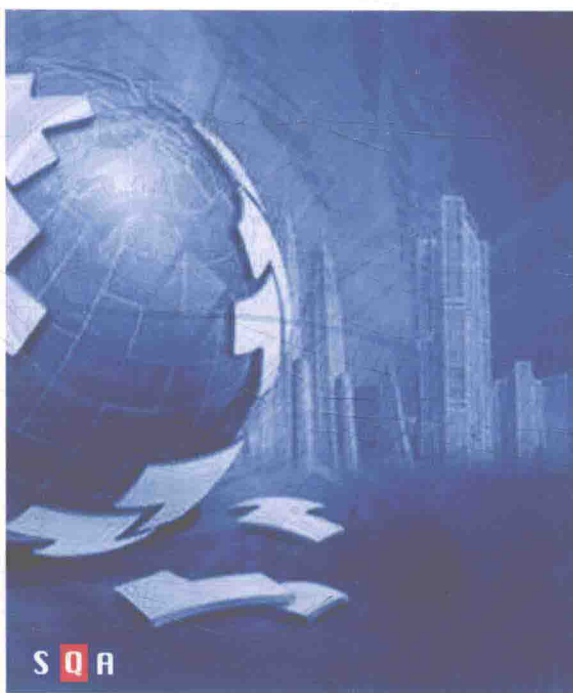


嵌入式软件测试与实践

- ◆ 嵌入式软件测试概述
- ◆ 嵌入式软件重要标准
- ◆ 嵌入式软件静态测试
- ◆ 嵌入式软件动态测试
- ◆ 嵌入式软件测试过程
- ◆ 嵌入式软件测试自动化
- ◆ 嵌入式软件可靠性测试
- ◆ 嵌入式软件安全性测试
- ◆ 嵌入式软件测试平台及环境
- ◆ 基于FPGA的嵌入式软件测试基础
- ◆ 基于FPGA的嵌入式软件测试技术



赵国亮 叶东升 董丽 赵琪 周健 辛中臣 编著



清华大学出版社

高等学校计算机应用规划教材

嵌入式软件测试与实践

赵国亮 叶东升 董 丽 编著
赵 琪 周 健 辛中臣

清华大学出版社

北 京

内 容 简 介

本书反映嵌入式软件主流测试需求,将技术分析与应用指导完美融为一体,技术先进,内容丰富。本书浓墨重彩地描述嵌入式软件测试基本概念、特点、分类、需求、体系架构和实用技术,介绍嵌入式软件的编码规范、质量度量、软件工程化、安全认证等相关标准,详述自动化测试、可靠性测试、安全性测试等重要技术,最后深入讲解FPGA软件的设计规范、测试标准、测试流程和测试方法。

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

嵌入式软件测试与实践 / 赵国亮 等编著. —北京:清华大学出版社, 2018

(高等学校计算机应用规划教材)

ISBN 978-7-302-48602-2

I. ①嵌… II. ①赵… III. ①软件—测试—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2017)第 253371 号

责任编辑:王 军 韩宏志

装帧设计:孔祥峰

责任校对:成凤进

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:27.25 字 数:629千字

版 次:2018年1月第1版 印 次:2018年1月第1次印刷

印 数:1~2800

定 价:79.80元

产品编号:070864-01

序一

《中国制造 2025》战略正在全面实施，工业 4.0 时代即将到来！物联网、智能设备、可穿戴设备、人机物融合技术迅猛发展；嵌入式系统的应用范围不断扩大，据估计终端数量将达到万亿级别，规模远超现有的个人电脑和手机。同时，人们开始关注嵌入式系统的质量问题；此类问题将造成严重的社会影响、经济损失和安全危害；例如，2016 年 10 月，由于联网的智能摄像头和数字摄像机被侵入，美国东部地区发生了大规模网络瘫痪事件。

高端装备和智能制造中的嵌入式软件是影响产品国际竞争力的重要因素，此类软件规模庞大，应用复杂，知识密集，技术繁杂，对安全性和可靠性有极高要求。但现实情况是，我国的嵌入式应用存在诸多问题：工业嵌入式软件开发基础和应用技术水平低下，工业嵌入式软件产品缺乏测试手段，技术人才严重不足，工业嵌入式软件质量监控手段不完善，企业重视程度不高。为保障高端装备和智能制造中的嵌入式软件质量，测试是主要手段！

本书内容丰富、技术先进，涉及嵌入式软件测试的标准、管理和技术，静态测试和动态测试技术，白盒测试、黑盒测试和测试自动化技术，针对嵌入式软件的可靠性和安全性的测试技术，嵌入式软件测试平台与环境，以及基于 FPGA 的嵌入式软件测试技术等。不仅介绍测试原理，还包括测试的具体指标和实施细节，并列举丰富案例，增强了可操作性。

