

申请上海交通大学硕士专业学位论文

启发式符号排序改进符号化模拟电路仿真器性能<sup>1</sup>

**学 校：** 上海交通大学

**院 系：** 微电子学院

**班 级：** Z0721091 班

**学 号：** 1072109062

**工程硕士生：** 王婷

**工程领域：** 微电子学

**导 师 I：** 施国勇（教授）

上海交通大学电子微电子学院

2009 年 11 月

---

<sup>1</sup> 此研究由上海市浦江人才基金（项目编号 07pj14053）和国家自然科学基金（项目编号 60876089）资助

**A Dissertation Submitted to Shanghai Jiao Tong University for  
Master Degree of Science**

**A Heuristic Algorithm to Improve  
the Performance of a Symbolic Analog Circuit  
Simulator**

**Author:** Wang, Ting

**Specialty:** Microelectronics

**Advisor I :** Prof. SHI, Guoyong

School of Microelectronics  
Shanghai Jiao Tong University  
Shanghai, P.R.China

November 18, 2009

---

# 启发式符号排序改进符号化模拟电路仿真器性能

## 摘要

基于二分决策图(BDD)的符号化仿真器性能很大程度上依赖于符号处理排序。本文对一个基于拓扑的符号化模拟电路仿真器的符号处理排序进行了研究，并提出根据电路功能模块对符号处理顺序进行重新排序的启发式排序方法。测试表明，相对于随机符号处理顺序，该启发式符号排序方法得具有更快仿真速度。

第一章介绍符号化仿真器的发展历史和符号化仿真器的潜在优势；第二章介绍了一种全新的基于拓扑方法的符号化仿真器的符号化分析原理以及二分判定图的基本概念，并介绍该仿真器中与符号处理顺序相关的有的、已有的符号排序优化策略；第三章介绍基本模拟电路知识，以及启发式排序算法中具体概念；第四章详细论述启发式排序算法的实现细节和符号处理顺序对仿真器性能影响的实验设计方法；第五章总结了符号排序实验中的重要实验数据，该数据是启发式排序算法的基本依据。

**关键词：**符号化，模拟电路，仿真器，启发式排序，算法

## **A Heuristic Algorithm to Improve a Symbolic Analog Circuit Simulator**

### **ABSTRACT**

Performance of the symbolic simulator of analog circuit which is based on the binary decision diagram is largely sensitive to ordering of symbols. In this paper, the effect of ordering to a topology-based symbolic analog circuit simulator has been studied and a heuristic algorithm is proposed, which can sort variables according to the three basic amplifier structures. For some circuits, testing results show that the performance by using the heuristic symbol sorting algorithm is much improved than the randomly-sorting method.

The first chapter introduces the history of symbolic circuit simulator and its superiority over numerical simulator. Chapter II firstly introduces BDD-based symbolic analysis of large linear analog circuits, and then presents the theory of topology-based circuit analysis. Chapter III presents some fundamental knowledge of analog circuits. The fourth chapter describes data structure of the heuristic ordering algorithm based on the topology of circuit blocks, and presents an experiment in order to find out the best symbolic sequential ordering. The fifth chapter describes the realization of heuristic ordering algorithm.

**Keywords:** symbolic, analog circuit, simulator, heuristic, algorithm

## 目 录

摘 要.....	I
ABSTRACT.....	II
第一章 绪论.....	9
1.1 符号化电路分析的历史.....	10
1.2 符号化分析方法的定义.....	11
1.3 符号化分析的基本方法与分类.....	12
1.4 与数值分析法相比, 符号化电路仿真器的潜在优势.....	14
1.4.1 数值化仿真器 SPICE 介绍.....	14
1.4.2 符号化仿真器的潜在优势.....	15
1.4.3 符号化分析和数值分析仿真器的优劣对比.....	15
1.5 基于拓扑法的符号化电路分析方法.....	16
1.6 小结.....	17
第二章 基于拓扑法的符号化模拟电路分析原理与算法.....	18
2.1 基于拓扑法的符号化模拟电路分析器的原理.....	18
2.1.1 基本前提条件.....	18
2.1.2 有向电路的图的构建规则.....	19
2.1.3 基于拓扑法的符号化模拟电路分析原理.....	21
2.1.4 举例.....	23
2.2 基于拓扑法的符号化模拟电路分析器的算法.....	24
2.2.1 二分判定图.....	24
2.2.2 符号排序对仿真器性能的影响.....	25
2.2.3 已有的启发式符号排序策略 (Heuristic Ordering).....	26
2.2.4 分离图的早期检测 (Early Disconnectivity Detection).....	28
2.3 小结.....	29
第三章 基于电路拓扑的符号排序的基本概念.....	30
3.1 介绍 BJT 和 MOS 小信号模型.....	30
3.2 定义基本端口模块.....	31
3.3 定义晶体管的共极状态.....	31
3.3.1 共射 (对应 MOS 共源) 极.....	32

3.3.2 共基 (对应MOS 共栅) 极.....	32
3.3.3 共集 (对应MOS 共漏) 极.....	32
3.3.4 共射 (对应MOS 共源) 极--射极 (对应MOS 源极) 衰减.....	33
3.3.5 共集-共射极.....	33
3.3.6 共集-共基极.....	34
3.4 定义直流偏置模块.....	34
3.4.1 简单镜像电流源.....	35
3.4.2 Beta-Helper 镜像电流源.....	35
3.4.3 带负反馈的镜像电流源.....	35
3.4.4 级联镜像电流源和 Wilson 电流源.....	36
3.5 定义功能模块.....	37
3.5.1 共射 (对应MOS 共源) 极模块.....	38
3.5.2 共基 (对应MOS 共栅) 极模块.....	38
3.5.3 共集 (对应MOS 共漏) 极模块.....	39
3.5.4 共集-共射极模块.....	40
3.5.5 共集-共基极模块.....	40
3.6 负载.....	41
3.6.1 无源负载.....	41
3.6.2 有源负载.....	41
3.7 可能不能正确识别的部分电路.....	43
3.8 小结.....	44
<b>第四章 基于电路拓扑的启发式符号排序算法的实现.....</b>	<b>45</b>
4.1 排序算法在符号化模拟电路仿真器中的作用.....	45
4.2 排序算法的对象.....	46
4.3 排序算法的数据结构.....	46
4.4 介绍能识别电路功能模块的搜索算法.....	48
4.4.1 识别镜像电流源结构.....	48
4.4.2 识别晶体管的三种共极状态.....	49
4.5 对功能模块重新排序算法设计.....	52
4.6 对未能正确检测电路结构的晶体管的算法排序设计.....	54
4.7 测试变量排序对仿真器影响的实验设计.....	55
4.7.1 Perl 脚本自动测试不同符号排序的实验基本原理.....	55
4.7.2 Perl 脚本自动测试不同符号排序的具体实现.....	56
4.7.3 变量排序实验的优点和局限性.....	58

4.8 小结 .....	59
<b>第五章 实验结果分析 .....</b>	<b>60</b>
5.1 不同拓扑结构电路的实验结果 .....	60
5.1.1 符号排序实验的主要测试电路 .....	60
5.1.2 单个晶体管模型的内部变量排序 .....	64
5.1.3 两个晶体管模型的混合变量排序 .....	66
5.1.4 两个功能模块的混合变量排序 .....	69
5.1.5 所有晶体管模型的内部变量排序 .....	72
5.1.6 晶体管与相邻电阻、电容的混合变量排序 .....	73
5.2 较大规模模拟电路的仿真结果 .....	74
5.3 小结 .....	75
<b>参 考 文 献 .....</b>	<b>76</b>
<b>致 谢 .....</b>	<b>78</b>
<b>攻读硕士学位期间已发表或录用的论文 .....</b>	<b>79</b>

## 图目录

图 1 简单的三阶 RC 电路.....	12
图 2 三阶 RC 电路的传输函数的表达式.....	12
图 3 有向图构建规则举例.....	19
图 4 理想运算放大器模型.....	20
图 5 拓扑法符号化模拟电路分析法举例.....	23
图 6 有效生成树、生成树和对应的生成项.....	24
图 7 二分判定图.....	25
图 8 简单的三阶 RC 电路.....	26
图 9 三阶 RC 电路的两个排序的 SDD 大小明显不同.....	26
图 10 简单的三阶 RC 电路的传输函数表达式.....	26
图 11 并联边预处理举例.....	27
图 12 孤立节点举例.....	28
图 13 有效边计算举例.....	28
图 14 晶体管的小信号模型.....	30
图 15 二端口、三端口和四端口元件.....	31
图 16 单双极型晶体管放大电路.....	33
图 17 单场效应管的放大电路.....	33
图 18 多双极型晶体管放大电路.....	34
图 19 双极管型简单镜像电流源电路.....	36
图 20 场效应管构成的简单镜像电流源.....	36
图 21 双极管构成的级联电流源和 Wilson 电流源.....	37
图 22 场效应管构成的级联镜像电流源和 Wilson 电流源.....	37
图 23 共射（或共源）模块.....	38
图 24 共基（或共栅）模块.....	39
图 25 共集（或共漏）模块.....	39
图 26 共集-共射极模块.....	40
图 27 共集-共基极模块.....	41
图 28 共射极（或共源极）互补型负载.....	42
图 29 共射极（或共源极）耗尽型负载.....	42
图 30 共射极（或共源极）耗尽型负载.....	43
图 31 ua725 的短路保护电路.....	43
图 32 符号化仿真器结构.....	45
图 33 单个源极-源极衰减电路.....	48
图 34 单个源极-源极衰减电路的数据结构.....	48
图 35 共源极和折叠型共源极电路.....	50
图 36 共栅极和折叠型共栅电路.....	51
图 37 共漏极和折叠型共漏极和上推下拉式电路.....	52
图 38 共射（对应 MOS 共源）极模块内部排序例子.....	53



图 39 共基（对应 MOS 共栅）极模块内部排序例子.....	54
图 40 共集-共射电路的内部排序例子.....	54
图 41 共集-共基的模块内部排序例子.....	54
图 42 基于 GRASS 的变量排序实验设计原理.....	55
图 43 脚本中存储的网表示意图.....	56
图 44 递归实现的全排列算法.....	57
图 45 全排列组合的总和.....	59
图 46 全排列组合的时间总和.....	59
图 47 ua741（包含 2 4 个 BJT 管）.....	61
图 48 ua725（包含 2 6 个 BJT 管）.....	62
图 49 三级运算放大器（包含 11MOS 管）.....	62
图 50 三级运算放大器电路图（包含 15 个 MOS 管）.....	63
图 51 双端转单端电路（包含 14 个 MOS 管）.....	63
图 52 端对端电路图（包含 22MOS 管）.....	64
图 53 单个晶体管内部最优排序和最差排序的仿真对比.....	66
图 54 CS-SD 单晶体管电路的 SDD 分布图.....	66
图 55 多晶体管电路.....	67
图 56 多晶体管模型混合排序对比图.....	68
图 57 双端转单端电路的模块化示意图.....	69
图 58 相邻的两个模块的混合排序和单个模块内部排序对比.....	70
图 59 模块 A&B 混合排序的 SDD 规模分布图.....	70
图 60 模块 A 排序的 SDD 规模分布图.....	71
图 61 模块 B 排序的 SDD 规模分布图.....	71
图 62 晶体管内部变量排序对比.....	73

## 表目录

表 1 单晶体管放大电路详细信息.....	65
表 2 单晶体管电路的最优和最差排序的仿真对比.....	65
表 3 多晶体管电路信息.....	67
表 4 多晶体管的模型混合排序的最优和最差排序的仿真对比.....	68
表 5 双端转单端的电路详细信息.....	69
表 6 双端转单端电路的两个模块混合排序的最优和最差排序的仿真对比.....	70
表 7 三级运放的电路信息.....	72
表 8 多晶体管的模型混合排序的最优和最差排序的仿真对比.....	72
表 9 补偿电容和电感与相邻晶体管排序的最优和最差排序的仿真.....	74
表 10 大规模电路的测试信息.....	74
表 11 启发式排序与随机排序仿真对比.....	75

# 第一章 绪论

集成电路技术先后经历了小规模集成(SSI: Small-Scale Integration), 中规模集成(MSI: Medium-Scale Integration), 大规模集成(LSI: Large-Scale Integration)和超大规模集成电路(VLSI: Very-Large-Scale Integration)等几个发展阶段。随着 VLSI 技术的出现, 集成电路技术进入了以“系统集成”为特征的全新时代。集成电路朝着超大规模集成(Ultra-Large-Scale Integration) 和晶片集成(Wafer-Scale Integration)的方向发展<sup>[1]</sup>。

LSI/VLSI 技术的发展推动了片上系统(Soc: System on Chip) 等大型系统级芯片的诞生。无论是消费类产品如电视、录像机, 还是通信类产品如电话、网络设备, 这些产品的核心都开始采用芯片作为它们的功能中枢。一个片上系统一般包括数字与模拟电路两部分。数字电路中的数字信号常取二值信息, 它用两个有一定数值范围的高或低电平来表示, 也可用两个不同状态的逻辑符号如“1”和“0”来表示。数字电路设计目前已基本实现设计过程的自动化, 相应的理论与技术也已经比较成熟, 有大量的 EDA (Electronic Design Automation) 工具运用于深亚微米数字电路设计; 例如可以使数字电路设计获得较好性能的逻辑和物理综合工具。然而模拟电路设计工具的发展仍然面临许多挑战, 由于模拟电路中电压、电流、频率和周期的变化相互制约的复杂性, 模拟电路的设计很大程度上依赖于设计师的经验, 市场上需要更强大的模拟集成电路方向的 EDA 工具。

在模拟电路设计验证自动化工具中, 模拟电路仿真器主要分为基于数值分析仿真器和基于符号化分析仿真器。基于加州大学伯克利分校(University of California, Berkeley)开发的数值电路仿真器 SPICE (Simulation Program with Integrated Circuit Emphasis) <sup>[2]</sup>是一种功能强大的通用模拟电路仿真器, 该仿真器被业界广泛使用。SPICE 主要用于 IC、模拟电路、数模混合电路、电源电路等电子系统的设计和仿真, 它可以分析电阻、电容、电感、互感、独立电压源、独立电流源和各种线性受控源、传输线以及有源半导体器件, 它也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。对于数值型仿真器, 一般当电路参数改变时, 需要通过重新仿真才能确定新的计算结果, 即虽然数值型仿真器可以很快地、很好的验证电路设计, 但它很难预计电路参数改变后的性能变化。为了了解各元件对电路性能的影响程度, SPICE

---

中还可以进行灵敏度分析，也称之为直流灵敏度分析，它是在直流工作点附近将所有的元件线性化后，计算各元件参数值变化时对电路性能影响的灵敏程度。通过对电路进行灵敏度分析，可以预先知道电路中各个元件对电路性能影响的重要程度。但是对于一个电路而言，计算所有元件的灵敏度往往是没有必要的，而且通过数值计算的方法来计算电路中所有元件之间的灵敏度也是不可能的。符号化的分析方法可以比较好的解决这个问题，电路级的符号化分析方法是一种通过电路自变量（时间与频率）、因变量（电压和电流）以及符号化的电路元件来计算电路行为特性的方法。最终分析的结果是电路的传输函数的符号化解析表达式，该表达式包含了丰富的电路行为和性能的参数。另外，符号化模拟电路仿真器在设计空间探索、最佳拓扑结构选择、可测试分析以及错误诊断、生成电路行为模型、加快迭代技术等方面具有更大的优势<sup>[3]</sup>。

随着 ASIC 设计规模的不断扩大、复杂度不断增加，同时，日益激烈的竞争促使 IC 设计厂商努力缩短电子产品进入市场的时间（Time-to-Market）。模拟电路的漫长设计周期及由工艺参数变化（variation）所带来的不稳定越来越不能满足集成电路系统的设计需求，如何提高验证的效率已成为一个巨大的挑战。目前数字集成电路设计过程在大量成熟的 EDA 工具的辅助下已基本实现自动化，而模拟电路设计还是主要依赖于经验丰富的模拟电路设计工程师，很多关键的参数的确定需要经过漫长的仿真，其主要原因是缺乏像数字设计中成熟的、强大的 EDA 辅助工具。为了能追上数字设计的步伐，为了降低模拟设计从系统定义到流片的时间与成本代价，需要功能全面和强大的电路仿真工具，而符号化的分析方法及其相应的仿真工具可以清晰地揭示电路的行为特性，也可以和其他的技术如电路的尺寸调节、可测性分析等结合使用，因此基于符号化的仿真工具将成为提高模拟电路设计能力的必不可少的辅助手段。

## 1.1 符号化电路分析的历史

基于符号的电路分析方法由来已久。上世纪 60 年代末及 70 年代初，符号化电路分析方法逐渐成为研究的热点，并提出了许多基于计算机的分析技术；由于当时计算机的处理能力较低，而且规模较大的电路的生成项个数呈指数倍增加，因此当时电路的符号化分析方法并未在工业界得到广泛应用。电路仿真器大致上分为三类：全符号化仿真器、纯数字化仿真器以及符号数值仿真器。基于图形的分析方法（例如生成树枚举法和信号流图法）被认为是最适合分析小规模全符号化电路；若仅仅将频率看作符号，则可以采用数值分析方法来分析较大规模的电路；结合上述两种方法的优点，则产生了符号数值混合分析的方法，即将部分关键的电路参数符号化，从而能较快的

---

分析较大规模的电路。随着 SPICE 数值仿真器的出现，其准确和快速地数值计算能力在工业界得到广泛应用，同时符号化电路分析不能处理较大规模的电路而逐渐被 SPICE 取代。

上个世纪 80 年代末，为了克服符号化分析只能处理较小规模的电路的情况，诞生了层次化分解和符号简化的分析方法。层次化分解方法的优点是不再通过展平整个电路来获得电路符号表达式，而是嵌套地产生该表达式。符号简化分析法是在符号化分析的过程中根据符号的阶数和频率来删除一些对分析结果影响非常小的信息。同时基于矩阵行列式的符号分析法得到发展，该方法像基于图形的符号化分析方法一样也可以对整体符号化的电路进行有效的分析。在 80 年代末期，随着计算机处理能力的不断提高以及集成模拟电路分析的需求，基于符号的电路分析方法才重新得到重视并成为研究的热点，诞生了许多成功的符号化仿真器，如 ISAAC<sup>[5][6]</sup>，ASAP<sup>[7][8]</sup>，SYNAP<sup>[9][10]</sup>，SAPEC<sup>[11]</sup>，SSPICE<sup>[12]</sup>，SCYMBAL<sup>[13]</sup>，SCAPP<sup>[14]</sup>和 GASCAP<sup>[15]</sup>，它们主要分析模拟集成电路和数字电路符号化布尔分析仿真器。在这些成功的 EDA 仿真工具中，基于行列式的分析法以及信号流图的分析法被公认为是最灵活和最有效的工具。

推动符号化电路分析方法发展的主要原因有两个，首先是计算机性能的大幅度提高以及相关的高效算法的发展，其次是模拟集成电路设计过程需要更强大和更有效的计算机辅助设计工具以及集成电路设计自动化的发展。

## 1.2 符号化分析方法的定义

符号化分析方法是计算电路行为特性的一种形式化的方法，电路中独立的自变量包括时间和频率，因变量包括电压、电流以及以符号形式存在的电路元件。符号化分析方法而且符号化分析方法是定性分析（定性分析是指给出电压和电流变化的趋势，例如增加、减小和不变）；它和数字化分析方法（变量和电路中元件都是以数字的形式存在）相互补充。

符号化分析方法主要研究线性电路的频域特性。对一个集总（lumped）的线性时不变电路，如果已知输入，则相应输出的传输函数可以表示成两个多项式相除的形式，其频域自变量为  $x$ （ $s$  对应连续时域自变量， $z$  对应离散时间自变量），电路中元件用符号表示（而非具体的数值），如下式：

$$H(x) = \frac{N(x, p_1, p_2, \dots, p_m)}{D(x, p_1, p_2, \dots, p_m)} = \frac{\sum_i x^i a_i(p_1, p_2, \dots, p_m)}{\sum_i x^i b_i(p_1, p_2, \dots, p_m)} \quad (1-1)$$

其中  $x$  是一个复数形式的频域变量，对应连续时域电路（continuous-time circuit）的  $s$  变量或离散时域电路（discrete-time circuit）的  $z$  变量； $p_1, p_2, \dots, p_m$  是电路元件符号； $a_i(p_1, p_2, \dots, p_m)$  和  $b_i(p_1, p_2, \dots, p_m)$  分别是由电路符号组成的多项式。对于完全符号化的电路（电路中所有的元件都用符号表示），多项式系数完全由电路元件的符号组成；对于符号化和数值型混合电路，多项式系数由两部分组成，分别是电路元件符号和具体数值。下面是一个电路的例子（图 1），符号化的电路仿真器可以得到如下的传输函数（图 2）：

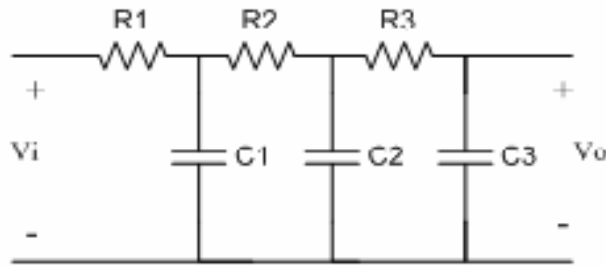


图 1 简单的三阶 RC 电路

Fig. 1 A three-order RC circuit.

$$H(s) = \frac{\frac{1}{R1} \frac{1}{R2} \frac{1}{R3}}{\frac{1}{R1} \frac{1}{R2} \frac{1}{R3} + \frac{1}{R1} \frac{1}{R2} C_3 + \frac{1}{R1} C_2 \frac{1}{R3} + \frac{1}{R1} C_2 C_3 + \frac{1}{R1} \frac{1}{R3} C_3 + C_1 \frac{1}{R2} \frac{1}{R3} + C_1 \frac{1}{R2} C_3 + C_1 C_2 \frac{1}{R3} + C_1 C_2 C_3 + C_1 \frac{1}{R3} C_3 + \frac{1}{R2} C_2 \frac{1}{R3} + \frac{1}{R2} C_2 C_3 + \frac{1}{R2} \frac{1}{R3} C_3}$$

图 2 三阶 RC 电路的传输函数的表达式

Fig. 2 Transfer functions of three-order circuit.

