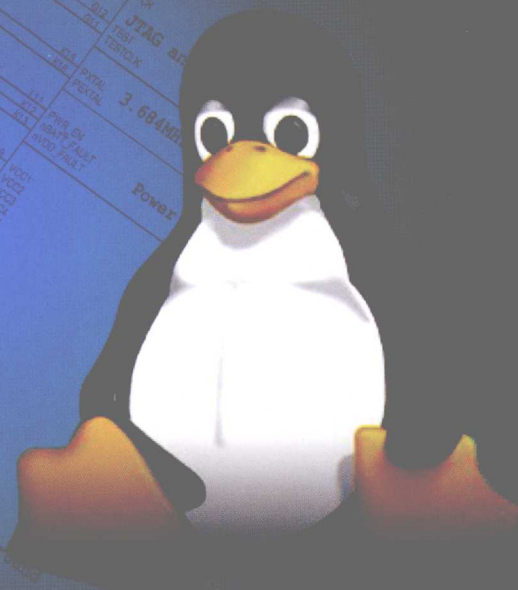


基于Linux 嵌入式原理与应用开发

赵国安 郁斌 薛琳强 编著
亿道公司 审校



清华大学出版社

基于 Linux 嵌入式原理与应用开发

赵国安 郁 斌 薛琳强 编著

亿道公司 审校



清华大学出版社

北 京

内 容 简 介

本书以 ARM 处理器和开源的 Linux 操作系统为研究对象, 综合介绍嵌入式系统软硬件及开发的相关内容, 帮助读者建立嵌入式系统开发的整体概念, 使读者在以后的实际应用开发中不断提高对嵌入式系统知识的了解和体会, 最终解决实际应用开发中的问题。

本书内容全面, 叙述言简意赅、清晰流畅, 讲解透彻、通俗易懂, 图例丰富, 主要章节的实例由亿道公司工程师验证。

本书可作为高等院校嵌入式系统相关课程的教材和参考书, 也可作为嵌入式系统开发人员的自学参考书。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

基于 Linux 嵌入式原理与应用开发/赵国安, 郁斌, 薛琳强编著. —北京: 清华大学出版社, 2008. 10

ISBN 978-7-302-18218-4

I. 基… II. ①赵… ②郁… ③薛… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2008) 第 110880 号

责任编辑: 涂 荣 郭 伟

封面设计: 张 岩

版式设计: 刘 娟

责任校对: 焦章英

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市李旗庄少明装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 15.5 字 数: 343 千字

版 次: 2008 年 10 月第 1 版 印 次: 2008 年 10 月第 1 次印刷

印 数: 1~5000

定 价: 30.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 028179-01

前 言

本书采用了 ARM 处理器和开放源代码的 Linux 操作系统作为研究对象,综合介绍了嵌入式系统的软硬件构造和系统开发的相关内容。想对嵌入式系统的知识进行初步了解,以及有志于嵌入式系统产品开发的读者,都能从本书中得到相应的知识。需要全面深入理解和实践本书内容的读者,需具备简单电子线路的基础知识和掌握至少一种编程语言。

本书从实际应用角度出发,着重于帮助读者建立一个嵌入式系统清晰的系统概念以及解决读者实际应用中的问题。以此为目的,本书采取了与国内大多数嵌入式系统书籍有所区别的章节划分和内容编排,全书共分为 8 章。

第 1 章是嵌入式系统的概述,对嵌入式系统概念、软/硬件基础知识、开发工具和发展趋势做了基础性的介绍。

第 2 章和第 3 章介绍了嵌入式 Linux 操作系统、开发环境的知识,并以 PXA255 开发板为操作实例(读者如果使用其他开发板,会发现原理和操作方法都是大同小异的),完整地给出了包括构建 Windows 和 Linux 下的开发环境、交叉编译环境、在实际开发板上移植 Linux 内核、移植和启动 U-Boot 等各种相关概念以及实际操作方法,图文并茂,使初级读者可以很快地构建自己的嵌入式实践环境,并清晰地理解嵌入式软件及其开发的整体概念。

第 4 章在介绍硬件系统总结构的基础上,以一个实际的 1Bit CPU 的设计过程引导读者建立一个完整的嵌入式系统硬件的概念;在第 5 章和第 6 章中,再讲解 ARM 处理器和 PXA270 开发板中纷繁复杂的体系结构、接口、内核、寻址、指令操作等细节内容。第 4、5、6 章带领读者深入嵌入式系统的全部硬件细节和接口驱动调试概念。读者也可以在掌握了整体概念的基础上,在实际操作中慢慢体会各个细节的作用和使用方法。

本书的最后部分着重于实践操作。第 7 章首先介绍 ADS 集成开发环境的相关知识,熟悉了开发工具后,第 8 章用一系列实验引导读者开始嵌入式开发的工作,深入了解嵌入式系统的实际应用。

