



面向“工程教育认证”计算机系列课程规划教材

Xilinx公司大学计划推荐教材

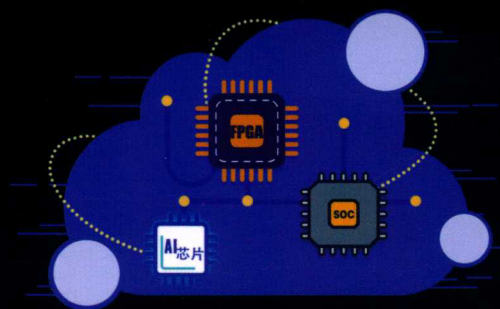
Xilinx—教育部产学合作教学改革项目

Computer Composition Principle Online Experiment Course
Teaching and Practice of FPGA Remote Experiment Platform

计算机组成原理在线实验教程

FPGA远程实验平台教学与实践

◎ 柴志雷 李佩琦 吴子刚 阳文敏 编著
Chai Zhilei Li Peiqi Wu Zigang Yang Wenmin



- “线上线下结合、课内课外一体化”的实验环境
- 基于RISC-V及开源项目进行实验
- 采用Python程序与所设计的硬件模块进行交互

清华大学出版社



计算机组成原理在线实验教程

FPGA远程实验平台教学与实践

教学资源

● 源代码

上述资源可到清华大学出版社网站本书页面下载。

本书特色

软硬件融合的计算机系统能力培养日益成为人才培养及行业发展的迫切需求，在此背景下，本书针对计算机组成原理实践环节涉及的实验平台、交互方式、实验内容进行了改进，以更好地适应系统能力培养的需求。本书具有如下特色：

- **实验平台方面。**介绍了“线上线下结合、课内课外一体化”的实验环境。课内采用线下方式，让学生对真实硬件有更直观的认识。课外采用在线实验，使学生可以随时随地远程访问实验设备进行实验。
- **交互方式方面。**介绍了采用Python程序与所设计的硬件模块进行交互的方式。以“软件定义交互”突破实验平台物理I/O的局限，在提升交互便利性的同时可锻炼学生的软硬件协同设计能力。
- **实验内容方面。**考虑到硬件开源开放的背景，本书实验内容由浅入深，最终介绍如何基于RISC-V及开源项目进行实验，使学生所学内容在将来可以更好地和企业要求进行对接。

作者简介

柴志雷 江南大学物联网工程学院副教授、研究生导师，获复旦大学计算机系统结构方向博士学位。中国计算机学会体系结构专业委员会及嵌入式系统专业委员会委员，江苏省计算机教育专委会副主任。长期从事计算机系统结构方面的教学与科研工作，研究兴趣为新型计算机体系结构、FPGA可重构计算。承担了国家自然科学基金、科技委创新特区、无锡市物联网专项资金等科研项目，教育部高等学校计算机类专业教学指导委员会“系统能力培养”课程建设试点院校项目、Xilinx—教育部产学研合作教学改革项目等教学项目。研究成果在FPGA、FPL、FPT、ACM TACO等FPGA会议与期刊发表学术论文30余篇，获得授权发明专利4项。研究成果还在美国安富利公司香港研发中心、上海电气集团机器人实验室等单位得到应用，获得中国商业联合会科技进步一等奖1项。

课件下载·样书申请



书圈

清华社官方微信号



扫我有惊喜

上架指导：计算机

ISBN 978-7-302-53779-3



9 787302 537793 >

定价：39.00元



面向“工程教育认证”计算机系列课程规划教材

Xilinx公司大学计划推荐教材

Xilinx—教育部产学合作教学改革项目

计算机组成原理在线实验教程

FPGA远程实验平台教学与实践

◎ 柴志雷 李佩琦 吴子刚 阳文敏 编著

Chai Zhilei

Li Peiqi

Wu Zigang

Yang Wenmin

清华大学出版社

北京

此为试读,需要完整PDF请访问: www.ertongbook.com

内 容 简 介

本书以线上线下相结合的方式,使用 FPGA 平台完成数字电路及计算机组成原理实验,并用 Python 编程的方式与自己设计的硬件系统交互,来进行系统验证与调试。全书包含三大部分内容:首先是实验所用的软硬件工具和平台;其次是实验设计方法;最后介绍实验内容安排的建议。书中主要讲述了 PYNQ 平台与基于 Python 的软硬件交互、Vivado 开发工具、硬件描述语言简介、RISC-V 开源项目及组成原理实验内容建议等。

本书可用作高等院校计算机类、电子信息类专业的数字电路与组成原理相关课程的实验教材,也可作为 FPGA 及嵌入式系统软硬件学习的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机组成原理在线实验教程:FPGA 远程实验平台教学与实践/柴志雷等编著. —北京:清华大学出版社,2020.1

