

【博客藏经阁丛书】

深入理解 Altera FPGA 应用设计

王敏志 编著
至芯科技 审校



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

014013267

【博客藏经阁丛书】

TP332.1

134

深入理解 Altera FPGA 应用设计

王敏志 编著
至芯科技 审校



北航

C1700514

TP 332.1
134



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

内 容 简 介

本书结合作者多年工作实践,以开发流程为主线,通过大量实例详细介绍了 FPGA 开发、调试方面的一些基本方法和独特技巧;提出了一种“自动化”开发设计的理念,即通过批处理加脚本的方式自动完成 FPGA 设计工程的建立、编译和转移,以及在使用 ModelSim 设计仿真时,一键式完成整个仿真流程的方法。对于 FPGA 设计调试和测试方面,则详细介绍了 SignalTap II 的使用,并重点介绍了 Virtual JTAG(虚拟 JTAG)。

本书可作为高等院校通信工程、电子工程、计算机、微电子等专业有一定 FPGA 开发基础的学生们的参考用书,也可作为硬件工程师、FPGA 工程师的工具书。

图书在版编目(CIP)数据

深入理解 Altera FPGA 应用设计 / 王敏志编著. —
北京:北京航空航天大学出版社,2014.1
ISBN 978-7-5124-1338-2

I. ①深… II. ①王… III. ①可编程序逻辑器件—系统设计 IV. ①TP332.1

中国版本图书馆 CIP 数据核字(2013)第 301450 号

版权所有,侵权必究。

深入理解 Altera FPGA 应用设计

王敏志 编著

至芯科技 审校

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:21.75 字数:464 千字

2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1338-2 定价:49.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

序

2013年11月我受 Altera 公司大学计划负责人陈卫中博士邀请,出席在南京召开的全国教师会议,巧遇北京航空航天大学出版社的董编辑,交谈后才知晓有这么一本书即将出版。她请求我浏览一下书稿,并对该书做一个客观的评价。两周前,北京至芯科技的雷总从北航出版社把书稿取来,我才有机会通读了书稿。

我没有时间反复品读书稿的全部内容,也不认识书的作者王敏志,对他的人品和技术水平一无所知。尽管如此,仅从书稿的质量我就能明显地感觉到作者的写作态度是很认真的,书中的许多内容的确是作者本人工作经验的累积,我能体会到作者想与读者交流分享经验的真诚愿望。本书文字简洁,内容通俗易懂,介绍了 FPGA 应用设计常用的综合和仿真工具中能显著提高工作效率的几个重要方法。因此这本书对于已进入 FPGA 设计行业,但还不太熟练的年轻工程师很有参考价值。

当然,任何人都不能期望只读几本书后就能掌握 FPGA 复杂应用设计的全部诀窍。完成 FPGA 复杂应用设计没有捷径可走,只能凭借设计者不屈不挠的顽强意志、科学严密的工作作风,并且在长期实践中坚持不断地学习,积累经验。

希望北航出版社的“博客藏经阁”系列图书,能为从事应用设计的工程师和学生提供互相交流设计经验的平台,使得我国的电子设计水平更上一层楼。

夏宇闻

北京航空航天大学电子信息工程学院退休教授

北京至芯科技公司 FPGA 培训顾问

2013-12-17

前言

FPGA 的规模越来越大,FPGA 的设计也越来越复杂,我们看到,FPGA 不但部分代替了 DSP 的功能,同时加速了和 CPU 的融合速度,FPGA 设计工程师面临既要充当硬件工程师又要充当软件工程师的要求。随着工艺的进步,FPGA 除了向更加高端的方向发展,同时也扩展了其在低端的应用市场,掌握 FPGA 开发技术可以提高电子工程师的竞争能力,同时也能扩展工程师岗位适应空间。

本书特点

EDA 本身就是指电子自动化设计,在本书中,笔者根据多年的实际工作经验总结了一个自动化设计开发理念:设计开发过程中存在许多需要重复操作的流程,而这些的操作都有相关的指令,所以我们可以将这些操作的指令组织在一个脚本文件里,然后通过批处理的方式达到“自动化”地跑这些需要重复操作的流程。这种“批处理加脚本”的开发、仿真和调试方法,是本书介绍的一大重点。而且实践证明,实际工作中也大大提高了笔者的效率。

合理地使用批处理可以提高设计的效率。开发后期占用读者更多时间的是设计的调试和测试,本书在介绍了常规的调试工具(SignalTap II)之后,使用了虚拟 JTAG 这个调试测试利器,这是因为在项目设计完成后进行系统测试的时候,虚拟 JTAG 可以让读者摆脱额外的测试接口,直接利用 FPGA 现成的 JTAG 口完成系统的测试工作。笔者借助虚拟 JTAG 完成了多通道 TDC、多通道 ADC 以及 PMT 能谱等测试,大大提高了测试效率和效果。

