

粗粒度可重构处理器的系统级功耗建模

穆昌根, 赵仲元, 绳伟光, 毛志刚

(上海交通大学 电子信息与电气工程学院, 上海 200240)

摘 要: 本文针对粗粒度可重构结构, 提出一种可根据不同结构参数进行拓展的系统级功耗建模方法. 该方法采用层次描述的方法, 分别从体系结构、电路和工艺层建立功耗模型, 再利用线性模型计算出系统的整体功耗. 对比仿真和实测数据的误差, 验证了该建模方法的有效性. 在探索粗粒度可重构架构的早期, 可以应用该方法来评估可重构处理器的功耗.

关键词: 粗粒度可重构处理器; 系统级功耗建模; 动态功耗; 静态功耗

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2018)09-0070-04

System Level Power Modeling for Course-Grained Reconfigurable Processor

MU Chang-gen, ZHAO Zhong-yuan, SHENG Wei-guang, MAO Zhi-gang

(School of Microelectronics, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: In this paper, we propose a system-level power modeling method that can be extended according to different structural parameters for coarse-grained reconfigurable architecture. The method described by the level of methods, respectively, from the architecture, circuit and technology level to establish the power model, and then use the linear model to calculate the overall power consumption of the system. Contrasting the errors of simulation and measured data, the validity of this modeling method is verified. Early in exploring coarse-grained reconfigurable architectures, this approach can be applied to evaluate the power consumption of reconfigurable processors.

Key words: coarse-grained reconfigurable architecture; system level power modeling; dynamic power; static power

1 引言

粗粒度可重构结构(CGRAs)通常由可重构单元阵列(PEA), 配置和数据存储器以及控制单元组成. 控制单元负责任务的调度和数据的传输, 可重构单元阵列负责加速执行内部循环程序. 在 CGRAs 中, 配置数据和配置字大小的降低, 既保留了可重构计算的高性能, 又节约了程序映射的时间.

高效而准确的系统级功耗模型可以帮助设计人员根据功耗要求, 在系统级架构设计中尽早地定位架构设计存在的问题, 有效缩短功耗仿真的时间. 此外, 系统级功耗仿真支持不同程序行为下可重构系统的功耗评估. 不同算法意味着不同的数据流图与

映射方式, 可重构阵列在运行时有不同的计算单元间数据交流的方式. 科学的系统级功耗评估同样会对编译器的调度与映射起着重要的指导作用.

本文针对粗粒度可重构结构, 提出了一种系统级功耗建模方法.

2 相关技术

2.1 可重构处理单元

经典的粗粒度可重构结构^[1]通常包括主控、DDR 内存、AHB 总线和可重构处理器单元(Reconfigurable Process Unit, RPU). 图 1 展示了本文用于评估功耗的粗粒度可重构处理架构中的 RPU 部分. 每个 RPU 中包含了一个 ARM7 协处理器、一个

可重构运算阵列(Process Element Array, PEA)和一块共享内存(Shared Memory, SM). 每个 PEA 由 16 个处理单元(Process Element, PE)组成, PE 之间通过互连网络 Router 访问. ARM7 负责从主存中拷贝数据到 SM 和在 CGRAs 之间分配调度任务. SM 在数据仲裁器的控制下, 为 PEA 提供数据输入和输出. 最后由 PEA 并行执行程序, 加速计算过程.