本书作者赵国亮先生长期从事和负责航天关键软件的质量保证及测评工作，在国防领域及重大工程的嵌入式软件测试方面拥有丰富的工程技术与管理经验，对国家高端装备和智能制造软件的质量要求及相应的测试方法和技术有深入研究，并负责过多个重大软件测评项目，包括载人航天工程、探月工程、2008 年北京奥运会安保系统等。这些成果很好地融入本书，使得本书具有较好的专业性及实用性。

总之，《嵌入式软件测试与实践》是一本集各种嵌入式软件测试技术和工具应用于一体的嵌入式软件测试专业书籍，可作为高等院校高年级本科生及研究生的“嵌入式软件测试”课程教材，可作为企业及机构的培训用书，也可供嵌入式软件测试人员自学。

赵国亮 2017.8.16

国家科技奖励评审专家、航天中认技术总监

序二

嵌入式系统是一种“完全嵌入受控器件内部，为特定应用而设计的专用计算机系统”。它以应用为中心，以计算机技术为基础，软硬件可裁剪，可满足应用系统对功能、可靠性、成本、体积、功耗等的严格要求。随着现代信息系统的不断进步，嵌入式系统的应用已经渗透到各行各业，其系统也日趋复杂，逐步向计算、通信、控制、远程协作和自治合为一体的智能系统发展。

然而，伴随嵌入式系统的广泛应用与网络化发展，其配套的应用开发技术也日新月异，在硬件发展日益稳定的情况下，其软件复杂度呈指数级增长，这必然导致软件问题日益突出。随着嵌入式系统渗透到各类军用、民用设备中，其质量问题也越来越受到广泛关注。如何保证系统的正确性已成为需要迫切关注的问题，越来越多的人认识到嵌入式系统的深度测试势在必行。

相对于一般商用软件的测试而言，由于嵌入式系统的自身特性，嵌入式软件测试有其自身的特点和测试困难：如目标系统资源有限、与硬件系统紧密关联、运行平台多样化、交叉式的测试模式，以及系统的安全性可靠性高、系统运行结果的正确性严格受时间限制等；其测试策略与一般商用软件有很大的不同，相应测试技术的发展一直属于测试领域内的高端技术领域范畴，可以说嵌入式软件是最难测试的一种软件。任何人或组织进行嵌入式软件的测试时，都应深入考虑应用系统的特性，结合自身实际情况，选定合理测试策略和测试方案。

目前，市面上针对嵌入式系统测试方面的书籍相对较少，航天中认的赵国亮及其团队长期从事嵌入式软件测试技术研究及工程实施，在软件测试及软件质量保证方面积累了丰富的经验，从而为本书的编写夯实了基础。本书以高端装备与智能制造软件质量监督检验中心为技术依托，并创新性提出安全装备软件概念与评价体系，对高端装备与智能制造软件质量保证有非常重要的意义。

本书系统论述嵌入式软件测试的技术体系架构，讲述嵌入式软件及测试技术的关键部分，阐述嵌入式软件特性、嵌入式软件测试技术、嵌入式软件测试工具与应用等方面，从嵌入式软件测试基础、嵌入式软件测试方法与技术、基于FPGA的嵌入式软件测试等方面对嵌入式软件测试进行了详细论述，将技术分析与应用指导融为一体。可以说这是一本反映嵌入式软件主流测试需求、集各类嵌入式软件测试技术及应用为一体的软件测试专业书籍，对嵌入式软件测试从业人员及有志于学习研究嵌入式软件测试的读者大有裨益。



2017.8.31

北京大学软件工程国家工程研究中心主任、研究员

作者简介



赵国亮，特聘教授、硕士研究生导师、研究员、航天中认软件测评科技有限责任公司总经理、中国航天软件评测中心副主任，从事软件工程与软件评测相关工作近 20 年，具有扎实的理论基础和丰富的实践经验，负责过载人航天工程、探月工程、2008 北京奥运安保系统等大型工程软件测评工作。目前担任国家科技专家库专家，国家认监委检测认证技术委员会委员，中国广核集团有限公司软件质量保证专家咨询委员会委员，中国航天科工集团 706 所科技委委员。主要研究方向包括：软件工程、嵌入式及分布式软件测试、虚拟化测试、代码可靠性测试、云测试等，曾撰写《软件代码可靠性测试实践》等专著，在中文核心期刊及学术会议发表学术论文 10 余篇。

前 言

2015年5月国务院发布了《中国制造2025》行动纲领,明确提出了实施高端装备创新工程,高端装备主要包括国防装备、航空航天装备、先进轨道交通装备等。智能制造是高端装备的有效支撑,包括高端数控机床、工业机器人、增材制造以及工业互联网等。高端装备和智能制造已结合为一个有机整体,其中,规模庞大的高安全、高可靠嵌入式软件系统对高端装备和智能制造行业给予了有力支撑,如A380中的软件规模超过了1000万行,我国轨道交通中仅第三代列车控制系统CTCS3的软件规模就达到60万行。因此保障高端装备和智能制造中的嵌入式软件质量显得异常重要。