电路中如果出现非线性元件或运算放大器，则应当先将非线性元件用相应的小信号模拟替代，此举将非线性电路转换成线性电路，然后才能用该符号化仿真器对电路进行分析。

### 1.3 符号化分析的基本方法与分类

符号化分析方法一般以描述电路的网表（netlist）作为输入（例如著名的 SPICE

---

语法文件)。由于符号化分析方法只能处理线性电路,如果电路中出现非线性元件(如二极管、场效应管和运算放大器等),则应先将非线性元件转换成线性的小信号等效模型(电容、电感、电阻和线性放大器等组成),其中电容  $C$  和电感  $L$  要表示成频域上的阻抗或导纳的形式( $C_s$  或  $L_s$ )。经过上述转换后,即可以进行相应的分析运算。

符号化分析方法大致可以分为两类:代数方法和拓扑方法。代数方法需要以符号参数为自变量,构建出描述线性电路的一系列方程组,对该方程进行求解从而获得最终的运算结果(例如通过计算电路行列式的代数余子式的方法来求解方程组);代数方法首先需要构建电路的方程组,并通过一系列的代数操作获得所需要的电路传输函数。拓扑方法一般不需构建电路方程,可以将整个电路看作一张图或一个网络,并且这个网络中每个分支节点有不同的权重;通过对这张图进行操作而获得符号化的传输函数的解析表达式(例如电路中基本回路法或生成树枚举法)<sup>[3]</sup>。

目前具体的符号化电路分析法只能处理集总(lumped)、线性的(linear)、时不变(time-invariant)的电路,其具体算法可以分为五种:

- 矩阵行列式法(matrix and determinant methods)。
- 符号流图法(signal-enumeration methods)。
- 生成树枚举法(tree-enumeration methods)。
- 参数提取法(parameter-extraction methods)。
- 数值插入法(interpolation methods)。

矩阵行列式法就是求解电路方程组,例如使用符号化的 Gauss 消去法或者递归使用 Laplace 展开法来求解电路方程组;符号流图法根据 Mason 公式,并依据明确定义的法则来寻找表征电路方程中各阶所对应的路径和回路来计算电路的结果。生成树枚举法是通过枚举电路图中的生成树,计算生成树所对应的最小生成项来得到计算结果,电路图中的每个分支都对应一个权重,生成项个数由这棵树的各分支的权重决定。参数提取法也是基于电路方程组的求解,将方程组对应的行列式进行递归分解,每次分解将行列式分解成包含所需提取参数的部分和不包含所需参数的部分,这样逐步分解最后得到所需的结果。数值插入法是基于电路某些工作点的数值分析结果,并对数值结果进行分析。

这些具体的算法都有各自的优势和劣势,矩阵行列式法(具体实现的仿真器有:ISAAC<sup>[5][6]</sup>、SYNAP<sup>[9][10]</sup>、SAPEC<sup>[11]</sup>和 SSPICE<sup>[12]</sup>)和符号流图法(具体实现的仿真器有:ASAP<sup>[7][8]</sup>、SCYMBAL<sup>[13]</sup>和 ACAPP<sup>[14]</sup>)是被公认为比较灵活和有效的方法。生成树枚举法还不能处理各种类型的受控源,并且难以避免产生可以消去的冗余项,这些冗余项会占用多余的 CPU 运行时间和内存,还会对后面的符号化近似运算产生

---

不利影响。参数提取法适用于小规模电路的符号化分析运算。数值插入法仅适用于以频率变量为符号的数值分析，该方法可以处理较大规模的电路<sup>[16]</sup>。

符号化分析算法较数值分析算法（例如 SPICE）有很多优势，但是目前数值仿真在工业界得到广泛应用。符号化分析算法的主要难度还是在于当电路结点和元件增加会导致算法的时间和空间复杂度呈指数倍增加，其分析的电路往往局限于规模较小的电路。

## 1.4 与数值分析法相比，符号化电路仿真器的潜在优势

### 1.4.1 数值化仿真器 SPICE 介绍

数值电路仿真器 SPICE（Simulation Program with Integrated Circuit Emphasis）<sup>[2]</sup>是一种功能强大的通用模拟电路仿真器，它采用修改结点分析法（MNA: Modified Nodal Analysis）。该仿真器由美国加州大学伯克利分校的计算机辅助小组利用 FORTRAN 语言开发而成，主要用于大规模集成电路的计算机辅助设计。1985 年，SPICE 被加州大学伯克利分校用 C 语言进行改写。1988 年 SPICE 被指定为美国国家工业标准。与此同时，各种以 SPICE 为核心的商业模拟软件对 SPICE 做了许多实用化工作，从而 SPICE 成为最流行的电子电路仿真软件。以下是该数值仿真器的优缺点分析<sup>[17]</sup>。

**优点：**对模拟电路进行直流、交流和瞬态等基本电路分析，实现了蒙特卡洛分析、最坏情况分析 & 优化设计等较为复杂的电路分析；可以对模拟电路、数字电路和数/模混合电路进行仿真；集成度较强，电路图绘制完成之后可直接进行电路仿真，并且可以随时观察分析与仿真结果。

**缺点：**瞬态分析（通常是利用数值积分法把非线性微分方程变成一组代数方程组，然后用高斯消去法来求解，因为这些线性方程仅仅在积分时刻点是有效的，而随着仿真器进展到下一个积分步长，积分方法必须重复来得到新的线性方程组，如果信号变化得特别快，积分步长应该取得非常小以便积分方法能收敛到正确的解，因此瞬态分析需要大量的数学操作。）是最费时的验证方法；电路无法仿真或仿真时因不能收敛而失败，特别是用于数模混合电路及脉冲电路时尤其如此。



---

## 1.4.2 符号化仿真器的潜在优势

符号化仿真器的潜在优势总结如下<sup>[3]</sup>:

- 1) 更好的揭示电路行为特性: 数值型符号仿真器可以迅速并准确的计算电路传输函数的数值并将结果进行输出; 数值型仿真器无法得知哪个元件参数对传输函数的变化的影响最大; 不得不进行大量的仿真来获得比较优异的电路性能。符号化仿真器的计算结果是电路传输函数的符号化解析表达式; 当电路参数发生改变的时候, 该解析表达式仍有效; 对传输函数进行灵敏度分析可以知道哪些元件对传输函数的影响比较大。
- 2) 电路分析模型的产生和电路尺寸的自动化调整。符号化的仿真器可以自动产生描述电路交流特性的分析模型; 电路行为特性的近似分析可为电路尺寸的自动化调整提供依据。
- 3) 避免数值仿真器在进行数值计算过程中的舍入误差以及数值计算的收敛性问题, 同时提高设计的可靠性和设计精度。
- 4) 电路结构的交互拓展。当电路拓扑结构调整后, 符号化仿真器可以快速的生成新的电路行为特性, 交互式可以方便地绘制不同的电路结构, 为电路结构的自动化调整提供依据。
- 5) 电路参数的迭代调整。如果电路结构不变, 符号化分析得到的传输函数的解析表达式是不会改变的。模拟电路设计中经常通过多次电路参数的迭代从而获得理想的性能; 针对符号化仿真器, 只需要将参数值代入电路表达式进行求值运算, 避免了重复跑仿真的复杂过程。
- 6) 自动生成运算放大器、滤波器等模拟电路模块的行为模型。搭建模块的时候不需要考虑具体的器件, 为快快速进行电路系统的行为分析提供便利。
- 7) 辅助模拟可测性分析和故障检测。

## 1.4.3 符号化分析和数值分析仿真器的优劣对比

符号化分析和数值分析仿真器分别有各自的优势, 在电路仿真方面有不同的应用。一般可以采用符号仿真和数值仿真相结合的方式。以下列出了符号化分析和数值分析仿真器的优缺点:

- 数值型仿真器的优势:
  - 1) 数值型仿真器提供描述电路特性的数据或网表。

- 
- 2) 拥有成熟的数值分析理论。
  - 3) 数值分析法的仿真速度快。
  - 4) 数值分析主要用于验证电路的功能特性, 检验电路是否符合设计的需求。
- 数值型仿真器的劣势:
    - 1) 数值仿真无法直观的揭示电路行为, 因为单纯的数值仿真结果无法知道某个电路元件对电路哪个元件参数对电路特性的影响比较大, 往往必需通过多次实验以及设计经验来找到关键元件。
    - 2) 数值分析不能提供智能化设计。数值分析仿真器不能提供形式化的结果, 只能提供精准的数值结果, 无法提供相关的智能化分析的设计修改建议, 并提示电路设计中存在的问题。
  - 符号化分析法的优势: 与数值型仿真器相比, 符号化仿真器提供的是全新的电路仿真结果。符号化仿真器的优势很明显 (其潜在优势在 1.4.2 章节中进行了详细的论述)。
  - 符号化仿真器的劣势<sup>[3]</sup>:
    - 1) 多年来, 符号化仿真器只能处理集总 (lumped)、线性的、时不变和离散时间域 (开关电容) 的模拟电路进行频域分析。
    - 2) 对于非线性的电路 (例如包含 MOS 场效应晶体管和 Bipolar 双极型晶体管), 这些非线性元器件在直流工作点附近可以近似看作是线性的元件, 并可以用小信号等效模型进行替换。因此可以计算电路传输函数 (Transfer Function)、共模抑制比 (CMMB: Common Mode Rejection Ratio) 电路、电源纹波抑制比 (Power Supply Rejection Ratio)、阻抗 (Impedances) 和噪声 (Noise) 的符号化解析表达式。
    - 3) 目前提出的符号化分析算法的效率较低, 当电路结点和元件增加会导致算法的时间和空间复杂度呈指数倍增加。
    - 4) 符号化分析方法的分析结果是符号化的解析表达式, 该表达式是由一系列的符号化的生成项组成。对于符号化仿真器来说, 这些生成项的数目随着电路尺寸 (电路元件个数) 的增长而呈指数倍增长。符号化模拟电路仿真器可处理的电路规模受到限制。

## 1.5 基于拓扑法的符号化电路分析方法

本论文将着重讨论一个基于拓扑分析法的符号化仿真器的性能优化。目前, 基于

---

符号流图的符号化分析方法被公认为最灵活和最有效的基于拓扑结构的符号化电路分析方法,但是该方法任然不能处理较大规模的电路<sup>[3]</sup>。诞生了全新的拓扑分析方法,即组成电路表达式的生成项与电路网络的生成树有直接联系,同时对某些特定电路元件的枚举方法进行了论证<sup>[18][19]</sup>。该方法的局限主要有:该方法只能处理特定的受控源,并不能处理所有受控源和理想运算放大器;没有提出比较好的计算机自动化分析算法。根据前面提到的全新的拓扑分析方法,<sup>[20][21]</sup>提出了基于二分判定图(BDD: Binary Decision Diagram)的实现机制,提出了清晰地生成树枚举理论,并定义了完整的、相应的图约化法则,并用递归遍历的方式确定生成项的正负符号。该仿真器可以处理较为一般的线性电路,并且对较大规模电路进行频域分析。该仿真器的实现是基于二分判定图,则该仿真器必须在仿真过程之前给定变量处理顺序,但是不同的便利处理顺序对仿真器的性能影响较大,比较糟糕的变量处理顺序会浪费大量的仿真时间并占有较多内存,从很大程度上限制了该仿真器可以处理电路的规模。为了让该符号化模拟电路仿真器可以更快地、更好地处理更大规模的电路,重新排序读入的模拟电路的网表并重新定义符号处理顺序就非常有意义。

## 1.6 小结

由上述的比较可以看出,符号化分析法相对数值仿真器有很多优势。其缺点是在当前的计算机处理能力的情况下,符号化分析法缺少理想的、高效的分析算法;符号化只能处理较小的电路规模。数值分析仿真器拥有成熟的理论,算法执行效率高,同时可以处理大规模电路。但是其本身的特点局限了其对模拟电路设计提供自动化支持,不能自动进行电路优化和优化电路的关键元件。因此,研究并对优化符号化仿真器,提高符号化模拟电路仿真器能够处理电路的规模,为模拟电路设计者提供更便捷、更智能的辅助工具。

本文的研究是基于<sup>[20][21]</sup>中实现的符号化模拟电路仿真器(GRASS: Graphic Reduction Analysis Symbolic Simulator),下面第二章首先介绍该仿真器的工作原理和仿真器固有的优化策略;第三章介绍启发式排序算法中概念;第四章介绍基于拓扑结构的启发式变量排序算法和变量排序实验设计,该实验为排序算法提供了理论依据;第五章介绍启发式变量排序算法的优化效果以及实验数据结果;第六章总结启发式排序算法对仿真器的贡献,以及未来排序算法可以深入研究的问题。

---

## 第二章 基于拓扑法的符号化模拟电路分析原理与算法

本章主要介绍一种符号化模拟电路仿真器 GRASS (Graph Reduction Analysis Symbolic Simulator) <sup>[21]</sup>, 它采用了全新的符号化分析的方法, 即对电路图几乎不作任何改变的情况下, 直接从电路拓扑结构计算得到电路的传输函数。该符号化分析方法尤其适合对电路拓扑进行分析的模拟电路。

### 2.1 基于拓扑法的符号化模拟电路分析原理

首先讨论基于电路拓扑的符号化模拟电路分析定理。<sup>[22][23]</sup>提出了基于电路拓扑的全新的符号化分析的方法, 可以处理一般线性有源电路 (general linear active circuits); 其操作的对象是电路拓扑结构图 (topological graph) 和符号参数 (parameter symbols), 无需建立电路方程组, 直接对电路的拓扑结构进行操作从而求得电路传输函数; 电路求解的过程就是对有效生成树和有效生成树对进行操作, 只需要计算有效生成树对的系数。分析方法处理电路的过程不存在生成项损失, 其解是精确解。因此, 该于拓扑法的符号化模拟电路分析法能适用于一般模拟放大电路, 而且其效率远远高于树枚举分析法。

本文所介绍的基于电路拓扑的符号化分析方法主要应用线性电路的分析; 在符号化分析前, 应当将电路转换成相应的有向图, 再利用相应的分析定理对电路求解。

#### 2.1.1 基本前提条件

- 1) 只能处理 5 类元件: 阻抗 ( $Z$ )、导纳 ( $Y$ )、四种类型的受控源 (压控电压源 VCCS、流控电流源 CCCS、压控电流源 VCCS 和流控电压源 CCVS), 独立源 (dependent sources) 和理想运算放大器 (nullors)。电路中不能包含非独立源;
- 2) 电路中只能含有一个独立源。由于线性电路具有叠加性, 则多独立源的结果可以通过简单的相加实现。因此, 当电路中含有多个独立源时, 需对不同的独立源分别进行符号化分析, 然后将结果叠加获得即可;

- 3) 每个受控源只能控制一个受控源;
- 4) 每个受控源只能受一个控制源的控制;
- 5) 电路中不含反馈电路。

对于符合 2.1.1 节上述条件的电路, 则可以依据以下规则将其转换为有向图。

### 2.1.2 有向电路的图的构建规则

电路的网表一般是无向、闭合的网表。电路图在分析前必须先转化成一个有向图, 该步骤按照以下规则进行。图 3 是电路转化的实际例子, 其从无向图 3 (a) 转换成图 3 (b) 的例子。

- 1) 受控源或独立源转化成有向边。电压源的方向就是沿着正极 (+) 到负极 (-), 电流源的方向和电流的方向保持一致;
- 2) 在有向图中, 对应的电压控制支路 (VC edge: controlling voltage branch) 添加一条有向边, 例如图 3 (b) 中的  $U_3$ ; 对应电流控制支路添加一个结点和一条单独的有向边, 例如图 3 (b) 中的  $I_1$ ;
- 3) 有向图中每条边对应于电路中的某个元件, 其权重为对应电路元件的名称;
- 4) 理想运算放大器用零器 (Nullor) 建模, 每个零器 (Nullor) 由一对零阻器 (Nullator) 和一个泛阻器 (Norator) 组成, 如图 3;
- 5) 输入输出端口 (I/O) 被抽象成一个受控源。

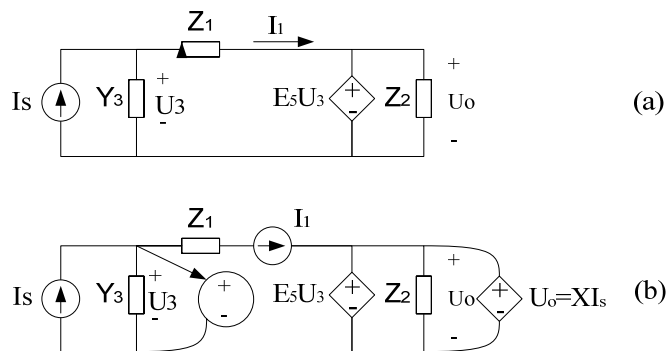


图 3 有向图构建规则举例  
(a) 待分析电路; (b) 转换后的电路

Fig. 3 A circuit example.

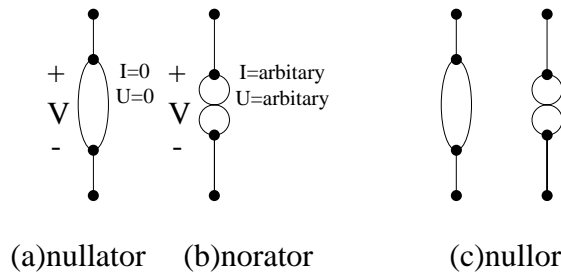


图 4 理想运算放大器模型

(a) 零阻器; (b) 泛阻器; (c) 零器

Fig. 4 An example of the ideal OP-AMPS model.

上述的有向图转化的规则 5 是非常重要的转换。对于线性电路来说，当电路的拓扑结构不变的时候，电路的输出完全取决于输入，即不同的输入值会有不同的输出值。电路的传输函数如公式 2-1 所示：

$$H(s) = \frac{\text{Output}}{\text{Input}} \quad (2-1)$$

(2-1) 可以写作：

$$H(s) * \text{Input} = \text{Output} \quad (2-2)$$

可以看出，线性电路的输出受输入控制。因此将输入输出抽象建模成一个受控源，其增益为电路传输函数，即需要的符号化分析结果。同样也可以看做输入受输出控制，其增益也就是传输函数的倒数。通常，流过理想电压源的电流为 0；理想电流源两端的电压应该为 0。

对于受控源而言，有如下规定：

- 1) 压控电压源 (VCVS):  $U_i = E_{ij} U_j, I_j = 0;$
- 2) 流控电流源 (CCCS):  $I_i = F_{ij} I_j, U_j = 0;$
- 3) 压控电流源 (VCCS):  $I_i = G_{ij} U_j, I_j = 0;$
- 4) 流控电压源 (CCVS):  $U_i = E_{ij} I_j, U_j = 0;$

该符号化分析电路对控制源有特别的要求。即对于要求流过电压控制源的电流为 0；同时电流源的两端电压为 0。基于该仿真器对受控源的特殊要求，同时线性电路的输入也可以看做受输入的控制，因此该符号化分析方法将输出作为控制电源，同时输入被视作理想电源，此举导致受控源（输入）就是传输函数的倒数。例如图 3 所示，其电路的输入输出看作一个压控电流源 (VCCS)，即输出时独立源，其输入信号是受控源（符合特定条件的理想受控源），且其增益为 X，

$$I_i = X * U_o \quad (2-3)$$

对于线性电路，(2-3) 式也可写作

$$H(s) = \frac{U_o}{I_i} = \frac{1}{X}。 \quad (2-4)$$

### 2.1.3 基于拓扑法的符号化模拟电路分析原理

基于电路拓扑的全新的符号化分析的方法的提出<sup>[22][23]</sup>，该方法可以处理一般线性有源电路（general linear active circuits）。符号化分析方法只能分析符合特定条件的电路。同时，对电路进行拓扑分析之前，也应当将电路转化成有向图。基于拓扑法的符号化模拟电路分析定理操作的对象是有向图。该定理基于电路的拓扑结构，通过枚举有向图的生成树来获得电路传输函数的符号化表达式的生成项。

需要注意的是，某些生成树没有有效的生成项。<sup>[21]</sup>本节的定理将详细定义有效生成树，以及有生成树获得传输函数的生成项，并如何得到正确的分析结果。

定理一（有效生成树定理）：一棵有效生成树不包含 Nullor，也不包含 VC（controlling voltage branch）和 CS（controlled current branch）器件的对应边，但是必须包含所有的 CC（controlling current branch）和 VS（controlled voltage branch）器件的对应边。

定理二（生成树对应生成项定理）：有效生成树对应的生成项由该生成树的分支对应的元件符号的乘积组成。其中所有的阻抗和导纳直接出现在生成项中；所有的 CC 和 VS 的权重为 1，且均不在乘积项中出现。

定理二中介绍了如何从生成树得到生成项。同时，阻抗器件在生成树中对应的生成项中以倒数的形式出现，即将阻抗改写为导纳的形式。

定理三（有效生成树定理）：有效生成树包含有效左生成树和有效右生成树。

- 1) 所有的电阻（Z）和导纳（Y）元件必须同时出现在有效左、右生成树中，或在左、右有效生成树中均不出现；
- 2) 所有的零阻器（NU）和泛阻器（NO）边必须出现在每一对有效生成树对中，其中零阻器存在于有效右生成树中，而泛阻器存在于有效左生成树中；
- 3) 所有的 CC（controlling current branch）和 VS（controlled voltage branch）必须出现在每一对有效生成树对中；CC 和 VS 边也可以成对出现在有效左、右生成树对中。所谓的成对出现是指 VS 的边出现在有效左子树中，其对应边 CC 出现在右有效生成子树中；CC 的边出现在有效右生成树中，其对应边

VS 出现在右有效生成左有效生成子树中；

- 4) 所有的 VC 和 CS 边可以出现在有效生成树中，也可以不出现。所谓的成对出现是指 CS 的边出现在有效左子树中，其对应边 VC 出现在右有效生成子树中；VC 的边出现在有效右生成树中，其对应边 CS 出现在右有效生成左有效左生成子树中；

