

上海交通大学硕士学位论文

模拟集成电路符号化低阶模型自动生成方法与应用¹

硕 士 研 究 生:

学 号:

导 师:

专 业: 电子科学与技术

所 在 单 位: 微纳电子学系

答 辩 日 期: 2016 年 1 月

授予学位单位: 上海交通大学

¹本研究由国家自然科学基金（项目号 61176129, 61474145）资助。

Dissertation Submitted to Shanghai Jiao Tong University
for the Degree of Master

Automatic Generation Method of Low-order Models for Analog Integrated Circuits with Applications

Candidate:

Student ID:

Supervisor:

Academic Degree Applied For: Master of Science

Speciality: Electronic Science and Technology

Affiliation: Dept. of Micro-Nano Electronics

Date of Defence: Jan, 2016

Degree-Conferring-Institution: Shanghai Jiao Tong University

模拟集成电路符号化低阶模型自动生成方法与应用

摘 要

如今，模拟集成电路设计仍然在很大程度上依赖于手工分析，这严重阻碍了 SoC 的整体设计进程。作为模拟集成电路设计中不可或缺的一环，可靠的宏模型缺乏系统化自动化的设计流程。本文讨论了电路拓扑结构和双图决策树（Graph Pair Decision Diagram, GPDD）符号化方法之间的联系，并提出拓扑简化的想法，开发了一套行之有效的自动化低阶模型生成算法。这套算法生成的简化符号化模型与原始电路无论在性能上，还是拓扑结构上都有很好的匹配。算法的有效性和鲁棒性得到了不同电路结构、不同电路元件尺寸的广泛验证。

本论文将提出的模型生成方法应用至两方面。其一是通过在简化电路模型添加电流限制，来实现自动化构造大信号分析中用到的时域模型，从而进一步分析其转换速率。另一个应用是使用符号化多端口构造方法下的敏感度计算用于分析电路的共模抑制比和电源抑制比。本文给出了相关的测试结果和算法性能结果。

关键词：低阶模型生成 拓扑简化 双图决策树（GPDD） 时域模型 共模抑制比（CMRR） 电源抑制比（PSRR） 符号化分析（Symbolic analysis）

Automatic Generation Method of Low-order Models for Analog Integrated Circuits with Applications

ABSTRACT

Nowadays, analog integrated circuit (IC) design still depends heavily on manual analysis, which greatly impedes the entire SoC design process. As one indispensable part of analog IC design, reliable macromodel for analysis lacks systematic and automatic generation procedure. This thesis discusses the relation between circuit topology and a symbolic method call Graph-Pair Decision Diagram (GPDD). The topological simplification is proposed with an efficient symbolic algorithm for automatic low-order model generation. According to the experimental results, the simplified model matches the original circuits in performance and topology to good agreement. The validity and robustness of proposed method have been verified by circuits with various topologies and sizings extensively.

The proposed model generation method has been applied to two scenarios. The first problem is to automatically construct the time-domain model for large-signal analysis, and the typical slew-settling behavior can be captured by the proposed model. The other one is to analyze common mode rejection ratio (CMRR) and power supply rejection ratio (PSRR) with symbolic calculation and sensitivity analysis under multi-port construction. In the thesis, relevant test results and algorithm performance are provided.

KEY WORDS: Low-order model generation, topological simplification, graph-pair decision diagram (GPDD), time-domain model, common mode rejection ratio (CMRR), power supply rejection ratio (PSRR), symbolic analysis

目 录

摘 要	i
ABSTRACT	ii
目 录	iii
图 录	vi
表 录	vii
第一章 绪论	1
1.1 模拟集成电路设计方法概述	1
1.2 符号化分析方法	2
1.2.1 符号化分析方法回顾	3
1.2.2 双图决策树 (GPDD) 符号化分析方法	4
1.3 电路模型符号化简化方法简介	7
1.4 课题研究内容与意义	8
1.5 主要贡献与章节安排	8
第二章 基于 GPDD 的电路简化原理	9
2.1 GPDD 中的二元操作	9
2.2 GPDD 与电路拓扑结构的对应关系	11
2.3 多端口构造原理	14
2.4 符号化敏感度分析方法	15
2.5 本章小结	18
第三章 模拟集成电路模型降阶方法	19
3.1 元件重要性选取方法	19
3.2 宏模型自动生成方法	21
3.2.1 元件预处理过程	22
3.2.2 简化特殊情况分析	24

3.3	时域宏模型符号化简化模型自动生成方法	26
3.3.1	传统的时域模型分析方法	26
3.3.2	符号化时域简化模型分析方法	28
3.4	降阶模型生成算法测试结果与分析	31
3.4.1	预处理简化降阶电路模型区别	32
3.4.2	多种电路结构简化结果分析	34
3.4.3	尺寸调整下的算法稳定性分析	40
3.4.4	简化符号化模型阶数比较	43
3.4.5	符号化时域简化模型测试结果	44
3.5	本章小结	44
第四章	多端口构造符号化约简方法探索	46
4.1	共模抑制比与电源抑制比介绍	46
4.2	CMRR 和 PSRR 在多端口构造下的计算方法	47
4.3	共模抑制比与电源抑制比测试结果	49
4.3.1	多端口构造方法的双图决策树的时间空间复杂度比较	49
4.3.2	敏感度方法进行电路参数优化共模电源抑制比	51
4.4	本章小结	53
第五章	结束语	54
5.1	主要工作与创新点	54
5.2	后续研究方向	55
附录 A	双图决策树对消项说明	56
A.1	符号化对消项概念介绍	56
A.2	双图决策树无加性对消项证明	56
A.3	双图决策树公因子对消项情况	57
A.3.1	公因子对消项反例	58
A.3.2	对自动低阶模型生成中电路图约减过程的影响	58
	参考文献	60
	致 谢	65
	攻读学位期间发表的学术论文	66

图 录

1-1 基本模拟集成电路设计流程	2
1-2 符号化分析流程	3
1-3 RC 低通滤波电路及对应原始图	5
1-4 RC 低通滤波电路生成树对	5
1-5 RC 电路的 GPDD 结构示例	6
1-6 GPDD 求值规则	6
2-1 导纳在极限取值情况下的拓扑结构改变	9
2-2 Nullor 元件符号	10
2-3 符号极限取值情况下 GPDD 的计算方法	11
2-4 RLC 电路示意图	13
2-5 符号极限取值情况下 GPDD 结构示例	13
2-6 多端口电路示意	14
2-7 多端口电路的 GPDD 结构	14
2-8 用于敏感度说明的两级运放电路	16
3-1 元件变化对运放增益和相位裕度的影响	20
3-2 元件重要性计算流程	20
3-3 降阶模型自动生成整体流程	22
3-4 虚地的产生原因	23
3-5 电路图约减过程中特殊情况说明	25
3-6 Slew-Settling 过程	26
3-7 运放时域宏模型	27
3-8 时域模型中的电流限制位置	29
3-9 Slew-settling 过程中输入管 g_m 的变化	29
3-10 S 型函数族	31
3-11 MOSFET 器件小信号模型	31
3-12 两级运算放大器电路图	32
3-13 无虚拟地的两级运放简化小信号电路	33
3-14 无虚拟地的两级运放频率响应曲线比较	33

3-15 有虚拟地的两级运放简化小信号电路	34
3-16 有虚拟地的两级运放频率响应曲线比较	35
3-17 折叠共源共栅运算放大器电路图	35
3-18 折叠共源共栅运放简化小信号电路	36
3-19 折叠共源共栅运放频率响应曲线比较	37
3-20 折叠共源共栅运放降阶模型中的 Nullor 等效	37
3-21 Voltage Buffer 补偿的两级运算放大器电路图	38
3-22 Current Buffer 补偿的两级运算放大器电路图	39
3-23 Voltage Buffer 补偿的运放简化小信号电路	39
3-24 Voltage Buffer 补偿的运放频率响应曲线比较	40
3-25 Current Buffer 补偿的运放简化小信号电路	40
3-26 Current Buffer 补偿的运放频率响应曲线比较	41
3-27 1 号尺寸调整方案的的运放简化小信号电路	42
3-28 2 号尺寸调整方案的的运放简化小信号电路	42
3-29 两级运放模型的 Slew-Settling 时域仿真结果	44
3-30 折叠共源共栅运放模型的 Slew-Settling 时域仿真结果	45
4-1 共模抑制比及电源抑制比定义	46
4-2 多端口方法构造 CMRR 及 PSRR 的 GPDD 结构	48
4-3 两级运放的 CMRR 及 PSRR 的频率响应结果	49
4-4 折叠共源共栅运放的 CMRR 及 PSRR 的频率响应结果	50
4-5 两级运放针对 W_1 的敏感度分析结果	51
4-6 两级运放针对 W_6 的敏感度分析结果	52
A-1 存在公因子对消项的示例电路	58
A-2 存在公因子对消项的示例电路	58

表 录

2-1	电路元件极限取值的拓扑结构	10
3-1	两级运放预处理计算各节点 AC 相应	32
3-2	折叠共源共栅运放预处理计算各节点 AC 相应	36
3-3	两级运放尺寸共同参数	41
3-4	两级运放的不同尺寸方案	41
3-5	不同尺寸下的两级运放元件重要性排序	43
3-6	电路符号化模型阶数比较	43
4-1	两级运放的单独构造与多端口构造的时空性能比较	49
4-2	折叠共源共栅运放的单独构造与多端口构造的时空性能比较	50
4-3	W_1 的 DC 敏感度分析结果及优化	52
4-4	W_6 的 DC 敏感度分析结果及优化	52

第一章 绪论

本章首先介绍模拟集成电路设计的概况,指出其设计过程中由于计算机辅助设计的缺失带来的设计效率低下的问题。进而回顾符号化分析方法在这一问题上所作出的诸多努力,并简要介绍并比较相应成果。最后介绍本文的主要内容,并给出文章组织安排。

1.1 模拟集成电路设计方法概述

随着半导体产业的不断发展与电子设备的不断的推陈出新,对集成电路设计的时效性及高效性提出了更高的要求。同时,越来越多的功能被集成在同一块芯片内部,从而推动了 SoC (System-on-Chip) 概念的形成,对电路设计带来更多困难与挑战 [1]。作为集成电路设计的两大重要分支,数字集成电路设计和模拟集成电路设计的设计方法决定了 SoC 设计效率,同时保证了电路性能与可靠性。其中,由于电子设计自动化 (Electronic Design Automation, EDA) 工具的成熟,数字集成电路设计的效率已远远超过模拟电路设计 [2]。数字电路在整个设计流程,如电路综合、验证、自动布局布线等,均有对应的 EDA 工具做相应的支持。除此之外,数字集成电路中由于有 IP (Intellectual Property) 核的支持,可以轻松地对功能模块进行重用,大大方便了数字电路工程师进行电路设计。然而相对应的,模拟电路由于其电路性能特征描述的困难以及针对不同应用设计变化复杂等原因一直导致其 IP 化进程并不十分顺利 [1, 3]。

模拟集成电路在现代 SoC 设计中基本只占这个芯片面积的 10%-50%,然而这个设计流程中有近 50%-90% 的时间用于模拟部分的设计与测试 [4]。而且往往由于模拟电路的设计错误,可能导致多次流片验证,增加了设计成本。模拟集成电路自与 1964 年诞生以来 [5],基本一直遵循图1-1中的设计流程。电路设计往往从整体系统建模开始,然后根据系统的整体性能指标,给出更具体的电路功能模块的性能指标。然后,参照模块性能,选择合适的电路结构,并结合电路仿真结果,确定元件参数。综合所有功能模块,在完成系统层面仿真。最后,手工绘制版图后,并检查无误后,送至工艺厂进行生产。由于上述设计流程限制,模拟电路设计往往受到以下条件制约:

- a) 上述大部分流程均为人工计算设计得到,缺乏计算机辅助设计流程,大部分情况下需要工程师的个人经验作为设计依据。
- b) 上述流程往往需要多次反复,根据仿真结果一再调整结果,带来设计效率的低下。

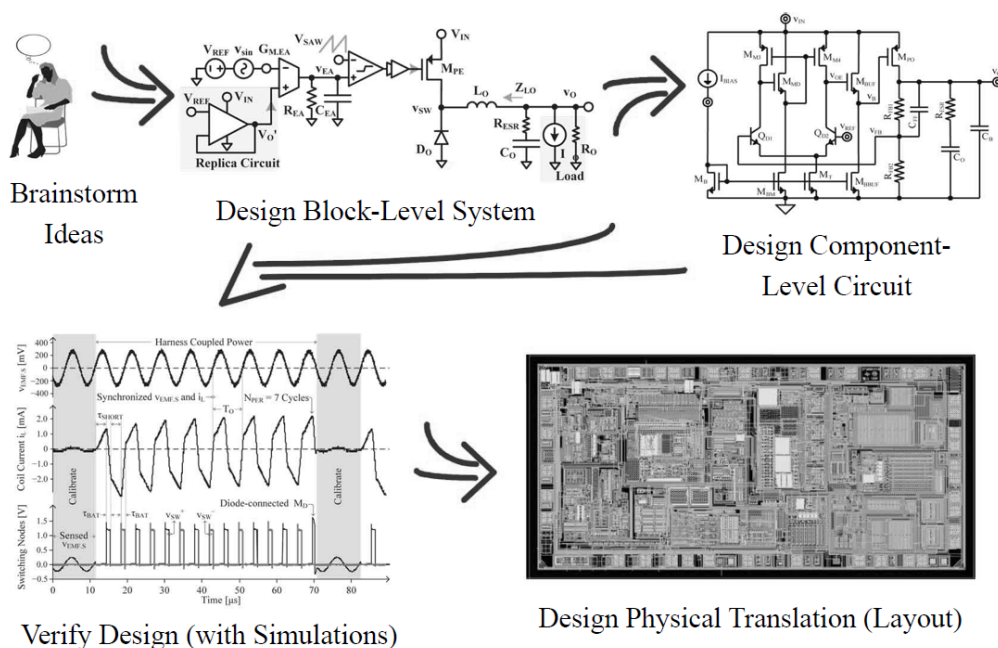


图 1-1 基本模拟集成电路设计流程 [4]

Fig 1-1 Basic analog integrated circuits design procedure [4]

c) 模拟电路仿真往往花费大量时间，特别是在系统级的仿真中，任何参数的调整带来巨大的时间损耗，拉长了整个设计周期。

可以看到，上述原因主要是由于模拟电路设计缺乏计算机辅助设计工具，导致设计方法因人而异，没有形成系统化的设计体系所造成的。其中，模拟电路的建模是关键的一环。如果存在系统化自动化的模拟电路建模方法，那么首先由于模型的建立方式的统一，会带来设计人员学习理解的方便。同时，由于所建立的电路模型中可以带有电路元件参数，直接将电路性能与电路元件取值联系在一起，方便了电路设计的优化流程。另外，自动化建模可以在系统仿真层面大大加快设计效率，提升仿真速度。故可以看到提出行之有效的系统化自动化模拟电路建模方案将在一定程度上有效地解决上述三个制约条件，对加快电路设计效率有着至关重要的作用。

1.2 符号化分析方法

目前，模拟电路仿真主要采用以 SPICE 为基础的数值化仿真工具 [6]，如 Cadence 公司的 Spectre 以及 Synopsys 的 HSPICE 仿真器等。数值仿真器以其稳定高效的数值求解结果成为业界的常用仿真工具。然而数值仿真方法最大的问题在于其仿真结果与电路元件参数的取值脱节，很难从电路的仿真结果对电路元件的取值反推相应的结果。这导

致电路工程师，特别是经验不丰富的工程师，在不确定电路元件与性能关系的情况下，只能对电路元件取值进行盲目的调整，以期达到所要求的性能指标。为了克服这一困难，本文采用符号化仿真方法对电路进行分析。

1.2.1 符号化分析方法回顾

上世纪 90 年代，另一种电路仿真方法符号化仿真方法得到了许多关注 [7]。与数值化仿真不同，符号化仿真方法会直接建立小信号电路传输函数与电路元件之间的表达式，然后通过所求的电路传输函数表达式进行求解，如图 1-2 所示的流程。

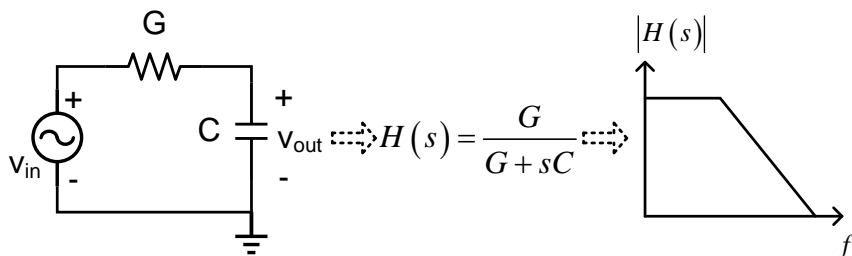


图 1-2 符号化分析流程

Fig 1-2 General symbolic analysis procedure

这种电路分析方式带来诸多数值化仿真不能得到的好处，主要可以归纳为以下几点：

- a) 可以直观地通过符号化公式告知电路设计者所有电路元件参数对电路的作用，有助于工程师进行优化调整及相应的分析。
- b) 在构造解析表达式后，在电路元件重新调整后，可以直接通过得到的公式进行仿真求解，避免费时的数值矩阵迭代过程。
- c) 可以通过求解得到的公式对电路进行表征，构建相应的电路模型，用于将来的模拟电路功能模块综合、验证等功能的实现。

构建符号化表达式十分多样，过去开发了多种不同的符号化求解方式，如矩阵行列式展开求解 [8]、利用 Mason 规则求解信号流图 [9, 10]、双图枚举 [11] 等。这些算法都可以求解一个线性电路的传输函数；针对如含有 MOSFET 的非线性电路，求解器则会通过静态工作点仿真得到非线性元件的小信号模型，并带入计算。

然而，由于符号化分析方法本身性质决定，随着电路规模的增大，符号化分析中符号化项的规模也呈指数级增长，[12] 对各种不同的符号化方法中的规模增长进行了验证

与比较。这在很大程度上限制了符号化方法在大规模电路中的应用,甚至当时提出的多数算法难以对一个完整的运放电路进行符号化分析。其次多数符号化方法,与节点分析法、回路分析法等传统电路分析方法差异比较大,很多算法都存在规则繁琐,不利于手工计算的问题,所以相对其推广也更加不如数值化分析方法。

1.2.2 双图决策树 (GPDD) 符号化分析方法

为解决符号化表达式增长过快的问题,Sheldon X.-D. Tan 于 2000 年附近提出了 DDD (Determinant Decision Diagram) 符号化方法。他们使用 Cramer 法则计算电路传输函数,并利用二分决策图 (Binary Decision Diagram, BDD) 结构 [13] 来存储计算过程中得到的符号化表达式 [14]。这种方法有效地缓解了符号化公式占用大量存储空间的问题,但是其所保存的符号化公式存在非常多的对消项。(对消项会在第三章有更详细的描述。)这造成了计算机计算过程中的数值误差的累积,其求解结果有一定误差。

2006 年附近, G. Shi 等人提出了双图决策树 (Graph-Pair Decision Diagram, GPDD) 方法 [15–18]。此方法也利用了 BDD 结构,对符号化项进行隐式枚举,从而保证了与 DDD 相当的高效的空间利用率。虽然,这种方法的符号化结果规模仍然随着电路规模呈指数级增长,但是由于相较早期的符号化方法,其增长速度较慢,以足以分析一个运放这样规模的电路。同时,该方法从理论保证了对消项不会在最终得到的符号化表达式中出现,从而可以得到非常精确的计算结果。

近几年来, GPDD 方法从软件实现仿真到电路建模优化等多方面得到了许多发展。[19–21] 中使用了层次化的方法,从而实现了较大规模模拟集成电路的分析; [22–24] 中利用符号化敏感度分析方法,对电路元件敏感性进行了探索,并开发了自动调整运放尺寸的算法; [25, 26] 通过 Moment Matching 的方法对电路零极点进行建模,从而得到运放时域电路模型; [27] 实现了对开关电容电路的仿真,从而得到了 z 域的 GPDD 仿真方法; [28, 29] 进一步将其应用到 Sigma-Delta 调制器中。

下面通过一例子对这种方法进行详细介绍。

例 1.1. RC 电路的 GPDD 构造示例

考虑如图1-3所示的 RC 低通滤波器电路,这里希望求取电路的从输入电压至电容两端输出电压的传输函数。根据电路结构,构造如图中右侧所示的图,将电路元件抽象为图中的边,用电路元件的导纳值作为元件的符号。其中电路的输入输出关系,我们用从输出控制输入的受控源 X 来进行标识,并作为 GPDD 结构的根节点。这里由于是输出电压控制输入电压,故为电压控制电压源 (VCVS),在图中即为对应的紫色 VC 和 VS 边。

