

系统集成芯片设计的若干关键技术

魏少军

清华大学微电子学研究所, 北京 100084

E-mail: wsj@public3.bta.net.cn

收稿日期: 2008-01-07; 接受日期: 2008-03-26

国家自然科学基金资助项目(批准号: 60676012)

摘要 系统集成芯片(system on chip, SoC)时代最大的特点是芯片高度复杂, 架构设计和软件成为芯片设计不可分割的部分; 随着半导体工艺技术进步到超深亚微米, 功耗已经成为 SoC 设计中不可避免的挑战. 便携式系统要求 SoC 不仅要省电, 而且应该能够感知电池的特点并按照某种最优的方式工作, 以最大限度地延长便携式系统电池的工作时间. 尝试从上述几个角度阐述 SoC 设计的若干关键技术.

关键词

SoC

能耗感知设计
软硬件协同设计
芯片供应商软件

今天的半导体工艺已经可以将数亿只晶体管集成在一颗芯片上. 单个芯片上晶体管数量的激增导致系统复杂度的大幅度上升, 传统的芯片概念已经难以描述所要实现集成电路芯片的特点. SoC 的概念应运而生. 尽管到目前为止, 人们并没有给出 SoC 的确切定义, 但也达成了如下一些共识: SoC 是高度复杂的超大规模集成电路, 使用至少一颗微处理器(MPU), 允许使用者从外部对 SoC 进行控制/编程, 采用超深亚微米工艺制造和主要采用第三方知识产权(IP)内核进行设计.

从上述人们达成的共识可以看出, SoC 与我们所熟知的专用集成电路(ASIC)有着很大的不同. 这种不同首先表现在 SoC 的着眼点是系统(S), 芯片(C)是实现系统的手段. 而 ASIC 的着眼点是芯片; 其次, SoC 的概念不仅仅包含硬件, 也包括软件, 软件在 SoC 中是不可或缺的重要组成部分, 而 ASIC 仅仅涉及硬件或少量外部控制代码; 第三, SoC 设计中大量采用第三方成熟的模块以缩短 SoC 的设计周期、降低风险, 而 ASIC 则强调全部自主设计; 最后, SoC 是一个高度复杂的系统, 往往采用最先进的半导体工艺技术制造, 而 ASIC 并不强调使用某种特定的制造工艺, 也不突出其系统的复杂性, 重点放在其专用性上, 因此, SoC 设计有着特定的要求, 不能简单地照搬 ASIC 设计技术. 可以预见的是, SoC 设计技术是在 ASIC 设计基础上进一步发展起来的新一代集成电路设计方法, 既与传统的集成电路设计方法一脉相承, 又有着 SoC 时代鲜明的特征.

SoC设计必须面对多重挑战，如高度复杂系统的架构设计和验证，数量巨大的晶体管被集成进单个芯片并高速运行时带来的功耗急剧上升，以及看似和芯片设计无关，但实际上密不可分的软件技术等。架构设计是SoC设计中必须引起重视的一个问题，不同的应用需要不同的系统架构。尽管由于软件的作用，SoC的功能可以在一个比较大的范围内调整，但是在成本和效率的平衡中，架构的选择仍然不可回避。由于系统的高度复杂性，确保一个包含数千万，甚至数亿只晶体管的芯片设计满足设计规格的要求是SoC设计过程面临的一大难题。验证已经成为SoC设计中费时最多的工作；Moore定律(Moore's Law)在过去 40 多年中精确地描述着集成电路技术的发展。尽管人们普遍认为在未来 15 年中它将仍然有效，但功耗问题给Moore定律的持续发展带来了巨大的挑战。不仅仅是等比例缩小(*scaling down*)导致晶体管栅极氧化层厚度过薄而引发漏电流增大，静态功耗成为关注的焦点，而且数量巨大的晶体管集成在单个芯片上并高速运行也必然导致动态功耗上升。两者迫使SoC必须克服前所未有的功耗难题。事实上，CPU的主频在达到 4GHz之后，转向双核、或者多核的很大原因是功耗问题无法解决。选择多核结构的中心思想是希望利用功耗与电压的二次方关系，通过多核来确保不再增加功耗条件下的性能提升。功耗问题是那样的严重，甚至有人预测，它将有可能是事实终结Moore定律的少数关键因素之一。软件是一个摆在芯片设计师面前的崭新课题。与ASIC基本上不需要软件不同，SoC产品的内涵中就包含了芯片和软件两部分。与Windows和Oracle等基础软件不同的是，SoC涉及的软件与芯片密切相关，在系统体系架构上形成了一个新的层次。