上述定理定义了有效生成树对的规则，下面举例说明。其中 (a, b) 表示 a 在有效左生成树中，b 在有效右生成树中。例如一个流控电压源 (CCVS) 元件，在有向图中对应两条有向边，有三种可能的生成树选择 (CC, CC)、(VS, VS) 和 (CC, VS)。有定理三 (3) 可知，CCVS 元件只能出现 (CC, CC) 和 (VS, VS) 组合，也就是 CC、VS 边同时出现在左有效生成树；或 (VS, CC) 组合，也就是 VS 边出现在左生成树，CC 出现在右生成树中。

推论一 (生成树边类型推论)：在一个有效生成树对中，左生成树只含有导纳、阻抗、VS (controlled voltage branch)、CS (controlled current branch)、CC (controlling current branch) 和 NO (不含 VC 和 NU)；右生成树只含有导纳、阻抗、VS (controlled voltage branch)、VC (controlling voltage branch)、CC (controlling current branch) 和 NU (不含 VC 和 NU 边)。

定理四 (生成树对应生成项定理)：由生成树得到的生成项由以下四部分的乘积组成：

- 1) 生成项符号 (+或-);
- 2) 所有出现的导纳和阻抗器件符号 (阻抗器件在最终以导纳的形式出现);
- 3) 带符号的受控源增益。

对于 CC 和 VS，其在左、右生成树中均出现，且边的取值为 1，故不影响生成项。生成项的符号有定理五确定；带符号的受控源增益由定理六确定。

定理五 (受控源符号定理)：四种受控源 (VCVS, CCCS, VCCS, CCVS) 在生成项中不仅表示为符号化的增益，其各自的符号也不同，具体如下：

$$\begin{aligned}
 VCVS &\leftrightarrow -E_{i,j}; \\
 CCCS &\leftrightarrow +F_{i,j}; \\
 VCCS &\leftrightarrow +G_{i,j}; \\
 CCVS &\leftrightarrow -H_{i,j};
 \end{aligned}
 \tag{2-5}$$

值得注意，当 VS 边出现的时候，增益前就要带上符号。

定理六 (生成树对应生成项符号定理)：假设  $A_L$  和  $A_R$  为有效生成树对中表示左



生成树和右生成树的约会入射矩阵 (Reduced Incidence Matrix) <sup>[3]</sup>。A<sub>L</sub> 和 A<sub>R</sub> 的行数相同 (等于电路结点数减 1); 它们的列数 (等于生成树边数, 即结点数减 1) 也相同, 并且按照如下的规则排列: 若 Y、Z、CC 和 VS 边分别同时出现在左、右生成树中, 则代表它们的列以相同的列数分别出现在 A<sub>L</sub> 和 A<sub>R</sub> 中; 若两条边成对出现, 则它们分别出现在 A<sub>L</sub> 和 A<sub>R</sub> 中, 且对应列数相同 (控制边在 A<sub>R</sub> 中, 受控边在 A<sub>L</sub> 中)。入射矩阵中所有的边均是有向的, 其对应电路图中的有向边 (源器件对应边 VC、CS、VS、CC), 其方向在矩阵中仍然保持不变; 对应无向边 (Y、Z、NU、NO) 则选择任意一个方向, 但需要保证这些元件在 A<sub>L</sub> 和 A<sub>R</sub> 中方向保持一致。

定理七 (生成项的零和定理): 所有生成项之和为零。

### 2.1.4 举例

通过一个简单的例子说明基于拓扑法的符号化模拟电路分析定理。对图 5 中的电路, 有电压源、电阻和电容构成了高通滤波器, 求电容两端的电压值。由 kirchoff 定理可以很快求出频域传输函数  $H(s) = \frac{RCS}{1 + RCS}$ 。

由上文提出的符号化电路分析方法, 将图 5 (a) 转化成相应的有向图 5 (b)。将输入输出看做压控电压源 (VCCS), 输出控制输入。

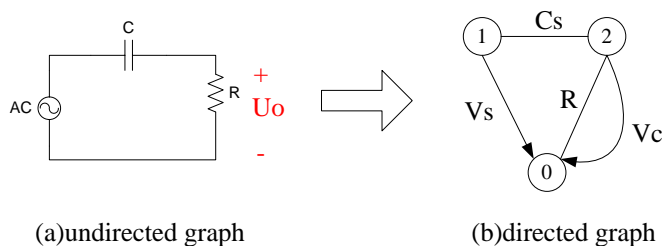


图 5 拓扑法符号化模拟电路分析法举例  
(a) 待分析电路; (b) 电路有向图

Fig. 5 An example of the enumeration rules.

然后根据定理一、三, 可以得到图 5 中所示的两棵有效生成树和一对有效生成对。它们的生成项分别为 Cs\*1, (1/R)\*1 和 -Cs\*X。

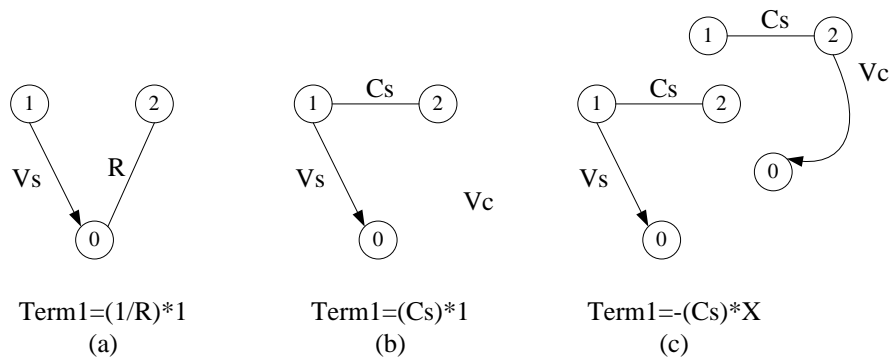


图 6 有效生成树、生成树和对应的生成项

(a) 有效生成树 1; (b) 有效生成树 2; (c) 有效生成树对

Fig. 6 Admissible tree, admissible tree pair and corresponding product terms.

由定理七，联立此三项得到等式  $Cs * 1 + \frac{1}{R} * 1 - Cs * X = 0$ 。

解方程得到传输函数  $H(s) = \frac{1}{X} = \frac{RCS}{1 + RCS}$ ，与 kirchoff 定理的计算结果一致。

## 2.2 基于拓扑法的符号化模拟电路分析器的算法

2.1 节介绍了基于电路拓扑结构的符号化电路分析定理，并计算出电路传输函数的符号化表达式。一般在对电路的求解采用枚举生成树，检验它们是否有效，但是，一般生成项的个数与电路的规模呈指数关系。当电路规模变大（电路中包含更多的晶体管和其它元件），需要大量的运算时间和存储空间才能得到正确的结果，这也是目前符号化仿真器面临的主要问题。

本章介绍的全新的符号化模拟电路仿真器引入二分判定图（Binary Decision Diagram）<sup>[4]</sup>，它是一种非常重要的数据结构，其优点是可以将某些需要复杂计算和存储的问题转化为近似线性的结果。它的缺点是同一个电路的不同的变量处理顺序会导致仿真器的运行时间不同，例如某些变量排序可能导致仿真器耗费全部的计算机内存，结果是无法得到正确的仿真结果。

### 2.2.1 二分判定图

二分判定图是最初由 Aker 提出，其目的是为了将布尔函数转换成判定图<sup>[26]</sup>。Lee 首先提出了直线化的转换程序<sup>[27]</sup>。1983 年 Byrant 提出将二分判定图算法应用于门极电路的研究领域，并提出简化有序的二分判定图（ROBDD: Reduced Ordered Binary

Decision Diagram) 的数据结构, 并提出基于这种数据结构的布尔函数操作的算法 [26][27]。下面是一个布尔函数的二分判定图:

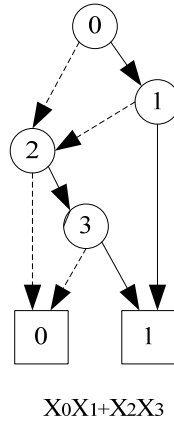


图 7 二分判定图

Fig. 7 Binary Decision Diagram.

图 7 是一张有向无环图, 有一个根结点, 有两个叶子节点 0 和 1 (方形结点)。虚线分支 (0-分支) 表示该结点的变量取值为 0 的布尔表达式的值, 实线分支表示该结点变量取 1 后的布尔表达式的值。当  $X_0=0, X_2=0$  时, 该表达式的值为 0。该图表示的布尔函数有三个变量, 该二分判定图只使用了三个结点; 若使用真值表 (truth table) 将会有 8 项 (生成项的个数和变量呈 2 的指数倍关系)。

二分判定图的优势:

- 1) 最大限度的实现判定子图共享, 否则判定子图的表达式与问题规模呈指数倍增长关系。
- 2) 简化有序的二分判定图 (ROBDD) 是一个规范的数据结构。即对同一个电路, 当采用有序的、简化的二分判定图, 其图的结构是唯一的。
- 3) 所有的布尔操作可以有简单的图论算法实现, 其中算法的复杂度和问题规模呈多项式关系。

目前, 二分判定图在数字集成电路计算机辅助设计工具中被广泛使用, 被广泛应用于电路行为验证 (behavioral simulation)、测试向量生成 (atpg: automatic test pattern generation)、故障仿真 (fault detection) 和逻辑综合 (logic synthesis)。

### 2.2.2 符号排序对仿真器性能的影响

该符号化模拟电路仿真器引入二分判定图 (Binary Decision Diagram), 在构建二

分判定图之前，需要首先给定一个变量处理顺序。该二分判定图对变量排列顺序特别敏感，即同一个电路的不同的变量处理顺序会导致仿真器的运行时间不同。如果 SDD 中结点较多，会消耗较多的计算机内存，需要更多的时间对树进行遍历，同时也会影响后续的仿真器的数值仿真时间。

图 8 是简单的三阶 RC 电路，图 9 是不同变量排序对应的 SDD (Symbolic Decision Diagram)。图 10 是该三阶 RC 电路对应的电路传输函数的符号化解析表达式。从图 9 可以看出，符号排序不同确实会导致 SDD 的大小显著不同。当处理更大规模电路时，符号排序将更严重地影响 SDD 的规模。

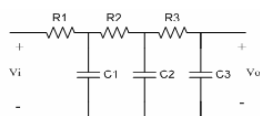


图 8 简单的三阶 RC 电路

Fig. 8 A three-order RC circuit.

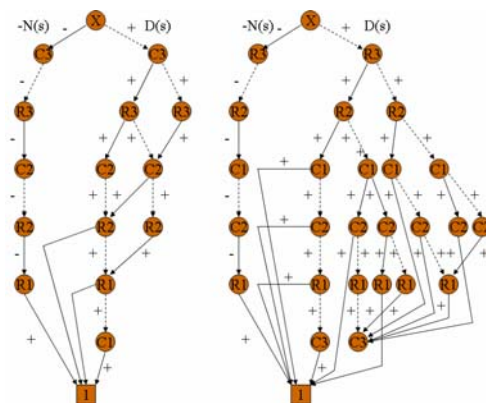


图 9 三阶 RC 电路的两个排序的 SDD 大小明显不同

Fig. 9 A contrast of two different symbol ordering.

$$H(s) = \frac{\frac{1}{R1} \frac{1}{R2} \frac{1}{R3}}{\frac{1}{R1} \frac{1}{R2} \frac{1}{R3} + \frac{1}{R1} \frac{1}{R2} C_3 + \frac{1}{R1} C_2 \frac{1}{R3} + \frac{1}{R1} C_2 C_3 + \frac{1}{R1} \frac{1}{R3} C_3 + C_1 \frac{1}{R2} \frac{1}{R3} + C_1 \frac{1}{R2} C_3 + C_1 C_2 \frac{1}{R3} + C_1 C_2 C_3 + C_1 \frac{1}{R3} C_3 + \frac{1}{R2} C_2 \frac{1}{R3} + \frac{1}{R2} C_2 C_3 + \frac{1}{R2} \frac{1}{R3} C_3}$$

图 10 简单的三阶 RC 电路的传输函数表达式

Fig. 10 Transfer function of three-order RC circuit.

### 2.2.3 已有的启发式符号排序策略 (Heuristic Ordering)

该仿真器在 2.2.2 中提到，在符号化分析前，需要首先为符号设定一个符号处理顺序。事实上，符号处理顺序在很大程度上将影响判定图的规模，这是所有的判定图

相关算法会面临的问题。

该符号化仿真器已有的启发式排序算法：

- 1) 输入输出受控源的符号总排在第一。2.1.2 章节提到，对于线性电路来说，当电路的拓扑结构不变的时候，电路的输出完全取决于输入，即不同的输入值会有不同的输出值。同样也可以看作输入受输出控制，其增益也就是传输函数的倒数。因此，将输入输出受控源的符号（**X**）排在首位是合理的优化策略，如式 2-6 中变量 **X** 是抽象的输入输出受控源的符号。

$$X = \frac{1}{H(s)} = \frac{Input}{Output} \quad (2-6)$$

这样处理的目的是建立二分判定图后，所有含有未知量 **X** 的生成项将由根结点的 1-分支（1-edge）引导，所有不含有未知量 **X** 的生成项具有 0-分支（0-edge）引导，这样未知量可以通过分别计算根结点两个分支所引导的判定图得到（传输函数等于未知量倒数的负值）。该优化策略使得分子和分母的最小生成项也可实现子图共享。

- 2) 并联元件预处理（Lumping Parallel Branches）

电路中的并联元件在有向图中表现为一组平行边（端点相同）。并联元件预处理只考虑非受控源类器件（电阻、电容和电感）。对于这些平行边，如果其中一条出现在某棵生成树中，则一定存在其他的生成树包含其余的平行边。也就是说，所有的非受控源的平行边在有效生成树对中式等价的。因此，在构建 **SDD** 之前对并联元件进行预处理可以有效的提高仿真器的效率。将多个并联的非受控源的元件等效成一个元件，该元件的符号等于相应元件的并联值。

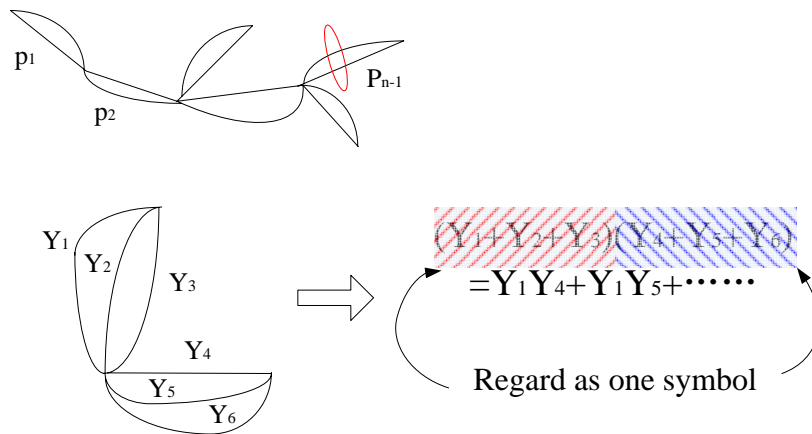


图 11 并联边预处理举例

Fig. 11 Example for Lumping Parallel Branches.

非受控源的并联边的预处理可以很大程度上减少电路中元件个数，可以很有效的提升电路的仿真器效率。

## 2.2.4 分离图的早期检测 (Early Disconnectivity Detection)

一旦某个图中出现不连通的两个分离的部分，那么这张图肯定不含有生成树。因此，在仿真器构造 SDD 的过程中检测约化子图，如果子图中含有分离的部分，该子图肯定不含有生成树，则该分支可以直接指向终结点 0，从而提前终止该分支的搜索。

图中分离部分的检测：

- 1) 出现孤立结点。当图中存在出现孤立结点，该图不可能产生生成树。图 12 的电路图中存在孤立结点。
- 2) 有效边的个数比图中节点数少 2 或者更多，其中有效边计算式将平行边的边数记为 1，自环边的边数记为 0。图 13 说明了计算有效边和结点个数，并以此判断图中是否存在孤立结点。

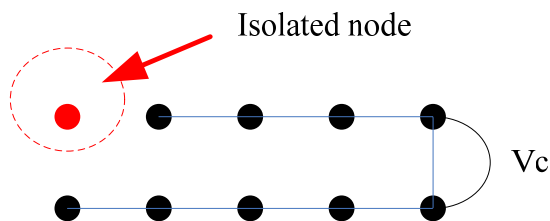
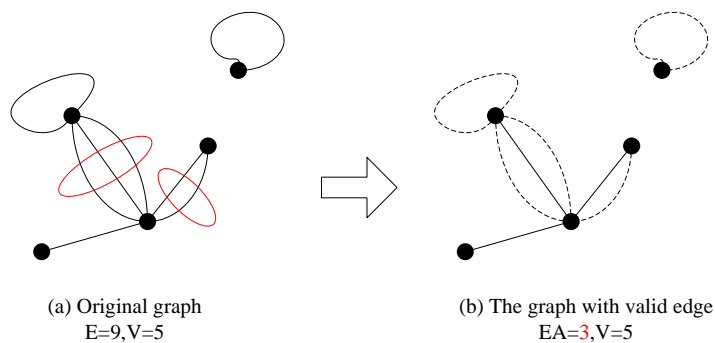


图 12 孤立节点举例  
Fig. 12 An example for isolated node.



$EA < V - 1$   
The graph is disconnected.

图 13 有效边计算举例  
Fig. 13 Example for available edge counting.

---

## 2.3 小结

本章节介绍了基于拓扑分析的符号化模拟电路仿真器的分析原理，并举例说明从电路图直接求出电路传输函数的符号表达式。该符号化仿真器引入二分判定图的数据结构，其优点是可将某些需要复杂计算和存储的问题转化为近似线性的结果。它的缺点是同一个电路的不同的变量处理顺序会导致仿真器的运行时间不同。由于该数据结构对变量排列顺序非常敏感，而比较好的变量排列顺序能很好的提高仿真器的运行效率，变量排列顺序可以有效提高仿真器效率。以下的启发式排序算法就是在对电路进行符号化分析前，重新确定仿真器的符号处理顺序。

## 第三章 基于电路拓扑的符号排序的基本概念

### 3.1 介绍 BJT 和 MOS 小信号模型

模拟放大电路中最基本的放大器件是 BJT 和 MOS 管，它是非线性器件，对含有非线性元件的电路分析不能直接运用电路理论中的公式和定律。从 BJT 的特性曲线可见，在小范围内它的非线性曲线可近似地用直线代替，此时，BJT 的电流、电压有线性关系存在，可以建立其小信号线性模型。一般模拟电路的分析包含晶体管的大信号模型 (Large-Signal Behavior) 和小信号模型 (Small-Signal Models) 分析。大信号模型主要用于计算晶体管中的总电流 (voltage)、总电压 (total voltage)、击穿电压 (Break Voltage) 等多种效应，例如二级混叠 (second-order effect)、镜像电流源的集电极电流扰动 (current-mirror variation)、Early 电压 (Early Voltage) 等关键参数需要详细分析<sup>[31]</sup>。

符号化模拟电路仿真器 (GRASS) 利用电路理论中的公式和定理对电路进行分析，它也只能分析线性元件 (详见 2.1 节)，而模拟电路包含双极型晶体管 (BJT: Bipolar Junction Transistor) 和场效应管 (MOS: Metal-Oxide-Semiconductor) 等非线性元件。因此需要将非线性元件转换成线性元件才能用 GRASS 仿真器进行分析。

在 GRASS 仿真器之前需要对电路进行预处理：首先，需要用 SPICE 仿真得到晶体管的静态工作点；根据静态工作点附近的电路参数计算出对应的小信号模型。其目的是将非线性电路元件转换成线性电路元件，一般双极型晶体管和场效应管分别采用图 14 (a) (b) 的小信号模型 (MOS 管采用 spice level 3 的小信号等效模型)。

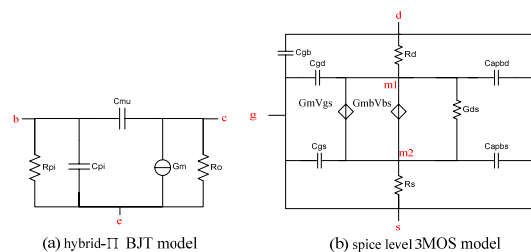


图 14 晶体管的小信号模型

Fig. 14 The small signal model for transistors.



本文提出的启发式符号排序基于电路拓扑，我们对一些常用模拟电路中典型的晶体管连接方式作分析，从中找到一些辨识规律。下面介绍电路中基本模块功能模块的定义。

### 3.2 定义基本端口模块

在处理 BJT 或 MOS 等三端口元器件的小信号模型时，这些元器件可看作一个三端口网络，此时整个电路可分解为许多二端口(电阻、电容或电感)、三端口（如图 14，hybrid- $\Pi$ 的 BJT 小信号模型有集电极、基极和发射极三个端口）或四端口网络（如图 15，spice level 3 的 MOS 小信号模型有漏极、栅极、源级和衬底四个端口），如图 14 所示。值得一提的是，当 MOS 的小信号模型的源极和衬底短路时，则此信号模型可以看作三端口元器件。

此端口模块划分的目的是区分电阻、电容和 BJT 或 MOS 的小信号模型。此举有利于变量排序算法的优化，5.2.2 的实验数据将说明晶体管的小信号模型中不允许出现非该小信号模型的电阻、电容和其它晶体管的模型模型（也有例外的情况，例如晶体管的某端口通过某个外部电阻或电容接地）。

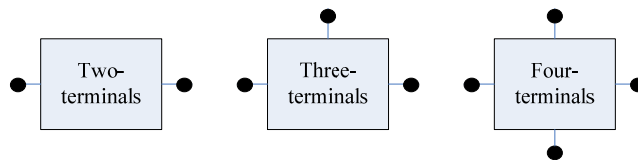


图 15 二端口、三端口和四端口元件  
Fig. 15 The complex element definition.

