第1章

# 介绍

● ARM Cortex-M3处理器初探

● ARM的各种架构版本

●指令集的开发

● Thumb-2指令集架构(ISA)

● Cortex-M3的舞台

●本书组织

●深入研究用的读物

## 1.1 ARM Cortex-M3 处理器初探

单片机市场的规模可以用“巨无霸”来形容，预计到2010时每年能有20G片的出货量。世界各地的器件供应商纷纷亮出自己的得意之作，他们提供的器件和架构也是各具特色。业界内部可谓是百花齐放，热闹非凡，好戏不断。各行各业对单片机能力的要求也一直“得寸进尺”，而且还又要马儿跑，又要马儿不吃草——处理器必须在不怎么增加主频和功耗的条件下干更多的活儿。另一方面，处理器之间的互连也在加深，看这一串串熟悉的字眼：串口，USB，以太网，无线数传……处理器如欲支持这些数据通道，就必须在片上塞进更多的外设。软件方面的情况也如出一辙：应用程序的功能一直在花样翻新，性能需求也是变本加厉：更高的运算速度，更硬的实时能力，更多的功能模块，更炫的图形界面，……所有这些要求单片机都得照单全收。在这个大环境下，ARM Cortex-M3处理器，作为Cortex系列的处女作，为了让32位处理器入主作庄单片机市场，轰轰烈烈地诞生了！由于采用了最新的设计技术，它的门数更低，性能却更强。许多曾经只能求助于高级32位处理器或DSP的软件设计，都能在CM3上跑得很快很欢。嵌入式处理器市场正在32位化，相信用不了多久，CM3就一定会在这美丽新世界中脱颖而出。比当年8051推动整个业界还有过之而无不及，再次放飞工程师们的梦想，让深埋于心底多年的夙愿迎来dreams come true的激动！

CM3的招牌功夫包括：

•性能强劲。在相同的主频下能做处理更多的任务，全力支持劲爆的程序设计。

• 功耗低。延长了电池的寿命——这简直就是便携式设备的命门（如无线网络应用）。

• 实时性好。采用了很前卫甚至革命性的设计理念，使它能极速地响应中断，而且响应中断所需的周期数是确定的。

• 代码密度得到很大改善。一方面力挺大型应用程序，另一方面为低成本设计而省吃俭用。

• 使用更方便。现在从8位/16位处理器转到32位处理器之风刮得越来越猛，更简单的编程模型和更透彻的调试系统，为与时俱进的人们大大减负。

• 低成本的整体解决方案。让32位系统比和8位/16位的还便宜，低端的Cortex-M3单片机甚至还卖不到1美元。

• 遍地开花的优秀开发工具。免费的，便宜的，全能的，要什么有什么。

基于Cortex-M3内核的处理器已渐成气候，以处处满溢的先进特性力压群芳。而且架构师们还在不停地求索降低成本的出路，同时很多组织也在尝试着实现“器件聚合”（device aggregation），使一个单一的小强芯片可以抵得上以前3、4块传统的单片机。

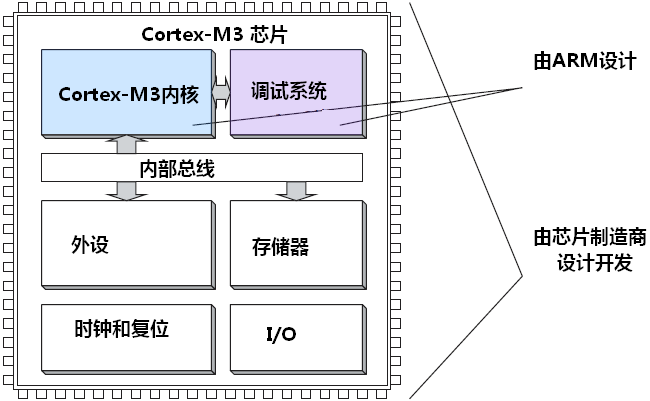
降低成本还有一招，就是使基础代码在所有系统中都可以重用，至少要方便移植。CM3的内核架构非常精工细作，使它与C语言成为了一个梦幻绝配。优质的C程序代码三下五除二就可以移植并重用，使升级和移植一下子从拦路虎变成了纸老虎。

