针对分布式系统的快速能耗估计方法及应用

粟雅娟, 魏少军
(清华大学微电子所, 北京 100084)

摘 要: 本文针对包含可变电压处理单元的实时分布式系统提出了快速能耗估计算法FAEE, 并在此基础上改进了低能耗分配方法. 和现有方法相比可获得几乎相同优化结果, 而CPU时间降低了约2个数量级.

关键词: 低能耗; 变电压; 快速估计; CPU时间; FAEE
中图分类号: TN402 文献标识码: A 文章编号: 0372-2112 (2005) 09-1706-04

A Fast Energy-Aware Design Technique for Distributed Real-Time Systems

SU Ya-juan, WEI Shao-jun
(*Institute of Microelectronics, Tsinghua University Beijing, P. R. 100084, China*)

Abstract: This paper presents a fast and efficient energy evaluation technique (FAEE) at system level according to frame-based real-time distributed system containing voltage variable processing element. Based on FAEE, a modified task assignment method is proposed. Compared with traditional method, CPU time of the latter is about 2 order of magnitudes of that of FAEE's method while energy saving of both techniques is almost the same.

Key words: low energy; variable voltage; fast estimation; CPU time; FAEE

1 引言

随着工艺加工水平的不断提高和系统规模的日益扩大, 出于节能和可靠性的要求, 能耗已经成为电子系统设计的一个主要考虑因素. 研究表明[1], 在设计流程中尽早引入低能耗措施, 可获得更优的能耗优化结果. 动态电压调度(DVS)是一种系统级的低能耗技术, 它针对工作电压可变处理器, 利用处理器的空闲时间, 有选择地降低其上运行任务的执行速度, 从而有效节省能耗.

软硬件协同设计是目前系统级设计的一种主流设计方法学[2], 它以对系统功能的抽象描述为输入, 经过系统架构生成、任务分配、任务调度和评价等四个步骤, 输出满足约束的系统硬件架构和功能单元(任务)在硬件及时间上的映射. 将DVS结合软硬件协同设计技术被证明是有效降低能耗的方法[3].

DVS降低能耗的来源是处理器的空闲时间, 而空闲时间的分布和系统任务分配及调度的结果密切相关, 尤其是对于多处理单元的分布式复杂系统. 因此, 理想的基于DVS技术的系统级低能耗设计应该结合软硬件协同设计技术, 从系统的抽象描述出发, 在系统架构生成、任务分配和任务调度等各个步骤均考虑系统能耗, 均衡各个模块间能耗和性能之间的折中.

在分配和调度中引入DVS的研究尚处于起步阶段, Marcus[3]首次将DVS技术引入系统设计流程以降低能耗, 优点在于能够找到全局最优解, 但由于所采用的DVS算法复杂度的限制, 对分配解的评价过程太长. Zhang[4]将电压调度建模为二次线性规划问题, 并提出了一种同时进行任务分配和调度的简单构造方法. 通过对任务赋优先级选择适合的处理单元, 避免在关键路径上放置过多的任务. 由于在分配和调度时, 仅仅考虑了在空闲时间上对任务达到均衡, 没有考虑任务在不同处理器上能耗特性的差异, 该方法不适用于处理器能耗差异较大的结构.

Frame-based系统[5]指所有任务的周期都相同的系统, 该类系统在语音和多媒体编码和传输等领域有广泛应用. 本文的方法主要针对frame-based系统提出能耗的快速估计方法FAEE, 并结合软硬件设计流程优化任务分配, 在保证优化能耗的同时, 极大降低了计算复杂度.

论文组织如下: 第二节给出系统结构和能耗模型, 第三节描述快速能耗估计方法FAEE和基于FAEE的低能耗分配算法, 第四节通过实验数据比较, 说明本文方法的有效性. 第五节总结论文工作.

2 模型

2.1 系统架构模型

假设系统架构事先确定. 系统主要由处理单元(PEs): 包括通用处理器、ASIP、ASIC和FPGA等; 存储单元: 包括DRAM、

收稿日期: 2004-04-09; 修回日期: 2005-01-18

ROM、SRAM 等和通信单元组成. 处理单元中包含电压可变的处理单元. 连接处理单元的通信单元是点对点的通信方式, 可以根据所连接的处理单元调整频率. 假设每个处理单元间有且只有一条通信连接.

