

捷配电子市场

www.dzsc.com

全球领先的电子元器件市场

IC

电子元器件

PDF

技术资料

电子资讯

电子通

技术文章

捷配一下

讨论

微信扫一扫

关注我们

最新型号: CVA1785N CVA2401T CVA2422TL CVA244 CVA3678R CVA373 CVA4508M CVB1151-000 CVC055BE... C

资料首页

技术方案

电路图

设计应用

新品速递

行业标准

基础电子

电子通

当前位置: 首页 >> 技术资料 >> EDA/PLD >> FPGA开发流程：详述每一环节的物理含义和实现目标

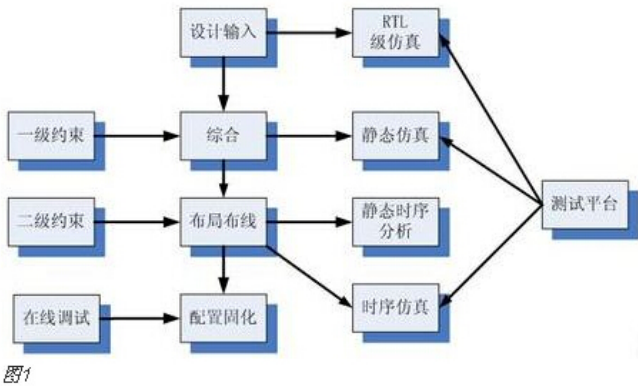
FPGA开发流程：详述每一环节的物理含义和实现目标

文章出处：电子技术设计 发布时间：2015/03/16 | 4421 次阅读

[一站式电子数码采购中心](#) [专业PCB打样工厂，24小时加急出货](#)

要知道，要把一件事情做好，不管是做哪们技术还是办什么手续，明白这个事情的流程非常关键，它决定了这件事情的顺利进行与否。同样，我们学习FPGA开发数字系统这个技术，先撇开使用这个技术的基础编程语言的具体语法、使用工具和使用技巧不谈，咱先来弄清楚FPGA的开发流程是什么。

FPGA的开发流程是遵循着ASIC的开发流程发展的，发展到目前为止，FPGA的开发流程总体按照图1进行，有些步骤可能由于其在当前项目中的条件的宽度的允许，可以免去，比如静态仿真过程，这样来达到项目时间上的优势。但是，大部分的流程步骤还是需要我们循规蹈矩的去做，因为这些步骤的输入是上一个步骤的结果，输出是下一个步骤的输入的关系，这样的步骤就必不可少。



有人看到这个流程图的时候，第一个发自内心的感叹是“啊，怎么这么麻烦啊，特别是之前从事软件开发转过来的。对于他们来讲，很少有接触到一种技术有如此多的环节来实现。但是这并不能说明FPGA开发的具体难度，与软件开发有输入、编译、链接、执行步骤对应的就是设计输入、综合、布局布线、下载烧写，FPGA开发只是为了确保这核心实现主干路每一个环节的成功性加了其他的修饰（约束）和验证而已。下面，我们将以核心主干路为路线，一一介绍每个环节的物理含义和实现目标。

5.1 设计输入

5.1.1 设计输入方式

从图1 FPGA开发流程中的主干线上分离出第一步设计输入横向环节，并做了进一步的细节的处理，如图2，从图上看到，设计输入方式有三种形式，有IP核、原理图、HDL，由此展开设计输入方式的探讨。

IC优质现货资源

品质保障
更放心

稳定库存
更安心

优质商家
更省心

点击搜索现货

排行榜	方案	新品	应用	电路图
1	2017年五种物联网架构解决方案			
2	开关电源EMI辐射超标整改方案			
3	装甲车辆蓄电池实时检测仪研究与实现			
4	汽车电子之安全气囊测试原理及解决方案			
5	模块化UPS分散旁路和集中旁路方案			
6	ON Semiconductor推出适用于家电、游戏机...			

捷多邦 PCB打样

24小时加急出货

服务热线：400-852-8880

每日推荐

如何设计不规则形状的PCB

我们预想中的完整PCB通常都是规整的矩形形状。虽然大多数设计确实是矩形的，但是很多设计都需要不规则形状的电路板，而这类形状往往不太容易设计。本文介绍了如何设计不规则形状的PCB。[全文]

[设计应用] 半桥谐振LLC+CoolMOS开关管电路解析

[设计应用] 运算放大器使用的6个注意事项

[设计应用] 通俗易懂的三极管工作原理

[设计应用] PCB设计经验

[基础电子] 常见总线技术CAN、USART、SPI、SCI

[设计应用] OFDM的基本原理剖析

[基础电子] 编码器基础知识

[技术方案] 汽车电子之安全气囊测试原理及解决方案

[设计应用] 39种电子元器件的检验要求与方法

PDF/DATASHEET

下载基地

PDF阅读器 下载>>

IC热门型号
OPA350U
cm2838gnim25tr
SP5655SB
FMC2A T148
P0118DA
51198-0500
CM1012A-ET
IMP706CPA
LM185AH-1.2

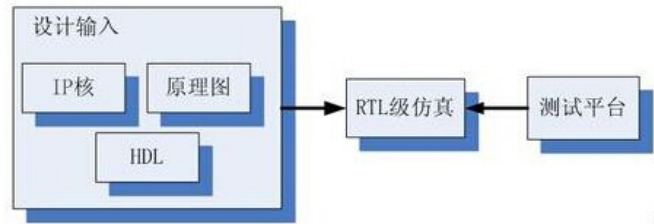


图2

原理图输入

原始的数字系统电路的设计可能大家还不可能想象，是用笔纸一个个逻辑门电路甚至晶体管搭建起来的，这样的方式我们称作原理图的输入方式。那个时候，硬件工程师们会围绕的坐在一块，拿着图纸来讨论电路。幸亏那时候的数字电路的还不是很复杂，要是放到今天，稍微大一点的系统，也算得上是浩大工程，稍微有点电路要修改的话，这个时候你要是一个没耐心或是一个急性子的人可能就就会丧失对这个领域的兴趣。话说回来，那个年代出来的老工程师们，电路基础功夫确实很扎实。

事情总是朝着好的方向发展的，后来出现了大型计算机，工程师们开始将最原始的打孔的编程方式运用到数字电路设计当中，来记录我们手工绘画的电路设计，后来存储设备也开始用上了，从卡片过度到了存储文本文件了，那个时候网表文件大致是起于那个时候。

需要注意的问题是原理图和网表文件的关系，原理图是我们最开始方便我们设计的一个输入方式，而网表文件是计算机传递原理图信息给下一道流程或是给仿真平台进行原理图描述仿真用的。设计输入方式不一样，但是对于功能仿真来讲，最终进度到仿真核心的应该是同一个文件，那么这个文件就是网表文件了。