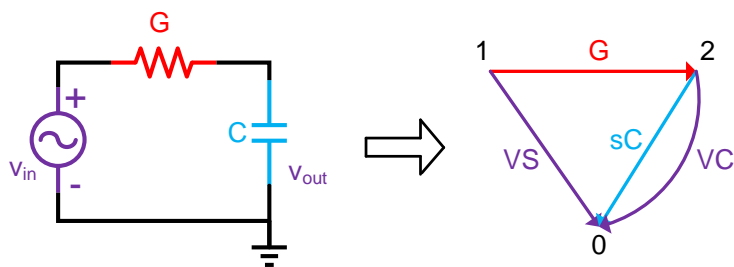


图 1-3 RC 低通滤波电路及对应原始图
Fig 1-3 RC low-pass filter & its corresponding graph

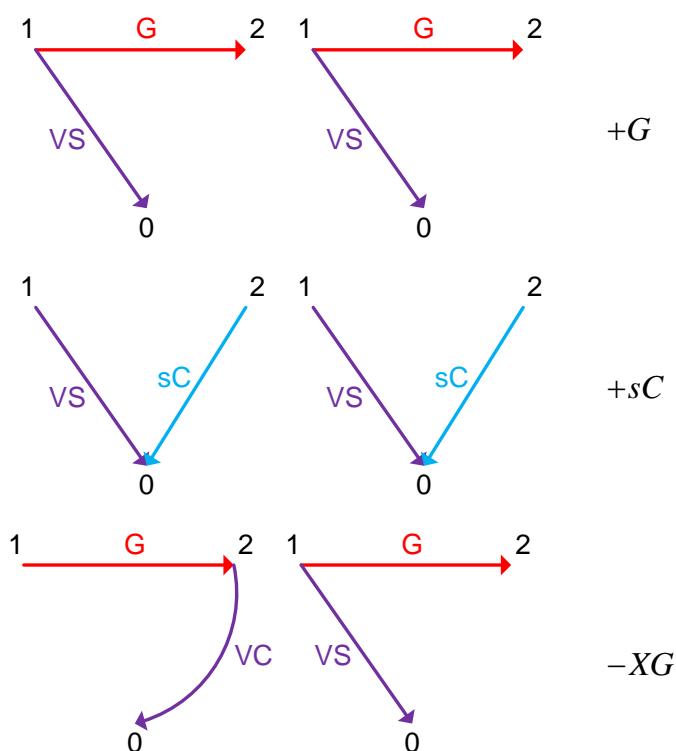


图 1-4 RC 低通滤波电路生成树对
Fig 1-4 Spanning tree pairs in RC low-pass filter

根据双图电路分析理论 [7], 可以知道在上述电路中可被接受的生成树对仅有图 1-4 中所示的三对, 同时每一个生成树对构成了一个符号化的项。如将这些项相加, 并使它等于 0, 则可求得电路的传输函数:

$$-XG + G + sC = 0 \Rightarrow H(s) = \frac{1}{X} = \frac{G}{G + sC} \quad (1-1)$$

为了能够快速枚举原始图中所有的符合条件的生成树对, GPDD 算法是用了类似图 1-5 所示的 BDD 结构来保存枚举结构 [16]。GPDD 结构每一层有且仅有一个元件符

号，每个 GPDD 节点有两个 2 儿子，左儿子用实线与下面的节点连接，右儿子用虚线与下面的节点连接。不同节点可以共享同一个儿子，这也是 GPDD 之所以可以用较少空间存储大量符号化项的原因，类似于多项式的公因式提取。所有的节点必然终结于 1 结点或 0 结点。可以看到这个 GPDD 结构中共有 3 条从根节点通往 1 结点的路径，如路径经过了实线，那么实线出发点的符号会出现在该符号项中；如是虚线，则不出现。经验证，可知道这 3 条路径正是指代前面的 3 个生成树对，从而可以保存了电路的传输函数的符号化表达。

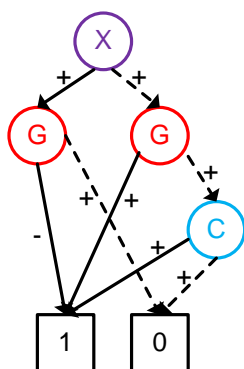


图 1-5 RC 电路的 GPDD 结构示例
Fig 1-5 GPDD structure of RC circuit

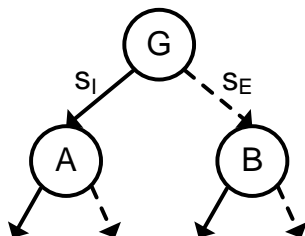


图 1-6 GPDD 求值规则
Fig 1-6 The evaluation rule for GPDD

这里简要介绍 GPDD 结构的计算方法。如图1-6所示，对于 GPDD 结构中某个节点 G 的求值，用 $f_G(G)$ 来标示。在递归求得其左右子 GPDD 结构的值后，用其符号本身 G 乘以以实线相连的 A 节点的值 $f_A(A)$ ，并加上虚线相连的 B 节点的值 $f_B(B)$ 。注意这里，需记得考虑实线及虚线上所标识的符号 s_I 和 s_E ， s_I 和 s_E 仅为正负关系。总结可如下式所示：

$$f_G(G) = s_I G f_A(A) + s_E f_B(B) \quad (1-2)$$

如使用式1-2可以验证得到式1-1的结果。根据该计算规则，可以通过自底向上遍历整个 GPDD 结构，求得电路传输函数。在根节点处，无需显式求解令方程等于 0 的传输函数值。假设根节点 X 左儿子为 N ，符号为 s_N ；右儿子为 D ，符号为 s_D 。那么电路传输函数即为：

$$s_N X f_N(N) + s_D f_D(D) = 0 \Rightarrow H(s) = \frac{1}{X} = -s_N s_D \frac{f_N(N)}{f_D(D)} \quad (1-3)$$

1.3 电路模型符号化简化方法简介

符号化方法的困难除了构造规模巨大之外，另一个影响其发展的更主要的原因是难以对生成的符号化表达进行分析。然而事与愿违，这一点正是符号化发展的初衷。可是由于符号化公式十分复杂，电路设计者很难根据其结果对电路进行分析。在手工分析过程中，为了简化计算，人们往往会根据自己的经验对电路进行简化，删去一些认为不重要的项，从而得到可以分析的电路。符号化分析的简化在两方面将有助于电路的分析，首先，由于符号化分析的简化，电路的规模会相应变小，从而生成的符号化表达式也会较小，从而解决了符号化表达规模巨大的问题；另外，符号化简化后，可以生成人可以看懂的公式，有助于电路工程师针对这些结果进行分析与处理。

文献中有大量有关符号化简化的内容。[30] 中指出电路符号化简化主要分为 3 种类型，符号化构造前简化 (Simplification Before Generation, SBG) [31]、符号化构造中简化 (Simplification During Generation, SDG) [32] 和符号化构造后简化 (Simplification After Generation, SAG) [8, 33]。SAG 是指首先进行符号化构造，然后通过对不重要的项的裁剪，得到简化的符号化表达公式。这种方法主要问题是由于其构造所造成的空间和时间代价过高。SDG 则是在生成符号化公式的过程中，逐步删减不重要的符号化项，这样一步步地生成简化后的符号化表达式。SBG 指的是在符号化构造之前就对电路进行相应的简化，直接对输入符号化工具的电路网表简化，从而避免了会生成过大规模的表达式。SAG 和 SDG 会裁剪电路中不重要的项，而往往这会导致电路中零极点等信息从符号化表达式中分解出来。然而 SBG 由于一开始就是对电路进行操作，故而保持良好的可分解性，然而 SBG 由于在符号化构造前简化往往缺乏参照，比较盲目，生成的结果不一定令人满意。

可以看到，从 SAG、SDG、SBG 的对比中，可以看到，电路的拓扑结构对符号化简化是十分有利的。主要原因在于电路拓扑本身自然包含了电路零极点等信息。[34] 提出了从拓扑入手进行简化的想法，但是从其测试的例子中，可以看出，对大规模模拟集成电路分析仍然有限。以往的大部分符号化方法都不能很好地表现出电路的拓扑结构，即使类似 DDD 这种比较紧凑的符号化方法，在简化过程中也丢失了电路拓扑信息。如果

有好的符号化方法可以表现电路的拓扑结构,那么简化问题会变得更加方便,GPDD 就是很好的含有电路拓扑信息的符号化工具,这将在二看到。2006 年,[35] 中介绍了符号化模型降阶方法,希望能够得到符号化的电路降阶模型,可是其主要针对简单的互连线网络的电路,必不能真正应用到实际的模拟电路设计中来。

1.4 课题研究内容与意义

本课题采用 GPDD 符号化仿真分析方法建立大规模模拟电路具体设计指标与电路元件相关参数之间的联系。从而在给定所需要的电路性能情况下,实现电路关键元件的抓取,以自动化系统地得到电路的符号化简化小信号模型。进一步,本课题在得到简化小信号电路后,通过生成时域宏模型,以实现大规模仿真情况下速度的显著提高。另外,本课题也将针对除去运放差分信号行为外的共模信号以及电源信号进行分析,从而辅助电路工程师全方面地考虑电路设计方法与性能指标。

这一课题可以达到指导电路设计、提供更多电路设计洞察力的目的,并可进一步加快模拟电路设计效率,缩短设计时间。本课题研究将有效地辅助模拟电路工程师进行电路设计,更加地深入理解电路工作原理,并提供更多有关电路信息。同时,本课题将提取设计过程中的关键环节,大大加快并简化电路设计流程,并通过生成的宏模型进一步缩短模拟电路的仿真时间。另外,本课题将尝试将整个流程尽可能自动化,以改变现有模拟电路设计模式。

1.5 主要贡献与章节安排

本文主要提出了模拟集成电路降阶模型的系统化自动生成方法,并对该方法做了大量测试,以证明其稳定性,并进一步尝试将这一方法运用于运放电路时域模型的建立与共模抑制比和电源抑制比的分析中。

本文一共分为五章。第一章介绍了模拟集成电路的设计现状,指出了其设计过程中缺乏系统性自动化分析方法的困难,并介绍了学术圈对这一问题所作出尝试,从而引出了本课题的研究目的。第二章给出了一些有关电路简化的 GPDD 理论,建立 GPDD 理论与电路拓扑之间的关系,从而为接下的章节打下基础。第三章提出了本文最关键的符号化自动化电路低阶模型的生成方法,详细分析了算法的各种细节,并给出充分的测试结果以证明算法的有效性与普遍性。另外,这一章也对时域电路模型的大信号分析建立进行了介绍与测试。第四章对应用于共模抑制比和电源抑制比分析的多端口 GPDD 构建方案进行了分析。最后一章总结全文,并指出本文创新点,并展望进一步的研究计划。本文还提供了两个附录,用以证明两个定理。

第二章 基于 GPDD 的电路简化原理

上一章介绍了电路模型对电路设计的重要性，可见电路低阶模型本身的提出也应有一定方法可以指导其生成。只有在拥有可靠的模拟集成电路低阶模型的情况下，电路设计工程师才有可能对电路做出进一步的理论分析，从而指导进一步的设计。课件电路模型的生成需要电路拓扑的支持。所以电路元件的拓扑结构对于分析本身有非常重要的作用。本章对一些 GPDD 理论进行了补充，来建立 GPDD 结构和电路拓扑之间的关系。并给出多端口电路的构造方法以及敏感度分析方法，以方便对多端口电路的分析及电路的优化。

2.1 GPDD 中的二元操作

GPDD 中有两种操作，分为将电路的元件值置为零，和将电路的元件值置为无穷大。我们发现电路的元件的极限取值可以代表新的简化的电路拓扑结构。这正可以成为我们对电路简化的基石。当我们考虑需要将一个电路中的元件删去时，即可认为是电路这个元件选取了极限的取值导致。当然一个电路元件删去过程中，往往存在两个电路元件删去方式：短路和断路。

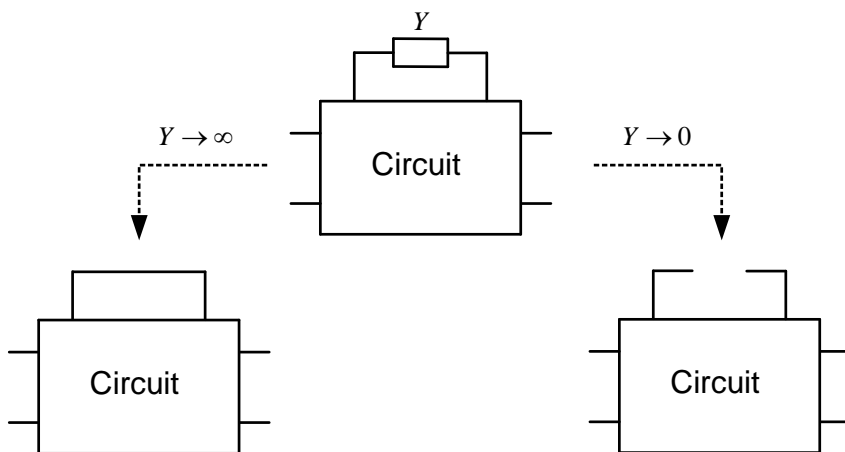


图 2-1 导纳在极限取值情况下的拓扑结构改变

Fig 2-1 Topological adjustment of impedance whose value changes to limit value

如图2-1所示，假设在一个电路中有一处导纳值为 Y 的阻抗。现在我们用两个极限的值 ∞ 和 0 去替代这个导纳值 Y 。可以看到当，导纳值 Y 趋向于无穷大时，由于此时其相应的阻抗 Z 为零，所以此时这个阻抗变为短路的电路结构；然而当导纳值 Y 趋向

于零时，由于此时其阻抗值 Z 为无穷大，这种情况下电路结构变为阻抗两端的两个节点变为了断路的状态。可以看到，这样我们就可以用电路元件的极限取值来替代线性阻抗元件 ($R/L/C$) 的拓扑变化。

然而，在小信号电路中，我们知道不仅存在线性阻抗元件 ($R/L/C$)，另外还有四种受控信号源，分别为：电压控制电压源 (VCVS)，电流控制电流源 (CCCS)，电压控制电流源 (VCCS)，电流控制电压源 (CCVS)。我们发现在以上两种极限取值仍然适用于这四种受控源，特别是在取无穷大情况下，它们都会成为一种称为 Nullor 的电路元件，如图2-2。

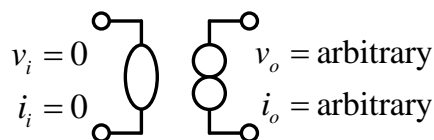


图 2-2 Nullor 元件符号

Fig 2-2 Nullor symbol

可以看到，Nullor 元件假设其输入端的输入电流和电压均为 0，有着任意大小的输出能力。Nullor 这种元件与传统的模拟电路学习中接入负反馈中的理想运算放大器的虚短虚断的性质是一致的。但由于 Nullor 本身往往可以与电路中别的元件合并，并且不会出现在电路最后的模型中，这一点会在本章中节3.4.2.1中看到。

表 2-1 电路元件极限取值的拓扑结构

Table 2-1 Circuit element topology whose value is infinity or zero

Symbol	Value	
	∞	0

为了说明受控源取极限值情况下电路拓扑结构的变化，这里仅以 VCVS 为例进行说

明。所有小信号分析中用到的阻抗和四种受控源的拓扑变化可以参考表2-1。我们知道 VCVS 的输入输出电压关系如下式所示：

$$v_o = E v_i \quad (2-1)$$

这里， v_i 和 v_o 分别为 VCVS 的输入输出电压， E 则为 VCVS 的放大倍数。首先我们考虑 E 取无穷大的情况，为了电路仍然能正常工作，我们知道输出电压 v_o 应为有限值，而其中 E 为无穷大，那么根据基本微积分的知识，我们知道此等式中的输入电压 v_i 为零。另外由于 VCVS 的输入端测量电压，所以本来就限定输入的电流 i_i 为零，所以其形成了 Nullor 的虚短虚断的性质。另外当 E 的取值为零时，根据 VCVS 本身的关系，即可得出输出端电压为零，所以输出端两端电压一致，即此端口可用一根导线连接起来，加之本为断路连接的输入端，故 VCVS 可约减为表2-1中的结构。

2.2 GPDD 与电路拓扑结构的对应关系

上一小节已经将 GPDD 的二元操作与电路的拓扑结构变化建立起了联系，而此时电路元件的极限求值成为了新的问题。由于在整个算法过程中，需要多次计算不同拓扑结构下电路性能表现，这要求了高效的计算方法的支持。然而 GPDD 结构本身蕴含了对电路拓扑结构，可以轻松方便地在其中对不同的电路拓扑结构求值，这支持了下一节所介绍的简化算法。

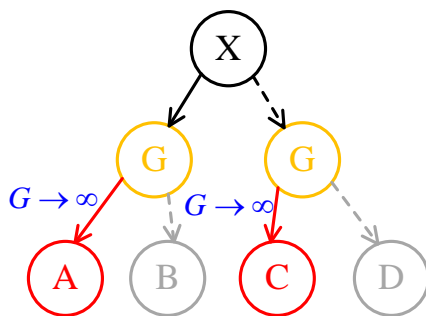


图 2-3 符号极限取值情况下 GPDD 的计算方法
Fig 2-3 Calculation rule for GPDD under limit value

我们知道根据电路基本原理，并且结合 GPDD 的计算规则，针对类似图2-3这样的 GPDD 结构，我们可以求得类似下式的电路传输函数表达式（为了说明的方便，这里忽略了 GPDD 中连接节点的边上的符号）：

$$H(s) = \frac{f_A(A)G + f_B(B)}{f_C(C)G + f_D(D)} \quad (2-2)$$

考虑目前我们需要对电路中的元件 G 求取极限值, 并计算新的符号化传输函数公式。这里假设元件 G 位于 GPDD 符号表中的第一层, 这里层数不影响极限取值的计算, 只是为方便说明特意假设。我们可以得到在 G 元件取无穷大情况下, 电路的新传输函数为:

$$\lim_{G \rightarrow \infty} H(s) = \frac{f_A(A)}{f_C(C)} \quad (2-3)$$

首先, 十分显然地, 可以看到取极限后首先原有的符号 G 从公式消失了, 这也暗示了电路拓扑的简化, 即 G 从电路中以短路的方式被删去了。另外, 可以发现, 在式2-3中所有的保留下的符号项均是在 GPDD 结构中以红色实线相连的红色节点 A 和 C 的值。由于, 我们知道, GPDD 结构中实线相连的节点均采用乘法进行计算, 故在求取极限的过程中, 其相应的系数得到保留。故当某一符号元件需取无穷大情况下的值时, 在 GPDD 计算过程中, 仅需计算 GPDD 对应符号所有节点左儿子的值即可, 所有的右儿子忽略。同样的, 我们知道当对一个电路元件取值零时, GPDD 计算中仅需考虑该元件所有节点右儿子的值即可, 左儿子的值则忽略, 而且算法设计简单, 仅需直接用 0 带入符号值即可。总结来说, GPDD 中某个节点左儿子承担了该节点符号取值无穷大情况 (阻抗短路, 受控源 Nullor) 下的电路性能, 而右儿子为该节点符号取值零情况 (阻抗断路, 受控源删去) 下的电路性能, 当符号本身有一定值时, 则为两者的折衷。

当然需要注意的是, 在求某一个符号为无穷大时, 由于特殊的 GPDD 规模缩小的算法, 类似 Reduction、Zero-Suppression 等算法 [16, 17], 可能存在有排序在 G 之上的符号直接与排序在 G 之下的符号节点相连的情况。这种情况需特别注意, 因为这种项中不包含 G 符号, 故取极限值时, 并非系数, 也需忽略。具体的 GPDD 中符号约减下的计算方法已在 [36] 有阐释, 这里不再赘述。

下面通过一个例子展示电路拓扑结构变化对 GPDD 结构的影响, 以直观了解这两者之间的关系。

例 2.1. RLC 电路向 RC 电路简化过程中 GPDD 结构比较

图2-4中展示了一个 RLC 串联的简易电路, 与之前给出的图1-3相比较, 这里多出了电感元件。可以看到如果这个电感 L 如果取值为零, 那么其导纳 $Y = (sL)^{-1}$, 即导纳为无穷大情况, 即可得到 RC 电路的电路图。

此 RLC 电路对应的 GPDD 结构展示在图2-5中的左侧。这张图右侧是展示的 RC 电路对应的 GPDD 结构。可以看到通过两侧曲线包围起来的结构是一致的, 而且均与电感符号 L 的左儿子相连, 这与之前的论述是相符的。同时可以看到, 由于 RC 的 GPDD 结构蕴含在 RLC 的 GPDD 结构中, 故可以直接在 RLC 的 GPDD 结构上对 RC 电路的

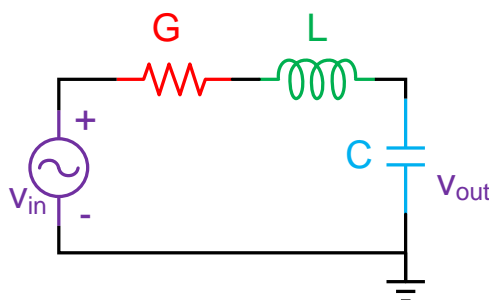


图 2-4 RLC 电路示意图
Fig 2-4 RLC circuit example

GPDD 进行求值。这表示了 GPDD 拥有在同一个 BDD 数据结构中求解多种不同电路拓扑的能力，这大大方便了符号简化电路自动生成算法的设计。这种能在同一个符号化结构中表达不同拓扑结构电路的能力是别的符号化方法不具备的，也是 GPDD 相比其他的符号化方法一大优势所在。

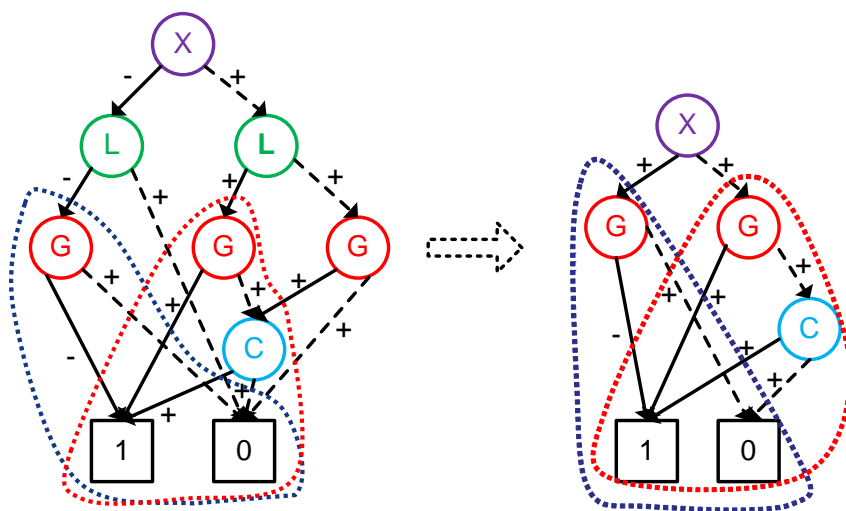


图 2-5 符号极限取值情况下 GPDD 结构示例
Fig 2-5 GPDD structure example under limit value

GPDD 对降阶模型的求解的优势在于在一次符号表达构造的情况下，可以在多次同一个结构中求得对应与不同简化模型的电路传输函数。符号化构造往往大量计算时间花费在符号化表达式的构造上，而其求值往往是高效的稳定的。故可以如将构造的时间平摊在之后的多次求值上，仍然得到了许多计算上的便利。同时，如采用传统的数值化求解器，需要对电路矩阵做行列合并等操作，操作复杂。

