

## 用于低功耗编译的 SPM 部件功耗模型研究

蒋湘涛 胡志刚 贺建彪

(中南大学信息科学与工程学院 长沙 410083)

**摘要:** 为了获得 SPM(Scratch-Pad Memory)部件最佳的使用效果, 需要合适的 SPM 性能和功耗模型来指导编译优化过程。现有的功耗模型只提供 SPM 部件的平均访问功耗, 没有反映电路实际功耗随电路不同输入而改变的特征, 限制了更进一步的优化。该文提出依照电路结构生成 SPM 部件的基本功耗模型, 并使用程序运行时信息生成模型中的参数因子, 用来反映不同应用程序运行时电路的实际活跃度。实验结果表明, 该功耗模型测量的能耗值在总体上与现有基于统计方法生成的功耗模型结果相一致, 同时能反映不同应用程序访问 SPM 部件时的功耗差异, 对编译器优化 SPM 部件的访问方式具有重要的指导意义。

**关键词:** SPM; 低功耗; 编译优化; 功耗模型

**中图分类号:** TP314; TN402

**文献标识码:** A

**文章编号:** 1009-5896(2009)04-0963-05

## Research on SPM's Energy Model for Low Power Compilation

Jiang Xiang-tao Hu Zhi-gang He Jian-biao

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**Abstract:** In order to achieve a better application effect of Scratch-Pad Memory(SPM), it's appropriate to construct a proper SPM's performance and power-consumption model to guide the compiling optimization process. The existing power-consumption modeling can only provide SPM's average access power-consumption which didn't reflect the characteristic that the actual circuit's power-consumption varied as the input varied. It restricts further optimization. This thesis proposes to construct SPM's basic power-consumption modeling according to the circuit structure and generate modeling's parameters based on program's runtime information. These can reflect the circuit's actual active degree when different programs executed. Shown in the experiment, the power-consumption measured is basically as accurate as the one that based on the existing statistical method on average. Furthermore, the new modeling can reflect the difference of power-consumption when different programs access SPM. This technique has an important guiding significance to the access mode as the compilers optimize SPM.

**Key words:** Scratch-Pad Memory(SPM); Low power; Compile optimization; Power model

### 1 引言

通过对嵌入式系统的能耗分析, 可以发现 cache 部件是嵌入式处理器中消耗能量的主要部件, 几乎占到整个片上处理器能耗的一半, 其中单指令 cache 子部件就占到整个处理器 27% 的能耗<sup>[1]</sup>。因此, 部分嵌入式处理器选择使用 SPM 部件来代替片中的 cache 部件, 在保障性能的前提下, 争取降低嵌入式处理器的能耗。尽管 SPM 部件和 cache 部件均主要由 SRAM 单元构成, 但两者在电路结构和操作方式上存在很大不同, SPM 部件拥有独立的地址空间, 嵌入式处理器能直接显示地访问 SPM 部件, 不需要地址映射电路, 使得每次访问 SPM 部件的能耗值远低于 cache 部件; 同时, 如果处理器所访问内容不在 SPM 部件中时则不产生能耗, 而 cache 部件无论是否命中均产生能耗。由此可见, SPM 部

件在降低系统功耗方面作用显著, 但它使用上比 cache 部件复杂, 由于它使用独立的地址空间, 其内部单元内容与程序元素的映射必须由编译器来完成, 如果没有相应基于 SPM 部件的低功耗编译器支持, 将无法充分发挥它在保障性能与降低功耗方面的作用。

基于 SPM 部件的低功耗编译优化途径主要有两个: 其一是静态或动态地管理程序运行时 SPM 部件中内容, 减少程序运行时处理器需要访问主存的次数; 其二是在编译过程中选择产生更低功耗的 SPM 部件访问序列。前者的关键问题是构造算法决定程序文件中哪些数据或指令置入 SPM 部件中, 以求最大化地重复访问 SPM 部件中的内容, 后者的关键问题是构造能反映 SPM 部件访问能耗随实际访问情况而变化的功耗模型。本文的研究工作主要针对第二个优化途径, 构造能用于优化 SPM 部件访问模式的功耗模型。

目前, 集成电路的电路级和门级功耗分析已经相对成熟, 结果相对精确, 但是仅能用于集成电路以及嵌入式系统的高层设计<sup>[2]</sup>。而可用于低功耗编译指导的体系结构级 SPM