系统应用描述为周期性任务集合. 周期 T 表示任务集合两次连续执行的时间间隔. 任务 i 的参数分为两类: 一类是由系统应用的约束所决定的参数, 包括: a_i: 任务到达时间, 规定了每个周期内任务 i 的被释放的时间, 默认值为 0; dl_i: 截止时间, 规定了每个周期内任务 i 的最迟执行结束时间, 由于系统为 frame based 系统, 所有任务的 dl_i 等于周期值 T. 另一类是和任务所在的处理单元 j 有关的参数, 包括: 执行时间 l_i 和功耗 P_i, l_i 和 P_i 都是 i 在 j 上以最大工作电压运行时得到的值.

2.2 能耗模型

任务 i 在处理单元 j 上的能耗 E_i, 以及执行时间 l_i 表示如下[4]:

$$l_i = k^* N_i^* \frac{v_i}{(v_i - v_T)^2} \tag{1}$$

$$E_i = N_i^* C_i^* v_i^2 \tag{2}$$

其中 k 是和 j 相关的常数, v_T 是 j 的阈值电压, α 是和工艺相关在 1.2~2 之间的常数(通常假定为 2), C_i 是每个周期的有效开关电容, N_i 是完成任务 i 所需要的时钟周期数.

$$E_{all} = \sum_{u=1}^{n} N_i^* C_i^* v_i^2 \tag{3}$$

E_{all} 为系统总能耗. DVS 的目标是在满足时间约束的条件下使得 E_{all} 最小. 随着工作电压的降低, E_i 降低而 l_i 升高, 说明任务的能耗和执行时间存在折衷. 在实际系统中, 处理单元的负载通常小于 1, 存在空闲时间, 可以利用 DVS 技术在保证时间约束的前提下, 有选择地降低某些任务工作电压来降低能耗.

为了描述能耗降低和执行时间变长之间的关系, 定义能耗执行时间微分系数 $\eta_i(v_i)$ 如下:

$$\eta_i(v_i) = \left| \frac{dE_i(v)}{dl_i(v)} \right|_{v=v_i} = \frac{2C_i}{k} * \frac{v_i^* (v_i - v_T)^3}{v_i + v_T} \tag{4}$$

$\eta_i(v_i)$ 表示任务当工作电压为 v_i 时能耗随着执行时间改变而变化的快慢程度, 是 v_i 的单调递增函数. 对于在固定电压处理单元上执行的任务, 定义其 $\eta(v_i)$ 恒为 0. DVS 方法的核心是在保证任务时间约束的前提下, 如何将处理单元上的空闲时间分配给其上的任务以使降低的能耗最大. 假设处理单元上的空闲时间被分成若干个无穷小的等份, 则将每份空闲时间分配给当 $\eta(v_i)$ 最大的任务能获得最大的能耗节约.

在我们前面的工作中[9], 有如下定理:

定理 1: 假设一个有硬时间约束系统在一个电压可变的处理器上执行, 最大电压时 $\eta(v)$ 的顺序为 $\eta_1(v_1) \geqslant \eta_2(v_2) \geqslant \ldots \geqslant \eta_n(v_n)$. 采用 DVS 技术降低能耗, 最小能耗时 $\eta(v)$ 保持最大电压时的顺序, $\eta_i(v_i) > \eta_{i+1}(v_{i+1})$ 的不等号当且仅当 τ_i 没有空闲时间时才出现.

3 算法和设计流程

3.1 快速能耗估计算法 FAEE

由式(1)、(2)和(4)可推出:

$$\eta_i = 2 \frac{E_i}{l_i} * \frac{v_i - v_T}{v_i + v_T} \tag{5}$$

令 $w_{1i} = 0.5 * \frac{v_i + v_T}{v_i - v_T}$, $w_{2i} = \frac{(v_i - v_T)^2}{v_i^2}$

则

$$\frac{E_i}{l_i} = w_{1i} * \eta_i, \quad l_i = w_{2i} \frac{k^* N_i}{v_i} \tag{6}$$

