

单片机与嵌入式系统应用



2001 (1-6)

Microcontrollers & Embedded Systems

合订本

CN 11-4530/V ISSN 1009-623X 邮发代号: 2-765 定价42元

清华大学出版社



MCU

权威 全面 实用

《单片机与嵌入式系统应用》

合订本

(2001年1~6)

《单片机与嵌入式系统应用》杂志社

— 北京 —

致 读 者

本合订本是在每期内容的基础上重新编排而成的。但是为了方便读者阅读和收藏，也为了在合订本中进一步体现我们选取文章的宗旨、方法，以及贯穿于各期中的主要思路，我们对合订本的内容作了适当的调整。其中，我们主要考虑的是以下几个方面：

一、本刊在各期中都尽量多地发表技术性文章，把其他内容的版面压缩到最小程度。一般地，每期80个版面中至少保持70个以上的版面是技术性文章。这就使本刊的内容很多，若在年底出版一个合订本，则会是一本砖头般厚的“大书”，不便于查阅使用。于是，我们决定把合订本分为两部分，1~6期出版一本，7~12期出版一本。1~6期的合订本在第六期出版的同时就推向市场，目的是让尽量多的读者及早使用；7~12期的合订本于年底推出。

二、期刊的内容无非是时效性弱的技术性文章和时效性强的信息性文章。此外，还有一些介于二者之间的广告。这些内容共同构筑了本刊的本质内涵，是缺一不可的。但是，合订本的主要目的是为了让读者保存和查阅，保留一些时效性很强的信息文章和广告就有点画蛇添足，因为这些内容一旦过时就基本上没有价值了。把这些用途不大，甚至过时就完全无用的东西放在合订本中与我们出版合订本的本意大相径庭。于是，我们在编排合订本时，就把这些内容取消了。但是，对技术性文章，包括编读往来中的几篇小技巧性文章都完整地予以保留。这样处理后，合订本的“含金量”就大大提高了。

三、月刊当然是按月推出的，每期的内容是相对独立的，但不是完全孤立的，各期的文章之间有着千丝万缕的联系，有的还是密不可分的。各期文章之间的这种联系，正是我们平时编辑思想的体现。为了在合订本中更明确地体现这种文章之间的联系，我们在编排合订本时，把文章次序作了适当的调整。总原则是，仍然按照栏目来分类，每个栏目中的文章则是以内容的相关程度来排序，而不是完全按照文章的发表先后排序。例如，在新器件新技术栏目中，关于AD μ C812的文章是先后在第一、二、三、四、六期发表的，但在合订本中则把他们调整到一起，从而使读者从合订本的目录中就可以感觉到这种文章的系列性。因此，我们建议读者在使用本合订本之前，先把目录认真地阅读一遍，不仅是检索，更重要的是能够对文章之间的相关性有一个明确的认识。除此之外，还有一些连载的文章，在合订本中干脆就变成了一篇大文章。例如，在学习园地中发表的《嵌入式C编程》是1~6期连载的，但在合订本中则是一篇文章，只是在标题上加了（一~六），表明是连载的，在内容上则完全没有连载的痕迹。这样处理后，原来刊物中加的一些编者按（编者注）就没有实质意义了，我们也就把这些内容删除了。

经过这样处理后，我们的合订本便不仅仅是“合订”了，而是有所综合，有所提高。服务读者是我们的职责和义务，及时推出合订本也是我们提高服务质量的手段。我们希望能够为读者做更多、更有益的工作，也期望得到读者的支持和指正。愿我们的期刊能够成为业界人士的乐园！

目 录

☆ 专家论坛

- 我国单片机 15 年 陈章龙 (1.1)
嵌入式计算机系统的展望 沈绪榜 (1.2)
嵌入式系统设计方法的演化
——从单片机到单片系统 窦振中 (2.3)
关于嵌入式应用开发技术 蔡建平 (3.6)
嵌入式实时操作系统的现状和未来
..... 何小庆 (3.13)
广义平台与平台模式 何立民 (1.15)
SOC 与单片机应用技术的发展
..... 李哲英 骆 丽 (2.18)
以 SoC 为中心的多学科融合与渗透
..... 何立民 (5.21)
片上系统设计与 EDA 夏宇闻 (4.23)
Home. Net 模型/架构研究 郭 旭 (6.26)

☆ 专题论述

- 嵌入式应用设计模式 马忠梅 (1.30)
嵌入式系统、智能家电及家居网络
..... 余永权 (1.32)
在单片机平台上建立规范化的液晶显示接口
..... 郭 强 (1.36)
按平台模式设计的虚拟 I²C 总线软件包 VIIC
..... 何立民 (2.38)
虚拟 I²C 总线软件包的开发及其应用
..... 周正干 李和平 (3.43)
嵌入式应用软件开发环境 δ OS/ λ Tool 的设计与实现
..... 郭 兵 熊光泽 陈丽蓉等 (3.47)
嵌入式软件系统的实时性设计
..... 陈丽蓉 熊光泽 雷航等 (3.50)
利用实时内核开发嵌入式多任务程序
..... 楚育军 刘守印 (6.54)
VxWorks 操作系统及实时多任务程序设计
..... 刘尉悦 张万生 邢 涛等 (5.57)
基于 VxWorks 的多 DSP (ADSP 21160) 系统的多
任务程序设计
..... 张万生 邢 涛 刘尉悦等 (6.60)
在 51 系列单片机上实现非抢先式消息驱动机制的
RTOS 许 俊 许克平 (5.63)
利用蓝牙技术实现家庭无线网络系统
..... 曾周末 张志荣 钱 莹 (5.66)

- 单片机通信性能分析和评价方法
..... 纪 松 李哲英 (2.69)
单片机智能化仪器嵌入式在线仿真设计方法
..... 徐爱钧 (5.72)
智能信息载体 iButton 及其应用
..... 王永红 凌志浩 (4.75)
MCS196 系列单片机分类与选型指南
..... 王晓君 刘彦民 张 英 (4.79)
基于 HART 协议的智能变送器设计
..... 刘焕成 刘智勇 尹征琦等 (6.82)

☆ 技术综述

- 单片机单总线技术
..... 黄宇飞 吴 江 秦旭等 (1.87)
DeviceNet 技术及其产品开发 贾 青 (1.91)
测控系统中实现数据安全的实用技术
..... 曾 刚 彭楚武 贺蓉等 (3.96)
电磁屏蔽效能稳定性研析
..... 张金铭 马 燕 (4.98)
恶劣环境中抗浪涌电压电路设计
..... 陶婷华 (3.103)
多 CPU 单片机系统设计
——在社区安防系统中的应用
..... 周建勇 何为民 (2.105)
利用 EPP 接口协议实现高速数据通信
..... 汪 洋 叶湘滨 (6.107)
模拟比较器的应用 (一~四)
..... 周航慈 涂水林 张福柳等 (2~5.110)
USB 2.0 特性及 USB 单片机 陈汝全 (5.124)
P87LPC764 位方式 I²C 总线编程规范
..... 陈 坚 朱 旻 (5.128)
PIC16F877 单片机的图形液晶显示模块接口技术
..... 严利平 卓清锋 刘和平 (6.132)
提高 PIC16C711 单片机片内 A/D 分辨率的方法
..... 李艳萍 蔡春富 (6.134)
通过无线收发芯片实现的单片机 DCTMS 技术
..... 杨 宁 (5.136)
嵌入式软件的测试方法和工具 邓世伟 (4.140)
80C196 系列单片机高精度浮点运算及数制转换子程
序 张克彦 (3.142)
彩色电视机中的 I²C 总线
..... 周艳琼 白 木 (4.147)

