

1、封面

采用学校统一印制的学位论文封面。

上海交通大学硕士学位论文

符号化矩量与敏感度方法用于抗串扰噪声布线¹

学 校： 上海交通大学
院 系： 微电子学院
专 业： 电路与系统

上海交通大学微电子学院

2009 年 12 月

¹此研究部分由国家自然科学基金（编号 60876089）和上海市浦江人才基金（编号 07pj14053）资助

**A Thesis Submitted to Shanghai Jiao Tong University for the Degree
of Philosophy Master**

**Symbolic Moment and Moment Sensitivity
Computation Method With Application to Crosstalk
Avoidance in Routing**

School: Shanghai Jiao Tong University

College: Microelectronic Department

Specialty: Circuits and System

School of Microelectronics
Shanghai Jiao Tong University
Shanghai, P. R. China

December, 2009

符号化矩量与敏感度方法用于抗串扰噪声布线

摘要

随着集成电路互联线横向尺寸的不断缩小, 互联线之间的寄生耦合效应越来越明显, 对芯片信号完整性提出了更高的要求, 与此同时, 芯片的面积, 功耗等因素也直接影响着芯片的性能和可靠性, 在设计时必须综合进行考虑。在导线分布效应不能被忽略的情况下, 线宽调整已经被证实有很好的效果, 但是由于运算复杂度的限制, 之前的线宽优化算法在估计导线信号延迟和不同线网之间的串扰时都采用了非常简单的数学模型。

本文提出了一种符号化矩敏感度计算方法, 只需要遍历一次符号化表达式树, 就能够计算出电路中各节点的加权矩关于电路中所有 R/C 元件的敏感度。符号化矩敏感度计算具有可重复性强、运算效率高的特点。本文将符号化矩敏感度计算方法应用到拉格朗日松弛法求解优化问题的过程中, 对更为复杂精确的物理优化模型进行求解。该物理优化模型采用 $D2M$ 作为节点延迟的估计, 并且在估计串扰时考虑了干扰线网上信号跳跃对受干扰线网信号延迟影响的最坏情况。结果表明, 该模型相对于传统的优化模型, 有很高的精确度和可靠性。

关键词:

符号化、矩敏感度、线宽调整、串扰、拉格朗日松弛法

Symbolic Moment and Moment Sensitivity Computation Method With Application to Crosstalk Avoidance in Routing

Abstract

As the VLSI feature size scaled down, the crosstalk and coupling effects between Interconnects become more and more significant and the chip is facing a serious signal integrity problem. At the same time, die size and power dissipation also directly affects the performance and reliability of the chip. Wire sizing has been proved been a good method on physical optimization. Due to high computation complexity, most traditional wire-sizing algorithm use simple mathematical model to estimate the signal delay and the effect of coupling.

This paper proposes a symbolic moment sensitivity computation method, which can be used to compute moment sensitivities versus all RC elements in the circuit by traverse the symbolic representation only once. Symbolic Moment and Moment Sensitivity evaluation method has the advantage of high efficiency and repeatability, which was applied to solve a more precise physical optimization model by the Author. The optimization model uses D2M as the metric of signal delay, account the worst case that aggressor can affect the victim when coupling is considered. The experiment result shows that such a model has higher fidelity when compared to traditional wire-sizing models.

Keywords:

Symbolic、Moment Sensitivity、Wire-Sizing、crosstalk、Lagrangian Relaxation

目 录

摘要.....	I
ABSTRACT.....	II
图片目录.....	V
第 1 章 绪论.....	1
1.1 引言.....	1
1.2 互连线模型及研究.....	2
1.2.1 互连线的传输线模型.....	2
1.2.2 互联线的集总 RC 模型.....	3
1.2.3 互联线的 RLC 模型.....	5
1.2.4 工艺尺寸缩小对互联线参数的影响.....	5
1.3 互联线上的信号完整性.....	6
1.3.1 互联线的串扰估计算法.....	7
1.3.2 互联线的信号延迟.....	13
1.3.3 互联线的延迟估计.....	15
1.3.4 符号化矩计算.....	16
1.4 以互联线为中心的设计流程.....	18
1.5 VLSI 物理设计中的优化.....	20
1.6 VLSI 物理优化算法.....	22
1.7 本文的主要研究内容及安排.....	23
第 2 章 符号化矩敏感度计算.....	25
2.1 矩敏感度介绍.....	25
2.2 符号化加权矩敏感度计算.....	26
2.2.1 矩敏感度计算的递归表达式推导.....	27
2.2.2 关于电阻元件的矩敏感度计算的符号化表示.....	28
2.2.3 关于电容元件的矩敏感度计算的符号化表示.....	32
2.3 符号化矩敏感度计算的复杂度分析.....	33
2.4 符号化敏感度计算的正确性测试.....	34
2.4.1 小规模电路测试.....	34

2.4.2 敏感度与极值点的分析.....	35
2.4.3 大规模电路测试.....	36
第3章 基于符号化计算的物理优化.....	38
3.1 物理优化的数学模型.....	38
3.1.1 优化问题的提出.....	38
3.1.2 带耦合串扰的信号延迟估计.....	39
3.1.3 优化问题的数学表示.....	40
3.2 拉格朗日松弛法求解优化问题.....	41
3.3 拉格朗日问题的求解.....	42
3.4 原优化问题的解.....	44
第4章 优化结果的测试与分析.....	46
4.1 测试分析方法.....	46
4.2 小测试案例的优化结果分析.....	47
4.3 大测试案例的优化结果分析.....	49
4.3.1 优化所需时间分析.....	50
4.3.2 优化前后节点延迟的比较.....	51
4.3.3 最大延迟上限和优化面积的关系.....	53
4.3.4 考虑串扰与不考虑串扰的比较.....	54
4.3.5 基于Elmore估计和D2M估计的比较.....	56
4.4 本章小结.....	58
第5章 总结.....	59
参考文献.....	60
附录一 拉格朗日松弛法.....	63
致谢.....	64
攻读硕士学位期间已发表或录用的论文.....	65

图片目录

图 1	传输线模型	2
图 2	单根导线的电气模型	3
图 3	耦合导线的电气模型	4
图 4	互联线的集总 RLC 模型	5
图 5	互联线的平板电容模型	6
图 6	两根相邻导线的传输线模型	7
图 7	电路约简规则	10
图 8	RC 电路分支的约简	10
图 9	静止攻击线网的约简	11
图 10	考虑电容耦合的串扰模型	12
图 11	互联线串扰对信号延迟影响的示意电路	13
图 12	相邻线网不同的信号跳变方向	14
图 13	受干扰线网节点延迟随信号跳变时间差的变化	14
图 14	树状 RC 电路	16
图 15	电容树	17
图 16	用于计算一阶矩的二叉决策树	17
图 17	VLSI 设计流程	19
图 18	以互联线为中心的设计流程	19
图 19	Elmore 延迟估计和 D2M 延迟估计精度的比较	23
图 20	矩敏感度的应用例子	25
图 21	矩敏感度计算公式的分解	27
图 22	用于说明符号化表示的示例电路	29
图 23	增量部分计算的符号化表示	29
图 24	增量部分计算示例	30
图 25	新权值计算的符号化表示	30
图 26	新权值计算的符号化表达式示例	31
图 27	n 阶加权矩敏感度的计算过程	31
图 28	电容敏感度计算增量部分的符号化表示	32
图 29	电容元件矩敏感度增量部分计算示例	33

图 30	小规模互联线例子	34
图 31	符号化矩敏感度计算方法和数值计算方法的比较	35
图 32	极值点的分析	36
图 33	节点规模和计算时间的关系	37
图 34	阶数和计算时间的关系	37
图 35	耦合情况下节点波形的叠加计算	39
图 36	拉格朗日松弛法求解过程	42
图 37	将布线结果转化成 RC 树	46
图 38	互相耦合的小规模布线树	47
图 39	优化前后节点最坏延迟的比较	48
图 40	优化前后面积的比较	48
图 41	互相耦合的大规模布线树	49
图 42	其他可能的测试案例	50
图 43	测试案例的 RC 模型	50
图 44	不同电路规模的优化时间	51
图 45	优化前后节点延迟的分布图	52
图 46	优化面积和延迟上限的关系	54
图 47	是否考虑相邻线网耦合对优化结果的影响	55
图 48	Elmore 延迟优化和 D2M 延迟优化的结果比较	57

表格目录

表 1	电路约简公式	10
表 2	优化前后的面积比较	53
表 3	是否考虑串扰对优化结果面积的影响	55
表 4	采用 Elmore 延迟估计和采用 D2M 延迟估计所得到的优化面积比较	57

第1章 绪论

1.1 引言

随着超大规模集成电路（VLSI）技术的飞速发展，芯片的特征尺寸越来越小，工作频率越来越高。这导致芯片上互连复杂度急剧增加，VLSI设计从以门设计为中心转变到以互连线设计为中心。这种情况下，VLSI设计开始面临许多新的挑战。

首先我们知道，当工艺尺寸逐渐进入深亚微米后，导线的横向尺寸不断减小，而导线的纵向尺寸却几乎维持不变。这种情况下，导线之间的耦合电容达到可以和导线本身电容相比拟的程度，耦合电容的存在会对互联线的信号延迟和信号完整性产生很大的影响，如果不对耦合电容的效果充分加以考虑，很有可能导致芯片设计的失败。这就使得原来单独考虑各个布线线网的仿真方法已经不再使用，在仿真和信号估计时不但要考虑当前线网的信号变化，还要考虑与当前线网相邻线网上的信号变化。

由于现代工艺下VLSI芯片的规模非常大，互联线上信号估计和优化都极具挑战。集成电路工艺尺寸的缩小导致单个芯片上的器件数量急剧增加，器件数量的增加导致互联线复杂度的增加，单个线网的节点数量可能达到数十万个，并且线网的长度也随着布线复杂度的增加而增长。设计复杂度的提高增加了仿真的规模，利用传统的仿真工具如SPICE，无论是在仿真时间还是在存储空间开销方面都已经不能满足需要了，更别提互联线的优化。

除了芯片设计复杂度的不断增加外，芯片工作频率也越来越高。这也对时序估计的精确度提出了更高的要求，如果过小的估计延时，可能会导致设计出来的芯片实际上没有满足时序条件，导致设计的失败；如果过大的估计延时，可能会导致设计出来的芯片没有最大程度的利用芯片上的资源，导致成本的增加。工作频率的提高也使节点延迟的冗余量变小，增加了芯片设计过程中时序约束的难度。

工作频率的提高和芯片规模的变大必然导致芯片功耗的增加。功耗已经成为芯片设计过程中一个非常棘手的问题。虽然大部分芯片在体系设计过程中会加入功耗控制单元，通过关闭某些不用的功能模块来节省功耗。但是这种方法无法解决芯片在满负荷工作情况下的功耗问题。为了最大程度的节省功耗，需要在芯片设计的各个阶段对功耗进行优化：在系统级设计时对芯片功能结构进行优化，减少不必要的功耗开支；在物理设计阶段对布局布线进行优化，尽量减小电路的负载，从而节省功耗。

芯片的成本也是设计过程中需要考虑的重要问题，芯片的制造成本和芯片的面积密切相关，面积越小，芯片成本越小。

芯片设计需要综合时序，功耗，面积等各方面的因素进行考虑，才能设计出低成本，高性能的芯片。如何在物理设计层面优化设计在相当长的一段时间内都是学术界研究的热点问题，在这方面有一些不少的成果，但是多多少少也存在着一些不足。因此研究更加精确的优化模型和更加高效的优化算法是非常必要的。

1.2 互连线模型及研究

随着器件尺寸的缩小，逻辑门电路速度随之提高，与此同时片上互连线却日趋复杂，互连线上的延迟已经成为片上延迟的主导部分。因此深入研究分析半导体工艺中互连线的电气特性及其对芯片功能的影响已经非常必要。对于互连线的电气特性，有很多文献做过相应的研究，归结起来，互连线的电气模型可以分为两大类：传输线模型和集总模型。传输线模型考虑了互连线电气特性的分布效应；集总模型采用电阻/电容等集总元件来近似互连线原有的电气特性，例如互连线的 RC 模型，RLC 模型等都属于集总模型。传输线模型和集总模型各有所长，也各有所短，下面分别简要介绍。

1.2.1 互连线的传输线模型

当互连线上信号的上升和下降时间变得足够快，可与信号波形在互连线的传输时间相比拟时，这时必须考虑互连线的传输线效应。传输线的基本性质是信号以波的形式传播通过介质。传输线中信号的传播是通过交替地使能量在电场与磁场之间转变来实现的。

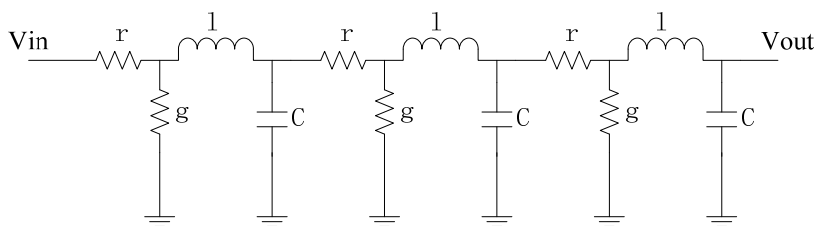


图 1 传输线模型

Figure 1 Transmission Line Model

传输线的模型如图 1 所示，一根互连线段可以分成很多小段，每一小段我们都用电阻，电感和电容来等效。其中 r 表示互连线每小段上的电阻， l 为互连线每

小段上的电感， g 为互联线每小段上对地的漏导， c 为互联线每小段上的电容。我们对图 1 中的电路模型列出基尔霍夫方程：

$$\frac{\partial v(x,t)}{\partial x} = -ri(x,t) - l \frac{\partial i(x,t)}{\partial t} \quad (1)$$

$$\frac{\partial i(x,t)}{\partial x} = -gv(x,t) - c \frac{\partial v(x,t)}{\partial t} \quad (2)$$

其中 $i(x,t)$ 和 $v(x,t)$ 表示在时间 t 时，传输线上某一点 x 的电流和电压。对于大部分传输线，漏导可以忽略（即 $g=0$ ），因此(1)(2)两式可以化为：

$$\frac{\partial v^2(x,t)}{\partial x^2} = rcv(x,t) + lc \frac{\partial i(x,t)}{\partial t^2} \quad (3)$$

在实际的应用过程中，求解上述传输线方程是非常复杂的。尤其是对于比较复杂的互联电路，精确的求解方程(3)几乎是不可能的，因此在应用中，通常需要进行进一步的简化。

1.2.2 互联线的集总 RC 模型

利用传输线模型来分析信号在互联线上的传输有很高的精确度，但是精确的求解麦克斯韦方程是非常复杂的，特别是在分析大规模互联电路时，采用传输线的分析方法是不可行的，我们需要有更简单的模型来刻画互联线上的信号。

目前比较广泛采用的是集总 RC 模型，该模型将某一段互联线等效成两个电容和一个电阻，因而大大简化了模型。如图 2 所示，

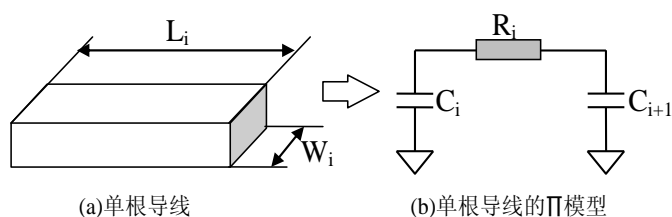


图 2 单根导线的电气模型

Figure 2 Electrical Model for Single Conductor

其中 R_i 为互联线的等效电阻， C_i 和 C_{i+1} 为互联线的等效电容。模型中电阻和电容的计算公式如式(4)和(5)所示：

$$R_i = \rho \frac{l_i}{w_i} \quad (4)$$

$$C_{i-1} = C_i = \frac{1}{2}(c_0 l_i w_i + c_f l_i) \quad (5)$$

式(4)中 ρ 为电阻率, l_i 为该段互联线的长度, w_i 为互联线的宽度。式(5)中 c_0 为单位面积电容, c_f 为边缘电容。导线的寄生电容值由 C_1 和 C_2 平分。 ρ 、 c_0 和 c_f 的取值和导线的介质、尺寸等有关。

导线除了有相对于衬底的电容之外, 导线与导线之间也存在着耦合效应。RC 模型还可以刻画相邻导线之间的耦合效应。其耦合电容的值可以用平板电容来近似。

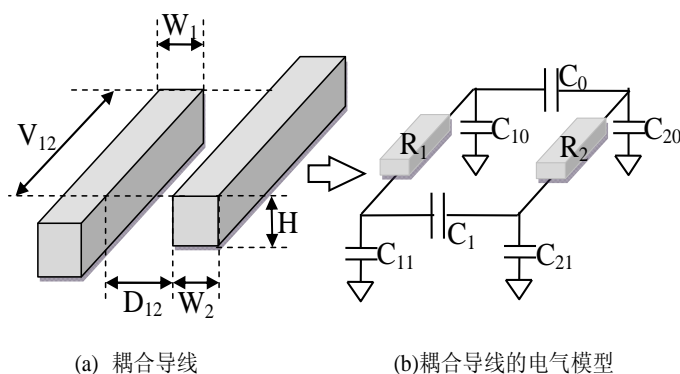


图 3 耦合导线的电气模型

Figure 3 Electrical Model for Coupling

在图 3 (a)中, 两根平行导线的重叠长度为 V_{12} , 导线的距离为 D_{12} , 导线高度为 H , 该导线的电气特性可以近似和图 3 (b)中的 RC 电路等效。其中 C_{10} , C_{11} , C_{21} 和 C_{20} 为导线和衬底之间的电容, C_0 和 C_1 为两根导线之间的耦合电容, 其近似值可以由下面式子计算得出。

$$C_1 = C_0 = \frac{1}{2} C_{couple} = \frac{\epsilon H V_{12}}{8\pi k D_{12}} \quad (6)$$

式(6)中 k 和 ϵ 为常数, C_{couple} 为两根平行导线总的耦合电容。 C_0 和 C_1 平分总耦合电容 C_{couple} 的值。

利用互联线集总 RC 模型, 可以比较直观的刻画互联线的寄生电气效应, 并且集总模型的求解复杂度也相对不高, 因此集总 RC 模型一度成为 EDA 工具中互联线建模的主流。

1.2.3 互联线的 RLC 模型

当互联线上信号的翻转速度足够快，并且互联线的电阻保持在一定范围内时，导线上的电感效应将越来越明显。这时就应该采用 RLC 模型来估计由电感效应引起的延迟、震荡和过冲。

互联线的 RLC 模型如图 4 所示，一根互联线被分成若干段，每段的电气效应用一个电阻、一个电容和一个电感来等效。同样，在 RLC 树模型的基础上，我们还可以考虑耦合电容，互感等效效应，对互连进行更加精确的建模。但同时电路的复杂性也随之提高。

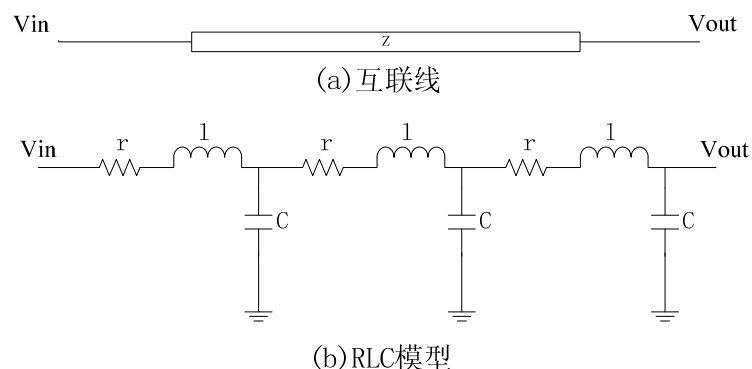


图 4 互联线的集总 RLC 模型
Figure 4 RLC Model for Interconnects

相对于传输线模型，RC 模型和 RLC 模型对互联线的电气效应做了更大程度的近似，因此模型精度上略低与传输线模型。但是大量的实验数据表明，RC 模型的精度已经能够满足大部分情况下的互联线电气效应估计。

