让一切变得更简单！™

英特尔®专版第 2 版

现场可编程逻辑门阵列（FPGA）

FOR DUMMIES®

学习内容：

- FPGA 的工作原理
- FPGA、ASSP、ASIC 之间的区别
- 将 FPGA 用作系统功能块

提供者：

intel

安德鲁·摩尔
以及罗恩·威尔森

现场可编程门阵列 (FPGA)

FOR DUMMIES®

英特尔®专版第 2 版

作者：安德鲁·摩尔

以及：罗恩·威尔森总编
英特尔®可编程解决方案事业部

WILEY

现场可编程逻辑门阵列 FPGA For Dummies®，英特尔®专版第 2 版
出版商
约翰 · 威利父子公司
美国新泽西州霍布肯市
河畔路 111 号，邮编：07030-5774
www.wiley.com

ISBN: 978-1-119-43732-1 (pbk); ISBN: 978-1-119-43735-2 (ebk)

美国制造

10 9 8 7 6 5 4 3 2 1

鸣谢

为本书上市做出贡献的部分人员如下：

项目编辑：詹妮弗 · 宾汉姆
采购编辑：卡蒂 · 摩尔
编辑经理：列夫 · 门格尔
业务发展代表：卡伦 · 哈坦
项目协调：玛格什 · 伊兰格凡
特别感谢：英特尔公司多人提供的帮助

目录

引言

现场可编程逻辑门阵列（FPGA）是一种让设计师能够在现场对定制数字逻辑进行编程的集成电路。FPGA 早在 20 世纪 80 年代就已出现，当时只是为了让所有设计团队能够制作定制逻辑。早期，如果在设计中使用 FPGA，则意味着即便只是执行一些简单功能，您也必须做大量编程工作。因此，设计师大都避免使用 FPGA。如果您在大学毕业之后，就再没有接触过，那么您可能需要再了解一番。

FPGA 已经从有用但简陋的接口设备，发展成为系统级集成电路，拥有自己的微处理器、存储块和接口。它将是明日之星。

现在，您可获得低廉开发套装、下载免费工具，开始探索属于您自己的世界，机不可失。本书将帮您了解 FPGA 的实际用途。

关于本书

如果您是系统设计师、有一定经验的工程师，或者自从工学院毕业后再没接触过 FPGA，那么本书适合您使用。

本书与 Altera 公司合作编写，该公司现已并入英特尔公司。

书中符号

在书中，作者偶尔会使用特殊符号标记重要信息。虽然您不会看到典型的笑脸等表情符号，但您绝对想要驻足观看！下面是您会看到的内容。

对于这个图标指示的信息，您会想放到硬盘、U 盘，或者保存在备忘录里，以备不时之需！

谁知道呢？您也可能只想记住个大概，然后在朋友面前显摆一番！

开个玩笑，我可不想让您把这种知识用来娱乐服务生和酒保！您应该停下来看看，因为这些知识虽然零散，但却可在将来减少麻烦。

其他内容

尽管本书信息量很大，但在短短 48 页里，我只能讲这么多！所以，如果您想更多了解 FPGA，只需访问这个网站：`www.intel.com/altera`。在该网站，您可更多了解英特尔公司的 FPGA 产品。您还可观看视频、参加网课，以及下载演示文件、阅读数据资料和白皮书等等！

第 1 章

大家的 FPGA

本章提要

- 介绍 FPGA
- 发现 FPGA 的功能
- 认识 FPGA 与 ASIC 的不同

热烈欢迎！如果您正要开始阅读本章节，不难想象，您可能是一名工程师，先前可能了解过现场可编程逻辑门阵列（FPGA），现在还想再多知道一些。或许，您想知道怎样在设计中使用 FPGA。在本章，我将向您介绍 FPGA，以及 FPGA 能够解决哪些问题。我还将说明 FPGA 的工作原理，探讨设计折衷以及 FPGA 的真实运算方式。

与其他硬件搭建方式相比，FPGA 有两大优势。首先，FPGA 让您能够只搭建您需要的硬件，您不再需要使用其他竞争对手都在用的专用应用标准产品（ASSP），也不再需要耗时、昂贵的专用应用集成电路（ASIC）并承担相关风险。

另处，同样重要的是，FPGA 能够进行定制，这意味着您在 FPGA 中的运算往往比在 ASSP 的微处理器内核里更加简单、快速和节能。

那么，到底为什么需要 FPGA？

很高兴您问到这个问题！ FPGA 是一种半导体设备，这种设备可以在制造完成之后再定义功能。即使产品已经现场安装完成，FPGA 也能让您编写产品特征和功能，根据新的标准进行改造，以及针对特定应用程序重新配置硬件，这也是为什么它被称为“现场可编程”门阵列（FPGA）的原因。门阵列属于二维逻辑门阵列。如果有足够多的门阵列，您就可以通过简单计算的相加，做一些有意义的事情。

换句话说，FPGA 为您提供了设计灵活性，它在不引入大量成本，并且不带来设计延期风险的条件下，让您能够修改系统组件的功能。

打个简单的比方，它就像汽车的后视摄像头。如果在图像传感器“看到”图像之后，您的摄像头系统需要 250 毫秒才能让图像帧实际显示到显示屏上，但是修订后的政府法规要求延时不得超过 100 毫秒，那么您可以在 FPGA 中调整图像信号处理的流水线，来达到新的延时要求。如果是基于微处理器的系统，则几乎不可能做到。在这个过程中，因为不必重新设计系统组件，也不必购买全新处理器，因此能够带来很大好处。

早期的 FPGA 电路体积庞大，无法安装到单个芯片上。设计者能做的就是搭建一个 FPGA 接口，客户能够对该接口重新编程和改造（例如，将键盘输入接口修改成触屏输入接口）。不过，不久设计师就认识到，他们能够利用 FPGA 搭建整个子系统，这意味着他们不再受限于仅使用 ASIC 实施子系统。由于电路组件的体积越来越小，设计师能够将越来越多的设备放置到同一芯片上——从而可以实现更加复杂的功能和更高速度的运算，这反过来又提高了计算速度，降低了功耗。

现代 FPGA 综合使用可配置的静态随机存取存储器（SRAM）、高速输入 / 输出管脚（I/O）、逻辑块和布线资源。具体而言，FPGA 包括可编程逻辑元件——顺理成章地，这些元件被称为“逻辑元件”；还包括由可重新配置的内部连线组成的分层结构，内部连线使得逻辑元件相互之间能够实现物理连接。您能够通过配置，让逻辑元件完成复杂的功能，或者简单执行逻辑门的工作，例如：“AND”和“OR”。多数 FPGA 还包括存储块（关于本话题的更多信息，请参见“FPGA 的基本组成部分”一节）。

ASIC 和 ASSP

*专用应用集成电路（ASIC）*是一种由电子组件组成的集成电路，例如：晶体管、电容器、电阻器等，这些组件被植入到晶元上；晶元由硅或其他半导体材料组成，并可按照特定用途定制。话音记录器和高频比特币矿机都是 ASIC。多年来，集成电路的组件体积已经缩小，这意味着在相同空间的情况下，可制成复杂度更高的电路。由于组件体积的缩小，现在有些 ASIC 足够容纳多个微处理器以及其他复杂的子系统。

*专用应用标准产品（ASSP）*则是专门针对特定应用市场的集成电路，可以销售给不止一个用户（因此才被称为“标准”），而 ASIC 的设计和销售只针对单个客户。许多智能手机和平板电脑在核心位置安装的微控制器和系统芯片都是 ASSP。

ASIC 和 ASSP 都是针对特定功能设计的。由于配置的严格控制，ASIC 和 ASSP 体积非常紧凑、成本低廉、速度快、功耗低，这些都是电子设计非常需要的特征。由于它们的功能在制造时已是硬连接，因此即便只是修改一小部分电路的功能，也并不容易。事实上，由于它们的电路永久性地植入到硅晶元上，因此您根本无法把它拆开，再换成其他线路。如果您需要修改已设计好的内容，只能放弃整个芯片，然后从头再来。

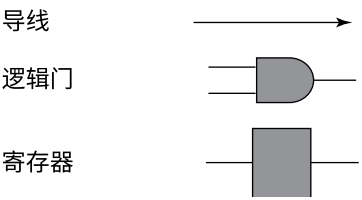
今天，“硬 IP”组件能够内嵌到 FPGA 结构中，它们在提供丰富的功能性的同时，也减少了功耗，降低了成本。目前纳入 FPGA 的部分硬 IP 组件有：存储块、计算电路、收发器、协议控制器，甚至还有中央处理单元（CPU）。不过，有一点需要记住，这些硬 IP 组件的定制程度并不像 FPGA 的其余部分！通过集成硬 IP 组件——例如：数字信号处理器（DSP），设计师在将这些常用组件添加到系统时，可以省去额外开发一个 DSP IP 的功夫。FPGA 制造商能够随手将硬 IP 组件并入 FPGA，因为这些硬 IP 组件的功能已经成为商品，在多数电子系统上都相当统一。

