

目 录

前言

第一章 引言	1
1.1 模拟集成电路设计的特征	1
1.2 模拟集成电路设计自动化及其任务	5
1.2.1 模拟集成电路设计自动化的层次式模型	5
1.2.2 模拟集成电路的表象与模块生成	6
1.2.3 模拟集成电路设计自动化系统的任务及必备条件	7
1.3 模拟集成电路及其设计自动化的重要性	8
1.4 模拟集成电路设计自动化的进展.....	10
1.5 本书的主要内容.....	12
第二章 电路级综合的数学模型	14
2.1 电路级综合与混整型非线性数学规划.....	14
2.1.1 解析算法与通用电路模拟器	15
2.1.2 目标函数	16
2.1.3 求解混整型非线性数学规划问题的经典算法及其不足	16
2.2 模拟退火法.....	17
2.3 模拟退火法的不足与简单改进策略.....	21
2.4 多分子模拟退火法.....	22
2.5 算例.....	24
2.6 小结.....	26
第三章 模拟电路的符号分析法	27
3.1 符号分析法的功能及其应用领域.....	27
3.2 符号模拟器的典型流程.....	31
3.3 建立符号电路方程组.....	33
3.3.1 表矩阵法.....	34
3.3.2 改进节点法	34
3.3.3 压缩改进节点法	35
3.3.4 器件失配.....	41
3.4 符号线性方程组的求解方法.....	42
3.4.1 符号求解法的基本思路	42
3.4.2 排列法	42
3.4.3 递归展开法	43
3.4.4 动态行列选择	45
3.4.5 数值解法与符号解法	46

3.5	符号逼近法	46
3.5.1	数值逼近与符号逼近	46
3.5.2	逼近误差	48
3.5.3	逼近算法	49
3.6	小结	52
第四章	模拟集成电路的结构级综合	53
4.1	模拟集成电路的行为描述	53
4.2	基于状态方程组的功能块构造法	54
4.3	给定 s 域传输函数 $H(s)$ 的模拟滤波器综合	56
4.4	基于最佳一致逼近的函数综合方法	59
4.4.1	基函数的选择	59
4.4.2	组合系数的确定与电路实现	60
4.5	综合实例	61
4.6	小结	64
第五章	模拟集成电路的单元级综合	66
5.1	拓扑选择与器件尺寸优化	66
5.2	单元级综合的数学描述	67
5.3	同时确定电路拓扑和器件尺寸的策略	69
5.4	高性能运算放大器的综合	70
5.4.1	层次式分解策略与高性能运放的拓扑综合	70
5.4.2	偏置电路的综合	71
5.4.3	输入级与中间级的综合	73
5.4.4	输出级的综合	75
5.4.5	高性能双极型运算放大器的“超级电路”	78
5.5	综合实例	80
5.6	小结	87
第六章	模拟集成电路的版图综合	89
6.1	模拟集成电路版图设计的特征	89
6.1.1	版图寄生参数	89
6.1.2	信号间的耦合效应	91
6.1.3	器件失配效应	92
6.2	半定制模拟集成电路版图设计技术	94
6.2.1	模拟阵列	94
6.2.2	参数化模拟单元	95
6.2.3	模拟标准单元	95
6.2.4	模拟宏单元	96
6.3	器件级布局算法	98
6.3.1	器件级布局的基本思想	98
6.3.2	拓扑约束	103

6.3.3	器件合并	105
6.3.4	权系数的选择	107
6.4	器件级布线方法	108
6.4.1	迷宫算法、线探法与线扩展法	108
6.4.2	线扩展法的基本思想	110
6.4.3	重布线策略	114
6.4.4	对称布线	116
6.4.5	消除连线串扰	118
6.5	小结	120
第七章	模拟集成电路的硬件描述语言	121
7.1	描述域的抽象	122
7.2	混合信号系统及其行为抽象	123
7.3	模拟算符与模拟事件	124
7.4	混合描述域及混合信号系统的描述	125
7.5	小结	127
参考文献		128

第一章 引言

随着超大规模集成电路技术的迅速发展,专用集成电路(ASIC)正朝着系统集成的方向发展,这类微型系统通常都既包含数字电路又包含模拟电路,其设计是一项非常复杂、繁重的工作,需要使用计算机辅助设计(CAD)工具以缩短设计时间、降低设计成本。对于数字电路而言,人们已经研究出了许多相当完美的 CAD 工具^[1];而模拟电路的设计工具和设计方法却为数不多^[2]。目前,绝大部分模拟集成电路是模拟电路专家们手工完成的——采用简化电路模型、使用电路仿真器对电路进行反复的模拟和修正,手工绘制其物理版图。

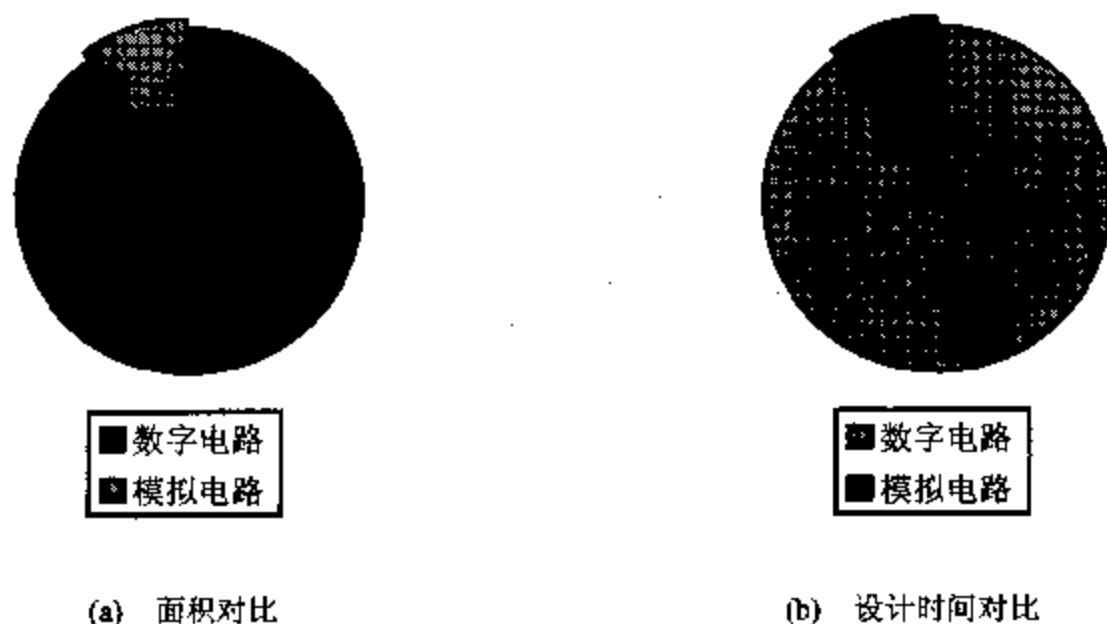


图 1.1 ASIC 芯片中模拟电路和数字电路所占的面积及所需要的设计时间

图 1.1 粗略地勾画了目前 ASIC 芯片中模拟电路和数字电路所占面积及所需设计时间的对比。从图 1.1 不难看出:尽管 ASIC 中模拟集成电路的规模(面积或器件数目)非常小,但其设计成本却占总设计成本的绝大部分,提高模拟集成电路设计自动化水平是降低 ASIC 设计成本的关键环节。事实上,落后的模拟集成电路设计水平已经成为制约系统集成技术发展的一个瓶颈,因此,人们在近年来开展了广泛的关于模拟集成电路设计自动化方法的研究,并取得了一些初步成果^[3-6]。

1.1 模拟集成电路设计的特征

抓住模拟集成电路设计的特征是深入开展模拟集成电路设计自动化方法研究的首要条件,为此,我们给出关于模拟集成电路设计的定义。

将关于电子系统(或子系统)的行为描述映射成模拟电路(包含电路拓扑结构和器件尺寸)和物理版图的过程称为模拟集成电路设计。

我们以调制解调器为例来介绍关于模拟集成电路设计的概念。图 1.2 是调制解调器及其应用实例。调制解调器包括发送和接收两部分,在两台计算机的通信过程中起桥梁作用,其中,发送器将其输入端的数字信号流(来自其中一台计算机)变换为符合 CITT

规范的模拟电压波形,然后通过电话线发送出去,具体的指标包括:波特率为 56.6K、精度达到 0.1%、带内噪声和失真小于 -60dB 等;而接收器的功能正好与此相反,它将输入端符合 CITT 标准的模拟信号(来自电话线)逆变换为数字信号,并以 56.6K 的波特率传递给另一台计算机。这些行为描述定义了调制解调器应该做什么(而不是怎么做)以及相应的性能指标;而调制解调器的设计就是首先将它分解为调制、解调、滤波、话路均衡和话路驱动等功能块,然后用相应的模拟电路来实现这些功能,使调制解调器的总体性能达到行为描述所给定的性能指标。

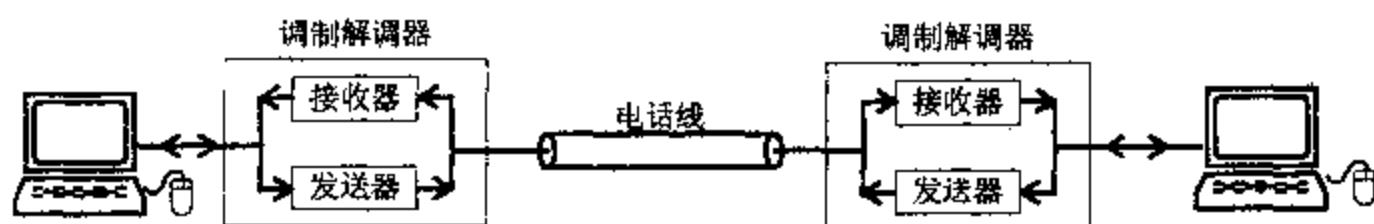


图 1.2 调制解调器及其应用实例

模拟集成电路设计比数字集成电路设计复杂得多,具体表现在模拟集成电路的种类繁多、电路性能多样化、电路设计难以积木化等诸多方面。下面将详细分析模拟集成电路设计的主要特征,它们是造成模拟集成电路设计自动化水平远远落后于数字集成电路这一局面的主要原因,同时也是研究模拟集成电路设计自动化方法所必须重点考虑的因素。

1. 层次不明晰

数字系统的设计可以分解为行为级、功能级、寄存器传输级、逻辑级、晶体管级等层次^[2],这种设计层次的划分已被人们所广泛接受;但对于模拟集成电路而言,其设计层次目前尚没有严格、通用的定义,设计者目前所采用的是一种十分松散、层间分界不够明晰的结构划分法。图 1.3 是 $\Sigma\Delta$ 模/数转换器($\Sigma\Delta A/D$)的一种基于结构的设计层次划分。从

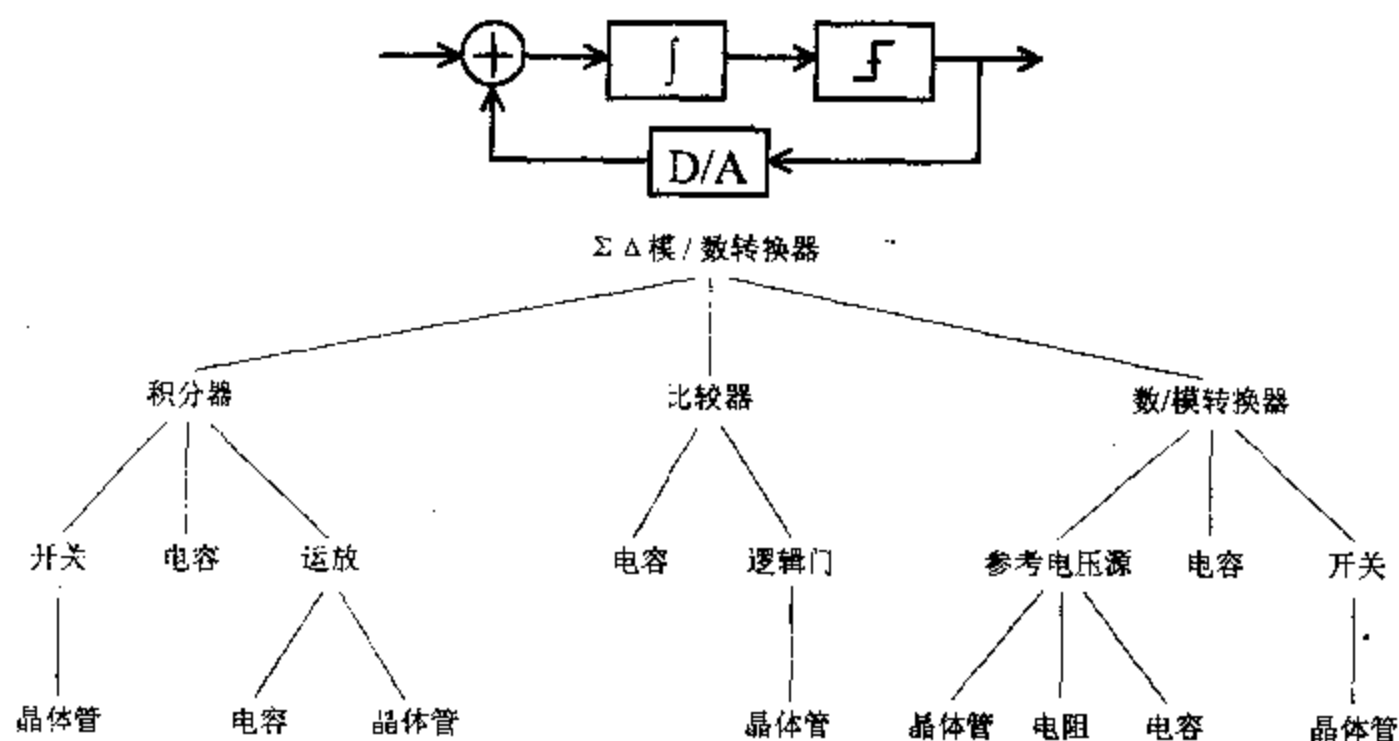


图 1.3 $\Sigma\Delta$ 模/数转换器的设计层次

图 1.3 不难看出,运放和比较器是非常接近的单元(在某些场合甚至是完全等同的),但却同时位于不同设计层中;而相同的功能也可以很容易用别的设计层的元件来置换,例

如放大信号的功能既可以用单个晶体管来实现,也可以用一个包含多个晶体管的运算放大器来实现。在行为描述方面,不同设计层之间也有极强的关联性。例如:电路的频率响应特性在所有设计层中都是十分重要的指标,而且,只有通过更低一层、甚至直到最底层的模块才能确定电路的频率响应,从而使得诸如频率响应、噪声等性能指标需要从最高层一直传递到最底层。造成这一“混乱”局面的根本原因在于缺乏严格的模拟集成电路设计层次的定义,其设计层的划分仅仅是一种形式上的基于模块结构的简单分解,而不像数字集成电路那样,设计层次的划分基于信号类型及其时域表象的本质特征。另外,模拟集成电路设计层次不明晰还表现在电压、电流和阻抗等变量同时出现在各个层次中。

应当看到,尽管模拟集成电路设计层次不明晰的问题在目前尚没有立竿见影的解决方法,但是,正如我们在本书后续章节所见,层次式的设计方法依然对模拟集成电路设计大有裨益^[8]。

2. 性能指标繁杂

描述模拟集成电路行为的性能指标非常多,以运算放大器为例,其性能指标包括功耗,低频增益,摆率,带宽,单位增益频率,相位余度,输入、输出阻抗,输入、输出动态范围,共模信号输入范围,建立时间,电源信号抑制比,共模信号抑制比,失调电压,噪声,谐波失真等数十项^[9];而且很难给出其完整的性能指标,例如运放的上述指标都还可以加入随温度变化的特性。另外,某些指标所涉及的范围还特别大,例如:对于生物医学应用而言,运放的增益大约定义在 1kHz 范围,但对于通信应用而言,运放的增益则需要定义在 100MHz 甚至更宽的范围。

在给定一组性能指标的条件下,通常可能有多个模拟电路符合所有指标,但对每一个符合指标的电路而言,它们仅仅是在一定范围内、对个别指标而言是最佳的,没有任何电路对所有指标、在所有范围内都是最佳的;更为严重的是常常存在相互矛盾的性能指标。因此,模拟集成电路设计者只能采用一些折衷方案。运放设计的典型问题就是对增益和带宽作折衷处理,这种折衷处理通常还会影响噪声、面积等性能指标。由于对性能指标的折衷处理是很复杂、很困难的多维问题,因而模拟集成电路设计者在处理这类问题时通常靠其直觉和长期积累的设计经验,即采用一些简便的经验法则。

3. 拓扑结构层出不穷

任何数字电路都是由逻辑门之类的单元所组成的,这些单元的功能单一、结构规范;而模拟电路则不然,没有规范的模拟单元可以重复利用。对于同样的功能,人们已经构造出了成百上千种电路拓扑结构,运算放大器、比较器和模拟乘法器就是如此。这些拓扑结构是有针对性的,每一个拓扑结构虽然都有其长处,但也有不足,都只能在一定范围内适合于个别或部分性能指标的要求。例如:对于较小的增益指标,用单级结构的运放就足够了;对于较大的增益指标,通常需要采用两级甚至多级结构的运放;对于较高的单位增益频率指标,采用折叠式共源共栅结构更为合适。

由于模拟集成电路正朝着高频、高精度、低噪声、低失真和低功耗等方向发展,为了克服现有拓扑的不足,越来越多的新拓扑将如雨后春笋不断涌现。对于特定的应用需求,模拟集成电路设计者可选择的拓扑结构非常之多,因此,对于给定功能的模拟集成

电路综合而言,如何自动确定最佳的电路拓扑结构就变得十分棘手。

4. 器件尺寸至关重要

在数字集成电路中,晶体管的数目虽然非常多,但绝大部分工作于开关状态,因而采用最小尺寸即可,只有那些处于关键延时路径上或需要驱动较重负载的晶体管才需要较大的尺寸。与此相反,模拟电路的晶体管数目虽然较少,但几乎每一个晶体管的尺寸均与电路性能有密切的关系^[8]。例如:运放的相位余度与电路中的所有电容都有关系,从而也就与所有晶体管的尺寸相关。这使得模拟集成电路设计者必须精心地设计电路中每一个晶体管的尺寸。由于 MOS 模拟电路中各 MOS 场效应管的沟道宽长比可能相差 1 000 倍甚至 10 000 倍,因此,尽管模拟电路所包含的器件数目相对较少,但确定(优化)所有器件的尺寸却极其费时、费力,这是导致数模混合型 ASIC 中模拟电路的设计时间远远超过数字电路设计时间的主要原因之一。

5. 受工艺涨落的影响极大

模拟集成电路的性能不仅与集成电路制造工艺密切相关,而且还受工作环境的影响。设计过程中必须充分考虑偏置条件和温度的变化、工艺涨落及寄生参数等对电路性能影响很大的因素,否则,它们会降低电路性能指标,甚至改变电路功能。例如:器件尺寸的失配将对运放的失调电压、共模信号抑制比和电源信号抑制比产生严重的影响,而衬底温度的变化也会改变运放的频率响应。模拟集成电路设计必须考虑工艺水平或器件之间的匹配精度,因为这是决定选用何种电路拓扑结构的重要因素。类似地,模拟集成电路设计还需要非常精确的器件模型。例如:在设计宽带 MOS 运算放大器时,就需要 MOS 器件模型的栅电阻和栅电容非常精确才行。

与数字集成电路的版图设计形成鲜明的对比,模拟集成电路的版图设计将不仅仅是关心如何获得最小的芯片面积,还必须精心设计匹配器件的对称性、细心处理因连线所产生的各种寄生效应。

6. 对系统级串扰十分敏感

模拟集成电路对系统级的串扰十分敏感,最典型的系统级串扰要数热反馈。例如:由于输出级的功率一般都比较,是芯片上的热点,它们将使芯片的不同部位出现温差,影响电路中那些温度敏感部分的性能^[10],所产生的性能波动又会传递回输出级并影响输出级的功耗,从而形成热反馈环。如果该反馈环中存在某种非线性,就会产生失真。类似地,如果某芯片上集成了多路数据采集系统(实际应用中非常普遍),则设计这类系统时必须采取有效措施以防止各路间的串扰。串扰问题在数模混合 ASIC 中更为突出^[11]:这种串扰主要来自数字部分和模拟部分的公共电源线、芯片的衬底,数字部分的开关切换将使电源信号出现毛刺并影响模拟部分的工作,当然,也可能通过衬底的耦合作用波及到模拟部分,从而降低模拟电路的性能。

综上所述,模拟集成电路设计过程中需要综合考虑各项性能指标,合理选择电路拓扑结构,反复确定(优化)器件尺寸,深入考虑工艺涨落、工作环境和各种串扰因素,并精心设计物理版图。因此,模拟集成电路设计是一项复杂、艰巨的工作,只有电路知识广

博、洞察力敏锐和经验丰富的专家才能胜任此项工作。

1.2 模拟集成电路设计自动化及其任务

模拟集成电路的传统手工设计方式效率极低,无法适应微电子工业的迅速发展,只有采用模拟集成电路的自动化设计系统才能降低设计成本、适应日益激烈的市场竞争。目前,关于模拟集成电路设计自动化的定义尚有歧义,本书采用下述比较通行的定义。

模拟集成电路设计自动化是指利用计算机来自动完成从模拟集成电路的系统级行为描述到生成物理版图之间的各项设计工作。例如:自动选择电路拓扑结构、优化器件尺寸和完成物理版图设计等。

1.2.1 模拟集成电路设计自动化的层次式模型

由于模拟系统或数模混合系统的设计是一项非常复杂的工作,实现从行为描述到物理版图的设计自动化是相当艰巨的,想一口气完成显然是不现实的,因此需要采用各个击破的策略:先将自动化设计任务分割为若干子任务,子任务又进一步分解为子子任务,……,然后逐个实现各项子任务的自动化设计,从而圆满实现模拟集成电路的设计自动化。这种各个击破的策略通常称为层次式自动化设计。图 1.4 是目前普遍采用的一种层次式模型,包括:

(1) 系统级:又称芯片级,电子系统(或芯片)是该设计层的基本元件,如数据采集芯片、调制解调芯片、MPEG2 解码系统等;

(2) 功能级:常规的模拟集成电路功能模块是该层的基本元件,如滤波器、A/D、D/A、PLL、PWM 等;

(3) 电路级:常见的模拟电路单元是该设计层的基本元件,如运算放大器、比较器、模拟相乘器、鉴频器、鉴相器、振荡器、偏置电路、参考电压源等;

(4) 器件级:集成电路元器件是该层的基本单元,如晶体管、电阻、电容等。

值得特别一提的是:由于模拟集成电路设计本身具有“层次不明晰”的特点,上述关于模拟集成电路设计自动化的层次划分也不够清晰,但不论如何,层次式设计自动化模式

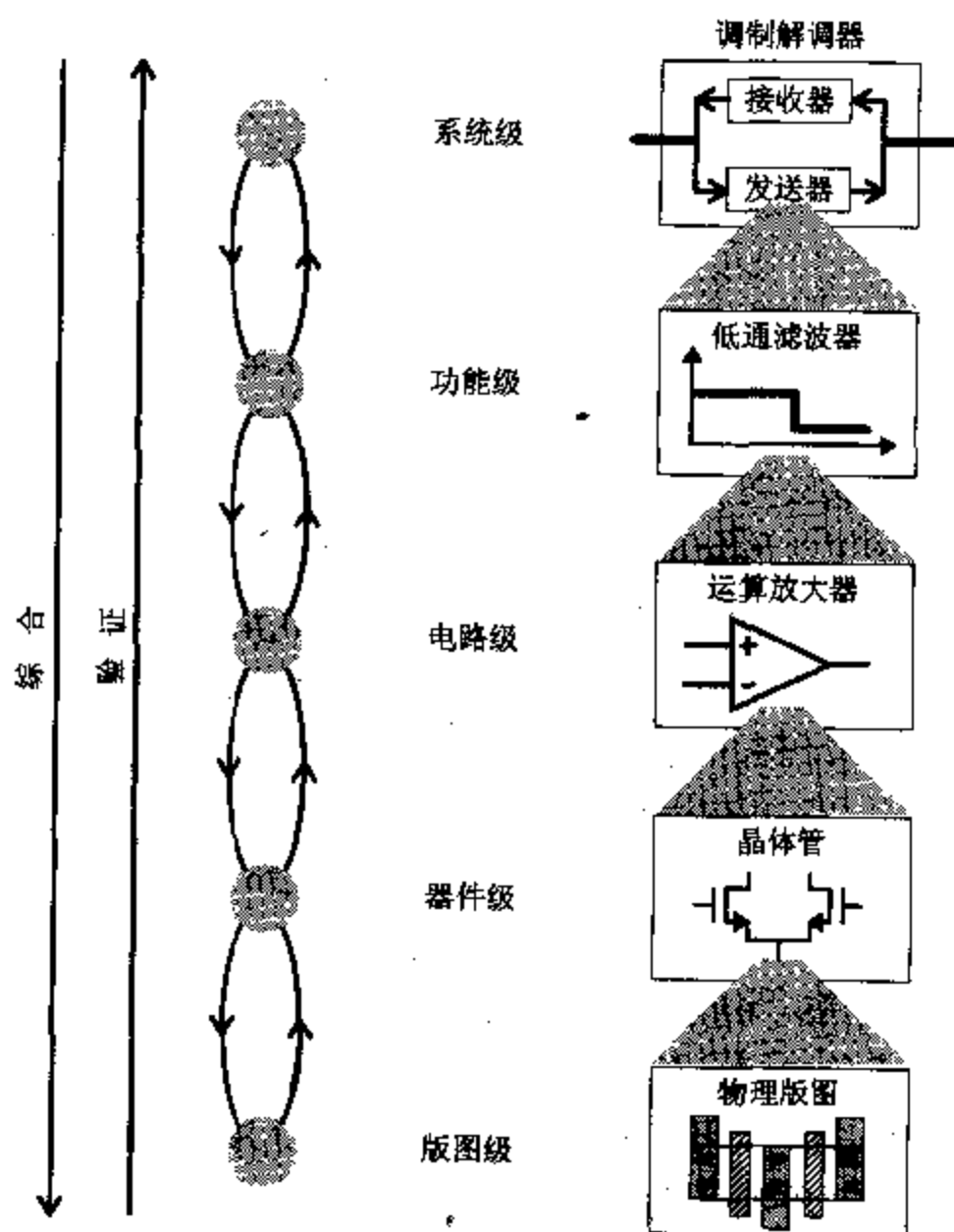


图 1.4 模拟集成电路设计自动化的层次式模型

对于解决模拟集成电路设计这一难题依然有十分重要的作用。为便于加深读者对模拟集成电路设计自动化层次的理解,图 1.4 给出了调制解调器的层次式自动化设计流程。

图 1.4 中使用了综合和验证这两个概念,它们是层次式设计自动化过程中常用的术语。本书中这两个术语的含义为

综合是将定义于某层次的系统映射到较低层次的过程。

验证是证实用某层次的元件所实现的子系统与较高层次的描述具有相同行为的过程。

1.2.2 模拟集成电路的表象与模块生成

模拟集成电路的表现形式称为表象。本书借鉴数字集成电路综合的成就,采用行为表象、结构表象和几何表象^[1]等三种不同的形式来刻画模拟集成电路,图 1.5 是运算放大器的三种表象:

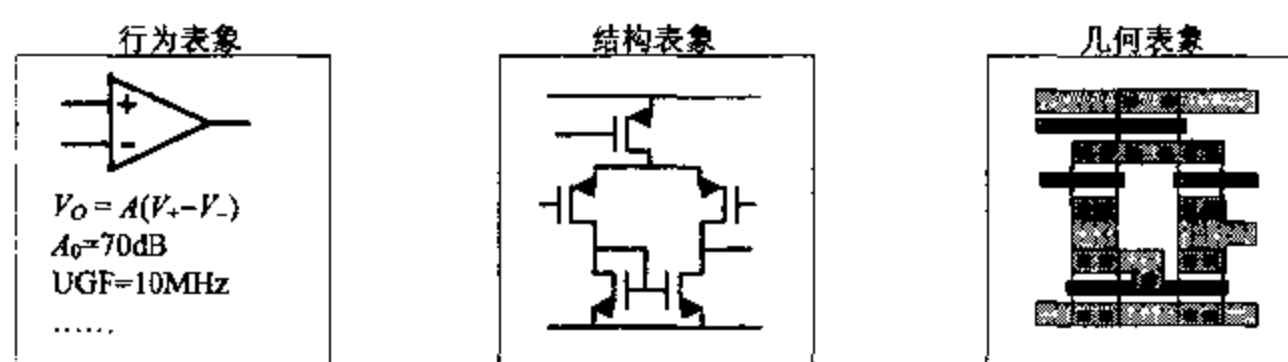


图 1.5 运算放大器的行为表象、结构表象和几何表象

(1) 行为表象:刻画一个模块(或系统)的行为(或功能),即定义模块做什么,但不包括怎么做。行为表象除定义模块的功能外,通常还给出其性能指标及工作环境。随着微电子工业的发展,模拟集成电路的硬件描述语言将成为刻画行为表象的主要手段^[12]。

(2) 结构表象:用较低层次的单元来表示一个模块(或系统)是如何实现的,即定义如何完成该模块的功能。低层次的单元及其联接关系是构成结构表象的两大要素,结构表象又包含多个层次,最底层次是晶体管、电阻、电容等集成电路中的器件。

(3) 几何表象:实际上是一个模块(或系统)的物理版图,定义了制造该系统的几何参数。几何表象总是与具体的集成电路工艺线相关的,而 CIF、GDS 和 EDIF 则是描述几何表象的主要文件格式。

模拟集成电路设计的过程就是将各模块从其行为表象映射到结构表象,再映射到几何表象的综合过程和与之反向的验证过程;而模块生成的任务则是实现这些过程的自动化。在模拟集成电路综合领域,将行为表象映射为结构表象的过程称为电路综合,而将结构表象映射为几何表象的过程则称为版图综合。对于模拟集成电路而言,其综合问题比数字集成电路复杂得多,因为几乎每一个器件的尺寸都对其性能有影响。因此,对于模拟集成电路而言,电路综合通常又分为拓扑选择(或拓扑生成)和确定器件尺寸(或尺寸优化)两个步骤。拓扑选择主要是根据行为描述,从拓扑库中选出一个能达到所需功能的电路结构;而尺寸优化则是采用优化算法确定选定拓扑中各器件的几何尺寸,使电路性能满足行为描述的要求,且面积尽可能小。特别地,当电路不太复杂(例如模拟集成电路单元的设计)时,最好同时进行拓扑选择和尺寸优化^[3]。为便于读者形象、直观地了解

拓扑选择、版图综合、结构表象等术语的含义及其相互关系,图 1.6 给出了模拟集成电路模块生成的具体过程。

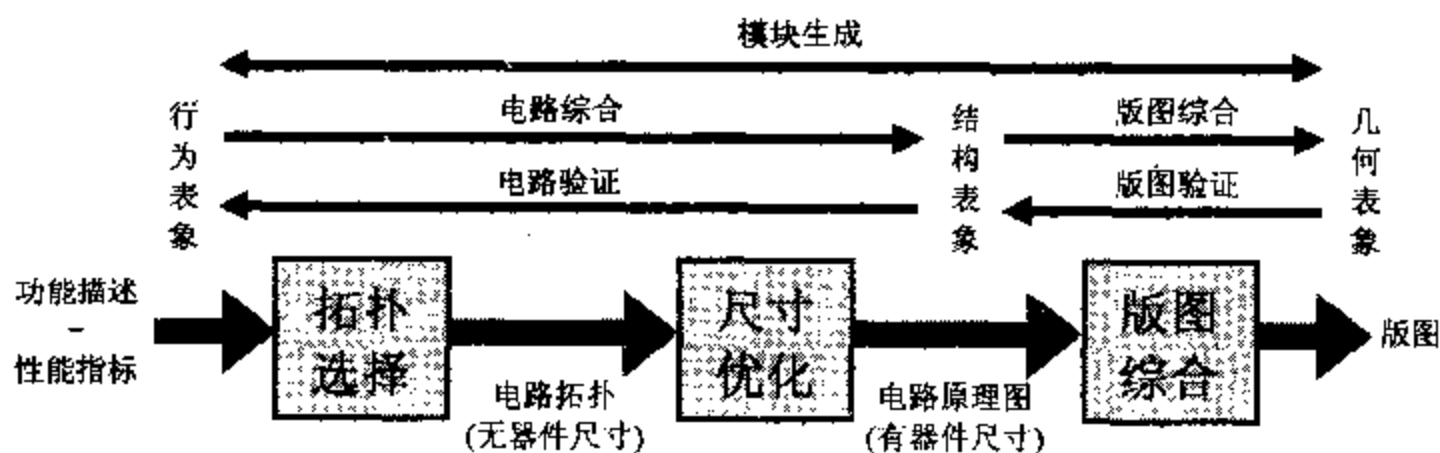


图 1.6 模拟集成电路的模块生成

1.2.3 模拟集成电路设计自动化系统的任务及必备条件

为解决模拟集成电路设计自动化这一难题,人们已经对模拟集成电路设计自动化方法和系统进行了多年的研究^[2-6]。开发模拟集成电路设计自动化系统的目的不外乎降低模拟集成电路的设计成本,具体表现如下:

- (1) 缩短从行为描述到形成物理版图的模拟集成电路的设计时间,确保该模拟集成电路在制造过程中是可测的,确保最终所获得的芯片能正常工作;
- (2) 将电路设计者从各种繁冗工作中解脱出来,使之有更多的精力专注于设计过程中那些富于创造性的工作;
- (3) 为系统设计者自动设计出满足给定性能指标的模拟电路;
- (4) 收集、整理模拟集成电路设计专家们现有的知识和经验,并将它们集成到 CAD 系统中,形成模拟集成电路设计的知识库,为日后使用服务;
- (5) 增长 CAD 工具使用者的见识,将系统内部的知识传授给那些初次设计模拟集成电路的人,缩短其学习时间。

为了实现上述目标,一个模拟集成电路自动化设计系统必须能够胜任模拟集成电路设计过程中的各项任务,能适应不同的设计模式和不同的设计者。更具体一些,模拟集成电路自动化设计系统应当具有下述必备条件:

- (1) 能支持层次式的模拟集成电路设计,这是实现复杂模拟集成电路(系统)的必由之路。
- (2) 对给定的拓扑结构,它所覆盖的电路性能毕竟是非常有限的,而不同应用背景所需的电路性能常常相距甚远,因此,模拟集成电路自动化设计系统不能仅仅局限于一些固定的电路拓扑,必须让设计者能够非常方便地将自己的电路拓扑加入到系统中。
- (3) 在通常情况下,模拟集成电路设计需要综合考虑多项性能指标,而且与设计者的知识和经验密切相关。但十分遗憾的是如何收集整理模拟集成电路设计专家们的智慧和经验在目前仍是一大难题,也正是因为缺乏结构化、公式化的模拟集成电路设计知识,才造成目前模拟集成电路设计成本极其昂贵的局面。另外,由于模拟集成电路设计者一般都没有程序设计的经验,也不清楚一个模拟集成电路设计自动化系统是如何实现的,他们即使有很好的设计经验,也没有能力将设计经验加入到系统中。因此,模拟集成

电路设计自动化系统应当具有“自学”能力,能从一个设计者那里学习一些设计经验,并传授给使用该系统的别的设计者。

(4) 由于在不同的应用场合中,电路性能和允许的设计时间可能大相径庭,针对具体情况采用恰当的设计模式是最明智的选择。例如:设计高性能的模拟集成电路时,由于各项性能指标都非常苛刻,通常需要采用全定制的设计方法;但对于所有性能指标都不太高的情形,则可以采用模拟电路标准单元或阵列来实现,同时,还希望能利用模拟集成电路自动化设计工具,获得能用双极工艺、CMOS 工艺或 BiCMOS 工艺来实现的模拟电路。因此,能够适应不同设计模式的需要是模拟集成电路自动化设计系统的又一必备条件。

(5) 能适应不同设计者的需要。通常需要考虑系统设计者和电路设计者这两类用户。对于系统设计者而言,由于他们不熟悉、也不关心电路实现的细节,他们期望能在最短的时间内自动设计出满足所有性能指标的模拟电路,他们所依赖的是模拟集成电路自动化设计系统所包含的设计经验。但对于电路设计者而言,情况就有所不同了,他们主要依赖自己的设计经验,并需要从行为描述开始,直接设计到每一个晶体管甚至物理版图。电路设计者的工作可分为两大部分,一部分极富创造性(譬如设计新的拓扑结构),另一部分则是费时、费力且琐碎的重复性工作(譬如优化器件尺寸、验证电路性能等)^[3,4]。模拟集成电路自动化设计系统不仅需要把电路设计者从琐碎工作中解脱出来,而且还应当允许他们将创造出的电路拓扑和新获得的设计经验加入到系统中。

1.3 模拟集成电路及其设计自动化的重要性

众所周知,数字电路具有抗干扰力强、集成度高、容易设计等优越性,随着数字信号处理技术的迅速发展,最初由模拟电路实现的功能(如乘法、滤波)大部分都可以用数字电路来实现,只不过所需的晶体管数目多一些而已,因此,当集成电路技术发展到 80 年代中后期、利用先进的微电子制造技术在一个芯片上集成数百万个晶体管已是游刃有余时,电子系统中的模拟电路所占的比例不断萎缩,而且,只要有可能,就用数字电路来取代相应的模拟电路^[13]。尽管 80 年代中后期从集成电路领域所掀起的数字化浪潮在今天仍然有增无减,但是,数字电路不论如何也不可能完全取代模拟电路。这是因为我们生活在一个模拟世界之中,人们所听、所见、所接触、所感受的都是模拟量,因而不可避免地会面对、当然也需要处理各种模拟量。离开了模拟电路,数字电路将无法从模拟世界获取被处理的数据,也不能将其处理结果转换为自然世界所能识别的物理量,因而也就不能控制自然世界。事实上,模拟集成电路近年来反而在工业界重新获得了广泛关注^[14-16],这一方面是因为专用集成电路正朝着系统集成的方向迅速发展^[17],另一方面,在许多需要高性能模拟集成电路的应用领域中,用数字集成电路是很难实现甚至不可能实现的^[13]。

更具体一些,模拟集成电路在下述领域的应用将是数字集成电路无法与之抗衡或根本就实现不了的:

(1) 接口电路。自然世界中的各种信号(也称为物理量,如语音、音乐、人体的电信号)在本质上都是模拟信号,在时间轴和幅度轴上都是连续的,因而任何与自然世界有

联系的电子系统都毫不例外地需要接口电路——其前端至少包含一个模/数转换器(A/D)以采集物理量,其后端也至少包含一个数/模转换器(D/A)以驱动或控制自然世界的实际系统。对一个完整的电子系统或集成电路的设计者而言,关于模拟电路或数/模接口部分的性能指标往往最具挑战性^[18],因为模拟电路常常是决定整个系统性能的关键因素。例如:对于为CD唱机设计的18位过采样D/A而言,其关键环节不在于其中的数字电路,而在于其中的模拟输出滤波器,因为该滤波器所产生的噪声和失真决定了整个系统所能达到的分辨率。

(2) 在非接口电路中,虽然用数字集成电路非常容易实现一些通常的信号处理功能,但在一些高速、高性能的应用领域,模拟集成电路在面积、功耗和整体性能方面都优于数字集成电路^[19],这些领域主要包括:

- ① 高频应用领域(包括GHz范围的双极型集成电路),如电视、蜂窝移动电话等;
- ② 低噪声的数据采集系统,如生物医学传感器^[20]、可植入生物“器官”^[14,15]、用于核研究的微粒/辐射检测器^[21]等;
- ③ 并行模拟信号处理,将用模拟信号处理单元所构成的神经网络集成到一个芯片上^[22,23]就是一个最典型的例子;
- ④ 功率电子学,如各种大功率的驱动设备、脉冲宽度控制器等。

由此可见,在高速、高性能、大功率等应用领域,模拟集成电路有其巨大的优势,但是,由于模拟集成电路的设计十分困难,同时又缺乏有力的模拟集成电路CAD工具,因此,在相当长的时期内,集成电路设计者总是极力避开模拟电路,并尽可能地用易于设计的数字电路来代替模拟电路^[13]。很显然,如果能研究出强有力的模拟集成电路自动化设计工具,模拟集成电路将不再是集成电路设计者所躲避的恶魔,而是令人爱不释手的玉佩;对于给定的应用背景和生产工艺,在用集成电路实现用户所需的功能时,设计者就可以在系统级确定究竟是数字电路合适还是模拟电路更为有效。当然,确定使用数字电路和模拟电路的准则不仅依赖于谁的面积更小、功耗更低、整体性能更优,还必须考虑可测性、可靠性、可控性和可再生性等诸多因素。在这种情况下,设计者就能够对所有性能和成本作适当的折衷,从而为电子系统找到最经济的集成电路实现方案。

由于人们对集成电路设计自动化工具的研究和开发在十余年前主要集中于数字集成电路领域,形成了数字系统及数字集成电路自动化设计技术飞速发展的局面,目前已经出现了许多很成熟的数字集成电路高层次综合工具^[24]和版图自动布局布线工具^[25]。但与此形成鲜明对比,模拟集成电路设计自动化技术的发展却步履蹒跚。实际上,迄今为止的绝大部分模拟集成电路都是通过传统手段(由设计者选定电路拓扑,靠反复模拟或试验来确定器件尺寸)实现的,其设计不仅繁琐、费时,而且一次成功率很低。

现今,随着微电子技术及其应用的迅速发展,人们迫切需要将模拟电路和数字电路集成到同一个ASIC芯片上以提高系统的整体性能^[17],这使得数模混合型系统集成电路将占据绝大部分ASIC市场。最近的研究表明:①在MOS和BiCMOS型ASIC中,数模混合型集成电路的份额在90年代初就已上升到60%,目前仍在继续上升;②一个数模混合型集成电路中,虽然模拟电路部分不足10%,但其设计成本却大大超过另外90%的数字电路的设计成本。由于提高ASIC利润的主要途径是缩短从给定ASIC的性能描述到制造出能工作的ASIC芯片的上市周期,因此,缓慢、低效率的模拟电路设计手段已

经成为制约 ASIC 设计的瓶颈,而采用模拟集成电路设计自动化工具则是冲破此瓶颈的限制、降低 ASIC 的设计成本,并最终提高 ASIC 厂商竞争力的必由之路。

1.4 模拟集成电路设计自动化的进展

过去,模拟集成电路的综合大多是在给定网络结构和电性能指标的条件下确定网络中各元件的最佳参数值^[26~28],这实质属于参数优化的范畴。近十年来出现的参数优化系统有 U. C. Berkeley 的 DELIGHT、SPICE^[29]、ECSTASY^[30], Texas Instruments Inc. 的 OASYS^[31]和清华大学的 YOSIC^[32]等。这些系统均采用通用的电路模拟器(例如 SPICE)来预测电路性能,因而能处理各种模拟集成电路的参数优化问题,但其计算成本十分昂贵,而且,只有在具有较好的初值时才能获得“最优解”。

鉴于基于电路模拟器的参数优化系统无法克服计算量开销大的缺陷,因此,新一代的模拟集成电路综合工具均利用解析公式来预估模拟集成电路的性能^[33]。获取解析公式的途径可分为两大类——手工推导法^[34~37]和符号分析法^[38~42]。

由于模拟集成电路的性能指标五花八门、结构灵活多变,因此,模拟集成电路综合的难点主要在于行为级综合、系统级综合和拓扑选择。比较典型的模拟集成电路综合系统有:

(1) 美国硅编译系统公司(现属 Mentor Graphics 公司)开发的 IDAC 系统初期没有采用层次式设计思想^[37],仅能处理简单的运放和比较器,后来引入了层次式设计思想^[43],能处理的电路也稍微多一些。IDAC 采用下降单纯形法^[44]来优化器件尺寸,但其最大缺陷是只能靠设计者自己选择电路结构,而不能自动进行拓扑选择。

(2) 美国伯克利大学开发的 OPASYN 系统^[45]包含由专家们事先设计好的各种模拟集成电路拓扑。该系统首先按一组启发式策略选定电路拓扑,然后再用数学规划法确定器件尺寸。

(3) 美国通用电气研究实验室开发的 An-Com 系统^[46]采用了层次式功能块构造法,在选择电路拓扑方面与 OPASYN 类似,但其启发式策略比 OPASYN 更复杂。

(4) 比利时鲁汶大学开发的 ARIADNE 系统^[40,47]是一个交互式专家系统,除采用层次式分解技术之外,它还采用了符号分析法获取电路的行为。该系统用模拟退火法^[24]优化器件尺寸,但在拓扑选择方面和 OPASYN 同出一辙。OPASYN, An-Com 和 ARIADNE 的共同缺陷是在没有给定器件尺寸之前很难给出拓扑选择的规则。

(5) 荷兰 Delft 工业大学开发的 Ampdes 系统^[48]为躲避 OPASYN 和 ARIADNE 所遇到的困惑,构造了一个包含 10^6 个拓扑的放大器库,希望能达到“有求必应”的境界,其拓扑选择功能仅比 IDAC 有所改进,能根据设计指标从库中搜索出适当的电路结构。其缺陷是不灵活,且速度慢,对于一般的小问题,需要在小型机上运行约半小时。

(6) 美国卡内基梅隆大学开发的 OASYS 系统^[49]是一个 CMOS 模拟集成电路的编译器,采用了层次式分解技术,在优化器件尺寸阶段发现所生成的拓扑不合适时还可以回溯,能处理的模拟电路类型稍多一些,但综合过程很费时,而且很难给出回溯规则。

(7) 荷兰 Twente 大学开发的 SEAS 系统^[50]与 OASYS^[49]很类似,每当一个被选拓扑的器件尺寸确定之后,SEAS 就按其电性能、被选用的次数等来计算该拓扑的分值。在

SEAS 中, 分值高的拓扑优于分值低的拓扑。SEAS 采用模拟进化法来进行拓扑选择, 但未能很好解决如何自适应地给各被选拓扑赋分值这一核心问题。

(8) 美国 AT&T 公司贝尔实验室开发的 BLADES 系统^[51]集成了 OASYS^[49]的功能, 是以 OPS5 为推理机制的专家系统, 它只能单纯地生成模拟集成电路, 不能进行器件尺寸优化。

(9) 美国南加州大学开发的 CAMP 系统^[52,53]在“生成”初始拓扑的同时就给定初始器件尺寸, 然后再用专家系统的思路对电路拓扑和器件尺寸作迭代改进。当找不到初始设计时, CAMP 就会陷入窘境。

(10) 英国帝国理工学院开发的 ISAID 系统^[54-56]引入了层次式分解和专家系统的思路来生成电路拓扑和确定器件尺寸, 具有 OASYS^[49]、An-Com 和 CAMP 的一些功能, 并用定性推理法(Qualitative Reasoning)对所生成的电路进行修正。ISAID 虽然引入了设计流程管理的概念, 但它的各软件模块之间缺乏有机的衔接, 有些工作还要靠手工完成。ISAID 的主要缺陷在于能生成的拓扑太少, 同时, 还需要用户给出适当的电路形式和对各电性能所施加的权重系数。

这些系统的共同点是局限于单级或两级运算放大器, 即: 只能处理特定的模拟集成电路单元, 而且总是先选择或“生成”一个模拟集成电路拓扑, 然后再优化器件尺寸。这些基于规则、专家知识和推理机制的系统均不能直接处理一般的模拟集成电路综合问题, 其主要原因之一是在未给定器件尺寸之前很难精确估计一个电路拓扑所能达到的各项电性能指标, 因此需要在拓扑选择和器件尺寸优化之间反复回溯。为解决该难题, F. Prosmans 曾尝试过用区间分析法处理拓扑选择问题, 将性能约束转换成线性方程组来求解^[57], 但是, 当性能参数较多时就比较困难了。

在单元级综合方面, P. C. Maulik 提出的“同时进行拓扑选择和器件尺寸优化”的策略^[58,59]是颇有价值的。他用一组二进制向量 Y 来描述电路拓扑, 在求解过程中首先将 Y 的每一个分量看作 $[0, 1]$ 上的连续变量来处理以获取一组初始解; 由于初始解的许多二进制分量的值既不接近 0 又不接近 1, 只好再用分支界定技术以进一步确定这些变量的值, 因而 P. C. Maulik 的方法不能完全地贯彻执行同时进行拓扑选择和器件尺寸优化的策略。当然, 该方法还不能适应层次式分解的设计流程, 也不能胜任结构级综合。

结构级综合的关键是如何将抽象的行为描述转换为具体的与该行为一致的模拟集成电路功能块描述。B. A. A. Antao 在贝尔实验室开发的 Gensim 系统^[60]能自动地将 s 域和 z 域的转移函数转换成数模混合模拟器所能识别的瞬态模型, 而 B. A. A. Antao 关于 AHDL 模型转换方法的研究则更为引人注目^[61], 并已形成模拟集成电路结构级综合的雏形^[62]。S. Somanchi 和 M. L. Manwaring 则从状态方程出发进行模拟集成电路的结构级综合^[63]。

由于模拟集成电路的性能受寄生参数、电源波动和器件之间的失配等因素的影响很大, 因此, 用统计模型^[32,36,64]全面准确地描述模拟集成电路的性能是非常必要的。在电路综合阶段, 除构造电路结构和优化器件尺寸之外, 还应当给出一组约束(例如: 差分对管应尽量对称分布, 且具有相同的外形)以指导物理版图综合^[65]。

在模拟集成电路物理版图综合方面, 最简单的方法是把一些典型电路的布局存入库中, 然后用它们作为初始布局^[45,66], 并根据实际电路的器件尺寸对初始布局作适当修改

就能获得所需的模拟集成电路版图,但该方法缺乏灵活性,因此,采用模拟退火法或各种生成式算法^[67-76]来处理模拟集成电路版图布局问题是当前研究的焦点。在布线时有的采用通道布线技术^[67],但是平面布线法(Area Routing)更适合模拟集成电路^[68]。为了达到用户所要求的性能指标,在物理版图综合阶段还应当考虑芯片的寄生参数^[65,69-71]、热效应^[77]等对芯片性能的影响,并在一定的拓扑约束之下进行版图综合^[65,66,77]。与数字集成电路的版图综合相比,模拟集成电路的版图综合要考虑更多、更细的因素,需要更丰富的设计知识作为支撑。在系统实现时有的将设计知识抽象成规则^[72,73],然后直接用这些规则指导布局布线;有的则用价格函数(Cost Function)表示设计知识和各种约束条件^[69,70],采用模拟退火法来获取最佳版图。卡内基梅隆大学开发的模拟集成电路布局布线系统 KOAN/ANAGRAM II^[69,70]十分灵活,它允许对称布局布线,允许多器件共用源漏极,而且还可以在器件上布线,所综合出的模拟集成电路单元版图已经可以同专家们所设计的版图相媲美。对模拟集成电路而言,在版图综合的最后阶段还需要采用版图压缩技术以进一步减小芯片面积^[75,78]。

1.5 本书的主要内容

到此为止,我们用一章的篇幅介绍了模拟集成电路设计的特点,工业界为什么对模拟集成电路设计自动化工具有十分迫切的需求,模拟集成电路设计自动化系统的目标、任务和最新进展,从而使我们对模拟集成电路综合的作用、研究对象和现状有了具体的认识。

本书将在第二章详细介绍模拟集成电路综合的数学模型。我们将分电路级综合和物理版图综合等两个层次来详细讨论模拟集成电路的自动化设计,将电路级综合的各种问题都抽象为一个混整型非线性数学规划问题,该问题的建立及其求解方法是本章的重点。

用手工推导模拟电路模块的性能表达式,并依据模拟集成电路设计专家们的经验进行简化的方法非常费时,为此,本书第三章介绍了模拟电路的符号分析法^[79],详细讨论了建立电路方程的约简改进节点法、线性方程组的符号求解法以及符号表达式的逼近法。利用符号分析法可以大大减少推导模拟电路性能表达式的时间,极大提高模拟集成电路的自动化设计水平。

本书第四章将介绍模拟集成电路的结构级综合方法,主要讨论状态方程组、 s 域的传输函数和解析函数等三种模拟集成电路的行为描述方式及其相应的综合方法和电路设计技巧。

本书第五章将介绍模拟集成电路的单元级综合方法,除简要介绍基于专家系统和人工智能推理机制的“两步模式”外,还详细介绍了同时确定电路拓扑结构和优化器件尺寸的“一步模式”,并给出了一些具体的单元电路综合实例。

本书第六章将介绍模拟集成电路的物理版图综合方法,详细讨论如何将模拟集成电路设计专家们的版图设计经验融入自动化布局、布线系统中。

古人云:千里之行始于足下。描述模拟集成电路的行为便是实现模拟集成电路设计自动化的首要环节,工业界和学术界都迫切需要一个语法格式统一、使用方便灵活、功

能强大的语言规范。为此，本书第七章将介绍模拟集成电路硬件描述语言的主要特征^[80]。其中的内容主要依据 VHDL-AMS 和 IEEE VHDL1076.1 标准，虽然它们目前还未被工业界接受，但可以肯定，它们即使不能成为未来的工业标准，也一定是未来模拟集成电路硬件描述语言的蓝本，并将对模拟集成电路设计自动化进程的发展起推动作用。

第二章 电路级综合的数学模型

目前模拟集成电路的自动化设计流程一般分为结构级综合(Structural Synthesis)、拓扑选择(Topology Selection)、尺寸优化(Sizing)和物理版图综合(Layout Synthesis)等四个阶段。结构级综合是将用户给定的关于模拟集成电路性能的抽象描述转化为一个用各种功能单元所构成的电路,该电路能实现所要求的电性能;拓扑选择是根据功能单元的性能指标和工作环境,决定用何种具体的电路结构来实现该单元的功能;优化器件尺寸是在获得电路结构的条件下,根据所需的电路性能指标和生产工艺条件确定每一个器件的“最佳”几何尺寸,以提高模拟集成电路的合格率;物理版图综合是将具有器件尺寸和满足一定约束条件的电路原理图映射成一个集成电路版图。图 2.1 直观地刻画了这四个阶段的作用和相互关系。

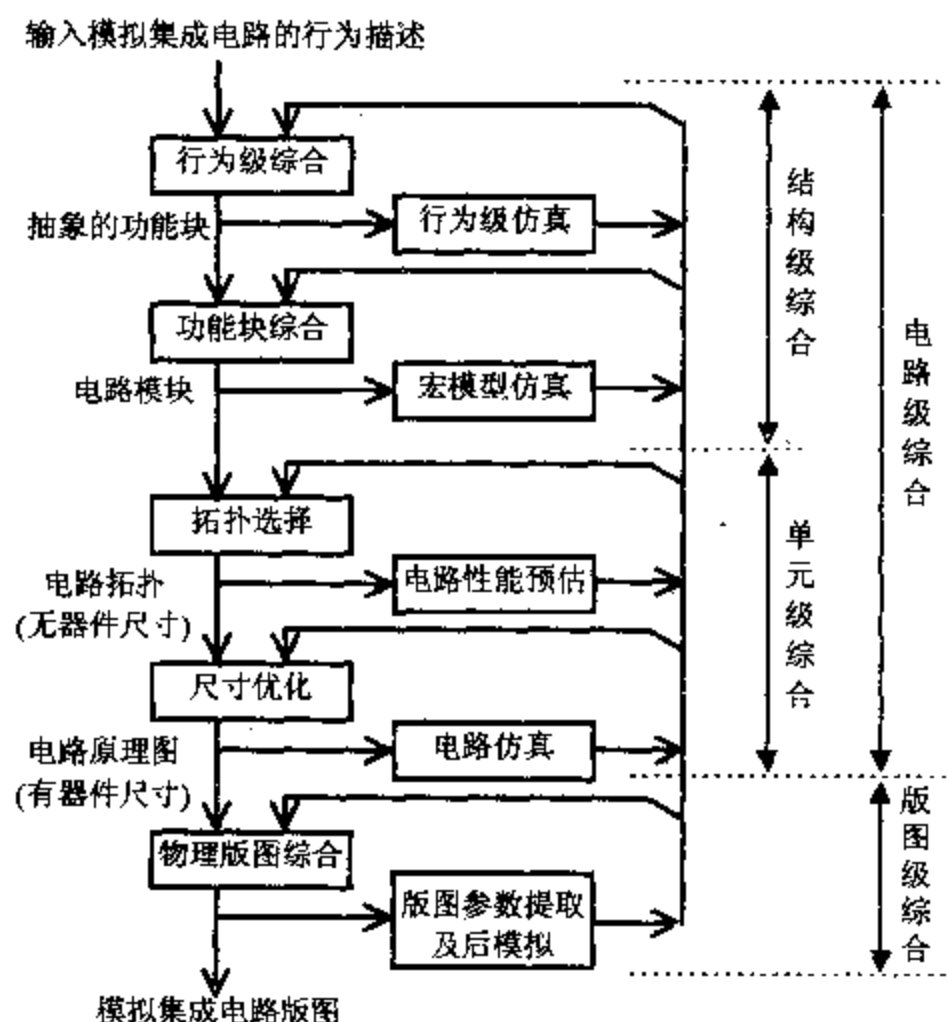


图 2.1 模拟集成电路的自动化设计流程

2.1 电路级综合与混整型非线性数学规划

物理版图综合的数学模型将在第六章中详细介绍,这里仅讨论电路级综合问题。由于电路综合的实质不外乎确定电路结构和优化器件尺寸,因此,我们用二进制向量 Y 表示模拟集成电路的拓扑结构,结构级综合和拓扑选择的主要任务就是求 Y 的最优值 Y^* ;类似地,我们用向量 X 和 σ_x 分别表示电路中各器件的几何尺寸及其标准差,则尺寸优

化的实质就是求 X 的最优值 X^* 。考虑到层次式设计需要,我们用 V 表示电路的工作条件,即偏置电压,假设电路性能为 $P(V, X, Y, \sigma_X)$,用户期望的电路性能指标为 P_E 。因为芯片面积依赖于电路拓扑结构和器件尺寸,我们用函数 $\text{Area}(X, Y)$ 表示,类似地,芯片合格率则用函数 $\text{Yield}(X, Y, \sigma_X)$ 表示。由于电路级综合的任务是在满足用户所给定性能指标的前提下,使芯片面积尽可能小、芯片合格率尽可能高,因此,模拟集成电路综合问题可以形式化地表示为下述混整型非线性数学规划问题:

$$\begin{aligned} \min_{X, Y, V} & \text{Area}(X, Y) / \text{Yield}(X, Y, \sigma_X) \\ \text{s. t. } & P_E - P(V, X, Y, \sigma_X) \leq 0 \end{aligned} \quad (2.1)$$

为叙述方便,我们假设问题(2.1)的最优解为 X^* , Y^* 和 V^* 。

2.1.1 解析算法与通用电路模拟器

问题(2.1)只是一种形式化的表示,为使之能够应用于实际的模拟集成电路综合系统中就必须首先确定计算 $P(V, X, Y, \sigma_X)$ 的方案。早期的系统采用通用电路模拟器来完成这项工作^[29-32],其优越性是能处理各种各样的电路结构,而且当增加新的电路拓扑时,对原模拟集成电路综合系统的改动量很小;其不足之处在于计算量太大,不适合于电路结构或性能指标比较复杂的电路级综合问题。因此,人们目前更趋向于采用解析算法^[33-42]。解析算法的主要优点是计算速度快,而且使电路设计者能全面、准确地把握“当器件尺寸和拓扑结构发生变化时电路性能的变化趋势”,这是电路模拟器及别的数值仿真方法所不能比拟的。

解析算法的关键是推导 $P(V, X, Y, \sigma_X)$ 的解析表达式,作为一个实例,我们考虑图 2.2 所示的简单情况。其中,仅有两个被选拓扑,向量 Y 是一维向量,我们用 y_1 表示, $y_1 = 1$ 表示选择图 2.2(a)所示的共源共栅拓扑, $y_1 = 0$ 表示选择图 2.2(b)所示的简单共源拓扑; V 是由 $V_{B,1}$, I_B 和 V_{DD} 所构成的三维列向量; X 是由 M1, M2 的沟道长度和宽度

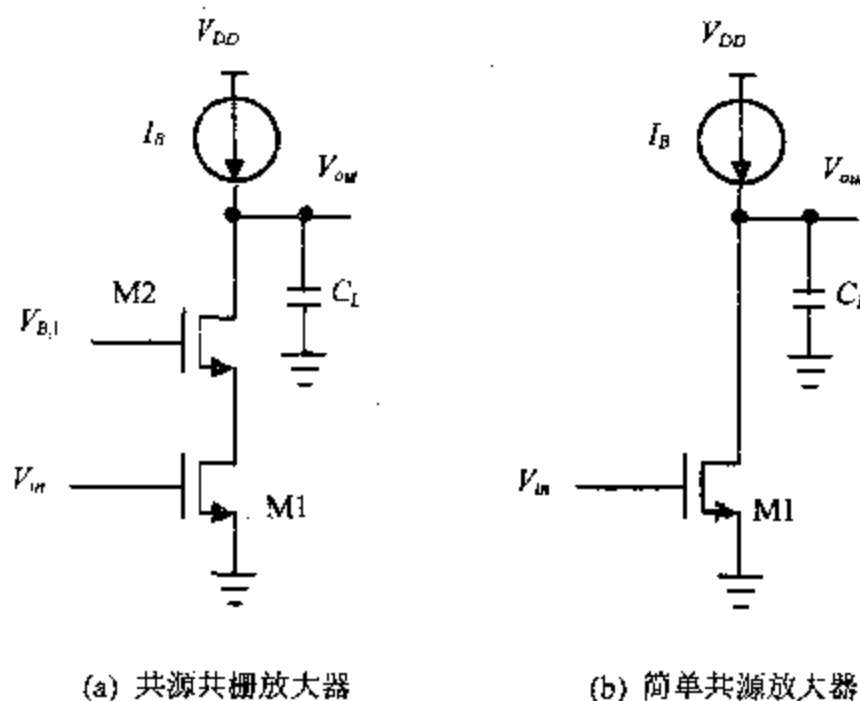


图 2.2 两种不同拓扑结构的放大器

所构成的四维列向量。如果我们只考虑输出电阻 R_{out} , 电压增益 A_V , 摆率 S_R 和单位增益频率 f_u , 则 $P(V, X, Y, \sigma_X)$ 是由这四个电路性能所构成的列向量, 并且很容易导出下述解析表达式:

$$P(V, X, Y, \sigma_X) = \begin{bmatrix} R_{out} \\ A_V \\ S_R \\ f_u \end{bmatrix} = \begin{bmatrix} y_1 R_{out}^C(X, V_{B,1}, V_{DD}, I_B) + (1-y_1) R_{out}^S(X, V_{DD}, I_B) \\ g_{m1}(X, V_{B,1}, V_{DD}, I_B) / (R_{out}^{-1} + sC_L) \\ I_{DS1}(X, V_{B,1}, V_{DD}, I_B) / C_L \\ g_{m1}(X, V_{B,1}, V_{DD}, I_B) / 2\pi C_L \end{bmatrix} \quad (2.2)$$

其中, $R_{out}^C(X, V_{B,1}, V_{DD}, I_B)$ 和 $R_{out}^S(X, V_{DD}, I_B)$ 分别为共源共栅放大器和简单共源放大器的输出电阻, $X = (W_1, L_1, W_2, L_2)^T$, 其中, L_i, W_i, g_{mi} 和 I_{DSi} 则分别代表第 i 个 MOS 场效应晶体管的沟道长度、沟道宽度、跨导和源漏极电流。

可以采用各种等效模型来推导 $R_{out}^C(X, V_{B,1}, V_{DD}, I_B)$ 和 $R_{out}^S(X, V_{DD}, I_B)$ 的解析表达式, 具体方法请参见有关模拟集成电路分析与设计的基础教材。这种基于等效模型的手工推导方法不仅非常费时, 而且容易出错。我们将在第三章介绍如何用计算机来推导这些解析公式。

2.1.2 目标函数

在问题(2.1)所示的形式化描述中, 目标函数被定义为芯片面积与合格率的商, 所反映的是芯片成本。但遗憾的是在一般情况下无法获得合格率 $Yield(X, Y, \sigma_X)$ 的解析表达式, 因此只能通过统计模拟来估计合格率^[32], 这使得问题(2.1)的求解极其费时。考虑到在求解问题(2.1)的初始阶段所获得的模拟电路一般都不满足用户所要求的性能指标, 估计其合格率是毫无意义的, 因此, 目前所有的模拟集成电路综合系统在求解问题(2.1)的过程中均假设 $Yield(X, Y, \sigma_X) = 1$, 只有在完成电路级综合之后才用统计模拟的方法来验证其合格率是否达到预期的水平。在这种假设之下, 电路性能就可以简单地表示为 $P(V, X, Y)$ 。

由于约束规划问题总可以通过引入罚因子将其转换为无约束规划问题^[81]。不失一般性, 我们假设问题(2.1)中有效约束的个数为 η , 则只要引入 η 个罚因子 $\delta_j, j=1, 2, \dots, \eta$, 就可以将问题(2.1)转换为下述无约束混整形非线性数学规划问题:

$$\min_{X, Y, V} \text{Area}(X, Y) + \sum_{j=1}^{\eta} \delta_j (P_{E,j} - P_j) \quad (2.3)$$

其中, 罚因子 δ_j 由下式确定:

$$\delta_j = \begin{cases} 0 & (P_j \geq P_{E,j}) \\ M & (P_j < P_{E,j}) \end{cases}, M \text{ 是很大的正数。}$$

2.1.3 求解混整型非线性数学规划问题的经典算法及其不足

模拟集成电路综合的各个环节都需要求解混整型非线性数学规划问题(2.3), 因此, 求解方法的好坏直接关系到模拟集成电路综合的成败。在求解混整型非线性数学规划问题(2.3)时, 保证全局收敛性是首要的, 因为倘若算法没有全局收敛性, 那么, 即使用户提出的性能指标是合理和可行的, 也会被错误地认为是无法实现的。当然, 我们更希望算法在具有全局收敛性的同时, 其计算量应尽可能小。

非线性数学规划问题的求解方法是比较成熟的^[81], 但混整型非线性数学规划问题则不然, 它属于 NP 完全问题, 其求解是相当困难的。求解混整型非线性数学规划问题的经

典方法主要有分支界定法^[82]、广义 Benders 分解法^[83]和 Outer-Approximation 法^[84]，后两种方法比较适合于设计变量的维数较高的情形，分支界定法虽然比较简单，但只适合于维数较低的情形。这三种方法的共同缺陷是全局收敛性差。用它们去求解问题(2.3)不能同时确定电路结构 Y 和优化器件尺寸 X ，换言之，在还不知道器件尺寸时就对电路拓扑的好坏进行“武断”的评估，由于模拟集成电路的性能与电路中每一个器件的几何尺寸是息息相关的，“武断”评估的恶果不仅是计算效率低，更为严重的是常常会找不到可行解，即满足用户给定的各项性能指标的集成电路。

P. C. Maulik 曾对如何求解问题(2.3)作过一些有益的探索^[58,59]。他将二进制向量 Y 当作 $[0,1]$ 上的连续变量来处理，并提出了一系列将逻辑表达式转化为代数不等式(组)的方法。例如：逻辑表达式“若 $y_1=1$ 或 $y_2=0$ ，则 $y_3=0$ ”就可以用下述不等式组表示：

$$\begin{cases} y_3 \leq 1 - y_1 \\ y_3 \leq y_2 \end{cases}$$

在完成上述处理后，问题(2.3)就退化为非线性数学规划问题，可以采用许多十分成熟的方法或系统来求解。P. C. Maulik 选用 NPSOL 优化包来求解退化后的非线性数学规划问题。很显然，用 NPSOL 所获得的二进制向量 Y 的许多分量都既不是 1 也不是 0。为了解决这种尴尬的局面，P. C. Maulik 只好转而求助于分支界定法，从而不可避免地需要反复求解退化后的非线性数学规划问题。因此，P. C. Maulik 的方法不仅计算量大，而且全局收敛性差；在某些情况下，P. C. Maulik 的方法甚至找不到问题(2.3)的可行解^[58,59]。

2.2 模拟退火法

由于无法用解析法直接获取(2.3)的最优解 X^* 、 Y^* 和 V^* ，因而只好采用迭代改进技术以逐步逼近 X^* 、 Y^* 和 V^* 。确定性迭代改进技术^[44,81]的计算量小，但却很容易收敛于非常糟糕的局部最优解，用这类方法去求解问题(2.3)容易出现找不到可行解的尴尬局面。模拟退火法^[41,85]、模拟进化法^[50,86]和遗传算法^[87]则采用随机迭代改进策略，具有全局收敛性是这类算法的魅力所在，其中又以模拟退火法最为成熟，该方法在数字集成电路的布局方面已取得了引人注目的成就，近年来在模拟集成电路的器件尺寸优化^[41]和单元级版图布局^[69]方面也取得了很大的进步。

模拟退火法最初是为求解下述最优化问题而提出的：

$$\min_Z E(Z) \quad (2.4)$$

其中，向量 Z 代表系统的状态， $E(Z)$ 表示系统处于状态 Z 时的能量函数。

众所周知，处于高温状态的晶体在缓慢冷却(退火)过程中，虽然单个分子的运动是不规则的，但只要退火过程足够慢，则该晶体最终能达到最低的能量状态。模拟退火法的基本思想是使优化迭代改进过程中 Z 的移动规则类似于晶体退火过程中分子的热运动，这样就可以使 Z 收敛于 $E(Z)$ 的全局最优点 Z^* 。

能量函数及其分布、退火策略、初始退火温度、随机移动量的产生、热平衡处理和收敛判据是模拟退火法求解混整型非线性数学规划问题(2.3)的六大要素，下面将逐一讨论这些内容。

1. 能量函数及其分布

用模拟退火法求解混整型非线性数学规划问题(2.3)时,模拟退火法的能量函数就是混整型非线性规划问题的目标函数,即:

$$E(Z) \equiv \text{Area}(X, Y) + \sum_{i=1}^q \delta_i (P_{E_i} - P_i) \quad (2.5)$$

其中, $Z = (X^T, Y^T, V^T)^T$ 。

常用的能量分布函数有 Boltzmann 分布、Fermi-Dirac 分布等。这里我们选择最简单、最常用的 Boltzmann 分布。Boltzmann 分布的物理意义为:在温度为 T 的热平衡系统中,一个分子的能量为 E 的概率 $\text{Prob}(E)$ 与 $e^{-E/kT}$ 成正比,其中, k 为 Boltzmann 常数。因此,在用模拟退火法求解(2.5)的极小值过程中,接受事件“优化变量由 Z_1 移动到 Z_2 ”的概率为:

$$\begin{aligned} \text{Prob}(Z_1 \rightarrow Z_2) &= \min \left\{ 1, \exp \left(-\frac{\Delta E}{kT} \right) \right\} \\ &= \min \left\{ 1, \exp \left(-\frac{E(Z_2) - E(Z_1)}{kT} \right) \right\} \end{aligned} \quad (2.6)$$

由(2.6)不难看出:①当 $E(Z_2) \leq E(Z_1)$ 时,就将 Z_1 移动到 Z_2 ,这与确定性优化法是一致的。②当 $E(Z_2) > E(Z_1)$ 时,确定性优化法拒绝将 Z_1 移动到 Z_2 ,这是造成确定性优化法不能“跳出”局部极小点的根本原因;与此相反,模拟退火法允许按大于零的概率将 Z_1 移动到 Z_2 ,在 Boltzmann 分布假设下,接受能量升高的随机移动的概率为 $\exp \left(-\frac{E(Z_2) - E(Z_1)}{kT} \right)$,这是确保模拟退火法具有全局收敛性的关键。

本书第 2.3 节将介绍关于模拟退火法的简单改进策略,第 2.4 节将介绍多分子模拟退火法,它们在利用上述特点的基础上,较好地解决了原模拟退火法中计算效率和全局收敛性的矛盾。图 2.3 示意性地描绘了模拟退火法与确定性优化算法的本质区别。

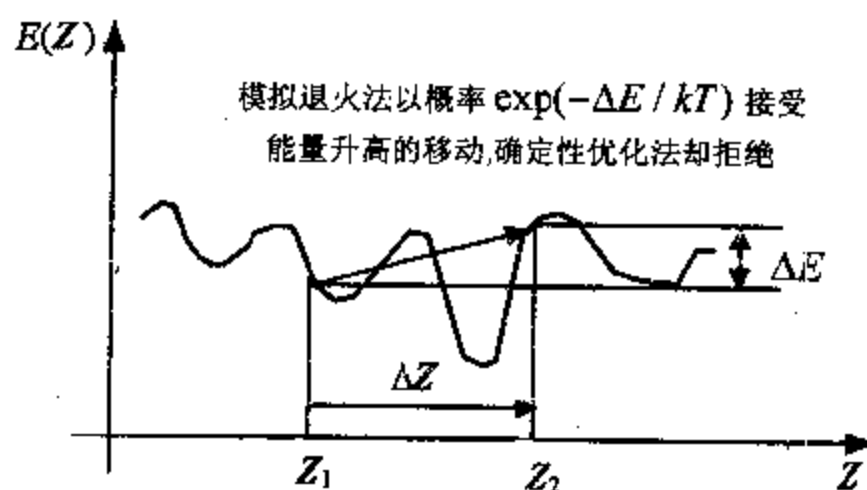


图 2.3 遇到能量升高到移动时,确定性优化法与模拟退火法所持的不同策略

2. 退火策略

退火策略的作用是决定如何控制优化过程中退火温度 T 的变化,它与具体问题是密切相关的,是模拟退火法成败的关键。常用的退火策略有:

(1) 优化变量每移动 m 步之后就将温度 T 减小到 $(1-\zeta)T$, 其中, $\zeta \in (0, 1)$ 。在通

常情况下, m 与优化变量的维数相关, 而 ζ 一般不大于 0.5, 否则将不能确保模拟退火法的全局收敛性; 当然, ζ/m 也不能太小, 否则计算成本会十分昂贵。目前 ζ 和 m 一般通过实验确定。

(2) 优化变量每移动 m 步之后就将温度 T 减小到 $T_0(1-m_c/N_{\max})^\alpha$, 其中, T_0 为初始退火温度, $N_{\max}$ 是总的优化变量移动的次数, m_c 是到目前为止优化变量的累计移动次数, 常数 α 一般取 1, 2 或 4, 若 α 越大, 则模拟退火法在低温时的迭代次数就越多。