本书在编写过程中得到了黄衍玺、李锐、王龙军等极大帮助,没有他们的努力也没有本书的最终成稿。

由于时间仓促,作者水平有限,书中难免有不妥和疏漏之处,敬请广大读者批评指正。

作 者

于紫金山麓

2008 年 4 月 24 日

目 录

第 1 章 嵌入式系统概述.....	1
1.1 嵌入式系统概述.....	1
1.1.1 概述.....	1
1.1.2 特点.....	2
1.1.3 系统构架.....	4
1.2 嵌入式系统硬件部分.....	4
1.2.1 集成电路的发展.....	4
1.2.2 传统微处理器 (8 位/16 位) 的更新.....	8
1.2.3 今日嵌入式 (32 位/16 位).....	9
1.3 嵌入式系统软件部分.....	10
1.3.1 嵌入式操作系统的特点.....	10
1.3.2 常用嵌入式操作系统.....	11
1.3.3 嵌入式系统编程语言.....	12
1.3.4 嵌入式系统开发.....	14
1.4 ARM 处理器.....	15
1.5 开发工具.....	16
1.5.1 软件开发工具.....	16
1.5.2 硬件开发工具.....	17
1.6 嵌入式系统的应用及发展趋势.....	18
第 2 章 嵌入式 Linux 及其构建.....	20
2.1 Linux 发展.....	20
2.1.1 Linux 发展简史.....	20
2.1.2 GNU 与 Linux.....	20
2.1.3 Linux 的优点.....	21
2.1.4 Linux 的版本.....	22
2.1.5 Linux 的内核.....	23
2.2 主流嵌入式 Linux 介绍.....	23
2.2.1 MontaVista Linux.....	24
2.2.2 μ Clinux.....	24

2.2.3	RTLinux	25
2.2.4	RTAI	25
2.2.5	eCos	26
2.2.6	实时化改造	27
2.3	构建嵌入式 Linux	28
2.3.1	搭建开发环境	28
2.3.2	获取 U-Boot	48
2.3.3	获取 Linux 内核	50
第 3 章	嵌入式 Linux 的移植过程	51
3.1	PXA255 简介与 Sitsang 开发板	51
3.1.1	PXA255 处理器 (CPU) 简介	51
3.1.2	Sitsang 开发板简介	53
3.2	存储器映射	55
3.2.1	ARM 存储系统概述	55
3.2.2	PXA255 的存储器映射	55
3.2.3	Sitsang 开发板存储器映射	61
3.3	GPIO (General-Purpose I/O, 通用 I/O)	62
3.3.1	GPIO 概述	63
3.3.2	GPIO 操作	63
3.4	中断机制	66
3.5	启动过程	67
3.5.1	Sitsang 开发板的启动过程	67
3.5.2	U-Boot 的启动过程	68
3.6	移植 U-Boot 简介	72
3.6.1	U-Boot 的基本结构	72
3.6.2	U-Boot 移植前期准备	73
3.6.3	U-Boot 移植的基本过程	74
3.7	移植 Linux 2.6 内核	79
3.7.1	Linux 2.6 内核源代码的基本结构	79
3.7.2	Linux 2.6 内核移植前期准备	80
3.7.3	Linux 2.6 内核的配置	81
3.7.4	交叉编译 Linux 2.6 内核	86
3.7.5	Linux 2.6 设备驱动移植介绍	89
3.8	构建根文件系统 (root filesystem)	90
3.8.1	什么是根文件系统	90
3.8.2	根文件系统与文件系统的区别	91

3.8.3 根文件系统与 Linux 内核的关系.....	92
3.8.4 使用 Busybox 制作根文件系统.....	92
第 4 章 了解嵌入式系统硬件.....	101
4.1 本章概述.....	101
4.2 计算机系统组成部分.....	102
4.3 计算机硬件.....	106
4.4 指令系统.....	107
4.4.1 指令流和数据流.....	108
4.4.2 指令周期.....	110
4.4.3 时序发生器.....	114
4.4.4 组合逻辑控制器.....	118
4.4.5 指令执行流程.....	119
4.5 I/O 子系统.....	120
4.5.1 I/O 子系统简介.....	120
4.5.2 基础 I/O 的概念.....	121
4.5.3 I/O 子系统.....	124
4.6 1bit 处理器芯片设计案例.....	129
第 5 章 ARM 体系结构.....	135
5.1 ARM 体系结构.....	135
5.1.1 ARM 嵌入式微处理器概述.....	135
5.1.2 典型的 ARM 处理器简介.....	138
5.2 ARM 接口.....	142
5.2.1 协处理器接口.....	142
5.2.2 AMBA 接口.....	144
5.2.3 JTAG 接口.....	145
5.3 ARM 处理器内核.....	148
5.3.1 ARM7TDMI 处理器内核.....	148
5.3.2 ARM9TDMI 处理器内核.....	150
5.3.3 ARM10TDMI 处理器内核.....	151
5.4 ARM 处理器核.....	153
5.4.1 ARM720T/ARM740T 处理器核.....	153
5.4.2 ARM920T/ARM940T 处理器核.....	153
5.5 ARM 编程模型.....	154
5.5.1 数据类型.....	154
5.5.2 处理器模式及工作状态.....	154