1.2.4 工艺尺寸缩小对互联线参数的影响

随着工艺尺寸的变小，单位面积内容纳的器件数量不断增加。导致器件之间的互联线拥挤度也随之增加，因此互联线的尺寸也必须随之变小。互联线尺寸的变小对互联线的电气特性产生了各种影响，下面分别加以分析。

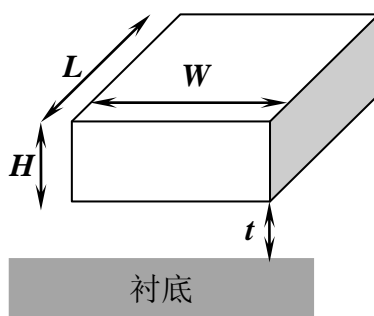


图 5 互联线的平板电容模型

Figure 5 Plate Capacitor Model for Interconnects

如图 5 所示，和互联线尺寸有关的参数有 L 、 W 、 H 、 t ，他们分别是互联线的长度、宽度、高度和电介质厚度。在工艺尺寸的缩小过程中， W 的值会随之按比例缩小；而 H 、 t 的值会略微缩小或者基本保持不变；互联线长度 L 会随着布线复杂度增加而增长。因此当工艺尺寸缩小时：

1. 导线的等效电阻增加。根据电阻的计算公式 $R=\rho L/S$ 可知，导线的横截面积减小，而导线长度增加，因此导线的等效电阻必然增加。

2. 导线电容基本保持不变。导线的横向面积减小，但是导线长度增加，导致导线和衬底之间的重合面积并没有减少很多，同时介质厚度 t 也基本保持不变。因此导线相对于衬底的电容基本保持不变。

3. 导线的耦合电容增加。工艺尺寸的变小导致相邻导线之间的距离越来越近；导线长度增加，而高度基本保持不变导致导线之间的重合面积增加，因此导线之间的耦合电容增加。

综合上面三方面的影响，随着工艺尺寸的缩小，线上延迟会变得越来越重要，并且耦合电容对互联线的影响也会大大增加。

1.3 互联线上的信号完整性

随着集成电路的制造工艺向深亚微米领域发展，器件的特征尺寸越来越小，互连线间的耦合电容、互感对互联线信号的影响越来越大，这使得实际互联线上的信号波形会偏离所期望的状态，严重的情况下会导致数字芯片的逻辑错误。互联线的信号完整性导致的问题大致可以分为以下三种情况：

- 时序问题。在同步时序电路中，我们需要对信号的到达时间进行严格的控制。如果某点的信号没有满足时序约束，则会出现时序失效，导致芯片的功能性错误。

- 噪声。噪声表现为毛刺的出现，毛刺会导致逻辑误判，从而影响电路的功能。
- 电磁干扰（EMI）。芯片内部的工作时钟越来越高，并且互联线复杂度和拥挤度的增加导致芯片内部的电磁环境非常恶劣。电磁干扰会造成较长互联线上的信号扰动，震荡，严重的情况下会导致逻辑误判。

1.3.1 互联线的串扰估计算法

在理想的状态下，不同线网上的信号应该是互不影响的。但是在实际情况中，随着互联线之间耦合效应的增加，不同的线网之间会产生信号的串扰。即一个线网上信号的跳变会耦合到另外一个线网上，在另外一个线网上形成毛刺或者扰动。

基于传输线理论的串扰估计

采用传输线模型来估计互联线上的串扰有非常高的精确度，但是由于准确求解传输线模型的难度非常大，一般在模型求解时需要进行简化和近似，从而降低模型求解的复杂度。

[17]提出了一种采用传输线模型来估计串扰的方法，假设有两根相邻的导线如图 6 所示。其中 R_s 为攻击线网的驱动电阻， R_v 为受干扰线网的驱动电阻，两根导线之间存在互感和电容耦合。

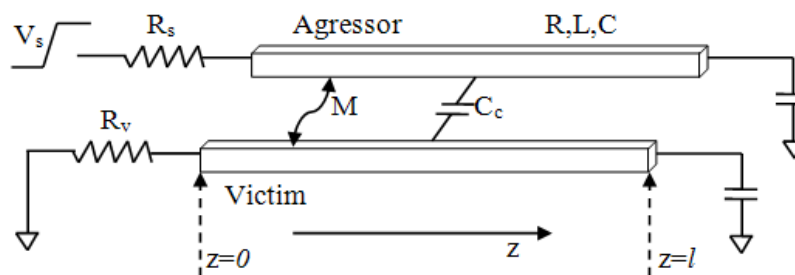


图 6 两根相邻导线的传输线模型

Figure 6 Transmission Line Model for Adjacent Interconnects

假设两根传输线中，其中一根是攻击线网，另外一根是受干扰线网。攻击线网的信号源为正跳变，而受干扰线网的信号源保持静止。因此对于攻击导线列出电流电压方程有

$$-\frac{\partial V_1}{\partial z} = (R + sL)I_1 + sMI_2 \quad (7)$$

$$-\frac{\partial I_1}{\partial z} = s(C + C_c)V_1 - sC_c V_2 \quad (8)$$

在式(7)和(8)中, V_1 , I_1 为攻击导线上某一点的电压和电流, V_2 , I_2 为受干扰导线上某一点的电压和电流。 M 和 C_c 分别为干扰导线和受干扰导线之间的互感和耦合电容。同理可以得到受攻击导线上的电流电压关系。

$$-\frac{\partial V_2}{\partial z} = (R + sL)I_2 + sMI_1 \quad (9)$$

$$-\frac{\partial I_2}{\partial z} = s(C + C_c)V_2 - sC_c V_1 \quad (10)$$

求解方程(9)(10)可得传输线上的电流和电压表达式为:

$$V_1 = (A_1 e^{-\gamma_e z} + A_2 e^{\gamma_e z}) + (A_3 e^{-\gamma_o z} + A_4 e^{\gamma_o z}) \quad (11)$$

$$V_2 = (A_1 e^{-\gamma_e z} + A_2 e^{\gamma_e z}) - (A_3 e^{-\gamma_o z} + A_4 e^{\gamma_o z}) \quad (12)$$

$$I_1 = \frac{1}{Z_{0e}} (A_1 e^{-\gamma_e z} - A_2 e^{\gamma_e z}) + \frac{1}{Z_{0o}} (A_3 e^{-\gamma_o z} - A_4 e^{\gamma_o z}) \quad (13)$$

$$I_2 = \frac{1}{Z_{0e}} (A_1 e^{-\gamma_e z} - A_2 e^{\gamma_e z}) - \frac{1}{Z_{0o}} (A_3 e^{-\gamma_o z} - A_4 e^{\gamma_o z}) \quad (14)$$

在式(11)(12)(13)和(14)中, γ_e , γ_o , Z_{0o} 和 Z_{0e} 的值分别为:

$$\gamma_e = \sqrt{sC(R + s(L + M))}, \gamma_o = \sqrt{s(C + 2C_c)(R + s(L - M))}$$

$$Z_{0e} = \sqrt{\frac{R + s(L + M)}{sC}}, Z_{0o} = \sqrt{\frac{R + s(L - M)}{s(C + 2C_c)}}$$

假设传输线是无损耗的, 即 $R=0$, 这对于大多数材料都是近似成立的, 根据流过驱动源电阻 R_s 和 R_v 的电压电流关系我们可以得到:

$$\frac{V_s - V_1(z=0)}{I_1(z=0)} = \frac{V_s - (A_1 + A_3)}{\left(\frac{A_1}{Z_{0e}} + \frac{A_3}{Z_{0o}}\right)} = R_s \quad (15)$$

$$\frac{-V_2(z=0)}{I_2(z=0)} = \frac{-(A_1 - A_3)}{\left(\frac{A_1}{Z_{0e}} - \frac{A_3}{Z_{0o}}\right)} = R_v \quad (16)$$

解上述两个方程进而可以推得在时刻 t , 攻击线网和受干扰线网上的电压表达式为:

$$V_{agg}(t) = 2A_1(t - t_{fe}) + 2A_3(t - t_{fo}) \quad (17)$$

$$V_{vic}(t) = 2A_1(t - t_{fe}) - 2A_3(t - t_{fo}) \quad (18)$$

其中 A_1 , A_2 的值为

$$A_1 = V_s \frac{Z_{0e}(Z_{0o} + R_v)}{(Z_{0e} + R_s)(Z_{0o} + R_v) + (Z_{0e} + R_v)(Z_{0o} + R_s)}$$

$$A_3 = V_s \frac{Z_{0o}(Z_{0e} + R_v)}{(Z_{0e} + R_s)(Z_{0o} + R_v) + (Z_{0e} + R_v)(Z_{0o} + R_s)}$$

该方法能够处理两根具有不同寄生参数导线之间的耦合情况，在上述传输线模型的求解过程中，用到了几个非常重要的假设：

- 负载阻抗远远大于导线本身阻抗。
- 互联线是无损耗的，即电阻 R 为 0。
- 只能处理电容耦合或者电感耦合 ($C_c=0$ or $M=0$)，但是[18]提出，电感耦合和电容耦合是满足叠加原理的，即在串扰估计时可以先单独计算电感耦合和电容耦合，随后将两者的效果叠加。

传输线分析比较不足的地方是目前还无法直接处理树形结构，并且模型求解的过程中用到了一些假设来简化模型。而实际 VLSI 设计中，大部分互联线都是树状结构的（一个源节点驱动若干个终端节点），并且互连线的电阻值随着工艺尺寸的缩小不断变大，假设 $R=0$ 导致的误差会越来越大。目前采用传输线模型的延时估计还只是停留在估计阶段，不能将其应用到互联线优化中。

基于电路约简的串扰估计

[26]提出了一种基于电路约简来估算串扰噪声的方法。我们知道，任意一点的电导 $Y(s)$ 都可以表示成泰勒级数的形式：

$$Y(s) = y_0 + y_1s^1 + y_2s^2 + y_3s^3 + y_4s^4 + \dots \quad (19)$$

其中 y_0 , y_1 , y_2 , y_3 分别为电导的 0 阶、1 阶和 2 阶泰勒系数。另外我们知道，任何一个 RC 电路结构都可以看成是图 7 中所示 3 种情况的组合。规则 1 表示电阻的串联，规则 2 表示电容的串联，规则 3 表示两个支路的合并。根据电导的计算公式，我们很容易根据前一个节点的电导 $Y(s)$ 推算出下一个节点的电导 $Y^*(s)$

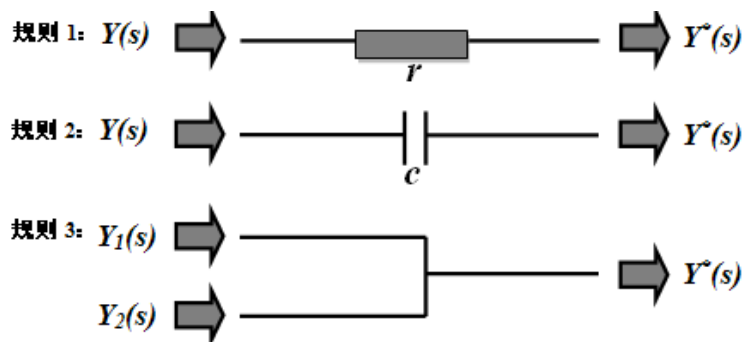


图 7 电路约简规则

Figure 7 Circuit Reduction Rule

对于图 7 中所示的三种形式的约简，我们很容易推导出电导泰勒展开系数的推导公式如表 1 所示

表 1 电路约简公式

规则一约简	规则二约简	规则三约简
$y_0^* = py_0,$ $y_1^* = p^2 y_1,$ $y_2^* = p^2 y_2 - p^3 r y_1^2,$ $y_3^* = p^2 y_3 - 2p^3 r y_1 y_2 + p^4 r^2 y_1^3$ 其中 $p = \frac{1}{1 + r y_0}$	$y_0^* = 0,$ $y_1^* = c,$ $y_2^* = \frac{-c^2}{y_0},$ $y_3^* = \frac{c^2(y_1 + c)}{y_0^2}$	$y_0^* = y_{1,0} + y_{2,0}$ $y_1^* = y_{1,1} + y_{2,1}$ $y_2^* = y_{1,2} + y_{2,2}$ $y_3^* = y_{1,3} + y_{2,3}$

通过上述的电导等效方法，任何树状电路都可以最终等效成一个等效电容。如图 8 所示。其中从(b)到(c)的过程需要用到一些近似。

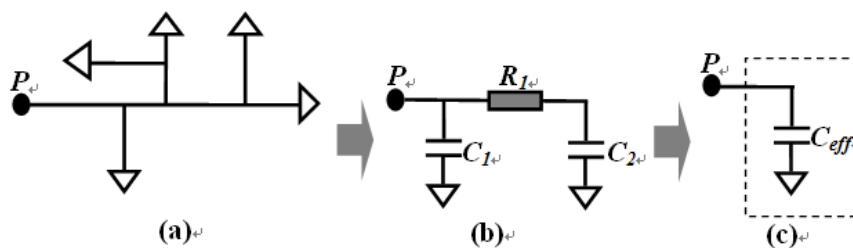


图 8 RC 电路分支的约简

Figure 8 Reduction for Circuit Branches

每个攻击线网都可以用 π 模型来进行建模，其中 C_x 为耦合电容。[26]通过 RC 树的约简将静态的攻击线网等效成一个电容。该方法估计串扰的方法步骤如下：

1. 对静态的攻击线网进行简化等效。一个受干扰线网可能会受到多个攻击线网的影响。即使在某一时刻只有一个攻击线网发生跳变，这时其他静止的攻击线网对受干扰线网上的信号跳变也会产生影响。通过简化其他静止的攻击线网，既将其他攻击线网对串扰的影响考虑在内，又在一定程度上简化了计算。
2. 对受干扰线网进行简化等效。
3. 提取寄生参数，对受干扰线网上的干扰信号进行估计。经过前面步骤的简化等效，最终所得到的 RC 电路规模已经非常小，这时可以用求解电路的方式来估计串扰的影响。

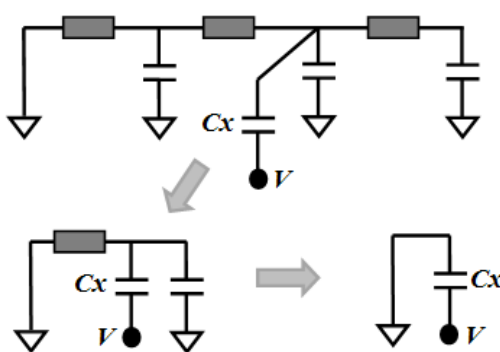


图 9 静止攻击线网的约简

Figure 9 Reduction of inactive aggressor

基于电路各阶矩的串扰估计

电路中节点的矩即该节点信号泰勒展开的各阶系数，其中常数项称之为 0 阶矩， s 的系数称之为 1 阶矩， s^n 的系数称之为 n 阶矩，电路中节点的矩已经被证明是一个比较容易求解的量。[20][19]提出了一种串扰估计方法，得出非常精炼的公式来计算串扰。并且提出了一种只要简单的遍历 RC 树就可以计算串扰的方法。但是该方法所得到的串扰估计是真实串扰的上限情况，因为真实情况下，攻击信号的输入不可能没有上限。该方法对串扰的估计随着信号上升时间的变快而精度降低。并且其估计所得出的串扰与线网的对地电容没有关系，所得出的结果也与攻击线网上的电阻无关，这显然是不合理。

为了解决上述缺点，[19]提出了一种通过矩匹配来求解受干扰线网上的波形的方法。首先我们知道，两根互相干扰的线网可以建模成如图 10 所示的 RC 网络。

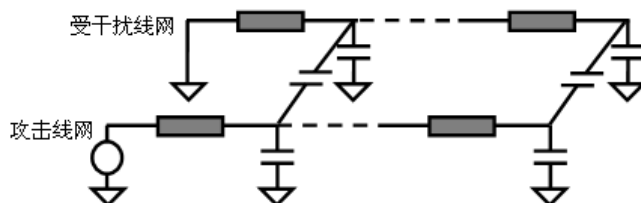


图 10 考虑电容耦合的串扰模型

Figure 10 Circuit Model Accounting Capacitance Coupling

对图 10 中的线网列出基尔霍夫方程可得：

$$\begin{bmatrix} C_1 & -C_c^T \\ -C_c & C_2 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -A_{11} & 0 \\ 0 & -A_{11} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} v_s \quad (20)$$

其中 C_c 为耦合电容， V_1 为攻击线网上的节点电压， V_2 为受干扰线网上的节点电压。对式(20)中的方程展开，并使对应项相等可得：

$$\begin{aligned} sC_1V_1 - sC_c^TV_2 &= -A_{11}V_1 + B_1V_s \\ sC_cV_1 - sC_2V_2 &= A_{22}V_2 \end{aligned} \quad (21)$$

假设攻击线网上的信号跳变是非阶跃的，则其信号跳变波形可以写成如下的拉普拉斯展开形式，同理受干扰线网上的信号跳变 $V_2(s)$ 也可以写成拉普拉斯展开的形式。

$$\begin{aligned} V_1(s) &= v_{10}s^{-1} + v_{11} + v_{12}s + v_{13}s^2 + v_{15}s^4 + \dots \\ V_2(s) &= v_{20} + v_{21}s + v_{22}s^2 + v_{23}s^3 + v_{24}s^4 + \dots \end{aligned} \quad (22)$$

将式(22)中的两式代入式(21)，并使相应系数相等，即可得到 $V_1(s)$ 和 $V_2(s)$ 中各阶泰勒系数的值。另外我们知道，受干扰线网上的信号可以近似表示成如式(23)所示的形式，其有两个极点

$$V_2(s) = \frac{a_0}{1 + b_1s + b_2s^2} = v_{20} + v_{21}s + v_{22}s^2 + v_{23}s^3 + v_{24}s^4 \quad (23)$$

利用待定系数法，我们可以求出 b_1 ， b_2 ， a_0 的值，将式(23)中的 $V_2(s)$ 做拉普拉斯反变换即可以得到受干扰线网的时域波形为：

$$V_2(t) = c_1e^{-t/k_1} + c_2e^{-t/k_2} \quad (24)$$

利用式子(24)所示的时域表达式我们可以很容易估计出信号的串扰，延迟等。

1.3.2 互联线的信号延迟

在 VLSI 电路中，特征尺寸的缩小在提高电路性能、减小芯片单位功能成本方面起到了非常大的作用。在 $0.18\mu\text{m}$ 工艺之前，电路的延迟主要是由门延迟引起的，随着 MOS 管特征尺寸的缩小，门延迟也随之减小，与此同时，互联线上的延迟有增无减。随着芯片几何尺寸的缩小，互连延时的比重越来越大。较长的互连延时使得总连线最小设计原则不再适用。总的技术发展的趋势为：

- 互连线延时成为电路性能的关键。尽可能的优化互连延时已经成为提高工作频率的必要手段。
- 相邻互连线之间的耦合电容将成为总电容中的主要成分，因为金属线的纵横比不断增大而线间的间隔不断缩小，但是耦合电容的值受线间隔的影响很大。如何合理的设置线间隔对超深亚微工艺下互连线的设计是很重要的。

当两根相互耦合的互联线上的信号翻转方向相同时，受干扰线网上的信号跳变速度加快，从而信号的延迟变小；当两根相互耦合的互联线上的信号翻转方向相反时，受干扰线网上的信号跳变速度变慢，从而导致信号的延迟变大。因此互联线之间的串扰会影响信号延迟的大小。

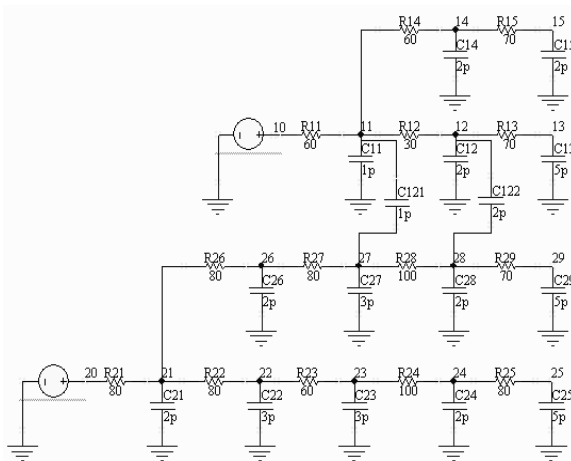


图 11 互联线串扰对信号延迟影响的示意电路

Figure 11 Circuit for demonstrating how coupling affects Signal delay

图 11 所示的是两个互相耦合的线网，其中 C_{121} 和 C_{122} 为耦合电容。其中源节点为 10 的线网为攻击线网，源节点为 20 的线网为受干扰线网。