本文将不去探讨集成电路设计的经典问题，如设计综合、可测性设计、版图设计等技术，而将主要探索 SoC 设计中特有的一些挑战，以抛砖引玉。

1 系统集成芯片的架构设计

架构设计是SoC设计中特有的工作，这不仅是因为系统的高度复杂性要求设计者在SoC设计初期就要高度关注系统架构的合理性，更要考虑SoC开发的高额成本，希望通过架构设计和软件配置，在需要的时候兼顾其他应用。令人遗憾的是，架构设计并没有引起SoC设计师的足够重视。很多SoC设计更多地是CPU+DSP和各种IP核的堆砌，缺少从应用中获取的系统架构信息的支持。图 1¹⁾和 2^[1]是两个SoC的架构图，显然它们十分相似。

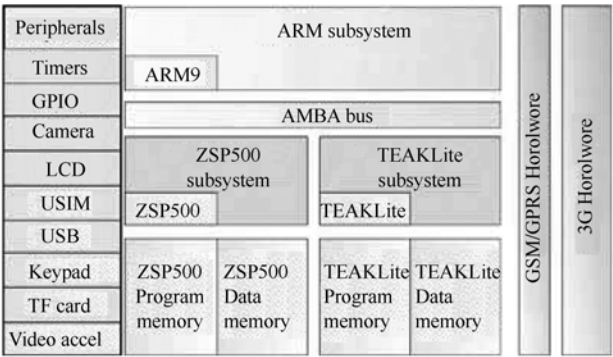


图 1 双模 3G 终端基带芯片系统框图

1) 王锦山. www.mbcom.cn, 2007-12-5

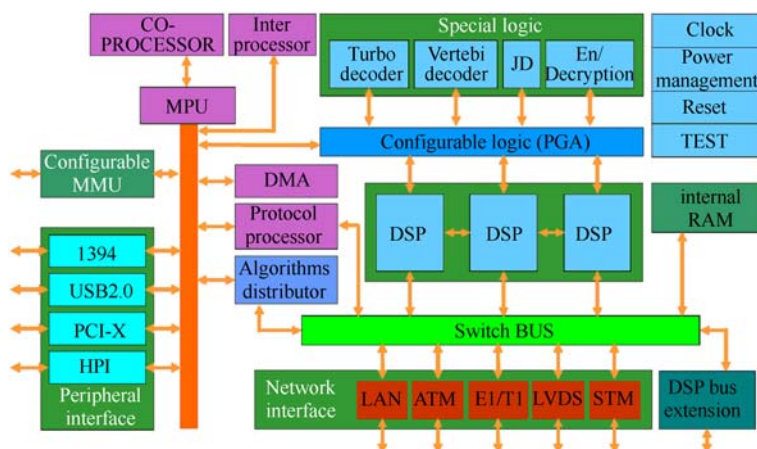


图 2 COMIP™ 系统框图

SoC 架构设计应该从系统任务图 TG(T, E)入手, 通过对系统任务图的分析, 确定所需的系统硬件和系统软件, 并将所需的硬件模块连接起来, 通过软件的运行, 有机地调动各部分资源, 协同完成所需的系统任务. 图 3 给出了一个系统任务图. 不难得出, 要完成这个系统任务图确定的任务, 需要未来的 SoC 具备处理各项任务 Task0-Task7 的硬件系统和一个用于管理系统运行的 CPU.

