

目 录

第一章 设计方法分类和计算机辅助的作用.....	1
§ 1.1 集成电路的设计要求	1
§ 1.2 集成电路的设计流程	3
§ 1.3 设计方法分类	4
§ 1.4 专用集成电路与设计方法	6
§ 1.5 各设计阶段计算机辅助的应用	7
§ 1.6 设计系统与设计软件	9
第二章 全定制、定制和半定制设计方法.....	11
§ 2.1 全定制版图设计方法	11
§ 2.2 符号法版图设计	11
§ 2.3 门阵列设计方法	13
§ 2.4 标准单元法	22
§ 2.5 通用单元法	26
§ 2.6 PLD 方法.....	26
§ 2.7 不同设计方法的比较	32
§ 2.8 混合模式设计方法	32
§ 2.9 层次式设计概念	34
第三章 硅编译器.....	39
§ 3.1 概述	39
§ 3.2 硅编译器与语言编译的比较	39
§ 3.3 硅编译器的输入描述语言	40
§ 3.4 硅编译器的内部组成	42
§ 3.5 软件环境与设计步骤	45
§ 3.6 简单实例——随机逻辑编译器	46
§ 3.7 硅编译器的优点与限制	51
第四章 逻辑模拟.....	52
§ 4.1 逻辑模拟程序分类	52
§ 4.2 逻辑模拟的一些基本概念	53
§ 4.3 逻辑模拟的方法	56
4.3.1 编译方式.....	56
4.3.2 表格驱动和面向事件方式.....	58
4.3.3 硬件加速方式.....	62
§ 4.4 SPLICE 模拟程序及其应用	63
§ 4.5 SPLICE 使用举例	65

第五章 时序分析	67
§ 5.1 时序分析的目的	67
§ 5.2 时序分析的方法	68
§ 5.3 MOTIS-C 时序分析程序及其应用	74
§ 5.4 MOTIS-C 使用举例	76
第六章 电路模拟	78
§ 6.1 电路分析的作用	78
§ 6.2 SPICE 电路模拟程序及其应用	78
§ 6.3 SPICE 使用举例	80
§ 6.4 器件模型及各参数的意义	82
6.4.1 二极管模型	82
6.4.2 双极型晶体管模型	83
6.4.3 结型场效应晶体管模型	89
6.4.4 MOS场效应晶体管模型	90
6.4.5 砷化镓 MESFET 场效应管模型	102
§ 6.5 模型参数的提取	102
第七章 电路模拟程序的内部结构和求解过程	108
§ 7.1 电路模拟程序的基本组成	108
§ 7.2 电路模拟程序的流程	109
§ 7.3 输入处理	110
§ 7.4 建立电路方程	118
§ 7.5 求解方法和过程	123
7.5.1 线性电路的直流分析	123
7.5.2 非线性电路的直流分析	126
7.5.3 交流分析	128
7.5.4 瞬态分析	129
§ 7.6 牛顿迭代解的收敛性改进	136
第八章 器件模拟	139
§ 8.1 器件模拟的目的	139
§ 8.2 器件模拟的内容	140
§ 8.3 器件模拟的求解方法	141
§ 8.4 物理参数的模型	149
§ 8.5 SEDAN II 器件模拟程序及其应用	151
§ 8.6 SEDAN II 程序应用举例	155
第九章 工艺模拟	157
§ 9.1 工艺模拟的作用	157
§ 9.2 工艺模拟的求解方法	158
§ 9.3 SUPREM II 工艺模拟程序及其应用	162
§ 9.4 SUPREM II 工艺模拟程序应用举例	164

§ 9.5 SUPREM II 中的工艺模型.....	167
第十章 计算机辅助版图设计.....	179
§ 10.1 版图设计系统的构成.....	179
§ 10.2 版图编辑器.....	180
§ 10.3 版图的生成与编辑.....	182
§ 10.4 版图验证和分析.....	185
§ 10.5 图形发生器文件.....	189
§ 10.6 格式的交换.....	189
附录 I 算法基础.....	191
附录 I CIF 格式.....	201
参考文献.....	204

第一章 设计方法分类和计算机辅助的作用

§ 1.1 集成电路的设计要求

自1959年以来,集成电路技术发生了惊人的变化。第一个设计出来的集成电路只有四个晶体管,而三十年以后的今天,在1989年,一个芯片上集成的晶体管数目已超过一千万个。

集成电路经历了SSI、MSI、LSI、VLSI阶段,目前已开始进入特大规模集成电路ULSI(Ultra Large Scale Integration)阶段。各阶段的主要特征可见于表1-1。

表 1-1 集成电路各阶段的特征

发 展 阶 段 主 要 特 征	MSI	LSI	VLSI	ULSI
元件数/芯片	$10^2 \sim 10^3$	$10^3 \sim 10^5$	$10^5 \sim 10^7$	$10^7 \sim 10^8$
特征线宽(μm)	10~5	5~3	3~1	<1
速度功耗乘积(μJ)	$10^3 \sim 10$	10~1	$1 \sim 10^{-2}$	$< 10^{-2}$
栅氧化层厚度(nm)	120~100	100~40	40~15	15~10
结 深(μm)	2~1.2	1.2~0.5	0.5~0.2	0.2~0.1
芯片面积(mm^2)	<10	10~25	25~50	50~100
被加工硅片直径(mm)	50~75	100~125	150	>150

随着集成技术的发展和集成度的迅速提高,集成电路芯片的设计越来越复杂,原有的传统方法——手工画图、刻红膜的方法已无法适应,急需在设计方法与设计工具方面来一个大的变革。这就是经常谈论的“设计革命”。各种计算机辅助设计手段的涌现以及设计方法学的诞生正是为了适应这种要求。

设计一个芯片时有以下几方面的要求。

第一是设计时间。设计时间不仅影响设计费用和产品成本,而且会由于设计时间过长,而失去新产品投入市场的机会。

在芯片设计中,版图设计尤其花费时间。按常规方法,一个版图绘制人员每天约画5~10个器件(包括检查和修改)。如果我们要设计一个有一万个MOS晶体管的芯片就需要40~80个人月;如设计一个有十万个器件的芯片则大约要花60个人年。而这又不可能单纯依靠增加版图设计人员来缩短设计周期。

微处理器Z8000的设计是一个典型例子。它采用N沟硅栅工艺,晶体管总数为17,500个,芯片面积为39.3 mm^2 。主要依靠人工来完成,总共要花费13,000个人时,其中约50%的工作量即6,600个人时用于版图设计。各设计阶段所花费的时间见表1-2。

第二是设计的正确性。目前,在一个芯片上所集成的是很复杂的电路甚至是一个子系统,要对这样的芯片进行一次修改,所要花费的代价是昂贵的,因此必须保证设计的正确

• 1 •

表 1-2 设计 Z80 和 Z8000 所花费的人月数

设计各阶段	品种 花费时间	Z80	Z8000	Z8000 芯片设计师的工作量在 总时间中所占比例
确定功能		5*	15*	40%
逻辑/电路设计		7*	11*	100%
版图设计		8* 10**	8* 30**	26%
测试程序编制		6*	10* 3**	65%

* 高级技术人员 ** 助理技术人员

无误。但随着集成度的增加,在设计中引入错误的几率也随着加大。对于小规模集成电路还可以采用人工检查,而对于大规模及超大规模集成电路即使投入大量的人力也难以检查出所有的错误。这就要求在设计每个阶段都要进行反复的验证和检查。

第三是设计成本。每个芯片的成本可以由下式计算而得:

$$C_T = \frac{C_D}{V} + \frac{C_F}{yn} \quad (1-1)$$

式中, C_D 为开发费用; C_F 为每片硅片的工艺成本; V 为生产数量; y 为平均成品率; n 为每片硅片上的芯片数目。

公式表明,对于小批量生产应减小开发费用。而对大批量生产,应增加成品率和每一硅片上的芯片数。为了增加 y 和 n ,又必须减小每个芯片的尺寸。减小芯片尺寸的途径有:

- (1) 减少每个功能所需的电路数目;
- (2) 减少器件数和每个器件的尺寸;
- (3) 减少布局和布线中的所谓“死区”(无用区)。

第四是产品的性能。集成电路的性能主要决定于所选择的器件结构和电路形式,但是对于高速和低功耗的电路来说不能忽视版图设计对电路性能的影响。我们可以从下列公式中得出这一结论。

电路的延迟时间 T_{rd} 可表示为

$$T_{rd} = T_{rd0} + \frac{V_L(C_w + C_G)}{I_F} \quad (1-2)$$

电路的功耗 P_d 可表示为

$$P_d \propto \begin{cases} I_F V_{DD} & \text{对有比电路} \\ f(C_w + C_G + C_0)V_L^2 & \text{对无比电路} \end{cases} \quad (1-3)$$

式中, V_L 为逻辑摆幅; I_F 为晶体管峰值电流; V_{DD} 为电源电压; C_w 为连线电容; C_G 为总的扇出栅电容; C_0 为总的自电容; f 为工作频率。很明显, I_F 的增加会导致速度提高但却使功耗增加, C_w 的减小 (C_w 正比于连线长度) 会带来高速和低功耗。所以得到一个紧凑的版图,使连线长度减少对于获得高性能的产品是必须的。但紧凑版图的获得必然增

加设计工作量。

最后还必须考虑采用何种工艺技术。版图设计强烈地依赖于工艺技术水平，而这体现在版图设计所采用的设计规则上。

设计规则是规定生产工艺中可以接受的尺寸要求和可以达到的电学特性。因此设计规则既是工艺制造时应该达到的规范，也是设计人员在设计时必须遵守的准则。设计规则规定了最小线宽、最小容差及最小器件尺寸等，同时规定了设计者可以使用的参数范围，如开启电压、薄层电阻等的允许范围。随着工艺水平的提高，各种器件尺寸会不同程度地下降。因此要求设计一个高密度的、高成品率的产品时就需要一套相当复杂的设计规则。但对于生产量很小的产品，就可以采用简化的设计规则。

综上所述，一个集成系统的设计就是在保证产品质量的前提下，正确地选择器件结构、电路形式、工艺方案和设计规则，同时要尽可能地减小芯片尺寸，降低设计成本和缩短设计周期。

§ 1.2 集成电路的设计流程

当前典型的设计流程如图 1-1 所示。现有流程的主要特点是顺序地进行系统设计、逻辑设计、电路设计和版图设计，以及逐次地在各层次进行反复验证和比较。

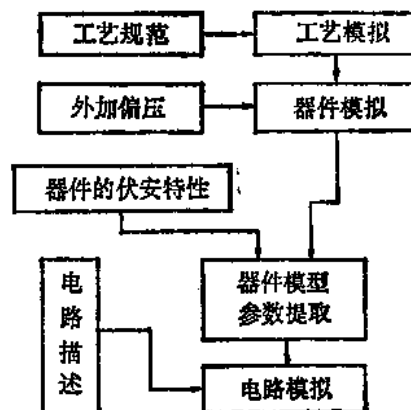
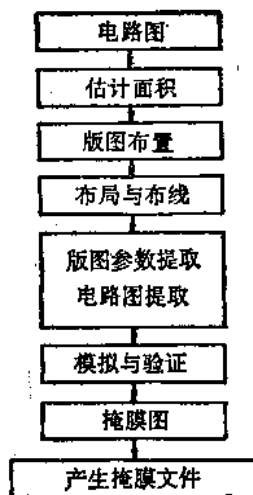
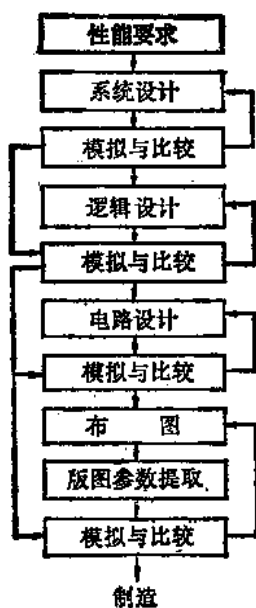


图 1-1 集成电路设计流程 图 1-2 布图的详细步骤 图 1-3 器件模型参数的提取过程

其中布图这一过程又可详细分解成以下几步，见图 1-2。布图以后的模拟常称为后模拟。它是把版图中的寄生参数和实际版图参数提取出来，再进行模拟，并与布图前的模拟结果相比较，看是否相符。若不符合设计要求，则应重新布图。电路图提取是为了验证实际的布图结果与原要求的电路图是否一致。

电路模拟时所需的器件模型参数可以设法从器件伏安特性的测量值中提取，也可以从

工艺模拟和器件模拟所得到的特性曲线中提取。这一过程示于图 1-3。

§ 1.3 设计方法分类

目前集成电路已渗透到各个应用领域。它的品种从微处理器、数字式电话一直到电子玩具和家用电器。由于品种的不同,在性能和价格上有很大差别,因而实现各种设计的方法和手段也就有所不同。

总的来讲,我们希望能在尽可能短的时间内用最低的成本来获得最佳的设计指标,而所用的芯片面积又是最小的。但实际上要全面达到这种要求是很困难的,只能进行某种折衷。

如果一个半导体制造厂想推出一种新的功能最强的微处理器芯片,就需要进行精心的设计。为了提高芯片的工作速度,就要采用最佳的随机逻辑网络,并把芯片设计得最紧凑,以节省每一小块面积。

但是有很多产品,产量不大或者不允许设计的时间过长,这时只能对芯片面积或性能作出某种牺牲,并尽可能采用一部分已有的、规则结构的版图。

为了争取时间或市场,也可先用很短的时间设计出芯片,在占领市场的过程中,再予以改进,进行一次再开发、再设计。

根据不同的设计要求,现有的各种设计方法可归纳如下:

一、全定制设计方法 (Full-Custom Design Approach)

它适用于要求得到最高速度、最低功耗和最省面积的芯片设计。这种方法通常采用随机逻辑网络,因为它能满足上述要求。但版图设计通常需要人来不断完善,以便把每个器件及内连接安排得最紧凑、最合适,因而特别花费时间。

Z8000 就是一个完全由随机逻辑网络组成的芯片(共 17,500 个晶体管),但是这样的芯片已不多见。目前很多芯片已采用或部分采用规则结构网络(Regular Structured Network),如 ROM、RAM 或 PLA,因为这类规则结构的版图设计比较容易,易于用自动设计手段实现。而部分随机逻辑网络也逐渐走向模块式结构(Modular-Structure),例如计数器、行波进位加法器等。这种标准的模块式结构的版图设计比非标准的随机逻辑要容易得多,且便于检查。70 年代末,在 LSI 芯片中随机逻辑网络的面积比例约占整个芯片面积的 30~40%。预计在 VLSI 芯片中这个比例会减小。但是应该指出,即使规则结构网络被广泛使用,随机逻辑网络仍对芯片的性能起着关键的作用。

为了提高全定制设计方法的效率,减少错误,现正采用层次式设计手段(Hierarchical Method)。

符号法版图设计法也属于全定制类。但实际上只能称它为“伪”全定制设计,因为要全局地减小芯片面积对符号法还有一定的困难。

二、定制设计方法 (Custom Design Approach)

它适用于芯片性能指标比较高而生产批量又比较大的产品设计。通常分为两种:

(1) 标准单元法 (Standard Cell Method),其商业名称为多元胞法(Polycell)。

(2) 通用单元法 (General Cell Method),或称为积木块法 (Building Block)。

以上通称库单元法。在设计时根据电路要求从库中调出所需单元和外围单元(包括

压焊块),进行自动布局和布线,最后得到被设计电路的掩膜版图。它的特点是元件得到充分利用,芯片面积较小,设计上的自由度较大。但建立一个单元库需要大量的初始投资。此外,制造周期较长,成本也较高。

三、半定制设计方法 (Semi-Custom Design Approach)

它适用于要求设计成本低、设计周期短而生产批量小的芯片设计。

门阵列 (Gate Array) 及最近发展起来的门海 (Sea of Gate) 就属于这一类。

门阵列是预先在芯片上生成由基本门 (或单元) 所组成的阵列,即完成连线以外的所有的芯片的加工步骤。

半定制的含意就在于对一批芯片作“单独处理”,即单独设计和制作接触孔和连线以完成电路的要求。这样就使从设计到芯片完成的整个周期大大缩短,因而设计和制造成本大大下降。但门阵列的门利用率一般较低,芯片面积较大。

四、可编程逻辑器件 (Programmable Logic Device) 法

其特点是“可编程”,往往由制造商提供通用器件,而由设计者根据需要进行“再加工”实现其特定的逻辑。如果说门阵列的“单独处理”由芯片制造商用 2~4 道掩膜工序来实现的话,PLD 的“可编程”则由设计者通过开发工具自己来完成,这就大大方便了用户。

可编程逻辑器件一般包括:

- (1) 可编程只读存储器 (PROM);
- (2) 可编程逻辑阵列 (PLA、FPLA);
- (3) 可编程阵列逻辑 (PAL);
- (4) 通用阵列逻辑 (GAL);
- (5) 其它可编程器件 (如可编程微顺序器、专用 PLD 等)。

五、混合模式设计方法

随着 VLSI 复杂性的增长,在整个芯片中只利用一种设计方法已认为是不经济的。因而提出了混合模式,即把不同的设计方法加以优化,并组合而构成一体。如近年提出的 IMSA(Integrated Modular and Standard cell design Approach) 就是一例。它把人工设计、标准单元法和 PLA 法用于一个芯片设计中,取得了满意的结果。

六、硅编译 (Silicon Compiler) 法

它是一种全自动的设计方法。利用这种方法可从集成电路的行为级描述直接得到该电路的掩膜版图。以硅编译器为基础的 VLSI 设计系统为系统设计人员提供了一种真正的设计自动化工具。

除以上六种外,还有如栅列阵 (Gate Matrix) 法等正在开发中的其它设计方法。

要对各种设计方法作出全面的比较是复杂的。现用图 1-4 说明采用不同方法时,成本与产量之间的关系。可以看出,为了得到合理的成本,不同的设计方法有不同的最小产量。对于全定制设计方法设计的芯片,只有当产量超过 10 万块时,它的价格才是可接受的。而对于门阵列芯片,生产量只要超过一万块时,就有明显的竞争能力。

从设计自动化的程度来比较不同的设计方法,可见图 1-5。其中硅编译法自动化程度最高,符号法的设计自动化程度最低。

还应该指出,自动化实现的难易与约束条件有关。一般讲,约束限制越多,实现自动化越容易。但约束条件会带来布局和布线上的种种限制,而导致芯片面积增加和性能降低。

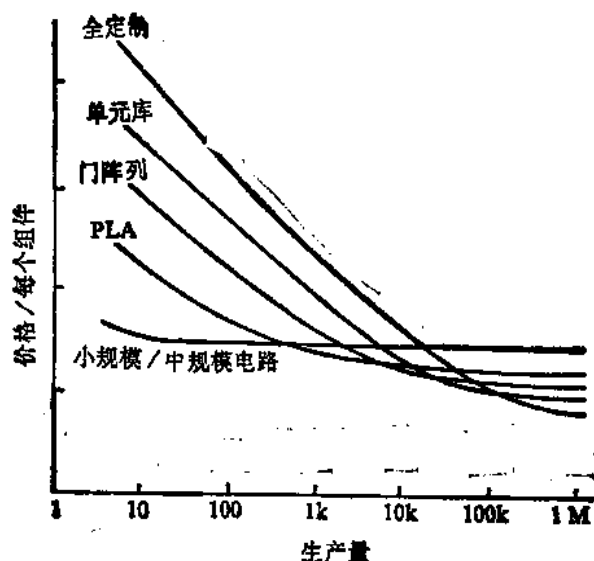


图 1-4 不同设计方法时成本与产量的关系

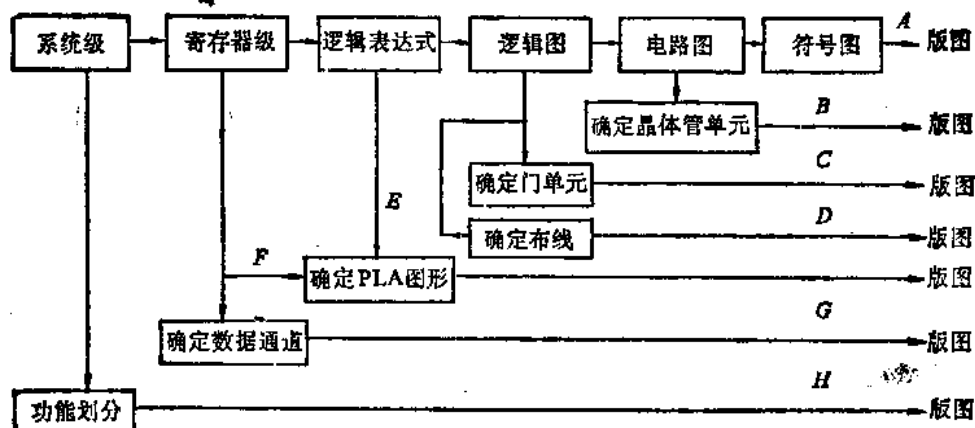


图 1-5 设计自动化程度的比较

A——符号法；B——晶体管单元法；C——标准单元法；D——门阵列法；E、F——PLA法；
G——数据通道法；H——硅编译法

§ 1.4 专用集成电路与设计方法

ASIC(Application Specific Integrated Circuit) 直译为“专用集成电路”。这种电路实际上早就存在。它面向专门用途，而区别于标准逻辑(Standard Logic)电路或通用存储器及通用微处理器电路。

这一术语得到广泛的传播和重视是在80年代中期以后。ASIC的提出和发展说明集成电路进入了一个新阶段。通用标准的集成电路和人工全定制电路已不能适应电子工业的急剧变化。各个厂家都希望生产具有特色、个性的产品，要求增加品种型号，加快产品更新。而只有ASIC才能迅速地生产出具有竞争力和保密性的新产品。

ASIC并不是一个学术名词，它的含义并不确切。广义地讲，几乎所有有专门用途的、

而又不属于标准逻辑电路或通用存储器及通用微处理器的、新开发的产品都可以称为 ASIC。ASIC 产品可以采用任何一种设计方法来实现。既可以采用全定制方法,也可以采用定制、半定制方法,或者用硅编译法。

目前对 ASIC 有另一种狭义的理解,即 ASIC 是根据用户的特定要求,能够以低研制成本、短的交货周期供货的半定制电路和定制电路。根据这一狭义的定义,实现 ASIC 的主要方法是门阵列和库单元法。

目前门阵列占 ASIC 产品的 60% 以上,已有系列化产品出现,规模可达数万门,十几万门的门阵列也已研制成功。门阵列现有 CMOS、NMOS、ECL、TTL 和 BiCMOS 等不同工艺。

标准单元目前的产值次于门阵列而列居第二位。但它的发展速度最快,最富有活力,因为可以在它的基础上开发含有 ROM、RAM、ALU 等的数字 ASIC,以及含有模拟电路的混合型 ASIC。标准单元也已有系列化产品出现。

ASIC 电路的蓬勃发展正在极大地推动着设计方法学的研究及设计工具的完善,同时也促使系统设计人员与芯片设计人员的结合与渗透。“门海”的出现就是一例。ASIC 中最广泛使用的是门阵列设计方法,但门阵列的固有弱点之一是通道固定,门的利用率低。为了解决这一问题,研究出了“门海”技术。在 ASIC 的研制方式上也更多地采用用户(系统设计人员)与厂商(芯片设计人员)共同结合,联合设计的方式。

§ 1.5 各设计阶段计算机辅助的应用

目前在集成电路的各个设计阶段,从系统设计一直到样片测试都已广泛采用计算机辅助。计算机辅助设计的作用是缩短设计时间,减少设计错误和反复,当需要改变设计时能做到及时而又方便。此外,计算机能用图表或文字报告分析及验证结果,并指出正确与否。

一个比较完整的设计系统示于图 1-6。

设计人员可以在彩色图形终端上,利用鼠标器或小数字化板,调用图形库中的各种逻辑符号来产生逻辑图,连接关系是根据逻辑网络的要求在系统功能键作用下,用连线连接而成。一旦完成了逻辑图,用户就可以将它存入逻辑数据库,以便今后调出进行检查和修改。

已产生的逻辑图可提交逻辑设计规则检查(如检查有无悬空点,是否违反扇出要求等),如检查无误可产生网单表(或称连接网表)。下一步可进行(版图前)逻辑模拟。数据库与现用的逻辑模拟程序之间有一接口。设计人员不必再书写逻辑模拟时所需的输入文件,而只要给出有关的命令和数据,以控制模拟的进行。

同样地,可以进行电路图的输入和电路模拟。在校完善的设计系统上,用户不仅可以调用图形库中的电路符号,甚至可以调用某一集成电路的管脚排列图,并利用数据库中该集成电路的性能信息,直接进行电路模拟。

在模拟工作完成后,设计人员可在图形终端上进行版图的生成和编辑。

对全定制设计系统,可利用图形库所提供的各种菜单和图形,通过人机交互方式,产生所需的直线、矩形、多边形等图形并进行增添、删除、放大缩小等编辑功能以产生所

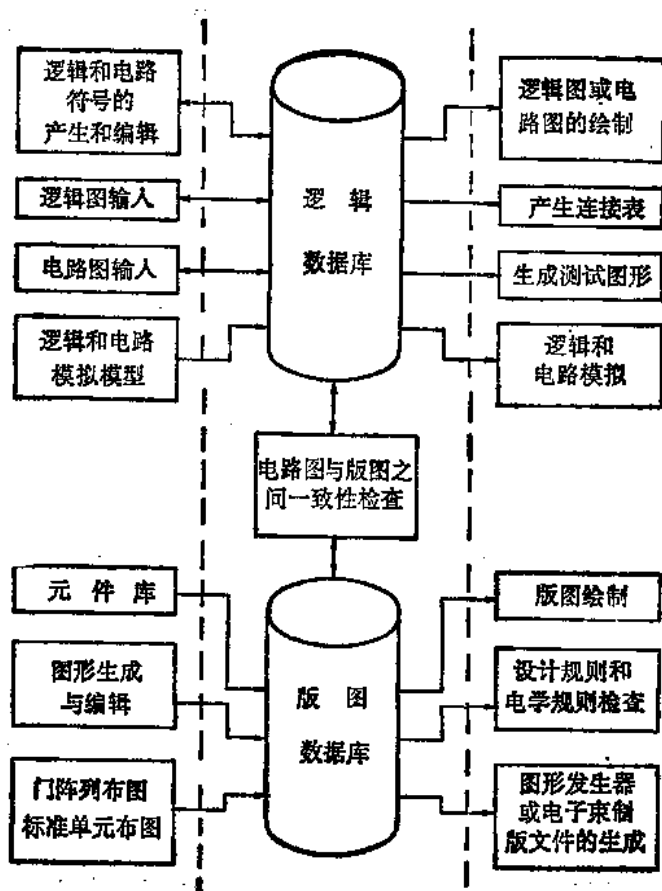


图 1-6 设计系统示意图

需的版图图形。

对于半定制、定制设计系统，由用户输入网单表，设计系统将自动进行布局布线，调用库内单元或宏单元完成正确的连接。如有个别地方无法自动连接，将进行人工干预。

在掩膜版图完成之后，将进行一系列的检查，“逆向”验证和后模拟（版图后）。

检查内容包括设计规则检查、电学规则检查、连线正确性检查。

“逆向”验证是将已得到的实际掩膜版图再次转换成电路图，并与原要求的电路图进行一次一致性比较。有的甚至将提取出来的电路图再次转换成逻辑图，并与原要求的逻辑图进行比较。

后模拟是将提取出来（版图设计后引入）的寄生参数加到模拟时的输入文件中，再进行一次模拟，以确认该版图设计是否符合产品的要求。因为过多寄生因素的引入，会使该设计达不到原来的要求。

计算机辅助除应用在电路设计阶段，另一重要应用是在器件和工艺设计阶段。

工艺计算机辅助是将各制造工序中的工艺规范作为输入，通过模拟确定杂质在各物理截面中的分布，这样可减少工艺“试流水”的盲目性，特别有利于制订合理的工艺规范。器件计算机辅助是将杂质分布和外加电压作为输入，提取出器件的电流-电压特性，及载流子和电位、电流分布。在了解器件特性和设计器件方面它已成为一种不可缺少的工具。近

年来已出现了将工艺模拟与器件模拟合在一起, 以及将它们再与电路模拟合在一起的一体化模拟程序, 从而进一步加强了计算机辅助设计的作用。

§ 1.6 设计系统与设计软件

近年来, 为了计算机辅助设计的需要, 开发了许多设计系统及软件工具。

一、人机交互图形系统

60 年代末首次出现了用于版图设计的人机交互图形系统。

它的主要功能有:

- (1) 交互式的图形输入和编辑, 并允许图形数据的多层嵌套。
- (2) 设计规则检查。
- (3) 转换到图形发生器或电子束制版设备的控制文件。

随着 VLSI 设计的需要, 这类系统的硬件配置逐渐从 16 位机转向 32 位机, 从小型机发展为超小型机。在软件方面增加了符号法 (包括符号法所用的压缩程序) 作为标准配置。其典型的硬件配置示于图 1-7。

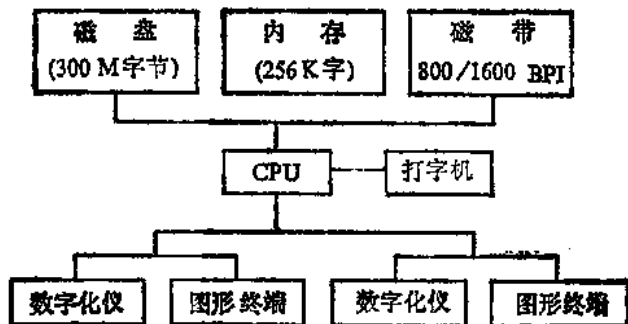


图 1-7 人机交互图形系统的硬件配置

二、CAE 工作站

80 年代初引入了设计工作站 (Engineering Workstation)。上述的人机交互图形系统, 一般称之为 CAD 系统, 实际上是一种灵巧的作图工具, 主要是针对版图设计的要求而开发的。而 CAE 工作站的开发则是为了给逻辑设计者提供一个充分的逻辑设计环境, 完成整个系统设计, 并允许与各种模拟程序合成一个整体。因此一般 CAE 系统都包括逻辑图 (电路图) 输入和逻辑模拟功能, 并进而扩展到电路模拟, 测试诊断等。初期的 CAE 工作在版图设计方面的功能不如上述的 CAD 系统, 一般只限于处理长方形和具有 45° 斜线的图形。

更完善的 CAE 工作站可以进行图形化的版图平面布置, 这时可完成高层次的模块布局, 在详细的版图设计以前先确定连线方案。在图形处理功能上也比早期的系统大大加强。

这类 CAE 工作站以一高性能的 CPU 作为控制器, 目前的工作站其主存为 1~8M 字节, 速度在 1 MIPS 以上。支持多窗口功能, 即在屏幕上有几个视口同时出现。一般这类工作站通过局部网与大型或巨型机相连。

随着CAE工作站的出现,最初形式的人机交互图形系统正被逐渐淘汰。CAE工作站中除了有全定制软件外,还逐渐加入了门阵列、标准单元法的自动设计软件,使功能进一步加强。

三、通用的软件工具

这类软件包括各类模拟分析用软件。它们有:功能级模拟、门级或开关级逻辑模拟、时序模拟、电路模拟、故障模拟、器件模拟、工艺模拟。

在验证软件方面,已有的商业用软件是:电学/设计规则检查、版图参数提取、版图与电路图一致性检查。

以上这些软件可以装入某一设计系统或单独运行于大型或巨型机上。

随着集成电路进入 ULSI 阶段,我们即将跨入一个新的领域,即集成系统 (System in Silicon) 领域。

这就要求在设计思想上有一个新的突破。设计人员必须在低层次上花费较少的时间,而把更多的精力集中于高层次即研究系统方面的问题。现有的 CAD 和 CAE 系统提供了设计常规芯片的功能,但依靠这种系统设计超级芯片是很困难的。因此新的系统必须具有一个统一的环境,以保证数据的可靠性和一致性,同时要求在各种设计工具之间具有快速的数据传递和流动,且具有方便的界面。这些要求将推动现有设计系统的进一步提高和新的软件工具的出现。

例如理想的设计方法将以综合处理整个设计为特征,如图 1-8 所示。

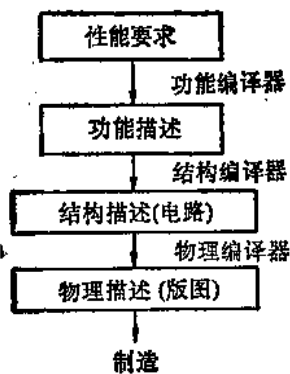


图 1-8 理想的设计方法

不久还将出现一种在算法难于处理情况下的程序设计方法,即专家系统,图 1-9 表示了这种专家系统的框图。

在新的模拟程序方面将会出现模拟小尺寸器件的新的二维、三维器件模拟程序,二维工艺模拟程序,以及模拟化合物半导体、异质结的器件模拟程序等。

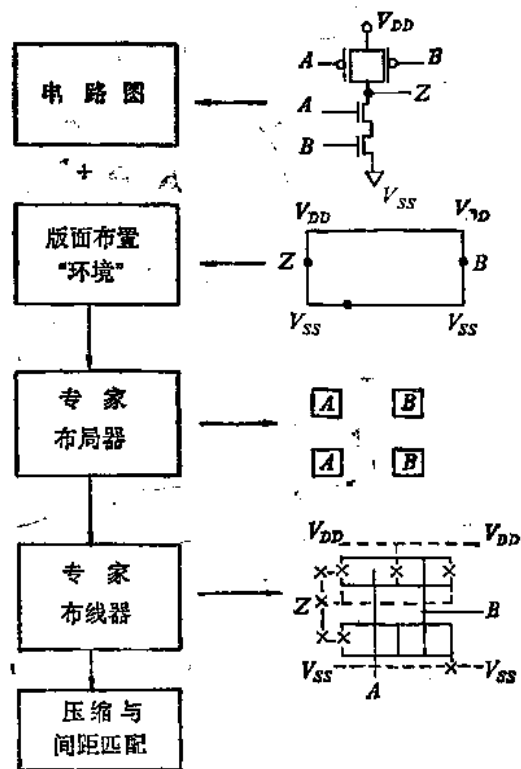


图 1-9 专家系统框图

第二章 全定制、定制和半定制设计方法

§ 2.1 全定制版图设计方法

这种方法是利用人机交互图形系统,由版图设计人员设计版图中的各个器件和连线。

全定制版图设计的特点是针对每个晶体管进行电路参数和版图参数的优化,这样可以得到最佳的性能(包括速度和功耗)以及最小的芯片尺寸。但由于它的效率很低,一个版图绘制人员每天约画 5~10 个器件,因而对于具有几十万或几百万个晶体管的超大规模集成电路来说,这种方法显然不适合。但对于具有重复性结构的网络如 RAM、RISC、阵列机等它仍然很有效。对这些网络中的单元进行精心的人工设计,然后利用人机交互图形系统所具有的重复或形成矩阵的功能得到整个网络的结构,其效率仍然很高。

采用这种方法时除了要求有人机交互的图形系统支持外,还要求有完整的检查和验证的CAD工具,因为设计人员不可避免会造成版图上的某些错误,需要通过自动的工具加以彻底地纠正。这些工具包括设计规则检查、电学规则检查、连接性检查、版图参数提取、版图与电路图一致性检查等。

§ 2.2 符号法版图设计

这是为了提高版图设计效率而开发的一种设计方法。它利用一组事先定义好的符号来表示版图中的晶体管、接触孔、多晶硅和铝引线等。典型的符号定义如图 2-1 所示。

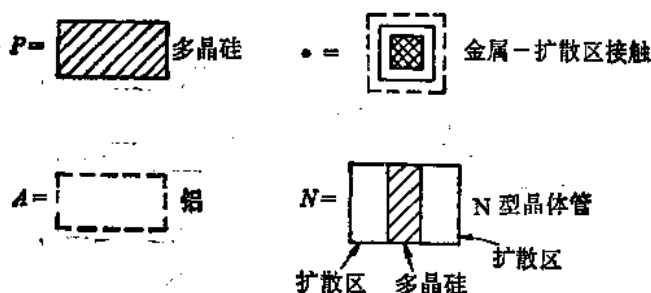


图 2-1 各种符号定义

设计人员根据网络要求画出一相应的符号图,自动转换程序再将这些符号自动地转换成版图图形。这时人们不必考虑版图规则的细节,也不存在违反设计规则的问题。

目前有三种符号法:(1)固定栅格式;(2)梗图式;(3)虚网格式。

固定栅格式是把芯片表面划分成均匀间隔的栅格。栅格大小表示最小的特征尺寸或布局上的容差。

设计人员根据要求将这些符号画在栅格图上,如图 2-2(a) 所示,经自动转换得到图

2-2(b) 所示的版图形式。

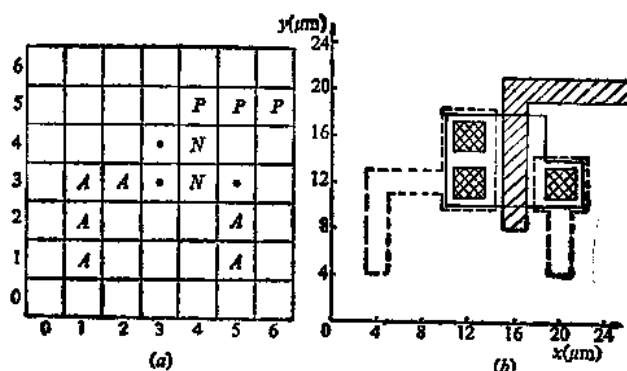


图 2-2 固定栅格的符号图及对应的掩膜形式

梗图式 (Stick Diagram) 是设计人员根据网络图画出一对应的梗形草图。符号法设计系统将其整理成规则的梗形图, 再进一步转换成版图形式。

由于梗形草图是自由画制, 符号之间的间距并不固定, 因而转换得来的版图可能有空余的空间, 这时符号法设计系统利用一种自动压缩程序在 x 和 y 方向分别将版图压缩到设计规则所允许的最小间距。整个过程示于图 2-3。

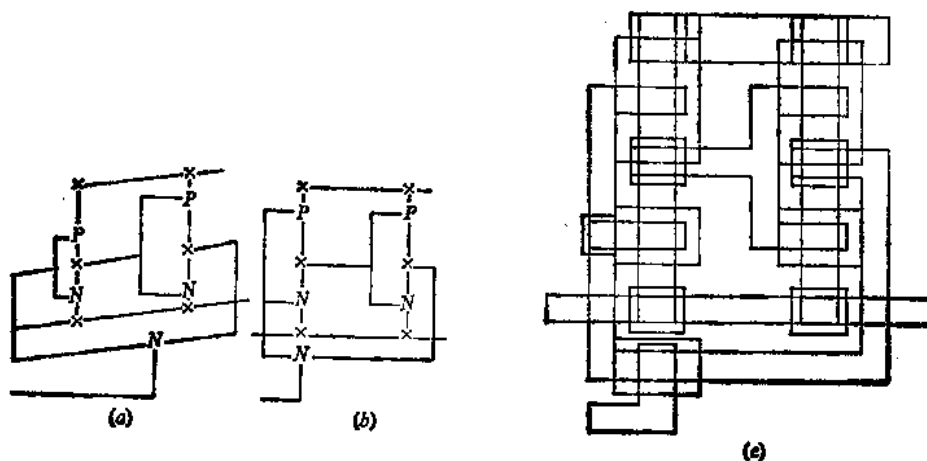


图 2-3 梗形图与对应的版图

虚网格符号法是在吸收了其它符号法的经验基础上发展起来的。

这里引入了一个“虚”网格的概念, 即网格线之间的最终间距取决于相邻网络上电路元件密度和相互关系, 它不再是固定的。这个概念通过图 2-4 可以得到很好的说明。

图 2-4(a) 中有四个图形, 并分别对中于各个网格线。如果采用 10 个单位的固定栅格, 线宽取 10 个单位, 间距也取 10 单位, 则得到的掩膜图为图 2-4(b)。如果采用虚网格方式, 网格线间距将按拓扑关系而变, 就可得到图 2-4(c) 所示的掩膜图。对某些虚网格压缩器来讲, 还可得到如虚线 (指小方格) 所示的掩膜图。

这样, 对于设计人员来讲, 在网络上进行布局时不必考虑任何设计规则的要求。

与人工全定制版图设计相比, 符号法有明显的优点, 它的设计工作量明显下降, 版图

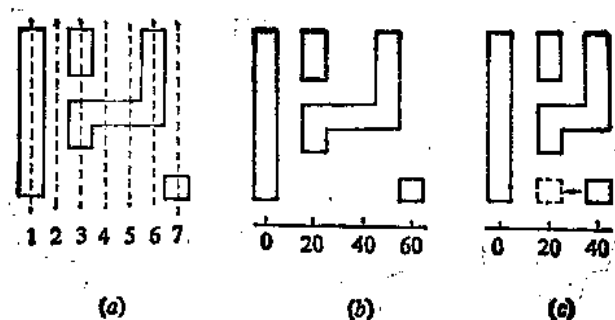


图 2-4 虚网格示意图

错误可显著减少，但仍保持很高的灵活性。另一突出优点是“工艺可更新”。因为设计描述是采用符号形式，基本的掩膜信息是从符号格式“译码”而来，所以只要修改相应的工艺文件，同一符号就可以得到不同尺寸的掩膜形式，这就大大有利于同一种电路的工艺更新。

符号法的最大困难在于设计编写一个功能完善的压缩程序。此外，版图布置上如不作全局优化，最后的封装密度可能较低，这也是符号法的问题。

符号法要求CAD系统能产生和编辑版图符号，辨认各种符号并进行转换，能进行自动的版图压缩。这种设计方法不需要设计规则检查工具。

§ 2.3 门阵列设计方法

门阵列是在一个芯片上把门排列成阵列形式。每个门具有相同的版图形状，门与门之间暂不相连，因此构成一个未完成的逻辑阵列。严格地讲，门阵列是把单元（若干器件）排列成阵列形式，每个单元内含有若干器件，通过连接单元内器件使每个单元实现某种类型门的功能，并通过各单元之间的连接实现电路的要求。

由于芯片内的单元是相同的，因此可以采用统一的掩膜，而且可完成连线以外的所有的芯片加工步骤（也就是金属化以前的所有工序），这样的芯片可以大量制造并储存起来。在需要时，可从中取出一部分加以“单独处理”。

所谓“单独处理”，就是根据网络的要求，考虑如何进行门的布局和门之间的连线，也就是对用于接触孔和连线的掩膜版（一般为2~4层掩膜）进行单独的设计和制作。然后再次进行工艺加工完成芯片的未完工序。因此可以把门阵列看成是一种掩膜可编程的功能器件。

可以看出，“单独处理”的制造工序只占总的制造工序的一小部分，因此从设计到芯片完成的整个周期大大缩短。一个典型的门阵列的制造周期是所有掩膜版都要单独设计和制作的常规芯片制造周期的一半。另外，制造成本大大下降，掩膜版所需的代价将是通常情况下的 $1/4 \sim 1/8$ 。

门阵列典型的版图布置见图 2-5。它有三个区域。芯片四周是 I/O 电路及压焊块，芯片中间为单元区和连线通道区。连线通道处于单元之间，连线为一系列垂直方向和水平方向的线段。如果门阵列允许有双层金属连线，金属层之间则通过“通孔”（Via）连接。一般第一层金属是水平的，第二层是垂直的。如果只允许单层金属，则水平线段为金属，垂直线段就必须采用多晶硅。以下具体讨论门阵列的各种单元和宏单元。

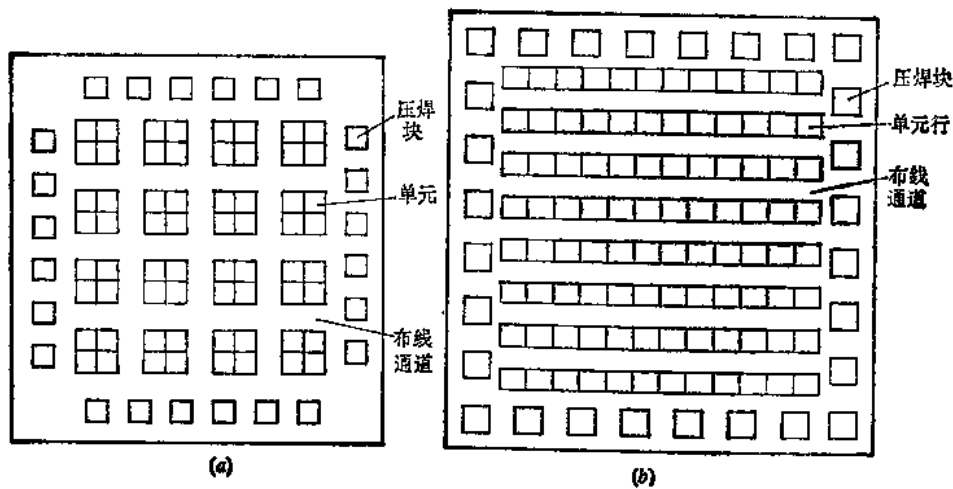


图 2-5 门阵列典型的版图布置

一、不同工艺的门阵列单元版图

1. CMOS单元的版图形式 图 2-6 是一个四管单元。利用该单元可构成二输入端与非门，其金属层及合成后的版图示于图 2-7。请注意，所有的输入和输出连线都引到单元的边界处，即连线通道的边缘，以便于进行单元之间的连线。将单元内部的连接与单元之间的连接分开处理，这体现了一种设计上的层次性。

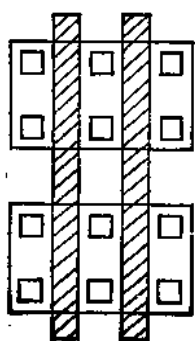


图 2-8 四管单元

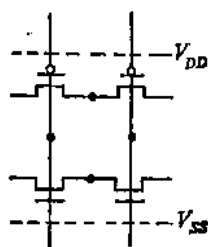


图 2-7 金属层与合成后版图

同样是四管单元也可以为另一形式，如图 2-8 所示。其中一对 NMOS 管和 PMOS 管的栅是相连的，另一对栅则是分开的。这就很容易实现锁存器中的传输门和倒相器。

门阵列中最简单的是四管单元，它是针对二输入与非门而优化设计的。但对于三输入与非门则要求采用两个单元，这时只利用了六个晶体管，其它两个晶体管的面积就浪费了。当然我们可以设计一种六管单元来支持三输入端与非门，但对单输入、二输入、四输入同样会产生无用晶体管。

看来好象二管单元最理想，因为任何类型的门都可用它建立起来而又不浪费晶体管。但实际上并不采用这种方案，因为每个单元都需要一个隔离环，它占有相当的面积，由于

这种隔离环的存在使二管单元的阵列总面积并不节约。

图2-9表示一种连续形式的MOS管阵列。它完全依靠金属连线的设计使单元具有应有的功能。图中的金属连线形式实现了一个二输入端与非门。

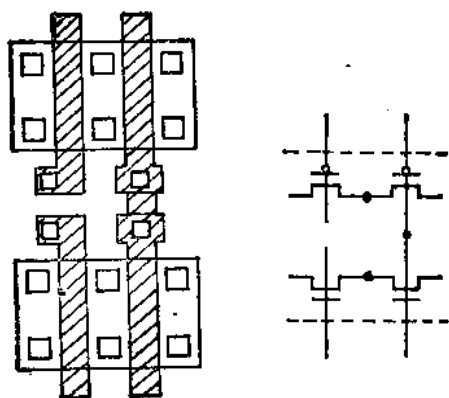


图 2-8 另一种形式的四管单元

单元中的MOS管可以有特定的尺寸,以满足某种功能的特殊需要。如图2-10的六管单元,其中两个管子尺寸较大,可实现静态锁存器的要求。

2. TTL门阵列的单元版图 如图2-11所示,它是一个四输入端的单元,由三个晶体管(其中一个为多发射极晶体管)和五个电阻组成。通过改变单元内的金属连线,可构成不同要求的门。例如:

(1) 低功耗门 用于驱动芯片内部的其它门,其电路图与对应的版图示于图2-12。

(2) 高功耗门 用于能驱动有多个扇出的内部门或作为外部门。其电路图与版图示于图2-13。

(3) 图腾输出门 用于高扇出或有长输出连线的输出门,这样的门由两个单元组成。单元A中利用三个晶体管 and 三个电阻。单元B只利用两个晶体管和一个电阻。两个单元之间的连接关系见图2-14。

(4) 扩展器 如图2-15所示。它是通过连接单元内的两个晶体管和一个电阻而

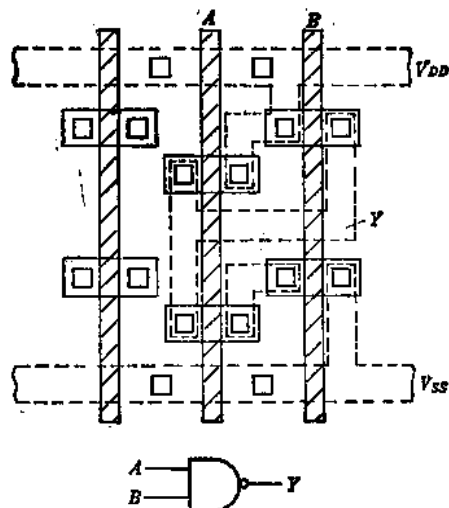


图 2-9 连续形式的MOS管阵列

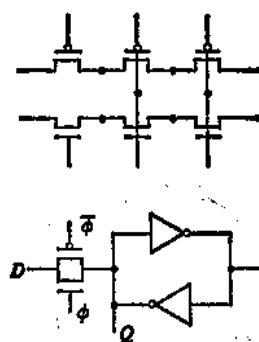


图 2-10 管子尺寸不同的六管单元

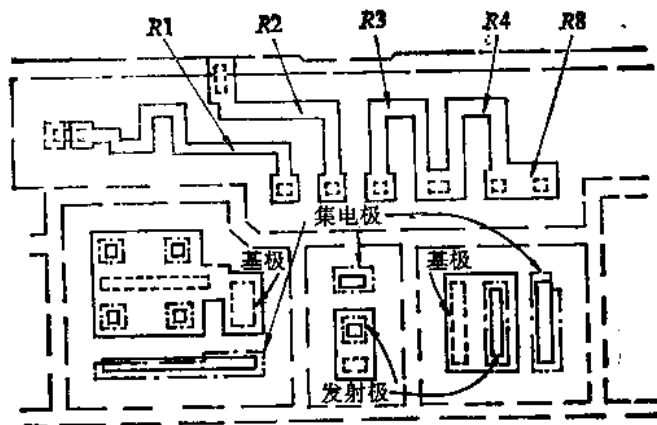


图 2-11 TTL 门阵列单元版图

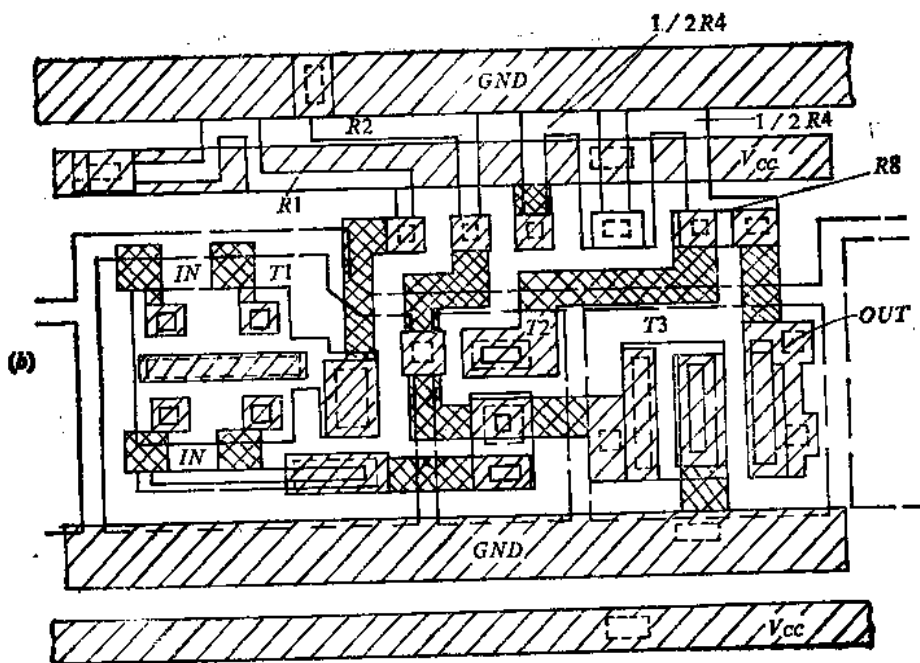
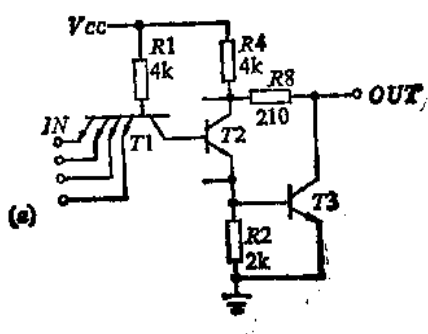


图 2-12 低功耗门的电路图与版图

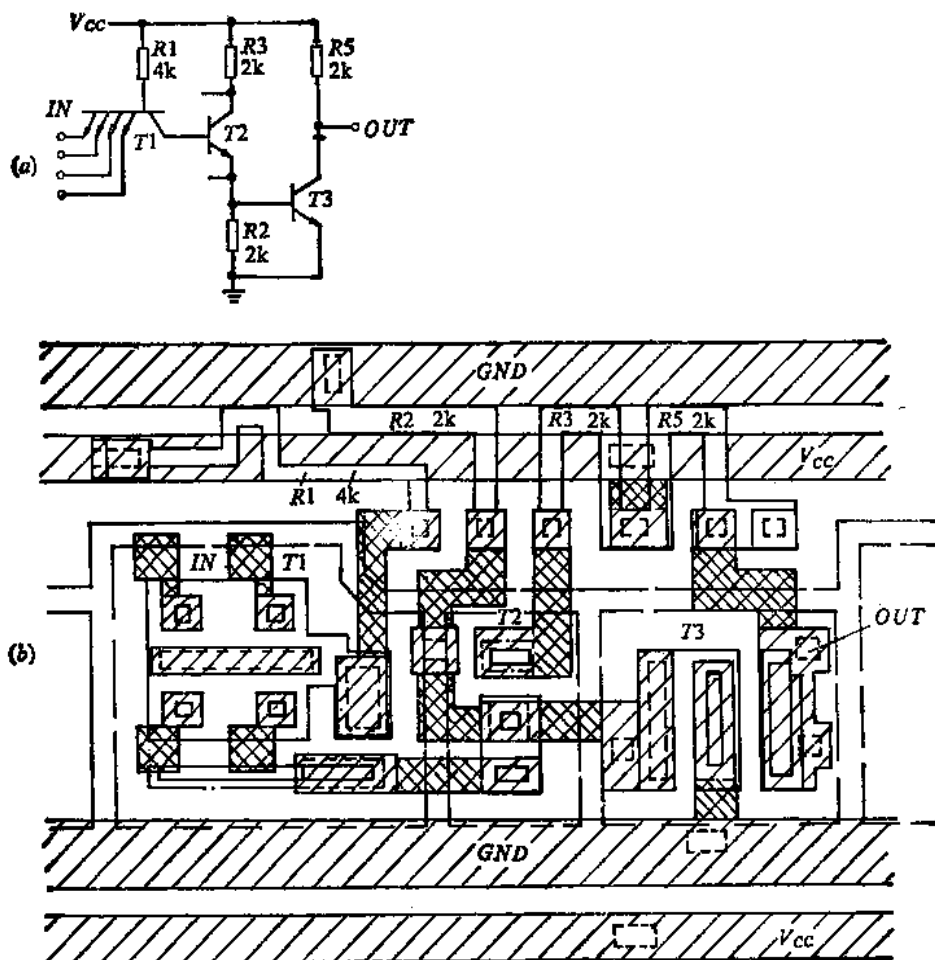


图 2-13 高功耗门的电路图与版图

成。

除了上述 CMOS, TTL 门阵列外, 还有其它形式的门阵列, 如 ECL, NMOS, PL, CMOS/SOS 等。

3. ECL 门阵列的单元版图 参见图 2-16, 利用它可以实现图 2-17 中的 ECL 门, 其中肖特基二极管被用于箝位。但这种版图采用三层金属连线。

4. E/D 型 NMOS 门阵列的单元版图 如图 2-18 所示, 它有四个增强型驱动管和一个耗尽型负载管。

二、宏单元阵列

宏单元是上述门阵列单元的一种发展。在宏单元阵列中, 可以实现晶体管级的逻辑设计, 例如门的串行选通。

图 2-19 是宏单元的一个例子。图中只描述了该单元的一半, 它含有 52 个双极型晶体管和 48 个电阻。通过连接单元中的各个元件可实现各种功能的门。例如图 2-20 就是利用这种单元得到的一个四输入端 ECL 异或门, 可以看出, 它实现了门的串行选通, 收集极

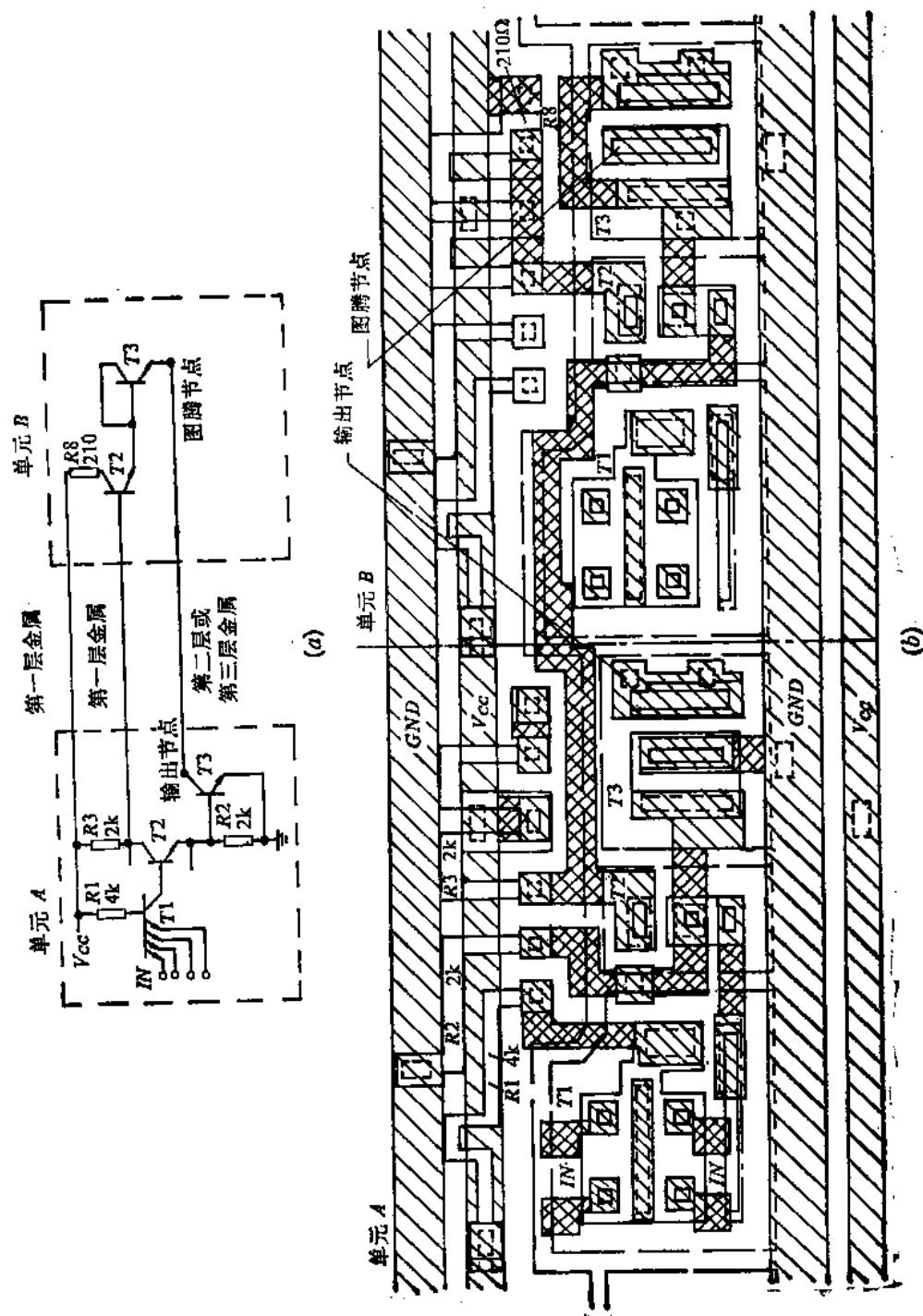


图 2-14 图腾门的电路图与版图

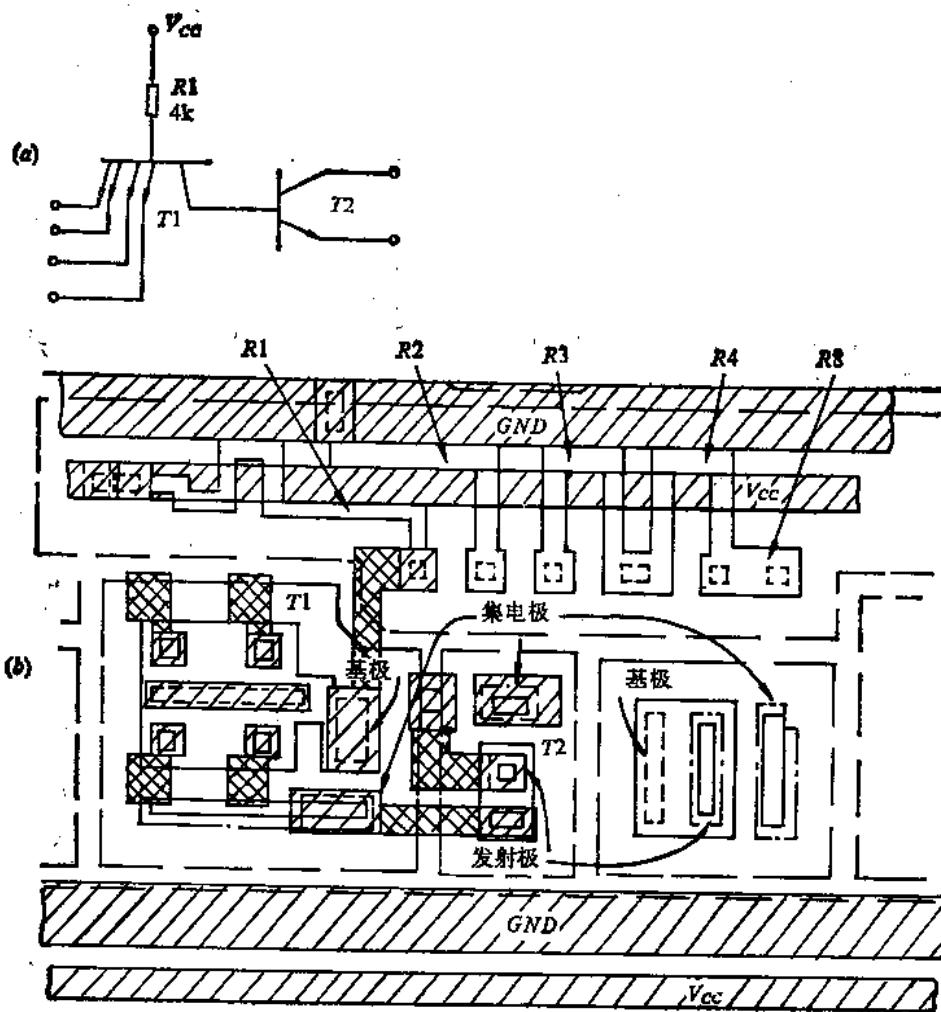


图 2-15 扩展器的电路图与版图

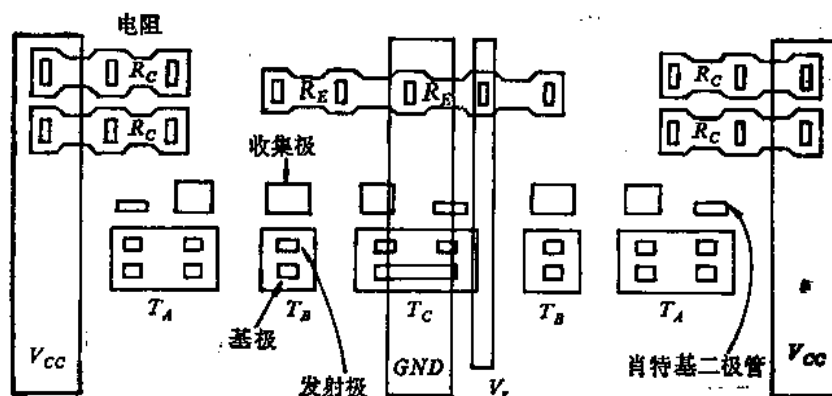


图 2-16 ECL门阵列单元版图

相连和发射极相连。由于宏单元不象一般门阵列那样需要大量的布线通道用于各个门之间的连接，因而芯片的集成度较高。

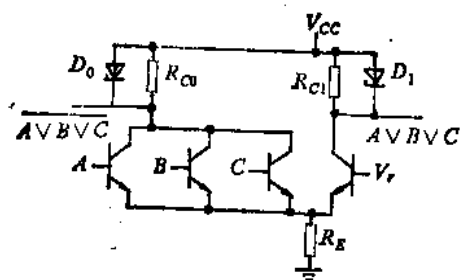


图 2-17 ECL 门的电路图

三、典型的门阵列设计流程

典型的流程图示于图2-21，一开始由用户提出逻辑图及一组测试矢量，先验证逻辑功能是否正确。下一步将该逻辑图转换成相应的门单元或宏单元，并进行布图前的逻辑模拟。在模拟完成后，选择合适的门阵列基片完成初始的布局，必要时要对逻辑图作适当修改，反复这一过程直到找到一个可以接受的方案。在确定方案后进行最后的自动布

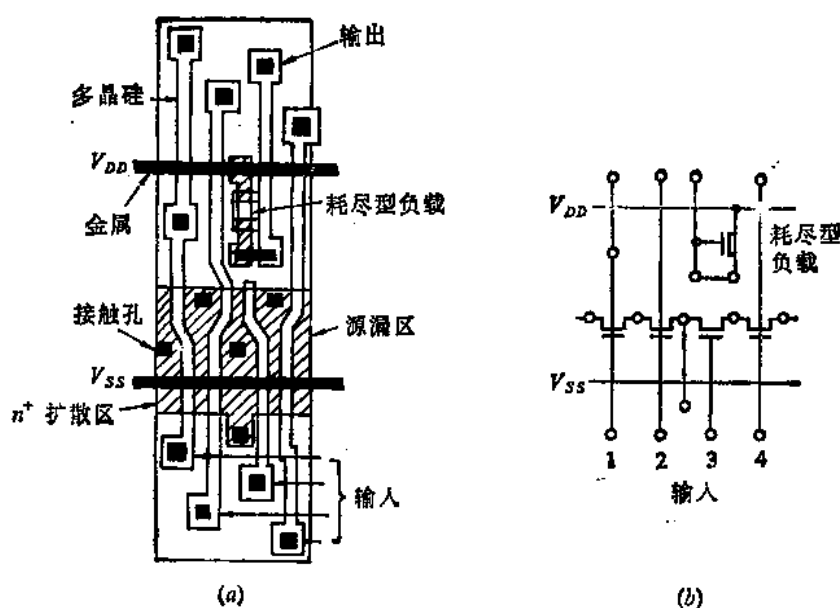


图 2-18 E/D NMOS 门阵列单元

局和布线。最后进行一次考虑所有寄生参量（布图后）的逻辑模拟。

在用户认可后，产生图形发生器或电子束制版设备所需的控制文件。然后送去加工掩膜版并制造芯片。

从以上讨论可以看出，门阵列的优点是明显的。它采用相同的单元，并已完成了连线以外的所有加工工序，需要定制的掩膜版只有两种至四种。CAD 所要完成的工作是根据网络要求进行门的自动布局和自动布线。因此设计周期大大缩短，成本也大大下降。

当工艺改变或单元结构需要变化时，只要对单元本身进行修改，CAD 软件的变化不大。由于上述这些因素，因而原始投资较低，即使要求芯片产量很低时，如几百或几千块芯片，其价格也是在可以接受的范围内。这一优点是门阵列在各个应用领域迅速得到推广的重要原因。

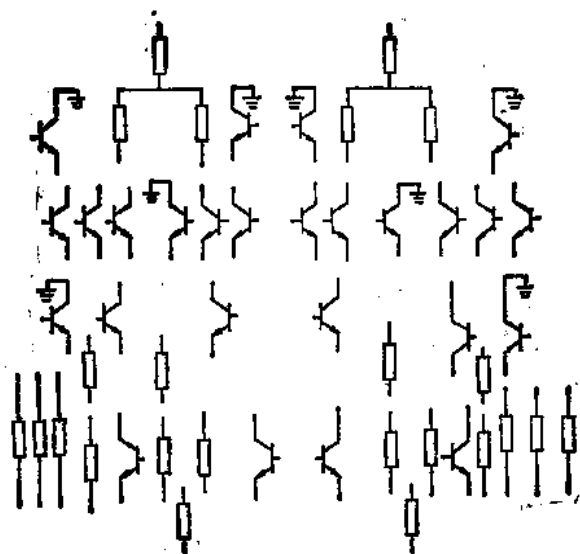


图 2-19 宏单元示意图

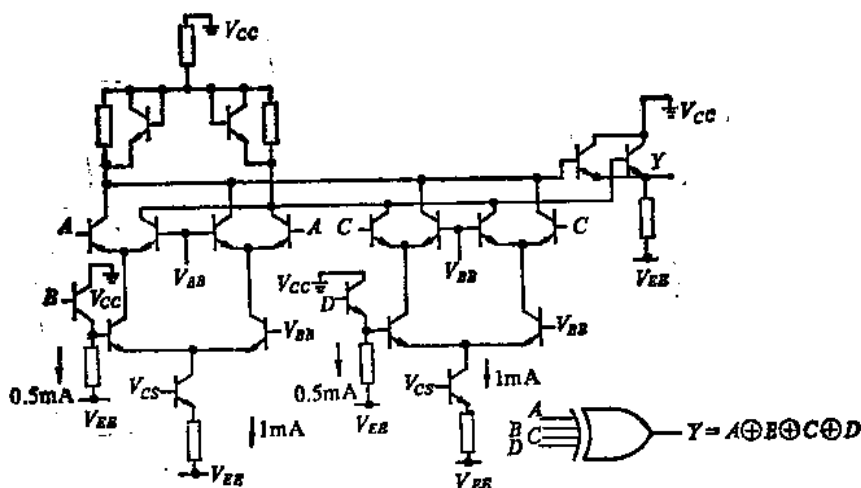


图 2-20 四输入端 ECL 异或门

但门阵列也存在不少缺点。首先是门的利用率较低，且无用晶体管的比例与网络的要求有关。在所提供的连线空间已全部用完，或芯片上所提供的 I/O 压焊块全部被占用后，即使有多余的门实际上也无法再加以利用。第二，单元中晶体管的尺寸，由于要适应各种不同的要求，一般设计得较大，因而速度较低，功耗较大。第三，门延迟的均匀性很难保持。而延迟的不均匀可能引起错误的输出讯号。在全定制设计中可以通过重新设计局部电路或调整晶体管尺寸来减少或消除这种不均匀性。但对于门阵列来讲，这种局部细致的调整是不可能的。第四，在某些情况下，利用自动布局和自动布线程序不能保证 100% 的布通率（特别是在单层金属连线时），而需要进行人工布线，这往往需要花费大量的时间。

为了克服芯片面积利用率较低的缺点，近年来提出了一些改进方案。例如单元采用栅

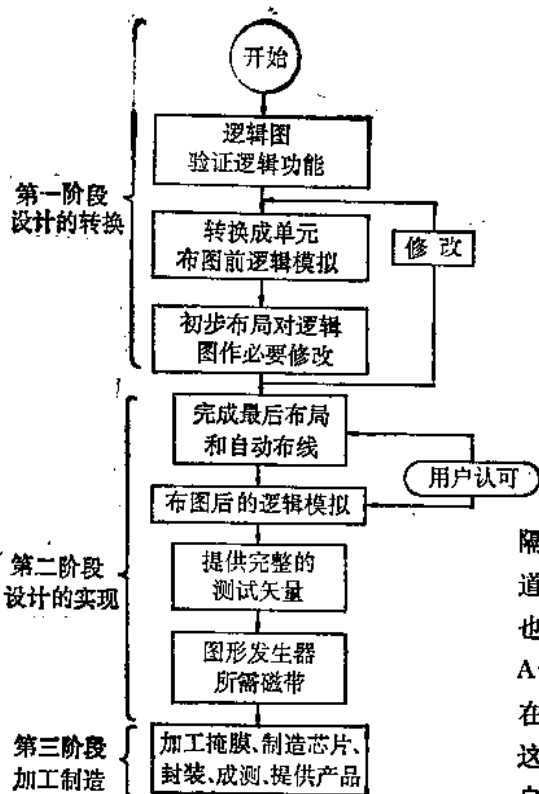


图 2-21 门阵列设计流程

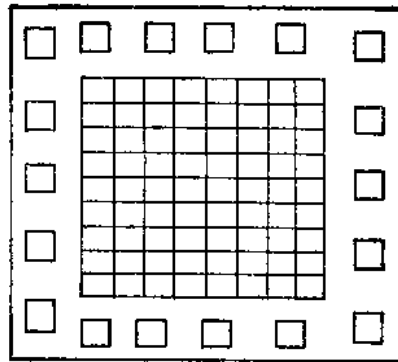


图 2-22 门海示意图

隔离方式，取消布线通道等方案。无布线通道门阵列常称为“门海” (Sea of Gate)，也有的称其为密集型门阵列 (Compacted Array)。门海的典型版图形式如图 2-22。在需要时把某些单元所占区域作为布线区，这样布线区的宽度可变。但这使自动布局和自动布线程序的开发难度加大。

§ 2.4 标准单元法

标准单元法是库单元设计方法中的一种。标准单元法中的各个单元具有同一高度（指版图尺寸），但宽度不等。单元内的每个器件事先都经过精心的设计，并经过设计规则和电学性能的验证。设计好的各个单元存入 CAD 系统的单元库中以便调用，单元的电学特性也存入 CAD 系统的单元模型库中。标准单元的示意图见图 2-23。

设计时将所需的单元从单元库中调出，将其排列成若干行，行间留有布线通道。然后根据网络的连接关系将各单元用连线联接起来，同时把相应的输入/输出单元和压焊块连接起来，即得到所要求的芯片版图。标准单元法的典型版图布置示于图 2-24。



图 2-23 标准单元示意图

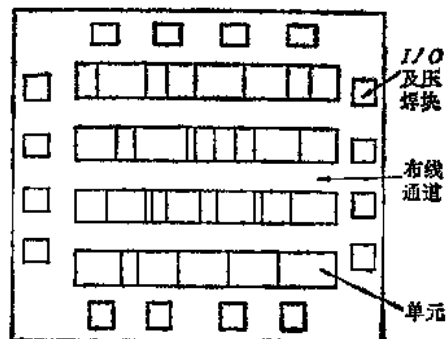


图 2-24 标准单元法的版图布置

芯片主要分三个区域：（1）四周的 I/O 单元和压焊块；（2）单元部分；（3）布线通道。由于标准单元本身的输入/输出信号端都处于单元的上下两端，因此单元之间的连线都处于布线通道内。

设计人员只要输入被设计电路的逻辑图，或输入一种电路描述文件，再加上输入压焊块排列次序，标准单元法自动布图系统将调用所需单元和相应的 I/O 电路及压焊块，进行自动布局和自动布线，得到被设计电路所对应的版图。虽然每个被调用的单元都是事先设计好的，但制造芯片时的各层掩膜都需要根据布图结果进行专门的加工定制。这点与门阵列方法有着根本的不同，即不同的电路需要一整套完全不同的掩膜版，因此无法事先完成部分加工工序。以下具体讨论标准单元和单元库的建立。

一、单元库

单元库中每个单元都各具有三种描述形式。

- （1）单元的逻辑符号（以 L 为特征）；
- （2）单元的拓扑版图（以 O 为特征）；
- （3）单元的掩膜版图（以 A 为特征）。

现以倒相器为例，分别画出其逻辑符号、拓扑版图和掩膜版图，示于图 2-25。单元的逻辑符号用以建立逻辑图。单元的拓扑版图描述单元掩膜版图的外形尺寸、输入输出端口的位置及其宽度。在拓扑版图上除标有单元名外，还有输入输出端口名和控制端口名，其名称与逻辑符号中的名称完全一致。

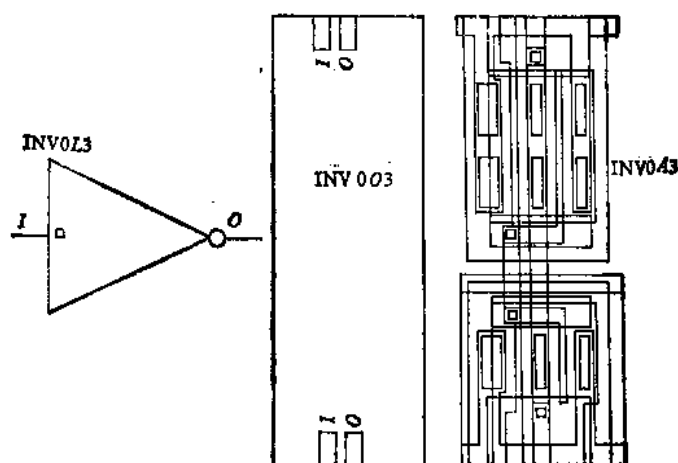


图 2-25 倒相器的三种描述形式

采用标准单元设计集成电路时，开始只调用单元的拓扑版图，因为它的引入会大大压缩数据的处理量，并有助于设计人员的直观检查。设计人员一般不需详细了解单元内部的版图细节，而只要掌握单元的主要特征。在自动布局和自动布线后得到的是被设计芯片的拓扑版图，最后需要经过一次转换，将所有单元、 I/O 单元及压焊块的拓扑形式转换成各自的掩膜形式，这样就得到需要的掩膜版图。