有了计算机的辅助，数字电路设计起来可以说进步了一大截，但是如果依然全部是基于逻辑门晶体管的话，还是比较繁琐。于是后来出现了符号库，库里包含一些常用的具有通行的器件，比如D触发器类的等等，并随着需求的发展，这些符号库不断的在丰富。与在原理图里利用这些符号库构建电路对应的是，由原理图得到的这个网表文件的描述方式也相应的得到扩展，那么这里网表文件里对电路符号的描述就是最开始的原语了。

作为最原始的数字电路ASIC设计输入的方式，并从ASIC设计流程延续到FPGA的设计流程，有着它与生俱来的优点，就是直观性、简洁性，以致目前依然还在使用。但是需要注意的是，这也是相对的，具体讨论见下一小节。

HDL输入

HDL全称是硬件描述语言Hardware Description Language，这种输入方式要追溯的话得到20世纪90年代初了。当时的数字电路的规模已足以让按照当时的输入方式进行门级抽象设计顾左顾不了右了，一不小心很容易出错，而且得进行多层次的原理图切割，最为关键的是如何能做到在更抽象的层次上描述数字电路。

于是一些EDA开始提供一种文本形式的，非常严谨，不易出错的HDL输入方式开始提供了。特别是在1980年的时候，美国军方发起来甚高速集成电路（Very-High-Speed Integrated Circuit）计划，就是为了在部队中装备中大规模需求的数字电路的设计开发效率，那么这个VHSIC硬件描述语言就是我们现在的VHDL语言，它也是最早成为硬件描述语言的标准。与之相对的是晚些时间民间发起的Verilog， 后来到1995年的时候，它的第一个版本的IEEE标准才出台，但是沿用至今。

前面提到HDL语言具有不同层次上的抽象，这些抽象层有开关级、逻辑门级、RTL级、行为级和系统级，如图3。其中开关级、逻辑门级又叫结构级，直接反映的是结构上的特性，大量的使用原语调用，很类似最开始原理图转成门级网表。RTL级又可称为功能级。

GMK325BJ106KN-T
SN74AHC373N
CL31B102KBCNNNC

最新产品

安森美 ON 1SMA5923BT3 原装进口 现货库存
PTC热敏电阻232266151113VISHAY
安森美 ON 1SMA5922BT3 原装进口 现货库存
德国JUMO久茂压力变送器液压传感器7030...
PCIH49W25M300A1/AA美商宝西POSITRO...
贴片电容2220 104K 100NF 4.7UF 475K 10...
安森美 ON 1SMA5921BT3 原装进口 现货库存
耐高温接近开关批发 | YM-YE138H
PTC热敏电阻232266193145(BC55)VISHAY
贴片电容2220 10UF 106K 105K 1UF 100V ...

热门产品

9.5TV头
5寸液晶电视
5寸数码管
5v稳压二极管
50M晶振
47UF电容
2位拨码开关
250v保险丝
ZP系列整流管
LCD封胶机

IC现货型号

2SA1015
74HC04
BAS16H
MC34063AL
OPA27U
LOC111P
SN74LVC573APWR
LTV-816S-TA1-C
SMAJ7.5CA
SM6T7V5CA

推荐电子百科

CCD芯片
xDSL
三轴加速度传感器
高顶灯
BiCMOS
差模电感
故障测试仪
视频叠加卡
电烙铁温度计
防爆控制箱

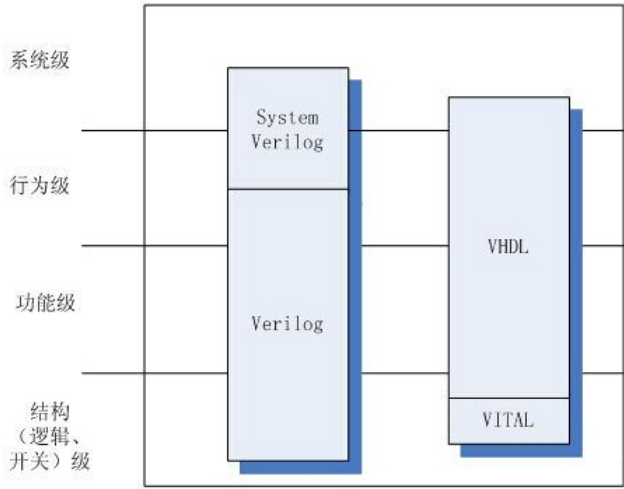


图3

HDL语言除了前面提到的两种外，历史上也出现了其他的HDL语言，有ABEL、AHDL、硬件C语言（System C语言、Handle-C）、System verilog等。其中ABEL和AHDL算是早期的语言，因为相比前面两种语言来讲，或多或少都有些致命的缺陷而在小范围内使用或者直接淘汰掉了。而因为VHDL和Verilog在仿真方面具有仿真时间长的缺陷，System verilog和硬件C语言产生了，从图3看，System Verilog是在系统级和行为级上为Verilog做补充，同时硬件C语言产生的原因还有就是有种想把软件和硬件设计整合到一个平台下的思想。

IP（Intellectual Property）核

什么是IP核？任何实现一定功能的模块叫做IP（Intellectual Property）。这里把IP核作为一种输入方式单独列出来，主要考虑到完全用IP核确实是可以形成一个项目。它的产生可以说是这样的一个逆过程。

在随着数字电路的规模不断扩大的时候，面对一个超级大的工程，工程师们可能是达到一种共识，将这规模巨大而且复杂的设计经常用到的具有一定通用性的功能给独立出来，可以用来其他设计。当下一次设计的时候，发现这些组装好的具有一定功能的模块确实挺好用的，于是越来越多的这种具有一定功能的模块被提取出来，甚至工程师之间用来交换，慢慢大家注意到它的知识产权，于是一种叫做IP知识产权的东西出来了，于是集成电路一个全新领域（IP设计）产生了。

IP按照来源的不同可以分为三类，第一种是来自前一个设计的内部创建模块，第二种是FPGA厂家，第三种就是来自IP厂商；后面两种是我们关注的，这是我们进行零开发时考虑的现有资源问题，先撇开成本问题，IP方式的开发对项目周期非常有益的，这也是在FPGA应用领域章节陈列相关FPGA厂家IP资源的原因。

FPGA厂家和IP厂商可以在FPGA开发的不同时期提供给我们的IP。我们暂且知道他们分别是未加密的RTL级IP、加密的RTL级IP、未经布局布线的网表级IP、布局布线后的网表级IP。他们的含义在后面陆续介绍FPGA的开发步骤的时候，相信大家能够恍然大悟。需要说明的是，越是FPGA靠前端步骤的时候提供的IP，他的二次开发性就越好，但是它的性能可能是个反的过程，同时也越贵，毕竟任何一个提供者也不想将自己的源码程序提供给他者，但是为了不让客户走向其他商家，只能提高价卖了，同时加上一些法律上的协议保护。那么越朝FPGA开发步骤的后端，情况就相反了，越是后端，IP核就会进一步优化，性能就越好，但是一些客户不要的功能就不好去了。

