面向三维多核片上系统的热感知硅后能耗优化方法

靳 松[1)] 韩银和[2)] 王 瑜[1)]

[1)](华北电力大学电气与电子工程学院电子与通信工程系 河北 保定 071001)

[2)](中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)

摘 要 高能效(Energy efficiency)已成为目前嵌入式多核片上系统(System-on-Chips,SoCs)设计中的首要优化目标.基于电压/频率岛设计的三维多核片上系统能够为构建高能效系统提供一种有力的解决方案.然而,不断增加的工艺偏差导致制造后芯片中电压/频率岛的性能参数偏离其额定值.在较大偏差的影响下,可能无法满足任务的截止时间约束.另外,已有的研究工作大多针对二维平台,无法很好地解决因三维集成而不断恶化的发热问题.面向采用电压/频率岛设计的三维多核 SoC,文中提出一个硅后优化框架,在最小化系统能耗的同时,能够满足任务截止时间和系统热约束.除了能效感知的任务调度和电压/频率分派方法,提出的优化框架还采用任务迁移平衡核栈的功耗以实现热优化.实验结果表明,与已有的热平衡方法比较,文中提出的方法能减少平均18.6%的能耗.同时,与经典的能耗优化方法比较,文中提出的方法能降低平均5.6℃的峰值温度.

关键词 系统能耗;三维多核片上系统;工艺偏差;电压/频率岛;任务调度;热优化

中图法分类号 TP393 **DOI号** 10.11897/SP.J.1016.2016.01763

Thermal-Aware Post-Silicon Energy Optimization on 3-D Multi-Core SoCs

JIN Song[1)] HAN Yin-He[2)] WANG Yu[1)]

[1)](*Department of Electronic and Communication Engineering, School of Electrical and Electronic Engineering, North China Electric Power University, Baoding, Hebei* 071001)

[2)](*State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

Abstract Energy efficiency has been become a primary design concern for embedded multi-core system-on-chips (SoCs). Three dimensional (3-D) multi-core SoC based on voltage/frequency islands (VFI) has been recognized as a promising solution for building an energy efficiency system. However, the ever-increasing process variation (PV) leads to performance parameter of the VFI deviating from the nominal values. As a result, violation of the task deadline constraint may occur at post-silicon under large variations. Moreover, the existing work commonly targeted 2-D platform, which cannot address the exacerbated thermal issues from 3-D integration. In this paper, we propose a post-silicon optimization framework targeting VFI-based 3-D multi-core SoCs to minimize system energy meanwhile still meeting task deadline and thermal constraints. Besides energy-aware task scheduling and voltage/frequency assigning, we also conduct task migrating to balance the powers across the core stacks for thermal optimization. Experimental results demonstrate that on average our framework can achieve an energy reduction of 18.6% over the prior thermal balancing algorithm. Moreover, on average a reduction of 5.6℃ in peak temperature is achieved by our framework, compared with the state-of-art energy optimization scheme.

Keywords system energy; 3-dimensional multi-core system-on-chip; process variation; voltage-frequency island; task scheduling; thermal constraint

---

收稿日期:2015-03-18;在线出版日期:2015-07-23.本课题得到国家自然科学基金(61204027)、河北省自然科学基金(F2013502274)、中央高校基本科研业务费专项资金项目(2014ZD32,13MS69)资助.**靳 松**,男,1977年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为大规模集成电路设计和测试、低能耗系统结构、系统可靠性设计及偏差容忍.E-mail:jinsong@ncepu.edu.cn.**韩银和**,男,1980年生,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为容错芯片、计算机体系结构和集成电路测试.**王 瑜**,男,1979年生,博士,讲师,主要研究方向为电气信息处理、模式识别.

1 引 言

集成电路制造工艺的不断进步大大增加了硅片上晶体管的集成密度.相应的,嵌入式系统设计开始转向多处理器/多核片上系统(Multi-Processor/Multi-core System-on-Chip,MP/Multi-core SoC).一般来说,多核 SoC 中包含多种处理芯核,例如通用处理器、数字信号处理器、图形处理器和低功耗处理器等.通过将不同类型的处理芯核集成到一个硅片,多核 SoC 能够提供完整的系统功能.对于不断增加的种类繁多的应用程序而言,多核 SoC 无疑具有广阔的应用前景.

对于高性能、高端片上系统而言,三维多核结构极具吸引力[1-2].一方面,三维集成能够克服传统二维芯片上普遍存在的全局互连延迟和功耗瓶颈问题.另一方面,多核结构不仅能提高系统吞吐量,还能够提高偏差影响下的系统鲁棒性.正是因为结合了上述优点,三维多核 SoC 非常适合用于复杂系统以解决未来种类繁多的应用需求.

由于嵌入式 SoC 经常采用电池供电,高能效、低能耗就成为一个重要的设计目标.近年来,人们向二维多核设计引入电压/频率岛(Voltage/Frequency Island,VFI)以优化系统能效[3-5].VFI 是将芯片上的处理芯核划分为不同的电压/频率域.每个 VFI 都可以运行在各自最优的电压和频率下.通过与任务调度相结合,上述设计方法能够实现细粒度的能耗管理和优化,且满足任务的截止时间约束.

然而,随着晶体管特征尺寸的不断缩小,芯片制造过程中引入的工艺偏差(Process Variation,PV)也日益严重,给多核 SoC 设计带来严峻挑战[6-7].在工艺偏差影响下,制造后的 SoC 芯片上处理芯核的性能参数(如频率、功耗)常常偏离设计阶段所指定的额定值且应被看作是随机变量.因而,对于量产 SoC 芯片而言,处理芯核的性能参数表现为统计分布.相应的,多核 SoC 所执行的应用程序的执行时间等参数同样具有了概率特征.这种执行时间的不确定性无法保证程序的执行在各种工艺拐点(process corner)处都能满足系统的实时性约束.

已有的研究工作通常是在设计阶段对 VFI 进行划分,或者假定芯片具备每核 VFI 的配置.然而,随着工艺偏差日益恶化,芯片制造后 VFI 的性能参数可能偏离设计额定值.因此,很难保证设计阶段所制定的 VFI 划分方案在硅后仍然是最优解.另一方面,对于大规模多核芯片来说,每核电压/频率域(即每核 VFI)配置很难实现.随着处理芯核数目的增加,每核 VFI 需要更多的片外电压规整器(regulator)或片上电压规整器.前者数量的增加会极大地提高封装成本;而后者功率传输效率较低[8].由此可见,对于大规模多核芯片,每个 VFI 包含多个处理芯核较为实际.

再者,与二维芯片相比,面向采用电压/频率岛设计的三维多核芯片进行能耗优化还需要解决一些新的挑战.例如,三维集成导致芯片的散热问题日益严峻.不断增加的功耗密度恶化了热斑(hot spot),同时在芯片上产生高温[9].工作负载的异质性(heterogeneous)则会在处理芯核间造成功耗偏差,导致芯片上出现热梯度(thermal gradient).高温和热梯度不仅会降低系统性能和可靠性,而且将会抵消优化系统能耗的努力.

为了解决上述问题,面向采用 VFI 设计的三维多核 SoC,本文提出一个硅后优化框架,在最小化系统能耗的同时,满足任务截止时间和系统热约束.首先,根据硅前确定的 VFI 划分方案以及工艺偏差造成的性能参数偏差,提出能效感知的任务调度算法.该算法统一考虑后续的电压/频率分派以最小化任务的执行能耗.随后,提出任务迁移算法,在任务图的执行过程中实现核栈间的功耗平衡,降低芯片温度.实验结果表明,与已有的热平衡方法比较,本文提出的方法能减少平均 18.6% 的能耗.同时,与经典的能耗优化方法比较,本文提出的方法能降低平均 5.6℃ 的峰值温度.

