



单片机与嵌入式系统丛书



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

嵌入式系统设计

Embedded Systems Design

An Introduction to Processes,
Tools and Techniques

覆盖嵌入式系统开发的全过程

- 硬件与软件划分
- 跨平台开发技术
- 固件调试
- 性能分析
- 测试与集成

[美] Arnold Berger 著
吕骏 译

CMP Books



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

单片机与嵌入式系统丛书

嵌入式系统设计

Embedded Systems Design

An Introduction to Processes, Tools and Techniques

[美] Arnold Berger 著

吕 骏 译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书详细介绍嵌入式系统开发的全过程，讲述在嵌入式开发环境下编程、调试系统的专业知识，带领读者进入嵌入式系统编程领域，使读者掌握嵌入式系统开发中的关键方法及技术，并把它们应用到实践中。

本书适于从事嵌入式系统研发的技术人员以及高校相关专业的师生阅读。

Copyright© 2002 by CMP Books. All right reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher; with the exception that the program listing may be entered, stored and executed in a computer system, but they may not be reproduced for publication.

本书中文简体专有翻译出版版权由 CMP Books 出版公司授予电子工业出版社。该专有出版版权受法律保护。未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。版权所有，侵权必究。

图书在版编目(CIP)数据

嵌入式系统设计 / (美)伯杰 (Berger,A.) 著; 吕骏译. —北京: 电子工业出版社, 2002.9

(单片机与嵌入式系统丛书)

书名原文: Embedded Systems Design

ISBN 7-5053-8014-1

I.嵌... II.①伯...②吕... III.微型计算机—系统设计 IV.TP36

中国版本图书馆 CIP 数据核字 (2002) 第 071088 号

责任编辑: 赵丽松

印 刷: 北京天宇星印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 14 字数: 282 千字

版 次: 2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷

印 数: 6 000 册 定价: 25.00 元

版权贸易合同登记号 图字: 01-2002-2108

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

献给 Shirley Berger

前言

为什么要写一本关于设计嵌入式系统的书呢？因为我在这个行业中的工作经验，以及最近的教学经历，使我确信这样一本书是大家需要的。

多年前，我在一家半导体厂商担任开发工具营销主管，主要职责是与软件开发工具主管交流客户的需求。曾经有段时间，我的工作促使客户在其激光打印机上使用我们的 RISC 处理器。我精通各种开发工具套件，所以在使用户确信我能为其设计小组提供充分的技术支持前，我不得不先解决一些特殊问题。

由于没有为这种处理器设计的在线仿真器（In-Circuit Emulator），所以需要围绕 ROM 仿真器、JTAG 端口和逻辑分析仪开发整套扩展支持工具。当我向用户解释这一切后，他们只是摇了摇头。我马上意识到：麻烦大了！虽然用户告诉他们需要所有的资料，但是真正需要的是关于嵌入式系统开发的培训。毫无疑问，研发小组轻轻松松就能雇来一些初出茅庐的软件工程师。可另一个相当大的麻烦是很难在这些人中找到合格的嵌入式软件开发人员，除了 Wintel 或 UNIX 以外，他们还应当精通其他软件开发技术。我由此产生了一个想法：在嵌入式系统设计上也许会有不同层次的培训需求。

不久前，我在华盛顿-波泰尔大学（UWB, University of Washington-Bothell）讲授导论性课程。而现在，我正在讲授嵌入式系统概论。接下来，将是实验室课程。这个课程终会成熟完善，使学生获得在嵌入式系统开发上的专业知识。这本书的许多内容源自我在 UWB 的工作，学生们提出的反馈意见影响了这本书的很多观点。与这些学生和教师的交流增强了我的信心，师生们的确需要这样一本书。

本书的主要内容

这本书不同于软件设计的教材，也不同于嵌入式软件设计的教材（尽管不可避免地要涉及一些代码和编程问题）。

许多学生的 C++ 和 Java 编程水平都比较高。因此，本书不讲授软件设计方法，而讲授在嵌入式环境下进行软件开发的独到之处。嵌入式软件开发与基于主机的软件设计有很大区别，我写这本书的目的就是为了帮助嵌入式软件开发人员了解这些