☆ 新器件新技术

新型数据采集系统芯片 AD μ C812	相韶霞 李刚 (1.150)
AD μ C812 的数据采集子系统及其 ISP 技术	颜荣江 (2.157)
基于 AD μ C812 和 WebChip/PS2000 的远程数据采集终端	张天宏 左洪福 黄向华 (3.160)
AD μ C812 在 LonWorks 智能节点上的应用	李桂桦 陈晓英 熊勇 (4.164)
AD μ C812 的串行外设接口 (SPI) 及其应用	李素芬 李刚 (6.166)
P89C51RX+/RX2 单片机及其 ISP/IAP 技术	梁合庆 (2.170)
新一代超低功耗单片机: MSP430F13x/14x 系列	潘卫江 胡大可 (3.177)
MSP430 单片机 Flash 存储器的特性及应用	潘卫江 胡大可 (4.181)
使用 MSP430F1121 中断须要注意的问题	韩炬 (1.184)
TOPSwitch-FX 系列单片开关电源原理	沙占友 王彦朋 王晓君 (1.187)
TOPSwitch-FX 系列单片开关电源的应用	沙占友 王晓君 王彦朋 (2.189)
TOPSwitch-FX 系列单片开关电源的快速设计法	沙占友 李春明 武卫东 (3.192)
Neuron 多处理器芯片及其应用	凌志浩 (2.195)
P87LPC764 单片机的 I ² C 总线显示电路	陈坚 (6.198)
PIC16F87X 单片机中断系统应用必须关注的问题	李学海 (5.201)
8031+PSD813 实现在线编程	孙树印 (1.206)
MCU 应用系统与 Internet 连接的一种新技术	赵葵银 唐勇奇 (2.210)
小尺寸、高可靠性的新型 μ P 监控芯片	Larry Barnes (4.213)
用 MAX3100 UART 构成 RS-485 通信网络	高亚男 傅隽 徐永根 (5.216)
用 MAX813L 设计单片机看门狗与电源监控电路	张良 (5.220)
新型电压监控芯片 TPS383X 的特性与应用	李保福 刘桂芝 (6.221)
IC 智能卡芯片 AT45D041 的原理及应用	曾周末 张志荣 钱莹 (5.224)
时钟芯片 DS1302 可靠起振的方法	屠运武 谷松 王甬生等 (1.228)

28SF040 闪速存储器及其在嵌入式系统中的应用	钱莹 张志荣 曾周末 (3.229)
HD7279A 的原理与应用	王改名 赫苏敏 王忠杰 (4.233)
W78E516 及其在系统编程的实现	崔冬建 王立德 史大北 (6.237)

☆ 应用天地

一种采用曼码调制的非接触 IC 卡读写程序编制	邢中柱 (1.241)
Temic-RF 卡读写器的设计	饶运涛 邹继军 (3.248)
加密 IC 卡保险柜控制器的设计	胡辉 肖洪兵 李叶紫等 (6.251)
在 VB 下 PC 机与 MCS-51 单片机的串行通信	朱旻 (1.254)
利用 VB 实现 PC 机与多单片机通讯	赵志国 冯建武 (2.256)
用 Win32 API 实现 PC 机与多单片机的串行通信	黄波 张晓晨 (5.259)
VC++ 中用 Active X 控件实现与单片机的串行通信	李志刚 王以法 (6.262)
8XC196 单片机串行口的 C 语言编程	王晓君 张英 唱春来 (4.264)
用 68HC908GP32 单片机实现 EPP 增强并口的接口技术	霍晓方 (2.267)
101/102 键盘信号分析与键盘接口的设计实现	石京民 (3.270)
用 PIC 单片机普通 I/O 脚检测模拟信号	张明峰 (4.272)
分布式测控系统远程通信一种智能中继器的设计	陈良光 金华频 (1.275)
射频接收系统晶体振荡电路的设计与分析	孙华芳 吴建辉 (1.278)
单片机协处理器在电力系统中的应用	陈倩诒 陈再清 (2.281)
一种用单片机控制的光谱数据采集系统	郑朝晖 张玉钧 刘文清等 (2.283)
智能化住宅小区安全服务系统	钱莹 曾周末 张志荣 (2.288)
利用单片机产生高压锯齿波和系统抗干扰设计	孙玉轩 (3.291)
便携打印机低温性能的实现	宋韶英 张尽红 史燕中等 (3.294)
由定时基准站管理的网型无线通信系统	李耀民 张向荣 (4.296)

采用数字式电位器的 PID 控制器
 何嘉斌 梁 枫 严 俊 (4.299)

MCS-51 单片机中计算正弦函数的一种方法
 彭宣戈 章 琛 (4.302)

用单片机实现野外作业仪表的加热和备用电池充电
 于振生 (4.306)

电饭煲模糊控制系统的设计 胡念平 (4.307)

按通用操作格式设计的多机通信软件
 陈爱萍 李朝建 (5.310)

快速折线式线性化方法 何学新 (5.314)

CPLD 在高速数据采集系统中的应用
 张 屹 扈 啸 刘国福 (5.315)

铁路雨量监测系统中的远程通信
 姚 云 赵 昆 (6.318)

高速铁路防风防雨安全监测报警系统
 彭势清 赵望达 徐志胜 (6.320)

基于分级通信模式的住宅小区自动电能计费系统的

研制 高玉芹 张允超 (6.323)

智能化电话报警器的研制
 鲁俊生 萧四友 阚先宏等 (6.326)

基于单片机的电力监控系统交流采样技术的实现...
 刘俊峰 李雪玲 (6.330)

☆ 学习园地

嵌入式 C 编程技术 (一~六)
 马志梅 (1~6.332)

☆ 编读往来

嵌入式系统基本概念探讨 (1.360)

80C196KC 使用注意事项两则 (4.360)

MCS-51 单片机地址指针及其应用 (5.361)

国际集成电路展中的单片机与嵌入式系统
 (6.362)

我国单片机 15 年

复旦大学 陈章龙

简介 结合全国性单片机学会成立 15 年的历程,回顾我国单片机的发展与现状,提出了我国的单片机应抓住嵌入式系统发展的契机,单片机的开发与应用应融入我国信息产业发展的大潮流中,使我国单片机开发应用更上一层楼。

关键词 单片机 嵌入式系统 单片机开发应用

我国单片机的发展若从 1986 年 11 月 2 日发起,并在 1987 年 10 月 27 日正式成立的全国性单片机学会算起,已走过了约 15 个年头。作为全国性单片机学会的负责人,回顾这前前后后的历程,有许多经验和教训值得吸取。我们可以把这 15 年的历程分成:探索(80 年代初)、起步(1986—1990 年)、开拓(90 年代上半时期)和发展(90 年代下半时期)等几个时期。

探索时期(80 年代初)

