

电源线/地线网络单点 SOR 统计分析方法

骆祖莹¹, 潘月斗², 余先川¹

(1. 北京师范大学信息科学与技术学院, 北京 100875; 2. 北京科技大学信息科学与技术学院, 北京 100083)

摘 要: 随着集成电路工艺进入纳米工艺时代, VLSI 规模在增大的同时, 还伴随着显著的工艺参数变化, 使得电源线/地线网络(P/G 网)分析从确定式分析算法转变为统计式分析算法, 迫切需要能够降低算法复杂度的局部分析算法. 为了计算设计者感兴趣的 P/G 网少数 IR 电压降比较大的问题节点电压变化, 必须分别计算出这些点的相关电阻向量, 本文提出了一种单点 SOR (Successive Over Relaxation: 连续过松弛-超级松弛) 的统计分析方法(SN-SOR). 与传统的全局 SOR 方法相比, SN-SOR 方法有如下三个优点: (1) 局部松弛. 由于计算一个问题节点 q 的相关电阻向量, 必须仅在 q 点加一个激励, 所以 SN-SOR 方法不是采用全局电路节点的顺序松弛方法, 而是采用从 q 点不断向周围节点进行松弛的波状松弛方法, 当某些节点的 IR 电压降小于一个极小的设定值时(即相关电阻足够小), 这些节点就不再向外进行松弛计算, 因此 SN-SOR 方法具有局部松弛的特性. (2) 高效. 与传统的全局 SOR 方法相比, SN-SOR 方法不仅松弛点非常少, 而且松弛次数也有所减少. (3) 低空间复杂度. 当计算出 q 点相关电阻向量后, SN-SOR 方法不是将所有相关电阻都存起来, 而是只将强相关电阻存起来, 对于大多数的弱相关电阻, 仅将少数代表节点上的弱相关电阻存起来, 所以本文方法的空间复杂度比较低. 大量的实验数据表明, 与全局 SOR 求解方法相比, SN-SOR 方法在保持较高精度(误差小于 0.38%)的前提下, 速度可以提高 20 倍.

关键词: VLSI; P/G 网; 统计分析; SOR; 单点分析

中图分类号: TP391.9 **文献标识码:** A **文章编号:** 0372-2112 (2007) 11-2043-07

Single-Node SOR Method for Statistic Analysis of Power/Ground Networks

LUO Zu-ying¹, PAN Yue-dou², XU Xian-chuan¹

(1. College of Information Science and Technology, Beijing Normal University, Beijing 100875, China;

2. College of Information Science and Technology, Beijing University of Science and Technology, Beijing 100083, China)

Abstract: With technology scaling into nanometer regime, VLSI chips become much more complex and suffer from rampant process variations. Therefore, statistical methods are replacing deterministic counterparts for power/ground(P/G) grid analysis and efficient local analysis methods are badly needed to reduce the analysis complexity. A novel single-node Successive Over Relaxation (SN-SOR) method is proposed to efficiently solve correlated resistor vectors and then to directly compute voltage variations for question nodes of large IR droop. Compared with traditional global SOR methods, the SN-SOR method shows following advantages. First one is locality. Because only one stimuli is located at the question node q to compute the correlated resistor vector R_q , SN-SOR relaxes nodes from q to its surround nodes as wave transmission and the wave stops at some nodes whose IR droop is less than one assigned little value. Second one is efficiency. SN-SOR only relaxes small parts of nodes in P/G circuits but also needs slightly less relaxation reiterations. Third one is low memory complexity. Among R_q , SN-SOR only needs some strong correlated resistors and less weak resistors of typical nodes to compute the voltage variation. Experiments show SN-SOR is 20 times faster than global SOR methods with only 0.38% accuracy loss.

Key words: VLSI; P/G network; statistic analysis; SOR; single-node analysis

1 引言

随着集成电路工艺进入纳米时代, 集成电路工作频率和功耗越来越高^[1], 为了降低功耗, 工作电压也越来

越低, 导致集成电路供电电压允许的波动范围越来越小^[2~9], 使得电源线/地线网络(P/G 网, 即集成电路供电网络)难于提供合适的供电电压, P/G 网设计复杂度越来越大, 必须经过大量的优化才能获得满意的设计结

果,因此,为了保证优化的速度,必须尽可能地降低 P/G 网分析算法的复杂度,目前也提出了大量的 P/G 网分析算法^[2~9].

在集成电路工艺进入纳米时代之前,由于工艺参数变化不是非常显著,对 P/G 网设计的影响比较小,P/G 网分析算法研究主要集中在确定式算法领域^[2~6],国内的研究也集中在这个领域^[2~4].随着纳米时代的来临,工艺参数变化变得比较显著,在 P/G 网设计中不得不正视其影响^[10],因此 P/G 网分析算法研究也由确定式算法研究转向统计式算法研究,并出现相应的研究成果^[7~9].文献[7]采用代数方法对系数矩阵进行变换,直接根据吸纳电流的变化,计算 P/G 网节点电压的变化,由于矩阵变换的复杂度非常高,文献[7]的缺点是计算复杂度非常高.文献[8]采用数学规划的方法,来计算 P/G 网节点电压的变化上限,由于数学规划方法本身的复杂度就非常高,文献[8]的缺陷也在于算法复杂度大.文献[9]对随机行走方法^[5]进行改进,使之能够计算 P/G 网节点电压的变化,但随机行走方法本身的复杂度非常高,同时还存在误差累积的问题.

