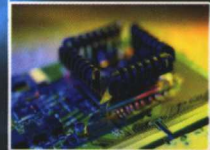




VHDL

芯片设计

陈荣 陈华◎著



碁峯

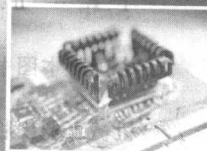
www.gotop.com.tw



机械工业出版社
China Machine Press

VHDL

芯片设计



陈荣 陈华◎著



机械工业出版社
China Machine Press

本书是为熟练掌握VHDL芯片设计而编写的基础教材。本书以VHDL语言为载体,详细介绍了功能仿真软件ModelSim、综合软件 Synplify、时序仿真软件 MaxplusII,并通过丰富的实例对照,全面介绍良好的VHDL编程风格。重点讲述数字电路设计的概念,内容涵盖VHDL行为模型、层次式模块化设计、组合逻辑电路设计、状态机设计、测试平台设计等,并以大量VHDL程序实例演示说明有关应用程序的设计过程,介绍主流的编程思想及方法,培养读者的代码编写能力及良好的设计风格。

本书内容深入浅出,覆盖面广,图文并茂,独具特色。既有丰富的理论知识,也有大量的实战范例,使读者能循序渐进,由浅入深。

本书是在作者多年实践基础上编写的,适合作为电子、电机等专业相关课程的教材,对于从事电子工程设计的科技人员,本书也是极佳的参考。

本书为经台湾基峰资讯股份有限公司独家授权发行的中文简体版。本书中文简体版在中国大陆之专有出版权属机械工业出版社所有。在没有得到本书原版出版者和本书出版者书面许可时,任何单位和个人不得擅自摘抄、复制本书的一部分或全部以任何方式(包括资料和出版物)进行传播。本书原版版权属基峰资讯股份有限公司。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2005-5337

图书在版编目(CIP)数据

VHDL芯片设计/陈荣,陈华著.-北京:机械工业出版社,2006.1

ISBN 7-111-17804-1

I.精… II.①陈… ②陈… III.硬件描述语言,VHDL-程序设计 IV.TP312

中国版本图书馆CIP数据核字(2005)第130458号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:范运年

北京京北制版厂印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷

787mm×1020mm 1/16·18.5印张

印数:0 001-4 000册

定价:35.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

本社购书热线:(010)68326294

序 言

好不容易写完了这一本《精通VHDL芯片设计》，其中的内容相当丰富，包含了VHDL的基本概念及程序设计风格。笔者在初学VHDL时，由于建立了良好的设计风格的概念，所以受益良多，由于程序的设计风格会影响综合及模拟的结果，因此本书也用不少篇幅介绍如何建立良好程序设计风格的概念。

本书同时介绍业界常使用的电脑教学设计工具，如模拟软件（ModelSim）、综合软件（Synplify）及时序模拟软件（MaxplusII），学会这些软件会对IC及FPGA设计有相当大的帮助。

本书适合工程学院电子、电机及大专程度读者使用，也适合从事研究的工程师作为参考用书。读者可将本书定位为一本工具书来阅读，本书的特色是由浅入深的主题介绍，并为VHDL语法提供了简单例子让读者参考研究。

最后要感谢家人的支持，没有他们本书不会顺利完成，他们是我努力的动力，也希望这本书能对我国芯片设计业的发展尽一点绵薄的力量。

目 录

序言	
第1章 VHDL设计概念	1
1.1 数字电路设计简介	1
1.2 ASIC和FPGA组件比较	2
1.2.1 ASIC	2
1.2.2 FPGA	4
1.3 FPGA和ASIC设计流程简介	5
1.4 数字电路设计所需考虑的因素	7
第2章 EDA软件介绍	10
2.1 功能仿真软件ModelSim	10
2.1.1 ModelSim简介	10
2.1.2 建立一个新的项目	11
2.1.3 基本VHDL仿真	14
2.1.4 除错	16
2.1.5 Finding names and values	19
2.1.6 使用Wave窗口	20
2.1.7 性能分析器仿真	22
2.1.8 Code Coverage仿真	25
2.1.9 ModelSim 常用指令集	28
2.2 综合工具Synplify	30
2.2.1 Synplify简介	30
2.2.2 Synplify的特色	31
2.2.3 Synplify FPGA设计流程	32
2.2.4 安装事项	32
2.2.5 Synplify用户接口	33
2.2.6 设定源文件	33
2.2.7 检查源文件	35
2.2.8 RTL View	35
2.2.9 Synplify Altera Flow	37
2.3 Altera MaxplusII EDA Tool	42
2.3.1 Maxplus II 版本介绍及安装方式	43
2.3.2 设计输入	53
2.3.3 功能仿真	57
2.3.4 平面布局	64
2.4 结论	70
第3章 初探HDL语言	71
3.1 HDL的好处	71
3.2 VHDL和Verilog的比较	71
3.3 如何选择电路的结构	72
3.4 HDL程序的组成	73
3.5 HDL程序结构	77
第4章 基本VHDL要素	79
4.1 标识符	79
4.2 数据对象	80
4.3 数据类型	82
4.3.1 标量数据类型	82
4.3.2 复合数据类型	84
4.3.3 数组数据类型	84
4.3.4 记录数据类型	85
4.4 运算操作符	85
第5章 VHDL行为模型	91
5.1 简介	91
5.2 实体声明	91
5.3 结构体	92
5.4 进程语句	93
5.5 变量赋值语句	94
5.6 信号赋值语句	95
5.7 Wait 语句	95
5.8 if语句	97
5.9 Case 语句	100
5.10 Null 语句	104
5.11 Loop语句	104
5.12 Exit 语句	109

