





手把手教你做一款MR智能眼镜


 FPGA硬件主板

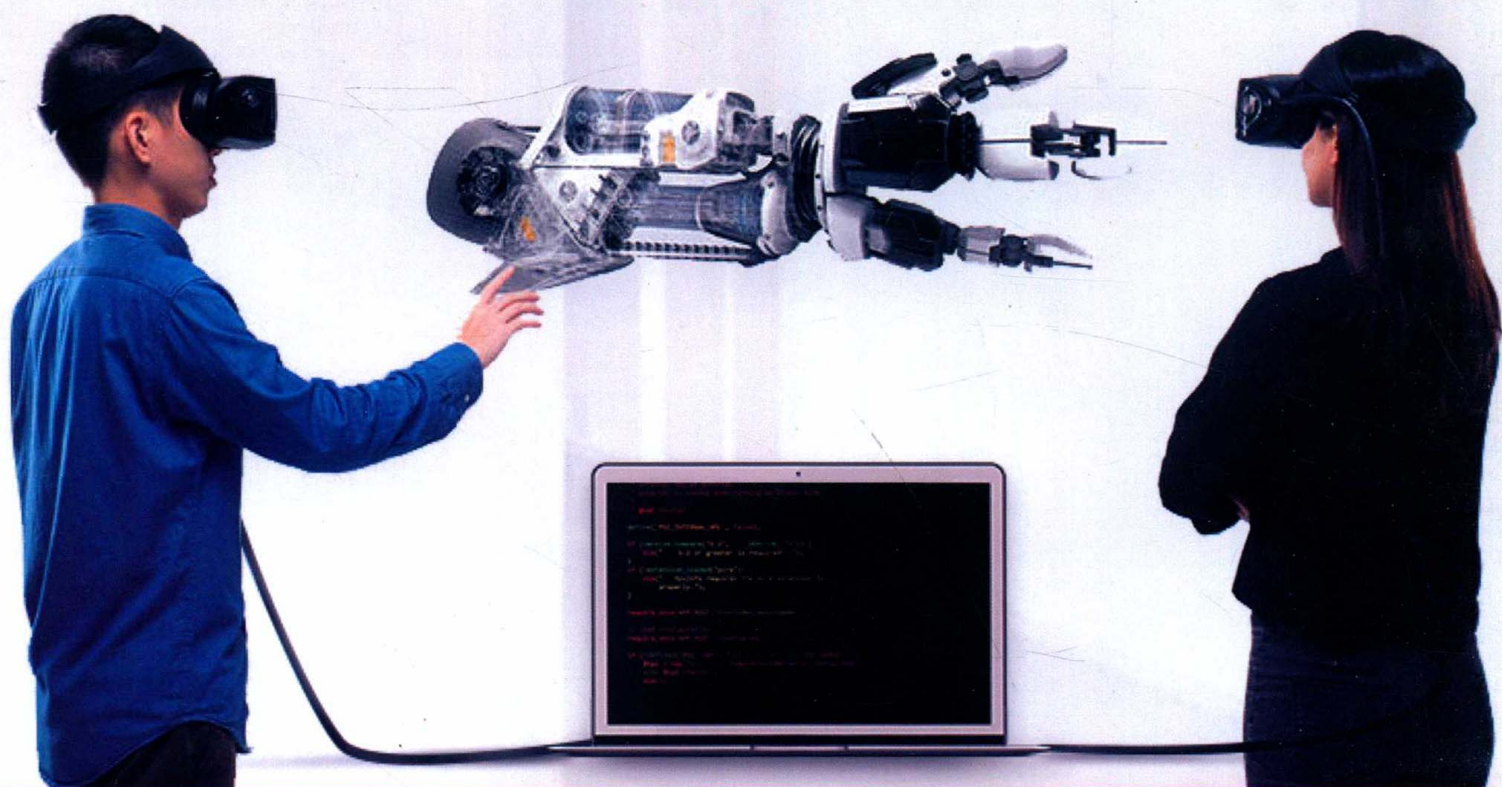
 图像采集与处理算法

 双目校准算法


 视频透视算法

 头显位置跟踪

 定位算法



MR 智能眼镜开发实战

疯壳团队 张基隆 陈至钊 崔为初 艾韬 编著 

 西安电子科技大学出版社
<http://www.xduph.com>

手把手教你做一款MR智能眼镜

MR 智能眼镜开发实战

疯壳团队 张基隆 陈至钊 崔为初 艾 韬 著

西安电子科技大学出版社

内 容 简 介

VR 和 AR 作为当前计算机科学应用领域的新热点,吸引了大批开发者的关注,相关产品也在迅速进入市场。介导现实(Mediated Reality, MR)是由“可穿戴技术之父”——多伦多大学教授 Steve Mann 提出的智能技术,它通过数字化现实与数字化画面达到完全的数字化视觉感知,完整地包含了虚拟现实与增强现实。

本书描述了 MR 智能眼镜研发的知识点、开发过程及实现步骤,包含了硬件及软件内容开发的所有部分,完整地介绍了 MR 包含的知识体系及实践指导,并提供了手把手的样例教学及代码。总的来说,MR 智能眼镜的研发包括了电路板设计、FPGA 设计与开发、驱动开发、计算机视觉软件开发、计算机图形学软件开发等内容。全书共 3 章,第 1 章和第 2 章讲述了开发准备和开发基础,第 3 章详细介绍了开发过程、工作代码等。

对于希望从事 VR/AR/MR 行业研发工作的在校大学生、开发者来说,这是一本入门教材;对于 VR/AR/MR 相关行业从业者来说,本书具有一定的参考与指导价值。

本书配套的源码、视频、套件以及书中所有链接内容都可以通过疯壳网站(https://www.fengke.club/GeekMart/su_fsqqeP0Z.jsp)获取。

图书在版编目(CIP)数据

MR 智能眼镜开发实战 / 疯壳团队等著. —西安: 西安电子科技大学出版社, 2018.12

ISBN 978-7-5606-5144-6

I. ① M… II. ① 疯… III. ① 移动终端—智能终端—开发 IV. ① TP334.1

中国版本图书馆 CIP 数据核字(2018)第 255720 号

策划编辑 高樱

责任编辑 黄菡 阎彬

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2018 年 12 月第 1 版 2018 年 12 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 6.5

字 数 149 千字

印 数 1~3000 册

定 价 25.00 元

ISBN 978-7-5606-5144-6 / TP

XDUP 5446001-1

如有印装问题可调换

前 言

虚拟现实简称 VR，是指利用电脑模拟产生一个三维的虚拟世界，提供用户关于视觉等感官的模拟，让用户身临其境，可以及时地、没有限制地观察并感知三维空间内的事物。自从 Facebook 于 2014 年 3 月以 20 亿美元收购 Oculus VR 以来，VR 技术得到了迅速发展，一定程度上成为 2014 至 2016 年的科技圈热点，并逐渐进入了商用及个人消费领域。