不同之处，换句话说，当不能使用 `printf()` 或 `mallo()` 这样的函数时，应该怎么办。

因为本书是关于嵌入式系统设计的，当然会讨论设计问题，但重点讨论在普通应用软件设计中不会遇到的问题。这些问题中最重要的是处理器的选择。作为嵌入式开发工具营销主管，我的职责之一是帮助那些设计工程师及其主管增强选用本公司处理器的信心。为给定产品选择处理器所要考虑的问题有哪些呢？大多数刚入门的工程师仅仅了解奔腾系列或 SPARC 系列处理器的体系结构，而这本书将扩展设计工程师的眼界。处理器选择可能是“一掷定乾坤”的重大决策。在我经历过的许多案例中，不少公司因此输得精光。

本书的读者对象

如果你是我的学生

如果你正在 UWB 学习我的课程，那你很有可能要购买这本书。因为人们天生就知道面包的哪一面涂有黄油（笑）。由于校方的原因，我不能要求你购买这本书。但我想只是出于减少笔记量的理由，你也应当购买一本。这本书里包含有 400 张以上教学幻灯片的内容，与你在课堂上从头到尾看到的基本相同。仔细地阅读这本书，将帮助你掌握很多解决问题的方法。这些问题是嵌入式系统设计人员在日常工作中经常要面对的。当你毕业后开始找工作时，知道一些关于嵌入式系统的知识对你是有极大帮助的。

如果你是其他学校的或刚毕业的学生

尽管你不在 UWB 学习设计嵌入式系统，阅读这本书，对你未来的职业之路也是十分重要的。嵌入式系统是工业界中一个最大且增长最快的领域，但是具有嵌入式系统开发经验的毕业生数量却很少，对这个领域的小小见解都能帮助你在众多的求职者中脱颖而出。

作为一个人力资源经理，当我面试求职者时，我常“刷掉”那些说“我很灵活，我能干任何事情”这类话的候选人。然而，假如有一人这样说：“朋友，我用过你们的产品，它不过是个二流货色，你们为什么要这样设计跟踪定义菜单呢？”，这才是我想雇佣的人。假如你只是想通过阅读这本书学到一些行话，以便在面试时给对方留下个好印象，那么在这本书上花些时间阅读也是值得的。

如果你是正在从事研发工作的工程师或开发人员

如果你是一个有经验的软件开发者，这本书将帮助你把握全局。假如你的本性并不关心全局，你可能会问：“为什么我要把握全局呢？我只是一个软件工程师，只关心技术问题，让市场开发人员和管理人员去关心全局吧。我只要能随时把快速排序算法写得漂漂亮亮就行了。”当然，事实的确如此。但从过去的经验知道：作为身处职场食物链底端的开发人员，不会参与项目中的关键决策，可一旦出了什么差错，却要替人背黑锅。我就曾经花了很多时间努力弥补在项目生命周期之初由某个大人物做出的错误决策。有很多次，我忙于修改代码而不能参加女儿的演唱会。千万不要让人把麻烦塞给你！这本书将阐明并帮助你认识到早期决策的关键性。它还帮助你对项目的早期决策施加影响力。这些决策将会直接决定你的事业成功与否。你一定能掌握自己的命运！

如果你是项目主管

我收回刚才对管理人员和销售人员的坏话，这本书也适合他们。如果你是一个希望项目进度稳定、产品准时上市的管理人员，那么这本书能告诉你哪里有“地雷与绊脚石”。它能保证你成功吗？不能，但它有益无害。

