第3章

FPGA开发流程

本章致力于深入解析现代硬件开发领域中极为关键的一部分——现场可编程门阵列 (Field-Programmable Gate Array, FPGA)的开发过程。理解 FPGA 背后的基本概念、特性、内部结构以及它是如何工作的,对于深入掌握硬件设计和优化至关重要。

我们将深入了解 FPGA 的内部结构,包括它如何通过灵活的逻辑块、可编程的输入/输 出以及配置逻辑实现复杂的电子设计。同时,我们会讨论 FPGA 工作的基本原理,解析它 是如何在提供高度灵活性的同时保证性能和效率。本章会重点介绍 FPGA 的开发流程及 其相关工具。我们将详述从设计思想到实际硬件落地的整个开发过程,包括设计、仿真、编 译、配置和测试等关键步骤。我们也将探讨几种主要的 FPGA 开发工具及其特点,为设计 师选择最合适的工具提供参考。

3.1 FPGA 的概念

在探讨现场 FPGA 之前,让我们深入了解 FPGA 到底是什么,它的特点,复杂的内部结构,以及它是如何工作的。本部分旨在为读者揭示 FPGA 技术的核心,带领大家一步步探索这一强大的硬件工具如何在电子设计和产品开发中扮演着不可替代的角色。通过这一系列的探讨,读者将能够获得关于 FPGA 基本概念的全面理解,为后续深入学习和应用 FPGA 技术打下坚实的基础。

3.1.1 FPGA 是什么

FPGA即现场可编程门阵列,是一种可编程的数字 逻辑芯片,它是在 PAL、GAL、CPLD 等可编程器件的基 础上进一步发展的产物。FPGA 是作为专用集成电路 (ASIC)领域中的一种半定制电路而出现的,既解决了 定制电路的不足,又克服了原有可编程器件门电路数量 有限的缺点。FPGA 芯片示意图如图 3.1 所示。

我们可以通过对 FPGA 编程实现大部分的数字功 能。可以说在数字世界里它无所不能,就像乐高的积木 一样可以搭建各种不同的功能模块,实现你所希望的各



图 3.1 FPGA 芯片示意图

种功能。首先,你已经掌握最基本的数字逻辑知识,学会了一种用来构建各种功能的工具语 言,其次,你要动脑(考验的是你的逻辑思维是否清晰),一个优秀的建筑师的作品是在脑子 里勾画出来的,而不是拿积木碰运气拼凑出来的。

FPGA 是一种非常灵活的电子器件,它允许你通过编程来定义其硬件功能。想象一下, 有一个电路板,上面有成千上万个小型电子开关(逻辑门),这些开关可以通过编程任意连接 起来,形成不同的电路。

FPGA的核心特点是它们可以在不改变物理硬件的情况下进行重新编程。这意味着你可以设计一个电路,将其通过编程方式加载到 FPGA上,如果需要改变功能或改进设计,只需重新编程 FPGA 即可,而不需要重新制造整个电路板。这使得 FPGA 非常适合用于原型设计、测试新的电子设计或者用于教育和研究。

在电子和计算机工程领域,FPGA 经常被用于模拟复杂的电路系统,例如处理器、信号 处理电路或者用于特定应用的定制电路。由于它们的高度可编程性和灵活性,FPGA 成为 电子工程师和学生实验新想法的理想选择。简单来说,如果你想设计一个电子系统,但不确 定最终需要什么功能,或者你希望能够快速迭代和测试你的设计,那么 FPGA 就是一个非 常有用的工具。

FPGA 行业的主要厂商包括 AMD(收购 Xilinx)、Intel(收购 Altera)、Microsemi(现为 Microchip Technology 的一部分),以及 Lattice Semiconductor。以下是这些主要厂商的简 要介绍。

- Xilinx: Xilinx 是 FPGA 行业的先驱和领导者之一,总部位于美国加州圣何塞。该公司于 1984 年推出了世界上第一款商用 FPGA,并持续在可编程逻辑设备领域进行创新。Xilinx 的产品广泛应用于通信、工业、科学和军事等领域。2021 年,Xilinx 被 AMD 收购。
- Intel(Altera): Altera 是另一家 FPGA 行业的重要玩家,于 1983 年成立,在硅谷有 很大的影响力。2015 年,Intel 收购了 Altera,从而进入 FPGA 市场。Intel FPGA 产品(以前的 Altera 品牌)被广泛应用于数据中心、通信网络和工业系统中。
- Microsemi(Microchip Technology): Microsemi 是一家提供半导体和系统解决方案的公司,特别是在高性能和高可靠性应用方面。它在 FPGA 领域也有一席之地,尤其 是在航空航天和国防市场。2018年, Microchip Technology 收购了 Microsemi。
- Lattice Semiconductor: Lattice Semiconductor 是一家总部位于美国的公司,专注于 低功耗、小尺寸和低成本的 FPGA 产品。Lattice 的 FPGA 主要用于消费电子、工 业、通信、计算机网络和汽车市场。

国产 FPGA 起步较晚,发展相对国外落后不少,主要的厂商有高云半导体、安路科技、 复旦微电子、紫光同创、京微齐力等,总体技术实力落后于国外。

3.1.2 FPGA 的特点

1. FPGA 和 CPLD 的区别

FPGA(现场可编程门阵列)和 CPLD(复杂可编程逻辑器件)是一回事吗?不是的,它 们都是可编程的数字逻辑芯片,但是有着不同的特性。它们在结构和适用场景上有一些关 键的区别。

- 1) 结构差异
- FPGA:拥有更复杂的可编程逻辑块,通常包括查找表(LUTs)、寄存器和逻辑门。
 FPGA的互连网络更加灵活和密集,这使得它们能够实现更复杂的逻辑功能和更高的处理速度。
- CPLD:相比之下,CPLD结构更为简单。它们通常由一个较小数量的较大的可编 程逻辑块组成,这些逻辑块通过一个相对固定的互连阵列连接。CPLD中的逻辑资 源通常基于宏单元或逻辑单元,这些单元包含固定数量的逻辑门和简单的互连 结构。
- 2) 逻辑容量和速度
- FPGA通常提供更高的逻辑容量和更复杂的逻辑处理能力,适合于处理大型和复杂的设计,如数字信号处理、图像处理和自定义处理器逻辑。
- CPLD 由于其简单的结构,适用于较小和相对固定的逻辑应用,如简单的逻辑控制、 状态机、接口管理等。
- 3) 编程和重配置性
- FPGA 支持较为复杂的重配置选项,可以容纳更大规模和更复杂的设计。它们通常 通过硬件描述语言编程。
- CPLD通常用于较简单的逻辑功能,编程相对简单,但其重配置能力和灵活性不如 FPGA。
- 4) 应用场景
- FPGA通常用于需要高度灵活性和复杂数据处理的场合,例如在需要高性能计算、 视频处理或者复杂的数字系统集成时。
- CPLD更适合小型和固定逻辑应用,如简单的控制系统、I/O扩展和固定逻辑替代, 尤其是在对功耗和成本有严格要求的情况下。

总的来说,FPGA提供了更高的灵活性和处理能力,适合复杂和高性能的应用;而 CPLD则更小、更节能,适合简单的逻辑应用和成本敏感的场合。

2. FPGA 和处理器的区别

FPGA 和微处理器、微控制器是一回事么?不是的。FPGA 相比于 PC 或单片机(无论 是冯•诺依曼结构还是哈佛结构)的顺序操作有很大区别,基于 CPU 架构的微处理器或微 控制器会执行预定义的指令(如加法、跳转、数据移动等)。这些指令是在软件层面上编写并 存储在内存中的,处理器按顺序或根据需要读取并执行这些指令。FPGA 则是通过硬件描 述语言编程的,这种语言描述的是电路的物理行为。FPGA 中的逻辑块和互连资源可以被 配置为直接实现特定逻辑功能的电路用于计算和信号处理。

FPGA的开发相对于传统 PC、单片机的开发有很大不同。虽然 FPGA 的开发需要使 用硬件描述语言编程,但是编写的代码在 FPGA 内部实现后是一个个电路模块,各个模块 可以并行运算。这导致 FPGA 开发入门较难,因为它需要的不只是编程能力,还有电路 设计能力,所以 FPGA 开发需要从顶层设计、模块分层、逻辑实现、软硬件调试等多方面 着手。

3. 为什么用 FPGA?

FPGA具有灵活的开发周期、更低的设计迭代成本、更低的一次性工程费用,易于评

估和实现的可选设计架构,新产品上市时间快。相比于 ASIC 和 MCU, FPGA 具有以下 优势:

- 功能强大,有大量并行处理结构。可以实现数字设计领域几乎所有的功能,如组合逻辑、时序逻辑、存储、处理器;现今的 FPGA 芯片集成了更多功能,比如 PLL 时钟产生、分配、驱动,支持各种高速接口规范的可编程 I/O 块,硬核化的 SPI/I²C 总线以及 ARM 内核等,增强的 DSP 单元,Altera 公司(现已被 Intel 收购)的 MAX10 甚至集成了串行 ADC 能够监测环境的温度。
- 开发快,上市时间短,适合原型设计或小批量产品。FPGA高度灵活,设计实现和后续优化的灵活性可以显著影响项目的进度、设计的复杂度,降低项目的风险,便于更改和升级。
- 重复编程/配置,灵活、快速。集成度高,可以通过选用不同规模的器件实现自己所需要的功能,内部功能模块之间的通信和接口的速度、性能都会较多个分立的芯片之间互连有明显的改善,节省板卡空间,便于调试。

3.1.3 FPGA 的内部结构

FPGA 的关键组件,包括可编程逻辑块(Logic Blocks)、可编程互连资源 (Programmable Interconnect Resources)、输入/输出块(I/O块)、存储元件、时钟管理、配置 逻辑、其他特殊功能块。FPGA 内部结构如图 3.2 所示。



图 3.2 FPGA 内部结构

- 可编程逻辑块: FPGA 的核心部分是逻辑单元,通常称为逻辑单元或逻辑块。这些 是小型、可编程的数字逻辑单元,可以执行简单的逻辑操作,如与、或、非等。逻辑单 元可以被编程组合来实现更复杂的功能。
- 可编程互连资源:为了使逻辑单元之间能够相互通信和连接,FPGA包含了大量的可编程互连资源。这些资源包括一系列的编程线路和开关,可以根据需要配置来创建不同的数据路径。
- 输入/输出块:这些块连接到 FPGA 的外围引脚,允许 FPGA 与外部世界通信。I/O
 块可以被配置为支持各种不同类型的外部接口标准。
- 存储元件:大多数 FPGA 包含一些形式的内部存储元件,如触发器、存储器块(如 RAM 块)和寄存器。这些元件可用于存储数据和中间计算结果。
- 时钟管理: FPGA 通常包含专用的时钟管理模块,如相位锁定环(PLLs),用于生成

和管理时钟信号。这些时钟信号对于同步 FPGA 内部操作至关重要。

- 配置逻辑: FPGA 在上电时通过一个称为配置过程的过程被编程。配置数据通常 存储在外部存储器中,比如闪存,当 FPGA 上电时,这些数据被用来设置逻辑块、互 联资源和 I/O 块的功能。
- 其他特殊功能块:一些 FPGA 还包含专用的硬件模块,如数字信号处理(DSP)块、 高速串行接口等,这些可以用于特定类型的计算任务。

