

基于蒙特卡洛方法的触发器聚类布局算法^①

王昊^②* * * * * 刘臻* * * * * 王剑* * * * * 章隆兵* * * * *

(^{*}计算机体系结构国家重点实验室(中国科学院计算技术研究所) 北京 100190)

(^{**}中国科学院计算技术研究所 北京 100190)

(^{***}中国科学院大学 北京 100049)

(^{****}龙芯中科技术有限公司 北京 100095)

摘要 针对触发器聚类的布局问题,提出了一种基于蒙特卡洛方法的触发器聚类布局算法。该方法的核心思想为设计合理的评分函数,并通过蒙特卡洛方法模拟触发器聚类在版图上的随机游走过程,获取样本,再通过评分函数评价样本质量,逐步地在可行域中搜索最优解。实验结果表明,相较于现有的方法,该方法在总线长与时序性能两方面,分别得到平均约 9% 与 17% 的改善与提升。此外,该方法具有较强的健壮性,适用于复杂度较高或者版图利用率较高的设计,并能够快速收敛,以较小的时间代价完成优化。

关键词 触发器聚类, 布局, 蒙特卡洛方法, 层次化设计方法

0 引言

随着半导体工艺的演进,晶体管的集成度进一步提高,使得超大规模集成电路(very large scale integration, VLSI)的设计复杂度日益增大。为了应对设计复杂度的挑战,设计人员通常采用层次化设计方法(hierarchical design methodology)。该方法根据物理设计的不同阶段,将整体设计划分为若干抽象层次,且具有以下优点^[1]: (1) 与扁平化设计方法相比,更容易完成工程变更指令(engineering change order, ECO); (2) 对于不同的抽象层次,物理设计可以并行推进; (3) 能有效地缩短设计的迭代周期; (4) 可以更好地优化时钟树结构。

一个重要的抽象层次是触发器聚类(register cluster),它作用于物理设计的布局(placement)与时钟树综合(clock tree synthesis)阶段。本文研究触发器聚类布局算法。在工程实践中,根据设计的实际

需求,如局部时钟系统优化与有用时钟偏差调度等,触发器将被划分成若干触发器聚类^[2]。在同一触发器聚类中的触发器,具有物理局部性(physical locality)^[2],即这些触发器在布局阶段,将被放置在附近的位置。又因为触发器是集成电路的基本时序单元,触发器的位置将决定集成电路的工作频率。因此,针对触发器聚类的布局进行优化,亦是在该层次上对触发器的布局进行优化,从而实现对设计时序性能的优化。本文着力研究触发器聚类的布局优化方法,提出了一种基于蒙特卡洛方法的触发器聚类布局算法。

1 相关工作

布局是专用集成电路(ASIC)流程中的一个重要阶段。数十年来,布局优化算法持续地被研究者们改进。当半导体工艺来到深亚微米节点,由于集成电路的设计复杂度急剧增长,布局优化算法需要

^① 国家“核高基”科技重大专项课题(2014ZX01020201, 2014ZX01030101),国家自然科学基金(61432016)和863计划(2013AA014301)资助项目。

^② 男,1988年生,博士生;研究方向:计算机体系结构;联系人,E-mail: wanghaocpu@ict.ac.cn
(收稿日期:2016-09-28)

一些新的特性。

近期的大多数研究关注于布局与布线 (routing) 之间的一致性。例如,文献[3,4]在布局阶段,通过解析方法估算线长与线密度;文献[5,6]将全局布线器与布局器结合在一起;文献[4,7,8]在布局阶段,引入线宽、布线通道等布线信息;文献[5,9,10]增加了布局与布线细粒度优化的步骤。这些研究工作在布局阶段维持了较好的可布线性,从而减轻了布线阶段的拥塞 (congestion) 情况。

然而,上述研究本质上都是对扁平化设计方法(即不考虑电路逻辑层次的设计方法)的改进,而文献[11]与[12]是近期少有的与层次化设计相结合的工作。文献[11]指出存储阵列的组织结构具有非常规整的特点,因而它在存储阵列的层次上,提出一种可控的针对标准单元的布局方法;文献[12]着眼于数据路径 (data-path) 中的触发器聚类的布局问题,并设计了一种基于线性规划与最小费用网络流的优化算法。虽然文献[12]已经认识到触发器聚类布局优化的重要性,但是它的方法存在以下局限性:(1) 数据路径是一类数据流较为简单的设计(文中将其比作比特流通道);(2) 数据路径是一类版图面积利用率(器件面积与版图面积的比值)较

