

工程师自学笔记系列丛书

Intel FPGA大学计划经理推荐



SoC FPGA

嵌入式设计和开发教程

梅雪松 宋士权 陈云龙 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

工程师自学笔记系列丛书

Intel FPGA 大学计划经理推荐

SoC FPGA 嵌入式设计和开发教程

梅雪松 宋士权 陈云龙 编著



北京航空航天大学出版社

内 容 简 介

本书以 Intel Cyclone V SoC FPGA 系列器件为例,介绍 SoC FPGA 器件的架构特点、常用电路设计以及软硬件开发流程和开发技巧。内容按照开发一个基于 SoC FPGA 的应用系统所需掌握的最基本的知识路线展开,首先从基本的 Linux 系统操作到一个最基础的应用系统框架分析,详细讲解应用系统的构建、BSP 文件的生成、启动引导文件的更新、Ubuntu 虚拟机安装配置、Linux 内核配置与编译。然后介绍如何在嵌入式 Linux 系统环境下,使用虚拟地址映射的方式编写相应的应用程序来实现该应用系统中各个功能 IP 的编程控制和调试。最后以两个实际的例子展示如何通过 HPS 和 FPGA 的片上通信桥实现软硬件联合开发的过程,包括 FPGA 侧逻辑开发、IP 总线封装、Linux 驱动程序的编写编译、Linux 应用程序的编写与运行等。

本书既可作为工程类应用、电子信息类专业本科生以及相关专业专科生的嵌入式系统基础类课程的教材,也可作为 SoC FPGA 自学人员以及从事 SoC FPGA 开发的工程技术人员的培训教材和参考用书。

图书在版编目(CIP)数据

SoC FPGA 嵌入式设计和开发教程 / 梅雪松, 宋士权, 陈云龙编著. -- 北京: 北京航空航天大学出版社, 2018.12

ISBN 978-7-5124-2239-1

I. ①S… II. ①梅… ②宋… ③陈… III. ①集成电路—芯片—设计—高等学校—教材 IV. ①TN402

中国版本图书馆 CIP 数据核字(2018)第 265090 号

版权所有,侵权必究。

SoC FPGA 嵌入式设计和开发教程

梅雪松 宋士权 陈云龙 编著

责任编辑 杨 昕

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:copyrights@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:20.75 字数:442 千字

2019 年 3 月第 1 版 2019 年 3 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2239-1 定价:69.00 元

他 序

随着半导体工艺的提升,芯片厂商将更多的功能集成到单一的半导体芯片上,芯片集成度的提高,随之带来的是应用设计复杂度的提高。Intel 公司在其 FPGA 芯片上集成了 ARM Cortex 处理器,从而形成一颗基于 FPGA 的 SoC 芯片,这是一个典型的可配置的单芯片系统。

目前,一颗主流的 FPGA 芯片除了逻辑单元外,还集成了嵌入式存储器块、锁相环、DSP 块,甚至高速收发器电路,并将 ARM Core 集成到 FPGA 芯片上,在带来功能高度集成的同时,再一次增加了应用设计的复杂度。

目前,基于 SoC FPGA 嵌入式系统设计的参考书和教材还很少,而小梅哥的这本书从最基本的概念讲起,由浅入深,再配合大量的截图,一步一步地介绍了整个设计的流程。该书内容涵盖了基于 SoC FPGA 的硬件系统搭建、Linux 操作系统的配置,以及软件的设计与调试方法等,手把手地将一个初学者带进 SoC FPGA 嵌入式系统的设计大门,非常值得推荐。

希望作者在本书的基础上,再接再厉,不断地写出更好的参考书,也希望广大读者对本书给予大力支持!