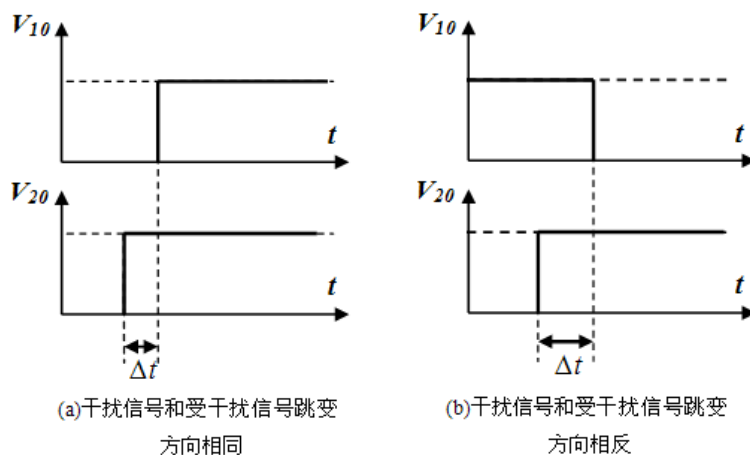


图 12 相邻线网不同的信号跳变方向

Figure 12 Signal Transition of Adjacent Electrical Nets

我们分两种情况对图 11 中所示受干扰线网的节点 29 进行测试，第一种情况如图 12 (a)中所示：源节点 V_{10} 和 V_{20} 都是正跳变。记 Δt 为 V_{10} 和 V_{20} 跳变边沿的时间差；第二种情况如图 12 (b)所示：源节点 V_{10} 为负跳变， V_{20} 为正跳变，记 Δt 为 V_{10} 和 V_{20} 跳变边沿的时间差。对这两种情况下采用 HSpice 仿真测算节点 V_{29} 的延迟与 Δt 的关系得到的关系如下图所示。

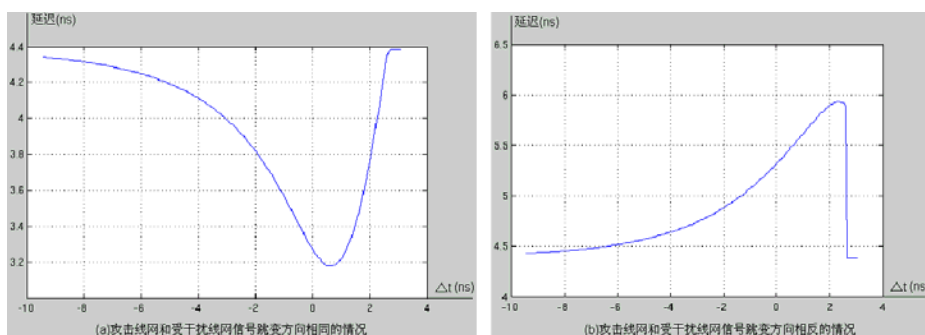


图 13 受干扰线网节点延迟随信号跳变时间差的变化

Figure 13 Signal delay Versus Different Signal Arrival time of Adjacent Nets

从上图 13 (a)可以看出，当两个线网的信号跳变方向相同时，节点的延迟随着 Δt 趋向于 0 变小，随着 Δt 从 0 变大而逐渐恢复原来值；当两个线网的信号跳变方向相反时，节点的延迟随着 Δt 趋向于 0 变大，随后再逐渐恢复原来值。

从上述仿真结果可以看出：当响邻导线的信号跳变方向相反时，导线上的信号延迟会显著变大；而相邻导线上的信号跳变方向相同时，导线上的信号延迟会变小。我们再进行时序约束时，如果没有将相邻导线上信号跳变的效果加以考虑，可能会导致过于乐观估计信号延迟，从而使设计出来的电路违反时序约束。

1.3.3 互联线的延迟估计

延迟估计中得到最广泛应用的是 Elmore 延迟估计了，为了进一步提高延迟估计的精度，还有其他各种更高阶的估计方法例如 D2M, Gamma 分布等。本节简要介绍几种延迟估计方法。

Elmore 延迟估计

Elmore 延迟估计是目前工业界最广泛应用的延迟度量，Elmore 延迟之所以获得这么广泛的应用，与其以下方面的特点是分不开的：

- Elmore 延迟在任意输入波形下，都是信号延迟的上限[21]。这使得在时序估计时，只要保证 Elmore 延迟估计是在延时上限以内，真实的延时也必定满足时序约束。我们可以说 Elmore 延迟估计有非常高的可靠性
- Elmore 延迟估计的计算公式非常简单，为：

$$Delay(i) = \sum_{j \in Path(i)} \left(-R_j \sum_{k \in Des(j)} C_k \right) \quad (25)$$

计算某个树状 RC 电路所有节点的 Elmore 延迟只需要遍历两次 RC 树即可，因此非常适用于较大规模 RC 电路的延迟估计。

但是 Elmore 延迟也有其不足，例如在靠近源节点处，Elmore 延迟对信号延迟的估计误差非常大，可达 100% 以上。还有 Elmore 延迟估计无法将相邻线网上信号跳变对受干扰线网信号延迟的影响考虑在内，因此其对处理存在耦合情况下的信号延迟时，并不是很准确。

高阶延迟估计

如果精确的估计信号的延迟一度引起了学术界的极大关注，利用更高阶矩来对延迟进行估计的方法层出不穷。例如[22]提出了一种基于前两阶 moment 的延时估计方法 (D2M)，其计算公式也非常简单：

$$Delay(i) = \ln 2 \frac{m_1^2}{\sqrt{m_2}} \quad (26)$$

D2M 能够达到很高的估计精度，特别是对靠近源节点的估计，精度远远高于

Elmore 延迟。D2M 只是一个经验公式，目前还没能给出一个确切的理论推导来证明 D2M 的精度。

除此之外，还有各种各样的延时估计方式，大部分的延时估计需要用到 2 阶、3 阶，甚至更高阶的矩。[23]提出了一种在非阶跃输入情况下的延时估计方法；[23]提出了一种采用时域时移的方法来降低计算阶数的方法。由于 RC 树的脉冲响应的时域波形在 $(0, +\infty)$ 的积分为 1，因此可以等效成一个概率分布函数，[25]通过将脉冲响应波形等效成 Gamma 分布的方法来估计延时。[13]提出了一种符号化计算 RC 电路矩的方法，为快速估计延时提供了理论基础和算法基础。

1.3.4 符号化矩计算

要采用高阶矩来估计节点的延迟，必须要快速计算 RC 树中节点各阶矩。传统的矩计算方法采用的是不断遍历 RC 树的数值计算方法，[13]提出了基于矩决策图的电路矩的符号化计算方法。它利用二叉决策图 (Binary Decision Diagram, BDD) 共享子图的特点，高效地将矩的计算公式存储为图的数据结构，实现了矩计算的符号化。

为了能够简化路径追踪的过程，[13]用一棵由所有电容组成的电容树 (CTree) 和所有电阻组成的多根节点 BDD 来表示矩的计算公式。我们把这个用于计算电路矩的数据结构称为矩决策图 (Moment Decision Diagram, MDD)。

图 14 所示的是一个简单的树状 RC 电路，下面我们以它为例，简要介绍如何采用符号化的方法来计算树状 RC 电路的矩。

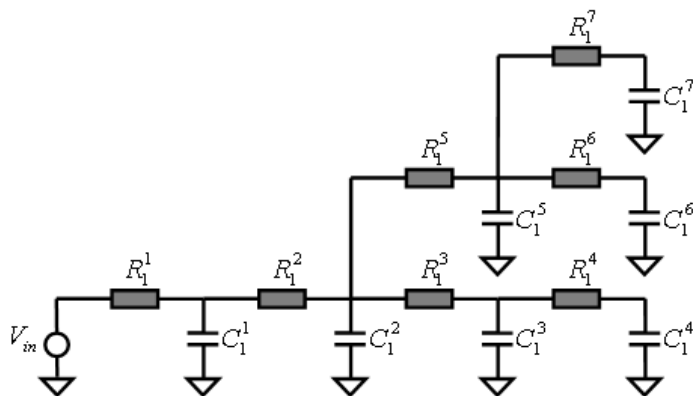


图 14 树状 RC 电路

Figure 14 Simple Case of RC Tree

第一步：建立一棵与电路结构相同的电容树 (CTree)。如图 15 所示，电容树的结构与 RC 电路的拓扑结构完全相同。

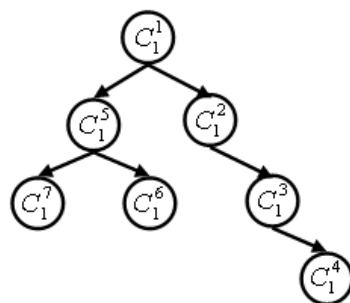


图 15 电容树

Figure 15 Capacitor Tree

第二步：建立一棵由电阻树和电容树构成的多根节点的二叉决策图。这个二叉决策图中的电阻树与电容树的拓扑结构完全相同，如图 16 所示。

当我们需要计算某一个节点的一阶矩时，只要从某个电阻节点出发，遍历整个二叉决策树即可。其中实线表示为加法运算，虚线表示为乘法运算。

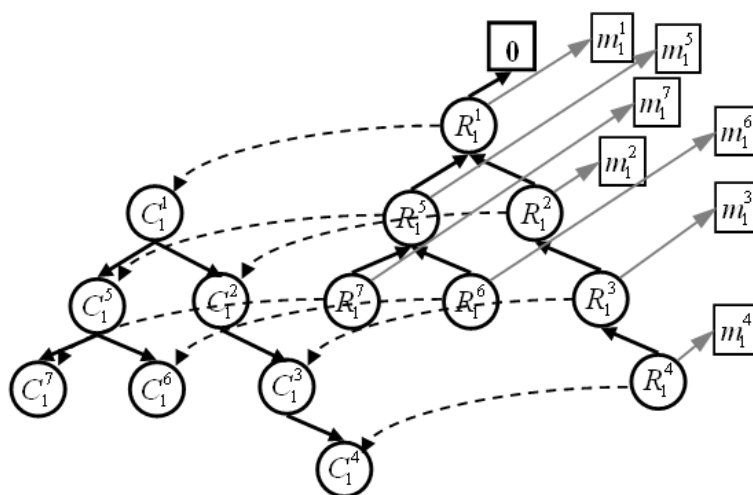


图 16 用于计算一阶矩的二叉决策树

Figure 16 Binary Decision Diagram for Calculating First Order Moment

当需要计算高阶矩的值时，我们只要将图 16 中的所有电容用其与低阶矩的乘积代替即可。该符号化方法还可以推广到能够处理耦合的情况[14]。

本文中所采用的符号化矩计算也正是基于这种方法，在此基础上，本文还对符号化方法进行了推广，使其能够快速计算矩关于电路中元器件的敏感度。

1.4 以互联线为中心的设计流程

VLSI 的自顶向下的设计流程如图 17 所示，大体上可以分为前端设计和后端设计。其中前端设计主要是指电路逻辑的设计，包括电路功能模块的划分，设计，模块的功能性验证等，前端设计的最终结果是电路网表。后端设计主要是基于电路网表对电路进行物理层面的设计，例如确定互联线走线，模块的布局等，与此同时对电路的时序，功耗等进行优化。

从门级网表到 GDS 的过程属于后端设计，后端设计中各个环节做的事情如下：

- DFT（可测性设计）。复杂 SOC 设计正面临着高可靠性、高质量、低成本以及更短的产品上市周期等严峻挑战。可测性设计通过在电路中插入扫描链来提高电路的可测试性，从而保证芯片的生产质量。
- 布局设计。布局规划最主要的工作就是确定芯片和硬核的形状大小，合理放置内部 IP 的位置，确定管脚位置，布置电源线网络。合理的布局规划能够提高芯片布线质量、芯片面积等。
- 时钟树生成。时钟树生成必须要等到物理综合之后完成，此时所有的标准单元的位置都已经基本确定。时钟设计的主要目标为：满足时钟偏斜要求；满足延迟要求；满足时钟信号的翻转速度要求，让时钟信号有足够强的驱动能力。
- 布线/RC 提取。布线是指在标准单元之间建立电气互连。随着 IC 尺寸变小和时钟频率的逐渐提高，互连线上寄生参数的影响对于芯片的速度和噪音等变得无法忽略。在芯片设计过程中必须精确提取互连线上的寄生参数来模拟新工艺下互联线寄生参数对信号的影响。
- 受 EDA 设计工具和验证方法学的限制，目前大部分数字集成电路都是同步时序电路，即电路中所有受时钟控制的单元（如触发器）都是由一个全局时钟来控制的。在同步时序电路中，首要的问题就是触发器的时序收敛问题。触发器的时序收敛保证了触发器输入端的数据在时钟信号有效沿来临之前就达到稳定状态；同时也保证了触发器输入端数据在时钟有效沿过后的一定时间内继续保持稳定。
- 物理验证是物理设计的最后一步。通过物理验证设计者可以确定他的设计是否满足设计规则，并且可以确定芯片最终的版图与逻辑门网表是否一致。不经过以上的检查，设计者就无法保证芯片版图的正确性，如果轻易流片生产的话就可能造成巨大的损失。

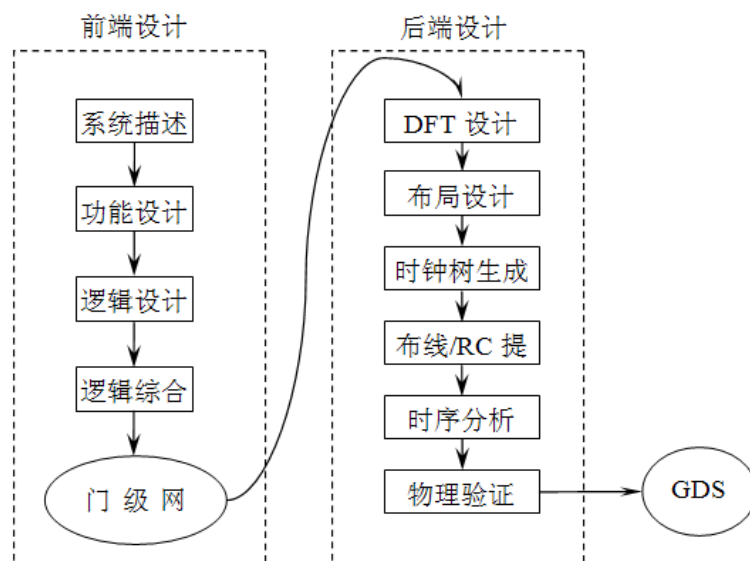


图 17 VLSI 设计流程

Figure 17 VLSI Design Flow

本文中所指的 VLSI 物理设计优化主要体现在布局布线完成之后，这时可以提取布线寄生参数的准确信息，对互联线的电气效应进行建模。可以准确分析节点的延迟是否满足时序要求，并根据条件进行相应的优化。

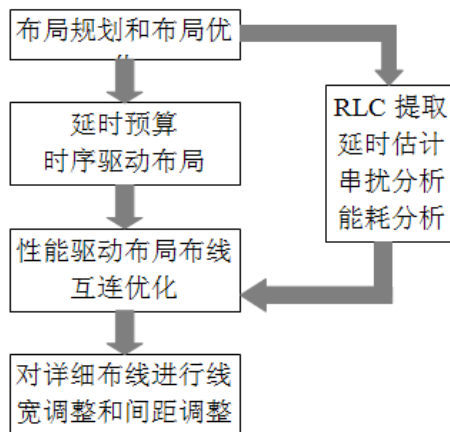


图 18 以互联线为中心的设计流程

Figure 18 Interconnects Focused Design Flow

以互联线为中心的设计流程如图 18 所示。在布局阶段对延时进行预算，从而优化标准单元的布局。在布线之后对互联线的寄生参数进行提取，进而对布局和布线进行反复调整。当完成布局布线之后，再对布线进行线宽调整和间距调整，从而优化物理设计的时序，面积，功耗等。

1.5 VLSI 物理设计中的优化

VLSI 物理设计中的优化包括功耗优化、面积优化、时序优化和信号完整性优化。一般设计时需要综合考虑这几个方面的优化，达到一个折中的结果。

功耗优化

集成电路中的功耗主要包括动态功耗和静态功耗，静态功耗是指电路中的信号处在稳定状态时 MOS 管漏电流产生的功耗。静态功耗主要和集成电路制造的工艺有关。动态功耗是指电路中的信号在翻转过程中，电路节电容充放电过程中产生的功耗。动态功耗的估算公式为：

$$P_d = \alpha CV^2 f$$

其中 V 为电路的电源电压， C 为电路的负载电容， f 为电路工作的频率， α 为某一小于 1 的实数。

集成电路中动态功耗占功耗的主导作用，这就是为什么在系统没有开启任何程序时，系统的风扇往往停转；而当系统开启了某一个 CPU 占用率比较高的程序时，CPU 温度立即上升，导致风扇开启。从动态功耗的估计表达式中我们可以看出，要降低电路的动态功耗有以下几种方法：

1. 降低电路的工作电压。由于电路的工作电压往往是由集成电路的工艺决定的，一般对于特定的工作电压都有特定的标准器件模块。因此设计过程中无法随意改变电路的工作电压来降低电路的动态功耗，
2. 降低电路的工作频率。降低电路的工作频率意味着给芯片减速，在对芯片速度要求越来越高的当前形势下，给芯片减速是不大现实的。并且随着集成电路的发展，芯片的速度越来越高。当然有些芯片有特定的功耗管理模块，在芯片工作负载不高的情况下适当调低电路的工作频率来降低功耗。
3. 减小负载电容。这也是目前最常用的方法，在早期的工艺中，门电容占负载电容的主要部分，随着工艺尺寸的减小，门节点电容越来越小，互联线电容在负载电容中所占的比例逐渐提高。同时在布线过程中，如果能够减小布线长度，就能够减小负载电容，因此有一些列的拓扑优化算法用来在布局过程中减小芯片的互联长度。

面积优化

芯片的面积和芯片的成本是成正比关系的，因此在芯片的设计过程中往往千方百计的减小芯片的面积。芯片的面积主要由三个部分组成：

标准门单元面积：在特定工艺下，标准门单元的面积一般是经过优化的。标准门单元一般是工程师综合考虑单元的工作稳定性和面积所给出的设计。

布线面积：布线面积即互联线所占用的面积，随着工艺尺寸的缩小，互联线的复杂度不断增加。现在布线面积已经占到芯片面积的 50% 以上。因此面积优化过程中布线优化是很重要的一个工作。

芯片空余面积：芯片空余面积是指互联线和期间之间，标准单元之间的间隔。这些间隔是为了增加芯片的工作稳定性而加入的。

时序优化

时序优化分为两个方面，时钟树时序的优化和一般线网的时序优化：

时钟树：时钟树要求始终边沿到达各个时钟节点的延迟相同或者保持在很小的范围内，否则就会产生时钟偏移。因此时钟树优化是一种等时延优化。

一般线网：一般线网的优化与时钟树上的时延优化不同，只要保证线网上的信号到达节点的延迟在一定范围内即可。

信号完整性优化

信号完整性优化是指尽量减小芯片上的信号完整性问题，解决的方法主要有：

1. 增加源节点的驱动能力，增加源节点的驱动能力可以提高该线网抵抗干扰的能力。
2. 加入屏蔽线。对于一些特别敏感的信号，需要进行特殊的保护，通常的方法为对这种信号进行屏蔽，最大程度减小别的信号对其的干扰。
3. 增加互联线之间的间距。增加互联线之间的间距可以减小耦合电容，从而减小芯片上的信号完整性问题。

1.6 VLSI 物理优化算法

物理布线优化主要有两种途径，一种是互连拓扑结构的优化，另外一种是对线宽等导线参数进行优化。互连拓扑优化是指在布局布线阶段就进行优化，通过调整标准单元的摆放，优化互联线走线来减小互联线长度，达到减小延时，节约面积等优化目的。而线宽调整则是在布局布线已经完成后进行的。本文中所指的物理优化是指通过线宽调整来进行物理设计的优化。