正是这些可以提高我们设计效率的技巧和组件,构成了本书的一大特色,也是笔者特别希望介绍给广大读者的。FPGA 的设计开发是一种应用工程,本书更多是在告诉大家“怎么做”和“怎么做会更好”;当然如果可能,也会尽量给出“为什么”要这样做,但这并不是我们工程应用的重点。

本书主要内容

本书以 Altera 的 FPGA 开发流程为基础,详细介绍 Altera FPGA 开发过程以及一些开发技巧,分为 9 章:

➤ 第 1 章 在介绍 FPGA 设计基本要求的基础上,主要介绍 FPGA 开发前的准

备事项。

- 第2章 介绍如何快速建立自己的第一个 Quartus II 工程,重点介绍应用 TCL 脚本来操作 Quartus II 工程。
- 第3章 重点介绍 Altera 推荐的 FPGA 开发技巧。
- 第4章 介绍了使用 ModelSim 仿真 Altera FPGA 设计,重点推荐自动化仿真的概念和方法。
- 第5章 介绍了时序分析,并使用 TimeQuest 对 Altera FPGA 设计进行时序分析。
- 第6章 介绍了如何对 FPGA 进行优化,重点介绍了 Altera 的增量式编译方法。
- 第7章 介绍了如何对 FPGA 设计进行调试和测试,在 SignalTap II 工具的基础上,推荐使用 Virtual JTAG 对 Altera FPGA 设计进行测试。
- 第8章 介绍了几个笔者项目实践中用到的实例,其中包括 FIFO 和 SDRAM 的特殊应用、高速串行接口(GXB)的仿真(特别是 Byte Ordering 模块的仿真)以及 FPGA TDC 设计实例。
- 第9章 介绍了 Altera FPGA 设计开发的一些比较高级的技巧,在了解 FPGA 器件结构的基础上使用一些优化手段,还可以了解到如何使用物理综合以及 DSE 工具对 Altera FPGA 设计工程进行优化。

本书读者对象

本书可作为高等院校通信工程、电子工程、计算机、微电子等专业有一定 FPGA 开发基础的学生的参考用书,也可作为硬件工程师、FPGA 工程师的工具书。

配套资料

本书提供各章实例完整工程文件、设计源文件,读者可以从北京航空航天大学出版社网站(www.buaapress.com.cn)的“下载专区”免费下载。

致 谢

在此需要特别感谢北京航空航天大学出版社工作人员对本书的关心和支持,尤其是编辑的大力支持。

最后感谢您选择了这本书,如果对书里的内容有什么批评、建议或者对书中的内容有任何的困惑都欢迎大家和我联系。

电子邮件:keyoul23@sina.com

博客:http://bbs.ednchina.com/BLOG_coyoo_41415.HTM

王敏志
2014 年 1 月



第 1 章 好好准备你的 FPGA 设计	1
1.1 FPGA 设计要求“软硬兼施”吗	1
1.2 如何选择一个合适的 FPGA	3
1.3 教你如何从顶层规划你的设计	7
1.4 进行早期功耗估算避免 FPGA 动力不足	9
1.4.1 早期功耗估算	9
1.4.2 精确功耗估算	13
1.5 规划并选择片内调试工具	16
1.6 小 结	17
第 2 章 快速建立你的第一个 FPGA 工程	18
2.1 FPGA 设计基本流程	18
2.2 手把手教你用 Quartus II 建立 FPGA 工程	20
2.3 教你如何利用脚本创建工程	24
2.3.1 Quartus II 各个设计流程对应的脚本命令	24
2.3.2 Quartus II TCL 包(Package)	25
2.3.3 执行与 Quartus II 有关脚本的入口	26
2.3.4 利用脚本创建工程	27
2.4 利用脚本约束你的工程	28
2.5 利用脚本自动化完成 FPGA 设计	29
2.6 实例分析	31
2.7 小 结	37
第 3 章 采用 Altera 的建议进行 FPGA 设计	38
3.1 利用 Quartus II 模板开始逻辑设计	38
3.2 同步 FPGA 设计方法及指导	39
3.2.1 同步设计的基本原则	39
3.2.2 异步设计的危害	40
3.3 使用 Altera 的宏函数	41
3.4 在 FPGA 中实现除法功能	41