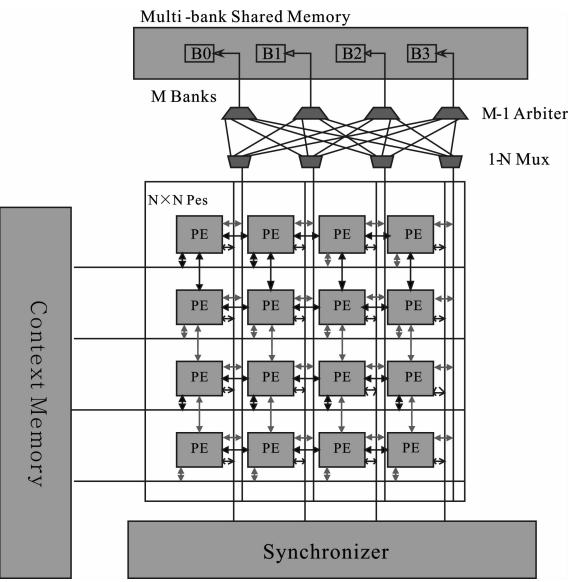


图 1 可重构处理器单元

RPU 功耗在整个系统中占比显著. 通过实验测出, 当 RPU 中包含一个 PEA 时, RPU 功耗占比大于 40%. 当含有四个 PEA 时, 功耗占比将达到 65%.

因此, 通过对 RPU 的功耗建模, 可以为分析可重构单元的结构设计提出重要的指导意义.

2.2 相关功耗模型

系统级功耗建模在系统架构设计具有前瞻性的意义, 因此针对于系统级的建模研究较多, 有很多的商业功耗模型可以借鉴. 本文的建模思想基于两款主流的商业功耗评估软件: McPAT 和 CACTI.

2.2.1 McPAT

McPAT (Multicore Power, Area, and Timing) 是一个结构级的模型框架, 集成了对通用多核处理器在功耗、面积和时序三个维度的模型建构^[2].

相比其它结构级功耗评估工具其优点在于: (1) McPAT 模型结构完整, 包括处理器核心、高速缓存、互连线、内存控制器和时钟网络等复杂的部件. (2) McPAT 使用 ITRS^[3] 提供的对动态功耗、静态功耗和短路功耗的技术预测 (technology projec-

tions), 因此仿真结果更为准确. (3) McPAT 的输入是. xml 文件, 内容包括体处理器体系结构的描述和程序行为的统计, McPAT 支持在输入文件中修改处理器结构参数.

2.2.2 CACTI

CACTI 模型可以用来仿真 Cache 和存储器的面积、动态/静态功耗和访存时间/周期时间, CACTI 中提供了用来和其他仿真工具的接口. McPAT 中集成了 CACTI 用来建模存储器部分的功耗, 如 RAM 结构和 CAM 结构. 此外, McPAT 增强了 CACTI 的功能, 如在 CAM 结构中增加了 DFF (D Flip-Flop) 阵列和栅漏电流功耗.

McPAT 对于电路细节概括过于抽象, 忽略了具体电路实现对功耗的影响. 比如 McPAT 在处理 ALU 等复杂单元时, 采用了通用处理器结构的经验公式, 这会给功耗评估带来较大的误差.

本文的功耗模型应用于粗粒度可重构处理器结构, 因此需要对 McPAT 功耗模型进行功能模块裁剪, 对于结构规整的阵列模块如指令存储器和寄存器堆等, 沿用 CACTI 功耗模型; 而对于复杂的逻辑结构如 ALU 采用门级功耗建模方法, 使用数字电路设计 EDA 工具仿真获得基本能耗参数, 并可根据工艺和电压进行拓展.

在输入文件上, 针对粗粒度平台上程序的执行特点, 可以手动统计出程序执行的指令数、调用 ALU 算子和 MUL 算子的次数、Load/Store 单元的访问次数以及程序执行所需的时钟周期数.

3 CMOS 电路功耗分类

如图 2 所示, CMOS 电路中功耗来源可分为两类: 静态功耗和动态功耗. 静态功耗又称为泄漏功耗, 产生的本质原因是电路中存在泄漏电流. 动态功耗包括短路电流功耗和开关功耗, 产生的本质是电路在开关切换时充放电容和短路电流的产生.