低的设计。换言之,对于一般的集成电路设计,该方法并不适用。

针对以上局限性,本文首先分析了触发器聚类的相互关系,并将其抽象成数据流图;在此基础上,提出了一种基于蒙特卡洛方法的触发器聚类布局算法,并通过实验验证了该方法的有效性与健壮性。

2 数据流分析

如果数据信号从一个触发器出发,经过组合逻辑单元,到达另一个触发器,那么这两个触发器之间存在一条数据流。图 1 是数据流的示例。数据信号通过组合逻辑的时延,不仅由设计功能所决定,而且与触发器的位置有密切的关系:所以通过分析触发器之间的数据流关系,明确数据信号在版图中的流向,才能有效地规划触发器在版图上的位置,使设计的时序性能在布局阶段得到充分的优化。按上文所述,触发器聚类是包含触发器的抽象层次。触发器的数据流关系应当在触发器聚类的层次中体现出来。因此,本文引入了一种重要的数据结构——数据流图。

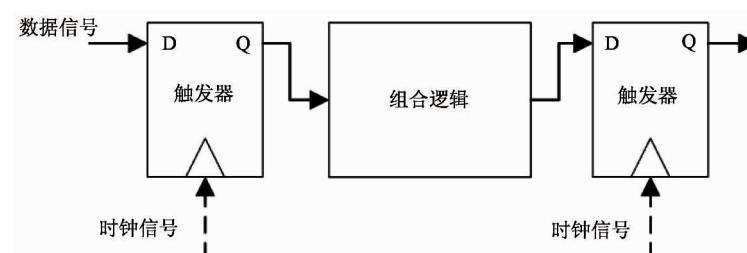


图 1 两个触发器之间的数据流示意图

图 2 是数据流图建立过程的一个简单示例。数据流图建立过程的形式化描述如下:

设 C 为触发器聚类的集合,令 $|C|$ 为 C 中的触发器聚类的个数。设 $G(V, E)$ 为 C 对应的数据流图,其中 V 为 G 的顶点的集合, E 为 G 的边的集合。对于任意 $c_i \in C$, 构造 $v_i \in V$ 。用 r 作为触发器的记号。如果存在从任意 $r \in c_j$ 流向任意 $r \in c_j$ 的数据流,那么构造边 $e_{ij} \in E$ 。对于任意 $e_{ij} \in E$, 如果 c_i 中

的 m 个触发器拥有流向 c_j 中的 n 个触发器的数据流,那么 e_{ij} 的权重 w_{ij} 为 $\max(m, n)$ 。

数据流图将数据流抽象成边,并用权重反映触发器聚类在数据流上的相关程度。从直观上来说,数据流的相关程度较为紧密的触发器聚类,在版图上的相对位置应当更为接近,以减小数据传播的时间代价,提升设计的时序性能;而不存在数据流关系的触发器聚类,则不需要对它们在版图上的相对位