5.13 Next 语句	109	第9章 层次式模块化设计	151
5.14 Assertion 语句	110	9.1 简介	151
5.15 Report 语句	111	9.2 类属和配置	151
5.16 信号赋值语句进阶探讨	112	9.2.1 类属	151
5.16.1 惯性延迟模型	112	9.2.2 配置	153
5.16.2 传输延迟模型	113	9.3 生成语句	162
5.17 建立信号波形	113	第10章 子程序及包	167
5.18 多进程	114	10.1 简介	167
第6章 数据流模型	117	10.2 子程序	167
6.1 简介	117	10.2.1 函数	167
6.2 并发性信号赋值语句	117	10.2.2 过程	170
6.3 并发性与顺序性信号赋值	118	10.3 运算符重载	171
6.4 探讨Delta delay	119	10.4 包	175
6.5 条件信号赋值语句	121	10.4.1 包声明	175
6.6 选择信号赋值语句	122	10.4.2 包主体	177
6.7 Unaffected值	123	第11章 组合逻辑电路设计	180
6.8 块语句	124	11.1 简介	180
第7章 结构化模型	127	11.2 多选器设计	180
7.1 简介	127	11.3 编码器设计	183
7.2 简单的范例	127	11.4 优先级编码器设计	187
7.3 元件声明	128	11.5 译码器设计	189
7.4 元件实例化	130	11.6 含有使能的译码器设计	192
7.5 层次式结构	131	11.7 四位地址译码器设计	194
7.6 其他范例	134	11.8 使用Generic N to M 位的 二进制译码器	197
第8章 VHDL 中的属性和配置	142	11.9 比较运算符	200
8.1 简介	142	11.10 算术逻辑单元设计	202
8.2 数值属性	142	第12章 时序逻辑电路设计	205
8.2.1 数值类型属性	143	12.1 简介	205
8.2.2 数值数组属性	143	12.2 D型触发器	205
8.2.3 数值块属性	144	12.3 触发器应用电路	212
8.3 函数属性	144	12.3.1 延迟电路	212
8.3.1 函数类型属性	144	12.3.2 微分器电路设计	214
8.3.2 函数数组属性	145	12.4 计数器设计	216
8.3.3 函数信号属性	147	12.5 同步计数器电路设计	220
8.4 类型属性	148	12.6 分频电路设计	220
8.5 范围属性	148		
8.6 信号属性	149		

第13章 状态机设计	227	15.3.12 使用有意义的标记名称	251
13.1 状态机简介	227	15.4 可移植性	252
13.2 Moore 状态机	227	15.4.1 使用IEEE 标准类型	252
13.3 Melay 状态机	231	15.4.2 不要使用实际的数值	252
第14章 测试平台	234	15.4.3 包	253
14.1 简介	234	15.4.4 转换 (VHDL至Verilog)	253
14.2 如何写测试平台	234	15.5 有关Clock和Reset的准则	253
14.3 波形产生器	235	15.5.1 避免使用混合频率触发	253
14.3.1 重复性和非重复的测试样本	235	15.5.2 避免对频率信号做处理	254
14.3.2 使用向量的方式	238	15.5.3 避免内部产生频率信号	254
14.4 整数转换成time数据类型	240	15.5.4 门时钟和低功率设计	255
14.5 将结果保存为文本文件	240	15.5.5 避免内部产生Reset信号	255
14.6 从文本文件中读取向量	242	15.6 Coding for Synthesis	256
第15章 RTL Coding Guideline	245	15.6.1 触发器的写法	256
15.1 简介	245	15.6.2 避免锁存器	257
15.2 初探Coding Guideline	245	15.6.3 避免使用组合逻辑的反馈	262
15.3 基本的代码注意事项	245	15.6.4 完整的敏感表	262
15.3.1 一般的命名方式	245	15.6.5 信号和变量赋值	264
15.3.2 结构名称命名规则	246	15.6.6 Case语句与if-then-else 语句	264
15.3.3 标题文件的注意事项	246	15.6.7 状态机的编程风格	265
15.3.4 注释	247	15.6.8 Partitioning for Synthesis	267
15.3.5 一行的长度	248	第16章 高级设计范例	269
15.3.6 缩进	248	16.1 数字闹铃电路设计	269
15.3.7 Port的顺序	248	16.2 最大公因子电路设计及验证	278
15.3.8 端口映射和类属映射	249	16.2.1 设计要求	278
15.3.9 实体、结构和配置	250	16.2.2 设计概念	278
15.3.10 使用函数	250	16.2.3 测试平台的设计	285
15.3.11 使用Loops和Arrays	250		