3.4.1 二进制快速除法	41
3.4.2 使用 Altera 除法函数	43
3.5 Altera 推荐的代码风格	45
3.6 在代码中使用寄存器而不是锁存器	53
3.6.1 锁存器电路结构	53
3.6.2 逻辑设计中哪些情况会产生锁存器	54
3.6.3 锁存器分析实例	56
3.7 使用 Altera“原语”模块	59
3.7.1 如何让设计中的 LCELL 不被软件优化	60
3.7.2 进位链以及如何应用在自己的设计中	62
3.8 小结	65
第4章 使用 ModelSim 进行仿真	66
4.1 ModelSim 仿真工具介绍	66
4.2 使用 ModelSim 软件进行仿真	68
4.2.1 仿真基础——利用 GUI 完成仿真	68
4.2.2 通过创建仿真工程来设计仿真	72
4.3 为 ModelSim 独立版本提取 Altera 仿真库	75
4.3.1 提取 Altera 仿真库的步骤	75
4.3.2 仿真 Altera 器件到底需要提取哪些库	77
4.4 教你如何脚本完成 ModelSim 的自动化仿真流程	81
4.5 ModelSim 使用问题实例	82
4.6 教你如何写 TestBench	86
4.7 实例练习	88
4.7.1 练习前准备	88
4.7.2 GUI 方式仿真实例	91
4.7.2 自动化创建工程仿真实例	92
4.8 小结	93
第5章 教你如何 TimeQuest 来分析你的设计	94
5.1 10 分钟学会使用 TimeQuest	94
5.1.1 时序分析的基本概念	94
5.1.2 教你使用 TimeQuest	95
5.2 时序分析的基础	98
5.3 了解什么是时序约束	104
5.3.1 时钟约束	104
5.3.2 I/O 约束	105

5.4 约束例外	111
5.4.1 多周期路径约束	111
5.4.2 假路径约束	114
5.5 TimeQuest 使用实例	116
5.6 小 结	124
第 6 章 对你的 FPGA 设计进行优化	125
6.1 增量编译使设计加速	125
6.1.1 什么是增量编译	125
6.1.2 认识什么是逻辑锁	128
6.1.3 开始使用增量编译	132
6.2 选择使用合适的设置和约束来优化设计	144
6.2.1 优化之前	145
6.2.2 时序优化	146
6.2.3 面积优化	151
6.3 小 结	152
第 7 章 对你的 FPGA 设计进行调试和测试	153
7.1 SignalTap II	153
7.1.1 教你快速认识 SignalTap II 调试模块	153
7.1.2 教你快速创建第一个 SignalTap II 调试模块并调试	154
7.1.3 教你使用 SignalTap II 高级功能——Storage Qualification	161
7.1.4 教你使用 SignalTap II 高级功能——Power-up Trigger	165
7.2 FPGA 测试利器 Virtual JTAG	168
7.2.1 你所要了解的 JTAG	168
7.2.2 告诉你什么是 Virtual JTAG	172
7.2.3 教你如何在设计中使用 Virtual JTAG Interface (VJI)	175
7.2.4 教你用脚本创建自己的 GUI 虚拟 JTAG 测试平台	180
7.2.5 单个 JTAG 连接多条电缆、多 FPGA 在虚拟 JTAG 中的应用	185
7.3 工程更改管理(ECO)	187
7.3.1 ECO 及其基本操作流程	187
7.3.2 ECO 那些事儿之属性编辑器	189
7.3.3 ECO 那些事儿之 LE 与 ALM	194
7.3.4 ECO 那些事儿之 ALM 的 DATAF 端口	200
7.4 对你的 FPGA 进行正确配置	204
7.5 小 结	216

第 8 章 设计实例应用分析	217
8.1 如何设计应用 Altera 的 FIFO	217
8.1.1 教你如何设计自己的同步 FIFO	218
8.1.2 教你如何使用 Altera 的同步 FIFO	222
8.1.3 教你如何使用 Altera 的异步 FIFO	227
8.2 教你如何向他人转移设计时保护自己的知识产权	228
8.2.1 FPGA 安全性设计——Altera 方案	229
8.2.2 如何加密转移自己的设计	233
8.3 FPGA 外挂接口之 SDRAM	235
8.3.1 SDRAM 芯片	235
8.3.2 SDRAM 控制器逻辑设计	239
8.4 高速串行接口设计没有看上去那么难	259
8.4.1 GXB 模块介绍	259
8.4.2 GXB 应用实例	265
8.5 教你如何在 FPGA 中设计 TDC	272
8.5.1 告诉你到底什么是 TDC	273
8.5.2 基于 FPGA 的 TDC 那些事儿之 3 大难题	277
8.5.3 基于 FPGA 的 TDC 那些事儿之设计资源 LAB	280
8.5.4 基于 FPGA 的 TDC 那些事儿之粗细时间	285
8.5.5 基于 FPGA 的 TDC 那些事儿之自动校准及测量精度	292
8.6 利用 FPGA TDC 测量 PLL 核抖动实例	295
8.7 小 结	300
第 9 章 Altera FPGA 高级设计技巧	301
9.1 器件结构对代码风格的影响	301
9.2 基本逻辑结构分析	302
9.3 可采用的设计技巧	309
9.4 专有资源利用以及优化关键路径	318
9.5 使用 Quartus II 的物理综合对设计进行优化	319
9.5.1 针对性能的物理综合优化选项	323
9.5.2 布线的物理综合优化	325
9.6 了解什么是寄存器打包	325
9.7 探索设计的高级手段——DSE	332
9.8 小 结	336
参考文献	337

