



清华大学

UCLA

TSINGHUA UNIVERSITY

系统级封装(SiP)中衬底拓扑布线算法的研究

(清华大学直博学位论文答辩报告)

清华大学计算机科学与技术系

计算机软件与理论

刘盛华

导师：洪先龙 教授

访问导师：何磊 教授 (UCLA)

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

芯片的封装 (package)

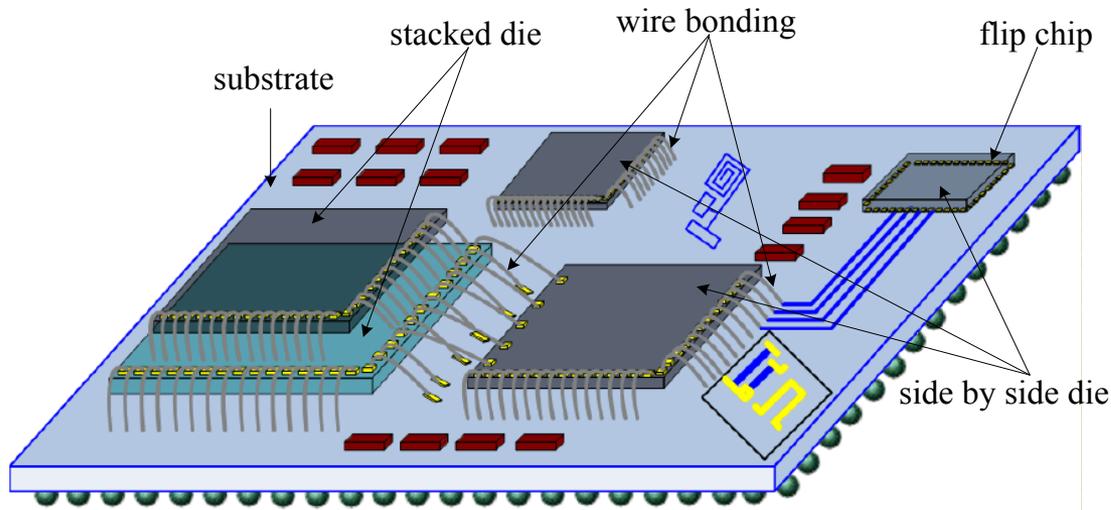


- **芯片的封装**则是给设计好的芯片加上一个衬底(substrate), 并将其I/O连接到衬底的引脚上
- 为了解决I/O数目迅速增多, 密度增大, 减少电感和串扰影响, 近期出现过如下类型的封装衬底(package substrate)
 - **引脚网格阵列封装**(PGA, pin grid array)
 - **球栅阵列封装** (BGA, ball grid array)

封装 (package)技术

- 目前广泛应用的BGA相比PGA有如下优点：
 - 能封装芯片的I/O密度大
 - 使用的焊接球价格低，热阻小，电感小
 - 出产率高
- 针对PGA/BGA衬底的封装，有两种主要技术运用在芯片与衬底的连接：
 - 线焊 (wire bonding, WB)
 - 倒装芯片焊 (flip chip, FC)
- 倒装芯片焊能完成密度更高的I/O的连接，缩短连接线长度，但是比线焊技术成本高

系统级封装SiP(System in Package)

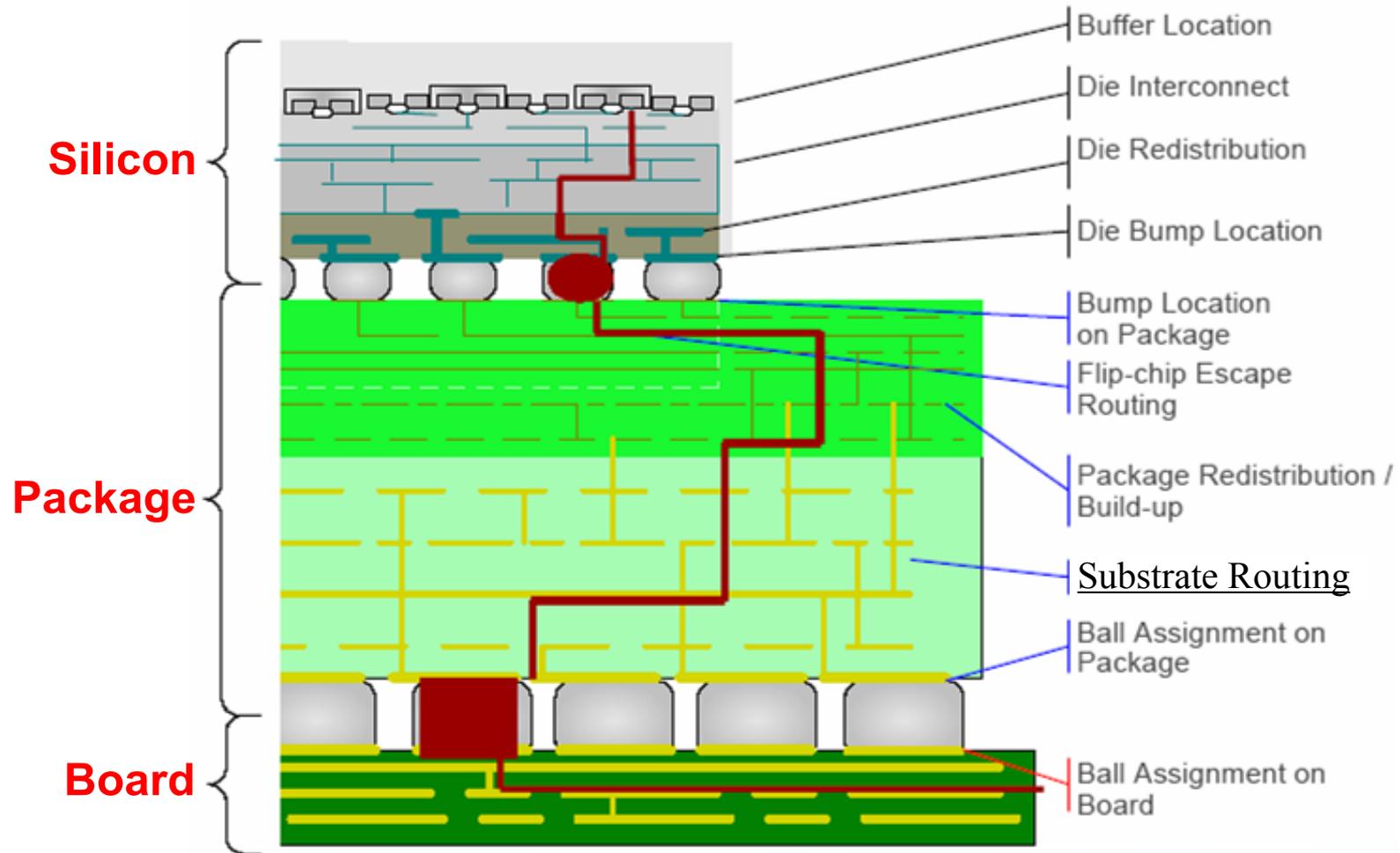


- **系统级封装(SiP)作为一个新技术被提出来**
 - ⊙ 为满足对封装后芯片**面积**和**功能**的日益增长的要求
 - ⊙ 封装自动设计问题更加复杂

封装中的布线 (package routing)

- 在BGA衬底封装中的**线焊** (WB)工艺下，
 - ⊙ 芯片被放置于衬底中的空穴中
 - ⊙ 四周排列着和芯片的I/O 单元相连的**焊块**(bonding pad)
 - ⊙ **衬底布线**会连接**焊块**到底部的**球形触点(balls)**上
- 在BGA封装中的**倒装芯片焊**(FC)工艺下，
 - ⊙ **RDL** (re-distribution layer)**布线**将芯片的I/O单元连到UBM (under bump metal)层的**焊盘阵列**(bump array)上
 - ⊙ **逃逸布线**(escape routing)将焊盘引出到衬底的各层上
 - ⊙ **引出点**(break points)排列在矩形的**逃逸边界框**(escape boundary)上
 - ⊙ **衬底布线**会连接引出点到底部**球形触点**(balls)上
- 本文研究工作的重点在于封装中**衬底布线**算法的研究，尤其是复杂的系统级封装(SiP)

封装布线示意图(源自Cadence)



衬底拓扑布线问题

- 根据封装布线的实际流程，我们假设在衬底布线之前，
 - ⊙ 逃逸布线(escape routing)或线焊(wire bonding)最终将芯片 I/Os 连接到衬底的相应布线层上
- 衬底布线在每一层的线网是希望尽量在同层完成（**平面布线**）
 - ⊙ 信号线网的通孔在工艺上往往被限制在球形触点附近的区域
 - ⊙ 衬底布线要避免垂直方向的绕线

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

衬底布线(substrate routing)现状 (1)

● 工业界

- ⊙ 目前衬底布线工具尚未成熟，布通率较低
- ⊙ 剩下相当数量的线网需要用手工完成布线
- ⊙ 尤其封装设计的灵活多样性和系统级封装(SiP)的出现对衬底布线更是个挑战

● 学术界

- ⊙ RDL布线近期研究
 - ⊙ [J. W. Fang, *et al*, ICCAD' 05]
 - ⊙ [J. W. Fang, *et al*, DAC' 07]
- ⊙ 逃逸布线(escape routing) 近期的研究
 - ⊙ [E. Winkler, EMTS' 96]
 - ⊙ [C. K. Cheng, *et al*, ICCAD' 00]
 - ⊙ [M. Horiuchi, *et al*, TCAD' 00]
 - ⊙ [D. F. Wong, *et al*, ICCAD' 02]
 - ⊙ [C. K. Cheng, *et al*, DAC' 06]