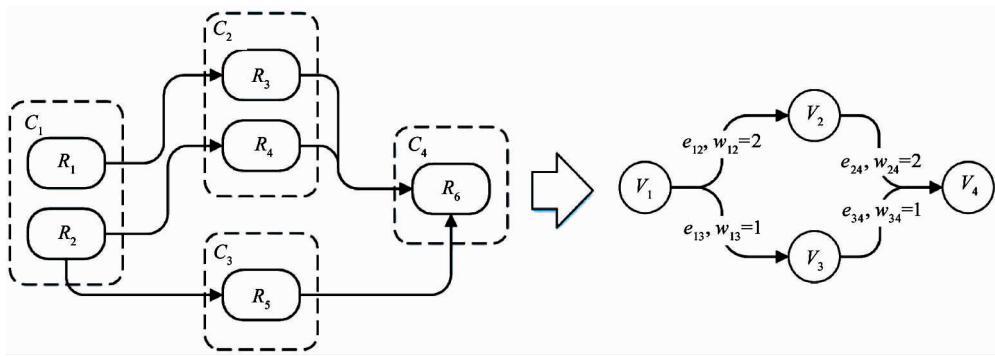


图2 数据流图建立过程示例

置加以约束,减轻触发器聚类算法的负荷。以此直观判断为基础,下节将量化分析数据流图在触发器聚类问题中的作用,并详细阐述本文提出的蒙特卡洛-触发器聚类布局算法。

3 蒙特卡洛-触发器聚类布局算法

蒙特卡洛方法是一类随机算法,它广泛应用于对物理过程或者生化过程的模拟,以及一些最优化问题的求解^[13]。以分子模拟计算为例,简要说明蒙特卡洛方法的基本思想:(1)生成随机的分子构型;(2)对该分子构型的某粒子做无规则改变,生成新的分子构型;(3)计算新的分子构型的能量;(4)比较新旧两个分子构型的能量变化:若新的分子构型能量低于旧的分子构型能量,则接受新的分子构型,并用该分子构型开始下一轮迭代;若新的分子构型能量高于旧的分子构型,则生成一个随机数与玻尔兹曼因子比较,若随机数小于玻尔兹曼因子,则接受新的分子构型,并用该分子构型开始下一轮迭代,否则放弃新的分子构型;(5)迭代步骤(2)~步骤(4),直至获得满足能量约束的分子构型。本文将蒙特卡洛方法应用于触发器聚类布局优化问题,提出一种蒙特卡洛-触发器聚类布局算法。本节分为算法流程与算法描述两部分。

3.1 算法流程

图3是算法的总体流程,其中设计网表、粗略布局与布局优化为传统的ASIC布局流程。设计网表是总体流程的输入,粗略布局是由布局器对设计网表的时序、面积等粗略评估后,完成的布局。根据设

计网表,还能获取触发器聚类,从而生成数据流图。触发器聚类布局以数据流图与粗略布局为输入,再将布局结果交由布局器完成布局优化。如图3所示,触发器聚类布局能够与传统布局流程很好地结合在一起。

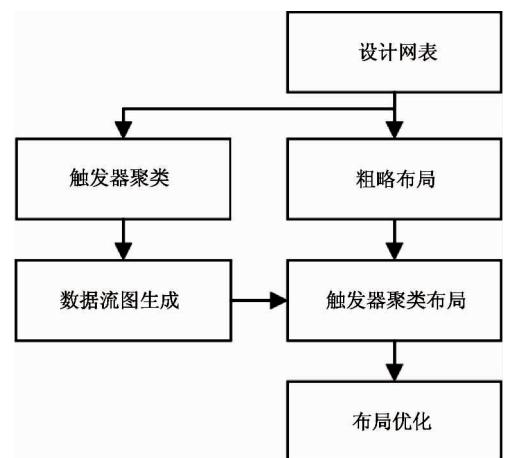


图3 算法总体流程图

3.2 算法描述

算法的起始阶段需要对触发器聚类的位置与尺寸进行初始化。触发器聚类位置的初始化方法类似于物理学中质心的确定方法:

对于任意 $c_i \in C$, 将 c_i 的初始中心坐标标记作 (x_i^0, y_i^0) 。对于任意 $r \in c_i$, 令 (x_r, y_r) 为 r 在粗略布局里的中心坐标。那么 (x_i^0, y_i^0) 可由下式决定:

$$\begin{cases} x_i^0 = \frac{\sum_{r \in c_i} x_r}{|c_i|} \\ y_i^0 = \frac{\sum_{r \in c_i} y_r}{|c_i|} \end{cases} \quad (1)$$

其中 $|c_i|$ 为触发器聚类 c_i 中的触发器个数。

触发器聚类的尺寸调节不应造成触发器聚类内部单元密度过高,否则触发器聚类内部的拥塞将损害局部时钟系统的质量。然而,如何调节尺寸并不是本文所关注的问题。为简单起见,本文规定触发器聚类的形状为正方形,且触发器聚类的面积利用率(即触发器聚类中的触发器总面积与该聚类面积的比值)与版图面积利用率相等。这个规定使触发

器聚类内部的单元密度不会高于版图其它部分的单元密度。

接着,给出算法使用的重要变量的定义:

(1) 定义重叠面积矩阵 \mathbf{O} ,量化触发器聚类相互重叠的程度。其构造方式为:对于任意 $c_i \in C$ 与任意 $c_j \in C$,如果 $i = j$,那么 $o_{ij} = 0$;否则 o_{ij} 的值等于 c_i 与 c_j 重叠部分的面积。 \mathbf{O} 是一个 $|C| \times |C|$ 的矩阵。算法 1 为重叠面积的计算方法。

算法 1:重叠面积计算

```

输入:触发器聚类  $c_i$  与  $c_j$ 
输出: $c_i$  与  $c_j$  的重叠面积
令 left_i、right_i、top_i、bottom_i 分别为  $c_i$  的左、右、上、下边界
令 left_j、right_j、top_j、bottom_j 分别为  $c_j$  的左、右、上、下边界
if left_i ≥ right_j, or right_i ≤ left_j, or top_i ≤ bottom_j, or bottom_i ≥ top_j then
    重叠面积 = 0
else
    x[0:3] left_i, right_i, left_j 与 right_j 的升序排列
    y[0:3] top_i, bottom_i, top_j 与 bottom_j 的升序排列
    重叠面积 = ( $x[2] - x[1]$ ) × ( $y[2] - y[1]$ )
end if

```

(2) 定义线长矩阵 \mathbf{L} ,记录任意两个触发器聚类之间的线长。其构造方式为:对于任意 $c_i \in C$ 与任意 $c_j \in C$,如果 $i = j$ 或者 $e_{ij} \notin E$,那么 $l_{ij} = 0$;否则, $l_{ij} = w_{ij}(|x_i - x_j| + |y_i - y_j|)$ 。 \mathbf{L} 是一个 $|C| \times |C|$ 的矩阵。

(3) 定义行向量 \mathbf{d} ,记录每一个触发器聚类离它的初始位置的偏移量。其构造方式为:对于任意 $c_i \in C$, $d_i = |x_i - x_i^0| + |y_i - y_i^0|$,其中 (x_i, y_i) 为 c_i 布局后的中心坐标。 \mathbf{d} 是一个 $1 \times |C|$ 的行向量。

然后,明确算法的优化目标:

(1) 消除触发器聚类之间的重叠。如果触发器聚类相互重叠,则触发器可放置的空间变小、时钟树的布线通道变少,不利于局部时钟系统的优化。

(2) 权衡总线长与触发器聚类相对初始位置的总偏移量。因为布局阶段缺少金属线的寄生参数,无法获取准确的时序信息,所以本文选择总线长以间接评估设计的时序性能。一般而言,总线长越小,数据信号的传播路径越短,传播时延也越小。虽然

根据的定义,总线长可以得到最小值,但如果触发器聚类相距过近,导致数据流方向上没有足够的空间放置组合逻辑单元,那么由此产生的局部拥塞将影响布局阶段的时序优化。因此,通过定义 \mathbf{d} ,引入总偏移量,限制触发器聚类的可移动范围,与总线长达到一个相对平衡点。

评分函数是蒙特卡洛方法的核心部分,算法的优化目标包含于评分函数之中。本文的评分函数定义如下:

$$H = \alpha \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} o_{ij} + \beta \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} l_{ij} + \gamma \sum_{i=1}^{|C|} d_i \quad (2)$$

其中 α, β 与 γ 为三个非负实数,是评分函数的权重系数。评分函数的第一项是触发器聚类的重叠面积之和,第二项、第三项分别为总线长与总偏移量。在给定权重系数的条件下,评分函数的值越小,算法优化的效果越好,触发器聚类布局的质量则越好。

算法 2 为蒙特卡洛-触发器聚类布局算法。其后附了算法中的两个函数。需要说明的是随机数在

其中的作用。算法 2 的随机数使算法以 $e^{H-H_{\text{new}}}$ 的概率接受质量较差的布局,从而跳出局部可行域,继续搜索全局最优解。函数 1 的随机数起到随机抽样的作用,每个触发器聚类被选中的概率与其重叠面积或者线长成正比例关系。函数 2 的随机数实现触发器聚类在版图上的随机游走,常量是每一次游走的步长。