2.3 多端口构造原理

传统的多端口 GPDD 构造方法会构造一个多根的 GPDD 结构，并在每个根中生成对应的图对，用于接下来的展开过程 [16]。本文则证明了当电路满足一定条件情况下，电路的 GPDD 构造可以使用一个单根的结构来替代，而不需要多根的展开过程。这么做的好处主要是可以复用原有的 GPDD 展开代码，避免了引入新的程序错误，加快了开发效率，减少花费在程序实现上的工作量。

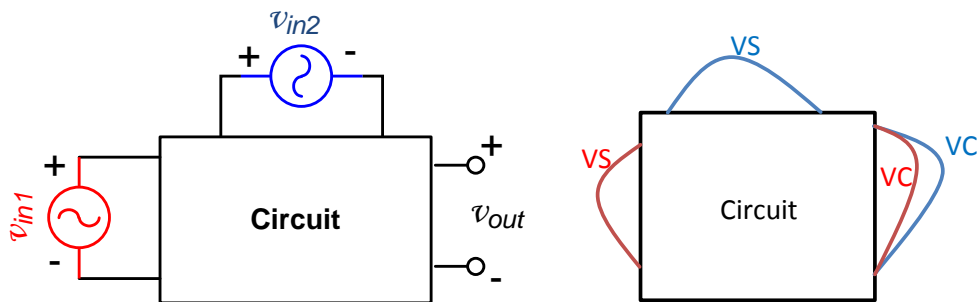


图 2-6 多端口电路示意
Fig 2-6 Multi-Port circuit example

假设我们考虑如图2-6左侧所示的电路结构。这里电路总共有两个输入端口，而且两个输入端口共享同一个电路输出端口。将这样的电路结构画成如图2-6右侧的图，可以看到在输出端口有两条并联的 VC 边。由于 GPDD 算法是对电路中生成树对的枚举过程，所以如果两条 VC 同时出现在最终的生成树对中，则会形成环路。所以如图2-7，必然不会出现两个输入对以实线边相连的情况。

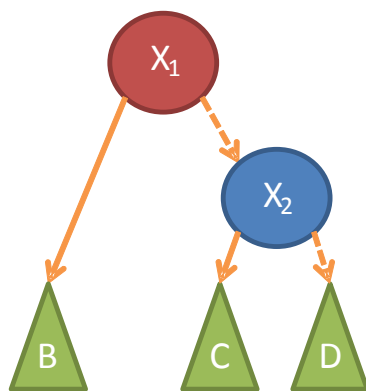


图 2-7 多端口电路的 GPDD 结构
Fig 2-7 GPDD structure for multi-port circuit example

这里我们用符号 X_1 和 X_2 代表两组输入输出关系。如需要对电路传输函数进行计算，根据本前面内容中介绍的元件拓扑结构与 GPDD 之间的关系，我们可以发现某个传

输函数即是在另一个输入输出对符号置零情况下得到的结果。区别于之前章节的记号, 这里使用比较简单易写的 $f(N)$ 来作为节点 N 的 GPDD 计算结果。所以可以根据下两式对电路的传输函数进行计算:

$$H_1(s) = \frac{v_{out}}{v_{in1}} = \frac{1}{X_1} = \frac{f(B)}{f(D)} \quad (2-4)$$

$$H_2(s) = \frac{v_{out}}{v_{in2}} = \frac{1}{X_2} = \frac{f(C)}{f(D)} \quad (2-5)$$

另外可以看到电路传输函数共享同一个分母, 这与基本电路两端口的性质是一致的。只要在满足如下定理的情况下, 这种情况在推广到更多的端口与更大的电路规模仍然成立。

定理 2.1. 在满足以下条件之一的情况下, GPDD 可以用单根构造的方式构造多端口电路, 且不会出现多个输入输出端口符号的交叉项:

- a) 所有端口的输出端均在同一端口测量同种信号 (电流或电压)。
- b) 所有端口的输入端均在同一端口施加同种信号 (电流或电压)。

这里使用生成树对枚举的方式对定理的正确给出说明。我们只对输出端可能出现的 VC 边和 CC 边进行说明, 输入端的 VS 和 CS 可以类比得到。

首先来考虑 VC 边, 我们知道, 对同一个端口的电压进行测量, 那么相应的所有的 VC 边必然并联。那么很明显如果有多个 VC 被包含的情况下, 必然形成环, 不会出现树的结构, 故生成的 GPDD 中必然不含有多个输入输出端口符号的符号化项。其次我们考虑 CC 边, 对同一个端口的电流进行测量, 那么相应的所有的 CC 边必然串联。根据 GPDD 理论 [17], 如需符号项需包含有 CC 边的符号, 那么在生成树对中的左树中 CC 边并不会包含在最终的树中。由于, CC 边串联, 故如有两个或以上输入输出符号被包含在符号项中时, 必然造成图中出现了分离的图, 而分离的图无法生成树结构。这样的话我们就可以得到定理 2.1 的条件 a, 条件 b 可以用类似的方法说明。

2.4 符号化敏感度分析方法

GPDD 理论中关于电路的敏感度分析将大大有助于模拟电路工程师加深对电路设计的理解, 帮助电路工程师对电路性能优化提供更多的具有洞察力的信息 [22–24]。

这里对符号化敏感度分析做一定回顾, 并给出考虑 CMRR 和 PSRR 特殊情况下的计算规则。敏感度定义如下:

$$Sens(H(s), p) = \lim_{\Delta p \rightarrow \infty} \left\{ \frac{\frac{\Delta H(s)}{H(s)}}{\frac{\Delta p}{p}} \right\} = \frac{p}{H(s)} \frac{\partial H(s)}{\partial p} \quad (2-6)$$

敏感度反应了电路传输函数 $H(s)$ 的变化对参数 p 的变化的所产生的影响。可以想象若敏感度绝对值很高，则 p 的变化会引起较大的数 $H(s)$ 的改变。若我们选择各个 MOS 晶体管的宽 W 为 p ，那么就可以求得电路元件尺寸与电路性能直接关系，则可以得到如下公式。

$$Sens(H(s), W_i) = \frac{W_i}{H(s)} \sum_j \sum_p \frac{\partial H(s)}{\partial g_{j,p}} \frac{\partial g_{j,p}}{\partial W_i} \quad (2-7)$$

其中 W_i 为电路中 i 号 MOS 管的宽，而 g_{ij} 则为 i 号 MOS 管的第 j 个小信号参数，如： g_m , g_{ds} , g_{mb} , c_{gs} 等。可以看到，若我们忽略别的晶体管对 i 号晶体管小信号参数的影响，则上述公式最右侧的式子中 $H(s)$ 可以由 GPDD 进行计算，而第二个偏导数可以由器件模型以及 DC 偏置点决定。我们通过一个例子来进一步解释式2-7的意义。

例 2.2. 运放中的敏感度公式意义

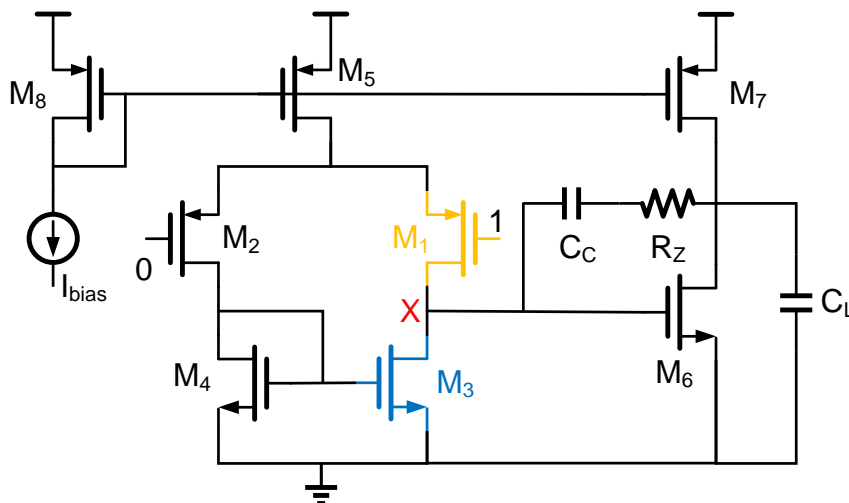


图 2-8 用于敏感度说明的两级运放电路

Fig 2-8 Two-stage opamp for sensitivity explanation

考虑图2-8中的运放电路，如果我们需要对这里的 M_1 管的宽 W_1 进行敏感度分析。根据式2-7中所示，可以得到如下的推导：

$$\begin{aligned}
Sens(H(s), W_1) &= \frac{W_1}{H(s)} \frac{\partial H(s)}{\partial W_1} \\
&= \frac{W_1}{H(s)} \left(\sum_p \frac{\partial H(s)}{\partial g_{1,p}} \frac{\partial g_{1,p}}{\partial W_1} + \sum_{j \neq 1,p} \frac{\partial H(s)}{\partial g_{j,p}} \frac{\partial g_{j,p}}{\partial W_1} \right) \\
&= \frac{W_1}{H(s)} \left(\frac{\partial H(s)}{\partial g_{m1}} \frac{\partial g_{m1}}{\partial W_1} + \dots + \frac{\partial H(s)}{\partial g_{ds3}} \frac{\partial g_{ds3}}{\partial V_X} \frac{\partial V_X}{\partial W_1} + \dots \right) \quad (2-8)
\end{aligned}$$

可以看到在最后一步中展开的两项中，其中第一项仅与 M_1 管的参数有关，而第二项还与 M_3 管有关。造成这种现象的原因主要在于当改变 W_1 的时候，必然会对 X 节点的电压造成影响，故会影响 M_3 管偏置情况，故此时可与 M_3 管中的小信号参数相关。

关于式2-7第一项偏导数的计算，其实十分类似我们选取元件取值为无穷大是的计算过程，首先我们假设 $H(s)$ 的公式如下式所示：

$$H(s) = \frac{N(s)}{D(s)} = \frac{N_1(s)Y + N_2(s)}{D_1(s)Y + D_2(s)} \quad (2-9)$$

我们知道线性电路的传输函数一定是一个有理多项分式。故分子 $N(s)$ 和分母 $D(s)$ 均为乘积项之和（Sum of Products, SOP）的形式，且显然分别为 GPDD 根节点的左右子结构。 Y 是电路中某个元件的导纳值或者受控源的系数。显然，我们可以将 $N(s)$ 和 $D(s)$ 分解为不包含 Y 这一项的 $N_1(s)$ 、 $N_2(s)$ 、 $D_1(s)$ 和 $D_2(s)$ 四个式子。可以注意到由于 GPDD 结构节点求值的方法，每个节点总是乘以左边的节点的值并加上右边节点的值。故可以想到 $N_1(s)$ 是 GPDD 根节点左侧子结构中忽略所有 Y 节点的右边连接关系得到的结果，以此类推，可以得到 $N_2(s)$ 、 $D_1(s)$ 和 $D_2(s)$ 相应地在 GPDD 中对应的结构。对于求取偏导数，我们有

$$\frac{\partial H(s)}{\partial Y} = \frac{\frac{\partial N(s)}{\partial Y} D(s) - N(s) \frac{\partial D(s)}{\partial Y}}{D^2(s)} = \frac{N_1(s) D(s) - N(s) D_1(s)}{D^2(s)} \quad (2-10)$$

上式表示传输函数针对某个电路元件 Y 的偏导数与 $N(s)$ 、 $D(s)$ 、 $N_1(s)$ 和 $D_1(s)$ 有关。可以注意到仅有与 Y 相乘的项得到了保留，相对应的在 GPDD 中，即仅有与 Y 节点的左边节点才计入计算，右边节点则忽略，这与求取元件无穷大取值情况下的 GPDD 计算是一致的。同样的，同时需要注意到由于 GPDD 的共享性质，有些从根节点到 1 结点的路径不会经过 Y 节点，这种项也不能计入其中。

2.5 本章小结

本章给出了一系列有关 GPDD 电路简化的理论与敏感度分析的方法，并通过多个电路实例向读者介绍这些理论的具体意义。这些理论建立了电路元件取值与电路拓扑结构之间的关系，为下一步的电路低阶模型的生成打下了坚实的理论基础。多端口构造理论在分析含有多端口的电路情况下，可以有大量的应用，如同时分析 CMRR、PSRR 等。敏感度分析往往可以用于电路的元件尺寸的优化改进等方面，方便了电路的调整。

第三章 模拟集成电路模型降阶方法

只有在拥有可靠的模拟集成电路低阶模型的情况下，电路设计工程师才有可能对电路做出进一步的理论分析，从而指导进一步的设计。本章给出了本文最关键的算法：采用拓扑简化方法得到的低阶模型自动生成算法，并给出了其相关的测试结果，佐证了算法的有效性。在建立电路拓扑与元件取值之间的关系后，使用 GPDD 这个便利的工具可方便求取电路拓扑变化情况下的电路性能的条件下，本章对符号化降阶模型生成算法进行全面的介绍。本文讨论中，我们主要针对运算放大器的设计进行讨论，故我们接下来的讨论将均针对运算放大器设计。

3.1 元件重要性选取方法

元件重要性是在符号简化工程中十分重要的一个指标。重要性表征了电路总体性能与某个单一元件的值的密切程度。如果一个元件的重要性程度高，那么代表电路性能对这个元件的值十分敏感，故在最终的降阶模型电路图中应有所保留。针对一个运放来说，只有其运放的差分增益以及相位裕度是我们最为关心的两个指标。

可以想象，如图3-1，当运放电路元件发生变化的过程中，其相应的差分增益与单位增益频率处的相位会发生变化。同样，如果一个元件对运放的差分增益贡献较大，那么在把这个元件从电路中以置为无穷大或者零的方式，将其从电路中删去后，电路的频率响应会发生剧烈变动。也就是说，一个运放的重要性基本等同于元件变化时的电路性能变化的程度。这一点，正是作为我们选取重要性指标的关键所在。

总结来说，元件重要性计算的整体流程如图3-2所示。在算法运行初期，我们得到了运放在一般情况下的差分增益 A_0 以及其在单位频率 UGF 处的相位 PM_0 。接下来，假设我们需要对电路中的元件 S 求取重要性，那么我们首先将 S 的值置为无穷大，取得此时的运放差分增益 $A|_G$ 以及其在单位频率 UGF 处的相位 $PM|_G$ ，并记录相应的操作为 $op1$ 。然后，将 S 的值置为零，取得此时的运放差分增益 $A|_{\overline{G}}$ 以及其在单位频率 UGF 处的相位 $PM|_{\overline{G}}$ ，并记录相应的操作为 $op2$ 。在获得了相应的变化之后，我们可以分别计算得到电路在两种情况下分别的电路 AC 性能的变动指标。这里我们采用了平方和开根的方式对增益和相位两个指标进行融合。同时，为了保证两种不同的物理量之间相互能够比较，采用了相对误差的计算方式，如下两式所示：

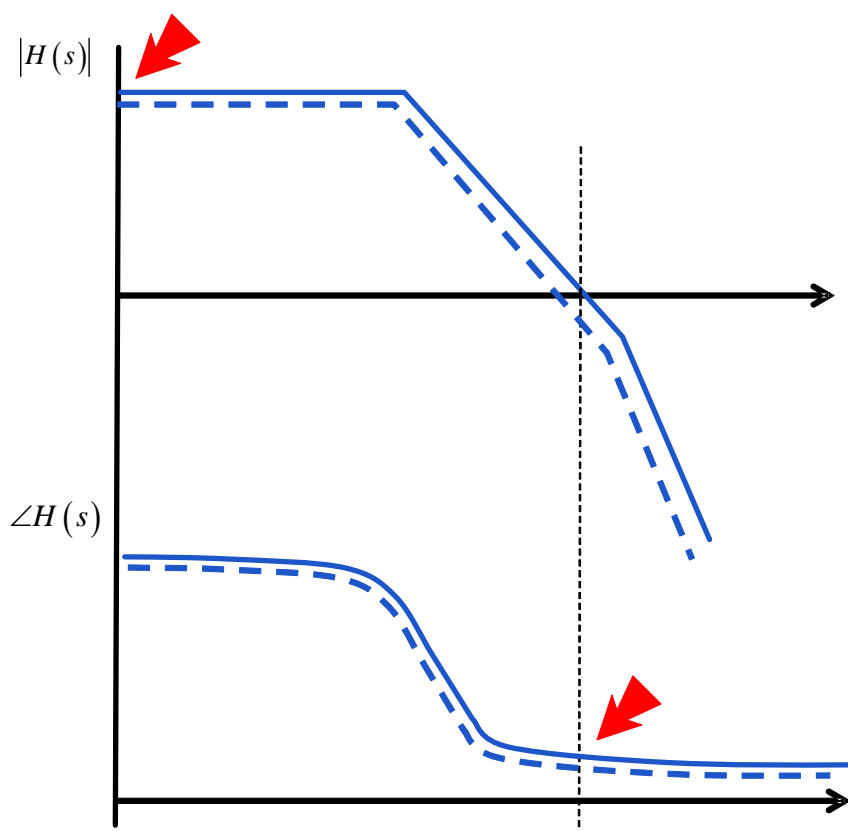


图 3-1 元件变化对运放增益和相位裕度的影响

Fig 3-1 The impact of element value on gain and phase margin

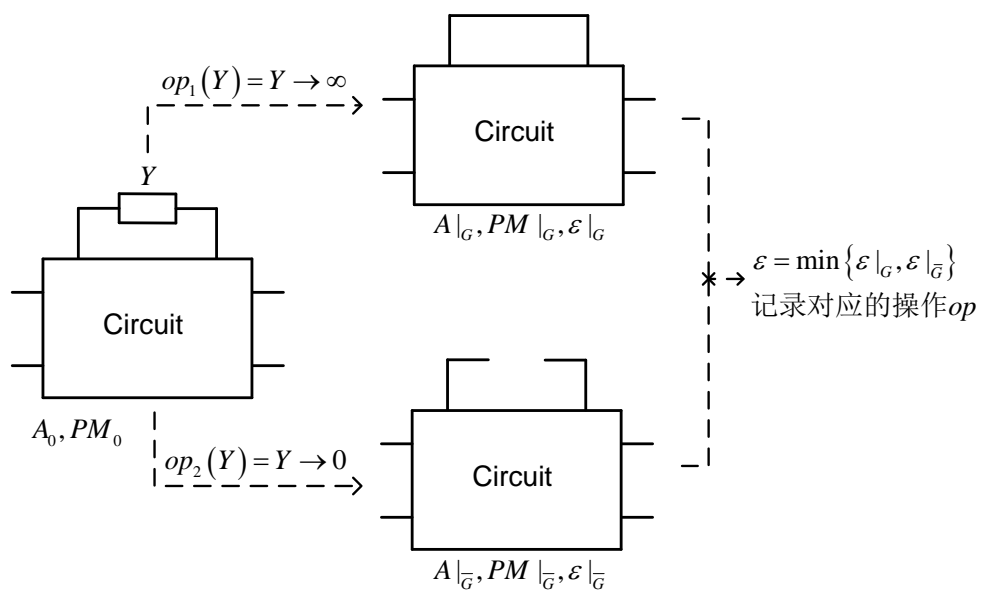


图 3-2 元件重要性计算流程

Fig 3-2 Element significance calculation process

$$\varepsilon|_G = \sqrt{\left(\frac{A|_G - A_0}{A_0}\right)^2 + \left(\frac{PM|_G - PM_0}{PM_0}\right)^2} \quad (3-1)$$

$$\varepsilon|_{\overline{G}} = \sqrt{\left(\frac{A|_{\overline{G}} - A_0}{A_0}\right)^2 + \left(\frac{PM|_{\overline{G}} - PM_0}{PM_0}\right)^2} \quad (3-2)$$

在获得将电路元件置为无穷大和零情况下相应的电路性能的变化 $\varepsilon|_G$ 和 $\varepsilon|_{\overline{G}}$ 之后, 我们选取其中较小的一个作为之后用于排序的重要性指标 ε , 同时记得记录对应的删去操作 op , 即如下式:

$$\varepsilon = \min\{\varepsilon|_G, \varepsilon|_{\overline{G}}\} \quad (3-3)$$

之所以选择较小的 ε 作为该元件的重要性, 主要是因为我们希望在删去电路元件的过程中, 应该尽可能小地改变电路的性能指标。即所选择的 ε 所对应的删去方式, 更适合作为该元件的删去方式, 因为删去该元件之后, 对电路产生的影响较小。如有一个电阻, 如果其值非常小, 可以近似视为短路时, 那么明显 $\varepsilon|_G$ 会小于 $\varepsilon|_{\overline{G}}$, 这时应将该电路短路, 即选择 $\varepsilon|_G$ 作为其重要性, 并进行接下来的操作。

可以看到重要性计算过程中需要求取在元件值为零和无穷大时分别的 AC 增益和 UGF 处的相位, 故需要对两种符号情况的两个频率点分别计算电路的传输函数。这总共需要对 GPDD 进行 4 次遍历过程, 才可以完成, 如忽略常数系数, 则重要性计算的时间复杂度约为 $|GPDD|$ 。

