

基于电热耦合效应的高性能处理器单元降温降耗算法

张昌明, 骆祖莹, 吴文川, 邢霄雄, 郑 献
(北京师范大学 信息科学与技术学院, 北京 100875)

摘 要: 基于已有的 TALK 算法, 提出了基于电热耦合效应的空闲时间均匀分配 (TCED) 算法. 实验表明: TCED 算法十分简便, 合理地利用空闲时间来降温, 解决了 TALK 算法在较低负载和极高负载情况下的不足. 因此 TCED 算法适用于高性能处理器单元各种工作负载的降温降耗.

关键词: 高性能处理器; 静态功耗; 电热耦合效应; 功耗优化算法

中图分类号: TP316.2

文献标识码: A

文章编号: 1000-7180(2010)05-0162-05

Algorithm of Lowing Temperature and Power Based on the Electric and Thermal - Coupling Effect for High - Performance Processor Unit

ZHANG Chang-ming, LUO Zu-ying,
WU Wen-chuan, XING Xiao-xiong, ZHENG Xian

(College of Information Science and Technology, Beijing Normal University, Beijing 100875, China)

Abstract: Based on the existing TALK algorithm, the paper put forward another more effective algorithm - free time evenly distributed technique based on the electric and thermal coupling effect (TCED), which is more convenient and make more rational use of free time to lower temperature and meet the shortfall of TALK when applying to low load and very high load. In sum, TCED algorithm is more universally applied to decrease the temperature and energy consumption for high - performance processor unit with various sizes of workload.

Key words: high - performance processor; static power consumption; electric and thermal - coupling effect; power optimization algorithm

1 引言

随着集成电路工艺进入纳米时代, 芯片的功耗和功率密度迅速增加. 而工艺的提高使得由漏电流产生的功耗即静态功耗所占比重越来越大, 已成为高性能芯片功耗的主要来源^[1]. 同时芯片中还存在静态功耗随内核温度增加而增加的电热耦合效应, 所以针对在运行中降低静态功耗的优化研究具有重要的理论意义和应用价值^[2]. 由于晶体管的漏电流与温度呈指数相关^[3], 如果一个系统没有正确地设计, 芯片的温度和漏电流功耗可能形成正反馈环路,

并最终导致热失控. 典型的 TALK 算法^[4]就是为了防止这种情况的发生而设计的.

然而 TALK 算法在低负载和极高负载情况下露出一些弊端. 鉴于 TALK 算法的弊端, 文中提出了一种改进算法——TCED 算法, 将工作时间与睡眠时间均匀分配, 就有效地克服了 TALK 算法在两种负载情况下的不足^[5].

2 研究基础

2.1 温度模型

芯片的温度变化情况为: 在执行态时, 系统温度

收稿日期: 2009-03-23; 修回日期: 2009-07-25

基金项目: 国家自然科学基金项目(60876025); 国家“八六三”计划项目(2009AA01Z126)

会升高到一个温度比较高的平衡状态 K_1 ; 在睡眠态, 系统温度会降低到一个温度较低的平衡状态 K_2 , 正常情况下 K_2 应和环境温度相等. 设起始温度为 T_{cur} , 那么在执行态和睡眠态经过时间 t 后的温度分别为如下所示:

$$T_{\text{rise}}(t) = K_1 - (K_1 - T_{\text{cur}}) \cdot e^{-\frac{t}{\tau}} \quad (1)$$

$$T_{\text{full}}(t) = K_2 + (T_{\text{cur}} - K_2) \cdot e^{-\frac{t}{\tau}} \quad (2)$$

式中 τ 是时间常数, 它由芯片晶体管的热阻 R_{th} 和热容 C_{th} 决定.

2.2 功耗模型

由于静态功耗特别突出, 而且受温度影响, 所以文中只讨论静态功耗. 静态功率可按如下计算:

$$P_{\text{leakage}} = N_g \cdot I_{\text{leakage}} \cdot V_{\text{dd}} \quad (3)$$

N_g 是晶体管的数目, I_{leakage} 是当前的漏电流值, 它是一个随温度变化的量, 在 65nm 工艺中, 可以按如下关系确定:

$$I_{\text{leakage}} = A \cdot T^2 \cdot e^{\frac{\alpha \cdot V_{\text{dd}} + \beta}{T}} + B \cdot e^{\gamma \cdot V_{\text{dd}} + \delta} \quad (4)$$

式中, $A, B, \alpha, \beta, \gamma, \delta$ 均为常数, T 是当前的温度, 文献[6]中有详细说明.