航天中认软件测评科技(北京)有限责任公司是在国家推动军民融合发展,加快推进经济发展方式转变和信息化与工业化融合,提升中国软件产业质量和效率的背景下,由中国航天科工集团第二研究院七〇六所(航天软件评测中心)、中国质量认证中心、中国家用电器研究院和北京航天科工军民融合科技成果转化创业投资基金合资建立。自成立以来,致力于将高标准、高可靠性要求的航天军用软件测评技术应用于民用软件市场,保障高端装备和智能制造业的嵌入式软件质量,提高软件水平。

本书力求从嵌入式软件测试的实际应用需求出发,借鉴赵国亮编写的《软件代码可靠性测试实践》以及蔡建平编写的《嵌入式软件测试实用技术》,对两本书中的嵌入式测试相关内容进行了整理,同时结合航天中认实际项目实验,在此基础上描述了嵌入式软件测试的解决方案和相关知识。

本书涉及三大方面内容,第一方面讲述嵌入式软件测试的基础知识,描述嵌入式软件特性及相关概念,针对国内外嵌入式软件编码规范、质量度量、软件工程化、软件测试等相关标准进行了详解;第二方面重点描述和分析嵌入式软件的测试方法和技术,包括静态测试和动态测试的测试技术、测试类型和测试工具实践,还涵盖了典型测试过程模型、测试自动化技术、可靠性测试、安全性测试相关的测试内容、测试方法和测试工具,此外针对嵌入式软件常用的测试平台和环境也进行了讲解;第三方面讲述基于FPGA的嵌入式软件测试,包括FPGA软件的设计规范和测试标准,以及测试过程中用到的测试技术方法和手段。

本书配套网站提供电子教案。读者可访问<http://www.tupwk.com.cn/downpage/>,输入本书中文书名或ISBN下载,也可直接扫描本书封底的二维码下载。

本书的完成得益于许多人的奉献及参与,赵国亮和叶东升对书籍的整体架构、思想及方向进行了把控,并对本书进行了审阅、修改和定稿。书中各章内容由董丽、赵琪、周健和辛中臣完成,第1章、第4章、第5章、第6章由董丽负责资料收集、起草和编写,第3章、第7章由赵琪负责资料收集、起草和编写,第2章、第8章、第9章由周健负责资料收集、起草和编写,第10章、第11章由辛中臣负责资料收集、起草和编写。本书很多

内容是对学术论文、其他书籍和互联网资料改编的结果，读者可以从参考文献中发现，本书参考、整理、借鉴和引用了大量文献，但很多网上资料由于转载或引用频繁无法找到原出处，在参考文献中无法明确标注。最后，我们感谢在书籍编著过程中对我们提供帮助及支持的单位。

本书面向嵌入式软件开发和测试相关人员以及想要深入学习嵌入式软件测试内容的人员。本书也是航天中认软件测试工程的一项重要成果。目前国内关于嵌入式软件测试方面的书不多，很多都是翻译或者培训用的，里面讲的仍是普通软件测试的内容。本书以传统的软件工程化思想为主，在大量嵌入式软件测试实例的基础上，系统介绍嵌入式软件测试的基本概念、相关标准、测试思想、测试技术方法和测试工具，力图反映嵌入式软件测试领域的最新发展和当前测试技术内容，深入浅出，使读者更容易理解和学习掌握。

赵国亮

2017.7.16

目 录

第 1 章 嵌入式软件测试概述	1	3.2.1 编码规则检查	95
1.1 嵌入式软件及其特性	2	3.2.2 安全性检查	115
1.1.1 嵌入式软件概述	2	3.3 嵌入式软件代码分析	119
1.1.2 嵌入式软件设计方法	3	3.3.1 错误检测主要内容	120
1.1.3 嵌入式软件测试的意义 和要求	8	3.3.2 运行时错误检测	121
1.2 嵌入式软件测试基础	11	3.4 软件代码审查	132
1.2.1 嵌入式软件静态测试	12	3.5 静态测试工具实践	136
1.2.2 嵌入式软件动态测试	12	3.5.1 软件静态测试工具	136
1.2.3 嵌入式软件测试管理	13	3.5.2 代码分析工具	140
习题与思考题	22	习题与思考题	146
第 2 章 国内外嵌入式软件相关 重要标准及规范介绍	23	第 4 章 嵌入式软件动态测试	147
2.1 国外标准及规范	23	4.1 嵌入式软件测试技术	147
2.1.1 嵌入式软件编码规范	23	4.1.1 嵌入式软件黑盒测试	147
2.1.2 软件工程化标准	32	4.1.2 嵌入式软件白盒测试	151
2.1.3 软件安全认证标准	37	4.2 嵌入式软件测试级别	156
2.1.4 软件测试标准	40	4.2.1 单元测试	156
2.2 国内标准及规范	42	4.2.2 集成测试	161
2.2.1 嵌入式软件质量度量	43	4.2.3 配置项测试	175
2.2.2 嵌入式软件编码规范	46	4.2.4 系统测试	178
2.2.3 软件工程化标准及规范	51	4.3 嵌入式软件动态测试工具	189
2.2.4 嵌入式软件测试指南	65	4.3.1 白盒测试工具	189
习题与思考题	69	4.3.2 黑盒测试工具	208
第 3 章 嵌入式软件静态测试	70	习题与思考题	212
3.1 嵌入式软件质量度量	70	第 5 章 嵌入式软件测试过程	213
3.1.1 嵌入式软件质量模型	71	5.1 嵌入式软件测试过程模型	213
3.1.2 嵌入式软件质量模型的 度量元	87	5.1.1 软件测试过程模型	213
3.2 嵌入式软件静态分析	95	5.1.2 软件代码可靠性测试 Z 模型	219
		5.2 嵌入式软件测试过程 管理工具 SQATest-TM	222
		5.2.1 SQATest-TM 简介	222