在传统的 VLSI 物理设计中，由于布线的电阻值与驱动电阻相比非常小，因此线上的延迟可以通过驱动该走线的驱动电阻与导线布线电容的乘积来确定[1]。缩短线上延迟基本上等同减少总的布线电容。因此传统的 VLSI 中，延迟优化主要是通过构造斯坦纳树来实现。

然而随着工艺尺寸的不断缩小，驱动电阻相对导线电阻的比值越来越小，布线总电容的最小化不再意味着线上延迟的最小化，导线上的分布电阻必须加以考虑。在这种情况下，通过调整导线各部分的线宽，可以使线上延迟实现最优化。学术界对线宽调整优化进行了大量的研究。

[2]最早提出通过导线线宽调整来实现延迟的最小化。[8]提出了将耦合电容考虑在内的全局线宽优化算法。[9][10]利用拉格朗日松弛法，在保证串扰，限制功耗和延迟在阈值以下的同时使导线布线面积最小化。[15]利用线宽调整的方法同时对串扰和延迟进行了优化。

大部分物理设计优化出于计算复杂度的考虑，都采用 Elmore 作为延迟估计[8][9][10]，Elmore 虽然形式简单易于计算，且被证明是信号延迟的上限，但其对靠近源节点的延迟估计过于悲观（误差可达 100% 以上），图 19 给出了同一棵 RC 树 Elmore 延迟估计、D2M 延迟估计[3]和 HSpice 仿真结果，可以看出 Elmore 延迟的估计误差非常大，而 D2M 则精确得多。如果采用 Elmore 延迟作为度量标准，必然会导致优化结果过于悲观；不少文献中的优化过程都考虑了串扰，但他们在考虑串扰时采用的是非常简单的模型，例如[9][10]在优化时只考虑了耦合电容之和，而其他大部分研究虽然进一步考虑了串扰对延迟的影响，但是也只是采用一阶的模型[15]。实际上大量的研究成果已经明确指出，干扰网线上的信号变化对受干扰线网产生很大影响。例如当干扰线网与受干扰线网上的信号翻转方向相反时，受干扰线网上信号延迟变大；在优化过程中如果没有将最坏的情况考虑在内，必然会导致优化结果的严重误差。

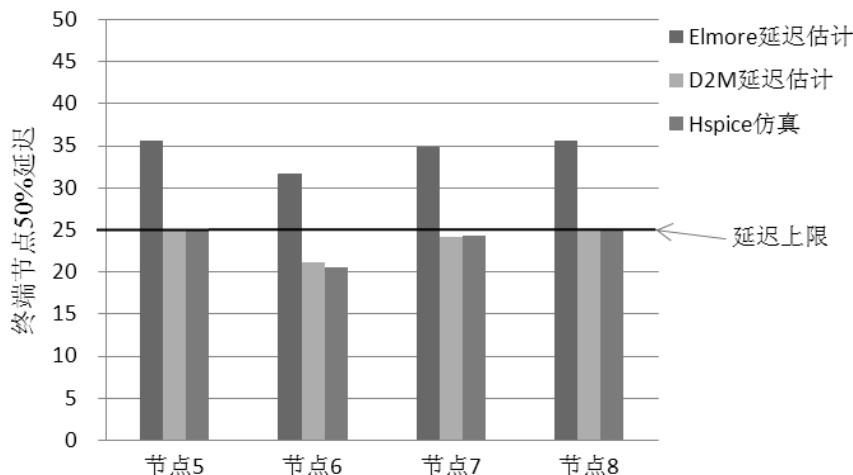


图 19 Elmore 延迟估计和 D2M 延迟估计精度的比较

Figure 19 Precision Comparison between Elmore Delay Estimation and D2M Delay Estimation

综上所述，目前的物理优化研究有两个局限：一是在进行优化时大多采用 Elmore 作为延迟估计，这使得优化的结果还有很大的余量。第二个是在考虑不同布线树干扰的时候采用的是非常简单的模型，例如[8]在优化时只考虑了耦合电容之和，而其他大部分研究虽然有考虑耦合对延迟的影响。但是都采用的是一阶模型，并没有将相邻线网上信号跳变的影响考虑在内。

1.7 本文的主要研究内容及安排

本文主要致力于 VLSI 物理互联线优化算法的研究，主要的研究内容包括以下几个方面：

1. 矩敏感度的符号化计算方法。基于之前提出的符号化矩计算方法，本文将进行了推广，得出了一种高效的符号化矩敏感度计算方法。该方法只要遍历符号化树一次，就可以求出加权矩关于电路中所有 R ，或者所有 C 的敏感度。并且本文中所提出的矩敏感度计算方法形式简单，很容易推广到任意高阶矩的敏感度计算。
2. 基于 VLSI 物理设计的特点，建立 VLSI 的物理优化模型，该模型相对于之前文献中的模型，才有更高阶矩来估计节点的延迟，并且考虑了最坏情况下的信号延迟。因此该模型有较小的冗余量，并且充分考虑了最坏的情况。
3. 采用拉格朗日松弛法求解优化模型，在这过程中应用了本文中所提出了符号化矩计算方法。测试结果发现，基于更高阶矩的优化模型能够很好的度量延迟的状况，因此优化的结果既能够保持在约束条件范围内，又能够最大程度的利用设计余

量，能够节约成本。

文章的最后提出了本文中尚未完成但是非常有意义的一些工作，并且提出了对未来 VLSI 物理设计优化工作的一些展望。

第2章 符号化矩敏感度计算

由于矩在估计互联线延迟、串扰等方面有非常大的作用。因此在进行 VLSI 延迟优化、串扰优化时，矩敏感度有非常重要的作用。如果能够快速的计算矩敏感度，必然会增加优化算法所能够处理的电路规模。本章主要提出了一种基于符号化表达式的矩敏感度计算方法，由于其可重复计算性强，计算效率高的特点，非常适合应用到各种优化算法中。

2.1 矩敏感度介绍

所谓矩敏感度，就是节点的矩关于电路中各个元器件的敏感度，矩敏感度在电路优化中有非常重要的作用，其运算速度的快慢直接影响到优化算法的效率。在传统的优化过程中，一般是通过数值方法来计算节点的矩敏感度，而本文是采用符号化的方法来计算节点的矩敏感度，利用若干次遍历符号树就可以求出各个节点的矩敏感度。本章中所提到的矩敏感度计算方法已经被应用到电路的优化过程中。

利用矩敏感度我们可以很方便的求得导线电气参数关于导线几何参数的敏感度，下面以导线的延迟敏感度作为例子来说明这个问题。我们知道 RC 树中的每根导线的电气参数都可以用 π 模型来等效，例如图 20 中所示的互联线，对其中的每根导线采用 π 模型等效，等到相应的 RC 电路，其中 R_1 为驱动电阻，用来等效信号源驱动能力的大小。

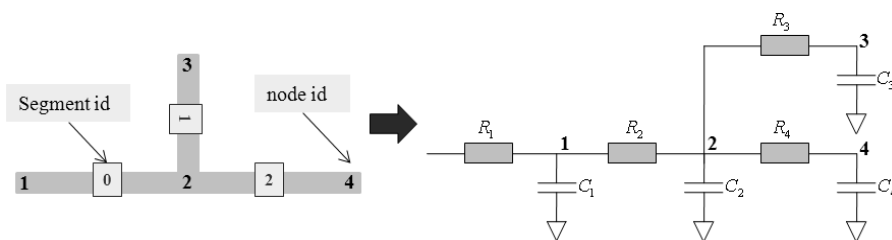


图 20 矩敏感度的应用例子

Figure 20 An Example of Moment Sensitivity Application

图中的每个电阻即是当前导线段的等效电阻，而对应节点的电容为与之相邻导线段等效电容之和。例如假设导线段 0、导线段 1 和导线段 2 的对地电容为 C_{s0} , C_{s1} ,

C_{s2} , 电阻值为: R_{s0} , R_{s1} , R_{s2} 。假设对于任意一个导线段, 电容和电阻的计算公式为:

$$\begin{aligned} C_{s0} &= p(W_{s0}), C_{s1} = p(W_{s1}), C_{s2} = p(W_{s2}) \\ R_{s0} &= q(W_{s0}), R_{s1} = q(W_{s1}), R_{s2} = q(W_{s2}) \end{aligned} \quad (27)$$

式(27)中函数 p 表示互联线线宽与互联线电容的关系, 函数 q 表示互联线线宽与互联线电阻的关系。则图 20 中 RC 模型的计算表达式为:

$$\begin{aligned} C_1 &= \frac{p(W_{s0})}{2} \\ C_2 &= \frac{(p(W_{s0}) + p(W_{s1}) + p(W_{s2}))}{2} \\ C_3 &= \frac{p(W_{s1})}{2} \\ C_4 &= \frac{p(W_{s2})}{2} \end{aligned} \quad (28)$$

我们知道, 节点的延迟可以用该节点的各阶矩来进行估计, 例如我们可以利用 D2M 来估计节点 3 的延迟, 其估算公式为:

$$Delay(3) = f(m_1^3, m_2^3) = \ln 2 \frac{(m_1^3)^2}{\sqrt{m_2^3}} \quad (29)$$

那么节点 3 延迟关于导线 0 宽度的敏感度可以表示成:

$$\begin{aligned} &\frac{\partial f(m_1^3, m_2^3)}{\partial W_{s0}} \\ &= \frac{1}{2} \frac{\partial f(m_1^3, m_2^3)}{\partial C_1} \frac{\partial C_1}{\partial W_{s0}} + \frac{1}{2} \frac{\partial f(m_1^3, m_2^3)}{\partial C_2} \frac{\partial C_2}{\partial W_{s0}} + \frac{\partial f(m_1^3, m_2^3)}{\partial R_2} \frac{\partial R_2}{\partial W_{s0}} \end{aligned} \quad (30)$$

利用这个敏感度值我们可以知道线宽的调整方向, 从而对物理设计进行优化。

2.2 符号化加权矩敏感度计算

由于矩在刻画 RC 电路节点延迟方面有非常大的作用, 节点延迟的优化不可避免的用到矩敏感度的计算, 通常的矩敏感度计算需要为每个元器件敏感度单独遍历一次 RC 树, 因此如果要计算关于电路中所有 RC 的矩敏感度, 计算量会非常的大。

本文提出了一种新的计算 RC 矩敏感度的方法, 通过构建符号化表达式, 只需要遍历若干次就可以求出所有 RC 元器件的矩敏感度。

2.2.1 矩敏感度计算的递归表达式推导

[27]提出了一种快速计算 RLC 电路各阶矩的方法,对于任意一个线性电路,只要反复求解其直流解,就可以求出相应节点各阶矩。因此很容易得出对于树状 RC 电路,其节点 i 的第 n 阶矩的计算可以由迭代计算表达式,如式(31)所示:

$$m_n^i = \sum_{j \in \text{Path}(i)} \left(R_j \sum_{k \in \text{Des}(j)} (-C_k m_{n-1}^k) \right) \quad (31)$$

那么 RC 树中各个节点 n 阶矩的加权和可以表示如式(32)所示

$$\sum_{i \in \text{Node}(T)} \lambda_i m_n^i = \sum_{i \in \text{Node}(T)} \left(R_i \left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(j)} (-C_k m_{n-1}^k) \right) \right) \quad (32)$$

其中 λ_i 为各阶矩的权值,其值可以为任意实数值。 n 阶矩的加权和关于电路中某一元器件 R_i 的敏感度可以表示为如式(33)所示。

$$\frac{\partial \left(\sum_{i \in \text{Node}(T)} \lambda_i m_n^i \right)}{\partial R_i} = \left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(j)} (-C_k m_{n-1}^k) \right) + \frac{\partial \left(\sum_{k \in \text{Node}(T)} \eta_k m_{n-1}^k \right)}{\partial R_i} \quad (33)$$

可以看出, n 阶矩加权和关于 R_i 的敏感度可以看成两部分之和:其中第二部分是 $n-1$ 阶加权重矩关于 R_i 的敏感度。其中 η_k 为新的权值,其计算公式如式(34)所示:

$$\eta_k = -C_k \sum_{i \in \text{Path}(k)} R_i \left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \quad (34)$$

从式(33)可以看出,和矩的计算一样,矩加权和的敏感度计算也可以表示成递归的形式。如图 21 所示,第 n 阶加权重矩敏感度可以表示成第 $n-1$ 阶加权重矩敏感度与一个增量之和。

$$\begin{array}{ccc} \text{n阶加权重矩敏感度} & \text{增量部分} & \text{n-1阶加权重矩敏感度} \\ \frac{\partial \left(\sum_{i \in \text{Node}(T)} \lambda_i m_n^i \right)}{\partial R_i} & = \left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(j)} (-C_k m_{n-1}^k) \right) + & \frac{\partial \left(\sum_{k \in \text{Node}(T)} \eta_k m_{n-1}^k \right)}{\partial R_i} \end{array}$$

图 21 矩敏感度计算公式的分解

Figure 21 Decomposition of Nth Order Weighted Moment Sensitivity

同理, n 阶矩加权和关于 RC 树中某个电容 C_i 的敏感度也可以表示成递归的形式,如式(35)所示。

$$\frac{\partial \left(\sum_{i \in \text{Node}(T)} \lambda_i m_n^i \right)}{\partial C_i} = \left(-m_{n-1}^i \sum_{j \in \text{Path}(i)} R_j \left(\sum_{k \in \text{Des}(j)} \lambda_k \right) \right) + \frac{\partial \left(\sum_{k \in \text{Node}(T)} \eta_k m_{n-1}^k \right)}{\partial C_i} \quad (35)$$

观察式(33)和式(35)我们可以看出加权矩的敏感度计算表达式具有两个特点：

递推特性。即 n 阶加权矩敏感度可以递推到 $n-1$ 阶加权矩敏感度。递推特性使 n 阶加权矩敏感度的计算可以最终表示成 0 阶加权矩敏感度的计算。并且递推算法非常适合计算机实现。

遍历特性。在实际的优化过程中，我们通常需要计算的是加权矩关于所有 R ，或者所有 C 的敏感度，即：假设一个 RC 树中有 n 个 R ， n 个 C 。如果加权矩关于每个 R 或者每个 C 的敏感度都需要单独计算一遍，那么要计算加权矩关于所有 R 或者所有 C 的敏感度就需要反复计算 n 次。而利用式(3)和式(4)，计算加权矩关于所有 R 或者所有 C 的敏感度不需要重复遍历 RC 树。

本文接下来将详细介绍如何将式(33)和式(35)中的表达式表示成符号化形式，以及如何快速计算这些符号化表达式。

2.2.2 关于电阻元件的矩敏感度计算的符号化表示

我们知道在 RC 树中，各个节点的 n 阶矩加权和的关于电路中某个电阻 R_i 的敏感度表达式如(33)所示。

$$\partial \left(\sum_{i \in \text{Node}(T)} \lambda_i m_n^i \right) / \partial R_i = \left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(i)} (-C_k m_{n-1}^k) \right) + \partial \left(\sum_{k \in \text{Node}(T)} \eta_k m_n^k \right) / \partial R_i \quad (33)$$

我们将式(36)称为从第 n 阶加权矩到第 $n-1$ 阶加权矩递推的增量部分，

$$\left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(i)} (-C_k m_{n-1}^k) \right) \quad (36)$$

式(33)中， η_k 为新的权值，其值可以由式(34)求出。因此从第 n 阶矩敏感度递推到第 $n-1$ 阶矩敏感度的过程中，我们分别需要计算增量部分的值和新权值 η_k 的值。

下面以图 22 中所示的电路为例，说明如何将式(33)中的矩敏感度计算公式表示成符号化形式。假设我们需要计算图 22 中所有节点的二阶矩加权和关于电路中所有电阻的敏感度。那么计算过程主要包括以下几个操作步骤：

1. 计算图 22 所示电路中所有节点的一阶矩。
 2. 计算二阶加权矩敏感度递推到一阶加权矩敏感度的增量部分(如式 36 所示)。
 3. 计算一阶加权矩敏感度计算所需要的新权值(式 33 中 η_i 的值)。
 4. 计算一阶加权矩敏感度递推到零阶加权矩敏感度的增量部分。
 5. 将步骤 2 和步骤 4 所得到的结果相加，得到的值就是所要求的加权矩敏感度。
- 下面详细介绍，如何用符号化表达式来表示增量部分和新权值的计算。

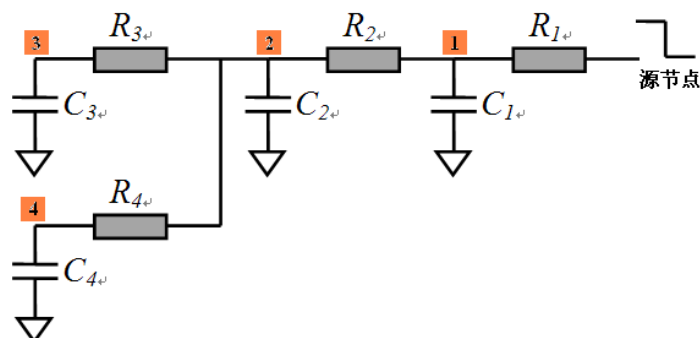


图 22 用于说明符号化表示的示例电路

Figure 22 An Circuit Example for Demonstrating Symbolic Representation

增量部分计算的符号化表达式

通过观察可以发现，增量部分的表达式(36)可以表示为图 23 所示的符号化形式。其中实线为加法运算，虚线表示乘法运算。线上的数字表示优先级，1 的优先级最高，4 的优先级最低。其中的各个节点的矩可以采用[13]中提到的符号化矩计算方法事先计算好，并将值保存在节点中。

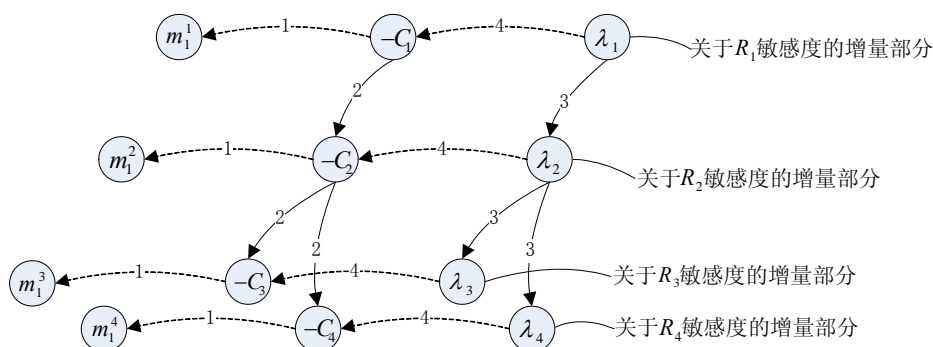


图 23 增量部分计算的符号化表示

Figure 23 Symbolic Representation for Incremental Part

假设我们需要计算 2 阶加权矩关于 R_2 敏感度的增量部分，那么根据图 23 我们得出符号化计算表达式如图 24 所示。其表示的计算表达式为：

$$(\lambda_2 + \lambda_3 + \lambda_4)(-C_2 m_1^2 - C_3 m_1^3 - C_4 m_1^4)$$

图中所表示的计算过程比较清晰，只要按照优先级顺序依次进行计算即可。加入我们需要计算关于所有 R 的敏感度的增量部分，我们只需要按照优先级顺序遍历图 23 中所示的符号化表达式即可。

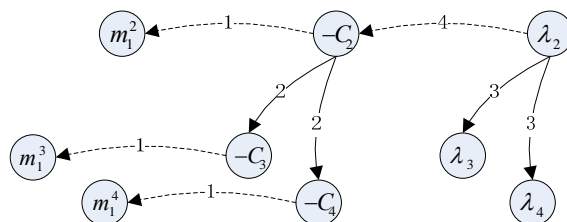


图 24 增量部分计算示例

Figure 24 An Example for Incremental Part Calculation

新权值的计算

根据加权矩敏感度计算的递推表达式(式 33), 每次降阶都需要计算新的权值 η_k 。第 n 阶加权矩敏感度到第 $n-1$ 阶加权矩敏感度递推过程中, 新权值 η_k 的计算表达式为如式(34)所示, 为:

$$\eta_k = -C_k \sum_{i \in Path(k)} R_i \left(\sum_{j \in Des(i)} \lambda_j \right)$$

