一种串扰和时延驱动的总体布线算法*

庄昌文** 范明钰 李春辉 虞厥邦

(电子科技大学光电子技术系 成都 610054)

黄 劲

(Ambow Corp. CA95110 USA)

【摘要】 提出了一种能够同时考虑串扰和时延，作为综合性能驱动的总体布线算法。该算法由基于边串扰度、变关键路径边容量的初始布线和基于Agent技术的拆线重布两部分组成，仿真实验表明，该算法是有效的。

关 键 词 VLSI物理设计； 总体布线； 时延； 串扰； Steiner树

中图分类号 TN710

VLSI物理设计由布局和布线组成。布线主要分总体布线和详细布线两个阶段。在总体布线中，每条线网的各部分被合理地分配到各个布线通道区，由此得到布线问题的明确定义[1]。

在深亚微米技术中，连线时延已大大超过了门时延，因而决定了整个系统的时延，并成为影响系统性能的一个关键因素；此外，随着芯片最小特征尺寸的不断减小，模块和互连线排布得更加紧密，电路的工作频率更高，导致相邻线网的耦合显著增加，从而使串扰成为高性能电路设计中的另一个不可忽略的因素。线网时延主要由源点到漏点的距离和线网的拓扑结构决定，而串扰主要由相邻线网间的耦合电容决定。

目前对时延的研究已相当深入，产生了许多时延驱动的布局、总体布线、详细布线算法[2-6]，但在串扰的研究方面基本都是在详细布线阶段才对串扰进行精确地计算[8-10]，这使得在详细布线阶段避免串扰的灵活性非常有限。于是，人们开始研究在总体布线阶段避免串扰，Xue等提出了一种在总体布线之后对布线结果进行调整以减少串扰的算法[11]；Zhou等提出了一种新的总体布线算法，该算法基于最小串扰Steiner树和拉格朗日松弛法技术，首次在总体布线期间考虑如何避免串扰，并给出了诸多问题复杂性的理论结果，但没有对时延进行研究。

本文通过采用一种广泛使用的互连线时延模型和一种简单的串扰计算模型[12]，尝试了在总体布线过程中同时考虑时延和串扰的方案[2,12]。具体作法是：将串扰和时延变换为通道容量表示，将问题转换为变通道容量的Steiner树问题，用一个Steiner树分别对各线网进行初始布线，若有线网违反串扰或时延约束，则采用拆线重布的方法来修正，拆线重布中采用面向Agent技术。

1 时延和串扰模型

1.1 时延模型

图1为采用分布式的RC时延模型[2,5]，其中u是线网n的源点引脚，v是漏点引脚，从u~v的时延del(u,v)存在上限del(u,v)

$$\mathrm{del}(u,v) \leqslant \mathrm{del}(u,v) = \beta(R_d + r_1 L(u,v))(C_1 W + C_{in}) \tag{1}$$

式中 β值为2.21；r_1是单位长度连线的电阻；C_1是单位长度连线的电容；$L(u,v)$是从u~v的路径长度；C_{in}是线网所有漏点引脚的负载电容之和。

1999年11月16日收稿

* 国家博士后基金资助项目

** 男 30岁 博士

1.2 串扰模型

本文采用了一种较一般的串扰模型[12]，如图2所示。图2a为两条相邻线网，图2b为其串扰模型，其中v_1，v_2是驱动器输出电压，R_1，R_2是驱动器电阻，C是线网间的耦合电容，C_1，C_2是除C之外的其他电容。在所有影响串扰的参数中，假设在布图阶段只有耦合电容可以控制，相离超过一个道的线网间的耦合电容忽略不计，只有在相邻道上平行的线网间才存在耦合电容，其电容C大小由下式决定