3.2 宏模型自动生成方法

降阶模型的建立, 我们将其看作一个电路拓扑元件简化的过程, 总的流程可参见图3-3的流程示意。

在拿到一个模拟运放电路后, 首先通过数值化仿真器 (如: HSPICE) 利用 DC 求解对电路进行求解, 以获得电路中所有非线性元件 (如: MOSFET) 的小信号参数。接下来, 对于完整电路进行 AC 分析, 求得相应的运放增益 A_0 和单位增益频率处相位 PM_0 (即相位裕度处), 这两个值将在接下来的元件重要性求值运用到。对于电路元件进行预处理, 做一定粗糙的简化工作, 这一步对于生成的最后的简化小信号模型的易于理解性方面有一定的作用, 将在3.2.1中具体介绍。根据初步得到的小信号模型电路, 进行 GPDD 符号化结构的构造, 以方便下一步的元件重要性计算。

这里我们定义了电路元件的重要性, 某个符号的重要性意味着该电路元件符号对

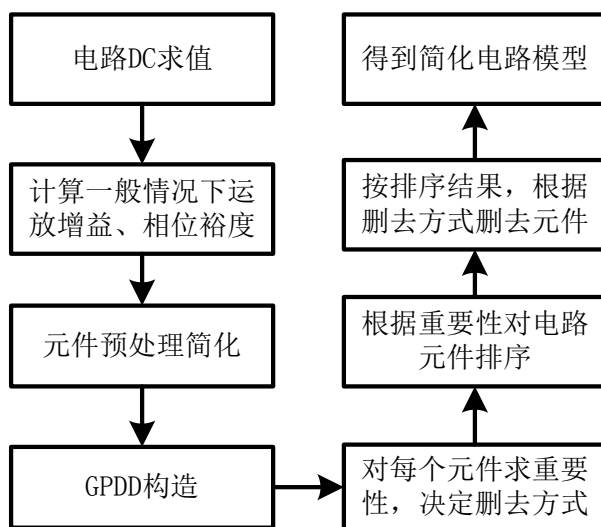


图 3-3 降阶模型自动生成整体流程

Fig 3-3 Entire procedure for low-order model generation

电路性能具有比较大的贡献。故高重要性的电路元件应予以保留，而低重要性的电路元件应予以删去，故如流程中所示，需要对元件的重要性进行排序。同时根据前几节的论述，一个符号有两种删去方式，分别为将该元件置为零或无穷大。我们在计算重要性的过程中即刻确定对应的删去方式，并最终在简化模型生成中以相应的删去方式删去电路元件，从而得到降阶的符号化模型。

这里有一个问题是模型需要多大精度，从而会决定简化的电路模型中保留多少相应的元件。在目前的研究进程中，我们仅人为指定需要保留多少电路元件，通过人为观察电路频率响应曲线，来决定当前的模型是否合适。

这种简化方法可以看到在符号化构建 GPDD 后，最关键的时间复杂度主要集中在对元件重要性求值的这一步上。别的步骤虽然也有一定的时间损耗，但根据我们的实践经验，相比重要性求值步骤来说可以忽略不计。重要性求值需要对每个电路元件进行计算，故若一个电路中有 N 个电路元件，构造得到的 GPDD 结构共有 $|GPDD|$ 节点。在上一小节中可以看到单个元件的重要性求值需 $O(|GPDD|)$ 的时间复杂度。故简化流程的时间复杂度主要由 $O(N|GPDD|)$ 决定。

3.2.1 元件预处理过程

之所以要对电路进行预先简化，其主要目的在于能够事先将一些信号不通过的节点识别出来，从而事先删去一些节点和元件，方便进一步简化算法。许多电路从理论上分析，往往会有一定的做法；然而在实际操作中，人们往往会进一步加入自己的理解与分

析，从而改变理论分析的结果，虽然过程中隐去了一些电路结构，却使进一步的分析与电路理解更加便利。这一点在后续的测试结果中也有所体现。这里以模拟电路中常见的由于输入差分对所造成的虚拟地作为例子，说明这种情况。

例 3.1. 虚拟地理论分析与实际操作区别

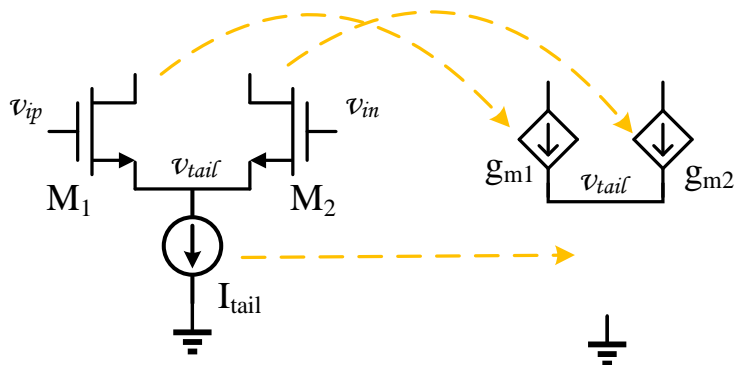


图 3-4 虚地的产生原因
Fig 3-4 Virtual ground illustration

可以看到，图3-4中展现了经典的模拟电路输入差分对，这样的电路结构一般由一个电流源作为两个输入管的偏置，输入结构两端对称。在这种结构下，根据传统的电路分析技巧，我们可以画出如图3-4右侧的小信号结构。其中理想电流作为断路，所以 v_{tail} 这一电路节点浮空，同时保留这里的两个 MOS 管的 g_m 电流源。我们可以根据 v_{tail} 这一电路节点的 KCL 关系，可列写方程有：

$$g_{m1}(v_{ip} - v_{tail}) + g_{m2}(v_{in} - v_{tail}) = 0 \quad (3-4)$$

由于两端的对称性我们知道 $v_{ip} = -v_{in}$ ，并且 $g_{m1} = g_{m2} = g_m \neq 0$ ，那么有

$$g_m v_{tail} = 0 \Rightarrow v_{tail} = 0 \quad (3-5)$$

所以由于尾电流的节点往往被作为虚拟地的存在，在电路分析中往往直接将这一点接到地上，而不是像小信号分析中将电路节点浮空。这种看似矛盾，实际合理的人为处理结果计算机往往很难自动识别，因为从电路基本理论分析，电路中的电流源应该使用断路的端口进行替代。即使使用 MOS 管进行替代，根据传统分析经验，我们知道 MOS 管往往会近似为一个 r_{ds} 所组成的阻抗，而单一的阻抗并不影响上述分析过程，虚地仍然存在，而往往这个阻抗往往比较大，最后仍然会作为浮空处理。但这与多数模拟电路工程师的设计经验不符，电路工程师会将类似的结构直接接到地上进行分析。

另两种常见的类似电路结构是偏置电路以及镜像电流源。如镜像电流源，往往小信号分析中，作为电流源的 MOS 管使用电压偏置也并不会影响电路的差分工作特性。造成这种现象的主要原因在于电路信号往往并不直接通过这些节点进行传递，故导致了小信号电路中较低的电压值。为避免这些电路结构对我们的分析造成影响，我们采取以下的方式对电路进行预处理：

1. 求得电路中所有节点的直流处的 AC 增益。
2. 将增益最小的节点接地，并删去应删去的元件。（元件的所有端口全接地。）
3. 如电路的传输相应曲线有较大变化，则终止预处理；否则，回到第 2 步继续处理。

在这种方案下，我们得到比较合理以及符合工程师习惯的简化电路模型，具体结果可以参照 3.4.1 小节。

3.2.2 简化特殊情况分析

在简化过程中，存在两种可能的简化特殊情况，需要区别处理。出现这些情况的主要原因是对应的电路中拓扑结构不合理造成的。为了使算法能尽可能稳定运行，在不影响电路性能的情况，尽可能多地删去较多的元件。应对这些情况做出合适处理，以使算法尽可能稳定。

3.2.2.1 元件重要性计算过程中处理

首先在重要性计算过程中，我们是将电路元件的值置为零或无穷大的情况下，然后计算相应的重要性。但是有可能出现这样的情况，比如某个电路元件承担了将信号从上一级传递到下一级的职责。那么我们在计算其重要性过程中，如将其值置为无穷大，那么其输出也为无穷大（在程序中显示其值为 INF）；如将其值置为零，那么其输出与输入无关（在程序中显示其值为 NAN）。这两种情况都是我们不期望看到的。所以应规定相应的规则，处理类似情况。我们规定如下规则，处理类似情况：

- a) 如没有删去方式会导致对应的重要性 ε 为 INF 或者 NAN，那么按照 3.1 中的重要性计算方式计算。
- b) 如有仅一种删去方式导致对应的重要性 ε 为 INF 或者 NAN，那么无论另一种删去方式的重要性 ε 大小，选择另一种删去方式，并记录重要性。
- c) 如果两种删去方式都会导致对应的重要性 ε 为 INF 或者 NAN，那么对该元件的操作方式记录为保留，即在后续处理中忽略该元件。

通过这种方式，将有效地避免上述出现的情况。保证算法运行的稳定，尽可能多地删去电路元件，以得到合理的低阶电路模型。

3.2.2.2 最终电路图中删去元件过程处理

另一种特殊情况发生在最终对电路进行电路图层面的约减过程中。因为在电路图的约减过程中，我们仍然需要监控电路的传输函数变化，来确定最终的小信号简化低阶模型的形式。有些电路需要多个电路元件删去后，才会出现电路传输函数的计算过程中的异常值 (NAN 或 INF)。这种情况在 GPDD 计算中，一种常见的情况是电路出现了孤立节点。然而由于 GPDD 的构造过程，是枚举电路生成树对，如果由于简化电路，孤立的节点不会出现生成树中，那么就无法计算对应的 GPDD 的值。这种情况往往需要对元件的删去方式进行翻转，因为往往这种情况并不影响最终的传输函数结果。下面用一个例子对这种情况进行说明。

例 3.2. 电路中多余元件对 GPDD 求值的影响

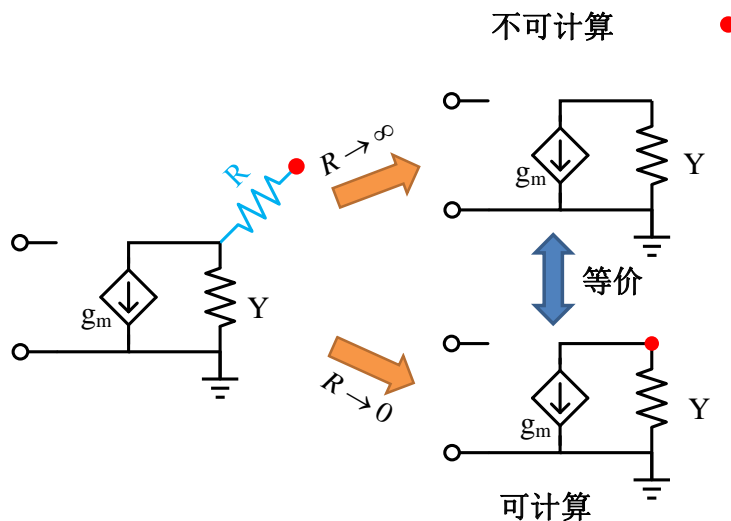


图 3-5 电路图约减过程中特殊情况说明

Fig 3-5 Special case in simplified circuit graph reduction

可以看到图3-5中展示了一个多余的电路元件电阻 R ，现在考虑将电阻 R 删去。可以看到不论如何删去 R 都不会影响到电路的传输函数的相应，可是在 GPDD 中，只有下一种情况是可以继续计算的，因为红色的节点仍然能包含最终的生成树中；而另一种情况则不包含。这种情况下，GPDD 求值存在困难。故如当我们发现类似图3-5中的情况发生时，及时地改变电路元件的删去方式，那么可以得到可计算的 GPDD 结构。

可以看到只要翻转删去方式，往往可以避免此类情况的出现；如翻转了仍无效果，可如上一小节3.2.2.1中所介绍的一样，保留该元件。究其深层次的原因，主要是因为 GPDD 并不完全没有对消项造成的，仍存在类似公因子对消项，这一点会在附录A中给出详细的说明。

3.3 时域宏模型符号化简化模型自动生成方法

通过上一节的分析，可以非常自然地得到电路的时域宏模型的生成方法。本节借用前一节分析得到的简化电路模型，尝试构造相应的电路结构，并加入电流饱和限制，从而得到可以用于时域大信号的电路模型，并通过电路测试结果来显示方法的效果。

3.3.1 传统的时域模型分析方法

对于传统的运算放大器，其阶跃响应输出往往呈现如图3-6所表现的形式。

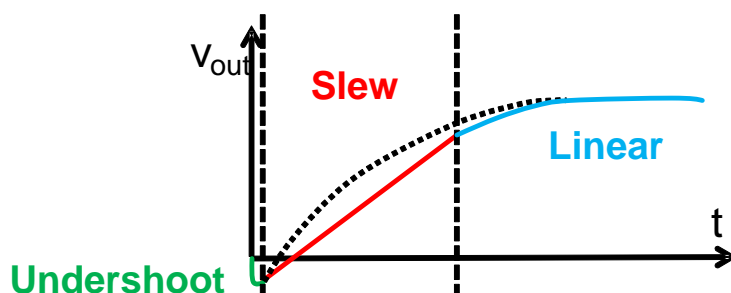


图 3-6 Slew-Settling 过程
Fig 3-6 Slew-Settling procedure

图中，首先绿色部分为 Undershoot 部分，为线性响应，往往由于运放反馈连接中的零点造成，蓝色部分为 Settling 过程的线性响应部分。其中最为关键的是红色的 Slew 部分。由于在实际情况中，运放的输出能力有限，不可能输出无限大的电流，所以在运放输出端给输出端所寄生的电容充电的过程中，电压不一定能以线性的工作方式与时间呈指数上升（如黑色虚线所示）。这种情况下由于电流输出已饱和，基本为以恒定值，故运放输出端电压随时间成比例增长，这条曲线的斜率称之为压摆率（Slew Rate, SR），一般压摆率的估计值可由输出电流 I 与输出端电容 C 的比值决定，如下式所示。

$$SR = \frac{I}{C} \quad (3-6)$$

为了对运放的 Slew-Settling 过程进行分析，往往需要对运放的这一过程进行建模。但是由于上述的公式十分粗糙，很难以用于比较精细的电路分析中。故有大量研究使用

各种方法对运放的时域模型进行建模。[37] 提出了三段式模型对这一过程进行建模。所谓三段式模型即根据运放响应的三个不同的过程（Undershoot, Slew, Settling）进行分段处理，从而形成一个用于表示运放行为的分段函数作为模型。[38, 39] 中使用了大信号分析，从电路器件模型出发，通过直接求解电路微分方程等方法对 Slew-Settling 行为进行分析。这两种方式虽然有一定处理此类问题的能力，但是相对来说，其公式处理十分繁琐同时由于电路被抽象成一个个分段函数，同时往往需要计算积分等复杂运算，很难回溯会具体的电路器件进行分析，也很难自动化进行。

1982 年，Chuang 等人提出了在线性系统中加入限制的方式来模拟电路的 Slew-Settling 模型 [40]。这种方法因其本质上是线性系统，仅通过电流的输出能力大小来限制其输出信号，故比较适用与可以分析零极点的符号化方法。今年来，G. Shi 等人通过结合符号化零极点分析与 Chuang 的电路模型 [25, 26]，提出了使用运放宏模型的方式对运放进行建模。这种方式不仅更为准确，同时提供了更明确的电路意义。一个二级运算放大器的电路宏模型如下图 3-7 所示：

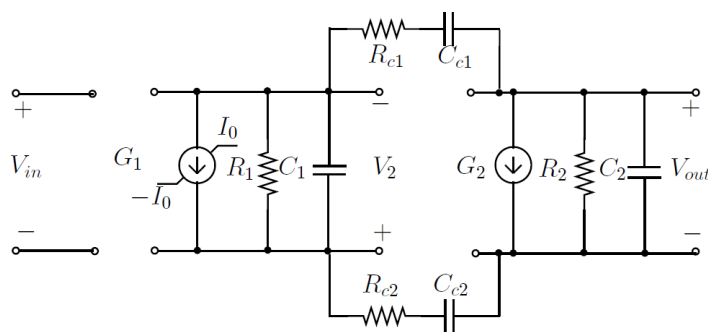


图 3-7 运放时域宏模型 [26]

Fig 3-7 Opamp macromodel in time domain[26]

可以看到通过提取二级运算放大器中各级的零极点，就可以得到每一级中跨导和相应的阻抗与电容。同时注意第一级运放的跨导有电流输出的限制，以模拟运放电路本身的电流输出限制。那么现在问题就主要在于宏模型电路中各个元件的参数选取上，通过 GPDD 结构与模匹配（Moment Matching）的方法可以得到主要的几阶项，从而得到各级的零极点关系。例如，通过 GPDD 计算，我们可以得到如下公式：

$$H(s) = \frac{N(s)}{D(s)} = \frac{b_0 s^0 + b_1 s^1 + \cdots + b_q s^q}{a_0 s^0 + a_1 s^1 + \cdots + a_r s^r} \quad (3-7)$$

对上式进行泰勒展开后，可以得到：

$$H_{ex}(s) = m_0 s^0 + m_1 s^1 + m_2 s^2 + \dots \quad (3-8)$$

其中系数可以通过联立上述两个公式后，对公式两边多次求导的方式，逐一得到相应的系数，如下式所示。

$$\begin{aligned} m_0 &= \frac{b_0}{a_0} \\ m_1 &= \frac{b_1 - m_0 a_1}{a_0} \\ m_2 &= \frac{b_2 - m_0 a_2 - m_1 a_1}{a_0} \end{aligned} \quad (3-9)$$

故宏模型电路中的跨导 G 、电阻 R 和电容 C 可以以上公式计算得到。这样就建立起了宏模型与电路实际元件之间的关系，有利于日后对于电路性能优化的进一步分析。这种方式的主要优势在于直接将电路模型中的零极点与电路元件关系挂上勾，从而电路模型不仅可以在时域，也可以在频域进行计算。更主要的是这种模型，传统仿真器中可以简单地通过编写网表得到，仿真方便便捷。

3.3.2 符号化时域简化模型分析方法

3.3.2.1 时域符号化简化流程

本方法的基本流程类似于 **Chuang** 模型，首先获取简化的小信号模型，然后在该小信号模型的基础上加入对输出电流饱和的限制。但是，本方法与 **Chuang** 模型有三点不同，分别为：

- a) 简化的小信号零极点模型可以直接通过第二章的符号化模型降阶算法得到。
- b) 限制电流加载运放中所有得以保留的 g_m 元件上，并用大信号中的通过管子的电流作为其限制电流。
- c) 采用了更加光滑的非线性函数，以更加真实地模拟实际情况。

其中第一点正是相比之前的方法的闪光点。因为之前的方法，即使是一般的零极点模型，往往很难建立模型与电路元件之间的关系，而给出简化小信号模型可以直观地给出两者之间的联系。

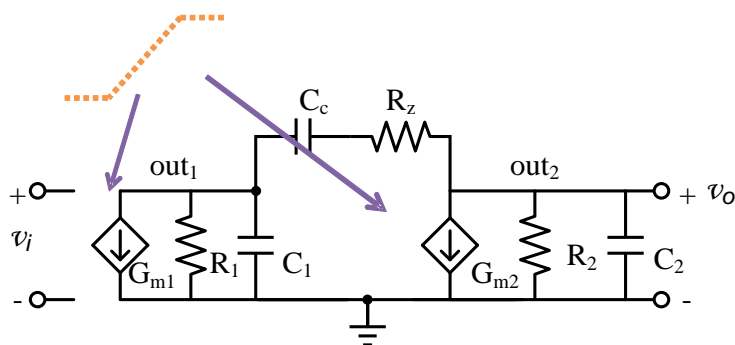


图 3-8 时域模型中的电流限制位置

Fig 3-8 Current limit position in slew-settling model

第二点可参考图3-8的示意，本方法将运放电路中所有存在的 g_m 均加入限制。[38]中指出，现代的模拟集成电路由于其功耗降低，电源电压降低等原因，造成电流输出饱和的原因不一定是由输入级决定的，可能是之后的电路充电速度不足导致。**Chuang** 模型中只考虑了第一级电路的输出饱和，对之后级的输出限制没有做出过多考量。如果在各个 g_m 上加入相应的电流限制，那么即可模拟各级中的电流限制。

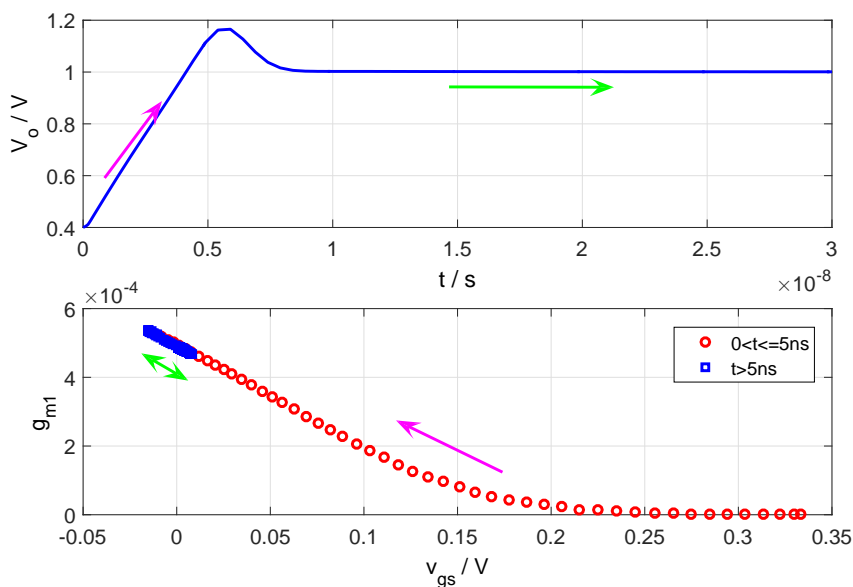
图 3-9 Slew-settling 过程中输入管 g_m 的变化Fig 3-9 g_m variation during the slew-settling procedure

