

基于扇形模板的数字集成电路规律性提取算法

潘伟涛, 谢元斌, 郝 跃, 史江义

(西安电子科技大学微电子学院宽禁带半导体材料与器件教育部重点实验室, 陕西西安 710071)

摘 要: 将数据挖掘方法用于数字集成电路规律性提取, 提出了一种扇形结构模板的规律性提取算法. 采用压缩式存储及删除缓冲器结构等方法, 降低了电路的存储空间. 通过边权值编码, 实现了逐级产生扇形频繁子电路的算法, 解决了传统算法对大规模数字集成电路规律性提取时间复杂度过高的问题. 实验结果表明该算法比 SPOG 与 TREE 算法更能充分提取电路的规律性, 规律性提取时间更短.

关键词: 频繁子电路; 数据挖掘; 规律性提取; 规则性系数

中图分类号: TP391.72 **文献标识码:** A **文章编号:** 0372-2112 (2010) 01-0199-05

An Algorithm for Regularity Extraction in Digital Integrated Circuits Based on Fan-Like Template

PAN Wei-tao, XIE Yuan-bin, HAO Yue, SHI Jiang-yi

(Key Laboratory of Wide Band-gap Semiconductor Materials and Devices of Ministry of Education, School of Microelectronic, Xidian University, Xi'an, Shaanxi 710071, China)

Abstract: By using the method of data mining in the extraction of functional regularity in digital ICs, a novel template called FAN generation algorithm is proposed. To save the memory, a more efficient compressed storage strategy and deleting the buffer structure method are used when dealing with very large scale ICs. With the weights of edges are encoded, an algorithm which used to generating fan-like frequent subcircuits gradually is proposed, therefore, the problem of high complexity during the extraction of functional regularity in very large scale ICs has been solved effectively. Experimental results show that this fan-like template algorithm is more effective and can obtain a better circuit covering result faster than the SPOG and TREE methods.

Key words: frequent subcircuits; data mining; regularity extraction; regularity index

1 引言

数字集成电路中规则性结构的提取在电路一致性检查, 产生高度规则性的布图以及提高芯片的性能及成品率等方面起着越来越重要的作用^[1-3]. 利用规律性结构进行布局布线, 能够提高芯片的可制造性及成品率, 降低设计的难度. 但随着芯片规模的不断扩大, 采用传统提取算法进行规律性提取需要付出的代价也越来越大.

较早的电路规律性提取方法, 是从起始节点开始, 通过对电路进行分类来提取模板库中模板的同构子电路实例^[4]; 文献[5]则是利用数据通道型电路某些特性将一些特殊连线作为起始点, 如高扇出的线网, 然后通过标号法生成电路模板进行规律性提取; 而文献[2]和[6]则直接将所有相同功能的节点作为根节点, 通过扩展根节点电路生成树形和单主要输出型两种模板进行规律性提取. 上述算法一个共同特点是首先选择起始

点, 然后再逐渐加入其它顶点进行扩展生成模板. 但上述算法一味追求将所有的规则性结构全部提取, 导致算法的时间复杂度很高, 不适合对规模较大的电路进行规律性提取. 另外, 在实践中发现, 由于上述算法片面追求生成模板最大化, 导致对电路覆盖时不易选择合适的模板, 从而得到令人满意的电路覆盖. 如用 SPOG 算法对基准电路 S38584^[7]进行规律性提取时, 能够得到规模在 1000 个标准单元以上的模板有 231 个, 但每个模板都仅有两个同构子电路实例. 而这些规模较大的模板往往会影响对整个电路的覆盖. 因此, 上述算法无法有针对性的对电路进行规律性提取, 常常为得到同构子电路实例大于某个值的频繁子电路, 需要提取出所有的同构子电路实例, 造成很大浪费. 另外, 国内对集成电路规律性提取方面的研究还处于起步阶段. 文献[8]针对电路数据流图提出一种模块生成与选择算法, 但该算法采用距离矩阵的形式对顶点进行排序, 复杂度过高, 得到的模块规模也有一定的局限性, 与实际应用有一定距离; 而文

献[9]虽然是针对实际电路门级网表进行的同构子电路提取,但无法自动产生模板。

综上所述,传统规律性提取算法主要存在两个问题:(1)自动生成模板无针对性,将所有模板全部提取,导致提取时间过长;(2)由于提取出所有模板,模板数量巨大,导致选择模板对电路进行覆盖时复杂度过高,无法得到好的覆盖。本文利用数据挖掘中频繁子图挖掘方法,提出一种扇形结构模板的规律性提取算法。能够在较短的时间内实现对电路的规律性提取,并实现对电路较好的覆盖。

