

申请上海交通大学研究生学位论文

互连线统计时序的符号化分析方法

姓 名：于雪红
导 师：施国勇 教授
学 校：上海交通大学
院 系：微电子学院
专 业：电路与系统

此研究由上海市浦江人才基金资助(No. 07pj14053)

上海交通大学微电子学院

2008 年 12 月

Ms. Dissertation Submitted to Shanghai Jiao Tong University

**STATISTICAL INTERCONNECT TIMING ANALYSIS
BASED ON SYMBOLIC METHOD**

Author: Xuehong YU

Advisor: Prof. Guoyong SHI

Specialty: Circuit and System

School of Microelectronics
Shanghai Jiao Tong University
Shanghai, PR.China

December, 2008

上海交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密，在_____年解密后适用本授权书。

本学位论文属于

不保密。

(请在以上方框内打“√”)

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

互连线统计时序的符号化分析方法

摘要

在深亚微米特别是纳米级工艺下，制造过程中的工艺偏差不能被忽略，物理版图中的互连线物理参数必须被建模为随机变量，从而使得互连线延迟也成为具有一定概率分布的随机变量。为了快速准确的得到延迟的概率分布，本文提出了一种高效准确的互连线延迟统计建模算法，基于创新性的符号化延迟算法，得到互连线延迟关于互连线物理参数偏差的闭合形式表达式，进而采用二阶非线性模型进行近似，最后运用特征函数和傅立叶变换直接计算得到互连线延迟的概率分布。该算法高效准确，避免了低效的蒙特卡罗模拟。实验证明，该算法可以处理含有三十万个以上节点的大规模互连线网络结构，得到的延迟概率分布与蒙特卡罗结果相比有很好的精度，误差稳定在 1% 以内。

关键词： 符号化延迟，互连线延迟，统计时序分析，二阶时序模型

STATISTICAL INTERCONNECT TIMING ANALYSIS BASED ON SYMBOLIC METHOD

ABSTRACT

An efficient yet accurate statistical timing model for IC interconnects is proposed in this paper. When considering process variations, interconnect physical parameters must be modeled as random variables, which makes interconnect delays become random variables with certain PDF. In order to get the PDF of the interconnect delay, a creative new symbolic delay calculation algorithm, expressing delays as closed functions of interconnect physical parameters, is employed. And based on closed-form delay metrics, delay is represented directly as function of physical parameters. Then by using a quadratic model to approximate the delay, we managed to obtain its PDF directly through Characteristic Function and Fourier Transform technique. This method can be used to handle interconnect circuits with more than 300,000 nodes with a high efficiency. And as compared to Monte Carlo Simulation, this method has a fairly good accuracy with only 1% error.

Keywords: symbolic delay, symbolic analysis, statistical timing analysis, interconnect delay, quadratic timing model

目 录

摘 要	I
ABSTRACT.....	II
第一章 绪论.....	8
1.1 工艺制造偏差问题.....	8
1.2 统计时序分析概述.....	10
1.3 符号化分析方法.....	12
1.4 本章小结.....	13
第二章 互连线时序分析算法.....	14
2.1 确定性时序分析方法.....	14
2.1.1 基于模型降阶算法的延迟估算.....	14
2.1.2 基于矩的延迟度量算法.....	16
2.1.3 延迟算法总结.....	21
2.2 统计时序分析方法.....	22
2.2.1 基于参数化模型降阶的统计延迟估算.....	22
2.2.2 基于延迟度量的统计延迟估算.....	25
2.3 本章小结.....	28
第三章 符号化互连线统计时序分析方法.....	29
3.1 互连线电气参数的统计建模.....	30
3.2 符号化矩算法.....	33
3.2.1 路径追溯算法.....	33
3.2.2 符号化矩求解原理.....	38
3.2.3 符号化矩算法实现.....	44
3.3 延迟的二阶近似模型.....	46
3.4 延迟概率分布的求解算法.....	49
3.5 算法流程及优势.....	50
3.6 本章小结.....	52

第四章 实验与结果分析	53
4.1 算法正确性验证.....	53
4.1.1 符号化矩算法验证.....	53
4.1.2 延迟度量算法比较.....	55
4.2 算法效率测试.....	65
4.2.1 单条互连线RC电路.....	66
4.2.2 RC树状互连线网络.....	68
4.2.3 小结.....	70
4.3 电路测试结果及分析.....	70
4.3.1 一般结构的互连线延迟测量.....	70
4.3.2 时钟树型电路的延迟测量.....	74
4.3.3 不同的最大偏差范围测试.....	78
4.3.4 偏差间的相关性影响测试.....	79
4.4 与已有方法的比较.....	80
4.4.1 基于线性模型的统计时序分析方法.....	81
4.4.2 基于AWE的统计时序分析算法.....	82
4.5 本章小结.....	82
第五章 总结与展望	83
5.1 总结.....	83
5.2 研究展望.....	83
参 考 文 献	84
攻读研究生学位期间已发表或录用的论文	88
致 谢	89

图 片 目 录

图 3-1 简单RC分布式模型	30
Fig.3-1 A simple RC distributed model.....	30
图 3-2 互连线物理参数示意图.....	32
Fig.3-2 Physical parameters of interconnect.....	32
图 3-3 简单RLC树状电路.....	34
Fig.3-3 A simple RLC tree-like circuit.....	34
图 3-4 RC树状电路示例	34
Fig.3-4 A simple RC tree-like circuitl	34
图 3-5 s域等价电路.....	35
Fig.3-5 Equivalent circuit in s domain.....	35
图 3-6 电容等价于电流源.....	36
Fig.3-6 Capacitance modeled as current source.....	36
图 3-7 直流计算等效电路.....	37
Fig.3-7 Equivalent DC analysis circuit	37
图 3-8 由q阶矩计算q+1阶矩的直流等效电路.....	37
Fig.3-8 DC equivalent circuit for computing $q+1^{\text{th}}$ order moment from q^{th} order moment	37
图 3-9 简单RC树状电路	39
Fig.3-9 A simple RC tree.....	39
图 3-10 电容树抽象结构.....	40
Fig.3-10 Extracted capacitance tree structure	40
图 3-11 矩决策树结构.....	42
Fig.3-11 Moment Decision Diagram structure.....	42
图 3-12 求解符号化矩的混合数据结构.....	43
Fig.3-12 Hybrid structure for calculation of symbolic moments	43
图 3-13 互连线统计时序的符号化分析方法.....	51
Fig.3-13 Symbolic analysis method for statistical timing modeling of interconnects	51
图 4-1 符号化算法一阶矩结果验证.....	54
Fig.4-1 Verification of symbolic moment algorithm.....	54
图 4-2 30 节点单条互连线延迟比较.....	56
Fig.4-2 Delay of single line including 30 nodes	56
图 4-3 含 30 节点的单条互连线各节点延迟的相对误差.....	57
Fig.4-3 Relative Error of delay at each node for single line with 30 nodes	57
图 4-4 100 节点单条互连线延迟比较.....	58

Fig.4-4 Delay of single line including 100 nodes	58
图 4-5 100 个节点的单条互连线不同延迟度量方法所得各点延迟的相对误差	59
Fig.4-5 Relative Error of Delay at each node for single line with 100 nodes	59
图 4-6 100 个节点的单条互连线各点延迟的相对误差放大图	60
Fig.4-6 Magnified Curve of Delay Relative Error	60
图 4-7 存在分支的 30 节点互连线网络各点延迟比较	61
Fig.4-7 Delay of interconnect network with fanouts including 30 nodes	61
图 4-8 30 节点互连线网络各点延迟的相对误差	62
Fig.4-8 Delay of interconnect network including 30 nodes	62
图 4-9 存在分支的 100 节点互连线网络延迟比较	63
Fig.4-9 Delay of interconnect network with fanouts including 100 nodes	63
图 4-10 100 个节点互连线网络不同延迟度量方法所得各点延迟的相对误差	64
Fig.4-10 Relative Error of Delay at each node for interconnect network with 100 nodes	64
图 4-11 四支路互连线电路	71
Fig.4-11 Interconnect with four chains	71
图 4-12 各节点延迟概率分布结果	73
Fig.4-12 Delay PDF of specified nodes	73
图 4-13 互连线网络各节点延迟PDF的误差	74
Fig.4-13 Error of delay PDF at each node	74
图 4-14 三阶段时钟型电路	76
Fig.4-14 3-stage Clock-tree Circuit	76
图 4-15 三阶段时钟型电路节点 302 的概率密度分布	77
Fig.4-15 PDF at node 302 of a 3-stage Clock-tree Circuit	77
图 4-16 三阶段时钟型电路节点 702 的概率密度分布	77
Fig.4-16 PDF at node 702 of a 3-stage Clock-tree Circuit	77
图 4-17 不同最大偏差下节点 31 的延迟概率分布	79
Fig.4-17 Different Delay PDF at node 31 under different 3σ	79
图 4-18 协方差 0.5 下节点 31 的延迟概率分布	80
Fig.4-18 Delay PDF at node 31 under W,H covariance of 0.5	80
图 4-19 线性方法与二阶近似模型的比较	81
Fig.4-19 Comparison of Linear Model with Quadratic Model	81

表 格 目 录

表 4-1 30 节点单条互连线延迟的相对误差统计	56
表 4-2 100 节点单条互连线延迟的相对误差统计	58
表 4-3 30 节点互连线网络延迟的相对误差统计	63
表 4-4 100 节点互连线网络延迟的相对误差统计	63
表 4-5 单条互连线末节点延迟二阶模型求解时间比较	66
表 4-6 单条互连线末节点延迟二阶模型求解时间比较	67
表 4-7 互连线网络末节点延迟二阶模型求解时间比较	68
表 4-8 互连线网络末节点延迟二阶模型求解时间比较	69
表 4-9 不同规模的互连线网络节点延迟PDF的平均误差	74

第一章 绪论

随着大规模集成电路进入深亚微米工艺时代，芯片尺寸越来越小，互连线层次越来越多，互连线的长度也达到了前所未有的量级。《国际半导体技术蓝图》(ITRS)^[1]给出的数据显示，在 2×2 平方毫米的芯片上，互连线的总长度可以达到 10km 的数量级。互连线的工艺尺寸越来越小，然而长度却相对的在增长，由此带来的互连线电阻和电容的增大使得互连线延迟已经超过门延迟，成为电路延迟的主要部分。这一变化使得互连线设计成为影响芯片性能、可靠性和成本等的关键因素，因此在设计阶段准确高效的分析互连线的时序信息变得更加重要。

时序分析是电路设计的核心，只有符合时序设计要求的电路才能够正确的工作。而时序模型，是保证时序分析准确性的关键因素。低质量的时序模型会导致设计者被迫进行过度设计来应对时序的不准确性，这势必增加芯片面积和制造成本，并且直接影响芯片的管脚布局和封装成本。而高质量的互连线时序分析方法不仅能为电路设计者提供参考，避免发生一些不必要的时序错误，同时也为电路的设计与优化等指明了方向。互连线的时序分析在互连线综合，延迟最小化设计，时序驱动的布局规划、布线等方面都有广泛应用。

深亚微米工艺尺寸的进一步缩小使得一些非线性因素、耦合、信号完整性、电源网络以及工艺、电压、温度等制程工艺的偏差(Process Variation)对电路时序参数的影响不再能被忽略^[2]，传统的不考虑偏差的时序模型已经不能满足设计的要求。而互连线存在的寄生效应，如电阻、电容和电感等，他们所显示的尺寸缩小特性并不与晶体管等有源器件相同，因此，考虑尺寸不断缩小带来的工艺偏差情况的互连线建模和时序分析对于当今的超大规模集成电路来说非常重要^[2]。

1.1 工艺制造偏差问题

随着深亚微米技术进一步下降到纳米量级，设计者在进行时序分析时面临着太多的

不确定性。制造过程中的误差和不精确造成了延迟的不确定性，并可能进一步造成良率(yield)的下降，这就是工艺制造偏差问题。

制造过程中使用的设备，制造工艺有限的精确度及不同的制造步骤都有可能是制造偏差产生的来源^[3]。例如用于平整化绝缘层和金属线表面的化学机械抛光技术(CMP)，由于投影光源波长大于芯片尺寸而造成的光学邻近效应(OPT)等都是制造偏差的来源。这些偏差使得电路器件和互连线等的物理参数产生了偏差，例如沟道长度，栅氧厚度，沟道掺杂浓度，互连线的厚度和高度等。其中，沟道长度及掺杂浓度偏差的影响最为明显。这些物理参数的偏差同样也造成了电路电气参数的偏差，从而最终导致了电路延迟的偏差。

工艺制造偏差根据确定性和随机性的不同，可以分为系统偏差和随机偏差。系统偏差描述的是物理参数偏差中那些被很好建模的，具有确定性变化的部分。大部分的系统偏差都来自于光学邻近效应，化学机械抛光以及金属填充过程。这些偏差可以通过对物理版图的分析得到，可以采用确定性的数值进行建模。随机偏差代表的是偏差中真正具有随机性的部分，因此它们必须被建模为随机变量。线边缘粗糙度(Line-edge Roughness, 即 LER)和掺杂扰动效应(Random-dopant Fluctuations, 即 RDF)等都是随机偏差产生的来源。

根据产生规模的不同，随机偏差进一步又可分为片间偏差和片内偏差。片间偏差对同一个晶圆(die)上的电路产生的影响相同，例如它们可能造成同一晶圆上所有器件的沟道长度均大于设计期望值。这种偏差主要来源于制造过程中不同晶圆，不同批(lot)，不同光罩(rectile)间参数的漂移。片内偏差则会对芯片内的每个器件产生不同的影响，例如它可能造成芯片上某些器件的沟道长度偏小，而另一些器件的沟道长度偏大。这种偏差的产生来源主要是不同光罩(rectile)间的参数漂移。

另外，值得注意的一点是，这些偏差之间的关系并不一定是相互独立的，它们可能存在密切的联系。例如，制造设备的偏差可能造成多个物理参数的偏差之间存在相关性(correlation)。如果多个掩膜使用同一透镜和光源进行曝光，则不同金属层甚至包括多晶硅层上的偏差会存在关联。另外，多个电气参数可能都依赖于同一个物理参数，从而导

致它们之间存在一定的相关性。例如互连线的电阻和电容。它们都受互联线宽度偏差的影响。互连线的宽度增加将导致线间的间距变小，因此线间耦合电容增大，而互连线的电阻变小。此外，栅氧厚度的变化也会导致晶体管驱动电流、阈值电压和栅极电容等的变化。因此，我们在对偏差进行建模时还必须考虑它们之间可能存在的相关性，忽略这种相关性必然会造成一定的误差。

在概率和统计的理论中，相关性是指两个随机变量之间存在的联系。考虑和不考虑相关性因素可能会导致差别很大的结果。如上讨论，从半导体工艺和器件的层面考虑，许多物理量之间会存在某些联系。而从芯片上电路逻辑的层面考虑，不同的结构也会造成物理量之间的关联，例如时钟和数据共享部分时序路径，数据通路上共享部分路径，相似的门类型，门的多个输入端之间以及接近的互连线之间等情况。在此，我们主要讨论工艺制造偏差引起的相关性问题。毋庸置疑，当考虑这些因素后，电路分析的复杂度进一步提升。

1.2 统计时序分析概述

在考虑工艺偏差的基础上，互连线的建模变得更加复杂。传统的处理偏差问题的方法可以分为两类，一类是通过蒙特卡罗方法，另一类则是通过建模不同的边界情况 (corner)，对电路进行多次分析。

给定各偏差变量的概率分布状况，就可以通过蒙特卡罗采样分析方法得到各电路参数的概率密度分布结果。蒙特卡罗方法很直接，通用，并且很准确，然而其缺点是仿真速度慢，计算复杂度随电路参数及规模的增长而呈指数级增长，因而不适于大规模系统的仿真。对于庞大的互连线寄生电路网表结构，其参数数量的巨大使得采用蒙特卡罗方法分析的成本非常高。

第二类方法，如静态时序分析方法，通过对处于不同边界情况 (corner) 下的电路进行分析来建模存在偏差的情况。最差边界 (worst case) 对应最高温度和最低电压下的电路，所有参数处于最差情况下，例如互连线延迟和门延迟等最大；最好边界 (best case) 对应最低温度和最高电压下的电路，所有参数处于最好的状况；典型边界 (typical case)

则对应电路处于通常温度范围和正常电压下的情况。这种基于边界分析的方法可以较好的建模片间偏差对电路的影响，因为它总是假定一个晶圆（die）上的电路同时都处于最坏或最好的情况下。然而片内偏差的存在使得一个晶圆上的电路不大可能同时都处于同样条件下，因此仍采用该方法对于延迟的估计会过于悲观，它并不能很好的建模片内偏差的情况。

工艺尺寸的不断缩小使得片内偏差对电路造成的影响不能被忽略，基于 corner 的确定性时序分析方法，如静态时序分析等，不再能够给出关于电路时序的准确信息，因而统计时序方法被提出，将电路参数的偏差建模为具有一定概率分布的随机变量，从而对片内偏差可以进行很好的建模。

互连线的电气参数如电阻，电容，电感等依赖于版图上的一些物理参数，如互连线的长度，宽度，高度，以及金属层间厚度等。制造偏差造成这些物理参数偏离设计的理想值，因此互连线的物理参数，及其导出的电气参数等，不再是确定的数值，而需要被建模为具有一定概率分布的随机变量，进而通过随机变量间的运算得到具有一定概率密度分布的延迟等信息。这些统计信息即可以被用于接下来的整个芯片的时序分析过程，得到大规模电路的延迟概率分布等信息。由于片间偏差对于同一芯片上电路的影响是相同的，在分析芯片电路时，其影响可用确定数值进行建模。因此，在本文中，我们仅讨论片内偏差对电路造成的影响。

针对互连线的统计建模和时序分析方法有很多^{[4][5][6][7][8][9][10][11]}，然而这些算法没有给出简单有效的闭合形式统计延迟估算模型，互连线延迟的概率分布最终只能通过蒙特卡罗方法采样分析得到，计算量大。这不适用于增量时序分析或者物理设计优化等需要快速准确得到延迟信息的场合，因此，既有效率又容易实现，能够保证准确性的互连线统计时序分析方法成为研究的热点。本文即提出了一种对互连线进行统计时序建模的符号化方法，以便快速准确的得到具有一定概率分布的延迟信息，提供芯片统计时序分析时所需的时序信息。

显然，采用随机方法建模后产生的参数数量将远远超过确定性分析方法中的参数数量。因为工艺偏差有很多来源，即便对时序参数采用最简单的正态分布假设，并采用线

性建模，也需要参数的平均值和其对各个偏差来源的敏感度系数。当对大规模电路进行分析时，将不可避免的引发“数据爆炸”问题。因此，如何在考虑制造偏差的情况下，仍能对电路进行快速准确的分析得到所需带有一定概率分布的信息成为关键问题之一。

1.3 符号化分析方法

基于数值分析开发的仿真器，它可以快速的对电路进行运算仿真，然而我们无法从结果中明确找到决定电路行为和性能的参数。采用蒙特卡罗仿真的互连线统计建模方法，为了得到目标量的概率分布信息，对每个采样点，需要不停的重复整个计算过程，最终得到统计结果。这种方法随着偏差随机变量和电路规模的增大，计算成本会迅速增加。

符号化电路分析方法产生的背景是在分析大规模集成电路的过程中，需要对一些关键器件改变参数，进行反复扫描分析，因此会产生许多重复的计算消耗。这部分消耗由于电路的大规模性，会造成很大的额外成本。为了避免大量的重复分析过程，提高效率，符号化的概念被提出，即将被扫描器件的参数表示为一个抽象的符号变量，通过一次性分析得到系统的符号化模型。随后只需改变符号变量的参数值就可以立即通过符号化模型计算得到相应的结果，无需重复扫描与分析，避免了大量的重复数值计算。

可以发现，器件重复扫描分析的过程与上述基于蒙特卡罗采样的统计建模方法有些许类似之处，都存在大量的重复计算，因此一个自然而然出现的问题是，我们是否可以将符号化方法应用到互连线的统计时序建模中，避免重复计算，从而得到高效的互连线统计时序分析算法呢？

Guoyong Shi等^[12]提出了一种符号化仿真的方法，是对互连线符号化模型降阶方法(Symbolic Model Order Reduction)的初步探索。在文中，作者提出了符号隔离方法和标准投影映射方法(nominal-projection)两种运用符号化概念的方法。符号隔离方法(Symbol Isolation)是将电路中一些符号化的元件隔离出来，仅保留与剩余电路的符号化接口，未被隔离出来的这部分剩余电路则可以采用标准的模型降阶算法进行简化与建模。通常模型降阶算法不会改变网络的端口，因此最后可将符号化电路与经过模型降阶得到的低阶

电路两部分重新结合在一起。另外，文中还提出了用于计算符号化Krylov子空间的映射方法（nominal-projection），使得参数存在微小偏差时，仍可以通过该符号化模型得到结果，无需重新计算。论文最后给出了参数变化情况下采用符号化模型降阶算法仿真的结果。原电路的参数被假设存在 50%的随机扰动，运用符号化模型降阶算法得到的结果与蒙特卡罗仿真结果相比有很好的精确度，误差被控制在 1%以内，从而说明符号化模型降阶方法能够较好地分析参数变化对于电路仿真结果变化的影响。

受该符号化模型降阶方法的启发，在本文中，我们将符号化分析方法应用到了互连线的统计时序分析过程中，提出了一种高效准确的互连线延迟分析方法。

1.4 本章小结

本章节讨论了工艺尺寸不断缩小带来的制造偏差问题，介绍了对偏差进行建模的互连线统计时序分析方法，并在最后探讨了符号化电路分析方法和最近提出的符号化模型降阶概念，为理解后续章节中符号化统计时序分析方法提供了必要的背景知识。

第二章 互连线时序分析算法

在介绍互连线的统计时序分析方法前，我们首先给出确定性互连线时序分析方法的介绍。许多互连线的统计时序分析方法正是由确定性时序分析方法发展而来。

2.1 确定性时序分析方法

互连线的延迟估算有很多方法，基本上可以分为两类。一类是基于模型降阶的算法，通过对互连线的低阶近似模型进行分析，得到延迟信息；另一类是延迟度量算法，其本质上是通过对模型降阶得到一阶或二阶系统进行修正，提出闭合形式的延迟计算方法。下面分别对两类方法进行介绍。

2.1.1 基于模型降阶算法的延迟估算

互连线的建模通常采用模型降阶方法^{[13][14][15][16][17]}。这是因为互连线的寄生电路网表结构是非常庞大的，若采用传统的SPICE仿真器进行电路仿真，数值准确，但效率非常低。寄生网表结构参数数量的巨大导致仿真器内的电路描述矩阵阶数非常高，而高阶的矩阵运算十分耗费时间，于是模型降阶思想被提出。它采用较低阶的电路模型近似原始高阶系统，在损失很小精度的情况下，得到低阶的系统传输函数，在保证准确描述原系统的特性的基础上，使总的运算次数与原来相比大大减少，因此得到了广泛的研究和应用。有了高阶互连线的低阶近似模型，我们就可以通过对其进行分析方便的提取互连线延迟等时序信息。

电路的零冲激瞬态响应定义为：

$$h(t) = y(t) |_{x(t)=\delta(t)} \quad (2.1 - 1)$$

其中， $y(t)$ 是电路的响应， $x(t)$ 是电路的输入激励。

传输函数是其经过拉普拉斯变换得到的 s 域函数，即 $H(s) = L\{h(t)\} = \int_0^{\infty} e^{-st} h(t) dt$ 。

利用低阶模型传输函数的极点—余式表达形式，见式 (2.1 - 2)，我们可以直接对其进行拉普拉斯反变换，方便的求得电路的时域冲激响应，见式(2.1-3)。

$$\begin{aligned} H(s) &= \frac{B(s)}{A(s)} = \frac{b_0 + b_1s + \cdots + b_{n-1}s^{n-1}}{1 + a_1s + \cdots + a_ns^n} \\ &= K \frac{(s - z_1)(s - z_2) \cdots (s - z_{n-1})}{(s - p_1)(s - p_2) \cdots (s - p_n)} = \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \cdots + \frac{k_n}{s - p_n} \end{aligned} \quad (2.1 - 2)$$

$$h(t) = \sum_{i=1}^n k_i e^{p_i t} u(t) \quad (2.1 - 3)$$

由于数字电路中的信号只有逻辑值 0 和 1，因此数字信号的变化均可以用阶跃信号或斜坡信号描述。电路的延迟即定义为当输入信号为上升或下降的阶跃信号时，输出信号跟随其变化至 50% 的位置所需要的时间。