2008-01-17 收到, 2008-06-26 改回

国家自然科学基金(60673165)和湖南省自然科学基金(07jj5077)资助课题

功耗模型则不够完善,文献[3]比较早地研究了 SPM 部件的功耗模型,基于 cache 部件的功耗模型,去除 tag RAM 电路等所产生的功耗,粗略地构造了 SPM 部件的功耗模型,由于没有完全体现 SPM 部件的电路特点,整体而言其结果误差较大。文献[4]也是从电路结构出发来构造 SPM 部件的功耗模型,同时还考虑了电路中的静态功耗,因而其功耗模型相对精确,是低功耗编译过程中使用最广泛的功耗模型;文献[5,6]提出了 SRAM 参数化的功耗模型,其特点是考虑了不同工艺条件下静态功耗差异。此外还有其它把 SPM 部件看作黑盒,通过统计方法来构造的功耗模型<sup>[7]</sup>。在就统计大量实验的平均值的而言,现有这些功耗模型的仿真值与物理测量值误差较小<sup>[4]</sup>,但就单次测量结果而言误差比较大,它们存在一致的问题是不考虑访问 SPM 部件时电路中实际输入值情况,不能反映电路实际功耗与电路电容变化息息相关的特点<sup>[8]</sup>,每次输入/输出操作能耗为定值,和程序运行时具体的 SPM 部件访问方式无关,仅与 SPM 部件的尺寸、结构以及制造工艺相关,从而仿真值与实际值可能存在比较大的误差。而将其应用于低功耗编译优化过程中时,限制了通过减少电路运行时的位翻转以及确定更优的 SPM 部件内容布局所带来的优化空间。为此本文提出基于 SPM 各子部件的内部电路特征以及工作方式来分别构造其体系结构级功耗模型,同时为了反映电路功耗在实际运行过程中由于不同电路输入而变化的特征,使用程序运行时信息构造功耗模型中相应的电路活跃度因子,由此来刻画电路的实际运行状态。程序运行信息反映了电路在运行时的实际输入、输出值,通过它们可以获知电路的开关电容变化情况,为生成更精细化的功耗模型提供了保障。

文章第2节介绍了本文生成体系结构级 SPM 部件功耗模型所使用的方法;第3节详细介绍新功耗模型的构造过程;第4节为基于新功耗模型的仿真实验结果,包括与现有功耗模型在能耗计算准确度方面的比较以及新功耗模型用于低功耗编译优化的应用示例。

## 2 SPM 部件体系结构级功耗模型构造方法

SPM 部件功耗跟所有 CMOS 电路一样,主要分为由漏电流产生的静态功耗和由各节点电容的充放电过程产生的动态功耗两部分组成<sup>[4]</sup>,其中静态功耗主要由电路制造工艺决定,而动态功耗则由实际的电路活动决定,在实际运行过程中差异明显,因此,如何准确描述电路活动对负载电容以及最终对动态功耗的影响是整个建模过程的关键。

CMOS 电路的动态功耗基本模型为  $P = aC V_{dd}^2 f$ ,其中  $a$  为电路的活跃度因子,取值介于 0 和 1 之间,表示平均每个时钟周期内电路的位翻转数,  $f$  为时钟频率,  $C$  为电路的负载电容,  $V_{dd}$  为电路供给电压。为了准确地计算电路的负载电容,文献[9]提出的方法是建立由相邻两次电路输入值为索引的开关电容表,在仿真的过程中通过查表得出每次输入所产生的开关电容,该方法建立的功耗模型精度很高,但是

十分复杂且难以实现,部件开关电容表的大小与电路可能的输入值呈幂指数增长,  $n$  个输入则需要建立大小为  $2^{2^n}$  项的表,其开销是模拟器无法承受的。文献[4-7,10]的方法是从电路结构出发,同时忽略电路实际活跃度的差异,根据统计值估算出各子电路的平均功耗模型,最后合成整体的功耗模型。

本文所采用的方法是从电路结构特征,以及电路实际运行情况对电路能耗影响两方面综合考虑来构造 SPM 的功耗模型,通过电路结构特征分析得出各个子部件的基本功耗模型,并在其中设置反映电路实际运行情况的动态因子,这些因子由程序运行时信息来实现,反映了电路运行时的实际活跃度,最终得到可以反映程序在不同 SPM 部件访问方式下产生相应功耗差异的动态功耗模型。而为了获取程序运行时信息所采取的方法是在性能模拟器 SimpleScalar 实现 SPM 部件,并将 SPM 部件功耗模型集成到 SimpleScalar 中,在模拟器执行程序过程中读取所需的相关信息。