2 算法基础

对电路进行规律性提取时,通常将电路的门级网表转化为有向图。图中的一个顶点表示电路中一个标准单元,图中的一条边对应电路中的一条连线,图中边的方向对应电路中数据流的方向。如图1所示电路 G 即为某电路转化成的有向图。

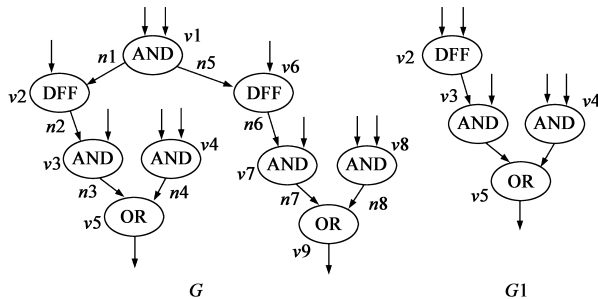


图1 示例电路

定义1 设 $G(V, E)$ 为某电路 C 的有向图表示。若图 $G'(V', E')$ 满足: $V' \subseteq V$ 及 $E' \subseteq E$,则称 G' 是 G 的子图。对于集合 $C(G) = \{C_1, C_2, \dots, C_n\}$,其中, $C_i = (V_i, E_i)$, $i = 1, 2, \dots, n$ 均为电路 C 的子电路。则称集合 $C(G)$ 是电路 C 的一个覆盖。

定义2 对于电路 C 的两个子电路 $C_m = (V_m, E_m)$ 和 $C_n = (V_n, E_n)$,若 V_m 和 V_n 之间存在一一对应的映射关系 f ,同时,按照同样的映射关系 f , E_m 和 E_n 也是一一对应,那么称电路 C 的两个子电路 C_m 和 C_n 是同构的。

定义3 设 C 代表整个电路, C 的模板指 C 中重复出现的子电路结构,而电路 C 中与此模板同构的子电路的数量称为该模板的支持度。

定义4 给定电路 C 及最小支持度阈值 $minsup$,则在 C 中,所有支持度大于等于 $minsup$ 的模板实例,称为频繁子电路。

传统算法复杂度较高的另一个原因是子图同构搜索算法比较复杂,一般的子图同构问题属于NP完全问题。本文对电路进行边权值编码,利用边权值编码可将子电路的同构问题转化为基于某种顺序遍历的边权值序列比较问题。

定义5 统计电路中所有标准单元的种类总数 r ,并给每种标准单元从1到 r 分配唯一的种类编号。规定内部连线 $E_{ab}(V_a, V_b)$,其中 V_a 和 V_b 分别是该内部连线的始端和终端标号,且二者种类编号分别为 r_a 和 r_b ,则定义该内部连线权值 W_{ab} 为:

$$W_{ab} = r_a \times 10^k + t_b, (0 < \max(t_b) \leq 10^k)$$

其中, k 为正整数, t_b 为输入边的顺序号。规定某标准单元输入边的顺序号从最左下方输入起顺时针方向从1递增编号。 k 的值取决于 t_b 的最大值。显然 W_{ab} 具有唯一性,也就是若 $W_{ab} = W_{mn}$,那么必有 V_a 和 V_m 种类序号相同,同时 $t_b = t_n$ 。如图1中电路 G ,标准单元总数为3,且所有标准单元最多输入为2,则取 k 为1。分别分配种类序号为AND为1, DFF为2, OR为3。则按照上述公式得到电路 G 中边 $n3$ 权值为11, $n4$ 权值为12。

3 扇形模板规律性提取算法

3.1 门级电路预处理

对电路进行有效的预处理是提高规律性提取效率的关键步骤。其目的主要有以下两点:一是尽量减小电路规模从而减少存储空间和处理时间;二是尽量保留原电路的有用信息从而使规律性提取结果更可信,准确性更高。首先,为了在不影响规律性提取结果的基础上尽量减小电路的规模,删除电路中不影响功能的缓冲器结构。对ISCAS89部分Benchmark电路删除缓冲器前后电路规模变化情况如表1所示。表中原电路及新电路分别指原电路及新电路中标准单元的个数。删除缓冲器结构,除能够降低电路规模外,还能排除缓冲器的影响,更充分的对同构子电路进行提取。