3.1 静态功耗

静态功耗是指电路在上电后, 不执行任何任务情况下所消耗的功耗. 集成电路中静态功耗主要来源于亚阈值泄露电流和栅端漏电流. 亚阈值泄露电流产生的本质是晶体管的“关不断”特性, 与电路的输入状态有关. 栅漏电流产生是由于栅极氧化层厚度过小时, 发生电子隧道效应形成的泄露电流. 这两部分功耗都在制造工艺进入深亚微米节点后, 越发显著. 65 nm 工艺后, 静态功耗可以占到总功耗的 50%, 低功耗设计问题已经成为集成电路业界的重要课题^[4].

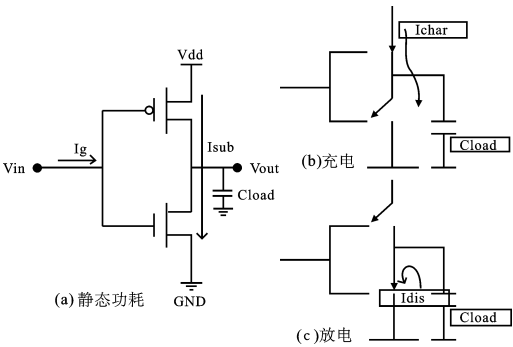


图 2 CMOS 电路功耗基本组成

3.2 动态功耗

门电路在翻转过程中会对电容进行充放电,这是动态功耗产生的本质.经典的动态功耗计算公式如下:

$$P_{\text{switch_dynamic}} = C_L V_{\text{dd}}^2 f \alpha \quad (1)$$

式中, C_L 是门电路的所有负载电容; V_{dd} 是工作电压; f 是工作频率; α 是开关活动因子.

此外,门电路在信号翻转瞬间会产生短路电流,这部分功耗也是动态功耗的构成部分. 计算公式如下:

$$P_{\text{short-circuit_dynamic}} = t_{\text{sc}} V_{\text{dd}} I_{\text{sc}} f \alpha \quad (2)$$

式中, t_{sc} 代表 PMOS 和 NMOS 同时导通的时间; I_{sc} 代表最大短路电流.

4 系统级功耗建模方法

图 3 展示了基于 McPAT 框架的功耗模型的建立方法,该系统级功耗建模针对粗粒度可重构处理器架构,采用层次建模方法,分别从微体系结构层和电路工艺层建立模型.

在微体系结构层,模型中包括了 PEA 的构成模块, PE 内部由算术逻辑运算单元 (ALU)、本地指令存储单元 (Local Instruction Memory, LIM)、和存放临时运算结果的寄存器堆单元 (Resister File, RF) 组成. 在电路和工艺层,模型支持动态功耗、短路功耗和静态泄漏电流功耗.

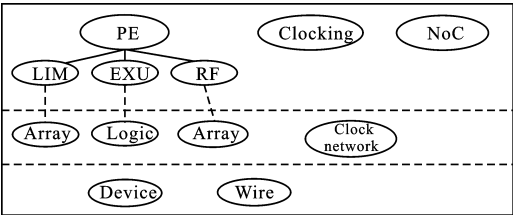


图 3 系统级功耗建模方法

4.1 体系结构级建模

体系结构级描述了粗粒度可重构处理架构的基

本组成模块. PEA 作为可重构阵列的计算载体, PEA 由 $N \times N$ 个 PE 组成, PE 之间由 Router 实现互联. PE 内部结构如图 4 所示.

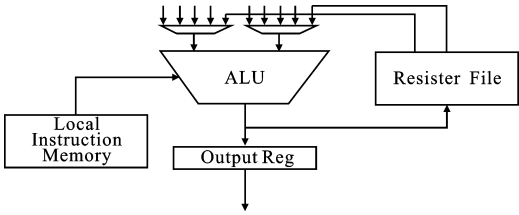


图 4 单个 PE 内部结构

4.1.1 本地指令存储

本地指令存储包括两部分:指令存储器和指令译码器. 指令存储器用于存放每个 PE 的配置包信息,在 PE 开始执行时,会从本地指令存储器读取每行配置信息. 本地指令存储器大小是 32×128 位,每条指令长度 64 位或者 128 位. 指令译码器用于解析每行配置信息对应的操作类型和操作数的来源.