该表达式很容易表示成符号化的形式如图 25 所示。与图 23 一样, 实线表示加法运算, 虚线表示乘法运算, 线上的数字表示优先级。其中 1 的优先级最高, 数字越大优先级越低。

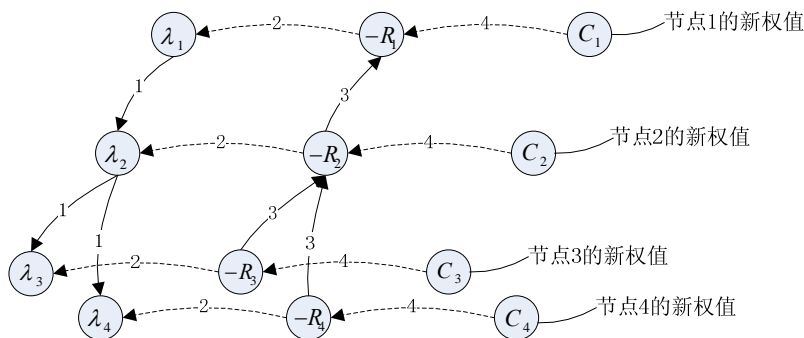


图 25 新权值计算的符号化表示

Figure 25 Symbolic Representation for New Weight Calculation

例如我们需要计算节点 2 的新权值, 那么计算的表达式可以表示成如图 26 所示。其所表示的计算表达式为

$$C_3(-R_1(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) - R_2(\lambda_2 + \lambda_3 + \lambda_4) - R_3(\lambda_3))$$

加入我们需要计算电路中所有节点的新权值, 我们只需要按优先级顺序计算所有

的乘法和加法即可。

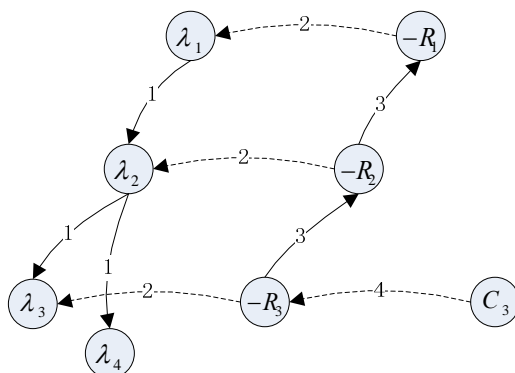


图 26 新权值计算的符号化表达式示例
Figure 26 An Example for New Weight Calculation

计算矩敏感度

计算矩敏感度是个递推的过程， n 阶加权矩敏感度的计算可以递推到 $n-1$ 阶加权矩敏感度的计算，直至递推到 0 阶加权矩敏感度（值为 0 ）的计算。其计算过程可以表示成一个树状的形式如图 27 所示。

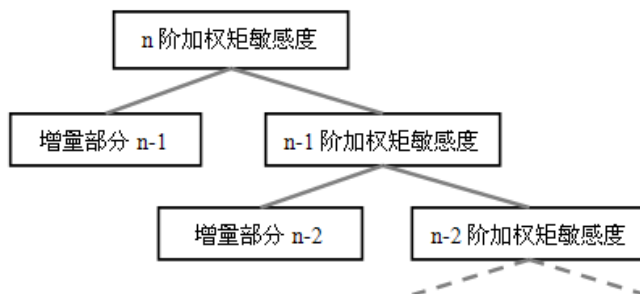


图 27 n 阶加权矩敏感度的计算过程
Figure 27 Calculation Flow of Nth order Weighted Moment Sensitivity

当需要计算 n 阶加权矩敏感度时，首先计算从 n 阶加权矩敏感度递推到 $n-1$ 阶加权矩敏感度的增量部分和 $n-1$ 阶加权矩敏感度的新权值，使原问题转化成 $n-1$ 阶加权矩敏感度的计算。

在每次的降阶过程中都会产生一个增量值，我们将 n 阶加权矩敏感度到第 $n-1$ 阶加权矩敏感度的递推过程中产生的增量部分即为 $H(n-1)$ 。那么 n 阶加权矩敏感度的值为： $H(0)+H(1)+ H(2)+\dots+H(n-1)$ 。

2.2.3 关于电容元件的矩敏感度计算的符号化表示

前面我们介绍了关于电阻元件的加权矩敏感度的符号化计算过程,关于电容元件加权矩敏感度的计算表达式如式(35)所示,为:

$$\frac{\partial \left(\sum_{i \in \text{Node}(T)} \lambda_i m_n^i \right)}{\partial C_i} = \left(-m_{n-1}^i \sum_{j \in \text{Path}(i)} R_j \left(\sum_{k \in \text{Des}(j)} \lambda_k \right) \right) + \frac{\partial \left(\sum_{k \in \text{Node}(T)} \eta_k m_n^k \right)}{\partial R_i}$$

可以看出,其表示形式基本上与关于电阻元件的加权矩敏感度计算表达式类似,即第 n 阶加权矩敏感度的计算可以递推到第 $n-1$ 阶。递推的过程中需要计算增量部分的值和新权值。其中新权值的计算表达式如式(37)所示:

$$\left(\sum_{j \in \text{Des}(i)} \lambda_j \right) \left(\sum_{k \in \text{Des}(i)} -C_k m_{n-1}^k \right) \quad (37)$$

其计算过程与电阻元件加权矩敏感度计算过程中的新权值计算完全相同。所不同的是增量部分的计算,从式(35)可以看出,从第 n 阶递推到第 $n-1$ 过程中需要计算的增量值为:

$$\left(-m_{n-1}^i \sum_{j \in \text{Path}(i)} R_j \left(\sum_{k \in \text{Des}(j)} \lambda_k \right) \right) \quad (38)$$

该增量值的计算可以表示成如图 28 所示的符号化表达形式,与上述的符号化表达式一样,实线表示加法运算,虚线表示乘法运算,线上的数字表示运算的优先级,1 的优先级最高,4 的优先级最低。

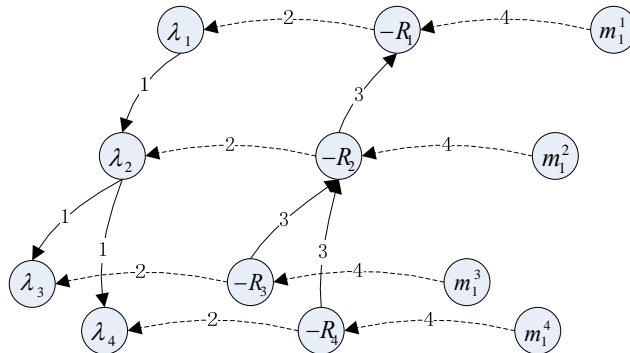


图 28 电容敏感度计算增量部分的符号化表示

Figure 28 Symbolic Representation of Incremental Part for Capacitance Sensitivity Computation

例如我们需要计算图 22 中节点 2 的增量部分,那么计算的表达式可以表示成如图 29 所示。其所表示的计算表达式为:

$$-m_1^2 (R_1 (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) + R_2 (\lambda_2 + \lambda_3 + \lambda_4))$$

如果我们需要结算关于所有电容的增量部分,只需要按优先级顺序遍历图 28 中

的所有边即可。

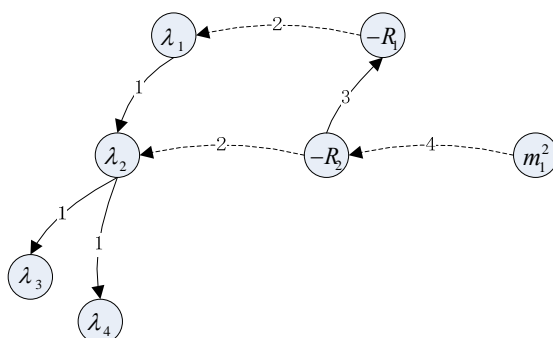


图 29 电容元件矩敏感度增量部分计算示例

Figure 29 An Example for Moment Sensitivity Computation

2.3 符号化矩敏感度计算的复杂度分析

传统的数值分析算法在矩敏感度计算时所需要的空间复杂度并不高，往往只需要存储原有的 RC 电路。但是数值算法需要反复遍历 RC 树，当 RC 树的规模越来越大时，数值计算方法会显得越来越力不从心。而章介绍的符号化矩敏感度计算方法在运算速度方面有很大的优势，并且消耗的内存和原 RC 树的大小成线性关系。

时间复杂度

假设我们需要计算某个 RC 电路 n 阶加权矩关于所有电阻元件的敏感度，假设该电路的规模为 N （电阻元件和电容元件的总和为 N ）那么所需要的计算总共包含以下几个部分：

1. 计算 RC 电路中各个节点的 $0 \sim (n-1)$ 阶矩，这个过程需要遍历节点规模为 $2(n-1)*N$ 的符号化表达式。
2. 在从 n 阶加权矩向 $n-1$ 阶加权矩递推的过程中，计算增量的过程需要遍历节点规模为 $3*N$ 的符号化表达式（如图 23 所示）。
3. 在从 n 阶加权矩向 $n-1$ 阶加权矩递推的过程中，计算新权值的过程需要遍历节点规模为 $3*N$ 的符号化表达式树（如图 25 所示）。

假设遍历一次节点规模为 N 的 RC 表达式树的时间复杂度为 $O(N)$ ，那么计算 n 阶加权矩关于 RC 树中所有 R 的敏感度需要时间复杂度为：

$$2(n-1)*O(N)+3n*O(N)+3n*O(N)=(8n-2)*O(N)$$

可以看出，计算的时间复杂度是与所需要计算的电路规模成线性关系，与所需要计算的阶数也成线性关系。

空间复杂度

显然意见，算法所需要的空间复杂度最大为 $(8n-2)*O(N)$ ，与所需要计算的电路规模成线性关系，与所需要计算的阶数也成线性关系。

只要适当的利用空间共享，算法的空间复杂度可以进一步进行优化，例如图 23、图 25 和图 28 中的权值是可以共享的，可以节省空间。除此之外还有其他一些可以共享的值。

实现最大程度的共享可以降低空间复杂度和时间复杂度，但是会带来程序复杂度的提高。

2.4 符号化敏感度计算的正确性测试

为了验证符号化矩敏感度算法的正确性，本文分别采用小规模测试电路和大规模测试电路对其进行了测试。小规模测试电路由于运算时间非常短，没有给出相应的运算时间；而对于大规模测试电路给出了优化时间。

2.4.1 小规模电路测试

对于如图 30 所示的小规模测试电路，首先对电路中的每根导线采用 π 模型等效，然后将符号化加权矩敏感度计算得出的结果与数值方法计算得出的结果进行比较。其中各个线段的线宽取值随机值。

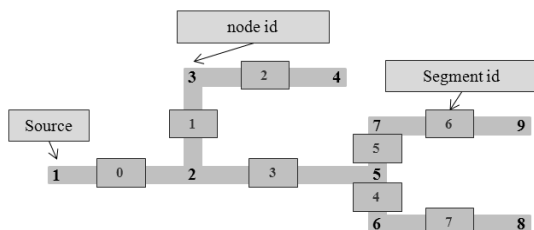


图 30 小规模互联线例子

Figure 30 A Small Case of Interconnects

两种方法得出的结果如图 31 所示。通过比较发现本报告中的方法得出的结果与

数值方法得出的结果非常接近，从而证明了符号化矩敏感度计算方法的正确性。

segment	width(um)	$\frac{\partial(m_2^4 + m_2^8 + m_2^9)}{\partial w_i}$		segment	width(um)	$\frac{\partial(m_3^4 + m_3^8 + m_3^9)}{\partial w_i}$	
		(My Program)	(Numerically)			(My Program)	(Numerically)
0	1.94	-1.16005E+08	-1.15939E+08	0	10.77	-1.40056E+12	-1.40058E+12
1	7.73	5.91888E+07	5.91892E+07	1	3.95	-2.02832E+12	-2.02837E+12
3	5.62	4.42384E+07	4.42397E+07	3	3.21	-1.01023E+12	-1.01045E+12
2	6.28	4.77271E+07	4.77274E+07	2	3.53	-1.65606E+12	-1.65610E+12
4	1.64	1.30792E+07	1.30849E+07	4	6.59	-8.82123E+11	-8.82129E+11
5	6.03	2.80533E+07	2.80535E+07	5	4.79	-1.01648E+12	-1.01649E+12
7	9.98	4.56166E+07	4.56168E+07	7	5.83	-1.62845E+12	-1.62846E+12
6	2.55	3.62858E+07	3.62875E+07	6	0.87	9.05059E+10	8.90005E+10

(a) 2 阶加权矩敏感度计算结果比较

(b) 3 阶加权矩敏感度计算结果比较

图 31 符号化矩敏感度计算方法和数值计算方法的比较

Figure 31 Comparison Between Symbolic Moment Sensitivity Computation and Numerical Moment Sensitivity Computation

2.4.2 敏感度与极值点的分析

为了从另一角度验证本报告中方法的正确性，我保持图 30 中所示 RC 树的其他边线宽不变，只改变 ID 号为 3 的边的线宽，在其线宽为不同值时计算加权矩的值。结果如图 32(A)所示，其中横坐标为 ID 号为 3 的边的线宽，纵坐标为加权矩的值。可以看出，在线宽取不同值时，电路中各阶矩的加权和存在一个最小值。图 32 (B) 中的横坐标为 ID 号为 3 的边的线宽，纵坐标为加权矩关于 ID 号为 3 的边的敏感度。从图 32 (A) 和图 32 (B) 得比较可以看出，A 图中最小值发生点的横坐标和 B 图中穿越 0 点的横坐标值基本接近。因此与预期完全符合，即极值点的敏感度为 0。

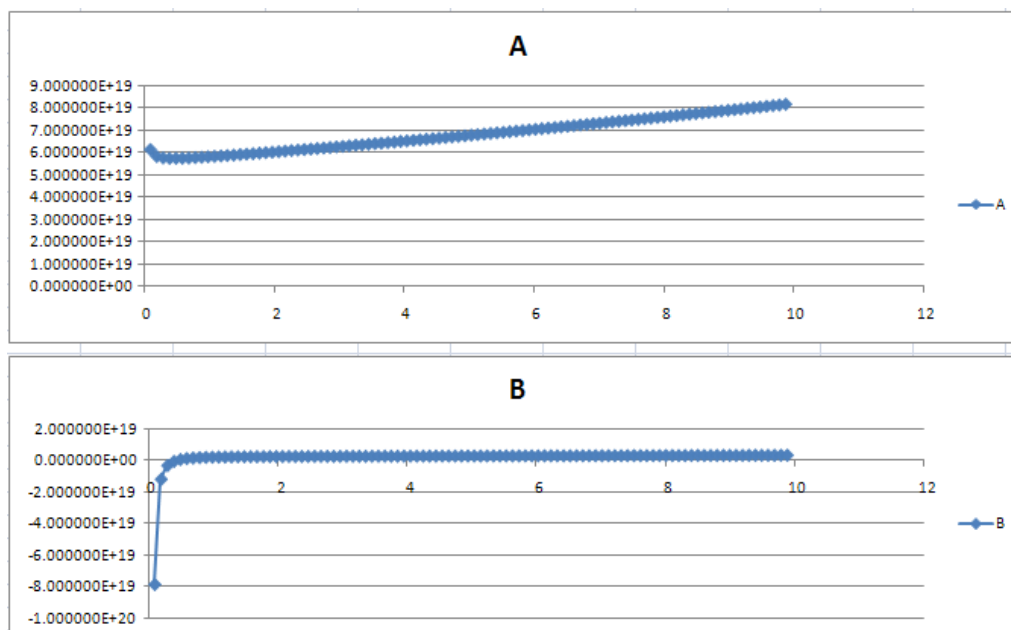


图 32 极值点的分析

Figure 32 Analysis of Maximum Value

2.4.3 大规模电路测试

为了进一步验证符号化矩敏感度算法的正确性和计算性能,本文还对大测试案例(100000 以上节点的电路)进行了测试,测试的方法也是将数值计算得出的结果和符号化计算得出的结果相比较,结果基本吻合,说明符号化矩敏感度计算是正确的。同时为了验证符号化矩计算过程的运算效率,本文还对大测试案例的运算时间进行了测试。测试机器配置为:主频 2.83GHz,双核 CPU,内存 8G。图 33 给出的是计算二阶加权矩敏感度情况下电路节点规模和计算时间的关系图。

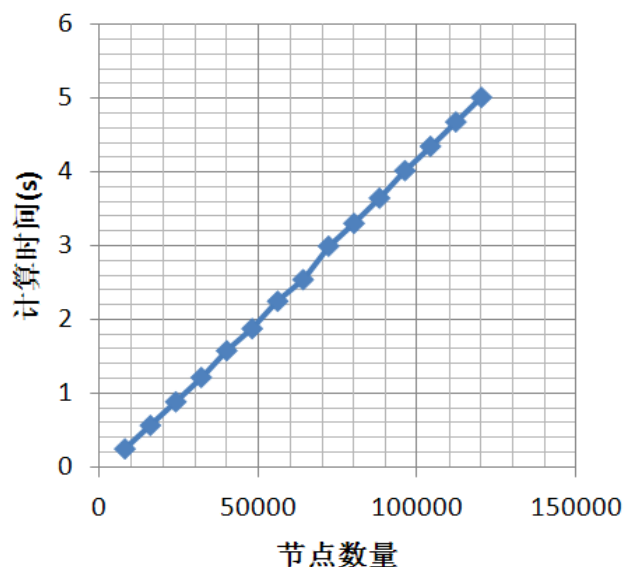


图 33 节点规模和计算时间的关系

Figure 33 The Relationship between Computation Time and Circuit Scale

从图 33 可以看出，计算时间随着节点数的增加而增加，两者呈线性关系。对于节点规模为 120000 的电路，符号化矩计算方法只需要消耗 5 秒左右的 CPU 时间。

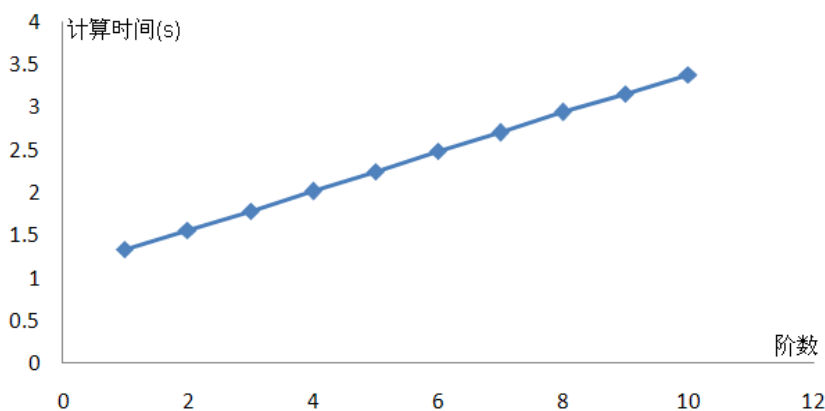


图 34 阶数和计算时间的关系

Figure 34 The Relationship between Moment order and Computation time

从图 34 可以看出，当电路的节点数量为 40000 时，计算时间和所需要计算矩敏感度的阶数呈线性关系。计算 10 阶加权矩关于电路中所有电阻元件的敏感度所需要的时间仅为 3.51。

第3章 基于符号化计算的物理优化

物理优化是指通过改变芯片的布局布线等物理参数来优化芯片的设计指标。本文中所提到的物理优化是指在布局布线已经完成之后,通过调整导线尺寸来达到优化目的的过程。

3.1 物理优化的数学模型

由于物理优化是一个需要全面考虑的问题,需要综合考虑多方面的指标,例如时延、面积、串扰等。优化的解往往不是某一性能指标达到最优,而是在保证达到关键指标的情况下尽可能优化非关键指标,最后达到满足所有条件的一个折中解。为了解决物理优化这个综合性的问题,我们需要为物理优化建立适当的数学模型。

3.1.1 优化问题的提出

在 VLSI 设计中,设计者需要保证各个节点的信号延迟在规定范围内;同时又希望布线导线本身有比较小的布线电容,以减轻驱动门的负载,从而节省功耗。因此物理设计要在保证各个节点的延迟在规定范围内的情况下,尽可能的减小导线的布线面积。我们将物理优化问题描述成(39)、(40)、(41)。

