第5卷第2期
1994年2月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 5, No. 2
Feb. 1994

用面向对象方法实现VLSI布图规划*

杨芙清 吴良芝 洪建顺

(北京大学计算机科学技术系,北京 100871)

摘要 在集成化软件工程支撑环境"青鸟"系统**上,基于模拟退火实现了VLSI的布图规划算法,算法能够处理矩形和L型模块.使用面向对象方法开发软件,显著缩短了程序的设计过程,利用青鸟系统支持开发,明显地提高了系统的开发效率.

关键词 超大规模集成电路,布图规划*,软件工程,面向对象.

布图规划(floorplan)设计[5]是VLSI电路布图设计的第一步,它是传统模块布局的推广,该问题可描述如下:给定一模块集及模块间的互连关系,将给定的电路模块放置于平面上以最小化两个目标函数的加权和:(1)包含所有模块的矩形区面积;(2)总的互连金属总线长的估计值.大多数已有的布图规划算法[1,2]都假定模块是矩形的.在实际应用中,一些模块是预设计的库宏单元,这些库宏单元不一定是矩形的,而是L型的(如图1).为了使用这些算法,通常必须用包围L型模块的矩形来代替它,这就导致芯片面积比实际需要大.即使在所有模块都是矩形的情况下,许多算法产生的布图规划是slicing结构的.通常,对slicing结构,在所有模块都很软(flexible)(大小和形状可改变)的情况下能够产生合理的好的结果.但在模块的重要部分是很硬(大小和形状不可改变)的情况下,slicing结构将引进不必要的死区(dead space).而允许产生non-slicing结构的算法可以充分利用芯片面积,并可达到减少死区面积的目的.

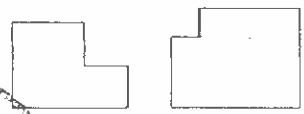

图1 L型模块

面向对象设计方法的基本特征是信息隐蔽(information hiding)、数据抽象(data abstraction)、类继承(inheritance)和动态连接(dynamic binding)及多态性(polymorphism).面向对象设计方法使得设计者可以把精力集中在系统设计上而不用担心系统中所使用的对象的细节,对象之间是通过消息(message)来联系的.它能够很好地解决软件可靠性、可修改性、可理解性和生产率的问题.同时为软件重用性打下了基础.使得新的软件开发可以利用

* 本文1991年9月9日收到,1991年12月29日定稿

** 集成化软件工程环境(青鸟系统)是国家"七五"重点攻关项目,由北京大学、复旦大学、北京航天航空大学、北京信息工程学院、中软总公司联合研制.

作者杨芙清,女,60岁,教授,主要研究领域为系统软件、软件工程、软件工程环境、智能化软件工程环境.吴良芝,57岁,副教授,主要研究领域为CAD.洪建顺,29岁,助教,主要研究领域为IC、CAD.

本文通讯联系人:杨芙清,北京 100871,北京大学计算机科学技术系

先前开发的软件从而加快了程序设计过程. 面向对象设计方法已在 VLSI 中的电路模拟中得到了应用[7].

基于以上原因,本文研究用面向对象方法来实现 VLSI 布图规划算法. 文中采用一种新的方法——波兰表达式来表示布图规划[5]. 本算法能够处理矩形和 L 型模块并使用模拟退火(SA)方法[8]来搜索最优布图规划. 通常本算法产生的布图规划是 non－slicing 结构. 在需要时也能产生 slicing 结构. 在以下几节中我们将描述算法的要点及其在面向对象环境中的实现,并给出了实验结果.

1　算法实现

1.1　类(class)的分层结构

应用面向对象设计方法,我们设计了类的分层结构,类的分层结构如图 2 所示.

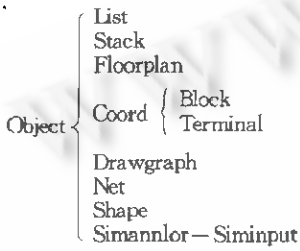

图2　类分层

图 2 的类分层中,子类继承了父类的属性与操作. 所有类都继承了其基类 Object 的属性与操作. Object 类抽象了其它类的共同属性与操作.