PE 在执行过程中,本地指令存储器的功耗来源于指令的读取. 本地指令存储器的系统级的建模公式为:

$$P_{\text{local_inst_mem}} = E_{\text{per_access_lim}} \times N_{\text{total_access}} / T_{\text{executing_time}} \quad (3)$$

式中, $E_{\text{per_access_lim}}$ 通过 RTL 建模,由 EDA 工具仿真得到. $N_{\text{total_access}}$ 可由程序中统计出, $T_{\text{executing_time}}$ 由仿真器计算执行周期数得到.

4.1.2 执行单元

ALU 是 PE 的执行单元,可以处理多个不同的操作类型. 包括算术运算,如加法、减法、绝对值减法和乘法等操作;逻辑运算单元,如与操作、或操作、异或操作和逻辑移位等操作.

ALU 通过一个 5 位的 OpCode 来选择具体的运算功能. 如图 4 所示 (未完全标出), ALU 将两个 32 位输入和一个 1 位的输入转化成一个 32 位的输出和一个 1 位的输出,1 位的输出用于条件控制或者进位.

ALU 的功耗来自于 PE 执行计算过程中对执行单元的访问操作. 由于乘法操作的复杂和能耗大,乘法单元的功耗单独建模. 执行单元的系统级的建模公式为:

$$P_{\text{alu}} = E_{\text{per_access_alu}} \times N_{\text{total_access}} / T_{\text{executing_time}} \quad (4)$$

$$P_{\text{mul}} = E_{\text{per_access_mul}} \times N_{\text{total_access}} / T_{\text{executing_time}} \quad (5)$$

4.1.3 寄存器堆

CGRA 中的寄存器堆分为全局寄存器堆和 PE 私有寄存器堆. 全局寄存器堆存在于 ARM7 主控

中,用于计算程序中串行部分. PE 私有寄存器用于存放 PE 中计算的输出结果,以用于本 PE 下个计算周期的输入.

$$P_{rf} = E_{\text{per_access_rf}} \times N_{\text{total_access}} / T_{\text{executing_time}} \quad (6)$$

4.1.4 Shared Memory

Shared Memory 作为紧邻 PEA 的数据存储,在数据仲裁器的控制下,为 PEA 提供数据输入和输出. 架构中的 Shared Memory 是一个 16Bank 的存储器,大小为 4096 字,每个字 32 位. 每个 Bank 和 PE 都连接到一个 Arbiter 上,多个 PE 访问同一个 bank 的优先级由 Arbiter 来确定.

$$P_{sm} = E_{\text{per_access_sm}} \times N_{\text{total_access}} / T_{\text{executing_time}} \quad (7)$$

4.2 电路工艺级建模

不同的体系结构最终都需要映射到具体的电路实现,电路级模型主要包括存储阵列模型、逻辑功能单元模型和时钟分布网络模型.

存储阵列在整个系统中通常占用面积大,是决定处理器工作速度的制约因素. 因此为了兼顾到不同的存储结构应用,模型中包括了三类主要的存储模型. 系统中的逻辑功能单元通常采用定制实现,内部结构不易获知. 模型中采用数字电路设计方法,利用 EDA 工具仿真出功能模块的功耗,并给出可根据工艺拓展的建模方法. 时钟分布网络负载大,时序要求严格,不同于一般的布线网络,需要独立建模.

为了反映出工艺迭代对功耗的影响,模型中采用 ITRS 制定的工艺发展路线建立工艺级模型.