设触发器聚类的数目为 n ,最大迭代次数为 a 。算法的初始化需要计对矩阵各个元素求和,时间复杂度为 $O(n^2)$ 。在 while 循环中,随机抽样的时间复杂度为 $O(n)$;更新 $\mathbf{O}、\mathbf{L}$ 与 H 时,只需要更新与被选

算法 2:蒙特卡洛-触发器聚类布局算法

输入:粗略布局与数据流图

输出:触发器聚类布局

初始化 $\mathbf{O}、\mathbf{L}$ 与 H

while 当前迭代次数 < 最大迭代次数 **do**

if 触发器聚类的重叠面积 > 0 **then**

$c_k = \text{random_selection}(\mathbf{O})$

else then

$c_k = \text{random_selection}(\mathbf{L})$

end if

$(x_k, y_k) = \text{random_walk}(c_k)$

 更新 $\mathbf{O}、\mathbf{L}$ 与 H ,得到 $\mathbf{O}_{\text{new}}、\mathbf{L}_{\text{new}}$ 与 H_{new}

 生成随机数 q

if $H < H_{\text{new}}$ 或 $q < e^{H-H_{\text{new}}}$ **then**

 接受新的布局

else

 回滚至上一个布局

end if

end while

函数 1:random_selection

参数: $|C| \times |C|$ 的矩阵 \mathbf{M}

生成随机数 q

$\text{bound} = 0$

for $i = 1 : |C|$

$$\text{bound} = \text{bound} + \frac{\sum_{j=1}^{|C|} m_{ij}}{\sum_{i=1}^{|C|} \sum_{j=1}^{|C|} m_{ij}}$$

if $q \leq \text{bound}$ **then**

return c_i

end if

end for

函数 2:random_walk

参数:触发器聚类 c

生成随机数 q

获取 c 的坐标 (x, y)

$$x = x + A \cos(2\pi q)$$

$$y = y + A \sin(2\pi q)$$

return (x, y)

中的触发器聚类相关的矩阵元素,所以时间复杂度为 $O(n)$;随机游走的时间复杂度为 $O(1)$ 。综上所述,算法的时间复杂度为 $O(n^2 + an)$ 。

4 实验与结果分析

4.1 实验环境

本文设计的实验是用 TCL 与 MATLAB 实现的。实验使用的硬件平台为 24 核的 x86 服务器,主频 2.1GHz。

为了验证蒙特卡洛-触发器布局算法的有效性,本文设计了两组实验:

(1) 分析评分函数中的权重对布局质量(主要从总线长与时序性能两方面评价)的影响;

(2) 在不同的版图面积利用率与设计复杂度下,对比蒙特卡洛-触发器布局算法与文献[12]的布局质量。

实验流程如图 3 所示,粗略布局与布局优化使用的布局器为 IC Compiler,获取触发器聚类的方法参照文献[14]。文献[15]为学术界常用的基准电路,但因为它的逻辑结构过于简单,无法体现触发器聚类的复杂性,所以实验使用的设计都来自工程实践。为了充分反映出评分函数的变化情况以及获得足够多的样本,最大迭代次数设为 10 万次。

4.2 实验结果分析

表 1 为第一组实验的结果。70% 的版图面积利用率是工程实践中的常用约束。 α 设为常量,并且是三者中的最大值。 β 与 γ 则分别在 [1,1000] 的区间里取值,共获得 7 组实验数据。实验结果表明,当 β 远大于 γ 时,比如 $\beta = 1000, \gamma = 1$ 或者 $\beta = 100, \gamma = 1$,由于过于强调总线长的优化,触发器聚类相距过近,甚至重叠在一起,导致组合逻辑单元在数据

流动的方向上缺少足够的可放置空间,使得设计的总线长与时序性能没有得到充分的优化。然而,当 $\beta = 1, \gamma = 1000$ 时,虽然总线长与时序违例是 7 组数据中的最优值,但因为过度限制了触发器聚类的可移动范围,所以没有完全消除触发器聚类的相互重叠,不利于构建局部时钟系统。当 $\beta = 1, \gamma = 1$ 或

者 $\beta = 1, \gamma = 10$ 时,触发器聚类的重叠面积为零,且设计的总线长与时序性能得到了较好的优化。综上所述,第一组实验证明:(1) 赋予 α 最大的权重值,能够有效地消除触发器聚类的重叠;(2) 合理权衡 β 与 γ 的权重值,能够让设计在总线长与时序性能两方面得到较好的优化。