我国的单片机开发应用始于 80 年代。随着 8080/85、6800 和 Z80 等 8 位微处理器(MPU)的应用和开展,国内的一些大学和研究所对推出不久的 8048 单片机也进行了探索性的尝试。一开始,在仪器仪表、实验仪器和简易控制器等方面进行了应用,并对单片机的单板学习机(性能评估板)、ROM 仿真和在线仿真 ICE(In-Circuit Emulator)等各种开发手段进行了探索。使人们觉得与 8 位微处理器相比,在控制、仪器仪表和家用电器等嵌入式应用中,具有简单、方便和性能价格比高等优点。因此,刚开始从事单片机开发应用的人虽不多,但发展很迅速,很快在全国各地都有从事单片机开发应用人员。首次在复旦大学召开的全国单片机学术交流会(全国首届单片机学术交流会)只经过短短几个月的筹备,就吸引了 300 余人参加,提交了 130 篇论文,并发起成立了全国性单片机学会。在我国,单片机开始被人们有所认识。

起步时期(1986—1990 年)

此时期,单片机越来越吸引人们的关注。随着全国性单片机学会的建立,频繁的全国性单片机学术交流会与展示会吸引了更多的人员加入单片机开发应用的行列。通过多年的摸索,单片机的开发应用也开始成熟了。单片机的开发采用了与国际接轨的在线仿真器 ICE 的方式,并推出了具有中国特色的廉价的高性能的各种在线仿真器,这大大推动了我国单片机的开发应用(据不完全统计,国产的单片机开发工

具每年已达 2 万台左右)。在应用上,人们也选用了国际上流行的 8051 单片机。同时,单片机也开始进入了大学的课堂。我国单片机的应用开始真正起步了。

开拓时期(1990—1995 年)

随着单片机在我国应用的深入,单片机的普及也得到了广泛的开展,出版了各类单片机书籍。各地的单片机学习班、短训班如雨后春笋般地涌出,各地也纷纷成立了当地的单片机分会。至 1995 年,已有北京、上海、天津、江苏和广东等 20 个省市建立了相应的单片机分会,单片机应用技术也日臻成熟,改变了以往单片机应用一味采用总线外接存储器与 I/O 的方式(该方式虽然使用方便,但没有真正发挥单片机的“单片”特点),开始逐步接受了单片机应姓“单”的理念。特别自 1991 年在天津召开全国单片机学术交流会后,国内也适应潮流,开始推广国际上流行的单片机(如 MOTOROLA、NEC 和 PHILIPS 公司单片机)的应用,改变了 Intel 公司单片机一统天下的状况。这表明我国的单片机应用真正走上了轨道。

发展时期(1995 年—至今)

在此时期,我国单片机开发应用向纵深发展,单片机应用技术日臻成熟。在 8 位单片机作为主流单片机的同时,也开展了 16 位单片机、RISC 型单片机及各种特殊性能单片机的应用;单片机的 I²C、CAN BUS 及嵌入式 Internet 等新技术也得到广泛的关注和研究;模块化设计和高级语言编程等理念也逐步被接受。国产单片机在我国单片机应用的大好背景下,于本世纪末也浮出了水面,并开始得到批量应用。这表明在新形势下,我国单片机发展的步伐越来越快了。

新时期的新理念

我国 15 年单片机的历程,时间虽然不算长,但发展是很快的。从对单片机一无所知,到能独立自行设

计、生产和批量应用,这是一个不小的飞跃。然而,从我国信息产业发展的要求来看,我们应该有两个新的理念。第一个理念是:

- 逐步采用 16/32 位高性能单片机;
- 应采用 C、KJava 等高级语言来编程;
- 采用实时多任务 OS 及其平台进行开发;
- 采用模块化、团队设计方式来从事项目的开发应用。

第二个理念是从事单片机开发应用的人员一定要融入国家信息产业发展的潮流之中:

- 我们一定要加入国产单片机的设计、开发与

应用的队伍,为我国国产单片机的开发应用添砖加瓦。

· 新型嵌入式系统(如手持式信息机、PDA、智能家庭控制器和信息家电等)已成为我国的 IT 主流产品。我们一定要与企业合作,投入到其开发应用行列中。

时代在变化,单片机技术在发展,我们的理念也应不断更新。我们应抓住机遇,融入到我国信息产业发展的潮流中去,使我国单片机开发应用更上一层楼。MEMS

嵌入式计算机系统的展望

中国计算机学会微机专业委员会主任 中国科学院院士 沈绪榜

从使用角度来说,计算机可分为两类:一类是独立使用的计算机系统,如个人计算机、工作站等;一类是嵌入式计算机系统,它是作为其他系统的组成部分使用的。不管是哪一种计算机系统,要能够迅速地向未来发展,都必须满足五个简单而又基本的条件:一是经济性,计算机要很便宜,让更多的人能买得起;二是小型化,人们携带起来方便;三是可靠性,能够在一般环境条件下或者是苛刻的环境条件下运行;四是高速度,能够迅速地完成数据计算或数据传输;五是智能性,使人们用起来更习惯,对人们更有使用价值。不过,对不少应用来说,嵌入式计算机系统对这些基本条件的要求往往是更苛刻的。这可以从一些嵌入式系统的成功与失败的例子清楚地看出来。所以,这里就从这五个基本条件出发,展望一下嵌入式系统发展的未来。

就经济性来说,个人计算机的普及要算是一个典型的成功例子。可惜的是,Xerox 的管理人员于 70 年代初实施其无纸办公室的计划时,虽然首先开发了个人计算机,但他们认为这种计算机对一般人来说可能是太贵了,因而没有制造与发展个人计算机。自动付款机系统要算是一个典型的失败例子。它要求超市中的每件商品都有一个存贮商品价钱的芯片。当商品小推车经过记账出口时,一个无线电信号使芯片传出它的价钱信息以自动记账。当信用卡“扫过”时,就给出清单,这样记账时就不用排队了。这个系统未得到使用,就是因为芯片的价钱还是太贵了。芯片技术能降低电子产品成本的速度,就连当代电子学革命之父,2000 年诺贝尔物理奖获得者杰克·基尔比也没有想到,他在 1959 年发明的芯片技术,会将电子产品的成本降低到了百万分之一的地步。芯片技术的

这种神奇的作用,恐怕就是摩尔预言神奇般灵验的主要原因之一吧!难怪尽管发展芯片技术的耗资是惊人的巨大,发达国家还是力争在芯片技术的竞争中要永远保持领先的地位,以便能主宰世界信息技术的发展。

就小型化来说,需要人们携带的电子产品,如心脏起搏器,小型化要求就非常明显了。电子产品的小型化程度也是受芯片技术的发展水平限制的。尽管到 2015 年微米技术将达到它的物理极限,但仍然有许多应用还有待芯片技术的进一步微型化,使其功能密度的进一步提高。为此,MEMS 技术、系统芯片技术得到了发展。不仅如此,人们还在致力于纳米技术与生物技术研究,以期能使芯片技术有可能达到更高的微型化程度。例如,日本人的研究目标是“制造出能进入管道内进行检修的微型机械,能进入血管内进行手术的微型机器人,生产微型机器人,生产微型机械部件的超小型化工厂,确保日本在未来微机械加工领域的领导地位,在基础研究方面实现纳米技术的 Ogata 计划”。由于嵌入式系统是针对特定应用对象设计的,利用这一情况,嵌入式微处理器的设计一般都具有结构多样性与应用灵活性的两大特点。为了微型化,低功耗也是一个重要的性能指标。

就可靠性来说,对于常规条件下使用的家电产品等,现在的芯片技术已使产品的可靠性达到了非常令人满意的程度。但对太空、人体等特殊环境下使用的产品的长寿命要求,仍然不是一项容易实现的指标,还有待于芯片技术的进一步发展完善。