什么是 FPGA

以下各节将详细探讨 FPGA 的组成部分。有些人可能已经很久没有了解过 FPGA，或者忘记了部分具体内容，我将使用简明的语言进行说明。

FPGA 的基本组成部分

任何数字设备都可使用三个简单部件搭建：导线、逻辑门以及寄存器（参见图 1-1）。寄存器将记忆一条信息，直至被告知寄存另一条信息。逻辑门执行简单的信号逻辑运算，导线将这两个部件相连。



感谢英特尔公司提供。

图 1-1：数字系统的基本组成部分。

逻辑门

逻辑门执行数字电路的核心功能，即对输入信息执行简单的逻辑运算——这种输入信息是指您的电脑用来表示 0 和 1 的电脉冲。

就其自身而言，这些简单运算并没有什么；但是，当您把数万或数百万计的简单逻辑运算放在一处，您能够完成非常了不起的事情。您电脑的 CPU 由数十亿计逻辑门组成，这些逻辑门能够让您的电脑完成很多炫酷事情。

为了发挥功能性电路的作用，逻辑门使用一种被称为“布尔代数”的算术方法。布尔代数是 1854 年由乔治 · 布尔首次提出。布尔代数不同于初等代数，初等代数的变量值是数字，主要运算是加法和乘法；布尔代数的基本运算是：与（AND）、或（OR）以及非（NOT）。在布尔代数中，变量值是“真”值，即“真”和“假”。

布尔代数的基本运算如下：

- **And**（与）：表示为 x AND y。如果 x 和 y 均为真，则得出的值为真；否则为假。
- **Or**（或）：表示为 x OR y。如果 x 或 y 为真，则得出的值为真；如果 x 和 y 均非真，则得出的值为假。
- **Not**（非）：表示为 NOT x。如果 x 为假，则得出的值为真；如果 x 为真，则得出的值为假。

布尔代数的最常见用途之一恰好是数字逻辑设计。最终，布尔代数直接映射到数字电路——数字电路的输入分别是 0 和 1，或者说是“假”和“真”。通过将这些执行简单 0 和 1 布尔运算的逻辑门连接起来，连接的逻辑门越多，您的系统越可执行高级功能。例如：逻辑门可以引导火星探测器，让数十亿设备连接到全球定位系统（GPS），甚至可以让您在手机上支持您喜爱的游戏。

寄存器

寄存器是存储数据，以备将来使用的简单设备。寄存器就像您可以快速访问的短期数据存放点；您可以在拨号之前，把别人给您的号码存放在这里。当您努力想要记起某件事情时（比如：您必须参加约会的时间），相关电话号码将被马上替换成约会起始时间。寄存器将保存您希望它保存的任何信息，直至您告诉它忘掉这些信息并记住新的信息。

导线

所有数字设备都有的第三个部件是导线，导线将所有寄存器和逻辑门连接到一起。为使各系统完成您想要它们完成的任务，必须将这些元件连接起来。这些任务可以是简单的 1 + 2，也可以是比较复杂的任务，例如：将正在读取蓝光影碟的蓝色 LED 灯所发出的脉冲信号，转换成在电视屏幕上播放的、明快的高分辨率图像。只要有适当数量的逻辑门和寄存器，并将它们使用导线相连，就可搭建您想要的任何数字系统。

FPGA 的创意解释

但是，稍等！什么逻辑门、寄存器以及导线，这些东西听起来太技术、太抽象了。难道为了了解 FPGA，还得需要一位经验丰富的电子工程师？并非如此！为了解释 FPGA 的工作方式，可以打两个比方。第一个比方是珠子和串绳。第二个比方是您孩提时玩的互锁积木。

珠子和互锁积木这两个比喻最终说明了两种使用逻辑元件搭建电子系统的办法。

珠子和串绳

珠子和串绳代表的方法，能够让设计师最为精细地控制电路图。设计师使用小珠子和细线绳将所有元件连接起来，最终形成精致但也非常复杂的电路图。但是，这种精细控制难免耗费成本。即使是最细微的修改，要想不返工重来，也几乎是不可能完成的任务。当您把珠子和串绳映射到数字电子设计图上时，您得到的设计图与 ASIC 或 ASSP 非常相似。

想象一下，有一个珠串组成的电路图摆在您的面前，这个珠串由许多不同颜色的珠子组成，这些珠子被摆放成不同花样，并且使用串绳连接到一起。通过使用这些简单组件，您可以制作几乎任何类型的电路图，从最简单的类型到最复杂的类型，取决的因素是珠子的数量、颜色以及您选择的排列方式。

现在，把珠子想象成寄存器和逻辑门，把串绳想象成导线。就像珠子和串绳一样，这些元件生成一个系统——这个系统能够进行各种类型计算，从最简单的到非常复杂的。您可以把珠子的不同颜色想象成不同类型的逻辑门，例如：AND、OR 或 NOT。您会看到，这些简单的数学运算如何变成非常复杂的运算，就像只是排列一下不同颜色的珠子，一旦用串绳串起，就会变成非常复杂的图案一样。

通过排列珠子的图案，您可制成美丽的设计图。但是，当您想要通过重新排列珠子或变更珠子颜色修改图案时，会出现什么情况？事情正是从这里变得复杂！为了改变图案，您不得不解开串绳，才能换个方向。不久，您会发现珠子的连接十分紧密，您无法单独改变一个部分的设计。您还会发现，即便只是对图案的微小改动，也必须把整个设计图都拆开。所以，这个灵活性不太强，不是吗？

玩具积木

积木的体积相对较大且敦实，只适合一起放在积木盘的某些位置点上。积木搭建的设计不如珠子和串绳做成的样式那样雅致和繁复。尽管如此，您在修改部分设计时，却不用拆散整件东西并重头再来，这也促使我转而关注第二种数子电子设计方式：FPGA。

或许，您曾经在孩提时或与自己的孩子一起，使用积木搭建过大楼、消防车和宇宙飞船。与此相同，您也可使用积木来完美地表现数字系统：您可使用积木搭建一个桌子，然后想象这些积木有些是逻辑门，有些是寄存器，还有些是连接导线。

现在，想象有人告诉您，她想修改桌子右下侧的图案，并且也许还要改变积木颜色。因为积木都是可以相互连接的部件，您可轻松取下右下角积木，并且替换成其他一组积木。由于桌子上的其他积木原封未动，您不必为了一小部分而重做整个设计。

FPGA 与 ASIC 的比较

通常情况下，FPGA 比 ASIC 的灵活性更高，并且更加符合成本效益。在以下各节，我们将说明原因。

成本与灵活性

通过使用 FPGA，您能够执行 ASIC 能够执行的任何逻辑功能。不仅如此，FPGA 的独特优势在于芯片制造完成之后，还能更新芯片功能，这在许多应用都是理想需求。FPGA 比 ASIC 更具成本效益，这是因为客户能够按照自身需求对 FPGA 进行编程，而不是像 ASIC 那样，必须联系销售商，才可设计和搭建一个满足其需求的 ASIC。

设计时间风险的降低与速度

如果您计划不惜成本，采用世界上最先进的半导体工艺，您总能设计出比最快的 FPGA 还要快的 ASIC。但是，几乎没人会使用最先进的工艺：最先进工艺的使用过程将充满风险、困难，还有无法忍受的昂贵。事实上，当新工艺出现时，只有少数几家公司会转而采用。其他公司使用的是一代、二代或三代之前的工艺。还有一个事实，现在最快的 FPGA 已经能够直接与上一代 ASIC 竞争。另外，FPGA 不仅降低了设计工作量，而且极大减少了风险。

举例来讲，如果您正在设计一个有特定功率效率和性能要求的系统，并且计划采用较旧的 65 纳米 ASIC，那么您是否知道，当前 20 纳米 FPGA 也能达到类似效果？

另外，使用 FPGA 将缩短您的设计时间，降低发生设计错误的风险，并且总体拥有成本低于 ASIC。对于多数应用而言，FPGA 的功耗将能满足您的需要。因此，总体拥有成本更低、灵活性更高的 FPGA 往往是最佳技术选择。

纳米（nm）是芯片晶体管尺寸的计量单位。几十年来，晶体管的尺寸不断缩小。请参见第 2 章。

如果选择 FPGA 进行系统设计，这将给设计师带来更大的配置能力，同时降低开发时间表受到影响的风险——这是因为，正如我们用积木比喻证明的，设计师能够在不影响其余设计的条件下，修改小部分的 FPGA 设计。