由于 P/G 网电路越来越大,并且 C4-PAD 阵列(图 1 中的小方块)供电的 P/G 网电路分析具有局域性^[3,5],设计人员更感兴趣的是 IR 电压降比较大的问题节点(图 1 中的

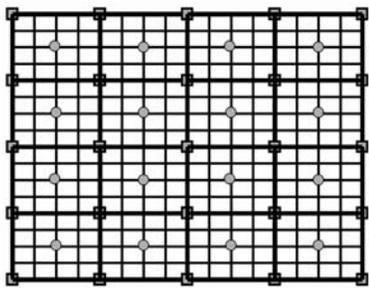


图 1 C4-PAD 阵列供电的 P/G 网示意

圆点)^[5],为了提高分析效率,文献[5]提出了随机行走算法,它是一种高效的单点求解方法,但由于它无法直接计算单点的相关电阻向量,所以这种方法无法直接应用到 P/G 网的单点统计分析中.文献[7,8]所提出的方法都是 P/G 网的全局或局部统计算法,也无法直接应用到 P/G 网的单点统计分析中.文献[9]虽然也是一种单点统计算法,但它无法解决电路参数的空间相关问题.ICCG 算法^[2]和 SOR 算法^[6]都可以直接计算单点的相关电阻向量,所以可以用于 P/G 网的单点统计分析,但这两种方法必须使用 P/G 网中全部节点来、进行单点相关电阻向量的求解,因此它们的效率是比较低的.

基于 SOR 算法^[6],本文提出了一种单点相关电阻向量的高效求解方法,即单点 SOR 算法(SN-SOR).由于计算一个问题节点 q 的相关电阻向量,必须仅在 q 点加一个激励,所以 SN-SOR 不是采用全局电路节点的顺序松弛方法,而是采用从 q 点不断向周围节点进行松弛的波状松弛方法,当某些节点的 IR 电压降小于一个

极小的设定值时(即相关电阻足够小),这些节点就不再向外进行松弛计算,因此 SN-SOR 是一种局部松弛方法.当计算出 q 点相关电阻向量后,本文方法不是将所有相关电阻都存起来,而是只将强相关电阻存起来,对于大多数的弱相关电阻,仅将少数代表点上的弱相关电阻存起来,所以 SN-SOR 的空间复杂度比较低.与 ICCG 算法和全局 SOR 算法相比,SN-SOR 可以快速而精确地计算出 P/G 网中问题节点的相关电阻向量,进而直接计算出这些问题节点的电压变化,大量的实验数据表明:与 5000 次模拟方法相比,SN-SOR 在几乎不损失精度(误差 0.03%)的前提下,速度可以提高 2 个数量级;与全局 SOR 求解方法相比,SN-SOR 在非常小的精度损失(误差 0.38%)前提下,速度可以提高 20 倍.

2 本文工作的研究基础

2.1 P/G 网的构成

在当今高端芯片设计中,一般采用 6 层或 8 层布线,P/G 网占用最上与最下两个层对,构成上下两个供电网,并用通孔(via)将双层网连起来.上层粗线网采用如图 1 所示的 Mesh 结构,作为理想电压源的 C4-PAD(图中的小方块),呈阵列状排列,相邻的 4 个 PAD 所围的区域称为一个 Block(块),所以 P/G 网可分为多个 Block,达到就近供电、多路冗余的目的.在下层细线网中,为了灵活地向逻辑单元供电,可以采用多种拓扑方式进行设计^[4],如网状、树状、和链状结构.

对如图 1 所示的 Mesh 结构 P/G 网,我们的前期工作表明:这种结构的 P/G 网具有较明显的局域性.在文献[3]的工作中,我们定义 α 为宽度比,即与 PAD 相连的粗线宽度和未与 PAD 相连的细线宽度之比,我们发现随着 α 的不断增大,P/G 网的局域性不断增强.

2.2 P/G 网 SOR 确定式分析方法

P/G 网统计分析一般采用直流分析方法^[7~9],下面介绍 P/G 网 SOR 确定式直流分析方法^[6],作为本文研究的基础.直流分析方法均采用改良的节点电压分析方法(MNA)^[2~9]对 P/G 网进行分析,设 I 为 P/G 网节点上的吸纳电流向量, V 为 P/G 网节点上所产生的 IR 电压降向量, G 为 P/G 网节点之间的连接电导矩阵,则 V 与 I 的关系为:

$$GV = I \quad (1)$$

在 SOR 算法中,先设所有节点电压的初始值为 V_{DD} .假设第 k 步松弛结束,得到节点电压向量为 V^k ,为了进行第 $k+1$ 步松弛,我们将 V^k 装入一个中间节点电压向量 \bar{V} .当我们对任意节点 p 进行第 $k+1$ 步松弛,需要进行如下计算:

$$V_p^{k+1} = (1.0 - \lambda) V_p^k + \frac{\lambda}{G_p} \left(\sum_{j=1}^{n_p} (G_{Lp(j)} \bar{V}_{Np(j)}) - I_p \right) \quad (2)$$

式中 I_p 为 p 点的吸纳电流, p 点有 n_p 个邻点, N_p 和 L_p 分别是 p 点的邻点集合和邻接线集合, $\bar{V}_{N_p(j)}$ 是第 j 个邻点的中间节点电压, $G_{L_p(j)}$ 是第 j 个邻点所连线的电导, G_p 是 p 点电导, 为 $G_p = \sum_{j \in L_p} G_{L_p(j)}$, λ 为过松弛因子, 可以根据 P/G 网电路参数计算出 λ 的最优值 B_{opt} [6], 本文采用一维搜索方法获得其最优值 B_{opt} .