就高速度来说,应用对它的要求似乎没有止境,许多人工智能应用就是受到了计算速度的限制。对互联网来说,很多应用还是受到传输速度的限制,加

密解密就是一个很重要的例子。也许人们常常提到的量子计算技术才能解决这个高性能要求的问题。

就智能性来说,现代的芯片计算机可以进行逻辑、符号和语言处理等这些被认为是大脑左半球的功能,而且达到了人类自己都感到惊奇的程度。但如何实现与有生命的组织一样灵活而精细的信息处理能力,如发现缺陷、识别和改正错误之类的生物功能等问题,目前尚未找到有效的途径。更不用说,各种生命形式中的自律性、自组织、自更新和自发展等最典型的生物功能如何在当前的芯片计算机中实现了。硅基芯片是人类智慧的结晶,它正在不断地实现各种人类自身功能的延伸。模糊推理芯片确实使智能家电得到了大力发展,神经网络芯片则在模拟人类的学习功能上迈进了一大步。芯片是智能化的支柱,人们

不仅利用它研制智能的机器,改造客观的世界,而且也在利用它研制嵌入到人体内的产品,修补人体的缺陷,增进自身的健康。

综上所述,嵌入式系统的发展主要体现在芯片技术的进步上,以及在芯片技术限制下的算法与软件的进步上。今天正在开发的嵌入式系统,到底哪些明天定会取得应用上的成功,这是很难预料的。因为这不仅要取决于技术的因素,还要取决于社会的因素。虽然预测未来是困难的,但不管怎样,展望未来,明天的嵌入式系统将会比今天的更便宜、更小巧、更可靠、更高效而且更智能化,因为这毕竟是它赖以发展并为人类所最能接受的简单而基本的条件。所以从技术上来讲,沿着这五个简单而基本的条件努力,恐怕是势在必行不可忽视的。MES

嵌入式系统设计方法的演化 ——从单片机到单片系统

扬州大学理学院 赛振中

摘要 在介绍嵌入式系统设计方法变化背景的基础上,综述嵌入式系统设计方法的不同层次,从单片机应用到单片系统设计的演化,并提出了发展战略。

关键词 嵌入式系统 设计 单片系统(SOC) 硬件描述语言(HDL) IP 内核

一、嵌入式系统设计方法变化的背景

嵌入式系统设计方法的演化总的来说是因为应用需求的牵引和 IT 技术的推动。

1. 随着微电子技术的不断创新和发展,大规模集成电路的集成度和工艺水平不断提高。硅材料与人类智慧的结合,生产出大批量的低成本、高可靠性和高精度的微电子结构模块,推动了一个全新的技术领域和产业的发展。在此基础上发展起来的器件可编程思想和微处理(器)技术可以用软件来改变和实现硬件的功能。微处理器和各种可编程大规模集成电路、半定制器件的大量应用,开创了一个崭新的应用世界,以至广泛影响着并在逐步改变着人类的生产、生活和学习等社会活动。

2. 计算机硬件平台性能的大幅度提高,使很多复杂算法和方便使用的界面得以实现,大大提高了工作效率,给复杂嵌入式系统辅助设计提供了物理基础。

3. 高性能的 EDA 综合开发工具(平台)得到长足发展,而且其自动化和智能化程度不断提高,为复杂的嵌入式系统设计提供了不同用途和不同级别集编辑、布局、布线、编译、综合、模拟、测试、验证和器件

编程等一体化的易于学习和方便使用的开发集成环境。

4. 硬件描述语言 HDL(Hardware Description Language)的发展为复杂电子系统设计提供了建立各种硬件模型的工作媒介。它的描述能力和抽象能力强,给硬件电路,特别是半定制大规模集成电路设计带来了重大的变革。目前,用得较多的有已成为 IEEE 为 STD1076 标准的 VHDL、IEEE STD 1364 标准的 Verilog HDL 和 Altera 公司企业标准的 AHDL 等。

由于 HDL 的发展和标准化,世界上出现了一批利用 HDL 进行各种集成电路功能模块专业设计的公司。其任务是按常用或专用功能,用 HDL 来描述集成电路的功能和结构,并经过不同级别的验证形成不同级别的 IP 内核模块,供芯片设计人员装配或集成选用。

IP(Intellectual Property)内核模块是一种预先设计好的甚至已经过验证的具有某种确定功能的集成电路、器件或部件。它有几种不同形式。IP 内核模块有行为(behavior)、结构(structure)和物理(physical)3 级不同程度的设计,对应主要有描述功能行为的“软 IP 内核(soft IP core)”、完成结构描述

的“固 IP 内核(firm IP core)”和基于物理描述并经过工艺验证的“硬 IP 内核(hard IP core)”3 个层次。这相当于集成电路(器件或部件)的毛坯、半成品和成品的设计技术。

软 IP 内核通常是用某种 HDL 文本提交用户,它已经过行为级设计优化和功能验证,但其中不含有任何具体的物理信息。据此,用户可以综合出正确的门电路级网表,并可以进行后续结构设计,具有最大的灵活性,可以很容易地借助于 EDA 综合工具与其他外部逻辑电路结合成一体,根据不同的半导体工艺,设计成具有不同性能的器件。可以商品化的软 IP 内核一般电路结构总门数都在 5000 门以上。但是,如果后续设计不当,有可能导致整个结果失败。软 IP 内核又称作虚拟器件。

硬 IP 内核是基于某种半导体工艺的物理设计,已有固定的拓扑布局和具体工艺,并已经过工艺验证,具有可保证的性能。其提供给用户的形式是电路物理结构掩模版图和全套工艺文件,是可以拿来就用的全套技术。

固 IP 内核的设计深度则是介于软 IP 内核和硬 IP 内核之间,除了完成硬 IP 内核所有的设计外,还完成了门电路级综合和时序仿真等设计环节。一般以门电路级网表形式提交用户使用。

TI, Philips 和 Atmel 等厂商就是通过 Intel 授权,用其 MCS-51 的 IP 内核模块结合自己的特长开发出有个性的与 Intel MCS-51 兼容的单片机。

常用的 IP 内核模块有各种不同的 CPU(32/64 位 CISC/RISC 结构的 CPU 或 8/16 位微控制器/单片机,如 8051 等)、32/64 位 DSP(如 320C30)、DRAM、SRAM、EEPROM、Flashmemory、A/D、D/A、MPEG/JPEG、USB、PCI、标准接口、网络单元、编译器、编码/解码器和模拟器件模块等。丰富的 IP 内核模块库为快速地设计专用集成电路和单片系统以及尽快占领市场提供了基本保证。

5. 软件技术的进步,特别是嵌入式实时操作系统 EOS(Embedded Operation System)的推出,为开发复杂嵌入式系统应用软件提供了底层支持和高效率开发平台。EOS 是一种功能强大、应用广泛的实时多任务系统软件。它一般都具有操作系统所具有的各种系统资源管理功能,用户可以通过应用程序接口 API 调用函数形式来实现各种资源管理。用户程序可以在 EOS 的基础上开发并运行。它与通用系统机中的 OS 相比,主要有系统内核短小精悍、开销小、实时性强和可靠性高等特点。完善的 EOS 还提供各种设备的驱动程序。为了适应网络应用和 Internet 应用。还可以提供 TCP/IP 协议支持。目前流行的