根据电路的瞬态冲激响应，电路阶跃响应的 s 域表达式为：

$$Y(s) = H(s) \cdot \frac{1}{s} \quad (2.1 - 4)$$

利用 $H(s)$ 的极点—余式表达形式，我们可以将 s 域阶跃响应表示为：

$$Y(s) = \sum_{i=1}^n \frac{k_i}{(s - p_i)s} = \sum_{i=1}^n \frac{k_i}{p_i} \left(\frac{1}{s - p_i} - \frac{1}{s} \right) \quad (2.1 - 5)$$

经过拉普拉斯反变换，其时域阶跃响应为：

$$y(t) = \sum_{i=1}^n \frac{k_i}{p_i} (e^{p_i t} - 1) u(t) \quad (2.1 - 6)$$

于是，互连线延迟 τ 可以通过求解 $\sum_{i=1}^n \frac{k_i}{p_i} (e^{p_i \tau} - 1) = 50\%$ 得到。

模型降阶方法使得对互连线时延的计算速度大大提高，已经在许多 EDA 工具中被应用。为了提高其计算精度，人们常常通过计算更高阶的矩来得到多极点的低阶近似模型。然而要得到多极点模型就必须求解非线性方程。这非常费时，并且会在计算高阶矩时产生不稳定极点。

2.1.2 基于矩的延迟度量算法

利用上述方法通过频域建立起来的低阶近似模型做时域的时延估计时必须进行频域到时域的转变，这不可避免的给计算带来额外的时间和误差。而建立频域模型需要进行矩阵求逆或分解和乘法运算，这给模型的获取造成极大的计算困难，对于前期的延迟估计来说，成本较高。另外，模型降阶算法有稳定性和保守性的问题，容易产生不稳定极点。因此，一些简洁高效的延迟度量算法^{[18][19][20][21][22][23]}（delay metrics）被提出。

研究时延特性的互连线模型主要是沿着矩匹配这条主线发展的，因此下面我们首先给出电路矩的定义，随后介绍基于矩的延迟度量算法。

2.1.2.1 电路矩的定义

将电路的传输函数 $H(s)$ 在 $s=0$ 点处进行展开，可以得到级数：

$$H(s) = m_0 + m_1 s + m_2 s^2 + m_3 s^3 + \dots \quad (2.1-7)$$

其中 $m_q = \frac{1}{q!} \frac{d^q H(s)}{ds^q} \Big|_{s=0}$ ，即被称为电路矩。

另一方面，根据 $H(s) = L\{h(t)\} = \int_0^{\infty} e^{-st} h(t) dt$ ，采用 e^{-st} 在 $s=0$ 点展开的级数，可以得到 $H(s)$ 的另一种表达式：

$$H(s) = \int_0^{\infty} h(t) [1 - st + \frac{1}{2} s^2 t^2 - \frac{1}{6} s^3 t^3 + \dots] dt = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} s^k \int_0^{\infty} t^k h(t) dt \quad (2.1-8)$$

则使(2.1-7)式等于(2.1-8)式可以推得：

$$m_q = \frac{(-1)^q}{q!} \int_0^{\infty} t^q h(t) dt \quad (2.1-9)$$

此即为电路矩的计算公式。可见其与概率理论中定义的第 q 阶矩 $\int_0^{\infty} t^q h(t) dt$ 有一个常系数的差别。在后面的讨论中，为了方便，我们有时也将电路矩简称为矩。

2.1.2.2 Elmore 延迟度量

Elmore延迟度量是最早提出的延迟度量算法。在二十世纪四十年代，Elmore提出了

用线性放大器冲激响应的第一阶和第二阶电路矩来评估电路的阶跃响应^[40]，即， $|m_1|$ 是对应阶跃响应的50%延迟的很好评估， $\sqrt{2\pi m_2 - m_1^2}$ 是阶跃响应上升时间的很好评估。

根据50%的延迟定义，若用 τ 表示延迟，则有

$$\int_0^{\tau} h(t)dt = 0.5 \quad (2.1 - 10)$$

若将电路瞬态冲激响应 $h(t)$ 看作概率密度函数，则 τ 相当于其中值(media)。

一阶矩的定义为：

$$|m_1| = -m_1 = \int_0^{\infty} th(t)dt \quad (2.1 - 11)$$

因此，一阶矩相当于随机变量 t 的均值，即 $T_D = E[t] = \int_0^{\infty} th(t)dt$ 。

可见，Elmore延迟的本质即是将非负的电路瞬态冲激响应 $h(t)$ 看作概率密度函数，用均值(mean) T_D 近似中值(media)，即50%延迟点 τ 。

如果冲激响应 $h(t)$ 的曲线是对称的，则Elmore延迟是完全准确的，没有误差。然而一般来说，电路的冲激响应 $h(t)$ 并不是对称的，而是具有正偏差的(positive skewed)，即曲线的右边部分很长，左边部分相对比较短。因此均值点和中值点并不相同。对于RC树状电路来说，已有研究证明，其均值总是大于中值点。因此Elmore延迟是互连线延迟的上界。当冲激响应的不对称性很大时，它会产生比较大的误差。因此通常Elmore延迟对远端节点的近似效果较好，但是对近端点，则近似误差比较大。

Penfield和Rubenstein等^[23]是首先把Elmore延迟应用到RC树状电路的分析中的研究者。它们发现Elmore延迟度量是阶跃响应主要时间常数的很好近似，即将Elmore延迟度量作为主要极点，有 $v(t) \approx 1 - e^{-\frac{t}{m_1}}$ ，其中 $k_1 = 1$ ， $p_1 = \frac{1}{m_1}$ 。50%的延迟点即 $v(\tau) = 0.5$ 。

求解该等式可以得到： $t_d = -\frac{1}{p_1} \ln(2k_1) = -m_1 \ln(2)$ 。显然， t_d 是Elmore延迟度量乘以一个标量 $\ln(2)$ ，我们称它为标量化的Elmore延迟。这种方法同Elmore延迟一样，有时会非

常不准确。

2.1.2.3 高阶延迟度量算法

为了提高准确性，需要对 Elmore 延迟度量进行改进。一个方向即是采用更高阶矩进行计算来获取更高的精度。

Kahang和Muddu等^[18]于 1995 年提出了三种延迟度量算法 (delay metrics)，均采用了前三阶电路矩 m_1 ， m_2 和 m_3 。

第一种度量算法是从三个电路矩出发，使用两阶降阶模型进行近似，计算得到其两个极点和余式，然后去掉次要极点和余式，将主要极点和余式代入到标量化的延迟度量表达式中得到。

两阶降阶模型的两个极点和两个余式的表达式分别为：

$$p_{1,2} = \frac{2}{m_1 \mp \sqrt{4m_2 - 3m_1^2}} \quad (2.1 - 12)$$

$$k_1 = -k_2 = -\frac{1}{\sqrt{4m_2 - 3m_1^2}} \quad (2.1 - 13)$$

将 k_1 和 p_1 代入标量化 Elmore 延迟计算公式 $t_d = -\frac{1}{p_1} \ln(2k_1)$ 中即得到：

$$DM1 = \frac{1}{2} (-m_1 + \sqrt{4m_2 - 3m_1^2}) \ln\left(1 - \frac{m_1}{\sqrt{4m_2 - 3m_1^2}}\right) \quad (2.1 - 14)$$

第二中度量算法是用上述两个极点产生的传输函数来计算，尝试把两个极点合并成为一个新的极点：

$$\frac{1}{p_d} = -\sqrt{\frac{1}{p_1^2} + \frac{1}{p_2^2}} = -\sqrt{2m_2 - m_1^2} \quad (2.1 - 15)$$

利用 $k_1=1$ ，代入标量化 Elmore 延迟公式可以得到：

$$DM2 = \sqrt{2m_2 - m_1^2} \ln(2) \quad (2.1 - 16)$$

最后一种度量算法是利用近似的极点来计算延迟，相当于把一阶矩加到冲激响应的

标准误差上。因为 $p_1 = \lim_{j \rightarrow \infty} \frac{m_j}{m_{j+1}}$ ，所以我们可以近似得到：

$$p_1 = \frac{m_2}{m_3}, \quad p_2 = \frac{m_2}{m_3} \left\| \frac{m_3(m_2 - m_1^2)}{m_1(m_1 m_3 - m_2^2)} \right\| \quad (2.1 - 17)$$

根据这两个极点算得余式为：

$$k_1 = \frac{(1 - m_1 p_2) p_1}{(p_1 - p_2)}, \quad k_2 = \frac{(1 - m_1 p_1) p_2}{(p_1 - p_2)} \quad (2.1 - 18)$$

将 k_1 和 p_1 代入标量化 Elmore 延迟公式即得到延迟度量 DM_3 。实际的延迟度量是通过对 DM_3 运行单步 Newton-Raphson 迭代得到的，因此改进的显式近似为：

$$DM_4 = DM_3 + \frac{0.5 - k_1 e^{p_1 DM_3} - k_2 e^{p_2 DM_3}}{k_1 p_1 e^{p_1 DM_3} + k_2 p_2 e^{p_2 DM_3}} \quad (2.1 - 19)$$

B. Tutuianu 等^[19]于 1996 年提出了利用一至三阶矩计算延迟的方法，仍然使用两极点模型进行近似，但假设其满足两个条件，即存在一个主极点，即 $p_1 < p_2$ ，且主要极点的余式总是正的。这些假设可以通过分析得证。因此我们可以假设时域函数中主极点对应的指数项是主要的，推导得到其极点及余式的表达式为：

$$p_1 = -\frac{m_2}{m_3}, \quad p_2 = p_1 \left| \frac{\left(\frac{1}{m_1} - \frac{m_1}{m_2} \right)}{\left(\frac{m_1}{m_2} - \frac{m_2}{m_3} \right)} \right| \quad (2.1 - 20)$$

$$k_1 = \frac{1 + m_1 p_2}{p_1 - p_2} p_1^2, \quad k_2 = -\frac{1 + m_1 p_1}{p_1 - p_2} p_2^2 \quad (2.1 - 21)$$

基于此提出了可以用于迭代求解延迟的初始值：

$$t_1 = \frac{1}{p_1} \ln\left(\frac{2k_1}{p_1}\right) \quad (2.1 - 22)$$

该初始值是一个很好的猜测，然后经过单步的 Newton-Raphson 方法迭代即可得到一个近似误差很小的解：

$$t_{NR} = t_1 + \frac{-0.5 + \frac{k_1}{p_1} e^{-p_1 t_1} + \frac{k_2}{p_2} e^{-p_2 t_1}}{k_1 e^{-p_1 t_1} + k_2 e^{-p_2 t_1}} \quad (2.1 - 23)$$

C. Alpert等^[20]提出了一种简单的延迟度量经验算法—D2M。他们发现标量化的Elmore延迟在近端节点高估了延迟，而在远端节点低估了延迟。通过经验，他们发现比率 $r = -\frac{m_1}{\sqrt{m_2}}$ 在近端小于1，而在远端则大于1，对于单RC电路，它等于1。因此他们用

比率 r 乘以标量化的Elmore延迟度量来计算得到新的D2M延迟度量：

$$D2M = -r m_1 \ln(2) = -\frac{m_1^2}{\sqrt{m_2}} \ln(2) \quad (2.1 - 24)$$

这种算法也可以看作是极点为 $-\frac{m_1^2}{\sqrt{m_2}}$ 的模型近似，是利用二阶矩进行修正的单极点模型。后来，苏舟等^[23]根据此种思想，又提出了一种新的互连线延迟度量：

$$t_d = -\frac{2m_1^2 - m_2}{m_1} \ln(2) \quad (2.1 - 25)$$

Y. I. Ismail等^[24]于2004年提出了利用高阶矩得到的通用的更为精确的延迟度量方法。作者采用一个通用的形式来估算波形变化至50%处的延迟，如下：

$$t_{0.5_q} = a_1 \cdot m_1 + a_2 \cdot \frac{m_2}{m_1} + a_3 \cdot \frac{m_3}{m_1^2} + \cdots + a_q \cdot \frac{m_q}{m_1^{(q-1)}} \quad (2.1 - 26)$$

其中， q 表示所用于近似的矩的最高阶数，0.5表示信号变化至50%所需的时间， $t_{0.5_q}$ 即表示用 q 阶矩近似的我们通常所说的50%的延迟。 a_1, \dots, a_q 表示对应项前的系数，经过对波形的分段近似得到。作者通过对各种互连线结构的仿真与分析，给出了使用分别使用二至七阶矩近似得到的延迟公式，如下：

$$t_{0.5_2} = -0.898 m_1 + 0.543 \frac{m_2}{m_1} \quad (2.1 - 27)$$

$$t_{0.5_3} = -1.0746m_1 + 0.2928\frac{m_2}{m_1} + 0.0911\frac{m_3}{m_1^2} \quad (2.1 - 28)$$

$$t_{0.5_4} = -2.13m_1 + 2.70\frac{m_2}{m_1} - 1.04\frac{m_3}{m_1^2} + 0.112\frac{m_4}{m_1^3} \quad (2.1 - 29)$$

$$t_{0.5_5} = -3.05m_1 + 5.59\frac{m_2}{m_1} - 4.36\frac{m_3}{m_1^2} + 1.75\frac{m_4}{m_1^3} - 0.291\frac{m_5}{m_1^4} \quad (2.1 - 30)$$

$$t_{0.5_6} = -3.56m_1 + 13.2\frac{m_2}{m_1} - 23.9\frac{m_3}{m_1^2} + 21.8\frac{m_4}{m_1^3} - 9.44\frac{m_5}{m_1^4} + 1.53\frac{m_6}{m_1^5} \quad (2.1 - 31)$$

$$t_{0.5_7} = -1.23m_1 - 8.32\frac{m_2}{m_1} + 38.8\frac{m_3}{m_1^2} - 63.8\frac{m_4}{m_1^3} + 51.1\frac{m_5}{m_1^4} - 19.9\frac{m_6}{m_1^5} + 3.00\frac{m_7}{m_1^6} \quad (2.1 - 32)$$

总的来说，所使用的矩的最大阶数越大，近似程度越好。

由上述讨论可见，延迟度量算法本质上仍然是通过模型降阶算法用低阶模型近似原系统，但通常只采用一阶和二阶系统近似，然后对其进行修正和改进，得到简单的闭合形式延迟度量算法。

2.1.3 延迟算法总结

综上所述，评估互连线延迟存在许多类不同的算法，不同类方法各有利弊。例如，基于模型降阶（如AWE算法^[13]）的延迟估计方法与延迟度量算法相比成本较高，特别是在设计前期的时序分析中。Elmore度量总是高估了延迟，并且在近端结点误差很大，DM1总是优于Elmore的结果，但是从理论分析， $4m_2 - 3m_1^2 > 0$ 不会总是满足，因此DM1不是一个稳健的方法。DM2没有比Elmore更好，在近端结点大大高估延迟，而在远端节点则大大低估延迟，不是一个好度量。DM3在近端节点误差很大，但在远端节点误差比较小，是计算远端节点延迟的一种好方法。D2M准确度最高，对近源点和远端点的近似均较好。

2.2 统计时序分析方法

制造偏差的存在带来的时序分析的不确定性问题使得在设计阶段准确建模和分析互连线的影响变得更加重要。统计时序分析算法通过对制造偏差进行恰当的建模，更好的解决了延迟的不确定性问题，避免了过度的保守估计，从而提高了设计的活力和性能。

考虑制造偏差后，互连线延迟，跃迁时间等时序信息必须建模为具有一定分布的随机变量，随之而来的问题便是，如何对互连线进行统计建模以得到互连线延迟和信号跃迁时间等参数的概率分布信息？

研究互连线统计时序的思路与研究确定性的互连线延迟方法相似，一些学者基于考虑偏差的参数化模型降阶算法进行分析，而另外一些学者则基于延迟度量算法进行研究。

2.2.1 基于参数化模型降阶的统计延迟估算

假设电路中存在 n 个制造偏差来源 $p_i, i = 1, \dots, n$, 参数化模型降阶方法即是以偏差 p_i 作为变量，将电路参数建模为偏差的一阶模型，即线性函数，通过改进确定性模型降阶的过程使其适用于多参数情况，从而得到以 s 和偏差 p_i 做为联合变量的低阶近似模型。

通过对其进行分析，就可得到以偏差作为参数的电路时域及频域的诸多信息，例如参数化的零极点，参数化的延迟和跃迁时间等。随后根据各偏差变量的概率分布状况，就可以通过蒙特卡罗采样分析方法得到延迟等各种电路性能参数的概率密度分布。

蒙特卡罗方法很直接，通用，并且很准确，然而仿真速度慢，不适于大规模互连线的仿真。因此，J.D.Ma等^[9]后来又提出了区间运算的方法，采用有限区间建模偏差的概率密度分布，经过区间上的代数运算得到延迟概率密度的有限分布区间，并考虑了偏差随机变量间的相关性。

区间运算方法（Interval Arithmetic）是数学计算中的一个古老的分支。它的关键思想是采用一段有限的区间代表变量可能的取值范围，研究能够代替数值代数运算操作的区间代数运算操作和一些基本的区间非线性运算，例如 $\exp()$ 和 $\log()$ 。为了降低统计时序

分析的复杂度，研究者将该方法应用到电路的统计建模中。它将随机变量表示为一个分布在有限区间上的变量，即该区间能够包括随机变量概率密度函数的大部分，从而使得随机变量间的操作转变为区间的计算，而最终计算得到的时序参数的区间就描述了它的概率密度分布区间。然而，经典区间运算方法的缺点是，由于没有用于保留区间变量间相关性的机制，区间方法的误差非常大，经过一系列复杂计算后得到的区间估计保守到几乎无用的地步。因此de Figueiredo和Stolfi^[39]在1997年提出了解决相关性问题的Affine区间形式（Affine Interval Form）。Affine区间由一个均值加上一组建模围绕均值的扰动的不确定性参数，即

$$\hat{x} = x_0 + \sum_{i=1}^N x_i \varepsilon_i \quad (2.2-1)$$

其中， ε_i 的取值区间为[-1,1]，被称为不确定项(Uncertainty Term)。

两个Affine区间的加法操作比较简单，假设 $\hat{x} = x_0 + x_1 \varepsilon_1 + x_2 \varepsilon_2$ ， $\hat{y} = y_0 + y_1 \varepsilon_1 + y_3 \varepsilon_3$ ，则两者相加可得到：

$$\hat{z} = \hat{x} + \hat{y} = (x_0 + y_0) + (x_1 + y_1) \varepsilon_1 + x_2 \varepsilon_2 + y_3 \varepsilon_3 \quad (2.2-2)$$

显然， \hat{z} 仍然是Affine形式，加减法等线性运算符能够很好的保留Affine区间形式。然而，非线性运算会带来一些问题，以乘法为例。

$$\begin{aligned} \hat{z} = \hat{x}\hat{y} &= (x_0 + \sum_{i=1}^n x_i \varepsilon_i)(y_0 + \sum_{i=1}^n y_i \varepsilon_i) \\ &= x_0 y_0 + \sum_{i=1}^n (y_0 x_i + x_0 y_i) \varepsilon_i + (\sum_{i=1}^n x_i \varepsilon_i)(\sum_{i=1}^n y_i \varepsilon_i) \end{aligned} \quad (2.2-3)$$

可见， \hat{z} 不再是Affine形式，因此我们需找到其Affine形式一种近似，例如：

$$\hat{z} \approx x_0 y_0 + \sum_{i=1}^n (y_0 x_i + x_0 y_i) \varepsilon_i + [R(\sum_{i=1}^n x_i \varepsilon_i)][R(\sum_{i=1}^n y_i \varepsilon_i)] \zeta = z_0 + \sum_{i=1}^n z_i \varepsilon_i + R(\hat{x}\hat{y}) \zeta$$

其中， $\zeta \in [-1,1]$ ，是一个新产生的不确定性变量， R 是半径运算符，即代表

$$R(\sum_{i=1}^n x_i \varepsilon_i) = \sum_{i=1}^n |x_i| \varepsilon_i$$

James.D.Ma 等基于上述理论，将经过改进的区间运算方法应用到参数化延迟的求解算法上，在保证计算准确度的基础上提高了运算效率。

首先，他们将考虑偏差因素的电阻，电容，电感等进行线性建模如下：

$$\begin{aligned} R_i &= R_{i,0} + \sum_{j=1}^l \Delta R_{i,j} \varepsilon_j + \sum_{j=1+1}^m \Delta R_{i,j} \varepsilon_j + \sum_{j=m+1}^n \Delta R_{i,j} \varepsilon_j \\ C_i &= C_{i,0} + \sum_{j=1}^l \Delta C_{i,j} \varepsilon_j + \sum_{j=1+1}^m \Delta C_{i,j} \varepsilon_j - \sum_{j=m+1}^n \Delta C_{i,j} \varepsilon_j \\ L_i &= L_{i,0} + \sum_{j=1}^l \Delta L_{i,j} \varepsilon_j - \sum_{j=1+1}^m \Delta L_{i,j} \varepsilon_j - \sum_{j=m+1}^n \Delta L_{i,j} \varepsilon_j \end{aligned} \quad (2.2 - 4)$$

这里 $R_{i,0}, C_{i,0}, L_{i,0}$ 代表电阻，电容，电感的标准值/均值， $\varepsilon_i (1 \leq i \leq n)$ 代表彼此独立的偏差变量， $\varepsilon_i \in [-1, 1]$ ， $\Delta R_{i,j}, \Delta C_{i,j}, \Delta L_{i,j}$ 代表电阻、电容、电感关于偏差 ε_j 的敏感系数。

基于这些 R, L, C 的多参数模型，计算得到 R, L, C 的取值区间，进一步利用 AWE 算法^[13]首先通过 MNA 算法得到电路的状态方程 G, C, B, L ，他们都是以区间变量为基本元素的矩阵。随后使用基于区间运算的 LU 分解求得电路矩，建立迭代求解矩阵，从而得到以区间为变量的 p 阶高次方程，最终得到降阶系统的零极点区间，进而求得延迟的区间。

由于该算法中存在大量的非线性乘法运算，区间运算每次产生的新变量会造成变量数目的爆炸性增长，因而 J.D.Ma 等将其进行改进，对乘法采用如下近似方法，避免了新变量的产生，从而大大降低了计算复杂度。

$$\hat{z} \approx z_0 + \sum_{i=1}^n z_i \left[1 + \frac{z_i}{\sum_{i=1}^n |z_i|} R(\hat{x}\hat{y}) \right] \varepsilon_i \quad (2.2 - 5)$$

最近，J. Zeng 等^[10]基于 AWE 算法，提出了采用二阶非线性模型对变量进行建模得到

互连线延迟统计信息的方法。作者首先将电导和电容矩阵等进行展开，近似表示物理参数如互连线宽度和高度的二阶模型。设 g 是电导矩阵 G 的元素，则有

$$g = \tilde{g} + g' \delta + \delta^T g'' \delta \quad (2.2 - 6)$$

其中， \tilde{g} 是均值； $g' \in R^{1 \times n}$ 是一阶项前的系数， $g'(k) = \frac{\partial g}{\partial \delta_k}$, $k = 1, \dots, n$ ； $g'' \in R^{n \times n}$ 是

二阶项前的系数矩阵， $g''(k, l) = \frac{1}{2!} \frac{\partial^2 \tilde{g}}{\partial \delta_k \partial \delta_l}$, $k, l = 1, \dots, n$ 。

电容矩阵 C 的二阶模型可以通过同样方法得到。

接下来，同样采用 AWE 算法，我们可以相继得到电路矩，极点和延迟的二阶模型：

$$\begin{aligned} m &= \tilde{m} + m' \delta + \delta^T m'' \delta \\ p &= \tilde{p} + p' \delta + \delta^T p'' \delta \\ t_d &= \tilde{t}_d + t_d' \delta + \delta^T t_d'' \delta \end{aligned} \quad (2.2 - 7)$$

采用二阶模型的好处是，作者可以直接通过延迟的特征函数和傅立叶变换计算得到延迟的概率密度分布（这一方法将在下章进行详细介绍），从而避免了繁琐而又耗时的蒙特卡罗采样方法。然而，该方法由于采用 AWE 算法计算得到延迟，具有 AWE 算法的一些固有缺点，例如稳定性问题。这是 AWE 方法本身原理存在的不稳定性及计算过程中的计算误差造成的。