增强现实简称 AR，是指透过摄影机影像的位置及角度精算并加上图像分析技术，让屏幕上的虚拟世界能够与现实世界场景进行结合与互动的技术。增强现实技术随着智能手机的发展与普及，在游戏娱乐、广告等领域得到越来越广泛的应用。

本书具有以下特点：

(1) 实用性强。本书以真实商业产品研发为例，详细讲解了 MR 智能头显的开发。

(2) 内容全面。本书覆盖了 MR 开发从基础知识到实战样例的全部内容，包含了虚拟现实与增强现实的开发实例。

(3) 实验可靠。本书所有代码、实战，都是商用产品开发过程当中成果，全部经过验证。

(4) 售后答疑。所有读者可在疯壳官网社区(https://www.fengke.club/GeekMart/su_fsqqePOZ.jsp)提问，笔者与众多开发者将一起维护好社区。

本书适用范围：

(1) 从事 VR/AR/MR 行业的研发工程师。

(2) 从事 VR/AR/MR 培训的机构与单位。

(3) 高校计算机科学相关专业教师与学生(本书可作为高校实验课程教材)。

本书由张基隆、陈至钊、崔为初、艾韬共同编写。感谢黄仁芳、顾天宇在本书编写过程中提供的帮助与支持。

编 者
2018 年 9 月

目 录

第 1 章 开发准备.....	1
1.1 MR 简介.....	1
1.2 硬件开发环境的搭建.....	1
1.2.1 硬件的准备和连接.....	5
1.2.2 建立 FPGA 工程.....	6
1.2.3 配置 FPGA.....	10
1.2.4 BIT 文件转换成 MCS 文件.....	14
1.2.5 烧录 MCS 文件到 Flash.....	16
1.3 软件开发环境的准备.....	18
1.3.1 下载并安装 OpenCV.....	18
1.3.2 添加 OpenCV 路径到系统环境变量.....	20
1.3.3 Visual Studio 中 OpenCV 工程的创建.....	23
第 2 章 开发基础.....	32
2.1 Camera 开发基础.....	32
2.2 USB 3.0 输出.....	34
2.2.1 UVC 简介及使用说明.....	34
2.2.2 FPGA 中 UVC 相关逻辑的说明.....	36
2.3 FPGA 简介.....	39
2.4 软件理论介绍.....	39
2.4.1 数字图像基础.....	39
2.4.2 摄像机模型和坐标系.....	40
2.4.3 计算机图形学基础.....	41
第 3 章 开发实战.....	46
3.1 硬件主板设计.....	46
3.1.1 图像传感器电路设计.....	46
3.1.2 DDR3 电路设计.....	49
3.1.3 USB 3.0 电路设计.....	49
3.2 FPGA 实现数字图像处理.....	53
3.2.1 顶层设计.....	53
3.2.2 上电时序.....	54
3.2.3 时钟管理.....	54
3.2.4 按键处理.....	54
3.2.5 曝光控制.....	56

3.2.6	摄像头信号重产生	57
3.2.7	像素位宽处理	58
3.2.8	Bayer 转 RGB.....	59
3.2.9	RGB 转 YC.....	60
3.2.10	DDR3 视频帧缓存	61
3.2.11	曝光融合.....	64
3.2.12	输出选择	67
3.2.13	直方图均衡	68
3.2.14	IMU 数据整合到视频流.....	69
3.3	双目摄像机校准与视频透视.....	71
3.3.1	相机内参数	71
3.3.2	如何求相机参数	73
3.3.3	参数文件说明	79
3.3.4	双目视频透视源码工程.....	80
3.4	VR 场景开发.....	82
3.4.1	3DOF VR 概念与简介	82
3.4.2	用 VMG-PROV 实现 3DOF VR 应用	83
3.5	利用 VMG 实现 SLAM 定位及环境感知 AR 应用.....	87
3.5.1	SLAM 与 ORB-SLAM2.....	87
3.5.2	利用 VMG-PROV 运行 ORB-SLAM2 进行定位并在 Unity 中制作 AR 应用 ...	88
附录	95
参考文献	98

第1章 开发准备

1.1 MR简介

介导现实，是由“可穿戴技术之父”——多伦多大学教授 Steve Mann 提出的一项智能显示技术。不同于虚拟现实和增强现实，介导现实是数字化现实 + 虚拟数字画面。

在 20 世纪七八十年代，为了增强自身视觉感知，让眼睛在任何情境下都能够“看到”周围环境，Steve Mann 设计出可穿戴智能硬件，这被看做是对 MR 技术的初步探索。

VR(Virtual Reality)是纯虚拟数字画面，而 AR(Augmented Reality)是在现实上叠加数字画面。目前主流近眼 AR 技术存在视场角小、虚拟物体无法遮挡真实物体等诸多问题。

AR 和 VR 技术都是 MR 的子集合，一副 MR 的眼镜可以实现 AR 和 VR 功能且具有视场角大、虚拟物体完美遮挡真实物体等特点。在此基础上，MR 眼镜还可实现修正现实、削弱现实等其他智能眼镜不具备的功能。

根据 Steve Mann 的理论，智能硬件最后都会从 VR、AR 技术逐步向 MR 技术过渡。有研究机构预估到 2020 年，全球头戴虚拟现实设备年销量将达 4000 万台左右，市场规模约 400 亿元人民币，加上内容服务和企业级应用，市场容量将超过千亿元人民币。如今，腾讯、阿里、暴风影音等一线科技企业也加入到 VR、AR 设备的研发中，势必会推动 VR、AR、MR 技术更快地前进。

1.2 硬件开发环境的搭建

开发 MR 智能眼镜，硬件是必不可少的。本书提供了一套开源的 MR 智能眼镜，命名为 VMG-PROV(以下简称 VMG)，其硬件核心采用 FPGA 设计。下面先重点介绍 FPGA 设计部分。

视频透视优于光学透视的一个关键点是对现实画面的修改。在软件上测试出来的算法如果能通过 FPGA 实现或加速，对于减少延时是有帮助的。在本书配套的 FPGA 工程中，包含了一个实时高动态范围图像合成和映射功能，即通过 FPGA 控制图像传感器高速变换曝光，然后通过曝光融合模块选择出由暗到亮每张图中最好的部分进行实时合成，从而生成“高动态范围图像”(High Dynamic Range HDR)，如图 1-1 所示。