文中用 D 表示一项任务的允许时间, 用 W 表示正常执行完所需的时间, 也就是工作负载. 定义函数 $x(t) \in \{0, 1\}$, 在时刻 t , 如果处于执行态, 则 $x(t) = 1$, 若处于睡眠态, 则 $x(t) = 0$. 所以总能量可以按下式计算:

$$\int_0^D P_{\text{active}} \cdot x(t) + P_{\text{sleep}} \cdot (1 - x(t)) dt \quad (5)$$

式中, P_{active} 是执行态的功率, P_{sleep} 是睡眠态的功率, P_{sleep} 通常是一个常数, 由于睡眠的总时间应为 $(D - W)$, 所以睡眠态的总能量也应为一个常数 $P_{\text{sleep}} \cdot (D - W)$. 问题就在于在满足(6)式条件下, 使执行期间消耗的能量最少.

$$\int_0^D x(t) dt \geq W \quad (6)$$

2.3 任务调度过程分析

为了便于处理, 将时间 D 分成 N 个时间片, 每片长度为 Δt , 而在一个时间片内所处的状态是一样的. 于是在时间 D 内就有一系列时间节点, $0 = t_0 < t_1 < t_2 < \dots < t_{N-1} < t_N = D$, 其中 $[t_i, t_{i+1}] = \Delta t$. 用 $x[i]$ 代表 $[t_i, t_{i+1}]$ 期间的工作状态, 同样, 1 为执行态, 0 为睡眠态. 另外, 由于每次唤醒都还需一定能量, 所以用 ω_i 表示时刻 t_i 是否是被唤醒, 即如果 $x[i-1] = 0, x[i] = 1$, 则 $\omega_i = 1, t_i$ 时刻被

唤醒, 其他情况 $\omega_i = 0$. 所以总能量为:

$$\sum_{i=0}^{N-1} (x[i] \cdot \int_{t_i}^{t_{i+1}} P_{\text{active}}(t) dt + (1 - x[i]) \cdot \int_{t_i}^{t_{i+1}} P_{\text{sleep}}(t) dt + \omega_i E_{\text{wakeup}}) \quad (7)$$

满足的条件为:

$$\sum_{i=0}^{N-1} x[i] \cdot \Delta t \geq W \quad (8)$$

由于睡眠态功率的固定性, 若执行任务的总时间刚好为 W , 总能量可以简化为:

$$\sum_{i=0}^{N-1} (x[i] \cdot \int_{t_i}^{t_{i+1}} P_{\text{active}}(t) dt + \omega_i E_{\text{wakeup}}) + P_{\text{sleep}} \cdot (D - W) \quad (9)$$

因此算法的主要目标是如何确定好 $x[i]$ 的值, 使式(9)得到最小值.

2.4 TALK 算法及其弊端

TALK 算法比较工作负载和空闲时间的比 $\eta = \frac{Wr}{Dr - Wr}$, 与睡眠温度下降速率和执行温度上升速

率比 $\theta = \left| \frac{dT_{\text{fall}}/dt}{dT_{\text{rise}}/dt} \right| = \frac{T_{\text{cur}} - K_2}{K_1 - T_{\text{cur}}}$, 如果前者大于后者说明处理器任务较重, 睡眠降温效率不高, 处理器就执行任务, 而如果前者小于后者说明任务较轻, 睡眠降温效率高, 就将处理器就置于睡眠状态以实现降温. TALK 算法在很多情况下实现了功耗的最优化, 然而, 它也不是一种完备的算法, 静态能量也有一部分浪费, 具体表现为:

(1) 当工作负载很小的情况下, 如当 $W \leq 0.4D$ 时, 处理机执行任务主要是在前一段时间, 而后一段时间就完全处于睡眠状态, 睡眠时间是降温所用的, 而当工作已经执行完毕再来降温是没有任何意义的, 这部分空闲时间可以在前面利用起来.

(2) 当负载较大时, 如 $W \geq 0.8D$ 时, 情况刚好与情况(1)相反, 处理机执行任务后面时间段的压力非常大, 而前一阶段相对较轻松, 这样, 在后阶段温度一直处于上升状态, 漏电流所造成的功率不断增大.

3 基于电热耦合效应的空闲时间均匀分配算法 TCED

针对 TALK 算法的不足, 文中提出了一种空闲时间片均匀分配的方法来执行任务. 算法执行过程为:

(1) 得到时间片数 $N = D/\Delta t$, 初始执行时间片数 $n = 0$, 空闲(睡眠)时间片数 $m = 0$.