5.2.2	SQATest-TM 测试过程	
	管理功能	224
	习题与思考题	229
第 6 章	嵌入式软件测试自动化	230
6.1	嵌入式软件测试自动化技术	230
6.1.1	嵌入式软件测试自动化过程模型	230
6.1.2	嵌入式软件自动化测试工作原理	232
6.2	嵌入式软件自动化测试框架	233
6.2.1	“黑盒”测试的自动化	234
6.2.2	单元测试的自动化	237
6.3	嵌入式软件自动化工具 ETest	238
	习题与思考题	245
第 7 章	嵌入式软件可靠性测试	246
7.1	软件可靠性分析	251
7.1.1	软件失效模式及影响分析	252
7.1.2	软件故障树分析	257
7.2	软件可靠性测试	260
7.2.1	软件可靠性增长测试	262
7.2.2	软件可靠性验证测试	263
7.2.3	软件可靠性摸底测试	266
7.2.4	软件可靠性测试用例设计方法	266
7.3	软件可靠性评估	273
7.3.1	软件可靠性评估参数	273
7.3.2	软件可靠性模型	274
7.3.3	操作剖面的构造	277
7.4	军工软件可靠性工程	281
	习题与思考题	283
第 8 章	嵌入式软件安全性测试	284
8.1	嵌入式软件安全性分析	284
8.1.1	嵌入式软件安全性需求	284
8.1.2	嵌入式软件安全性定义	286
8.2	嵌入式软件安全性测试要求	288
8.2.1	嵌入式软件安全性过程	288
8.2.2	嵌入式软件安全性测试类型	292
8.3	嵌入式软件安全性测试技术及方法	294
8.3.1	嵌入式软件安全性测试技术	294
8.3.2	嵌入式软件安全性测试方法	300
8.4	安全装备软件认证	303
8.4.1	安全装备软件定义	303
8.4.2	安全装备软件认证及方法	304
	习题与思考题	308
第 9 章	嵌入式软件测试平台及环境	309
9.1	全实物仿真测试	309
9.1.1	全实物仿真测试概念	310
9.1.2	全实物仿真测试应用	310
9.2	全数字仿真测试	311
9.2.1	全数字仿真测试技术及环境	311
9.2.2	全数字仿真测试开发支持	312
9.3	半实物仿真测试	314
9.3.1	半实物仿真测试技术及环境	314
9.3.2	半实物仿真开环测试	316
9.3.3	半实物仿真闭环测试	317
9.3.4	半实物仿真测试的技术发展	317
9.4	仿真测试策略	318
9.4.1	基于仿真环境的测试策略	318