2 相关研究现状

面向采用 VFI 设计的多核芯片优化能耗方面,Ogras 等人[10]提出了 VFI 划分与静态电压/频率分派的混合方法优化二维 NoC 的系统能耗.他们首先将每个处理芯核看成是单独的 VFI,按照任务调度的结果确定每个处理芯核的最低操作电压.随后将处理芯核进行合并以形成新的 VFI.VFI 的数目取决于系统设计约束.考虑通信能耗,文献[11-12]提出了电压调节和任务调度算法相结合的方法优化二维多核芯片的能耗.其中,文献[11]整体考虑任务的执行能耗和通信能耗,提出了有效的启发式算法进行任务的调度.文献[12]则将任务调度完成后的处理芯核电压分派模型化整数线性规划问题进行求解.面向二维 NoC,Jang 等人[13]提出了 VFI 感知的能

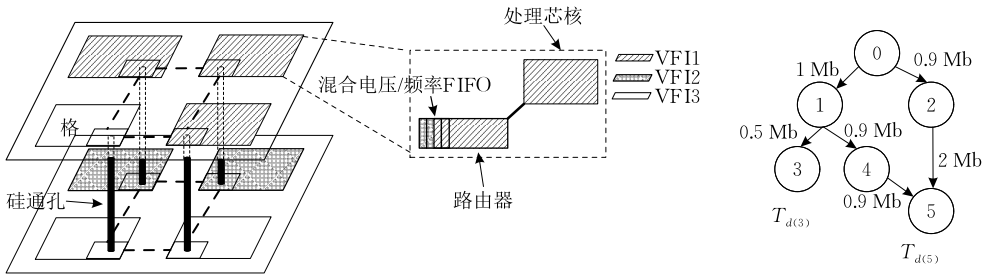
耗优化框架，通过在设计阶段确定最优的处理芯核映射以及路由算法来优化系统能耗. 假定多核芯片具有每核电压/频率域的配置，Zhang 等人[14]提出了动态电压/频率调节和任务调度相结合的方法优化系统能耗. 然而，他们的方法不适合大规模多核芯片. 另外，由于热特性的差异，上述文献提出的方法无法直接应用于三维芯片. 面向三维多核芯片，Cheng 等人[15]提出了考虑热约束的任务分配和调度算法优化通信能耗. 然而，他们的工作忽略了计算能耗的优化.

在三维芯片的热优化方面，也有许多有价值的研究工作. 硬件方面，Goplen 等人[16]提出插入热过孔来加速三维芯片的散热. 针对三维芯片特定的面积，他们借助表面元素分析迭代计算热过孔的插入位置以及调整热传导性，从而有效地降低芯片温度. Wong 等人[17]提出温度感知的热过孔布局方法来降低芯片温度. 基于随机行走算法，他们构建了一个新的热分析模块，而热分析的结果则被用来指导确定热过孔的插入位置. Bakir 等人[18]提出采用流体冷却方法帮助三维芯片散热. 他们的方法集成了电气、光和微流体互连. 软件方面，文献[19]提出动态热管理技术处理程序运行时的热紧急(thermal emergency)情况. 他们提出 Adapt3D 算法，通过考虑处理芯核的热历史信息以及三维芯片特性来平衡芯片温度. Zhou 等人[20]提出均衡线程分配算法，通过考虑线程间的热差异进行线程分组并调度到核栈上，以实现三维多核芯片的热均衡. Zhu 等人[21]提出结合任务调度和电压调节实现三维多处理器 SoC 的热优化. 通过考虑工作负载和异构处理芯核本身的热差异，提出了操作系统级的动态热管理技术. 然而，上述研究工作均只关注热优化，忽略了系统能耗的优化.

3　背景知识介绍

3.1　目标平台和应用

参考商用多核处理器以及目前对于三维多核芯片的研究工作，本文中的目标平台定义如下. 如图 1(a)所示，目标平台为三维同构多核 SoC. 平台包含多个处理芯核层[1,22]. 每一层采用格状(tile)结构，每个格内包含一个集成私有高速缓冲存储器(Cache)的处理芯核以及一个路由器[5]. 采用片上网络(Network-on-Chip，NoC)实现格间通信. 同一层 NoC 为网状(mesh)结构；层间通信通过硅通孔(Through Silicon Via，TSV)总线实现. 本文假定处理芯核可以工作在几个不同的离散电压/频率范围内. VFI 由片外电压规整器支持. 每个电压规整器支持一个 VFI 的电压域. 每个处理芯核假定拥有自己的数字锁相环(DLL)部件以实现独立的频率域. 不同 VFI 间的数据同步由 VFI 边界处的混合电压/频率先入先出(FIFO)缓存支持.



(a) 三维多核SoC　　(b) 通信任务图

图 1　目标平台和目标应用示意图

目标应用为具有高确定性的通信任务图. 如图 1(b)所示，通信任务图表现为有向非循环图. 图中，顶点表示任务. 很多工业级基准任务图中都给出了每个任务在不同类型处理芯核上执行的功耗和延迟. 图中有方向的边则表示任务间的控制和数据依赖关系. 即某一任务必须在其所有前继任务执行完成且完成数据通信后才能开始执行. 有向边上标示的数字表示任务间的通信量. 一般情况下，每一个叶节点处均会有一个截止时间约束(如图 1(b)中的 $T_{d(3)}$ 和 $T_{d(5)}$). 表示这一任务通路(由多个任务串联组成，一般由初始任务节点开始到某一叶节点任务结束)所规定的最晚完成时间.

3.2　能耗模型

根据上述对目标平台和目标应用的描述，可以将系统能耗 E_{sys} 表示为计算能耗与通信能耗之和：

$$E_{sys}=E_{comp}+E_{comm} \tag{1}$$

式(1)中，E_{comp} 和 E_{comm} 分别表示计算和通信能耗.

计算能耗主要源自任务在处理芯核上的执行.

对于包含 n 个处理芯核的芯片，E_{comp} 可以表示为

$$E_{\text{comp}}=\sum_{i=1}^{n}(NC_j\cdot C_i\cdot V_i^2) \tag{2}$$

式(2)中，NC_j 表示任务的执行周期数，C_i 表示核 i 每周期的平均开关电容，V_i 表示核 i 的操作电压.

为了计算总的通信能耗，参考文献[10,15]，首先定义由核 i 到核 j 传输一位所消耗的位能耗：

$$E_{\text{bit}}=\sum_{i}^{n_V}(E_{\text{bit}}^{R}(i)+E_{\text{bit}}^{\text{Link}}(i)+E_{\text{bit}}^{\text{FIFO}}(i))\frac{V_i^2}{V_{dd}^2} \tag{3}$$

式(3)中，E_{bit}^{R} 表示路由器消耗的位能耗，$E_{\text{bit}}^{\text{Link}}$ 表示互连线消耗的位能耗，$E_{\text{bit}}^{\text{FIFO}}$ 表示混合电压/频率 FIFO 消耗的位能耗，n_V 表示根据路由算法得到的核 i 到核 j 的跳数(hop)，V_i 表示 VFI_i 的工作电压.

