

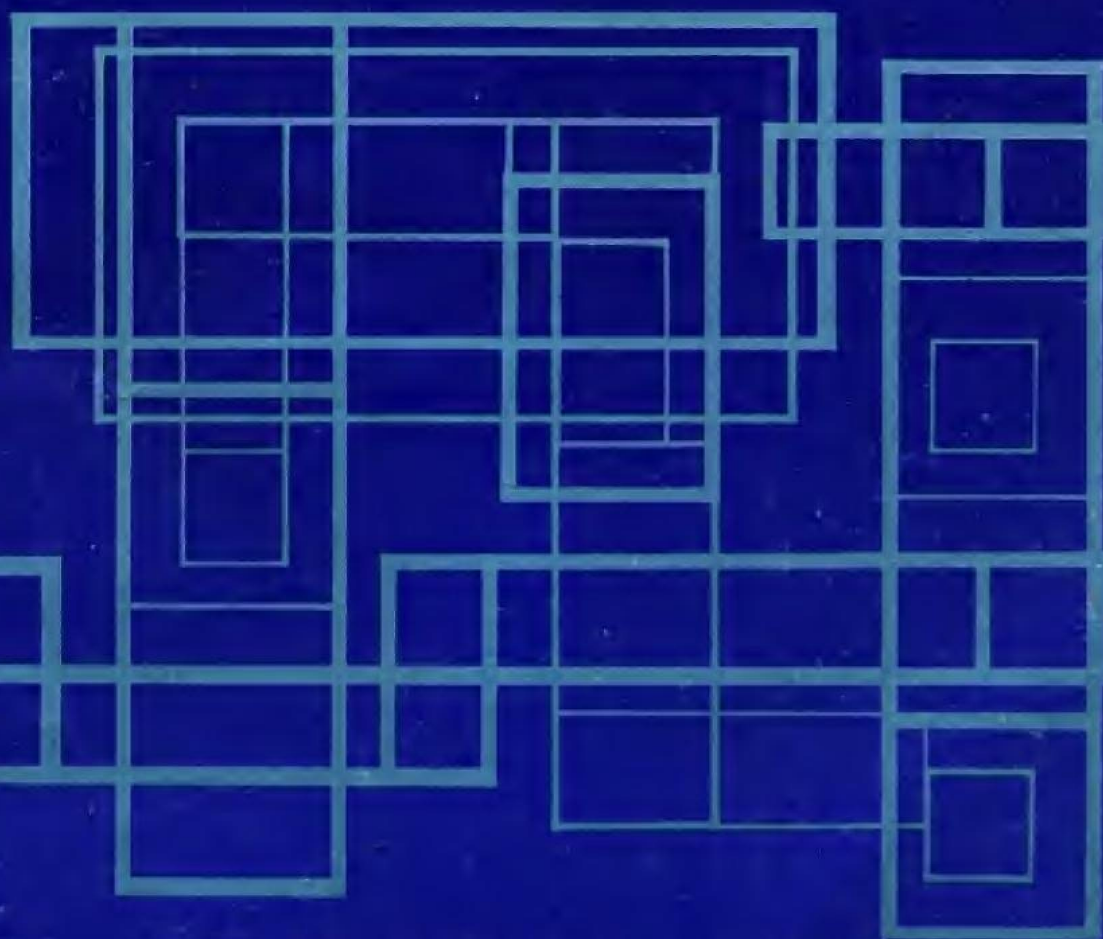
高等学校教学用书

# 超大规模集成电路

## 《硅编译和芯片自动设计技术》

[美] RONALD F. AYRES

解志华 译 陈兆龙 余志平 校



电子工业出版社

## 内 容 简 介

本书较详细地介绍了硅编译和芯片自动设计技术。全书分三大部分共九章。第一部分论述集成电路版图,即硅编译器的目标语言,介绍了有关程序设计的大部分概念以及相应的专门适用于这种程序设计语言的表示法。第二部分论述逻辑行为,即硅编译器的一种源语言,介绍如何将行为语言翻译成第一部分所介绍的表现(这种表现直接用作即刻进行版图综合的参数)。第三部分阐述从行为源语言有效而实际地翻译成版图目标语言的各种形式和一般技术,揭示了硅编译技术超过手工设计技术的最根本优点。

本书可用作高等院校超大规模集成电路计算机辅助设计方面本科生、研究生的教科书,也可作为计算机辅助设计系统的设计师手册,以及用作一个大型软件应用的研究实例。

## 超大规模集成电路

### 《硅编译和芯片自动设计技术》

〔美〕 RONALD F. AYRES

解志华 译

陈兆龙 余志平 校

责任编辑 詹善琼

电子工业出版社出版(北京市万寿路)

