



手把手教你学系列丛书

周兴华 倪敏娜 编著
周兴华单片机培训中心 策划

手把手

教你学ARM Cortex-M0

——基于LPC11XX系列



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

——基于 LPC11XX 系列

周兴华单片机培训中心 策划

北京航空航天大学出版社

内 容 简 介

本书以 NXP 公司的 LPC11XX 系列 ARM 处理器为例,从零开始,手把手地教初学者学习 ARM 设计知识,在介绍 LPC11XX 各单元基本特性的同时,使用入门难度浅、程序长度较短且又能立竿见影的初级实例,循序渐进地帮助初学者逐步掌握 ARM 的设计知识,实践为主,辅以理论。本书的实例均经作者实际测试并在实验板上正常运行,实用性非常强,读者既可以拿来直接在产品中使用,也可以进一步改良升级。

本书贯彻“手把手教你学”系列丛书相同的教学方式。本书可作为大学本科或专科、中高等职业技术学院学校、电视大学等的教学用书,也可作为 ARM 爱好者的入门自学用书。

图书在版编目(CIP)数据

手把手教你学 ARM Cortex-M0:基于 LPC11XX 系列 /
周兴华,倪敏娜编著. -- 北京:北京航空航天大学出版社, 2015.12

ISBN 978-7-5124-1969-8

I. ①手… II. ①周… ②倪… III. ①微处理器
IV. ①TP332

中国版本图书馆 CIP 数据核字(2015)第 290931 号

版权所有,侵权必究。

手把手教你学 ARM Cortex-M0——基于 LPC11XX 系列

周兴华 倪敏娜 编著

周兴华单片机培训中心 策划

责任编辑 张冀青

*

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:emsbook@buaacm.com.cn 邮购电话:(010)82316936

北京泽宇印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:33 字数:703 千字

2016 年 1 月第 1 版 2016 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1969-8 定价:79.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前言

借助于手机及其他掌上电子产品的普及推广,32 位微处理器 ARM 也因此迅速发展壮大。在世界范围内,ARM 处理器正在成为工程师设计移动产品的首选。

传统的 CISC(Complex Instruction Set Computer,复杂指令集计算机)体系由于指令集庞大,指令长度不固定,指令执行周期有长有短,使指令译码和流水线的实现在硬件上非常复杂,给芯片的设计开发和降低成本带来了很大的困难。

ARM 处理器采用了当今的处理器设计主流技术 RISC,RSIC 的英文全称为 Reduced Instruction Set Computer,即精简指令集计算机。RISC 技术把设计重点放在了如何使计算机的结构更加简单合理以及提高运算速度上。RISC 结构优先选取使用频率最高的简单指令,避免复杂指令,将指令长度固定,指令格式和寻址方式种类减少,以控制逻辑为主,不用或少用微码控制等措施来达到上述目的。因此 RISC 结构的处理器的指令较少,运行速度快,抗干扰能力强。

ARM 处理器目前以 Cortex 为前缀进行命名,而且每一个大的系列里又分为若干小的系列。

ARM Cortex 处理器采用全新的 ARMv7 架构,根据使用的对象不同,划分为 3 大系列:Cortex-A 系列、Cortex-R 系列、Cortex-M 系列。

在 Cortex-M 系列里,ARM 面向微处理器产业分别推出了 Cortex-M0、Cortex-M3、Cortex-M4 三款嵌入式处理器,大部分半导体厂商主要集中于 Cortex-M3 内核的生产。Cortex-M4 是 Cortex-M3 的升级版,较于 Cortex-M3 具备更高的信号处理能力;而 Cortex-M0 则是 Cortex-M3 的精简版,它以低价格(与 8 位单片机相当)进入市场,但是其超高的性能(每秒运行近 5 000 万次)是 8 位单片机无法企及的,因此 ARM 终结 8 位单片机指日可待。

ARM 处理器的应用领域很广,由于运行快、性能强、功耗低,以前 8051 单片机不能胜任的许多领域它均可一展身手,包含工业控制领域、无线通信领域、网络应用、消费类电子产品、成像和安全产品、移动互联网、3G 领域、科研及军事等。目前 ARM 微处理器约占据了 32 位 RISC 微处理器 75% 以上的市场份额。