值得一提的是，CM3并不是第一个被拿去做万金油型处理器的内核。那廉颇虽老却依然骁勇的ARM7/ARM9处理器，在通用嵌入式处理器市场中德高望重，至今拥有无数铁杆粉丝。半导体业界的群英们，像NXP（philips）、TI、Atmel、OKI、ST等，都以ARM为内核，做出了各自身怀绝技的32位MCU。ARM7作为最受欢迎的32位嵌入式处理器，被载入了亮煌煌的几页史册——每年超过10亿片出货量，为各行各业的嵌入式设备中都找得到它们的身影。

CM3作为ARM7的后继者，大刀阔斧地改革了设计架构。从而显著地简化了编程和调试的复杂度，处理能力也更加强大。除此之外，CM3还突破性地引入了很多时尚的甚至崭新的技术，专门满足单片机应用程序的需求。比如，服务于“使命-关键”应用的不可屏蔽中断，极度敏捷并且拥有确定性的嵌套向量中断系统，原子性质的位操作，还有一个可选的内存保护单元。这些令人惊艳和振奋的新特性，让老的ARM玩家们再次找到“初恋”时烈焰迸发的感觉，也使萍水相逢就有激爽触电般的体验！相信读者一旦有机会用到了它，就会为它的秀外慧中而赞叹，爱不释手！

### 1.1.1 从Cortex-M3 处理器内核到基于Cortex-M3的MCU

Cortex-M3处理器内核是单片机的中央处理单元（CPU）。完整的基于CM3的MCU还需要很多其它组件。在芯片制造商得到CM3处理器内核的使用授权后，它们就可以把CM3内核用在自己的硅片设计中，添加存储器，外设，I/O以及其它功能块。不同厂家设计出的单片机会有不同的配置，包括存储器容量、类型、外设等都各具特色。本书主讲处理器内核本身。如果想要了解某个具体型号的处理器，还需查阅相关厂家提供的文档。



### 1.1.2 ARM及ARM架构的背景

#### 1.1.2.1 一路走来

让我们回顿一下ARM的进化史，你会知道为什么会有品种如此之多的ARM处理器和ARM架构。

ARM在1990年成立，当初的名字是“Advanced RISC Machines Ltd.,”，当时它是三家公司的合资——它们分别是苹果电脑，Acorn电脑公司，以及VLSI技术（公司）。在1991年，ARM推出了ARM6处理器家族，VLSI则是第一个吃螃蟹的人。后来，陆续有其它巨头：包括TI, NEC, Sharp, ST等，都获取了ARM授权，它们真正地把ARM处理器大面积地辅开，使得ARM处理器在手机，硬盘控制器，PDA，家庭娱乐系统以及其它消费电子中都大展雄才。

现如今，ARM芯片的出货量每年都比上一年多20亿片以上。不像很多其它的半导体公司，ARM从不制造和销售具体的处理器芯片。取而代之的，是ARM把处理器的设计授权给相关的商务合作伙伴，让他们去根据自己的强项设计具体的芯片，这些伙伴可都是巨头啊。基于ARM低成本和高效的处理器设计方案，得到授权的厂商生产了多种多样的的处理器、单片机以及片上系统(SoC)。这种商业模式就是所谓的“知识产权授权”。

除了设计处理器，ARM也设计系统级IP和软件IP。为了挺它们，ARM开发了许多配套的基础开发工具、硬件以及软件产品。使用这些工具，合作伙伴可以更加舒心地开发他们自己的产品。

## 1.2 ARM的各种架构版本

ARM十几年如一日地开发新的处理器内核和系统功能块。这些包括流行的ARM7TDMI处理器，还有更新的高档产品ARM1176TZ(F)-S处理器，后者能拿去做高档手机。功能的不断进化，处理水平的持续提高，年深日久造就了一系列的ARM架构。要说明的是，架构版本号和名字中的数字并不是一码事。比如，ARM7TDMI是基于ARMv4T架构的（T表示支持“Thumb指令”）；ARMv5TE架构则是伴随着ARM9E处理器家族亮相的。ARM9E家族成员包括ARM926E-S和ARM946E-S。ARMv5TE架构添加了“服务于多媒体应用增强的DSP指令”。

后来又出了ARM11，ARM11是基于ARMv6架构建成的。基于ARMv6架构的处理器包括ARM1136J(F)-S，ARM1156T2(F)-S，以及ARM1176JZ(F)-S。ARMv6是ARM进化史上的一个重要里程碑：从那时候起，许多突破性的新技术被引进，存储器系统加入了很多的崭新的特性，单指令流多数据流（SIMD）指令也是从v6开始首次引入的。而最前卫的新技术，就是经过优化的Thumb-2指令集，它专为低成本的单片机及汽车组件市场。