新华书店北京发行所发行 各地新华书店经销

电子工业出版社计算机排版室排版

顺义县李史山胶印厂印刷

开本:787×1092毫米 1/16 印张:29.125 插页7 字数:7270千字

1991年4月第1版 1991年4月第1次印刷

印数:1,000册 定价:8.40元

书号:ISBN7-5053-1193-X/TN·350

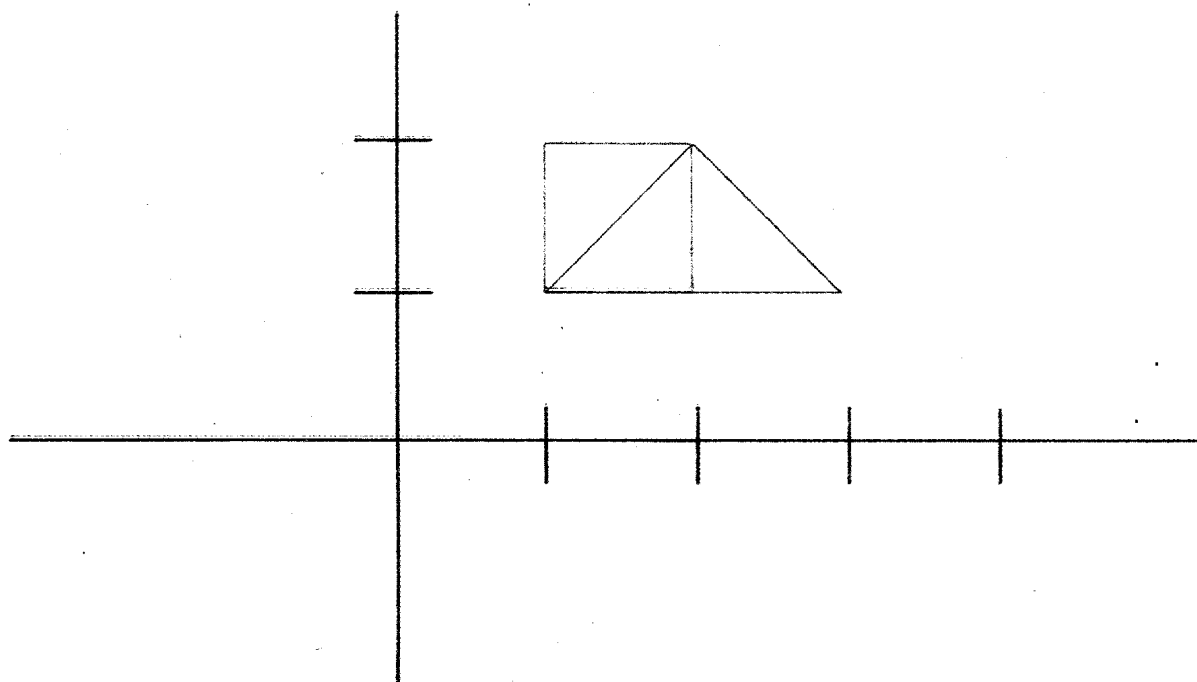


图1.3 框和多边形

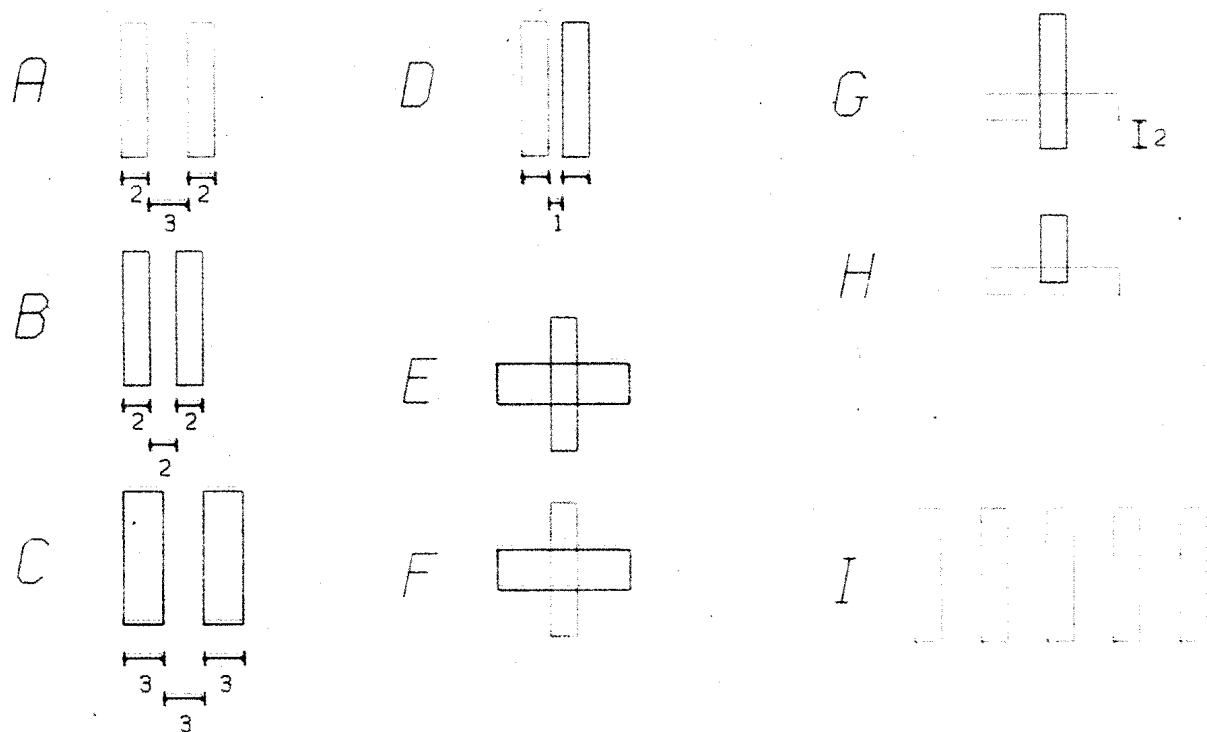


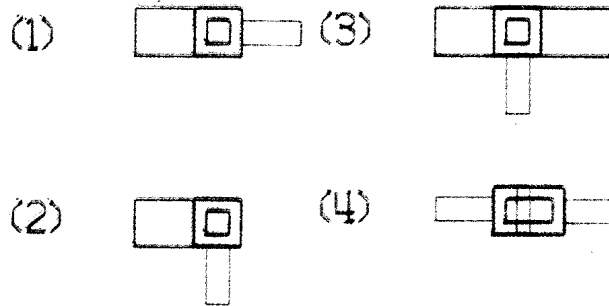
图1.4 设计规则

RCB 

GCB 

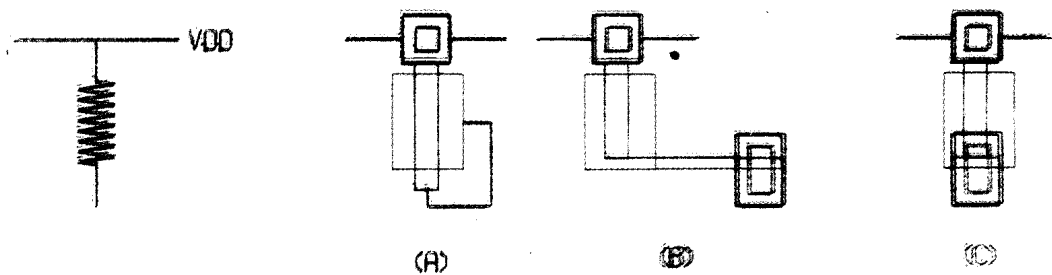
RCG 

(A)



(B)