图3-9给出了两级运算放大器 slew-settling 的过程中输入管的 g_m 随着输入交流小信号的变化。图中箭头方向代表了时间的流向。在 Slew 过程（红色圆圈）中， g_m 快速爬升至所需要的稳态情况的值；而在 Settling 过程（蓝色方块）中， g_m 稳定在某一值不再

变化。以往，我们往往假设 g_m 的函数是一个分段函数，在未饱和前，输入电压和输出电流成正比；饱和后，输出稳定电流。这样的话 g_m 仅有两个值，其一是稳态情况下的跨导，另一个是 0。然而在图中明显看到这样的假设是不合理的，一个办法就是重新决定施加给 g_m 的非线性函数。

3.3.2.2 非线性函数选取

非线性函数有许多不同的形式。一类合适的函数是 S 型函数族 $S_i(x)$ 。这里列举了五种不同的函数方程供参考：

$$S_0(x) = \begin{cases} 1 & x \geq 1 \\ x & -1 \leq x < 1 \\ -1 & x < -1 \end{cases} \quad (3-10)$$

$$S_1(x) = \tanh(x) \quad (3-11)$$

$$S_2(x) = \frac{2}{\pi} \operatorname{gd}\left(\frac{\pi}{2}x\right) = \frac{2}{\pi} \arcsin\left(\tanh\left(\frac{\pi}{2}x\right)\right) \quad (3-12)$$

$$S_3(x) = \frac{x}{\sqrt{1+x^2}} \quad (3-13)$$

$$S_4(x) = \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \quad (3-14)$$

图3-10中展示了所有这 5 中 S 型函数的曲线。可以看到原先在 Chuang 模型中使用的非线性函数即为这里的 $S_0(x)$ 。图中的箭头显示了这些函数在正半平面或负平面中不互相相较，且呈现出了一定顺序。这类函数有一些共同的特征，如它们都绝对单调递增，可总结如下：

$$\lim_{x \rightarrow +\infty} S_i(x) = 1 \quad (3-15)$$

$$\lim_{x \rightarrow -\infty} S_i(x) = -1 \quad (3-16)$$

$$\left. \frac{d}{dx} S_i(x) \right|_{x=0} = 1 \quad (3-17)$$

如果有一个跨导的 DC 偏置电流是 I_0 ，其增益为 g_m ，那么对这类函数只需要做如下的变换即可得到我们所需要的非线性的 g_m 。

$$I_0 S_i\left(\frac{g_m}{I_0}x\right) \quad (3-18)$$

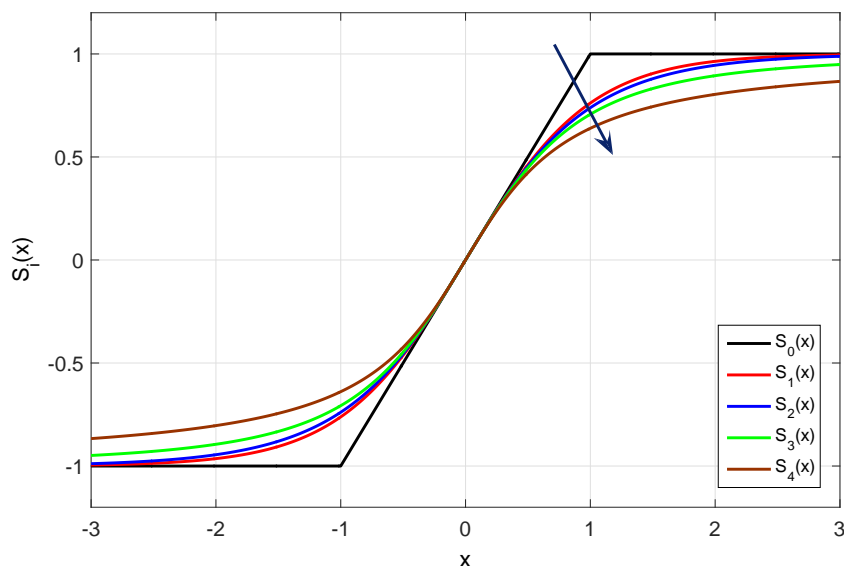


图 3-10 S 型函数族

Fig 3-10 Sigmoid function family

3.4 降阶模型生成算法测试结果与分析

以上的算法从 GPDD 构造步骤开始，均使用 C/C++ 编写，并在个人电脑的 Linux Mint 16 的虚拟机上进行测试。在本文中所有的含有 MOSFET 的测试电路，均采用 TSMC $0.18\mu m$ 工艺，使用 $1.8V$ 的电源电压。所有 MOSFET 的衬底均接至电源或衬底，即考虑衬底偏置效应。并采用如图3-11的 MOSFET 管小信号模型，作为电路简化的起点。

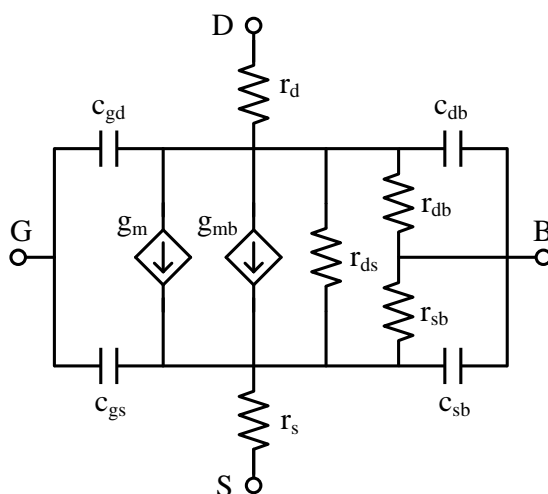


图 3-11 MOSFET 器件小信号模型

Fig 3-11 Small signal model for MOSFET device

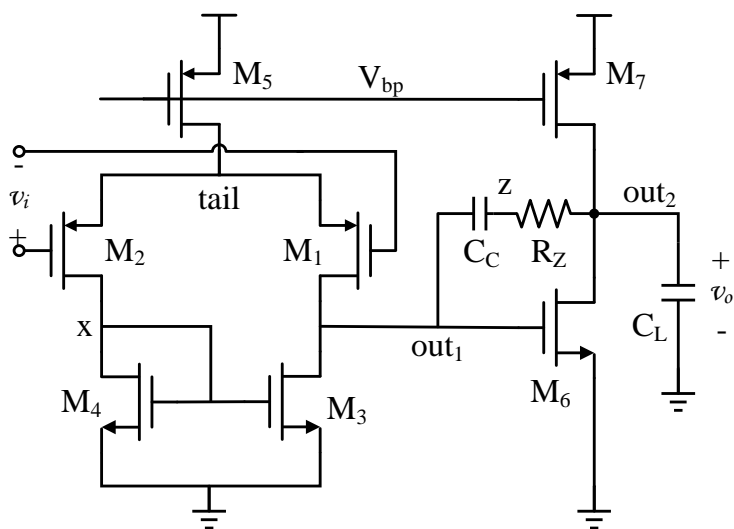


图 3-12 两级运算放大器电路图
Fig 3-12 Two-stage opamp schematic

本文最常用的测试电路使用如图3-12所示的传统两级运算放大器，其具体的MOS管的大小以及偏置电压的选取可以参考3.4.3中的表3-3和表3-4中的0号尺寸方案。

3.4.1 预处理简化降阶电路模型区别

首先我们观察电路元件预处理步骤对简化电路的效果。在表3-1中列出了原始电路的各个电路节点的AC增益情况，并从小到大排序。可以看到那些不会通过信号的电路节点（如 V_{bp} 和 $tail$ ）都有比较小的AC增益。在继续尝试将 x 节点接地后，会发现电路性能发生了巨大的变化，故只有 V_{bp} 和 $tail$ 被接地。

表 3-1 两级运放预处理计算各节点AC相应
Table 3-1 AC Response for each node in two-stage opamp

节点名	直流处 AC 增益	是否预先接地
V_{bp}	20μ	✓
$tail$	$215m$	✓
x	$384m$	×
out_1	82	×
z	$5.4k$	×
out_2	$5.4k$	×

可以在图3-13中看到，这是传统两级运算放大器的简化结果，并在图3-14给出了完整电路的HSPICE、GPDD以及简化电路的GPDD仿真频率响应曲线的比较结果。在简

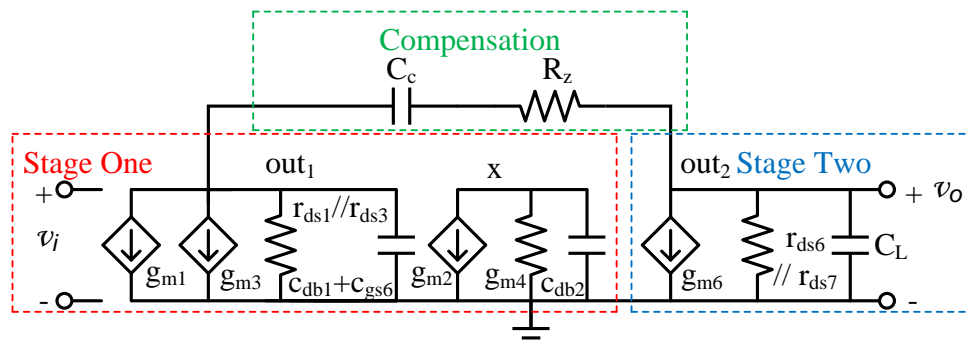


图 3-13 无虚拟地的两级运放简化小信号电路

Fig 3-13 Simplified small-signal circuit for two-stage opamp without virtual ground

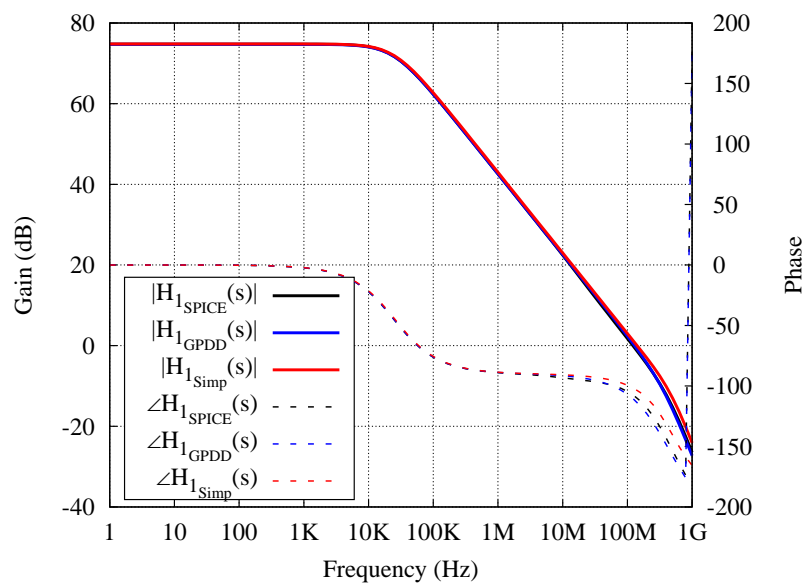


图 3-14 无虚拟地的两级运放频率响应曲线比较

Fig 3-14 Frequency response comparison for two-stage opamp without virtual ground

化的小信号电路图中，可以清晰地看到电路被明显地分为了两级，而且每一级由一个传统的 $g_m RC$ 的极点模型替代，保留下最为关键的电路元件。其中第一级中有一个 g_{m2} 、 g_{m4} 和 c_{db2} 组成的小电路结构，这主要起到了运放中通过镜像管 M4，给 M3 提供输入信号的拷贝的作用。同时补偿的电容 C_c 和调零电阻 R_z 均在自动生成的简化降阶模型中得到保留。对比 [41] 中的对两级运放的分析，给出了相应的 DC 增益，可参考下式。

$$A_v = A_{v1}A_{v2} = -g_{m1}(r_{o1}||r_{o3})g_{m6}(r_{o6}||r_{o7}) \quad (3-19)$$

可以看到所有公式的有关的电路元件均得以得到保存。同时利用并不是很多的电容元件完成了频率响应的模拟。可以看到这三个电容分别模拟了 [42] 中描述的电路中主极点、次极点和镜像极点。所以，该算法得到的电路模型还是十分准确的，而且易于电路工程师分析与理解。

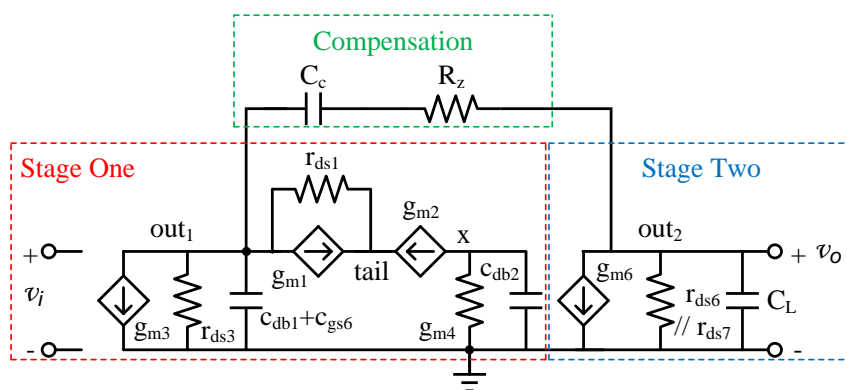


图 3-15 有虚拟地的两级运放简化小信号电路

Fig 3-15 Simplified small-signal circuit for for two-stage opamp with virtual ground

为了考察如不做元件预处理的运算放大器简化结果，这里给出了没有预先使 $tail$ 节点接地的简化电路结构，如图3-15所示，其相应的频率响应比较结果在图3-16中给出。可以看到如果没有实现做元件的预处理，会得到含有虚拟地的电路结构。明显经过预处理的电路结构简化结构更令人满意且简洁。在将来的设计中，可以将这一功能放开给工程师选取，看是否有必要进行预处理简化。

3.4.2 多种电路结构简化结果分析

3.4.2.1 折叠共源共栅运算放大器拓扑简化结果

除了普通的二级运算放大器外，本算法也适用于别的电路结构，如图中3-17所示的折叠共源共栅电路结构。

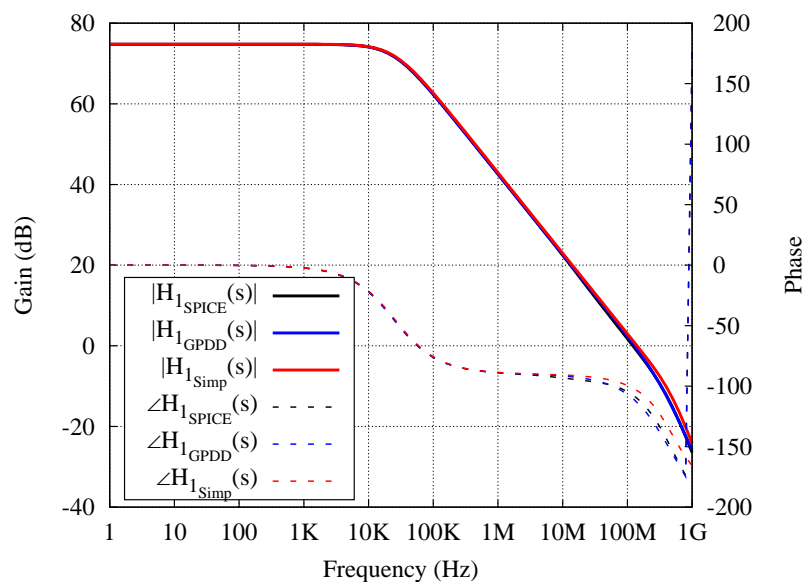


图 3-16 有虚拟地的两级运放频率响应曲线比较

Fig 3-16 Frequency response comparison for two-stage opamp with virtual ground

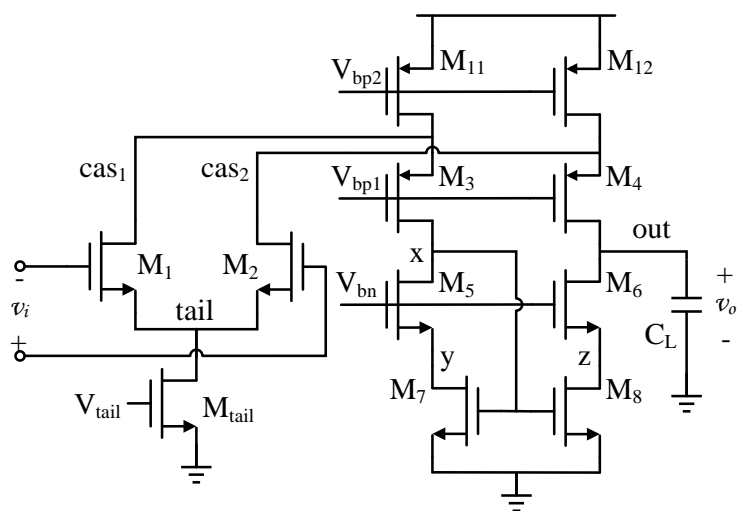


图 3-17 折叠共源共栅运算放大器电路图

Fig 3-17 Folded-cascode opamp schematic

表 3-2 折叠共源共栅运放预处理计算各节点 AC 相应
Table 3-2 AC Response for each node in folded-cascode opamp

节点名	直流处 AC 增益	是否预先接地
V_{bn}	0	✓
V_{bp1}	0	✓
V_{tail}	712p	✓
V_{bp2}	48n	✓
$tail$	447m	✓
cas_1	2.7	×
cas_1	24.6	×
x	3.6	×
y	2.6	×
z	82	×
out	3.0k	×

表3-2中展示了需要预先接地的电路节点。可以看到同样是运放的电路的多个偏置节点以及尾电流处构成的虚拟地需要预先接地，这与我们之前的预期是相符的。

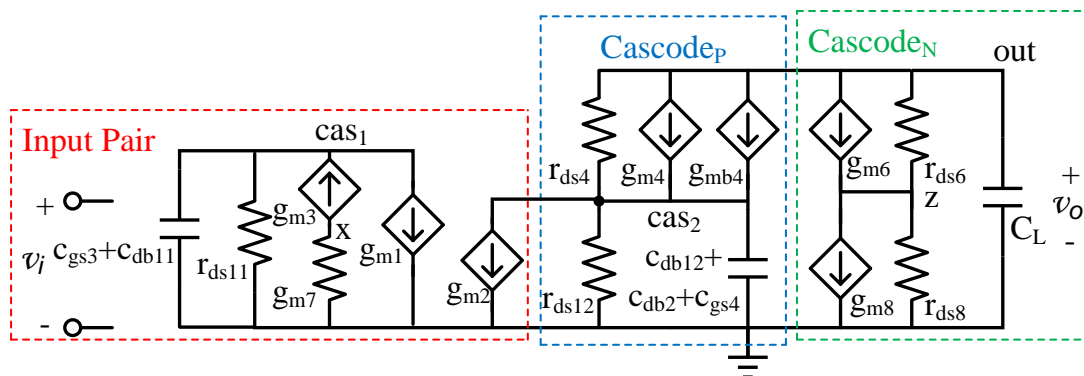


图 3-18 折叠共源共栅运放简化小信号电路

Fig 3-18 Simplified small-signal circuit for folded-cascode opamp

简化得到的电路结构如图3-18所示，其相应的频率响应比较在图3-19中展示。可以看到简化的小信号电路正如 [41, 42] 等多本传统模拟集成电路教材中所学习的电路共源共栅复合 MOS 管结构的小信号模型。同样对比 [41] 中给出的 DC 增益公式：

$$\begin{aligned}
 A_v &= G_m R_o = -g_{m1} [(R_{out,4}) || (R_{out,6})] \\
 R_{out,4} &= [g_{m4} (r_{ds2} || r_{ds12})] r_{ds4} \\
 R_{out,6} &= (g_{m6} r_{ds8}) r_{ds6}
 \end{aligned} \tag{3-20}$$

可以看到除了 r_{ds2} 之外，所有的电路符号均在公式中出现。替代 r_{ds2} 位置的是电流

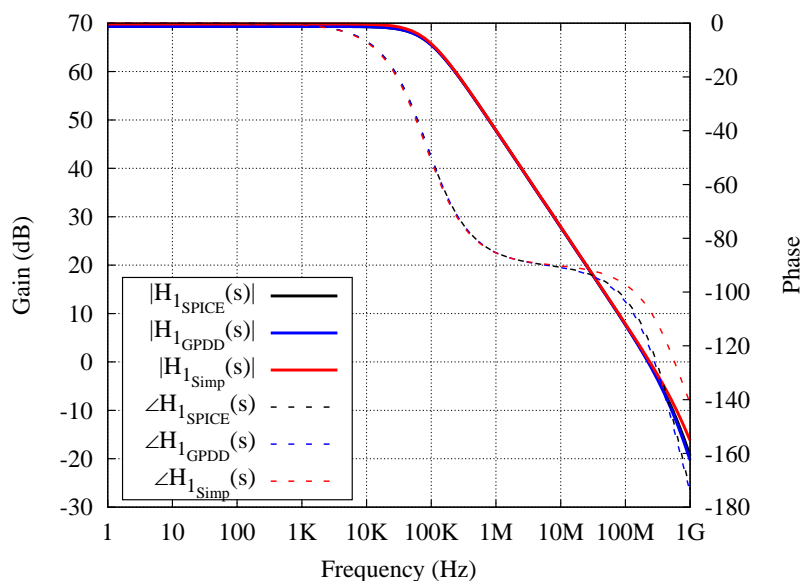


图 3-19 折叠共源共栅运放频率响应曲线比较

Fig 3-19 Frequency response comparison for folded-cascode opamp

管 M_{12} 的 r_{ds12} ，说明在这里电路的电流源阻抗较小，可能会提醒电路工程师对电流源的输出阻抗进行优化，或者采用更好的电路结构等。

另外此电路中还有一处需要多加说明，电路的输入级中 g_{m3} 和 g_{m7} 所构成的通路并不是直接算法简化得到的电路。算法简化得到的电路如图3-20左侧所示。其中 g_{m5} 被简化为 Nullor，故形成了虚短虚断的特性。通过简单分析，即可发现可以将 g_{m5} 和 g_{m7} 合并为一个电路元件接在电路节点 x 和地之间，并以 g_{m7} 作为其导纳值。