3.1.4 FPGA 是如何工作的

FPGA 是基于查找表结构,采用了逻辑单元阵列的模式来实现组合逻辑,每个查找表 连接到一个 D 触发器的输入端,触发器再来驱动其他逻辑电路或驱动 I/O,由此构成了既 可实现组合逻辑功能,又可实现时序逻辑功能的基本逻辑单元模块,这些模块间利用金 属连线互相连接或连接到 I/O 块。FPGA 的逻辑是通过向内部静态存储单元加载编程数 据来实现的,存储在存储器单元中的值决定了逻辑单元的逻辑功能以及各模块之间或模块 与 I/O 块间的连接方式,并最终决定了 FPGA 所能实现的功能,FPGA 允许无限次的编程。

以上內容显得有些晦涩难懂,我们可以这样理解,FPGA 就像一块空白的数字电路板, 你可以根据需要来"绘制"上面的电路。当你开机时,它会从一个叫做配置文件的东西中加 载设计。这个文件告诉 FPGA 如何连接它内部的小组件,这些组件包括可以执行简单逻辑 操作(比如加法或乘法)的逻辑单元,以及连接这些单元的线路。你可以把它想象成乐高积 木,你可以用这些积木按照图纸搭建出各种各样的结构。

FPGA的亮点在于它的灵活性。就像你可以重复使用同一堆乐高积木来构建不同的模型一样,FPGA可以被重新编程来执行各种不同的任务,从简单的数据处理到复杂的控制系统。

3.2 FPGA 的开发流程与工具

在 FPGA 的开发过程中,熟悉和掌握正确的开发流程与工具是至关重要的。本节将针 对 FPGA 的开发流程与工具进行深入讨论,旨在为那些寻求专业知识和技能提升的硬件工 程师和开发人员提供宝贵的资源与指南。通过本部分的学习,读者将能够获得关于 FPGA 开发流程和工具的深入理解,学习到如何选择和使用合适的工具,以及如何高效地管理整个 开发过程。

3.2.1 FPGA 的开发流程

FPGA 的开发流程通常可以分为几个主要步骤。

- 需求分析与设计规划:明确 FPGA 设计的目标和需求,进行初步的设计规划。
- 硬件描述语言编写:使用硬件描述语言(如 VHDL 或 Verilog)编写设计代码,描述 硬件的逻辑和功能。
- 代码仿真:在编写代码的过程中进行仿真测试,确保代码按预期工作。
- 综合:将硬件描述语言代码转换成电路网表,这一步骤通常由 FPGA 设计软件自动 完成。

- 布局布线(Place & Route):确定电路中各 个元素的物理位置,并进行布线,确保它们 能够正确连接。
- 生成比特流:将布局与布线的结果转换成
 FPGA芯片可以理解的比特流文件。
- 下载配置:将比特流下载到 FPGA 芯片上, 对芯片进行配置。
- 硬件测试与验证: 在实际硬件上测试 FPGA 设计,确保它在真实环境中按预期工作。
- 迭代优化:根据测试结果对设计进行必要的 修改和优化。

FPGA开发流程图如图 3.3 所示。

3.2.2 FPGA 开发工具

1. EDA 工具

在 FPGA 设计的各个阶段, FPGA 厂商和 EDA 软件公司提供了很多优秀的 EDA 工具, 尤其是 FPGA 厂商提供的集成开发环境(IDE)。熟练掌握这些设计工具能够有效提高设计的效率, 但是必须明白这些 EDA 软件只是一个工具, 核心的 FPGA 设计流程是不变的。

目前主流的 FPGA 厂商 Xilinx(现被 AMD 收购)、Altera、Intel、Lattice,都有独立的开发平台,每个产品系列的工具会有不同,安装时需要选择对应的软件工具。

1) Lattice 公司

支持 Windows 和 Linux 平台的 Lattice Diamond、Lattice Radiant。Lattice Diamond 软件界面如图 3.4 所示。

Projekt: Release Notes FP6A Design Guide ③ Import ingLPVER Project Bergin Function Design Function ③ Import ingLPVER Project Lattice Diamond 3.12 Release Notes HDL. Coding Guidelines ③ Import ingLPVER Project Lattice Diamond 3.12 Release Notes HDL. Coding Guidelines ③ MAN_GATE ③ Import ingLPVER Project Other Tool Guides ③ STEP_V21_Code Lattice Olamond 3.12 Release Notes Other Tool Guides ③ STEP_V21_Code Lattice Olamond 3.12 Please Function ModelSim Users Manual Lattice Synthesis Classer Signer Guide ModelSim Users Manual Lattice Olamond 3.12 Please Notes Timing Classer ModelSim Olaser Manual Lattice Olamond 3.12 Please Motes Total Reference Manual Lattice Guide ModelSim User Manual Lattice Diamond 3.12 Please Indiation Guide Syngilly Pro Language Support Reference Manual Lattice Diamond 3.12 Installation Guide for Windows Syngilly Pro Language Support Reference Manual Lattice Diamond 3.12 PleA Libraries (HTNL version) Total Xettice on the Web Lattice on the Web Software Update Center Manuary 3.12 PRGA Libraries (PTOF Version) Answer Database		
Reference Guides Lattice on the Web Diamond 3.12 FPGA Libraries (HTML version) Lattice Semiconductor Diamond 3.12 FPGA Libraries (PDF version) Answer Database Lattice Semiconductor Lattice Semiconductor	fanual	~
Carrently uning Lattice Diamond 3.12 Help (HTML version) Intellectual Property 1.12.2.240.2 Getting update information from Lattice Server <		~
File List Process Hierarchy		

图 3.4 Lattice Diamond 软件界面



图 3.3 FPGA 开发流程图

2) Intel FPGA

支持 Windows 和 Linux 平台的 Quartus Prime Lite Edition。Quartus Prime Lite Edition 软件界面如图 3.5 所示。



图 3.5 Quartus Prime Lite Edition 软件界面

3) Xilinx 公司的 Vivado Design Suite

Vivado Design Suite 软件界面如图 3.6 所示。

4 Vivado 2023.1	X
Bile Flow Iools Window Help Crowk Access	
AMDC	
MLEditon	
	î latera de la companya de
Quick Start	
Create Project 3	
Open Example Project >	
Tasks	
Manage # >	
Upen Naroware Manager 3 Vivado Store 3	
Learning Center	
Decumentation and Tutorials >	
Quick Take Videos > What's New in 2023.1 >	
	7 _ 0 6 ×
	^
	> ``

图 3.6 Vivado Design Suite 软件界面

4) 在线编译工具

如苏州思得普科技有限公司开发的"小脚丫"在线 FPGA 综合设计平台,在浏览器里运行,因此可以支持任何一种操作系统,且能够支持 Lattice 的 XO2 系列 FPGA 和 Altera 的 MAX10 系列 FPGA,适合初学者使用。"小脚丫"在线 FPGA 综合设计平台界面如图 3.7 所示。

	首页 知识库	电子森林	淘宝店铺					Eng	lish 登录 注册
t the		C THREE	(D) FPGAIR®1	⊻#∓#			() EHAR		
□ 设计文件	■ 创建								
回 仿真文件	项目名称								- 1
▲ 查看日志		1 字母、数字、1	「划线 (字母开头) 最	少2个字符,最多10	0个字符				- 1
₽ 运行结果	设备	请选择开发板型	号			٠			- 1
		③ 选择你的开发制	ž						- 1
	项目标签	如:组合逻辑时	序逻辑						- 1
		回车或者点击7	拉选项新建,建议标	签数 3-5个,如:	组合逻辑时序逻	98R			- 1
	描述					^			I
						*			I
	权限	◎ 公开	○ 保密						
			提交						•

图 3.7 "小脚丫"在线 FPGA 综合设计平台界面

除原厂的集成开发环境之外,许多第三方的专业工具也可以用来做 FPGA 开发,最常用的是数字电路的综合工具和仿真工具。比如综合工具 Synplify 和仿真工具 Modelsim。

5) 综合工具 Synplify Premier

Synplify Premier 软件界面如图 3.8 所示。

	Synplify Premier with D	esian Planner					
Run	Ready					Demot	& Examples Search 5
Open Project	Project Fires Design Hierarchy		Project Status Implementation Directory Proce	is View			
Dose Project	Moand Mer : now_1 (xtht_top) - Xinu Zynij (XC72020 CL	5494 -1	1	Project	Gettings		
Id File_	B D Viol	P Insternetation Options - MiconFilter - my 1	Date from	and here	er Ingienen	tation Name	are 1
large File	S chi top vhd [work]			And the second se	Pipelining		1
d implementation	a cot top sloon bitmap anay V vitid [e	Device Options Constraints Implements	ation Results Treing Report High Reliability VHC	b Indexervators	Resource	Sharing	0
elementation Options	S chf top counting bloom add vhd jwei S chf hip counting bloom overy vhat iw	Technology.	Patt Package Speed	10%_1	Disable II	O Insettion	1
PAR Instementation	a cot top counting bloom remove and	Altere Zyra	• XC72020 • (CLG484 •) (-1		Click Co	14CLON	1
erer Long	2 B par 1 - Xinx P&R	Device Mapping Options			Series .		and the second second
		[anne	Trees		Real Time	Memory	Date/Time
Hency (MPL)	1 1	Control Culton	Value		6m:02x		26/54/2916
· · · · · · · · · · · · · · · · · · ·	1	Disatile LO Insetton	2		1.44	0.000	26/04/2016
Synthesis II	8 1	Disable Sequential Optimizations	0		and the	10000	10.02.25
atic Compile Point		Update Compile Point Tening Data			0m:03s	14:5MB	26/04/2016
ue on Error		Enable Advanced LUT Contening	¥			_	10,24,20
Compiler 9		Anisotated Properties for Analyst	<u>19</u>				26/04/2016
Explore II		Verification Mode	<u>u</u>			4110.00	26/54/22/56
ang St		Automatic Real/Write Charle Insertion for RAM			0m.00a	133MD	162428
ng 😥	1 1	Implicit Initial Value Support for Implantialiad Pr	Init/ves				
outed Compilation					annary .		
uted Synthesis					VO Register bits (total_	ia tad)	0
		C-	afauración de la coloción		(DSP48s (dep_used)		0 (229)
		00	oniguración de la solución		Harmotical Ains (and)		
					Concercion of State of State		
		Click on an option to becapion		TT	annan an a		
					Est Prog (est 229.4 MHz	Ted	Stack (stack)
				SYNOPSYS"	Timing Report	t View	1.000
			OK Cancal He	Parties (search			
Contraction P							

图 3.8 Synplify Premier 软件界面

Synplify Premier 是一款由 Synopsys 公司开发的行业领先的 FPGA 综合工具。综合 是将高级硬件描述语言(如 VHDL 或 Verilog)编写的设计转换为门级网表的过程,这个网 表随后用于 FPGA 或 ASIC 的布局与布线。

Synplify FPGA 综合工具以其能对 FPGA 设计产生高性能和低成本而成为业界的标准 工具。Synplify 软件支持最新的 VHDL 和 Verilog 语言结构,包括 System Verilog 和 VHDL-2008。该软件也支持多种不同的 FPGA 架构,如 Altera、Achronix、Lattice、 Microsemi and Xilinx。

Synplify Premier 综合过程包括三方面内容。

(1) 对 HDL 源代码进行编译, Synplify Premier 将输入的 HDL 源代码翻译成 boolean 表达式。

(2) 对编译的结果优化,通过逻辑优化消除冗余逻辑和复用模块,这种优化是针对逻辑 关系的,与具体器件无关。

(3) 对优化的结果进行逻辑映射与结构层次上的优化,最后生成网表; Synplify 将编译 生成的逻辑关系映射成 FPGA 的底层硬件模块和原语(primitive),生成网表并优化。