根据计算所得的位能耗，包含 m 个通信事务的任务图总的通信能耗可以表示为

$$E_{\text{comm}}=\sum_{k=1}^{m}E_{\text{bit}}\times Q_k \tag{4}$$

其中，Q_k 表示核 i 与核 j 之间通信事务 k 的数据通信量.

3.3 延迟模型

基于任务 j 的执行周期 NC_j，该任务在操作频率为 f_i 的核 i 上的执行时间可以表示为

$$T_{\text{exe}}=NC_j/f_i \tag{5}$$

对于通信事务 k，通信延迟表示为

$$T_C=\sum_{i}^{n_V}\frac{NC_R}{f_i}+\sum_{j}^{n_V-1}\frac{NC_{\text{FIFO}}}{f_j}+\frac{Q_k}{W} \tag{6}$$

式(6)中，NC_R 表示一个位片(flit)经过一个路由器及外部互连所需的时钟周期数，NC_{FIFO} 表示位片经过混合电压/频率 FIFO 所需时钟周期数，W 为系统带宽. 很明显，式(6)中的前两项表示头片(header flit)的传输延迟. 最后一项表示数据串行化延迟.

3.4 热模型

图 2 给出了三维多核芯片的热模型示意图[23]. 该模型将芯片面积划分为网格. 每个格对应一个热模型元素，包含热电阻、热电容和一个电流源. 格内的温度假定为均匀的. 这种细粒度的格状热模型可以很容易地与 HotSpot[24] 软件进行结合以计算芯片温度.

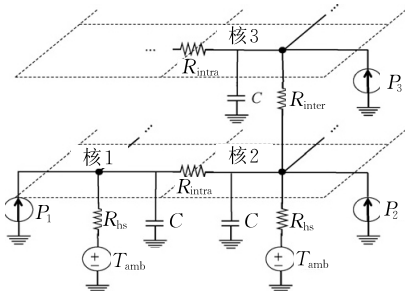


图 2 三维多核芯片热模型示意图

参照此热模型图，核 2 和核 3 的温度可以计算如下：

$$\begin{aligned}T_3&=P_3\cdot R_{\text{inter}}+T_2,\\ T_2&=(P_2+P_3)\cdot R_{\text{hs}}+T_{\text{amb}}\end{aligned} \tag{7}$$

式(7)中，P_2 和 P_3 表示核 2 和核 3 的功耗，R_{inter} 表示垂直方向上核 2 与核 3 之间的热电阻，R_{hs} 表示处理芯核与周围材料之间的热电阻，T_{amb} 表示周围环境温度.

由式(7)可以看出，假定热电阻参数为常量的情况下，处理芯核执行程序时的温度主要取决于它的功耗. 另外，如文献[23]所示，不同层的处于垂直方向的处理芯核之间有着较强的热相关($R_{\text{intra}}\cong 16R_{\text{inter}}$). 由此可见，在垂直方向的核栈间保持功耗平衡能够有效地平衡芯片的温度.

3.5 统计偏差模拟

出于实验的需求，我们借助统计偏差模拟分析和计算参数偏差影响下三维多核 SoC 中处理芯核的频率和功耗分布以及分布的相关性信息. 偏差模拟首先从模型化晶体管一些典型物理参数(如沟道长度和阈值电压)的偏差开始. 在模拟中，片间随机性偏差、片内随机性和具有空间相关性的系统性偏差全部予以考虑. 其中，同一层的硅片里器件的参数偏差主要取决于片内偏差. 由于三维芯片不同层的硅片一般来自于不同的晶圆(wafer). 因此，模拟中，不同层的硅片间的片间偏差假定为独立的. 按照上述约定，某一层硅片中晶体管的某一参数的偏差可以表示为

$$\Delta P=\Delta P_{\text{inter}}+\Delta P_{\text{sys}}+\Delta P_{\text{ran}} \tag{8}$$

式(8)中，ΔP_{inter} 表示片间随机性偏差，ΔP_{sys} 表示片内系统性偏差，ΔP_{ran} 表示片内随机性偏差.

模拟中，首先按照多核芯片的版图，将整个硅片面积划分成许多相等尺寸的网格(grid). 在每一个格内，均包含唯一一个表示片间随机性偏差、片内系统性和随机性偏差的随机变量. 随机变量假定服从标准正态分布. 同时，我们采用文献[25]提出的 VARIUS 模型来刻画片内系统性偏差分布的空间相关性. 根据 VARIUS 模型，任意两个网格内片内系统性偏差分布的空间相关性可以表示为

$$\rho(r)=\begin{cases}1-\dfrac{3r}{2\varphi}+\dfrac{1}{2}\left(\dfrac{r}{\varphi}\right)^3, & r\leq\varphi\\ 0, & r>\varphi\end{cases} \tag{9}$$

式(9)中，φ 表示任意两个网格内器件的参数分布相关性变为零的物理距离范围，r 表示两个网格的物理距离，$\rho(r)\in[0,1]$，表示空间相关性的程度，$\rho(r)$ 取值越接近于 1，表示参数分布的空间相关性也就越大. 反之，越接近于 0，则表示参数的分布相关性越小，更适合表现为独立分布.

基于上述的偏差模型，我们采用蒙特卡洛模拟来获取晶体管沟道长度和阈值电压的统计分布数据. 随后，将参数的偏差分布数据送入关键通路模型[7,25]以获取不同类型处理芯核的频率和功耗分布信息. 关键通路模型以四扇出标准与非门为基本逻辑单元，借助 HSPICE 电路仿真确定不同工艺节点下，不同类型的处理芯核所对应的逻辑门的级数. 一个处理芯核内所包含的关键通路数目则可以将处理芯核的版图面积除以具有高度相关性的单位面积(如文献[25]建议采用 0.02 mm^2)获得.

4　优化框架

图 3 给出了本文提出的优化框架的示意图. 如图所示，面向已完成 VFI 划分的三维多核平台，首先采用能效感知的任务调度算法将任务分配到处理芯核上. 与已有的研究工作不同，本文提出的任务调度算法在调度任务时即考虑为后继的处理芯核电压/频率分派保留优化空间. 因此，在算法中采用任务的最低操作电压和频率作为指导参数对任务调度和电压/频率分派进行统一. 同时，任务调度后，任务图的整个执行时间被划分为许多连续的时间片段. 以这些时间片段为参考，任务迁移算法在处理芯核间迁移或交换少量已调度的任务，在整个任务图执行期间实现核栈的功耗平衡. 与已有的在线任务迁移方法不同，本文提出的任务迁移同样是在设计阶段，即任务实际开始执行前完成的. 这样就避免了在线任务迁移所引入的性能和硬件方面的开销.