FPGA厂商提供一般常用的IP核，毕竟为了让大家用他们家的芯片，但是一些特殊需要的IP核还是需要付费的。当然这里需要说明的是FPGA厂商的IP是很少可以交叉用的，这一点很容易想，对厂家来讲不会做这种给竞争者提供服务事情的。IP厂商一般会高价的提供未加密的RTL级源码，有时FPGA厂商为了扩大芯片市场占有率，会购买第三方的IP做进一步的处理后免费提给该FPGA使用者。

5.1.2 输入方式使用探讨

在上面我们介绍了三种输入方式，有些地方会讲到第四种输入方式，就是门级网表文件输入的形式，我们这里并没有把它归为一种输入方式，原因在于，本身这些门级网表文件的产生还是源于介绍的三种输入方式中的一种或是几种混合的方式。所以这里没有把它归为一类。

好了，在上面三种输入方式介绍的基础上，我们来探讨一下这令人眼花缭乱的输入方式，探讨的目的就是为了让我们的使用他们。

首先，来总结一下三者的优缺点，其实是两种，因为IP核不管是哪个层次，或者在原理图中被以符号的形式被例化，或者在HDL中被模块例化。所以这里集中探讨的是原理图和HDL的优缺点。原理图的优点就是结构直观性，HDL的优点是严密性、支持甚宽的抽象描述层次、易于移植、方便仿真调试等等，缺点就是不具备对方的优点。当时出现HDL的时候，人们确实是想着原理图该退出历史舞台了，但是到现在它还依然存在着。存在即是有道理的，存在就得用它，但是又得使用HDL，于是存在一种混合编程的形式。除了顶层模块用原理图之外，其他的内部子模块全部使用HDL来描述，HDL描述的模块可以通过工具转换成符号，然后在顶层模块中引用这些符号，这就完成混合编程。

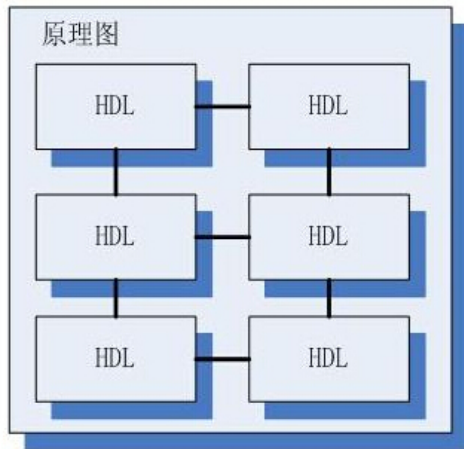


图4

在接触的很多FPGA的初学者很容易被原理图的输入方式给迷惑，甚至爱的深沉，加上本身其他输入方式的繁琐的输入的厌恶，更是爱的无法自拔。当开始强制性要求开始时养成多用HDL输入的习惯的时候，有些甚至有着痛心疾首般的痛苦，但是随着学习的深入，做的东西越来越大，尝到HDL输入方式带来的甜头的时候，就会觉得那个苦没有白吃。

我觉得原理图输入方式从现在的一些线索看来，在今后的某一天将会服役终结。首先是找到了原理图自身带有优势的替代品，那就是主流FPGA集成环境中的综合器和第三方综合器都具有RTL视图生成功能，这个视图完全展示了项目的结构组成，可以上下分层，最大的好处就是可以检查核实写的RTL级代码的综合后电路情况。还有一条线索是，大家用的仿真软件Modelsim并没有提供原理图输入的支持，是原理图的设计必须在集成环境转换成RTL级代码或是综合成网表形式来做仿真，也是一件繁琐的事。原理图的离开只是时间问题。

至于目前HDL选择哪一种比较好，这个问题放到开始将HDL基础语法知识的地方进行探讨。这里要说明的是，并不是我们这里讲Verilog使用就否定其他的HDL语言。各种HDL的争端从未停止过，现在还是依然存在四种开发人，第一种是使用Verilog/System Verilog的人，第二种是使用VHDL的人，第三种就是使用System C的人，第四种是混合型的人，到底哪种好，也有也许是时间问题吧，时间证明一切。

5.2 综合

不管你是采用单一的输入方式，还是采用的是混合编程（这种在很多跨公司合作项目中会碰见，也许A公司用的是VHDL，B公司用的是Verilog，那这个项目中很大可能采用混合型），我们统称得到设计输入后，都得把设计输入得到一个可以和FPGA硬件资源相匹配的一个描述。假设FPGA是基于LUT结构的，那么我们就得到一个基于LUT结构门级网表。在这个过程中，又可以分为如图两个步骤。

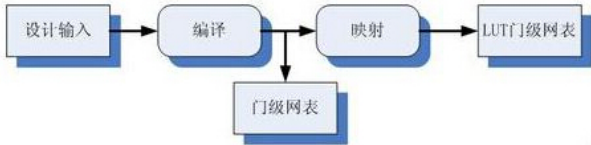


图5

需要说明的是在Altera的开发流程中，将编译、映射过程按照我们叙述的合称综合，而在Xilinx开发流程中，由设计输入得到门级网表的过程叫做综合，而映射过程归结到其叫做实现的某一子步骤中。但是整体的流程还是遵循这个顺序的，只是叫法一些外表性的不一样而已。

5.2.1 编译

原理图、HDL、IP核这些都将通过编译后生成门级的网表，这里生成门级网表的过程其实是早起ASIC的步骤，直接生成门电路网表。这个时候的网表文件和具体的器件无关，也就是说，生成的门电路网表也是一种平台移植的媒质。

5.2.2 映射

我们通过编译得到一张门级网表之后，与早先ASIC开发流程中在这个门级网表布线后去做掩膜不同，接下来就得考虑如何与我们选择的硬件平台结合起来，毕竟我们使用的硬件平台是由一个一个的LUT（假设这类FPGA）组成的。那么这个结合的过程就是映射过程。

这个过程其实很复杂，首先需要把形成的网表逻辑门给规划成一些小的组合，然后再去映射到LUT中，这个过程中规划按照一定的算法和章程进行。不同的算法和章程就会得到不同的映射，不同的映射就会为后面的过程提供不同的选择，最终生成性能不一样的电路了。

我们把讲基于SRAM技术的FPGA的二选一多路器拿出来举个例子，如图6，可以按照红色线将二选一多路器完全劈成两边，原来的一个表就可以规划到其他两个表或表内容中，因为被劈成的两部分可单独成表，也可以被规划到其他电路形成的表里。