我还努力和那些管理人员分享创意。例如，当我还是一个研发项目主管时，我使用一个简单的窍门帮助团结项目组并集中大家的力量。甚至在产品定义之前，就用些泡沫材料做一个盒子，这个盒子的外形与最终产品很相似，面板上写有项目代号，像 Gerbil 或其他有点幽默的名字。我们一下就有了一个产品原型的清晰“图像”。它出现在我们面前，吸引了我们的注意。接下来，我在家里为项目小组及其家人举办一个宴会。这些简单的方法帮助我把组员的注意力集中到首要的目标上。并且，它还帮助我组成了一个“外围支持组”，当需要每周工作 60 到 80 个小时的时候，我们就会得到家庭成员的支持。

外围支持是十分重要的，管理人员不应忽视它。作为一个研发主管，我相信我在工程师的个人生活中有着巨大的影响力。我可以大幅提升他们的工资，但若解雇他们则会影响其婚姻生活。我会尽我所能按时交付最好的产品，你也应当如此。

嵌入式设计人员和主管不应当一次又一次犯同样的错误。希望这本书会向你们展示一些我多年来学习到的最好的实践经验。由于嵌入式系统设计看上去就像位于电子工程和计算机科学之间的交叉地带，所以本书不会大量介绍我们曾学习或使用过的方法和工具。

本书的结构

在这本书的大部分地方，文字材料将按照经典的嵌入式系统生命周期模型来组织。这个模型多年来都能满足市场人员和现场销售工程师的需要。它的优势在于它能准确地表示嵌入式系统的开发过程。作为硬件和软件并行开发的嵌入式系统开发过程，当没有简单的模型能真正表达出嵌入式开发过程的精妙之处时，按照某种集成步骤进行表示的方法，看上去是抓住了这个过程的本质。

预备知识

本书假定读者熟悉应用软件开发的一些术语。知道一些关于 C 语言、汇编语言和基本数字电路的知识是很有帮助的，但也不是必须的。这本书只有很少几段描述了具体的 C 语言编码技术，并且对这本书的其余部分并不是不可或缺的。对大多数程序员来说，这些技术应当很容易理解。同样，为了理解本书中的那些以 Motorola 68000 汇编语言编写的例子，你也不必成为一个专业汇编语言程序员。假如你有足够的数理逻辑背景，知道 AND 和 OR 这些符号，就足以应付本书中关于数字电路方面的内容了。

简而言之，任何人只要在学校上过编程课程或者具有与之相当的编程经验，就不会对这些内容感到困难。

致 谢

我衷心感谢那些直接或间接帮助这本书面世的人们。Perry Keller 使我第一次感受到 ICE（在线仿真器）的乐趣与力量，我永远欠他一份人情。Stan Bowlin 是一位最好的仿真器设计大师，我很荣幸曾与他共事。我从 Stan Daniel Mann（一位 AMD 的同事）身上学到了很多，他帮助我弄清了一个产品到底是怎样设计出来的。

这份手稿由 Robert Ward、Julie McNamee、Rita Sooby、Michelle O'Neal 和 Catherine Janzen 进行了修改。Justin Fulmer 重新绘制了许多插图。Rita Sooby 与 Michelle O'Neal 完成了排版工作。最后，Robert Ward 和我的同事兼朋友 Sid Maxwell，复查了书稿中的技术细节。谢谢他们！

Arnold Berger

简介

在 1970 年，微处理器的问世使控制领域得到了解放。复杂的控制系统最初由简单的设备组成，以微处理器这样的部件作为主要的控制和反馈器件。如果你在剩余物资商店找到了一个老式电传打字机型的计算机终端，并把它的内部结构与现在的彩色喷墨打印机进行比较，就会发现它们有很大的区别。

汽车排放物在过去的 20 年间减少了 90%，这主要归功于在发动机管理系统中成功地应用了微处理器。开环燃油控制系统的特性由汽化器决定。现在的闭环燃油喷射系统使用了多个传感器，它能在相当宽的操作范围内控制排放物。如果没有微处理器作为控制部件，如此大的性能改变是不可能的。

微处理器现在已经广泛使用在汽车上。一辆新型豪华汽车可能装配了 70 个以上的微处理器。它们控制着从发动机火花塞、传动轴一直到避免由于关门时产生的压力而使司机耳朵胀痛的控制系统等众多部件。