面向“工程教育认证”计算机系列课程规划教材

ISBN 978-7-302-53779-3

I. ①计… II. ①柴… III. ①计算机组成原理—实验—高等学校—教材 IV. ①TP301-33

中国版本图书馆 CIP 数据核字(2019)第 200064 号

责任编辑:刘 星
封面设计:刘 键
责任校对:梁 毅
责任印制:丛怀宇

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载:<http://www.tup.com.cn>,010-83470236

印 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:11

字 数:265 千字

版 次:2020 年 1 月第 1 版

印 次:2020 年 1 月第 1 次印刷

定 价:39.00 元

产品编号:068985-01

前言

PREFACE

随着人类社会迈向万物互联的物联网时代,借由人工智能技术替代手工方式对大数据进行高效实时的处理已成为社会发展的必然选择,万物互联的未来之路一定走向万物智能,而人工智能的构成要素包含数据、算法及计算力。在此背景下,无处不在的人工智能对计算系统的计算能力提出了挑战性的需求,目前如火如荼的边缘计算即是对这一挑战的积极应对。而使得计算力需求更加充满挑战的原因是:长期以来半导体行业遵循的摩尔定律和登纳德缩放定律已难以为继,这意味着芯片计算能力的提升已经无法在保持架构基本不变的情况下仅靠工艺的提升来完成,也意味着未来我们将更多地需要依赖架构创新,为应用定制芯片、根据应用需求深度优化软硬件系统。因此高校培养的人才也要比过去更多地掌握硬件设计及软硬件协同优化的知识,这也是近年来教育部高等学校计算机类教学指导委员会一直在大力推进“计算机系统能力”培养的原因。

经过若干年的努力,与系统能力紧密相关的课程,尤其是以“计算机组成原理”为代表的课程的教学及实践已经有了长足的进步。越来越多的学校开始采用 FPGA 作为实验教学的平台,并尝试将“数字逻辑”“计算机组成”“体系结构”“编译原理”甚至“操作系统”等多门课程的实践环节贯通,以此培养学生的软硬件系统能力。以清华大学、北京航空航天大学、浙江大学、华中科技大学、东南大学、同济大学等为代表的一些高校在此方面做了大量的工作并已取得了丰硕的成果。

总体上看,目前围绕计算机组成原理进行的软硬件系统能力培养具有如下特点:

(1) 从实验平台来看,已采用 FPGA 的学校主体上仍是以线下的 FPGA 实验设备为载体进行实验,从时间和空间上限制了学生像软件类的实验一样随时随地进行。学生实验的达成度严重受限于实验课时。

(2) 从实验的调试和交互方式来看,目前的实验主要使用 FPGA 实验装置上的拨码开关、LED、串口等物理 I/O 进行,由于数量有限,极大地影响了调试的便利性。

(3) 从实验内容来看,鉴于《计算机组成与设计:软硬件接口》一书的巨大影响力,多数学校选择的实验内容是兼容 MIPS 指令集的处理器的设计。虽然 MIPS 指令集非常经典,用于教学时理论及实践素材都十分丰富,但难掩其产业界影响偏小、产学研融合深度和广度有限的缺憾。因此常见的组合是,在讲授组成与体系结构时采用 MIPS 架构,但在讲授后续嵌入式系统时却几乎都切换成了 ARM 体系。

因此,为了结合人工智能时代计算系统的发展趋势进一步提升学生的系统能力,在前期教指委推动的系统能力培养工作基础上,计算机组成原理的实验教学还可以在如下方面进一步提升:

(1) 实验平台方面：探索“线上线下结合、课内课外一体化”的实验环境。课内采用线下方式，让学生对真实硬件有更直观的认识。课外采用在线实验，使学生可以随时随地通过远程访问实验设备进行实验。线上线下的使用方式保持一致，从而为学生提供可与软件实验相媲美的实验条件。尽管目前越来越多的高校认识到相比传统的插线式实验平台，采用FPGA进行组成原理实验教学将是未来的发展趋势。但现状是为数不少的学校仍然采用的是传统的插线式平台，在切换到FPGA方式时缺少便捷的“试水”途径。因此，线上方式及相应的参考内容可为系统能力培养较为薄弱的地区提供一种易于推进的模式。

(2) 实验的调试和交互方式：本书采用Python程序与所设计的硬件模块进行交互。鉴于Python语言已成为人工智能时代最为流行的编程语言，使用Python可以突破实验平台物理I/O的局限，在提升交互便利性的同时可锻炼学生的软硬件协同设计能力，为学生今后从事人工智能计算相关的工作奠定良好的系统基础。对于线上方式，更没有必要刻板地模仿线下设备的物理I/O交互却又难以达到线下使用的真实感。