在设计的不同阶段，设计系统将分别调用单元库中的这三种描述形式，示于图 2-26。

二、单元设计

标准单元的版图设计与通常的版图有所不同。它要求输入/输出端、电源线和地线都

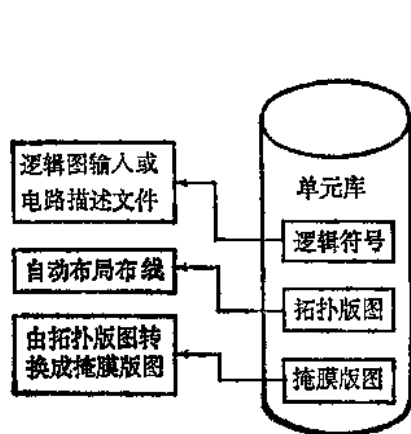


图 2-26 调用不同描述形式的各设计阶段

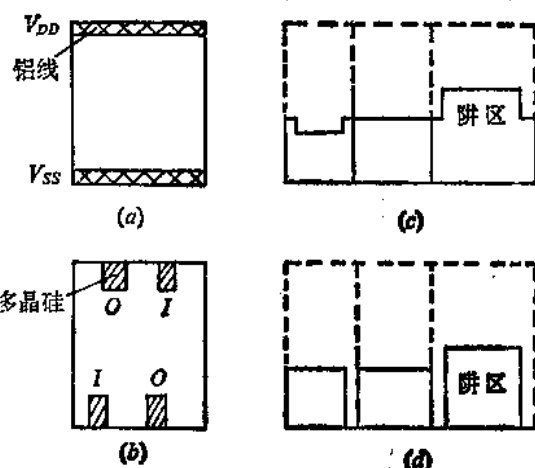


图 2-27 标准单元版图设计的特点

排列在单元的上下两端，且希望每一个输入/输出端在单元的上下两个方向都能引入或引出，这样会有利于通道布线。

现以 CMOS 工艺为例具体说明单元设计的特点：

(1) 单元的最上端布以水平走向的、贯穿整个单元的铝线，作为电源线 V_{DD} 。单元的最下端布以水平走向的、贯穿整个单元的铝线，作为地线 V_{SS} 。这样在单元拼接时，电源线和地线可直接相连，如图 2-27(a) 所示。

(2) 标准单元各输入/输出端均采用多晶硅从上下两个方向引出，如图 2-27(b) 所示。

(3) 由于各单元中 P 型 MOS 管和 N 型 MOS 管所占面积的比例很不相同，因而无法采用阱区等高结构。但考虑到单元排列时的拼接，采取单元边缘处阱区等高方案，示于图 2-27(c)；或阱区与单元边缘间留有一定间距，示于图 2-27(d)。

(4) 为了消除单元中的门锁效应 (Latch-up)，在每个标准单元内部应加有足够多的衬底接 V_{DD} 孔和阱区接 V_{SS} 孔，或加必要的保护环。

标准单元的单元库中，同一功能的单元应有几种不同的型号（具有不同的驱动能力）供设计者选择。以倒相器为例，应有输入级、内部级、缓冲级及输出级之分。与非门、或非门、传输门和触发器等都应设计出不同的型号。这样，在逻辑图转换成标准单元电路图

时就有较大的灵活性。但同一功能的单元具有过多的型号，会增加单元设计的工作量，加大初始投资，故应作折衷考虑。

此外，标准单元库中有一种特殊单元，称为连线单元。其拓扑版图与掩膜版图示于图 2-28。其高度与其它标准单元相同。当需要在两个布线通道之间穿过一根连线时就从库中调用它并插入标准单元行中，然后把连线接到连线单元的上下端口，以完成连接。

三、标准单元法的设计流程

标准单元法的设计流程示于图 2-29。该

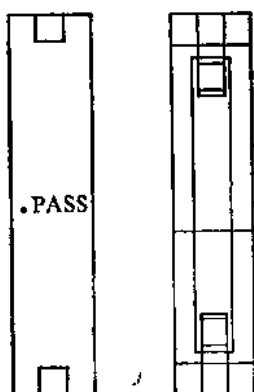


图 2-28 连线单元

设计流程与门阵列法的流程很相似，但有着基本的不同点。(1) 在门阵列法中逻辑图是转换成门阵列所具有的单元或宏单元，而标准单元法则转换成单元库中所具有的单元。(2) 门阵列设计时要先选定某一种门复杂度的基片，因而门阵列的布局和布线是在最大的门数目、最大的压焊块数目、布线通道的间距都确定的前提下进行的。标准单元法则不同，它的单元数、压焊块数取决于具体芯片的要求，而且布线通道的间距是可变的，当布线发生困难时，通道间距可以随时加大，因而布局和布线是在一种不太受约束的条件下进行的。(3) 门阵列设计中所得到的掩膜版只有2~4块，而标准单元设计后得到的是所有的各层掩膜版。

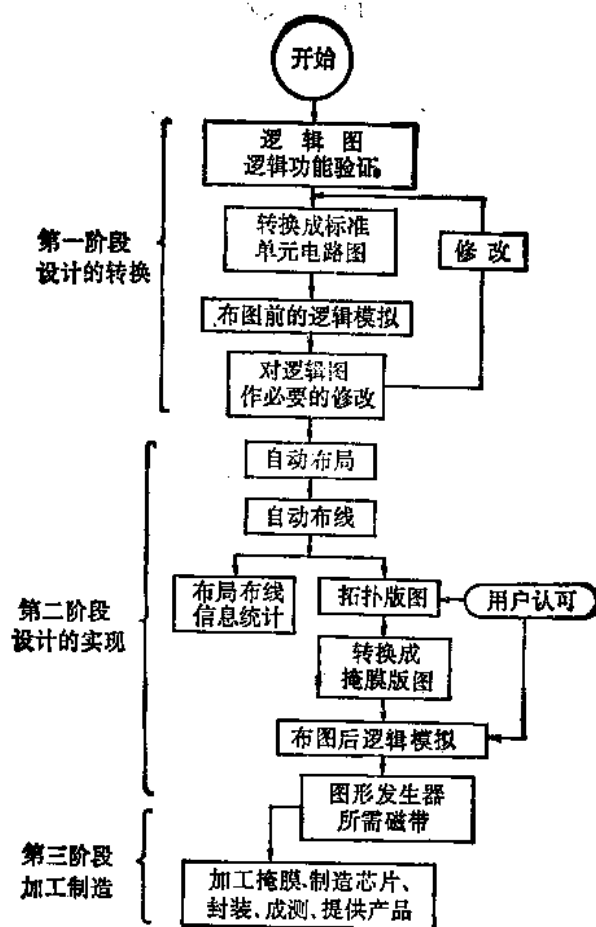


图 2-20 标准单元法设计流程

通过上述讨论，可以看出标准单元法有某些突出的优点：

- (1) 芯片面积的利用率比门阵列法要高。芯片中没有无用的单元（或晶体管）。
- (2) 单元可以根据网络的要求临时加以特殊设计，这样可避免延迟的非均匀性。
- (3) 可以保证100%的布通率。
- (4) 可以与全定制设计法相结合。如图2-30中的宏单元可以是PLA、ROM、RAM或ALU，它们是通过全定制方法设计的，在芯片设计时把它们调到相应的位置，然后与标

准单元各端口相连。这极大地扩展了标准单元法的应用范围。

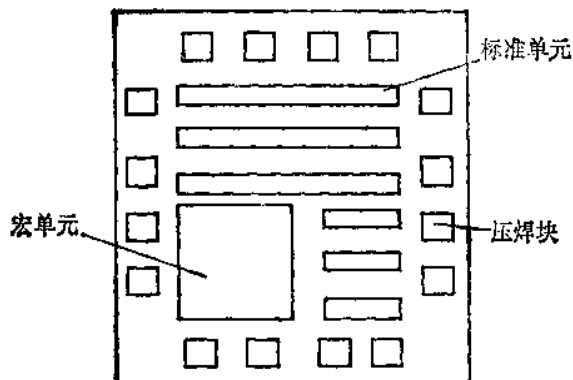


图 2-30 有宏单元的标准单元法

但标准单元也存在缺点和问题，

(1) 原始投资大。建立一个标准单元库需要投入大量的人力和物力。据报导，建立一个小型的单元库约需十二个人月。

(2) 成本较高。由于掩膜版全部需要定制，芯片加工需要专门进行，因此芯片产量要达到某一定额（几万~十万块），其成本才可接受。

(3) 单元的修改和更新要付出相当的代价。如何建立一个在比较长的时间内能适应技术发展的单元库是

一个突出的问题。当前解决的办法是建立一种器件尺寸可变的可变参量单元，及开发一种单元生成器软件来自动地生成各种标准单元。

§ 2.5 通用单元法

通用单元法是库单元设计方法的一种。与标准单元不同之处，它既不要求每个单元（或称积木块）等宽，也不要求等高。每个单元都可以根据最合理的情况单独进行版图设计，因而可获得最佳性能。设计好的单元存入库内。此外，与标准单元有所不同的是，它没有统一的布线通道。通用单元法示意图见图 2-31。

采用这种方法设计时，先把各所需单元调到屏幕上作出合适的版图布置，一般把那些会影响延时的单元尽量靠拢安放，然后通过软件工具预估其性能，如果不合适可调整版图布置。一旦确定下来，再进行详细的布局布线，这时原版图布置会有所变动以适应布线的要求。在布局布线完成以后，再进行一次（版图后）模拟以确定该芯片设计是否符合要求。

由于布线通道不规则，加上各单元的连线端口处于单元的四周，且端口位置也不规则，给自动布局和自动布线带来很大困难。采用这种设计方法的布图系统（又称 Building Block Layout System）现已推出，但仍在不断改进之中。

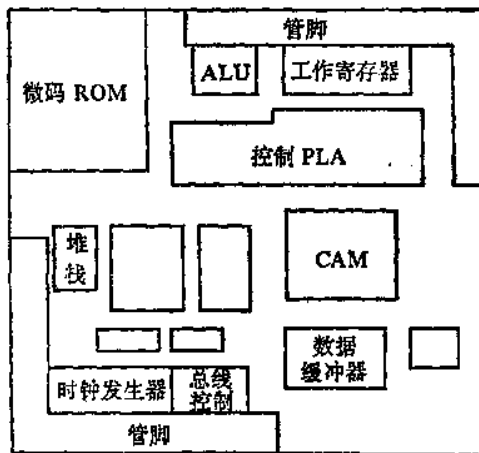


图 2-31 通用单元法示意图

§ 2.6 PLD 方法

根据布尔理论，任何组合逻辑都可以由“与”的“或”来实现。因此我们可以设计出一种

由输入变量构成的“与”矩阵，再将其输出(乘积项)馈入“或”矩阵的结构，如能对该两矩阵进行适当的“编程”，便可得到所需要的各种逻辑功能。PLD方法正是从这一基本概念演变和发展起来的。

70年代初出现的 PROM 器件，采用固定“与”矩阵及可编“或”矩阵方式，可以说是一种早期的 PLD 器件。70年代中推出可编“与”矩阵及可编“或”矩阵的 PLA(Programmable Logic Array)器件。由于两个矩阵均可编，从而大大缩小了“与”矩阵尺寸，并能最大限度地利用共同的积项，使门利用率提高。70年代末，有人根据对大量 PLD 应用实例的分析，提出了可编“与”矩阵及固定“或”矩阵方案，并认为通常情况下“或”门乘积项输入端只要六个就足够，这就是后来被广泛采用的 PAL (Programmable Array Logic) 器件。80年代初发明的 GAL(Generic Array Logic)器件又使 PLD 器件进入了一个新阶段，它不仅提供了高性能(高速和低功耗)，而且可以由用户来确定输出结构。因此它既可编程又是可重新配置的(Reconfigurable Logic)。以下讨论几种典型的 PLD 器件。

一、PLA

如图 2-32 所示，PLA 包含一个 AND 矩阵及一个 OR 矩阵。两个矩阵都可以编程。“与”矩阵的输入为 n 个，输出为 p 个(称乘积项)。“或”矩阵的输入为 p ，输出为 m 个。因而 PLA 器件的面积与 $p(2n+m)$ 有关。通常的 PLA，其 n 为 10~30， p 为 40~150， m 为 6~30，延迟时间约为 30~75ns。

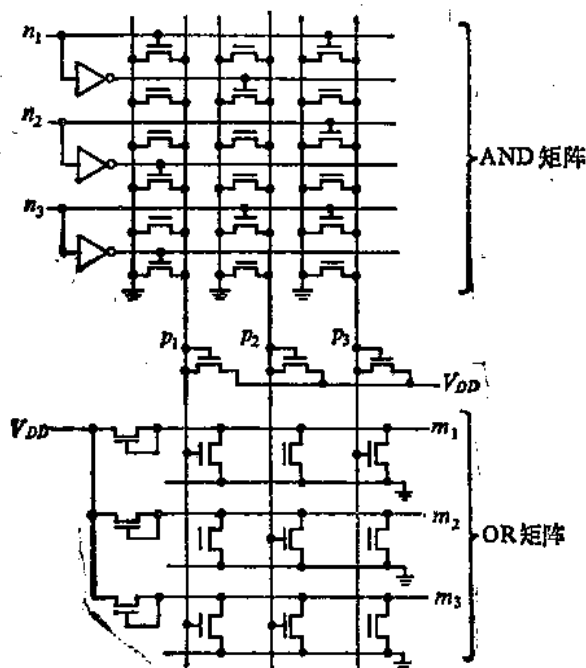


图 2-32 PLA 结构

根据图 2-32 栅的连接(编程)可得到下列功能：

$$p_1 = \overline{n_1} + \overline{n_2} + \overline{n_3} = \overline{n_1 \cdot n_2 \cdot n_3}$$

$$p_2 = n_1 + n_3 = \overline{n_1 \cdot n_3}$$

$$p_3 = n_1 + n_2 + n_3 = \overline{\overline{n_1} \cdot \overline{n_2} \cdot \overline{n_3}}$$

因而

$$\overline{m_1} = \overline{p_1 + p_3} = \overline{p_1} \cdot \overline{p_3} = \overline{n_1 \cdot n_2 \cdot n_3} + \overline{n_1 \cdot n_2 \cdot \overline{n_3}}$$

$$\overline{m_2} = \overline{p_2} = \overline{n_1 \cdot n_3}$$

$$\overline{m_3} = \overline{p_1 + p_2} = \overline{p_1} \cdot \overline{p_2} = \overline{n_1 \cdot n_2 \cdot n_3} + \overline{n_1 \cdot \overline{n_2} \cdot \overline{n_3}}$$

除了组合逻辑外, PLA 还可以实现时序网络, 见图 2-33。这时“或”矩阵的某些输出可连到主从触发器, 而触发器的输出再反馈到“与”矩阵的输入端。

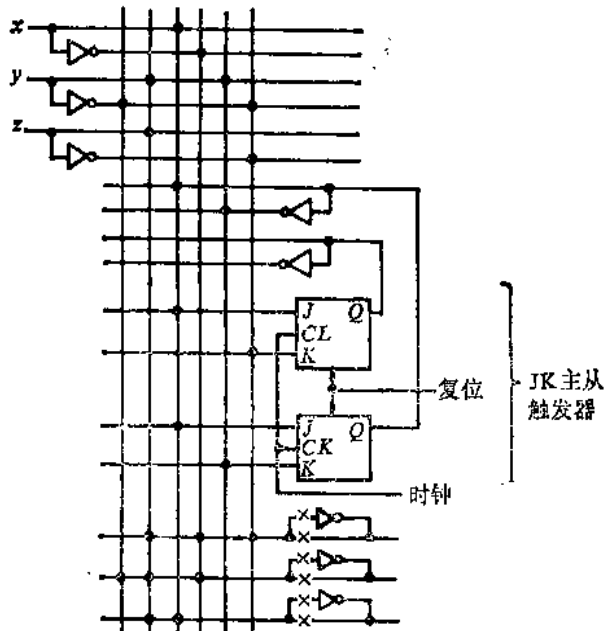


图 2-33 可实现时序网络的 PLA

可以看出, 对掩膜“编程”PLA, 矩阵中每个器件的栅可以根据需要有选择地与垂直线或水平线相连。这种选择性的连接只需专门定制一块掩膜版, 其它掩膜版不必专门设计, 这就大大降低了成本。此外 PLA 的版图很容易自动生成 (由于它的规则结构), 使设计周期大大缩短。设计验证也特别容易, 只要检查栅的连接是否正确就可。

在自动生成 PLA 时, 设计人员只要输入布尔方程 (或状态表), 生成系统就可以通过软件首先进行布尔最小化, 减少乘积项。设计人员也可事先采取某种对输入变量的预处理, 或对输出信号求“反”等措施以减少输入变

量或乘积项。这些措施都是为了减小 PLA 面积, 因为晶体管数目的减小可减少功耗和扇出数, 使 PLA 的速度加快。在最小化以后即可转换成 PLA 版图, 得到“编程”的掩膜版。

PLA 的主要缺点是: (1) 速度慢。由于采用矩阵形式, 因而有较长的连线, 特别是乘积项较多时。(2) 占用面积较大。(3) 器件的利用率很低。

二、折迭式 PLA

为了解决 PLA 面积大、器件利用率低的问题, 近年来发展了多种折迭方案。如列折迭、行折迭, “与-或-与”结构等。行折迭的示意图见图 2-34, 列折迭的示意图见图 2-35。上述的行折迭和列折迭可通过分析 PLA 的“特征矩阵” (Personality Matrix) 来得到。

图 2-36 是近年来开发的折迭方案。它将单侧输入输出改为双侧, 并采用“与-或-与”矩阵形式。图中是有四个 PLA 的折迭式结构, 虚线表示四个 PLA 的边界, 这些边界是切割连线的结果。PLA 周围有几根金属总线环, 每根总线环可切成几段作为很多单独的连接线。这样, 任何输入或锁存器的输出就可以连到任何译码器的输入端, 而任何锁存器的输出可以作为 PLA 的输出。图 2-36 中还有两组横向总线, 可以用来连接“与”矩阵的乘积项, 具体连接方式见图 2-37。

图中实线为被保留的金属线, 虚线表示被去除的金属线。图 2-37 中示出上两根横向总线和 AND 矩阵的乘积项相连。第三根横向总线与总线环相连, 这种横向总线的存在增

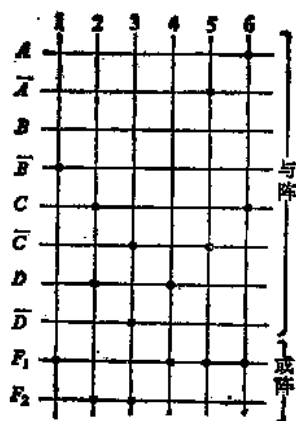


图 2-34 行折迭示意图

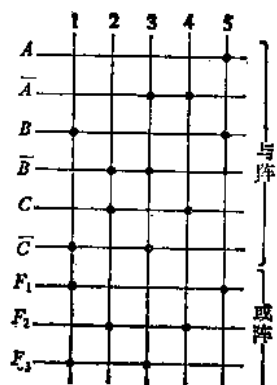
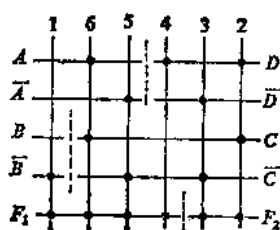


图 2-35 列折迭示意图

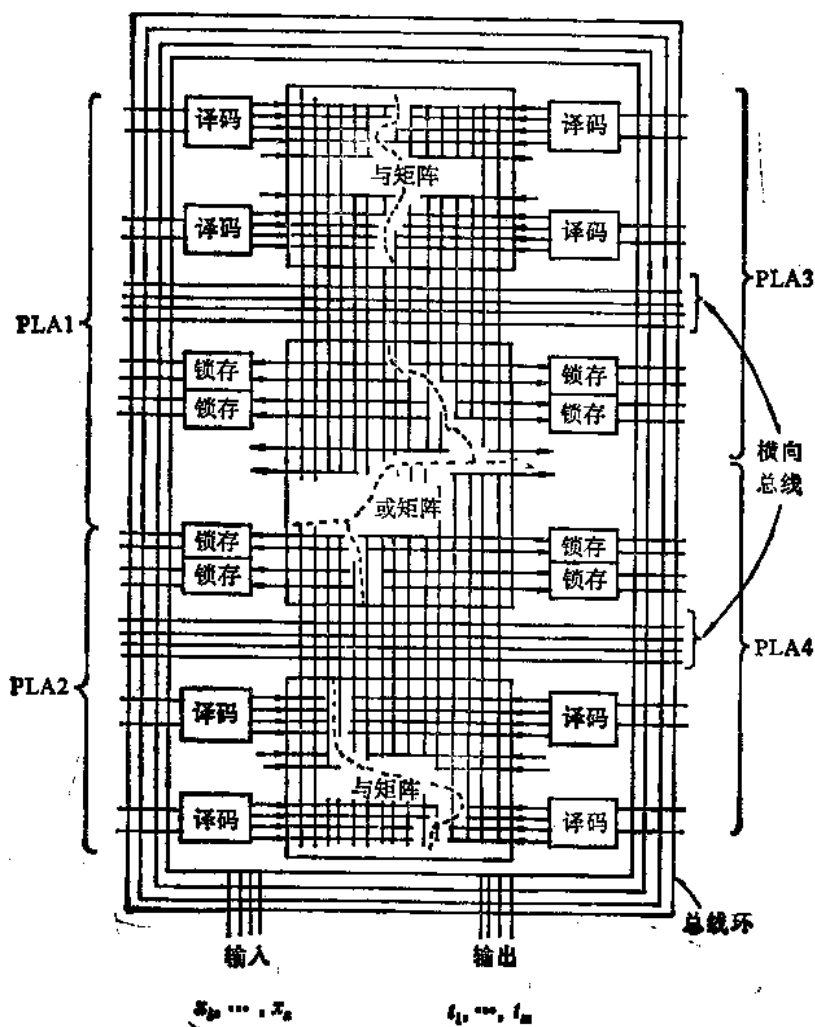
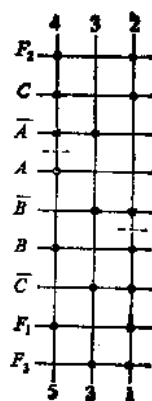


图 2-36 折迭式 PLA 结构

加了连接的灵活性。但总线环与横向总线是在不同的金属层上，即这里采用了双层布线技术，与普通的 PLA 不同，它需要增加一层定制的掩膜。

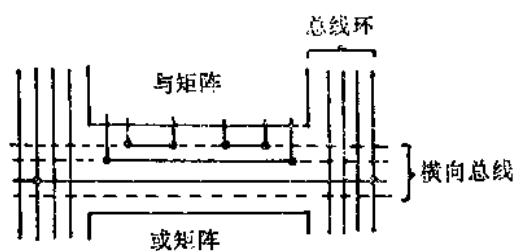


图 2-37 横向总线的用途

目前已有自动进行折迭的 PLA 生成系统出现，它大大提高了 PLA 设计自动化的水平。

三、GAL

它与 PAL 一样采用可编“与”矩阵及固定“或”矩阵结构。其逻辑图示于图 2-38。但与 PAL 相比有两点基本的不同处。

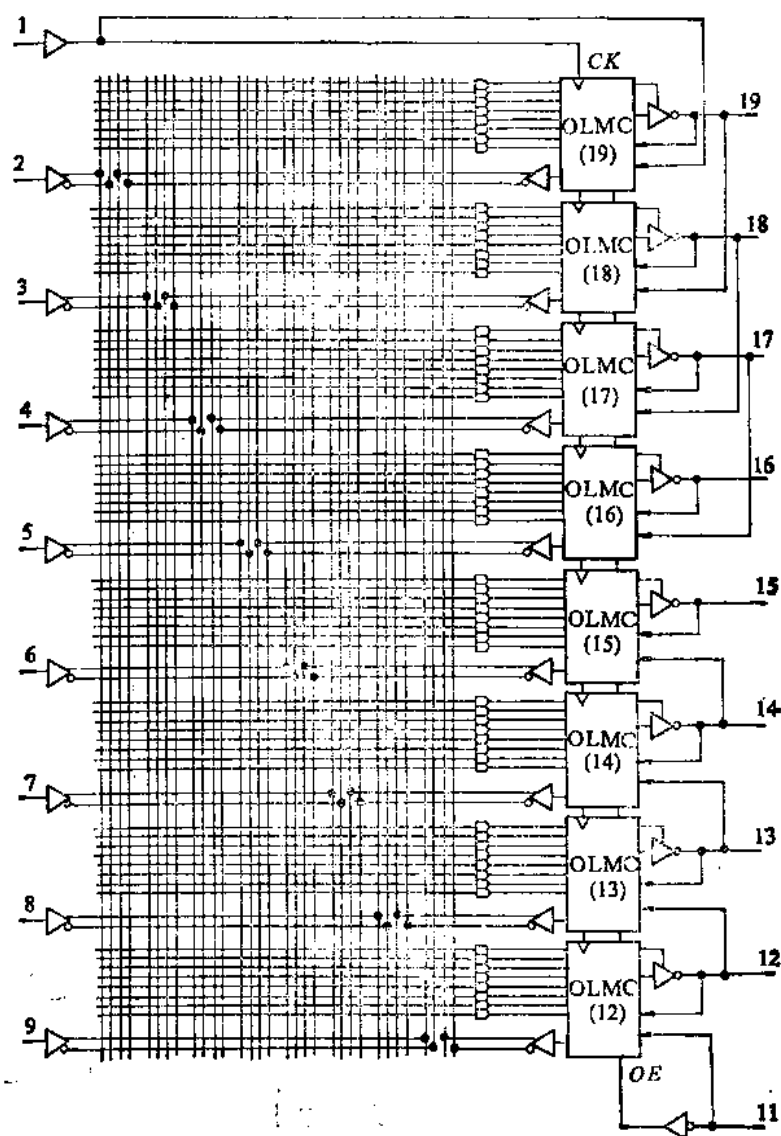


图 2-38 GAL 逻辑图

（1）它采用 CMOS 的浮栅新工艺制造各晶体管，因而可以采用电擦电写来编程，电擦写的速度极快，每次编程时间不到 1 秒，而且可以重复编程（可重复编程 100 次以上）。

同样由于采用 CMOS 工艺而使器件速度提高（延时为 25~35ns），功耗下降（工作时约 45mA，保持时约 35mA）。

（2）它采用了一种可编程输出逻辑宏单元 OLMC（Output Logic MacroCell），并可以通过四个变量变换输出结构。OLMC 的逻辑图示于图 2-39。从图中可看出“与”矩阵的输出项馈入每个 OLMC。OLMC 的输出可以分别地设置输出极性为正逻辑或负逻辑（高有效或低有效），还可以设置成组合逻辑输出（非同步）或寄存器输出（同步）。有一公共的输出允许（使能）连接到所有的输出端，此外乘积项也可用来作为各自的输出允许控制。输出端还是双向的，当三态输出缓冲器被禁止时，该端可作为输入，将变量送入“与”矩阵。

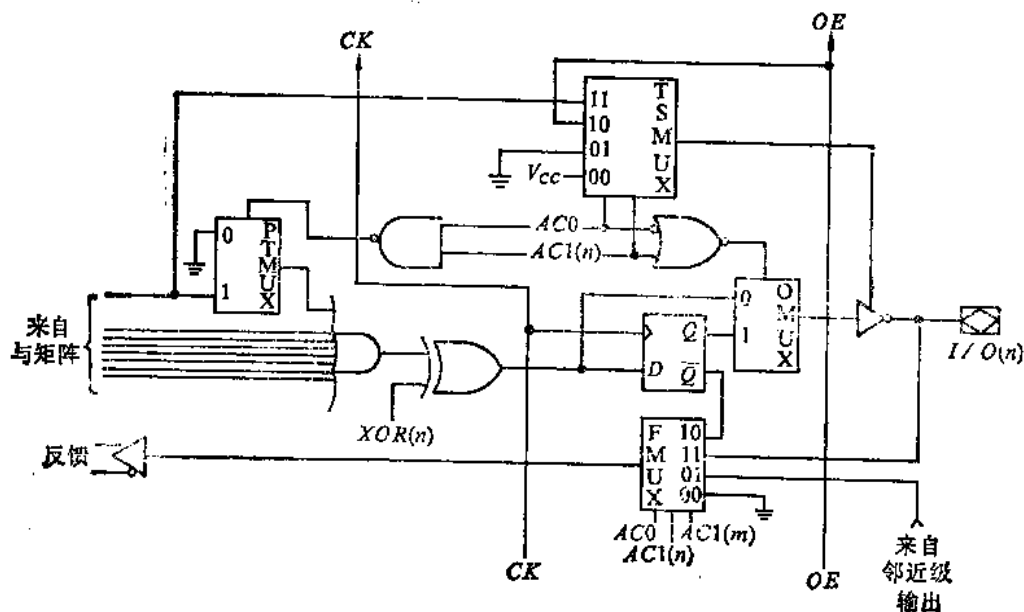


图 2-39 OLMC 逻辑图

不同的配置可以通过改变 82 位结构控制字中的四个变量 SYN、AC0、AC1(n) 和 XOR(n) 来获得。其中 SYN 决定输出是寄存器输出还是单纯的组合逻辑输出。AC0 和八个 AC1(n) 控制三态输出缓冲常开或禁止，以及决定是否由乘积项控制三态缓冲器。它们还决定通过 FMUX 的反馈项来自何处（外部、内部还是邻近输出端），并选择输出状态。八个 XOR(n) 将决定输出极性为高有效还是低有效。

如该四个变量中的 SYN 为 0，AC0 为 1，AC1(n) 为 0，则图 2-38 中管腿 1 为 CK，管腿 11 为 $\overline{OE}$ ，输出状态为寄存器方式。这时在“或”矩阵与三态缓冲器之间插入了一个受时钟控制的寄存器，而且其状态可反馈到“与”矩阵作为内部输入变量，即所谓的“埋态”（Buried State）。OLMC 改变成图 2-40 所示的结构。

所谓改变这四个变量的值，实际上是通过寻址方式去改变相应单元的晶体管浮栅的状态。

GAL 的编程是先通过软件编译将布尔表达式（或编程语言的逻辑描述）编译成可写入 GAL 的编程文件，即统一的标准格式 JEDEC 文件。再送入硬件编程器完成 GAL 的物理编程。

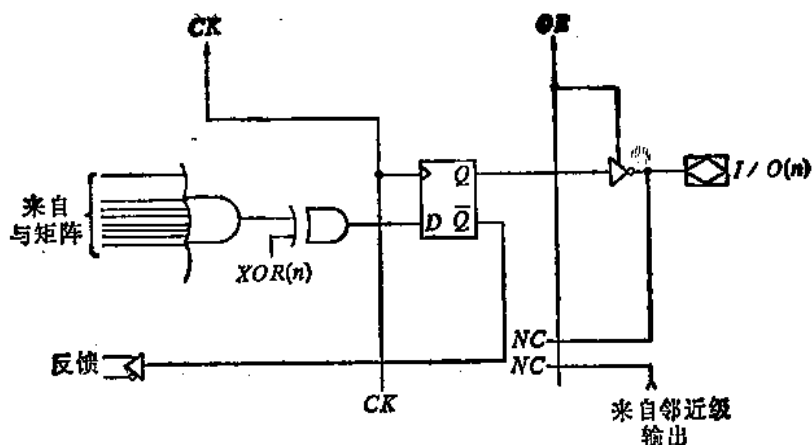


图 2-40 寄存器输出状态

通过以上分析可以看出, GAL 可实现任意的组合、时序逻辑功能,而又大大地简化了设计过程,降低了系统的体积和成本,提高了系统可靠性,特别适用于正在开发中的系统。但目前 PAL 和 GAL 的规模只限于 2,000 门以下,而且门的平均利用率只有 30%~50%。因此各 PLD 厂家都在致力于结构的改进和创新,如将输出逻辑宏单元改进为多功能 I/O 宏单元,增加“埋态”宏单元等以提高门利用率。可以预期,PLD 的性能将随着芯片工艺技术和编程工具的发展以及新结构的提出而有新的突破。

§ 2.7 不同设计方法的比较

现从几个方面对全定制、符号法、通用单元法、标准单元法、门阵列法和 PLA 生成法等不同方法进行综合比较,并列于表 2-1。

表 2-1 不同设计方法的综合比较

设计方法	设计效率	功 能 面 积	电路速度	设计出错率	可 测 性	重新设计的可能性
全 定 制	×	☆	☆	☆	△	×
符 号 法	△	○	○	—	△	—
通用单元	×	○	○	○	—	—
标准单元	—	—	○	△	—	○
门 阵 列	○	△	—	△	—	○
PLA(ROM)	☆	×	×	×	☆	☆

☆ 最高(最大) 高(大) 中等 △ 低(小) 最低(最小) ×

§ 2.8 混合模式设计方法

混合模式设计方法发挥各个设计方法的各自优点,根据不同模块的特点采用不同的设

计方法,然后综合地应用到一个芯片的设计中。

现以一个 16 位微处理器芯片为例,说明采用混合模式方法比单一的标准单元法所带来的好处。

根据对微处理器芯片各部分的分析,对位片和阵列结构的网络如 ALU、移位器、寄存器采用人工全定制的模块,对于某些随机网络采用 PLA 结构(如能得到较高的门利用率的话),而对于定序器、控制器这样的随机结构逻辑网络则采用标准单元法。

具体各功能块采用何种设计方法见表 2-2。

表 2-2

功能类别	功能块名称	网 络 结 构	
		IMSA 芯 片	标准单元法芯片
数 据	SPM	位片结构	RAM
	Data 寄 存	位片结构	标准单元
	ALU	位片结构	标准单元
	Data I/O	位片结构	标准单元
	PSW CC	位片结构	标准单元
控 制	IRDEC	PLA	标准单元
	CS	ROM	ROM
	#DEC	PLA+标准单元	标准单元
存 储	ATT	RAM	RAM
	ATL	位片结构+标准单元	标准单元
	地 址 I/O	位片结构	标准单元
时 钟	CLK	标准单元	标准单元
	BC	标准单元	标准单元
其 它	MISC	PLA	标准单元
	Counter	标准单元	标准单元

该芯片采用 $2\mu\text{m}$ 硅栅、双层铝线 CMOS 工艺。芯片可工作在 5MHz 时钟频率。采用混合模式,此芯片的电路和版图设计共花费 20 人月的工作量。其中 10 人月用于(人工全定制)设计位片式模块,4 人月用于设计 PLA 和存储器。而采用单一的标准单元法设计时,共用 4 人月,其中 3 人月用于设计 ROM 和 RAM。设计的结果示于表 2-3。可以看出,采用 IMSA 方案后虽然晶体管数目比 SCA 方案有所增加,但 IMSA 方案的芯片总面积(包括电源线和外围连线)只有 40.71mm^2 ,与 SCA 方案的 53.96mm^2 相比,节省芯片面积约 25%,节省面积的程度是相当可观的。如果能建立起位片模块的单元库,则设计工作量也必将大大减少。

表 2-3

功能类别	器 件 数 目		面 积 (μm^2)	
	IMSA 芯 片	标准单元法芯片	IMSA 芯 片	标准单元法芯片
数 据	6,410	8,420	4.75	8.29
控 制	44,760	31,660	10.10	11.41
存 储	5,960	7,130	4.45	6.43
时 钟	2,740	2,610	4.53	4.52
其 它	6,910	5,800	16.88	23.31
总 计	66,180	55,620	40.71	53.96

§ 2.9 层次式设计概念

随着集成电路集成度越来越高,在设计方法上必需实现某种观念上的转变,即从平铺式(Flat Design)设计转变为层次式(Hierarchy)设计。层次性就是把整个芯片划分成若干能独立处理的模块,而这些模块又被进一步划分成若干个子模块,重复这一过程直到子模块的复杂程度降低到可以很容易加以处理时为止。采用了层次式设计方法,就有助于集体性地进行设计,每个设计人员可负责设计芯片中的一个部分。

层次式的概念可以应用到所有种类的 VLSI 设计方法中,包括门阵列和标准单元方法中。但由于最复杂的还是全定制芯片,因此在全定制设计中对层次性的要求最迫切、也最强烈。

一、层次的划分

典型的可分成下列五层:

× 芯片

× 宏功能块

× 功能块

× 基本电路

× 晶体管

最低层次是晶体管层,在全定制方法中设计人员可在该层次优选器件尺寸。第二层为基本电路层,如与非、或非、异或、锁存器等。第三层为功能块层。它是将基本电路或将一些晶体管连接起来形成比基本电路更复杂的逻辑或有存储功能的电路,例如寄存器、奇偶发生器、移位寄存器、ALU、ROM 及 PLA 等。在复杂的定制或半定制芯片设计时采用功能块是一种常用的作法,但在门阵列和标准单元法中,功能块多半只限于 ROM、RAM 和 PLA。第四层为宏功能块层。它是把一些功能块连接起来形成芯片的“子”功能,例如控制部件、运算部件等。层次的顶部为芯片,芯片由若干宏功能块和功能块组成,一个芯片可以只采用功能块而没有宏功能块,这说明层次的数目完全由设计的有效性来决定。

整个设计过程是自顶而下地进行,即从芯片的整个逻辑功能出发,逐渐分解到最低层次。

二、电路(功能)库

层次式设计依赖于库的存在以及库的完整性。我们可以利用一个已有的库,在此基础上增加一些新的基本电路或功能块,或者建立一个新的库。

库内应有各基本电路及功能块的基本数据为:功能指标、电路类型、电学指标、逻辑模型、测试模型、版图数据。该库应有一个公共的接口,允许在一个芯片或多个芯片的设计过程中无约束地调用库内的各种数据。

设计的层次性要求逻辑模型也是有层次的。例如一个 ALU,它有一个高层次的模型供芯片级的模拟之用,同时具有一个低层次的模型供 ALU 本身模拟时用。低层次的模型把 ALU 表示为基本电路如与非、或非门等。

版图的图形数据是制造芯片的各层掩膜版所需要的。为了允许层次式的检查,版图数据应具有一种高层次的“拓扑”数据,在版图全局布置时我们将使用这种“拓扑”版图。

三、版图全局布置 (Floorplan)

有了芯片逻辑功能的描述,有了电路(功能块)库,我们就可以具体考虑芯片的版图全局布置并估算芯片的尺寸。

芯片的逻辑功能会提出功能块的类型和数量,电路/功能块库将提供每个功能块的尺寸、性能和功耗。这样我们就可以了解功能块对面积的要求,可以估算出电源线、讯号线以及 I/O 所占有的空间,从而得到芯片面积的初步估计。

芯片的版图全局布置是要确定每个功能块和 I/O 在芯片上的最佳布局,而又能保证功能块之间的连线都能连通。版图布置的目标之一是使功能块与功能块之间的连线最短,从而使芯片中的连线所占面积最小。连线最短可以降低电路的负载,因而得到较好的性能和较低的功耗。在版图全局布置时,我们往往把有高连接度的功能块分在同一组,然后再考虑该组内部的相对布置。有了芯片的版图布置,就可以分析限制芯片性能的临界通路(Critical Path),得到该通路上的总延迟以及分配给该通路上各功能块的分延迟。临界通路的分析会反过来影响和改变版图布置。对于临界通路上的功能块在布局时应优先加以考虑。

在很多情况下,版图布置不是一次就能确定下来,往往需要多次反复,或者经过几种方案的比较才加以确定。

在版图全局布置以前,芯片的划分只基于逻辑功能描述,而没有实际的物理和电学因素。所以版图全局布置的过程实际上是对原始的划分(即基于逻辑功能描述的划分)的一种重新评价。考虑到一些实际的物理和电学因素后,可对原始划分作出必要的修改,如把某一功能从一个功能块移到另一功能块,或把一个功能块分离到芯片的不同区域,或把相邻的功能块合并等等。任何这种功能块定义的改变应及时反映到逻辑模型中,以保证逻辑模型与物理模型的一致性。

四、逻辑设计与软件化模型

对于一个芯片来说,自顶向下的逻辑设计过程示于图 2-41。

首先将被设计的芯片分割成各模块和子模块,然后用软件化模型和布尔逻辑模型分别描述它们,再编译成芯片模型后进行逻辑模拟。

现以 1K×18 的通道缓冲器为例,经模块划分后如图 2-42 所示。这里包括七个主要的功能块:(1)10 位加 1 器;(2)2-1 多路开关器(10 位);(3)4-1 多路开关器(18 位);

(4)读写控制器；(5)时钟控制器；(6)四个 256×18 RAM；(7)10 位寄存器。

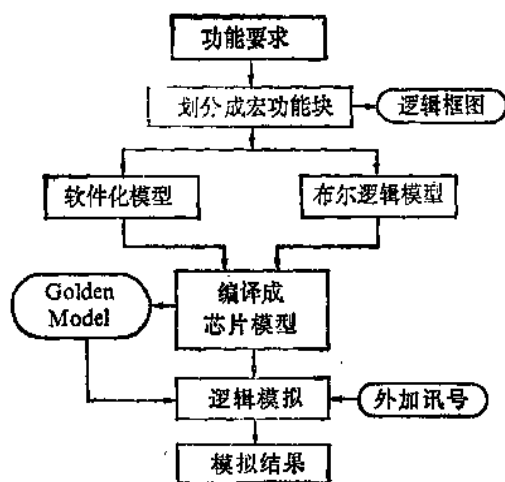


图 2-41 自顶向下的逻辑设计过程

旧有的设计方法是在模块划分后直接用布尔型逻辑单元（如AND、OR、INV等）来表示各模块的逻辑，通常认为这是最快的设计过程。但这样做会限制设计的灵活性，而且一个设计人员必须等待整个系统都完成了逻辑设计后才能评价他自己设计的那部分。

这里介绍一种新的设计方法，就是采用软件程序来定义每个功能块。例如对于加1器来讲，软件程序可以有如下形式：

$$\text{OUTPUT} = \text{INPUT} + 1$$

这样一句语句可以等效于 30~40 个布尔逻辑单元（即一个 10 位加 1 器所需要的逻辑单元数）。

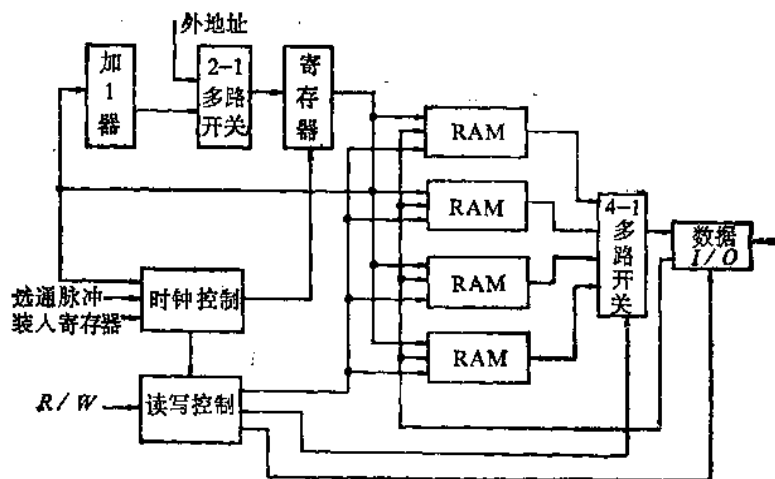


图 2-42 $1K \times 18$ 通道缓冲器

如果所有的功能块都能用这种软件化模型来描述，那被设计的芯片首先可以用软件化模型来验证。在软件化模型中还可以加入延迟时间，这样可在较早的阶段估计出芯片的时钟周期。

通过软件化模型，每个功能块的功能指标就可得到验证。而且由于软件化模型比布尔逻辑模型简单，修改容易，这就大大减少了设计人员的工作量。此外最大的优点是，每个设计人员不必等待别人的进度而可继续完成他所承担的设计任务。

此外，更重要的作用是：可以把已建立起来的软件化模型作为一种“Golden Model”（极好的模型），利用它来验证芯片的其余部分。当然建立的 Golden Model 必须经过完整地模拟以确保它完全按功能要求进行工作。

有了软件化模型，每个设计人员就可以继续他所从事的设计任务。例如通道缓冲器中

的加 1 器就可进一步改为布尔逻辑模型,并以此为基础作进一步模拟。由于所有其它的功能块都采用了软件化模型,并已证明是正确的,所以当进一步模拟时,如出现任何错误,显然是加 1 器新的布尔逻辑模型所引起的。这样设计人员就可以集中力量去寻找加 1 器本身所存在的设计问题,然后可迅速找出问题并加以纠正。但在常规的方法中,首先要把错误从整个设计中加以“孤立”出来,这往往需要花费相当多的时间。

“Golden Model”还起着一种验证模型的作用。例如我们可以同时运行两种芯片模型,一个是与芯片性能指标完全一致的“Golden”模型,另一个是被检查的新的结构模型。在运行这两种芯片模型后,比较两者的输出,就可发现新的结构模型是否存在错误。

采用这种软件化模型进行设计的另一优点是很容易实现设计的修改。例如要求地址既可加 1 又可加 2 时,就修改加 1 器的软件化模型,使其成为,

```

IF COUNT-BY-2 IS ON
THEN
    OUTPUT=INPUT + 2
ELSE
    OUTPUT=INPUT + 1
    
```

对于软件来讲,这种修改是相当简单的。

五、宏功能块设计

其设计流程图示于图 2-43。设计从宏功能块的逻辑描述形式开始,并依靠现有的基本电路/功能块库。全局性限制来自芯片版图全局布置后对宏功能块所提出的某些要求。它内部所含的各个功能块可以通过模块发生器(或模块编译器)自动产生,也可通过人工全定制方法来设计,最后得到宏功能块的图形数据。

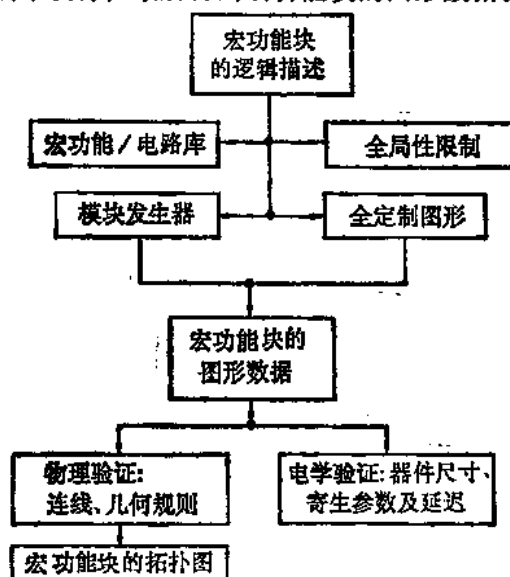


图 2-43 宏功能块设计流程

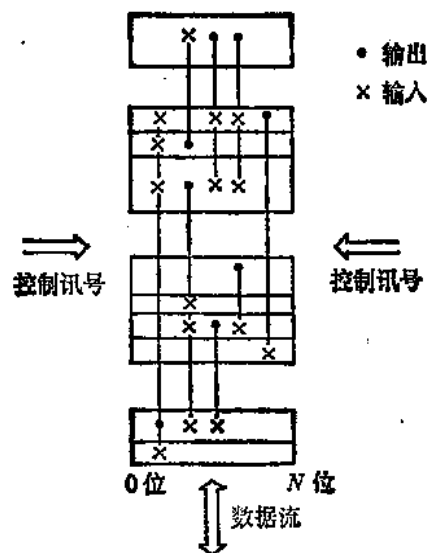


图 2-44 “位栈”版图形式

如果芯片的功能可以分成控制流和数据流两部分时,可考虑采用图 2-44 那样的“位栈”(Bit Stack)版图形式。在位栈中,各功能块具有“位片”式的特征,每个功能块都设计成 $1 \times N$ 。将各功能块垂直叠放成栈,且每位都一一对准。图上的每根垂直连线表示一根

第 N 位总线。控制流从两边引入, 采用水平连线。

控制部分是随机逻辑, 可以采用 PLA 形式或随机逻辑宏单元。它们可以通过自动设计工具来实现。数据流部分设计成“位片”式, 并根据总线的位数 (8 位、16 位或 32 位) 加以扩展。为了得到紧凑的版图, 一般采用人工全定制方法。设计完成后可存入库内, 以便下次调用。

所产生的宏功能块需经物理和电学两方面的验证。物理验证包括连线正确性检查和几何规则检查。电学验证包括器件尺寸检查, 寄生参数计算和延迟计算。如检查无误则产生该宏功能块拓扑版图。以上讨论也适合于功能块设计。

六、芯片设计

这一步是根据芯片版图全局布置放置各宏功能块的拓扑图, 并根据芯片的逻辑描述将各宏功能块连接起来。其流程图示于图 2-45。

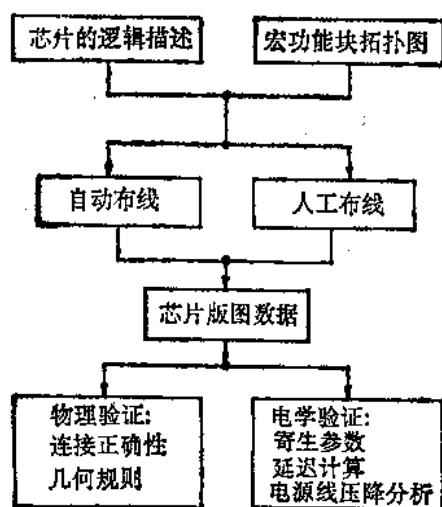


图 2-45 芯片设计流程

对芯片设计来讲, 这时的最低层次为宏功能块, 而且被处理的是拓扑图, 这样可大大压缩待处理的数据量。

连线可以是自动的, 也可以用人工布线。连线时最难处理的是电源线和时钟线。这时绝对不希望有连线再“穿过”各宏功能块, 否则原设计好的宏功能块将推翻重来。

芯片设计完成后, 同样需要进行物理和电学验证。但这时不需要对各宏功能块的内部进行物理检查, 而只要检查各宏功能块拓扑图之间的间距, 拓扑图与连线之间的间距, 各连线本身的宽度及它们之间的间距。

同时检查各宏功能块之间连线正确性。电学验证方面也只要计算全程连线 (即不包括各宏功能块内部连线) 的电阻和电容值, 计算其延迟。此外需要分析电源线及地线上的电压降是否合适。由此看出, 在验证上也贯穿了层次式的概念, 使验证工作量大大减少而加快了设计过程。

第三章 硅 编 译 器

§3.1 概 述

硅编译器是一种理想的自动化工具,利用它可将集成电路的行为级描述直接转换成版图,从而避免了大量繁复而且易出错误的版图设计,并省去了版图验证工作。

VLSI 的设计和制造过程涉及到许多完全不同的学科,电路的结构形式与硅片上几何图形的形式存在着很大的差别,以致在设计一个集成电路时常常有以下的分工:系统设计人员确定系统要求,同时给出原理图或逻辑图。接着集成电路设计师再根据逻辑图或电路图设计版图,得到所需的掩膜图。对于系统设计人员来说,版图设计是一件陌生而又困难的任务。目前系统设计人员人数众多,约有几十万人,但集成电路设计人员只有几万名,集成电路设计人员的短缺已成为阻碍集成电路发展的一个重要因素,因此必须探讨新的设计方法。

硅编译器作为基础的新设计系统为 VLSI 设计开辟了一个新的途径。

它从系统的行为级描述出发,直接得到掩膜信息。同时它还提供逻辑、时序和电路模拟。因此这种系统不仅仅提供辅助设计的能力,而且真正为系统的设计提供了一种设计自动化的手段。

硅编译器的思想于 1978 年由 Martin Rem 首先提出来。1979 年由 David Lawrence Johannsen 首次设计了数据通道硅编译器。80 年代初先后出现了“Mac Pitts”、“CAPRI”、“MODEL”、“FIRST”等编译器,并在市场上推出了商业性的硅编译器系统。

1985 年 3 月利用硅编译器对 W65CS02 芯片进行了设计。该芯片是与八位 CMOS 6502 微处理器完全相容的一个芯片,它有 64 条指令,178 条操作码和 15 种寻址方式。原设计采用人工方式,由一个芯片设计师和两个绘图人员共花费了九个月时间,芯片所占面积为 $137 \times 168 \text{mil}^2$ ($3.48 \times 4.26 \text{mm}^2$)。现用硅编译器进行设计只用了不到一个月的时间,但所占面积较人工设计的大。据报导,Metalogic 公司用标称 $3\mu\text{m}$ NMOS 工艺,所需芯片面积为 $296 \times 263 \text{mil}^2$ ($7.52 \times 6.68 \text{mm}^2$)。Seattle 公司采用 $3\mu\text{m}$ CMOS 工艺,芯片占用面积为 $262 \times 222 \text{mil}^2$ ($6.65 \times 5.64 \text{mm}^2$)。

尽管到目前为止,硅编译器在应用上还有一定的限制,但由于它对系统设计人员具有强烈的吸引力,因此仍有不少人从事这方面的研究和开发。

§3.2 硅编译器与语言编译的比较

硅编译器的“编译”一词来源于软件术语。在语言编译时,“编译”就是用编译程序将源语言产生目标程序的过程,也即将高级语言转换成低级的机器执行代码的过程。这样程序员就可以不再采用低级的机器语言来编写程序而从很多具体繁琐的执行细节中解脱出来。

与此相对应的,在 VLSI 设计中把版图描述认为是机器代码,而把行为级描述认为是高

级语言。因此把行为级描述直接转换成版图描述的过程也认为是一种“编译”过程。硅编译一词就来源于此。这种硬件编译器就称为硅编译器。同样 VLSI 设计人员可以从版图设计的细节中解脱出来，而只需要按照一定的格式进行行为级的描述。

硅编译与语言编译两者有很多相似之处，但也存在着一些原则的差别。现将两者的特点与差别比较如下：

一、语言编译的特点

1. 程序员用高级语言编写程序，而不必知道机器代码的细节、代码放置的物理地址，甚至不需要知道在目标机器中采用的是什么指令系统。

2. 允许程序员对高级语言程序作出修改，并很快地进行试运行。当发现某一错误时，可立即修改而形成新版本。但如使用机器代码编写程序时，如有更动，则可能完全重写。

3. 不用高级语言几乎无法编写和调试大的软件系统。

4. 经编译后的机器代码可保证正确无误。不必运行检查程序来检验该机器代码是否正确执行了源程序。

二、硅编译的特点

1. 芯片设计者按输入描述语言（行为级）设计芯片，而不必知道有关版图、连线等的具体细节、某一晶体管在版图中的具体位置，甚至不必知道所采用的设计规则是什么。

2. 允许芯片设计者对设计作出修改，而且可以比较不同的设计方案。而在手工设计版图时，要作出修改是很困难的，要比较不同的方案几乎是不可能的。

3. 不用自动（或半自动）设计工具几乎不可能进行 VLSI 的设计。

4. 得到的版图可保证正确无误，不必再运行设计规则检查程序、电学规则检查程序等来检验设计是否符合各种规则，而这一步往往花费很长时间。

三、语言编译与硅编译的差别

1. 语言编译产生一维的结果，代码存放时按照顺序的地址进行。而硅编译则产生二维结果，在芯片上的版图是二维的（即 x 和 y 方向），对多层布线的芯片来讲甚至是三维的。

2. 语言编译的程序中任何地址之间的转移按同一速度进行，执行 GOTO 指令时没有任何障碍。但对硅编译来讲，芯片设计中连线及信息传递非常重要，且芯片中两点之间的通讯速度与具体位置有关。也不允许连线在芯片中任意地穿越，而必须设法绕过其它模块或其它连线。

§ 3.3 硅编译器的输入描述语言

输入描述语言是对设计对象作出行为级的描述，并作为硅编译器的输入。

不同的硅编译器会有不同的描述语言，现以 FIRST 硅编译器为例介绍如下。

FIRST 的全名为“First Implementation of Realtime Signal Transforms”。它是专用于数字讯号处理系统的硅编译器。其描述格式如下：

CONSTANT~~

OPERATOR~~

NAME[parameters] (input controls→output controls)

```

input signals→output signals
...
END
OPERATOR...
...
CHIP...
...
END

```

现有四级级联的数字滤波器，见图 3-1。它由加法器、乘法器和延迟器所组成。其输出为

$$Y = \sum_{i=1}^4 C_i X_{n-i} \quad (3-1)$$

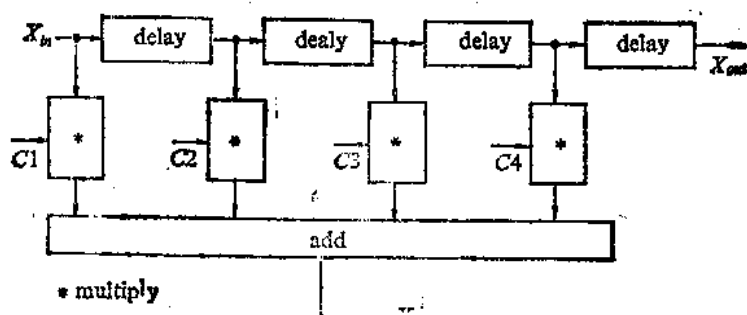


图 3-1 四级级联数字滤波器

如该四级滤波器由“Twostage”单元所组成，可见图 3-2。而“Twostage”单元的框图示于图 3-3。

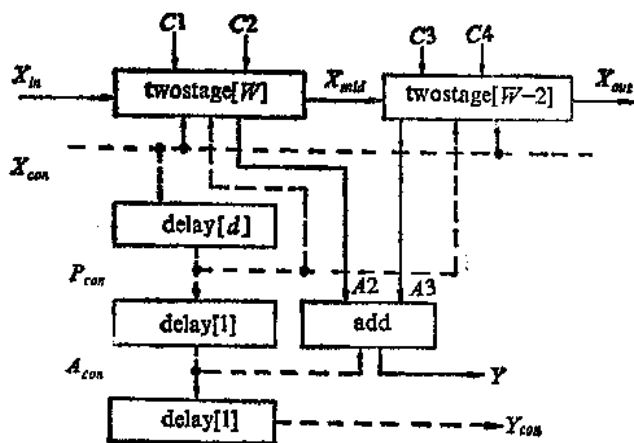


图 3-2 “Twostage”组成的四级滤波器

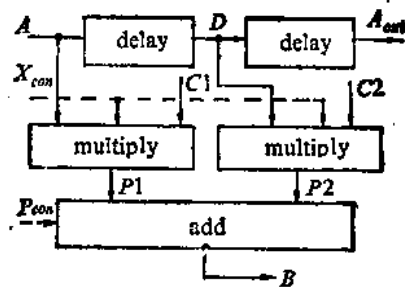


图 3-3 “Twostage”单元框图

现先对“Twostage”单元作描述，得下列结果

OPERATOR Twostage[n] (X_{con} , P_{con})

A , $C1$, $C2 \rightarrow A_{out}$, B

SIGNAL D , $P1$, $P2$

```

Delay[n] A→D
Delay[n] D→Acon
Multiply(Xcon) A, C1→P1
Multiply(Ycon) D, C2→P2
Add(Pcon) P1, P2→B
END

```

接着再对整个四级滤波器作出描述。

```

CONSTANT W=10! 数据字长度
          d=5! 等待乘法器运算的时间
CHIP FIR (Xconp→Yconp)Xinp, C1p, C2p,
          C3p, C4p→Xoutp, Yp
CONTROL Pcon, Acon, Xcon, Ycon
SIGNAL Xmid, A2, A3, Xin, Y, C1, C2, C3, C4, Xout
PADORDER VDD, Xinp, Xoutp, Yp, GND, Xconp, Yconp,
          CLOCK, C1p, C2p, C3p, C4p
Padin(Xconp→Xcon) Xinp, C1p, C2p, C3p, C4p
          →Xin, C1, C2, C3, C4
Padout(Ycon→Yconp) Xout, Y→Xoutp, Yp
Twostage[W] (Xcon, Pcon) Xin, C1, C2 →Xmid, A2
Twostage[W-2] (Xcon, Pcon) Xmid, C3, C4
          →Xout, A3
Add(Acon) A2, A3→Y
Delay[d] (Xcon→Pcon)
Delay[1] (Pcon→Acon)
Delay[1] (Acon→Ycon)
END

```

可以看出，四级滤波器的描述语言中调用了已定义的 Twostage[n]。

这种行为级描述语言对于系统设计人员，包括芯片设计人员都是十分容易掌握的。只要正确的对设计对象作出描述，硅编译器在接受输入描述后经“编译”就可得到所要求的版图描述。

§ 3.4 硅编译器的内部组成

硅编译器内部组成的框图示于图 3-4。

首先根据性能要求作出实现此性能的说明，因为同样的性能要求，可以有不同的方式加以实现。例如要求进行算术操作

$$Z = (A + B) * C - D$$

其实现方式可以有：

- (1) 通过一个功能块来执行。如图 3-5。

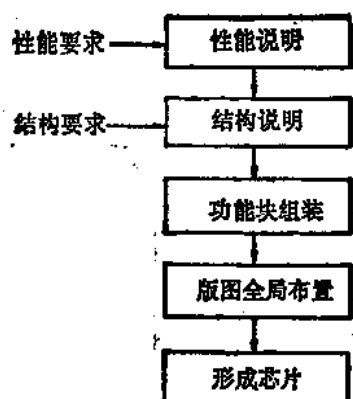


图 3-4 硅编译器内部组成

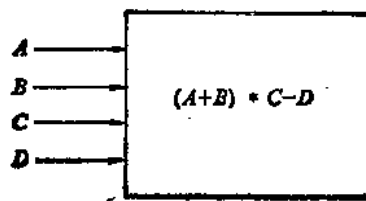


图 3-5

(2) 通过简单算术操作的组合来完成。如图 3-6。

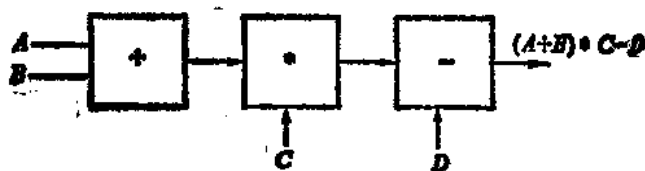


图 3-6

(3) 通过 + / - 操作器和时序及多路控制来完成。如图 3-7。

(4) 采用更一般的“数据通道”形式。示于图 3-8。

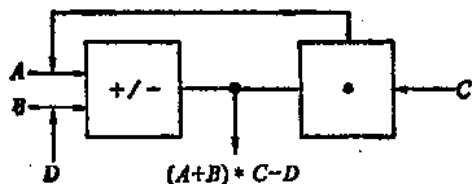


图 3-7

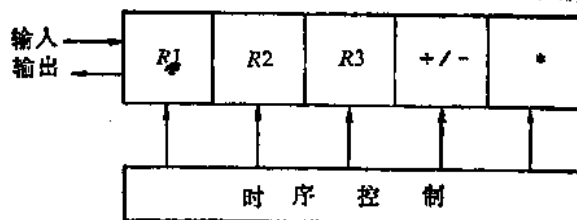


图 3-8

控制操作如下：

$A \rightarrow R1, B \rightarrow R2, R1 + R2 \rightarrow R3$

$C \rightarrow R1, R3 * R1 \rightarrow R2$

$D \rightarrow R1, R2 - R1 \rightarrow R3$

$R3 \rightarrow OUTPUT$

当然还有其它实现方式。

在确定了实现方式后，还要输入结构要求，如逻辑形式、电路形式、时钟要求和所采用的工艺规则（如几微米的设计规则）。

下一步进行功能块的组装。功能块由较低层次的单元所组成，最低层次的单元一般称

为基本单元(Leaf Cell)。大多数硅编译器采用存于数据库中的单元或功能块,把库中已有的单元或功能块组装成所需的新功能块。

库中的功能块一般有:(1)寄存器、ROM、RAM、堆栈;(2)PIA、ALU、计数器;(3)时序控制器、时钟发生器;(4)数据通道;(5)I/O 压焊块等。对于模拟电路,单元库的建立还处于初步阶段。

组装功能块时也可以根据要求进行“可编程”,例如根据状态表或布尔方程组设计PIA。

版图全局布置是将各功能块放置在芯片上的合适位置。例如“数据通道”的版图全局布置可以如图 3-9 所示。

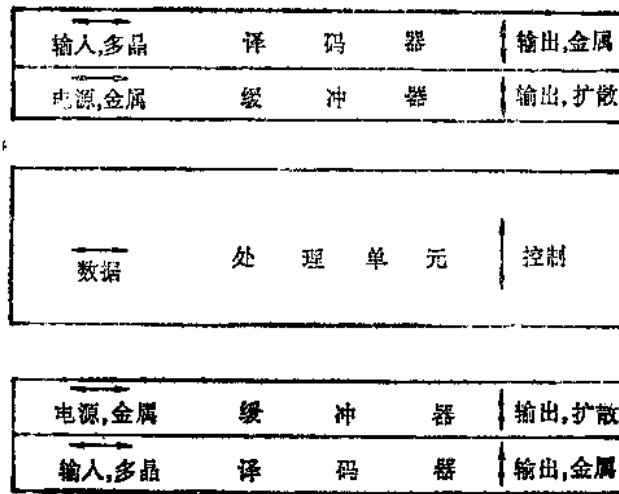


图 3-9 数据通道版图全局布置

版图全局布置一般分两类:(1)特定的平面结构;(2)功能块可任意排列。

第一类是按固定的、特殊的平面结构排列功能块。这时很容易进行布置,生成版图很快,但较浪费面积。

第二类是按内连最合理来排列功能块。进行这种版图布置时会有相当的困难,生成版图较慢。但一旦布置成功,芯片的面积较小。

特定的平面结构有以下几种:

- (1) 门阵列;
- (2) 标准单元;
- (3) 栅矩阵 (Gate Matrix);
- (4) 后进先出 (下推) 自动机, 见图3-10;
- (5) 简单微处理器, 见图 3-11, 按水平方向划分位, 按垂直方向划分功能块;
- (6) FIRST 结构, 见图 3-12。将加法器、乘法器、延迟器等放置于两边, 中间留有一布线通道用于内连接。

根据全局布置将各功能块放在应有位置后,再作布线,包括连接输入输出单元和压焊块,形成芯片得到各掩膜层信息。

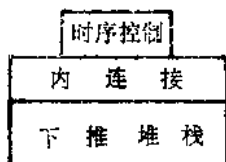


图 3-10

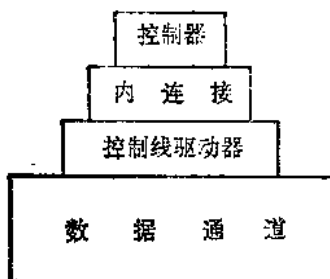


图 3-11

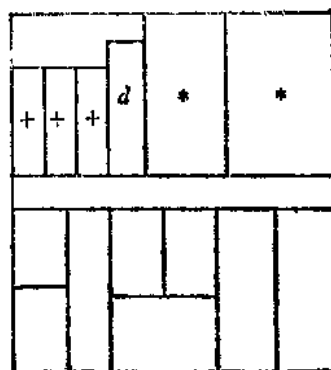


图 3-12

§ 3.5 软件环境与设计步骤

硅编译器的设计环境示于图 3-13。

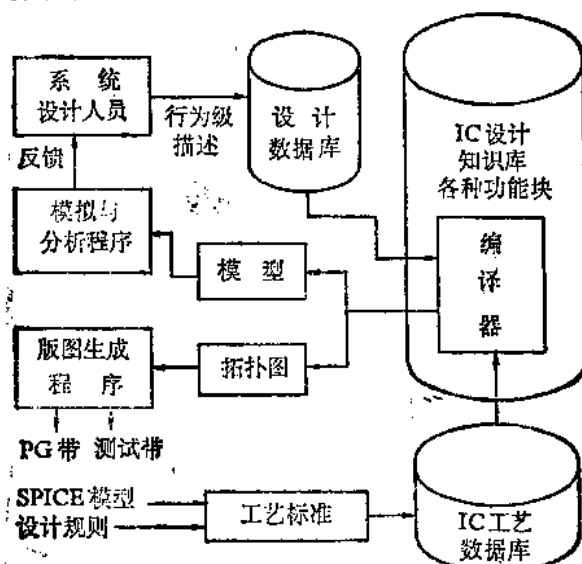


图 3-13 硅编译器设计环境

设计人员通过菜单驱动的用户接口调用所需功能块并显示在屏幕上，产生设计对象的探讨性结构。在完成了这些功能块的描述和整个芯片的描述之后，由模型产生器产生功能分析用模型、时序分析用模型及拓扑图。再由功能模拟程序验证功能是否正确，由时序分析程序分析芯片的速度和功耗，由拓扑图给出芯片所占尺寸。这些信息将反馈给设计人员，由设计人员判断是否需要作出修改，或者在几个方案中作出比较，也就是在“功能级”进行优化。这称为第一阶段。在确定一种结构后再进行细致的完善和改进，最后产生图形发生器或电子束制版设备所需的磁带、测试用磁带和其它设计文件。这称为第二阶段。

“两阶段”设计方法的框图示于图 3-14(a)。详细过程见图 3-14(b)。

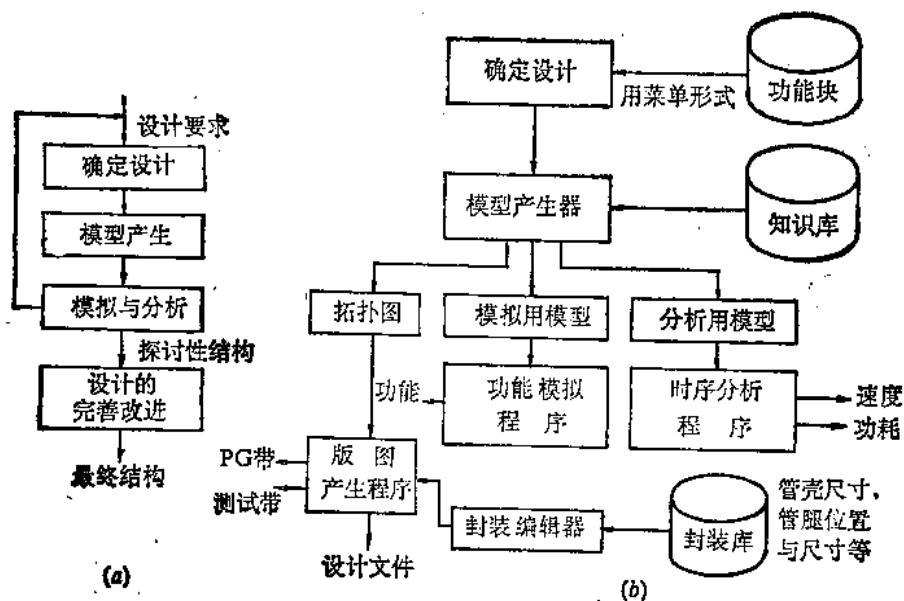


图 3-14 “两阶段”设计方法

§ 3.6 简单实例——随机逻辑编译器

在此随机逻辑编译器 (RLC) 实例中, 版图全局布置有一定的限制, 属于特定平面结构类。各个门在水平方向排成一行, 门为垂直放置, 电源线在上, 地线在下, 内连信号线处于中间。版图全局布置的示意图见图 3-15。

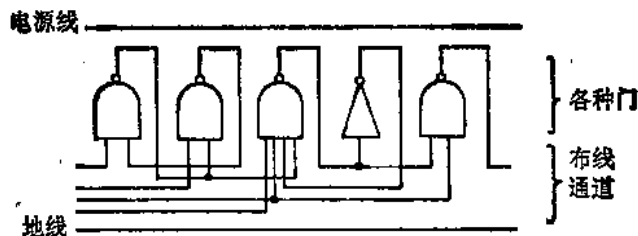


图 3-15 RLC 的版图布置

由于我们没有限制门的数目, 门之间的连接关系也不作限制, 因此电源线和地线的确切位置要根据具体电路的要求才能确定下来, 同样各个门在芯片上的物理位置也暂时无法肯定。

电源线的位置可用“全程变量”加以参量化。例如用变量 YVDD 作为电源线中心点的坐标, 用变量 POWER 作为电源线的宽度, 变量 YGND 作为地线中心的坐标。

为了具体确定各个门单元的物理位置, 需要在门单元生成子程序与编译器之间进行信息交换。即编译器所需要的一些信息由单元生成器计算出来, 而单元所需的一些信息由编译器计算而得, 并相互进行交换。

具体地说, (1) 编译器给出门单元的原点坐标 (一般为左下角的坐标), 门单元生成子程序返回给编译器该门单元的宽度。这样编译器就可以计算出下一个门单元的原点坐标。

(2) 编译器给出每根内连线的 Y 坐标, 而门单元生成子程序给出连线的终止位置 X 坐

表 3-1 RLC程序清单

```

TYPE    SIGNAL_WIRE=(FROM:GATE
                      TO:GATES
                      NAME:QS
                      PHYSICAL:PHYSICAL_WIRE
                      VLEFT,VRIGHT,VHEIGHT:INT);

SIGNAL_WIRES={SIGNAL_WIRE};

GATE= (INPUTS:SIGNAL_WIRES
       OUTPUT:SIGNAL_WIRE
       TYPE:GATE_TYPE
       INDEX:INT);

GATES= { GATE };

GATE_TYPE=SCALAR(NAND,NOR,INVERT);

CHIP= (GATES:GATES
       INPUTS,OUTPUTS,SIGNALS:SIGNAL_WIRES
       SIGNAL_COUNT:INT
       NAME,DESCRIPTION:QS);

DEFINE PHYSICAL(SW:SIGNAL_WIRE)=PHYSICAL_WIRE: SW.PHYSICAL ENDDFN

DEFINE PHYSICAL(SWS:SIGNAL_WIRES)=PHYSICAL_WIRES:
  BEGIN  VAR S=SIGNAL_WIRE;
  (COLLECT S\PHYSICAL FOR S $E SWS; )
  END
ENDDFN

DEFINE PACK(C:CHIP):
  BEGIN  VAR SWS=SIGNAL_WIRES;H=INT;G=GATE;S=SIGNAL_WIRE;
  DEFINE SORT(SWS:SIGNAL_WIRES)=SIGNAL_WIRES:
    BEGIN  VAR OUT=SIGNAL_WIRES;W=SIGNAL_WIRE;I,J,K=INT;
    DO  OUT:=NIL;
      WHILE DEFINED(SWS):DO
        I:=-1;
        FOR W $E SWS;&& FOR J FROM 1 BY 1;DO
          IF W.VLEFT>I THEN
            I:=W.VLEFT;
            K:=J; FI
          END
          OUT:==SWS[K] (&
          SWS(K-):=SWS(K+1-);
        END
      END
    GIVE OUT
  END

```

```

ENDDEFN
DEFINE DRAW_WIRE(LEFT:INT):
  BEGIN  VAR W=SIGNA_WIRE;I=INT;
  IF THERE_IS W.VLEFT>LEFT FOR W $E SWS;&& FOR I FROM 1 BY 1;
  THEN  SWS(I-)=SWS(I+1-);
        @(W).VHEIGHT:=H;
        DRAW_WIRE(W.VRIGHT); FI
  END
ENDDEFN
FOR G $E C.GATES;&& FOR H FROM 1 BY 1;DO @(G).INDEX:=H; END
FOR S $E C.SIGNALS; DO
  @(S).VLEFT:=IF DEFINED(S.TO)
    THEN S.FROM.INDEX MIN MIN G.INDEX FOR G $E S.TO;
    ELSE S.FROM.INDEX FI;
  @(S).VRIGHT:=S.FROM.INDEX MAX MAX G.INDEX FOR G $E S.TO;;
END
FOR S $E C.INPUTS;DO @(S).VLEFT:=0;END
FOR S $E C.OUTPUTS;DO @(S).VRIGHT:=999999; END
SWS:=C.SIGNALS\SORT;
WHILE DEFINED(SWS);&& FOR H FROM 1 BY 1;DO DRAW_WIRE(-1);END
END
ENDDEFN
DEFINE SETUP_DIMENSIONS(C:CHIP):
  BEGIN  VAR G=GATE;S=SIGNA_WIRE;H=REAL;
  POWER:=WIDTH(+.25 FOR G $E C.GATES;)MAX 4;
  YGND:=-9.*(MAX S.VHEIGHT FOR S $E C.SIGNALS;)-4-POWER/2;
  YVDD:=6+POWER/2 MAX 9;
  END
ENDDEFN
DEFINE INITIALIZE_WIRES(C:CHIP):
  BEGIN  VAR S=SIGNA_WIRE;
  FOR S $E C.SIGNALS; DO
    @(S).PHYSICAL:=(LEFT:999999
      RIGHT:-999999;
      HEIGHT:1-9*S.VHEIGHT
      NAME:S.NAME);
  END
  FOR S $E C.INPUTS; DO
    @(S).PHYSICAL.LEFT:=-999999.;
  END
  FOR S $E C.OUTPUTS; DO
    @(S).PHYSICAL.RIGHT:=999999;
  END
END
ENDDEFN