另外，该方法对电阻、电容、矩、极点等每一步得到的变量都采用二阶模型近似，因此需要求解每个变量二阶模型每一项前的系数，这样造成的计算量非常大。在后面的实验中，我们将给出实验结果说明该方法在保证计算精度方面的缺点。

2.2.2 基于延迟度量的统计延迟估算

基于参数化模型降阶算法的统计时序分析方法虽然可以得到比较好的精度，然而其需要的计算量非常大。这些计算量部分来自于模型降阶算法本身不小的计算量，部分来自于大量偏差随机变量带来的计算复杂度。且用于获得统计信息的蒙特卡罗方法需要多

次采样重复计算，因而这种方法并不能被应用于时序综合和优化等需要快速获得时序信息的场合。而基于延迟度量的延迟估算方法由于其简洁的闭合形式表达式，提供了设计高效统计时序分析方法的可能性。

K. Agarwal等^[7]在假设互连线物理参数的偏差和延迟均为高斯分布的前提下，给出了互连线延迟关于物理参数偏差随机变量的统计线性模型。

这一线性模型是通过三步近似得到的。

第一步，首先将电阻，电容等电气参数表示为物理参数的线性模型，见下式。

$$\begin{aligned} R &= R_{nom} + a_1 \Delta W + a_2 \Delta T \\ C &= C_{nom} + b_1 \Delta W + b_2 \Delta T + b_3 \Delta H \end{aligned} \quad (2.2 - 8)$$

其中， R_{nom} 和 C_{nom} 代表无偏差时的标准电阻和电容值， $\Delta W, \Delta T, \Delta H$ 表示互连线宽度、厚度及 ILD 层厚度各自存在的偏差，系数 a_i 和 b_i 代表各参数相对偏差的敏感系数。

第二步，运用路径追溯算法（path-tracing）手动推导，将电路矩表示为物理参数的线性模型，省略出现的二阶项，见下式。

$$\begin{aligned} m_1 &= m_{1(nom)} + k_W \Delta W + k_T \Delta T + k_H \Delta H \\ m_2 &= m_{2(nom)} + A_W \Delta W + A_T \Delta T + A_H \Delta H \end{aligned} \quad (2.2 - 9)$$

其中， $m_{1(nom)} = m_1(R_{i(nom)}, C_{i(nom)})$ ， $m_{2(nom)} = m_2(R_{i(nom)}, C_{i(nom)}, m_{1(nom)})$ 表示不存在偏差时的标准一阶和二阶电路矩，其各项系数经手动推导可得：

$$\begin{aligned} k_W &= m_1(R_{i(W)}, C_{i(nom)}) + m_1(R_{i(nom)}, C_{i(W)}) \\ k_T &= m_1(R_{i(T)}, C_{i(nom)}) + m_1(R_{i(nom)}, C_{i(T)}) \\ k_H &= m_1(R_{i(nom)}, C_{i(H)}) \\ A_W &= m_2(R_{i(nom)}, C_{i(nom)}, k_W) + m_2(R_{i(nom)}, C_{i(W)}, m_{1(nom)}) + m_2(R_{i(W)}, C_{i(nom)}, m_{1(nom)}) \\ A_T &= m_2(R_{i(nom)}, C_{i(nom)}, k_T) + m_2(R_{i(nom)}, C_{i(T)}, m_{1(nom)}) + m_2(R_{i(T)}, C_{i(nom)}, m_{1(nom)}) \end{aligned}$$

$$A_H = m_2(R_{i(nom)}, C_{i(nom)}, k_H) + m_2(R_{i(nom)}, C_{i(H)}, m_{1(nom)}) \quad (2.2 - 10)$$

第三步，通过采用延迟度量 D2M 算法将延迟近似表示为物理参数的线性模型，见下式。

$$\begin{aligned} T_D &= \ln 2 * \frac{(m_{1(nom)} + k_W \Delta W + k_T \Delta T + k_H \Delta H)^2}{\sqrt{m_{2(nom)} + A_W \Delta W + A_T \Delta T + A_H \Delta H}} \\ &\approx \ln 2 * \frac{(m_{1(nom)})^2}{\sqrt{m_{2(nom)}}} (1 + S_W \Delta W + S_T \Delta T + S_H \Delta H) \end{aligned} \quad (2.2 - 11)$$

其中，各项前的系数为：

$$\begin{aligned} S_W &= \frac{2k_W}{m_{1(nom)}} - \frac{A_W}{2m_{2(nom)}} \\ S_T &= \frac{2k_T}{m_{1(nom)}} - \frac{A_T}{2m_{2(nom)}} \\ S_H &= \frac{2k_H}{m_{1(nom)}} - \frac{A_H}{2m_{2(nom)}} \end{aligned} \quad (2.2 - 12)$$

根据式(2.2-11)进一步可推得其均值和标准差为：

$$E(T_D) = \ln 2 * \frac{(m_{1(nom)})^2}{\sqrt{m_{2(nom)}}} \quad (2.2 - 13)$$

$$stdev(T_D) = \ln 2 * \frac{(m_{1(nom)})^2}{\sqrt{m_{2(nom)}}} \sqrt{(S_W^2 \sigma_W^2 + S_T^2 \sigma_T^2 + S_H^2 \sigma_H^2)} \quad (2.2 - 14)$$

由于多个高斯随机变量经过加减法运算后得到的随机变量仍为高斯分布，而描述高斯分布仅需要均值和标准差两个参数，因此计算上述两参数即可得到描述延迟的概率密度分布情况。

作者将其与蒙特卡罗采样得到的标准延迟概率分布进行比较，得到了平均为 1% 的均值误差和 4% 的标准差误差结果。然而这些数据仅仅来自于对节点数小于 30 的小规模电路进行测试的结果。此方法对大规模互连线结构的效果并不明确。虽然小规模电路的结果表明，线性模型近似造成的误差并不是非常大，然而在大规模的电路中，忽略不断

累积的非线性误差可能会造成非常大的影响。

另外，该方法假设偏差随机变量之间，如互连线宽度，金属层厚度和ILD厚度之间，是互相独立的。然而，根据我们前面对制造偏差的讨论，我们知道，这些偏差变量之间有可能存在非常复杂的关系，因此忽略他们之间的相关性有可能造成非常大的影响，从而导致算法精度不能满足要求。在第四章的实验结果部分，我们将给出采用此方法对大型互连线结构进行仿真的结果。

2.3 本章小结

本章首先讨论了不考虑制造偏差的传统的确定性时序分析方法，给出了基于模型降阶算法与基于电路矩的两类不同延迟估算方法的介绍和比较；然后给出了在考虑制造偏差问题后，对其进行建模的互连线统计时序分析方法，着重介绍了两种已有的分别基于参数化模型降阶方法和延迟度量方法的互连线统计时序分析方法。

第三章 符号化互连线统计时序分析方法

统计时序分析通过对制程偏差进行恰当的建模,更好的解决了延迟的不确定性问题,避免了过度的保守估计,从而提高了设计的活力和性能。在本章中,我们将介绍一种新的基于符号化方法的互连线统计时序分析方法,可以快速准确的得到互连线的统计时序信息。

该算法利用了创新性的符号化矩算法,并将其进行扩展,利用电阻和电容等电气参数的统计模型将矩建模为互连线宽度和高度等物理版图参数偏差的闭合形式模型;进一步通过基于矩的闭合形式延迟度量算法,获得了延迟关于互连线物理参数偏差的闭合形式模型。该闭合模型经由上述一次性的符号化方法分析流程即可得到。闭合形式模型的存在使得我们接下来对电路的分析过程更加简单快速,例如,只需给定各物理参数的数值,利用该闭合模型即可快速准确的得到延迟的数值,适用于需要快速得到时序信息的一些电路设计阶段,如物理设计优化等。当将该模型应用到互连线的统计时序分析中时,为了保留互连线物理参数到延迟间的复杂的非线性映射关系,同时也为了避免低效的蒙特卡罗采样分析方法,我们进一步提出了采用二阶模型对延迟的闭合形式模型进行近似,利用概率理论中的特征函数概念,通过傅立叶变换方法,直接得到延迟的概率密度分布结果。二阶模型中的各个系数,包括延迟关于各个互连线物理参数的一阶导数与二阶导数等,由于利用了延迟的闭合形式模型,可以快速准确的得到。同时,利用特征函数与傅里叶变换的方法可以快速的得到延迟的概率密度分布,因此,本文所提出的整个创新性的算法流程具有非常好的运算效率,过程简洁,近似步骤少,从而保证了准确性。

下面,我们将详细介绍该算法流程中的各个阶段,并在本章节的最后给出算法流程的总结与优势探讨。在本文中,对互连线的建模,我们采用了分布式的 RC 模型。互连线的分布式模型有 RC 和 RLC 两种,在频率较低的情况下,电感的影响可以忽略,一般采用 RC 模型,有研究认为当 $L < 6\text{nH/cm}$ 时就可以忽略 L 的影响。

一个简单的互连线 RC 分布式模型见下图,其互连线分段数为 4,输入端电压为 $V_{in}(t)$,

输出端电压为 $V_{out}(t)$ 。每段电阻为 R_i ，电容为 C_i 。在 RC 模型中，每一个节点与地之间都存在电容，非地节点之间没有电容，并且没有电阻连到 GND 上。

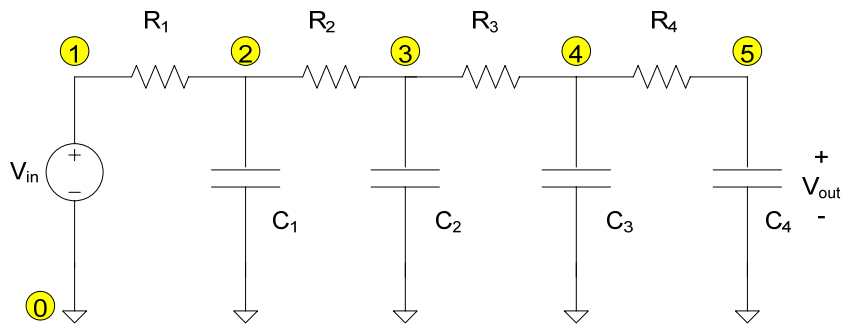


图3-1 简单 RC 分布式模型

Fig.3-1 A simple RC distributed model

3.1 互连线电气参数的统计建模

互连线 RC 模型中存在的寄生参数，如电阻 R ，电容 C 等统称为互连线的电气参数。想要得到互连线延迟的信息，首先关键的即是要提取精确的寄生参数。

寄生参数的准确建模是一个复杂的问题，它们存在许多非线性特征，例如在 130nm 的铜工艺下，金属线的面电阻值(sheet resistance)并不是一个常数，它是互连线宽度的非线性函数。这主要是由于互连线导体边缘处的电子散落造成的。不过，与寄生电容的提取相比，寄生电阻的提取和建模简单很多。寄生电容有许多来源，例如互连线与衬底之间的面电容和边缘电容，不同层互连线间的边缘电容，面电容以及同层互连线之间的耦合电容等。对这些寄生电阻，电容的建模是否准确将会影响到互连线延迟分析的准确性。

文献中存在许多对互连线电路、电容和电感等寄生参数的提取方法^{[30][31]}，从简单的闭合形式模型到复杂的提取技术等都有。它们通过对版图上互连线物理参数的提取得到，例如互连线的长度，宽度，高度，以及金属层间厚度等。

基于统计分析算法效率和准确性的考虑，我们希望有简单准确的闭合形式寄生参数模型，因此采用 J.Cong 等^[31]提出的分段式互连线电阻、电容模型：

$$R = \rho \cdot \frac{L}{HW} \quad (3.1 - 1)$$

$$C = L \left(\frac{2\pi\epsilon_0\epsilon_r}{\log\left(\frac{t_{ox}}{H}\right)} + \frac{(W - \frac{H}{2})\epsilon_0\epsilon_r}{t_{ox}} \right) \quad (3.1 - 2)$$

其中， ρ 代表电阻率， ϵ_0 为真空介电常数， ϵ_r 为材料相对介电常数，该段互连线的长度为 L ，宽度为 W ，高度为 H ，绝缘层的厚度为 t_{ox} ，如图 3-2 所示。读者当然也可以采用其他寄生参数的解析模型。

制造偏差会造成这些物理参数偏离设计的理想值，因此互连线的物理参数，及其导出的电气参数等，不再是确定的数值，而需要被建模为具有一定概率分布的随机变量。我们将考虑制造偏差后的新问题采用数学语言描述如下。

假设 S 代表制造生产的芯片的样本空间，则 $z \in S$ 代表制造出来的一个芯片样本。令 $\delta(z) = \{\delta_1(z), \dots, \delta_p(z)\}$ 代表 z 上的 p 个偏差来源，则 $\delta(z)$ 中的偏差随机变量具有一些共同的性质，如下：

1) $\delta(z)$ 中偏差随机变量的期望均为 0。若某偏差随机变量的期望值不为 0，则我们可以通过将其从原变量减去而获得期望值为 0 的新随机变量。

2) $\delta(z)$ 中的偏差随机变量符合正态分布，即 $\delta = [\delta_1, \delta_2, \dots, \delta_p]^T \sim N(0, \Sigma)$ 。其中 $\Sigma = E\{\delta\delta^T\}$ 是 $p \times p$ 维的相关矩阵，即对角线上是偏差随机变量本身的方差，其他元素为不同偏差随机变量间的相关性。该矩阵通常来说不是单位阵，这是因为不同偏差来源之间可能存在相关性。这里我们为了阐述的方便性，暂时假设物理参数均为正态分布，而实际上他们可以具有任何形式的概率密度函数（PDF）。

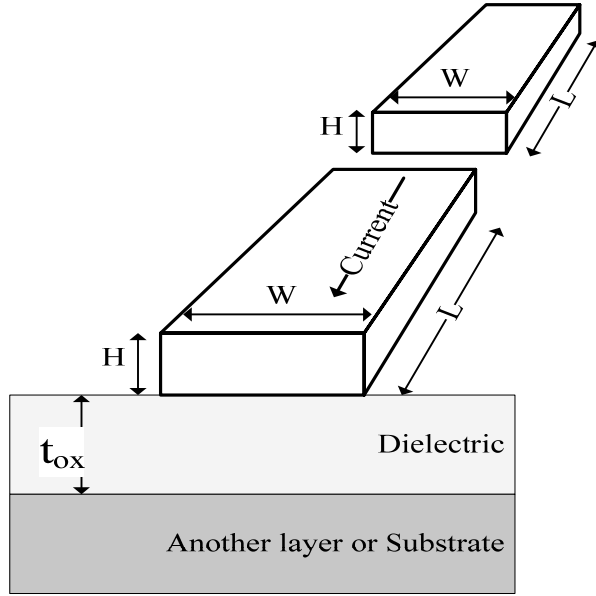


图3-2 互连线物理参数示意图

Fig.3-2 Physical parameters of interconnect

根据上述讨论，若只考虑宽度和高度偏差的影响，即取宽度 W 和长度 H 两个偏差来源，其他参数均为确定性数值。则将这些物理参数偏差建模为随机变量后得到的每段互连线电阻、电容的统计建模模型如下：

$$R = \rho \frac{L}{(\tilde{H} + \delta_H)(\tilde{W} + \delta_W)} \quad (3.1 - 3)$$

$$C = L \left(\frac{2\pi\epsilon_0\epsilon_r}{\log\left(\frac{t_{ox}}{\tilde{H} + \delta_H}\right)} + \frac{(\tilde{W} + \delta_W - \frac{\tilde{H} + \delta_H}{2})\epsilon_0\epsilon_r}{t_{ox}} \right) \quad (3.1 - 4)$$

上式中， \tilde{W}, \tilde{H} 为理想条件下宽度和高度的均值/标准值。 $\delta(z) = [\delta_w, \delta_H]^T \sim N(0, \Sigma)$ 是二维偏差随机变量， Σ 为二维相关性矩阵。

如上所述，在该统计时序算法中，我们采用的是原始精确的寄生参数非线性模型，完整的保留了电阻、电容等与互连线版图物理参数间复杂的非线性映射关系，没有对其进行任何一阶、二阶或高阶的近似。同时，我们没有对偏差随机变量施加任何独立性的

假设，有别于以往任何的互连线统计时序分析方法。没有近似及独立性假设等限制条件的存在，我们的算法将给出更准确的结果，能够处理更多的问题。这一点在该章节接下来的部分及下一章中的实验结果与分析中将给出更多的说明。

3.2 符号化矩算法

由于 RC 树状模型本身结构的特点，基于矩的互连线延迟分析方法有很好的效率。因此为了得到互连线的延迟信息，我们首先需要计算电路的各阶矩。电路矩的计算并不是简单的事，阶数越高，电路规模越大，则需要的计算量越大。若根据电路矩的定义式进行计算，见式(2.1-9)，则我们首先要求得电路的冲激响应 $h(t)$ ，然后在 $[0, \infty)$ 的区间上对 $t^q h(t)$ 进行积分运算。在实际中，这显然是不可行的。值得庆幸的是，根据矩概念的意义，即传输函数各不同指数项 s^n 前的系数，电路矩可以直接对应到实际电路中存在的物理量，也就是说矩所代表的概念是存在相应的物理意义的。基于此，许多计算矩的方法被提出。

L.Pillage等^[14]在1994年提出的快速互连线评估算法 (Rapid Interconnect Circuit Evaluation, 简称为RICE)中使用了一种被称为路径追溯算法(path-tracing)的矩计算方法。该算法专门针对R(L)C树状电路，通过两次遍历即可计算得到任意节点的任意阶矩，复杂度仅是电路节点数的线性函数，即 $O(n)$ ，其中 n 代表电路节点数。我们提出的符号化矩算法这是基于这一算法的计算原理，因此下面我们首先给出路径追溯算法的介绍。

3.2.1 路径追溯算法

首先，我们给出树状电路的定义，然后以一个简单的 RC 树状电路为例，介绍计算各阶电路矩的基本过程，最后给出路径追溯算法的阐述。

如果一个电路的生成树只包括电压源、电感和电阻，不包含电容及电流源，那么该电路即称之为树状电路(tree-like)。图 3-3中给出了一个简单的R(L)C电路及其对应的生成树的例子^[14]。

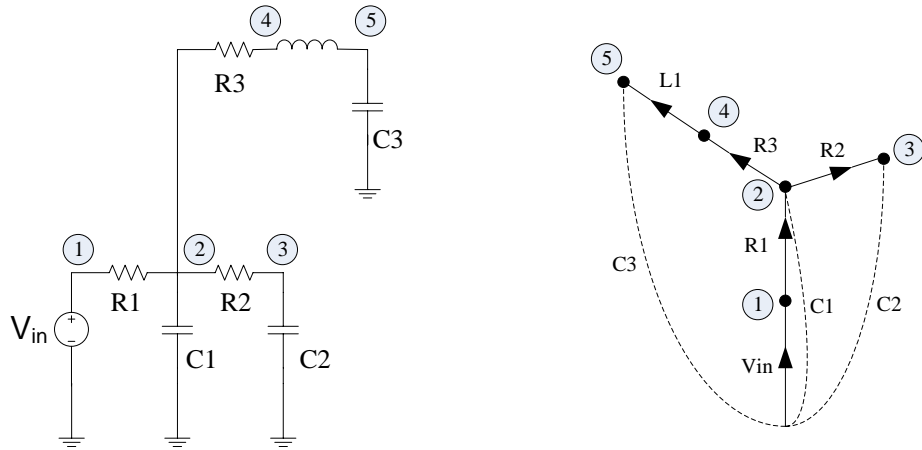


图3-3 简单 RLC 树状电路

Fig.3-3 A simple RLC tree-like circuit

为了阐述计算电路矩的基本原理，下面我们以图 3-4 中所示的简单互连线 RC 模型为例进行推导分析。图中输入电压源为 V_{in} ，各支路上的电阻值表示为 $R_i, i=1,2,3,4$ ，电容值为 $C_i, i=1,2,3,4$ ，使用带圈的数字表示节点的编号，每个节点上的电压，即每个电容上对应的电压表示为 $V_i^{C_i}, i=1,2,3,4$ 。

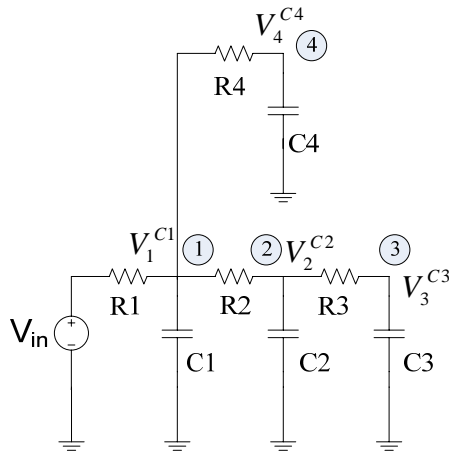


图3-4 RC 树状电路示例

Fig.3-4 A simple RC tree-like circuit

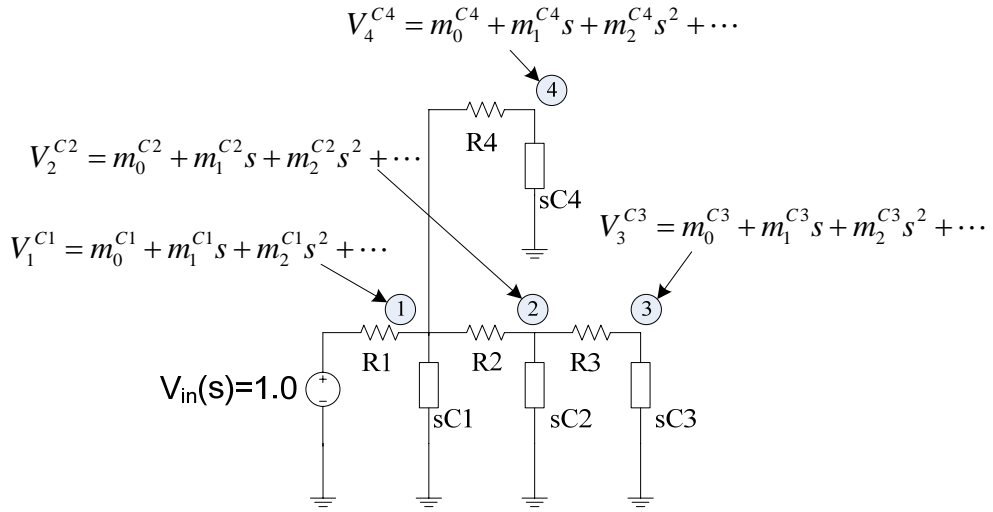


图3-5 s域等价电路

Fig.3-5 Equivalent circuit in s domain

各节点电压经拉普拉斯变换后得到的频域冲激响应可以用图 3-5中的电路求解。其中，电容被s域的导纳代替。 m_j^C 代表不同节点i上，阶数为j的电路矩。节点，或者说电容上的电压可以展开成为s的级数，如图中所示：

$$\begin{aligned}
 V_1^{C1} &= m_0^{C1} + m_1^{C1}s + m_2^{C1}s^2 + \dots \\
 V_2^{C2} &= m_0^{C2} + m_1^{C2}s + m_2^{C2}s^2 + \dots \\
 V_3^{C3} &= m_0^{C3} + m_1^{C3}s + m_2^{C3}s^2 + \dots \\
 V_4^{C4} &= m_0^{C4} + m_1^{C4}s + m_2^{C4}s^2 + \dots
 \end{aligned} \tag{3.2-1}$$

进一步，利用该结果及各电容对应的 s 域导纳便可以求得各电容上的电流：

$$\begin{aligned}
 I^{C1} &= sC_1(m_0^{C1} + m_1^{C1}s + m_2^{C1}s^2 + \dots) \\
 I^{C2} &= sC_2(m_0^{C2} + m_1^{C2}s + m_2^{C2}s^2 + \dots) \\
 I^{C3} &= sC_3(m_0^{C3} + m_1^{C3}s + m_2^{C3}s^2 + \dots) \\
 I^{C4} &= sC_4(m_0^{C4} + m_1^{C4}s + m_2^{C4}s^2 + \dots)
 \end{aligned} \tag{3.2-2}$$