(3) 实验内容和设计方法：本书介绍基于RISC-V的开源开放生态的处理器设计。鉴于开放指令集RISC-V在学术、工业界关注度的急剧增长及众多RISC-V指令集的开源处理器项目的快速发展，从目前看其最有希望构建起允许学生探究底层实现细节并促进产学研深度融合的生态系统，而这是目前其他指令集系统不具备的优势。在硬件开源开放的背景下，虽然逐条增加指令的增量式处理器设计方法依然重要，但如何基于开源资源快速完成自己的系统也应该开始引起更多的重视。因此，本书建议的实验内容由浅入深，最终介绍如何基于RISC-V及开源项目进行实验，以使学生的所学在未来可以更好地和企业界进行对接。当然，本书提供的实验范例也仅供参考，各学校目前所采用的内容也都可以方便地在本书所介绍的平台上进行。

本书介绍如何以线上及线下方式使用PYNQ平台进行实验。PYNQ可以支持Python，它的核心芯片为Xilinx的SoC方式的Zynq FPGA。基于该实验平台，不仅可以满足本书中冯·诺依曼架构的经典处理器的实验教学需求，而且为学生进一步学习非冯·诺依曼架构的硬件加速器奠定好实验平台的基础。在当前人工智能发展对计算能力不断提出更高需求的大背景下，针对特定问题设计专用加速器成为趋势，而这些专用加速器通常都是非冯·诺依曼架构的。因此，学生在学习计算机组成原理课程，掌握经典的冯·诺依曼架构之后，进一步学习突破冯·诺依曼架构限制的专用加速器的知识是培养学生系统能力的未来趋势。

本书包含三大部分内容：首先是实验所用的平台及开发工具；其次是设计方法；最后介绍实验内容安排的建议。具体如下：

第1章 PYNQ开发平台介绍。本章介绍本书实验所用的PYNQ平台，除介绍PYNQ开发板及开发板所用的主芯片Zynq-7020 FPGA芯片之外，还介绍了该款开发板所支持的PYNQ框架及其理念。

第2章 实验环境的准备。本章介绍在采用PYNQ进行组成原理实验时，线下和线上的实验环境分别如何准备，如何通过线上或线下的方式使用PYNQ平台。

第3章 基于PYNQ的组成原理实验流程概览。本章主要介绍本书线上及线下所用的FPGA实验平台——PYNQ上组成原理实验的概要流程，为读者建立一个整体的概念，

方便理解后续的章节内容。

第4章 Vivado 开发流程。本章介绍在 PYNQ 平台上进行组成原理实验时,处理器及硬件模块设计所用的工具——Vivado,详细介绍使用 Vivado 进行开发时的流程及一些关键开发知识。

第5章 基于 Python 的 I/O 交互。本章主要介绍如何基于 Jupyter Notebooks,通过编写 Python 代码和用户设计的硬件系统进行交互,完成组成原理实验的验证和调试。

第6章 硬件描述语言简介。本章主要介绍 Verilog/VHDL 语言的基本设计方法和基本模块示例,更详细的介绍请参考相关书籍。

第7章 基于开源 CPU 的组成原理实验。本章介绍基于开源指令集 RISC-V 及开源的 RISC-V 处理器项目进行组成原理实验的理念及设计方法。同时为了扩展读者的知识面,也简单介绍了常规的逐条增加指令的增量式设计方法。

第8章 实验内容设计。本章给出一些供参考的组成原理实验内容的设计,高校在开展组成原理实验教学时可以直接选用本章的实验,也可以根据各自学校的实际情况进行内容的重新设计和规划。

本书前言部分及第1~5、7、8章由柴志雷编写;第6章及第4、5、7、8章中的代码部分由李佩琦、吴子刚编写;阳文敏提供了在线平台的支持和实验内容的审核;李康等也参与了部分内容的编写和校对;中北大学的秦品乐教授和于一老师参与了本书的审阅并对内容安排及呈现方式提出了宝贵的修改建议。全书由柴志雷统稿及审阅。在本书内容设计及成稿的过程中,江南大学计算机科学与技术专业的本科生提供了宝贵的反馈意见,唐雨馨、刘昊鑫、李莹莹、李慧琳等同学参与了实验案例的设计。在本书的撰写过程中,得到了江南大学物联网学院及计算机科学与技术系的大力支持。本书的工作还得到了教育部高等学校计算机类专业教学指导委员会“系统能力培养”课程建设试点院校项目、Xilinx—教育部产学合作教学改革项目的支持。在成书的过程中还得到了南京大学袁春风教授、太原理工大学强彦教授、赛灵思(Xilinx)创新研究院陆佳华先生等众多专家的鼓励与支持。在此一并表示感谢。