衬底布线(substrate routing)现状 (2)

- 学术界(续)
 - 衬底布线(substrate routing)的近期研究
 - [W.W. Dai, *et al*, ASPDAC' 95]
 - [W.W. Dai, *et al*, ICCAD' 96]
 - [C. C. Tsai, *et al*, TCAD' 98]
 - [S.S. Chen, *et al*, ASPDAC' 99]
 - [Y. Kubo, *et al*, ISPD' 05]
- 与对逃逸布线研究相比较，衬底布线的研究相对比较缺乏和落后，因此成为具有挑战性并且为工业界所亟需解决的问题

博士论文工作的研究内容

- 首先研究了芯片上快速可布性分析的算法
- 提出衬底布线问题的模型，设计实现衬底拓扑布线 (topological routing) 算法
- 在衬底布线问题建模方面
 - ⊙ 提出 *end-zone* 的概念，用来解决交错通孔 (*staggered via*) 的分配问题.
- 针衬底拓扑布线问题
 - ⊙ 利用均匀插入粒子的 *Delaunay* 三角剖分，将衬底平面离散化
 - ⊙ 以 A^* 搜索算法为蓝本，提出 ADS^* 寻路算法，不仅可以绕障碍而且可以动态推动 (*dynamic pushing*) 已布线网，并很好支持 *end-zone*.
 - ⊙ 与 *Negotiation-based* 布线算法的结合，通过添加历史信息 and 拥挤罚值到 DS^* 的费用函数，来提高消除拥挤和布通率的能力.
 - ⊙ 提出将衬底拓扑布线问题模型化为物理学上扩散 (*diffusion*) 的过程，这在拥挤驱动布线算法中是一次新的尝试.

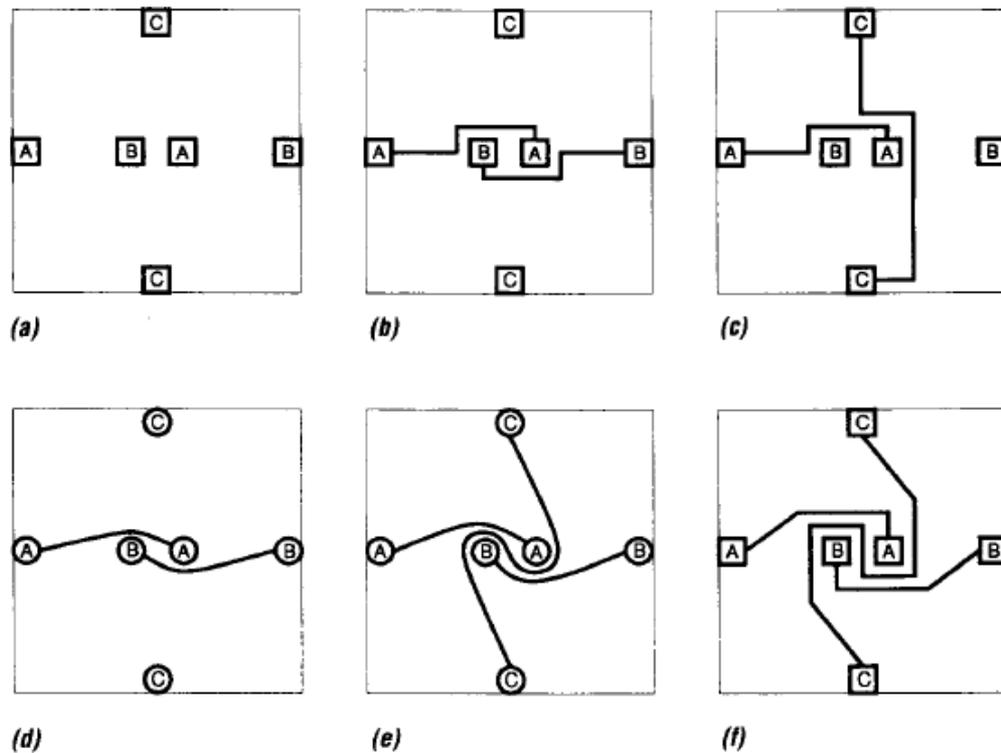
内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

典型平面布线算法

- 平面布线是衬底布线的一个重要约束
- Dana Richards证明了单层平面布线问题属于NP完全问题[D. Richards, IEEE trans. On Computers,1984]
- 单层总体布线算法[M. Sarrafzadeh, *et al*, TCAD, 1994]
 - 单层曼哈顿走线的总体布线问题
 - 利用拥挤信息(density)作为花费函数局部搜索
 - 该算法将布线平面划分成规则的方格，局限于规则障碍下曼哈顿的走线
 - 对于高引脚密度的布线，运行效率不高
- SURF平面拓扑布线算法[Wayne Dai *et al*, D&T of Computers, 1993]
 - 该算法能够用挤(squeeze)的方法解决这个布线难题，提高布通率

SURF平面拓扑布线算法



- 一个简单的布线难题(a); 无论布线顺序如何, 最终都至少会有一条线网无法布通(b)和(c); 但是rubber-band布线连接A和B线网后(d); 线网C能够挤(squeeze)过去(e); 最后, 可以将结果扩展为(f)所示
- 然而, 只有在布线顺序为A-B-C或B-A-C时才可布通, 仍然受布线顺序的影响

典型衬底布线算法 (1)

- [C. C. Tsai, et al, TCAD' 98]
 - ⊙ 针对PGA衬底和线焊接(WB)工艺下的多层衬底布线
 - ⊙ 将衬底规则得划分为四个区域，针对每个区域分别布线
 - ⊙ 实现了快速，均匀的布线
- [S.S. Chen, et al, ASPDAC' 99]
 - ⊙ 可以处理上面算法中跨划分区的连接
- [Y. Kubo, et al, ISPD' 05]
 - ⊙ 在只有两层布线层的BGA衬底和线焊接(WB)工艺下的布线
 - ⊙ 该问题中通孔的数目和位置有限
 - ⊙ 通过对通孔的分配来完成衬底布线

典型衬底布线算法 (2)

- 上述三个算法，对封装衬底设计的要求比较**固定**，**不灵活**
 - 即衬底布线线网的**起点**同**球栅的行/列**平行
- 这对于封装设计的**多样化**和新提出的**系统级封装(SiP)**都是无法满足要求的
- [Y. Kubo, et al, ISPD' 05]
 - 对交错通孔的工艺要求，在算法上给予了支持和考虑
 - 但这种模型局限于只有**两层**的衬底布线，不能很好解决如今**多层**的封装衬底布线问题
- [W.W. Dai, et al, ICCAD' 96]
 - 利用最小花费-最大流的算法求解衬底布线问题
 - 解决的问题中每个线网的终点(引脚pin)都是**可交换的**(interchangeable)
 - 然而，实际中有很多设计者对线网终点的**连接**有很多**电性能方面与设计方面的约束**，甚至会指定线网的**连接关系**(网表)

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

OpenAccess数据平台

- 目前IC业界使用的数据平台的不一致，对规模日益增长的集成电路设计带来了数据交换的极大不便
 - Synopsys Milkyway Database
 - Avanti开发的MDX C-API
 - Synopsys 对外公开的MAP-in Program
 - Mentor Graphics的Falcon
 - Magma的私有数据平台 (database)
- OpenAccess(OA)作为一个开源的EDA数据平台为工业界和学术界广范使用
 - 降低了IC业界间的数据交换的花销
 - 丰富的API能够支持数字，模拟，数模混合的设计数据
 - 支持65nm以下的设计规则的表达
 - 专业数据的压缩技术能够减少内存消耗

可布性分析算法的研究和实现

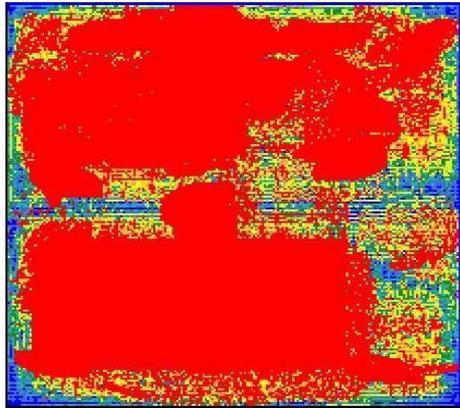
- 为了跟业界结合，我们的工作则选取了OA数据平台
- 初期在OA平台上，针对可布性分析算法的进行了研究和实现[S. Liu, *et al*, ICCAS' 06] [S. Liu, *et al*, CAD&CG' 08]
 - 利用改进的概率方法对拥挤度预估
 - 在概率指导下进行快速布线

测试用例	单元数	线网数	网格	EOPA	可布性评估器		SSTT	
				runtime(s)	runtime(s)	σ	runtime(s)	σ
freecpu	3111	3040	74 × 68	0.08	0.11	1.42	7.93	1.29
u05614	32498	36452	205 × 205	0.63	1.09	2.26	182.69	1.99
u08421	48810	54743	251 × 251	0.99	1.68	2.35	426.33	2.08
u11228	65058	72968	290 × 290	1.06	2.13	2.24	488.74	2.07
u14035	81242	91129	324 × 324	1.67	3.08	2.42	813.80	2.13
u28070	162162	181934	458 × 458	3.61	7.21	2.45	1728.25	2.20