图 1-1 基于 FPGA 的实时高动态范围图像

VMG-PROV 中开源的整个系统架构如图 1-2 所示。从图像传感器进来的原始数据将完全在 FPGA 上进行图像信号处理(Image Signal Process, ISP), 之后合成的 HDR 视频流将通过 USB 3.0 传到 PC 中, 通过 FPGA 配合 PC 端的需求, 进一步解决视频透视(Video See Through, VST)的画面延时问题。

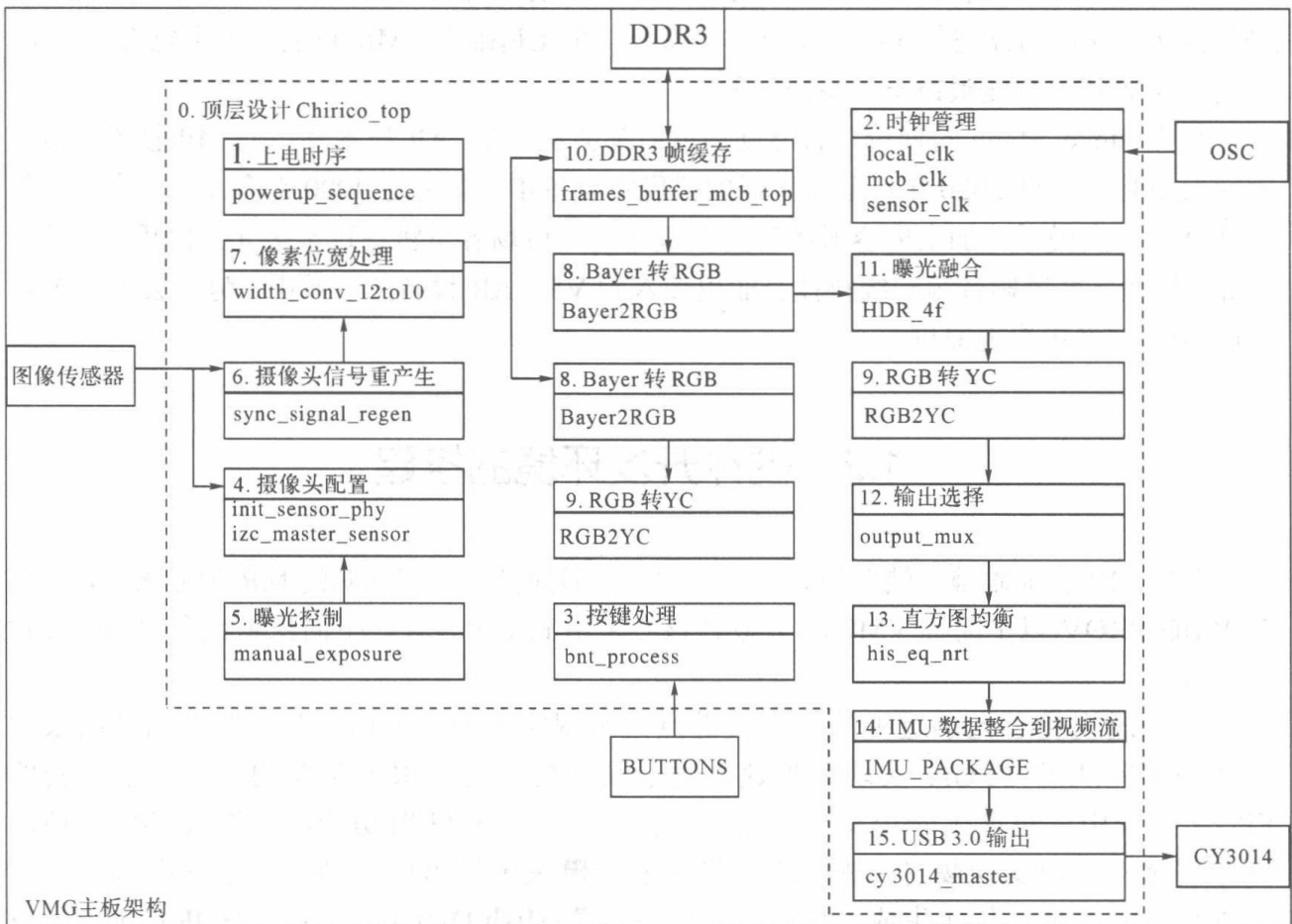


图 1-2 VMG-PROV 主板架构

在这个工程中，所有 FPGA 模块的源代码根据表 1-1 所示文档顺序列出。

表 1-1 FPGA 源代码列表

0	顶层	.../top/RTL/Chirico_top.v
		.../top/UCF/Chirico_top.ucf
1	上电时序	.../powerup_sequence/RTL/powerup_sequence.v
2	时钟管理	.../clock/IP/local_clk/local_clk.xco
		.../clock/IP/mcb_clk/mcb_clk.xco
		.../clock/IP/sensor_clk/sensor_clk.xco
3	按键处理	.../bnt_process/RTL/bnt_process.v
4	摄像头配置	.../config_sensor/RTL/local_clk/init_sencor_phy.v
		.../config_sensor/RTL/mcb_clk/i2c_master_sensor.v
5	曝光控制	.../auto_exprsure/RTL/ae_signal.v
		.../auto_exprsure/RTL/hdr_ae_signal.v
		.../auto_exprsure/RTL/auto_exposure.v
		.../auto_exprsure/IP/ae_weight_x.xco(ae_weight_1080.coe)
		.../auto_exprsure/IP/ae_weight_y.xco(ae_weight_1080.coe)
		.../auto_exprsure/IP/ae_fifo.xco
		.../auto_exprsure/IP/div_ae.xco
		.../auto_exprsure/IP/ev_rom.xco(ar0230_ev.coe)
...	.../his_eq/IP/his_ram_dp.xco	
6	摄像头信号重产生	.../sensor_sync_signal_regen/RTL/sensor_sync_signal_regen.v
7	像素位宽处理	.../width_conv_12to10/IP/width_conv_12to10_rom.xco(width_conv_12to10.coe)
8	Bayer 转 RGB	.../Bayer2RGB/RTL/Bayer2RGB.v
		.../Bayer2RGB/RTL/ROLLINGBUFF_bayerRGB.v
		.../Bayer2RGB/IP/double_port_ram_bayer2rgb.xco
9	RGB 转 YC	.../YC_RGB_conversion/RTL/RGB2YC.v
		.../YC_RGB_conversion/RTL/RGB2YCbCr.v
		.../YC_RGB_conversion/RTL/YCbCr2YC.v
10	DDR3 帧缓存	.../frames_buffer/RTL/frames_buffer_mcb_top.v
		.../frames_buffer/RTL/M_mcb_write.v
		.../frames_buffer/RTL/M_mcb_read.v