6) ModelSim

ModelSim 软件界面如图 3.9 所示。



图 3.9 ModelSim 软件界面

ModelSim 是由 Mentor Graphics(现为 Siemens EDA 的一部分)开发的一款行业领先的硬件描述语言(HDL)仿真软件。它被广泛用于验证 FPGA 和 ASIC 设计的正确性。以下是 ModelSim 的一些关键特点。

(1) 多语言支持: ModelSim 支持主流的硬件描述语言,包括 VHDL、Verilog 和 SystemVerilog,使其适用于多种设计流程。

(2) 高效的仿真性能: ModelSim 提供高效的仿真性能,这对于减少复杂设计的验证时间尤为重要。

(3) 用户友好的界面: 它具有直观的用户界面,包括代码编辑器、波形查看器和调试工具,使得设计师可以轻松地编写、测试和分析他们的 HDL 代码。

(4)强大的调试功能: ModelSim 提供了广泛的调试功能,包括断点、单步执行和变量 跟踪,这些功能使得定位和修复设计错误更加高效。

(5) 波形分析:软件能够生成详细的波形输出,帮助设计师可视化信号在时间上的变化,从而更容易理解和调试设计。

(6) 集成开发环境(IDE): ModelSim 通常与其他 EDA 工具集成,如综合工具和时序分 析工具,形成一个完整的设计和验证流程。

(7) 广泛的适用性:从学术研究到工业应用,ModelSim 在电子设计自动化(EDA)领域 内有着广泛的应用。

ModelSim 的这些特点使其成为了 FPGA 和 ASIC 设计师进行 HDL 代码验证和仿真 的首选工具之一。它不仅适用于大型复杂设计的验证,也适合小型或学术项目,使其在电子 设计领域内极为受欢迎。

2. 小脚丫(STEP)FPGA 学习开发平台

FPGA的学习门槛相比单片机要高,不管是硬件成本还是设计难度,对于初学者来说都不够友好。小脚丫(STEP)FPGA开发板是苏州思得普信息科技公司专门针对 FPGA初学者打造的一款性价比高、学习门槛低的学习模块系列。该系列中所有板子的大小兼容标准的 DIP40 封装,尺寸只有52mm×18mm,非常便于携带;并通过 USB 端口进行供电和下载,板上选用的芯片兼具了 FPGA和 CPLD的优点,瞬时上电启动,无需外部重新配置FPGA,是学习数字逻辑绝佳的选择。而且能够直接插在面包板上或以模块的方式放置在其他电路板上,即插即用,大大简化了系统的设计。小脚丫(STEP)FPGA开发板如图 3.10 所示。



图 3.10 小脚丫(STEP)FPGA 开发板

其主要产品有两个系列:基于 Lattice XO2-4000HC FPGA 的 STEP MXO2 开发板和 基于 Intel(Altera) MAX10M02/08 FPGA 的 STEP MAX10 开发板。两个产品系列除核心 的 FPGA 芯片不一样以外,其他板载资源和外观都是相同的。

两款 FPGA 都是小容量的,逻辑资源在 10KB 以下,并且内部集成了 Flash,而一般的 FPGA 都需要用户单独外配一片 Flash 用于存储配置文件,FPGA 在上电时从外部 Flash 加载配置文件并运行。STEP FPGA 系列开发板所选的两款 FPGA 无需用户外配 Flash, 配置文件之间下载到芯片内部,集成度较高。

Lattice 系列开发板小脚丫 STEP-MXO2 硬件资源介绍如图 3.11 所示。 硬件资源如下。

- (1) 核心器件: Lattice LCMXO2-4000HC-4MG132。
- 132 脚 BGA 封装,引脚间距 0.5mm,芯片尺寸 8mm×8mm;
- 上电瞬时启动,启动时间<1ms;
- 4320个 LUT 资源,96Kb 用户闪存,92Kb RAM;
- 2+2 路 PLL+DLL;
- 嵌入式功能块(硬核):一路 SPI、一路定时器、两路 I²C;





图 3.11 小脚丫 STEP-MXO2 开发板硬件资源介绍

- 支持 DDR/DDR2/LPDDR 存储器;
- 104 个可热插拔 I/O 接口;
- 内核电压 2.5~3.3V。

(2) 板载资源。

- 两位7段数码管;
- 两个 RGB 三色 LED;
- 8 路用户 LED;
- 4 路拨码开关;
- 4 路按键;
- 外部时钟频率为12MHz;
- 36 个用户可扩展 I/O 接口(其中包括一路 SPI 硬核接口和一路 I²C 硬核接口);
- 支持的开发工具 Lattice Diamond;
- 一路 Micro USB 接口;
- 板上集成 FPGA 编程器;
- 支持 MICO32/8 软核处理器;
- 板卡尺寸 52mm×18mm。

Intel(Altera)系列开发板如图 3.12 所示。



. . .

硬件资源如下。

(1) 核心器件: Intel(Altera) 10M08SCM153。

- 153 脚 BGA 封装,引脚间距 0.5mm,芯片尺寸 8mm×8mm;
- 上电瞬时启动;
- 8000个LE资源,最大172KB用户闪存,378KbRAM;
- 2路 PLL;
- 24 路硬件乘法器;
- 支持 DDR2/DDR3L/DDR3/LPDDR2 存储器;
- 112 个用户 GPIO 接口;
- 3.3V 电压供电。

(2) 板载资源。

- 两个 RGB 三色 LED;
- 2 路用户 LED;
- 4 路拨码开关;
- 2 路按键;
- 36个用户可扩展 I/O 接口;
- 支持的开发工具 Intel Quartus Prime;
- 一路 Micro USB 接口;
- 一个 10 引脚的 JTAG 编程接口;
- 板卡尺寸 52mm×18mm。

3.3 FPGA开发流程示例

3.3.1 Lattice Diamond 开发 FPGA 实例(以 STEP MXO2 开发 板为例)

下面我们开始可编程逻辑的开发,以控制 LED 交替闪烁为例,完成自己的第一个程序。

1. Lattice Diamond 软件的安装

Lattice Diamond 软件的安装注册激活过程我们不再赘述,读者可以在 Lattice 官网注册登录之后获取 Diamond 的安装包,通过邮箱和计算机的网络地址免费申请许可证,打开软件激活以后即可使用。下面我们重点讲解使用 Diamond 开发 FPGA 的流程。

2. 新建工程

1)新建 Project

双击运行 Diamond 软件,首先新建工程:选择 File→New→Project→Next 选项。 Diamond 软件新建 FPGA 工程页面如图 3.13 所示。

٠	Lattice Diamond - Start Page			- 0	×
File	Edit View Project Design	Process Too	ls Window Help		
	New	•	P File Ctrl+N		
	Open	•	📴 Project Ctrl+Shift+N 📔 🗷 🙆		
	Add	•	Implementation		e ×
	Close	Ctrl+F4	Strategy		
	Close All Close Project		Project: A Release Notes FPGA Lattice Diamond Software Design	D esign Guide Planning	^
8	Save	Ctrl+S	間 Xew 3.12 Release Notes HDL Co	ding Guidelines	
	Save As		User Guides Timing (Closure	
ģ	Save All	Ctrl+Shift+S	Lattice Diamond 3.12 Other	Fool Guides	
	Save Project		Currently running Lattice Diamond software Lattice Diamond 3.12 User	im Users Manual	
	Save Project As		Guide ModelSi	m GUI Reference	
	Archive Project		Lattice Sunthacic Engine for Malitudi		*
	Discard Modified Preferences				
識	Print Preview				<i>6</i> ×
	Print	Ctrl+P			
	Recent Files	×			
	Recent Projects	•	fo Rind Results		
	Exit			Mem Usage: 114,420	к

图 3.13 Diamond 软件新建 FPGA 工程页面

2) 工程命名

我们将新工程命名为 LED_shining,选择工程保存目录。Diamond 软件新建工程命名-保存目录页面如图 3.14 所示。

3) 添加设计文件

下一步可以添加相关设计文件或约束文件(如果已经有设计文件和约束文件,我们可以选择添加进工程):这里我们新建工程,没有相关文件,不需添加,直接单击 Next 按钮。

4) 器件选择

按照 STEP FPGA 开发板器件 LCMXO2-4000HC-4MG132C 配置(器件型号必须确认 正确,否则在引脚设置时会报错)。Diamond 软件新建 FPGA 工程-选择器件型号页面如 图 3.15 所示。

5) 选择综合工具

Synplify Pro(第三方)和 Lattice LSE(原厂)都可以,我们就使用 Lattice LSE。

🚸 New Projec	t	?	×
Project Na Enter a the proj	∎e name for your project and specify a directory ect data files will be stored.	where	
Project: —			
<u>N</u> ame:	LED_shining		
Location:	F:/LED_shining ~	B <u>r</u> owse.	
T	÷		
Implementa	lion:		
Na <u>m</u> e:	impl1		
Loca <u>t</u> ion:	F:/LED_shining/impl1		
	< <u>B</u> ack Next >	Can	icel

图 3.14 Diamond 软件新建工程命名-保存目录页面

Select Device: Family: LatticeXP2 MachXO MachXO3D MachXO3D MachXO3L MachXO3LF Platform Manager Platform Manager 2 Perform Manager 2 Perform Manager 2 Perform Manager 2 Perform Manager 2 Perform Manager 2 Perform Manager 2 Select ASC Device: Part Names Device	×	Device: LCMXO2-2000ZE LCMXO2-4000HC LCMXO2-4000ZE LCMXO2-7000HC LCMXO2-7000HE LCMXO2-7000ZE CSBGA132 External ASC Numbers 0	× •	<pre>Device Informatic Voltage: LUT: Registers: EER Bits: EER Blocks: Dist RAM: DSP: PLL: DLL: PCS: APIO: PIO Cells: PIO Pins: Max Programmable</pre>	2.5V/3.3V 4320 4320 92.2K 10 34560 - 2 0 - 2 2 0 - 2 280 105 IO5:104
---	---	---	--------	---	---

图 3.15 Diamond 软件新建 FPGA 工程-选择器件型号页面

Diamond 软件新建 FPGA 工程-选择综合工具页面如图 3.16 所示。

6) 工程信息确认

上面选择的所有信息都在这里,确认没有问题,直接单击 Finish 按钮。Diamond 软件 新建 FPGA 工程-完成新建页面如图 3.17 所示。

3. 添加设计文件

工程已经建好,我们下面添加设计文件,选择 File→New→File 选项。选择 Verilog

Files(选择自己使用的硬件描述语言),Name 填写 LED_shining,然后单击 New 按钮,这样 我们就创建了一个新的设计文件 LED_shining.v,然后我们就可以在设计文件中进行编程 了。Diamond FPGA 开发-添加设计文件页面如图 3.18 所示。

💀 New Project	?	×
Select Synthesis Tool Specify a synthesis tool for the implementation.		
Synthesis Tools:		
○ <u>S</u> ynplify Pro		
O Precision		
Lattice LSE More information		
< <u>Back</u>	Next >	Cancel

图 3.16 Diamond 软件新建 FPGA 工程-选择综合工具页面

🛷 New Project	?	×
Project Information The new project will be generated with the following specifications.		
Project:		
Project Name: LED_shining Project Location: F:/LED_shining Implementation Name: impl1 Device: LCMXO2-4000HC-4MG132C Synthesis Tool: Lattice LSE		
< Back Einish	Car	ncel