最小化:

$$O(W) = \sum_{i \in \text{Seg}(T)} w_i l_i \quad (39)$$

约束条件:

$$D(w_1, w_2, w_3, \dots, w_i) \leq D_{upper}, \forall i \in \text{Sink}(T) \quad (40)$$

$$W_{lower} \leq w_i \leq W_{upper}, \forall i \in \text{Sink}(T) \quad (41)$$

式(39)为优化目标,使物理布线面积最小; (40)保证各个终端节点的信号延迟在延迟上限 D_{upper} 内。与之前大部分优化算法[9][10]所不同的是, (40)式中的延迟采用 D2M 来估计节点的延迟,并且式子中的矩为相邻线网的信号跳转方向相反的情况下

当前节点的矩。(41)保证导线宽度在一定范围内。

3.1.2 带耦合串扰的信号延迟估计

对于单棵 RC 树，信号的延迟可以采用 D2M 来估计，其计算表达式为：

$$Delay(i) = \ln 2 \frac{(m_1^i)^2}{\sqrt{m_2^i}}$$

其中 $Delay(i)$ 表示节点 i 的延迟， m_1^i 和 m_2^i 分别表示节点 i 的一阶矩和二阶矩，其值可以用[13]中提到的符号化方法求出。但是前面我们有提到，在耦合电容不可忽略的情况下，当前线网的信号延迟会受到与其相邻线网的影响。当相邻线网的信号跳变方向与当前线网的信号跳变方向相反时，信号延迟会加大；当相邻线网的信号跳变方向与当前线网的信号跳变方向相同时，延迟变小。因此在估计线网上的信号延迟时，必须将相邻线网的跳变考虑在内。

式(7)需要保证信号的延迟在延迟上限范围内，因此我们在估计信号延时的时候需要考虑延时最大的情况，即相邻线网上的信号跳变方向与当前线网上的信号跳变方向相反的情况。

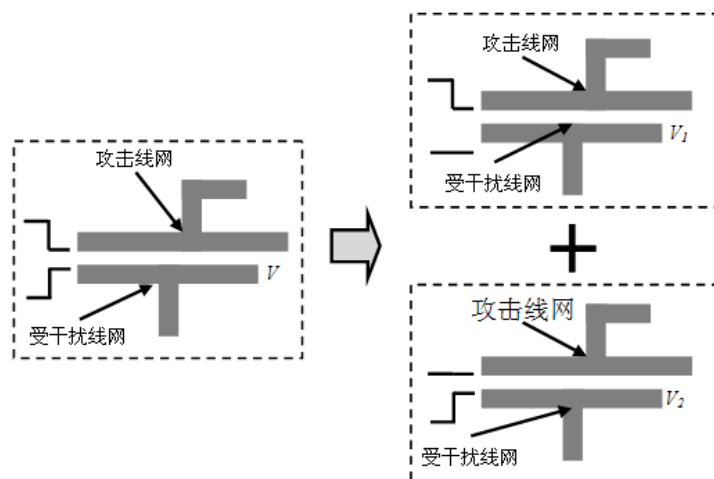


图 35 耦合情况下节点波形的叠加计算

Figure 35 Additivity of Signal When Account segment coupling

我们可以采用电路的叠加原理来处理相邻线网都有信号跳变的情况。图 35 中，受干扰线网上的信号负跳变，攻击线网上的信号正跳变，我们需要估计受干扰线网上某一点的信号 V 。我们可以将这种情况看成以下两种情况的叠加：第一种情况是受干

扰线网上没有信号跳变，而攻击线网上的信号负跳变，记这种情况下节点上的电压值为 V_1 ，第二种情况是攻击线网上的信号没有跳变，而受干扰线网上的信号正跳变，记这种情况下节点的电压为 V_2 ，真实情况下的节点电压是这两种情况的叠加：即 $V=V_1+V_2$

将 V 、 V_1 和 V_2 拉普拉斯展开得：

$$(m_0 + m_1s + m_2s^2 + \dots) = (m_0^1 + m_1^1s + m_2^1s^2 + \dots) + (m_0^2 + m_1^2s + m_2^2s^2 + \dots)$$

匹配各阶系数可得式(42)

$$m_0 = m_0^1 + m_0^2, m_1 = m_1^1 + m_1^2, m_2 = m_2^1 + m_2^2, \dots \quad (42)$$

式(42)告诉我们，如果我们需要计算电路当前状态下的各阶矩，我们可以根据叠加原理将电路的状态分解，然后将各个状态下电路各阶矩分别相加，即可得到当前状态的各阶矩。

假设我们需要计算耦合情况下受干扰线网最坏延迟情况下某个节点的矩，计算方法如下：

1. 假设受干扰线网处于静止状态，攻击线网信号处于负跳变状态。计算此时受干扰线网节点的各阶矩： $m_0^1, m_1^1, m_2^1, \dots$ 。
2. 假设攻击线网处于静止状态，受干扰线网的信号处于正跳变状态，计算此时受干扰线网节点的各阶矩。 $m_0^2, m_1^2, m_2^2, \dots$

3. 根据叠加原理最后求得在最坏延时情况下，受干扰节点的各阶矩为：

$$(m_0^1 + m_0^2), (m_1^1 + m_1^2), (m_2^1 + m_2^2), \dots$$

得到了最坏延时情况下节点的各阶矩，接下来就可以利用各种延时估计方法（例如 D2M, Elmore 等）对该节点的信号延时进行估计了。

3.1.3 优化问题的数学表示

我们已经知道，互联线中某个节点的信号延时可以采用 D2M 来估计，将其带入约束条件(40)可得：

$$\ln 2 \frac{(m_1^i)^2}{\sqrt{m_2^i}} \leq D_{upper}$$

该式子中由于带有根号和分式, 因此在作为约束条件进行优化时会带来计算上的不便, 为了降低计算的复杂度, 对该式子进行适当的变形可得:

$$(m_1^i)^4 \leq \left(\frac{D_{upper}}{\ln(2)} \right)^2 m_2^i$$

因此原优化问题可以表示为:

最小化:

$$O(W) = \sum_{i \in \text{Seg}(T)} w_i l_i \quad (43)$$

约束条件:

$$(m_1^i)^4 \leq \left(\frac{D_{upper}}{\ln(2)} \right)^2 m_2^i, \forall i \in \text{Sink}(T) \quad (44)$$

$$W_{lower} \leq w_i \leq W_{upper}, \forall i \in \text{Sink}(T) \quad (45)$$

式(43)、(44)和(45)所示的就是我们要求解的优化问题。接下来我们就来介绍如何利用之前提出的符号化矩敏感度计算方法, 结合拉格朗日松弛法来求解上述优化问题。

3.2 拉格朗日松弛法求解优化问题

拉格朗日松弛法通过引入拉格朗日乘子来减少约束条件个数, 从而求解原问题的最优解。对(43)、(44)和(45)中的约束条件引入拉格朗日乘子, 原优化问题可以转化为(46)和(47)的形式。

最小化:

$$SO(W) = \sum_{i \in \text{Sink}(T)} w_i l_i + \sum_{i \in \text{Sink}(T)} \lambda_i \left((m_1^i)^4 - \left(\frac{D_{upper}}{\ln(2)} \right)^2 m_2^i \right) \quad (46)$$

约束条件:

$$L \leq w_i \leq U, \lambda_i > 0 \quad (47)$$

其中 D_{upper} 为节点信号延迟的上限， λ_i 为拉格朗日乘子，其值为大于 0 的实数。

从式(46)我们可以看出：当式(44)中的约束条件不满足时，即 $(m_1^i)^4 > \left(\frac{D_{upper}}{\ln(2)}\right)^2 m_2^i$ 时，

会增大式(46)中目标函数 $SO(W)$ 的值。相当于加了一个“惩罚”在目标函数上。

拉个朗日的求解就是找到一组 λ_i 的值，使得式(46)和式(47)所表示的优化问题的解就是原问题的解。关于拉格朗日松弛法求解过程的基础知识，请参考附录一。总的来说拉格朗日松弛法的求解可以分为子问题的求解和原问题的求解，子问题的求解是指固定权值 λ_i 的情况下，调整线宽使式(46)中的目标函数最小。原问题的求解是指调整权值 λ_i ，使拉格朗日子问题的最优解逼近原优化问题的最优解。拉格朗日松弛法求解优化问题的流程如图 36 所示。

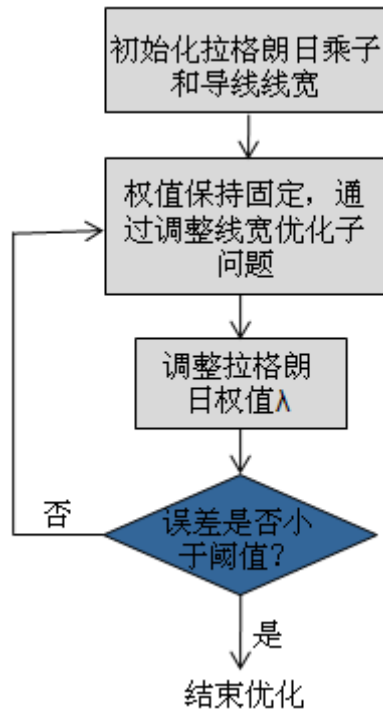


图 36 拉格朗日松弛法求解过程

Figure 36 How to Solve The Optimization Problem Using Lagrangian Relaxation

3.3 拉格朗日子问题的求解

(9)称为拉格朗日子问题，求解拉格朗日子问题一般通过梯度法求解，我们将求

解拉格朗日问题的过程描述为如下形式：

拉格朗日问题求解算法

输入：物理布线树的 RC 模型，拉格朗日乘子 $\lambda_1 \sim \lambda_i$ ，线宽的上下限 W_{upper} 、 W_{lower} 和节点延迟上限 D_{upper}

输出：使式(9)取得最小值的 $w_1 \sim w_i$

S1: 初始化线宽 $w_1 \sim w_i$

S2: 根据布线树和线宽 $w_1 \sim w_i$ 的值计算 RC 模型

S3: 对于 RC 树中任意节点 i 计算 η_i 、 μ_i 的值：

$$\eta_i = 4\lambda_i (m_i^i)^3, \forall i \in \text{Node}(T)$$

$$\mu_i = \lambda_i \left(\frac{D_{upper}}{\ln(2)} \right)^2, \forall i \in \text{Node}(T)$$

S4: 计算 (9) 关于各个导线宽度的梯度

$$\frac{\partial(SO(W))}{\partial w_i} = l_i + \frac{\partial \left(\sum_{i \in \text{Sink}(T)} \eta_i m_i^i \right)}{\partial w_i} + \frac{\partial \left(\sum_{i \in \text{Sink}(T)} \mu_i m_i^i \right)}{\partial w_i}$$

S5: 根据敏感度向量调整导线的线宽

$$w_i = \begin{cases} \text{Min}\{w_i + \Delta W, W_{upper}\}, & \text{if} \left(\frac{\partial(SO(W))}{\partial w_i} < 0 \right) \\ \text{Max}\{w_i - \Delta W, W_{lower}\}, & \text{if} \left(\frac{\partial(SO(W))}{\partial w_i} > 0 \right) \\ w_i, & \text{if} \left(\frac{\partial(SO(W))}{\partial w_i} = 0 \right) \end{cases}$$

S6: 重复 S2-S6 直到 $SO(W)$ 近似取到最小值

在拉格朗日问题的求解过程中，拉格朗日乘子是固定的。求解过程的输入为拉格朗日乘子和线宽的初始值，子问题的求解为通过调整线宽使优化目标最小。在 S4 的求解梯度过程中，需要求解电路节点一阶矩和二阶矩关于电路中所有导线宽度的敏感度，该求解效率直接影响了整个优化算法的效率。传统的 RC 树状电路的矩和矩敏感度可以用反复遍历 RC 树的方式求出。本文利用符号化的方法来求解矩的敏感度，首先根据 RC 网络构造矩敏感度计算的符号化表达式，每次计算时，只需要遍历一次该表达式即可。利用符号化方法求解矩的敏感度大大提高了求解的效率。

3.4 原优化问题的解

拉格朗日松弛法的主要原理是通过调节拉格朗日乘子,使拉格朗日问题的最小值最大化。优化模型的拉格朗日求解的过程描述如下:

拉格朗日最优化算法

输入: 物理布线树

输出: 使(6)最小且满足条件(7)(8)的线宽

S1: 初始化线宽 $w_1 \sim w_i$ 的值

S2: 初步估算拉格朗日乘子的值,以减少拉格朗日乘子在优化过程中调整的次数。

$$\lambda_1 = \lambda_2 = \dots = \lambda_n = \frac{\sum_{i \in \text{Sink}(T)} (m_i)^4}{\sum_{j \in \text{seg}(T)} (w_j l_j)}$$

S3: 计算(9)关于各个拉格朗日乘子的梯度

$$\frac{\partial(SO(W))}{\partial \lambda_i} = (m_i)^4 - \left(\frac{D_{upper}}{\ln(2)} \right)^2 m_i^2$$

S4: 调整拉格朗日乘子的值从而增大拉格朗日问题的最小值。

$$\lambda_{i+1} = \begin{cases} \lambda_i + \varpi_i, \text{ if } \left(\frac{\partial(SO(W))}{\partial \lambda_i} > 0 \right) \\ \text{Max}\{0.9\lambda_i, \lambda_i - \varpi_i\}, \text{ if } \left(\frac{\partial(SO(W))}{\partial \lambda_i} < 0 \right) \\ \lambda_i, \text{ if } \left(\frac{\partial(SO(W))}{\partial \lambda_i} = 0 \right) \end{cases}$$

其中 $\varpi_1, \varpi_2, \dots, \varpi_i$ 满足:

$$\varpi_i \leq \varpi_{i+1}, \text{ Limit } \varpi_i = 0, \sum_{i=0}^{\infty} \varpi_i \rightarrow +\infty$$

S5: 调用拉格朗日问题求解算法计算拉格朗日问题的最小值

S6: 重复 S2-S6, 直至满足

$$SO(W) - \sum_{i \in \text{Sink}(T)} w_i l_i < \text{ErrorBound}$$

拉格朗日原问题求解过程的输入是物理布线树,输出的是使原问题(式 43、44、45 所示)最优化的导线线宽值。S2 中初始化拉格朗日乘子的目的是为了使得拉格朗日乘子有个合理的预估值,以使接下来的调整次数较少。S4 中,调整拉格朗日乘子的

方法有很多种，S4 中所选用的方法是通过多次实验得出，既具有较好的收敛速度，又能够保证求解过程的收敛。

在拉格朗日原问题的求解过程中，需要反复调用子问题的求解过程，因此子问题求解效率会直接影响到朗格朗日原问题的求解效率。

第4章 优化结果的测试与分析

在前面的章节中,我们介绍了如何利用符号化矩敏感度计算方法对更为精确的物理优化模型进行求解。为了验证优化结果的正确性和优化算法的效率,本文分别采用本优化算法对小测试案例和大测试案例进行了优化,并将优化结果与 HSpice 仿真结果进行了比较。

4.1 测试分析方法

Hspice 仿真需要的是 RC 网表,而优化算法的输入为 VLSI 物理布线结果。图 37 说明了如何将布线结果转化为 RC 网表。

- 首先对于输入布线结果中的每一个独立导线段(以终端节点或者分支节点为端点的导线段)采用 π 模型进行等效,如图 37(b)所示,其中 π 模型中 RC 的计算公式如式(4)和(5)所示。
- 如果得到的电路中某些节点上并联有多个电容(例如节点 2 上并联了三个电容),则将这些电容合并,得到的电路如图 37(c)即是我们需要的 RC 树状电路。
- 图 37(c)所示电路对应的网表即可输入 Hspice 进行仿真,测试节点的延迟等信息。

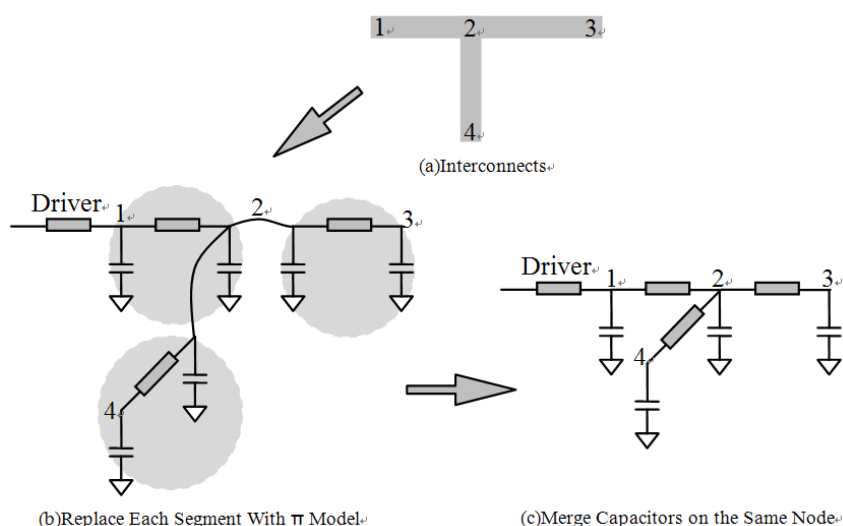


图 37 将布线结果转化成 RC 树

Figure 37 Convert Routing Pattern into RC Tree

4.2 小测试案例的优化结果分析

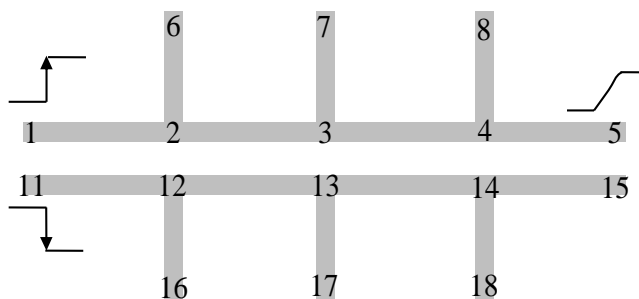


图 38 互相耦合的小规模布线树

Figure 38 Small Scale Routing Tree With Coupling Effects

图 38 所示的是一个小规模测试案例，其中节点 1 和节点 11 分别是两个线网的信号源，节点 5, 6, 7, 8, 15, 16, 17, 18 分别是线网的终端节点。本文对该案例采用上述优化算法进行优化。优化过程中采用的参数为：

延迟上限： 25ns。

导线线宽范围： 0.18 μm ~9.99 μm

驱动电阻值： 50 Ω

导线可取线宽值：

$$w_i = W_{lower} + k \frac{(W_{upper} - W_{lower})}{1000}, k = 1, 2, 3 \dots 1000$$

图 38 所示案例优化前后节点最坏延迟值（HSpice 仿真得出）的比较如图 39 所示。这里的最坏延迟是指在节点 11 负跳变、节点 1 正跳变的情况下，节点 5, 6, 7, 8 的信号延迟。从图 39 可以看出：

1. 在未作优化时各导线段线宽取最小值，此时节点的延迟远远超出了延迟上限 (25ns)。
2. 不考虑干扰信号跳变进行优化得到的布线面积比考虑干扰信号跳变进行优化得到的布线面积要小，这是因为考虑干扰信号跳变时条件限制更加严格。
3. 考虑了干扰信号跳变的优化结果虽然布线面积稍大，但节点延迟能够保证在延迟上限范围内。而不考虑干扰信号跳变进行优化得到的优化结果中，其节点最坏延迟可能会超出延迟上限(25ns)。
4. 优化结果中延迟最大的终端节点的信号延迟都非常接近于延迟上限 25ns。这是很合理的结果：因为如果终端节点的延迟还没有达到延迟上限，那么线宽还有再调

整的“余量”，当前的优化结果可能不是最优的。

5. 在优化过程中，取较小的线宽间隔会取得更好的优化结果，但是会增加优化问题求解的时间。本例中的线宽间隔所达到的优化结果已经非常接近连续线宽所达到的优化结果(最大延迟节点的延迟值非常接近于 25ns)

