

芯片动态门限静态功耗的优化技术

李先锐¹, 葛海波², 来新泉¹, 李玉山¹

(1. 西安电子科技大学电路设计研究所 西安 710071; 2. 西安邮电学院电信系 西安 710061)

【摘要】提出了一种双阈值电压的动态门限静态功耗优化算法。该算法通过直接统计电路门级节点的松弛裕度, 利用静态时序分析其最大松弛裕度及邻节点松弛裕度特征, 区分电路中的关键与非关键节点并分步调整其相应的阈值电压, 从而有效地实现了对CMOS电路静态功耗的优化设计。基于ISCA85基准实验电路集, 采用该技术和以往的算法进行了对比验证。结果表明, 该算法在不降低静态功耗优化效率的同时, 优化时间缩短了95%以上, 适合于超大规模电路静态功耗优化。

关键词 双阈值; 动态门限; 静态功耗优化; 静态时序分析

中图分类号 TN710.9

文献标识码 A

doi:10.3969/j.issn.1001-0548.2009.03.030

Optimization Techniques of Static Power Dissipation in Chip with Dynamical Threshold

LI Xian-rui¹, GE Hai-bo², LAI Xin-quan¹ and LI Yu-shan¹

(1. Research Institute of Design Circuit, Xidian University Xi'an 710071;

2. Department of Telecommunication, Xi'an Institute of Post and Telecommunications Xi'an 710061)

Abstract A novel dynamic threshold static power optimization algorithm is presented. After accounting the relaxation margins of gate level nodes, the maximum margin in circuits and characteristics in adjacent gates are analyzed with static timing, the key and non-key nodes are then divided, their thresholds are adjusted within stages, and the efficient power optimization of CMOS circuits is finally implemented. Compared with existing algorithms, experimental results with ISCA85 benchmark circuits show that processing time of the algorithm can be reduced 95% at least without reducing the optimization efficiency of static power.

Key words Dual-threshold; dynamical threshold; static power optimization; static timing analyzing

随着深亚微米工艺的进步、器件尺寸的缩小以及电源电压的降低, 静态功耗对电路总功耗的贡献在日益增加, 如何减低静态功耗及对电路静态功耗优化已成为当前的一个热点。双阈值电压设计思想的提出, 为解决亚阈值导电产生静态功耗提供了一个行之有效的方法^[1-2]。其基本原理是在电路的非关键路径上使用高阈值电压, 在关键路径上使用低阈值电压, 在保证整个电路性能的基础上实现电路静态功耗的最小。

目前, 已经有许多双阈值电压设计的思想及优化算法^[1-8]。文献[1]在MOS管级模拟了不同阈值下整个电路的功耗与延迟, 提出了迭代最大集算法, 具有精度高、优化效率高的特点, 但算法复杂度太高, 很难在实际中得以应用。文献[5]提出了统计阈值算法, 根据修正的边集松弛裕度, 确定优化门限。但

电路优化到一定阶段后, 此算法迭代次数急剧增加, 影响了优化速度及效率。

本文基于双阈值电压设计的思想, 提出了联合电路门级节点的最大松弛裕度特征及其所有邻节点松弛裕度分布特征, 确定分割门限, 区分电路中关键节点与非关键节点集的算法。同时, 在算法具体实现时, 通过分布调整电路阈值电压, 加快算法收敛速度, 实现CMOS电路静态功耗优化。与文献[1]中算法相比, 该算法在不降低优化效率的同时, 降低了算法复杂度, 实用性强; 和文献[5]相比, 通过邻节点松弛裕度特征的引入, 减少了迭代次数, 并降低了算法发散概率, 适合大规模电路的优化。

1 电路模型描述

一个组合逻辑电路可以用一个有向的非循环图

收稿日期: 2008-07-08; 修回日期: 2009-02-26

基金项目: 国家部委重点预研项目(D1120060967, Y30306270105); 教育部超高速电路与电磁兼容重点实验室专项项目(YZCB2008001)