EOS 有 3Com 公司的 Palm OS、Microsoft 公司的 Windows CE 和 Windows NT Embedded4.0、日本东京大学的 Tron 和各种开放源代码的嵌入式 Linux 以及国内开发成功的凯思集团的 Hopen OS 和浙江大学的 HBOS。

二、嵌入式系统设计方法的变化

过去擅长于软件设计的编程人员一般对硬件电路设计“敬而远之”,硬件设计和软件设计被认为是性质完全不同的技术。

随着电子信息技术的发展,电子工程出身的设计人员,往往还逐步涉足软件编程。其主要形式是通过微控制器(国内习惯称作单片机)的应用,学会相应的汇编语言编程。在设计规模更大的集散控制系统时,必然要用到已普及的 PC 机,以其为上端机,从而进一步学习使用 Quick BASIC, C, C++, VC 和 VB 等高级语言编程作系统程序,设计系统界面,通过与单片机控制的前端机进行多机通信构成集中分布控制系统。

软件编程出身的设计人员则很少有兴趣去学习应用电路设计。但是,随着计算机技术的飞速发展,特别是硬件描述语言 HDL 的发明,系统硬件设计方法发生了变化,数字系统的硬件组成及其行为完全可以用 HDL 来描述和仿真。在这种情况下,设计硬件电路不再是硬件设计工程师的专利,擅长软件编程的设计人员可以借助于 HDL 工具来描述硬件电路的行为、功能、结构、数据流、信号连接关系和定时关系,设计出满足各种要求的硬件系统。

EDA 工具允许有两种设计输入工具,分别适应硬件电路设计人员和软件编程人员两种不同背景的需要。让具有硬件背景的设计人员用已习惯的原理图输入方式,而让具有软件背景的设计人员用硬件描述语言输入方式。由于用 HDL 描述进行输入,因而与系统行为描述更接近,且更便于综合、时域传递和修改,还能建立独立于工艺的设计文件,所以,擅长软件编程的人一旦掌握了 HDL 和一些必要的硬件知识,往往可以比习惯于传统设计的工程师设计出更好的硬件电路和系统。所以,习惯于传统设计的工程师应该学会用 HDL 来描述和编程。

三、嵌入式系统设计的 3 个层次

嵌入式系统设计有 3 个不同层次。

1. 第 1 层次:以 PCB CAD 软件和 ICE 为主要工具的设计方法。

这是过去直至现在我国单片机应用系统设计人员一直沿用的方法,其步骤是先抽象后具体。

抽象设计主要是根据嵌入式应用系统要实现的功能要求,对系统功能细化,分成若干功能模块,画出系统功能框图,再对功能模块进行硬件和软件功能实现的分配。

具体设计包括硬件设计和软件设计。硬件设计主要是根据性能参数要求对各功能模块所需要使用的元器件进行选择 and 组合,其选择的基本原则就是市场上可以购买到的性价比最高的通用元器件。必要时,须分别对各个没有把握的部分进行搭试、功能检验和性能测试,从模块到系统找到相对优化的方案,画出电路原理图。硬件设计的关键一步就是利用印制板(PCB)计算机辅助设计(CAD)软件对系统的元器件进行布局和布线,接着是印制板加工、装配和硬件调试。

工作量最大的部分是软件设计。软件设计贯穿整个系统的设计过程,主要包括任务分析、资源分配、模块划分、流程设计和细化、编码调试等。软件设计的工作量主要集中在程序调试,所以软件调试工具就是关键。最常用和最有效的工具是在线仿真器(ICE)。

2. 第2层次:以 EDA 工具软件和 EOS 为开发平台的设计方法。

随着微电子工艺技术的发展,各种通用的可编程半定制逻辑器件应运而生。在硬件设计时,设计师可以利用这些半定制器件,逐步把原先要通过印制板线路互连的若干标准逻辑器件自制专用集成电路(ASIC)使用,这样,就把印制板布局和布线的复杂性转换成半定制器件内配置的复杂性。然而,半定制器件的设计并不需要设计人员有半导体工艺和片内集成电路布局和布线的知识和经验。随着半定制器件的规模越来越大,可集成的器件越来越多,使印制板上互连器件的线路、装配和调试费用越来越少,不仅大大减少了印制板的面积和接插件的数量,降低了系统综合成本,增加了可编程应用的灵活性,更重要的是降低了系统功耗,提高了系统工作速度,大大提高了系统的可靠性和安全性。

这样,硬件设计人员从过去选择和使用标准通用集成电路器件,逐步转向自己设计和制作部分专用的集成电路器件,而这些技术是由各种 EDA 工具软件提供支持的。

半定制逻辑器件经历了可编程逻辑阵列 PLA、可编程阵列逻辑 PAL、通用阵列逻辑 GAL、复杂可编程逻辑器件 CPLD 和现场可编程门阵列 FPGA 的发展过程。其趋势是集成度和速度不断提高,功能不断增强,结构趋于更合理,使用变得更灵活和方便。

设计人员可以利用各种 EDA 工具和标准的

CPLD 和 FPGA 等,设计和自制用户专用的大规模集成电路。然后再通过自下而上的设计方法,把用半定制器件设计自制的集成电路、可编程外围器件、所选择的 ASIC 与嵌入式微处理器或微控制器在印制板上布局、布线构成系统。

3. 第3层次:以 IP 内核库为设计基础,用软硬件协同设计技术的设计方法。

20 世纪 90 年代后,进一步开始了从“集成电路”级设计不断转向“集成系统”级设计。目前已进入单片系统 SOC(System on a chip)设计阶段,并开始进入实用阶段。这种设计方法不是把系统所需要用的所有集成电路简单地二次集成到 1 个芯片上,如果这样实现单片系统,是不可能达到单片系统所要求的高密度、高速度、高性能、小体积、低电压、低功耗等指标的,特别是低功耗要求。单片系统设计要从整个系统性能要求出发,把微处理器、模型算法、芯片结构、外围器件各层次电路直至器件的设计紧密结合起来,并通过建立在全新理念上的系统软件和硬件的协同设计,在单个芯片上完成整个系统的功能。有时也可能把系统做在几个芯片上。因为,实际上并不是所有的系统都能在一个芯片上实现的;还可能因为实现某种单片系统的工艺成本太高,以至于失去商业价值。目前,进入实用的单片系统还属简单的单片系统,如智能 IC 卡等。但几个著名的半导体厂商正在紧锣密鼓地研制和开发像单片 PC 这样的复杂单片系统。

单片系统的设计如果从零开始,这既不现实也无必要。因为除了设计不成熟、未经过时间考验,其系统性能和质量得不到保证外,还会因为设计周期太长而失去商业价值。

为了加快单片系统设计周期和提高系统的可靠性,目前最有效的一个途径就是通过授权,使用成熟优化的 IP 内核模块来进行设计集成和二次开发,利用胶粘逻辑技术 GLT(Glue Logic Technology),把这些 IP 内核模块嵌入到 SOC 中。IP 内核模块是单片系统设计的基础,究竟购买哪一级 IP 内核模块,要根据现有基础、时间、资金和其他条件权衡确定。购买硬 IP 内核模块风险最小,但付出最大,这是必然的。但总的来说,通过购买 IP 内核模块不仅可以降低开发风险,还能节省开发费用,因为一般购买 IP 内核模块的费用要低于自己单独设计和验证的费用。当然,并不是所需要的 IP 内核模块都可以从市场上买得到。为了垄断市场,有一些公司开发出来的关键 IP 内核模块(至少暂时)是不愿意授权转让使用的。像这样的 IP 内核模块就不得不自己组织力量来开发。

