SOPC Builder 一起为建立 SOPC 设计提供标准化的图形环境,其中 SOPC 由 CPU、存储器接口、标准外围设备和用户自定义的外围设备等组成。SOPC Builder 允许用户选择和自定义系统模块的各个组件和接口,它将这些组件组合起来,生成对这些组件进行实例化的单个系统模块,并自动生成必要的总线逻辑。DSP Builder 是帮助用户在易于算法应用的开发环境中建立 DSP 设计的硬件表示,缩短了 DSP 设计周期。

8) 软件开发 Software Builder 是 Quartus Prime 软件的集成编程工具,它可以将软件源文件转换为用户配置 Excalibur 器件的闪存格式编程文件或无源格式编程文件。Software Builder 在创建编程文件的同时自动生成仿真器初始化文件,仿真器初始化文件指定了存储单元的每个地址的初始值。

9) 基于块的设计 LogicLock 模块化设计流程支持对复杂设计的某个模块独立地进行设计、实现与优化,并将该模块的实现结果约束在规划好的 FPGA 区域内。

10) EDA 界面 EDA 界面中的 EDA Netlist Writer 生成时序仿真所需要的包含延迟信息的文件,如.vo、.sdo 文件等。

11) 时序逼近 时序逼近即通过控制综合和设计的布局布线来达到时序目标。使用时序逼近流程可以对复杂的设计进行更快的时序逼近,减少优化迭代次数并自动平衡多个设计约束。

12) 调试 SignalTap II 逻辑分析器和 SignalProbe 功能可以分析内部器件节点和 I/O 引脚,同时在系统内以系统速度运行。SignalTap II 逻辑分析器可以捕获和显示 FPGA 内部的实时信号行为。SignalProbe 可以在不影响设计现有布局布线的情况下将内部电路中特定的信号迅速布线到输出引脚,从而无须对整个设计另作一次全编译。

13) 工程更改管理 工程更改管理即在全编译后对设计做的少量修改或调整。这种修改是直接在设计数据库上进行的,而不是修改源代码或配置文件,这样就无须重新运行全编译而快速地实施这些更改。

除了上述集成工具外,Quartus Prime 软件还提供第三方工具的链接。第三方工具包括综合工具和仿真工具,其中综合工具有 Synplify、SynplifyPro 和 LeonardoSpectrum;仿真工具有 ModelSim 和 Aldec HDL 等,它们都是业内公认的专业综合和仿真工具,以其功能强大、界面友好、易学易用而被广泛使用。

1.2.3 Quartus Prime 软件的用户界面

Quartus Prime 软件的默认启动界面如图 1-2-4 所示,由标题栏、菜单栏、工具栏、资源管理窗口、编译状态显示窗口、信息显示窗口和工程工作区等组成。