### 3.3 定义晶体管的共极状态

双极型晶体管（BJT）和场效应管（MOS）都有三种电路结构可以提供放大作用。在共射（对应 MOS 共源）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从集电极（对应 MOS 漏极）离开该晶体管；在共基（对应 MOS 共栅）的电路结构中，输入信号从晶体管的发射极（对应 MOS 的源极），并从集电极（对应 MOS 漏极）流出晶体管；在共集（对应 MOS 共漏）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从发射极（对应 MOS 源极）离开该晶体管。以上的电路结构的输入阻抗、输出阻抗、电压增益和电流增益各不相同。在很多情况下，

---

对多级放大器的分析可以化简为对一系列单个放大晶体管的组合的分析<sup>[31]</sup>。

根据模拟电路中对电路结构的分析，将具有放大功能的晶体管分为共射（对应 MOS 共源）、共基（对应 MOS 共栅），共集（对应 MOS 共漏）三种状态，以下详细论述三种电路的拓扑电路结构。模拟电路中 BJT 晶体管分别有共射（对应 MOS 共源）、共基（对应 MOS 共栅），共集（对应 MOS 共漏）三种应用，如图 16、17 所示。对于 BJT 晶体管电路，经常会出现共集-共基结构(CC-CB)、共集-共射结构(CC-CE)的复合型电路，如图 18 所示。通常电路结构均可分解为上述的 5 种基本电路。

许多集成电路放大器由多级放大级组成，这些放大级分别提高电源增益、电流增益、电压增益和输入到输出的传输函数。在电路分析时，可以将每个晶体管看做当个晶体管，则对整个电路的分析就化简为对一系列单个晶体管的分析。然而一些晶体管的组合经常出现，而且这些组合可以看做一个子电路的结构，并看作放大电路。随着技术的发展，这些电路结构的变化日新月异。例如，Darlington 双晶体管放大电路结构广泛应用于双极型晶体管集成电路设计；MOS 管可以提供的电流增益增益、输入阻抗却只能用于纯场效应管集成电路；另一方面，共集-共基极电路能用于 BJT 和 MOS 晶体管集成电路技术。

### 3.3.1 共射（对应 MOS 共源）极

在共射（对应 MOS 共源）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从集电极（对应 MOS 漏极）离开该晶体管。以射极（对应 MOS 源极）作为公共端，实现了电压放大又实现了电流放大，输入电阻在三种电路中居中，输出电阻较大，频带较窄。常做为低频电压放大电路的单元电路，如图 16、17（a）。

### 3.3.2 共基（对应 MOS 共栅）极

在共基（对应 MOS 共栅）的电路结构中，输入信号从晶体管的发射极（对应 MOS 的源极），并从集电极（对应 MOS 漏极）流出晶体管。以基极（对应 MOS 栅极）为公共端。只能放大电压不能放大电流，输入电阻小，电压放大倍数和输出电阻与共射电路相当，频率特性是三种接法中最好的电路。常用于宽频带放大电路，如图 16、17（b）。

### 3.3.3 共集（对应 MOS 共漏）极

在共集（对应 MOS 共漏）的电路结构中，输入信号加载在晶体管的基极（对应

MOS 栅极)，并从发射极（对应 MOS 源极）离开该晶体管。以集电极（对应 MOS 漏极）为公共端。只能放大电流不能放大电压，是三种接法中输入电阻最大、输出电压最小的电路，并具有电压跟随的特点。常用于电压放大电路的输入级和输出级，在功率放大电路中也常采用射极输出的形式，如图 16、17（c）。

### 3.3.4 共射（对应 MOS 共源）极--射极（对应 MOS 源极）衰减

在共射（对应 MOS 共源）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从集电极（对应 MOS 漏极）离开该晶体管；同时射极（对应 MOS 源极）接交流地信号。然而，电路在实际应用的时候通常会串联非零电阻接交流地信号（如图 16、17（d）所示，晶体管的射极通过串联电阻  $R_e$  再接地。该电路相对于共射级放大电路来说，其引入电阻  $R_e$ （与射极相连的衰减电阻）的负反馈。

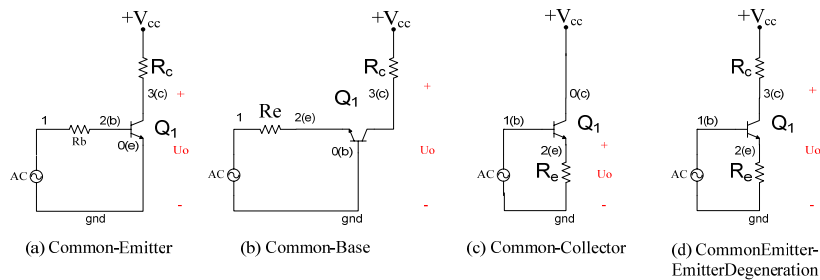


图 16 单双极型晶体管放大电路

(a) 共射极电路；(b) 共基极电路；(c) 共集极电路；(d) 共射极-射极衰减电路

Fig. 16 Amplifier stage for BJT.

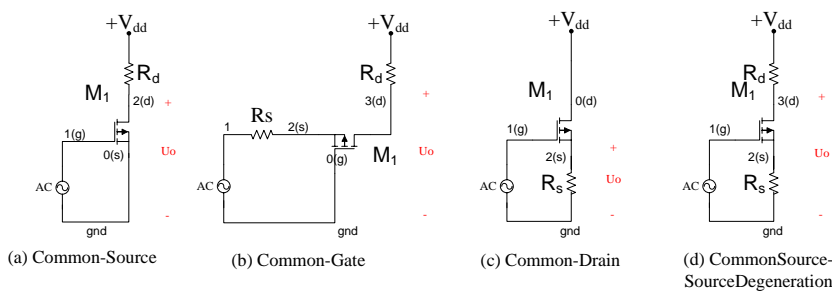


图 17 单场效应管的放大电路

(a) 共源极电路；(b) 共栅极电路；(c) 共漏极电路；(d) 共源极-源极衰减电路

Fig. 17 Amplifier stage for MOS.

### 3.3.5 共集-共射极

共集-共射电路比上述三种电路多一个晶体管，该结构可以提供更高的电流增益和更高的输入阻抗。电路结构如图 18（a）所示， $I_{bias}$  给  $Q1$  提供静态直流工作点，在某

些对电路精确性要求不要的情况下，可以用电阻来代替  $I_{bias}$  镜像电流源。

### 3.3.6 共集-共基极

共集-共基极电路首次出现是应用于真空管 (Vacuum Tube) [32][33]。真空管电路中有三个端口，分别为“阳极”(cathod)、阴极(anode)和栅极(grid)。阳极是释放电子流的地方；阴极是吸收电子流的地方；栅极在阳极和阴极之间，电子流不需通过栅极才能到阴极，该级可以控制电子的流量。电路结构如图 18 (b) 所示。

共集-共基极电路得到广泛的应用，它可以提供较高的输出阻抗，并能减小电容反馈的影响，电路可以在高频区工作。高输出阻抗的电路能提出输出电压不稳定的电压源的直流参考偏置值，同时该电路还可以提供很高的电压增益。

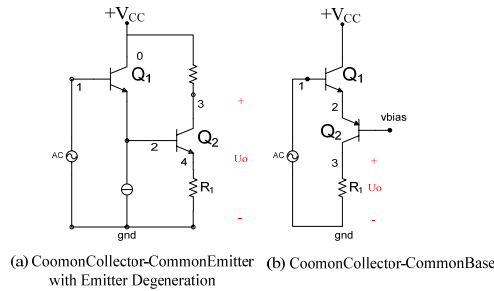


图 18 多双极型晶体管放大电路  
(a) 共集-共射极电路; (b) 共集-共基极电路

Fig. 18 Multiple-Transistor amplifier stage.

## 3.4 定义直流偏置模块

近年来由有源器件组成的镜像电流源在模拟电路中有很多的应用，它作为该结构的电路结构可以提供精准的偏置电流，同时也可以提供有源负载。镜像电流源结构的电路提供的偏置电流源对电源电压和温度非常不敏感。通常在需要提供较小的电流源偏置时，镜像电流源的电路比使用电阻更为经济实惠；在晶体管电路中使用有源负载时，镜像电流源可以提供较大的输出阻抗，同时可以得到较高的电压增益。

在模拟电子电路尤其是模拟集成电路中，广泛采用电流源作偏置电路。电流源偏置电路不仅能使电路性能具有不随温度及电源电压变化的高稳定性，而且能将基准源电流镜像成很多的电流源供电路使用，节约芯片面积功耗和简化设计。而且能有效地

---

减小电路的直流功率损耗。三极管和场效应管的输出特性均具有恒流特性，只要配以合适的外电路，使器件工作在放大区合适的工作点上，就能获得更接近理想的恒流特性。通常，为了保证某些具有放大作用的晶体管正常工作在饱和区，需要提供稳定的直流电源；一般采用高输出阻抗、高增益的镜像电流源。

### 3.4.1 简单镜像电流源

简单镜像电流源(Simple Current Mirror)可由两个参数对称的双极结型晶体管(或场效应管)基极对接而成，如图 19、20 (a)，晶体管 Q1 (或场效应管 M1) 被二极管短接，集电极和基极的电压被强制为 0。

### 3.4.2 Beta-Helper 镜像电流源

Beta-Helper 镜像电流源 (Simple Current Mirror with Beta Helper)，双极型晶体管的输出阻抗通常是有限值，图 19 (a) 所示的简单镜像电流源的输出电流很难和基准电流完全相等；为了减小镜像电流源和基准电流源的误差，可以增加一个晶体管 Q2 (如图 19 (b) 所示)，得到较为精准的镜像电流源。

场效应管的输出电阻是无穷大，如果两个场效应管的参数的微小差别而导致简单镜像电流源和基准电流不同，一般不采用 Beta-Helper 镜像电流源的结构来提高镜像电流源的精度。Beta-Helper 镜像电流源可以增加双极型晶体管和场效应管的带宽。

### 3.4.3 带负反馈的镜像电流源

如图 19 (c) 所示，双极型晶体管的射极串联负反馈电阻 (Degeneration) 也可以提高简单镜像电流源电路的输出精度。

场效应管组成的镜像电流源的源极很少串联电阻，事实上场效应管受其固有的电阻控制。因此，增加场效应管的栅极氧化层面积 (gate area) 可以增加输出电流值；增加沟道长度 (channel length) 提高其输出阻抗。保持电流和  $V_{gs}-V_t$  (栅源电压) 不变，则  $W/L$  ( $W$ : gate area,  $L$ : length of channel) 是恒定值。因此，为了增加沟道长度，且保证沟道的宽度和长度的比值不变的方法可以提高镜像电流源的输出阻抗。增加输出阻抗需要耗费更多的芯片面积。

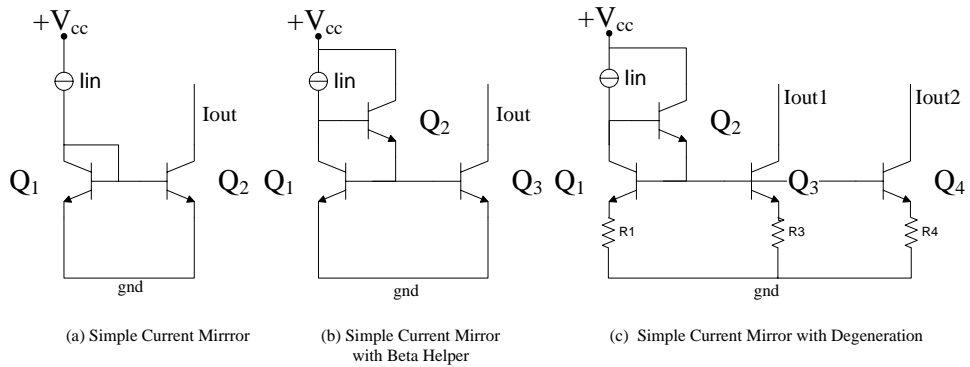


图 19 双极管型简单镜像电流源电路  
 (a) 简单镜像电流源; (b) Beta-Helper 镜像电流源; (c) 多输出电流、射极衰减的 Beta-Help 镜像电流源

Fig. 19 Simple current mirror for BJT transistors.

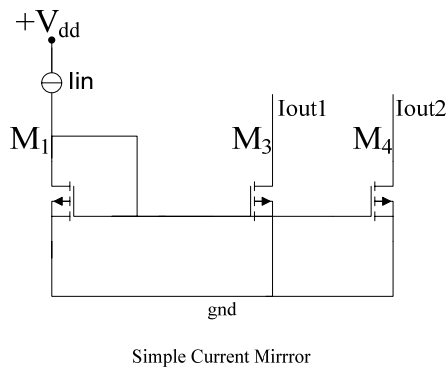


图 20 场效应管构成的简单镜像电流源  
 Fig. 20 Simple current mirror for MOS transistors.

### 3.4.4 级联镜像电流源和 Wilson 电流源

级联镜像电流源 (Cascode Current Mirror) 电路可以提供非常高的输出阻抗。镜像电流源高输出阻抗的理想特性, 因此通过级联的拓扑连接方式获得更高的输出阻抗是很自然的方法。Wilson 电流源电路不仅有较高的增益和输入阻抗, 其系统增益误差却很小 (如图 21、22 所示)。



### 3.5.1 共射（对应 MOS 共源）极模块

双极型晶体管工作于共射（对应 MOS 共源）极状态，其对应共射（对应 MOS 共源）极模块。如图 23 所示，依据输入信号、输出信号的不同，主要分为单输入单输出的共射（对应 MOS 共源）模块、双输入单输出的折叠型（folded）对称共射（对应 MOS 共源）模块和双输入双输出的折叠型（folded）共射（对应 MOS 共源）模块。

#### 1) 单输入、单输出的共射（对应 MOS 共源）模块

模块中只包括一个共射（对应 MOS 共源）极晶体管及其负载电路。

#### 2) 双输入单输出的折叠型（folded）共射（对应 MOS 共源）模块

模块中包括两个共射（对应 MOS 共源）极、射极（对应 MOS 源极）相连的晶体管及其负载电路。

#### 3) 双输入双输出的折叠型（folded）共射（对应 MOS 共源）模块

该模块和双输入单输出的结构相同，区别是从两个共射（对应 MOS 共源）极晶体管的集极（或漏极）同时引出信号。

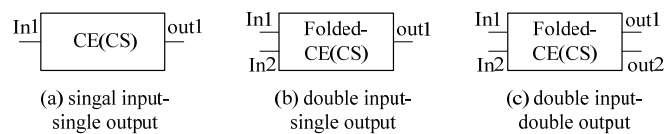


图 23 共射（或共源）模块

(a) 单输入、单输出；(b) 双输入单输出的折叠型；(c) 双输入双输出的折叠型

Fig.23 Common-Emitter (or Common-Source) module.

### 3.5.2 共基（对应 MOS 共栅）极模块

双极型晶体管工作于共基（对应 MOS 共栅）极状态，其对应共基（对应 MOS 共栅）极模块。如图 24 所示，依据输入信号、输出信号的不同，主要分为单输入单输出共基（对应 MOS 共栅）极模块、双输入单输出的折叠型共基（对应 MOS 共栅）极模块和双输入双输出的折叠型共基（对应 MOS 共栅）极模块。

#### 1) 单输入、单输出的共基（对应 MOS 共栅）极模块

模块中只包括一个共基（对应 MOS 共栅）极晶体管及其负载电路。

#### 2) 双输入单输出的折叠型共基（对应 MOS 共栅）极模块

模块中包括两个共基（对应 MOS 共栅）极、基极（对应 MOS 共栅）相连的晶体管及其负载电路。



### 3) 双输入双输出的折叠型共基（对应 MOS 共栅）极模块

该模块和双输入单输出的结构相同，区别是从两个共基（对应 MOS 共栅）极晶体管的集极（或漏极）同时引出信号。

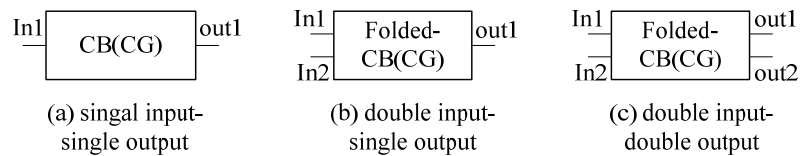


图 24 共基（或共栅）模块

(a) 单输入、单输出；(b) 双输入单输出的折叠型；(c) 双输入双输出的折叠型

Fig. 24 Common-Base (or Common-Gate) module.

### 3.5.3 共集（对应 MOS 共漏）极模块

双极型晶体管工作于共集（对应 MOS 共漏）极状态，其对应共集（对应 MOS 共漏）极模块。如图 25 所示，依据输入信号、输出信号的不同，主要分为单输入单输出的共集（对应 MOS 共漏）极模块、双输入单输出的折叠型共集（对应 MOS 共漏）极模块和双输入双输出的折叠型共集（对应 MOS 共漏）极模块。

#### 1) 单输入单输出共集（对应 MOS 共漏）极模块

模块中只包括一个共集（对应 MOS 共漏）极晶体管及其负载电路。

#### 2) 单输入单输出的对称的共集（对应 MOS 共漏）极模块

模块中只包括两个共集（对应 MOS 共漏）极晶体管及其负载电路。该模块主要指上推下拉式电路，其由两个共集（或共漏）晶体管组成，输入信号和输出信号相同。

#### 3) 双输入单输出的折叠型共集（对应 MOS 共漏）极模块

该模块中只包括两个共集（对应 MOS 共漏）极晶体管及其负载电路。

#### 4) 双输入双输出的折叠型共集（对应 MOS 共漏）极模块

该模块和双输入单输出的结构相同，区别是从两个共集极（或共漏极）晶体管的射极（或源极）同时引出信号。

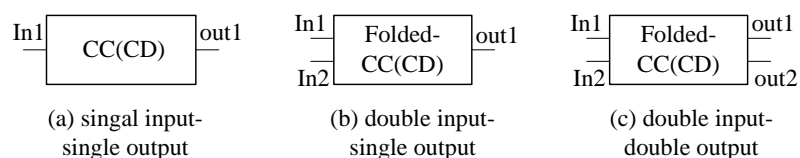


图 25 共集（或共漏）模块

(a) 单输入、单输出；(b) 双输入单输出的折叠型；(c) 双输入双输出的折叠型

Fig. 25 Common-Collect (or Common-Drain) module.

### 3.5.4 共集-共射极模块

如图 26 所示，该模块包括一对或两对双极型晶体管，每对的晶体管分别是共集极和共射极。该结构很少应用于场效应管电路。

#### 1) 单输入单输出的共集-共射极模块

该模块包含一对双极型晶体管，一个晶体管工作于共集极，另一个晶体管工作于共射极，单输入信号连接共集极晶体管的基极，单输出信号是共射极晶体管的集极。例如图 47 中的 Q16 和 Q17，此对双极型晶体管属于共集-共射极电路。

#### 2) 双输入单输出的折叠型共集-共射极模块

该模块包含两对双极型晶体管，一对晶体管工作于共集极，另一对晶体管工作于共射极，双输入信号分别连接两个共集极晶体管的基极，单输出信号连接某个共射极晶体管的集极。如图 48 所示，Q6 和 Q9 是共基极，Q7 和 Q8 是共射极，输入信号分别连接 Q6 和 Q9 的基极，输出信号为 Q8 的集极。

#### 3) 双输入双输出的折叠型共集-共射极模块

该模块和双输入单输出的结构相同，区别是从两个共射极（共源极）晶体管同时引出信号。

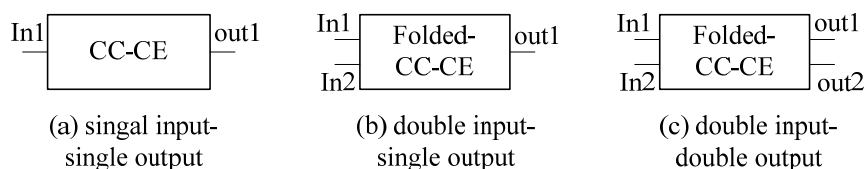


图 26 共集-共射极模块

(a) 单输入单输出；(b) 双输入单输出的折叠型；(c) 双输入双输出的折叠型

Fig. 26 CommonCollect-CommonEmitter module.

### 3.5.5 共集-共基极模块

如图 27，包括一对或两对双极型晶体管，每对的晶体管分别是共集极和共射极。该结构很少应用于场效应管电路。

#### 1) 单输入单输出的对称共集-共基极模块

该模块包含一对双极型晶体管，一个晶体管工作于共集极，另一个晶体管工作于共基极，单输入信号连接共集极晶体管的基极，单输出信号是共基极晶体管的集极。

## 2) 双输入单输出的折叠型共集-共基极模块

该模块包含两对双极型晶体管，一对晶体管工作于共集极，另一对晶体管工作于共基极，双输入信号分别连接两个共集极晶体管的基极，单输出信号连接某个共基极晶体管的集极。如图 47 所示，Q1 和 Q3 是共基极，Q2 和 Q4 是共射极，输入信号分别连接 Q1 和 Q2 的基极，输出信号为 Q4 的集极。

## 3) 双输入双输出的折叠型共集-共基极模块

该模块和双输入单输出的结构相同，区别是从两个共基（对应 MOS 共栅）极晶体管的集极（或漏极）同时引出信号。

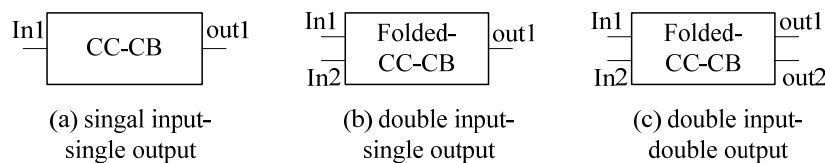


图 27 共集-共基模块

(a) 单输入单输出；(b) 双输入单输出的折叠型；(c) 双输入双输出的折叠型

Fig. 27 CommonCollector-CommonBase module.

## 3.6 负载

负载是指连接在电路中的电源两端的电子元件。电路中不应没有负载而直接把电源两极相连，此连接称为短路。常用的负载有电阻、引擎和灯泡等可消耗功率的元件。不消耗功率的元件，如电容，也可接上去，但此情况为断路。模拟电路中，依据负载是否需要额外的电源供电，分为无源负载和有源负载。

### 3.6.1 无源负载

纯电阻元件是一种无源元件，其工作不需要额外的电源供电。即纯电阻就是简单的无源负载。例如图 16、17 中纯电阻元件  $R_c$ ，就是无源负载。

### 3.6.2 有源负载

利用恒流源代替放大电路中的负载，就构成有源负载放大电路。这种放大电路不仅单级电路电压放大倍数高，还可以改善放大电路的其他性能。因此，这种有源负载放大电路已成为模拟集成电路设计特色之一。例如图 47 中的 Q13B 就是有源负载，

它由镜像电流源的输出电路构成，其特点是高增益、高阻抗。

1) 共射极（或共源极）互补型负载（complementary load）

图 28 (a) 晶体管 T1 和 T2 是共射极电路，其负载为 pnp 镜像电流有源负载；图 28 (b) 晶体管 T1 和 T2 是 MOS 共源极电路，其负载为 p 沟道 MOS 镜像电流有源负载电路。

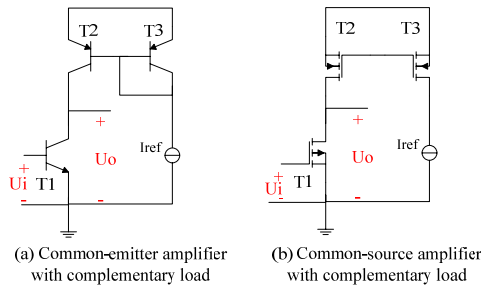


图 28 共射极（或共源极）互补型负载

Fig. 28 Common-emitter (or common-source) amplifier with complementary load.