通过对任务图中各个任务的调度, 确定它们各自运行的时间和次序, 再根据所需的硬件资源将各个任务与这些硬件资源绑定, 在 CPU 的统一控制下协调运行. 这一设计方法借用了早期的“行为综合(behaviors synthesis)”理论, 根据任务运行的需要来规划硬件资源, 具有资源数量和种类尽可能节约, 资源的使用科学合理的特点.

SoC 设计的第二个特点是必须考虑软件. 长期以来, 从事集成电路设计的技术人员大多忽略软件的作用, 教学和科研中人为的软、硬件条块划分也导致芯片和软件两部分人才在知识上的割裂. 在 SoC 时代, 软件设计已经成为 SoC 设计中的一个难点. 图 4 给出了一个 SoC 架构设计的层次描述. 显然, 集成电路技术人员所熟悉的是软硬件的界限, 而对 SoC 设计中必须关注的应用和架构的分离则没有足够的理解.

在传统的集成电路芯片设计中, 我们往往只关注硬件, 如包含哪些基本部件, 哪些是输入输出, 电路的基本功能是什么, 等等. 根本的思路是用电路(硬件)实现所需的功能. 但是, SoC 的着眼点则不仅仅使用硬件, 还要用软件实现所需的系统功能. 因此, 设计 SoC 的出发点就是

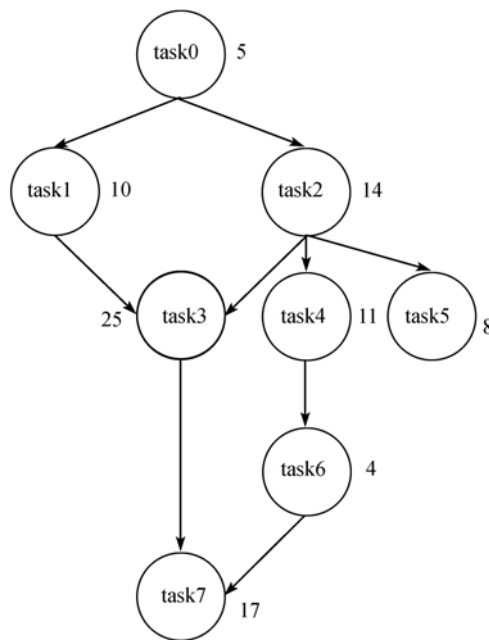


图 3 系统任务图

要从系统的需求出发, 从架构上考虑如何用软件和硬件实现这些功能, 这要求 SoC 设计师必须了解软件、懂软件。

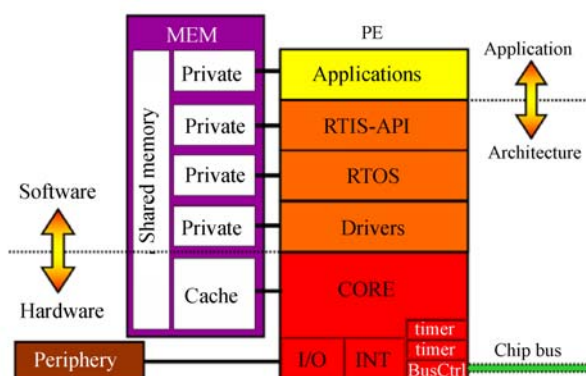


图 4 SoC 的架构设计分层

架构设计首先要实现的是软硬件划分, 这也是架构设计中的难点。软硬件划分需要确定划分的准则、需要确定哪些任务由硬件来完成、哪些任务由软件来实现。到目前为止, 人们还没有找出实现软硬件划分的通行准则, 更多的是针对具体要求一事一议。基于系统任务图的划分方法虽然还不能彻底解决问题, 但是可以比较清晰地将一个复杂问题分解成一系列子任务。针对子任务的运算复杂性分析, 则可以比较客观地给出该子任务是用硬件, 还是用软件实现更合理。