5.5.3	寄存器组织	155
5.6	ARM 基本寻址方式	156
5.6.1	寄存器寻址	156
5.6.2	立即寻址	156
5.6.3	寄存器移位寻址	157
5.6.4	间接寻址	157
5.6.5	基址变址寻址	158
5.6.6	块复制寻址	158
5.6.7	多寄存器寻址	158
5.6.8	相对寻址	159
5.6.9	堆栈寻址	159
第 6 章	ARM 指令集格式	160
6.1	指令格式	160
6.1.1	基本格式	160
6.1.2	第二操作数	160
6.1.3	条件码	161
6.2	ARM 指令分类	162
6.2.1	存储器访问指令	162
6.2.2	ARM 数据处理指令	167
6.2.3	乘法指令	172
6.2.4	跳转指令	173
6.2.5	协处理器指令	175
6.2.6	杂项指令	177
6.2.7	伪指令	180
6.3	Thumb 指令集	182
6.3.1	Thumb 指令集	182
6.3.2	Thumb 指令集的优点	183
6.3.3	Thumb 指令集与 ARM 指令集的区别	183
第 7 章	ARM ADS 集成开发环境的使用	185
7.1	ADS 集成开发环境组成介绍	185
7.1.1	命令行开发工具	185
7.1.2	ARM 运行时库	195
7.1.3	GUI 开发环境 (Code Warrior 和 AXD)	197
7.1.4	实用程序	200
7.1.5	支持的软件	200

7.2 使用 ADS 创建工程.....	201
7.2.1 建立一个工程.....	201
7.2.2 编译和链接工程.....	204
7.2.3 使用命令行工具编译应用程序.....	209
7.3 用 AXD 进行代码调试.....	211
7.4 本章小结.....	214
第 8 章 嵌入式实验.....	215
实验一 系统引导.....	215
实验目的.....	215
实验内容.....	215
实验步骤.....	219
实验二 键盘.....	220
实验目的.....	220
实验内容.....	220
实验步骤.....	220
实验三 IRQ 中断处理.....	222
实验目的.....	222
实验内容.....	222
实验步骤.....	222
实验四 定时器.....	225
实验目的.....	225
实验内容.....	225
实验步骤.....	225
实验五 串口传输.....	227
实验目的.....	227
实验内容.....	227
实验步骤.....	228
实验六 实时时钟.....	231
实验目的.....	231
实验内容.....	231
实验步骤.....	231
实验七 LCD 控制器.....	233
实验目的.....	233
实验内容.....	233
实验步骤.....	233
参考文献.....	238

第 1 章 嵌入式系统概述

嵌入式系统是把计算机直接嵌入到应用系统中，它融合了计算机软/硬件技术、通信技术和微电子技术。随着微电子技术和半导体技术的高速发展，超大规模集成电路技术和深亚微米制造工艺已十分成熟，从而使高性能系统芯片的集成成为可能，并推动着嵌入式系统向最高级构建形式，即片上系统 SOC (System on a Chip) 的水平发展，进而促使嵌入式系统得到更深入、更广阔的应用。嵌入式技术的快速发展不仅使其成为当今计算机技术和电子技术的一个重要分支，同时也使计算机的分类从以前的巨型机/大型机/小型机/微型机变为通用计算机/嵌入式计算机（即嵌入式系统）。

1.1 嵌入式系统概述

嵌入式系统是集成电路发展过程中的一个标志性成果，它把计算机直接嵌入到应用系统中，融合了计算机软/硬件技术、通信技术和微电子技术，是一种微电子产业和信息技术产业的最终产品。微电子产业是许多国家优先发展的产业。以超深亚微米工艺和 IP 核复用技术为支撑的系统芯片技术是国际超大规模集成电路发展的趋势和 21 世纪集成技术的主流。

1.1.1 概述

通常，计算机连同一些常规的外设是作为独立的系统而存在的，而并非为某一方面的专门应用而存在。例如一台 PC 机就是一个计算机系统，整个系统存在的目的就是为人们提供一台可编程、会计算、能处理数据的机器。可以用它作为科学计算的工具，也可以用它作为企业管理的工具。所以，人们把这样的计算机系统称为“通用”计算机系统。但是有些系统却不是这样。例如，医用的 CT 扫描仪也是一个系统，里面有计算机，但是这种计算机（或处理器）是作为某个专用系统中的部件而存在的，其本身的存在并非目的而只是手段。像这样“嵌入”到更大的、专用的系统中的计算机系统，就称之为“嵌入式计算机”、“嵌入式计算机系统”或“嵌入式系统”。从字面上讲，后者似乎比前者更为广义，因为系统中常常还包括一些机电、光电、热电或者电化的执行部件，但是实际上却往往不作严格的区分。在不致引起混淆的情况下，一般把这三者用作同义词，并且一般总是指系