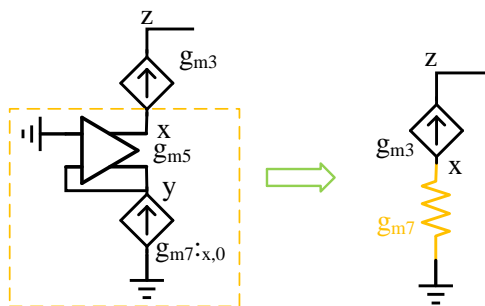


图 3-20 折叠共源共栅运放降阶模型中的 Nullor 等效

Fig 3-20 Equivalent circuit topology for nullor in simplified folded-cascode opamp

3.4.2.2 多种补偿结构的两级运算放大器拓扑简化结果

另外，本课题还针对另两种常见的电路补偿方式进行简化模型生成的测试，分别为图3-21中的 **Voltage Buffer** 补偿方法和图3-22中的 **Current Buffer** 补偿方法。这两种方法都尝试解决由于补偿电容所导致的零点引入的问题，尝试通过打破前馈通路，形成单向的反馈的方法对电路的频率性能进行优化。

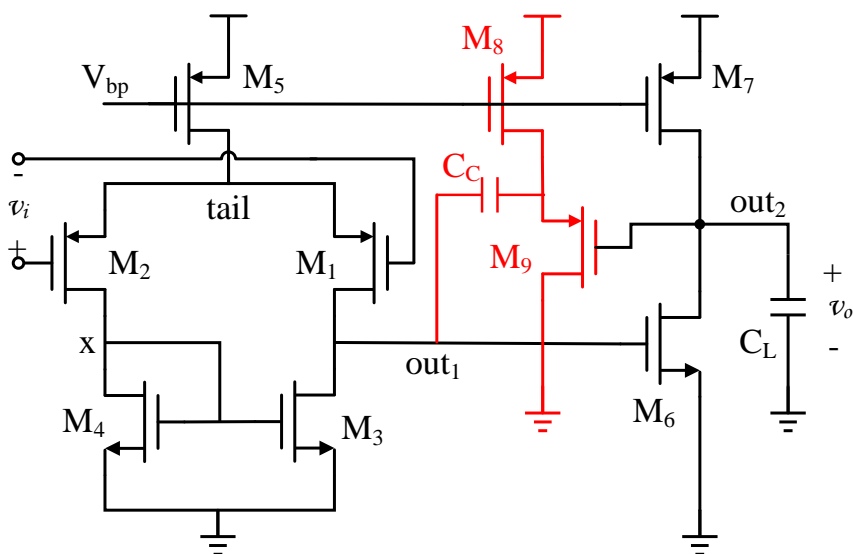


图 3-21 Voltage Buffer 补偿的两级运算放大器电路图

Fig 3-21 Two-stage opamp schematic with voltage buffer compensation

Voltage Buffer 的补偿方式比较简单，相应的 **Sizing** 过程也更为方便，在原始电路调整完成后，基本只需考虑补偿电路的部分的调整即可。**Current Buffer** 的尺寸调整方法往往不是很容易。在图3-22中可以看到，由于有两条 DC 通路（蓝色的补偿电路通路和绿色的第一级输出端通路）影响节点 out_1 ，所以 out_1 的 DC 偏置变得较为困难，同时由于更多的阻抗接入 out_1 节点，其直流增益也相应下降。

图3-23和图3-24中分别给出了 **Voltage Buffer** 补偿的两级运放电路的简化小信号电路图与对应的频率响应比较图。可以看到，除却补偿部分，其余的电路结构与使用调零电阻的补偿方式一致。补偿部分根据 [43, 44] 的分析，可以看到补偿电路与文献中通常的分析一致，打破了前馈的通路。

同样的，图3-25和图3-26中分别给出了 **Current Buffer** 补偿的两级运放电路的简化小信号电路图与对应的频率响应比较图。也可以看到，除却补偿部分，其余的电路结构与使用调零电阻的补偿方式基本一致。但是可以看到由于补偿的加入，运放的第一级增益部分挂上了 r_{ds12} 这个阻抗，减小了电路增益。并且由于 c_{gs9} 的出现，使第一级形成的极点位置更低。补偿部分根据 [42, 45, 46] 的分析，可以看到本算法抓出了不同文献中

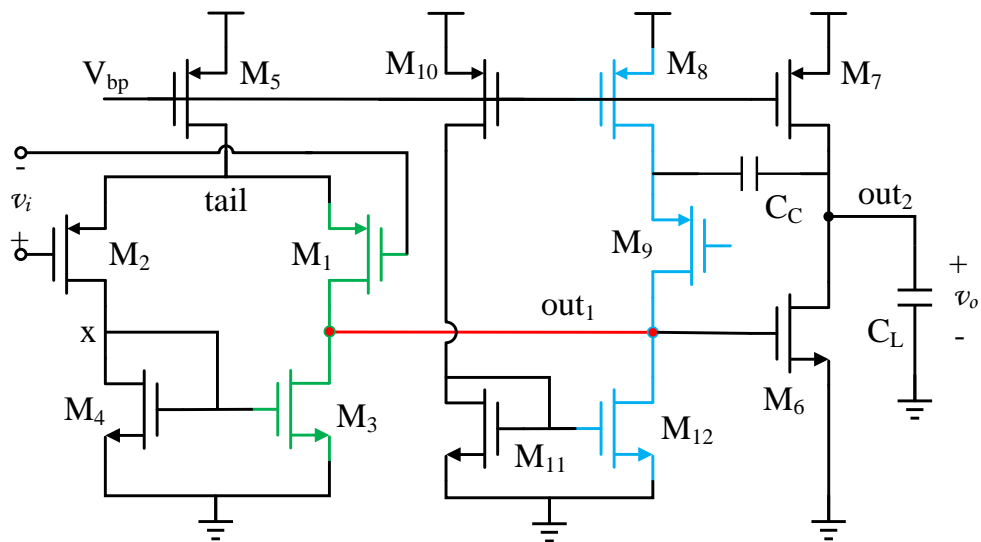


图 3-22 Current Buffer 补偿的两级运算放大器电路图

Fig 3-22 Two-stage opamp schematic with current buffer compensation

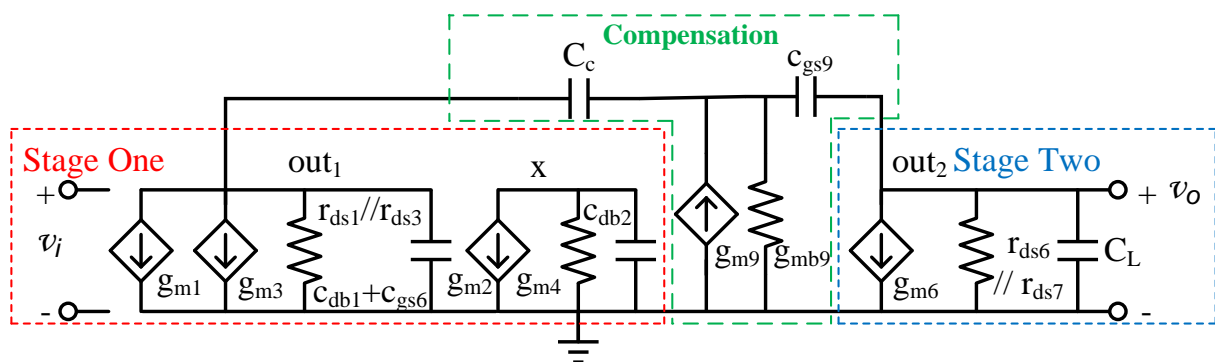


图 3-23 Voltage Buffer 补偿的运放简化小信号电路

Fig 3-23 Simplified small-signal circuit for two-stage opamp with voltage buffer compensation

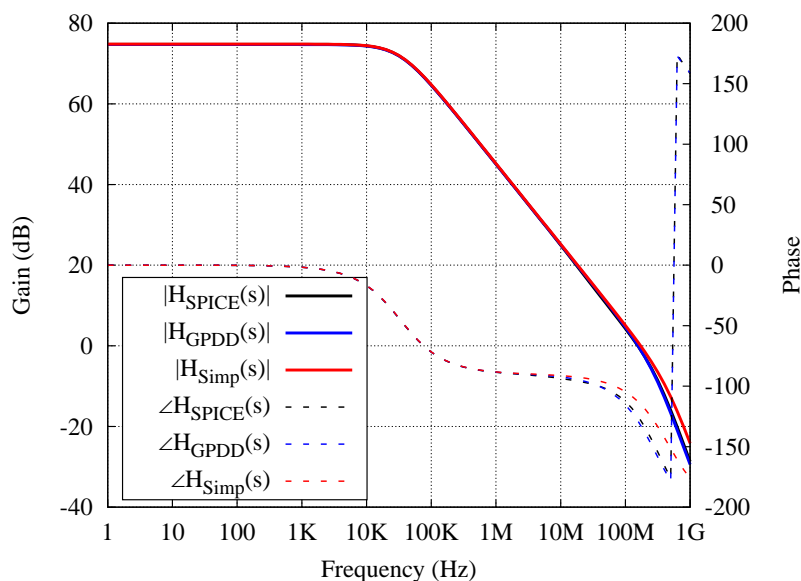


图 3-24 Voltage Buffer 补偿的运放频率响应曲线比较

Fig 3-24 Frequency response comparison for two-stage opamp with voltage buffer compensation

提到的需要考虑的电路元件，证明了该算法在不同电路拓扑结构情况下的有效性。

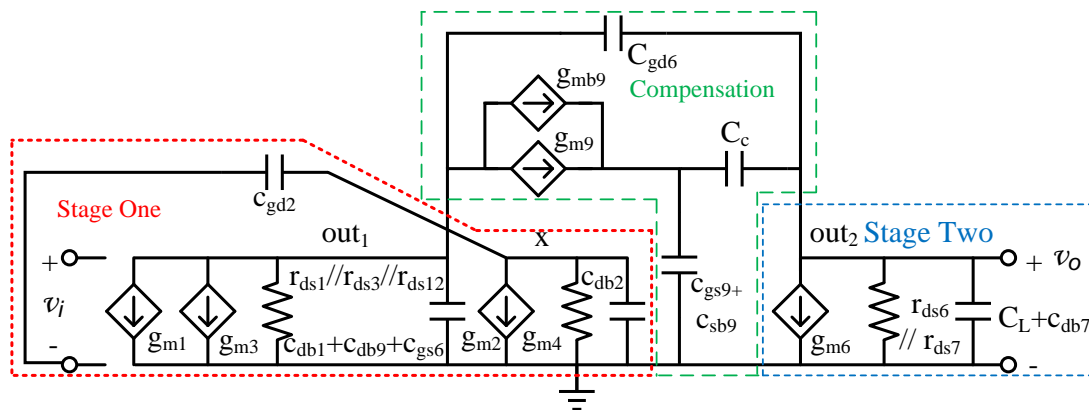


图 3-25 Current Buffer 补偿的运放简化小信号电路

Fig 3-25 Simplified small-signal circuit for two-stage opamp with current buffer compensation

3.4.3 尺寸调整下的算法稳定性分析

同时，针对同一种电路拓扑结构，在运放有不同的尺寸调整情况下，也应该保证该算法的稳定性。故这里对两级运算放大器进行了三次的尺寸调整过程。表3-3中给出了三个运放共同的参数，而表3-4给出了三次运放尺寸参数调整过程中的具体调整方法。

为了保证结果有一定的可比较性，这里三种尺寸方案都保存了 18 个电路元件，作

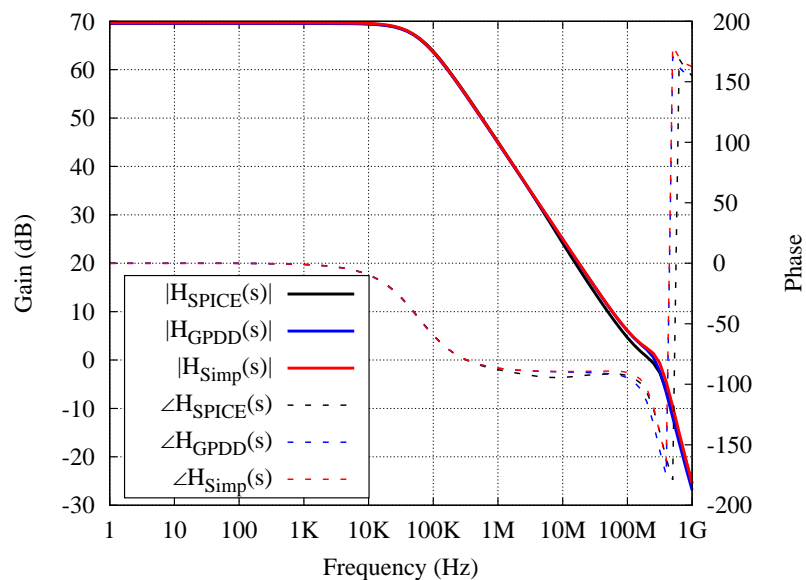


图 3-26 Current Buffer 补偿的运放频率响应曲线比较

Fig 3-26 Frequency response comparison for two-stage opamp with current buffer compensation

表 3-3 两级运放尺寸共同参数

Table 3-3 Common parameter sizing for two-stage opamps

参数	值
电流偏置电压 V_{bp}	1.12V
负载电容 C_L	2pF
所有 MOS 管的长 L	0.5 μm
供电电压 V_{dd}	1.8V

表 3-4 两级运放的不同尺寸方案

Table 3-4 Sizing strategy for two-stage opamps

参数	0 号尺寸方案	1 号尺寸方案	2 号尺寸方案
输入偏置电压	0.9V	0.7V	1.1V
$W(M_1)$	20 μm	40 μm	25 μm
$W(M_2)$	20 μm	40 μm	25 μm
$W(M_3)$	5 μm	1.5 μm	1 μm
$W(M_4)$	5 μm	1.5 μm	1 μm
$W(M_5)$	48 μm	12 μm	10 μm
$W(M_6)$	30 μm	33.5 μm	32 μm
$W(M_7)$	144 μm	144 μm	144 μm
R_z	1.75k Ω	1.75k Ω	1.5k Ω
C_c	350fF	350fF	250fF

为比较。图3-27和3-28分别展现了1号尺寸方案和2号尺寸方案的简化电路结果图；0号尺寸方案的结果已在之前的图3-13中给出。可以看到三者之间并不存在巨大区别。0号和2号尺寸方案的简化模型甚至一模一样，1号与另外两者的区别仅在于镜像电容和第二级电容上做出了取舍。

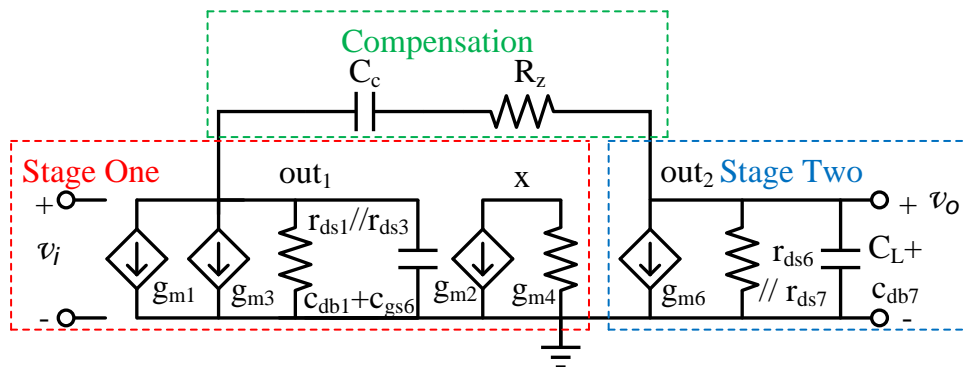


图 3-27 1 号尺寸调整方案的的运放简化小信号电路

Fig 3-27 Simplified small-signal circuit for two-stage opamp with 1st sizing method

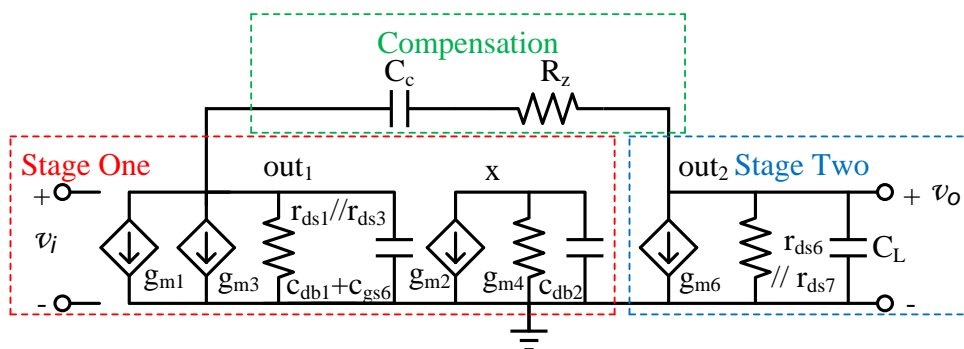


图 3-28 2 号尺寸调整方案的的运放简化小信号电路

Fig 3-28 Simplified small-signal circuit for two-stage opamp with 2nd sizing method

可以在表3-5中发现所有尺寸调整方案中发生变化的元件 C_{db2} 和 C_{db7} 均在所有的 18 个保留下的元件重要性排在末尾。这说明这两个元件原本对电路的贡献较小，故发生了调整，并不令人惊讶，仍然在可接受范围内。另外可以看到虽然其他的元件的重要性顺序发生了一定的变化，但并不基本都局限于比较小的范围内，并不影响整体的简化流程。可以看到本课题提出的算法在同一电路上不同偏置，不同参数尺寸的情况下，仍然可以稳定运行，具有较高的可信度。

表 3-5 不同尺寸下的两级运放元件重要性排序 (编号靠前重要性越大。)

Table 3-5 Symbol order of opamp under different sizings (Symbol with smaller No. is more significant.)

编号	0 号尺寸方案	1 号尺寸方案	2 号尺寸方案
1	X	X	X
2	C_c	g_{m6}	g_{m6}
3	g_{m6}	C_c	C_c
4	C_L	C_L	C_L
5	R_z	R_z	R_z
6	g_{m1}	r_{ds7}	g_{m1}
7	E_1	c_{gs6}	E_1
8	g_{m4}	g_{m1}	g_{m4}
9	E_2	E_1	E_2
10	g_{m3}	g_{m4}	g_{m3}
11	g_{m2}	E_2	r_{ds7}
12	r_{ds1}	g_{m3}	g_{m2}
13	r_{ds7}	g_{m2}	r_{ds1}
14	c_{gs6}	r_{ds1}	c_{gs6}
15	r_{ds6}	r_{ds3}	r_{ds3}
16	r_{ds3}	r_{ds6}	r_{ds6}
17	c_{db2}	c_{db1}	c_{db2}
18	c_{db1}	c_{db7}	c_{db1}

3.4.4 简化符号化模型阶数比较

最后, 本小节给出了模型降阶前和降阶后的模型阶数的数据, 具体详见表3-6。模拟电路中模型的阶数由电路的零极点个数决定, 这里测试得到的零极点个数通过 HSPICE 的零极点分析得到。可以看到在所有的测试结果中都显示出显著的零极点个数减少。多数电路仅只剩下 3-4 个极点和 1-2 个零点, 这样规模的模型已经可以通过人为手工分析, 给电路工程师带来了许多便利。

表 3-6 电路符号化模型阶数比较

Table 3-6 Model order comparison between original and simplified circuits

电路类型	原始电路		简化电路	
	极点个数	零点个数	极点个数	零点个数
折叠共源共栅运放	9	8	3	1
调零补偿两级运放	6	6	4	2
Voltage Buffer 补偿两级运放	6	6	3	2
Current Buffer 补偿两级运放	7	7	4	2

3.4.5 符号化时域简化模型测试结果

首先我们针对两级运放电路进行了电路时域模型的建模，图3-29中展示了在不同 S 型函数作用下的电路的 Slew-Settling 行为。可以看到 $S_0(x)$ 、 $S_1(x)$ 和 $S_2(x)$ 在 Slew 阶段表现出了良好的对压摆率的估计行为。而 $S_4(x)$ 则非常适于对于电路稳定时间的估计。这里并没有画出相应的 $S_3(x)$ 的曲线，因为其 HSPICE 仿真并不收敛。可以发现图中的箭头标志了这些函数作用下的时域曲线也呈现了一定的顺序，且与上一小节中的次序一致。故可以预估 $S_3(x)$ 会取得最好的电路近似程度。

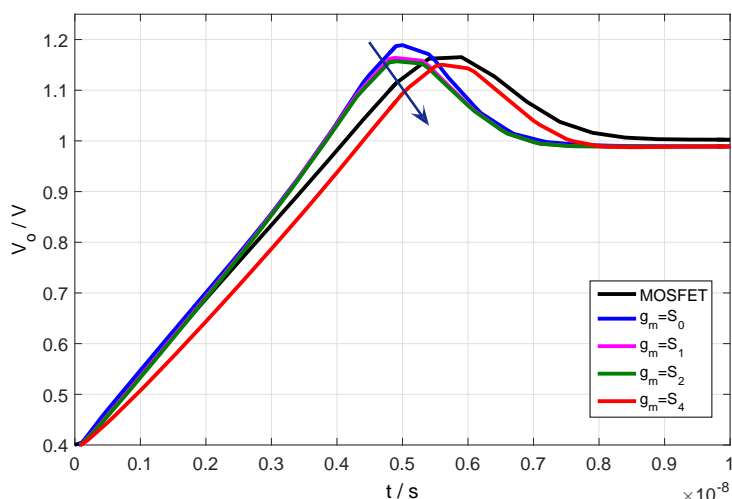


图 3-29 两级运放模型的 Slew-Settling 时域仿真结果
Fig 3-29 Simulation Results of slew-settling behavior for two-stage opamp model