第1章 VHDL设计概念

1.1 数字电路设计简介

在过去，数字设计都是使用画电路图的方式，但现在数字电路设计由于计算机辅助设计软件（EDA）的功能愈来愈强大，模拟合成所需的时间愈来愈短，而设计的门数（gate count）由原本几千门增加到数十万门，甚至现在利用基于硅知识产权来设计系统芯片，其门数更达到几百万门，所以以往传统设计的方式已不合时宜，将逐渐被取代，而系统设计者总是想要加快产品上市的时间，以提升自己的竞争力并保证自己的设计正确性，为了要达到快速且功能无误，所以要慎选设计方法。通常在数字电路设计中主要分为自顶向下的设计流程（Top Down design methodology）、自底向上的设计流程（Bottom Up design methodology）、平面设计流程（Flat design methodology），在下面分别详细介绍。

1. 自顶向下的设计方式

其最主要流程为：系统→模块划分→功能验证→实现

- 1) 先将设计分成若干个模块。
- 2) 对于每个模块都定义及验证其功能仿真是否合乎设计所需。
- 3) 对于每个模块作定义及验证其时序模拟是否合乎设计所需。
- 4) 定义及验证整个系统的功能是否合乎所需。

而自顶向下设计方式的优点主要为：

- 1) 容易设计，控制简单。
- 2) 设计时间短。
- 3) 工程师不必具备专业的IC设计基础。
- 4) 增加产品的成品率，并且缩短IC的设计周期，以加快产品上市的时间。
- 5) 模块化设计并配合模块复用（Reuse）的设计方式。
- 6) 可以增加设计的弹性。
- 7) 有较容易和较好的设计验证方法。

另一方面，其缺点为：

- 1) 所设计的IC速度较慢。
- 2) 不容易产生数据路径。
- 3) 所设计的芯片面积较大。

2. 自底向上的设计方式

其主要流程为：晶体管→逻辑门→功能验证→模块→系统

其主要优点为：

- 1) 容易设计有规则，可以重复的组件，如RAM、ROM、寄存器等。
- 2) 可以精密地调整晶体管的尺寸。

■ 所设计的IC速度较快

■ 单位面积的晶体管密度较高

但缺点为:

- 1) 设计者必需了解IC的布局方式。
- 2) 不容易设计复杂的控制单元。
- 3) 设计时间较长。

自顶向下和自底向上的设计方式均属层次式设计, 通常层次式设计尽量不要超过三层以上, 因为会造成组件间连接的困难度及除错的困难度, 层次式设计往往用于设计规模较大的电路。

3. 平面设计方式

如果设计者所设计的电路规模比较小, 这种设计方式就没有层次的概念, 也就是说所有的设计及组件都在同一层次中进行, 必需注意的是详加定义每个组件的规格及组件相互间的连接方式。

以上这三种设计方法均包含了硬件描述语言、综合及仿真验证, 而现在更可以利用专用集成电路(ASIC)和现场可编程门阵列(FPGA)来实现想要设计的硬件。超大规模集成电路(VLSI)设计技术与制造工业的快速成长, 使得所设计的系统越来越复杂, 所以数字系统的工程师需要: 1) 易学易用的硬件描述语言; 2) 功能强大的计算机辅助设计软件; 3) 可以快速原形化的IC组件。希望可以用以上三个工具来加快获利时间和上市时间。

其中在设计初期我们必须制定所需的规格, 如工作频率大约要到多少? 工作电压大约几伏特? IC的输入输出管脚数为多少? 如何将一个大的设计划分成好几个功能模块? 而每个功能模块的功能是什么? 这都要详细地去定义, 定义得越清楚, 在后面设计IC时, 就会避免很多不必要的麻烦。如果是由最下层的功能模块来做设计, 每个功能模块做完功能仿真和时序仿真验证后, 将每个功能模块加以整合, 并且再做整体的功能和时序仿真的验证, 即可完成整体的设计, 称为自底向上的设计方式。

在本书中, 主要针对硬件描述语言(VHDL)做说明, 除了介绍VHDL如何使用之外, 最重要的是介绍VHDL的程序风格。如何建立良好VHDL程序风格才能让设计速度加快, 芯片的面积更省, 这是相当不容易的。本书中所有的VHDL程序代码并不仅仅能够作时序上的模拟, 也可以作电路的综合。同时综合出来的电路放在程序代码下方以供设计者参考比较。本书使用Synplify 7.0计算机辅助设计软件和Altera公司所出的MaxplusII软件来作为模拟验证的工具。