FPGA 拥有令人惊讶的易用性

部分设计师有个错误的印象，以为如果采用现代 FPGA 设计系统，将意味着要做一个功能出来，设计师需要处理数百万计的逻辑门和海量连接器。但是，假如当真如此，FPGA 使用量不会增长，使用 FPGA 的公司应该不会超过五六家。

好消息是：FPGA 的设计师们已经完成了大量重体力劳动，并添加了通常需要的组件，例如：时钟发生器、动态随机存取存储器（DRAM）控制器、外围组件互连（PCI）高速控制器，甚至还有整个多核微处理器。所以，您只需专心定制那些您的应用所需的专门功能。

硬 IP 组件

在上文，我提到硬 IP 组件，这种组件是嵌入 FPGA 的知识产权组件，例如：DRAM 控制器、PCIe 控制器、时钟发生器以及大型存储块。事实上，今天 FPGA 上的硬 IP 组件实在太多，以至于这些组件实际上变成“片上系统”（SoC）。

不仅系统设计师需要的大部分常用功能被嵌入硬 IP 组件，甚至许多不太常用的功能也被加入，例如：雷达或通信用高速串行收发器，以及信号处理用数字信号处理器（DSP）倍增器—累加器。今天，甚至双核 ARM（一个微处理器设计品牌）CPU 子系统也

可嵌入。事实上，今天高端 FPGA 可能只有一半模具面积用于可编程逻辑，另一半则完全由硬 IP 组件组成。现在的设计师通常首先选择已经嵌入所需 IP 组件的 FPGA，然后根据特定应用，使用可编程逻辑对 FPGA 进行定制。

并行操作与降阶法

现在的设计师已经非常擅于制作工具，他们的工具能够非常巧妙地将复杂操作变为简单操作（“降阶法”），还可通过一系列可同时运算的指令，执行同一项复杂操作（“并行操作”）。

如何理解上述内容？下面我将举例说明：微处理器几乎可以做任何事，只需执行指令即可。如果程序说“乘法”，微处理器将从存储器载入指令、解码指令、载入各个数字、将数字相乘，然后将结果保存。其中每一步都需要时间和能量。但是，如果您想要做的是将一个数字乘以 2？这只是移位运算。通常，如果您要做乘法运算，并且您知道其中一个系数是定值，那么您就可将这个复杂运算减化成简单运算，同时节省您的处理时间和功耗。微型控制器不同于 FPGA，它没有智能工具，无法在可能的情况下，将乘法运算减化成加法运算。所以，微型控制器不得不执行乘法运算，导致执行速度降低，并且功耗加大。

在矢量数学等数字计算上，FPGA 可谓大出风头。矢量数学不仅仅用在物理课堂：每当程序员需要对每组数字执行相同运算，并且各组数字数量庞大时，都会使用矢量数学。在这里，FPGA 有一个很大的优势：尽管微处理器不得不单独处理每个数字——或者至多每次同时处理几个数字，但是您可对 FPGA 进行编程，让 FPGA 同时（并行）进行多项运算。如果您有一个 128 元矩阵，您可以搭建 128 条算术“流水线”，使所有这些运算可以同时执行，让您获得极大的性能和功耗优势。

往往 ASIC 或 ASSP 并不是恰当的选择！

第 2 章

FPGA 到底包含哪些组件?

本章提要

- 可编程逻辑结构与 I/O
- 向前发展
- 硬 IP 组件和 CPU 的嵌入
- 解决现代设计流程问题

阅读本章后，您就会了解 FPGA 是什么。如果您很早就接触了解过 FPGA，您也一定会对近年来 FPGA 的发展感到惊叹。FPGA 已不再是单纯的可编程逻辑门阵列。现在的 FPGA 包含了一些“开包即用”的内嵌硬件，这些内嵌硬件能够执行许多常见的功能。本章将为您介绍 FPGA 的具体组件，对其未来的发展进行展望，并阐述了其设计流程。

基础知识——可编程逻辑结构与 I/O

顾名思义，现场可编程逻辑门阵列（FPGA）的核心就是包含了很多逻辑门和 I/O 接口电路的简单集成电路。I/O 接口电路从数据源处取得数据，然后将数据从另一端输出到其他系统或子系统。

本书第一章讲述了电气系统的基本组成部分：逻辑门、导线和寄存器。电气系统的核心部分是一个矩形硅片，硅片表面蚀刻着导线和晶体管。这种硅片被称为集成电路（IC）。

晶体管是一种半导体器件，一般由硅制成。晶体管可用于切换和 / 或放大电信号，并且至少有三个连接器（或端子）与外界电路连接。在切换电信号时，晶体管在两个端子之间施加电势能（电压），在第三个端子上施加另一个电势能，进而在前两个端子之间，形成从晶体管一端流到另一端的电流，当第三个端子上的电压被移除时，没有电流通过电路。晶体管还可通过施加电压或电流来放大输入功率，使晶体管的输出功率大于输入功率。

晶体管类似于管道行业使用的弹簧阀。当您对阀门施加力时，水流流过管道（相当于导线）。当您停止作用力时，阀门关闭，水流停止流动。晶体管也是同样原理 : 当您对晶体管施加电势能时，电流流经晶体管，进入与电路连接的导线；当您移除电势能时，没有电流通过。

晶体管发展史

历史学者认为晶体管是 1947 年由约翰 · 巴丁、沃尔特 · 布拉顿和威廉 · 肖克利首次发明的。晶体管之所以伟大，是因为它开启了固态电子和集成电路的时代。与被晶体管取代的真空管相比，晶体管具有体积小，功耗低的优点，因此晶体管可以应用在很多小型设备上，您可以随身携带手机，一边播放 YouTube 网站上最新的萌猫视频，一边导航，开车来到您最喜爱的法越风味餐厅。如果您是技术发烧友，则可以在派对上娱乐一番，向朋友们解释说，“晶体管”这个词是约翰 · R · 皮尔斯自己造的，本义是“传输电阻器”。没准这点小趣闻还可能让您在智力竞赛 Jeopardy! 上派上用场，帮您赢得惊天奖金和奖励。

布尔代数实际上是对输入值“真”或“假”（或数字表示的 1 或 0）的数字运算。逻辑门是对输入值 0 或 1 进行各种布尔代数运算的设备。由于晶体管是通过施加或移除电位来切换电信号，因此您可以通过排列一组晶体管来创建一个逻辑门，让它执行一种布尔代数运算，比如：AND、OR 或 NOT（关于布尔代数运算，详见第一章）。

FPGA 的核心就是这样一个蚀刻到集成电路上的逻辑门阵列，并且这些阵列的蚀刻方式允许您进行重新配置。如果您愿意，可以回顾一下第一章中将 FPGA 与 LEGO 积木进行对比的实验。

简单来说，FPGA 就像放置到长条桌上的大型彩色矩形阵列，并且这个阵列能够按照所有者（FPGA 程序员）的意愿进行排列。

摩尔定律

摩尔定律最早出现在 1965 年 4 月 19 日《电子学杂志》上发表的一篇名为“让集成电路填满更多元件”的文章中。在这篇文章里，摩尔准确地预测到，在晶元（搭建集成电路的硅片盘）保持不变的情况下，电路的复杂性将每 2 年增加一倍。到了 20 世纪 70 年代末，摩尔定律被广泛用作复杂芯片上晶体管数量限制的公式。更为惊奇的是，在摩尔定律首次出现后的 50 年内，摩尔定律的预言一直非常准确。

简而言之，摩尔定律是说：由于每隔 18 到 24 个月，内置集成电路的尺寸可缩小 30%，进而集成电路中可容纳的晶体管数量也会翻倍。摩尔定律的确让事情变得更加复杂和困难。例如，在小面积的硅片上添加所有的导线和晶体管，经过蚀刻加工，很难保持原始电路设计的布局完整性，而且投射的镜像也会有些变形或失真，如线路比设计的宽或窄，晶片的圆角出现扭曲等。

过去，设计师只发现了 FPGA 的简单用途，例如：作为计算机的简单接口，创建基本逻辑功能。除此之外，基本没有其他的用处。

向前发展

当在一小片硅晶元上添加数百万个导线和晶体管时，为了消除可能发生的蚀刻问题，设计师将晶元掩盖起来。

当缩放芯片以添加数百万甚至数十亿个晶体管时，还会出现另一个问题：晶体管太小，因此极难关断晶体管（切断电位）。随着晶元上的晶体管数量不断翻倍，设计变得越来越复杂，并且晶体管漏电流越来越多，即使芯片没有做任何计算，就已经在消耗功率。另外，晶体管的性能也会下降，这就需要做大量的工作来避免芯片运行速度的下降。