当计算出 V_p^{k+1} 后, 就用 V_p^{k+1} 去替换 \bar{V}_p . 当电路中所有点都松弛过一遍后, 中间节点电压向量 \bar{V} 就是第 $k+1$ 步松弛结果 V^{k+1} . 如果 V^{k+1} 中所有点的误差绝对值均小于一个极小的设定值 ϵ_1 , 则结束松弛, 求解出 P/G 网的节点电压向量 V . 节点 p 的误差 e_p 可按下式计算:

$$e_p = \sum_{j=1}^{n_p} \{G_{L_p(j)}(\bar{V}_{N_p(j)} - \bar{V}_p)\} - I_p \quad (3)$$

2.3 P/G 网统计分析的形式化

在 P/G 网统计分析中, 均考虑吸纳电流 I 的变化 ΔI [7-9], 最近还进一步考虑了连接电导 G 的变化 [9], 由于 G 的变化 ΔG 可以折算为 I 的变化 ΔI [9], 所以本文仅考虑吸纳电流 I 的变化 ΔI , 即根据 ΔI 来快速计算出节点电压变化 ΔV :

$$G\Delta V = \Delta I \quad (4)$$

由于纳米工艺参数变化比较大, 所以 ΔI 相对比较大, 本文取 ΔI 的 $3\sigma = 60\%$. 类似于文献 [9], 本文将 ΔI 定义为正态分布变量 $N(I\mu, I\sigma)$, 并存在空间相关, 在图 1 中, 本文定义 1 个 block(块)内的所有点为完全相关, 块 b_i 和块 b_j 之间的相关系数 C_{b_i, b_j} 可用下式计算

$$C_{b_i, b_j} = f(x_{b_i, b_j}, y_{b_i, b_j}) \quad (5)$$

式中 x_{b_i, b_j} 和 y_{b_i, b_j} 是块 b_i 和块 b_j 之间的横向和纵向距离. 如果块 b_i 和块 b_j 之间的距离越小, 两块之间的相关系数越大, 即相关性越强; 块 b_i 和块 b_j 之间的距离越大, 两块之间的相关系数越小, 即相关性越弱.

3 现有的 P/G 网单点统计分析方法

3.1 相关电阻向量的现有求解方法

设 R_q 为 q 点 (IR 电压比较大的问题节点) 和所有点之间的相关电阻所组成的相关电阻向量. 为了求取 R_q , 在电路上加一个激励向量 $I_q = (0, 0, \dots, 1, \dots, 0, 0)^T$, 即仅在 q 节点上加一个单位吸纳电流, 其余点均不接吸纳电流. 这时求解出来的电压向量 V_q , 由于仅在 q 节点上加一个单位吸纳电流, 所以 R_q 可表示为:

$$R_q = R \times I_q = G^{-1} \times I_q = V_q \quad (6)$$

我们可以采用现有的 ICCG 算法 [2] 和 SOR 算法 [6], 使用全局电路来计算相关电阻向量 R_q , 尽管计算精度高, 但速度慢、需要存储空间较多.

由于在 R_q 中, 如果 p 点离 q 点比较近, 其相关电阻

$r_{p,q}$ 比较大, 如果 p 点离 q 点比较远, 其相关电阻 $r_{p,q}$ 比较小, 即如图 1 所示 Mesh 状 P/G 电路, 相隔较远的 Block 之间的相关性比较小 [3,5]. 由于 P/G 电路具有较强的局域性 [3], 为了降低求取 R_q 的复杂度, 可以简单地取 q 点周围的 9 个 Block 组成一个局部电路, 来求 V'_q , 根据 I_q 和 V'_q 所计算出的 R'_q , 它和 R_q 的区别在于 q 点对 9Block 以外所有点的相关电阻均为 0, 也就是忽略 q 点和 9Block 以外所有点的弱相关关系. 与传统的全局方法相比, 9Block 方法的优点是速度快, 需要存储空间较少, 但是以牺牲一定计算精度为代价的.

3.2 单点电压变化的分析方法

对于问题节点 q , 我们可以用 R_q 来计算此点电压变化的期望 $E(V_q) = v\mu_q$:

$$v\mu_q = E(V_q) = \sum_{p=1}^N (r_{p,q} I\mu_p) \quad (7)$$

式中 N 是 P/G 网的节点数, $I\mu_p$ 是 P/G 网 p 节点吸纳电流变化的期望值. 由于 ΔI 是空间相关的参数变化, 在计算 q 点电压变化的均方差 ω_q 必须考虑空间相关系数:

$$(\omega_q)^2 = \sum_{bi=1}^M \left\{ \omega_{q, bi} \sum_{bj=1}^M (C_{bi, bj} \times \omega_{q, bj}) \right\} \quad (8)$$