(2) 比较负载量 W 与允许时间 D , 分三种情况: ① $W > D$, 不能完成任务, 停止执行; ② $W = D$, 刚好在没有空闲时间的情况下完成任务, $x[i] = 1$, for $0 \leq i < N$ ($x[i]$ 表示第 i 个时间片); ③ $W < D$, 转入第(3)步.

(3) 得到空闲时间与负载量的比, $q = \frac{D - W}{W}$, 如果 $q > 1$, 说明负载较小, 将在空闲时间片中插入工作时间片, 转入第(4)步; 如果 $q \leq 1$, 说明负载较大, 将在工作时间片中插入空闲时间片, 转入第(5)步.

(4) $x[0] = 1, n = 1, j = 1$, 转入第(6)步.

(5) $x[i] = 1$, for $0 \leq i \leq \frac{W/dt}{(D - W)/dt + 1}$, $n = \frac{W/dt}{(D - W)/dt + 1}, j = \frac{W/dt}{(D - W)/dt + 1}$, 转入第(6)步.

(6) 如果 $\frac{m}{n} \geq q, x[i] = 1, n++$, 否则 $x[i] = 0, m++$, for $j \leq i < N$.

通过上述方法, 便可以实现空闲时间片的均匀分配, 得出 $x[i]$ 后, 便可以很容易计算出消耗的能量和温度变化情况. 这种空闲时间均匀分配的方法便可避免 TALK 算法执行任务与睡眠搭配不合理的弊端. 这样分配后, 执行任务的平均温度会得到降低, 如图 1、图 2 所示. 就是在 $D = 10s, W = 2s$ 和 $W = 9s$ 时按 TALK 算法和 TCED 算法的温度变化情况.

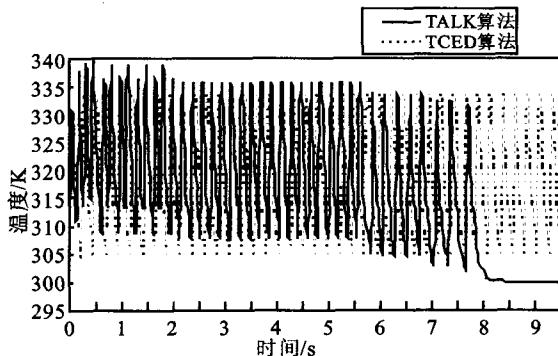


图 1 20%负载时 TCED 算法与 TALK 算法温度变化比较

从图 1 可以得知, 在工作负载较小时, TCED 算法很好地利用了空闲时间来睡眠降温, 使每次工作了之后温度降到足够低后才进行下一次工作. TALK 算法后阶段虽然温度很低, 但已经没有工作了. 同时 TCED 算法也同 TALK 算法一样能够保证任务完成. 这样, TCED 算法在执行任务期间的平均

温度就要比 TALK 算法低, 所消耗的静态能量也较少, 进一步节省了功耗.

图 2 可以看到, 在工作负载较大时, TALK 算法在前期降温频率比较高, 它的平均温度比 TCED 算法要低, 但是在后半阶段, TALK 算法一点降温的空间都没有, 一直持续在最高温度处工作, 而 TCED 算法还能够继续有部分时间处于睡眠状态, 可以实现降温, 避免了系统在高温持续工作的现象. 从总体上看, TCED 算法相比 TALK 算法使整个过程的平均温度得到降低.

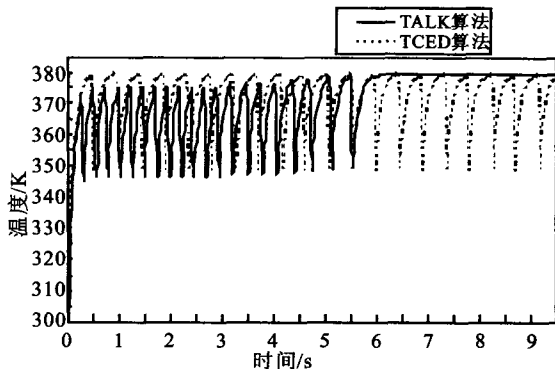


图 2 90%负载时 TCED 算法与 TALK 算法温度变化比较

这两个现象说明, TCED 算法能够排除 TALK 算法执行任务时片面性的缺点, 它虽然简单却在每个时刻都从全局出发判断是执行任务还是睡眠降温, TCED 算法使整个允许时间 D 得到合理的协调利用. 从式(4)可知, 平均温度得到降低, 漏电流也就得到减小, 静态功耗也就会降低.

4 实验仿真结果及分析

文中分两步模拟比较 TCED 和 TALK 算法, (1) 在 $D = 10s$, 时间片长度 Δt 为 $50ms$ 时, 不同工作负载 W 的情况下, 比较两种算法所消耗的静态能量; (2) 基于一些实际的作业进行模拟, 比较 TCED 算法与 TALK 算法.