(3) 如果将模拟退火法和单纯形法相结合^[44], 则优化变量每移动 m 步之后就将温度 T 减小到 $\max\{\gamma T, \beta(E_s - E_c)\}$, 其中, 常数 $\gamma \in (0, 1)$, β 为 1 左右的常数, E_s 是退火过程中所获得的最优解的能量, E_c 是当前单纯形的各顶点中的最低能量。

在确定退火策略时, 关键是要保证退火过程足够慢, 即: 对于某一退火温度, 必须使分子作足够多次不规则的布朗运动。关于模拟退火法中如何选择退火策略的详细内容请参阅文献[44, 88, 106]。

3. 初始退火温度

初始退火温度的选择一般以能量函数的平均值为依据。随机产生 N 个点 $Z^{(1)}, Z^{(2)}, \dots, Z^{(N)}$, 用这 N 个点求出平均能量变化:

$$\Delta \bar{E} = \frac{1}{N-1} \sum_{j=1}^{N-1} |E(Z^{(j+1)}) - E(Z^{(j)})| \quad (2.7)$$

假设模拟退火法对能量升高的变量移动的初始接受率为 P_0 , 在实际应用中 P_0 一般取比较接近于 1.0 的纯小数, 则初始退火温度 T_0 为:

$$T_0 = -\frac{\Delta \bar{E}}{k \ln P_0} \quad (2.8)$$

4. 随机移动量的产生

在模拟退火法中, 通常使随机移动量 ΔZ 与分子的自由程成正比。从物理学规律可知, 在温度为 T 的热平衡系统中, 分子的平均动能正比于 kT , 而分子的自由程又与其平均动能的平方根成正比。因此, 模拟退火法中随机移动量 ΔZ 应满足:

$$\|\Delta Z\|_2 \propto \sqrt{T}$$

为此, 我们首先产生随机移动方向 Z^R , Z^R 的第 j 个分量是变量 Z_j 的定义域上的均匀分布随机数, 然后只要按下述方式调整 Z^R 的长度即可获得 ΔZ :

$$\Delta Z = L_0 \sqrt{\frac{T}{T_0}} \times \frac{Z^R}{\|Z^R\|_2} \quad (2.9)$$

其中, L_0 是初始步长。

5. 热平衡处理

自然界中的退火过程是一个温度“缓慢降低”的过程, 达到热平衡时, 系统吸收的能量等于系统释放的能量, 系统达到温度 T 的热平衡状态的必要条件是系统的分子之间有足够的机会进行能量交换。在模拟退火法中, “缓慢降低”的含义是指退火温度 T 不能降低得太快, 一般要求 $T_{n+1} > 0.5T_n$, 这是由退火策略^[44, 88]所保证的; 而“达到温度 T 的

热平衡状态”的含义则是指让变量 Z 在温度 T 下作 m 次随机移动的尝试, m 不能太小, 这是确保达到热平衡的关键, 一般按下述方式选择 m :

$$m = n_{\text{walk}} \times (\text{dim}_x + \text{dim}_v) + 5 \times \text{dim}_y \quad (2.10)$$

其中, dim_x , dim_v , dim_y 分别是向量 X , V , Y 的维数, n_{walk} 一般取 2~6 之间的整数。

6. 收敛判定

任何优化算法都必须有相应的收敛判据, 模拟退火法中收敛判据的种类大致可以分为基于退火温度的判据、基于目标函数值的判据、基于目标函数变化量的判据和基于计算量的判据^[44, 106], 其具体形式为:

(1) 基于退火温度的判据

若 $T < T_{\text{stop}}$ 或 $T < \xi T_0$, 则终止退火过程

其中, T_{stop} 一般选 $10^{-4} \sim 10^{-6}$, ξ 一般取 $10^{-5} \sim 10^{-8}$

(2) 基于目标函数值的判据

若 $E(Z_{\min}) < \epsilon$, 则终止退火过程

其中, $Z_{\min}$ 是当前所获得的最优解, ϵ 一般取 $10^{-4} \sim 10^{-6}$

(3) 基于目标函数变化量的判据

若连续 N_c 次 (N_c 一般取 5~10) 降低退火温度后, 最优解所对应的能量均没有明显的变化, 则终止退火过程

```

SimulatedAnnealing_MINLP() {
    var k, T, T0, Y0, X0, V0, Z, delta_Z, E_Z, E_Z_plus_delta_Z, delta_E, U;
    初始化 Boltzmann 常数 k, 计算初始温度 T0;
    T ← T0;
    生成初始拓扑结构 Y0、初始器件参数 X0、初始偏置条件 V0;
    由 Y0, X0 和 V0 组成 Z;
    计算状态 Z 对应的能量 E_Z;
    until (满足收敛条件) {
        until (达到温度 T 的热平衡状态) {
            产生随机移动量 delta_Z;
            计算 Z+delta_Z 对应的能量 E_Z_plus_delta_Z;
            计算能量增量 delta_E = E_Z_plus_delta_Z - E_Z;
            if (delta_E 小于零) { /* 接受移动量 delta_Z */
                Z ← Z + delta_Z;
                E_Z ← E_Z_plus_delta_Z;
            } else {
                产生 (0, 1) 上的均匀分布随机数 U;
                if (U 小于 exp(-delta_E / k / T)) { /* 接受移动量 delta_E */
                    Z ← Z + delta_Z;
                    E_Z ← E_Z_plus_delta_Z;
                } else { /* 拒绝接受移动量 delta_Z */
                    保持 Z 不变;
                }
            }
        } /* until (达到温度 T 的热平衡状态) */
        按退火策略降低温度 T;
    } /* until (满足收敛条件) */
    return Z;
}
    
```

图 2.4 模拟退火法的伪代码

(4) 基于计算量的判据

若退火过程中, 优化变量的移动次数已经达到 $N_{\max}$ 或计算能量函数的总次数超过 $10N_{\max}$, 则终止优化过程

其中, $N_{\max} = \min(10^8, 10^N)$, $N = \dim_x + \dim_y + \dim_r / 8.0$

为减少计算量, 将上述多个收敛判据组合起来也是可行的。为便于读者进行程序设计, 图 2.4 以伪代码的形式给出了用模拟退火法求解混整型非线性数学规划问题(2.3)的算法流程。

2.3 模拟退火法的不足与简单改进策略

在前一节中, 我们首先扼要地介绍了用模拟退火法求解混整型非线性数学规划问题(2.3)的基本思想, 然后又给出了相关的算法流程。本节将首先分析造成模拟退火法计算量大的根源所在, 然后再介绍一些简单的改进策略。

模拟退火法的不足之处是计算量大, 从模拟退火法的算法流程可以看出, 计算量大主要是由下述诸方面因素所造成的:

- (1) 初始退火温度 T_0 很高, 使外层 until 循环的次数很大, 从而计算成本昂贵;
- (2) 温度 T 下降十分缓慢, 这是造成外层 until 循环次数多的又一因素;
- (3) 在给定温度 T 下达到热平衡的速度很慢, 这是造成内层 until 循环次数多的原因, 内层循环次数多是造成计算成本昂贵的又一原因;

(4) 求 ΔE 的时间长, 这主要是由于计算电路性能 $P(V, X, Y)$ 相当慢所造成的, 在模拟集成电路综合过程中用解析公式代替通用电路模拟器的目的就是减少 $P(V, X, Y)$ 的计算量。

在减小计算量的同时必须保证模拟退火法的全局收敛性, 否则将会得不偿失。因此, 我们不能简单地通过减小 T_0 、或增加 T 的下降速度、或加速热平衡过程等手段来减少计算量, 因为这些手段会违背退火过程的物理特征, 严重影响模拟退火法的全局收敛性。

我们在前一节已经指出: “模拟退火法不是简单地拒绝能量升高 ΔE 的移动, 而是以概率 $\exp(-\Delta E/kT)$ 接受这种移动。” 该特征是模拟退火法能跳出局部极小点, 从而收敛于全局极小点的根本原因。因此, 减小计算量的策略应确保模拟退火法仍然具有这一特征。

不难看出, $\exp(-\Delta E/kT)$ 将随着 T 的下降迅速减小, 因此, 在模拟退火过程中, 能量升高的事件主要发生在温度 T 较高的状态; 而在温度 T 较低时, 变量 Z 虽然能随机移动, 但能跳出局部极小点的概率已经非常小了。由此可见, 模拟退火法的全局收敛性是在温度 T 较高时达到的。因此, 减小模拟退火法计算量的基本出发点是有效地限制模拟退火法后期优化变量 Z 的随机移动。

上述对模拟退火法的分析表明, 可以采用下述简单策略使模拟退火法在后期具有确定性优化算法的特征, 从而有效地限制优化变量 Z 的随机移动, 减少模拟退火法的计算量。

(1) 当温度较低 ($T \leq T_c$) 时, 采用确定性优化算法 (例如: 单纯形法或各种拟牛顿法^[44, 81, 106]) 求解。很显然, 如何判定从哪个温度开始采用确定性优化算法 (即选定 T_c) 则

是相当棘手的事情。

(2)可以采用一些新的迭代公式,该迭代公式在温度较高时具有模拟退火法的特征,但在温度较低时却表现出确定性优化算法的特性。我们先按模拟退火法产生 ΔZ 、计算 ΔE ,然后按下述迭代公式计算:

$$\begin{cases} \xi = T/(T_c + T) \\ Y^{(n+1)} = Y^{(n)} + \tau \cdot \Delta Y \\ X_i^{(n+1)} = X_i^{(n)} + \tau \left(\xi \cdot \Delta X_i - (1 - \xi) \lambda_i^{(n)} \frac{\partial E}{\partial X_i} \right) \\ V_i^{(n+1)} = V_i^{(n)} + \tau \left(\xi \cdot \Delta V_i - (1 - \xi) \lambda_i^{(n)} \frac{\partial E}{\partial V_i} \right) \end{cases} \quad (2.11)$$

其中, $T_c \ll T_0$, 它表示确定性优化算法开始起主导作用的温度, n 表示迭代次数, $\lambda_i^{(n)}$ 是确定性优化算法在第 n 次迭代时对第 i 个变量所施加的步长控制因子,

$$\tau = \begin{cases} 1, & (U \leq \exp(-\Delta E/kT)) \\ 0, & \text{其他} \end{cases}, U \text{ 是 } [0, 1] \text{ 上的随机数。}$$

上述两种简单改进策略的差别在于:前者是突然从模拟退火法切换到确定性优化法,而后者则是平滑过渡。这两种策略虽然都能在一定程度上减少计算量,但它们的不足之处在于:若 T_c 过于靠近 T_{stop} , 则虽然对模拟退火法的全局收敛性没有什么影响,但所减少的计算量非常有限;反之,若 T_c 较大,即比较靠近初始退火温度 T_0 , 则虽然可以大幅度减少计算量,但却会冒损失全局收敛性的风险。

2.4 多分子模拟退火法

在求解混整型非线性数学规划问题(2.3)时, P. C. Maulik 的方法由于基于 NPSOL^[90]和分支界定法^[81], 因而不具备全局收敛性;模拟退火法具有很好的全局收敛性,但计算量又太大;前面提出了关于模拟退火法的简单改进策略,它们虽然能在一定程度上减少模拟退火法的计算量,但却冒着牺牲全局收敛性的危险。不过,前面将模拟退火法和确定性优化算法结合起来的思路是可行的,关键是需要科学、合理和切实可行的结合策略。本节将详细介绍一种全局收敛性好、而计算量又比较小的多分子模拟退火法^[3,91,92]。

上一节提出的两种简单改进策略不可避免地会出现不能收敛于全局最优解的情形,因为模拟退火法中分子的移动有或然性,用一个分子的信息来制定何时采用确定性优化法及如何采用确定性优化法的策略是不十分科学、不太可靠的,由此所获得的结论也就会偏离实际情况。从物理学的角度来看,用多个分子($Z^{(n,j)}, j=1,2,\dots,N_F$)模拟一个实际系统将比用单分子更合理些,因为,实际退火过程中的各种宏观物理表象都是系统中所有分子共同作用的结果,单个分子的运动有或然性,而多个分子的总体行为却有必然性。因此,依据多个分子所作出的结论就会更科学、更合理。

很显然, N_F 越大模拟结果也就越“逼真”,但增大 N_F 又必然会增加计算量,为此,一般将 N_F 选为优化变量的个数,即向量 Z 的维数。前一节已介绍过模拟退火法中对热平衡的处理:在单个分子的情形, Z 在温度 T 下作 m 次随机移动的尝试后就认为达到了

温度为 T 的热平衡状态。在多分子的情形，我们仍然在温度 T 下作 m 次随机移动的尝试，其中，每一次尝试只对一个被随机选中的分子进行。这样，多分子的情形并未增加任何计算量，但用它们的平均行为来控制退火过程却更符合自然界的规律，因而，对增加算法的稳定性、提高算法的效率都是大有裨益的。

在热平衡的处理和退火策略方面不能简单照搬原模拟退火法的方案，否则，将和前面所介绍的那两种简单改进策略一样，计算量和全局收敛性将分别成为鱼和熊掌，不可兼得。

在多分子的情形，我们用分子的平均能量作为制定退火策略的依据。由于系统的平均能量为：

$$\bar{E}^{(n)} = \frac{1}{N_P} \sum_{j=1}^{N_P} E(\mathbf{Z}^{(n,j)}) \quad (2.12)$$

因此，每当优化变量移动 m 步或 $\bar{E}^{(n)}$ 下降 ζkT 之后，就应当将退火温度由 T 降低到 $(1-\zeta)T$ ，当 $T < T_c$ 时就采用确定性优化算法。很显然，采用确定性优化算法时选择 $\mathbf{Z}^{(1,j)}$ ， $j=1,2,\dots,N_P$ 中哪一个作为初值是很关键的，不言而喻，所选中的点应当和全局最优解在同一个“山谷”中。遗憾的是全局最优解不可预知，因此，如何确定多分子模拟退火法后期确定性优化算法的初值就显得至关重要。

由于在 $T < T_c$ 的状态下，模拟退火法已经很难跳出局部最优解，同时，模拟退火法又能以概率 1 收敛于全局最优解，因此， $\mathbf{Z}^{(1,j)}$ ， $j=1,2,\dots,N_P$ 中绝大部分的点均与全局最优解位于同一个“山谷”中，图 2.5 是 7 个分子的示意图。

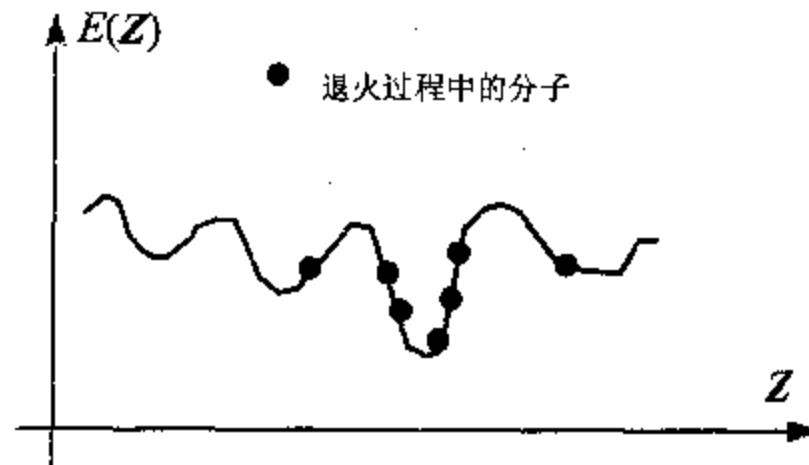


图 2.5 在临界温度时分子的位置分布示意图

由于在临界温度时，绝大多数分子都位于同一个能量谷中，因此，对于某给定的分子，我们可以根据与它邻近的分子的个数来判定该分子是否与全局最优解位于同一个能量“山谷”中。考虑到分子间的邻近程度总是与分子的自由程相关，我们将分子 $\mathbf{Z}^{(n,j)}$ 的秩定义为：

$$\text{Rank}(\mathbf{Z}^{(n,j)}) \equiv \sum_{\substack{k=1 \\ k \neq j}}^{N_P} \theta(\|\mathbf{Z}^{(n,k)} - \mathbf{Z}^{(n,j)}\|) \quad (2.13)$$

其中，函数 θ 定义为：

$$\theta(x) = \begin{cases} 1, & x \leq L_0 \sqrt{\frac{T_c}{T_0}} \\ 0, & \text{其他} \end{cases} \quad (2.14)$$

很显然, 在所有分子 $Z^{(1,j)}$, $j=1, 2, \dots, N_p$ 中, 秩最大的分子一定和全局最优解位于同一个“山谷”中, 将它选作多分子模拟退火法后期确定性优化算法的初值, 就可以确保多分子模拟退火法的全局收敛性, 因为, 后期的确定性优化算法不会跳出该“山谷”, 而且总可以收敛于该“山谷”的谷底。

需要特别指出的是: 由于引入了分子的秩这一概念和上述判定全局最优解所在能量谷的方法, 已没有必要精确选定临界温度 T_c , 因为一旦退火温度降低 3 至 4 个数量级之后, 大部分分子就已经位于全局最优解所在的“山谷”中。因此, 把比初始温度低 5 个数量级的温度定义为临界温度就既具有很好的全局收敛性, 又可以较大幅度地减少计算量。

2.5 算 例

下面将通过几个算例来说明多分子模拟退火法既能保持原模拟退火法的全局收敛性, 又有很高的效率。

例 2.1 求三驼峰函数(3 Humpback Camel Function)^[93]的全局极小点:

$$\min_{Z_1, Z_2} E(Z) = 2Z_1^2 - Z_1Z_2 + Z_2^2 - 1.05Z_1 + \frac{Z_1^6}{6}, \quad Z = (Z_1, Z_2)^T \quad (2.15)$$

图 2.6 是三驼峰函数的图像。

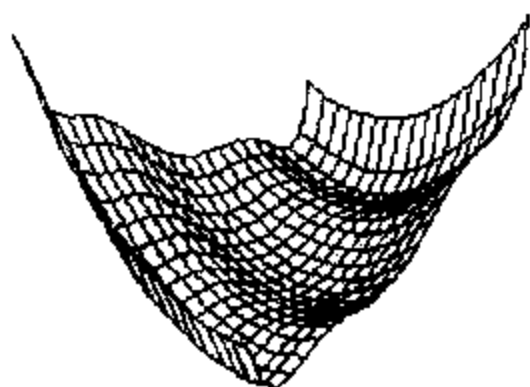


图 2.6 三驼峰函数的图像

很容易证明: ①三驼峰函数有两个鞍点, $E(-1.075423, -0.5377115) = 0.877309$ 和 $E(1.075423, 0.5377115) = 0.877309$; ②三驼峰函数有三个极小值点, $E(-1.7475523, -0.87377615) = 0.298638$, $E(1.7475523, 0.87377615) = 0.298638$ 和 $E(0, 0) = 0$, 其中, $(0, 0)$ 是全局极小值点。

为详细揭示原模拟退火法的缺陷, 表 2.1 给出了用原模拟退火法(C 语言源程序来自于文献[44])求解问题(2.3)的过程和相应的最优解。从表 2.1 不难看出: 模拟退火法后期的迭代改进(第 1 162 次~第 2 106 次)花费了许多不必要的时间, 所获得的最优解为 $E(0.000135, -0.000151) = 7.9822797 \times 10^{-8}$ 。我们采用多分子模拟退火法后, 总共只需 1 067 次迭代(计算成本差不多减少了一半), 所获得的最优解为 $E(5.91424 \times 10^{-7}, 1.42782 \times 10^{-6}) = 1.8938 \times 10^{-12}$, 这是全局最优解。该算例表明: 多分子模拟退火法仍然具有全局收敛性, 而且能有效地减小原模拟退火法的计算量。

表 2.1 用模拟退火法求三驼峰函数的全局极小值的过程

迭代次数	退火温度 T	Z_1	Z_2	$E(Z)$	目前的最低能量
0	1.0E+05	-2.0	2.0	9.8666667E+00	9.8666667E+00
22	8.0E+04	-3.75000	1.00000	2.887912E+02	5.8281249E-01
583	2.418E+02	-1.63763	-0.06053	9.3109036E-01	3.9663921E-01
604	1.934E+02	1.01143	0.08188	1.0494627E+00	3.9663175E-01
733	5.071E+01	0.40601	1.82233	2.8829050E+00	3.4002987E-01
777	3.245E+01	0.04653	-2.08335	4.4416104E+00	3.4002885E-01
926	6.806E+00	1.37440	-0.89161	7.4296224E-01	3.2753792E-01
948	5.445E+00	0.22344	0.63732	3.6102842E-01	1.1816269E-01
1 162	5.846E-01	-1.41843	-1.77259	1.7587184E+00	1.0399865E-03
1 183	4.677E-01	-0.44945	-0.02698	3.5113974E-01	3.0127382E-03
1 293	1.532E-01	0.14931	-0.13338	8.1772135E-02	3.4039386E-04
1 549	1.053E-02	-0.00684	0.06017	4.1255604E-03	1.5875282E-04
1 570	8.425E-03	0.01776	0.01098	5.5628630E-04	1.5875265E-04
1 699	2.209E-03	0.00440	0.02116	3.9336117E-04	4.9256851E-05
1 763	1.131E-03	-0.01276	0.00125	3.4311988E-04	2.5750622E-05
1 827	5.790E-04	0.00089	-0.00243	9.6518001E-05	1.8940944E-05
1 913	2.371E-04	0.00633	-0.00289	1.0678191E-04	1.4103841E-06
2 106	3.183E-05	0.00007	-0.00061	4.2459999E-07	7.9635996E-08
全局极小点		0.000135	-0.000151		7.9635996E-08

例 2.2 求 Kowalik 问题^[94]的最优解:

$$\min_Z E(Z) = \sum_{i=1}^{11} \left(a_i - \frac{Z_1(1.0 + b_i Z_2)}{1.0 + (Z_3 + b_i Z_4)b_i} \right)^2 \quad (2.16)$$

其中, $Z_j \in [-2.0, 2.0], j=1, 2, 3, 4$, 常数 a_i, b_i 由表 2.2 给出。

该函数有多个局部极小值点。多分子模拟退火法经过约 49 000 次迭代后能获得全局最优解:

$$E(0.192873, 0.186018, 0.120225, 0.133833) = 3.07557 \times 10^{-4}$$

其计算量仅为原模拟退火法(C 语言源程序来自于文献[44])的 65%。对于不同的随机数序列, 多分子模拟退火法虽然有时会收敛于局部极小点(原模拟退火法也是如此), 但所获得的目标函数值均小于 8.2×10^{-4} 。

表 2.2 Kowalik 问题中常数 a_i, b_i 的值

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0.1957	0.1947	0.1735	0.1600	0.0844	0.0627	0.0456	0.0342	0.0323	0.0235	0.0246
b_i	0.25	0.5	1.0	2.0	4.0	6.0	8.0	10	12	14	16

例 2.3 求下列问题的全局最优解^[95]:

$$\min_Z E(Z) = 1.0 + \left(\sum_{i=1}^4 W_i^2 X_i^2 \right) \left[1.0 + \frac{y_5}{0.045} \times \sum_{i=1}^4 \left(W_i X_i - \sum_{j=1}^4 y_j \right)^2 \right] \quad (2.17)$$

其中, $Z = (X_1, \dots, X_4, y_1, \dots, y_5)^T$, $X_1, \dots, X_4 \in [-3.0, 3.0]$, $y_1, \dots, y_5$ 是二进制变量, $W_1 = 1.0$, $W_2 = 3.0$, $W_3 = 10.0$, $W_4 = 30.0$ 。

该问题有许许多多的局部极小点, 也有多个全局最优点。改进后的模拟退火法所获得的最优解为:

$$E(-0.002322, -0.000945, 0.000330, 0.000114, 1, 1, 0, 0, 1) = 1.000036$$

这是全局最优解之一。

2.6 小 结

本章针对模拟集成电路综合的实质是选择(生成)电路拓扑和优化器件尺寸, 将各种层次的电路级综合问题抽象为混整型非线性数学规划问题, 因此, 电路级综合问题的核心任务就是如何建立和求解该混整型非线性数学规划问题。

本章的重点是介绍混整型非线性数学规划问题的求解方法。我们首先扼要地介绍了用模拟退火法求解混整型非线性数学规划问题的基本思想, 紧接着又分析了造成模拟退火法计算量大的症结, 并详细地探讨了模拟退火法具有全局收敛性的根源, 在此基础上将模拟退火法和确定性优化算法结合起来以兼顾全局收敛性和计算量。

本章所介绍的多分子模拟退火法具有很好的全局收敛性, 且其计算量比模拟退火法小很多。本章所给出的优化算例说明多分子模拟退火法既保持了原模拟退火法的全局收敛性, 又大大提高了计算效率, 非常适合于求解混整型非线性数学规划问题。在后续章节中, 我们还将用多分子模拟退火法来处理各种实际问题, 到那时我们将再次领略到该方法在全局收敛性和计算速度方面所表现出的优越性能。

本章简要介绍了建立混整型非线性数学规划问题的方法之一——手工推导法, 利用符号分析法可以很容易自动地建立该混整型非线性数学规划问题, 这部分内容将在第三章讨论。

第三章 模拟电路的符号分析法

近年来,模拟电路的符号分析法(Symbolic Analysis)越来越受到人们的关注,这主要有两方面的原因。一方面,类似于 SPICE^[96]之类的通用电路模拟器虽然能给出非常精确的数值解,但却不能给出电路性能的函数表达式,不便于人们分析电路参数对电路性能的影响趋势;而符号分析法却可以完成这项工作,从而帮助电路设计者对电路的工作原理和电路性能有更系统、深入的认识。另一方面,电路级综合的核心环节是问题(2.3)的建立与求解,对于直流工作点和增益等性能的解析表达式而言,手工推导的复杂程度还可以忍受,但对于失调、噪声和谐波失真等性能,手工推导其解析表达式通常是苦不堪言的事;如果采用符号分析法,则可以自动推导出电路性能 $P(V, X, Y)$ 的解析表达式,使向模拟集成电路设计自动化系统中加入新的电路拓扑变得非常容易。本章将首先介绍符号分析法的功能及其应用领域,然后介绍符号分析法的基本理论,包括符号电路方程的建立、符号方程组的求解及符号表达式的逼近方法等。

3.1 符号分析法的功能及其应用领域

人们对具体电路的认识一般可分为两个方面:定性认识和定量认识。SPICE 之类的电路模拟器将时间轴(或频率轴)分解为一系列的离散点,然后用数值方法计算出每一个离散的时间点(或频率点)的电路性能,从而获得关于电路性能的定量描述。而符号分析法虽然仍属于定量分析的范畴,但它用一系列符号变量来表示电路元件参数、时间和频率,并用这些符号变量的函数(简称“符号函数”)来描述电路性能,符号函数不仅可以提供电路性能的定量信息,还可以帮助电路设计者对电路性能进行定性分析。

符号分析法能处理仅包含线性(或弱非线性)时不变集总元件的电路,只能提供复频域(s 域和 z 域)电路性能表达式,但时域符号分析技术目前尚不够成熟。

由于线性(或弱非线性)时不变系统的行为可以由其零状态响应来描述,而零状态响应又可以用其复频域的传输函数来表示,其一般形式为^[97]:

$$H(x) \equiv \frac{\text{Output}(x)}{\text{Input}(x)} = \frac{\sum_{i=0}^M x^i b_i(p_1, p_2, \dots, p_m)}{\sum_{i=0}^N x^i a_i(p_1, p_2, \dots, p_m)} \quad (3.1)$$

其中, x 为复频率变量,是我们常用的 s 或 z ; $\text{Output}(x)$ 和 $\text{Input}(x)$ 分别为输入输出信号, $p_1, p_2, \dots, p_m$ 为表示电路参数的符号变量;分子、分母中的系数 $b_i(\cdot)$ 和 $a_i(\cdot)$ 是符号变量的函数表达式,一般为多个符号变量之积的和,即:

$$a_i(p_1, p_2, \dots, p_m) = \sum_{k=1}^q c_{ik} \left(\prod_{j=1}^m p_{ij} \right)$$

$$b_i(p_1, p_2, \dots, p_m) = \sum_{k=1}^r d_{ik} \left(\prod_{j=1}^m p_{ik_j} \right)$$

符号分析的任务就是根据用户输入的具体电路、分析类型(s 域或 z 域)和输入输出信号, 求出(3.1)所示的 $H(x)$ 。

当然, 根据 $H(x)$ 中系数 $b_i(\cdot)$ 和 $a_i(\cdot)$ 的符号化程度, 我们又可以将 $H(x)$ 分为全符号化传输函数、半符号化传输函数和纯数值化传输函数等三类^[98]。这三种符号传输函数各有其优势, 一般的符号分析系统都能提供这些传输函数。下面是三个具体的实例:

(1) 全符号化传输函数

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{1 + RCs + LCs^2}$$

(2) 半符号化传输函数(将上述全符号化传输函数中的 R 和 C 数值化)

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{1 + 10^{-7}s + 10^{-11}Ls^2} \quad (R = 10\text{k}\Omega, C = 10\text{pF})$$

(3) 纯数值化传输函数(将上述全符号化传输函数中的 R, C 和 L 都数值化)

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{1 + 10^{-7}s + 5 \times 10^{-15}s^2} \quad (R = 10\text{k}\Omega, C = 10\text{pF}, L = 0.5\text{mH})$$

符号分析法具有许多数值仿真方法所不具备的优越性, 在模拟集成电路设计领域的应用非常广阔, 具体表现在如下几方面。

一、帮助设计者深入地认识电路的本质

目前的商用电路模拟软件(例如 SPICE, HSPICE^[99])虽然能够达到很高的精度, 计算效率也很不错, 但是, 当电路设计者输入电路结构、器件尺寸、工艺参数和工作环境后, 这类电路模拟器只能以表格或曲线的形式给出关于电路性能的具体数据, 设计者可以根据这些数据判断电路性能是否满足要求。例如: 对于图 3.1 所示带自举电容(Boot-

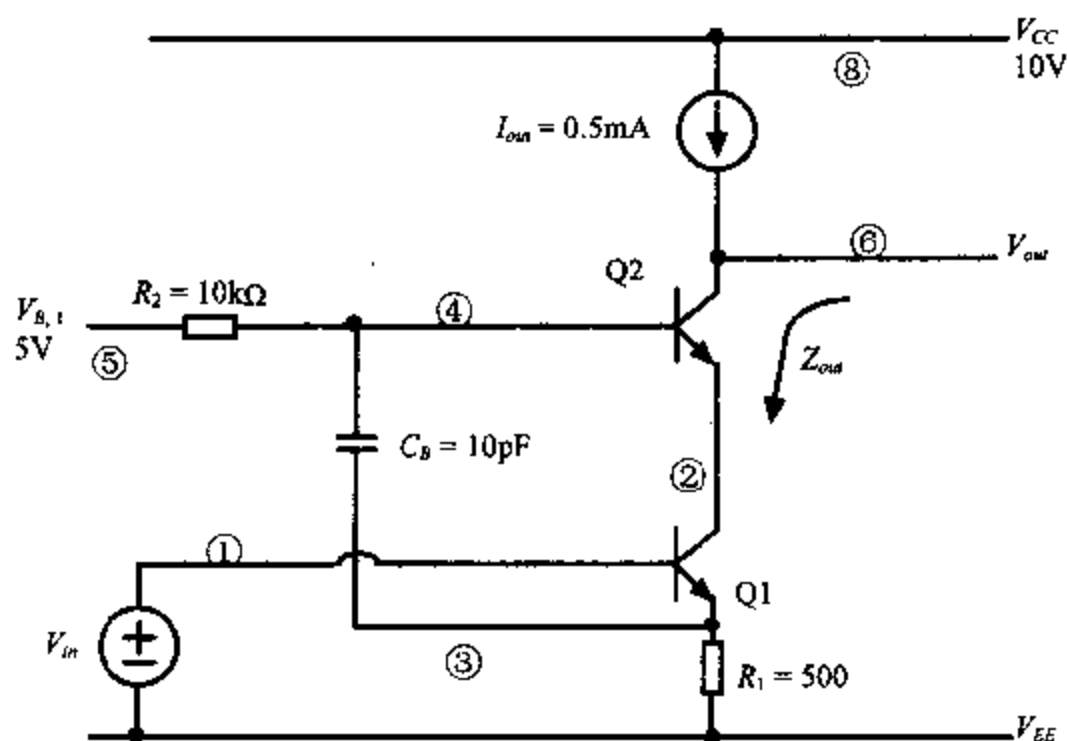


图 3.1 带自举电容的双极型放大器

strap Capacitor) 的双极型放大器, 电路设计者可以用 SPICE 获得输出电阻的频率特性(参见图 3.2), 但是, 该频率特性曲线是一些“孤立”的数据, 设计者无法从这些数据中获得“影响转折点 A 和 B 的主要电路参数”等更本质的信息。换言之, 通用电路模拟器

提供的数值结果所包含的信息毕竟太少、太表面化，主要存在下述局限性：

- (1) 不能找出电路的潜在问题；
- (2) 不能给出哪些元件是影响电路性能的关键因素；
- (3) 当电路性能不满足要求时，不能告诉设计者如何解决电路中存在的问题；
- (4) 当需要了解电路参数的变化对电路性能的影响，或综合考虑多项性能指标时，设计者只能进行反复的电路模拟，因而非常费时、费力。

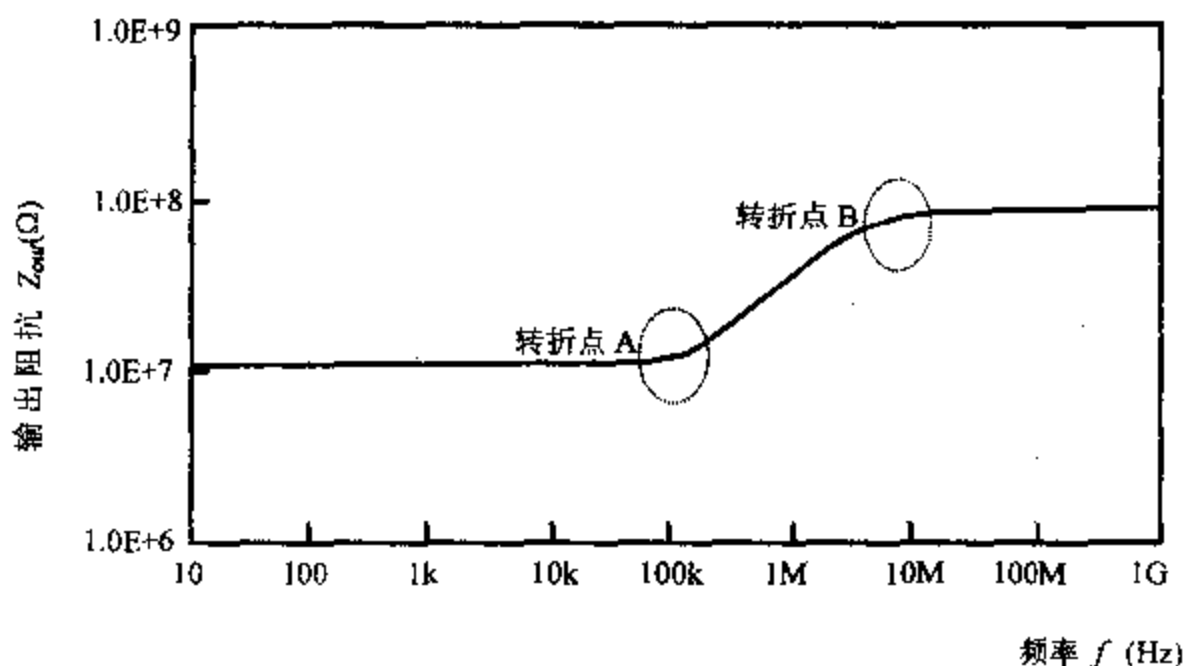


图 3.2 输出阻抗的频率特性(数值模拟结果)

由于通用电路模拟器采用的是数值算法，因此无法克服上述局限性。符号分析法以解析函数的形式给出电路性能，能揭示电路性能与电路参数之间的内在联系，是帮助设计者认识电路本质的有利工具。例如，如果我们采用符号仿真器 ISAAC^[40]来分析图 3.1 的输出阻抗随频率的变化特性，则只要先输入下述 SPICE 格式的电路描述文件：

```
* 带自举电容的双极型放大器
VCC 8 0 10
VB1 5 0 5
Vin 1 0 AC 1
Iout 8 6 0.5m
R1 3 0 500
R2 5 4 10K
CB 3 4 100P
Q1 2 1 3 NPN
Q2 6 4 2 NPN
.MODEL NPN NPN VAF=90
.END
```

并指定分析节点 6 的输出阻抗 $Z_{out}(s)$ ，在双极型晶体管的模型参数满足 $g_m \gg g_\pi \gg g_o$ 的条件下，就可以获得下述结果：

$$Z_{out}(s) = \frac{g_{m2}(g_{m1} + g_1)(g_2 + sC_B)}{g_{o2}g_{\pi2}(g_2(g_{m1} + g_1) + sC_B(g_1 + g_{\pi1}))}$$

如果再考虑到图 3.1 满足条件 $g_{m1} \gg g_1 \gg g_{\pi1}$ ，上式还可以进一步简化为：

$$Z_{out}(s) = \frac{g_{m2}g_{m1}(g_2 + sC_B)}{g_{o2}g_{\pi2}(g_2g_{m1} + sC_Bg_1)} \quad (3.2)$$

利用式(3.2), 我们很容易求出图 3.1 的低频输出阻抗和高频输出阻抗:

$$\lim_{f \rightarrow 0} Z_{out}(s) = \frac{g_{m2}}{g_{\pi2}g_{o2}} \quad (3.3)$$

$$\lim_{f \rightarrow \infty} Z_{out}(s) = \frac{g_{m1}g_{m2}}{g_1g_{\pi2}g_{o2}} = \frac{g_{m1}}{g_1} \lim_{f \rightarrow 0} Z_{out}(s) \quad (3.4)$$

当然, 从式(3.2)的零极点还可以直接导出图 3.2 中转折点 A 和 B 的位置:

$$f_A = \frac{1}{2\pi R_2 C_B} \quad (3.5)$$

$$f_B = \frac{g_{m1}R_1}{2\pi R_2 C_B} \quad (3.6)$$

因此, 电路设计者只要分析式(3.2)~(3.6)就可以认清电路的本质, 从而发现电路的潜在问题, 探索提高电路性能的有效途径。由此可见, 符号分析法能帮助电路设计者掌握电路工作原理、理清电路参数与电路性能之间的复杂关系。

2. 改善电路设计的交互环境

在符号仿真器所获得的解析函数的基础上, 输出可视化电路性能是迎刃而解的事, 如果再增加友好的图形用户界面, 则可以为电路设计者提供一个交互式设计环境^[40,100], 使改进电路结构或改善电路性能之类的工作变得快捷、容易。这里仍然以前面讨论过的图 3.1 为例, 图 3.3 是式(3.2)~(3.6)的图示化输出阻抗, 当电路设计者改变电路结构或器件参数时, 只要再进行一次符号分析, 并将新的图示化输出阻抗叠加到图 3.3, 设计者就可以获得准确而又直观的对比结果, 完成新电路设计。

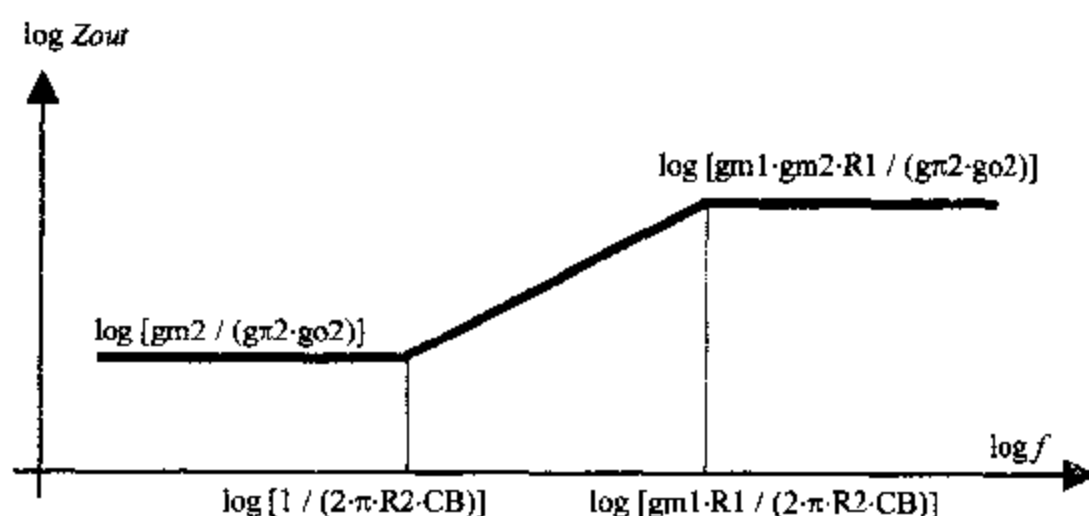


图 3.3 符号分析法所获得的图示化输出阻抗

3. 为模拟集成电路设计自动化系统提供解析电路模型

由于符号分析法能自动推导出电路性能的符号表达式, 因此, 只要将符号模拟器集成到模拟集成电路设计自动化系统中, 就可以免去手工推导电路性能解析表达式的劳苦; 而且当电路设计把新的电路拓扑加入到系统中时, 只需描述电路结构, 不必给出电路性能的表达式, 更不需要为新的电路拓扑编写程序代码或采用编译、链接等手段重新生成应用程序。比利时鲁汶大学所开发的 ASAIC 系统^[101]就成功地实现了该功能。

4. 集成电路的优化设计、统计模拟和形式验证

在统计模拟过程中,需要反复计算同一电路的若干项性能,只不过每次计算的电路参数或工作频率不同而已,合格率估计^[32,102]和优化设计就属于这种情况,用符号模拟法处理这类问题的效率极高^[40]。由于符号表达式能够清晰地反映器件尺寸之间的失配,因而能够很准确地计算噪声、失调、电源信号抑制比等电路性能。

采用数值算法的电路仿真器只能通过反复模拟来验证在特定条件下(通常不可能覆盖所有情况)电路性能是否符合要求。利用符号分析法所得到的解析表达式,我们可以证明电路在哪些条件下能正常工作、在哪些环境下不能很好地工作,这就是所谓的形式验证。当然,符号模拟法也可以用于数字集成电路的形式验证和时序诊断^[103-105],而且其计算效率比传统的逻辑仿真器高。

5. 复杂电路、混合信号系统的模拟

尽管稀疏矩阵技术、波形松弛技术使传统电路仿真器的收敛性和计算速度都有较大的提高^[96,106,107],但基于数值算法的电路仿真器所能处理的电路规模依然很小,目前,还没有一个数值仿真器能处理包含 10 000 个器件以上的电路。利用符号模拟技术则可以比较好地解决该问题。我们可以采用符号分析法导出模拟集成电路单元的性能函数,并用它来预估电路性能,从而使模拟系统的行为级模拟结果能够和电路级仿真结果很好地吻合。由于采用解析公式,不仅消除了收敛问题,而且还有助于提高计算效率。在复杂电路的时域分析中,将符号模拟法和数值算法结合起来是很好的策略^[108],例如:可以将符号模拟法获得的电路性能作为松弛法的初值,从而减小计算量。另外,对于设计者已经完成的电路设计,电路参数都是已知的,可以用符号模拟法导出纯系数型符号表达式(传输函数中仅含变量 s 或变量 z),因此,特别适合于分析控制系统的稳定性。

随着深亚微米技术的出现,功耗已成为电路设计者和系统设计者所关心的主要问题。特别是在数模混合系统中,芯片功耗密度的增加将使集成电路设计者必须同时考虑电效应和热效应;而系统集成技术的迅速发展将使集成电路设计者不得不面对其他电与化学物质、电与光、电与机械等混合信号系统,而符号分析法在这类混合信号的仿真领域也有非常明显的优势^[109]。

3.2 符号模拟器的典型流程

集成电路的符号分析法已有近 30 年的历史,已经出现了许多关于线性、时不变、集总元件电路的符号分析方法,我们可以将它们分为五类:符号线性方程组法^[40,110,111]、信号流图法^[112,113]、树枚举法^[114]、参数提取法^[115]、拉格朗日多项式插值法^[116,117]。

树枚举法只能处理包含电阻、电感、电容和压控电流源的时不变系统,而且目前还没有直接生成最优树的方法,因此其应用领域有限,效率也比较低;参数提取法用系统的不定导纳矩阵 Y_{ind} 来描述电路性能,它要求每个符号变量只出现在 Y_{ind} 的同一行或同一列中,或 Y_{ind} 只包含极少个符号变量的情形,其应用领域也比较窄;拉格朗日多项式插值法的不足是对数值误差特别敏感,特别是数值计算误差的传递将产生许多“冗余”的

符号表达式,不利于引导电路设计者抓住本质问题。因此,本书将不讨论这些方法,有兴趣的读者请阅读相关文献[114—117]。符号线性方程组法和信号流图法都是通用方法,适用面广,而且本身不会约减符号信息(用户将多个器件命名为同一符号变量的情况例外),有利于电路设计者获得全面的信息。

信号流图法将电路转换为一个带权重的有向图,该图的节点是表示原电路节点电压或支路电流的符号变量,而其中的权重则是原电路中的元件,我们称之为符号信号流图。因此,从节点 N_i 到 N_j 的传输函数就可以根据符号信号流图的结构,用下述梅森(Mason)增益公式获得^[118]:

$$\frac{N_j(x)}{N_i(x)} = \frac{\sum_k p_k \Delta_k}{\Delta} \quad (3.7)$$

其中,

$\Delta = 1 - (\text{一阶环路增益和}) + (\text{二阶环路增益和}) - (\text{三阶环路增益和}) + \dots$;

p_k = 从节点 N_i 到 N_j 的第 k 条前向路径的增益;

Δ_k = Δ 中与 p_k 不相交叠的环路增益和;

n 阶回路指 n 个互不接触的环路;

环路增益等于环路中各支路权重的积;

路径增益等于路径中各支路权重的积。

一旦建立了符号信号流图,求传输函数的工作就是枚举符号信号流图各阶回路和前向路径,具体算法参见文献[98]。

由于符号信号流图通常是根据电路方程组来建立的,这是信号流图法的首要环节,并且与符号线性方程组法是雷同的,因此,本章后续部分将详细介绍符号线性方程组的建立、求解及符号函数逼近法。不过,在讨论这些核心算法之前,我们先给出符号模拟器的流程图,以便读者能对符号模拟器的主要工作环节及其相互依赖关系有一个全面、清晰的认识。

图 3.4 是一个典型的符号模拟器的流程图^[40]。其中,输入处理模块的功能与 SPICE^[96]或 HSPICE^[99]等通用电路仿真器的输入处理基本一致,都是先读入电路网表,展开子电路,然后确定分析类型,并用等效电路模型来转换电路中的 MOS 场效应晶体管、双极型晶体管等器件;而符号模拟器的拓扑检查主要包括:零口器(Nullator)数目与非口器(Norator)数目是否一致,是否有非法的零输入阻抗回路,是否有非法的无穷大阻抗所形成的割集。在分析类型方面,目前的符号模拟器只能进行 s 域分析和 z 域分析。建立符号电路方程组的本质仍然是改进节点法,可以很容易地将器件的失配信息填入符号电路方程组中;符号方程组的求解主要是计算符号矩阵的行列式值;而符号函数逼近法主要是为了简化符号表达式,去除那些对电路性能影响很小的项,使用户更容易把握电路性能的本质;输出模块的主要功能是输出形如式(3.1)的传输函数。由于计算上直接输出上下脚标和分数线之类的表达式比较繁琐,而电路综合软件识别这类符合人们视觉的表达式(简称“习惯表达式”)又非常困难,因此,符号模拟器输出的表达式中,所有变量都不包含下脚标,而采用与 C 语言比较类似的表示形式。图 3.3 已经使用了这种表示。下面列出了部分习惯表达式与符号表达式的对应关系:

含下脚标的变量	$g_{\pi 1}$	$\Leftrightarrow$ 符号表达式 $g_{\pi 1}$
器件 Q_1 的模型参数	R_o	$\Leftrightarrow$ 符号表达式 $Q1 \# Ro$
含乘法和加法的表达式	$g_{\pi 1} R_o + g_{m5} R_C$	$\Leftrightarrow$ 符号表达式 $g_{\pi 1} \cdot Ro + g_{m5} \cdot RC$
含分数线的表达式	$\frac{g_{m1} + g_{m2}}{g_{\pi 1} R_o}$	$\Leftrightarrow$ 符号表达式 $(g_{m1} + g_{m2}) / (g_{\pi 1} \cdot Ro)$
含幂指数的表达式	$\sqrt[4]{I_b g_m^n}$	$\Leftrightarrow$ 符号表达式 $Ib^{(1/4)} \cdot gm^n$

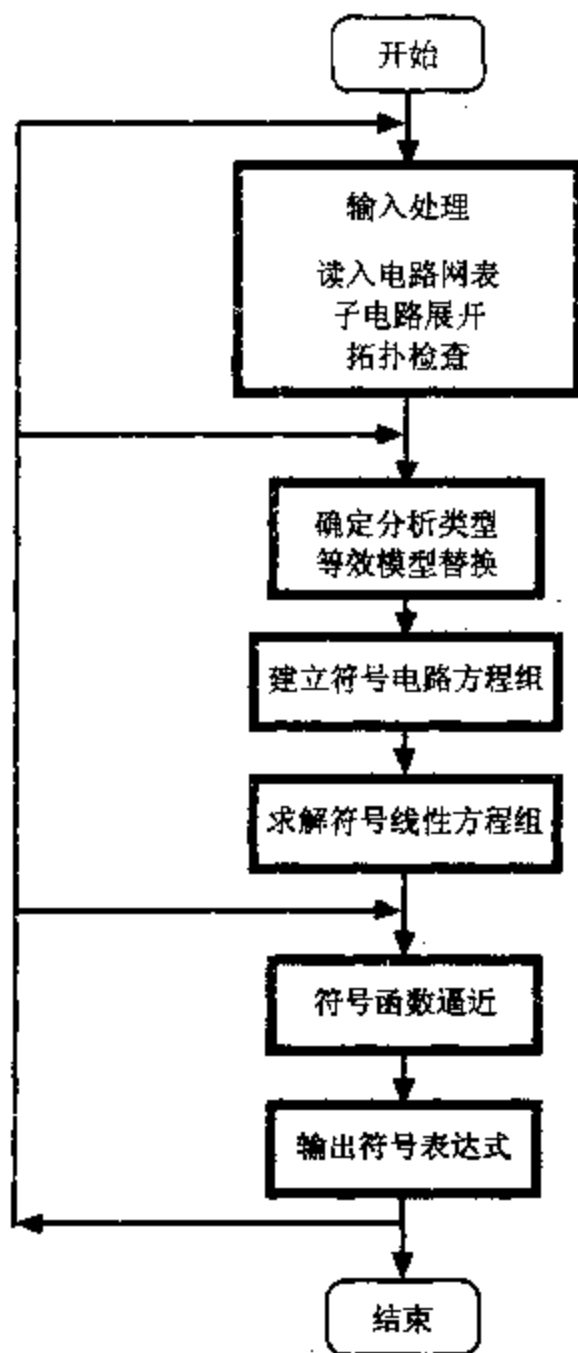


图 3.4 符号模拟器的典型流程图

3.3 建立符号电路方程组

符号模拟器能处理的电路基本元件包括：电阻、电容、电感、独立电压源、独立电流源、压控电压源 (VCVS)、压控电流源 (VCCS)、流控电压源 (CCVS)、流控电流源 (CCCS)、零口器和非口器等。对于包含二极管、双极型晶体三极管、MOS 场效应管、结型场效应管等器件的集成电路，可以利用器件的等效模型将它变换为只含基本元件的电路。图 3.5 是双极型单管放大器及其小信号等效电路。建立符号电路方程的本质和电路模拟器建立代数方程组是一致的，主要是列电路的 KCL、KVL 方程及元件的电压电流特性方程 (简称 VCR 方程)。

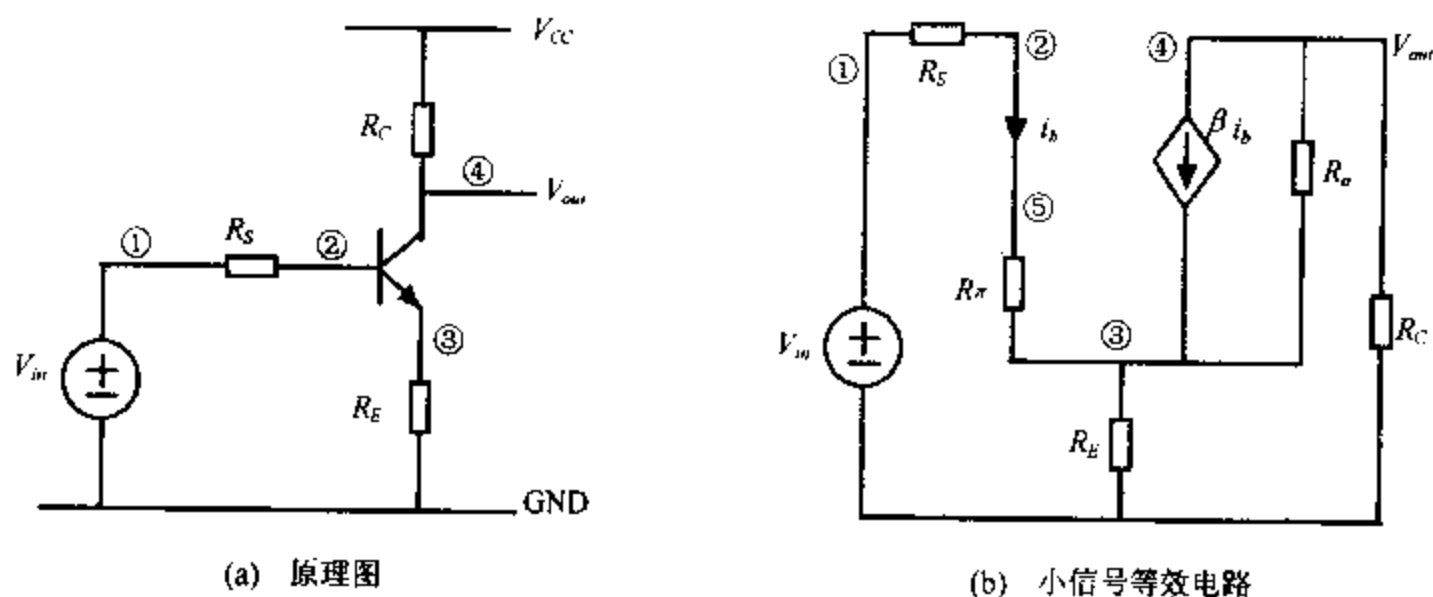


图 3.5 双极型单管放大器及其小信号等效电路

3.3.1 表矩阵法

表矩阵法将支路电压、支路电流和节点电压作为变量，在建立电路方程时可以选电路中的任何节点作为电压参考点(在实际应用中通常选电路的直流零电位节点)。表矩阵法是一组独立的 KCL 方程、KVL 方程和 VCR 方程。如果某电路包含 n 个节点和 b 条支路，则其表矩阵的一般形式如下：

$$\begin{array}{l} \text{KCL} \\ \text{KVL} \\ \text{VCR} \end{array} \begin{bmatrix} 0 & 0 & A \\ -A^T & E & 0 \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} V \\ I \\ U \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \quad (3.8)$$

其中， V 是 $(n-1)$ 维列向量，表示节点电压； I 和 U 都是 b 维列向量，分别表示支路电流和支路电压； A 是 $(n-1) \times b$ 阶矩阵，它是电路的关联矩阵； E 是 b 阶单位矩阵； Y 是 b 阶导纳矩阵； W 是 b 维列向量，它包含独立电压源、独立电流源以及电容和电感的初始条件对电路的影响。至于生成关联矩阵 A 的具体过程、基本电路元件与矩阵 Y 和 Z 的对应关系请参考文献[107]。

对于包含 n 个节点和 b 条支路的电路而言，用表矩阵法所建立的符号电路方程的维数为 $(n-1+2b)$ 。因此，尽管用表矩阵法建立符号方程组的过程非常简单，但其所建立方程存在系数矩阵维数较高的缺陷。

3.3.2 改进节点法

为克服表矩阵法的缺陷，人们提出了节点法。节点法仅把节点电压作为未知变量，因此，对于包含 n 个节点和 b 条支路的电路而言，节点法所形成的方程组是 $(n-1)$ 维的。节点法的关键是建立电路的导纳矩阵和等效电流源向量。节点法的不足之处是不能直接处理独立电压源、零阻抗支路、CCVS 和 CCCS^[107,119]。改进节点法是在节点法的基础上建立起来的，它既保持了节点法方程阶数低、方法简单的优点，又克服了节点法的不足，在采用数值算法的电路模拟器中得到了广泛的应用^[98,107,119]。

改进节点法的未知变量除节点法所使用的节点电压之外，还包括部分支路电流。在改进节点法中，对除参考节点之外的所有节点都列出其 KCL 方程，每当遇到电感支路

或非压控型支路(通常称这类支路为困难支路),就将该支路电流作为未知变量,并增加一个描述该困难支路特性的方程(简称 BR 方程)。因此,改进节点法在处理电感、独立电压源、VCVS、CCCS、零口器和非口器时,都需要引入一个支路电流变量,并增加一个 BR 方程;在处理 C CVS 时,则需要为控制支路和受控支路各引入一个支路电流变量,即增加两个 BR 方程。对于包含 n 个节点和 m 条困难支路的电路而言,用改进节点法所建立的 s 域电路方程组的一般形式为:

$$\begin{array}{l} \text{KCL} \\ \text{BR} \end{array} \quad \begin{bmatrix} Y & A \\ B & Z \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} J \\ P \end{bmatrix} \quad (3.9)$$

其中, V 和 J 都是 $(n-1)$ 维列向量,分别表示电路的节点电压和独立电流源; I 和 P 都是 m 维列向量,分别表示困难支路电流和独立电压源; Y 是 $(n-1)$ 阶导纳矩阵;而 A , B 和 Z 分别是 $(n-1) \times m$, $m \times (n-1)$ 和 $m \times m$ 阶矩阵,它们描述困难支路电流间的相互关系。

每一类基本元件都与式(3.9)中的矩阵元素有一一对应关系,这种对应关系通常称为元件送值表。因此,只需根据电路(或等效电路)的拓扑结构和元件送值表就可以非常方便地形成式(3.9)所示的电路方程组。表 3.1 给出了 8 类基本元件的送值表,其中 RHS 表示式(3.9)的右端向量。其余常用元件的送值表请参见文献[107,119]。

例 3.1 用改进节点法建立图 3.5(a)所示双极型单管放大器的 s 域符号电路方程组。

我们先根据双极型晶体管的小信号等效电路,得到图 3.5(b)所示的等效电路,根据改进节点法选择未知变量的原则,节点电压 V_1, V_2, V_3, V_4 , 电源 V_{in} 的电流 I_E 和 CCCS 的控制支路电流 i_b 是未知变量,结合常用元件的送值表,很容易就得到下述 s 域的符号方程组:

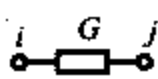
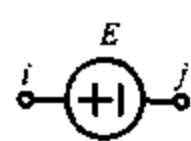
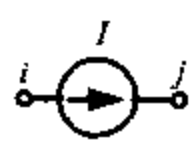
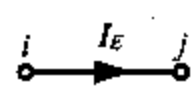
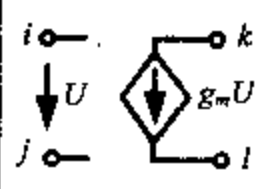
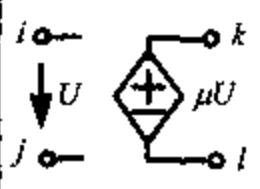
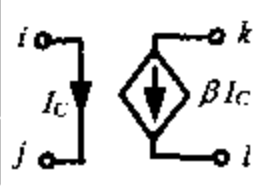
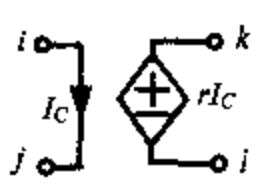
$$\begin{bmatrix} R_S^{-1} & -R_S^{-1} & 0 & 0 & 0 & 0 & -1 \\ -R_S^{-1} & R_S^{-1} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & R_\pi^{-1} + R_E^{-1} + R_O^{-1} & -R_O^{-1} & -R_\pi^{-1} & -\beta & 0 \\ 0 & 0 & -R_O^{-1} & R_O^{-1} + R_C^{-1} & 0 & \beta & 0 \\ 0 & 0 & -R_\pi^{-1} & 0 & R_\pi^{-1} & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ i_b \\ I_E \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_{in} \\ 0 \end{bmatrix} \quad (3.10)$$

其中, $V_1 = V_{in}$, $V_4 = V_{out}$ 。

3.3.3 压缩改进节点法

细心的读者或许已经注意到:用改进节点法所获得的符号方程依然包含许多冗余的未知变量。例如:对于求从输入 V_{in} 到输出 V_{out} 的传输函数而言,式(3.10)中的变量 V_1 , V_5 , I_E 和 i_b 就是冗余的。目前,压缩改进节点法(Compacted Modified Nodal Analysis, 简称 CMNA)是建立符号电路方程最有效的方法^[120]。

表 3.1 常用元件送值表

元件	符号	送值表	特性方程																
可用导纳 直接描述 的元件 R, L, C	 $R \Rightarrow G = 1/R$ $C \Rightarrow G = sC$ $L \Rightarrow G = 1/sL$	<table><tr><th>列 \ 行</th><th>i</th><th>j</th></tr><tr><th>i</th><td>G</td><td>$-G$</td></tr><tr><th>j</th><td>$-G$</td><td>G</td></tr></table>	列 \ 行	i	j	i	G	$-G$	j	$-G$	G	欧姆定律							
列 \ 行	i	j																	
i	G	$-G$																	
j	$-G$	G																	
电压源		<table><tr><th>列 \ 行</th><th>i</th><th>j</th><th>RHS</th></tr><tr><th>BR</th><td>1</td><td>-1</td><td>E</td></tr></table>	列 \ 行	i	j	RHS	BR	1	-1	E	$V_i - V_j = E$								
列 \ 行	i	j	RHS																
BR	1	-1	E																
电流源		<table><tr><th>列 \ 行</th><th>i</th><th>j</th><th>RHS</th></tr><tr><th>i</th><td></td><td></td><td>1</td></tr><tr><th>j</th><td></td><td></td><td>-1</td></tr></table>	列 \ 行	i	j	RHS	i			1	j			-1	$I_i = -I_j = I$				
列 \ 行	i	j	RHS																
i			1																
j			-1																
短路支路		<table><tr><th>列 \ 行</th><th>i</th><th>j</th><th>I_E</th></tr><tr><th>i</th><td></td><td></td><td>1</td></tr><tr><th>j</th><td></td><td></td><td>-1</td></tr><tr><th>BR</th><td>1</td><td>-1</td><td></td></tr></table>	列 \ 行	i	j	I_E	i			1	j			-1	BR	1	-1		I_E 为任意值 $I_i = -I_j = I_E$ $V_i = V_j$
列 \ 行	i	j	I_E																
i			1																
j			-1																
BR	1	-1																	
VCCS		<table><tr><th>列 \ 行</th><th>i</th><th>j</th></tr><tr><th>k</th><td>g_m</td><td>$-g_m$</td></tr><tr><th>l</th><td>$-g_m$</td><td>g_m</td></tr></table>	列 \ 行	i	j	k	g_m	$-g_m$	l	$-g_m$	g_m	$I_k = -I_l = g_m (V_i - V_j)$							
列 \ 行	i	j																	
k	g_m	$-g_m$																	
l	$-g_m$	g_m																	
VCVS		<table><tr><th>列 \ 行</th><th>i</th><th>j</th><th>k</th><th>l</th><th>RHS</th></tr><tr><th>BR</th><td>$-\mu$</td><td>μ</td><td>1</td><td>-1</td><td>E</td></tr></table>	列 \ 行	i	j	k	l	RHS	BR	$-\mu$	μ	1	-1	E	$(V_k - V_l) = \mu (V_i - V_j)$				
列 \ 行	i	j	k	l	RHS														
BR	$-\mu$	μ	1	-1	E														
CCCS		<table><tr><th>列 \ 行</th><th>i</th><th>j</th></tr><tr><th>k</th><td>β</td><td>$-\beta$</td></tr><tr><th>l</th><td>$-\beta$</td><td>β</td></tr></table>	列 \ 行	i	j	k	β	$-\beta$	l	$-\beta$	β	$I_i = -I_j = I_C$ $I_k = -I_l = \beta I_C$							
列 \ 行	i	j																	
k	β	$-\beta$																	
l	$-\beta$	β																	
CCVS		<table><tr><th>列 \ 行</th><th>i</th><th>j</th><th>I_C</th></tr><tr><th>i</th><td></td><td></td><td>1</td></tr><tr><th>j</th><td></td><td></td><td>-1</td></tr><tr><th>BR</th><td>1</td><td>-1</td><td>$-r$</td></tr></table>	列 \ 行	i	j	I_C	i			1	j			-1	BR	1	-1	$-r$	$V_k - V_l = rI$
列 \ 行	i	j	I_C																
i			1																
j			-1																
BR	1	-1	$-r$																

CMNA 法是在改进节点法基础上发展起来的, 用 CMNA 法建立 s 域符号电路方程组时, 只对参考节点之外的节点列出 KCL 方程, 并对困难支路增加一个 BR 方程, 当然, 未知变量也是节点电压和困难支路的电流, 因此其符号方程的形式与式(3.9)完全一致。CMNA 法的核心思想是只保留那些对计算传输函数必不可少的变量, 并去除一些冗余方程。前者称为列压缩, 后者称为行压缩。

用 CMNA 法建立 z 域电路方程组时, 其行压缩和列压缩的方法与建立 s 域的符号电路方程组是一样的, 但其出发点已不再是 KCL 方程, 而是电荷守恒定律和支路特性方程, 未知变量是节点电压和电容支路的电荷^[120,121]。对于有 N 相时钟的电路而言, 用 CMNA 法所建立的 z 域符号电路方程的一般形式为^[121,122]:

$$\left[\begin{array}{cccccc|cccc} A_1 & 0 & \cdots & 0 & 0 & -z^{-1}E_1 & B_1 & 0 & \cdots & 0 \\ -E_2 & A_2 & \ddots & 0 & 0 & 0 & 0 & B_2 & & \\ 0 & -E_3 & \ddots & 0 & 0 & 0 & & & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & & \ddots & \\ 0 & 0 & \ddots & E_{N-1} & A_{N-1} & 0 & & & & 0 \\ 0 & 0 & \cdots & 0 & -E_N & A_N & 0 & \cdots & 0 & B_N \\ \hline C_1 & 0 & & \cdots & & 0 & D_1 & 0 & \cdots & 0 \\ 0 & C_2 & & & & & 0 & D_2 & & \\ & & & \ddots & & \vdots & & & \ddots & \\ \vdots & & & \ddots & \ddots & & \vdots & & \ddots & \\ & & & & 0 & & & & & 0 \\ 0 & \cdots & & & 0 & C_N & 0 & \cdots & 0 & D_N \end{array} \right] \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_{N-2} \\ V_{N-1} \\ V_N \\ Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_{N-2} \\ J_{N-1} \\ J_N \\ F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} \quad (3.11)$$

其中, $V_k, Q_k (k=1, 2, \dots, N)$ 分别是电路在第 k 相时钟时的节点电压向量和附加支路的电荷向量; F_k, J_k 分别是电路在第 k 相时钟时的电压源向量和电荷源向量; 分块矩阵 A_k, B_k, C_k, D_k 和 E_k 构成 z 域符号电路方程的系数矩阵, 由用 CMNA 法建立的 s 域符号电路方程组所获得。

在建立 z 域符号电路方程组(3.11)时, 首先为电路建立 N 个理想相(时间常数为 0), 它们分别对应于电路的 N 相时钟, 用 CMNA 法建立这 N 个理想相的 s 域符号电路方程组, 将这些方程组中的 s 置为 1 就可以获得(3.11)中的 A_k, B_k, C_k, D_k, Q_k 和 F_k ; 求 E_k 的过程稍微繁琐一些, 不过仍然是从电路的 N 个理想相出发, 并用改进节点法建立相应的 s 域符号电路方程组, 我们将该方程组的系数矩阵记为 E_k^* , 然后根据第 k 个理想相对 E_k^* 作行压缩, 根据第 $(k-1)$ 个理想相作列压缩, 便可以得到 E_k 。建立 z 域符号电路方程的详细细节请参见文献[122]。

1. 行压缩

CMNA 法首先对改进节点法所建立的符号方程组进行行压缩。在 CMNA 法中, 只有必不可少的支路电流才作为符号方程组的未知变量或右端向量, 即 CMNA 法只引入下述支路电流:

- (1) 部分独立电源的电流 (未知变量)
- (2) 被检测的支路电流 (未知变量)
- (3) CCVS 和 VCVS 的被控制支路的电流 (未知变量)
- (4) CCCS 和 VCCS 的被控制支路的电流 (右端向量)

与改进节点法相比, CMNA 法没有引入短路支路、非口器、CCVS 和 VCVS 的控制支路电流; 对于独立电压源而言, 只有当其电流是输出信号或输出信号的一部分时, 才出现在符号方程组中。

在改进节点法中, 如果某电流控制支路的节点分别为 $n+$ 和 $n-$ (参见图 3.6), 则该支路电流将同时出现在两个方程中——节点 $n+$ 和 $n-$ 的 KCL 方程, 而且在符号矩阵中的值分别为 $+1$ 和 -1 。CMNA 法进行行压缩时, 对这类支路仅列一个 KCL 方程, 因为将改进节点法关于节点 $n+$ 和 $n-$ 的 KCL 方程加起来就可以消去该支路电流。

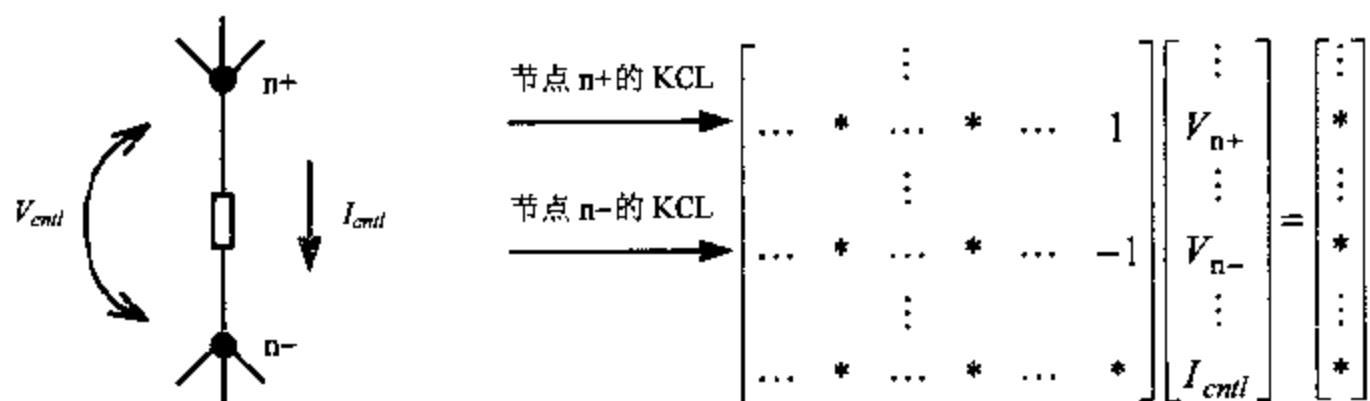


图 3.6 电流控制支路及与之对应的改进节点法的符号方程组

2. 列压缩

列压缩在行压缩之后进行, 其作用是消去那些相关的节点电压。由于电流计、短路支路及零口器的两端电压 V_+ 和 V_- 是相等的, 即:

$$V_+ - V_- = 0 \quad (3.12)$$

类似地, 电路中的两个节点之间的电压差为常数 E , 则改进节点法中存在下述方程:

$$V_+ - V_- = E \quad (3.13)$$

该方程是式(3.12)的一般形式, 我们称 V_+ 和 V_- 是相关的。形如(3.12)或(3.13)的方程都将在 CMNA 法的行压缩中被删去, 而且在未知变量中只保留 V_+ 或 V_- 。删除规则为: 如果 V_+ 或 V_- 都既不是输入信号又不是输出信号, 且它们都不是参考节点, 则可以保留其中任何一个; 如果 V_+ 和 V_- 中只有一个是输入或输出信号, 则保留该变量; 如果 V_+ 或 V_- 中有一个是参考节点, 则消去另一个节点。

列压缩的过程非常简单。我们假设 V_+ 和 V_- 分别位于式(3.9)中未知向量的第 m 行和第 k 行, 只要先将式(3.9)的系数矩阵的第 k 列元素加到第 m 列, 然后将系数矩阵的第 k 列元素乘以 E 后再加到右端向量, 最后删去对应于(3.12)或(3.13)的方程: 系数矩阵的第 k 列和未知变量的第 k 个元素, 就可以从式(3.9)中删除未知变量 V_- , 实现列压缩。

特别值得一提的是: 当电路中存在多个互联的电流计、短路支路、零口器和直流电压源时 (参见图 3.7(a)), 则需要特别注意。例如: 如果在处理这类器件时, 先决定用与

节点 a 相关的变量 V_a 消去与节点 b 相关的变量 V_b ，而后来处理别的器件时又需要用与节点 d 相关的变量 V_d 消去 V_a ，这时就需要重新处理与节点 b 相关的所有变量，因此，非常繁琐且效率极低。采用适当的顺序进行列压缩便可以解决反复处理同一个节点的问题。具体算法如下：

首先为电路中的所有电流计、短路支路、独立电压源和零口器建立一个无向图，记为 $G(E, V)$ ，假设该无向图包含 P 个非连通子图 $G_1(E_1, V_1), G_2(E_2, V_2), \dots, G_P(E_P, V_P)$ ，则列压缩过程可以分别处理 $G_j(E_j, V_j), j=1, 2, \dots, P$ 中的每一个器件。在处理子图 $G_j(E_j, V_j)$ 时，如果其顶点集合 V_j 中包含电路的参考节点（通常是交流地），则将该节点选为初始顶点 V_{0j} ，否则，从 E_j 中任选一个顶点作为初始顶点 V_{0j} ，然后从该顶点出发遍历图 $G_j(E_j, V_j)$ ，并将所有顶点对应的节点电压用顶点 V_{0j} 对应的节点电压来表示。因此， $G(E, V)$ 中所有顶点对应的节点电压就被压缩为 P 个独立的节点电压 $V_{01}, V_{02}, \dots, V_{0j}$ 。