第 1 章

好好准备你的 FPGA 设计

开始 FPGA 设计之前必须进行相关的准备工作,至少要有一个目标,即器件选型。也许有人会说,我是逻辑工程师,不关心器件选型,那么同事或队友是否需要关注呢?硬件工程师在设计电路板之前都需要与 FPGA 工程师讨论商定基本的引脚分配,或者逻辑工程师至少要从硬件工程师那里获取最终的引脚分配才能开始逻辑设计。诸如此类的准备工作在开始 FPGA 设计之前,你做好了吗?

本章教你如何进行 FPGA 设计的前期准备工作,主要内容如下:

- FPGA 设计要求“软硬兼施”吗;
- 如何帮助项目选择一个合适的 FPGA;
- 教你如何从顶层规划你的设计;
- 早期功耗估算避免 FPGA 动力不足;
- 规划并选择片内调试工具。

1.1 FPGA 设计要求“软硬兼施”吗

通常来说,从事 FPGA 开发的都是在大学里学通信、电子或者计算机等相关专业的学生,所以笔者觉得 FPGA 开发应该定义为硬件工程师。许多大型公司(比如中兴和华为)会分得更细一些,比如有专门的 FPGA 工程师、甚至逻辑工程师,电路板设计也分为原理图设计和 PCB 设计等。那么 FPGA 设计要不要“软硬兼施”呢?笔者的回答是肯定的,除非刚毕业就能找到提供逻辑工程师岗位的公司。

笔者曾经任职过的一家国有企业的总工(同时兼任某大学博士生导师)总结过一句话,大致意思是:“以后的趋势看来是硬件软件化了”。那时候只是做 IC 原型设计, FPGA 只是一个验证芯片,工作的重点是逻辑设计。他那句话其实包含两层意思,第一层是通过逻辑设计来实现各种各样的电路,第二层是通过在 FPGA 中实现软硬核来实现基于 FPGA 的嵌入式设计。

笔者的经历应该和大部分工程师相似,从设计电路板开始。研究所的设计流程一般是电路板“个人负责制”,即谁设计的电路板谁负责到底,所以设计 FPGA 就必然要求软硬件都要兼顾。

系统定下来之后就是分配单板任务,这个时候就需要开始进行项目规划并撰写规划书。下面就以笔者目前正在维护的一个板子为例来说明如何做到“软硬兼施”。

如图 1-1 所示,这其实是一块 TDC 板,在笔者还未掌握设计基于 FPGA 的 TDC 之前是采用这种架构。板子主要包括一片 FPGA 和 6 片 TDC 芯片,当然还有其他(如供电、ARM 接口等)模块。

