智能 CAD 工具*

杨乔林

(中国科学院计算技术研究所)

AI CAD TOOLS

Yang Qiaolin

(Institute of Computing Technology, Academia Sinica)

ABSTRACT

This paper describes some AI CAD tools: a deductive logic CAD design data base, a layout language which includes an inference engine and a deductive layout data base, and a circuit recognition program which can recognize basic logic cells from transistor nets and complex logic circuits from basic logic nets.

摘 要

本文介绍我们开发的运用人工智能技术的 CAD 工具：演绎 CAD 数据库；具有推理能力并内含演绎版图数据库的版图语言；能从晶体管网络中识别基本逻辑单元和从基本逻辑单元网络中识别复杂逻辑电路的电路识别工具。

§1. 引 言

随着科学技术的发展，设计自动化和计算机辅助设计，已取得很大成功，已深深地渗透到各行各业之中．有些行业，如微电子工业，离开了这些技术，简直寸步难行．但是基于算法的传统 CAD 技术，其一些基本的弱点——缺乏人类专家所具有的，基于知识、经验和事实的判断推理能力，并没有发生根本性的改变．在设计关键之处，需要人类专家进行判断和做出决定，在某些设计环节(如设计层次之间的转换——从系统级到逻辑级的转换)还没有太多的作为．

* 1989年8月25日收到．

八十年代开始，在人工智能技术发展的推动下，人们希望利用人工智能来弥补传统 CAD 的不足.〔1〕是一个基于规则的布线实验系统；〔2〕是一个基于知识的硬件设计实验系统；〔3〕是利用 Prolog 语言的逻辑综合实验系统；〔4〕是基于知识和算法的 IC 版图自动生成试验系统.

KBAMGE 是我们开发的一个基于知识和规则的电路模块生成环境[5]，它从符号布尔表达式、有限状态机或真值表的描述出发，经过其各种工具的推演、化简和变换，最后得到相应的 CMOS 电路和版图.

在 KBAMGE 基础上，为了进一步探求人工智能原理和技术在 CAD 技术中的运用，我们分析和考察了传统 CAD 数据库的一些缺陷(如数据冗余、缺乏推理能力等)，建立了一种演绎型 CAD 数据库的模型——PRODB.

版图语言是 IC CAD 中的重要工具，有低级语言(如 GPSⅡ，相当于机器代码指令)、中级语言(相当于汇编级)和高级语言之分.〔6〕、〔7〕是与 C 和 PASCAL 相类似的版图语言；〔8〕是我们开发的嵌于 PASCAL 的版图语言，而这些都属于过程式的版图语言，并且没有推理能力，也难于将知识和规则融合到版图的描述之中.为了克服这些缺点，我们开发了一种具有推理能力的高级版图语言 PROLAY.

此外，本文还介绍从晶体管网络中识别基本逻辑单元电路和从基本逻辑单元电路网络中认别复杂逻辑电路的系统——PRORECOG.

§2. 演绎 CAD 数据库

传统的 CAD 数据库技术通常使用层次模型、网络模型或关系模型，由于没有推理演绎的功能，不能利用数据库的传递或继承关系，获取缺省的数据，也就是说不能充分利用数据共享的功能.本文介绍一个演绎 CAD 数据库 PRODB，它是使用框架网络模型，适宜于电路各种知识的表达，另外由于具有推理演绎的功能，可以利用所属关系和同类关系进行数据传递和继承，因而具有较好的数据共享功能，也就减少了系统数据的冗余量.

在电子领域 CAD 数据库中，往往需要存储如下几方面的内容:

※电路的所属和类型

※电路功能的符号布尔表达式

※电路功能的有限状态机描述

※电路的真值表

※电路的框图表示

※电路的结构表示

※电路的 I/O 引线

※电路版图描述

※电路构成的晶体管网络

※电路特征和参数，如芯片面积、延迟、功耗、电平等.