续表一

10	帧缓存	.../frames_buffer/RTL/frames_buffer_mcb.v
		.../frames_buffer/IP/mcb_wr_fifo_36x2048/mcb_wr_fifo_36x2048.xco
		.../frames_buffer/IP/mcb_rd_fifo_32x2048/mcb_rd_fifo_32x2048.xco
		.../frames_buffer/IP/mcb_rd_addr_fifo_32x256/mcb_rd_addr_fifo_36x256.xco
		.../frames_buffer/IP/user_design/...
11	曝光融合	.../HDR/RTL/HDR_4f.v
		.../HDR/RTL/div_uu.v
		.../HDR/RTL/div_su.v
		.../HDR/IP/weight_sum_rom.xco(gaussian.coe)
12	输出选择	.../output_mux/RTL/output_mux.v
13	直方图均衡	.../his_eq/RTL/his_eq_nrt.v
		.../his_eq/RTL/arithmetic_unit.v
		.../his_eq/RTL/create_cumulative_sum.v
		.../his_eq/RTL/cumulative_sum.v
		.../his_eq/RTL/float_separate.v
		.../his_eq/RTL/equalize.v
		.../his_eq/RTL/histogram.v
		.../his_eq/RTL/maxuhp1.v
		.../his_eq/RTL/taylor_series_expansion.v
		.../his_eq/IP/equalize_ram.xco
		.../his_eq/IP/float_to_float.xco
		.../his_eq/IP/float_add.xco
		.../his_eq/IP/float_div.xco
		.../his_eq/IP/float_mult.xco
.../his_eq/IP/float_sub.xco		
.../his_eq/IP/float_to_fixed.xco		
.../his_eq/IP/his_ram_dp.xco		

续表二

14	IMU 数据整合 到视频流	.../IMU_package/RTL/IMU_PACKAGE.v
		.../IMU_package/RTL/uart_rx.v
		.../IMU_package/IP/imu_data_fifo.xco
		.../IMU_package/IP/rom_imu_encode.xco
15	USB 3.0 输出	.../USB_cy3014/RTL/cy3014_master.v
		.../USB_cy3014/IP/cy3014_data_fifo.xco

下载 FPGA 源码地址：https://github.com/LokiZhangC/VisionerTech_VMG-PROV_FPGA 或 <https://www.fengke.club/GeekMart/su-fsqoqePOZ.jsp>。

下载好之后，所有 FPGA 源码自动保存在 design 文件夹下，或将下载好的 FPGA 源代码保存在根目录下的 xxx\design 文件夹下。

1.2.1 硬件的准备和连接

在开始调试之前需拆开 VMG-PROV 面板，拆开面板之后的 VMG-PROV 主板如图 1-3 所示(参考 VMG 快速安装手册)。VMG-PROV 主板上有两块 FPGA，所以提供了两个 JTAG 调试口。图 1-3 中圆圈处即为 JTAG 调试口。

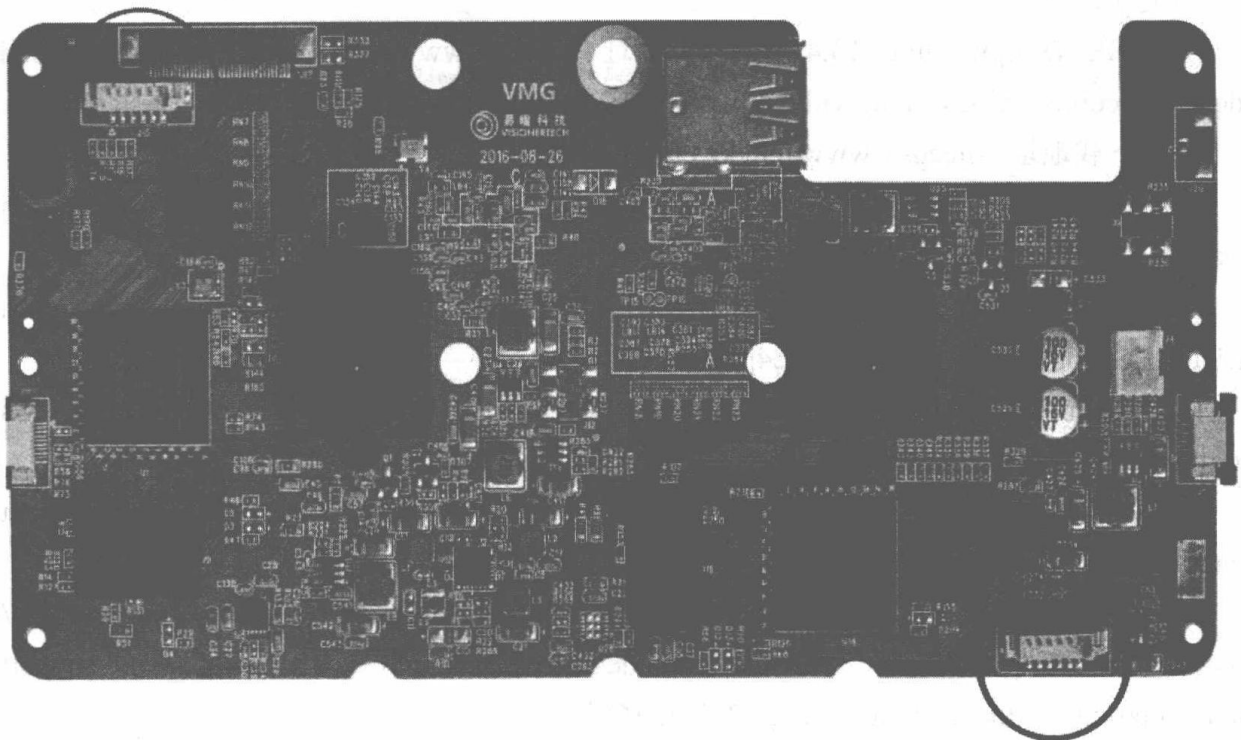


图 1-3 VMG-PROV 主板

按照图 1-4 所示将相关连线接好，其中 FPGA 烧录器通过 USB 线连接 PC，VMG-PROV 自带的 USB 线和 HDMI 线也与 PC 连接。连接好之后 VMG-PROV 会自动开机上电，至此