$$C=\alpha\frac{l}{d^{\lambda}} \tag{2}$$

式中 l是耦合连线长度；d是相邻线网间的距离；λ是一个常数，约为2；α是一个常数。

线网i受到的串扰为

$$\varphi_i=\sum_{j\neq i}s_{ij}C_{ij} \tag{3}$$

式中 s_{ij}是线网i和线网j之间的串扰敏感度，s_{ij}为0或者为1。根据以上假设，串扰计算可简化为

$$\varphi_i=\sum_{j\neq i}s_{ij}l_{ij} \tag{4}$$

式中 l_{ij}是线网i和线网j相邻平行布线的长度。

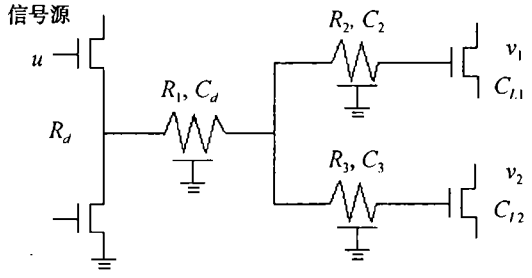

图1 时延参考模型

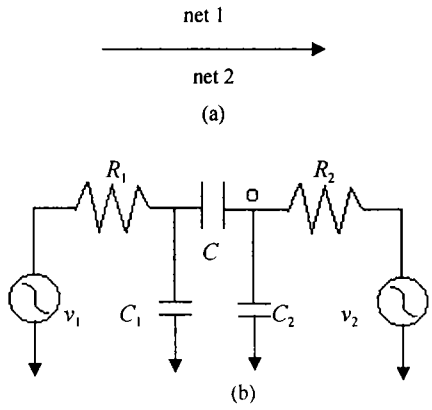

图2 两条线网的串扰模型

2 问题的描述

总体布线通常按一定的规则将给定的版图划分为许多小单元，通过划分过程形成了许多通道(布线区域)，通道与通道相交点的集合构成了总体规划图G的节点集V，连接两个节点的通道集合构成了G的边集E，将诸线网的各引脚映射到总体规划图中便形成了总体布线图$G=(V, E)$。假设所有线网的集合为N，任何线网$i\in N$的引脚集合$\Im_i\subseteq V$。任意边$e\in E$，具有一定的布线容量C_e和长度l_e传统的总体布线就是在G中将各线网的引脚互连起来，同时应满足通过任何边e的线网数目g_e不超过C_e，并尽量使所有线网的连线总长达到最小。

在时延和串扰驱动的总体布线中，除了应满足上述目标以外，还应满足时延约束和串扰约束。时延分析有基于线网和基于关键路径两种技术，因前者过于严格且不准确[1]，本文采用了后者。假设关键路径的集合为P，对任意$p\in P$有

$$p=(v_1, v_2, , v_k)\qquad v_i\in V\qquad 1\leqslant i\leqslant k$$

每条关键路径p_i有一个时延约束$\tau(p)$，即该路径上所有线网时延之和应不超过该路径的时延约束，有

$$\sum_{(u\to v)\in p}\mathrm{del}(u,v)\leqslant\tau(p) \tag{5}$$

式中 $(u\to v)$是指一条从u~v的路径，u是线网的源点，v线网的漏点。

另外，每个线网都有一个串扰约束，即能容忍的最大串扰，线网i所受到的由式(4)决定的串扰φ_i应小于该线网的串扰约束γ_i，有

$$\varphi_i<\gamma_{i0}\qquad 1\leqslant i\leqslant|N| \tag{6}$$

3　总体布线算法

满足时延或串扰的总体布线问题都是NP-完全问题，但同时满足二者的算法十分复杂，本文采用了一种新的思路来避免陷入诸多的NP-完全问题而导致算法低效。

首先，在总体布线阶段，没有线网在一个通道内的具体布线信息，自然就没有线网间的相邻信息，如线网间距和平行走线长度等，因此不能直接计算线网受到的串扰。在文献[12]中，往某通道上增加新的线网时，选择一种串扰增加最少的方式将线网布在某道上，文献[11]中采用分区和线网在区内排序的方式，在区内寻找一个串扰最小的布线方式，两者实际上都在通道内为各线网分配具体布线位置，这种方法不可避免地引入了一系列新的NP-完全问题，使克服串扰问题变得异常复杂。本文在计算线网的串扰时，不分配线网在一个通道内的具体走线位置，而是按概率来计算线网所受到的串扰，极大地提高算法速度，并引入了一个表征通道串扰情况的量——串扰度。