统中的核心部分，即嵌入在系统中的计算机。

不过，虽然计算机在整个大系统中只是一个部件，却通常起着相当于“大脑”的作用。所以事实上所嵌入的计算机就是整个系统的核心，而系统中的其他部件则是其外部设备。只不过这些外部设备不同于常规的计算机外部设备，而所嵌入的计算机的作用和目的又只限于对这些外部设备的控制和管理而已。所以也可以说，常规的计算机系统是面向计算（包括数值和非数值）和处理的，而嵌入式计算机则一般是面向控制的。

所谓将计算机“嵌入”到系统中，一般并不是指直接把一台通用计算机原封不动地安装到目标系统中，也不只是简单地把原有的机壳拆掉并安装到机壳中，而是指为目标系统构筑起合适的计算机系统，再把它有机地植入、甚至融入目标系统。

虽然嵌入式系统在工业、服务业、消费电子等领域的应用范围不断扩大，且已渗透到人们日常生活的方方面面，但是依然很难给它下一个明确的定义。嵌入式系统本身是一个外延很广的名词，凡是与产品结合在一起的、具有嵌入式特点的控制系统都可以叫做嵌入式系统。国际电工与电子工程师协会（即 IEEE）曾经把嵌入式系统定义为“控制、监视或者辅助设备、机器和车间运行的装置”。显然该定义具有突出的应用色彩，不过现在看来，似乎并未充分体现嵌入式系统现今的学术内涵。目前国内普遍认同的嵌入式系统的定义是“嵌入式系统是将计算机嵌入到某个应用系统内的一种计算机体系结构形式”。换言之，嵌入式系统是以应用为中心，以计算机技术为基础，软硬件能灵活变化以适应所嵌入的应用系统，对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。简言之，嵌入式系统就是嵌入式计算机系统的简称。

从 20 世纪 80 年代初，嵌入式系统的研发开始用商业级“操作系统”编写嵌入式应用软件，结果使嵌入式应用的开发周期缩短、成本降低。这些嵌入式操作系统均具有嵌入式技术的典型特点：采用占先式的调度，响应的时间很短，任务执行的时间可以确定；系统内核很小，且具有可剪裁性、可扩充性和可移植性，可以移植到各种型号的微处理器（单片机）上；具有较强的实时性和可靠性，适合于嵌入式应用。20 世纪 90 年代以后，随着诸多应用领域对嵌入式系统实时性要求的提高，各种应用软件的规模不断扩大，又促使嵌入式系统的实时内核逐渐发展为实时多任务操作系统（RTOS），并作为一种软件平台逐步演变为目前国际上流行的嵌入式操作系统。

在嵌入式操作系统迅速发展的同时，系统芯片的制造与设计技术也在不断进步。系统芯片就是把一个完整的最终产品的主要功能单元集成到一块或一组大规模集成电路芯片上，这是现代集成电路工艺技术——深亚微米技术迅速发展的必然结果。系统芯片制造技术发展主要体现在硅圆片的尺寸逐渐增大，硅晶片的特征线宽逐步减小，同时芯片的集成度不断提高。系统芯片技术的发展，使得嵌入式系统硬件进一步向微型化、高集成化发展，从而为嵌入式系统的应用开辟更为广阔的天地。

1.1.2 特点

嵌入式系统既然是计算机系统，就不可避免地由三大部分构成：CPU、内存和输入输

出手段。此外,当然还得有将这三大部分连接起来的“总线”。这是所有计算机系统的共性,是万变不离其宗的。但与以 PC 机为代表的通用计算机系统比较,嵌入式有它的特殊性,其特点概括如下。

(1) 可靠性与稳定性对嵌入式系统有着特别重要的意义。

即使逻辑上的系统结构相同,在物理组成上也会有所不同。其所使用的元器件,包括接插件、电源等的质量和可靠性要求都比较高,元器件的“平均无故障时间 MTBF (Mean Time Between Failure)”成为关键参数,此外允许的环境温度也是需要重点考虑的问题。

(2) 嵌入式系统的软硬件均是面向特定应用对象和任务设计的,具有很强的专用性。

嵌入式系统的主要目的是控制,需要由嵌入式系统提供的功能以及面对的应用和过程都是预知的、相对固定的,而不像通用计算机那样有很大的随意性。控制对象的精度要求一般不会超出整型数所能提供的分辨率,所以浮点运算一般没有必要。而且由于是专用的系统,在可编程方面就不需要那么灵活。嵌入式系统不会用于作为应用软件的环境,在嵌入式系统上通常也不会运行一些大型的软件。所以,嵌入式系统对 CPU 计算能力的要求并不像通用计算机那么高。