图1.5 层间电的连接



源栅短接的晶体管

“掺杂的”晶体管和分开的反馈

完整的上拉晶体管

图1.10 上拉电阻器

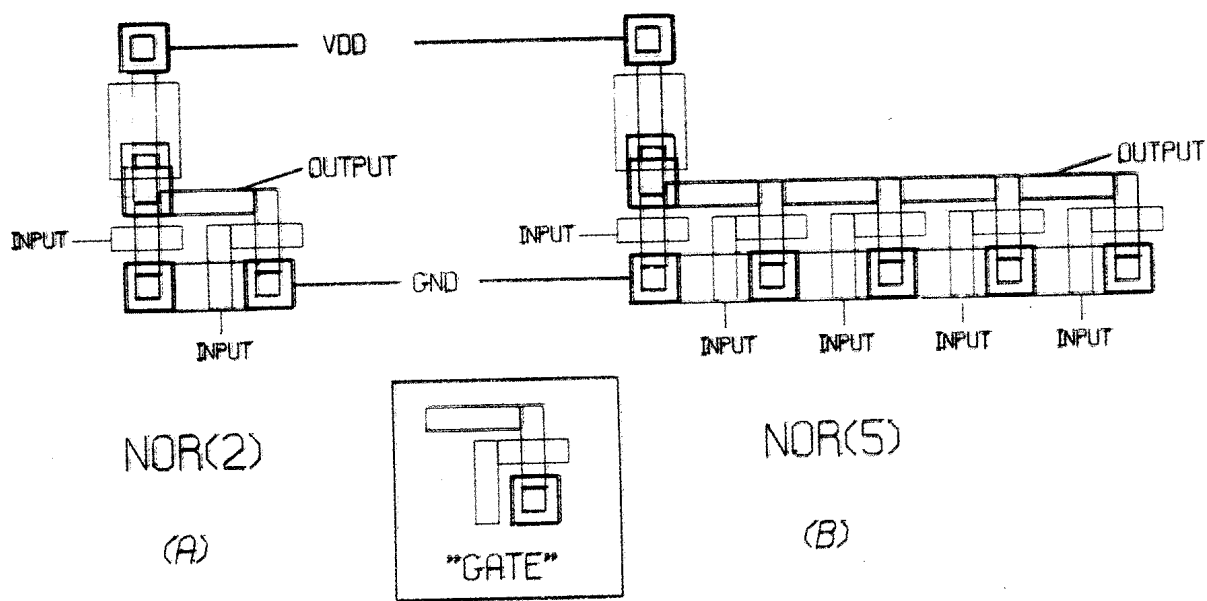


图1.15 NOR 门

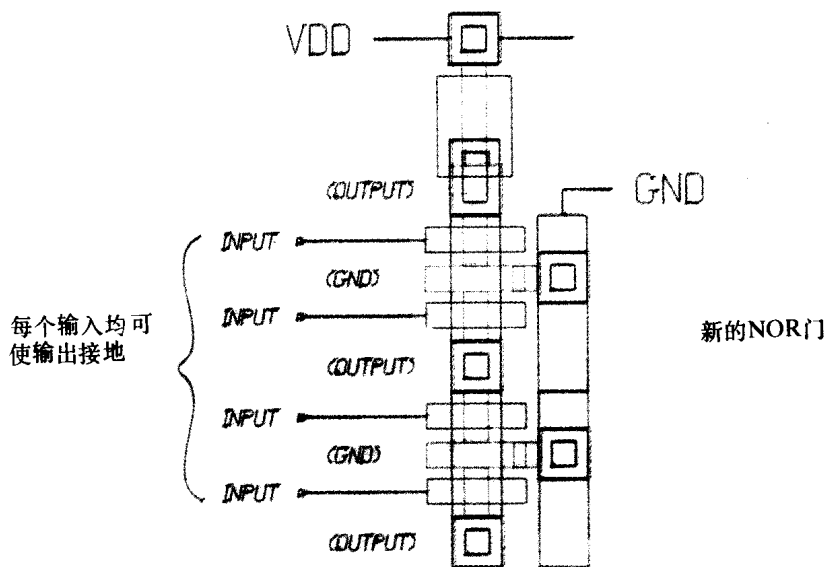


图1.16 另一种NOR门

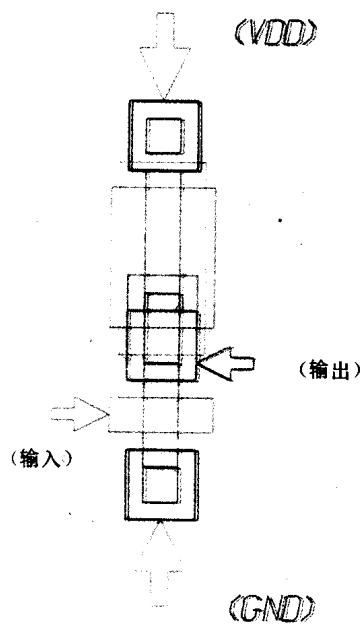
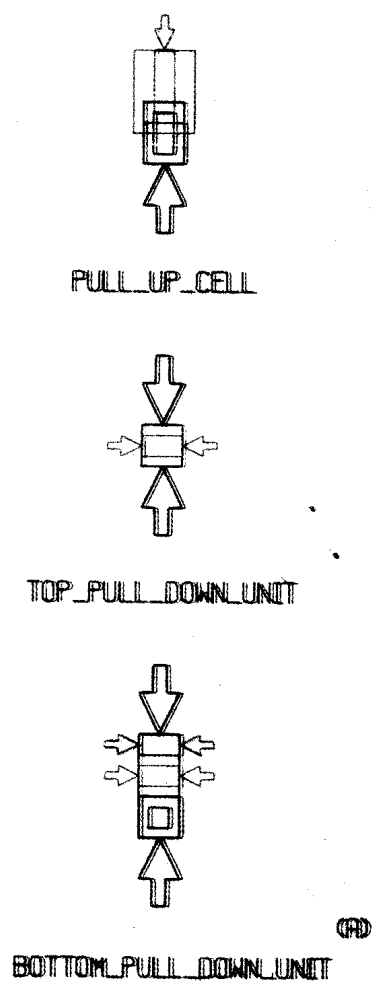