类 Stack 描述了栈的性质,如进栈、出栈等. 由它可产生整数栈、字符栈等. 类 List 描述了链表的性质,如插入、删除等. 类 Floorplan 描述了芯片的性质及其上的操作,如 SA 算法中的状态产生函数(即布局变换),面积计算和实例化布图规划等操作. 类 Coord 描述了坐标的属性及其上的操作. 类 Block 描述了模块的所有共同属性及对这些属性的操作. 类 Terminal 则描述了 I/O 端口的所有属性及其上的操作. 类 Net 描述了线网的共同属性及其上的操作. 类 Shape 描述了模块几何形状的属性及其上的操作. 类 Simannlor 控制整个模拟退火过程. 类 Drawgraph 描述了 SunView 窗口的属性及对窗口的操作,类 Siminput 对输入数据进行语法检查,并建立程序内部使用的数据结构(主要是模块链表和线网链表).

1.2　L 型模块的处理

设一个模块由形状、面积和朝向(orientation)三个属性描述,模块的每一种形状、面积和朝向选择称为模块的实例.

设 Ω 为任意大小和朝向的矩形和 L 型几何图形的集合. 对每个模块 i,记 Gi⊂Ω,表示此模块所有可能实例的集合. 设 A 为 Ω 中的几何图形,若 A 为矩形、则其朝向索引定义为 0,若 A 为 L 型模块,则被定义为 1、2、3 或 4. 如图 3 所示.

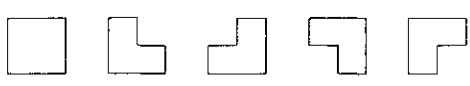

图3　方位索引号

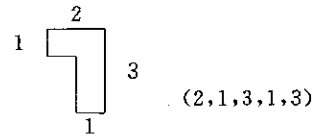

图4　几何图形的表示

显然,A的面积可由其四条最外边的边界确定.设x_1和x_2,$x_1 \geqslant x_2$,为两条最外水平边的长度,y_1和y_2,$y_1 \geqslant y_2$为两条最外垂直边的长度,s为A的朝向索引,则A可由五元组(x_1, x_2, y_1, y_2, s)表示,图4所示的五元组为(2,1,3,1,3).为了用波兰表达式表示布图规划及处理L型模块.首先定义矩形模块和L型模块的操作.这种操作能够将矩形和L型几何图形组合成更大的矩形和L型几何图形.在此定义了一个一元操作符"$\rightharpoonup$"和四个二元操作符"$*_1$"、"$*_2$"、"$+_1$"、"$+_2$"对Ω中的几何图形进行操作[5].对于所有$A \in \Omega$,$\rightharpoonup A$定义为包含A的最小矩形.称$\rightharpoonup A$为A的补.更精确地说,有$\rightharpoonup(x_1, x_2, y_1, y_2, s) = (x_1, x_2, y_1, y_2, 0)$,如图5所示.一元操作符提供了将L型几何图形置于矩形区的可能性.