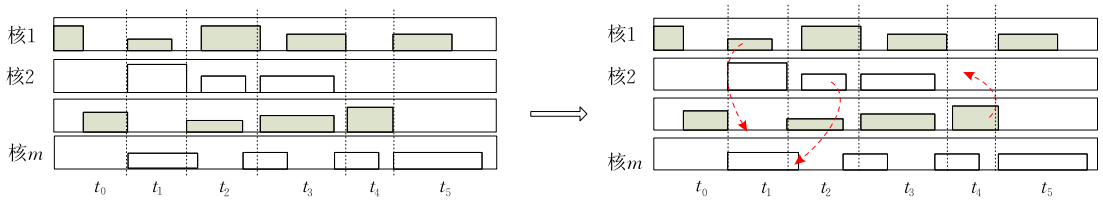


图 3　提出的优化框架示意图

4.1　能效感知的任务调度

面向多核平台，以优化能耗为目的的任务调度已被证明为 NP 困难问题[11]. 为了有效地优化能耗，必须统一考虑任务调度与接下来的处理芯核电压/频率分派. 即任务调度算法必须为后续的电压/频率调节保留尽可能多的优化空间. 为了实现上述目的，本文采用任务的最低操作电压/频率实现任务调度与电压/频率调节的有机统一. 任务的最低操作电压/频率表示为了最小化任务执行能耗且同时满足截止时间约束，执行任务的处理芯核运行时所需的最低电压和频率. 本文提出的任务调度算法将具有相同最低操作电压/频率的任务尽量分配到同一个处理芯核上. 这种调度策略可以保证处理芯核采用所调度的任务的最低电压和频率运行，从而为后续的电压/频率调整保留最大的优化空间.

算法 1 为所提出的任务调度算法的伪代码.

算法 1.　能效感知的任务调度算法.

输入：任务图，可用离散的电压/频率范围

输出：任务调度结果，执行时间序列(ETS)

预处理步骤：

1. 额定电压/频率下，为每个任务计算 E_{comp}；
　　//即任务功耗与执行时间的乘积
2. 为任务图中每条任务通路计算总的 $slack$ 和 E_{comp}；
3. FOR 每个任务 T_i
4. 　FOR T_i 所在的每条通路 P_j
5. 　　$T_{slack(i,j)}=[T_i\_E_{comp}/P_j\_E_{comp}]\times P_{j\_slack}$；
　　//按照任务能耗与所在通路总能耗的比值分配
6. 　$T_{i\_slack}=\min(T_{slack(i,j)})$；　//若某个任务处于多条通路，取计算所得执行时间余量的最小值
7. 　记录 T_{i_start}，T_{i_finish}，$T_{i\_deadline}$；　//计算任务开始执行时间、执行结束时间和截止时间约束
8. FOR 每个任务
9. 　计算任务的最低操作电压及剩余 $slack$；

任务调度：

/*FTL：全任务列表，RTL：就绪任务列表，rt：就绪任

务，ACL：可用处理芯核列表，ac：可用处理芯核，STL：调度时间列表，st：调度时间点 */

10. WHILE(!$FTL.IsEmpty$()){ //全任务列表不空，表示还有任务没有调度
11. st=$STL.GetHead$(); //获得当前调度时间节点
12. $RTL.Add$(当前st下rts)；$ACL.Add$(当前st下acs)；//构建就绪任务列表，构建空闲处理芯核列表
13. WHILE(!$RTL.IsEmpty$()){ //当前就绪任务列表不空，执行调度
14. rt=$RTL.GetHead$(); //取出就绪任务列表中第一个就绪任务
15. IF ACL 中所有 ac 不具备与 rt 相匹配的最低电压
16. 将 rt 调度到空闲的 ac，标记 ac.V/F 为 rt.V/F；
17. ELSE
18. 将 rt 调度到具有最大通信量的匹配 ac；
19. $STL.Add(rt.deadline)$; $STL.Del(st)$; $FTL.Del(rt)$;}}//任务的截止时间加入调度时间节点列表，删除完成调度的任务
20. 根据任务调度结果生成ETS; //对齐任务执行时间，划分执行时间片段

算法首先执行预处理步骤. 这一步骤首先为每个任务分配执行时间余量($slack$)以及计算任务的最低操作电压/频率. 在我们提出的算法中，任务的执行能耗越大，所分配的执行时间余量越多. 为高能耗任务分配较多的执行时间余量能够提供更大的电压/频率调节空间，从而更大程度地降低执行能耗. 任务执行时间余量的分配过程如下. 首先，计算每个任务在额定电压/频率下的执行能耗(行#1)，即该任务在额定电压/频率下的功耗和执行时间的乘积. 同样，为任务图中每一条任务通路(任务串联组成的通路，由任务图中某一个初始任务节点开始一直到某一个叶节点结束)计算总执行能耗和总执行时间余量(行#2). 其中，任务通路的总能耗等于该通路上所有任务计算能耗之和；而任务通路的总执行时间余量可表示为该条通路叶节点上的截止时间约束与通路上所有任务额定执行时间总和之间的差值. 可用如下公式表示：

$$P_{i\_slack}=P_{i\_deadline}-\sum_{m}T_{i\_exetime} \tag{10}$$

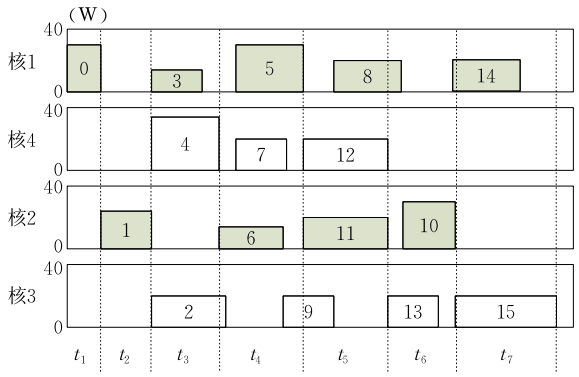
式(10)中，P_{i_slack} 表示任务通路 i 总的执行时间余量，$P_{i\_deadline}$ 表示任务通路 i 叶节点上的截止时间约束，$T_{i\_exetime}$ 表示任务 i 在额定电压/频率下的执行时间，m 表示任务通路上的任务数目.

随后，对于每条通路上每一个任务，按照该任务与所在任务通路的总能耗比值分配执行时间余量(行#3～5). 对于处于多条任务通路交叉点上的任务，按每条任务通路计算所得的执行时间余量可能不同. 这种情况下，取计算所得的执行时间余量中最小值作为该任务的执行时间余量(行#6). 随着执行时间余量的分配，任务的开始时间、结束时间和截止时间约束均可确定(行#7). 同时，对任务的最低电压/频率也可进行计算(行#8～9)，过程如下. 任务的执行时间余量为其截止时间和执行时间之差，可表示为 $T_{\text{exe}}-NC/f_i$. 其中，T_{exe} 表示任务的执行时间；NC 表示任务的执行周期；f_i 表示任务所在处理芯核工作频率. 由上述公式可知，随着处理芯核工作电压/频率的降低，任务执行时间将增加，而执行时间余量则会减少. 因此，执行时间余量接近或等于零时的电压/频率则确定为任务的最低操作电压/频率.