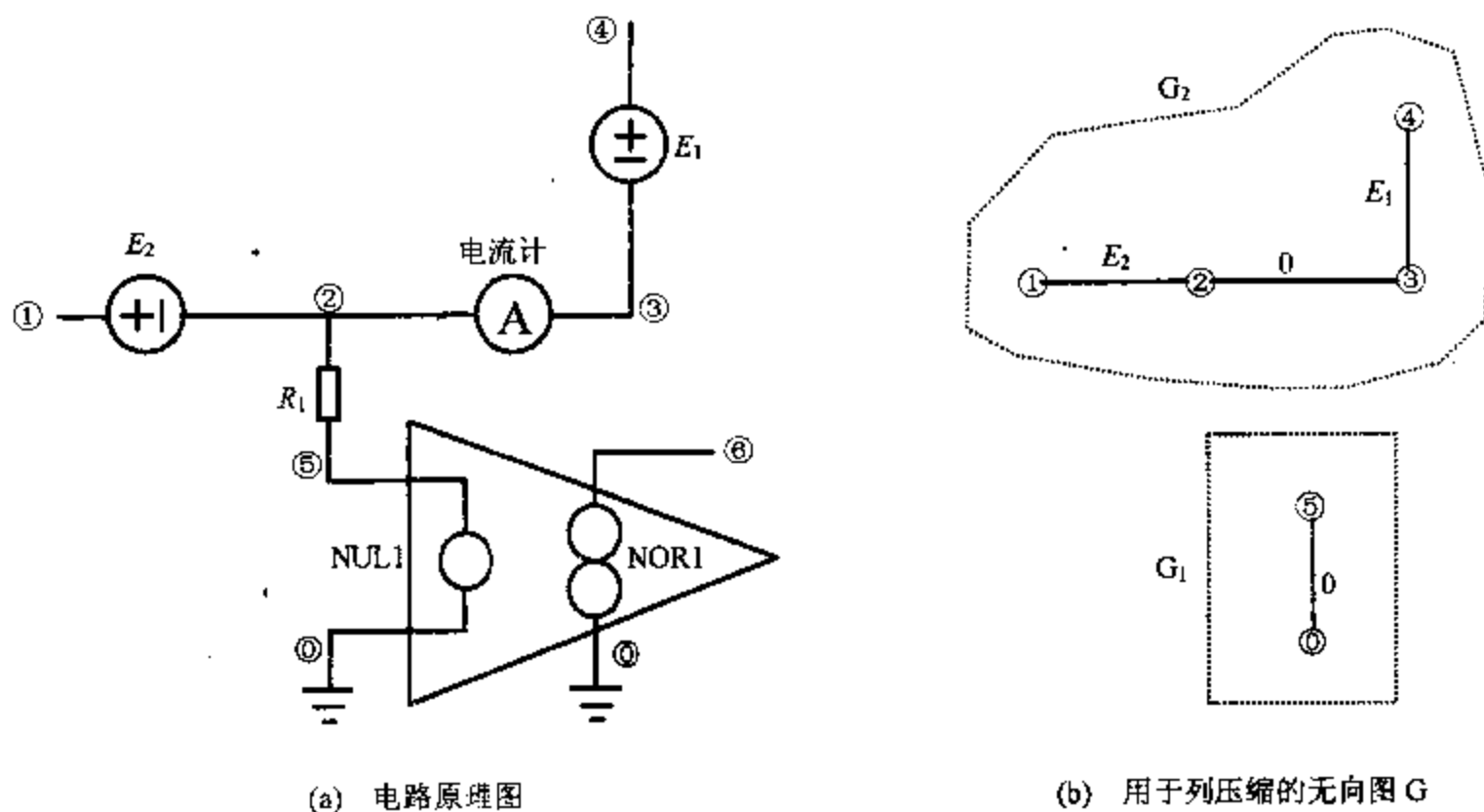


图 3.7 一个范例电路及列压缩时所采用的无向图

我们以图 3.7(a) 所示的电路为例介绍 CMNA 法的列压缩过程。图 3.7(a) 包含两个电压源、一个零口器和一个电流计，它们所构成的无向图 G 如图 3.7(b) 所示，该无向图包含两个不连通的子图 G_1 和 G_2 。在 G_1 中，由于节点 0 是参考节点，因而将它选作列压缩的初始节点，因此，节点电压 V_5 被 V_0 替换。在处理图 G_2 时，由于不包含参考节点，如果将节点 1 选作列压缩的初始节点，则对图 G_2 的遍历结果将依次为节点 2, 3, 4，从而得到下述变量替换关系：

$$\begin{aligned} V_2 &\diamond V_1 - E_2 \\ V_3 &\diamond V_2 + 0 \diamond V_1 - E_2 \\ V_4 &\diamond V_3 + E_1 \diamond V_1 - E_2 + E_1 \end{aligned}$$

由此可见，通过上述处理， G 中每一个连通子图都只有一个独立的节点电压，但填系数矩阵的过程将发生变化。例如：采用 CMNA 法建立图 3.7(a) 的符号电路方程时，节点电

压 V_4 被表示为 $(V_1 - E_2 + E_1)$, 其含义是: 凡是联接到节点 4 的元件, 当按送值表 3.1 建立 s 域的符号电路方程时, 应当将原 MNA 法中填入的与 V_4 对应的列元素填入与 V_1 和 E_1 对应的列中, 同时还要将这些元素乘以 -1 后再填入与 E_2 对应的列中。

例 3.2 用 CMNA 法建立图 3.8 所示的两级有源滤波器的 s 域符号电路方程组。

我们假设图 3.8 中的运算放大器都是理想的, 即: 输入端可视为零口器, 输出端可视为非口器, 增益无穷大。采用改进节点法时, 节点电压 $V_1, V_2, \dots, V_7$ 以及输入信号 V_{in} 和 3 个运放的输入端口电流都是未知变量, 因此, 改进节点法需要用一个 11×11 维的系数矩阵来描述该电路的特性。然而, 如果采用 CMNA 法来建立符号电路方程组, 由于支路电流对计算传输函数没有影响, 可以全部略去; 由于运放输入端等效为零口器, V_2, V_4 和 V_6 都将被参考节点(信号地)替代, 因此, 只需将 V_3, V_5 和 V_7 作为未知变量, 对节点 2, 4 和 6 列写 KCL 方程即可。下面是用 CMNA 法所获得的 s 域符号电路方程组:

$$\begin{bmatrix} R_2^{-1} + sC_1 & 0 & R_5^{-1} \\ R_4^{-1} & sC_2 & 0 \\ 0 & R_6^{-1} & R_7^{-1} \end{bmatrix} \begin{bmatrix} V_3 \\ V_5 \\ V_7 \end{bmatrix} = - \begin{bmatrix} R_1^{-1} \\ R_3^{-1} \\ 0 \end{bmatrix} V_{in}, \quad \text{其中, } V_7 = V_{out}$$

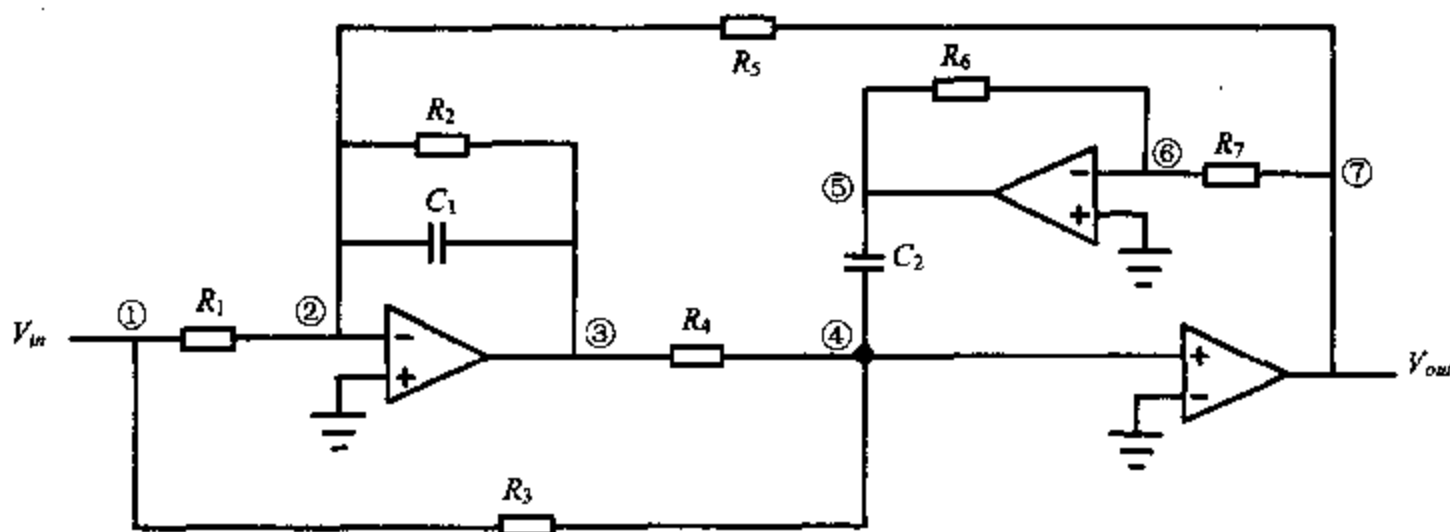


图 3.8 两级有源滤波器

例 3.3 用 CMNA 法建立图 3.9 所示的两级开关电容滤波器的 z 域符号电路方程组。图 3.9 中的 $\phi 1$ 和 $\phi 2$ 是两相不重叠时钟。

我们假设开关和运算放大器都是理想的。理想放大器的含义参见例 3.2; 理想开关指其导通电阻为零, 而关断电阻为无穷大。如果采用改进节点法, 则需要一个 98×98 维的系数矩阵来描述其电路特性^[121]。如果采用 CMNA 法, 未知变量只有 $\phi 1$ 相时钟时节点 2, 4 的电压 $V_{2,1}, V_{4,1}$ 和 $\phi 2$ 相时钟时节点 2, 4 的电压 $V_{2,2}$ 和 $V_{4,2}$; 电荷守恒方程也只有 4 个: 在 $\phi 1$ 相时钟时节点 1, 3 处的电荷守恒方程和在 $\phi 2$ 相时钟时节点 1, 3 处的电荷守恒方程。下面是用 CMNA 法所获得的 z 域符号电路方程组:

$$\begin{bmatrix} -C_5 - C_6 & -C_4 & z^{-1}C_5 & z^{-1}C_4 \\ -C_{11} - C_{12} & 0 & C_{12} & C_7 \\ C_5 & C_4 & -C_5 & -C_4 \\ C_{12} & 0 & -C_{12} & 0 \end{bmatrix} \begin{bmatrix} V_{out,1} \\ V_{2,1} \\ V_{out,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} -(C_1 + C_3)V_{in,1} + (C_2 + C_3)z^{-1}V_{in,2} \\ -(C_{10} + C_3)V_{in,1} + (C_{10} + C_3)z^{-1}V_{in,2} \\ C_3V_{in,1} - C_3V_{in,2} \\ C_{10}V_{in,1} - C_{10}V_{in,2} \end{bmatrix}$$

其中, $V_{in,1}$ 和 $V_{in,2}$ 分别表示输入信号 V_{in} 在 $\phi 1$ 相时钟和 $\phi 2$ 相时钟时的值, 且 $V_{out,1} = V_{4,1}$, $V_{out,2} = V_{4,2}$ 。

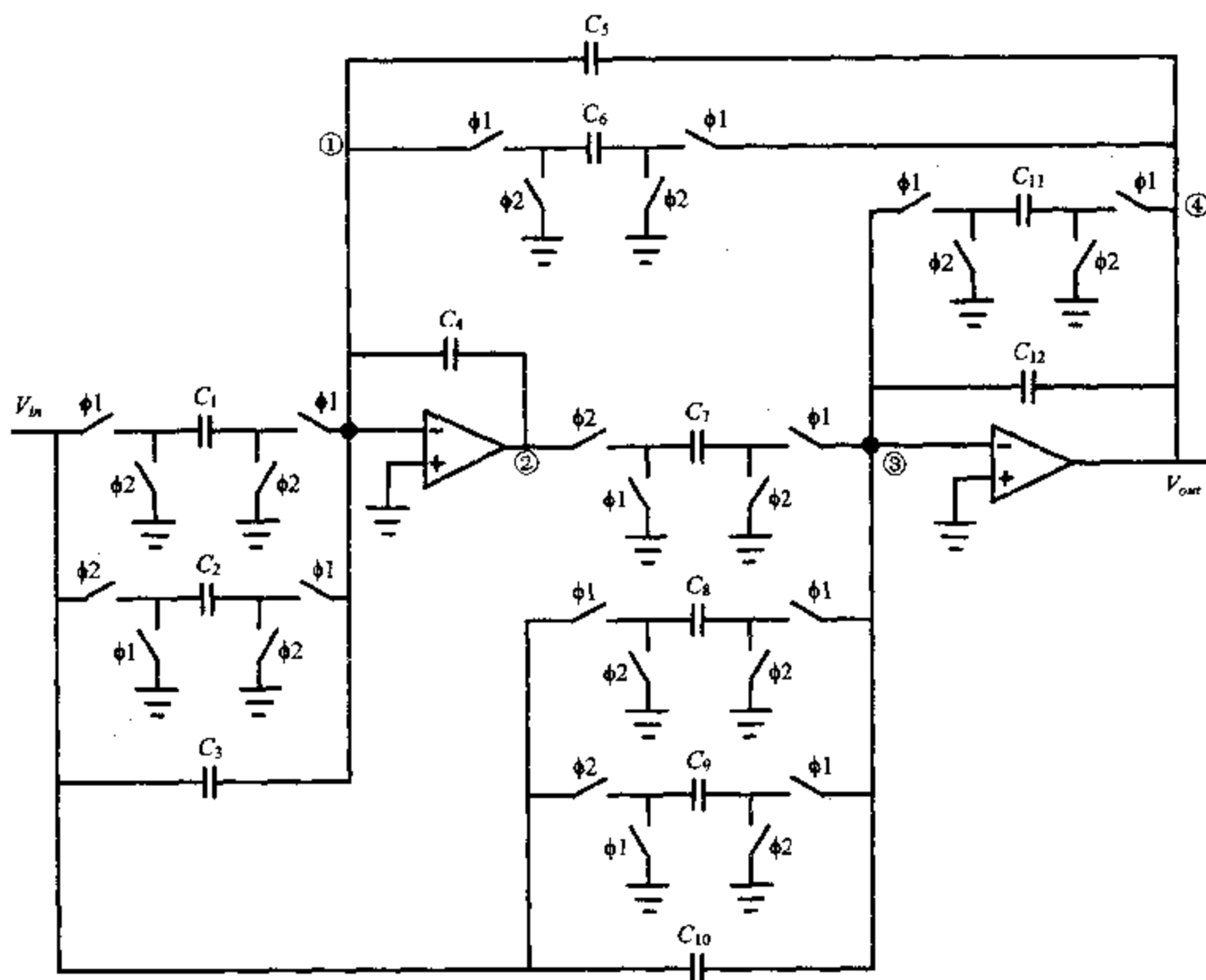


图 3.9 两级开关电容滤波器

3.3.4 器件失配

我们在第 1.1 节分析模拟集成电路设计的特征时已经明确指出：工艺涨落对模拟集成电路性能的影响非常大。例如：如果设计者需要图 3.8 中的电阻 R_1 和 R_2 具有相同的阻值，但由于生产过程中存在各种涨落因素， R_1 和 R_2 是不可能绝对相等的，集成电路设计者通常将该情形称为 R_1 和 R_2 之间出现了失配。为保证电路有足够高的合格率或性能指标能满足要求，电路设计者非常迫切地需要了解 R_1 和 R_2 的失配程度对电路性能的影响。为此，只要选 R_1 (或 R_2) 为参照物，则 R_2 和 R_1 的失配程度 ξ 定义为：

$$\xi \equiv \frac{R_2 - R_1}{R_1}$$

在用 CMNA 法建立符号电路方程组时，只要将符号参数 R_2 置换为 $(1+\xi)R_1$ 即可获得下述符号方程组：

$$\begin{bmatrix} (1+\xi)^{-1}R_1^{-1} + sC_1 & 0 & R_5^{-1} \\ R_4^{-1} & sC_2 & 0 \\ 0 & R_6^{-1} & R_7^{-1} \end{bmatrix} \begin{bmatrix} V_3 \\ V_5 \\ V_7 \end{bmatrix} = - \begin{bmatrix} R_1^{-1} \\ R_3^{-1} \\ 0 \end{bmatrix} V_{in}, \quad \text{其中, } V_7 = V_{out}$$

从上式求出的传输函数将包含 R_2 和 R_1 的失配程度 ξ ，只要分析该符号函数随符号变量 ξ 的变化情况就可以准确把握 R_2 和 R_1 的失配程度对电路性能的影响。

3.4 符号线性方程组的求解方法

前一节介绍了如何用 CMNA 法建立线性(或弱非线性)时不变电路的 s 域和 z 域符号方程组, 本节将详细讨论求解这些符号方程组的方法。

3.4.1 符号求解法的基本思路

符号线性方程组(3.9)和(3.11)都可以表示为下述一般形式:

$$\mathbf{A}_{n \times n} \mathbf{V}_{n \times 1} = \mathbf{B}_{n \times 1} \quad (3.14)$$

其中, n 是符号线性方程组的维数, $\mathbf{A}$ 是系数矩阵, $\mathbf{V}$ 是未知向量, 常数向量 $\mathbf{B}$ 称为右端向量。

众所周知, 当 $\mathbf{A}$ 和 $\mathbf{B}$ 都是纯数值的情形, 可以采用消元法或 LU 分解法等效率非常高的数值算法来获得式(3.14)的数值解^[106]。但对于符号线性方程组而言, 消元法或 LU 分解法将产生许多中间项, 合并或消去这些中间项是非常费时的; 消元法和 LU 分解法的另一弱点是: 当我们只关心 $\mathbf{V}$ 的个别分量时, 它们都必须同时计算 $\mathbf{V}$ 的其他分量, 对于求电路的传输函数而言, 它们都会浪费许多不必要的 CPU 时间。因此, 目前一般不采用这两种方法来求解符号线性方程组。

在求解符号线性方程组时, 克莱姆(Cramer)法则^[123]是较常用的方法。根据克莱姆法则, 如果矩阵 $\mathbf{A}$ 是非奇异的, 即 $|\mathbf{A}| \neq 0$, 则方程组(3.14)的未知变量 v_j (向量 $\mathbf{V}$ 的第 j 个分量) 可以表示为:

$$v_j = \frac{|\mathbf{C}_{(j)}|}{|\mathbf{A}|}, \quad j = 1, 2, \dots, n \quad (3.15)$$

其中, $\mathbf{C}_{(j)}$ 也是 $n \times n$ 阶矩阵, 用右端向量 $\mathbf{B}$ 置换系数矩阵 $\mathbf{A}$ 的第 j 列即可得到该矩阵。因此, 从输入信号 v_i 到输出信号 v_o 的传输函数 $H(x)$ 为:

$$H(x) \equiv \frac{v_o}{v_i} = \frac{|\mathbf{C}_{(o)}|}{|\mathbf{C}_{(i)}|}, \quad (1 \leq i, o \leq n) \quad (3.16)$$

仔细观察式(3.16)和(3.15)后不难看出: 从电路的符号方程组(3.14)求传输函数的关键是如何计算符号矩阵的行列式值。由于矩阵 $\mathbf{A}$ 具有稀疏性, $\mathbf{C}_{(o)}$ 和 $\mathbf{C}_{(i)}$ 在通常情况下也是稀疏矩阵, 应当充分利用这一特性以减小计算量。

3.4.2 排列法

根据定义, $n \times n$ 阶矩阵 $\mathbf{A}$ 的行列式值为^[123]:

$$|\mathbf{A}| = \sum_{(j_1, j_2, \dots, j_n)} (-1)^{\tau(j_1, j_2, \dots, j_n)} a_{1, j_1} a_{2, j_2} \cdots a_{n, j_n} \quad (3.17)$$

其中, $a_{i, j}$ 是 $\mathbf{A}$ 的第 i 行第 j 列元素, $\tau(j_1, j_2, \dots, j_n)$ 是 n 元排列 $(j_1, j_2, \dots, j_n)$ 的逆序, 而求和是对自然数 $1, 2, \dots, n$ 的所有 n 元排列进行的。式(3.17)中乘积不为零的项所对应的排列通常称为非零排列。

直接采用式(3.17)来计算行列式值的计算复杂度为 $O(n!)$, 当矩阵的阶次 n 较大时计算量太大。很显然, 如果矩阵 $\mathbf{A}$ 是稀疏矩阵, 则只需对式(3.17)中的非零排列求和即可。A. Liberator 提出了一种求式(3.17)中非零排列的递归算法^[124], 该算法利用矩阵的

稀疏性大大减少了式(3.17)的计算量。图 3.10 给出了该算法的伪代码 gen.nonzero.permutation(), 其输入是一个列表 colnrlist, 该列表的第 k 个元素的初值是矩阵 A 的第 k 行中非零元素的列号, 矩阵 A 的非零排列将输出到列表 output.list 中, 而 level 则是递归的深度。

```

gen_nonzero_permutation (colnrlist, level) {
  var colnr, list_buffer, n;
  n ← colnrlist 包含的元素数目;
  WHILE (colnrlist 不是空列表) {
    IF (level 等于 n) {
      将 colnrlist 中的元素输出到 output_list 中;
    } ELSE {
      FOR (colnrlist[level] 中的每一个列号 colnum) {
        list_buffer ← colnrlist;
        list_buffer[level] ← colnum;
        删除 list_buffer[level+1], list_buffer[level+2], ..., list_buffer[n] 中的行号 colnum;
        gen_nonzero_permutation (list_buffer, level + 1);
      }
    }
  }
}

```

图 3.10 求非零排列的递归算法

利用 gen.nonzero.permutation() 所输出的非零排列, 可以很容易地获得矩阵 A 的行列式值。

例 3.4 考虑下述 4×4 阶矩阵

$$\begin{bmatrix} A & B & 0 & 0 \\ 0 & C & D & E \\ 0 & 0 & F & G \\ H & I & J & 0 \end{bmatrix}$$

当采用递归算法 gen.nonzero.permutation() 求其非零排列时, 列表 colnrlist 的初值为:

$$\text{colnrlist} = ((1, 2), (2, 3, 4), (3, 4), (1, 2, 3))$$

而递归算法 gen.nonzero.permutation() 输出的非零排列的列表 output.list 为:

$$\text{output.list} = ((1, 2, 4, 3), (1, 3, 4, 2), (1, 4, 3, 2), \\ (2, 3, 4, 1), (2, 4, 3, 1))$$

因此, 该矩阵的行列式值为:

$$-A \cdot C \cdot G \cdot J + A \cdot D \cdot G \cdot I - A \cdot E \cdot F \cdot I - B \cdot D \cdot G \cdot H + B \cdot E \cdot F \cdot H$$

3.4.3 递归展开法

根据拉普拉斯(Laplace)定理^[123], $n \times n$ 阶矩阵 A 的行列式值等于它的所有 k ($1 \leq k < n$) 阶子式的行列式值与相应的代数余子式的乘积之和, 即:

$$|A| = \sum_{M_k} |M_k| |M'_k| \quad (3.18)$$

其中, M_k 是 A 的一个 k 阶子式, M'_k 是对应于 M_k 的代数余子式, 求和对所有 k 阶子式进行。当 $k=1$ 时, 则意味着可以按任意行或列进行展开, 因此, 式(3.18)就等价于:

$$|A| = \sum_{\substack{j=1 \\ a_{ij} \neq 0}}^n (-1)^{i+j} a_{ij} |M_{ij}|, \quad \forall i \in \{1, 2, \dots, n\}$$

或

$$|A| = \sum_{\substack{i=1 \\ a_{ij} \neq 0}}^n (-1)^{i+j} a_{ij} |M_{ij}|, \quad \forall j \in \{1, 2, \dots, n\} \quad (3.19)$$

其中, M_{ij} 称为 a_{ij} 的余子式, 它是划去 A 的第 i 行和第 j 列元素后所得到的矩阵。

很显然, 计算 M_{ij} 的行列式值时还可以继续采用上述方法, 直到所有余子式的阶次为 2 为止。因此, 可以按 (3.19) 采用递归的方式计算符号矩阵 A 的行列式值。我们称这种方法为递归展开法。但需要特别注意的是: 某些余子式在递归展开过程中可能出现多次, 为避免重复计算, 应当记忆已经计算过的余子式。图 3.11 给出了具有余子式记忆能力的递归展开算法 `laplace_expansion_with_minors_memo()` 的伪代码。

例 3.5 用递归展开法求例 3.4 中的行列式值。

下面是按矩阵的第 1 列元素展开的具体过程:

$$\begin{aligned} \begin{bmatrix} A & B & 0 & 0 \\ 0 & C & D & E \\ 0 & 0 & F & G \\ H & I & J & 0 \end{bmatrix} &= A \begin{bmatrix} C & D & E \\ 0 & F & G \\ I & J & 0 \end{bmatrix} - H \begin{bmatrix} B & 0 & 0 \\ C & D & E \\ 0 & F & G \end{bmatrix} \\ &= A \left[C \begin{bmatrix} F & G \\ J & 0 \end{bmatrix} + I \begin{bmatrix} D & E \\ F & G \end{bmatrix} \right] - H \left[B \begin{bmatrix} D & E \\ F & G \end{bmatrix} - C \begin{bmatrix} 0 & 0 \\ F & G \end{bmatrix} \right] \\ &= -A \cdot C \cdot G \cdot J + A \cdot I \cdot D \cdot G - A \cdot I \cdot E \cdot F \\ &\quad - H \cdot B \cdot D \cdot G + H \cdot B \cdot E \cdot F \end{aligned}$$

在上述计算过程中, 余子式 $\begin{bmatrix} D & E \\ F & G \end{bmatrix}$ 虽然出现了两次, 但由于记忆了余子式, 因而只需计算一次即可。

```

laplace_expansion_with_minors_memo(matrix) {
    var det, minor, a_ij, i, j, row_i, coln_j;
    det ← 0;
    确定按第 i 行或第 j 列展开;
    FOR (矩阵 matrix 的第 i 行或第 j 列的所有非零元素 a_ij) {
        IF (a_ij 的余子式 minor_ij 已存在于记忆表 minor_list 中) {
            det ← det + (-1)i+j * a_ij * minor_ij 的行列式值;
        } ELSE {
            将第 i 行元素存入 row_i 中;
            将第 j 列元素存入 coln_j 中;
            删去矩阵 minor_ij 的第 i 行和第 j 列元素;
            minor ← laplace_expansion_with_minors_memo(matrix);
            det ← det + (-1)i+j * a_ij * minor 的行列式值;
            将 minor 存入记忆表 minor_list 中;
            将 row_i 和 coln_j 分别加入到 minor 的第 i 行和第 j 列;
        }
    }
    return det;
}

```

图 3.11 具有余子式记忆功能的递归展开算法

3.4.4 动态行列选择

细心的读者或许已经注意到算法 `laplace.expansion.with.minors.memo()` 中还存在一个“悬而未决”的核心问题，确定按第 i 行或按第 j 列展开的具体准则是什么？不言而喻，该准则应当有利于减少计算量，而准则的类型也不外乎静态准则和动态准则两类。

静态准则首先将矩阵的列(行)号按相应列(行)中非零元素的个数进行升序排列，即：将列(行)中非零元素最少列(行)号排在最前面，将列(行)中非零元素最大的列(行)号排在最后，假设所获得的序列为 $j_1, j_2, \dots, j_n$ ，然后按 j_m 列展开 $(n-m)$ 阶余子式。该方法在第一次展开时产生的余子式数目最少，有利于减少计算量，但在求后续余子式的行列式值时，该优越性就不复存在了。静态准则的优点是简单、易于实现，其不足是会产生一些为零的乘积项，因此效率不高。

动态准则是每次计算余子式的行列式值时，都按该余子式中非零元最少的列(行)进行展开。动态准则每次都要寻找余子式中非零元最少的列(行)，实现起来虽然比较麻烦，但很少产生乘积为零的项，因此计算效率比较高。

为进一步展示静态准则与动态准则的差异，让我们考察下述矩阵：

$$\begin{bmatrix} 0 & A & B & C & D \\ 0 & E & 0 & F & 0 \\ G & 0 & 0 & H & I \\ J & 0 & 0 & 0 & 0 \\ K & L & 0 & 0 & M \end{bmatrix}$$

如果采用静态准则，由于非零元最少的列号为 3，其次为 1, 2, 4, 5，因此，首先按第 3 列展开，然后按第 1 列展开，…。所获得的行列式值为：

$$\begin{aligned} & B \cdot \{-G \cdot [E \cdot 0 + L \cdot 0] + J \cdot [E \cdot (H \cdot M - I \cdot 0) + L \cdot (F \cdot I - H \cdot 0)] \\ & \quad - K \cdot [E \cdot (H \cdot 0 - I \cdot 0)]\} \\ & = B \cdot J \cdot E \cdot H \cdot M + B \cdot J \cdot L \cdot F \cdot I \end{aligned}$$

如果采用动态准则，首次依然按第 3 列展开，并获得下述余子式：

$$\begin{bmatrix} 0 & E & F & 0 \\ G & 0 & H & I \\ J & 0 & 0 & 0 \\ K & L & 0 & M \end{bmatrix}$$

在继续计算该余子式的行列式值时，由于该余子式第 3 行的非零元最少，因此，应当按第 3 行展开，…。所获得的行列式值为：

$$\begin{aligned} & B \cdot J \cdot [E \cdot (H \cdot M - I \cdot 0) - F \cdot (0 - I \cdot L)] \\ & = B \cdot J \cdot E \cdot H \cdot M + B \cdot J \cdot L \cdot F \cdot I \end{aligned}$$

由此可见，动态行列选择所产生的中间项比静态准则少许多，是减少计算量的有效措施。

3.4.5 数值解法与符号解法

众所周知,各种消元法和 LU 分解法是常用的线性方程组的数值解法,这类数值算法的核心问题是保证数值算法的稳定性和计算效率,而行主元选择和列主元选择是行之有效的常用技术^[106]。对于符号分析而言,通常只关心个别节点电压或支路电流之间的关系,没有必要求出所有的未知变量。因此,消元法和 LU 分解法都不适合于求解符号线性方程组。另外,用消元法或 LU 分解法求解符号线性方程组时,还会产生许多中间项,需要对最后结果进行同类项的合并,将增加许多额外的计算量,这是消元法和 LU 分解法最大的弱点。

求解符号线性方程组时已经不存在数值不稳定的问题,求行列式的值是符号解法的核心问题。利用系数矩阵的稀疏性来减少计算量是主要的手段。本节介绍的算法 `gen_nonzero.permutation()` 和 `laplace.expansion.with.minors.memo()` 都采用了该手段,而动态行列选择技术则是进一步减小递归展开法计算量的有效措施。这两种方法所产生的中间项很少,因为,同一元件仅出现在系数矩阵的个别列(或行)中。

细心的读者一定会注意到:求式(3.16)中的传输函数 $H(x)$ 时,分子、分母中的两个行列式只有两列不同,而且

$$H(x) = \frac{|C_{(o)}|}{|C_{(j)}|} = - \frac{|[A_j \quad B \quad A_{-(o,j)}]|}{|[A_o \quad B \quad A_{-(o,j)}]|} \quad (3.20)$$

其中, A_j 和 A_o 分别是由式(3.14)中系数矩阵 A 的第 j 列和第 o 列元素所构成的列向量, B 依然是(3.14)中的右端向量,而 $A_{-(o,j)}$ 则是从系数矩阵 A 中划去第 o 列和第 j 列后所剩下的部分。

因此,可以采用式(3.20)来计算(3.16)所定义的传输函数 $H(x)$ 。在计算(3.20)中分子、分母的行列式值时,如果第一次都按各自的第 1 列展开则可以大大减少计算量,因为分子、分母中的余子式有许多是相同的。当然,在计算这些余子式的行列式值时,仍需采用动态行列选择技术和具有余子式记忆功能的递归展开算法。

3.5 符号逼近法

对于实际电路,先按第 3.3 节的 CMNA 法建立其符号线性方程之后,然后再用第 3.4 节的算法获得传输函数 $H(x)$,以这种方式所获得的符号表达式通常都特别长,其中许多项是不重要的,它们的存在反而会影响设计者认识电路的本质。符号逼近的任务就是剔除这些项,使符号表达式更加简洁、明了。

3.5.1 数值逼近与符号逼近

逼近理论在数值分析领域有着特别重要的作用^[106],数值逼近问题一般是按下述方式提出的:

给定函数 $g(x)$, $x \in \Omega$, 从定义于 Ω 上的函数集合 Γ_n (例如阶次不高于 n 的多项式或连分式)中寻找函数 $f(x)$, 使 $f(x)$ 对 $g(x)$ 的逼近误差(通常用某种范数来定义)尽可能

低或不超过最大误差限 $\epsilon_{\max}$ 。图 3.12 是一元函数的数值逼近实例。

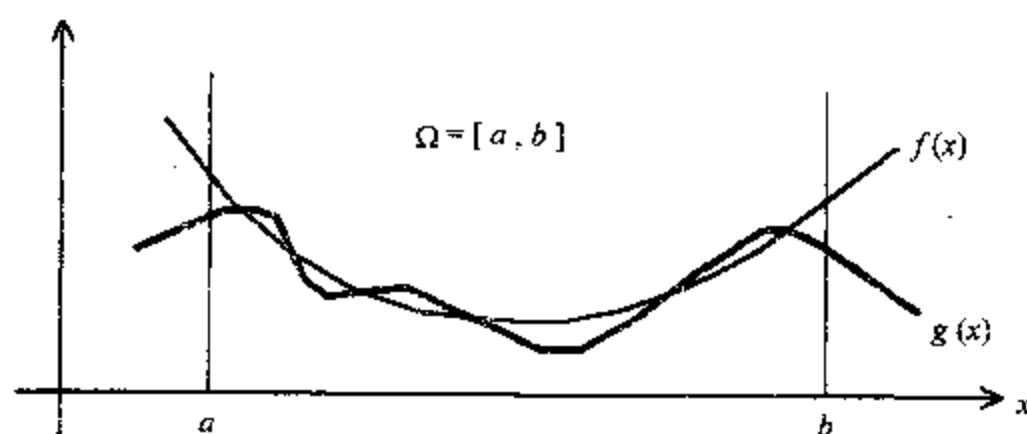


图 3.12 数值逼近示意图

然而，在模拟集成电路设计自动化领域，符号逼近的任务主要是减小符号表达式的复杂度，即：在给定误差要求的前提下，使式(3.1)的分子、分母包含尽可能少的项。在本书中，符号表达式的复杂度指符号表达式中所包含的项数。例如：按式(3.1)所定义的传输函数 $H(x)$ 的复杂度为 $(m+n+2)$ ，而式(3.2)和(3.3)所表示的输出阻抗的复杂度分别为 4 和 2。

不言而喻，增加符号表达式的复杂度可以减小逼近误差，但却会增加计算量。因此，在进行符号逼近时，需要对符号表达式的复杂度和误差作折衷处理。图 3.13 是复杂度与误差关系的直观图像(又称 C- ϵ 图像)，图 3.13 还给出了符号逼近法的 3 个轨迹(简称“逼近轨迹”)，我们在稍后将会看到，逼近误差的度量方式决定了逼近轨迹。

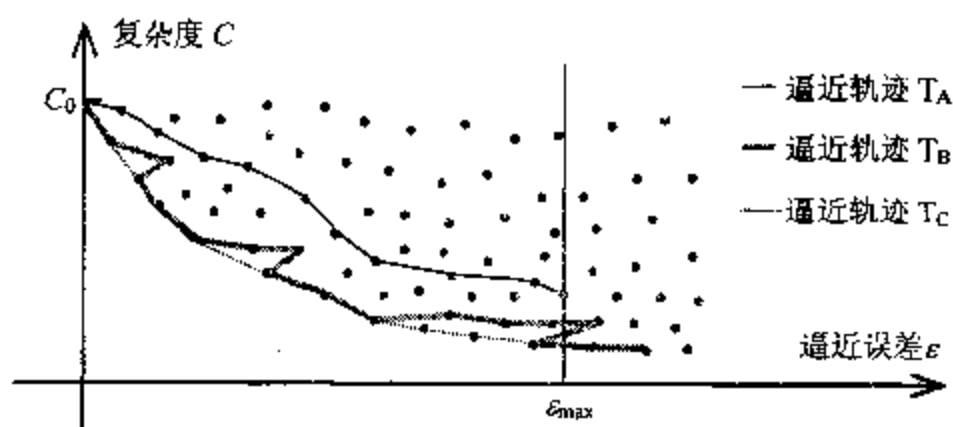


图 3.13 C- ϵ 图像及逼近轨迹示意图

由于符号逼近法的目的是为电路设计者或模拟集成电路自动综合系统提供精确、复杂度低的电路性能的符号表达式，因此，符号逼近法通常应具备下述功能：

(1) 能允许用户十分方便地控制逼近误差。用户不仅可以给定一个最大误差限 $\epsilon_{\max}$ ，而且还可以随时增大或减小逼近误差，以便能综合考虑符号表达式的复杂度和精度这两个因素，因为用户通常希望得到一个一阶模型(误差最大)以分析影响电路性能的主要因素，同时还希望得到二阶或高阶模型(精度更高一些)，以便验证所设计的电路是否满足性能指标的要求。

(2) 模拟集成电路的特性通常与工作频率密切相关，用户有时只关心直流或低频特性，有时只关心高频特性，有时只关心中心频率 f_0 附近的特性，而有时可能需要考虑整

个频域的特性,因此,符号逼近法应当允许用户方便地指定他感兴趣的频率范围。

(3) 符号逼近的结果应当与符号表达式中各项的顺序无关。

(4) 由于需要根据电路参数(例如:器件的几何尺寸、SPICE 模型参数等)来估计符号表达式中每一项的幅度,而电路参数总会受工艺涨落因素的影响,因此,符号逼近法不应当对电路参数十分敏感,否则,符号逼近法所获得的结果将背离实际情况。

(5) 在满足逼近误差的条件下,符号表达式应当为用户提供尽可能多的信息,因此,符号逼近法不能将 $\epsilon_{\max}$ 简单地理解为一个数值参数。在符号逼近过程中通常会遇到若干项的幅度都相等的情况,从符号表达式中删去其中几项时逼近误差仍然满足要求,但将它们全部删去后,逼近误差将超过 $\epsilon_{\max}$ 。在这种情形下,正确的选择是将这些项全部保留下来,以便为用户提供尽可能丰富的信息。更一般的准则是:如果某一项出现在最后的符号表达式中,则幅度不低于该项的所有项必须全部保留下来。

因此,在定义逼近误差和选择符号逼近算法时,都应当考虑上述 5 个方面的因素,本节后续部分将逐一讨论这些内容。

3.5.2 逼近误差

假设原始符号表达式为 $g(P, x)$, $P \in \Omega$, P 是由电路参数所构成的向量, x 为复频率,即我们常用的 s 或 z ; $f(P, x)$ 是对 $g(P, x)$ 的符号逼近,它们都是形如式(3.1)的符号表达式。逼近误差是刻画 $f(P, x)$ 对 $g(P, x)$ 的逼近程度的唯一指标,常用的逼近误差有最坏情况误差、标称误差(又称累计有效误差)和累计绝对误差。

1. 最坏情况误差

如果用户感兴趣的复频率范围为 $[x_l, x_u]$, 则用 $f(P, x)$ 逼近 $g(P, x)$ 的最坏情况误差 ϵ_w 定义为:

$$\epsilon_w \equiv \max_{\substack{P \in \Omega \\ x \in [x_l, x_u]}} \left| \frac{g(P, x) - f(P, x)}{g(P, x)} \right| \quad (3.21)$$

采用最坏情况误差来定义 $f(P)$ 对 $g(P)$ 的逼近程度时需要搜索 Ω 中的所有点,因此,计算量非常大。另外,最坏情况误差也过于悲观了一些,不利于简化符号表达式。

2. 标称误差

如果用户已经完成了一个电路设计,我们将电路参数的设计值(又称标称值)记为 P_0 , 则用 $f(P, x)$ 逼近 $g(P, x)$ 的标称误差 ϵ_N 定义为:

$$\epsilon_N \equiv \max_{x \in [x_l, x_u]} \left| \frac{g(P_0, x) - f(P_0, x)}{g(P_0, x)} \right| \quad (3.22)$$

标称误差又称累计有效误差,它的计算量比较小,也比较客观地反映了 $f(P, x)$ 对 $g(P, x)$ 的逼近程度。即使用户还没有完成电路设计,也可以用标称误差来度量逼近程度,在这种情况下,通常用估计值 P_e 来代替电路参数的标称值 P_0 。

如果我们将 $g(P, x)$ 中的各项按幅度大小的顺序排列起来,在进行符号逼近时,先删除幅度最小的项,然后删除剩余项中幅度最小的项,如此往复,直到不能再删除新的

项为止,那么,当被删除的项既有大于零的项、也有小于零的项时,逼近轨迹将出现许多Z字形(参见图3.13的逼近轨迹 T_B)。因此,标称误差和复杂度之间没有一一对应的关系,不同的逼近公式可能具有相同的标称误差,同时,标称误差对逼近算法的控制不够严格,所获得的逼近结果与 $f(P, x)$ 中各项的先后顺序有关。这是模拟集成电路设计者和模拟集成电路自动化设计工具的开发人员都不愿看到的局面。

3. 累计绝对误差

为解决最坏情况误差和标称误差的不足, H. Walscharts 提出了用累计绝对误差来度量逼近程度的思想^[121]。假设 $g(P, x)$ 所包含的所有项为 $T_j(P, x)$, $j \in N_A$, $f(P, x)$ 是从 $g(P, x)$ 中删除项 $T_d(P, x)$ 后所获得的结果,其中, $d \in N_D$, $N_D \subset N_A$, 则用 $f(P, x)$ 逼近 $g(P, x)$ 的累计绝对误差 ϵ_A 定义为:

$$\epsilon_A = \frac{\max_{x \in [x_l, x_u]} \sum_{d \in N_D} |T_d(P_0, x)|}{\max_{x \in [x_l, x_u]} \sum_{j \in N_A} |T_j(P_0, x)|} \quad (3.23)$$

很显然,累计绝对误差与标称误差的重要区别在于:累计绝对误差是将所删除的项的绝对值相加,删除的项越多,累计绝对误差一定会越大。因此,累计绝对误差和符号表达式的复杂度之间具有一一对应的关系,而逼近轨迹也是平滑、单调的曲线(参见图3.13的逼近轨迹 T_C)。

应当看到:标称误差是最精确的度量尺度,但对电路参数非常敏感;累计绝对误差的估计结果稍微偏大一些,但对电路参数不太敏感,而且能保证符号逼近结果与原符号表达式中各项的先后顺序无关,因而有较好的稳定性;最坏情况误差则最为保守,计算量也最大。因此,一般的符号分析系统都同时采用标称误差和累计绝对误差这两种度量方式,它们各有千秋,并可以让用户指定;只有最坏情况分析系统才采用最坏情况误差。

3.5.3 逼近算法

符号逼近的本质是如何删除式(3.1)中那些不太重要的项,不同逼近算法之间的区别主要在于选择被删除项的策略,我们可以将这些策略分为三类:全局策略、局部策略和混合策略。

1. 全局策略

全局策略是令 $f(P, x) = g(P, x)$,接着从 $f(P, x)$ 中选出一项,并删除之;然后再从 $f(P, x)$ 的剩余项中选出一项,并删除之;重复该过程直到逼近误差达到用户给定的最大误差限 $\epsilon_{\max}$ 为止,即:

$$\min C_F, \quad s. t. \epsilon_F \leq \epsilon_{\max} \quad (3.24)$$

其中, C_F 和 ϵ_F 分别是最终的符号逼近表达式的复杂度和逼近误差。

从式(3.24)不难看出,在给定逼近误差要求的情况下,使符号表达式的复杂度最低是选择被删除项的准则。我们可以按下述方式进行:

首先选择采用什么逼近误差(标称误差或累计绝对误差),并根据 $g(P, x)$ 中各项的

幅度大小, 将它们按升序方式排列起来, 然后每次都简单地删除第 1 项, 直到逼近误差达到 $\epsilon_{\max}$ 为止。

基于全局策略的符号逼近法需要把 $g(P, x)$ 中的各项按幅度大小排列起来, 这需要花费额外的 CPU 时间。由于排序的计算量为 $O(C_0 \log(C_0))$, C_0 是 $g(P, x)$ 的复杂度, 因此当 $g(P, x)$ 包含的项数很多时, 额外计算量会相当大。

2. 局部策略

由于符号逼近的目的是获得逼近误差满足要求且复杂度最低的表达式, 至于按什么顺序删除 $g(P, x)$ 中那些不足轻重的项并不重要, 因此可以采用局部策略来选择被删除的项。局部策略虽然计算速度很快, 但却很难预估最终的逼近误差和符号表达式的复杂度^[12]。当用户给定 $\epsilon_{\max}$ 之后, 我们总是希望采用局部策略的最终结果对应于 C - ϵ 图像上的点 X , X 是累计绝对误差逼近轨迹与垂直线 $\epsilon = \epsilon_{\max}$ 的交点, 参见图 3.14。

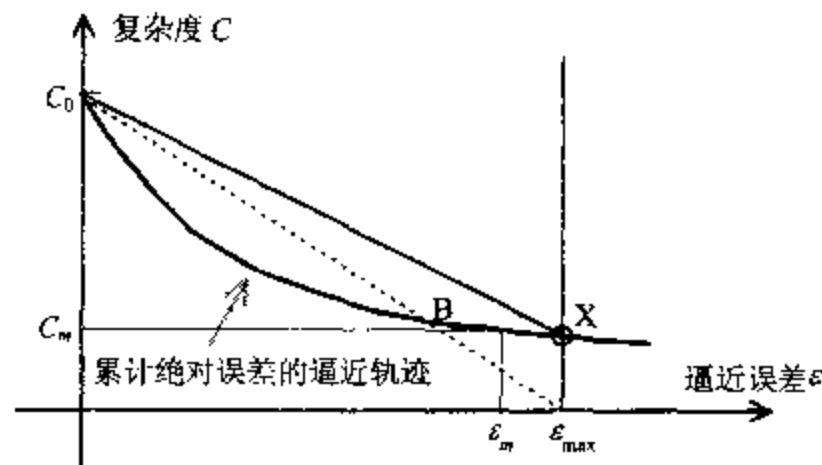


图 3.14 局部策略的由来

局部策略依然是每次删除一项, 我们假定经过 m 次逼近后就达到了最终结果, 且其累计绝对误差为 ϵ_m , 复杂度为 C_m 。如果下式成立, 则可以保证最终结果对应于 C - ϵ 图像上的点 X 或位于累计绝对逼近轨迹上且靠近点 X 。

$$\left| \frac{dC}{d\epsilon} \right| = \left| \frac{C_{j+1} - C_j}{\epsilon_{j+1} - \epsilon_j} \right| \geq \left| \frac{C_m - C_0}{\epsilon_m - \epsilon_0} \right|, \quad 1 \leq j < m \quad (3.25)$$

其中, C_j 和 ϵ_j 分别表示第 j 次逼近结束后符号表达式的复杂度和逼近误差。虽然在逼近过程中 C_m 是未知的, 但是, 考虑到 $C_m > 0$, $C_{j+1} - C_j = 1$, $\epsilon_0 = 0$, 且 $\epsilon_m \leq \epsilon_{\max}$, 则式(3.25)可以简化为:

$$|\epsilon_{j+1} - \epsilon_j| \leq \frac{\epsilon_{\max}}{C_0} \quad (3.26)$$

因此, 如果采用累计绝对误差来度量符号逼近的程度, 则第 j 次从 $g(P, x)$ 中选出的项 $T_j(P, x)$ 是可以删除的, 当且仅当下式成立:

$$\max_{x \in [x_l, x_u]} |T_j(P_0, x)| \leq \frac{\epsilon_{\max}}{C_0} \sum_{j \in N_A} \max_{x \in [x_l, x_u]} T_j(P_0, x) \quad (3.27)$$

其中, $\frac{1}{C_0} \sum_{j \in N_A} \max_{x \in [x_l, x_u]} T_j(P_0, x)$ 通常又称为符号函数 $g(P, x)$ 在 P_0 处的“项平均值”(因

为 C_0 是 $g(P, x)$ 所包含的项数), 为后面叙述方便, 我们将它记为 $E_{\text{term}}(\cdot)$, 即:

$$E_{\text{term}}(g(P_0, x)) \equiv \frac{1}{C_0} \sum_{j \in N_A} \max_{x \in [x_l, x_u]} T_j(P_0, x)$$

按上述准则删除 $g(P, x)$ 中不足轻重的项可以确保逼近轨迹与累计绝对误差的逼近轨迹重合, 在给定 $\epsilon_{\max}$ 后, 最终所获得的符号表达式的复杂度是最低的; 在一般情况下, 式 (3.27) 是一种安全估计, 最终的逼近误差 ϵ_m 常常比 $\epsilon_{\max}$ 小很多。因此, 采用局部策略的最终逼近结果可能位于累计绝对误差的逼近曲线上, 但在点 X 的前面且远离点 X。实质上, 逼近结果一般位于图 3.14 中点 B 的左边。

特别值得一提的是: 如果 $T_j(P_0, x)$ 中有的项为正、有的项为负, 则式 (3.27) 对删除项的约束就显得太松了, 会出现逼近误差大于 $\epsilon_{\max}$ 的情况。因此, 在局部策略中, 被删除项 $T_j(P, x)$ 除满足 (3.27) 之外, 还应当满足下述准则:

$$\max_{x \in [x_l, x_u]} |T_j(P_0, x)| \leq \epsilon_{\max} \max_{x \in [x_l, x_u]} |g(P_0, x)| \quad (3.28)$$

上述准则是显而易见的, 否则, 删除 $T_j(P, x)$ 后将使标称误差大于 $\epsilon_{\max}$ 。

3. 混合策略

H. Walscharts 将局部策略和求解策略结合起来^[121], 先遍历符号表达式 $g(P, x)$ 的每一项 $T_j(P, x)$, 如果 $T_j(P, x)$ 满足式 (3.27), 则从 $g(P, x)$ 中删除 $T_j(P, x)$, 这样便可得到一个中间结果 $g^*(P, x)$; 然后采用全局策略将 $g^*(P, x)$ 简化为最终结果 $f(P, x)$ 。由于 $g^*(P, x)$ 比 $g(P, x)$ 少很多项, 因此采用全局策略时所花费的排序时间会减少很多; 该混合策略还能确保最终结果 $f(P, x)$ 对应于图 3.14 中的点 X。

如果我们将符号表达式中具有相同幅度的项称为一个“串”, 则许多实际电路的符号表达式 $g(P, x)$ 通常都包含多个“串”, 因此, 上述混合策略有可能将某个“串”中的部分项删除了, 但由于受最大逼近误差 $\epsilon_{\max}$ 的约束, 又必须保留该“串”的另外一些项, 因而不能为电路设计者保留足够多的信息。为了满足“如果某一项出现在最后的符号表达式中, 则幅度不低于该项的所有项必须全部保留下来”这一准则, 还需要将 $g(P, x)$ 中满足下述条件的项 $T_r(P, x)$ 放回到 $f(P, x)$ 中:

$$\max_{x \in [x_l, x_u]} |T_r(P_0, x)| \geq \frac{\max_{x \in [x_l, x_u]} |T_{\text{last}}(P_0, x)|}{\lambda} \quad (3.29)$$

其中, 参数 λ 称为回收因子, 它一般在 1 到 4 之间^[79]; $T_{\text{last}}(P, x)$ 是最后一个因不满足式 (3.28) 或最大误差要求的项。

在采用全局策略完成符号逼近后又回收一些项, 这势必会导致逼近误差大于 $\epsilon_{\max}$ 的局面, 为此, 我们对局部策略作更为严格的限制, 即将 (3.27) 改变为:

$$\max_{x \in [x_l, x_u]} |T_j(P_0, x)| \leq \frac{1}{\lambda} \epsilon_{\max} E_{\text{term}}(g(P_0, x)) \quad (3.30)$$

图 3.15 给出了包含上述各种混合策略的符号逼近算法的伪代码 symbolic.approximation(), 其中, expr 是被逼近的符号表达式, PO 是电路参数的标称值, epsilon.max 是用户给定的最大误差限, lambda 是回收因子, 该伪代码所采用的是累计绝对误差。

```

symbolic_approximation(expr, PO, epsilon_max, lambda) {
  var accumulated_err, maximum_err, term, maximum_mag, eff_mean_val, C0, deleted_tab;
  accumulated_err ← 0;
  maximum_err ← 0;
  C0 ← 0;
  maximum_mag ← epsilon_max * 符号表达式 expr 在 PO 处的最大幅度;
  FOR (expr 中的每一项 term) {
    C0 ← C0 + 1;
    maximum_err ← maximum_err + epsilon_max * term 在 PO 处的最大幅度;
  }
  eff_mean_val = maximum_err / lambda / C0;
  FOR (expr 中的每一项 term) { /* 局部逼近策略 */
    IF (term 在 PO 处的幅度 小于或等于 eff_mean_val) {
      从 expr 中删除 term, 并将 term 插入到 deleted_tab 中;
      accumulated_err ← accumulated_err + term 在 PO 处的幅度;
    }
  }
  将 expr 中的各项按其 PO 处的幅度从小到大排列起来;
  WHILE (accumulated_err 小于 epsilon_max) {
    term ← expr 的第 1 项;
    accumulated_err ← accumulated_err + term 在 PO 处的幅度;
    IF (accumulated_err 小于或等于 epsilon_max 且
        term 在 PO 处的幅度 小于 maximum_mag) {
      从 expr 中删除 term, 并将 term 插入到 deleted_tab 中;
    } else {
      将 deleted_tab 中幅度 大于或等于 (term 在 PO 处的幅度 / lambda) 的所有项放回 expr;
      RETURN expr;
    }
  }
}

```

图 3.15 混合符号逼近法 symbolic approximation() 的伪代码

3.6 小 结

本章首先介绍了符号分析法的应用背景及符号模拟器的典型流程; 然后讨论了建立符号电路方程的表矩阵法、改进节点法和 CMNA 法, CMNA 法是最适合于符号分析的方法, 本章详细介绍了 CMNA 法的各个环节, 以及用 CMNA 法建立 s 域和 z 域符号电路方程的特点, 并介绍了如何处理器件失配的方法; 紧接着, 本章用较多的篇幅介绍了符号线性方程组求解的关键环节——求行列式的符号表达式, 所介绍的具有余子式记忆和动态行列选择的递归展开法不仅计算量小, 而且所产生的中间项很少, 所获得的符号表达式具有式(3.1)的形式, 是若干乘积项的和; 最后, 本章还进一步讨论了符号逼近法, 通过讨论逼近误差与逼近轨迹的对应关系、引入全局逼近策略和局部逼近策略, 最后形成了算法 symbolic_approximation(), 该算法采用累计绝对误差和混合逼近策略, 能获得满足给定逼近误差限要求的复杂度最低的符号表达式, 并能保留足够多的信息。本章介绍的符号分析法非常适合于为模拟集成电路设计自动化系统提供电路性能的解析表达式。

第四章 模拟集成电路的结构级综合

结构级综合是将给定的行为描述转化成用功能块实现的电路结构,它是模拟集成电路综合领域最困难、最不成熟的课题,这主要是因为模拟集成电路的种类繁多、结构千变万化。本章将讨论结构级综合的两个核心问题:模拟集成电路的行为描述(抽象的性能描述)及相应的结构级综合算法^[125,126]。至于如何将功能块转化为具体电路则是单元级综合^[127,128]所面临的课题,我们将在下一章讨论。

4.1 模拟集成电路的行为描述

模拟集成电路的行为描述和综合算法是模拟集成电路结构级综合中既相互影响又密切联系的两个方面。对于大多数的多输入、多输出模拟集成电路系统,一般采用状态方程组来描述其行为^[12,125,129-131]。假设某模拟集成电路的动态行为为:

$$\begin{cases} Y = f(X, Q, \dot{Q}) \\ \dot{Q} = g(X, Q) \end{cases} \quad (4.1)$$

其中, Y 是由输出端口构成的向量, X 是由输入端口构成的向量, Q 是状态向量, 函数 f 和 g 定义了该模拟集成电路的行为。由于 $\dot{Q}$ 经一组积分器处理后将输出 Q , 因此, 图 4.1 所示的框图就可以实现式(4.1)描述的行为。由此可见, 与这种行为描述相对应的模拟集成电路结构级综合算法的核心任务是用模拟集成电路功能块来实现函数 f 和 g 所要求的性能。

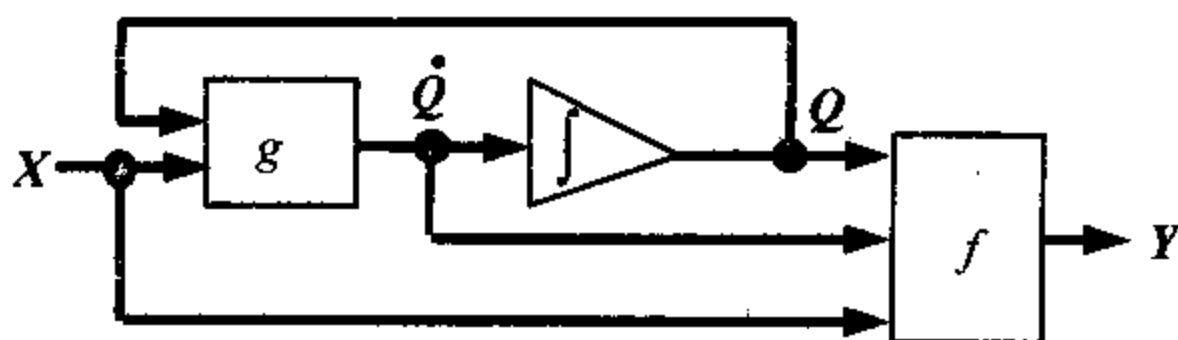


图 4.1 行为描述(4.1)的抽象功能块实现

对于单输入、单输出系统, 还常用下述 s 域的传输函数 $H(s)$ 来描述其行为:

$$H(s) = \frac{\sum_{i=0}^m b_i s^{m-i}}{\sum_{j=0}^n a_j s^{n-j}}, \quad m, n \in \mathbf{N}, \quad a_j, b_i \in \mathbf{R}, \quad a_0 = 1 \quad (4.2)$$

那些用频域通带、阻带特性给出的行为描述可以很容易地转化成式(4.2)的形式^[132], 当然, 也可以将式(4.2)转化为状态方程组(4.1)的形式。

在另外一些场合中, 将模拟集成电路的行为描述为(4.1)或(4.2)的形式是非常困难的, 但是, 用解析函数或表格函数来描述却非常方便。我们用 x 表示输入信号, y 表示输出信号, 则其行为可表示为:

$$y = f(x), \quad x \in [a, b] \quad (4.3)$$

IEEE 的 VHDL 1076.1 “标准”给出了一整套语法规则以描述模拟集成电路和混合信号系统的行为^[12]，虽然该标准没有受到工业界的欢迎，但 Mast^[133]和 Verilog-A^[134]却极有可能成为新的标准或事实上的工业标准。这些模拟集成电路的硬件描述语言为电路设计者描述电路行为提供了有力的工具，它们所描述的行为都可以直接或间接地转换为(4.1)、(4.2)、(4.3)或 z 域的传输函数。考虑到基于 z 域行为描述的综合方法在本质上与 s 域没有区别，因此，在后续几节中，我们将分别针对(4.1)、(4.2)和(4.3)这三种行为描述，详细讨论模拟集成电路的结构级综合方法。

4.2 基于状态方程组的功能块构造法

如果模拟集成电路的行为是以状态方程组(4.1)的形式给出的，则结构级综合算法的核心是如何分解函数 f 和 g 。为简便起见，我们假设 f 和 g 为“简单”函数，即它们满足下述条件^[125]：

- (1) 不含高阶微分；
- (2) 不含超越函数；
- (3) 仅包含加法、减法、乘法、微分和积分等五种运算。

对于不满足上述条件的“复杂”函数，可以通过引入新的状态变量或作适当的变换将它们化简为“简单”函数。例如，下述状态方程虽然包含高阶微分项和超越函数 $\sin x$ ，

$$\ddot{x} = -x + \dot{x}(1 - x^2)\sin x \quad (4.4)$$

但是，只要引入状态变量 y 和 z ，并令 $y = \dot{x}$ ， $z = \sin x$ ，就可以将(4.4)转化为下述仅含“简单”函数的状态方程组：

$$\begin{cases} \dot{y} = x \\ \dot{z} = yz - yzx^2 - x \end{cases} \quad (4.5)$$

实现 $y = \dot{x}$ 所要求的功能是很容易的，至于如何实现 $z = \sin x$ 所要求的行为，将在第 4.4 节中讨论。当 f 和 g 中包含除去加法、减法、乘法、微分和积分等之外的运算时，我们只要将与之相关的项看作超越函数，问题就迎刃而解了。

下面将以式(4.5)为例，详细介绍基于状态方程组的功能块构造法。

对于“简单”函数而言，我们先作语法分析，然后就可以用加法器、乘法器和积分器来实现。利用 YACC 这一编译器的生成工具^[135,136]可以很容易地实现“简单”函数的结构级综合。但是，由于不易实现中间代码优化，因而结构级综合结果中会有一些冗余模块。例如：对行为描述(4.5)的综合结果如图 4.2(a)所示，其中用了两个乘法器 MUL1 和 MUL2，有一个是冗余的。我们称这种包含冗余模块的语法分析结果为初始综合结果，简称初始结果。

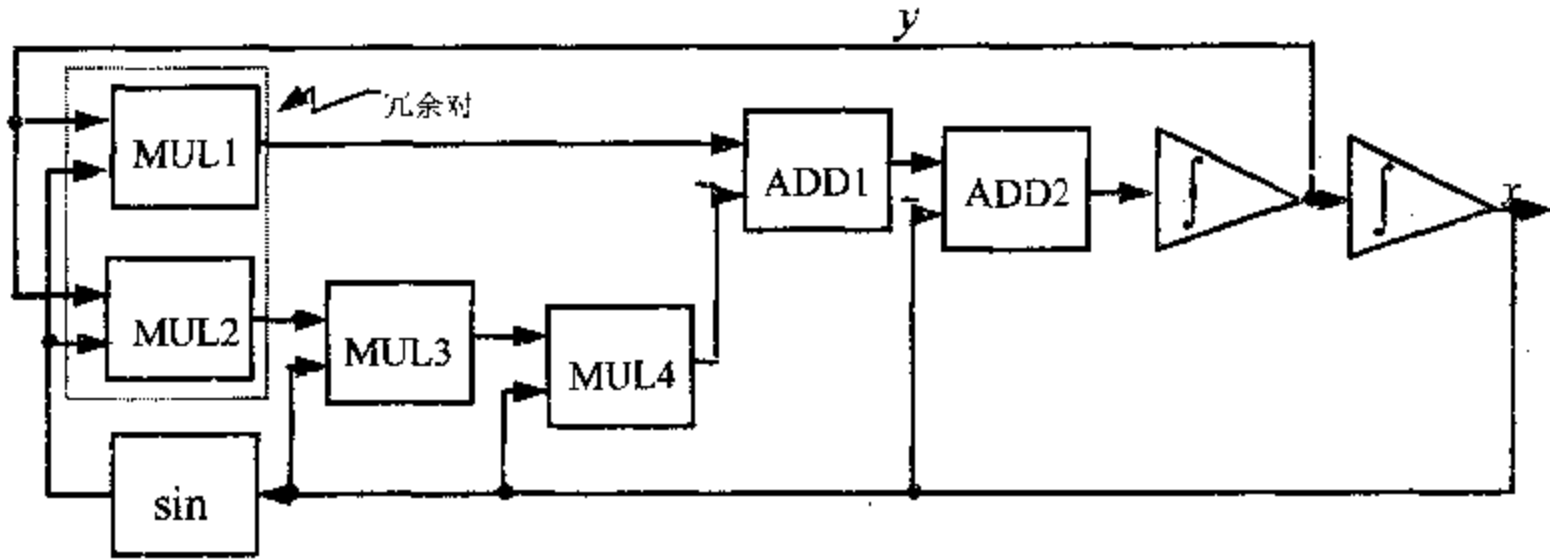
为了删除初始结果中的冗余模块，需要对初始结果作进一步处理。很显然，两个模块中有冗余块的充要条件为：

- (1) 两模块的功能相同；
- (2) 两模块的输入端完全等效。

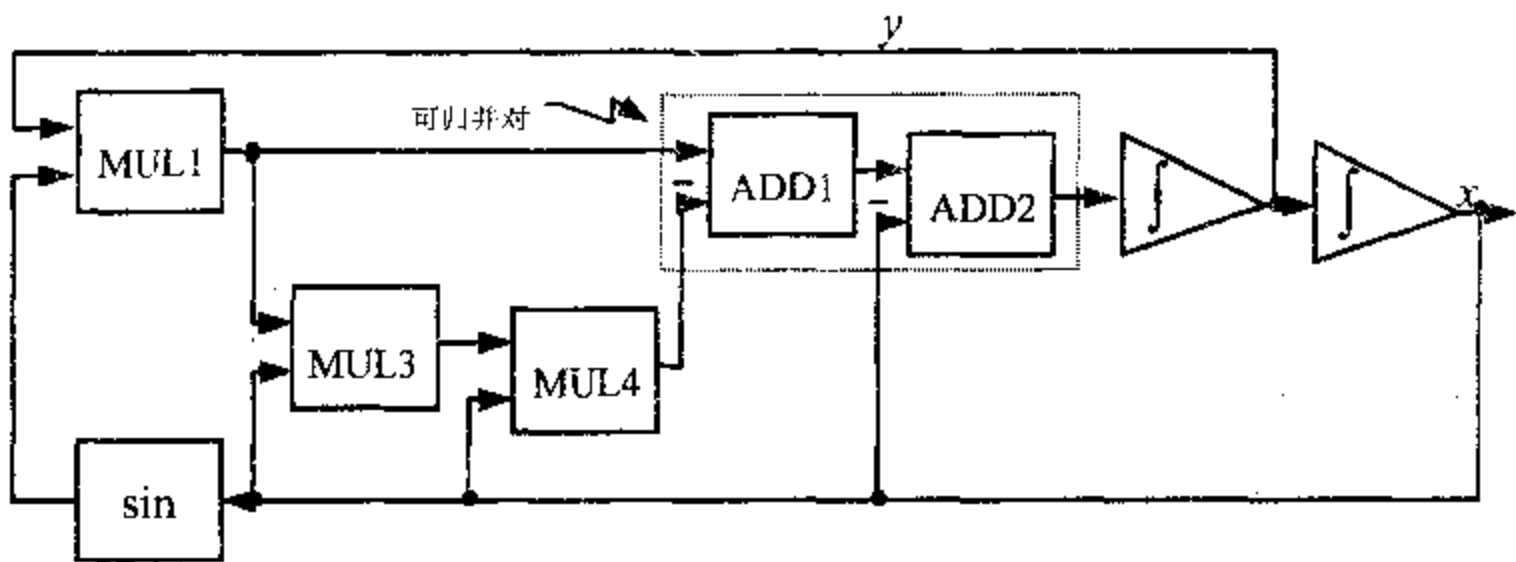
我们称有冗余的两个模块为冗余对，为叙述方便，我们将冗余对记为： $\times (A(NA)@B(NB))$ ，其中， A 和 B 分别是冗余对中两个模块的名称， NA ， NB 分别为模

块 A 和 B 的输出端的节点号。如果一组模块中两两为冗余对，则称之为冗余组。由于初始综合结果中每个模块最多含两个输入端，因此，判定有无冗余对或冗余组的工作非常容易。

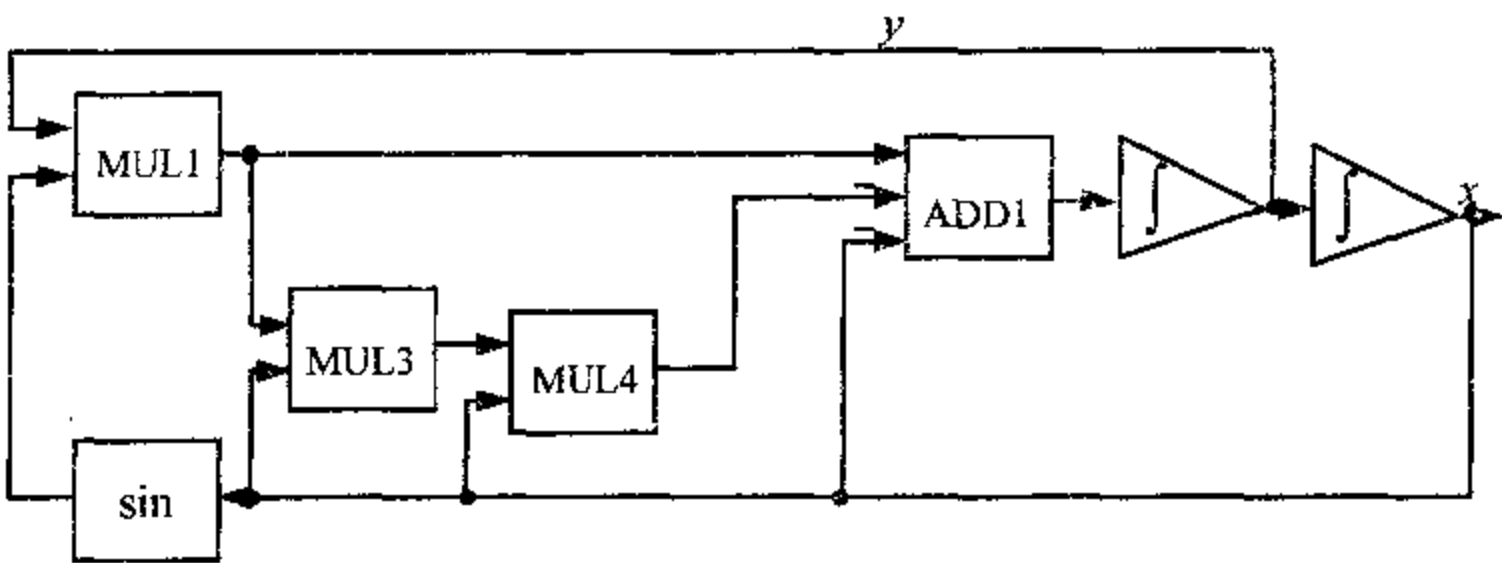
找出冗余对后，就可以根据初始结果中信号的模式(电流或电压)来进行去冗处理。对于电压模式(参见图 4.2(a))，设初始结果包含※(A(NA)@B(NB))，则只要简单地



(a) 行为描述(4.5)的初始综合结果



(b) 对初始综合结果(a)作去冗处理后的结果



(c) 对去冗处理后的结构(b)作加法器归并后的结果

图 4.2 语法分析、去冗处理、加法器归并三步式模拟集成电路结构级综合流程

删除模块 B, 并将初始结果中的节点 NB 全部改为 NA 即可, 图 4.2(b) 是去冗处理后的结果。对于电流模式, 在删除冗余模块时需要增加一个电流镜, 将保留模块的输出电流复制一份, 然后再用它去驱动后面的模块。对冗余组的去冗处理也是类似的, 就不再赘述了。

在去冗处理后, 还要作加法器归并, 即将两个二输入加法器归并为一个三输入加法器。加法器 A 可归并入加法器 B 的充要条件为:

A 的输出端与、且只与 B 的一个输入端相联接。

我们将加法器 A 可归并入加法器 B 的情形记为 $U(A(N1, N2, N) \rightarrow B(N, N3, N4))$, 其中, N 和 N4 分别是 A 和 B 的输出端, N1 和 N2 是 A 的输入端, N 和 N3 是 B 的输入端。如果 $U(A(N1, N2, N) \rightarrow B(N, N3, N4))$ 或 $U(B(N1, N2, N) \rightarrow A(N, N3, N4))$, 则称 A 和 B 为可归并对。图 4.2(b) 中的 ADD1 和 ADD2 就是可归并对。加法器归并的关键是识别可归并对, 可以用有向图的处理方法完成这项工作。一旦识别出 $U(A(N1, N2, N) \rightarrow B(N, N3, N4))$, 则只需在经去冗处理后的结构中增加一个三输入加法器, 其输入端分别为 N1, N2 和 N3, 输出端为 N4, 然后再删除加法器 A 和 B 即可。对图 4.2(b) 作加法器归并后的结果如图 4.2(c) 所示。实现两个多输入端加法器归并的过程与此类似, 就不再重复了。

顺便指出: ①结构级综合过程中不能作乘法器归并, 因为多输入乘法器不易实现; ②若乘法器的一个输入端为常数, 则应将它变成放大器。

4.3 给定 s 域传输函数 $H(s)$ 的模拟滤波器综合

滤波器的特性通常是由频域或 s 域的传输函数来描述的, 用式(4.2)所示的 s 域的传输函数 $H(s)$ 来描述其行为既方便又实用。不失一般性, 我们仅讨论传输函数为下述形式的模拟滤波器综合问题:

$$H(s) = \frac{\sum_{i=0}^m b_i s^{m-i}}{\sum_{j=0}^n a_j s^{n-j}}, \quad m, n \in \mathbf{N}, \quad m < n, \quad a_j, b_i \in \mathbf{R}, \quad a_0 = 1 \quad (4.6)$$

$m \geq n$ 时的情况与(4.6)之间并无本质差异。我们虽然可以先将(4.6)转化成(4.1)的形式, 然后再通过语法分析、去冗处理和加法器合并等一系列步骤获得一个传输函数为 $H(s)$ 的电路结构, 但由于(4.6)的形式非常规范, 而且是单输入单输出的情形, 因此, 还可以采用一些更简洁的方法。

利用梅森增益公式^[118], 我们很容易将式(4.6)表示成图 4.3 所示的信号流图, 其中, X_{in} 是滤波器的输入, X_{out} 是滤波器的输出。由于 $1/s$ 的功能可以用积分器实现, a_j, b_i 可以由跨导放大器实现, 而各节点处的加法运算又可以由电流镜和运放实现, 因此, 很容易将图 4.3 “翻译”成由积分器、跨导放大器、电流镜和运放等模块所构成的电路^[125], 然后就可以根据需要决定采用 OTA-C^[137]、MOSFET-C^[138]、开关电容电路^[139]和开关电流电路^[140]等之中的哪一种电路来实现。

上述思想对数字滤波器设计而言是可行的, 但对模拟滤波器而言尚存在不少问题。因为, 要保证滤波器具有 $H(s)$ 的性能, 就必须在同一芯片中实现 $(m+n)$ 个跨导放大器, 并使每个跨导放大器的跨导值均非常准确, 这是极端困难的。考虑到对式(4.6)的分子、