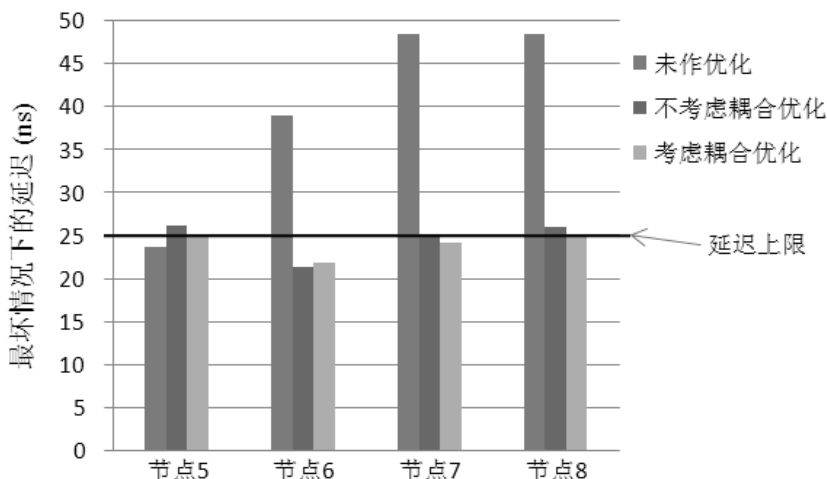


图 39 优化前后节点最坏延迟的比较

Figure 39 Compare Signal Delays Under the Worst Cases for Routing Tree before Optimization and After Optimization

优化前后布线面积的比较如图 40 所示，考虑攻击线网上信号跳变的影响进行优化得到的最终面积($77.42\mu\text{m}^2$)要比不考虑攻击线网上信号跳变的影响进行优化所得到的布线面积($66.27\mu\text{m}^2$)略大，但是两者相差不大。

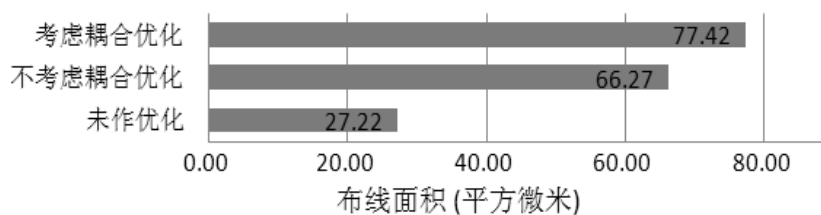


图 40 优化前后面积的比较

Figure 40 Routing Areas Comparison

对以上简单小案例的测试表明，本优化算法的结果在预期之中，考虑最坏延迟情况下的优化能够保证信号延迟在延迟上限以内，在布线面积方面也没有太大的代价。

4.3 大测试案例的优化结果分析

从小规模电路测试结果,我们看到了基于拉格朗日松弛法得出的优化解具有较好的合理性。为了进一步验证优化算法运行的效率和正确性,本文对如图 41 所示的较大规模的测试案例进行了测试。图 41 所示的是两个互相耦合的线网,负跳变的线网为攻击线网,正跳变的线网为受干扰线网,两个线网很多导线段之间存在耦合。

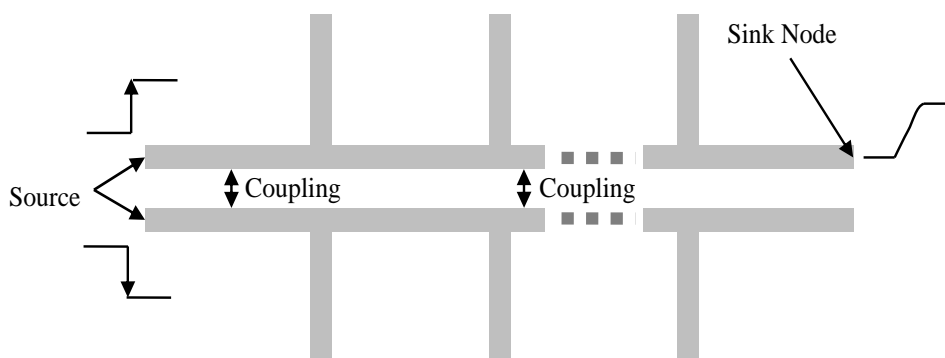


图 41 互相耦合的大规模布线树

Figure 41 Large Scale Routing Trees With Coupling Effects

之所以选用图 41 所示电路作为测试案例,是因为其具有普遍代表性:

1. 该测试案例的终端节点离源节点距离从近到远都有分布,延迟较大节点和延迟较小节点的优化情况都能够反应出来。
2. 该测试案例的两个线网之间存在很多耦合导线段,当将这些导线段之间的距离调成很大时,就相当于两个线网之间没有存在耦合情况;而当将这些导线段之间的距离调成很小时,就相当于两个线网之间强耦合的情况。
3. 图 42 给出了两种其他的测试案例,可以看出图 42(a)和图 42(b)在拓扑结构上是和图 41 中的电路类似的。只要适当调节图 41 中各个耦合线段的间距,导线重合长度以及导线段的长度就能与其他很多电路等效。
4. 采用图 41 作为测试案例还有一个重要原因是其构造非常简单,只要适当的将其节点编号,就能够根据编号判断出当前节点是靠近源节点还是远离源节点,是与其他导线存在耦合还是没有耦合。

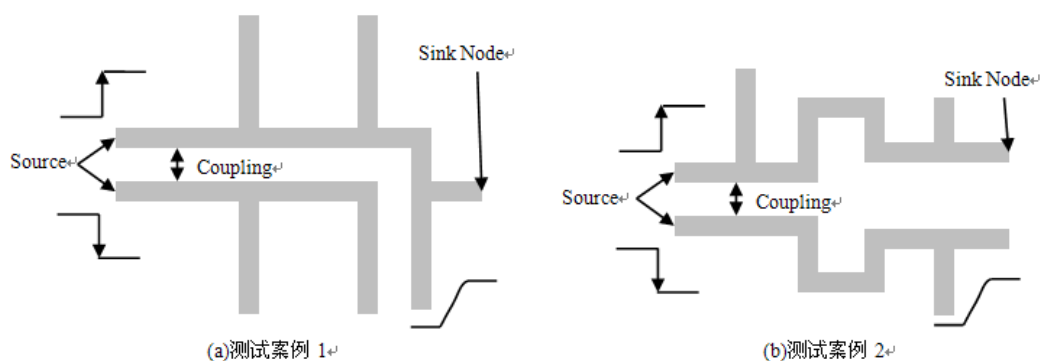


图 42 其他可能的测试案例
Figure 42 Other Possible Test Cases

大测试案例到 RC 网表的转换采用的也是图 37 种所示的过程。首先对于输入布线结果中的每一个独立导线段（以终端节点或者分支节点为端点的导线段）采用 π 模型进行等效，如果得到的电路中某些节点上并联有多个电容，则将这些电容合并并将图 41 中所示的布线结果进行转化所得到的 RC 电路如图 43 所示。

运行优化程序的测试机器主频为 2.83GHz 的，双核 CPU，内存 8G。

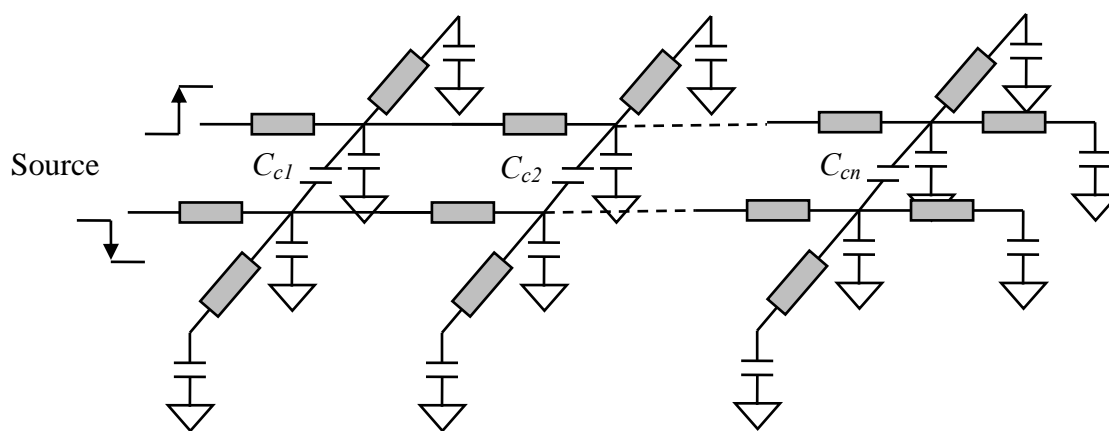


图 43 测试案例的 RC 模型
Figure 43 RC Model of Test Case

4.3.1 优化所需时间分析

优化算法的优化时间直接影响到优化算法所能处理的电路规模，为了测试算法的时间复杂度，本文对图 41 所示具有 400~3200 个节点的测试电路进行了优化。优化时所取的基本参数如下：

导线线宽范围：0.18 μm ~1.99 μm

驱动电阻值：50 Ω

导线可取线宽值：

$$w_i = 0.18 + k \frac{(1.99 - 0.18)}{100}, k = 1, 2, 3 \dots 100$$

测试案例规模和优化所需时间的关系如图 44 所示，其中横坐标为电路的节点数量，纵坐标为优化所需时间。可以看出，优化所需时间基本上是和节点的规模呈线性关系。当电路的规模为 400 个节点时，优化所需时间大约为 50 秒；当电路的规模增加到 3200 个节点时，优化所需要的时间大约为 9 分钟左右。

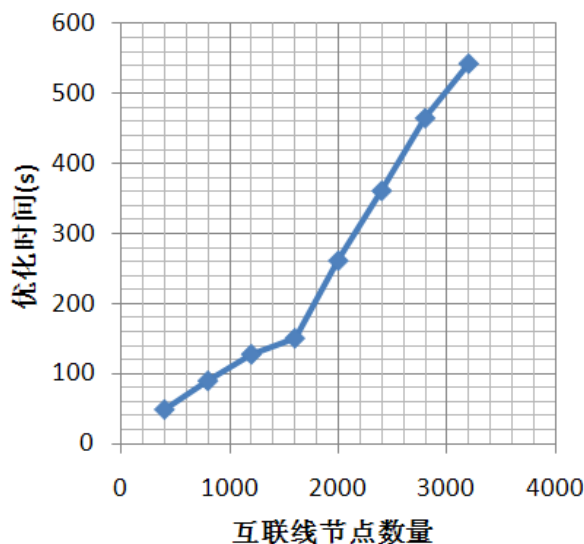


图 44 不同电路规模的优化时间

Figure 44 Time Consumption of Optimization for Different Circuit Scales

当然优化时间与优化精度、导线线宽取值间隔、最大延迟上限的取值有关。例如增加优化精度显然会增加优化的时间。但是不管怎样图 44 说明，基于拉格朗日松弛法，采用符号化矩敏感度的优化算法在处理较大规模电路的优化问题时是可行的。

4.3.2 优化前后节点延迟的比较

物理优化中，需要保证各个节点的延迟在延迟上限范围内。为了验证本优化算法在控制节点延迟方面的有效性，本文对图 41 所示具有 3200 个节点的测试电路进行了优化。优化过程中所取的参数为：

延迟上限： 16000ns。

导线线宽范围： $0.18\mu\text{m}$ - $1.99\mu\text{m}$ ，

导线可取线宽值：

$$w_i = 0.18 + k \frac{(1.99 - 0.18)}{100}, k = 1, 2, 3 \dots 100$$

优化所用的时间为 510 秒。图 45 给出了优化前后电路节点的延迟比较。其中横坐标为节点的延迟(单位为 10000ns)，纵坐标为节点数。可以看出，在优化之前，电路中节点的最大延迟可达 100000ns，远远超出延迟的最大上限 16000ns，并且节点的延迟在 0~100000ns 几乎均匀分布。

采用本文中算法进行优化之后，所有节点的延迟被限制在延迟上限 16000ns 以内，并且很多节点的延迟接近延迟上限的值。这说明优化算法在保证节点延迟的同时，充分利用了设计余量。

优化之前所有导线取最小线宽 $0.18\mu\text{m}$ ，此时导线面积为 $6209.136\mu\text{m}^2$ 。优化后的导线面积为 $21656.35\mu\text{m}^2$ ，仅仅比所有导线为最小线宽时的面积增加了 3 倍。而最大延迟足足缩小了 6 倍左右。

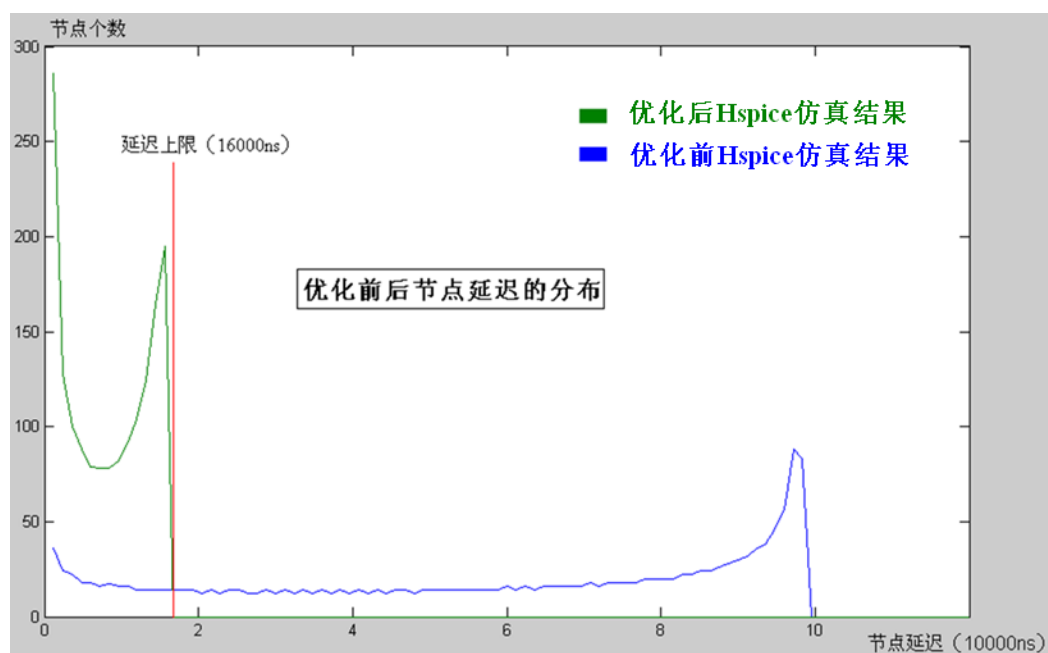


图 45 优化前后节点延迟的分布图

Figure 45 Delay Distribution for Circuits before Optimization and after Optimization

表 2 列出了对于不同节点规模的电路优化前后的面积比较。可以看出，为了保

证节点的最大延迟在延迟上限以内，优化算法对线宽进行了调整，从而导致导线面积的增加。导线面积具体增加的量和延迟上限的取值有关，当延迟上限非常小时，优化算法需要使劲全力减小节点的最大延迟，必然会导致布线面积的增加。当延迟上限非常大时，节点最大延迟肯定能保证在延迟上限范围内，优化算法会尽量缩小布线面积，因此得到的最终布线面积会较小。

表 2 优化前后的面积比较

电路节点数	最小线宽面积(μm^2)	优化后面积(μm^2)	面积增加百分比
400	765.936	3772.8094	392.575%
800	1543.536	5488.7134	255.593%
1200	2321.136	8907.2482	283.745%
1600	3098.736	8588.2054	177.152%
2000	3876.336	10627.0463	174.152%
2400	4653.936	14296.1825	207.185%
2800	5431.536	18509.9299	240.786%
3200	6209.136	21656.3566	248.782%

4.3.3 最大延迟上限和优化面积的关系

优化结果面积和最大延迟上限是个相互制衡的量，当最大延迟上限足够大时，面积调整所受的限制很小，这时优化所得到最终优化面积会比较小；而当最大延迟上限比较小时，这时必须增大面积来尽可能满足最大延迟上限的要求，这种情况下所得到的优化面积肯定相对较大。

本文对具有 2800 个节点的电路进行了优化，所取的参数为：

导线线宽范围： $0.18\mu\text{m}-1.99\mu\text{m}$ ，

导线可取线宽值：

$$w_i = 0.18 + k \frac{(1.99 - 0.18)}{100}, k = 1, 2, 3 \dots 100$$

优化过程中分别对延迟上限取不同的值，得到的优化面积和延迟上限的关系如图 46 所示。可以看出，优化面积随着延迟上限的增加而减小。

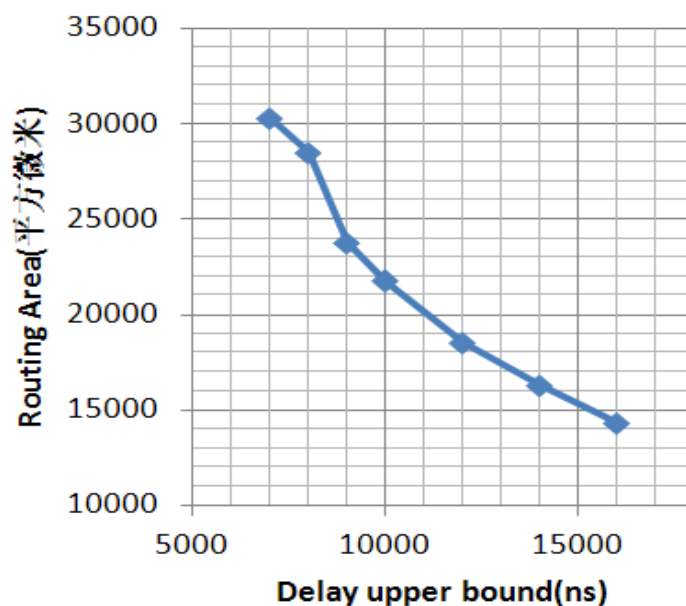


图 46 优化面积和延迟上限的关系

Figure 46 The relationship between Optimization Area and Delay Upper Bound

4.3.4 考虑串扰与不考虑串扰的比较

在物理布线中，当前线网上的信号延迟还受到与之相邻线网上信号跳变的影响，当相邻线网上的信号跳变与当前线网的信号跳变方向相反时，当前线网上的信号延迟会变大。本文的算法中充分考虑了相邻线网上信号跳变对当前线网的影响，使当前线网在最坏情况下的信号延迟保持在延迟上限以内。

为了说明是否考虑相邻线网之间的耦合对优化结果的影响，本文对图 41 所示具有 3200 个节点的测试电路进行了两次优化，其中一次考虑了干扰线网上信号跳变对受干扰线网信号延迟的影响，第二次没有考虑干扰线网上信号跳变对受干扰线网信号延迟的影响。优化过程中所取的参数为：

延迟上限： 15000ns。

导线线宽范围：0.18 μ m-1.99 μ m，

导线线宽可取值：

$$w_i = 0.18 + k \frac{(1.99 - 0.18)}{100}, k = 1, 2, 3 \dots 100$$

优化的结果如图 47 所示。图中横坐标为 HSpice 仿真得出的节点在最坏延迟情况下（相邻线网上的信号跳变方向与当前线网的信号跳变方向相反）的延迟。纵坐标

为节点数量。可以看出，考虑了干扰线网上信号跳变所得出的优化结果，所有节点的最坏延时都在延时上限以内(15000ns)。而没有考虑干扰线网上信号跳变所得出的优化结果中，有相当一部分节点的最坏延迟已经超过了延迟上限。可见在深亚微米工艺下，耦合电容不可忽略的情况下必须在优化时考虑相邻线网之间信号串扰的影响，否则得出的优化结果极有可能违反时序约束条件。