```

```

DEFINE DRAW_CELLS(C:CHIP)=MRG:
  BEGIN  VAR X=REAL;G=GATE;M=MRG;
  {COLLECT DO M:=CASE G.TYPE OF
    NOR: NOR(G.INPUTS\PHYSICAL,G.OUTPUT\PHYSICAL,CWIDTH)
    NAND:NAND(G.INPUTS\PHYSICAL,G.OUTPUT\PHYSICAL,CWIDTH)
    INVERT:INVERT(G.INPUTS\PHYSICAL,G.OUTPUT\PHYSICAL,CWIDTH)
    ENDCASE;
    GIVE M
  FOR G $E REVERSE(C.GATES);}
  END
ENDDFN

DEFINE DRAW_WIRES(C:CHIP)=MRG:
  BEGIN  VAR S=SIGNAL_WIRE;LEFT,RIGHT=REAL;
  DO LEFT:=CWIDTH+5;
    RIGHT:=-2;
  GIVE {COLLECT WIRE(BLUE,3,{S.PHYSICAL.LEFT#S.PHYSICAL.HEIGHT;
    S.PHYSICAL.RIGHT#.})
    FOR S $E C.SIGNALS;
    EACH_DO @(S.PHYSICAL).LEFT:=MAX LEFT;
      @(S.PHYSICAL).RIGHT:=MIN RIGHT;;;
    BOX(BLUE,CWIDTH+3#YVDD_POWER/2\TO 4#YVDD+POWER/2);
    BOX(BLUE,CWIDTH-1#YGND-POWER/2\TO 0#YGND+POWER/2)}
  END
ENDDFN

DEFINE LOAD(S:SIGNAL_WIRE)=REAL:
  BEGIN  VAR G=GATE;T=SIGNAL_WIRE;
  (+ CASE G.TYPE OF
    NOR: 1
    INVERT: 1
    NAND: +1 FOR T $E G.INPUTS;
  ENDCASE FOR G $E S.TO;)*Q_LOAD +
  LOAD(BLUE,WIDTH(BLUE),S.PHYSICAL.RIGHT-S.PHYSICAL.LEFT)
  END
ENDDFN

DEFINE COMPILE(C:CHIP)=MRG:
  BEGIN  VAR M=MRG;G=GATE;S=SIGNAL_WIRE;
  DO CWIDTH:=0;
    PACK(C);
    SETUP_DIMENSIONS(C);
    INITIALIZE_WIRES(C);
    M:=DRAW_CELLS(C);
    M:={M;DRAW_WIRES(C)};
  GIVE M
  END
ENDDFN

```

标。这样连线子程序就可以确定连线的具体位置。

第一个信息交换是通过直接的参量传送。编译器给门生成子程序一个 REAL 值，即原点坐标。每个门通过全程变量 CWIDTH 返回该门单元的宽度。

第二个信息交换则通过 PHYSICAL-WIRE 这一数据类型进行。编译器给 PHYSICAL-WIRE 以 Y 值，门单元查找此信息，并把 X 值给予 PHYSICAL-WIRE，这样每根连线就伸展到规定的 X 位置。

表 3-1 是这种编译器的程序清单，它是用 ICL 语言编写的。

该编译器包括以下几部分：

- (1) 门单元生成子程序 DRAW-CELLS;
- (2) 产生连线子程序 DRAW-WIRES;
- (3) 连线初始化子程序 INITIALIZE-WIRES;
- (4) 分配通道子程序 PACK;

此子程序给每条内连线分配一个通道号 INDEX，并希望不同的内连线尽可能共用一个通道，以减小通道数。

- (5) 门单元排列子程序 SORT;

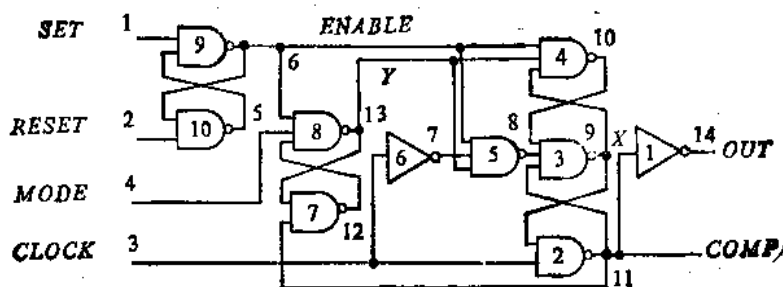


图 3-16 脉冲同步电路

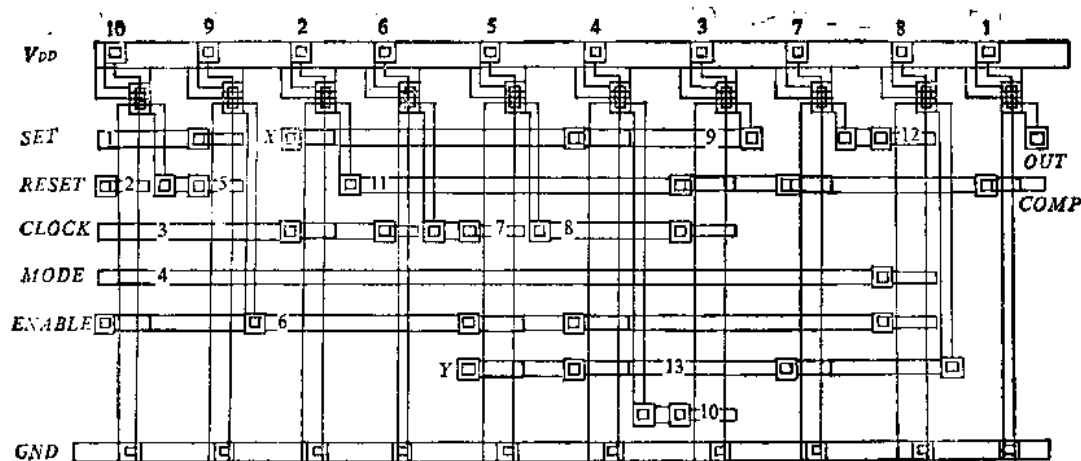


图 3-17 “编译”后所得版图

此子程序按照连线长度最小和连线通道数最小来排列各个的门次序。

(6) 全程变量设置子程序 SETUP-DIMENSIONS

此子程序确定 YVDD, YGND 的大小以及根据通道号 INDEX 确定各条连线的高度。COMPILE(C:CHIP) 过程调用以上这六个子程序完成整个芯片的设计。

对于图 3-16 所示的脉冲同步电路, 经编译后得到图 3-17 这样的版图。

图 3-17 所示的版图形式是在采取图 3-15 那样的特定的、有限制的全局布置后所取得的, 也就是说表 3-1 对应的硅编译器只能设计这种特定结构的芯片。

§ 3.7 硅编译器的优点与限制

从本章的讨论可以看出硅编译器的优点是明显的。设计人员可以用自己熟悉的语言对所设计的集成电路芯片作行为级的描述, 且可在“功能级”上进行优化, 比较不同的方案。在方案确定前可进行功能模拟和时序分析, 并得到拓扑图的尺寸, 这就能帮助设计人员作出判断。一旦确定了方案, 硅编译器将自动地得到掩膜版图及封装信息, 这就大大地缩短了设计周期, 免除了大量繁复、易出错误的版图设计阶段, 并保证没有设计规则和电学规则的违反。所以它是一种很有吸引力的设计方法。

硅编译器可以说是与工艺无关的。设计人员只要规定采用何种电路形式, 何种制造工艺就可得到相应的版图。但这是以大量的硅编译程序的编写工作量为代价的。如果工艺发生变化, 设计规则发生变化, 必然要求编译器内部程序的重写, 这项工作同样是很繁重的。目前正在发展一种库内单元和功能块能随设计规则的变化而改变的“可变单元”库。

此外, 硅编译器厂商还开发了一种工具提供给用户, 使他们能按照规定的语言编写出一个功能块编译器, 并把该功能块编译器送入原有的硅编译器中以丰富原编译器的功能。

目前硅编译器的应用受到限制的一个重要因素是版图全局布置。在一个硅编译器中, 可能有的版图全局布置的类型是有限的。全局布置越固定, 越受约束, 则硅编译器可设计的芯片类型就越少。反之, 全局布置越任意, 可设计的芯片类型就越多。但这种布置上的任意性, 会给硅编译器程序的编写带来极大的困难。从另一角度看, 版图全局布置越特定, 则所得版图通常更优化, 性能也更佳。为了解决这一矛盾, 往往设计成几种编译器, 每种编译器只适合于某一种类型的芯片或者只适用于芯片中某种类型的功能块。

当前硅编译器所设计的芯片面积比起人工设计或其它方法设计的还大很多, 也许这是一个较难解决的问题。因此这种全自动设计方法还有待不断完善。

第四章 逻辑模拟

§ 4.1 逻辑模拟程序分类

在超大规模集成电路的设计过程中,一旦确定了该芯片的功能要求后,首先是进行逻辑设计。逻辑设计的自动化技术正在得到大力的开发。但是不管是自动进行逻辑设计还是采用人工设计,在逻辑设计完成后,验证逻辑设计的正确性都是一个非常重要的问题,经过验证就可以把逻辑设计中的错误消灭在电路设计和版图设计之前。此外,在版图设计之后,还需要进行考虑寄生效应的“后”逻辑模拟,以进一步确认芯片设计是否符合性能要求。

因此,逻辑模拟不仅是逻辑设计阶段一个必不可少的步骤,也是整个设计验证中的一个不可缺少的步骤。

作为逻辑设计过程中的第一步是进行简单的逻辑验证,它只需验证布尔方程的有效性,通常只采用零延迟或单位延迟,两值或三值模拟。设计者可以改变设计或输入讯号而重复模拟直到符合设计要求。第二步是要进行逻辑延迟模拟。这时需要考虑各个器件之间实际的时序关系,包括尖峰、冒险竞争等问题。这时的逻辑延迟是一种预估性延迟,它是从每个器件的扇出数中估算而来。进行这一步有两个作用:(1)把时序错误消灭在版图设计之前;(2)在版图前预估芯片的性能。如果模拟结果不符合要求,则要作出必要的修改。

第三步是在版图设计以后的模拟,它是精确的延迟模拟,延迟时间是从版图数据中计算得来,因而在这三次模拟中它是最精确的。

图 4-1 示出这三步的前后次序。

按照被模拟器件抽象化的层次,逻辑模拟程序可以分成四类,示于表 4-1。

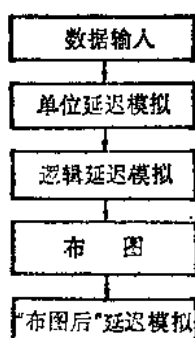


图 4-1 设计验证过程

表 4-1 逻辑模拟程序分类

程序分类	被模拟对象的层次
开关级	晶体管
门级	与门、或门等
功能级	加法器等
寄存器传输级	如 I/O 部件, 存储器, 总线

开关级模拟是一种专门的逻辑模拟,它比通常的门级模拟可更精确地模拟 MOS 电路。这时把逻辑门进一步展开成晶体管,而每个晶体管被看成是一个独立的、理想的开关。在

开关级模拟中可以模拟双向传输门、不同的讯号强度等问题。典型的开关级模拟程序有 MOSSIM、ESIM 等。

门级模拟所模拟的器件为逻辑门，例如倒相器、与门、或门等，每个器件只允许有一个输出端。模拟结果显示输入/输出行为。在某些门级模拟程序中也允许有“宏门”存在。但在这层次上，芯片中哪些是控制讯号、哪些是数据讯号已难于判别，因为门级的描述仅仅是一种构造性描述，它是把已知的逻辑门行为和内连接关系描述出来。典型的程序有 TEGAS、CADAT 等。

功能级模拟所模拟的器件为功能块，例如锁存器、触发器、计数器和寄存器等。这些器件的输出端可以多于一个。

寄存器传输级模拟具有这样的能力，凡是能用寄存器传输语言（例如 DDL 语言）描述的电路都可以被模拟。例如输入/输出控制部件在寄存器传输级模拟程序中只作为一个器件。其它部件可以是存储器、组合逻辑部件和总线等。在这一层次上，控制讯号和数据讯号是完全分开的。典型的描述语言有 CDL 和 DDL 等。

实际的逻辑模拟程序往往不只是针对某一层次，例如门级模拟程序可以模拟触发器这样的功能块。此外目前还开发了一种混合级模拟程序，它所模拟的器件可以从寄存器传输级直到门级或开关级。这样设计人员就可以进行 VLSI 电路的设计而又能获得精确的模拟结果，对于有精确需要的部分可进行门级模拟而其余部分只进行寄存器传输级模拟。

从程序内部执行过程来看，完成模拟的方式有三种：（1）面向活动的。它以固定的时间间隔求得元件的逻辑值，而不管在这时间间隔中逻辑值有没有发生变化。（2）面向事件的。它只对那些参数值发生变化的元件求值，这时对于未发生变化的元件就不必求值。

（3）面向过程的。它允许设计者孤立地模拟每个模块，每个模块内部的事件与其它模块中出现的事件无关。

§ 4.2 逻辑模拟的一些基本概念

一、信号值

习惯上逻辑模拟得到的信号状态是逻辑值而不是电压值。

存在于实际电路中的信号值有两类：一种是稳态值即<0>、<1>和<Z>（高阻态，常出现在三态电路中）。另一种是非稳态值如<E>（脉冲或尖峰态）和<S>（振荡态）。

有一些信号值只存在于逻辑模拟程序中，如<X>（不确定），<U>（从 0 上升到 1），<D>（从 1 下降到 0），<T>（0 与 1 之间转换）等。

最简单的逻辑模拟程序只有二值即<0>和<1>。但二值模拟有严重缺陷，近年来已很少使用。它的弊端是所有信号只能设置为<0>或<1>，而不能变成<0>或<1>以外的其它值。但在实际电路中如 R-S 触发器，由于输出端只能是<0>和<1>，因此在 R 端和 S 端不能同时加<1>信号。为此发展了三值模拟，引入了一个不确定值<X>。当然还可以发展成四值即<0>，<1>，<X>和<Z>。

多值逻辑模拟通常是在采用精细的延迟时间时才有效，例如采用最小-最大延迟时间时可能有六值，即<0>，<1>，<X>，<U>，<D>和<E>（指出现尖峰、脉冲或竞争）。

二、延迟模型

当脉冲信号通过元件时总是存在一定的传播时间, 在建立逻辑模拟的模型时必须加以考虑。但时间精度考虑得越细, 模拟程序在执行上则越复杂。因此在处理延迟时间上有以下几种方法。

(1) 零延迟 所有的元件都认为没有延迟即 $\Delta T=0$ 。这时只能检查布尔表达式的正确性, 对于异步时序电路就无法适应。

(2) 单位延迟 这时假定所有元件具有相同的延迟, 即 $\Delta T=1$ 。 ΔT 本身的绝对值根据电路情况而定。它也只能用于逻辑验证。

(3) 指定延迟 对于不同的元件类型赋以不同的延迟时间, 或者对于各个元件赋以不同延迟。例如同样的与非门, 对 TTL 类型可赋以 $\Delta T=20$, 而对 ECL 类型则赋 $\Delta T=5$ 。

如果对于上升时间和下降时间分别指定不同的值, 则更接近实际。为了使模型更精确, 还可根据器件的扇入和扇出数赋以不同的延迟时间。例如两输入端与门则 $\Delta T=20$, 而对四输入端与门则赋以 $\Delta T=24$ 。

采用这种模型后, 可完成尖峰分析以及有限的竞争冒险分析。

(4) 最小-最大延迟 这时延迟由一对最小和最大值来确定。最小值对应于元件开关动作时可能的最早时间, 而最大值对应于可能的最晚时间。例如对于一个与门, 可设其最小延迟为 3, 最大延迟为 4, 则 ΔT 可写成 $\Delta T=3+1$ 。

单位延迟与最小-最大延迟的比较示于图 4-2。

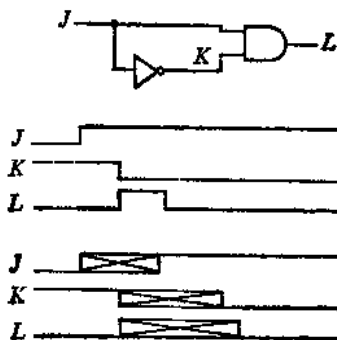


图 4-2 单位延迟与最小-最大延迟比较

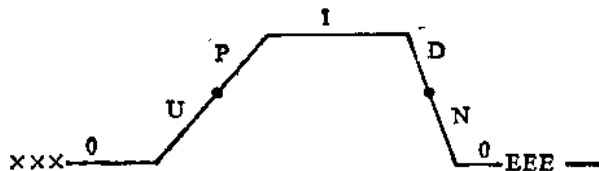


图 4-3 惯性延迟的多值图

(5) 惯性延迟 这是指在逻辑门的输入改变后, 在某一时刻 t_1 前门的输出不变, 在某一时刻 t_2 后开始变化, 而在时刻 t_3 发生确定的变化。由此再增加两个值 P 和 N。

在输入改变和输出还没有变化这一段时间内, 门的输出被认为保持在原有状态。在惯性延迟以后输出改变, 同时输出的扇出元件得到处理。

惯性延迟的多值图示于图 4-3。

三、逻辑强度

对于一些新的逻辑模拟程序引入了这一概念, 这对 MOS 电路中分析传输门和一些线与、线或等“线连”逻辑非常有用。譬如在 $\langle 0 \rangle$, $\langle 1 \rangle$, $\langle x \rangle$ 三值外还增加了三种强度如 $\langle \text{Driving} \rangle$, $\langle \text{Weak} \rangle$ 或称 $\langle \text{Resistive} \rangle$ 和 $\langle \text{High impedance} \rangle$, 这常被称作九态逻辑模拟。九态之间的关系见图 4-4。从强度来讲, $\langle \text{Driving} \rangle$ 最强, $\langle \text{Weak} \rangle$ 或 $\langle \text{Resistive} \rangle$ 次之, $\langle \text{High impedance} \rangle$ 最弱。从阻抗的角度考虑, 则其次序反之。 $\langle \text{High impedance} \rangle$ 阻抗

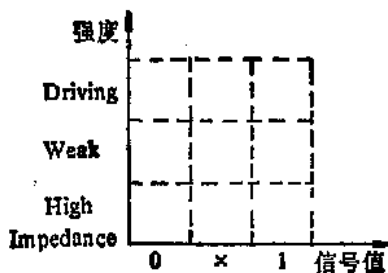


图 4-4 九态之间的关系

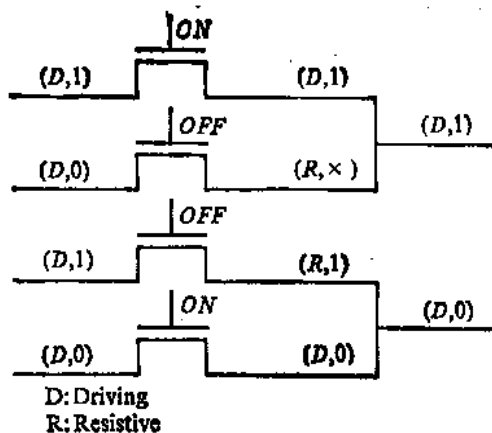


图 4-5 按规则求出节点状态

最大，而<Driving>阻抗最小。

当两个门（或多个门）控制同一节点而相互竞争时，该节点的状态服从以下规则：

（1）当所有的输入信号具有相同的状态并连在一起时，该节点的状态与输入信号相同，其强度变成为输入信号中强度最强者。

（2）当输入信号具有不同的状态时，则该节点的状态取决于输入信号中强度最强的那个信号的状态。

（3）如果输入信号具有不同的状态而又具有相同的强度，则该节点的状态不确定。其例子示于图 4-5。

也可设置四种强度，如 <Driving>，<Weak>，<Weaker>，<High impedance>。强度数目的增加并不产生附加的 CPU 时间。

四、求值过程

在逻辑模拟程序中对每种类型的元件都相应地编写一个求值过程（子程序）。

例如对二输入与非门有

```
DO 10 I=1, NG(1)
10 OUT (G1(I))=1 - (IN(A1(I,1))•AND•IN(A1(I,2)))
  :
```

这里，G1 为二输入与非门的类型号；

NG(1) 为二输入与非门的数目；

IN(A1(I,1))，IN(A1(I,2)) 分别代表两个输入端的逻辑值，OUT(G1(I)) 表示该门的输出逻辑值。

若给定了元件输入端信号状态，该求值过程就通过计算并考虑延迟而得到输出端信号的变化时间和逻辑状态。

功能计算的基础是三值或多值逻辑代数运算法则。对逻辑门将计算出新的输出值 (<0>，<1>，<Z>或<x>)。对功能部件如移位寄存器、ROM、RAM 等也通过求值过程得出新的状态，所不同的在求值子程序中允许引入功能块的参数和可选项。

通常程序还提供一种用户自定义的功能。由用户对自定义的功能部件进行逻辑功能描述，并通过编译将自定义的求值过程加到模拟程序中去。

五、选择性跟踪

为了减少需要求值的元件数，在程序中通常采用选择性跟踪。凡是输出有变化的就加以跟踪，无变化的不再跟踪，这就可以大大减少运算时间。据估计，对于通用计算机，发生变化的元件数只占总元件数的 2.5%，在这种情况下，采用这种技巧可使模拟速度加快四十倍。选择性跟踪的示意图见图 4-6。

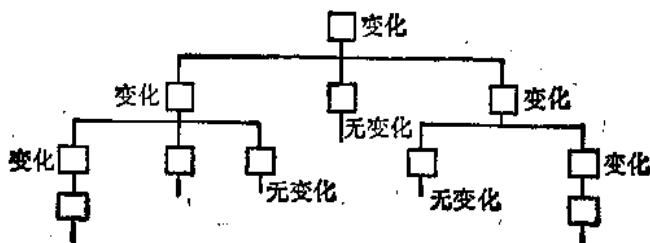


图 4-6 选择性跟踪

六、选择性事件

对于那些输入有变化但输出没有变化的元件并不送入事件表，不再求值。例如对于与门，只要有一个输入端为零，即使其它输入端有变化时也不将此与门送入事件表。这样也可大大加速模拟过程。

§ 4.3 逻辑模拟的方法

逻辑模拟程序的框图如图 4-7 所示。模拟的方法有好几种，现分别讨论如下。

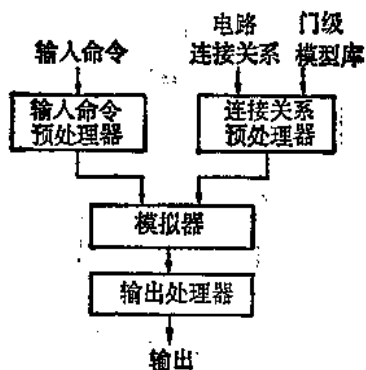


图 4-7 逻辑模拟程序框图

4.3.1 编译方式

编译方式的模拟程序是每隔一个时间步长对每个逻辑门进行求值。

编译方式是将被描述的逻辑电路转换成一个可执行的程序。每个逻辑元件按其功能编写成一个子程序，再按各逻辑元件之间的连接关系把各子程序连接起来构成程序源。

它对每个逻辑门先进行定义。每个门有门号 S 。 $INP(S)$ 表示 S 门具有的输入端

数。 $FI(S, j)$ 是门的扇入关系，其中 $j=1, 2, \dots, INP(S)$ ，表示输入端号。 $VAL(S, t)$ 表示逻辑门 S 在时间 t 时的逻辑值。 $F(S, \text{value of inputs})$ 表示 S 门的逻辑功能。

现以图 4-8 的三输入端或非门为例说明如何进行定义。定义结果有

$$\begin{aligned} INP(S) &= 3 & FI(S, 1) &= 5 \\ FI(S, 2) &= 9 & FI(S, 3) &= 2 \end{aligned}$$

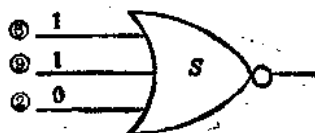


图 4-8

$$VAL(S, t) = 0 \quad F(S, \{1, 1, 0\}) = 0$$

然后根据时间 $t-1$ 时的输入求出时间 t 时的输出值。

对整个电路来讲, 从 $t=1$ 到 T 时间, 对所有的 S 门, 求出 $VAL(S, t) = F(S, t-1$ 时各输入端的值)。

现以图 4-9 的电路为例。根据门的定义和连接关系得表 4-2。

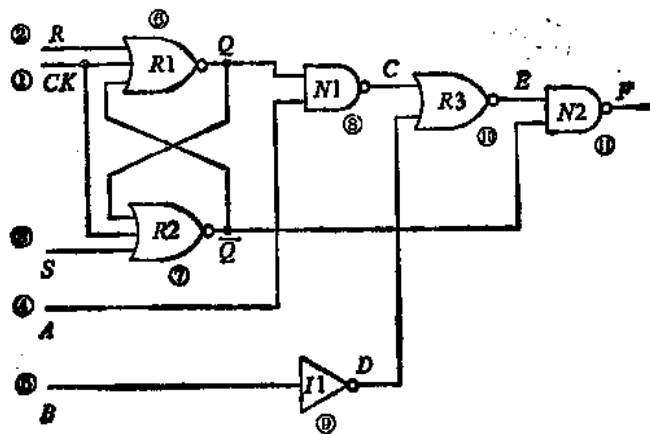


图 4-9 被模拟电路

表 4-2

门 名	门 号	INP (S)	F1 (S,1)	F1 (S,2)	F1 (S,3)
CK	1	0			
R	2	0			
S	3	0			
A	4	0			
B	5	0			
R1	6	3	1	2	7
R2	7	3	1	3	6
N1	8	2	6	4	
I1	9	1	5		
R3	10	2	8	9	
N2	11	2	10	7	

在时间点 $t=1, t=2, t=3$ 对每个门求值, 其结果示于表 4-3。从表中看出, 如采取这种方法, 在三个时间点上总共要对 6 个门进行 18 次求值, 这样太花费时间, 特别当电路规模很大时。为了减少需要求值的门数, 可采用一种排序方法。排序的原则是: 按照每个元件与外输入节点或存储元件之间的深度来排序。对存储元件规定是零序。

对图 4-9 电路的排序结果见表 4-4。排序后的模拟过程就按照升序次序逐级对门求值。同序的元件其计算顺序可以随意, 但对不同序的元件必须服从由低到高逐步计算的原则, 这类似于信号的传递过程。按排序原则计算可得表 4-5 的结果。可以看出, 在四个时间点上总共只对 6 个门进行了求值, 大大减少了求值时间。

表 4-3

	CK	R	S	A	B	R1	R2	N1	I1	R3	N2
初 始 值	1	0	1	1	1	0	0	1	0	0	1
输入变化	0	0	1	1	1						
$t=1$						1	0	1	0	0	1
$t=2$						1	0	0	0	0	1
$t=3$						1	0	0	0	1	1

表 4-4

门	序 号
R1, R2	0
N1, I1	1
R3	2
N2	3

在排序模拟时可将零延迟和单位延迟两种模式结合起来。在单位延迟模式时,系统时钟变化,所有的存储元件的内容得到修改。在零延迟模式时,存储元件的输出传播到组合逻辑元件。

这种方法的优点是编写模拟程序比较容易,能处理同步逻辑,但往往忽略竞争和冒险情况,也难于处理多延迟模拟和故障模拟。

表 4-5

	CK	R	S	A	B	R1	R2	N1	I1	R3	N2
初 始 值	1	0	1	1	1	0	0	1	0	0	1
输入变化	0	0	1	1	1						
$t=1$						1	0				
$t=2$								0	0		
$t=3$										1	
$t=4$											1

4.3.2 表格驱动和面向事件方式

在这种方式中把需要模拟的电路用各种表格来描述。为了计算各个元件的逻辑值同样需要对各类元件编写一个子程序。但在模拟时只计算那些输出发生变化的元件,也就是面向事件方式。

所谓事件,是指一个元件的一个输出线上具有了一个新的逻辑值。新的值可能与原有值不同,但也有可能相同,如果相同则取消此事件。面向事件,就是对输出线发生变化的元件进行求值。同时还采取选择性跟踪以节省运行时间。

表格分成功能描述表、电路描述表、扇入扇出表等。现仍以图 4-9 所示电路为例加以具体说明。

1. 功能描述表

每种类型的元件具有一项，它包括元件的类型名、输入端数和输出端数等。对于图 4-9 电路可以参见表 4-6。

表 4-6 功能描述表

存 储 位 置	门 类 型 名	输 入 端 数	输 出 端 数
201	INV	1	1
202	NAND2	2	1
203	NOR2	2	1
204	NOR3	3	1

2. 电路描述表

每个元件都有一项，次序按元件号大小排列。它包括元件名、功能表指针、扇入扇出表指针、扇出数、门延迟和逻辑值等，详见表 4-7。

表 4-7 电路描述表

门 号	门 名	功能表指针	扇入扇出表指针	扇 出 数	延 迟	逻 辑 值
6	R1	204	301	2	3	1
7	R2	204	306	2	3	0
8	N1	202	311	1	2	0
9	I1	201	314	1	1	1
10	R3	203	316	1	2	0
11	N2	202	319	1	2	1

3. 扇入扇出表

它包含每个元件的扇入扇出信息，见表 4-8。

在这种方式中还设立一个时间数组，时间数组中每一项代表模拟时间过程中的一个时间点，时间按固定增量增加，因此事件只能出现在固定时间增量的整数倍的那些时间点上。

时间数组 TQ 见图 4-10。数组中每项存放一个指针，该指针指向在那一时刻发生的事件表。

事件的执行是沿着输出线进行的。先找到该元件扇出连接的所有元件。再将这些元件安排在一个新的时间点上（或称未来事件）。每个未来事件用一个元件名和该元件输出线上的值来描述。同一时间发生的各个事件合起来形成一个事件表。待 TQ 指向该时刻时，

表 4-8 扇入扇出表

存储位置	门	说 明	存储位置	门	说 明
301	CK	扇 入	312	R1	扇 入
302	R	扇 入	313	R3	扇 出
303	R2	扇 入	314	R	扇 入
304	N1	扇 出	315	R3	扇 出
305	R2	扇 出	316	N1	扇 入
306	CK	扇 入	317	I1	扇 入
307	S	扇 入	318	N2	扇 出
308	R1	扇 入	319	R3	扇 入
309	N2	扇 出	320	R2	扇 入
310	R1	扇 出	321	F	扇 出
311	A	扇 入			

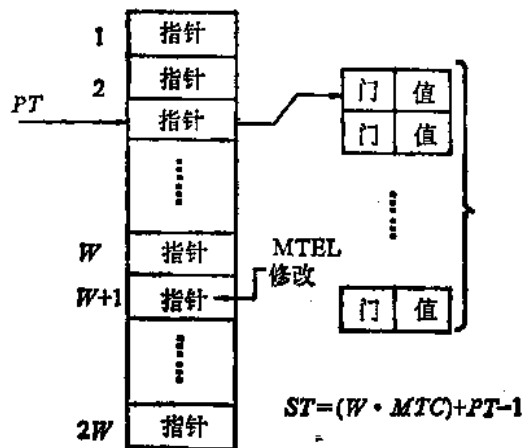


图 4-10 时间数组和事件表

这些元件按照前级输出线上的新值即新的输入值重新求值。每个元件根据其类型和模型由相应的求值过程加以求值。

采用非零延迟模型时，当输入 CK 从 1 $\rightarrow$ 0 变化时，面向事件的模拟过程见表 4-9。在当前时间处理各个事件，修改每个元件的输出，接着跟踪扇出得到被连接的元件，将这些元件插入到新的时间点上，即被时间数组 TQ 中新的指针所指向。程序框图示于图 4-11。

很明显，如果要求模拟的时间很长，就需要一个很大的时间数组。为了克服这个矛盾，现采用周期性循环技巧。让 TQ 数组具

表 4-9

时 间	事 件	变 化	扇 出	安排在新时间
0	CK	1 $\rightarrow$ 0	R1 R2	R1 $\rightarrow$ 13 R2 $\rightarrow$ 13
3	R1	0 $\rightarrow$ 1	N1 R2	N1 $\rightarrow$ 15 R2 $\rightarrow$ 16
	R2	无变化		
5	N1	1 $\rightarrow$ 0	R3	R3 $\rightarrow$ 17
6	R2	无变化		
7	R3	0 $\rightarrow$ 1	N2	N2 $\rightarrow$ 19
9	N2	无变化		

有 $2W$ 项（而不是 W 项），并令 MTC 为一个变量，它表示在时间数组中循环的周期数，此变量从零开始。 PT 指示当前时间，即 TQ 数组中正在处理的当前项，但它是从1开始。所以模拟时间

$$ST = (W \cdot MTC) + PT - 1$$

另外开辟一个存储空间，这个区域称宏时间事件表。在此表中存储那些在 TQ 时帧范围以外的时间点所发生的事件。 TQ 的时帧是从 $MTC \cdot W$ 到 $(MTC \cdot W) + W - 1$ 。宏时间事件表也是按事件出现的时间先后排列的。宏时间事件表示于图4-12。

当 PT 进行到 $W+1$ 项时，总是安排其发生一个称为“宏时间事件表修改”事件。

这个事件的动作如下：

(1) 将第 $W+1$ 项到 $2W$ 项替代 TQ 数组中的第1到第 W 项。其示意图见图4-13。

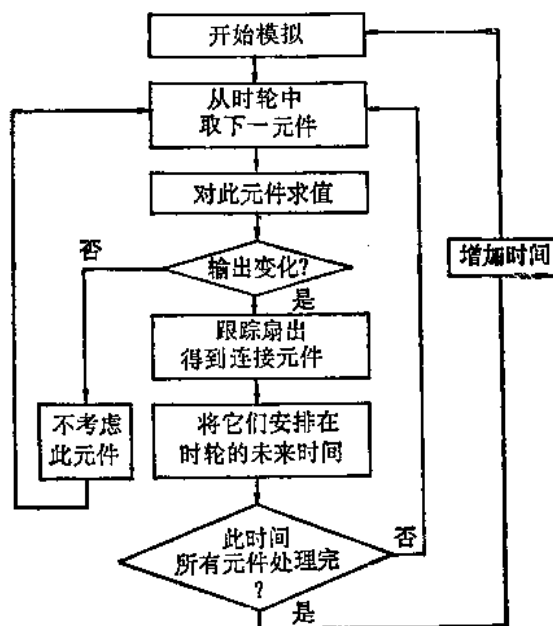


图 4-11 程序框图

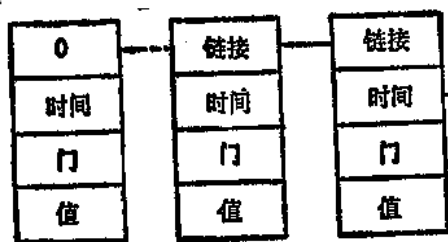


图 4-12 宏时间事件表 (MTEL)

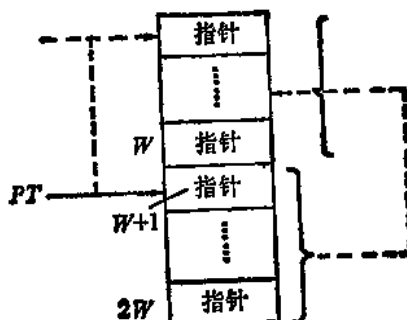


图 4-13 TQ 数组的修改

(2) MTC 增 1。

(3) PT 设置为 1。

(4) 再次安排修改事件将在 PT 指向 $W+1$ 项时发生。

(5) 搜索宏时间事件表，看在 $MTC \cdot W$ 时间点到 $(MTC \cdot W) + W - 1$ 时间点之间是否有事件出现。如果存在事件，就把它从宏时间事件表中移出，转到事件表中。

从以上讨论可以看出，被模拟的事件总是被安排在合适的表中。如果事件发生在 TQ 数组的时帧范围内，就放置在由 TQ 数组指针所指向的事件表内。如果事件在 TQ 数组时帧范围之外发生，于是就将它们插入宏时间事件表中。

所以这样的时间数组相当于一个时轮 (Timing Wheel)，见图 4-14。

时轮的每个槽相当于模拟时间的不同时刻。每个槽指向那一时刻的事件表，处理当前事

件时由求值过程计算出新的输出信号并记录下来,按选择性跟踪得到被连接的元件,并作为未来事件插入时轮中对应某个时刻的槽中。

事件驱动的模拟程序虽然编写比较复杂,但它能处理组合逻辑电路,以及同步和非同步电路。此外,它还能处理延迟,因而被广泛采取。

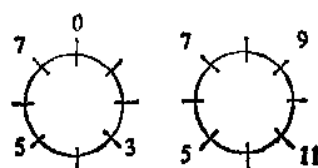


图 4-14 时 轮

4.3.3 硬件加速方式

随着集成电路规模和复杂度的增加,花费在逻辑模拟和故障模拟方面的时间越来越长。

现有模拟程序的模拟速度难于加速的主要原因是由于在通用计算机中必须顺序地处理各个元件,而在实际电路中信号是沿着各条通路同时传播的。此外,在通用计算机中每个求值过程和信号传递都要用较长的运行时间。

为了提高速度,近年来开发了一种特殊的硬件逻辑模拟器,常称为 Hardware Engine。这种硬件模拟器的特点有两个:(1)平行处理;(2)由硬件实现算法。其硬件结构分别有:硬件事件驱动、多处理器结构和阵列处理器结构。

例如 IBM 公司的 YSE (Yorktown Simulation Engine),它采取多处理器结构和流水线方式,共用 256 个平行的逻辑处理器,每个处理器能以平行方式模拟 4096 个门,因而最多能处理一百万个门。其算法采用编译方式。

采取硬件事件驱动结构的有 ZYCAD 公司的 Logic Evaluator(LE)。它的事件处理器可扩展到 16 个。每个事件处理器采用五级流水线机制并执行典型的事件驱动方式算法。最多它能处理一百五十万个门。类似的硬件模拟器,有 Daisy CAE 系统中的逻辑模拟器“Megalogician”,它采取事件驱动的流水线方式,其框图示于图 4-15。另一个例子是 Valid Logic 的 CAE 系统,它的逻辑模拟器称为“Real Fast”。它在流水线的安排上略有不同,见图 4-16。

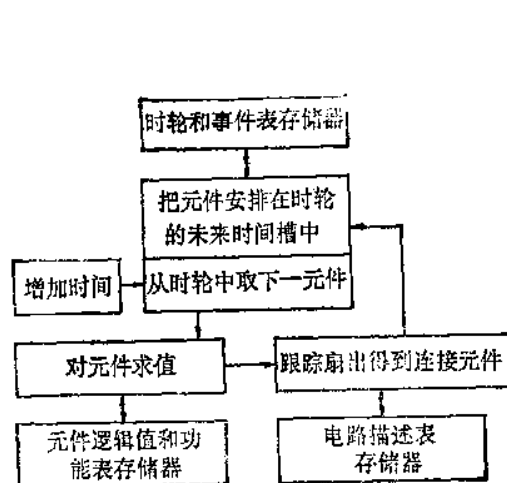


图 4-15 “Megalogician”中逻辑模拟器

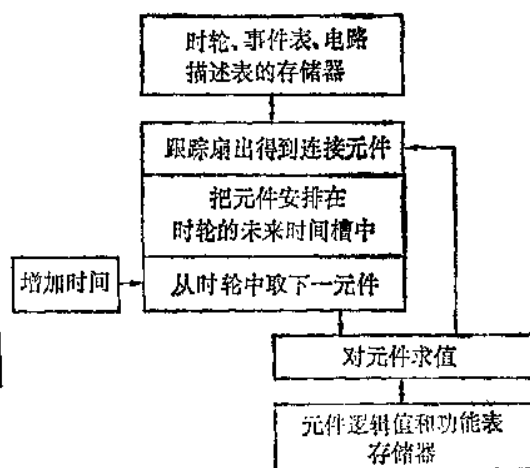


图 4-16 “Real Fast”逻辑模拟器

采用阵列处理器结构的有 NTT 公司的 AAP (Adaptive Array Processor) 模拟器。它采用单指令流多数据流的 (SIMD) 细胞阵列处理器, 内部有一个包含 256×256 个处理单元 (PE) 的阵列, 一个数据缓冲存储器, 一个具有指令存储器的阵列控制单元以及一个接口单元。其框图示于图 4-17。由于采用了大规模的阵列处理器, 可以完全实现平行的门求值过程和信号传播, 因而其求值和传播类似于真正的逻辑电路的行为, 所以可以认为这种模拟器是一种“可编程”硬件模拟器。

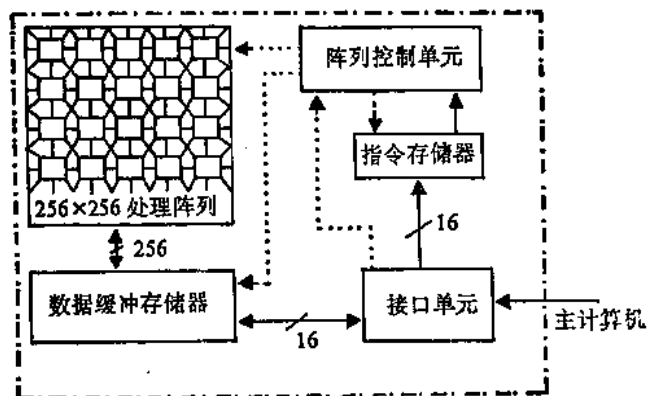


图 4-17 阵列处理器结构的模拟器

将这种结构的模拟器的模拟速度与事件驱动的软件模拟程序 (ELSA) 的模拟速度作了比较, 对于 2105 个门的例子, 前者的速度比后者快 770 倍。

§ 4.4 SPLICE 模拟程序及其应用

SPLICE 是由美国加州大学伯克利分校开发的一种用于 MOS 大规模集成电路的混合型模拟程序。但这里的混合型不是前面提到的从寄存器传输级到开关级或门级的混合, 而是指逻辑模拟和电路模拟的混合。

它可以对一个电路同时进行逻辑模拟和电路模拟, 对于那些需要详细知道电压波形的部分 (指该电路的某一部分) 可应用电路模拟, 而对那些只要进行功能验证或了解一阶时序信息的部分 (指该电路的其余部分) 可采用逻辑模拟。

该程序主要包括三个部分, 由 RATFOR FORTRAN 语言写成。第一部分为 BLT (Berkeley Language Transfer)。它用作输入处理器。包含电路描述的输入描述文件将送入 BLT, 它将进行语法检查和内连接是否有错的检查。第二部分为 SPLICE1。它根据用户的要求完成实际的模拟任务。第三部分为 SPLOT。它是后处理器, 一旦完成了模拟任务, 由它产生波形输出。如果 SPLOT 不产生任何波形输出, 这说明由于某些错误, SPLICE1 不能够模拟该电路。

SPLICE 可完成 MOS 电路的非线性、时域瞬态分析, 包括对模拟电路 (Analog MOS Circuit) 作严格的瞬态分析, 但不能进行线性交流分析。

以下主要介绍逻辑模拟部分的特点及应用。

SPLICE 逻辑模拟时主要采用事件驱动和选择性跟踪方式。在延迟模型方面采用惯性

延迟模型, 对每个门可规定固定的延迟值, 也可根据逻辑门输出节点上的电容值来确定延迟值。对于动态负载延迟, 目前的逻辑模拟还无法进行 (指 SPLICE 1.7 版本)。

在逻辑模拟时, 除了能得到<0>, <1>和<x>三种信号值外, 还可以规定逻辑强度。当两个或两个以上的门要同时控制同一节点的逻辑状态时就需要由强度来确定该节点的逻辑行为。这种情况在 MOS 逻辑电路中, 由于使用传输门或通导管 (Pass Transistor) 而经常发生。

SPLICE 中隐含的逻辑强度有三级, 即 <Forcing> <Soft> 和 <High impedance>。九态的相互关系见图 4-18。但逻辑强度的级数可以由用户通过 nstrengths 语句加以改变。

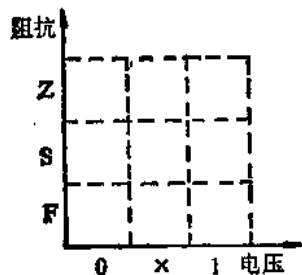


图 4-18 SPLICE 中九态的相互关系

逻辑门的强度可以通过两种办法加以确定。一种是针对某一种工艺规定一组强度, 例如

CMOS 工艺 Forcing 1, Forcing 0, Forcing x

NMOS 工艺 Soft 1, Forcing 0, Soft x

PMOS 工艺 Forcing 1, Soft 0, Soft x

如不加说明, 则认为是 CMOS 工艺。另一种办法是通过模型参数 zstr、ostr 和 xstr 对每个门逐一规定

强度。

zstr 是规定输出为<0>时的强度;

ostr 是规定输出为<1>时的强度;

xstr 是规定输出为<x>时的强度。

在 SPLICE 中规定 <High impedance> 强度为 1。

在逻辑模拟时允许的逻辑元件有:

(1) 标准布尔门: 缓冲门、倒相门、与门、与非门、或门、或非门、异门、同门。

(2) 传输门: NMOS 单向传输门和双向传输门、PMOS 单向传输门和双向传输门、CMOS 单向传输门和双向传输门。

(3) 特殊门: 电阻门 (用于降低节点的强度)、延迟门 (用于增加信号的延迟)。

(4) 逻辑信号源。

输入描述文件的格式有如下规定:

(1) 第一行必须规定程序名为 "SPLICE" (可以是小写 splice)。

(2) 第二行必须是标题行。此标题将在输出时被打印出来。

(3) 最后一行必须以 END 结束。

(4) 在 END 语句前应规定 GO 语句。在读入 GO 语句后执行各种分析, 如未读入则不执行。

(5) 在第二行与 GO 语句之间, 插入元件卡、模型卡和分析控制语句。其顺序是任意的。

(6) 在描述文件的任何地方可插入注释卡。

应注意的是, 在 SPLICE 中每个元件都必须有一对应的模型卡。这包括每个电阻、每个电容和电压源等电学元件。这是与 SPICE 的要求不同的。

分析控制语句有 PLOT, PRINT, TIME 和 WPlot 等。

此外还有一些可选项语句用来改变隐含值。

§ 4.5 SPLICE 使用举例

现有一锁存器电路，其逻辑图示于图 4-19。

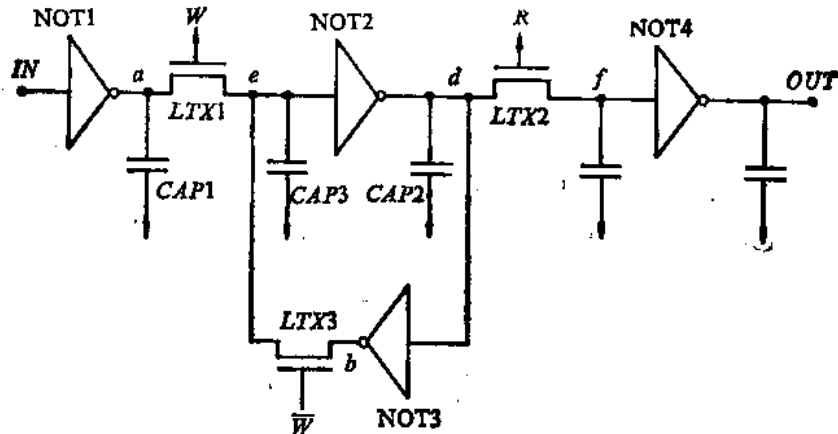


图 4-19 锁存器电路逻辑图

该电路的输入描述文件，见表 4-10。经模拟后所得波形示于图 4-20。

表 4-10 输入描述文件

```
splice
LATCH CELL TEST USING 9-STATE LOGIC SIMULATION
cmos
;instance name      nodes      model name

not1                a,in          inv2
ltx1                e,write,a    ltx1
not2                d, e          inv2
not3                b, d          inv2
ltx3                e,writebar,b  ltx1
ltx2                f,read,d      ltx1
not4                out,f         inv2

;inputs
input1              write         inputA
input2              in            inputB
input3              read          inputC
input4              writebar      inputD

nodecap1            a             c :20f
nodecap2            d             c :20f
nodecap3            e             c :20f

;analysis requests
time 1ns 200ns
;print in,a,write,e,d,read,f,out
wplot in,a,write,e,d,read,f,out
```

```

;model cards
;INPUT LOGIC SOURCES
.model inputA lsrc : v0=0, v1=1, d=0ns,
+
+           p=100ns, 0, 15ns, 20ns, 35ns, 40ns
.model inputB lsrc : v0=1, v1=0, d=0ns,
+
+           p=200ns, 0, 90ns, 95ns, 200ns
.model inputC lsrc : v0=0, v1=1, d=0ns,
+
+           p=100ns 0, 50ns, 55ns, 70ns, 75ns,100ns
.model inputD lsrc : v0=1, v1=0, d=0ns,
+
+           p=100ns, 0, 15ns, 20ns, 35ns, 40ns

;INVERTERS
; tr=risetime, tf=falltime, trc=static risetime factor
;   tfc=static falltime factor, ci=input capacitance,
;   co=output capacitance
.model inv2 inverter : tr=4ns, tf=4ns, trc=30k,
+
+           tfc=20k, ci=100f, co=50f
;UNI-DIRECTIONAL TRANSFER GATE
;   ton=turn on time, toff=turn off time
;   cio=input/output capacitance, cg=gate capacitance
.model ltx1 nmosutxg : ton=2ns, toff=3ns,
+
+           cio=50f, cg=100f

;CAPACITOR MODEL
.model c gcapr
go
end

```

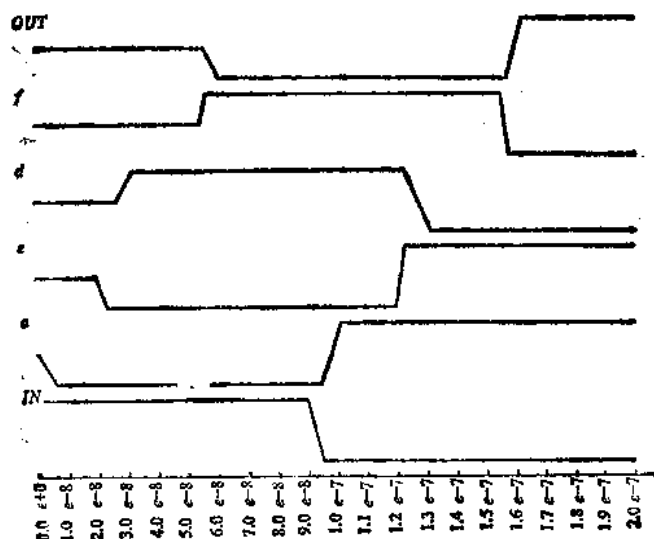


图 4-20 波形图

第五章 时序分析

§ 5.1 时序分析的目的

在电路设计阶段,除了要保证电路的逻辑功能正确之外,还应保证时序容限上的正确性。因此除了通过逻辑模拟进行功能验证,还要通过时序分析程序检查各种时序关系是否满足要求。

时序关系上的限制分为两类:一类是对存储元件的,另一类是对组合电路的。

对存储元件来讲,这种限制包括对脉冲宽度的要求以及信号的建立时间和保持时间。现以 D 触发器为例说明时序上的要求,见图 5-1。

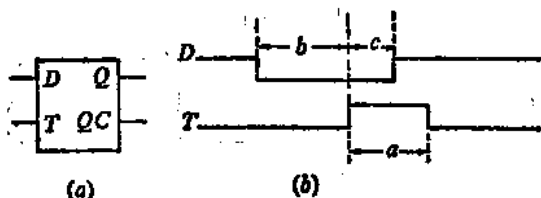


图 5-1 D 触发器及输入讯号的波形

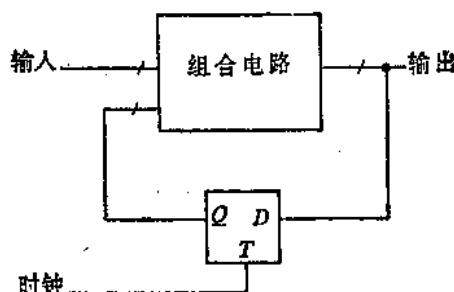


图 5-2 同步时序电路

图(b)中 a 是时钟的最小持续时间,若小于该脉冲宽度,触发器工作将不正常。 b 为信号的建立时间,它是信号送入输入端的时间和时钟脉冲到来时间之间的间隔。建立时间也必须大于某个值。保持时间 c 是时钟信号发生有效的变化之后输入信号必须继续原有状态的时间。同样它也必须大于某个值才能保证存储元件正确的操作。

对于组合逻辑电路的时序要求可以图 5-2 为例。对于这样的典型电路,要正确的工作就必须保证在组合电路中信号的传递时间要小于时钟周期时间。

要进行这样的时序容限分析,光靠逻辑模拟是不充分的。但如果采用通用的电路模拟程序进行分析,需要花费很多计算机机时,特别对于规模很大的电路,所付出的代价过大。这是由于电路模拟时需要进行多次迭代来求解电路联立方程。

因此,人们专门开发了一种着重分析时序关系的分析程序,称为时序(或时域)分析程序。它比通常的门级逻辑模拟的功能要强,它能提供设计所需的详细波形及时序关系。但由于这类程序采取了一些特殊的处理方法和技巧,因而在求解时间上要短得多。例如与通用电路模拟程序 SPICE 相比要快两个数量级。但在求解精度上要比 SPICE 低,大约只能达到 SPICE 求解精度的 90%。

通用电路模拟程序求解精度高,但计算时间长,而时序分析程序求解速度快,但精度低,两者各有特点。因此在实际使用时往往把两者结合起来。对于那些影响整个网络性能的关键性部分或临界通路部分往往采用电路模拟程序进行局部的、精确的模拟。而对整个电

路则采用时序分析, 确定该电路是否会产生时序关系上的错误, 是否能达到预期的速度。

所以在逻辑设计的第一阶段即逻辑验证完成以后, 就要利用有延迟功能的逻辑模拟程序或时序分析程序进行逻辑延迟模拟, 但一般讲后者的分析精度较高。

美国贝尔实验室最早开发了时序分析程序, 称为 MOTIS, 随后美国加州大学伯克利分校又提出了 MOTIS-C 程序。近年来贝尔的 MOTIS 程序有了很大的扩充, 它既可用于逻辑验证, 又能进行版图设计前和版图设计后的时序验证。程序可以进行单位延迟模拟, 多延迟模拟以及混合模拟。MOTIS 不仅用于分析 MOS 电路而且能分析双极型电路。但双极型电路只能用逻辑门来定义, 这些逻辑门的本征延迟存在库中, 但能根据负载电容自动调整延迟。对于版图设计后的时序验证, 是通过版图设计系统把版图中的引线电容提取出来送入 MOTIS 程序, 以得到更精确的模拟。目前 MOTIS 程序已可分析几十万个晶体管的网络。

除了 § 5.2 节中将要讨论的时序分析方法外, 还有其它类型的时序分析技巧, 如通路枚举法、临界通路法等。

§ 5.2 时序分析的方法

时序分析程序的框图如图 5-3 所示。它主要由输入处理器和主模拟器两部分组成。

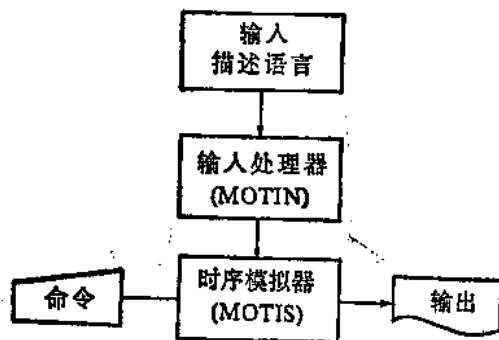


图 5-3 时序分析程序框图

下面以 MOTIS-C 程序为例讨论时序分析的方法及其特点。

它既有通常逻辑模拟程序的特点, 而在具体处理每个逻辑门时又类似于电路模拟程序。这体现在它不列出大的稀疏矩阵, 也不求解联立方程, 而是让信号顺序地从一个门传送到下一个门。另外在器件模型方面作了简化和近似, 对电容也作了限制。为了提高求解速度, 采用查表和计算相结合的办法, 而这些表是事先计算好的。

一、简化的器件模型

通常不考虑二级效应的 MOS 晶体管的电流-电压方程表示如下:

$$I_{DS} = K_F \frac{W}{L} \left\{ \left(V_{GS} - V_{BI} - \frac{V_{DS}}{2} \right) V_{DS} - \frac{2}{3} \gamma \left[(V_{DS} + 2\phi_F - V_{BS})^{3/2} - (2\phi_F - V_{BS})^{3/2} \right] \right\} (1 + \lambda V_{DS}) \quad (5-1)$$

这里 $V_{BI} = V_{T0} - \gamma \sqrt{2\phi_F}$ 。

如果给出了 V_{T0} , K_F , γ , $2\phi_F$, λ 这五个参数, 再给定器件的宽长比 W/L 值, 则 I_{DS} 将取决于端电压 V_{DS} , V_{GS} 和 V_{BS} 。

如对上式加以简化, 在线性区有

$$I_{DS} = K_F \frac{W}{L} \left[\left(V_{GS} - \bar{V}_T - \frac{V_{DS}}{2} \right) V_{DS} \right] (1 + \lambda V_{DS})$$

$$= K_F \frac{W}{L} \left[\left(V_G - \bar{V}_T - \frac{\bar{V}}{2} \right) (V_D - V_S) \right] [1 + \lambda (V_D - V_S)] \quad (5-2)$$

其中

$$\bar{V}_T = V_{BI} + \gamma \sqrt{\bar{V} + 2\phi_F + V_B} \quad (5-3)$$

$$\bar{V} = \frac{V_D + V_S}{2} \quad (5-4)$$

这里 $\bar{V}$ 为 V_D 和 V_S 的平均电压。 $\bar{V}_T$ 为一个平均开启电压, 可认为是源和漏的中点的开启电压。

通过计算可以证明简化方程 (5-2) 的结果与公式 (5-1) 的结果十分接近。

在饱和区有

$$I_{DSAT} = K_F \frac{W}{L} \left[\left(V_G - V_S - \bar{V}_T - \frac{V_{DSAT}}{2} \right) V_{DSAT} \right] [1 + \lambda (V_D - V_S)] \quad (5-5)$$

为了判断 MOS 晶体管是否进入了饱和状态, 需要知道晶体管的夹断电压, 其表达式为

$$V_{DP} = V_G - V_{BI} + \frac{\gamma^2}{2} \left[1 - \sqrt{1 + \frac{4(V_G - V_{BI} - 2\phi_F - V_B)}{\gamma^2}} \right] \quad (5-6)$$

当 $V_D \geq V_{DP}$ 时处于饱和区, $V_D < V_{DP}$ 时为线性区。

在 MOTIS-C 的求解过程中, 需要知道微分导纳

$$Y = \frac{\partial I_{DS}}{\partial V_{DS}} \Big|_{V_{GS}=\text{常数}, V_{BS}=\text{常数}} \quad (5-7)$$

现将线性区公式改写为

$$I_{DS} = I_{DS0} (1 + \lambda V_{DS})$$

其中

$$I_{DS0} = K_F \frac{W}{L} \left[\left(V_{GS} - \bar{V}_T - \frac{V_{DS}}{2} \right) V_{DS} \right] \quad (5-8)$$

这时求得微分导纳 Y 为

$$Y = \frac{\partial I_{DS0}}{\partial V_{DS}} (1 + \lambda V_{DS}) + I_{DS0} \lambda \quad (5-9)$$

但公式 (5-9) 得到的 Y 值与从公式 (5-1) 所求得的 Y 值并不一致, 其误差主要是由 $\frac{\partial I_{DS0}}{\partial V_{DS}}$ 项引起的。因此将式 5-8 中的 $\bar{V}_T$ 用 V_{TD} 代替。 V_{TD} 由下式表示:

$$V_{TD} = V_{BI} + \gamma \sqrt{\bar{V}_D + 2\phi_F + V_B} \quad (5-10)$$

这样 Y 值就修正为

$$\begin{aligned} Y = & K_F \frac{W}{L} (V_{GS} - V_{TD} - V_{DS}) (1 + \lambda V_{DS}) \\ & + K_F \frac{W}{L} \left[\left(V_{GS} - \bar{V}_T - \frac{V_{DS}}{2} \right) V_{DS} \right] \lambda \end{aligned} \quad (5-11)$$

饱和区的微分导纳为

$$Y = K_F \frac{W}{L} \left[\left(V_{GS} - \bar{V}_T - \frac{V_{DSAT}}{2} \right) V_{DSAT} \right] \lambda \quad (5-12)$$

可以看出, 经过修正后的 Y 值, 从非饱和过渡到饱和时将是连续的。

二、造表与查表

为了减少计算工作量，在 MOTIS-C 程序中采用了查表方式。

为此需要事先造好表格，现有两种表格，一是夹断电压随 V_G 的变化表格，另一是平均开启电压随 $\bar{V}$ 的变化表格。

从公式 (5-6) 看出，由于 V_{BT} 、 γ 和 $2\phi_F$ 已知， V_B 为给定的常数，所以 V_{DP} 只是栅电压 V_G 的函数，因而可以建造一个一维的表格。

在夹断电压随 V_G 变化表格中，令 V_G 从零伏变化到 $2V_+$ 伏 (V_+ 为电源电压，其隐含值为 5 伏)。 V_G 的增量 $\Delta V_G = \frac{2V_+}{63}$ ，因而 V_G 共有 64 个点。表格的形式如表 5-1，有

$V_{DP}(I)$ 各值， $I=1, 2, 3, \dots, 64$ 。

表 5-1 V_{DP} 随 V_G 变化表

$V_G (1)$	$V_G (2)$		$V_G (64)$
$V_{DP} (1)$	$V_{DP} (2)$		$V_{DP} (64)$

对于 $\bar{V}_T$ 来讲，同样可以建造一个一维表格，即 $\bar{V}_T$ 与 $\bar{V}$ 的关系。

在平均开启电压随 $\bar{V}$ 变化的表格中，也令 $\bar{V}$ 从零伏变化到 $2V_+$ 伏。 $\bar{V}$ 的增量 $\Delta \bar{V} = \frac{2V_+}{63}$ ，因而 $\bar{V}$ 共有 64 个点，这样得到 $\bar{V}_T(I)$ 值， $I=1, 2, 3, \dots, 64$ 。其表格形式与表

5-1 一样。

从以上讨论可知，只要一旦给定了 V_{T0} 、 γ 、 $2\phi_F$ 和 V_B ，就很容易造出这两种表格。

在 MOTIS-C 的输入语句中有表格产生语句 TABLE。可以通过改变语句中的 V_{T0} 、 K_F 、 γ 、 $2\phi_F$ 、 λ 值而产生一个新的表格，即新的 $V_{DP}(I)$ 和 $\bar{V}_T(I)$ 值。如不加以说明，程序将调用旧表格。

有了这两种表格，就减少了计算工作量。在计算 I_{DS} 和 Y 时，程序根据给定的 V_G 和 $\bar{V}$ (即 $(V_D + V_S)/2$) 值在表中查找相应的 V_{DP} 和 $\bar{V}_T$ 。接着将查表所得的 V_{DP} 和 $\bar{V}_T$ 代入公式 (5-2)，(5-11)，(5-12)，求得电流和微分导纳值。但在求线性区微分导纳时，还要求一次 V_{TD} 。可以通过查找同一个 $\bar{V}_T$ 表得到 V_{TD} ，所不同的只是将 $\bar{V}$ 用 V_D 值代入而已。

造表过程在 READIN 子程序中完成。查表时根据晶体管的类型是驱动管、负载管或传输门分别调用子程序 MOST、MOSL 和 MOS2。

MOST 的程序框图示于图 5-4。

在实际查表时，给定的 V_G 和 $\bar{V}$ 并不正好等于表中所有的 $V_G(I)$ 和 $\bar{V}(I)$ ，因此只能寻找最接近于给定值的 $V_G(I)$ 和 $\bar{V}(I)$ 值。为此 I 值可由下列公式求得

$$I = \min \left(\max \left(\text{INT} \left(V_G \frac{1}{\Delta V_G} \right) + 1, 1 \right), 64 \right) \quad (5-13)$$

采用这种方式求得的 V_D 和 $\bar{V}_T$ 存在着一定的误差。

三、宏元件模型

除了 NMOS 晶体管模型和 PMOS 晶体管模型外, MOTIS-C 有专门的语句可以描述一系列宏元件。

现有程序可以处理的宏元件有任意输入端的与非门、任意输入端的或非门、任意输入端的与或非门、任意输入端的或与或非门、任意输入端的 CMOS 与非门、任意输入端的 CMOS 或非门、CMOS 传输门、具有 Q 和 $\bar{Q}$ 端, 以及时钟控制的 D 型触发器。

对于图 5-5 中由两个驱动管并联所组成的二输入端或非门, 程序通过两次调用 MOST 分别求出 T_{D1} 和 T_{D2} 的 I_{D1} , I_{D2} 以及 Y_{D1} , Y_{D2} 。再通过 MOSL 求得负载管的 I_L 和 Y_L 。对于 n 个输入端的或非门则对 MOST 进行 n 次调用。

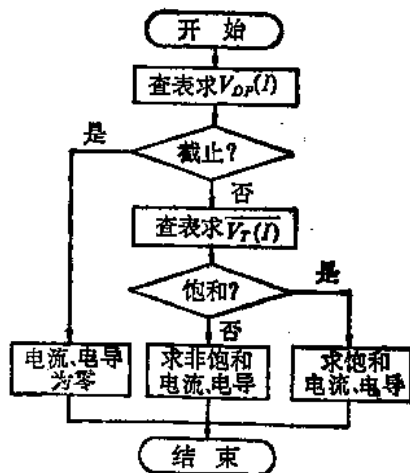


图 5-4 MOST 子程序框图

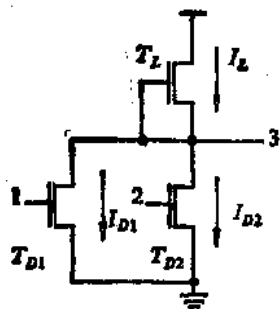


图 5-5 二输入端或非门

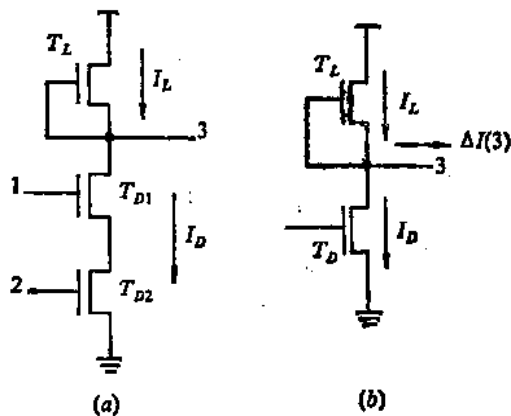


图 5-6 二输入端与非门及等效门

对于图 5-6(a) 中由两个驱动管串联所组成的二输入端与非门, 则要作简化处理。子程序 AND12 将两个串联的驱动管 T_{D1} 和 T_{D2} 等效成一个 T_D 管, 见图 5-6(b)。

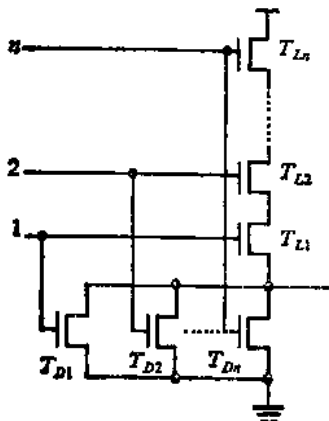


图 5-7 CMOS 或非门

此等效管的宽长比等于原 T_{D1} 和 T_{D2} 的宽长比, 并把 V_G 选取为 V_{G1} 和 V_{G2} 中的较小值。然后调用子程序 MOST 求得 I_D 和 Y_D 。这种等效方法会带来误差。对于多输入端的与非门, 按同样的原则处理。

对于图 5-7 那样的 CMOS 或非门也按照上述原则加以处理。由于 T_{D1} , T_{D2} , ..., T_{Dn} 均为 NMOS 驱动管并彼此并联, 因此通过循环调用 MOST 子程序。调用 MOST 时, 各管的 V_{DS} 相同, 但各管的 V_{GS} 不同。 T_{L1} , T_{L2} , ..., T_{Ln} 均为 PMOS 管且彼此串联, 这时将 n 个负载管等效成一个 PMOS 管, 其宽长比等于各个负载管的宽长比, 而 V_{GS} 取各管中的最小值。

然后调用MOS2子程序求 I_L 和 Y_L 。

四、求解的基本方程

前面谈到时序分析程序的特点是不求解电路的联立方程，而是认为信号顺序地从一个门传送到下一个门。而且在求解门的节点电压时把该节点看作是孤立节点。

求解节点时的基本方程为

$$V(N) = V_r(N) + I(N)/Y(N) \quad (5-14)$$

式中， N 为节点号， $V(N)$ 为当前时刻的节点电压， $V_r(N)$ 为前一时刻的节点电压， $I(N)$ 为当前时刻流入和流出节点 N 的所有电流的代数和， $Y(N)$ 为当前时刻节点 N 的微分导纳。

现以图5-6的与非门为例求出输出节点的电流、微分导纳及节点电压。

先求得负载管 T_L 的电流 I_L ，再求得等效驱动管的电流 I_D ，这样节点3的当前时刻电流为前一时刻输出节点电流与负载管电流、驱动管电流的代数和，即

$$I(3) = I_r(3) - I_D + I_L \quad (5-15)$$

这里规定流入节点的电流为正，流出节点的电流为负。

微分导纳则是前一时刻输出节点导纳与负载管电导、驱动管电导之和，即

$$Y(3) = Y_r(3) + G_L + G_D \quad (5-16)$$

最后得到输出节点的当前电压为

$$V(3) = V_r(3) + I(3)/Y(3) \quad (5-17)$$

五、传输门的近似处理

对于传输门，我们可以忽略栅电压 V_G 的微小变化对输入、输出端点的影响，但对于输入与输出两个节点电压之间的相互影响则不能忽略。为了将输入、输出两端仍看成是孤立的，在此采取某种近似。

对图5-8这样的传输门，并已知 $V(N_G)$ ， $V(N_D)$ 和 $V(N_S)$ ，调用MOS2求出当前时刻传输门的电流 I_T 及电导 G_T ，并得到下列电流公式

$$I(N_S) = I_r(N_S) + I_T + G_T \Delta V_D \quad (5-18)$$

$$I(N_D) = I_r(N_D) - I_T + G_T \Delta V_S \quad (5-19)$$

但我们并不知道当前时刻的 ΔV_D 和 ΔV_S ，这里采用一种近似，用上一时刻的、已知的 δV_D 和 δV_S 代入。

δV_D 和 δV_S 分别为

$$\delta V_D = V(N_D) - V_r(N_D) \quad (5-20)$$

$$\delta V_S = V(N_S) - V_r(N_S) \quad (5-21)$$

因有

$$I(N_S) = I_r(N_S) + I_T + G_T [V(N_D) - V_r(N_D)] \quad (5-22)$$

$$I(N_D) = I_r(N_D) - I_T + G_T [V(N_S) - V_r(N_S)] \quad (5-23)$$

导纳公式则分别为

$$Y(N_S) = Y_r(N_S) + G_T \quad (5-24)$$

$$Y(N_D) = Y_r(N_D) + G_T \quad (5-25)$$

六、电容的处理

MOTIS-C程序中把电容处理成一个电导与一个电流源的并联，示于图5-9。

电容电流的公式有

$$I = C \frac{dV}{dt} = \frac{C}{\Delta t} [V(N) - V_r(N)] \quad (5-26)$$

但是在 Δt 时间内, 电容 C 的充放电速度并不相同, 因而 I 是变化的。为简化起见取平均值, 即

$$\bar{I} = \frac{I_{n+1} + I_n}{2} \quad (5-27)$$

将此式代入 (5-26) 式得

$$\frac{I_{n+1} + I_n}{2} = \frac{C}{\Delta t} [V(N) - V_r(N)] \quad (5-28)$$

移项后有

$$\begin{aligned} I_{n+1} &= \frac{2C}{\Delta t} [V(N) - V_r(N)] - I_n \\ &= Y_c(N) [V(N) - V_r(N)] - I_n \end{aligned} \quad (5-29)$$

这里导纳值

$$Y_c(N) = \frac{2C}{\Delta t} \quad (5-30)$$

七、瞬态分析

瞬态分析时从 $RTIME=0$ 一直进行到用户所规定的 T_{stop} 为止。RTIME 根据 DELT 而增加, 即

$$RTIME = RTIME + DELT \quad (5-31)$$

RTIME 的初始值设置为零。

在程序中规定了一个内部时间步长 DELT,

$$DELT = C_{min}(V_+ - V_-)/I_{max} \quad (5-32)$$

式中, C_{min} 为该节点的集中电容;

I_{max} 为流经晶体管的最大电流 (一般取 $V_{GS}=2V_+$, $V_D=V_{DSAT}$ 时的电流);

V_+ 为电源电压 V_{DD} ;

V_- 为电源电压 V_{SS} 。

用户可以通过 TIME 语句中的 C_{min} 值来调整这个步长, 即

$$DELT = C_{min} \cdot C_{min}(V_+ - V_-)/I_{max} \quad (5-33)$$

改变 C_{min} 值可增加或减小内部时间步长, 其隐含值为 1。

先求出 $RTIME=RTIME+DELT$ 时刻的输入电压值, 并处理电容求出其电流和电导值。再根据前述方法, 对不同的宏元件类型调用相应的子程序 ANDI2, ORI2, ANDOR, ORAND, TRSFR, LATCH 等求出输出节点的瞬态电流和微分导纳。

然后代入基本方程 (5-14) 式求得输出节点的新的电压值。

如电路中有悬浮电容, 则调用子程序 FCAP 进行处理。

接着根据电路中最大电压变化 ΔV 的值对时间步长进行调整。

如果 ΔV 大于规定的最大电压变化 ΔV_{max} (ΔV_{max} 的值为 $V_+/64$), 这时就要减小内部步长, 令 $DELT=DELT/4$, 并重新调用各相应的子程序求解。如求得的电压变化 ΔV

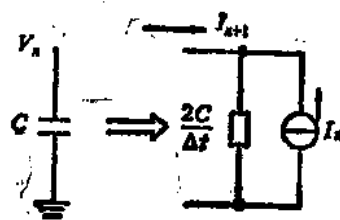


图 5-9 电容的等效

仍大于 $\Delta V_{\max}$ ，则再次令 $\text{DELT} = \text{DELT}/4$ 并重新求解，如此继续直到 $\Delta V \leq \Delta V_{\max}$ 满足时为止。

如果 ΔV 小于规定的最小电压变化 $\Delta V_{\min}$ ($\Delta V_{\min}$ 的值为 $V_t/256$)，这时就加大内部步长，令 $\text{DELT} = 2\text{DELT}$ ，并重新求解。

如求得的电压变化 ΔV 仍然不能满足 $\Delta V \geq \Delta V_{\min}$ 的要求，则再次加大 DELT ，如此反复直到满足时为止。

所以要求 $\Delta V_{\min} < \Delta V < \Delta V_{\max}$ ，是为了得到较精确的解。

瞬态分析一直进行到 $\text{RTIME} > T_{\text{stop}}$ 时结束，并记录运行时间，打印各项结果，绘制波形图。瞬态分析由子程序 TRAN 完成，其框图示于图 5-10。

贝尔实验室所开发的时序分析程序则采用固定时间步长的方式。时间增量取为 1ns 。所以采用这样大小的时间增量，是因为它比典型的过渡时间要小，因而可以求得详细的变化过程。但在满足上述要求的同时又应使时间增量尽可能大，这样可避免过长的运行时间。

另外一点不同之处是在求输出电压时采用以下积分公式

$$V(t_2) = V(t_1) + \frac{1}{C_L} \int_{t_1}^{t_2} (I_D - I_L) dt \quad (5-34)$$

式中， t_1 为起始时间； t_2 为终止时间； C_L 是负载电容； I_L 为负载晶体管电流； I_D 为等效驱动管电流。积分时采用改进的后向欧拉法，以得到稳定的解。然后将求得的解（输出电压）存入一数据区。在下一步长时，其它门将从该数据区中取出数据作为输入电压值。

§ 5.3 MOTIS-C 时序分析程序及其应用

MOTIS-C 的输入描述语言与 SPLICE 类似。它以标题行开始，以 END 行结束。标题行以后可以有元件描述行、模型描述行、全程变量行、分析控制行、表格产生命令行。

一、在 MOTIS-C 中具有模型类型有：

(1) 晶体管模型

TRANS 用于 NMOS 增强型晶体管；

TRAN2 用于 NMOS 耗尽型或 PMOS 增强型晶体管；

TRANC 用于 CMOS 晶体管；

(2) 宏元件模型

NAND2 多输入端与非门;

NOR2 多输入端或非门;

ANDOI 多输入端与或非门;

ORANI 多输入端或与或非门;

(3) 触发器模型

LATCH 具有 Q 和 $\overline{Q}$ 端和时钟控制的 D 型触发器;

(4) 电容模型

CAPCR 接地或悬浮电容。悬浮电容可连接在电源节点与一个非电源节点之间,但不能两端都连接电源节点。

(5) 电压(时钟)源模型 SOURCE。见图 5-11。图中, V_0 , V_1 为高低电平; t_0 , t_r , t_1 , t_f 分别为低电平、上升、高电平和下降时间。 dl 为第一时钟周期开始前的延迟时间。

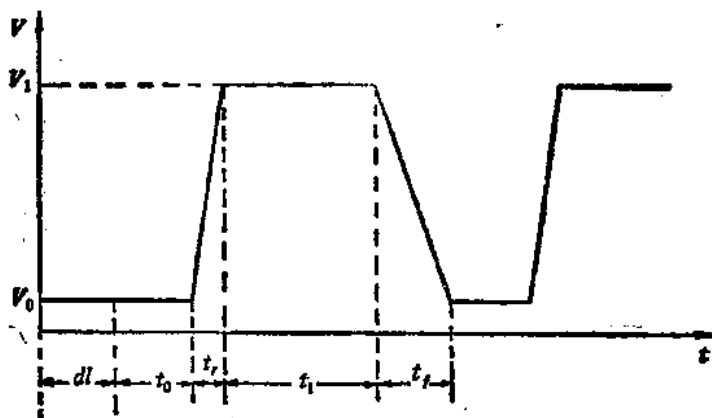


图 5-11

分析控制语句有 PLOT, TIME, PRESET 等。

二、全程变量

V^+ 是宏模型漏端或负载管的电源电压。它必须是正值,隐含值 5 伏。 V_{BG} 是背栅电压。对 NMOS 管该值为正。

三、表格产生语句为

TABLE (type) (V_{T0} K_F Gamma Phi Lambda) (pmos/nmos) 可以通过此语句产生新表,只要改变各参数值就行。各参数的含义、单位及隐含值见表 5-2。该语句中的 type 用来规定表的类型。其类型有