图6.1 用箭头表明端口的INVERTER-CELL



二种下拉单元的列

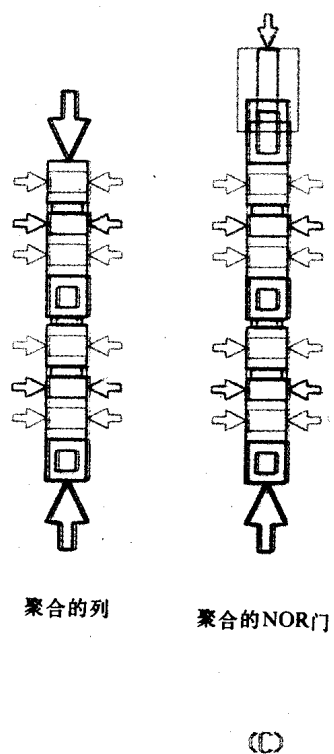


图6.3 NOR门结构的单元

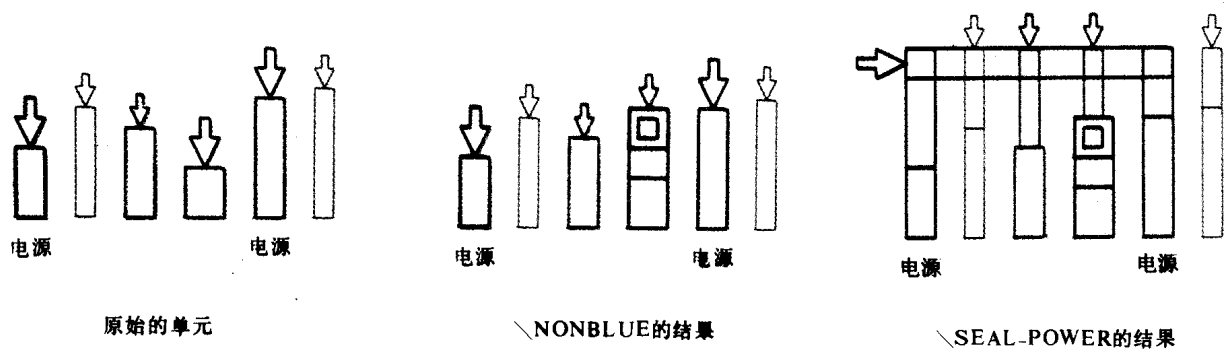
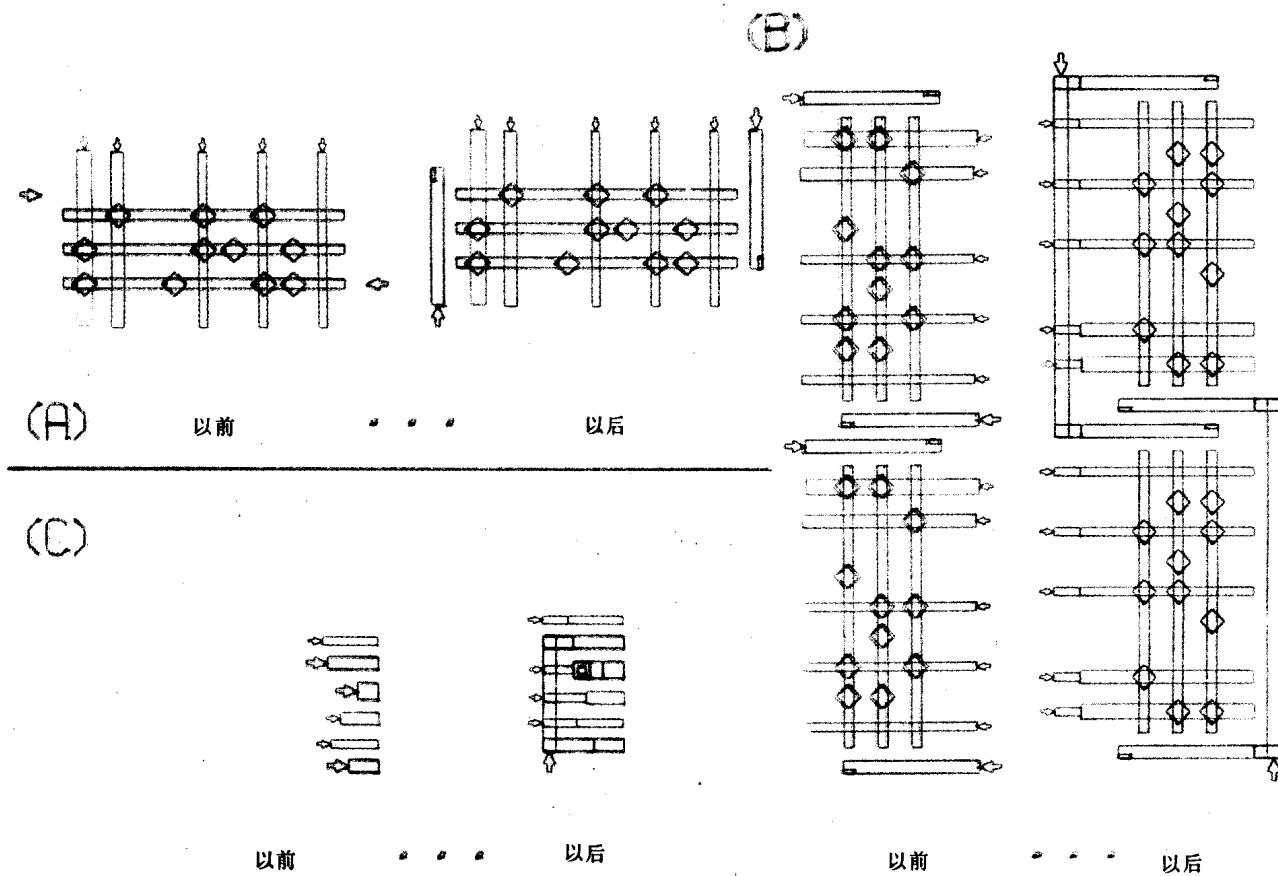
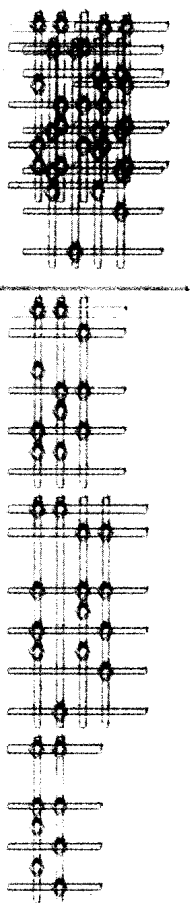


图6.10 一个单元的生长



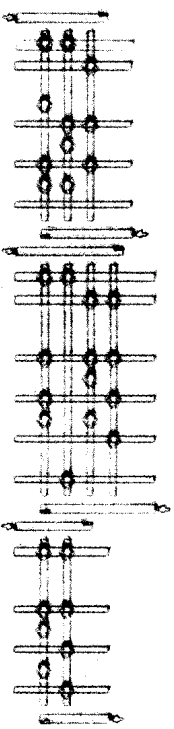
CELL \SEAL\_SIDE\_POWER -> CELL

图6.11 \SEAL\_SIDE\_POWER的例子



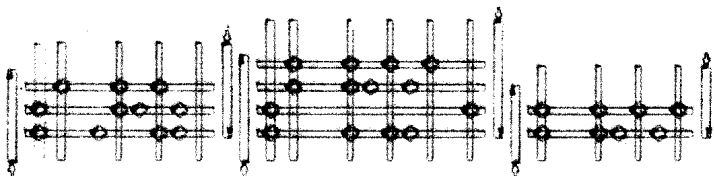
(b)