SoC 的软件架构设计与计算机软件设计有着很大的不同。首先, SoC 的软件架构设计是基于系统任务图来实现的, 具有明显的任务相关性; 其次, SoC 软件更多地起到联系硬件资源和上层实时操作系统(RTOS)的作用, 有着清晰的硬件烙印; 第三, SoC 软件对 SoC 的性能有着密切的影响, 优化是重点; 最后, SoC 软件的维护升级需要长期进行, 技术支持和服务不可或缺。

2 系统集成芯片的功耗设计

长期以来集成电路工艺技术按照等比例缩小的方式不断进步, 等比例缩小事实上是对影响晶体管特性的一组大约 20 多个参数(图 5)按照系数 α 缩小或扩大, 理想情形下, 系数 α 在推进等比例缩小的同时, 将保持功率谱密度为常数, 这意味着等比例缩小将不会出现功率急剧增大的现象。但是由于 MOS 晶体管的栅氧厚度无法线性地任意减薄和较高的电源电压总能给出较好的电路特性, 所以, 栅氧厚度和电源电压的实际值与理论值产生了很大的偏差^[3](图 6)。

其实, 在 CMOS 晶体管特征尺寸到达公认的极限(1.4 nm)之前, 栅氧厚度就已经达到了极限。设想一个只有 5~6 个原子厚度的栅氧层, 1% 的缺陷将产生 10~100 倍的漏电流。静态漏流已经成为人们不可忽略的灾难性问题^[2]。其实, 这已经不是人们第一次遇到这样的情形了。早在 20 世纪 80 年代, 当主流集成电路工艺从双极型转向 CMOS 的时候就出现过一次。虽然从双极跳到 CMOS 产生了一次崖跳^[3](“功耗悬崖”, 图 7), 但是平面工艺的进步使得 CMOS 的面积劣势(使用两只晶体管)没能阻碍芯片集成度的持续提升。今天我们再次遇到了“功耗悬崖”, 但令人感到困惑的是, 我们至今还看不清“悬崖”下的落脚点在哪里。

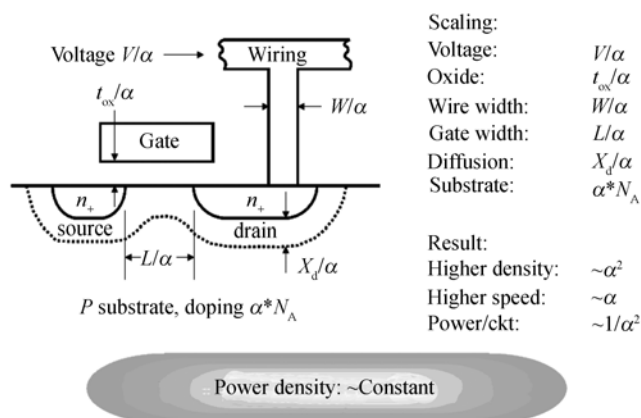
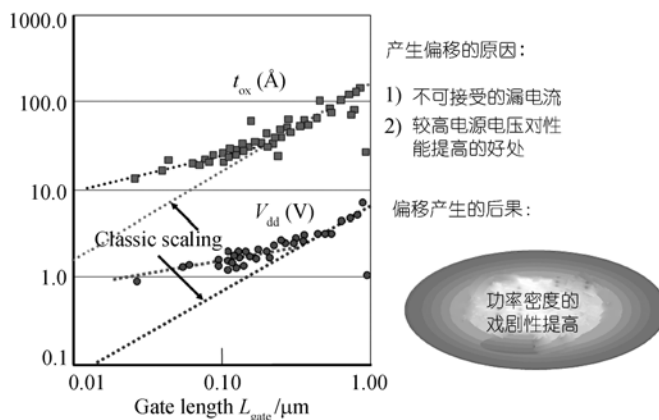


图 5 等比例缩小的相关参数

图 6 实际工作产生的偏差^[3]