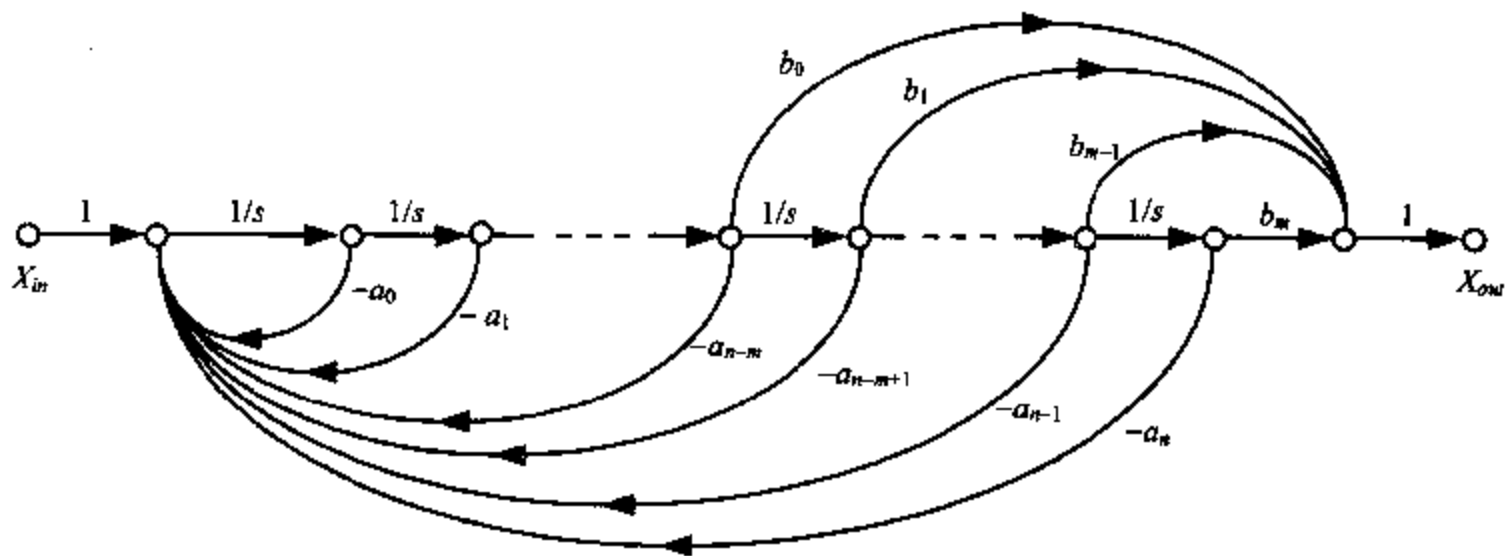


图 4.3 $H(s)$ 的信号流图表示

分母同时乘以任何非零常数均不会改变 $H(s)$ 的特性, 换言之, 我们只需保证 a_j, b_i 的相对比值精确即可。因此, 一种比较明智的做法是用跨导值为 1 的放大器和电流比为 a_j 的电流镜来实现图 4.3 中 a_j 所要求的性能^[126](参见图 4.4)。图 4.4 中电流镜的综合是非常简单的, 我们只要令 $(W/L)_3 = (W/L)_1$, 电流比 $(W/L)_2 / (W/L)_1 = a_j$ 就行了。这种技巧有助于提高跨导值的相对精度, 从而使结构级综合结果易于实现。

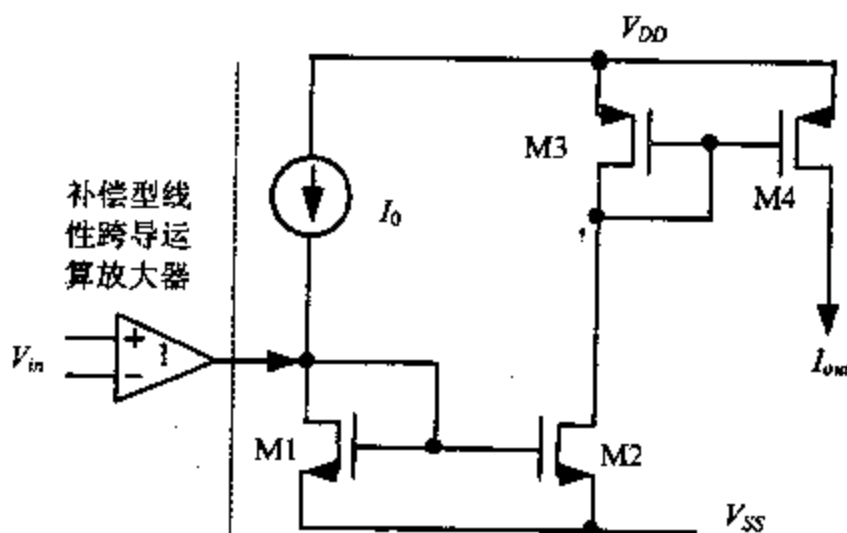


图 4.4 用电流镜提高跨导值的匹配精度

特别值得一提的是: 图 4.4 中的单位增益跨导放大器需要精心设计, 除需要减小其非线性失真之外, 还需要减小各种工艺涨落因数对其跨导值的影响, 可以借鉴文献[141—143]中的电路技巧。图 4.5 所示的电路结构可以较好地解决该问题, 该电路中 MOS 场效应管 M1 和 M4 的几何尺寸均为 W/L , M2 的几何尺寸为 $(1-\lambda)W/L$, M3 的几何尺寸为 $\lambda W/L$, λ 为纯小数。M3 的引入可以大大抑制输出电流中的二次谐波分量, 确保该跨导运算放大器的线性特性, 同时还有助于稳定其跨导值^[142]。

当然, 积分器的设计也十分考究, 可借鉴文献[144—146]中的技术以降低积分器的谐波失真、提高滤波器的频响特性, 为节省篇幅, 这方面的内容就不赘述了。

我们可以事先设计好若干跨导放大器和积分器, 并将它们存入库中, 因此, 模拟滤波器综合就简化为电流镜的设计和跨导放大器、积分器的选择, 综合结果仅包含积分器、跨导放大器、电流镜和加法器, 而且, 所有积分器是一样的, 所有跨导放大器也是相同的。

由此可见, 采用上述策略后, 模拟滤波器的综合结果很有规律, 其物理版图的综合

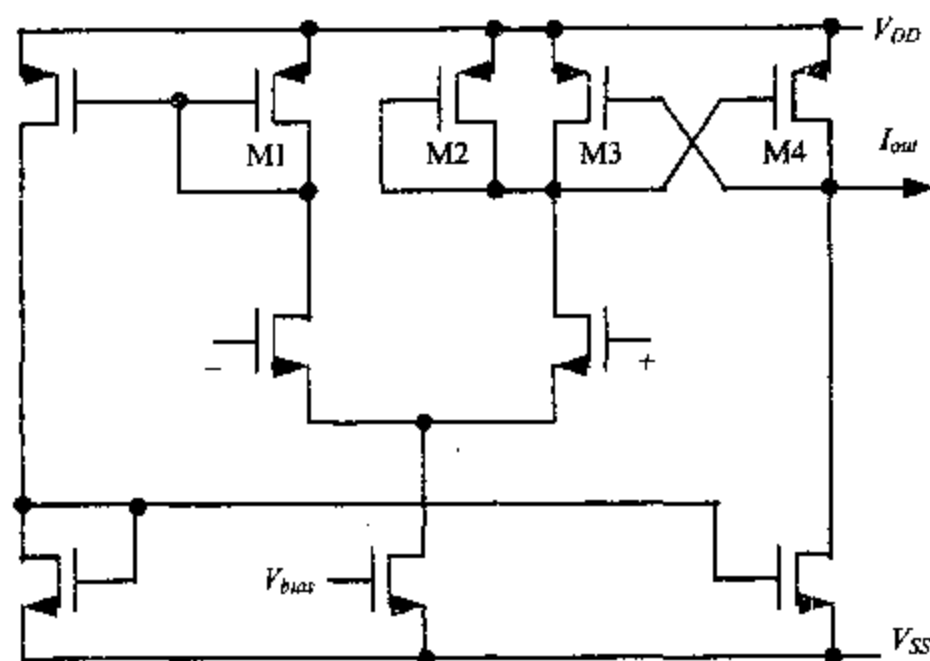


图 4.5 带补偿的线性跨导运算放大器

也变得非常容易。事实上，我们可以按下列三个步骤生成模拟滤波器的物理版图：

- (1) 产生积分器、跨导放大器的物理版图，并按 $H(s)$ 的具体形式产生加法器和所需电流镜的物理版图；
- (2) 按 $H(s)$ 的具体形式将各模块有机地联接起来；
- (3) 进行设计规则和电规则检查，压缩所获得的物理版图以提高模拟滤波器的性能。

其中，第一步是很容易实现的，因为我们可以事先设计好积分器和跨导放大器的物理版图，并将其作为标准单元存起来，至于加法器和电流镜的物理版图就更容易生成了；第二步可以采用目前已非常成熟的标准单元布局布线技术；第三步主要是为了进一步节省芯片面积和提高滤波器性能，除注意遵守设计规则外，还应特别注意不要将同一电流镜中的匹配器件分开。

很容易看出，传输函数(4.6)还可以分解为下述形式：

$$H(S) = \prod_i \frac{q_{2,i}S^2 + q_{1,i}S + q_{0,i}}{p_{2,i}S^2 + p_{1,i}S + p_{0,i}} \quad (4.7)$$

或

$$H(S) = \sum_i \frac{r_{1,i}S + r_{0,i}}{p_{2,i}S^2 + p_{1,i}S + p_{0,i}} \quad (4.8)$$

其中， $p_{2,i}^2 + p_{1,i}^2 \neq 0$ ， $q_{2,i}^2 + q_{1,i}^2 + q_{0,i}^2 \neq 0$ ， $r_{1,i}^2 + r_{0,i}^2 \neq 0$

若 $p_{2,i} = 0$ ，则 $q_{2,i} = r_{1,i} = 0$

从式(4.7)或(4.8)出发，我们可以用二阶有源滤波器构造出级联或并联的结构，在电路实现和物理版图综合方面同样需要利用前面介绍的各种技巧。至于 $H(s)$ 为超越函数的情形，可以用培德(Padé)逼近^[106]将它先表示成式(4.6)的形式，然后再按前述步骤进行。当然，我们也可以采用最佳一致逼近技术，用切比雪夫(Chebyshev)多项式表示 $H(s)$ ，这是数字滤波器设计中常用的手段。当用模拟电路实现时，这种方法需要高性能的四象限模拟乘法器，不容易实现，因此，不宜采用这种方法来处理模拟滤波器的综合问题。

4.4 基于最佳一致逼近的函数综合方法

有些模拟集成电路的性能很难用式(4.1)或(4.2)来描述,图4.6所定义的某电路的输入-输出特性曲线就是一例,但是,用式(4.3)来描述图4.6所定义的电路性能却非常方便。对于这类问题,综合的核心任务是构造一个功能为函数 $f(x)$ 的电路。我们在第4.2节中遇到的超越函数 $z=\sin x$ 就是式(4.3)的具体实例。

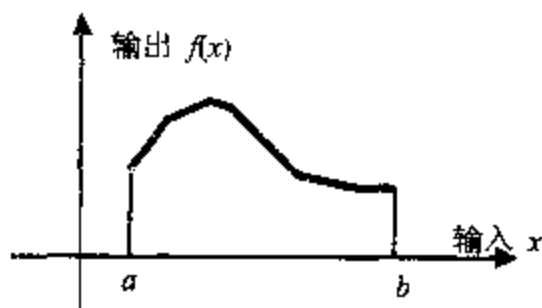


图 4.6 输入-输出转移特性曲线

函数综合的基本思路是将 $f(x)$ 分解为一组基函数的线性组合,关键是如何分解 $f(x)$,如何选择基函数,如何用模拟电路精确实现基函数所要求的功能,以及如何用模拟电路实现基函数的线性组合运算。

假设 $\{\psi_n(x)\}_{n=1}^{\infty}$ 是 $[a,b]$ 上的一组线性不相关函数,称为基函数,则, $\forall f(x), x \in [a,b]$ 均可以表示为:

$$f(x) \approx \sum_{n=1}^N c_n \psi_n(x), \quad c_n \in \mathbf{R} \quad (4.9)$$

其中, N 由期望的逼近误差决定,系数 c_n 是下述最佳一致逼近问题的解:

$$\min_C E(C) = \max_{x \in [a,b]} \left| f(x) - \sum_{n=1}^N c_n \psi_n(x) \right|, \quad C = [c_1, c_2, \dots, c_N]^T \quad (4.10)$$

上述基于最佳一致逼近的模拟集成电路结构级综合方法需要解决三个核心问题:基函数 $\{\psi_n(x)\}_{n=1}^{\infty}$ 的选择、最佳一致逼近问题(4.10)的求解和组合运算的实现。由于可以采用第二章所介绍的多分子模拟退火法来求解最佳一致逼近问题(4.10),因此,这里仅讨论基函数的选择及组合运算的电路实现技巧。

4.4.1 基函数的选择

用泰勒(Taylor)级数、傅立叶(Fourier)级数或沃泰拉(Volterra)级数等来逼近 $f(x)$ 是数学上广泛采用的方法,但由于用模拟集成电路来实现运算 x^n 或 $\sin(n\omega x)$ 非常困难,因此,将 $\{\psi_n(x)\}_{n=1}^{\infty}$ 选为泰勒级数、傅立叶级数或沃泰拉级数都不合适。从模拟集成电路综合的角度来看, $\psi_n(x)$ 应具备下述两个特征:

- (1) 能够用简单的单元实现,为便于作线性组合运算,各单元的输出最好是电流;
- (2) 能适应尽可能多的 $f(x)$ 。

为此,我们首先考察图4.7所示的电路,该单元由两对双极型差分对组成,它们的发射极面积之比为 A_n ,但偏置电流均为 I_0 ,偏置电压均为 $V_{B,n}$,输入电压均为 $n \cdot x$ 。容易证明,差分输出电流($\Delta I = I_+ - I_-$)为^[147]:

$$\Delta I = \alpha_F I_0 \left\{ \tanh \left(\frac{n \cdot x - V_{B,n} + \beta_n}{2V_T} \right) - \tanh \left(\frac{n \cdot x - V_{B,n} - \beta_n}{2V_T} \right) \right\} \equiv \alpha_F I_0 \psi_n(x) \quad (4.11)$$

其中, $V_T = kT/q$ 称为热电压, k 为 Boltzmann 常数, q 为电子电荷, T 为绝对温度; α_F

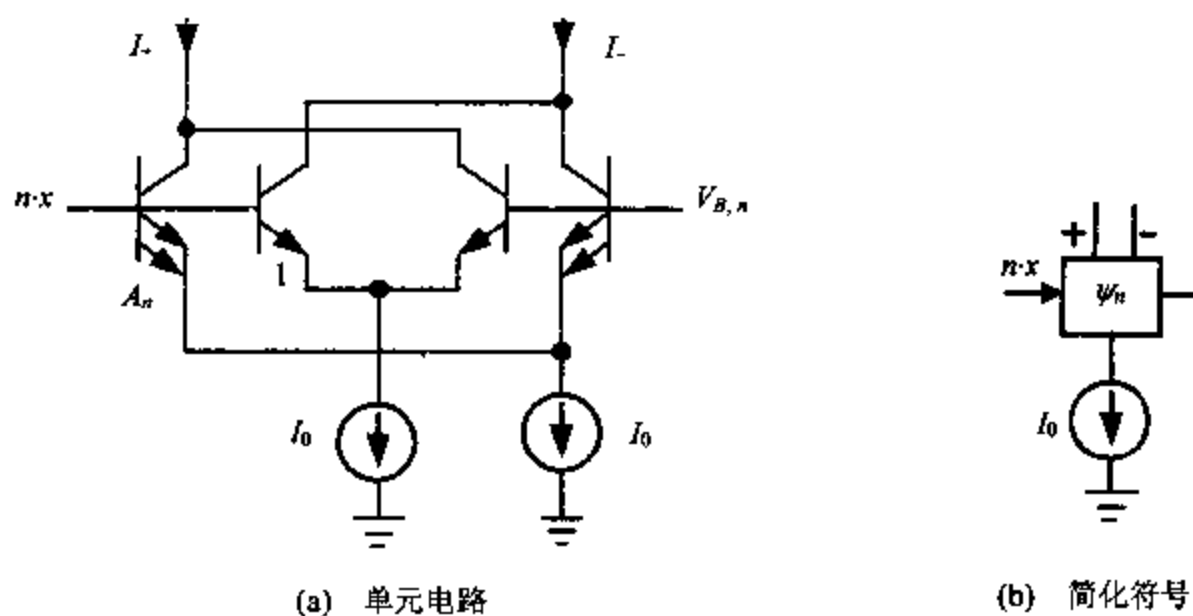


图 4.7 产生 $\psi_n(x)$ 的双极型电路单元及其简化符号

是共基极直流电流放大倍数；且 $\beta_n = V_T \ln A_n$ 。

我们将 $\Delta I / (\alpha_F I_0)$ 选作基函数 $\psi_n(x)$ ，与之对应的单元电路的输出是电流，因此，按这种方式选择的基函数特别适合于作线性组合运算。很容易证明， $\forall N > 1$ ， $\psi_1(x)$ ， $\psi_2(x)$ ， $\dots$ ， $\psi_N(x)$ 都是线性不相关的。因此， $\forall \epsilon > 0$ ， $\exists N, c_n, \beta_n, V_{B,n}$ ，使下式成立：

$$\max_{x \in [a, b]} \left| f(x) - \sum_{n=1}^N c_n \psi_n(x) \right| < \epsilon \quad (4.12)$$

因此，可以用图 4.7 所示电路作为产生基函数的功能块，文献[147]中还有另外几种单元电路，也可以用来产生基函数。

4.4.2 组合系数的确定与电路实现

在选定基函数 $\{\psi_n(x)\}_{n=1}^{\infty}$ 后，对于任何定义于 $[a, b]$ 上的函数 $f(x)$ ，我们总可以用第二章中介绍的多分子模拟退火法求出最佳一致逼近问题(4.10)的解，从而获得用基函数 $\{\psi_n(x)\}_{n=1}^{\infty}$ 逼近 $f(x)$ 的组合系数 c_n ；在获得组合系数 c_n 之后，实现式(4.9)的线性组合运算就迎刃而解了。图 4.8 是采用双极型电路来实现该组合运算的电路原理图，若是 MOS 电路，其结构与此雷同。在图 4.8 中，参数 $C_n = c_n / \alpha_F$ ，且联接规则如下：

- (1) 若 $c_n > 0$ ，则产生 ψ_n 的单元按图 4.8 联接；
- (2) 若 $c_n = 0$ ，则删去产生 ψ_n 的单元；
- (3) 若 $c_n < 0$ ，则互换图 4.8 中 ψ_n 单元的两个输出端的联接点。

容易验证，图 4.8 中的 I_{out} 和 V_{out} 在数值上均为 $\sum_{n=1}^N c_n \psi_n(x)$ ，用它来逼近 $f(x)$ 可以达到足够高的精度。很显然，当 $f(x)$ 为电流量纲时，则应去掉图 4.8 中的 M^* 模块。

从前面的分析不难看出，按式(4.11)选择基函数 $\{\psi_n(x)\}_{n=1}^{\infty}$ 和用图 4.8 所示的电路结构来实现线性组合运算有下述四方面的优点：

- (1) 能在比较宽的范围内很好地逼近函数 $f(x)$ ；
- (2) 能用简单的双极型集成电路单元(图 4.7)实现 $\psi_n(x)$ 所要求的功能，如果采用 MOS 工艺，实现 $\psi_n(x)$ 功能的电路也非常简单，其拓扑结构与图 4.7 雷同^[51, 148, 149]；
- (3) β_n (即 A_n) 和 c_n 一样都可以作为设计变量，这有助于用较少的项去逼近很复杂的函数 $f(x)$ ，从而减少芯片面积；

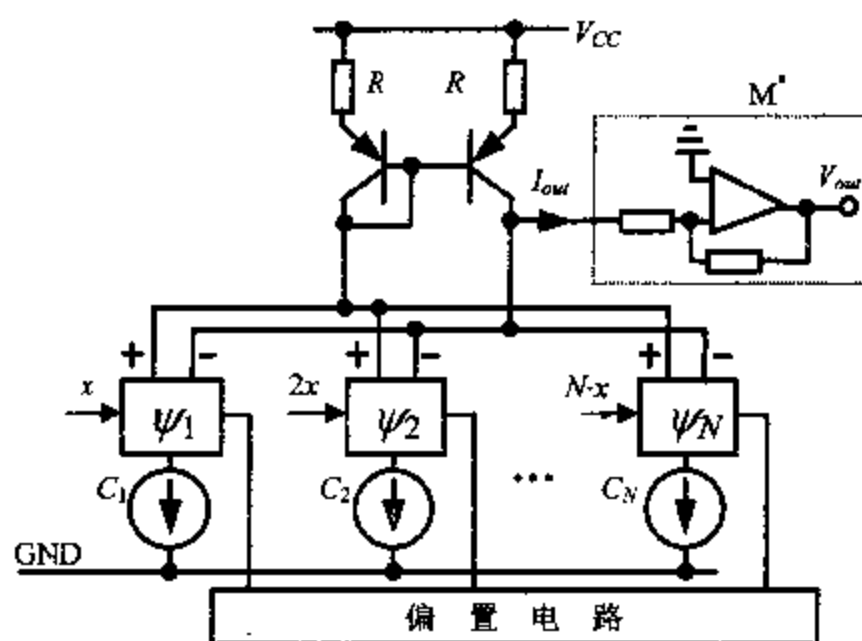


图 4.8 实现线性组合运算的双极型集成电路结构

(4) 直接调节产生 $\psi_n(x)$ 的功能块的偏置电流就可以调节式(4.9)中的组合系数 c_n ，而不需额外的电路，因而有助于提高电路性能和减小芯片面积。

在模拟集成电路的函数综合方面，一种最直接的方法就是根据 $f(x)$ 的具体形式构造一个函数表格，然后通过查表的方式实现映射 $x \rightarrow f(x)$ ^[150]，我们称该方法为直接映射法。直接映射法具有通用性，但需要用 A/D 将输入的模拟信号转变成数字信号，还需要 D/A 将从表格中查到的数字信号转换成模拟信号输出，其缺陷是既降低了信号处理的速度，又增加了电路的复杂性和生产成本。Fattarusio 和 Meyer 则利用 MOS 场效应晶体管在饱和区的特性，用分段线性逼近法进行函数综合^[151]；后来，Sanches-Sinencio 等人采用跨导运算放大器来综合非线性函数^[152]，其实质仍属于分段线性逼近法。分段线性逼近法克服了直接映射法需要 A/D、D/A 和信号处理速度慢的缺陷，虽然能胜任线性函数、有理分式等的综合，在某些场合甚至是非常有效的，但该方法的通用性差。与这些方法相比，基于最佳一致逼近的函数综合法是一种通用的方法，不会降低信号的处理速度，电路结构和其中的功能模块都很简单，而且易于实现。不过，为使该方法有很高的效率，应当有多组基函数 $\{\psi_i(x)\}_{i=1}^{\infty}$ 以保证用尽量少的级数 N 综合出用户所需的函数 $f(x)$ ，这对于减小芯片面积、提高集成电路的合格率是至关重要的。

4.5 综合实例

例 4.1 综合下述振荡器：

$$\dot{x} = -x + 0.5x(1 - x^2) \quad (4.13)$$

我们令 $y = \dot{x}$ ，则： $\dot{y} = 0.5y - 0.5yx^2 - x$ ，经过语法分析、去冗处理和加法器归并等处理后所综合出的功能块结构如图 4.9 所示。很容易验证综合结果是正确的。

例 4.2 N 阶切比雪夫(Chebyshev)滤波器的频域特性为：

$$H(j\omega) * H(j\omega) = \frac{1}{1 + \epsilon^2 V_N^2(\omega/\omega_c)} \quad (4.14)$$

其中， $V_N(x) = \cos(N \cos^{-1} x)$ 。

作为模拟滤波器综合实例，我们要求切比雪夫滤波器具有下述特性：

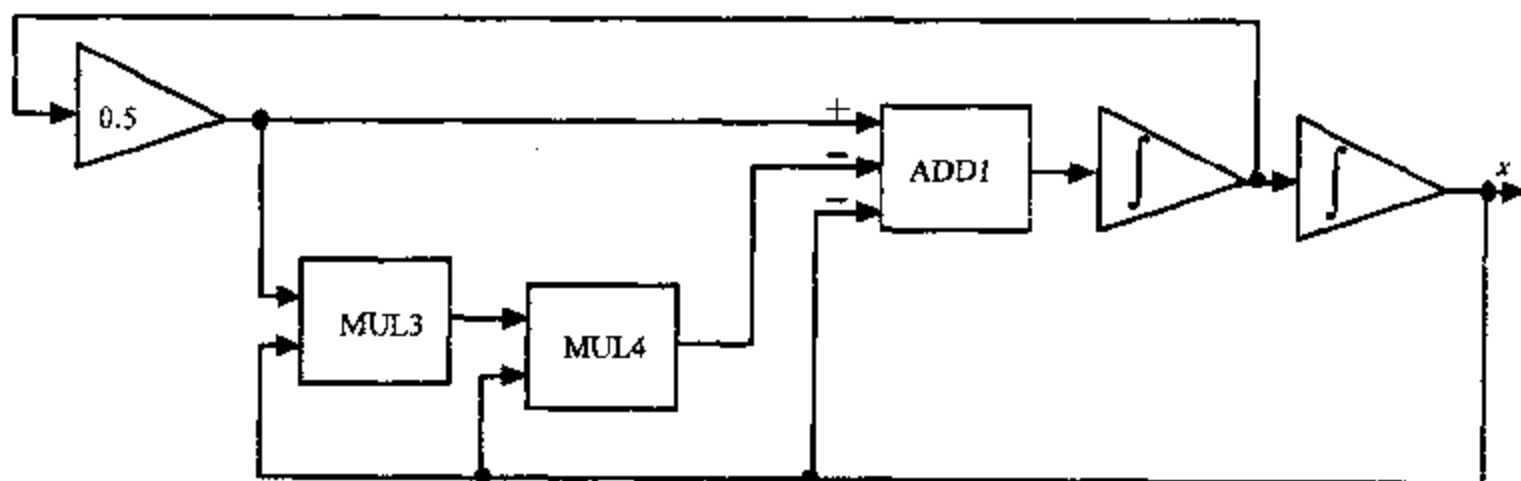


图 4.9 振荡器的功能块描述

$$20\log_{10}|H(j0.2\pi)| \geq -1, \quad 20\log_{10}|H(j0.3\pi)| \leq -15 \quad (4.15)$$

我们首先将式(4.14)和(4.15)转化成 s 域的传输函数^[132], 则

$$H(s) = \frac{0.03862}{s^4 + 0.5986s^3 + 0.5739s^2 + 0.1842s + 0.0429} \quad (4.16)$$

比较式(4.6)和(4.16), 便可以综合出该切比雪夫滤波器的电路结构(参见图 4.10)。为验证综合结果是否正确, 我们先用 SPICE 中的行为受控源替代图 4.10 中的积分器、跨导放大器和电压放大器, 然后进行交流分析, 图 4.11 是用 SPICE 所获得的幅频特性和相频特性曲线。

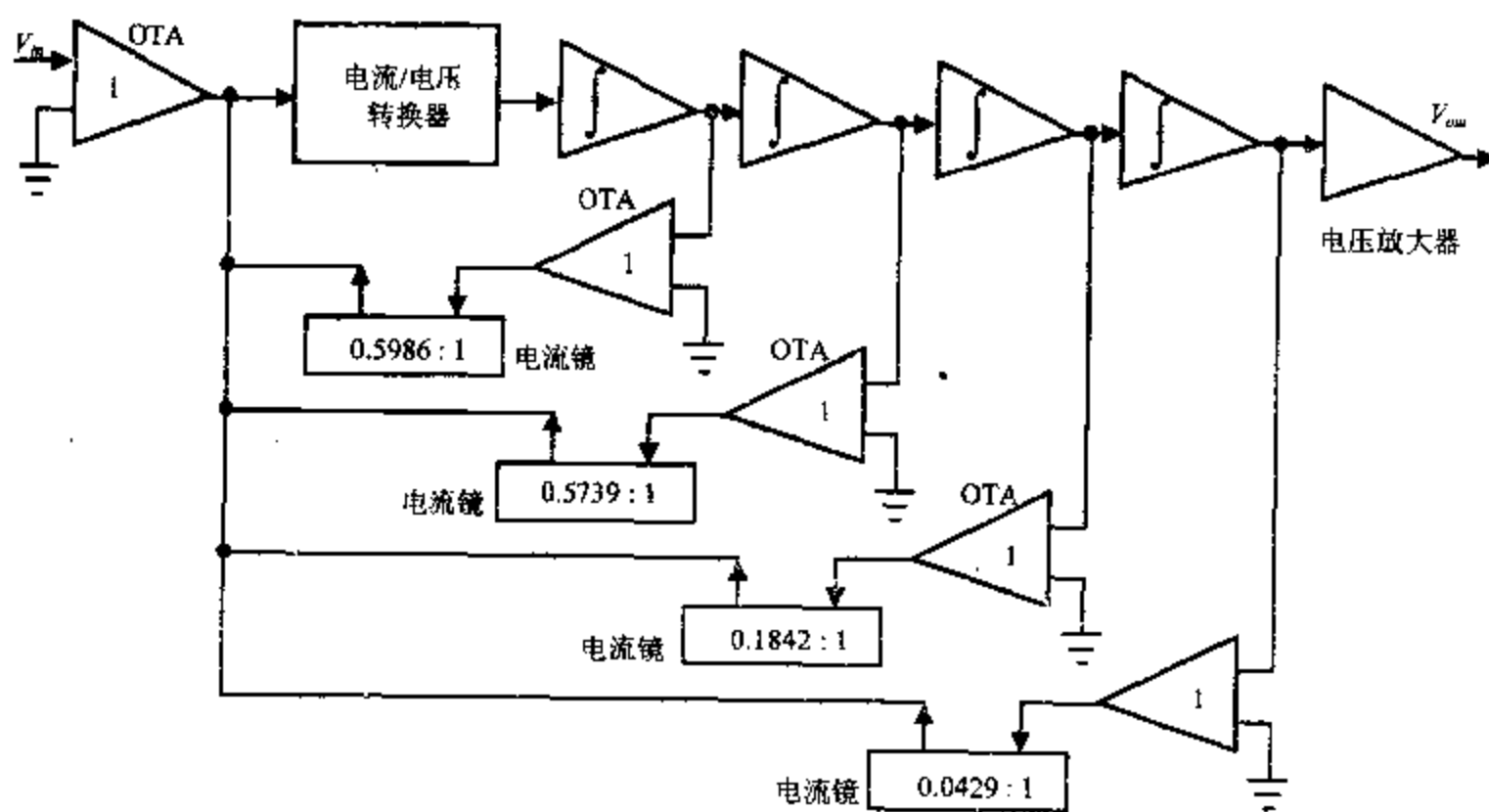


图 4.10 切比雪夫滤波器的功能块结构

例 4.3 1985 年, J. W. Fanttaruso 和 R. G. Meyer 曾手工设计 MOS 三角/正弦波形变换器^[153]; 后来, O. Ishizuka 等人又获得了更好的正弦波发生器^[149]。我们用前面介绍的最佳一致逼近法来综合一个三角/正弦波形变换器。

采用本文的方法时, 实质是构造一个功能为 $y = \sin(5\pi x)$ 的单元电路, 其中, x 代表输入信号, y 代表输出信号, 而且都以伏特(V)为单位。取 $N=5$ 就可以使逼近误差小于 5%。为进一步考察所综合出的电路性能, 我们用 SPICE 对该电路进行了模拟(负载电阻为 $2k\Omega$), 图 4.12 是 SPICE 模拟结果。从图 4.12 可以看出, 综合结果是令人满意的, 所

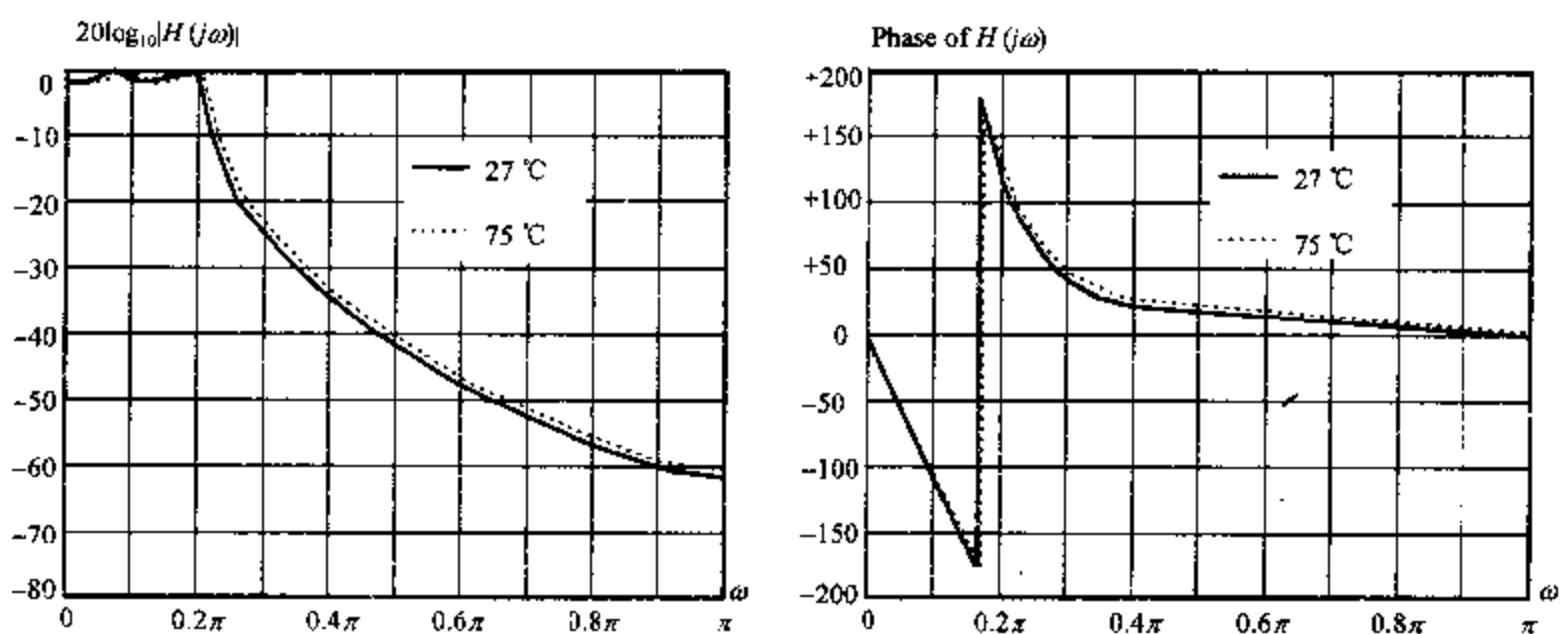


图 4.11 切比雪夫滤波器的 SPICE 模拟结果

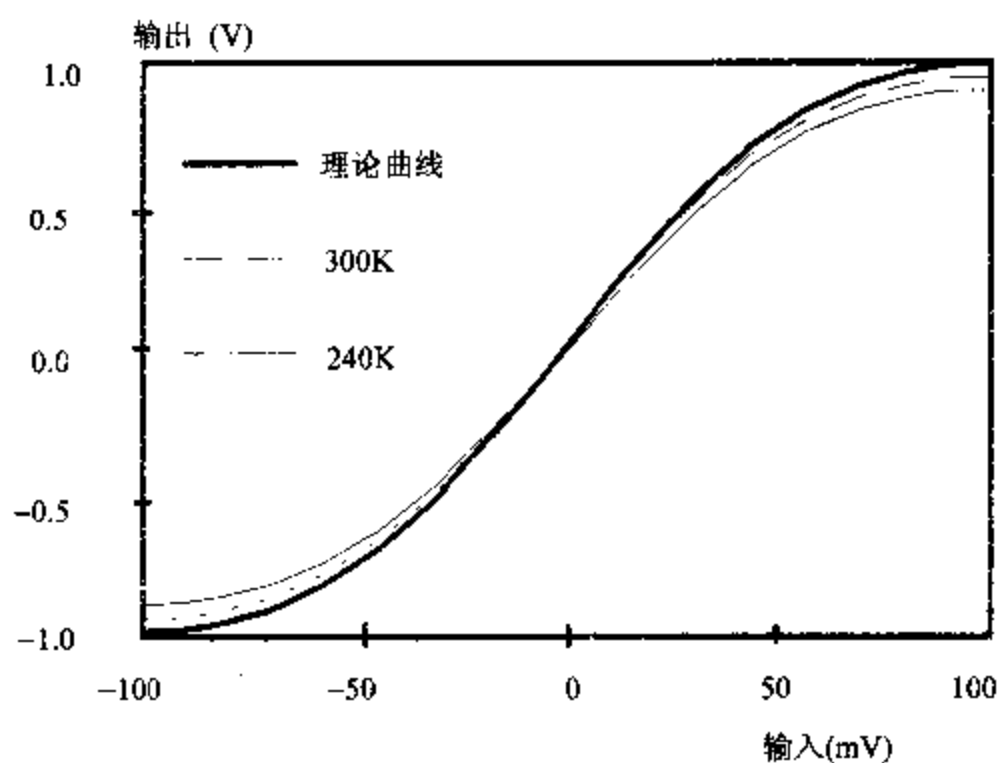


图 4.12 所综合的三角/正弦波形变换器在不同温度下的特性

综合出的电路具有较好的温度特性。

本章所介绍的最佳一致逼近法的通用性强，但所需器件数比 J. W. Fanttaruso 和 R. G. Meyer 的方案多，当然芯片面积也会更大一些。需要特别提醒读者注意的是：本章介绍的基函数和电路实现方案没有专门考虑温度特性的影响，当工作温度变化量增加时逼近误差也会增大，只有采用温度稳定性极好的单元电路来实现基函数所要求的特性才能解决该问题。

例 4.4 我们再讨论一个电压变换器的实例，其传输特性如图 4.13 所示。类似于这样的传输特性，用手工方式几乎是无法设计的，但用本章介绍的最佳一致逼近法，则可以很容易地获得图 4.14 所示的电路结构(本例采用的是双极工艺)。为便于对比，我们将该电路的传输特性也描绘在图 4.13 中。从图 4.13 中不难看出，该综合结果是令人满意的。

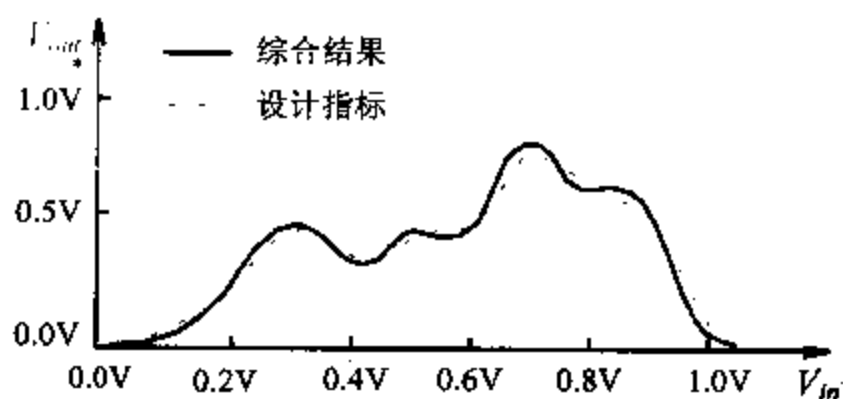


图 4.13 电压变换器的设计指标与综合结果对比

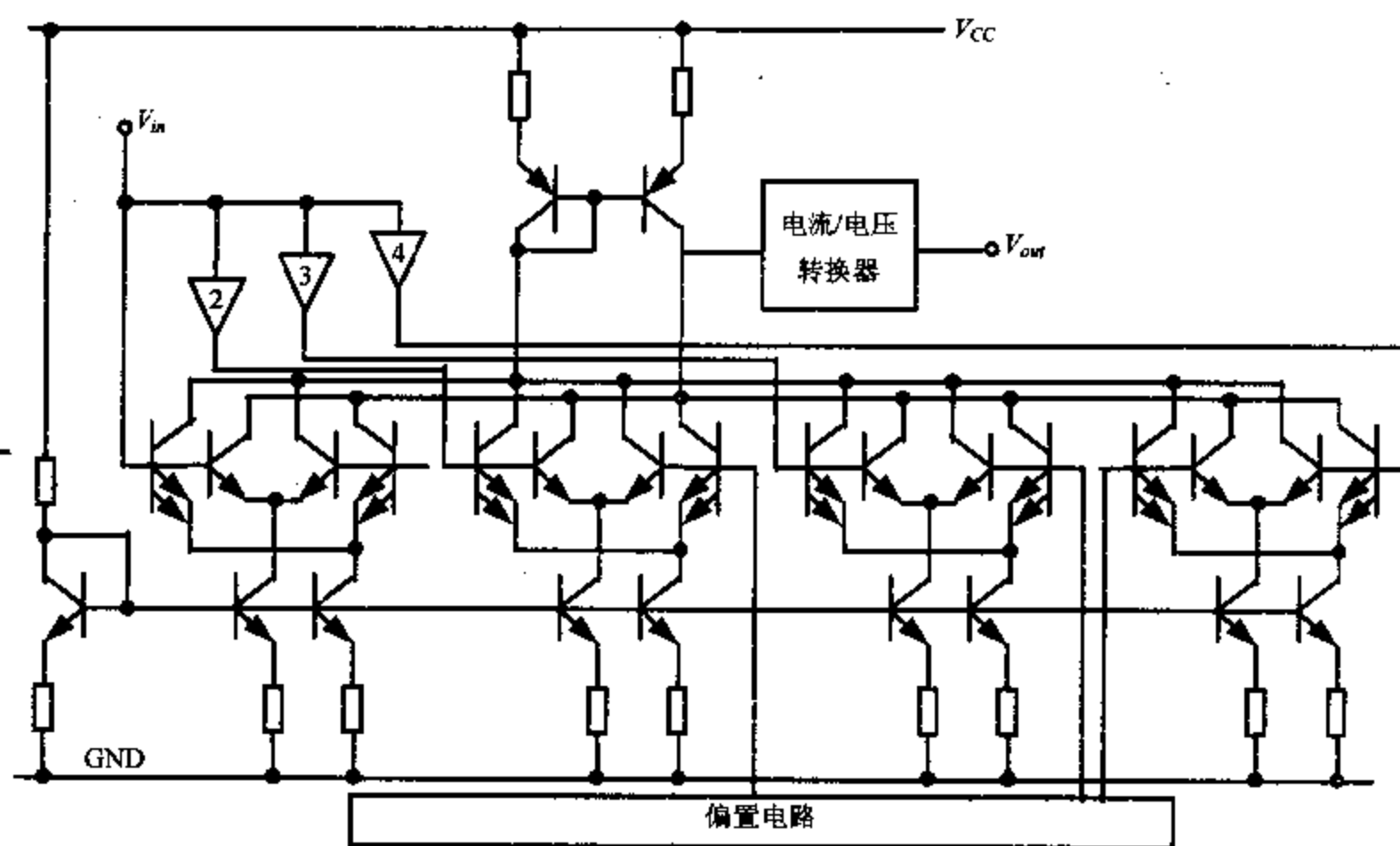


图 4.14 最佳一致逼近法所综合出的电压变换器的电路结构

4.6 小 结

本章首先介绍了关于模拟集成电路性能的三种抽象描述——时域状态方程组、 s 域的传输函数和用解析函数表示的输入输出特性；针对这三种抽象描述的不同特点，本章分别讨论了三种适于模拟集成电路结构级综合的方法：

(1) 对于时域状态方程组，我们采用基于语法分析的功能块构造法，并采用了去冗处理和归并处理等手段来优化在语法分析阶段所获得的结构级综合结果；本章引入了“简单函数”的概念，它不仅使语法分析变得非常容易实现，而且使结构级综合具有更明晰的层次，这种结构化、层次化的方法既是模拟集成电路设计的发展方向，也是模拟集成电路综合系统的发展方向。

(2) 对于 s 域传输函数，我们借鉴了数字滤波器综合的技术，但是，在模拟集成电路中，直接设计、实现图 4.3 所要求的 $(m+n)$ 个跨导放大器是费时费力、且很难实现的。

为此,本章给出了一种非常巧妙的电路结构(参见图 4.4),该方案不仅使繁冗的跨导放大器的设计简化为电流镜的设计,而且提高了各跨导放大器之间的匹配精度,从而使所综合出的滤波器更容易实现。另外,由于采用本章的方法所综合出的电路结构十分规范,因此,后续的物理版图综合可以采用目前已相当成熟的标准单元的布局布线算法。

(3) 对于以解析函数给出的输入输出特性,本章介绍了一种基于最佳一致逼近的函数综合法,并详细讨论了基函数的选择与电路实现技术;而最佳一致逼近问题的求解方法则可以采用第二章中介绍的多分子模拟退火法。综合实例表明该方法是切实可行的。

本章还给出了若干结构级综合的实例,这些实例不仅表明本章的所有算法是正确的,而且还证明了第二章所提出的多分子模拟退火法具有较好的全局收敛性,适合于求解模拟集成电路结构级综合中的优化问题。

结构级综合结果中仅包含积分器、乘法器、跨导放大器和加法器等少数几种常用的模拟集成电路单元,因此,本章所介绍的方法能减轻单元级综合的负担。

特别值得一提的是:随着微电子技术的发展,模拟集成电路的硬件描述语言将成为描述模拟系统、数模混合系统和混合信号集成系统的主要工具。虽然模拟集成电路的硬件描述语言^[12,133,134]采用了十分灵活的语法和非常丰富的内容,但不论如何,关于模拟电路行为的描述总可以转化为本节所采用的三种基本形式或人们惯用的单元描述模式(本书将在下一章讨论这种情况)。当然,由于模拟电路今后的发展方向是朝高性能、特别是混合信号集成系统的方向发展,因此,模拟集成电路的硬件描述语言中有许多关于描述光电混合系统、热电混合系统以及机电混合系统的语法规则,这些是本章的方法所不能直接解决的,因为本章的行为描述没有涉及光、热、速度等物理量。不过,本章关于结构级综合的思想和方法仍然能为解决这类问题提供借鉴。

第五章 模拟集成电路的单元级综合

第四章介绍了关于模拟集成电路行为的三种抽象描述：状态方程组、 s 域的传输函数 $H(s)$ 和解析函数 $f(x)$ ，以及相应的结构级综合方法。由于结构级综合结果是用各种模拟集成电路单元所构成的，因此，只有将这些单元转换为具体的电路形式（包含器件尺寸和约束条件），才能精确、全面地预估模拟集成电路的性能，才能综合其物理版图。本章将详细介绍如何把单元模块转化为具体电路形式的模拟集成电路单元级综合方法。

5.1 拓扑选择与器件尺寸优化

模拟集成电路单元具有可层次分解的特征，即：一个模拟集成电路单元可以分解为若干子单元（例如：输入级、输出级、恒流源等），而每个子单元又可以分解为一些基本的器件（例如：MOSFET、BJT、二极管、电阻、电容等）。众所周知，每个子单元一般都有许多种实现方案，因此，对于同一类模拟集成电路而言，其拓扑结构的种类非常繁多。这是模拟集成电路综合很难进行的根本原因。图 5.1 是某模拟集成电路单元 C 的抽象结构及数据路径示意图，其中，模块 $B_1 \sim B_6$ 为子单元，单元 C 必须包含模块 B_6 ，但模块 $B_1 \sim B_5$ 既可以存在于该单元中也可以从该单元中删去。如果我们假设：在分支①处有两种选择，路径 $L1$ 或 $L2$ ；在分支②处也有两种选择，路径 $L11$ 或 $L12$ ；在分支③、④和⑤处均只有一种选择，分别为 $L3$ 、 $L4$ 和 $L5$ ；当然，只有选择了路径 $L1$ 才能选择路径 $L11$ 或 $L12$ ，也只有选择了路径 $L2$ 才能选择路径 $L5$ ，那么，单元 C 有七种实际可行的结构。

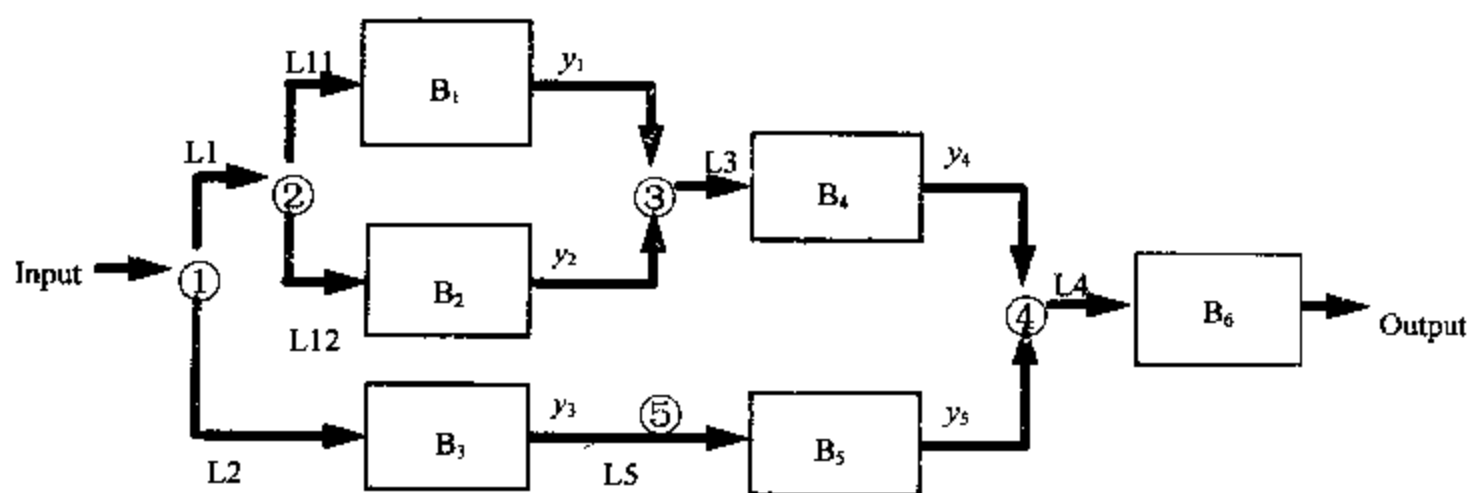


图 5.1 模拟集成电路单元的抽象结构及数据路径示意图

模拟集成电路单元综合一般分两步：首先进行拓扑选择，然后再优化器件尺寸，我们称这种方法为两步式单元电路综合法，简称“两步模式”。拓扑选择的任务是确定一个单元应包含哪些子模块，每个子模块应采用何种具体的电路结构。目前，主要有三类关于拓扑选择的方法：第一类是基于规则（或专家知识）的推理方法^[45-47, 51-56, 154]，其代表系统包括 OPASYN^[45]，An.Com^[46]，ARIADNE^[40, 47]，BLADES^[51]，CAMP^[52, 53] 和

ISAID^[54-56]等；第二类是采用层次式分解^[46,47,49,50,54-56]的编译方法，其代表系统有OASYS^[49]和SEAS^[50]，由于可层次式分解是模拟集成电路单元所固有的特征，因而一些基于规则推理的系统（如：An.Com, ARIADNE 和 ISAID 等）也采用了层次式分解技术；第三类是采用区间分析的拓扑选择法^[57]。在模拟集成电路的器件尺寸优化方面，如果按系统的特征分类，主要有基于电路模拟器的器件尺寸优化系统^[28,32]和用解析公式近似描述电路性能的器件尺寸优化系统^[34-42]，而获取模拟集成电路性能的解析公式的方法又可以再细分为手工推导法^[34-37]和符号分析法^[38-42]；如果按优化算法分类，则主要有确定性优化法^[28,29,34]和统计性优化法^[32,39,41,133]。

众所周知，一个子单元究竟采用什么样的电路拓扑结构不仅决定于用户所要求的电路性能指标，还依赖于该子单元周围别的子单元的情况，而且，只有在器件尺寸优化结束之后才有可能知道一个子单元的拓扑结构是否最佳。实质上，对于未给定器件尺寸的电路拓扑，我们仅能粗略估计它所能达到的电路性能指标，也就是说，只有在完成器件尺寸优化之后才能确定一个单元的拓扑结构是否可行。由此观之，“两步模式”在拓扑选择时所确定的拓扑有可能达不到用户所期望的性能指标，而且只有在器件尺寸优化失败后才有明确结论，因而需要在拓扑选择和器件尺寸优化之间反复回溯，在最坏情况下需要对每个可能的电路拓扑作一次器件尺寸优化才能完成单元电路的综合任务。在单元电路结构复杂，可选电路拓扑众多时，“两步模式”的计算成本将非常昂贵。更为严重的是，由于综合系统中预先设计的电路拓扑是有限的，而用户所期望达到的电路性能指标却参差不齐，因而常常出现综合失败的局面，这是专家系统、人工智能方法等到目前为止不能较好解决模拟集成电路综合问题的症结所在。

同时进行拓扑选择和器件尺寸优化^[58,59,89,155]是克服“两步模式”缺陷的有效方法，我们将这种方法简称为“一步模式”。“一步模式”的优点是拓扑结构和器件尺寸是同时给出的，因而能够准确预估电路性能，从而获得“最合适”的拓扑结构和“最佳”的器件尺寸。

在本章后续几节中，我们将首先给出“一步模式”的数学描述，然后介绍相应的求解策略，最后给出一些模拟集成电路单元的综合实例。

5.2 单元级综合的数学描述

由于任何一种模拟集成电路单元都有它们自己的共性，这使得我们可以为每一类模拟集成电路单元构造一个“超级电路”，“超级电路”的每一种有效的简化拓扑就分别对应于一个实际的模拟集成电路单元。假设某“超级电路”包含 l 个可简化模块和 $(b-l)$ 个不可简化模块，将这些模块分别记为 $B_i (i=1, 2, \dots, l)$ 和 $B_{l+1}, B_{l+2}, \dots, B_b$ ；我们再引入一个 l 维二进制向量 $Y = (y_1, y_2, \dots, y_l)^T$ ，用 Y 表示如何简化该“超级电路”： $y_i = 0$ 表示从该“超级电路”中删除模块 B_i ， $y_i = 1$ 表示保留该“超级电路”中的模块 B_i 。图 5.1 就是一个 $l=5$ 的“超级电路”。至于“超级电路”的工作条件和其中的器件尺寸则分别用 m 维向量 $V_B = (V_{B,1}, V_{B,2}, \dots, V_{B,m})^T$ 和 n 维向量 $X = (X_1, X_2, \dots, X_n)^T$ 表示。很显然， V_B 和 X 的取值还受一些约束条件的制约，我们记为 $\Gamma_i(X, V_B) \leq 0, i=1, 2, \dots, \gamma$ ；由于“超级电路”中模块 B_i 和 $B_j (i \neq j)$ 可能是相互关联的（例如：保留 B_i 就必须保留 B_j ），

因此 y_i 还必须满足一组用逻辑表达式描述的约束条件。在图 5.1 中, $y_1, y_2, \dots, y_5$ 就应满足下述约束:

$$\begin{cases} \text{IF } y_4 = 1 \text{ THEN } y_1 \text{ OR } y_2 = 1 \\ \text{IF } y_5 = 1 \text{ THEN } y_3 = 1 \\ y_3 \text{ AND } (y_1 \text{ OR } y_2) = 0, \quad y_1 \text{ AND } y_2 = 0 \end{cases} \quad (5.1)$$

由于逻辑型约束可以转化为代数型约束^[59], 因此, 我们将 Y 应满足的逻辑约束条件记为 $\Phi_i(Y) \leq 0, i=1, 2, \dots, \eta$ 。这样, 模拟集成电路单元综合问题就可以表示为:

$$\begin{aligned} \min_{X, Y, V_B} & \sum_{i=1}^l y_i \cdot A_i(X, V_B) + \sum_{i=l+1}^b A_i(X, V_B) \\ \text{s. t. } & \begin{cases} P_i(V_B, X, Y, \sigma_X, P_E) \leq 0, & i = 1, 2, \dots, k \\ \Gamma_i(X, V_B) \leq 0, & i = 1, 2, \dots, \gamma \\ \Phi_i(Y) \leq 0, & i = 1, 2, \dots, \eta \end{cases} \end{aligned} \quad (5.2)$$

其中, $A_i(X, V_B)$ 表示模块 B_i 的面积, σ_X 是 X 的标准差, P_E 代表用户所期望的性能指标, $P_i(V_B, X, Y, \sigma_X, P_E) \leq 0$ 则表示性能约束, 由于 P_E 的某些分量(如运放的共模信号抑制比 $CMRR$ 和电源信号抑制比 $PSRR$ 等)主要决定于器件之间的失配程度, 因此, P_i 也是 σ_X 的函数。需要特别强调的是: ①“超级电路”的构造和函数 $P_i(V_B, X, Y, \sigma_X, P_E)$ 的建立是先决条件, 是精确估计电路性能和同时确定电路拓扑和器件尺寸的基础; ②式(5.2)中仅考虑了芯片面积对成本的影响, 而忽略了芯片合格率对成本的影响, 这主要是为了简化矛盾, 否则, 问题(5.2)的求解将十分困难。

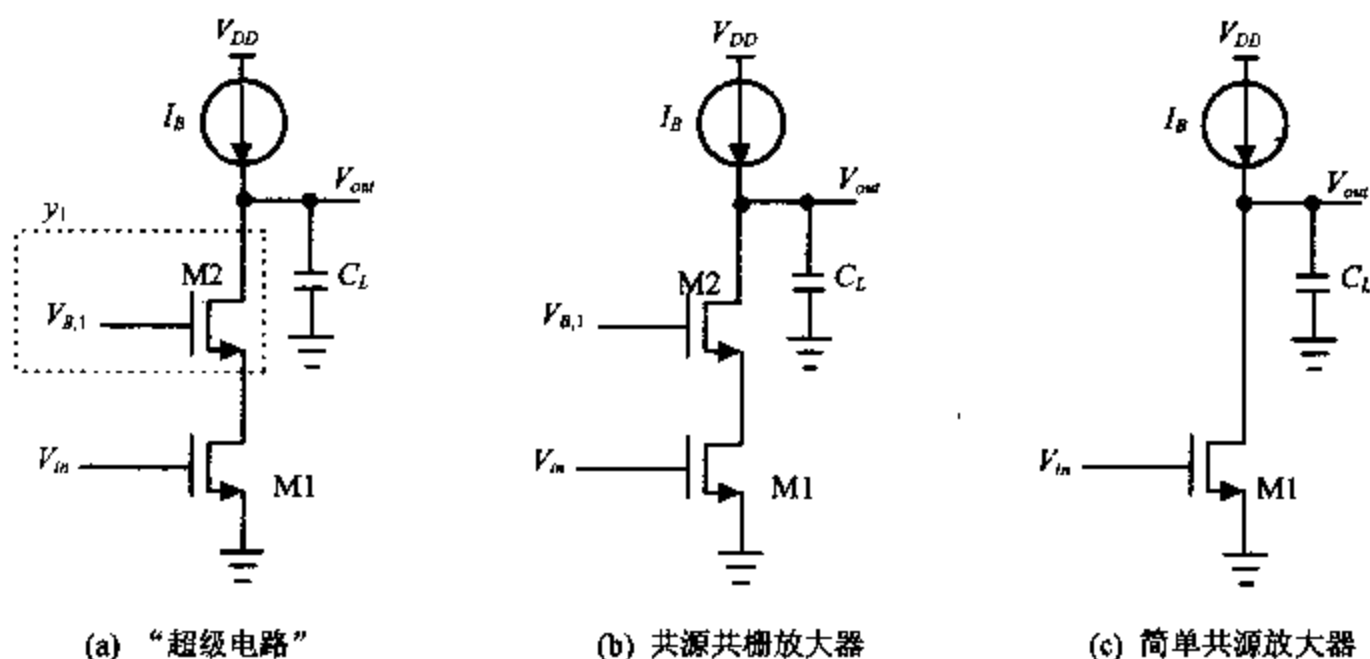


图 5.2 共源类放大器的“超级电路”及其简化电路

在结束本节之前, 我们再给一个简单实例以具体说明如何构造式(5.2)。图 5.2(a)是共源类放大器的“超级电路”, 当 $y_1=1$ 时, 该“超级电路”简化为共源共栅放大器, 参见图 5.2(b); 当 $y_1=0$ 时, 该“超级电路”简化为简单共源放大器, 参见图 5.2(c)。容易证明, 该“超级电路”的输出电阻 R_o 、电压增益 A_v 、摆率 S_R 和单位增益带宽 f_u 可以分别表示为:

$$\begin{cases} R_o = y_1 R_o^C(X, V_{B,1}, V_{DD}, I_B) + (1 - y_1) R_o^S(X, V_{DD}, I_B) \\ A_v = g_{m1}(X, V_{B,1}, V_{DD}, I_B) R_o \\ S_R = I_{DS1}(X, V_{B,1}, V_{DD}, I_B) / C_L \\ f_u = g_{m1}(X, V_{B,1}, V_{DD}, I_B) / 2\pi C_L \end{cases} \quad (5.3)$$

其中, $R_o^C(X, V_{B,1}, V_{DD}, I_B)$ 和 $R_o^S(X, V_{DD}, I_B)$ 分别为共源共栅放大器和简单共源放大器的输出电阻, $X = (W_1, L_1, W_2, L_2)^T$, 其中, L_i, W_i, g_{mi} 和 I_{DSi} 则分别代表第 i 个 MOSFET 的沟道长度、沟道宽度、跨导和源漏极电流。如果要求共源类放大器的电压放大倍数、摆率和单位增益带宽分别不低于 A_{VE}, S_{RE} 和 f_{uE} , 并且不对 $X, V_{B,1}, V_{DD}$ 和 I_B 的取值范围作任何限制, 由式(5.2)可以看出, 这是 $k=3, \gamma=\eta=0$ 的情形, 因此, 共源类放大器综合问题可以表示为:

$$\begin{aligned} & \min_{X, y_1, V_{B,1}} W_1 L_1 + y_1 W_2 L_2 \\ & s. t. \begin{cases} A_{VE} - g_{m1}(X, V_{B,1}, V_{DD}, I_B) R_o \leq 0 \\ S_{RE} - I_{DS1}(X, V_{B,1}, V_{DD}, I_B) / C_L \leq 0 \\ 2\pi C_L \cdot f_{uE} - g_{m1}(X, V_{B,1}, V_{DD}, I_B) \leq 0 \end{cases} \end{aligned} \quad (5.4)$$

5.3 同时确定电路拓扑和器件尺寸的策略

“一步模式”首先应当解决的问题是为每一类模拟集成电路单元建立一个“超级电路”, 并写出(5.2)中 P_i 的表达式。我们将在下一节中给出几个“超级电路”, 本节将扼要地介绍问题(5.2)的求解方法。

问题(5.2)是一个混整型非线性数学规划问题, P. C. Maulik 首先将二进制变量 y_i 当作 $[0,1]$ 上的连续变量来处理^[58,59], 然后用 NPSOL^[90] 来获得一组初始解, 很显然, 按这种方式所获得的初始解中会有一些 y_i 既不靠近 0, 也不靠近 1, 因而不能确定二进制变量 y_i 应取何值才是最佳的; 为进一步确定 y_i , P. C. Maulik 只好求助于分支界定法^[82]。由此可见, P. C. Maulik 的求解方法仍然不能同时确定拓扑结构 Y 和器件尺寸 X , 因而也会陷入“两步模式”曾面临过的困境之中。

同时确定(5.2)中的 Y, X 和 V_B 是实现“一步模式”的关键, 这里采用第二章所介绍的多分子模拟退火法, 该方法有很好的全局收敛性, 并能同时确定电路结构和器件尺寸^[89]。模拟退火法中能量函数按下式定义:

$$E(Z) = \sum_{i=1}^l y_i \cdot A_i(X, V_B) + \sum_{i=l+1}^b A_i(X, V_B) + \sum_{i=1}^k C_{P,i} P_i + \sum_{i=1}^{\gamma} C_{\Gamma,i} \Gamma_i + \sum_{i=1}^{\eta} C_{\Phi,i} \Phi_i \quad (5.5)$$

其中, $Z = (y_1, \dots, y_l, X_1, \dots, X_n, V_{B,1}, \dots, V_{B,m})^T$, 系数 $C_{P,i}, C_{\Gamma,i}, C_{\Phi,i}$ 的取值规则如下:

$$C_{P,i} = \begin{cases} 0 & \text{if } P_i \leq 0 \\ > 0 & \text{else} \end{cases}, C_{\Gamma,i} = \begin{cases} 0 & \text{if } \Gamma_i \leq 0 \\ > 0 & \text{else} \end{cases}, C_{\Phi,i} = \begin{cases} 0 & \text{if } \Phi_i \leq 0 \\ > 0 & \text{else} \end{cases} \quad (5.6)$$

在处理优化设计变量时, 采用下述与 P. C. Maulik 不同的策略^[3]:

- (1) 保持 y_i 为二进制变量;

(2) 将 X 的每一个分量按生产工艺所允许的最小尺寸(又称特征尺寸)离散化,例如:对于 $1\mu\text{m}$ 工艺, X 的每一个分量只能取 $1\mu\text{m}$, $2\mu\text{m}$, $3\mu\text{m}$, ..., 这样既能使器件尺寸满足设计规则的要求,又可以提高模拟退火法的效率;

(3) 对 V_B 的离散化要比对 X 的离散化繁琐些,因为我们不能预先确定应按何种尺度作离散化,为此,我们将 V_B 的每一个分量看作 7 个变量的和,这 7 个变量分别按 $1\text{V}(1\text{A})$, $0.1\text{V}(0.1\text{A})$, $10\text{mV}(10\text{mA})$, $1\text{mV}(1\text{mA})$, $0.1\text{mV}(0.1\text{mA})$, $10\mu\text{V}(10\mu\text{A})$ 或 $1\mu\text{V}(1\mu\text{A})$ 来离散化,即:用 7 个离散化的电压源的串联(或电流源的并联)来代替 1 个电压源(或电流源)^[39]。

5.4 高性能运算放大器的综合

随着信息科学和微电子技术的迅速发展,需要将模拟电路(如 A/D, D/A, 模拟滤波器等)同数字电路集成到同一芯片中。在绝大多数数模混合型专用集成电路中,模拟电路的比重都很小,但是,由于在模拟集成电路的设计过程中需要处理非常复杂的设计约束,因而其设计成本反而比其中的数字电路高许多。运算放大器是最常用的一种模拟集成电路单元,是各种模拟集成电路的核心,其花样和种类极其丰富^[48],应用范围也极其广阔^[127,156-163]。例如:我们在第四章中所遇到的跨导放大器、积分器、电流/电压转换器等一般都是用运算放大器实现的。不言而喻,不同应用背景所需要的运放性能是各不相同的,因此,实现运算放大器的自动综合将会大幅度降低数模混合型专用集成电路的设计成本,缩短专用集成电路的上市周期,提高其市场竞争力。

在过去十年中,人们在模拟集成电路的设计自动化,特别是在运算放大器的自动综合方面取得了一定进展^[45,46,48-56,58,59,66,91,164,165],这些综合方法大多针对普通性能的两级运放,不能实现高性能运算放大器的自动综合。考虑到随着传感器、生物医学器件^[162]、系统集成技术、移动通信和各种便携式电器^[163]的发展,几乎所有的数模混合型专用集成电路都离不开诸如宽带、高速、低电压、低功耗和低噪声等高性能的运算放大器^[156-163],本节将以双极型工艺为例,介绍高性能运算放大器的电路结构和相应的自动综合方法。

5.4.1 层次式分解策略与高性能运放的拓扑综合

数字集成电路综合系统均采用了层次式分解(Hierarchical Decomposition)方法,由于模拟集成电路本身具有可层次式分解的特征,因而,这种思想也渗透到了模拟集成电路领域^[46,49,51,54,55]。

就运算放大器而言,不同的应用背景所侧重的性能是各不相同的:有时要求运放有较低的噪声,有时需要宽频带的运放,而有时则希望运放的功耗尽可能低;当然,适应不同应用背景的运算放大器的拓扑结构也是千差万别的。尽管如此,我们还是可以将运

放的结构分解为图 5.3 的模式。运算放大器的拓扑综合就是根据用户所要求的运放性能指标确定偏置电路、输入级、中间级和输出级应采用何种电路拓扑才是最佳的。拓扑综

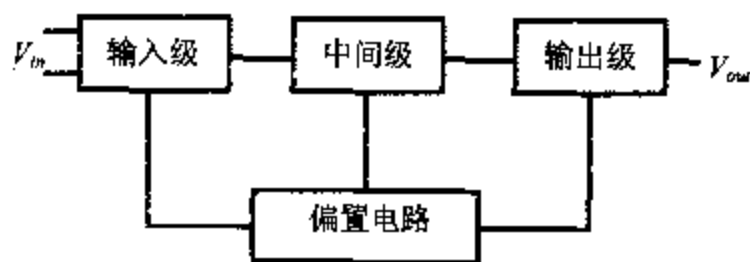


图 5.3 运算放大器的一般结构

合一般又分为结构综合、功能块综合和确定电路形式等三个层次。

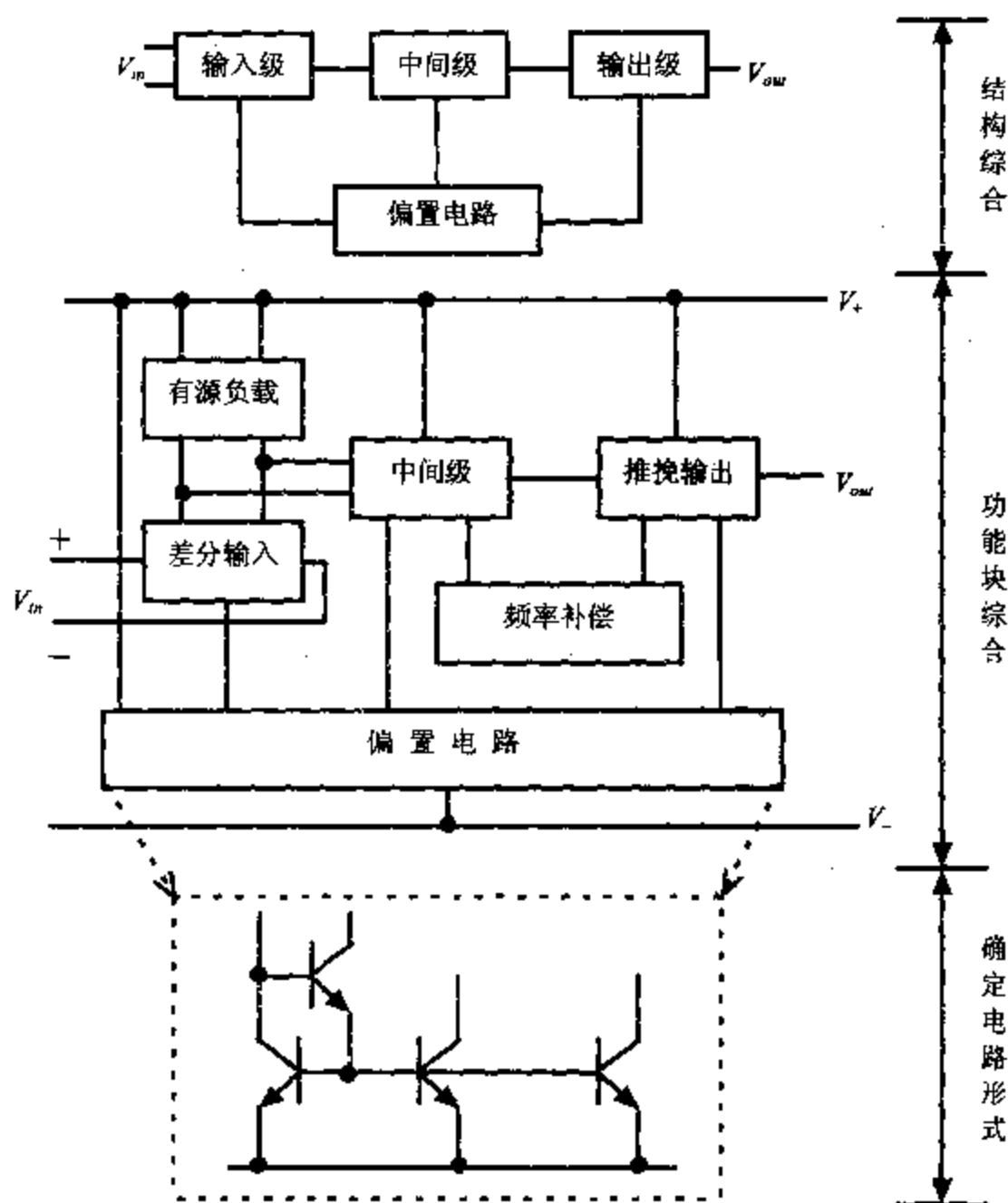


图 5.4 运算放大器的层次式分解示意图

图 5.4 是运算放大器的层次式分解示意图。层次式分解的第一步是结构综合，其任务是根据用户所给定的电路性能指标，决定采用单级运放还是多级运放，以及是否需要输出级等，结构综合的结果是用功能块描述的抽象电路；第二步是功能块综合，其任务是用各种电路模块来实现结构综合所分解出的各功能块；第三步是为每一个电路模块选择适当的具體电路。应当特别指出，在所综合出的具体电路中，有些器件之间是有相互约束关系的（例如：输入级的两个差分对管应完全对称），这是模拟集成电路网表（netlist）有别于数字集成电路的主要特征，这种约束关系在布局布线时至关重要^[65,69-71]。

5.4.2 偏置电路的综合

运放的各项性能指标均受偏置电路的影响，因此，抗电源噪声强、热稳定性好的偏置电路是保证高性能运放稳定工作的必要条件。这里，我们以双极型工艺为例介绍一种高稳定性的恒流源。至于高稳定性的 MOS 偏置电路，可以参见文献[161]。

普通的 Widlar 或 Wilson 恒流源的输出电流受电源电压和双极型晶体管结温的影响很大，不利于降低运放的失调和噪声，也不利于提高运放的电源信号抑制比。利用能隙电压源的设计思想，我们采用图 5.5 所示的恒流源作为高性能运放的偏置电路。当双极型晶体管 Q1 和 Q0 具有相同的面积因子、且 $V_{CC} - V_{EE} > 4V_{BE}$ 时，Q0~Q8 都将导通，