在对折叠共源共栅运放的电路测试中，我们发现只有 $S_1(x)$ 的作用下，电路才可以顺利由 HSPICE 求解出来，仿真结果如图3-30。这里结果十分值得怀疑，因为其中出现了奇怪的折角，这在电路稳定的情况下不应该出现。同时，此时本方法生成的时域运放模型并不能很好地抓取电路的时域特性，说明时域模型生成方法仍存在问题，需要进一步分析验证。

3.5 本章小结

本章对本文的关键技术符号降阶模型的自动生成进行了详细介绍，并分析了加入各个流程原因以及需要考虑的特殊情况。该算法的有效性与稳定性得到了不同电路拓扑结构，不同尺寸参数情况下的验证。通过广泛大量的测试，可以看到本文提出的符号化降阶算法有效并且稳定地降低了电路模型的阶数，并且得到十分合理、易于电路工程师进

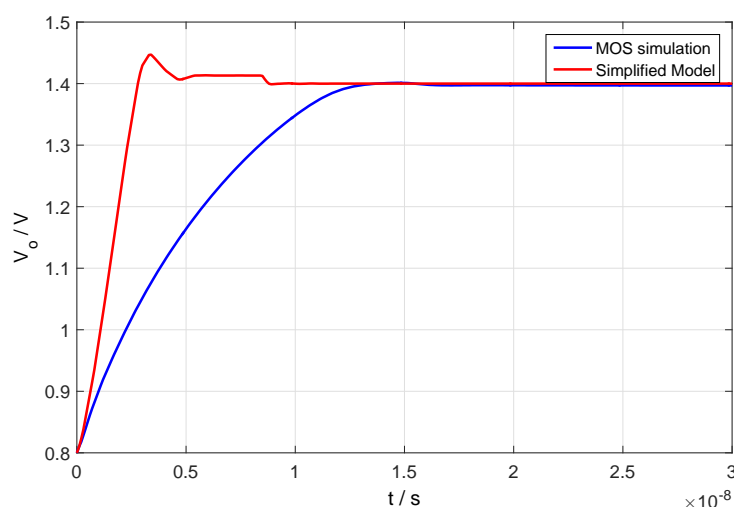


图 3-30 折叠共源共栅运放模型的 Slew-Settling 时域仿真结果

Fig 3-30 Simulation Results of slew-settling behavior for folded-cascode opamp model

行理解的简化电路小信号模型。这里采用的关键技术为符号化拓扑的简化的方法。这一方法相对于传统的一些符号化简化方法，总结下来，有以下几方面优势：

- a) 可以直接给出简化后的电路拓扑结构，易于模拟电路工程师进行分析。
- b) 在过程中，同时得到电路的符号化公式，可直接提供给用户做进一步分析。
- c) 得到的简化结果直接对应于原始电路中的电路元件，有助于做尺寸调整，避免了传统方法中简化结果无法对应会电路的问题。
- d) 区别于传统矩阵降阶方法，是一种启发式的符号化模型降阶方法。

另外，本章对电路的时域模型构造也进行了阐释，回顾了时域模型生成方法的相关历史，并提出了自己的时域模型简化方法，并对方法进行了测试。这种方法的优势在于其生成的模型是符号化的，并且大部分流程可以自动化，不需要过多的电路经验也可以对电路模型进行分析。可以看到目前符号化时域简化模型分析方法仅针对部分电路可以成功使用，但是仍有许多电路存在分析困难的情况。需要进一步通过非线性函数选取和系统层面理论分析对电路模型生成的方法进行论证。

第四章 多端口构造符号化约简方法探索

本章对多端口电路的构造进行探索, 由于期望将来能希望针对多端口情况下的模拟电路模型降阶方法提供基础。

4.1 共模抑制比与电源抑制比介绍

运算放大器除了差分增益响应以及相对应的相位变化, 别的设计指标也是极为关键的。在近几年兴起的生物电路中, 由于生物电信号的较为微弱的特性, 共模及电源噪声信号的影响对相关的运算放大器设计工作提出了很高的要求 [47–49]。目前对这两方面指标的自动化辅助工具的研究比较欠缺, 故十分需要有相应的工具对此种类型的电路分析工作进行指导与辅助。

根据 [41] 中的定义, 可根据图4–1中的示意给出共模抑制比 (Common Mode Rejection Ratio, $CMRR$) 和电源抑制比 (Power Supply Rejection Ratio, $PSRR$) 的定义。

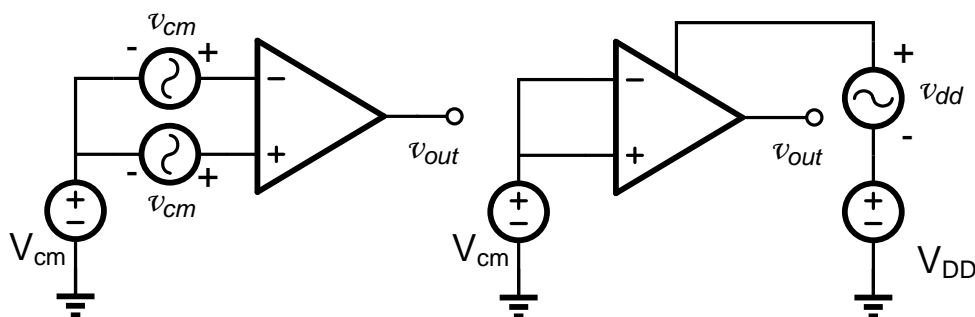


图 4–1 共模抑制比及电源抑制比定义

Fig 4–1 $CMRR$ and $PSRR$ definition

首先给出 $CMRR$ 的定义, 如图4–1左侧所示, 一个差分运算放大器有两个差分输入端, 如果在两个输入端输入叠加在共模信号上的差分信号。用 v_d 和 v_{cm} 代表信号的差模和共模部分, 那么我们可以得到运放的输出信号的差模部分有 $v_{out}^d = A_v v_d$; 而共模输出信号有 $v_{out}^{cm} = A_{cm} v_{cm}$ 。这里 A_v 和 A_{cm} 代表了运放的差模和共模增益。那么可以将 $CMRR$ 定义为差模增益与共模增益的比值, 如下式所示。

$$CMRR = \left| \frac{A_v}{A_{cm}} \right| \quad (4-1)$$

接着给出 **PSRR** 的定义，如图4-1右侧所示，运放的电源电压部分给上一个小信号激励 v_{dd} 。那么运放输出中由电源处提供的信号可以表示为 $v_{out}^{ps} = A_{ps}v_{dd}$ ，这里 A_{ps} 代表了运放的电源增益。故可以将 **PSRR** 定义为差模增益与电源增益的比值，如下式所示。

$$PSRR = \left| \frac{A_v}{A_{ps}} \right| \quad (4-2)$$

可以看到运放的 **CMRR** 和 **PSRR** 均是通过小信号计算得到，故可以使用符号化分析得到相应的结果。同时可以看到在分析 **CMRR** 和 **PSRR** 的过程中用到 3 个增益： A_v 、 A_{cm} 和 A_{ps} 。这三者的输出端均为测量电压，且为同一个端口的测量，故满足上一节所给出的定理2.1，那么可以使用单根的 **GPDD** 多端口的符号化分析方法对 **CMRR** 和 **PSRR** 进行分析。

4.2 **CMRR** 和 **PSRR** 在多端口构造下的计算方法

CMRR 和 **PSRR** 的分析总共涉及 3 个输入输出对：

1. 差分输入到输出端的差模电压增益 A_v
2. 共模噪声到输出端的共模电压增益 A_{cm}
3. 电源扰动到输出端的电源电压增益 A_{ps}

这三组输入输出对对应了不同的输入信号端口。因此在符号化构建的过程中，我们只需要在输入网表顶部给出 3 组 **VCVS** 的受控源，即可完成符号化网表的构建。这三组受控源这里分别用 X_v 、 X_{cm} 和 X_{ps} 三个符号给出，经过多端口符号化构造过程，可以得到类似于图4-2的 **GPDD** 结构。

这里的子 **GPDD** 结构由 N_{Av} 、 N_{cm} 、 N_{ps} 和 D 这四个 **GPDD** 节点为根组成。我们将使用他们的符号化结果来计算得到 **CMRR** 和 **PSRR** 的值。那么根据 **GPDD** 的求值规则，辅以符号化电路元件约减的性质，可以得到之前的三个增益可用下式计算得到。

$$A_v = \frac{1}{X_{Av}} = -s_1 s_4 s_5 s_6 \frac{f(N_{Av})}{f(D)} \quad (4-3)$$

$$A_{cm} = \frac{1}{X_{cm}} = -s_2 s_5 s_6 \frac{f(N_{cm})}{f(D)} \quad (4-4)$$

$$A_{ps} = \frac{1}{X_{ps}} = -s_3 s_6 \frac{f(N_{ps})}{f(D)} \quad (4-5)$$

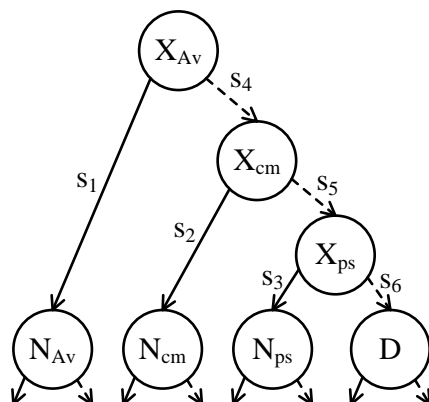


图 4-2 多端口方法构造 CMRR 及 PSRR 的 GPDD 结构
Fig 4-2 GPDD Structure for multi-port construction of CMRR and PSRR

同时，根据 CMRR 和 PSRR 的定义，可以简化 CMRR 和 PSRR 在 GPDD 中的计算如下：

$$CMRR = s_1 s_2 s_4 \frac{f(N_{Av})}{f(N_{cm})} \quad (4-6)$$

$$PSRR = s_1 s_3 s_4 s_5 \frac{f(N_{Av})}{f(N_{ps})} \quad (4-7)$$

值得注意的是，这里 GPDD 的构造过程只需要一次。只要知道所有电路符号元件的值，而后 A_v ，CMRR 和 PSRR 的数值结果可以通过自底向上遍历 GPDD 结构同时得到，节省了计算时间。这样的计算效率在之前提出的一些符号化方法中是不可行的 [8]。

根据第二章中介绍的敏感度计算方法，通过简单的推导即可知道，CMRR 和 PSRR 的敏感度也可以通过如下的计算得到：

$$Sens(CMRR, W_i) = Sens(A_v, W_i) - Sens(A_{cm}, W_i) \quad (4-8)$$

$$Sens(PSRR, W_i) = Sens(A_v, W_i) - Sens(A_{ps}, W_i) \quad (4-9)$$

这意味着我们需要求得三个增益的敏感度，即可得到 CMRR 和 PSRR 的敏感度。同时，由于所有的计算均在同一个 GPDD 上进行，所以可以高效地求得相应的结果。

4.3 共模抑制比与电源抑制比测试结果

4.3.1 多端口构造方法的双图决策树的时间空间复杂度比较

这里的测试电路选用了图3-12和图3-17中的折叠共源共栅运放和两级运放结构。

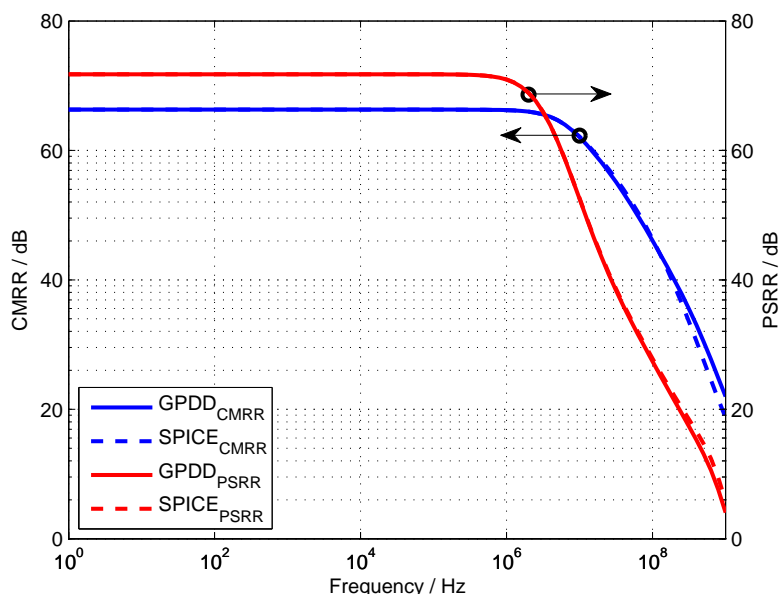


图 4-3 两级运放的 $CMRR$ 及 $PSRR$ 的频率响应结果
Fig 4-3 Frequency response for $CMRR$ and $PSRR$ in two-stage opamp

相应的小信号电路元件参数通过 HSPICE 的数值仿真结果得到，并构建相应的 GPDD 结构。计算得到的 $CMRR$ 和 $PSRR$ 的频率响应曲线在图4-3和图4-4中展示。可以看到仿真结构与 HSPICE 的数值结果十分吻合，所以 GPDD 符号化计算的结果是有效的。

表 4-1 两级运放的单独构造与多端口构造的时空性能比较

Table 4-1 Comparison between separated and multi-port constructions for the two-stage opamp

情况	GPDD	构造时间 (μs)
单独构造 A_v 的 GPDD	2251	522.0
单独构造 A_{cm} 的 GPDD	2615	408.7
单独构造 A_{ps} 的 GPDD	3933	632.0
总计	8799	1562.7
多端口构造 GPDD	4871	730.1
改进程度	44.6%	2.1x

我们也统计我们的软件实现性能来证明多端口构造得到的 GPDD 共享结果的优势。首先，我们针对三种增益 A_v 、 A_{cm} 和 A_{ps} 分别单独构造 GPDD 结构，并统计相应的构

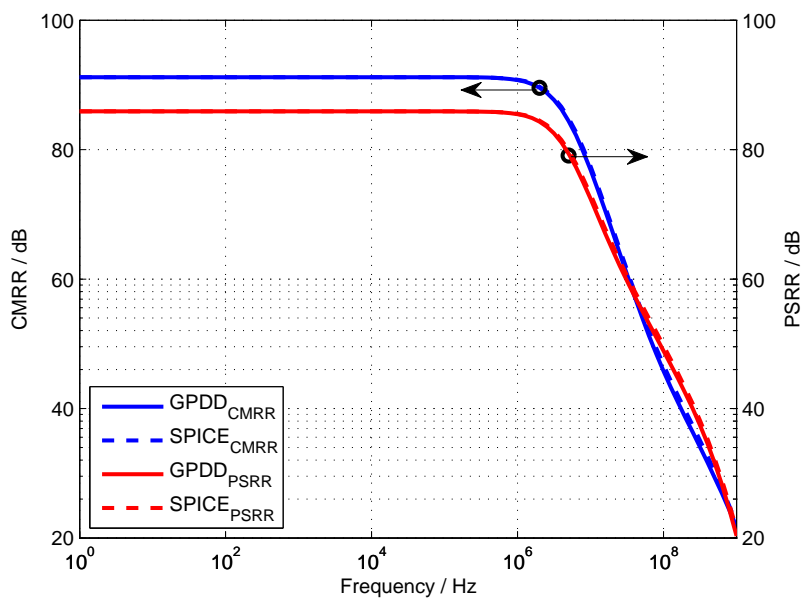


图 4-4 折叠共源共栅运放的 $CMRR$ 及 $PSRR$ 的频率响应结果
Fig 4-4 Frequency response for $CMRR$ and $PSRR$ in folded-cascode opamp

表 4-2 两级运放的单独构造与多端口构造的时空性能比较

Table 4-2 Comparison between separated and multi-port constructions for the folded-cascode opamp

情况	GPDD	构造时间 (s)
单独构造 A_v 的 GPDD	24228	155.2
单独构造 A_{cm} 的 GPDD	25706	122.0
单独构造 A_{ps} 的 GPDD	32918	173.6
总计	82852	450.8
多端口构造 GPDD	35424	78.7
改进程度	57.2%	5.7x

造时间和 GPDD 节点个数。其相应的结果在表4-1和表4-2中展现。总计一栏将上述三个增益分别的表现进行了求和，反映分别构造总复杂度。然后采用多端口方式构造同时对三个增益构造 GPDD。从表中的数据可以明显地看到多端口电路构造方式的优势，利用更少的时间和更少的计算空间完成了整体的计算。

4.3.2 敏感度方法进行电路参数优化共模电源抑制比

最后，我们对 CMRR 和 PSRR 的敏感度进行计算，并尝试通过敏感度进行电路优化。这里仍然针对两级运放进行分析，我们对其中的第一级和第二级的输入 MOS 管 M_1 和 M_6 的尺寸求取敏感度。相应的敏感度结果在图4-6和图4-6中展现。

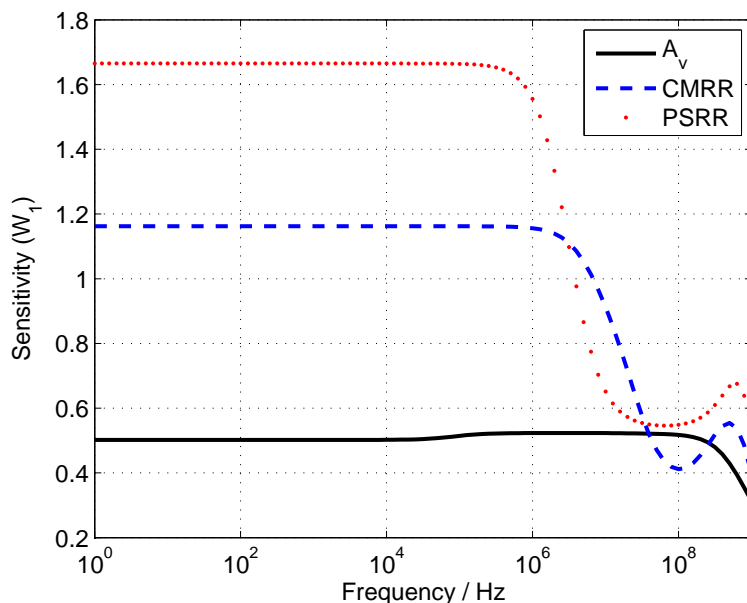


图 4-5 两级运放针对 W_1 的敏感度分析结果
Fig 4-5 Sensitivity results of W_1 in two-stage opamp

根据图中所展现的结果，我们看到 W_1 与三种电路指标 A_v 、CMRR 和 PSRR 均正相关，故增大 W_1 将同时增大三种性能指标。然而 W_6 情况则不同，如增大 W_6 的话，差分增益 A_v 会有所提升，CMRR 基本保持不变，而 PSRR 则会下降。

我们在实际对运放进行了尺寸优化 (W_1 从原有的 $20\mu m$ 变为 $30\mu m$ ， W_6 从原有的 $30\mu m$ 变为 $31\mu m$) 后，将相应的数据变化记录于表4-3和表4-4中。

可以看到在尺寸调整后，三种电路指标 A_v 、CMRR 和 PSRR 均按预期发生了变化。同时由于敏感度的定义，即可知道电路性能变化比率与敏感度基本成正比，这在表中的数据也得到了相应的验证。高敏感度带来了更快的性能增长。另外，还应观察到在

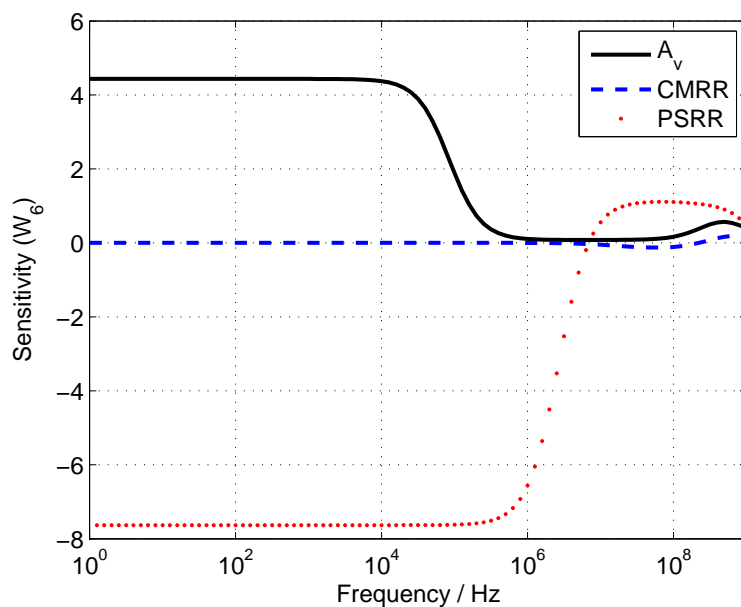


图 4-6 两级运放针对 W_6 的敏感度分析结果
Fig 4-6 Sensitivity results of W_6 in two-stage opamp

表 4-3 W_1 的 DC 敏感度分析结果及优化

Table 4-3 Sensitivity analysis at DC of W_1

指标	$W_1 = 20\mu m$		$W_1 = 30\mu m$		比率
	值	敏感度	值	敏感度	
A_v	2.84K (69.1dB)	0.502	3.41K (70.7dB)	0.401	+20.0% (+1.6dB)
$CMRR$	2.07K (66.3dB)	1.162	3.05K (69.7dB)	0.802	+47.3% (+3.4dB)
$PSRR$	3.87K (71.8dB)	1.665	6.92K (76.8dB)	1.237	+78.8% (+5.0dB)