表 1 第一组实验的实验结果

设计	REG	UTL	权重系数			蒙特卡洛-触发器布局算法			
			α	β	γ	TWL ($\times 10^3 \mu\text{m}$)	WNS (ns)	TNS (ns)	OVA (μm^2)
1	728	70%	1000	1000	1	575	-0.27	-47.9	5245.5
				100	1	571	-0.25	-50.1	94.6
				10	1	568	-0.24	-39.7	0.0
				1	1	561	-0.25	-37.6	0.0
				1	10	559	-0.23	-42.1	0.0
				1	100	564	-0.28	-44.0	0.3
				1	1000	530	-0.23	-34.6	5.9

REG: 触发器数量 UTL: 版图面积利用率 TWL: 总线长 WNS: 最坏时序违例 TNS: 总体时序违例 OVA: 总重叠面积

表 2 为第二组实验的结果。从触发器数量可以看出,设计 2 ~ 设计 4(见图 4)的设计复杂度呈递增关系。对于每一个设计,又分 10%、30%、50%、

70%、90% 等 5 个版图面积利用率进行对比实验。实验结果表明,当设计复杂度与版图面积利用率较低时,如版图面积利用率为 10% 或 30% 的设计 2,

表 2 第二组实验的实验结果

设计	REG	UTL	文献[12]			蒙特卡洛-触发器布局算法			改善幅度		
			TWL ($\times 10^3 \mu\text{m}$)	WNS (ns)	TNS (ns)	TWL ($\times 10^3 \mu\text{m}$)	WNS (ns)	TNS (ns)	TWL	WNS	TNS
2	1746	50%	10%	971	-0.30	-229.0	961	-0.31	-252.1	1.03%	-3.33%
			30%	780	-0.25	-187.5	767	-0.26	-195.8	1.67%	-4.00%
			70%	726	-0.28	-208.0	719	-0.27	-196.5	0.96%	3.57%
			90%	731	-0.28	-204.7	698	-0.26	-186.8	4.51%	7.14%
			10%	710	-0.34	-256.1	678	-0.29	-228.5	4.51%	14.71%
3	3936	50%	10%	1190	-0.18	-36.5	1079	-0.16	-36.7	9.33%	11.11%
			30%	907	-0.14	-26.8	830	-0.13	-23.4	8.49%	7.14%
			70%	959	-0.16	-32.8	737	-0.13	-24.2	23.15%	18.75%
			90%	784	-0.18	-31.4	718	-0.14	-25.9	8.42%	22.22%
4	7415	50%	10%	884	-0.42	-37.0	674	-0.24	-13.4	23.76%	42.86%
			30%	3659	-0.18	-28.5	3751	-0.14	-20.1	-2.51%	22.22%
			70%	3298	-0.17	-86.4	2922	-0.15	-58.9	11.40%	11.76%
			90%	2918	-0.10	-18.1	2706	-0.06	-11.3	7.27%	40.00%
			10%	3276	-0.24	-31.4	2390	-0.12	-12.5	27.05%	50.00%
平均值									9.22%	17.44%	20.66%