1.2 ASIC和FPGA组件比较

数字逻辑组件如ASIC、FPGA, 随着集成电路和工艺的进步也有相当大的改变, 而它们共同的优点是: 电路性能佳、可编程逻辑设计、成本较低、实现出来的电路具有保密性, 都比以往的使用TTL系列组件所设计的快上许多。

1.2.1 ASIC

ASIC为Application Specific Integrated Circuit的缩写, 中文全称为“专用集成电路”。这种集成电路的设计需求, 是提供设计者在所需的特殊场合所使用的集成电路组件。在组件分类上分别分为两种。

1. 标准集成电路组件

如:

- 1) 中央处理单元 (CPU)。
- 2) 内存: DRAM、SRAM、ROM、EPROM、EEPROM。
- 3) 计算机主板上的南北桥芯片。
- 4) 显示卡上的绘图芯片。
- 5) 加法器/减法器、比较器、多选器。
- 6) 手机上的通信芯片。

2. 专用集成电路组件

如:

- 1) 全定制 (Full Custom)。
- 2) 半定制 (Semi Custom)。
 - 门阵列 (Gate Array)
 - 标准单元 (Standard Cell)
 - 积木块 (Cell Based)
- 3) 现场可编程门阵列 (FPGA - PLD)。

以上均属于特殊用途使用的组件,一方面可以增加电路设计的安全性,另一方面使系统的整合更有效率,但是如果要做系统的整合尚需要注意下列几项:

- 1) 较少的面积,较快的速度。
- 2) 较低的功率消耗。
- 3) 较高的可移植性。

上述每一项都有其重要性,面积和速度主要是互补的,可以用较多的面积去加快速度,或是用速度去换面积,全依设计者的考虑,其中如何让电路速度加快,可以利用流水处理 (Pipeline)、并行 (Parallel) 处理等设计方式,后续章节均会加以讨论。

专用集成电路组件的半导体工艺分为四种方式,分别说明如下:

(1) 门阵列

利用门阵列的方式来设计ASIC,就是由集成电路厂商提供部分完成的晶体管布局,并且加上数层掩模版(通常为四层)来构成各个晶体管的连接关系,以达成所需电路设计需求。这种制作方式的优点是所变动的掩模版数少,因此非重复性工程 (Non-Recurring Engineering, NRE) 费用较少,工艺流程也减少许多;但是主要的缺点为受所选母体限制,因此单价成本提升,设计的整合性也较低。而门阵列包含了数千个或数百个等效逻辑门,通常占整体门数的70%~90%。

(2) 标准单元

利用标准单元的方式来设计ASIC,都是以集成电路厂商所有的标准单元为基础,所设计的数字逻辑电路均是由这些标准单元 (Cell) 综合出来的,因此设计较有弹性;但缺点为设计所需的掩模版层数,较门阵列方式多(约十二层以上)。标准单元设计方式,可以使成本降低,且整合性较高,但相对地要付出较高的NRE费用及较长的制作时间。

(3) 积木块

积木块的设计方式是综合门阵列和标准单元的优点来加以设计。积木块是将设计的数字逻辑

电路中的不同模块，均以单元的方式来建立，如此需要改变其中一个模块中的电路设计时，只要改变相对应的掩模版即可。所以可以利用这个优点来做个规划，试想如果将那些不必改变的单元以固定的掩模版来制作，另外再将可变化的电路设计部分以另外一些掩模版来制作，所以只需要第一次制作ASIC时所需要的每一层的掩模版费用，往后只要改变所需更动的掩模版即可。

(4) 全定制

全定制设计方式是全部由客户委托的电路设计为主要的考虑，将电路做最佳的设计整合，以得到一个价格最合理且性能优良的解决方式，但是所付出的代价是更高的NRE费用并且大部分的电路模块和逻辑组件都是经由人工慢慢设计出来的，因此设计所需的时间也较多。

所以要使用四种的哪一种设计方式，全依设计者考虑的因素来决定，如成本、NRE的费用、整合性等。

1.2.2 FPGA

现场可编程门阵列 (Field Programmable Gate Array, FPGA) IC, FPGA组件有比逻辑组件容易使用的优点，它最常使用的逻辑门有与门、或门、多选器、触发器等，之后利用已经设计好的组件做个连接使之达到设计者所要求的功能，但是相应的硬件设计弹性会降低许多。

通常FPGA组件可编程的结构主要分为下列两种：

1. 反熔丝结构 (Anti-Fuse)

顾名思义，其本身就是像熔丝数组一样，将我们的电路设计下载至FPGA上，我们称之为烧录，原理就是将FPGA的熔丝烧断，因此若将电源关掉，则FPGA内的数据不会遗失，此结构称为反熔丝结构。

2. 静态随机存取内存 (SRAM)