有 $E_i^* l_i^2 = C_i^* N_i^* w_{2i}^2 * k^2 * N_i^2$

结合式(6), 令 $\rho_i = w_{2i}^2 w_{1i}$

$$l_i = \left(\frac{C_i N_i w_{2i}^2 k^2 N_i^2 w_{1i}}{\eta_i} \right)^{\frac{1}{3}} = \rho_i^{\frac{1}{3}} \left(\frac{C_i k^2 N_i^3}{\eta_i} \right)^{\frac{1}{3}} \tag{7}$$

对于 frame based 单处理器系统, 如果在一个周期内, 处理器的空闲时间足够大以使得所有任务的 η_i 值相等, 将所有任务执行时间相加得到:

$$\sum_{i=1}^{n} l_i = T \Rightarrow \sum_{i=1}^{n} \rho^{\frac{1}{3}} \left(\frac{C_i k^2 N_i^3}{\eta_i} \right)^{\frac{1}{3}} = T \tag{8}$$

当 $v_i > 4v_T$ 时, 可近似 $w_{1i} \approx 0.5$, $\rho_i \approx 0.5$

$$\eta_i = \left[\frac{\rho^{\frac{1}{3}} (C_i k^2 N_i^3)^{\frac{1}{3}}}{T} \right]^3 \tag{9}$$

可见, 在处理器空闲时间足够长的条件下, 各个任务工作电压降低后的执行时间以及 η_i 值很容易用式(7)和式(9)的近似式求得. 任务执行时间确定后能耗随之确定.

由于任务能耗分布不均匀, 在实际情况下当处理器的空闲时间全部分配给任务后, 不能保证所有任务的 η_i 为统一值. 根据定理 1, 我们用构造方法来求使得系统能耗最小的任务执行时间和 η_i 并据此计算能耗. 算法伪代码如下:

```
Input: task set{τ1, τ2 ……τn} with defined timing and power parame-
       ter under maximum voltage;
Output: minimal energy consumption Emin;
    1. Order the task set in decreasing ηi value;
    2. index= 2; break= 0;
    3. while( index< n and break= = 0){
    4.  T' = period value of system;
    5. T= T' - Σ_{i= index+1}^{n}
    6. compute optimal ηi for {τ1τ2 …… τindex}  by formula (9)
    7.  if ( ηi > ηindex+1){
    8.  index= index+ 1; }
    9.  else {break= 1;}
    10. }
    11. compute Emin by ηi and li value;
```

FAEE 估计算法的复杂度为 $O(n)$.

3.2 基于 FAEE 的低能耗分配

FAEE 是针对单处理器系统的估计方法, 而在实际系统中由于负载要求和提高并行度的需要, 经常采用多处理器系统. 由于任务之间没有依赖关系, 可以将每个处理器上的任务视为一条任务链, 整个系统由多条不相关的任务链构成. 多处理器系统的最终优化的能耗值等于每条任务链用 FAEE 得到的 Emin 的合值.

如前文所述, 系统 DVS 后能耗的减小程度是和分配以及

调度有关的,由于本文讨论的是 frame based 系统,一旦分配确定后,同一处理器上所有任务的可用空闲时间是相等的,即调度的策略对 DVS 效果没有影响,仅仅需要考虑分配对能耗的影响. 在软硬件协同设计中,分配问题是 NP hard 问题,很难通过启发式算法解决,因此采用了合适于全局搜索的遗传算法来搜索最优的低能耗分配方案.

本文选择 Schmitz 提出的一种适用于分布式系统的 DVS 方法 PVDVS[3] 作为本文方法的比较基础,它是现有文献中提到的效果最佳的 DVS 构造算法. PVDVS 的输入是系统架构,任务分配和调度,任务的时间和功耗参数以及系统的时间约束,输出是使得系统能耗最小的电压调度. PVDVS 的算法复杂度依赖于每次迭代所扩展的空闲时间的步长 Δt 的选取,假设任务数为 n, $k=\min($所有任务的空闲时间$)/\Delta t$, PVDVS 算法的复杂度为 $O(nk)$. k 的值一般选择为 50~ 100 之间.

基于 PVDVS 的流程和基于 FAEE 的流程分别如图 1(a)和 1(b)所示,除了对于分配的评估方法不同外,流程的其他部分均相同.