根据该结果即可将电容用相应数值的电流源代替，见图 3-6。

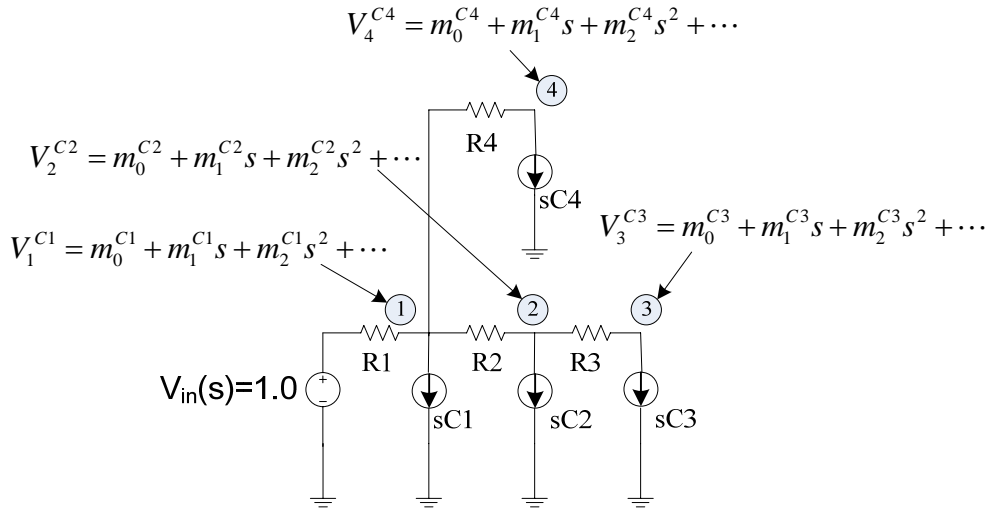


图3-6 电容等价于电流源

Fig.3-6 Capacitance modeled as current source

分析上述各电容电压及电流的计算式可以发现，令 $s=0$ ，则 $V_i^{C_i} = m_0^{C_i}, i=1,2,3,4$ ，因此可以通过此时的各电容电压求得各节点的零阶矩 m_0 ，此时的电容等价电流源由于没有常数项，因此在 $s=0$ 时均相当于开路。因此，我们可以假设此时所有的电容等价电流源均为0，继而通过直流电路分析算得各节点的电压，即零阶电路矩 m_0 ，见图 3-7。当有电感存在时，它们此时被等价于电压为0的电压源，因而对零阶矩的计算没有任何影响。

一阶电路矩是各节点电压频域表达式中 s 项前的系数。通过观察电容电压及电流的 s 域展开级数，并运用电路分解和叠加原理，我们知道，电压中的一阶 s 项是由电流中的一阶 s 项产生的，即电压中一阶项 s 的系数，一阶矩 $m_1^{C_i}, i=1,2,3,4$ ，可以通过电流源中一阶项 s 前的系数 $C_i m_0^i, i=1,2,3,4$ 计算得到。因此，从电路运算的叠加原理出发，我们可以将各电容等价电流源的电流表示为 $C_i m_0^i, i=1,2,3,4$ ，通过直流电路分析得到各节点的电压，即一阶电路矩。同理，各节点的 $q+1$ 阶电路矩均可以由 q 阶电路矩计算得到的电流

源，通过直流分析和叠加原理计算得到，如图 3-8所示。

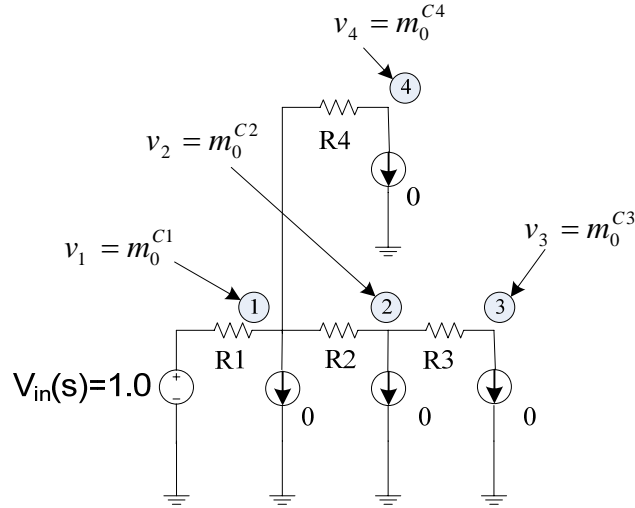


图3-7 直流计算等效电路

Fig.3-7 Equivalent DC analysis circuit

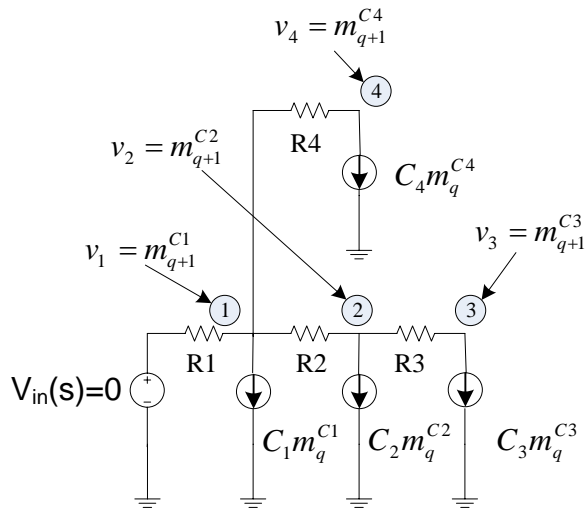


图3-8 由 q 阶矩计算 q+1 阶矩的直流等效电路

Fig.3-8 DC equivalent circuit for computing $q+1^{\text{th}}$ order moment from q^{th} order moment

根据上述讨论，我们知道，计算电路矩时首先需要计算电容的等效电流源数值，然后通过直流分析得到各电容即各节点的电压。路径追溯算法(path-tracing)通过对电路的

两次遍历即可完成这些工作，得到电路矩。第一次遍历用于计算所有支路的电流数值，第二次遍历即可计算得到所有节点的电压。

电流遍历过程：从电路生成树中的叶节点开始，采用后序深度搜索遍历每个节点。当节点被遍历到时，求得与它相连的所有后续支路的电流和，即为流过该节点的电流。后序深度遍历保证了在计算每一个节点的电流时，其所有后续节点的电流已计算得到。

电压计算遍历过程：上述遍历过程得到的通过各节点的电流和乘上节点支路对应的电阻即可以得到该节点与前续节点间的电压差。从生成树的根节点开始，对每个节点进行前序深度遍历，每个节点的电压等于前续节点的电压减去两节点间的支路电压差。使用前序遍历保证了在计算子节点的电压时，所有父节点的电压已得到。

3.2.2 符号化矩求解原理

路径追溯算法是一种数值计算方法，两次遍历过程通常采用递归方式实现，因此每次计算任一节点的某阶电路矩时，都需要重新计算所有节点的所有低阶电路矩，因此存在着大量的重复计算。为了避免这些重复的分析和计算过程，我们基于G.SHI^[41]提出的思想，首先实现了一种创新的求解符号化电路矩的算法。

该算法针对大规模互连线R(L)C树状模型，采用一种创新性的数据结构表示和存储电路矩的符号化计算表达式，从而将电路矩表示为电阻和电容等参数的符号化模型。这种算法可以快速准确的获得任意阶电路矩的符号化表达式，我们姑且称之为符号化矩算法。下面，我们以图3-9中所示的简单互连线RC树状模型为例，阐述符号化矩算法的基本原理。

如前文所述，RICE 算法中提出的用于计算电路矩的路径追溯算法通过对电路的两次遍历即可完成所有的工作。第一次遍历用于计算所有支路的电流数值，而第二次遍历就可计算得到所有节点的电压。两次遍历过程的方向是完全相反的。支路电流的计算需要将其所有后续节点的电路首先计算得到，因此需使用后序深度遍历；而节点电压的计算需在获得前续节点电压的基础上进行，因此需使用前序深度遍历。

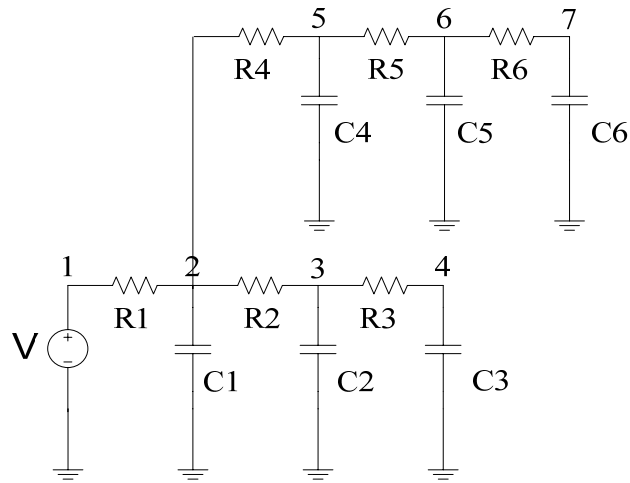


图3-9 简单 RC 树状电路

Fig.3-9 A simple RC tree

为了计算通过各节点的电流和，我们首先提出了由电路结构抽象而来，描述其连接关系的电容树结构，对应图 3-9 中的 RC 电路的电容树结构如图 3-10 所示。该结构与真实的电路结构相仿，每个抽象的节点中存储了其对应电容的电容值，抽象节点上隐含了加法和乘法操作，即每个抽象节点代表流过其存储的电容对应的电路节点上的电流和，这些电流根据所求矩阶数的不同由其对应的电容值乘上比所求矩低一阶的矩求得。抽象节点间的连接关系代表电路结构中的连接关系，父节点比子节点离电压源更近，如图 3-9 所示。因此存储 C_1 的抽象节点是整棵电容树的根节点，这代表流过电路节点 2 的电流是所有流过电路节点 3, 4, 5, 6, 7 的电流之和，与实际电路结构相符。

以一阶电路矩计算过程中的电流计算为例，在得到零阶矩的基础上，各抽象节点代表的电流值见图 3-10 中的黄色框中所示。各支路的电流可方便的通过遍历该电容树得到。流过父节点的电流等于所有流过子节点的电流与父节点电容所产生的电流之和。父节点电容对应的电流即为该电容与该支路对应节点上的低一阶电路矩的乘积。因此，电容树结构中的节点连接关系表示的是加法操作，每个节点中存储的电容信息隐含表示的是电容与低一阶电路矩的乘法运算，这一点在后面符号化矩的完整数据结构中会显示的更加清楚。

一阶矩计算过程中的支路电流计算

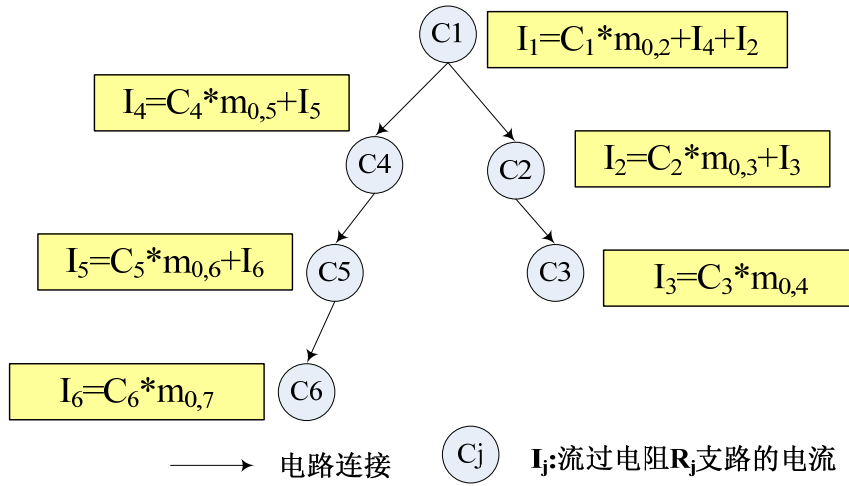


图3-10 电容树抽象结构

Fig.3-10 Extracted capacitance tree structure

在得到各支路电流的基础上，我们即可以计算各电路节点的电压，即电路矩。这一部分，我们采用一种不同于上述电容树结构的创新性数据结构，其灵感来源于二分决策图（BDD）。

二元判决图(Binary Decision Diagram, 简称BDD)最初由Aker等^[32]，用于表示一个基于有序变量集上的布尔函数。该结构的规范性和紧凑性使得存储表达式的空间需要大大降低，从而能够大大的提高操作速度，因此BDD自产生之后受到了广泛的关注。后来，Robert Bryant^[33]提出了简化的二元判决图（Reduced Ordered Binary Decision Diagram, 简称为ROBDD），由于其生成树的唯一性，被大量应用到各种逻辑综合工具以及电子设计自动化研究中。

二元判决图的页节点只有 0 和 1 两种，父节点的左右两分支分别代表不同的决策结果，例如左分支（0 分支）代表变量 x 取值为 0 后的布尔表达式，右分支（1 分支）则代表变量 x 取值为 1 后的布尔表达式。每个父节点只存在这两个分支，最终到达的叶子节点只能为 1 或 0。这样构造的二元判决图即隐性的存储了一个布尔函数表达式。二元判决图（BDD）使得对布尔函数的布尔操作可以通过简单的图论算法实现，大大降低了

计算复杂度。

在这里，我们借鉴二元判决图的思想，对其结构进行改进，在分支上加入多项式运算操作，从而使其能够隐性的给出符号化电路矩的求解公式，这种新的数据结构，我们姑且称之为矩决策树结构（Moment Decision Diagram，简称 MDD）。

在该抽象树结构中，存在两类节点。对应图 3-9 中的简单互连线模型的矩判决树结构如图 3-11 所示，以该矩决策树为例，树中的黄色节点代表电路矩，黄色节点中存储的电阻代表电路中该节点与前一节点间相连的电阻值。蓝色节点代表通过所标明支路的电流。节点间的虚线连接表示乘法操作，即父节点中所标明的电阻与该分支指向的子节点代表的电流相乘，此即为电流流经该电阻上的压降。节点间的点线连接表示加法操作，即父节点所代表的电路矩等于该加法分支所指向的节点矩加上右分支的运算结果，即电阻上的压降值。图中最下方也给出了一个三个节点组成的单元结构示例及说明。由于每个节点只有加法和乘法两个分支，矩判决树结构是类似二元判决图（BDD）的二分树结构。电路参数，例如电阻和电容等作为符号存储在节点中，因此该结构通过分支上的不同运算隐性的给出了电路矩的符号化计算公式。

注意到，该结构中的电流节点可由上文介绍的电容树计算得到，因此我们可以将两种数据结构进行结合，得到一种混合结构，仍以图 3-9 中的简单 RC 电路模型为例，其所对应的混合抽象结构如图 3-12 所示。图中下方由 C1~C6 构成的树是解析网表后得到的代表电路结构的电容树，上方 R1~R6 构成的树是符号化矩的求解树——矩决策树。每一个节点的矩都可以由加法分支指向的前一个节点的矩和乘法分支指向的电流以及节点代表的电阻进行运算得到。

图 3-12 中给出的结构仅仅是由某阶矩算得其高一阶矩的结构，不难发现，该结构是通用的。即从零阶矩开始，我们可以通过构造此结构得到各节点的一阶矩，不停重复此结构即可获得任意阶矩的计算公式。因此，即该结构隐性的存储了各节点任意阶矩的闭合形式表达式，此即为符号化矩的思想。当需要计算某节点某阶矩的数值时，我们只需找到代表该矩的节点，将该节点及其子节点中的各电阻值、电容值代入计算即可。

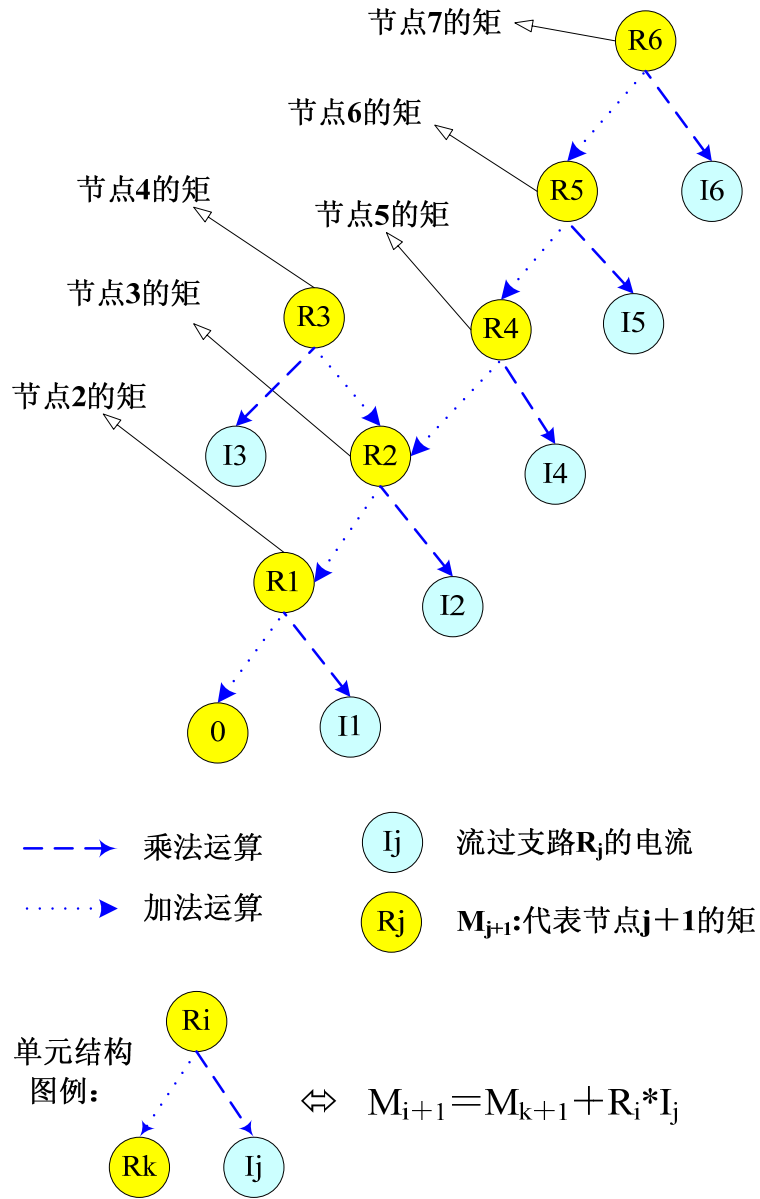


图3-11 矩决策树结构

Fig.3-11 Moment Decision Diagram structure

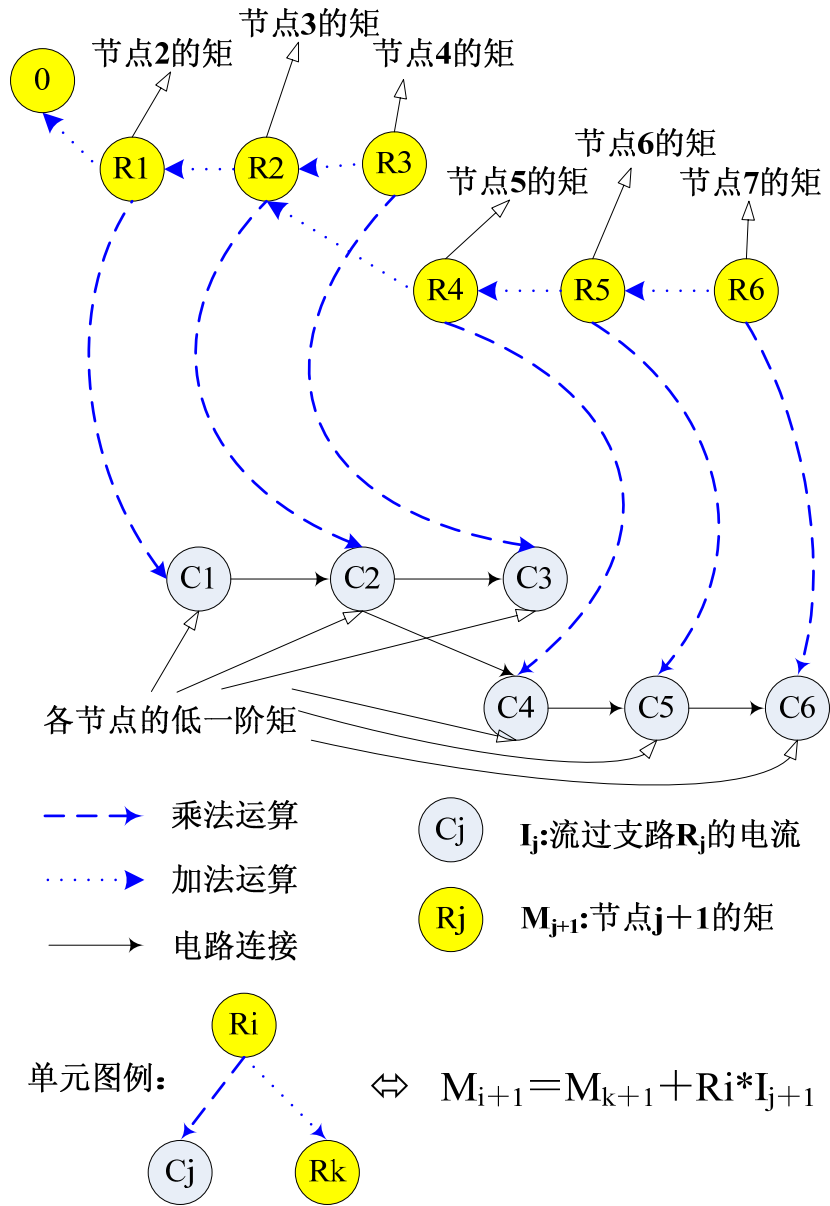


图3-12 求解符号化矩的混合数据结构

Fig.3-12 Hybrid structure for calculation of symbolic moments

符号化矩算法避免了手工进行公式的推导和近似，不再受限于类似^[7]中只能推导少数几个低阶电路矩的状况，可以快速准确的得到任何节点、任何阶电路矩关于电阻和电容等参数的闭合形式表达式。

利用符号化矩算法，只需解析一次输入的互连线网表结构即可建立起相应的抽象数

据结构，将电路的关系存储其内。并且，使得计算矩的效率大大提高。任意阶矩都可根据已计算过的低阶矩结果继续进行增量部分的计算得到，避免了数值计算方法中每次都重复计算所有阶矩的状况，因而大大提高了运算效率。这为我们后面提出的延迟二阶近似模型奠定了很好的基础。

3.2.3 符号化矩算法实现

符号化矩算法的实现主要分为两部分，一是对输入网表的解析，二是其混合树结构的建立。最后，任意节点上的任意阶矩都可以通过对该混合结构树的简单遍历得到。

该算法的输入网表采用类似 SPICE 语法的文本文件描述。C++解析器由 PCCTS (Purdue Compiler Compiler Tool Set) 生成。PCCTS 的好处是可以将编译器生成工具 Lex 和 Yacc 的功能集中在一个文件中完成，产生 C++代码的解析器。

在使用该解析器读入网表进行解析的同时，首先建立起电容树的抽象结构，同时存储各元件的参数值。为了能够存储存在任意多个分支的电路结构，适用如下定义的电容树节点：

```
typedef struct _treeNode_
{
    char *name;        //name of the element represented by the node, eg. R12, C4, L22
    int index;        //index of the element
    double value;    //value of the node
    EdgeType type;   //enum EdgeType of the edge, Y/Z/CC/VC/CS/VS/R/C/L
    int startnode;   //index of the starting circuit node of the edge
    int endnode;     //index of the end circuit node of the edge
    struct _treeNode_ *firstchild;    //the first edge connected to this node
    struct _treeNode_ *siblingchild;  //other edges connected to this node if there are
}treeNode;
```

其中，指针 firstchild 用于指向当前节点的后续节点，而指针 siblingchild 用于指向与

当前节点相连接的父节点的其他扇出节点。通过这两个指针的使用，我们就可以实现存储具有任意扇出的电路。**Name** 用于存储元器件在网表中的名字，**index** 用于存储电容的编号，该编号是网表中电容非地端节点的编号减 1，由于所产生的网表中的节点编号是唯一的，因此电容的编号均是唯一的。节点对应的电阻的编号默认与电容编号相同。**Value** 用于存储该电容的参数值，**EdgeType** 用于表示该元器件所处的支路是否为电容支路或电源支路。**Startnode** 存储该电容所在支路近源端点的编号，**endnode** 存储支路另一端点的编号。每个电容节点隐含的代表了电流值，遍历其子节点构成的树即可得到其电流值，因为我们无需存储该电流值。

为了能方便快速的得到任何节点的电流，我们使用一个简单的哈希表，将节点的编号作为他们的哈希值，存储在哈希表中相应的位置。由于网表中节点的编号是唯一的，因为哈希表中的每个单元只指向一个节点。这样使得我们可以随时得到任何节点的信息。