本书中涉及的一些源代码及资源,请在清华大学出版社官方网站本书页面获取。

最后,受个人能力所限,本书的内容难免会存在不妥之处,敬请读者批评指正,我们将会持续改进。

作 者

2019年7月

第5章 基于 Python 的 I/O 交互	54
5.1 Jupyter Notebook 介绍	54

目录

CONTENTS

第 1 章 PYNQ 开发平台介绍	1
1.1 PYNQ 开发板	1
1.1.1 PYNQ Z1	1
1.1.2 PYNQ Z2	2
1.2 Zynq 7020 芯片	3
1.3 PYNQ 框架	5
1.4 PYNQ 平台的使用	5
第 2 章 实验环境的准备	6
2.1 线下方式实验环境的准备	6
2.1.1 在本机安装 Vivado 软件	6
2.1.2 PYNQ 板卡的准备	6
2.1.3 使用 Jupyter Notebook 与 PYNQ 建立连接	9
2.2 线上方式实验环境的准备	10
第 3 章 基于 PYNQ 的组成原理实验流程概览	12
3.1 整体开发流程介绍	12
3.2 Vivado 开发流程概览	13
3.3 基于 Python 的硬件交互	14
第 4 章 Vivado 开发流程	15
4.1 创建工程	15
4.2 设计输入	20
4.2.1 原理图方式	20
4.2.2 Verilog/VHDL 方式	23
4.3 仿真	26
4.4 综合	28
4.5 引脚绑定(I/O 处理)	29
4.5.1 物理引脚的绑定	30
4.5.2 与监控模块(PS)的连接	30
4.6 实现	38
4.7 TCL 使用介绍	39
4.8 实例演示	40
4.8.1 原理图方式	41
4.8.2 Verilog 方式	49
第 5 章 基于 Python 的 I/O 交互	54
5.1 Jupyter Notebook 介绍	54

5.1.1	Jupyter 组件	54
5.1.2	Notebook 基础	55
5.1.3	Notebook 用户界面	56
5.2	使用 PYNQ Overlay 加载流文件	57
5.3	Python 引脚绑定	57
5.4	基于 Python 调试组合逻辑	58
5.5	基于 Python 调试时序逻辑	59
5.6	实例演示	61
5.6.1	上传 .bit 和 .tcl 文件	61
5.6.2	基于 Python 的 I/O 交互	61
第 6 章	硬件描述语言简介	63
6.1	“模块”的描述	63
6.1.1	输入/输出端口说明	64
6.1.2	数据对象和数据类型	64
6.1.3	顺序语句与并行语句	65
6.2	模块基本用法示例	65
6.2.1	八位乘法器	65
6.2.2	译码器	66
6.2.3	八位二进制比较器	67
6.2.4	JK 触发器设计	68
6.3	层次化设计	69
6.3.1	描述方式	69
6.3.2	层次化设计的写法	70
6.4	VHDL 语言基础	72
6.4.1	标识符	72
6.4.2	数据对象	72
6.4.3	数据类型	73
6.4.4	数据类型转换	74
6.4.5	运算符	75
6.4.6	运算符优先级	76
6.4.7	VHDL 常用语法	76
6.5	Verilog HDL 语言基础	81
6.5.1	数据类型	81
6.5.2	数字表示形式	81
6.5.3	parameter 定义常量	82
6.5.4	宏定义 'define	82
6.5.5	运算符及表达式	82
6.5.6	运算符优先级	84
6.5.7	Verilog HDL 常用语法	85
第 7 章	基于开源 CPU 的组成原理实验	87
7.1	RISC-V 指令集	87
7.2	基于 RISC-V 的逐条增加指令式实验	90
7.2.1	5 级流水介绍	91

7.2.2	单条指令的 RISC-V 处理器设计	91
7.2.3	2 条指令的 RISC-V 处理器设计	101
7.2.4	3 条指令的 RISC-V 处理器设计	103
7.2.5	10 条指令的 RISC-V 处理器设计	106
7.3	开源 RISC-V 处理器蜂鸟 E200 介绍	109
7.4	基于开源项目的 CPU 综合实验	112
7.4.1	从完整 SoC 项目中抽取出 CPU 内核上板验证	112
7.4.2	删减掉特定的部分并补全	112
7.4.3	扩展开源处理器的流水线级数	113
7.4.4	优秀工作的遴选方法	113
第 8 章	实验内容设计	114
8.1	基于原理图的实验	115
8.1.1	全加器	115
8.1.2	译码器	117
8.1.3	多路选择器	119
8.1.4	触发器与寄存器	120
8.1.5	移位寄存器	122
8.1.6	计数器	124
8.1.7	有限状态机	126
8.1.8	运算器/ALU	128
8.1.9	存储器	129
8.1.10	寄存器堆	130
8.1.11	总线	131
8.1.12	微程序控制器	134
8.2	基于 Verilog HDL 的实验	138
8.2.1	全加器	138
8.2.2	译码器	140
8.2.3	多路选择器	141
8.2.4	触发器与寄存器	142
8.2.5	移位寄存器	143
8.2.6	计数器	146
8.2.7	有限状态机	147
8.2.8	运算器/ALU	149
8.2.9	存储器	150
8.2.10	寄存器堆	151
8.2.11	总线	153
8.2.12	微程序控制器	155
8.2.13	中断	159
8.2.14	基于开源项目的 CPU 内核的实现	160
8.2.15	为开源 CPU 增加指令	161
8.2.16	增加开源 CPU 的流水线级数	162
参考文献	164