## 3 SPM 部件的动态功耗模型(DF\_Model)

本节首先介绍了 SPM 部件的结构,依照 SPM 部件的工作过程分别介绍每个阶段所涉及电路的结构和其特点,并按上节所述的方法来为 SPM 部件中各子部件单元生成其基本功耗模型,最后介绍了模型中各动态因子的实现。

### 3.1 SPM 部件结构

SPM 部件主要包括地址解码器、存储单元阵列以及外围的列电路三大组成部分。其中地址解码器的作用是根据地址总线上的值完成地址译码并驱动存储阵列中的字线(Word line);存储阵列由存储单元组成,每个存储单元存储 1bit 信息;外围的列电路包括多路复用以及信号放大等辅助电路。具体的 SPM 部件工作过程按照访问时电路时序可以分为译码、驱动字线、访问存储单元、以及数据输出 4 个阶段,下面从 SPM 的电路结构并结合其每个阶段的工作方式来分析其电路特征,并由此确定该每个阶段所涉及电路的功耗模型。

### 3.2 SPM 各子部件功耗模型

(1)解码器 译码是指由地址总线输入地址值至解码器中完成地址解码工作。图 1 为解码器的电路结构,包括地址线、3-to-8 解码块,以及由或非门等构成的字线驱动电路。按其工作方式分成 3 个阶段,下面分别构造各个阶段时电路的功耗模型。

在第 1 阶段,地址值从地址总线经缓冲器输入到各个解码单元块的输入地址线上,此时电路的负载电容受地址线上的位翻转数量以及地址线上读写端口数影响,据此可得该阶段功耗:

$$E_{DD1} = a_1 \cdot C_{stage1} \cdot V_{dd}^2 \cdot N_{decoder} \cdot ports \quad (1)$$

式(1)中  $N_{decoder}$  为 3-to-8 解码单元的数量,  $ports$  为输入地址线上的读写端口数,  $a_1$  为输入地址线中发生电路位翻转的比

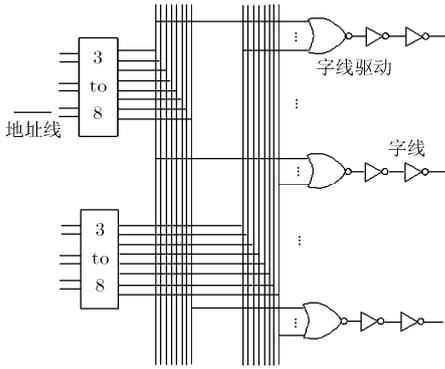


图 1 解码器电路结构图

例值即该部分电路的活跃度因子,  $C_{stage1}$  包含线路漏电容以及门电容等,  $V_{dd}$  为工作电压,  $E_{DD1}$  为该阶段能耗。此外, 式(1)以及下列各式中的电容变量单位为 pF, 电压变量单位为 V, 能耗变量单位为 nJ。

解码器工作的第 2 个阶段是由多个 3-to-8 解码单元完成实际译码工作, 其功耗由每个解码单元产生:

$$E_{DD2} = a_2 \cdot C_{stage2} \cdot V_{dd}^2 \cdot N_{decoder} \cdot ports \quad (2)$$

式(2)中  $C_{stage2}$  指的是由 3-to-8 解码单元中 8 个与非门(NAND)以及 1 个用于驱动的非门构成的电路电容, 而  $a_2$  则指的是或非门和与非门组成的电路实际发生位翻转的比例。

解码器最后的工作就是由或非门和一组驱动翻转器来驱动字线。

$$E_{DD3} = a_3 \cdot C_{stage3} \cdot V_{dd}^2 \cdot N_{decoder} \cdot ports \quad (3)$$

式(3)中  $C_{stage3}$  包括翻转器的门限电容以及或非门的漏电容,  $a_3$  为该部分电路的活跃度因子, 尽管每次可能充电驱动不同的字线, 但是由于实际上每次仅驱动一根字线, 故可以近似地取值为 1。