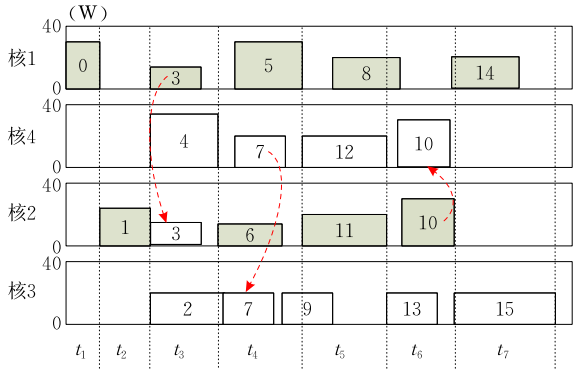
接下来，在每一个调度时间节点，算法将就绪任务调度到处理芯核上. 每次一个任务调度完成，该任务的截止时间即被加入到调度时间列表中成为新的调度时间节点. 在每个调度时间节点，当前的就绪任务和空闲处理芯核分别被加入到就绪任务列表和空闲处理芯核列表中(行#11～12). 就绪任务表示该任务调度前，其所有的前继任务均已调度完毕. 在全部任务图调度的初始阶段，一般会有一些处理芯核从未被分配任务. 这时，对于某个就绪任务，如果当前所有的空闲处理芯核(即从未被分配任务的处理芯核)均没有被标记为与就绪任务相同的最低操作电压/频率，则该任务将被调度到任意一个空闲处理芯核上. 随之，按调度的任务的最低操作电压/频率标记该处理芯核(行#15～16). 相反，如果有些处理芯核已被标记为与就绪任务相同的最低操作电压/频率，则就绪任务将被调度到与该任务有最大通信量的处理芯核上(行#17～18). 与某一就绪任务有最大通信量的处理芯核是指所有已经调度到这个处理芯核的任务与该就绪任务有最大通信量. 根据通信能耗的计算公式(式(4))，将就绪任务调度到与它有最大通信量的处理芯核上可以有效地降低通信能耗. 这是因为数据的通信不需经过片上网络. 确定最大通信量处理芯核的方式则是按照通信任务图中定义的数据相关性(即任务间的通信数据)，计算该就绪任务与每个处理芯核上已经调度的任务之间的通信量，最后找到有最大通信量的那个处理芯核. 上述任务调度过程不断重复，直到成功调度完任务图中的所有任务.

根据任务调度结果，算法随即生成执行时间序列(行#20). 在执行时间序列中，任务图的执行过程被划分成许多连续的执行时间片段. 执行时间片段

的划分采用一种粗粒度方式进行.通过简单地对齐处理芯核上所调度的任务的执行时间划分执行时间片段.图 4(a)展现了将一个含 16 个任务的任务图调度到 4 个处理芯核时划分所得的执行时间序列.执行时间序列包含 7 个时间片段(t_1～t_7).调度的任务以矩形表示.矩形的高度表示任务的执行功耗.填充矩形的颜色则表示处理芯核所运行的最低操作电压/频率.每个时间片段的划分尽量对齐任务的执行时间.即每个时间片段内尽量有尽可能多的同时执行的任务;相邻时间片段之间有尽可能少的重叠执行的任务.



(a) 任务调度及时间序列划分结果



(b) 任务迁移示例

图 4　任务调度及任务迁移示例

4.2　任务迁移算法

任务迁移算法在处理芯核之间迁移或交换已调度的任务.目的是在每个执行时间片段中平衡核栈的功耗,以降低芯片温度.迁移任务时,算法主要利用处理芯核空闲时间以及任务经过电压/频率调节后的剩余执行时间余量.这里需要说明的是,本文所提出的任务迁移算法并非传统意义上的在线任务迁移.本文中的任务迁移与任务调度算法相似,均是在设计阶段,即实际任务执行之前完成.目的是为了平衡核栈间的功耗,实现芯片的热平衡并且降低芯片温度.任务图实际执行期间,即可按照预先确定的任务-处理芯核对应关系进行任务的分配.

算法 2 给出了任务迁移算法的伪代码.

算法 2.　任务迁移算法.

输入:任务调度结果,电压/频率分派结果,核栈,执行时间序列 ETS

输出:任务迁移及最终调度结果

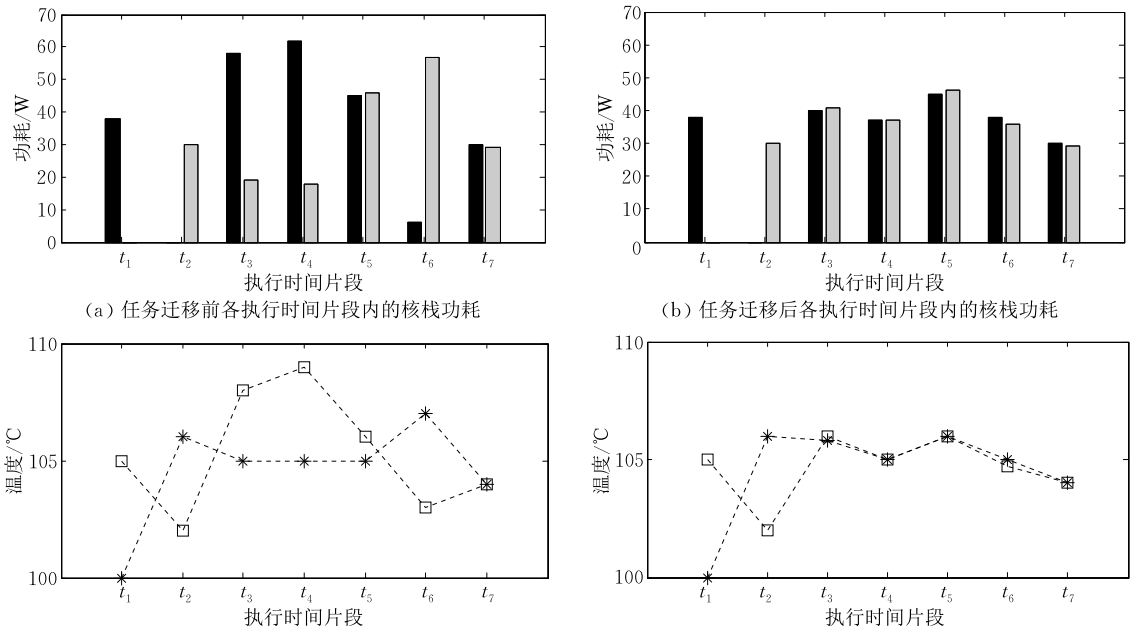
/ $*co$:处理芯核; CS:核栈; CSL:核栈链表; te:执行时间片段; ΣP:核栈总功耗; LPT:核栈中某个核上的高功耗任务; SPT:核栈中某个核上的低功耗任务; IT:空闲时间; ET:执行时间 $*$ /

1. FOR ETS 中每个执行时间片段 te {
2. 为每个 CS 计算 ΣP,计算总功耗标准差 $\Sigma P.\sigma$;
　　//为每个核栈计算总功耗,并计算所有执行片段内的标准差
3. IF $\Sigma P.\sigma$ > 阈值 { //如果标准差大于设定的阈值,启动任务迁移
4. 将 CSL 中所有 CSs 以 ΣP 按降序排列;
5. $i=0$; $j=CSL.GetLengh()$; $CS_h=CSL.GetAt(i)$;
6. WHILE(1){
7. $CS_t=CSL.GetAt(j)$;
8. IF $CS_t.co.IT$ 与 $CS_h.co.SPT.ET$ 相匹配
　　//低功耗核栈处理芯核的空闲时间匹配高功耗核栈任务的执行时间
9. 在 CS_h 和 CS_t 之间迁移任务; break;
10. ELSE IF $CS_h.SPT$ 和 $CS_t.LPT$ 能够交换
　　//可以直接交换任务
11. 在 CS_h 和 CS_t 之间交换任务; break;
12. ELSE
13. $j=j-1$;}
14. 从 CSL 中移除 CS_h 和 CS_t;
15. IF $CSL.GetLenth()>=2$
16. 跳回第 5 步 }}

在每个执行时间片段,对每个核栈计算总功耗 ΣP.同时,计算所有核栈总功耗的标准差 $\Sigma P.\sigma$.如果标准差大于指定阈值(例如实验中取 5%),则启动任务迁移.首先,将所有核栈按总功耗大小按降序排列.随后,针对具有最大总功耗的核栈 CS_h,找出具有最小总功耗的匹配核栈 CS_t.这里,匹配的含义有两层:(1)匹配的核栈其中一方能够为另一方提供空闲时间以供任务迁移;(2)匹配双方可以交换任务.如果条件 1 满足,CS_h 中的低功耗任务(SPT)将被迁移到 CS_t 上(行#8～9).如果条件 2 满足,CS_h 中的低功耗任务将与 CS_t 中的高功耗任务(LPT)进行交换(行#10～11).本次任务迁移完成后,从 CSL 中移除 CS_h 和 CS_t.上述过程不断重复,直到在所有执行时间片段内都进行了任务迁移(行#14～16).