在解析网表结束后，接下来，我们按照上文介绍的矩决策树算法在电容树的基础上建立混合的抽象树结构。观察上文介绍的抽象树结构会发现，电容树所存储的连接关系与矩决策树中的连接关系几乎相同，只是方向相反，因此，我们只需遍历一次电容树即可建立其对应的矩决策树结构。矩决策树中的节点结构定义如下：

```
//Structure of the Polynomial BDD with '+' & '*' operation
typedef struct _bddNode_
{
    char *name;
    double value;          //value of moment at the node
    int node_index;       //ele_index = node_index-1
    EdgeType type;
    int mmt_order;        //order of the moment
    treeNode* multbranch; //point to capacitor tree
    struct _bddNode_ *addbranch;//point to bdd nodes
}
```

```
}bddNode;
```

其中, `name` 存储电阻在网表中的名字, `value` 存储网表中读入的电阻数值, `node_index` 代表该节点的编号, 其相应的电阻编号为节点编号减一。此关系是随机产生网表时为了标记的简单而采用的, 以简单的限制降低程序额外的复杂度。`EdgeType` 表明该支路为电源支路还是电阻、电容支路等, `mmt_order` 代表该节点所代表的矩的阶数, `multbranch` 代表乘法分支, 即指向电容树中的节点, `addbranch` 代表加法分支, 指向子决策树节点。同样, 我们使用一个简单的哈希表以保证随时快速找到任意我们需要的节点信息。

3.3 延迟的二阶近似模型

基于上面一节我们提出的符号化矩算法, 我们得到了任意节点上的任意阶矩关于电阻和电容的符号化表达式, 即隐性的给出了计算矩的准确的闭合形式表达式。将其与本章第一节介绍的电阻和电容的统计模型相结合, 我们就得到了电路矩关于偏差随机变量, 如宽度偏差、高度偏差等的准确的非线性闭合形式表达式。

在此基础上, 为了获得互连线延迟关于偏差随机变量的闭合形式表达式, 我们进一步采用基于矩的延迟度量算法由电路矩直接计算得到互连线延迟。基于矩的延迟度量算法提供了闭合形式的计算公式, 其本质上是采用修正后的一阶和二阶等低阶系统近似原高阶系统, 根据经验或对电路的分析提出用于修正低阶近似系统的主极点和余式等的各种方法, 从而得到简单的计算公式。这种闭合形式的延迟度量算法可以提供简单高效的估计, 然而由于采用低阶近似系统, 与利用高阶矩匹配的模型降阶算法相比, 准确度稍差。而在处理互连线的统计时序分析问题, 我们希望找到快速而又准确的算法, 在保证误差的基础上能够高效的得到互连线延迟的概率密度分布, 因此具有闭合形式公式的延迟度量算法是最好的选择。经过实验和误差分析, 针对不同的应用, 我们采用了不同的延迟度量算法, 分别是适合于单条互连线的C.Alpert等^[20]提出的D2M算法, 适合于互连线树状网络的B. Tutuianu等^[19]提出的单步New-Raphson迭代求解算法, 及适用于求解远端点延迟的Y. I. Ismail等^[24]提出的使用高至七阶矩的高阶延迟度量算法。详细的实验数据可参照第四章。这里, 我们首先以使用一阶至三阶矩的m3NR算法^[19]为例, 介绍

说明本文采用延迟度量矩阵的整体算法的流程。

利用 m3NR 算法的公式，我们可以将延迟表示为偏差随机变量的函数，得到其闭合形式表达式如下：

$$t_{NR} = p_1 \ln\left(\frac{2k_1}{p_1}\right) + \frac{-0.5 + \frac{k_1}{p_1} e^{-p_1 t_1} + \frac{k_2}{p_2} e^{-p_2 t_1}}{k_1 e^{-p_1 t_1} + k_2 e^{-p_2 t_1}} = f(\delta(z)) \quad (3.3 - 1)$$

其中， $\delta(z)$ 如本章第一节中所述，代表电路中存在的的所有偏差随机变量。

至此，我们将互连线延迟建模成为物理参数偏差随机变量的函数，且没有进行任何线性或高阶的近似，完整的保留了从物理参数到延迟的复杂的非线性映射关系和各物理参数间的相关性。

到现在为止，大部分的统计时序分析方法通常都有一个前提假设，即认为互连线延迟的概率密度分布是高斯分布，由于线性操作能够保留高斯分布的特性，因此在假设电容电阻等电气参数的概率分布也为高斯分布的前提下，只需将互连线延迟建模为电气参数等的线性模型，然后通过蒙特卡罗方法采样即可得到互连线延迟的概率密度分布结果。然而由于互连线延迟与互连线的电气参数及物理参数之间存在很复杂的非线性关系，即使假设互连线宽度 W 和高度 H 等的偏差为具有正态分布的随机变量，他们的乘积和商并不一定是正态分布，而经过复杂非线性运算得到的互连线延迟复合正态分布的可能性更小。因此，采用线性模型在偏差很大的情况下，例如最大可能偏差在 35%~50% 的时候，忽略非线性部分造成的误差会很大。因此为了避免蒙特卡罗方法，降低后续概率分布计算的复杂度，同时保证能够建模非线性关系对延迟概率密度分布的影响，这里我们提出一种延迟的二阶近似模型。

利用上面得到的延迟关于偏差变量的闭合形式表达式，将其在零点处进行泰勒展开，保留二次及二次以下的项，忽略高次项，即可得到二阶近似模型。假设互连线的延迟为 t_d ，即有：

$$t_d = \tilde{t}_d + t_d' \delta + \delta^T t_d'' \delta \quad (3.3 - 2)$$

其中, $\delta = [\delta_1, \dots, \delta_p]^T \sim N(0, \Sigma)$, 是代表 p 个偏差来源的偏差随机变量, 具有本章第一节中所讨论的性质。 Σ 是 $p \times p$ 维的相关性矩阵。 \tilde{t}_d 是不存在偏差时的延迟均值, 可由 $\delta = [\delta_1, \dots, \delta_p]^T = 0$ 计算得到。 $t'_d \in R^{1 \times p}$ 是偏差随机变量一次项前的系数, 即延迟在偏差零点处的一阶导数值, $t'_d(k) = \frac{\partial t_d}{\partial \delta_k}, k = 1, \dots, p$ 。 $t''_d \in R^{p \times p}$ 是二次项的系数矩阵, 即延迟在偏差零点处关于各偏差变量的二阶导数值, $t''_d(k, l) = \frac{1}{2} \frac{\partial^2 t_d}{\partial \delta_k \partial \delta_l}, k, l = 1, \dots, p$ 。

延迟的一阶及二阶导数可采用数值计算方法得到, 即根据定义, 假设偏差变量存在一个很小的变化, 通过变化前与变化后的数值求得导数。由此原理推导得到的数值求导公式如下:

$$\left. \frac{\partial t_d}{\partial \delta_k} \right|_{\delta_k=0} \approx \frac{t_d(\Delta \delta_k) - t_d(0)}{\Delta \delta_k} \quad (3.3-3)$$

$$\left. \frac{\partial^2 t_d}{\partial \delta_k \partial \delta_l} \right|_{\delta_k=0, \delta_l=0} \approx \frac{t_d(\Delta \delta_k, \Delta \delta_l) - t_d(0, \Delta \delta_l) - t_d(\Delta \delta_k, 0) + t_d(0, 0)}{\Delta \delta_k \cdot \Delta \delta_l} \quad (3.3-4)$$

其中, $\Delta \delta_k, \Delta \delta_l$ 等均是所给定的偏差的微小变化量。

由该公式可知, 若存在 p 个偏差来源, 则需要求解 p 个延迟的一阶导数, p^2 个延迟的二阶导数。在求解每个一阶导数时均需要计算两次不同偏差下的延迟数值, 而计算每个二阶导数时需要重复计算四次不同偏差下的延迟数值。因此, 总的延迟计算次数为 $2p + 4p^2$ 次。当所分析电路的规模很大时, 使用数值方法计算各节点的延迟将耗费很多时间, 尤其是求解远端点的延迟或使用高阶矩的延迟度量算法时。而本文提出的算法流程由于采用了符号化方法, 只需分析一次就可以得到延迟关于偏差变量的闭合形式表达式。因此, 对导数的计算可通过简单的改变各变量的数值计算得到, 无需重复整个分析过程, 因而大大减少了运算时间, 提高了算法的效率。具体的算法效率测试数据在第四章中给出。

3.4 延迟概率分布的求解算法

在得到了互连线延迟关于偏差随机变量的二阶近似模型后，本文采用一种基于特征函数的方法^[34]，可以直接由二阶模型分析计算得到延迟的概率密度分布，无需蒙特卡罗仿真，因而具有很好的效率和很低的计算复杂度。

在本文中，我们假设偏差随机变量具有高斯分布，且它们之间存在一定的相关性。这是寄生参数提取文献中使用的假设。下面我们首先给出多维高斯变量的特征函数等基础知识。

根据概率理论中的定义，随机变量 X 的特征函数可由其概率密度分布函数计算得到：

$$C_X(w) = E(e^{iwx}) = \int_{-\infty}^{+\infty} e^{iwx} f_X(x) dx \quad (3.4-1)$$

其中， $f_X(x)$ 是随机变量 X 的概率密度函数。

反之， $f_X(x)$ 可以通过对特征函数 $C_X(w)$ 的傅立叶变换直接计算得到：

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-iwx} C_X(w) dw \quad (3.4-2)$$

均值为 μ 、方差为 σ^2 的一元高斯随机变量，其概率密度函数为：

$$f_\xi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.4-3)$$

根据定义可以得到其对应的特征函数为：

$$\Phi_\xi(u) = e^{j\mu u - \frac{\sigma^2 u^2}{2}} \quad (3.4-4)$$

而对于均值为零、协方差矩阵为 $B = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$ 的二元高斯随机变量 ξ_1, ξ_2 而言，其特征

函数可以通过其二元概率密度函数推导得到：

$$f_{\xi_1\xi_2}(x_1, x_2) = \frac{1}{2\pi\sqrt{1-r^2}} \exp\left(-\frac{1}{2(1-r^2)}[x_1^2 - 2rx_1x_2 + x_2^2]\right) \quad (3.4-5)$$

$$\Phi_{\xi_1\xi_2}(u_1, u_2) = \exp\left(-\frac{1}{2}[u_1^2 + 2ru_1u_2 + u_2^2]\right) \quad (3.4-6)$$

进一步，扩展到 n 元高斯随机变量 ξ ，其均值、协方差矩阵（正定的）分别为：

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} \quad (3.4-7)$$

则其 n 元概率密度和特征函数可推导得到：

$$f_{\xi}(x) = \frac{1}{[(2\pi)^n |B|]^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T B^{-1}(x-\mu)\right) \quad (3.4-8)$$

$$\Phi_{\xi}(U) = \exp\left(j\mu^T u - \frac{1}{2}u^T B u\right) \quad (3.4-9)$$

其中， $x = (x_1 \ x_2 \ \dots \ x_n)^T$ ， $u = (u_1 \ u_2 \ \dots \ u_n)^T$

根据上述结论，由延迟 t_d 的二阶模型(3.3-2)及偏差随机变量具有高斯分布的假设可以推得其特征函数的计算公式为：

$$C_D(w) = |\Omega|^{-\frac{1}{2}} \exp\left\{i w \tilde{t}_d - \frac{1}{2} w^2 (t_d'^T \Sigma^{\frac{1}{2}} \Omega^{-1} \Sigma^{\frac{1}{2}} t_d')\right\} \quad (3.4-10)$$

其中 $\Omega = I - 2i w \Sigma^{\frac{1}{2}} t_d' \Sigma^{\frac{1}{2}}$ 。于是根据定义式(3.4-2)，接下来只需对 $C_D(w)$ 进行傅立叶变换就可立即得到延迟 D 的概率密度分布。

至此，我们就得到了一个完整的可用于快速准确的计算互连线延迟等时序信息概率密度分布的统计时序分析方法。

3.5 算法流程及优势

总结前面各小节所介绍的方法，本文所提出的符号化统计时序分析算法的完整流程如图 3-13。

在给定互连线分布式 RC 模型后，电阻和电容等参数首先被建模成为物理参数的偏差随机变量的统计模型，例如互连线的宽度，高度和金属层的厚度等；进而利用符号化矩算法可求解得到任意目标节点上的任意阶电路矩的符号化表达式，即闭合形式表达式；将其与闭合形式延迟度量矩阵结合，我们就得到了互连线延迟等时序信息关于物理

参数偏差随机变量的闭合形式非线性模型；为了保留延迟等时序信息与物理参数之间的非线性关系对延迟的概率密度分布所造成的影响，同时提高计算效率，我们将互连线延迟采用二阶模型进行近似，进一步通过其特征函数的解析表达式和傅立叶变换直接得到互连线延迟等时序信息的概率密度分布结果，实现了我们的互连线统计时序分析的最终目标。这种新的算法流程具有很多优势，例如只需分析一次即可得到延迟的符号化表达式，大大加快了存在大量重复计算的二阶模型近似过程。且整个流程中由于采用了符号化算法，将延迟的闭合形式表达式隐性的存储在计算机内部，只需进行一次延迟的二阶模型近似，无需在每一阶段，如计算电容、电阻时，计算矩和计算延迟时均进行近似，因而运算过程简单高效。该方法与已有的基于模型降阶算法的统计时序分析方法相比，计算量非常小，速度非常快，且有较好的准确度，对大部分端点的近似误差非常小。

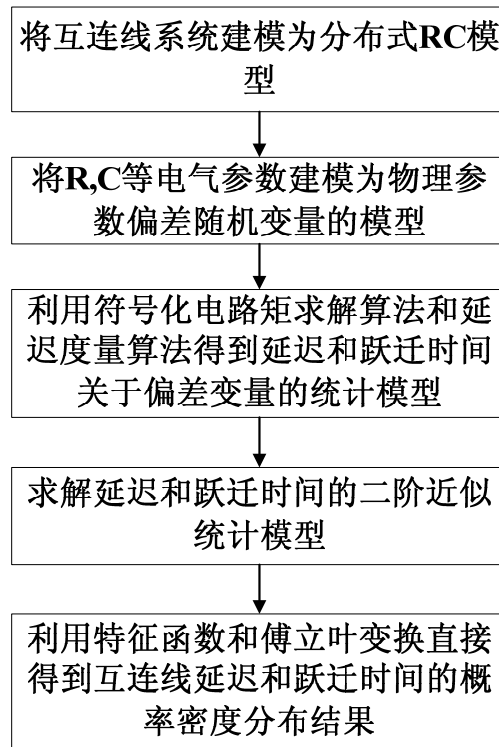


图3-13 互连线统计时序的符号化分析方法

Fig.3-13 Symbolic analysis method for statistical timing modeling of interconnects

拥有了互连线延迟的概率分布信息，我们就可以得到延迟等随机变量概率分布的 3σ 处对应的最大和最小延迟数值。该统计时序信息可以被应用到诸多场合，例如设计优化

问题。为了寻找最优化的互连线设计结构以满足设计性能和信号可靠性等设计要求，例如最优的互连线拓扑结构，最优的互连线宽度和间隔，最优的缓冲器插入位置和大小等，我们往往会采用一些成本函数(cost function)给出设计好坏的度量标准。延迟做为关键信息，通常是成本函数的决定性因素之一。例如有些成本函数即以串扰和延迟做为参数^[35]：

$$\text{costfunction} = \alpha \cdot f(\text{crosstalk}) + \beta \cdot f(\text{delay})$$

得到了延迟的统计信息后，我们还可以进一步得到成本函数（costfunction）对制造偏差（process variation）的敏感度（sensitivity），以辅助进行更好的设计。

另外，本文提出的算法得到的延迟的二阶近似模型还可以被应用到集成电路的统计静态时序分析算法（SSTA）中。现在许多的统计静态时序分析算法为了建模延迟分布中的非线性特征，纷纷采用二阶模型对门延迟及互连线延迟进行建模，这就需要获得互连线延迟的二阶模型。我们这里得到的互连线二阶模型可以直接用于此种算法中，提供研究的基础模型。另外，还有一些统计静态时序分析算法是基于离散化的延迟概率密度分布进行的^{[36][37][38]}，因此我们得到的互连线延迟概率分布能够为之提供最直接和准确的输入信息。

文中算法不受偏差来源个数的影响，可用于包含任意多个偏差来源的情形。另外，我们这里对偏差随机变量没有施加任何独立性的假设，有别于以往的绝大部分统计建模算法，并且近似步骤少，因此减少了误差的产生，这一点也可由实验结果得证。

3.6 本章小结

本章提出了一种新的快速准确的分析互连线统计时序信息的算法流程，通过结合电气参数的统计模型，创新性的符号化矩算法，闭合形式延迟度量算法与二阶模型近似方法，并通过特征函数及傅立叶变换的相关运算，能够快速准确的得到互连线延迟的概率密度分布信息。

第四章 实验与结果分析

本文采用 C++ 编程语言实现算法，能够处理任意互连线 RC 树状结构，通过解析类似 SPICE 语法的网表文件，建立起符号化矩的抽象矩决策树结构，最终计算得到目标节点互连线延迟的二阶模型；然后利用 MATLAB 软件的矩阵运算特性，通过特征函数的分析计算和傅立叶变换得到互连线延迟的概率分布结果。

本章首先给出符号化互连线统计时序算法的正确性验证介绍，接着将该算法应用于大量测试电路，给出了算法的效率测试结果与针对不同电路结构的误差分析结果，包括单根传输线，一般的互连线网络及存在 sizing 的时钟树网络等，并讨论了最大偏差范围及偏差间的相关性对延迟概率密度分布的影响，在最后将其分别与两种已有的统计时序分析算法分别进行了比较。

实验证明，该统计时序分析方法得到的概率分布与 SPICE 软件蒙特卡罗仿真结果相比，误差稳定在 1% 以内，且运算效率高，可以处理大规模的电路，包括含有十万节点以上的单条互连线结构以及含有三十万节点以上的互连线分支网络结构。

4.1 算法正确性验证

在测试整个互连线的符号化统计时序分析流程之前，我们首先在不考虑偏差的情况下，验证算法设计的正确性，并分析利用基于矩的延迟度量算法计算互连线延迟与真实结果相比存在的误差。稍后由实验结果可知，这部分误差是造成统计时序分析结果误差的主要部分。

4.1.1 符号化矩算法验证

为了保证符号化矩算法所得各节点电路矩结果的正确性，我们分两步对其进行正确性验证。

首先，对相同的输入互连线网表文件，我们将符号化矩算法得到的各节点的一阶矩

结果与 SPICE 电路仿真软件的结果进行比较。由第二章中的讨论可知，节点的一阶矩即为该节点处的 Elmore 延迟信息，而 Elmore 延迟做为对节点真实延迟的较好近似，应与 SPICE 仿真结果中提取到的该节点对应 50%变化处的延迟时间相近。利用该方法，我们对随机产生的大量互连线输入网表进行了测试，并对网表结构中的所有节点都进行比较，从而可以确定符号化矩算法一阶矩计算的正确性，初步保证了符号化矩算法实现的正确性。

对一个随机产生的具有 100 个节点的互连线树状结构进行测试，利用符号化矩算法得到的 Elmore 延迟和 SPICE 仿真结果如下。由图中可见，两者的变化趋势是完全一样，虽然由于 Elmore 延迟的误差使得两者存在数值上的差异，但我们仍可以肯定算法计算一阶矩结果的正确性。

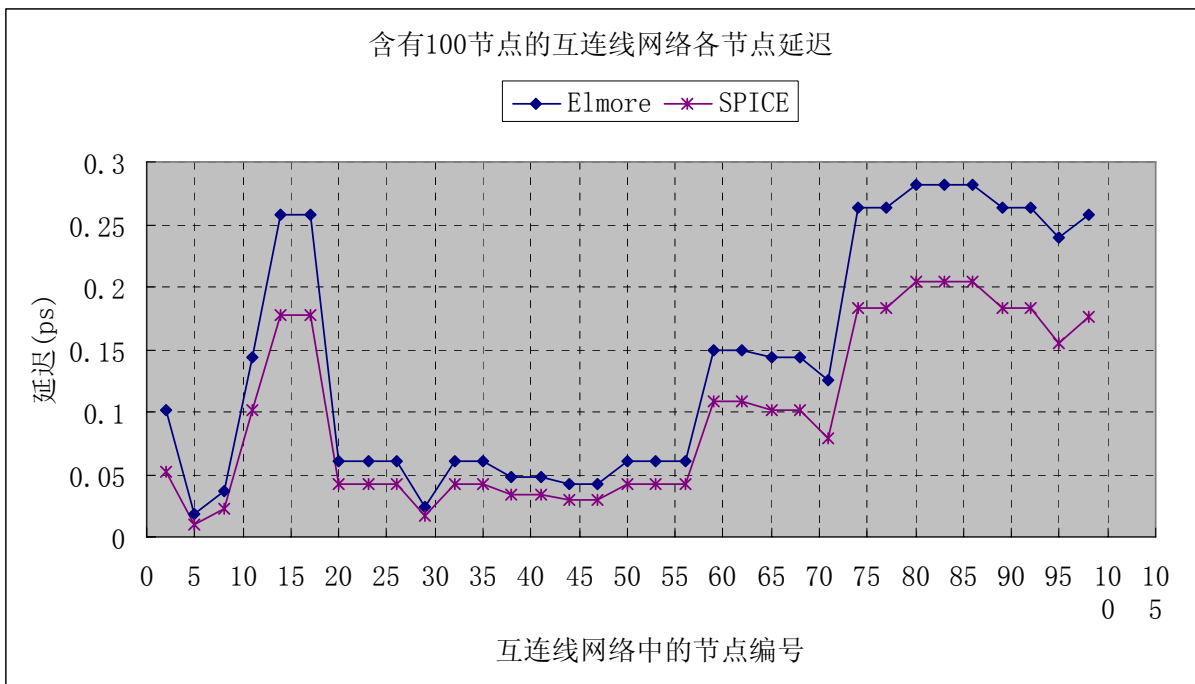


图4-1 符号化算法一阶矩结果验证

Fig.4-1 Verification of symbolic moment algorithm

其次，为了验证符号化矩算法计算高阶矩的准确性，我们将其与非符号化方法，即数值方法，实现的路径追溯算法结果进行比较。对同一个随机产生的互连线网表输入，

若两种方法得到的任意节点上的任意阶矩结果均相同，则可保证符号化矩算法计算高阶矩的正确性。此处，我们略去得到的完全一样的延迟测试结果，本文中使用的算法已经过该步测试。

4.1.2 延迟度量算法比较

在本文提出的互连线统计时序分析方法中，我们利用了延迟度量矩阵由矩计算得到延迟。在第二章中，我们给出了到目前为止所有已提出的延迟度量算法的介绍，包括 Elmore 延迟，标量化的 Elmore 延迟，DM1 算法，DM2 算法，D2M 算法，B. Tutuianu 等提出的利用一至三阶矩及单步 Newton-Raphson 迭代的方法（暂且简称为 m3NR 算法）及 Y. I. Ismail 等提出的可利用高至七阶矩的近似方法（根据所使用的矩的最高次数的不同暂且分别称之为 Hm2Delay, Hm3Delay, Hm4Delay, Hm5Delay, Hm6Delay 和 Hm7Delay 算法）。实验证明，针对不同结构的电路，他们的效果不尽相同。下面我们将利用不同延迟度量矩阵得到的节点延迟与 SPICE 软件的真实仿真结果进行比较，给出具体结果与误差的分析。

如第三章所述，本文采用的是互连线的分布式 RC 树状模型，在进行分布式建模时每段互连线的长度相同，因此每段互连线的电阻、电容等参数均相同。而不同的互连线结构可以被建模成具有不同段数及不同分支结构的分布式 RC 模型，因此为了证明本文所提算法的普适性，即可以处理具有任意段数及分支数的互连线结构，我们的测试电路均为随机产生。所产生测试电路中所包含的总节点数表明了电路的规模，节点的编号按照从近源点到远端点的方向增长，因此小编号的节点代表近源点，节点编号越大，离电源的距离越大，编号最大的节点即是电路中离电源距离最远的节点。