(1) DRIVE 表用于所有的 NMOS 增强型晶体管即 TRANS 管以及宏模型中的驱动管。

(2) GATE2 表用于所有的 PMOS 增强型晶体管或 NMOS 耗尽型晶体管,即 TRAN2 类型器件。

(3) LOAD 表用于宏模型中的负载晶体管,它只模拟栅源相连的耗尽型晶体管。

在产生 GATE2 表时,要用到 pmos/nmos 这个参数。其隐含值为 nmos。如果赋予此

表 5-2 模型参数及隐含值

参 数	含 义	单 位	隐 含 值		
			NMOS	PMOS	负载管
V_{th}	零偏置下开启电压	V	0.5	-3	-3
K_p	本征跨导	$\mu A/V^2$	20	20	20
Gamma	体 阈 值	$V^{1/2}$	0.7	0.7	0.7
Phi	表 面 势	V	0.6	0.6	0.6
Lambda	沟道长度调制	1/V	0.0	0.0	0.0

项任何大于零的数，则所有 TRAN2 类型的器件将成为 PMOS 器件。对 DRIVE 表和 LOAD 表，由于这类表中只有 NMOS 器件，因此不必改动此参数而取隐含值。

由于 MOTIS-C 程序并没有采用动态存储分配，因此被分析电路中各类型的器件数有一定的限制，在使用时应注意这点。

§ 5.4 MOTIS-C 使用举例

现有二级倒相器电路图示于图 5-12。输入波形图示于图 5-13。该电路的输入描述文件见表 5-3。经模拟所得波形示于图 5-14。

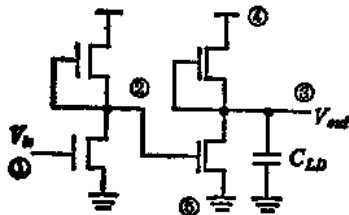


图 5-12 二级倒相器电路图

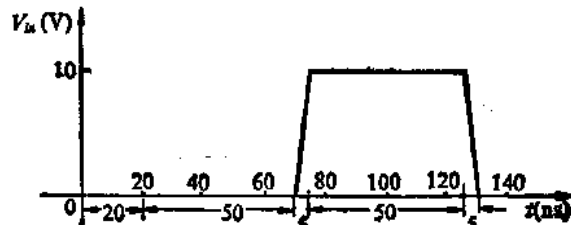


图 5-13 输入波形图

表 5-3

```

SIMPLE INVERTER CASCADE
*
* SOURCE MODEL
MODEL INP SOURCE (10 0 50N 5N 50N 5N 20N)
MODEDL TIE SOURCE (10 0 1M 0 1M 0)
V+=10
    
```

```

* CAPACITOR MODEL
MODEL CAP CAPCR 1PF
* DEVICE MODELS AND MACROMODELS
MODEL INVI ANDOI (1 1 0.5 30F 30F 30F 1 0)
MODEL EDRV TRANS (1 10F 30F 10F)
MODEL DLOD TRANS2 (0.5 10F 30F 10F)
*
* CIRCUIT DESCRIPTION
* INVERTER USING MACROMODEL (FIRST STAGE)
I1 1 2 INVI
*
* INVERTER BUILT FROM DISCRETE DEVICES (SECOND STAGE)
EDR 3 2 5 EDRV
DLD 4 3 3 DLOD
VDD 4 0 TIE 1
GND 5 0 TIE 0
*
* VOLTAGE INPUT AND LOAD CAPACITOR
VIN 1 0 INP 0 1
CLD 3 0 CAP
*
* ANALYSIS REQUESTS
TIME 200NS 2NS
PLOT 3 2 1 5 4
*
* TABLE GENERATION VTO KP GAMMA PHI LAMBDA
TABLE DRIVE 0.8 20U 0.7 0.6 0.0
TABLE LOAD -3 20U 0.7 0.6 0.0
TABLE GATE2 -3 20U 0.7 0.6 0.0
PRESET (1,0) (2,10) (3,5) (4,10) (5,0)
END

```

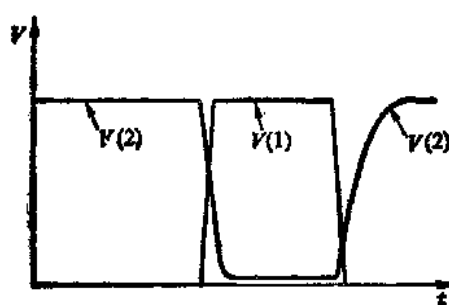


图 5-14 模拟得到的波形图

第六章 电路模拟

§ 6.1 电路分析的作用

在逻辑设计完成之后,需要进行电路设计。电路设计的任务是根据所要求的电路性能,例如速度、功耗、电源电压、逻辑操作类型,信号电平的容限等确定电路的结构和各元器件的参数;同时应考虑工艺上可能发生的偏差与使用温度上的变化等,使所设计的电路仍能达到规定的性能。到目前为止,利用CAD方法进行全自动的电路设计在实际上还有困难,实际情况往往是设计者根据设计指标,提出电路框图,进行电路结构的设计和元器件参数的初步确定,然后利用电路模拟程序对该电路进行模拟分析,再根据分析结果进行修改,经过多次反复,最后得到符合要求的电路。

传统的验证方法是采用“实验装置”法,即根据电路搭成一实验板,使用仪器仪表作实地测试以检验该电路是否符合要求。但随着规模的扩大、集成度的提高,已无法在实验装置上搭接这样的电路,即使能搭接成功的话,也与实际的集成电路差别太大,已失去其物理验证的意义。此外,传统方法也无法进行容差分析和极限条件的检验。

随着计算机技术和计算方法的发展,就可能将要分析的电路问题列出数学形式的电路方程,然后对电路方程求解。这时不需要任何实际的元器件和各种调试工具,而且可以改变各种条件进行分析模拟甚至可以进行破坏性的模拟。由于计算机运算速度快、精度高,因此有可能采用更复杂的、更精确的电路模型(如考虑二级效应和寄生效应等)以提高设计精度,而在传统的解析法中,往往只能采用较简单的数学表达式。

电路分析的精确度不仅与器件模型本身有关,还与给定的器件模型参数的正确性有密切的关系。为此在电路设计之前,需要从所选择的工艺方案中得到某些模型参数值(如薄层电阻值、氧化层厚度、单位面积结电容等),或者对同一类工艺的实际电路进行测量,从测量值中推算出这些模型参数值。

电路分析除了在版图设计以前进行外,有的还在版图设计以后进行,即所谓的“后”模拟。这时的目的是把实际版图中所引入的寄生效应考虑进去,以检验版图设计前后电路性能上的差异。

电路的计算机辅助分析就其内容上讲可分成两个方面,一是电路模拟,二是电路优化。电路优化是在指定的性能指标及电路的结构条件下,确定电路中元器件参数的最佳值。

§ 6.2 SPICE 电路模拟程序及其应用

自1973年以来国外先后开发了各种电路模拟程序如ADVICE、ASTAP、CIRCUS2、SLIC、SPICE等,其中以SPICE2最为著名并得到广泛应用。在1986年美国加州大学伯克利分校又推出SPICE3版本。它采用C语言编写。在被分析的元件中增加了开关器(电压控制开关和电流控制开关),在半导体器件中增加了GaAs场效应管(MESFET),

在 MOS 场效应管中增加了短沟 BSIM 模型（或称 MOS4 模型）。此外还增加了均匀分布的 RC 线模型、半导体电阻电容模型。SPICE3 与 SPICE2 相比在数据结构方面有较大改动，采用层次式模块化结构，特别突出模块的独立性。但在求解方法上无根本的差别。

电路模拟程序 SPICE 可对电路进行非线性直流分析、非线性瞬态分析和线性交流分析。

被分析电路中的元器件可包括电阻、电容、电感、互感、独立电压源、独立电流源、四种受控源（线性电压控制电流源、线性电压控制电压源、线性电流控制电流源、线性电流控制电压源）、传输线以及最通用的半导体器件即二极管、双极型晶体管、结型场效应管和 MOS 场效应管，还包括 SPICE3 版本中增加的开关器、RC 线、半导体电阻电容和 GaAs 场效应管。

直流分析用来决定电路的直流工作点。这时电路中的电感短路，而电容开路，在进行瞬态分析之前，SPICE 程序自动地先进行直流分析，用来决定瞬态的初始条件。同样，在交流小信号分析之前也先自动地进行直流分析，以便决定非线性器件的线性化小信号模型参数。在直流分析中还可以计算传输函数（输出变量与输入源之比）的直流小信号值，以及输入电阻和输出电阻。还可以用直流分析产生直流转移曲线，即直流输出变量随某个独立电压源或电流源而变化的曲线。此外，SPICE 还能求出指定输出变量对于电路参数的直流小信号灵敏度。

瞬态分析是在用户规定的时间间隔内计算输出变量作为时间的函数。对于大信号正弦模拟，可以对输出波形进行傅里叶分析，从而得到频率域的傅里叶系数。

交流小信号分析用来计算交流输出变量作为频率的函数。程序先计算直流工作点，并确定电路中所有非线性器件的线性化小信号模型参数，然后在用户指定的频率范围内对这个线性化的电路进行分析。

电阻和半导体器件产生的白噪声，也可用交流小信号分析部分来模拟。对这种分析不需要其它附加的输入数据。若要对闪烁噪声源进行模拟，就要在相应的器件模型卡中设置闪烁噪声系数 K_f 和闪烁噪声指数 A_f ，这两个参数值或采用程序内部隐含值。

电路的失真特性也可以作为交流小信号分析的一部分加以模拟。这种分析是在假定把一个或两个信号频率加在输入端的情况下完成的。

SPICE 还允许分析不同温度时的电路特性。最低的温度是 -223°C 。如不加以说明，温度的隐含值为 27°C (300K)。

可以说 SPICE 包括了电路模拟所要求的基本内容，但它不能进行容差分析，即不能计算由于元器件的容差造成电路性能参数上的统计分布。

SPICE 的输入格式是一种自由格式。电路的输入描述文件的格式如图 6-1。

输入的第一张卡片必须是标题卡，而且不能省略。最后一张卡片必须是结束卡。其余卡片的顺序是任意的。在输入描述文件的任何地方可插入注释语句，程序对注释语句

标 题 卡	
元 件 卡	
模 型 卡	
子电路	.SUBCKT 卡
	元件卡、模型卡
	子电路调用
	.ENDS 卡
控 制 卡	
结 束 卡	

图 6-1 输入描述文件格式

只作原样打印而不作处理。SPICE 输入描述文件中允许进行子电路定义(用一组元件卡来定义),程序会自动将这组元器件插入到子电路被调用的地方。子电路的大小和复杂性没有限制,而且允许子电路中还包含其它子电路。

元件卡是说明该元件的拓扑关系和元件值的。每个元器件给予一个元器件名,元器件名的第一个字母说明该元器件的类型,SPICE中对各种类型元器件所对应的英文字母已有规定。元器件名不能重复。元器件的节点号用一正整数表示,节点号可以不连续,但不能相重,接地点规定为零节点。

模型卡是用来说明元器件的模型参数的。由于半导体器件各有很多参数,而且一个电路中许多器件常常采用同一种参数值,因而 SPICE 中设计了模型卡。在模型卡中定义一组器件模型参数并赋予一个唯一的模型名,在元器件卡中就可引用此模型名,表明此元器件具有这一组模型参数值,这样就不需要在每个元件卡上指定所有的模型参数值。

控制卡是控制程序的运行和规定分析及输出的内容。如温度卡、工作点卡、交流卡、瞬态分析卡、打印卡、绘图卡以及可选项卡等。

§ 6.3 SPICE 使用举例

现有触发器电路示于图 6-2。该电路的输入描述文件见表 6-1。经模拟后得到的输出结果示于图 6-3。

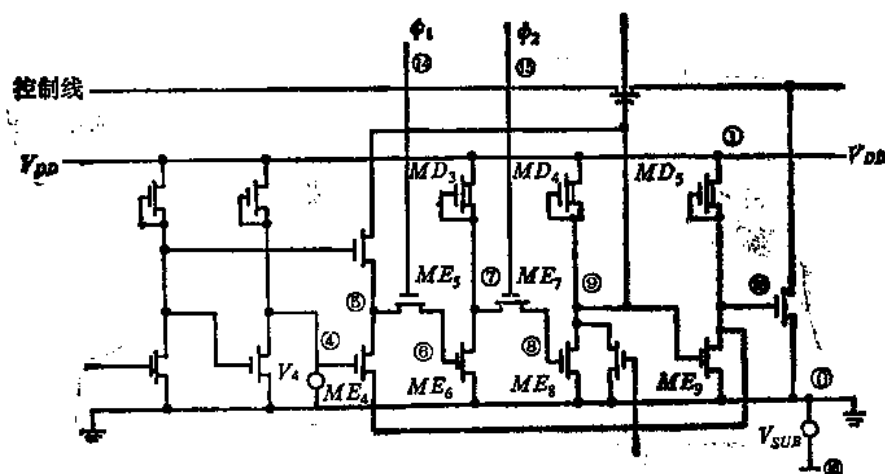


图 6-2 触发器电路图

表 6-1 输入描述文件

```

*TOGGLE CELL
*
*INPUT LISTING
*
VDD 1 0 DC 5V
VBB 16 0 DC -2.5V
V1 14 0 PULSE 0V 5V 20NS 10NS 10NS 55NS 160NS
V2 15 0 PULSE 0V 5V 100NS 10NS 10NS 55NS 160NS
VN4 4 0 DC 5V
    
```

```

ME6 7 6 0 16 MENH1 L=5U W=10U
MD3 1 7 7 16 MDEP1 L=25U W=5U
ME8 9 8 0 16 MENH1 L=5U W=10U
MD4 1 9 9 16 MDEP1 L=25U W=5U
ME9 10 9 0 16 MENH1 L=5U W=10U
MD5 1 10 10 16 MDEP1 L=25U W=5U
ME4 10 4 5 16 MENH1 L=5U W=5U
ME5 5 14 6 16 MENH1 L=5U W=5U
ME7 7 15 8 16 MENH1 L=5U W=5U
*
CN5 5 0 0.03P
CN6 6 0 0.05P
CN7 7 0 0.04P
CN8 8 0 0.06P
CN9 9 0 0.18P
CN10 10 0 0.18P
*
.TRAN 5NS 500NS
.PLOT TRAN V(14) V(15) V(9) V(10) (0,5)
.WIDTH OUT =90
*
***** ENHANCEMENT LEVEL 1 *****
*
.MODEL MENH1 NMOS (LEVEL=1 VTO=0.75 KP=15E-6
+GAMMA=0.46 CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4
+CJSW=1.0E-9 JS=1.0E-7 LD=1.25E-6 XQC=0.4)
*
***** DEPLETION LEVEL1 *****
*
.MODEL MDEP1 NMOS (LEVEL=1 VTO=-4.7 KP=12E-6
+GAMMA=0.7 CGSO=4.5E-10 CGDO=4.5E-10 CJ=1.0E-4
+CJSW=1.0E-9 JS=1.0E-7 LD=1.25E-6 XQC=0.4)
*
.END

```

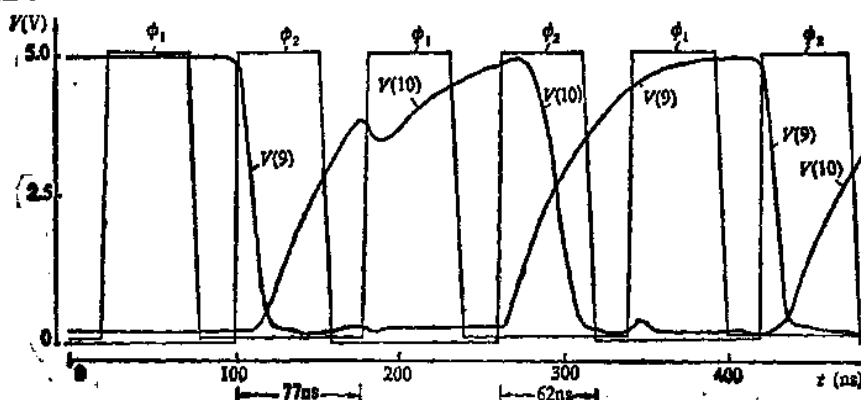


图 6-3 模拟结果

§ 6.4 器件模型及各参数的意义

对于允许出现的各种元器件，在程序中必须有相应的数学模型来描述它们，也就是把元器件用计算机能够进行计算的公式表示出来。由于器件模型实际上是器件电学特性的一种近似，因此采用何种模型和参数在很大程度上决定了分析的精度和速度。

一般来讲，复杂的模型可以得到高精度的解，但它需要较多的内存容量，而且导致计算时间的增加。但模型太粗糙会导致分析结果的不可靠。因此所用模型的复杂程度要根据实际需要而定，在满足一定的计算精度条件下，应尽可能采用较简单的模型。合理的器件模型应遵循以下原则：（1）能正确反映器件的电学性能；（2）适宜于在计算机上作数值求解；（3）模型中各个参数与器件的工艺设计有明确的对应关系，或者能以某种方法测量而得。

目前构成器件模型的方法有两种：一种以器件的工作原理为基础，从器件的数学方程式出发；另一种把器件看成“黑盒”，从测量其端点的工作特性出发形成模型。

下面讨论 SPICE 中的各个器件模型，它们以上述的前一种方法为基础。

6.4.1 二极管模型

该模型示于图 6-4。它既适合于结型二极管，也适用于肖特基势垒二极管。

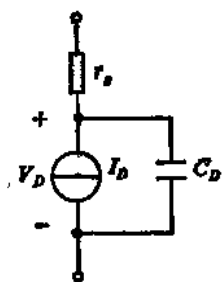


图 6-4 二极管模型

直流时，它由一个等效体电阻 r_s 与一个非线性电流源 I_D 相串联来表示。非线性电流的表达式为

$$I_D = I_s \left\{ \exp\left(\frac{V_D}{nV_t}\right) - 1 \right\} \quad (6-1)$$

式中， $V_t = \frac{kT}{q}$ ， I_s 为饱和电流， n 为发射系数。

二极管的电荷存储效应由两部分组成。一部分是结耗尽层中的电荷，另一部分是注入载流子形成的电荷存储。电荷 Q_D 的表达式为

$$Q_D = C_{j0} \int_0^{V_D} \left[1 - \frac{V}{\phi_B} \right]^{-m} dV + \tau_s I_s \left[\exp\left(\frac{V_D}{nV_t}\right) - 1 \right] \quad (6-2)$$

因而结电容

$$C_D = \frac{\partial Q_D}{\partial V_D} = C_{j0} \left[1 - \frac{V}{\phi_B} \right]^{-m} + \frac{\tau_s I_s}{nV_t} \exp\left(\frac{V_D}{nV_t}\right) \quad (6-3)$$

式中， C_{j0} 为零偏压时结耗尽层电容， ϕ_B 为结电势， m 为梯度因子， τ_s 为渡越时间。当结的正向偏压较大时，上述结电容的表示式有很大偏差，因而在 $V_D > FC \phi_B$ 时有下式，即

$$C_D = \frac{C_{j0}}{(1-FC)^m} \left[1 + \frac{m}{\phi_B(1-FC)} (V_D - FC \phi_B) \right] \quad (6-4)$$

式中， FC 为正偏耗尽层电容公式中的系数。

交流时，二极管的小信号线性化模型示于图 6-5。图中

$$g_D = \left. \frac{\partial I_D}{\partial V_D} \right|_{\text{工作点}} = \frac{I_s}{nV_t} \exp\left(\frac{V_D}{nV_t}\right) \bigg|_{\text{工作点}} \quad (6-5)$$

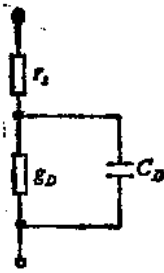


图 6-5 二极管小信号线性化模型

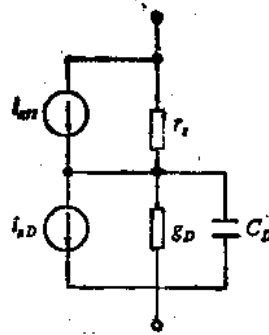


图 6-6 二极管噪声模型

二极管在交流分析时的噪声模型示于图 6-6。其中由电阻 r_s 产生的热噪声为

$$i_{nr} = \sqrt{\frac{4kT}{r_s} \Delta f} \quad (6-6)$$

二极管的散粒噪声和闪烁噪声为

$$i_{nD} = \sqrt{2qI_D \Delta f + \frac{K_f I_D^{A_f}}{f}} \quad (6-7)$$

式中, f 为频率; Δf 为频带宽度; K_f 为闪烁噪声系数; A_f 为闪烁噪声指数。

在二极管模型中考虑了结饱和电流随温度的变化, 于是有

$$I_s(T) = I_s(T_{nom}) \left(\frac{T}{T_{nom}} \right)^{XTI} \exp \left[\frac{qE_g}{kT} \left(\frac{T}{T_{nom}} - 1 \right) \right] \quad (6-8)$$

式中, E_g 为禁带宽度; XTI 为饱和电流的温度系数; T_{nom} 为标称温度, 隐含值 27°C (300K)。

二极管的反向击穿用反向二极管电流的指数增长来模拟。由反向击穿电压 V_{br} 和反向击穿时电流 I_{br} 两参数决定。

因此二极管的模型参数共有十四个, 即: I_s , r_s , n , τ_t , C_{j0} , ϕ_B , m , E_g , XTI , K_f , A_f , FC , BV , I_{br} 。

6.4.2 双极型晶体管模型

SPICE 中有两种双极型模型, (1) Ebers-Moll 模型; (2) Gummel-Poon 模型。

一、EM 模型

它具有简单直观的优点, 在多数情况下采用 EM 模型已经足够。

在直流的一维小注入条件下, 并忽略基区宽度随 V_{bc} 的变化, 则 EM 方程为:

$$\begin{cases} I_E = -I_{ES} \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] + a_R I_{CS} \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] \\ I_C = a_F I_{ES} \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] - I_{CS} \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] \end{cases} \quad (6-9)$$

从方程可导出直流模型, 示于图 6-7。图中

$$\begin{cases} I_E = I_{ES} \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] \\ I_C = I_{CS} \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] \end{cases} \quad (6-10)$$

这里 I_{ES} 和 I_{CS} 分别是结电压为零时, 发射极-基极反向饱和电流及集电极-基极反向饱和电流。 α_F 和 α_R 是共基接法时的正向及反向电流增益。

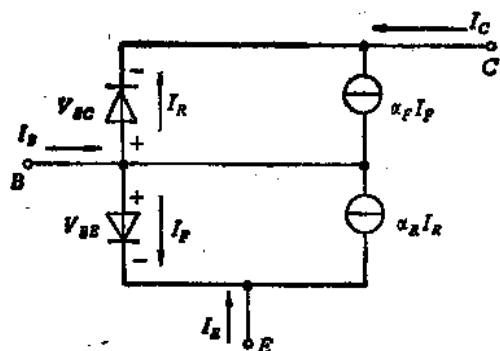


图 6-7 EM 直流模型

可以证明 (6-9) 方程满足互易定理, 即

$$\alpha_F I_{ES} = \alpha_R I_{CS} = I_S \quad (6-11)$$

这里的 I_S 称为晶体管传输饱和电流。

已知共发射极接法时的正向电流增益 β_F 和反向电流增益 β_R 为

$$\begin{cases} \beta_F = \frac{\alpha_F}{1 - \alpha_F} \\ \beta_R = \frac{\alpha_R}{1 - \alpha_R} \end{cases} \quad (6-12)$$

将 (6-11) 和 (6-12) 代入 (6-9) 式可得

$$\begin{cases} I_C = I_S \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] - I_S \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] - \frac{I_S}{\beta_R} \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] \\ I_E = -I_S \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] + I_S \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right] - \frac{I_S}{\beta_F} \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right] \end{cases} \quad (6-13)$$

若用

$$I_{EC} = I_S \left[\exp\left(\frac{qV_{BC}}{kT}\right) - 1 \right]$$

$$I_{CE} = I_S \left[\exp\left(\frac{qV_{BE}}{kT}\right) - 1 \right]$$

代入 (6-13) 式就得到

$$\begin{cases} I_{BE} = I_{EC} - I_{CE} \left(1 + \frac{1}{\beta_R} \right) \\ I_{BC} = -I_{CE} \left(1 + \frac{1}{\beta_F} \right) + I_{EC} \end{cases} \quad (6-14)$$

因而

$$I_B = -(I_C + I_E) = \frac{I_{EC}}{\beta_F} + \frac{I_{CE}}{\beta_R} \quad (6-15)$$

若令 $I_{CT} = I_{EC} - I_{CE}$, 并把 I_{CT} 看作单一电流源, 则模型变为混合 π 模型。这时再将发射极电阻、集电极电阻、基极电阻考虑进去则得到图 6-8。

因而描述 EM 直流模型的参数有六个: I_S , β_F , β_R , R_B , R_E , R_C 。

如考虑存储效应后, 其模型改为图 6-9。图中 C_{JE} 和 C_{JC} 分别为发射结和集电结势垒电容, C_{DE} 和 C_{DC} 分别为发射结和集电结扩散电容, C_{ss} 为集电极-衬底电容。

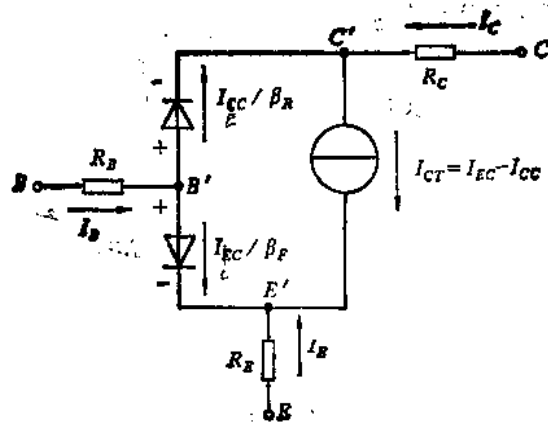


图 6-8 π 模型

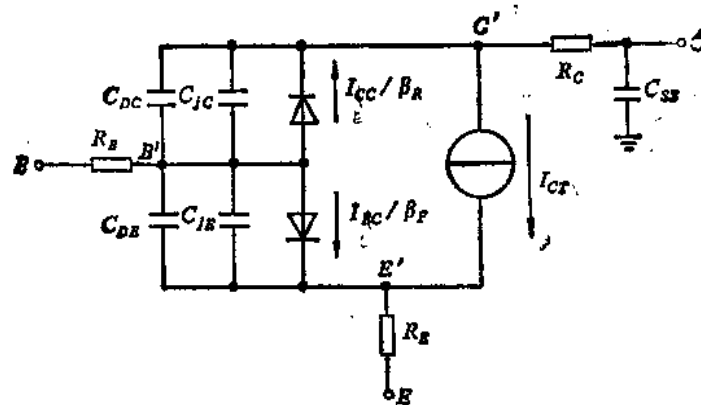


图 6-9 考虑存储效应后的 EM 模型

$$\begin{cases} C_{JE}(V_{B'E'}) = C_{JE0} \left(1 - \frac{V_{B'E'}}{\phi_B}\right)^{-m_B} \\ C_{JC}(V_{B'C'}) = C_{JC0} \left(1 - \frac{V_{B'C'}}{\phi_C}\right)^{-m_C} \\ C_{DE} = \frac{I_S}{V_T} \tau_F \exp\left(\frac{V_{B'E'}}{V_T}\right) \\ C_{DC} = \frac{I_S}{V_T} \tau_R \exp\left(\frac{V_{B'C'}}{V_T}\right) \\ C_{CS} = C_{CS0} \left(1 - \frac{V_{C'S'}}{\phi_S}\right)^{-m_S} \end{cases} \quad (6-16)$$

当正向偏压 $V > FC \cdot \phi$ 时, 所有耗尽层电容都要作相应的修改。

因此模型参数有十二个: C_{JE0} , ϕ_B , m_B , C_{JC0} , ϕ_C , m_C , C_{JS0} , ϕ_S , m_S , τ_F , τ_R , FC 。

如再考虑 I_S , 正反向 β 和结反向饱和电流随温度的影响以及噪声模型, 应再增加 E_A , XTB , XTI , K_f 和 A_f 。

二、GP 模型

GP 模型与 EM 模型相比较, 主要有主方面的差别: (1) 传输电流部分增加了 Q_b 项以反映大电流特性和 Early 效应; (2) 增加了两个非线性元件以反映小电流下空间电荷区复合电流的影响引起 β 值的下降; (3) 考虑了发射系数的影响; (4) 集电结电容的分布性能。

GP 模型中电流表达式为

$$\begin{cases} I_c = \frac{I_s}{Q_b} \left[\exp\left(\frac{qV_{B'E'}}{n_F kT}\right) - \exp\left(\frac{qV_{B'C'}}{n_R kT}\right) \right] - \frac{I_s}{\beta_R} \left[\exp\left(\frac{qV_{B'C'}}{n_R kT}\right) - 1 \right] \\ \quad - I_{sc} \left[\exp\left(\frac{qV_{B'C'}}{n_C kT}\right) - 1 \right] \\ I_b = \frac{I_s}{\beta_F} \left[\exp\left(\frac{qV_{B'E'}}{n_F kT}\right) - 1 \right] + \frac{I_s}{\beta_R} \left[\exp\left(\frac{qV_{B'C'}}{n_R kT}\right) - 1 \right] \\ \quad + I_{sb} \left[\exp\left(\frac{qV_{B'E'}}{n_e kT}\right) - 1 \right] + I_{sc} \left[\exp\left(\frac{qV_{B'C'}}{n_C kT}\right) - 1 \right] \end{cases} \quad (6-17)$$

式中, $Q_b = \frac{Q_1}{2} (1 + \sqrt{1 + 4Q_2/Q_1^2})$,

$$Q_1 = \frac{1}{1 - \frac{V_{B'C'}}{V_{AF}} - \frac{V_{B'E'}}{V_{AR}}},$$

$$Q_2 = \frac{I_s}{I_{KF}} \left[\exp\left(\frac{qV_{B'E'}}{n_F kT}\right) - 1 \right] + \frac{I_s}{I_{KR}} \left[\exp\left(\frac{qV_{B'C'}}{n_R kT}\right) - 1 \right].$$

在以上方程中, n_F 为正向电流发射系数, n_R 为反向电流发射系数, n_e 为 BE 结 泄漏发射系数, n_C 为 BC 结 泄漏发射系数, I_{sb} 为 BE 结 泄漏饱和电流, I_{sc} 为 BC 结 泄漏饱和电流, V_{AF} 为正向欧拉电压, V_{AR} 为反向欧拉电压, I_{KF} 为正向 β_F 的电流上升梯度转折点, I_{KR} 为反向 β_R 的电流上升梯度转折点。

在 I_c 的表达式中, 第一项是集电极电流中两个结电流相互作用的部分即传输部分。第二项是集电极注入电流所产生的复合电流。第三项是小电流下 BC 结空间电荷区的复合电流。

在 I_b 的表达式中, 前两项是基区复合电流, 后两项是小电流下 BE 结、BC 结空间电荷区的复合电流。

在 Q_b 项中的 V_{AF} 和 V_{AR} 反映基区调制效应对晶体管输出电导的影响。 I_{KF} 和 I_{KR} 反映大电流下 β 值的变化。

这时 GP 模型比起 EM 模型增加了十个模型参数: $n_F, n_R, n_e, n_C, I_{sb}, I_{sc}, V_{AF}, V_{AR}, I_{KF}, I_{KR}$ 。若用户不给出这十个参数, 程序就自动简化为 EM 模型。

此外 GP 模型中的基极电阻 $R_{bb'}$ 包括有源基区外的串联电阻和有源基区内的扩展电阻两部分。

基极电阻根据是否设定 I_{RB} 而有不同的表达式。

若没有设定 I_{RB} , 则

$$R_{BB'} = R_{BM} + \left(\frac{R_B - R_{BM}}{Q_B} \right) \quad (6-18)$$

若设定 I_{KB} , 则

$$R_{BB'} = 3(R_B - R_{BM}) \left[\frac{\operatorname{tg}(Z) - Z}{Z \operatorname{tg}^2(Z)} \right] + R_{BM} \quad (6-19)$$

式中

$$Z = \frac{-1 + \left[144 \frac{I_B}{\pi^2 I_{KB}} + 1 \right]^{1/2}}{\frac{24}{\pi^2} \left(\frac{I_B}{I_{KB}} \right)^{1/2}}$$

I_{KB} 为基极电阻向最小值下降并处于一半时的电流。 R_{BM} 为大电流时的最小基极电阻。

如果没有给定 I_{KB} , 则 $R_{BB'}$ 有两项: 第一项为 R_{BM} 。第二项反映扩展电阻受基区宽度调制效应和大电流特性的影响。

如果给定 I_{KB} , 则当 $I_B \rightarrow \infty$ 时, $R_{BB'} \rightarrow R_{BM}$ 。这说明大电流时基区调制效应使扩展电阻变得很小, $R_{BB'}$ 的主要成份是有源基区外的串联电阻。当 $I_B \rightarrow 0$ 时, $R_{BB'} \rightarrow R_B$, 即为零偏压时的基极电阻。

对于瞬态和交流特性, GP 模型也作了若干修正。

(1) 考虑正向渡越时间 τ_F 随偏置的变化

设有效的正向渡越时间为 τ_{FF} , 其表达式为

$$\tau_{FF} = \tau_F \left\{ 1 + X_{FF} \left[\exp \left(\frac{V_{B'C'}}{1.44 V_{FF}} \right) \right] \left[\frac{I_F}{I_F + I_{FF}} \right] \right\} \quad (6-20)$$

式中, $I_F = I_{SB} \left[\exp \left(\frac{q V_{B'E'}}{n_F k T} \right) - 1 \right]$, X_{FF} 为 τ_F 随偏置变化的系数, V_{FF} 为 τ_F 随 V_{BE} 变化的电压, I_{FF} 为影响 τ_F 的大电流参数。此公式说明当 I_F 与 I_{FF} 可比时, 随着 I_F 的增加, τ_{FF} 将增大。

(2) 引入超相移参数 P_{FF}

P_{FF} 描述交流分析中跨导 g_m 的延迟 (线性附加相移), 同时用于瞬态分析的贝塞尔多项式近似中。超相移参数 P_{FF} 被定义为在频率

$$f = \frac{1}{2\pi\tau_F} \quad (6-21)$$

时的超前相移。

(3) 考虑 BC 结电容沿 $R_{BB'}$ 电阻的分布特性

在 GP 模型中把 C_{BC} 电容分成两个电容 C_{BC1} 和 C_{BC2} , 它们分别跨接在 $R_{BB'}$ 的两侧。这两个电容的分配比例由参数 X_{CJC} 确定。 X_{CJC} 表示 BC 结电容 C_{BC} 连到内部基区节点上的百分比。

GP 模型总结于图 6-10。图中

$$I_{LB} = I_{SB} \left[\exp \left(\frac{q V_{B'E'}}{n_F k T} \right) - 1 \right], \quad I_{LC} = I_{SC} \left[\exp \left(\frac{q V_{B'C'}}{n_C k T} \right) - 1 \right].$$

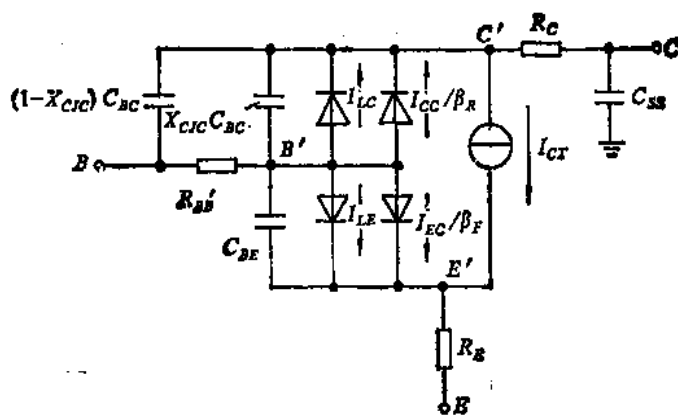


图 6-10 GP 模型

三、小信号线性化模型

双极型晶体管的小信号线性化模型示于图 6-11。正向注入的发射结可等效为 g_r 和 $C_{DE} + C_{JE}$ 并联。而 C_{BC} 为集电结扩散电容和势垒电容的并联，即

$$C_{BC} = \tau_R \left[\frac{q I_E}{n_R k T} \exp\left(\frac{q V_{B'E'}}{n_R k T}\right) \right] + C_{JG0} \left[1 - \frac{V_{B'C'}}{\phi_C} \right]^{-m_C} \quad (6-22)$$

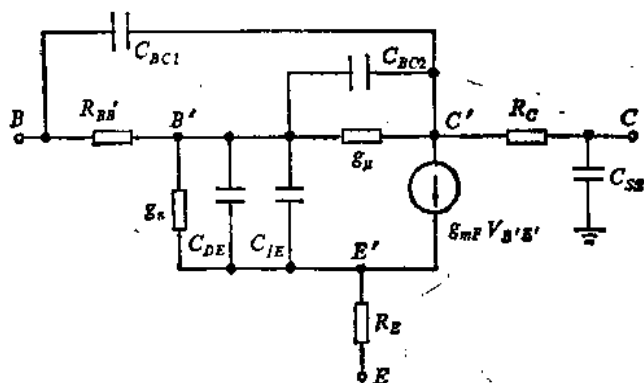


图 6-11 小信号线性化模型

根据 X_{CJC} ，可以计算分别跨接在 $R_{BB'}$ 两端的电容值为

$$\begin{cases} C_{BC1} = C_{BC}(1 - X_{CJC}) \\ C_{BC2} = C_{BC}X_{CJC} \end{cases} \quad (6-23)$$

集电极-衬底电容 C_{SS} 则为

$$C_{SS} = C_{JG0} \left(1 - \frac{V_{C'S'}}{\phi_S} \right)^{-m_S} \quad (6-24)$$

而

$$\begin{cases} g_r = \frac{\partial I_B}{\partial V_{B'E'}} \Big|_{\text{工作点}} \\ g_\mu = \frac{\partial I_B}{\partial V_{B'C'}} \Big|_{\text{工作点}} \\ g_{mF} = \frac{\partial I_B}{\partial V_{B'E'}} \Big|_{\text{工作点}} \end{cases} \quad (6-25)$$

6.4.3 结型场效应晶体管模型

N 沟 JFET 的模型示于图 6-12。其中栅源、栅漏极为两个 PN 结。非线性漏源电流 I_D 受栅源电压 V_{GS} 的控制。电阻 r_s 和 r_d 分别为源区和漏区电阻。对于 P 沟 JFET，则端电压 V_{GD} 、 V_{GS} 、 V_{DS} 的极性与图中正好相反，电流方向也相反。

$V_{DS} > 0$ 时为正向区，漏源电流

$$I_D = \begin{cases} 0 & V_{GS} - V_{T0} > 0 \\ \beta(V_{GS} - V_{T0})^2(1 + \lambda V_{DS}) & 0 < V_{GS} - V_{T0} < V_{DS} \\ \beta V_{DS}[2(V_{GS} - V_{T0}) - V_{DS}](1 + \lambda V_{DS}) & 0 < V_{DS} < V_{GS} - V_{T0} \end{cases} \quad (6-26)$$

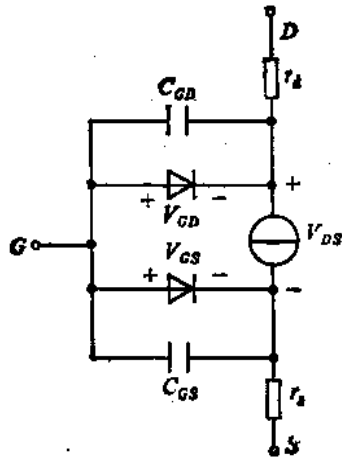


图 6-12 结型场效应管模型

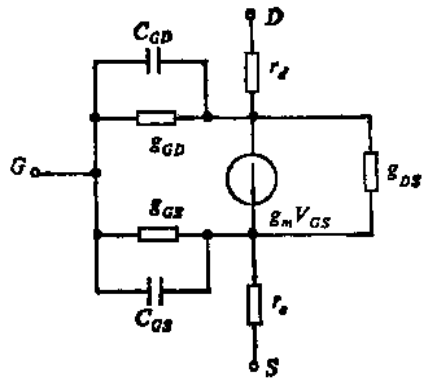


图 6-13 JFET 线性化模型

$V_{DS} < 0$ 时为反向区，漏源电流

$$I_D = \begin{cases} 0 & V_{GD} - V_{T0} < 0 \\ -\beta(V_{GD} - V_{T0})^2(1 - \lambda V_{DS}) & 0 < V_{GD} - V_{T0} < -V_{DS} \\ \beta V_{DS}[2(V_{GD} - V_{T0}) + V_{DS}](1 - \lambda V_{DS}) & 0 < -V_{DS} < V_{GD} - V_{T0} \end{cases} \quad (6-27)$$

式中， β 为跨导系数； V_{T0} 为阈值电压； λ 为沟道长度调制系数。模型中的二极管用理想二极管来表示，有

$$\begin{cases} I_{GD} = I_s \left[\exp\left(\frac{V_{GD}}{V_i}\right) - 1 \right] \\ I_{GS} = I_s \left[\exp\left(\frac{V_{GS}}{V_i}\right) - 1 \right] \end{cases} \quad (6-28)$$

电荷储存效应由两个栅结的耗尽层电容来模拟，即

$$\begin{aligned} C_{GD} &= C_{GD0} \left[1 - \frac{V_{GD}}{\phi_B} \right]^{-1/2} \\ C_{GS} &= C_{GS0} \left[1 - \frac{V_{GS}}{\phi_B} \right]^{-1/2} \end{aligned} \quad (6-29)$$

同样，当正向偏压 $V > FC \phi_B$ 时，耗尽层电容需要作修改。

交流小信号时的线性化模型如图 6-13。图中 g_{GD} 和 g_{GS} 是两个栅结的电导, 互导 g_m 和输出电导 g_{DS} 由下式计算而得

$$\begin{cases} g_m = \frac{\partial I_D}{\partial V_{GS}} \Big|_{\text{工作点}} \\ g_{DS} = \frac{\partial I_D}{\partial V_{DS}} \Big|_{\text{工作点}} \end{cases} \quad (6-30)$$

6.4.4 MOS 场效应晶体管模型

SPICE 中提供了四种 MOS 场效应模型, 用变量 *LEVEL* 来指定所用的模型。

LEVEL=1 MOS1 模型——SHICHMAN-HODGES 模型;

LEVEL=2 MOS2 模型——考虑二阶效应的解析模型;

LEVEL=3 MOS3 模型——半经验模型;

LEVEL=4 MOS4 模型——BSIM 模型 (Berkeley Short-Channel IGFET Model)。

一、MOS1 模型

这是 MOS 晶体管的一阶模型。适用于精度要求不高的长沟道 MOS 管。其模型示于图 6-14。电流方程如下。

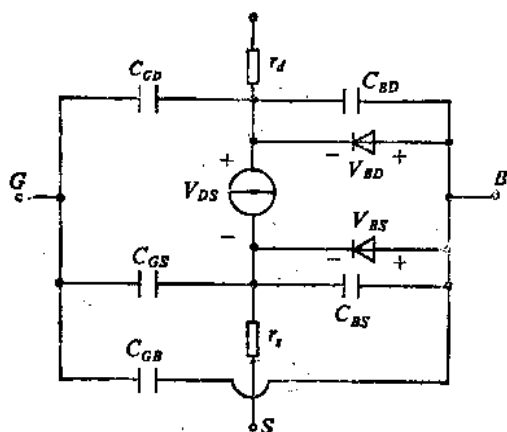


图 6-14 MOS1 模型

线性区 $V_{GS} - V_{TH} > V_{DS}$

$$I_{DS} = \frac{\beta}{2} [2(V_{GS} - V_{TH})V_{DS} - V_{DS}^2] \quad (6-31)$$

饱和区 $V_{GS} - V_{TH} \leq V_{DS}$

$$I_{DS} = \frac{\beta}{2} (V_{GS} - V_{TH})^2 \quad (6-32)$$

式中, $\beta = \frac{W}{L} \mu_n C_{ox}$ 为几何跨导参数, $K_F = \mu_n C_{ox}$ 为本征跨导系数。但饱和区漏电流因沟道长度调制效应会随 V_{DS} 的增加而略有增大, 故有

$$I_{DS} = \frac{\beta}{2} (V_{GS} - V_{TH})^2 (1 + \lambda V_{DS}) \quad (6-33)$$

式中, λ 为沟道长度调制系数。

阈值电压 V_{TH} 根据定义是使表面势变化 $2\phi_F$ 所需的栅极电压。因而

$$\begin{aligned} V_{TH} &= V_{FB} + 2\phi_F - \frac{Q_B}{C_{ox}} \\ &= V_{FB} + 2\phi_F + \frac{\sqrt{2q\epsilon_{Si}N_{SDB}}}{C_{ox}} (2\phi_F - V_{BS})^{1/2} \end{aligned} \quad (6-34)$$

V_{T0} 为 $V_{BS}=0$ 时的阈值电压, 所以

$$V_{T0} = V_{FB} + 2\phi_F + \frac{\sqrt{2q\epsilon_{s1}N_{SUB}}}{C_{ox}} \sqrt{2\phi_F}$$

$$= V_{BI} + \gamma \sqrt{2\phi_F} \quad (6-35)$$

式中, $V_{BI} = V_{FB} + 2\phi_F$; $\gamma = \frac{\sqrt{2q\epsilon_{s1}N_{SUB}}}{C_{ox}}$ 称为体效应参数。因此可得

$$V_{TH} = V_{T0} + \gamma(\sqrt{2\phi_F - V_{BS}} - \sqrt{2\phi_F}) \quad (6-36)$$

从上述讨论可知 MOS1 直流特性所需的模型参数为: V_{T0} , K_p , γ , $2\phi_F$, λ , R_D , R_S 。但器件参数 K_p 和 γ 也可通过输入工艺参数 μ_0 , T_{ox} , N_{SUB} 而由程序计算而得。

二、MOS2 模型

当 MOS 晶体管的几何尺寸减小时, 必须考虑器件的二阶效应; (1) 沟道长度对阈值电压的影响; (2) 漏栅静电反馈效应对阈值电压的影响; (3) 沟道宽度对阈值电压的影响; (4) 迁移率随表面电场的变化; (5) 沟道夹断引起的沟道长度调制效应; (6) 载流子漂移速度限制而引起的电流饱和效应; (7) 弱反型导电。

以下分别讨论这些效应。

1. 短沟道效应

当沟道长度小于 $5\mu\text{m}$ 时, 应考虑源区和漏区耗尽层存在对阈值电压的影响。

这时采用二维梯形模型来近似, 见图 6-15。

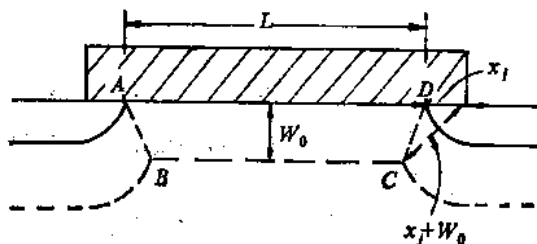


图 6-15 二维梯形模型

在一维模型中, 沟道耗尽层电荷 $Q_s \propto W_0 L$, 而在二维模型中 $Q_s' \propto$ 梯形面积 $ABCD$, 所以 Q_s 与 Q_s' 之比为

$$\frac{Q_s'}{Q_s} = 1 - \frac{X_j}{L} \left(\sqrt{1 + \frac{2W_0}{X_j}} - 1 \right) \quad (6-37)$$

因而体效应系数 γ 应改为 γ_s , 其值为

$$\gamma_s = \gamma \frac{Q_s'}{Q_s} = \gamma \left[1 - \frac{X_j}{L} \left(\sqrt{1 + \frac{2W_0}{X_j}} - 1 \right) \right] \quad (6-38)$$

阈值电压 V_{TH} 也改为

$$V_{TH} = V_{T0} + \gamma_s (\sqrt{2\phi_F - V_{BS}} - \sqrt{2\phi_F}) \quad (6-39)$$

这说明源漏区耗尽层电荷的存在使阈值电压下降。

2. 静电反馈效应

随着 V_{DS} 的增加, 在漏这一边的耗尽层宽度会有所增加。如图 6-16 所示。

这时漏区和源区的耗尽层宽度 W_D 和 W_S 分别为

$$W_D = X_D \sqrt{2\phi_F - V_{BS} + V_{DS}}$$

$$W_S = X_D \sqrt{2\phi_F - V_{BS}}$$

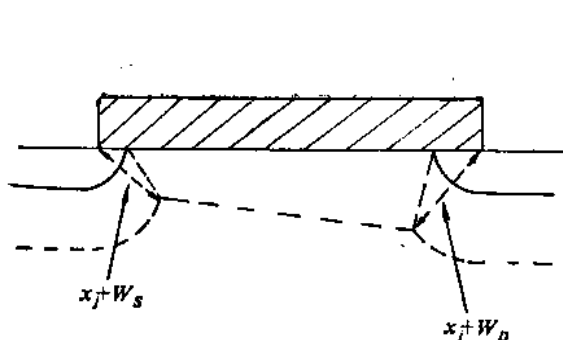


图 6-16 漏区耗尽层加宽

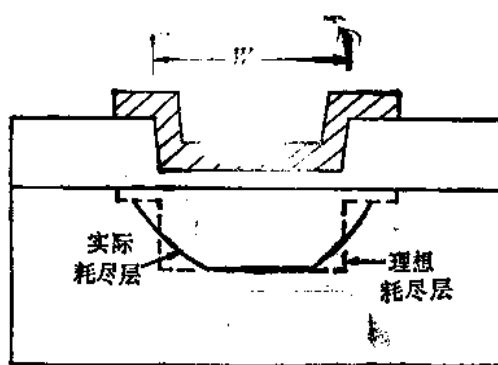


图 6-17 窄沟道效应

式中, $X_D = \sqrt{\frac{2\epsilon_{Si}}{N_{SUB}}}$ 因此 γ_s 值应修改为

$$\begin{aligned} \gamma_s &= \gamma \left\{ 1 - \frac{1}{2} \frac{X_j}{L} \left[\left(\sqrt{1 + \frac{2W_D}{X_j}} - 1 \right) + \left(\sqrt{1 + \frac{2W_s}{X_j}} - 1 \right) \right] \right\} \\ &= \gamma(1 - \alpha_D - \alpha_s) \end{aligned} \quad (6-40)$$

这说明随着 V_{DS} 的增加, 因 W_D 的增加使阈值电压进一步下降。

在实际使用 SPICE 时, 如不给出模型参数 α_j , 则上述修正就不再进行。

3. 窄沟道效应

当沟道宽度 W 变窄时, 要考虑所谓“边缘”效应。因为实际的栅总有一部分要覆盖在场氧化层上 (沟道宽度以外), 因此在场氧化层下面也会引起耗尽电荷。这部分电荷虽然很少, 但当沟道宽度 W 很窄时, 它在整个耗尽电荷中所占的比例将增大。栅电压比起没有“边缘”效应时的要加得较大才能使沟道反型。

窄沟道效应的示意图见图 6-17。

这时 V_{TH} 被修正为

$$\begin{aligned} V_{TH} &= V_{FB} + 2\phi_F + \gamma_s \sqrt{2\phi_F - V_{BS}} \\ &\quad + \delta \frac{\pi\epsilon_{Si}}{4C_{ox}W} (2\phi_F - V_{BS}) \end{aligned} \quad (6-41)$$

上式中最后一项表示窄沟道效应。 δ 为一经验系数。如给定此模型参数, 阈值电压会有所上升。

4. 迁移率修正

在电流方程中实际的表面迁移率要受到 V_{GS} 、 V_{DS} 和 V_{BS} 变化的影响。现有以下经验公式:

$$\mu_s = \mu_0 \left[\frac{U_{CRIT}\epsilon_{Si}}{C_{ox}(V_{GS} - V_{TH} - U_{TRA}V_{DS})} \right]^{U_{EXP}} \quad (6-42)$$

式中, U_{CRIT} 为临界电场强度; U_{EXP} 为迁移率下降的临界场指数; U_{TRA} 为横向电场系数; μ_0 为表面迁移率。实际上在 MOS2 模型中不考虑 V_{DS} 的影响, 即认为 $U_{TRA} = 0$ 。

综合以上讨论, 根据 Grove 所推导的漏电流

$$I_{DS} = \mu_0 C_{ox} \frac{W}{L} \left\{ \left(V_{GS} - V_{FB} - 2\phi_F - \frac{V_{DS}}{2} \right) V_{DS} - \frac{2}{3} \gamma_s [(V_{DS} + 2\phi_F - V_{BS})^{3/2} - (2\phi_F - V_{BS})^{3/2}] \right\} \quad (6-43)$$

将修改为

$$I_{DS} = \beta \left\{ \left(V_{GS} - V_{BIN} - \frac{\eta V_{DS}}{2} \right) V_{DS} - \frac{2}{3} \gamma_s [(V_{DS} + 2\phi_F - V_{BS})^{3/2} - (2\phi_F - V_{BS})^{3/2}] \right\} \quad (6-44)$$

式中, $\beta = \mu_s C_{ox} \frac{W}{L}$,

$$\eta = 1 + \delta \frac{\pi \epsilon_{s1}}{4 C_{ox} W};$$

$$\gamma_s = \gamma (1 - \alpha_D - \alpha_s);$$

$$V_{BIN} = V_{FB} + 2\phi_F + \delta \frac{\pi \epsilon_{s1}}{4 C_{ox} W} (2\phi_F - V_{BS});$$

$$\mu_s = \mu_0 \left[\frac{V_{CRIT}}{V_{GS} - V_{TH} - U_{TKA} V_{DS}} \right]^{U_{BXP}};$$

$L = L_0 - 2L_D$ (L_0 为几何沟道长度, L_D 为横向扩散距离)。

5. 沟道长度调制效应

当 V_{DS} 增大时, MOS 器件的漏端沟道被夹断并进入饱和。 V_{DS} 进一步增大, 该夹断点向源区移动, 从而使沟道的有效长度减小, 这就是沟道夹断所引起的沟道长度调制效应。

当 V_{DS} 增大到恰好使夹断点出现在漏极边缘处的临界电压称为饱和电压 $V_{DSAT} = V_{GS} - V_{TH0}$ 。

在考虑了短沟和窄沟效应之后, V_{DSAT} 修正为

$$V_{DSAT} = \frac{V_{GS} - V_{BIN}}{\eta} + \frac{1}{2} \left(\frac{\gamma_s}{\eta} \right)^2 \left\{ 1 - \sqrt{1 + 4 \left(\frac{\eta}{\gamma_s} \right)^2 \left(\frac{V_{GS} - V_{BIN}}{\eta} + 2\phi_F - V_{BS} \right)} \right\} \quad (6-45)$$

当 $V_{DS} > V_{DSAT}$ 时, 沟道有效长度为

$$L_{eff} = L - \Delta L \quad (6-46)$$

其中 $L = L_0 - 2L_D$ 。如已知沟道长度调制系数 λ , 则

$$\Delta L = \lambda L V_{DS}$$

如 λ 未给定, 则由下式计算出 ΔL

$$\Delta L = X_D \left[\frac{V_{DS} - V_{DSAT}}{4} + \sqrt{1 + \left(\frac{V_{DS} - V_{DSAT}}{4} \right)^2} \right]^{1/2} \quad (6-47)$$

此时电流公式中的 L 应改为 L_{eff} , 即

$$\frac{W}{L_{eff}} = \frac{W}{L - \Delta L} = \frac{W}{L(1 - \lambda V_{DS})} \quad (6-48)$$

在短沟器件中, V_{DS} 很高时会出现穿通, 这个现象在 SPICE 中没有加以模拟, 因此 $L_{eff} = L - \Delta L$ 的公式仅适用于 $L_{eff} > W_B$ 时 (W_B 为零偏压时的耗尽层宽度)。当 $L_{eff} < W_B$ 时, L_{eff} 会出现负值, 因此将 L_{eff} 公式改为

$$L_{eff} = \frac{W_B}{1 + \frac{\Delta L - L_{max}}{W_B}} \quad (6-49)$$

式中, $L_{max} = L - W_B$ 。采用(6-49)式后可以防止数学上的不收敛。

6. 载流子有限漂移速度引起的电流饱和

在短沟道器件中, 在沟道夹断以前就出现饱和。这是因为载流子的漂移速度达到了最大极限因而引起漏源电流的饱和。也就是说, 对于同样的几何尺寸比, 同样的工艺和偏置, 短沟道器件比起长沟道器件来讲饱和电流小。

MOS2 模型中, 引入了参数 V_{max} 表示载流子的最大漂移速率,

$$V_{max} = \frac{I_{DSAT}}{WQ_{CHAN}} \quad (6-50)$$

将 I_{DSAT} 和 Q_{CHAN} 代入上式得

$$\begin{aligned} V_{max} = \mu_s \left\{ \left(V_{GS} - V_{B1N} - \frac{\eta V_{DSAT}}{2} \right) V_{DSAT} - \frac{2}{3} \gamma_s [(V_{DSAT} + 2\phi_F - V_{BS})^{3/2} - (2\phi_F - V_{BS})^{3/2}] \right\} & \left\{ L_{eff} [V_{GS} - V_{B1V} - \eta V_{DSAT} \right. \\ & \left. - \gamma_s (V_{DSAT} + 2\phi_F - V_{BS})^{1/2}] \right\}^{-1} \end{aligned} \quad (6-51)$$

这里 L_{eff} 是由有限载流子漂移速率所决定的有效沟道长度。

$$L_{eff} = L - X_D \sqrt{ \left(\frac{X_D V_{max}}{2\mu_s} \right)^2 + (V_{DS} - V_{DSAT}) } + \frac{X_D^2 V_{max}}{2\mu_s} \quad (6-52)$$

可以看出

$$\begin{cases} V_{max} = f_1(I_{eff}, V_{DSAT}) \\ L_{eff} = f_2(V_{max}, V_{DSAT}, V_{DS}) \end{cases} \quad (6-53)$$

因此, 为了求解 V_{DSAT} , 需要求解两个非线性联立方程。在 SPICE 中采取了简化方法:

(1) 取 $L_{eff} = L$ 代入 V_{max} 公式, 在给定的 V_{GS} 、 V_{BS} 条件下求出 V_{DSAT} 。(2) 利用这个 V_{DSAT} 和 V_{DS} 求出 L_{eff} 值, 并进而求得 I_{DSAT} 。但因为这时的 V_{DSAT} 并不精确, 所以得到的 L_{eff} 也不精确。(3) 引入实验参数 N_{EFF} , 使模拟结果与实际测量相符合, N_{EFF} 是用来修正 X_D 公式中的 N_{SUB} , 因此 X_D 公式改为

$$X_D = \sqrt{ \frac{2\epsilon_{s1}}{qN_{SUB}N_{EFF}} } \quad (6-54)$$

N_{EFF} 称为总沟道电荷 (固定的和可动的) 系数。 N_{EFF} 的存在使 X_D 下降, 因而可调整 L_{eff} 值, 使模拟结果与测量特性相一致。

7. 弱反型导电

MOSFET 并不是一个理想的开关, 实际上在阈值电压以前就存在电流, 这个电流称为弱反型电流或次开启电流。对于工作在弱反型区的器件, 模拟这一行为很重要。

SPICE 中定义一个新的阈值电压 V_{ON} 。它标志着器件从弱反型进入了强反型。即 $V_{GS} \leq V_{ON}$ 时为弱反型, $V_{GS} \geq V_{ON}$ 时为强反型。如图 6-18。

现定义 V_{ON} 为

$$V_{ON} = V_{TH} + \frac{n k T}{q} \quad (6-55)$$

其中 $n = 1 + \frac{C_{FS}}{C_{ox}} + \frac{C_D}{C_{ox}}$

式中, $C_{FS} = q N_{FS}$

$$C_D = \frac{\partial Q_D}{\partial V_{BS}} = \left[-\gamma_s \frac{\partial \sqrt{2\phi_F - V_{BS}}}{\partial V_{BS}} - \frac{\partial \gamma_s}{\partial V_{BS}} \sqrt{2\phi_F - V_{BS}} + \delta \frac{\pi \epsilon_{si}}{4 C_{ox} W} \right] C_{ox}$$

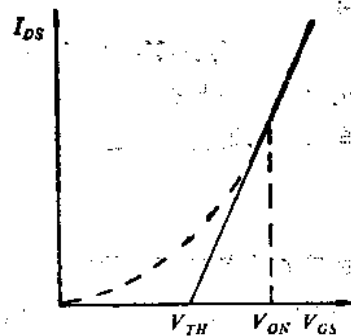


图 6-18 器件从弱反型进入强反型

式中 N_{FS} 是分析弱反型所必须规定的一个参数, 它也是计算 V_{ON} 的一个参数。

但这里的 N_{FS} 并不代表真正的表面快态密度, 而是为了与实验曲线相符合而设定的一个参数, 它需要从实际测量次开启电流中提取出来

公式中的 $\sqrt{2\phi_F - V_{BS}}$ 一项, 当 V_{BS} 由负变为正时, 将使计算发生困难, 为此将平方根改为 $F(V_{BS})$, 即

$$F(V_{BS}) = \frac{\sqrt{2\phi_F}}{1 + \frac{V_{BS}}{4\phi_F}} \quad (6-56)$$

在 V_{BS} 变正时, $F(V_{BS})$ 趋于零。

在弱反型导电时, 漏源电流方程为

$$I_{DS} = \beta \left\{ \left[V_{ON} - V_{BS} - \frac{\eta V_{DS}}{2} \right] V_{DS} - \frac{2}{3} \gamma_s [(V_{DS} + 2\phi_F - V_{BS})^{3/2} - (2\phi_F - V_{BS})^{3/2}] \right\} \exp \left[\frac{q}{n k T} (V_{GS} - V_{ON}) \right] \quad (6-57)$$

此方程保证电流在 V_{ON} 点上连续。

综上所述, 在 MOS2 模型中增加了下列模型参数: $X_i, \delta, U_{CRIF}, U_{EXP}, L_D, V_{max}, N_{EFF}, N_{FSO}$

三、MOS3 模型

它是一个半经验模型。适应于 $L \leq 2\mu m$ 和 $W \leq 2\mu m$ 时的小尺寸器件。这时考虑的器件二阶效应有:

- (1) 由于二维电位分布使阈值电压受到器件长度和宽度的影响;
- (2) 静电反馈对阈值电压的影响;
- (3) 由于热电子效应使线性区和饱和区之间的过渡减缓;
- (4) 由于热电子速度饱和引起饱和电压和饱和电流的下降。

MOS3 中的模型参数大多与 MOS2 相同, 但新引入了四个参数, 即 $\eta, \delta, \text{DELTA}, \text{KAPPA}$ 和 THETA 。以下分别讨论上述效应及四个参数的意义。

1. 阈值电压

MOS3 中考虑了短沟道效应、窄沟道效应和静电反馈效应对阈值电压的影响。 V_{TH} 的表达式为

$$V_{TH} = V_{FB} + 2\phi_F - \sigma V_{DS} + \gamma F_s \sqrt{2\phi_F - V_{BS}} + F_N(2\phi_F - V_{BS}) \quad (6-58)$$

这里的 σ 为静电反馈系数, 表明阈值电压随 V_{DS} 的增加而线性的下降。常数 σ 反比于氧化层电容及沟道长度的三次方, 即

$$\sigma = ETA \frac{\Omega}{C_{ox} L^3} \quad (6-59)$$

这里 Ω 为经验值, 等于 $8.15 \times 10^{-22} \text{F} \cdot \text{m}$ 。且引入了 ETA 这个模型参数。

F_s 为短沟道效应的校正因子。在 MOS3 中采用改进的梯形耗尽层模型, 考虑了圆柱形电场分布的影响, 如图 6-19。

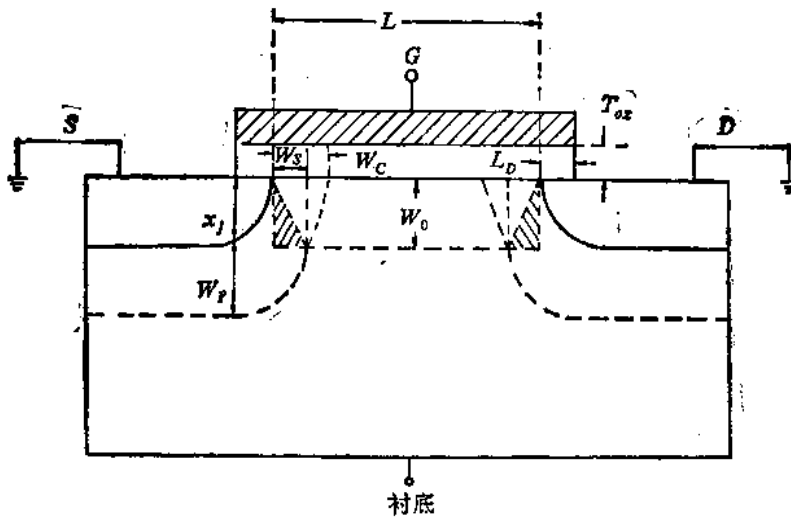


图 6-19 修正后的梯形模型

图中, W_c 为圆柱结耗尽层宽度; $W_p = X_D \sqrt{2\phi_F - V_{BS}}$ 为平面结耗尽层宽度, 并有以下关系

$$\frac{W_c}{X_j} = d_0 + d_1 \frac{W_p}{X_j} + d_2 \left(\frac{W_p}{X_j} \right)^2$$

$$\left(\frac{W_0}{W_p + X_j} \right)^2 + \left(\frac{W_s + L_D}{W_c + L_D} \right)^2 = 1$$

这里, d_0, d_1, d_2 为经验值, $d_0 = 0.0631353$; $d_1 = 0.8013292$; $d_2 = -0.01110777$ 。

MOS3 模型中 F_s 为

$$F_s = 1 - \frac{X_j}{L} \left[\frac{L_D + W_c}{X_j} \sqrt{1 - \left(\frac{W_p/X_j}{1 + W_p/X_j} \right)^2} - \frac{L_D}{X_j} \right] \quad (6-60)$$

与 MOS2 相同, 如不设定 X_j , 则短沟道效应不予考虑。

F_N 为窄沟道效应校正因子, 除了上节所述的附加体电荷的“边缘”效应外, 窄沟道效应还包括场注入和非等平面等其它“边缘”效应。 F_N 值为

$$F_N = DELTA \frac{\pi e_{s1}}{2C_{ox}W_f} \quad (6-61)$$

这里引入 $DELTA$ 这一参数, 称为阈值电压宽度效应系数。

2. 表面迁移率调制

这个效应采用最简单的经验公式来表示, 有

$$\mu_s = \frac{\mu_n}{1 + THETA(V_{GS} - V_{TH})} \quad (6-62)$$

这里 $THETA$ 被称为迁移率调制系数。

3. 热电子效应使过渡减缓

热电子的速度饱和使线性区的导电电流减小, 这使线性区与饱和区之间的过渡减缓。在线性区中这种效应由下列公式模拟:

$$\mu_{EFP} = \frac{\mu_s}{1 + \frac{\mu_s}{V_{max}L} V_{DS}} \quad (6-63)$$

如果不给出 V_{max} 参数, 则 μ_{EFP} 等于 μ_s , 此效应不予考虑。

4. 饱和电压的下降

根据公式(6-50)可以计算饱和电压, 即

$$I_{DSAT} = W Q_{CHAN} V_{max}$$

现将 I_{DSAT} 和 Q_{CHAN} 代入上式。

对 MOS3 模型, 有

$$I_{DSAT} = \frac{W}{L} \mu_s C_{ox} \left[V_{GS} - V_{TH} - \frac{1+F_B}{2} V_{DSAT} \right] V_{DSAT} \quad (6-64)$$

式中, $F_B = \frac{\gamma F_s}{4\sqrt{2\phi_F - V_{BS}}} + F_N$ 。而 Q_{CHAN} 为

$$Q_{CHAN} = C_{ox} [V_{GS} - V_{TH} - (1 + F_B) V_{DSAT}] \quad (6-65)$$

将(6-64)和(6-65)代入(6-50)式得到

$$\mu_s \left[V_{GS} - V_{TH} - \frac{1+F_B}{2} V_{DSAT} \right] V_{DSAT} = L [V_{GS} - V_{TH} - (1 + F_B) V_{DSAT}] V_{max} \quad (6-66)$$

从公式(6-66)可解得

$$V_{DSAT} = \frac{V_{GS} - V_{TH}}{1 + F_B} + \frac{V_{max}L}{\mu_s} - \sqrt{\left(\frac{V_{GS} - V_{TH}}{1 + F_B} \right)^2 + \left(\frac{V_{max}L}{\mu_s} \right)^2} \quad (6-67)$$

若未给出 V_{max} , 则饱和电压将由沟道夹断所决定。

5. 沟道长度调制

当 V_{DS} 大于 V_{DSAT} 时, 载流子速度饱和点的位置逐渐移向源区, 造成沟道长度调制效应。沟道长度的减小 ΔL 由下式决定, 即

$$\Delta L = X_D \sqrt{\left(\frac{E_F X_D}{2} \right)^2 + KAPPA(V_{DS} - V_{DSAT})} - \frac{X_D^2 E_F}{2} \quad (6-68)$$

式中 E_F 为夹断点处的横向电场, 有

$$E_p = \frac{I_{DSAT}}{G_{DSAT}L} \quad (6-69)$$

其中 I_{DSAT} 和 G_{DSAT} 分别为饱和区的饱和电流及输出电导。

以上引入了 $KAPPA$ 这一参数, 称为饱和场因子。与 MOS2 一样, 当 ΔL 接近 L 时, L_{eff} 要作修正。

在 MOS3 模型中对弱反型导电问题的分析仍引用 MOS2 中的公式。同样, 当设定 N_r 时, 程序就转向弱反型导电分析。

综上所述, 在 MOS3 模型中增加了四个经验参数: ETA , $DELTA$, $THETA$, $KAPPA$ 。

四、MOS4 模型 (BSIM 模型)

此模型考虑了小尺寸晶体管时的一些特殊效应, 它们包括: (1) 载流子迁移率与垂直电场的关系; (2) 载流子速度饱和; (3) 漏感应引起表面位垒下降; (4) 源和漏对耗尽层电荷的共享效应; (5) 离子注入后杂质非均匀分布的影响; (6) 沟道长度调制效应; (7) 弱反型区 (次开启区) 导电效应; (8) 电参数随几何尺寸的变化。

但 BSIM 的模型公式比较简单, 因而计算效率较高。

阈值电压由下列公式表示:

$$V_{TH} = V_{FB} + \phi_s + K_1 \sqrt{\phi_s - V_{BS}} - K_2 (\phi_s - V_{BS}) - \eta V_{DS} \quad (6-70)$$

式中, V_{FB} 为平带电压; ϕ_s 为表面反型时的电势; K_1 为体效应系数; K_2 为源漏区耗尽电荷共享系数; η 为漏感应引起的表面位垒下降系数。

线性区的漏源电流公式为

$$I_{DS} = \frac{\mu C_{ox}}{(1 + U_1 V_{DS})} \frac{W}{L} \left[(V_{GS} - V_{TH}) V_{DS} - \frac{a}{2} V_{DS}^2 \right] \quad (6-71)$$

式中

$$\mu = \frac{\mu_0}{1 + U_0 (V_{GS} - V_{TH})} \quad (6-72)$$

$$a = 1 + \frac{g K_1}{2 \sqrt{\phi_s - V_{BS}}} \quad (6-73)$$

$$g = 1 - \frac{1}{1.744 + 0.8364 (\phi_s - V_{BS})} \quad (6-74)$$

为了加速电路模拟, 这里采用了数值近似来代替漏电流随衬底偏压成 $3/2$ 次方变化的关系。

饱和区的漏电流公式为

$$I_{DS} = \frac{\mu C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2}{2aK} \quad (6-75)$$

式中

$$K = \frac{1 + v_a + \sqrt{1 + 2v_a}}{2} \quad (6-76)$$

$$v_c = \frac{U_1(V_{GS} - V_{TH})}{a} \quad (6-77)$$

以上的 μ_0 为载流子表面迁移率, U_0 为迁移率下降系数, U_1 为速度饱和系数。

弱反型区电流成份 $I_{DS\text{weak}}$ 为

$$I_{DS\text{weak}} = \frac{I_{exp} I_{lims}}{I_{exp} + I_{lims}} \quad (6-78)$$

式中

$$I_{exp} = \mu_0 C_{ox} \frac{W}{L} \left(\frac{kT}{q} \right)^2 e^{1.8} e^{\frac{V_{GS} - V_{TH}}{n} \left(\frac{q}{kT} \right)} \left[1 - e^{-\frac{q}{kT} V_{DS}} \right] \quad (6-79)$$

$$I_{lims} = \frac{\mu_0 C_{ox}}{2} \frac{W}{L} \left(3 \frac{kT}{q} \right)^2 \quad (6-80)$$

总的漏源电流将是强反型与弱反型电流之和, 即

$$I_{DS} = I_{DS\text{strong}} + I_{DS\text{weak}} \quad (6-81)$$

综上所述, MOS4 模型共有八个模型参数: V_{TH} , ϕ_s , K_1 , K_2 , η , μ_0 , U_0 , U_1 。

五、电容模型

SPICE 中有两种电容模型: (1) Meyer 模型, 它是隐含的模型; (2) 电荷控制模型 (Ward 和 Dutton 模型)。

现先讨论 Meyer 模型, 它是把随电压而变的电容分段加以线性化。电容包括 PN 结势垒电容和栅电容。

(1) PN 结势垒电容 C_B 和 C_{BS} 它们由两部分组成, 即底部势垒电容和侧壁势垒电容。于是有

$$C_j(V) = C_j(0) \frac{A_j}{\left(1 - \frac{V}{\phi_B}\right)^{m_j}} + C_{jsw}(0) \frac{P_j}{\left(1 - \frac{V}{\phi_B}\right)^{m_{jsw}}} \quad (6-82)$$

式中, $C_j(0)$ 为单位底面积零偏压结电容; $C_{jsw}(0)$ 为单位长度侧壁零偏压结电容; A_j 为底部电容面积; P_j 为侧壁电容周长; m_j 为底电容的梯度因子; m_{jsw} 为侧壁电容的梯度因子。当偏压 $V > FC \phi_B$ 时, $C_j(V)$ 同样要作相应的修正。

(2) 栅电容 C_{GS} , C_{GD} , C_{GB} 这些电容包括随偏压而变以及不随偏压而变两部分, 于是

$$\begin{cases} C_{GS} = C'_{GS} + C''_{GS} \\ C_{GD} = C'_{GD} + C''_{GD} \\ C_{GB} = C'_{GB} + C''_{GB} \end{cases} \quad (6-83)$$

而固定不变的部分是栅极与源区、漏区的交叠氧化层电容以及栅与衬底间的交叠氧化层电容 (在场氧化层上), 即

$$\begin{cases} C''_{GS} = C_{GS0} W \\ C''_{GD} = C_{GD0} W \\ C''_{GB} = C_{GB0} L \end{cases} \quad (6-84)$$

式中, C_{GS0} 为每单位沟道宽度的栅源交叠电容; C_{GD0} 为每单位沟道宽度的栅漏交叠电容; C_{GB0} 为每单位沟道长度的栅-衬底交叠电容。

随偏压而变的栅电容是栅氧化层电容与空间电荷区电容的串联。已知单位面积栅极上的电荷 Q_g 为

$$Q_g = \frac{2}{3} C_{ox} \left[\frac{(V_{GS} - V_{DS} - V_{TH})^3 - (V_{DS} - V_{TH})^3}{(V_{GS} - V_{TH})^2 - (V_{DS} - V_{TH})^2} \right] \quad (6-85)$$

所以随偏压而变的部分为

$$\begin{cases} C'_{GS} = WL \frac{\partial Q_g}{\partial V_{GS}} = \frac{2}{3} WLC_{ox} \left\{ 1 - \frac{(V_{GS} - V_{DS} - V_{TH})^2}{[2(V_{GS} - V_{TH}) - V_{DS}]^2} \right\} \\ C'_{GD} = WL \frac{\partial Q_g}{\partial V_{GD}} = \frac{2}{3} WLC_{ox} \left\{ 1 - \frac{(V_{GS} - V_{TH})^2}{[2(V_{GS} - V_{TH}) - V_{DS}]^2} \right\} \end{cases} \quad (6-86)$$