REG: 触发器数量 UTL: 版图面积利用率 TWL: 总线长 WNS: 最坏时序违例 TNS: 总体时序违例

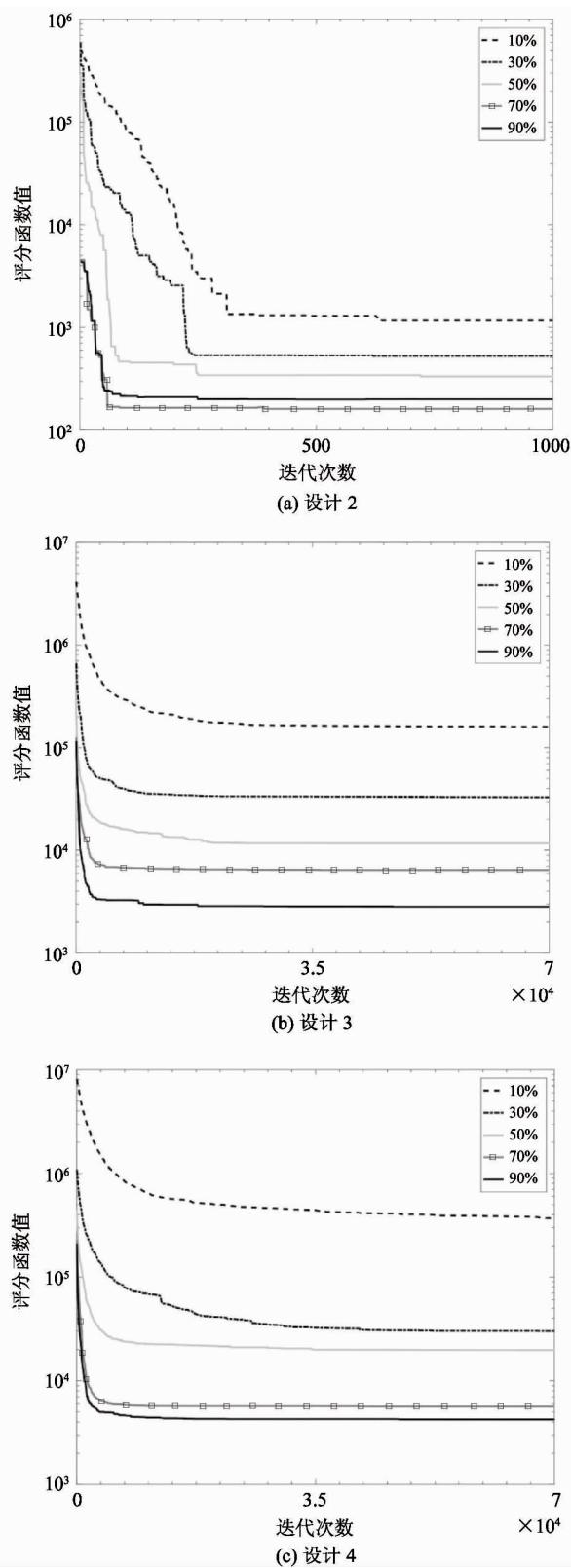


图4 不同版图面积利用率下的评分函数的收敛曲线

两种方法生成的布局质量几乎相等。但随着设计复杂度或者版图面积利用率增大,本文的算法生成的布局质量逐渐优于文献[12]的方法。在版图面积

利用率为90%的设计4中,文献[12]的方法甚至不能完成触发器聚类的布局。究其原因在于该方法所使用的线性规划模型。较高的设计复杂度与版图面积利用率,意味着需要更多的约束条件来描述触发器聚类之间的数据流关系,以及约束它们之间的距离以避免出现相互重叠的情况。实际上,这些约束的完备性是对可行域大小的悲观估计,使线性规划模型得不到全局最优解,甚至不可解。相对地,本文的算法从可行域的某一处开始搜索局部最优解,并且具有一定概率跳出局部可行域、向全局最优解逼近。因此,它比文献[12]的方法具有更强的健壮性。

需要说明的是,线长矩阵计算的是触发器聚类之间的总线长,而表1与表2中的总线长是布局器在布局阶段对金属线长的估算,但表1的实验结果证明线长矩阵对布局器在总线长方面的影响是符合预期的,进一步验证了本方法的有效性。

通过记录每一次迭代的评分函数值,可以分析算法的收敛速度。图4展示了设计2~设计4的评分函数的收敛曲线。从图上可以看出,对于不同的设计复杂度与不同的版图面积利用率,本文的算法都能够迅速地收敛。另外,收敛曲线并不是严格单调递减的曲线,它存在一些由于算法接受质量较差的布局而产生的“毛刺”。

5 结论

本文着力研究触发器聚类布局问题,将蒙特卡洛方法与该问题结合,提出了一种新的触发器聚类布局算法。实验证明,该算法能够消除触发器聚类的重叠,而且与现有的方法相比,它在总线长与最坏时序违例两方面,能分别得到平均约9%与17%的改善。对于复杂度较高或者版图面积利用率较高的设计,该算法依然适用,具有较强的健壮性。此外,该算法具有快速收敛性,能够以较低的时间代价完成优化。在未来的研究工作中,如何在布局阶段更精确地估计设计的时序性能,是该算法的改进目标,从而不需要设计者在总线长与总偏移量之间权衡,提高算法的自动化程度。