图 3.17 Diamond 软件新建 FPGA 工程-完成新建页面

🚸 New File			?	×					
Categories:		Source Files:							
Source F	iles	Schematic Files		^					
Other Fil	es	Ya Verilog Files							
M VHDL Files									
	EDIF Netlist Files								
	🖹 Preference Files								
	SDC Files								
Verilog Fil	les								
Name:	Name: LED_shining Ext: v								
Location:	Location: F:/LED_shining Browse								
	🗹 Add to	Implementation impl1							
		New] Clo	se					

图 3.18 Diamond FPGA 开发-添加设计文件页面

程序源码如代码 3.1 所示,将代码复制到设计文件 LED_shining.v中,并保存。

```
module LED shining (
   input clk,
                    //clk = 12mhz
   input rst n,
                    //rst n,active low
   output led1,
                    //led1 output
   output led2
                    //led2 output
);
parameter CNT 1S = 12 000 000 - 1;
                                          //time 1S
parameter CNT 05S = CNT 1S >> 1;
                                          //time 0.5S
reg [23:0] cnt;
always @(posedge clk or negedge rst n) begin
    if (!rst_n) cnt <= 1'b0;</pre>
    else if (cnt > = CNT_1S) cnt < = 1'b0;</pre>
    else cnt < = cnt + 1'b1;</pre>
end
wire clk_div = (cnt > CNT_05S)? 1'b1 : 1'b0;
assign led1 = clk div;
assign led2 = \sim clk div;
endmodule
```

代码 3.1 LED_shining 模块 Verilog 代码

4. 综合

程序编写完成,需要综合,在软件左侧 Process 栏,选择 Process,双击 Synthesis Design,对设计进行综合,综合完成后 Synthesis Design 显示对勾(如果显示叉号,说明代码 有问题,根据提示修改代码),Diamond FPGA 开发-综合页面如图 3.19 所示。



图 3.19 Diamond FPGA 开发-综合页面

通过综合工具,我们的代码就被综合成了电路,生成的具体电路,我们可以通过选择 Tools→Netlist Analyzer 查看(仅限 Lattice 的综合工具,第三方综合工具无法查看), Diamond FPGA 开发-查看综合结果页面如图 3.20 所示。





5. 分配引脚和布局布线

综合生成电路后,分配引脚,选择 Tools→Spreadsheet View 选项,按照图 3.21 分配 FPGA 引脚,然后设置 IOTYPE 为 LVCMOS33,保存。Diamond FPGA 开发-分配引脚页 面如图 3.21 所示。

) Start	Page 🛛 🛄 Rep	ports 🗵	🖉 LI	ED_shinin	ig. v 🖾 🚺	🔍 Netli	st Analyzer	🛛 🖉 Spr	eadsheet	Viev \star 🗷	8	
	Name	Group Bv	Pin	BANK	BANK VCC	VREF	IO TYPE	PULLMODE	DRIVE	SLEWRATE	CLAMP	Ī
1	🗸 🐉 All Ports	N/A	N/A	N/A	N/A	N/A	LVCMOS33		N/A	N/A		
1.1	🗸 🍉 Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
1.1.1	✓ I Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1
1.1.1.1	🗈 clk	N/A	C1	5	Auto	N/A	LVCMOS33	DOWN	NA	NA	ON	1
1.1.2	▶ rst_n	N/A	L14	1	Auto	N/A	LVCMOS33	DOWN	NA	NA	ON	
1.2	❤	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1
1.2.1	led1	N/A	N13	1	Auto	N/A	LVCMOS33	DOWN	8	SLOW	OFF	1
1.2.2	led2	N/A	M12	1	Auto	N/A	LVCMOS33	DOWN	8	SLOW	OFF	1
<											2	>
Port A	ssignments Pin	Assignment	s Cl	lock Reso	urce Rout	e Priori	ty Cell Ma	pping Glob	al Prefer	ences Ti	ning Pr ◀	•

图 3.21 Diamond FPGA 开发-分配引脚页面

在软件左侧 Process 栏,选择 Process,直接双击 Place & Route Design 完成布局布线,如 果布局布线报错则查看引脚分配和器件型号是否匹配。Diamond FPGA 开发-布局布线页 面如图 3.22 所示。

6. 输出配置文件

双击 Export Files,所有布局布线输出依次完成。结束后,所有选项显示对勾。Diamond FPGA 开发-生成配置文件页面如图 3.23 所示。

到此,我们完成了第一个程序流文件的生成,下面可以下载到 FPGA 中。

7. 下载配置到 FPGA

STEP MXO2 开发板的编程芯片已经集成到小脚丫开发板上,因此只需要一根 USB 线和计算机相连,就可以完成供电和编程的功能,驱动安装好以后就可以开始编译下载程序了。



图 3.22 Diamond FPGA 开发-布局布线页面

图 3.23 Diamond FPGA 开发-生成配置文件页面

将编译完成的程序下载到开发板,将开发板、下载器和计算机连接。Diamond FPGA 开发板下载连接如图 3.24 所示。



图 3.24 Diamond FPGA 开发板下载连接

选择 Tools→Programmer 选项,选择下载器 HW-USBN-2B(FTDI),然后单击 OK,进 入 Programmer 界面。在 Programmer 界面,单击右侧 Detect Cable,自动检测 Cable 显示 HW-USBN-2B(FTDI),然后单击图 3.25 中的 Program。Diamond FPGA 开发-程序下载 页面如图 3.25 所示。



图 3.25 Diamond FPGA 开发-程序下载页面

显示 PASS,加载完成,观察 STEP MXO2 开发板的 LED 交替闪烁,则表示成功了。

8. 设计仿真

上面我们完成了整个工程的开发过程,例程较为简单,对于复杂的工程开发需要预仿真和后仿真等,保证最终的程序设计逻辑和时序符合我们的设计要求。仿真软件很多,这里我们使用软件自带的 Modelsim 软件进行功能仿真:首先我们添加 testbench 文件,和前面添加设计文件一样,选择 File→New→File→Verilog Files 选项,Name 填写 LED_shining_tb, 然后单击 New 按钮。FPGA 设计仿真-添加仿真文件页面如图 3.26 所示。

🚸 New File			?	×
Categories:		Source Files:		
Source F	iles	🕼 LDC Files		^
Other Fil	es	Power Calculator Files		
		Reveal Project Files		
		Schematic Files		
		Y Verilog Files		
		VHDL Files		
		EDIF Netlist Files		~
Verilog Fil	les			
Name:	LED_shin	ing_tb Ext:	v	•
Location:	F:/LED_s	hining Br	owse	
	🗹 Add to	Implementation impl1		
		New	Clos	9

图 3.26 FPGA 设计仿真-添加仿真文件页面

测试源码如下,复制到 LED_shining_tb. v 文件并保存。为了方便仿真,我们在 LED_ shining_tb. v 调用 LED_shining 模块时将 CNT1S 重新赋值为 19。LED_shining 模块仿真 文件 LED_shining_tb. v 如代码 3.2 所示。

代码 3.2 LED_shining 模块仿真文件 LED_shining_tb. v

```
"timescale 1ns / 100ps
module LED shining tb;
parameter CLK_PERIOD = 10;
reg clk;
initial clk = 1'b0;
always # (CLK PERIOD/2) clk = ~clk;
reg rst_n; //active low
initial begin
    rst n = 1'b0;
     #20;
    rst n = 1'b1;
end
wire led1, led2;
LED shining # (.CNT 1S ( 19 )) u LED shining (
   .clk
                             (clk
                                      ),
   .rst n
                             (rstn),
```

.led1	(led1),
.led2	(led2)
);		
endmodule		

然后在软件左侧 Process 栏,选择 File List,找到 LED_shining_tb.v(必须保存过),右击,选择 Include for→Simulation 选项。FPGA 设计仿真-设置仿真文件属性页面如图 3.27 所示。



图 3.27 FPGA 设计仿真-设置仿真文件属性页面

准备工作完成,我们选择 Tools→SimulationWizard→Next 选项,开始建立仿真工程。 Lattice Diamond 3.12版本软件自带 ModelSim 仿真工具,直接调用 ModelSim(默认),工程 名称填写 LED_shining_tb,工程路径默认即可,然后单击 Next 按钮。FPGA 设计仿真-选 择仿真工具页面如图 3.28 所示。

Simulation Wizard		?	×
Simulator Project Enter a name f will be stored	t Mame or your simulator project and specify a directory where the project	file	
Project name:	LED_shining_tb		
Project location:	F:/LED_shining		
Simulator			
A . 1 10'			
W Model Sim			

图 3.28 FPGA 设计仿真-选择仿真工具页面

选择 RTL 级仿真。下一步勾选 Copy Source to Simulation Directory,然后单击 Next 按钮,调用 ModelSim 软件。FPGA 设计仿真-选择 RTL 级仿真页面如图 3.29 所示。

启动 ModelSim 软件,可以直接查看 testbench 文件中变量的时序变化,想要看 LED_shining 模块中的变量的时序,可以通过图 3.30 中的步骤添加信号至 WAVE 窗口。FPGA 设计仿真-在 ModelSim 中添加观察信号页面如图 3.30 所示。

Simulation Wizard		?	×
Process Stage Indicate what process stage of the FPGA implements Available stages are automatically displayed.	ation strategy you wish to simul	ate.	
Process Stage			
• RTL			
○ Post-Map Gate-Level			
O Post-Route Gate-Level+Timing			
	< Back Next >	Car	ncel



ModelSim Lattice FPGA Edition 2020.3 File Edit View Compile Simulate Add Objects	Tools Layout Bookmarks Wir	ndow Help			- 🗆 X
B·#₽\$\$ XB&22 0·A\$	M 8.8.98.4	\$ III II I	to t . 1	Layout Simulate	
ColumnLayout Default		o 🖬 🛲 🎥 🛛 💁	1 4 =] 100 ps		1110 QQQBBB
Sim - Default tristance Design unit	V Name	Value	Kind 관 I Now 한 🕨	Processes (Active)	Type (filtered)
LED_shining_tb LED_shining LED_shining LED_shining LED_shining LED_shining LED_shining	 CNI_IS CNT_05S Clk 选择后右键 	19 9 5t0	Par Inter Par Inter <u>Net</u> In		
≝ #vsim_capacity#		Ver Declaration Ver Memory Contents Add Wave Yee Add Wave To Add Data Yee Add Data Add to LPF Copy Print Inser threatpoint Toggle Coverage Modify	CHHW Out Out Out Inter CHHO CHHO CHHO CHHO		
ALUbrary X Project X Sim X	<u>+ </u>	Radix	,	· ·	<u>اد</u> به هايا:
<pre># Loading work.LED_shining_tb # Loading work.LED_shining # .main_pane.wave.interior.cs.body.pw.wf</pre>					
VSIM 2>	Project : LED_shining_tb Now: 1 us D	elta: 2	sim:/LED_shining_tb/u_LED_shining	v	.

图 3.30 FPGA 设计仿真-在 ModelSim 中添加观察信号页面

在 WAVE 窗口仿真相应的时间长度,观察信号的时序。FPGA 设计仿真-在 ModelSim 中观察信号时序页面如图 3.31 所示。

Wave	가 있는 것을 하는 것을 하는 것을 가지 않는 것을 하는 것을 했다. 전	□ ×
File Edit View Add F	ormat Tools Bookmarks Window Help	
Wave - Default		+ # ×
B · 🖨 🖬 🖏 🚭	₭₦₶⊇⊇!◎·₩₽│∜₡₶₶₡₶₫」₽ ★ ← → ₽ 500 m + 1 B B B B B B # ● 9 Q Q Q Q A A & & &	
LEFFFFF		
\$1.	Mags	
4 clk St		-
🔹 rst_n 🛛 St:		
■-4 cnt 10	18 19 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 6 7 8 9 10 11 12	1 2
* clk_div St:		
* led1 St		
* led2 St		
Now	500 ns 250 ns 300 ns 350 ns 400 ns	
Cursor 1	120 ns	
<u> </u>		
190700 ps to 446200 ps	្រះរ	11.