4.2.1 阵列结构

功耗模型中主要包括三类基本阵列结构, RAM-(Random Access Memory, 随机存取存储器), CAM-(Content-Addressable Memory, 内容可寻址存储器)和 DFF-(D flip-flop, D 触发器)阵列. 这些阵列结构可以直接调用 CACTI 的接口实现^[5]. 其中 RAM 阵列结构比较规则,功耗主要来源于对子线和位线的充放电,随着读写端口的增加功耗值也随之增加. CAM 阵列和 RAM 在阵列结构上相似,两者主要区别在于译码逻辑和外部端口不同.

4.2.2 逻辑功能部件

根据数字电路成熟的设计流程,使用 Verilog 描述出功能单元的 RTL 级模型,逻辑功能验证后借助 EDA 工具来分析模块的功耗.

Synopsys 公司的 Prime Time PX 工具结合特定工艺厂商提供的单元库自带的功耗模型和用户提供的门级网表、时钟约束、仿真得到的翻转率文件和线网寄生参数等信息,计算出整个电路的功耗.

4.2.3 时钟分布网络

时钟分布网络用来为不同时钟域分配不同频率的时钟,本质上是一种层次结构的互连网络. 时钟分布网络和一般互连结构区别在于:(1)有着严格的时序约束要求,用于驱动大负载;(2)采用波浪流水线(wave-pipelined)结构^[6],不同于普通互连的流水线结构,占据芯片总功耗较大. 基于以上原因,McPAT 为时钟分布网络单独建立模型.

4.2.4 工艺级模型

McPAT 目前支持 90 nm、65 nm、45 nm、32 nm 和 22 nm 工艺节点. 模型中根据 ITRS 的路线图(roadmap)推导不同工艺节点下对应的器件参数,根据文献^[5,7]推导不同工艺下的连线参数. ITRS 在 36nm 后引入绝缘体上硅(Silicon-On-Insulator, SOI)工艺,25nm 后引入双栅(Double gate, DG)工艺.

5 实验结果及分析

为了验证本文提出的系统级和门级相结合的功耗模型的有效性,从 MiBench^[8]选取 14 个测试用例,分别用于在实验板和功耗模型中收集功耗数据. 实验板包括了 ARM7 主控、DDR 内存、AHB 总线和 RPU 阵列,实测中可以直接测出 CGRA 部分的功耗.

因此,为了去除 ARM7 部分的功耗,获得 PEA 部分的实测功耗,分别在 4PEA 和 1PEA 上执行测试用例. 1PEA 的执行功耗由公式(8)计算得到:

$$P_{1\text{PEA}} = (P_{4\text{PEA}+\text{ARM7}} - P_{1\text{PEA}+\text{ARM7}}) / 3 \quad (8)$$

表 1 不同 Kernel 的实测和仿真功耗对比			
Kernel	实测/w	仿真/w	误差/%
aes	0.010 845 221	0.006 821 78	-37.10
autcor	0.010 144 875	0.009 144 87	-9.86
bezier	0.007 666 859	0.006 563 03	-14.40
dotp	0.013 742 904	0.009 976 81	-27.41
fil	0.012 436 363	0.010 101 01	-18.78
fir	0.009 447 235	0.007 484 65	-20.77
gemm	0.005 924 439	0.005 085 28	-14.16
ifft	0.004 393 465	0.004 059 83	-7.59
maxsubstring	0.004 491 081	0.002 528 6	-43.69
mcf	0.004 891 636	0.004 440 88	-9.21
montecarlo	0.004 874 562	0.003 418 05	-29.88
nbody	0.003 659 878	0.002 412 15	-34.09
spmv	0.006 892 498	0.006 023 93	-12.60
unstructured	0.006 187 533	0.004 178 38	-32.47
		平均误差	-21.58

表 1 显示了实测功耗和仿真功耗的数值大小和误差对比,可以看出本文提出的模型较实测功耗偏
(下转第 78 页)

132-137.

[7] 邱云飞, 王威, 刘大有, 等. 基于方差的 CHI 特征选择方法[J]. 计算机应用研究, 2012, 29(4): 1304-1306.