在演绎 CAD 数据库 PRODB 中，每种电路的各种知识和特性，我们使用框架 view 来描述:

view(cktname, [ako, is_a, expr, fsm, truth, schem,

struc，iop，layout，transistor，delay，
size，fanout，power，path〕）

这里 ako，is_a，expr 等分别是框架的槽项，其值用以表示和刻划电路的知识和特性。例如：

is_a(ff_{11}，rs_ff).表示电路 ff_{11} 是一种 rs_ff 电路。

expr (ckts，"$x_1 * x_2+x_3 * x_4$")表示电路 ckts 的符号布尔表达式是"$x_1 * x_2+x_3 * x_4$"。而表示 I / Oprot 的是：

iop(ckts，〔inputlist〕，〔outputlist〕，〔statelist〕，〔biderectlist〕）

fsm(ckts，〔fsm(　)，fsm(　)…〕）是有限状态机的描述；而电路的内部结构为：

struc (ckts，〔ckt(ckt$_1$，〔in$_1$〕，〔out$_1$〕，〔state$_1$〕，〔bi_derict$_1$〕），
ckt(ckt$_2$，〔in$_2$〕，〔out$_2$〕，〔state$_2$〕，〔bi_derict$_2$〕)〕）

与一般的 CAD 数据库相比较，PRODB 除了具有查询、检索和数据库管理功能外，它最大的特点是具有推理演绎的功能，利用 is_a，ako 实施对象之间数据的传递与继承，因而，减少了数据的冗余量。

对于 PRODB 的推理演绎过程，最好是以例子来说明。例如，我们以查询某一电路 X_ckt 的芯片面积为例，考虑它的推理查询过程。

$$find\_size(X\_ckt,\ Size):\ -size(X\_ckt,\ Dimens,\ Unit),\ dimens(Dimens,\ Unit,\ Size),\ !\qquad (1)$$

$$find\_size(X\_ckt,\ Size):\ -is\_a(X\_ckt,\ Ckt),\ find\_size(Ckt,\ Size),\ !\qquad (2)$$

$$find\_size(X\_ckt,\ Size):\ -calc\_size(X\_ckt,\ Size),\ !\qquad (3)$$

在这个面积的演绎查询过程中，可以分为三种情况。(1) 是直接查询，通过检索电路 X_ckt 的 size 槽的值，通过 dimens_size 的变换，检索出电路芯片的面积。在 size 槽值不存在时，通过 is_a 关系子网络，在同类电路中，检索芯片面积尺寸，这就是(2)所完成的任务。is_a(x，y)网络可能有若干个节点，所以(2)的推理也是一个复杂递归过程。

在(1)、(2)都失败时，推理机转向另一个推理查询过程(3)——计算芯片的面积尺寸。calc_size 是另一个复杂的推理演绎过程，它首先通过直接查询 X_crt 的布尔表达式，利用表达式-版图自动变换谓词(参看〔5〕)，找出该表达式所相应的版图，然后利用版图边界导出谓词 find_border，找出它的边界，然后求出相应的芯片面积。

在非表达式描述的情况下，calc_size 要通过查询该电路 X_ckt 的结构组成 struc，通过其构成子电路，计算其芯片面积大小。

其实，各种推理演绎过程都是相类似的。例如对于 CMOS 门阵列线路，有时需要查询电路 ckta，带上电路 cktx 的 netx 后其时间延迟的变化，那么除了查询 ckta 的单位扇出(fanout)时间延迟-△td 外，还需查询 cktx 的 netx 的负载量(即 fanout 数)，二者的乘积，便是所需求的时间延迟的变化量。fanout 数的推理查询过程如下所示：

find_fanout(*Cktx*, *Netx*, *Fanout*):－*fanout*(*Cktx*, *Netx*, *Fanout*), !.　　(4)

find_fanout(*Cktx*, *Next*, *Fanout*):－*struc*(*Cktx*, *Ckt*),
　　find_struc_fanout(*Ckt*, *Next*, *Fanout*),
　　db_assert(*fanout*(*Cktx*, *Netx*, *Fanout*)), !.　　(5)

这里(4)是对 fanout 直接查询．当数据库中无该项数据时，要通过 find_struc_fanout 查询，对于与 next 相联结的电路，还需递归地调用(4)，(5)所示的 find_fanout 谓词，(5)中演绎查询成功时，利用 db_assert 将查询结果存入数据库，在以后再次进行相同的查询时，直接检索即可．

此外，PRODB 还提供其他特殊的演绎查询，如寻求最长逻辑路径延迟 path_td 等等．

PRODB 使用 pop_up 菜单作为用户与系统之间的界面，对于查询功能相对固定的 CAD 数据库来说，pop_up 菜单是十分适用的．其主菜单与各种子菜单如下所示．