4.1 D 为 $10s$ 时, 不同 W 情况下两种算法的耗能比较

从图 3、图 4 中可知, 当 $D \leq 0.4W$ 时, 平均可节省 5% 的能量, 当 $D \geq 0.8W$ 时, 平均可节省 1.5% 的能量, 当 $0.4D < W < 0.8D$ 时, TALK 算法和 TCED 算法的功耗几乎相等, 在这个阶段 TALK 算法的睡眠和执行的搭配比较合理, 所以 TCED 算法比起 TALK 算法就表现不出优越性.

4.2 标准作业仿真

对不同时间片大小的一些标准作业进行仿真,

比较 TALK 算法与 TCED 算法的静态能量消耗情况,如表 1 所示.

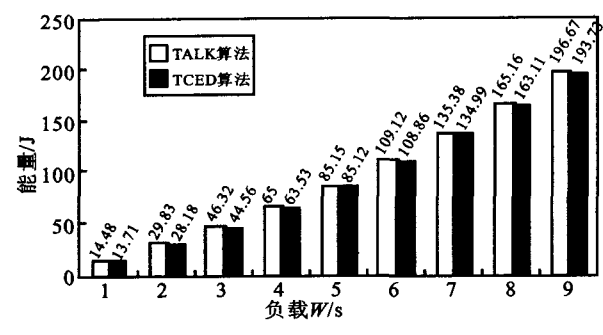


图 3 TCED 算法与 TALK 算法耗能比较

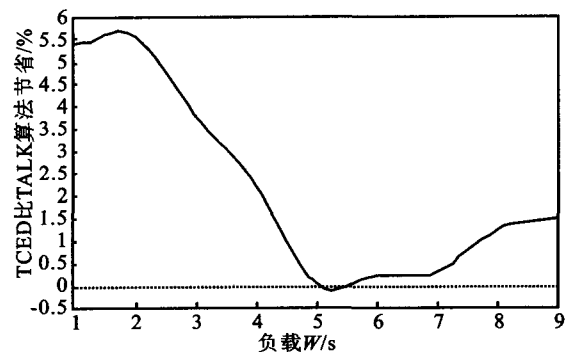


图 4 TCED 算法比 TALK 算法能量节省百分比与负载大小关系

表 1 TCED 算法与 TALK 算法对标准作业耗能仿真比较

作业	允许时间 D/s	工作负载 W/s	$\Delta t = 100\text{ms}$			$\Delta t = 50\text{ms}$			$\Delta t = 20\text{ms}$		
			TALK	TCED	节省	TALK	TCED	节省	TALK	TCED	节省
MPEG4	60	50	1077.5	1058.0	1.8%	1058.5	1046.4	1.1%	1046.5	1038.0	0.8%
CH2	1	0.3	4.89	4.56	6.8%	4.59	4.41	3.9%	4.44	4.33	2.5%
CO	1	0.15	2.34	2.22	5.2%	2.28	2.14	6.1%	2.12	2.02	4.5%
Airflow	2	0.2	3.14	2.95	6.1%	2.89	2.75	4.8%	2.73	2.65	2.8%
ADSL1	0.576	0.285	5.00	4.97	0.6%	4.89	4.87	0.4%	4.48	4.51	-0.5%
ADSL2	2.048	0.864	15.19	14.71	3.2%	13.87	13.46	2.9%	13.74	13.70	0.3%
Bmk1	1	0.4	6.73	6.52	3.1%	6.46	6.30	2.4%	6.22	6.14	1.3%
Bmk2	1	0.5	8.48	8.48	0.0%	8.33	8.29	0.4%	8.21	8.24	-0.3%
Bmk3	1	0.6	10.55	10.53	0.2%	10.41	10.40	0.1%	10.37	10.40	-0.3%
Bmk4	1	0.7	13.01	13.00	0.1%	12.81	12.81	0.0%	12.78	12.81	-0.2%
Bmk5	1	0.8	15.78	15.51	1.7%	15.52	15.23	1.9%	15.45	15.32	0.8%

在对这些标准作业进行仿真的过程中,时间片的长度参考文献[4]分成了三种情况:100ms,50ms,20ms.即允许时间 D 被分成了这样一些小单元, K_1 设定为 300K, K_2 是根据不同作业的实际情况在 363K~388K 之间,每次唤醒需要消耗的能量为 483 μ J,而在睡眠状态的功率为 50 μ W,晶体管数目 N_g 为 4.5 亿个,参数的设定标准见文献[7].在这些作业中,第一个是依据流媒体中的 MPEG-4 编码技术,第二到第四个作业参照相关文献设定,第五、六个作业是 ADSL 标准的初始化序列中得来的,其余五个作业是为了代表不同的大小负载而专成设置以进行模拟仿真.通过这些标准作业的仿真,进一步基于事实说明了 TCED 算法的高效性.