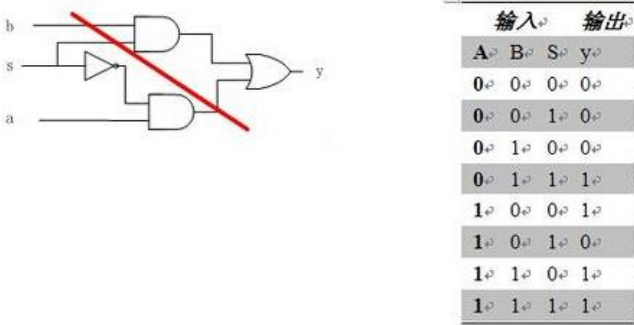


图6

映射的工程比较复杂，运算量也很大，也是为什么FPGA开发过程中，一直存在的一个问题，形成最终的可配置二进制文件的时间非常长，特别是一些大一点的项目，时间消耗比较长的一个点就是映射了，至于具体的映射算法就超出了书的范围了。再强调的是，映射是和器件有关的，即使是同一个系列，不同型号的FPGA内部就够也是有区别的，好比从外观看都是一个单元楼内的单元房，但是每个单元房内装修、家具摆设等都是不一样的。

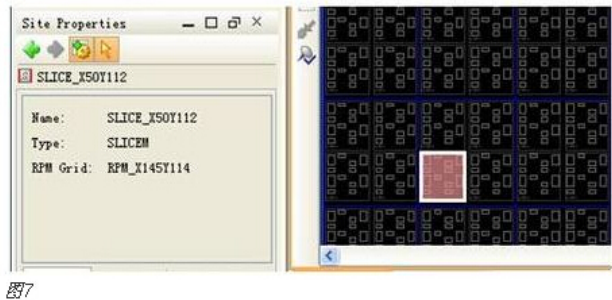
5.3 布局布线

5.3.1 布局

讲到这一块，正好有这么一个例子来讲解这个概念。近来报道朝鲜希望在俄罗斯远东地区租用数十万公顷的土地用来培育农产品。咱先撇开今后的购买的成功与否，假设成功了，并且有了这个详细的希望培育的农作物的品种及数量，有各式各样的蔬菜、主粮、禽类畜养场、果树等等。我们前面做的那些流程得到的LUT门级网表就好比这样的清单。

得到这样的清单之后，我们再假设在这十万公顷的土地上，阳光、水资源、温差都有一定的分布。大家知道农作物的成长和 high 产量或者与阳光有关，或者与水资源有关，或者与温差有关，并且禽类的畜养材料与农作物的副产有关。所以接下来要做的事情就是根据现有的自然条件和农产品的所需环境特点合理布局，哪些地适合做什么。

紧接着我们回到FPGA开发中来，我们通过前面的步骤得到的清单就是LUT门级网表。网表里提供的仅仅是从逻辑关系上一些LUT结构的连接。我们需要将这些LUT结构配置到FPGA具体的哪个位置。需要说明的是，FPGA里任何硬件结构都是按照横纵坐标进行标定的，图中选中的是一个SLICE，SLICE里面存放着表和其他结构，它的位置在X50Y112上。不同的资源的坐标不一样，但是坐标的零点是公用的。



在FPGA里布局需要考虑的问题是，如何将这些已有的逻辑上连接的LUT及其它元素合理的放到现有的FPGA里，达到功能要求的时候保证质量。具体点就比如，乘法器这样的电路适合放在RAM附近，当然，硬件乘法器的硬件布局一般也是在存储器附近，有利于缩短乘法的延时时间，什么样的电路需要配置高速等等。

十万公顷的地布局规划好了，农产品就会有很好的丰收，同样FPGA开发布局布好了，由FPGA搭建起来的电路就会更加稳定和扩展性。

5.3.2 布线

上小节中，我们把十万公顷的地给安排好了，哪些地该种什么。具体实施之前还有一些是必须做的，比如农作物的浇灌，没有一个很好的灌溉系统是一个问题；再比如丰收了得采摘吧，这个时候，能够让大卡车到达每一块农地的公路枢纽也是需要解决的问题。将每一块或者相关的田地连接的灌水系统和公路的建设，就好比我们这个布线的过程。

我们在FPGA内通过布局，知道那些LUT具体分布到哪个SLICE，但是一方面如何让这些SLICE连接起来，二方面如何让输入的信号到达相应的开始处理点和如何让输出到达输出IO上，并且连接的电路整体性能好，这就是布线这个环节需要完成的内容。要达到布线最优话，当然这里面设计到布线算法和很多细节问题，比如涉及到布线资源、PLL资源分布。但是这些对我们理解布线这个概念没有很多益处，暂且不深入，本质上就是一个线路求最优的问题。

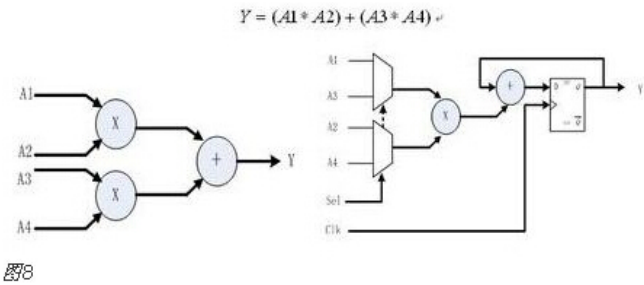
5.4 约束

约束，在图1上看到，在综合和布局布线这两个流程环节里都出现了，我们暂且规定其为约束一和约束二，或者说综合约束和布局布线约束，布局布线约束又可以分为位置约束、时序约束。约束，就是对这些环节操作定制规则。一般开发环境会对这些约束有个默认，这些默认的设置对大部分情况下还是适用的，但是通常布局布线约束中的I/O约束是我们每一个工程都必须给定的。同时开发工具开放其他约束接口，允许我们设置这些规则，具体的有哪些约束怎么去做在后面介绍工具使用的时候进行讨论，这里先明白这些约束的基本概念。

综合约束

相信大家已经下意识的将综合约束和综合过程挂在一起了，没错，综合约束确实是在综合过程中做的，用来指导综合过程，包括编译和映射。我们已经知道综合过程是将RTL级电路描述转换到FPGA上的硬件单元（LUT）中，形成以FPGA存在的硬件单元构成的电路。

我们还是拿前面有过的例子来说明，不同的约束将导致生成性能不同的电路。综合这么一个完成式***能的电路，没有加资源共享得到的电路如图8左边所示的电路，而加了资源共享的约束后，得到的电路结构如图8右边的电路。



通过之前的分析，得到左边的电路结构资源消耗多但是速度快，而右边的结构消耗资源少，但是速度慢，乘法器需要分时复用。

当然这只是一个例子，但是足以说明，不同的综合指导原则也就是综合约束，将会产生不同的电路。当得到的电路性能不能满足需求的时候，适当考虑综合约束，来达到一个速度和面积的转换的效果，实现性能的提升。电路实现的速度和消耗的面积是贯穿在FPGA开发过程中两个矛盾的问题，综合约束是其中一种小范围内实现速度和面积平衡点移动的方式。