本书以 NXP 公司的 LPC11XX 系列 ARM 处理器为例,从零开始,手把手教初学者学习 ARM 设计知识。在介绍 LPC11XX 各单元基本特性的同时,使用入门难度浅、程序长度较短且又能立竿见影的初级实例,循序渐进地帮助初学者逐步掌握 ARM 的设计知识,实践为主,辅以理论。本书的实例均经作者实际测试并在实验板

上正常运行,实用性非常强,读者既可以拿来直接在产品中使用,也可以进一步改良升级。

另外,现在 ARM 芯片及开发套件的价格已经降到了非常低的水平,并且开发软件的界面也非常友好,因此学习 ARM 的时代已经到来。

本书的编写工作得到了北京航空航天大学出版社相关领导的大力支持,在此表示衷心的感谢!

本书在编写过程中参考了相关书籍及网络的部分流通资料,在此一并致谢!

由于作者水平有限,书中必定还存在不少缺点或漏洞,诚挚欢迎广大读者提出意见并不吝赐教。

如果读者朋友需要书中介绍的学习器材或参加 ARM 的设计培训班学习,可与作者联系,联系方式如下:

地址:上海市闵行区莲花路 2151 弄 57 号 201 室

邮编:201103

电话(传真):(021)64654216 13774280345

技术支持:zxh2151@sohu.com zxh2151@163.com

培训中心主页:<http://www.hlelectron.com>

周兴华

2015 年 8 月



第 1 章 概 述	1
1.1 快速学会 ARM 处理器设计	2
1.2 使用 C 语言的优点	3
1.3 开发 LPC1114 使用的 C 编译器	4
第 2 章 ARM 的发展	5
2.1 什么是 ARM	5
2.2 处理器 RISC 技术简介	5
2.3 ARM 处理器	6
2.4 ARM 公司的优势及前景展望	12
第 3 章 ARM Cortex - M0 内核架构体系简介	14
3.1 LPC1114 结构和特性	15
3.2 LPC1114 存储器和外设地址映射	17
3.3 LPC1114 系统配置	17
3.4 LPC1114 中断控制	22
第 4 章 开发/实验工具及入门程序	25
4.1 CMSIS 标准简介	25
4.2 LPC1114 开发工具	28
4.3 LPC1114 实验工具	30
4.4 LPC1114 开发过程的文件管理及项目设置	33
4.5 第一个 LPC1114 入门程序	35
第 5 章 C 语言基础知识	46
5.1 标识符与关键字	46
5.2 数据类型	47
5.3 常量、变量及存储方式	48
5.4 数 组	49
5.5 运算符	52
5.6 流程控制	59
5.7 函 数	64
5.8 指 针	67

5.9	结构体	71
5.10	共用体	77
5.11	LPC11XX 开发中 C 语言的常用方法	79
5.12	中断函数	80
第 6 章	LPC11XX 引脚及系统时钟应用	82
6.1	LPC11XX 引脚功能	82
6.2	LPC11XX 系统时钟设置	87
6.3	LPC11XX 典型系统时钟设置程序	92
6.4	系统时钟应用实验——LPC1114 的 P0.1 引脚输出主时钟的信号频率	95
第 7 章	GPIO 特性及应用	97
7.1	GPIO 介绍	97
7.2	GPIO 寄存器	101
7.3	GPIO 寄存器设置	102
7.4	GPIO 应用实验——按键控制发光二极管的亮灭	102
第 8 章	LPC11XX 外中断应用设计	107
8.1	嵌套向量中断控制器	107
8.2	中断源	107
8.3	NVIC 控制函数	108
8.4	中断函数及写法	109
8.5	LPC11XX 外中断相关 GPIO 寄存器	109
8.6	LPC11XX 外中断相关 GPIO 寄存器设置	112
8.7	GPIO 外中断应用实验——外中断输入控制发光二极管的亮灭	112
第 9 章	系统节拍定时器特性及应用	115
9.1	系统节拍定时器相关寄存器	116
9.2	系统节拍定时器应用实验——精确延时	117
第 10 章	TFT - LCD 的驱动显示	120
10.1	TFT - LCD 显示器	120
10.2	TFT - LCD 显示器模块的引脚功能	121
10.3	ILI9325/ILI9328 的几个重要寄存器及控制命令	122
10.4	TFT - LCD 显示的相关设置	126
10.5	TFT - LCD 应用实验——彩色液晶屏显示多种颜色及图形	126
第 11 章	字库制作及 TFT - LCD 的中英文显示	139
11.1	Flash 存储器 W25Q16	139
11.2	中英文显示的原理	157
11.3	编写生成 GBK_Proj. hex 应用程序的源代码	158

