

文章编号:1001-9081(2006)04-0757-04

嵌入式系统软/硬件协同设计技术综述

熊光泽,詹瑾瑜

(电子科技大学 计算机科学与工程学院,四川 成都 610054)

(gzxiong@uestc.edu.cn)

摘要:随着微电子技术和计算机技术的飞速发展,嵌入式产品广泛应用于消费电子、智能家电、通信设备等多个领域。介绍了嵌入式系统现状,分析了今后的发展趋势,阐述了传统方法的缺陷,介绍了一个新的设计方法学——SoC(片上系统)嵌入式系统软/硬件协同设计,并较详细分析了支撑该方法学的相关技术。

关键词:嵌入式系统;协同设计;重用;片上系统;IP核;软件构件;协同综合;测试调度

中图分类号: TP39 **文献标识码:** A

Survey on techniques of SoC hardware/software co-design

XIONG Guang-ze, ZHAN Jin-yu

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu Sichuan 610054)

Abstract: With the increasing development of the technology in microelectronics and computer science, embedded applications are widely used in many fields, such as consumer electronics, intelligent electronic household appliances, and communication equipments. The present status of embedded system was introduced, the trend in the future was analysed, the deficiency of the traditional design methods was presented. The hardware/software co-design methodology of SoC embedded systems was proposed, the supporting techniques were given.

Key words: embedded system; co-design; reuse; System on Chip (SoC); IP Cores; components; co-synthesis; test scheduling

1 嵌入式系统设计的趋势

嵌入式系统广泛应用于国民经济和国防各个领域,发展非常迅速。调查数据表明,嵌入式系统的增长为每年 18%,大约是整个信息技术产业平均增长的两倍;目前世界上大约有 2 亿台通用计算机,而嵌入式处理器大约 60 亿个。嵌入式系统产业是 21 世纪信息产业的重要增长点。

随着嵌入式系统用户需求和相关技术的迅速发展,嵌入式系统设计的发展趋势如下:

- (1) 用户需求不断增加,嵌入式系统设计的复杂性不断提高,设计规模不断增大,设计对象由单机走向分布式系统;
- (2) 嵌入式系统应用领域不断扩大,不同场合对系统设计的功能、功耗、实时性、面积等需求各不相同,嵌入式系统设计要求由单目标走向多目标;
- (3) 嵌入式技术大量应用于手机等用电池供电的移动设备中,系统设计的功耗和体积限制要求不断增强,嵌入式产品的集成度越来越高;
- (4) 半导体技术不断发展,硬件的集成度不断提高,系统级芯片 SoC(片上系统)的诞生使得将 CPU、存储器和 I/O 接口等 IP 核集成在单个硅片上成为可能,并且逐步成为当今嵌入式系统设计的主流;
- (5) 嵌入式产品更新速度加快,系统设计周期不断缩短,新产品问世时间不断减少,系统设计更加强调设计重用,软件系统更多地采用构件重用,硬件系统更多地采用 IP 核重用;
- (6) 不同于在通用计算机系统上开发软件,嵌入式系统

是一个软件和硬件并存的系统,设计时要从软件和硬件两个领域来综合考虑问题,它们互相联系、相互补充和互相制约。

2 传统设计方法的局限性

过去的嵌入式系统硬件部分设计容易,系统开发主要是在目标电路板上进行编程和交叉调试,所以那时的嵌入式系统开发很大精力都放在软件上,通常认为只要软件开发质量能够很好地保证,就可以很好地满足整个系统的功能和性能需求。传统的嵌入式系统设计流程如图 1 所示,系统软件和硬件划分完成以后,首先进行硬件子系统的设计与实现,再进行软件子系统的设计与实现,整个系统设计与实现过程基本上是一个串行过程。

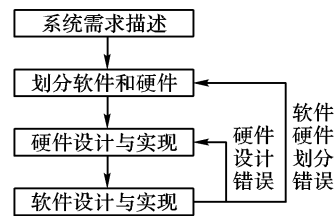


图 1 传统的嵌入式系统设计流程

然而随着嵌入式产品需求不断增加、硬件集成度不断提高,传统的嵌入式开发方法逐渐暴露出很多不足之处^[1]:

- (1) 软件和硬件的开发过程割裂,缺乏沟通。现在的嵌入式系统在设计早期就分开进行,由于异构型系统软硬件复杂,这种设计过程使软件和硬件不能得到协调优化,导致了设

收稿日期:2005-10-15 基金项目:国家 863 计划资助项目(2003AA1Z2210)

作者简介:熊光泽(1938-),男,四川丹棱人,教授,博士生导师,主要研究方向:嵌入式计算、实时操作系统、普适计算、系统仿真技术;詹瑾瑜(1978-),女,黑龙江绥化人,博士研究生,主要研究方向:VLSI 设计与测试、SoC 软/硬件协同综合、SoC 形式化协同验证、SoC 测试调度、实时嵌入式系统。

计效率低下;

(2)设计自动化层次低。系统级设计由设计人员手工完成,设计的好坏依赖于设计人员的经验,而随着系统规模不断提高,设计复杂度将往往超出人的思维范围,导致设计不当;

(3)设计过程串行化增加了设计周期。目前的设计流程主要采用先硬件后软件的开发模式,即实现了硬件的物理原型后才开始开发软件,因此串行设计不能充分及时地进行全系统综合考虑,导致设计失误增加,设计过程拖延;

(4)缺乏设计重用支持。目前的嵌入式系统设计几乎都是从零开始的,没有很好地利用过去开发的成果,导致产品问世周期增长,市场竞争力下降。

图2显示了嵌入式系统设计复杂性和设计生产率的增长趋势。由此可知,目前设计生产率的提高速度赶不上设计复杂度增加的速度,设计方法和工具成为制约嵌入式系统发展的瓶颈,嵌入式系统设计方法学期待革新。

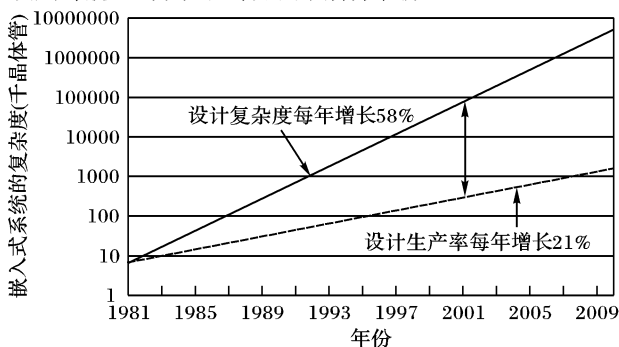


图2 设计复杂度和设计生产率的增长趋势

3 软/硬件协同设计方法

针对嵌入式系统设计面临的问题与挑战,研究者们开始探索新的设计方法学——软/硬件协同设计(Hardware/Software Co-Design)方法学。软/硬件协同设计方法学的研究始于20世纪90年代初期,第一届International Workshop on Hardware/Software Codesign会议于1993年召开,它标志着软/硬件协同设计方法学的研究正式展开,软/硬件协同设计领域正式确立。软/硬件协同设计不仅是一种设计技术,同时也是一种新的设计方法学,其核心问题是在设计过程中协调软件子系统和硬件子系统。

与传统的嵌入式系统设计方法不同,软/硬件协同设计强调软件和硬件设计开发的并行性和相互反馈,如图3所示,克服了传统方法中把软件和硬件分开设计所带来的种种弊端,协调软件和硬件之间的制约关系,达到系统高效工作的目的,软/硬件协同设计提高了设计抽象的层次,拓展了设计覆盖的范围。与此同时,软/硬件协同设计还强调利用现有资源,即重用构件和IP核,缩短系统开发周期,降低系统成本,提高系统性能,保证系统开发质量。