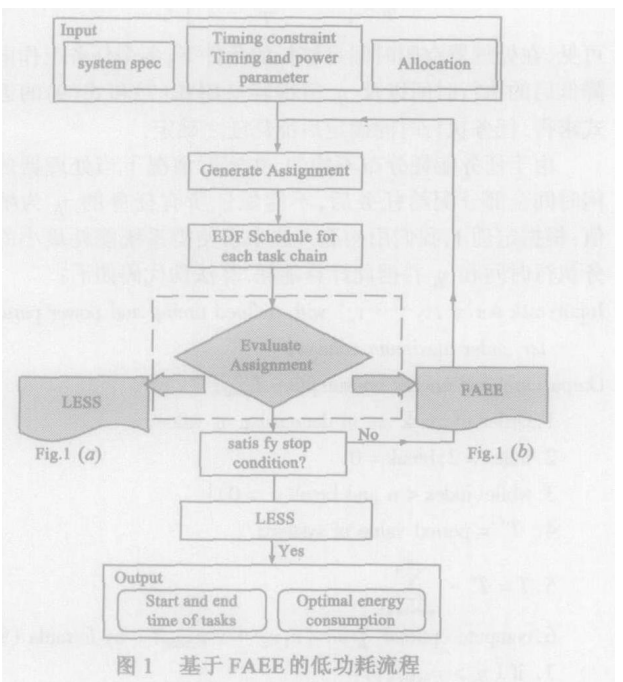

图 1 基于 FAEE 的低功耗流程

由于初始解的随机性,以及遗传进化过程中的随机性,分配解中可能会出现某个处理器负载过重超出 deadline. 为了描述由于违反时间约束给分配解带来的负面影响,在分配解的评价函数中引入了时间约束违反因子 r_i:

如果 $ft_i - dl_i > 0$, $r_i = \left(1 + \frac{|ft_i - dl_i|}{LCM(dl)}\right)^2$; (10)

否则 $r_i = 1$; 其中 ft_i 为任务 i 的执行结束时间.

评价函数为: $F(\text{assign}) = \sum_{i=1}^{n} E_{\min} * \prod_{i=1}^{n} r_i$ (11)

评价函数 $F(\text{assign})$ 的值越小,优先级越高.

假设遗传算法中采用的分配样本数为 m,进化代数为 n. 基于 FAEE 的流程复杂度为 $O(mn^2)$,而基于 PVDVS 的流程复杂度为 $O(mn^2k)$,两者相差 K 倍,可以推断采用 FAEE 最为低功耗循环优化流程里最内层的算法,相对 PVDVS 方法在保证优化精度的前提下,可以提高计算效率.

4 试验结果

为验证本文方法的有效性,用多组随机产生的例子进行测试. 在 Pentium 3/850MHz/256MB 计算机上用 C++ 程序实现了 FAEE 和 PVDVS 的算法流程.

由于整个流程的核心是 FAEE,所以先对 FAEE 准确性作评估. 试验一检验 FAEE 的精确性. 例子为随机产生的 20 组任务集合运行在单个处理单元上,任务数从 5 到 20 不等,任务在处理单元上的时间和功耗参数在 10~ 500 数值间随机产生.

结果如图 2 所示,其中 E_{FAEE} 为 FAEE 算法得到的能耗值, E_{PVDVS} 为 PVDVS 算法得到的能耗值. 参数 $\Delta d = (E_{FAEE} - E_{PVDVS})/E_{PVDVS}$, 它表示 E_{FAEE} 相对 E_{PVDVS} 的偏差. 由图可见, Δd 在 - 5% 左右浮动,而且对于不同的任务集合基本保持这个值,具有较好的一致性.

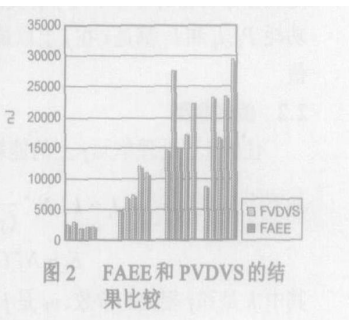

图 2 FAEE 和 PVDVS 的结果比较

试验 2 验证 FAEE 用于低能耗分配的有效性. 随机产生的 4 组例子,每组包含 10 个例子. 4 组的任务数和处理器单元个数分别为 20, 20, 50, 50 和 2, 4, 5, 10 个,时间和功耗参数随机产生,以保证试验的覆盖性. 分配的遗传算法实现中,个体数目为 100,进化代数为 40 代,染色体交配重叠概率为 50%,最大变异概率为 20%. 每个试验重复 20 次,表 1 结果是 20 次