如果没有机载计算机，F-16 战斗机就不能成为可靠的空中武器平台。当飞行员通过传统的控制系统控制飞行时，机载计算机在使飞机能保持在空中飞行的前提下尽可能满足飞行员的各种要求。

一架现代喷气式客机装配有 200 个以上的机载专用微处理器。

最能刺激微处理器性能提高的动力来自于电子游戏市场。尽管从 Nintendo、Sony 一直到 Sega 等产品都不是真正的嵌入式系统，但它们正在推动令人惊奇的技术进步。在微处理器论坛上，John Turley 描述了一种 200MHz 的 RISC 处理器，它将应用到下一代游戏平台上。在 25 美元的价位上，它能在一个时钟周期内完成一个 4 阶的矩阵变换。

为什么嵌入式系统与众不同

好了，以上内容的确给人留下了深刻的印象。现在就让我们来探讨一下是什么使嵌入式系统如此与众不同，至少不同到足以让某人写上一大本书。

了解典型的嵌入式系统与桌面 PC 之间的不同之处对我们认识嵌入式系统极有帮助。

- 嵌入式系统一般专用于特定的任务，而 PC 是一个通用计算平台
- 嵌入式系统得到多种类型的处理器和处理器体系结构的支持
- 嵌入式系统通常极其关注成本

- 嵌入式系统有实时约束
-

过一会儿，你将有充分的机会去学习关于实时的知识。而现在，只要知道实时事件是指当其发生时，系统必须马上进行处理(实时)的外部事件(相对于嵌入式系统而言)。

- 嵌入式系统使用的操作系统一般是实时操作系统 (RTOS)，而不是 Windows 9x、Windows 2000、UNIX、Solaris 或 HPUX 这样的操作系统
- 嵌入式系统软件故障造成的后果比桌面系统严重得多
- 嵌入式系统大多都有功耗约束
- 嵌入式系统经常在极端的环境下运行
- 嵌入式系统的系统资源比桌面系统少得多
- 嵌入式系统通常在 ROM 中存放所有的目标代码
- 嵌入式系统需要专用工具和方法进行设计
- 嵌入式微处理器包含专用调试电路

嵌入式系统一般用于特定的任务，而 PC 是一个通用计算平台

嵌入式系统微处理器的另一个名称是专用微处理器。它专用于某个特定的任务，或者是很少几个任务。如果要更改其任务，就要废弃整个系统并重新进行设计。用在移动心脏监视器或心肌震颤消除器上的处理器用不着运行电子表格或字处理软件。

然而对于通用处理器，比如此时正在使用的奔腾处理器，必须支持大量需要多种处理需求的应用软件。由于 PC 必须以较低的只适于支持轻量级应用软件的性能来支持复杂的软件，所以桌面处理器真是威力无比！

因此，不论是从经济上的，还是从工程学上的观点，都不会考虑把 AMD K6 或类似处理器应用到厨房角落里的咖啡机上。

这并不意味着某些人不会做出这样的蠢事。例如，一家法国公司用 AMD29000 处理器设计了一种真空吸尘器。AMD29000 是一种 RISC CPU，用它来驱动激光打印机引擎都绰绰有余。

嵌入式系统得到多种类型的处理器和处理器体系结构的支持

除 Intel x86 或 Sun SPARC 系列处理器平台外，大多数学习我的计算机体系结构或嵌入式系统课程的学生很少有在其他平台上编程的经验。第一次课外作业项目猛然惊醒了这些学生，他们开始研究行业资料并寻找合适的处理器。

这些学生正在学习由 40 多个半导体厂商提供的 140 种以上不同类型的微处理器。这些厂商为赢得下一代宽体喷气客机或下一代基于因特网的苏打水机的微处理器订单无时无刻地在互相竞争。