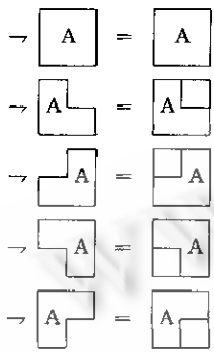

图5　一元运算符

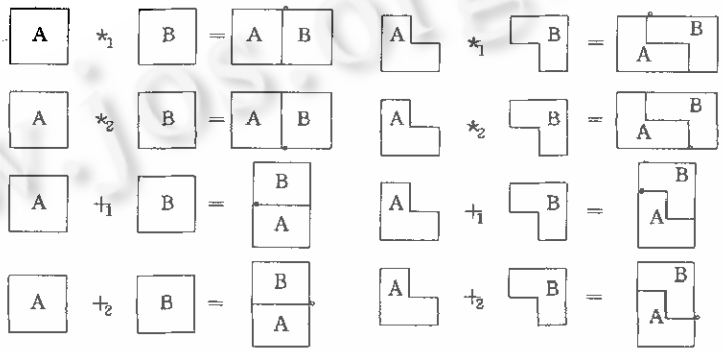

图6　二元运算符

四个二元操作符,定义了将Ω中的两个几何图形尽可能紧凑地叠放在一起,构成Ω中更大的几何图形的方法.设A、B$\in \Omega$,则$A *_1 B$和$A *_2 B$为将A和B水平并排放置且B放在A的右边的几何图形.$A +_1 B$和$A +_2 B$为将A和B垂直叠放且B放在A的上部的几何图形.新的几何图形的朝向索引和面积由两个几何图形的朝向和面积决定且尽可能紧凑压缩在一起.这样,每一个二元操作符可由其两个操作数的25种朝向组合操作描述.所以共有100种可能的操作.图6表示了其中的八种.(图6中。相应于产生几何图形的三种可能.图7给出了这个标记.图8为一个例子).

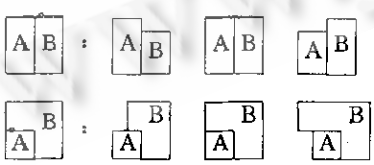

图7　圆圈标记

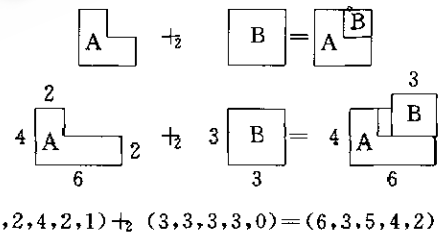

图8　二元运算符的例子

关于波兰表达式与布图规划之间一一对应的说明可见[5].

1.3　算法

设α为波兰表达式,f_α表示由α产生的布图,设$A(\alpha)$为f_α的面积,若(x_1, x_2, y_1, y_2, s)

$\in\Omega$是包含 f_a 的最小 L 型图形，则 $A(\alpha)=x_1\cdot y_1$（这里，$A(\alpha)$定义为包含(x_1,x_2,y_1,y_2,s)的最小矩形的面积）. $W(\alpha)$为 f_a 的布线总长度的估计值这里线网长度估计值采用半周长法[3]. 代价函数为 $cost(\alpha)=A(\alpha)+\lambda\cdot W(\alpha)$. 其中 λ 为一常数，用于控制面积和布线长度的相关重要性.

设 E 为代数系统$(\Omega,*_1,*_2,+_1,+_2,\rightarrow)$代数表达式的集合. E_p 为相应于 E 中表达式的波兰表达式的集合. 我们的布图规划算法使用模拟退火算法搜索 E_p 中的波兰表达式 α 使得 $cost(\alpha)$最小化. 设 $\alpha\in E_p$ 为一波兰表达式，显然 α 具有如下形式：$\alpha_1\beta_1\alpha_2\beta_2\cdots\alpha_{2n-1}\beta_{2n-1}$，其中 n 个 α_i 为 n 个模块的几何图形，另外 n−1 个 α_i 为二元操作符，每个 β 或者为一元操作符"→"或者为空字符串"ε". 现定义四种变换(move)来修改 α. 这四种变换定义如下：M_1：对某个 i 修改 α_i，若 α_i 为操作数，则存在 K，使得 $\alpha_i=A\in G_k$，选择 $A'\in G_k$ 且 $A'\neq A$ 置 $\alpha_i\leftarrow A'$，这相应于选择模块另一个实例；若 α_i 为二元操作符，则选择一个不同于 α_i 的另一个二元操作符. M_2：将 β_i 改为集合中$\{\varepsilon,\rightarrow\}$的另一元素. M_3：交换两个操作数 α_i 和 α_j，这相应于交换两个模块. M_4：对某个 i，交换 α_i 和 α_{i+1}，其中之一为操作数另一个为二元操作符. 注意 M_1，M_2，M_3 变换总是产生在 E_p 中的波兰表达式. 而 M_4 变换可能产生非波兰表达式. 我们只选用产生波兰表达式的 M_4 变换. 这四种类型的变换能有效地保证通过一系列变换将一波兰表达式转换为另一波兰表达式.