表 1 基于 FAEE 的低能耗分配和文献[3]方法的能耗和运算时间比较

Benchmarks		PVDVS		FAEE	
		Energy(nJ)	CPU Time	Energy(nJ)	CPU Time
20 nodes on 2 PEs	1	9199	288s	9188	< 1s
	2	8668	146s	8558	1s
	3	10569	91s	10371	< 1s
	4	19946	59s	19859	< 1s
	5	25439	100s	25192	1s
20 nodes on 4 PEs	1	5249	256s	5350	2s
	2	4906	288s	4952	5s
	3	5957	261s	5898	4s
	4	10886	414s	11037	5s
	5	12434	197s	12429	2s
50 nodes on 4 PEs	1	26445	3695s	26234	1s
	2	41383	2610s	41745	2s
	3	42797	2770s	42856	5s
	4	42966	2797s	43010	2s
	5	35396	3843s	35254	7s
50 nodes on 10 PEs	1	16845	7232s	16957	18s
	2	21120	1081s	21090	25s
	3	31334	1929s	31369	22s
	4	20575	5003s	20982	36s
	5	19308	4932s	19393	52s
average		20571	1899s	20587	9.6s

结果的均值.

试验结果如表 1 所示. 结果表明, 对于我们随机产生的 20 个例子, 基于 FAEE 的分配算法所得到的能耗和基于 PVDVS 的分配算法的结果一致, 存在的不到 1% 的误差可以认为是遗传算法本身具有的进化的随意性所带来的微扰. 基于 PVDVS 方法的计算时间高出基于 FAEE 方法约 2 个数量级, 验证了第 3 节关于 FAEE 算法复杂度低的结论. 随着试验例子规模的扩大, 基于 FAEE 的低能耗分配方法在计算时间上的优势将更为明显.

5　结论

本文讨论了一种基于 fame-based 系统低复杂度的能耗估计算法 FAEE, 并在此基础上提出了快速低能耗分配方法. 和同类方法相比, 在获得相同优化结果的基础上, 复杂度降低了约两个数量级. 试验数据表明了算法的有效性.

参考文献:

[1] A Raghunathan, N K Jha, S Dey. High Level Power Analysis and Optimization[M]. Dordrecht Norwell: Kluwer Academic Publishers, 1997.

[2] K Buchenrieder, A Sedlmeier. Industrial HW/SW Co-Design[A]. Proceedings of the NATO Advanced Study Institute on Hardware/Software Co-Design[C]. Tremezzo, Italy: IEEE press, 1995. 453-466.

[3] Marcus T. Schmitz, Bashir M. Al-Hashimi. Energy-efficient mapping and scheduling for DVS enabled distributed embedded systems[A]. Proc of the IEEE conference on DATE[C]. San Jose: IEEE Press, 2002. 721-725.

[4] Yumin Zhang, Xiaobo Hu, Danny Z Chen. Task scheduling and voltage selection for energy minimization[A]. Proc of the IEEE conference on DAC[C]. San Jose: IEEE press, 2002. 183-188.

[5] W A Horn. Some simple scheduling algorithms[J]. Naval Research Logistics Quarterly, July 1974, 21(4): 177-185.

作者简介:

粟雅娟　女, 1975 年出生于贵州省, 1998 年获电子科技大学硕士学位, 现在清华大学微电子所攻读博士学位, 主要研究方向为系统级低功耗方法学研究. E-mail: syj00@mails.tsinghua.edu.cn.

魏少军　男, 1958 年 5 月出生于北京, 工学博士, 现任清华大学教授, 博士生导师; 大唐电信科技股份有限公司总裁; 中国 ICCAD 联谊会副理事长; 中国通信学会青年工作委员会委员; 中国电子学会高级会员, IEEE 有价值会员, 研究方向为: 深亚微米集成电路设计方法学研究, 面向设计再利用的 SOC(System on a Chip) 设计方法学研究和高层次综合技术研究.