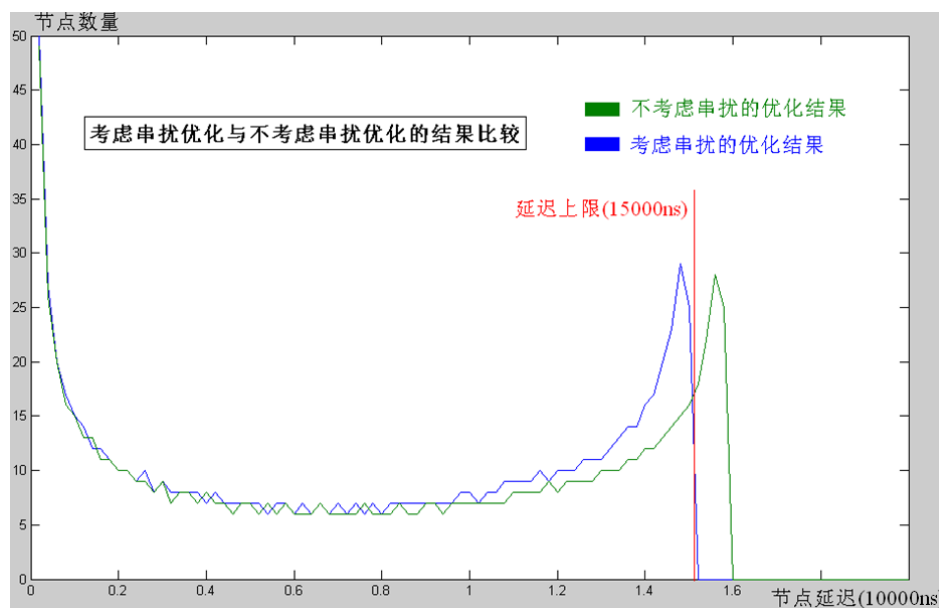


图 47 是否考虑相邻线网耦合对优化结果的影响

Figure 47 Compare Optimization Accounting Coupling to Optimization That don't Account Coupling

表 3 给出了考虑串扰情况进行优化所得到的布线面积与不考虑串扰情况进行优化所得到的布线面积的比较。从表中可以看出，考虑串扰情况进行优化所得到的布线面积会稍大，这是因为考虑串扰情况时的限制条件更加严格。但是两者的面积相差并不大，考虑串扰进行优化相对于不考虑串扰进行优化，面积增加的量大约为 8% 以内。因此考虑串扰情况所进行的优化并不会对成本带来多大的影响。

表 3 是否考虑串扰对优化结果面积的影响

节点数	不考虑串扰优化后面积 (μm^2)	考虑串扰优化后面积 (μm^2)	面积增加百分比
400	3772.8094	4041.0079	7.109%
800	5488.7134	5690.0578	3.668%
1200	8907.2482	9283.7426	4.227%
1600	8588.2054	8975.2558	4.507%
2000	10627.0463	10827.2138	1.884%
2400	14296.1825	14874.0214	4.042%
2800	18509.9299	19151.8862	3.468%
3200	21656.3566	22687.3181	4.761%

4.3.5 基于 Elmore 估计和 D2M 估计的比较

过去的优化算法中，基本上都采用 Elmore 作为延迟估计，究其原因有两点：Elmore 延迟为信号真实延迟的上限，利用 Elmore 作为延迟估计可以确保真实的信号延迟在延迟上限范围内；另外一个原因是 Elmore 计算形式非常简单。

但是 Elmore 作为延迟估计也有一些缺点：

1. Elmore 延迟作为一阶估计，不能将相邻线网上的信号跳变考虑在内。虽然理论上证明对于树状电路的情况，Elmore 延迟是真实延迟的上限，但是不能保证在相邻线网之间的串扰存在的情况下依然如此。
2. Elmore 延迟在估计延迟时有较大的冗余，这使得采用 Elmore 作为延迟估计的优化算法所得到的结果有较大的设计余量。

为了说明采用 D2M 作为延迟估计的必要性，本文对图 41 所示具有 3200 个节点的测试电路进行了两次优化，其中一次采用 Elmore 作为延迟估计，另外一次采用 D2M 作为延迟估计。优化过程中所取的参数为：

延迟上限： 15200ns。

导线线宽范围： $0.18\mu\text{m}-1.99\mu\text{m}$ ，

导线线宽可取值：

$$w_i = 0.18 + k \frac{(1.99 - 0.18)}{100}, k = 1, 2, 3 \dots 100$$

图 48 给出了优化结果的比较，其中横坐标为节点的延迟(单位：10000ns)，纵坐标为节点数量。从优化结果可以看出，采用 Elmore 延迟作为延迟估计所得到的结果中，节点的延迟值离最大延迟上限还有较大的余量，因此从直观上理解，电路应该还有进一步优化的空间；而采用 D2M 延迟估计所得到的优化结果中，节点的延迟非常逼近最大延迟上限，几乎将电路优化到“极致”。

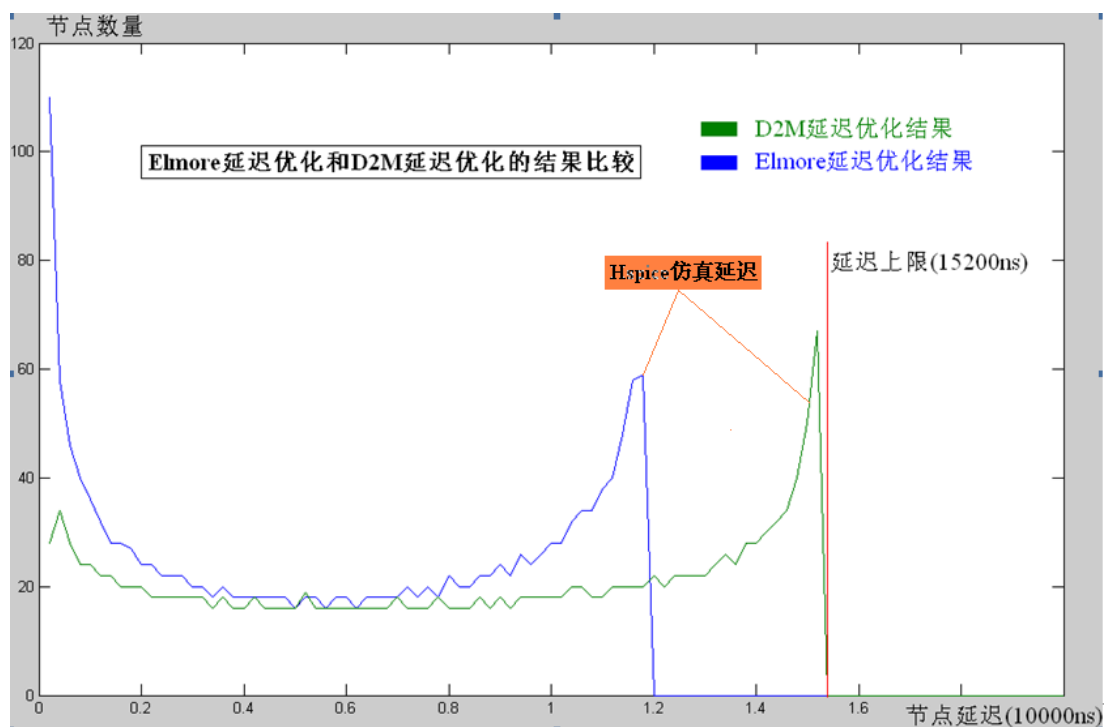


图 48 Elmore 延迟优化和 D2M 延迟优化的结果比较

Figure 48 Optimization using Elmore Delay Metric Versus Optimization using D2M Delay Metric

由于采用 Elmore 作为延迟估计有较大的设计余量，因此对于相同的延时约束而言，采用 Elmore 估计进行优化所消耗的资源肯定更多，表 4 给出了采用 Elmore 延迟估计和采用 D2M 延迟估计所得到的优化面积的比较。

表 4 采用 Elmore 延迟估计和采用 D2M 延迟估计所得到的优化面积比较

电路节点数	延迟上限 (ns)	D2M 优化面积 (μm^2)	Elmore 优化面积 (μm^2)	面积增加
400	200	3772.8094	无解	#VALUE!
800	1000	5488.7134	6949.7309	26.619%
1200	2000	8907.2482	11405.678	28.049%
1600	5000	8588.2054	12130.4984	41.246%
2000	8000	10627.0463	14180.4778	33.438%
2400	10000	14296.1825	21590.7142	51.024%
2800	12000	18509.9299	25236.0058	36.338%
3200	15200	21656.3566	27336.2234	26.227%

从表中可以看出，对于有些情况(例如表中节点数为 400 的情况)，采用 D2M 延迟估计可以得到最优化解，而采用 Elmore 延迟估计却无解。这是因为 Elmore 延迟估

计的误差造成的, 过大的估计了节点的误差导致本来满足时序约束的节点被判断成不满足。

对于表中的其他情况可以看出, 采用 Elmore 作为延迟估计所得到的布线面积会远大于采用 D2M 作为延时估计所得到的布线面积。这充分说明了采用 D2M 作为延时估计能够更大程度的利用设计余量, 得出最经济的设计结果。

4.4 本章小结

本章对文中提出的物理优化算法进行了测试, 并且比较分析了优化结果, 测试结果说明了以下几点:

1. 本文中所提出的优化算法所需要的优化时间基本上和所优化的电路规模呈线性关系。
2. 采用 D2M 作为延迟估计相对于 Elmore 作为延迟估计, 更能够最大程度的利用设计余量, 得到既合格又经济的设计。
3. 在设计时考虑相邻线网上信号跳变对受干扰线网上信号延迟的影响是必不可少的, 否则得出的设计有可能会违反时序约束。
4. 是否考虑相邻线网上信号跳变对受干扰线网上延迟的影响对优化结果的面积大小影响并不大。

第5章 总结

本文提出了一种符号化矩敏感度算法,能够通过遍历一遍符号化表达式树快速计算出电路中节点各阶矩关于电路中所有 RC 元件的敏感度。并采用该算法对相对精确的物理模型进行优化。该物理模型相对传统的线宽优化模型进行了三点改进:

- 采用 D2M 延迟估计作为信号延迟的度量。相对于大部分优化算法所采用的 Elmore 延迟度量, D2M 具有更高的精度,因此优化所得到结果的设计余量较少。
- 在估计相邻线网耦合时考虑了相邻线网上信号的变化对受干扰线网信号延迟的变化。随着 VLSI 工艺往深亚微米方向发展,耦合电容对电路的影响越来越明显。由于耦合电容的存在,一个线网上的信号跳变可能会对其他线网上的信号延迟产生影响,如果不对这种影响加以考虑,可能会导致时序约束的不满足。
- 采用了符号化矩和矩敏感度算法作为求解的核心。本文的优化模型相对于先前所提出的优化模型,有更高的复杂度。正是基于符号化矩敏感度算法计算效率高,可重复性强的特点,才给模型的求解提供了强有力的支持。

实验测试结果表明本优化算法能够更好的利用设计余量,并且得到的结果能够充分考虑串扰的最坏影响。其优化结果要优于原有的优化算法。

鉴于工艺不确定性的增强,如何在设计过程中就对工艺不确定性加以考虑是个很重要的问题。因此本文接下来的工作是要将工艺不确定性因素引入到优化算法中,以提高产品良率。

参考文献

- [1]. J. Cong and K. S. Leung, "Optimal Wire-sizing Under Elmore Delay Model," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst., vol. 14, no. 3, pp. 321-336, Mar. 1995.
- [2]. J. Cong, K. S. Leung and D. Zhou, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model," UCLA Computer Science Tech. Report CSD-9200043, Los Angeles, CA 90024, Sept. 1992.
- [3]. C. Alpert, A. Devgan, and C. Kashyap, "A two moment RC delay metric for performance optimization," International Symposium on Physical Design (ISPD), pp. 69-74, 2000.
- [4]. C. P. Chen and D. F. Wong, "Optimal wire sizing function with fringing capacitance consideration," in Proc. Design Automation Conf., pp. 604-607, June 1997.
- [5]. J. P. Fishburn, "Shaping a VLSI wire to minimize Elmore delay," Proceedings of the European Design and Test Conference, pp. 244-251, 1997.
- [6]. C. P. Chen, Y. W. Chang, and D. F. Wong, "Fast performance-driven optimization for buffered clock trees based on lagrangian relaxation," in Proc. Design Automation Conf., pp.405-408, June 1996.
- [7]. J. Cong, L. He, C. K. Koh, and Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," in Proc. Int. Conf. Computer Aided Design, pp.628-633, Nov. 1997.
- [8]. J. Cong, L. He, C-K. Koh, and D. Z. Pan, "Interconnect sizing and spacing with consideration of coupling capacitance," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst., vol. 20, pp.1164-1169, Sept.2001.
- [9]. J. I. Hui-Ru, C. Yao-Wen, and J. Jing-Yang, "Crosstalk-Driven Interconnect Optimization by Simultaneous Gate and Wire Sizing," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst., vol. 19, no. 9, pp. 999-1010, Sept. 2000.
- [10]. C. P. Chen, C. N. Chu and D. F. Wong. "Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation," IEEE Trans. Computer-Aided Design of

- Integrated Circuits Syst., vol. 18, no. 7, pp. 1014-1025, July 1999.
- [11]. T. Bogdan, D. Florentin and P. Lawrence, "An Explicit RC-Circuit Delay Approximation Based on the First Three Moments of the Impulse Response," 33rd Annual Conference on Design Automation (DAC'96), pp.611-616, 1996.
- [12]. P. Lawrence, A. Emrah and L. Tao, "h-gamma: An RC Delay Metric Based on a Gamma Distribution Approximation of the Homogeneous Response," 1998 International Conference on Computer-Aided Design (ICCAD '98), pp.19-25, 1998.
- [13]. Z. G. Hao, G. Shi, "Sensitivity Approach to Statistical Signal Integrity Analysis of Coupled Interconnect Trees," (ASPDAC'10), 2010
- [14]. A. Liu, "A Symbolic Approach to the Signal Integrity of Multi-coupled RLC Tree Circuits with Application," Master Thesis, School of Microelectronics, Shanghai Jiao Tong University, 2009.
- [15]. J. Alpert, et. al. "Buffer insertion for noise and delay optimization," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 11, pp. 1633-1645, 1998.
- [16]. http://en.wikipedia.org/wiki/Lagrangian_relaxation, 2009.11.1
- [17]. K. Agarwal, D. Sylvester, and D. Blaauw, "Modeling and Analysis of Crosstalk Noise in Coupled RLC Interconnects," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 5, pp. 892-901, May 2006.
- [18]. J. E. Lorival, D. Deschacht, Y. Quere, T. Le Gouguec, F. Huret, "Additivity of Capacitive and Inductive Coupling in Submicronic Interconnects," International Conference on Design and Test of Integrated Systems in Nanoscale Technology, pp. 300-304, 2006.
- [19]. M. Kuhlmann and S. S. Sapatnekar, "Exact and Efficient Crosstalk Estimation," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, no. 7, pp. 858-865, July 2001.
- [20]. Devgan, "Efficient coupled noise estimation for on-chip interconnects," in Proc. IEEE/ACM International Conference on Computer-Aided Design, pp. 147-151, Nov. 1997.
- [21]. G. Rohini, T. Bogdan and P. Lawrence, "The Elmore Delay as a Bound for RC Trees with Generalized Input Signals", IEEE Trans. Computer-Aided Design of Integrated Circuits Syst. Vol. 16, No. 1, pp. 364-369, Jan. 1997.

- [22]. C. J. Alpert, A. Devgan, C. Kashyap, "A Two Moment RC Delay Metric for Performance Optimization", Proceedings of the 2000 International Symposium on Physical Design, pp. 69 –74, 2000.
- [23]. C. V. Kashyap, A. Devgan, "Closed-Form Expressions for Extending Step Delay and Slew Metrics to Ramp Inputs for RC Trees", IEEE Trans. Computer-Aided Design of Integrated Circuits Syst. Vol.23, No.4, pp. 509-516, April 2004.
- [24]. Shebaita, C. Amin, "Expanding the Frequency Range of AWE via Time Shifting", Proceedings of the 2005 IEEE/ACM International conference on Computer-aided Design, pp. 935-938, 2005.
- [25]. L. Tao, A. Emrah, and P. Lawrence, "h-gamma: An RC Delay Metric Based on a Gamma Distribution Approximation of the Homogeneous Response," Proceedings of the 1998 IEEE/ACM international conference on Computer-Aided Design, pp. 19-25, 1998.
- [26]. L. Ding, D. Blaauw, and P. Mazumder, "Accurate Crosstalk Noise Modeling for Early Signal Integrity Analysis," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst. Vol.22, NO.5, pp. 627-634, May 2003.
- [27]. L. R. Curtis and T. P. Lawrence, "RICE: Rapid Interconnect Circuit Evaluation Using AWE," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst., vol. 13, no. 6, pp. 763-776, June 1994.
- [28]. T. Lawrence and A. Ronald, "Asymptotic Waveform Evaluation for Timing Analysis," IEEE Trans. Computer-Aided Design of Integrated Circuits Syst., vol. 9, no. 4, pp. 352-366, April 1990.

附录一 拉格朗日松弛法

拉格朗日松弛法[16]是一种将约束条件转移到目标函数上的方法，当约束条件不满足时，则优化目标会变差。

假设有如下的优化问题：

$$\begin{aligned} \text{Max: } & c^T x \\ \text{s.t. } & \begin{cases} (1) A_1 x \leq b_1 \\ (2) A_2 x \leq b_2 \end{cases} \end{aligned}$$

将约束条件（2）引入目标函数可得：

$$\begin{aligned} \text{Max: } & c^T x + \lambda^T (b_2 - A_2 x) \\ \text{s.t.: } & A_1 x \leq b_1 \end{aligned}$$

其中 λ 为非负权值，如果约束条件（2）没有被满足，则目标函数的值会变小；如果约束条件（2）被满足，则目标函数会变大。假设 x 为原优化问题的最优解， x' 为拉格朗日松弛问题的最优解，我们可以得到下面这个不等式：

$$c^T x \leq c^T x + \lambda^T (b_2 - A_2 x) \leq c^T x' + \lambda^T (b_2 - A_2 x')$$

第一个不等式成立是因为原问题的最优解肯定满足约束条件（2），第二个不等式成立是因为 x' 是拉格朗日松弛问题的最优解。

拉格朗日松弛法的求解过程就是不断搜索 λ 的值，使拉格朗日松弛问题的最优解最小。即：

$$\min : P(\lambda), \text{ s.t.: } \lambda > 0$$

其中 $P(\lambda)$ 的定义为：

$$\begin{aligned} \text{Max: } & c^T x + \lambda^T (b_2 - A_2 x) \\ \text{s.t.: } & A_1 x \leq b_1 \end{aligned}$$

拉格朗日松弛法的求解过程就是不断搜索 λ 值的过程，每一个求得的 $P(\lambda)$ 值都是原问题最优解的上限。求得的 $P(\lambda)$ 越小就越接近最优解。

致谢

本论文是在上海交通大学微电子学院施国勇教授的直接指导下完成的，施老师在研究过程中提供的建议和对研究方向的指引是我的研究工作顺利完成的基础。硕士两年半生活是我学习科研工作的起步阶段，在这过程中施老师深厚的理论功底和对学术研究的热爱给了我很大的启迪。他时常告诉我们，做学术研究一定要做世界一流的，为了拓展我们的视野，他鼓励我们假期到工业界知名企业实习，以帮助我们了解业界最先进的技术。除此施老师作为一名学者的操守和品行时刻影响着我們，他会认真阅读每个学生的学术论文，并且帮助我们反复修改。我硕士论文能够顺利完成，施老师倾注了很多的心血，因此首先要感谢的是施老师。

同时要感谢微电子学院郝志刚博士的热心指导和鼓励。郝志刚博士在模型降阶和信号完整性等领域有很好的造诣，在我硕士阶段的学习研究过程中和硕士论文的完成过程中，他给予了很多的帮助，非常感谢他。

此外还要感谢交大微电子学院 EDA 实验室的研究生谭焜元、李骥、陈硕、王婷、曾媚、谢边村、马迪铭，虽然他们大部分人的研究方向与我不太一样，但是正是有了他们的陪伴才有了充满生气和活力的实验室。特别要感谢的是李骥，他花费了大量的时间来管理和配置实验室的服务器，给实验室的每个人创造了良好的科研环境。EDA 实验室的上届学长于雪红、孟晓璇和黄世杰也给予了我很大的帮助，并且传授了不少他们的研究成果，很感谢他们。

最后，我要特别感谢我的家人。他们给予了我很好的启蒙和关爱，使我一步步成长。我人生旅途中的每一步都充斥着他们的心血，我对他们的感谢是无法用言语表达的。

攻读硕士学位期间已发表或录用的论文

1. 陈安, “符号化矩敏感方法在信号完整性优化中的应用”, 信息技术 2009