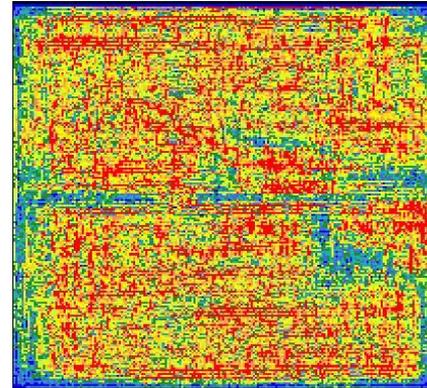
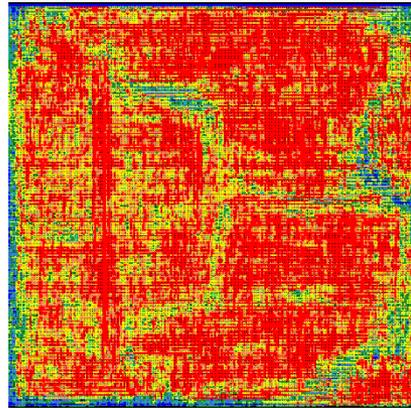
EOPA: 概率估计算法[J. Lou, ISPD'01]

SSTT: 总体布线器[T. Jing, JCST'03]

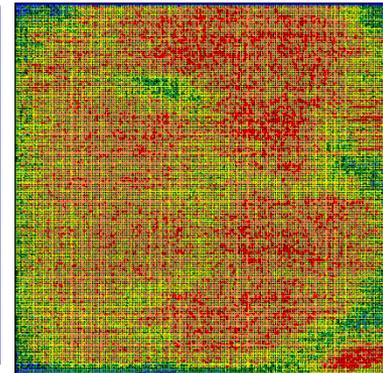
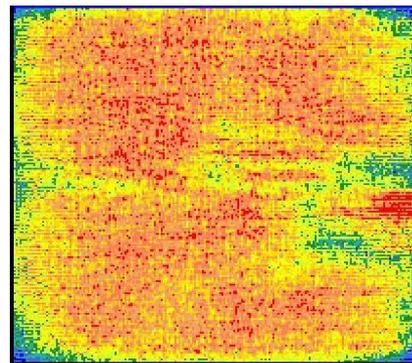
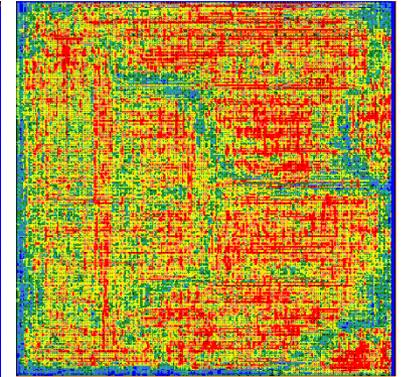
拥挤分布图的比较



EOPA



可布性分析器



SSTT

例子u05614和u06421的拥挤分布图

相关工作的发表刊物

- 该工作发表在“IEEE ICCAS' 06会议集”。
- 并申请专利一项，申请号：ZL 00610012271.4，授权日期：7/23/2008，授权号：CN 100405379C
- 更多关于OpenAccess平台下的实现细节和更丰富的实验分析等发表于“计算机辅助设计和图形学学报，第20卷，第12期，2008年12月”。

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

Delaunay 三角剖分的数据结构表示

- Delaunay三角剖分是一个离散化布线平面的好的选择
 - 在衬底布线中几乎所有线网起点**密布**一排
 - 考虑预先布好的走线和各种**多边形**障碍
- 考虑到三角剖分的**花费**又不失其**性能**，选用 [F. Bossen, M.S. thesis CMU, 1996]介绍的均匀插入粒子的三角剖分(PCDT)

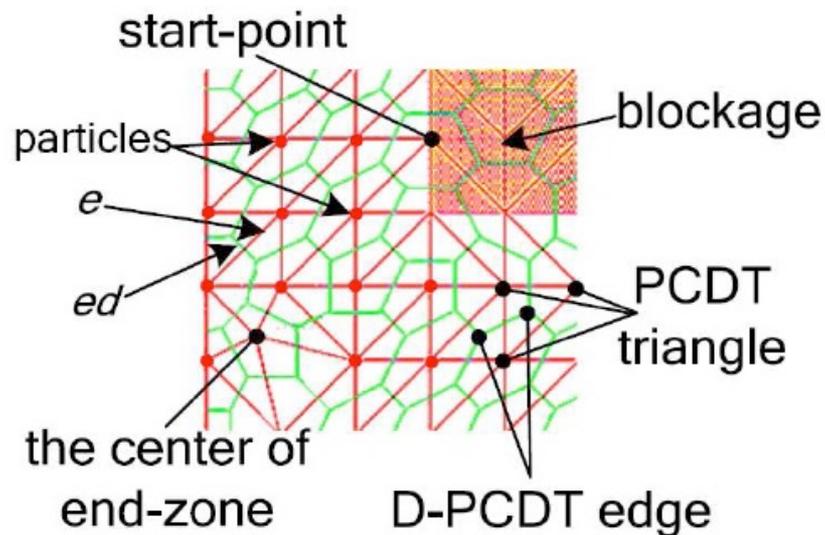
线网在对偶图中走线，因此走线容量为

$$C_{ed} = l_e \quad (\text{三角形边长})$$

拥挤度为

$$\eta_{ed} = \frac{\sum_i (w_i + s_i)}{C_{ed}}$$

其中， w 和 s 为布线线宽和线间距



针对衬底布线平面的PCDT三角剖分的局部放大图

衬底布线中end-zone模型的提出

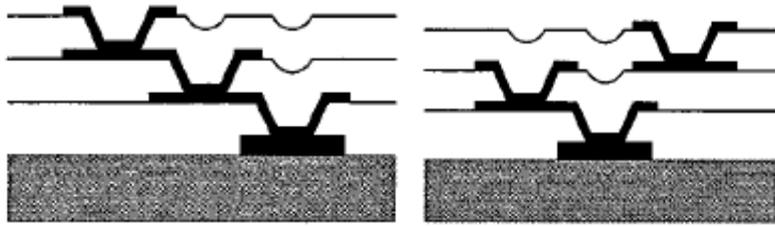


图1. 交错通孔

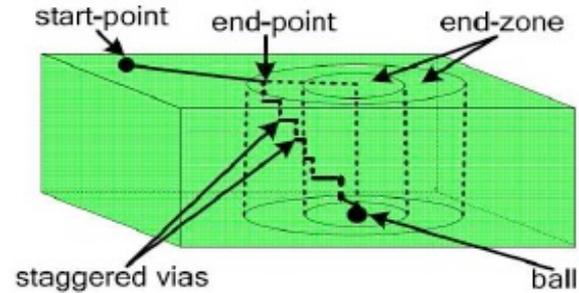


图2. end-zone 模型

- 封装衬底布线的工艺下，往往最后需要通过**交错通孔**(staggered via)将信号线网连接到BGA层的球形触点上
- 制造工艺对交错通孔的相邻孔对的间距有要求，被叫做via pitch
 - ⊙ 最小孔对间距表示为 md_i
 - ⊙ 最大孔对间距表示为 pd_i ，减少交错孔对布线平面电性能完整性的影响
- end-zone模型则定义了两个半径，分别如下
 - ⊙
$$d_1 = \sum_i md_i \quad d_2 = \sum_i pd_i$$
 - ⊙ 衬底布线首先将起点连接到该平面相应的end-zone (以 d_2 为半径的圆)内一点，然后利用交错通孔和底层的球形触点相连 (图2)

衬底拓扑布线问题形式化描述

● 给定

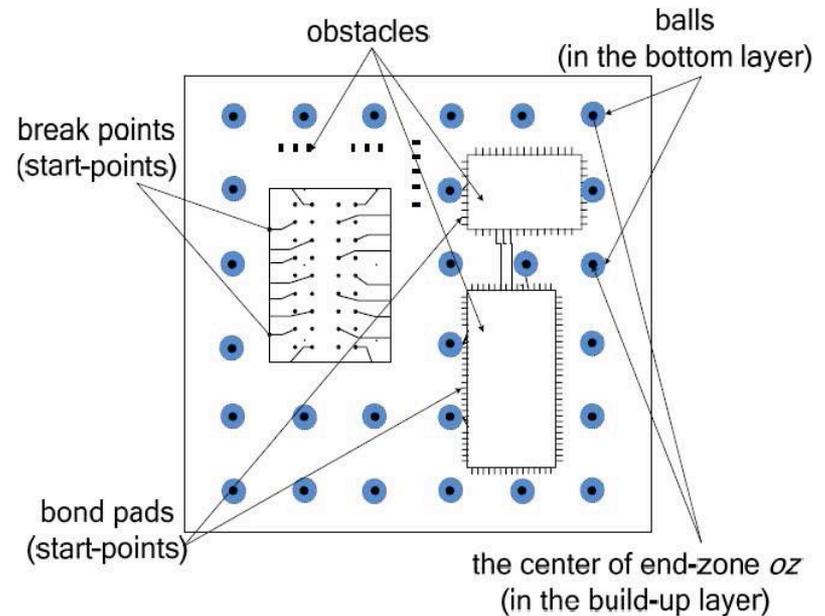
- 起点集合start-points,
- end-zones (与底层的球栅阵列相对应),
- 网表netlist (定义了每个起点同end-zone的对应关系),
- 障碍obstacles (包括逃逸布线区, 线焊工艺中的空穴, 预布线网, 通孔, 等等),