表1 ISCAS89部分电路删除缓冲器前后规模变化情况

Benchmark	原电路	新电路	减少(%)
S1196	547	493	9.9
S5378	2958	1850	37.5
S9234	5808	2796	51.9
S13207	8589	3939	54.1
S15850	10306	5118	50.3
S38584	20679	16486	20.3

其次,基于有向图表示的电路中标准单元间的连接关系可通过邻接矩阵表示。为降低电路存储空间,采用改进的跳维稀疏存储结构,只存储邻接矩阵中的非零元素。同时将电路的连接关系转化为每一根有效输入线网的属性,在很大程度上降低了电路的存储空间。

3.2 扇形模板提取算法

为实现对电路规律性的快速提取,可以先提取得到小规模模板,然后再逐级扩展生成较大规模模板。扇

形模板的提取算法如图 2 所示。

算法 1 扇形模板提取算法

```

输入 电路  $G$ , 整数  $m$ , 取  $minsup$  为  $t$ 
输出 电路中所有满足条件的扇形频繁子电路
1:  $n = 1$ //统计电路中每种标准单元出现的频率. 若出现频率小于  $t$ ,
   则删除;对所有出现频率大于等于  $t$  的标准单元进行标记. 计算所有
   顶点的有效输入权值.
2: do
3:   $n = n + 1$ 
4:  for 每个  $n - 1$  元扇形模板  $F_{n-1}$ 
5:     $Subcircuit\_Mining(F_{n-1})$ //搜索  $F_{n-1}$  所有同构实例
6:    for 每个  $F_{n-1}$  的  $n - 1$  元同构实例  $S_{n-1}$ 
7:       $S_n \leftarrow S_{n-1} + e$ //在最左顶点处扩展一条边,  $WFS$ 
8:    end for
9:  end for
10:   $F_n = Sort(S_n)$ //统计  $n$  元扇形子电路的种类及出现频率, 若
   某种  $n$  元扇形子电路出现频率大于等于  $t$ , 标记为  $n$  元扇形频繁
   子电路;删除  $n$  元扇形非频繁子电路
11: while  $F_n \neq \Phi \ \& \ n! = m$ 
12: Result =  $\cup F_n$ 

```

图 2 扇形模板提取算法

扇形模板的提取算法是一个逐级迭代算法,每次迭代都会增加一条边.结合边权值编码,给定扇形模板规模的上限 m ,取最小支持度阈值 $minsup$ 为 t ,则利用算法 1 就能搜索出所有规模不大于 m 的扇形频繁子电路.为便于区分,一个扇形子电路 C_n 对应一个节点链表 S_n 及相应的特征链表 F_n .节点链表 S_n 用来表示扇形子电路顶点标号,根节点标号为链表的第一项.如图 1 边 n_2 的节点链表为: $\{v_3, v_2\}$.特征链表 F_n 用来表示扇形子电路内部连线特征,根节点的种类编号为链表的第一项,其余各项为对应边的权值.如图 1 边 n_2 的特征链表表示为: $\{1, 21\}$.算法 1 子函数 $Subcircuit_Mining(F_{n-1})$ 表示在所有特征链中搜索与特征链 F_{n-1} 对应项完全相同的同构链实例.所有特征链对应项完全相同的链称为同构链.算法 1 子函数 $Sort(S_n)$ 表示对电路中所有节点链按照其对应的特征链进行分类,从而确定各种特征链出现的频率.

扩展扇形子电路时,采用宽度优先搜索扩展产生一个有唯一顺序的链表.为进行宽度优先搜索,链表中的每个顶点均有一个级别号和一个输入序号,其中输入序号与边权值公式中对应边的顺序号相同.进行宽度优先搜索的方向为逆数据流方向.电路中顶点的先后顺序采用标号方式进行区分.如 $\{v_0, v_1, v_2, \dots\}$ 中,若 $a < b$,则表示 va 级数小于 vb ;或二者级数相同,但 va 的输入序号小于 vb .我们称 v_0 为根节点,需要下次进行扩展的顶点为最左顶点.如图 1 电路 G_1 所示,若 v_5 为根节点,且其级数为 0,输入序号为 0(规定根节点

的级数和输入序号均为 0);则 v_3 的级数为 1,输入序号为 1; v_4 的级数为 1,输入序号为 2; v_2 的级数为 3,输入序号为 1.因此,电路 G_1 仅有一种顶点序列为 $\{v_5, v_3, v_4, v_2\}$.每次扩展时,在级数最小的未被扩展完全的顶点中选择输入序号最小的顶点作为最左顶点.对最左顶点进行扩展时,选择未扩展的输入中输入序号最小的输入边作为需要扩展的边.