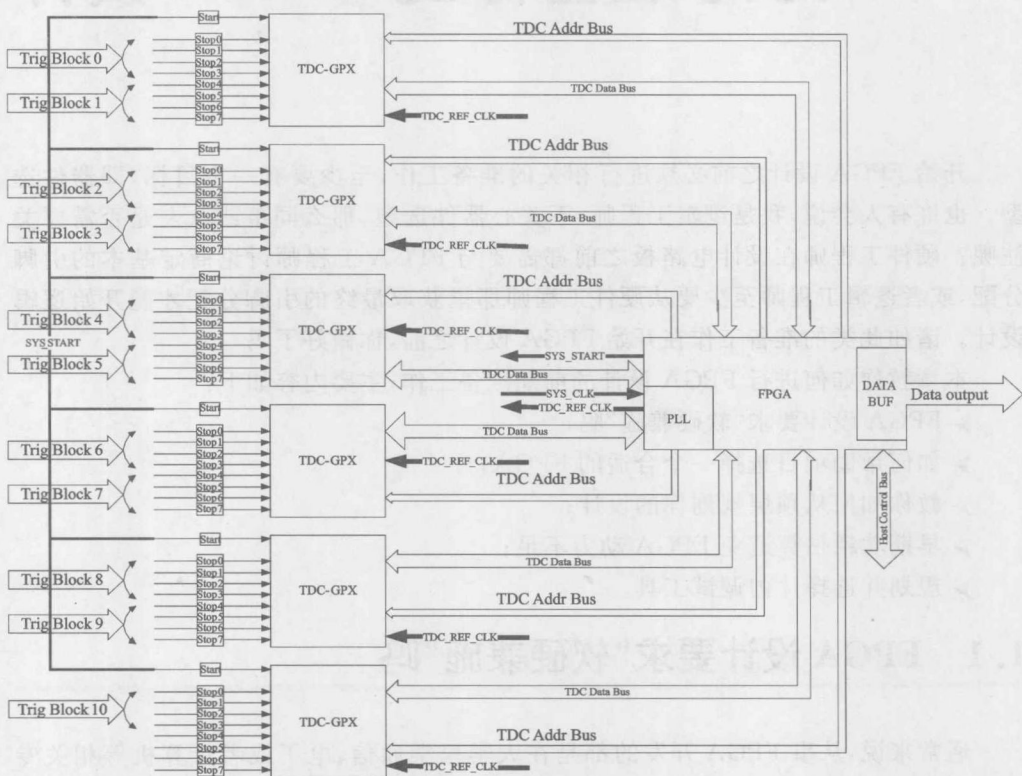


图 1-1 电路板主要结构框图

拿到分配的任务,首先根据系统要求分析本板的需求,比如系统中需要测量的 trigger 信号有 44 个,而我们选择的 TDC 芯片最多只有 8 个测量通道,所以需要至少 6 片 TDC 芯片。TDC 芯片确定下来之后,那么就要分析每一片 TDC 芯片需要连接到 FPGA 的 I/O,这样得出 FPGA 总共需要多少 I/O 才能承受这些 TDC 芯片引脚,再加上 FPGA 其他对外接口,那么 FPGA 所需 I/O 数量就出来了,这是 FPGA 选型的基础。FPGA 型号以及其他事项确定以后,就可以开始设计电路板了。如果只是硬件工程师这应该是没错的,但是同时笔者还是 FPGA 工程师,那么就必须还要

撰写 FPGA 设计规划书;至少在电路板设计之前,将 FPGA 所有用到的 I/O 引脚进行分配并在 Quartus II 软件里分析验证无误后提交下一步电路设计,否则电路设计好以后因为 I/O 分配不合理再去修改电路设计就太麻烦了。

虽然 FPGA 逻辑设计看起来只占图 1-1 所示电路设计的一小部分,但更重要的是逻辑设计(其实质就各种各样“电路”的实现过程),这其中就要求设计者具有一定的硬件电路基础。

1.2 如何选择一個合适的 FPGA

FPGA 的灵活性方便了很多电子设计,那么项目中使用 FPGA 到底要使用它的什么呢?回答了这个问题就解决了 FPGA 选型的问题。从一开始为项目选择一个合适的 FPGA 是不太容易的,所以必须在系统设计的时候将任务分析得非常详细,细化到不能再细化,这样虽然电路板还未设计,但是 FPGA 中要实现的电路已了然于胸。尽管如此,笔者还是建议大家在选型的时候尽量考虑一点裕量,好在各家 FPGA 供应商的各个系列都基本支持相同封装器件的上下兼容,Altera 称为器件的 Migration。

这里还是以图 1-1 的板子为例,当时规划这个板子的时候我们希望用到 FPGA 如下资源:

- 340 个左右的 I/O 引脚;
- 500 kbit 左右的片内 RAM;
- 至少一个发送的 GXB 通道,即 Transmitter。

根据以上主要要求,我们确定选择 Arria GX 系列的 EP1AGX60EF1152C6 型号,而且其可以向上兼容到 EP1AGX90EF1152C6,考虑了设计裕量。器件选择除了上述已知的需求以外,还要考虑器件规模越大、I/O 越多意味着可以实现更大、更复杂的设计,成本付出也更多。留出裕量同时也是为了将来产品升级或者功能扩展的需要;另外,裕量的另外一个重要考虑是要保留足够的逻辑和存储器资源给片内调试工具,因为 SignalTap II 需要消耗片内 RAM 来缓存、抓取用于观测的信号。

如果是已经存在的 FPGA 设计,而只改变了电路板的设计,比如从其他厂家的 FPGA 转到 Altera 的 FPGA,那么可以先在 Quartus II 软件中编译之前的工程,编译的时候在 Setting 里设置使能 Auto device selected by the Fitter 选项,那么就可以通过编译结束报告的资源消耗量从而较快速地选择到合适的器件。