(3) 有些嵌入式系统需要长期连续运行(如电话交换机)。

在这些系统长期的运行过程中,如果某个部件损坏,是不允许将系统关闭断电后加以替换的,而必须在系统仍旧加电、继续运行的条件下加以替换。这就是所谓“热替换(Hot Swapping)”或“热插拔”,即系统还在运行的条件下将电路板插入系统的总线,或者从总线上拔下电路板。不过,这一般是指整个模块或电路板的替换,而不是指个别元器件的替换。还有一种类似但稍为简单一些的要求是在系统还在运行的条件下增加一个模块,称为“热插入”。无论是热替换还是热插入,都要求硬件和软件两方面的配合。

(4) 有些要求高可靠的嵌入式系统还需要采用“容错(Fault Tolerance)”技术。

即系统在损坏时能自动切换到它的备份,或者对系统进行重构。最常见也是最简单的容错技术是电源的自动切换。一些嵌入式系统往往带有备份电源,一旦检测到电源故障就自动切换到备份电源并发出警报。还有一种很常见的容错技术是一些数据通信设备中对链路的“自动保护切换(APS)”。在电信系统中,最可能发生故障的“部件”就是物理的链路(例如两个城市之间的电缆),所以实践中常常以一定的比例为这些电缆配上备份,例如每3条平行的链路就配上一条作为备份。而在两端的设备中,则各有相应的设施,使得监测到某条链路发生故障时,就很快切换到备份链路继续运行。

(5) 许多嵌入式系统都有实时要求,需要有对外部事件迅速做出反应的能力。

以前,“嵌入式系统”几乎是“实时系统”的代名词,近年来出现了许多不带实时要求的嵌入式系统,这两个词的区别才又变得显著起来。但是,即使是现在,多数嵌入式系统还是有着不同程度的实时要求。

(6) 在系统组成上,因为嵌入式系统常常用于控制目的,其外设接口自然就比较多样,并且数量往往比较多。

(7) 与通用计算机相比,嵌入式系统一般都不带用于大容量存储目的的外部设备,也就是不带磁盘。而操作系统的映像和可执行程序一般都存放在只读存储器(ROM)或“闪存(Flash Memory)”中。

(8) 许多嵌入式系统的人机界面也有其特殊性。

许多嵌入式系统都不提供图形人机界面,而只提供一个面向字符的控制台接口。不过往往还带有如小型的 LCD 显示屏、发光二极管(LED)等辅助的显示设备,甚至报警装置。

1.1.3 系统构架

嵌入式系统早期主要应用于军事及航空航天等领域,以后逐步广泛应用于工业控制、仪器仪表、汽车电子、通信和家用消费电子类等领域。随着 Internet 的发展,新型的嵌入式系统正朝着信息家电 IA (Information Appliance) 和 3C (Computer、Communication & Consumer) 产品方向发展。

嵌入式系统采用“量体裁衣”的方式把所需的功能嵌入至各种应用系统中。根据应用形式的不同,可有 IP (Intellectual Property) 级、芯片级和模块级 3 级不同的体系架构。

(1) IP 级的架构也就是系统级芯片 SOC (System on Chip) 的形式。把不同的 IP 单元,根据应用的要求集成在一块芯片上,各种嵌入式软件也可以以 IP 方式集成在芯片中。

(2) 芯片级架构是根据各种 IT 产品(应用系统)的要求,选用相应的处理器(MCU、DSP、RISC 及 MPU 等)芯片、RAM、ROM (EPROM/EEPROM/FLASH) 及 I/O 接口芯片等组成相应的嵌入式系统;相应的系统软件或应用软件也以固件形式固化在 ROM 中。这是目前嵌入式系统最常见的形式。

(3) 模块级架构是以 X86 处理器构成的计算机系统模块嵌入到应用系统中。这样可充分利用目前常用 PC 机的通用性和便利性。不过,此方式不但要缩小体积、增加可靠性,而且还要把操作系统 OS 改造为嵌入式操作系统 OS,把应用软件固化在固态盘中。此种嵌入式系统较多地出现在工业控制和仪器仪表中。

嵌入式系统由嵌入式处理器、嵌入式系统软件和嵌入式应用软件组成。Intel 公司推出的个人互联网用户架构 PCA 就是嵌入式系统的开发应用平台。在 PCA 的通信子系统内的嵌入式处理器是基于 StrongARM/XScale 的处理器,PCA 可以配备 Windows CE、嵌入式 Linux 和 VxWorks 等各种嵌入式 OS,同时,为了便于开发各种应用软件,提供了各种组件及中间件。