DX水平地对准这些单元

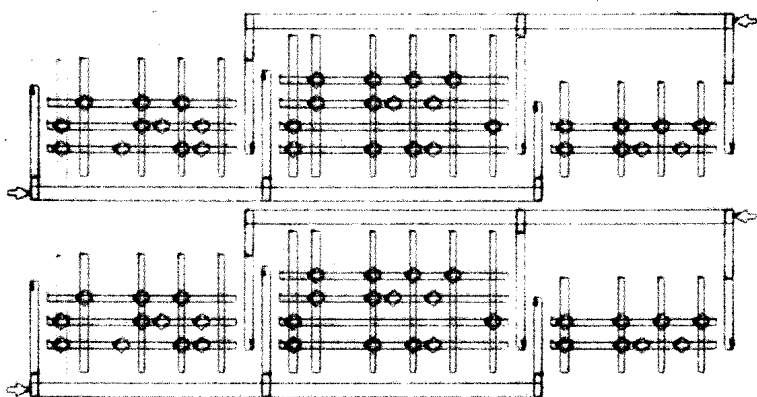


(c)

首先，每个单元的 \SEAL-SIDE-POWER



(d) 旋转后的结果

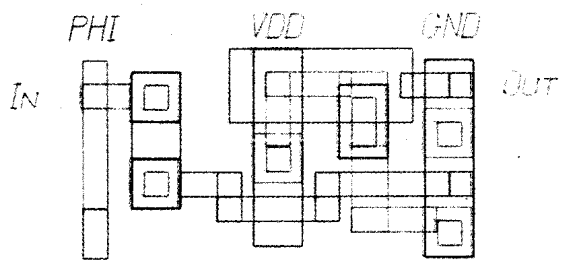


(e) 同样地应用于(c)的二次复制

CELLS \UNION.DX REAL

CELL

图6.12 \UNION.DX的例子



(H) 一个时钟元件

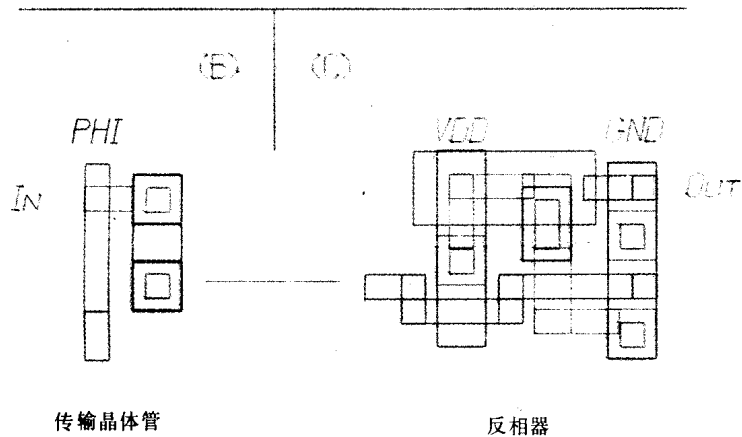
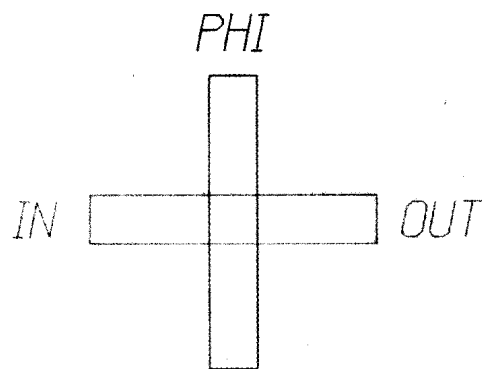


图9.16



(A)

(B)



(C)



(D)

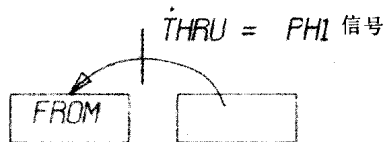


图9.17 有条件的依赖关系

## 译 者 的 话

本书为超大规模集成电路计算机辅助设计 (VLSI CAD) 方面的学术著作, 内容论述硅编译和版图设计。

众所周知, 集成电路制造技术进展迅速。而设计技术显得落后, 设计一个微电子系统的时间要用人年计算, 比如设计一个10万元件的集成电路, 一位设计人员需要工作20多年。因此, 集成电路尤其是 VLSI 的设计应当尽可能利用先进的计算机辅助设计 (CAD) 工具。可以断言, 如果没有CAD技术强有力的支持, 就不可能有强大的器件工业。

硅编译是当今CAD中最活跃的领域。硅编译可以将电性能和版图设计直接联系起来, 这是一种突破。实用的硅编译器可在一小时内产生出芯片的版图。无疑, 硅编译有助于解决所谓的硬件危机。

VLSI CAD的研究工作在迅速发展, VLSI CAD技术的应用在日益兴起, 本书的翻译出版正是为了满足这些需要, 在此之前, 未见有中文版的硅编译和版图设计方面的著作, 本书的出版发行填补了这一空缺。可以预见, 今后VLSI的设计技术会有很大的发展。自然, 本书就有很强的生命力。

该书是作者在加州理工学院从事教学和研究工作的总结, 在产业上有实用性, 在学术上有挑战性, 并使作为高年级大学生或研究生的教学用书。值得指出, 原著图中有些符号 (比如电阻, 反相等) 不符合现行的标准化要求, 译书保留其原状, 未作改动, 请读者鉴谅。译者因经验不足和水平所限, 译书中在专业技术与语言文字二方面定有不少缺点和毛病, 甚至错误, 欢迎读者指正。

本书之所以能与读者见面, 承蒙有关方面和许多人士的帮助、鼓励、赞助和支持。没有他 (她) 们, 书就难以问世。清华大学微电子研究所教授陈兆龙、余志平审核全部译稿, 北京集成电路设计中心高级工程师王正华对序、前言和第一章的译稿也提出了宝贵的意见, 北京第二外国语学院教授王遵仲在本人着手翻译原著时作了具体帮助, 同时得到李维六、杜立柱、王国光、周文琴、田建民、李志琴、谢连贵、严隽达诸位的帮助和支持。值得提到赵菊馨、京鸿和婷婷他们多年来为我创造良好的工作环境, 保证充分的时间, 我有条件得以顺利译作。在此, 对所有为本书作过贡献的朋友谨致谢意。

译 者

一九九〇年九月 北京

## 序

本书介绍硅编译和芯片自动设计技术这一新兴领域,这也许是当前和未来一段时间内,超大规模集成(VLSI)电路芯片设计方面最引人注目的领域。

本书打算给读者提供一个建立、理解和钻研硅编译的技术和概念的基础。本书可用作超大规模集成电路计算机辅助设计方面研究生一学年课程的教科书;也可作为计算机辅助设计系统的设计师手册;以及用作一个大型软件应用的研究实例。