如上所述,每个扇形子电路分别有一个用来记录顶点的节点链 $\{v_0, v_1, v_2, \dots\}$ 及一个用来描述对应边的特征链 $\{e_0, e_1, e_2, \dots\}$.如算法 1 的第四步到第九步,第一个循环用来发现所有两个顶点的频繁子电路 $v_1 \xrightarrow{e_1} v_0$.第二轮循环则在第一轮频繁子电路的基础上发现包含 $v_2 \xrightarrow{e_2} v_1$ 结构的频繁子电路.如此循环直至搜索到所有满足要求的频繁子电路.当所有扇形子电路的支持度小于 $minsup$ 或扇形子电路的规模达到 m 时,循环结束.同时,每次循环过程中,只有支持度大于或等于 $minsup$ 的频繁子电路才可以被选作模板且被保留参与下一轮的扩展.而支持度小于 $minsup$ 的非频繁子电路则被删除.如图 1 电路 G 所示,取 $minsup$ 为 2, m 为 3,则初始化之后得到所有标准单元全部为规模为 1 的频繁子电路.第一轮循环后得到 n_2 与 n_6, n_3 与 n_7, n_4 与 n_8 为三种规模为 2 的频繁子电路.第二轮循环后,由于宽度优先搜索得到的链表具有唯一性,故仅得到一种规模为 3 的频繁子电路,且其节点链分别为 $\{v_5, v_3, v_4\}$ 及 $\{v_9, v_7, v_8\}$,特征链均为 $\{3, 11, 12\}$.第三轮循环时发现规模为 3,则循环终止.故得到的扇形频繁子电路为 $\{v_5, v_3, v_4\}$ 及 $\{v_9, v_7, v_8\}$.

3.3 模板覆盖

根据算法 1 产生的模板,从中选取满足一定要求的模板对整个电路进行覆盖.覆盖时被选中的模板依次对整个电路进行覆盖.对电路中被覆盖到的节点进行标记,已标记过的节点不再参与下一个模板的覆盖,直至所有模板全部完成覆盖为止.覆盖率表示模板对整个电路覆盖的情况.覆盖率的大小取决于用来进行覆盖的模板的选择规则.传统方法有两种,一种是 LFF (Largest Fit First, 最大模板优先).将产生的模板按照规模从大到小进行排序,每次都选取剩余模板中规模最大的模板对电路进行覆盖.另一种是 MFF (Most frequent Fit First, 最频繁模板优先).将产生的模板按照支持度从大到小进行排序,每次都选取剩余模板中支持度最大的模板对电路进行覆盖.

传统算法^[2,4-6]由于算法自身限制,无法灵活判断得到的模板是否为最优模板.而算法 1 则可以提取到能够产生模板的电路结构中最为核心的部分,更有利于提高模板对整个电路的覆盖率.另外,算法 1 每一步都

能选择最频繁的节点进行规律性提取,使得进行规律性提取时有的放矢,避免了传统算法中由于盲目提取而造成的大量不必要操作,能够最快的提取到支持度最高且规模适当的模板并直接得到其相应的所有同构子电路实例.在利用算法1得到的模板进行电路覆盖时,仅用查看相应模板实例中是否存在已经被覆盖过的节点即可,避免了传统算法进行覆盖时需要再对整个电路进行一遍同构子电路搜索,简化了流程,可显著提高模板覆盖的效率.

4 实验结果与分析

实验所用的硬件配置为 P IV2.66 GHz, 1G 内存. 实验数据来自 ISCAS89 部分 Benchmark 电路. 分别采用 SPOG、TREE 及本文算法(FAN)进行规律性提取. 实验结果如表2和表3所示. 表2为按 LFF 原则对电路进行覆

盖得到的结果,表3为按 MFF 原则对电路进行覆盖得到的结果. 表2和表3中第一列为目标电路名称及其规模. 电路规律性提取结果采用覆盖率 CI (Covering Index) 和规则性系数 RI (Regularity Index) 来表示^[2,6]. 其中 CI 指被覆盖到的电路面积占整个芯片面积的比例,表示该电路规律性提取的程度; RI 表示所有模板面积之和占芯片总面积的比例. RI 的大小表示现有电路可以利用所提取的模板,进行再综合或布局布线等流程可得到优化程度的大小. 若 RI 较大,表示可优化的程度较大;若 RI 较小,则表示该电路可优化的余地较小^[6]. $M(U)$ 中 M 表示进行覆盖时所用到的模板的个数, U 表示所有模板的实例总数; T 表示规律性提取时间,单位为 s ; $G(R)$ 中 G 表示模板的最小规模阈值, R 表示选作用来进行覆盖的模板的最小支持度阈值.