器件基本选定以后就需要将 FPGA 的 I/O 分配给外围的各个接口,比如 TDC、CAN 总线以及 SDRAM 等。记录下分配的 I/O 列表(强烈建议这时候开始使用 TCL 脚本记录引脚分配),记录格式如下:

```
#FPGA 主时钟
set_location_assignment PIN_D19 -to clkln
```

```
set_instance_assignment -name IO_STANDARD "DIFFERENTIAL LVPECL" -to clk_in
```

其实上述格式语法和 QSF 文件是一样的,建立这样一个以 tcl 为后缀的文件非常有利于管理引脚的分配。引脚分配完且进行电路设计之前务必要在 Quartus II 软件里分析该 I/O 分配的合理性。这是因为某些特殊情况下 I/O 的应用是相互排斥的。曾经遇到一个案例:选用的器件是 Stratix 系列,FPGA 工程中使用了 LVDS 高速差分通道进行数据传输,但是工程编译在综合的时候报告引脚分配错误,检查发现是 LVDS 差分引脚附近有 I/O 被用作了单端(Single-ended)引脚。仔细看手册可以看到,其中明确提出差分引脚附近必须间隔几个 pad 的 I/O 引脚才能用在单端 I/O。现在也知道其实可以通过在 Quartus II 修改 Toggle Rates 约束,从而在不改版的情况下解决这个问题。当然,最好还是能避免这个错误,方法是早期分析 I/O 分配。

这个所谓的 I/O 分析器位于 Quartus II 软件 Assignment 菜单下的 PinPlanner 工具中,一般可以在 Quartus II 的工具栏里直接找到这个工具的启动按钮,如图 1-2 所示。

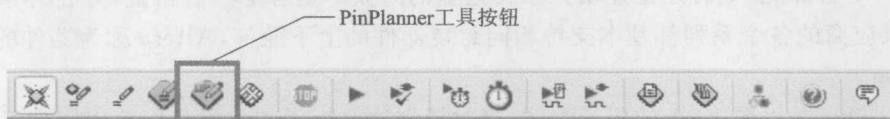


图 1-2 PinPlanner 按钮

启动后 PinPlanner 界面如图 1-3 所示。Process 菜单下有两个命令分别是 Enable live I/O Check 和 Start I/O Assignment Analysis,也可以在图 1-3 的左下方直接启动这两个命令。根据名称可以知道第一个是实时检查 I/O,一般是客户在利用 PinPlanner 进行 I/O 分配的时候一边分配,一边实时检查,一旦遇到非法分配即时报告 error。而第二个是当设计者已经完成了 I/O 分配后进行 Analysis 时,分析结果以报告的形式提交给客户,与普通编译一样。

前面提到差分引脚和单端引脚分配有些情况存在互斥限制,比如差分引脚周围就有至少隔多少个 pad 行(注:pad 是 pin 对应与 FPGA die 中的“引脚”)才能配置单端引脚的限制等。不能依据图 1-3 所示的引脚(pin)物理位置关系来确定内部 die 中 pad 之间的物理位置关系,查看 pin 对应的 pad“物理位置”关系可以通过图 1-4 所示的 View→Pad View 菜单项查看。

注意观察图 1-5 所示局部放大的 Pad View 界面,虚线分隔开了不同的 pad 行和列。如果已经分配好了差分引脚,那么一般相邻两个 pad 行或者 pad 列中的 pad 对应的 pin(I/O)就不能再被分配为单端引脚使用了。

下面举一个实际的例子来说明。如果 FPGA 外挂有 DDR SDRAM,那么已经使用到的 DDR 的 VREF 引脚附近必须间隔两个 pad 才能分配普通的 pin(普通的 pin 就是非 DQ、DM 以及 DQS 等 pin),如图 1-6 所示。