11.4	中文字库的下载	160
11.5	从 W25Q16 中提取点阵码函数及中英文显示驱动函数	163
11.6	TFT-LCD 应用实验——彩色液晶屏显示多种颜色及中英文字符	167
第 12 章	通用异步串口 UART 特性及应用	170
12.1	UART 相关寄存器	173
12.2	UART 应用实验——查询方式接收数据包	188
12.3	UART 应用实验——中断方式接收数据包	192
第 13 章	16 位计数器/定时器特性及应用	195
13.1	CT16B0/1 相关寄存器	197
13.2	CT16B0 定时中断实验——控制发光二极管闪烁	205
13.3	CT16B1 捕获中断实验——红外遥控信号接收解调	211
第 14 章	32 位计数器/定时器特性及应用	218
14.1	CT32B0/1 相关寄存器	220
14.2	CT32B0 定时查询实验——控制发光二极管闪烁	228
14.3	CT32B0 定时中断实验——控制发光二极管闪烁	235
14.4	CT32B0 匹配输出实验——匹配时翻转输出方波信号	237
14.5	CT32B0 PWM 输出实验——输出调宽脉冲信号	239
14.6	CT32B1 捕获实验——P1.0 跳变为低则捕获一次定时器的值	242
14.7	CT32B1 外部计数实验——P1.0 跳变为低一次则定时器的值增加 1	245
第 15 章	模数转换器特性及应用	248
15.1	时钟供应和功率控制	248
15.2	ADC 相关寄存器	249
15.3	ADC 转换及中断	253
15.4	ADC 应用实验	254
第 16 章	I2C 总线接口特性及应用	264
16.1	I2C 快速模式 Plus	265
16.2	I2C 总线接口相关寄存器	265
16.3	I2C 总线接口实验	273
第 17 章	SSP 总线特性及电阻式触摸屏应用	290
17.1	SSP 相关寄存器	291
17.2	电阻式触摸屏	297
17.3	低电压输入/输出触摸屏控制器 XPT2046	297
17.4	XPT2046 工作原理	299
17.5	XPT2046 的控制字	301
17.6	笔中断接触输出	303

17.7	触摸屏应用实验	303
第 18 章	看门狗定时器特性及应用	324
18.1	时钟和功率控制	325
18.2	WDT 相关寄存器	326
18.3	WDT 应用实验	328
第 19 章	2.4 GHz 无线收发模块 NRF24L01 特性及应用	336
19.1	NRF24L01 结构及引脚功能	336
19.2	NRF24L01 工作模式	338
19.3	NRF24L01 工作原理	338
19.4	NRF24L01 配置字	339
19.5	NRF24L01 通信实验	339
第 20 章	FatFS 文件系统及电子书实验	352
20.1	FatFS 文件系统分析	352
20.2	FatFS 文件系统移植	354
20.3	基于 FatFS 文件系统的 SD 卡实验	359
20.4	电子书阅读实验	377
第 21 章	电源管理特性及深度掉电与唤醒实验	388
21.1	运行模式	389
21.2	睡眠模式	389
21.3	深度睡眠模式	390
21.4	深度掉电模式	390
21.5	电源管理相关寄存器	391
21.6	进入深度掉电与唤醒实验	393
第 22 章	数码相框显示及 GUI 实验	396
22.1	数码相框的构成和图像文件的处理	396
22.2	数码相框设计实验	397
22.3	GUI 图形界面设计实验	400
第 23 章	Flash 存储器 W25Q16 的图片存取及显示实验	409
23.1	对图片取模生成二进制文件	409
23.2	将图片二进制文件发送到 W25Q16 中	410
23.3	DownLoad_PIC 源程序文件及分析	411
23.4	Show_PIC 图片读取及显示源程序文件	413
23.5	实验效果	414
第 24 章	RTX Kernel 实时操作系统	415
24.1	概述	415
24.2	RTX Kernel 实时操作系统的基本功能及进程间的通信	417