● 求解

- 一个拓扑布线的解决方案

● 满足

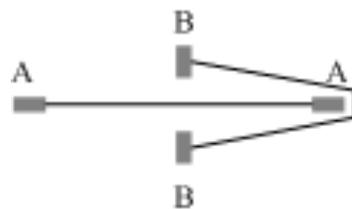
- 所布线网不交叉
- 满足布线拥挤度的要求
- 并且总线长最短



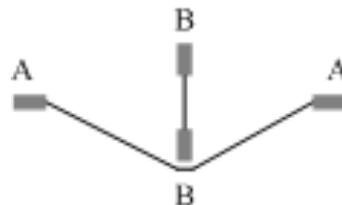
某一层上的衬底布线图 (SRG)

布线顺序问题

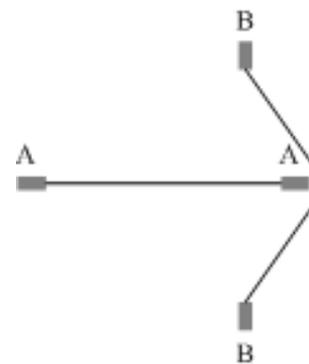
目标：布线总线长最短



距离长的先布



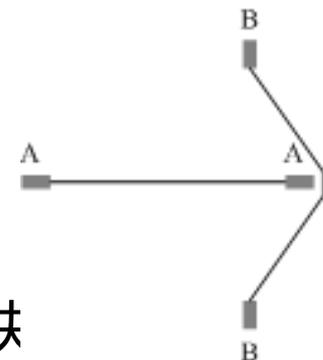
距离短的先布



布线顺序很难决定

动态推动(dynamic pushing)

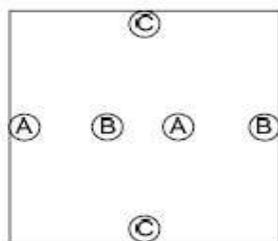
- 从图示布线的结果观察，形成该结果的方式
 - 线网A先布好后，线网B通过绕线完成布线
 - 线网B先布好后，线网A推动B至A的终点
- 动态推动(dynamic pushing)思想被提出来用来解决平面布线中绕线的顺序依赖
 - 利用动态规划的技术，通过费用函数决定布线过程中遇到已布线网时，采用绕线(detouring)还是推动(push)已布线网
 - 以上图为例：
 - 若线网A先布，则线网B在布线过程中动态决定绕线(detouring)；
 - 若线网B先布，则线网A在布线过程中动态决定推动(push)已布线网。
 - 将布线顺序量化到费用函数中，弱化布线顺序对结果的影响



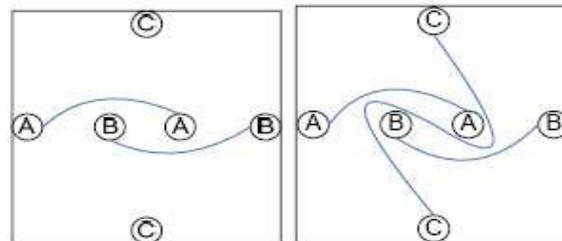
布线难题

- 如图展示了动态推动(dynamic pushing)如何解决该布线的难题

(A, B)-C

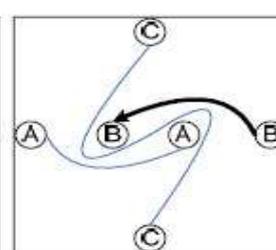
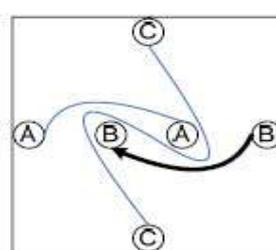
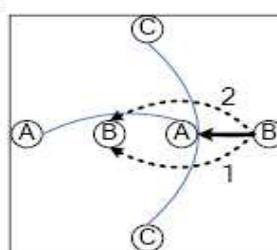
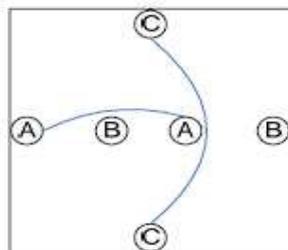


(a)



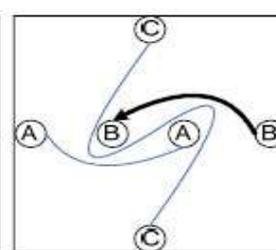
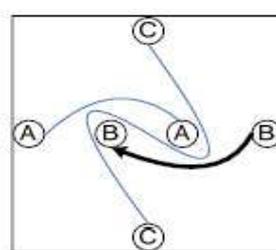
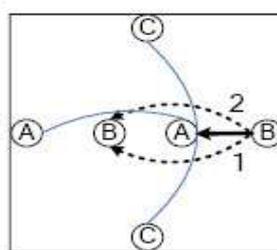
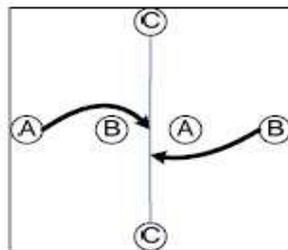
(b)

A-C-B或B-C-A



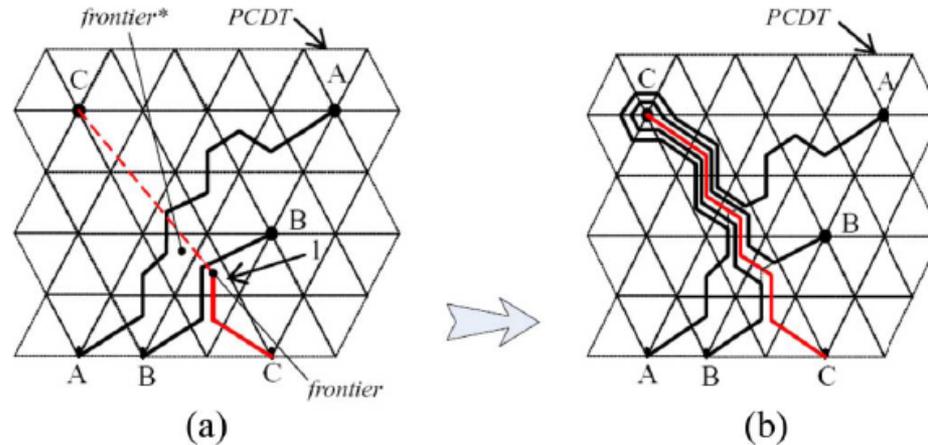
(c)

C-(A, B)



(d)

动态推动的实现



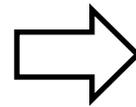
- 以A*搜索算法为框架，在三角剖分的离散平面上搜索 (ADS^*)

- 费用函数为两部分之和：实际费用 rc + 预估费用 ec

$$w = w_0 + s_0$$

$$w^* = w + 2 \sum_i (w_i + s_i)$$

Initial: $rc = p_0 \times w$



$$rc^* = rc + \Delta p \times w^*$$

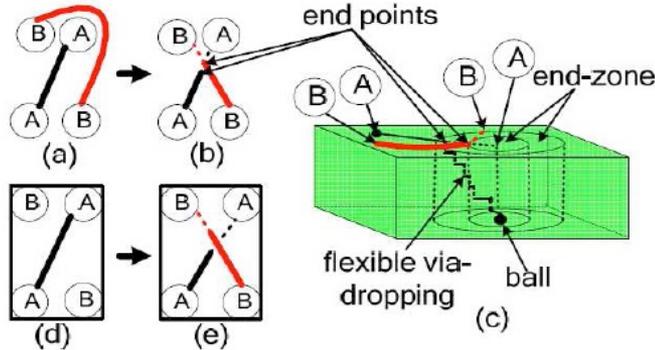
$$ec = h_0 \times w$$

$$ec^* = h \times w^*$$

- 其中 w_0 和 s_0 为布线线宽和间距， p_0 为起点 C 到邻接三角形重心的距离， h_0 为估计当前距离终点的距离。
- 当线网 C 推动已布线网时， w^* 会增加，费用函数增大；当线网 C 绕线时，总行走距离增加，费用函数亦增大

End-zone模型下的布线

减少总线长



提高布通率

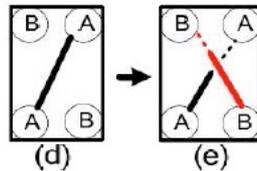


图1. end-zone模型的布线和传统布线对比

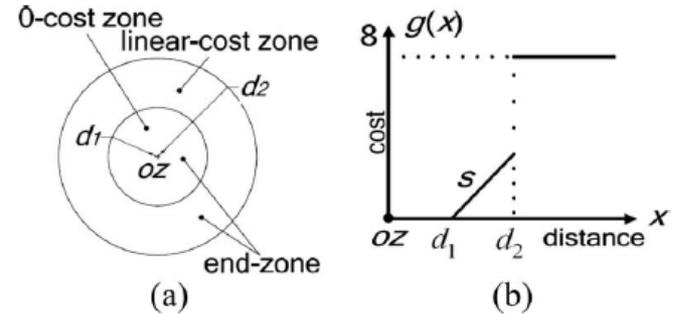


图2. end-zone模型的费用函数

- end-zone模型对于布线，起到了减少总线长和提高布通率的积极作用
- 虽然以 d_2 为半径的圆内都称之为end-zone，但是考虑到交错通孔间距对平面的电性能完整性的影响
 - ⊙ 以 d_1 为分界定义，布线停在end-zone各处的费用函数如图2(b)， s 为斜率