9.4.2 基于交叉调试的测试策略.....	321	11.2 FPGA 静态测试.....	378
9.5 基于虚拟仪器的嵌入式软件测试.....	324	11.2.1 FPGA 编码错误检测.....	379
9.5.1 虚拟仪器的概念.....	324	11.2.2 静态时序分析方法.....	379
9.5.2 虚拟仪器技术在测试中的应用.....	327	11.2.3 时序约束分析.....	381
9.6 基于虚拟样机的嵌入式系统仿真测试.....	332	11.2.4 功耗分析测试.....	384
9.6.1 虚拟样机技术的概念.....	332	11.2.5 逻辑资源分析测试.....	385
9.6.2 虚拟样机及虚拟测试.....	333	11.2.6 静态测试工具介绍.....	385
9.7 嵌入式仿真测试工具.....	335	11.3 FPGA 仿真测试.....	396
9.7.1 全数字仿真测试工具.....	335	11.3.1 FPGA 仿真测试技术介绍.....	396
9.7.2 嵌入式应用仿真工具.....	342	11.3.2 FPGA 仿真测试验证方法.....	396
习题与思考题.....	346	11.3.3 测试激励及测试脚本要求.....	397
第 10 章 基于 FPGA 的嵌入式软件测试基础.....	347	11.3.4 仿真测试内容与测试工具.....	398
10.1 FPGA 介绍.....	347	11.3.5 仿真测试工作准备.....	401
10.1.1 FPGA 原理.....	347	11.3.6 仿真测试工作步骤.....	401
10.1.2 FPGA 应用基础.....	348	11.3.7 仿真测试工具使用流程.....	403
10.1.3 FPGA 系统开发过程.....	349	11.4 FPGA 软件形式验证.....	407
10.1.4 FPGA 系统开发工具支持.....	353	11.4.1 FPGA 软件形式验证介绍.....	407
10.2 FPGA 设计规范要求.....	361	11.4.2 形式验证内容与验证工具.....	407
10.2.1 语法设计要求.....	361	11.4.3 形式验证测试条件与测试准备.....	408
10.2.2 通用设计要求.....	363	11.4.4 等价性检查工作步骤.....	410
10.2.3 跨时钟域设计要求.....	367	11.4.5 等价性检查工具使用流程.....	410
10.2.4 常见设计缺陷.....	368	11.5 SOC 系统验证.....	416
10.3 FPGA 相关标准规范.....	370	习题与思考题.....	417
10.3.1 Actel 器件编码规范.....	371	参考文献.....	418
10.3.2 Altera 器件编码规范.....	374		
习题与思考题.....	375		
第 11 章 基于 FPGA 的嵌入式软件测试技术.....	376		
11.1 FPGA 测试流程及基本方法.....	376		

第1章 嵌入式软件测试概述

本章导读

本章概述嵌入式软件测试, 主要介绍嵌入式系统的自身特点、嵌入式软件设计与开发、嵌入式软件测试需求及嵌入式软件测试类型, 重点阐述嵌入式系统的特点、嵌入式软件静态测试与动态测试等, 并概要介绍嵌入式软件设计方法。

本章涉及的知识要点:

- 掌握嵌入式软件的定义、特点与分类;
- 理解嵌入式软件设计方法;
- 理解嵌入式软件测试管理;
- 掌握嵌入式软件的静态测试、动态测试要素;
- 理解嵌入式软件测试过程。

随着经济的发展和科技的进步, 信息技术的迅猛发展使人类进入了数字化时代, 而伴随着计算机技术发展起来的嵌入式技术也得到了巨大发展, 改变了人们的日常生活。嵌入式系统作为提高国民经济和生产力的有效手段被广泛应用于各个领域。根据 IEEE(国际电气和电子工程师协会)的定义, 嵌入式系统(Embedded System, ES)是“用于控制、监视或辅助操作机器和设备的装置”。嵌入式系统还可以定义为“以应用为中心、以计算机技术为基础, 软件硬件可裁剪, 功能、可靠性、成本、体积、功耗要求严格的专用计算机系统。”嵌入式系统由嵌入式硬件和嵌入式软件两部分组成, 硬件是支撑, 软件是灵魂。

嵌入式软件是基于嵌入式系统设计的软件, 它是计算机软件的一种, 同样由程序及其文档组成, 可分为操作系统、支撑软件、应用软件三类, 是嵌入式系统的重要组成部分。嵌入式操作系统负责嵌入系统的全部软、硬件资源的分配、调度工作, 控制、协调并发活动; 支撑软件是用于帮助和支持软件开发的软件, 通常包括数据库和开发工具; 应用软件是针对特定应用领域、基于某一固定的硬件平台、用来达到用户预期目标的计算机软件。嵌入式系统中软件所占的比重越来越大, 相应的由嵌入式软件失效而引起的故障比例也越来越大。因此, 提高嵌入式软件的质量成为目前迫切需要解决的问题。众所周知, 软件测试是保证高质量、高可靠性软件的重要手段, 对嵌入式软件来说更是如此。实践证明, 在软件开发过程中, 通过软件测试来发现其中的缺陷是提高软件质量和可靠性的一种非常有效的方法。

1.1 嵌入式软件及其特性

1.1.1 嵌入式软件概述

嵌入式软件广泛应用于国防、工控、家用、商用、办公、医疗等领域，由于用户任务可能有时间和精度上的要求，因此有些嵌入式应用软件需要特定嵌入式操作系统的支持，嵌入式应用软件和普通应用软件有一定区别，它不仅要求其准确性、安全性和稳定性等方面能够满足实际应用的需要，而且还要尽可能地进行优化，以减少对系统资源的消耗，降低硬件成本。嵌入式软件产业发展极其迅猛，已成为软件体系的重要组成部分。嵌入式系统产品正不断渗透到各个行业，嵌入式软件作为包含在硬件产品中的特殊软件形态，在整个软件产业的比重日趋提高。各行业巨大的智能化装备需求拉动了嵌入式软件及系统的发展。与传统的通用计算机系统不同，嵌入式系统面向特定应用领域，根据应用需求定制开发，并随着智能化产品的普遍需求渗透到各行各业。随着硬件技术的不断革新，硬件平台的处理能力不断增强，硬件成本不断下降，嵌入式软件已成为产品的数字化改造、智能化增值的关键性、带动性技术。