SRAM电路主要的结构为双稳态电路，在被写入数据后，它会维持原来状态直到电源被中断或是写入其他的值为止。其中需注意到的一点是，若将电源关掉，FPGA内的数据会被洗掉，直到下次重新做预充 (Configuration) 后，FPGA才会恢复原来功能。

所以在设计初期，尽量使用以SRAM为设计基础的FPGA来作为设计工具，因为使用反熔丝结构的FPGA设计时，若设计出现错误，则整个FPGA就无法再使用。除非设计者很有把握，否则建议在设计初期使用以SRAM为设计基础的FPGA，若发生错误，只要重新再将设计下载至FPGA上，等到整个设计成熟后，再使用反熔丝结构的FPGA。因为在烧断熔丝后FPGA功能已固定，往后使用过程不用重新做预充的动作，可以加快开机的速度。

在FPGA组件结构中，又可以依照它的可编程逻辑模块电路在芯片的排列方式，分为二个主要的结构，分别说明如下：

(1) 矩阵型结构

矩阵型结构是把可编程的逻辑模块排列成二维的矩阵，它是由许多的逻辑单元 (Logic Cell) 经由可编程的垂直通道 (Vertical Channel) 及水平通道 (Horizontal Channel) 的连接所构成。它的结构和门阵列方式很类似，不同之处第一点在于门阵列是由晶体管组成，而FPGA组件是由逻辑单元组成，第二点在于复杂可编程逻辑器件 (CPLD) 中的微单元 (Macro Cell) 比FPGA组件的逻辑单元大一些，但是FPGA的逻辑单元的数量比CPLD的微细胞多，可以利用逻辑单元的互相串并联来满足复杂电路的需求。Xilinx和Altera公司等所生产的FPGA产品主要是采用矩阵型的结构。

(2) 列向量结构

在列向量结构中最主要的特点是所有的可编程逻辑模块均紧密的排成一列，列和列之间的区域称为布线通道（Routing Channel）。Actel公司的FPGA产品采用列向量结构。

1.3 FPGA和ASIC设计流程简介

IC设计的设计流程主要区分为全定制、积木块及FPGA 三种，在前面已经大致介绍了这三种设计的差别，接下来要介绍ASIC和FPGA的设计流程之间的差别，二者的前端设计（Front Design）大致相同，在后端设计，也就是在物理层级的设计流程中才有差异，表1-1说明了其差异。

表 1-1

ASIC 设计流程	FPGA 设计流程	
1. Design Entry设计输入	1. Design Entry设计输入	↑
2. Function simulation功能仿真	2. Function simulation功能仿真	前端
3. Synthesis 综合	3. Synthesis 综合	↓
4. PreSim 前仿真	4. Place&Route布局布线	↑
5. Layout 电路布局	5. Timing Simulation 时序模拟	后
6. Postsim后仿真	6. Prototype原型机	端
7. Prototype原型机		↓
8. Verification电路验证		

首先必须注意，无论是在IC设计或是FPGA设计流程中，前端设计部分都是相同的，因为所设计的电路在经过计算机辅助设计工具综合、功能验证时，都没有涉及综合出来的电路要放置什么样的组件（CPLD、FPGA等），所以在前端设计ASIC和FPGA设计流程是相同的，设计流程图如图1-1所示。

1. 规格制定 (Spec)

在设计初期必须先制定规格，如工作频率大约要到多少？工作电压大约几伏特？IC的输入输出管脚数（I/O Pins）为多少？关键路径（Critical path）会发生在哪儿？如何将一个大的设计划分成几个功能模块，每个功能模块也要详细定义其功能是什么。定义得越清楚，在后面设计IC时，就会避免越多不必要的麻烦。

2. 电路结构设计 (Architecture design)

在制定好规格之后，依据这种规格去设计电路结构，是利用流水结构设计（Pipeline design）还是先进先出（First in First out）设计，都要谨慎去考虑。

3. 设计输入 (Design Entry)

利用HDL硬件描述语言，如VHDL、Verilog、AHDL等，以电路图、波形图、状态图、状态表作为设计输入。在传统的设计中几乎都是使用电路图输入的方式（Schematics entry），这种设计的优点为可以让设计者一目了然。然而如今设计的复杂度越来越高，传统电路图设计已不敷使用，所以电路图设计几乎都使用在规模较小的设计上，若想要设计大规模的电路，可能无法实行。为了适应如今越来越大的电路设计，可以利用硬件描述语言，来加快产品上市的时间。

假若比对手晚推出产品，即使产品性能较好、价钱较低，但对方产品已占有很大的市场，获利也会相对降低许多。所以在好几百万门的IC设计上，HDL已大部分取代了电路图输入的方式。当然也可以使用波形图或状态表设计输入，但是难度相对要提高很多，可能电路综合不出来或是综合的结果非预期，除非很有经验的工程师，否则建议最好使用HDL设计输入的方式。