硅编译技术有助于解决目前公认的“硬件危机”,即制造新的全定制集成电路芯片时遇到的极大困难。这种困难之所以成为“危机”,只是因为现已存在着容易用来制造功能很强、而又非常需要的各类计算机的工艺技术。然而,最棘手的问题却在于,如何将人的意图(即新芯片的行为描述)转化成芯片的“版图”(即一个完整的几何表现),而成熟的工艺根据版图可方便地制造出可靠的工作芯片。

任何旨在解决这个硬件设计危机的步骤,都会带来两个极其重要的结果。首先,成本大为降低,因而市场上会有更多的产品,这些产品将使每个人的生活更加愉快。而且,这还会使这些芯片工艺的应用范围大为扩展,使我们大家都获得以前难以想象的更大的好处。其次,消除了我们思想方法中人为划分(例如硬件和软件的划分)形成的某些障碍,因而扩大了我们对于信息处理和信息本身特性的科学认识,并使我们对于信息及其演化有了更为统一的看法。

芯片设计过程和此后的芯片制造过程的成本比较,明显地揭示出设计乃是“瓶颈”。设计过程产生芯片的版图,而制造过程则把版图转化成真正的芯片。就目前技术水平而言,设计过程,从开始规定芯片的性能(通常是拟定一个简短的文件来规定芯片的预期行为)到完成实用的设计,可能要花几百万美元和一年或一年以上的的时间。一旦完成设计(那时此设计已落后于当时的技术水平约一年时间),而制造过程即使是小批量生产的试样,也还可能要花 10,000 美元以下的费用和不到一个月的时间。

我们怎么会习惯这样巨额的设计经费呢?设计过程的结果是产生一个版图,即大约一百万个单独图形构成的图,而所有这些单独图形在一起必须满足制造工艺或称“语法”的限制。在这使人迷惑的详图中哪怕有一个错误,也会导致整个芯片不能工作。不但这一巨大的图形集合体要符合硅“语法”,而且还要用这种硅语言写成适当的语句,以便芯片如意执行。此外,设计必须满足电学的限制,这是因为整个工艺技术是以电学特性为基础的。

本书系统叙述“硅编译”这个自动产生版图的技术,它是解决硬件危机的一种切实可行的办法。我们创造了“硅编译”技术,并把“硅编译器”这个术语引入计算机科学,用以表示完全自动地将行为描述转译成芯片的版图,正好像软件技术中“编译器”这一术语表示完全自动地将高级语言转译成机器语言一样。人的干预是随意的、有限的,但绝不能影响最终版图行为的正确性。

实用的硅编译器已能在一小时内,根据一周中所研究的行为描述产生出芯片的版图。这些芯片是从通用微处理器到专用微处理器。这些硅编译器只不过是这样的第一批辅助设计工具,它们实现了从行为描述到正确版图的整个设计过程的自动化。它们利用硅本身的潜力,而没有

旧工艺固有的思想方法和各种限制,如“门阵列”和“多元胞”技术采用的印制电路板布局与布线的作法,因而效率较高。

各种半自动技术已在工业界和学术界出现,它们可以孤立地和\或部分地解决有限的、明确定义的问题。如果不考虑全局编译,即使彻底解决了一些局部问题,也往往使整个芯片没有什么改善,或毫无改善,甚至会产生相反的效果,特别是当一些约束不严的接口留给容易出错的人工干预时更是如此。与此相反,本书强调全局问题和详细解决。有时,甚至用非最佳方法解决局部问题,可见灵活性和潜力都是很大的。

人们也许会感到惊奇的是,完全自动化需要的努力,实际上比独立地用实用办法解决大部分局部问题所需的努力要少得多。这主要因为,制订编译器的人完全控制着整个自动化过程。这自然促使问题更有效地解决,其总效果要超过集中的最佳局部解决。

这种综合统一的解决办法,使过去想不到的一些工艺技术采用硅编译可能特别合适。编译器解决问题的复杂程度极高,因而促进了一些手工设计无法达到的新技术。(我们最近已经看到,由于 Pascal 语言和 LISP 语言机器的出现,一种技术已适应了软件编译器。)

硅编译领域在电气工程领域和软件工程领域之间架起了桥梁。我们今天在芯片设计领域中遇到的问题和态度,和一二十年前在软件领域中存在过的问题和态度相同。在芯片设计和软件两方面,设计者都面临基础工艺技术迅速发展的局面,而这种局面改变了以及改变着不同实施方法的相对成本。卡脖子的问题不再是基础工艺技术的限制,而是人们在进行设计(版图或程序)时的局限性。

软件编译工程师们过去和现在面对的问题,都是如何最有效地利用不断增长的存储器容量和速度。软件编译器刚出现时,人们不太愿意应用它们,因为担心失去效率和控制。然而,今天这些担心几乎已不存在,现在大多数软件设计者都不再问是否要用编译器,而是考虑用哪一种编译器(语言)更好。目前,随着硅芯片容量不断地迅速增长,芯片设计者也面临着同样的问题,即如何最好地将人们的技能表现在这个慷慨的硅片上。

我们详细地研究一个硅编译器,为了使所说明的各种技术和概念保持连续性。这里选取了硅编译器 RELAY (REcursive Logic ArraYs),因为它是通用的和能够说明适用于各种硅编译器的许多技术和概念。RELAY 包括:输入用同步逻辑表示的行为描述;转化成最优的不相交形式(disjunctive form);进行行为级模拟和转化成同步可程序逻辑阵列(PLA)的版图结构;进行布局和布线;以及建立电学性能模型。

RELAY 还特别适于说明芯片上逻辑再分配或再组织所产生的广泛而细微的变化与优化。后面这种作用清楚地揭示出硅编译器新的灵活性和用处,而这种灵活性和用处使得软件编译器对包括平面布局在内的所有高级设计阶段都是不可缺少的——在这些阶段,实际上集中了多数使人头痛的较重要的问题,因为解决这些问题花费是很大的。

从这些努力中得到的最有价值的信息是一批技术和新的思想方法,它们有助于建立这些和其它的硅编译器。本书主要就是讨论这些技术和概念。

事实上,这些技术不仅适用于硅芯片,而且也适用于一些有待发明的工艺,而这些工艺的功能都是通过二(或三)维版图(图形)表示的技术规格得到的。

本书不详细讨论版图设计技术,因为这些技术依赖于制造厂家和制造工艺。我们也不一般地讨论芯片的物理基础。其它教科书讲述这些方面,并为我们的抽象思维提供了严格的基础。例如,只有在必要时我们才提到电学性质。我们重点讨论用来产生各种版图和用来产生更具特