File	Load	Consult	Circuit	Ako
		List	Fanout	Is_a
Dbms	Edit	Save	Td	Consist
		Delete	Dtd	Schem
Quit	Save	Add_in	Load_td	Expr
		Retriev	Size	Iop
	Quit		Power	Fsm
			Path_td	Layout
				Transiat
(a)	(b)	(c)	(d)	(e)

(a) 主菜单
(b) File 子菜单
(c) Dbms 子菜单
(d) Retrieve 子菜单
(e) Circuit 子菜单

§3. 推理型版图语言

象其他语言那样，版图语言也有其低、中、高级之分．低级的版图语言也就是显示设备或制版设备的机器指令，是非常难读的冗长的数码串．中级版图语言，也就是一些中间格式语言，如 CIF 那样，命令段使用 ASCII 码，而坐标段使用数字．象高级算法语言那样，高级版图语言[6][7][8]，在命令段、坐标数据段，都可以使用变量，因此可以进行参

数化，相对位置的版图设计，许多硅编译系统的单元模块自动生成，都是使用这类高级版图设计语言。

前述的这类基于过程式语言的版图语言，不象描述式语言那样使用方便，更重要的是缺乏推理演绎的功能，也很难将知识和规则融合到版图设计中去。因此，我们在 KB／L〔5〕的基础上，增加许多新的谓词，使其成为一种具有推理演绎的版图语言 PROLAY。下面是它的一些主要特点：

1. PROLAY 内含一个演绎版图数据库。
2. PROLAY 具有 PROLOG 语言的匹配、推理、回溯等功能。
3. 它是一种描述性版图语言。
4. 它是一种参数化，相对位置型的版图语言。
5. 具有自动对接两个版图的功能。
6. 具有计算边界，找 I／O 点，图形变换(平移、旋转、镜象)等功能。
7. 具有缺省值和同类版图参数传递的功能。
8. PROLAY 具有交互、解释和编译三种工作模式。
9. 易于扩展和引进新的功能。

内含一个演绎版图数据库是 PROLAY 的一大特点。版图参数的传递，缺省值的获取，及版图特征点的检索，都是演绎数据库运行的结果。因此，在二个单元的二个 I／O port 之间进行联线的过程，便是对二个单元 I／O port 查询，然后在二点之间进行走线的结果。

二个单元的二个 I／O port 的对接(abut)是另一个版图推理过程的例子。abut(cella, cellb, porta, portb, Dd, cellc)便是代表这个版图功能的谓词。它的含义是以单元 cella 的 porta 点和单元 cellb 的 portb 为参考点，并保持它们之间的距离为 Dd，对接起来生成新的单元 cellc (图 1)。

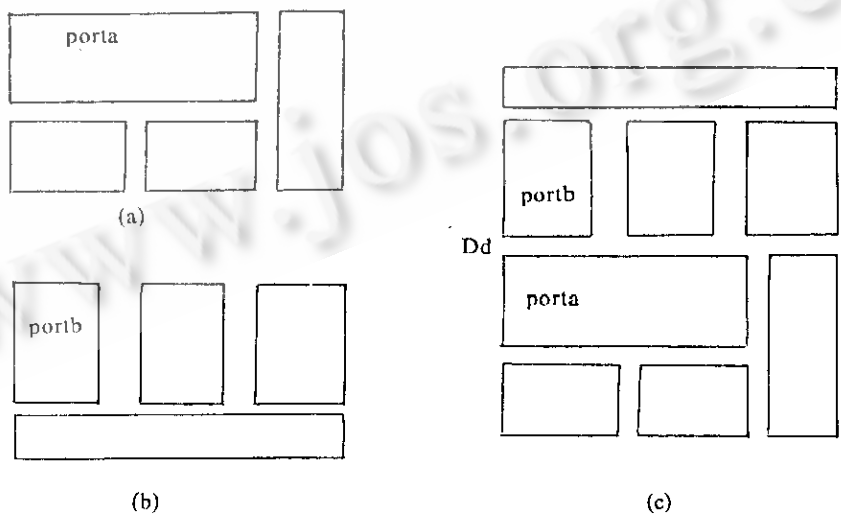

图 1 abut 谓词 a) cella; b) cellb; c) cellc

abut 的工作过程是这样：首先对 cella，cellb 进行查询，并找出其相应的 porta 和

portb，然后判断这些点在单元中的方位，进而推断对单元 cellb 要作什么样的图形变换，需要平移多少，旋转什么样的角度和做什么样的镜象变换．从图 1 可以看出，cellb 作了平移和以 x 为轴的镜象变换，从而使二个单元能以这些点为基准，将它们对接起来．