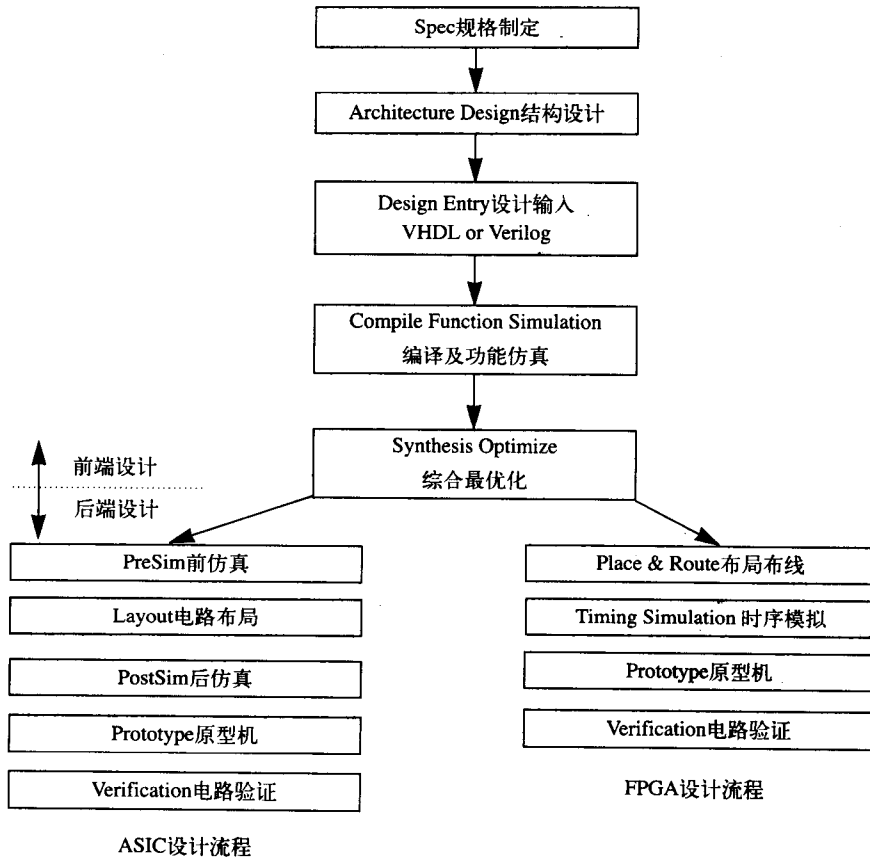


图1-1 FPGA和ASIC设计流程简介

4. 功能仿真 (Function Simulation)

功能仿真最主要是告诉设计者电路功能是否正确，也可以得知设计者所写的HDL程序代码是否能达到设计的目的。功能是否正确，这对设计者还算简单，可能是打字打错或是语法错误等，但是要让你的HDL程序代码能够达到规格所需，就不是这么简单了，必须借助计算机辅助设计工具 (ModelSim、MaxplusII等)。一般设计者在仿真器中看到的结果大部分是波形图，设计者可以设计测试平台 (test bench) 以产生输入信号并送入设计中，从波形图看出是不是设计者想要的结果，如果不是则需修改HDL程序代码，反复地做功能上的仿真，直到设计可以达到规格所需。这里要注意的是，由于功能仿真只能检验功能上是否正确，而不能检验出时序上的错误，时序上的错误要等到做布局布线 (P&R) 之后产生实际的物理模型，才会有延迟的发生，这到后面布局布线部分再来详加探讨。

5. 综合 (Synthesis)

同样也是借助计算机辅助工具 (FPGA express、Synplify等), 简单地说, 就是将设计者所写的HDL (VHDL、Verilog) 综合出电路并且使电路最优化, 也就是说综合出来的电路都是最简化设计。在综合的过程中, 设计者需要制定设计约束 (Design Constraint), 确定工作的频率是多少? 面积是否要最小? 是否并行处理等? 这些都要告诉综合器 (Synthesizer), 综合器会将你的需求转换成所需的设计, 综合器所产生的文件为网表文件 (netlist file), 也就是单元和单元的连接关系。

以上五点为前端设计部分, 后端ASIC和FPGA的设计流程就不尽相同。其中ASIC部分为:

1. 前仿真 (PreSim)

由于综合出来的文件为网表文件, 之后设计者可以利用网表文件来做前仿真的工作, 以完成电路的初步验证。

2. 电路布局 (Layout)

在做完了前仿真之后即可利用计算机辅助设计工具来进行电路布局的工作, 电路布局完成后, 会产生出一个代表实际电路的物理模型, 所以会有延迟现象存在。

3. 后仿真 (PostSim)

由于电路布局后产生实际的物理模型, 在布局后仿真的过程中, 如果和设计者的规格不符, 则必须视错误的情形, 重新更改设计输入, 如HDL程序代码, 并且要重新做综合的工作, 经过反复的模拟验证无误后, 才可以送至工艺设计公司。