(2)存储阵列 存储阵列是由存储单元构成的矩阵, 每个存储单元由 6 个晶体管构成, 存储单元横向共享一根字线以供选中该组单元, 纵向共享两根位线(即读/写端)以供读取或写入 1 位信息。

存储阵列的工作分为字线选通和存储单元访问两个阶段, 字线选通主要是锁定相应的一组存储单元, 其功耗模型为

$$E_{WL} = a_4 \cdot C_{wl} \cdot V_{dd}^2 \cdot ports \quad (4)$$

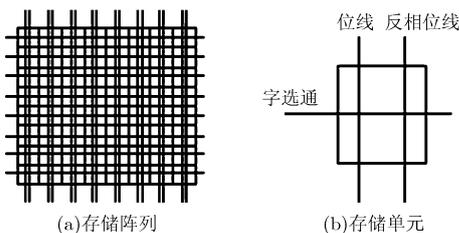


图 2 存储阵列

式(4)中  $C_{wl}$  为电路电容, 主要包括字线驱动器的扩散电容, 以及与位线相连的逻辑门电容等。  $a_4$  为电路活跃度因子, 跟  $a_3$  因子的情况类似, 每次只有一根字线被选通使用, 故近似取值为 1。

接下来通过位线访问存储单元。每条位线被预先充电, 当字线变为高电平时, 位线根据存储单元里存储值为 0 或 1 来产生相应的输出, 此时电路的开关电容主要为位线在预充电过程中的电路电容, 以及读写过程中每个单元的有效负载电容, 由此可以推出访问单元的功耗模型:

$$E_{MEM} = a_5 \cdot ncols \cdot (C_{pre} + C_{readwrite}) \cdot V_{dd}^2 \cdot ports \quad (5)$$

式(5)中  $C_{pre}$  即为位线在预充电过程中的电路电容,  $C_{readwrite}$  为存储单元的有效负载电容,  $ncols$  即由字线选中的存储阵列栏数。  $a_5$  表示是电路中位线发生值翻转的比例, 由于每个单元中的两个位线总是有一根要么充电、要么放电不发生改变, 该值小于 0.5。

(3)列电路 访问 SPM 部件的最后阶段是输入/输出数据值, 主要由列电路来完成, 涉及到列多路复用电路、信号放大以及输入/输出驱动电路的功耗与实际数据值关系不大, 一般采用经验公式来计算其相应功耗<sup>[3]</sup>, 故该部分电路总的功耗模型为

$$E_{COL} = cols \cdot V_{dd} / 8 \cdot 0.5 \cdot 10^{-3} + a_6 \cdot C_{mux} \cdot V_{dd}^2 \cdot ports \quad (6)$$

式(6)中前一部分为信号放大和输出驱动电路的功耗, 后一部分为列多路复用电路的功耗。

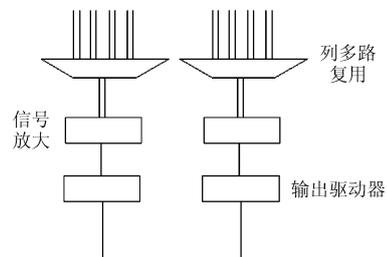


图 3 列电路

$cols$  为列数, 电路活跃度因子  $a_6$  的值取决于输入/输出的实际数据值,  $C_{mux}$  为多路复用电路的有效负载电容, 主要包括线路漏电容和门电容。

### 3.3 DF\_Model 功耗模型中 a 系统因子的实现

DF\_Model 功耗模型中 a 系统因子是根据程序运行时信息来实现的, 反映了各子电路的实际活跃度情况。程序运行时信息包括访问 SPM 部件的地址值以及每次输入/输出 SPM 部件的实际数值。具体的实现是在 SimpleScalar 性能模拟器中加入 SPM 部件模块, 并在访问 SPM 部件处加入相关代码, 根据实际的地址值和数据值来计算 a 系列因子, 详细的实现方式见表 1。

$a_2$  中的 3-to-8 解码器门电路翻转次数是查表完成的。由于电路翻转情况与电路输入(地址线中的值)存在对应关系,

表1 a系列因子的实现

活跃度因子	具体实现
$a_1$	(相邻读写地址的位翻转次数) / (访问次数 $\times 32$ )
$a_2$	( $\Sigma$ 3-to-8 解码器门电路翻转次数) / (访问次数 $\times 32 \times N_{\text{decoder}}$ )
$a_3$	$\approx 1$
$a_4$	$\approx 1$
$a_5$	(数据值中 bit 位为 0 的数目) / (访问次数 $\times 32 \times 2$ )
$a_6$	(相邻数据值的位翻转次数) / (访问次数 $\times 32$ )