2) 共射极（或共源极）耗尽型负载（depletion load）

如图 29 所示，负载也可以是耗尽型晶体管，以为该晶体管的行为特性很像线性区的镜像电流源，同时栅极短接源极。

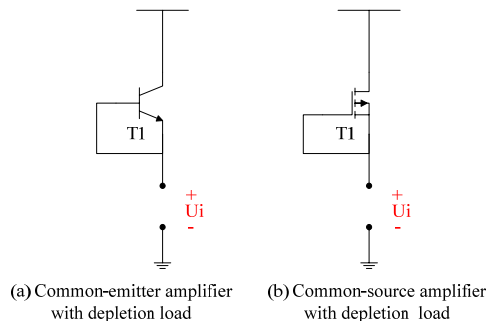


图 29 共射极（或共源极）耗尽型负载

Fig. 29 Common-emitter (or common-source) amplifier with depletion load.

3) 共射极（或共源极）二极管型负载（diodne connect load）

如图 30 示，负载也可以是二极管短接的晶体管，其特性很像线性区的镜像电流源，栅极短接源极，负载电阻小于或等于其负载晶体管的跨导的倒数（reciprocal of the transconductance of the load）。

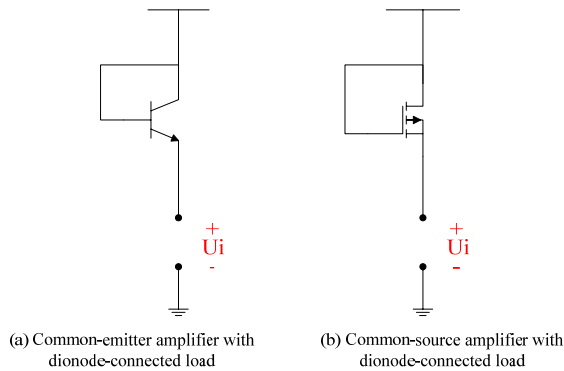


图 30 共射极（或共源极）耗尽型负载

Fig. 30 Common-emitter (or common-source) amplifier with diode-connected load.

### 3.7 可能不能正确识别的部分电路

双极型集成电路的工作电压为 $-15\sim+15V$ （相对于 MOS 集成电路是高电平），为了保证双极型晶体管的正常工作，当电流超过设定电流的时候，自动断电以保护电路，因此双极型集成电路中需要设计短路保护电路。针对双极型晶体管中的短路保护电路，各种非常巧妙的短路电路设计形式变化多样。例如，图 31 是 a725 的输出电路，有一些短路保护的晶体管，很难通过程序检测短路保护的晶体管。

互补金属氧化物半导体(CMOS: Complementary Metal-Oxide Semiconductor)的特征尺寸不断缩小，金属氧化物半导体(MOS: Metal-Oxide Semiconductor)的栅氧厚度越来越薄，MOS 管能承受的电流和电压也越来越小。因此一般采取全芯片静电放电（ESD: Electrostatic Discharge）保护结构的设计。因此 MOS 电路中几乎没有短路保护电路设计。

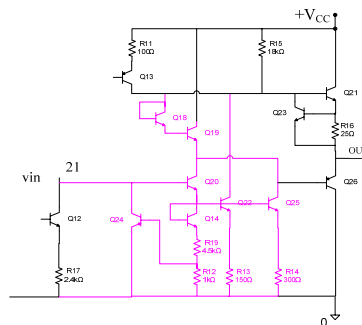


图 31 ua725 的短路保护电路

Fig. 31 Short-circuit protecting circuit of ua725.

---

### 3.8 小结

本章节主要介绍了一些模拟电路设计的基本概念，例如模拟电路晶体管的三种放大状态，例如共射（对应 MOS 共源）、共基（对应 MOS 共栅）和共集（对应 MOS 共漏）极三个状态。同时简单介绍了双极型晶体管和场效应管的小信号模型的基础知识，并引入基本端口模块的概念；该概念的提出主要和启发式排序算法结合，降低排序算法的复杂度，没有特殊的含义。由于模拟电路设计的复杂性，某些模拟电路很难被程序自动识别，本文只完成了共射（对应 MOS 共源）、共基（对应 MOS 共栅）和共集（对应 MOS 共漏）极三个状态的检测；对于模拟电路中千变万化的过流保护电路的设计还没有合适的电路识别算法。

## 第四章 基于电路拓扑的启发式符号排序算法的实现

### 4.1 排序算法在符号化模拟电路仿真器中的作用

符号化模拟电路仿真器（GRASS）有主要有三部分组成（如图 32），包括网表解析器、符号化分析器和求值运算器（提供各种数值分析结果）。

网表解析器由 PCCTS（Purdue Compiler Compiler Tool Set）生成。PCCTS 具有类似编译器生成工具 Lex 和 Yacc 的功能，并可以产生 C++代码的相关语法的解析器。经过网表解析器后，我们可以获得 SPICE 网表中描述的电路信息。

符号化分析器读入网表解析器所提供的电路信息，并建立电路对应的有向图，然后对有向图进行符号化分析，同时构建图约化判定图（SDD: Symbolic Decision Diagram）。在构建图约化判定图之前，由启发式排序算法自动生成符号处理顺序，并依照此顺序构建图约化判定图。如图 5 所示，符号化仿真器的计算结果为传输函数的符号化解析表达式，即  $H(s) = \frac{1}{X} = \frac{RCS}{1 + RCS}$ 。

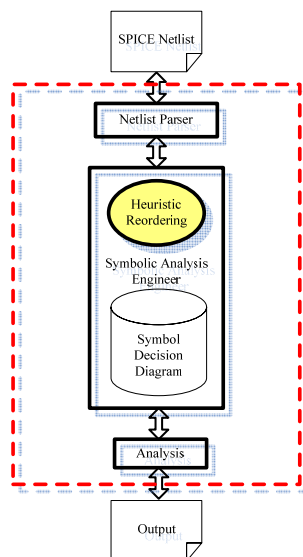


图 32 符号化仿真器结构

Fig. 32 The Symbolic Circuit Simulator.

数值化分析器是基于图约化判定图所提供的电路的相应的数值分析的结果，比如

---

计算不同频率下传输函数的振幅、相位、符号化表达式所含的生成项个数和一些分析的统计数据，并进行数值输出。

## 4.2 排序算法的对象

解析器读入电路网表，并根据网表自动生成初始符号处理顺序，排序算法的对象就是初始符号处理顺序。因为元件可以出现在网表文件的任意位置，同时该仿真器对符号处理顺序特别敏感，符号排序算法的目标是对任意组合的网表重新生成新的符号处理顺序。2.2.3 章节介绍了该符号化模拟电路仿真器的固有的优化策略，即第一个处理输入输出受控源符号和并联元件的预处理机制。

该仿真器在读入网表之后，首先会依据前面提到的第一个处理输入输出受控源符号和并联元件的预处理机制对元件符号重新排序，得到 Slist 数组。

- 1) Slist 数组的第一个元素是输出元件符号，第二个元素是输入元件符号。
- 2) 复合元件：网表中并联元件被看作同一个新元件（复合元件）。该新元件是某两点之间的所有的平行元件的总和，它是一个抽象的、实际不存在的元件。

Slist[i]中保存的是电路中某两点之间的、某个电路元件的符号（包括抽象的并联预处理的复合符号）。

该启发式排序算法在仿真器保持仿真器原有的优化策略的基础上对电路网表进行优化，因此保持 Slist[0]中的内容不变，在保持 Slist 数组中的内容不变的情况下，对 Slist 数组进行重新排序（即交换数组中某些行的内容从而构建新的 Slist 数组）。

## 4.3 排序算法的数据结构

该排序算法的数据结构的主要目的是为了将某个晶体管的小信号模型看做一个抽象的元件，排序算法仅考虑晶体管之间、晶体管和补偿电容、晶体管和电阻之间的排序。此举可以大大减小问题的规模。例如，每个双极型晶体管的小信号模型包含 4~5 个元件，而每个场效应管的小信号模型包含 8~9 个元件，因此需要排序的元件个数大大减小。

- 1) NodeTable 数组对应的新网表中的结点：该仿真器读入 SPICE 网表，该 SPICE 网表中某元件的端点号可以是任意的数字。当网表解析器读入该 SPICE 网表时，会对电路中存在的结点进行重新标记，即电路中所有的结点的编号和原来网表中的编号不一致。但是连接地的端点仍然是 0。如图 33 所示，R1 和



---

R2 分别都连接地信号，图 34 中在 R1 和 R2 都连接在 NodeTable[0]。

- 2) 电路中电阻、电容、电感和晶体管分别对应抽象的复杂元器件：3.2 章节中将电阻、电容、电感和晶体管分别抽象成二端口、三端口或四端口元件。其目的就是为了将晶体管看做一个整体，简化处理的问题规模。5.1.3 章节对两个相邻的双极型晶体管的小信号模型进行了排序实验，实验数据表明将某个晶体管的小信号模型看做一个整体是有利于提高仿真器的效率。
- 3) 抽象的二端口、三端口或四端口复杂元件：电阻、电容和电感对应二端口元件；射极（对应 MOS 栅极）和衬底短接的晶体管可看作三端口元件；射极（对应 MOS 栅极）和衬底不短接的晶体管可看作四端口元件。
  - a) 电阻元件：可看作二端口元件。该元件的类型定义为`_RR`(电阻类型)，其电阻元件的保存在 `Elem_table[0]`中；`num` 的值为 1，其为 `Elem_table` 中元件总和。
  - b) 电容元件：可看作二端口元件。该元件的类型定义为`_CC`(电阻类型)，其电容元件的保存在 `Elem_table[0]`中；`num` 的值为 1，其为 `Elem_table` 中元件总和。
  - c) 电感元件：可看作二端口元件。该元件的类型定义为`_LL`(电阻类型)，其电感元件的保存在 `Elem_table[0]`中；`num` 的值为 1，其为 `Elem_table` 中元件总和。
  - d) 晶体管元件：可看作三端口或四端口元件。该元件的编号定义为 `id`（例如 Q16 的 `id` 为 16，`rpi16` 的元件保存在 Q16 的 `Elem_table` 数组中）。该元件的类型定义为`_BJT`，该小信号模型中包含的元件个数保存在 `num` 中，相应的小信号模型都保存在 `Elem_table` 中（例如某 BJT 小信号模型包含 `cpix`、`rpix`、`cmux`、`rox`、`gmx`，其五个元件都保存在 `Elem_table` 中，元件个数保存在 `num` 中）。`Terminal` 中保存的是小信号模型的三个或四个端口的结点信息（BJT 的 `Terminal` 中保存`[collector]`、`[base]`、`[emitter]`三个端口号，同时 MOS 小信号模型中保存的是`[drain]`、`[gate]`、`[source]`、`[base]`的四个端口号）。

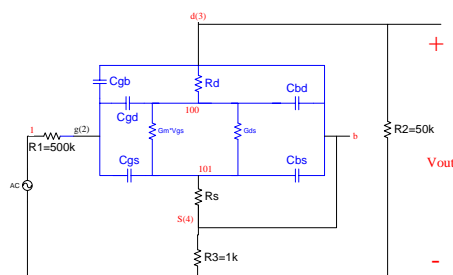


图 33 单个源极-源极衰减电路

Fig. 33 A simple commonsource-sourcedegeneration circuit.

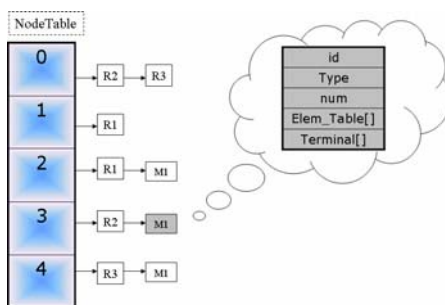


图 34 单个源极-源极衰减电路的数据结构

Fig. 34 Data structure for a simple circuit.

## 4.4 介绍能识别电路功能模块的搜索算法

在模拟电子电路尤其是模拟集成电路中，广泛采用电流源作偏置电路。它可以提供偏置电路和较大的有源负载。为了让晶体管可以稳定的工作在放大状态，通常也需要提供稳定的交流地信号。镜像电流源的输出端可以提供稳定的直流电压，同时也是理想的交流地信号。因此，为了判断出具有放大电路的晶体管的共极状态，需要首先识别电路中的镜像电流源电路。

### 4.4.1 识别镜像电流源结构

如 3.3.3 章节中所示，常见的镜像电流源主要分为简单镜像电流源、带有 Beta-help 的镜像电流源、带反馈的镜像电流源、级联镜像电流源和 Wilson 镜像电流源，该算法能识别以上几种常见的镜像电流源电路。

镜像电流源电路是由晶体管组成，可分为三端口或四端口元件（基极和栅极短接

---

或栅极和源极短接时看作三端口元件)。同时镜像电流源都是将某参考电流源进行输出。因此需要在仿真器读入的 SPICE 网表中写上理想输入电流源的输入结点信息。

根据电路中理想电流源的输入结点，对电路图进行遍历：

- 1) 从电流源的输入结点开始，找出电路中所有的镜像电流源；
- 2) 将镜像电流源的输出结点标记为交流地信号。

#### 4.4.2 识别晶体管的三种共极状态

3.3 章节中介绍了晶体管的三种共极状态，分别为共射（对应 MOS 共源）、共基（对应 MOS 共栅），共集（对应 MOS 共漏）三种状态。在共射（对应 MOS 共源）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从集电极（对应 MOS 漏极）离开该晶体管；在共基（对应 MOS 共栅）的电路结构中，输入信号从晶体管的发射极（对应 MOS 的源极），并从集电极（对应 MOS 漏极）流出晶体管；在共集（对应 MOS 共漏）的电路结构中，输入信号加载在晶体管的基极（对应 MOS 栅极），并从发射极（对应 MOS 源极）离开该晶体管。

假设电路中所有的镜像电流源的结构均已检测出，并且镜像电流源的输出结点标记为交流地信号。图 35、36 中节点 2 被标记为交流地信号。已知交流信号的输入节点为 11。下面介绍三种共极状态的检测：

- 1) 共射（对应 MOS 共源）极
  - a) 非对称共射（对应 MOS 共源）极：
    - i. 射极（或源极）连接交流地信号；
    - ii. 基极（或栅极）连接输入信号；
    - iii. 集极（或漏极）输出信号不接地；
    - iv. 晶体管的集极（或漏极）有负载；
    - v. 晶体管的集极（或漏极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。
  - b) 对称共射（对应 MOS 共源）极：
    - i. 两个晶体管的射极（或源极）连接，且连接交流地信号；
    - ii. 两个晶体管基极（或栅极）不相同，分别连接两个输入信号；
    - iii. 两个晶体管集极（或漏极）不相同，均不接地；
    - iv. 两个晶体管的集极（或漏极）有负载；

- v. 其中某个晶体管的集极（或漏极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。

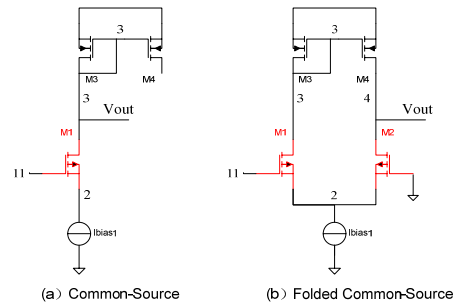


图 35 共源极和折叠型共源极电路

Fig. 35 Common-source and folded common-source circuit.

2) 共基（对应 MOS 共栅）极

a) 非对称共基（对应 MOS 共栅）极：

- i. 基极（或栅极）连接交流地信号；
- ii. 射极（或源极）连接输入信号；
- iii. 集极（或漏极）输出信号不接地；
- iv. 晶体管的集极（或漏极）有负载；
- v. 晶体管的集极（或漏极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。

b) 对称共基（对应 MOS 共栅）极：

- i. 两个晶体管的基极（或栅极）连接，且连接交流地信号；
- ii. 两个晶体管射极（或源极）不相同，分别连接两个输入信号；
- iii. 两个晶体管集极（或漏极）不相同，均不接地；
- iv. 两个晶体管的集极（或漏极）有负载；
- v. 其中某个晶体管的集极（或漏极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。

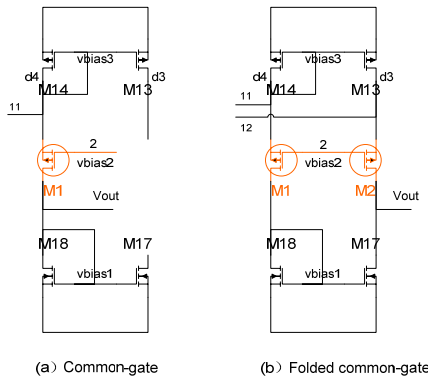


图 36 共栅极和折叠型共栅电路

Fig. 36 Common-gate and folded common-gate circuit.

### 3) 共集（对应 MOS 共漏）极

#### a) 非对称共集（对应 MOS 共漏）极：

- i. 集极（或漏极）连接交流地信号；
- ii. 基极（或栅极）连接输入信号；
- iii. 射极（或源极）输出信号不接地；
- iv. 晶体管的射极（或源极）有负载；
- v. 晶体管的射极（或源极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。

#### b) 对称共集（对应 MOS 共漏）极：

- i. 两个晶体管的集极（或漏极）连接，且连接交流地信号；
- ii. 两个晶体管基极（或栅极）不相同，分别连接两个输入信号；
- iii. 两个晶体管射极（或源极）不相同，均不接地；
- iv. 两个晶体管的射极（或源极）有负载；
- v. 其中某个晶体管的射极（或源极）是输出结点或其输出结点是某些晶体管的输入结点，即漏极是电路的输出结点或漏极还连接着非镜像电流源电路的某晶体管的栅极或源极（对应 BJT 管的基极或射极）。

#### c) Push-pull 电路：该电路和对称共集（对应 MOS 共漏）极的区别是 M1 和 M2 的栅极（对应 BJT 基极）相连作为输入信号，同时源极（对应 BJT 的射极）相连作为输出信号。该电路的检测与对称共集（对应 MOS 共漏）极相同，区别是输入结点和输出结点相同的电路就 push-pull 电路。

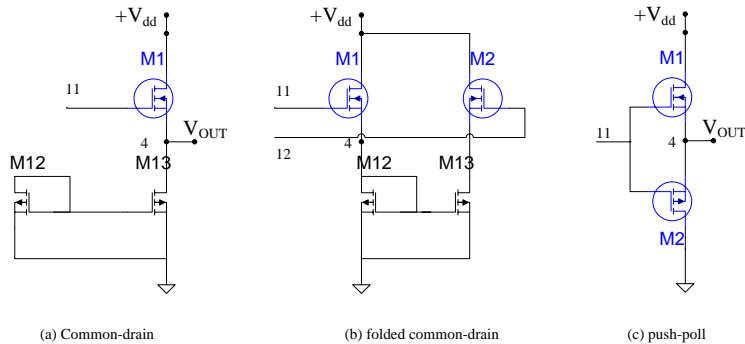


图 37 共漏极和折叠型共漏极和上推下拉式电路  
 Fig. 37 Common-drain, folded common-drain and push-poll circuit.

## 4.5 对功能模块重新排序算法设计

4.4 节详细介绍了检测模拟电路中晶体管的共极状态的搜索算法，本节主要介绍对功能模块进行重新排序的算法。

根据大量的实验数据表明，当电路中只存在共射（对应 MOS 共源极）电路和共集（对应 MOS 共漏极）电路时，与小信号流动方向相反的模块排序的优化效果较好；当出现共基（对应 MOS 共栅）、共集-共射电路和共集-共基电路时，与小信号流动方向一致的模块排序效果较好。整个电路的排序被分为功能模块和镜像电流源的排序：

- 1) 镜像电流源电路只为一个功能模块提供直流偏置电流，例如图 38 (a, d)，则模块之间的排序为先处理 d 模块再处理 a 模块。
- 2) 镜像电流源电路为两个或两个以上的功能模块提供直流偏置电流，例如图 38 (b, c, d, e)，先处理功能则模块之间排序为 ebcf 或 fcbe，即栅极相连的晶体管应当排序很紧凑。
- 3) 非对称共射（对应 MOS 共源）极功能模块：先处理负载，再处理共射（对应 MOS 共源）极晶体管；
- 4) 对称共射（对应 MOS 共源）极功能模块：如果存在提供直流电压偏置的晶体管，其栅极和某个共射（对应 MOS 共源极）的栅极相连，则先处理提供直流电压偏置的晶体管，再处理负载晶体管（如果是互补型负载则先处理基极和射极短接或栅极和源极短接的晶体管，再处理镜像晶体管），最后处理共射（对应 MOS 共源）极晶体管。否则，处理负载晶体管，最后处理共射（对应 MOS 共源）极晶体管。如图 38 (e) 中 M3 晶体管，其漏极和栅极相连，

该晶体管提供直流电压偏置，则应当先处理 M3，再处理负载 load1 和 load2，再处理 M1 和 M2。

- 5) 非对称和对称的共基（对应 MOS 共射极）电路：先处理共基（对应 MOS 共射极）电路，再处理给基极或栅极提供直流偏置电压的晶体管。如图 39（a）（b）中，先处理 M1 和 M2，再处理提供提供 vbias 的电路，再处理负载晶体管；图 39（c, d）中，M1 和 M2 是对称共栅极晶体管，同时 M3 和 M4 也是对称共栅极晶体管，同时其互为对方的负载，在此情况下，针对图 39（c）的电路，先处理 M1 和 M2，再处理提供提供 vbias 的电路，而不处理负载 M3 和 M4。
- 6) 非对称的共集-共射电路：如图 40（a），先处理负载，再处理共射极晶体管，最后处理共集极晶体管。
- 7) 对称的共集-共射电路：如图 40（b），先处理两个负载（晶体管），再处理与之相邻的共射极晶体管和共集极晶体管（如果是互补型负载则先处理基极和射极短接晶体管）。
- 8) 非对称的共集-共基电路：先处理负载，再处理共基极晶体管，最后处理共集极晶体管。
- 9) 对称的共集-共基电路：先处理负载（晶体管），再处理与之相邻的共基极晶体管和共集极晶体管（互补型负载则先处理基极和射极短接的晶体管）。
- 10) Push-poll 电路：该电路中的两个晶体管的排序两种皆可。

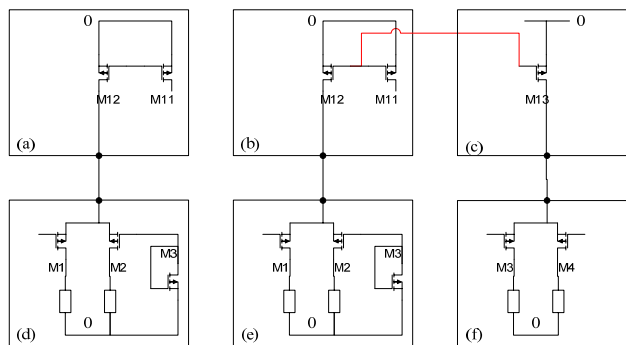


图 38 共射（对应 MOS 共源）极模块内部排序例子

Fig. 38 A CE (or CS) module example.