[8] 中国科学院计算技术研究所. NLPir[EB/OL]. [2017-03-14]. <http://ictclas.nlpir.org/>.

[9] Chih-Jen Lin, LIBSVM[EB/OL]. [2016-11-22]. <https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>.

(上接第 73 页)

小,平均功耗误差为-21.58%。偏小的原因在于 CGRA 部分内部结构细节无法获取,在建模时略去了对该部分结构的建模。从仿真数据误差来看,本文提出的粗粒度可重构处理器功耗模型具有一致性,可以反映出系统的整体功耗趋势,因此可以作为粗粒度系统结构初期设计功耗的评估工具。

6 结束语

系统级功耗模型可以为系统结构设计的初期提供功耗评估,本文提出的粗粒度可重构结构的系统级功耗模型,兼顾了系统级的功耗评估速度和门级结构的仿真精度。实验结果显示,预测功耗和板级实测功耗相比,平均误差在-21.58%,说明了该系统级功耗模型的有效性。因此,本文提出的功耗模型可以帮助系统结构设计者根据功耗要求,准确高效地定位出结构问题,同时对编译器的调度和映射起到指导作用。

本文提出的功耗模型目前只考虑了 CGRA 中的 RPU 部分,没有包括 ARM 主控、内存访问以及互连结构的功耗。在深亚微米工艺时代,互连功耗在集成电路中占比越来越高,本文对这部分功耗建模尚未涉及。因此,后续的工作将在此基础上进行更加完整的功耗建模,为体系结构设计人员增加更大的设计探索空间。

参考文献:

[1] Zhongyuan Zhao, Weiguang Sheng, Naifeng Jing, et al. Resource-saving compile flow for coarse-grained reconfigurable architectures [C] // ReConFigurable Computing and FPGAs (ReConFig), 2015 International Conference on. Riviera Maya, Mexico, 2015.

[2] Sheng Li, Jung Ho Ahn, Richard D, et al. McPAT;

作者简介:

宋呈祥 男,(1989-),硕士研究生.研究方向为机器学习与自然语言处理. E-mail:scx929@163.com.

陈秀宏 男,(1964-),博士(后),教授.研究方向为模式识别,图像处理及人工智能等.

牛强 男,(1992-),硕士研究生.研究方向为机器学习与图像处理.

an integrated power, area, and timing modeling framework for multicore and manycore architectures [C]//Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on. New York, NY, USA, 2009.

[3] International technology roadmap for semiconductors[EB/OL]. [2017-10-10]. <https://www.itrsgroup.com>.

[4] Mukherjee R, Mondal S, Memik S O. Thermal sensor allocation and placement for reconfigurable systems [C] // Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design. Washington, USA, 2006; 437-442.

[5] An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model[EB/OL]. [2017-10-13]. <http://www.hpl.hp.com/research/cacti>".

[6] Weste N, Harris D. CMOS VLSI design: a circuits and systems perspective[M]. 3rd ed. USA: New Jersey, Addison Wesley, 2004.

[7] Horowitz M, Ho R, Mai K. The future of wires [EB/OL]. (1999-10-01). <http://velox.stanford.edu/>.

[8] Guthaus T, R Ringenberg M, Ernst J S, et al. MiBench: a free, commercially representative embedded benchmark suite[C] // IEEE International Workshop on Workload Characterization. [s. l.]. 2001; 3-14.

作者简介:

穆昌根 男,(1991-),硕士研究生.研究方向为粗粒度可重构处理器系统级功耗建模. E-mail:alfred_mu@sju.edu.cn.

赵仲元 男,(1989-),博士研究生.研究方向为可重构计算编译架构.

绳伟光 男,(1977-),助理研究员.研究方向为高可靠集成电路设计.

毛志刚 男,(1962-),教授.研究方向为系统级芯片设计.