代表着集成电路最高水平的 CPU 最先遇到这个瓶颈。随着特征尺寸的不断缩小, CPU 的运算速度也在成倍提高。早在几年前, 主频 3GHz 或更高的 CPU 就已经问世了, 但是这两年人们再没见到过去 10 多年 CPU 主频快速提升的惊人场景。究其根本原因, 是功耗在作祟。简单地提高主频在带来性能提升的同时, 也带来诸多的问题。由功耗提升带来的散热问题是如此之严重, 以至于人们不敢再奢求提高 CPU 主频了。面对不断提升 CPU 性能的呼声, 人们只能转而求助于双核 CPU、甚至是多核 CPU 架构, 利用功耗与电压的二次方关系来化解 CPU 主频不能继续升高带来的性能提升瓶颈。显然, 双核和多核 CPU 的出现并没有满足人们对 CPU 性能线性提升的要求, 双核 CPU 的性能并不是两颗同类 CPU 的性能叠加。多核 CPU 事实上是以牺牲成本来换取性能的有限提高。功耗的问题是如此之严重, 以至于有人认为, 它可能是事实终止 Moore 定律的一个最重要的因素。

尽管人们在集成电路的设计中采用了各种技术以降低功耗, 例如提升晶体管的阈值电压以减少漏电流, 降低静态功耗; 在电路设计过程中有意识地甄别那些不需工作的部分, 通过门控时钟(clock gating)迫使其暂停工作等, 但还不能有效地解决问题, 功耗的问题仍然在困扰着

我们. SoC 设计由于要集成 CPU 和 DSP 等高速单元, 且集成众多的部件, 故功耗问题不可避免. 而 SoC 的主要应用领域往往是移动通信终端等便携式设备, 对低功耗的要求是显而易见的. SoC 的低功耗设计技术必然成为人们关注的焦点.

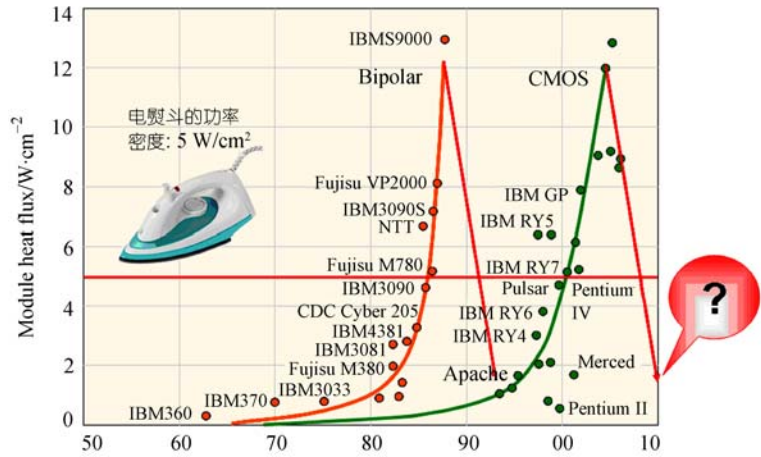


图 7 “功耗悬崖”

SoC 的低功耗技术大致可以分为如下几类: 一是充分利用功耗和电压的二次方关系, 通过最大限度地将运算并行化来降低功耗; 二是以 IP 核为单位实现整体的电源或时钟控制, 让那些不需工作的电路全部关断, 以节省功耗; 三是使用多电压技术, 仅让那些需要高速工作的部件运行在较高的电源电压下, 其他的则运行在较低电源下, 以节约能耗; 四是发展能耗感知技术, 让 SoC 能够根据供电电池的特性来规划自身的工作节奏, 达到减低能耗、延长电池使用时间的目的.

多电压供电系统的设计技术已经被普遍采用^[4,5], 例如在很多集成电路芯片设计中内核往往采用较低的工作电压, 而输入输出则采用较高的电压, 以适配外部系统. 但是根据不同的任务, 对同一个运算资源采取不同的供电电压, 以达到降低能耗的目的还是近几年来兴起的一个研究热点. 图 8 给出了一个按照时间轴分布的任务图调度的例子^[6]. 显然, 对于那些不在关键路径上的任务, 完全可以通过降低其供电电压换取能耗的降低. 这给出了一个新概念, 那就是电源电压的调度^[7,8]. 按照这一作法, 可以在确保系统性能的前提下, 尽可能地降低系统能耗. 图 9 给出了另外一个例子^[9], 该例子通过对电池特性的分析并根据每个任务所需的电流大小进行调度, 电流大的先执行, 并据此进行任务的电压调度, 实现电池寿命的最大化.