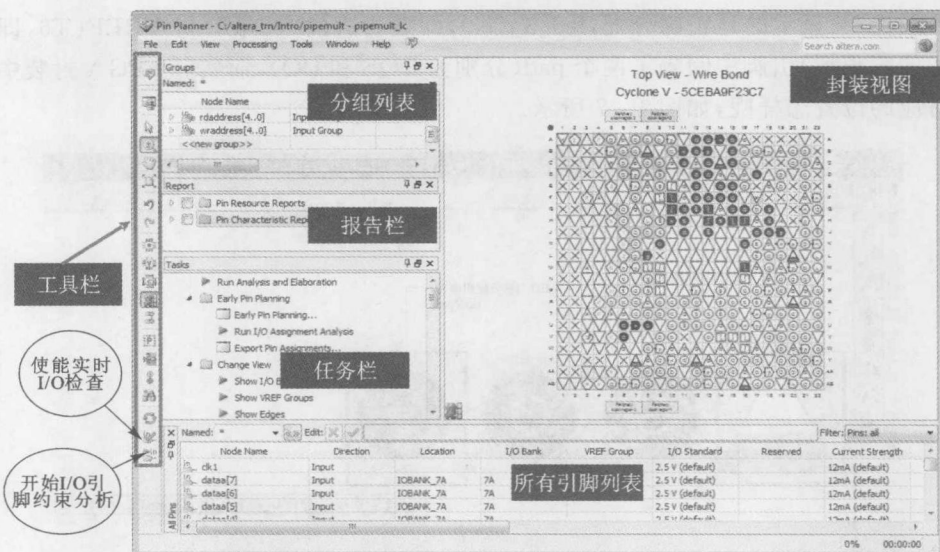


图 1-3 PinPlanner 界面

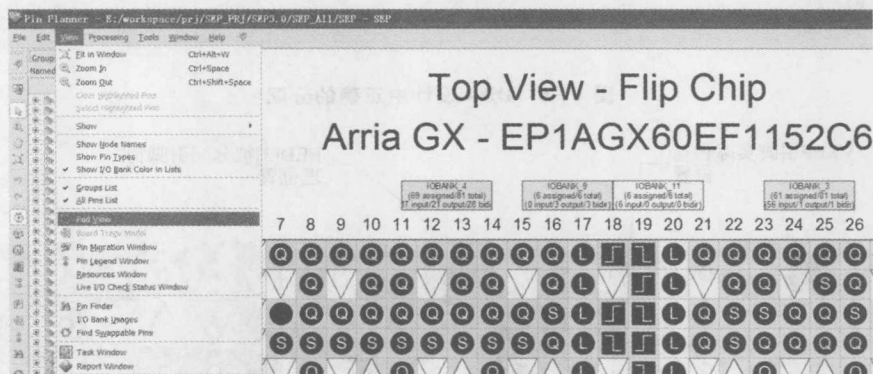


图 1-4 在 PinPlanner 打开 Pad View

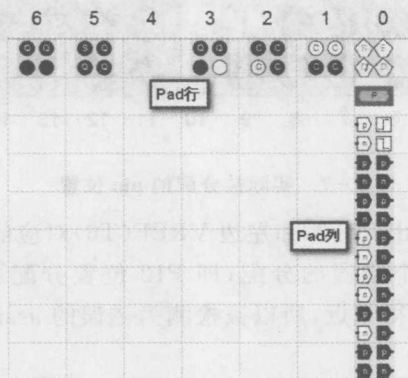


图 1-5 放大后的 Pad View 界面

图 1-6 显示的是正确分配, led[3](N9, 即最右边的“红球”)与 VREF(T6, 即最左边的三角形)引脚中间隔了两个 pad(分别为 DQS 和 DQ), 而实际 FPGA 封装中它们对应的位置怎样呢, 如图 1-7 所示。