式中 M 是 P/G 网的 Block 数, $C_{i,j}$ 是块 b_i 和块 b_j 之间的 ΔI 相关系数, $\omega_{q, bi}$ 是块 b_i 内所有节点的 ΔI 对 q 节点电压变化均方差所产生的分量, 由于块 b_i 内所有节点是完全相关的, 所以可以用下式计算 $\omega_{q, bi}$:

$$\omega_{q, bi} = \sum_{p \in N_{bi}} (r_{p,q} \times I\sigma_q) \quad (9)$$

式中节点 p 属于块 b_i 的节点集合 N_{bi} . 基于 q 点电压变化的期望 $v\mu_q$ 和均方差 ω_q , 我们可以进一步计算出 3sigma 偏差界 vB_q :

$$vB_q = v\mu_q + 3 \times \omega_q \quad (10)$$

4 P/G 网单点 SOR 统计分析方法 (SN-SOR)

4.1 相关电阻向量的 SN-SOR 求解方法

如前所述, 为了求取 R_q , 在电路上加一个激励向量 $I_q = (0, 0, \dots, 1, \dots, 0, 0)^T$, 即仅在 q 节点上加一个单位吸纳电流, 所求出的电压向量 V_q 就等于 R_q . SN-SOR 求解 R_q 的算法流程如下:

Step 1 初始化. 所有点的电压为 V_{DD} , q 节点作为边缘节点, 第一个放到松弛集合 A 中.

Step 2 如果 A 中所有节点都松弛过后, 转到 Step 7. 否则, 从 A 中顺序取出下一个节点 p , 转到 Step 3.

Step 3 如果 p 点是已经松弛好的节点, 则返回 Step 2. 否则转到 Step 4.

Step 4 对于 p 点进行松弛计算后, 分三种情况进行处理. (a) 如果节点 p 是边缘节点, 并且 $V_{DD} - V_p$ 大

于一个极小的设定值 ε_2 , 则将 p 点标为内部点, 转到 step5. (b) 如果节点 p 是边缘节点, 并且 $V_{DD} - V_p \leq \varepsilon_2$, 返回 Step 2. (c) 如果节点 p 是内部节点, 转到 Step 6.

Step 5 将节点 p 邻点中从未松弛过的节点都标为边缘节点, 加到松弛集合 A 中, 返回 Step 2.

Step 6 对于内部节点 p , 如果本次松弛结果与上次松弛结果之间的差值绝对值、小于一个极小的设定值 ε_3 , 则认为这个内部节点是已经松弛好的节点, 在以后的松弛操作中不再对其进行松弛操作; 否则, 认为此点还是未松弛好的节点. 返回 Step 2.

Step 7 如果 A 中还有节点的松弛误差大于设定值 ε_1 , 就返回 Step 2, 重新松弛一遍. 否则, 程序结束, 输出结果.

在以上的算法中, 对于离 q 点比较近的点 p , 由于相关电阻比较大, 其电压降 $V_{DD} - V_p$ 大于一个极小的设定值, p 点就被标为内部点, 将松弛操作向周围传播出去. 反之, 如果点 p 离 q 点比较远, 由于相关电阻比较小, 即使它被标为边缘节点, 但由于其电压降 $V_{DD} - V_p$ 不大于一个极小的设定值 ε_2 , p 点就不能改变其边缘节点的特征, 使松弛操作不再向周围传播. 因此本文的 SN-SOR 求解方法, 对于图 1 所示的由 C4-PAD 供电的 P/G 网, 是一个很好的局部算法. 它可以在保证精度的前提下, 减少参与松弛的节点数, 提高求解的速度.

同时, 在 Step 6 中, 当本次松弛结果与上次松弛结果之间的差值绝对值、小于一个极小的设定值 ε_3 , 则认为这个内部节点已经是松弛好的节点, 在以后的松弛操作中不再进行松弛操作. 这是一项刹闸措施, 能够在松弛操作中, 跳过大量已松弛好的内部节点, 从而大大提高了本文单点松弛算法的效率.

4.2 相关电阻向量的压缩

对于求出来的 R_q , 不是全部都存起来, 而是参照 9Block 方法, 我们将 q 点与其周围的 9Block 内节点之间的连接作为强连接, 逐点存起来; 对于剩余的所有 Block, 我们存入这些 Block 中心点的相关电阻, 作为每个 Block 的代表, 以降低存储空间的需求. 这样压缩后 R_q 就变为 \hat{R}_q .

4.3 单点电压变化的分析方法

对于一个 $(N_2 + 1) \times (N_2 + 1)$ 个 C4-PAD 所组成的供电阵列, 整个 P/G 电路共包含 $M = N_2 \times N_2$ 个 Block, 组成全体块集合 BS. 具体到一个问题节点 q , 它周围的 9Block 组成了强连接块集合 SBS (Strong-Block-Set), 剩余的 $M - 9$ 个 Block 组成了弱连接块集合 WBS (Weak-Block-Set). 对于节点 q , 我们必须用压缩后的 \hat{R}_q 来计算此点电压变化的期望 $v\mu_q$:

$$v\mu_q = E(v_q) = \sum_{b=1}^M (v\mu_{q,b}) \quad (11)$$

式中 $v\mu_{q,b}$ 是块 b 中所有节点对 $v\mu_q$ 的贡献. 对于属于 SBS 集合的块 sb , 其所包含的节点就组成一个点集合 N_{sb} , $v\mu_{q, sb}$ 可以用下式计算:

$$v\mu_{q, sb} = \sum_{j \in N_{sb}} (r_{j,q} I\mu_j) \quad (12)$$

式中 j 是块 sb 中任一个节点, $I\mu_j$ 是节点 j 所连吸纳电流的均值. 对于属于 WBS 集合的块 wb , 其所包含的节点就组成一个点集合 N_{wb} , $v\mu_{q, wb}$ 可以用下式计算:

$$v\mu_{q, wb} = r_{q, wbc} \sum_{j \in N_{wb}} I\mu_j \quad (13)$$

式中 wbc 是块 wb 中的中心节点, $r_{q, wbc}$ 是中心节点 wbc 和节点 q 之间的相关电阻. 因此节点 q 电压变化的期望 $v\mu_q$ 可以用下式计算:

$$v\mu_q = \sum_{sb \in SBS} \sum_{j \in N_{sb}} (r_{j,q} I\mu_j) + \sum_{wb \in WBS} (r_{q, wbc} \sum_{j \in N_{wb}} I\mu_j) \quad (14)$$