嵌入式软件具有如下特点。

1. 系统内核小

嵌入式系统所能提供的资源相对有限，所以内核较传统的操作系统要小得多，以满足系统硬件的限制。

2. 专用性强

嵌入式系统通常面向用户、面向产品、面向特定应用。其中软件系统与硬件的结合非常紧密，一般要针对硬件进行系统移植。嵌入式系统中的 CPU 与通用型 CPU 的最大不同就是前者大多工作在为特定用户群设计的系统中。通常，嵌入式系统 CPU 都具有低功耗、体积小、集成度高等特点，能把通用 CPU 中许多由板卡完成的任务集成在芯片内部，从而有利于整个系统设计趋于小型化。在对嵌入式系统的硬件和软件进行设计时必须重视效率、去除冗余、针对用户的具体需求对系统进行合理配置，才能达到理想性能。与此同时，系统设计还受市场供求关系的影响。嵌入式处理器的发展也体现出稳定性，一个体系一般要存在 8~10 年的时间。一个体系结构及相关的片上外设、开发工具、库函数、嵌入式应用产品合在一起，可以构成一套复杂的知识系统。

3. 系统精简

嵌入式系统一般没有系统软件和应用软件的明显区分，不要求其功能设计及实现上过于复杂，这样一方面利于控制系统成本，同时也利于实现系统安全。

4. 高实时性操作系统

为提高执行速度和系统可靠性,嵌入式系统中的软件一般都固化在存储器芯片或单片机中,而不是存储于磁盘等载体中。由于嵌入式系统的运算速度和存储容量仍然存在一定程度的限制,另外,由于大部分嵌入式系统必须具有较高的实时性,因此对程序的质量,特别是可靠性,有着较高的要求。

5. 基于硬件和软件的嵌入式系统分类

因为嵌入式系统由硬件和软件两大部分组成,所以其分类也可以从硬件和软件角度进行划分。按表现形式(硬件范畴)来分类,嵌入式系统分为芯片级嵌入(含程序或算法的处理器)、模块级嵌入(系统中的某个核心模块)和系统级嵌入 SOC(System On a Chip)。按实时性要求(软件范畴)来分类,嵌入式系统分为非实时系统、软实时系统、硬实时系统。其中硬实时是指系统对响应时间有严格的要求,如果系统响应时间不能满足,就要引起系统崩溃或致命的错误。软实时是指系统对响应时间有要求,但是如果系统响应时间不能满足,不会导致系统出现致命的错误或崩溃。此外,还有其他分类方法,例如,按嵌入式系统软件复杂程度来分类,嵌入式系统分为循环轮询、有限状态机、前后台、单处理器多任务和多处理器多任务等分类系统。按应用来分类,嵌入式系统分为信息家电类、移动终端类、通信类、汽车电子类和工业控制类等。

1.1.2 嵌入式软件设计方法

考虑到嵌入式系统的软硬结合、时间限制、性能约束及对外交互的特点,实时应用通常是一个特殊过程,它常被硬件体系结构、软件体系结构、操作系统特性、应用需求、编程语言及开发和调试环境的变化所驱动。因此,嵌入式应用与通常说的计算机应用是有很大区别的,它不但要求满足功能需求,还要求满足性能需求,甚至性能需求放在第一位。而性能需求体现在限定时间的约束或代码大小的约束上,如一个处理算法能否满足严格的时间约束,是否需要使用特殊的硬件来完成其任务;操作系统是否满足我们进行高效的中断处理、多任务和通信的需求,编程语言是否支持并发程序设计等问题。总之,嵌入式系统的开发或嵌入式应用的设计和通常意义的计算机应用开发或计算机软件设计有很大的不同。嵌入式软件要求固态化存储,代码质量高、可靠性强,许多应用场合要求实时、多任务处理。因此在设计上,不但要考虑软件的设计,还要考虑硬件的设计;不但要考虑功能的设计,还要考虑性能的设计。随着技术的发展,应用需求越来越高,应用程序的体积急剧增长,复杂程度也越来越高,必须使用更好的分析设计方式,使应用程序结构性更好、占用内存更少、运行效率更高,同时更容易维护。

1. 硬件/软件协同设计技术

嵌入式系统的设计与当前硬件所能提供的支持(包括开发手段)及软件技术的发展紧密相连。应用先进的硬件设计和开发技术,我们不但可以获得所需的性能(如速度、电源、费用、可靠性等),还能取代部分软件,完成特定的功能。