调试的硬件环境搭建完成。

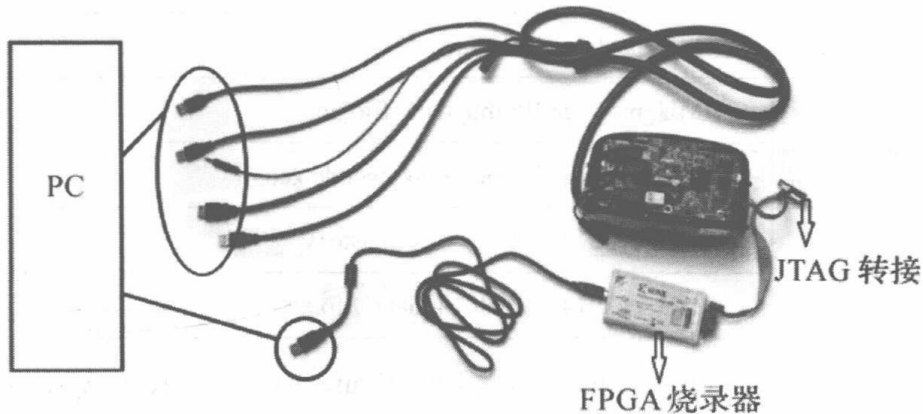


图 1-4 VMG-PROV 硬件连接

1.2.2 建立 FPGA 工程

1. 开发工具及相关文档

VMG-PROV 主板使用的是美国 Xilinx 公司的 Spartan-6 LX45 CSG324，开发软件使用的是 Xilinx ISE Design Suite 13.4 (以下简称 ISE)。在使用之前需要安装相关软件工具并申请或购买相关 License。

相关文档网址如下：

(1) ISE Design Suite 13.4 的下载地址：<https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/archive.html>。

备用下载地址：<https://www.fengke.club/GeekMart/>。

(2) 关于 ISE 的文档可在 Xilinx 的官网中找到，网址如下：<https://www.xilinx.com/support.html#documentation>。

(3) Spartan6 lx45 csg324 芯片管脚封装定义网址如下：<https://www.xilinx.com/support/packagefiles/s6packages/6slx45csg324pkg.txt>。

(4) Xilinx 约束文件指南网址如下：https://china.xilinx.com/support/documentation/sw_manuels/xilinx14_7/cgd.pdf。

(5) Spartan6 时钟资源用户手册网址如下：https://www.xilinx.com/support/documentation/user_guides/ug382.pdf。

(6) 更多关于 Xilinx Spartan6 DDR3 控制器的用法请在下列网址查阅：https://www.xilinx.com/support/documentation/user_guides/ug388.pdf、https://www.xilinx.com/support/documentation/ip_documentation/mig/v3_92/ug416.pdf。

2. 操作步骤

(1) 在“开始”菜单中找到 Xilinx ISE 开发软件并点击打开，如图 1-5 所示。

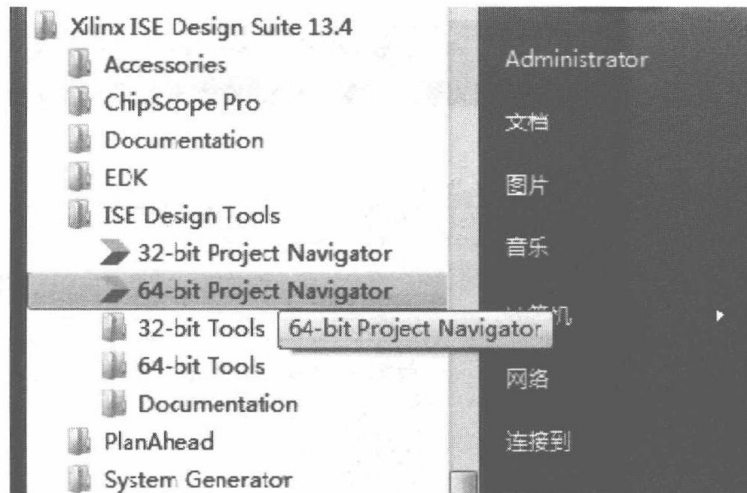


图 1-5 打开 Xilinx ISE 软件

(2) 在 ISE 菜单栏选择“File”→“New Project”，在图 1-6 所示的界面选择工程存放位置和工程名称，工程存放位置和名称可以自定义，选好之后点击“Next”按钮。

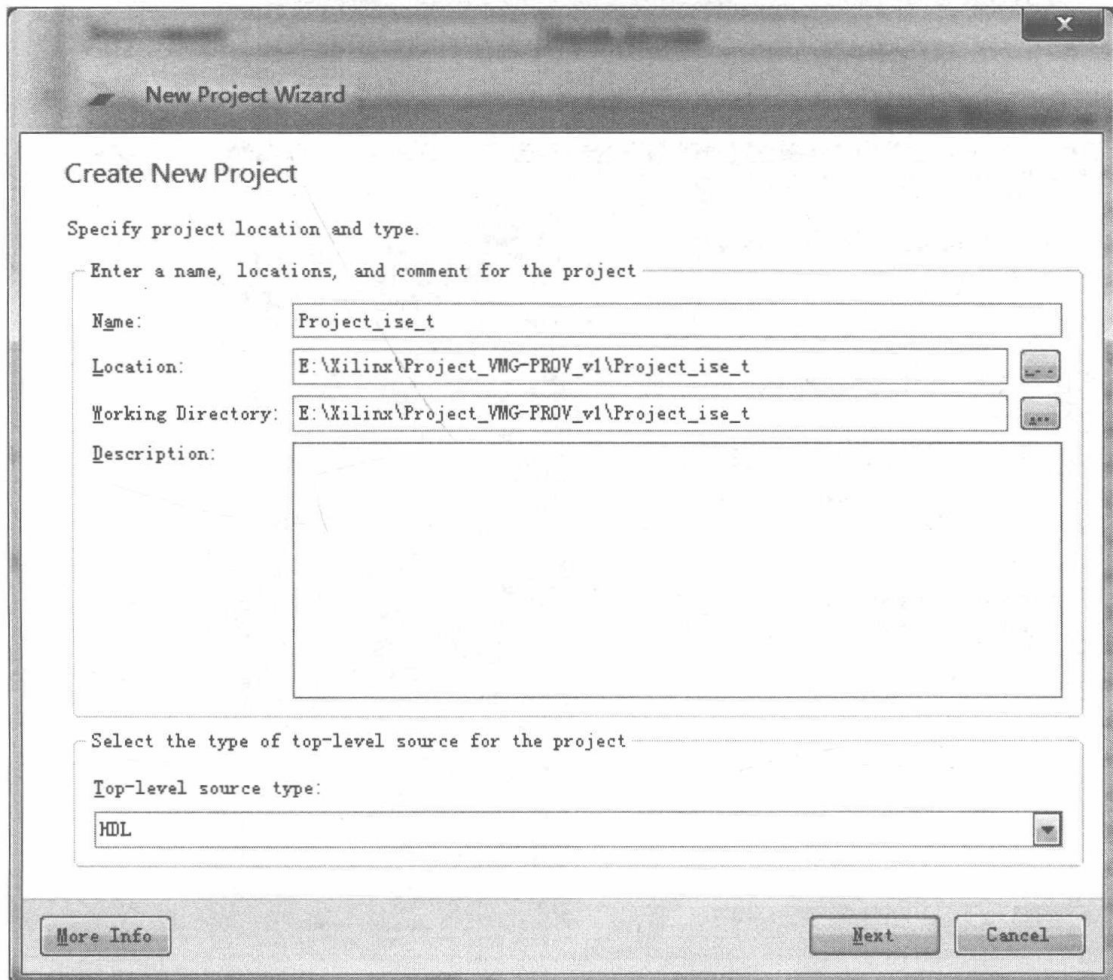


图 1-6 新建 Xilinx ISE 工程

(3) 在图 1-7 所示界面的 Family、Device、Package 和 Speed 栏分别选择 Spartan6、XC6SLX45、CSG324 和-2，这些选项将 FPGA 的芯片型号确定为 Spartan-6 LX45 CSG324。选好之后点击“Next”按钮，再点击“Finish”按钮。