对于节点 q , 我们同样可以用式 (8) 来计算此点电压变化的均方差 $v\sigma_q$. 在压缩后的 \hat{R}_q 中, 对于属于 SBS 集合的块 sb , 由于其所属节点 j 与节点 q 之间的相关电阻 $r_{j,p}$ 都存在 \hat{R}_q 中, 所以可以直接用式 (9) 来计算式 (8) 中的 $v\sigma_{q, sb}$. 但对于 WBS 集合的块 wb , 由于只存储中心节点 wbc 和节点 q 之间的相关电阻 $r_{q, wbc}$, 所以必须用下式计算 $v\sigma_{q, wb}$:

$$v\sigma_{q, wb} = r_{q, wbc} \times \sum_{j \in N_{wb}} I\sigma_j \quad (15)$$

式中 N_{wb} 是块 wb 所属节点的集合, 节点 j 是块 wb 所属的一个节点.

5 实验结果

5.1 实验电路参数

在实验中, 我们设计了 8 个如图 1 所示的实验电路, 每个电路均由 $(N_2 + 1) \times (N_2 + 1)$ 的 C4-PAD 阵列供电, $V_{DD} = 2.0V$, $N_2 = 30$, 所以每个电路均有 900 个 Block, 两个相邻 C4-PAD 之间共有 $N_1 + 1$ 根线, 包括与 PAD 相连的 2 根粗线、和与 PAD 相连的 $N_1 - 1$ 根细线, 每个电路共有 $(N_2 \times N_1 + 1) \times (N_2 \times N_1 + 1) - (N_2 + 1) \times (N_2 + 1)$ 个非 PAD 节点, 最大规模电路共有 1.44M 个节点.

由于 N_1 逐渐增大, 每个 Block 的节点也呈二次方的增加, 所以每个 C4-PAD 所供电流也迅速增加, 为了有效降低最大 IR 电压降 V_{ust} , 根据我们的经验^[3], 我们逐渐调大粗线与细线之间的线宽比. 同时为了降低 V_{ust} , 前 6 个电路中每个节点的电流变化服从 $N(0.01, 0.002)$ 正态分布, 而在后 2 个大电路中, 我们适当调低了每个节点的电流变化, 设为 $N(0.001, 0.0002)$ 正态分布.

表 1 实验电路的具体参数

| 序号 | N_2 | N_1 | 电路规模 | Block 数 | 细线电导 | 线宽比 | 粗线电导 | 节点吸纳电流分布 | |
|----|-------|-------|---------|---------|------|-----|------|----------|--------|
| | | | | | | | | 期望值 | 均方差 |
| 1 | 30 | 4 | 13680 | 900 | 10 | 3 | 30 | 0.01 | 0.002 |
| 2 | 30 | 10 | 89640 | 900 | 10 | 3 | 30 | 0.01 | 0.002 |
| 3 | 30 | 15 | 176280 | 900 | 10 | 4 | 40 | 0.01 | 0.002 |
| 4 | 30 | 20 | 360240 | 900 | 10 | 5 | 50 | 0.01 | 0.002 |
| 5 | 30 | 25 | 518880 | 900 | 10 | 6 | 60 | 0.01 | 0.002 |
| 6 | 30 | 30 | 810840 | 900 | 10 | 7 | 70 | 0.01 | 0.002 |
| 7 | 30 | 35 | 1041480 | 900 | 10 | 8 | 80 | 0.001 | 0.0002 |
| 8 | 30 | 40 | 1441440 | 900 | 10 | 9 | 90 | 0.001 | 0.0002 |

5.2 实验方案设计

本文方法用标准的 C 语言书写,硬件平台为 T5500 (1.66GHz 双核 CPU)-1024M 内存的笔记本电脑,本文程序只用一个 CPU 来执行,另一个 CPU 可以应付 Windows 的后台任务,以保证程序运行时间的稳定性。

为了验证本文方法的有效性,我们采用如下 5 种方法进行比较,主要进行电路最大 IR 电压降 (V_{ust}) 节点上的电压变化均方差、和运行时间这两个方面的比较。其中第 2~5 个算法均为直接求解方法,整个求解过程运行时间为 T_{Solv} (单位为秒),包括两部分时间,全部节点电压变化期望值的求解时间 T_{mean} , 和问题节点电压变化均方差的求解时间 T_{std} , T_{std} 的主要组成部分是相关电阻向量的求解时间 T_R 。对于具有 900 个 Block 的实验电路,本文取 900 个问题节点,即每个 Block 中的最大 IR 电压降节点算作一个问题节点。