1.2 嵌入式系统硬件部分

嵌入式系统的广泛应用得益于电子技术的高速发展,下面讲述电子技术的发展以及最新嵌入式系统中采用的核心部件的发展历史和概况。

1.2.1 集成电路的发展

当前,我们正经历着一场新的技术革命。虽然第三次技术革命包含了新材料、新能源、

生物工程、海洋工程、航空航天技术和电子信息技术等，但影响最大、渗透性最强、最具有新技术革命代表性的乃是以微电子技术为核心的电子信息技术。

信息是客观事物状态和运动特征的一种普遍形式，是继材料和能源之后的第三大资源，是人类物质文明与精神文明赖以发展的三大精神支柱之一。目前我们正处在一场跨越时空的新的信息技术革命中，它将比历史上的任何一次技术革命对社会、经济、政治、文化等带来的冲击都更为巨大，它将改变我们的生产方式、生活方式、工作方式以及治理国家的方式。

实现社会信息化的关键是各种计算机和通讯机，但是其基础都是微电子。1946年2月在美国莫尔学院诞生了第一台名为电子数值积分器和计算机（Electronic Numerical Integrator and Computer）的计算机，即 ENIAC 的问世。当时的 ENIAC 由 18000 个电子管组成，占地面积 150 平方米，重 30 吨，功率为 140 千瓦，足以发动一辆机车。然而这个庞然大物的运算速度只有每秒 5000 次，存储容量只有千位，平均稳定运行时间只有 7 分钟。设想一下，这样的计算机能够进入办公室、企业和家庭吗？所以当时曾有人认为，全世界只要有 4 台 ENIAC 就足够了，可是现在全世界的计算机不包括微机在内就有几百万台，造成这个巨大变革的技术基础就是微电子。现在，电子信息产业已经成为世界第一产业，毫无疑问，21 世纪将是信息化的世纪，因此 21 世纪的以微电子技术为核心的集成电路产业也必将得到高速发展。

在集成电路发展过程中，1947 年发明晶体管，1958 年美国德克萨斯公司制成第一块 IC。半个世纪以来，IC 一直在按摩尔定律发展，即 18 个月集成度提高一倍，同时其成本降低一倍。特征尺寸从 3、2、1.5、1.0、0.8、0.5、0.35、0.25、0.18 μm ，计划向 0.15、0.13、0.09、0.065 μm 发展。圆片尺寸从 76、100、120、150 到 200、300mm，计划向 400、450mm 发展。目前，以美国为代表的最先进的 IC 发展水平是 0.18 μm ，0.13 μm 已经形成，正在研究 90nm 技术，200mm、300mm 硅圆片的量产 IC 技术已经成熟。

集成电路的应用范围广泛，门类繁多。可以按照集成电路的结构类型、规模（集成度）、使用的基片材料、电路功能及应用领域等进行分门别类。按照 IC 的应用性质可以将 IC 分为通用 IC 和专用 IC（ASIC），下面讨论 ASIC。

传统的 IC 设计以 ASIC 设计为主。ASIC 产品具有体积小、可靠性高、功耗小、仿制困难、批量生产后成本低等优点。20 世纪 80 年代，随着微电子技术产品制造成本的不断下降和设计自动化技术及其工具的长足发展，刺激了系统设计师自己设计专用电路的思想，以弥补 MPU 标准产品功能的不足和过剩的缺陷，推动了一个产品满足特定用户需求的 ASIC 的发展，其中特别是预测母片的掩模编程技术，大大平衡了微电子生产厂家和电子整机厂家间的不同利益，受到了普遍重视。但是这种掩模硬编程专用电路（M-ASIC）不够方便，因此在 20 世纪 80 年代末 90 年代初又发展了可编程门阵等一类用户自编程专用电路（U-ASIC）。最近十年 PLD 技术已经成为设计数字系统的首选技术，在很多场合 PLD 取代了传统意义上的 ASIC 芯片，“在系统编程技术”方便了用户，降低了设计成本和风险，缩短了设计周期，从而使微电子产品特性进入了一个全新的概念。当前，ASIC 和 PLD 技术的门数越来越多，芯片主频越来越高，两者也互相融合、优势互补，都向着 SoC 方向发展。

ASIC 大致可以分为：门阵列电路、标准单元电路、可编程逻辑电路和全定制电路。其中前 3 类电路又被称为“半定制电路”。

一般而言，“半定制电路”采用带有基本单元的“母片”（又称“基板”），或采用标准单元库中的标准单元。设计师所关心的只是利用这些标准单元，根据系统的要求来选择布线方案以构筑不同的系统。当然，标准单元的意义已由早期的门、触发器，扩展到当前的微处理器、存储器等，这些规模较大的标准单元在专业上又称为“核”。“全定制电路”则从基本单元到整个系统均由系统设计师来构筑。所以当今的专用 IC 发展趋势是：今天的电路板是明天的专用芯片。

可编程逻辑器件简称 PLD，是新一代的数字器件。它不仅具有很高的速度和可靠性，而且具有用户可重复定义的逻辑功能即具有可重复编程的特点。因此，可编程逻辑器件使数字电路系统的设计非常灵活，并且大大缩短了系统研制的周期，缩小了数字电路系统的体积和所用芯片的品种。