24.3	RTX Kernel 实时操作系统的任务管理	418
24.4	RTX Kernel 实时操作系统的库函数	421
第 25 章	RTX Kernel 实时操作系统实验	438
25.1	延时——时间间隔延迟实验	438
25.2	事件——信号标志发送/接收实验	440
25.3	邮箱——内存池及邮箱实验	452
25.4	互斥——互斥体实验	470
25.5	信号量——信号量的传送与接收实验	478
第 26 章	RTX Kernel 实时操作系统应用设计实践	482
26.1	文件系统实验	482
26.2	手写画板实验	489
26.3	数码相框实验	494
26.4	外部中断实验	498
26.5	用户定时器实验	504
26.6	循环定时器实验	507
26.7	综合实验	510
参考文献		516

ARM 处理器的应用领域很广, 除了工业控制、无线通信、网络应用、消费类电子产品、设备安全产品、医疗仪器、3G 等领域。

ARM 处理器优点是体积小、功耗低、成本低、性能高, 支持 Thumb16 位/ARM15 位双指令集, 能很好地兼容 8 位/16 位器件, 大量使用寄存器, 指令执行速度更快, 大多数数据操作都在寄存器中完成, 寻址方式简洁简单, 执行效率高, 指令集精简。

目前, 面向工业及控制领域的 ARM 嵌入式微处理器有 Cortex-M0、Cortex-M1、Cortex-M3 三款。Cortex-M0 处理器是市场上体积最小、功耗最低、性价比最高的 ARM 处理器。该处理器拥有超小的封装尺寸, 低功耗模式, 低功耗模式下的低功耗运行, 以及处理器的软件节能实现 32 位的性能, 节省了系统 18 位功耗的功耗。Cortex-M0 可大大节约系统成本, 同时保留了 Cortex-M3 等处理器丰富处理器的工具链和二进制指令集。它在不超过 12K 门的面积内, 功耗仅 50 μ W/MHz (0.05 μ W/MHz), 可支持创建超低功耗的硬件和混合信号器件。

NXP 公司的 LPC111X 是全球第一款基于 Cortex-M0 的芯片, 是市场定价最低的 ARM 微控制器之一。LPC111X 内部运行频率为 50 MHz, 具有低功耗待机模式、唤醒中断控制等中断能力, 并拥有睡眠、深度睡眠和深度待机三种模式。LPC111X 还配备了多达 128 KB 的闪存和 16 KB 的 SRAM, 另外, 还有 10 位 ADC 以及 UART、SPI 控制器、I2C 总线等串行接口。另外, 该微控制器还支持多达 48 个 GPIO 引脚, 4 个通用计数器、定时器, 集成的电源管理单元、时钟发生器等等。NXP 公司还

第 1 章

概 述

自从广大读者跟着“手把手教你学”系列丛书学习单片机设计后,由于教学方式新颖,入门难度低,已经有超过百万的读者学会或者基本学会了单片机的开发设计。

进入新世纪后,电子信息技术与计算机技术的发展尤其令人瞩目。当前,ARM 系列处理器的价格已经降到可与 8 位单片机媲美的水平,而其性能却超越 8 位机达好几代的程度。不管是手机还是其他工业控制产品,ARM 系列芯片的使用呈爆炸性增长。ARM 处理器的应用领域很广,包含工业控制、无线通信、网络应用、消费类电子产品、成像和安全产品、移动互联网、3G 等领域。

ARM 处理器优点是体积小、功耗低、成本低、性能高,支持 Thumb(16 位)/ARM(32 位)双指令集,能很好地兼容 8 位/16 位器件,大量使用寄存器,指令执行速度更快,大多数数据操作都在寄存器中完成,寻址方式灵活简单,执行效率高,指令长度固定。

目前,面向工业及控制领域的 ARM 嵌入式微处理器有 Cortex - M0、Cortex - M3、Cortex - M4 三款。Cortex - M0 处理器是市场上体积最小、功耗最低、最节能的 ARM 处理器。该处理器拥有超小的硅片面积、低功耗和最小的代码足迹,以 8 位处理器的价位就能实现 32 位的性能,省去了使用 16 位器件的步骤。Cortex - M0 可大大节约系统成本,同时保留了 Cortex - M3 等功能较丰富处理器的工具并与二进制兼容。它在不到 12K 门的面积内,功耗仅为 $85 \mu\text{W}/\text{MHz}$ (0.085 mW),可支持创建超低功耗的模拟和混合信号器件。