位置约束

没错，你又想对了，位置约束和我们布局有关系，它就是指布局的策略。根据所选择的FPGA平台现有硬件资源分布来决定我们布局。

其中最典型的位置约束就是I/O约束。一个典型的系统是既有输入也有输出的，而不管是输入还是输出，都是从I/O上为端点的。输入从哪个端点进来，输出从哪个端点出去，输入是需要支持什么样的电气特性的端点，输出又是需要支持什么样的电气特定的端点。这些都是I/O约束做的事情。任何一个工程，都必须有这么一个约束。

还有一种典型的位置约束是在增量编译里涉及的物理界定。增量编译的出现就是因为FPGA开发过程中综合和布局布线的长耗时性而提出的。思想就是把FPGA切成很多个小块的FPGA，然后约定具体哪块小FPGA放置什么模块，实现什么样的功能，从物理上进行界定。当修改工程后，开发平台就会检测哪些小FPGA内没有进行修改，哪些进行了修改，然后将修改过的部分重新进行综合布局布线步骤。这样一来，相比原来修改一点，全工程重新经过那些过程来讲，时间节省下来了。

时序约束

估计没有多少悬念了，时序约束很大程度上和布线有关。为什么要做这个约束？

由于一方面信号在芯片内传递是需要消耗时间的，另一方面大量存在的寄存器有反应时间，而我们开发的最开始的时候这些时间都是理想化的。但是考虑到真实情况下，如果跑的速度比较高，达到了200M这么个速度，当然这个高速和具体的芯片有关，高性能的芯片本身跑的速度可以达到很高，200M相对来说就不是高速，对一些低性能芯片还可能达不到200M。这个时候，这些时间达到了同样一个系统时间数量级的时候，很可能影响电路的性能了。某一刻，该来的信号没有来，默认的话就会采集错误信号了。

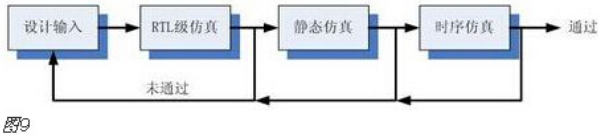
为了让这些硬件本身带来的延时时间更理想化，我们就要对这些决定时间延时的因素优化来减少时间延时。对于寄存器本身的反应时间这个因素我们开发者是无能为力的，我们要做的优化就是布线了。是走直线还是走其他，不仅仅决定于自身这条路径，还和整个系统布线有关，好比水桶原理，系统性能决定于最差路径延时。

时序约束做的就是这些事情，但是时序约束并不是指具体去连接每一条线，这个工作就像前面那些流程一样都是由软件去实现的，先用软件自己默认原则布线，然后对其结果分析，不满足时序要求的，我们再将具体的问题路径做一些指导约束。时序约束的添加，主要包括周期约束、输入偏移约束和输出偏移约束。具体的过程在后面章节介绍工具使用时会有具体动手的指导。

5.5 FPGA开发仿真

在经过上面从设计输入到综合再到布局布线过程的介绍后，我们来集中探讨一下，在这些过程中涉及到的相应的仿真。

仿真，字面上讲就是模拟真实状况。我们FPGA设计里面的仿真，就是模拟真实电路的状况，查看电路是不是我们需要的电路。如果我们把FPGA开发形成电路当作一个产品的生产过程，那么在FPGA开发流程中含有的三种仿真（RTL级仿真、静态仿真和时序仿真）就好比产品线中的三道检测站。如图9，这三道工序任何一道出了问题，修改设计后都得重新走这三道卡，所以尽量在把问题发现在源头。



5.5.1 测试平台

所谓testbench,即测试平台，详细的说就是给待验证的设计添加激励，同时观察输出响应是否符合设计要求。测试平台，测试平台在做功能仿真、静态仿真和时序仿真的时候都需要用到。刚开始的对于一些初学者，遇到的都是一些简单的东西，测试平台相应的也很简单，用一个文件就可以很清晰的呈现测试结构。对于一些复杂的项目，测试就没有那么简单了，由此还专门产生一个行业——测试行业。这个时候我们要用到一个概念就是结构化测试。

一个完整的测试平台如下图10所示，它是分结构组成的，其中对设计测试结果的判断不仅可以通过观察对比波形得到，而且可以灵活使用脚本命令将有用的输出信息打印到终端或者产生文本进行观察，也可以写一段代码让它们自动比较输出结果。

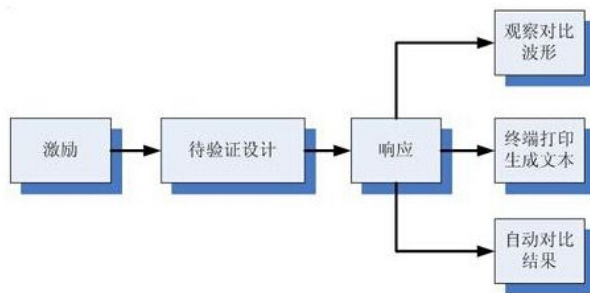


图10

测试平台的设计是多种多样，可以使用灵活的Verilog的验证脚本，但是它也是基于硬件语言但是又服务于软件测试的语言，有时并行有时顺序，只有掌握这些关键点，才能很好服务测试。需要说明的一点是，不管大家是已经在用Verilog在写测试平台还是刚学习写测试平台，那么建议大家还是能用到System Verilog中相对Verilog新的语法还是尝试的用，System Verilog是一种趋势，它本身就是向下兼容的第三代的Verilog。

5.5.2 RTL级仿真

这里RTL级仿真属于第一道检测，有些场合称作功能仿真，为了突出和后面的静态仿真的区别，以免在后面介绍静态仿真的时候大家弄的很头大，我们还是这样称呼。它是对工程在寄存器转送级进行的描述时进行测试，查看其在RTL级描述的时候实现的功能的正确性。

关于RTL级仿真，如果设计中设计到原理图输入的话，在一些仿真工具中是不支持的，比如Modelsim，这个时候要进行功能上的验证，可以将原理图转换成HDL描述，或者直接把整个工程转换成LUT门级网表后进行后面要讲到的静态仿真完成。

所有逻辑功能的验证都希望在RTL级做，尽可能的将问题发现在RTL级仿真过程中，减少后面发现问题带来的反复。

5.5.3 静态仿真

静态仿真，有些地方给的外号叫门级仿真，确切的说应该是综合后的LUT门级网表。是在综合过程后做的仿真。有些开发平台下将静态仿真具体又分为编译仿真和映射仿真，比如ISE就是这样做的，但是个人觉得应该很少场合做这个编译仿真。静态仿真的目的就是验证当工程到了用LUT门级网表描述的时候，从功能上检查验证工程的正确性。