则:

$$I_{E2} = \frac{V_{BE}}{R_2}$$

$$I_{E4} \approx I_{E6}$$

$$I_{E3} \approx I_{E5}$$

$$V_{BE4} + V_{BE5} = V_{BE3} + V_{BE6} + R(I_{E3} + I_{E6})$$

其中, I_{Ej} 和 V_{BEj} 分别为第 j 个晶体管的发射极电流和发射结电压。由此可得:

$$\frac{kT}{q} \ln \frac{I_{E4}}{I_S A_4} + \frac{kT}{q} \ln \frac{I_{E5}}{I_S A_5} = R(I_{E3} + I_{E4}) + \frac{kT}{q} \ln \frac{I_{E3}}{I_S A_3} + \frac{kT}{q} \ln \frac{I_{E4}}{I_S A_6}$$

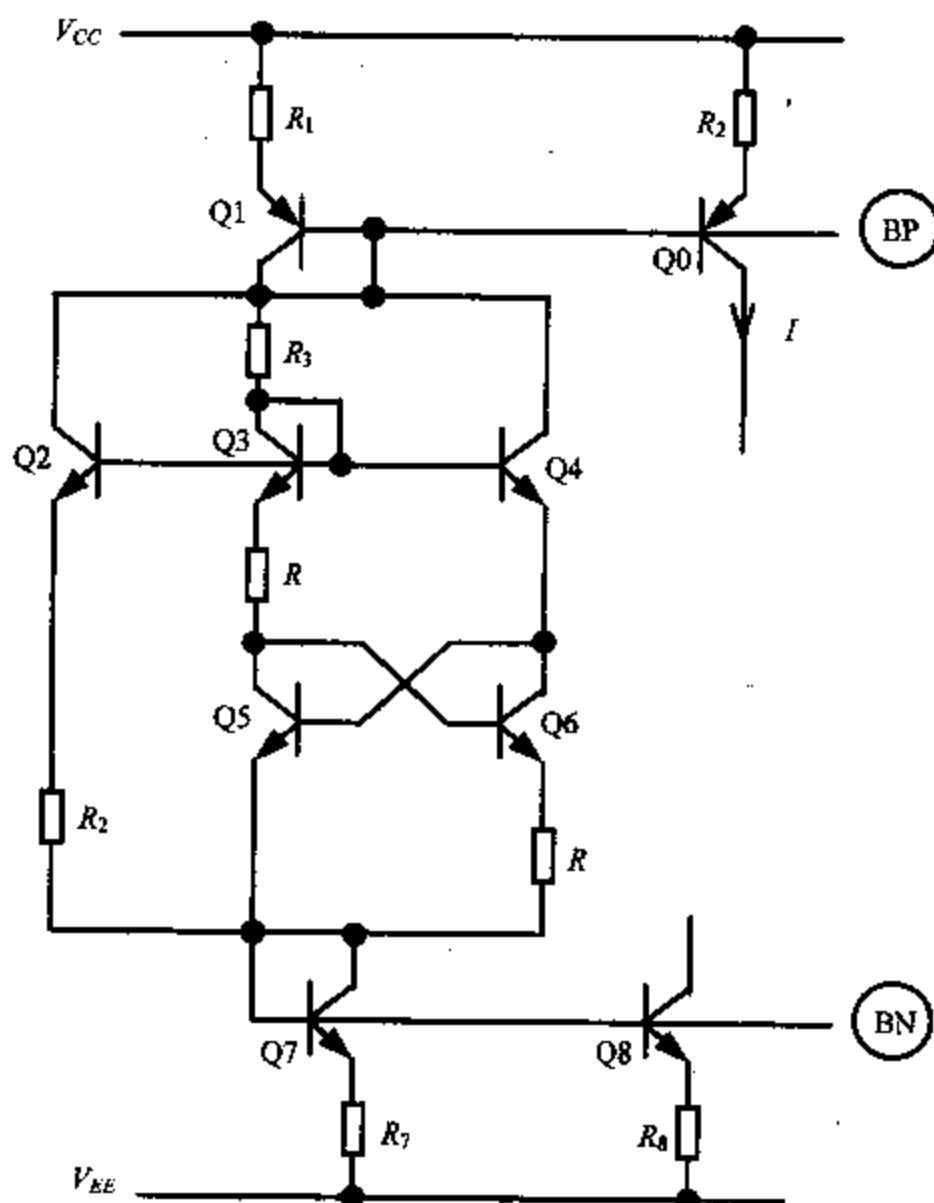


图 5.5 高稳定性的双极型恒流源

其中, k 为 Boltzmann 常数, q 为电子电荷, T 为绝对温度, I_S 为双极型晶体管的饱和电流。从上式很容易导出:

$$I_{E3} + I_{E4} = \frac{kT}{qR} \ln \frac{A_3 A_6}{A_4 A_5}$$

因此, 图 5.5 所示恒流源的输出电流为:

$$I = \frac{R_1}{R_2} (I_{E2} + I_{E3} + I_{E4}) = \frac{R_1}{R_2} \left\{ \frac{kT}{qR} \ln \left(\frac{A_3 A_6}{A_4 A_5} \right) + \frac{V_{BE}}{R_2} \right\} \quad (5.7)$$

由式(5.7)可以看出: ① I 与电源电压无关, 因此, 该偏置电路不仅不受电源波动的影响, 而且当电源电压大幅度降低时(当然, 电源电压不能低于 $4V_{BE}$)仍然能稳定工作, 可适用于低电压、低功耗的场合; ②由于 V_{BE} 具有负温度系数, 因此, 只要使

$A_3A_6 > A_4A_5$, 并适当地选择 R_1 和 R_2 就可以减小温度变化对输出电流 I 的影响。

图 5.5 所示的高稳定性恒流源所需的器件较多, 相应的芯片面积也就比较大, 因此, 在一般的运算放大器中, 偏置电路只要选用 Widlar 或 Wilson 恒流源即可。我们按第四章所介绍的方法, 引入 3 个二进制变量 y_1, y_2 和 y_3 , 用它们表示偏置电路的结构, 即:

$y_1=1$ 表示具有高稳定性的电路结构

$y_2=1$ 表示 Widlar 恒流源

$y_3=1$ 表示 Wilson 恒流源

由于变量 y_1, y_2 和 y_3 中有且只有一个为 1, 因此, 它们应满足约束条件:

$$y_1 + y_2 + y_3 = 1 \quad (5.8)$$

这里, y_1, y_2 和 y_3 的引入主要是为了建立“超级电路”和同时进行拓扑选择和器件尺寸优化, 我们将在后面作详细说明。

5.4.3 输入级与中间级的综合

输入级对运放的带宽(f_{BW})、压摆率(S_R)、失调电压(V_{OS})、共模信号抑制比($CMRR$)、电源信号抑制比($PSRR$)和噪声性能等均有很大的影响, 采用差分对作为输入以减小失调、提高 $CMRR$ 和 $PSRR$ 是必要的。对于宽带运放, 采用折叠式共射共基结构的输入级(图 5.6)是提高共模信号输入范围的一条有效途径; 对于 MOS 电路则可以采用折叠式共源共栅输入级以提高共模信号的输入范围^[66,166]。在图 5.6 中, 运放的开环主极点由补偿电容 C 和电阻 R_1 和 R_2 决定, 因此, 其单位增益频率 f_u 和摆率 S_R 可以表示为:

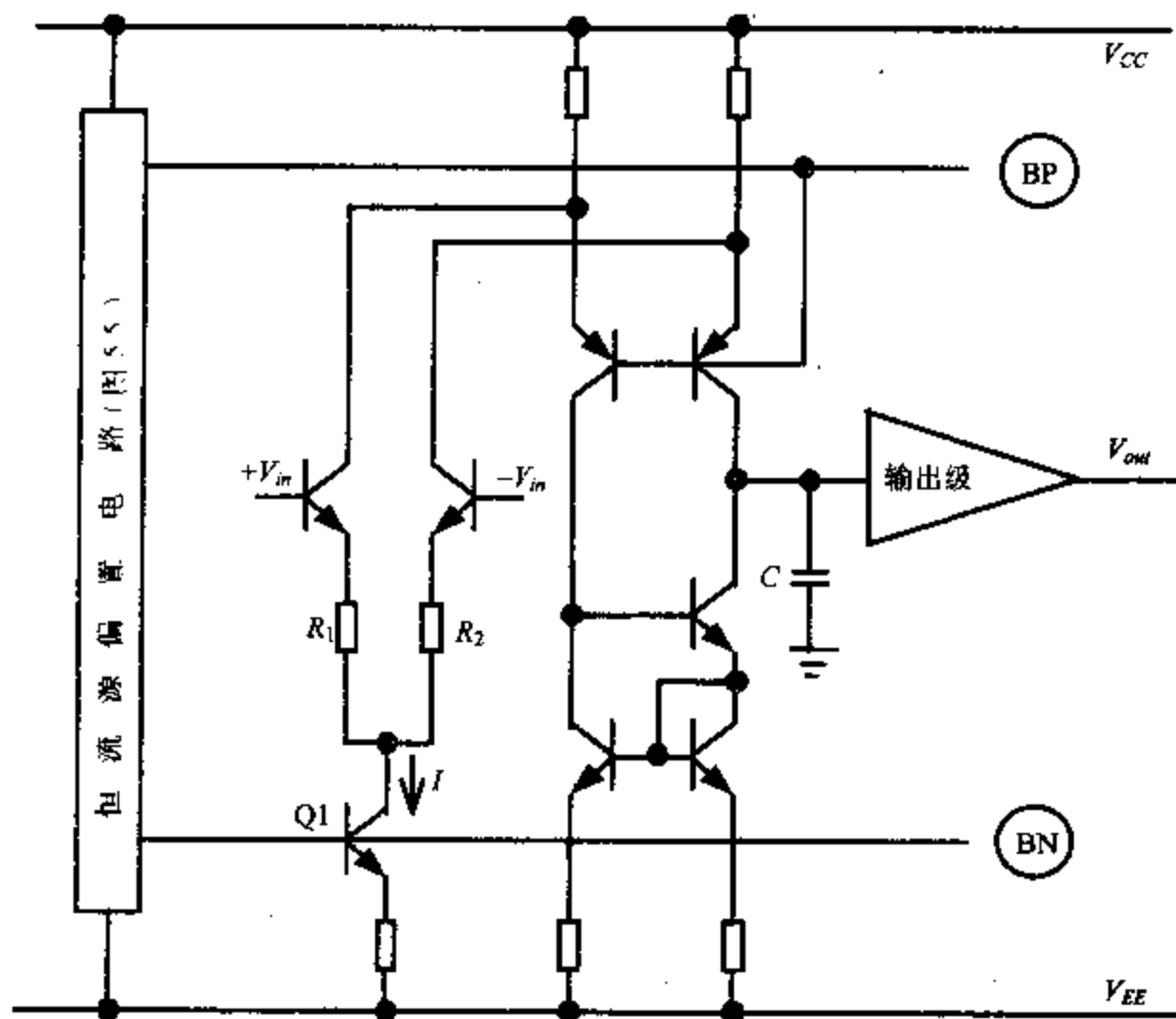


图 5.6 折叠式共射共基输入级

$$f_u = [2\pi(R_1 + R_2)C]^{-1}, S_R = I/C \quad (5.9)$$

其中, I 是流过 $Q1$ 的集电极电流。考虑到噪声主要来自于 R_1 和 R_2 , 所以噪声电压的均方值为:

$$\bar{V}_n^2 = 4kT(R_1 + R_2)f_u \quad (5.10)$$

从式(5.9)、(5.10)不难看出, 若要求运放有较低的功耗则应使 I 较小, 为保证 S_R 和 f_u 不变, 就必须增大 $(R_1 + R_2)$, 从而导致热噪声增加的局面。在电路综合过程中, 应根据不同的需要来确定 I, C, R_1 和 R_2 的值。

如果我们增大输入级差分对管的尺寸, 则可以减小晶体管的基区电阻, 这对提高运放的增益和降低热噪声都是有益的; 但是, 其副作用是会增大极间电容, 从而使运放的带宽下降。因此, 若对运放的增益、热噪声和带宽等指标均有要求时, 确定输入级差分对管的尺寸是很重要的。为了保证输入级的对称性, 将差分对管分为 4 个对称晶体管是非常必要的^[167]。

为减小失调电压和信号的非线性失真, 常常将中间放大级分为两级, 前一级为缓冲级, 并采用平衡对称结构, 从而有效地减小后级电路的不对称性对输入级的影响; 当然, 后一级也应尽量采用平衡对称结构。图 5.7 是一种适于低失调、低失真系数的双极型运放的输入级和中间级结构。

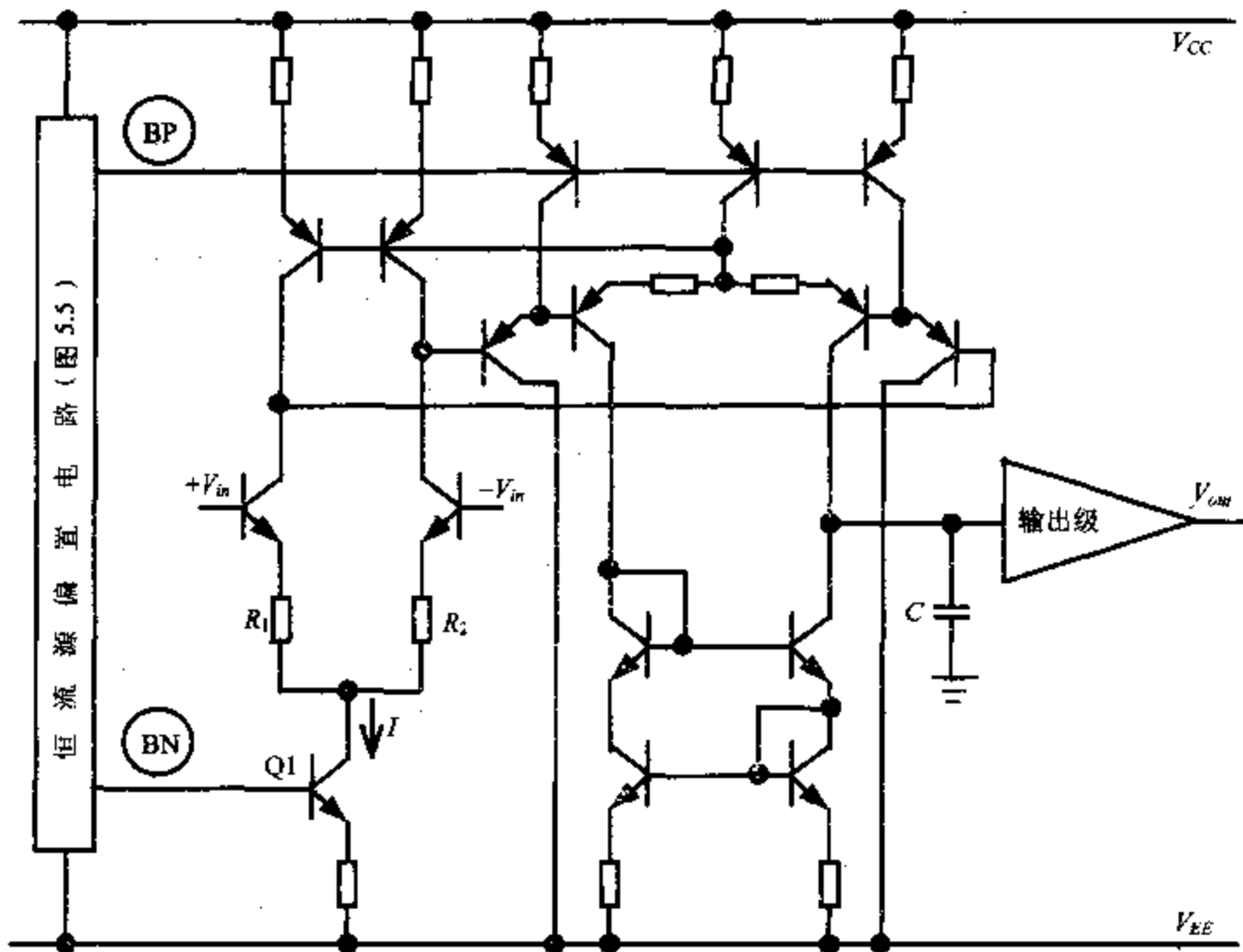


图 5.7 平衡式输入级、中间级

中间增益级对运放的带宽和增益均有直接的影响, E. M. Cherry 和 D. E. Hooper^[168]利用负反馈技术所设计的宽带放大器是值得借鉴的, 图 5.8 是具体的电路结构。图 5.8 的特点是 $Q5, Q6$ 分别和 R_5, R_6 构成串联负反馈, $Q7, Q8$ 分别和 $Q11, R_7$ 以及 $Q12, R_8$ 构成并联负反馈, 而且 R_5 和 R_6 在高频时被电容 C_S 短路, 这对于提高运放的带

放稳定性(相位余量)的影响,这种结构适合于负载较重的场合。在图 5.9(b)、(c)中, Q1, Q2 也有采用复合达林顿管的情形。图 5.9(b)、(c)不适合于作低功耗运放的输出级,因为没有充分利用 Q3, Q4 和 R。图 5.9(d)是低功耗运放所采用的输出级电路拓扑。

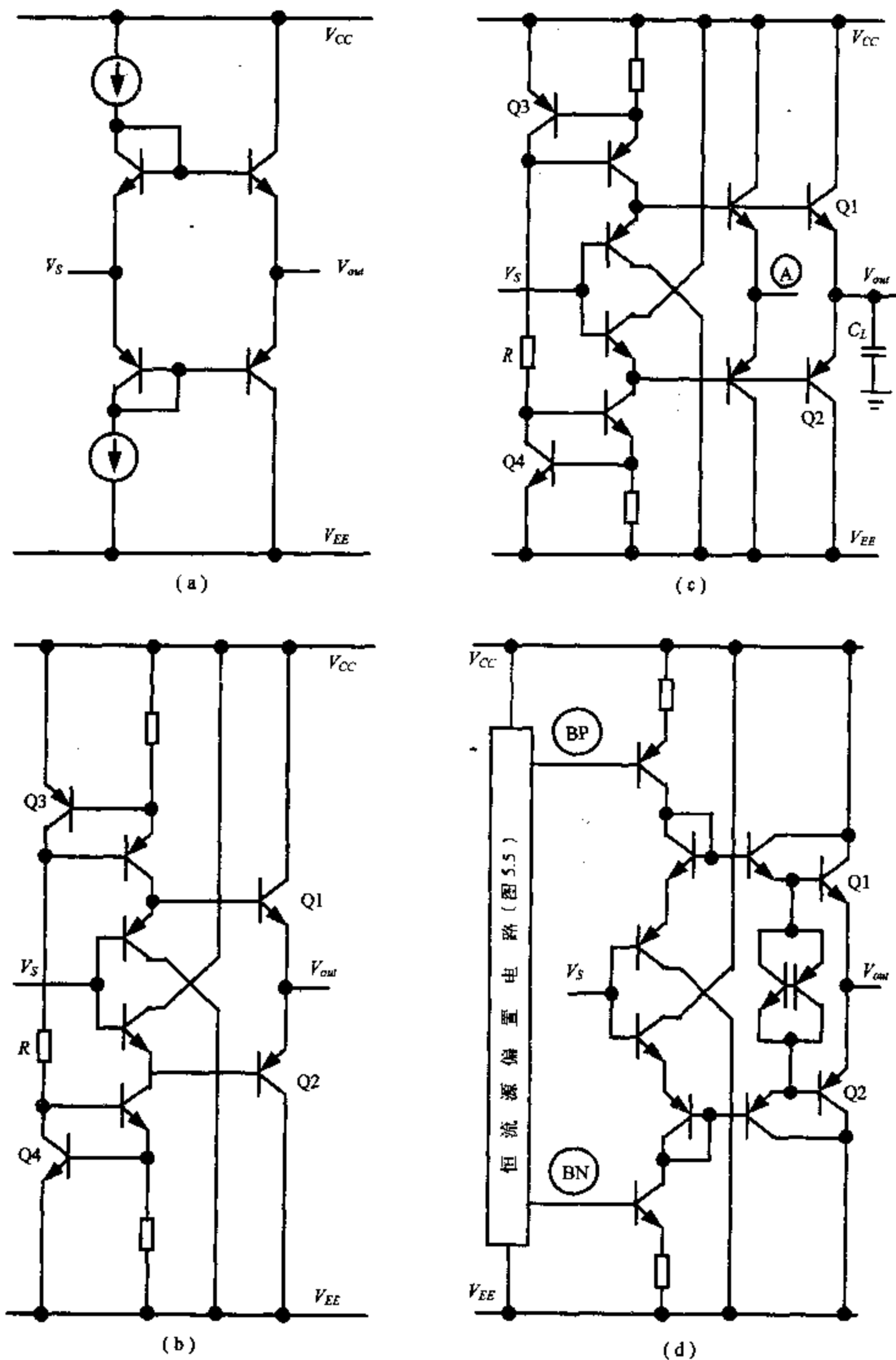


图 5.9 双极型输出级(单端)的电路结构

倘若需要双端输出,可以采用 W. Pohlmann 提出的输出级^[127]。图 5.10 是具体的电路结构。这是从 E. M. Cherry 和 D. E. Hooper 的负反馈宽带放大器^[168]演变而来的,其中, Q11 和 Q12 最好采用最小尺寸的晶体管,以便最大程度地减小结电容对工作速度的影响。

在数模混合信号处理芯片中,如果运放是驱动片内电路,且负载很轻,就可以不要输出级,这是减小芯片面积、降低芯片功耗的最佳方案。

图 5.9 和图 5.10 所示的 5 种输出级各有特点,也应根据具体的设计指标来确定选用适当的输出级。为了表示输出级的拓扑选择结果,需要引入 7 个二进制变量 $y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}$, 具体规则如下:

$y_7=1$ 表示通常的甲乙类放大器, 图 5.9(a)

$y_8=1$ 表示图 5.9(b)的结构

$y_{12}=1$ 表示图 5.9(b)中 Q1, Q2 为复合达林顿管的情形

$y_9=1$ 表示图 5.9(c)的结构

$y_{13}=1$ 表示图 5.9(c)中 Q1, Q2 为复合达林顿管的情形

$y_{10}=1$ 表示输出级为低功耗型输出级, 图 5.9(d)

$y_{11}=1$ 表示输出级为图 5.10 所示的双端输出型输出级

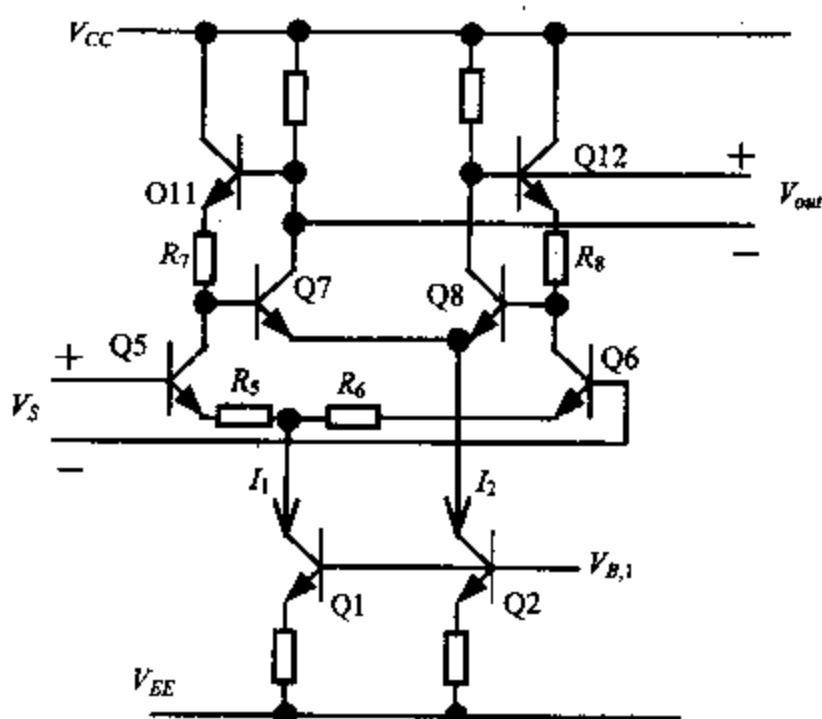


图 5.10 双极型输出级(双端输出)的电路结构

当然 $y_7 \sim y_{11}$ 这 5 个变量不能有两个或两个以上同时为 1, 但可以同时为 0, 这相当于不要输出级的情形。当 $y_{12}=1$ 时, y_8 必须为 1; 而 $y_{13}=1$ 时, y_9 也必须为 1。因此, 它们应满足约束条件:

$$\sum_{i=7}^{11} y_i \leq 1$$

$$\text{IF } y_{12} = 1, \text{ THEN } y_8 = 1$$

$$\text{IF } y_{13} = 1, \text{ THEN } y_9 = 1$$

我们可以将其中的逻辑型约束改写为下述代数型约束:

$$\begin{cases} \sum_{i=7}^{11} y_i \leq 1 \\ y_8 \geq y_{12} \\ y_9 \geq y_{13} \end{cases} \quad (5.12)$$

5.4.5 高性能双极型运算放大器的“超级电路”

在前几节中,我们介绍了高性能的偏置电路、几种输入级、中间级和输出级,综合高性能模拟运算放大器的关键是根据设计指标确定这些模块的电路结构和相应的器件尺寸。为此,我们将各种输入级、中间级和输出级组合起来,形成一个高性能双极型运放的“超级电路”^[165],参见图 5.11。该“超级电路”的主要特征是它包含有一组表示电路结构的二进制向量 $Y = (y_1, y_2, \dots, y_{15})^T$ 。

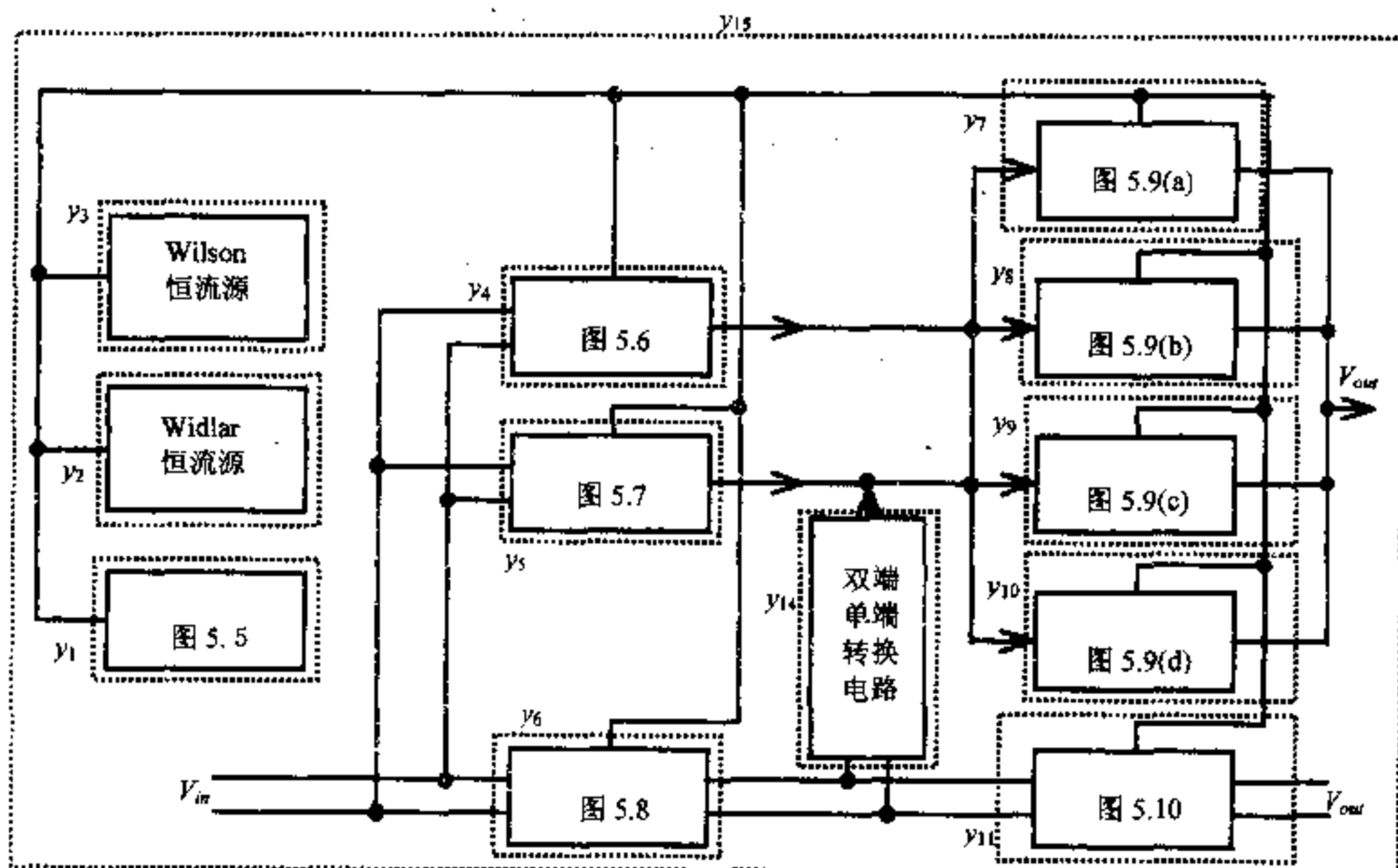


图 5.11 高性能运算放大器的“超级电路”

我们首先讨论 Y 应遵守的约束条件。由于前面已经对 $y_1 \sim y_{13}$ 作过详细的说明,就不再重复了,这里主要讨论 y_{14} 和 y_{15} 。 $y_{14}=1$ 表示保留“超级电路”中的双端/单端转换电路, $y_{14}=0$ 则表示去掉“超级电路”中的双端/单端转换电路; $y_{15}=1$ 表示采用双极工艺, $y_{15}=0$ 则表示采用别的集成电路生产工艺,如 MOS 工艺。很显然,当采用图 5.10 所示的输出级时,输入级和中间级应采用图 5.8 的结构,因此, y_6 和 y_{11} 应满足下述逻辑约束条件:

$$\text{IF } y_{11} = 1, \text{ Then } y_6 = 1$$

我们同样可以将上述逻辑约束改写为下述代数约束:

$$y_6 \geq y_{11} \quad (5.13)$$

类似地,当保留“超级电路”中的双端/单端转换电路时,输入级和中间级只能采用图 5.8 的结构,而输出级则不能选择图 5.10 的电路结构,因此, y_6, y_{11}, y_{14} 应满足下述逻辑

辑型约束条件:

$$\text{IF } y_{14} = 1, \text{ THEN } y_8 = 1, y_{11} = 0$$

即:

$$\begin{cases} y_{11} + y_{14} \leq 1 \\ y_8 \geq y_{14} \end{cases} \quad (5.14)$$

我们将“超级电路”中向量 Y 应满足的约束条件(5.8)、(5.11)、(5.12)、(5.13)和(5.14)罗列起来就可以得到下述约束条件:

$$\begin{cases} y_1 + y_2 + y_3 = 1 \\ y_4 + y_5 + y_6 = 1 \\ \sum_{i=7}^{11} y_i \leq 1 \\ y_8 \geq y_{12} \\ y_9 \geq y_{13} \\ y_{11} + y_{14} \leq 1 \\ y_8 \geq y_{11} \\ y_8 \geq y_{14} \end{cases} \quad (5.15)$$

上式是 Y 应当遵守的代数型约束,它相当于式(5.2)中的 $\Phi_i(Y) \leq 0, i=1, 2, \dots, \eta$ 。当然,将它们写成逻辑型约束也是很容易的事,这里就不一一介绍了。

建立“超级电路”的另一个重要任务是给出其电性能的解析表达式。设 X 代表器件参数, σ_X 是其标准差, V_B 表示偏置条件,则可以按第 5.2 节所介绍的方法写出该“超级电路”的各种性能函数的表达式 $P(Y, X, V_B, \sigma_X)$ 。

这里,我们仅以电压增益和输出电阻为例介绍如何写出“超级电路”的电性能 $P(Y, X, V_B, \sigma_X)$ 的解析表达式。

假设:图 5.6、图 5.7、图 5.8、图 5.9(a)、图 5.9(b)、图 5.9(c)、图 5.9(d)和图 5.10 的电压增益分别为 $G_{V,4}(X, V_B, \sigma_X)$, $G_{V,5}(X, V_B, \sigma_X)$, $G_{V,6}(X, V_B, \sigma_X)$, $G_{V,7}(X, V_B, \sigma_X)$, $G_{V,8}(X, V_B, \sigma_X)$, $G_{V,9}(X, V_B, \sigma_X)$, $G_{V,10}(X, V_B, \sigma_X)$ 和 $G_{V,11}(X, V_B, \sigma_X)$; 其输出电阻分别为 $R_{O,4}(X, V_B, \sigma_X)$, $R_{O,5}(X, V_B, \sigma_X)$, $R_{O,6}(X, V_B, \sigma_X)$, $R_{O,7}(X, V_B, \sigma_X)$, $R_{O,8}(X, V_B, \sigma_X)$, $R_{O,9}(X, V_B, \sigma_X)$, $R_{O,10}(X, V_B, \sigma_X)$ 和 $R_{O,11}(X, V_B, \sigma_X)$; 而双端/单端转换电路的电压增益和输出电阻分别为 $G_{V,14}(X, V_B, \sigma_X)$ 和 $R_{O,14}(X, V_B, \sigma_X)$ 。则:

$$\begin{cases} G_{V,8}(X, V_B, \sigma_X) = y_{12}G_{V,8D}(X, V_B, \sigma_X) + (1 - y_{12})G_{V,8S}(X, V_B, \sigma_X) \\ R_{O,8}(X, V_B, \sigma_X) = y_{12}R_{O,8D}(X, V_B, \sigma_X) + (1 - y_{12})R_{O,8S}(X, V_B, \sigma_X) \\ G_{V,9}(X, V_B, \sigma_X) = y_{13}G_{V,9D}(X, V_B, \sigma_X) + (1 - y_{13})G_{V,9S}(X, V_B, \sigma_X) \\ R_{O,9}(X, V_B, \sigma_X) = y_{13}R_{O,9D}(X, V_B, \sigma_X) + (1 - y_{13})R_{O,9S}(X, V_B, \sigma_X) \end{cases}$$

其中,函数 $G_{V,8D}$ 和 $G_{V,8S}$ 分别为图 5.9(b)中 Q1, Q2 为复合达林顿管和单管时的电压增益,别的函数的意义与此雷同。因此,图 5.11 所示的“超级电路”的电压增益 $A_V(Y, X, V_B, \sigma_X)$ 可以表示为:

$$A_V = y_{15} \times \left(\sum_{j=4}^6 y_j G_{V,j} \right) \times (y_{14} G_{V,14} + 1 - y_{14}) \times \left(1 + \sum_{j=7}^{11} (y_j G_{V,j} - y_j) \right)$$

$$+ (1 - y_{15})A_{V, \text{other}} \quad (5.16)$$

其中, $A_{V, \text{other}}$ 代表别的“超级电路”的电压增益。图 5.11 所示的“超级电路”的输出电阻 $R_o(Y, X, V_B, \sigma_X)$ 可以表示为:

$$R_o = y_{15} \times \left\{ \sum_{j=7}^{11} y_j R_{o,j} + \left(1 - \sum_{j=7}^{11} y_j \right) \times \left[\sum_{j=4}^6 y_j R_{o,j} + y_{14} R_{o,14} - y_{14} y_8 R_{o,6} \right] \right\} \\ + (1 - y_{15}) R_{o, \text{other}} \quad (5.17)$$

其中, $R_{o, \text{other}}$ 是别的“超级电路”的输出电阻。

仿效上述建立 $A_v(Y, X, V_B, \sigma_X)$ 和 $R_o(Y, X, V_B, \sigma_X)$ 的解析表达式的思路, 写出带宽、功耗、电源信号抑制比、摆率、等效噪声电压、相位余量等电性能的表达式是很容易的, 就不一一枚举了。

给定 Y, X 和 V_B 之后估算芯片面积是很容易的事, 因此利用 $P(Y, X, V_B, \sigma_X)$ 和式 (5.15) 就可以将高性能模拟集成运算放大器的综合问题转化为形如式 (5.2) 的混整型非线性数学规划问题, 然后, 可以采用多分子模拟退火法同时进行拓扑选择和器件尺寸优化, 从而获得满足用户需求的电路结构和器件尺寸。

5.5 综合实例

例 5.1 宽带运算放大器的综合。“超级电路”如图 5.11 所示。图 5.12 是所获得的电路结构 ($y_1 = y_4 = y_8 = y_{15} = 1$, 别的 y_i 为 0), 表 5.1 是设计指标和综合结果的对比 (负载为 $2k\Omega$), 该表中还同时列出了双极型晶体管的部分 SPICE 模型参数, 在本章后面所给出的例 5.2 中, PNP 管和 NPN 管的模型参数与此相同。

表 5.1 宽带运算放大器的设计指标、综合结果及所使用器件的 SPICE 模型参数

	设计指标	综合结果	SPICE 模型名	NPN 管	PNP 管
增 益	$> 60\text{dB}$	60dB	BF	100	40
单位增益带宽	$> 500\text{MHz}$	650MHz	TF	5.0ps	10ps
摆 率	$> 150\text{V}/\mu\text{s}$	160 V/ μs	VAF	30V	10V
相 位 余 量	$> 45^\circ$	50°	RB	50 Ω	40 Ω
共模信号抑制比	$< -60\text{dB}$	-70dB	RE	0.5 Ω	1.0 Ω
电源信号抑制比	$< -60\text{dB}$	-65dB	CJE	0.4pF	0.8pF
等效噪声电压	$< 8\text{E}-9 \text{ V}/\sqrt{\text{Hz}}$	$5.5\text{E}-9 \text{ V}/\sqrt{\text{Hz}}$	CJC	0.2pF	0.5pF
非线性失真	$< -65\text{dB}$	-75dB	CJS	0.3pF	1.0pF
功 耗	$< 200\text{mW}$	150mW	IKF	400mA	300mA

例 5.2 低功耗运算放大器的综合。“超级电路”如图 5.11 所示。设计指标与前一个实例大致相同，负载仍然为 $2\text{k}\Omega$ ，只是要求功耗下降 50% 表 5.2 是具体的设计指标和综合结果，所综合出的低功耗运算放大器如图 5.13 所示 ($y_1 = y_4 = y_{10} = y_{15} = 1$ ，别的 y_i 为 0)。图 5.12 和图 5.13 的共同点是采用了折叠式共射共基输入级(当然，偏置电流是不一样的)，它们的差异主要在输出级。倘若是片内电路，且负载很轻，则可以去掉图 5.13 中的输出级，从而使功耗低于 38mW ，并进一步减小芯片面积。

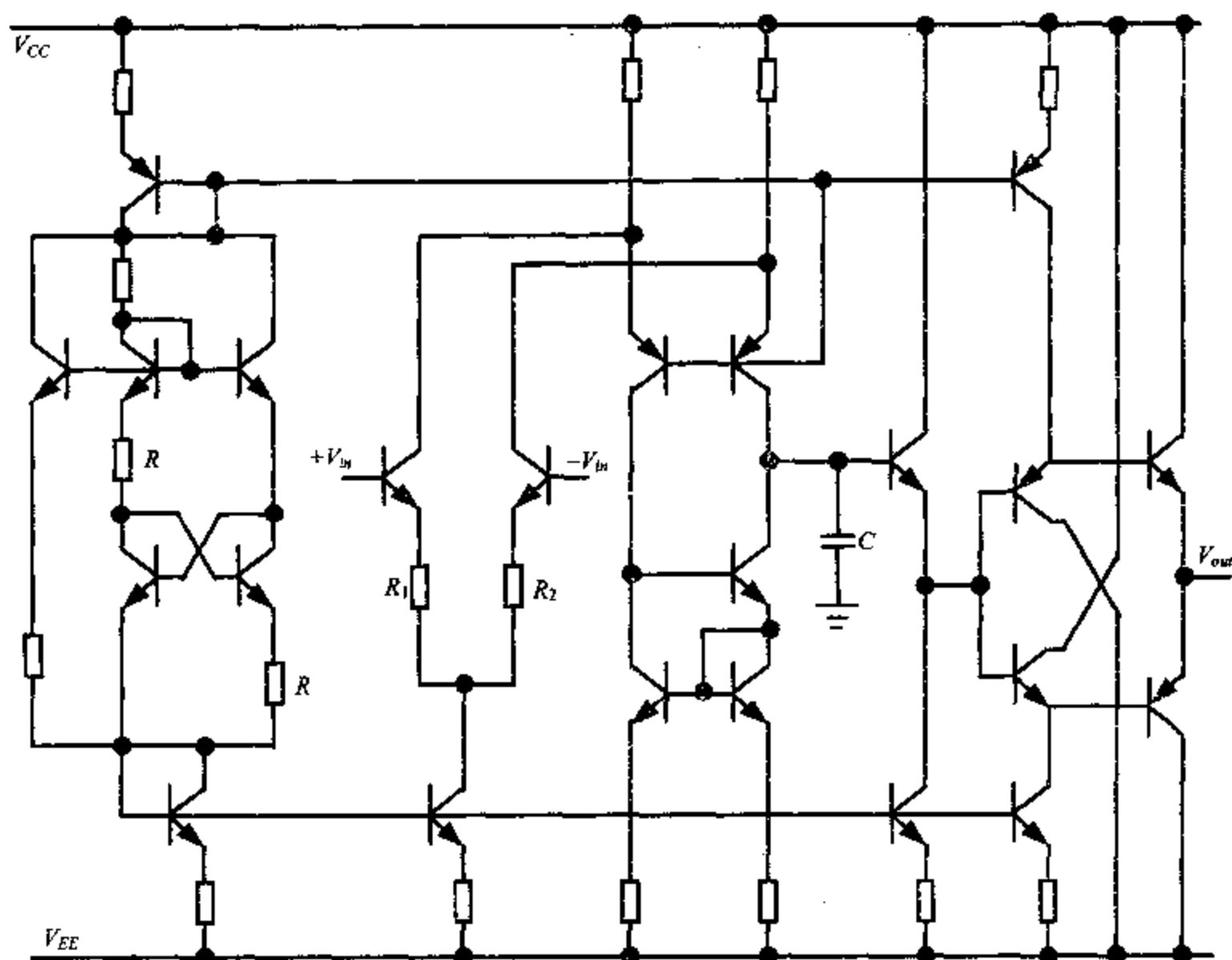


图 5.12 宽带运算放大器的电路结构

表 5.2 低功耗运算放大器的综合结果

	低频增益	单位增益带宽	相位余量	摆 率	等效噪声电压	功 耗
设计指标	$>50\text{dB}$	$>300\text{MHz}$	$>45^\circ$	$>120\text{ V}/\mu\text{s}$	$<12\text{ E}-9\text{ V}/\sqrt{\text{Hz}}$	$<100\text{mW}$
综合结果	54dB	400MHz	55°	$128\text{ V}/\mu\text{s}$	$9.0\text{ E}-9\text{ V}/\sqrt{\text{Hz}}$	70mW

例 5.3 双极型共射类放大器的“超级电路”如图 5.14 所示。

这是一个比较简单的“超级电路”，仅包含 4 个二进制变量，当 $y_1 = y_2 = 0$ 时，该“超级电路”简化为共射极反向放大器；当 $y_1 = 1, y_2 = 0$ 时，该“超级电路”简化为简单共射共基电路；当 $y_1 = y_2 = 1$ 时，该“超级电路”对应于一个带自举(Bootstrap)电容的共射共基电路， C_B 的作用是提升共射共基电路在高频时的输出阻抗 R_o 以进一步展宽放大器的频带^[79]，当然，共射极反向放大器是不需要自举电容 C_B 的。该“超级电路”中的射

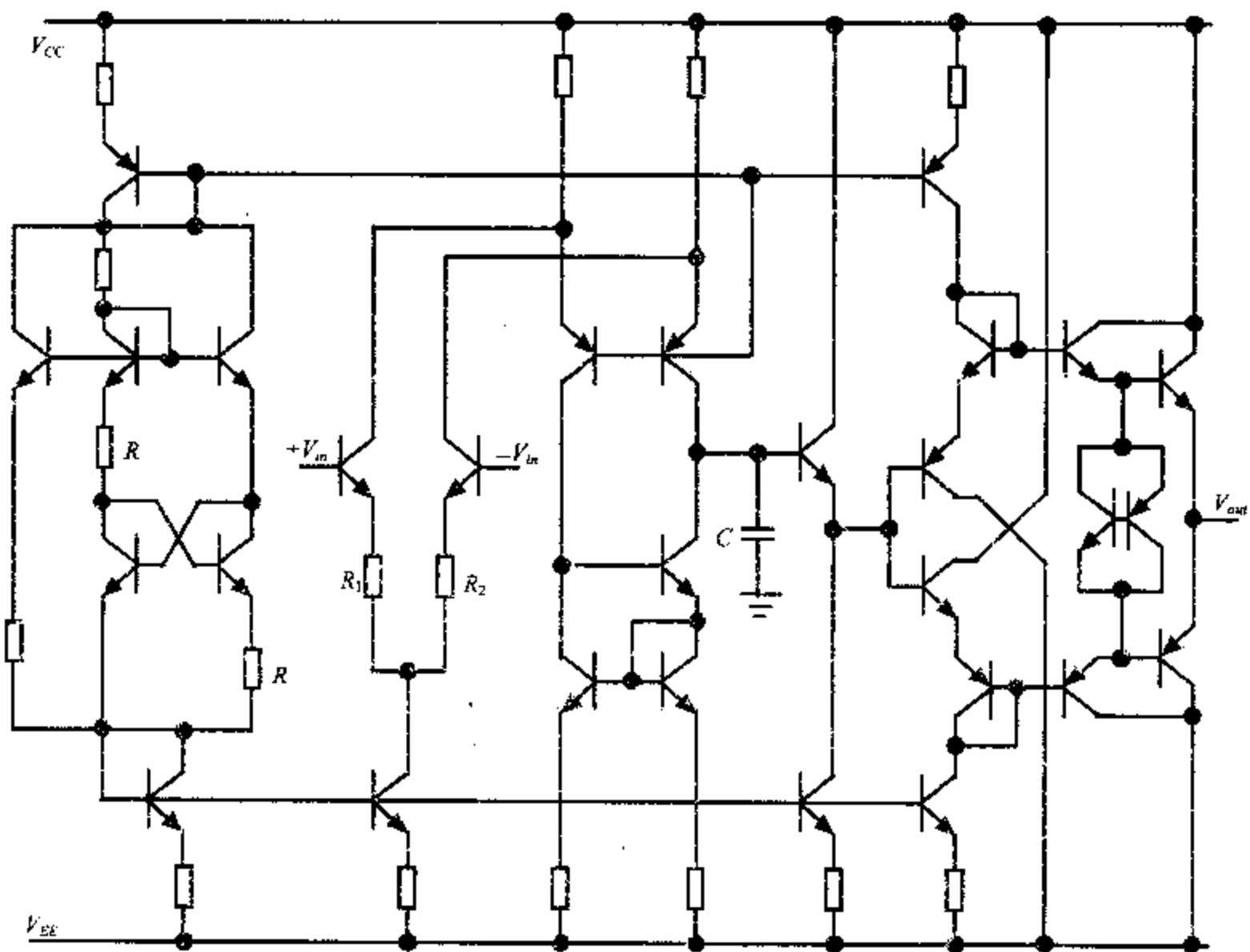


图 5.13 低功耗运算放大器的电路结构

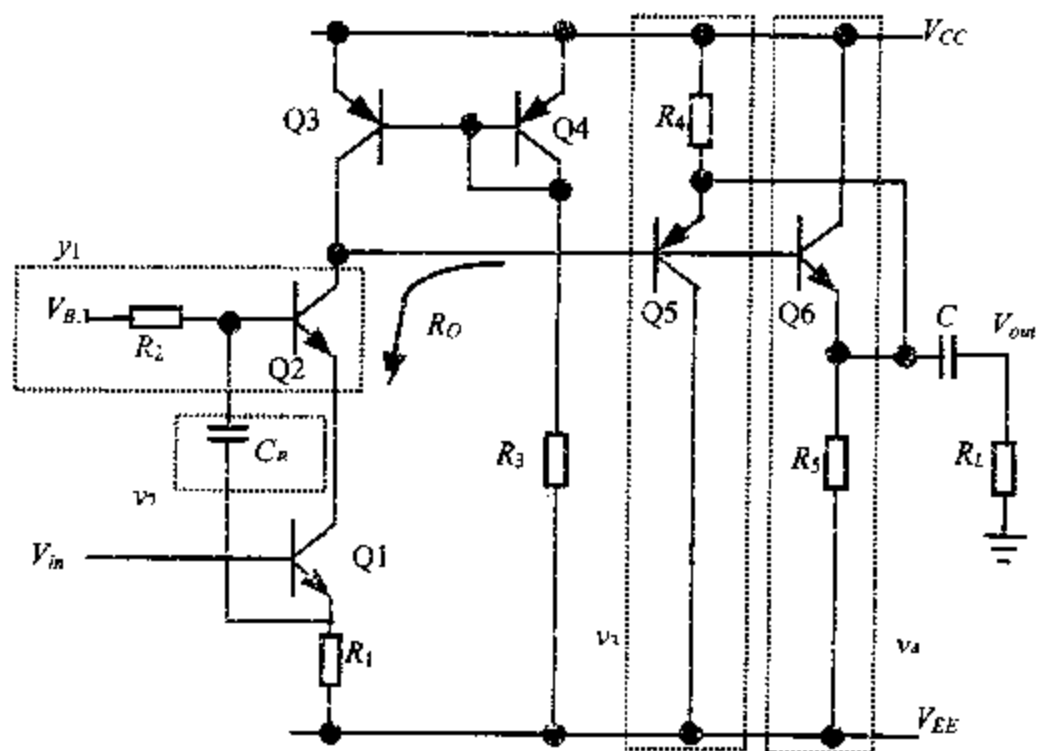


图 5.14 双极型共射类放大器的“超级电路”

极跟随器的作用是模拟放大器在拉电流 ($y_3=1, y_4=0$) 或灌电流 ($y_3=0, y_4=1$) 状态下的工作情况, 当 $y_3=y_4=0$ 时, 则为没有射极跟随器的情形, 当然, 不能同时接这两个射极跟随器。因此, 该“超级电路”中二进制变量应满足的约束条件为:

$$y_3 \text{ AND } y_4 = 0$$

$$\text{IF } y_2 = 1, \text{ THEN } y_1 = 1$$

由此可见, 该“超级电路”含 9 种有效简化形式。

我们假设双极型晶体管满足该条件 $g_m \gg g_\pi \gg g_o$, 采用本书第三章介绍的符号分析法, 则可以得到该“超级电路”的输出电阻 R_o 的近似表达式:

$$\begin{cases} G_O^C = \frac{g_{o,2}g_{\pi,2}(g_2(g_{m,1} + g_1) + y_2sC_B(g_1 + g_{\pi,1}))}{G_{m,2}(G_{m,1} + g_1)(g_2 + y_2sC_B)} \\ G_i^P = \frac{sC(g_L + g_4) + g_Lg_4}{(sC(g_L + g_4) + g_Lg_4 + g_{m,5}(sC + g_L))R_{B,5}} \\ G_i^N = \frac{sC(g_L + g_5) + g_Lg_5}{(sC(g_L + g_5) + g_Lg_5 + g_{m,6}(sC + g_L))R_{B,6}} \\ R_o = \frac{1}{g_{o,3} + (1 - y_1)g_{o,1}g_1/(g_{o,1} + g_1) + y_1G_O^C + y_3G_i^P + y_4G_i^N} \end{cases} \quad (5.18)$$

其中, $g_{m,i}$, $g_{\pi,i}$ 和 $g_{o,i}$ 分别表示第 i 个双极型晶体管的 g_m , g_π 和 g_o , $R_{B,i}$ 表示第 i 个双极型晶体管的基区电阻, g_i 是电阻 R_i 的电导, g_L 表示负载 R_L 的电导。而电压增益 A_v 则可以表示为:

$$A_v = g_{m,1} \times R_o \times A_{\text{buffer}} \quad (5.19)$$

其中, A_{buffer} 是输出级的电压增益, 即:

$$A_{\text{buffer}} = \frac{y_3(1 + \beta_5)R_4}{R_{B,5} + (1 + \beta_5)R_4} + \frac{y_4(1 + \beta_6)R_5}{R_{B,6} + (1 + \beta_6)R_5} + 1 - y_3 - y_4 \quad (5.20)$$

其中, β_i 是第 i 个双极型晶体管的正向电流放大倍数。

这样, 我们就通过式(5.18)、(5.19)和(5.20)将双极型共射类放大器的“超级电路”的电性能 A_v 表示成了电路结构($y_1 \sim y_4$)和器件尺寸(g_i , $g_{m,i}$, $g_{\pi,i}$ 和 $g_{o,i}$ 等)的解析函数。类似地, 我们还可以导出该“超级电路”的功耗、低频增益和 3dB 带宽等电性能的解析表达式, 为节省篇幅, 就不一一枚举了。

如果我们以低频增益、3dB 带宽等作为双极型共射类放大器综合问题的性能约束, 则可以很容易建立起形如(5.2)所示的数学描述, 然后就可以利用前面介绍的方法, 根据电路设计者所需的电性能指标(该性能指标也可能是在结构级综合时所提出的), 综合出所需的放大器。

表 5.3 是关于共射类放大器单元综合的 3 个实例, 从该表可以看出: 所有性能指标均满足要求, 而且, 当设计指标比较低时, 单元级综合方法能自动选择简单的单元电路, 综合结果令人满意。

表 5.3 共射类放大器单元综合实例

次 数	低频增益 A_v (dB)		3dB 带宽(MHz)		电路结构
	设计指标	综合结果	设计指标	综合结果	
1	30	30	1.0	2.0	$y_1 = y_3 = 1, y_2 = y_4 = 0$
2	30	35	10.0	15.4	$y_1 = y_2 = y_4 = 1, y_3 = 0$
3	15	20	0.1	0.2	$y_1 = y_3 = y_2 = y_4 = 0$

例 5.4 两级 CMOS 运算放大器的“超级电路”如图 5.15 所示, 该“超级电路”是

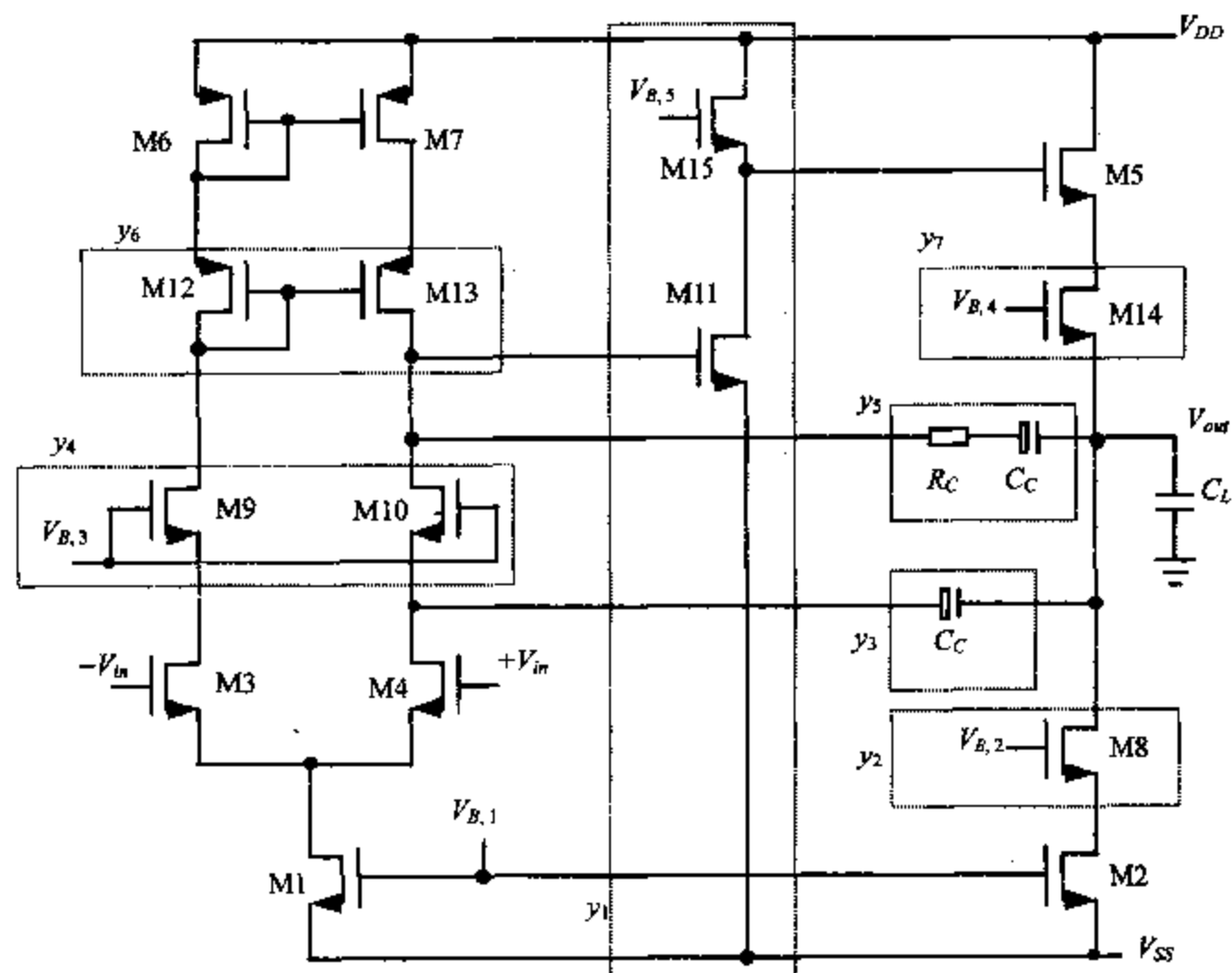


图 5.15 两级 CMOS 运算放大器的“超级电路”

由 P. C. Maulik 首先给出的, P. C. Maulik 还导出了该“超级电路”的低频增益(A_0)、单位增益带宽(f_u)、相位余量(p_m)、压摆率(S_R)、共模信号抑制比(CMRR)和电源信号抑制比(PSRR)等电性能的解析表达式,相应的混整型非线性数学规划问题的具体形式请参见文献[58,59],但是,在具体求解该混整型非线性数学规划问题时不采用 NPSOL 优化包和分支界定法,采用多分子模拟退火法以同时进行拓扑选择和器件尺寸优化^[91]。

表 5.4 对比了在相同设计指标下 P. C. Maulik 的综合结果^[59]和多分子模拟退火法综合结果^[91],由于 P. C. Maulik 的求解方法不具备全局收敛性,所综合出的运放的增益不仅没有达到用户的要求,而且芯片面积也比较大,因此,其综合结果不及本书所介绍的方法。从表 5.4 不难看出:多分子模拟法能根据不同的设计指标综合出切实的、面积尽可能小的单元电路。表 5.5 给出了关于两级 CMOS 运放的 4 个综合实例,这 4 个实例之间仅有设计指标 A_0 不同。

表 5.4 两级 CMOS 运算放大器综合结果对照表

	设计指标	综 合 结 果	
		P. C. Maulik 的方法	多分子模拟退火法
$A_0(\text{dB})$	≥ 66	63	66
$f_u(\text{MHz})$	≥ 5	10	7.4
$S_R(\text{V}/\mu\text{s})$	≥ 10	13	10
p_m	$\geq 45^\circ$	58°	50°

	设计指标	综 合 结 果	
		P. C. Maulik 的方法	多分子模拟退火法
$PSRR$ (dB)	≤ -60	-69	-63
$CMRR$ (dB)	≤ -65	---	-68
面积 (μm^2)	---	4 357	3 920

表 5.5 设计指标中只有低频增益不同时的综合结果对照表

低频增益 A_0 (dB)		面 积 (μm^2)	两级 CMOS 运算放大器的拓扑结构
指 标	综合结果		
65	71	3 972	$y_4=y_5=1, y_1=y_2=y_3=y_6=y_7=0$
80	80	4 085	$y_4=y_5=y_7=1, y_1=y_2=y_3=y_6=0$
95	102	4 415	$y_2=y_4=y_5=y_6=y_7=1, y_1=y_3=0$
115	118	4 905	$y_1=y_2=y_4=y_5=y_6=y_7=1, y_3=0$
其他指标	$f_u \geq 5\text{MHz}, p_m \geq 45^\circ, S_R \geq 2\text{V}/\mu\text{s}, CMRR \leq -40\text{dB}, PSRR \leq -40\text{dB}$		

例 5.5 考虑带恒流源负载和输出缓冲级的 MOS 放大器，其“超级电路”参见图 5.16。当 $y_1=1$ 时，该“超级电路”对应于共源共栅放大器，当 $y_1=0$ 时就简化为简单共源放大器；当 $y_2=1$ 时是带输出缓冲级的情形，当 $y_2=0$ 时则代表没有输出缓冲级的情形；当 $y_3=1$ 时是带补偿的恒流源负载，当 $y_3=0$ 时是简单恒流源负载。用符号分析法很容易获得该“超级电路”在 V_1 处的等效负载电阻 $R_{O,1}$ ：

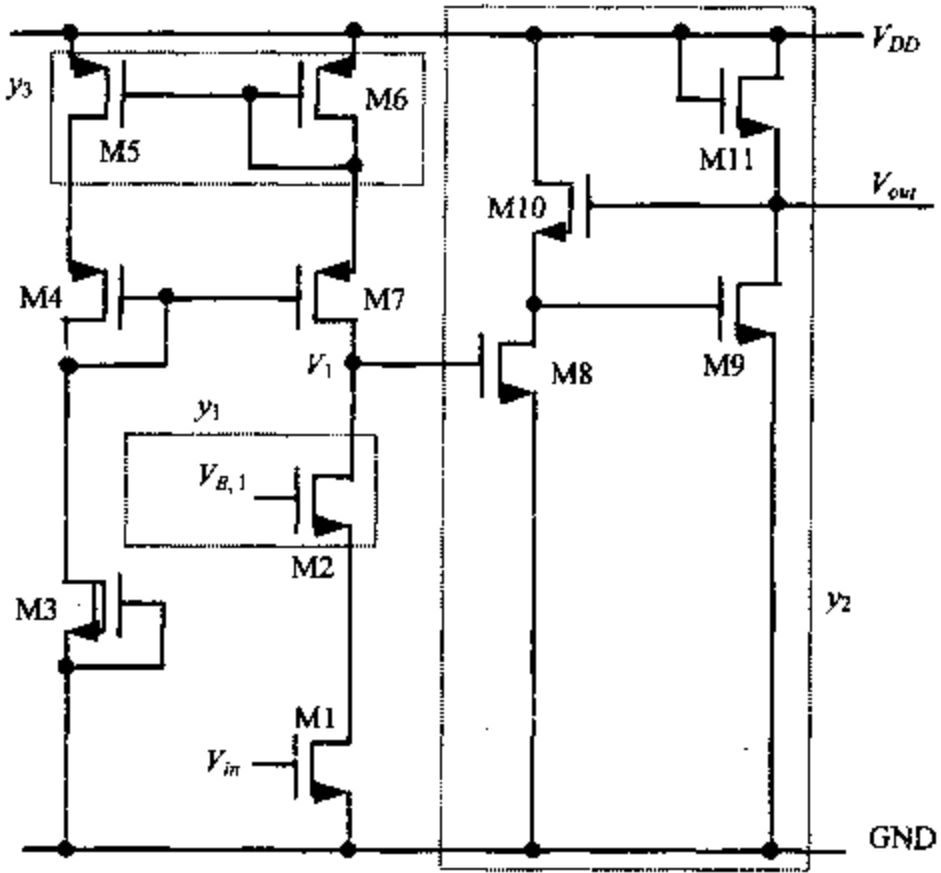


图 5.16 带恒流源负载和输出缓冲级的 CMOS 放大器的“超级电路”

$$R_{O,1} = \frac{1 - y_1}{y_2 g_{i0} + g_o + g_{d,1}} + \frac{y_1 (g_{d,1} + g_{d,2} + g_{m,2})}{g_{d,1} g_{d,2} + (g_o + y_2 g_{i0}) (g_{d,1} + g_{d,2} + g_{m,2})} \quad (5.21)$$

其中, $g_{m,i}$ 是第 i 个 MOS 场效应晶体管的跨导, $g_{d,i}$ 是第 i 个 MOSFET 的源漏极体电导, g_{i0} 是由 M8~M11 所构成的输出级的输入电导, g_o 是由 M3~M7 所构成的恒流源的输出电导, 在 M4 和 M7 具有相同器件尺寸的条件下, g_o 可以表示为:

$$g_o = \frac{y_3 g_{m,6} g_{d,7} (g_{d,3} + g_{d,5})}{g_{m,5} g_{m,7}} + (1 - y_3) g_{d,7} \quad (5.22)$$

则该“超级电路”的低频增益 A_0 和输出电阻 R_o 可以表示为:

$$\begin{cases} A_0 = g_{m,1} R_{O,1} \times \frac{g_{m,9} g_{m,8}}{g_{m,10} (g_{m,11} + g_{m,9})} \\ R_o = \frac{y_2}{g_{m,11} + g_{m,9}} + \frac{(1 - y_1)(1 - y_2)}{g_o + g_{d,1}} + \frac{y_1(1 - y_2)(g_{d,1} + g_{d,2} + g_{m,2})}{g_{d,1} g_{d,2} + g_o (g_{d,1} + g_{d,2} + g_{m,2})} \end{cases} \quad (5.23)$$

类似地, 我们还可以写出该“超级电路”的 3dB 带宽和功耗的电性能的解析表达式, 从而建立起相应的混整型非线性数学规划问题。

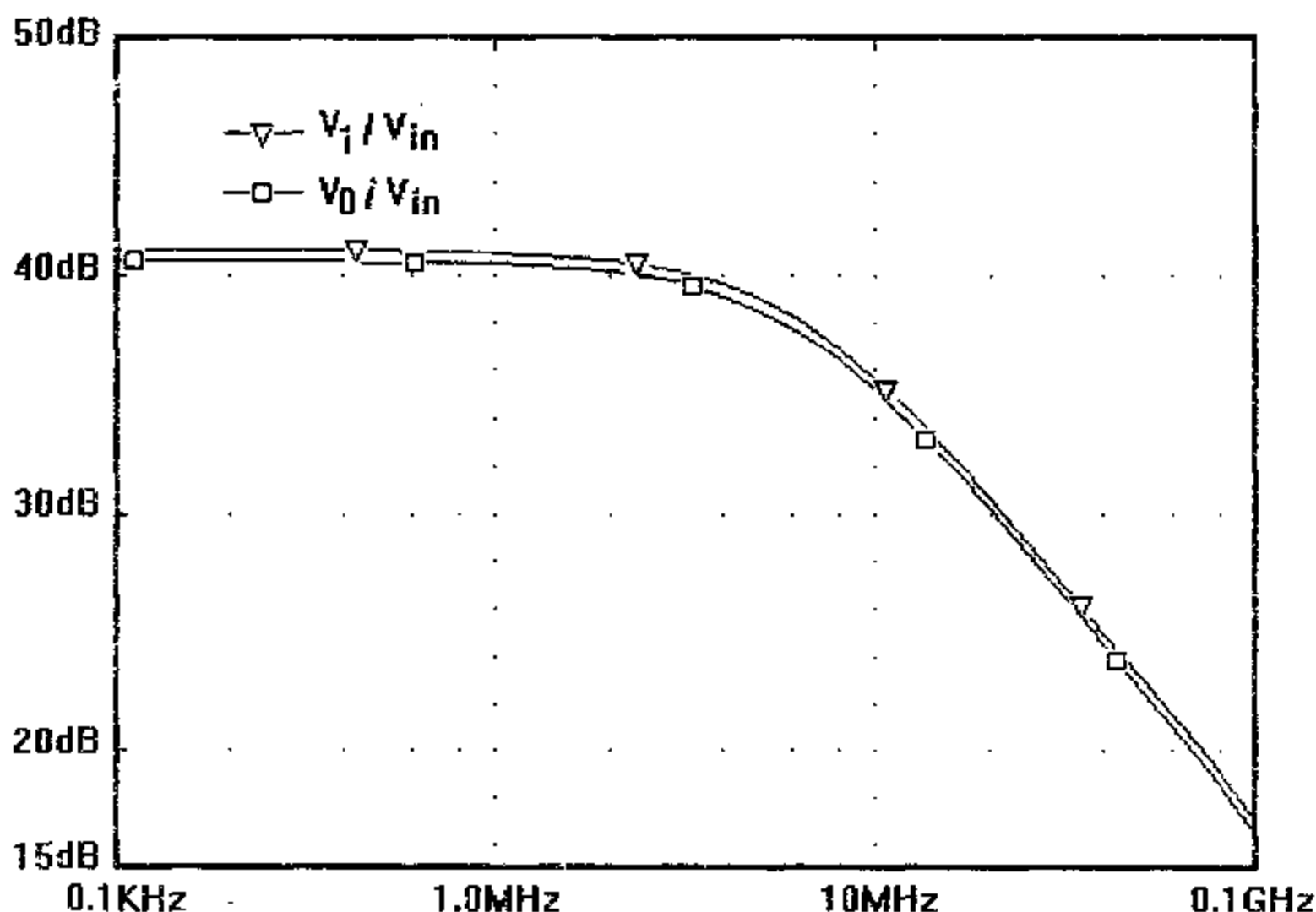


图 5.17 CMOS 放大器(综合结果)的幅频特性曲线

作为综合实例, 我们要求该放大器的低频增益 $A_0 \geq 40\text{dB}$, 3dB 带宽 $f_{BW} \geq 5\text{MHz}$, 为保证该放大器有一定的带负载能力, 还要求其输出电阻 $R_o \leq 10\text{k}\Omega$ 。综合结束后电路结构为 $y_1 = 0, y_2 = y_3 = 1$, 即: 带补偿型恒流源负载和输出缓冲级的简单共源放大器就能达到我们所要求的指标。

用电路模拟器 SPICE 对所综合出的 MOS 放大器的仿真结果分别如图 5.17 和图 5.18 所示, 其中, 图 5.17 是幅频特性, 图 5.18 是输出电阻 R_o 随频率变化的特性。由此

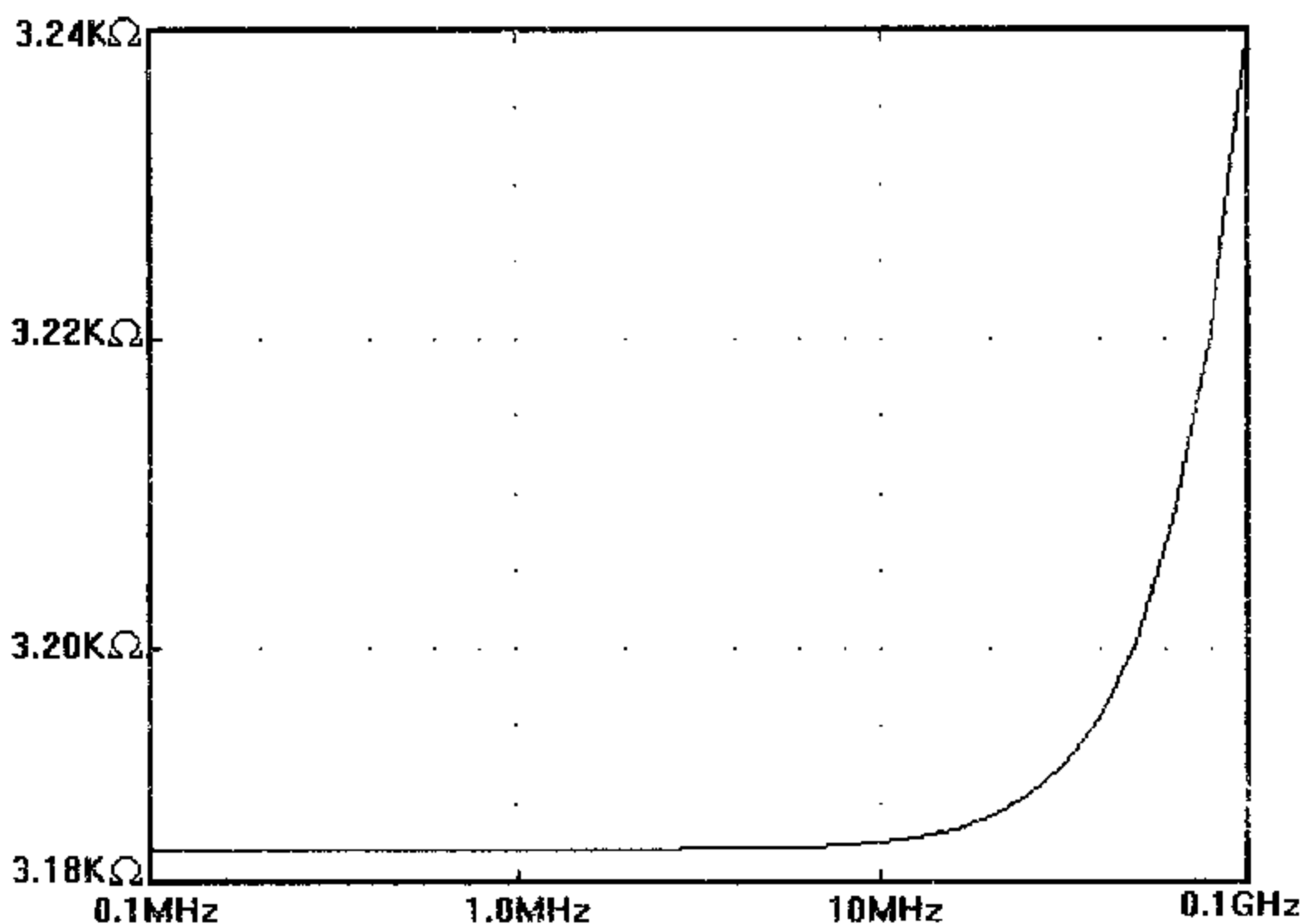


图 5.18 CMOS 放大器(综合结果)的输出电阻随频率变化的特性曲线

可以看出：低频增益 $A_0 \approx 41\text{dB} > 40\text{dB}$ ，3dB 带宽 $f_{BW} \approx 8\text{MHz} > 5\text{MHz}$ ，输出电阻 $R_o < 3.2\text{k}\Omega < 10\text{k}\Omega$ ，它们均满足设计指标的要求。