在第二章中，我们将学习更多关于处理器选择过程的内容。而现在只大概介绍一下可供选择的范围。

嵌入式系统通常极其关注成本

说“通常”是因为在“火星漫游者”中嵌入式处理器的成本可能不会排在其设计小组的成本限制清单的前 10 名。然而，如果能在每套发动机管理计算机系统中节约 10 个美分，那你就成为众多汽车厂商的英雄。成本在大多数嵌入式应用产品中是个令人头痛的问题。

在大多数情况下需要注意的成本是系统成本。处理器成本固然是一个因素，但是，如果使用高度集成的微控制器，而不是微处理器与分立外设的组合，就能节省许多印制电路板、连接器，并能使用功率更小的电源，从而节约相当多的成本。这方面的问题将在第三章中详细讨论。

嵌入式系统有实时约束

当笔记本电脑正在备份我的文章时，由于电脑正忙于做备份的相关工作，这时若我想用键盘，就会看到一个沙漏符号。如果我的笔记本电脑不是放在办公桌上，而是用在一架商务喷气机的雷达天线控制上，并且它的主要功能是提供碰撞警告，那么这种暂停就会造成一场灾难。

实时约束一般分为两类：时间敏感性约束和时间关键性约束。如果任务是时间关键性约束的，它必须在某个时间范围内完成，否则由其控制的功能就会失效。控制飞行器稳定飞行的控制系统就是个很好的实例。如果反馈速度不够快，其控制算法就会失效，飞行器就再也不能在空中翱翔。

而时间敏感任务就能以平和得多的方式处理超时引起的后果。例如，一台喷墨打印机平均处理周期从 4.5ms 延长到 6.3ms，后果不过是打印速度从每分钟 3 页下

降到每分钟 2 页。

嵌入式系统使用的操作系统一般是实时操作系统（RTOS）

就像嵌入式处理器一样，嵌入式操作系统也是多姿多彩的。我的学生必须在他们的课外作业项目中选择一个嵌入式操作系统。他们并不需要指定每个任务所需的 CPU 时间，RTOS 能自动进行安排。它对随时都要运行的任务授予最高优先级。如果其他的任务得不到足够的 CPU 时间，那一定是程序员惹的祸。

大多数商业嵌入式操作系统与桌面操作系统之间的另外一种区别是：有些事情不会在 RTOS 中出现，比如许多 Windows 用户经常碰到的蓝屏死机。

嵌入式系统软件故障造成的后果比桌面系统严重得多

还记得关于千年虫的大行动吗？这些行动的目的是要保证基于计算机的基础设施能持续正常运行。寻找与替换嵌入式处理器浪费了大量金钱，而这一切只是为了杜绝由于日期出错而造成的问题。

我们都知道由于医用放射治疗仪发生故障会造成什么样的悲惨结局。我们怎么知道什么样的代码才是完全没有错误的呢？怎样才能完全测试需要全天候正常运行的复杂软件呢？

然而，嵌入式系统对软件故障的容忍比桌面 PC 差得多，这是在此得出的一个重要结论。但这并不意味着嵌入式系统中的软件从来不会出错，在大多数嵌入式系统中一般都包括一些机制，比如看门狗定时器，它在软件失去控制后使之重新开始正常运行。在第九章我将提供更多关于软件测试的内容。

嵌入式系统大多都有功耗约束

对于许多读者，他们只见过在桌面 PC 中的奔腾或 AMD K6 处理器。这些 CPU 需要配备庞大的散热片和冷却风扇以防止处理器把自己“烤焦”。但这不是对桌面系统特别严重的限制，许多桌面 PC 有足够大的内部空间提供良好的通风能力。但许多嵌入式系统就没有如此充足的电能供应，比如有一种嵌入式系统，它挂在野狼的脖子上，而这些野狼成年累月地游荡在怀俄明州或蒙大拿州原野上，因此这种系统必须在一小块电池的支持下工作很长时间。