作者简介: 李先锐(1974-), 女, 博士生, 主要从事数模混合集成电路低功耗设计方面的研究。

$G(V, E)$ 表示, 其中 V 代表电路中的各个逻辑门, 也称门级节点^[1]。有向边集 E 则表示各级门之间的连线, 方向是从输入指向输出, 称为边。如果门 V 的输出连接至门 U 的一个输入端, 则 V 和 U 之间就存在一条从 V 到 U 的边。由于一个逻辑门可能存在多个扇入与扇出, 所以一个门级节点可能对应多个边。每个节点 $v \in V$ 都具有一个延迟属性 $d(v)$, 其值为该节点对应的本征延迟。对延迟 D 中的每一个节点 v , 定义其到达时间为:

$$t_a(v) = \max_{u \in I_v} [t_a(u) + d(v)] \quad (1)$$

定义 v 的要求时间为:

$$t_r(v) = \min_{u \in O_v} (t_r(u) - d(v)) \quad (2)$$

式中 I_v 和 O_v 分别为 v 的扇入和扇出节点集合, 则节点 v 的松弛裕度 $S(v)$ 为:

$$S(v) = t_r(v) - t_a(v) \quad (3)$$

以上处理流程称为静态时序分析。如果 $S(v) \geq 0$, 说明信号会在延迟所要求的时间之前到达, 电路能正常工作, 否则就会发生时序违反, 电路将存在竞争冒险现象。节点的 $S(v)$ 为0或很小, 则称该节点处于关键路径上, 如果增加该节点的阈值, 则会破坏电路的时延。通常节点的松弛裕度越大, 则该节点被优化的可能性越大^[1]。如果该节点的所有扇入节点和扇出节点的松弛裕度越大, 表明该节点距离关键路径越远, 该节点被优化的概率也相应较大。对于被优化可能性较大的节点, 采用高阈值电压器件, 不影响整个电路的时延, 则称该节点优化成功; 如果电路延时被破坏, 表明该节点不能被优化。

2 动态门限算法

根据逻辑门节点的松弛裕度, 对非关键节点进行逐节点优化, 则每个门节点都需要一次静态时序分析, 运算量大。为降低运算量, 采用分步方法, 即初始优化及细优化。在初始阶段, 根据整个电路的所有节点的最大松弛裕度 $\text{Max}(S(v))$, 确定优化门限, 完成初始优化。在细优化阶段, 算法接合统计节点的松弛裕度及其邻节点平均松弛裕度的特征, 决定细优化阶段的门限, 该算法称为动态门限算法。算法流程如图1所示。

2.1 电路遍历

电路节点的松弛裕度及电路是否存在竞争是通过静态时序分析算法完成的。静态时序分析根据电路中各节点的拓扑关系, 遍历电路得到各个门节点

的 $t_a(v)$ 、 $t_r(v)$ 、 $S(v)$, 故电路遍历直接影响整个电路的优化速度。电路的遍历分为深度优先和宽度优先搜索算法。由于在静态时序分析时, 首先要获得整个电路的最大路径延迟, 它对电路中路径深度更为敏感, 所以本文采用深度优先搜索算法。电路遍历时又会产生大量的中间电路节点集合, 对计算机的内存提出很高的要求^[3]。为此, 引入了动态电路链表, 用电路节点的入栈和出栈操作来代替大量的内存分配。下面以获得电路门节点的到达时间 $t_a(v)$ 的前向搜索过程为例, 说明该算法的实现流程。

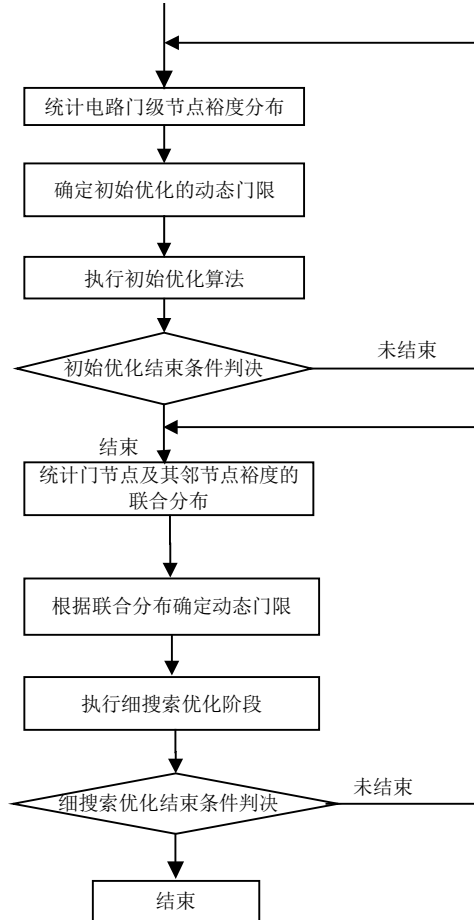


图1 算法流程图

```

procedure Forward( $G$ , gateLink) { /*  $G$  =
gate node gateLink = gate link */
get all fanout node gFanout numbers of  $G$ 
for each fanout node gFanout {
create { $G$ , gFanout} node
add { $G$ , gFanout} node to gateLink tail
if the fanout node is the output node
compute  $t_a(v)$ , remove tail of gateLink
else
Forward (gFanout, PathLink)
}
}