5.6 小 结

在模拟集成电路单元的自动综合方面，先拓扑选择后器件尺寸优化的“两步模式”有其固有的缺陷——综合过程需要在拓扑选择和器件尺寸优化这两个阶段反复回溯，而同时确定电路拓扑和器件尺寸的“一步模式”能有效克服“两步模式”的固有缺陷。本章主要讨论了实现“一步模式”的两个主要环节——“超级电路”模型的建立和模拟集成电路单元综合问题(5.2)的求解策略。考虑到“超级电路”的建立是实现同时确定电路结构和优化器件尺寸的首要环节，而高性能运算放大器又是模拟集成电路的核心单元，因此，本章以高性能运算放大器的自动综合问题为例，详细讨论了运算放大器的层次式分解策略；高性能运算放大器的偏置电路、输入级、中间级和输出级等基本模块，在这些基本模块中，有的适于宽带运放，有的适于低功耗运放，有的适于高速运放，而有的则适于低失真运放；在此基础上，本章全面地建立高性能双极型运算放大器的“超级电路”的各个主要环节——约束方程 $\Phi_i(Y) \leq 0$ 的建立过程、用解析函数表示“超级电路”电性能的方法。

本章给出了宽带运放和低功耗运放等两个综合实例，由于采用了较好的电路结构，所综合出的运算放大器不仅频带宽、功耗低，而且在工作速度(摆率)、抗电源波动和非线性失真等方面的性能也是相当不错的。这些实例表明图 5.11 所示的“超级电路”是恰

当的。本章还给出了几个模拟集成电路单元的综合实例，这些综合实例表明同时确定拓扑结构和优化器件尺寸的“一步模式”明显优于“两步模式”，是一种行之有效的模拟集成电路单元级综合方法；这些实例同时也展示出分子模拟退火法所具有的全局收敛性。

本章所建立的适合于高性能运算放大器的“超级电路”对设计高性能运放或模拟电子系统都非常有价值，虽然该“超级电路”目前只涉及双极工艺，但其中的方法和部分电路技巧同样适合于 CMOS 工艺。

还应当看到，构造一个“超级电路”来代表所有的模拟集成电路单元是不现实的，因此应当为每一类型的模拟集成电路单元构造一个“超级电路”。一个完整的模拟集成电路设计自动化系统中应当包含多少个“超级电路”主要受结构级综合的影响，结构级综合需要用多少种功能块来表示电路行为，则单元级综合就应当包含多少个“超级电路”单元。不言而喻，模拟集成电路自动化设计系统中所包含的“超级电路”越多，不仅其自身的适应面越广，而且也更有利于结构级综合，这也是我们在第四章讨论结构级综合方法强调只用较少的单元电路进行结构综合的原因。对于第四章中所讨论的结构级综合方法而言，只要建立乘法器、积分器、运算放大器、加法器、跨导放大器、偏置电路等的“超级电路”即可。但是，考虑到模拟电子系统和混合信号系统的发展，进一步建立 A/D, D/A, PWM 和 PLL 等单元的“超级电路”也是非常必要的。

第六章 模拟集成电路的版图综合

由于微电子技术的迅速发展,芯片的集成度越来越高,同时在一个芯片上集成模拟电路和数字电路已经成为专用集成电路的主要发展方向。标准单元、门阵、门海等结构化的半定制方法虽然成功地解决了数字集成电路的版图设计问题,但对于模拟集成电路和数模混合集成电路而言,这些方法却显得黔驴技穷,其根本原因在于模拟集成电路具有拓扑结构繁多、器件尺寸参差不齐、版图千差万别的特点,而基于固定版图库的方法不具备处理不规则版图的能力。因此,数模混合集成电路中的模拟集成电路部分的版图大都是按全定制模式设计出来的,因而需要设计者有丰富的经验,并需花费很多时间。采用模拟集成电路版图综合工具是减少设计时间最有效的途径。

模拟集成电路的版图设计有两个目标:使芯片面积最小和将寄生器件对电路性能的影响降至最低幅度。前者是为了降低芯片的成本,后者是保证所设计出的版图和原电路具有相同的功能。在手工设计版图时,专家们通过反复的几何尺寸优化和采用一些特有的版图设计经验(通常又称为模拟电路的约束)来实现这两个目标。本章将以器件级的版图布局、布线方法为中心,详细介绍如何将几何尺寸优化和版图设计经验结合起来,以形成模拟集成电路的版图综合系统,并使之具备模拟集成电路版图设计专家们的水平。

目前,模拟集成电路的版图设计大致可以分为四类:模拟阵列、参数化模拟单元、模拟标准单元和模拟宏单元。从它们所生成的版图质量来看,只有模拟宏单元可以和模拟集成电路版图设计专家们相比,因此,本章重点介绍模拟宏单元法,介绍如何将专家们的智慧融会到版图综合系统中,使之具有设计时间短、版图质量高的特点。当然,在介绍版图综合方法之前,我们将首先介绍模拟集成电路版图设计应当重点考虑的因素,以便读者能更好地理解本章所介绍的布局、布线策略。

6.1 模拟集成电路版图设计的特征

在讨论模拟集成电路版图综合方法之前,了解版图对模拟电路性能的影响是十分必要的。由于模拟电路的功能和性能指标强烈地依赖于电路中每一个元件参数,而版图中不可避免的寄生参数将使元件参数偏离其设计值,从而影响电路性能,因此版图寄生参数、信号间的耦合效应和器件失配效应是影响模拟集成电路性能三类主要因素,是版图综合应当特别注意的问题。

6.1.1 版图寄生参数

在电路原理图中,器件之间的连线被理想化了——连线电阻和电容都视为零。然而,实际情况并非如此,用于连线的每一个导体都会引入电阻和电容,通常称之为寄生电阻和寄生电容,它们将影响电路性能。由于这类寄生电阻和寄生电容分别与导体的长宽比和面积成正比,因此,那些对保证电路性能起关键作用的连线应当尽可能短,从而要求

将相应的被联接的器件放置得越近越好。

除连线之外,器件的几何形状也会引入寄生电阻和寄生电容。在MOS电路中,这类寄生参数主要由栅极的形状决定。由于栅极的面积是由电路设计所确定的,因此版图综合阶段不可能减小器件的栅电容,但是,却可以通过调整器件的几何形状以减小另一些寄生电容,例如pn结电容。

MOS晶体管的源漏区pn结电容与pn结的面积和周长成正比、随pn结的反向偏置电压呈非线性变化的趋势,因而可以通过最小化有源区尺寸来减小MOS晶体管源漏区的寄生电容。特别地,采用折叠式结构可以使两个栅区共用一个源极或漏极,从而减小源极或漏极的寄生电容,参见图6.1。如果同一类型的两个MOS晶体管的源漏极是联接在一起的,可以通过器件合并的手段,让它们共享源漏极,这不仅可以减小寄生电容、节省互连线、减小源漏区的扩散体电阻,还特别有助于减小芯片面积、增加版图的器件密度。图6.2是两个器件共享源漏极的实例,从图6.2(b)可以看到,通过器件合并,消除了连线覆盖电容 C_{cover} ,源漏极pn结电容 C_{SB1} , C_{DB2} 和相应的体电阻都减小了一半。类似地,如果两个MOS晶体管的栅极是联接在一起的,则只要器件之间的间距允许,就可以采用墩式结构,省去不必要的金属连线以减少寄生电容、节省芯片面积。图6.3是墩式结构的实例。

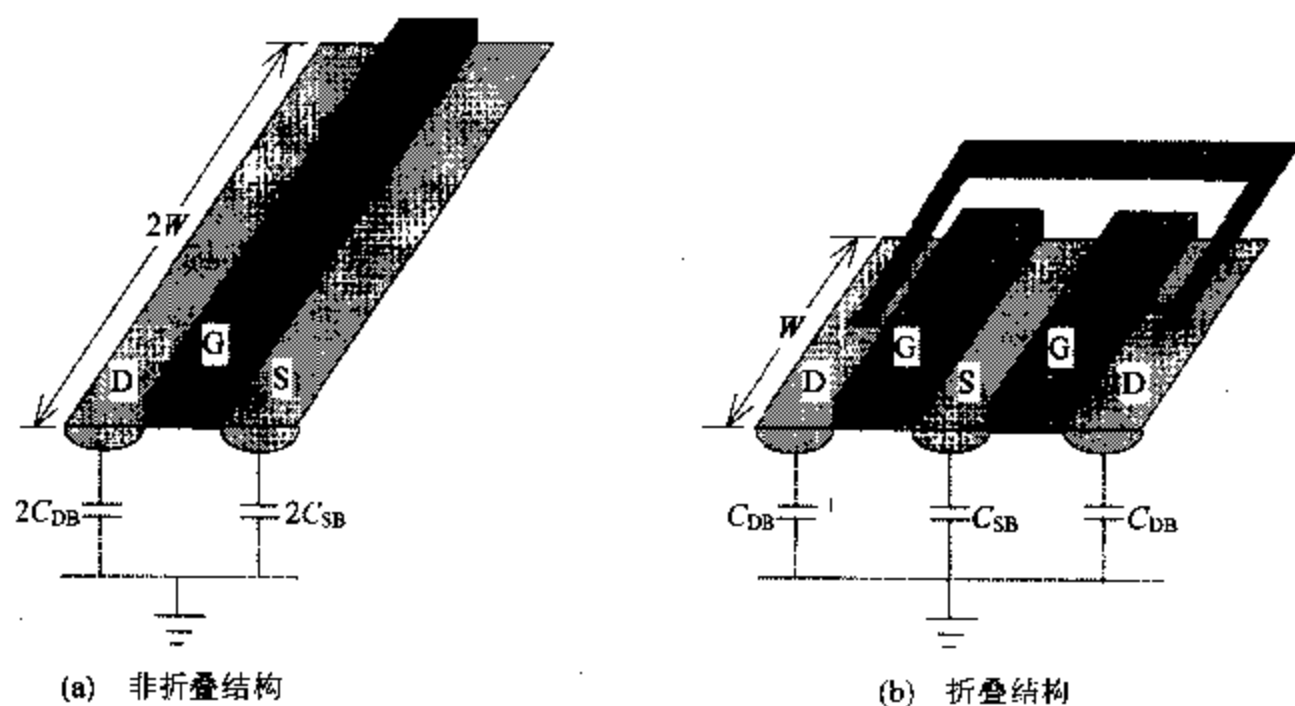


图 6.1 折叠式器件结构减小源极 pn 结电容的实例

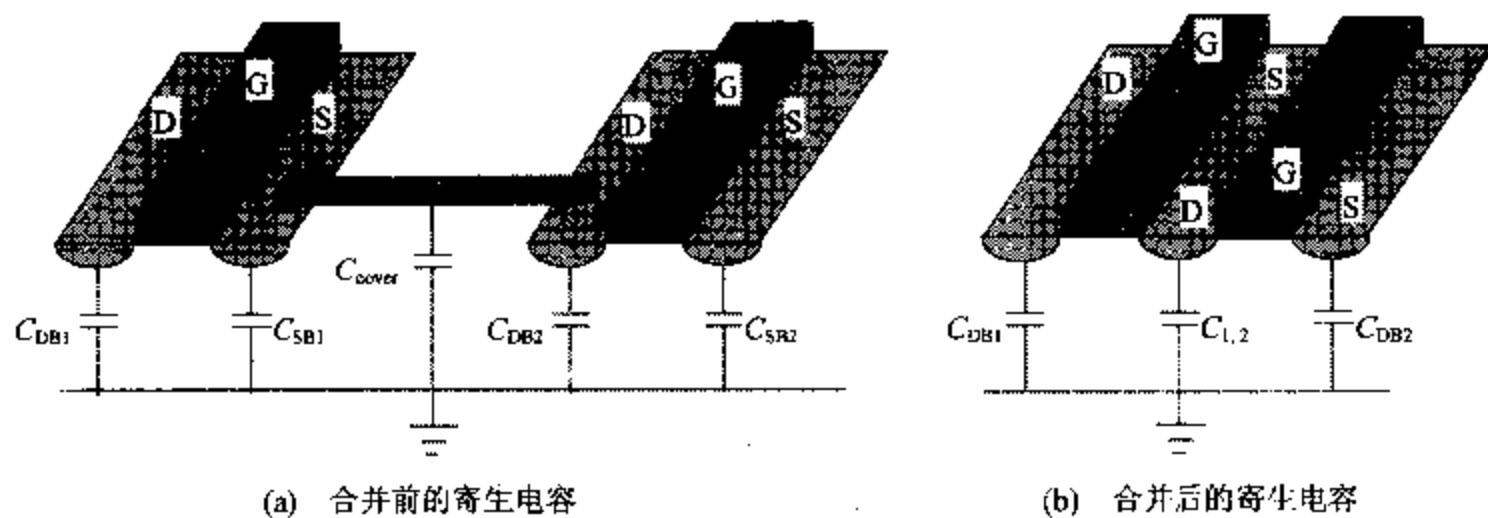


图 6.2 器件合并减少寄生电容的实例, 其中 $C_{1,2} = C_{\text{SB1}} = C_{\text{DB2}}$

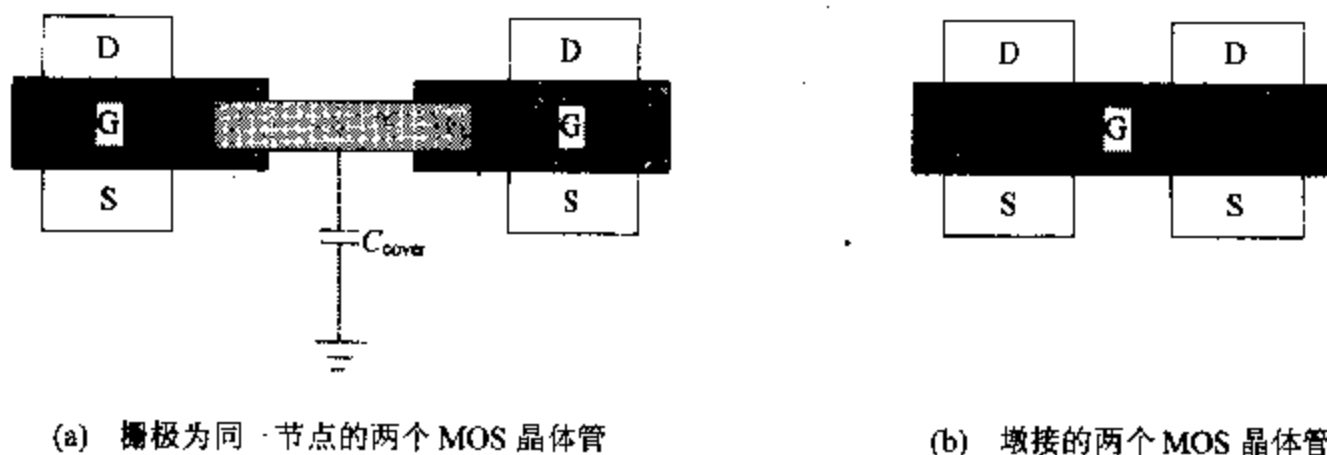


图 6.3 采用墩接减少寄生电容、连线电阻和芯片面积的实例

6.1.2 信号间的耦合效应

版图中除包含各种寄生参数之外,还存在各种信号间的耦合效应,通常又称之为串扰,主要来自于电阻、电容和热效应。耦合效应的出现将使电路的噪声增大,严重时还会通过反馈作用影响电路的稳定性。

信号线之间的电容是产生耦合效应的形式之一。两个导体在下述情况时会形成较大的电容:

- (1) 两导体位于不同层、且相互交叠,形成交叠电容;
- (2) 两导体位于同一层、且邻近或并行,形成平行电容。

图 6.4(a)是产生交叠电容和平行电容的实例。可以分别通过减小导体间的交叠面积和并行长度的方法来减小交叠电容和平行电容,从而达到减小耦合效应的目的。很显然,如果两个导体上的信号同步变化(同时升高、同时降低),那么,即使它们邻近、并行,串扰效应也是可以忽略的。因此,如果两个导体必须并行排列,则在它们之间放一条与地或固定参考电压相连的导体,将有效屏蔽它们之间的串扰效应。图 6.4(b)是用屏蔽导体减小串扰效应的例子。

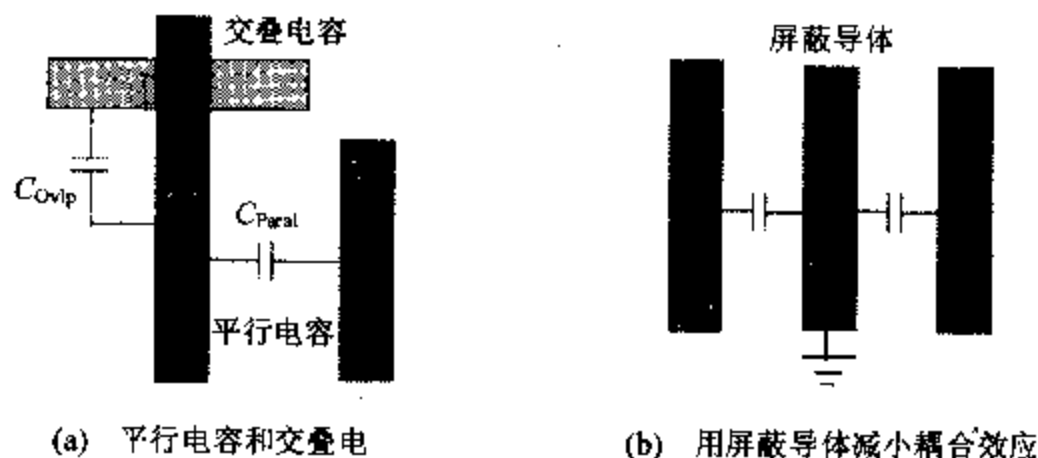


图 6.4 耦合效应及减小耦合效应的措施

寄生电阻也可能产生耦合效应。不论是多晶硅、扩散区形成的导体,还是金属(铝或铜)连线,其电阻值都不为零,当流过导体的电流出现波动时,该导体上的电压将会随之波动,这种波动的电压实质上就是噪声,它将耦合到与该导体相联接的所有器件。在模拟集成电路、数模混合集成电路中,电源线的寄生电阻耦合效应会对电路性能产生严重影响,因此,应当尽量缩短或适当加宽电源线以减小电源线的电阻。另外,将模拟电路部

分的电源线同数值部分分开,让电流较大的器件使用独立的电源线也是行之有效的措施。

随着集成电路特征尺寸的不断缩小和工作频率的迅速增加,衬底已成为信号耦合的主要载体,衬底噪声将通过寄生电容或寄生电阻耦合到电路中的所有节点,从而对电路性能产生严重的影响,图 6.5(a)是衬底噪声耦合的实例。可以采用保护环(通常是与地等电位、且电阻值比较低的扩散区)来降低衬底噪声的耦合效应,削弱衬底噪声对电路敏感部分的影响,图 6.5(b)是采用保护环减小衬底噪声耦合效应的实例。串扰效应在数模混合集成电路中尤为显著,因为,其数字部分的工作频率很高,开关切换时所产生的噪声会通过电源线、并行信号线和衬底耦合到模拟部分,从而严重影响模拟部分,乃至整个芯片的性能。

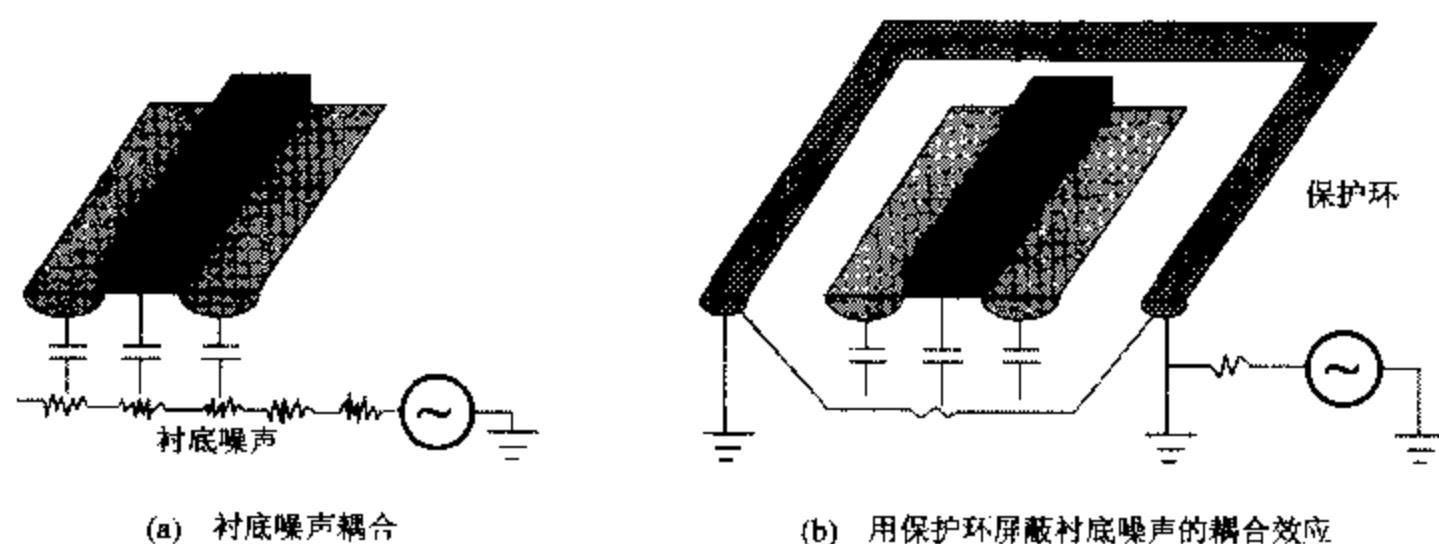


图 6.5 衬底噪声的耦合效应及其屏蔽技术

除上述电信号的耦合效应之外,热耦合效应在模拟集成电路中也很突出。由于硅的导热性能很好,器件的功耗发生变化时将导致局部温度的波动,通过热耦合效应影响那些对温度十分敏感的器件,有时还会形成热反馈环^[10],严重影响电路性能。热耦合效应在双极型工艺和 Bi-CMOS 工艺中尤为突出。由于功耗较大的器件是模拟集成电路中主要的噪声源,因此在版图布局时让温度敏感器件远离功耗较大的器件是减少热耦合效应的简便方法,让那些需要匹配的器件对称地分布在噪声源的周围也是减小热耦合效应的有效策略。

6.1.3 器件失配效应

在电路级综合阶段,模拟集成电路设计者常常会使用一些完全相同的器件(简称“匹配器件”),而版图中的寄生效应和各种工艺涨落因素将使这些匹配器件出现偏差,通常称之为器件失配。当器件失配较大时会对电路的失调等性能产生严重的影响。匹配器件的形状、走向和间距是产生器件失配效应的三类主要因素。

由于器件性能总是和器件的面积相关的,而光刻过程中不可避免的对准误差和刻蚀偏差将使器件的几何尺寸偏离版图的设计值,我们称之为几何失真。在版图设计时,使匹配器件具有相同的形状以保证其几何失真有较好的一致性可以减小器件失配效应。几何失真主要决定于集成电路工艺的特征尺寸,在电路性能和芯片面积允许的条件下增加匹配器件的面积也可以减小器件失配效应^[173]。

减小器件失配效应的另一策略是让匹配器件具有相同的走向，这是因为许多工艺过程(例如离子注入)将产生各向异性的几何失真。图 6.6 示意性地描绘了因走向不一致所产生的失配效应，由于离子注入时离子束与硅片不垂直，沿横向排列的 MOS 晶体管的沟道变短，但纵向排列的 MOS 晶体管的沟道却变宽。

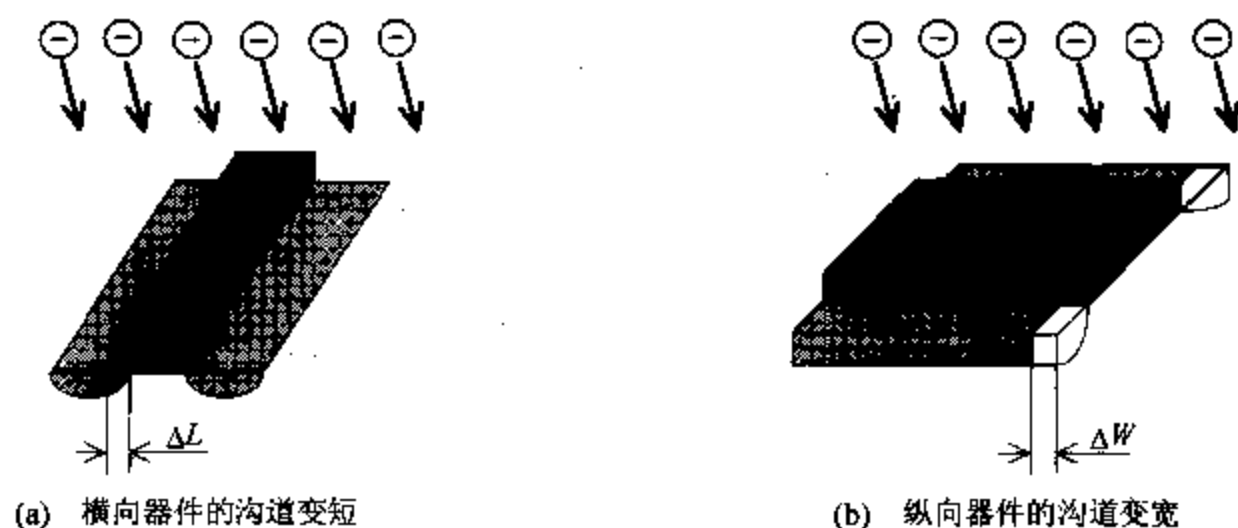


图 6.6 走向不一致产生器件失配效应的例子

迁移率、氧化层厚度、衬底杂质浓度、外延层的电导率等工艺参数在空间分布的不均匀性也会产生器件失配效应。图 6.7 是因氧化层厚度不均匀产生器件失配效应的例子。从图 6.7 不难看出，匹配器件之间的距离越大，失配效应就越严重。因此，在不违背几何规则的情况下，应当让匹配器件尽可能靠近。

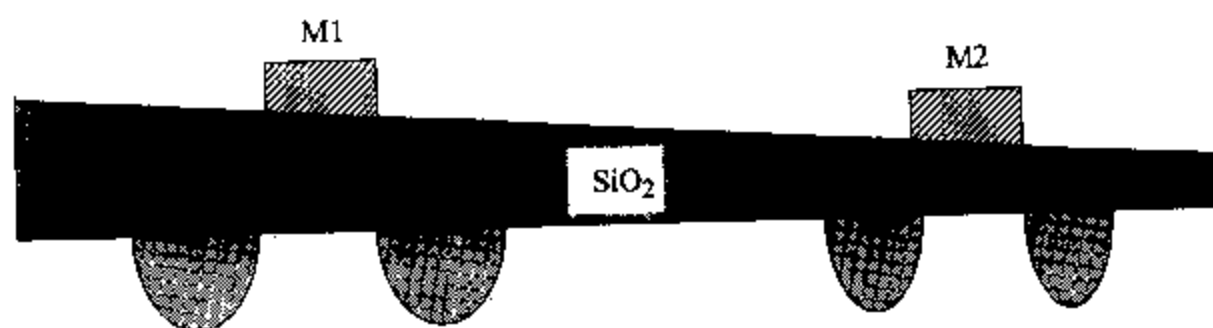
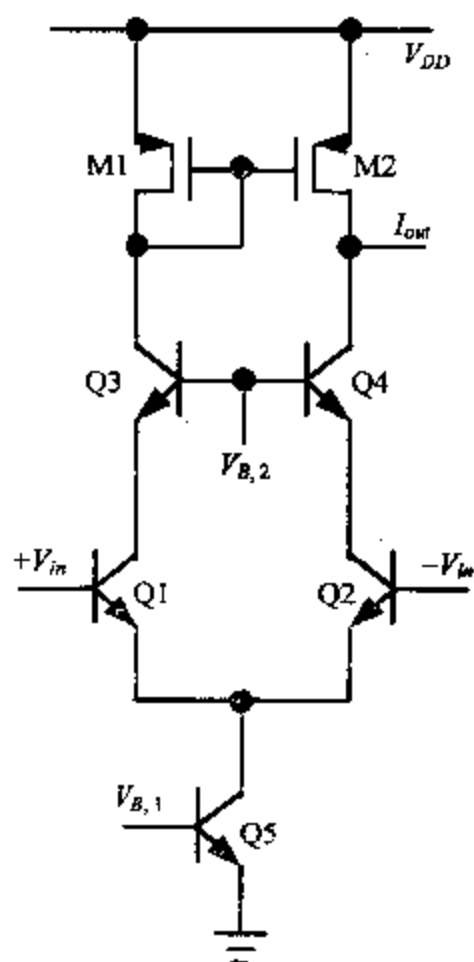


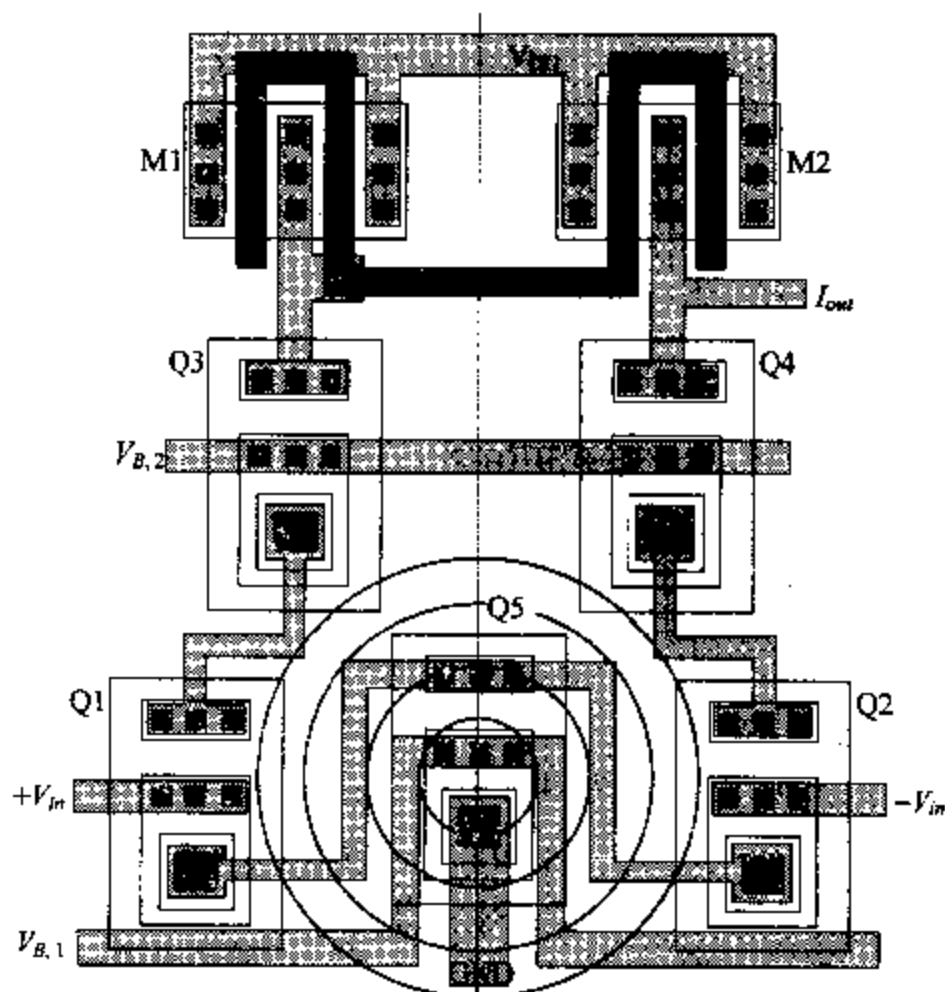
图 6.7 栅氧化层厚度不均匀产生器件失配的例子

除上述因素之外，电路中的热源(功耗较大的器件)、连线的寄生电阻和寄生电容等都可能产生器件失配效应。由于不可能将匹配器件放在同一位置，而电路中的热源和寄生参数又是无法去除的，因此，减小器件失配效应的唯一途径是将匹配器件对称地排列在版图中——将它们集中起来，使之具有相同的形状、相同的走向、对称地分布在热源周围，并尽可能使其连线也完全对称。

注重布局、布线的对称性是模拟集成电路版图综合过程中最重要、最艰巨的任务。例如：对于图 6.8(a)所示的跨导运算放大器，差分输入晶体管 Q1 和 Q2 是匹配器件，而恒流源负载中，M1 和 M2，Q3 和 Q4 也分别是匹配器件；由于提供偏置电流的晶体管 Q5 的功耗比较大，会形成一个热源，因此在版图综合时还应当让匹配器件对称地分布在 Q5 的周围。图 6.8(b)是 KOAN/ANAGRAM II^[6.69,70]所综合出的版图，从这里可以看出：匹配器件不仅形状相同、走向一致、全部对称地分布在器件 Q5 的周围，而且连线的对称性也非常好，从而确保了匹配器件的一致性，对减小电路的失调、降低噪声都大有裨益。



(a) 原理图



(b) 具有良好对称性的版图

图 6.8 BiMOS 跨导放大器及其版图

6.2 半定制模拟集成电路版图设计技术

虽然模拟集成电路的版图设计大都采用全定制方式，靠版图设计专家们手工完成，但是，为了提高模拟集成电路的设计自动化水平、降低设计成本、缩短数模混合型 ASIC 的上市时间，一些半定制模拟集成电路版图设计方法和系统正逐步走向成熟。本节将讨论模拟阵列、参数化模拟单元、模拟标准单元和模拟宏单元等四类半定制模拟集成电路版图设计技术。

6.2.1 模拟阵列

为了提高设计速度和降低制造成本，中小规模的模拟 ASIC 可以用模拟阵列来实现。集成电路制造商提供的模拟阵列包含若干双极型晶体管，采用模拟阵列设计版图时，设计者只需用两层或三层金属连线，按所需的拓扑结构，将阵列上已经制作好的晶体管联接起来即可。图 6.9 是用模拟门阵设计版图的示意图。IBM 已开发出一个模拟门阵的自动化设计系统^[174]，该系统借助数字集成电路的布局工具来分配阵列中的各晶体管，使之尽可能满足电路设计者的需要；并采用迷宫算法进行布线。由于模拟阵列中的双极型晶体管尺寸不能随意调整、数字集成电路布局布线工具又存在许多局限性，用模拟阵列所获得的版图将影响电路的性能。对于规模不大、性能和精度不太高的双极型模拟集成电路而言，模拟阵列的局限性是可以接受的。但模拟阵列不适合于 MOS 工艺，因为 MOS 模拟集成电路的设计与双极型模拟集成电路设计不同，很难用有限几种尺寸的

MOS 晶体管来设计 MOS 模拟集成电路。当然,和其他所有门阵策略一样,硅片利用率不高是模拟阵列的又一缺陷。

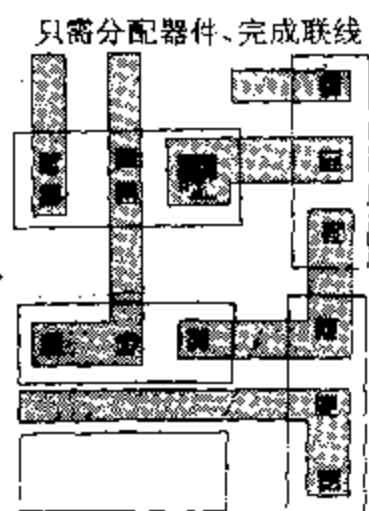


图 6.9 模拟阵列示意图

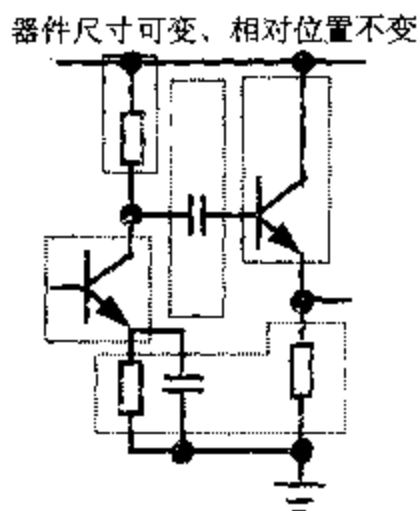


图 6.10 参数化模拟单元示意图

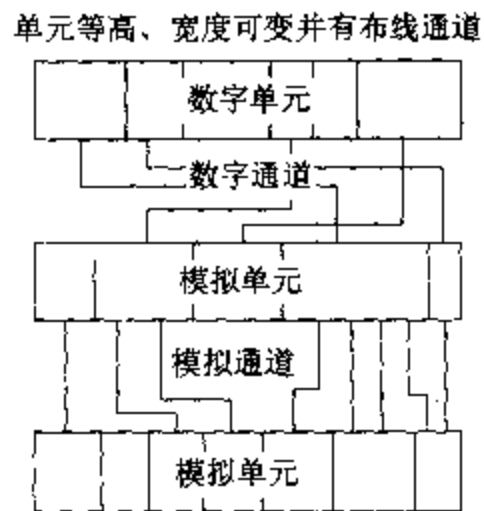


图 6.11 模拟标准单元示意图

6.2.2 参数化模拟单元

参数化模拟单元是最为成熟的模拟集成电路版图自动综合技术^[45,175,176],采用参数化模拟单元技术的版图综合系统为每一种单元提供若干个拓扑结构,而对于每一个具体的拓扑结构而言,系统都提供一个基本版图(或初始版图),并在保持各器件的相对位置不变的情况下,根据用户所要求的器件尺寸生成最终版图。图 6.10 是参数化模拟单元的示意图。由于这类版图综合系统中所有单元的拓扑结构是固定不变的,只有器件尺寸可变,因此,人们通常将这类单元称为参数化单元。拓扑固定这一特征使得参数化模拟单元系统中的基本版图可以手工优化,并能够采用一些简单的策略来估计版图寄生参数、处理器件合并和对称性布局布线,因而能生成质量很高的版图。除此之外,受工艺变化的影响较小是其另一优势。参数化模拟单元的主要缺陷是需要保存和维护许许多多的拓扑结构以适应各种需要,而且,每增加一个拓扑结构时,更新系统所花费的工作量很大。因此,用参数化模拟单元处理开关电容滤波器等只需少数几种单元的模拟集成电路综合问题是非常合适的。

6.2.3 模拟标准单元

模拟标准单元^[176]与参数化单元不一样,这类系统不自动生成器件级的版图,而是使用一些预先定义的等高单元,习惯上称为标准单元。在通常情况下,标准单元按行规整地排列,行间留有布线通道以便根据用户的需要将标准单元有机地联接起来,图 6.11 示意性地描绘了模拟标准单元的特征。虽然可以沿袭数字电路的通道布线法,但有限的标准单元却很难适应拓扑结构灵活多变、器件尺寸参差不齐的模拟集成电路。模拟标准单元还存在另一缺陷——通道布线法不能保证匹配器件的位置、寄生参数和连线等的对称性。当然,模拟标准单元可以将模拟信号线和数字信号线分布在不同的布线通道中,从而有效地减小数模混合 ASIC 中的串扰效应^[177]。由于开关电容滤波器只需有限的几种单元,而防止开关噪声影响模拟部分的性能是版图设计的主要难题,因此 A. Barlowd 等人用这些策略取得了很好的效果^[179]。

6.2.4 模拟宏单元

模拟宏单元是一种更通用的器件级版图综合方法,是受数字宏单元法启发所形成的。用模拟宏单元生成版图的过程包括模块生成、器件布局和布线等三个环节,图 6.12 示意性地描绘了模拟宏单元的工作模式。

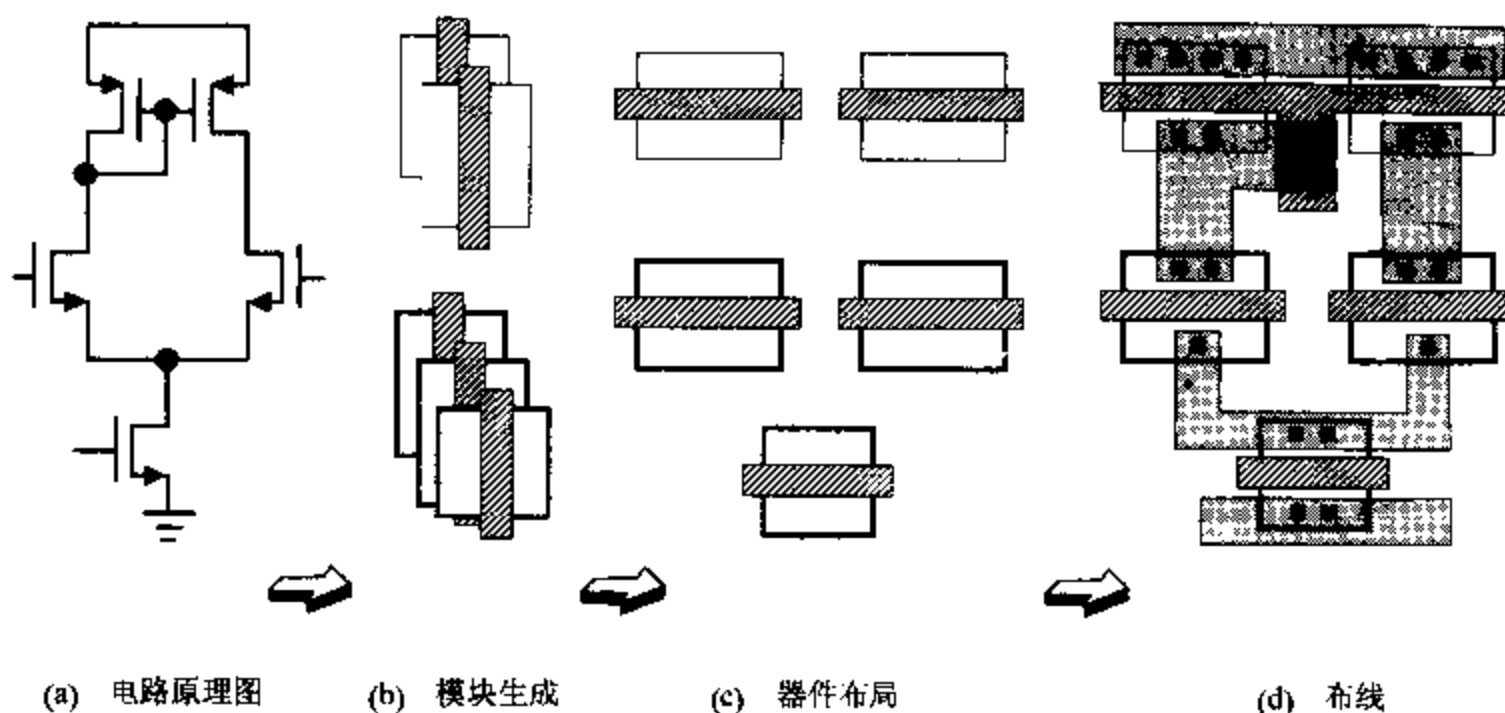


图 6.12 模拟宏单元的工作模式

1. 模块生成

模块生成的作用是根据器件(或器件组)的尺寸,为每一个器件(或器件组)产生适当的版图。在模拟宏单元法中,器件组主要包括差分对、镜象电流源、共源共栅结构等需要严格匹配的器件。为了增加后续器件布局的灵活性,当用户给定器件的沟道长度、沟道宽度之后,模块生成应当能够为该器件产生多个电性能相同但几何形状不同的版图,当然模块生成阶段还应当将器件放入 P 阱或 N 阱中,并完成阱与衬底的联接。在现有的许多模拟宏单元系统中,模块生成部分还包含一些特殊的优化功能,可以处理器件合并、寄生参数的匹配等问题^[5,67,175,178,180]。

2. 器件布局

器件布局的任务是将已经生成的各模块排列起来,使芯片面积最小、且器件之间的连线尽可能短,并支持对称布局和匹配器件的集中排列。器件布局采用模拟退火法^[85,108]来获得最佳布局,该算法采用递归方式将版图作二分切割,从而形成一个切片树,而每一个切片的位置都代表一个器件在版图中的位置。图 6.13 是切片式布局的示意图。该器件布局模式的优点在于可以非常容易地与通道布线法结合起来,因为可以在布局阶段生成一些用于连线的切片。但遗憾的是它不适合于器件级的模拟集成电路版图设计^[70],因为使用专门的布线通道既不适合于实现布线的对称性,也不能实现在器件上布线的要求,当电路中包含许多小尺寸器件时,这种切片结构的版图利用率极低。另外,该方法还不容易扩展到器件版图为非矩形形状的场所。我们在后面将会看到,平铺式布局法(Flat Placement)^[181]比较适合于模拟集成电路器件合并、对称布局、对称布线等要求比较高的

特点。为了使该方法能够适应模拟集成电路,一些模拟宏单元系统^[6,67,69,175,178,180]在布局时能保证匹配器件与对称线之间的距离基本相等,从而有效抑制器件失配效应。

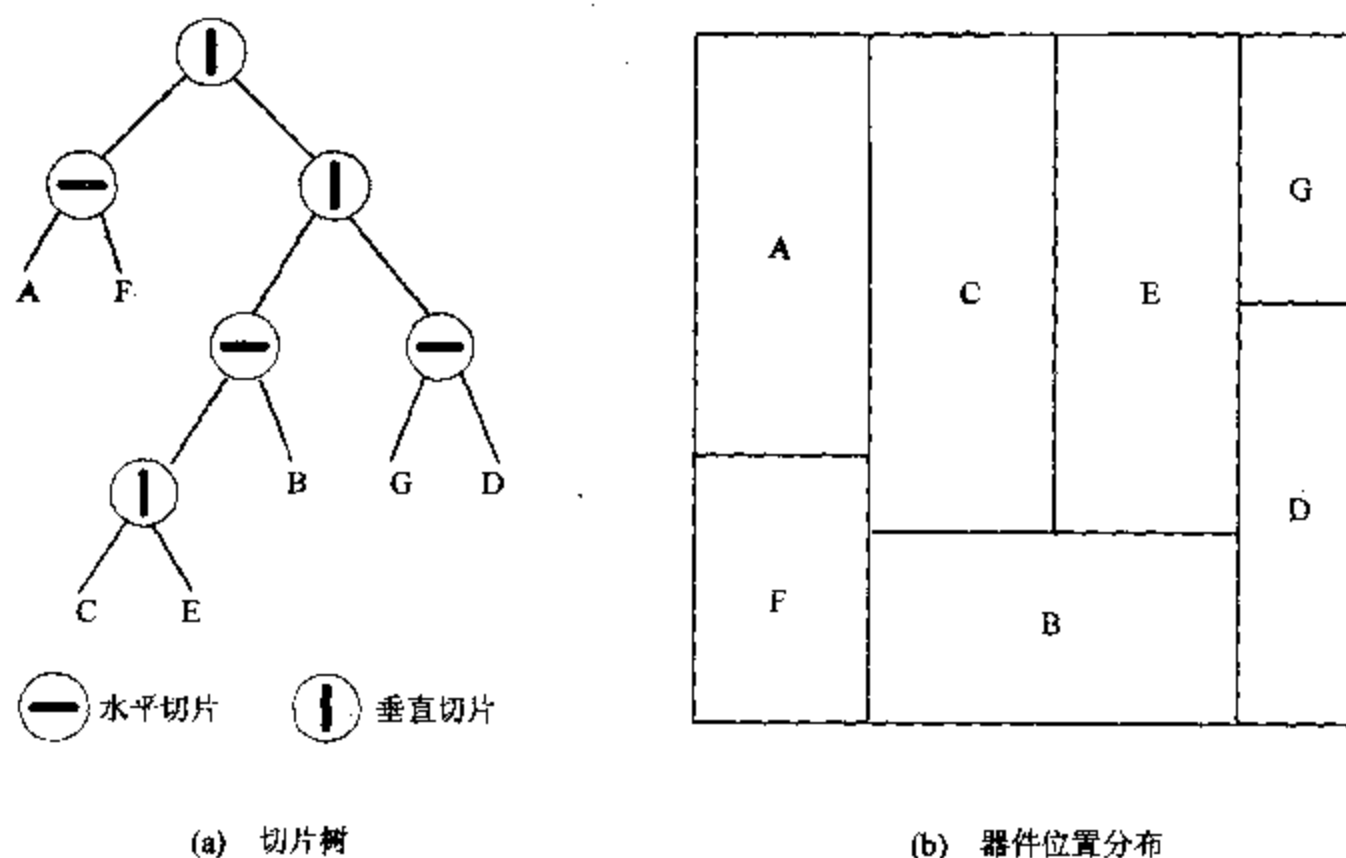


图 6.13 切片树及其对应的器件位置分布

3. 布线

通道布线法^[6,67,180,182]和迷宫算法^[45,70,183]是两类常用的模拟宏单元的布线算法。通道布线法将连线放置在切片(器件)的分割处,可以在不影响器件布局结果的情况下非常方便地增加布线通道的宽度,但其缺陷是不能在器件上走线。从电规则特性上看,该缺陷虽然是可以接受的,但它对模拟集成电路器件级的版图设计模式的抽象毕竟不够完美^[70],因而存在版图不紧凑、质量欠佳等不尽人意之处。迷宫算法采用一些非通道的形状来表示连线路径,迷宫算法的长处在于能找到平面上两个点之间的最短路径,因此有助于缩短连线、减小因布线所产生的寄生电阻和寄生电容。迷宫算法没有采用通道,当需要增加布线空间时,就很难对器件布局结果进行恰如其分的调整。因此,迷宫算法需要将精确估计连线空间的“重任”回踢给布局算法。为了提高模拟集成电路版图的质量,通道布线法^[67,182]和迷宫算法^[70]都采用了一些启发式策略以减少信号线之间的串扰。在众多布线系统中,只有卡内基梅隆大学的 ANAGRAM I 能够在器件上布线^[70]。

综上所述,宏单元法是一种通用、灵活的器件级版图综合方法,模拟宏单元系统能够为任何拓扑结构、任何器件尺寸的模拟电路生成版图。模拟宏单元法的缺陷是所生成的版图带有浓厚的数字电路版图的色彩,这使得其版图质量仍然不能同模拟版图专家们的手工杰作相媲美。但是,通过改善器件布局和布线的质量可以极大地提高其版图质量^[6,69,70],使之更适合于模拟集成电路的版图设计。由于模拟宏单元具有通用性和灵活性,它将是模拟集成电路版图综合工具的发展方向,关键的问题是需要实现能支持器件合并、能集中并对称分布匹配器件的布局系统^[69],实现对称布线和抑制耦合效应的布线系统^[70]。因此,下面将用较多的篇幅详细介绍适合于模拟集成电路的器件级布局、布线算法。

6.3 器件级布局算法

层次式布局比较适合丁单元很多且尺寸比较规范的数字集成电路,而平铺式布局(Flat Placement)^[181]更适合于模拟集成电路。平铺式布局首先根据电路规模的大小,预估一个面积适中的方块,然后将所有器件放置到该方块上,器件的位置用它在该方块上的绝对坐标来表示。平铺式布局是一种无网格布局方法,除要求最终结果满足设计规则外,在布局过程中不要求器件之间的相对位置满足任何约束。平铺式布局采用模拟退火法调整各器件的相对位置,最终使芯片面积最小^[184,185]。

6.3.1 器件级布局的基本思想

从第二章关于模拟退火法的讨论和前面关于版图布局问题的分析不难看出:布局问题的描述、新布局的生成和对布局结果的评估是器件级布局算法的核心问题。

1. 布局状态

对于给定的模拟电路,布局空间应当足够大,以便在退火过程中器件能自由移动,否则将不利于减小芯片面积,甚至无法放置所有器件和完成后续的布线工作;但如果布局空间过大,则会为搜索额外空间花费不必要的计算量。因此,通常按下述方式来估计布局空间的面积(A_{Field})、长度(L_{Field})和宽度(W_{Field}):

$$\begin{cases} A_{\text{Field}} = k \sum_{j=1}^{N_d} A_{\max}(d_j) \\ L_{\text{Field}} = W_{\text{Field}} = \sqrt{A_{\text{Field}}} \end{cases} \quad (6.1)$$

其中, N_d 是电路中器件的个数, $A_{\max}(d_j)$ 是第 j 个器件 d_j 的最大面积,常数 k 称为比例因子, k 不宜过大、也不宜过小。

上述确定布局空间的方式将使最后的布局结果趋向正方形,我们也可以在保证布局空间的面积不变的情况下,根据需要调整其宽长比。

在数字集成电路的布局系统中,器件用一个黑盒子表示,黑盒子的长度和宽度分别对应器件版图的长度和宽度,器件的连线端口用黑盒子边框上几何点来表示,即:连线端口只有位置信息,但不占任何面积,也没有周长。对于模拟集成电路而言,仅简单地表示器件版图的长度和宽度是不够的,因为这种表示方式掩盖了器件内部的结构,既不利于进行器件合并,也不能在布局阶段使匹配器件具有相同的走向和外形,当然,在布局阶段保留所有掩模层的几何数据也是没有必要的。因此,KOAN^[6,69]所采用的器件表示法是比较好的,它用一组称为保护框的矩形来表示器件的版图,这些保护框除了能表示器件版图的边框外,还能表示器件的源极、漏极和栅的位置与形状,这些保护框是器件合并和对称布局所必需的重要信息;器件的连线端口位于边框上,连线端口的面积为零,仅包含位置信息,连线端口的数目和器件版图的边框尺寸有助于准确地估计连线空间。为便于处理,器件的位置都以布局空间的中心为参照点。

上述关于布局空间和器件版图的表示方法既简便又灵活,如果有必要,还可以在布

线阶段对布局结果进行适当的再调整。尽管这种调整非常容易,但是,我们总希望布局结果尽可能合理,希望布线阶段的再调整越少越好。因此,合理、准确地估计连线所需的空

间是十分必要的。

连线空间越小,连线的布通率就越低,连线空间过大,则会浪费芯片面积,但遗憾的是在器件布局阶段不可能预知布线的形状和位置,因此决定连线空间的大小和位置是十分困难的。C. Sechen 所提出的布线环(Wire-space Halos)估计法^[181]是值得借鉴的。由于目前的模拟宏单元法已经实现了在器件上布线^[6,70],造成布线失败的原因主要是被连接的器件端口太拥挤,因此,根据器件局部的连接拥挤度来估计布线环的宽度是比较合理的。J. M. Cohn 采用单位周长的连线端口数目来度量器件的连接度取得了令人满意的结果^[6,69]。假设器件 d_j 具有 $N_{\text{Terminal}}(d_j)$ 个连线端口,且周长为 $L_{\text{Perimeter}}(d_j)$,则其布线环的宽度为:

$$W_{\text{Halo}}(d_j) = \lambda \frac{N_{\text{Terminal}}(d_j)}{L_{\text{Perimeter}}(d_j)} \quad (6.2)$$

其中,常数 λ 是通过反复实验所获得的比例因子。在模拟集成电路中,由于器件的连线端口数一般在 2 到 4 之间,因此,布线环宽度主要由其版图的周长决定,周长越小,布线环宽度就越大。

在用模拟退火法进行平铺式布局的过程中,各器件在布局空间中的一种位置分布、器件版图的表示形式和布线环统称为布局状态,使布局状态逐步迁移到最佳状态是平铺式布局的目的。为便于读者形象地认识布局空间、器件与连线端口的表示方法以及布线环等概念,图 6.14 给出了模拟集成电路的器件级布局示意图。

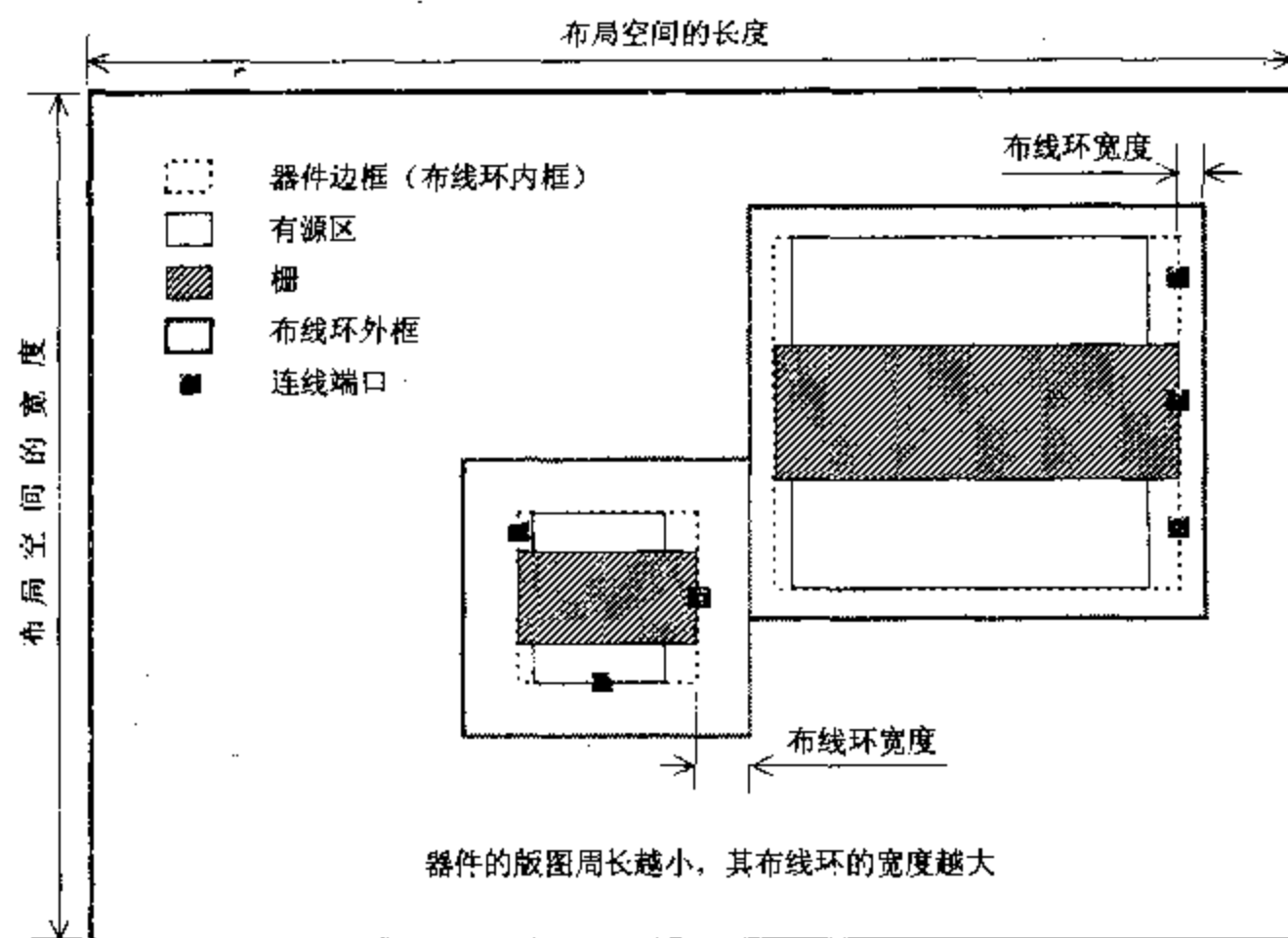


图 6.14 器件级布局示意图

2. 状态迁移

在布局过程中,从旧布局状态变化到新布局状态的过程称为状态迁移。器件重定位和重整形是两类基本的状态迁移。

器件重定位又分为平移、旋转、镜象和对换等四种形式,它们都不会改变器件的版图外形,分别按下述方式进行:

平移:随机选择一个器件,然后将它平移到布局空间的一个新位置上,水平方向和垂直方向的移动量是随机产生的。参见图 6.15(a)。

旋转:随机选择一个器件,在保证器件中心位置不变的情况下作逆时针旋转,对于模拟集成电路而言,一般只允许 8 个旋转角度,因此,旋转角度 φ 是能被 45° 整除的随机数。参见图 6.15(b)。

镜象:随机选择一个器件,在保证器件中心位置不变的情况下作水平镜象或垂直镜象。参见图 6.15(c)。

对换:随机选择两个器件,互换其中心位置。参见图 6.15(d)。

器件重整形主要有连线端口平移和外形变换等两种,它们与重定位的区别在于会改变器件版图的外形,分别按下述方式进行:

端口平移:随机选择一个连线端口,将它平移到该器件边框的另一位置,该位置也是随机选择的。参见图 6.15(e)。

外形变换:随机选择一个器件,将其外形变换为另一个电特性不变的形状,然后调整器件的位置,使变换前后器件版图的中心位置不变。外形变换的主要方式是采用折叠式器件结构。参见图 6.15(f)。

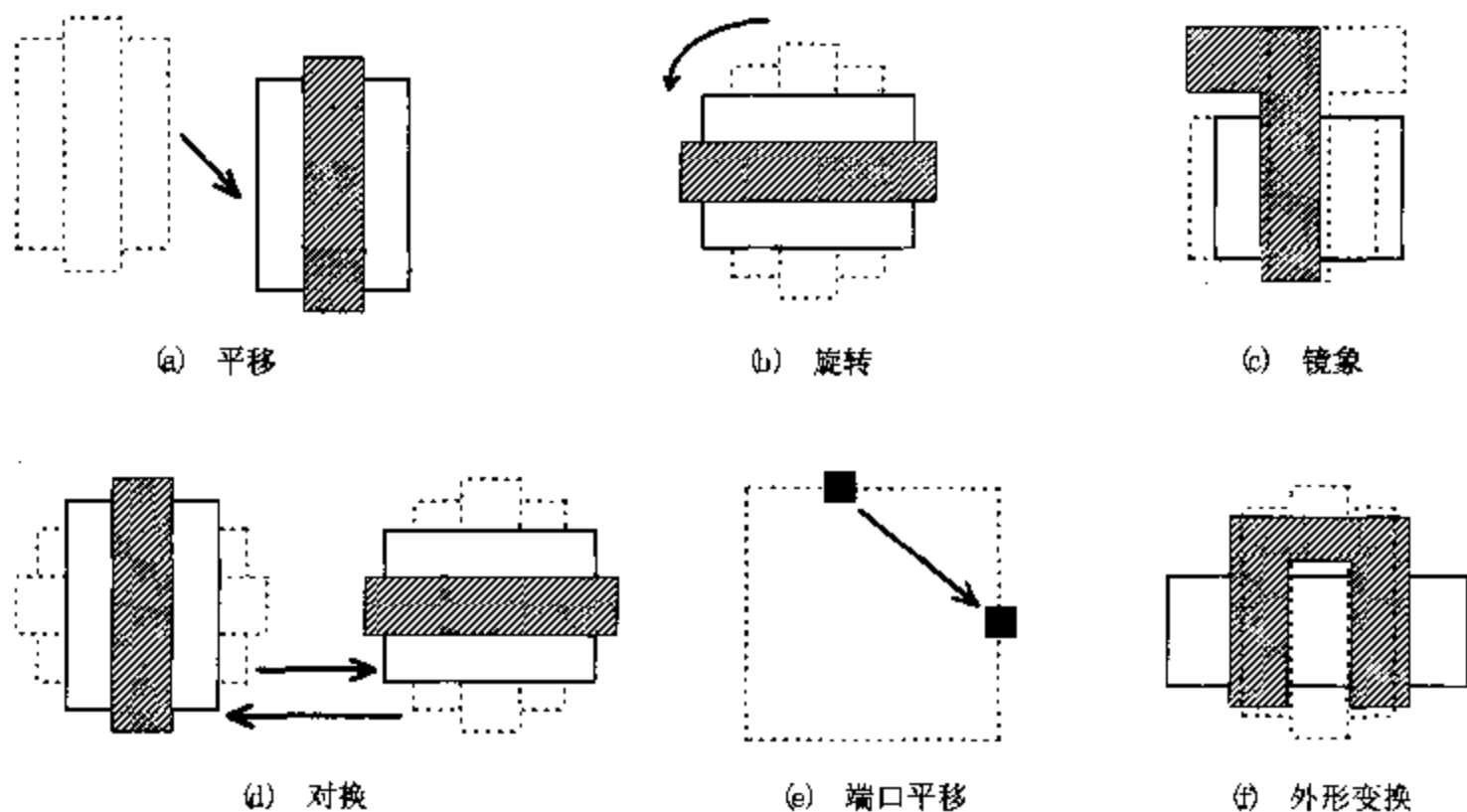


图 6.15 状态迁移示意图

3. 目标函数

前面介绍的状态迁移主要是随机地形成新的布局状态,而目标函数(又称价格函数)

则是模拟集成电路器件级布局系统评价布局状态(或状态迁移)的尺度。在通常情况下,布局状态的目标函数越小,其对应的布局结果就越好。

在处理器件级布局问题时,需要折衷考虑目标函数的复杂性、状态迁移及其实现的复杂性等因素。例如:对于采用切片布局^[184]的系统而言,其状态迁移非常复杂,因为它必须保证新的布局状态是合法的,即新状态中没有器件重叠等违反设计规则的问题,但其目标函数却非常简单,因为目标函数中不必考虑非法布局所引入的惩罚项;如果采用前面介绍的平铺式布局法,状态迁移及其实现都非常简单,但由于迁移后的新状态中存在一些非法的布局,因此必须引入比较复杂的罚函数项,以保证能最终获得合法的布局结果。

追求芯片面积最小和连线最短是模拟集成电路版图综合的基本目标,同时,平铺式布局系统^[6, 69]还必须保证最终的布局状态满足所有设计规则,没有非法的重叠,因此,平铺式布局系统的目标函数 $C_{\text{Total}}(X)$ 为:

$$C_{\text{Total}}(X) = \alpha_{\text{Overlap}} C_{\text{Overlap}}(X) + \alpha_{\text{Area}} C_{\text{Area}}(X) + \alpha_{\text{Length}} C_{\text{Length}}(X) \quad (6.3)$$

其中, X 表示布局状态, $C_{\text{Overlap}}(X)$, $C_{\text{Area}}(X)$ 和 $C_{\text{Length}}(X)$ 是器件重叠、版图面积和连线长度所对应的目标函数,常数 α_{Overlap} , α_{Area} 和 α_{Length} 则是相应的权重系数。

计算版图面积所对应的目标函数 $C_{\text{Area}}(X)$ 最简单,只要计算包含所有器件的最小矩形框(简称版图边框)的面积即可。对于模拟集成电路而言,使 $C_{\text{Area}}(X)$ 最小不但可以提高硅片的利用率,而且还能够缩小匹配器件的间距,使版图更好地满足电路设计者的要求。在布局过程中,记录版图边框是非常容易的,只要每完成一次状态迁移后就检查该器件与版图边框的位置关系,并按下述规则进行即可:

- (1) 如果该器件超出了原版图边框,则增大版图边框,使之能包含移动后的器件;
- (2) 如果在状态迁移前该器件是决定原版图边框的某条边的唯一因素,则重新计算新的版图边框;
- (3) 其他情况则保持原版图边框不变。

目标函数 $C_{\text{Overlap}}(X)$ 的计算比较复杂,对于包含 N_d 个器件的版图而言, $C_{\text{Overlap}}(X)$ 定义为:

$$C_{\text{Overlap}}(X) = \sum_{j=1}^{N_d} \sum_{i=j+1}^{N_d} [A_{\text{IllegalOverlap}}(d_i, d_j, X)]^2 \quad (6.4)$$

其中, $A_{\text{IllegalOverlap}}(d_i, d_j, X)$ 表示在布局状态 X 下器件 d_i 和 d_j 之间非法交叠部分的面积。非法交叠主要有下述三类:

- (1) 位于不同掩模层的连线端口的非法交叠,这类非法交叠违反了设计规则,例如: p 型扩散区与 n 型扩散区之间的交叠;
- (2) 位于相同掩模层的连线端口的非法交叠,这类非法交叠违反了电学规则,造成不同线网(电路中的节点)的短路,例如:两个栅极节点不同的 MOS 晶体管的栅掩模之间的交叠;
- (3) 不同器件的布线环出现了交叠,这会使布线阶段没有足够布线空间,从而导致布线失败的局面。

在用模拟退火法对布局状态 X 进行迭代改进的优化过程中,必须使目标函数 $C_{\text{Overlap}}(X)$ 最终为零,否则,布局结果将是错误的。正因如此,我们才用非法交叠面积的

平方和来表示 $C_{\text{Overlap}}(X)$ ，这有助于促使那些较大的交叠面积迅速减小到零。

需要特别指出：按式(6.4)直接计算 $C_{\text{Overlap}}(X)$ 的效率非常低，因为需要检查每两个器件之间的交叠，计算复杂度为 $O(N_d^2)$ 。其实，非法交叠只出现在邻近的器件之间，因此，只要将布局空间划分为若干大小相等的格子，并检查这些格子内各器件之间的交叠即可^[181]，从而使计算复杂度降为 $O(N_d)$ 。

在器件级布局阶段，目标函数 $C_{\text{Length}}(X)$ 是所有线网长度的有权和，即：

$$C_{\text{Length}}(X) = \sum_{j=1}^{N_{\text{Net}}} \beta_j L_{\text{Net}}(j) \quad (6.5)$$

其中， N_{Net} 是版图中线网的数目， $L_{\text{Net}}(j)$ 是在布局状态 X 下第 j 条线网的长度， β_j 是第 j 条线网的权重系数， β_j 的默认值为 1。在模拟集成电路中，对寄生参数特别敏感的关键线网应越短越好，对于这类线网，应当选择较大的权重系数。

计算 $C_{\text{Length}}(X)$ 的关键是如何精确、快速地估计 $L_{\text{Net}}(j)$ 。如果一个线网 AB 仅包含两个端口 A 和 B ，则其连线长度的估计很简单。设 A 和 B 在布局空间中的坐标分别为 (x_1, y_1) 和 (x_2, y_2) ，则该线网的长度 $L_{\text{Net}}(AB)$ 可以用下述矩形距离来度量：

$$L_{\text{Net}}(AB) = |x_2 - x_1| + |y_2 - y_1| \quad (6.6)$$

在布局过程中，一般不采用两点间的直线距离来度量线网长度，因为两个端口之间的连线一般是多段横平竖直的折线，而不是直线，因此，直线距离不仅计算量大，而且不如矩形距离精确。

对于包含多个端口的线网而言，通常采用半周长、全图、几何重心、最短生成树或最短 Steiner 树来估计线网长度。假设某线网包含 N_T 个端口，这五种方法的具体过程如下：

半周长法：选择一个包含所有 N_T 个端口的中心的矩形，用该矩形周长的一半来度量该线网的长度。图 6.16(a) 是半周长法的示意图。半周长法的计算复杂度为 $O(N_T)$ 。

全图法：以 N_T 个端口的中心为顶点构成的全图包含 $N_T(N_T-1)/2$ 条边，所有这些边的连线长度之和即为该线网的长度。图 6.16(b) 是全图法的示意图。全图法的计算复杂度为 $O(N_T^2)$ ^[186]。

几何重心法： N_T 个端口的中心构成一个多边形，该多边形的重心 G 即为线网的重心，几何重心法用各端口中心与 G 之间的 N_T 条连线长度之和来度量线网的长度。图 6.16(c) 是几何重心法的示意图。几何重心法的计算复杂度为 $O(N_T^2)$ 。

最短生成树法：从 N_T 个端口的中心为顶点构成的全图中选出一个连线最短的树，用该树的 (N_T-1) 条边的连线长度之和来度量该线网的长度。图 6.16(d) 是最短生成树的示意图。最短生成树的计算复杂度为 $O(N_T^2)$ ^[186]。

最短 Steiner 树法：最短 Steiner 树法^[187] 与最短生成树法非常相似，但在形成全图时还引入了另一个顶点——Steiner 点，由此所获得的树称为 Steiner 树，它包含 N_T 条边。连线最短的 Steiner 树是布线时可能实现的最短线网，用最短 Steiner 树中 N_T 条边的连线长度之和来度量线网长度最为精确。确定最短 Steiner 树是一个 NP-完全问题。目前已经有一些确定 Steiner 树的启发式算法，其计算复杂度在 $O(N_T \log N_T)$ 和 $O(N_T^2)$ 之间^[186]，但不能保证所获得的是最短 Steiner 树。图 6.16(e) 是最短 Steiner 树的示意图。

对于模拟集成电路的器件级布局而言，半周长法、全图法和几何重心法所估计的线

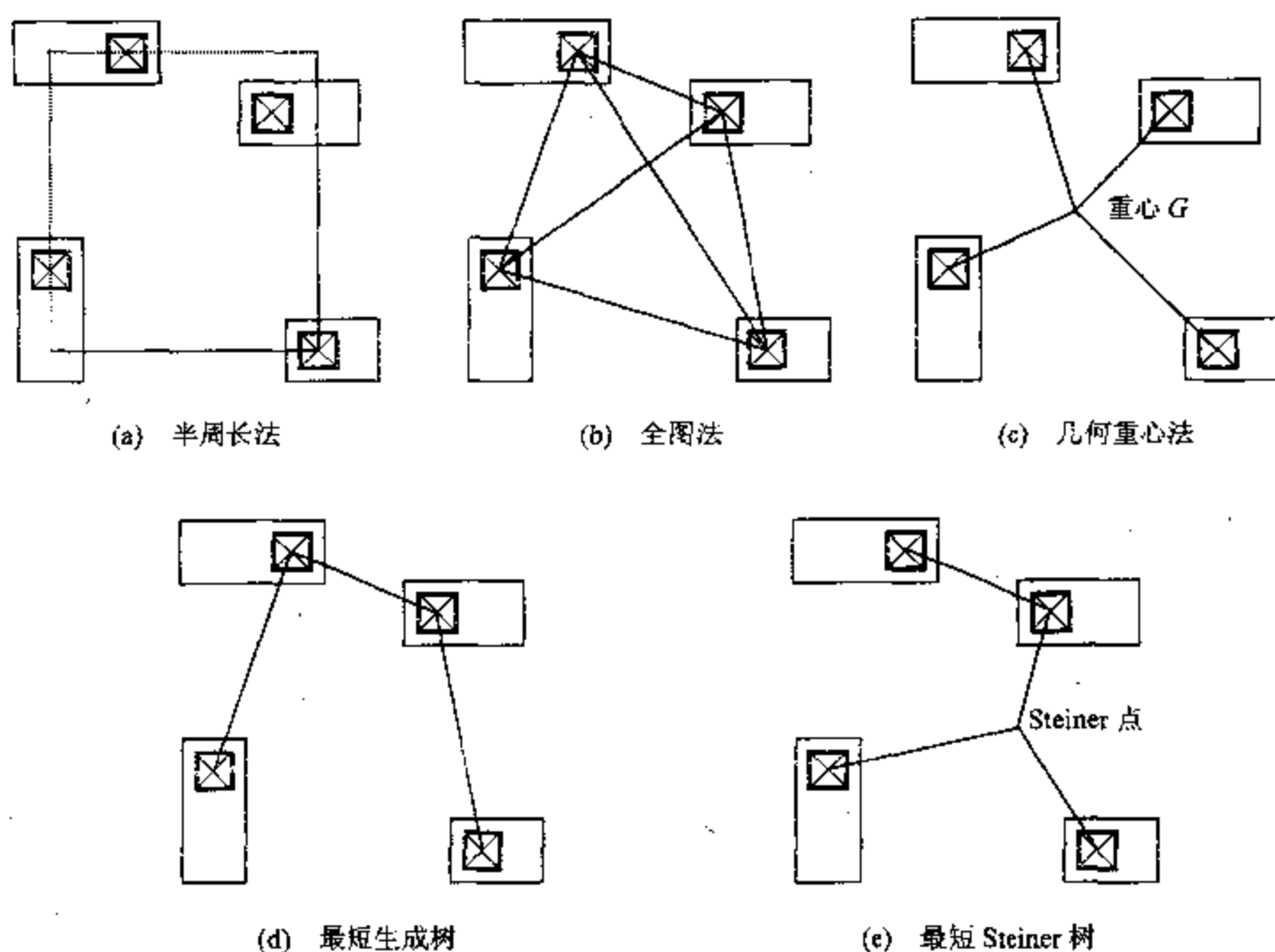


图 6.16 线网长度的估计方法

网长度都太粗糙，而最短 Steiner 树虽然比较精确，也有助于使版图的连线最短，但求 Steiner 树太复杂，因此，用最短生成树估计线网长度是比较好的选择^[69]。