怎样使嵌入式系统凭着很少的电能就能长期运行呢？人们一般把这个任务留给硬件工程师。然而，责任的边界并不总是清晰明确。硬件设计人员可能有，也可

能没有软件体系结构约束的概念，处理器选择一般也不会由软件设计人员的参与下进行，所以需要软件和硬件设计人两者共同负起责任。如果整个系统设计有严格的功耗预算，系统中的处理器就必须在大部分时间处于“睡眠模式”，只有在定时器发出脉冲时它才会“醒来”。软件必须围绕这种特性进行设计。换句话说，系统完全是由中断驱动的。

功耗约束影响了系统设计决策的方方面面。功耗约束影响处理器的选择、内存体系结构的设计。系统要求的功耗约束很有可能决定软件是用汇编语言编写，还是用 C 或 C++ 语言编写，这是由于必须在功耗预算内使系统达到最高性能。功耗需求由 CPU 时钟速度以及使用的其他部件(RAM、ROM、I/O 设备等)的数量决定。

因此，从软件设计人员的观点看来，功耗约束可能成为压倒性的系统约束，它决定了软件工具的选择、内存的大小和性能的高低。

速度与功耗

几乎所有的现代 CPU 都使用互补金属氧化物半导体(CMOS)制作。CMOS 设备的简单门电路结构由两个 MOS 晶体管组成，一个是 N 型，一个是 P 型(这就是互补的意思)。N 型在上，P 型在下，它们像图腾柱一样堆叠在一起。两种晶体管的行为像个完美的开关。当输出是高电平，即逻辑电平 1，P 型晶体管就关闭，N 型晶体管就连接到输出以提供输出电压(5V 或 3.3V)，门电路把电压输出到电路的其余部分。

当逻辑电平是 0 时，情况正好相反。随着 N 型晶体管关闭，P 型晶体管在下一阶段接地。这种电路的拓扑结构的性质十分有意义，这使它吸引了来自功耗控制方面的注意力。如果电路是静态的(不改变状态)，电能损失相当少。事实上，如果这种设备没有其固有的、在室温或更高温度下产生的极少的漏电流，它的功耗几乎为零。

当电路切换时，正如在 CPU 中无时无刻不在发生的一样，事情就有所不同了。当门电路切换逻辑电平时，N 型和 P 型晶体管同时打开一段时间。在这段时间，电流通过这两个晶体管从电源线流到地线。电流流动意味着电能消耗，还意味着发出热量。时钟速度越快，每秒发生的切换次数就越多，也就意味着更多的电能损失。现在，想一想 500MHz 奔腾或其他处理器，它们包含有 1 千万个这样的晶体管，这样你就明白了为什么桌面 PC 消耗了如此多的能量。事实上，在现代处理器中，CPU 速度与其功耗的之间的关系在坐标轴上几乎是一条完美的直线。因此，那些把 CPU 超频，想从中榨出最后一丝计算能力的人都知道散热片与冷却风扇是多么重要。

嵌入式系统经常在极端的环境下运行

嵌入式系统无所不在。嵌入式系统也许运行在飞行器上、两极的冰天雪地中、亚利桑那州凤凰城外盛夏骄阳下的黑色卡马洛汽车里。尽管确认系统能在以上条件下运行通常是硬件设计人员的工作，但也意味着要同时考虑到硬件与软件。严酷的环境一般意味着更高的温度与湿度。军用设备必须达到一长串环境要求指标，并且还要用翔实的测试数据来进行证明。如果你曾惊讶于为什么很简单的处理器，比如英特尔公司的 8086，当它用在火箭上时，单价却达几百美元，那么你就想一想它的设计要求和工作环境。设备必须适合其工作环境，比如在太空中，这个事实常常决定某种设备能否在众多候选者中脱颖而出。

对环境的关注经常超过对其他方面的关注，比如功耗约束。由于处理器必须与外界隔绝，因此要把它们密封在硅胶涂层下，这将严重削弱其散热能力，所以处理器类型和速度也成为了制约因素之一。

不幸的是，在许多设计工作中，环境约束一般留到项目将近结束时才会考虑。当产品在测试时，硬件设计人员发现产品发热超出了预算，这常常意味着要降低其时钟频率以减少发热量，但这又使软件没有足够的时间完成其任务，问题又转化成要改进软件以提升执行效率。就这样一环套着一环，而产品发布时间也因此一拖再拖。