从表 1 中的数据发现,TCED 算法相比 TALK 算法在一定负载的情况下减小了静态能量的消耗,这也主要是得益于 TCED 算法能够更加恰当地利用空闲时间来降温.还可以发现,随着 Δt 的减小,

功耗的节省量会减少,这是因为时间片长度减小后,时间片数目增加,TCED 算法的唤醒次数也同比例地增大,而 TALK 算法的唤醒次数相对而言却增加得比较慢,每次唤醒是需要一定能量的.所以,TCED 算法更适合在时间片稍大的时候($\Delta t > 20\text{ms}$)应用.

5 结束语

考虑温度对漏电流影响的电热耦合效应,为了节省静态功耗,文中基于 TALK 算法,提出了 TCED 算法,这种算法具有简单的时间复杂度,克服了 TALK 算法在较低负载和极高负载时出现的不足,有效利用空闲时间来进行降温,使系统工作的平均温度得到降低.在 $D \leq 0.4\text{W}$ 时,TCED 算法相比 TALK 算法平均节省 5% 的能量,在 $D \geq 0.8\text{W}$ 时,平均可节省 1.5% 的能量,而在其它情况 TCED 算法也不会比 TALK 算法消耗更多的静态能量,所

以,可以用 TCED 算法取代 TALK 算法.另外,TCED 算法的不足是随着时间片长度 Δt 减小到一定程度时,唤醒次数会大量增加,能量消耗反而会增加,所以 TCED 算法在 $\Delta t \leq 20\text{ms}$ 时通常是不适用的,有关这种情况的算法优化还有待研究.

参考文献:

- [1] 徐懿,李丽,高明伦,等. 纳米集成电路静态功耗机理及低功耗设计技术[J]. 微电子学与计算机,2007,24(5): 184 - 188.
- [2] 张孝坤,王继安,徐赏林,等. 一种可用于模块化设计的热关断电路[J]. 微电子学与计算机,2007,24(6):130 - 136.
- [3] Berkeley BSIM3 device models. URL[EB/OL]. [2009 - 06 - 10]. <http://www-device.eecs.berkeley.edu/bsim3/>.
- [4] Lin Yuan, Sean Leventhal, Gang Qu. Temperature - aware leakage minimization technique for real - time systems[J]. Computer - Aided Design,2006(6):761 - 764.
- [5] Skadron K, Abdelzaher T, Stan M R. Control - theoretic techniques and thermal - RC modeling for accurate and localized dynamic thermal management[J]. High - Performance Computer Architecture,2002(2):17 - 28.
- [6] He L, Liao W, Stan M R. System level leakage reduction considering the interdependence of temperature and leakage[J]. Design Automation Conference,2004(41):12 - 17.
- [7] Skadron K. Temperature - aware microarchitecture: modeling and implementation[J]. ACM Trans on Architecture and Code Optimization,2004, 1(1): 94 - 125.

作者简介:

张昌明 男,(1987 -).研究方向为低功耗设计.

骆祖莹 男,(1968 -),博士,副教授.研究方向为低功耗 IC 设计和 IC 物理设计.

(上接第 161 页)

- [4] Hongsheng Li, Yangquan Chen. A fractional order proportional and derivative (FOPD) controller tuning algorithm [C]// Chinese Control and Decision Conference. China: Yantai, 2008:4059 - 4063.
- [5] Serdar E Hamamci. Stabilization using fractional - order PI and PID controllers[J]. Nonlinear Dyn, 2008(51):329 - 343.
- [6] Dingyu Xue, Yangquan Chen. A comparative introduction of four fractional order controller[C]// Congress on Intelligent Control and Automation. China, Shanghai,2002.
- [7] 高尚,杨静宇. 群智能算法及其应用[M]. 北京:中国水利水电出版社,2007.
- [8] Passio K M. Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors[J]. Journal of Optimization Theory and Applications, 2002;115(3):603 - 628.
- [9] 周开利,康耀红. 神经网络模型及其 MATLAB 仿真程序设计[M]. 北京:清华大学出版社,2007,11.

作者简介:

胡海波 女,(1985 -),硕士研究生.研究方向为控制理论与控制工程.

黄友锐 男,(1971 -),教授,硕士生导师.研究方向为计算机控制和智能信息处理.