这 3 个层次各有各的应用范围。从应用开发角

度看,在相当长的一段时间内,都是采用前2种方法。第3层次设计方法对一般具体应用人员来说,只能用来设计简单的单片系统。而复杂的单片系统则是某些大的半导体厂商才能设计和实现的,并且用这种方法实现的单片系统,只可能是那些广泛使用、具有一定规模的应用系统才值得投入研制。还有些应用系统,因为技术问题或商业价值问题并不适宜用单片实现。当它们以商品形式推出相应单片系统后,应用人员只要会选用即可。所以,3个层次的设计方法会并存,并不会简单地用后者取代前者。

初级应用设计人员会以第1种方法为主;富有经验的设计人员会以第2种方法为主;很专业的设计人员会用第3种方法进行简单单片系统的设计和应用。但所有的设计人员都可以应用半导体大厂商推出的用第3种方法设计的专用单片系统。

结束语

目前,在我国3个层次的设计分别呈“面”、“线”、“点”的状态。习惯于第1层次设计方法的电子信息系统设计人员需要逐步向第2层次过渡和发展;第2层次设计方法要由“线”逐步发展成“面”;第3层次设计方法需要国家有关部门根据IT发展战略和规划,组织各方面力量攻关、协调发展。第3层次设计方法要由“点”逐步发展成“线”。MES

参考文献

- 1 Chen James C. Circuit Aging: A New Phenomenon For Soc Design. Electronic Design, 2000
- 2 梁合庆. 今日的嵌入式系统. 浙江大学学报(增刊), 1998

关于嵌入式应用开发技术

北京系统工程研究所 慕建平

摘要 在阐述嵌入式系统概念和特性的基础上,介绍嵌入式系统设计中的硬件和软件协同设计、实时理论算法模型、实时设计表示、实时设计过程、实时软件设计方法等。文中还对应用软件开发的最小编程环境和实时操作系统、实时数据库以及软件的应用测试进行了阐述。

关键词 嵌入式系统 嵌入式应用开发 软件测试

随着生产和科学技术的发展,特别是微电子技术的迅猛发展,原型技术和设计技术的不断成熟,以及计算机技术迅速地向非计算机领域的渗透,嵌入式计算机系统或者简称嵌入式系统越来越普遍地得到应用,也越来越显示出其商用的巨大市场回报。嵌入式系统上的投资或开销正在逐年上升。其中,硬件费用上升幅度慢慢减小,而软件费用上升幅度迅速增加。10年以后,嵌入式系统将在计算机应用领域占主导地位。

那么,什么是嵌入式系统呢?回答是:嵌入式系统是将计算机硬件和软件结合起来,构成一个专门的计算装置,完成特定的功能或任务。它是一个大系统或大的电子设备中的一部分,工作在一个与外界发生交互并受到时间约束的环境中,在没有人工干预的情况下进行实时控制。其中,软件用以实现有关功能并使其系统具有适应性和灵活性;硬件(处理器、ASIC、存储器等)用以满足性能甚至安全的需要。

嵌入式系统通常具有如下特性:

(1) 完成单一或一组紧密相关的特定功能;

(2) 具有高性能和实时的要求,并且这些要求正不断增加;

(3) 系统作为设备的一部分,其运行一般不需要人工干预;

(4) 系统的电源可靠性和安全性,通常是影响设计的重要因素;

(5) 处理器的选择是嵌入式系统设计的关键一步(系统的硬件尺寸、电源以及开发费用,如设计费用和开发环境的费用等,与处理器紧密相连)。

嵌入式系统的用途很广,如移动通信设备(手机)、数码相机或摄像机、洗衣机、微波炉、汽车刹车和引擎控制、打印机和复印机、多媒体应用设备、工业仪器和仪表以及医疗器械和设备等。对于各个嵌入式系统其要求也是千变万化的,既要求体积小(如手机)、性能高(如信号处理),又要求安全可靠(如核电站)等。从这里我们可以看到嵌入式系统应用的多样性和特殊性,以及与之对应的市场竞争的激烈性。

考虑到嵌入式系统软硬件结合的时间限制、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。它常常是被硬件体系结构、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的。因此,嵌入式应用与通常说的计算机应用有很大不同。它不但要求满足功能需求,还要求

满足性能需求,甚至把性能需求放在第一位。而这种性能的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,是否需要建造特殊的硬件来完成其任务;操作系统是否满足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与通常意义下的计算机应用开发或计算机软件设计有很大的不同。它不但要考虑软件的设计,同时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。事实上,嵌入式系统开发的最大问题是设计问题,而不是实现问题。

一、嵌入式系统设计方法

嵌入式系统产生某种动作,以响应外部事件的要求。为了完成这个功能,嵌入式系统在软件的控制下通过硬件来高速地获取数据,并进行处理,而后产生响应动作。整个过程是在严格的时间和可靠性的约束下进行的。由于这些约束相当苛刻,嵌入式系统通常只用于满足单方面的应用。

实时特征是嵌入式系统的主要特征。根据截止时间的要求,可将实时分为硬实时和软实时。硬实时要求的响应时间范围很严格,如响应不及时,将产生严重后果。例如,飞行控制、汽车自动刹车等,就要求必须在规定的时间内及时完成所有的处理。软实时的时间限制稍宽,响应时间晚点也可以接受,不致产生严重后果。例如,通信设备中的数字电话交换机等,只要求能够有效地处理系统负载。

嵌入式系统通常需要处理不同类型的任务:一类是周期性任务,即按照固定时间间隔执行的任务;另一类是非周期性的任务,常常是随机性任务,要求在任务出现的任意时刻都能进行相应的处理。一般对于非周期性任务的处理都有响应要求,这样当新的事件到来时,即使系统正在处理别的任务,也必须及时响应,从而导致了需要同时处理多个互相竞争的请求,包括中断请求。而对于一个连续的输入数据流处理,必须保证其数据不会丢失。除此之外,还要能够响应异步事件,这样,到达序列和数据容量是很难事先预测的。

嵌入式系统与外界有联系,因而有可能造成系统负荷过载。此时,要允许系统性能可以合理降级。由于资源有限,导致有些任务必须等待处理,甚至造成任务丢失。因此,对于不同的任务要根据它们的重要程度和时间约束,来决定其优先处理的次序。原则是保证重要任务。

尽管要求所有软件都必须可靠。但嵌入式系统

在可靠性、重新启动和故障恢复方面有更为特别的要求,而且通常有用于备份的内置冗余。另外,还要考虑系统的降级使用。

综上所述,实时嵌入式系统的设计相对于通常的计算机应用设计要复杂得多,涉及面也更为广泛,特别是嵌入式系统的物理尺寸朝着小巧、紧凑的方向发展,更增加了设计的复杂性,因此,需要对嵌入式系统的设计技术进行较为全面的研究。

1. 硬件/软件协同设计技术

嵌入式系统的设计与当前的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,不但可以获得所需的性能(如速度、电源、费用、可靠性等),而且还能取代部分软件,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,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,这表明了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、模型及有关设计概念

嵌入式系统的软件设计与一般的软件设计有很大的不同,它涉及到更多的计算机理论(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说实时嵌入式应用