C'_{GS} 是氧化层电容 WLC_{ox} 与沟道-衬底电容 C_{sc} 的串联。

在不同的工作区，这些电容值的变化如图 6-20。

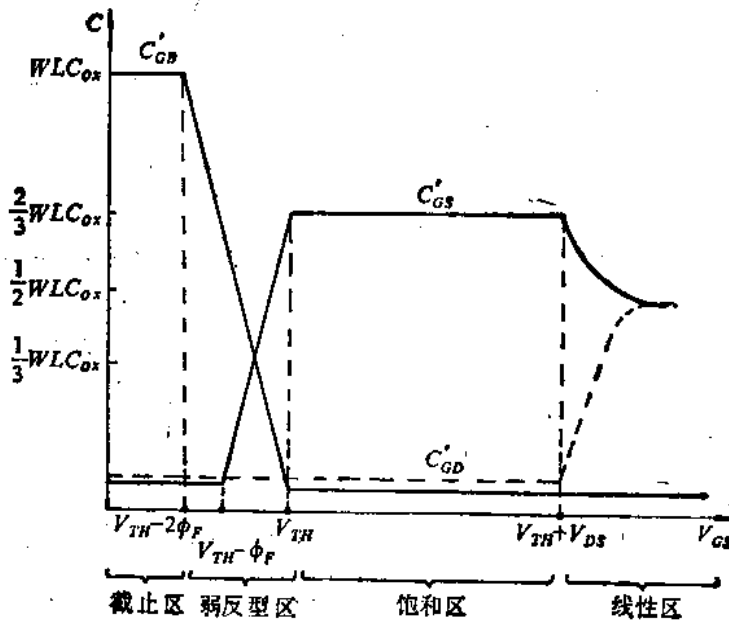


图 6-20 不同工作区电容值的变化

①截止区 $V_{GS} < V_{TH} - 2\phi_F$ 时

$$C'_{GD} \approx 0, C'_{GS} \approx 0, C'_{GB} = WLC_{ox}$$

②弱反型区 当 $V_{TH} - 2\phi_F < V_{GS} < V_{TH} - \phi_F$ 时

$$C'_{GD} \approx 0, C'_{GS} \approx 0$$

$$C'_{GB} = -WLC_{ox} \left(\frac{V_{GS} - V_{TH}}{2\phi_F} \right)$$

当 $V_{TH} - \phi_F < V_{GS} < V_{TH}$ 时

$$C'_{GS} = \frac{2}{3} WLC_{ox} \left(\frac{V_{GS} - V_{TH}}{\phi_F} \right) + \frac{2}{3} WLC_{ox}$$

③饱和区 当 $V_{GS} > V_{TH}$ 且 $V_{DS} \geq V_{GS} - V_{TH}$ 时

$$C'_{GD} \approx 0, C'_{GB} \approx 0$$

$$C'_{gs} = \frac{2}{3} WLC_{ox}$$

④线性区 当 $V_{gs} > V_{TH}$ 且 $V_{ds} < V_{gs} - V_{TH}$ 时

$$C'_{gs} \approx 0$$

$$C'_{gs} \text{ 和 } C'_{gd} \text{ 都趋于 } \frac{1}{2} WLC_{ox}$$

电荷控制模型是电荷守恒的电容模型，它遵守以下守恒定律

$$Q_c = -(Q_s + Q_D) \quad (6-87)$$

沟道电荷、栅极电荷与衬底电荷服从以上关系。与这些电荷相关联的器件端电流分别为

$$\begin{cases} i_g = \frac{dQ_c}{dt} \\ i_s = \frac{dQ_s}{dt} \\ i_s + i_D = \frac{d(Q_s + Q_D)}{dt} \end{cases} \quad (6-88)$$

其中 Q_s 和 Q_D 分别是沟道电荷 Q_c 分配给源区和漏区的电荷，即

$$Q_c = Q_s + Q_D \quad (6-89)$$

在线性区 Q_s 和 Q_D 平分沟道电荷 Q_c 。但在饱和区则不是平均分配，而有

$$\begin{cases} Q_D = X_{qc} Q_c \\ Q_s = (1 - X_{qc}) Q_c \end{cases} \quad (6-90)$$

式中 X_{qc} 为沟道电荷在漏端的分配系数，其值从零至 0.5。它作为一个系数，同时也是模型的标志。当 $X_{qc} > 0.5$ 或不作规定时，程序就采用 Meyer 模型。当 $0 < X_{qc} < 0.5$ 时，程序就采用电荷控制模型。

六、小信号模型

MOSFET 的线性化小信号模型示于图 6-21。

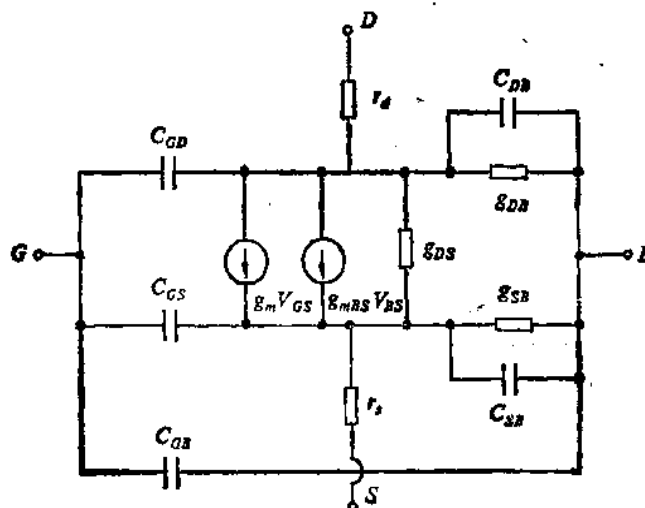


图 6-21 线性化小信号模型

图中 g_{DS} 和 g_{SS} 是两个衬底结的等效电导。其中

$$\begin{cases} g_m = \frac{\partial I_D}{\partial V_{GS}} \Big|_{\text{工作点}} \\ g_{DS} = \frac{\partial I_D}{\partial V_{DS}} \Big|_{\text{工作点}} \\ g_{SS} = \frac{\partial I_D}{\partial V_{BS}} \Big|_{\text{工作点}} \end{cases} \quad (6-91)$$

上面讨论了 MOS 器件的主要模型。有关噪声模型以及各模型参数随温度的变化在这里就不再赘述。

6.4.5 砷化镓 MESFET 场效应管模型

GaAs MESFET 直流时的电流公式为

$$\begin{cases} I_D = \frac{\beta(V_{GS} - V_T)^2}{1 + b(V_{GS} - V_T)} \left[1 - \left(1 - \frac{\alpha V_{DS}}{3} \right)^3 \right] (1 + \lambda V_{DS}) & \text{在 } 0 < V_{DS} < \frac{3}{\alpha} \text{ 时} \\ I_D = \frac{\beta(V_{GS} - V_T)^2}{1 + b(V_{GS} - V_T)} (1 + \lambda V_{DS}) & \text{在 } V_{DS} \geq \frac{3}{\alpha} \text{ 时} \end{cases} \quad (6-92)$$

因而, GaAs MESFET 模型中共有五个模型参数; 其中, V_T 为夹断电压; β 为跨导系数; b 为与掺杂分布有关的参数; α 为饱和电压参数; λ 为沟道长度调制系数。

§ 6.5 模型参数的提取

前面已经谈到, 电路模拟的精度除了依赖于器件模型本身, 还强烈地取决于所给定的模型参数值是否正确。因此能否精确地提取和确定晶体管的模型参数是一个普遍和重要的问题。通常用于提取模型参数的方法有两类: 一类是利用器件各工作区特点的分段个别提取法; 另一类是直接拟合特性曲线的优化提取法。

一、分段个别提取法

它是利用不同工作区的特性曲线或利用不同的关系曲线来提取一个或几个模型参数

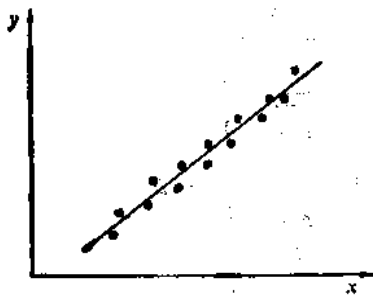


图 6-22 线性回归法

值。一种器件模型的各个参数需要通过多次分段才能得到。由于是利用不同的工作区和不同的关系曲线, 因而无法考虑模型参数之间的相互影响, 得到的模型参数值往往不够精确。但这种方法简单直观, 数学处理容易。

分段个别提取法最常采用线性拟合法, 其算法基础为线性回归, 也有采用抛物线拟合法的。

线性回归问题是已知某一函数为线性函数, 即

$$y = a + bx \quad (6-93)$$

如已知若干个实验测量值 x_i 和 y_i , 即可通过实验测量数据求得系数 a 和 b 。如图 6-22。线

性回归的计算公式如下：

$$b = \frac{\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n y_i\right) - N\left(\sum_{i=1}^n x_i y_i\right)}{\left(\sum_{i=1}^n x_i\right)^2 - N\left(\sum_{i=1}^n x_i^2\right)} \quad (6-94)$$

$$a = \frac{\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n x_i y_i\right) - \left(\sum_{i=1}^n y_i\right)\left(\sum_{i=1}^n x_i^2\right)}{\left(\sum_{i=1}^n x_i\right)^2 - N\left(\sum_{i=1}^n x_i^2\right)} \quad (6-95)$$

式中 N 为测量点个数。测量点个数越多，求得的 a , b 值就越精确。

我们可以建立一个以微型机为核心的模型参数提取系统，它控制测量仪器并获得测量数据，经以线性回归法为基础的软件包处理后得到提取值，并在系统上显示和打印出来。

现举例说明如下：

(1) V_{TH} , γ 和 N_{SUB} 的提取

测量 MOS 晶体管在饱和状态下的 $I_{DS} \sim V_{GS}$ (在不同 V_{BS} 条件下) 特性。因饱和区电流公式为

$$I_{DS} = \frac{\beta}{2} (V_{GS} - V_{TH})^2$$

经移项整理得线性方程

$$\sqrt{\frac{2I_{DS}}{\beta}} = V_{GS} - V_{TH} \quad (6-96)$$

因此可根据 $\sqrt{I_{DS}} \sim V_{GS}$ 测量结果，经线性回归得到线性曲线在横坐标上的截距值 V_{TH} ，同时可由其斜率求得 β 值 (或 K_F 值)。

由于在不同的 V_{BS} 下得到了一组 V_{TH} (在 $V_{BS}=0$ 时为 V_{T0})，即有

$$V_{TH} = V_{T0} + \gamma (\sqrt{2\phi_F - V_{BS}} - \sqrt{2\phi_F})$$

因而可根据 $V_{TH} \sim \sqrt{2\phi_F - V_{BS}}$ 曲线求得其斜率，即 γ 值。

进一步可根据公式

$$N_{SUB} = \frac{\gamma^2}{2q\epsilon_{Si}} \left(\frac{\epsilon_{ox}}{T_{ox}} \right)^2$$

通过给出 T_{ox} 值而求得 N_{SUB} 。

(2) μ_0 , U_{CRIT} , U_{EXP} 的提取

已知

$$\mu_3 = \mu_0 \left[\frac{\epsilon_{Si} U_{CRIT}}{C_{ox} (V_{GS} - V_{TH} - U_{TR4} V_{DS})} \right]^{U_{EXP}}$$

经移项整理得

$$\lg \frac{\mu_3}{\mu_0} = U_{EXP} \left[\lg \left(\frac{\epsilon_{Si}}{C_{ox}} U_{CRIT} \right) - \lg (V_{GS} - V_{TH} - U_{TR4} V_{DS}) \right] \quad (6-97)$$

这里将 $\frac{\mu_s}{\mu_0}$ 用 $\frac{K_{PI}}{K_{P0}}$ 代入 (K_{PI} 为不同 V_{GS} 时本征跨导参数, 而 K_{P0} 取所有本征跨导参数的最大值)。因而有 $\lg\left(\frac{K_{PI}}{K_{P0}}\right)$ 与 $\lg(V_{GS} - V_{TH} - U_{TR4} V_{DS})$ 的关系曲线。由截距和斜率可分别求得 U_{CRIT} 和 U_{EXP} 。

(3) 结电容的提取

$$\text{已知} \quad C_J(V) = C_J(0) \frac{A_s}{\left(1 - \frac{V}{\phi_B}\right)^{m_J}} + C_{JSW}(0) \frac{P_s}{\left(1 - \frac{V}{\phi_B}\right)^{m_{JSW}}}$$

首先对大面积矩形结电容进行测量, 这时可忽略 $C_J(V)$ 的第二项侧壁电容。从 $C-V$ 曲线中得到 $V=0$ 时的 C_{J1} 值, 因 A_s 已知, 所以可求得 $C_J(0)$, 即

$$C_J(0) = \frac{C_{J1}}{A_s} \quad (6-98)$$

有了 $C_J(0)$ 后, 再对梳状结电容进行测量, 求出不同偏置电压 V 时的电容值 $C_J(V)$ 。再通过下式求得侧壁电容值 $C_K(V)$

$$C_K(V) = C_J(V) - C_J(0) \frac{A_s}{\left(1 - \frac{V}{\phi_B}\right)^{1/2}} \quad (6-99)$$

因这时已知 $C_J(0)$, A_s , m_J , 所以可通过求得 $C_J(V) \sim V$ 的关系而得到 $C_K(V) \sim V$ 的关系。

如作 $\lg C_K(V)$ 与 $\lg(C_{JSW}(0)P_s) - m_{JSW} \lg\left(1 - \frac{V}{\phi_B}\right)$ 的曲线, 其中 P_s 为已知的梳状

结电容的周长, 就可从该曲线的斜率和截距中求得 m_{JSW} 和 $C_{JSW}(0)$ 值。

为了进行电容模型参数的提取, 需要制作专门用于测量的矩形和梳形电容。

此外我们可以看到, 为了提取所有的模型参数, 需要作多次的分段提取, 但提取又是从不同的公式而不是从一个统一的公式出发的, 所以如把提取出来的参数值再回到模拟方程中, 所得结果并不一定与测量曲线相拟合。

二、优化提取法

它是通过对测量曲线的整体拟合 (或称全局拟合) 来提取模型参数, 因此它是一个非线性拟合问题。

设模型 I 有 n 个模型参数 $x_1, x_2, \dots, x_n$, 现有 m 个测量值 $i_1, i_2, \dots, i_m$, 且 $m > n$, 则目标函数可写成以下形式:

$$\begin{aligned} S(x_1, x_2, \dots, x_n) &= \sum_{j=1}^m [I_j(x_1, x_2, \dots, x_n) - i_j]^2 \\ &= \sum_{j=1}^m f_j^2 \end{aligned} \quad (6-100)$$

式中, $f_j = I_j - i_j$ 。

优化过程就是求出一组模型参数, 使目标函数 S 取最小值。因 f 一般是非线性函数。

且目标函数具有平方项之和的形式,所以这是一个非线性最小二乘问题。

模型参数优化提取的原理图 示于图 6-

23。

非线性曲线拟合需采用非线性最优化方法,这大致分成两类:一类为直接搜索法,另一类是以目标函数梯度为基础的解析法。后一类中有线性化法(Gauss-Newton法)、最速下降法、阻尼最小二乘法及修正的阻尼最小二乘法(MDLS)等。

但优化提取法存在多值解,并且提取过程中存在多种误差,往往造成曲线拟合最好时所得的某些参数会失去其物理意义或偏离正常值过大。为了解决这一问题,引入考虑物理约束条件的目标函数,如有

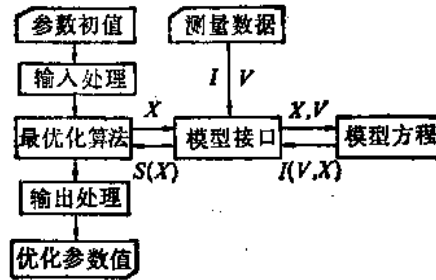


图 6-23 模型参数优化提取原理图

$$OF = \sum_i W_i (\Delta y_i)^2 + \sum_j W_j (\Delta S_j)^2 + R \sum_k W_k (\Delta P_k)^2 \quad (6-101)$$

这里 W_i , W_j , W_k 皆为权重因子。目标函数仍保持具有平方项之和的形式。上式中第一项是曲线拟合时通常所采用的误差函数;第二项是将 MOS 管输出饱和和区斜率的相对误差引入目标函数;第三项为惩罚项。引入惩罚项的目的是使目标函数不仅能反映曲线拟合程度的好坏,还能控制参数的变化,如要求参数之间服从某种关系(等式约束),或希望参数仅在一定的、合理的范围内变化(不等式约束)。第三项中的 R 为罚因子; ΔP_k 为

$$\Delta P_k = \frac{PV_k}{(PV_k - BOUND_k)} \quad (6-102)$$

式中, PV_k 为第 k 个参数的参数值; $BOUND_k$ 为该参数的上下边界值(约束)。

罚因子 $R = C^{-ITER}$, 其中 $0 < C \leq 1$; $ITER$ 是循环次数。

这样,当模型参数值偏离物理约束时,目标函数值将增大,这相当于在参数值的上下边界设置了一个势垒,使参数值难于趋近上下边界,物理约束将成为一种自动的反馈过程。罚因子的作用是使惩罚项的影响随着循环次数的增加而逐渐减小,以保证参数提取的客观性。

从以上讨论可以看出,优化整体提取的优点是从同一方程出发,一次提取所有的直流模型参数,而且考虑了各模型参数之间的相互影响。如果把提取出来的模型参数值回代到直流特性方程,所得的结果将与测量曲线有相当好的吻合。但它有着明显的缺点:一是算法比较复杂,程序编写工作量大;二是提取的某些参数会偏离正常值过大,甚至完全失去物理意义,因为它是一种纯数学上的拟合。我们可以通过一些措施例如在目标函数中引入惩罚项等来消除非物理解,但每个参数值是否完全正确仍然是一个问题。

下面我们介绍一种统计优化算法。它的主要思想是在参数空间中随机取样,通过选择取样空间而逐步逼近最优点。它可以克服牛顿法及其改进方法中所固有的缺点,如提取值与参数初值的选取有强烈的依赖关系,当参数空间维数增大时难于给出恰当的初值,求解过程中要求偏微分,有时无法收敛等问题。此算法不用求导数,优化结果受参数初值的影响小,程序编写工作量也较小。

统计优化算法的框图，见图 6-24。若有 n 个参数需要提取，且已知测量值

$$I_{mk}(dV(i), dW(j), dL(j))$$

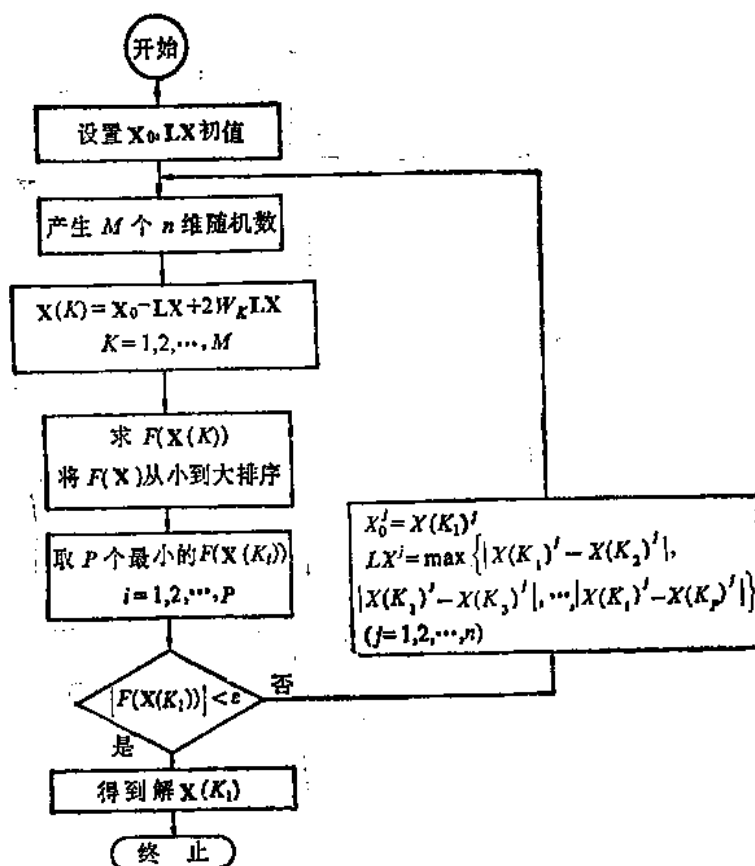


图 6-24 统计优化算法框图

现设置各提取参数的初始值 X_0 ，及该参数的上下界幅度 LX ，即参数上界为 $X_+ = X_0 + LX$ ，下界为 $X_- = X_0 - LX$ ，其中

$$X_0 = (X_0^1, X_0^2, \dots, X_0^n)$$

$$LX = (LX^1, LX^2, \dots, LX^n)$$

则提取步骤如下：

(1) 送入初始值和幅度 X_0 和 LX ；

(2) 产生 n 维随机数

$$W_K = (W^1, W^2, \dots, W^n)$$

随机数在 $[0, 1)$ 之间，即产生的随机数是在 $[X_0^1 - LX^1, X_0^1 + LX^1)$, $[X_0^2 - LX^2, X_0^2 + LX^2)$, ..., $[X_0^n - LX^n, X_0^n + LX^n)$ 范围内。

程序规定产生 M 个随机数，即 W_K , $K=1, 2, \dots, M$ (M 可设定为 200)。

(3) 计算参数值

$$X(K) = X_0 - LX + 2W_K LX \quad K=1, 2, \dots, M$$

式中， $2W_K LX = (2W_K^1 LX^1, 2W_K^2 LX^2, \dots, 2W_K^n LX^n)$

(4) 根据下列评价函数

$$F(X) = \sum [I_{nn}(dV(i), dW(j), dL(j), X(K)) - I_{nn}(dV(i), dW(j), dL(j))] / \text{测试点个数}$$

求出 $F(X(K))$ 值, 并将其从小到大依次排序。

(5) 取 $F(X(K_i))$ 的最初 P 个, 即

$$F(X(K_i)), i=1, 2, \dots, P \quad (P \text{ 可设定为 } 5)$$

(6) 判断 $|F(X(K_i))| < \varepsilon$ 。 ε 为误差要求。

如满足, 则 $X(K_i)$ 为所求解并退出循环。如不满足则修改 X_0 和 LX 值。

(7) 修改后的 X_0 和 LX 值为

$$X_0^j = X(K_1)^j$$

$$LX^j = \max\{|X(K_1)^j - X(K_2)^j|, |X(K_1)^j - X(K_3)^j|, \dots, |X(K_1)^j - X(K_p)^j|\} \quad (j=1, 2, \dots, n)$$

(8) 重复以上 (1)~(6) 的步骤。如满足误差要求, 则退出循环。如不满足, 再次执行 (7) 和 (8)。

采取这种随机数统计优化方法所获得的提取结果与初始设定值无关, 且运行时间较短, 不存在不收敛问题, 此外它不要求微分, 因而可适应和处理各种数学形式的模型方程。

第七章 电路模拟程序的内部结构和求解过程

§ 7.1 电路模拟程序的基本组成

第六章主要讨论了电路模拟的作用、模拟的内容以及器件模型和提取方法,但没有涉及电路模拟程序本身的结构和采用的算法。

电路模拟程序已开发并投入使用的很多。它们在功能和规模上虽有明显差别,但在结构上非常类似,一般包括五个部分,即输入处理、列方程、器件模型的处理、求解和输出。

输入处理是对程序所规定的“电路描述语言”(输入语言)进行编译,作词法和语法检查,建立合适的数据结构,把各类数据分类进行存储,如发现输入格式的书写上有错误,将打印出错信息。

列方程是根据输入的拓扑关系、元器件的参数以及分析要求建立电路方程。

器件模型处理是把电路中出现的各类元器件用相应的数学模型来描述。特别是对于非线性元器件如二极管、三极管、非线性电阻、非线性电容等要作线性化处理才能求解。

求解过程是对线性代数方程组、非线性代数方程组和常微分方程组求数值解。它是电路模拟程序的主要部分。

输出是产生用户所要求的列表、作图以及对程序的使用情况进行统计分析。

现以 SPICE 2 程序为例说明。它由根部和七个主要模块组成,程序约有一万多条 FORTRAN 语句和少量 C 语句。其组成示于图 7-1。

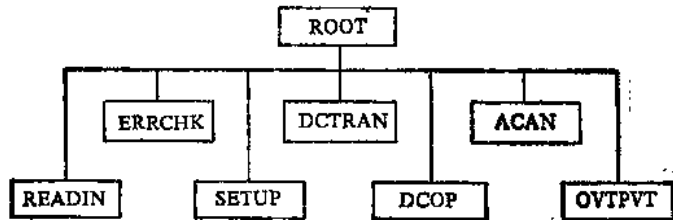


图 7-1 SPICE 2 程序内部组成

根是程序的主控部分,它按照不同的分析要求调用不同的模块。凡是要被一个以上模块所调用的子程序都包括在根之中。

READIN 模块包括 READIN、CARD、FIND、RUNCON 等四个主要的子程序。它读入输入文件,进行词法和语法分析和检查,分配内存并建立相应的链表结构。

ERRCHK 模块包括 ERRCHK、PUTNOD、GETNOD、SUBCKT、TOPCHK、MODCHK 等子程序。它对输入的初始数据进行预处理,建立内部节点有序表,处理信号源,建立信号源的断点表,进行子电路的展开,进行错误检查。如果发现错误,则退出 ERRCHK,并打印出错信息。

SETUP 模块是通过建立双链表来描述系数矩阵,并对系数矩阵进行行置换以消除奇

异问题。同时确定系数矩阵各元素的分配地址。

DCTRAN 模块是 SPICE2 中最大也是最复杂的模块。它完成直流工作点分析、瞬态初始条件分析、直流转移曲线分析和瞬态分析。

DCOP 模块打印非线性元件的工作点,打印线性化小信号模型参数,包括计算电路的功耗。

ACAN 模块进行交流小信号分析。

OVTPT 模块产生分析结果的各种列表和图形。

§ 7.2 电路模拟程序的流程

主程序的流程图如图 7-2。

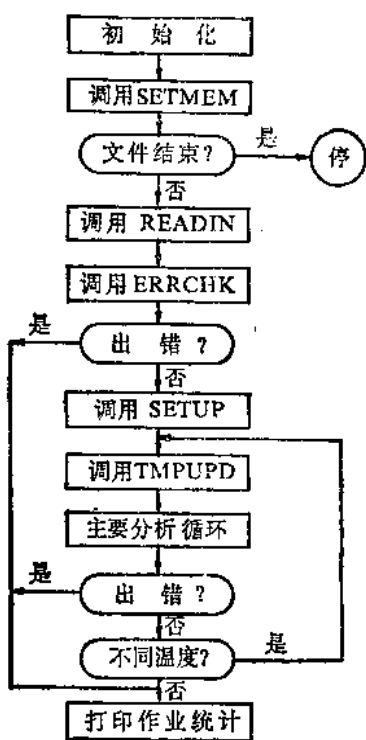


图 7-2 主程序流程图

SPICE2 可执行三种分析即直流分析、交流小信号分析和瞬态分析。分析类型由 MODE 标志。

MODE 分析模式:

直流分析 MODE=1,

瞬态分析 MODE=2,

交流小信号分析 MODE=3。

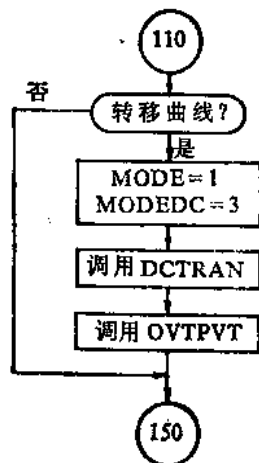


图 7-3 直流转移特性流程图

而直流分析又分成三种情况,由 MODEDC 标志。

MODEDC 直流分析类型:

直流工作点 MODEDC=1,

瞬态分析初始工作点 MODEDC=2,

直流转移曲线 MODEDC=3。

因而主要分析循环包括四部分:

(1) 直流转移特性 (MODE=1, MODEDC=3),

(2) 直流工作点 (MODE=1, MODEDC=1),

(3) 交流小信号分析 (MODE=3);

(4) 瞬态分析初始工作点 (MODE=1, MODEDC=2) 和瞬态分析 (MODE=2)。

它们的流程图分别示于图 7-3、图 7-4、图 7-5 和图 7-6。

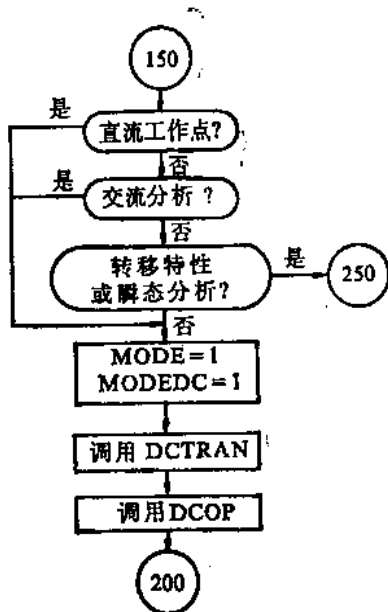


图 7-4 直流工作点流程图

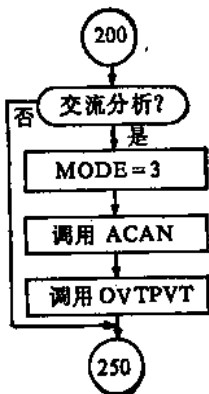


图 7-5 交流小信号流程图

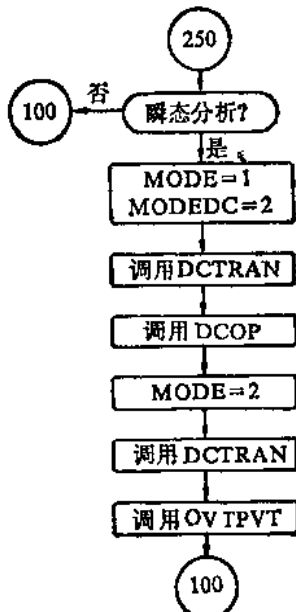


图 7-6 瞬态分析流程图

如果要求不同的分析温度，就再调用 TMPUPD 子程序，对与温度有关的参数进行修正后再次转入主分析循环。

§ 7.3 输入处理

SPICE2的输入格式是自由格式。每张输入卡有若干个字段，每个字段之间由界符隔开，多余的空格不计。如果下一张卡片的第一列上有“+”号，表示此卡是上一张卡片的继续行。

输入语言中允许有多种数的表示形式，有整数、浮点数、整数指数和比例因子等。

元器件卡的第一个字符必须是字母，这个字母作为关键字并对应于各元器件。模型卡、控制卡等其它卡片的第一个字符为“.”且关键字不是一个字符而是一串字符（各关键字见“SPICE 通用模拟程序用户指南”）。

这种自由格式对用户带来了方便，但使输入语言的编译工作量增大。

一、词法和语法分析

SPICE 2处理一张卡片时，先将卡片内容读入数组 AFIED 中。AFIED 为 10 个元素的双精度变量数组，每个字节存放一个字符，共能存放 80 个字符。然后对读入的内容进行词法分析，通过读出第一或第二个字符，确定这个字段的属性，拼写出名字或数字，并将有关信息即属性、界符、界符所在列数和内容分别存入 ICODE、IDELIM、ICOLUM、和 IFIELD 数组中。

如果是名字，在 ICODE 中存“1”，如果是数字则存“0”，如遇到卡片结束，在 ICODE

中存“-1”。界符只起分隔作用并无实际意义，因此只存放两个字段间所遇到的第一个界符，其余的略去。将第一界符存入 IDELIM，而该界符的所在列数存入 ICOLUM 中。卡片中每一字段的内容（名字或数字）需经过拼写再存入 IFIELD 中。

拼写名字比较简单。先扫描每一字段的前两个字符，如果第一个字符是字母，则当前字段为名字段，因此连续扫描接着的七个字符（多于七个则忽略）或扫描到出现界符为止。将拼写到的名字存入 IFIELD 中。如果扫描得到的第一个字符是“.”符，则接着扫描第二个字符，如果第二个字符为字母，说明当前字段也是名字段。这时将“.”连同其它字符一起存入 IFIELD，同样，八个字符以上的字符被忽略。

拼写数比较繁复，这里不作叙述。但应说明一点，在 SPICE2 中所有的数都被转换成双精度，而不管其原来是否为双精度型。

词法分析后紧接着对该卡进行语法分析，语法分析是按不同的卡片类型分别进行的。因为不同的卡片所允许的字段数不同，具有的名字不同，可选项的数目也不同，因而统一处理有困难，分别处理会加快速度。

为了区分卡片类型，每种卡片有一个类型标识码 ID。具有 20 个元素的 AIDE 数组中存放着元器件卡中第一字段的关键词。所以只要将每张元器件卡第一字段的第一字符与数组 AIDE 所存放的关键词进行对照，就可从数组元素下标得到该输入卡的 ID 值。在 AIDE 数组中存放的关键词与卡片类型的对应关系见表 7-1。对于其它卡（包括控制卡），则对照 AIDC 数组，因 AIDC 数组中存放着这些卡的关键词，其对应关系见表 7-2。同

表 7-1 AIDE 中卡片类型与 ID 的关系

卡片名	ID	关键词
电阻卡	1	R
电容卡	2	C
电感卡	3	L
互感卡	4	K
电压控制电流源卡	5	G
电压控制电压源卡	6	E
电流控制电流源卡	7	F
电流控制电压源卡	8	H
独立电压源卡	9	V
独立电流源卡	10	I
二极管卡	11	D
BTJ 卡	12	Q
JFET 卡	13	J
MOSFET 卡	14	M
缺	15	/
缺	16	/
传输线卡	17	T
缺	18	/
子电路调用卡	19	X
缺	20	/

表 7-2 AIDC 中卡片类型与 ID 的关系

ID	关键词
1	AC
2	DC
3	DISTORTI
4	END
5	ENDS
6	FOURIER
7	MODEL
8	NOISE
9	OP
10	OPTIONS
11	PLOT
12	PRINT
13	SUBCKT
14	SENSITIV
15	TRANSIEN
16	TF
17	TEMPERAT
18	WIDTH
19	NODESET
20	IC

样将输入卡“.”后面的字符与数组 AIDC 所存放的关键字进行对照,就从元素下标得到该输入卡的 ID 值。有了 ID 值,程序将根据不同的 ID 转到不同的子程序进行分析处理。

语法分析时扫描各字段,对各字段的内容进行辨认,分析是否符合该卡片的语法要求。在辨认出专用词汇后,将名字后面跟随的数字存入内存格式指定的单元中(名字本身不再存储)。如不符合语法要求,就认为出错并打印出错信息。

对控制卡,若发现错误,只是撤消该出错卡片,但仍继续运行。对控制卡以外的卡片,一旦发现错误将停止运行等待修改。

二、动态存储分配及用户区

电路模拟程序对输入数据在内存中的存储分配一般有两种方式:静态分配方式和动态分配方式。

所谓静态分配就是在数据输入前就已分配好各类数据在内存中应处的位置,而这些位置是固定不变的。采取这种方式的优点是数据的处理比较简单。譬如按这种方式,电阻和电容的数据就分别存放在不同的内存位置,只要读入电阻卡,其数据就存放在事先规定的某一区,而读入电容卡时,其数据就存放在另一事先规定的区域。静态分配的缺点是不能适应不同特点和不同规模的电路。为了增加适应性,需要开辟足够大的存储空间,但有时即使开辟很大的存储空间也无法满足要求。例如原开辟的用于存放电阻的数据区可存放 100 个电阻,但当某电路中的电阻数超过 100 个时,程序对该电路就无法进行运算,因为没有空间来存放第 101 个电阻及所有超越的电阻。

动态分配正是为了克服静态分配的弱点而提出的。它事先不规定各类元器件在内存中的存放位置,而是在读入过程中动态地加以确定。程序逐一扫描输入卡,每扫描一张卡片,就根据输入卡的类型计算出应分配的存储单元数,然后进行相应存储单元的分配。接着扫描下一输入卡并重复上述过程,直到所有的卡片都输入完毕。这样不管电路的特点和规模有多大的差别,输入数据在内存中总是占有一个连续的存储区,不但存储空间得到了充分的利用,也不会产生某个电路无法计算的情况。

SPICE2 中对输入数据就是采取这种动态方式。它开辟了若干公用区,其中一个公用区为 BLANK 区,专门用来存放描述电路的各种数据、计算分析时的中间结果和输出结果。该公用区中 VALUE 数组存放双精度数, NODPLC 数组存放整数, CVALUE 数组存放复型数。它们由 EQVALENCE 语句实现等价。VALUE 数组的大小可以根据所采用的计算机的内存规模具体加以确定,例如 2G5 版本中 VALUE 数组具有 20 万个双精度字。

虽然 BLANK 区中的 VALUE 数组很大,但程序一开始只向 VALUE 数组申请一千个双精度字,对应于 NODPLC 数组为两千个字的存储单元,这两千个存储单元称为用户区,见图 7-7。

为了存放各类数据,在用户区中又建立了十四个数据块区。各数据块的名称和使用如下:

- (1) IELMNT 块:存放各类元器件卡、模型卡以及各种输出变量的内部表。
- (2) ITEMPs 块:存放电路分析时的温度值。
- (3) ISBCKT 块:存放子电路定义卡的位置。
- (4) IUNSAT 块:存放模型名、子电路名及互感元件的电感名、电流控制电流(电压)源中的电压源名。

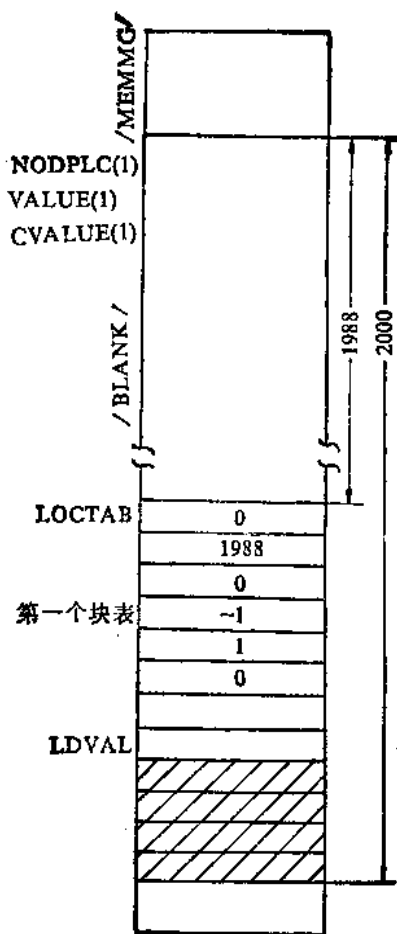


图 7-7 用户区初始存储

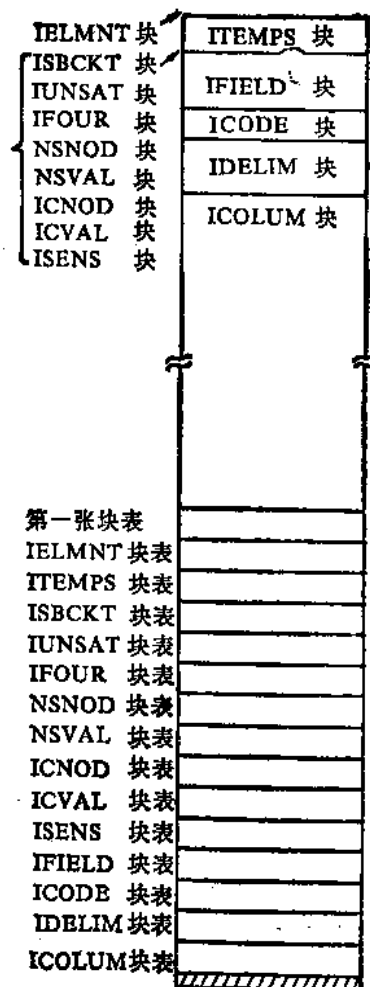


图 7-8 有数据块后的用户区

- (5) IFOUR 块: 存放富里叶分析时输出变量内部表的位置。
- (6) NSNOD 块: 存放 NODESET 卡的节点号。
- (7) NSVAL 块: 存放 NODESET 卡相应节点的电压值。
- (8) ICNOD 块: 存放 IC 卡的节点号。
- (9) ICVAL 块: 存放 IC 卡相应节点的电压值。
- (10) ISENS 块: 存放灵敏度分析时输出变量内部表的位置。
- (11) IFIELD 块: 存放卡片上各个字段的内容。
- (12) ICODE 块: 存放相应字段的属性。
- (13) IDELIM 块: 存放各字段间第一个定界符。
- (14) ICOLUM 块: 存放相应定界符所在的列数。

这十四个数据块中有五个一开始就确定了它们所占的字数。它们是 IFIELD、ICODE、IDELIM、ICOLUM 及 ITEMPS 块, 其所占字数分别为 100、52、100、52 和 4。这是因为在词法分析时就要把卡片中的内容和特征分别存入前四块。对 ITEMPS 块, 头两个字存

放温度的隐含值 27℃，因此也有必要事先确定。

其它九个数据区需要开辟多大，将根据输入卡的具体类型而定。一开始这些块的大小为零，即不占有内存空间。

在建立初始存储用户区时，在用户区底部建立起第一块表，它占八个字，LOCTAB 为块表入口地址，LDVAL 为出口地址。当在用户区中建立十四个数据块时，除了原有的第一块表外，还对应各数据块分别建立起自己的块表。块表中记录了各块有关存储的信息，以便进行动态管理。有数据块后的用户区见图 7-8。

块表中每个字的内容如下：

- 字 1 本块在 BLANK 区中的首地址（相对地址）；
- 字 2 本块可用字数；
- 字 3 本块实际占用字数；
- 字 4 本块指针的地址（绝对地址）；
- 字 5 本块中每一项所占字数；
- 字 6 溢出大小；
- 字 7 无用；
- 字 8 无用。

如果用户区的空闲空间已不能满足数据块扩充的需要，程序将向 NODPLC 数组再申请两千个字。因此用户区是以两千字的整数倍逐渐扩大。在扩充后，第一块表和各数据块的块表将再次移到新用户区的底部。

当某些数据块在以后的分析计算中不再需要时，这些数据块就被释放，释放出来的空间可用于其它数据块。例如词法分析所用的 IFIELD、ICODE、IDELIM、ICOLUM 块在所有输入卡都读完后就可释放。

从以上讨论我们可以体会到动态存储分配的优点。

三、存放格式和单向链表结构

每张卡片在内存的存放格式见图 7-9。

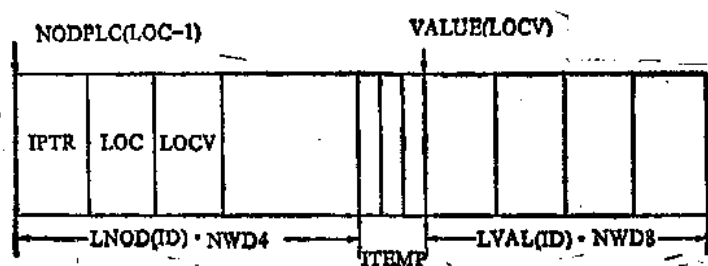


图 7-9 存放格式

内存格式由整型字（由 NODPLC 数组元素表示）和双精度字（由 VALUE 数组元素表示）两个区域组成。整型区用来存放指针（包括卡片名指针、链指针等）和节点号等。双精度区用来存放卡片名、数值以及可选项等。

每张卡片所占内存的大小由下式确定：

$$KTMP = LNOD(ID) \cdot NWD4 + LVAL(ID) \cdot NWD8 + ITEMP \quad (7-1)$$

式中, $NWD4=1$; $NWD8=2$; $LNOD(ID)$ 为存放整型变量的数目, $LVAL(ID)$ 为存放双精度变量的数目; $ITEMP$ 为整型区中整型字的调整量。对于不同的 ID , 可从表 7-3 中找到相应的 $LNOD$ 和 $LVAL$ 值。

表 7-3 ID 与 $LNOD$ 、 $LVAL$ 的关系

名 称	ID	LNOD (整型变量数)	LVAL (双精度型变量数)
电 阻	1	9	5
电 容	2	13	4
电 感	3	15	4
互 感	4	7	2
电压控制电流源	5	14	1
电压控制电压源	6	15	1
电流控制电流源	7	14	1
电流控制电压源	8	15	1
独立电压源	9	12	4
独立电流源	10	7	4
二 极 管	11	17	3
BTJ	12	37	4
JFET	13	26	4
MOSFET	14	34	16
传 输 线	17	34	9
子电路调用	19	5	1
子电路定义	20	5	1
二极管模型	21	4	19
BTJ 模型	22	4	55
JFET 模型	23	4	17
MOSFET 模型	24	4	46
直流打印	31	21	1
瞬态打印	32	21	1
交流打印	33	21	1
噪声打印	34	21	1
失真打印	35	21	1
直流绘图	36	21	17
瞬态绘图	37	21	17
交流绘图	38	21	17
噪声绘图	39	21	17
失真绘图	40	21	17
直流分析输出变量	41	8	1
瞬态分析输出变量	42	8	1
交流分析输出变量	43	8	1
噪声分析输出变量	44	8	1
失真分析输出变量	45	8	1

以电阻卡为例。因电阻卡的 $ID=1$ ，可查出 $LNOD(1)=9$ ， $LVAL(1)=5$ ，如再设 $ITEMP=3$ ，则 $KTMP=22$ ，即电阻卡占 22 个字，示于图 7-10。再如独立电压源，它的 $ID=9$ ，可查出 $LNOD(9)=12$ ， $LVAL(9)=4$ ，所以 $KTMP=20$ 。而独立源函数有五种，其函数代码为：

- 代码=1 脉冲源 代码=4 分段线性源
 代码=2 正弦源 代码=5 单频频率调制源
 代码=3 指数源

这五种源函数中的各参数则单独存放，其位置由表指针 t_p 指向。

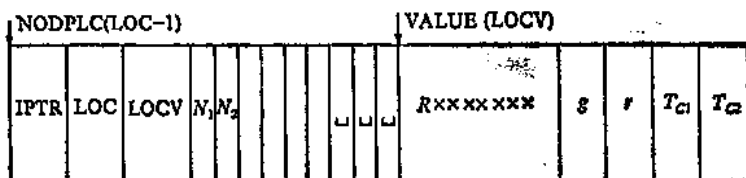
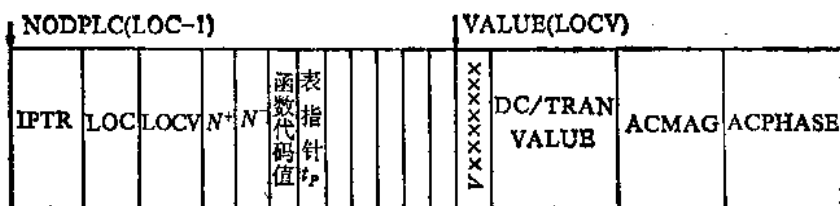
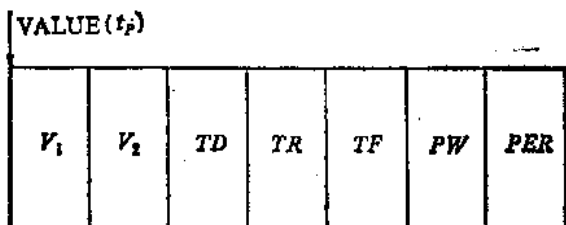


图 7-10 电阻卡存放格式

图 7-11(a) 是独立电压源的存放格式。图 7-11(b) 是源函数为脉冲源时，各参数的存放格式。



(a)



(b)

图 7-11 独立电压源及脉冲源各参数存放格式

为了便于寻找同一类卡片，SPICE2 采用了单向链表的数据结构，同类卡片形成一链。

具有 50 个元素的 $LOCATE(ID)$ 数组是链头位置的踪迹表。它记录了每一链表的链头位置，即每一链表中第一张卡片存放的地址。例如 $LOCATE(1)$ 中存放了第一张电阻卡的地址， $LOCATE(9)$ 中存放着第一张独立电压源卡的地址。而每张卡中有链指针 LOC ，它指向下一个同类型卡片的所在位置。每输入一张同类卡片，就把存放地址回填到上一个同类卡片的 LOC 单元中。当 $LOC=0$ 时，表示此卡是该链的链尾。链结构形式见图 7-12。

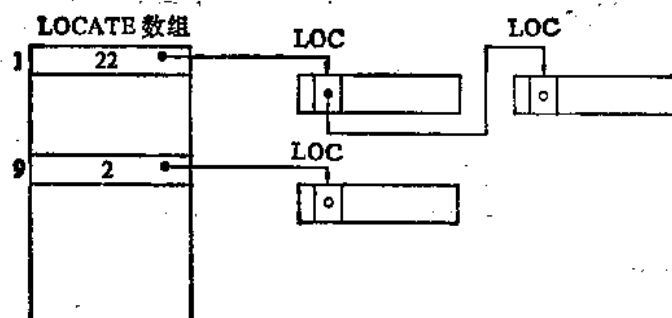


图 7-12 单向链结构

另有一数组 JELEN 记录每一链的长度，即同类卡片数目。因此每一张卡片在内存中的位置及相应存储单元中的内容就很容易寻找。

四、语法分析举例

现以电阻卡为例说明语法分析将如何进行。电阻卡的一般形式为

$R \times \times \times \times \times \times \times$ N1 N2 VALUE <TC=TC1<, TC2>> 按语法规则，此卡的节点数为两个，并处于第二和第三字段。第四字段应为数字且大于零。第五字段为可选项，且名字为 TC，TC 后面为数字，并允许有两个数字。如不给定，则该两值自动为零。

在求得该卡的 ID=1 后，根据 KTMP=22 将 IELMNT 块扩大 22 个字。

先将电阻名 $R \times \times \times \times \times \times \times$ 存入 VALUE(LOCV)。再判断该卡应有的节点数，按语法规则为两个，因此将第二字段、第三字段的数字作为节点号分别存入 NODPLC (LOC+2) 和 NODPLC (LOC+3) 单元中。接着检查第四字段是否为数字且是否大于零。如果是，则认为它是电阻值，存入 VALUE (LOCV+2) 中。如不是则打印出错信息。再检查第五字段是否有名字 TC，如有则将该名字后的两个数字分别存入 VALUE (LOCV+3) 和 VALUE(LOCV+4)。如名字不是 TC，说明有错就打印出错信息。遇到卡片结束或存完 TC 后的两个数字，就认为处理完毕。

五、节点的处理

SPICE2 的电路描述中规定节点号必须是正数，不能为负，但不必在数字上连续。程序中建立了一个存储块 JUNODE 用来存放用户给出的节点号。但为了以后处理时方便，程序将它们转换成内部节点号，它将是连续的自然数。

现以图 7-13(a) 为例。圆圈内的数字为用户规定的节点号，存放在 JUNODE 中的情况如图 7-13(b)。

这时将 JUNODE 存储块的排列序号作为内部节点号。因而 R_1 的内部节点号为 2 和 3， R_2 的内部节点号为 4 和 5， Q_1 的内部节点号为 4、3 和 1。这些内部节点号要回填到 NODPLC 数组中原来存放用户节点号的单元中去。

除了上述工作外，在输入处理阶段还要进行子电路的展开，产生子电路中各元器件

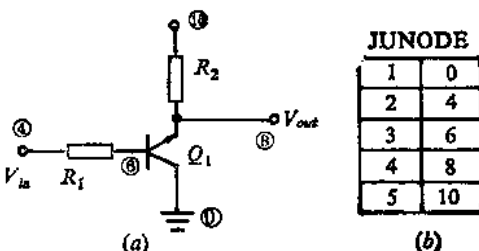


图 7-13 用户节点号与内部节点号

的全名,对瞬态分析时的独立电压源或电流源计算出各时间点的值(时刻值),进行拓扑关系检查,对一些元件值进行预处理(如求出电阻的倒数,存入 VALUE(LOCV+1) 单元)等等,在这里就不再详述。

§ 7.4 建立电路方程

一、改进节点法

列方程的方法很多,有节点法、改进节点法、状态变量法、拓扑矩阵法等。一般希望方程组容易建立和求解,方程式的数目越少越好。

在 SPICE 2 中采用改进节点法。节点法是以节点电位为未知量,对每个独立节点利用基尔霍夫定律列出方程组。但它不能表示以电压为变量的元件,如电感 (L)、传输线 (T)、独立电压源 (V)、电流控制电压源 (H)、电压控制电压源 (E)。我们常常把含有上述元件的支路称为“困难”支路。此外,节点法不能直接在节点方程中将未知的电流量作为输出变量。

改进节点法就是针对上述问题而提出的一种改进方法。它除了仍以节点电位作为未知量外,还可以将支路电流作为新的未知量。

用改进节点法建立起来的电路方程其矩阵形式如下:

$$\begin{bmatrix} Y_R & B \\ C & D \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} J \\ F \end{bmatrix} \quad (7-2)$$

式中, Y_R 为节点导纳矩阵的简化形式,但不考虑“困难”支路的贡献; B 、 C 、 D 为关系式矩阵; V 、 I 为未知节点电位和未知支路电流; J 、 F 为已知电流和已知电位矢量。 B 表示节点与“困难”支路之间的关系,且有

$$B_{ij} = \begin{cases} 0 & i \text{ 节点与 } j \text{ “困难”支路无关;} \\ 1 & i \text{ 节点是 } j \text{ “困难”支路的正节点;} \\ -1 & i \text{ 节点是 } j \text{ “困难”支路的负节点} \end{cases}$$

C 表示“困难”支路与节点之间的关系,且有

$$C_{ij} = \begin{cases} 0 & j \text{ 节点与 } i \text{ “困难”支路无关;} \\ 1 & j \text{ 节点是 } i \text{ “困难”支路的正节点;} \\ -1 & j \text{ 节点是 } i \text{ “困难”支路的负节点} \end{cases}$$

D 表示“困难”支路与“困难”支路间的关系, d_{ij} 为 i “困难”支路与 j “困难”支路之间的函数关系值。

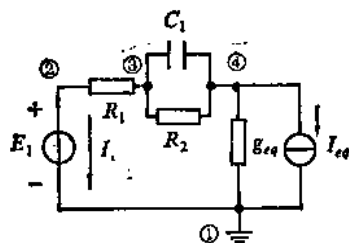


图 7-14 电路图

现以图 7-14 电路为例说明如何应用改进节点法。

整个电路共有四个节点,一个“困难”支路,所以方程个数 NSTOP 为

$$\begin{aligned} \text{NSTOP} = & \text{NUMNOD} + \text{JELCNT}(3) \\ & + \text{JELCNT}(6) + \text{JELCNT}(8) \\ & + \text{JELCNT}(9) + 2\text{JELCNT}(17) \end{aligned} \quad (7-3)$$

式中, NUMNOD 为节点总数;

JELCNT 依次为电感、电压控制电压源、电流控制电压源、独立电压源和传输线的个数。

由于节点①是接地点, V_1 总是为零, 所以实际有效的系数矩阵单元数为 $(NSTOP - 1) \times (NSTOP - 1)$ 。

根据基尔霍夫定律可以求出上列电路的方程组如下:

$$\begin{cases} \frac{1}{R_1} V_2 - \frac{1}{R_1} V_3 + I_1 & = 0 \\ -\frac{1}{R_1} V_2 + \left(\frac{1}{R_1} + \frac{1}{R_2} + C_1 \frac{d}{dt} \right) V_3 - \left(\frac{1}{R_2} + C_1 \frac{d}{dt} \right) V_4 & = 0 \\ -\left(\frac{1}{R_2} + C_1 \frac{d}{dt} \right) V_3 + \left(g_{eq} + \frac{1}{R_2} + C_1 \frac{d}{dt} \right) V_4 & = -I_{eq} \\ V_2 & = E_1 \end{cases} \quad (7-4)$$

其矩阵形式为

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 \\ -\frac{1}{R_1} & \left(\frac{1}{R_1} + \frac{1}{R_2} + C_1 \frac{d}{dt} \right) & -\left(\frac{1}{R_2} + C_1 \frac{d}{dt} \right) & 0 \\ 0 & -\left(\frac{1}{R_2} + C_1 \frac{d}{dt} \right) & \left(g_{eq} + \frac{1}{R_2} + C_1 \frac{d}{dt} \right) & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_2 \\ V_3 \\ V_4 \\ I_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -I_{eq} \\ E_1 \end{bmatrix} \quad (7-5)$$

可以看出, 一个内阻为零的独立电压源用节点法无法求得该支路上的电流, 但采用改进节点法后就可把该支路电流作为未知量。这样独立电压源就可以很方便地通过支路电流和节点电位这两个方程式来表示。

二、稀疏矩阵与双链表结构

在多数电路中与某一节点有关联的节点数毕竟是少数, 因此列出的系数矩阵中有不少零元素, 而且随着电路规模的加大, 方程数目的增加, 矩阵中的零元素越来越多, 因而系数矩阵的稀疏性加大。为了节省内存空间, 我们没有必要存储这些零元素。所以通常采用稀疏矩阵存储技术, 只存储矩阵中的非零元素。

SPICE2 中采用了双链表数据结构来描述系数矩阵。双链表记录了非零元素的位置指针以便于查找、插入和互换。系数矩阵中各元素的值存于 LVN 数组中。

双链表由 IRPT、JCPT、IROWNO、JCOLNO 四个数组组成, 分别称为列链表、行链表、行号表和列号表。各数组元素的内容为:

(1) IRPT 的内容, 即 $NODPLC(IRPT + I)$ 的内容

当 $I \leq NSTOP$ 时, 存放第 I 列第 1 个非零元素的位置指针;

当 $I > NSTOP$ 时, 存放同一列中下一个非零元素的位置指针。

(2) JCPT 的内容, 即 $NODPLC(JCPT + I)$ 的内容

当 $I \leq NSTOP$ 时, 存放第 I 行第 1 个非零元素的位置指针。

当 $I > NSTOP$ 时, 存放同一行中下一个非零元素的位置指针。

(3) IROWNO 的内容, 即 NODPLC(IROWNO+I) 的内容

当 $I \leq \text{NSTOP}$ 时, 无用;

当 $I > \text{NSTOP}$ 时, 存放位置指针 I 的非零元素的行号。

(4) JCOLNO 的内容, 即 NODPLC(JCOLNO+I) 的内容

当 $I \leq \text{NSTOP}$ 时, 无用;

当 $I > \text{NSTOP}$ 时, 存放位置指针 I 的非零元素的列号。

图 7-15 表示一稀疏矩阵 (它对应于公式 7-5)。根据以上规定, 可得到双链表如表 7-4。

	2列	3列	4列	5列
2行	x	x		x
3行	x	x	x	
4行		x	x	
5行	x			

图 7-15 稀疏矩阵

对于 IRPT 和 JCPT 数组, 其内容为零时表示它是该列或该行的最后一个非零元素。

双链表的含义可以通过图 7-16 加以说明。对于行链表 JCPT, 总是同一行的前一列非零元素指向下一列非零元素。对于列链表 IRPT, 总是同一列的前一行非零元素指向下一行非零元素。可以看出, 采取这种双链表结构后, 可以很快搜索到矩阵中的某一

元素。通过 JCPT 可直接得到与此元素同一行的下一个非零元素, 并通过 JCOLNO 表知道其所在列号。同样通过 IRPT 可直接得到与此元素同一列的下一个非零元素, 并通过 IROWNO 表知道它所在的行号。

表 7-4 双链表

INDEX	IRPT	IROWNO	JCOLNO	JCPT
1	0	0	0	0
2	6	0	0	6
3	7	0	0	8
4	10	0	0	14
5	13	0	0	14
6	8	2	2	7
7	9	2	3	13
8	14	3	2	9
9	11	3	3	10
10	12	3	4	0
11	0	4	3	12
12	0	4	4	0
13	0	2	5	0
14	0	5	2	0

三、系数矩阵中各导纳值的存放

前面已谈到, 双链表只是描述系数矩阵中非零元素的分布情况。各非零元素的值则存放在另外开辟的 LVN 数组中。对于交流分析由于是复数运算, 因而实部存入 LVN 数组,

虚部则存入 IMVN 数组。

现以图 7-14 的电路图和式(7-5) 矩阵为例说明如何填入。

程序按 LOCATE 数组中 ID 的顺序逐个找出 LOC，再根据 LOC 从内存中取出元件的节点号。因而总是从电阻开始，如电路图中无电阻，则跳过而寻找电感的 LOC，并依此类推。

现先取 R_1 ，其节点号(指内部节点号)为 2 和 3。根据改进节点法， R_1 对导纳矩阵的贡献有四项，即矩阵中的 (2, 2)、(2, 3)、(3, 2)、(3, 3) 元素，因此把 R_1 的导纳 $1/R_1$ 、 $-1/R_1$ 、 $-1/R_1$ 、 $1/R_1$ 分别填入系数矩阵的这四项中。

从双链表查出这四项对应的 INDEX 分别为 6、7、8、9。因而按同样的 INDEX 把以上正负导纳填入 LVN 数组。

接着取 R_2 ，其节点号为 3 和 4，它对导纳矩阵的贡献同样有四项，即 (3, 3)、(3, 4)、(4, 3)、(4, 4)。从双链表查出这四项对应的 INDEX 分别为 9、10、11、12。因而 R_2 的正负导纳将分别填入 LVN(9)、LVN(10)、LVN(11) 和 LVN(12)。

在处理完电阻链后，按同样方法处理其它各链。

对于独立电压源，除取其正负节点外，还需求出“困难”支路号 IBR 。并从双链表中查出 $(N1, IBR)$ 、 $(N2, IBR)$ 、 $(IBR, N1)$ 、 $(IBR, N2)$ 所对应的 INDEX。再将系数矩阵中有关的 B_{ij} 、 C_{ij} 或 D_{ij} 代入 LVN 数组。在本例中，独立电压源的正负节点号为 2 和 1，“困难”支路号 IBR 为 5。从双链表中查出 (2, 5) 和 (5, 2) 所对应的 INDEX 为 13 和 14，因此将 1 分别送入 LVN(13) 和 LVN(14)。

方程组右端矢量一般存放在 LVN 数组的最初几个存储单元中(除 LVN(1) 单元外)。对本例即存入 LVN(2) 至 LVN(5) 单元。

经以上处理后，LVN 数组中的内容示于表 7-5。

四、消除矩阵中的奇异问题

当系数矩阵主对角线上有零元素(严格地为零)时则在求解时将导致失败，这称为奇异问题。

为了解决这个问题，现采取行行交换。如对方 7-15 这样的矩阵，就需要将第二行

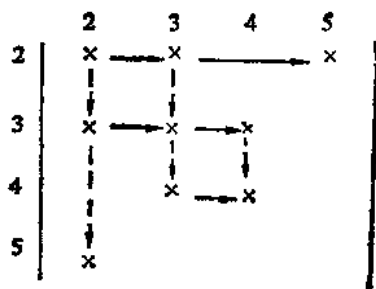


图 7-16 双链的含义

表 7-5 LVN 数组内容

INDEX	VALUE
1	
2	
3	
4	
5	
6	$\frac{1}{R_1}$
7	$-\frac{1}{R_1}$
8	$-\frac{1}{R_1}$
9	$\frac{1}{R_1} + \frac{1}{R_2} + C_1 \frac{d}{dt}$
10	$-\frac{1}{R_2} - C_1 \frac{d}{dt}$
11	$-\frac{1}{R_2} - C_1 \frac{d}{dt}$
12	$g_{eq} + \frac{1}{R_2} + C_1 \frac{d}{dt}$
13	1
14	1

与第五行进行交换，使主对角线上的元素全为非零。

行与行进行交换的原则如下：

将电流未知量所在行的方程与此“困难”支路相连的节点电压所在行进行交换。节点必须满足以下条件：

(1) 不接地；(2) 主对角线元素不为零，即该节点有导纳项；(3) 只连接一个电压源。如正负节点都符合上述条件则取负节点。

现以图 7-17 的电路为例说明。

节点 1 接地显然不符合。节点 2 有两个电压源同样不符合。而节点 3 只连接一个电压源且该节点有导纳项，因此符合上述条件。

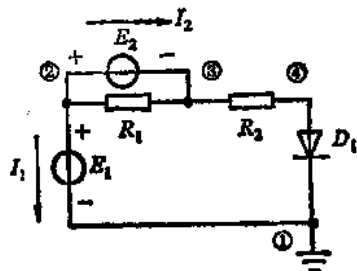


图 7-17 电路图 (多电压源)

所以将节点 3 所在行与 I_2 所在行交换。交换后就认为节点 2 只连接一个电压源，这时节点 2 又符合条件，就再进行节点 2 所在行与 I_1 所在行的交换。如果电路中有多个“困难”支路，可从最大“困难”支路号的支路开始进行交换。

对于图 7-15 的矩阵，在进行第二行与第五行交换以后，双链表将修改成表 7-6。

表 7-6 修改后的双链表

INDEX	IRPT	IROWNO	JCOLNO	JCPT
1	0	0	0	0
2	14	0	0	14
3	9	0	0	8
4	10	0	0	11
5	13	0	0	6
6	0	5	2	7
7	0	5	3	13
8	6	3	2	9
9	11	3	3	10
10	12	3	4	0
11	7	4	3	12
12	0	4	4	0
13	0	5	6	0
14	8	2	2	0

SPICE2 中还建立了 IRSWPF、IRSWPR、ICSWPF、ICSWPR 这四个数组来记录互换前后的对应关系。IRSWPF 和 ICSWPF 记录在原始矩阵中的位置，而 IRSWPR 和 ICSWPR 记录在当前矩阵中的位置。

$IRSWPF(I)=K$ 表示当前第 I 行是原始的第 K 行；

$ICSWPF(J)=L$ 表示当前第 J 列是原始的第 L 列；

$IRSWPR(K)=I$ 表示原始的第 K 行是当前第 I 行；

$ICSWPR(L)=J$ 表示原始的第 L 列是当前第 J 列
交换前和交换后各数组的内容见表 7-7。

表 7-7 交换前和交换后数组内容

	数 组 名	IRSWPF	IRSWPR	ICSWPF	ICSWPR
	序 号				
交 换 前	1	1	1	1	1
	2	2	2	2	2
	3	3	3	3	3
	4	4	4	4	4
	5	5	5	5	5
交 换 后	1	1	1	1	1
	2	5	5	2	2
	3	3	3	3	3
	4	4	4	4	4
	5	2	2	5	5

因此如知道某元素在原始矩阵中的位置,现需知道它在当前矩阵中的位置则查IRSWPR和ICSWPR。例如已知原始矩阵中有(2,3)位置的元素,从IRSWPR(2)=5和ICSWPR(3)=3可得到该元素在当前矩阵中的位置为(5,3)。反之,若知道某元素在当前矩阵中的位置,而想知道它在原始矩阵中的位置则查IRSWPF和ICSWPF。

应注意,行行交换过程中,LVN数组中的内容不变。所以取某元素的值时必须根据新的位置指针,才能从LVN数组中得到正确的值。

§ 7.5 求解方法和过程

电路模拟程序所分析的电路有线性电路和非线性电路,而分析的内容有直流、瞬态和交流的不同,因此在求解方法上就存在着差别。

对于线性电路的直流分析和交流分析所列出的方程组是实系数或复系数线性代数方程组。对于非线性电路的直流分析或交流分析则列出实系数或复系数非线性代数方程组。对于瞬态分析,所建立的电路方程组是常微分方程组。

因此一般模拟程序包括三类数值解:线性代数方程组求解,非线性代数方程组求解和常微分方程组求解。

现分别对线性电路的直流分析、非线性电路的直流分析以及线性和非线性电路的交流分析和瞬态分析的求解方法和过程讨论如下。

7.5.1 线性电路的直流分析

这时电路中只含线性元件,且电路中的电容开路,电感短路。所有的时间求导为零。

这时的电路方程组是线性代数方程组。

通常采用高斯消去法或 LU 分解法进行求解 (详见附录 I-3)。

一、填入项和马柯维兹判据

在消去法和 LU 分解法的运算过程中, 原始矩阵元素为零的元素有可能变为非零, 这新产生的非零项称为“填入”项。产生“填入”项会失去系数矩阵的稀疏性, 同时带来处理上的麻烦, 即要把“填入”项再插入到双链表结构中。为此应设法尽量减少“填入”项。

从正消算法的公式 (I-33) (见附录 I-3) 可以看出“填入”项产生的原因。(I-33) 式为

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}$$

如原来 $a_{ij}^{(k)}$ 为零, 但当 $a_{ik}^{(k)}$ 和 $a_{kj}^{(k)}$ 都不等于零时, 则第 $k+1$ 次消去后的 $a_{ij}^{(k+1)}$ 变为非零, 这就是所谓的“填入”项。而它又是通过乘除运算得来的, 通常乘除运算所用时间在四则运算中最长, 因而称为长运算 (或长操作)。

“填入”项不仅增加存储量, 需要改变稀疏矩阵存储的双链表, 而且还增加了长运算 (或长操作), 因此应尽量减少它的产生。而“填入”项的多少是与矩阵中非零元素的分布状况有关, 也就是说, 对一个稀疏矩阵可能存在一种最优的行列次序, 使得高斯消去或 LU 分解过程中产生的“填入”项最少。

马柯维兹判据是一种启发性算法。通过它可减少“填入”项。这种算法的基础是: 若选取某个矩阵元素 a_{ij} 为主元, 这时可能产生的最多“填入”项个数是 $(r_i - 1)(c_j - 1)$ 。这里 r_i 是第 i 行中的非零元素个数, c_j 是第 j 列中的非零元素个数。

马柯维兹判据在选取主元时应使 $(r_i - 1)(c_j - 1)$ 的值最小。现以矩阵 (7-6) 为例说明。对矩阵 (7-6) 应选 a_{21} 为主元, 因为 $(r_i - 1)(c_j - 1) = (2 - 1)(2 - 1) = 1$, 在所有非零元素的对应值中, 此值为最小值。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (7-6)$$

在选定 a_{21} 为主元后, 整个方程组要重新排列 (行置换时, 所求未知量的先后次序不变; 列置换时, 未知量先后次序跟着置换)。经重新排列后得

$$\begin{bmatrix} a_{21} & a_{22} & 0 & 0 \\ a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{32} & a_{33} & 0 \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_2 \\ b_1 \\ b_3 \\ b_4 \end{bmatrix} \quad (7-7)$$

再经一次消去, 系数矩阵变为下式

$$\begin{bmatrix} a_{12}^{(2)} & a_{13}^{(2)} & a_{14}^{(2)} \\ a_{22}^{(2)} & a_{33}^{(2)} & 0 \\ a_{42}^{(2)} & a_{43}^{(2)} & a_{44}^{(2)} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1^{(2)} \\ b_3^{(2)} \\ b_4^{(2)} \end{bmatrix} \quad (7-8)$$

这时再次选取主元, 但发现 $a_{12}^{(2)}$, $a_{33}^{(2)}$, $a_{44}^{(2)}$ 这三个元素具有相同的 $(r_i - 1)(c_j - 1)$ 值。遇到这种情况时, SPICE2 选取 $(c_j - 1)$ 最小的为主元。在此例中, 从 $a_{12}^{(2)}$ 或 $a_{44}^{(2)}$ 中任

选一个。若选 $a_{11}^{(3)}$ 为主元, 则矩阵变成下式

$$\begin{pmatrix} a_{11}^{(3)} & a_{12}^{(3)} & a_{13}^{(3)} \\ 0 & a_{22}^{(3)} & a_{23}^{(3)} \\ a_{31}^{(3)} & a_{32}^{(3)} & a_{33}^{(3)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1^{(3)} \\ b_2^{(3)} \\ b_3^{(3)} \end{pmatrix} \quad (7-9)$$

二、主元选取和阈值主元法

在消去和 LU 分解过程中, 都要将矩阵中同列各元素除以主对角线元 a_{kk} 。如果 a_{kk} 很小, 由于机器字长的限制以及计算时舍入误差的引入, 可能使结果带来很大误差。而在电子电路中各元件的电导值会相差很大, 可能有 $10^3 \sim 10^{-5}$ 的数量级变化, 这样会使求解结果与精确值相差甚远。当然这可以通过选取字长较长的计算机或采用双精度运算来解决。另一途径就是选取绝对值较大的元素作为主元, 这称之为选主元。

选主元的方法有列主元法、行主元法和全主元法。列主元法是在 k 列中选出绝对值最大者作为主元。行主元法是在 k 行中选取绝对值最大者。全主元法则在 k 行、 k 列右下部的子矩阵的所有元素中选最大者作为主元。就数值计算的稳定性看, 全主元法比部分主元法优越, 但计算工作量较大。

在 SPICE 2 中采用了折衷的办法。

(1) 先在第二列 (实际上就是第一列) 中选取一个最大值 MAXVAL, 并令

$$\text{EPSREL} = \text{PIVREL} \times \text{MAXVAL}$$

凡大于等于 EPSREL 的值都可允许作为主元。这种方法称为阈值主元法或数值主元法。PIVREL 的大小可以通过可选项卡加以重置, 其隐含值为 10^{-3} 。但所选主元值也不能小于预定的主元绝对最小值 PIVTOL, 其隐含值为 10^{-13} , 它也可通过可选项卡加以重置。所以

$$\text{EPSREL} = \text{DMAX1}(\text{PIVREL} \times \text{MAXVAL}, \text{PIVTOL}) \quad (7-10)$$

(2) 再在主对角线上寻找大于等于 EPSREL 的元素, 因通常主对角线上的元素值较大。并在可选的主元中采用马柯维兹判据, 对乘积最小的就确定其为主元。如在主对角线上找不到, 就在矩阵下三角中寻找, 这是因为稀疏矩阵具有对称性这一特点, 所以不必在全矩阵中寻找。

因此主元的选取原则有两个: 一是它必须大于某一值, 二是对该元素采用马柯维兹判据所得的乘积最小。

选定主元后, 需要进行行列置换, 相应地修改双链表和 IRSWPF, IRSWPR, ICSWPF, ICSWPR 数组。

综上所述, 线性电路的直流分析在建立了电路方程以后有以下步骤:

(1) 根据阈值主元法和马柯维兹判据选择主元。

(2) 根据被选主元的现有位置进行矩阵的行列置换。对双链表的内容进行修改, 记录互换前后位置的对应关系。

(3) 对系数矩阵进行高斯消去或 LU 分解。如产生“填入”项则要扩充双链表, 更改链指针并修改非零元素的个数。

(4) 根据高斯消去法或 LU 分解法的有关公式求出未知量 x 。

对线性直流分析, 只要求解一次即可得到未知节点电位和未知支路电流。

7.5.2 非线性电路的直流分析

这时电路中含有非线性元件。但电容仍开路，电感保持短路，所有的时间求导为零。因此求解的是一个非线性代数方程组。

求解非线性方程的基本方法是在一个初始的“猜解”处对非线性元件进行线性化近似，这样就可得出一组线性化的方程组，然后采用求线性代数方程组的同样方法对其求解，但第一次求得的解并不是真正的解。因而要在第一次求得的解处对元件再一次作线性化近似，列出新的线性化方程组重新求解。如果第二次得到的解与第一次求得的解不在规定的误差范围内，则再次重复上述过程（重复此过程称为迭代），直到前后两次解在规定的误差范围内为止。这时最后一次的迭代解即为最终解，达到了收敛。

所以非线性电路的直流分析与线性电路的直流分析有两点差别：

(1) 先要求出非线性元件的线性化小信号等效模型及相应的参数值。

(2) 列出线性化代数方程组，并通过多次迭代求解。

对于非线性元件，在不同的电压值时其线性化模型参数的数值会发生变化，所以每次赋入系数矩阵的值也随着变化。但由于电路的拓扑关系并没有改变，因而描述电路的系数矩阵结构并不变化。只是主元值要重新检查是否符合要求，如不符合则要重新选取。

求解非线性方程的方法很多。在 SPICE 2 中采用牛顿-拉夫森法（详见附录 I-2）。

牛顿-拉夫森法的迭代公式为

$$J(x^k)(x^{k+1} - x^k) = -F(x^k) \quad (I-20)$$

每进行一次迭代，需要作以下运算：

(1) 计算 $F(x^k)$ ；

(2) 计算 $J(x^k)$ ；

(3) 求解上列系数矩阵为 $J(x^k)$ 、右端矢量为 $-F(x^k)$ 的线性代数方程组；

(4) 得到解 x^{k+1} 。

这时一个非线性方程组的求解问题就转化成熟悉的线性方程组的求解问题。

现讨论求解时的收敛问题和直流伴随模型。

一、收敛要求

利用迭代法求解，最终解的获得以下式为判据

$$|x^{k+1} - x^k| < \varepsilon_a + \varepsilon_r \max\{|x^k|, |x^{k+1}|\} \quad (7-11)$$

式中， ε_a 为绝对误差， ε_r 为相对误差。求解电压时， ε_a 即 SPICE2 可选项卡中的 VNTOL，其隐含值为 $1\mu\text{V}$ 。求解电流时， ε_a 为可选项卡中的 ABSTOL，隐含值为 1pA 。 ε_r 为相对误差，即 SPICE2 可选项卡中的 RELTOL，其隐含值为 0.1% 。这些误差要求可以通过可选项卡加以重置。

SPICE2 不仅要求每个节点电位都满足 (7-11) 公式，而且要求所有非线性元件的支路电流也都满足该式，因为只是节点电位满足是不充分的。

二、帮助收敛的几种方法

对 SPICE2 来讲，在有些情况下会发生求解不收敛的现象。这时，程序将打印出最后一次迭代的节点电压，并终止作业。这时打印出来的电压未必正确，甚至未必接近正确的解。这种情况在双稳态电路、无稳态电路、再生开关电路、具有正反馈的电路中可能遇到。

为此, SPICE2 中提供了几种方法来帮助收敛。

1. 采用.NODESET 卡

用户可以设定电路中某些节点以致全部节点的初始电压。在开始迭代时, SPICE2 将首先在这些节点上连接一个等于.NODESET 卡设定电压值的电压源, 然后进行迭代求解直到收敛。然后去掉这些电压源, 这时将存在这些电压源时所获得的收敛解作为迭代初值, 再进行迭代直到求得真正的解。所以.NODESET 卡设置的电压值并不影响最终的直流解而仅起帮助收敛的作用。

2. 采用 OFF 可选项

求直流解时首先假定这些元件是截止的。在收敛求得解后再取消 OFF, 并以取消前的收敛解作为继续迭代的初始值。所以 OFF 的设定并不影响最终解。

有 OFF 项时, 元件处于截止状态, 相当于不存在此元件, 见图 7-18。

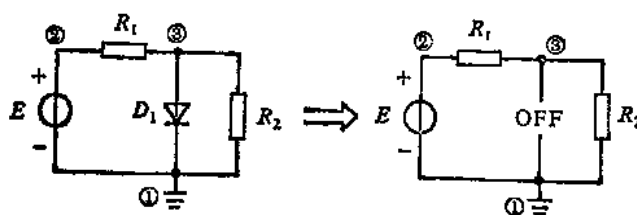


图 7-18 有 OFF 项时的等效电路

3. 在器件卡中设定初始条件可选项

这时在第一次迭代求直流解时, 将以这些设定的器件工作点作为初始值。但在完成第一次迭代以后, 这些设定值就不再起作用。这种方法对求解的收敛过程的约束力最弱。

4. 采用.IC 卡, 但.TRAN 卡中无 UIC 项

用户可在.IC 卡中设定某些节点或全部节点的初始电压值。SPICE2 将在这些节点上连接等于初始电压值的电压源, 然后迭代求解直至收敛。这时的收敛值就作为直流分析的最终解。这点是与.NODESET 卡根本不同的。在以后的瞬态分析中再去掉这些电压源, 并以这个解作为瞬态分析的初始条件。由于.IC 卡中设置的初始电压值将影响到最终解, 因此用户必须特别小心地加以规定。