定义1　串扰度是一个表征通道串扰程度的量，通道e上的串扰度k_e计算如下

$$k_e=\begin{cases}0 & g_e\leqslant\dfrac{C_e}{2}\\[2ex] \dfrac{l_e g_e\sum\limits_{i,j\in e}s_{ij}}{C_e} & g_e>\dfrac{C_e}{2}\end{cases} \tag{7}$$

式(7)表明，通道的串扰度与通道长度、经过通道的线网数目成正比，与通道容量成反比。 其次，根据布局结果信息和总体布线后线网具体的拓扑结构，可以计算出线网中源点到各漏点的时延。为了保证关键路径能够满足时延约束，本文引入了时延关键边的概念。

定义2　在总体布线图中，时延关键路径上的边为时延关键边。

在布线之前，将所有时延关键边e的通道容量C_e按一定的比例缩小，这是为了控制在关键路径上的线网数目，借此减少关键路径上的时延，以满足时延约束。

本文将时延和串扰在一定程度上转化为通道的容量表示，以加快算法的速度。总体布线算法(CDDGR)分为两部分：1) 基于变通道容量的Steiner树的初始布线部分；2) 采用面向Agent的拆线重布部分。

3.1　基于Steiner树的初始布线算法

步骤1　根据布局结果构造总体布线图$G=(V,E)$，初始化关键路径上的时延约束、各个线网的串扰约束和线网间的敏感度矩阵$\boldsymbol{S}(n,n)$, $n=|N|$。

步骤2　将关键路径上的时延约束转化为通道容量表示，如关键路径$p\in P$，路径p上的所有边的布线容量按下式减小

$$C_e=\xi C_e\qquad 0<\xi<1\qquad e\in p \tag{8}$$

式中　ξ为通道限制因子，即限制通道中的走线数目，以达到控制路径上时延的目的。实验中取$\xi=0.98$。

步骤3　求解线网i的Steiner树。本文采用了CFD Steiner树算法[17]，因为该算法考虑了时延、线长，因此时间复杂度低。

步骤4　计算线网i所经过边的串扰度，若某边e成为了新的最大串扰度边且大于0，则将边的容

量C_e减少，如$C_e=C_e-1$。然后记录下这一操作，表明边e的真实可用布线容量更大一些，这样做是为了抑制更多的线网经过该边。

步骤5 若还有线网未解得Steiner树，转步骤3，否则转步骤6。

步骤6 算法结束。

算法在步骤2和步骤4中将相应边的容量减小，因此通过这些边的线网数相应减少，使关键路径上的时延得到控制，同时各通道(边)上的串扰也均匀化。由此得到了在一定程度上同时考虑了时延和串扰的初始布线结果，为总体布线的第二部分快速求解到满意解奠定了基础。

3.2 面向Agent的拆线重布技术

步骤1 按式(5)分别计算各关键路径上的时延并判断是否满足时延约束，记录下违反时延约束的路径及其上时延最大的线网。

步骤2 计算各线网所受到串扰：

1) 在每条边上(通道)计算各线网的串扰值。为了简化，我们按概率来计算线网的串扰值，根据经过边e的线网数目g_e、线网i在边e上与其他线网的串扰敏感度之和以及边e的通道容量C_e，可以计算线网i与一条敏感线网相邻的概率η和与两条敏感线网相邻的概率μ：