难道芯片的发展即将走到尽头？现有的设计造成大量的能量汇集到芯片上的一小块区域，这可能熔断芯片内部的导线！工程师们走到了十字路口：他们必须从根本上改变晶体管的设计方式，从平面布局转向 FinFET。所谓“FinFET”，是指搭建在硅晶元上的非平面、多门、场效应晶体管（FET），目的是减少晶体管占用的表面积。“FinFET”的名称来自于构成晶体管导电沟道的“鱼鳍”状硅片（Fin）。

芯片设计师未来将面对怎样的改变？在并不遥远的过去，芯片上只有大约 20 个晶体管。在未来几年，芯片上将搭载百亿计晶体管。在能够设计出这些芯片之前，设计师将首先设计出能够靠边缘放置的晶体管、不同类型的导线，还有许多其他变更。芯片设计将深入很多未知领域——芯片将能自我诊断、补偿和治愈。事实上，随着芯片设计师不断应用摩尔定律，未来将成为令芯片设计师振奋的时代。

摩尔定律在集成电路中得到了成功的应用。今天的集成电路可能包含数百万甚至数十亿计晶体管，能够执行非常复杂的运算，例如：高速数据联网、高级三维图形计算，或者在互联网上传送和播放高清影片。摩尔定律不仅适用于集成电路，而且其应用还带来了 FPGA 的革命。

硬 IP 组件与集成 CPU

今天，在硬件尺寸相同的条件下，在 FPGA 内实现的设计与在 ASIC 实现的相同设计相比，往往能够提供相同的能效和速度。其中部分原因是，FPGA 供应商在 FPGA 内部嵌入了大量预定义硬件来实现一些常用功能，包括标准接口甚至整个微型计算机。

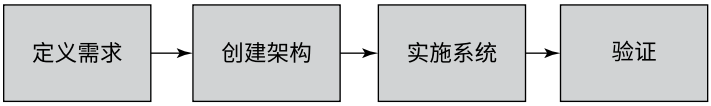
FPGA 可以进行现场编程，因此在硬件不变的情况下，设计师能根据设计变化对硬件进行重新编程；如果设计时采用的 ASIC，一旦设计变更，这意味着必须重新搭建硬件，来实现新的设计。这是 FPGA 相对 ASIC 的一大优势。越来越多的设计师偏向于在设计中使用 FPGA，而不是 ASIC。如果您想要在设计中使用 FPGA，请阅读下文，以更多了解 FPGA 的设计流程，同时充分利用 FPGA 的内嵌硬件。

FPGA 的现代设计流程

FPGA 设计之初，首先是“参考设计”，参考设计是可供他人复制的系统技术蓝图，包含系统的核心元件。参考设计通常由应用工程师来完成，作为销售支持工作的一部分，只是客户性质有所不同。逐渐地，参考设计不再是销售工具——参考设计本身就已是产品！

创建功能框图

那么，系统设计流程到底怎样？图 2-1 所示为高级系统设计流程的简单框图。

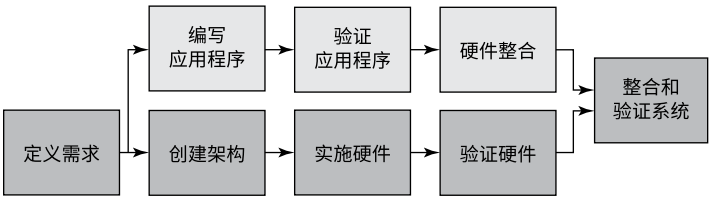


感谢英特尔公司提供。

图 2-1：系统设计流程。

系统设计流程似乎与您想的一样。首先，您要定义需求，按照定义，创建系统架构。在这里，您可决定您需要的设计实现组件。然后，利用自己规划的架构实现系统。最后，验证您的系统是否满足全部需求。

图 2-1 所示为系统设计流程简化视图。其中，“制作架构”和“实现系统”是需要动手的两个环节。在这两个环节，您将决定系统架构，以及搭建系统设计实现所需要的硬件和软件应用程序。您还可将“定义需求”和“验证”两个环节进一步细分为单独的流程，这个流程可被称为“软件应用流程”。图 2-2 将“软件应用流程”添加到图 2-1 的系统设计流程图。



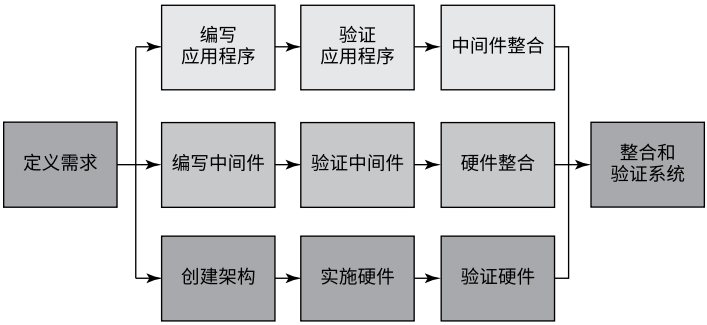
感谢英特尔公司提供。

图 2-2：系统设计流程和软件应用流程。

在“定义需求”与“整合与验证系统”两个环节之间有一些淡色色块，这些色块构成应用程序设计流程。在该环节，您在编写和验证应用软件之后，将这些应用程序与硬件整合之后，可以验证系统是否符合设计要求。

应用程序的设计流程就是开发系统实现所需的应用程序的开发步骤。根据相关系统将要部署的应用程序类型，设计师往往还必须考虑相关系统将要运行的不同平台（例如：汽车、通信等等）。往往，不同的应用程序领域都有现成的软件和硬件标准，以确保为这些系统开发的应用程序能够执行常见功能，并且能够相互兼容。

以安卓手机为例。安卓操作系统包含一些常用组件，这些组件可为安卓系统开发的所有应用程序使用。不仅如此，安卓操作系统还为摄像头的访问和应用程序之间的数据共享设置了平台标准。设计师往往在其设计的应用程序中，包含通常所谓的“中间件”。中间件是一个软件层，虽然并不实现产品核心功能，但是提供了一个执行行业标准或协议的软件层面。设计师制作中间件的目的，是将设计师设计的应用程序逻辑与具体标准或开发平台的专用逻辑相隔离（例如：Android 系统或 Apple iOS 系统）。中间件往往还可在许多其他应用程序中使用。在应用程序中添加中间件，还有一个重要功用：中间件使得您的应用程序兼容未来的标准和平台有了可能性。图 2-3 中显示了中间件开发时如何适应系统设计流程的。

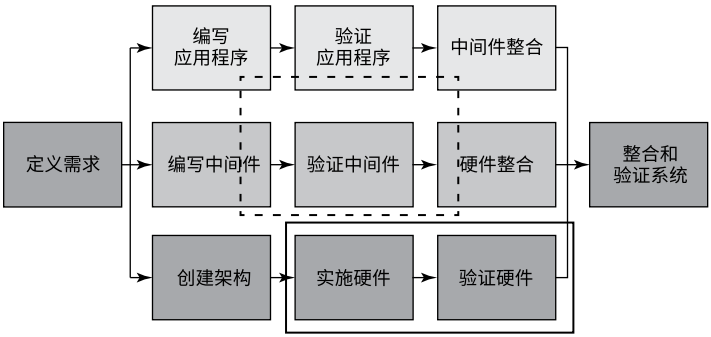


感谢英特尔公司提供。

图 2-3：系统设计流程中的中间件。

利用现有 IP 组件代替功能块

从框图来看，当您搭建应用程序以及将应用程序整合到系统时，您的工作就像是事先分配好的。但是，FPGA 制造商经过多年实践，已经知道大多数系统应该具有很多同类的功能。例如：网络数据 I/O 接口、图形处理以及微处理器等功能是普遍需要的，因此让每个系统设计人员重复设计这些组件显得意义不大。比较合理的做法应该是让这些功能类型“开包即用”。近年来，FPGA 制造商已经逐渐将这些常见功能或知识产权类组件（IP）加入他们的产品。这些 IP 组件可以是嵌入芯片的硬件、提供给用户的软件，或者——就是 FPGA——用户可以加入可编程逻辑的硬件设计。现在，您可利用现有 IP 组件，替换框图中的部分功能块——工作即可简单完成。图 2-4 所示为现有 IP 组件与设计框图的配合。



感谢英特尔公司提供。

图 2-4：利用现有 IP 组件代替功能块。

实线矩形所示为使用嵌入芯片的硬件和可编程逻辑 IP 组件达到的效果：“实施硬件”和“验证硬件”等许多环节已经代您完成。另外，图 2-4 中的虚线矩形所示为现有 IP 组件与设计框图的配合情况。这里，现有 IP 组件替代了部分应用程序和中间件的编写工作，代您实现了相关功能。

真实的“虫”

如果您非常喜欢研究词源，那么您一定得知道，软件中首次出现“虫”这个词，要归功于计算机先行者——格蕾斯·霍珀。1947 年，她发现一台机电式继电器困住了一只蛾子，于是把程序执行中出现的相关小故障称为“虫”（Bug）！