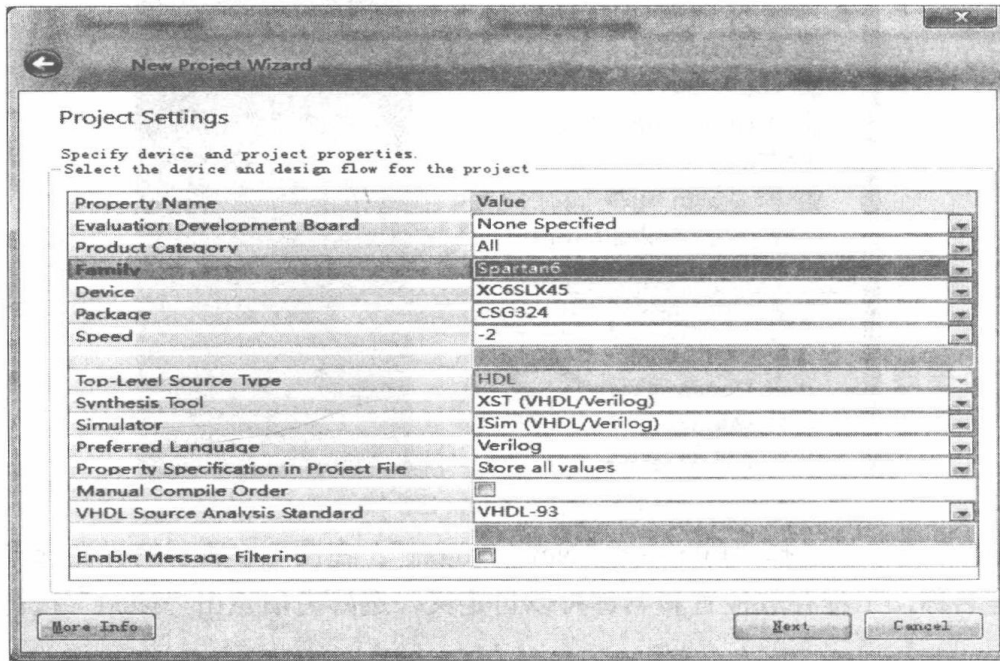


图 1-7 在新建工程中选择 FPGA 型号

至此，一个新的工程就建好了，如图 1-8 所示。

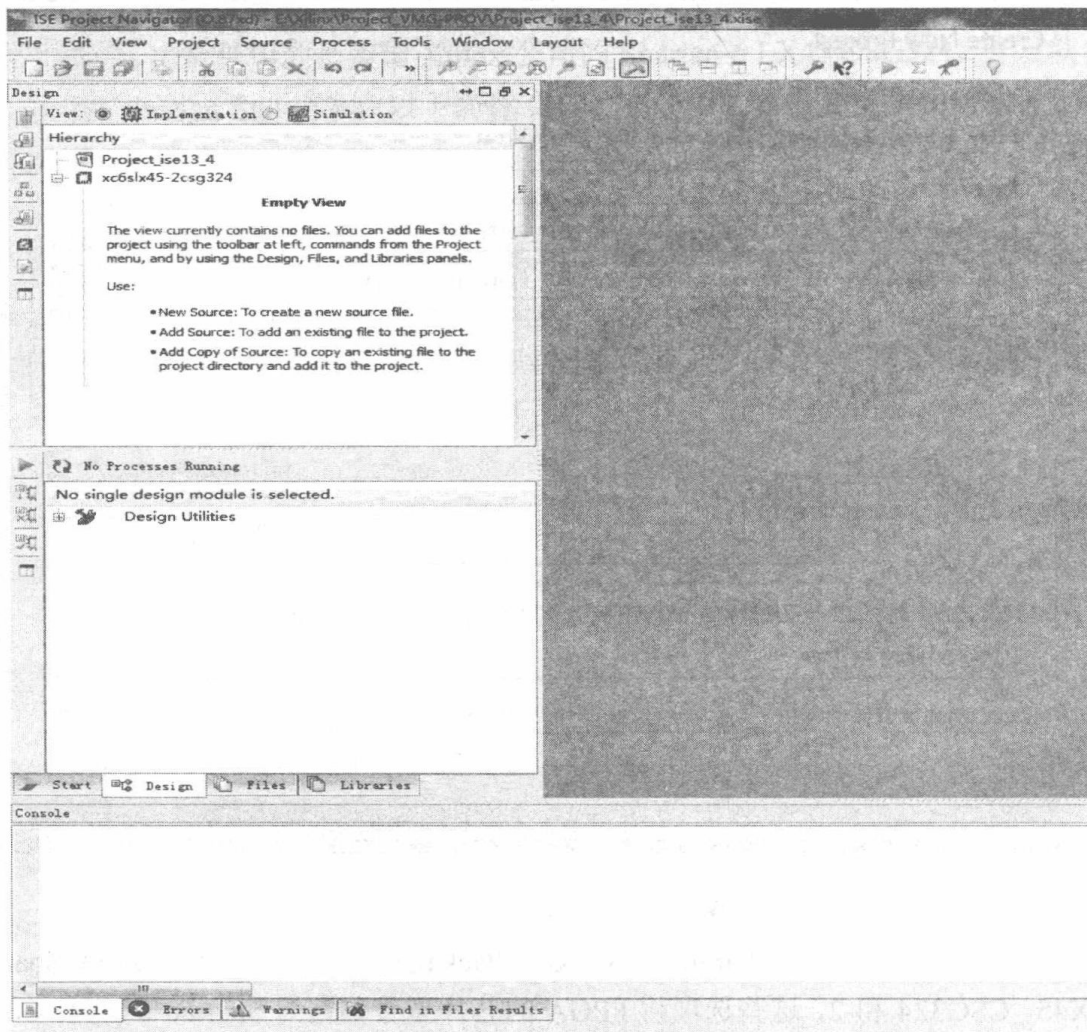


图 1-8 新建的空工程

接下来需要添加所有的源代码(RTL)和 IP。在 Hierarchy 界面点击鼠标右键,选择“Add Source”,如图 1-9 所示,依次添加 design 文件夹下各个模块的源代码(RTL)和 IP 文件(IP 文件的后缀是 .xco)。