的不同,所涉及到的问题域也不同,因此采用的设计方法和设计手段也不同。但从设计角度来说,无外乎功能、结构和时间。为了在设计上清楚、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,常用的设计表示或模型有:数据流/控制流程图、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、面向对象模型和功能模型等。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,有针对性地选择(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)和设计有关模型。大多数的模型是用图来表示的(一般采用节点、边构造其图),并且有相应的语法、语义,也可进行层次表示,甚至还提供模型设计语言。而用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够灵活地应用;②要有基于模型的设计工具(甚至要支持可视化的设计);③要提供设计所需要的辅助工具;④要能够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。其一般的设计原则(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、并行和同步、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,是基于数据的数据化方法(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。70年代末,提出了信息

隐藏概念和并行系统设计方法。到了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、形式化方法等得到了具体应用。另外,实时嵌入式设计中的性能分析越来越迫切,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。

选择设计方法是以开发者采用的设计策略为基础的,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,但它不一定就适用于嵌入式系统,如面向对象方法就不能很好地解决实时设计中所涉及到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、异步处理以及与操作系统、硬件及其他外部系统元素间的耦合等问题。相应地,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,即OMT/Real-Time或UML-RT软件开发方法,要求与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(Unified Modeling Language)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。

另外,为了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—

套规范化的方法,寻找类、对象等进行建模、设计等一系列工作,并将它们纳入到软件工程中。实践表明,这是解决用户参与的很好方法。

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,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。因为,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的。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。能够用以拼装其他软件的模块,我们常称之为构件。构件一般要遵循某个事先定义的接口标准。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,可以大大提高软件的开发效率,保证软件的质量。特别是对于实时嵌入式软件,涉及到大量的与时间相关的算法,如果将它们模块化或构件化,并按照特定的模式/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,有利于开发的自动化。

通过构件重用来开发实时软件,其关键是组合问题。事实上,实时软件系统的组合不能仅仅是功能的组合,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,并在系统一级上保证其性能要求,同时还要能够在系统出现问题后系统能够降级使用。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到的理论和算法设计、表示和模型技术以及设计过程、设计方法及构件重用技术等。但对于大多数设计者来说,他们缺乏设计经验,不知怎样并且何时才能最佳地应用这些技术,造成开发软件(特别是开发可重用软件)的困难。为此,人们提出软件设计定式(Patterns)和框架(Frameworks)的概念和方法。所谓定式表示的是解决在特定的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

结构和设计重用。定式收集各种静态和动态结构以及解决在特定领域里建立应用而产生问题等是成功方法,对于将要重用的小型结构(如构件)归档是很有用的。这些小型结构通常是对象结构的抽象,有经验的开发者用它们解决实时软件中的问题(如死锁、调度、同步和资源分配等)。研究和应用定式,开发者就能够开发比较成熟、可靠的软件。而框架则是可重用的、大致完整的应用,我们能够用它构造相关的应用。是一个集成的用来为一类相关的应用提供一个可重用结构的部件集合。它支持详细设计和代码的重用。

定式和框架两者都是通过收集成功的软件开发策略来推动重用,它们的主要差别是框架的着重点在具体的设计、算法及应用特定编程语言进行实现的重用上,而定式的着重点则是放在抽象设计和软件小型结构的重用上。

很显然,应用设计定式和框架技术,能够改进软件质量,缩短软件开发时间,也有利于开发的自动化。特别是实时嵌入式应用,涉及大量的各种各样的问题,而这些问题的解决是非常棘手的。对于新手,有一个漫长的学习和实验过程。如果将这些问题及解决这些问题的算法或模型以设计定式和框架的形式表示,并在其他实时嵌入式软件开发过程中重用它们,将会大大提高软件生产率 and 软件质量。目前,国外在这方面有很多的成功经验,我们应加以借鉴和研究,如:研究设计定式的描述和框架的结构;研究设计定式的设计和框架的构造;研究它们单独应用方式和综合应用方式;研究基于设计定式和框架的软件开发技术等。

二、嵌入式应用开发支持研究

嵌入式应用软件典型的开发方式是“宿主/目标机”方式:首先,利用宿主机上丰富的资源及良好的开发环境开发和仿真调试目标机上的软件。然后,通过串行口或网络将交叉编译生成的目标代码传输并装载到目标机上,并用交叉调试器在监控程序或实时内核/操作系统的支持下进行实时分析和调试。最后,目标机在特定的环境(如分布式环境)下运行。

1. 实时编程及最小编程环境

运用软件设计方法进行实时嵌入式软件的设计,应用编程语言完成实时嵌入式软件的实现。前面介绍了实时嵌入式软件的特征,如功能复杂/多样,分离部件的并行控制、与硬件打交道的机制、高可靠性和安全、满足响应时间要求等。所有这些特征最好是编程语言能够有相应的语言特征(如多任务支持)或手段/机制支持其实现,或者能够用操作系统提供的功

能或采用低级手段来满足实时嵌入式的特殊要求。

应用高级语言或实时编程语言开发实时嵌入式软件,最基本的要求是要有一个最小编程环境,如交叉编译、交叉调试器、宿主机和目标机间的通信工具、目标代码装载工具、目标机内驻监控程序或实时内核/实时操作系统等。其中交叉编译最为关键。考虑到我国实时嵌入式应用刚刚起步,交叉编译及工具主要以引进为主、改造为辅。而目前国际上的自由软件组织之一——GNU,提供的基于各种平台并公开源码的GCC/G++及Gnat等交叉编译和Gdb交叉调试器正逐步推广且渐渐被人们接受。我们可以在它们的基础上分析并改造成适合我国国情的交叉编译和相应的开发环境。其中要重点考虑编译的优化、运行、系统在裸机/操作系统上的运行(要求能够剪裁和配置等);同时还要考虑提供各种系统接口,提供数据库、图形、网络和分布计算等的联编和有关实时嵌入式应用开发的各种工具包。

2. 实时内核和实时操作系统

所谓实时操作系统就是能进行实时处理的操作系统。实时操作系统分为两类:一类是专为实时应用设计的专用操作系统,如:pSOS、VxWorks、VRTX32、RTEMS等,它们的核心是实时内核;另一类是在通用操作系统的基础上增加实时功能,如:实时Linux、实时Unix、实时POSIX等,其扩展部分是实时执行程序(execute)。实时执行程序的使用使得通用操作系统的实时性能变得可行,执行程序在行为上类似于一个应用程序,它负责执行大量的操作系统功能(尤其是那些影响实时性能的功能),它比通用操作系统更快、更有效。

根据内核所具有的功能,实时内核分为如下几类:

- (1) 毫微内核(有分配功能);
- (2) 微内核(毫微内核基础上增加了任务调度);
- (3) 内核(在微内核基础上增加了任务之间的同步);
- (4) 执行程序(自身具有存储块、I/O服务及其他复杂功能的内核);
- (5) 操作系统(提供了用户界面、安全/保密、文件管理系统等的执行程序)。

实时内核设计时要考虑轮询、协同、中断驱动及前/后台工作等需求,提供对任务、中断、时间和多处理器等的全面管理,并要求用高级语言实现(可移植性考虑)。设计出的实时内核要求紧凑、高效、专用、可移植性好。