Intel FPGA 大学计划经理

袁亚东

2018 年 9 月于上海

目 序

Intel 公司的 SoC 器件是集成了 ARM 公司双核/四核 Cortex-A9/A53 的 FPGA,包括集成了双核 Cortex-A9、速度从 925 MHz(Cyclone V SoC)到 1.5 GHz (Arria 10 SoC)的低端和中端 SoC,以及高端的集成四核 Cortex-A53、速度是 1.5 GHz 的 Stratix 10 SoC,性能远超任何软核 CPU。例如运行在 Intel 自己 FPGA 上的 NIOS II 软 CPU,速度仅为 100 MHz(Cyclone 各系列)和 200 MHz(Stratix 各系列)。另外,硬核与软核(包括软 CPU)相比还有一个优势,就是硬核不需要 FPGA 开发软件进行编译和布局布线,因此不可能出现任何有关时序的问题,而时序问题又是 FPGA 设计中比较容易出问题的地方。

Intel SoC 器件的 ARM 部分开发软件采用了 ARM 公司著名的 DS-5 软件,直接与 Intel 的 FPGA 部分开发软件 Quartus 无缝连接,无论是 Quartus 还是 DS-5 都可以使用 USB 下载电缆仿真和下载调试。用户可以完全免费地使用 DS-5 软件开发 Android 和 Linux 等操作系统下的应用层软件,大大降低了用户的开发成本。作为 ARM CPU 的开发者,应该相信 ARM 公司是最了解 ARM CPU 的,DS-5 作为 ARM 公司自己开发的用来开发所有 ARM CPU 器件的开发软件,不仅集成了业内编译效果最好、产生二进制代码最紧凑、执行效率最高、优化效果最好的 AMCC 编译器,而且集成开发环境(IDE)的功能也最为强大、最为全面。可以说,DS-5 软件是开发 ARM CPU 的最佳选择。

Cortex-A 系列集成了硬件浮点处理单元 NEON 和 FPU,性能可以媲美各个 DSP 厂家的中高端浮点和定点 DSP,例如著名的 TMS320C6455、67XX 以及虎鲨系列,其仅比 TMS320C68XX 系列稍差一些。而集成了 Cortex-A 系列 ARM 核的 SoC FPGA 与单片的 Cortex-A 产品相比,还有集成了 FPGA 部分的优势,也就是说,除了可以使用 FPGA 部分的 I/O 来扩充 ARM 系统的 I/O 数量外,还可以把一些在 CPU 上顺序执行的、无法实现更高性能的算法放在 FPGA 部分来并行执行,这极大地提高了产品的性能和灵活性。

与市场上类似的集成了 Cortex-A 的 SoC 相比,Intel 公司的 SoC 使用了一个非



常有实用价值的电源技术——FPGA 部分和 ARM 部分的供电电源是互相独立的。其中任何一方即使不启动甚至断电,另一方依旧可以正常工作。即使 ARM 部分不启动,FPGA 部分依旧可以作为一个完全独立的系统,通过 JTAG 或者 AS、PS 方式加载配置文件并运行,即如果用户用不到 HPS 部分,那么在使用 Intel SoC FPGA 时,可以完全将其当成一个独立的 FPGA 来使用。另外,两者可以通过互相给对方停电来降低功耗,也就是说,给对方的电源输出控制引脚发出一个控制电平,甚至可在对方电源回路中增加一个功率 MOS 管,通过控制 MOS 管的通断来实现完全的电源关断。这个技术在低功耗产品上非常有价值。ARM 部分还可以在低频下运行,例如 Cyclone V SoC 就可以稳定可靠地运行在 400 MHz,进一步降低功耗。这在一些对性能要求不高,而功耗要求苛刻的场合拥有很大的实用价值。

Intel SoC 器件支持 ARM 公司 Cortex-A 系列的 AMP 功能,即双核可以同时运行 2 个相同或者不同的操作系统,也可以一个核运行操作系统,另外一个核运行裸跑模式。

在提高 SoC 整体性能方面,Intel 公司的 SoC 在 FPGA 和 SoC 部分的互联桥上,使用了高达 128 位宽度的数据总线,而且这个桥的宽度是可以配置的,根据实际应用需求可以配置为 32 位、64 位、128 位位宽。相较于一些固定 64 位总线宽度的产品,其在 FPGA 和 ARM 的片上通信带宽方面,性能更加优异,使用也更加灵活。

在提高用户的整机产品可靠性方面,Intel 公司的 SoC 主要有两方面优势:

第一,电源方面没有上电和关电顺序要求,可以大大降低对外围电源设计的复杂度,降低成本,提高产品的可靠性。

第二,通过增加 ECC(Error Correcting Code 错误检查和纠正)功能,大幅提高用户整机产品的可靠性。在 L2 高速缓存、片内 RAM、Quad SPI 控制器、NAND 控制器、SD/MMC/SDIO 控制器、DMA 控制器、10M/100M/1G 以太网控制器、USB 2.0 OTG 控制器上全面使用 ECC,对于工业控制类产品和一些民用产品,特别是那些运行在复杂电磁环境、恶劣气候环境、有高可靠性要求的产品上,例如发电厂、变电站、武器装备、航空航天、安防、高铁、精密数控机床、医疗器械等要求高可靠性的产品上有非常大的实用价值。

在小体积或者便携式产品中,SoC 与 FPGA+ARM 或者 FPGA+DSP 相比具有 PCB 面积更小、功耗和价格更低的优势。

总之,Intel 公司的 SoC 产品是一类集成度高、性能优异、稳定可靠、成本低、功耗低的非常优秀的产品。因为各种原因,国内目前详细介绍 Intel SoC 的书籍还比较少,由网友小梅哥等编写,骏龙公司支持的本书通过大量详细的资料和例子带领读者熟悉 SoC 架构和开发流程,助力用户实现从 0 到 1 的跨越。

骏龙科技

宋士权

2018 年 9 月于西安

前言

随着集成电路生产工艺的不断进步,芯片中晶体管的数量也在不断增加。GAL、PLD、CPLD、FPGA 等一系列可编程逻辑数字集成电路相继诞生。新型架构芯片的出现,不仅得益于集成电路设计和制造工艺的进步,而且更离不开实际的应用需求。现代数字系统大多朝小型化、集成化方向发展,而作为一种通用的可编程逻辑器件,FPGA 以其灵活的现场可编程特性、强大的并行处理能力,在众多高性能数字系统中都有应用。

鉴于很多高性能的数字系统都离不开 FPGA 器件,为了进一步降低硬件系统的复杂度,各大 FPGA 厂家都推出了基于各自 FPGA 的软核 CPU 方案。通过该方案,设计者能够将原本需要通过外置 CPU 实现的功能转移到 FPGA 芯片上,使用通用逻辑搭建的软核 CPU 来实现,在保证系统功能和性能的前提下,简化硬件系统设计,提高系统集成度,降低维护难度。但是受限于 FPGA 中通用逻辑的可运行最大频率,各种软核 CPU 的运行频率都较低,一般不超过 200 MHz,再加上架构的原因,各种软核 CPU 的软件生态都远远赶不上各种常用的单片机。因此,基于软核处理器的应用开发受到了较大的限制。

在此情形下,众多的 FPGA 厂商又相继推出了集成高性能 ARM 硬核处理器和通用可编程逻辑的新型架构芯片。这其中就包括以 Xilinx 公司的 Zynq 系列为代表的全可编程芯片以及 Intel 公司的可编程片上系统芯片(SoC FPGA)。

无论是 Xilinx 公司的 Zynq 系列芯片,还是 Intel 公司的 SoC FPGA 芯片,两者的架构都是相似的,都是在同一个晶片上将高性能的 ARM Cortex-A 系列处理器与 FPGA 有机结合,并辅以各种常见的外设,实现完整的系统。依赖于 ARM 处理器强大的开发工具链和软件生态环境,使用这一类器件进行系统设计、开发和调试的时间将被大大缩短。

得益于 Intel 大学计划提供的众多学习机会和学习资料,笔者从 2014 年年底开始接触 Intel 公司的 SoC FPGA,并开始了断断续续的学习。但是由于在这之前,作者并没有任何基于 Linux 系统开发的基础,因此学习的过程非常艰辛,学习的资料主



要来自网络博文和一些讲解 Linux 开发的书籍。幸好在这个学习的过程中得到了很多朋友的无私帮助,尤其是骏龙科技的工程师,也是本书的第二作者宋士权,他多次为我提供最新的一手学习资料,让我最终坚持了下来。

在决定编写本书之前,市面上还很难找到系统讲解 Intel SoC FPGA 开发的书籍,作者作为一个草根创业者,既对该器件的开发抱有极大的兴趣,同时又受限于个人的知识积累,因为学习过程本身就是一个不断试错的过程,所以本书从某种意义上来说,可以算做作者学习 SoC FPGA 开发的学习笔记。