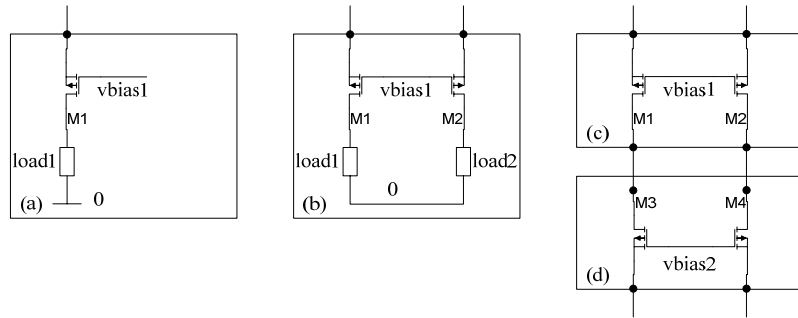


图 39 共基（对应 MOS 共栅）极模块内部排序例子  
Fig. 39 A CB (or CG) module example.

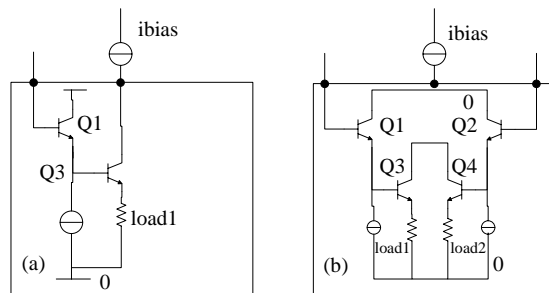


图 40 共集-共射电路的内部排序例子  
Fig. 40 A CC-CE module example.

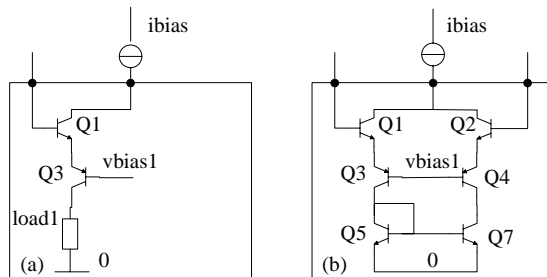


图 41 共集-共基的模块内部排序例子  
Fig. 41 A CC-CB module example.

## 4.6 对未能正确检测电路结构的晶体管的算法排序设计

3.7 章节中讲述了双极型集成电路工作于较高电压 (-15V~+15V)，同时 CMOS 集成电路由于工作于低电压(0-3.3V)，同时较多采用静电放电设计，因此短路保护电路常见于双极型集成电路，而 CMOS 集成电路中却鲜见短路保护电路。

与共集共射电路或共集共基电路相比，短路保护电路的晶体管分别比较密集，如



47 和 48 中短路保护电路；实际上，该短路保护电路的符号顺序对仿真器影响不大。

针对未能正确检测电路结构的电路，采用一下策略：

- 1) 先排序短路保护中镜像电流源电路
- 2) 从上一级电路的输出结点为出发点，首先排序与该节点相连接的晶体管，随后采用广度优先遍历的次序对短路保护电路进行排序。
- 3) 统计已排序的晶体管个数，并保证所有的晶体管都已重新排序。

## 4.7 测试变量排序对仿真器影响的实验设计

为了测试不同符号处理顺序对仿真器性能的影响，同时以不改变仿真器为前提，选择使用 Perl 脚本不停更换网表，并让仿真器反复读入不同的排序的文件（netlist），同时统计仿真器运行结果。其目的是为了在短时间内测试大量的排序组合。

### 4.7.1 Perl 脚本自动测试不同符号排序的实验基本原理

基于 GRASS 仿真器的变量排序的实验方法由两部分组成（见图 42），包括 Perl 脚本和符号化模拟电路仿真器。该脚本主要负责读入 SPICE 网表，生成某个全排列序列，并根据该序列生成新网表，统计仿真器的输出信息来保存最优和最差的网表。仿真器就对读入网表，构建分析电路对应的 SDD（Symbol Decision Diagram），因为 SDD 的规模直接影响到该仿真器的执行效率，因此对 SDD 的结点数电路进行标准数据流输出，以便于脚本统计相关信息。

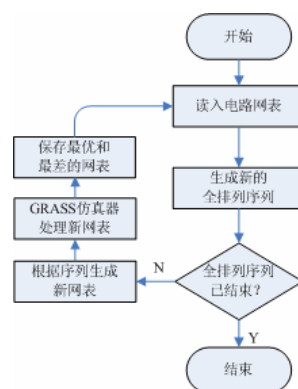


图 42 基于 GRASS 的变量排序实验设计原理

Fig. 42 Process of symbol permutation experiment.

Perl 脚本自动测试不同符号排序的实验设计方法的最大优点是自动化分析网表，

节省时间，避免重复性劳动；自动统计排序最优、最差的网表，可以在短时间内得到大量的、关于排序的重要信息。该实验方法的缺点是随着测试电路规模的增大，测试的全排列序列的程序时间复杂度为  $n!$ （例如，自然数  $n$  的全排列序列有  $n!$ ），因此只能对部分电路进行全排列排序的测试。

#### 4.7.2 Perl 脚本自动测试不同符号排序的具体实现

Perl 脚本主要负责读入 SPICE 网表，生成某个全排列序列，并根据该序列生成新网表，统计 GRASS 仿真器的输出信息来保存最优和最差的网表。

- 1) 读入 SPICE 网表：读入 SPICE 网表之后，将相应的信息存储到 `$line` 数组中。GRASS 仿真器的网表解析器（parser）将 “\*\*order\_digit” 看做注释行，不对该行作任何处理；同时 Perl 脚本以 “\*\*order\_digit” 为关键字，将读入的信息存储到相应的下标为 digit 的 `@$line` 数组中。如图 43 所示，自 “\*\*order\_1” 到 “\*\*order\_2” 的前一行的元件信息存入 `$line[1]` 中，当 Perl 脚本将整个 SPICE 网表读入结束时 `@line` 数组中保存了所有的电路网表。

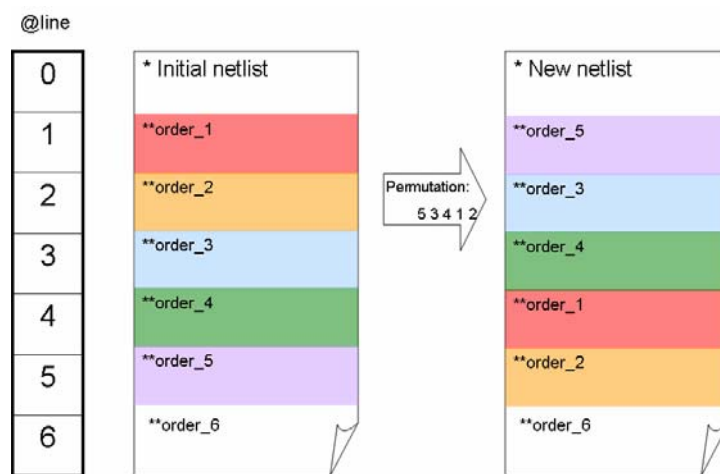


图 43 脚本中存储的网表示意图

Fig. 43 An example of new SPICE netlist according to initial netlist.

- 2) 自动生成某个全排列序列：采用了经典的 Heap 算法自动生成全排列序列。全排列是将一组数按一定顺序进行排列，如果这组数有  $n$  个，那么全排列数为  $n!$  个。该算法由三部分组成：
  - a) 读入自然数  $n$ 。该自然数决定了生成全排列序列的规模。例如，如果  $n=1$ ，全排列序列为  $\{1\}$ ；如果  $n=2$ ，全排列序列为  $\{1, 2\}$ ， $\{2, 1\}$ ；如果  $n=3$ ，

- 全排列序列为{1, 2, 3}, {1,3, 2}, {2, 1, 3}, {2 3 1}, {3, 1, 2}, {3, 2, 1}。
- b) 子程序 Perm。如图 44 所示，采用深度优先搜索算法，即对每一位上的数进行枚举而求得所有排列；使用递归算法完成全排列排序。全排列是将一组数按一定顺序进行排列，如果这组数有  $n$  个，那么全排列数为  $n!$  个。如果用  $P$  表示  $n$  个元素的排列，而  $P_i$  表示不包含元素  $i$  的排列， $(i)P_i$  表示在排列  $P_i$  前加上前缀  $i$  的排列，那么， $n$  个元素的排列可递归定义为：如果  $n=1$ ，则排列  $P$  只有一个元素  $i$ ；如果  $n>1$ ，则排列  $P$  由排列 $(i)P_i$  构成 ( $i=1、2、\dots、n-1$ )。根据定义，容易看出如果已经生成了  $k-1$  个元素的排列，那么， $k$  个元素的排列可以在每个  $k-1$  个元素的排列  $P_i$  前添加元素  $i$  而生成。例如 2 个元素的排列是 1 2 和 2 1，对与个元素而言， $p_1$  是 2 3 和 3 2，在每个排列前加上 1 即生成 1 2 3 和 1 3 2 两个新排列， $p_2$  和  $p_3$  则是 1 3、3 1 和 1 2、2 1，按同样方法可生成新排列 2 1 3、2 3 1 和 3 1 2、3 2 1。
- c) 子程序 Swap。如图 44 所示，其主要作用是交换两个数字的值。

```

Swap(int &a, int &b)
{ 交换a 和b的值; }

Perm(int list[], int k, int m)
{ //生成list[k:m]的所有全排列方式
  int i;
  if(k>m) {输出一个全排列序列}
  else{// list[k:m]有多个全排列序列
    //递归产生这些排列方式
    for(i=k; i<=m; i++){
      交换list[k]和list[i]的值;
      Perm(list, k+1, m);
      交换list[k]和list[i]的值;
    }
  }
}

```

图 44 递归实现的全排列算法

Fig. 44 An example of permutation algorithm.

- 3) 根据某个全排列序列生成新网表。以图 43 为例，读入的网表文件被分成 7 块存储在 @line 中。生成自然数 1~5 的全排列序列，该序列对应 \$line[1]~\$line[5] 中的信息。生成新的网表的过程就是将 @line 中的信息输出到特定的文件中，其中 @line[0]和 \$line[6]的相对位置与初始网表相同，改变的是 Sline[1]~\$line[5] 之间的相对位置。值得一提的是，该网表读入的最大自然数比生成的全排列

---

序列的最大自然数多 1 (例如图 43 所示, \*\*order\_6 的最大关键字为 6, 但是生成的全排列序列为{5, 3, 4, 1, 2}, 全排列序列中最大的自然数为 5, 即  $6 = \max\{5, 3, 4, 1, 2\}+1$ )。

- 4) 统计 GRASS 仿真器的输出信息来保存最优和最差的网表。当生成新的特定的网表文件后, 让 GRASS 仿真器去处理该网表, 并将关键信息 SDD 的结点数进行流输出。Perl 脚本一直跟踪 GRASS 仿真器的输出信息, 统计 SDD 最大和最小的数字, 并将对应的网表存储下来。在 GRASS 仿真器中, 与 SDD 结点数相关的参数是 `tpddnodes` 和 `Tpddnodes_AfterReduced`, 考虑到构建 SDD 耗费的内存较大, 因此程序设计成 `tpddnodes` 最小的时候对应的是最优的排序网表; `tpddnodes` 相等的情况就认为两个网表的排序是等价的。

### 4.7.3 变量排序实验的优点和局限性

该实验方法的最大优点是自动修改网表, 并对同一个电路可以自动生成所有可能电路排序, 并自动统计 SDD (Symbol Decision Diagram) 的规模, 并以此为依据统计出最优的和最差的网表。节省时间, 避免重复性劳动, 同时可以在短时间内得到大量的、关于排序的重要信息。

该实验方法的缺点是随着测试电路规模的增大, 测试的全排列序列的程序时间复杂度为  $n!$  (例如, 自然数  $n$  的全排列序列有  $n!$ ), 因此只能对部分电路进行全排列测试。

下图 45 表示随着自然数  $n$  的增加, 生成全排列序列的个数呈  $n!$  倍增长。下图 46 表示的是随着自然数  $n$  的增加, 得到全部的全排列序列的运行时间。值得强调的是, 图 46 中计算的时间仅仅是为了得到全排列序列所耗费的时间, 不包括 GRASS 仿真器的运行时间。因此, 在实际的电路测试中, 考虑到 GRASS 仿真器所耗费的时间, 一般不选取大于自然数 9 的全排列序列 ( $n \leq 9$ ) (例如 {1, 2, 3, 4, 5, 6, 7, 8, 9})。

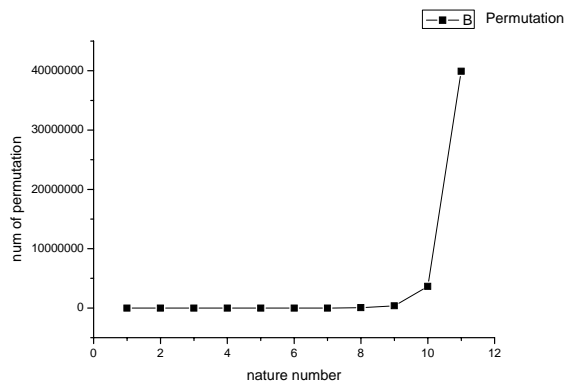


图 45 全排列组合的总和.

Fig. 45 Sum of permutaion.

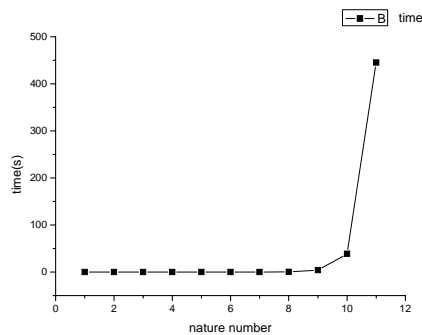


图 46全排列组合的时间总和

Fig. 46 Sum of time for permutation.

## 4.8 小结

本章采用穷举法来测试各种符号处理顺序，结合 perl 脚本编程实现全自动测试和统计实验数据，并在下一章节测试大量网表。由于该实验方法只能测试一定数量的全排列数据，因此该实验方法只能一定程度上接近最优变量排序。分析大量实验数据后，本节提出了启发式变量排序算法，该算法依赖于大量的测试网表的经验总结。大量的电路测试表明，根据电路功能进行模块划分，考虑模块间排序、模块内的晶体管级排序和小信号流动方向可以明显的提高仿真器的效率。

---

## 第五章 实验结果分析

本章将介绍基于 GRASS 仿真器的变量排序的实验设计方法，并设计了一系列基于该实验的电路测试例子，得到电路排序的重要信息。通过这些实验数据，总结并提出了基于电路拓扑结构的启发式变量排序算法。

### 5.1 不同拓扑结构电路的实验结果

符号化模拟电路仿真器（GRASS）只能处理非线性元器件（详见 2.1 节），而模拟电路中大量包含双极型晶体管(BJT: Bipolar Junction Transistor)和场效应管（MOS: Metal-Oxide-Semiconductor）等非线性电路元件。因此在处理模拟电路之前，需要用 SPICE 仿真得到晶体管的静态工作点，并计算出 BJT 和 MOS 管在静态工作点附近的小信号模型，将非线性电路元件转换成线性电路元件。一般模拟电路中包含大量晶体管、负载电阻、补偿电容和补偿电感等元件，由于晶体管模型的内部变量个数以及晶体管的规模远远超过这些简单的电阻、电容和电感，所以几乎可以忽略这些元件对仿真器性能的影响。

综上，基于符号化模拟电路仿真器的变量排序实验重点考察以下三个方面：单个晶体管模型内部变量排序、两个晶体管模型内部变量的混合排序、两个相同或不同的功能模块的晶体管级（BJT 和 MOS 级别的晶体管）的混合排序以及补偿电容和补偿电阻与相邻晶体管的排序。

#### 5.1.1 符号排序实验的主要测试电路

选取不同规模的模电电路进行符号排序的实验，目的是测试不同排序对仿真器性能的影响。测试电路如下：

电路一：ua741 运放（图 47，含 24BJT），有 81 条边和 24 个结点。

电路二：ua725 运放（图 48，含 26BJT），有 98 条边和 31 个结点。

电路三：三级运放（图 49，含 11MOS），有 80 条边和 34 个结点。

电路四：三级运放（图 50，含 15 MOS），有 106 条边和 44 个结点。

电路五：双端转单端电路（图 51，含 16 MOS），有 106 条边和 44 个结点。

电路六：端对端电路（图 52，含 22MOS），有 168 条边和 61 个结点。

测试平台选用的是 AMD Athlon 64\*2 Dual Core Processor 4400+, 2G 的内存的个人 PC。

对于上述的测试电路集合，计算其传输函数的符号化表达式并作数值分析。下列的参数可以说明电路的规模：**#Edge** 代表电路所含的分支数，对应电路的所有的元器件的个数（包括 BJT 和 MOS 小信号模型中的电阻和电容）；**#Edge\_lump** 代表将预处理并联元件后电路所含的分支数（即将并联的多条边看作一个复合的元器件，对应图上的操作就是将多条边等效为一条边）；**#Node** 代表电路所含的节点数（包括小信号模型中的内部节点数）；**#Symb** 代表电路所含的符号数（仿真器将输入/输出抽象成为一个符号  $X$ ，即  $N(s)/D(s)=1/X$ ，因此符号数比并联预处理之后的分支数多 1，即  $\#Symb=\#Edge\_lump+1$ ）；**#Term** 代表传输函数符号表达式所含生成项的数目；**|SDD|** 为图约化判定图所含节点的个数；**Time** 是仿真时间；**Memory** 为仿真所需的内存空间。**|SDD|**的大小、仿真时间和内存与电路元件处理的顺序相关，但对于同一电路而言，**#Term** 是恒定不变的。

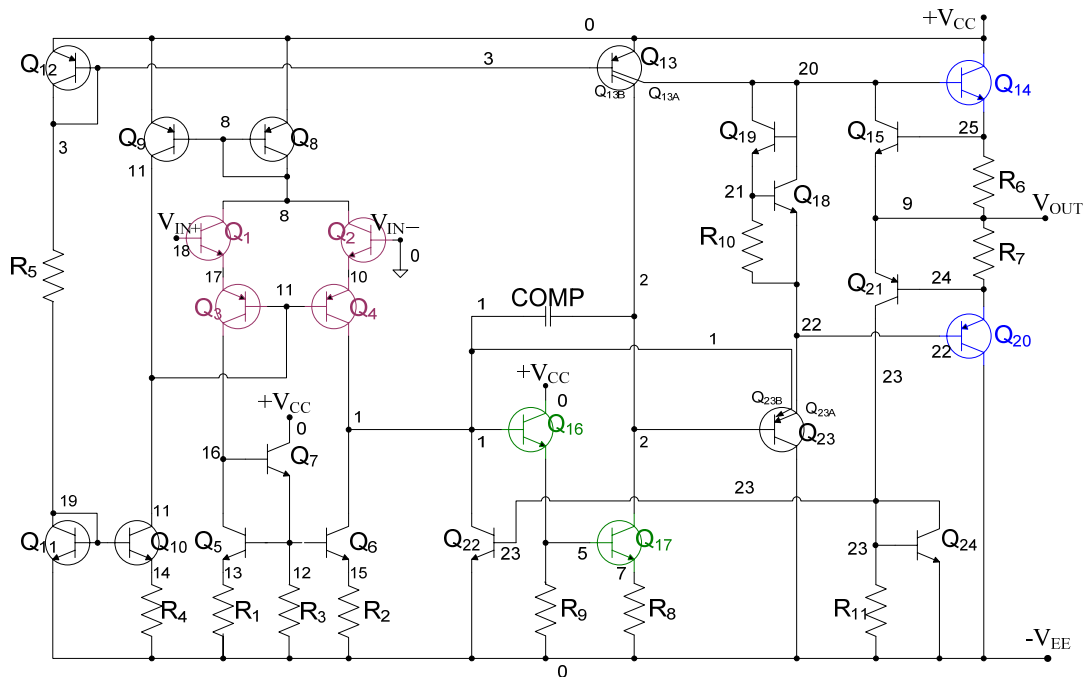


图 47 ua741 (包含 24 个 BJT 管)

Fig. 47 ua741 circuit (including 24 BJT transistors).