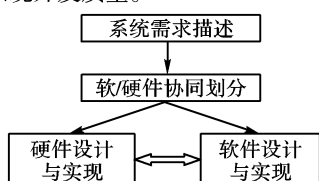


图3 嵌入式系统软/硬件协同设计流程

4 软/硬件协同设计方法学

目前嵌入式系统软/硬件协同设计方法学研究还处于发展阶段,许多技术仍未成熟和实用化,但是它将给嵌入式系统设计带来革命性的变化,极大地提高设计生产力,软/硬件协

同设计方法学及其相关技术的研究意义重大。

4.1 软/硬件协同综合技术

嵌入式系统是软件和硬件一体化的系统,系统中很多功能既可以由硬件来完成,也可以由软件来实现,硬件速度快,而软件成本低,这就需要权衡系统的时间、成本等性能指标之间的关系。设计嵌入式系统时,要分析不同的软件和硬件组合情况,决定系统各个模块由硬件完成或是软件完成,这是一个非常重要的工作,这个划分系统软件和硬件的过程被称为嵌入式系统软/硬件协同综合或者协同划分。目前嵌入式系统软/硬件协同综合相关技术还不成熟,面向SoC的商业化自动综合设计工具尚在孕育之中,而嵌入式系统结构日益复杂,开发时间要求日益紧迫,使得软/硬件协同综合问题成为嵌入式系统软/硬件协同设计方法学的关键问题之一^[2]。

现有的嵌入式系统软/硬件协同综合方法主要存在三方面缺陷:1)在传统的嵌入式设计方法中,划分软件和硬件的工作是由设计者手工完成的^[3],划分结果的好坏依赖于设计者的经验,而且效率低下,在系统设计开发过程中变得日益突出,所以需要一种自动化程度高的嵌入式系统软/硬件协同综合方法;2)设计一个嵌入式系统不仅要考虑系统功能的需求,还要考虑其性能需求和限制条件,然而很多嵌入式系统软/硬件协同综合方法^[4-7]只能评价一种或少数几种性能指标,因此需要一种能够同时评价多种性能指标(如价格、面积、功耗、时间性等)的嵌入式系统软/硬件协同综合方法;3)嵌入式系统硬件日益复杂,集成度越来越高,SoC已经成为设计嵌入式产品的主流技术,然而现有方法^[8-13]并没有考虑到SoC设计的一些特殊要求(如IP核重用问题),因此需要一种适应于SoC设计的软/硬件协同综合方法。

4.2 软/硬件协同验证技术

嵌入式产品竞争越来越激烈,只有缩短产品问世时间才能在竞争中取胜,因此必须在系统设计和开发过程中尽早发现系统中存在的各种问题和错误。由于嵌入式系统自身的特点,系统不仅对功能有特殊要求,同时对各种性能指标(如功耗、面积、成本、时间性等)也都有严格的限制,如果在系统设计开发完毕才对各个指标进行评价,那么一旦某些指标不满足要求势必会推迟系统问世的时间。因此需要一种新的技术在系统设计开发完成以前的各个不同阶段根据系统性能指标要求对设计方案综合评价,以验证系统实际的合理性和可行性。

目前嵌入式系统软/硬件协同验证的研究主要有两个方向,即仿真验证和形式化验证。

仿真验证方法是在硬件开发以前用硬件描述语言(如Verilog HDL^[14]、VHDL^[15]、SystemC^[16]和HandleC^[17]等)完成硬件子系统的描述,用软件来搭建硬件以达到系统软件和硬件并行开发的目的,能够完成系统的联合调试,灵活地纠正软件 and 硬件的设计错误,避免了人力和物力的浪费。但是硬件描述语言能够描述的硬件只是普通嵌入式系统硬件的一个子集,不适合描述大型的复杂系统,而且仿真系统与真实系统相比,功能和性能偏差也很大,因此仿真验证方法通常只能验证一些简单系统的逻辑需求,对于功能复杂或实时性要求较强的系统就无能为力了。

形式化验证方法是建立被验证系统的数学模型,然后用数学方法证明被验证系统的正确性以及各种性能指标是否满足要求。形式化验证方法比仿真方法更精确,所以目前更多的国内外专家都将研究重点转移到形式化验证方法上。卡内基梅隆大学计算机系的Edmund M. Clarke教授和他的学生从20世纪90年代就开始对硬件电路的形式化验证方法进行研究,提出了一种基于有限状态机的模式检验理论^[18],近年