NXP 公司的 LPC11XX 是全球第一颗基于 Cortex - M0 的芯片,是市场定价最低的 ARM 控制器之一,LPC11XX 内核的运行频率为 50 MHz,具有嵌套向量中断控制、唤醒中断控制等中断能力,并拥有睡眠、深度睡眠和深度掉电三种模式。LPC11XX 还配备了多达 128 KB 的闪存和 16 KB 的 SRAM。另外,还有 10 位 ADC 以及 UART、SPI 控制器、I2C 总线等串行接口。其他外设则包括多达 42 个 GPIO 引脚、4 个通用计数器/定时器、集成的功率管理单元、时钟发生器等。NXP 公司还

将提供多种封装形式供客户按需选择。

基于以上的原因,读者迫切需要在新时代的背景下快速掌握新型 ARM 处理器的设计(尤其是性价比最高的 Cortex - M0 芯片设计)。在“手把手教你学”系列丛书出版后,曾经有很多读者给笔者来信来电,表示“手把手教你学”系列丛书的教学方式很适合他们学习,跟着该教程学习实验后,渐渐地就从不理解到了解、从不懂到学会单片机的设计了。因此他们从心底是非常喜欢“手把手教你学”系列丛书的。

因此,笔者在编写的 ARM M0 培训讲义的基础上,采用“手把手教你学”系列的编写风格,编写《手把手教你学 ARM Cortex - M0——基于 LPC111X 系列》一书,希望读者能快速、轻松地学会 ARM 处理器设计。

1.1 快速学会 ARM 处理器设计

快速学会 ARM 处理器设计最好的办法是采用 C 语言编程。

为了提高编制计算机系统和应用程序的效率,改善程序的可读性和可移植性,最好的办法是采用高级语言编程。目前,C 语言逐渐成为国内外开发单片机的主流语言。

C 语言是一种通用的编译型结构化计算机程序设计语言,在国际上十分流行,它兼顾了多种高级语言的特点,并具备汇编语言的功能。它支持当前程序设计中广泛采用的由顶向下的结构化程序设计技术。一般的高级语言难以实现汇编语言对于计算机硬件直接进行操作(如对内存地址的操作、移位操作等)的功能,而 C 语言既具有一般高级语言的特点,又能直接对计算机的硬件进行操作。C 语言有功能丰富的库函数、运算速度快、编译效率高,并且采用 C 语言编写的程序能够很容易地在不同类型的计算机之间进行移植。因此,C 语言的应用范围越来越广泛。

用 C 语言来编写目标系统软件,会大大缩短开发周期,且明显增加软件的可读性,便于改进和扩充,从而研制出规模更大、性能更完备的系统。

因此,用 C 语言进行嵌入式处理器程序设计是 ARM 开发与应用的必然趋势。采用 C 语言进行设计也不必对 ARM 芯片和硬件接口的结构有很深入的了解,编译器可以自动完成变量存储单元的分配,编程者就可以专注于应用软件部分的设计,大大加快了软件的开发速度。采用 C 语言可以很容易地进行嵌入式处理器的程序移植工作,有利于产品中嵌入式处理器的重新选型。

C 语言的模块化程序结构特点,可以使程序模块共享,不断丰富。C 语言可读性的特点,更容易借鉴前人的开发经验,提高自己的软件设计水平。采用 C 语言,可针对常用的接口芯片编制通用的驱动函数,可针对常用的功能模块、算法等编制相应的函数。这些函数经过归纳整理可形成专家库函数,供广大的工程技术人员和爱好者使用、完善,也可大大提高国内嵌入式控制器软件设计的水平。

1.2 使用 C 语言的优点

1. 语言简洁,使用方便

C 语言是现有程序设计语言中规模最小的语言之一,而小的语言体系往往能设计出较好的程序。C 语言的关键字很少,ANSI C 标准一共只有 32 个关键字、9 种控制语句,压缩了一切不必要的成分。C 语言的书写形式比较自由,表达方法简洁,使用一些简单的方法就可以构造出相当复杂的数据类型和程序结构。

2. 可移植性好

用过汇编语言的读者都知道,即使是功能完全相同的一种程序,不同的嵌入式处理器也必须采用不同的汇编语言来编写。这是因为汇编语言完全依赖于芯片硬件,而新器件的更新换代速度非常快,我们每年都要跟新的处理器打交道。如果每接触一种新的处理器就要学习一次新的汇编语言,那么我们将一事无成,因为每学一种新的汇编语言,少则几个月,多则时间更长,那么我们还有多少时间真正用于产品开发呢?

