

跟涛哥一起学嵌入式 09：ARM启动过程中你不知道的小秘密

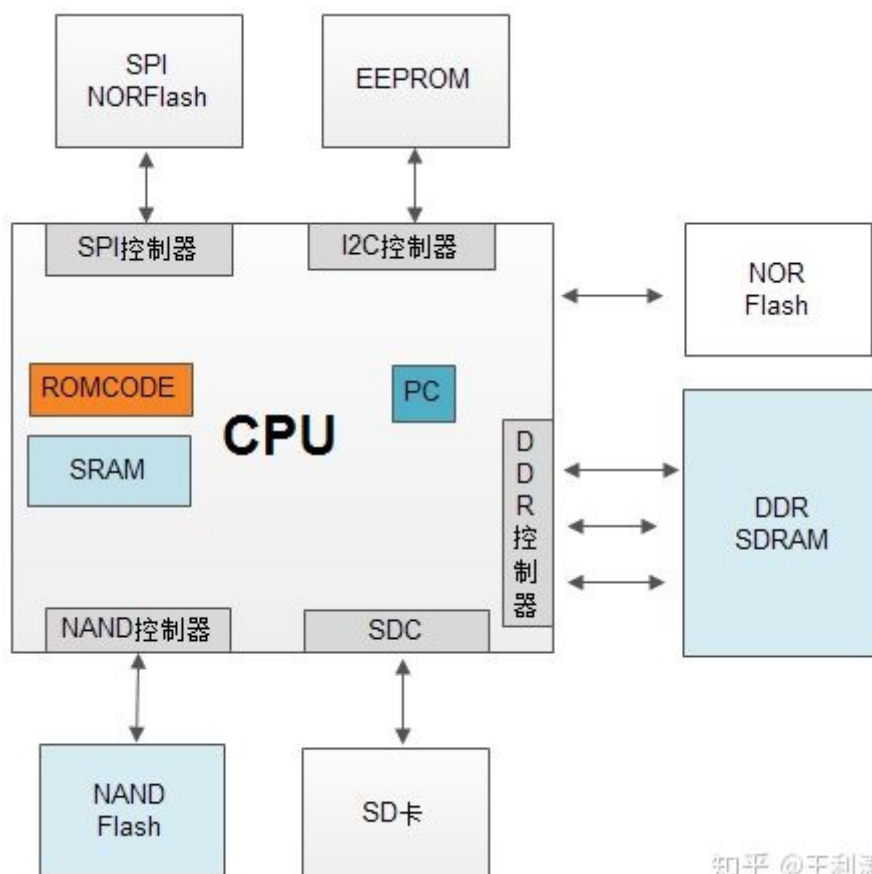
文档说明	作者	日期
来自微信公众号：宅学部落(armLinuxfun)	wit	2018.8.14
嵌入式视频教程淘宝店： https://wanglitao.taobao.com/		
联系微信：brotau(宅学部落)		

学过嵌入式、玩过开发板的同学可能都知道这些常识：我们的ARM开发板有多种启动方式，可以从 NOR Flash启动、可以从 NAND Flash 启动、也可以从 SD 卡启动。当系统上电后，ARM 会从 U-boot 启动，然后 U-boot将自身代码拷贝到内存 DDR SDRAM 中，接着 PC 指针指向内存，从内存中不断地取指令、翻译指令、运行指令。

然而，现实果真如此吗？U-boot 是系统上电运行的第一行代码吗？ARM 支持多种启动方式是如何实现的？

很多资料或者手册上可能都没有提及到这个话题，或者一笔带过。今天就给大家扒一扒芯片启动过程中的那些小秘密。

其实，在现代 ARM处 理器中，U-boot 并不是系统上电运行的第一行代码。一般在芯片 SOC 内部，会集成一片ROM，在 ROM 上会固化一段启动代码，如下图：



系统上电后，会首先运行固化在芯片内部的这部分 ROMCODE 代码。这部分代码的主要工作就是初始化存储接口、建立存储映射。它首先会根据 CPU 外部管脚或 efuse 值来判断系统的启动方式：NOR、NAND 还是 SD 卡启动。

如果我们设置系统从 NOR Flash 启动，这段代码就会将 NOR Flash 映射为零地址，然后系统复位，跳转到 U-boot 的中断向量表中的第一行代码，即 NORFlash 中的第一行代码去执行。

如果我们设置系统从 NAND Flash 或 SD 卡启动，我们知道除了 SDRAM 和 NOR Flash 支持随机读写，可以直接运行代码外，其它的Flash设备是不支持直接运行代码的，只能将代码拷贝到内存中执行。因为此时 DDR 内存还没有初始化，所以一般会先将 NAND Flash 或 SD 卡中的一部分代码(前4K)拷贝到芯片内部的 SRAM 中去执行，然后在这4K代码中再进行各种初始化、代码拷贝等工作，最后系统才会跳到 SDRAM 内存中去执行代码。

由此可见，CPU 并不是万能的，它里面除了一些寄存器、逻辑算术运算单元，也没有什么复杂的逻辑了。很多不同的启动方式需要通过CPU管脚来设置，需要我们自己写程序进行判断。这些程序写好后是固化在芯片内部的，出厂时固化在芯片内部，无法更改。因此，这部分启动流程相对于芯片使用者、开发者来说是无须关心的，我们只需要按照芯片手册设置自己的启动方式就可以了。

小结

本文主要给大家科普了 ARM 启动过程中，系统上电复位之前的一点小秘密：包括系统启动模式选择、存储映射、代码拷贝等一些基本流程。这些准备工作做好之后，才会发生系统复位，跳转到我们ARM的中断向量表去执行。通过本节学习，相信大家对嵌入式系统的整体启动流程的理解会有一个更完整的认识。

本文根据《C语言嵌入式Linux高级编程》第 9 期：CPU 和操作系统入门，第 9.14 节改编，关注嵌入式学习，视频教程，可以关注微信公众号：armlinuxfun（宅学部落）或QQ群：475504428

专注嵌入式、Linux精品教程：<https://wanglitao.taobao.com/>

嵌入式技术教程博客：<http://zhaixue.cc/>

联系 QQ：3284757626

嵌入式技术交流QQ群：475504428

微信公众号：宅学部落(armlinuxfun)