图 3.31 FPGA 设计仿真-在 ModelSim 中观察信号时序页面

至此,我们完成了使用 Diamond 软件进行 FPGA 开发的完整流程。

3.3.2 Intel Quartus Prime 开发 FPGA 实例(以 STEP MAX10 开发板为例)

1. 软件安装

Quartus Prime 是 Intel(原 Altera)公司的综合性 PLD/FPGA 开发软件,设计能力强大,接口直观易用,具有运行速度快、界面统一、功能集中、易学易用等特点。Quartus Prime 有 Lite、Standard 和 Pro 三个版本,不同版本可适用的器件和功能不同,其中 Lite 可用于 MAX10 系列 FPGA 的开发,而且是免费的。

下载安装包注意:选择软件版本和操作系统,同时勾选 Quartus Prime、ModelSim-Intel FPGA Edition 和 MAX 10 系列器件支持。大家可以去 Intel FPGA 官网下载,具体下载和 安装过程我们不再赘述。

2. 新建工程

1) 创建工程

双击 Quartus 系列软件图标,启动软件,单击 File→New Project Wizard 选项或单击 Home 页面中的 New Project Wizard 图标。Quartus Prime Lite Edition 软件界面如图 3.32 所示。

Eile Edit View Project Assignments Proce	ssing Iools Window Help	Search I	ntel FPGA
	- 1000 BFFR0 BT93 .		
Project Navigator 🔺 Hierarchy 🔹 🔍 🖲 🕸 🕷	in Home ≭	IP Catalog	(¥ 8 @
A Compilation Hierarchy	Pocont Projects	Device Family MAX 10 (DA/DD/DF/DC)	/SA/SC/SL) 🔻
	Recent Projects	۹,	× =
Tasks Compilation # 18 @ 8 ▼ ▶ Compile Design ▶ ▶ Analysis & Synthesis ▶ ■ Fitter (Pitca & Route)	Tutorial Video: KBG Tentacebess Totorial Video: KBG Tentacebess Totorial Video: KBG Tentacebess	Winstalled IP Project Directory No Selection Available Ubray Bask Functions DSP Interface Protocols Memory Interfaces and Con Processors and Peripherals University Program Search for Partner IP	trollers
Timing Analysis	Close page after project load Don't show this screen again intel	+ Add	
M O A A A ♥ <-riter>> Type ID Message	60 Eind 66 Find Negt		Þ

图 3.32 Quartus Prime Lite Edition 软件界面

2) 工程目录、名称填写

(注意:工程目录中不能有汉语、空格等字符)。Quartus Prime 新建工程-设置工程名称和目录页面如图 3.33 所示。

- 工程目录:选择新建工程的目录。
- 工程名称:填写工程名称。
- 顶层模块名称:设计文件中顶层模块名称要跟工程名称相同。
- 工程类型:选择 Empty project。

Eile Edit View Project Assignments Proces	ng Iools Window Help	Search Intel FPGA
D ≈ 8 + 0 € > C	- 1440 B F F K Q B & Q & Q & Q & Q & Q & Q & Q & Q & Q	
Project Navigator 🔺 Hierarchy 👻 🤉 🕫	it Home ≭	IP Catalog 夏回回
A Compilation Hierarchy		Device Family MAX 10 (DA/DD/DF/DC/SA/SC/SL) *
	New Project Wizard Project	(a, X) %
		👻 🕷 Installed IP
	New Project Witzard	 Project Directory
	Cor	No Selection Available
	Directory, Name, Top-Level Entity	Basic Functions
		► DSP
	What is the working directory for this project?	Interface Protocols
	E:/FPGA_Projects	Memory Interfaces and Controllers
	What is the name of this project?	Processors and Peripherals
Tasks Compilation ▼ ■ 🗐 🛛 📾	LED_shining	Search for Partner IP
Task	What is the name of the top-level design entity for this project? This name is case sensitive	U Statento Farbern
▼ ► Compile Design	utorial Vider and must exactly match the entity name in the design file.	
Analysis & Synthesis	utorial Video LED_shining	
Fitter (Place & Route)	utorial Video Use Existing Project Settings	
Assembler (Generate program	I Close page	
· · · · · · · · · · · · · · · · · · ·	Don't show	Intel. + Add.
Type ID Message		
8		
Be +	Hetp <back next=""> Einish Cancel</back>	•

图 3.33 Quartus Prime 新建工程-设置工程名称和目录页面

3) 添加设计文件

如果已有设计文件,在当前页面选择并添加,也可以不添加,工程新建完成后再创建新文件。

4) 器件选择

根据开发平台使用的 FPGA 选择对应器件型号(10M02SCM153I7G/10M08SCM153C8G) 页面如图 3.34 所示。

Device	Board							
Select the You can in	family and stall addit	d device you want ional device supp	to target ort with t	for compilation he Install Devic	n. es com	mand on the Tools me	enu.	
Device fam	nily	rsion of the Quart	lus Phine	Software in wh	ich you	Show in 'Available de	vices' list	
Eamily:	MAX 10 (I	DA/DD/DF/DC/SA	/SC/SL)		•	Pac <u>k</u> age:	MBGA	
Device	MAX 10	SC			•	Pin <u>c</u> ount:	153 🔹	
Target dev	ice					Core sp <u>e</u> ed grade:	7 •	
 <u>Auto</u> <u>Spec</u> <u>O</u>the 	device se i fic device r: n/a	lected by the Fitte	er able devi	ices' list		Name filter:	l devices	
A <u>v</u> ailable c	levices:							
Na 10M04SC	ne M153I7G	Core Voltage 3.3V	LEs 4032	Total I/Os	GPI 112	Os Memory Bits 193536	Embedded multiplier 9-bit elements	
10M08SC ∢	M153I7G	3.3V	8064	112	112	387072	48 1	

图 3.34 Quartus Prime 新建工程-选择器件型号页面

5) EDA 工具选择

选择第三方 EDA 工具,如果有需要可以选择第三方的综合或仿真工具(第三方工具需 要另外安装并设置启动路径),这里我们选择使用 ModelSim 工具仿真。Quartus Prime 新 建工程-选择综合和仿真工具页面如图 3.35 所示。

DA tools:			vere	
Tool Type	Tool Name	Format(s)		Run Tool Automatically
Design Entry/Synt	Synplify	▼ VQM		Run this tool automatically to synthesize the current design
Simulation	ModelSim	 Verilog HDL 	•	Run gate-level simulation automatically after compilation
Board-Level	Timing	<none></none>	*	
	Symbol	<none></none>	•	
	Signal Integrity	<none></none>	•	
	Boundary Scan	<none></none>	*	

图 3.35 Quartus Prime 新建工程-选择综合和仿真工具页面

6) 工程信息确认

确认工程相应的设置,如需调整单击 Back 按钮返回修改,若确认设置,则单击 Finish 按钮。Quartus Prime 新建工程-工程信息确认页面如图 3.36 所示。

S New Project Wizard	×						
Summary							
When you click Finish, the project will be created with the following set	ttings:						
Project directory: E:/FPGA_Projects							
Project name:	LED_shining						
Top-level design entity:	LED_shining						
Number of files added:	0						
Number of user libraries added:	0						
Device assignments:							
Design template:	n/a						
Family name:	MAX 10 (DA/DD/DF/DC/SA/SC/SL)						
Device:	10M08SCM153I7G						
Board:	n/a						
EDA tools:							
Design entry/synthesis:	Synplify (VQM)						
Simulation:	ModelSim (Verilog HDL)						
Timing analysis:	0						
Operating conditions:							
Core voltage:	3.3V						
Junction temperature range:	-40-100 � C						
Help	< Back Next > Einish Cancel						

图 3.36 Quartus Prime 新建工程-工程信息确认页面

3. 添加设计文件

(1)选择 File→New 选项,或单击工具栏中的 New 按钮,选择 Verilog HDL File 文件 类型,单击 OK 按钮,Quartus 软件会新建并打开 Verilog 文件。

(2) 在新建的 Verilog 文件中进行 Verilog HDL 代码编写、保存,文件名为 LED_shining. v,程序源码如代码 3.3 所示。Verilog HDL 代码编写、保存页面如图 3.37 所示。

🕲 Quartus Prime Lite Edition - E:/FPGA_Projects/LED_shining - LED_shining	- 🛛 🗡
Elle Edit View Project Assignments Processing Tools Window Help	earch Intel FPGA
□ □ ◎ 目 → □ □ つ ○ □ LED_shining → ノ ダ ぢ � ◎ ▶ ▶ ¥ & ♀ ◎ & ቅ ඔ ●	
Project Navigator & Hierarchy + QEES New X IP Catalog	9 8 8
EntityInstance	X I
MAX 10: 10M085CM153I7G New Quartus Prime Project	
LED_shining th Project Directory	
AHDL File AHDL File No Selection Availat	le
FDIF File Vibrary	
Qsys System File Basic Functions	
State Machine File DSP	
SystemVerlag HDL File Interface Protocols	
Verified IDE Ella Memory Interfaces a	nd Controllers
VHDC File Processors and Peri	oherals
✓ Memory Files	
Tasks Compilation + #300 Hexadecimal (Intel-Format) File Search for Partner IP	
Task 🔺 🔻 Verification/Debugging Files	
▼ Compile Design In-System Sources and Probes File	
Analysis & Synthesis Logic Analyzer Interface File Signal Tax Logic Analyzer Ele	
Fitter (Place & Route) University Program VWF WF	
Assembler (Generate program Given Queues New Identification	
K Cancel Help Obcommate	
A Contraction of the second seco	
All O A A Y < <filter>> ØBEind ØFind Next</filter>	
Tune TD Merrage	
Pressage Pressage O 25302DeFault device 10M08DAF484C8G is automatically selected for the device family MAX 10	
 Antipological production of the second s	
The second secon	
System(1) Trocessing	

图 3.37 Verilog HDL 代码编写、保存页面

代码 3.3 Intel FPGA 开发案例 LED_shining. v 模块

```
module LED_shining
(
                 //clk_in = 12mhz
input clk_in,
input rst_n_in, //rst_n_in, 低电平有效
               //led1 输出
output led1,
output led2
                 //led2 输出
);
parameter CLK DIV PERIOD = 12 000 000;
reg clk div = 0;
//wire led1,led2;
assign led1 = clk div;
assign led2 = ~clk div;
reg[24:0] cnt = 0;
always@(posedge clk_in or negedge rst_n_in) begin
    if(!rst n in) begin
           cnt < = 0;
           clk div < = 0;</pre>
    end else begin
           if(cnt = = (CLK_DIV_PERIOD - 1)) cnt <= 0;</pre>
           else cnt < = cnt + 1'b1;</pre>
           if(cnt <(CLK_DIV_PERIOD >> 1)) clk_div <= 0;</pre>
```

```
else clk_div <= 1'b1;
end
end</pre>
```

endmodule

4. 分析综合

(1)选择菜单栏中的 Processing→Start→Start Analysis & Synthesis 选项,或单击工具栏 中的 Start Analysis & Synthesis 按钮, Quartus Prime 分析综合操作步骤页面如图 3.38 所示。