可编程逻辑器件 PLD 分类框图如图 1.1 所示。

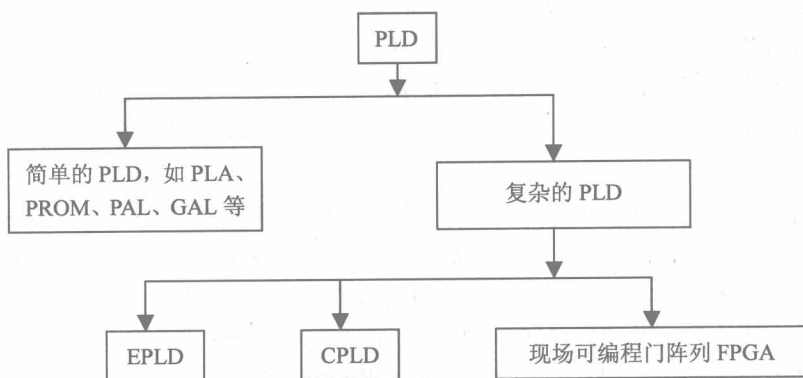


图 1.1 PLD 分类框图

1. 简单 PLD

简单的 PLD 是由“与”阵列及“或”阵列组成，能有效地以“积之和”的形式实现布尔逻辑函数。从技术实现上，输入到 PLD 的信号必须首先通过一个“与”阵列，在这里形成输入信号的组合。每组相“与”的组合被称为布尔表达式的子项或 PLD 术语中的乘积线。这个乘积线在第二个“或”门阵列中被相加。简单 PLD 在“与”、“或”阵列的基础上有 3 种基本类型，可根据阵列能否编程来区分：（1）可编程只读存储器即 PROM，它的“与”阵列固定，“或”阵列可编程；（2）可编程阵列逻辑即 PAL，它的“与”阵列可编程，“或”阵列固定，GAL 器件是 PAL 器件的增强型，内部结构相同，但可被配置成不同的功能；（3）可编程逻辑阵列即 PLA，它的“与”阵列和“或”阵列都可编程。

2. 复杂 PLD

复杂的可编程逻辑器件是由 PAL 或 GAL 发展而来的，基本上是扩充原始的可编程逻辑器件。它通常是由可编程逻辑的功能块围绕一个位于中心和延时固定的可编程互联矩阵构成。

为了增加电路密度而不使性能或功耗受到损失，复杂的可编程逻辑器件在结构上引入

了各种特性。如：引入分页系统，分页的目的在于仅使阵列的一部分在任何给定的时刻被加电；按备份模式放置阵列，或者靠变换检测自动地控制加电，或者采用外部指令加以控制。在实现级上，“与”阵列与“或”阵列需要用缓冲器分开，因为这些一般是倒相器。在两个阵列中实际逻辑一般是相同的。某些公司已经引入了折叠 PLA，它仅用了—个实际阵列，但可以将乘积项反馈回阵列。也允许在单个器件中实现多个逻辑。

目前，复杂 PLD 又延伸出 3 个分支，即可擦除可编程的逻辑器件 EPLD、CPLD 和现场可编程门阵列 FPGA。

EPLD 可擦除可编程器件分为两类：一类是基于 UVEPROM 和 CMOS 技术的 EPLD；另一类是基于 E2CMOS 工艺的 EEPLD。

FPGA 现场可编程门阵列器件通常由布线资源围绕的可编程单元（或宏单元）构成阵列，再由可编程 I/O 单元围绕阵列构成整个芯片。构成阵列的逻辑单元由布线通道中的可编程连线连接起来实现一定的逻辑功能。一个 FPGA 可能包含静态存储单元，它们允许内连的模式在器件被制造以后再被加载或修改。

CPLD 主要是由可编程逻辑宏单元 LMC 围绕中心的可编程互连矩阵单元组成。其中 LMC 逻辑结构较复杂，并具有复杂的 I/O 单元互连结构，可由用户根据需要生成特定的电路结构，完成一定的功能。由于 CPLD 内部采用固定长度的金属连线进行各逻辑块的互连，所以设计的逻辑电路具有时间可预测性，避免了分段式互连结构时序不完全预测的特点。

到 20 世纪 90 年代，CPLD 发展更为迅速，不仅具有电可擦写特性，而且出现了边界扫描及在线可编程高级特性。同时除了以 LMC 为基础的内部的 AND、OR 阵列外，外围的 I/O 模块扩大了其在系统中的应用范围和扩展性。1992 年 Lattice 公司率先推出了系统编程 ISP 技术，并推出 isp-LSI1000 系列 ISP 器件。ISP 指采用直接或间接方式对电子系统中的多个 PLD 器件进行编程或改写技术，它可以达到对器件及整个系统进行重构的目的。其重构时间可延续到系统完成之后。

对于许多 CPLD 来说，修改后的逻辑不能完全符合原有的引脚输出，或已修改了关键的定时路径。这样，为了修改逻辑，就不得不重新进行电路布线。对于高密度 CPLD 器件尤其是这样。许多厂家的 CPLD 在进行层次互连后，性能降低，延迟不稳定。