(1)5000 次的模拟算法^[9]。采用全局 SOR 算法模拟 5000 次,整个模拟过程运行时间为 T_{sim} (单位为秒)。

(2)全局 SOR 求解算法^[6]。先用全局 SOR 算法求出全部节点电压变化期望值,找出问题节点,为了便于 IC 设计中的多次分析,必须搜索最佳 SOR 松弛系数 B_{opt} ;再用全局 SOR 算法求出问题节点的相关电阻向量,并根据相关电阻向量计算出问题节点的电压变化,由于此过程的 B_{opt} 和求期望值的 B_{opt} 略有不同,也必须进行搜索。

(3)全局 ICCG 求解算法^[2]。先用全局 ICCG 算法求出全部节点电压变化期望值,再用全局 ICCG 算法找出问题节点的相关电阻向量,并根据相关电阻向量计算出问题节点的电压变化。

(4)9Block 局部 ICCG 求解算法。先用全局 ICCG 算法求出全部节点电压变化期望值,再用 9Block 局部 ICCG 算法找出问题节点的相关电阻向量,并根据相关电阻向量计算出问题节点的电压变化。

(5)SN-SOR 求解算法。先用全局 SOR 算法求出全部节点电压变化期望值,找出问题节点,再用 SN-SOR 算法找出问题节

点的相关电阻向量,并根据相关电阻向量计算出问题节点的电压变化,此过程中也必须搜索 SN-SOR 算法的 B_{opt} 。

5.3 实验数据与分析

我们分别用本文方法和 4 个对比方法在实用电路上进行实验,当一种方法的运行时间超过 10800 秒(即 3 个小时)后,就不再用此方法进行更大电路的实验,所有实验数据被放在表 2 至表 4 中。

首先在表 2 中,我们比较不同方法的运行速度。在第 2 列中,我们可以清楚地看到 5000 次 SOR 模拟方法是最耗时的,只能对前 3 个小规模电路进行分析,对于更大的电路,由于太耗时,所以无模拟结果(NA)。对于第 3 个电路,尽管只有 176K 个节点,模拟方法要耗时 20388 秒(5.6 个小时),因此模拟方法无法直接应用到 P/G 网统计分析中,在本文中,它可以在表 3 中,作为评判其它方法分析精度的依据。在第 3 列和第 5 列中,分别列出全局 SOR 求解算法和全局 ICCG 求解算法的运行时间,我们可以看出这两种算法的分析效率较高,可以分析较大的电路(第 4 和第 5 个电路),但对于更大的电路(后 3 个电路),还是比较耗时,因此对于后 3 个电路,这两种方法无模拟结果(NA)。在第 7 列和第 9 列中,分别列出 SN-SOR 求解算法和 9Block 局部 ICCG 求解算法的运行时间,我们可以看出这两种算法的分析效率非常高,可以分析所有电路,因此这两种算法均可以分析大电路,比较而言,9Block 局部 ICCG 求解算法的分析效率更高些。

对于前 3 个电路,本文用模拟方法来计算其它算法的加速比,分别放在第 4、6、8、10 列中,以第 3 个电路为例,全局 SOR 求解算法和全局 ICCG 求解算法的加速比分别为 6.9 和 7.92,SN-SOR 求解算法和 9Block 局部 ICCG 求解算法的加速比分别为 108 和 773,这表明 SN-SOR 求解算法和 9Block 局部 ICCG 求解算法比模拟算法快 2 个数量级。对于中间 2 个电路,本文用全局 SOR 求解算法来计算其它求解算法的加速比,分别放在第 6、8、10 列中,以第 5 个电路为例,全局 ICCG 求解算法的

表 2 不同方法之间的效率比较

| 序号 | 模拟方法运行时间 | 全局 SOR | | 全局 ICCG | | SN-SOR | | 9Block 局部 ICCG | |
|----|----------|----------|------|----------|------|---------|--------|----------------|--------|
| | | 运行时间 | 加速比 | 运行时间 | 加速比 | 运行时间 | 加速比 | 运行时间 | 加速比 |
| 1 | 283.24 | 45.00 | 6.29 | 122.36 | 2.31 | 11.02 | 25.71 | 2.44 | 116.22 |
| 2 | 6141.33 | 1053.49 | 5.83 | 856.84 | 7.17 | 75.19 | 81.68 | 11.77 | 521.96 |
| 3 | 25130.09 | 3643.31 | 6.90 | 3173.30 | 7.92 | 230.78 | 108.89 | 32.48 | 773.61 |
| 4 | NA | 9674.59 | Base | 7490.86 | 1.29 | 495.66 | 19.52 | 71.22 | 135.83 |
| 5 | NA | 20388.97 | Base | 14453.09 | 1.41 | 950.38 | 21.45 | 122.22 | 166.82 |
| 6 | NA | NA | NA | NA | NA | 1566.39 | Base | 183.41 | 8.54 |
| 7 | NA | NA | NA | NA | NA | 2395.02 | Base | 281.77 | 8.50 |
| 8 | NA | NA | NA | NA | NA | 4019.77 | Base | 397.80 | 10.11 |