参考文献

- [1] Reda S, Chowdhary A. Effective linear programming based placement methods. In: Proceedings of International Symposium on Physical Design, San Jose, USA. 2006. 186-191
- [2] Yang L, Zhao J Y, Fan B X, et al. Logical clustering for fast clock skew scheduling. In: Proceedings of International Midwest Symposium on Circuits and Systems, Cancun, Mexico, 2009. 208-211
- [3] Tsota K, Koh C K, Balakrishnan V. Guiding global placement with wire density. In: Proceedings of International Conference on Computer-Aided Design, San Jose, USA, 2008. 212-217
- [4] Lu J W, Chen P W, Chang C C, et al. ePlace: electrostatics based placement using Nesterov's method. In: Proceedings of Design Automation Conference, San Francisco, USA, 2014. 706-711
- [5] He X, Huang T, Chow W K, et al. Ripple 2.0: high quality routability-driven placement via global router integration. In: Proceedings of Design Automation Conference, Austin, USA, 2013. 1044-1049
- [6] Hu J, Kim M C, Markov I L. Taming the complexity of coordinated place and route. In: Proceedings of Design Automation Conference, Austin, USA, 2013. 1031-1037
- [7] Li S, Koh C K. Analytical placement of mixed-size circuits for better detailed-routability. In: Proceedings of Asia and South Pacific Design Automation Conference, Singapore, 2014. 41-46
- [8] Chen Y F, Huang C C, Chiou C H, et al. Routability-driven blockage-aware macro placement. In: Proceedings of Design Automation Conference, San Francisco, 2014. 724-729
- [9] Lin T, Chu C. POLAR 2.0: An Effective Routability-Driven Placer. In: Proceedings of Design Automation Conference, San Francisco, USA, 2014. 718-723
- [10] Liu W H, Koh C K, Li Y L. Optimization of placement solutions for routability. In: Proceedings of Design Automation Conference, Austin, USA, 2013. 1050-1058
- [11] Teman A, Rossi D, Meinerzhagen P, et al. Controlled placement of standard cell memory arrays for high density and low power in 28nm FD-SOI. In: Proceedings of Asia and South Pacific Design Automation Conference, Chiba, Japan, 2015. 81-86
- [12] Cho M, Xiang H, Ren H X, Ziegler M M, et al. Latch-Planner: latch placement algorithm for datapath-oriented high-performance VLSI designs. In: Proceedings of International Conference on Computer-Aided Design, San Jose, USA, 2013. 342-348
- [13] Giordano N J, Nakanishi H. Computational physics (second edition), USA: Pearson, 2005. 235-474
- [14] Yang L, Fan B X, Cong M, et al. Register relocation to optimize clock network for multi-domain clock skew scheduling. In: Proceedings of International Symposium on Circuits and Systems, Paris, France, 2010. 3180-3183
- [15] Brglez F. Benchmarks at CBL up to 1996. <http://people.engr.ncsu.edu/brglez/CBL/benchmarks/Benchmarks-upto-1996.html>: NC State University, 2015

An algorithm for register cluster placement based on the Monte Carlo method

Wang Hao * * * *, Liu Zhen * * * *, Wang Jian * * * **, Zhang Longbing * * * ***

(* State Key Laboratory of Computer Architecture (Institute of Computing Technology, Chinese Academy of Sciences), Beijing 100190)

(** Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(*** University of Chinese Academy of Sciences, Beijing 100049)

(**** Loongson Technology Corporation Limited, Beijing 100095)

Abstract

This study focused on the placement of register clusters, and proposed a novel placement algorithm based on the Monte Carlo method. The main idea of the algorithm is to simulate the random walking process of register clusters, use a reasonable scoring function to evaluate the quality of the samples gained by the Monte Carlo method, and eventually search for the optimal solution in the feasible region. Experiments prove that, compared to the existing approach, this algorithm can achieve about 9% and 17% improvement in the total wire length and the timing performance respectively. This algorithm is robust, and fits for the designs with the high complexity or high utilization ratio of layout. In addition it converges fast, and achieves the optimization at a low cost of time.

Key words: register clustering, placement, Monte Carlo method, hierarchical design methodology