现场可编程门阵列 FPGA，与 PAL、GAL 器件相比，它的优点是可以实时地对外加和内置的 RAM 或 EPROM 编程，实时地改变器件功能，实现现场可编程（基于 EPROM）或在线重配置（基于 RAM 型）。它是科学实验、样机研制、小批量产品生产的最佳选择器件。

FPGA 通常包含 3 类可编程资源：可编程逻辑功能块、可编程 I/O 块和可编程互连。可编程逻辑功能块是实现用户功能的基本单元，它们通常排列成一个阵列，散布于整个芯片；可编程 I/O 块完成芯片上逻辑与外部封装脚的接口，常围绕着阵列于芯片四周；可编程内部互连包括各种长度的线段和编程互连开关，它们将各个可编程逻辑块或 I/O 连接起来，构成特定功能的电路。不同生产厂家生产的 FPGA 在可编程逻辑块的规模、内部互连线的结构和采用的可编程元件上存在较大的差异。较常见的是 Xilinx 和 Altera 公司的 FPGA 器件。

FPGA 一般用于逻辑仿真。电路设计师设计一个电路首先要确定电路（先可用 VHDL 进行行为级描述，然后做逻辑综合），然后进行软件模拟及优化，以确认所设计电路的功能及性能。然而随着电路规模的不断加大，工作频率的不断提高，将会给电路引入许多分布

参数的影响,而这些影响用软件模拟的方法较难直接反映出来,所以有必要做硬件仿真。

不管上述电子工艺和器件如何发展,有一个共同的目标就是:以最快的速度、最合理的价格和最高的可靠性为用户应用提供可行的产品。为了满足用户的需求,根据工艺和设计技术的发展,最新的研究内容包含了 SoC 和 IP 等方面的最新内容,给电子系统提供了大量可以直接使用的内核和外设芯片。

1.2.2 传统微处理器(8位/16位)的更新

下面讨论一下近 20 年来传统微控制器的更新。归纳而言,有以下几个方面。

1. 微控制器的 CPU 核仍以 CISC 为主,但向 RISC 演化

在传统微控制器领域内还是以当初 Intel 确立的哈佛结构(程序和数据存储器分离的体系)和复杂指令集系统(CISC)为主。只有少数厂家生产精简指令集计算机,但仍然保留了哈佛的分立存储器结构,并没有采用冯·诺依曼的程序、数据相合并的存储结构,少数生产 RISC 的计算机厂家有 Microchip 的 8 位 PIC12/16/17/18Fxxx 控制器、Atmel 的 8 位 AVR 微控制器、SCENIX 的 8 位 SX18/20/28AC 和 SX48/52BD 微控制器。这 3 个厂家都使用了快闪存储器(FLASH)存放程序。

而 Vautomation Co.的 VUSB 芯片在 8 位 RISC MCU V8 的基础上,增加了 USB 核,由其上的固件支撑 USB 的工作。V8 MCU 和 USB 核共占 10000 个门。USB 核可用于主机端或设备端。USB 命令可用 C 语言、API 以及硬件描述语言 VHDL 进行修改。在不改动硬件的条件下就可改变 USB 的包长和缓冲器长。若要改动硬件,只需联系 Vautomation 厂家,便可通过 Internet 提供实现代码。

2. 提升指令执行速度

提高 8 位机的振荡频率和减少每机器周期包含的振荡周期数都可以提高指令的执行速度。如 Philips 公司把 12MHz 的 80C51 从每机器周期所包含振荡器周期数由 12 改为 6,获得 2 倍速,Winbond 公司由 12 改为 3,获得 4 倍速。目前,8 位微控制器的频率一般可以高至 33MHz~40MHz,所以,可以比较容易地从原来的 1MIPS 提升到 10MIPS。但是,目前指令执行速度提升到 50MIPS~100MIPS 的都是 RISC 指令的 8 位微控制器。

提升 8 位微控制器工作频率已经受到普遍的重视。Hitachi 半导体厂,1988 年设计其 8 位 H8 微控制器时,设定的目标是要集成更多的外设和支持 C 语言,追求高性能而不以廉价为目的。片上提供 0.5MB 的 FLASH 存储器。最近还增加了 USB 和 Bluetooth 通信接口,也考虑生产带有 Ethernet 接口的芯片,还计划将通常 32 位机上的应用(如 TCP/IP、语音识别和合成)技术下移到 H8。但是,便携式设备的屏蔽措施较少,很难通过家用电器 B 类 FCC 的条款规定,几经改进设计方法,最后才把内部噪声降下来并抑制了外部的噪声。

3. 集成大容量片上 FLASH 存储器,实现 ISP、IAP

近年来,8 位微控制器竞相采用 FLASH 存储器已成趋势,因为它集成度高、价格便宜、