第 1 章

CHAPTER 1

PYNQ 开发平台介绍

1.1 PYNQ 开发板

1.1.1 PYNQ Z1

PYNQ Z1 是由美国迪芝伦公司(Digilent)推出的一款支持 PYNQ 框架(PYNQ 框架将在 1.3 节中介绍)的 FPGA 开发平台,其开发板如图 1-1 所示。

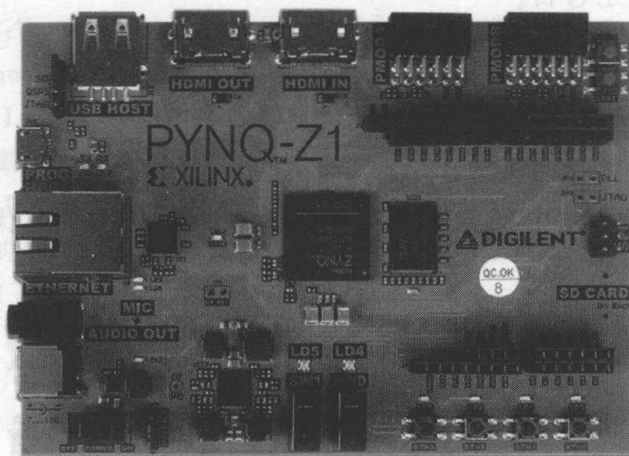


图 1-1 PYNQ Z1 开发板

PYNQ Z1 的具体配置如下:

- (1) 核心芯片: Zynq XC7Z020-1CLG400C。
- (2) 存储:
 - 带有 16 位总线@1050Mbps 的 512MB DDR3。
 - 16MB Quad-SPI 闪存,具有出厂编程的全球唯一标识符(兼容 48 位 EUI-48/64™)。
 - MicroSD 插槽。
- (3) 供电: 由 USB 或任何 7~15V 电源供电。
- (4) USB 和以太网:
 - 千兆以太网 PHY。

- USB-JTAG 编程电路。
 - USB-UART 桥。
 - USB OTG PHY(仅支持主机)。
- (5) 音频和视频:
- 具有脉冲密度调制(PDM)输出的驻极体麦克风。
 - 3.5mm 单声道音频输出插孔。
 - HDMI 接收端口(输入)。
 - HDMI 源端口(输出)。
- (6) 开关、按钮和 LED:
- 4 个按钮。
 - 2 个滑动开关。
 - 4 个 LED。
 - 2 个 RGB LED。
- (7) 扩展连接器:
- 两个标准 Pmod 端口。
 - 16 个 FPGA I/O 接口(与树莓派接口共享 8 个引脚)。
- (8) Arduino/chipKIT 屏蔽连接器:
- 49 个 FPGA I/O 口。
 - 6 个 XADC 的单端 0~3.3V 模拟输入。
 - 4 个 XADC 的差分 0~1.0V 模拟输入。

1.1.2 PYNQ Z2

PYNQ Z2 是台湾厂商 TUL 推出的支持 PYNQ 开源框架的第二代开发平台,其开发板如图 1-2 所示。

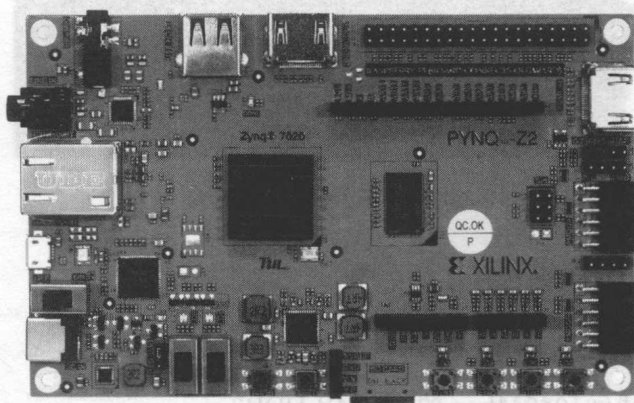


图 1-2 PYNQ Z2 开发板

PYNQ Z2 的具体配置如下:

- (1) 核心芯片: Zynq XC7Z020-1CLG400C。
- (2) 存储:
 - 带有 16 位总线@1050Mbps 的 512MB DDR3。

- 16MB Quad-SPI 闪存,具有出厂编程的全球唯一标识符(兼容 48 位 EUI-48/64)。
 - MicroSD 插槽。
- (3) 供电:由 USB 或 7~15V 电源供电。
- (4) USB 和以太网:
- 千兆以太网 PHY。
 - USB-JTAG 编程电路。
 - USB-UART 桥。
 - USB OTG PHY(仅支持主机)。
- (5) 音频和视频:
- 具有 24 位 DAC 且支持 I²S 协议的 3.5mm TRRS 插孔。
 - 3.5mm 线路输入插口。
 - HDMI 接收端口(输入)。
 - HDMI 源端口(输出)。
- (6) 开关、按钮和 LED:
- 4 个按钮。
 - 2 个滑动开关。
 - 4 个 LED。
 - 2 个 RGB LED。
- (7) 扩展连接器:
- 两个标准 Pmod 端口。
 - 16 个 FPGA I/O 接口(与树莓派接口共享 8 个引脚)。
- (8) Arduino 屏蔽连接器:
- 24 个 FPGA I/O。
 - 6 个 XADC 的单端 0~3.3V 模拟输入。
 - Raspberry Pi 连接器。
 - 28 个 FPGA I/O(与 Pmod A 接口共享 8 个)。
 - 低延时控制。

从上面的介绍可见,PYNQ Z1 和 PYNQ Z2 都采用了同一型号的核心芯片,在开发板的配置上有很多的相似之处,因此对于计算机组成原理实验来说具体使用哪一款板卡都是可以的。

它们的主要区别在于扩展接头和音频系统。PYNQ-Z2 使用 Raspberry Pi 接头取代了 PYNQ-Z1 上的 chipKIT 接头。PYNQ-Z1 具有带 PWM 输入的集成 MIC 和单声道 PDM 音频输出。PYNQ-Z2 具有完整的 ADI 音频编解码器,还带有耳机、麦克风和线路输出。

1.2 Zynq 7020 芯片

从 1.1 节中的介绍可发现,PYNQ Z1 和 PYNQ Z2 开发板使用的核心芯片都为 Zynq 7020 芯片,具体型号为 Zynq XC7Z020-1CLG400C。

Zynq 7020 是 Xilinx 公司推出的全可编程 SoC 芯片 (AP SoC, All Programmable System on Chips), 与其他 Zynq 7000 系列芯片具有相同的架构, 它主要由 PS (Processing System, 处理器) 和 PL (Programmable Logic, 可编程逻辑 FPGA) 两大部分组成, 所以 PYNQ 可被看作一个具有可编程逻辑模块的计算机。其内部架构如图 1-3 所示。

Zynq XC7Z020-1CLG400C 芯片内的模块如下:

- (1) 主频 650MHz 的双核 ARM Cortex-A9 处理器。
- (2) DDR3 内存控制器, 具有 8 个 DMA 通道和 4 个高性能 AXI4 从端口。
- (3) 高速外设控制器: 1GHz 以太网, USB 2.0, SDIO。
- (4) 低速外设控制器: SPI, UART, CAN, I²C。
- (5) 可通过 JTAG, Quad-SPI 闪存和 MicroSD 卡进行可执行流文件的烧写。
- (6) PL 部分采用的是 Artix-7 系列的可编程逻辑, 具体资源如下:
 - 13 300 个逻辑块, 每个逻辑块具有 4 个 6 输入 LUT 和 8 个触发器。
 - 630KB 的快速 Block RAM。