- ADS^*

$$ec = \begin{cases} (h_0 - d_2) \times w + (d_2 - d_1) \times \min(s, w), & h_0 > d_2 \\ (h_0 - d_1) \times \min(s, w), & d_1 \leq h_0 \leq d_2 \end{cases}$$

$$ec^* = \begin{cases} (h - d_2) \times w^* + (d_2 - d_1) \times \min(s, w^*), & h > d_2 \\ (h - d_1) \times \min(s, w^*), & d_1 \leq h \leq d_2 \end{cases}$$

布线的迭代

- 对于拆线重布，运用了[R. Nair, TCAD, 1987]提出的迭代方案
 - 一次拆掉一个线网，并且重布
 - 迭代过程中，每一个线网都要被拆线重布一次

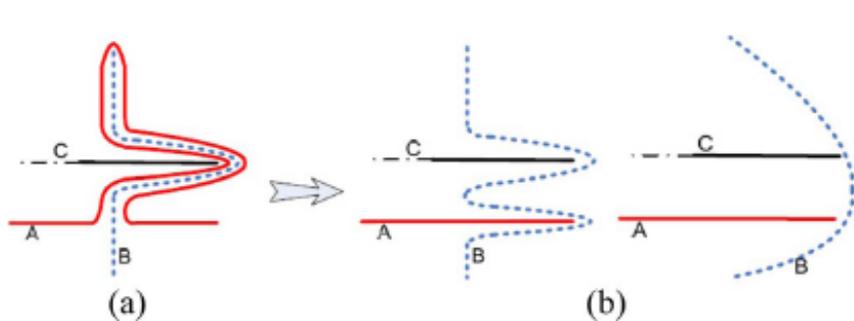


图1

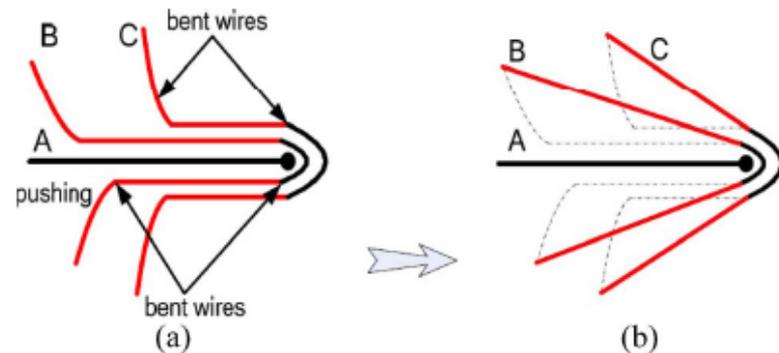


图2

- 在本工作中，拆线重布
 - 用来消除动态推动(dynamic pushing)所产生的弯曲线网
 - 定义线网长距比 $\delta = l/u$ ，即线网长度与起点和终点距离的比
 - 重布顺序
 - 长距比高的线网优先，这样的线网需要优先重布“拉直”（如图1）
 - 后被推动(pushing)的线网优先，因为外层弯曲的线网有“拉直”的空间（如图2）

实验结果 (1)

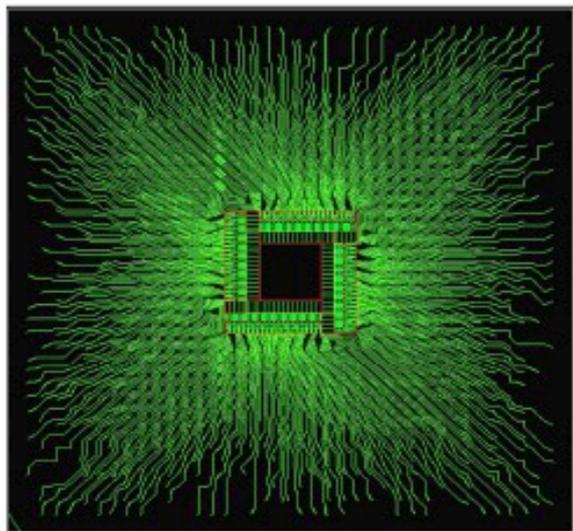
实验例子和结果

Name of circuit	Package type	Package size*	Die(s) size*	Number of nets	Number of failed nets		Average wire length (mm)		Runtime (s)	
					BKM	new	BKM	new	BKM	new
case 1	2-2-2	35000 × 35000	14000 × 15000	474	52	16	9.10	8.49	2.06	11.33
case 2	2-2-2	30000 × 30000	9000 × 10500	543	10	0	8.24	7.99	1.90	5.34
case 3	3-1-3	40000 × 40000	9300 × 9300	800	306	49	1.65	1.37	13.81	16.86
case 4	3-2-3	35000 × 35000	12000 × 12000	506	82	6	22.90	17.80	6.90	8.46
case 5	3-2-3	40000 × 40000	20000 × 22000	891	98	45	5.93	5.27	2.60	13.65
case 6	4-2-4	40000 × 40000	20000 × 23000	990	198	51	9.10	7.93	9.95	26.62
case 7	4-2-4	45000 × 45000	20000 × 19000	1009	100	14	23.10	18.90	5.91	26.14
case 8	1-0-1	12000 × 12000	3900 × 6700 4400 × 5700 3200 × 4400	349	60	22	1.83	1.73	15.24	1.51
case 9	2-2-2	37500 × 37500	11000 × 10000 4700 × 3800 4600 × 5500	538	30	9	9.78	9.38	95.17	110.93
total	—	—	—	6100	936	212	—	—	—	—
average	—	—	—	—	—	—	10.20	8.76 (-13.9%)	—	—

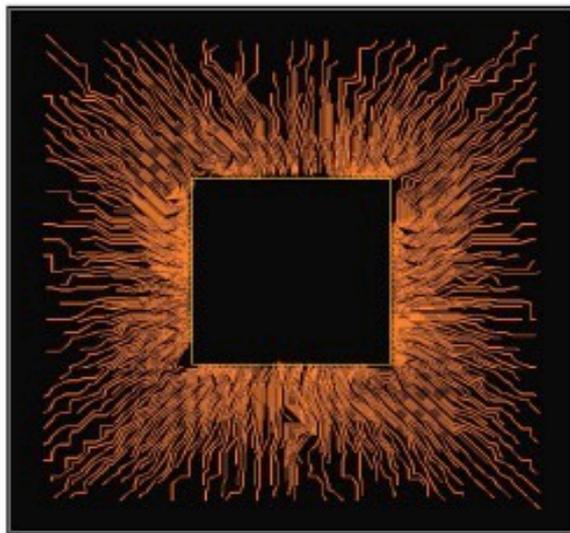
(*: Package size and Die(s) size are given by width × length (μm) in rectangle.)

- 实验中，通过和工业界中已有的衬底布线算法(BKM)相比，
 - ⊙ 我们的算法减少了4.5倍的无法布完的线网数目
 - ⊙ 平均已布线网线长也缩短了13.9%

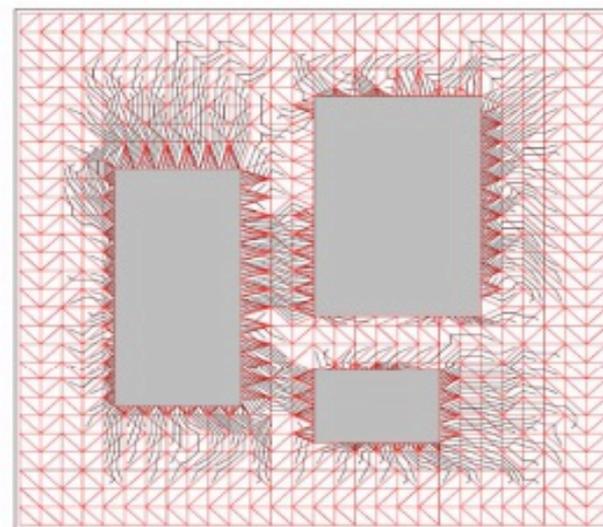
实验结果 (2)



(a) case 3



(b) case 4



(c) case 8

某一层衬底布线层的布线情况

相关工作的发表刊物

- 该科研工作发表于“DAC' 08论文集”
- 更详细的介绍，相关证明，后处理过程，及实验分析发表于“IEEE Transaction on CAD, Vol.28, No.2, Feb. 2009”

Negotiation-based 衬底拓扑布线

- 在前面的研究中，对于拥挤的消除采用了传统的拥挤度控制的方法
 - 随着拆线重布的迭代拥挤度的阈值逐渐逼近于最终的拥挤条件
 - 该方式导致在拥挤度要求高的情况下，布通率下降
- 利用 *negotiation-based* 算法 [McMurchie and Ebeling 1995] 的思想对目前算法的费用函数和拆线重布进行改进
 - 该思想曾应用在 ASIC 的芯片布线中，并取得较好的结果 [Roy and Markov 2007] [Cho et al. 2007]
- 在布线的路径搜索算法 DS^* 中，穿过边 e 时的花费函数为
 - $$NC_e = rc + ec$$
- 考虑到布线时当前的拥挤度和以前布线迭代时所产生的拥挤度，对路径搜索的影响，结合 Negotiation-based 思想，新费用函数
 - $$NC'_e = (rc + h_e) \times p_e + ec$$
 - 其中 p_e 为当前穿过边 e 的拥挤度， h_e 则反映了历史布线迭代所产生的拥挤的影响（如下式， h_{inc} 是常量）

$$h_e^{k+1} = \begin{cases} h_e^k + h_{inc}, & \text{if } e \text{ has overflow} \\ h_e^k, & \text{otherwise} \end{cases}$$