实时内核的多处理器支持应包括处理同构和异构系统的能力,其内核程序应具有自动补偿不同处理

器之间体系结构的差别(如字节交换等)。这使得从一个处理器族到另外一个处理器族的转换变得异常容易,且不需要重新设计。

为满足实时软件开发的要求,我们必须自主地研制实时内核。该实时内核符合POSIX实时扩展标准接口(支持系统的开放性和可移植性要求),必须提供高性能环境以保证具有多线程和多任务能力、同族及异族多处理机系统、事件驱动、基于优先级、抢先调度、各种实时调度支持(如单调速率调度)、任务之间的通信和同步、优先级继承、响应中断管理、动态存储分配、高级用户配置和剪裁,甚至支持多平台、多目标机、多语言、网络和分布计算(如对多种通信协议、远程访问、CORBA技术等的支持)等特征的嵌入式应用需要。

另外,要研究通用操作系统实时化技术,如实时Linux以及实时POSIX的实现技术等。

3. 实时数据库

前面已经提到,实时系统的正确性不仅仅依赖于产生结果的逻辑特性,而且还依赖于这些结果的时间特性。像许多数据库处理系统一样,实时系统常具有数据库管理功能,而且能够进行数据并行处理的分布式数据库常常是首选。在数据库中,实时意味着除了通常对逻辑要求一致(如数据项值的约束)以外,还有事物处理完成时间的约束及数据访问和更新时间的约束等。例如,用于 C^3I 中的传感数据有效时间间隔很短(如目标跟踪),因此事物对它的访问及处理必须在很短的时间内完成,否则就会丢失数据。这就要求我们能够确定事物处理在最糟糕情况下的执行时间,而这实际上是一件非常复杂的事情,因为事物处理通常涉及到对各种资源(如CPU、I/O、缓冲和数据等)的访问,甚至多个事物之间还存在对资源的竞争从而影响了响应时间。另外,处理或查询的数据量大小对实时响应也有很大的影响。因此,为了保证事物处理在规定的时间内完成并且保持数据库的一致性,调度算法必须同时考虑硬件资源和数据资源的调度。由此可见,研制和开发实时数据库管理系统相对于普通的数据库管理系统所面临的挑战和制约要大的多。

为了满足时间约束和结果可确定的基本需求,不仅要通常的调度方法和事物管理进行重新设计,而且还要提出在通常数据库系统中没有考虑到或者在实时系统中要增加的新概念,以及对应的新技术。如实时情况下的数据截止时间和事物截止时间、事物调度、多任务处理、并行控制和资源管理、事物处理中断后的备份和恢复、容错服务质量和安全、实时数据库语言(如实时SQL)、实时分布式数据库的事物处理和多媒体处理(如视频会议)的体系结构(要求系统具

有开放性)、互操作技术和构件技术(包括功能、时间、容错及安全等方面),以及实时数据库应用等技术。

4. 实时分布式中间件

实时嵌入式信息应用系统正连续不断地增长。这些系统共同面临的技术问题是如何实时地访问和处理分布的、面向对象的数据(这些数据由多个用户通过高性能的客户机/服务器网络来共享),且要求不断增强其数据访问和处理功能。随着功能增强的需求加大和用户的增多,实时嵌入式信息系统的复杂程度也迅速增大。并且,这些系统通常是相当关键的,它们承担着重要使命。因此,迅速增长的分布式实时系统需要对各种服务质量 QoS(如响应时间、紧急程度及吞吐量等)进行全面的支持,如: C³I 系统、电讯、军用/民用航空、internet 上多媒体的音/视频流信号和图像等。

提高软件生产率和软件质量的需求激发了人们使用 DOC 中间件的愿望。DOC 中间件位于应用和底层(即复杂的分布式实时嵌入式系统中的操作系统、协议及硬件)之间。它能够使系统设计人员从部件位置、编程语言、OS 平台、通信协议和互连以及硬件依赖中解放出来,从而大大地简化了软件的开发。

理论上,中间件简化了实时应用的创建、构造及配置,且不产生时间和空间的消耗。但实际上,目前的技术已阻碍了对实时系统进行有效的、可预见的、且可升级的中间件的开发。尤其是基于 COTS 的 DOC 中间件常常缺少:

- (1) QoS 声明和规范的支持;
- (2) 与高速网络技术集成;
- (3) 效率、可预见性、可升级和优化。

这些限制了对性能有严格要求的系统的的速度,如视频会议和电子对抗等。

在分布式环境下,考虑实时应用的特点(响应时间、可靠性、正确性、完整性、并行性和分布式等),提出了支持中间件应用的实时 CORBA 的各种需求:

(1) 功能需求。如:追逐事件能力、全局时钟能力、各种调度方案的支持、基于查询优先级和继承优先级、多线程和客户/服务支持、多通信协议支持、失败通知支持等。

(2) 操作需求。使 CORBA 的功能具有时间特性,共享存储、垃圾回收,支持对远程对象大数量的并行连接等。

(3) 实现需求。占用内存小,线程安全和再入。

(4) 修改 ORB 和它的服务需求。动态调用接口、线程抽象层、对部件和服务基于优先级的查询、优先级继承、优先级升级、时间认知、所有功能最坏情况下的执行时间发布、有关协议、对象移动和透明分布

等。

(5) 为实时增加新的服务需求。新的实时服务(如调度服务)、定时异常服务(客户/服务两端)等。

(6) IDL 扩充需求。限时调用作为 IDL 的部分内容。

目前的关键问题是要突破对应上述需求的相关技术,研制支持这些需求的开发工具和环境,如:实时 IDL 的 stub 和 Skeletons、实时对象分配器、ORB 的 QoS 接口、实时 I/O 子系统、高速网络分配器等。另外,要对用于实时嵌入式系统中的面向对象中间件标准化。

目前,分布式实时系统的开发趋势是:编程采用可重用部件;对整个分布式应用部件采用远程方法调用;在异构环境下尽量定义标准的软件低层基础;在系统中确保 QoS 的需求。

三、嵌入式应用的测试技术研究

软件测试在整个软件开发过程中处于相当重要的地位,其测试费用占项目总费用的 25% 以上,对于嵌入式软件则花费更大。嵌入式应用软件的测试同人们通常使用的传统的软件测试相比有较大的差别,即除了要考虑和运用传统的测试技术外,还要考虑与时间和硬件密切相关的测试技术运用,如多处理器情况下的测试问题。因此,嵌入式应用的日益复杂导致嵌入式软件的设计、开发、测试和维护日趋复杂。因嵌入式应用的复杂性带来的问题和事故也越来越多,如:金融在线事物处理系统瘫痪、电话网络瘫痪和电话系统崩溃、电脑控制机场行李运送系统出现故障、商务飞机坠毁等。从这些事例中我们看到嵌入式应用的重要性和复杂性。为此,我们必须下大力气,研究和掌握嵌入式应用的测试技术,确保嵌入式应用安全、可靠。

1. 软件分析和仿真技术

实时嵌入式系统最大的特点是它具有一组动态属性,即中断处理和上下文切换、响应时间、数据传输率和吞吐量、资源分配和优先级处理、任务同步和任务通信等。所有这些性能属性可以很容易地说清楚,但要测试或验证它们(特别是时间确认)是很困难的。如系统的响应时间是否得到满足,系统资源是否足以满足计算需求,有关算法的执行速度是否足够快,甚至分布式环境下的与时间相关的服务质量是否达到要求等。这些是需要进行分析才能知道的。而对实时嵌入式系统进行分析需要建模和仿真,需要数学工具(如 McCabe 实时分析方法)的支持。状态图、排队和网络模型使得程序分析员能够评价总体响应时间、处理率以及其他时间和大小问题。形式化的分析工