不管是Altera还是Xilinx的开发平台，都直接支持静态仿真，但是由于各自厂家的仿真器专业性不强，我们还是用第三方仿真工具比较多。这时候在第三方工具下的输入必须是经过综合工具综合出来的涵盖工程所有信息的LUT门级网表文件了。一般专业一点的第三方综合工具是不具备综合功能的，至少我们在使用Modelsim的时候，并没有要求我们添加工程用到的具体的哪一款型号FPGA的信息。这也是静态仿真的外号门级仿真指的是LUT门级网表仿真的依据。

5.5.4 时序仿真

时序仿真是在布局布线之后做的，在前面介绍时序约束的时候讲到，布线延时问题影响到了电路的性能的时候可以做时序约束。那么这个延时问题的获得就可以通过时序仿真获得，当然还有一种获得延时出现超载情况，这个属于下面小节介绍的静态时序分析了。

一般情况下，电路进行完布线过程后，会生成一个延时信息文件，我们简称SDF(standard delay format)文件，Quartus平台下是以.sdo文件形式存在的。里面含有三种延时信息，分别为最小值、典型值、最大值，存在的形式是最小值：典型值：最大值，一般缩写min:typ:max。这里也体现了，在FPGA里的延时信息是不能够精确获得的，只能是逼近，因为本身同一器件中，不同的区域的逻辑门也很可能和其他的区域内同种的逻辑门的延时不一样。我们这里举一个例子来说明一下这三种值的含义。

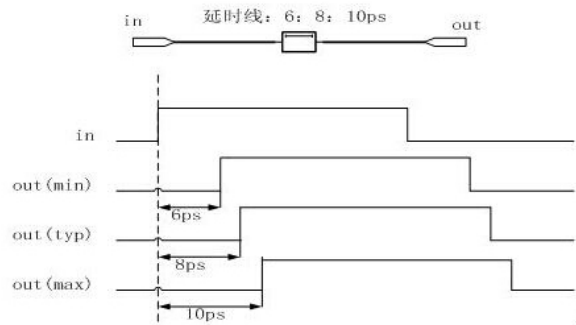


图11

如上图，这是一个描述一段延时线的延时信息，给的延时信息从in端点到out端点，输入发生跳变后，分别以最小值、典型值和最大值将信号跳变传递到out端点。我们这里只是一段延时线，在延时信息文件里还有一类延时信息，就是一些具有逻辑功能的cell延时，这个时候信号跳变又分为由高变低和有低变高，因为这两种跳变在这些器件里的三种延时值是不一样的，得分别探讨，具体分别以某一种情况类推。

在做后仿真的时候，只需要在做完静态仿真后的基础上添加布线的延时信息后，再分析逻辑功能是否满足要求。后方针的平台 使用情况和前面一样，一般采用第三方仿真工具，典型的是Modlesim，具体操作过程见软件相关操作章节。

5.6 静态时序分析

静态时序分析，简称STA（Static Timing Analysis），这个过程做的话一把是在是做后仿真前做的。在布局布线后，会生成时序分析报告，该报告是分析工具利用从布线的路劲上提取出寄生参数后精确计算出来的。该报告中会提示出一些关键路径，所谓关键路径就是指延时信息比较突出的信号节点流，通过分析可以得到不满足时序要求的路径，这个过程就是STA过程。

静态时序分析的特点就是不需要输入向量就能穷尽所有的路径，且运行速度很快、占用内存较少，不仅可以对芯片设计进行全面的时序功能检查，而且还可利用时序分析的结果来优化设计。很多设计都可以在功能验证的成功的基础上，加上一个很好的静态时序分析，就可以替代耗时非常长的后仿真了，这是一种很有保障性的化简流程方式。后仿真相对静态时序分析来说还具有逻辑验证，在加上延时信息的基础上分析逻辑。

5.7 在线调试

在线调试也称作板级调试，它是将工程下载到FPGA芯片上后分析代码运行的情况。有人会以为，我们不是已经做了仿真了，甚至是时序仿真都通过了，还会存在问题么？在实际中，存在这么些情况我们需要用到在线调试：

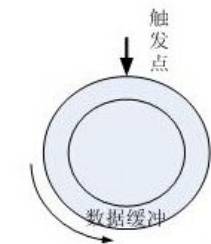


图12

仿真不全面而没有发现的FPGA设计错误。很多情况下，由于太复杂，无法做到100%的代码覆盖

率；

在板级交互中，存在异步事件，很难做仿真，或者仿真起来时间很长，无法运行；

除了本身FPGA外，还可能存在板上互连可靠性问题、电源问题和IC之间的信号干扰问题，都可能导致系统运行出错；

其他潜在问题。

在线调试的方式主要有两种，一种是利用外部测试设备，把内部信号传送到FPGA针脚上，然后用[示波器](#)或者逻辑分析仪观察信号；另一种就是利用嵌入式逻辑分析仪，在设计中插入逻辑分析仪，利用JTAG边缘数据扫描和开发工具完成数据交互。

嵌入式逻辑分析仪的原理相当与在FPGA中开辟一个环形存储器，存储器的大小决定了能够查看的数据的深度，是可以人为设定的，但是不得超出资源。在FPGA内部，根据设置的需要查看的信号节点信息和驱动的采样时钟，对信息进行采样，并放置到设定的存储空间里，存储空间是环形的，内容随时间更新。然后通过判断触发点来检查采集数据，一旦满足触发条件，这个时候会停止扫描，然后将触发点前后的一些数据返回给PC端的测试工具进行波形显示，供开发者进行调试。

目前的调试工具都是和本身的FPGA开发平台挂钩的，不同FPGA厂商都会有开发软件平台，嵌入式逻辑分析仪也就不同。Altera 厂家提供的是SignalTapII，而 Xilinx厂家提供的是 ChipScope，这些工具的具体使用在后面工具中详解。

当然这里除了嵌入式逻辑分析仪外，各厂家还提供了一些其他的在线调试工具，例如SignalProbe等等，但是或多或少的用的人不是很多，有兴趣的可以找到该功能使用的说明手册。

5.8 配置及固化

好了，到了我们最后一个环节就可以完成FPGA的流程了。这一部分我们分四个小节来讲，首先是针对大家很多人不是太清楚的FPGA配置过程安排的，随后一节为了更加深理解，举了altera 的FPGA叙述配置全过程，第三小节是探讨FPGA主要的配置模式，最后一节就是正对这些配置模式展开的对比选择探讨。

5.8.1 FPGA配置过程

在FPGA正常工作时，配置数据存储于SRAM中，这个SRAM单元也被称为配置存储器（configure RAM）。由于SRAM是易失性存储器，因此在FPGA上电之后，外部电路需要将配置数据重新载入到芯片内的配置RAM中。在芯片配置完成之后，内部的寄存器以及I/O管脚必须进行初始化（initialization），等到初始化完成以后，芯片才会按照用户设计的功能正常工作，即进入用户模式。