4. 原型机 (Prototype)

在完成了设计模拟与验证之后, 即进行原型的制作, 最后得到样品的部分, 再进行最后的电路测试验证部分, 如无误, 即可投入生产。

另外对FPGA的后端设计部分说明如下:

1. 布局布线 (Place & Route)

在综合过后所产生的网表文件, 依照设计者所写的设计约束, 借助计算机辅助设计工具做门级的布局及布线工作。一般在综合时, 设计者所写的设计约束会被综合器转换成布局布线看得懂的指令, 也就是说不用在布局布线中另外写设计约束。如果设计者没有设计约束, 计算机辅助设计工具不知道该布成何种线路, 可能就是默认的线路, 也许就没有达到规格的需求。所以写设计约束可以让布局布线工具知道该变成频率为50MHz, 也就是说在关键路径 (Critical path) 上用较快的组件来完成, 在比较不在乎时序的部分, 用一般组件来达成。

2. 时序模拟 (Timing Simulation)

在做完布局布线之后, 计算机辅助设计工具会产生含有时间延迟的HDL网表文件, 这是设计者所能做的最后一项把关工作。时序模拟要做的不只是在功能仿真上得到的结果, 在做完布局布线之后结果是否正确, 还要验证上一级的输出信号到下一级的建立时间 (setup time) 和保持时间 (hold time) 是否合乎需求。如果不足够, 那么可能得到数据不正确。所以设计者不要忽略时序模拟, 如果不重视它, 若你是反熔丝结构的FPGA设计, 顶多损失一颗几千元的FPGA, 若你的设计是ASIC的话, 那损失的钱可是好几百万的NRE费用了。

1.4 数字电路设计所需考虑的因素

数字电路设计要考虑的因素非常多, 如果不注意, 会导致设计者所设计的系统无法达到规

格要求，一些需注意事项如下所示：

1. 传输延迟

输入信号通过组合逻辑电路（Combinational Logic）而产生出来的输出信号，不可能同步产生，因为在通过组合逻辑电路一定会花费时间，称为传输延迟时间，如图1-2所示。在做时序模拟时，一定都会有传输延迟的现象发生。

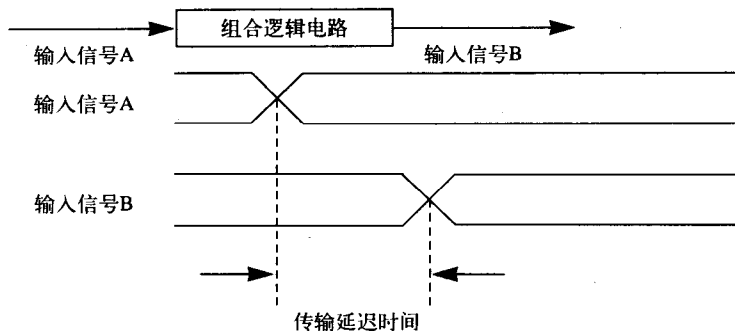


图1-2 传输延迟波形图

2. 门延迟和线延迟

只要输入信号经过一个组件，如与门或是或门，而产生输出信号，必然会产生组件延迟的现象，而门和门间的延迟称之为线延迟（Interconnect delay）。

3. 输入偏斜和输出偏斜

输入输出偏斜最主要发生原因是输入输出的转移时间，也就是高电平变至低电平或是低电平变至高电平所需的时间。注意偏斜时间（Skew time）越短越好，也就是信号能够迅速转换状态。

4. 建立时间和保持时间

信号触发时，输入信号要在触发前的一段时间建立数据，这段时间称为建立时间，而输入信号也要在触发后保持一段时间，这段时间称为保持时间。建立时间和保持时间在电路设计中相当重要，如果不能满足建立时间和保持时间，得出的数据可能会有误，需注意。

5. 脉冲宽度

脉冲宽度就是一个信号脉冲所能维持在一定水平上的宽度，这其中有一个毛刺（Glitch）现象，若你的脉冲宽度小于预设的值时，可以予以忽略。

6. 输出延迟时间

输入信号通过时序（Sequential）电路中时需要注意的是，当触发信号触发时，输出信号会经过一段时间才会生效，这段时间称为输出延迟时间。

7. 工作电压

由于工艺技术日新月异，从原来的0.25mm到现在的0.13mm工艺，工作电压也从原来的 +5V 到现在的 +3.3V 或 +1.8V 等。由于工作电压降低，所以功率也会随之降低。不过要注意的是，通常集成电路的工作电压可以容许到正负10%，也就是说在工作电压为 +5V 时，若使用为 5.1V，IC 也是可以动作的。