C 语言是通过编译来得到可执行代码的。统计资料表明,不同机器上的 C 语言编译程序,80%的代码是公共的,而 C 语言的编译程序便于移植,在一种处理器上使用的 C 语言程序不加修改或稍加修改即可方便地移植到另一种结构类型的处理器上去。这些都增强了我们使用各种处理器芯片进行产品开发的能力。

3. 表达能力强

C 语言具有丰富的数据结构类型,可以根据需要采用整型、实型、字符型、数组类型、指针类型、结构类型、联合类型、枚举类型等多种数据类型来实现各种复杂数据结构的运算。C 语言还具有多种运算符,灵活使用各种运算符可以实现其他高级语言难以实现的运算。

4. 表达方式灵活

利用 C 语言提供的多种运算符,可以组成各种表达式,还可采用多种方法来获得表达式的值,从而使用户在程序设计中具有更大的灵活性。C 语言的语法规则不太严格,程序设计的自由度比较大,程序的书写格式自由灵活。程序主要用小写字母来编写,而小写字母是更容易阅读的,这些充分体现了 C 语言灵活、方便和实用的特点。

5. 可结构化程序设计

C 语言是以函数作为程序设计的基本单位的。C 语言程序中的函数相当于汇编语言中的子程序。C 语言对于输入和输出的处理也是通过函数调用来实现的。各种

C 语言编译器都会提供一个函数库,其中包含有许多标准函数,如各种数学函数、标准输入输出函数等。此外,C 语言还具有自定义函数的功能,用户可以根据自己的需要编制满足某种特殊需要的自定义函数。实际上,C 语言程序就是由许多个函数组成的,一个函数相当于一个程序模块,因此 C 语言可以很容易地进行结构化程序设计。

6. 可直接操作计算机硬件

C 语言具有直接访问处理器物理地址的能力,可以直接访问片内或片外存储器,还可以进行各种位操作。

7. 生成的目标代码质量高

众所周知,汇编语言程序目标代码的效率是最高的,这就是为什么汇编语言仍是编写计算机系统软件的重要工具的原因。但是统计表明,对于同一个问题,用 C 语言编写的程序生成代码的效率仅比用汇编语言编写的程序低 10%~20%。

尽管 C 语言具有很多的优点,但和其他任何一种程序设计语言一样也有其自身的缺点,如不能自动检查数组的边界,各种运算符的优先级别太多,某些运算符具有多种用途等。但总的来说,C 语言的优点远远超过了它的缺点。经验表明,程序设计人员一旦学会使用 C 语言,就会对它爱不释手,尤其是单片机应用系统的程序设计人员更是如此。

1.3 开发 LPC11XX 使用的 C 编译器

开发 LPC11XX 嵌入式处理器的 C 编译器主要有 ARM Real View(简称 Real View MDK)、IAR EWARM(简称 IAREW)和 LPCXpresso。

Real View MDK 是 ARM 旗下 Keil 公司的开发工具,Real View MDK 集成了业内最领先的技术,包括 μ Vision4 集成开发软件与 Real View 编译器,支持目前所有的 ARM Cortex 核处理器、自动配置启动代码、集成 Flash 烧写模块、强大的 Simulation 设备模拟、性能分析等功能。

从 51 单片机开始,我们就和 Keil 公司软件打交道了,Keil 的开发软件在业内公认是使用方便、界面友好的优秀产品,因此我们的学习也采用 Real View MDK 集成开发环境。

IAREW 是瑞典 IAR SYSTEMS 公司开发的集成开发环境(IDE),包含嵌入式编译器、汇编器、连接定位器、库管理器、项目管理及调试器等。其特点是编译效率高、功能齐全,但价格高昂。

LPCXpresso 是 NXP 公司配合针对 ARM Cortex 内核的 LPC 系列处理器推出的一款低成本开发工具,全面支持 LPC11XX、LPC12XX、LPC13XX 及 LPC17XX 等系列容量不超过 128 KB 的处理器。