由于作者能力有限,虽然在写作过程中投入了大量的精力和时间,但错误与不妥之处在所难免,读者在阅读本书时如果发现任何疏漏,都可及时反馈给我们,以便我们及时更正。读者可以通过本书配套的网站 www.corecourse.cn 留言,反馈在阅读本书过程中所发现的问题,或者在学习过程中遇到的疑惑;同时,也可以在该网站上以本书名作为关键词,搜索与本书相关的软硬件配套资源。

作 者

2018 年 12 月



第 1 章 SoC FPGA 软硬件系统开发概述	1
1.1 Intel SoC FPGA 系列	1
1.1.1 Cyclone V SoC FPGA	2
1.1.2 Arria V SoC FPGA	2
1.1.3 Arria 10 SoC FPGA	2
1.1.4 Stratix 10 SoC FPGA	3
1.1.5 SoC FPGA 应用领域与前景	3
1.2 Intel Cyclone V SoC FPGA 介绍	4
1.2.1 什么是 SoC FPGA	4
1.2.2 SOPC	4
1.2.3 SoC FPGA 与 SOPC 之间的差异	5
1.2.4 SoC FPGA 架构的优势	6
1.3 Cyclone V SoC FPGA 器件硬件设计概述	9
1.3.1 FPGA I/O 和时钟	10
1.3.2 SoC FPGA JTAG 电路设计	12
1.4 AC501-SoC 开发板介绍	13
1.4.1 布局及组件	13
1.4.2 轻触按键	14
1.4.3 用户 LED	15
1.4.4 时钟输入	16
1.4.5 GPIO 接口	17
1.4.6 DDR3 SDRAM	18
1.4.7 通用显示扩展接口	19
1.4.8 USB 转 UART	21



1.4.9 以太网收发器	21
1.5 本章小结	23
第2章 SoC FPGA 开发板的使用	24
2.1 安装 SoC FPGA 开发工具	24
2.2 SoC FPGA 的配置数据烧写与固化	24
2.2.1 SoC FPGA 启动配置方式介绍	24
2.2.2 sof 文件的烧写方式	26
2.2.3 jic 文件的生成和烧写	28
2.3 在 SoC FPGA 上运行 Linux 操作系统	32
2.3.1 SoC FPGA 中的 HPS 启动流程介绍	32
2.3.2 HPS 启动方式介绍	33
2.3.3 制作启动镜像 SD 卡	34
2.3.4 准备硬件板卡	35
2.3.5 开机测试	39
2.4 开发板 Linux 系统常用操作	40
2.4.1 查看目录	40
2.4.2 设置和修改用户密码	40
2.4.3 查看和编辑文件	41
2.4.4 设置 IP 地址	43
2.4.5 挂载 SD 卡的 FAT32 分区	45
2.4.6 挂载 U 盘	46
2.4.7 文件操作	47
2.4.8 目录操作	48
2.4.9 停止某个进程	49
2.4.10 重启和关机	50
2.5 本章小结	50
第3章 SoC FPGA 开发概述	51
3.1 SoC FPGA 开发流程	51
3.1.1 硬件开发	51
3.1.2 软件开发	53
3.2 AC501-SoC FPGA 开发板的黄金参考设计说明	53
3.2.1 GHRD	53
3.2.2 打开和查看 GHRD	54
3.2.3 组件参数配置详解	57

3.3 本章小结	62
第4章 手把手修改 GHRD 系统	63
4.1 修改 GHRD 工程	63
4.1.1 打开 GHRD 工程	63
4.1.2 添加 UART IP	64
4.1.3 关于 HPS 与 FPGA 数据交互	64
4.1.4 连接 UART IP 信号端口	65
4.1.5 分配组件基地址	67
4.1.6 生成 Qsys 系统的 HDL 文件	68
4.1.7 添加 uart_1 的端口到 Quartus 工程中	69
4.1.8 分配 FPGA 引脚	71
4.1.9 生成配置数据二进制文件	72
4.2 制作 Preloader Image	72
4.2.1 打开 SoC EDS 工具	73
4.2.2 生成 bsp 文件	74
4.2.3 编译 Preloader 和 U-Boot	77
4.2.4 更新 Preloader 和 U-Boot	79
4.2.5 Win 10 系统下更新失败问题	80
4.2.6 使用新的 U-Boot 启动 SoC	81
4.3 制作设备树	82
4.3.1 设备树制作流程	82
4.3.2 准备所需文件	82
4.3.3 生成 .dts 文件	83
4.3.4 生成 .dtb 文件	84
4.4 运行修改后的工程	85
4.5 本章小结	87
第5章 使用 DS-5 编写和调试 SoC 的 Linux 应用程序	88
5.1 启动 DS-5	88
5.2 创建 C 工程	91
5.3 编译工程	94
5.4 建立 SSH 远程连接	95
5.4.1 创建远程连接	95
5.4.2 复制文件到目标板	101
5.4.3 运行应用程序	102