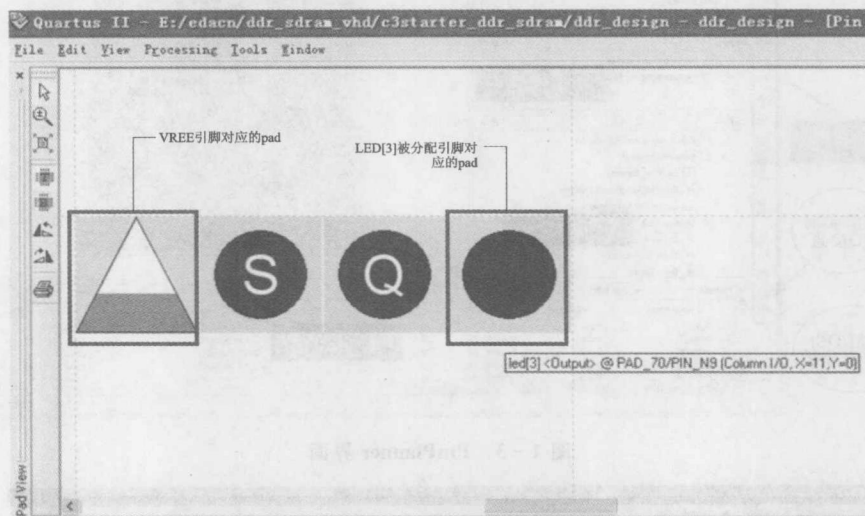


图 1-6 DDR 设计中正确的分配

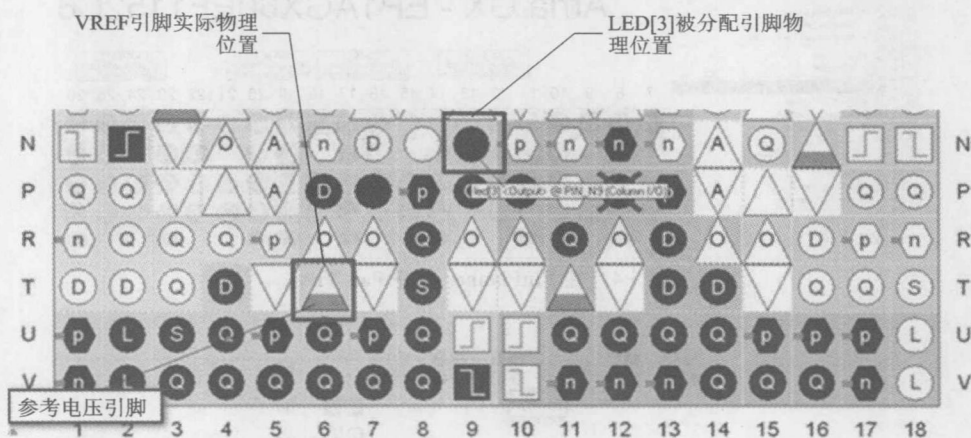


图 1-7 实际被分配的 pin 位置

从图 1-7 其实很难看出 led[3]和左边 VREF(T6)对应的 pad 位置关系。另外, 图 1-7 同时显示了另一个错误的分配, 即 P12 位置分配了 led[1], 其与另一个 VREF(T11)对应的 pad 靠得太近, 所以被检测为错误的 assignment, 它们的 pad 对应图如图 1-8 所示。

所以进行 I/O 分配分析非常重要。如果个人负责单板设计, 则一般都会进行这

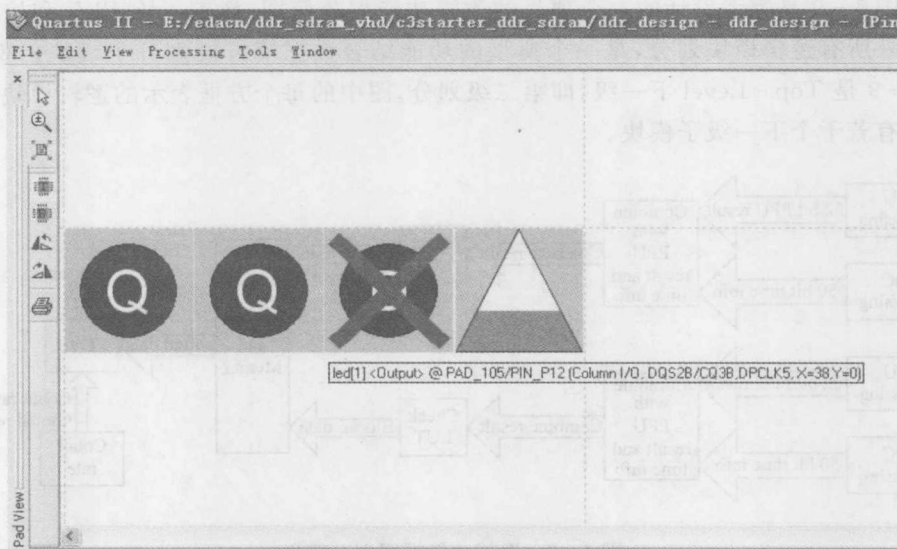


图 1-8 错误的 I/O 分配被实时检测出来

个分析;如果团队合作,I/O 由电路设计工程师分配,那么 FPGA 设计工程师一定要在进行电路设计(至少是在 Layout)之前将其分配的 I/O 列表拿到并分析确认。如果是 FPGA 工程师分配的,一定要经过分析才能提交给电路设计工程师。

1.3 教你如何从顶层规划你的设计

前面提到了系统设计,系统设计完成以后就是系统任务分配,之后是单板设计。在单板电路设计过程中,可以并行地进行 FPGA 的逻辑设计。与系统设计对应,开始进行逻辑之前一般需要进行任务划分,使整个设计形成比较合理的层次结构。

建议从顶层开始规划设计,而且顶层只进行 I/O 引脚定义,然后自顶向下划分并定义相对独立、功能单一各种子模块。那么应该依据什么样的原则来划分子模块呢?以下的原则可以参考:

- 按照功能划分;
- 按数据流方向划分;
- 按通信划分;
- 按逻辑划分。

一般以不同功能划分不同模块为主要原则,然而有时候设计中有些模块与好几个功能模块关系紧密,所以需要按照数据流方向来划分,同时可以结合逻辑关系、通信方式来综合考虑。总之,最终目的是要将模块划分得越独立越好,所以子模块下还可以再向下划分子模块。