$$\eta == \frac{2s_e(g_e - s_e - 1)}{(C_e - 1)(C_e - 2)}$$

$$\mu == \frac{2s_e(s_e - 1)}{(C_e - 1)(C_e - 2)}$$

$$s_e = \sum_{j\neq i,j\in e} s_{ij}$$

2) 线网i在边e上与敏感线网相邻的长度$l_{ei}=(2\mu+\eta)l_e$，由此求得线网i在边e上受到的串扰。

3) 将线网i在所有经过的边上的串扰值求和，便得到了线网i受到的串扰。

4) 重复上述1)~3)，直到所有线网的串扰值计算完毕。

5) 判断各线网是否满足串扰约束，若有线网违反串扰约束，记录这些线网。

步骤3 根据步骤1和步骤2的计算结果，将违反时延约束的关键路径上的时延最大的线网和不满足串扰约束的线网拆掉。若所有时延和串扰约束都得到满足，转步骤6。

步骤4 对拆掉的线网重新布线，线网重布过程中采用了面向Agent技术。每条线网由一个Agent去完成布线，所有Agent同时从各自线网的源点出发，并行布线。每个Agent总是朝各漏点的方向前进，并按照如下的规则来选择下一个结点：假设当前结点为v_i，为了确定下一个结点，Agent将选择串扰敏感度小和非时延关键路径的边，尽量避免串扰敏感度大和违反时延约束的关键路径。当Agent无法同时向靠近漏点的方向前进时，Agent会自动生成新的子Agent，子Agent将独立去找寻部分节点，当线网中所有节点都被遍历后，便完成了该线网的重新布线。

步骤5 当所有线网的重布都完成后，转步骤1。

步骤6 算法结束。

4 实验结果

我们用C语言在IBM 赛扬300GL/UNIX操作系统下实现了时延和串扰驱动的总体布线算法，并用3个BBL布图模式的基准电路来测试该算法的有效性。3个基准电路是Ami33，Ami49和Xerox，其规格描述如表1所示。

表1 基准电路规格

电路	宏单元数	线网数	引脚数
Ami33	33	123	442
Ami49	49	408	953
Xerox	10	203	696

借助于文献[13]中的基于模拟退火的布局算法生成了上述基准电路的布局结果。在每个电路中选取了一些引脚多的线网作为关键线网

(上述基准电路本身没有实际的时延信息)，如在电路Ami33中选取6个线网，并定义了关键路径集。

我们将CDDGR和另外3个总体布线算法的布线线长进行了比较，其结果如表2所示。其中第1个算法是KMB启发式方法[14]，第2个是Mickey介绍的 M-route方法[15]，最后一个是IKMB方法[16]。

表2　CDDGR与其他算法的线长/计算时间(s)比较

电路	KMB	Mikey M-route	IKMB	CDDGR
Ami33	56 770/*	55 865/ 2.3	55 815/ 30.3	55 915/ 8.4
Ami49	371 362/*	361 378/ 20.0	360 927/ 20.8	360 869/ 21.5
Xerox	568 480/*	561 935/ 4.2	561 935/ 6.8	561 935/ 6.3

从总体布线的线长和时间复杂度来看，CDDGR在考虑了串扰和时延的情况下，在一般微机上达到了与最好总体布线算法近似的结果[17]，而其算法却没有同时考虑时延和串扰约束。

我们还测试CDDGR减少串扰的能力。因为没有基准电路包含串扰信息，改变电路Ami33线网间敏感度比率，变化范围为80%~100%，即80%~100%的线网间的串扰敏感度为1；并且改变线网的串扰容忍度，即改变线网的串扰约束值，这可以通过改变该线网与敏感线网并行相邻的长度占线网总长度的比率来实现。分别改变上述两者，统计总体布线算法布线的串扰结果，串扰结果由平均没有满足串扰约束的线网数目来表示。

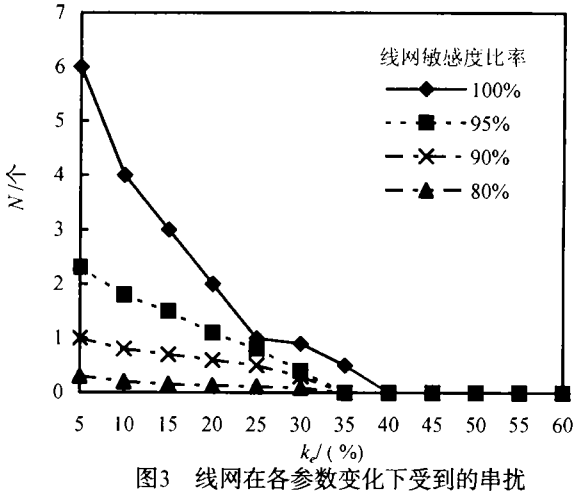

图3　线网在各参数变化下受到的串扰

如图3所示，随着线网的串扰容忍度增加和线网间敏感度的减小，没有满足串扰约束的线网数目变小。这是由于随着线网的串扰容忍度增加，线网就越能与敏感线网并行走更长线；另外，线网间的敏感度越大，线网就越可能受到别的线网引起的串扰。实验结果表明，CDDGR能有效减少线网受到的串扰。