```

}

2.2 最大松弛裕度及与邻节点裕度特征

初始阶段算法优化门限为 $\alpha \text{Max}(S(v))$, 由于每次优化后, $\text{Max}(S(v))$ 的值是不断变化的, 它随电路优化过程中松弛裕度分布的变化, 自动调整候选优化门节点集合的大小, 减少了优化失败的次数。具体算法如下:

```

Procedure coarseoptima(G){
    static timing analyze, get S(v)
    select SVthread =  $\alpha \text{Max} S(v)$ 
    for each gate
        
$$V_t = \begin{cases} V_{th} & S(v) \geq \text{SVthread} \\ V_{il} & \text{else} \end{cases}$$

    }
    
```

仿真表明, α 的取值在 [0.4,0.6] 之间时比较合理, 本文选取 0.5。通过初始阶段的算法处理, 被节点的优化比例一般会超过 60%。初始阶段优化后, 剩余节点的松弛裕度分布都比较平坦, 且裕度值的动态范围也比较小。继续采用初始阶段的算法会造成大量的重复回溯处理, 为此, 在细搜索阶段引入邻近节点的平均松弛裕度。根据相邻节点的平均松弛裕度 S_{adj} 以及该节点的松弛裕度 $S(v)$ 形成一个新的权值:

$$S = \beta S(v) + (1 - \beta) S_{adj} \quad (4)$$

式中 β 为可调参数, 通过改变 β 的值, 调整本节点的松弛裕度及其相邻节点的平均松弛裕度在新的权值中所占的权重。通过仿真, 本文中 β 值设为 0.3。由式(4)可看出, 节点松弛裕度越大, 其权值也就越大。同时, 如果该节点所有的扇入节点和扇出节点的平均松弛裕度越大, 则该权值也就越大。权值越大的节点, 被优化的概率也就越高。在得到所有权值后, 会形成一个权值分布。然后根据权值分布, 确定域值及候选节点集的范围。细搜索阶段算法的处理与初始阶段基本相同, 将初始阶段中的 $S(v)$ 用新的权值 S 代替, 且 α 取 0.8 即可。

2.3 算法的有效性

以 C5315 为例, 图 2 所示为算法运行过程中电路松弛裕度分布变化图。其中 C5315-A 表示电路优化前的松弛裕度分布, C5315-B 表示电路经过初始优化后的松弛裕度分布, C5315-C 表示进入细优化阶段后的首次与邻节点加权而形成的松弛裕度分布。横坐标表示松弛裕度值, 纵坐标表示具有该松弛裕度值的节点占整个电路节点的比例。

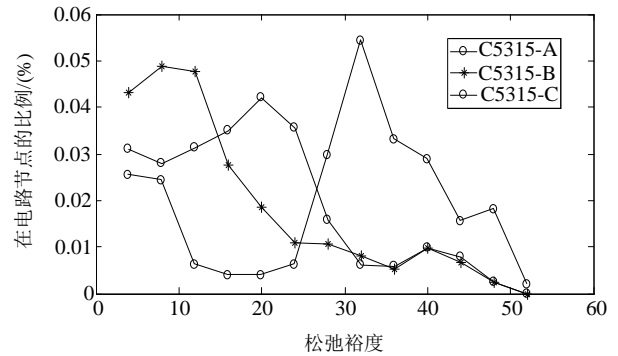


图2 C5315电路松弛裕度分布变化图

从图2可以看出, 优化前, 松弛裕度大于 20 的节点占电路总节点 75% 以上, 对这些松弛裕度较大的节点, 可以直接优化而不会发生优化失败, 故初始阶段选取的候选优化边集可相对较大。起始阶段通常有 70% 的节点优化成功。初始优化后, 电路松弛裕度分布如 C5315-B, 裕度分布基本集中在 15 以下, 且分布较为集中, 即整个电路节点松弛值及动态范围都比较小, 如果继续采用初始优化算法, 则优化失败的概率将会显著增加。C5315-C 为通过与邻节点加权后的松弛裕度分布, 通过加权的方法, 使 15 以下节点比例降低, 15~30 区间的节点比例增加, 加大了松弛裕度值的动态范围, 使得电路优化迭代次数降低。

3 实验结果

动态门限算法在 CPU 1.8 GHz, WIN2000 平台的 PC 机上, 采用 C 语言实现。基于 ISCA85 基准实验电路, 本文采用单位延时模型, 对动态门限算法和文献[1]的结果进行了对比验证。实验结果如表 1 所示。