表 3 V_{test} 节点电压变化均方差之间的方法比较

| 序号 | 模拟方法 | 全局 SOR | | 全局 ICCG | | SN-SOR | | 9Block 局部 ICCG | |
|----|----------|----------|-------|----------|-------|----------|-------|----------------|-------|
| | | 均方差 | 误差 | 均方差 | 误差 | 均方差 | 误差 | 均方差 | 误差 |
| 1 | 0.000514 | 0.000514 | 0.00% | 0.000514 | 0.00% | 0.000514 | 0.00% | 0.000509 | 0.97% |
| 2 | 0.004754 | 0.004754 | 0.00% | 0.004754 | 0.00% | 0.004753 | 0.02% | 0.004620 | 2.82% |
| 3 | 0.010828 | 0.010828 | 0.00% | 0.010828 | 0.00% | 0.010825 | 0.03% | 0.010507 | 2.96% |
| 4 | NA | 0.019410 | Base | 0.019410 | Base | 0.019393 | 0.09% | 0.018826 | 3.01% |
| 5 | NA | 0.030393 | Base | 0.030393 | Base | 0.030278 | 0.38% | 0.029471 | 3.03% |
| 6 | NA | NA | NA | NA | NA | 0.041013 | Base | 0.039936 | 2.63% |
| 7 | NA | NA | NA | NA | NA | 0.005347 | Base | 0.005222 | 2.34% |
| 8 | NA | NA | NA | NA | NA | 0.007410 | Base | 0.007217 | 2.60% |

表 4 SN-SOR 方法与全局 SOR 方法之间在运行效率方面的具体比较

| 序号 | 电压变化期望值的求解 | | | 全局 SOR 方法 | | | SN-SOR 方法 | | | | | |
|----|-------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|--------|--------|-------|
| | T_{mean} | B_{opt} | N_{SOR} | T_{std} | B_{opt} | N_{SOR} | T_{std} | B_{opt} | N_{SOR} | 松弛点 | 点数比 | 加速比 |
| 1 | 1.1 | 1.400 | 17 | 43.9 | 1.390 | 15 | 9.92 | 1.400 | 15 | 1409 | 10.30% | 4.43 |
| 2 | 22.5 | 1.759 | 53 | 1032.0 | 1.760 | 47 | 52.73 | 1.739 | 44 | 12301 | 13.72% | 19.57 |
| 3 | 73.9 | 1.831 | 83 | 3572.6 | 1.819 | 64 | 156.92 | 1.820 | 64 | 24885 | 14.12% | 22.77 |
| 4 | 187.5 | 1.871 | 112 | 9483.4 | 1.860 | 86 | 308.13 | 1.861 | 83 | 40129 | 11.14% | 30.78 |
| 5 | 343.7 | 1.890 | 141 | 20059.6 | 1.890 | 114 | 606.72 | 1.880 | 104 | 53458 | 10.30% | 33.06 |
| 6 | 661.8 | 1.909 | 168 | NA | NA | NA | 904.63 | 1.899 | 119 | 70033 | 8.64% | NA |
| 7 | 1120.8 | 1.923 | 167 | NA | NA | NA | 1274.23 | 1.912 | 137 | 87000 | 8.35% | NA |
| 8 | 2278.5 | 1.929 | 188 | NA | NA | NA | 1741.31 | 1.929 | 166 | 124025 | 8.60% | NA |

加速比仅为 1.41, 这表明全局 SOR 求解算法和全局 ICCG 求解算法的效率差不多, 而 SN-SOR 求解算法和 9Block 局部 ICCG 求解算法的加速比分别为 21 和 166, 这表明它们比全局 SOR 求解算法快 1 个和 2 个数量级. 对于后 3 个大电路, 本文用 SN-SOR 求解算法来计算 9Block 局部 ICCG 求解算法的加速比, 放在第 10 列中, 以第 8 个电路为例, 9Block 局部 ICCG 求解算法的加速比为 10, 这表明 9Block 局部 ICCG 求解算法比 SN-SOR 求解算法快 1 个数量级.

表 3 列出了所有算法 V_{test} 节点电压变化均方差的计算值(第 2、3、5、7、9 列)和计算误差(第 4、6、8、10 列). 由于分析效率的影响, 模拟方法仅对前 3 个小电路有计算结果, 全局 SOR 求解算法和全局 ICCG 求解算法也只有前 5 个电路的计算结果. 对于前 3 个小电路, 本文用模拟方法来分析其它算法的分析精度, 发现全局 SOR 求解算法和全局 ICCG 求解算法无计算误差, SN-SOR 求解算法的分析精度也非常高, 误差仅为 0.03%, 而 9Block 局部 ICCG 求解算法的计算误差比较大, 达到了 2.96%. 对于中间 2 个中规模电路, 本文用全局 SOR 求解算法和全局 ICCG 求解算法来分析后两个局部算法的分析精度, 发现 SN-SOR 求解算法的分析精度比较高, 误差仅为 0.38%, 而 9Block 局部 ICCG 求解算法的计算误差比较大, 达到了 3.03%. 对于后 3 个大电路, 本文用 SN-SOR 求解算法来分析 9Block 局部 ICCG 求解算法的分析精度, 发现 9Block 局部 ICCG 求解算法的计算误差还是比较大, 达到了 2.63%.

结合表 2 和表 3 的分析结果, 我们可以得出如下的结论: 对于模拟方法、全局 SOR 求解算法、全局 ICCG 求解算法、SN-SOR 求解算法、和 9Block 局部 ICCG 这 5 种方法, 它们的分析效率依次提高, 同时分析精度依次降低. 由于分析效率比较低, 前 3 种方法不能用于实际大规模 P/G 网的统计分析; 而对于后 2 种方法, 由于 9Block 局部 ICCG 求解算法的计算误差相对比较大, 所以只有本文的 SN-SOR 求解算法、能够做到算法效率与精度的平衡, 对实际大规模 P/G 网进行高效而精确地统计分析.