FPGA上电以后首先进入配置模式（configuration），在最后一个配置数据载入到FPGA以后，进入初始化模式（initialization），在初始化完成后进入用户模式（user-mode）。在配置模式和初始化模式下，FPGA的用户I/O处于高阻态（或内部弱上拉状态），当进入用户模式下，用户I/O就按照用户设计的功能工作。

5.8.2 举例——altera FPGA配置全过程

一个器件完整的配置过程将经历复位、配置和初始化等3个过程。FPGA正常上电后，当其nCONFIG管脚被拉低时，器件处于复位状态，这时所有的配置RAM内容被清空，并且所有I/O处于高阻态，FPGA的状态管脚nSTATUS和CONFIG_DONE管脚也将输出为低。当FPGA的nCONFIG管脚上出现一个从低到高的跳变以后，配置就开始了，同时芯片还会去采样配置模式（MSEL）管脚的信号状态，决定接受何种配置模式。随之，芯片将释放漏极开路（open-drain）输出的nSTATUS管脚，使其由片外的上拉[电阻](#)拉高，这样，就表示FPGA可以接收配置数据了。在配置之前和配置过程中，FPGA的用户I/O均处于高阻态。

在接收配置数据的过程中，配置数据由DATA管脚送入，而配置时钟信号由DCLK管脚送入，配置数据在DCLK的上升沿被锁存到FPGA中，当配置数据被全部载入到FPGA中以后，FPGA上的CONF_DONE信号就会被释放，而漏极开路输出的CONF_DONE信号同样将由外部的上拉电阻拉高。因此，CONF_DONE管脚的从低到高的跳变意味着配置的完成，初始化过程的开始，而并不是芯片开始正常工作。

INIT_DONE是初始化完成的指示信号，它是FPGA中可选的信号，需要通过Quartus II工具中的设置决定是否使用该管脚。在初始化过程中，内部逻辑、内部寄存器和I/O寄存器将被初始化，I/O驱动器将被使能。当初始化完成以后，器件上漏极开始输出的INIT_DONE管脚被释放，同时被外部的上拉电阻拉高。这时，FPGA完全进入用户模式，所有的内部逻辑以及I/O都按照用户的设计运行，这时，那些FPGA配置过程中的I/O弱上拉将不复存在。不过，还有一些器件在用户模式下I/O也有可编程的弱上拉电阻。在完成配置以后，DCLK信号和DATA管脚不应该被浮空（floating），而应该被拉成固定电平，高或低都可以。

如果需要重新配置FPGA，就需要在外部将nCONFIG重新拉低一段时间，然后再拉高。当nCONFIG被拉低时，nSTATUS和CONF_DONE也将随即被FPGA芯片拉低，配置RAM被清，所有I/O都变成三态。当nCONFIG和nSTATUS都变为高时，重新配置就开始了。

5.8.3 配置模式

这一块分成两部分，一部分是在线调试配置，另一块是固化，即将工程配置到相应存储单元中，上电后，通过存储在存储器中的内容配置FPGA。

在线配置

第一部分在线调试配置过程是通过JTAG模式完成的，如图13所示，在JTAG模式中，PC和FPGA通信的时钟为JTAG接口的TCLK，数据直接从TDI进入FPGA，完成相应功能的配置。

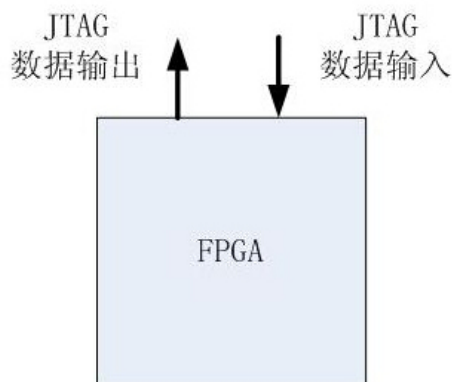


图13

JTAG接口是一个业界标准接口，主要用于芯片测试等功能。FPGA基本上都可以支持JTAG命令来配置FPGA的方式，而且JTAG配置方式比其他任何方式优先级都高。JTAG接口有4个必需的信号TDI，TDO，TMS和TCK以及1个可选信号TRST构成，其中：

TDI，用于测试数据的输入；

TDO，用于测试数据的输出；

TMS，模式控制管脚，决定JTAG电路内部的TAP状态机的跳变；

TCK，测试时钟，其他信号线都必须与之同步；

TRST，可选，如果JTAG电路不用，可以讲其连到GND。

固化

第二部分固化程序到存储器中的过程可以分为两种方式，主模式和从模式。主模式下

FPGA器件引导配置操作过程，它控制着外部存储器和初始化过程；从模式下则由外部计算机或控制器控制配置过程。主、从模式从传输数据宽度上，又分别可以分为串行和并行。

(1)、主串模式

主串模式是最简单的固化模式，如图14所示,这个模式过程不需要为外部存储器提供一系列地址。它利用简单的脉冲信号来表明数据读取的开始，接着由FPGA提供给存储器时钟，存储器在时钟驱动下，将数据输入到FPGA Cdata_in端口。

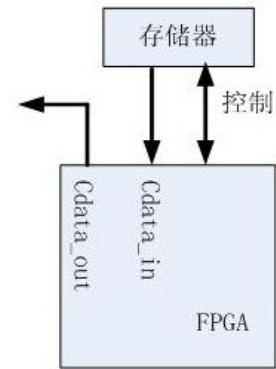


图14

(2) 主并模式

主并模式其实和主串模式的一样机理，只不过是在主串的基础上，同周期数内传送的数据变成8位，或者更高，如图15。这样一来，主并行相比主串行的数度要优先了。现代有些地方已采用这种方式来配置FPGA的了。

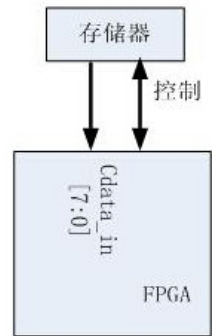


图15

(3) 从并模式

从上面看到，主模式下的连接还是很简单的。但是有时候，系统可能用其他微处理器来对FPGA进行配置。这里的微处理器可以指FPGA内嵌的处理器，比如说Nios。微处理器控制着何时配置FPGA，从哪读取配置文件。如图16，这种方式的优点是处理器可以灵活随时变更FPGA配置，同时配置的速度也快。微处理器先从外部存储设备里读取一个字节的数，然后写到FPGA里。