功耗对于 SoC 来说是不可避免的问题. 要解决这个问题, 靠底层的优化设计收效不会很大, 应该转换思路, 在系统层面来研究, 以获得较大的改善. 这也从另外一个侧面证明了在 SoC 设计中系统知识的重要性.

3 系统集成芯片的软件设计

SoC 的软件不仅仅包含称为芯片提供商软件(IC vendor software)和用于 SoC 本身运行的软件, 还包括提供给用户和帮助用户在 SoC 上开发应用的软件, 我们称之为开发器软件. 前者在

SoC 产品价值中的比例将不断提升, 并有可能在集成电路技术走向 32nm 的过程中, 成为 SoC 产品的主要赢利点(图 10 芯片提供商软件^[10]). 后者是 ASIC 设计人员以前没有接触过的. 应该说, 开发后者的难度并不比设计一颗 SoC 的难度小, 有的时候甚至难度更大. 此前, 只有少数通用器件的开发商, 如 CPU, DSP 和 FPGA 的厂商有能力开发这类软件, 而大部分芯片设计工程师缺少这方面的积累, 这必须引起我们的重视.

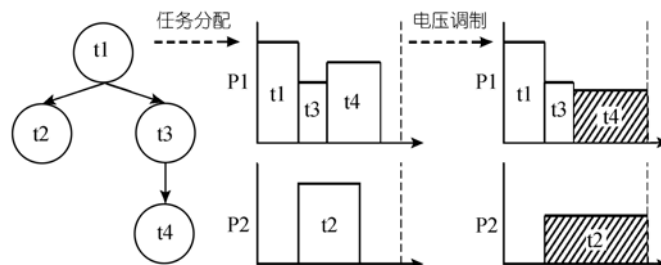


图 8 多电压调度技术

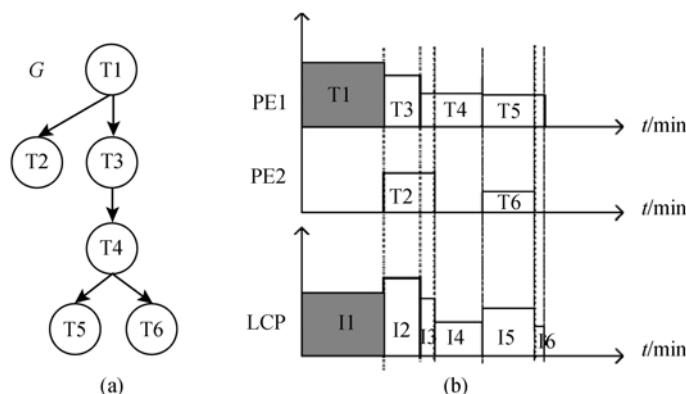


图 9 电池感知的变电压多处理器调度

软件在集成电路开发过程中的作用已经越来越重要. 在大部分无制造半导体(Fabless)公司中, 软件工程师的数量在不知不觉中, 渐渐地超越芯片设计工程师的数量. 在部分先进的 Fabless 公司中, 软件工程师的数量和芯片设计工程师的比例达到 3:1, 这一比例在未来有可能会达到 5:1 甚至 8:1. 我们没有理由再对软件视而不见了. 今天, 我们已经可以毫不犹豫地说, 今后的竞争将是软件的竞争, 谁拥有卓越的软件开发能力, 谁就将拥有 SoC 的天下.