故预先根据解码器中逻辑真值表统计出电路翻转情况,在运行时根据两次 SPM 部件访问时的电路输入就可以查出实际的门电路翻转次数。

## 4 实验结果及分析

### 4.1 DF\_Model 模型实验结果

表2是利用本文实现的DF\_Model功耗模型对Mibench测试集<sup>[11]</sup>中程序进行实验的结果。SPM尺寸设定为8KB,2个读/写端口,电路工艺参数选定为0.5 $\mu\text{m}$ 。DF\_Model模型中各子电路电容计算是采用文献[4]中的方法,根据电路所使用的工艺参数以及电路中线路的长宽和各门电路的电气特征来计算。在此配置条件下,目前应用最为广泛Cacti功耗模型计算的功耗值为0.45nJ。

从表2实验结果可以明显地看出,不同类型的程序运行时所导致的电路活跃程度各不相同,产生的功耗也不尽相同。当访问SPM部件的次数达到一定量级时,由此产生的功耗总量差异就会十分可观。同时本文构造的DF\_Model功耗模型在不同应用程序的总体统计平均值下与Cacti生成功耗值一致,实验中应用程序的平均每次访问SPM部件功耗值约为0.45nJ,符合总体统计值,表明结果比较准确可用于实际低功耗优化编译指导。

更重要的通过上述的profile结果,可以清楚地了解各子电路所占功耗总量的百分比,从而有可能通过相应的编译优化手段来降低各子部分所产生的功耗。

### 4.2 模型用于优化应用示例

下面使用bubble-sort程序来说明DF\_Model功耗模型和应用普遍的Cacti<sup>[4]</sup>功耗模型所能提供信息的差异,以及这些信息对编译器进行低功耗优化的指导作用。bubble-sort程序对50个元素进行冒泡排序,其程序代码特征是在嵌套循环中对一组相邻存储的元素进行线性反复多次读取,文献[12]对此类代码的低功耗优化方案是由编译器使用格雷编码方式来对数据地址进行编码,而非常规的二进制编码方式。

图4中的结果为使用该优化方案(对应到DF\_Model2坐标的结果)和未使用该优化方案时程序所产生的功耗以及电路中部分活跃度因子的情况。

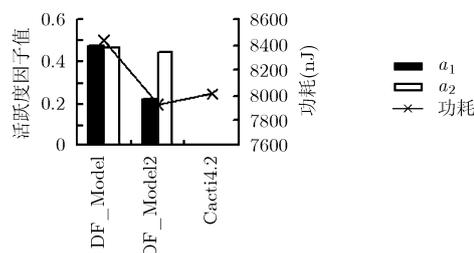


图4 DF\_Model与Cacti仿真结果比较图

由于采用格雷码编址方式使得解码器中地址输入线的位翻转减少,从图4中可以看到电路活跃度因子 $a_1$ 显著减小,总体功耗也因此降低,而使用Cacti等功耗模型无法体现出这两者的差异,优化前后的程序访问SPM的次数相同,因而产生相同的功耗值,无法为编译器提供足够的指导信息。

## 5 结束语

通过获得程序运行时信息生成SPM部件中电路活跃程度的参数因子,结合常规的部件级电路功耗模型方法来构造SPM部件的动态功耗模型。实验表明,基于该模型的profile信息能充分反映不同的SPM部件访问方式和内容布局方式所带来的功耗差异,可用于指导编译器优化访问SPM部件时产生的功耗。以后将在此基础上将模型更精确化,按照物理性的功耗测量结果对本文提出的仿真模型作进一步改进。