ARMv6的设计中还有另一个重大的决定：虽然这个架构要能上能下，从最低端的MCU到最高端的“应用处理器”都通吃，但不能因此就这也会，那也会，但就是都不精。仍须定位准确，使处理器的架构能胜任每个应用领域。结果就是，要使ARMv6能够灵活地配置和剪裁。对于成本敏感市场，要设计一个低门数的架构，让她有极强的确定性；另一方面，在高端市场上，不管是要有丰富功能的还是要有高性能的，都要有拿得出手的好东西。

最近的几年，基于从ARMv6开始的新设计理念，ARM进一步扩展了它的CPU设计，成果就是ARMv7架构的闪亮登场。在这个版本中，内核架构首次从单一款式变成3种款式。

* 款式A：设计用于高性能的“开放应用平台”——越来越接近电脑了
* 款式R：用于高端的嵌入式系统，尤其是那些带有实时要求的——又要快又要实时。
* 款式M：用于深度嵌入的，单片机风格的系统中——本书的主角。

让我们再进距离地考察这3种款式：

* 款式A（ARMv7-A）：需要运行复杂应用程序的“应用处理器”[译注1]。支持大型嵌入式操作系统（不一定实时——译注），比如Symbian（诺基亚智能手机用），Linux，以及微软的Windows CE和智能手机操作系统Windows Mobile。这些应用需要劲爆的处理性能，并且需要硬件MMU实现的完整而强大的虚拟内存机制，还基本上会配有Java支持，有时还要求一个安全程序执行环境（用于电子商务——译注）。典型的产品包括高端手机和手持仪器，电子钱包以及金融事务处理机。

［译注1］：这里的“应用”尤指大型应用程序，像办公软件，导航软件，网页浏览器等。这些软件的使用习惯和开发模式都很像PC上的软件，但是基本上没有实时要求。

* 款式R（ARMv7-R）：硬实时且高性能的处理器。标的是高端实时［注1］市场。那些高级的玩意，像高档轿车的组件，大型发电机控制器，机器手臂控制器等，它们使用的处理器不但要很好很强大，还要极其可靠，对事件的反应也要极其敏捷。
* 款式M（ARMv7-M）：认准了旧世代单片机的应用而量身定制。在这些应用中，尤其是对于实时控制系统，低成本、低功耗、极速中断反应以及高处理效率，都是至关重要的。

Cortex系列是v7架构的第一次亮相，其中Cortex-M3就是按款式M设计的。

［注1］：通用处理器能否胜任实时系统的控制，常遭受质疑，并且在这方面的争论从没停止过。从定义的角度讲，“实时”就是指系统必须在给定的死线（deadline，亦称作“最后期限”）内做出响应。在一个以ARM处理器为核心的系统中，决定能否达到“实时”这个目标的，有很多因素，包括是否使用“实时操作系统”，中断延迟，存储器延时，以及当时处理器是否在运行更高优先级的中断服务例程。

本书认准了Cortex-M3就一猛子扎下去。到目前为止，Cortex-M3也是款式M中被抚养成人的独苗。其它Cortex家族的处理器包括款式A的Cortex-A8（应用处理器），款式R的Cortex-R4（实时处理器）。

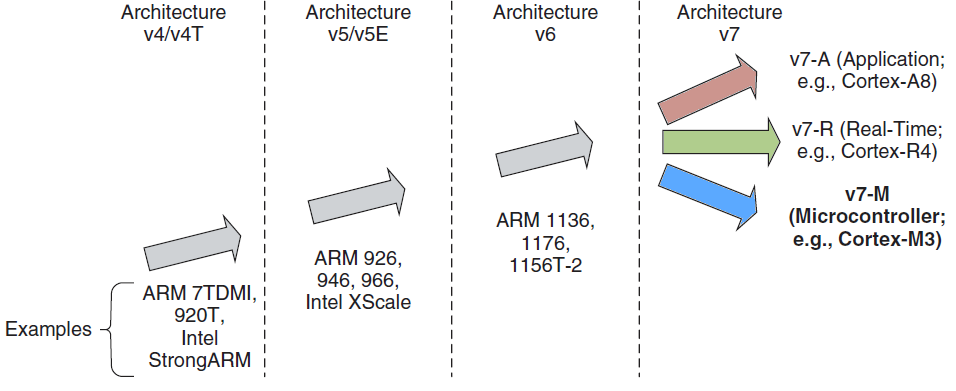


图1.2 ARM处理器架构进化史

ARMv7-M的私房秘密都记录在《The ARMv7-M Architecture Application Level Reference Manual》中（本书也讲了很多“System Level”的内容——译注），ARM已经将其公开。《Cortex M3 Technical Reference Manual》中则记录了实现v7-M时的很多细节和花絮。又因为v7M中列出的指令有一些是可选的，而CM3裁掉了一部分，所以在这个文档中重新列出了被CM3支持的指令集。