以图 4(a)调度的任务图为例.如图 4(b)所示,

处理芯核 1、4 和 2、3 各组成两个核栈. 现在我们考虑时间片段 t_3 中的功耗情况. 在没有采取任务迁移之前，t_3 期间，包含处理芯核 1 和 4 的核栈总的功耗为任务 3 和任务 4 总功耗之和. 很明显，这个总的功耗远远大于包含处理芯核 2 和 3 的核栈的总功耗(即任务 2 的功耗). 因此，为了在执行时间片段 t_3 中平衡核栈的功耗，任务 3 可以从核 1 迁移到核 2. 原因在于任务 1 和任务 6 之间的空闲时间能够容纳任务 3 的执行. 同理，为了平衡时间片段 t_6 中的功耗，任务 10 可以从核 2 迁移到核 4. 从而保证两个核栈中均有一个执行的任务，以实现在 t_6 中的功耗平衡. 值得注意的是，原本任务 2 与任务 9 之间的空闲时间不足以容纳任务 7 的执行. 幸运的是，任务 7 的前继任务，即任务 2 在电压/频率调节后仍保留有一定的执行时间余量. 因此，迁移算法减少任务 2 的执行时间余量(并没有造成违背任务截止时间的现象)，将任务 7 的执行时间提前，从而可以将任务 7 从核 4 迁移到核 3. 图 5 展示了施行任务迁移前后两个核栈的功耗以及生成的温度. 很明显，任务迁移算法有效地在所有执行时间片段中平衡了核栈的功耗. 相应的，功耗平衡不仅平衡了核栈温度，而且有效地降低了芯片温度.



(a) 任务迁移前各执行时间片段内的核栈功耗

(b) 任务迁移后各执行时间片段内的核栈功耗

(c) 任务迁移前各执行时间片段内的核栈峰值温度

(d) 任务迁移后各执行时间片段内的核栈峰值温度

图 5 任务迁移前后功耗和温度对比

5 实验及讨论

5.1 实验配置说明

5.1.1 实验平台

实验在一个格状 NoC-总线结构的三维多核 SoC 模拟平台上进行. 平台拓扑设为 $4\times4\times2$，即两层处理芯核堆叠，每层处理芯核数目设定为 16 个. 同一层的处理芯核采用网状结构 NoC 互连. 而不同层的处理芯核之间则通过多 TSV 总线进行通信. 芯片绑定方式假定为面向背(Face-to-back)绑定策略. 处理芯核假定为 TILE64 多核处理器中采用的 VLIW 处理器[26]. 处理芯核在 1.0V 额定操作电压下的操作频率设为 500 MHz，可运行在五个不同的电压级别下[0.7 V，0.8 V，0.9 V，1.0 V，1.1 V]. 通过将处理芯核模型化为 4 扇出与非门链，采用基于 45 nm PTM 晶体管模型[27]的 HSPICE 仿真来评估处理芯核在不同供电电压下的最大操作频率. 路由器采用 4 级流水线结构，包含 5 个端口. 除了用于二维平面中东、西、南和北方向通信的 4 个端口外，第 5 个端口用于连接垂直总线以实现垂直方向上的数据交换. 相同格内的处理芯核和路由器假定具有相同的操作频率. 采用确定性 x-y-z 路由算法来避免活锁和死锁. 在 VFI 边界处采用混合电压/频率 FIFO 实现数据同步. 式(3)中的位能耗参考文献[15]计算.

5.1.2 任务图

实验中采用两组任务图. 第 1 组取自工业级基准任务图 E3S①. E3S 中的任务图均给出了所包含

① Dick R. Embedded System Synthesis Benchmarks Suites (E3S). Available: http://ziyang.eecs.umich.edu/~dickrp/e3s/

的任务在各种实际的处理芯核上执行时的功耗和延迟. 不过，E3S 中的任务图所包含的任务数目一般小于实验平台中的处理芯核数目. 因此，参照实验平台中处理芯核的数目，实验中将基准程序中多个任务图组合成新的任务图. 第 2 组采用 TGFF① 生成 6 个伪随机任务图(TG1～TG6)，每个任务图包含 80～100 个任务. 任务图生成过程中，通过更改任务的入度、出度以及通信量来覆盖不同类型的任务. 表 1 列出实验采用的任务图统计信息.

表 1　任务图统计信息

任务图	任务数	平均通信量/Mbit	入/出度
Consumer	12	3.2	1.1/1.1
Auto-industry	24	0.006	0.8/0.8
Networks	13	9.8	1/1
Telecomm	30	0.004	1/1
TG1	88	0.008	1.2/1.2
TG2	98	6.0	1.5/1.5
TG3	100	4.6	2/2
TG4	101	8.8	2/2
TG5	102	7.0	2/2
TG6	100	9.2	2/2

5.1.3　偏差影响下的 VFI 划分图

本文采用文献[25]提出的 VARIUS 模型对参数差偏差进行建模. 标准偏差设为参数期望值的 10%，并进一步分为 6%的片间偏差和 8%的片内偏差. 片内偏差平均分为系统性和随机性偏差两部分. 刻画芯片二维平面上系统性偏差相关性的最大物理距离设为 0.5[25].

整个芯片面积划分为 64 个网格. 每个处理芯核占据一部分网格并被模型化为 100 条四扇出的与非门链. 通过应用 VARIUS 模型，借助 HSPICE 蒙特卡洛模拟获得处理芯核的频率分布数据. 将频率分布的均值作为处理芯核的额定操作频率. 同样，5 种供电电压下处理芯核的操作频率也采用 HSPICE 进行评估.

根据获得的处理芯核的 5 种电压/频率组合，实验中采用文献[10]提出的方法划分 VFI. VFI 划分采用两层统一的方式. 也就是说，同一个 VFI 可能包含垂直方向上位于不同层的处理芯核. 划分过程中，操作电压/频率相接近的处理芯核会被划归于一个 VFI. 每个 VFI 中包含的处理芯核数目可能不同. VFI 的划分从每核 VFI 开始，随后两两合并，直到最终所有处理芯核同属一个 VFI 结束. 对所有划分粒度的 VFI 方案，取能耗最低的那一个作为最终 VFI 划分方案. 表 2 列出与各任务图对应的 VFI 划分结果.

表 2　VFI 划分结果统计信息

任务图	划分后 VFI 数目
Consumer	2
Auto-industry	4
Networks	3
Telecomm	4
TG1	9
TG2	10
TG3	10
TG4	9
TG5	9
TG6	10

5.1.4　热模拟方法

采用 HotSpot 5.0 计算任务执行时的芯片温度. 该软件支持基于网格的三维芯片热模拟[24]. 热模拟参数参考文献[15]中的数据，具体值如表 3 所示. 模拟所需的功耗痕迹(trace)文件通过计算每个调度间隔任务的平均执行功耗获得. 温度计算中只考虑处理芯核的静态温度及芯片的峰值温度.