实验比较

Test Case ID	Package Type	Package Size*	Die(s) Size*	Number of Nets	Number of failed nets	
					[Liu's]	Nego
Q1	2-0-2	10000×10000	75007×7700	315	51	41
B2	2-2-2	35000×35000	14000×15000	474	31	30
F3	2-2-2	30000×30000	9000×10500	543	24	22
P4	3-1-3	40000×40000	9300×9300	800	135	135
A5	3-2-3	35000×35000	12000×12000	506	64	63
A6	3-2-3	40000×40000	20000×22000	891	60	57
X7	4-2-4	40000×40000	20000×23000	990	45	45
A8	4-2-4	45000×45000	20000×19000	1009	16	16
S9	1-0-1	12000×12000	3900×6700 4400×5700 3200×4400	349	22	20
S10	2-2-2	37500×37500	11000×10000 4700×3800 4600×5500	538	32	32
total	—	—	—	6415	480	461

*: package size 和 die size 以长×宽 (μm) 给出

[Liu's]表示本文前面的工作；Nego表示结合negotiation-based思想后的改进算法

- 选取拥挤要求较高的10个测试例子进行比较
 - ⊙ 未布成功的线网从480个减少到了461个
 - ⊙ 改进效果不明显

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

拥挤消除的相关工作

- 早期的平面布线Surf系统[Staepelaere et al. 1993]
 - [Dai et al., DAC 1991]完成拓扑布线，生成橡皮筋草图(rubber-band sketch)
 - 将橡皮筋草图转化为辐条草图(spoke sketch)，然后生成有几何位置的布图
 - Surf假设布线的终点是固定的
 - Surf在划分成网格(bin)的布线平面上，进行总体布线生成拓扑结构，然后在网格中进行局部布线消除拥挤
- 目前，[M. Cho and D. Pan, DAC 2006]提出的芯片的布线器，BoxRouter，在同类布线器中有相当好的布通率
 - 在拆线重布消除拥挤时，拥挤窗(congestion window)内的所有线网同时会被拆除
 - 被拆除的线网，则利用混合整数规划(ILP)的方法进行同时绕线
 - 而，ILP方法是用在Manhattan走线的假设前提下，并且能否容易扩展到non-Manhattan的衬底布线还是未知

D-router的特点

- D-Router也是基于end-zone模型
 - ⊙ 在布线和拥挤消除中都提供了灵活性，提高了布同率
- D-router布线过程
 - ⊙ 以任何拓扑布线结果的输入为初始解
 - ⊙ 本工作以[Liu et al., DAC, 2008] 中不考虑拥挤的拓扑布线结果为输入
 - ⊙ 然后，对每一处高度拥挤的区域，
 - ⊙ 将该区域的线网平移到邻近非拥挤区，以减少布线拥挤
- 我们提出的D-router
 - ⊙ 利用扩散的方法进行迭代消除拥挤
 - ⊙ 避免了仅仅限于局部的网格(bin)中消除拥挤的问题
- 相比较BoxRouter而言，D-Router
 - ⊙ 本质上只对拥挤窗中的部分线网进行拆线重布，而且只拆掉了线网中位于拥挤区的部分
 - ⊙ 我们的迭代是依次找到拥挤窗，而BoxRouter则是通过拥挤窗的膨胀变大的方法
 - ⊙ 解决了non-Manhattan的衬底布线

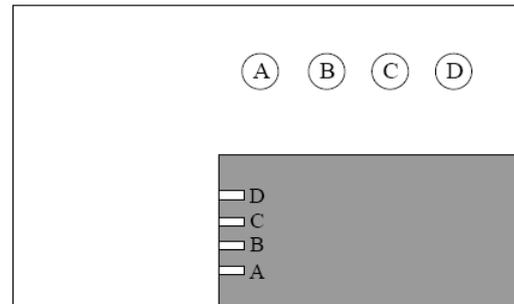
典型案例

- 布线中拥挤消除，往往是通过拆线重布
- 线网重布的顺序，对消除拥挤，提高布通率起了很大作用
- 下面的典型例子说明了为什么D-router在消除拥挤的时候不受重布顺序的影响

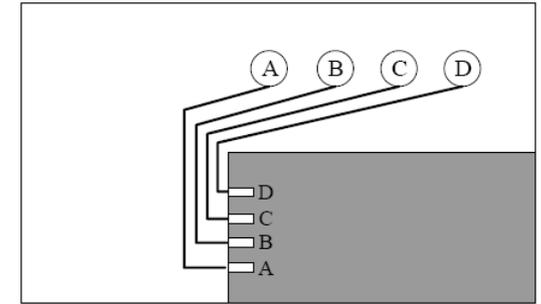
✓ 布线顺序D-C-B-A得到结果(b)

✓ 布线顺序A-(BCD)会先得到结果(c)，但随着拥挤约束下的拆线重布，传统基于A*和Maze算法的布线器会得到的到(d)的结果

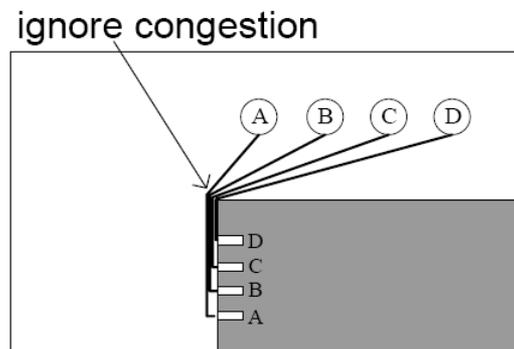
✓ D-router则通过扩散的方式，将拥挤消除，得到(b)的结果



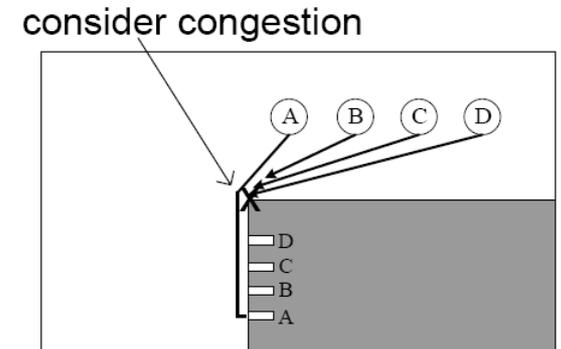
(a)



(b)



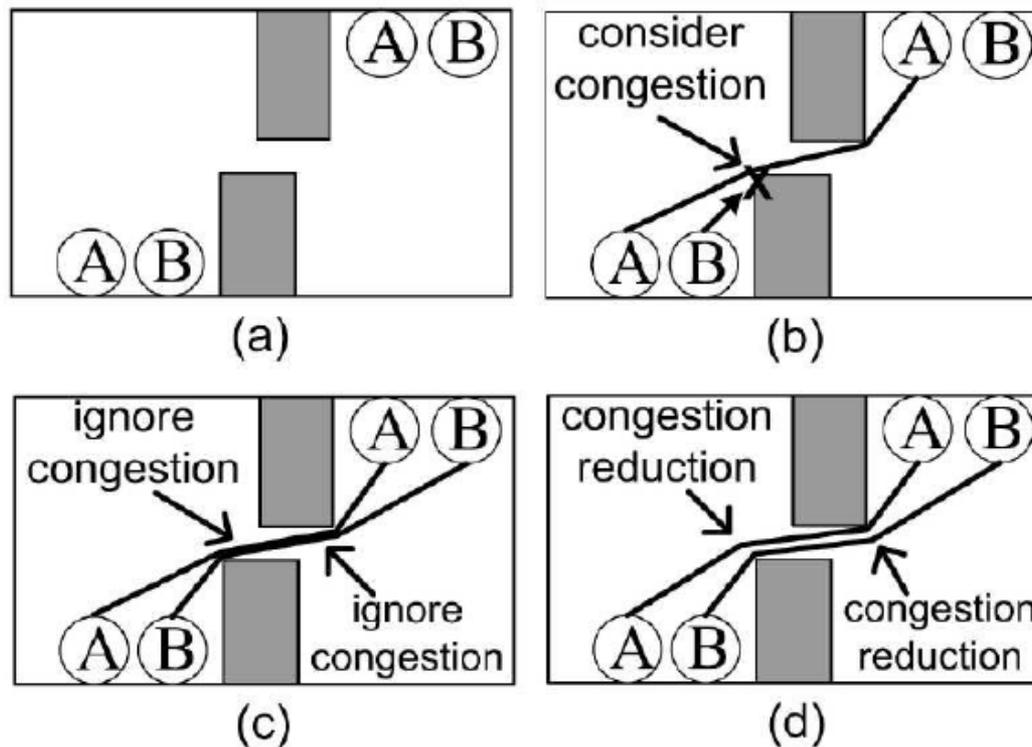
(c)



(d)