### 1.2.1 处理器命名法

以前，ARM使用一种基于数字的命名法。在早期（1990s），还在数字后面添加字母后缀，用来进一步明细该处理器支持的特性。就拿ARM7TDMI来说，T代表Thumb指令集，D是说支持JTAG调试(Debugging)，M意指快速乘法器，I则对应一个嵌入式ICE模块。后来，这4项基本功能成了任何新产品的标配，于是就不再使用这4个后缀——相当于默许了。但是新的后缀不断加入，包括定义存储器接口的，定义高速缓存的，以及定义“紧耦合存储器（TCM）”的，于是形成了新一套命名法，这套命名法也是一直在使用的。

表1.1 ARM处理器名字

|  |  |  |  |
| --- | --- | --- | --- |
| 处理器名字 | 架构版本号 | 存储器管理特性 | 其它特性 |
| ARM7TDMI | v4T |  |  |
| ARM7TDMI-S | v4T |  |  |
| ARM7EJ-S | v5E |  | DSP,Jazelle**[译注3]** |
| ARM920T | v4T | MMU |  |
| ARM922T | v4T | MMU |  |
| ARM926EJ-S | v5E | MMU | DSP,Jazelle |
| ARM946E-S | v5E | MPU | DSP |
| ARM966E-S | v5E |  | DSP |
| ARM968E-S | v5E |  | DMA,DSP |
| ARM966HS | v5E | MPU（可选） | DSP |
| ARM1020E | v5E | MMU | DSP |
| ARM1022E | v5E | MMU | DSP |
| ARM1026EJ-S | v5E | MMU 或 MPU**[译注2]** | DSP, Jazelle |
| ARM1136J(F)-S | v6 | MMU | DSP, Jazelle |
| ARM1176JZ(F)-S | v6 | MMU+TrustZone | DSP, Jazelle |
| ARM11 MPCore | v6 | MMU+多处理器缓存支持 | DSP |
| ARM1156T2(F)-S | v6 | MPU | DSP |
| Cortex-M3 | v7-M | MPU（可选） | NVIC |
| Cortex-R4 | v7-R | MPU | DSP |
| Cortex-R4F | v7-R | MPU | DSP+浮点运算 |
| Cortex-A8 | v7-A | MMU+TrustZone | DSP, Jazelle |

［译注2］：Jazelle是ARM处理器的硬件Java加速器。

［译注3］：MMU，存储器管理单元，用于实现虚拟内存和内存的分区保护，这是应用处理器与嵌入式处理器的分水岭。电脑和数码产品所使用的处理器几乎清一色地都带MMU。但是MMU也引入了不确定性，这有时是嵌入式领域——尤其是实时系统不可接受的。然而对于安全关键（safety-critical）的嵌入式系统，还是不能没有内存的分区保护的。为解决矛盾，于是就有了MPU。可以把MPU认为是MMU的功能子集，它只支持分区保护，不支持具有“定位决定性”的虚拟内存机制。

到了架构7时代，ARM改革了一度使用的，冗长的、需要“解码”的数字命名法，转到另一种看起来比较整齐的命名法。比如，ARMv7的三个款式都以Cortex作为主名。这不仅更加澄清并且“精装”了所使用的ARM架构，也避免了新手对架构号和系列号的混淆。例如，ARM7TDMI并不是一款ARMv7的产品，而是辉煌起点——v4T架构的产品。

## 1.3 指令集的发展

为了增强和扩展指令系统的能力而奋斗，多少年来这一直是ARM锲而不舍的精神动力。

由于历史原因（从ARM7TDMI开始），ARM处理器一直支持两种形式上相对独立的指令集，它们分别是：

* 32位的ARM指令集。对应处理器状态：ARM状态
* 16位的Thumb指令集。对应处理器状态：Thumb状态

可见，这两种指令集也对应了两种处理器执行状态。在程序的执行过程中，处理器可以动态地在两种执行状态之中切换。实际上，Thumb指令集在功能上是ARM指令集的一个子集，但它能带来更高的代码密度，给目标代码减肥。这对于要勒紧裤腰带的应用还是很经济的。