下面，我们以随机产生的分别含有 30 个节点与 100 个节点的单条互连线结构和存在分支的互连线树状结构，给出不同延迟度量矩阵的测试结果。他们可以分别代表不同结构的电路给出具有代表性的结果。网表中的电阻、电容等数值均以 ITRS^[1]所给出的 0.10 微米制造工艺下的版图物理参数为参照进行计算得到。每段互连线的电阻值均在 1—10 欧姆量级，电容值在 1—10fF 量级。

下面给出的各测试例子结果均包括使用不同延迟度量矩阵及 SPICE 求得的测试电路各节点的延迟, 各节点延迟与 SPICE 比较的相对误差以及各测试电路所有节点相对误差的平均值和标准差。

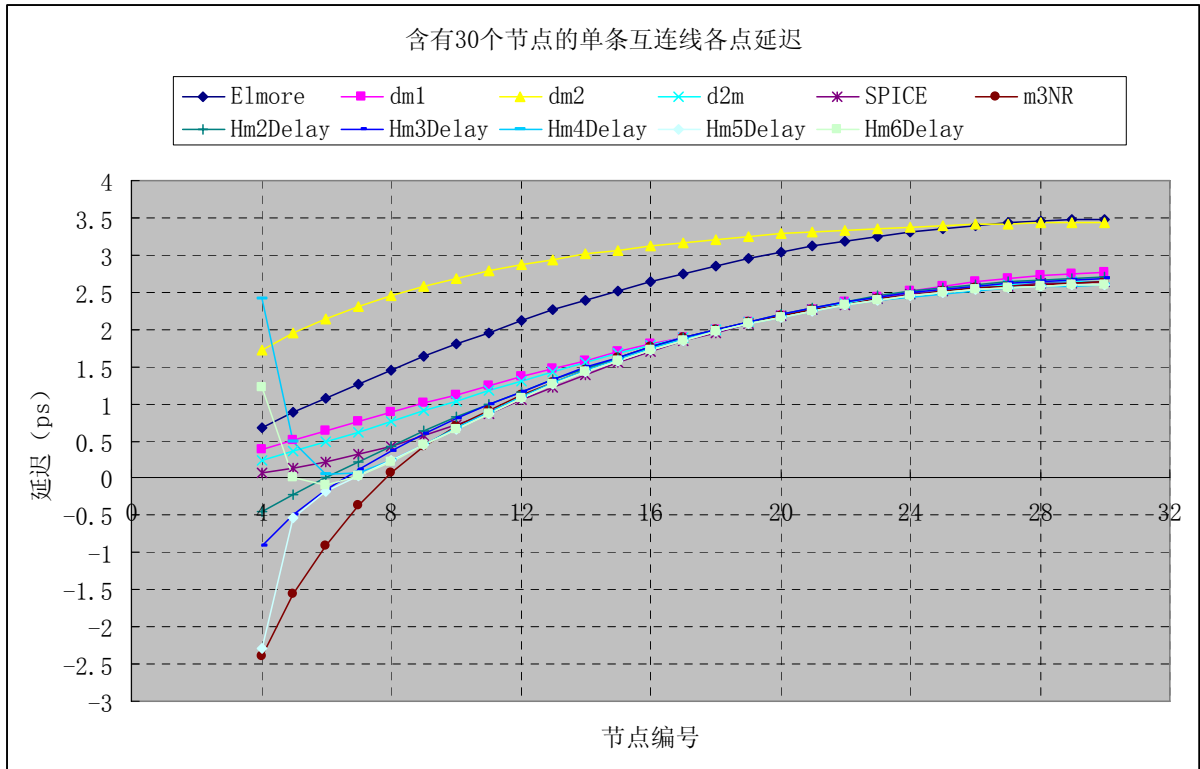


图4-2 30 节点单条互连线延迟比较

Fig.4-2 Delay of single line including 30 nodes

表4-1 30 节点单条互连线延迟的相对误差统计

Table4-1

统计所有节点延迟的相对误差 (%)						
	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	248.16	112.20	777.37	55.75	2.63E+03	
标准差	488.43	258.41	2133.95	102.94	1.12E+04	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	4.71E+02	1.38E+03	2.80E+04	1.10E+05	2.14E+05	2.56E+08
标准差	1.93E+03	6.12E+03	1.42E+05	5.72E+05	1.12E+06	1.35E+09
统计所有编号在后 2/3 的节点						

	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	55.58	10.05	81.20	7.19	1.73	
标准差	32.74	14.94	66.38	12.24	2.01	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	4.18	3.68	1.91	1.33	1.29	1.47
标准差	4.02	3.76	1.38	1.48	1.49	1.49

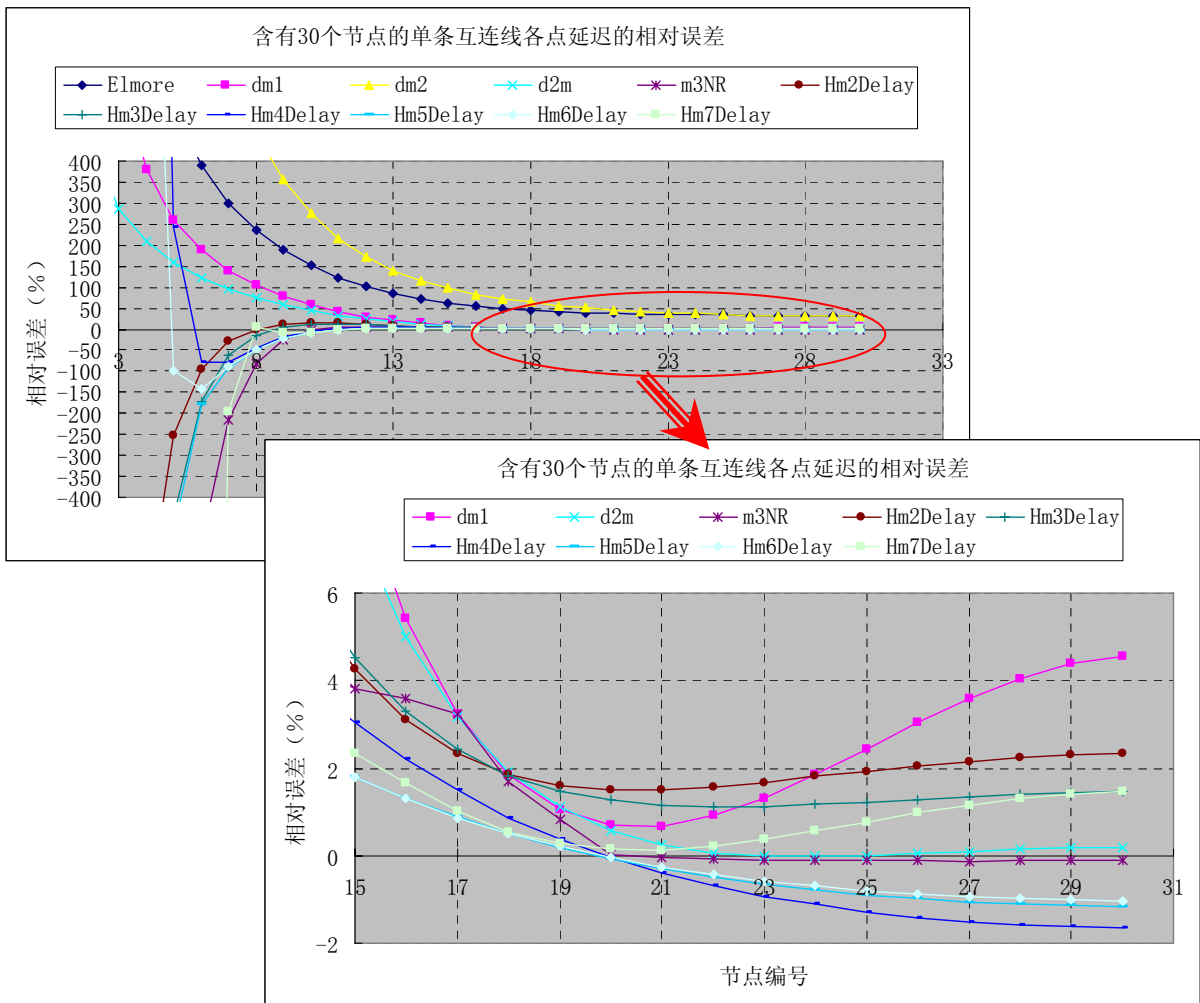


图4-3 含 30 节点的单条互连线各节点延迟的相对误差

Fig.4-3 Relative Error of delay at each node for single line with 30 nodes

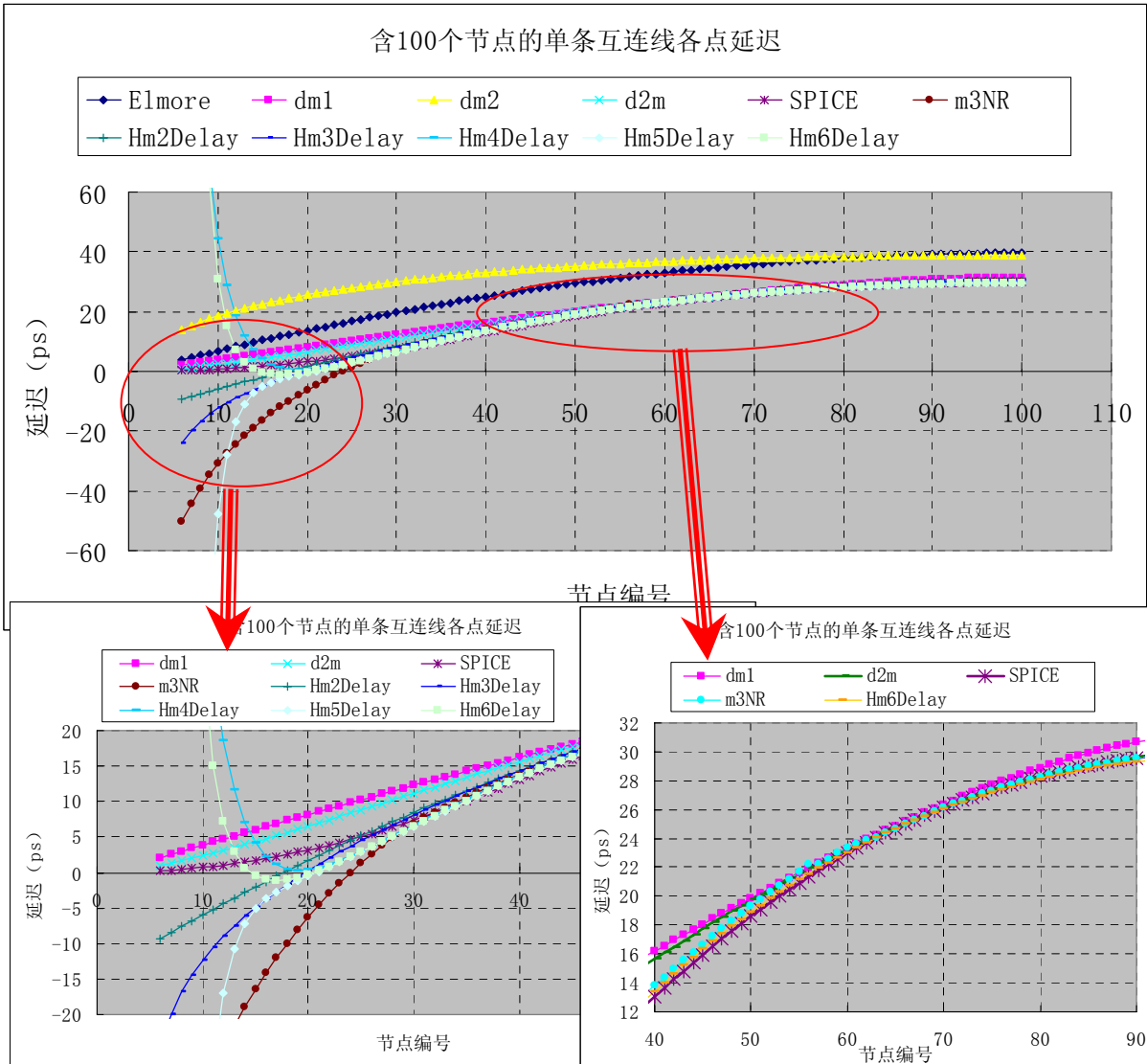


图4-4 100 节点单条互连线延迟比较

Fig.4-4 Delay of single line including 100 nodes

表4-2 100 节点单条互连线延迟的相对误差统计

Table4-2

统计所有节点延迟的相对误差 (%)						
相对误差	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	3.54E+02	1.67E+02	1.64E+03	6.76E+01	-1.56E+04	

标准差	1.03E+03	5.36E+02	7.66E+03	1.43E+02	1.19E+05	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	-2.11E+03	-1.41E+04	1.45E+06	-2.25E+07	1.69E+08	1.16E+00
标准差	1.51E+04	1.19E+05	1.34E+07	2.21E+08	1.65E+09	5.35E+12
统计所有编号在后 2/3 的节点						
	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	52.88	8.72	75.72	6.15	1.61	
标准差	28.59	12.61	58.08	10.53	2.07	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	3.93	3.51	0.52	0.01	0.08	0.85
标准差	3.71	3.62	1.83	1.43	1.36	1.20

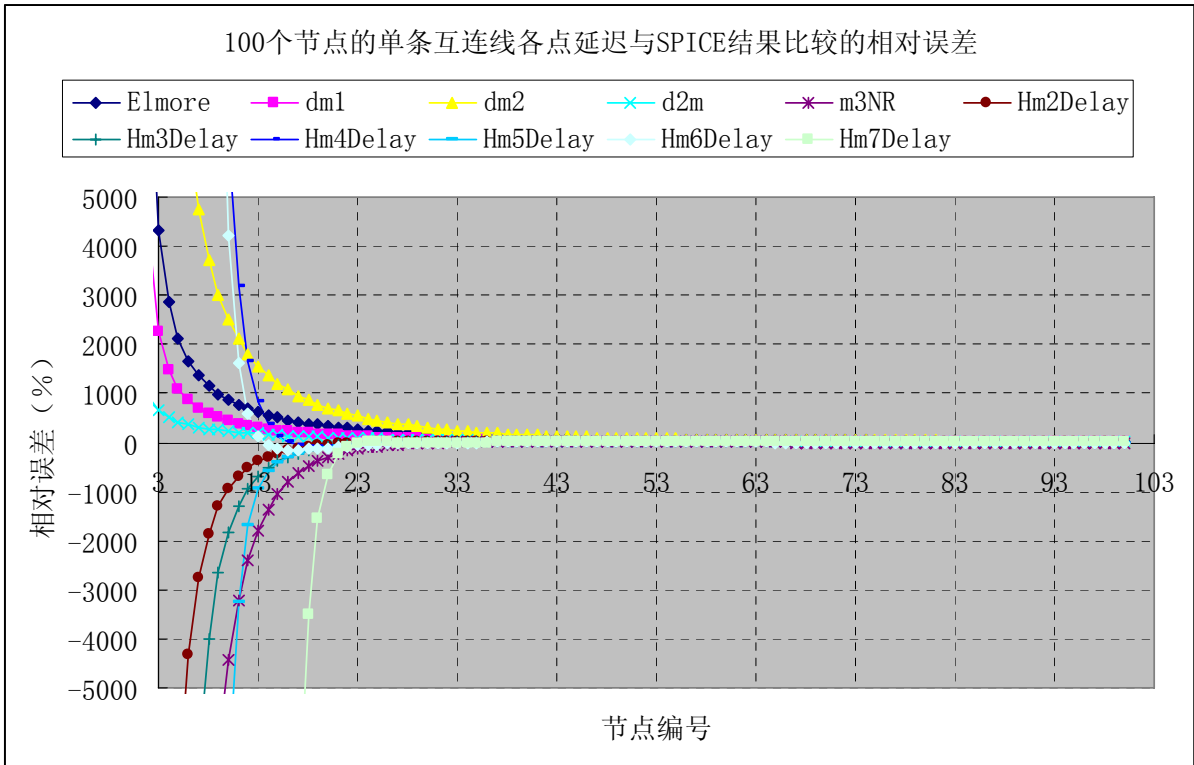


图4-5 100 个节点的单条互连线不同延迟度量方法所得各点延迟的相对误差

Fig.4-5 Relative Error of Delay at each node for single line with 100 nodes

上图中后 2/3 节点的相对误差放大图如下：

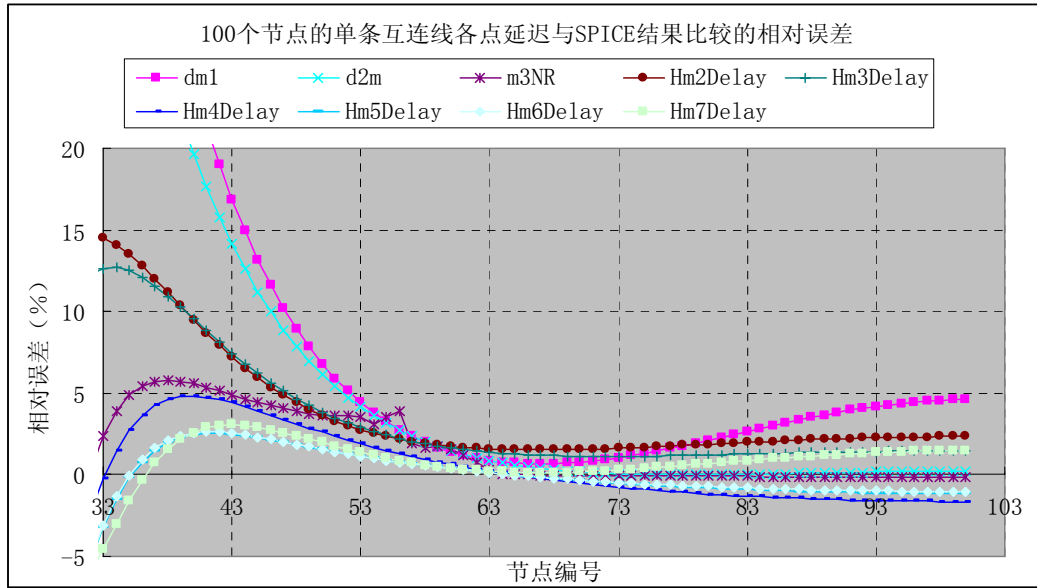


图4-6 100个节点的单条互连线各点延迟的相对误差放大图

Fig.4-6 Magnified Curve of Delay Relative Error

在图 4-2 和图 4-4 中分别给出了单条互连线上各节点使用不同延迟度量算法得到的延迟估算值及 SPICE 仿真得到的准确值。观察曲线可发现，Elmore 延迟和 dm2 算法对节点延迟的估算最保守，然而他们和 dm1 算法、d2m 算法一样，得到的估算值始终是正的，因此这些方法比较稳定。而 m3NR 算法和利用高阶矩的 Hm*Delay 算法在近源点会得到负的延迟估算值，有悖实际电路的物理特性，因此没有意义。除了 Elmore 延迟和 dm2 算法，其他延迟度量算法对远端点的延迟近似程度均好于对近源点的延迟近似，在节点编号相当于最大节点编号的 2/3 处近似程度最好。各种延迟度量算法对各节点延迟估算与 SPICE 结果比较的相对误差也显示在图 4-3，图 4-5 和图 4-6 中。从中可以发现，近源点延迟估算的相对误差非常大，有些算法的估算值误差太大以至于完全没有实用价值。另外，近源点的估算误差均大于远端点，若只考虑距离电源最远的 2/3 部分节点的误差，则得到的节点误差平均值远远小于所有节点误差的平均值，这些数据在表 4-1 和表 4-2 中给出。d2m 算法较稳定，其对近源点和远端点延迟的近似均较好；m3NR 算法对近源点的估算会产生负值，但对 2/3 远端点的延迟估算误差非常小，分别只有 1.73% 和 1.61%，其标准差也非常小，显示其估算误差波动很小；Hm*Delay 系列算法对近源

点估算也会产生不可容忍的巨大误差和负值，但对后 2/3 节点的估算误差是最小的，如 Hm5Delay 算法仅有 1.33% 和 0.01%，Hm6Delay 算法仅 1.29% 和 0.08%，因此这些算法是对远端点延迟的最好近似算法。

以上都是对单条互连线结构给出的测试结果，下面我们分别对具有 30 个节点和 100 个节点的存在分支的互连线网络进行测试。这些互连线网络均为随机产生，最大可能分支数为 9，最小为 1，各节点的扇出数随机决定，节点仍然大致按照距离电源的远近编号。

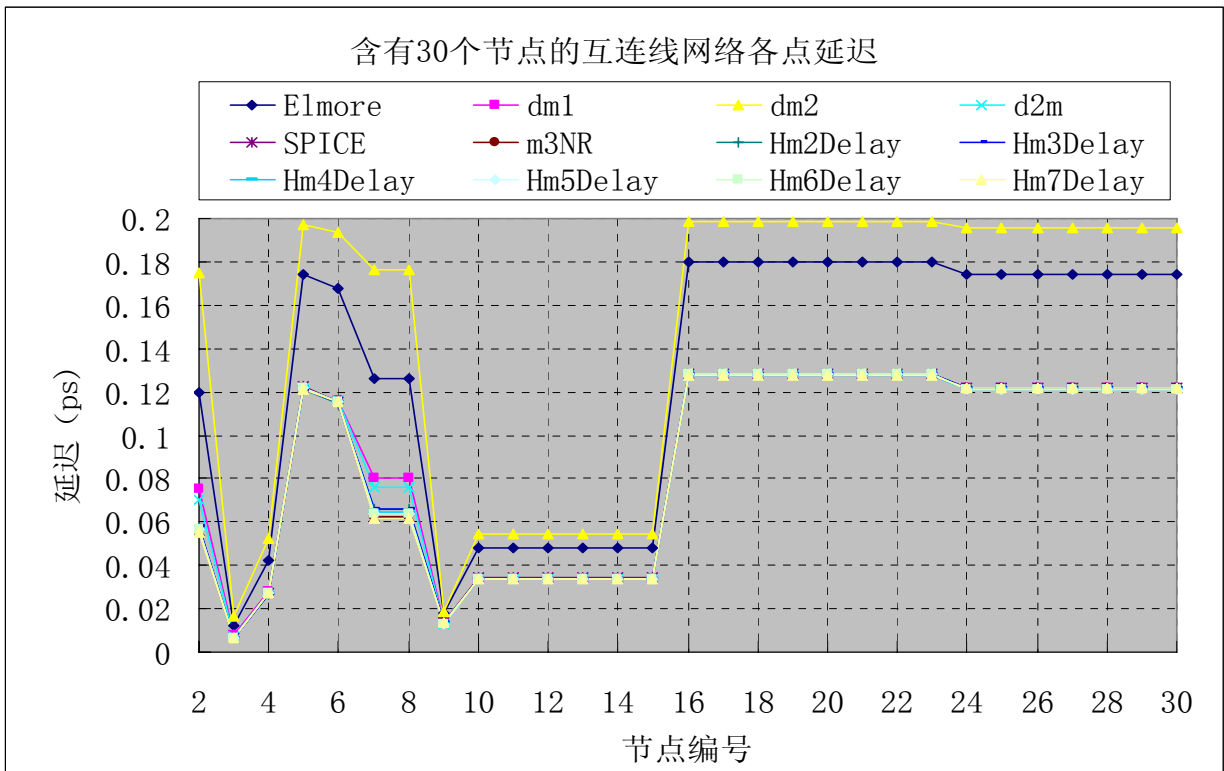


图4-7 存在分支的 30 节点互连线网络各点延迟比较

Fig.4-7 Delay of interconnect network with fanouts including 30 nodes

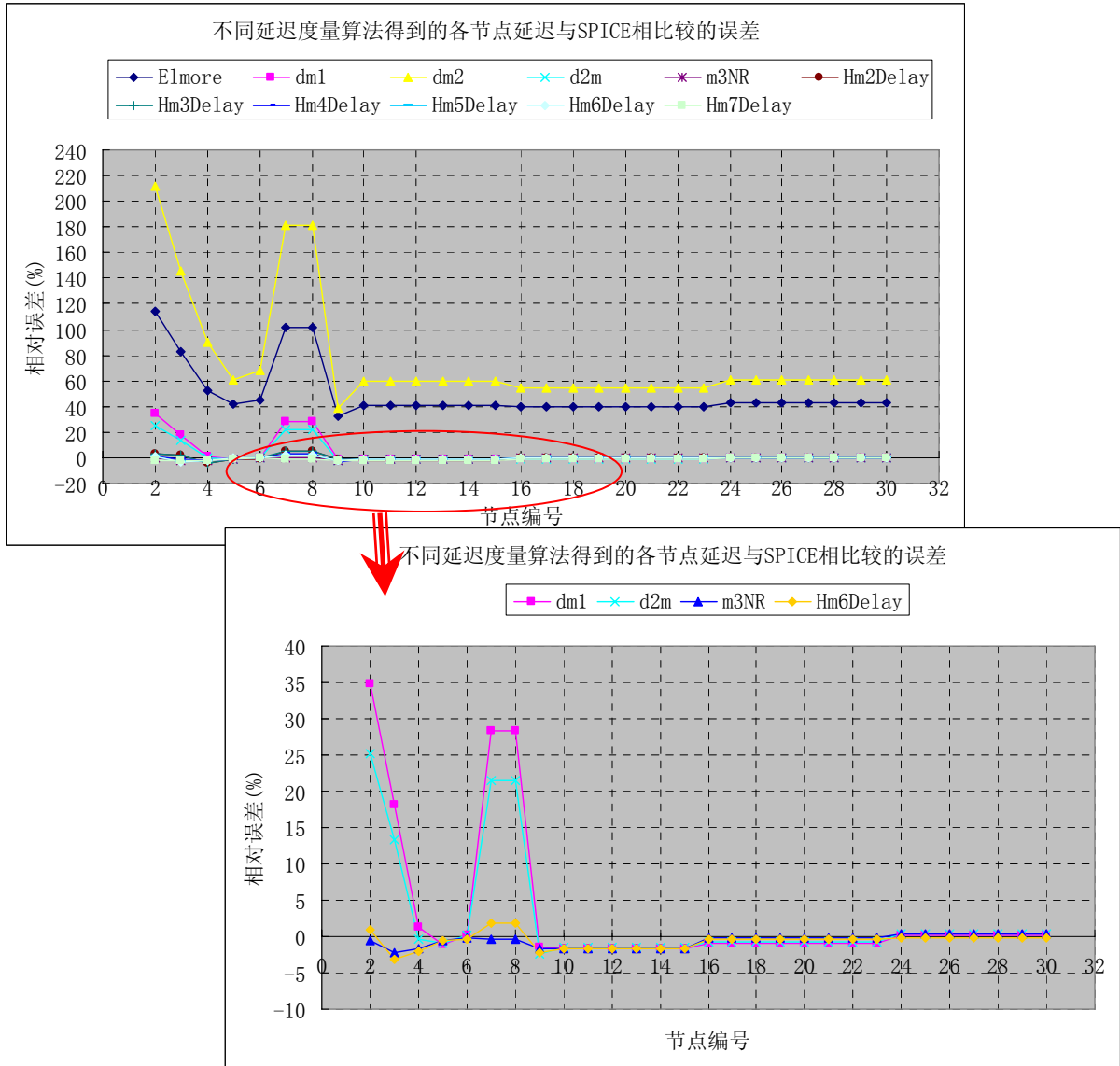


图4-8 30 节点互连线网络各点延迟的相对误差

Fig.4-8 Delay of interconnect network including 30 nodes

表4-3 30 节点互连线网络延迟的相对误差统计

Table4-3

相对误差 (%)	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	49.65	4.55	75.66	3.56	0.73	
标准差	20.74	9.43	43.27	6.92	0.67	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	1.34	1.33	1.01	0.94	0.95	1.05
标准差	1.46	1.47	0.86	0.81	0.80	0.68