5. 在采用.IC 卡的同时.TRAN 卡中有 UIC 项

UIC 表示用户不要求在瞬态分析前先求解静态工作点。这时 SPICE2 将直接根据.IC 卡的节点电压求出各元件上的端电压并作为瞬态分析时第一次迭代的初始条件。由于瞬态分析前不再计算直流偏置解, 所以要非常小心地设定.IC 卡上的初始值。正由于它们就是直流偏置点, 因此要作出正确的设定往往是很困难的。

三、元件的直流伴随模型

在求解非线性方程时, 如对非线性元件采用直流伴随模型将使求解过程更为直观。现以二极管为例。

二极管的非线性特性方程为

$$i = I_s [e^{qV/(kT)} - 1]$$

将它在 k 次迭代点 V^k 作台劳展开, 并略去高次项, 得

$$\begin{aligned}
i^{k+1} &\approx I_s [e^{qV^k/(kT)} - 1] + \left. \frac{\partial i}{\partial V} \right|_{V^k} (V^{k+1} - V^k) \\
&= I_s [e^{qV^k/(kT)} - 1] + \frac{qI_s}{kT} e^{qV^k/(kT)} (V^{k+1} - V^k) \\
&= I_s [e^{qV^k/(kT)} - 1] - \frac{qI_s}{kT} e^{qV^k/(kT)} V^k + \frac{qI_s}{kT} e^{qV^k/(kT)} V^{k+1} \\
&= I^k + G^k V^{k+1}
\end{aligned} \tag{7-12}$$

式中

$$I^k = I_s [e^{qV^k/(kT)} - 1] - \frac{qI_s}{kT} e^{qV^k/(kT)} V^k \tag{7-13}$$

$$G^k = \frac{qI_s}{kT} e^{qV^k/(kT)} \tag{7-14}$$

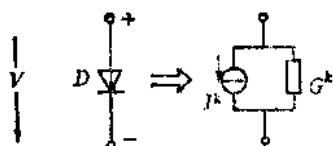


图 7-19 二极管的直流伴随模型

这样，一个在第 k 次迭代时的二极管，可用电流源 I^k 和电导 G^k 的并联来表示。如图 7-19。

如有图 7-20 这样包括二极管 D_2 和 D_4 的电路，现可利用二极管直流伴随模型得到其等效电路，并直接得到电路方程组。等效电路示于图 7-21。其电路方程组为

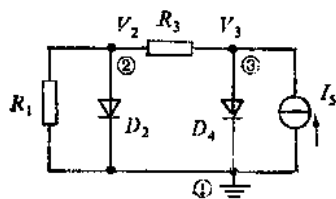


图 7-20 电路图

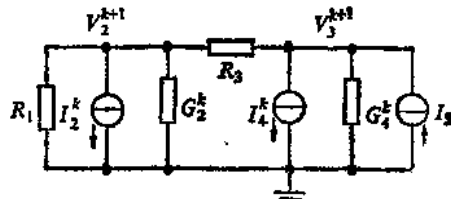


图 7-21 有二极直流伴随模型的等效电路图

$$\begin{cases} \left(\frac{1}{R_1} + \frac{1}{R_3} \right) V_2^{k+1} + G_4^k V_3^{k+1} - \frac{1}{R_3} V_3^{k+1} = -I_2^k \\ -\frac{1}{R_3} V_2^{k+1} + G_4^k V_3^{k+1} + \frac{1}{R_3} V_3^{k+1} = -I_4^k + I_5 \end{cases} \tag{7-15}$$

对于其它非线性元件如双极型晶体管和 MOS 晶体管等也可求出相应的伴随模型。

7.5.3 交流分析

在交流时，电路中除电阻值是实数外，所有的导纳系数、电压变量和电流变量都是复数。因此所建立的电路方程组是复系数线性或非线性方程组。如电容的导纳为 $j\omega C$ ，电感的导纳为 $1/(j\omega L)$ ，电压 $v = v_0 \sin(\omega t + \phi_1) = v_0 e^{j\omega t + \phi_1}$ ，电流 $i = i_0 \sin(\omega t + \phi_2) = i_0 e^{j\omega t + \phi_2}$ ，它们都是频率 ω 的函数。

交流分析时，SPICE2 首先计算直流工作点（对线性电路只要一次求解，对非线性电路需要多次迭代）。然后确定在此直流工作点时所有非线性元件的线性化小信号模型参数。

由于电路中的非线性元件已等效成交流小信号的线性模型，所以对这样的一个“线性”电路，给定一个频率值后只要一次求解就可完成。对于不同的频率，系数矩阵及右端矢量值将作相应的变化，但同样只要一次求解就可得出另一频率时的解。与直流求解时的不同点有：

一、主元的选择

这时采取简单的办法。程序将矩阵主对角线第一元素的实部与虚部分别取模并相加得到 GDIAG，将它与 PIVTOL 相比。如大于 PIVTOL，就取其为主元，否则就以 PIVTOL 为主元。

二、复型运算

交流分析时所有数值计算都要按照复型运算，即实部与实部相加减，虚部与虚部相加减。乘运算为

$$(a_1 + ib_1)(a_2 + ib_2) = (a_1a_2 - b_1b_2) + i(a_1b_2 + a_2b_1)$$

除运算为

$$\frac{a_1 + ib_1}{a_2 + ib_2} = \frac{a_1a_2 + b_1b_2}{a_2^2 + b_2^2} + i\frac{a_2b_1 - a_1b_2}{a_2^2 + b_2^2}$$

在求得起始频率时的解后，根据交流卡的三种类型 DEC、OCT、LIN 求出新频率值，再修改系数矩阵和右端矢量值，然后求出新解直到终止频率值为止。

因此，对于交流分析，在求得直流工作点和非线性元件的交流小信号线性模型后，求解过程并不复杂。

7.5.4 瞬态分析

瞬态分析要求对一个非线性微分方程组进行求解。方程式的一般形式如下：

$$\begin{cases} F(x, \dot{x}, t) = 0 \\ x(t_0) = x_0 \end{cases} \quad t_0 \leq t \leq T \quad (7-16)$$

式中， x 为未知变量， $\dot{x}$ 为 x 的导数， t 为时间。

一、数值积分法

对这样的非线性微分方程通常采用数值积分法。电路模拟程序中广泛使用的积分方法称线性多步法。其关系式为

$$y_{n+1} = \sum_{i=0}^p a_i y_{n-i} + h_{n+1} \sum_{i=-1}^p b_i \dot{y}_{n-i} \quad (7-17)$$

它是根据前面 $p+1$ 个时间点上求得的 y 和 $\dot{y}$ 值，计算出新时间点 t_{n+1} 时的 y_{n+1} 值。这里 h_{n+1} 为步长， $h_{n+1} = t_{n+1} - t_{n0}$ 。

其特例为单步法。前向欧拉、后向欧拉和梯形法就属于单步法。

1. 前向欧拉法

取 $p=0$ ， $a_0=1$ ， $b_0=1$ ，其它所有系数都为零。得

$$y_{n+1} = y_n + h_{n+1} \dot{y}_n \quad (7-18)$$

2. 后向欧拉法

取 $p=0$ ， $a_0=1$ ， $b_{-1}=1$ ，其它所有系数都为零。得

$$y_{n+1} = y_n + h_{n+1} \dot{y}_{n+1} \quad (7-19)$$

3. 梯形法

取 $p=0$, $a_0=1$, $b_{-1}=b_0=-\frac{1}{2}$, 其它所有系数都为零。得

$$y_{n+1} = y_n + \frac{h_{n+1}}{2} (\dot{y}_{n+1} + \dot{y}_n) \quad (7-20)$$

积分方法表明了每一 t_{n+1} 时刻 y_{n+1} 与 $\dot{y}_n$ 或 $\dot{y}_{n+1}$ 的关系, 如果我们知道了 $\dot{y}$ 与 y 及 t 之间的函数关系, 求解就容易了。现以后向欧拉为例。

如果 $\dot{y}_{n+1}$ 能表示成为 $f(y_{n+1}, t_{n+1})$ 时, 则公式(7-19)就变成

$$y_{n+1} = y_n + h_{n+1} f(y_{n+1}, t_{n+1}) \quad (7-21)$$

对上列的代数方程 (一般讲是非线性的) 就可用牛顿-拉夫森法求得 t_{n+1} 时的解 y_{n+1} 。所以求解瞬态解时, 首要任务是把储能元件的微分方程转换成等效的“准直流”代数方程。这样就可象求直流解时一样, 通过牛顿-拉夫森法对此非线性代数方程组求解。因此, 数值积分法把一个求 N 个时间点的瞬态分析的任务简化为 N 次重复的求“准直流”解的问题。

二、储能元件的离散化模型

对于储能元件可以求出 $\dot{y}_{n+1} = f(y_{n+1}, t_{n+1})$ 的关系。现以梯形积分法为例说明。

1. 电容

电容中的电流与端电压服从以下关系, 即

$$i = C \frac{dV}{dt} \quad \text{或} \quad \dot{V} = \frac{dV}{dt} = \frac{i}{C} \quad (7-22)$$

利用梯形公式(7-20), 即

$$Q_{n+1} = Q_n + \frac{h_{n+1}}{2} (\dot{Q}_{n+1} + \dot{Q}_n)$$

移项整理后得

$$\dot{Q}_{n+1} = \frac{2}{h_{n+1}} Q_{n+1} - \left(\dot{Q}_n + \frac{2}{h_{n+1}} Q_n \right) \quad (7-23)$$

因而

$$i_{n+1} = \frac{2C}{h_{n+1}} V_{n+1} - \left(i_n + \frac{2C}{h_{n+1}} V_n \right) \quad (7-24)$$

上式说明电容可以由一个电导和电流源的并联来等效, 如图 7-22。 $G_{en} = \frac{2C}{h_{n+1}}$, I_{en}

$$= - \left(i_n + \frac{2C}{h_{n+1}} V_n \right).$$

2. 电感

电感中的电流与端电压服从以下关系, 即

$$V = L \frac{di}{dt} \quad \text{或} \quad \dot{i} = \frac{di}{dt} = \frac{V}{L} \quad (7-25)$$

同样利用梯形公式(7-20)得到

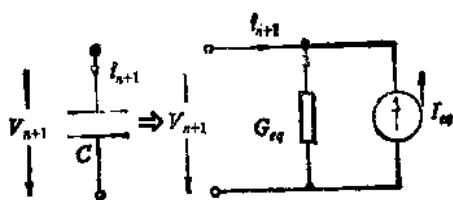


图 7-22 电容的离散化模型

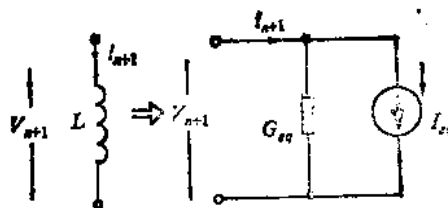


图 7-23 电感的离散化模型

$$\phi_{n+1} = \phi_n + \frac{h_{n+1}}{2} (\dot{\phi}_{n+1} + \dot{\phi}_n) \quad (7-26)$$

将公式(7-25)代入上式得

$$i_{n+1} = \frac{h_{n+1}}{2L} V_{n+1} + \left(i_n + \frac{h_{n+1}}{2L} V_n \right) \quad (7-27)$$

上式说明电感可以由一个电导和电流源的并联来等效。如图 7-23。因此对梯形法

$$G_{eq} = \frac{h_{n+1}}{2L} \quad I_{eq} = \left(i_n + \frac{h_{n+1}}{2L} V_n \right)$$

对于其它非线性储能元件，如二极管和三极管中的电容，也可同样处理。

三、步长的确定

在时间为零这一点，无法预估下一步长。SPICE2 中第一个时间点与零时间点之间的步长取为 DELMAX/10。

DELMAX 是 TSTEP 和 (TSTOP-TSTART)/50 中较小的一个，即

$$\text{DELMAX} = \text{DMIN0}(\text{TSTEP}, (\text{TSTOP} - \text{TSTART})/50.0) \quad (7-28)$$

式中，TSTEP 为行打印的输出增量；TSTOP 为用户规定的瞬态分析终止时间；TSTART 为初始时间。

第一个时间点以后的步长可以预估。其方法有：

(1) 迭代次数控制法

这是一种经验的方法。它根据迭代次数来修改步长，在修改步长后重新进行迭代。

SPICE2 中设置了两个量。瞬态分析时迭代次数的下限 $ITL3=n_1$ ， n_1 的隐含值为 4。

另一量是瞬态分析时迭代次数的上限 $ITL4=n_2$ ， n_2 的隐含值为 10。

当用上一次的步长进行求解，并在 n_2 次迭代内不能给出收敛解时，将时间步长缩小为原来步长的 1/8，重新迭代求解。如仍无效再一次缩小直到得到收敛解。然而若迭代次数小于 n_1 时已能给出收敛结果，则将步长加倍作为下一新步长。但为了保持求解的精度，不允许步长超过某一规定的最大步长 h_{max} 。SPICE2 中有一可选项 LVLTIM=x，如设 x=1 就采用以上方法。

(2) 局部截断误差 (LTE) 控制法

数值积分所求得解是近似解，所产生的误差称为截断误差。所谓 t_{n+1} 时刻的局部截断误差是假设所有以前的时刻 ($t_0, t_1, t_2, \dots, t_n$) 的解都是精确解，仅在 t_{n+1} 时计算得到的 y_{n+1} 与精确解 $y(t_{n+1})$ 之间所存在的误差。

由于步长与局部截断误差之间存在着密切的关系，为了使解的误差保持在一个合理的范围内，在瞬态分析时需要对步长进行调整。

根据分析, 梯形法的局部截断误差为

$$\text{LTE}_{n+1} = -\frac{h_{n+1}^3}{12} \frac{d^3 y}{dt^3}(\xi) \quad (7-29)$$

其单位对电容而言为电荷, 对电感而言为磁通。上式可近似为

$$|\text{LTE}_{n+1}| \approx \left| C_{K+1} h_{n+1}^{K+1} \frac{d^{K+1} y}{dt^{K+1}}(t_{n+1}) \right| \quad (7-30)$$

式中, K 为阶数, 对梯形法 $K=2$ 。

从公式 (7-30) 可以看出 LTE 与步长 h 有着密切的关系。 h 越大, 则局部截断误差越大。但 LTE 的值是有限制的, 它必须满足

$$|\text{LTE}_{n+1}| \leq \varepsilon_a + \varepsilon_r \max[|y_{n+1}^A|, |y_{n+1}^B|] \quad (7-31)$$

这样就可以利用 LTE 作为调整步长的一种手段, 这就是截断误差控制法的基本思想。

从公式 (7-30) 得到步长 h_{n+1} 必须满足式

$$h_{n+1} \leq \left[\frac{|\text{LTE}_{n+1}|}{\left| C_{K+1} \frac{d^{K+1} y}{dt^{K+1}}(t_{n+1}) \right|} \right]^{1/(K+1)} \quad (7-32)$$

才能使局部截断误差满足收敛要求。为计算方便, 现用差分代替 (7-32) 式中的微分, 因

$$\frac{d^{K+1} y}{dt^{K+1}}(t_{n+1}) = (K+1)! DD_{K+1}(t_{n+1}) \quad (7-33)$$

代入 (7-32) 式得

$$h_{n+1} \leq \left[\frac{|\text{LTE}_{n+1}|}{|C_{K+1}| |(K+1)! DD_{K+1}(t_{n+1})|} \right]^{1/(K+1)} \quad (7-34)$$

式中

$$DD_{K+1}(t_{n+1}) = \frac{DD_K(t_{n+1}) - DD_K(t_n)}{\sum_{i=0}^K h_{n+1-i}} \quad (7-35)$$

为第 $K+1$ 次均差公式。所以第一次均差公式有

$$DD_1(t_{n+1}) = \frac{y_{n+1} - y_n}{h_{n+1}}$$

而第二次均差公式有

$$DD_2(t_{n+1}) = \frac{DD_1(t_{n+1}) - DD_1(t_n)}{h_{n+1} + h_n} = \frac{\frac{y_{n+1} - y_n}{h_{n+1}} - \frac{y_n - y_{n-1}}{h_n}}{h_{n+1} + h_n}$$

其它依次类推。

利用截断误差调整步长的步骤如下:

(1) 先给定一 h_{n+1} (由上一时间点预估而得)。令 $\text{DELTA} = h_{n+1}$ 。

(2) 在此步长下求出 y_{n+1} 和 $DD_2(t_{n+1})$ (对梯形法而言), 代入公式 (7-34)。

若满足, 说明此步长是可取的。若不满足, 则以公式 (7-34) 右端值作为新步长 h_{n+1} 。

即

$$h_{new} = \left[\frac{|LTE_{n+1}|}{|C_{K+1}| |(K+1)! DD_{K+1}(t_{n+1})|} \right]^{1/(K+1)} \quad (7-36)$$

这时的 h_{new} 比原来的 h_{n+1} 已缩小。令 $DELNEW = h_{new}$ 。

(3) 在新的时间点重新进行计算。即通过下列语句

$T = T - DELTA$

$DELTA = DELNEW$

$T = T + DELTA$

修改为新时间点，再迭代求解。

(4) 如果收敛，则再次检查公式(7-34)是否满足。如满足，说明此时间步长是可取的。如不满足，则重复上述过程，直到满足为止。

(5) 下一时间点的时间步长取当前时间点的步长作为预估值。

根据经验，用公式(7-36)求得的 h_{n+1} 一般偏小，为此在 SPICE2 中设置一可选项 TRTOL 称瞬态误差容限，其隐含值为 7。则公式(7-36)修改为

$$h_{new} = \left[\frac{e_r |LTE_{n+1}|}{|C_{K+1}| |(K+1)! DD_{K+1}(t_{n+1})|} \right]^{1/(K+1)} \quad (7-37)$$

式中， e_r 即 TRTOL(瞬态误差容限)。

在 SPICE2 中，隐含的控制时间步长的方法是截断误差控制法。

四、根据断点位置修正步长

所谓断点就是电压源或电流源中的突变点。遇到断点时，要对预估的步长进行修正。

当 t_n 与 $t_n + h_{n+1}$ 之间有断点时，应减小 h_{n+1} ，令 $h_{n+1} = DEL1$ 。如图 7-24(a) 所示。当断点的位置与 $t_n + h_{n+1}$ 相差很小时 ($< 0.01h_{n+1}$)，应加大 h_{n+1} ，令 $h_{n+1} = DEL2$ 。如图 7-24(b) 所示。

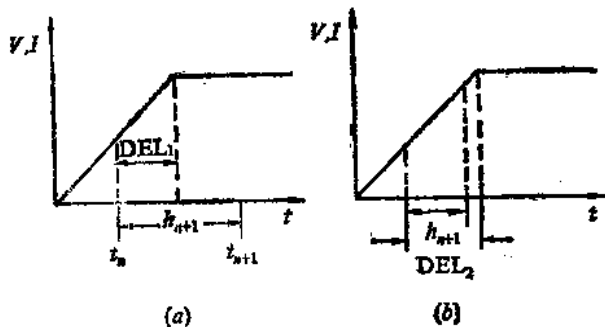


图 7-24 根据断点位置修正步长

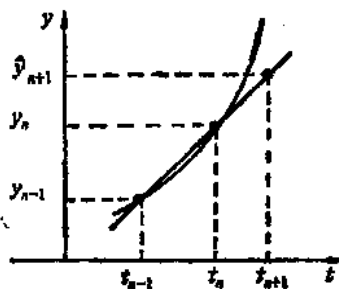


图 7-25 线性外推法示意图

采取这一措施可以避免在快速过渡区造成误差的增大。

五、迭代初始值

在瞬态分析中，利用线性外推法可预测下一时间点的初始值。正确地预测初始值，可减少迭代次数加快收敛。线性外推法如图 7-25 所示。

$$\hat{y}_{n+1} = y_n + \frac{y_n - y_{n-1}}{h_n} h_{n+1} \quad (7-38)$$

六、变阶积分法——基尔 (Gear) 法

SPICE2 中数值积分的隐含方法是梯形法，同时提供了基尔法作为可选项。

基尔法是一种求解精度较高的多步法。所谓多步法，就是在求 y_{n+1} 时需要已知 $y_n, y_{n-1}, y_{n-2}, \dots$ 等更多时间点的值。

基尔法的公式如下：

$$y_{n+1} = \sum_{i=0}^{K-1} a_i y_{n-i} + h_{n+1} b_{-1} \dot{y}_{n+1} \quad (7-39)$$

或表示成

$$\dot{y}_{n+1} = \sum_{i=0}^K a_i y_{n+1-i} \quad (7-40)$$

对不同的阶，有公式如下：

$$\left. \begin{aligned} \text{一阶} \quad y_{n+1} &= a_0 y_n + b_{-1} h_{n+1} \dot{y}_{n+1} \quad (\text{即后向欧拉}) \\ \text{二阶} \quad y_{n+1} &= a_0 y_n + a_1 y_{n-1} + b_{-1} h_{n+1} \dot{y}_{n+1} \\ \text{三阶} \quad y_{n+1} &= a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + b_{-1} h_{n+1} \dot{y}_{n+1} \\ \text{四阶} \quad y_{n+1} &= a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + a_3 y_{n-3} + b_{-1} h_{n+1} \dot{y}_{n+1} \\ \text{五阶} \quad y_{n+1} &= a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + a_3 y_{n-3} + a_4 y_{n-4} + b_{-1} h_{n+1} \dot{y}_{n+1} \\ \text{六阶} \quad y_{n+1} &= a_0 y_n + a_1 y_{n-1} + a_2 y_{n-2} + a_3 y_{n-3} + a_4 y_{n-4} + a_5 y_{n-5} + b_{-1} h_{n+1} \dot{y}_{n+1} \end{aligned} \right\} \quad (7-41)$$

各阶基尔法的系数值如表 7-8。

表 7-8 基尔法的系数值

阶	a_0	a_1	a_2	a_3	a_4	a_5	b_{-1}
1	1						1
2	4/3	-1/3					2/3
3	18/11	-9/11	2/11				6/11
4	48/25	-36/25	16/25	-3/25			12/25
5	300/137	-300/137	200/137	-75/137	12/137		60/137
6	360/147	-450/147	400/147	-225/147	72/147	-10/147	60/147

可以看出基尔法与梯形法的不同点在于解 y_{n+1} 时除了应知道 y_n 和 $\dot{y}_{n+1}$ 外，还必需了解 $K-1$ 个时间点的 y 值（即 $y_{n-1}, y_{n-2}, \dots, y_{n-K+1}$ ）。

因此在使用高阶基尔公式时至少应有一个起码的时间点数的解已经求出。例如对于二阶，“起始”的时间点数为 $K-1=1$ ，即需要已知 y_{n-1} 。对于三阶，“起始”时间点数为 $K-1=2$ ，即需要已知 y_{n-1}, y_{n-2} 。其它阶数可类推。

所以对于第一个时间点只能使用一阶基尔法。在六个解都已求出后，就可以任意采用一至六阶中的任意一个。

为了采用基尔法，必须开辟相应的存储区把前几个时间点的解保存起来。

各阶基尔公式的局部截断误差见表 7-9。

由上表可以看出，高阶的截断误差较小，因此在给定的误差容限时，高阶的基尔公式

可以选用较大的步长,这意味着瞬态分析用较短的时间就能完成。而另一方面,如果要求计算精度提高时,就应该采用高阶公式。

对于基尔法,隐含方法同样采用截断误差控制步长法。因此同样可以根据公式(7-36)来预估步长。

采用基尔法的优点是在计算过程中可以改变阶数(即采用不同阶的基尔公式)以获得最佳的效果。

SPICE2中采用的变阶策略如下:

如现在采用的是 K 阶公式,在计算出 K 个时间点后,利用 $K-1$ 阶、 K 阶和 $K+1$ 阶公式求出 h_{n+1} 。如 $K=1$,则不计算 $K-1$ 阶。如 $K=6$,则不计算 $K+1$ 阶的步长。然后选择产生最大步长的那个阶作为变阶的对象(如果产生最大步长的为本阶,则不变阶)。但是否变阶,还要判断 $K-1$ 阶和 $K+1$ 阶所预估的步长是否比本阶的步长大 R_0 倍, R_0 的典型值为 $1.05\sim 1.5$ 。如果大 R_0 倍则变阶,否则也不变阶。如果 $K-1$ 阶和 $K+1$ 阶所预估的步长相等时,则优先采用 $K-1$ 阶,因为阶数越低,算法越简单,需要运算的时间也越少。

由于每一步都要进行变阶会耗费很多机时,因此在实际上只有在本阶的预估步长 $h_{new} \geq 0.9h_n$ 时才允许作变阶前的计算并作出判断是否变阶。

对于一般正常的误差要求,变阶所付出的代价将抵消掉采用高阶公式的优点。而要求严格容差时,变阶基尔算法比起梯形法来讲优点比较明显。

综上所述,瞬态分析的求解过程可归纳如下:

- (1) 根据步长求出下一时间点。
- (2) 求电压源或电流源在新时间点上的值。
- (3) 根据数值积分公式计算储能元件的等效电导和等效电流值。
- (4) 建立该时间点的非线性代数方程组,通过牛顿-拉夫森法进行“准直流”求解。如收敛,检查该步长是否可取。若可取,则存储变出变量。若不可取,则求 h_{new} ,并在新时间点重新迭代求解。

如不收敛,则将步长缩小为原步长的 $1/8$,重新迭代求解。如步长缩小到某一规定的内部最小步长 h_{min} ,则退出分析。

- (5) 如采用变阶基尔法,则判断是否应该变阶。如认为应变阶,则求出新步长和确定新阶数,重新进行步骤(1)~(4)。

- (6) 预估下一步长。

- (7) 检查断点位置,以确定是否修改下一时间步长。

重复上述过程,直到分析时间超过用户所设定的终止时间为止。

表 7-9 基尔公式的局部截断误差

阶	局部截断误差
1	$-\frac{1}{2}h^2_{n+1} \frac{d^2y}{dt^2}(\xi)$
2	$-\frac{2}{9}h^3_{n+1} \frac{d^3y}{dt^3}(\xi)$
3	$-\frac{3}{22}h^4_{n+1} \frac{d^4y}{dt^4}(\xi)$
4	$-\frac{12}{125}h^5_{n+1} \frac{d^5y}{dt^5}(\xi)$
5	$-\frac{10}{137}h^6_{n+1} \frac{d^6y}{dt^6}(\xi)$
6	$-\frac{60}{1029}h^7_{n+1} \frac{d^7y}{dt^7}(\xi)$

§ 7.6 牛顿迭代解的收敛性改进

至今为止, 牛顿-拉夫森法仍是求解非线性方程组的最为基本的和有效的方法。它在收敛域中具有平方律的收敛速度。但它对初始“猜解”(初值)有较高的要求, 即尽可能希望初值能接近于最终解。如初值不在收敛域内, 牛顿迭代可能发散。

为了改进牛顿法的收敛性, 人们从两个方面进行大量的研究。一是在牛顿迭代法基础上提出了一些改进算法。如阻尼牛顿法、拟牛顿法等。另一方面是如何得到一个足够好的初值以保证牛顿迭代法的收敛。

现结合电路模拟的情况介绍一下后一方面的工作进展。

一、源步进法

源步进法在 SPICE2G6 版本中得到了应用。此法的基本思想是: 对同一电路, 在低偏置条件下的直流解一般要比高偏置时容易求解。因此我们先求得容易求解的低偏置时的解, 再将该解作为下一步高偏置时的迭代初值。如此逐步提高偏置, 直到达到用户所给定的偏置电压值为止。

在 SPICE2G6 中当规定可选项卡的 $ITL6 \neq 0$ 时, 程序即转入源步进法来计算直流工作点。此法的程序框图如图 7-26 所示。图中, $SFACTR$ 为偏置电压比例因子。当 $SFACTR=1$ 时, 偏置电压达到用户所给定值; $BOUND$ 为偏置电压比例因子的下限, 即可以允许的最低偏置; $FRACN$ 为调态因子。由它决定下一迭代的偏置电压比例因子。

程序分三个主要部分。第一部分先在一个较低的偏置下求得收敛解。第二部分逐步提高偏置值, 如迭代成功, 再次提高偏置。第三部分如迭代失败, 则再次降低偏置以求得收敛解, 并转到第二部分。如此反复, 最后在用户给定的偏置电压下得到该电路的直

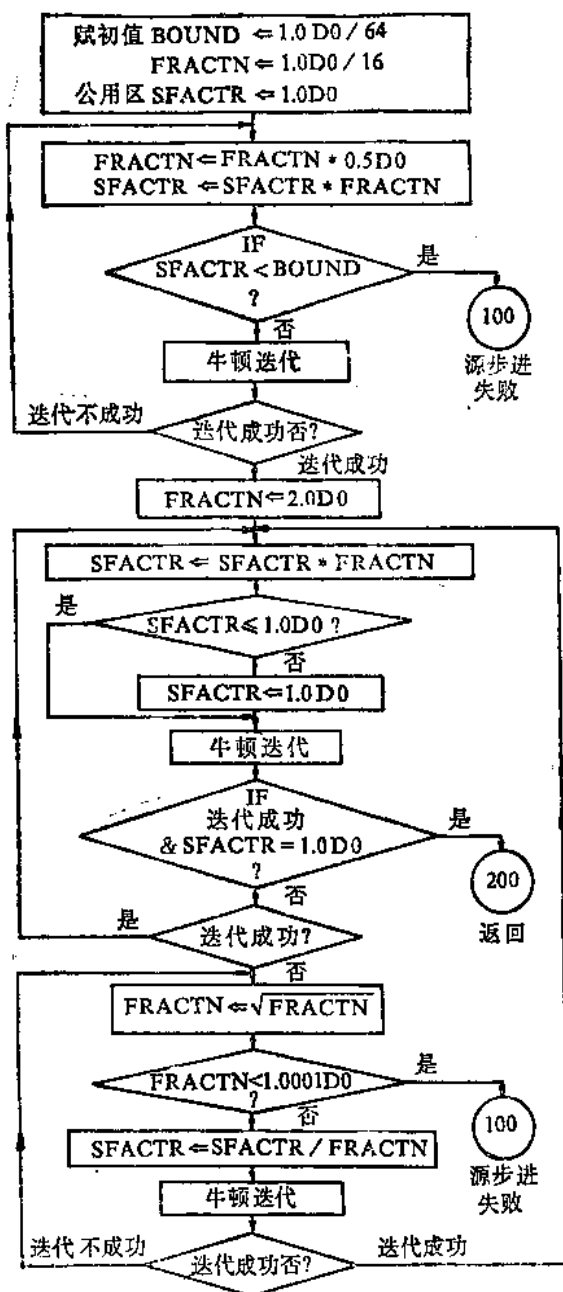


图 7-26 源步进法程序框图

流工作点解。

二、牛顿预测法

此法的目的是设法预测牛顿法的初始“猜解”，以保证牛顿迭代的收敛。

我们已知在电路模拟中通过牛顿-拉夫森法要求解的是一组非线性方程组

$$F(x)=0 \quad (7-42)$$

如把电源电压作为参量 p ，而与要求解的变量 x 加以分开，则上式 (7-42) 可改写为

$$F(x, p)=0 \quad (7-43)$$

通常的牛顿-拉夫森法是在 p 为固定的条件下，通过初始“猜解” x^0 并经过多次迭代得到待求解 x 。收敛的 x^* 就视为方程 $F(x, p)=0$ 的解。这里的 x 是节点电位或支路电流， p 为电源电压或输入讯号。

如果把 p 看成独立变量，于是有

$$F_x(x^*, p)\Delta x = -F_p(x^*, p)\Delta p \quad (7-44)$$

这里 F_x 是已知的为获得 x^* 时的最后一次迭代所用的雅可比行列式。

F_p 通常具有很简单的形式。例如，当电路中只有一个电源时， F_p 是一单位矢量，其形式为 $(0, \dots, 0, \pm 1)^T$ 。

牛顿预测法的基本思想是：如已知 p 时的精确解 x^* ，若要在 $p + \Delta p$ 偏置条件下求解方程 (7-43) 时，则 $x^* + \Delta x$ 将是牛顿迭代时的一个很好的初始“猜解”。

在电路模拟中，通常选 p 为很小的偏置值，以保证牛顿迭代收敛到 x^* 。然后提高 p 值，利用公式 (7-44) 求得 $p + \Delta p$ 条件下的初始“猜解” $x^* + \Delta x$ ，再以这个初值进行牛顿迭代得到 $p + \Delta p$ 时的精确解。

如果 Δp 取得太大，想通过一步就达到用户所给定的偏置值，则会导致初始“猜解”过于粗糙，同样会使牛顿迭代失败。但 Δp 取得过小，就需要很多步才能使 p 达到用户给定值，因而会失去牛顿预测法在速度上的优点。

为此采用一种类张弛法，其原则如下：

(1) 先从一个能保证得到方程精确解 x^* 的很小的 p 开始（通常在很小的 Δp 时总是收敛的，而且只需很少的几次迭代）。

(2) 把给定的 Δp (p 与用户给定值之间的差值) 分成 N 个子增益 δp_i ，即 $\Delta p = \sum_{i=1}^N \delta p_i$ (δp_i 不一定是均匀的)。然后逐步提高 p 值。

(3) 用每次求得的 $x + \delta x_i$ 和 $p + \delta p_i$ 值对前一次 (即 $i-1$ 次) 的雅可比行列式进行一次修改，并以这个修改过的雅可比行列式作为预测下一个 δx 的基础。

现以只有一个电压源为例列出算法的实现步骤如下：

(1) 存放雅可比行列式 A (正常中止时牛顿迭代过程中最后一次迭代的雅可比行列式) 和收敛解 x 。

(2) 列出方程 (7-44) 为以下形式

$$A\delta x = b$$

这时 b 中除了对应于电压源的项应设置为 δp 外，其它 b 项都设置为零。因假设只有一个电压源，故 δp 可简化为 δV 。

(3) 解上列线性方程得 δx ，修改 x 为

$$x = x + \delta x$$

(4) 用修改过的 x 和 p 计算出新的雅可比行列式 A 。

(5) 回到(2),但代入下一子增量 δp (即 δV)。

重复(2)~(5),直到所有的 δV_i 都被处理过为止。最后一次得到的 x 将是方程(7-43)进行牛顿-拉夫森迭代时的初始“猜解”。这个初始“猜解”将能保证牛顿迭代的收敛。

为了消除这种方法所产生的累积误差以确保初始“猜解”更接近精确解,甚至可以在增加 p 的过程中进行几次全牛顿-拉夫森迭代。

除了以上介绍的一阶牛顿预测法外,还可应用二阶牛顿预测法以获得更接近精确解的初始“猜解”,使牛顿-拉夫森法具有更好的收敛性和更快的收敛速度。一阶和二阶牛顿预测法已试用于 SPICE3A7,实验证明此法是可行的。

第八章 器件模拟

§ 8.1 器件模拟的目的

半导体器件的计算机模拟是在给定的物理结构和掺杂分布条件下,通过软件程序直接求解半导体器件的基本方程,从而得到器件的电学特性,如伏安特性和一些电学参数。

早在40年代末,在双极型晶体管问世后不久就出现了晶体管的理论模型。但那时为了能用分析方法求解微分方程,不得不作许多近似处理,例如小注入近似、均匀基区近似等等。但实际的晶体管并不满足这些近似,因而分析结果难于真实地反映实际情况。

近年来,随着计算机技术和数值求解方法的迅速发展,使我们有可能用数值分析方法直接求解半导体器件的基本方程,因而器件模拟得到了普遍重视和迅速发展。特别是VLSI的发展要求器件的尺寸越来越小,人们需要更深入地了解影响器件性能的各种因素,以便于控制器件的性能和进行优化设计。而这种参数的调整、性能的研究工作都可以在实际制作器件以前加以全面的展开,这就大大地推动了器件的改进和新器件的研制。

最近几年我们可以把工艺模拟与器件模拟结合起来,可将工艺模拟所求得的半导体内部的杂质分布直接送入器件模拟程序,使器件模拟的效率大为提高,并能直接研究工艺条件、工艺参数对器件性能的影响,这也是器件模拟得到广泛应用的一个重要原因。

通过器件模拟我们还可对电路模拟程序提供各种器件的精确模型,以提高电路模拟程序的精度。目前已有一种把器件模拟和电路模拟统一在一起的模拟程序,如MEDUSA程序。

最初的器件模拟工作只限于一维分析,它的优点是能得到比较严格的、形象的器件工作状态,而花费的代价比较小。但随着VLSI集成电路的发展,器件尺寸日益缩小,一维分析已不能适应要求,因而从70年代初开始发展了二维器件分析,并进而发展到三维分析,但这些分析工作都要花费大量的计算机机时,一般都要求在大型机上进行。

一开始器件模拟的对象都集中于硅器件,近年来由于化合物半导体器件的发展,很多程序开发工作转向不均匀能带结构的器件,以及砷化镓或其它多元化合物半导体器件。

目前器件模拟程序有很多种。在一维器件模拟方面有美国斯坦福大学开发的SEDAN程序,西德亚琛大学开发的MEDUSA程序等。其中SEDAN I发表于1979年,1982年开发了SEDAN II,1985年又发表了SEDAN III,可分析的器件类型除硅二极管和晶体管外,还有肖特基二极管和多晶硅发射极晶体管。此外还把应用范围扩大到半导体化合物器件,包括二元的GaAs,三元的 $\text{Al}_x\text{Ga}_{1-x}\text{As}$ 和 $\text{Hg}_{1-x}\text{Cd}_x\text{Te}$ 。

在二维器件模拟方面,比较著名的程序有CADDET,MINIMOS,PISCES和FIELDAY等。其中的CADDET是同时求解泊松方程和一种载流子的电流连续性方程。PISCES和FIELDAY的功能很强,能进行MOSFET和双极型器件的稳态解和瞬态解,甚至还可以进行交流小信号分析。如把几个器件连在一起进行模拟,还可研究闩锁效应(Latch-up效应)。PISCES还提供了很多可选项,既可利用耦合法同时求解泊松方程和二种载流子

的连续方程,也可采用解耦顺序法依次求解三个方程。在网络的设置上既提供了矩形网格,也提供三角元网格可供用户选择。

目前在三维器件模拟方面能投入实用的程序还不多。除美国 IBM 公司的 FIELDAY (二维和三维器件分析程序)外,还有 CADDETH 程序能对 MOSFET 和双极型器件进行稳态分析。

§ 8.2 器件模拟的内容

现以 SEDAN III 为例说明可模拟的内容。它能求解一维器件结构中的静电势、电子浓度、空穴浓度与空间及时间的关系,并在此基础上获得不同偏压条件下器件的伏安特性和各种器件参数。

SEDAN III 可以模拟的器件类型有:(1)二极管;(2)双极型晶体管;(3)具有多晶硅发射极的晶体管;(4)肖特基二极管;(5)任意的半导体-绝缘体-半导体结构;(6)具有源接触的 MOS 电容器。

构成器件的材料除硅外,可以有二元化合物 GaAs,三元化合物 $Al_xGa_{1-x}As$ 和 $Hg_{1-x}Cd_xTe$,因而可以模拟异质结二极管和异质结双极型晶体管。

器件模拟时的输入信息应包括器件结构、材料成份、掺杂分布和外加偏压。

输出结果既可以是稳态解,也可以是瞬变过程的解。输出的电学特性可包括:(1)电场沿器件深度的变化曲线;(2)静电势沿器件深度的变化曲线;(3)电子浓度、空穴浓度及净掺杂浓度在器件内的分布;(4)电子和空穴准费米能级随深度的变化;(5)能带图随深度的变化;(6)电子和空穴迁移率随深度的变化;(7)电子和空穴寿命随深度的变化;(8)电子和空穴电流密度随深度的变化;(9)晶体管的直流增益随集电极电流密度的变化;(10)晶体管端电流密度随偏压的变化;(11)二极管电流密度随偏压的变化;(12)晶体管结电容随偏压的变化;(13)二极管结电容随偏压的变化;(14)晶体管截止频率随偏压的变化;(15)MOS 电容器硅-二氧化硅界面处及衬底中电子、空穴和净电荷随偏压的变化;(16)MOS 电容器的电容随偏压的变化。

此外还可输出给定偏压下各层的薄层电阻、晶体管发射极和基极的 Gummel 数、晶体管的欧拉电压和各种击穿电压以及二极管的击穿电压等。

再以二维的 PISCES 为例。它既可以模拟双极型器件,也可以模拟 MOSFET 器件。器件结构既可以允许平面形状也允许非平面的形状。

在器件材料方面除硅外,还允许 GaAs 材料。MOSFET 或 MESFET 的栅导电层可以是多晶硅、铝、钼、钨及钼或钨的硅化物。栅绝缘层可以是 SiO_2 、 Si_3N_4 或蓝宝石。

可以模拟的器件类型有:MOSFET (包括 CMOS)、MESFET、结型器件和肖特基接触器件。

器件模拟时的输入信息同样应包括器件结构、材料成份、掺杂分布和外加偏压。掺杂分布可以由解析表达式来确定,也可以将 SUPREM III 的输出作为输入。

输出可包括稳态解、或瞬态解,也可以有交流小信号分析。

输出的图形可以有静电势、载流子浓度、电场强度和电流密度的二维等值分布图。此外还可以画出网格图以及器件的外边界、界面和电极的位置等。

§ 8.3 器件模拟的求解方法

一、基本方程

器件模拟体现在数学上的任务就是根据一定的边界条件求解基本方程,得到静电势 ψ 、两种载流子(电子 n 和空穴 p)在空间的分布及随时间的变化。

基本方程有泊松方程

$$\nabla(\epsilon \nabla \psi) = -\rho \quad (8-1)$$

和载流子连续方程

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{j}_n - U \quad (8-2)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{j}_p - U \quad (8-3)$$

式中, ψ 为静电势; ϵ 为介电常数; ρ 为空间电荷密度; n , p 为电子和空穴浓度; $\mathbf{j}_n$, $\mathbf{j}_p$ 为电子和空穴电流密度; U 为净复合率, $U=R-G$ 。如果净掺杂浓度由施主杂质和受主杂质所组成,则空间电荷密度可表示为

$$\rho = q(p - n + N_D^+ - N_A^-) \quad (8-4)$$

式中, N_D^+ 和 N_A^- 为施主杂质和受主杂质浓度。

电子和空穴的电流密度取决于输运方程

$$\mathbf{j}_n = -q\mu_n n \nabla \phi_n \quad (8-5)$$

$$\mathbf{j}_p = -q\mu_p p \nabla \phi_p \quad (8-6)$$

式中, μ_n , μ_p 为电子和空穴的迁移率; ϕ_n , ϕ_p 为电子和空穴的准费米势。值得指出的是,公式(8-5)、(8-6)对非均匀能带结构的半导体器件也是适用的。通常电流密度可表示为漂移流和扩散流之和,即

$$\mathbf{j}_n = qn\mu_n \mathbf{E}_n + kT\mu_n \nabla n \quad (8-7)$$

$$\mathbf{j}_p = qp\mu_p \mathbf{E}_p - kT\mu_p \nabla p \quad (8-8)$$

一般讲,电场 $\mathbf{E}_n$ 和 $\mathbf{E}_p$ 是不相等的,只有在均匀能带结构且不考虑能带变窄效应,并采用玻尔兹曼统计时, $\mathbf{E}_n$ 才等于 $\mathbf{E}_p$ 且

$$\mathbf{E} = -\nabla \psi \quad (8-9)$$

因此在一维情况下,器件的基本方程有

$$\frac{\partial}{\partial x} \left(\epsilon \frac{\partial \psi}{\partial x} \right) = -q(p - n + N_D^+ - N_A^-) \quad (8-10)$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \frac{\partial}{\partial x} j_n - U \quad (8-11)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \frac{\partial}{\partial x} j_p - U \quad (8-12)$$

$$j_n = qn\mu_n E + qD_n \frac{\partial n}{\partial x} \quad (8-13)$$

$$j_p = qp\mu_p E - qD_p \frac{\partial p}{\partial x} \quad (8-14)$$

如假设 ε 为常数, 并对上式进行归一化, 可得下列简式

$$\frac{\partial^2 \psi}{\partial x^2} = n - p - N \quad (8-15)$$

$$\frac{\partial n}{\partial t} = \frac{\partial j_n}{\partial x} - U \quad (8-16)$$

$$\frac{\partial p}{\partial t} = -\frac{\partial j_p}{\partial x} - U \quad (8-17)$$

$$j_n = n\mu_n E + \mu_n \frac{\partial n}{\partial x} \quad (8-18)$$

$$j_p = p\mu_p E - \mu_p \frac{\partial p}{\partial x} \quad (8-19)$$

式中, $N = N_D^+ - N_A^-$

二、基本方程的离散化

为了求得数值解, 首先要对偏微分方程进行空间和时间的离散化。

我们可以把空间的离散化看作是在有限的求解空间中布上网格, 然后用中心差分法把微分方程转换成差分方程 (详见附录 I-1)。

在一维情况下, 可将被分析的器件划分成若干个格点, 如图8-1所示。网格的示意图, 见图8-2。

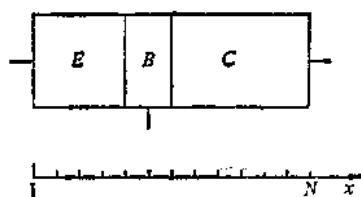


图 8-1 把求解空间划分成 N 个格点

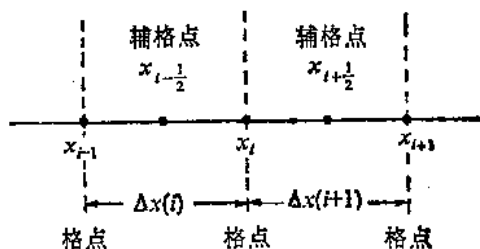


图 8-2 网格的示意图

现将基本变量 ψ 、 n 、 p 定义在格点上, 并认为 j_n 、 j_p 、 μ_n 、 μ_p 、 E 各变量在第 i 格点和第 $i+1$ 格点间为常数, 即定义在辅格点上。最简单的为等步长差分网格。但由于器件内部 ψ 、 n 和 p 的变化在某些区域会非常剧烈, 而在其它某些区域其变化又会非常缓慢, 为了得到足够的求解精度, 往往采用变步长的差分网格。

在二维器件模拟程序中, 最常用的网格有两种: 矩形和三角形。如图8-3所示。矩形网格中最典型的是变步长差分网格, 每个格点与周围的4个格点相联。这种网格简单、规则性强, 由于可以变步长, 网格的效率有所改善, 但在处理非平面边界时这种网格会遇到较大困难。三角形网格则能克服这些缺点, 可很方便地处理各种非平面问题, 此外, 由于能够进行局部加密, 因此网格的效率很高。对三角形网格, 每个格点和与此格点相毗邻的三角形有关。

变步长矩形网格应用于 MOS 器件的一个例子示于图8-4。网格的间距在 x 方向和 y 方向都是可变的。

在器件模拟时, 由于程序的执行时间和存储量的需求直接与离散空间的格点数有关, 因而在模拟时, 减少格点的数目就成为一个重要的问题。

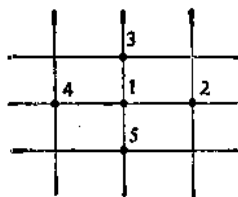


图 8-3 二维网格

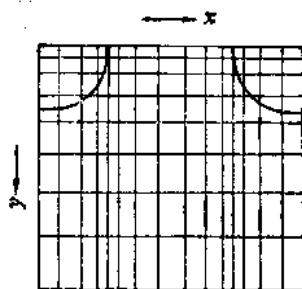
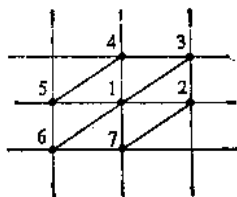


图 8-4 变步长矩形网格

对一维泊松方程 (8-15) 采用中心差分法进行离散化后有

$$\frac{1}{\Delta x(i+1)}[\psi(i+1) - \psi(i)] - \frac{1}{\Delta x(i)}[\psi(i) - \psi(i-1)] = n(i) - p(i) - N(i) \quad (8-20)$$

$$\frac{1}{2}[\Delta x(i+1) + \Delta x(i)]$$

对输运方程 (8-18)、(8-19) 同样可以采用一般的有限差分法。但当相邻两格点之间的电位差超过某个值 (如 $2kT/q$) 时会出现数值上的不稳定性。

Scharfetter 和 Gummel 提出了一种方法可以避免这种不稳定性。将公式 (8-18) 重新排列得

$$\frac{dn}{(j_n/\mu_n) - nE} = dx \quad (8-21)$$

对上式从 $x(i)$ 到 $x(i+1)$ 进行积分, 并假设在 i 到 $i+1$ 格点范围内, j 、 μ 、 E 为常数并分别等于 $j_n(i + \frac{1}{2})$ 、 $\mu_n(i + \frac{1}{2})$ 和 $E(i + \frac{1}{2})$, 则得到

$$\ln \left\{ \frac{[j_n(i + \frac{1}{2})/\mu_n(i + \frac{1}{2})] - n(i+1)E(i + \frac{1}{2})}{[j_n(i + \frac{1}{2})/\mu_n(i + \frac{1}{2})] - n(i)E(i + \frac{1}{2})} \right\} = -[x(i+1) - x(i)]E(i + \frac{1}{2}) \quad (8-22)$$

所以电子电流密度为

$$j_n(i + \frac{1}{2}) = -\mu_n(i + \frac{1}{2}) \frac{\Delta \psi(i+1)}{\Delta x(i+1)} \left[\frac{n(i+1)}{1 - e^{-\Delta \psi(i+1)}} + \frac{n(i)}{1 - e^{-\Delta \psi(i+1)}} \right] \quad (8-23)$$

式中, $\Delta x(i+1) = x(i+1) - x(i)$;

$$\Delta \psi(i+1) = \psi(i+1) - \psi(i) = -E(i + \frac{1}{2})\Delta x(i+1)$$

再利用伯努利 (Bernoulli) 函数

$$B(x) = \frac{x}{e^x - 1} \quad (8-24)$$

代入 (8-23) 式, 得电子电流密度

$$j_n\left(i+\frac{1}{2}\right) = \frac{\mu_n\left(i+\frac{1}{2}\right)}{\Delta x(i+1)} \{B[\Delta\psi(i+1)] \cdot n(i+1) - B[-\Delta\psi(i+1)] \cdot n(i)\} \quad (8-25)$$

根据同样过程，我们可得空穴电流密度

$$j_p\left(i+\frac{1}{2}\right) = \frac{\mu_p\left(i+\frac{1}{2}\right)}{\Delta x(i+1)} \{-B[-\Delta\psi(i+1)] \cdot p(i+1) + B[\Delta\psi(i+1)] \cdot p(i)\} \quad (8-26)$$

从公式 (8-23) 可以看出，当 $E \rightarrow 0$ 时

$$j_n\left(i+\frac{1}{2}\right) \rightarrow \mu_n\left(i+\frac{1}{2}\right) \frac{n(i+1) - n(i)}{\Delta x(i+1)} \quad (8-27)$$

即 j_n 只取决于扩散项。当 $E \rightarrow \infty$ 或 $-\infty$ 时

$$\begin{aligned} j_n\left(i+\frac{1}{2}\right) &\rightarrow n(i+1) \cdot \mu_n(i+1) \cdot E\left(i+\frac{1}{2}\right) & E \rightarrow \infty \text{ 时} \\ j_n\left(i+\frac{1}{2}\right) &\rightarrow n(i) \cdot \mu_n(i+1) \cdot E\left(i+\frac{1}{2}\right) & E \rightarrow -\infty \text{ 时} \end{aligned} \quad (8-28)$$

即漂移项成为主要因素。

在所有条件下，该离散化方程是稳定的。当两格点间的电位差很小时，该表达式变成标准的有限差分形式，电流只取决于扩散项。当电位差很大时，就接近漂移电流的表达式。由于它的极好的稳定性，目前几乎所有的器件模拟程序中都采用这一离散化方程。

连续方程对时间的离散化可以采用后向欧拉法，因而有

$$\left. \begin{aligned} \frac{n(i, t+\Delta t) - n(i, t)}{\Delta t} &= \frac{j_n\left(i+\frac{1}{2}, t+\Delta t\right) - j_n\left(i-\frac{1}{2}, t+\Delta t\right)}{\frac{1}{2}[\Delta x(i+1) + \Delta x(i)]} - U(i, t+\Delta t) \\ \frac{p(i, t+\Delta t) - p(i, t)}{\Delta t} &= -\frac{j_p\left(i+\frac{1}{2}, t+\Delta t\right) - j_p\left(i-\frac{1}{2}, t+\Delta t\right)}{\frac{1}{2}[\Delta x(i+1) + \Delta x(i)]} - U(i, t+\Delta t) \end{aligned} \right\} \quad (8-29)$$

式中， $n(i, t+\Delta t)$ 是在位置 $x(i)$ 和时间 $t+\Delta t$ 时的电子浓度； $p(i, t+\Delta t)$ 是在位置 $x(i)$ 和时间 $t+\Delta t$ 时的空穴浓度。

在离散化后，泊松方程和连续方程就共同构成一个非线性差分方程组，总方程数为 $3N$ 个 (N 为格点数)。如在格点 1 和格点 N 处的 ψ 、 n 、 p 为已知，则总方程数为 $3(N-2)$ 个，且有 $3(N-2)$ 个变量。

三、边界条件

器件的边界通常包括欧姆接触、绝缘材料界面和反射边界。

对于欧姆接触，我们假设符合热平衡状态和电中性条件，即有

$$np = n_i^2 \quad (8-30)$$

$$n - p - N = 0 \quad (8-31)$$

求解上述方程, 可得欧姆接触处的载流子浓度 n_s 和 p_s

$$n_s = \frac{\sqrt{N^2 + 4n_i^2} + N}{2} \quad (8-32)$$

$$p_s = \frac{\sqrt{N^2 + 4n_i^2} - N}{2} \quad (8-33)$$

由于热平衡, 故准费米势 $\phi_n = \phi_p$ 。对于玻尔兹曼统计, 有

$$n = n_i e^{q(\phi - \phi_n)/(kT)} \quad (8-34)$$

$$p = n_i e^{q(\phi_p - \phi)/(kT)} \quad (8-35)$$

如取 $\phi_p = \phi_n = \phi$ 且等于零, 就可得到欧姆接触处的静电势

$$\psi_s = \frac{kT}{q} \ln \frac{n_s}{n_i} = -\frac{kT}{q} \ln \frac{p_s}{n_i} \quad (8-36)$$

对于绝缘界面 (半导体和绝缘材料界面), 电位移满足下列公式

$$\mathbf{n} \cdot (\epsilon_s \nabla \psi)_s - \mathbf{n} \cdot (\epsilon_i \nabla \psi)_i = Q_{int} \quad (8-37)$$

式中, $\mathbf{n}$ 为从硅向外的单位法向矢量; ϵ_s 、 ϵ_i 分别为半导体和绝缘层的介电常数; Q_{int} 为单位界面电荷。

通常不解绝缘层中的泊松方程, 而用一垂直于界面的电位降来代替, 故 (8-37) 式改为

$$\mathbf{n}(\epsilon_i \nabla \psi)_i - \epsilon_i \frac{\psi_s - \psi}{d_i} = Q_{int} \quad (8-38)$$

式中, ψ_s 为栅极的静电势; d_i 为绝缘层厚度。

对于绝缘界面, 假设在界面处存在表面复合, 且表面复合速率为 R_s , 则电流密度应服从

$$\mathbf{j}_n \cdot \mathbf{n} = -qR_s \quad (8-39)$$

$$\mathbf{j}_p \cdot \mathbf{n} = qR_s \quad (8-40)$$

除欧姆接触区和绝缘界面区外, 器件的其它边界都假设为反射边界, 这时符合

$$\mathbf{j}_n \cdot \mathbf{n} = \mathbf{j}_p \cdot \mathbf{n} = \nabla \psi \cdot \mathbf{n} = 0 \quad (8-41)$$

即电流只能从接触区流进或流出。这种假设会引起一定的误差, 但只要反射边界远离器件的有源区, 这种误差可减小到被忽略的程度。

现就 NPN 双极型晶体管为例说明边界条件的具体运用。

双极型晶体管有三个外边界, 分别处于 $x=0$ (发射区边界), $x=L$ (集电区边界) 和 $x=B$ (基区边界)。假设发射区接触处取零电位, 且发射极与集电极的准费米能级之差等于两端外加偏压 V_{CB} , 因而有

$$\phi_p(0) = \phi_n(0) = 0 \quad (8-42)$$

$$\phi_p(L) = \phi_n(L) = V_{CB} \quad (8-43)$$

这样外边界处的静电势和载流子浓度就可以表示为外加偏压和杂质浓度的函数。发射区边界有

$$n(0) = n_i(0) e^{q\phi(0)/(kT)} \quad (8-44)$$

$$p(0) = n_i(0) e^{-q\phi(0)/(kT)} \quad (8-45)$$

$$\psi(0) = \frac{kT}{q} \ln \left\{ \sqrt{1 + \left[\frac{N(0)}{2n_i(0)} \right]^2} + \frac{N(0)}{2n_i(0)} \right\} \quad (8-46)$$

收集区边界有

$$n(L) = n_i(L) e^{q(\psi(L) - V_{CB})/(kT)} \quad (8-47)$$

$$p(L) = n_i(L) e^{q(V_{CB} - \psi(L))/(kT)} \quad (8-48)$$

$$\psi(L) = V_{CB} + \frac{kT}{q} \ln \left\{ \sqrt{1 + \left[\frac{N(L)}{2n_i(L)} \right]^2} + \frac{N(L)}{2n_i(L)} \right\} \quad (8-49)$$

在基区接触处要求多子浓度的准费米势应等于外加的发射极-基极偏压 V_{BE} ，即在格点 B 处有

$$\phi_p(B) = V_{BE} \quad (8-50)$$

因而基区接触处空穴浓度有

$$p(B) = n_i(B) e^{q(V_{BE} - \psi(B))/(kT)} \quad (8-51)$$

如果在基极端点与内部基区接触处之间存在外部基极电阻，则应考虑基极电阻上的压降

$$j_p R_B = \phi_p(B) - V_{BE} \quad (8-52)$$

这时基区接触处多子的连续方程应包括横向流动的电流 j_p 。

四、基本变量的选择

至今有三组基本变量可供选择。最通常是采用直接出现在基本方程中的变量，即 ψ 、 n 和 p 。它的优点是：经过离散化后的差分方程，对方程的基本变量来说具有对称、正定的系数矩阵。其主要缺点是：变量的变化范围过大，从几个数量级甚至到十几个数量级的变化。因为 ψ 线性变化时，载流子浓度将按指数规律变化。

第二组基本变量是 ψ 、 ϕ_n 和 ϕ_p 。选用这组变量的明显优点是它们都按线性变化，变化范围均匀。但是除泊松方程保留对称、正定系数矩阵外，连续方程将失去这一特点。

第三组基本变量是 ψ 、 u 和 v ，常被称作 Slotboom 变量。 u 和 v 分别被定义为

$$u = e^{-q\phi_n/(kT)} \quad (8-53)$$

$$v = e^{q\phi_p/(kT)} \quad (8-54)$$

变量 ψ 、 u 、 v 与变量 ψ 、 n 、 p 之间有下列关系

$$n = n_i u e^{q\psi/(kT)} \quad (8-55)$$

$$p = n_i v e^{-q\psi/(kT)} \quad (8-56)$$

这组变量在本质上与第一组变量具有相同的特点，但在某种情况下具有较好的收敛性质。

五、典型算法

前面谈到，泊松方程和连续性方程在离散化后得到一组非线性代数方程组。目前，求解这组非线性代数方程组的方法主要有两种：一种为解耦法（或称 Gummel 法），它是对泊松方程和连续性方程依次顺序求解，再重复循环直到收敛。另一种为耦合法（或称 Newton 法）。它是将三个方程作为一个整体，并用牛顿-拉夫森法进行求解。其示意图见图 8-5。

解耦法是一种简单迭代法。由于三个方程分别求解，因而形成的系数矩阵其稀疏性较强，并可采用迭代算法求解线性方程组（例如采用高斯-塞得尔法，逐次超松弛法等）。另外，它在单次迭代时所用的计算量和存储量都较小，这在一定程度上可以补偿收敛速度较慢的缺点。

由于解耦法求解基本方程时是把它们作为非耦合的, 因此这种方法仅适合于弱耦合的情况, 譬如双极型晶体管小注入的时候。实验还指出, 在低电压小电流时, 解耦法有较快的收敛速度, 但随着电压增大因而电流加大时, 收敛速度减慢, 在大注入时甚至不收敛。

实际上半导体基本方程是相互耦合的, 应该作为一个整体同时求解。这就是采用耦合法的原因。耦合法是一种有效和可靠的方法, 它能求解小注入时的方程, 也能适用于大注入条件。而且它的收敛速度很快, 即有二次方的收敛速度。但由于同时解三个方程, 所得到的系数矩阵其稀疏性较差, 一些快速的迭代算法难于求解线性方程组, 通常采用高斯消去法或LU分解法。耦合法的缺点是每次迭代所用的计算量和存储量都较大, 并且对初始值要求很高, 如初值不合适, 开始的几次迭代无法达到二次方的收敛速度, 甚至会出现不收敛的现象。

有人提出将这两种方法结合起来, 先用解耦法求解, 当进入到耦合法具有二次方收敛速度的区域时, 再用耦合法以充分利用两者的优点。

六、耦合法(牛顿法)

现以第一组基本变量 ψ 、 n 和 p 为例, 说明耦合法的求解过程。

将公式(8-20)、(8-29)重新排列, 并定义

$$F_\psi = \frac{\psi(i+1)}{\Delta x_2} - \left(\frac{1}{\Delta x_1} + \frac{1}{\Delta x_2} \right) \psi(i) + \frac{\psi(i-1)}{\Delta x_1} + p(i) - n(i) + N(i) \quad (8-57)$$

式中, $\Delta x_1 = \Delta x(i) \frac{\Delta x(i+1) + \Delta x(i)}{2}$; $\Delta x_2 = \Delta x(i+1) \frac{\Delta x(i+1) + \Delta x(i)}{2}$ 。

$$\left. \begin{aligned} F_n &= \frac{j_n\left(i + \frac{1}{2}, t_1\right) - j_n\left(i - \frac{1}{2}, t_1\right)}{\frac{1}{2}[\Delta x(i+1) + \Delta x(i)]} - U(i, t_1) - \frac{n(i, t_1) - n(i, t_0)}{\Delta t} \\ F_p &= -\frac{j_p\left(i + \frac{1}{2}, t_1\right) - j_p\left(i - \frac{1}{2}, t_1\right)}{\frac{1}{2}[\Delta x(i+1) + \Delta x(i)]} - U(i, t_1) - \frac{p(i, t_1) - p(i, t_0)}{\Delta t} \end{aligned} \right\} \quad (8-58)$$

式中, $t_1 = i + \Delta t$; $t_0 = t$; $\Delta t = t_1 - t_0$ 。

现利用牛顿-拉夫森法对 $F(\psi_h, n_h, p_h) = 0$ 的非线性差分方程组进行线性化, 得到

$$J^* \begin{bmatrix} d\psi_i \\ dn_i \\ dp_i \\ \vdots \end{bmatrix} = - \begin{bmatrix} F_{\psi_i}^* \\ F_{n_i}^* \\ F_{p_i}^* \\ \vdots \end{bmatrix} \quad i=2, 3, \dots, N-1 \quad (8-59)$$

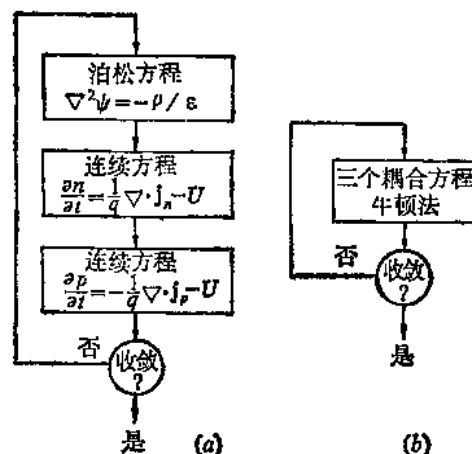


图 8-5 两种典型的器件模拟算法

(a) 解耦法; (b) 耦合法

这里, J^k 是 $F(\psi_k, n_k, p_k)$ 的雅可比矩阵,

$d\psi_i, dn_i, dp_i$ 是 ψ_i, n_i, p_i 的第 $k+1$ 次迭代结果与第 k 次迭代结果之差,

$F_{n,i}^k, F_{p,i}^k, F_{\psi,i}^k$ 是第 k 次迭代时的 $F_{n,i}, F_{p,i}, F_{\psi,i}$ 值。

(8-59) 线性方程组可以采用高斯消去法或 LU 分解法求解, 得到第 $k+1$ 次的 ψ_i, n_i 和 p_i 。

七、解耦顺序法 (Gummel 法)

现以第三组基本变量 ψ, u 和 v 为例, 说明解耦法的求解过程。

为了计算方便, 现采用以下归一化方程

$$-\lambda^2 \nabla^2 \psi + e^{\psi} u - e^{-\psi} v - N = 0 \quad (8-60)$$

$$\frac{\partial n}{\partial t} - \nabla \cdot (\mu_n e^{\psi} \nabla u) + U = 0 \quad (8-61)$$

$$\frac{\partial p}{\partial t} - \nabla \cdot (\mu_p e^{-\psi} \nabla v) + U = 0 \quad (8-62)$$

$$j_n = \mu_n e^{\psi} \nabla u$$

$$j_p = -\mu_p e^{-\psi} \nabla v$$

式中, $\lambda^2 = eV_T / (q l^2 C_0)$ (l 为器件的线度, C_0 为最大净掺杂浓度); $u = e^{-\psi} n, v = e^{\psi} p$ 。归

一化时所采用的系数见表 8-1。其它物理量的归一化系数可以从以上五个归一化因子中导出。

表 8-1 归一化系数

变 量	归 一 化 因 子
电 势 ψ	$V_T = kT/q$
浓度 n, p	C_0 (最大净掺杂浓度)
长 度	l (器件的线度)
迁移率	μ
时 间	$\tau = l^2 / (\mu V_T)$

为简单起见, 现只考虑稳态情况。公式 (8-60)、(8-61)、(8-62) 可改写为

$$-\lambda^2 \nabla^2 \psi + e^{\psi} u - e^{-\psi} v - N = 0 \quad (8-60a)$$

$$\nabla \cdot (\mu_n e^{\psi} \nabla u) - U = 0 \quad (8-61a)$$

$$\nabla \cdot (\mu_p e^{-\psi} \nabla v) - U = 0 \quad (8-62a)$$

现利用牛顿-拉夫森法对其线性化, 得到线性方程

$$-\lambda^2 \nabla^2 d\psi + (n + p) d\psi + e^{\psi} du - e^{-\psi} dv = -b_1 \quad (8-63)$$

$$\nabla \cdot (j_n d\psi + \mu_n e^{\psi} \nabla du) = -b_2 \quad (8-64)$$

$$\nabla \cdot (j_p d\psi + \mu_p e^{-\psi} \nabla dv) = -b_3 \quad (8-65)$$

上列方程的右端

$$b_1 = -\lambda^2 \nabla^2 \psi + e^{\psi} n - e^{-\psi} p - N \quad (8-66)$$

$$b_2 = \nabla \cdot (\mu_n e^{\psi} \nabla u) - U \quad (8-67)$$

$$b_3 = \nabla \cdot (\mu_p e^{-\psi} \nabla v) - U \quad (8-68)$$

这里我们忽略了 U 对 ψ, u 和 v 的偏导数。

有人对解耦法进行了修正。原来的解耦法是同等处理电子连续方程和空穴连续方程, 现在认为器件中以一种载流子电流为主 (譬如说 $j_n \gg j_p$), 这种假设对于 MOSFET 管是合适的。由于忽略了 $j_p d\psi$ 项, 公式 (8-65) 就改写成

$$\nabla \cdot (\mu_p e^{-\psi} \nabla dv) = -b_3 \quad (8-69)$$

dv 就可直接解出。

再考虑到 λ 很小, 通过忽略 (8-63) 式中的 $\lambda^2 \nabla^2 d\psi$ 项而得到 $d\psi$ 的近似值

$$d\psi \approx \frac{e^{-v} dv - e^v du - b_1}{n + p} \quad (8-70)$$

将 (8-70) 式代入 (8-64) 式后, (8-64) 式可改写为

$$\nabla \cdot \left(-j_n \frac{e^v}{n + p} du + \mu_n e^v \nabla du \right) = -b_2 - \nabla \cdot \left(j_n \frac{e^{-v} dv - b_1}{n + p} \right) \quad (8-71)$$

这样我们就把顺序求解 (8-63)、(8-64)、(8-65) 的过程改为依次求解 (8-69)、(8-71) 和 (8-63) 式。

其步骤如下:

(1) 给定 ψ , u , v 的初始值;

(2) 计算 b_1 , b_2 , b_3 ;

(3) 解

$$\nabla \cdot (\mu_p e^{-v} \nabla dv) = -b_3$$

(4) 解

$$\nabla \cdot \left(-j_n \frac{e^v du}{n + p} + \mu_n e^v \nabla du \right) = -b_2 - \nabla \cdot \left(j_n \frac{e^{-v} dv - b_1}{n + p} \right)$$

(5) 解

$$-\lambda^2 \nabla^2 d\psi + (n + p) d\psi = -b_1 + e^{-v} dv - e^v du$$

(6) 修改

$$\psi \rightarrow \psi + d\psi, \quad u \rightarrow u + du, \quad v \rightarrow v + dv$$

重复 (2)~(6) 步骤, 直到符合收敛条件为止。

§ 8.4 物理参数的模型

一、迁移率

它是载流子在晶格中受到散射的一种度量。它取决于温度 (声子散射)、掺杂浓度 (由于离化杂质和缺陷) 及电场强度。其关系式如下

$$\mu(C_T, T, E) = \frac{\mu_0(C_T, T)}{\left\{ 1 + \left[\mu_0(C_T, T) \frac{|E|}{v_{SAT}} \right]^\beta \right\}^{1/\beta}} \quad (8-72)$$

这里, β 为常数。对电子, $\beta_n = 2$ 。对空穴, $\beta_p = 1$;

v_{SAT} 为极限漂移速度;

$\mu_0(C_T, T)$ 为低电场时的迁移率, 它是总杂质浓度 C_T 和温度 T 的函数

$$\mu_0(C_T, T) = \mu_{min} + \frac{\mu_0}{1 + (C_T/C_{ref})^\alpha} \quad (8-73)$$

式中 μ_{min} , μ_0 , C_{ref} 和 α 都是温度的函数。这些参数值见表 8-2。

表中的 $T_n = T/(300K)$, T 的单位为绝对温度 K。

表 8-2 迁移率各有关参数值

参 数	μ_{min} ($\text{cm}^2/\text{V}\cdot\text{s}$)	μ_0 ($\text{cm}^2/\text{V}\cdot\text{s}$)	C_{ref} (cm^{-3})	α
电 子	$88T_n^{-0.57}$	$7.4 \times 10^5 T^{-2.32}$	$1.26 \times 10^{17} T_n^{2.4}$	$0.88 T_n^{-0.146}$
空 穴	$64.3 T_n^{-0.57}$	$1.36 \times 10^4 T^{-2.32}$	$2.35 \times 10^{17} T_n^{2.4}$	$0.88 T_n^{-0.146}$

二、能带变窄

随着杂质浓度的增加, 约在杂质浓度 $> 10^{17} \text{cm}^{-3}$ 时, 晶格的周期性会受到破坏。对这种影响我们用能带变窄来加以等效。具体公式如下

$$E_g(C) = E_{g0} - V_1 \left(\ln \frac{C}{C_0} + \sqrt{\ln^2 \left(\frac{C}{C_0} \right) + A} \right) \quad (8-74)$$

式中, E_{g0} 为本征材料的能带宽度; C 为掺杂浓度; V_1 为 9meV ; C_0 为 $(1 \sim 1.5) \times 10^{17} \text{cm}^{-3}$; A 等于 0.5 。

三、载流子复合与寿命

在器件模拟中, 我们通常考虑两种复合机构: 即复合中心复合 (称 Shockley-Read-Hall 复合) 和三粒子复合 (称 Auger 复合)。

在单能级陷阱, SRH 的复合速率为

$$U_{\text{SRH}} = \frac{n p - n_i^2}{\tau_p (n + n_1) + \tau_n (p + p_1)} \quad (8-75)$$

式中, $n_1 = n_i \exp \left[\frac{q}{kT} \left(\frac{E_t}{q} - \psi \right) \right]$,

$$p_1 = n_i \exp \left[\frac{q}{kT} \left(\psi - \frac{E_t}{q} \right) \right],$$

E_t 是陷阱能级与本征费米能级之差,

τ_p, τ_n 是空穴与电子寿命。

三粒子复合过程是碰撞离化的逆过程, 它在空穴-电子对复合的同时把过剩的能量传递给第三个载流子。这种过程由下式决定

$$U_A = (C_n n + C_p p) (n p - n_i^2) \quad (8-76)$$

式中, C_n, C_p 是对应于电子和空穴俘获系数的实验系数。总的复合速率是上两者之和

$$U = U_{\text{SRH}} + U_A \quad (8-77)$$

实验表明, 少子寿命强烈地依赖于杂质浓度。经验上有以下关系

$$\tau = \frac{T_0}{1 + (C_x/C_0)} \quad (8-78)$$

对于电子和空穴, τ_0, C_0 和 $C_n(C_p)$ 典型值见表 8-3。

表 8-3 τ_0 , C_0 , $C_n(C_p)$ 典型值

	τ_0 (s)	C_0 (cm ⁻⁴)	$C_n(C_p)$ (cm ³ /s)
电 子	5×10^{-7}	5×10^{16}	$(0.5 \sim 2.8) \times 10^{-21}$
空 穴	5×10^{-7}	5×10^{16}	9.9×10^{-22}

§ 8.5 SEDAN II 器件模拟程序及其应用

SEDAN II 是一维器件模拟程序。虽然它只是求解一维空间,但它能够模拟用不同材料制备的器件,把应用范围从硅材料推广到化合物半导体,它还能模拟多层结构(多晶硅-氧化层-硅)。SEDAN II 中除了考虑玻尔兹曼统计外,还考虑了费米-狄拉克统计。

SEDAN II 中选用 Ψ 、 n 和 p 作为基本变量,并采用耦合法同时求解泊松方程和连续性方程。在离散化时采用盒式(Box)离散法。对非线性代数方程的线性化是利用牛顿-拉夫森法,求解线性方程时则采用高斯消元法。

一、基本方程的形式

为了反映非均匀能带结构以及费米-狄拉克统计的影响,SEDAN II 中对基本方程作了必要的更动。

泊松方程有

$$\frac{\partial}{\partial x} \left[-\epsilon \frac{\partial}{\partial x} (\Psi - \theta) \right] = q(p - n + N_D^+ - N_A^-) \quad (8-79)$$

这里 Ψ 称为广义电势,它被定义为

$$-q\Psi = \frac{1}{2} \left(E_c + E_v - kT \ln \frac{N_c}{N_v} \right) \quad (8-80)$$

式中, E_c , E_v 分别为导带底和价带顶能量; N_c , N_v 分别为导带中电子有效态密度和价带中空穴有效态密度。

在非均匀能带结构中, Ψ 并不等于实际的静电势。(8-79) 式中的 θ 为能带结构参数

$$\theta = \frac{1}{q} \left[\frac{1}{2} \left(E_g + kT \ln \frac{N_c}{N_v} \right) + \chi \right] \quad (8-81)$$

其中 χ 为电子亲和力。如果 θ 在整个空间是一常数,即 E_g , N_c , N_v 和 χ 都与空间位置无关,则 (8-79) 式简化为通常的泊松方程。

连续方程有

$$\frac{\partial n}{\partial t} = \frac{1}{q} \frac{\partial j_n}{\partial x} - R + G \quad (8-82)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \frac{\partial j_p}{\partial x} - R + G \quad (8-83)$$

载流子输运方程有

$$j_n = kT\mu_n \frac{\partial n}{\partial x} + qn\mu_n \left[-\frac{\partial}{\partial x} \left(\psi + \frac{kT}{q} \ln n_{in} \right) \right] \quad (8-84)$$

$$j_p = -kT\mu_p \frac{\partial p}{\partial x} + qp\mu_p \left[-\frac{\partial}{\partial x} \left(\psi - \frac{kT}{q} \ln n_{ip} \right) \right] \quad (8-85)$$

式中

$$n_{in} = n_i \gamma_n \quad (8-86)$$

$$n_{ip} = n_i \gamma_p \quad (8-87)$$

γ_n 和 γ_p 是考虑费米-狄拉克统计后的简并因子，它们分别定义为

$$\gamma_n = \frac{F_{1/2}(\eta_n)}{\exp(\eta_n)} \quad (8-88)$$

$$\gamma_p = \frac{F_{1/2}(\eta_p)}{\exp(\eta_p)} \quad (8-89)$$

式中， $F_{1/2}$ 为 1/2 阶的费米积分， $\eta_n = \frac{E_{Fn} - E_c}{kT}$ ， $\eta_p = \frac{E_v - E_{Fp}}{kT}$ ， E_{Fn} ， E_{Fp} 分别为电子和空穴的费米能级。对于玻尔兹曼统计 γ_n 和 γ_p 都等于 1，而对于费米-狄拉克统计，它们的值总是小于 1。