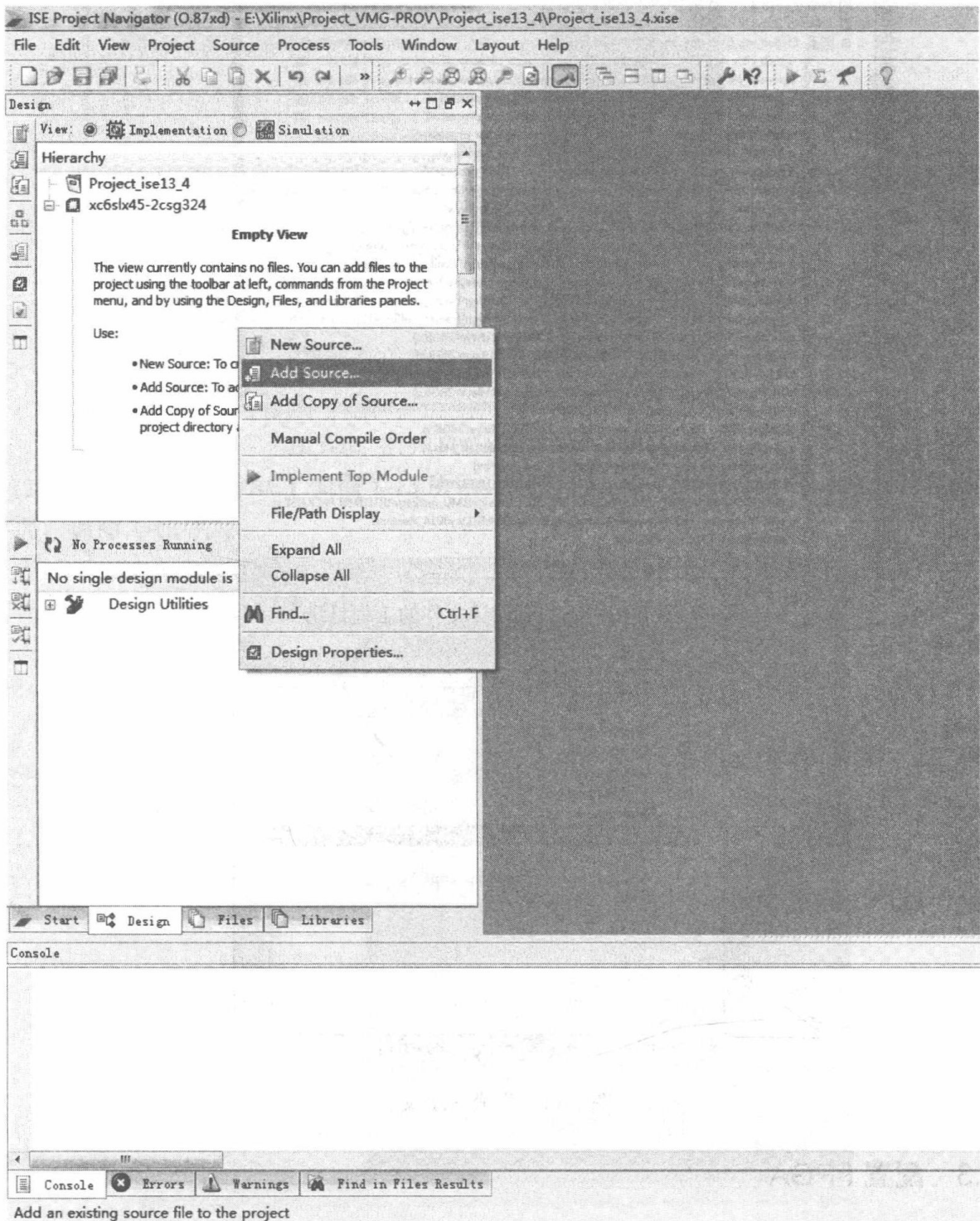


图 1-9 在新建的工程中添加设计文件

添加完源代码(RTL)和 IP 之后的层级结构如图 1-10 所示。

至此,一个完整的 FPGA 工程就搭建好了,双击“Processes”界面下的“Generate Programming File”,如图 1-11 所示,ISE 软件工具会自动进行综合(Synthesize)和布局布线(Implement Design),最终生成 BIT 文件(文件后缀为 .bit,整个过程需要 20 分钟左右)。