第 2 章

ARM 的发展

2.1 什么是 ARM

ARM 英文全称 Advanced RISC Machines, 是英国一家电子公司的名字, 该公司成立于 1990 年, 是苹果公司、Acorn 计算机公司和 VLSI Technology 的合资企业。

ARM 也可以理解为是一种技术, ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司, 作为知识产权供应商, 其本身不直接从事芯片生产, 靠转让设计许可由合作公司生产各具特色的芯片, 世界各大半导体生产商从 ARM 公司购买其设计的微处理器核, 根据各自不同的应用领域, 加入适当的外围电路, 从而形成自己的 ARM 微处理器芯片进入市场。目前, 全世界有几十家大的半导体公司都使用 ARM 公司的授权, 因此 ARM 技术获得了更多的第三方工具、制造、软件的支持, 又降低了整个系统成本, 使产品更容易进入市场被消费者接受, 更具有竞争力。

ARM 还可以认为是采用 ARM 技术开发的 RISC 处理器的通称。ARM 微处理器已遍及工业控制、消费类电子产品、通信系统、网络系统及无线系统等各类产品市场, 基于 ARM 技术的微处理器应用约占据了 32 位 RISC 微处理器 75% 以上的市场份额, ARM 技术正在逐步渗入到我们生活的各个方面。

2.2 处理器 RISC 技术简介

传统的 CISC (Complex Instruction Set Computer, 复杂指令集计算机) 体系由于指令集庞大, 指令长度不固定, 指令执行周期有长有短, 使指令译码和流水线的实现在硬件上非常复杂, 给芯片的设计开发和成本的降低带来了极大困难。

随着计算机技术的发展, 需要不断引入新的复杂的指令集, 为支持这些新增的指令, 计算机的体系结构会越来越复杂。然而, 在 CISC 指令集的各种指令中, 其使用

频率却相差悬殊,大约有 20% 的指令会被反复使用,占整个程序代码的 80%,而余下的 80% 的指令却不经常使用,在程序设计中只占 20%。显然这种结构是不太合理的。

针对这些明显的弱点,1979 年美国加州大学伯克利分校提出了 RISC(Reduced Instruction Set Computer,精简指令集计算机)的概念。RISC 并非只是简单地减少指令,而是把着眼点放在了如何使计算机的结构更加简单合理地提高运算速度上。RISC 结构优先选取使用频率最高的简单指令,避免复杂指令,将指令长度固定,指令格式和寻址方式种类减少,以控制逻辑为主,不用或少用微码控制等来达到上述目的。

加州大学伯克利分校的 Patterson 教授领导的研究生团队设计和实现了“伯克利 RISC I”处理器,他们在此基础上又发展了后来 SUN 公司的 SPARC 系列 RISC 处理器,并使得采用该处理器的 SUN 工作站名震一时。与此同时,斯坦福大学也在 RISC 研究领域取得了重大进展,开发并产业化了 MIPS(Million Instructions Per Second)系列 RISC 处理器。

2.3 ARM 处理器

2.3.1 ARM 处理器的发展历程

1978 年 12 月 5 日,物理学家 Hermann Hauser 和工程师 Chris Curry,在英国剑桥创办了 CPU(Cambridge Processing Unit)公司,主要业务是为当地市场供应电子设备。

1979 年,CPU 公司改名为 Acorn 计算机公司。起初,Acorn 公司打算使用摩托罗拉公司的 16 位芯片,但是发现这种芯片速度太慢也太贵。一台售价 500 英镑的机器,不可能使用价格 100 英镑的 CPU! 于是他们转而向 Intel 公司索要 80286 芯片的设计资料,但是遭到拒绝,被迫自行研发。

ARM 的设计是 Acorn 计算机公司于 1983 年开始的开发计划。这个团队由 Roger Wilson 和 Steve Furber 带领,着手开发一种类似于高级 6502 架构的处理器。Acorn 有一大堆构建在 6502 处理器上的电脑,因此能设计出一颗类似的芯片则意味着对公司有很大的优势。

1985 年 4 月 26 日,Roger Wilson 和 Steve Furber 设计了他们自己的第一代 32 位、6 MHz 的处理器,用它做出了一台 RISC 指令集的计算机——ARM1(见图 2-1),简称 ARM(Acorn RISC Machine)。这就是 ARM 这个名称的由来。该计算机由美国加州 San Jose VLSI 技术公司制造,而首颗真正能量产的“ARM2”于次年投产。ARM2 具有 32 位的数据总线、26 位的寻址空间,并提供 64 MB 的寻址范围与 16 个