嵌入式系统的系统资源比桌面系统少得多

现在，我正在我的桌面 PC 上键入这份手稿，PC 同时正播放着一张老 CD。我拥有 256MB RAM、26GB 硬盘空间并且在 SCSI 卡上连接着 ZIP、JAZZ、软驱和 CD-RW 驱动器。在我面前的是漂亮的 19 英寸 CRT 显示器。我可以通过键盘和鼠标输入数据。仅仅数一数系统中的总线信号，就有以下几种：

- 处理器总线
- AGP 总线
- PCI 总线
- ISA 总线
- SCSI 总线
- USB 总线
- 并口总线
- RS-232C 总线

我支配着数不胜数的系统资源，轻轻松松就能完成工作。在我的指尖会拥有如

此强大的威力。这是对驱动计算机工业迅猛发展的技术与经济力量的一曲赞歌。

现在想一想控制录像机的嵌入式系统。很明显，它所能管理的资源比前面提到的桌面系统少得多。当然，这是因为它是专门用来执行很少几个确定任务的。由于设计时要考虑到经济性(整部录像机的零售价也不过 80 美元而已)，所以不能奢望录像机中的 CPU 还会具有通用性。这意味着这种类型的 CPU 只能管理很少的资源，它的成本更低、结构更简单。然而，这也意味着软件设计人员需要一遍又一遍地设计标准输入和输出(I/O)。输入与输出端口的数量常常受到很大限制，以致设计人员不得不串行化使用 1 到 2 个输入设备，使它们一刻也不得空闲。

嵌入式系统通常在 ROM 中存放所有的目标代码

甚至 PC 也不得不在 ROM 中存放部分代码。为了初始化自己，几乎所有的系统都要在 ROM 中存放部分代码(启动代码)。然而，大多数嵌入式系统必须把所有的代码都存放在 ROM 中。这意味着对存放在 ROM 中的代码映像的大小有极严格的限制。除此之外，由于代码是存放在 ROM 中，更需要改变的是系统设计方法。

举一个例子，在这个例子中，当打开嵌入式系统后，必须有代码初始化系统，以便其余的代码能够运行，这就是建立运行时环境，比如初始化 RAM、放置变量、测试内存完整性、使用校验和方法测试 ROM 完整性以及其他初始化任务。

从调试的观点看，代码在 ROM 中意味着一些事实。第一，调试器不能在 ROM 中设置断点。要设置断点，调试器必须能够用特殊指令取代用户指令，比如用自陷指令或软件中断指令。自陷指令迫使程序转移到调试器中适当的入口。在一些系统中，可以通过把应用软件加载到 RAM 中来避开这个问题。当然，前提是有足够的 RAM 空间来容纳所有的应用软件代码映像、存放变量并提供动态内存分配所需的

空间。

当然，作为商品社会，只要有需要，就会有人提供解决方案。在这种情况下，已经发展出支持嵌入式系统开发过程的专用工具套件，从而给我们提供了一条避开以上困境的道路。下面我们将讨论这种方法。

嵌入式系统需要专用工具和特殊方法

在第四章我们将详细讨论 8 种类型的工具。因为嵌入式系统在很多方面与其他系统有非常大的区别，所以不要对使用专用工具与方法来建立和测试嵌入式软件的做法感到吃惊。留意一下上面的例子，它就需要在存放于 ROM 中的指令的分界处设置断点。

ROM 仿真器

有几个公司制造硬件辅助产品，比如 ROM 仿真器(ROM Emulator)。图 1 展示了名为 NetROM 的 ROM 仿真器，它是由 Applied Microsystems 公司生产的。NetROM 是通用的仿真器工具。从目标系统的观点来看，ROM 仿真器就像标准 ROM 设备。它带有机械尺寸、电气特性与其仿真的 ROM 设备一模一样的连接器。连接器的作用是把来自目标系统上 ROM 插座的信号传递到位于电缆另一端的主电路中。此电路提供高速 RAM，它通过来自主机的单独信道进行快速写入操作。因此，目标系统把 ROM 仿真器看作 ROM 设备，但软件开发人员却把它视为 RAM 设备，他们能在其中轻易修改代码，并能通过调试器设置断点。