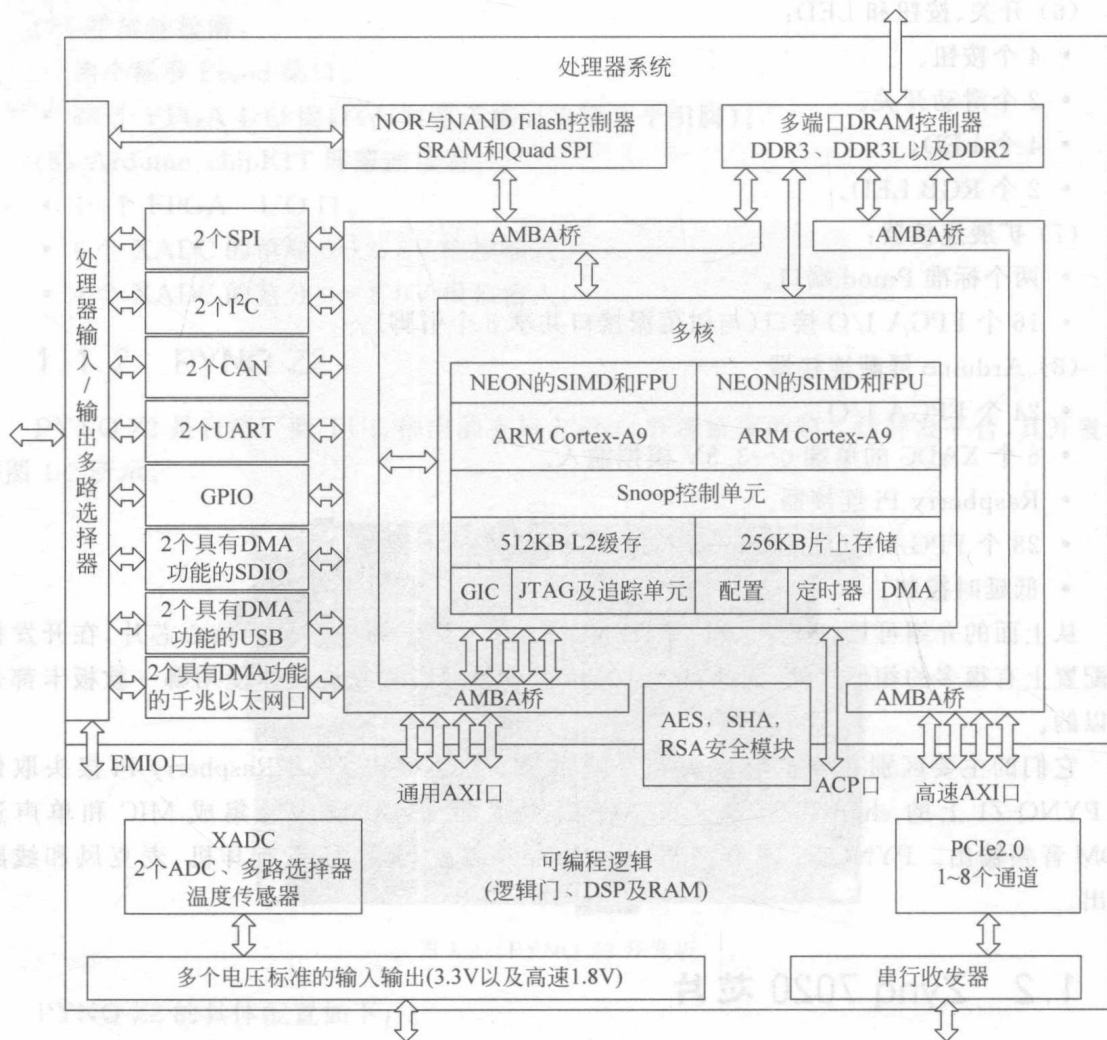


图 1-3 Zynq7000 芯片架构示意图

- 4 个时钟管理单元,每个有一个锁相环(PLL)和混合模式时钟管理器(MMCM)。
- 220 个 DSP。
- 片上模数转换器(XADC)。

1.3 PYNQ 框架

PYNQ 是一个开源框架,旨在使基于 Xilinx Zynq 的嵌入式系统设计更加容易。PYNQ 开源框架主要包含 PYNQ 硬件库(Overlay)以及 Overlay 的设计和调用方法。PYNQ 框架中实现了用来加载和使用 Overlay 的 Python 库,允许通过 PS 中运行的 Python 来控制 PL 中的 Overlay,Overlay 即在 PL 中运行的一个具体硬件模块实现。

因为 FPGA 设计通常需要具有硬件知识,PYNQ 的 Overlay 层由硬件设计人员创建,并提供相应的 PYNQ Python API。而软件开发人员就可以使用 Python 接口来调用和控制所需的硬件 Overlay 构成满足需求的软硬件协同计算系统。

Overlay 由三部分构成:

- 配置 PL 端 FPGA 的比特流文件(bit-stream)。
- 包含 IP 核信息的 Vivado Block Design TCL 文件。
- 读取 IP 核属性的 Python API。

PYNQ 框架的理念就是将软件和硬件开发隔离开,硬件人员仅关注特定 Overlay 的设计而无须关注整个应用,而软件或应用开发人员基于 Python 可快速开发适合不同需求的应用,并通过调用 Overlay 获得硬件加速来保证整个系统的运行效率。随着 Overlay 的不断积累和丰富,PYNQ 生态就会在应用的开发效率和系统的计算能效两方面都获得更好的表现。目前 PYNQ 官方支持的开发板有 Digilent 的 PYNQ Z1 开发板,TUL 的 PYNQ Z2 开发板及 Xilinx 自有的 ZCU104 开发板。此外,PYNQ 还可以支持安富利 Ultra96 及其他的一些第三方 Zynq 类的板卡。PYNQ 官网 <http://www.pynq.io/board.html> 提供了所支持的板卡的映像文件,同时介绍了如何为第三方板卡制作映像文件。

1.4 PYNQ 平台的使用

如前所述,PYNQ 平台就是一台由支持 Python 编程且具有可编程逻辑块的异构计算芯片构成的计算机。它可以放在现场使用,当然也可以部署在云端通过远程访问进行使用。关于 PYNQ 的实验环境的具体准备,详见第 2 章。