来模式检验理论也开始应用于嵌入式系统的软/硬件验证。断言抽象本来是一种验证纯软件系统正确性的形式化验证方法,首先由 Graf 和 Saidi 在文献[19]中提出,近年来很多学者对其进行了改进,同样使其应用到了软件和硬件并存的系统验证中。Luis Alejandro Cortes 等人提出了 PRES (Petri net based Representation for Embedded System) 形式描述方法^[20-21],用 Petri 网对传统嵌入式系统软件和硬件进行描述,再用 Alur 等人在文献[22]中提出的基于混合自动机模型对此模型进行验证。清华大学蒋屹新等人提出了基于 Petri 网的模型检验研究^[23],但是此方法是一种仅针对软件系统的形式化验证方法,没有硬件方面的考虑,无法直接有效地应用于验证嵌入式产品。综合分析上述的方法,它们都只是针对传统的嵌入式系统或特定的软件系统的验证方法,没有考虑到基于 IP 核和构件重用的嵌入式系统软/硬件协同验证相关问题,这是今后研究的重点。

4.3 性能指标评价技术

嵌入式系统是以应用为中心、对系统功能和性能指标都有严格要求的专用计算机系统。标准评价各性能指标是保证嵌入式系统需求的必要条件。

4.3.1 成本

系统成本由开发成本、生产成本等多种因素组成,不考虑开发成本而仅考虑系统实现软件和硬件的成本是片面的、不现实的。成本参数有累加的特点,若将系统划分成几个子系统,则系统总成本等于各个子系统成本之和。系统成本公式描述如下:

$$C(\text{system}) = \sum_{i \in \text{system}} C_i$$

4.3.2 面积

随着嵌入式系统相关技术在消费电子、智能家电和通信设备等领域的广泛应用,嵌入式硬件的集成度越来越高,产品对硬件面积的要求也越来越严格,众多体积小、重量轻的嵌入式产品纷纷问世。系统硬件面积和各硬件组成部分直接相关,基本上可以通过各个硬件单元面积累加的方法获得。系统硬件面积公式描述如下:

$$S(\text{system}) = \sum_{i \in \text{hardware}(\text{system})} S_i$$

4.3.3 功耗

功耗已经被公认为是一个设计现代电子产品的重要指标,它是针对增加便携式产品电池的使用寿命提出来的,同时还要考虑到尽可能降低芯片封装和冷却装置的成本,提高系统可靠性以及环境因素等。由于电池材料技术满足不了系统日益增长的功耗需求,功耗问题已经成为系统设计过程中要重点考虑的性能指标之一。随着设计层次不断提高,为降低系统功耗所做的响应处理也被提到了更高的层次。

系统功耗由静态功耗和动态功耗两部分组成,其计算公式描述如下:

$$P(\text{system}) = P_{\text{static}} + P_{\text{dynamic}}$$

$$P_{\text{static}} = \sum_{i \in \text{system}} P_{\text{static}}(i)$$

$$P_{\text{dynamic}} = \sum_{i \in \text{system}, \text{Task} \in \text{Task}(\text{software}(\text{system}))} P(i, \text{Task})$$

4.3.4 时间性

大多数嵌入式系统都是实时系统,它们对系统的时间性有着或多或少的要求,时间性成为嵌入式系统设计的一个重要性能指标。时间性由硬件运行时间和软件运行时间组成,硬件运行时间是系统硬件运行所用的时间,可以通过将各个硬件单元运行时间进行累加得到;软件运行时间是系统中所有任务运行所用的时间,它与系统 CPU 的类型和任务数量有

关。系统运行时间的公式描述如下:

$$T(\text{system}) = T_{\text{hardware}} + T_{\text{software}}$$

$$T_{\text{hardware}} = \sum_{i \in \text{hardware}(\text{system})} T_{\text{hardware}}(i)$$

$$T_{\text{software}} = \sum_{i \in \text{hardware}(\text{system}), \text{Task} \in \text{Task}(\text{software}(\text{system}))} T(i, \text{Task})$$

4.4 SoC 测试调度问题

随着微电子技术的飞速发展,集成电路制造技术不断进步,芯片的特征尺寸越来越小,系统集成度越来越高,使得制造 SoC 成为可能^[24],为了适应现代电子产品快速增长的功能需求和日益紧迫的时间需求,设计者通常采用可重复使用的 IP 核来搭建 SoC,不仅能够简化设计过程,还有利于缩短产品制造周期和开发成本,因此 SoC 技术逐步成为当前嵌入式产品设计的主流技术。