5.5 远程调试	103
5.5.1 GDB 设置	103
5.5.2 GDB 连接和调试	106
5.6 使用 WinSCP 实现多系统传输文件	108
5.6.1 为什么要使用 WinSCP	108
5.6.2 安装 WinSCP	109
5.6.3 建立远程主机连接	109
5.6.4 新建远程连接	112
5.6.5 调用 PuTTY 终端	112
5.7 本章小结	113
第 6 章 基于虚拟地址映射的 Linux 硬件编程	114
6.1 什么是虚拟地址映射	114
6.2 虚拟地址映射的实现	115
6.3 基于虚拟地址映射的 PIO 编程应用	117
6.3.1 PIO 外设的虚拟地址映射	117
6.3.2 在 DS-5 中建立 PIO 应用工程	118
6.3.3 添加和包含 HPS 库文件	119
6.3.4 添加 FPGA 侧外设硬件信息	121
6.3.5 PIO IP 核介绍	124
6.3.6 PIO 核寄存器映射	125
6.3.7 PIO IP 核应用实例	128
6.3.8 合理的程序退出机制	131
6.3.9 关于按键消抖	133
6.4 基于虚拟地址映射的 UART 编程应用	134
6.4.1 UART 核介绍	134
6.4.2 UART 寄存器映射	134
6.4.3 UART IP 核应用实例	136
6.4.4 UART IP 核板级调试	144
6.4.5 小 结	145
6.5 基于虚拟地址映射的 I ² C 编程应用	145
6.5.1 OpenCores I ² C IP 简介	146
6.5.2 OpenCores I ² C IP 寄存器映射	146
6.5.3 I ² C IP 核应用实例	149
6.5.4 小 结	161
6.6 本章小结	161

第 7 章 基于 Linux 应用程序的 HPS 配置 FPGA	162
7.1 制作 Quartus 工程	163
7.2 生成 rbf 格式配置数据	163
7.3 编译 Linux 配置 FPGA 应用程序	165
7.4 在系统重配置 FPGA 实验	166
7.5 本章小结	168
第 8 章 编译嵌入式 Linux 系统内核	169
8.1 安装 VMware	170
8.2 安装 Ubuntu 系统	171
8.2.1 使用现成的 Ubuntu 系统镜像	171
8.2.2 安装全新的 Ubuntu 系统	175
8.3 下载 Linux 系统源码	182
8.4 设置交叉编译环境	185
8.5 配置和编译内核	189
8.5.1 快速配置内核	189
8.5.2 保存内核配置文件	195
8.5.3 编译内核	195
8.5.4 使用内核启动开发板	197
8.6 本章小结	199
第 9 章 Linux 设备树的原理与应用实例	200
9.1 什么是设备树	200
9.2 设备树的基本格式	201
9.3 设备树加载设备驱动原理	206
9.4 编写 I ² C 控制器设备节点	208
9.5 加载 OC_I2C 驱动	211
9.6 使用 RTC	212
9.7 使用 EEPROM	216
9.8 编写 SPI 控制器设备节点	217
9.9 本章小结	219
第 10 章 基于 Linux 标准文件 I/O 的设备读/写	220
10.1 什么是文件 I/O	220
10.2 基于文件 I/O 操作的一般方法	220