Quartus Prime Lite Edition - E:/FPGA_	Projects/LEC	D_shining - LED_shining					- 0 ×
Eile Edit View Project Assignments	Processing	<u>T</u> ools <u>W</u> indow <u>H</u> elp					Search Intel FPGA
	Stop Pr	ocessing	Ctrl+Shift+C				
	▶ Start Co	ompilation	Ctrl+L	[in a st	(Fig. 1)
Project Navigator 📓 Files 🔹 🤊	😨 Analyze	e Current <u>F</u> ile				IP Catalog	488
Files	Start		•	Pa Start Hierarchy Elaboration		٩	× ÷
I LED_shining/LED_shining.v	Update Memory Initialization File		🛸 Start Analysis & Elaboration 🔹 👻 🕷 Installed IP				
			🐇 Start Analysis & Synthesis	Start Analysis & Synthesis Ctrl+K		2.1.1	
			独 Start Partition Merge	點 Start Partition Merge	No Selection A	/ailable	
			Start Eitter	ter			
			🗟 Start Assembler		► DSP		
			Start Timing Analyzer	g Analyzer Ctrl+Shift+T Jetlist Writer n Assistant	Interface Proto	cols	
			Start EDA Netlist Writer		Memory Interfaces and Controllers.		
1		<pre>17 reg[24:0] cnt=0; 18 Falways@(posedge clk in or neged)</pre>	Start Design Assistant		Processors and	Peripherals	
	19 if(!rst_n_in) begin			Ctrl+Shift+P	University Prog	ram	
Tasks Compilation 👻 =	20 21	clk_div<=0		lllllllllllllllllllllllllllllllllllll		Search for Partner I	<i>t</i> .
Task	- 22	end else begin		👫 Start Rapid Recompile			
▼ ► Compile Design	23	else cnt <	= cnt + 1'b1;	🗟 Start Signal Probe Compilation	Ctrl+Shift+I		
▶ ▶ Analysis & Synthesis ◄	25	if(cnt<(CLI	<pre>if(cnt<(CLK_DIV_PERIOD> else clk_div <= 1'b1;</pre>	Start I/O Assignment Analysis			
 Fitter (Place & Route) 	27	end end		😍 Start Check & Save All Netlist Changes			
 Assembler (Generate program 	ram 28	end		Start Equation Writer (Post-synthesis) Start Equation Writer (Post-fitting)			
Timing Analysis	- 30	endmodule				-	
4	F 4			Start Test Bench Template Writer		+ Add	
8 8 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ter>>	66 Eind.	. #Find Next	🕭 Start EDA Synthesis			
Type TD Message 0 253020 Default device 4 System (1) Processing	10M08DAF	484C8G is automatica	lly selected	for the device family MAX 10			► 000000

图 3.38 Quartus Prime 分析综合操作步骤页面

(2) Quartus 软件会完成分析综合,若设计没有问题,综合 Tasks 栏中 Analysis & Synthesis 会变成深色,同时左侧出现对勾。Quartus Prime 综合完成显示综合结果页面如 图 3.39 所示。



图 3.39 Quartus Prime 综合完成显示综合结果页面

(3) 综合完成后可以选择 Tools→Netlist Viewers→RTL Viewer 查看电路。Quartus Prime 查看综合后电路页面如图 3.40 所示。



图 3.40 Quartus Prime 查看综合后电路页面

5. 引脚约束

(1)选择 Assignments→Device 打开器件配置页面,然后单击页面中的 Device and Pin Options…选项打开器件和引脚选项页面。Quartus Prime 配置器件和引脚选项页面如 图 3.41 所示。

Device Device Board						
Select the family ar You can install addi To determine the v	nd device you want itional device supp ersion of the Quart	to targe ort with t tus Prime	t for compilation the Install Device software in wh	n. es con ich yo	nmand on the Tools me ur target device is supp	enu. orted, refer to the <u>Device Support List</u> webpage.
Device family					Show in 'Available de	evices' list
Eamily: MAX 10	(DA/DD/DF/DC/SA	/SC/SL)		•	Pac <u>k</u> age:	MBGA
Device: All				Pin <u>c</u> ount:	153 🔹	
Target device					Core speed grade:	7
<u>Auto device s</u> <u>Specific device</u>	elected by the Fitte re selected in 'Avail	er able dev	ices' list		Name filter: V S <u>h</u> ow advanced	d devices
Other: n/a				>	Device and Pin Optic	ons
A <u>v</u> ailable devices:						
Name	Core Voltage	LEs	Total I/Os	GP	IOs Memory Bits	Embedded multiplier 9-bit elements
10M04SAM153I70	G 3.3V	4032	112	112	193536	40 1
10M04SCM153I70	5 3.3V	4032	112	112	193536	40 1
Migration Devices.	. 0 migration devi	ces sele	cted			
					[₩ Buy Software OK Cancel Help

图 3.41 Quartus Prime 配置器件和引脚选项页面

(2) 在 Unused Pins 选项中配置 Reserve all unused pins 为 As input tri-stated 状态。 Quartus Prime 配置引脚模式页面如图 3.42 所示。

S Device and Pin Options - LED_sh	ining X
Device and Pin Options - LED_sh Category: General Configuration Programming Files Unused Pins Dual-Purpose Pins Capacitive Loading Board Trace Model I/O Timing Voltage Pin Placement Error Detection CRC	Unused Pins Specify device-wide options for reserving all unused pins on the device. To reserve individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, use the Assignment Editor. Reserve all unused pins As input tri-stated with weak pull-up
CvP Settings Partial Reconfiguration	Description: Reserves all unused pins on the target device in one of 5 states: as inputs that are tri-stated, as outputs that drive ground, as outputs that drive an unspecified signal, as input tri-stated with bus-hold, or as input tri- stated with weak pull-up. Reset OK Cancel Help

图 3.42 Quartus Prime 配置引脚模式页面

(3) 在 Voltage 选项中配置 Default I/O standard 为 3.3-V LVTTL 状态。Quartus Prime 配置引脚电平页面如图 3.43 所示。

Device and Pin Options - LED_shi	ining		×
Category:			
General	Voltage		
Configuration Programming Files	Specify voltage options fo	r the device.	
Unused Pins Dual-Purpose Pins	Default I/O standard:	3.3-V LVTTL	•
Capacitive Loading	VCCIO I/O bank <u>1</u> voltage:	n/a in MAX 10 use Pin Planner to adjust VCCIO voltage	*
I/O Timing	VCCIO I/O bank2 voltage:	n/a in MAX 10 use Pin Planner to adjust VCCIO voltage	Ŧ
Vottage Pin Placement Error Detection CRC CvP Settings Partial Reconfiguration	Core voltage: 1.2V		
	Description:		
	Specifies the default I/O s	tandard to be used for pins on the target device.	eset
		OK Cancel	Help

图 3.43 Quartus Prime 配置引脚电平页面

(4)选择菜单栏中的 Assignments→Pin planner 选项,或单击工具栏中的 Pin planner 图标,进入引脚分配界面。在 Pin Planner 页面中将所有端口分配对应的 FPGA 引脚,Quartus Prime 引脚分配页面如图 3.44 所示,然后关闭(自动保存)。



图 3.44 Quartus Prime 引脚分配页面

6. 编译(布局布线&生成配置文件)

选择菜单栏中的 Processing→Start Compilation 选项,或单击工具栏中的 Start Compilation 按钮,开始所有编译,等待 Tasks 列表中所有选项完成,Quartus Prime 编译完成页面如图 3.45 所示。



图 3.45 Quartus Prime 编译完成页面

7. 下载到 FPGA

(1)使用 micro-usb 线将 STEP-MAX10 开发板连接至计算机 USB 接口,选择菜单栏 中的 Tools→Programmer 选项,或单击工具栏中的 Programmer 按钮,进入烧录界面。选择 Hardware Setup 选项,选择硬件驱动为 USB-Blaster[USB-0]。Quartus Prime 编程下载 及驱动设置页面如图 3.46 所示。

Programmer - E:/FPGA_Projects/LED_shining - LED_shining - [LED_shining.cdf]										\times
<u>Eile Edit V</u> i	ew Processing	Tools <u>W</u> indov	w <u>H</u> elp					Search I	ntel FPC	A O
∔Hardware	Setup USB-Bl	aster [USB-0]		Mode: JT	AG	• F	^o rogress:			
Enable re	al-time ISP to allow	w background	👋 Hardware Setup							×
▶ [%] Start	File	Device	Hardware Settings	JTAG Settir	ıgs					
Stop	output_files/L	10M08SC	Select a programmin hardware setup appli	g hardware se es only to the	tup to use when pr current programm	ogramming de er window.	evices. Th	is program	nming	
× Delete			Currently selected ha	ardware: US	B-Blaster [USB-0]				2	•
Add File			Hardware frequency: Available hardware it	ems						Hz
Save File	TDI		Hardware	Server	Port			Add Hard	dware	
Add Device		4153	USB-Blaster	Local	USB-0			Remove H	ardware	
I≌Down									Clo	se

图 3.46 Quartus Prime 编程下载及驱动设置页面

(2)确认驱动为 USB-Blaster[USB-0]后,选择 Add File 添加工程输出文件中的 pof 格式文件,勾选 Program 列和 Verify 列,单击 Start 按钮进行 FPGA 加载。Quartus Prime 选择编程文件并启动下载页面如图 3.47 所示。

Programmer - E:/FPGA_Projects/LED_shining - LED_shining - [LED_shining.cdf]*	_	
<u>Eile Edit View Processing Iools Window H</u> elp	Search	Intel FPGA
Hardware Setup USB-Blaster [USB-0] Mode: JTAG		
File Device Checksun Usercode Program, Verify Blank-Examine	Security Bit	Erase Reac ISP ecks CLAMI
Dutput_files/LED_shining.pof 10M08SCM153 0266D 00089 V Auto Deter UFM V		
X Delete		
Change File Save File Add Device TDJ TDJ TDJ TDJ TDJ TDJ TDJ TDJ		
1% Down		

图 3.47 Quartus Prime 选择编程文件并启动下载页面

(3) FPGA 加载完成,界面中 Progress 状态显示 100%(Successful)。

至此,我们完成了 LED_shining 示例的开发流程,可以观察开发板现象。

8. 设计仿真

Quartus Prime 在安装时可选择仿真软件是否一起安装,最新的版本 Quartus Prime 23.1 可一起安装适用 Intel FPGA 的 QuestaSim,之前的版本可选择安装 ModelSim-Intel FPGA。当然我们也可以自己安装第三方仿真软件,除 QuestaSim 和 ModelSim 以外还可以安装 Active-HDL、Riviera-PRO 等仿真软件。所以仿真软件是独立存在的,Quartus Prime 只是预留了启动接口。

(1) 仿真文件:提前准备测试文件(Textbench)LEDshiningtb.v,测试文件源码见代码 3.2。

(2)选择菜单栏中的 Assignments→Settings 选项,或单击工具栏中的 Settings 按钮,进入设置界面。选择菜单栏中的 Simulation 选项,单选 Compile test bench,单击 Test Benches 按钮。Quartus Prime 仿真设置界面如图 3.48 所示。