表2 最大模板优先规律性提取结果

BECHMARK	TREE				SPOG				FAN				
	CI (%)	RI (%)	$M(U)$	T (s)	CI (%)	RI (%)	$M(U)$	T (s)	CI (%)	RI (%)	$M(U)$	T (s)	$G(R)$
S1196(493)	44.1	10.8	18(81)	0.625	51.5	14.8	27(97)	1	60.5	14.3	26(117)	0.3	2(3)
S5378(1850)	56.3	3.3	21(337)	15.7	52.1	4.3	22(257)	35.7	80.2	8.5	43(407)	2.0	2(10)
S9234(2796)	55.6	2.4	17(434)	54.9	61.3	5.0	34(445)	127.8	75.1	7.3	47(516)	3.3	2(10)
S13207(3939)	64.9	2.5	26(728)	123.2	56.0	4.2	38(493)	302.3	79.6	7.0	65(765)	8.2	2(10)
S15850(5118)	49.2	1.4	18(807)	341.5	60.1	3.4	41(723)	693.2	80.1	4.8	60(1055)	16	2(15)

表3 最大支持度优先规律性提取结果

BECHMARK	TREE				SPOG				FAN				
	CI (%)	RI (%)	$M(U)$	T (s)	CI (%)	RI (%)	$M(U)$	T (s)	CI (%)	RI (%)	$M(U)$	T (s)	$G(R)$
S1196(493)	45.6	5.9	13(103)	0.609	55.4	8.2	18(144)	0.859	53.4	6.0	15(141)	0.3	2(3)
S5378(1850)	42.5	1.2	11(415)	16.1	57.3	2.4	17(538)	35.7	67.8	1.8	17(603)	2.0	2(10)
S9234(2796)	50.4	0.6	9(820)	55.9	63.6	1.0	12(985)	127.4	65.4	1.7	22(774)	3.3	2(10)
S13207(3939)	42.3	0.9	14(881)	122.6	65.6	0.8	16(1106)	302.3	67.7	1.2	22(1120)	8.1	2(10)
S15850(5118)	55.3	0.6	13(1628)	343.3	63.7	0.7	15(1828)	686.6	62.5	1.1	25(1455)	16	2(15)

由表2和表3可知,本文提出的扇型模板可以在较短的时间内得到更高的规则性覆盖率和规则性系数. 图3和图4是对电路 S5378 改变 $minsup$ 的情况下,采用 LFF 原则,用三种不同方法得到的 CI 和 RI 结果. 由图可知,本文提出的 FAN 算法在 $minsup$ 改变的情况下与

传统算法相比仍具有很大优势. 图5为三种不同算法在不同阈值情况下的运行时间比较. 由图可知, FAN 运算时间要小于其它两种传统算法. 采用算法1得到的扇形模板对电路进行覆盖具有很好的灵活性. 若某个较大的模板无法对电路进行覆盖,则可以逐渐降低该模板

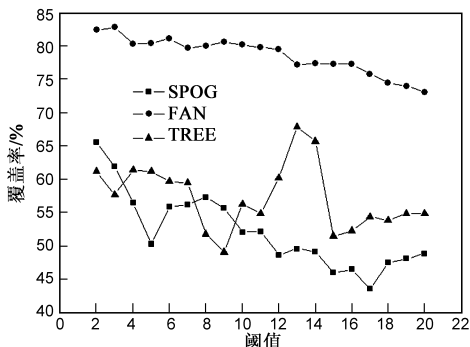


图3 规则性覆盖率(CI)随阈值变化曲线

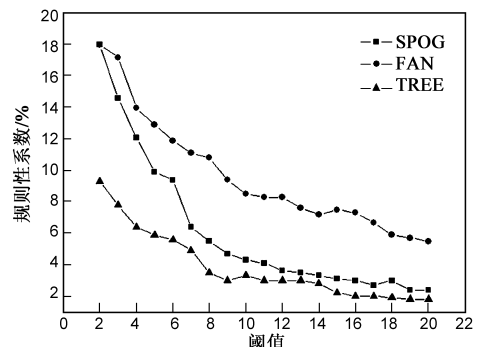


图4 规则性系数(RI)随阈值变化曲线

的规模来对电路进行覆盖,从而选取模板最适合的大小对电路进行覆盖.传统算法得到模板的层次化程度较小,对电路进行覆盖时灵活性较差,故得到的覆盖率也较低.