10.2.1	文件描述符	220
10.2.2	打开设备(open)	221
10.2.3	向设备写入数据(write)	221
10.2.4	读取设备数据(read)	222
10.2.5	杂项操作(ioctl)	222
10.2.6	关闭设备(close)	223
10.2.7	其他操作	223
10.3	使用文件 I/O 实现 I ² C 编程	223
10.4	本章小结	226
第 11 章 FPGA 与 HPS 高速数据交互应用		227
11.1	FPGA 与 HPS 通信介绍	227
11.1.1	H2F_LW_AXI_Master 桥	229
11.1.2	H2F_AXI_Master 桥	229
11.1.3	F2H_AXI_Slave 桥	230
11.2	AXI 与 Avalon-MM 总线的互联	230
11.3	Avalon-MM 总线	230
11.4	Avalon-MM Slave 接口	232
11.5	基本 Avalon-MM Slave IP 设计框架	234
11.5.1	端口定义	234
11.5.2	寄存器和线网定义	235
11.5.3	Avalon 总线对寄存器的读/写	235
11.5.4	用户逻辑使用寄存器	236
11.6	PWM 控制器设计	237
11.6.1	PWM IP 核端口设计	238
11.6.2	PWM IP 核寄存器定义	239
11.6.3	读/写 PWM 寄存器	239
11.6.4	Platform Designer 中封装 PWM IP	241
11.7	Avalon-MM Master 接口	255
11.7.1	常见的通用 Avalon-MM Master 主机	256
11.7.2	DMA Controller	256
11.7.3	Scatter-Gather DMA Controller	256
11.7.4	Modular Scatter-Gather DMA	258
11.7.5	Avalon-MM Master 模板	260
11.8	高速数据采集系统	263
11.8.1	安装 Avalon-MM Master 模板	263

11.8.2	完善 Qsys 系统	265
11.8.3	修改 Quartus 中的 Qsys 例化	269
11.8.4	测试逻辑设计	271
11.9	本章小结	275
第 12 章	Linux 驱动编写与编译	276
12.1	基本字符型设备驱动	276
12.1.1	字符型设备驱动框架	277
12.1.2	PWM 控制器驱动的完整源码	286
12.1.3	驱动编译 Makefile	291
12.1.4	Ubuntu 下编译设备驱动	292
12.1.5	字符型设备驱动验证	293
12.2	基于 DMA 的字符型设备驱动	297
12.2.1	Avalon-MM Master Write 驱动	298
12.2.2	Avalon-MM Master Write 测试	304
12.3	本章小结	311
附录 A	外设地址映射	312
附录 B	HPS GPIO 映射	314
参考文献		315

第 1 章

SoC FPGA 软硬件系统开发概述

随着信息技术的高速发展,各行各业正趋向于通过资源整合和并购的方式来获得更强、更稳固的竞争力,芯片架构亦是如此。单“芯”SoC 方案(System on Chip)拥有的低功耗、低成本、低布线面积以及高整合、高性能、高带宽(内部互联)的优势正推动其引领电子系统设计的潮流。

在传统的芯片架构中,处理器、DSP、FPGA 往往都各自独立、互不相干。当一个系统需要用到处理器、DSP、FPGA 中的多个元件时,采用板级集成的方式,在一块电路板上设计复杂的电路,将它们通过 PCB 板走线连接在一起。以此种方式设计的系统不仅设计生产成本高,而且受限于 PCB 走线和 I/O 引脚性能的影响,无法实现高带宽的数据通信。设计师们都期待着有这样一种芯片,能够同时拥有处理器、FPGA、DSP 的特点,各个架构的优势强强联合,以提升电子系统设计的便利性。

为顺应时代的需求,亦如当年麻雀虽小、功能俱全的“单”片机出世,各大传统 FPGA 厂家都顺势推出了带有嵌入式硬核处理器的 SoC FPGA。如 Intel FPGA 部门基于不同应用推出的带有 Cortex-A9、Cortex-A53、至强 CPU 等一系列涵盖低、中、高端的 SoC FPGA 器件,以及 Xilinx 推出的带有 Cortex-A9、Cortex-A53 处理器的 Zynq 系列 FPGA。

集成处理器和 FPGA 器件具有划时代意义。这样 ARM 和 FPGA 的优势共存一体,即 ARM 的顺序控制、丰富外设、开源驱动,与 FPGA 的并行运算、高速接口、灵活定制相得益彰。再加上其内部多条高速桥接总线,使其数据交互链路畅通无阻。