另外,软件的设计离不开硬件的支持(如处理器、高级缓冲、并行能力等),而且高性能、多功能的硬件允许我们在软件开发和设计上采用好的方法,应用好的语言,使用好的工具,而不用把精力放在代码的紧凑和效率上,从而提高了软件开发效率,保证了软件质量。反之,应用好的软件设计技术和开发手段,我们可以充分发挥硬件的作用,提高系统的性能。

最后,由于嵌入式应用具有多样性,特别是嵌入式系统是硬件和软件的混合体,这表明在嵌入式系统的设计中以及在保证系统性能的前提下,要综合考虑硬件和软件的任务分工(包括考虑用硬件代替软件,或用软件置换硬件),要考虑硬件设计(如运用 ASIC、FPGA 及 HDL/VHDL 进行固件设计的技术)和软件设计的并行/交替和协同,要考虑硬件和软件的合成。因此,硬件和软件的协同设计在嵌入式系统设计中是一项很重要的技术。

2. 实时理论、算法、模型及相关设计概念

嵌入式系统的软件设计与一般的软件设计有很大的不同,它涉及更多计算机理论(如并发计算理论、队列理论、调度理论、可靠性理论等)和基于这些理论的算法(如资源访问控制算法、循环调度算法、优先级驱动算法、时钟驱动算法、EDF 算法、N-Version 算法等)以及有关的模型技术(工作负荷模型、周期性任务模型、多帧任务模型、处理器共享模型、漏斗模型等)。另外,实时设计在概念上,除了通常的软件设计概念外,有限状态机、并发任务等概念对于实时嵌入式系统的设计相当重要。上述这些是实时嵌入式系统设计的基础,对它们进行全面研究,并掌握它们,能够大大地提高嵌入式系统开发、设计和应用的水平。

3. 实时设计表示——模型技术

实时嵌入式应用的多样性是指:实时嵌入式应用的不同,所涉及的问题域也不同,因此采用的设计方法和设计手段也不同。但从设计角度看,无外乎功能、结构和时间。为在设计上清楚、无歧义地表示它们,也为了使设计文档化,甚至设计过程自动化,必须采用各种方法表示这些设计——即设计表示和模型。

目前常用的设计表示和模型有数据流/控制流图、任务结构图、MASCOT 图、结构图、结构图表、实体结构图、JSD 网络图、对象图、类结构图、状态转移图、状态图、Petri 网、离散事件模型、面向对象模型、功能模型等。对于这些设计表示或模型,除了考虑它们的各自特点、应用范围和自身局限性外,还要考虑它们的可用性及混合使用情况。另外,还可根据具体的应用情况,针对性地选择和设计有关模型(例如,数字信号处理 DSP 应用采用数据流模型,控制加强器应用采用有限状态机模型,HW 模拟采用模拟模型,事件驱动应用采用响应模型等)。大多数模型是用图表示的(一般采用节点、边构造其图),并有相应的语法、语义,也可进行层次表示,甚至还提供模型设计语言。

应用模型进行软件设计的关键是:

- (1) 对各种模型要有很深入的了解并能灵活地应用;
- (2) 要有基于模型的设计工具(甚至要支持可视化的设计);
- (3) 要提供设计所需的辅助工具;

- (4) 要能对设计过程进行管理。

总之,应用模型设计技术,要有配套的工具支持。

4. 实时设计过程

嵌入式软件开发和设计通常与通常的软件开发一样,具有生命周期和相应生命周期的各个阶段(概念、需求、概要设计、详细设计、实现、测试及维护),各个阶段可选择相关模型进行设计支持。整个开发和设计的过程可沿用瀑布模型、原型模型、螺旋模型及螺旋周期式模型等模式。一般的设计原则(如抽象、模块化、信息隐藏、完整性、维护性、可重用及可验证等)都适用于它。但嵌入式软件具有自身独有但又相当重要的设计概念,如有限状态机模型、并行、同步、时间约束以及空间约束等,这些设计概念直接或间接地影响着实时嵌入式软件的实时性能。实时嵌入式系统有关性能的满足是设计过程中要自始至终考虑和关注的问题,也需要有相应的工具支持(如性能设计表示或模型表示,性能分析、测试或评估工具等)。

5. 实时软件设计方法

软件设计用到设计方法是自20世纪70年代初开始的。首先是结构化编程方法(自顶向下,逐步求精);然后是基于数据的结构化方法(面向数据流设计,数据结构化设计),为数据库设计奠定基础的物理数据和逻辑数据分离方法;20世纪70年代末,提出了信息隐藏概念和并行系统设计方法。到了20世纪80年代,各种软件设计方法逐渐成熟并得到具体应用,如实时结构化分析和设计(RTSAD)方法、实时系统设计途径(DARTS)方法、杰克逊系统开发方法(JSD)以及面向对象设计方法(OOD)。进入20世纪90年代后,软件设计方法进一步得到发展,如综合设计方法(综合、集成不同的设计方法)、领域分析和设计方法(基于领域的可重用部件和体系结构)、硬件/软件协同设计方法以及知识库技术、形式化方法得到了具体应用。另外,实时嵌入式设计中的性能分析需要越来越迫切,实时性能分析技术也不断完善,这就为设计者从各种设计方案中选择满足性能要求的设计方案提供了便利。支持软件设计的CASE工具和软件开发环境不断推出,特别是启发式设计工具极大地方便了实时嵌入式软件的设计,声明和设计执行工具帮助设计者检验其设计,实时性能分析工具则分析其设计是否满足性能要求。