布线难题

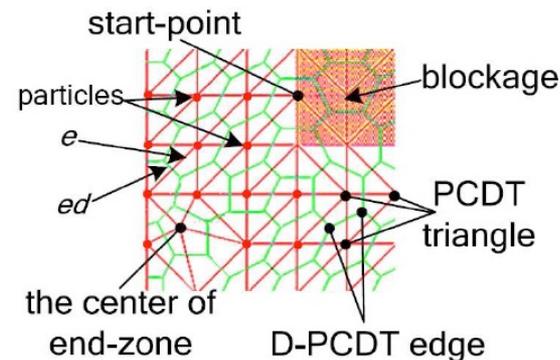
- 除上述典型案例外，在只有两个线网布线时，问题也有可能变得很难
 - 如下图所示的布线情况，我们前阶段的研究工作[Liu et al., DAC, 2008]则无法完全布通



传统的拆线重布无法找到合适的布线顺序

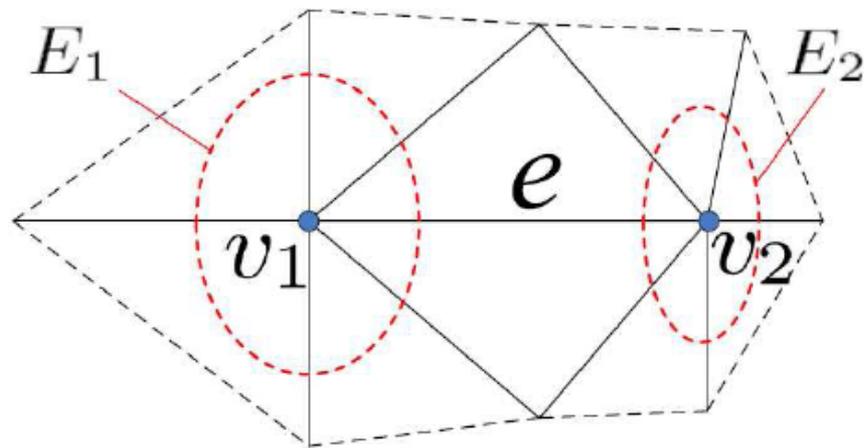
扩散 (diffusion)

- D-router模拟杂质扩散的过程
 - ⊙ 拥挤度高的部位杂质浓度大
 - ⊙ 通过将高浓度区域的线网“扩散 (diffusion)”至周围低浓度区域，减少布线拥挤
- 线网宽度和间距的离散值，决定了线网扩散的过程是不连续的
- 我们仍基于三角剖分后的平面，模拟实现线网的扩散 (diffusion)
 - ⊙ 每条三角形的边作为扩散时移动的原子单位
 - ⊙ 原子扩散，即将一个线网段或终点(end-point)，从所在三角形的边移动到其邻边上的过程
- 因此，D-router的实现是基于局部化的、非解析的扩散模型的



扩散窗 (diffusion window)

- 定义PCDT三角形边 e 在 t 时刻的浓度 $d_e(t)$ 为其对偶边 ed 的拥挤度
 - ⊙ $d_e(t) = \eta_{ed}(t)$
- 扩散窗 (diffusion window) 是完成扩散的一个区域
 - ⊙ 从PCDT平面选取具有高浓度的边 e , 称之为扩散源(diffusion source)
 - ⊙ 扩散窗由扩散源 e 和其左右邻边集 E_1 和 E_2 组成 (如下图示)



扩散窗 (diffusion window)

扩散方向的选择

- 扩散窗中的扩散

- ⊙ 当线网从边 e 移向边集 E_1 或 E_2 时，有可能穿过其中的多条边
- ⊙ 扩散边集 Edf 则被定义为移动某线网时，将要穿过的 E_1 或 E_2 中的边集
- ⊙ 扩散边集 Edf 的浓度为

$$d_{Edf}(t) = \max_{e_i \in Edf} \{\eta_{e_i}(t)\}$$

- 扩散方向

- ⊙ 首先，定义扩散速率

$$v_{e+}(t) = -(d_{Edf+}(t) - d_e(t))/d_e(t)$$

$$v_{e-}(t) = -(d_{Edf-}(t) - d_e(t))/d_e(t)$$

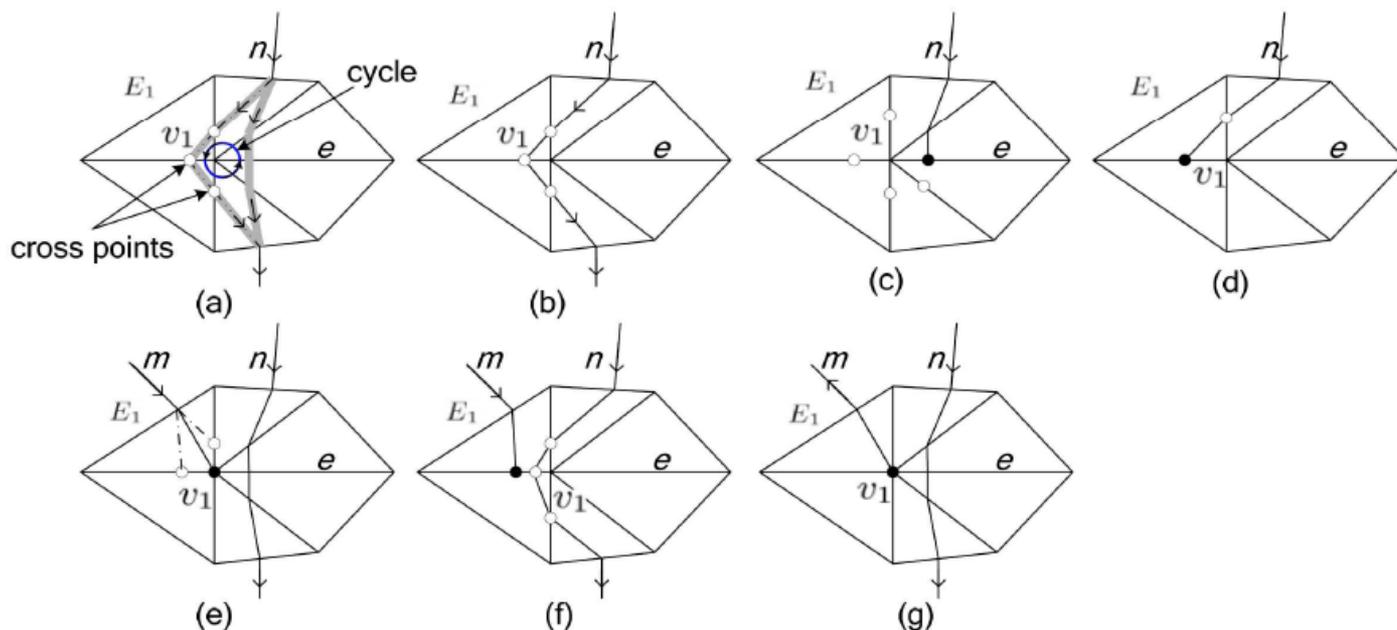
其中 Edf^+ 和 Edf^- 是边集 E_1 和 E_2

- ⊙ 选择扩散源 e 两边的速率比较高的方向，进行扩散
- ⊙ 即，优先选择向较低浓度和降低拥挤度的方向扩散

Momentary-diffusion操作

● Momentary-diffusion操作

- 在扩散窗里，将某个线网段或终点从扩散源到选定方向的一次原子移动



情形 1: 一般情况, 图(a)-(b) ;

情形 2: 线网 n 终点落在扩散源 e 上, 图(c)-(d) ;

情形 3: 线网 m 停在扩散源 e 的顶点 v_1 , 线网 n 与 m 相邻, 图(e)-(f)

情形 4: 线网 m 起始于扩散源 e 的顶点 v_1 , 线网 n 与 m 相邻, 图(g)

扩散窗扩散平衡条件

- 条件I：给定阈值 ψ^* ，如果拥挤度 $d_e(t) \leq \psi^*$ ，那么扩散达到平衡。
- 条件II：当下一个momentary-diffusion是过度扩散(over diffusion)时，扩散达到平衡
 - ⊙ 过度扩散(over diffusion)指这样一个momentary-diffusion操作后，扩散源的浓度反而小于了它的扩散边集 Ed 的浓度
- 条件III：扩散被障碍阻挡或者迭代过程中被禁止时，扩散达到平衡

扩散迭代的收敛

- 在扩散迭代的过程中，维护了一个最大堆 H 和一个禁忌表 Tb
 - ⊙ 最大堆 H 维护了所有可能的扩散源，并以边 e 的浓度为值建堆
 - ⊙ 堆顶 e 被选择为扩散源，扩散平衡后 e 从 H 中被释放
 - ⊙ 禁忌表 Tb 维护了所有不能够再有线网扩散到其上的边 e
 - ⊙ 当由于扩散窗扩散平衡条件 Π 达到平衡时，该扩散源 e 将被添加到禁忌表 Tb
 - ⊙ 当任何禁忌边 e 的邻边的浓度降低时，该禁忌边 e 才会被释放

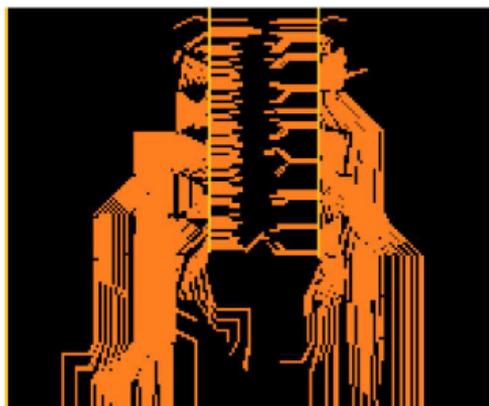
实验结果

- 对negotiation-based实验的10个例子，进行测试比较
 - [Liu's]为[S. Liu, DAC' 08][S. Liu, TCAD' 09]的工作
 - Nego为negotiation-based的改进版