为缺失的功能块编码

现有 IP 组件能做的只是实现一些常见功能，例如：帮助车载导航系统访问 GPS 数据。FPGA 的其余现场可编程工作仍需设计师完成。最终，您是想让 FPGA 与您的应用程序完全匹配，不是吗？

现代 FPGA 的编程工作比您想象的简单得多。FPGA 的编程环节包括：确定程序设计模块，选择一种高等级语言或硬件描述语言（HDL），在文本编辑器里编写代码，设计综合（参见后文），对设计布局布线，然后将设计载入 FPGA。在设计载入 FPGA 之后，可能需要一个“除虫”（调试）周期，以修复功能性错误。

在软件和硬件开发术语中，您会遇到一个词“虫”（Bug）。“虫”是指计算机软件中的令人费解的缺陷，这个缺陷带来不正确或意外的结果。“除虫”（调试）指消除缺陷，直至整个设计按照规定功能正常运行。

当您对相关设计的运行情况满意之后，下一步工作就是程序归档，最后把程序提交给客户。

那么，如何对那些无法从库里导入的功能块进行编程呢？在早期的 FPGA 设计中，您只有一个选择——硬件描述语言 HDL；顾名思义，HDL 语言使用硬件术语——导线和逻辑门——来描述功能块。这使得 FPGA 编程局限于硬件工程师和坚定的发烧友。但今

天，由于 OpenCL 等高等级软件编程工具的出现，即使是没有专门硬件技能的软件设计师，也能指定子系统功能、编写代码，以及生成可以载入正常 FPGA 设计流程的硬件描述文件。有了这些工具，在硬件工程师只能提供极少支持，甚至完全没有支持的情况下，您仍可实现硬件加速，让现代 FPGA 的力量向更多受众敞开。

对于那些仍然需要详细设计的功能块，设计师将使用 HDL 语言。Verilog 是实现 FPGA 设计的常用 HDL 语言。Verilog 语句与常用的通用型编程语言——C 语言非常相似。但是，不同在于 C 语言用于定义在计算机上运行的程序，Verilog、VHDL 以及其他硬件描述语言描述的是设计师想要在 FPGA 中创建的硬件——由逻辑门、寄存器和导线组成的相互连接的网络。您在编写 Verilog 程序时，可以使用简单的文本编辑器按照正确的语法来编写自己的程序。

在 HDL 语言设计编写完成之后，下一步是编译 HDL 设计。在 FPGA 编程里，综合工具将 HDL 语言设计作为输入内容，将其转化成由逻辑门、寄存器和导线组成的网络，并且这个网络被配置成实现 HDL 语言描述的功能。然后，将通过其他工序，挑选将在 FPGA 中使用的特定逻辑门、寄存器和导线，并且创建一个编程文件，该编程文件将在 FPGA 启动后，对 FPGA 进行配置。

因此，您编写的 HDL 代码将被直接映射到特定 FPGA 设备的可用物理硬件元件。在微处理器编程过程中，程序逻辑被映射到处理器必须执行的指令列表。所以，这项功能能够将逻辑直接转化成可以执行的硅片逻辑门，这是一个与众不同的、强大的功能。

在这一过程中，设计工具还可链接到硬 IP 组件设计，即已经嵌入 FPGA 的、经过预定义的硬件功能块。在现代工具流程里，您要做的只是指定您是否需要使用硬 IP 或软 IP 功能块，以及您想要的连接方式。您只需要对还不属于 IP 组件的功能块编写 HDL 代码。

验证系统设计

在完成代码编译之后，需进行测试，然后才可将代码配置到 FPGA。过去，设计师面对的可编程逻辑芯片要简单得多，他们在测试时，只是验证能否正常工作。但是，由于现代 FPGA 的复杂性，早期的调试手段已不再可行。

FPGA 设计调试通常在模拟环境下进行。模拟器是一些软件应用程序，这种应用程序通常可以模拟您的设计行为。模拟过程通过软件实现，并且通过软件，您能够看到每个寄存器的表现。然后，再将您的设计放入 FPGA。

代码的调试和验证通常反复进行，直至您确信 HDL 代码按照预想运行。开发人员大都使用一种叫作“Testbench”的工具，来验证 FPGA 在现实世界能否正常运行。Testbench 是由您自己设计的，将软件仿真与实际硬件相结合，构成了实际的系统模型，该系统中包含了您的 FPGA。FPGA 大都包含数以万计或十万计的逻辑门，因此您不可能全部测试。Testbench 主要用于测试主要设计区域中实际工作的逻辑门。模拟环境可帮您隔离某些特定设计区域，以及在这些区域添加调试辅助工具，让您的设计能够按照您的意愿运行。

任何高质量的软件应用程序，都需要大量文档来确切地告诉客户或最终用户该应用程序的定义方式，包括一些附加说明、警告等等。FPGA 的文档要求与任何微控制器编程文档相同，当然，内容截然不同。

将系统映射到 FPGA 硬件中

最终，这些组件经过综合后，必须载入 FPGA，以执行系统的逻辑门。

像任何系统一样，如果硬件正确，那么相关设计就可发展为包含漏洞修复和功能增强的方案。HDL 代码的编辑能力使得设计、调试和验证工作可以在同一环境下完成，这将有助于您加快 FPGA 的上市时间。

在系统内验证设计

当设计被编入硬件之后，应确保一切都按照其既定目标运行。那么，对于 FPGA 而言，所谓的“运行”到底是什么？“运行”阶段有时也被称为“闭环”阶段。对于任何硬件器件，都有一定的性能标准。在许多应用里，功耗都是一项重要的设计标准。以智能手机为例。为实现可以接受的电池续航能力，智能手机有严格的功率要求。您可不想让自己的智能手机变成用电大户，否则您会发现手机才用不久就没电了。速度是另一个重要标准。设计师还应该测试和确保每个网状结构（逻辑门之间的导线连接）达到时序要求。最后，确保每个时钟和电源管脚都连接到您的 FPGA。

在 FPGA 的设计工具环境下，您可输入自己的设计，选择您想要的 IP 组件块，然后将您的设计转化成为 FPGA 中实际存在的硬件元件。当设计还在软件中，方便测试时，您可验证该设计是否达到预期和所需速度，甚至还可估算该设计将要消耗的功率。

现在，到了见证奇迹的时刻：将经过测试的设计载入原型板上的目标 FPGA，接通系统电源，然后验证每个部分的运行是否达到预期。您现在拥有了完美符合您要求的定制硬件——比您收到首批 ASIC 芯片样品要提前几个月。

第 3 章

FPGA 系统

本章提要

- 理解作为基本功能组件的 FPGA
- 使用片上系统（SoC），将系统引入 FPGA

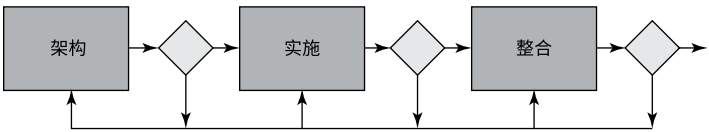
本章将为您介绍 FPGA 的实际应用。FPGA 可作为一个系统的基本功能组件，随着 FPGA 变得越来越大，可将整个数字系统引入 FPGA，实现一个片上系统。

在本章，我将介绍电子系统的概念。并且为您介绍高端汽车中复杂的片上系统。

系统设计中的 FPGA

本节将让您深入了解系统设计流程，以及 FPGA 如何起到重要作用（第 2 章已经介绍了部分基础知识）。

图 3-1 所示为传统的系统设计模型。您看到的每个方块之间的菱形表示相关设计过程的决策点。



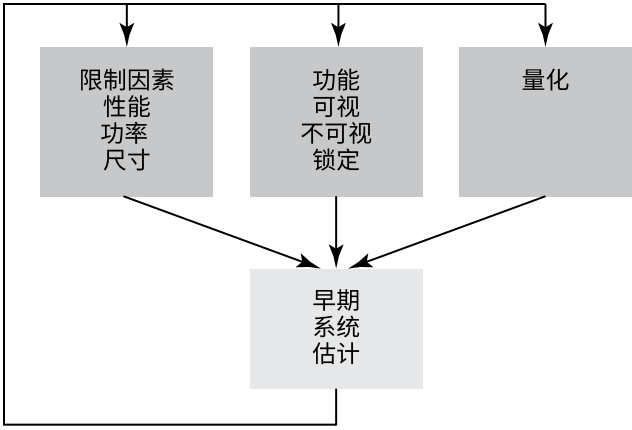
感谢英特尔公司提供。

图 3-1：系统设计与决策点。

在系统设计中，决策点是您需要提出问题的点，例如：

- **系统必须实现哪些功能？**这个问题在需求定义阶段出现，是最基本的问题。这个问题的答案往往由产品经理与客户协作提供，并且对需求收集阶段有推动作用。
- **能否利用我的现有设计并做出改动？**经常已有现成的系统，只要做一些改动，就可满足系统需求。
- **有多少系统需要通过软件处理？**这是在设计和实现阶段问到的一个重要问题。通过软件处理的系统量决定了可以使用的硬件类型。FPGA 和微控制器可供软件编程。
- **我能够购买多少现成硬件？**许多时候，系统设计的功能模块可能已经涵盖在商用硬件器件中。在这种情况下，与利用软件实现该设计或设计定制硬件相比，购买该硬件或取得 IP 组件授权可能更为经济。
- **是否正常运行？**这是集成阶段的最基本问题，答案必须为“是”，然后才可进行系统部署。如果答案为“否”，那么您必须反复检查设计和实现阶段，直至把问题解决。