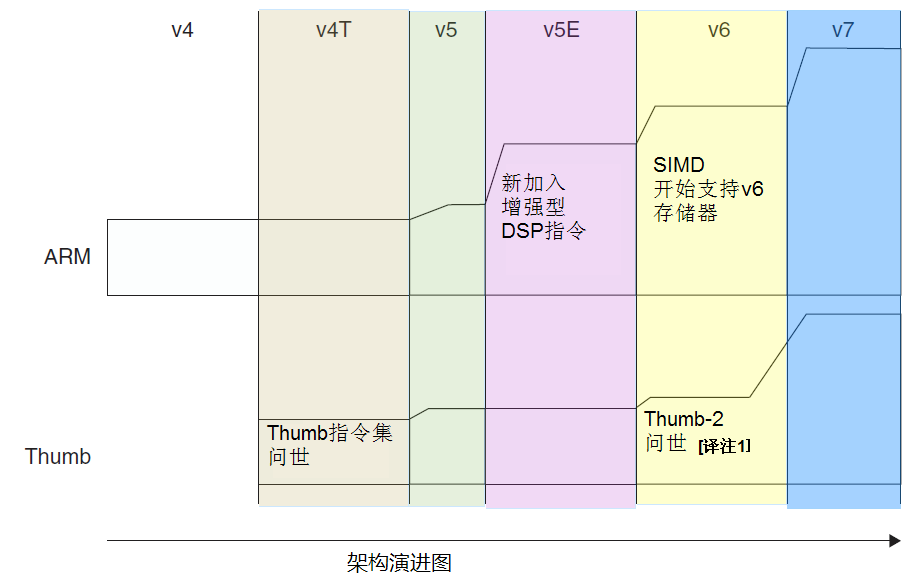


图1.3 指令集演进图

译注1: 原书把Thumb-2的问世时间放到v7中，但根据其它权威文献的记录，似有误，应在v6中问世。（如《ARM and Thumb-2 Instruction Set Quick Reference Card》中的描述）

随着架构版本号的更新，新好指令不断地加入ARM和Thumb指令集中。附录2中给出的内容，就是Thumb指令在架构进化过程中的改变记录。Thumb-2是2003年盛夏的果实，它是Thumb的超集，它支持both 16位和32位指令。

指令集的详细说明在《The ARM Architecture Reference Manual》（简称为ARMARM）中。每次ARM出新版本时此手册都有更新。到了v7时，因为以前的单一架构被分成了3个款式，这个规格书也就跟着变成了3本。为Corex-M3的ARMv7-M架构而写的那本叫《ARMv7-M Architecture Application Level Reference Manual(Ref2)》，对于软件开发人员，那里面把该说的都说了。

## 1.4 Thumb-2指令集体系体系结构（ISA）

Thumb-2真不愧是一个突破性的指令集。它强大，它易用，它轻佻，它高效。Thumb-2是16位Thumb指令集的一个超集，在Thumb-2中，16位指令首次与32位指令并存，结果在Thumb状态下可以做的事情一下子丰富了许多，同样工作需要的指令周期数也明显下降。

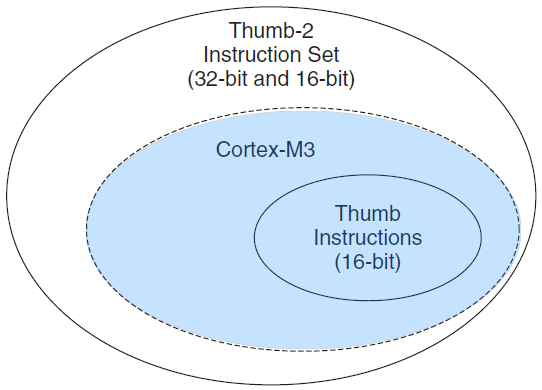


图1.4 Thumb-2指令集与Thumb指令集的关系

从图中可见，Cortex-M3勇敢地拒绝了32位ARM指令集，却把自己的处理能力以身相许般地全托给Thumb-2指令集。这可能有些令人意外，但事实上这却见证了Cortex-M3的用情专一：在内核水平上，就已经为适应单片机和小内存器件而抉择、取舍过了。但她没有嫁错郎， 因为Thumb-2完全胜任在这个领域挑大梁。不过，这也意味着Cortex-M3作为新生代处理器，不是向后兼容的。因此，为ARM7写的ARM汇编语言程序不能直接移植到CM3上来。不过，CM3支持绝大多数传统的Thumb指令，因此用Thumb指令写的汇编程序就从善如流了。