32 位的暂寄存器。暂寄存器中有一个作为程序计数器,其前面 6 位和后面 2 位用来保存处理器状态标记(Processor Status Flags)。ARM2 是当时最简单实用的 32 位微处理器,仅容纳了 30 000 个晶体管(6 年后摩托罗拉的 68000 包含了 70 000 个),之所以精简是因为它不含微码(这大概占了 68000 的晶体管数的 $1/4 \sim 1/3$),而且与当时大多数的处理器相同,它没有包含任何的高速缓存。这个精简的特色使它只需消耗很少的电能,却能发挥比 Intel 80286 更好的性能。后继的处理器“ARM3”则备有 4 KB 的高速缓存,使它能发挥更佳的性能。

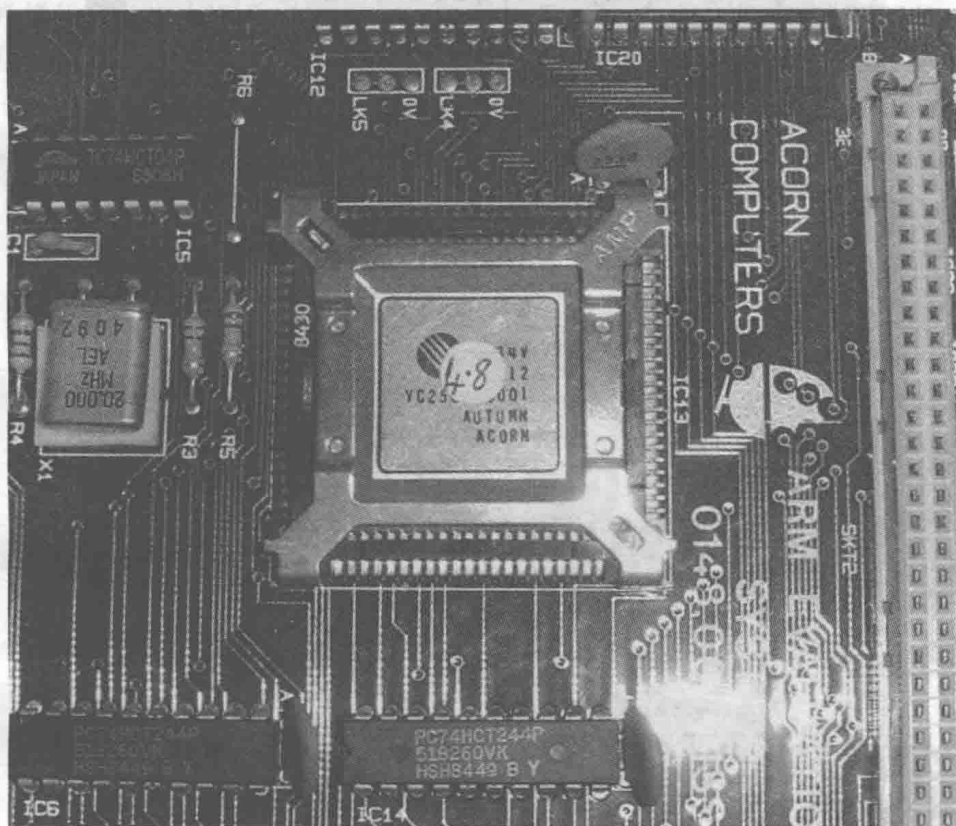


图 2-1 用在 BBC Micro 上的 ARM1 第二代处理器

20 世纪 80 年代后期,Acorn 很快开发出 ARM 的台式机产品,形成英国的计算机教育基础。

1990 年 11 月 27 日,Acorn 公司正式改组为 ARM 计算机公司。苹果公司出资 150 万英镑,芯片厂商 VLSI 出资 25 万英镑,Acorn 本身则以 150 万英镑的知识产权和 12 名工程师入股。那时,公司的办公地点非常简陋,为一个谷仓(见图 2-2),图 2-3 为 ARM 工程师们当年在谷仓开会的场景。

初创时期的 ARM 没有商业经验,没有管理经验,当然也没有世界标准这种远景,运营资金紧张,工程师人心惶惶,最后 ARM 决定自己不生产芯片,转而以授权的