在图 3-2 中，您可能会看到，在需求定义阶段，设计师必须考虑设计限制因素，例如：性能、功耗和尺寸等。系统的功能也很重要——包括哪些功能可见、哪些不可见或锁定。最后，设计师对设计进行实验。这一过程促使进行早期的系统评价，得出系统的实际尺寸，范围以及系统实现的需求。



感谢英特尔公司提供。

图 3-2：定义系统需求。

FPGA 的设计过程是不断反复的过程——您首先有一个系统理念，然后将该理念细化并定义为“事务”。一个事务将包括输入、处理和输出。您可以把这里的事务想象成银行交易：您把钱交给出纳员，出纳员拿到钱，然后把钱交给银行。基本上，一个事务就是在两个系统组件之间发生的信息共享行为，并且这种事务可以在系统任何位置发生。

在事务定义完成之后，您要实现这些事务，然后再验证这些事务能否正常运行，并且符合您最初设置的功能需求和限制因素。这也是一个不断反复的过程，需要决定哪些功能在硬件完成，哪些在软件完成，哪些在现有 IP 组件完成，哪些必须重新编写。

FPGA 与自动化电子系统

本节给出了一个现实世界的实例。事实上，您可能今早还在运用该系统。我将说明的是片上系统（SoC）——其中许多都是 FPGA，以及片上系统如何在现代汽车中应用的。

动力传动系统

想象一辆汽车传动系的组件，以及这些组件如何在法规、安全、成本以及功能等的要求下全都实现电子化：

- **引擎：**引擎里的电子系统按照动力需求、排放量、路面平整情况、起动周期和策略等，控制着燃料、点火和阀门。
- **变速器：**现代变速器包含电子系统，可控制齿轮比、换档顺序、速度信号、动力需求以及发动机转速（每分钟转速，RPM）。
- **制动装置：**为安全起见，电子系统——不仅指地板上的脚踏板——能够控制制动力。
- **转向装置：**高端汽车拥有复杂的动力转向装置，可按照多个输入项目，控制转向比、转向反馈和转向角度，电子系统可以掌控转向。
- **轮胎：**近年来，随着汽车的进步，可以监控轮胎气压的电子传感器也被加入汽车，这样驾驶员就可知道什么时候给轮胎充气——不仅提高燃料效率，还可延长轮胎寿命。

信息娱乐系统

“信息娱乐系统”指的是汽车上的信息系统和娱乐系统，许多汽车都有复杂的电子信息娱乐功能，例如：

- **显示器和控制器：**今天的汽车都安装了由电子系统控制的速度表等仪表。
- **娱乐：**现代汽车有一些高级功能，例如：数字式 AM/FM 广播、卫星广播、CD 机，以及可以存放整个音乐库的数字音频播放器。有时，还安装数字视频系统，让您的孩子在长途旅行中不至于无聊。
- **舒适度：**现在，驾驶员和乘客分别有灯光和多区域空调控制系统，所以，即使在同一辆车内，他们也可以“要风得风，要雨得雨”。

- **权限控制：**今天的汽车都安装有电动锁、电动门、电动窗以及电动安保系统。另外，还有一些常用安全功能，例如：门窗控制装置，可防止儿童在车辆行驶过程中打开门窗。
- **被动安全系统：**安全系统可以确定车内人数，并且在感知即将发生碰撞时，及时应对。

辅助驾驶系统

辅助驾驶系统包括近年来汽车设计中出现的炫酷技术。辅助驾驶系统让车辆更加安全！辅助驾驶系统包括：

- **灯光、倒车、偏道和防撞系统：**汽车往往装备高级照明系统、指示灯以及前镜显示器，可在汽车偏离车道或将要撞上其他车辆或物体之前，向驾驶员发出警告。
- **传感器，包括摄像头、激光器和雷达：**这些传感器用于帮助驾驶员在倒车或变道时看到盲区，极大降低了事故发生几率。

FPGA 的重要性

今天，多数汽车系统都依赖于那些低成本微控制器，进行检测或操作。汽车行业的设计趋势是系统整合，以及系统自动化程度的不断提高。随着系统越来越复杂，处理和存储需求也在飙升。利用卡尔曼滤波器进行传感器融合值得考虑。

传感器融合是指将来自不同数据源的传感数据组合，这样产生的信息要优于来自单个传感器的信息，例如：立体视觉（将两个摄像头摆在稍微不同的视角，通过组合这两个摄像头的二维图像，来计算深度信息）。卡尔曼滤波器算法使用的往往是来自不同类型的传感器的、一定时期内的、包含噪声（随机变化）的实测数据，所以生成的估算值要比基于单一实测数据更加精确。卡尔曼滤波器一般用于引导、导航和车辆控制。

随着汽车系统的整合，微控制器已经被纳入片上系统。随着系统的智能化和自动化程度越来越高，片上系统正在向多核处理器 / DSP 集群的方向发展。为了满足模型数量的爆炸性增长、新品推

出年度的变化、总线架构的发展以及安全需求的不断提高，目前趋势是 FPGA 片上系统，这是应对这些挑战、满足频繁更新需求的唯一可行方案。

汽车还只是现代系统越来越依赖电子元件的实例之一。电子系统越来越复杂化，并且变化速度也越来越快；片上系统需求也在发展，甚至在新品推出年度就可能变化。相同趋势也出现在其他具有复杂行为的各种产品中，从汽车和火车到电网甚至家电，甚至您的烤箱也在其中。

第 4 章

放眼未来：异构计算与 OpenCL

本章提要

- 什么是异构计算
- 什么是 OpenCL

作为行业趋势，FPGA 将在异构计算范式下发挥重要作用。开放式计算语言（OpenCL）是一个行业标准开发平台，用于在异构环境下进行 FPGA 编程。

本章将为您介绍异构计算的必要性，以及新出现的软件语言平台。

异构计算

在数据中心内部有一个重要趋势，即计算架构的转变：从多核 CPU 转向异构计算。所谓“异构计算”，是指使用不止一个类型的处理器，实现一个具有专业处理能力的系统。例如，图形渲染系统就是一个异构计算系统，该系统使用 CPU 和 GPC（图形处理单元）在计算机上渲染三维图形。GPU 尤其擅长渲染三维场景，以及在数学意义上，对大型数据集执行密集计算。CPU 在后台工作，可以执行操作系统任务和数据联网任务。随着系统的整合，

异构计算系统越来越标准化，并且必须包含几种不同的处理器架构。

平行计算是计算机具有的同时执行多个计算的能力，其原理是将大问题拆分成小问题，然后同时解决（即“平行计算”）。平行计算有多种形式：比特级、指令级、数据及任务等。平行计算已不再局限于高性能计算（例如：IBM 公司的国际象棋手“深蓝”）。随着功耗越来越成为内嵌电子系统的设计考量因素之一，平行计算已成为计算机架构的主要范式，并且大都以多核处理器形式出现。

所谓“数据平行”，是将数据分散给多个处理器，这样就可平行执行。多核处理器的做法往往是：将一个程序的多个实例“分包”给各个处理器，以同时执行这些指令。所谓“任务平行”，是由一个处理器将计算机的代码块（线程）“分包”给不同处理器，以平行执行这些代码块。

为什么在 FPGA 上使用 OpenCL？

对异构计算的需求引发了一些利用新硬件的新编程语言的出现。其中之一就是苹果公司最早开发的 OpenCL。OpenCL 是一个可编写程序的框架结构，这种程序可在异构平台上执行，包括 CPU、GPU、DSP、FPGA 以及其他类型处理器。OpenCL 包括内核（在硬件设备上执行的功能）开发语言，以及让主程序能够控制内核的应用程序接口（API）开发语言。OpenCL 允许进行任务平行计算和数据平行计算。