Category:	Device/Board
General Files Libraries 'P Settings 'P Catalog Search Locations Design Templates 'Operating Settings and Conditions Voltage Temperature 'Compilation Process Settings Incremental Compilation 'EDA Tool Settings Design Entry/Synthesis Simulation Board-Level 'Compiler Settings 'YHDL Input Defign Analyzer Assembler Design Assistant Signal Tap Logic Analyzer Logic Analyzer Interface Power Analyzer Settings SSN Analyzer	Simulation Specify options for generating output files for use with other EDA tools. Iool name: ModelSim Rug gate-level simulation automatically after compilation EDA Netlist Writer settings Eormat for output netlist: Verilog HDL Time scale: 1 ps Output directory: simulation/modelsim Map illegal HDL characters Dptions for Power Estimation Construct Mathematical Parts & Draw & D
	Generate Value Change Dump (VCD) file script Script Settings Design instance name: More EDA Netlist Writer Settings NativeLink settings O Log Compile test bench: Use script to set up simulation: Script to compile test bench: More NativeLink Settings Reset WEinv Software OK Cancel Apply Help

图 3.48 Quartus Prime 仿真设置界面

(3) 单击 Test Benches 按钮后,在弹出的对话框中单击 New 按钮,填写 Test bench name,按照目录添加测试文件,如图 3.49 标识顺序,最后单击 OK 按钮回到设计界面。 Quartus Prime 仿真设置-添加测试文件界面如图 3.49 所示。

(4) 选择菜单栏中的 Tools→Run Simulation Tool→RTL Simulation 选项,或单击工 具栏中的 RTL Simulation 按钮,Quartus 软件会自动启动 Modelsim 软件。Quartus Prime 仿真设置-启动 RTL 仿真界面如图 3.50 所示。

(5) ModelSim 软件启动后自动完成代码编译, ModelSim 仿真-启动界面如图 3.51 所示。

∠ Test Benches	\times	✓ New Test Bench Settings ×
Specify settings for each test bench.		Create new test bench settings.
Existing test bench settings: Name Top Level Module	1 New Design Instance Run For Edit Delete OK Cancel Help	2 Iest bench name: LED_shiningtb Top level module in test bench: LED_shiningtb Use test bench to perform VHDL timing simulation Design instance name in test bench: NA Simulation period Image: Imag
 EDA Tool Settings Design Entry/Synthesis Simulation Board-Level Compiler Settings VHDL Input Verilog HDL Input Default Parameters Timing Analyzer Assembler Design Assistant Signal Tap Logic Analyzer Logic Analyzer Interface Power Analyzer Settings SSN Analyzer 	Output girectory: simulation/modelsim Map illegal HDL characters Options for Power Estimation Generate Value Change Dump (VCD) f Design instance name: More EDA Netlist Writer Settings NativeLink settings Ogne © Compile test bench: Use script to set up simulation: Script to compile test bench: More NativeLink Settings	End simulation at 5 - Test bench and simulation files Eile name: 3 Add 4 File Name Library HDLV Remove Up Down Properties OK Cancel Help Test Benches Reset

图 3.49 Quartus Prime 仿真设置-添加测试文件界面

Quartus Prime Lite Edition - E:/FPGA_Projects/LED_shining - LED_shining

<u>File Edit View Project Assignments Proce</u>	essing	<u>T</u> ools <u>W</u> indow <u>H</u> elp	
		Run Simulation Tool	■ <u>B</u> TL Simulation
		🐁 Launch Simulation Library Compiler	🕅 <u>G</u> ate Level Simulation
Project Navigator 🖹 Files 🔹 🤜 🗷	•	L 🗞 Launch Design Space Explorer II	
🏲 Files	53	S Iiming Analyzer	
LED_shining/LED_shining.v	26	Advisors):1k;
ELD_shining/LED_shiningtb.v	27 28 29 30 31	 Chip Planner Design Partition Planner Netlist Viewers 	•
Tasks Compilation ▼ = 100 88	32 33 34 35 36 37 38 39	Signal Tap Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor Signal Probe Pins Programmer	
Task	41	JTAG Chain Debugger	
Compile Design	42	Fault Injection Debugger	
Analysis & Synthesis	44	System Debugging Tools	Lctive low
Fitter (Place & Route)	45 46	IP Catalog	
? Assembler (Generate program	47	Nios II Software Build Tools for Eclipse	
► ► Timing Analysis	48	晷. Platform <u>D</u> esigner	
• • •	•	/ Tcl Scripts	(10)

图 3.50 Quartus Prime 仿真设置-启动 RTL 仿真界面

(6)选择需要观察波形的信号,右击,在弹出的快捷菜单中选择 Add Wave 选项,这样 就将对应信号添加至 Wave 窗口。ModelSim 仿真-添加观察信号界面如图 3.52 所示。

(7) ModelSim 仿真-设置仿真时间启动仿真界面如图 3.53 所示。在弹出的 Wave 窗口中 单击工具栏中的 Restart 按钮,在弹出的 Restart 窗口单击 OK 按钮,复位仿真 Wave 窗口。修 改工具栏中仿真时间,单击工具栏中的 Run 按钮,进行仿真,仿真波形如图 3.53 所示,仿真完成。

ModelSim SE-64 10.1c						-		×
File Edit View Compile S	Simulate Add Str	ucture Tools Layout	Bookmarks	Window Help				
B-68 B % & % B	a@⊇⊇ ©·	И В Нер	3	\$##\$\$ \$				
<u></u> <u></u> • • • <u></u> • <u>_</u> • <u>_</u>	100 ps 🝨 🔃 🔃	B 2 0 1 1 1 1 4	1 * 2	* * 🕀 🛣 📙 Layou	t Simulat	te 💌	1	
ColumnLayout AllColumns		▼	ā•@ [M II 0 10 II ALL #	N 3	II Bo	T	
] 3+ € ≫ 3+ 3+ € +E :	⊊·×·@ <u></u> ⊠		h					
📶 Library 🛨 🖻 🛛	🕮 sct 🕬 🛨 🗗 🗶	😰 sim - Default 🚃 🛲	+#×	Si Objects:	WW		ault 2000	± ª ×
Name _	▼Name	Instance 1	7 Design unit	▼ Name	Value K	ind Mode		^
⊕ work			LED_shining	CLK_DIV_PERIOD	1100 Pa	ara Internal		
+altera_Insim_ver		🖬 📕 LED_shining_uut	LED_shining	🌮 dk_in	St0 N	iet In		
		— #ASSIGN#15	LED_shining	III rst_n_in	St1 N	et In		
+		- #ASSIGN#14	LED_shining	🐁 led1	St0 N	et Out		
+ fiftyfivenm_ver		ALWAYS#19	LED_shining	🐁 led2	St0 N	et Out	Welcome	to th
+-1 lpm_ver		- #INITIAL#13	LED_shining	🚸 dk_dv		egisInternal	NOTE:	Exte
+ sgate_ver		- #INITIAL#7	LED_shining	🖸 🛷 ant	00000P	ack Internal	1	
		ALWAYS#9	LED_shining		_			
⊕floatfixlib		#vsim_capacity#		Processes (Active)	=1000			
⊕_ff_ machxo2				▼ Name	Type (filte	red) S 📩		
🕺 🕺 🕺 🕺 🕺 🕺 🕺 🕺 🕺	ļ			A #ASSIGN#14	Assinn			~
<u> </u>	• •	4	•	•		•	4	>
Q Transcript								+ # ×
4 min -all			- crited a					-
Tun -arr								
								_
1								-
Now: 3,661,642,920 ns Delta: 0	sim:/LED_shining_t	b/LED_shining_uut	Кеер	0				

图 3.51 ModelSim 仿真-启动界面

🏨 Library 🚃 📰 🖃 🕈	×	🛗 ect 📖 🛨 🗗 🗙	💭 sim	- Default	=:::::=	- # # X	Di Objects :				:+ # ×
▼Name	-	▼ Name	♥ Insta	nce	Ū	Design unit	▼ Name		Value	Kind	Mode
• M work			0- 1	LED_shining_tt	b	LED_shining	🔷 CLK_	DIV_PERIOD	1100	Para	Internal
• altera_Insim_ver	-11		9	LED_shinin	g_uut	LED_shining	🥠 dk_in	F	St0	Net	In
altera_mf_ver	11			-@ #ASSIC	GN#15	LED_shining	🥔 rst_n	_in	St1	Net	In
altera_ver	1			-@ #ASSIC	GN#14	LED_shining	🔶 le	View Declaratio	m		ŧ
• fiftyfivenm_ver	-1			L@ #ALWA	AYS#19	LED_shining	🐟 le	View Memory (ontents		1
• Ipm_ver				#INITIAL#	13	LED_shining	b 🔶	there is a second of a			ernal
• sgate_ver				#INITIAL#	7	LED_shining	• 🔶 🛛	Add Wave		Ctrl+W	ernal
• rt_work				#ALWAYS:	#9	LED_shining	1	Add Wave Nev	v		
🛨 📶 floatfixib			8	#vsim_capacit	y#		Proce	Add Wave To			, d ×
+ machxo2							▼Name	Add Date fam		CH ID	5 1
fill mc2 lib (emptv)	-						<u> </u>	Add Datatiow		C01+D	A -
1 ·		• •	1			•	1	Add to			· 🗉

图 3.52 ModelSim 仿真-添加观察信号界面



图 3.53 ModelSim 仿真-设置仿真时间启动仿真界面

3.3.3 小脚丫 FPGA(STEP FPGA)线上开发平台

FPGA的集成开发软件(IDE)往往都非常庞大,所需安装空间少则几 GB 多则几十GB,运行时对内存和 CPU 也有要求。针对以上问题,小脚丫 FPGA 开发了一款 FPGA 线 上开发平台,将 IDE 安装至服务器端,用户在网页端完成所有的开发操作和信息交互,目前 支持 STEP MXO2 和 STEP MAX10 FPGA 开发板的开发设计。

在小脚丫网站 www.stepfpga.com 注册账号后就可以体验使用线上设计工具,基于浏 览器端的开发环境,无需下载 FPGA 设计工具到本地计算机,使用方便简单。STEP FPGA 线上开发平台主页如图 3.54 所示。



图 3.54 STEP FPGA 线上开发平台主页

1. 新建项目

单击新建项目会进入创建新项目界面,左侧是项目文件组成,可以在这里查看项目所需 文件以及软件生成的日志等。在上方是 FPGA 设计的流程图标,可以按顺序创建源文件、 综合、分配引脚、产生编译文件、仿真以及下载工程和编译文件。

填写项目的一些基本信息,例如项目名称、设备、项目标签、描述等。STEP FPGA 线上 开发平台-新建项目页面如图 3.55 所示。

2. 创建源文件

提交创建项目后进入编辑界面,单击界面上的"+"号可以新建源文件(也可以从其他项目复制文件)。STEP FPGA 线上开发平台-新建文件页面如图 3.56 所示。

3. 设置顶层文件

源文件创建完成后需要确认顶层文件,STEP FPGA 线上开发平台-设置顶层文件页面 如图 3.57 所示,单击左侧栏文件名称后的箭头图标。设置完成后文件名称后的箭头图标消 失,说明该文件目前是顶层文件。如果项目中包含多个源文件,也进行同样设置。

4. 逻辑综合

设置好顶层文件后单击"逻辑综合",开始进行综合。综合完成后显示日志,如果有错误

	首页 知识库	电子森林	淘宝店铺				
		〇 合 管脚分配					
■ 设计文件	🖃 创建						
■ 仿真文件	项目名称	LED_shining					
▲ 查看日志		字母、数字、下	「画线 (字母开头) 最	少2个字符,最多10	0个字符		
₽ 运行结果	设备	STEP-MXO2-LF	°C			\$	
		选择你的开发极	Ź.				
	项目标签	LED流水灯				Add	
		回车或者单击下	拉选项新建,建议标	签数 3-5个,如:	组合逻辑。时序道	逻辑	
	描述						
						•	
	权限	◎ 公开	○ 保密				
			提交				