auto_abut 是建立在 abut 基础之上的另一个版图演绎谓词，它的含义是：给定二个单元和它们之间的联结关系．auto_abut 从中导出单元对接的"最佳"基准点对，和它们之间所需的距离(如布线区宽度或设计规则要求的宽度)，然后利用 abut 谓语，完成二个单元版图之间的对接．auto_abut 其说明如下：

```
auto_abut(Cella, Cellb, Connect_list, Cellc): -
    port_select(Cella, Cellb, Connect_list, Porta, Portb),
    evalu(Cella, Cellb, Connect_list, Dd),
    abut(Cella, Cellb, Porta, Portb, Da, Cellc).
```

其中 port_select 是从联结中导出 porta，portb 的谓词，而 evalu 是估算 Dd 大小的谓词．

版面布局(floor_planning)的原始构造 floor_plan_construct，可以看成是利用 auto_abut，abut 的一系列推理演绎过程，其输入数据为：

1. 参与布局的单元清单，即 cell_list 表．
2. 单元间的联线清单，wire_list 表．
3. 外引线排列 pin_list 表．

floor_plan_construct 的工作过程是这样：若参与布局的单元表为空时，则工作结束，得到最后的布局结果．否则从单元表中选出该次布局的单元，并生成已布局的单元和选出单元之间的联线表，利用 auto_abut 自动实施对接，在删除已布单元及其联线后，递归重复同一推理过程，直到全部单元布完为止．

§4. 电路识别工具

在某些设计系统中，往往需要人机交互地解决一定数量的任务，因而可能引入一些人为的错误．因此，需要使用逆向检查和分析，检查逻辑上和联结上的正确性，以及电特性是否达到了既定的要求．

在逆向设计系统中，需要对版图进行处理，分析版图设计上的特点，识别该版图所代表的电路．所有这些，都需要从版图数据抽出该版图包含的晶体管网络及晶体管级的各种参数．其次还要从这些晶体管网络中，识别出它们所构成的基本逻辑单元网络(如反向器 INV，与非门 NAND 和或非门 NOR 等)；然后再从这些基本逻辑单元网络中，识别出它们所构成的复合逻辑电路(如触发器等)．这种机械化识别过程，不但便于在逻辑级进行逆向验证，同时也便于人们对逆向设计的对象的了解．这里所介绍的是一个基于 PROLOG 的电路识别工具 PRORECOG，它能完成上述的二项任务．

PRORECOG 主要由如下几个部分构成：规则库(包括电路识别规则库、错误检测规则库和表达式推演规则库)，推理机，工作数据区和数据库，其结构如图 2 所示．它的工作过程如下：对数据库中的数据进行划分，抽出要识别的电路块送入工作数据区，推理机根据工作数据区提供的信息，在规则库中选取适当的规则进行电路识别，如果电路识别不了，就可能有如下二种情况：一是数据有错，也就是设计有错，二是新型或不规则的电

路．所以，在识别失败后，推理机便导向错误检测阶段，调用错误检测规则，找出数据存在的错误．如果在数据中找不出错误，那么这是一个新的或不规则的电路，推理机从而转向符号逻辑表达式的推演，找出被识别电路的逻辑功能．

PRORECOG 的工作还分成二个层次，第一个层次是从晶体管级数据运行上述的工作过程，识别基本逻辑单元电路，第二个层次是从基本逻辑单元电路级数据运行上述的工作过程，识别复杂的逻辑电路．

例如：开始的数据库的内容是：

ckt(p, 〔a〕, 〔vdd, c〕), ckt(p, 〔b〕, 〔vdd, c〕, ckt(n, 〔a〕, 〔c, n1〕),
ckt(n, 〔b〕, 〔n1, vss〕), ckt(p, 〔c〕, 〔d, vdd〕), ckt(n, 〔b〕, 〔vdd, d〕),
ckt(p, 〔b〕, 〔vdd, n2〕), ckt(p, 〔a〕, 〔n2, e2〕), ckt(n, 〔a〕, 〔vss, e〕),
ckt(n, 〔b〕, 〔vss, e〕), ckt(p, 〔e〕, 〔n3, f〕), ckt(p, 〔d〕, 〔vdd, n3〕),
ckt(n, 〔e〕, 〔vss, f〕), ckt(n, 〔d〕, 〔vss, f〕),

首先执行 level(0)的识别过程：

1. 电路块划分，从数据库中抽出

ckt(p, 〔e〕, 〔n3, f〕), ckt(p, 〔d〕, 〔vdd,n3〕). (1)
ckt(n, 〔e〕, 〔vss, f〕).ckt(n, 〔d〕, 〔vss, f〕).