文献[1, 5]分别对各自的算法复杂度做了详细的分析, 但其分析基础为完成一次电路扫描所需的运算量, 其中文献[1]由于采用了 SPICE 模型, 对所有的节点集以及与该节点相连的边集都做了处理, 运算量为 $O(E(V+E))$ 。而本文算法采用了单位延时模型, 且只对节点处理, 运算复杂度为 $O(V)$ 。在实际算法运行中, 存在着大量重复的电路扫描, 耗费了算法的大部分时间, 如何减少重复扫描次数, 文献[1,5]都未提及。考虑到运行平台的差异性, 本文算法的耗时缩短了 95% 以上。另外, 从表 1 和文献[1]的优化比例来看, 对于逻辑层次结构比较复杂的电路, 如 C432、C1355、C6288 等电路, 动态门限算法优势也比较明显。这主要是因为, 在细搜索阶段, 动态门限算法采用了松弛裕度与邻节点裕度相结合, 在选择候选节点时, 能够最大限度地提取候选集并降低的算法发散概率。此外, 随电路规模的增大,

文献[1]算法耗时显著增大,而本文算法确相对平缓,对大规模电路,更具有实际应用意义。

表1 本文实验结果及与文献[1]的比较

电路	总逻辑门	非关键门	优化比例/(%)		耗时/m	
			文献1	本文算法	文献1	本文算法
C432	160	81	33.04	5.70	89.0	0.033
C880	383	319	75.48	21.54	84.0	0.016
C1355	546	120	67.50	29.69	100.0	0.850
C1908	880	789	56.53	45.28	76.6	2.400
C2670	1 193	1 083	76.17	57.71	79.7	2.180
C3540	1 669	1 527	78.15	179.18	77.8	2.450
C5315	2 307	2 193	90.30	127.28	89.8	2.780
C6288	2 416	2 135	42.98	1 398.82	57.0	6.210
C7552	3 512	3 380	*	*	90.3	2.130

4 结 论

静态功耗优化是目前低功耗设计中的研究热点,双阈值电压设计是降低电路的静态功耗较为行之有效的方法。本文提出的动态门限算法实现了快速有效地区分电路中的关键与非关键路径,是实现双阈值电压设计的一种有效的途径,也为CMOS电路的其他优化算法^[9]提供了新的实现途径。实验结果表明该算法在保持优化效率的同时,有效地降低了对电路重复扫描的次数,提高了优化速度,优化时间缩短了95%以上,特别适合于对大规模集成电路静态功耗的优化。

参 考 文 献

[1] WANG Qi, VRUDHULA S B K. Algorithms for minimizing standby power in deep submicrometer, dual- V_t CMOS circuits[J]. IEEE Transactions on CAD of IC and System, 2002, 21(3): 306-318.

[2] ZHANG Tao, BOULT T E, JOHNSON R C. Two thresholds are better than one[C]//Computer Vision and Pattern Recognition 2007 IEEE Conference. [S. l.]: IEEE, 2007.

[3] WEI L, CHEN Z, JOHNSON M, et al. Design and optimization of low voltage high performance dual threshold CMOS circuits[C]//Proceedings of Design Automation Conference. [S. l.]: Asia and South Pacific, 1998.

[4] KONRAD E, TH OMAS K, ROGER L, et al. Algorithms for leakage reduction with dual threshold design techniques[C]//International Symposium on System-on-Chip. Tampere, Finland: [s. n.], 2006.

[5] 徐勇军, 骆祖莹, 李晓维. 双阈值CMOS电路静态功耗优化[J]. 计算机辅助设计与图形学学报, 2003, 15(3): 264-269.

XU Yong-jun, LUO Zu-ying, LI Xiao-wei. Optimization of static power for dual threshold CMOS circuits[J]. Journal of Computer Aided Design & Computer Graphics, 2003, 15(3): 264-269.

[6] CHEN Jian, XIE Bing, GUAN Yong-liang, et al. An on-off-keying optical receiver with dual thresholds and erasure zone[J]. Photonics Technology Letters, 2007, 19(2): 1592-1594.

[7] KULKARNI S H, SYLVESTER D M, BLAAUW D T. Design-time optimization of post-silicon tuned circuits using adaptive body bias[J]. Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(3): 481-494.

[8] KURSUN V, FRIEDMAN G E. Sleep switch dual threshold voltage domino logic with reduced standby leakage current[J]. Very Large Scale Integration (VLSI) Systems, 2004, 12(5): 485-496.

[9] NING Ning, WU Shuang-yi, WANG Xiang-zhan, et al. A novel power optimization method by minimum comparator number algorithm for pipeline ADCs[J]. Journal of Electronic Science and Technology of China, 2007, 5(1): 75-80.

编辑 税红