无论是 Xilinx 公司的 Zynq 全可编程系列 FPGA,还是 Intel 公司的 SoC FPGA,其基本架构都是在同一个硅片上集成 FPGA 和 CPU,并通过高速、高带宽的互联架构连接起来。它们本质相同,架构和性能也都非常相似。熟悉一种器件的使用和开发思路,掌握其开发流程,就可以快速地过渡到另一种器件上。因此,本书以 Intel 公司的 Cyclone V SoC FPGA 器件为例,讲解这类新型异构芯片的开发方法。

1.1 Intel SoC FPGA 系列

针对不同的应用领域,Intel 公司的 PSG 部门设计开发了各种逻辑资源以及性能优异的 SoC FPGA 器件,如图 1.1.1 所示。

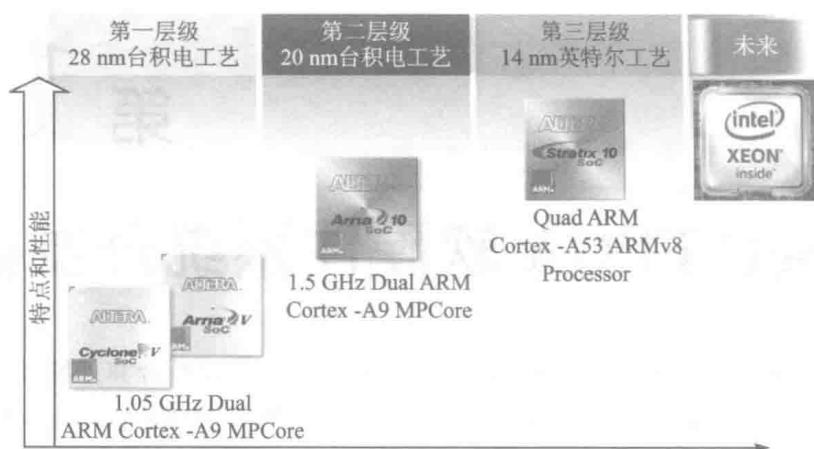


图 1.1.1 Intel SoC FPGA 器件图谱

1.1.1 Cyclone V SoC FPGA

Cyclone V SoC FPGA 基于台积电 28 nm 工艺,系统成本和功耗都非常低,其性能和成本优势适合大批量应用。FPGA 与前几代相比,其总功耗降低了 40%,具有高效的逻辑集成功能,提供可选的集成收发器,并且支持精度可调的 DSP 模块,数字信号处理性能高达 150 GMACS 和 100 GFLOPS。Cyclone V SoC FPGA 提供了 3 大类可选类型,如下:

- ① 具有基于 ARM 的 HPS 的 Cyclone V SE SoC FPGA。
- ② 具有基于 ARM 的 HPS 和 3.125 Gbps 收发器的 Cyclone V SX SoC FPGA。
- ③ 具有基于 ARM 的 HPS 和 5 Gbps 收发器的 Cyclone V ST SoC FPGA。

1.1.2 Arria V SoC FPGA

Arria V SoC FPGA 同样基于台积电 28 nm 芯片生产工艺,对于远程射频前端、LTE 基站以及多功能打印机等中端应用,在成本、功耗和性能上达到了均衡。高速的 FPGA 架构、快速 I/O 以及高速收发器数据速率等特性进一步提高了系统性能,具有丰富 DSP 块的 Arria V FPGA 架构性能达到 1 600 GMACS 和 300 GFLOPS。Arria V SoC FPGA 提供 2 大类可选类型,如下:

- ① 具有基于 ARM 的 HPS 和 6.553 6 Gbps 收发器的 Arria V SX SoC FPGA。
- ② 具有基于 ARM 的 HPS 和 10.312 5 Gbps 收发器的 Arria V ST SoC FPGA。

1.1.3 Arria 10 SoC FPGA

Arria 10 SoC FPGA 采用了台积电 20 nm 的芯片生产工艺。作为高端 SoC FPGA 器件,其性能本身比 Stratix V 高出 15%,而功耗却比 Arria 系列的前代产品 Arria V 节省了 40%。同时,Arria 10 SoC FPGA 提供了更加丰富的高性能 IP 支持,