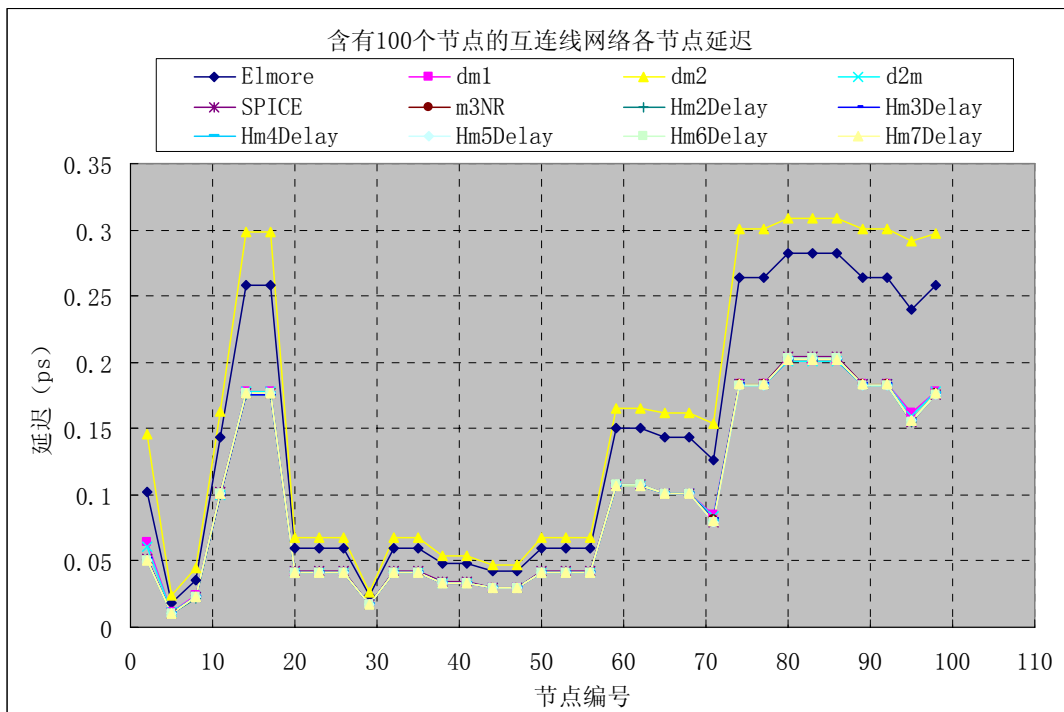


图4-9 存在分支的 100 节点互连线网络延迟比较

Fig.4-9 Delay of interconnect network with fanouts including 100 nodes

表4-4 100 节点互连线网络延迟的相对误差统计

Table4-4

相对误差 (%)	Elmore	dm1	dm2	d2m	m3NR	
均值 (%)	45.90	2.23	68.74	1.72	0.74	

标准差	11.31	4.54	25.15	3.07	0.68	
	Hm2Delay	Hm3Delay	Hm4Delay	Hm5Delay	Hm6Delay	Hm7Delay
均值 (%)	0.95	0.85	0.78	0.70	0.71	0.96
标准差	0.75	0.65	0.51	0.55	0.55	0.77

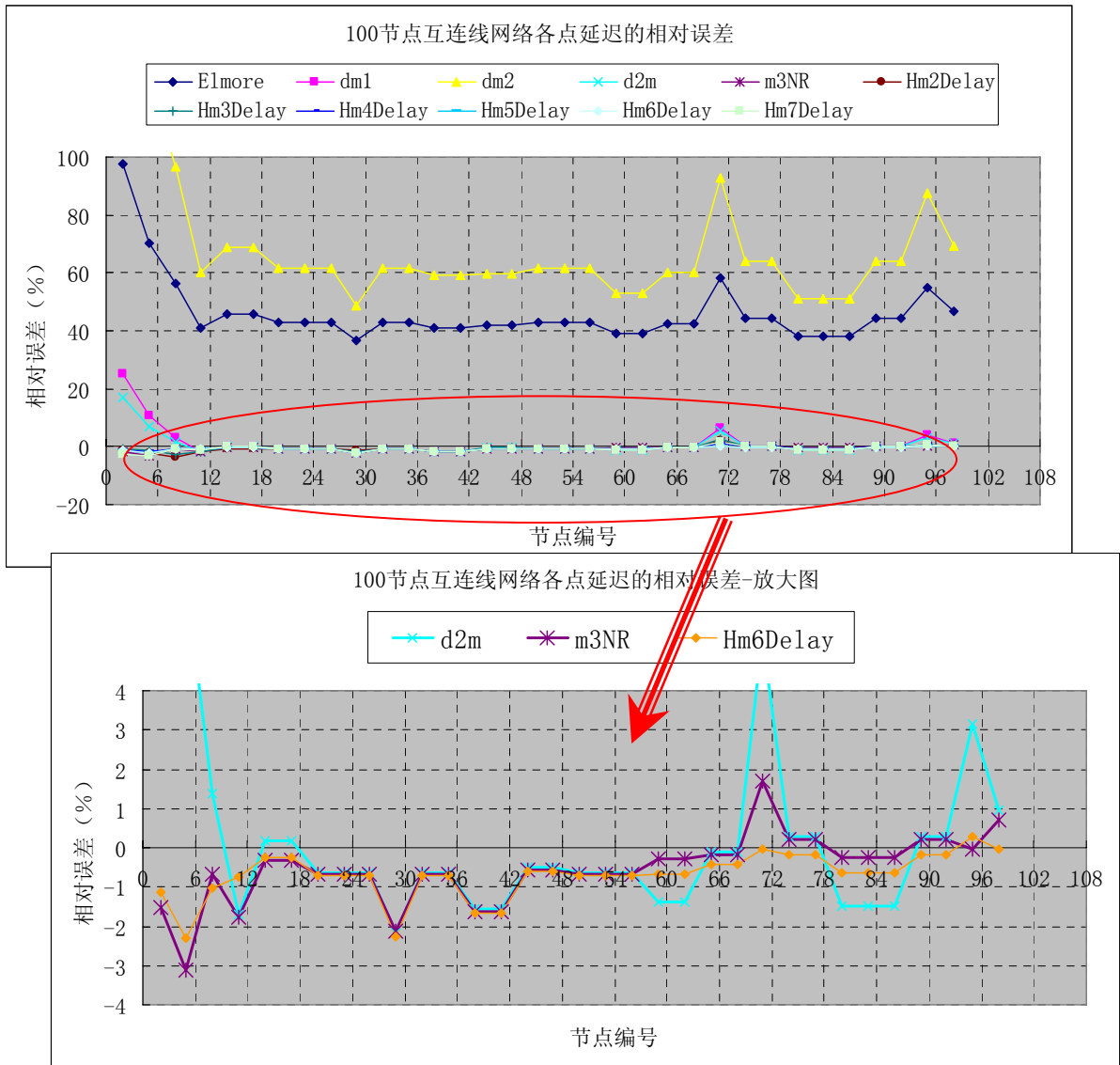


图4-10 100个节点互连线网络不同延迟度量方法所得各点延迟的相对误差
 Fig.4-10 Relative Error of Delay at each node for interconnect network with 100 nodes

由图 4-7 和图 4-9 可知, 对于互连线网络各节点的延迟估算, 与单条互连线不同的是, 各种延迟度量算法没有产生负值的情况。Elmore 延迟和 dm2 延迟仍然是估算误差最大的两种方法, 其他方法的相对误差均比较小。例如表 4-3 和表 4-4 给出的误差平均值数据显示, m3NR 算法与 Hm5Delay 和 Hm6Delay 的误差非常相近, 都是非常好的延迟估算方法; d2m 算法次之, 但其相对误差范围仍然可以接受。这样的结果也是与理论分析一致的。D2m 算法只采用了一至两阶矩进行计算, 而 m3NR 还采用了三阶矩, 因此理应具有更好的精度。Hm5Delay 和 Hm6Delay 算法分别采用最高至五阶和六阶的矩进行计算, 因此也具有更高的精度。

总的来说, DM2 算法的延迟估算误差最大, Elmore 延迟度量次之, 它们与 SPICE 仿真结果均差别非常大; D2M 是适用于所有电路结构, 对所有节点的延迟近似程度都较好的算法; DM1 算法与 D2M 算法的结果相近, D2M 的结果稍优于 DM1 算法, m3NR 和 Hm*Delay 系列算法对互连线网络的电路延迟近似程度非常好, 对远端点的延迟近似非常好, 但在近源点可能产生负的估算值, 不够稳定。因此, 在分析单条互连线网络时, 我们实用 D2m 算法, 而在计算互连线网络的节点延迟时, 我们使用 m3NR 算法或者 Hm5Delay 算法。

4.2 算法效率测试

正如上一章中我们所介绍的, 符号化方法使得整个算法的计算效率大大提高。我们只需对输入的互连线网表结构解析一次建立起相应的抽象结构, 即可将电路的关系存储在计算机内部。由于在计算同阶电路矩时, 每一节点的电路矩依赖于前序节点的电路矩, 符号化矩算法使得任意阶矩都可根据已计算过的低阶矩结果通过增量部分的计算得到, 避免了大量的重复计算。而求解延迟的二阶近似模型时, 需要求解延迟的均值及一阶和二阶项前的敏感度系数, 因此需要计算大量的延迟信息。由于对延迟概率分布的求解避免了使用蒙特卡罗方法, 通过特征函数及傅立叶变换运算几乎不消耗任何计算时间, 因此, 下面我们给出对求解延迟二阶近似模型效率的测试。

影响互连线延迟的因素有互连线的长度, 多端互连线网络树结构, 输入信号波形,

以及器件的驱动能力等。在此，我们假设输入信号波形均为理想阶跃波形，不考虑期间的驱动能力，针对不同的互连线结构，随机产生大量含有不同节点数的电路，于 Intel 1.83GHz，2G 内存的双核计算机平台上进行测试。

4.2.1 单条互连线 RC 电路

首先，我们针对单条互连线结构，随机产生大量含有不同节点数的电路，并给出各测试电路中最后一个节点延迟二阶模型的求解时间。这是因为在计算同阶电路矩时，每一节点的电路矩均依赖于前序节点的电路矩，因此最后一个节点电路矩的计算只有在所有节点的电路矩得到后才能进行，因而计算其延迟所需时间的最长。为了测试该算法所能处理的最大电路规模，我们总是选定电路中的最后一个节点进行延迟二阶模型的求解。另外，虽然延迟度量矩阵只用到最高七阶矩的计算，但为了给出符号化算法计算电路矩的效率，我们还给出了求解最后一个节点的十阶矩，五十阶矩，一百阶矩及二百阶矩需要耗费的时间。

表 4-5 中第一栏“输入网表总节点数”首先给出了输入测试随机电路所包含的节点数，这一参数反应了电路的规模。“测试节点编号”一栏给出了所测试节点的编号，由于电路的节点编号从数值 0 开始，因此该栏内容均表示电路中的最后一个节点，即距离电压源最远的节点。第三栏“解析网表时间”包含了解析输入网表及建立矩决策树结构所需的时间。“计算延迟二阶模型所需时间”一栏则给出了分别采用符号化矩方法和一般的数值计算矩方法所耗费的时间。表 4-6 中则给出了各测试节点上的不同高阶矩求解时间比较。表中的 UNTEST 表示因为其测试时间过长或不可行而没有对该项内容进行测试。

表4-5 单条互连线末节点延迟二阶模型求解时间比较

Table4-5

输入网表 总节点数	测试节点 编号	解析网表 时间(s)	计算延迟二阶模型所需时间(s)	
			符号化矩方法	数值方法

11	10	0	0	0
31	30	0.015	0	0.063
51	50	0.016	0	0.578
101	100	0.016	0.015	9.172
201	200	0.015	0.015	150.578
501	500	0.047	0.094	UNTEST
1,001	1000	0.094	0.406	UNTEST
2,001	2000	0.219	1.625	UNTEST
5,001	5000	0.86	10.171	UNTEST
10,001	10000	2.844	40.594	UNTEST

表4-6 单条互连线末节点延迟二阶模型求解时间比较

Table4-6

测试节点 编号	计算时间(s)						
	一阶矩	二阶矩	三阶矩	十阶矩	五十阶矩	一百阶矩	两百阶矩
10	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0
100	0	0	0	0.015	0.046	0	0
200	0	0	0	0	0.11	0	0
500	0	0.016	0.016	0.062	0.687	0.969	1.5
1000	0.016	0.047	0.063	0.219	2.344	3.453	5.656
2000	0.094	0.187	0.266	0.906	10.125	14.577	23.452
5000	0.563	1.125	1.657	5.547	64.485	92.454	150.937
10000	2.219	4.453	6.672	22.328	223.188	365.251	597.97

由上述两表中的测试结果可以看出，符号化矩算法使得矩的计算效率大大提高。十阶以下的电路矩的计算几乎不消耗任何时间，在含有 10000 个节点的电路中，十阶矩的计算也仅需 22 秒。另外，在含有 2000 个节点的单条互连线结构中，两百阶矩的计算仅需 23 秒左右的时间。不过其随着电路规模的增大，可以发现，高阶矩的计算时间增长较快。例如含有 5000 个节点的电路二百阶矩的计算需要 150 秒的时间，而对含有 10000 个节点的电路，更是达到了不可忍受的 600 秒左右的时间。但是，由于在模型降阶算法中，一般希望得到降阶后的电路在几十阶的量级，因此并不需要两百阶矩的计算。

相对低阶矩的计算时间而言，我们发现，解析网表占用了总时间较大的部分，这是

使用符号化矩算法所付出的代价。然而由于我们对每个输入电路只需进行一次读入和解析的过程，因此该部分所消耗的时间可以忽略。

由表 4-5 中的延迟二阶模型求解时间的比较，我们可以完全认识到符号化矩算法所带来的优势。基于非符号化方法的延迟二阶模型求解只能处理到含有 200 个节点的单条互连线结构，且需要漫长的 150 秒时间，而对节点数目更多的电路，由于耗费时间过长，无法进行测试。然而，基于符号化矩算法的延迟二阶模型求解可以轻松的处理到含有 10000 个节点以上的单条互连线结构，只需耗费 40 秒。且在节点数小于 1000 的结构中，其求解时间均远远小于 1 秒，可以“瞬间”得到延迟结果。

4.2.2 RC 树状互连线网络

针对存在分支的互连线电路，我们也随机产生了大量的电路，进行了同样的测试。由于随机电路产生时，从电源开始，电路节点的编号从 0 开始增长，因此编号越大的节点，距离电源越远，延迟二阶模型计算所需的时间也越长，因此我们同样取编号最大的节点进行测试。另外，随机电路的分支节点数及所有分支节点上平均存在的分支数也在表格中给出。对延迟二阶模型和高阶矩的测试结果分别如表 4-7 和表 4-8 所示。

表4-7 互连线网络末节点延迟二阶模型求解时间比较

Table4-7

输入网表 总节点数	分支 节点数	叶节 点数	分支节 点平均 扇出	解析网 表时间 (s)	目标测试 节点	计算延迟二阶模型所 需时间(s)	
						符号化矩 方法	数值方法
10	1	9	9	0	10	0	0
30	6	24	4.83	0	30	0	0
50	12	38	4.08	0	50	0	0
100	24	76	4.125	0.016	100	0	0
200	43	157	4.63	0.016	200	0	0.078
500	99	401	5.04	0.031	500	0.015	0.203
1000	204	796	4.9	0.094	1000	0.015	0.719
2000	400	1600	5	0.219	2000	0.016	2.5

5000	1016	3984	4.92	0.797	5000	0.032	28.031
10,000	2031	7969	4.92	2.578	10000	0.078	105.25
20,000	4007	15993	4.99	8.875	20000	0.172	UNTEST
50,000	9990	40010	5	51.718	50000	0.516	UNTEST
100,000	20050	79950	4.99	205.172	100000	0.906	UNTEST
200,000	39934	160066	5.01	813.109	200000	2.031	UNTEST
300,000	60067	239933	4.99	1579.204	300000	3.109	UNTEST

表4-8 互连线网络末节点延迟二阶模型求解时间比较

Table4-8

目标 测试 节点	计算时间(s)						
	一阶矩	二阶矩	三阶矩	十阶矩	五十阶矩	一百阶矩	两百阶矩
10	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0
200	0	0	0	0	0.015	0.015	0.015
500	0	0	0	0	0.016	0.016	0.031
1000	0	0	0	0	0.016	0.031	0.047
2000	0	0	0	0	0.032	0.047	0.094
5000	0	0	0	0	0.14	0.203	0.328
10,000	0	0	0	0.031	0.25	0.375	0.625
20,000	0.016	0.016	0.016	0.063	0.61	0.875	1.438
50,000	0.031	0.047	0.078	0.203	1.672	2.641	4.578
100,000	0.031	0.063	0.094	0.328	3.5	5.172	8.516
200,000	0.078	0.156	0.234	0.781	8.203	11.984	19.687
300,000	0.125	0.234	0.359	1.078	UNTEST	UNTEST	UNTEST

上述表格中的数据表明，与上一节中的单条互连线测试数据相比较，基于符号化矩的延迟二阶模型求解可处理更大规模的互连线分支网络电路。例如，含有三十万个节点的测试电路，该方法也仅需短短的3秒时间就可以计算得到节点上的延迟二阶模型，而相形见绌的数值方法，仅能处理少于10000个节点的电路。实际上，符号化方法可以处理多于三十万个节点的电路，然而由于程序网表解析能力的限制，无法测试。含有二十万个节点和三十万个节点的输入网表解析需要超过800秒以上的时间。同样，对高阶矩

的测试也主要受此限制。如表 4-8 中所示，含有 20000 个节点的互连线网表，其节点的二百阶矩计算也仅需 19 秒，但其网表解析过程时间过长。

4.2.3 小结

综合比较两种电路测试的结果，我们发现，互连线的深度是影响算法效率的关键。对单条互连线电路而言，其节点数即为互连线的深度，该算法仅能处理最大深度为 10000 左右的电路规模。而存在分支的互连线电路中，其深度取决于距离电源最远的节点。通常来说，由于存在许多分支，节点的深度远远小于电路中的总节点数，因此该算法可以处理含有三十万个节点以上的电路。

因此，与电路规模相比，即电路中的总节点数，电路中节点的最大深度才是制约算法效率更关键的因素。由于通常的电路是存在非常多的分支的，很少会出现深度为 10000 以上的节点，因此基于符号化方法的延迟二阶模型求解算法可以处理实际出现的几乎任何规模的电路，这正是该算法的一大优势。

4.3 电路测试结果及分析

在测试了算法的效率之后，本节给出基于符号化方法的互连线统计时序分析算法的测试结果。

4.3.1 一般结构的互连线延迟测量

根据 ITRS^[1]给出的工艺参数，我们取 0.11 μm 工艺下互连线版图的物理参数信息如下：电阻率 $\rho = 2.7 \times 10^{-8} \Omega \cdot \text{m}$ ，每段互连线长度为 $L = 1 \times 10^{-5} \text{ m}$ ，宽度均值为 $W = 0.28 \times 10^{-6} \text{ m}$ ，高度均值为 $H = 0.32 \times 10^{-6} \text{ m}$ 。在本节中，我们暂时仅假设互连线的宽度和高度两个参数为存在偏差的来源，其 3σ 对应的最大偏差比例均为 35%，绝缘层厚度 $t_{\text{ox}} = 0.5 \times 10^{-6} \text{ m}$ ，相对介电常数为 $\varepsilon_r = 3.9$ ，真空介电常数为 $\varepsilon_0 = 8.85418 \times 10^{-12} \text{ F} \cdot \text{m}^{-1}$ 。

我们将随机产生的输入互连线结构给出节点延迟的概率密度分布结果，并将其与蒙

特卡洛采样方法统计得到的延迟概率密度分布进行比较，分析其误差。

首先以一个具有四条支路，31 个节点的小规模互连线RC树状电路为例，给出其电路近源点，中间点及远端点上的算法测试结果及与蒙特卡罗仿真结果的比较。该电路的电路结构如图 4-11所示。