应该指出，当能带结构是均匀的情况，即 E_g ， N_a ， N_v 和 χ 这些与能带结构有关的参数都是常数时，上列方程就简化为通常的输运方程。

二、盒式离散方法

盒式离散方法是另一种对微分方程的差分近似方法。它是将原来的微分方程先求积分，然后用数值方法来近似该积分方程，从而建立差分方程。也就是说采用积分形式（而不是微分形式）来描述基本的物理定律，然后将这种表述应用到离散了的网格空间中的每一个子区间，以逐个形成差分方程。

对于半导体器件的基本方程来讲，除微分形式外，另一种描述形式就是积分形式。而这种描述形式所要阐明的基本物理含义是完全相同的。譬如泊松方程为描述静电系统的微分形式，其对应的积分形式就是高斯定律。同样对于连续性方程，除前面已讨论的微分形式外，其对应的积分形式为连续性定律。

现以高斯定律为例说明如何进行空间离散化。为简单起见，现讨论一维情况。选取一个包含第 i 个格点的子区间，该封闭区间为 $\left[\frac{x_{i-1} + x_i}{2}, \frac{x_i + x_{i+1}}{2} \right]$ 。我们对该区间应用高斯定律就得到

$$e_{i+\frac{1}{2}} E_{i+\frac{1}{2}} - e_{i-\frac{1}{2}} E_{i-\frac{1}{2}} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \rho(x) dx \quad (8-90)$$

在离散空间中，一般格点间的间距选得足够小，以致我们可以假设多数物理量在一个子区间中是一常数。现假设电荷密度在 $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right]$ 区间里是均匀分布的，且等于 ρ_i ，则 (8-90) 式的右边项为

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \rho(x) dx = \rho_i \frac{\Delta x_{i+1} + \Delta x_i}{2} \quad (8-91)$$

公式左边项中的电场强度则很容易采用 x_{i-1} , x_i , x_{i+1} 处的电势值来表示。现采用中心差分法可得

$$\varepsilon_{i+\frac{1}{2}} E_{i+\frac{1}{2}} - \varepsilon_{i-\frac{1}{2}} E_{i-\frac{1}{2}} = \varepsilon_{i+\frac{1}{2}} \frac{\psi(i) - \psi(i+1)}{\Delta x_{i+1}} - \varepsilon_{i-\frac{1}{2}} \frac{\psi(i-1) - \psi(i)}{\Delta x_i} \quad (8-92)$$

于是我们就得到该子区间中高斯定律的离散化方程

$$\varepsilon_{i+\frac{1}{2}} \frac{\psi(i) - \psi(i+1)}{\Delta x_{i+1}} - \varepsilon_{i-\frac{1}{2}} \frac{\psi(i-1) - \psi(i)}{\Delta x_i} = \rho_i \frac{\Delta x_{i+1} + \Delta x_i}{2} \quad (8-93)$$

如对被模拟区间中的每一个格点都可以写成这样的方程, 就得到了一组所要求的求解代数方程。

可以看出, (8-93) 式与对微分形式求得的离散化方程 (8-20) 是完全一样的。但是应该指出, 式 (8-93) 的右端项是一积分近似, 如果 ρ 在 x_i 处不连续, 或需要更高的近似精度时, 就可以利用积分得到右端项。但在 (8-20) 式中则要求 ρ_i 在 x_i 处是唯一的, 这两者是不相同的。

如果我们把每一个积分区间都看成为一个盒子, 从以上分析可知, 为求解 (8-90) 式, 我们只需求得盒子边上的物理量 (在这里就是 E), 及盒子内部另一物理量 (在这里为 ρ) 的总含量, 即积分值即可。这时不必考虑盒内物理量的变化梯度 (微分), 这就是称这种离散方法为盒式法的由来。

以上谈到的 ρ 在 x_i 处不连续的问题在实际中是存在的, 例如异质结的结面。这种不连续问题在其它情况中也屡有所见, 如材料的突变界面上, 譬如硅和二氧化硅界面、HEMT (高电子迁移率晶体管) 中高掺杂的 AlGaAs 和未经掺杂的 GaAs 沟道区的界面, 由于材料的介电常数不同, 电场在垂直于界面的方向上将不再连续, 如果界面上还存在面电荷, 则电位移矢量的垂直分量也不再连续。在盒式法中, 由于在积分形式中允许被积函数在某些孤立点、线上导数不存在甚至函数本身不连续, 因此在处理非连续材料界面问题上, 处理有关物理量的导数不存在或本身不连续时有着相当的灵活性, 这就是在数值分析中, 采用积分形式的明显好处。

由于盒式法具有物理意义清楚、数学处理比较简单等优点, 因而已成为器件和工艺模拟中最为常用的一种离散方法。

三、网格结构

SEDAN II 把求解空间 (沿器件的深度方向) 划分成若干个格点。总的格点数不能超过 401 个。

网格可以自动地或人工地产生。在人工产生网格时, 总的网格空间可以分解成若干子区 (最多可以有 10 个子区)。同一子区内的格点间距是均匀的, 不同子区中的格点间距可以是不等的。因而人工生成网格时, 用户必须规定子区的序号、每个子区内的网格数以及设置每个子区内网格的格点间距。

当网格由程序自动产生时, 在模拟区内产生的是非均匀网格。生成网格的算法为: 首先把格点设置在物理结或界面/晶粒边界处, 且规定这个格点处的网格最小。而与结的距离或与界面的距离越远, 则网格越来越增大, 且按几何级数的规律增大, 也就是说

$$d_{i+1} = r d_i \quad (8-94)$$

式中, d 为格点间距; i 为序号; r 为比例因子 (它大于 1), 由用户给定或取隐含值。

初始的最小网格间距根据被模拟器件的类型由两种不同方法产生。

对于具有多晶硅发射极的晶体管, 在多晶硅层与衬底之间存在一个有一定厚度的界面, 或者既存在界面也存在着晶粒边界。这时, 取界面和晶粒边界的厚度作为初始的格点间距。由于它的厚度很小 (例如厚度为 1.75nm), 因而比例因子 r 不能选得太小, 不然会使总的格点数超过 401 个。

对于通常的结型器件 (二极管和晶体管) 和 MOS 电容器, 初始格点间距取决于该层的层厚、分配给该层的网格数及给定的比例因子。它的值由下式决定

$$d_0 = \frac{r-1}{r^n-1} W \quad (8-95)$$

式中, d_0 为初始的最小的格点间距; W 为该层厚度; r 为比例因子; n 为分配给该层的网格数。

SEDAN II 中设置了两个比例因子。一个用来作为晶体管发射区和基区的比例因子, 也作为二极管第一层、MOS 结构中衬底的比例因子, 另一个用来作为晶体管集电区的比例因子。

应注意, 各层中格点数的总和不能超过 401 个, 不然就引起出错。

四、输入文件格式

SEDAN II 中规定有 15 种语句, 按其用途可分为三类。

第一类: title comment end

第二类: material grid device profile composit lifetime bias model transien
compute

第三类: print plot

各语句的排列次序有一定的要求, 即: (1)title; (2)material; (3)device; (4)

除 (1)(2)(3)(5) 以外的其它语句; (5)end。

语句由语句名和一系列参量定义符所组成。参量定义符由参量名、等号和参量值三者构成一个整体。参量值可以有三种类型, 即逻辑型、字符型和数字型。

对逻辑型参量, 如果出现该参量名, 则其值为真, 也可以在等号后加一 Y 以表示其值为真。如不出现该参量名, 则其值取隐含值。

对字符型参量, 一种是在参量名的等号后面加一字符串, 另一种是字符串本身就是参量定义符而不再需要参量名。

对数字型参量, 是通过等号赋以所需数值。

如某一语句的输入多于一行, 则继续行的第一列必须加一“+”符号。

器件中的杂质分布可以有三种办法加以确定。一是将工艺模拟程序的输出文件直接送入器件模拟程序 SEDAN II。可以直接连接的工艺模拟程序有 SUPREM I, SUPREM II 和 SUPRA。二是由用户指定杂质分布的解析表达式。三是给出一数据文件, 该文件由深度及与其对应的净掺杂浓度两列数据组成, 数据是以 ASCII 形式出现的。

SEDAN II 的输出语句有打印语句和绘图语句。打印语句控制有关信息及模拟结果的打印。绘图语句规定不同变量的绘图格式。有两类基本作图类型。一类是随深度而变的, 另一类是随偏压而变的。后一类作图, 只能在整个模拟完成后才能得到。而随深度变化的

变量，其作图可以在一个偏压条件计算完成后得到。

§ 8.6 SEDAN III 程序应用举例

此例取 SUPREM III 的输出文件作为输入，以确定器件中的杂质分布。模拟时的输入文件见表 8-4。

表 8-4 SEDAN III 输入文件

title	Analysis of normal npn transistor
comment	profile input from SUPREM III export file
material	si
device	type=trans ferm=y temp=300
grid	autog depth=3.5 nemit=50 sf1=1.01 sf2=1.02
profile	suprem3 file=mls3
bias	vbef=0.0 vbel=0.8 vbes=0.1
+	vcef=0.0 vcel=3.0 vces=1.5
comment	smaller bias step for higher bias
bias	vbef=0.85 vbel=0.95 vbes=0.05 vcef=3.0 vcel=3.0
model	srhr=y auge=y bgnw=y
compute	ft
print	head=y site=y sout=y screen=n
plot	vbe=0.0 vce=0.0 conc poten jc jb beta
end	

模拟后的输出有杂质浓度分布、Gummel 图和电流增益与收集极电流密度的关系曲线，现分别示于图 8-6，图 8-7 和图 8-8。图中 n^+ , n , p 的单位为 cm^{-3} , J_B, J_C 单位为 A/cm^2 。

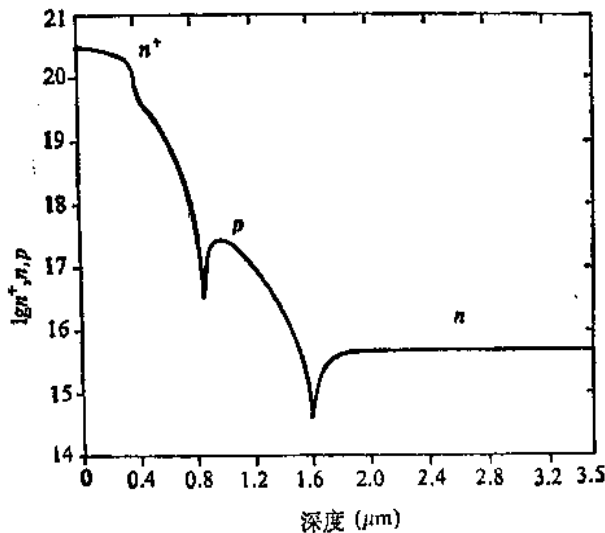


图 8-6 杂质浓度分布

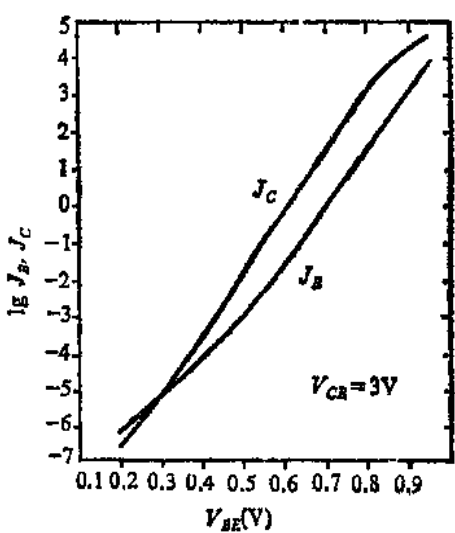


图 8-7 Gummel 图

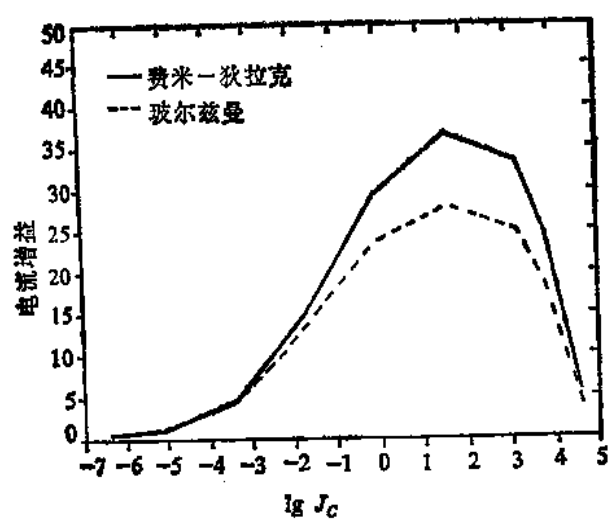


图 8-8 电流增益随收集极电流的变化曲线

第九章 工 艺 模 拟

§ 9.1 工艺模拟的作用

工艺模拟已成为半导体器件和集成电路工艺设计中的一种重要工具。

为了得到高性能、高可靠的集成电路，需要合理地选择和调整各工艺过程，选择最佳的工艺条件，即进行工艺优化。过去只能采用“实验”法即采取试流水的方法，进行多次反复的试验，但这种方法既费时又费钱，有时还得不到合理的结果。

随着计算机技术和计算方法的发展，以及我们对于各工艺的基本物理过程的深入了解，现在已可能将要分析的工艺过程列出数学形式的模型，或者列出比较公认的经验公式，这样就可以利用计算机进行数值求解。这时不再需要实际的工艺设备和准备各种实验条件，就能对各种不同的工艺条件进行模拟分析。因此，近年来工艺模拟技术得到了飞速发展。当然，模拟的精度在很大程度上取决于工艺模型的数学表达式是否正确，以及采用何种算法。

作为一个好的工艺模拟程序，应该能够对于任何工艺制造步骤，计算出器件内部的结构特性和杂质分布。结构特性就是设法跟踪不同材料层在位置上的变化，及由于工艺过程所引起的变形，这包括各材料层的厚度和宽度的变化。杂质分布是求出杂质粒子在半导体内部的传输和再分布。同时可以预测各工艺参数的偏差对工艺结果的影响程度。

如果进一步把工艺模拟结果与器件模拟甚至电路模拟结合起来，就可以得到该工艺条件下的器件性能以及电路性能，这是一项十分吸引人的工作。这示于图 9-1。

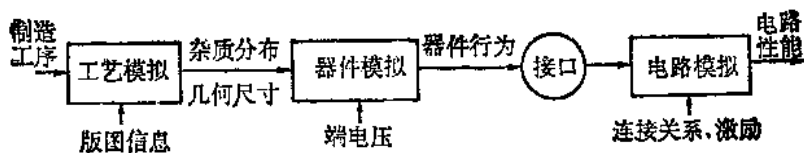


图 9-1 工艺、器件、电路模拟的结合

如果反过来，人们能从器件的性能要求出发，设法找到一套合理的工艺步骤和工艺参数，那对器件开发和工艺开发的意义就更大。而这正是我们所要追求的目标。

目前比较著名的工艺模拟程序有 SUPREM、BICEPS、SAMPLE 和 ICECREAM 等。

SUPREM 是美国斯坦福大学所开发的。SUPREM I 发表于 1977 年，1978 年更新为 SUPREM II，I 和 II 版本只能求出硅和二氧化硅中的一维杂质分布。1983 年对原版本作了较大的修改，得到了 SUPREM III，除了硅和二氧化硅外还可模拟多晶硅和氮化硅中的杂质分布。但 SUPREM III 也只限于一维的分析。

随着器件尺寸的日益缩小，各工艺之间的影响既有垂直的，也有水平的。只限于一维的讨论已不能适应需要。1982 年发表的 BICEPS 工艺模拟程序既可以模拟一维的，也能

够处理二维的。它能模拟扩散、离子注入、沉淀积、氧化、外延和腐蚀等工艺。SAMPLE 是专用于模拟光学及电子束光刻工艺的模拟程序。

§ 9.2 工艺模拟的求解方法

工艺模拟所要模拟的工艺过程有氧化、扩散、离子注入、外延生长、高温或低温淀积、光刻、腐蚀等。这些工序大致可分成两类：一类是经过高温处理的，如氧化、扩散、外延、高温淀积等。另一类是非高温处理，如离子注入、低温淀积、光刻、腐蚀等，前一类工序会引起杂质的扩散和再分布，其中氧化工序还要考虑氧化层随时间的增长，外延工序还要计算硅外延层的增厚，即都要考虑界面的移动。后一类工序多数不必考虑杂质浓度的再分布，如光刻、腐蚀等工序，但离子注入会引起新的杂质分布，但它与杂质扩散有所不同。所以工艺模拟程序中有不同的子程序来处理不同的工艺过程。但由于工艺模拟中主要关心的是不同材料层中的杂质分布情况，因此主要的计算问题是求解杂质的连续方程。

一、杂质流的连续性方程

连续性方程有以下形式

$$\frac{\partial}{\partial t} \int_{V(t)} C dV = \int_{V(t)} (g - l) dV - \oint_{S(t)} \mathbf{F} \cdot \mathbf{n} dS \quad (9-1)$$

式中， C 为杂质的化学浓度； $S(t)$ 为封闭表面的表面积； $V(t)$ 为封闭表面所包围的体积； $\mathbf{F}$ 为杂质流向量； $\mathbf{n}$ 为 S 表面的外法向单位矢量； g 为单位体积杂质产生率； l 为单位体积杂质损失率。

在一维情况下，上式可表示为

$$\frac{d}{dt} \int_{x_1}^{x_2} C(x) dx = \int_{x_1}^{x_2} (g - l) dx - [F(x_2) - F(x_1)] \quad (9-2)$$

其中 $F(x_1)$ 与 $F(x_2)$ 分别为流入 x_1 截面与流出 x_2 截面的杂质流，示于图 9-2。流向截面内部的杂质流为正，流出截面的为负。如忽略产生-损失项，(9-2) 式可简化为

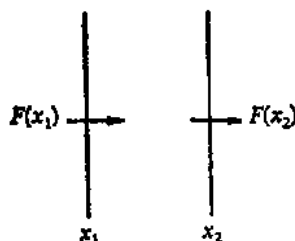


图 9-2 杂质流的流向

$$\frac{d}{dt} Q(x_1, x_2) = -[F(x_2) - F(x_1)] \quad (9-3)$$

式中 $Q(x_1, x_2) = \int_{x_1}^{x_2} C(x) dx$ 为杂质质量。(9-3) 式也可写成以下形式，即

$$\frac{dC}{dt} = -\frac{dF}{dx}$$

在固态中，杂质流由扩散流和漂移流两项所组成，即

$$F = -D \left(\frac{dC}{dx} \right) \pm D \left(\frac{q}{kT} E \right) C. \quad (9-4)$$

式中第二项对施主杂质为正，对受主杂质为负。 D 为扩散系数， E 为电场强度， C 为电

学上有效的杂质浓度。由于扩散系数不是一个常数，而与杂质浓度、杂质带电状态等有关，因此在固态中的杂质流连续方程是一个非线性微分方程。

二、空间离散化和时间离散化

为了进行数值求解，我们同样要对连续方程进行空间离散化。为此要把求解空间沿 X 方向划分成若干网格。每个网格中包含一个格点，它处于网格的中点，也就是网格的边界处于两相邻格点的中间。整个结构的两端是半个网格，因而结构的两个边界正好落在格点上，这示于图 9-3。

如果认为网格中的杂质浓度是均匀的，并定义在格点上，则求解杂质分布的问题就成为计算网格中点即格点上的杂质浓度 $C(x_i)$ 。

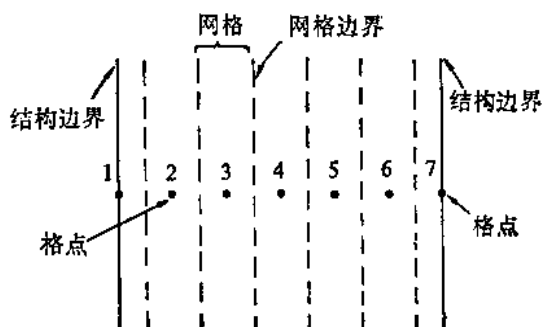


图 9-3 网格图

一维空间离散化的情况示于图 9-4。现选取一个包含第 i 个格点的子区间，并对该子区间应用连续方程就得到

$$\frac{d}{dt}Q_i = -\left[F_D\left(x_{i+\frac{1}{2}}\right) - F_D\left(x_{i-\frac{1}{2}}\right)\right] \quad (9-5)$$

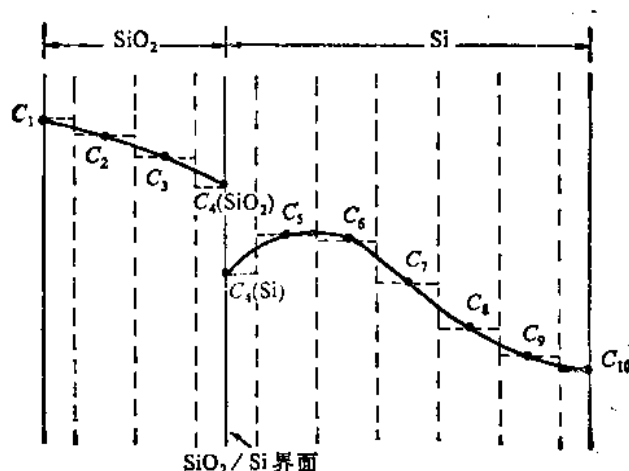


图 9-4 一维空间离散化

式中， $Q_i = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} C(x) dx$ ， F_D 为流过网格边界的杂质扩散流。

在求解杂质浓度随时间的变化时，令

$$H_i(t) = \frac{d}{dt}Q_i(t) \quad (9-6)$$

设时间 t_0 的杂质浓度已知，则下一时间 t_1 的杂质浓度可用下列方程求出

$$\int_{t_0}^{t_1} H_i(t) dt = Q_i(t_1) - Q_i(t_0) \quad (9-7)$$

现采用隐式后向欧拉法对时间离散化, 即假设在 $t_1 - t_0$ 时间间隔内 $H_i(t) = H_i(t_1)$, 这样 (9-7) 式就改写为

$$H_i(t_1) = \frac{Q_i(t_1) - Q_i(t_0)}{t_1 - t_0} \quad (9-8)$$

在经过以上离散化后 (空间上采用中点积分, 时间上采用后向欧拉法), 微分方程变为代数方程。

非线性代数方程则采用通常的牛顿-拉夫森方法加以线性化, 最后用高斯消去法求解线性代数方程。

三、固定边界

对于起始边界, 第 1 个网格的连续方程有

$$\frac{d}{dt} Q_1 = - \left[F_D \left(x_{1+\frac{1}{2}} \right) - F_s(0) \right] \quad (9-9)$$

式中, $F_s(0)$ 是经过起始边界处的杂质流。

对于终端边界, 第 n 个网格的连续方程有

$$\frac{d}{dt} Q_n = - \left[F_s(x_n) - F_D \left(x_{n-\frac{1}{2}} \right) \right] \quad (9-10)$$

这里, $F_s(x_n)$ 为经过终端边界处的杂质流。

如果结构中存在一界面 (例如 SiO_2/Si 界面), 该界面处于第 i 个格点处。由于杂质分凝的原因, 在界面上的杂质浓度不再连续。格点 i 上有两个不同的浓度, 见图 9-4。这时连续方程可分别写成

$$\frac{d}{dt} Q_i(\text{SiO}_2) = - \left[F_s(x_i) - F_D \left(x_{i-\frac{1}{2}} \right) \right] \quad (9-11)$$

$$\frac{d}{dt} Q_i(\text{Si}) = - \left[F_D \left(x_{i+\frac{1}{2}} \right) - F_s(x_i) \right] \quad (9-12)$$

其中 $F_s(x_i)$ 为流经 x_i 界面的杂质流, 它可表示为

$$F_s = h(C_{\text{SiO}_2}/m_{eq} - C_{\text{Si}}) \quad (9-13)$$

式中, h 为表面质量转换系数; m_{eq} 为平衡时的分凝系数。

四、移动界面

当界面发生移动时, 在总的杂质流中除了杂质扩散流及分凝效应产生的杂质流外, 还应增加一项界面移动造成的感生杂质流, 即

$$F = F_D + F_s + F_b \quad (9-14)$$

而感生杂质流的大小取决于

$$F_b = -C(t)v \quad (9-15)$$

式中, v 为界面移动速率。

现以氧化过程为例说明界面移动时算法的实现。

图 9-5 为界面移动前后, 即 t_0 和 $t_0 + \Delta t$ 时刻网格的示意图。边界 x_1 是氧化气氛 (气相) 与 SiO_2 之间的界面。边界 x_2 是 SiO_2 与 Si 之间的界面。M 是 SiO_2 与 Si 交界处的格点号。 $x_{M-\frac{1}{2}}$ 是 M 网格的左边界。移动后仍应为 $M-1$ 到 M 的中点位置。

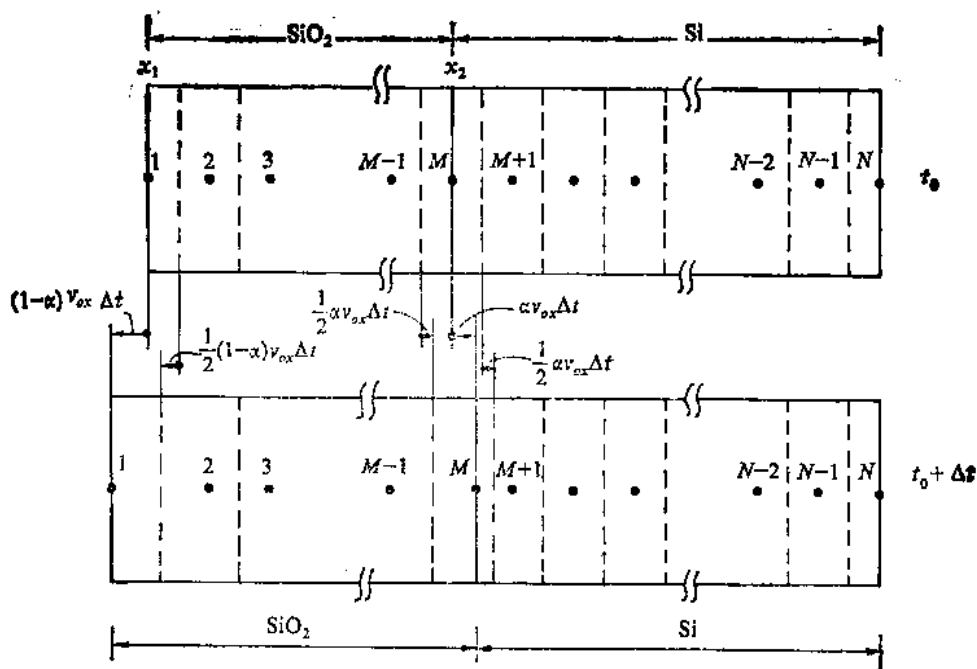


图 9-5 SiO₂-Si 界面移动前后网格变化

氧化时, x_1 和 x_2 都在移动, 并有

$$\frac{dx_1}{dt} = -(1-a)v_{ox} \quad (9-16)$$

$$\frac{dx_2}{dt} = av_{ox} \quad (9-17)$$

式中, a 为氧化掉的硅层与转换成为二氧化硅层的厚度之比 (≈ 0.44); v_{ox} 为氧化层的生长速率。因此, SiO₂ 与 Si 界面的移动速率, 即 M 格点的移动速率为 av_{ox} , 而 M 格点左边界($x_{M-\frac{1}{2}}$ 处) 的移动速率为 $\frac{1}{2}av_{ox}$ 。

在界面移动时, 会影响界面左右邻近的几个网格。现以界面左侧 SiO₂ 一边的 M 网格, 列出其连续方程

$$\begin{aligned} & Q(x_M, t_0 + \Delta t) - Q(x_M, t_0) \\ &= \int_{x_{M-\frac{1}{2}}, t_0 + \Delta t}^{x_M, t_0 + \Delta t} C(x, t_0 + \Delta t) dx - \int_{x_{M-\frac{1}{2}}, t_0}^{x_M, t_0} C(x, t_0) dx \\ &= \int_{t_0}^{t_0 + \Delta t} \left[F_D\left(x_{M-\frac{1}{2}}, t\right) + F_S\left(x_{M-\frac{1}{2}}, t\right) - F_S(x_M, t) - F_B(x_M, t) \right] dt \\ &= \int_{t_0}^{t_0 + \Delta t} \left[F_D\left(x_{M-\frac{1}{2}}, t\right) - \frac{1}{2}av_{ox}C\left(x_{M-\frac{1}{2}}, t\right) - F_S(x_M, t) + av_{ox}C(x_M, t) \right] dt \end{aligned} \quad (9-18)$$

再以外延生长为例说明界面移动时算法的实现。

图 9-6 表示外延生长时被模拟空间的离散化情况。图中的实线表示气相与固相之间的界面，而虚线表示网格的边界。在每个网格中的杂质浓度（即 C_i, C_{i+1}, C_{i+2} 等）被认为是均匀的。现已知 $t=t_0$ 时固相硅中杂质的分布，算法的实现分成两步。

第一步：在 t_0 时间，原界面不动，但增加一新格点 x_{i-1} ，新格点处的 C_{i-1} 由进入 $i-1$ 网格的杂质流大小决定（如图 9-6(b) 的箭头所示），这时不考虑固相硅内的杂质再分布。

第二步：界面移动到新的位置。并以图 9-6(b) 中的杂质分布为基础计算 Δt 时间内的杂质再分布，如图 9-6(c) 所示。图 9-6(c) 中的各个箭头表示穿过各网格边界的扩散流。应注意，这时在气-固相界面处认为没有杂质流流入。

图 9-6(d) 是在 Δt 时间后由于扩散流的作用而得到的新的杂质分布。它将是计算下一时间步长时的初始杂质分布（初始条件）。

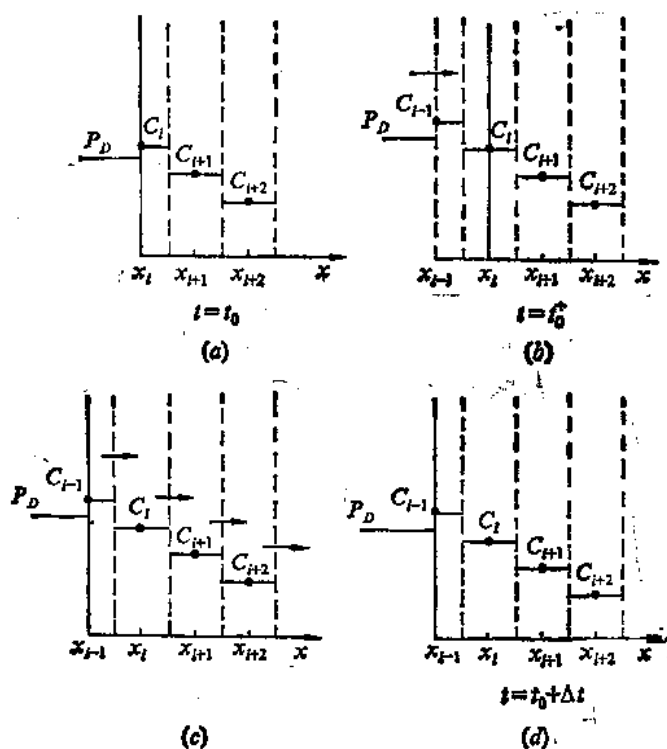


图 9-6 外延生长时数值求解方法

§ 9.3 SUPREM III 工艺模拟程序及其应用

一、模拟的内容

现有 SUPREM III 8520 版本可以模拟以下各工艺过程：(1) 氧化和非氧化气氛下的高温扩散；(2) 硅和氮化硅的氧化；(3) 离子注入；(4) 硅的外延生长；(5) 低温淀积；(6) 各种材料的腐蚀。

被模拟的结构可以多至十层，每一层可以由十种材料所组成，而同一种材料可以允许在几层中。

目前程序可模拟的材料有单晶硅、多晶硅、二氧化硅、氮化硅、铝这五种。可以允许的掺杂物质是硼、磷、砷、锑。但掺杂物质也可扩展到十种。

SUPREM II 可以求得经过各种工艺过程后的参数：(1) 各种材料层的厚度；(2) 杂质在不同材料层中的分布；(3) 某些电学参数，如薄层电阻、电子和空穴浓度、电导率及开启电压等。并可针对某一层所规定的偏压，求出这些参数值随外加电压的变化。

二、网格结构

在 SUPREM II 中，把整个结构划分成若干个网格。网格划分越细，求解精度越高，但计算时间则越长。现有版本允许最多有 500 个格点。

每个网格中包含一个格点，它处于网格的中点，也就是网格的边界正处于相邻格点的中间。整个结构的两端是半个栅格，因而结构的两个边界正好落在格点上。

在 SUPREM II 中，相邻格点的距离可以由用户加以调整，以满足杂质分布陡变处的计算精度要求，也就是在陡变处设法缩小格点间距，增加网格数。格点的距离可由 INITIALIZE 或 DEPOSITION 语句加以调整，也可以在模拟过程中通过 GRID 语句加以调整。

网格可以是均匀分布，也可设置成呈抛物线上升或下降式分布。这可通过四个参数加以控制。

- (1) 层厚 THICKNESS (单位为 μm)；
- (2) 标定的格点间距 DX (单位为 μm)；
- (3) 采用标定格点间距的那点与层表面的距离 XDX (单位为 μm)；
- (4) 层内网格数 SPACES。

如要求网格为均匀分布，用户只需规定材料厚度，程序就认为 XDX 处于零点并取隐含的 DX 值。这时

$$SPACES = \frac{THICKNESS}{DX} \quad (9-19)$$

如用户认为 DX 不合适，则可以设置新的 DX 值，程序将采用定义值代替隐含值。

如用户希望在某一范围内有较多的格点，但又要求总格点数仍在 500 个以内，这时可采用不均匀间距。如果要求层内的网格数比均匀分布时的网格数小，并在规定的 XDX 处采用标定的 DX 值，则在 XDX 两边的格点间距将抛物线性地增加。反之，如要求层内的网格数比均匀分布时的多，则在 XDX 处两边的格点间距将抛物线性地下降。这三种情况示于图 9-7。

三、输入文件格式

SUPREM II 中规定了一套输入语句的格

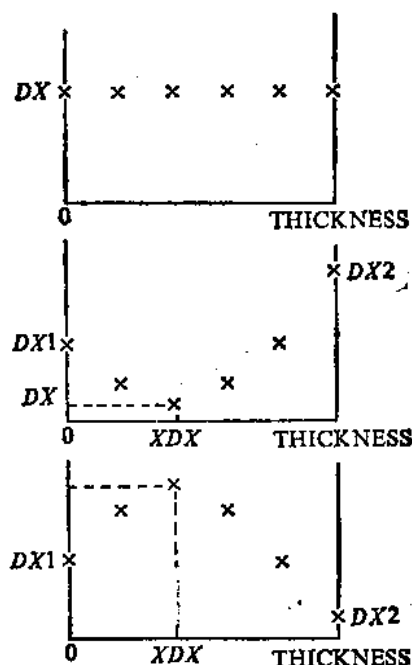


图 9-7 均匀和不均匀网格分布

式。并规定有 36 种语句。

输入文件的第一句语句必须是 TITLE。它输入一串字符对输入文件作一说明。该字符串将在输出设备上打印出来。

接着由 INITIALIZE 语句建立一初始结构,并输入材料和杂质的有关参数,也可输入一含有初始结构信息的文件名。此语句必须在所有其它语句(除 TITLE 和 COMMENT 语句以外)之前。

在确定了初始结构和各参数值后,就可通过 DIFFUSION、IMPLANT、EPITAXY、DEPOSITION、ETCH 等语句描述各工艺过程和工艺条件。

对 DIFFUSION 语句,如只规定时间和温度,就表示只模拟非氧化气氛下的扩散推进。如只规定时间和温度,以及氧化气氛而不说明杂质时,则表示只模拟氧化过程。

如要改变掺杂物质的物理参数和模型参数,可通过 IMPURITY 语句或 BORON, PHOSPHORUS, ARSENIC, ANTIMONY 语句加以改变。

有关材料的各系数可通过 MATERIAL 语句或通过 SILICON, OXIDE, POLYSILICON, NITRIDE, ALUMINUM 语句加以改变。

有关材料与杂质之间相互关系的一些系数,可通过 SEGREGATION, VOL. RATIO, MOBILITY 语句加以控制。

氧化速率和气氛可通过 DRYO2, WETO2 和 NITROGEN 语句加以确定。

如果用户改变了任一系数,可以通过 SAVE 语句把修改的一组系数保存在一个用户命名的系数文件,而后可用 INITIALIZE 或 LOADFILE 语句将其读入。

SUPREM II 中还规定 COMMENT 语句用于注释,该语句的继续行符号为 \$。

模拟结果的输出语句有 PRINT 和 PLOT 语句。为了保存当前结构及模型系数,可利用 SAVE 语句。

此外,还有 ELECTRICAL 语句对当前结构的泊松方程求数值解。BIAS 语句是求解泊松方程时设置导电层和半导体中的偏置电压。END.ELEC 语句是一系列 BIAS 语句的中止语句。

输入文件最后由 STOP 语句结束。

§ 9.4 SUPREM II 工艺模拟程序应用举例

现有一器件采用 NMOS 硅栅工艺制造而得。其横截面如图 9-8 所示。它采用的工艺步骤如下:

- (1) 衬底原始材料为高阻<100>P 型硅,衬底浓度为 $1 \times 10^{15} \text{cm}^{-3}$ 。
- (2) 在 1000°C 干氧气氛中氧化 40 min, 生成 40nm 的初始氧化层。
- (3) 在二氧化硅上淀积 80nm 的氮化硅层。
- (4) 把有源区以外的氮化硅去除。
- (5) 在场区表面注硼,以增加 P 型杂质。能量为 150keV, 剂量为 $1 \times 10^{13} \text{cm}^{-2}$ 。
- (6) 在 1000°C 湿氧下氧化三小时, 生长场氧化层。
- (7) 去除氧化硅、氮化硅, 腐蚀到硅表面。
- (8) 注硼以调整开启电压。注入能量为 50keV, 剂量为 $4 \times 10^{11} \text{cm}^{-2}$ 。

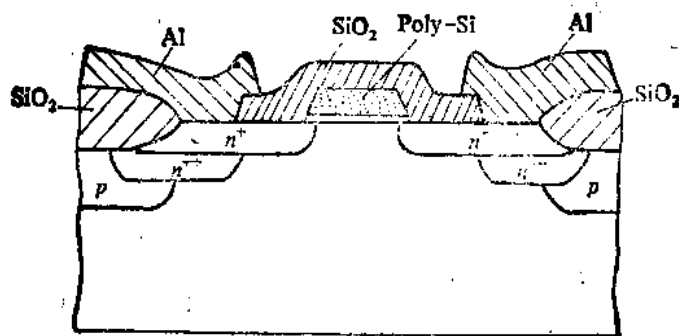


图 9-8 器件的横截面

(9) 在 1050℃干氧气氛中（加氯气，其比例为 3%）氧化 30 min，生长 40nm 的栅氧化层。

(10) 在 600℃下淀积 0.5μm 的多晶硅，

(11) 用 POCl₃ 在预淀积炉中掺磷。温度 1000℃，时间 25min。

(12) 把栅区和连线以外的多晶硅腐蚀掉。

(13) 把源漏区的氧化层腐蚀掉。

(14) 离子注入砷，形成源区和漏区。注入能量 150keV，剂量 $5 \times 10^{15} \text{cm}^{-2}$ 。

(15) 在 1000℃干氧中推进源漏区砷杂质 30 分钟。

(16) 光刻源、漏、栅区的接触孔。

(17) 利用 CVD 工艺在硅表面淀积掺磷的二氧化硅，厚度 0.75μm，磷浓度为 $1 \times 10^{21} \text{cm}^{-3}$ 。

(18) 在 1000℃下回流磷硅玻璃，时间 30min。

(19) 再次光刻接触孔。

(20) 蒸发铝，厚度 1.2μm。

(21) 光刻铝层形成内互连。

如要求模拟下述三个垂直截面处杂质分布：(1) 通过栅区中心的截面即模拟沟道区；(2) 通过源区或漏区截面；(3) 通过场区截面。这时可通过六个输入文件分别加以模拟。

第一个文件 描述工艺开始直到源区和漏区形成以前，即不包括砷注入及以后的工艺。这是模拟沟道区的形成。

第二个文件 利用第一个文件的结果作为起始并完成所有工艺（不包括腐蚀源漏区上的多晶硅和二氧化硅）。这是模拟栅区的形成。

第三个文件 利用第一个文件的结果作为起始并完成所有工艺。这是模拟源区或漏区的情况。

第四个文件 完成栅区 V_T 的计算。

第五个文件 类似于第一个文件，所不同的只是模拟场区的工艺，得到多晶硅淀积和掺杂后的场区情况。

第六个文件 在完成所有工艺步骤后的场区情况。

表 9-1 列出了第一个文件的输入格式。其模拟结果示于图 9-9。该图表示沟道注入、

栅氧生长、多晶硅淀积和掺杂后，在多晶硅、二氧化硅和硅中的磷分布和硼分布。图中浓度 C 单位为 cm^{-3} 。

表 9-1 输入文件

Title	SUPREM-III Example 1. NMOS Silicon Gate
Comment	Active device region initial processing.
\$	File S3EX1A
Comment	Initialize silicon substrate.
Initialize	(100) Silicon, Boron Concentration=1e15
+	Thickness=1.5 dx=.005 xdx=0.02 Spaces=150
Comment	Grow pad oxide,400A.
Diffusion	Temperature=1000 Time=40 Dryo2
Comment	Deposit 800A of CVD Nitride.
Deposit	Nitride Thickness=.0800 Spaces=15
Comment	Grow field oxide.
Diffusion	Temperature=1000 Time=180 Weto2
Print	Layer
Comment	Etch to silicon surface.
Etch	Oxide all
Etch	Nitride all
Etch	Oxide all
Comment	Implant boron to shift the threshold voltage.
Implant	Boron Dose=4e11 Energy=50
Comment	Grow gate oxide
Diffusion	Temperature=1050 Time=30 Dryo2 HCL%=3
Comment	Deposit polysilicon
Deposit	Polysilicon Thickness=0.5 Temperature=600
Comment	Heavily dope the polysilicon using POCl3
Diffusion	Temperature=1000 Time=25 dTmin=.3
+	Phosphorus Solidsolubility
Print	Layer
Plot	Chemical Boron
+	Xmax=1.5 Clear ^Axis Linetype=2
Plot	Chemical Phosphorus
+	Xmax=1.5 ^Clear ^Axis Linetype=3
Plot	Chemical Net
+	Xmax=1.5 ^Clear Axis

```

Comment      Save the structure at this point.
*            The simulation runs are split
*            for the gate and source/drain regions.
Save          Structure File=S3E1AS
Stop          End of SUPREM-III Example 1.

```

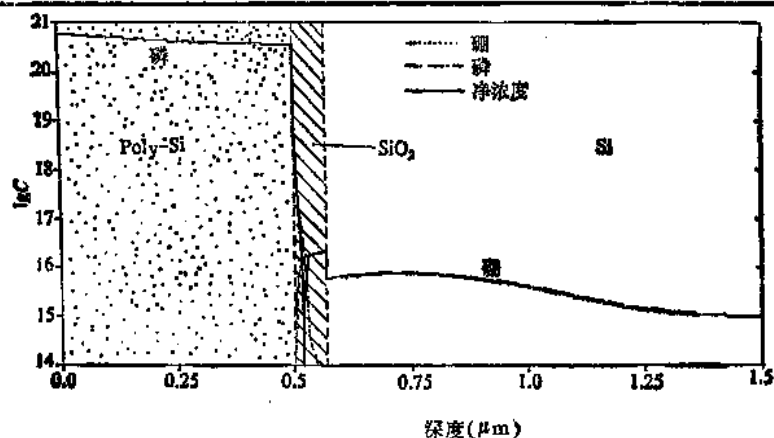


图 9-9 工艺模拟结果

§ 9.5 SUPREM III 中的工艺模型

一、热氧化模型

二氧化硅在硅上的生长速率可由下列生长规律进行描述

$$\frac{(x-x_0)^2}{B} + \frac{(x-x_0)}{B/A} = t \quad (9-20)$$

式中, x 为氧化层厚度; x_0 为 $t=0$ 时的氧化层厚度; t 为时间; B 和 B/A 为抛物线和线性速率常数。这两个速率常数分别反映两种氧化过程: (1) 氧化剂在二氧化硅层中的扩散过程 (2) 氧化剂在硅/二氧化硅界面处的反应过程。

B 正比于二氧化硅中氧化剂的溶解度, 且正比于二氧化硅中氧化剂的扩散系数。

B/A 也正比于二氧化硅中氧化剂的溶解度, 且正比于硅/二氧化硅界面的反应速率常数。

近年来, 工艺上更多采用低温氧化和薄氧化层, 因而 B/A 越来越成为氧化速率限制的主要因素。很多重要的工艺现象如氧化增强扩散都反应在 B/A 这个常数中。

对水汽和干氧氧化, 有

$$B = B^{(i)} \left[\frac{1}{2} (P_i + P_{i-1}) \right] [1 + \delta C_i^*] \quad (9-21)$$

式中, $B^{(0)}$ 为本征抛物线速率常数, 第二项与氧化剂的压强有关, 第三项与衬底掺杂有关, 第四项与 HCl 气氛有关。

对水汽氧化, 有

$$B/A = (B/A)^{(0)} \left[\frac{1}{2} (P_i + P_{i-1}) \right] [1 + r(C_v - 1)] \eta \alpha \quad (9-22)$$

式中, $(B/A)^{(0)}$ 为本征的线性速率常数, 第二项与氧化剂的压强有关, 第三项与衬底掺杂有关, 第四项与 HCl 气氛有关, 第五项与晶向有关。

对于氧化, 除修改压强这一项外, 还增加一项与薄氧有关的快速初始氧化项, 因而

$$B/A = (B/A)^{(0)} \left[\frac{1}{2} (P_i^{1/2} + P_{i-1}^{1/2}) \right] [1 + r(C_v - 1)] \eta \alpha [1 + K e^{-x/L}] \quad (9-23)$$

式中, 第六项与增强氧化现象有关。在公式 (9-21)、(9-22)、(9-23) 中, P_i 为 t_i 时间的压强, P_{i-1} 为 t_{i-1} 时间的压强。

1. 本征 $B^{(0)}$ 和 $(B/A)^{(0)}$ 值

对于低掺杂 <111> 晶向的单晶硅, 在干氧化且氧化物压强为一个大气压时

$$B^{(0)} = C_1 e^{-E_1/(kT)} \quad (9-24)$$

式中, $C_1 = 7.72 \times 10^2 \mu\text{m}^2/\text{h}$; $E_1 = 1.23\text{eV}$ 。

而

$$(B/A)^{(0)} = C_2 e^{-E_2/(kT)} \quad (9-25)$$

式中, $C_2 = 6.23 \times 10^5 \mu\text{m}/\text{h}$; $E_2 = 20\text{eV}$

在水汽氧化时, 不同温度范围有不同的激活能。这时的速率常数公式仍为

$$B^{(0)} = C_1 e^{-E_1/(kT)}$$

但 C_1 和 E_1 则不同。

当 $T > 950^\circ\text{C}$ 时 $C_1 = 4.2 \times 10^2 \mu\text{m}^2/\text{h}$;

$$E_1 = 0.78\text{eV}。$$

当 $T < 950^\circ\text{C}$ 时 $C_1 = 1.7 \times 10^4 \mu\text{m}^2/\text{h}$;

$$E_1 = 1.17\text{eV}。$$

对 $(B/A)^{(0)}$ 仍有 (9-25) 式同样形式, 但 C_2 和 E_2 不同。

当 $T > 900^\circ\text{C}$ 时 $C_2 = 1.77 \times 10^5 \mu\text{m}/\text{h}$;

$$E_2 = 2.05\text{eV}。$$

当 $T < 900^\circ\text{C}$ 时 $C_2 = 2.07 \times 10^6 \mu\text{m}/\text{h}$;

$$E_2 = 1.6\text{eV}。$$

对于一个大气压下的“湿氧”氧化 (通过 95°C 的水起泡器), 已证明它对应于一个水汽氧化压力 $P_{\text{H}_2\text{O}} = 0.92$ 大气压压强。

高温时 E_1 的下降被认为是二氧化硅的粘滞流动而减弱了二氧化硅中的应力, 因而使扩散更容易。低温时 E_2 的下降被认为是二氧化硅中的应力会有利于键的断裂, 所以使界面反应过程加速。

没有发现 B 与晶向有明显的关系。但对于 B/A 则观察到在水汽和干氧化时, <111> 晶向的 B/A 值是 <100> 晶向值的 1.7 倍。其它晶向的 B/A 值在 <111> 和 <100> 值之间。

近年来发现在二氧化硅粘滞流动温度 (约 950℃) 以上和以下时, B 和 B/A 与晶向的关系是不同的。在该温度以下, $B^{(d)}$ 与晶向有关, 但 SUPREM III 中没有模拟这一点。

2. 与压强的关系

从公式 (9.21) 和 (9.22) 可知, 对水汽, B 和 B/A 值与水汽的压强成线性关系。对于氧, B 仍为线性关系。但公式 (9-23) 表明, B/A 则是 P 的 n 次方关系。 n 值为 0.7~0.8, 在 SUPREM III 中取 $n=0.75$ 。

3. 与衬底掺杂的关系

对于掺磷情况, 发现 B 值随掺杂浓度的增加 ($>10^{19}\text{cm}^{-3}$) 而有所增加, SUPREM III 中采用下列经验公式

$$B=B^{(d)}[1+\delta C_r^*]$$

$$\text{式中} \quad \delta=9.63 \times 10^{-18} \exp\left(\frac{2.83\text{eV}}{kT}\right) \quad (9-26)$$

$$q=1.28 \exp\left(\frac{-0.176\text{eV}}{kT}\right) \quad (9-27)$$

C_r 是硅/二氧化硅界面处衬底一边总的杂质化学浓度。该公式也可用于其它 N 型杂质如砷和锑。但对于 P 型杂质例如硼, B 与掺杂的关系还不清楚。

在衬底高掺杂时, 氧化速率有明显的增加。这是因为衬底高掺杂时增加了衬底总的空位浓度, 而空位参加硅/二氧化硅界面处的氧化反应, 因此线性速率常数 B/A 值与衬底增加的空位浓度直接有关。SUPREM III 中采用下列公式

$$B/A=(B/A)^{(d)}[1+r(C_v-1)]$$

$$\text{式中} \quad r=2.62 \times 10^3 \exp\left(\frac{-1.10\text{eV}}{kT}\right) \quad (9-28)$$

$$C_v = \frac{1 + C_v^+ \left(\frac{n_i}{n}\right) + C_v^- \left(\frac{n}{n_i}\right) + C_v^{2-} \left(\frac{n}{n_i}\right)^2}{1 + C_v^+ + C_v^- + C_v^{2-}} \quad (9-29)$$

C_v 为归一化后的总的空位浓度。公式 (9-29) 中的 C_v^+ 、 C_v^- 、 C_v^{2-} 分别是用中性本征空位浓度归一化后的单正电、单负电、双负电本征空位浓度, 并有以下表达式

$$C_v^+ = \exp[(E^+ - E_i)/(kT)] \quad E^+ = 0.35\text{eV} \quad (9-30)$$

$$C_v^- = \exp[(E_i - E^-)/(kT)] \quad E^- = E_g - 0.57\text{eV} \quad (9-31)$$

$$C_v^{2-} = \exp[(2E_i - E^- - E^{2-})/(kT)] \quad E^{2-} = E_g - 0.12\text{eV} \quad (9-32)$$

4. 与 HCl 气氛的关系

由于加入氯气, 氧化速率有明显提高。目前 SUPREM III 中采用经验公式。

对于 HCl/O₂ 氧化条件, η 和 ε 随温度和 HCl 比例的变化关系见表 9-2 和表 9-3。在表中无法找到的温度点与 HCl 比例, 可采用线性插入法求出。

对于 HCl/H₂O 氧化条件, 发现随 HCl 比例的增加, 氧化速率有所下降, 现用下式进行计算:

$$\eta=1-(\theta/100) \quad (9-33)$$

$$\varepsilon=1-(\theta/100) \quad (9-34)$$

5. 薄氧化层的干氧化

表 9-2

θ \ T	900℃	1000℃	1100℃
η			
0(%HCl)	1.0	1.0	1.0
1(%HCl)	1.750	1.250	1.621
3(%HCl)	1.750	1.486	2.207
5(%HCl)	1.750	1.486	2.207
7(%HCl)	1.750	1.486	2.207
10(%HCl)	1.750	1.486	2.207

表 9-3

θ \ T	900℃	1000℃	1100℃
s			
0(%HCl)	1.0	1.0	1.0
1(%HCl)	1.083	1.658	1.355
3(%HCl)	1.250	1.840	1.490
5(%HCl)	1.444	2.075	1.641
7(%HCl)	1.639	2.332	1.816
10(%HCl)	2.028	2.769	2.102

在干氧中生长很薄的氧化层 ($<50\text{nm}$) 时与生长较厚的氧化层的氧化速率有很大的不同。这称为增强氧化现象。

现根据实验数据得到以下经验公式来描述初始阶段氧化速率的提高

$$\frac{dx}{dt} = \frac{B}{2x_i + A} + K' e^{-x_i/L} \quad (9-35)$$

式中, L 与温度无关, 且等于 7nm ; K' 是温度的函数。

二、杂质扩散模型

半导体中的杂质一般通过空位或间隙这两种机构进行扩散。

在非氧化条件下, 空位扩散是主要的, 因而 SUPREM II 中选择了空位扩散这一模型。

硅中的空位除了中性空位外, 还有其它类型的空位。总的扩散系数可以认为是各类空位的扩散系数之和。

$$\begin{aligned} D &= D^0 + D^- + D^{2-} + D^+ \\ &= D_i^0[V^0] + D_i^-[V^-] + D_i^{2-}[V^{2-}] + D_i^+[V^+] \end{aligned} \quad (9-36)$$

式中, D_i^0 为中性空位本征扩散系数; D_i^- 为单负电空位本征扩散系数; D_i^{2-} 为双负电空位本征扩散系数; D_i^+ 为单正电空位本征扩散系数; V^0 、 V^- 、 V^{2-} 、 V^+ 分别是用各相应带电状态的本征空位浓度归一化后的中性、单负电、双负电、和单正电空位浓度, 即 $V^0=1$,

$V^- = \frac{n}{n_i}$, $V^{2-} = \left(\frac{n}{n_i}\right)^2$; $V^+ = \frac{p}{n}$ 或 $\frac{p}{n_i}$ 。这里, n_i 为该工艺温度下的本征电子浓度。

因而公式 (9-36) 可改写为

$$D = D_i^0 + D_i^- \frac{n}{n_i} + D_i^{2-} \left(\frac{n}{n_i}\right)^2 + D_i^+ \left(\frac{p}{n_i}\right) \quad (9-37)$$

但通常公式 (9-37) 中不是所有项都起同样的作用。对某种杂质来讲, 只有某几项起主要作用, 其它项可以忽略。

在氧化条件下, 通常扩散系数要加大, 这是因为在氧化情况下, 硅/二氧化硅界面会把很多间隙原子注入硅衬底, 结果使扩散系数中间隙扩散的成份增加。

1. 硼杂质

硼是受主, 在硅中带负电。它主要通过 V^0 和 V^+ 空位进行扩散, 因而

$$D_B = D_i^0 + D_i^+ \left(\frac{p}{n_i} \right) \\ = \left[0.037 + 0.72 \left(\frac{p}{n_i} \right) \right] \exp \left(\frac{-3.46 \text{ eV}}{kT} \right) (\text{cm}^2/\text{s}) \quad (9-38)$$

可以看出,除了重掺杂的N型硅外, D^+ 起主要作用。

2. 砷杂质

砷是施主,在硅中带正电。它主要通过 V^0 和 V^- 空位进行扩散,因而

$$D_{As} = D_i^0 + D_i^- \left(\frac{n}{n_i} \right) \\ = 0.066 \exp \left(\frac{-3.44 \text{ eV}}{kT} \right) + 12.0 \left(\frac{n}{n_i} \right) \exp \left(\frac{-4.05 \text{ eV}}{kT} \right) (\text{cm}^2/\text{s}) \quad (9-39)$$

在实际的扩散温度范围内, D_i^0 与 D_i^- 的大小相仿,因而其相应的扩散系数的重要性取决于掺杂浓度。

3. 铟杂质

铟是施主,在硅中带正电。它主要通过 V^0 和 V^- 空位进行扩散,因而

$$D_{In} = D_i^0 + D_i^- \left(\frac{n}{n_i} \right) \\ = 0.214 \exp \left(\frac{-3.65 \text{ eV}}{kT} \right) + 15.0 \left(\frac{n}{n_i} \right) \exp \left(\frac{-4.08 \text{ eV}}{kT} \right) (\text{cm}^2/\text{s}) \quad (9-40)$$

在扩散温度内, D_i^- 大于 D_i^0 。N型硅中 D_i^- 起主要作用。

4. 磷杂质

磷杂质有复杂的扩散行为。其典型的杂质分布见图9-10。特别是在双极型器件中,当发射区高掺杂时会出现所谓反常现象,如“尾部”区加强扩散,基区宽度变宽。

对于浓度梯度变化不急剧的情况,磷的扩散主要通过 V^0 、 V^- 、 V^{2-} 空位进行,因此

$$D_P = D_i^0 + D_i^- \left(\frac{n}{n_i} \right) + D_i^{2-} \left(\frac{n}{n_i} \right)^2 \\ = 3.85 \exp \left(\frac{-3.66 \text{ eV}}{kT} \right) \\ + 4.44 \left(\frac{n}{n_i} \right) \exp \left(\frac{-4.00 \text{ eV}}{kT} \right) \\ + 41.2 \left(\frac{n}{n_i} \right)^2 \exp \left(\frac{-4.37 \text{ eV}}{kT} \right) \\ (\text{cm}^2/\text{s}) \quad (9-41)$$

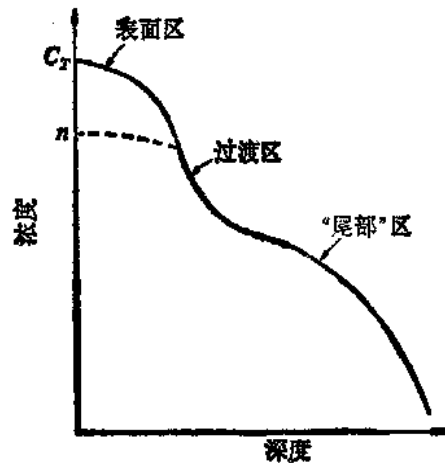


图 9-10 磷扩散杂质分布

在低掺杂时, D^0 起主要作用。对于重掺杂,由于 $(n/n_i)^2$ 的关系, D^{2-} 将起主要作用。

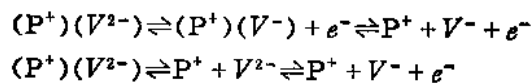
(一) 表面区

在高浓度的表面处, 可以认为 D^{2-} 起主要作用。在浓度达到 $3 \sim 4 \times 10^{20} \text{cm}^{-3}$ 以前, 扩散系数将随表面浓度的增加而提高, 但在这以后, 进一步提高浓度, 扩散系数反而下降, 这是由于失配的原因, 当磷的溶解度很高时会引起晶格内应力的增加, 结果使禁带宽度变窄, 使 n_i 上升。在离子注入时也发现高剂量时扩散系数会下降, 这可能是离子注入引起晶格失序, 同样会引起禁带宽度变窄。两者相加得

$$\Delta E_g = \Delta E_{g\text{失配}} + \Delta E_{g\text{失序}} \quad (9-42)$$

(2) “尾部”区

实验发现所有在磷区下面的其它杂质的扩散系数都有一个很大的增强, 在尾部包括磷杂质本身在内。这种现象常用 $(P^+)(V^{2-})$ 对解离而造成 V^- 空位的过饱和来解释。解离过程有



这些过量的空位将使尾区的磷加速扩散, 也使其它杂质的扩散系数加大。

尾区的扩散系数由下式表示

$$\begin{aligned} D_{\text{TAIL}} &= D_i^0 + D_i^- \frac{[V^-]}{[V^-]_i} \\ &= D_i^0 + D_i^- \left\{ \frac{n_s^2}{n_s^2 n_i} \exp\left(\frac{3\Delta E_g}{kT}\right) \left[1 + \exp\left(\frac{0.3\text{eV}}{kT}\right) \right] \exp\left(-\frac{x-x_s}{L_v}\right) \right\} \quad (9-43) \end{aligned}$$

式中, n_s 为表面电子浓度; n_s 为一特征浓度, 是费米能级处于导带以下 0.11eV 时的电子浓度; x_s 是 $n=n_s$ 处的位置, 即表面区的终止点; L_v 为过剩空穴的衰减长度, 它与工艺条件有关。在 SUPREM III 中取 $L_v = 25\mu\text{m}$ (在 $900^\circ\text{C} \sim 1000^\circ\text{C}$ 之间)。

(3) 过渡区

过渡区从 x_s 处的扩散系数变化到尾区的 D_{TAIL} 。过渡区的厚度在 857°C 时约 150nm, 1000°C 时约为 50nm, 超过 1500°C 时将消失。对这一过渡区的情况研究得还不够。但是可以认为 $(P^+)(V^{2-})$ 的解离并不是在某一点发生的, 而是在一定的距离内不断发生, 因此形成过渡区。

分析表明, 在该区中扩散系数随深度而增加, 并正比于 n^{-2} 。因此过渡区的扩散系数是将 x_s 处的 D_v 乘以 $(n_s/n)^2$, 直到变化到尾区的开始为止。

(4) 基区宽度变宽

由于磷扩散造成的过剩空位的衰减长度很大, 因而空位的过饱和将在整个尾区甚至在合金结处都存在。这样不仅加大了尾区中磷的扩散系数, 而且也增强了尾区和尾区以下区域中其它杂质的扩散。例如, 发射极中磷杂质的存在, 会增强基区中硼的扩散, 而造成基区宽度变宽。硼扩散系数的增强程度用一增强系数 G 来表示。

$$D_{\text{基区}}(B) = D(B) \cdot G \quad (9-44)$$

5. 氧化增强扩散

实验表明, 正在氧化着的表面下的扩散行为与非氧化条件下的扩散行为很不一样。

对硼、磷、砷呈氧化增强现象 (OED), 对锑则出现氧化阻碍现象 (ORD)。

发生 OED 时扩散系数有

$$D = D_N + \Delta D_{ox} \quad (9-45)$$

式中, D_N 为非氧化条件下的扩散系数; ΔD_{ox} 为氧化增强扩散项, 它与氧化条件有关。对于 OED 而言, ΔD_{ox} 有以下公式

$$\Delta D_{ox} = f_H D_{Si}^i \left(\frac{C_v^i}{C_v} \right) \frac{K_1 \exp(-x/L_I)}{\rho [1 + K_2 P_{HCl}^{0.5}]} \left(\frac{dx}{dt} \right)^{0.5} \quad (9-46)$$

式中, $f_H = K e^{-E_A/(kT)}$ 。它由实验确定, 因由于间隙机构而引起的替代扩散的比例, 随杂质而不同; D_{Si}^i 为非氧化条件下本征空位扩散时的扩散系数; C_v^i 为本征情况下硅中空位浓度; C_v 为硅中空位浓度; ρ 为扭曲强度。对硼、磷、砷的 f_H 有

$$f_H(B) = 4.09 \exp\left(\frac{-0.48 \text{ eV}}{kT}\right)$$

$$f_H(P) = 5.50 \exp\left(\frac{-0.57 \text{ eV}}{kT}\right)$$

$$f_H(As) = 6.85 \exp\left(\frac{-0.63 \text{ eV}}{kT}\right)$$

间隙原子的浓度随 x 深度而下降, 并服从

$$K_1 \exp(-x/L_I)$$

的关系。其中 L_I 为间隙原子的衰减长度。此外, 过剩间隙硅原子的浓度近似地随氧化速率的 n 次方而变化, 即正比于 $\left(\frac{dx}{dt}\right)^n$ 。 n 值从 0.3~1.0, 多数为 0.4~0.5。在 SUPREM III 中取 $n=0.5$ 。

在 HCl 气氛中氧化时, 氯与硅的反应会产生空位, 这些空位会与间隙复合, 因而减小了注入衬底中的过剩间隙原子的数目。因此随着 HCl 成份的增加, ΔD_{ox} 会有所下降。

对锑杂质会出现 ORD 现象。因为对于锑来讲, 通过间隙扩散的可能性很小, 即 $f_H(Sb)$ 很小。扩散几乎完全通过空位。在氧化过程中, 空位起作用的扩散的减少将超过可忽略的间隙扩散的增大, 其总效果是扩散被阻碍。

三、离子注入模型

对于砷和磷杂质, 离子注入后的分布可以由两个半高斯曲线来表示。

对于硼杂质, 发现其分布更符合修正后的 Pearson IV 分布。

从表面到峰值后的某一距离内, 杂质分布可用标准的 Pearson IV 分布加以描述。但对于峰值以后的区域就不再符合。这个区域呈指数性变化, 因此必须加以修正, 即在浓度下降到峰值的一半处, 在标准的 Pearson IV 分布上连接一个指数性下降曲线。这种修正示于图 9-11。

SUPREM III 中还提供了一种利用玻尔兹曼传输方程求解杂质分布的方法。

实际的器件往往是在硅衬底上生长或淀积一层甚至几层不同成份的薄膜 (如二氧化硅、光刻胶等), 因而被注入的对象是一种多层材料。

现以砷注入覆盖有二氧化硅层的硅材料为例, 说明注入多层目标时的过程, 这示于图 9-12。

注入的砷离子进入二氧化硅层后可以被硅原子和氧原子所散射。被撞击的硅和氧原子会产生向衬底方向的运动。入射离子中的一部分会由于反弹作用而留在二氧化硅层内, 另

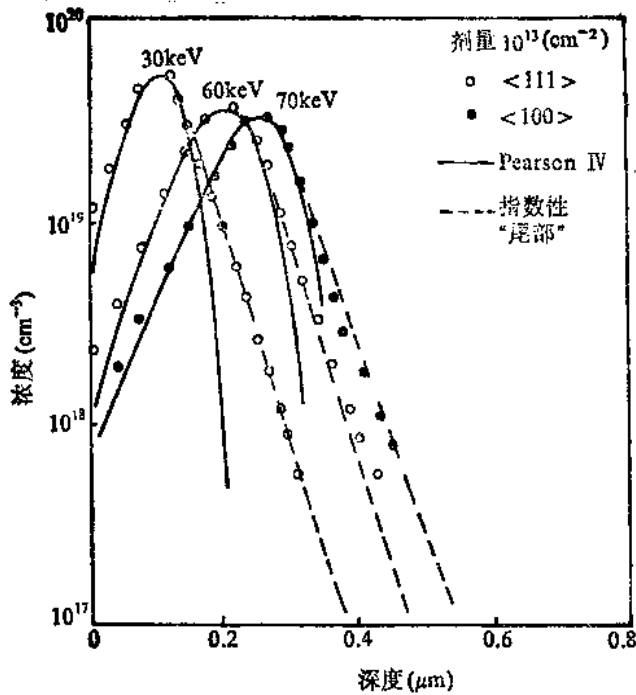


图 9-11 Pearson IV 分布及修正

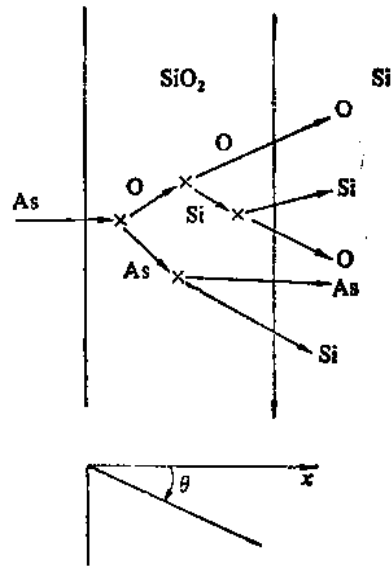


图 9-12 离子注入多层目标的过程

一部分会进入硅衬底。留在二氧化硅中的砷离子会与入射原始束流再碰撞而引起二级注入。

现利用玻尔兹曼传输方程求出原始粒子和反弹粒子的动量 P 作为距离 x 的分布函数。这时假设目标是无定形的或以随机方向排列的，在所有平行于表面的方向上具有平移对称性。这样处理多层目标时就很简单，只要改变原子数密度(每单位体积的克分子数)和散射截面的值，就可得到动量随深度的分布。

现假设注入离子具有动量 P ，经散射后变成动量 P' 。被散射的粒子其动量低于某一固定值后就认为停止运动，而从动量分布中除去。目标原子的散射截面为 $d\sigma(P \rightarrow P')$ 及 $d\sigma(P' \rightarrow P)$ ，前者使 P 减小，后者使 P 增大。 $F(P)$ 和 $F(P')$ 为具有动量 P 和动量 P' 的粒子的动量分布，因而根据玻尔兹曼传输方程有

$$\frac{\partial F(P)}{\partial x} = N \left[\frac{F(P') d\sigma(P' \rightarrow P)}{\cos \theta_{P'}} - \frac{F(P) d\sigma(P \rightarrow P')}{\cos \theta_P} \right] + Q(P) \quad (9-47)$$

式中， N 为目标原子数密度； θ 为粒子与 x 轴间的夹角； $Q(P)$ 为处在静止状态而重新获得动量的粒子产生项。

假设 $x=0$ 处的分布函数为已知。反弹粒子的分布在 $x=0$ 处为零，原始粒子的动量分布取 δ 函数，则

$$F(P, 0) = \Phi \delta(P - P_0) \quad (9-48)$$

这里， Φ 是总的注入剂量 (cm^{-2})，

P_0 是入射束中的离子动量。

根据这一初始条件，通过数值积分就可得到 $x > 0$ 处的动量分布。

目前 SUPREM II 还不能模拟离子注入时的“沟道”效应。

四、外延模型

外延掺杂过程可由图 9-13 来表示。

在外延反应腔中分成三个区：（1）主气流区；（2）界面区；（3）吸附层。主气流由载流气体携带硅源和杂质源进入反应腔，在抵达界面层（它在硅表面上部，约 5mm 厚）时，杂质分子通过质量传输穿过该层而到达吸附层。吸附层位于硅的表面，它吸附杂质原子和硅原子。

外延掺杂过程可有以下步骤。现以掺砷为例（参见图 9-13(b)）：

（1）气相中砷分子 AsH_3 靠近硅表面，并被吸附。

（2）一旦吸附的 AsH_3 被分解，就形成砷原子。

（3）砷原子会沿表面移动，在到达某一结合位置时进入硅晶格。

（4）砷原子被接着抵达的硅原子快速覆盖，这时掺杂原子将受硅中扩散过程所支配。

实验表明，外延掺杂过程主要取决于吸附层中的物理机构，即外延掺杂的过渡过程取决于吸附层中达到一种新的稳态所需要的时间。

在 SUPREM II 中是通过求解费克第二定律得到杂质分布

$$\frac{\partial C(x, t)}{\partial t} = D \frac{\partial^2 C(x, t)}{\partial x^2} \quad \infty > x \geq x_i \quad (9-49)$$

其初始条件为 $C(x, 0) = f_1(x)$ ，而边界条件分别有

$$D \frac{\partial C}{\partial x} \Big|_{x \rightarrow \infty} = 0$$

$$D \frac{\partial C}{\partial x} \Big|_{x=x_i} = f_2(t)$$

式中， C 是固体硅中杂质浓度； D 是杂质在固体硅中的扩散系数； x_i 是移动的气-固相界面的位置。

在单位面积的吸附层中，杂质的净变化率可由下式表示

$$K_A \frac{dC(x, t)}{dt} = K_m \left[P_D^0(t) - \frac{C(x, t)}{K_P} \right] - gC(x, t) + D \frac{\partial C(x, t)}{\partial x} \Big|_{x=x_i} \quad (9-50)$$

这里第一项表示吸附层中杂质增加的速率。它是杂质从气相进入吸附层和从吸附层返回气相的交换的结果。第二项表示在外延生长过程中，因硅原子的覆盖而从吸附层中移走掺杂原子的速率。第三项表示吸附层与体硅之间杂质原子的扩散交换。

式中， K_m 为外延掺杂受速率限制时的动力学系数； P_D^0 为掺杂物分压强； C 为外延生长时

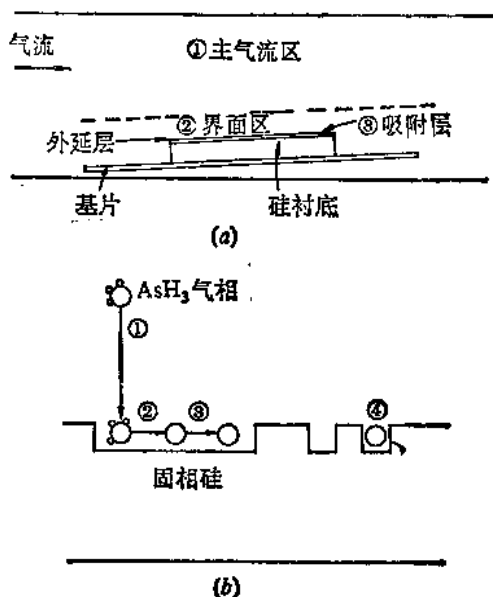


图 9-13 外延掺杂过程

固相硅中的杂质浓度; K_F 是反映气相中掺杂物浓度与固相硅中杂质浓度之间关系的热力学常数; K_A 是反映吸附层中掺杂物浓度与固相硅中杂质浓度之间关系的热力学常数; g 是外延生长速率。移项整理后可得

$$D \frac{\partial C(x,t)}{\partial x} \Big|_{x=x_f} = -K_m \left[P_D(t) - \frac{C(x,t)}{K_F} \right] + gC(x,t) + K_A \frac{dC(x,t)}{dt} = f_2(t) \quad (9-51)$$

再加上另一边界条件

$$D \frac{\partial C(x,t)}{\partial x} \Big|_{x=\infty} = 0$$

就可求解费克第二定律。

五、多晶硅的模型

目前还没有充分的数据说明晶粒尺寸、多晶硅中的掺杂分布与工艺参数如温度、时间、多晶层厚度之间的关系。因此 SUPREM III 中不得不采用简化模型。为了求得数值解, 还采取迭代方法。SUPREM III 中计算多晶硅物理和电学参数的流程图示于图 9-14。

SUPREM III 中假设所有的晶粒都具有相同的尺寸, 且形状是简单的球形。初始的晶粒大小在常压 CVD 时有

$$L = 2.1 \times 10^6 \exp\left(\frac{-0.645\text{eV}}{kT}\right) \text{ \AA} \quad (9-52)$$

对低压 CVD, 晶粒尺寸近似为

$$L = 1\text{nm} \quad T < 600^\circ\text{C}$$

$$L = 0.25Z \quad T \geq 600^\circ\text{C} \quad (Z \text{ 为多晶层厚度})$$

此外还假设初始的掺杂分布是均匀的

$$C_T = C_G = Q_{\text{POLY}}/Z \quad (9-53)$$

式中, Q_{POLY} 为单位面积多晶层中掺杂量; C_G 为每个晶粒中掺杂浓度; C_T 为多晶层中掺杂浓度。

1. 晶粒生长

晶粒生长被认为是一种受扩散控制的过程。硅原子穿过边界时, 晶粒边界会变化。在

平衡时两个方向的扩散相抵消, 边界不变。但在非平衡时, 由于边界两边硅原子自由能的差别, 就存在一个驱动力而引起净的扩散运动。这个驱动力与边界的曲率和晶粒大小有关, 现设

$$F = ab^2\lambda/L \quad (9-54)$$

式中, λ 为晶粒边界界面能; b 为晶格常数; a 为无尺寸的几何因子。该值近似为 6。

晶粒生长速率正比于扩散流, 即

$$\frac{dL}{dt} = F\mu \quad (9-55)$$

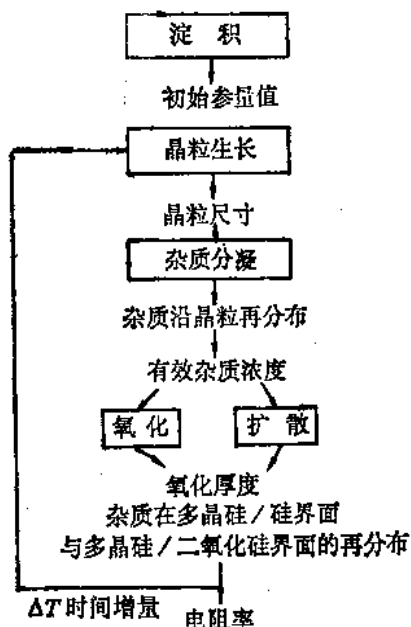


图 9-14 计算多晶硅物理和电学参数的流程图

式中, μ 为等效的边界“迁移率”, $\mu = \frac{D^s}{kT}$ 。 D^s 是晶粒边界处的硅自扩散系数。将公式 (9-54) 和 μ 的表达式代入公式 (9-55) 得

$$\frac{dL}{dt} = \left(\frac{ab^2}{kT} \right) \left(\frac{D^s \lambda}{L} \right) \quad (9-56)$$