对于一个成熟的嵌入式产品,人们总是追求成本最小化,嵌入式产品成本一般包括设计费用、制造费用和包装测试费用。SoC 设计方法可以缩短设计周期,降低设计费用。但仅仅降低设计费用是不够的,还需要尽可能降低测试费用,即要缩短产品的测试时间^[25-26]。随着设计复杂度的增加,SoC 上重用的 IP 数量越来越多,为了缩短芯片测试的时间,需要尽可能并行地测试芯片上的 IP 核。芯片上的 IP 核规模有大有小,各 IP 核所需要的测试数据端口也有多有少,而片上的封装引脚和测试总线数目是一定的,为此这样可以将几个较小的 IP 核输入输出端动态分配到同一测试总线上,允许它们同时进行测试,因此需要在 SoC 设计中进行测试调度^[27-28]。

SoC 测试包括三个子问题:测试访问机制 (Test Access Mechanism, TAM) 设计问题、核封装设计问题和测试调度问题^[29-30]。

在 SoC 中,IP 核嵌入到芯片中作为芯片的一部分,因此无法从芯片引脚直接访问到 IP 核的输入输出端口,必须为 IP 核提供相应的测试访问通道。测试访问机制是将测试激励从芯片的输入引脚传送到核输入端口上,再将核输出端口的测试响应传送到芯片的输出引脚上,常用的 TAM 结构有并行直接访问结构、串行访问结构、测试总线访问结构、可寻址测试端口访问结构等。

核封装提供一个 IP 核与 TAM 间的界面,它可以提供多种操作模式,如正常工作模式、核测试模式、互连测试模式和分流模式,另外核封装还能匹配核端口数量与 TAM 宽度。

测试调度是一个确定 SoC 中各 IP 核测试开始与结束的时间过程,它的原则就是尽可能缩减测试时间。目前已经有国内外专家从事这个问题的研究,并取得了一些初步成果。Chakrabarty^[31-33]证明了 SoC 测试调度问题的一个 NP-Complete 问题,并建立了测试调度问题的整数线性规划 (Integer Linear Programming, ILP) 模型,虽然基于 ILP 的测试调度方法取得了很好的实验效果,但是其计算复杂度高,不适合于测试大规模 SoC,目前一些专家从事采用启发式算来解决 SoC 测试调度问题的研究。

4.5 可测试性设计技术

随着嵌入式产品复杂性的增加,根据设计或研制完毕的系统来制定测试方案已经无法适应实际生产的需要,要求在系统设计时提早考虑测试的问题,即衡量一个系统设计的好坏不仅要看其功能和性能的优劣,还要考察其测试的方便程度。因此需要从可测试性度量、可测试性机制的设计与优化、测试信息的处理与故障诊断等方面进一步研究嵌入式系统的可测试性设计技术,提高设计效率,缩短设计周期。

4.5.1 可测试性度量

要提高产品的可测试性,首先要对产品的可测试性水平

进行描述,也就是进行可测试性度量。可测试性度量方法需满足精确性和简单性两个要求,所谓精确性是指可测试性度量方法能准确地预计产品测试程度生成的困难,并且定位到产品的某一部分,从而便于对产品进行更改;而简单性要求则是指度量可测试性的计算量应小于测试程度生成的计算量,否则,可测试性度量方法就会失去实际的应用意义。目前可测试性度量方法主要是针对数字电路系统^[34-36]。

4.5.2 可测试性机制的设计与优化

可测试性设计需首先将某种能方便测试的可测试性机制引入到产品中,再提供获取被测对象内部测试信息,因此合理、有效的设计可测试性机制是成功提高产品可测试性水平的基础,现有的可测试性机制设计方法有:LFSR方法、电平灵敏设计方法、IDDQ方法、边界扫描机制等^[37-38]。可测试性机制的引入虽然可以提高系统的可测试性指标、降低产品的全寿命周期费用,但是会在一定程度上增加产品的成本。因此,综合权衡可测试性机制的性能和费用,进行可测试性机制的优化设计是可测试性设计技术能否成功应用的另一个重要因素^[39]。