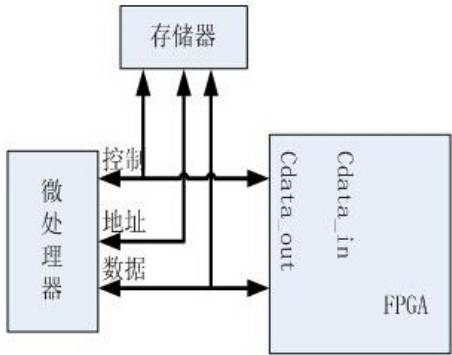


图16

(4) 从串模式

理解了从并模式，从串模式就不用很多解释了，它的特点就是节约FPGA管脚I/O。

(5) 多片级联

多片模式有两种，一种是采用菊花链的思想，多片FPGA共享一个存储器，另外一个是可以使用其他存储器配置不同的FPGA。如果所示是一个共享型的结构，显示启动了。这里分主FPGA和从FPGA，主FPGA和存储器是使用串行主模式来配置，而后面那个的配置是通过第一配置好的FPGA上微处理器进行协调的。

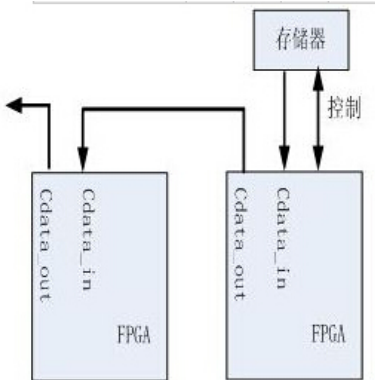


图17

5.8.4 模式选择

现今FPGA应该可以支持上面五种配置模式，是通过3个模式引脚来实现的，具体的映射如下表，在今后模式还是有可能增加的。

模式引脚	模式
0 0 0	主串模式
0 0 1	从串模式
0 1 0	主并模式
0 1 1	主串模式
1 x x	仅用JTAG模式

在PS模式下，如果你用电缆线配置板上的FPGA芯片,而这个FPGA芯片已经有配置芯片在板上,那你就必须隔离电缆线与配置芯片的信号。一般平时调试时不会把配置芯片焊上的,这时候用电缆线下载程序。只有在调试完成以后,才把程序烧在配置芯片中,然后将芯片焊上.或者配置芯片就是可以方便取下焊上的那种。这样出了问题还可以方便地调试.。

对FPGA芯片的配置中，可以采用AS模式的方法，如果采用EPCS的芯片，通过一条下载线进行烧写的话，那么开始的"nCONFIG,nSTATUS"应该上拉，要是考虑多种配置模式，可以采用跳线设计。让配置方式在跳线中切换,上拉电阻的阻值可以采用10K 一般在做FPGA实验板的时候,用AS+JTAG方式,这样可以用JTAG方式调试,而最后程序已经调试无误了后,再用AS模式把程序烧到配置芯片里去。

5.9 开发工具总结

在围绕图1把FPGA开发流程讲完后，这里对每个环节中设计的相关软件进行总结，如下表所示。毕竟充分利用各种工具的特点，进行多种EDA工具的协同设计，对FPGA的开发是非常重要的。充分利用了这些EDA工具的优点，能够提高开发效率和系统性能。

公司	软件	说明
Altera	Quartus II	集成开发环境
	SOPC Builder	嵌入式系统开发工具
	DSP Builder	数字信号处理开发软件
	SignalTap II	嵌入式逻辑分析仪
Xilinx	ISE	集成开发环境
	EDK	嵌入式系统开发工具
	System Generator	数字信号处理开发软件
	ChipScope	嵌入式逻辑分析仪
Lattice	IspLever	集成开发环境
Actel	Libero IDE	集成开发环境
Mentor Graphics	Modelsim	仿真软件
Aldec	ActiveHDL	仿真软件
Synplicity	Synplify、Synplify Pro	综合工具
Synopsys	FPGA Compiler II	综合工具
其他

表中列出的每种EDA工具都有自己的特点。一般由FPGA厂商提供的集成开发环境，如Altera Quartus II和Xilinx ISE，在逻辑综合和设计仿真环节都不是非常优秀，因此一般都会提供第三方EDA工具的接口，让用户更方便地利用其他EDA工具。为了提高设计效率，优化设计结果，很多厂家提供了各种专业软件，用以配合FPGA芯片厂家提供的工具进行更高效的设计。

比较常见的使用方式是：FPGA厂商提供的集成开发环境、专业逻辑仿真软件、专业逻辑综合软件一起使用，进行多种EDA工具的协同设计。比如Quartus II+ModelSim+FPGA Compiler II，ISE+ModelSim+Synplify Pro等等。

技术资料出处:电子技术设计
该文章仅供学习参考使用，版权归作者所有。

因本网站内容较多，未能及时联系上的作者，请按本网站显示的方式与我们联系。 

相关技术资料：

[基于FPGA的DDR3多端口读写存储管理系统设计](#)

机载视频图形显示系统主要实现2D图形的绘制，构成各种飞行参数画面，同时叠加实时的外景视频。由于FPGA具有强大逻辑资源、丰富IP核等优点，基于FPGA的嵌入式系统架构是机载...

[用FPGA来加速采用OpenCL的多功能打印机图像处理](#)

摘要：在高性能计算、娱乐和科学计算市场，OpenCL的采用在持续增长。OpenCL的灵活性和便携性使之成为了一个开发图像处理应用的优秀平台。然而，OpenCL尚未应用到硬拷贝打印...

[FPGA四大设计要点解析及应用方案集锦](#)

本文叙述概括了FPGA应用设计中的要点，包括，时钟树、FSM、latch、逻辑仿真四个部分。FPGA的用处比我们平时想象的用处更广泛，原因在于其中集成的模块种类更多，而不仅...

[基于FPGA 的太阳能并网逆变器的研究](#)

系统概述 新能源发电成为21世纪解决能源危机的必经出路，光伏发电、风电、核电等新能源发电是目前新能源发电研究的几大方向。这几种新能源各有各的特点，我们选择了最靠...

[基于FPGA和LCOS技术的3D视频前端处理与显示系统设计](#)

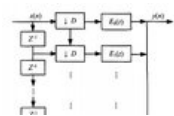
目前，3D显示已逐渐融入到人们的生活当中，因其能再现真实的三维场景，受到了人们的广泛热爱，红蓝3D图像获取的基本原理如图1所示。摄像头采集同一场景的两路视频图像信号...

顶我一下

11次

为您推荐

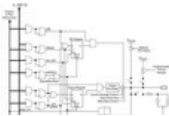
换一换



宽带数字下变频器的FPGA实现



80C196KC供应商_80C196KC



详解FPGA芯片结构以及开发流程



从硬件角度讨论FPGA开发框架

【关闭】 【推荐】 【打印】 【收藏此页】

分享：[QQ空间](#) [人人网](#) [开心网](#) [腾讯微博](#) [新浪微博](#) [搜狐微博](#) [网易微博](#)

上一篇：[基于FPGA的DDR3多端口读写存储管理系统设计](#)
下一篇：[宽带数字下变频器的FPGA实现](#)

发表评论

本文已有(0)条评论

发表技术资料评论，请使用文明用语

验证码： 0716

字符数不能超过255

用户评论

暂且没有评论!