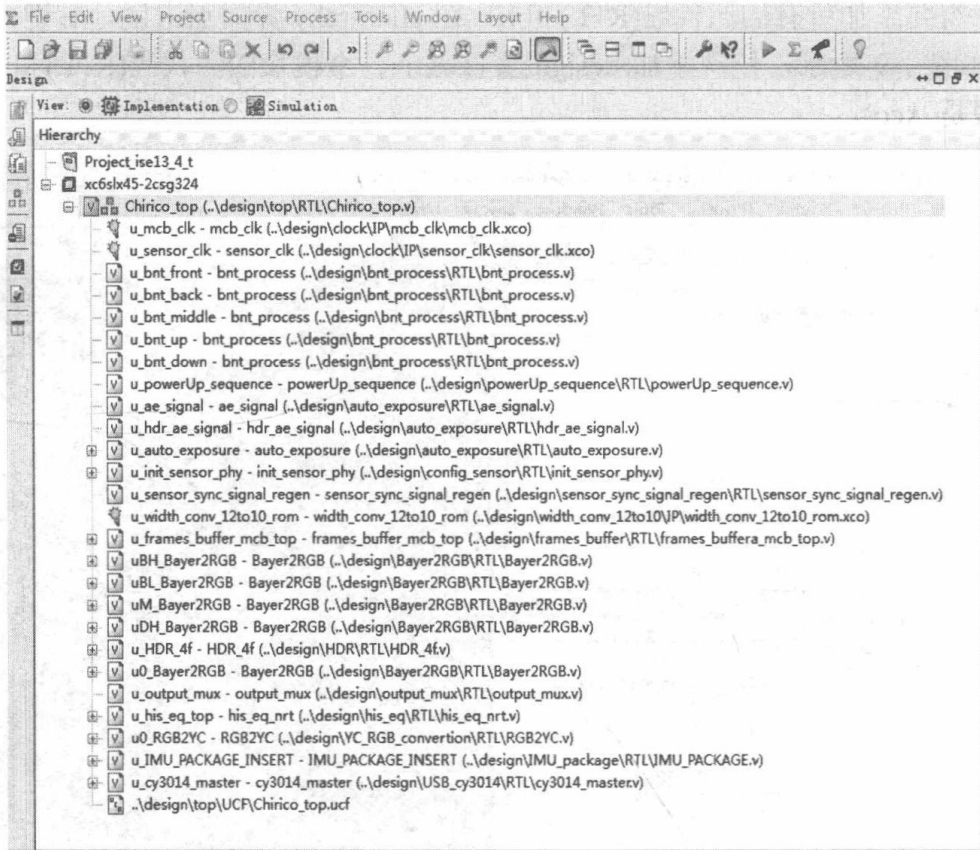


图 1-10 添加完设计文件的工程目录

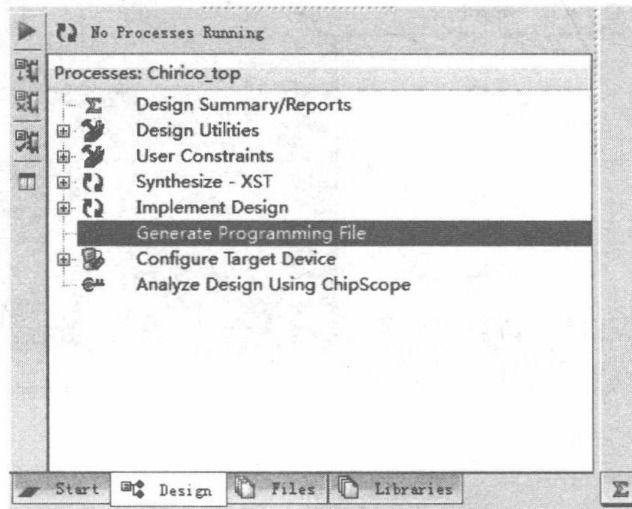


图 1-11 生成 BIT 文件

1.2.3 配置 FPGA

使用 ISE 13.4 开发工具，当 FPGA 代码通过综合、布局布线之后，会生成 BIT 文件，用于直接配置 FPGA。在调试过程中一般只需要将 BIT 文件快速配置到 FPGA，查看功能验证是否正确即可。

配置步骤如下：

- (1) 在“开始”菜单找到 ISE 安装目录，打开 iMPACT，如图 1-12 所示。

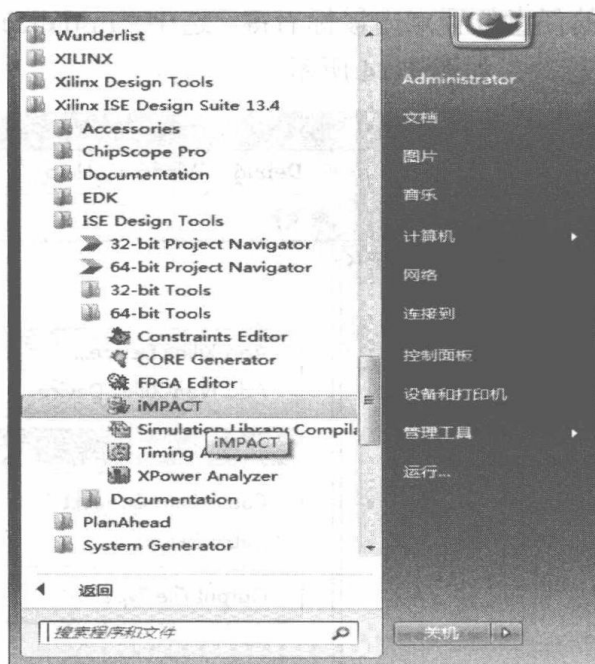


图 1-12 打开 FPGA 配置软件 iMPACT

(2) 第一次打开 iMPACT 如图 1-13 所示，双击左上角“Boundary Scan”，打开边界扫描界面，如图 1-13 所示。

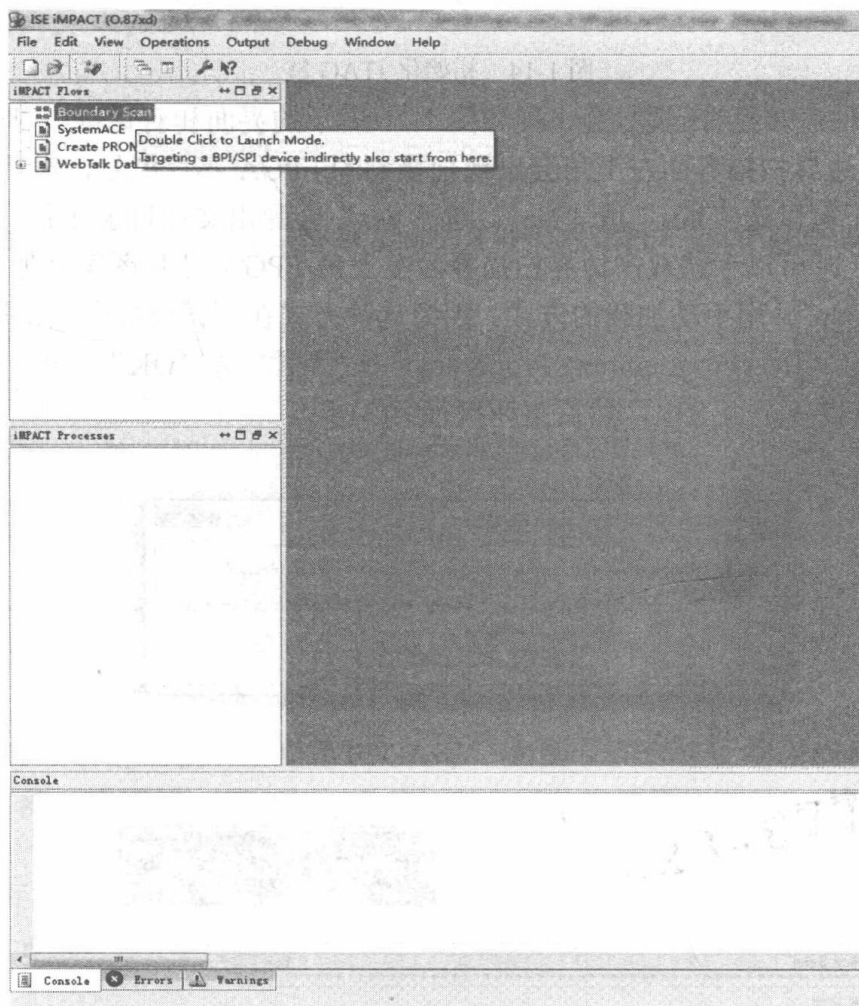


图 1-13 iMPACT 的边界扫描界面