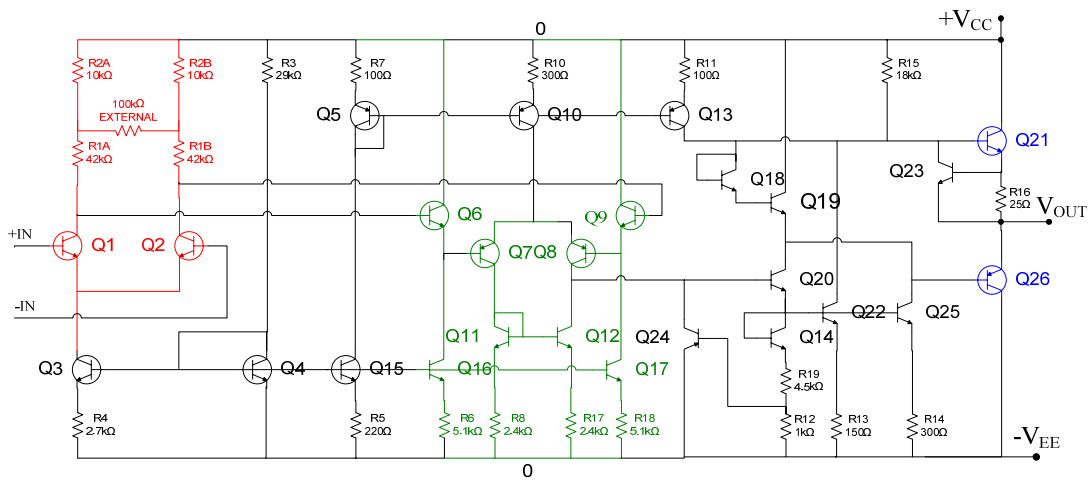


图 48 ua725 (包含 26 个 BJT 管)

Fig. 48 ua725 circuit (including 26 BJT transistors).

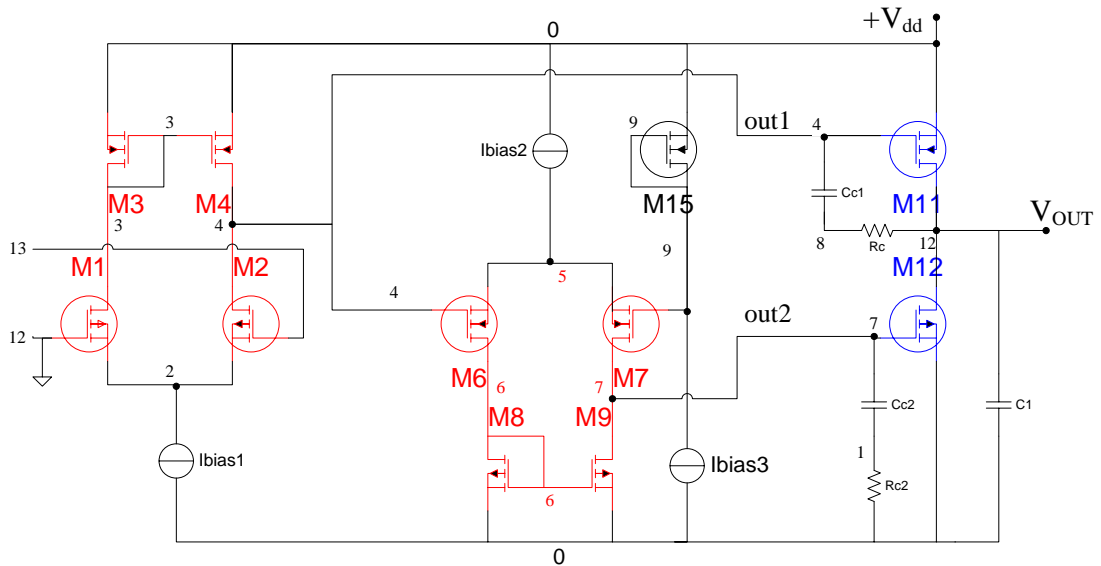


图 49 三级运算放大器 (包含 11MOS 管)

Fig. 49 Three-stage OPAMP (including 11 mos transistors).



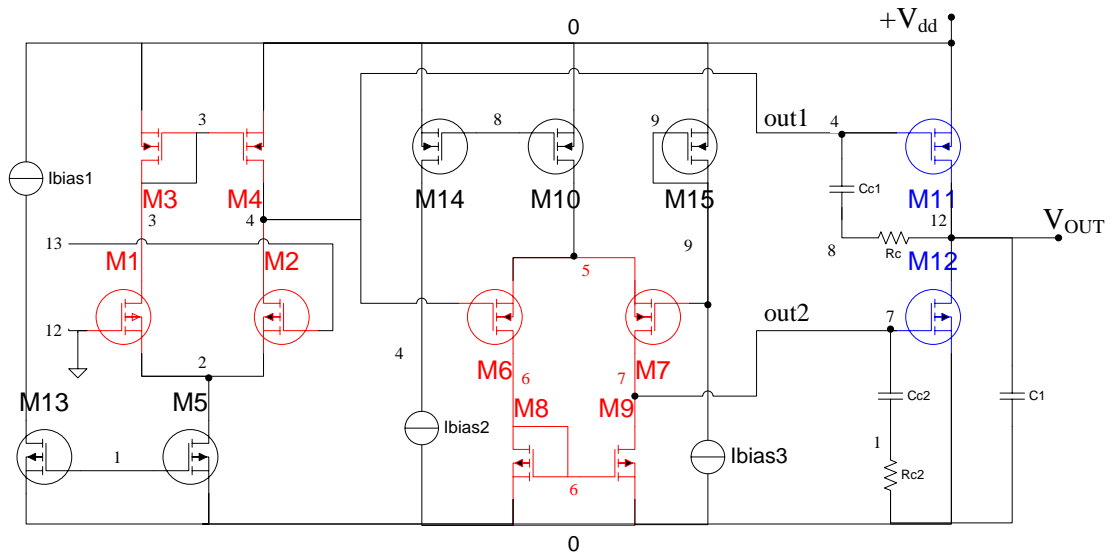


图 50 三级运算放大器电路图 (包含 15 个 MOS 管)  
 Fig. 50 Three-stage OPAMP (including 15 mos transistors).

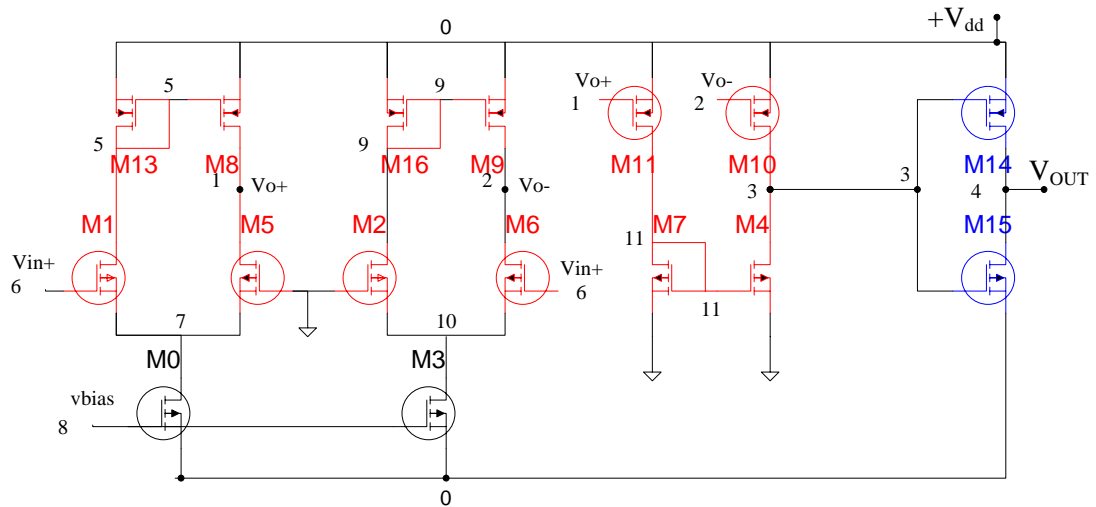


图 51 双端转单端电路 (包含 14 个 MOS 管)  
 Fig. 51 Differential to single circuit (including 14 MOS transistors).

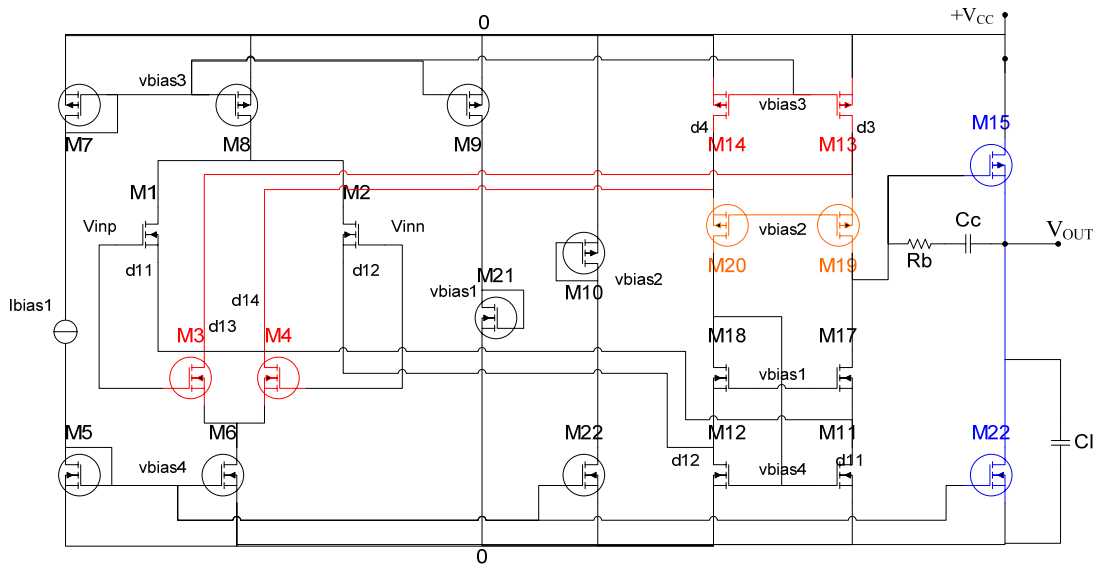


图 52 端对端电路图 (包含 22MOS 管)

Fig. 52 Rail to Rail circuit (including 22 MOS transistors).

### 5.1.2 单个晶体管模型的内部变量排序

BJT 和 MOS 晶体管的小信号模型如图 16、17 所示，都是只包含一个晶体管、不同共极状态的电路，本小节的测试是基于晶体管与电路中其它元件的相对排序不变的情况下，研究了单个晶体管内部变量排序对仿真器性能的影响，并统计出最优和最差的晶体管内部排序。测试电路的信息保存在表 1 中，最优和最差的符号处理顺序所生成的 SDD 结点个数保存在表 2 中。

#### 1) 双极型晶体管的小信号模型

如图 16，单个双极型晶体管有共射极 (CE: Common Emitter)、共基极 (CB: Common Base)、共集极 (CC: Common Collector) 和共射-射极衰减电路 (CE-ED: CommonEmitter-EmitterDegeneration)。表 1 中列出了对应图 15 电路中的边、结点和符号数等相关信息，表 2 中列出单个晶体管内部变量全排序对仿真器性能的影响。

#### 2) 场效应管的小信号模型。

如图 17，单个场效应管的应用有共源极 (CS: Common Source)、共栅极 (CG: Common Gate)、共漏极 (CD: Common Drain) 和共源-源极衰减电路 (CS-SD: Common Source - Source Degeneration)。表 1 中列出了图 16 电路中电路图的各项参数，表 2 中列出单个晶体管内部变量全排序对仿真器性能的影响。

No.	circuit	#Edge	#Edge_lump	#Node	#Symb	#Term
1	CE	10	5	4	6	8
2	CB	10	5	4	6	8
3	CC	10	5	4	6	9
4	CE-ED	11	7	5	8	26
5	CS	14	10	6	11	71
6	CG	14	10	6	11	71
7	CD	14	10	6	11	121
8	CS-SD	15	11	7	12	171

表 1 单晶体管放大电路详细信息  
Table 1 Circuit information including single transistor.

		Best ordering			Worst ordering		
		SDD	Time(s)	Memory(M)	SDD	Time(s)	Memory(M)
1	CE	15	0	0	22	0	0
2	CB	15	0	0	21	0	0
3	CC	16	0	0	20	0	0
4	CE-ED	32	0	0	46	0	0
5	CS	48	0	0	113	0	0
6	CG	48	0	0	116	0	0
7	CD	72	0	0	163	0	0
8	CS-SD	78	0	0	181	0	0

表 2 单晶体管电路的最优和最差排序的仿真对比  
Table2 Contrast of the best and worst ordering for single transistor.

图 53 列出单个晶体管（包括 CE、CB、CC、CE-ED、CS、CG、CD、CS-SD 共 8 种电路）结构的最优和最差排序仿真对比，图 53 中的横坐标的自然数分别对应表 1 和 2 中的行号，即对于不同的电路网表；其纵坐标对应 SDD 的结点个数（SDD 的数字越大，代表仿真器的运行时间更长）。可见对于单晶体管电路，不同共极状态的晶体管的模型内变量排序对仿真器的影响不大。图 54 统计了 CS-CS 电路的不同变量排序的 SDD（Reduced SDD）的分布图，该图统计了 SDD 个数相同的符号处理顺序的总和。

总结，针对单晶体管电路来说，模型内部变量排序仿真器的影响很小；较好的变量排序远远小于较差的变量排序。

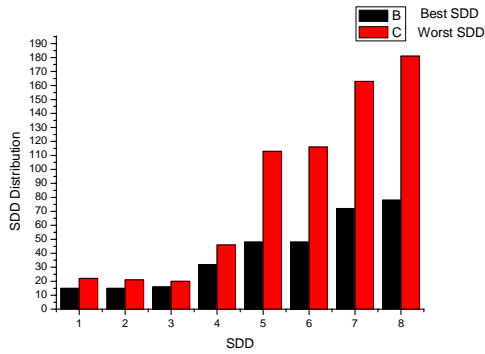


图 53 单个晶体管内部最优排序和最差排序的仿真对比  
 Fig. 53 Contrast of the best and worst ordering for simulator.

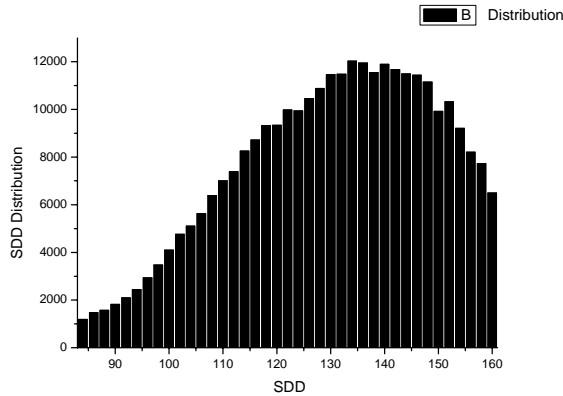


图 54 CS-SD 单晶体管电路的 SDD 分布图  
 Fig. 54 SDD distribution for a CS-SD circuit (including a MOS transistor).

### 5.1.3 两个晶体管模型的混合变量排序

通过 5.1.2 小节的实验说明，单晶体管电路的晶体管模拟内部变量排序对仿真器的性能影响一般不大。本节讨论相邻的两个晶体管模型的变量的混合排序，选取 ua741（如图 47）的第一部分放大电路。本实验重点相邻的晶体管小信号模型的所有排列顺序对仿真器性能的影响。本节实验的目的是为了“是否应当将晶体管的小信号模型”看作一个整体（多元器件的复合元件），是否存在相邻的两个晶体管的内部变量混合排序的效果优于将两个晶体管分别看作两个复合元件：

- 1) 双极型晶体管的小信号模型（如图 14 (a)）

- A. 两个 BJT 小信号模型的混合排序的实验，采用混合型 H 小信号模型，进行并联预处理的效率（如 3.3.2 章节），则两个 BJT 小信号模型最多有 8 个不同符号，其全部测试样本案例为  $8! = 40320$ ，可以在有限的时间内可以完成实验。
- B. 测试多晶体管电路，包括共集-共射电路、共集-共基电路和 Beta-Help 电路（如图 55 (b)）。

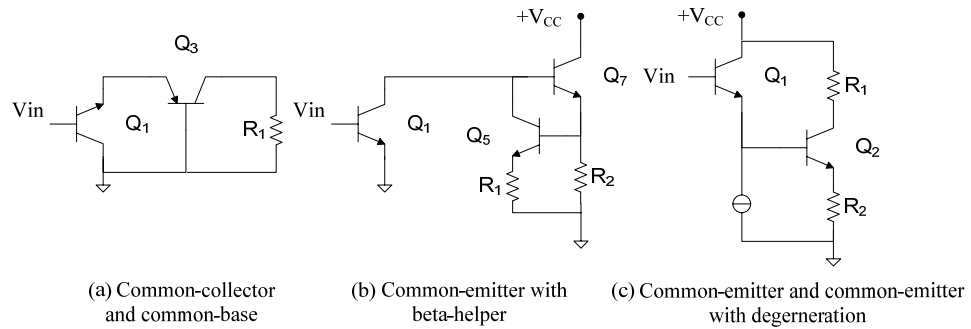


图 55 多晶体管电路

(a) 共集-共基电路; (b) beta-help 负载电路; (c) 共集-共射电路

Fig. 55 An example of multiple transistors.

2) 场效应管的小信号模型（如图 14 (b)）

- A. 两个场效应管的小信号模型的混合排序的实验，由于有 18 个变量，需要测试的样本案例达到  $18! = 6.4e+15$ ，在有限时间内无法完成该实验设计（详见 5.1.2 节）。因此，基于 MOS 晶体管的两个管子的混合排序实验以失败告终。
- B. MOS 小信号模型的复杂度高于 BJT 小信号模型，其规模为后者的两倍多，同时 MOS 小信号模型中有两个内部结点。因此，直觉上认为 MOS 小信号的混合变量排序的效果一定很差。即对 MOS 小信号模型，某个晶体管的小信号模型中不能插入非该模型的元器件（电阻、电阻和其它晶体管模型）。

No.	circuit	#Edge	#Edge_lump	#Node	#Symb	#Term
1	CC-CB	15	7	4	8	12
2	Beta-Help	22	11	5	12	63
3	CC-CE	16	10	5	11	52

表 3 多晶体管电路信息

Table 3 Information for circuits including multiple transistors.

			Best ordering			Worst ordering		
			SDD	Time (s)	Memory(M)	SDD	Time (s)	Memory (M)
1	CC-CB	Q1 & Q3	18	0	0	32	0	0
		Q1	18	0	0	22	0	0
		Q3	18	0	0	20	0	0
2	Beta-Help	Q5 & Q7	48	0	0	60	0	0
		Q5	48	0	0	49	0	0
		Q7	48	0	0	51	0	0
3	CC-CE	Q1 & Q2	46	0	0	91	0	0
		Q1	55	0	0	58	0	0
		Q2	46	0	0	53	0	0

表 4 多晶体管的模型混合排序的最优和最差排序的仿真对比

Table 4 Contrast of the best and worst ordering for multiple transistors.

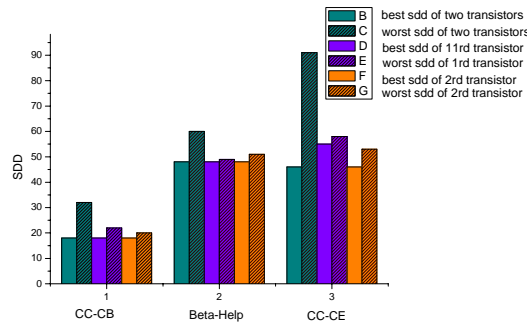


图 56 多晶体管模型混合排序对比图

Fig. 56 Contrast of the best and worst ordering for multiple transistors.

图 56 将表 4 以图形化显示，横坐标对应表 4 的行号，即对应图 56 所示的三种晶体管电路。每个电路都对应了三个实验（实验分别对应图 56 中三种颜色的柱状图）：两个晶体管中变量混合排序实验（绿色）、将第一个晶体管中变量混合排序且第二个晶体管随机排序实验（紫红色）、在第一个晶体管最优变量排序的基础上第二个晶体管的混合排序实验（橘黄色）。该图可以清楚说明，最优或比较好的变量排列顺序一定不存在与两个晶体管混合排序的网表中。因此，将晶体管看做整体，分别对每个晶体管进行排序，可以很好的获得最优或比较好的排序结果。

本节实验数据说明：首先，考虑相邻晶体管的变量排序时，将晶体管的小信号模型看做一个复合型元件，即不考虑两个晶体管的混合排序。该实验数据缩小了变量排序实验的测试案例规模，简化了实验数据的分析；为后面的变量排序实验提供依据，能保证找到的变量排序是比较优秀的变量排序；更大的优点是，对于含有 MOS 管的模拟电路分析（MOS 含有内部结点），需要分析的变量个数仅为所有 MOS 之和（MOS

管的问题规模远远大于电路中的补偿电容、补偿电阻和耦合电容), 同时不需要考虑每个小信号模型内部结点, 从而减小电路的分析复杂度。

### 5.1.4 两个功能模块的混合变量排序

对图 51 所示的双端转单端的电路进行分析。图中 M0 和 M3 的栅极相连, 其作用就是为电路提供直流偏置电流, 为了排除镜像电流源的干扰, 在测试电路中直接将 M0 和 M3 从网表中删去, 且此操作不会影响电路功能的仿真。

对双端转单端的电路 (图 51) 作简单分析:

- 1) 模块 A: 包括晶体管 M1、M5、M8 和 M13, 不包括 M0;
- 2) 模块 B: 包括晶体管 M2、M6、M9 和 M16, 不包括 M3;
- 3) 模块 C: 包括晶体管 M11、M10、M4 和 M7;
- 4) 模块 D: 包括晶体管 M14 和 M15。

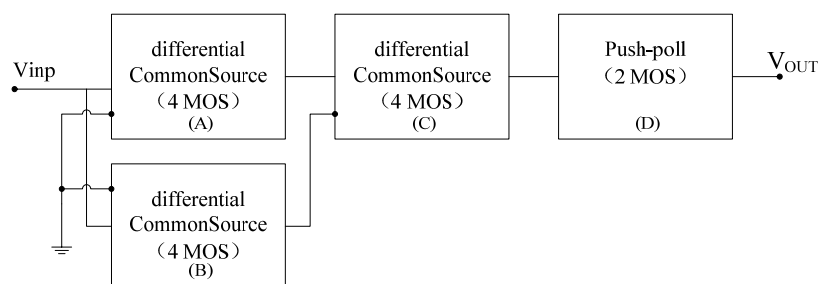


图 57 双端转单端电路的模块化示意图

Fig. 57 An example of differential to single circuit.