表 3　热模拟配置参数

参数	值
底层硅片衬底厚度	150 μm
其他层硅片衬底厚度	50 μm
铜金属层厚度	0.42 μm
硅材料热传导性	100 W/(m-K)
散热片热传导性	400 W/(m-K)
Hotspot 格分辨率	64×64
周围介质温度	27℃

5.2　实验结果

出于比较目的，本文修改并实现了文献[20]提出的热平衡算法，使之适用于基于 VFI 设计的三维多核平台. 文献[20]考虑任务间的热特性差异，在每个调度时间节点将高功耗和低功耗的任务组合在一起调度到一个核栈上. 在本文后续的内容中将文献[20]提出的方法称为 TB 算法. 同时，本文还实现了文献[10]提出的能效感知的任务调度算法. 他们的方法通过考虑赋予不同能耗任务以不同的优先级指导任务的分配和调度. 而在 VFI 划分过程中，他们首先将每个处理芯核看成是单独的 VFI，按照任务调度的结果确定每个处理芯核的最低操作电压. 随后将处理芯核进行合并以形成新的 VFI. VFI 的数目取决于系统设计约束. 在本文的后续内容中将文献[10]提出的方法称为 EAS 算法.

5.2.1　能耗优化结果

图 6 给出采用 3 种方法后的能耗优化结果及优化过程中的芯片峰值温度数据. 为了更为清晰地展

① Available: http://ziyang.eecs.umich.edu/～dickrp/tgff/

示结果对比,EAS 和本文提出的方法所取得的能耗优化结果均以 TB 方法取得的结果为参照进行归一化处理.

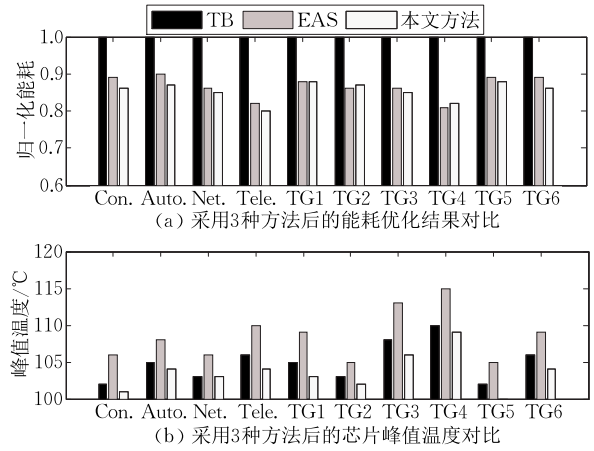


图 6 能耗优化结果和峰值温度数据

由图 6(a)可见,在能耗优化方面,本文提出的方法以及 EAS 算法的表现明显优于 TB 算法. 相比较与 TB 算法,本文提出的方法能减少平均 18.6% 的能耗. 上述结果得益于本文提出的能效感知的任务调度算法. 通过将具有相同最低操作电压/频率的任务调度到同一个处理芯核,处理芯核可运行在所调度的任务的最低电压/频率上,从而最大程度地降低任务执行能耗. 相反,TB 算法将高功耗和低功耗任务组合在一起调度到同一个核栈上. 这种做法虽然可以实现核栈间的功耗平衡,但却难以对能耗的优化产生积极作用.

另一方面,由图 6 所示,虽然在能耗优化方面,本文提出的方法与 EAS 算法的效果相当. 然而,相比较于 EAS 算法,本文提出的方法在优化能耗的同时可以有效地降低芯片温度. 如图 6(b)所示,实施本文提出的方法时,芯片温度略低于 TB 算法. 在达到几乎相同的能耗优化结果的前提下,实施本文提出的方法所产生的芯片温度大大低于 EAS 算法. 与 EAS 算法比较,本文提出的方法能降低平均 5.6℃ 的峰值温度. 以上对于温度的优化效果主要得益于本文提出的任务迁移算法. 通过在核栈间交换或迁移少量的任务,以较小的影响能耗优化为代价,本文的方法可以实现有效的功耗平衡,同时降低了芯片温度.

由上面两方面的比较结果可知,相对于 TB 和 EAS 算法,本文提出的方法可以在能耗优化和降低温度两方面取得最佳的平衡.

5.2.2 热优化结果

图 7 给出了采用 3 种优化方法后的任务图执行过程中的芯片平均温度数据. 实验中,两组共 10 个任务图一个接一个的连续执行. 每个任务图的执行时间被均匀地划分为 10 个执行时间片段. 随后,将总共 100 个时间片段内的功耗痕迹(power trace)送入 HotSpot 以计算所有核栈的平均温度. 如图 7 所示,TB 算法和本文提出的方法均能在核栈间实现温度平衡,达到较为理想的热优化效果. 比较图 7(a)和(c),采用本文提出的方法后,芯片平均温度还略低于 TB 算法. 另一方面,由图 7(b)所示,采用 EAS 方法后,芯片温度出现明显波动偏差. 这是因为 EAS 算法在优化能耗的过程中并没有考虑热优化问题. 上述数据也表明,面向三维芯片的能耗优化方法必须将热问题考虑在内.

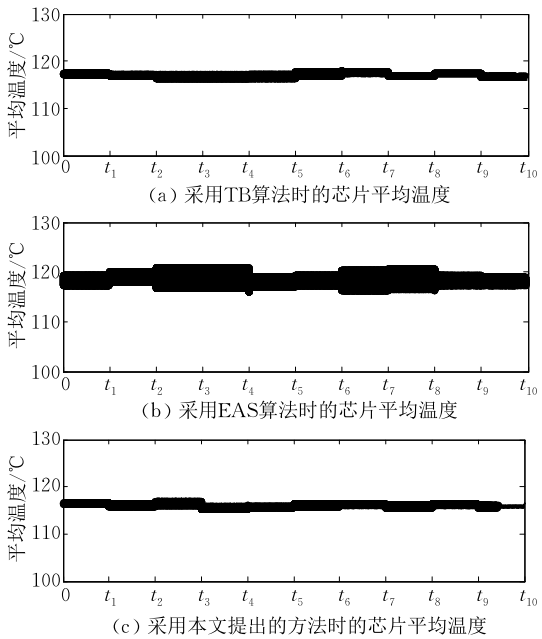


图 7 10 个任务图连续执行时的平均温度数据统计

6 结 论

面向采用 VFI 设计的三维多核 SoC,本文提出一个硅后优化框架,在最小化系统能耗的同时,满足任务截止时间和系统热约束. 提出的优化框架统一考虑任务调度和电压/频率分派,通过识别任务的最低操作电压/频率指导任务调度策略. 同时,通过任务迁移算法平衡核栈的功耗,以达到降低芯片温度的目的. 实验结果表明,本文提出的方法能够在降低系统能耗的同时,有效降低芯片温度.