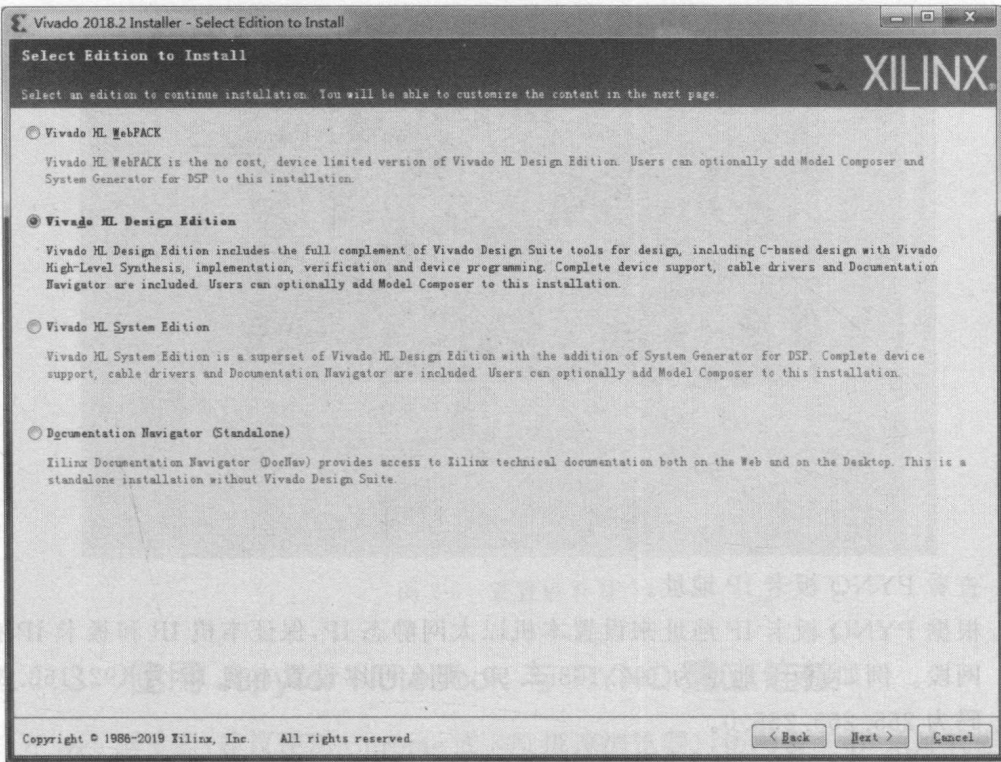


图 2-1 安装版本选择页面

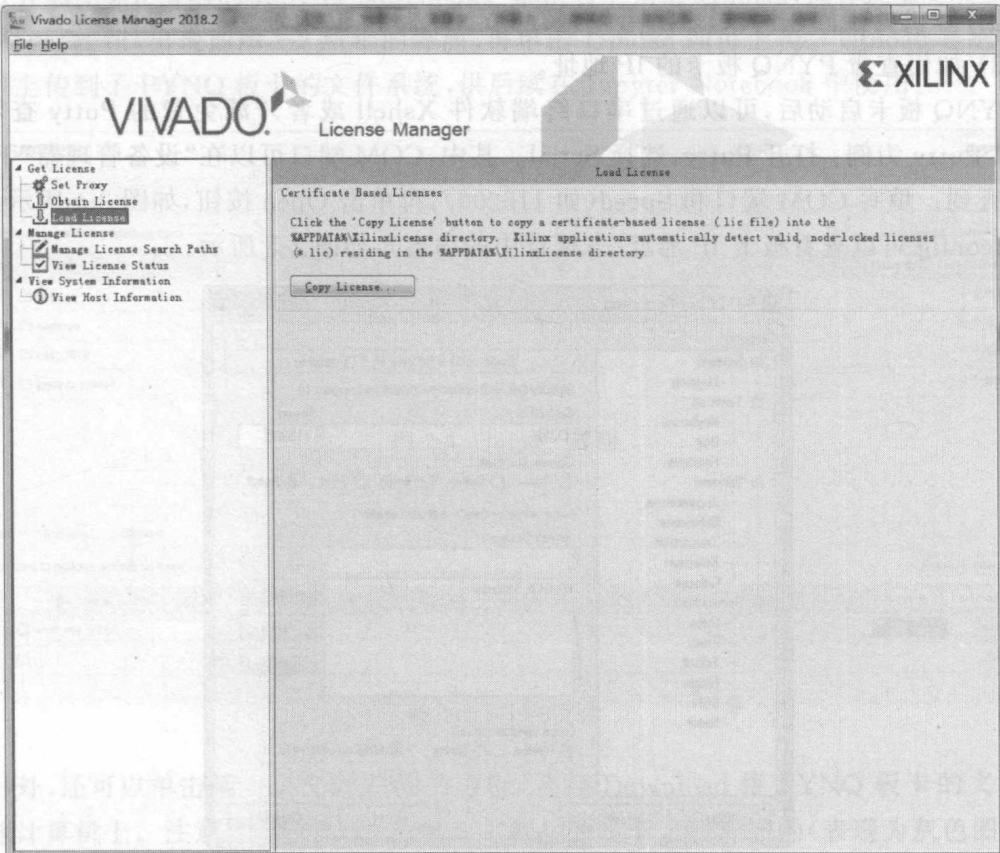


图 2-2 安装证书