加入工作数据区中．

2. 推理机根据 level，以及工作数据库的信息转向工作数据区中的数据的识别．为了加速识别过程，推理机中包含了一些识别规则选择方法的元规则．这样识别出这是一个 NOR 电路，其输入为〔d, e〕，输出为〔f〕．如此递归地执行这个识别过程，找出

ckt(*nand*, 〔*a*, *b*〕, 〔*c*〕), *ckt*(*inv*, 〔*c*〕, 〔*d*〕),
ckt(*nor*, 〔*a*, *b*〕, 〔*e*〕), *Dckt*(*nor*, 〔*d*, *e*〕, 〔*f*〕), (2)

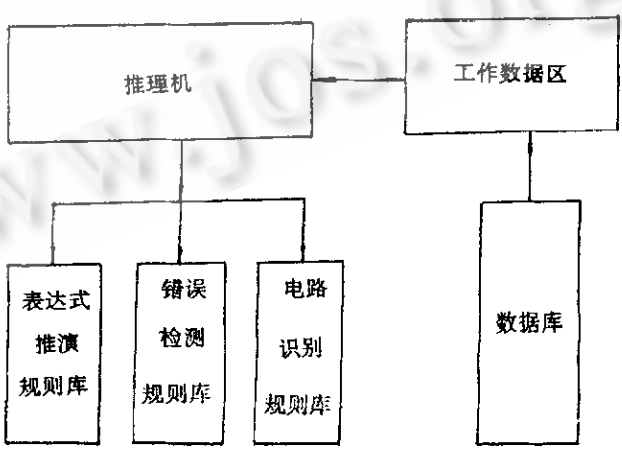

图 2 PRORECOG 结构图

由于没有不能识别的数据块，因而 level(0)圆满地完成，从而 PRORECOG 转入 level(1)的对数据块(2)的识别过程，在运行分块识别过程后，识别出一个 xor 电路：

ckt(xor，〔a，b〕，〔f〕).

§5. 结束语

上面我们简要介绍了几个运用人工智能技术的 CAD 工具。在演绎 CAD 数据库中，可以实现十分复杂的推理演绎型查询，在直接数据不存在的情况下，可以使用同类的关系，实现数据的传递和继承，或更进一步使用其他间接的推理查询。

对于版图语言 PROLAY，它内含一个演绎的版图数据库，许多过程性的版图工作，例如版图的安置，都可以通过一系列的推理演绎过程来实现；同时，也可以较方便地将一些启发式的规则，融合到它的推理过程中。

PROGECOG 能从晶体管网络中识别基本逻辑单元网络，然后再从其中识别复杂逻辑电路，因而可用于逆向设计和正向设计的逆向检查之中。

这些 CAD 工具都是用编译型 Prolog 写成，运行在 IBM PC.PSⅡ上。目前，它们正用于兼容计算机反向设计自动化环境 REDA 中[8]

参考文献

[1] R. Joobban et al., "Weaver: A Knowledge Based Routing Expert", 22nd Design Automation Conf., June 1985.

[2] Tamia Mano et al., "Knowledge-Based Expert System for Hardware Logic Design", 1986 Fall Joint Computer Conf., pp. 977-986.

[3] S. Costango et al., "Toward an Expert System for Logic Circuit Synthesis", IEEE Compcon Spring 1987 pp. 462-467.

[4] Y-L Steve Lin et al, "LES: A Layout Expert System", 24th Design Antomation Conf., 1987 pp. 672-678.

[5] 杨乔林，"基于知识和规则的电路模块生成环境"(1988年中科院计算所优秀论文).

[6] R. J. Lipton et al.,"All: A procedural Language to Describe VLSI Layouts", 19th Design Automation Conf., 1982, p. 468.

[7] S. A. Ellis, "A Symbolic Layout Design System", Proc. CAS Intern. Sym. 1982, pp. 670-676.

[8] 杨乔林，"XY／L 嵌入版图设计语言"，计算机研究与发展，86, No.3.

[9] 杨乔林，"兼容计算机逆向设计自动化环境"，中科院计算所，89，8.