4.5.3 测试信息的处理与故障诊断

为了提高产品质量和可靠性、降低系统全寿命周期费用的目标,要求可测试性技术能够方便、快速地获取有关被测产品状态的信息,确定产品工作正常与否、性能是否良好、是否存在故障以及故障类型,以便采取设计调整、故障排除等后续工作。在对复杂的对象进行测试时,难点往往不在于如何获取测试信息,而在于如何对所获取的大量信息进行处理。例如,对于一个具有 N 个测试点的数字电路而言,所能获取的测试信息的总量为 $N * 2^N$ 位,随着 N 的增大,测试信息总量将呈指数型增长,因此能否对所获取的测试信息进行有效处理并对可能存在的故障进行精确诊断是可测试性设计技术成功应用的关键之一。

4.6 低功耗设计技术

随着嵌入式系统广泛应用于手机、PDA等便携式电子设备,功耗成为设计中需要重点考虑的性能指标。嵌入式系统所采用的CMOS功耗主要由动态功耗、静态功耗和短路功耗组成,要降低CMOS电路的动态功耗可以通过减小等效电容、降低电源电压、降低工作频率以及节点的翻转概率来实现;而要降低CMOS电路的短路功耗和静态功耗则需要调整阈值电压、电源电压等,低功耗设计技术实际上就是通过改变这些参数以达到降低系统功耗的目的,改变这些参数可以在不同的设计层次上进行。

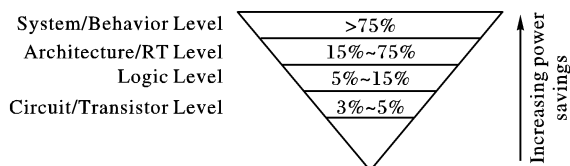


图4 不同抽象层次优化对功耗的改善程度

与嵌入式系统设计流程相对应,低功耗设计技术贯穿于系统设计、软件设计、逻辑设计、硬件设计等各个阶段,一种常用的设计层次分类方法^[40]是:系统级/行为级(System/Behavior)、结构级/寄存器传输级(Architecture/RTL)、逻辑级(Logic)和电路级/晶体管级(Circuit/Transistor)。研究表明^[41],抽象层次越高,优化的空间越大,采用低功耗技术可降低的功耗比例越大,效果越明显,如图4所示,因此嵌入式系统的低功耗设计技术需要从系统级入手进行研究。

5 结语

嵌入式系统软/硬件协同设计方法学是一个非常广泛的研究课题,主要包括:系统建模、软/硬件协同综合、设计功能和性能指标评价技术、软/硬件协同仿真、软/硬件协同验证、SoC测试调度技术等方面,并且还分为不同的设计层次。本文介绍了嵌入式系统现状,分析了今后的发展趋势,阐述了传统方法的缺陷,引出了一个新的设计方法学——嵌入式系统软/硬件协同设计方法学,并介绍了支撑新方法学的相关技术。

近年来,国内外对嵌入式系统软/硬件协同设计方法学的研究开始重视,但是仍处于发展状态,许多相关技术仍未成熟和实用化,没有成型的商业产品问世,这给我们带来了机遇和挑战。

参考文献:

- [1] ADAMS JK, THOMAS DE. The Design of Mixed Hardware/Software Systems [A]. 33rd Design Automation Conference [C]. 1996.
- [2] HENKEL J, ERNST R. An Approach to Automated Hardware/Software Partitioning Using a Flexible Granularity that is Driven by High-Level Estimation Techniques [J]. Very Large Scale Integration (VLSI) Systems, 2001, 9(2): 273 - 289.
- [3] GUPTA R, MICHELI GD. Hardware-Software Cosynthesis for Digital Systems [J]. IEEE Design & Test of computers, 1993, 26(4): 29 - 41.
- [4] ERNST R, HENKEL J, BENNER T, *et al.* The COSYMA Environment for Hardware/Software Cosynthesis of Small Embedded Systems [J]. Microprocessors and Microsystems, 1996, 20(3): 159 - 166.
- [5] HENKEL J, ERNST R. High-Level Estimation Techniques for Usage in Hardware/Software Co-Design [A]. Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference [C]. 1998. 353 - 360.
- [6] ELES P, PENG Z, KUCHCINSKI K, *et al.* System Level Hardware/Software Partitioning Based on Simulated Annealing and Tabu Search [J]. Design Automation Embedded System, 1996, 2(3): 5 - 32.
- [7] FILBO FC, MACIEL P, BARROS E. A Petri Net Based Approach for Hardware/Software Partitioning [J]. Integrated Circuits and Systems Design, 2001, 15(7): 72 - 77.
- [8] DANIEL R, PETER S, PAUL S. A Detailed Cost Model for Concurrent Use with Hardware/Software Co-Design [A]. Proceedings of the 39th Conference on Design Automation [C]. 2002. 269 - 274.
- [9] LEVMAN J, KHAN G, ALIREZAIE J. Hardware-Software Co-Synthesis of Partially Re-Configurable Embedded Systems Optimized for Reduced Power Consumption [A]. Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering [C]. 2003. 1835 - 1840.
- [10] PARAMESWARAN S. Code Placement in Hardware/Software Co-Synthesis to Improve Performance and Reduce Cost [A]. Proceedings of Design, Automation and Test in Europe [C]. 2001. 626 - 632.
- [11] DOBOLI A. Integrated Hardware/Software Co-Synthesis for Design of Embedded Systems under Power and Latency Constraints [A]. Proceedings of Design, Automation and Test in Europe [C]. 2001. 612 - 619.
- [12] LEVRNAN J, KHAN G, ALIREZAIE J, *et al.* Hardware-Software Co-Synthesis of Partially Re-Configurable Embedded Systems Optimized for Reduced Power Consumption [J]. Electrical and Computer Engineering, 2003, 5(3): 1835 - 1840.

从表 1 的对比分析可以发现: WinCE. NET 提供的功能比较多, 比较适合作为多媒体终端(PDA、智能手机、移动终端)的平台软件; RTLinux 功能单一, 一般适合学校学生学习嵌入式操作系统用; VxWorks 实时性较强, 网络组件较多, 比较适合开发通信产品; RTEMS 实时性最好, API 兼容性最好, 它广泛用于美国军方武器系统。

3 结语

本文首先对实时操作系统的实时性、可靠性以及标准兼容性等关键问题进行了分析, 并把这些用户所关心的问题与实时操作系统具体的实现技术进行了关联。通过这种关联, 我们可以通过对实时操作系统具体的实现技术以及它们的体系结构进行对比, 达到深入理解它们在实时性、可靠性等方面的优缺点的目的。最后对几种应用最广泛的实时操作系统的特性进行了详细的对比。通过这种对比来反映出它们在实时性、可靠性以及开发性等实时系统最为关注的问题上的支持情况, 这就为实际应用中的实时系统选型问题提供了方法上

的指导。

参考文献:

- [1] TOMIYAMA H, CHIKADA S, HONDA S. An RTOS based Approach to Design and Validation of Embedded System[A]. IEEE International Symposium on VLSI Design, Automation and Test[C]. 2005. 185 - 187.
 - [2] HAOBO Y, GERSTLAUER A, GAJSKI D. RTOS Scheduling in Transaction Level Models[A]. IEEE International Conference on Hardware/Software Codesign and System Synthesis[C]. 2003. 31 - 36.
 - [3] BOKE C, GOTZ M, HEIMFARTH T. Configurable Real-time Operating Systems and Their Application[A]. Proceedings of the Eighth International Workshop on Object Oriented Real-time Dependable Systems[C]. 2003. 148 - 155.
 - [4] YANBING L, POTKONJAK M, WOLF W. Real - time Operating System for Embedded Computing[A]. IEEE International conference on Computer Design[C]. 1997. 388 - 392.
-
- (上接第 760 页)
- [13] VESTIAS MP, NETO HC. System-Level Co-Synthesis of Dataflow Dominated Applications on Reconfigurable Hardware/Software Architectures[A]. Proceedings of the 13th IEEE International Workshop on Rapid System Prototyping[C]. 2002. 130 - 137.
 - [14] Verilog HDL[EB/OL]. <http://www.verilog.com/>, 2005.
 - [15] VHDL[EB/OL]. <http://www.vhdl-online.de/>, 2005.
 - [16] SystemC[EB/OL]. <http://www.systemc.org/>, 2005.
 - [17] HandleC[EB/OL]. <http://www.celoxica.com/>, 2005.
 - [18] CLARKE EM, GRUMBERG O, PELED D. Model Checking[M]. MIT Press, 1999.
 - [19] GRAF S, SAIDI H. Construction of Abstraction State Graphs with PVS[A]. Proceedings of the CAV'97[C]. 1997. 72 - 83.
 - [20] CORTES LA, ELES P, PENG Z. Formal Coverification of Embedded Systems Using Model Checking[A]. Proceedings of the 26th Euromicro Conference[C]. 2000. 106 - 113.
 - [21] CORTES LA, ELES P, PENG Z. Verification of Embedded Systems Using Petri Net Based Representation[A]. Proceedings of the 13th International Symposium on System Synthesis[C]. 2000. 149 - 155.
 - [22] ALUR R, HENZINGER TA, HO PH. Automatic Symbolic Verification of Embedded Systems[J]. IEEE Transactions on Software Engineering, 1996, 22(3): 181 - 201.
 - [23] JIANG YX, LIN C, QU Y, *et al.* Research on Model Checking Based on Petri Nets[J]. Journal of Software, 2004, 15(9): 1265 - 1276(In Chinese with English abstract).
 - [24] HASSE J. Design Methodology for IP Providers [A]. Proceedings 1999 Design, Automation and Test in Europe(DATE 1999) [C]. Washington DC: IEEE Computer Society, 1999. 728 - 732.
 - [25] ZORIAN Y, MARINISSEN EJ, DEY S. Testing Embedded-Core Based System Chips[J]. Computer, 1999, 32(6): 130 - 143.
 - [26] CHENG KT, DEY S, RODGERS M, *et al.* Test Challenges for Deep Sub-Micron Technologies[A]. Proceedings of 37th Design Automation Conference[C]. ACM Press, 2000. 142 - 149.
 - [27] CHAKRABARTY K. Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2000, 19(10): 1163 - 1174.
 - [28] LARSSON E, PENG Z. Test Scheduling and Scan-Chain Division Under Power Constraint[A]. Proceedings of 10th Asian Test Symposium[C]. Washington DC: IEEE Computer Society, 2001. 259 - 264.
 - [29] GHOSH I, JHA NK, DEY S. A Low Overhead Design for Testability and Test Generation Technique for Core-Based Systems[A]. Proceedings of IEEE International Test Conference 1997[C]. Washington DC: IEEE Computer Society, 1997. 50 - 59.
 - [30] KORANNE S. Formulating SoC Test Scheduling as a Network Transportation Problem[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(12): 1517 - 1525.
 - [31] CHAKRABARTY K. Design of System on Chip Test Access Architecture using Linear Programming[A]. Proceedings of 18th IEEE VLSI Symposium[C]. Washington DC: IEEE Computer Society, 2000. 127 - 134.
 - [32] CHAKRABARTY K. Optimal Test Access Architecture for System-on-a-Chip[J]. ACM Transactions on Design Automation of Electronic System, 2001, 6(1): 26 - 49.
 - [33] IYENGAR V, CHAKRABARTY K, MARINISSEN EJ. Test Wrapper and Test Access Architecture Co-Optimization for System-on-Chip[J]. Journal of Electronic Testing: Theory and Applications, 2002, 18(4): 213 - 230.
 - [34] 向东. 数字系统测试及可测试性设计[M]. 北京: 科学出版社, 1997.
 - [35] 陈光禹. 可测试性设计技术[M]. 北京: 电子工业出版社, 1997.
 - [36] 丁瑾. 可靠性与可测试性分析设计[M]. 北京: 北京邮电大学出版社, 1996.
 - [37] WAYERER M, GOLDEMAND G. Testability of Electronics Circuits [M]. Prentice Hall, 1992.
 - [38] Test Technology Overview[EB/OL]. <Http://www.cedcc.psu.edu/ee497f/rassp-43/>, 1998.
 - [39] EVANS JS, MCHUGH P, SEDMAK RM, *et al.* Integration of DFT into RASSP[A]. Proceedings of the 2nd RASSP Conference[C]. 1995. 217 - 222.
 - [40] ONG PW, YAN RH. Power-Conscious Software Design-A Framework for Modeling Software on Hardware[A]. Proceedings of the International Symposium on Low Power Electronics and Design[C]. 1994. 36 - 37.
 - [41] Synopsys Inc. Managing Power in Ultra Deep Submicron ASIC/IC Design[Z]. Synopsys White Paper, 2002.