积分后得

$$L^2 = L_0^2 + \int_0^t \left(\frac{2ab^2}{kT} \right) (D^s \lambda) dt$$

即

$$L = \sqrt{L_0^2 + K_V^s t} \quad (9-57)$$

其中

$$K_V^s = \left(\frac{2ab^2}{kT} \right) (D^s \lambda)$$

当初始晶粒尺寸 L_0 很小时, L 随时间的平方根而增大。

然而在实际情况中, 如重掺杂的多晶硅中, 自扩散系数 D^s 和晶粒边界界面能 λ 在晶粒生长中都会变化, 因而 SUPREM III 中采用增量形式

$$\Delta L = \left(\frac{ab^2}{kT} \right) \left(\frac{D^s \lambda}{L} \right) \Delta t \quad (9-58)$$

在 Δt 时间内, 认为 D^s 和 λ 是常数。

当晶粒尺寸变得与多晶硅层厚度 Z 可比时, 晶粒将与上下界面相遇, 这时发现晶粒生长速率下降。

2. 杂质分凝

在高温时, 多晶硅层中的某些杂质 (如磷和砷) 会向晶粒边界分凝。由于边界处的杂质在电学上是不活泼的, 因而减少了有效的掺杂程度。这种分凝过程是退火温度、时间、晶粒大小和掺杂浓度的函数。SUPREM III 中粗略地用一随时间成指数变化的函数加以近似

$$C_G(t) = C_G(\text{平衡}) + \Delta C_G \exp(-t/\tau) \quad (9-59)$$

式中, $\tau = 4.55 \times 10^{-4} \exp\left(\frac{0.86\text{eV}}{kT}\right) \text{h}$ 。

每次高温处理后, 晶粒内的平衡杂质浓度都要按照 (9-59) 公式的结果和新的晶粒尺寸作一次修改, 并求出新的电学上有效的杂质浓度。

除了晶粒与晶粒边界之间的杂质分凝, 还有多晶硅与氧化界面之间及多晶硅与单晶硅界面之间的杂质分凝。

3. 氧化

可以利用单晶硅的氧化模型加以适当修改以用于模拟多晶硅的氧化。

多晶硅中存在不同的晶向, 主要有三个晶向即 $\langle 100 \rangle$ 、 $\langle 111 \rangle$ 和 $\langle 110 \rangle$, 通常 $\langle 110 \rangle$ 是主要的。因此多晶硅的 B/A 速率常数可对这三个晶向的 B/A 作加权平衡, 即

$$B/A = K_1(B/A)_{\langle 100 \rangle} + K_2(B/A)_{\langle 110 \rangle} + K_3(B/A)_{\langle 111 \rangle}$$

$$\approx (B/A)_{\langle 110 \rangle}$$

(9-60)

B 速率常数与晶向无关, 因此可采用单晶硅的 B 值。

对于低掺杂的多晶硅, 实践证明这种近似是可取的。但在重掺杂时就不再符合, 重掺杂时多晶硅的氧化速率比不掺杂时快得多。这是因为重掺杂时晶粒边界处的杂质分凝会增加电学上有效的杂质浓度, 而有效的杂质浓度又增加总的空位浓度, 因而影响 B/A 值。对于很重的掺杂, 可以让 B/A 值接近无穷大, 而 B 仍保持单晶硅的值。

SUPREM III 先计算多晶硅中电学上有效的掺杂浓度 n , 再用单晶硅氧化时的模型求等效的 B/A 值, 然后代入氧化模型求氧化层厚度。显然这种方法是不严格的。

4. 扩散

多晶硅中的扩散比单晶中快得多, 其主要原因是沿着晶粒边界的快速扩散。目前还很难有一个适当的模型。SUPREM III 中假设多晶硅中的扩散是如此之快, 以致相对于热处理时间来讲, 多晶硅内的杂质分布是均匀的。

5. 氧化增强扩散

上部多晶界面在氧化时会产生过量的硅间隙原子, 但当硅间隙原子向衬底硅扩散时, 晶粒边界会吸收这些间隙原子而使它们明显减少, 因此氧化增强扩散现象就减弱。

现近似认为间隙原子随深度指数性衰减, 并把单晶硅中 OED 公式用到多晶硅/体硅结构的硅衬底中。

$$D = D_N + \Delta D_{ex} \exp(-Z/Z_I) \quad (9-61)$$

式中, Z_I 是等效间隙原子的衰减长度。它与工艺有关, 典型值为 100nm。

六、氮化硅的氧化模型

氮化硅也要被氧化, 虽然这种氧化过程很慢。SUPREM III 中采用一经验公式。

在水汽氧化时

$$\Delta x_n = 1.1 \times 10^7 \left[\exp\left(\frac{-1.9\text{eV}}{kT}\right) \right]^{0.7} P_{\text{H}_2\text{O}} \quad (9-62)$$

对于干氧氧化

$$\Delta x_n(\text{O}_2) \approx 0.15 \Delta x_n(\text{H}_2\text{O})$$

这一经验公式对于 900~1000°C 时, H_2O 的压力在 1~10 个大气压内范围内可提供足够精确的模拟结果。但对于低温或压力超出上述范围时则模拟结果不再正确。

从以上讨论可以看出, 很多模型有一定的适用范围, 有些模型还是经验性的。在使用 SUPREM III 时必须了解模型的限制及适用范围, 以便对模拟结果作出正确的判断。有些参数在使用时可根据实际情况加以适当的修改, 程序的语句设计中提供了这种可能性。

第十章 计算机辅助版图设计

§ 10.1 版图设计系统的构成

无论是人机交互图形系统或者 CAE 工作站中用于全定制的版图设计站 (在 Daisy 系统中称为 Chip Master, 在 Mentor 系统中称为 Chip Station), 其硬件配置包括 CPU、主存储器、磁盘、彩色监视器、键盘、鼠标器等。有的系统还配有磁带机、协处理器以及图形加速器。在软件方面配置有版图编辑器 (Layout Editor)、版图验证和分析软件 (Layout Verification and Analysis Package)、图形发生器文件生成软件和格式转换用软件。

以 Daisy 系统的 Chip Master 为例, 进入版图编辑器状态后, 在彩色监视器上将显示如下状况, 示于图 10-1。图形终端的屏幕上将出现能显示图形的区域, 即窗口。窗口中有网格状的亮点, 称为栅格。它相当于方格纸上的格线, 但栅格的距离可以调整, 也可以通过命令使栅格不显示在屏幕上。但应该记住, 虽然栅格并不显示, 但实际它仍在起作用, 任何线条或点必然落在格点所在的位置上。窗口中有光标出现, 光标可通过鼠标器在屏幕上自由光滑地移动。窗口的左边为竖立式菜单, 菜单中列有常用的命令, 只要把光标移动到菜单中的被选项, 再按下鼠标器相应按钮, 该命令就立即

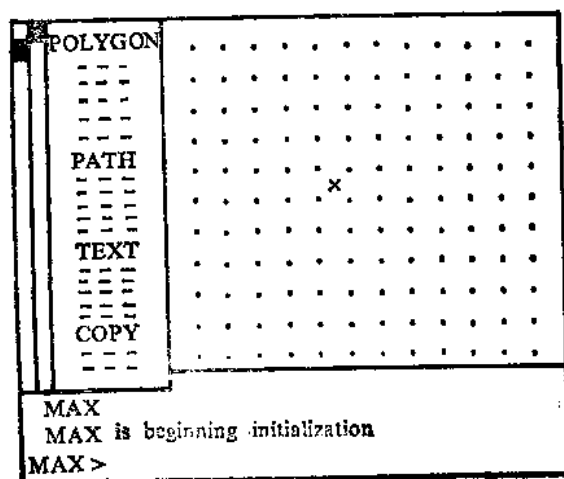


图 10-1 进入版图编辑状态后屏幕的状况

表 10-1 菜单内容举例

...RECTANGLE...
IMPLANT
NPLUS
PPLUS
POLY
CONTACT
METAL1
VIA
METAL2
...PATH...
POLY
METAL1
METAL2
...TEXT...
POLY TXT
METAL1 TXT
METAL2 TXT
BUSS1 TXT
BUSS2 TXT
USER TXT
.....
COPY
CUT-BOX
CUT-LINE
DELETE
FLIP
ROTATE
REDO
UNDO

执行, 这样可避免使用者不断敲打键盘, 使精力集中于设计工作本身。屏幕的下方为显示字符数字的区域, 送入的命令及当前状态将在该区域中显示出来。

竖立式菜单的内容, 举例于表 10-1。菜单的内容可以根据用户的需要加以改变。除了退出版图编辑状态, 该菜单总是出现在屏幕的左边而不消失。

§ 10.2 版图编辑器

版图编辑器是一种用于版图设计的交互式图形编辑工具。它支持层次式设计方法, 提供多窗口的设计环境, 除竖立式菜单外还提供一种弹出式菜单 (Pop-up Menu), 对颜色、调色板和填充图案可由用户编程加以改变, 此外还有一种重要的 UNDO 功能。在 32 位精度的情况, 在 x 和 y 轴上分别可有 4×10^9 个数据点。

一、单元和层次结构

图形编辑器提供基本的几何图形和文本。几何图形有: (1) 矩形 (Rectangle); (2) 多边形 (Polygon); (3) 具有一定宽度的线条 (Path); (4) 圆 (Circle); (5) 圆环 (Ring)。文本的类型有:

(1) 字符型文本;

(2) 笔划型文本。

利用这些基本几何图形的各种组合形成的几何形状就可表示版图的掩膜形式, 这些几何形状的集合称为单元 (Cell)。我们可以利用这些单元组成一个芯片。其相互关系示于

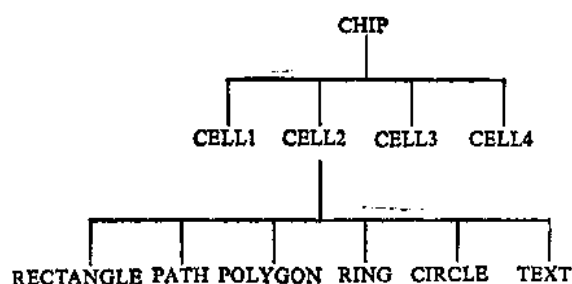


图 10-2 几何图形、单元和芯片的关系

图 10-2。

一个单元可以被放置在芯片中的多个位置, 一个被放置在某一位置上的单元称其为单元的实例 (Instance), 也就是说实体是单元的调用形式。单元之间可以通过单元实体相互调用, 但不允许出现递归调用。

为了支持自顶向下和自底向上的设计方法, 可以把各种单元组织在一个单元库

(Library) 中。库是一种特殊的文件。

在库中的单元允许有层次结构, 即允许单元的嵌套。

每个单元既可以是“父”单元, 也可以是“子”单元。如果当前窗口中显示的是 A 单元, 而在 A 单元中的单元实体为 B 单元, 则 A 单元是 B 单元的“父”单元, B 单元则是 A 单元的“子”单元。如当前窗口显示的是 B 单元, 而在 B 单元中的单元实体为 C 单元, 则 B 单元是 C 单元的“父”单元, 而 C 单元则是 B 单元的“子”单元。“父”与“子”单元的相互关系见图 10-3。

我们可以通过可视深度 VIEW-DEPTH 这一命令使某一层次的单元在当前窗口中显示其细部 (称为可视)。

如认为 A.CELL 在当前窗口中已被打开, 这时送入以下命令:

VIEW-DEPTH 0

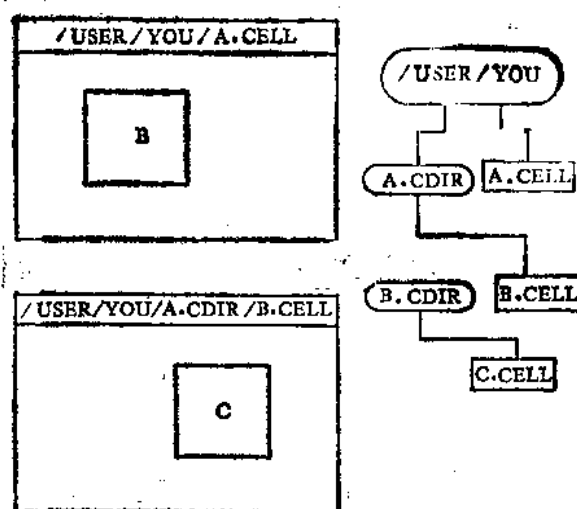


图 10-3 父/子单元的相互关系

则窗口中仅出现 A.CELL 的边框 (Bounding Box)。边框是包围该单元的最小矩形。在边框的中央有单元名“A”，示于图 10-4。

为了显示 A.CELL 中的各个图形，这时需送入命令

VIEW-DEPTH 1

在层 1 时，所有 A.CELL 中的图形变为可视。但如 A.CELL 含有任何其它单元实体时，这些单元实体也仅显示其边框，而实体的细部并不显示。如图 10-5 中，A.CELL 中含有 B 单元的实体。所以我们可以改变可视深度而看到低层次单元的细部。

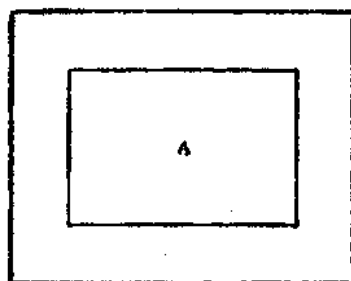


图 10-4 在可视深度为 0 时的 A.CELL

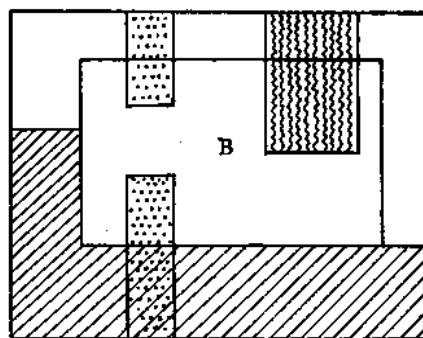


图 10-5 可视深度为 1 时的 A.CELL

应该注意，低层次单元的可视并不意味着就可以对该层次的单元进行编辑。在上述例子中可以被编辑的图形仅仅是层 1 时的那些图形。如要编辑较低层次时，可采用 CHANGE 命令，它使层次下降一级，该层的单元在当前窗口中将变成顶层，并充满整个窗口。我们也可以进入 Edit-in-Place 模式。这时可以对任何低层次的单元进行编辑，被编辑的低层次单元不充满整个窗口，但有辉亮出现，其它高层次单元的图形仍留在窗口中，可视但不可编辑。

二、多窗口环境和弹出式菜单

多窗口对大的芯片设计带来了很大方便,并加快设计周期。我们可以把不同的设计内容显示在不同的窗口中。例如整个设计显示在一个窗口中,而把需要观察或编辑的局部加以放大而显示在另一窗口中。在一个窗口内对某一单元进行的修改将会立即反映到所有显示同一单元的窗口中。甚至可以利用两个窗口来建立一个新的图形。

多窗口环境的示意图见图10-6。在多窗口环境下,任何一个窗口可以有任意的大小,并可处在屏幕的任何位置,这都取决于用户的需要。

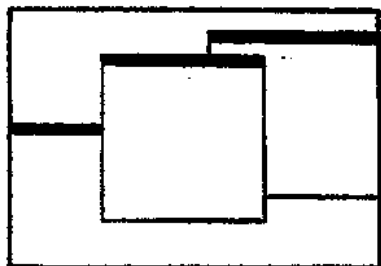


图 10-6 多窗口示意图

但当前哪个窗口为“有效”窗口,则取决于光标。当光标移入某一窗口,则该窗口变为有效,所有的命令将对该窗口起作用。当在该窗口中的工作完成后,可将该窗口移到后台,把下一次需要编辑的窗口移到前面。这有如办公桌上的公文处理一样。但在一个时间只有一个窗口是有效的。

弹出式菜单是版图编辑器所提供的一种可选项。它的优点是不占有屏幕的有效空间,仅在需要时加以弹出,不需要时立即消失,这使图形空间能够得到充分的利用。菜单的弹出位置是根据当前光标的位置而定。弹出式菜单具有多层,相互堆叠在一起。菜单的内容是可编程的,以适应不同的工作内容和适应不同用户的需要。命令在菜单中的位置也是可以改变的。此外,更为重要的一个作用是可以把几个命令或功能合在一起组成一个宏命令而定义在菜单上。

三、图形的层次

集成电路的版图具有多层次的性质,每一层对应于一块掩膜版,掩膜版的数目因工艺而不同。版图编辑器中把每一块掩膜版表示为一“层”(Layer)。并定义某一层号对应于某一特定的掩膜版。例如定义 layer3 为扩散掩膜版,layer4 表示多晶硅掩膜版等等。每一层赋予一种颜色。在屏幕上显示的图形可以是某一层,也可以是若干层或所有层的重叠。

每一层所对应的是哪块掩膜版以及该层的颜色都可以根据用户的需要加以修改。

§ 10.3 版图的生成与编辑

建立一个版图文件的基本步骤如图10-7。

版图编辑器所提供的命令列于表10-2(以 Daisy 系统的 Mask Editor(简称 MAX)为例)。这些命令可分成以下几类:(1)建立图形命令;(2)编辑命令;(3)配置命令;(4)观视命令;(5)层次和库命令;(6)库管理命令;(7)访问和获得信息命令;(8)与交互规则检查有关的命令;(9)与编辑器本身有关的命令。

建立图形命令见表10-3。

编辑命令见表10-4。

配置命令用来设置设计环境的特性,见表10-5。

观视命令用来改变屏幕的状态,见表10-6。

其它命令在此就不作详述。

利用以上所述命令，我们就可以生成图形以及对生成的图形进行各种编辑，以形成所需的版图。

如需建立一个矩形，可发出以下命令

```
RECTANGLE {POINT} {POINT}
{EXECUTE}
```

这里用两个相对的坐标点来决定矩形的边界。
{EXECUTE}用来结束该命令。我们可利用鼠标器上的 {POINT}键来确定坐标点。把光标移到所需的格点，按下 {POINT} 键，再移动光标到相对的另一格点，再按下 {POINT} 键，就完成了矩形的生成。这示于图10-8。

按同样原则，我们可以生成一个多边形。如图10-9所示。图中的 {POINT}号表示送入该点的次序。

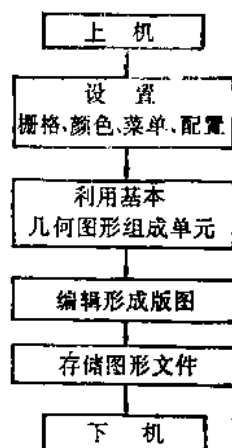


图 10-7 建立版图文件的基本步骤

表 10-2

ARRAY	LIB_DELETE	POINT
ATTRIBUTE	LIB_FETCH	POLYGON
BOOLEAN	LIB_INVENTORY	RECTANGLE
CHANGE	LIB_PURGE	RFDO
CIRCLE	LIB_REFRESH	REDRAW
CLOSE_CELL	LIB_RESTORE	RING
COPY	LIB_REVERT	ROTATE
CUT	LIST_COMMAND	SAVE
DEF	LOCATE_PARAMETER	SCALE_GRID
DELETE	LOCATOR	SEARCHPATH_CELL
DISTANCE	MAGNIFY_CELL	SELECT
DRC	MAKE_CELL	SET_GRID
DROOP	MASH	SIZING
ERROR_DELETE	MODIFY	SMASH
EXIT	MOVE	TARGET
FIND	OPEN_CELL	TEXT
FIT	PAN	UNDO
FLIP	PARAMETER	UNSELECT
HIERARCHICAL_CELL	PARENT	UPDATE
INSPECT	PATH	VIEW_DEPTH
LIB_ADD	PLACE_CELL	ZOOM
LIB_COPY	PLOT_AREA	
LIB_CREATE	PLUCK	

如欲对上述的多边形作逆时针的 90° 旋转，可发出以下命令

```
ROTATE {EXECUTE}
```

就得到图 10-10。这时旋转的原点是图 10-9 多边形的左下点。我们也可以在 ROTATE 命令中规定一个旋转的原点。

表 10-3

命 令	功 能
CIRCLE	建立一个圆
PARAMETER	在图形上附加、删除、修改参数
PATH	建立一线条
POLYGON	建立一多边形
RECTANGLE	建立一矩形
RING	建立一圆环
TEXT	在版图中放置一文本

表 10-4

命 令	功 能
COPY	拷贝图形或实体
CUT	分割图形或切去图形的边
DELETE	从一个单元中除去图形
FLIP	绕一轴翻转图形或实体
MOVE	移动图形或实体
REDO	重新执行前一次的命令
ROTATE	绕原点旋转图形或实体
SELECT	选择图形或实体
UNDO	使上一次命令的功能消失
UNSELECT	取消选择

表 10-5

命 令	功 能
ATTRIBUTE	设置当前的编辑模式
DEF	设置当前的编辑模式
MODIFY	改变某些编辑属性
SCALE_GRID	设置栅格间距
SET_GRID	决定屏幕上有多少格点是可见的

表 10-6

命 令	功 能
FIT	改变单元尺寸使其充满窗口
PAN	使当前光标位置移到窗口中央
REDRAW	重画屏幕上的内容
TARGET	指定一个目标窗口。在一个窗口中作 PAN 或 ZOOM 动作时, 在目标窗口中可看到这些动作的结果
VIEW_DEPTH	设置显示的层次
ZOOM	改变可视区域的尺寸

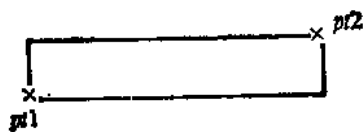


图 10-8 矩形的生成

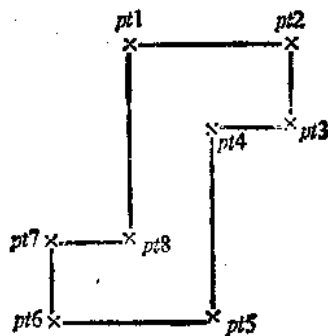


图 10-9 多边形的生成

如上述矩形形成于多晶硅层, 多边形形成于扩散层, 再在接触孔层建立二个矩形, 就得到我们所熟悉的一个 MOS 晶体管的掩膜形式。这示于图 10-11。

如对上述组合图形进行选择后, 就可以对其作拷贝。其过程示于图 10-12 和图 10-13。选择一组图形时, 发出 SELECT 命令, 再利用光标确定两个坐标点, 其次序和方向如图 10-12 所示, 即从左上方移至右下方, 则该图形即被选择。在选择后发出 COPY 命令, 并用两个坐标点决定拷贝前后两个图形的相对位置。如图 10-13 所示。应注意, 拷贝后被拷贝图形的选择状态仍存在, 如无其它需要, 一般应取消此选择。

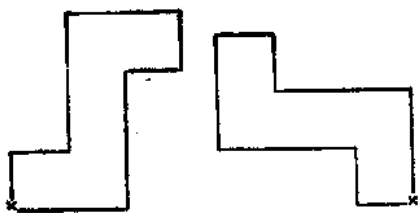


图 10-10 对图 10-9 多边形旋转的结果

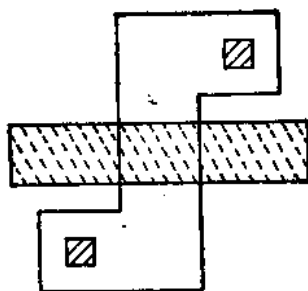


图 10-11 一个 MOS 晶体管的掩膜形式

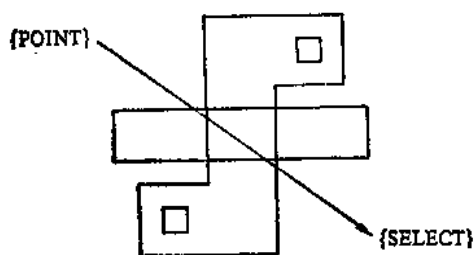


图 10-12 图形的选择

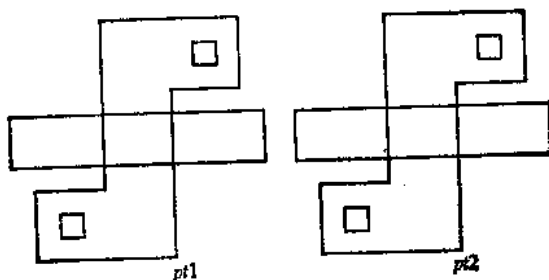


图 10-13 图形的拷贝

§ 10.4 版图验证和分析

版图验证和分析一般包括几何设计规则检查 (DRC), 电学规则检查 (ERC), 版图与电路图一致性检查 (LVS) 和版图参数提取 (LPE)。

一、几何设计规则检查

它是为了确保每个几何图形本身以及相互之间的正确关系而规定的在生产工艺中可以接受的尺寸。对于不同的工艺, 存在着不同的设计规则。即使相同的工艺, 由于各个工厂本身的需要或习惯, 设计规则也有不少差异。图 10-14 表示了硅栅 NMOS 工艺的若干几何设计规则的例子, 其中 a 为宽度, b 为间距, c 为覆盖, d 为伸展。

几何设计规则的检查可以由一系列基本操作来描述。典型的基本操作是布尔运算、检查性运算、拓扑性运算和几何性运算。这些基本图形运算示于图 10-15。现以这样一条设计规则, 即“接触孔必须离开 MOSFET 的栅区有一定的距离”为例。它可分解成以下运算, 先将多晶硅层与扩散层作 AND 操作得到 MOSFET 的栅区, 再检查该栅区与接触孔之间的距离, 如

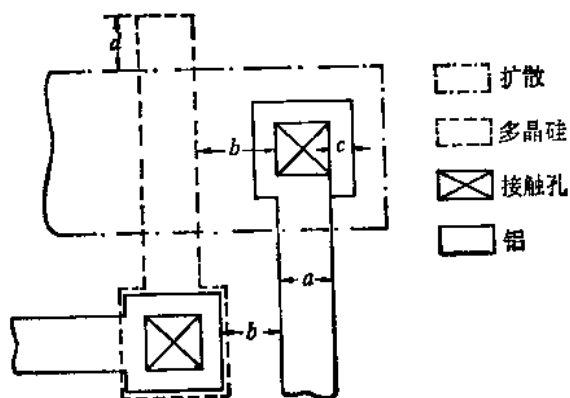


图 10-14 几何设计规则

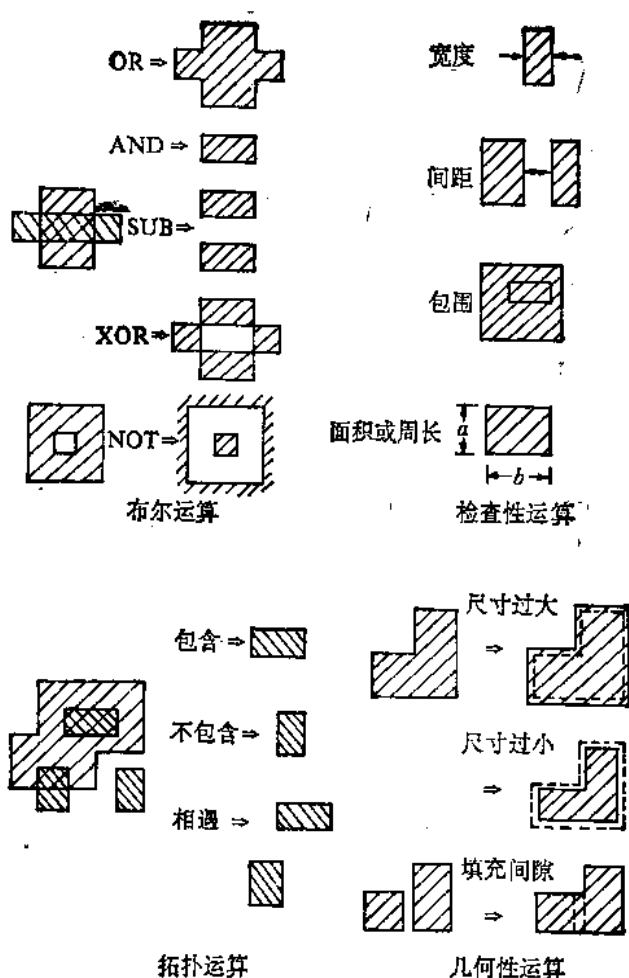


图 10-15 基本的图形运算

间距符合设计规则的要求，则通过检查。如间距小于规则要求，则发出出错信息。

二、电学规则检查

经过几何设计规则检查以后，该电路将正确地按照规则要求进行制造，但这并不能保证该电路一定能正确的工作。电路能否正确工作将取决于电路的行为分析。在简单的版图规则检查和复杂的行为级分析之间可以有一种中间性的检查，这就是电学规则检查。

电学规则是这样一些电路性能，它们只要知道版图的几何形状和连接关系就能够加以确定，而不需要了解电路的行为。这些包括开路、短路、孤立节点（浮节点）、非法器件（包括不正确的注入类型）、对电源和对地的连接是否正确以及是否存在小于最小工艺尺寸的器件等。所以电学规则检查是对网络作分析检查。

例如短路就是两个不应连接的线网之间（例如电源与地之间）存在通路。这种检查很容易，只要保证电源与地两个线网相互孤立，没有连接之处。进行这种短路检查时应注意是否有晶体管直接连接在电源与地之间，虽然这不是完全的静态短路，但这在电学上也是不允许的。

对于孤立节点（或浮点）的检查，可从外输入端开始（包括电源端、地端、时钟信号

输入端)。根据几何形状和连接关系逐步传播(检查)到版图内部,凡是能抵达的元件都作上标记。如果存在这样的节点,即无法通过电路中的元件依传播方向抵达该节点并由该节点继续传播抵达输出端,则称该节点为孤立节点。

三、版图与电路图一致性检查

它是先将实际的版图恢复成电路图,再与原要求的电路图相比较。

一般地说,版图数据只包含几何信息,而没有明显的电路信息。因此必须经过下述两个过程才能提取出电路图:(1)电路元件的识别;(2)连接性分析。不同的电路元件其结构存在差别,因而识别过程会随元件类型和采用的工艺不同而有所不同。识别过程分两步:第一步是预处理。先把与工艺有关的因素加以去除。第二步才是与工艺无关的识别。现以 NMOS 为例说明识别的具体过程。图 10-16 为其版图形式及得到的沟道区和源/漏区。

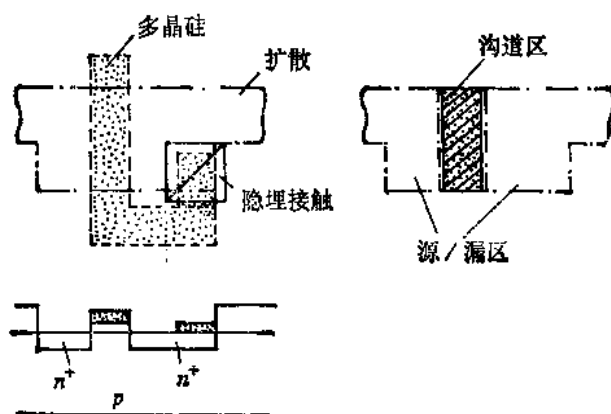


图 10-16 NMOS 版图及识别出的沟道、源漏区

第一步(预处理):

(1) 沟道区 作以下运算

$$\langle \text{沟道区} \rangle = (\langle \text{扩散层} \rangle \text{AND} \langle \text{多晶硅层} \rangle) \text{SUB} \langle \text{隐埋接触} \rangle$$

(2) 实际扩散区 作以下运算

$$\langle \text{实际扩散区} \rangle = \langle \text{扩散层} \rangle \text{SUB} \langle \text{沟道区} \rangle$$

这样就得到了沟道区和源/漏区。

第二步:

找出所有三个相接(即相遇)多边形的集合。一个多边形为栅区,其它两个多边形分别为源区和漏区,并赋以管号。把这些数据存入数据库中作为 NMOS 管的栅极端和源/漏极端。

再以双极型晶体管为例说明。双极型晶体管的版图形式示于图 10-17。

第一步(预处理)

(1) 发射极接触 作以下运算

$$\langle \text{发射极接触} \rangle = \langle \text{接触孔} \rangle \text{CONTAINED} (\langle n^+ \text{扩散层} \rangle \text{AND} \langle \text{基区扩散层} \rangle \text{AND} \langle \text{隔离层} \rangle)$$

(2) 基极接触 作以下运算

$$\langle \text{基极接触} \rangle = \langle \text{接触孔} \rangle \text{CONTAINED} (\langle \text{基区扩散层} \rangle \text{AND} \langle \text{隔离层} \rangle)$$

(3) 集电极接触 作以下运算

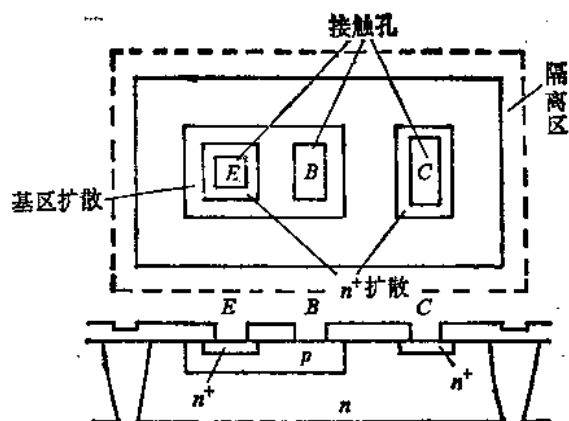


图 10-17 双极型晶体管版图

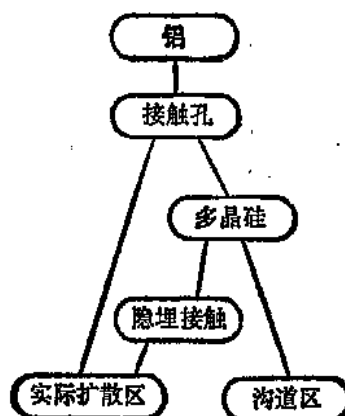


图 10-18 各层之间电学上等效关系

$\langle \text{集电极接触} \rangle = \langle \text{接触孔} \rangle \text{CONTAINED}(\langle n^+ \text{扩散层} \rangle \text{AND} \langle \text{隔离层} \rangle \text{AND} (\text{NOT} \langle \text{基区扩散层} \rangle))$

第二步:

找出这三个多边形的集合, 并赋以管号。把这些数据存入数据库作为晶体管的发射极端、基极端和收集极端。

其它电路元件如横向晶体管、二极管、电阻、电容等也用同样的方法加以识别。

连接性分析是找出各元件端口之间连接关系。我们认为在同一导电层中相连或相重的多边形在电学上是等效的。在不同层上的多边形之间在电学上的等效性取决于工艺。图 10-18 表示硅栅 NMOS 工艺各层之间电学上等效的关系。图中的连接线表示如该两层上的多边形相重叠, 则两层上的多边形在电学上等效。例如多晶硅上的多边形与扩散区上的多边形通过隐埋接触层上的孔而在电学上相连。

连接性分析步骤如下:

(1) 按照图 10-18 中的各条连接线, 搜索相应两层中有无互相重叠或相交的多边形对。并把存在重叠或相交的多边形的关系存于一个连接图中。

(2) 在该连接图中找出所有相连成份的集合, 把每一这样的集合看成为一个线网。

(3) 找出各元件端口的多边形。通过检索线网和这些多边形是属于哪个元件, 而得到线网与元件之间的关系。

经过上述步骤提取出来的电路图可用图表示出来。这时有两种画法: 一是面向版图的, 即把电路元件的符号和连线画在原始版图的同一位置上。另一是面向功能的, 它是考虑信号流, 电路对称性等因素而画成的。但是要对画出的图进行人工检查和分析实际上是很困难的, 因而只能依靠计算机来进行一致性分析。这时通常是将原始电路图的网表与提取出来电路图的网表加以比较。

四、版图参数提取

它是从实际版图中提取一些关键的电学参数, 如 MOS 管的宽长比、栅电容、交叉耦合电容等。在后模拟中将用到这些参数。

这时我们可以利用一致性检查过程中已获得的数据, 再进行简单的计算就可得到所需结果。如 MOS 管的宽长比, 在得到 MOS 管沟道区多边形 (或矩形) 的形状后, 就可得到

该管的宽长比值。

§ 10.5 图形发生器文件

图形文件不能用来直接制造掩膜版。通常掩膜版是利用光学的图形发生器或利用电子束技术的电子束曝光系统来制作的,因此 CAD 系统必须产生一种能被掩膜制造设备可读的芯片描述文件。

光学图形发生器是利用光透过一个可变的矩形光阑(狭缝),在涂有感光胶的铬版上得到矩形图形。由于这种制造设备只能得到大小不对的矩形(最大矩形的尺寸也是有限制的),因此一个具有某种形状的图形必须事先加以分割,使其成为若干个矩形。其示意图见图 10-19。

通过矩形分割所得到的矩形群决定了光阑在运动过程中的变化,也就是这些大小不同的矩形的数据控制着图形发生器光阑狭缝的开启尺寸。

为了加速图形发生器制作掩膜版的进程,经矩形分割后的数据需按 x 和 y 方向进行排序,这可避免图形发生器的来回往复运动。

经过分割和排序得到的图形文件称为图形发生器制版文件。

对于电子束曝光系统来讲,其情况也类似。虽然这种设备中有计算机控制电子束的扫描,但它可以接受的图形也只限于矩形、平行四边形和梯形三种,其它各种图形都由上述三种图形组合而成。因此同样需要进行分割和排序。

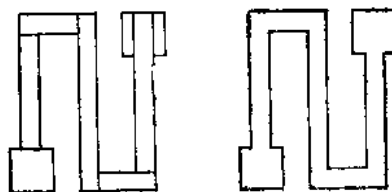


图 10-19 矩形分割

§ 10.6 格式的交换

一个好的 CAD 系统除了能与制造掩膜版的设备以及测试电路的仪器进行通讯外,还希望能与其它 CAD 系统交换设计。

目前在集成电路领域中,比较通用的交换格式有:

- (1) Calma GDS II 格式;
- (2) CIF 格式 (Caltech Intermediate Format);
- (3) EDIF 格式 (Electronic Design Interchange Format)。

表10-7列出了这些格式所具有的功能。

CalmaGDS II 格式是一种应用广泛的格式。它所以成为一种标准格式是由于它进入商业领域较早,其它制造商为了与其兼容而不得不采用它。从表 10-7 中可以看出,除了能表示掩膜的几何图形外,它还可以表示拓扑信息和其它属性,包括文本、数组、结构和层次。这是因为 GDS II 格式不仅是一种输出格式,而更重要的它是 Calma CAD 系统的数据库的完整描述。但它是一种中间格式,不能直接用于制作掩膜,而必须经过一次转换。GDS II 是二进制的流式 (Stream) 格式,它占有较少的空间但无法使人读懂。

对于 CIF 格式,不仅机器可读且人也可读。它用一组文本式命令来设置掩膜层和图形。

表 10-7 输出格式的功能

功能 格式名	层次性	对人的 可读性	内 容			几 何 图 形				
			拓 扑	几何图形	行 为	线	多边形	圆	曲 线	文本
GDS I	有	无	有	有	无	有	有	无	无	有
CIF	有	有	无	有	无	无	有	有	无	无
EDIF	有	有	有	有	有	有	有	有	有	有

如 LAYER(L) 设置掩膜层

BOX(B) 矩形

WIRE(W) 线条

ROUNDFLASH(R) 圆

POLYGON(P) 封闭多边形

但它不能进行较高层次的图形操作如画曲线和文本。CIF 格式允许层次式描述, 因此很容易建立 CIF 文件, 而又能用于描述复杂图形。

此外, CIF 格式中规定了一组控制语句, 如

DS 子程序定义的开始

DE 子程序定义的结束

DD 子程序定义的删除

0~9 附加用户指定的信息

END CIF 文件终止

CIF 格式还允许把多个 CIF 文件聚集在一个文件中, 这样就可描述“多设计项目”(Multi-project) 芯片。但它也是一种中间格式, 无法直接送入掩膜版制造设备, 而必须进行一次转换。目前 CIF 格式被广泛用于各类 CAD 系统的信息交换。例如 Calma CAD 系统与 Applicon CAD 系统之间的交换过程如下:

Calma GDS I $\rightleftharpoons$ CIF $\rightleftharpoons$ Apple 860
(Calma CAD 系统) (Applicon CAD 系统)

Applicon CAD 系统的格式称 Apple 860 格式。

最新提出的交换格式为 EDIF 格式。EDIF 是把 VLSI 设计的所有方面都归纳在单一的表示法中。在 EDIF 版本 200 中, 它可描述 10 个视窗 (View)。它不仅能表示几何图形, 还能包括拓扑和行为级, 因而不丢失任何高层次的信息。它不仅可以用作各种 CAD 系统之间的交换格式, 而且对制造设备来讲也是一种有用的通讯介质。EDIF 的设计不仅人容易读懂, 而且机器很容易进行“翻译”。

EDIF 是一种文本性格式, 它类似于 LISP 编程语言。每个语句的开始是关键字, 接着是参数。每个语句由括号括起来。而且, 整个 EDIF 文件是结构化的。参数可以是其它一些语句, 所以这意味着, 一个 EDIF 文件只含有一个语句。

虽然 EDIF 企图包含 CAD 系统中所有感兴趣的内容, 但它仍然是不完善的, 正在不断地修改之中。EDIF 的设计者希望它能代替所有其它格式而成为一种工业标准。

附录 I 算法基础

在大量模拟工具中，我们要求解的是非线性的偏微分方程。

一般讲，求微分方程的解析解是很困难的，只有一些很简单的微分方程，例如常系数常微分方程才能得到分析形式的解。

对大多数微分方程，我们需要采用数值求解，即求出区间内某些点上的函数近似值。如果该区间内的点取得足够密，则这种数值解会是实际过程的一种很好的近似。由于这种方法是在区间内分点求值，故称为离散数值求解法。

求解非线性微分方程的第一步是对微分方程进行离散化，使其成为非线性代数方程，通常离散数值分析法中，最常用的是有限差分法和有限元法。

如采用有限差分法，则经离散化的非线性微分方程将变为非线性差分方程。

求解的第二步是对非线性差分方程加以线性化。通常采用的有牛顿-拉夫森法和阻尼牛顿法等。

第三步是对线性代数方程求解，求解的方法通常有直接法和迭代法两种。

在此附录中，我们将介绍一些最常用的算法。

I-1 微分方程离散化——有限差分法

一、一阶导数的有限差分近似

为简单起见，我们只讨论一维情况。设有函数 $u(x)$ ，其自变量区间为 $0 \leq x \leq x_L$ 。

现将整个区间 $[0, x_L]$ 划分为 L 段，则有 $L+1$ 个离散点 x_i ($i=0, 1, 2, \dots, L$)。每一离散点称为网格点 (Mesh Point 或 Grid Point)。

$$\begin{array}{c} \leftarrow h_- \rightarrow \leftarrow h_+ \rightarrow \\ \cdot \quad \cdot \quad \cdot \\ \hline x_{i-1} \quad x_i \quad x_{i+1} \end{array}$$

相邻网格点距离称网格间距。

$$x_{i+1} - x_i = h_+$$

(I-1)

$$x_i - x_{i-1} = h_-$$

1. 前差分近似

根据台劳定理，函数在网格点 x_{i+1} 的值 $u(x+h_+)$ 可展开为

$$u(x+h_+) = u(x) + h_+ \left. \frac{du}{dx} \right|_x + \frac{h_+^2}{2} \left. \frac{d^2u}{dx^2} \right|_{x+\theta_1 h_+} \quad 0 \leq \theta_1 \leq 1 \quad (\text{I-2})$$

所以

$$\left. \frac{du}{dx} \right|_x = \frac{u(x+h_+) - u(x)}{h_+} - \frac{h_+}{2} \left. \frac{d^2u}{dx^2} \right|_{x+\theta_1 h_+} \quad (1-3)$$

如取一级近似, 则

$$\left. \frac{du}{dx} \right|_x = \frac{u(x+h_+) - u(x)}{h_+} \quad (1-4)$$

即为前差分近似。其误差为

$$|E| \approx \frac{h_+}{2} \max[x, x+h_+] \left| \frac{d^2u}{dx^2} \right| \quad (1-5)$$

这说明误差 E 和 h_+ 的一次方成正比。

2. 后差分近似

同样根据台劳定理, 函数在网格点 x_{i-1} 的值 $u(x-h_-)$ 可展开为

$$u(x-h_-) = u(x) - h_- \left. \frac{du}{dx} \right|_x + \frac{h_-^2}{2} \left. \frac{d^2u}{dx^2} \right|_{x-\theta_2 h_-} \quad 0 \leq \theta_2 \leq 1 \quad (1-6)$$

因而

$$\left. \frac{du}{dx} \right|_x = \frac{u(x) - u(x-h_-)}{h_-} + \frac{h_-}{2} \left. \frac{d^2u}{dx^2} \right|_{x-\theta_2 h_-} \quad (1-7)$$

如取一级近似, 则

$$\left. \frac{du}{dx} \right|_x = \frac{u(x) - u(x-h_-)}{h_-} \quad (1-8)$$

即为后差分近似。其误差为

$$|E| \approx \frac{h_-}{2} \max[x-h_-, x] \left| \frac{d^2u}{dx^2} \right| \quad (1-9)$$

也与 h_- 的一次方成正比。

3. 中心差分近似

如在台劳展开时取高次项, 则得到误差更小的中心差分近似

$$\begin{cases} u(x+h_+) = u(x) + h_+ \left. \frac{du}{dx} \right|_x + \frac{h_+^2}{2} \left. \frac{d^2u}{dx^2} \right|_x + \frac{h_+^3}{6} \left. \frac{d^3u}{dx^3} \right|_{x+\theta_3 h_+} & 0 \leq \theta_3 \leq 1 \\ u(x-h_-) = u(x) - h_- \left. \frac{du}{dx} \right|_x + \frac{h_-^2}{2} \left. \frac{d^2u}{dx^2} \right|_x - \frac{h_-^3}{6} \left. \frac{d^3u}{dx^3} \right|_{x-\theta_4 h_-} & 0 \leq \theta_4 \leq 1 \end{cases} \quad (1-10)$$

将上两式相减得

$$\begin{aligned} u(x+h_+) - u(x-h_-) &= (h_+ + h_-) \left. \frac{du}{dx} \right|_x \\ &\quad + \frac{(h_+^3 + h_-^3)}{6} \left(\left. \frac{d^3u}{dx^3} \right|_{x+\theta_3 h_+} + \left. \frac{d^3u}{dx^3} \right|_{x-\theta_4 h_-} \right) \end{aligned} \quad (1-11)$$

近似后则得中心差分近似表达式

$$\left. \frac{du}{dx} \right|_x = \frac{u(x+h_+) - u(x-h_-)}{h_+ + h_-} \quad (1-12)$$

其误差 $|E|$ 与 $(h_+^3 - h_+h_- + h_-^3)$ 成正比。

如 $h_+ = h_- = h$, 则 (I-12) 式改为

$$\left. \frac{du}{dx} \right|_x \approx \frac{u(x+h) - u(x-h)}{2h} \quad (\text{I-13})$$

二、二阶导数的有限差分近似

为得到二阶导数的有限差分近似表达式, 在台劳展开时取更高一级的形式

$$\begin{cases} u(x+h_+) = u(x) + h_+ \left. \frac{du}{dx} \right|_x + \frac{h_+^2}{2} \left. \frac{d^2u}{dx^2} \right|_x \\ \quad + \frac{h_+^3}{6} \left. \frac{d^3u}{dx^3} \right|_x + \frac{h_+^4}{24} \left. \frac{d^4u}{dx^4} \right|_{x+\theta_5 h_+} \quad 0 \leq \theta_5 \leq 1 \\ u(x-h_-) = u(x) - h_- \left. \frac{du}{dx} \right|_x + \frac{h_-^2}{2} \left. \frac{d^2u}{dx^2} \right|_x \\ \quad - \frac{h_-^3}{6} \left. \frac{d^3u}{dx^3} \right|_x + \frac{h_-^4}{24} \left. \frac{d^4u}{dx^4} \right|_{x-\theta_6 h_-} \quad 0 \leq \theta_6 \leq 1 \end{cases} \quad (\text{I-14})$$

将上两式相加并取近似得

$$\left. \frac{d^2u}{dx^2} \right|_x = \frac{\frac{u(x+h_+) - u(x)}{h_+} - \frac{u(x) - u(x-h_-)}{h_-}}{\frac{1}{2}(h_+ + h_-)} \quad (\text{I-15})$$

即二阶导数有限差分近似表达式。

如 $h_+ = h_- = h$, 则公式 (I-15) 简化为

$$\left. \frac{d^2u}{dx^2} \right|_x = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \quad (\text{I-16})$$

其误差 $|E|$ 与 h^2 成正比。

I-2 非线性方程的线性化

一、牛顿-拉夫森 (Newton-Raphson) 法 (简称 NR 法)

如有一非线性方程组 $F(\mathbf{x})=0$ 。解此方程的方法是在初始“猜解” $\mathbf{x}^0 = \mathbf{x}_0$ 处作台劳展开, 而得改进解 $\mathbf{x}^1$ 。

台劳展开后并截去高次项得

$$F(\mathbf{x}^1) = F(\mathbf{x}^0 + \mathbf{x}^1 - \mathbf{x}^0) \approx F(\mathbf{x}^0) + F'(\mathbf{x}^0)(\mathbf{x}^1 - \mathbf{x}^0) = 0 \quad (\text{I-17})$$

移项后有

$$F'(\mathbf{x}^0)(\mathbf{x}^1 - \mathbf{x}^0) = -F(\mathbf{x}^0) \quad (\text{I-18})$$

这里 F' 称为雅可比矩阵。

由于已知 $\mathbf{x}^0$, $F(\mathbf{x}^0)$ 和 $F'(\mathbf{x}^0)$, 因而可求得 (I-18) 线性方程组的 $\mathbf{x}^1$ 。 $\mathbf{x}^1$ 比 $\mathbf{x}^0$ 更接近最终解。现将 $\mathbf{x}^1$ 作为“初始”解, 在 $\mathbf{x}^1$ 点再作线性化, 得

$$F'(x^1)(x^2 - x^1) = -F(x^1) \quad (I-19)$$

即可求得 x^1 。

如此反复进行直到得到最终解。NR 法的迭代公式的一般形式为

$$F'(x^k)(x^{k+1} - x^k) = -F(x^k) \quad k=0, 1, \dots \quad (I-20)$$

或

$$x^{k+1} = x^k - J(x^k)^{-1} \cdot F(x^k) \quad (I-21)$$

这里 $J(x^k)$ 是 $F(x)$ 的雅可比矩阵

$$J(x^k) = \begin{bmatrix} \left. \frac{\partial F_1}{\partial x_1} \right|_{x_1^k} & \left. \frac{\partial F_1}{\partial x_2} \right|_{x_2^k} & \dots & \left. \frac{\partial F_1}{\partial x_n} \right|_{x_n^k} \\ \left. \frac{\partial F_2}{\partial x_1} \right|_{x_1^k} & \left. \frac{\partial F_2}{\partial x_2} \right|_{x_2^k} & \dots & \left. \frac{\partial F_2}{\partial x_n} \right|_{x_n^k} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial F_n}{\partial x_1} \right|_{x_1^k} & \left. \frac{\partial F_n}{\partial x_2} \right|_{x_2^k} & \dots & \left. \frac{\partial F_n}{\partial x_n} \right|_{x_n^k} \end{bmatrix} \quad (I-22)$$

k 代表第 k 次迭代。

当有

$$|x^{k+1} - x^k| < \varepsilon_s \quad (I-23)$$

或

$$\left| \frac{x^{k+1} - x^k}{x^k} \right| < \varepsilon_r \quad (I-24)$$

时, 就认为迭代收敛, 即 x^{k+1} 就是方程 $F(x)=0$ 的解。式中 ε_s 和 ε_r 为预先设置的绝对误差和相对误差值

当满足条件

(1) $J(x)$ 连续且 $J(x)$ 的变化有界;

(2) $J(x)$ 是非奇异的。

则可证明, 当初始解 x^0 足够接近最终解时, 牛顿-拉夫森法必然收敛, 且收敛速度为 $\varepsilon_{k+1} \leq C\varepsilon_k^2$, 即有两次方的收敛速度。这里 $\varepsilon_k = |x^k - x^*|$ 表示第 k 次迭代后近似解与最终解的偏离。

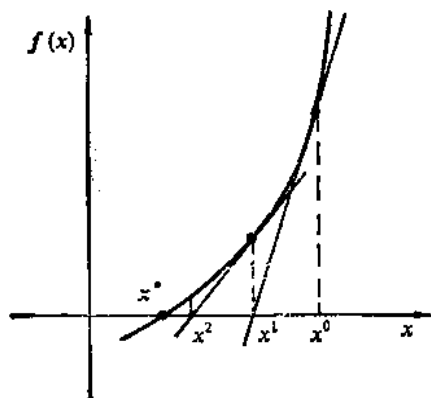


图 I-1 牛顿-拉夫森法的几何解释

牛顿-拉夫森法的几何解释如图 I-1 所示, 图中 x^* 为最终解。

所谓在 x^0 处对 $f(x)$ 作线性化, 就是在 x^0 处作一切线代替原来的非线性曲线。该线性化方程的解为 x^1 , 然后在 x^1 处再作一切线得一新的线性化方程, 其解为 x^2 。如此继续, 则解 x^{k+1} 比 x^k 会逐渐逼近 x^* 。

二、阻尼牛顿法

这是为了改进牛顿-拉夫森法的收敛性而提出来的。

现让

$$\omega_k = x^{k+1} - x^k \quad (I-25)$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{Q}_k \quad (1-26)$$

选择 δ , 使其满足充分下降条件 (Sufficient Decrease Condition)

$$1 - \frac{\|F(x^{k+1})\|}{\|F(x^k)\|} > \delta t_k \quad (1-27)$$

现介绍一种选择 t_k 的算法如下:

初始的 t_0 通常选接近 1 的数, 譬如说 0.99。按公式 (I-26) 可计算出 x^1 , 并判断这时是否满足充分下降条件。如果满足公式 (I-27), 则可求得下一步的试探值 t_1 , 计算公式如下:

$$t_{k+1} = \frac{t_k}{t_k + (1-t_k)\|F(\mathbf{x}^{k+1})\|/5\|F(\mathbf{x}^k)\|} \quad (1-28)$$

这时可反复利用上列公式,使 t_k 更接近于 1。

如果 $\mathbf{x}^1$ 不满足充分下降条件, 则根据上一次已获成功的 t_{k-1} (如 $k=0$ 时, 则必须改用另一个 t_0) 按下式作一系列的减小步骤

$$\delta_k^j = \delta_{k-1} (\delta \| \mathbf{x}^k \| / \| \omega_k \|)^{j^2/4^2} \quad (\text{I-29})$$

直到 t_j^k 满足充分下降条件为止。这里 $j=1, 2, \dots, l$ 。 l 为一整数, 通常取其为 9。

I-3 线性代数方程的求解

一、直接法

1. 高斯消去法

如有线性代数方程组 $Ax=b$, 高斯消去法是通过正消把 A 矩阵逐次变成上三角矩阵, 再通过反向代入法求出未知量 x 。

如方程组有下列形式

[illegible]

现利用第一个方程式将方程组中第二到第 n 个方程式中的 x_1 项消去, 其方法是将这些方程式减去第一方程式与 a_{1i}/a_{11} 的乘积 ($i=2, \cdots, n$)

$$\begin{aligned} & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ -) & \frac{a_{21}}{a_{11}}a_{11}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 + \cdots + \frac{a_{21}}{a_{11}}a_{1n}x_n = b_1 \frac{a_{21}}{a_{11}} \quad (I-31) \\ \hline & \quad \quad \quad a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \end{aligned}$$

其中

$$\left. \begin{aligned} a_{22}^{(2)} &= a_{22}^{(1)} - \frac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{12}^{(1)} \\ a_{23}^{(2)} &= a_{23}^{(1)} - \frac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{13}^{(1)} \\ a_{2n}^{(2)} &= a_{2n}^{(1)} - \frac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{1n}^{(1)} \\ b_2^{(2)} &= b_2^{(1)} - \frac{a_{21}^{(1)}}{a_{11}^{(1)}} b_1^{(1)} \end{aligned} \right\} \quad (1-32)$$

再利用得到的方程式将方程组中第三个到第 n 个方程式中的 x_2 项消去。如此重复以上步骤，原始方程组就转化为 $Ux=d$ ，其中 U 为上三角矩阵。

如用矩阵形式表示消去过程则有

$$\begin{aligned} & \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & b_3^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right] \rightarrow \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array} \right] \\ \rightarrow & \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} & b_n^{(3)} \end{array} \right] \rightarrow \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{array} \right] \end{aligned}$$

矩阵中各元素上标(1)表示原始值，(2)，(3)， $\cdots$ ，(n)表示第2，3， $\cdots$ ， n 次消去后的值。

因而正消算法可总结如下：

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)} \quad (1-33)$$

其中 $k=1, 2, \cdots, n$; $i=k+1, \cdots, n$; $j=k+1, \cdots, n$ 。

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} b_k^{(k)} \quad (1-34)$$

这里 $a_{kk}^{(k)}$ 是每一步消去过程中矩阵主对角线元素，通常称为主元。正消过程的必要条件是主元 $a_{kk}^{(k)} \neq 0$ ，否则会导致求解失败。但只要系数矩阵 A 不是奇异的，仍可通过改变行列次序，使正消过程进行下去。

上三角矩阵的反向代入算法为：

$$\begin{cases} x_n = b_n^{(n)} / a_{nn}^{(n)} \\ x_k = \left(b_k^{(k)} - \sum_{j=k+1}^n a_{kj}^{(k)} x_j \right) / a_{kk}^{(k)} \quad k=n-1, \cdots, 1 \end{cases} \quad (1-35)$$

2. LU 分解法

这种方法是将 A 矩阵变成上三角矩阵 U 和下三角矩阵 L 相乘的形式

$$LUx=b \quad (I-36)$$

或写成

二、迭代法（或称逐步逼近法）

对于大的线性方程组，特别是系数组成的矩阵有很多零元素时，一般不用直接法，因为它要占用大量存储单元。这时改用迭代法，此法不必存储系数矩阵中的零元素，同时计算程序也比较简单。

1. 高斯-雅可比法

此法是将 A 矩阵分解为 L 、 D 和 U 矩阵。这里 L 为下三角矩阵， D 为对角线元素矩阵， U 为上三角矩阵。其形式有

$$U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

(注: 这里的上、下三角矩阵与前面讨论的 L 、 U 矩阵的不同点是所有的对角线元素都为零。)

將 $A=L+D+U$ 代入公式 $Ax=b$ 得

$$(L + D + U)x = b$$

移项后有

$$D\mathbf{x} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x} \quad (1-43)$$

其迭代公式为

$$D\mathbf{x}^{k+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^k \quad (\text{I-44})$$

高斯-雅可比法可以如下理解。设有一组线性方程组

[illegible]

设系数矩阵非奇异, 上式可改写为

[illegible]

现选择一组初始解 $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)}$, 并代入式 (I-30), 得到新的迭代解

$$\left\{ \begin{aligned} x_1^{(1)} &= \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \dots - a_{1n}x_n^{(0)}) \\ x_2^{(1)} &= \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)} - \dots - a_{2n}x_n^{(0)}) \\ x_3^{(1)} &= \frac{1}{a_{33}} (b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)} - \dots - a_{3n}x_n^{(0)}) \\ &\vdots \\ x_n^{(1)} &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1^{(0)} - a_{n2}x_2^{(0)} - \dots - a_{n,n-1}x_{n-1}^{(0)}) \end{aligned} \right. \quad (\text{I}-46)$$

用新解 $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots, x_n^{(1)}$ 再代入公式 (I-45) 得 $x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \dots, x_n^{(2)}$ 。以此类推，
当

$$M^{(k+1)} = \max |x_i^{(k+1)} - x_i^{(k)}| < \varepsilon \quad (I-47)$$

时认为收敛，停止迭代。

这样，可得到一般计算式如下，

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right] \quad (I-48)$$

2. 高斯-塞得尔 (Gauss-Seidel) 法

此法同样将 A 矩阵分解为 L 、 D 和 U 矩阵，并将 $A=L+D+U$ 代入公式 $Ax=b$ 得

$$(L + D + U)x = b$$

移项后有

$$(L + D)x = b - Ux \quad (I-49)$$

其迭代公式为

$$(L + D)x^{k+1} = b - Ux^k \quad (I-50)$$

高斯-塞得尔法的特点分析如下。同样有一组线性方程组如 (I-30) 式, 在系数矩阵非奇异时得 (I-45) 式。现选择一组初始解 $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)}$ 代入第一式得

$$x_1^{(1)} = \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \dots - a_{1n}x_n^{(0)})$$

计算 $x_1^{(1)}$ 时, 立即使用已计算得到的新值 $x_1^{(1)}$, 即

$$x_2^{(1)} = \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)} - \dots - a_{2n}x_n^{(0)})$$

同样, 计算 $x_3^{(1)}$ 时, 立即使用已计算得到的新值 $x_2^{(1)}$, 即

$$x_3^{(1)} = \frac{1}{a_{33}} (b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)} - \dots - a_{3n}x_n^{(0)})$$

如此类推, 完成一次迭代, 得到 $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots, x_n^{(1)}$ 。然后开始新的循环。

其特点是: 每一个新计算得到的 x 值立即被用到下一式的计算中, 而不是象高斯-雅可比法那样在下一循环中再使用本循环中的计算结果。

这样, 其一般计算式为

$$\begin{aligned} x_i^{(k+1)} &= \frac{1}{a_{ii}} [b_i - a_{i1}x_1^{(k+1)} - \dots - a_{i,i-1}x_{i-1}^{(k+1)} \\ &\quad - a_{i,i+1}x_{i+1}^{(k)} - \dots - a_{in}x_n^{(k)}] \\ &= \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right] \end{aligned} \quad (\text{I-51})$$

三、逐次超松弛法 (Successive Over Relaxation)

此法是在校正量上乘一加速因子 ω (或称松弛因子) 来促使迭代过程加快。

$$x^{k+1} = \omega x_G^{k+1} + (1 - \omega)x^k \quad (\text{I-52})$$

其中 x_G 为高斯-塞得尔迭代结果。其矩阵形式为

$$(D + \omega L)x^{(k+1)} = [(1 - \omega)D - \omega U]x^{(k)} + \omega b \quad (\text{I-53})$$

ω 的值一般为 $0 < \omega < 2$ 。当 $\omega = 1$ 时, 逐次超松弛法就简化为高斯-塞得尔法。

选取最佳松弛因子相当困难。实际上常采用试算法, 即从同一个初始向量出发, 选取不同的松弛因子并迭代相同次数, 然后比较残余向量

$$r^{(k+1)} = b - Ax^{(k+1)}$$

抛弃那些有较大残余向量 $|r^{(k+1)}|$ 的松弛因子, 而选用得到较小残余向量的 ω 作为松弛因子。

附录Ⅱ CIF 格式

CIF 格式提供一组基本图素——多边形、矩形、圆和线条，用以描述集成电路芯片中不同层次上的两维图形。

CIF 的每个语句包含一个关键字或字母，接着是参数，最后用分号(;)结束。参数之间必需用空格隔开。对每行的语句数没有限制，对任何字段的具体列号也没有限制。可以在任何地方插入注释，只要把注释的内容放在圆括号内。

CIF 语句分为两类，一类为几何类，一类为控制类。

几何类语句有：

LAYER (L)	设置掩膜层
BOX (B)	矩形
WIRE (W)	线条
ROUNDFLASH (R)	圆
POLYGON (P)	多边形
CALL (C)	调用子程序

控制类语句有：

DS	子程序定义的开始
DF	子程序定义的结束
DD	删除子程序定义
0~9	用户指定的信息
(" ")	注释
END	CIF 文件终止

现具体讨论各语句的作用和语法：

1. LAYER 为几何图形设置掩膜层。LAYER 后面以单一的层名作为参数。例如，
L NC

即把该语句后面的所有几何图形都设置在 NMOS 的接触孔层，直至出现下一个 LAYER 语句时为止。

对 NMOS 和 CMOS 工艺，CIF 中的层次名见表Ⅱ-1。

2. BOX 它通过长度、宽度、中心位置和方向（可选项）来描述一个矩形。其格式如下：

B length width xpos ypos [rotation]

对于图Ⅱ-1中的矩形，其 CIF 表示法为

B 25 60 80 40 -20 -20

在 CIF 中，方向不用角度来度量，而用“方向矢量”来表示。“方向矢量”有两个坐标，第一个整数是“方向矢量”沿 x 轴的分量，第二个整数是沿 y 轴的分量。“方向矢量”的幅度是无意义的。例如(10,5)定义了一个 30° 的逆时针转动，(10, -10)将是顺时针转

表 I-1 CIF 中的层名

NMOS		CMOS	
NM	金属层	CMF	金属1层
NP	多晶硅层	CMS	金属2层
ND	扩散层	CPG	多晶硅层
NC	接触孔层	CAA	有源区层
NI	注入层	CSG	选择区层
NB	隐埋层	CWG	阱区层
NG	钝化层	CC	接触孔层
		CVA	通孔层
		COG	钝化层

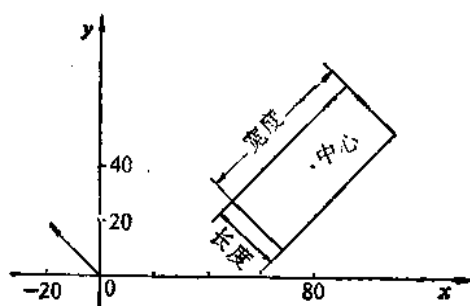


图 I-1 BOX 在 CIF 中的表示法

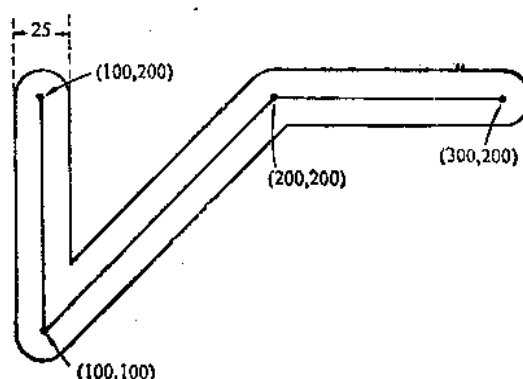


图 I-2 线条在 CIF 中的表示法

动 45° 。这样可避免采用三角函数，也避免了角度的度量单位如何选择的问题。

3. WIRE 它用来描述线条。WIRE 后面的参数为宽度以及任意数目的坐标点。线条具有一定的宽度，端部为圆角。例如

W 25 100 200 100 100 200 200 300 200

表示图 I-2 中的图形。

4. ROUNDFLASH 它通过直径和圆心坐标来描述一个圆。其格式为

R diameter center-coordinate

例如

R 20 30 40

将在(30,40)圆心处画一半径为 10 的圆。

5. POLYGON 它通过一系列的坐标点来描述一个封闭多边形。由于是封闭的，第一点的坐标必然与最后一点的坐标相重合，因而在语句中不必重复。多边形可以有任意的复杂性，包括凹形和自相交。

对于图 I-3 中的多边形，其 CIF 表示法为

P 150 100 200 200 200 300 100 300 100 200

6. CALL 调用其它语句的集合。该集合由 DS 开始，DF 结束。所有的子程序在定义时给予一个号，在调用时，以这个号来识别。

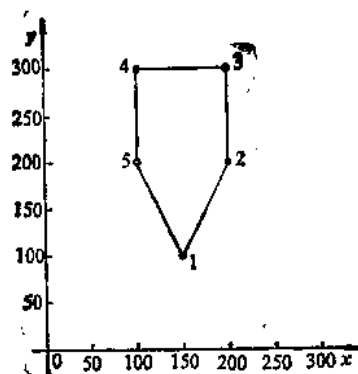


图 1-3 任意多边形

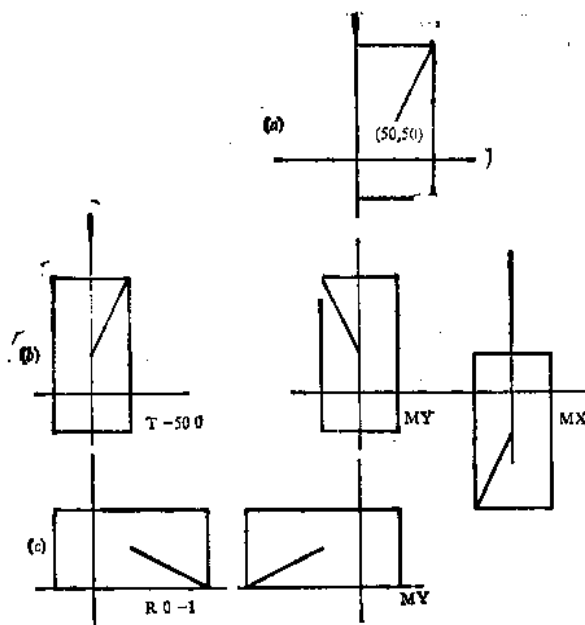


图 1-4 CALL 中的转换作用

例如在某子程序中有一 LAYER 语句和一 BOX 语句,并定义该子程序为 4。现有 C4 语句时,将在该层上画出一矩形。

除了简单的调用外,还可以加上转换信息,以改变子程序内图形的几何位置。转换信息包括移动、转动和镜向对称。

移动 T x y (offset 量)

转动 R x y (方向矢量终点坐标)

镜向 MX 对 X 轴作镜向对称

MY 对 Y 轴作镜向对称

现举例如下:

如有 Subroutine 10; BOX 100 200 50 50; 示于图 1-4(a)。现有语句 C 10 T -50 0 MY MX; 其转换结果示于图 1-4(b)。如有语句 C 10 R 0 -1 MY; 则转换结果示于图 1-4(c)。

7. DS 和 DF 定义子程序的开始和结束。DS 的变量是子程序号和比例因子。对 DF 无变量。比例因子包括一个分子和一个分母,通过它对子程序中的图形进行放大或缩小。子程序的定义可有以下例子:

DS 10 2 20;

B 10 20 5 5;

W 1 5 5 10 15;

DF;

CIF 子程序允许嵌套调用。如子程序在调用之前已定义过,则允许向前引用。例如:

DS 10;

DF,
DS 11,
...

DF,
C 10,

以上次序是合法的。而以下次序则是非法的。

C 11,
DS 11,
...
DF,

8. DD 用于删除定义。它有一参数。若子程序的序号等于或大于该参数值, 则该子程序被删除。这个语句对于合并多个 CIF 文件很有用。

9. 0~9 CIF 可以用数字 0~9 加以扩展。虽然它还不是 CIF 的正式的规定, 但已有某种确定的习惯。这些扩展见表 I-2。

表 I-2 CIF 中的扩展

语 句	扩 展 功 能
0 x y layer N name;	在规定的层和位置上设置有名节点
0V x1 y1 x2 y2 ...xn yn;	画矢量
2A "msg" T x y;	在规定的位上方加上“信息”
2B "msg" T x y;	在规定的位下方加上“信息”
2C "msg" T x y;	在规定的中央加上“信息”
2L "msg" T x y;	在规定的左边加上“信息”
2R "msg" T x y;	在规定的右边加上“信息”
cellname;	说明单元名称
994 label x y;	在规定的位上加上标号

10. END (E) 是 CIF 文件的最后一个语句, 它表示文件的结束。它不包含分号。

参 考 文 献

- (1) R. Rice, "VLSI, the Coming Revolution in Application and Design", Compcon Spring '80, pp. 19—20, Feb. 1980
- (2) J. E. Solomon, "Computer-Based Design for Tomorrow's Super Chip", ISSCC '86 Keynote Address, pp. 54—55, Feb. 1986
- (3) J. D. Williams, "STICKS—A Graphical Compiler for High Level LSI Design", Proc. National Computer Conference, pp. 289—295, 1978
- (4) N. Weste, "Virtual Grid Symbolic Layout", Proc. 18th DA Conference, pp. 225—233, June 1981
- (5) F. R. Ramsay, "Automation of Design for Uncommitted Logic Arrays", Proc. 17th DA Conference

- rence, pp.100—107, June 1980
- [6] C.Tanada, S.Murai, H.Tsuji, "An Integrated Computer Aided Design System for Gate Array Masterslice; part 2 the Layout Design System MARS-M3", Proc.18th DA Conference, pp.812—819, June 1981
 - [7] T.Matsuda, T.Fujita, "LAMBDA: A Quick Low Cost Layout Design System for Masterslice LSI's", Proc.19th DA Conference, pp.802—808, June 1982
 - [8] G.Persky, D.N.Deutsch, D.G.Schweikent, "LTX——A Minicomputer-Based System for Automated LSI Layout", Journal of Design Automation and Fault Tolerant Computing, Vol. 1, No.3, pp.217—255, May 1977
 - [9] T.Adachi, H.Kitazawa, M.Nagatani, T.Sudo, "Hierarchical Top-down Layout Design Method for VLSI Chip", Proc. 19th DA Conference, pp.789—791, June 1982
 - [10] S.Horiguchi, H.Yoshimura, R.Kasai, T. Sudo "An Automatically Designed 32b CMOS VLSI Processor", ISSCC'82, pp.54—55, 1982
 - [11] K.Kani, H.Kawanishi, A.Kishimoto, "ROBIN: A Building Block LSI Routing Program", IEEE Proc. ISCAS, pp.658—661, April 1976
 - [12] S.Kang, W.M.vanCleemput, "Automatic PLA Synthesis from a DDL-P Description", Proc. 18th DA Conference, pp.391—397, June 1981
 - [13] G.D.Hachtel, A.R.Newton, A.L.Sangiovanni-Vincentelli, "Techniques for Programmable Logic Array Folding", Proc.19th DA Conference, pp.147—155, June 1982
 - [14] R.Beresford, "Comparing Gate Arrays and Standard-Cell ICs", VLSI Design, 7:8, pp. 30—38, Dec.1983
 - [15] R.Kasai, K.Fukami, K.Tansho, "An Integrated Modular and Standard Cell IC Design Method", ISSCC'84, pp.12—13, 1984
 - [16] M.H.Lang, P.E.McCormick, "Hierarchical Design Methodologies: A VLSI Necessity", Advances in CAD for VLSI, Vol.6, Chapter 4, Elsevier Science Publishers B.V., Netherlands, 1986
 - [17] D.L.Johannsen, "Bristle Blocks: A Silicon Compiler", Proc. 16th DA Conference, pp.310—313, 1979
 - [18] D.L. Johannsen, "Silicon Compilation", Ph.D. Dissertation, Technical Report 4530:81, California Institute of Technology
 - [19] P.Denyer, D.Renshaw, N.Bergmann, "A Silicon Compiler for VLSI Signal Processors", Proc. of 1982 European Solid State Circuit Conference
 - [20] R.A.Saleh, J.E.Kleckner, A.R.Newton, "Iterated Timing Analysis and SPLICE 1", ICCAD' 83, pp.139—140, Sept.1983
 - [21] P.L.Flake, P.R.Moorby, G.Musgrave, "An Algebra for Logic Strength Simulation", Proc. 20th DA Conference, pp.615—615, June 1983
 - [22] G.F.Pfister, "The Yorktown Simulation Engine: Introduction", Proc.19th DA Conference, pp.51—54, June 1982
 - [23] T.Kondo, et al., "A Large Scale Cellular Array Processor: AAP-1", Proc.Computer Science Conference, pp.100—111, March 1985
 - [24] S.P.Fan, et al., "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits", Proc.ISCAS, p.700, 1977
 - [25] L.W.Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Memo No.

- ERL-M520, May 1975, University of California, Berkeley
- [26] A. Vladimirescu, S. Liu, "The Simulation of MOS Integrated Circuits Using SPICE2", Memo No. ERL-M80/7, Feb. 1980, University of California, Berkeley
 - [27] E. Khalili, "TECAP: An Automated Characterization System", Technical Report, March 1979, Stanford University
 - [28] 辻本順一 "MOSトランジスタ・モデル自動合わせにみ方法" 昭和59年度電子通信学会総合全国大会, 日本
 - [29] Z.P. Yu, W.J. Zhao, Z.L. Yang, Y.E. Lien, "A Novel Algorithm for Improving Convergence Behavior of Circuit Simulation", Proc. 26th DA Conference, pp. 626—629, June 1989
 - [30] H.K. Gummel, "A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations", IEEE Trans. on Electron Devices Vol. ED-11, pp. 455—465, 1964
 - [31] D.L. Scharfetter, H.K. Gummel, "Large-Signal Analysis of a Silicon Read Diode Oscillator", IEEE Trans. on Electron Devices, Vol. ED-16, pp. 64—77, 1969
 - [32] E.M. Buturla, P.E. Cottrell, "Simulation of Semiconductor Transport Using Coupled and Decoupled Solution Techniques", Solid State Electronics, 23, p. 331, 1980
 - [33] P.A. Markowich, C.A. Ringhofer, E. Langer, S. Selberherr, "A Singularly Perturbed Boundary Value Problem Modelling a Semiconductor Device", IEEE Trans. on Electron Devices, Vol. ED-30, p. 1165, 1983
 - [34] Z.P. Yu, R.W. Dutton, "SEDAN I — A Generalized Electronic Material Device Analysis Program", Technical Report, July 1985, Stanford University
 - [35] M.R. Pinto, C.S. Rafferty, R.W. Dutton, "PISCES I: Poisson and Continuity Equation Solver", Technical Report, Sept. 1984, Stanford University
 - [36] C.P. Ho, S.E. Hansen, P.M. Fahey, "SUPREM II — A Program for Integrated Circuit, Process Modeling and Simulation", Technical Report, July 1984, Stanford University
 - [37] B.W. Lindsay, B.T. Preas, "Design Rule Checking and Analysis of IC Mask Design", Proc. 13th DA Conference, pp. 301—308, June 1976
 - [38] T. Mitsuhashi, T. Chiba, M. Takashima, "An Integrated Mask Artwork Analysis System", Proc. 17th DA Conference, pp. 277—284, June 1980
 - [39] M. Takashima, T. Mitsuhashi, T. Chiba, "Program for Verifying Circuit Connectivity of MOS/LSI Mask Artwork", Proc. 19th DA Conference, pp. 544—550, June 1982
 - [40] E. Barke, "A Layout Verification System for Analog Bipolar Integrated Circuits", Proc. 20th DA Conference, pp. 353—359, June 1983