我们使用的温度调度形式为 $T_{k+1}=\gamma\cdot T_k$，这里 γ 一般取为 0.8−0.99 之间，在当前实现中，我们的终止条件为：(1)在某一温度下，可接受的变换很少；(2)温度很低.

2 结 果

我们在集成化软件工程支撑环境"青鸟"系统上(运行于 SUN 3/160)用面向对象语言实现了布图规划算法. 使用的语言为青鸟系统上的 OTDL(对象类型描述语言)(用于描述类)和 OECL(对象执行控制语言)(其功能是用于建立对象，通过一组互相影响对象来完成定义工具的任务). 如图 9 所示为我们的结果. 表一给出了 non−slicing 结构的布图规划与 slicing 结构的布图规划结果的比较. A_S 为由算法产生的 slicing 结构的布图规划面积，A_L 为由算法产生的 non−slicing 结构的布图规划的面积. A_L/A_S 值愈小表示 non−slicing 结构愈有利. 从这些测试例子中，我们可看出引进 L 型模块和 non−slicing 结构后能充分利用芯片的面积.

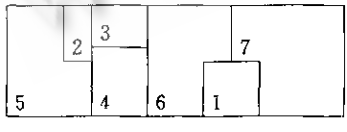

图9

表 1

问题	N	M	A_L/A_S	CPU
P1	7	3	0.774	830.6
P2	10	4	0.923	3107.7
P3	15	6	0.854	6727.2

注：N——模块数　M——L 型模块数　CPU 时间单位为秒

致谢　在论文写作过程中得到了吉利久副教授帮助. 此外，还得到了陈钟、苏渭珍等同志的

帮助.在此向他们表示衷心的感谢.

参考文献

1 Kazuhiro Ueda. CHAMP:chip floorplan for hierarechical VLSI layout design. IEEE Trans. on CAD, 1985;CAD-4(1):12-22.

2 David P. MASON:a global floorplanning approach for VLSI design. IEEE Trans. on CAD, 1986;CAD-5(4):477-489.

3 Satoshi Goto, T Matsuda. Chapter 2:patition,assignment & placement. Layout Design & Verification, 1986.

4 Wang D F. A new algorithm for floorplan design. 23rd DAC, 1986:101-107.

5 Wang D F. Floorplan design for rectangulr and L-shaped modules. Proc. ICCAD/IEEE, 1987.

6 Geoffrey A Pascoe. Elements of object-oriented programming. BYTE, Aug. 1986:135-134.

7 Jeff A Watts. ROOMS:a relaxation-based, object-oriented. Mixed-Mode Simulator. IEEE 1990 Custom Integrated Circuits Conference.

8 姚新.模拟退火及其应用.计算机研究与发展,1990;27(7):1.

9 吕天珍等.面向对象的设计方法.计算机科学,1987(4):18.

IMPLEMENT FLOORPLAN FOR VLSI WITH OBJECT-ORIENTED METHODS

Yang Fuqing, Wu Liangzhi and Hong Jianshun

(*Department of Computer Science and Technology, Peking University, Beijing* 100871)

Abstract　:In this paper, based on simulated annealing approch, the authors first implement floorplan algorithm for VLSI using object-oriented method on the "Jade Birds" system of integrated software engineering supporting environment. This algorithm can process rectangular and L-shaped modules. Programming process can be cut short obviously when object-oriented method of software development is applied . It raises developing effectiveness of system obviously to use "Jade Birds" system supporting development work.

Key words　VLSI, floorplan*, software engineering, object-oriented.