6.3.2 拓扑约束

前面已多次提到，要求两个甚至多个器件相互匹配是模拟集成电路设计的特点，对于布局过程而言，如何满足电路级综合时所提出的这些拓扑约束是器件级布局的主要任务之一。通过改进基本的状态迁移和目标函数，就可以妥善地处理拓扑约束问题。考虑到层次式版图综合的需要，本小节还将讨论调整版图宽长比的具体措施。

1. 对称性状态迁移

前面介绍的基本状态迁移不能保证匹配器件的版图对称性，不能很好地满足拓扑约束的需要，因此，需要引入器件自对称、对称迁移、固定器件走向和固定器件中心等对称性状态迁移。

器件自对称的实现比较简单，只要分两个阶段实现即可。首先，在产生器件的初始版图时应当使之具有自对称性；其次，如果需要对某器件进行随机的折叠重整型，则必须确保重整型后的器件仍然具有自对称性。例如：如果某器件原来的漏极位于器件版图的最左、最右侧，参见图 6.1(b)，则重整型后的器件版图必须保持该特性不变。

对称迁移又分为对称平移、对称旋转、对称镜象、对称互换和对称重整型，主要是保证迁移前具有对称性的两个器件在迁移后仍然具有相同的对称性。

固定器件走向的主要目的是让具有自对称性的器件在状态迁移前后其对称轴的方向

保持不变, 实现起来也是非常容易的。

固定器件中心对实现布局的几何对称性和电性能对称性都是非常必要的。对于电路中那些功耗较大的器件而言, 它们常常是电路中的热源, 例如: 图 6.8(a) 中的晶体管 Q5 就是这种情况。必须将这类器件的版图中心固定在受它影响的那些匹配器件的对称轴上, 参见图 6.8(b)。

2. 走向约束

由于离子注入等工艺具有各向异性的特点, 如果匹配器件的走向不一致, 则会造成比较严重的失配效应, 从而违背拓扑约束。简单而有效的解决方法就是让所有器件的栅极平行。可以引入下述关于器件走向的目标函数 $C_{\text{Orientation}}(X)$:

$$C_{\text{Orientation}}(X) = \sum_{j=1}^{N_d} \delta_{\text{Orientation}}(d_j) \quad (6.7)$$

其中, $\delta_{\text{Orientation}}(d_j)$ 是在布局状态 X 下器件 d_j 的方向特征函数, 如果 d_j 的栅极沿垂直方向, 则其特征函数值为 0, 否则为 1。

上述选择 $C_{\text{Orientation}}(X)$ 的方式将使器件的栅极沿垂直方向排列, 这不仅符合版图设计专家们的审美习惯(因为他们总是习惯于将器件的栅极沿垂直方向排列), 而且还有另外两方面的作用: ①可以采用使多晶硅连线沿垂直方向、金属连线沿水平方向的简单布线策略, 既可以简化布线工作, 又能提高布线效率; ②有助于实现器件合并, 本书将在第 6.3.3 节中详细讨论该问题。

我们只要将器件走向目标函数 $C_{\text{Orientation}}(X)$ 加入到式(6.3)所定义的总目标函数中, 就可以使模拟退火法获得满足器件走向约束的布局结果。考虑到器件走向约束后的总目标函数为:

$$C_{\text{Total}}(X) = \alpha_{\text{Overlap}} C_{\text{Overlap}}(X) + \alpha_{\text{Area}} C_{\text{Area}}(X) \\ + \alpha_{\text{Length}} C_{\text{Length}}(X) + \alpha_{\text{Orientation}} C_{\text{Orientation}}(X) \quad (6.8)$$

其中, 常数 $\alpha_{\text{Orientation}}$ 是器件走向约束的权系数, 可由实验决定。

3. 间距约束

由于迁移率、氧化层厚度、衬底杂质浓度、外延层的电导率等工艺参数在空间分布的不均匀性是产生器件失配效应的重要因素, 减小这些因素对匹配器件影响的有效措施, 是让匹配器件尽可能靠近。因此, 需要引入关于匹配器件间距的约束。对于给定的工艺水平, 当器件间距超过 D_{max} 时, 迁移率、氧化层厚度、衬底杂质浓度、外延层的电导率的差异将会比较大, 并可能对器件性能产生比较明显的影响, 换言之, 器件级模拟集成电路的布局系统必须保证匹配器件的横向间距和纵向间距均小于 D_{max} 。因此间距约束的目标函数 $C_{\text{Separation}}(X)$ 可以定义为:

$$C_{\text{Separation}}(X) = \sum_{j=1}^{N_s} \delta_{\text{Separation}}(S_j) \quad (6.9)$$

其中, N_s 是版图中间距约束的个数; S_j 是在布局状态 X 下第 j 个间距约束的版图距离, 它是遵循间距约束的两个器件的横向间距和纵向间距的最大值; $\delta_{\text{Separation}}(S_j)$ 是 S_j 的度量函数, 它定义为:

$$\delta_{\text{Separation}}(S_j) = \begin{cases} 0, & S_j \leq D_{\max} \\ (S_j - D_{\max})^2, & S_j > D_{\max} \end{cases} \quad (6.10)$$

在上式中采用平方项的目的是促使违反间距约束的器件在模拟退火过程中能够被迅速调整到间距小于 $D_{\max}$ 的范围内, 其作用与式(6.4)雷同。考虑间距约束后的总目标函数为:

$$C_{\text{Total}}(X) = \alpha_{\text{Overlap}} C_{\text{Overlap}}(X) + \alpha_{\text{Area}} C_{\text{Area}}(X) + \alpha_{\text{Length}} C_{\text{Length}}(X) \\ + \alpha_{\text{Orientation}} C_{\text{Orientation}}(X) + \alpha_{\text{Separation}} C_{\text{Separation}}(X) \quad (6.11)$$

其中, 常数 $\alpha_{\text{Separation}}$ 是间距约束的权系数, 由实验决定。

4. 宽长比约束

在层次式版图设计中, 常常需要单元版图的宽长比为给定值, 我们称之为宽长比约束。采用与前面处理走向约束和间距约束相似的方法, 我们引入关于宽长比约束的目标函数 $C_{\text{AspectRatio}}(X)$:

$$C_{\text{AspectRatio}}(X) = \left(R_{\text{Aspect}} - \frac{W_{\text{Field}}}{L_{\text{Field}}} \right)^2 \quad (6.12)$$

其中, R_{Aspect} 是预期的版图宽长比, L_{Field} 和 W_{Field} 分别是在布局状态 X 下布局空间的长度和宽度。很显然, 当 R_{Aspect} 比较大时, 器件走向将与水平方向平行; 当 R_{Aspect} 比较小时, 器件走向则会与垂直方向平行。考虑到宽长比约束后, 总目标函数为:

$$C_{\text{Total}}(X) = \alpha_{\text{Overlap}} C_{\text{Overlap}}(X) + \alpha_{\text{Area}} C_{\text{Area}}(X) \\ + \alpha_{\text{Length}} C_{\text{Length}}(X) + \alpha_{\text{Orientation}} C_{\text{Orientation}}(X) \\ + \alpha_{\text{Separation}} C_{\text{Separation}}(X) + \alpha_{\text{AspectRatio}} C_{\text{AspectRatio}}(X) \quad (6.13)$$

其中, 常数 $\alpha_{\text{AspectRatio}}$ 是宽长比约束的权系数, 由实验决定。

6.3.3 器件合并

器件合并是减小芯片面积、减少寄生参数最有效的手段, 通过器件合并, 还可以减小布线的难度和计算量, 因为凡是合并的节点都不需要连线了。在模拟集成电路中, 下述几种情况需要进行器件合并:

(1) **合并 MOS 有源扩散区**: 如果两个 MOS 晶体管的源漏极在电路中为同一个节点, 则可以合并, 参见图 6.2。类似地, 当器件版图采用折叠式结构时, 同一个晶体管的源漏极通常是由金属连线所联接的, 如果两个 MOS 晶体管的这类连线在电路中为同一个节点, 也可以合并。

(2) **合并 MOS 栅极**: 如果两个 MOS 晶体管的栅极在电路中为同一个节点, 则可以采用图 6.3 所示的墩接方式进行合并。

(3) **合并 MOS 阱区**: 如果两 MOS 晶体管的类型相同, 且其衬底在电路中为同一个节点, 则可以合并, 参见图 6.17(a)。类似地, 阱区的接触孔也是可以合并的。

(4) **合并双极型晶体管的隔离环**: 由于双极型集成电路的隔离环总是接电路中的最低电位, 因此任意两个双极型晶体管的隔离环都是可以合并的, 参见图 6.17(b)。

(5) **合并双极型晶体管的集电极**: 如果同类型的双极型晶体管的集电极在电路中为同一个节点, 则可以合并其集电极, 参见图 6.17(c)。

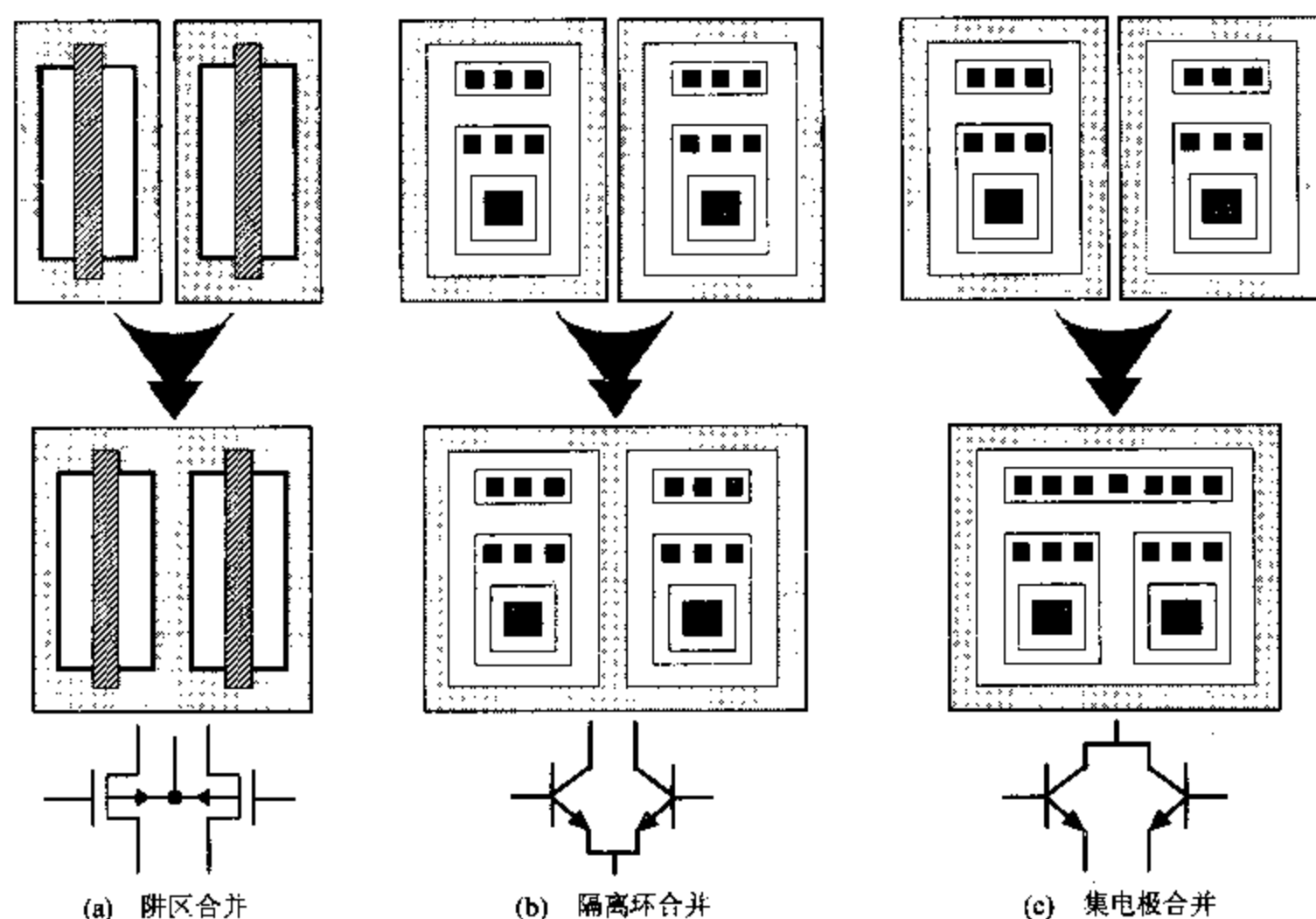


图 6.17 器件合并示意图

为了使器件级的布局系统能够处理器件合并问题，需要在状态迁移中加入支持器件合并的操作，并在总的目标函数中反映器件合并的程度。

与处理拓扑约束中的对称性迁移一样，需要在基本的状态迁移中扩充一些能反映器件合并的操作。对称性迁移主要是处理两个器件，而器件合并则需要处理多个器件。扩充后的状态迁移过程为：随机选择一个器件 d_j ，如果 d_j 与任何器件之间都没有任何几何共享，则按基本状态迁移和对称性迁移对该器件进行操作；否则按下述步骤进行：

- (1) 找出与 d_j 有几何共享的所有器件，记为 $G_{\text{Share}}(d_j) = \{d_j, d_{s,1}, d_{s,2}, \dots, d_{s,K}\}$ 。
- (2) 为器件 d_j 随机选择一种状态迁移方式：平移、旋转、对换或重整型。
- (3) 如果选择的状态迁移为重整型，就为 d_j 随机选择一种合法的形状，然后随机选择一种对齐方式以确定重整型后 d_j 的中心位置，可选对齐方式为：

- ① 让重整型后 d_j 的版图与其原版图保持左下角对齐；
- ② 让重整型后 d_j 的版图与其原版图保持左上角对齐；
- ③ 让重整型后 d_j 的版图与其原版图保持右下角对齐；
- ④ 让重整型后 d_j 的版图与其原版图保持右上角对齐；
- ⑤ 让重整型后 d_j 的版图与其原版图保持中心对齐。

不同的对齐方式将使原有的器件合并或者被保留、或者被破坏。

- (4) 如果选择的状态迁移为平移或旋转，则需要“只对 d_j 作相应的操作”和“对 $G_{\text{Share}}(d_j)$ 中的所有器件都作相应的操作”之间进行随机抉择。如果随机抉择结果为后者，则将 $G_{\text{Share}}(d_j)$ 中所有器件作为一个整体以进行相应的状态迁移，从而使他们之间的几何共享保持不变；如果随机抉择结果为前者，当 d_j 的迁移量很大时，迁移后的 d_j 可能与 $G_{\text{Share}}(d_j)$ 中别的任何器件之间都没有几何共享，则需要将 d_j 从 $G_{\text{Share}}(d_j)$ 中删去。

(5) 如果选择的状态迁移是器件对换, 则首先选择一个与 d_j 对换的器件 d_j^+ , 然后按与步骤(4)雷同的方式, 决定只对换 d_j 和 d_j^+ , 还是需要为 $G_{\text{Share}}(d_j)$ 中的所有器件都选择一个对换对象, 并作对换操作。

为了使模拟退火法能够处理器件合并问题, 同样需要在总目标函数 $C_{\text{Total}}(X)$ 中加入关于器件合并的目标函数 $C_{\text{Merge}}(X)$ 。为此, 我们需要改进对布局问题的描述, 主要是使关于器件版图的描述包含更丰富的内容。在 6.3.1 中, 我们简单地借鉴数字集成电路版图布局的方法, 将连线端口看作几何点, 仅包含位置信息, 没有面积, 也没有周长, 这种抽象过于简单, 不能处理器件合并问题。为此, 我们对连线端口 t 增加两项信息: 可合并的面积 $A_{\text{Mergeable}}(t)$ 和可合并的周长 $L_{\text{MergeablePerimeter}}(t)$, 这样, 我们就可以用未被合并的周长和面积来度量版图中器件合并的程度。因此, 关于器件合并的目标函数 $C_{\text{Merge}}(X)$ 定义为:

$$C_{\text{Merge}}(X) = \sum_{j=1}^{N_N} \eta_j \left\{ \sum_{i=1}^{N_{t,j}} \xi_{ij} [A_{\text{Mergeable}}(t_{ij}) - A_{\text{Merged}}(t_{ij})] + \sum_{i=1}^{N_{t,j}} \rho_{ij} [L_{\text{MergeablePerimeter}}(t_{ij}) - L_{\text{MergedPerimeter}}(t_{ij})] \right\} \quad (6.14)$$

其中, N_N 是电路中线网的条数, $N_{t,j}$ 是第 j 条线网的连线端口的数目, η_j 是第 j 条线网的权系数, t_{ij} 是第 j 条线网的第 i 个连线端口, ξ_{ij} 和 ρ_{ij} 分别是关于 t_{ij} 的面积和周长的权系数, $A_{\text{Merged}}(t_{ij})$ 和 $L_{\text{MergedPerimeter}}(t_{ij})$ 分别是在布局状态 X 下连线端口 t_{ij} 已和别的连线端口共享的面积和周长。

类似于前面处理器件走向约束和版图宽长比约束的方法, 我们只要将关于器件合并的目标函数 $C_{\text{Merge}}(X)$ 加入到总目标函数 $C_{\text{Total}}(X)$ 中即可:

$$C_{\text{Total}}(X) = \alpha_{\text{Overlap}} C_{\text{Overlap}}(X) + \alpha_{\text{Area}} C_{\text{Area}}(X) + \alpha_{\text{Length}} C_{\text{Length}}(X) + \alpha_{\text{Orientation}} C_{\text{Orientation}}(X) + \alpha_{\text{Separation}} C_{\text{Separation}}(X) + \alpha_{\text{AspectRatio}} C_{\text{AspectRatio}}(X) + \alpha_{\text{Merge}} C_{\text{Merge}}(X) \quad (6.15)$$

其中, 常数 α_{Merge} 是关于器件合并的权系数, 由实验决定。

6.3.4 权系数的选择

在式(6.15)所示的总目标函数中, 共有 7 个权系数, 它们的取值将直接影响布局系统的效率和布局结果的质量。

在所有 7 个权系数中, 只有 α_{Overlap} 与版图的可行性(或可制造性)相关, 因此必须使 α_{Overlap} 足够大, 以保证 $\alpha_{\text{Overlap}} C_{\text{Overlap}}(X)$ 在 $C_{\text{Total}}(X)$ 中起最主要的作用, 从而保证布局结果中不包含非法的交叠。然而, 如果 α_{Overlap} 过大, 则不但会掩盖芯片面积、连线长度等因素的作用, 而且会严重影响布局系统的效率, 使计算量大大增加。因此, 在模拟退火过程中动态地调整 α_{Overlap} 是非常必要的^[65]。动态调整 α_{Overlap} 的基本思想是: 每达到一个退火温度的热平衡状态时, 就检查当前的非法交叠目标函数 $C_{\text{Overlap}}(X)$, 如果它大于零, 则意味着版图中存在非法交叠, 因而需要在下一个退火温度时增大权系数 α_{Overlap} 以尽快消除版图中的非法交叠; 如果 $C_{\text{Overlap}}(X)$ 为零, 则表示版图中已经没有非法交叠, 可以在下一个退火温度时适当减小权系数 α_{Overlap} 。所以, 当模拟退火法在温度 T_i 下达到热平衡后, 就按下

述方式调整退火温度 T_{i+1} 时的权系数 $\alpha_{\text{Overlap}}^{(i+1)}$:

$$\alpha_{\text{Overlap}}^{(i+1)} = \begin{cases} \max(\alpha_{\text{Overlap-max}}, \tau \alpha_{\text{Overlap}}^{(i)}), & C_{\text{Overlap}}(X^{(i)}) > 0 \\ \min(\alpha_{\text{Overlap-min}}, \tau^{-1} \alpha_{\text{Overlap}}^{(i)}), & C_{\text{Overlap}}(X^{(i)}) = 0 \end{cases} \quad (6.16)$$

其中, $X^{(i)}$ 表示温度为 T_i 的热平衡状态所对应的布局状态, $\alpha_{\text{Overlap-max}}$ 和 $\alpha_{\text{Overlap-min}}$ 分别是最大和最小的权系数, $\alpha_{\text{Overlap-max}}$ 取 $100\alpha_{\text{Area}}$ 就足够了, 而 $\alpha_{\text{Overlap-min}}$ 一般为 $2\alpha_{\text{Area}}$, 常数 τ 一般取 1.5 到 5 之间的常数。

权系数 $\alpha_{\text{AspectRatio}}$, $\alpha_{\text{Separation}}$ 和 $\alpha_{\text{Orientation}}$ 是为了消除版图中的失配效应而引入的, 它们的作用与 α_{Overlap} 有些相似, 当相应的约束条件满足时, 对应的目标函数就为零, 否则, 就希望迭代改进过程能迅速消除那些不满足约束的版图因素, 当然, 它们与 α_{Overlap} 的不同之处在于这类约束不影响版图的可制造性。因此, 这三个权系数也应取相当大的值, 以保证当 $C_{\text{Overlap}}(X)$ 为零时, 这些项在总目标函数中起主要作用。在实际应用中, 这三个权系数一般取 $300\alpha_{\text{Area}}$ 至 $3\,000\alpha_{\text{Area}}$ 。

权系数 α_{Area} , α_{Length} 和 α_{Merge} 与评估布局好坏的三个基本指标相关, 它们将直接影响布局结果的质量。作为最理想的选择, 应当使它们与各自对应的目标函数(由最终布局结果来衡量)的乘积为 1, 但遗憾的是我们无法预知最终布局结果。许多实验表明: $C_{\text{Area}}(X)/C_{\text{Length}}(X)$ 通常在 2 至 4 之间^[69]。因此, 选择 $\alpha_{\text{Length}} = 3\alpha_{\text{Area}}$ 在大多数情况下都是很不错的。然而, $C_{\text{Area}}(X)/C_{\text{Merge}}(X)$ 的值却比较分散, 选择 $\alpha_{\text{Merge}} = \alpha_{\text{Area}}$ 是最直接、最简单的办法。J. M. Cohn 曾对选择权系数 α_{Area} , α_{Length} 和 α_{Merge} 进行过数百次实验^[69], 用选择不同的权系数组合来测试了数百个电路, 结果发现不同的权系数组合虽然对计算效率和最终的布局结果都有影响, 但是, 在最终结果中, $C_{\text{Area}}(X)$, $C_{\text{Length}}(X)$ 和 $C_{\text{Merge}}(X)$ 的离散性并不大。换言之, 尽管不同的权系数组合会得到不同的布局结果, 但采用模拟退火法对布局进行迭代改进后, 总能获得一个相当不错的布局结果。因此, 选择 $\alpha_{\text{Length}} = 3\alpha_{\text{Area}} = 3\alpha_{\text{Merge}}$ 是切实可行的。

6.4 器件级布线方法

模拟集成电路的器件级布线法与数字集成电路不同, 需要考虑对称布线、减小连线串扰等许多特殊的因素。考虑到模拟集成电路的布局都相当紧凑, 布线空间通常是比较拥挤的, 如何提高算法的布线成功率是模拟集成电路布线算法的核心问题。在通常情况下, 仅仅依靠简单的策略来确定布线顺序是无济于事的, 实际应用表明: 使布线器具有重布线(Rip-up)能力是必要的^[70]。本节将首先扼要地分析迷宫算法、线探法和线扩展法的特点, 阐述线扩展法更适合于模拟集成电路布线问题的缘由; 然后介绍布线空间的表示、数据结构、路径的表示与扩展等关于线扩展法的基本思想; 最后介绍如何改进基本的线扩展法, 使之具有重布线、对称布线和减小连线串扰等模拟集成电路版图设计所需要的功能。

6.4.1 迷宫算法、线探法与线扩展法

迷宫算法(Maze Routing)和线搜索法(Line-Search Routing)是两类最基本的平面布线算法。基于迷宫算法的布线器用等间距的网格将布线空间分为若干大小相等的格子,

并用一系列相互邻接的格子来表示连线路径。布线空间就好像一个迷宫，只有空格子是可以使用的，非空格子(被器件或连线占用)是不能布线的区域。迷宫算法从一个连线端口(简称源单元)出发，按广度优先或价格优先的准则搜索空格子，当搜索到另一个连线端口(简称目标单元)时，则找到了连接源单元和目标单元的路径，从而嵌入一条连线，当电路中的所有连线都被嵌入后或无法布通某条线网时就停止。迷宫算法的优点是能找到最经济的连线，而且，只要源单元和目标单元之间存在一条路径，迷宫算法就一定能成功地找到这条路径，并完成布线任务。

线搜索法用一组互联的线段(而不是邻接的格子)来表示连线路径。由于一条线段可以表示许多格子，因此，线搜索法比迷宫算法更节省内存，当然在大多数情况还能节省计算时间。线探法(Line-Probe Routing)^[188]是最为有名的线搜索法。线探法首先从源单元和目标单元处扩展最长的逸出线(Escape Line)，又称探测线，然后继续从已有的探测线扩展一些与已有探测线垂直的新探测线，当发源自源单元的探测线与发源自目标单元的探测线相交时，则获得了一条连线路径。图 6.18 是线探法的示意图，其中 A 和 B 分别表示源单元和目标单元。线探法属于无网格布线法，可以比较灵活地处理对称布线、连线串扰消除等模拟集成电路版图所关心的问题。尽管线探法所获得的连线更直一些(转折点少)，但所获得的连线一般不是最经济的，更有甚者，当源单元和目标单元之间存在路径时，线探法有时却不能成功地“探测”出该路径。因此，线探法比较适合于布局稀疏、有许多空余布线空间的情形；当布局比较紧凑、连线十分拥挤时，则迷宫算法更合适。

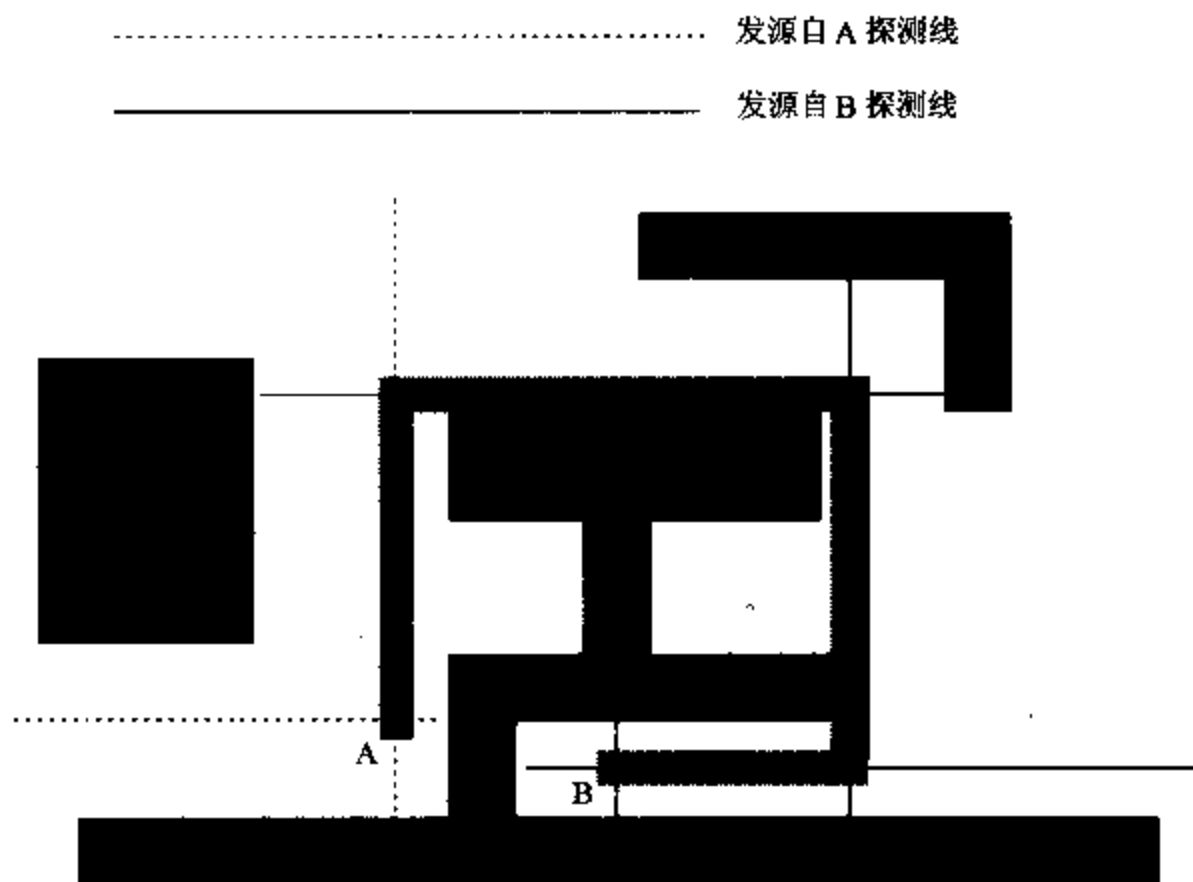


图 6.18 线探法示意图

线扩展法(Line-Expansion Routing)^[189]则是将迷宫算法和线探法的优点结合起来，用一组线段来表示一条路径(与线探法类似)，在搜索新路径时采用布线成本优先的策略(与迷宫算法类似)，但又与迷宫算法不同，线扩展法是从泛路径(Partial Path)开始搜索，而不是从网格开始搜索。所谓泛路径是从源单元出发，并终止于布线空间中某一单元的一组互联线段的集合。基于线扩展法的布线系统为被连接的线网保存一组泛路径，通过不断扩展这些泛路径直到某条泛路径达到目标单元为止。在泛路径扩展时，总是向

布线成本最低的泛路径中加入一些线段,从而形成一些新的泛路径。线探法和线扩展法的主要区别在于:前者希望路径中的线段越长越好,只有当遇到障碍时才转折;而线扩展法中构成路径的线段可以在任何有利于减小布线成本的地方转折。因此,线扩展法在具备处理对称布线和消除连线串扰等功能的同时,又继承了迷宫算法的优越性——布线成本最低,而且只要连线通路存在,布线就一定能够成功。线扩展法最适合于解决布局紧凑的模拟集成电路的布线问题。

6.4.2 线扩展法的基本思想

从前面关于线扩展法的讨论不难看出,布线空间和泛路径的表示、泛路径的选择与扩展策略是线扩展法的基本要素。由于线扩展法、线探法和迷宫算法都是串行布线法,即:每次处理电路的一条或一段线网,直到所有连线都嵌入后才结束;布线顺序不仅会影响布线速度,还同时影响整体布线质量,在许多情况下还会因为布线顺序选择不妥而造成布线失败的局面,因此,如何根据模拟集成电路的版图特征确定恰当的布线顺序也是值得讨论的。

1. 角勾链数据结构与布线空间的表示方法

对于模拟集成电路的器件级布线器而言,不可能采用先进行粗略布线,然后再进行后期压缩和优化改进的方式,因为版图压缩过程会破坏布局和布线的对称性,这也是模拟集成电路的版图自动化设计工具必须在布局阶段就采用紧密布局的原因。虽然近年出现了一些能处理对称布线的版图压缩工具^[190],但没有任何工具能处理消除连线串扰的问题。因此,必须精细地描述布线空间。Jhon K. Ousterhout 提出的角勾链(Corner Stitching)数据结构^[191]能很好地处理该问题,Ousterhout 用一些矩形瓷砖(Tile)来表示布线空间中的区域。E类瓷砖(Empty Tiles)代表可布线的区域;F类瓷砖(Filled Tiles)代表被器件或已经嵌入的连线所占据的区域,是不可利用的。角勾链使布局空间中的瓷砖形成有序二维链表,具有下述优越性:

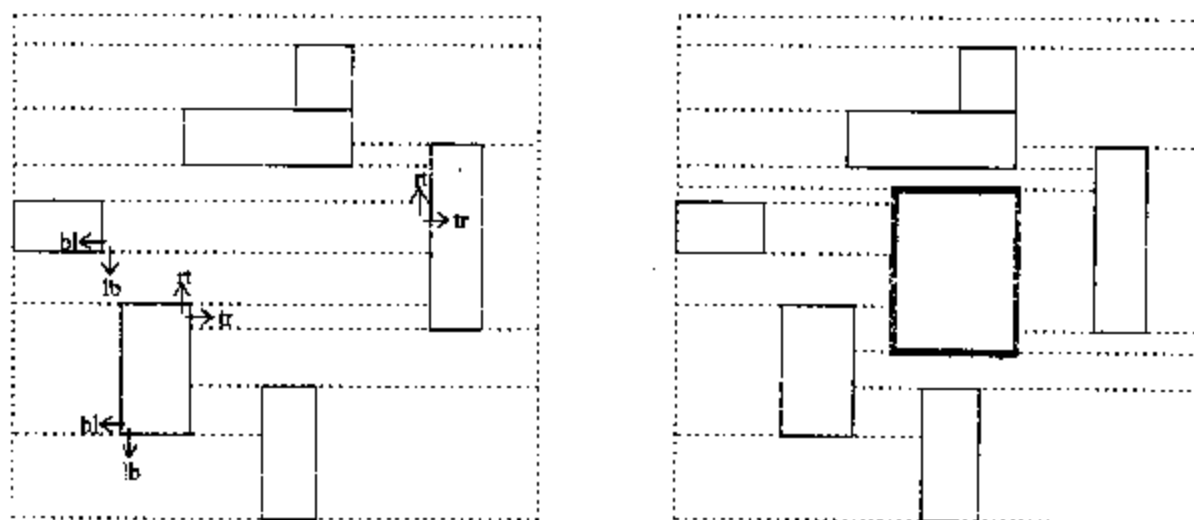
(1) 能快速定位包含某给定点的瓷砖,能快速访问任何瓷砖的邻接瓷砖,这些特性非常有利于查询版图的局部几何信息——有无可利用的布线区域、线网的间距,从而保证布线满足设计规则、快速估计连线串扰;

(2) 能支持几何区域的合并、分解、插入和删除,这些特性对查找可利用的布线空间、实现对称布线非常有用;

(3) 瓷砖划分采用最长、最宽且水平方向优先的原则,可用最少的瓷砖来精确描述布线空间,消除了迷宫算法计算效率低和存储空间大的弊端。

图 6.19(a)是布线空间中瓷砖划分的示意图,其中边框为实线的瓷砖表示 F 类瓷砖(共 6 个),边框为虚线的瓷砖表示 E 类瓷砖(共 15 个),图 6.19(a)还示意性地画出了两个瓷砖的角勾链指针 rt, tr, lb 和 bl。在布局空间中插入、删除 F 类型的瓷砖后,角勾链数据结构也应作相应的变化,具体算法请读者参阅文献^[191]。图 6.19(b)是在图 6.19(a)中增加一个 F 类型的瓷砖(用粗线框表示)后,布局空间中瓷砖的变化情况(共 7 个 F 类瓷砖和 18 个 E 类瓷砖),但为简便起见,图中省略了角勾链数据结构的变化。

由于许多版图采用多层布线,因此每一层掩模都对应一个角勾链数据结构;为提高



(a) 瓷砖的划分及角勾链指针 (b) 增加 F 类型的瓷砖后布局空间的变化情况

图 6.19 布线空间的瓷砖表示法

布线效率,某些层还需要按最长、最宽且垂直方向优先的原则进行瓷砖划分。另外,为了支持在器件上布线,器件上所有可布线区域都是 E 类瓷砖。

2. 布线单元与泛路径的表示方法

基于线扩展法的布线系统采用线段的链表来表示连线路径,由于线段没有宽度,不适合于无网格的模拟集成电路布线问题,为此,我们引入一个称为布线单元的数据结构,它描述一个特定掩模层(连线层)的矩形区域(即布线空间中的瓷砖),并用布线单元的链表来表示泛路径,则可以解决该问题。布线单元包含矩形区域的左下角坐标 (x_l, y_l) 和右上角坐标 (x_r, y_r) 、该布线单元自身的布线成本 C_{wire} 、到最邻近单元的最小布线成本的估计值 C_{Target} 、所在的掩模层 Layer 以及指向该泛路径中上一个布线单元的指针 Last Cell。需要特别说明的是:一个布线单元可能同时属于多条泛路径。图 6.20 是一个包含 5 条泛路径的示意图。

布线器总是从一组源单元中选出一个布线单元,接着试图朝所有可布线的方向或别的布线层进行扩展,扩展过程就是搜索邻近的布线单元,让它们指向该源单元以形成一组泛路径,然后不断重复地选择一条泛路径(不一定是源单元)进行扩展以形成新的泛路径。因此,泛路径的扩展过程将形成一组树,每棵树的根节点就代表一个源单元,它实质是泛路径的链尾,而每一个叶子节点实质是一条泛路径的链头,叶子节点记录了所有可行的连线通路。当某叶子节点到达连线的目标单元时,则找到了一条完整的连线路径。

布线单元中描述矩形区域的信息来自它所对应的布线空间中的瓷砖。布线单元还包含两项与布线价格相关的布线成本信息,其中 C_{wire} 的计算非常简单,是该矩形区域的面积与它所在布线层的单位面积的布线成本之积,而 C_{Target} 的计算则稍微复杂一些。假设当前的布线单元为 $T_{Routing}$,与之邻近的单元为 T_{Target} ,则它们之间的布线成本的估计值为:

$$C_{Estimate}(T_{Routing}, T_{Target}) = W_{Routing} [L_{Direct}(T_{Routing}, T_{Target}) \times C_{Layer}(T_{Routing} \rightarrow Layer) + C_{Contact}(T_{Routing} \rightarrow Layer, T_{Target} \rightarrow Layer)]$$

其中, $W_{Routing}$ 是连线宽度, $T_{Routing} \rightarrow Layer$ 和 $T_{Target} \rightarrow Layer$ 分别是 $T_{Routing}$ 和 T_{Target} 所在的连线层, $L_{Direct}(T_1, T_2)$ 是布线单元 T_1 和 T_2 之间的直接距离(不考虑它们所对应的瓷砖之

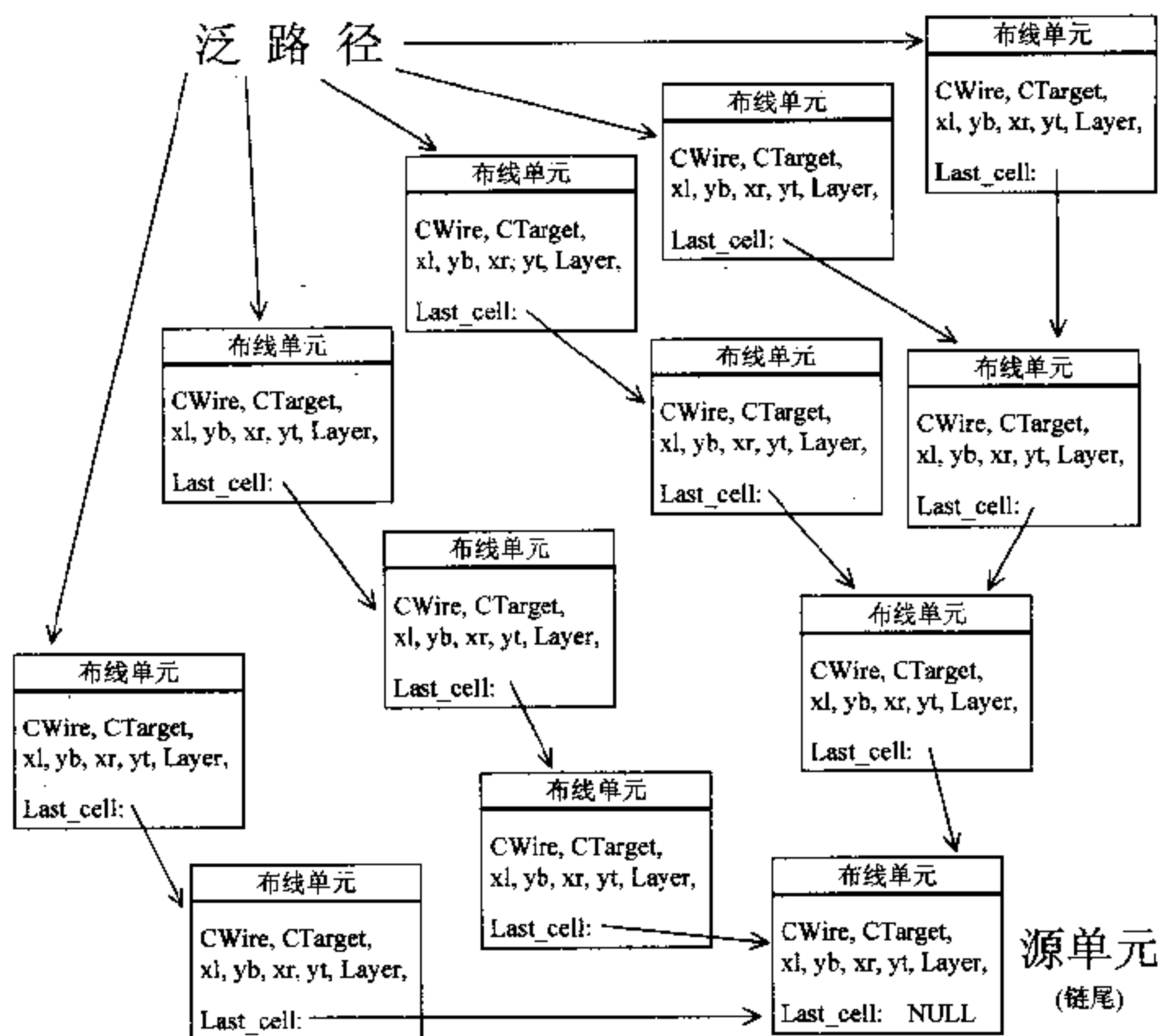


图 6.20 泛路径的表示方法

间有无 F 类型的瓷砖), $C_{l,ayer}(l)$ 是第 l 层单位面积的布线成本, $C_{Contact}(l, k)$ 是第 l 层布线和第 k 层布线之间单位宽度的接触孔的布线成本, 很显然, $C_{Contact}(l, l) = 0$ 。因此, 如果布线单元 $T_{Routing}$ 的邻近布线单元的集合为 $T_{Neighbors}$, 则其 C_{Target} 为:

$$C_{Target} = \min_{Tile \in T_{Neighbors}} C_{Estimate}(T_{Routing}, Tile) \quad (6.17)$$

在基于线扩展法的布线系统中, 如果需要在支持布线成本优先的泛路径扩展策略, 则估计泛路径的布线成本是必要条件。如果将布线单元 $p \rightarrow head$ 加入到泛路径 $p \rightarrow last$ 后形成新的泛路径 p , 则泛路径 p 的布线成本 $C_{Path}(p)$ 为:

$$C_{Path}(p) = \sum_{r \in p} C_{Wire}(r) = C_{Wire}(p \rightarrow head) + C_{Path}(p \rightarrow last) \quad (6.18)$$

其中, r 是构成泛路径 p 的布线单元, $C_{Wire}(r)$ 是布线单元 r 的布线成本。

3. 泛路径的扩展

如何从已有的泛路径中选择一条待扩展泛路径, 并由此扩展出一系列新的泛路径是线扩展法的核心内容。

选择待扩展泛路径的主要策略有深度优先、广度优先和布线成本优先等三种策略。深度优先策略总是选择最新生成的泛路径作为扩展对象, 当该泛路径不能继续扩展时, 则选择该布线单元的父单元作为扩展对象。深度优先策略将搜索许多长而无用的泛路径,

并忽略那些较短的路径,获得的连线路径最长;其另一主要缺陷是没有考虑布线成本,因此不适合于布线问题。广度优先策略则不然,当生成新的泛路径时,将相应的单元附加到一个先入先出(FIFO)的队列尾,然后选择位于对列头的布线单元进行扩展,广度优先策略本质上是李氏迷宫算法^[192]。广度优先策略不会搜索那些长而无用的路径,所形成的连线路径最短,用队列来实现的效率也很高,但是,由于没有考虑布线成本,因而也不适合于模拟集成电路的布线问题。

布线成本优先的策略则不然,该策略在选择扩展泛路径时总是选择布线成本最低者进行扩展,但随着扩展过程的进行,每条泛路径的布线成本都在发生变化,而扩展过程通常会产生数十万条泛路径,因此,如何快速查找布线成本最低的泛路径是很重要的。由于堆的排序与查找效率非常高,可以采用简单的二岔堆排序^[106]来解决该问题。我们让二岔堆中的每一个节点指向一个布线单元(泛路径的链头),这些节点按它所代表的泛路径的布线成本进行排序,堆顶(又称根节点)指向布线成本最低的泛路径。因此,布线器只要先将所有源单元插入到堆中,每次扩展时均从堆顶将布线成本最低的泛路径删除(删除过程将使别的布线成本最低的泛路径成为新的堆顶),将它作为扩展对象,并接着搜索其邻近的布线单元以形成新的泛路径(新的泛路径的成本是其链头单元的布线成本与原泛路径的布线成本之和),然后再将所有新生成的泛路径插入到堆中(插入过程会对堆中的元素重新进行排序,并使布线成本最低的泛路径位于堆顶)。在扩展过程中,所有源单元都将逐步发展为一棵一棵的树,树的叶子节点代表所有的泛路径,二岔堆中的节点和这些叶子节点之间是一一对应的。

4. 线网的嵌入顺序

对于串行布线器而言,由于每次只能嵌入一条连线,先嵌入的线网将可能阻挡后面的线网,从而造成布线失败,其根本原因在于串行布线器缺乏“先见之明”。先嵌入那些容易被阻挡的连线是解决该问题的基本思路。对于模拟集成电路而言,具有对称性约束的线网可选择的泛路径相对较少,因此容易被先嵌入的线网阻挡。需要消除连线串扰的线网也是如此。因此,应当在嵌入所有具有对称性约束的线网后,再嵌入那些没有对称性约束的线网;如果线网的对称性要求相当,则需要消除连线串扰的线网应当优先嵌入;如果对称性要求和消除连线串扰的要求都相当,应当优先嵌入最短生成树较短的线网,因为这类线网更容易被阻挡。

上述关于选择线网嵌入顺序的策略有助于提高布线成功率,但对于布局非常紧凑的模拟集成电路而言,单靠上述启发式策略是不够的,还必须使布线器具有较强的重布线能力,我们将在第 6.4.3 节中详细讨论。

5. 接触孔与连线层的更换

对于许多模拟集成电路而言,由于布局紧凑、连线拥挤、而且还必须考虑对称性和串扰消除等因素,单一连线层通常是无法布通的,必须采用多层布线。用线扩展法处理多层布线问题时,泛路径扩展过程不仅需要在同一层布线空间中进行扩展,还需要在已有的泛路径上加入接触孔以形成新的泛路径。由于接触孔一般是方的,因此估计接触孔的布线成本非常简单。但是,如果接触孔的边长大于连线的宽度,则不同的对齐方式将

产生不同的泛路径。如果待扩展泛路径的链头是接触孔,则需要产生 4 条新的泛路径,分别由位于另一布线层、且与该接触孔的 4 条边邻接的布线单元所生成。

6.4.3 重布线策略

前面介绍的布线顺序虽然有助于提高布线成功率,但是,对于布局紧凑、对称性要求较高、且需要消除串扰的模拟集成电路而言,几乎没有任何既定的布线顺序能保证布线一定成功。因此,在发现当前线网被已嵌入的线网所阻挡时,应当采用适当的事后防范措施。最有效(或许是唯一成功的)措施是废弃一些已经嵌入的线网,在完成当前线网的嵌入后,再重新嵌入这些被废弃的线网,我们称之为重布线(Rip-up Routing 或 Rerouting)。重布线策略虽然给人以亡羊补牢的感觉,但对于器件级模拟集成电路的布线器而言,这是必须采用的策略。

重布线策略面临的首要问题是如何选择被废弃的线网,如果我们每次只废弃一条已嵌入的线网,则可能因为缺乏总体考虑而影响布线质量,更有甚者会出现无限循环的局面。例如:当嵌入线网 B 时需要废弃已嵌入的线网 A,在重新嵌入线网 A 时又需要废弃已嵌入的线网 B,从而无限循环下去。人们虽然提出了一些废弃线网的随机选择策略^[193],但对于模拟集成电路而言,由于布线空间非常拥挤,可选择的线网数目很少,随机选择策略没有什么作用^[70]。在许多情况下,需要废弃多条已嵌入的线网,否则,不可能对改进布线成功率产生显著作用。

在选择被废弃的线网时最理想的原则是:被废弃的线网应尽可能少,但同时又能为被阻挡的线网提供足够的布线空间。Excellerator 系统^[194]中采用了一种非常精致的策略。Excellerator 在对泛路径进行扩展的过程中,当发现新生成的泛路径 p 与已经嵌入的线网 $\mathcal{N}_{\text{Embedded}}$ 之间违反了设计规则时,并不立即拒绝该泛路径,而是将该泛路径的布线成本加大 $C_{\text{Rip-up}}(\mathcal{N}_{\text{Embedded}})$, $C_{\text{Rip-up}}(\mathcal{N}_{\text{Embedded}})$ 反映了废弃并重新嵌入线网 $\mathcal{N}_{\text{Embedded}}$ 的成本,由重新嵌入线网 $\mathcal{N}_{\text{Embedded}}$ 时所增加的连线长度决定。 $C_{\text{Rip-up}}(\mathcal{N}_{\text{Embedded}})$ 实质上反映了废弃 $\mathcal{N}_{\text{Embedded}}$ 之后再重新嵌入时因绕过 p 所增加的布线成本。Excellerator 的策略使得新嵌入的线网可以在采用较短的路径(需废弃已嵌入的线网)和绕过障碍物(自身的线网长度较大)之间作适当的抉择,有助于提高布线成功率。但是,Excellerator 仍然不能避免反复循环的尴尬局面,因而不得不限制每条线网最多只允许被废弃 2 次。

ANAGRAM-II 采用了一种更简单的策略^[69]:任何线网 $\mathcal{N}$ 都有一个重布线成本 $C_{\text{Rip-up}}(\mathcal{N})$,每当线网 $\mathcal{N}$ 被废弃时,并不计算因重布线所增加的连线长度,而是简单地将 $C_{\text{Rip-up}}(\mathcal{N})$ 加大一个固定的值 C_R ,这不仅减少了计算量,而且使 $C_{\text{Rip-up}}(\mathcal{N})$ 随线网 $\mathcal{N}$ 被放弃次数的增加而增加,当循环次数较多时, $C_{\text{Rip-up}}(\mathcal{N})$ 将非常大,迫使新嵌入的线网选择绕过障碍物的路径,而不是废弃已经嵌入的线网,从而有效防止因重布线所导致的无限循环。

当采用上述重布线策略后,只要对式(6.18)所示的布线成本作适当的修订,就可以使泛路径的扩展过程能自动处理重布线问题,从而使线扩展法具有重布线能力。我们假设将布线单元 $p \rightarrow \text{head}$ 加入到泛路径 $p \rightarrow \text{last}$ 后形成新的泛路径 p ,且嵌入泛路径 $p \rightarrow \text{last}$ 和 p 时需要废弃并重新嵌入的线网集合分别为 $\mathcal{N}_{\text{Rip-up}}(p \rightarrow \text{last})$ 和 $\mathcal{N}_{\text{Rip-up}}(p)$,则泛路径 p 的布线成本为:

$$C_{\text{Path}}(p) = C_{\text{Wire}}(p \rightarrow \text{head}) + C_{\text{Path}}(p \rightarrow \text{last}) + \sum_{\mathcal{N} \in \mathcal{V}_{\text{Rip-up}}(p) - \mathcal{V}_{\text{Rip-up}}(p \rightarrow \text{last})} C_{\text{Rip-up}}(\mathcal{N}) \quad (6.19)$$

在采用上述重布线策略后, C_R 将对布线质量产生直接的影响。如果我们让废弃已嵌入线网的过程发生在泛路径的扩展过程中(而不是嵌入新线网之后), 则一条线网可以被反复嵌入多次, 且每次嵌入时布线环境都有所变化, 这种迭代改进过程消除了简单串行布线器的弊端, 有助于提高布线质量。

由于重布线时有可能废弃一条或多条已嵌入的线网, 何时再嵌入这些被废弃线网是影响布线质量和计算量的关键因素。图 6.21 给出了一种典型的重布线策略, 线网嵌入队列的初始状态是由简单的线网嵌入策略所决定的(参见第 6.4.2 节), 布线器每次从队头取出一条线网, 然后试图扩展其泛路径, 并将泛路径扩展过程中废弃的线网插入到队列中, 如此往复, 直到线网嵌入队列为空时才停止。

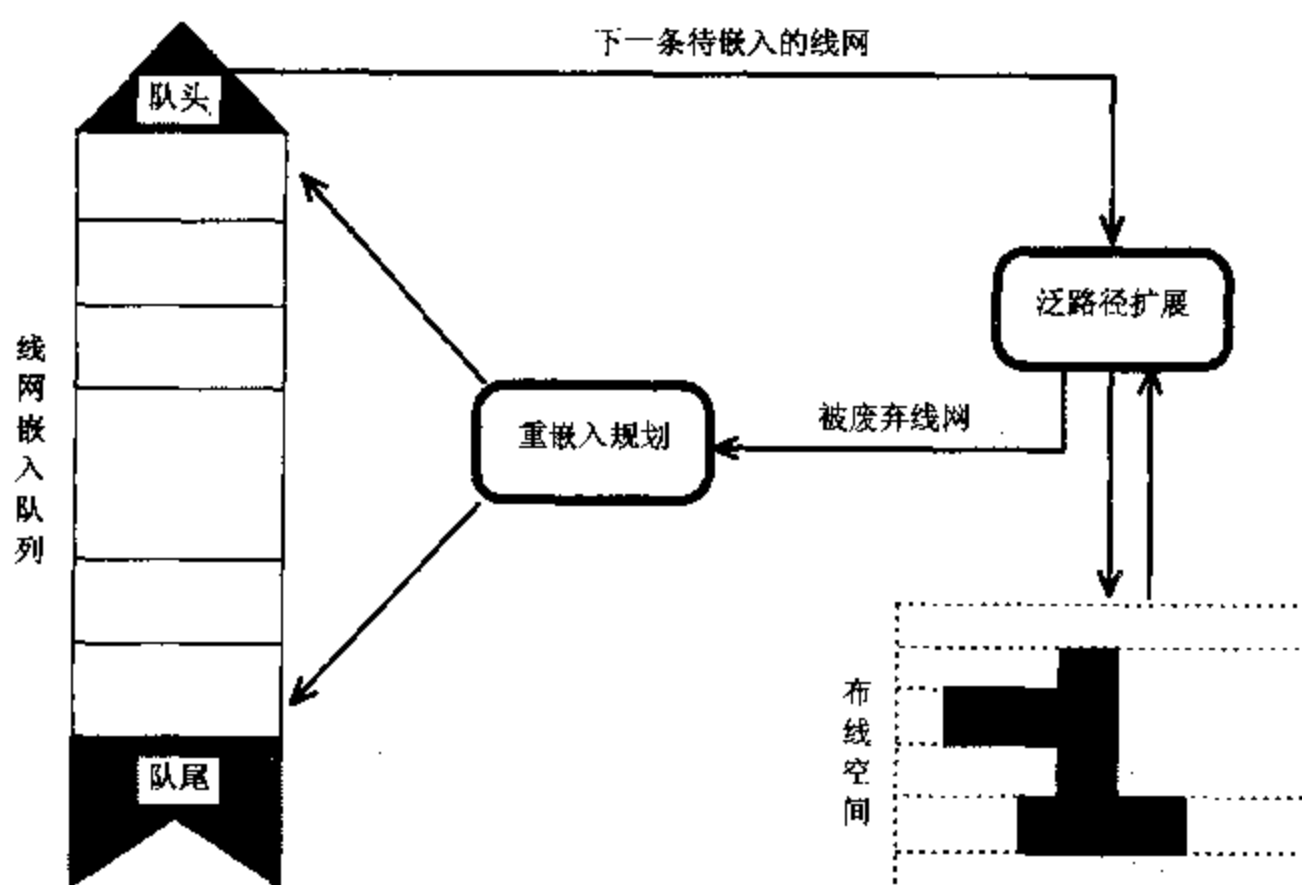


图 6.21 确定线网嵌入顺序的示意图

在图 6.21 所示的重布线策略中, 泛路径扩展模块将被废弃的线网传递给重嵌入规划模块, 然后由重嵌入规划模块确定被废弃线网的重嵌入顺序。如果只是简单地将废弃线网附加到线网嵌入队列的队尾, 则初始的线网嵌入顺序对布线过程的影响非常小, 不能满足模拟集成电路对对称性和消除连线串扰的要求。考虑到电路中关键线网的长度对电路性能的影响很大, 布线器应当使这些线网尽可能短, 而前述简单的重嵌入策略显然不能满足这一要求。为了解决该问题, 我们可以为线网 $\mathcal{N}$ 引入一个嵌入优先级 $\text{EmbeddingPriority}(\mathcal{N})$, 其取值范围在 1 到 10 之间。在初始化线网嵌入队列时, 嵌入优先级较大的线网靠近队头; 当已嵌入的线网 $\mathcal{N}$ 被废弃时, 其重布线成本不是简单地增加 C_R , 而是增加 $C_R \times \text{EmbeddingPriority}(\mathcal{N})$ 。当然, 为了确保先嵌入关键线网、具有对称性的线网和需要消除连线串扰的线网, 可以让它们的优先级大于特优值 P_s , 并为这类线网单独分配一个特殊队列, 只有当特殊队列为空时, 布线器才开始嵌入优先级低于 P_s 的普通队列。重嵌入规划模块也总是将优先级高于 P_s 的线网附加到特殊队列的队尾, 将优先

级低于或等于 P_s 的线网附加到普通队列的队尾。在实际应用中 P_s 一般取 7 或 8。引入嵌入优先级后,增加了用户对布线器的控制,为了保证布线的总体质量和成功率,优先级大于 P_s 的线网不应过多。

6.4.4 对称布线

由于受热效应、工艺涨落和版图寄生效应的影响,除了采取对称布局措施之外,使某些线网具有对称性是提高版图质量的必要手段。

不失一般性,我们假定对称轴位于布线空间中 $x=0$ 处(即布线空间的纵轴),我们在后面将会看到,该假设并不要求改变布线空间的描述方式,也不需要布线空间附加任何额外的约束。

在布线系统中,对称线网实质是一对线网($\mathcal{N}_{\text{Left}}, \mathcal{N}_{\text{Right}}$),其中,线网 $\mathcal{N}_{\text{Left}}$ 的所有联接端点都位于对称轴的左边,这些端点关于对称轴的镜象对称点则构成线网 $\mathcal{N}_{\text{Right}}$ 的所有联接端点。对称布线的任务是使对称线网的所有连线都关于对称轴形成左右镜象对称。由于对称轴左右可利用的布线区域是不同的,因此不能简单地先嵌入一条线网,然后再作镜象映射以获得另一条线网。

其实,我们只要将 $\mathcal{N}_{\text{Left}}$ 和 $\mathcal{N}_{\text{Right}}$ 作为一个整体,并对前面介绍的泛路径扩展过程稍作改动即可。在线扩展过程中,我们依然按前面的方法对线网 $\mathcal{N}_{\text{Left}}$ 的泛路径进行扩展,当加入布线单元以形成新的泛路径时,只有当该布线单元及其关于对称轴的镜象对称单元都合法时,才形成新的泛路径。图 6.22 是对称线网的泛路径扩展示意图。当需要将位于第 $Layer$ 层、矩形区域为 $\{(-x, y), (-X, Y)\}$ 的扩展单元加入右边线网的泛路径时,不仅需要检查该单元的合法性,同时还检查其镜象对称单元(位于第 $Layer$ 层、矩形区域

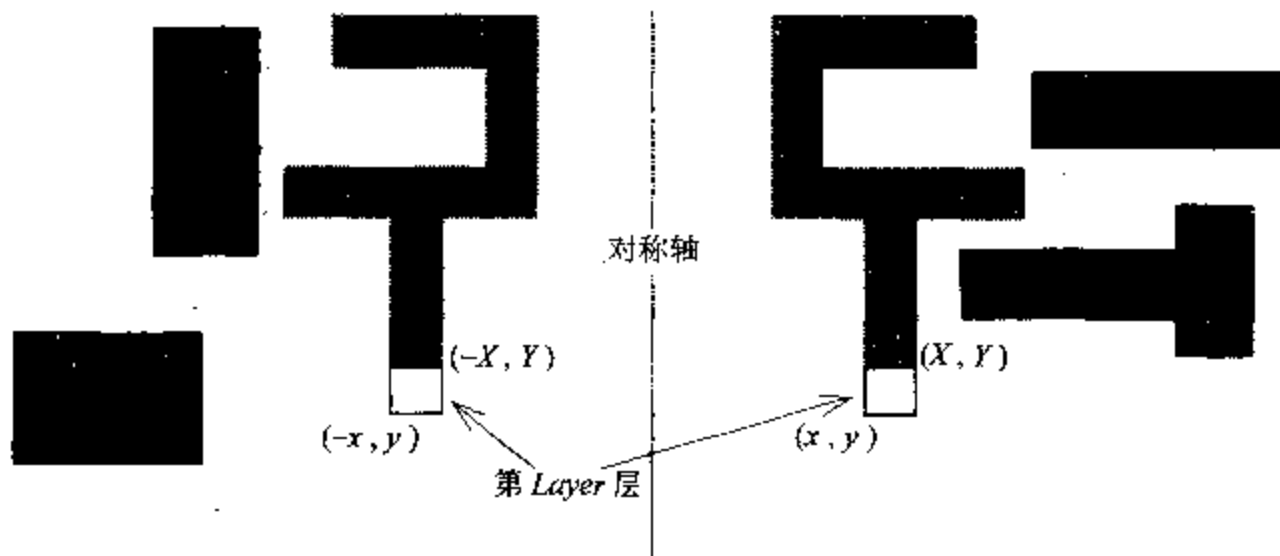


图 6.22 对称线网的泛路径扩展示意图

为 $\{(x, y), (X, Y)\}$ 的合法性,只要其中有一个不合法,就拒绝生成新的泛路径。需要注意的是:合法性检查不单是线网是否违反几何规则,还必须考虑线网离对称轴过近时,对称线网会相互交叠,从而违反电规则。因此,还要求扩展单元满足下述约束条件:

$$x > \text{MinimumSpace}[Layer][Layer]/2 \quad (6.20)$$

其中, $\text{MinimumSpace}[Layer][Layer]$ 是第 $Layer$ 层线网的最小间距。

上述泛路径扩展策略继承了前面介绍的简单泛路径扩展策略(每次只处理一条线网)的全部优越性,因此,只要对称布线是可能的,就一定可以完成对称线网的嵌入,并能

保证布线成本依然是最低的。

另外，还需要对重布线策略作简单修改。如果某线网需要废弃对称线网中的一条线网，则同时废弃这一对线网，然后再由重嵌入规划模块决定该对称线网的重嵌入顺序，从而确保重布线策略不会改变线网的对称性。

应当看到，如果我们让电路中的一组信号线（例如两个差分输入信号）具有对称性，则嵌入这些线网后，可利用的布线空间也具有一定的对称性，从而使电源线（电源线的联接端点通常具有对称性或部分对称性）或别的线网也可以具有对称性，这对提高版图的性能是非常有益的。

在模拟集成电路中，常常还要求某些线网具有自对称性——其连线端点对称地分布在对称轴的左右。在差分电路、偏置电路和时钟发生电路中，让某些关键线网具有自对称性是平衡驱动能力和负载特性的有效措施。嵌入自对称线网的过程非常简单，通常分两步进行：第一步，将自对称线网拆成两个子线网，对称轴左边的节点为一条线网，对称轴右边的线网为另一条线网，根据自对称性要求，这两条子线网构成对称线网，因此，可以采用前面的对称布线法嵌入这两条子线网；第二步，从这两条子线网上选一对对称点 T_{Left} 和 T_{Right} （不一定是连线端点），采用对称布线技术嵌入连接 T_{Left} 和 T_{Right} 的线网。不言而喻，在第二步中不能使用式(6.20)所给定的约束。图 6.23 是自对称线网的实例。

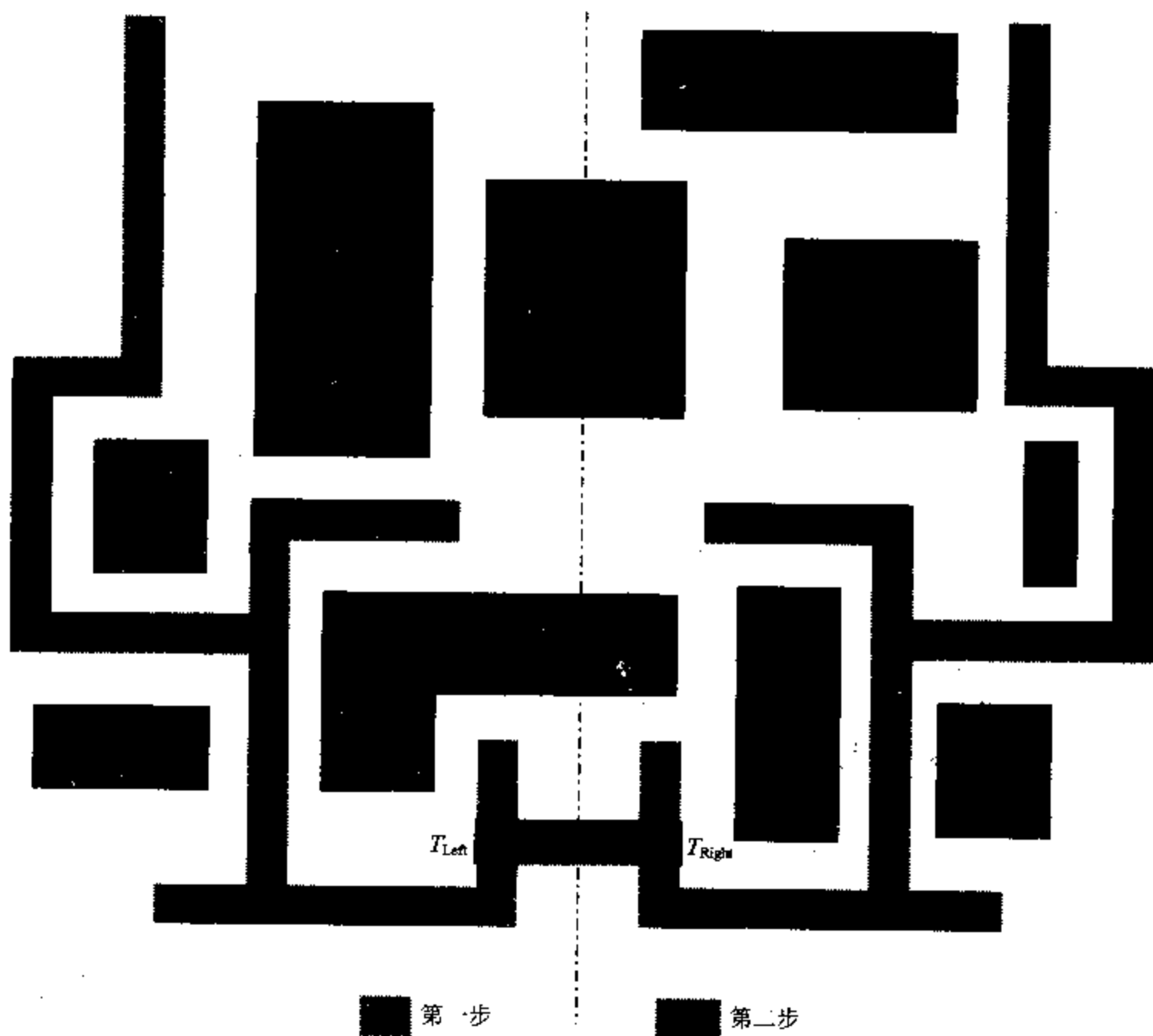


图 6.23 自对称线网示意图

6.4.5 消除连线串扰

模拟集成电路中的连线一般都不能作为理想的孤立导体来处理,连线的几何形状及其周围环境是决定节点(线网)之间的寄生电容的重要因素,这些寄生电容对电路性能有较大的影响。我们在第6.3.2节和第6.4.4节介绍了采用对称布局和布线技术使匹配器件具有完全相同的寄生电容和寄生电阻,使电源线网、偏置电路中的某些关键线网平衡分布,这些措施并不能减少版图中的寄生电容,因而只对具有对称性的电路有作用。然而,模拟电路中的某些节点(例如高增益电路的输入节点和存储电荷的节点)通常对寄生电容所产生的耦合噪声信号非常敏感,这类节点所对应的线网称为敏感线网;类似地,数据采样电路中的时钟和高增益电路的输出节点是产生噪声信号的主要节点,它们所对应的线网称为噪声线网;而敏感线网和噪声线网之间的寄生电容则称为耦合电容。为了提高模拟集成电路和数模混合集成电路的性能,减少耦合电容是消除连线串扰的主要手段,而减小敏感线网和噪声线网之间的交叠电容和平行电容(参见图6.4(a))、采用屏蔽导体(参见图6.4(b))是在布线阶段减少噪声耦合效应的具体措施。

为了使模拟集成电路布线器能有效处理连线串扰问题,我们将电路中的线网分为三类:中性线网、互作用线网和屏蔽线网。中性线网不需要考虑连线串扰效应,只要不违反设计规则,这类线网可以嵌入在任何环境中,但不能用作屏蔽导体。电路中的绝大部分线网都属于中性线网。如果需要在布线阶段尽可能减少两个节点之间的寄生电容,则这两个节点所对应的线网称为互作用线网。电源、地以及电路中的某些低阻节点(例如参考电压源的输出节点)所对应的线网可以用来屏蔽互作用线网之间的耦合效应,称为屏蔽线网。在实际应用中,用户只需指定屏蔽线网和互作用线网即可,凡是未指定的线网都作为中性线网处理。

1. 耦合电容的简化模型

在处理对称布线问题时可以简单地改变泛路径扩展策略,使之拒绝非对称的布线结构,但是,对于串扰问题而言,由于耦合电容在许多情况下是不可能完全消除的,因此,必须引入度量耦合效应强弱的模型,耦合效应强的线网具有较大的布线成本,而耦合效应弱的线网具有较小的布线成本,从而使线扩展法能够选择耦合效应弱的线网结构。

由于耦合效应的强弱与耦合电容成正比,因此,如何计算耦合电容是必须解决的首要环节。考虑到泛路径扩展过程需要反复计算泛路径的布线成本,即使是中等精度的定量模型^[195],其计算量也非常大,所以,采用定性简化模型是非常必要的。对于布线问题而言,定性简化模型已经足够了,因为布线器的目的是获得耦合电容最小的布线结果,至于耦合电容的具体数值并不重要。

交叠电容的估计非常简单,可以采用平板电容模型——电容值与交叠面积成正比,因此,布线单元 T_{Sen} 和 T_{Noisy} 之间的交叠电容为:

$$C_{\text{Ovp}}(T_{\text{Sen}}, T_{\text{Noisy}}) = A_{\text{Unshield}}(T_{\text{Sen}}, T_{\text{Noisy}}) C_{\text{UnitOvp}}[T_{\text{Sen}} \rightarrow \text{Layer}][T_{\text{Noisy}} \rightarrow \text{Layer}] \quad (6.21)$$

其中, $A_{\text{Unshield}}(T_{\text{Sen}}, T_{\text{Noisy}})$ 是布线单元 T_{Sen} 和 T_{Noisy} 之间的未被屏蔽的交叠面积(参见图6.24), $T_{\text{Sen}} \rightarrow \text{Layer}$ 和 $T_{\text{Noisy}} \rightarrow \text{Layer}$ 分别代表布线单元 T_{Sen} 和 T_{Noisy} 所在的连线层,而

$C_{\text{UnitOvlp}}[i][j]$ 则是第 i 层连线和第 j 层连线间单位交叠面积的电容值。

对于平行电容，我们仍然采用平板电容模型，并且只考虑距离小于 d_{max} 的平行连线之间的电容，不考虑任何边缘效应。因此，布线单元 T_{Sen} 和 T_{Noisy} 之间的平行电容为：

$$C_{\text{Parallel}}(T_{\text{Sen}}, T_{\text{Noisy}}) = \frac{L_{\text{Eff}} C_{\text{UnitParallel}}[T_{\text{Sen}} \rightarrow \text{Layer}][T_{\text{Noisy}} \rightarrow \text{Layer}]}{d + d_0[T_{\text{Sen}} \rightarrow \text{Layer}][T_{\text{Noisy}} \rightarrow \text{Layer}]} \quad (6.22)$$

其中， $C_{\text{UnitParallel}}[i][j]$ 是第 i 层连线和第 j 层连线间单位长度的平行电容， d 和 L_{Eff} 分别是布线单元 T_{Sen} 和 T_{Noisy} 的距离和有效长度(参见图 6.25)， $d_0[i][j]$ 是第 i 层连线和第 j 层连线间的最短距离，主要由工艺参数决定。很显然， $d_0[i][i]=0$ 。