图 3.55 STEP FPGA 线上开发平台-新建项目页面

	首页	知识库	电子森林	淘宝店铺				
💮 LED shining 👻	> 漂代码							
■ 设计文件	(142177			
图 仿真文件				新建文件	#			×
▲ 查看日志				文件名	LED_shining]		
₽ 运行结果				1字母、	数字、下划线 (字	母开头) 最少2个	字符,最多100个字符	
				类型	.v			•
					□ 设为仿真文	任	取消提到	×

图 3.56 STEP FPGA 线上开发平台-新建文件页面

的话会显示报错信息。STEP FPGA 线上开发平台-逻辑综合页面如图 3.58 所示。

5. 引脚分配

综合通过后进行引脚约束。这里主要对 FPGA 引脚的编号进行分配。STEP FPGA 线上开发平台-引脚分配页面如图 3.59 所示,小脚丫板卡可配置的引脚分为板上部分 (Internal)和扩展部分(External)。板子上的数码管、拨码开关、按键、LED 和三色灯属于板上部分。板子两侧扩展的 IO 引脚属于扩展部分。

	首页	知识库	电子森林	淘宝店铺				
	> 源代码	。 逻辑综合	く 管理分配) FPGA映射	 _{文件下载}	-₩- 65頁	☆ 示例項目	! 使用说明
■ 设计文件	LED_shin	ing.v × (+)					
	1 mo 4 (3 in	dule LED_sh	ining //clk in =	12mhz				
■ 仿真文件	4 in 5 ou	put rst_n_i tput led1,	n, //rst_n_ //led1 输出	in, 低电平有效	文			
☑ 查看日志	7);	tput ledz	//1ea2 1 田					
₽ 运行结果	9 pa 10	rameter CLK	DIV_PERIOD	= 12_000_000;				
	11 re 12	g clk_div=0	;					
	13 //	wire led1,1	ed2;					
	14 as	sign led1 =	clk_div;					
	16	sign ieuz =	~CIK_UIV,					
	17 re	g[24:0] cnt	=0;					
	18 al	ways@(posed)	ge clk_in or	negedge rst_	n_in) begin			
	19	if(!rst_n	_in) begin					
	20	clk d	iv<=0:					
	22	end else	pegin					
	23	if(cn	t==(CLK_DIV_	PERIOD-1)) cn	t <= 0;			
	24	else if(cm	CNT <= CNT +	1'01; FRTOD>>1)) cl	k div <= 0.			
	23 24 25	if(cn else if(cn	c==(CLK_DIV_ cnt <= cnt + t<(CLK_DIV_P	1'b1; ERIOD>>1)) cl	τ <= 0; k_div <= 0;			



	首页	知识库	电子森林	淘宝店铺			
😭 LED shining 👻	> 源代码		○ 引脚分配			会 示例项目	! 使用说明
■ 设计文件	LED_shini	ng.v × de	bug.log ×	\oplus			
LED_shining.v 📋 🕤	debu	g.log					
▣ 仿真文件	Succe	ss					
▲ 查看日志							
₽ 运行结果							

图 3.58 STEP FPGA 线上开发平台-逻辑综合页面

分配引脚过程很简单,单击想要分配的外设,会自动弹出所有的输入输出信号,选择确 定就可以。所有信号分配完毕单击保存按键,如果分配错误可以选择重置。板上外设被分 配使用后会以黄色来标记。

6. 映射生成流文件

分配完成后单击"FPGA映射"按钮来产生 FPGA 的配置文件,完成后生成日志,如果

有错误会显示报错信息。STEP FPGA 线上开发平台-FPGA 映射页面如图 3.60 所示。

	首页	知识库	电子森林	淘宝店铺					English	group003 +
			EN EN							
	〇 管F	份配								
	开发机	ž •	Internal	External						
回 仿真文件			T ¤**o o	003500000	> ³⁰ 00000 ²⁵	00000181	Key	Inout	Switch	Innut
	-		- L - L		E 111		rst_n_in	L14 ×	Switch1	M7 ×
☑ 查看日志	STEP-MX	D2 V2.2	H	<u> </u>		<u> </u>	KEY2	M13 ×	Switch2	M8 ×
	_	_	- T. M.				KEV3	M14 ×	Switch3	M9 ×
₽ 运行结果	1		D. 0 0	0 0.0 0 0 0 0		0 0 0 0 D .	KEY4	N14 ×	Switch4	MT0 ×
		_			10 15	20	Clock	Input	LED	Output
	STEP-M	KO2-C	数码管1	Output	数码管2	Output	dk_in 🗲	C1 ×	led1	N13 ×
			SEG-A1	A10 ×	SEG-A2	C12 ×	UART	Input	led2	M12 ×
		_	SEG-B1	CI1 X	SEG-B2	B14 ×	RXD	A3 X	LED3	P12 ×
	• III .		SEG-C1	F2 X	SEG-C2	лх	TXD	A2 X	LED4	M11 ×
			SEG-D1	E1 X	SEG-D2	H1 X			LED5	P11 ×
	STEP-MX	O2-LPC	SEG-E1	E2 ×	SEG-E2	H2 X	RGB_LED	Output	LED6	N10 ×
		_	SEG-F1	A9 ×	SEG-F2	B12 ×	R_LED1	M2 ×	LED7	N9 ×
			SEG-G1	89 ×	SEG-G2	A11 ×	G_LED1	NZ X	LEDB	P9 × 🗸

图 3.59 STEP FPGA 线上开发平台-引脚分配页面

	首页	知识库	电子森林	淘宝店铺							
	> 源代码) FPGA映射		-₩- 仿真	★ 示例项目	! 使用说明			
🗉 设计文件	LED_shining	g.v × d	ebug.log ×	run.log ×	(\pm)						
	######		*****	****	*****	+##					
I 仿真文件	### Lat	tice Family :	MachXO2								
	### De	vice : LCMXC	02-4000HC								
▲ 查看日志	### Package : CSBGA132 ### Speed : 5										
₽ 运行结果											
	######		*****	*****	******	+##					
	Optimiz The to	ation goal =	Balanced								
	Target f	requency =	1.000000 MHz.								
	Maximu Timing	ım fanout = path count =	1000. = 3								
	BRAM u DSP usa	utilization = " age = true	100.000000 %								
	DSP uti	lization = 10	% 0000000								
	resolve	_mixed_drive	rs = 0								
	tix date	a clocks = 1									

图 3.60 STEP FPGA 线上开发平台-FPGA 映射页面

7. 下载文件

可以在下载界面下载打包整个工程,也可以只下载需要的 FPGA 编程流文件(Lattice 小脚丫产生 jed 文件, Intel 小脚丫产生 pof 和 sof 文件)。STEP FPGA 线上开发平台-下载 文件页面如图 3.61 所示。

8. 仿真工具使用

小脚丫线上设计工具也可以进行在线功能仿真操作,方便验证设计。

1) 新建仿真文件

单击新建文件,会弹出窗口,定义文件名称,然后记得勾选"设为仿真文件"选项。



STEP FPGA 线上开发平台-新建仿真文件页面如图 3.62 所示。



	首页 知识库 电子森林 淘宝店铺
💮 LED shining 🔸	
■ 设计文件	LED_shining.v × (+)
回 仿真文件	1 moule Leu_snining 2 (3 input clk_in, //clk_in = 12m 4 input cet_n in //ctk_in
☑ 查看日志	5 output led1, //led1 输出 文件名 LED_shiningtb
₽ 运行结果	<pre>7); 8 parameter CLK_DIV_PERIOD = 12 10 11 reg clk_div=0; 12 //wire led1,led2; 14 assign led1 = clk_div; 15 assign led2 = ~clk_div; 16 17 reg[24:0] cnt=0; 18 always@(posedge clk_in or negedge rst_n_in) begin 19 if(irst_n_in) begin 20 clk_div<=0; 21 clk_div<=0; 22 end else begin 23 if(cnt=(CLK_DIV_PERIOD-1)) cnt <= 0; 24 else cnt <= nt + 1'b1; 25 if(cnt<(CLK_DIV_PERIOD-1)) clk_div <= 0; 26 clk_div <=0; 27 clk_div <=0; 28 clk_div <=0; 29 clk_div <=0; 20 clk_div <= 0; 20 clk_div <= 0; 20 clk_div <= 0; 20 clk_div <= 0; 21 clk_div <= 0; 22 clk_div <= 0; 23 clk_div <= 0; 24 clk_div <= 0; 25 clk_div <= 0; 26 clk_div <= 0; 27 clk_div <= 0; 28 clk_div <= 0; 29 clk_div <= 0; 20 clk_div <= 0; 21 clk_div <= 0; 22 clk_div <= 0; 23 clk_div <= 0; 24 clk_div <= 0; 25 clk_div <= 0; 26 clk_div <= 0; 27 clk_div <= 0; 28 clk_div <= 0; 29 clk_div <= 0; 20 clk_div <= 0; 21 clk_div <= 0; 21 clk_div <= 0; 22 clk_div <= 0; 23 clk_div <= 0; 24 clk_div <= 0; 25 clk_div <= 0; 26 clk_div <= 0; 27 clk_div <= 0; 28 clk_div <= 0;</pre>

图 3.62 STEP FPGA 线上开发平台-新建仿真文件页面

2) 编辑仿真文件

编辑仿真文件然后保存。STEP FPGA 线上开发平台-编辑仿真文件页面如图 3.63 所示。

3) 仿真并查看波形

单击"开始仿真",结束后可以看到输出波形,如果出错请查看出错信息。STEP FPGA 线上开发平台-查看仿真波形页面如图 3.64 所示。

	首页 知识库 电子森林 淘宝店铺
	《小 長: ○ → → ② ① 第代語 運輸会会 管部分配 FPGAMM 文件下載 65萬 示明第日 使用液明
回 设计文件	$\texttt{LED_shining.v} \times \texttt{LED_shiningtb.v} \leftrightarrow $
回 仿真文件	1 `timescale 1ns / 100ps 2 module LED_shiningtb; 3
	4 parameter CLK_PERIOD = 10; 5
∑ 查看日志	6 reg clk; 7 initial clk = 1'b0; 8 always #(CLK PERIOD/2) clk = ~clk:
₽ 运行结果	<pre>9 10 reg rst_n; //active low 11 initial begin 12 rst_n = 1'b0; 13 #20; 14 rst_n = 1'b1; 15 end 16 17 wire led1,led2; 18 LED_shining #(.CNT_1S (19)) u_LED_shining (19 .clk (clk), 20 .rst_n (rst_n), 21 .led1 (led1), 23 .led2 (led2) 24); 27 </pre>



🗊 decode24 👻	>	i Ballon	O THOM	() FPGA8981	.⊥ xa∓u	≁ ⊬ ₀,≖	e e e e e e e e e e e e e e e e e e e	() 10.112041				
🗉 设计文件				仿真文件:	tb.v		仿真时间:	2000	0 ns +	开始仿真		
				740 ns		750 ns	760 ns	0 ns	770	ns	780 ns	
图 仿真文件	a	F		11	2	_	3	<u> </u>		1	-	2
	01											
▲ 倉棚日志	led	X		d	ь		7	•		d		ь
₽ 运行结果	i i											
								CTRI SHIF	*風标決轮 「 * 風标決轮	编說還形囿 ::::::::::::::::::::::::::::::::::::		

图 3.64 STEP FPGA 线上开发平台-查看仿真波形页面