过去十几年间，处理器的硬件频率遭遇所谓“功率墙”——处理器的频率无法进一步提升。CPU 制造商们并未积极提高时钟频率，而是主动向 CPU 里添加更多处理核心，同时强化指令集，使多个指令能够同时执行，以及在不提高时钟频率的条件下，加快程序的执行速度。软件公司也在忙于开发一些软件，让大块的计

算机代码（线程）能够真正地平行执行。这些线程可在单独的处理器核心执行，而不像过去那样的伪平行计算：线程并不在单独的处理器核心上执行，而是被操作系统切分成时间片，使得表面看来似乎在平行计算。

FPGA 原本就是平行计算的，因此可以完美适合 OpenCL 的平行计算能力。除了典型的数据平行和任务平行，FPGA 还提供管道平行；在管道平行模式下，可通过推拉配置生成任务，每个任务从上一个任务接收数据，并且可发生或不发生主机交互。OpenCL 允许您在熟悉的 C 语言环境下开发代码。然后，利用 OpenCL 提供的其他能力，您可将代码分隔成可以平行执行的正常软件与内核。这些内核代码可以发送给 FPGA，而不必知道 FPGA 设计师的低等级编码习惯。一般而言，当使用 OpenCL 开发 FPGA 代码时，软件开发人员和系统设计师将有几个优势：

- **简单性和便利性：**多数软件开发人员都熟悉 C 语言，但并不熟悉低等级的 HDL 语言。OpenCL 让您保持在较高的编程等级，确保系统能够有更多软件开发人员参与。
- **代码剖析：**通过使用 OpenCL，您可以剖析代码，判断其中影响性能的部分；这些部分可作为 FPGA 的内核，进行硬件加速。
- **性能：**效能功耗比是系统设计的终极目标。通过利用 FPGA，您将在高性能与高能效解决方案之间取得平衡。
- **效率：**FPGA 拥有精细的平行架构，通过 OpenCL，您可生成达到高性能所需的逻辑。
- **异构系统：**有了 OpenCL，您可开发面向 FPGA、CPU、GPU 以及 DSP 的无缝内核，获得真正的异构系统设计。
- **代码重复利用：**理想的软件开发是实现代码的重复利用。对于软件开发人员和系统设计师而言，代码的重复利用往往可望而不可及。OpenCL 内核允许提供“便携”代码，您自己的代码可实现跨项目、世代和种类使用，从而延长代码寿命。

今天，OpenCL 语言的开发和维护者是柯罗诺斯技术联盟。

多数 FPGA 制造商都为 OpenCL FPGA 的开发提供软件开发工具包（SDK）。

第 5 章

FPGA 的五种应用

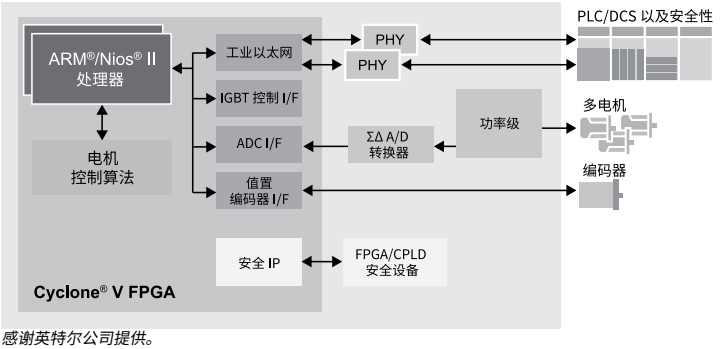
本章提要

- 介绍现代 FPGA 在现实中的应用

FPGA 经历了很长的发展历程，当初要使用 FPGA 实现一个复杂的系统，需要做大量的逻辑门编程工作。今天的 FPGA 有内嵌功能组件，例如：网络接口、存储模块，甚至 ARM 核心。英特尔公司内嵌 ARM 核心的 FPGA 被称为“FPGA 片上系统”（SoC FPGA），以此表明芯片功能的强大。当今的现场可编程部分有时还不到芯片面积的一半。本章重点介绍 FPGA 在多个行业技术领域的应用情况。

单设备电动控制器

电机和电动控制器在任何工业设计中都很常见。当您走进任何工厂或工业设施时，您会发现千差万别的机械之间有一件共同的东西——这些机械都采用电机作为动力。多数电机控制系统都采用微控制器技术设计。然而，微控制器却不能满足复杂电机控制算法的性能需求，例如：直接转矩控制（DTC）或无传感器磁场定向控制（SFOC）。过去，曾经使用数字信号处理器（DSP）克服这个问题，但当涉及高性能时，其成本效益通常无法与 FPGA 匹敌。您可在单个 FPGA 片上系统上，搭建一个复杂的、可伸缩的、高性能的电机控制系统（参见图 5-1）。



感谢英特尔公司提供。

图 5-1：电机控制器。

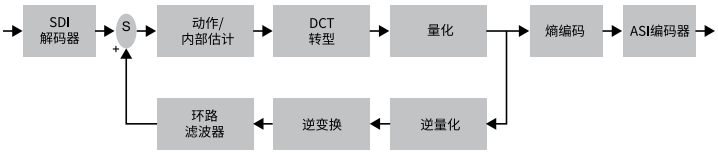
例如，可使用 FPGA 片上系统插入一个电机控制模块，该模块有两个独立控制的直流电机，以及一个简单的光反馈系统。FPGA 片上系统包含一个管理反馈信息和控制信号的内嵌处理器，这样两台电机可独立运动。该处理器读取反馈系统提供的数据，并且执行相关算法，实现两台电机之间的运动同步和转速控制。通过使用 FPGA 片上系统，您可搭建自己的 IP 组件，该 IP 组件定制方便，可在其他电机控件上运行。使用 FPGA 片上系统代替微控制器控制电机动作，有几个优势：

- **系统一体化：**通过将工业网络、安全组件、功率级接口以及 DSP 控制算法等集成在单个设备上，不仅减少了部件，而且降低了原料成本、功耗需求并提升了可靠性。
- **性能可伸缩：**您可在整条产品线上只使用一个可伸缩平台。FPGA 片上系统允许您使用速度更快、等级更高的控制回路，从而能够实现更高性能，提高效率和延长机械寿命。
- **功能安全性：**由于自动化系统更多负责运行具有潜在危险性的设备，因此监管部门要求机械控制电子系统确保不会造成危害。有了 FPGA 片上系统和适合的设计流程，可减少在政府及行业安全规范合规方面花费的时间和经济成本。

电视广播

电视广播站采用串行数字接口（SDI）标准，在 75 欧姆同轴电缆（与有线电视 / 卫星电视接收器和电视之间的电缆相同）上传输无压缩数字视频。每次视频图像质量改进，该标准都要扩容。最新标准被称为“3-Gbps (3G)-SDI”，能够在演播室内传输 4K ultraHD 信号。伴随着这些变化，FPGA 也迎来另一个闪光点！FPGA 解决方案提供核心收发器，能够在同一台收发器上，兼容所有三种 SDI 速率（SD SDI、HD SDI 以及 3G-SDI）。

不仅如此，演播室内还有其他许多变化。新的数字技术可帮助剪辑视频流、改进或修正画量，以及压缩图像，以供电缆或卫星线路传输。最新的压缩标准 H.265（也被称为“高效视频编解码标准”）极大减少了影视节目的编码位数。但是，该标准需要海量计算。许多设备供应商都发现，一边要把功能打包到一个片上系统，一边又要解决快速发展造成的压力——二种需求的再次组合，FPGA 是最佳解决方案（参见图 5-2）。



感谢英特尔公司提供。

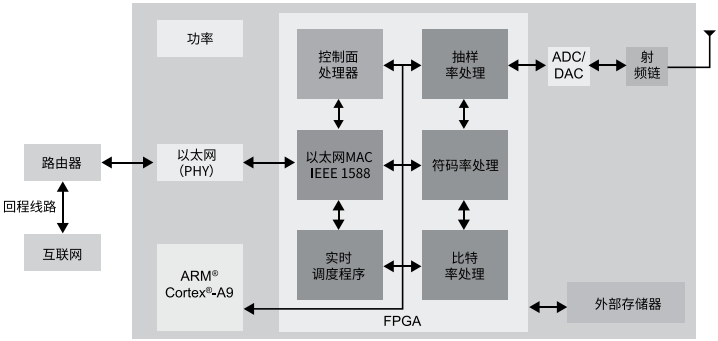
图 5-2：广播。

无线数据

4G 无线技术的到来，给人们的工作和生活带来无与伦比的变化。无线技术让您能够随身携带电话，您不仅可以随地拨打电话，而且可以随地浏览网页和发布最新消息！最新的技术迁移是从 4G 标准迁移到 5G 技术。

对于基站和移动运营商而言，随着他们努力降低支出以及扩展和升级网络，他们的主要需求是：可伸缩因素、低功耗、低成本以及可编程能力。制造商在寻求提高产能和差异化时间，并将此作为成功推出产品和获得竞争优势的关键因素。