表2 DF\_Model模型仿真实验结果

各部分系数	程序名称											
	quicksort		basicmath		jpeg		dijkstra		gsm		bitcount	
	参数	功耗(%)	参数	功耗(%)	参数	功耗(%)	参数	功耗(%)	参数	功耗(%)	参数	功耗(%)
$a_1$	0.24	4.53	0.40	7.79	0.39	6.44	0.36	6.79	0.39	7.71	0.39	7.70
$a_2$	0.48	11.78	0.50	12.40	0.32	11.60	0.49	11.84	0.47	11.87	0.50	12.62
$a_3$	1.0	1.36	1.0	1.40	1.0	1.45	1.0	1.35	1.0	1.41	1.0	1.42
$a_4$	1.0	0.75	1.0	0.77	1.0	0.80	1.0	0.74	1.0	0.78	1.0	0.78
$a_5$	0.42	23.90	0.32	18.45	0.31	18.53	0.39	22.14	0.32	18.53	0.29	17.27
$a_6$	0.17	57.68	0.37	59.20	0.38	61.12	0.23	57.13	0.37	59.70	0.42	60.20
平均功耗(nJ)	0.46		0.45		0.44		0.48		0.44		0.44	

## 参考文献

- [1] Montanaro J, Witek R, and Anne K, *et al.* A 160MHz, 32b, 0.5W CMOS RISC microprocessor[J]. *IEEE Journal of Solid-State Circuits*(JSSC), 1996, 31(11): 1703-1712.
- [2] 李曦, 王志刚, 周学海, 等. 面向低功耗优化设计的系统级功耗模型研究[J]. *电子学报*, 2004, (02): 205-208.  
Li Xi, Wang Zhi-gang, and Zhou Xue-hai, *et al.* Study on system-level power model for low power optimization[J]. *Acta Electronica Sinica*, 2004, (02): 205-208.
- [3] Banakar R, Steinke S, and Lee B, *et al.* Scratchpad memory: A design alternative for cache on-chip memory in embedded systems[C]. In *Proceedings of the Tenth International Symposium on Hardware/Software Codesign*, Colorado, 2002: 73-78.
- [4] Tarjan D, Thoziyoor S, and Jouppi N P. CACTI 4.0 [EB/OL]. <http://www.hpl.hp.com/techreports/2006/HPL-2006-86.pdf>.
- [5] Do M Q, Drazdziulis M, and Edefors P L, *et al.* Parameterizable architecture-level SRAM power model using circuit-simulation backend for leakage calibration[C]. In *Proceedings of 7th International Symposium & Exhibits on Quality Electronic Design, ISQED 2006*, San Jose, March 2006: 557-563.
- [6] Do M Q, Drazdziulis M, and Edefors P L, *et al.* Leakage-conscious architecture-level power estimation for partitioned and power-gated SRAM arrays[C]. In *Proceedings of 8th International Symposium & Exhibits on Quality Electronic Design, ISQED 2007*, San Jose, March 2007: 185-191.
- [7] 李佳, 徐勇军, 李晓维, 等. 体系结构级功耗分析方法[J]. *系统仿真学报*, 2004, 16(12): 2821-2824.
- Li Jia, Xu Yong-jun, and Li Xiao-wei, *et al.* Power analysis methods in architectural level[J]. *Journal of System Simulation*, 2004, 16(12): 2821-2824.
- [8] Najm F. A survey of power estimation techniques in VLSI circuits[J]. *IEEE Trans. on Very Large Scale Integration (VLSI) System*, 1994, 2(4): 446-455.
- [9] Mehta H, Owens R M, and Irwin M J. Instruction-level power profiling [C]. In *Proceeding of International Conference on Acoustics, Speech and Signal Processing*. Atlanta, Georgia, 1996: 3326-3329.
- [10] Brooks D, Tiwari V, and Martonosi M. Wattch: A framework for architecture-level power analysis and optimizations [C]. In *Proceeding of the 27th Intel Symp. on Computer Architecture, ISCA 2000*, Vancouver, British Columbia, 2002: 83-94.
- [11] Guthaus M R, Ringenberg J S, and Ernst D. MiBench: A free, commercially representative embedded benchmark suite[C]. In *IEEE 4th Annual Workshop on Workload Characterization*. Austin, 2001: 1-12.
- [12] Cheng W C and Pedram M. Low power techniques for address encoding and memory allocation[J]. *IEEE Trans. on Very Large Scale Integration(VLSI) System*, 2001, 6(4): 245-250.
- 蒋湘涛: 男, 1978年生, 博士生, 研究方向为低功耗编译优化、嵌入式系统.
- 胡志刚: 男, 1963年生, 教授, 博士生导师, 主要研究领域为分布式系统、网络计算、嵌入式系统、软件工程.
- 贺建飏: 男, 1964年生, 副教授, 研究方向为计算机网络、智能信息处理、嵌入式系统.