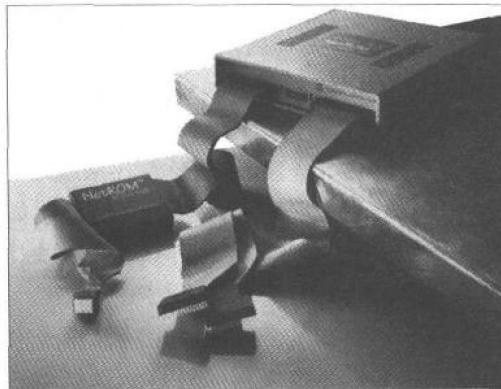


图 1 NetROM

在这本书中，术语“硬件辅助”指的是用来支持纯软件调试方案的附加专用设备。比如 Applied Microsystems 公司生产的 NetROM。

嵌入式微处理器包含专用调试电路

当今的嵌入式微处理器较过去的最大区别是芯片上都强制性地包含有专用调试电路。这一点与刚才讲到的嵌入式系统特性几乎是背道而驰的。刚才我还在喋喋不休地反复强调嵌入式系统的经济性问题，在每块微处理器中加上仅仅用于调试的

电路似乎是个愚蠢的做法。事实上，这种想法很是流行了一段时间。嵌入式芯片厂商最开始时是制造一种加入调试电路的特殊版本，把它们提供给工具开发商使用。最后，大多数厂商发现为所有芯片加入调试电路更经济。这并没有阻止他们继续限制人们了解其调试电路工作内幕，他们还是在每块芯片中为硬件辅助工具提供了调试“钩子”。

引人注目的是，这些厂商都认识到了具有片上调试电路是嵌入应用产品采纳其芯片的必要条件之一。也就是说，他们的芯片必须能提供很好的方案，解决嵌入式系统设计及调试问题，这样才会使面临上市压力的开发小组在考虑其嵌入式系统芯片时，采纳这些厂商的芯片。

小结

现在你已经知道嵌入式系统有哪些与众不同之处。在以下各章中，我们将一步步地在实践中考察嵌入式系统的设计过程。

开始几章我们将把重点放在设计过程本身上，将描绘嵌入式系统设计生命周期并阐明影响处理器选择的各个因素。接下来几章着重讨论用来建立、测试及调试完整系统的技术与工具。

我会在讨论过程中紧密结合当前内容发表一些评论，这些评论包括嵌入式系统生意经以及一些正在成长中的、极具潜力的新技术。

尽管一些工程师习惯于把设计看成一个理性的、需求驱动的过程，但在现实世界中，许多对设计过程有重大影响的决策常常不是由工程师做出的，而是由与项目需求无关的标准决定的。例如，在许多项目中，选用某种处理器的决策与所要解决问题的工程参数毫无关系。在许多情况下，设计小组的任务不过是照本宣科而已。这本书极有希望把那些身处逆境、疲惫不堪的工程师们重新武装起来。

目 录

前言.....	IX
简介.....	XIII
第一章 嵌入式设计生命周期.....	1
概述.....	1
产品定义.....	3
硬件与软件划分.....	6
迭代与实现.....	9
详细的硬件与软件设计.....	9
硬件与软件集成.....	10
产品测试与发布.....	13
维护和升级现有产品.....	15
小结.....	16
参考文献.....	16
第二章 选择过程.....	17
封装芯片.....	19
优越的性能.....	22
EEMBC.....	25
执行基准测试.....	26
可选的 RTOS.....	27
语言与微处理器支持.....	28
工具兼容性.....	28
性能.....	29
设备驱动程序.....	29
调试工具.....	29
标准兼容性.....	30