5　结 束 语

本文基于边串扰度的概念和变通道的总体布线图，提出了一种时延和串扰驱动的总体布线算法，实验结果表明该算法是有效的，不仅能达到较优的总连线长度、满足时延和串扰约束，同时具有较低的时间复杂度。

参 考 文 献

1　洪先龙，严晓浪，乔长阁. 超大规模集成电路布图理论和算法. 北京：科学出版社，1998

2　Elmore W C. The transient response of damped linear network with particular regard to wide band amplifiers. J Applied Physics, 1948, 19(1): 55~63

3　Rubinstein J, Penfield P, Horowitz M A. Signal delay in RC tree networks. IEEE Trans on CAD, 1983, 2(3): 202~211

4　Srinivasan A, Chaudhary K, Kuh E S. RITUAL: a performance driven placement algorithm. IEEE Trans Circuits and System II, 1992, 39(11): 825~840

5　Hong X L, Xue T X, Huang J, *et al*. TIGER: an efficient timing-driven global router for gate array and standard cell layout design. IEEE Trans on CAD, 1997, 16(11): 1 323~1 331

6 Lienig J. A parallel genetic algorithm for performance-driven VLSI routing. IEEE Trans on Evolutionary Computation, 1997, 1(1): 29~39

7 Tseng H P, Scheffer L, Sechen C. Timing and crosstalk driven area routing. Proc ACM/IEEE Design Automation Conf, 1998: 378~381

8 Chaudhary K, Oniozawa A, Kuh E S. A spacing algorithm for performance enhancement and crosstalk reduction,. ICCAD, 1993

9 Gao T, Liu C L. Minimum crosstalk channel routing. IEEE Trans on CAD, 1996, 15(5): 465~474

10 Zhou H, Wong D F. An optimal algorithm for river routing with crosstalk constraints. Proc ICCAD, 1996

11 Xue T, Kuh E S, Wang D. Post global routing crosstalk synthesis. IEEE Trans on Computer-Aided Design, 1997, 16(12): 1 418~1 430

12 Hai Zhou, Wong D F. Global routing with crosstalk constraints. Proc ACM/IEEE Design Automation Conf, 1998: 374~377

13 Wong D F, Liu C L. A new algorithm for floorplan design. Proc Dac, 1986

14 Kou K, Markowsky G, Berman L. A fast algorithm for steiner trees. Proc Acta Informatica, 1981, 15: 141~145

15 Chen D, Sechen C. Mickey: a macro cell global router. Proceedings of the European Conference on Design Automation, 1991: 248~252

16 Alexander M J, Robins G. New performance-driven FPGA routing algorithm. IEEE Trans on Computer-Aided Design of Integrated Circuits and System, 1996, 15(12): 1 505~1 517

17 Hong X L, Xue T X, Kuh E S, *et al*. Performance-driven steiner tree algorithm for global routing. Proc ACM/IEEE Design Automation Conf, 1993: 177~181

A Crosstalk and Delay Driven Global Routing Algorithm

Zhuang Changwen Fan Mingyu Li Chunhui Yu Juebang

(Department of Opto-electronic Technology, UEST of China Chengdu 610054)

Huang Jin

(Ambow Crop., CA95110 USA)

Abstract In this paper, a crosstalk and delay driven (integrated performance driven) global routing algorithm (CDDGR) is proposed which consists of two parts: initial routing based on the degree of edge-crosstalk and variable critical-path-edge, and a rip-up and rerouting based on Agent technique. The experimental results indicate that CDDGR is effective.

Key words VLSI physical design; global routing; delay; crosstalk; Steiner tree