鉴于本文算法源于 SOR 算法^[6], 我们进一步对全局 SOR 求解算法和本文的 SN-SOR 求解算法进行详细地比较, 表 4 给出了比较数据. 由于本文方法用全局 SOR 算法来计算 P/G 网电压变化的期望值, 所以我们可以对全部节点均有激励的全局 SOR 算法、单点激励的全局 SOR 算法、和单点激励的 SN-SOR 算法进行比较. 从最佳松弛系数 B_{opt} 来看, 三种 SOR 算法的 B_{opt} 近似相等(第 3、6、9 列), 但全部节点均有激励的全局 SOR 算法 B_{opt} 最大(第 3 列). 在 3 种 SOR 算法均采用 B_{opt} 的情况下, 全部节点均有激励的全局 SOR 算法的松弛次数 N_{SOR} 最大(第 4 列), SN-SOR 算法 N_{SOR} 最小(第 10 列). 由于 SN-SOR 算法的松弛节点(第 11 列)占全部节点数的比例比较小(第 12 列), 其范围在 8~15% 之间, 以及本文 SN-SOR 算法的各种加速策略, 在计算 900 个问题节点电压变化均方差的运行时间 T_{std} 方面, SN-SOR 方法(第 5 列)是大大快于全局 SOR 算法(第 9 列), 最大加速比(第 13 列的第 5 个电路)为 33 倍, 而对于 1.44M 的第 8 个电路, SN-SOR 方法的 T_{std} 仅为 1741.31 秒, 约为半个小时, 速度非常快, 能够用于大规模实际 P/G 网电路电压变化的统计分析.

6 结论

纳米工艺产生较大的工艺参数片内变化, 使 P/G 网分析的复杂度大大提高. 针对空间相关的吸纳电流源变化, 本文提出了一种高效的单点 SOR 统计分析方法(SN-SOR), 它可以快速地求解 P/G 中一个节点的相关电阻向量, 并对相关电阻向量进行压缩, 进而快速计算出此节点电压变化的均方差. 因此, 本文的 SN-SOR 求解方法、可以快速对 P/G 网中一些 IR 电压降较大的问题节点进行高效地统计分析. 大量的实验结果表明: 与模拟方法相比, SN-SOR 求解方法在保证 0.03% 误差

的前提下,其加速比可以达到 2 个数量级;与全局 SOR 求解方法相比,SN-SOR 求解方法在保证 0.38% 误差的前提下,其加速比可以达到 1 个数量级(21 倍).因此,本文 SN-SOR 方法可以快速而精确地计算出 P/G 网少数问题节点的电压变化.

参考文献:

- [1] 骆祖莹. 芯片功耗与工艺参数变化:下一代集成电路设计的两大挑战[J]. 计算机学报,2007,30(7):1054-1063.
Luo Z Y. Power consumption and process variations: two challenges to design of next-generation ICs[J]. Chinese Journal of Computers,2007,30(7):1054-1063. (in Chinese)
- [2] Wu X H, et al. Area minimization of power distribution network using efficient nonlinear programming techniques [J]. IEEE Trans on CAD,2004,23(7):1086-1094.
- [3] Luo Z Y. Efficient optimization methodology in early-stage design of mesh-structured on-chip power/ground(P/G) Networks [A]. Proceedings of IEEE International Conference on Solid-state and Integrated Circuit Technology [C]. Piscataway, NJ, USA: IEEE Press,2006. 1911-1913.
- [4] Luo Z Y, et al. Time-domain analysis methodology for large-scale RLC circuits and Its applications[J]. Science in China F Series,2006,49(5):665-680.
- [5] Qian H F, et al. Random walks in a supply network[A]. Proceedings of IEEE/ACM Design Automation Conference [C]. New York, NY, USA: ACM Press,2003. 93-98.
- [6] Zhong Y, et al. Fast algorithms for IR drop analysis in large power grid[A]. Proceedings of IEEE/ACM International Conference on Computer Aided Design [C]. New York, NY, USA: ACM Press,2005. 351-357.
- [7] Ferzli I A, et al. Statistical Verification of Power Grids Considering Process-Induced Leakage Current Variations [A]. Proceedings of IEEE/ACM International Conference on Computer Aided Design [C]. New York, NY, USA: ACM Press,2003.

770-777.

- [8] Kouroussis D, et al. Incremental Partitioning-Based Vectorless Power Grid Verification [A]. Proceedings of IEEE/ACM International Conference on Computer Aided Design [C]. New York, NY, USA: ACM Press,2005. 358-364.
- [9] Li P. Variational analysis of large power grids by exploring statistical sampling sharing and spatial locality [A]. Proceedings of IEEE/ACM International Conference on Computer Aided Design [C]. New York, NY, USA: ACM Press,2005. 644-650.
- [10] Samaan S B. The impact of device parameter variation on the frequency and performance of VLSI chips [A]. Proceedings of IEEE/ACM International Conference on Computer Aided Design [C]. New York, NY, USA: ACM Press,2004. 343-346.

作者简介:



骆祖莹 男,北京师范大学信息科学与技术学院,博士,副教授.主要研究方向为低功耗 IC 设计和 IC 物理设计,已发表论文 50 余篇.
E-mail:luozy@bnu.edu.cn



潘月斗 男,北京科技大学信息学院自动化系,博士,副教授.主要研究方向为计算机辅助设计及计算机数字控制.

余先川 男,北京师范大学信息科学与技术学院,博士,教授.主要研究方向为计算机体系结构与信息处理,已发表论文 70 余篇.