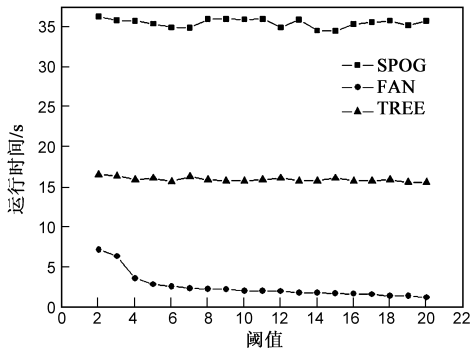


图5 三种不同算法运算时间随阈值的变化情况

另外, FAN 算法所提取出的扇形频繁子电路,以 EDIF200 文件格式输出,实现与 Cadence 等主流 EDA 工具无缝衔接,使得该算法具有良好的实用性,便于对所得到的规律性提取结果分析理解.同时,该算法用 C 语言实现,具有较强的平台通用性和移植性,可以在 Microsoft Windows, Linux, SUN Solaris 等平台上运行.

5 结论

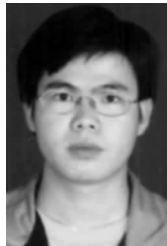
将数据挖掘思想应用于集成电路的规律性提取中,提出一种基于扇形模板的规律性提取算法.利用边权值编码,有效解决了传统算法根节点规模增加而导致运算时间急剧增加的问题,降低了算法复杂度.另外,由于采用逐级扩大规模的方法,得到层次化较强的模板,可以对电路进行更好的覆盖,突破了传统算法一味追求模板最大化而带来的局限性.

参考文献:

- [1] V Kheterpal, V Rovner, et al. Design methodology for IC manufacturability based on regular logic bricks[A]. In Proc. Design Automation Conf[C]. Anaheim: IEEE Press, 2005. 353 - 358.
- [2] A P E Rosiello, F Ferrandi, D Pandini, D Sciuto. A hash-based approach for functional regularity extraction during logic synthesis[A]. IEEE Computer Society Annual Symposium on VLSI[C]. New York: IEEE Press, 2007. 92 - 97.
- [3] Y Feng, H A Mantooth. Algorithms for automatic model topology formulation[J]. IEEE Trans on Computer-Aided Design, 2009, 28(4): 502 - 515.

- [4] D S Rao, F J Kurdahi. On clustering for maximal regularity extraction[J]. IEEE Trans on Computer-Aided Design, 1993, 12(8): 1198 - 1208.
- [5] T Kutzschebauch. Efficient logic optimization using regularity extraction[A]. In Proc. Intl. Conf. on Computer Design[C]. Austin: IEEE Press, 2000. 487 - 493.
- [6] A Chowdhary, S Kale, et al. Extraction of functional regularity in datapath circuits[J]. IEEE Trans on Computer-Aided Design, 1999, 18(9): 1279 - 1296.
- [7] Maksim Jenihhin. ISCAS' 89[OL]. <http://www.pld.ttu.ee/~maksim/benchmarks/iscas89/>, 2007.
- [8] 郎荣玲, 戴冠中. 集成电路的模块生成与选择算法[J]. 电子学报, 2005, 33(11): 1955 - 1958.
Lang Rongling, Dai Guanzhong. Algorithms for template generation and selection in integrated circuits[J]. Acta Electronic Sinica, 2005, 33(11): 1955 - 1958. (in Chinese)
- [9] 李长青, 汪雪林, 等. 辐射路匹配: 从门级到功能模块级的子电路提取算法[J]. 计算机辅助设计与图形学学报, 2006, 18(9): 1377 - 1382.
Li Changqing, Wang Xuelin, et al. Radiate path matching: a subcircuit extraction algorithm from gate level to functional level[J]. Journal of Computer-Aided Design & Computer Graphics, 2006, 18(9): 1377 - 1382. (in Chinese)

作者简介:



潘伟涛 男, 1981 年 11 月出生于河南漯河. 西安电子科技大学微电子学院博士研究生. 研究方向为 SoC 设计方法学及 VLSI 设计自动化等.

E-mail: wtpan@mail.xidian.edu.cn



谢元斌 男, 1982 年 9 月出生于江西德兴. 西安电子科技大学微电子学院博士研究生. 研究方向为 VLSI 设计及可测性研究等.