点的完整版图软件技术。

本书包含许多有关程序设计的示例,这对于增强人们的信心,坚定人们“能做”的态度,说明任何计算机程序设计中必须考虑的全部重要细节,都是必要的。为此目的,我们选择了程序设计语言 ICL(Integrated Circuit Language),此语言是为了研究超大规模集成电路而专门开发的。ICL 适用于这类教科书,主要是因为 ICL 能隐含地处理许多程序设计细节(对于许多应用而言)。ICL 并不使我们担负与 VLSI 毫无关系的程序设计细节(例如,指针、存储器管理和各种其它难处理的问题)。事实上,ICL 很适合作为一种明确的形式表示法,就象在数学书中数学表示法那样有用。

我们把 ICL 表示法作为本书例子的组成部分而加以解释,这样读者可以实际应用这些例子,至少在理解时是如此。事实上,本书选取的所有程序设计例子,都曾在计算机上运行过,为的是验证它们的正确性,和产生许多图例。ICL 是一种完整的工作语言,它已经唯一地用于形成和实现二个完整的硅编译器(即 RELAY 和 BRISTLE BLOCKS),而且也已经在非常经济地根据版图制造芯片方面为美国的芯片设计师服务。

老练的程序设计者可能会怀疑,这种语言所提供的十分简单的表示法,是否会使效率显著降低。令人惊奇的是,至今的经验证明,情况恰恰相反。这在很大程度上或许是由于语言人格化提供了十分直接通讯的缘故。

## 引 言

集成电路制造方面的巨大进展,导致电路集成度大约每二年翻一番。随着电路集成度的增加,电路复杂性也增加。这种进展使今天的单片微处理器和大容量存储器成为可能。这种进展也使整个设计过程的复杂性大为增加。本书所介绍的硅编译技术就是对付这种增加了复杂性的一种方法,这种方法在软件领域中早已存在,因为在该领域中,极大的复杂性首先成为一个严重的问题。

一个集成电路的产生经历五个基本步骤。首先,试制者拟订一个功能级的技术规格。然后,将那个技术规格形式化,并进行模拟以建立对其正确性的信心。随后,主要的任务是将所需的性能翻译成硅的二维语言,执行这些语言就产生几套非常复杂的交叉重叠的图形。将这几套图形(即总称为版图)送到硅芯片制造厂,然后从制造厂取回大量的芯片。最后,测试并封装每一个芯片,于是试制工作便告完成。

主要费用花在将形式化的功能技术规格翻译在二维版图上。一般来说,翻译者可用纸和笔或用计算机图形编辑器来创作一张一张的版图。这个过程必须极端小心,比软件研制还要小心得多,因为周转周期很长和一次试做(芯片制造)的费用很高。手工绘制版图不仅时间冗长而且单调乏味,很容易出错。甚至计算机图形编辑器也是接受设计者所绘的一切,而不顾硅二维语言所具有的语法或约束。

为了验证是否符合硅的约束,已研制出制版前检验版图用的软件。一些程序检查版图,并对某些违反几何约束的错误加以标志。其它一些程序则试图直接从版图再产生原理图。这些技术可以早期发现错误,从而使为排除版图错误所需的投片次数大为减少。然而,这样的程序不可能发现所有各类错误,仍然要有人制出版图来才行。

这种硬件研制技术与现行的软件研制技术不同。一方面,程序设计者具有可将高级描述翻译成机器语言的编译器。另一方面,版图设计者具有反编译器,即可读硅语言而重建高级描述的程序。

反编译器是很难建立的,因为它们必须接受硅语言中所有可能的表达方式,包括设计所用的一切技巧。例如,全部设计规则检查程序,或是不给所有错误都作上标志,或是给许多完全正确的情况做上标志,好象它们是错误了似的。

而建立编译器却要容易得多,它仅需接受有明确定义的形式语言。对这种形式语言的唯一要求,就是它要能描述数字行为。

可以保证,由工作硅编译器所产生的版图,在功能技术规格和硅语言的约束二方面都是正确的。这样,编译器的使用者(例如,版图的设计者)就可解脱出来,以便集中精力完成更富有创造性的任务。他们可以试验各种各样的结构,和迅速权衡各种版图与定时方案。他们可以发现版图的“瓶颈”,并用手工方式去布放这些部位。类似地,软件设计者用手工给那些消耗大量计算机资源的程序写程序。

尽管用硅编译器产生的版图比用手工产生的版图要占较多的面积,但是有二个因素鼓励人们去应用硅编译器。首先,就未来的工艺技术而言,我们可能最终使密度为今天的 100 倍。

这实际上意味着面积为今天的 100 倍,或者说速度为今天的 100 倍。当可用的存储器和计算机速度使得人们用机器语言写程序变得无利可图时,软件编译器就为人们所接受了。硅设计方面的情况也是这样。

其次,就目前的工艺技术而言,人们能够买到做在一个芯片上的完整的微处理器;然而,在系统中使用这样的微处理器时,它所需外部逻辑的芯片数可能要比微处理器的芯片数多得多,特别是当外部逻辑用市场上买到的小规模集成(SSl)或中规模集成(MSI)电路芯片来实现时更是如此。用硅编译器时设计时间最少,因而有利于制做一个单一芯片来实现外部逻辑。在一个全定制的芯片上实现许多 SSI 或 MSI 芯片的功能,可以降低制造印制电路板的费用,费用降低对于大批量生产特别有吸引力。

### 一个硅编译器的剖析

硅编译器有两个需要注意的问题,对于硅编译器的使用者来说,这二点是十分显然的:

- 1 源语言:使用者用以规定新集成电路芯片所需完成的功能或行为的语言。
- 2 目标语言:在称为版图的一个十分复杂的二维彩色图形中所利用的硅的能力。

硅编译器的质量或可用性由下列两条来衡量:

- 1 源语言提供行为描述的难易程度。
- 2 制成芯片的硅面积利用率和工作速度。

这两条考虑中第一条或许更为重要,而且更易为人们所忽视。自从集成电路技术诞生以来,设计者们一直是直接使用目标语言,并且最关心硅资源的有效利用。

然而,一个集成电路芯片的最终价值如何,通常要看它在一个较大的系统内解决特定问题的能力。倘若芯片的工作不能满足该系统的需要,即使是最有效的芯片也失去其实际价值。如果系统设计者必须改造系统和在该给定芯片周围加上新的芯片才能接受该给定芯片所提供的功能,那么最有效的芯片也是没有得到有效的利用。

这样,仅以注意硅效率的狭窄观点而设计的集成电路芯片,只不过是把责任或低效率推给它最终所在的整个系统而已。

与此相反,首先考虑具体应用所设计的芯片,如果它在整个系统中的作用完全满足该系统的要求,那么这个芯片就能承受硅的某些低效率。这种功能优化的芯片设计,主要不是在硅的领域,而是在可能实现的行为或功能领域。硅编译器的“源语言”恰恰适用于这一高级领域。