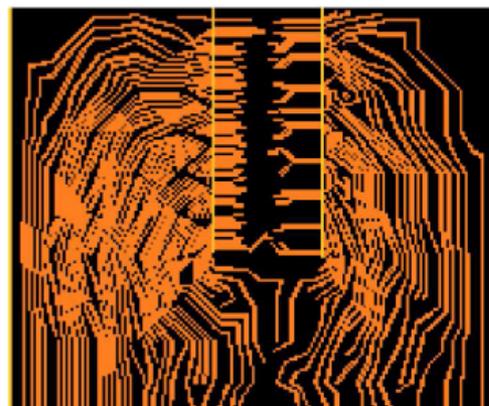
Test Cases ID	Number of failed nets			Wire length (mm)			Runtime (s)		
	[Liu's]	Nego	D-router	[Liu's]	Nego	D-R	[Liu's]	Nego	D-router
Q1	51	41	26	1.64	1.69	1.70	5.34	9.79	7.17
B2	31	30	0	6.98	6.98	7.17	11.39	17.84	9.00
F3	24	22	0	7.79	7.80	7.96	14.36	14.41	16.91
P4	135	135	48	11.90	11.90	12.30	41.64	20.04	13.87
A5	64	63	7	14.90	14.90	16.30	15.27	17.65	10.92
A6	60	57	15	4.98	4.99	4.93	12.12	18.77	12.87
X7	45	45	8	6.55	6.54	6.53	39.51	45.11	25.38
A8	16	16	0	18.50	18.50	18.70	44.55	47.24	9.32
S9	22	20	0	1.67	1.67	1.65	2.11	3.2	0.96
S10	32	32	0	9.53	9.53	7.90	284.17	286.34	3.01
total	480	461	104 1/4.6x 1/4.4x	—	—	—	—	—	—
average	—	—	—	8.45	8.46	8.51	46.05	47.04	10.94 1/4.2x 1/4.3x

- 同[Liu's]相比D-router
 - 未布成功的线网减少为104，为原来的1/4.6
 - 运行时间则降低为原来的1/4.2

布线结果



(a)

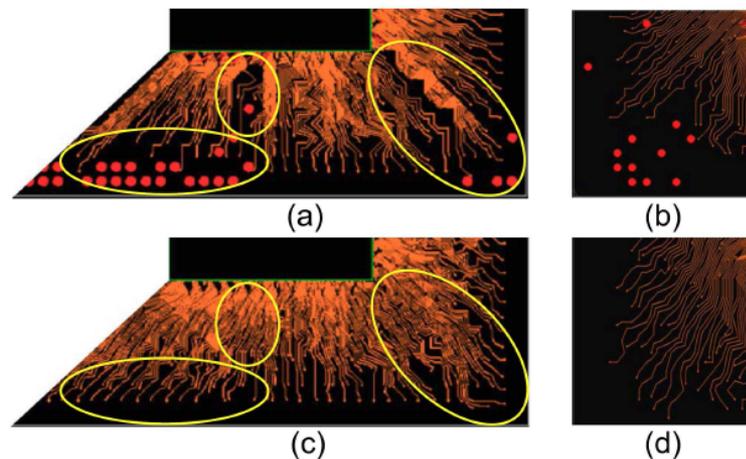


(b)

(a)为D-router布线前(b)为布线后结果

- 左边图示了例子B2的一个局部放大图

- (a) (b)是来自[Liu' s]的结果
- (c) (d)是D-router的布线结果
- 红色亮点为未布通的线网



相关工作的发表刊物

- 该工作发表于“ISPD’ 09会议论文集”。
- 更多的细节，复杂度说明，以及实验比较等形成的论文被“ACM Trans. on Design Automation of Electronic System, TODAES”所接受。

内容组织

- 背景及现状
 - ◎ 封装
 - ◎ 封装中的布线
 - ◎ 封装中衬底布线现状
- 已有相关工作概述
- 博士论文工作的研究内容
 - ◎ 基于OpenAccess数据平台的芯片中快速可布性分析算法
 - ◎ 基于动态推(dynamic pushing)和end-zone的衬底拓扑布线算法
 - ◎ Negotiation-based衬底拓扑布线
 - ◎ 衬底拓扑布线中扩散(diffusion)驱动的拥挤消除算法
- 总结与展望

本文的创新

- 提出衬底布线的 *end-zone* 模型的概念
 - ⊙ 解决了衬底布线工艺中，交错通孔(staggered via)分配问题
 - ⊙ 使布线问题更加具有灵活性，提高布通率
- 提出基于A*的寻路算法的ADS*算法
 - ⊙ 采用了动态推(dynamic pushing)的布线思想
 - ⊙ 该思想解决了衬底平面布线时，绕线关系对布线顺序的依赖
 - ⊙ 该算法支持实现了 *end-zone* 模型上的寻路
- 提出基于negotiation-based思想的衬底布线算法
 - ⊙ 增加了迭代过程中ADS*寻路算法对于历史拥挤度和当前拥挤度的考虑
- 提出封装衬底中扩散(diffusion)驱动的拥挤消除算法
 - ⊙ 这在拥挤驱动布线算法中是一次新的尝试
 - ⊙ 该算法解削弱了拥挤驱动的拆线重布时，重布顺序对布线结果的影响

总结和展望

- 我们首先研究了芯片上的布线拥挤估计算法
 - 改进的概率预估的算法，和概率指导下的布线给出了快速的拥挤估计
- 针对日益复杂的芯片封装工艺下的衬底拓扑布线算法进行了研究
 - 尤其是系统级封装(SiP)下的衬底布线
 - 动态推(dynamic pushing), negation-based的考虑历史和当前拥挤的布线迭代，扩散(diffusion)驱动的拥挤消除，以及end-zone模型使算法削弱了对布线顺序的依赖，提高了效率和布通率。
- 然而，衬底布线工艺下，对于球形触点的分配不是一成不变的
 - 考虑球形触点重分配的灵活性的布线
 - 考虑到信号线网的RLC相关的信号完整性约束
 - 衬底上的power domain的约束
 - 以及I/O时延相关的differential pair的约束
 - 是值得进一步研究的

发表论文列表 (1)

- [1] (会议) *Shenghua Liu*, Xianlong Hong, Tong Jing, Jingyu Xu, FREe: A Fast Routability Estimator, In: Proceedings of IEEE ICCAS, 2006, Guilin, China, pp.2454-2458. (EI索引, 索引号9319582)
- [2] (专利) 一个快速的集成电路可布性分析方法, 申请日期: 6/15/2006, 申请号: ZL 00610012271.4, 授权日期: 7/23/2008, 授权号: CN 100405379C, 发明人: 洪先龙, 经彤, *刘盛华*, 许静宇 (本人为第三发明人, 前两位为正副导师, 专利终止日期2028年7月23, 专利权期限为20年)
- [3] (会议) *Shenghua Liu*, Guoqiang Chen, Tom Tong Jing, Lei He, Tianpei Zhang, Robi Dutta, Xian-Long Hong, "Topological Routing to Maximize Routability for Package Substrate," In Proc. of 45th DAC, Anaheim, June 2008, pp.566-569. (A类国际知名会议, EI索引, 索引号083811565835)
- [4] (期刊) *刘盛华*, 洪先龙, 经彤, 许静宇, "集成电路物理设计中的快速可布性评估算法," 计算机辅助设计和图形学学报, 第20卷, 第12期, 2008年12月, pp1541-1549. (EI索引)
- [5] (期刊) *Shenghua Liu*, Guoqiang Chen, Tom Tong Jing, Lei He, Tianpei Zhang, Robi Dutta, Xian-Long Hong, "Substrate Topological Routing for High Density Packages," IEEE trans. on CAD, Vol.28, No.2, Feb. 2009, pp207-216. (SCI/EI indexing 5-year impact factor 1.867)

发表论文列表 (2)

- [6] (会议) *Shenghua Liu*, Guoqiang Chen, Tom Tong Jing, Lei He, Robi Dutta, Xian-Long Hong, “Diffusion-Driven Congestion Reduction for Substrate Topological Routing” , In Proc. of ISPD 2009, Mar. 2009, pp175-180. (国际知名会议)
- [7] (会议) *Shenghua Liu*, Yuchun Ma, Xian-Long Hong, Yu Wang, “Simultaneous Slack Budgeting and Retiming for Synchronous Circuits Optimization” , In Proc. of 15th ASP-DAC, 2009, pp. 49-54. (最佳论文提名, 领域知名会议)
- [8] (期刊) *Shenghua Liu*, Guoqiang Chen, Tom Tong Jing, Lei He, Robi Dutta, Xian-Long Hong, “Effective Congestion Reduction for IC Package Substrate Routing,” ACM Trans. on Design Automation of Electronic System, 2010. (SCI索引, 5-year impact factor 0.761, 录用待发表)
- 已发表论文和专利共计7篇, 一篇已经接受待发表
 - 其中论文均为**第一**作者; 专利按学校要求为**第三**作者前两位作者为正副导师。

谢谢!