8. 工作频率

在规格中，要将整个系统的工作频率大约能到多少先做个预估，在布线布局过后，会产生

出实际的物理模型，也就是会考虑时序的问题。在系统中会有很多延迟发生，有传输延迟、门延迟、线延迟等，这些延迟都会直接影响到工作频率。换句话说延迟越多，工作频率越慢。如果要改善这个问题，需重新思考整个电路的结构，该如何设计才会减少延迟的现象，要是看用流水处理的电路结构还是先进先出的电路结构。

反复进行时序模拟和验证，以保证能够达到规格的要求。

9. 关键路径分析

通常在做电路综合时，计算机辅助设计工具会将你的HDL转换成电路的格式，也就是会产生网表文件，在这之中，会显示关键路径。什么是关键路径呢？简单的说就是时间延迟最长的一条路径，它决定了电路的最高工作频率，所以做关键路径分析是很重要的。设计者应知道如何将时间延迟因素平均分配给电路中的每一个路径，而尽量不要有关键路径产生，以提升工作频率。

10. 功率损耗的问题

如何降低功率损耗是目前很热门问题，为什么要降低功率损耗呢？目的就是为了要省电。由于在不同工作电压功率消耗的情形不同，所以工作电压为 +1.8V时功率损耗比工作电压 +3.3V还低。但是这有个缺点，工作电压越低，时间延迟越长，所以这要做个权衡，是否要牺牲工作频率来换取功率，全看规格所需。当设计出省电的IC应用在手机或是数字个人助理（PDA）时，消费者使用时间就会相对增长，设计出的产品就会比较有竞争力。

11. 扇入和扇出

每个组件都有一定的扇入和扇出能力，若一个缓冲器后面接了二个缓冲器，表示它的扇出能力为2，如图1-3所示。若电路设计超过电路所能承受的扇出和扇入能力，电路可能会正常工作，但是也有可能不会正常工作，也就是说会有不稳定的现象发生。所以在设计电路时需考虑扇入和扇出能力。

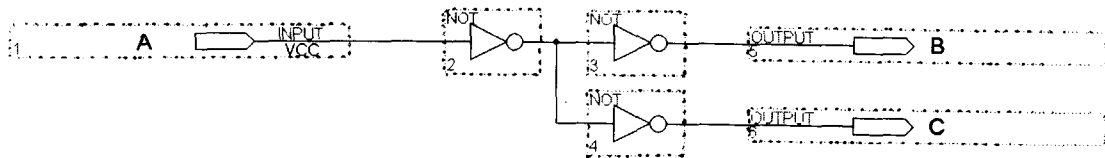


图1-3 扇出为2

第2章 EDA软件介绍

2.1 功能仿真软件Modelsim

2.1.1 ModelSim简介

ModelSim是一套功能相当强大的功能仿真软件，其标识如图2-1所示，当然还有其他如Synopsys、Candence和第三方所开发的仿真软件，在此只针对ModelSim来做说明。ModelSim可以支持硬件描述语言，如VHDL、Verilog，比较特别的是还可支持混合型，即同时包含VHDL和Verilog仿真。

ModelSim的版本又分为ModelSim SE及ModelSim PE，版本差异比较如下所述。

1. ModelSim Special Edition

- 主要针对ASIC及高层的FPGA设计者使用
- 其功能为波形比较 (Wave compare)、高级数据流窗口 (Enhance Dataflow Window)、C-debug、性能分析 (Performance Analyzer) 及程序代码覆盖率分析 (Code Coverage)
- 支持VHDL、Verilog、混合HDL设计

2. ModelSim Personal Edition

- 主要针对FPGA市场
- 支持VHDL、Verilog、混合HDL设计
- 其功能为波形比较

必须注意利用ModelSim为设计平台的HDL设计流程和前面介绍的设计流程可能有些出入，图2-2为其设计流程。

ModelSim为Mentor Graphic公司的功能仿真软件，读者可以从网站上下载ModelSim的试用版，网址为 www.model.com，也可以从网站上找到ModelSim的Flash教学文件，其网址为：

http://www.model.com/products/product_tours/flash/default_CD.html

ModelSim评估版网址为：

<http://www.model.com/products/release.asp>

本章将以Step by Step的方式来介绍ModelSim的使用。

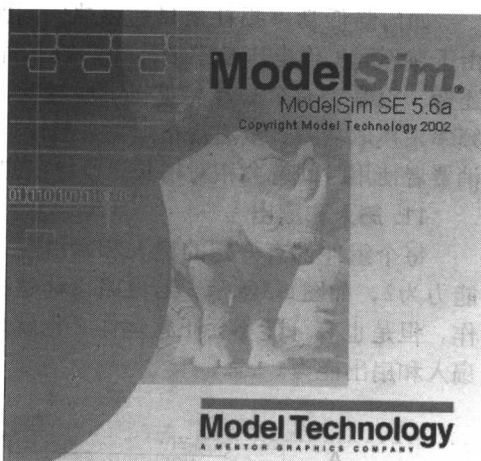


图2-1 ModelSim Logo



图2-2 设计流程