4 结论

本文初步探讨了架构设计、功耗设计和软件设计这 3 个 SoC 设计特有的技术问题, 从讨论中可以清晰地看到 SoC 设计与传统的 ASIC 设计有着很大的不同. ASIC 设计尝试以电路(硬件)实现所需的系统功能, 而 SoC 尝试以硬件和软件的协同来实现系统功能. SoC 的灵活性和处理复杂问题的能力显然得到了很大的提升. 架构设计是 SoC 中最重要的内容, 简单地复制常规的 CPU+DSP 方案并不一定能够带来最优的性能, 而基于系统任务图的方式, 通过对任务

的调度和资源分配,有可能是设计 SoC 架构的最佳途径,同时也可以有效地解决软硬件划分

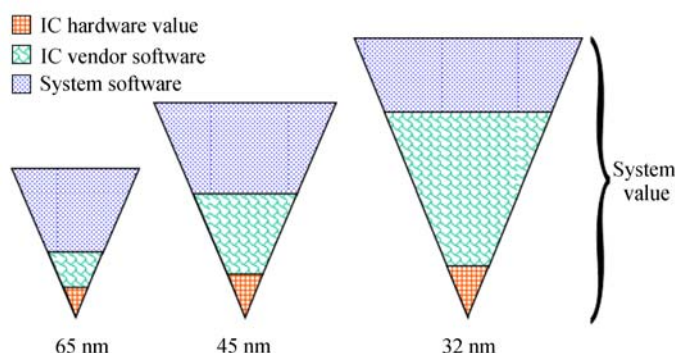


图 10 芯片提供商软件在 SoC 中的价值分布变化

等长期困扰我们的难题,软件架构的问题也可以得到很大程度的解决.功耗问题已经开始困扰集成电路设计工程师们.以便携式应用为主要对象的 SoC 面临的功耗挑战是严峻的.系统级功耗设计、多电压系统设计和能耗感知的多处理器设计技术有可能在严峻的功耗难题下取得重要突破.软件是 SoC 开发中必须面对的难题,其难度也许比设计一个 SoC 更难,但是这种难度并不体现在技术上,而是 SoC 设计师的知识缺陷造成的.随着芯片供应商软件在 SoC 成本中的比例越来越高,我们不可避免地要涉足这一领域,这就要求我们的知识要更新,要更加丰富起来.软件将是 SoC 竞争的制高点,这个今天看似“滑稽”的论断也许明天就会变为现实.

参考文献

- 1 Wei S J. Developing service-oriented communications SoC platform. In: Mobile Communications Specific ASIC Workshop. Shanghai, 2004
- 2 Yan L. Joint dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems. *IEEE Trans Integr Circ Comput Aid Des*, 2005, 24(7): 1030—1041 [\[DOI\]](#)
- 3 Meyerson B S. The shift from optimizing cost to maximizing opportunity. Synopsys Executive Forum, May 2005
- 4 Yao F, Demers A, Shenker S. A scheduling model for reduced CPU energy. *Proceedings of 36th Annual Symposium on Foundations of Computer Science*. Los Alamitos: IEEE Computer Society Press, 1995: 374—382
- 5 Zhu D, Melhem R, Childers B. R. Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems. *IEEE Trans Parall Distr Syst*, 2003, 14(7): 686—700 [\[DOI\]](#)
- 6 Wang Z D. Research on the methods for energy aware embedded system design. PhD Dissertation. Beijing: Tsinghua University, 2007
- 7 Cai Y, Reddy S M, Pomeranz I, Al-Hashimi B M. Battery-aware Dynamic Voltage Scaling in Multiprocessor Embedded System. In: *IEEE International Symposium on Circuits and Systems*. New York: IEEE Press, 2005. 616—619
- 8 Rao R, Vrudhula S, Chang N. Battery optimization vs energy optimization: which to choose and when? In: *Proc ICCAD*. Washington: IEEE Computer Society, 2005. 438—444
- 9 Xie Y F. Research on system-level energy-aware design methods for portable applications. PhD Dissertation. Beijing: Tsinghua University, 2007
- 10 Industry Restructuring. IBS Report, 2007, 16(5): 14