在支持了both 16位和32位指令之后，就无需烦心地把处理器状态在Thumb和ARM之间来回的切换了。这种事在ARM7和ARM9是司空见惯的，尤其是在使用大型条件嵌套，以及执行复杂运算的时候，能精妙地移形换影于不同状态之间，那可是当年要成为大虾的基本功。

Cortex-M3是ARMv7架构的掌上明珠。和曾经红透整个业界的老一辈ARM7相比，Cortex-M3则是新生代的偶像，处处闪耀着青春的光芒活力。比如，硬件除法器被带到CM3中；乘法方面，也有好几条新指令闪亮登场，用于提升data-crunching的性能。CM3的出现，还在ARM处理器中破天荒地支持了“非对齐数据访问支持”。

## 1.5 Cortex-M3处理器的舞台

高性能＋高代码密度＋小硅片面积，3璧合一，使得CM3大面积地成为理想的处理平台：

* 低成本单片机：CM3与生俱来就适合做单片机，甚至简单到用于做玩具和小电器的单片机，都能使用CM3作为内核。这里本是8位机和16位机统治最牢固的腹地，但是CM3更便宜，更高性能，更易使用，所以值得开发者们转到这个新生的ARM32位系统中来，哪怕花点时间重新学习。
* 汽车电子：CM3也是汽车电子的好俅。CM3同时拥有非常高的性能和极低的中断延迟，打入实时领域的大门。CM3处理器能支持多达240个外部中断，内建了嵌套向量中断控制器，还可以选择配上一个存储器保护单元（MPU）。所有这些，使它用于高集成度低成本的汽车应用最合适不过了。
* 数据通信：CM3的低成本＋高效率，再加上Thumb-2的强大位操作指令s，使CM3非常理想地适合于很多数据通信应用，尤其是无线数传和Ad-Hoc网络，如ZigBee和蓝牙等。
* 工业控制：在工控场合，关键的要素在于简洁、快速响应以及可靠。再一次地，CM3处理器的中断处理能力，低中断延迟，强化的故障处理能力（fault-handing，以后fault就不再译成中文了——译注），足以让它能昂首挺胸地踏入这片热土。
* 消费类产品：以往，在许多消费产品中，都必须使用一块甚至好几块高性能的微处理器。你别看CM3只是个小处理器，它的高性能和MPU机制可是足以让复杂的软件跑起来的，同时提供健壮的存储器保护。

目前在市场上已经有了好多基于Cortex-M3内核的处理器产品，最便宜的还不到1美元，让ARM终于比很多8位机还便宜了。

本书的组织

Chpt 1和2， Cortex-M3的介绍和概览

Chpt 3-6， Cortex-M3的基础知识

Chpt 7-9， 异常与中断

Chpt 10和11， 论述在Cortex-M3的编程

Chpt 12-14， Cortex-M3的硬件特性

Chpt 15-16， Cortex-M3的调试支持

Chpt 17-20， 在Cortex-M3上的应用软件开发

附录s

## 1.6 深入研究用的读物

本书并没有面面俱到地谈及Cortex-M3的技术细节。本书靠前的章节用来做Cortex-M3新手的敲门砖，同时也是CM3处理器的增值参考资料。如果要进一步地学习，就需要从ARM网站下载下面这些重量级的权威资料：

《The Cortex-M3 Technical Reference Manual》，深入了处理器的内心，编程模型，存储器映射，还包括了指令时序。

《 The ARMv7-M Architecture Application Level Reference Manual》第2版，对指令集和存储器模型都提供了最不嫌繁的说明。

其它半导体厂家提供的，基于CM3单片机的数据手册。

如想了解更多总线协议的细节，可以去看《AMBA Specification 2.0》（第4版），它讲了更多AMBA接口的内幕。

对于C程序员，可以从《ARM Application Note 179: Cortex-M3 Embedded Software Development》（第7版）中得到一些编程技巧和提示。

本书假设你已经涉足过嵌入式编程，有一些基本知识和经验。如果你是位产品经理或者是想先浅浅地尝一尝，请先读第2章，试着找找感觉再决定要不要深入学习。这一章浓缩了全书的精华，走马观花地讲了Cortex-M3内核。