表 4-4 W_6 的 DC 敏感度分析结果及优化

Table 4-4 Sensitivity analysis at DC of W_6

指标	$W_6 = 30\mu m$		$W_6 = 31\mu m$		比率
	值	敏感度	值	敏感度	
A_v	2.84K (69.1dB)	4.43	3.04K (69.7dB)	-0.20	+7.04% (+0.6dB)
$CMRR$	2.07K (66.3dB)	6.1e-8	2.07K (66.3dB)	9.4e-8	+0% (+0dB)
$PSRR$	3.87K (71.8dB)	-7.63	3.32K (70.4dB)	-2.83	-14.2% (-1.4dB)

电路尺寸改变后电路的敏感度也发生了变化。例如，表4-4中的有关 A_v 的敏感度已经由正值变为了负值，说明继续增大 W_6 将对电路的增益带来不好的效果。

4.4 本章小结

本章尝试了对一种用于 **CMRR** 和 **PSRR** 的计算的符号化多端口电路构造计算方法进行了探索，以期望用于将来的针对 **CMRR** 和 **PSRR** 的电路简化方法。首先，介绍了 **CMRR** 和 **PSRR** 的基本概念与相应的计算流程，并利用敏感度分析方法对现有电路进行调整优化。最后，通过算法的时间空间的复杂度，验证了 **GPDD** 构造的高效性，并用敏感性分析辅助电路优化等测试证明了这种构造方法的有效性。

第五章 结束语

5.1 主要工作与创新点

本文主要对电路的简化小信号模型生成算法进行了介绍,并尝试将其中的内容应用到时域模型的生成。并对 **CMRR** 以及 **PSRR** 分析进行了探索,为进一步的研究提供基础。下面将针对这三方面做逐一总结。

低阶符号化电路模型的自动生成是本文的开创性工作。过去的符号化简化工作往往仅针对电路的传输函数表达式,而不是针对电路的拓扑结构。相比的单纯的符号化公式,本文生成的简化的电路拓扑结构可以带给电路设计工程师更多电路的信息。本文通过挖掘电路拓扑结构与元件取值的关系引出了符号化构造方法 **GPDD** 与电路拓扑结构的关系。因为这种关系, **GPDD** 承担了算法的主要计算职责。我们给出了符号化简化的主要流程,并特别提出了用于电路简化的元件重要性的概念,同时辅以元件的预处理方法,可以自动化系统化地生成可供人理解的小信号电路。通过对特殊情况的讨论,提高了算法的可靠性,避免程序中途中止。文中给出详细的测试结果以说明该算法的有效性与鲁棒性,在针对各种电路拓扑和电路参数尺寸的情况下,本文的算法都可以给出相应的符号化小信号模型。这一工作将大大方便模拟电路工程师对电路的分析工作,因为模型的自动生成,工程师可以直接定位电路的关键问题所在。另外在模拟电路 **IP** 化的进程中,本文中提出的方法可以尝试作为模拟电路的高层模型,以在不丢失大量精度的前提下,加速系统级的仿真的速度。

时域电路模型的生成通过在自动生成的简化小信号电路加入电流饱和限制以形成可以进行大信号分析的宏模型。本文创新地提出了对电路中多级均加入饱和电流限制的方法,并给出了多种非线性函数,用以形成饱和限制电流。这一模型生成方法目前的测试表明,已存在其可适用的情况;如需广泛使用,需进一步优化模型。

在如今电源电压日益降低的年代, **CMRR** 和 **PSRR** 成为了衡量电路的抗共模噪声和电源干扰的能力的重要指标。本文采用了多端口的 **GPDD** 构造方式对电路的差模增益、共模抑制比和电源抑制比进行了分析。我们证明了多端口构造的适应性条件,为同时分析多个端口的电路的提供了良好的基础。通过 **GPDD** 由于节点共享造成的高效性,为同一时间分析三种电路参数提供了可能。本文还通过符号化敏感度求值等方法计算得到了电路优化方向,以同时优化多个电路参数,并更好进行折衷。进一步将来可以尝试多端口的降阶模型生成方法。

5.2 后续研究方向

本文中所涉及的多个话题均有可以进一步研究的价值，如以下可能的研究方向：

- a)* 降阶模型生成中的元件重要性仍有许多可以挖掘的地方，如在重要性计算中单纯考虑运放的 **CMRR** 或者 **PSRR**，或许可以得到用于分析 **CMRR** 和 **PSRR** 的对应的小信号电路。若同时结合多端口的构造策略，尝试得到能同时考虑多个运放指标的电路模型。这一方面实验室中已由本科同学进一步开展研究，并讨论其在电路失配情况下的应用。
- b)* **CMRR** 和 **PSRR** 的自动优化流程可以在考虑敏感度的情况下，通过模拟退火等多种优化方法尝试考虑多个电路性能指标的模拟集成电路的自动优化流程。
- c)* 多端口的构造方式可以应用电路的噪声分析中，因为考虑电路噪声时，一般都考虑某一节点上的噪声，但整个电路中可能有多处噪声源，这正适应于本文提出的构造哦啊条件。
- d)* 分析时域模型的进一步拓展可能性，尝试从理论上对生成时域模型进行分析。

附录 A 双图决策树对消项说明

本附录介绍了两种类型的符号化对消项，并介绍了其在 GPDD 这种符号化构造方法中的特性，并介绍了对消项的存在对电路简化中影响。

A.1 符号化对消项概念介绍

我们知道构造符号化表达式的过程中，我们会将没有可以互相抵消的项约去，以得到最简洁的电路表达式。这样有利于电路分析，抛开无关紧要的项，同时也减少了运算过程中的复杂度，减少了不必要的数值误差。符号化对消项主要有两类：加性对消项和公因子对消项。加性对消项指的是可以在符号化表达中正负抵消的项。而公因子对消项指的是在符号化表达式中成为分子分母共同的因子可以相互抵消的项。比如，在如下所示的公式中：

$$H(s) = \frac{ab + bc - ab}{bd} \quad (\text{A-1})$$

这里 ab 和 $-ab$ 为加性对消项，而这里的 b 也是公因子对消项。加性对消项会在对符号化公式求值的过程中引入不必要的数值误差，而公因子对消项除了数值误差意外，也在我们进行符号化约减的过程中引入了困难。

过去过去大量的符号化方法中，都存在对消项的问题，由于大量冗余的符号化对消项的存在，大大降低了内存空间的使用效率，同时引入了本应在符号化计算过程中避免的数值计算误差。之所以没有通过一些方法匹配得到符号化表达中的对消项，主要原因在于进行这种匹配时间复杂度非常高，需要对所有的符号化项进行两两比较，基本不可行。[16] 中宣称，GPDD 结构中不存在符号化对消项。经笔者验证发现，GPDD 当中的确不存在加性符号化对消项，但有可能出现公因子对消项。下面针对这两种不同的对消项给出证明。

A.2 双图决策树无加性对消项证明

我们知道 GPDD 中每一条从根节点出发，到达 1 结点的路径可以构造出一个符号化项，称这样的路径为 P ，由 P 构造得到的符号化项为 T_P 。由于 GPDD 中每一层仅一个符号，且每个符号有且仅有两种在 P 中出现可能性（包含或不包含），故不同的路径 P 必然不同。如果我们可以证明路径 P 和符号化项 $|T_P|$ （加上绝对值去除符号的影响）

构成的两个集合一一对应，那么由于 P 两两不相同，且两个集合均为有限集合，那么 $|T_P|$ 也必然两两不相同，从而不可能出现加性对消项。那么证明 GPDD 中不存在加性对消项的任务就转变为了证明两种集合一一对应的问题。

考虑某个 GPDD 构造的符号表共有 N 个符号，分别为 $\mathbf{S} = \{S_1, S_2, \dots, S_N\}$ 。一个符号化项 $|T_P|$ 可以写作这样的形式。

$$|T_P| = \prod_{i=1}^N S_i^{\alpha_i} \quad (\text{A-2})$$

其中这里的 α_i 可以由路径 P 的边的性质决定，如下所示。

$$\alpha_i = \begin{cases} 1 & S_i \text{ 下连接的边在 } P \text{ 中为实线} \\ 0 & S_i \text{ 下连接的边在 } P \text{ 中为虚线或不存在} \end{cases} \quad (\text{A-3})$$

这样，我们根据 T_P 的定义可以直接得到 $P \rightarrow |T_P|$ 的映射关系，如果可以找到一种方法形成的 $|T_P| \rightarrow P$ 映射，那么 $P \leftrightarrow |T_P|$ 之间的一一映射则可完成。

假设路径 P 由这样的边集组成 $\mathbf{E} = \{E_1, E_2, \dots, E_N\}$ 。其中 E_i 为元件 S_i 这个节点下连接的边。如 S_i 节点不存在，则忽略该 E_i 。给出任意一个 T_P ，我们可以决定 P 的连接情况如下所示。

$$E_i = \begin{cases} S_i \text{ 下的实线} & \alpha_i = 1 \\ S_i \text{ 下的虚线} & \alpha_i = 0 \end{cases} \quad (\text{A-4})$$

故这样形成了 $|T_P| \rightarrow P$ 的映射，故由 P 和 T_P 组成的两个集合之间一一映射。故可以有以下定理。

定理 A.1. 双图决策图结构中不存在加性对消项。

从上述证明过程中，可以看到，不仅 GPDD 中不存在加性对消项，而且每个符号项前的系数仅为 ± 1 。

A.3 双图决策树公因子对消项情况

虽然，GPDD 中不存在加性对消项，但是却有可能出现公因子对消项。我们通过举反例的方式对这一点进行说明，并介绍其在电路简化过程中造成的问题。

A.3.1 公因子对消项反例

我们考虑这样一个如图A-1所示的两级电路。但我们其实只观测第一级的电压响应，所以第二级所用的元件对第一级没有任何贡献。

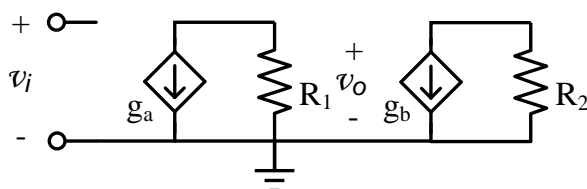


图 A-1 存在公因子对消项的示例电路
Fig A-1 Circuit example with common factor

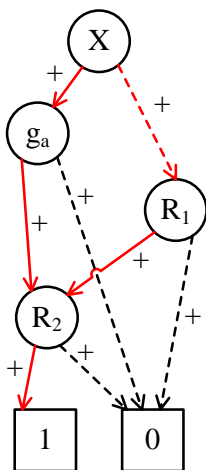


图 A-2 存在公因子对消项的示例电路
Fig A-2 Circuit example with common factor

但是可以在图A-2中看到，其实其中第二级点元件 R_2 仍然存在于 GPDD 结构中。根据这样的 GPDD 结构，经过计算，我们可以得到如下的电路传输函数：

$$H(s) = -\frac{g_a G_2}{G_1 G_2} \quad (\text{A-5})$$

可以看到这里 R_2 形成了符号化公因子对消项。因为 GPDD 中所有到 1 结点的路径均经过 R_2 节点，并且 R_2 节点的右儿子均为零。

A.3.2 对自动低阶模型生成中电路图约减过程的影响

在图的约减过程中，根据图A-2的情况下，如果我们考虑 R_2 的导纳为零，即断路情况。根据我们计算有元件取极限值的算法，GPDD 在 R_2 这个节点会选择右儿子节点

的值，在这里即为 0。可以看到这导致整个 GPDD 结构中没有一条到 1 结点的路径，那么必然会在计算传输函数过程中，得到 $0/0$ 的结果，从而造成了计算的错误。如果将 R_2 短路即可避免这一情况。

这种情况如果在一开始元件关键性计算的过程就发生，那么会立刻归为 3.2.2.1 中的第一种特殊特殊情况。如果是在发生了多个元件删去后的结果，那么就会成为 3.2.2.2 中第二种特殊情况。

参考文献

- [1] R. Saleh, S. Wilton, S. Mirabbasi, A. Hu, M. Greenstreet, G. Lemieux, P. P. Pande, C. Grecu, and A. Ivanov, "System-on-Chip: Reuse and Integration," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1050–1069, June 2006.
- [2] G. Agarwal, and P. Bare, "Why is design automation and reuse of analog designs increasingly trailing the digital world?," in *Proc. IEEE Int'l Conf. on VLSI Design (VLSID)*, Jan. 2009, p. 17.
- [3] Z. Li, L. Luo, and J. Yuan, "A study on analog IP blocks for mixed-signal SoC," in *Proc. Int'l Conf. on ASIC (ICASIC)*, Oct. 2003, pp. 564–567.
- [4] G. A. Rincon-Mora, Analog integrated circuit design: why? [Online]. Available: http://users.ece.gatech.edu/~rincon/classes/ana_why.pdf.
- [5] T. H. Lee, "Tales of the continuum: a subsampled history of analog circuits," *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 4, pp. 38–51, Fall 2007.
- [6] L. W. Nagel, and D. O. Pederson, *SPICE: Simulation program with integrated circuit emphasis*. Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1973.
- [7] P.-M. Lin, *Symbolic Network Analysis*. New York: Elsevier, 1991.
- [8] G. Gielen, H. Walscharts, and W. Sansen, "ISAAC: a symbolic simulator for analog integrated circuits," *IEEE J. Solid-State Circuits*, vol. 24, no. 6, pp. 1587–1597, Dec. 1989.
- [9] G. Nebel, U. Kleine, and H.-J. Pfeleiderer, "Symbolic pole/zero calculation using SANTAFE," *IEEE J. Solid-State Circuits*, vol. 30, no. 7, pp. 752–761, July 1995.
- [10] M. H. Fino, and L. J. Mourão, "SymbSI-a program for the symbolic signal flow graph generation of switched current circuits," in *Proc. IEEE Int'l Conf. on Electronics, Circuits and Systems (ICECS)*, Sept. 1998, vol. 3, pp. 211–214.
- [11] S.-D. Shieu, and S.-P. Chan, "Topological formulation of symbolic network functions and sensitivity analysis of active networks," *IEEE Trans Circuits and Systems*, vol. 21, no. 1, pp. 39–45, Jan. 1974.

- [12] P. Wambacq, G. Gielen, and W. Sansen, “Symbolic network analysis methods for practical analog integrated circuits: a survey,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 10, pp. 1331–1341, Oct. 1998.
- [13] R. E. Bryant, “Graph-based algorithms for boolean function manipulation,” *IEEE Trans. Computers*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.
- [14] C.-J. Shi, and X.-D. Tan, “Canonical symbolic analysis of large analog circuits with determinant decision diagrams,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 1, pp. 1–18, Jan. 2000.
- [15] 陈薇薇, 符号化模拟电路仿真器的实现与应用 [硕士论文], 上海交通大学, 2007年。
- [16] G. Shi, “Graph-pair decision diagram construction for topological symbolic circuit analysis,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 2, pp. 275–288, Feb. 2013.
- [17] G. Shi, S. X.-D. Tan, and E. Tlelo-Cuautle, *Advanced Symbolic Analysis for VLSI Systems: Methods and Applications*. Berlin: Springer, 2014.
- [18] G. Shi, “A survey on binary decision diagram approaches to symbolic analysis of analog integrated circuits,” *Analog Integrated Circuits and Signal Processing*, vol. 74, no. 2, pp. 331–343, 2013.
- [19] H. Xu, G. Shi, and X. Li, “Hierarchical exact symbolic analysis of large analog integrated circuits by symbolic stamps,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2011, pp. 19–24.
- [20] X. Li, H. Xu, G. Shi, and A. Tai, “Hierarchical symbolic sensitivity computation with applications to large amplifier circuit design,” in *Proc. IEEE Int’l Symp. on Circuits and Systems (ISCAS)*, May 2011, pp. 2733–2736.
- [21] Y. Song, and G. Shi, “Hierarchical graph reduction approach to symbolic circuit analysis with data sharing and cancellation-free properties,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2012, pp. 541–546.
- [22] G. Shi, and X. Meng, “Variational analog integrated circuit design via symbolic sensitivity analysis,” in *Proc. IEEE Int’l Symp. on Circuits and Systems (ISCAS)*, May 2009, pp. 3002–3005.

- [23] B. Weng, and G. Shi, "Design optimization of MOS operational amplifiers using finite difference sensitivity," in *Proc. Int'l Symp. on Integrated Circuits (ISIC)*, Dec. 2011, pp. 277–280.
- [24] J. Chen, G. Shi, A. Tai, and F. Lee, "A size sensitivity method for interactive MOS circuit sizing," in *Proc. IEEE Int'l New Circuits and Systems Conference (NEWCAS)*, June 2012, pp. 169–172.
- [25] H. Zhang, and G. Shi, "Symbolic behavioral modeling for slew and settling analysis of operational amplifiers," in *Proc. IEEE Int'l Midwest Symp. on Circuits and Systems (MWSCAS)*, Aug. 2011, pp. 1–4.
- [26] A. Zhang, and G. Shi, "A symbolic SC integrator model for fast time-response simulation," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, May 2015, pp. 1949–1952.
- [27] J. Cheng, G. Shi, A. Tai, and F. Lee, "Symbolic fault modeling for switched-capacitor circuits," in *Proc. IEEE Region 10 Conference (TENCON)*, Oct. 2013, pp. 1–4.
- [28] J. Cheng, and G. Shi, "Symbolic computation of SNR for variational analysis of sigma-delta modulator," in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2014, pp. 443–448.
- [29] J. Cheng, G. Shi, and A. Zhang, "A fast SNR estimation method for sigma-delta modulator design," in *Proc. IEEE Region 10 Conference (TENCON)*, Oct. 2013, pp. 1–4.
- [30] W. Daems, G. Gielen, and W. Sansen, "Circuit simplification for the symbolic analysis of analog integrated circuits," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 4, pp. 395–407, Apr. 2002.
- [31] J.-J. Hsu, and C. Sechen, "DC small signal symbolic analysis of large analog integrated circuits," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 12, pp. 817–828, Dec. 1994.
- [32] F. V. Fernández, P. Wambacq, G. Gielen, A. Rodríguez-Vázquez, and W. Sansen, "Symbolic analysis of large analog integrated circuits by approximation during expression generation," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, May 1994, pp. 25–28.
- [33] X.-D. Tan, and C.-J. R. Shi, "Interpretable symbolic small-signal characterization of large analog circuits using determinant decision diagrams," in *Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE)*, Mar. 1999, pp. 448–453.

- [34] V. Živković, and P. Petković, “Topological level of symbolic approximation in analog circuit design,” in *Proc. Int’l Conf. on Microelectronics (ICMEL)*, Sept. 1995, vol. 2, pp. 521–524.
- [35] G. Shi, B. Hu, and C.-J. R. Shi, “On symbolic model order reduction,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1257–1272, July 2006.
- [36] 胡翰彬, 模拟电路符号化表达的增量式构造研究 [学士论文], 上海交通大学, 2013 年。
- [37] A. Pugliese, F. Amoroso, G. Cappuccino, and G. Cocorullo, “Analysis of the impact of high-order integrator dynamics on SC Sigma-Delta modulator performances,” *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 57, no. 3, pp. 618–630, Mar. 2010.
- [38] M. Yavari, N. Maghari, and O. Shoaiei, “An accurate analysis of slew rate for two-stage CMOS opamps,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 164–167, Mar. 2005.
- [39] H. Rezaee-Dehsorkh, N. Ravanshad, R. Lotfi, and K. Mafinezhad, “Modified model for settling behavior of operational amplifiers in nanoscale CMOS,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 56, no. 5, pp. 384–388, May 2009.
- [40] C. Chuang, “Analysis of the settling behavior of an operational amplifier,” *IEEE J. Solid-State Circuits*, vol. 17, no. 1, pp. 74–80, Feb. 1982.
- [41] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed. Singapore: John Wiley & Sons, Inc., 2001.
- [42] P. E. Allen, and D. R. Holberg, *CMOS Analog Integrated Circuit Design*. New York: Oxford University Press, 2002.
- [43] A. Grasso, D. Marano, G. Palumbo, and S. Pennisi, “Improved reversed nested miller frequency compensation technique with voltage buffer and resistor,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 54, no. 5, pp. 382–386, May 2007.
- [44] S. Pakala, M. Manda, P. Surkanti, A. Garimella, and P. Furth, “Voltage buffer compensation using flipped voltage follower in a two-stage CMOS op-amp,” in *Proc. IEEE Int’l Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2015, pp. 1–4.
- [45] G. Palmisano, and G. Palumbo, “A compensation strategy for two-stage CMOS opamps based on current buffer,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 3, pp. 257–262, Mar. 1997.

- [46] J. Mahattanakul, “Design procedure for two-stage CMOS operational amplifiers employing current buffer,” *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 52, no. 11, pp. 766–770, Nov. 2005.
- [47] A. Harb, and M. Sawan, “New low-power low-voltage high-CMRR CMOS instrumentation amplifier,” in *Proc. IEEE Int’l Symp. on Circuits and Systems (ISCAS)*, July 1999, vol. 6, pp. 97–100.
- [48] R. Abdullah, and E. Sánchez-Sinencio, “A biopotential amplifier with dynamic capacitor matching for improved CMRR,” *Analog Integrated Circuits and Signal Processing*, vol. 82, no. 1, pp. 47–55, Jan. 2015.
- [49] P. M. Furth, S. H. Pakala, A. Garimella, and C. Mohan, “A 22dB PSRR enhancement in a two-stage CMOS opamp using tail compensation,” in *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2012, pp. 1–4.

致 谢

攻读学位期间发表的学术论文

- [1] 第一作者, "Topological symbolic simplification for analog design," in *Proc. IEEE Int'l Symp. on Circuits and Systems (ISCAS)*, 2015, pp. 2644-2647.
- [2] 第一作者, "Incremental symbolic construction for topological modeling of analog circuits," in *Proc. IEEE Int'l Conf. on ASIC (ASICON)*, 2013, pp. 1-4.