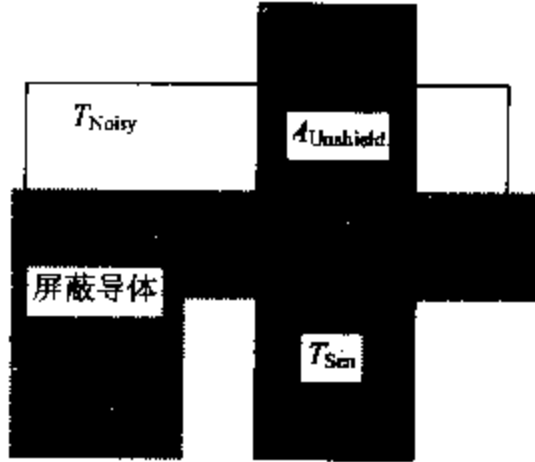


图 6.24 平行电容的简化模型

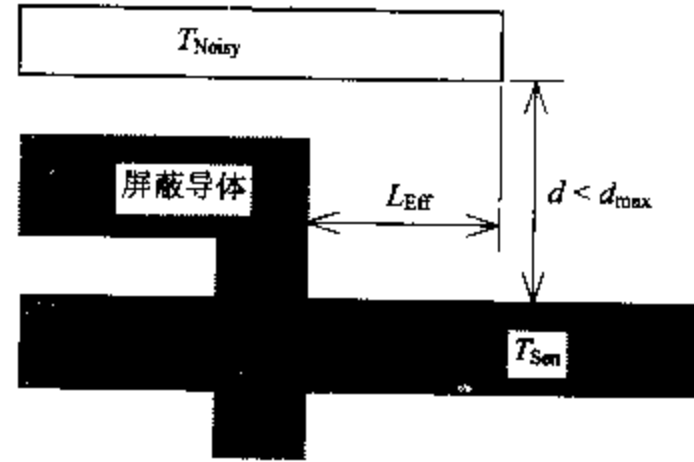


图 6.25 平行电容的简化模型

2. 考虑串扰效应的布线成本

当考虑串扰效应后，布线单元之间的耦合电容越大，相应的布线成本也就越高，我们只要在计算泛路径的布线成本时考虑因串扰效应所引入的布线成本，则不需对前面介绍过的泛路径扩展策略作任何改动，就可以有效地消除连线串扰，从而使布线器仍然具有无网格、支持对称布线和具有重布线的功能。

假设将布线单元 $p \rightarrow \text{head}$ 加入到泛路径 $p \rightarrow \text{last}$ 后形成新的泛路径 p ，则考虑串扰效应后，泛路径 p 的布线成本为：

$$\begin{aligned} C_{\text{Path}}(p) &= \sum_{r \in p} [C_{\text{Wire}}(r) + C_{\text{Crosstalk}}(r)] \\ &= C_{\text{Wire}}(p \rightarrow \text{head}) + C_{\text{Crosstalk}}(p \rightarrow \text{head}) + C_{\text{Path}}(p \rightarrow \text{head}) \end{aligned} \quad (6.23)$$

其中， r 是构成泛路径 p 的布线单元， $C_{\text{Crosstalk}}(t)$ 表示因布线单元 t 存在串扰效应所增加的布线成本，它定义为：

$$C_{\text{Crosstalk}}(t) = \sum_{r \in \Omega(t)} \alpha_{\text{Crosstalk}}(n_r, n_t) [C_{\text{Ovlp}}(r, t) + C_{\text{Parallel}}(r, t)] \quad (6.24)$$

其中， $\Omega(t)$ 表示与布线单元 t 的距离小于 d_{max} 的布线单元的集合， n_r 和 n_t 分别表示布线单元 r 和 t 所对应的电路节点，而 $\alpha_{\text{Crosstalk}}(n_r, n_t)$ 是为消除节点 n_r 和 n_t 之间的串扰所施加的权系数。

由于布线系统中耦合电容采用的是简化模型，因此， $\alpha_{\text{Crosstalk}}(n_r, n_t)$ 是影响布线质量的重要因素^[70]，需要根据单位面积的布线成本 $C_{\text{Layer}}(l)$ 来确定串扰效应的权系数。图 6.26 是一个典型的例子，其中噪声线网对应的电路节点为 n_1 ， A 和 B 是待嵌入线网 n_2 的两端点，如果 $\alpha_{\text{Crosstalk}}(n_1, n_2)$ 过小，则布线器将选择路径 1；如果 $\alpha_{\text{Crosstalk}}(n_1, n_2)$ 过大，

布线器将选择路径 3；如果 $\alpha_{\text{Crosstalk}}(n_1, n_2)$ 适中、且平行电容小于交叠电容，则布线器将选择路径 2。

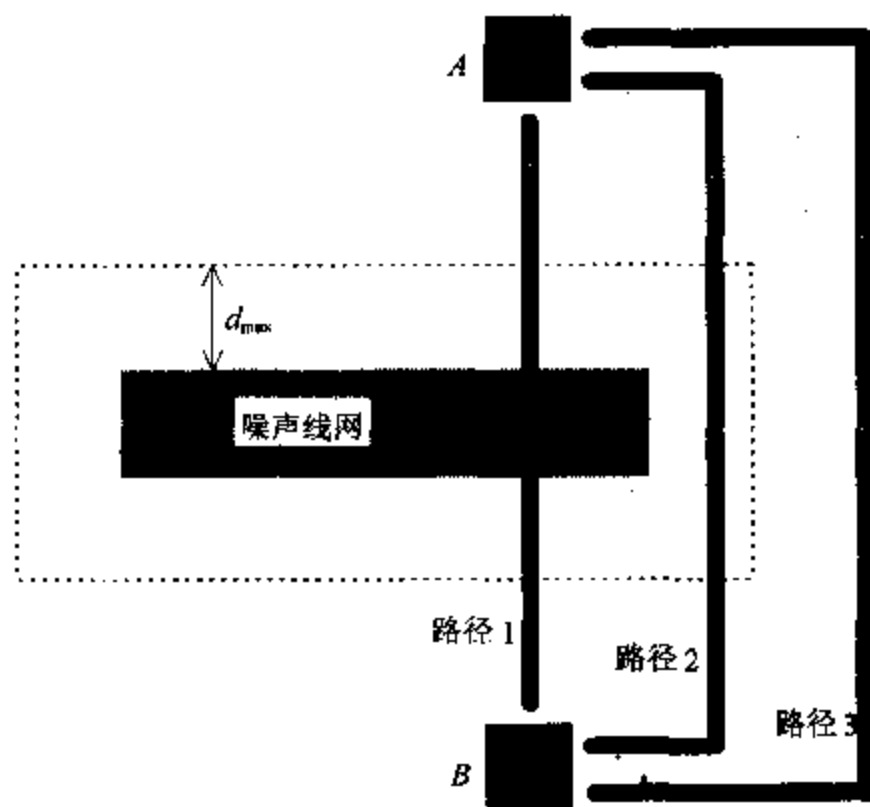


图 6.26 串扰效应的权系数对布线结果的影响

6.5 小 结

本章首先介绍了模拟集成电路版图设计时所关心的主要问题，然后针对这些问题详细介绍了器件级的布局、布线算法。

在处理器件级布局问题时，采用全局收敛性很好的模拟退火法来获得最佳的布局结果，其中，关键的问题是如何保证布局结果符合设计规则，如何综合考虑芯片面积、连线长度、对称性和器件共享等模拟集成电路版图布局阶段的主要问题，本章所介绍的目标函数、新布局的产生方法和权系数的选择策略较好地处理了这些关键问题，只要对它们稍加改动就可以处理别的布局问题。

在处理器件级布线问题时，本章所介绍的线扩展法是一种无网格布线算法，具有迷宫算法和线探法的优点，能支持对称布线、重布线和消除串扰效应，由于采用了角勾链数据结构来表示布线空间，而且，器件上可布线的区域都是 E 类布线单元，因此，该方法还具有在器件上布线的能力。线扩展法的关键是定义泛路径的布线成本和给出扩展泛路径的具体算法，前者“告诉”布线器什么样的布线结构更好一些，后者“告诉”布线器如何获得最佳布线。

第七章 模拟集成电路的硬件描述语言

在传统的模拟集成电路设计过程中,设计者通常都采用电路级的元器件及其联接关系来描述电路结构,SPICE 格式的输入网表文件^[96]是人们长期依赖的法宝。例如:对于图 7.1 所示的 Gilbert 乘法器^[196],其 SPICE 格式的输入网表文件如下:

```
.SUBCKT gilbert.mul.ckt in1p in1n in2p in2n outp outn
```

```
Vcc      vcc      0  DC   5.0
```

```
Iee      3        0  DC   1.0m
```

```
R1       vcc      outp   500
```

```
R2       vcc      outn   500
```

```
Q1       outp     in1p    1    mod.npn
```

```
Q2       outn     in1n    1    mod.npn
```

```
Q3       outp     in1n    2    mod.npn
```

```
Q4       outn     in1p    2    mod.npn
```

```
Q5       1        in2p    3    mod.npn
```

```
Q6       2        in2n    3    mod.npn
```

```
.ENDS
```

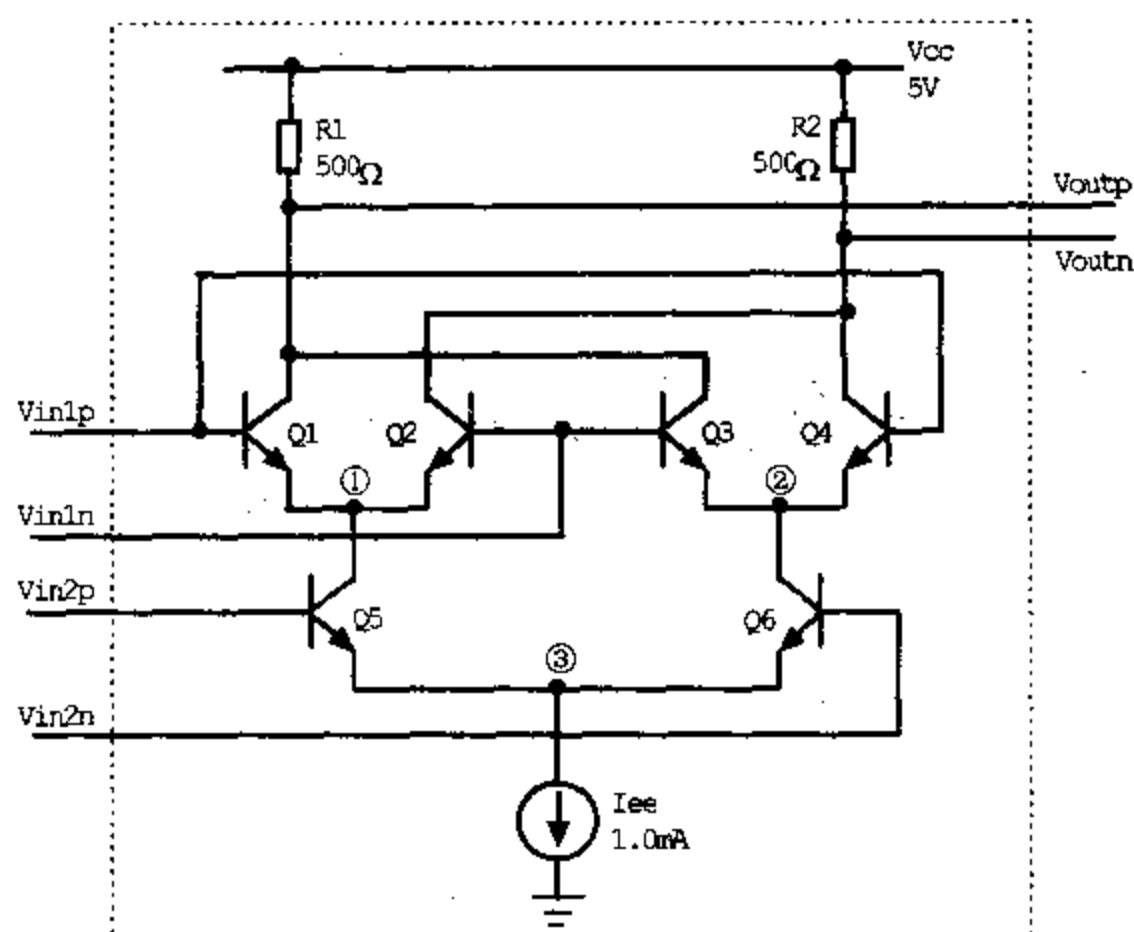


图 7.1 Gilbert 乘法器的原理图

上述描述方式有两个致命的缺陷:①如果设计者不知道(或尚没有完成)模拟系统

(或模块)的具体电路结构就无法描述其行为,因而不适合于系统级设计与仿真;②所描述的模拟系统依赖于生产工艺线(在前面的网表文件中,晶体管模型 mod.npn 来自实际的生产工艺线),不利于在不同的生产厂家之间重复利用已有的模块进行系统设计。很显然,如果我们能够将图 7.1 所示的 Gilbert 乘法器描述为:

$$V_{outp} - V_{outn} = (V_{in1p} - V_{in1n}) * (V_{in2p} - V_{in2n});$$

其中, * 代表乘法运算,则不仅可以使所描述的系统不依赖于具体的生产工艺线,而且特别容易实现混合信号系统的仿真与验证^[12]。上述描述称为行为描述。模拟集成电路的硬件描述语言(简称 AHDL)^[12, 197]的优越性也就在于解决了传统描述方式的缺陷,而各种模拟集成电路综合工具的任务就是将用 AHDL 所描述(定义)的系统映射成具体的电路结构或物理版图。

目前,工业界不十分推崇模拟集成电路硬件描述语言的 IEEE 1076.1 标准^[197],并出现了 MAST^[133]和 VERILOG-A^[134]等 AHDL 的不同规范,但是,随着时间的推移,必然会出现一些被用户广泛接受的标准。本章将介绍目前这些规范所共有的本质特征(主要来自 VHDL-AMS 标准草案),它们将是未来 AHDL 标准的重要组成部分,不论是模拟系统设计者还是模拟集成电路设计自动化系统的研究与开发人员,都需要了解和掌握 AHDL 标准的本质特征。

7.1 描述域的抽象

目前,数字集成电路的硬件描述语言 VHDL^[198]支持电路级、开关级、门级、寄存器传输级(RTL)和行为级等多个层次的描述模式,但由于目前关于模拟集成电路设计层次的抽象尚不够完善,因此,AHDL 主要是对 SPICE 格式的输入网表描述作了扩展,增加了行为级描述,并将电信号的行为描述推广到光、热等其他领域。图 7.2 示意性地对比了 AHDL 和 VHDL 对描述域的抽象,而目前的 MAST^[133]和未来的 VERILOG-AMS^[134]都将同时支持 AHDL 和 VHDL 对描述域的抽象。

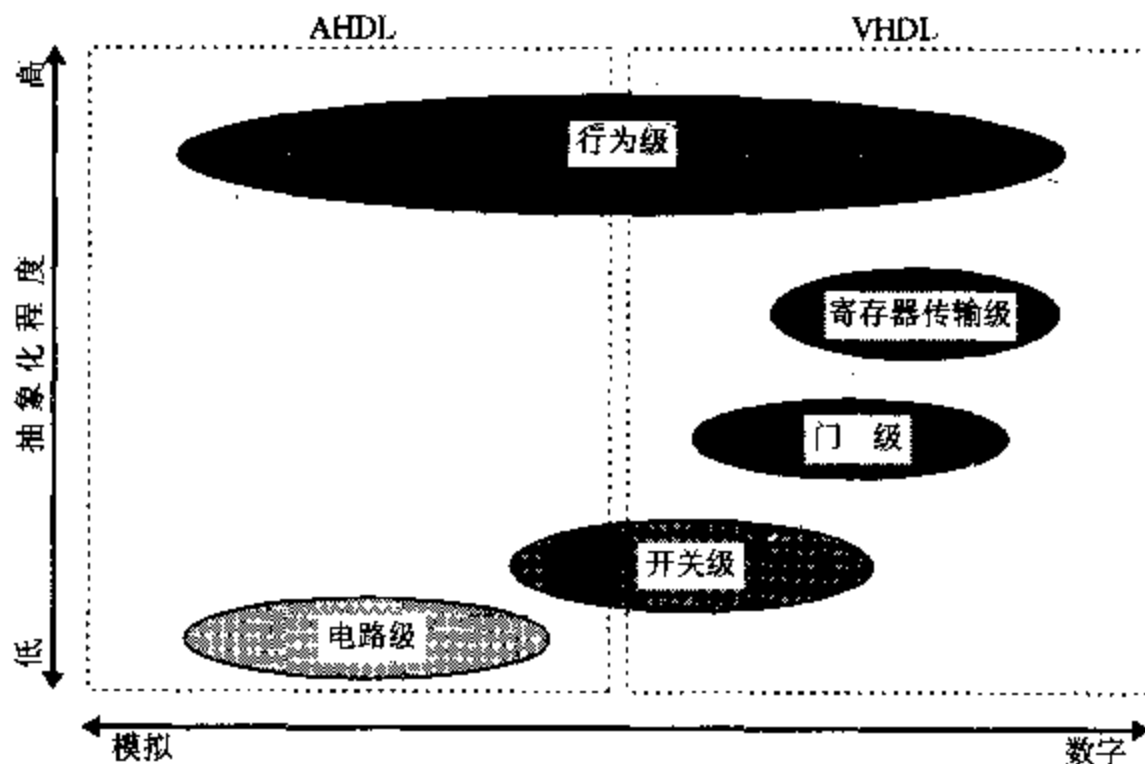


图 7.2 AHDL 和 VHDL 对描述域的抽象

从图 7.2 不难看出, AHDL 主要是对 SPICE 的描述模式作了扩充, 能够支持行为描述。例如: 对于图 7.1 所示的 Gilbert 乘法器, 采用 MAST 语言可以描述为:

```
template gilbert.mul.mast in1p in1n in2p in2n outp outn=gain
  electrical in1p, in1n, in2p, in2n, outp, outn    #定义端口
  number gain=1.0                                #为输入参数赋初值
{
  var i iout                                       #表示 iout 以电流为单位
  val v v1, v2, vo                                #表示 v1, v2 和 vo 以电压为单位
  values {
    v1=v(in1p) - v(in1n)                          #求输入信号的值
    v2=v(in2p) - v(in2n)
    vo=gain * v1 * v2                              #实现乘法运算
  }
  equations {
    i(outp) += iout                                #定义输出端的电流方向
    i(outn) -= iout
    iout; v(outp) - v(outn) = vo                  #将运算结果赋给电路的输出端
  }
}
```

其中, 粗体部分是 MAST 的关键字, # 后面的内容是注释语句。从上述描述可以看出, 设计者已经不需要知道具体的电路结构, 只需要了解 Gilbert 乘法器的行为和端口特性即可。当然, MAST 语言还留有较多的 SPICE 痕迹, 而下述 VERILOG-A 的描述则显得更加行为化(其中, // 后面的内容是 VERILOG-A 的注释语句)。

```
module gilbert.mul.vrlg(in1p, in1n, in2p, in2n, outp, outn);
  inout in1p, in1n, in2p, in2n, outp, outn; //定义输入输出端口
  electrical in1p, in1n, in2p, in2n, outp, outn; //电特性端口
  parameter real gain = 1.0; //定义常数
  analog begin
    //定义 Gilbert 乘法器的行为: 输出电压为两个输入电压之积
    V(outp, outn) <+ gain * V(in1p, in1n) * V(in2p, in2n);
  end
endmodule
```

7.2 混合信号系统及其行为抽象

众所周知, 基尔霍夫(Kirchhoff)电压定律(KVL)、电流定律(KCL)和支路特性(BR)是处理电路问题的基石, 但在混合信号系统中, 我们所面对的已不仅仅是电压、电流, 还有温度、浓度、重量、光强等物理量, 如果采用场论的方法来处理这类问题虽然可以到达很高的精度, 但计算量大, 而且不容易同现有的电路分析、综合软件相结合, 采用集总参数模型、按照与集总电路一致的模式来处理混合信号系统才是比较行之有效的。因此, AHDL 引入了位势(potential)和流量(flow)的概念。

对于保守系统而言,用基尔霍夫位势定律(KPL)和流量定律(KFL)分别替换 KVL 和 KCL。KPL 是指环路中各支路位势差之和为零(参见图 7.3);而 KFL 是指流出任何节点的流量和为零(参见图 7.4)。至于支路特性的描述则需要结合物理背景,利用 AHDL 提供的算符和函数给出相应的行为描述。KPL 和 KFL 使混合信号系统的抽象与描述变得比较容易。

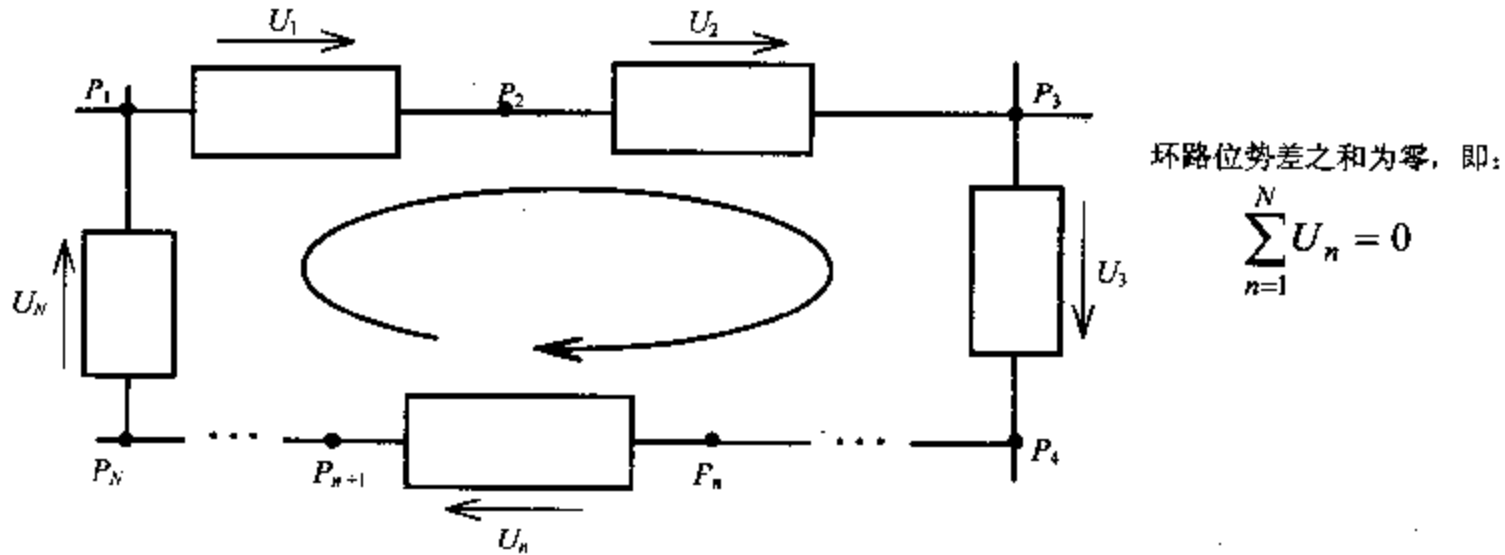


图 7.3 KPL 示意图

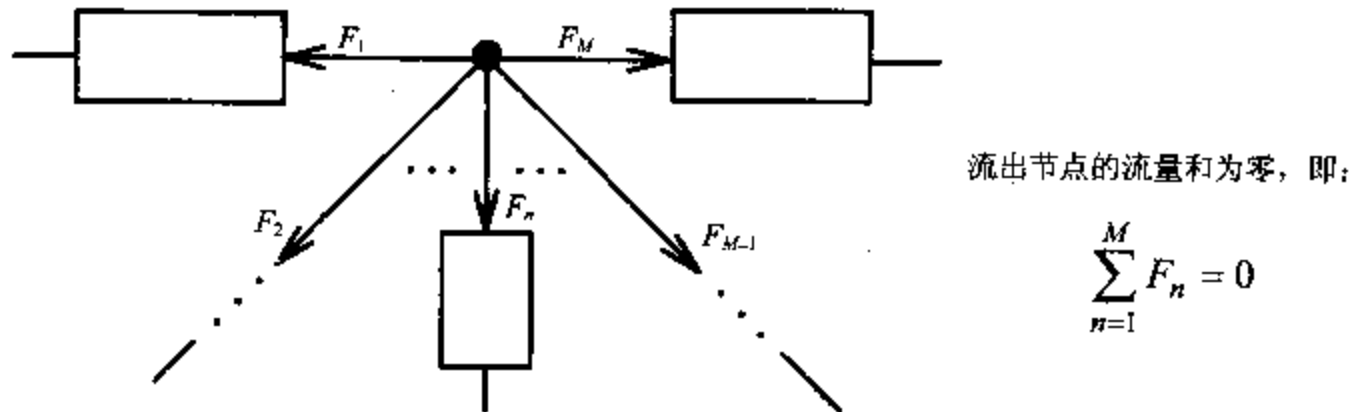


图 7.4 KFL 示意图

信号流系统是 AHDL 对实际系统的一种抽象。信号流系统与保守系统不同,它只有(或设计者只关心)位势,而没有流量(或设计过程不必考虑流量)。理想放大器就是最典型的信号流系统。图 7.5 是将放大器作为信号流系统来处理时的符号和相应的行为描述。从图 7.5 可以看出,将放大器作为信号流系统时,不需要考虑其输入端口和输出端口的电压电流关系,而仅仅关心其放大信号的行为。

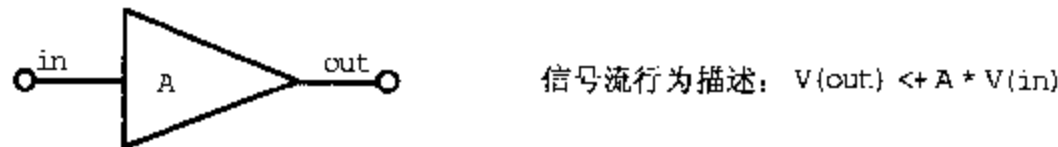


图 7.5 放大器及其信号流行为描述

7.3 模拟算符与模拟事件

为方便设计者描述系统的行为, AHDL 定义了一系列模拟算符,包括微分算符 $\text{ddt}(\cdot)$, 积分算符 $\text{idt}(\cdot)$, 延时算符 $\text{delay}(\cdot)$, 过渡算符 $\text{transition}(\cdot)$, 摆率算符 $\text{slew}(\cdot)$, 拉氏变换算符 $\text{laplace.zp}(\cdot)$, $\text{laplace.zd}(\cdot)$, $\text{laplace.np}(\cdot)$, $\text{laplace.nd}(\cdot)$,

Z 变换算符 $zi_zp(\cdot)$, $zi_zd(\cdot)$, $zi_np(\cdot)$, $zi_nd(\cdot)$ 。过渡算符主要为上升沿或下降沿为零的理想信号加入适当的过渡过程；摆率算符主要是用来限制信号的最大变化速度，使分段线性信号变平滑；拉氏变换算符和 Z 变换算符的作用分别是描述形如式(3.1)的 s 域和 z 域的传输函数。关于这些算符的语法格式和含义请参见文献[12, 133, 134, 197]，为节省篇幅，表 7.1 只给出了微分、积分和延时等 3 个简单算符的含义。

表 7.1 微分算符、积分算符和延时算符的含义

微分算符:	$g = ddt(f)$	$g(t) = \frac{df(t)}{dt}$
积分算符:	$g = idt(f, C)$	$g(t) = \int_0^t f(\tau) d\tau + C$
延时算符:	$g = delay(f, \text{delta } t)$	$g(t) = f(t - \text{delta } t)$

为了在混合信号系统中进行条件控制，AHDL 也引入了 if-else-then, case, while 和 when 等语句^[197]，而支持这些语句的关键是需要判断是否有相应的事件发生。在模拟系统中，判断事件是否发生的依据是检测指定信号是否到达某阈值。因此，AHDL 引入了模拟事件算符 cross($\cdot$)，它用来检测指定的信号在何时过零点，并在该时刻产生一个事件，以激励与之相关的子系统(模块)。另外，AHDL 还定义了另一个模拟事件算符 timer($\cdot$)，其语法格式为 timer(start, period)，它将使混合信号仿真系统从 start 时刻开始、每隔 period 段时间就产生一个事件。

7.4 混合描述域及混合信号系统的描述

AHDL 对 SPICE 的电路网表描述模式进行了拓展，引入了行为描述模式。在描述混合信号系统时，AHDL 允许纯电路网表的描述模式、纯行为描述模式和电路网表与行为模式的混合模式，从而使系统设计和电路级设计都具有很大的灵活性。这里，我们以 VERILOG-A 为例来说明该问题。VERILOG-A 描述一个模块(系统)时不仅需要描述其行为，还需要声明构造和连接该模块所必须的接口。这里以调制解调器(参见第 1.1 节)为例，描述该模块的语法格式为：

```

module modem(din, dout);
    inout din, dout;
    electrical din, dout;
    parameter real carry_freq = 28.8E+3;    //参数定义
    结构描述部分
    analog begin
        行为描述部分
    end
endmodule

```

其中，行为描述部分又称为行为描述域，前面已经介绍过这方面的内容，而结构描述部分与 SPICE 的输入网表格式非常相似(当然也可以采用一致的格式)，每一个元件是一个语句，描述其性能以及和别的元件的联接关系，又称为结构描述域。这类语句的

格式为：

```
module.type #(.param.name(param.value)) instance.name(port.list);
```

其中, module.type 是元件的类型; #(.param.name(param.value)) 是可选项, 如果有该可选项, 则表示将模块 module.type 中的参数 param.name 设置为 param.value; instance.name 是元件名, 与 SPICE 的输入网表文件一样, 同一电路中不同元件必须采用不同元件名; port.list 描述元件 instance.name 的所有端口, 它必须和模块 module.type 定义的端口一一对应。例如: 利用前面定义的 modem 模块, 可以将载频为 56.6K、输入输出端口分别为 input 和 output 的调制解调器 hyes.modem 描述为:

```
modem #(.carry_freq(56.6E+3)) hyes.modem(input, output);
```

为便于读者了解 AHDL 所支持的各种描述域, 这里仍然以调制解调器中的发送器为例。我们在第 1.1 节已经介绍了发送器的基本功能, 图 7.6 是其系统结构。

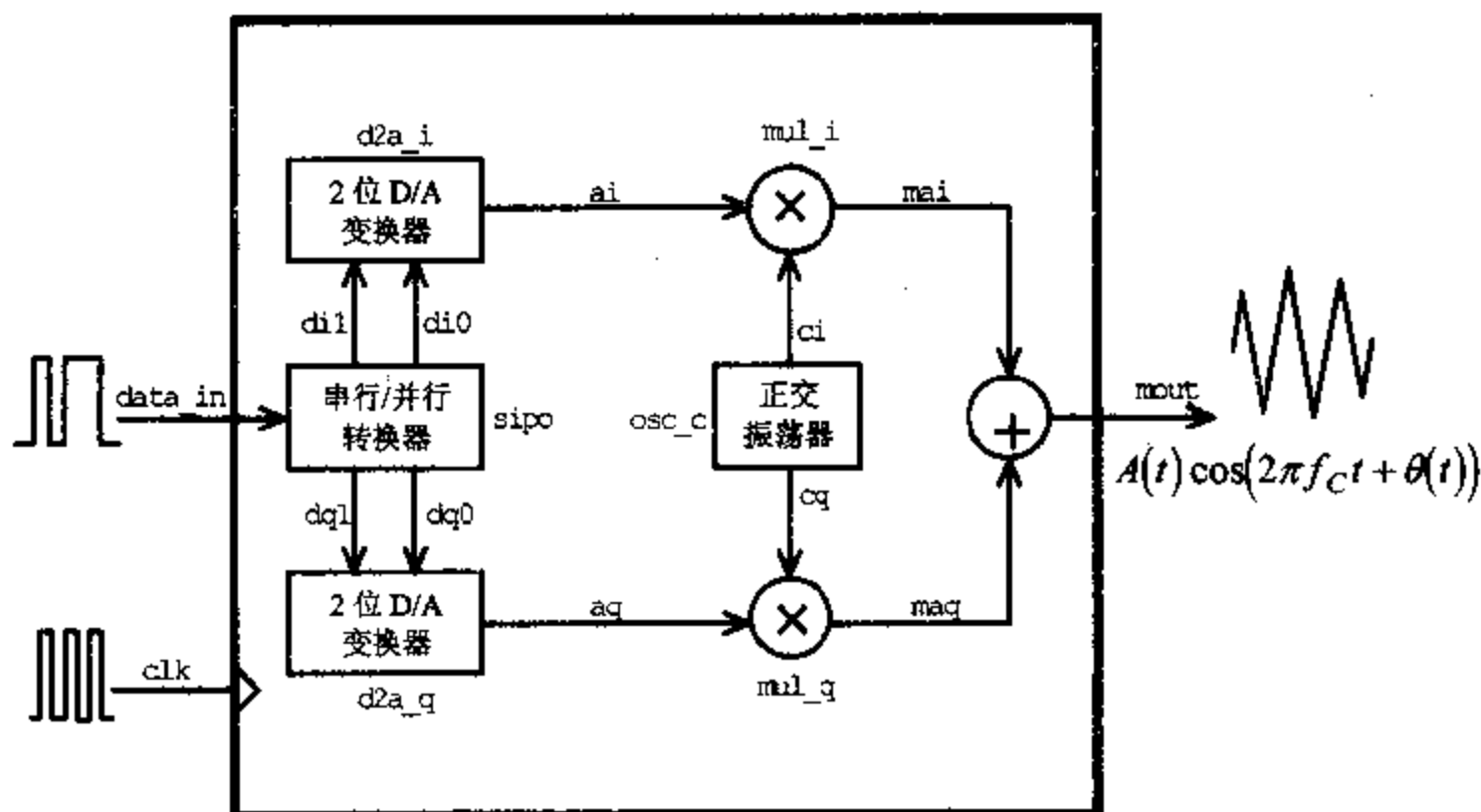


图 7.6 调制解调器中发送器的结构

如果 2 位 D/A 变换器定义为 d2a(d1, d0, aout, clk), 串行/并行转换器定义为 s2p(din, di1, di0, dq1, dq0, clk), 正交振荡器定义为 quadrature.osc(ci, cq), 其振荡频率和幅度分别由参数 freq 和 mag 表示, 乘法器采用图 7.1 所示的 Gilbert 乘法器, 并利用 SPICE 格式的子电路 gilbert.mul.ckt 来描述, 则图 7.6 所示的发送系统可以描述为:

```
module modem.mixed.level(data.in, clk, mout);
    inout data.in, clk, mout;
    electrical data.in, clk, mout;

    parameter real fc=28.8E3, a=1.0;

    electrical di1, di0, dq1, dq0, ai, aq, ci, cq, mai, maq;
    s2p sipo(data.in, di1, di0, dq1, dq0, clk);
    d2a d2a.i(di1, di0, ai, clk);
    d2a d2a.q(dq1, dq0, aq, clk);
```

```

quadrature.osc # (.freq(fc), .mag(a)) osc.c(ci, cq);

xmul.i ci 0 ai 0 mai gilbert.mul.ckt
xmul.q cq 0 aq 0 maq gilbert.mul.ckt

analog begin
    V(mout) <+ V(mai) + V(maq);
end
endmodule

```

其中，粗体部分就是 SPICE 的子电路调用形式^[96]。我们也可以用第 7.1 节中所定义的行为模块 `gilbert.mul.vrlg()` 来描述，只要将上面两行粗体部分替换为下述形式即可：

```

gilbert.mul.vrlg muli(ci, 0, ai, 0, mai, 0);
gilbert.mul.vrlg mulq(cq, 0, aq, 0, maq, 0);

```

从上面的实例不难看出，由于 AHDL 对 SPICE 进行了扩展，在引入行为描述的同时，还支持混合描述模式，使混合信号系统的描述变得非常简洁。

7.5 小 结

在未来的 ASIC 芯片中，数模混合系统、混合信号系统将越来越普遍，而 AHDL 将是描述这类系统的有力工具。本章以目前工业界最具影响力的 MAST 和 VERILOG-A 为背景，简要介绍了 AHDL 的主要特征。AHDL 最核心的部分是对 SPICE 的网表输入格式进行了扩展，对模拟系统的描述域进行了抽象，并引入了行为描述和实现行为描述的模拟算符和模拟事件；而 KPL 和 KFL 则将集总参数的电路理论推广到光、热等领域，从而简化了混合信号系统的描述与仿真。可以肯定，模拟电路及混合信号系统的硬件描述语言 (VHDL-AMS) 将在近期内形成为工业界广泛接受的标准，并在模拟系统、混合信号系统的仿真与综合方面得到广泛应用。

参 考 文 献

- [1] Gajski D. (editor), *Silicon Compilation*, Addison Wesley, 1988.
- [2] 杨华中, 刘润生, 汪慧, 范崇治, 模拟集成电路设计自动化技术, 电子科学学刊, Vol. 18 (2), 1996年3月, pp. 202—208.
- [3] 杨华中, 模拟集成电路综合方法的研究, 清华大学博士学位论文, 1998. 6.
- [4] Gielen G., "Design Automation for Analog Integrated Circuits", Ph. D. Dissertation, Katholieke Universiteit Leuven, 1990.
- [5] Carley L. R., et al, "ACACIA: The CMU Analog Design System", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1989, pp. 4. 3. 1—5.
- [6] Cohn J. M., Garrod D. J., Rutenbar R. A., Carley L. R., "KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing", IEEE J. Solid-State Circuits, Vol. SC-26 (3), Mar. 1991, pp. 330—342.
- [7] Camposano R., "High-Level Synthesis — A Tutorial", presymposium European Design Automation Conf. (EDAC), 1990.
- [8] Harjani R., Rutenbar A., Carley L. R., "A Prototype Framework for Knowledge-Based Analog Circuit Synthesis", in Proc. ACM/IEEE Design Automation Conf. (DAC), 1987, pp. 42—49.
- [9] Sansen W., Gielen G., Op't Eynde F., "Operational Amplifier Design", COMETT Report 643D, Nov. 1988.
- [10] Van Petegem W., De Wachter D., Sansen W., "Electrothermal Simulation of Integrated Circuits", in Proc. IEEE Semi. Thermal and Temp. Meas. Sym. (STTMS), 1990, pp. 70—73.
- [11] Verghese N. K., Schmerbeck T. J., Allstot D. J., "Simulation Techniques and Solutions for Mixed-Signal Coupling in Integrated Circuits", Kluwer Academic Publishers, 1995.
- [12] Vachoux A., Berg J. M., Levia O., Rouillard J., "Analog and Mixed-Signal Hardware Description Languages", Kluwer Academic Publishers, 1997.
- [13] Carley L. R., Rutenbar R. A., "Automatic Synthesis of Analog Integrated Circuits—A Tutorial", presymposium IEEE Intern. Conf. Computer Aided Design (ICCAD), 1988.
- [14] Vittoz E. A., "Future of Analog in The VLSI Environment", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1990, pp. 1584—1587.
- [15] Tsividis Y. P., "Analog MOS Integrated Circuits—Certain New Ideas, Trends, and Obstacles", IEEE J. Solid State Circuits, Vol. SC-22 (3), June 1987, pp. 317—321.
- [16] Cole B. C., "The Changing Analog World", Electronics, Sept. 1989, pp. 73—80.
- [17] Santo B., Chen K. T., "Technology'90: Solid State", IEEE Spectrum, Jan. 1990, pp. 41—43.
- [18] Op't Eynde F., "High-Performance Analog Interfaces for Digital Signal Processors", Ph. D. Dissertation, Katholieke Universiteit Leuven, 1990.
- [19] Allen P. E., "A Tutorial—Computer-Aided Design of Analog Integrated Circuits", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1986, pp. 608—616.
- [20] Steyaert M., "Monolithic Low-Power Data-Aquisition System for Biomedical Purpose", Ph. D. Dissertation, Katholieke Universiteit Leuven, 1987.
- [21] Callewaert L., et al, "Front End and Signal Processing Electronics for Detectors at High Luminosity Colliders", IEEE Trans. Nuclear Science, Vol. NS-36 (1), Jan. 1989, pp. 446—450.
- [22] Nilson C. D., Darling R. B., Pinter R. B., "Shunting Neural Network Photodetector Arrays in Analog CMOS", IEEE J. Solid-State Circuits, Vol. SC-29 (10), Oct. 1994, pp. 1291—1296.
- [23] Mead C., "Analog VLSI and Neural Systems", Addison-Wesley, 1989.

- [24] Pangrle B. M. , Gajski D. D. , "Design Tools for Intelligent Silicon Compilation", *IEEE Trans. Computer-Aided Design*, Vol. CAD-6 (6), Nov. 1987, pp. 1098-1112.
- [25] Sechen C. , Sangiovanni-Vincentelli A. , "The Timber Wolf Placement and Routing Package", *IEEE J. Solid-State Circuits*, Vol. SC-20 (2), Apr. 1985, pp. 510-522.
- [26] Rohrer R. A. , "Fully Automated Network Design by Digital Computer", *Proc. IEEE*, Vol. 55 (11), Nov. 1967, pp. 1929-1939.
- [27] Director S. W. , Rohrer R. A. , "Automated Network Design: The Frequency Domain Case", *IEEE Trans. Circuit Theory*, Vol. CT-15 (3), June 1968, pp. 330-337.
- [28] Brayton R. K. , Hachtel G. D. , Sangiovanni-Vincentelli A. L. , "A Survey of Optimization Techniques for Integrated-Circuit Design", *Proc. IEEE*, Vol. 69 (10), Oct. 1981, pp. 1334-1362.
- [29] Nye W. , Riley D. C. , Sangiovanni-Vincentelli A. , Tits A. L. , "DELIGHT. SPICE, An Optimization-Based System for The Design of Integrated Circuits", *IEEE Trans. Computer-Aided Design*, Vol. CAD-7(2), Apr. 1988, pp. 501-519.
- [30] Shyu J. M. , Sangiovanni-Vincentelli A. , "ECSTASY, A New Environment for IC Design Optimization", in *Proc. IEEE Int. Conf. Computer Aided Design (ICCAD)*, Nov. 1988, pp. 484-487.
- [31] Arora R. , Dasgupta U. , Hoocevar D. , Goff L. , "OASYS: A Tool for Aiding in Design of High Performance Linear Circuits", in *Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS)*, May 1990, pp. 1911-1914.
- [32] 杨华中, 范崇治, 刘润生, 集成电路合格率优化方法的研究, *电子学报*, Vol. 22 (11), 1994 年 11 月, pp. 106-109.
- [33] Carley L. R. , Rutenbar R. A. , "How to Automate Analog IC Design", *IEEE Spectrum*, Vol. 25 (8), Aug. 1988, p. 26-30.
- [34] Maulik P. C. , Carley L. R. , Allstot D. J. , "Sizing of Cell-Level Analog Circuits Using constrained Optimization Techniques", *IEEE J. Solid-State Circuits*, Vol. SC-28 (3), Mar. 1993, pp. 233-241.
- [35] Nagaraj N. S. , "A New Optimizer for Performance Optimization of Integrated Circuits By Device Sizing", in *Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS)*, May 1992, pp. 2102-2105.
- [36] Henderson P. K. , "Sizing of Analog Circuits for Small-signal Gains", in *Proc. European Design Automation Conf. (EDAC)*, Sept. 1992, pp. 469-473.
- [37] Degrauwe M. G. R. , et al, "IDAC, An Interactive Design Tool for Analog CMOS Circuits", *IEEE J. Solid-State Circuits*, Vol. SC-22 (6), Dec. 1987, pp. 1106-1116.
- [38] Fernandez F. V. , et al, "Symbolic Analysis of Large Analog Integrated Circuits by Approximation During Expression Generation", in *Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS)*, Vol. 1 May 1994, pp. 25-28.
- [39] Ochotta E. S. , Rutenbar R. A. , L. R. Carley, "Equation-Free Synthesis of High-Performance Linear Analog Circuits", in *Proc. of The 1992 Brown/MIT Conf. on Advanced Research in VLSI and Parallel Systems*, The MIT Press, pp. 129-143.
- [40] Gielen G. G. E. , Walscharts H. C. C. , Sansen W. M. C. , "ISAAC: A Symbolic Simulator for Analog Integrated Circuits", *IEEE J. Solid-State Circuits*, Vol. SC-24(6), Dec. 1989, pp. 1587-1597.
- [41] Gielen G. G. E. , Walscharts H. C. C. , Sansen W. M. C. , "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing", *IEEE J. Solid-State Circuits*, Vol. SC-25 (3), June 1990, pp. 707-713.
- [42] Sharif-Bakhtiar M. , Ahamd M. A. , "Symbolic Analysis of Electronic Circuits Based on A Tree Enumeration Technique", *IEE Proc. Part G*, Vol. 140 (1), Feb. 1993, pp. 68-74.
- [43] Degrauwe M. G. R. , et al, "Towards an Analog System Design Environment", *IEEE J. Solid-State Circuits*, Vol. SC-24 (3), June 1989, pp. 659-671.
- [44] Press W. H. , Flannery B. P. , Teukolsky S. A. , Vetterling W. T. , *Numerical Recipes in C*, (2nd ed.), New York, Cambridge Univ. Press, 1992.
- [45] Koh H. Y. , Sequin C. H. , Gray P. R. , "OPASYN: A Compiler for CMOS Operational Amplifiers", *IEEE Trans.*

Computer-Aided Design, Vol. CAD-9(1), Feb. 1990, pp. 113-125.

- [46] Berkcan E., d'Abreu M., Laughton W., "Analog Compilation Based on Successive Decomposition", in Proc. ACM/IEEE Design Automation Conf. (DAC), June 1988, pp. 369-375.
- [47] Swings K., Donnay S., Sansen W., "HECTOR: A Hierarchical Topology-Construction Program for Analog Circuits Based on a Declarative Approach to Circuit Modeling", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1991, pp. 5.3/1-4.
- [48] Stoffels J., van Reeuwijk C., "Ampdes: A Program for The Synthesis of High-Performance Amplifiers", in Proc. European Design Automation Conf. (EDAC), Sept. 1992, pp. 474-479.
- [49] Harjani R., Rutenbar R. A., Carley L. R., "OASYS: A Framework for Analog Circuit Synthesis", IEEE Trans. Computer-Aided Design, Vol. CAD-8 (6), Dec. 1989, pp. 1247-1266.
- [50] Ning Z.-Q., Mouthaan T., Wallinga H., "SEAS: A Simulated Evolution Approach for Analog Circuit Synthesis", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1991, pp. 5.2/1-4.
- [51] El-Turky F. M., Perry E. E., "BLADES: An Artificial Intelligence Approach to Analog Circuit Design", IEEE Trans. Computer-Aided Design, Vol. CAD-8 (3), June 1989, pp. 680-692.
- [52] Fung A. H., Chen D. J., Lai Y. N., Sheu B. J., "Knowledge-Based Analog Circuit Synthesis with Flexible Architecture", in Proc. IEEE Int. Conf. Computer Design (ICCD), Oct. 1988, pp. 48-51.
- [53] Sheu B. J., Lee J. C., Fung A. H., "Flexible Architecture Approach to Knowledge-Based Analog IC design", IEE Proc. Part G, Vol. 137 (4), Aug. 1990, pp. 266-274.
- [54] Toumazou C., Makris C. A., Berrah C. M., "ISAID—An automated Analog Design System", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1990, pp. 531-535.
- [55] Toumazou C., Makris C. A., "Analog IC Design Automation; Part I—Automated Circuit Generation; New Concepts and Methods", IEEE Trans. Computer-Aided Design, Vol. CAD-14 (2), Feb. 1995, pp. 218-238.
- [56] Makris C. A., Toumazou C., "Analog IC Design Automation Part I—Automated Circuit Correction by Qualitative Reasoning", IEEE Trans. Computer-Aided Design, Vol. CAD-14 (2), Feb. 1995, pp. 239-254.
- [57] Proesmans F., "Topology Selection", ESAT-MICAS Seminal, Dec. 16, 1993.
- [58] Maulik P. C., Carley L. R., Rutenbar R. A., "Integer Programming Based Topology Selection of Cell-Level Analog Circuits", IEEE Trans. Computer-Aided Design, Vol. CAD-14 (4), Apr. 1995, pp. 401-412.
- [59] Maulik P. C., "Formulations for Optimization-Based Synthesis of Analog Cells", Research Report No. CMU-CAD-92-50, Carnegie Mellon Univ., Oct. 1992.
- [60] Antao B. A. A., "Automatic Analog Model Generation for Behavioral Simulation", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1992, pp. 12.2/1-4.
- [61] Antao B. A. A., Saleh R., "Simulation Model Transformations for Analog Hardware Description Languages", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1995, pp. 25.6/1-4.
- [62] Antao B. A. A., "Architectural Exploration for Analog System Synthesis", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1995, pp. 25.4/1-4.
- [63] Somanchi S., Manwaring M. L., "Analog Synthesis from Behavioral Descriptions", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1993, pp. 2079-2082.
- [64] Abel C. J., Michael C., Ismail M., Teng C. S., "Characterization of Transistor Mismatch for Statistical CAM of Submicron CMOS Analog Circuits", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1993, pp. 1401-1404.
- [65] Pillan M., Sciuto D., "Constraint Generation and Placement for Automatic Layout Design of Analog Integrated Circuits", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), Vol. 1, May 1994, pp. 355-358.
- [66] Onodera H., Kanbara H., Tamaru K., "Operational-Amplifier Compilation with Performance Optimization", IEEE J. Solid-State Circuits, Vol. SC-25 (2), Apr. 1990, pp. 466-473.
- [67] Rijmenants J., et al., "ILAC: An Automated Layout Tool for Analog CMOS Circuits", IEEE J. Solid-State Circuits, Vol. SC-24 (2), Apr. 1989, pp. 417-425.

- [68] Malavasi E., Sangiovanni-Vincentelli A., "Area Routing for Analog Layout", IEEE Trans. Computer-Aided Design, Vol. CAD-12 (8), Aug. 1993, pp. 1186-1197.
- [69] Cohn J. M., "Automatic Device Placement for Analog Cells in KOAN", Research Report No. CMUCAD-92-07, Carnegie Mellon Univ., Feb., 1992.
- [70] Garrod D. J., "Device-Level Routing of Analog Cells in ANAGRAM II", Research Report No. CMUCAD-91-59, Carnegie Mellon Univ., Aug., 1991.
- [71] Choudhury U., Sangiovanni-Vincentelli A., "Automatic Generation of Parasitic Constraints for Performance-Constrained Physical Design of Analog Circuits", IEEE Trans. Computer-Aided Design, Vol. CAD-12 (2), Feb. 1993, pp. 208-224.
- [72] Yeong-Yil Yang, Chong-Min Kyung, "HALO: An Efficient Global Placement Strategy for Standard Cells", IEEE Trans. Computer-Aided Design, Vol. CAD-11(8), Aug. 1992, pp. 1024-1031.
- [73] Meyer zu Bexten V., Moraga C., Klinke R., Brockherde W., Hess K.-G., "ALSYN: Flexible Rule-Based Layout Synthesis for Analog IC's", IEEE J. Solid-State Circuits, Vol. SC-28 (3), Mar. 1993, pp. 261-268.
- [74] Shiraishi Y., Kirura M., Kobayashi K., Hino T., Seriuchi M., Kusaoke M., "A High-Packing Density Module Generator for Bipolar Analog LSIs", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1990, pp. 194-197.
- [75] Mehranfar S. W., "A Technology-Independent Approach to Custom Analog Cell Generation", IEEE J. Solid-State Circuits, Vol. SC-26(3), Mar. 1991, pp. 386-393.
- [76] Malavasi E., Choudhury U., Sangiovanni-Vincentelli A., "A Routing Methodology for Analog Integrated Circuits", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1990, pp. 202-205.
- [77] Ohtsuka T., Kurosawa N., Kunieda H., "The Improvement in Performance-Driven Analog LSI Layout System LIBRA", IEICE Trans. Fundamentals, Vol. E76-A(10), Oct. 1993, pp. 1626-1635.
- [78] Lee J.-C., Gowda S. M., Sheu B. J., "Fully Automated Layout Generators for High-Performance Analog VLSI Modules", IEEE Region 10 Intern. Conf. (TENCON), Nov. 1989, pp. 893-896.
- [79] Gielen G., Sansen W., Symbolic Analysis for Automated Design of Analog Integrated Circuits, Kluwer Academic Publishers, 1991.
- [80] Vachoux A., Berg J. M., Levia O., Rouillard J., Analog and Mixed-Signal Hardware Description Languages, Kluwer Academic Publishers, 1997.
- [81] Luenberg D. G. 著, 夏尊铨等译, 线性与非线性规划引论, 科学出版社, 1982.
- [82] Nemhauser G. L., Wolsey L. A., Integer and Combinatorial Optimization, New York, Wiley, 1988.
- [83] Geoffrion A. M., "Generalized Benders Decomposition", J. Optimization Theory and Applications, Vol. 10 (10), Oct. 1972, pp. 237-260.
- [84] Duran M. A., Grossmann I. E., "An Outer-Approximation Algorithm for A Class of Mixed-Integer Nonlinear Programs", Math. Programming, Vol. 26, 1986, pp. 307-339.
- [85] Kirkpatrick S., Jr. Galatt C. D., Vecchi M. P., "Optimization by Simulated Annealing", Science, Vol. 220 (4598), May 13, 1983, pp. 671-680.
- [86] Kling R. M., Banerjee P., "ESP: Placement by Simulated Evolution", IEEE Trans. Computer-Aided Design, Vol. CAD-8(5), May 1989, pp. 245-256.
- [87] Goldberg D. E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley Publishing Company, Inc., 1989.
- [88] Huang M. D., Romeo F., Sangiovanni-Vincentelli A., "An Efficient General Cooling Schedule for Simulated Annealing", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1986, pp. 381-384.
- [89] Yang H. Z., Liu R. S., H. Wang, Fan C. Z., "Simultaneous Topology Selection and Sizing for Synthesis of Analog Cells", in Proc. IEEE TENCON'95, Hong Kong, Nov. 1995, pp. 159-162.
- [90] Gill P., Murray W., Saunders M., Wright M., "User's Guide for NPSOL (Version 4.0)", Tech. Report SOL 86-2, Stanford Univ., Jan. 1986.

- [91] Yang H. Z., Fan C. Z., Wang H., Liu R. S., "Simulated Annealing Algorithm with Multi-Molecule: an Approach to Analog Synthesis", IEEE ED&TC96, Paris, Mar. 11th-14th, 1996, pp. 571-575.
- [92] 杨华中, 范崇治, 汪蕙, 刘润生, 多分子模拟退火法及其在模拟集成电路综合中的应用, 电子学报, Vol. 25(5), 1997年5月, pp. 82-85.
- [93] Dixon L. C. W., Szego G. P., (ed.), Towards Global Optimization, North-Holland Pub., 1975.
- [94] Walser G. W., Hansen E. R., Sengupta, "Test Results for A Global Optimization Algorithm", in Proc. SIAM Conf. on Num. Opt., 1984, pp. 272-287.
- [95] Vetterling W. T., Teukolsky S. A., Press W. H., Flannery B. P., Numerical Recipes Example Book [C], (2nd ed.), Cambridge Univ. Press, 1992.
- [96] Nagel L. W., "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Ph. D. Dissertation, U. C. Berkeley, May 1975.
- [97] Adhy P. R., "Applied Circuit Theory: Matrix and Computr Methods", Ellis Horwood, 1980, pp. 263-343.
- [98] Chua L. O., Lin P. M., "Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques", Prentice-Hall, 1975.
- [99] Meta Software Inc. (Now Avanti Corp.), "HSPICE User's Manual", Feb. 1996.
- [100] Konezykowska A., Bon M., "Automated Design Software for Switched-Capacitor IC's with Symbolic Simulator SCYMBAL", in Proc. ACM/IEEE Design Automation Conf. (DAC), June 1988, pp. 363-368.
- [101] Gielen G., Swings K., Sansen W., "An Intelligent Design System for Analog Integrated Circuits", in Proc. European Design Automation Conf. (EDAC), Sept. 1990, pp. 169-173.
- [102] Director S. W., Maly W., Strojwas A., "Statistically Based IC Process Simulation and Applications in VLSI Design and Manufacturing", Kluwer Academic Publishers, 1990.
- [103] Vos F., Trullemans C., "A Switch Level Symbolic Simulator", in Proc. ESSCIRC, 1989, pp. 244-247.
- [104] Bolsens I., "Electrical Correctness Verification of MOS Digital Circuits Using Expert System and Symbolic Analysis Techniques", Ph. D. Dissertation, Katholieke Universiteit Leuven, 1989.
- [105] Bose S., Fisher A. L., "Verifying Pipelined Hardware Using Symbolic Logic Simulation", in Proc. IEEE Int. Conf. Computer Design (ICCD), Oct. 1989, pp. 217-221.
- [106] 杨华中, 汪蕙, 数值计算方法与 C 语言工程函数库, 科学出版社, 1996.
- [107] Vlach J., Singhal K. 著, 汪蕙, 李普成, 刘润生, 范崇治译, 电路分析的计算机方法, 科学出版社, 1992.
- [108] Ruehli A. E. (editor), "Circuit Analysis, Simulation and Design", Advances in CAD for VLSI series, Vol. 3, Part 2, North-Holland, 1987.
- [109] Van Petegem W., De Wachter D., Sansen W., "Electrothermal Simulation of Integrated Circuits", in Proc. IEEE Semiconductor Thermal and Temperature Measurement Symposium (STTMS), 1990, pp. 70-73.
- [110] Seda S. J., Degrauwe M. G. R., Fichtner W., "A Symbolic Analysis Tool for Analog Circuit Design Automation", in Proc. IEEE Intern. Conf. Computer Aided Design (ICCAD), 1989, pp. 488-491.
- [111] Tsai M. K., Shenoit B. A., "Generation of Symbolic Network Functions Using Computer Software Techniques", IEEE Trans. Circuits And Systems, Vol. CAS-24(6), June 1977, pp. 344-346.
- [112] Starzyk J., Sliwa E., "Flowgraph Analysis of Large Electronic Networks", IEEE Trans. Circuits And Systems, Vol. CAS-33 (3), March 1988, pp. 302-315.
- [113] Mielke R. R., "A New Signal Flowgraph Formulation of Symbolic Network Functions", IEEE Trans. Circuits And Systems, Vol. CAS-25(6), June 1978, pp. 334-340.
- [114] Chow J. T., Willson Jr. A. N., "A Microcomputer-Oriented Algorithm for Symbolic Networks Analysis", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1985, pp. 575-577.
- [115] Sannuti P., Puri N. N., "Symbolic Network Analysis - An Algebraic Formulation", IEEE Trans. Circuits And Systems, Vol. CAS-27(8), Aug. 1980, pp. 679-687.
- [116] Singhal K., Vlach J., "Symbolic Analysis of Analog and Digital Circuits", IEEE Trans. Circuits And Systems, Vol. CAS-24 (11), Nov. 1977, pp. 598-609.

- [117] Fidler J. K., Sewell J. I., "Symbolic Analysis for Computer-Aided Circuit Design — The Interpolative Approach", IEEE Trans. Circuits Theory, Vol. CT-20(6), Nov. 1973, pp. 738—741.
- [118] 郑君理, 杨为理, 应启珩, 信号与系统 (下册), 高等教育出版社, 1989.
- [119] Chua L. O., Desoer C. A., Kuh E. S., Linear and Nonlinear Circuits, McGraw-Hill, 1987.
- [120] 董在望, 尹达衡, 模拟集成电路原理与系统 (开关电容部分), 高等教育出版社, 1987.
- [121] Walscharts H., Gielen G., Sansen W., "Symbolic Simulation of Analog Circuits in s- and z-domain", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1989, pp. 814—817.
- [122] Walscharts H., "Silicon Compilation for Switched Capacitor Filters", Ph. D. Dissertation, Katholieke Universiteit Leuven, 1991.
- [123] 丘维声, 高等代数 (上册), 高等教育出版社, 1996. 6.
- [124] Liberatore A., Manetti S., "SAPEC—A Personal Computer Program for The Symbolic Analysis of Electric Circuits", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1988, pp. 897—900.
- [125] 杨华中, 汪慧, 刘润生, 范崇治, 基于状态方程行为描述的模拟集成电路的自动综合, 电子学报, Vol. 26(2), 1998年2月, pp. 53—56.
- [126] 杨华中, 汪慧, 范崇治, 刘润生, S-域行为描述的模拟集成电路综合方法, 电子学报, Vol. 24(5), 1996年5月, pp. 62—67.
- [127] Pohlmann W., "A Silicon-Bipolar Amplifier for 10 Gbit/s with 45 dB Gain", IEEE J. of Solid-State Circuits, Vol. SC-29(5), May 1994, pp. 551—556.
- [128] Berkcan E., Yassa F., "Towards Mixed Analog/Digital Design Automation; A Review", in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1990, pp. 809—815.
- [129] Mentor Graphics Corp., AccuSim II HDL-A/DEV User's and Reference Manual, V8.4-1, 1994.
- [130] Dumlugol D., Webber D., "Analog Modeling Using Event-Driven HDL's", in Proc. 7th Intern. Conf. VLSI Design, Jan. 1994, pp. 53—56.
- [131] Chadha R., Visweswariah C., Chen C-F., "M³—A Multilevel Mixed-Mode Mixed A/D Simulator", IEEE Trans. Computer-Aided Design, Vol. CAD-11(5), May 1992, pp. 575—585.
- [132] Openheim A. V., Schafer R. W. 著, 董士嘉, 杨耀增 译, 数字信号处理, 科学出版社, 1986.
- [133] Analogy, Inc., "MAST Language Reference Manual", Release 4.1, 1996.
- [134] Fitzpatrick D., Miller I., "Analog Behavioral Modeling with The Verilog-A Language", Kluwer Academic Publishers, 1998.
- [135] 杨华中, UNIX 应用教程 (第 10.8 节), 人民邮电出版社, 1996. 9.
- [136] Johnson S. C., "YACC: Yet Another Compiler Compiler", Computing Science Technical Report No. 32, 1975, Bell Lab., Murray Hill, NJ 07974. See Also, Sourble Computer Inc., UNIX Programmer's Reference Manual, 1990.
- [137] Geiger R. L., Sanchez-Sinécio E., "Active Filter Design Using Operational Transconductance Amplifiers: A Tutorial", IEEE Circuits and Devices Magazine, Vol. 1(2), Mar. 1985, pp. 20—30.
- [138] Tsividis Y., Banu M., Khoury J., "Continuous-Time MOSFET-C Filters in VLSI", IEEE Trans. Circuits And Systems, Vol. CAS-33(2), Feb. 1986, pp. 125—140.
- [139] Gregorian R., Martin K. W., Temes G. C., "Switched-Capacitor Circuit Design", Proc. IEEE, Vol. 71(8), Aug. 1983, pp. 941—966.
- [140] Toumazou C., Hughes J. B., Battersby N. C., Switched-Currents, an Analog Technique for Digital Technology, London: Peter Peregrinus Ltd., 1993.
- [141] Adams W. J., Ramirez-Angulo J., "Extended Transconductance Adjustment/Linearisation Technique," Electronics Letters, Vol. 27(10), May, 1991, pp. 842—844.
- [142] Parker A. E., Haigh D. G., "Compensation of 2nd Harmonic Distortion in A 4-FET Linearised Transconductor Circuit," in Proc. IEEE Intern. Sym. Circ. Syst. (ISCAS), May 1993, pp. 1089—1092.
- [143] Onodera K., Fujii N., Takagi S., Hotta M., "Design of Low-Distortion MOS OTA Based on Cross-Coupled

- Differential Amplifier and Its Application for Active Filters", *IEICE Trans. Fundamental*, Vol. E77-A (2), Feb. 1994, pp. 363-370.
- [144] Takagi S., Fujii N., "Novel Highly Linear MOS Integrator Using a Negative Impedance Converter (NIC)", *Electronics Letters*, Vol. 30 (10), May 1994, pp. 746-748.
- [145] Takagi S., Czarnul Z., Fujii N., "A Synthesis of Highly Linear MOS Circuits and Their Applications to Filter Realization", *IEICE Trans. Fundamentals*, Vol. E77-A (2), Feb. 1994, pp. 351-355.
- [146] Stevens A. E., Miller G. A., "A High-Slew Integrator for Switched-Capacitor Circuits", *IEEE J. Solid-State Circuits*, Vol. 29 (9), Sept. 1994, pp. 1146-1149.
- [147] K. Kimura, "A Bipolar Four-Quadrant Analog Quarter-Square Multiplier Consisting of Unbalanced Emitter-Coupled Pairs and Expansions of Its Input Ranges", *IEEE J. Solid-State Circuits*, Vol. SC-29, No. 1, Jan. 1994, pp. 46-55.
- [148] Cohen M. H., Andreou A. G., "Analog CMOS Integration and Experimentation with An Autoadaptive Independent Component Analyzer", *IEEE Trans. Circuits and Systems II*, Vol. CAS-42 (2), Feb. 1995, pp. 65-77.
- [149] Ishizuka O., Tang Z., Matsumoto H., "MOS Sine Function Generator Using Exponential-Law Technique", *Electronics Letters*, Vol. 27 (21), Oct. 1991, pp. 1937-1939.
- [150] Dent A. C., Cowan C., "Linearization of Analog-to-Digital Converters", *IEEE Trans. Circuits and Systems*, Vol. CAS-37 (6), June 1990, pp. 729-737.
- [151] Fattarusio J. W., Meyer R. G., "MOS Analog Function Synthesis", *IEEE J. Solid-State Circuits*, Vol. SC-22 (6), Dec. 1987, pp. 1056-1063.
- [152] Sanches-Sinencio E., Ramirez-Angulo J., Linares-Barranco B., Rodriguez-Vazquez A., "Operational Transconductance Amplifier-Based Nonlinear Function Synthesis", *IEEE J. Solid-State Circuits*, Vol. SC-24 (6), Dec. 1989, pp. 1576-1586.
- [153] Fanttaruso J. W., Meyer R. G., "Triangle-to-Sine Wave Conversion with MOS Technology", *IEEE J. Solid-State Circuits*, Vol. SC-20 (4), Apr. 1985, pp. 623-631.
- [154] Klinke R., Fiedler H.-L., Hosticka B. J., Kokozinshi R., Munster I., "Rule-Based Analog Circuit Design", in *Proc. European Design Automation Conf. (EDAC)*, Sept. 1992, pp. 480-4.
- [155] Kruiskamp W., "DARWIN: CMOS OPAMP Synthesis by Means of Genetic Algorithm", in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, June 1995, pp. 25.1-4.
- [156] Goodenough F., "Dozing IC Op Amps Wake Up for Input Signal", *Electronic Design*, Vol. 39 (23), Dec 1991, pp. 49-50, 52-54.
- [157] Wright J., "Detail Design Trims Video Amp Performance", *Electronic Product Design*, Vol. 12 (12), 1991, 33-34.
- [158] Schaumann R., Ghausi M. S., Laker K. R., *Design of Analog Filters*, Englewood Cliffs, NJ; Prentice-Hall, 1990.
- [159] Steyaert M., Sansen W. M. C., "Low-Power Monolithic Signal-Conditioning System", *IEEE J. Solid-State Circuits*, Vol. SC-25 (2), Apr. 1990, pp. 609-612.
- [160] Eschauzier R. G. H., Huijsing J. H., *Frequency Compensation Techniques for Low-Power Operational Amplifiers*, Kluwer Academic Publishers, 1995.
- [161] Sakurai S., Ismail M., *Low-Voltage CMOS Operational Amplifiers, Theory, Design and Implementation*, Kluwer Academic Publishers, 1995.
- [162] van de Plassche R. J., Sansen W. M. C., Huijsing J. H., (ed.), *Analog Circuit Design, Low-Power, Low-Voltage, Integrated Filters and Smart Power*, Kluwer Academic Publishers, 1995.
- [163] Sansen W. M. C., Huijsing J. H., van de Plassche R. J., (ed.), *Analog Circuit Design, Mixed A/D circuit Design, Sensor Interface Circuits and Communication Circuits*, Kluwer Academic Publishers, 1994.
- [164] Ning Z.-Q., et al, "Analog Circuit Design Automation for Performance", in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, New York, May 1992, pp. 8.2/1-4.

- [165] 杨华中, 汪蕙, 刘润生, 吴佑寿, 一种适于自动综合的高性能双极型运放的超级电路, 清华大学学报 (自然科学版), 第 37 卷第 10 期, 1997 年 10 月, pp. 37—40, 44.
- [166] Ribner D. B., Copeland M. A., "Design Techniques for Cascoded CMOS Op Amps with Improved PSRR and Common-Mode Input Range", IEEE J. Solid-State Circuits, Vol. SC-19 (6), Dec. 1984, pp. 919—925.
- [167] Laker K. R., Sansen W. M. C., Design of Analog Integrated Circuits and Systems, McGraw-Hill, Inc., 1994.
- [168] Cherry E. M., Hoper D. E., "The Design of Wide-Band Transistor Feedback Amplifiers", IEE Proc. Part G, Vol. 110 (1), Feb. 1963, pp. 375—389.
- [169] Ishii K., Ichino H., Kobayashi Y., Yamaguchi C., "High-Bit-Rate, High-Input-Sensitivity Decision Circuit Using Si Bipolar Technology", IEEE J. Solid-State Circuits, Vol. SC-29 (5), May 1994, pp. 546—50.
- [170] Choi T., Kaneshiro R., Brodersen R., Gray P., Jett W., Wilcox M., "High-Frequency CMOS Switched-Capacitor Filters for Communications Application", IEEE J. Solid-State Circuits, Vol. SC-18 (6), Dec. 1983, pp. 652—64.
- [171] Haspeslagh J., Sansen W. M. C., "Design Techniques for Fully Differential Amplifiers", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), May 1988, pp. 12.2/1—4.
- [172] M. Banu, Khoury M., Tsividis Y., "Fully Differential Operational Amplifier with Accurate Output Balance", IEEE J. Solid-State Circuits, Vol. SC-23 (6), Dec. 1988, pp. 1410—4.
- [173] Pelgrom M., Duinmaier A., Aad C. J., Welbers A., "Matching Properties of MOS Transistors", IEEE J. Solid-State Circuits, Vol. SC-24 (5), Oct. 1989, pp. 1433—1440.
- [174] Trnka J., Hedman R., Koehler G., Ladin K., "A Device Level Auto Place and Wire Methodology for Analog and Digital Masterslices", in Proc. IEEE Intern. Solid State Circuits Conference (ISSCC), Feb. 1988, pp. 260—261, 396—397.
- [175] Berkcan E., d'Abreu M., "Physical Assembly for Analog Compilation of High Voltage ICs", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1988, pp. 14.3/1—7.
- [176] Berkcan E., Kim C. K., Carrin B., d'Abreu M., "From Analog Description to Layout: A New Approach to Analog Silicon Compilation", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1989, pp. 4.4/1—4.
- [177] Kim C. K., Berkcan E., et al., "A New Floor Planning Algorithm for Analog Circuits", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1989, pp. 3.2/1—4.
- [178] Smith L., et al., "A CMOS-Based Analog Standard Cell Product Family", IEEE J. Solid-State Circuits, Vol. SC-24 (2), April 1989, pp. 370—379.
- [179] Barlow A., Takasuka K., et al., "An Integrated Switched Capacitor Filter Design System", in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 1989, pp. 4.5/1—5.
- [180] Mogaki M., Kato N., Chikami Y., Yamada N., Kobayashi Y., "LADIES: An Automatic Layout System for Analog LSI", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1989, pp. 450—453.
- [181] Sechen C., "Chip-Planning, Placement and Global Routing of Macro/Custom Cell Integrated Circuits Using Simulated Annealing", in Proc. ACM/IEEE Design Automation Conf. (DAC), 1988, pp. 73—80.
- [182] Gyurcsik R. S., Jeon J. C., "A Generalized Approach to Routing Mixed Analog and Digital Signal Nets in a Channel", IEEE J. Solid-State Circuits, Vol. SC-24 (2), April 1989, pp. 436—442.
- [183] Mehranfar S. W., "A Technology-Independent Approach to Custom Analog Cell Generation", IEEE J. Solid-State Circuits, Vol. SC-26 (4), Mar. 1991, pp. 386—393.
- [184] Wong D., Liu C., "A New Algorithm for Floor-Plan Design", in Proc. ACM/IEEE Design Automation Conf. (DAC), 1986, pp.
- [185] Wong D., Leong H., Liu C., (editors), Simulated Annealing for VLSI Design, Kluwer Academic Publishers, 1988.
- [186] Ohtsuki T., Layout Design and Verification, North Holland, 1986.
- [187] Bern M., Plassmann P., "Steiner Problem with Edge Lengths 1 and 2", Information Processing Letters, Vol. 32 (4), Sept. 1989, pp. 171—176.

- [188] Hightower D. W. , "A Solution to Line-Routing Problem on the Continuous Plane", in Proc. ACM/IEEE Design Automation Conf. (DAC), 1969, pp. 1—24.
- [189] Heyns W. , Sansen W. , Beke H. , "A Line-Expansion Algorithm for The General Routing Problem with A Guaranteed Solution", in Proc. ACM/IEEE Design Automation Conf. (DAC), 1980, pp. 243—249.
- [190] Okuda R. , Sato T. , Onodera H. , Tamaru K. , "An Efficient Algorithm for Layout Compaction Problem with Symmetry Constraints", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1989, pp. 148—151.
- [191] Ousterhout J. K. , "Corner Stitching: A Data-Structuring Technique for VLSI Layout Tools", IEEE Trans. Computer-Aided Design, Vol. CAD—3 (1), Jan. 1984 pp. 87—100.
- [192] Ohisaki T. , Maze-Running and Line-Search Algorithms in Layout Design and Verification (pp. 121—128), North Holland, 1986.
- [193] Garrod D. J. , Rutenbar R. A. , Carley L. R. , "Automatic Layout of Custom Analog Cells in ANAGRAM", in Proc. IEEE Int. Conf. Computer Aided Design (ICCAD), Nov. 1988, pp. 544—547.
- [194] Poirier C. J. , "Excellerator: Custom CMOS Leaf Cell Layout Generator", IEEE Trans. Computer-Aided Design, Vol. CAD—8 (7), July 1989, pp. 176—179.
- [195] Ghione G. , "Efficient, CAD-Oriented Model for The Characteristic Parameters of Multiconductor Buses in High Speed Digital GaAs ICs", Analog Integrated Circuits and Signal Processing, Vol. 5 (1), Jan. 1994, pp. 67—75.
- [196] 张凤言, 模拟乘(除)法器——分析、参数与应用, 科学出版社, 1988.
- [197] IEEE 1076.1 Working Group, "VHDL 1076.1 Design Objective Document", IEEE Press, 1995.
- [198] IEEE, "IEEE Standard VHDL Language Reference Manual", ANSI/IEEE Std. 1076, 1993.

[G e n e r a l I n f o r m a t i o n]

书名 = 模拟集成电路的自动综合方法

作者 = B E X P

页数 = 1 3 6

下载位置 = <http://202.118.180.121/ebook/sjy01/diskyd/yd09/12/!00001.pdg>