参 考 文 献

[1] Borkar S. 3D integration for energy efficient system design//Proceedings of the ACM/IEEE Design Automation Conference (DAC). San Diego, USA, 2011: 214-219

[2] Sekiguchi T, Ono K, Kotabe A, et al. 1-Tbyte/s 1-Gbit DRAM architecture using 3-d interconnect for high-throughput computing. IEEE Journal of Solid-State Circuits, 2011, 46(4): 828-837

[3] Herbert S, Marculescu D. Characterizing chip-multiprocessor variability-tolerance//Proceedings of the ACM/IEEE Design Automation Conference (DAC). Anaheim, USA, 2008: 313-318

[4] Dorsey J, Searles S, Ciraula M, et al. An integrated quad-core opteron processor//Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, USA, 2007: 102-103

[5] Howard J, Dighe S, Vangal S, et al. A 48-Core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. IEEE Journal of Solid-State Circuits, 2011, 46(1): 173-183

[6] Humenay E, Tarjan D, Skadron K. Impact of process variations on multicore performance symmetry//Proceedings of the ACM/IEEE Design, Automation and Test in Europe (DATE). Nice, France, 2007: 1-6

[7] Bowman K A, Alameldeen A R, Srinivasan S T, et al. Impact of die-to-die and within-die parameter variations on the clock frequency and throughput of multi-core processors. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(12): 1679-1690

[8] Yan G, Li Y, Han Y, et al. AgileRegulator: A hybrid voltage regulator scheme redeeming dark silicon for power efficiency in a multicore architecture//Proceedings of the IEEE International Symposium on High Performance Computer Architecture(HPCA). New Orleans, USA, 2012: 1-12

[9] Hung W, Link G M, Xie Y, et al. Interconnect and thermal-aware floorplanning for 3D microprocessors//Proceedings of the IEEE International Symposium on Quality Electronic Design(ISQED). San Jose, USA, 2006: 98-104

[10] Ogras U Y, Marculescu R, Marculescu D. Design and management of voltage-frequency island partitioned networks-on-chip. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(3): 330-341

[11] Hu J, Marculescu R. Communication and task scheduling of application-specific networks-on-chip. Computers and Digital Techniques, 2005, 152(5): 643-651

[12] Varatkar G, Marculescu R. Communication-aware task scheduling and voltage selection for total systems energy minimization//Proceedings of the ACM/IEEE International Conference on Computer-Aided Design(ICCAD). San Jose, USA, 2003: 510-517

[13] Jang W, Duo D, Pan D Z. A voltage-frequency island aware energy optimization framework for networks-on-chip//Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD). San Jose, USA, 2008: 264-269

[14] Zhang Y, Hu S, Danny Z. Task scheduling and voltage selection for energy minimization//Proceedings of the ACM/IEEE Design Automation Conference(DAC). New Orleans, USA, 2002: 183-188

[15] Cheng Y, Zhang L, Han Y, et al. Thermal-constrained task allocation for interconnect energy reduction in 3-D homogeneous MPSoCs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2013, 21(2): 239-249

[16] Goplen B, Sapatnekar S. Thermal via placement in 3D ICs//Proceedings of the IEEE International Symposium on Physical Design(ISPD). San Francisco, USA, 2005: 167-174

[17] Wong E, Lim S K. 3D floorplanning with thermal vias//Proceedings of the ACM/IEEE Design, Automation and Test in Europe(DATE). Munich, Germany, 2006: 1-6

[18] Bakir M S, King C, Sekar D, et al. 3D heterogeneous integrated systems: liquid cooling, power delivery, and implementation//Proceedings of the IEEE Custom Integrated Circuits Conference (CICC). San Jose, USA, 2008: 663-670

[19] Coskun A K, Ayala J L, Atienza D, et al. Dynamic thermal management in 3D multicore architectures//Proceedings of the ACM/IEEE Design, Automation and Test in Europe (DATE). Nice, France, 2009: 1410-1415

[20] Zhou X, Yang J, Xu Y, et al. Thermal-aware task scheduling for 3D multicore processors. IEEE Transactions on Parallel Distribution System, 2010, 21(1): 60-71

[21] Zhu C, Gu Z, Shang L, et al. Three-dimensional chip-multiprocessor run-time thermal management. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(8): 1479-1492

[22] Clermidy F, Darve F, Dutoit D, et al. 3D embedded multi-core: Some perspectives//Proceedings of the ACM/IEEE Design, Automation and Test in Europe(DATE). Grenoble, France, 2011: 1327-1332

[23] Kang K, Kim J, Yoo S, et al. Runtime power management of 3-D multi-core architectures under peak power and temperature constraints. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2010, 30(6): 905-918

[24] Link G M, Vijaykrishnan N. Thermal trends in emerging technologies//Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED). San Jose, USA, 2006: 8-632

[25] Smruti R S, Brian G, Radu T, et al. VARIUS: A model of process variation and resulting timing errors for microarchitects. IEEE Transactions on Semiconductor Manufacturing, 2008, 21(1): 3-13

[26] Shane B, Bruce E, John A, et al. TILE64 processor: A 64-core SoC with mesh interconnect//Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, USA, 2008: 88-598

[27] Zhao W, Cao Y. New generation of predictive technology model for sub-45 nm early design explorations//Proceedings of the IEEE International Symposium on Quality Electronic Design(ISQED). San Jose, USA, 2006: 590-595

JIN Song, born in 1977, Ph. D., associate professor. His research interests include VLSI design and testing, computer architecture, with emphasis on energy-efficient systems, design for reliability and variation tolerance.

HAN Yin-He, born in 1980, Ph. D., professor. His research interests include fault tolerant chip design, computer architecture and VLSI testing.

WANG Yu, born in 1979, Ph. D., lecturer. His research interests include electrical information processing, pattern recognition.

Background

Three dimensional (3-D) multi-core provides a promising solution for the design of high-end system-on-chips (SoCs). 3-D integration not only can overcome global interconnect latency and power bottlenecks in 2-D chips. The multi-core structure can increase system throughput and being more robust to various variabilities. For embedded SoCs, energy efficiency has become a primary design concern due to power supply commonly relies on the battery. Recently, concept of voltage-frequency island (VFI) was introduced into 2-D multi-core designs to optimize system energy. With VFI design, the cores on the chip are partitioned into different islands. Each island can operate at its own voltage and frequency. Combining with task scheduling, such design paradigm helps to implement fine-grained energy management meanwhile still meeting deadline constraints of the executed tasks.

However, energy optimization targeting VFI-based 3-D chip confronts some new challenges. The rigorous thermal issues from 3-D integration exacerbate hot spot and raise the chip's temperature. On the other hand, heterogeneous workloads executed cause power variation, resulting in the thermal gradient across the chip. High temperature and thermal gradient not only degrade system performance and reliability, but also offset the effort of system energy optimization. Therefore, we still need to carefully balance energy optimization and thermal balance when we target at 3-D multi-core SoCs.

At present, there have been some studies on energy optimization of multi-core SoCs based on VFI design paradigm. However, these studies more or less have some drawbacks needed to be overcome. For example, VFI-based energy optimization on 2-D multi-core chip generally ignores the thermal issue, hence being unfit to be applied to 3-D multi-core chip. On the other hand, some optimization schemes targeting 3-D chip commonly focus on reducing thermal or temperature of the chip while neglecting the energy optimization. This paper aims to overcome above mentioned problems. We proposed a post-silicon optimization framework targeting VFI-based 3-D multi-core SoCs to minimize system energy meanwhile still meeting task deadline and thermal constraints. Besides energy-aware task scheduling and voltage/frequency assigning, we also conduct task migrating to balance the powers across the core stacks for thermal optimization.

This work is mainly supported by the National Natural Science Foundation of China (No. 61204027), named as "On Improving System Performance of 3-D Multi-Core in the Presence of Parameter Variability". This project aims to improve performance and energy efficiency of 3-D multi-core chip under parameter variability. This paper focuses on "Energy Optimization of 3-D Multi-Core" part of the project. Our group has been working in the area of energy efficiency system in recent years and has published many related papers.