在硅编译技术领域中最有创新的进展大概是围绕着源语言设计而反复出现的。虽然有许多语言(例如,几乎所有的软件程序设计语言)能够表示行为,但是硅编译的源语言则必须:

- 1 能直接描述硅制芯片的各种行为(即平行性)
- 2 同时具有总体完整性,使该语言始终是描述行为的语言,而不仅仅是设计版图几何图形的语言。

源语言压倒一切的重要性,在今天的软件领域中是显而易见的。某一时期以来,人们一直根据编译器所支持的源语言来选择软件编译器,而不是根据它最终产生目标(机器)语言代码的效率来选择编译器。当总体上现实地考虑全部努力时,无生命资源的有效利用相对人们的方便而言处于次要地位。

### 软件约束和硅约束的比较

软件和硬件的发展围绕着同样的基本考虑:软件设计者设计存储器的一维阵列,而集成电路设计者设计硅的二维面积。每种情况都必须满足种种约束,才能获得实用的产品。此外,无论哪一种情况都包含着多次修改。

在软件约束和硬件约束之间存在一些基本的差异。例如转向语句(GOTO)用软件实现时仅耗费 GOTO 指令本身所需的存储器。指令位置和目标地址之间的距离,对执行时间和存储器的耗费都不起作用。与此相反,硅的 GOTO 则要求有一定面积来容纳连接二个位置的“连线”。较长的连线耗费较多的面积和较多的执行时间。而且,为了避免不必要的短路,硅中的连线必须躲开障碍物。软件 GOTO 没有障碍物;它不需要躲开存储器中的某个字组。

软件和硅之间的另一个差异涉及硅中所需的不同层次。硅中的一个信号可以在几层中的任何一层上表示出来。在某一点,信号传送时必须改变层次,这就要求三种或更多种图形组合起来实现这一转换。可是,软件中的赋值语句一般不需要这样的转换;存储器中所有的存储单元处在同一层上。对于这一规律,软件的确有少数例外,例如,在 IBM 360 型机中通用寄存器和浮点寄存器之间的区别;CDC6600 型机中 A、B 和 X 寄存器;以及在更抽象的数据类型定义域中,整型(INTEGER)和实型(REAL)之间的区别。然而,软件编译器通过隐含的翻译方便地掩盖了这些引起麻烦的区别。硅编译器则通过自动引入适当的图形来实现层间转换,从而掩盖了层间区别。

### 组织的作用

为了用任何方法都可产生版图,必须将功能描述分割成一个概念组织,或者说层次结构。在层次结构中每一层完成一个足够简单的功能,使其实现很有把握。每一层包含更深的层次,并有“粘结剂”将这些更深的层组合成现在这个层。在软件中,由嵌套的程序实现层次结构。在程序中,“粘结剂”是除了子程序调用以外的全部程序。在层次逻辑中,“粘结剂”是把深层次的输入和输出联系起来的一组方程式。在硅中,“粘结剂”则是一个附加的版图,它把从这些深层得到的各个版图集成起来。

硅的二维几何图形使层次结构显得更为重要。当一些版图从更深的层次形成时,如果这些版图不能有效地装配在一起,或者如果在各单元之间传递信号要耗费大量面积,那么使这些版图形成整体所需的“粘结剂”就会变得过多。改变层次结构,就可以改变从较深层次形成的版图的形状。它还可以改变单元之间的距离,因而有可能减小传递信号所需的面积。

在二维硅中,层次结构的效果经常迫使人们重新组织层次结构。当人们用手工产生版图时,重新组织层次结构的费用是特别大的,因为它破坏一些已设计好的单元,引入一些有待设计的新单元。

## 本书的结构

本书分为三大部分,第一、二两部分是第三部分的基础。

第一部分论述集成电路版图,即硅编译器的目标语言。第二部分论述逻辑行为,即硅编译器的一种源语言。第三部分把第一、二两部分结合起来,介绍各种从源行为到目标版图的完整翻译。

第一章介绍硅介质的能力,特别是 NMOS 工艺。这一章给出基本的版图形式,和保持几何与电学完整性所需要的约束。然而,这一章的主要着眼点是,使读者熟悉含在丰富的程序设计语言中的版图描述。我们称这样的版图描述为“参数化的”(Parameterized)版图描述。

在程序设计语言中嵌入版图描述的选择有下列优点:

- 1 参数化的版图描述所提供的好处是巨大的,极小量的仔细描述可以产生各种有用的、单独的版图,而每个版图都是相当可信的。
- 2 任何硅编译器,即一个计算机程序,都必须产生版图,即目标语言的一种表现。因此,这种版图的产生就是硅编译器程序本身描述的一部分。

第一部分的第二章提出了一种实施方法,从而完善了第一章关于版图综合的介绍。本章介绍版图的表现和分析技术。这一章还介绍有关程序设计的大部分概念,以及相应的专门适用于这种程序设计语言的表示法。

本书第二部分论述逻辑行为,或者说功能描述。真正的功能行为是一个与实现介质无关的现象。我们因此建立一种描述行为的语言,这种语言与集成电路版图无关。我们详细探讨了这种语言本身的完整性,而且最终提出把这种语言嵌入我们的程序设计语言中的公式化形式。

本部分的第二章介绍如何将行为语言翻译成第一部分所介绍的表现,这种表现直接用作即刻进行版图综合的参数。至此,将版图描述和行为描述二者嵌在同一程序设计语言内的好处就变得十分清楚了。

本部分第三章充实了行为语言,主要是为了人们使用方便。它引入了以结构化方式描述逻辑行为的能力。因此,即使对大型系统的描述而言,仍保持高度的可信性。这一章包括许多实例。

第三部分阐述从行为源语言有效地和实际地翻译成版图目标语言时的各种形式和一般技术。无论特定的源语言和特定的目标语言(例如,NMOS)如何,所介绍的大多数技术都同样适用。关键性的概念包括:为中间计算选择的表示法,为这些中间数据所假设和保持的数学性质(公理或规约)。所介绍的各种技术,其主要差别在于赋予中间数据的数学性质。每一技术最终都提供不同特征和应用的版图。(这些技术使得“门阵列”法和“多元胞”法成为极其有限的特例。)

这部分的第一章介绍一种包括几何和行为二个领域的表示法,还讨论规约(公理)在硅编译中所起的作用。本章完整地介绍如何一般地将结构化的行为语言翻译成一个具有特定特征的版图。本章还概述了一些构成“硅编译器”的技术。

这部分的第二章也许揭示了硅编译技术超过手工设计技术的最根本的优点。本章特别为逻辑重新组织提出了方便、简单和高水平的方法。尽管这些以简单程序方式提供的重新组织方