该电路的详细信息见表 5。首先将电路拆分为以上 4 个功能模块, 模块内部任意排序, 按照模块 DCAB 的顺序构建初始网表。下面的实验是基于此初始网表。

No.	circuit	#Edge	#Edge_lump	#Node	#Symb	#Term
1	diff2sing	128	96	39	97	3.8e+18

表 5 双端转单端的电路详细信息

Table 5 Information of a differential to single circuit.

		Best ordering			Worst ordering		
		SDD	Time(s)	Memory(M)	SDD	Time(s)	Memory(M)
1	C & D 模块	8426	2.3	53.2	16769	6.3	96.3
	D 模块	8426	2.8	47.1	8426	2.3	47.1
	C 模块	8426	2.9	59.4	10056	3.1	63.5

2	A & B 模块	5110	2.0	41.0	81475	9.9	202.8
	A 模块	7362	2.8	57.3	10099	3.4	69.6
	B 模块	5010	2.6	53.2	10797	2.9	59.4

表 6 双端转单端电路的两个模块混合排序的最优和最差排序的仿真对比  
Table 6 Contrast of the best and worst ordering for a differential to single circuit.

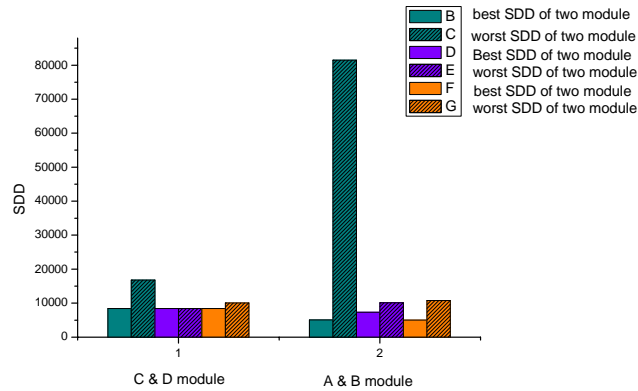


图 58 相邻的两个模块的混合排序和单个模块内部排序对比  
Fig. 58 Contrast of the best and worst ordering for two adjacent modules.

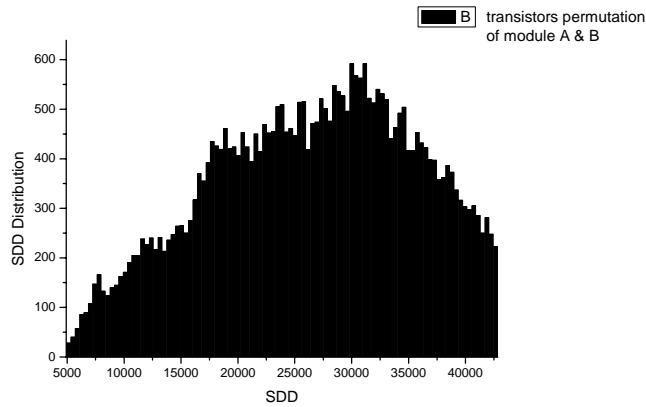


图 59 模块 A&B 混合排序的 SDD 规模分布图  
Fig. 59 SDD distribution of hybrid ordering of A & B modules.



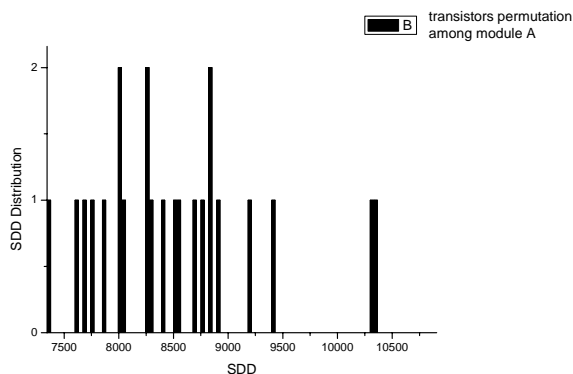


图 60 模块 A 排序的 SDD 规模分布图

Fig. 60 SDD distribution for all permutation of module A.

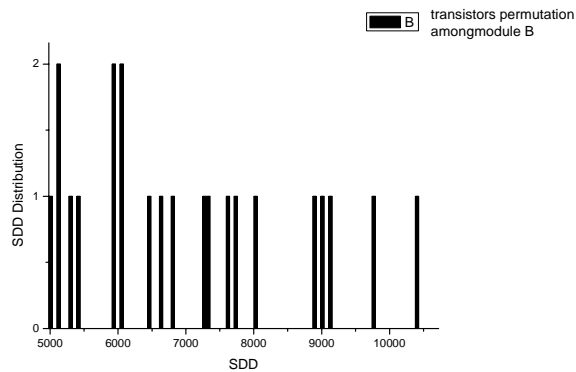


图 61 模块 B 排序的 SDD 规模分布图

Fig. 61 SDD distribution for all permutation of module B.

将单个晶体管看做整体，忽略晶体管内部排序的情况下：

- 1) 如图 60、61，模块内部的变量处理顺序对仿真器的效率影响不大。当考虑模块内部晶体管之间排序时，最优和最差的网表排序的 SDD 的规模很接近。
- 2) 如图 60、61，相邻的两个模块的最优排序很可能是两个模块内部排序、模块间排序的结果，即肯定不会是两个相邻晶体管混合排序的情况。即将电路按照功能模块划分并排序，只考虑模块内部的晶体管之间的排序，以及模块之间的排序。例如，图 57 中将整个电路分成 4 个模块，则只考虑 4 个模块的排序（共有  $4! = 24$  种可能性）。假如按照信号从输出到输入的顺序进行模块间排序，则模块处理顺序为 D-C-A-B 或 D-C-B-A。两种排序组合生成 SDD 的个数很接近。

- 3) 图 59、60 和 61 对比，模块间混合排序的最差符号处理顺序往往比模块内排序要差很多倍，说明最优变量处理顺序很可能存在于模块内符号排序；同时模块内部排序的 SDD 分布较为平均，模块内的晶体管处理顺序对仿真器影响不大。

### 5.1.5 所有晶体管模型的内部变量排序

以图 49 所示的三级运放电路进行仿真。首先将整个电路划分成差分放大模块 A、差分放大模块 B 和 push-poll 模块 C，并按照从输出到输入的信号传递方向对网表进行排序，得到一个比较好的符号排序。本章节的目的是考察每个晶体管内部排序对仿真器的影响有多大。实验就是依次对每个晶体管进行测试，测试的网表应当保持该晶体管和其它晶体管的相对位置未变；例如，实验 1 测试 M1 晶体管的全排列网表，其中 M1 和其它晶体管的相对位置未变；实验 2 的初始网表是实验 1 中的最佳网表；依次类推，最后实验得到的最佳排序网表已经非常接近最佳变量排序。

No.	circuit	#Edge	#Edge_lump	#Node	#Symb	#Term
1	Mos_11	106	79	34	80	3.70e+15

表 7 三级运放的电路信息

Table 7 Information of a three-stage OPAMP (including 11 mos transistors).

Mos_11		Best ordering		Worst ordering	
		SDD	Reduced SDD	SDD	Reduced SDD
1	M1	9949	5190	10565	5309
2	M2	9610	5059	10515	5313
3	M4	9610	5054	10004	5292
4	M3	9540	5013	9673	5041
5	M11	9528	5011	10731	5095
6	M6	9517	5000	15035	5585
7	M5	9303	4891	12523	6124
8	M8	9303	4891	10717	5581
9	M7	9303	4891	9778	4983
10	M10	9303	4891	9749	5079
11	M9	9299	4890	9399	4932

表 8 多晶体管的模型混合排序的最优和最差排序的仿真对比

Table 8 Contrast of the best and worst ordering among model for three-stage OPAMP.

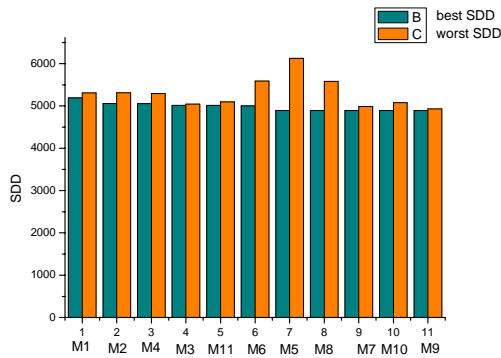


图 62 晶体管内部变量排序对比

Fig.62 Contrast of the best and worst ordering among model.

从图 62 可以看出，考虑晶体管内部的变量排列顺序对该网表的性能提升不大。其主要原因有：不考虑晶体管内部排序的情况下，只考虑晶体管间的变量排序已经非常接近最优变量排序；MOS 晶体管很紧凑，同时有两个内部结点，其模型本身也比较复杂，通过模型内部排序提升仿真器性能的空间有限。因此，对于含有 BJT 和 MOS 小信号模型的电路，可以不考虑模型内部的变量排序，仅仅考虑模型间的变量排序已经可以很好的提升仿真器性能。

### 5.1.6 晶体管与相邻电阻、电容的混合变量排序

以图 49 为测试对象，测试的初始网表考虑功能模块间排序和模块内部的晶体管之间排序得到的网表。该网表中存在五个补偿电阻和电容，功能模块划分时将其并入相连的功能模块。该实验就是测试四个补偿电阻和电容与第一个模块中两个晶体管的排序情况。该实验数据主要考察晶体管和相连的补偿电阻和电容的符号处理顺序。表 7 中列出了电路的详细信息。表 9 中保存了晶体管和相邻电阻、电容的排序数据。

从实验数据可以说明：

- 1) 补偿电阻和电容对仿真器效率的影响很小，一般不超过 8%。
- 2) 排序时一般不考虑补偿电阻和电容，只考虑电路中存在的晶体管。
- 3) 较好的符号处理顺序是优先处理晶体管，再处理相邻的补偿电阻和电容。

		Best ordering			Worst ordering		
		SDD	Time(s)	Memory(M)	SDD	Time(s)	Memory(M)
1	M11 Cc1 Rc	5197	2.15	51.2	5278	2.20	55.3

2	Cc1 Rc	5197	2.15	51.2	5197	2.15	51.2
3	M12 Rc2 Cc2 C1	5150	2.13	51.1	5565	2.52	59.4
4	Rc2 Cc2 C1	5197	2.15	51.2	5254	2.19	55.3

表 9 补偿电容和电感与相邻晶体管排序的最优和最差排序的仿真

Table 9 Contrast of the best and worst resistors, capacitance and mos transistor.

## 5.2 较大规模模拟电路的仿真结果

前面的测试多是用穷举法来找到较小规模电路的最优变量排序，并总结某些拓扑结构电路的特点。但是遇到较大规模的电路，由于电路中包含较多的符号变量，而电路生成项的个数和电路中的符号数成指数倍关系，通常仿真器运行效率低，或者由于计算机内存的限制而无法正确得到仿真结果。对于较大规模的模拟电路就不可能用穷举法来获得最优的变量排序。因此，对于较大规模电路，只能通过启发式排序算法来提高仿真器效率。

本节测试了较大规模的模拟电路，并比较了随机排序以及启发式排序对仿真器性能的影响，表 1 列举了测试的网表详细信息，表二对比了随机排序以及启发式排序对仿真器性能的影响。Edge\_lump 指并联预处理之后边的个数；SDD 指哈希共享之后生成的 SDD 的结点个数；term 指传输函数表达式中最小生成项的个数。

实验数据表明，当电路中晶体管个数较大时，随机的排序很难保证仿真器快速运行，甚至无法得到正确的结果；对于某些电路，启发式排序算法可以很明显地提高仿真器的效率。

	#Edge	#Edge_lump	#Node	#Symb	#Term
Ua741	160	80	24	81	1.73e+05
Ua725	166	97	31	98	5.09e+17
mos_11	106	79	34	80	3.70e+15
Mos_15	142	105	44	106	4.30e+20
rail-to-rail	186	142	54	143	6.31e+26

表 10 大规模电路的测试信息

Table 10 Information for large scale analog circuit.

	Random ordering			Heuristic ordering		
	SDD	Time(s)	Memory(M)	SDD	Time(s)	Memory(M)
ua741		X		7258	1.922	30.72
ua725		X		60816	20.114	311.3
Mos_11	177736	8.6	90.1	4890	2.112	43
Mos_15		X		5449	3.03	61.4
rail-to-rail		X		770704	59.914	1212.4

表 11 启发式排序与随机排序仿真对比

Table 11 Contrast of heuristic ordering and random ordering for GRASS simulator.

注：X 表示无法完成的仿真。

### 5.3 小结

借助于 Perl 脚本，本章节通过不断生成新的重新排序的网表完成了大量网表数据的测试，并根据电路的拓扑连接关系总结电路功能模块与变量排序算法之间的关系。在总结经验之后，提出以单个晶体管为基本单位、将电路划分成不同功能块、参考小信号流动方向，将整个电路重新排序；仿真器读入重新排序的网表，并对之进行处理，从而提高仿真器性能。

与随机的变量排序相比，启发式变量排序算法可以很明显的提高仿真器的性能。

---

## 参 考 文 献

- [1] 周良柱. VLSI 与数字信号处理系统设计. 国防科技大学出版社. 1990 (1): 1~3.
- [2] P. Wambacq, G. Gielen, and W. Sansen, "A cancellation-free algorithm for the symbolic simulation of large analog circuits," in IEEE Int'l Symposium on CAS, 1992, pp. 1157-1160.
- [3] G. Gielen, P. Wambacq, and W. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," in Proc. , 1994, pp. 287-304.
- [4] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," in IEEE Trans. Computers, 1986, vol. 37, pp. 677-691.
- [5] W. Sansen, G. Gielen, and H. Walscharts, "A symbolic simulator for analog circuits," in Proc. Int'l on SSCC, 1989, pp. 204-205.
- [6] G. Gielen, H. Walscharts, and W. Sansen, "ISAAC: A symbolic simulator for analog integrated circuits," in IEEE J. of Solid-State Circuits, 1989, vol. 24, no. 6, pp. 1587-1597.
- [7] F. FEMANDEZ, A. RODRIGUEZ-VAZQUEZ, J. HUERTAS, A tool for symbolic analysis of analog integrated circuits including pole/zero extraction, in Proc. ECCTD, 1991, pp. 752-761.
- [8] F. FEMANDEZ, A. RODRIGUEZ-VAZQUEZ, J. HUERTAS, Interactive ac modeling and characterization of analog circuits via symbolic analysis, Kluwer J. Analog Integrated Circuits and Signal Process, 1991, vol. 1, pp. 183-208.
- [9] S. Seda, M. Degrauwe, and W. Fichtner, "A symbolic analysis tool for analog circuit design automation," in IEEE Int'l. Conference on CAD, 1988, pp. 488-491.
- [10] S. Seda, M. Degrauwe, and W. Fichtner, "Lazy-expansion symbolic expression approximation in SYNAP," in Proc. IEEE/ACM conference on CAD, 1992, pp. 310-317.
- [11] S. Manetti, "New approaches to automatic symbolic analysis of electric circuits," in IEE Proc. G on Devices and Systems, 1991, vol. 138, pp. 22-28.
- [12] G. Wierzba et al., "SSPICE-A symbolic SPICE program for linear active circuits," in Proc. Midwest Symposium on CAS, 1989, vol. 2, pp. 1197-1201.
- [13] A. Konczykowska and M. Bon, "Automated design software for switched-capacitor IC's with symbolic simulator SCYMBAL," in Proc. on DAC, 1988, pp. 363-368.
- [14] M. Hassoun and P. Lin, "A new network approach to symbolic simulation of large-scale networks," in Proc. Int'l Symposium on CAS, 1989, pp. 806-809.

- 
- [15] L. Huelsman, "Personal computer symbolic analysis programs for undergraduate engineering courses," in Proc. Int'l Symposium on CAS, 1989, pp. 798-801.
- [16] G. Gielen, "Symbolic analysis methods and applications – an overview," in Proc. Int'l Symposium on CAS, 1992, vol. 3, pp. 1141-1144.
- [17] PSPICE
- [18] Z. Yin, "Symbolic network analysis with the valid trees and the valid tree-pairs," in IEEE Int'l Symposium on CAS, vol. 5, pp. 335-338, 2001.
- [19] X. Li and Z. Yin, "The algorithm and program scheme to find out all valid trees and valid tree-pairs," in Proc. Int'l Conference on ASIC, vol. 1, pp. 298-301, 2003.
- [20] W. Chen and G. Shi, "Implementation of a Symbolic Circuit Simulator for Topological Network Analysis," in IEEE Conference on CAS, 2006, pp. 1368-1372.
- [21] G. Shi and W. Chen, "A Graph Reduction Approach to Symbolic Circuit Analysis," in Asia and South Pacific Symposium on DAC, 2007, pp. 197-202.
- [22] Z. Yin, "Symbolic network analysis with the valid trees and the valid tree-pairs," in IEEE Int'l Symposium on Circuit and System, 2001, vol. 5, pp. 335-338.
- [23] X. Li and Z. Yin, "The algorithm and program scheme to find out all valid trees and valid tree-pairs," in Proc. Int'l Conference on ASIC, 2003, vol. 1, pp. 298-301, 2003.
- [24] J. Starzyk and A. Konczykowska, "Flowgraph analysis of large electronic networks," IEEE Trans. on CAS, 1986, vol. 33, no. 3, pp. 302-315.
- [25] G. Minty, "A simple algorithm for listing all the trees of a graph," IEEE Trans. on Circuit Theory, 1965, vol. 12, p. 120-120.
- [26] S. B. Akers, "Binary decision diagrams," IEEE Trans. on Computers, 1976, vol. 27, pp. 509-516.
- [27] C. Lee. Representation of switching circuits by binary-decision programs. Bell System Technical Journal, 38:985-999, July, 1959.
- [28] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," in IEEE Trans. Computers, 1986, vol. 37, pp. 677-691.
- [29] K. race, R. Rudell, and R. Bryant, "Efficient implementation of a BDD package," in Proc. IEEE/ACM on DAC, 1990, pp. 40-45.
- [30] L. Chua and P. Lin, Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques. Englewood Cliffs, NJ: Rentice-Hall, 1975, ch. 14.
- [31] P.L. R. GRAY, P. J. HURST, S. H. LEWIS, R. G. MEYER, Analysis and Design of Analog Integrated Circuits, 4th edition, pp 174, John Wiley & Sons, 2001.
- [32] F. V. Hunt and R. W. Hickman. "On Electronic Voltage Stabilizers," in Review of Scientific Instruments, 1939, vol. 10, pp. 6-21.
- [33] H. Wallman, A. B. Macnee, and C. P. Gadsden, "A Low-Noise Amplifier," Proc. of the IRE, 1948, vol. 36, pp. 700-708.

---

## 致 谢

硕士研究生阶段的学习是快乐的、辛苦的、丰富的以及值得我一生去回忆和喝彩的时光。我很庆幸自己在交通大学读研究生，在这里我明白做人勤奋、做事踏实、饮水思源、爱国荣校的道理。时光如梭，转眼间两年半的研究生的学习生涯就已接近尾声。

首先要感谢的就是我亲爱的爸爸妈妈。无论在什么时候父母都给我无限支持和无微不至的关怀，他们给了我充分的自由空间去选择发展方向。开心的时候父母和我一起分享快乐，当遇到困难和挫折的时候，父母永远是我的避风港，我在他们的理解、支持中重新获得自信和勇气，积极克服困难，乐观面对学习和生活。

学习和研究的工作中我最要感谢的是我硕士阶段的指导老师施国勇教授。您的悉心指导和帮助让我很快明确了自己研究的方向；您严谨治学、谦虚做人和实事求是的精神给我们树立了榜样，我深刻认识到作学问和做人同样都要勤奋刻苦、踏踏实实；您宽广的学识、过人的智慧以及快捷的思维，帮助我们寻找正确的解决问题的方向；您时时刻刻都在强调编程和实际解决问题的能力，在您的指导下，我的编程能力有了极大的提高，同时也深深爱上了计算机编程语言。再要感谢的是李章全老师，您的讲课幽默风趣，将晦涩难懂的知识讲解的幽默风趣，让大家在很轻松的状态下学习课本知识；您的个人经历也具有传奇色彩，听您说在美国的学习和生活也让我们开阔了眼界，明白了精彩的人生应当是学习和工作同样出色。同时也要感谢付宇卓教授、赵峰、谢景等老师，你们的教学为我打下了扎实的知识基础。另外，学院的胡薇薇老师、杨薛雯老师、郁美娟老师、陆谨老师等，感谢你们平时的辛勤的行政工作，为我们创造了适合学习和生活的环境。

不得不提的还有我们 07 级微电子学院的所有同学：有艺术天赋的穆静、乐观勇敢的陈姗姗、聪明睿智的赵丽丽、天真可爱的杨玲、美丽大方的刘平平、女强人的肖怡静、口才绝佳能神侃的曾媚、幽默风趣的陶文清、办事利索的陆秀丽、非常聪明的强倩、学习能力强的陈硕、善于思考问题的刘俊、非常酷的庄芙蓉、很酷的帅哥谭焜元、心地善良乐于助人外加编程能力极强的李骥、聪明勤奋的陈安、优雅待人和气的戴焱、乐于助人的韩世英、低调的聪明的谢边村、乐观开朗的陈还等等一班同学，虽然大家专业不同，但是非常开心大家能在一个学院学习，一同进步，友谊弥足珍贵。

思源湖畔的学习即将接近尾声，但是精彩的人生才刚刚开始。饮水思源，爱国荣校，交通大学作为国内的一流学府，培养了许多优秀的人才，我也很开心能和钱学森等世界知名人士成为校友；交大的学习氛围和身边牛人的谦虚态度，让我认识到宁静致远、勤奋学习、谦虚做人、踏实做事是一种美德。我满怀期待的憧憬未来，追求卓越，勤俭谦诚，描绘精彩人生，为个人、家庭、学校和祖国争得荣誉。



## 攻读硕士学位期间已发表或录用的论文

- [1] 王婷. 已经发表一篇学术论文. 启发式符号排序改进符号化模拟电路仿真器性能. 信息技术.  
(已录用)