选择设计方法以开发者采用的设计策略为基础,例如,基于功能分解设计策略选择RTSAD方法,基于并行任务结构设计策略采用DARTS方法,基于信息隐藏设计策略选择OOD方法。当然,采用何种设计策略,与应用紧密相连,很难为嵌入式系统设计确定通用的设计原则。而且,有些方法尽管很流行或很受欢迎,但它不一定适用于嵌入式系统,如面向对象方法就不能很好地解决实时设计中涉及的中断和上下文切换、多任务和多处理器的并行处理、任务间的通信和同步、数据和通信速率大幅度的变化、时间约束、硬实时性能需求、异步处理问题,也不能很好地解决与操作系统、硬件及其他外部系统元素间的耦合等问题。相应地,我们必须研究或改进对象模型技术(OMT),使之能够处理实时问题,满足实时开发者的需求,OMT/Real-Time或UML-RT软件开发方法需要与UML完全向上

兼容。其中, UML-RT(UML for Real-Time)软件开发方法是由 ObjecTime 公司和 Rational 公司共同开发的, 它是对 UML 1.1 的扩展, 即与 ROOM(Real-Time Object-Oriented Modeling Language)合并而成。UML-RT 作为一个完整的实时模型标准, 能够辅助或指导软件开发人员设计/开发复杂的、事件驱动的、实时的软件系统(如电信、航空/航天、国防、自动控制等)。

另外, 为正确地获取用户真正的需求, 强调用户的真正参与, Ivar Jacobson 提出了使用实例(Use-Case)驱动的面向对象的软件方法。该方法从组成系统的实际操作入手。首先分析系统是如何使用的, 强调系统使用时与各种不同类型的用户交互时的状况(从使用实例出发)。得到使用实例模型后, 提出一套规范化方法针对类、对象等进行建模、设计等一系列工作。并将它们纳入软件工程中。实践表明, 这是解决用户参与的好方法。

随着嵌入式应用日益广泛, 涉及领域日益增多, 由此导致它的应用规模越来越大, 应用复杂程度越来越高。这样, 软件设计方法和支持软件设计的工具越来越重要。它们能够减小嵌入式应用的复杂性, 提高软件设计能力和效率。因此, 对软件设计方法(特别是实时嵌入式软件设计方法)和与之相关的设计支持工具进行研究是很有必要的。

6. 构件、重用及组合

人们常提到的模块化软件指的是软件基本上是由模块拼装而成。模块的显著特征就是它的结构简单、声明和实现分离、数据抽象、信息隐藏、功能封装。能用来拼装其他软件的模块我们常称为构件。模块和构件的区别是构件一般要遵循某个事先定义的接口标准。显然, 构件是可以多次使用或重新使用的。通过构件组装软件这种方式, 可以大大提高软件开发效率, 保证软件的质量。特别是对于实时嵌入式软件, 由于涉及大量与时间相关的算法, 如果将它们模块化或构件化, 并按特定的模式/定式和框架在软件开发中重用它们, 其意义不言而喻。另外, 基于构件进行软件开发, 有利于软件开发的自动化。

通过构件重用来开发实时软件的关键是组合问题。事实上, 实时软件系统的组合不能仅是功能的组合, 而应是功能、时间和容错的组合。组合时离线方法和在线方法都要考虑, 并且组合后的结果要经得起检验。这样, 组合后的系统满足嵌入式系统的高性能、容错要求, 即动态地满足其实时约束, 并在系统级别保证其性能要求, 同时还要能够在系统出现问题后能够降级使用系统。实际上, 应用组合技术需要其他配套的技术、工具和环境支持, 如容错技术、编程方法、分析工具及操作环境等。研究组合技术必须综合考虑, 只有这样, 才能够达到降低软件开发费用、缩短软件开发周期、提高软件质量的目的。

7. 设计定式和框架

前面介绍了实时嵌入式软件开发和设计可能涉及的理论、算法、设计表示、模型技术、设计过程、设计方法及构件重用技术等。但对于大多数设计者来说, 由于缺乏设计经验, 不知怎样(以及何时)才能最佳地应用这些技术。为此, 人们提出软件设计定式(Pattern)和框架(Framework)的概念和方法。所谓定式, 是指解决在特定上下文里开发软件所产生问题的方法, 这些方法可重新使用。它支持软件结构和设计的中重用。定式收集各种静态和动态结