现在，许多 FPGA 都配备内嵌式低延时高级网络 IP 组件和产能强化工具，使得制造商能够利用 FPGA 的性能、功率、价格和产能优势，把精力主要放在产品差异化上，而不是对无线设施基本组件的机械式编程（参见图 5-3）。



感谢英特尔公司提供。

图 5-3：无线基础设施。

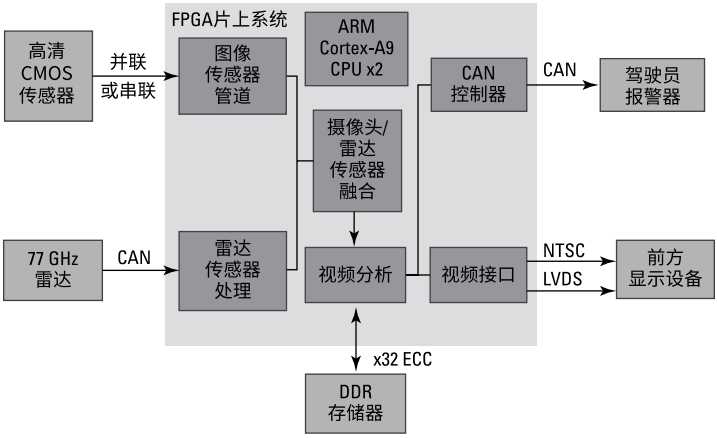
自动驾驶辅助摄像头

在汽车行业，一个重点发展领域是技术驱动型组件的爆炸式增长。即便是价格低廉的汽车，也配备了导航系统、视频娱乐系统和摄像头等新奇玩意儿。

驾驶辅助系统和倒车摄像头是最重要的安全创新成果之一，使得汽车比以前更加安全。当您在倒车时，如果对车后情形一清二楚，真是惬意极了。

前向摄像头系统由高速视频处理系统、复杂的传感器融合系统以及实时数据分析系统组成；当发生驾驶员打瞌睡或偏道等情况时，该系统可使车辆执行修正措施。前向摄像头结合了雷达和激光传感器等各类传感器实际功用。鉴于每种传感器提供数据的方式都不同，从而形成多架构设计挑战。

传统 DSP 处理器或微处理器没有同时进行实时视频处理和分析的能力。另外，HDR 技术（高动态范围图像技术，让摄像头平等地看到一个场景内的亮区和暗区）也是确保视频分析准确性的一个要件。与传统非 HDR 摄像头相比，HDR 技术几乎使得信号处理能力需求增长三倍，这也使得几乎只有最昂贵的 DSP 才能满足这种性能需求。但是，不需要 DSP 或微控制器，您可将整个摄像系统整合到一个单独的、低成本的 FPGA 片上系统。您可使用 FPGA 逻辑开发硬件平行处理引擎，以及综合运用 FPGA 片上系统硬处理器系统的软件算法，以此提高系统性能。图 5-4 所示为构成车辆可视系统一部分的 FPGA 片上系统。



感谢英特尔公司提供。

图 5-4：车辆可视系统中的 FPGA。

高性能计算

高性能计算（HPC）市场是今天增长最为迅速的计算领域。在金融、医学成像、生物科学、军事以及许多其他能够受益于 FPGA 逻辑和存储资源的行业，开发应用程序专用协处理器极为重要。想象一下金融市场，以及那些在不同地点传输的、令人难以置信的贸易、预测和价格计算数据。在这些交易中，一分钱的几分之几都有重要意义，因此高速、精确的浮点算术绝对必要。

在高性能计算里，浮点是数字表示法，用一系列数字或数位表示真实数字。应用程序需要浮点数据类型，以获得比整数计算更为精确的计算结果。浮点运算需要更多的处理器逻辑，因此也需要更多能力。常见的浮点应用包括：

- 快速傅里叶变换（FFT）
- 雷达
- 生物科学
- 有限冲击响应滤波器（FIR）
- 金融期权交易
- 矩阵数学（在三维图形和图像处理上广泛使用）
- 分子动力学
- 地震成像和医学成像

“协处理器”是用于补充主处理器或中央处理器（CPU）的计算机处理器。协处理器通常被用于执行浮点计算、信号处理、字符串处理、编码或与外围设备的 I/O 接口。协处理器负责密集型运算，从而解放了 CPU，使 CPU 能够专注于计算机的核心功能。

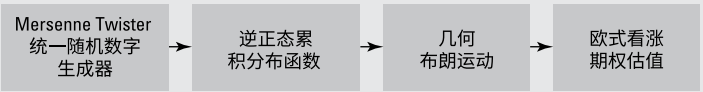
所有高性能计算市场都需要高产能、高性能和低功耗的协处理器。

好消息是，最新的英特尔 FPGA 不仅内嵌 DSP 功能，而且内嵌浮点计算是硬浮点。所以，程序员们不必将程序从浮点格式转成整数格式，就可在 FPGA 加速服务器上运行。这种能力在以下几个方面大有可为：

- **数据应用：**数据库和金融市场提速
- **功能组件：**供金融市场使用的随机数字发生器，供军事和信号处理应用使用的百万点 FET 晶体管
- **运算法则：**SRCs CARTE、Impulse 以及 AutoESL 系统生成算法

案例研究：蒙特卡洛与布莱克斯科尔斯计价法

在金融市场，最重要的基准之一是通过蒙特卡洛与布莱克斯科尔斯计价法计算的期权价格。在金融术语里，“期权”是指一个合约，该合约授予买家在给定日期之时或之前，按照特定价格购买或出售一定资产的权利（而非义务）。作为蒙特卡洛与布莱克斯科尔斯计价法，其计价基础是随机模拟基础股价，以及对数百万不同通道上的预期收益求取的平均值。附图为该计价法的框图。



感谢英特尔公司提供。

在多数计算机模拟系统中，设计师都要使用某种随机数字发生器，用于模拟现实系统的随机性，并将模拟数据输入模拟系统。在蒙特卡洛与布莱克斯科尔斯计价法中，模型制作者通常使用的是一种被叫作“梅森龙卷风”（Mersenne）的随机发生器。“梅森龙卷风”是一种非常快速且高质量的伪随机数字发生器——模拟系统的理想之选。相关随机数字序列被输入逆正态累积密度函数（一种用于指定随机数字分布的概率函数），以生成正态分布数字序列。利用这些随机数字，通过几何布朗运动（一种常用的股价预测算法）模拟股价运动行为。在每次模拟通道结束时，将记录看涨期权收益，并且计算平均值，以生成预期收益价值。整个算法可用大约 300 行 OpenCL 代码实现，因此可将该算法从 FPGA 搬运到 CPU 和 GPU。如附图所示，在功耗、性能和效率上，FPGA 解决方案都超过 CPU 和 GPU。

在该图中，FPGA 与 CPU 和 GPU 的比较标准有三个：功耗、每秒模拟次数，以及每秒功率－有效模拟的比例。FPGA 本身即是平行计算——这意味着可通过编程，将复杂计算变成平行计算。FPGA

平台	瓦特功率 (W)	每秒性能模拟次数 (Bsims/s)	每秒·瓦特效率模拟次数 (Msims/s/W)
CPU	130	0.032	0.0025
GPU	212	10.1	48
FPGA 解决方案	45	12.0	266

感谢英特尔公司提供。

能够完成的平行计算数量多于 CPU 和 GPU，因此可提高执行速度和功率效率——这两项优势恰好是电子系统设计的核心。

在系统设计上，加入 FPGA 的力量与灵活性

作为工程师的您是否对 FPGA 有所耳闻？或许您在大学学过 FPGA，只是已经生疏？不必紧张！有了这本“及时雨”，您会在进行 FPGA 系统设计时胸有成竹。

- *认识 FPGA 以及 FPGA 的工作方式——发现 FPGA 的本质和原理*
- *检查可编程组构——发现 FPGA 的可编程组构，以及 FPGA 的设计和开发方法*
- *将 FPGA 作为一个系统功能块——看到您的系统被吸收、合并成为 FPGA 片上系统*
- *认识异构计算的世界——发现异构计算正在改变系统的开发和通信方式*

打开本书，您将看到：

- FPGA 的现代设计流程
- FPGA 设计的长处和短处
- FPGA 系统设计在多个行业的应用，包括军事和汽车行业
- FPGA 系统设计的未来，包括异构计算和 OpenCL

访问网站 Dummies.com®，即可获得视频、分步演示和实操文章，还可在线购物！

安德鲁·摩尔，软件工程师，拥有超过15年软件设计经验。罗恩·威尔森，英特尔公司宣传策略家，拥有工程和技术杂志行业背景。

WILEY

ISBN 978-1-119-43732-1

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.