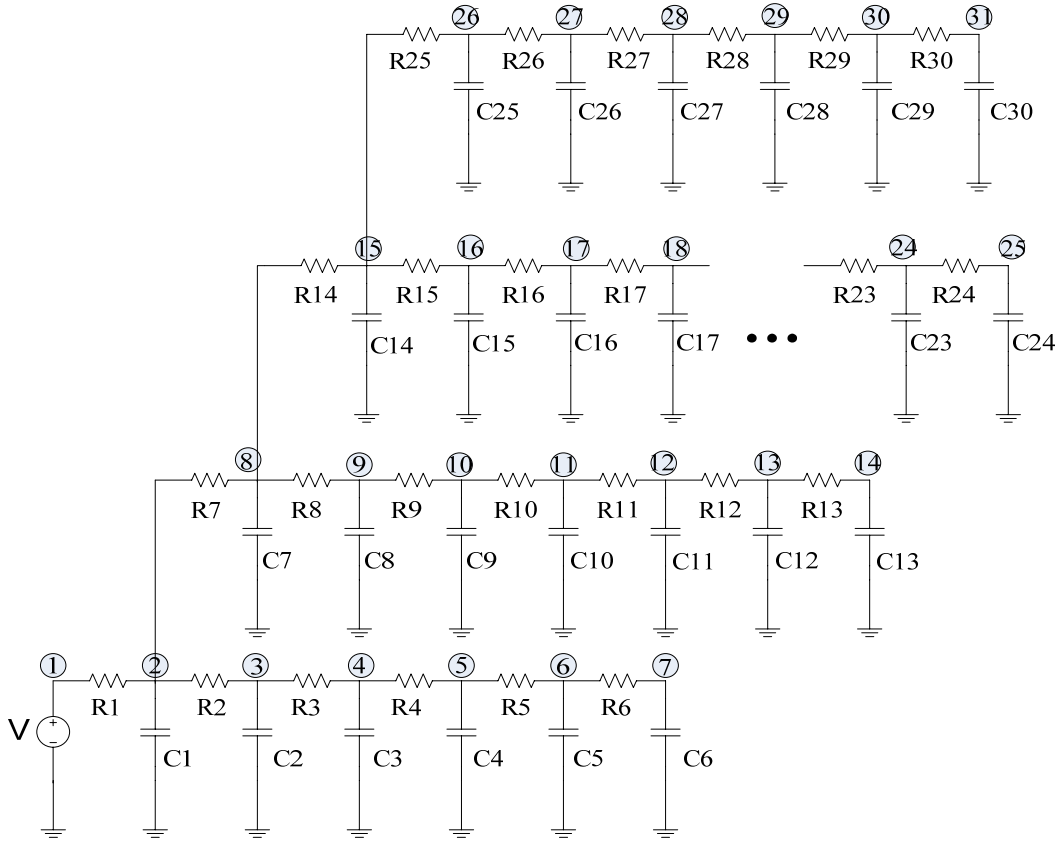
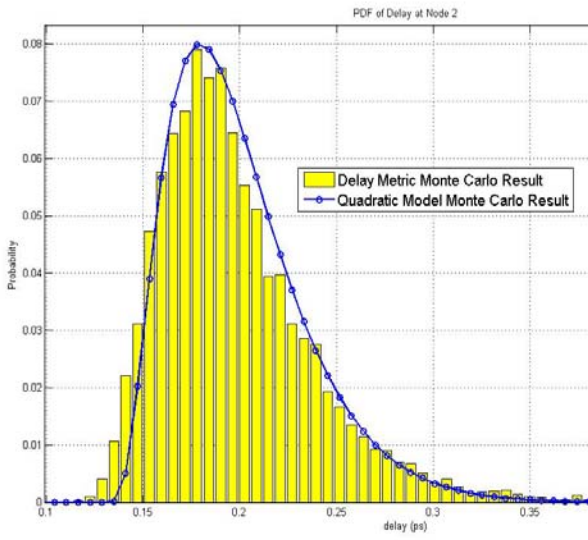


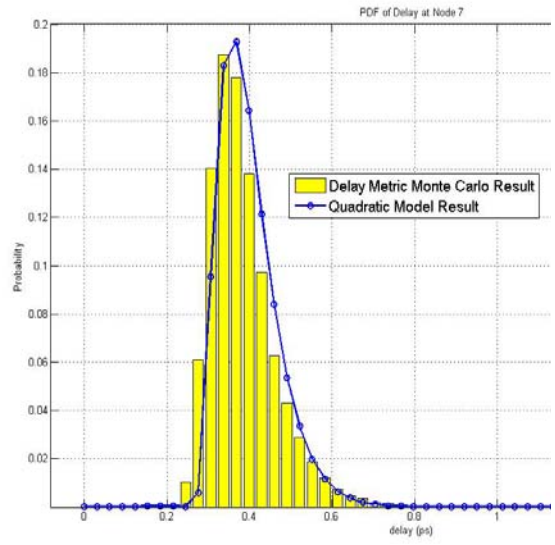
图4-11 四支路互连线电路

Fig.4-11 Interconnect with four chains

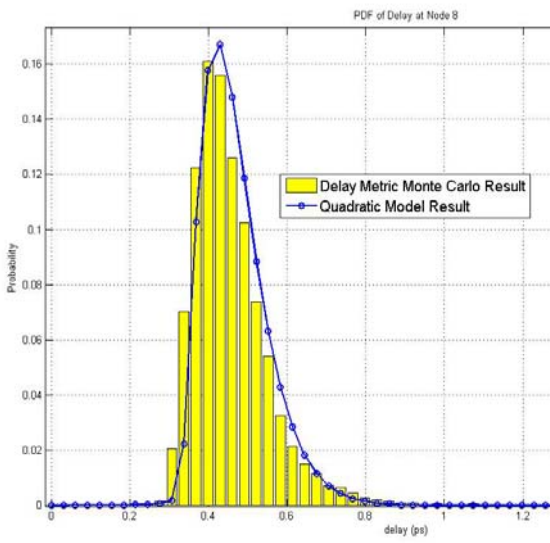
选取该电路上几个具有代表性的节点，它们的延迟概率分布测试结果及蒙特卡罗仿真攻击结果如图 4-12所示。图中蓝色带圈线表示本文二阶模型统计分析算法计算得到的结果，黄色条形图表示蒙特卡罗采样方法统计得到的概率密度分布结果。



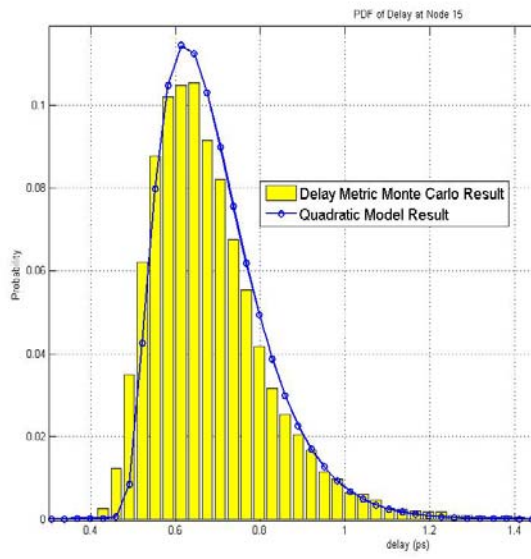
(a) 节点 2



(b) 节点 7



(c) 节点 8



(d) 节点 15

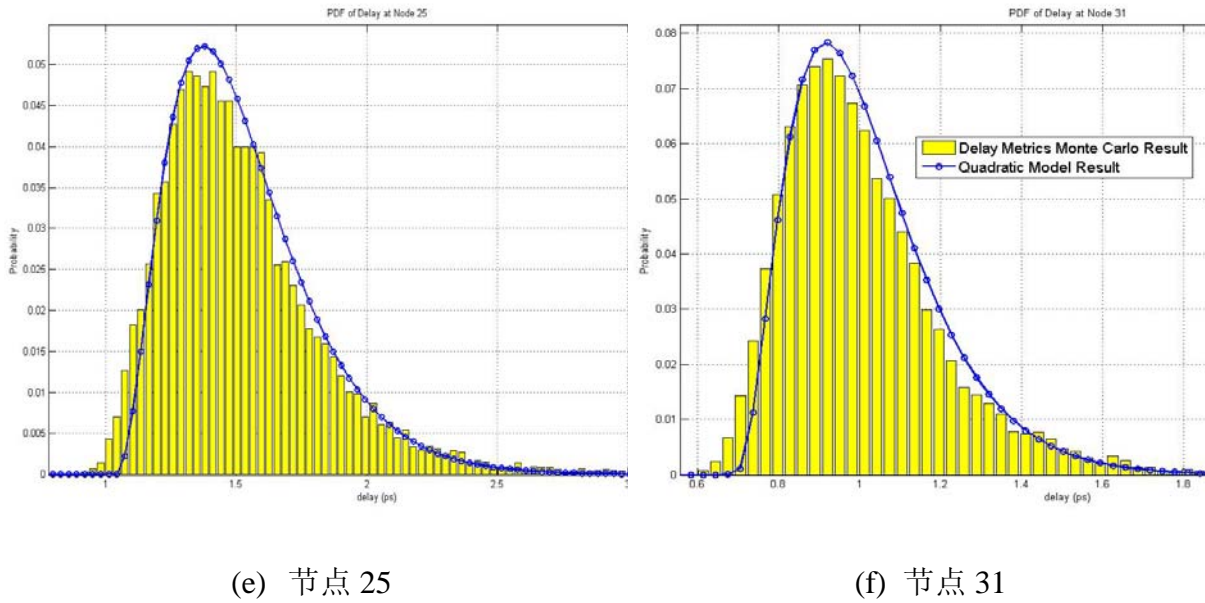


图4-12 各节点延迟概率分布结果

Fig.4-12 Delay PDF of specified nodes

由上述结果可以看出，在假定互连线宽度及高度偏差为高斯分布的前提下，通过蒙特卡罗仿真得到的节点延迟不再是高斯分布，具有明显的不对称特性。峰的左边部分较短，而右边部分则有很长的尾巴。因此，我们不能简单的将互连线延迟建模为具有高斯分布的随机变量。

由两种方法结果的比较可以看出，本文对延迟的统计建模方法与蒙特卡罗结果吻合的很好。二阶模型很好的建模了互连线延迟关于互连线宽度、高度等物理参数的非线性关系，给出了与真实结果近似程度非常好的延迟概率密度分布结果。

延迟的概率密度分布可以被用来检查电路的时序性能是否满足要求，例如延迟落在某一范围内的概率是否达到了我们希望的标准，而通常这一范围选取均值左右 3σ 的区间。在这里，我们得到的概率密度分布已不是高斯分布，然而为了估算二阶模型的误差，我们仍然可以通过比较某一固定范围上其概率结果与蒙特卡罗结果的差别来给出大概的误差分析。因此，我们选取蒙特卡罗结果均值上下 3σ 的区间，求解该区间上两种方法得到的概率值，计算误差，得到各节点的误差信息如图 4-13所示。

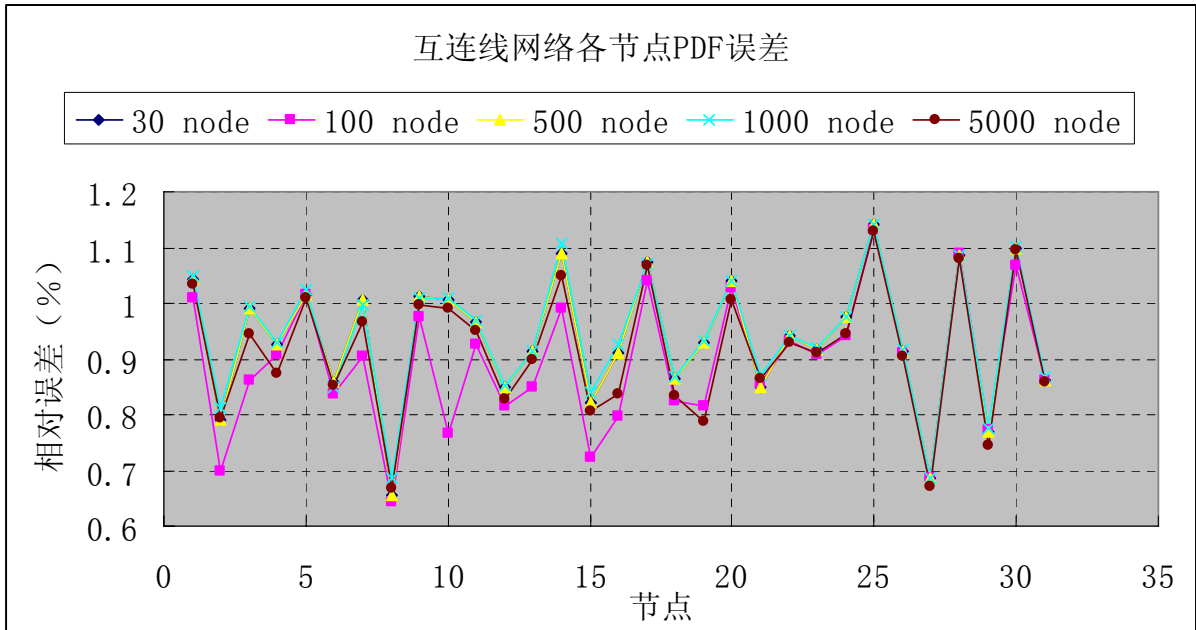


图4-13 互连线网络各节点延迟 PDF 的误差

Fig.4-13 Error of delay PDF at each node

由上图可见，各节点的误差基本处于一个稳定的范围内，不随节点离源点的不同距离而变化。进一步，我们给出对随机产生的含有不同节点的互连线网络中所求得节点延迟 PDF 误差的平均值。

表4-9 不同规模的互连线网络节点延迟 PDF 的平均误差

Table4-9

网表节点数	30	50	100	500	1000	5000
延迟 PDF 相对误差 (%)	0.902428252	0.718424	0.862845	0.939454	0.940716	0.914017

4.3.2 时钟树型电路的延迟测量

在大部分大规模集成电路中，各时序元件之间的数据传输是由一个控制信号同步控制的，这就是时钟信号。它是设计高速集成电路时必须重视的问题，处理好时钟信号可以大幅度提高电路性能。描述时钟信号质量的参数有：时钟信号的渡越时间，时钟信号的插入延迟等。

时钟信号的渡越时间是指时钟信号在时序元件的时钟端口的上升下降时间。时钟信号的插入延迟是指时钟信号的输入端到它控制的时序元件的延迟时间。要改进集成电路的性能，提高其工作频率，必须降低时钟信号的插入时间。这就需要分析时钟树的延迟。

时钟树型电路与我们之前讨论的电路不同的一点是，它通常采用 **sizing** 策略，即互连线的宽度随分支深度的不同而不同，因此在同一个电路结构中存在多个不同的宽度偏差来源。在本节中，我们探讨存在多个宽度偏差来源的时钟树电路。工艺参数值的设定与之前相同，但假设时钟树网络存在不同的宽度设计值，即离全局时钟源最近的采用最大宽度的互连线进行布线，若干段后，采用减半宽度的互连线进行布线，具有相同宽度的互连线段为一个阶段。例如， W_1 取 $0.56e-6$ 微米， W_2 取 $0.28e-6$ 微米， W_3 取 $0.14e-6$ 微米等，以此类推，宽度偏差变量的维数取决于时钟树型电路中不同的阶段数。

下图给出了我们随机产生的一个具有三个阶段的时钟树型电路示例。由于该电路是对称结构的，因此我们只需测试节点 2, 102, 302, 702 等即可大概得到电路延迟估算的整体印象。

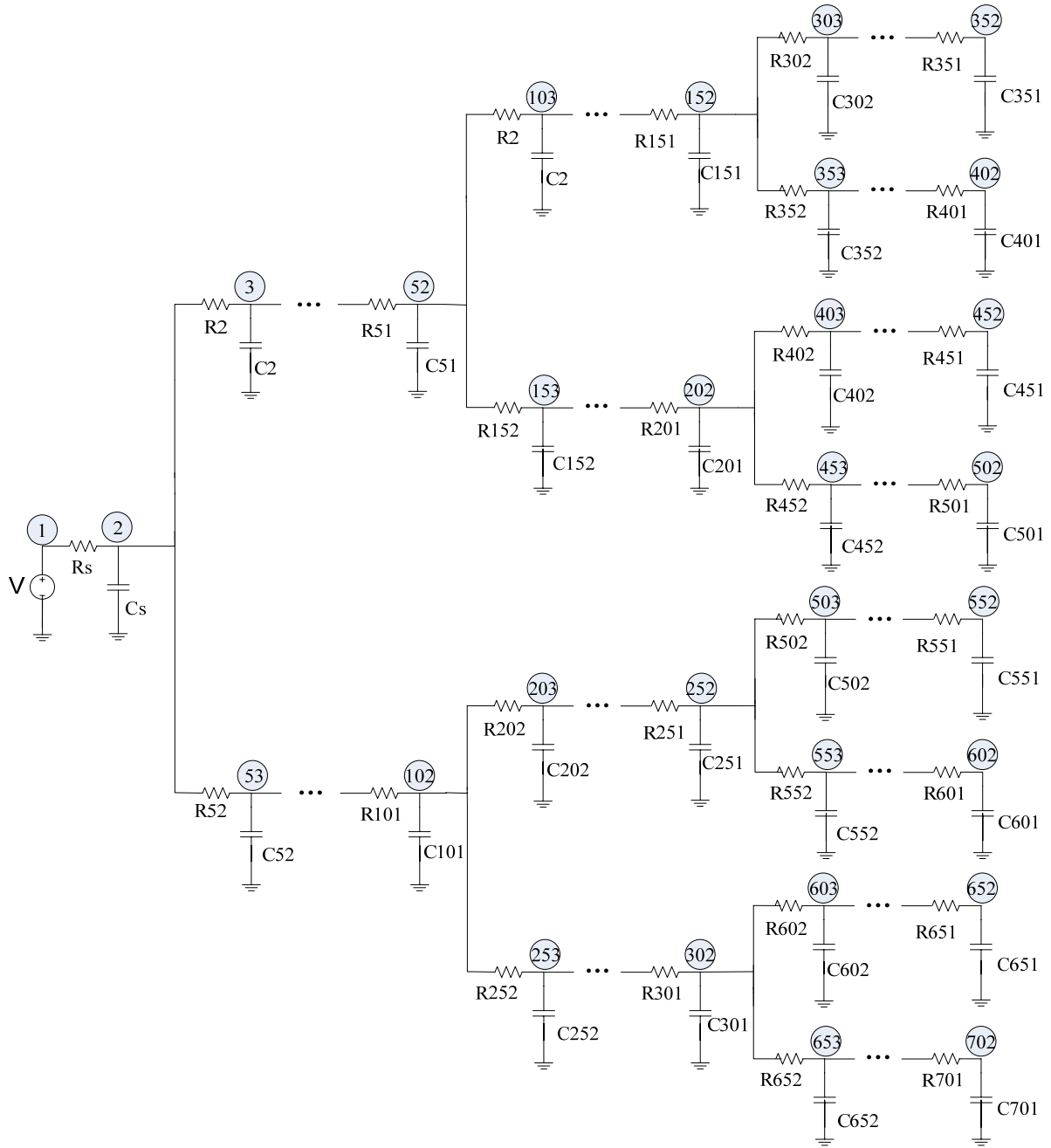


图4-14 三阶段时钟型电路

Fig.4-14 3-stage Clock-tree Circuit

上述电路仿真得到的节点延迟概率密度分布图如下：

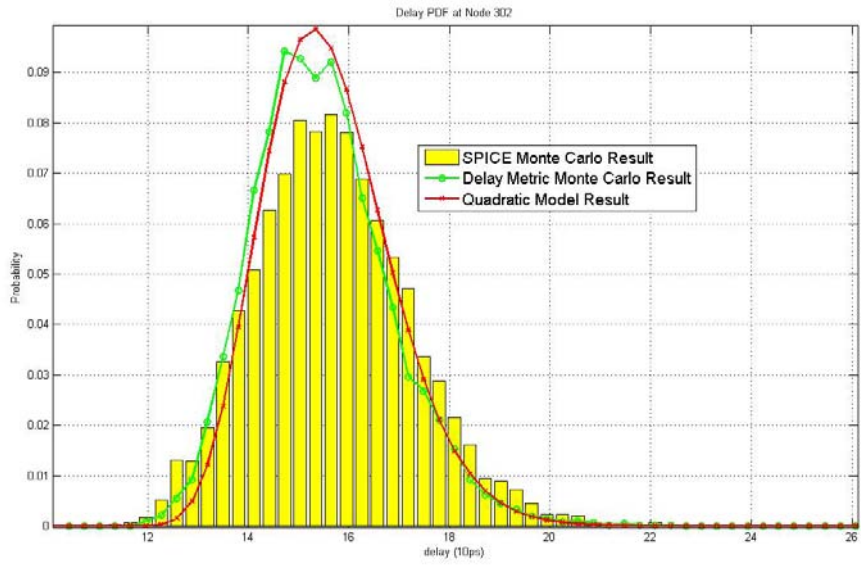


图4-15 三阶段时钟型电路节点 302 的概率密度分布

Fig.4-15 PDF at node 302 of a 3-stage Clock-tree Circuit

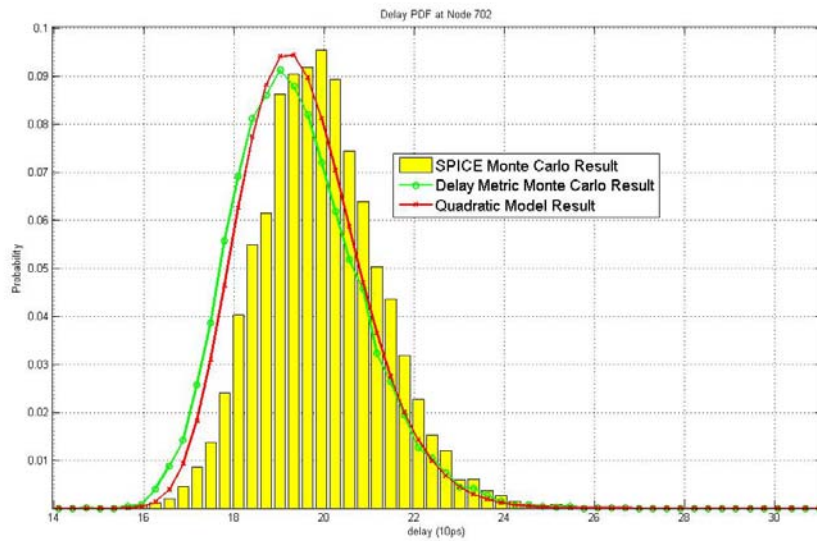


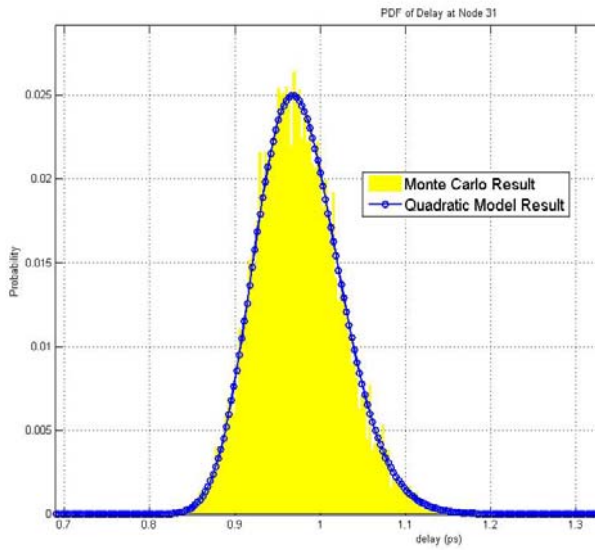
图4-16 三阶段时钟型电路节点 702 的概率密度分布

Fig.4-16 PDF at node 702 of a 3-stage Clock-tree Circuit

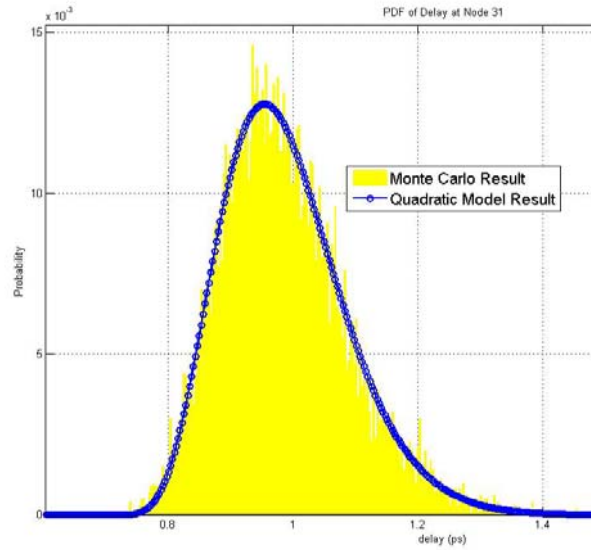
4.3.3 不同的最大偏差范围测试

上述电路的测试结果均是假设宽度和高度的最大偏差范围为 35% 而得到的，即 35% 对应随机变量概率密度分布的 3σ 处。在这一节，我们将探讨不同的偏差范围所造成的延迟概率密度分布结果的变化。

同样以上节中具有四个支路，30 个节点的电路为例进行阐述，分别假设宽度和高度的最大偏差范围为 10%，20%，35% 和 50%，计算得到的节点 25 处的延迟概率分布结果分别如下：



(a) 10% 最大偏差



(b) 20% 最大偏差

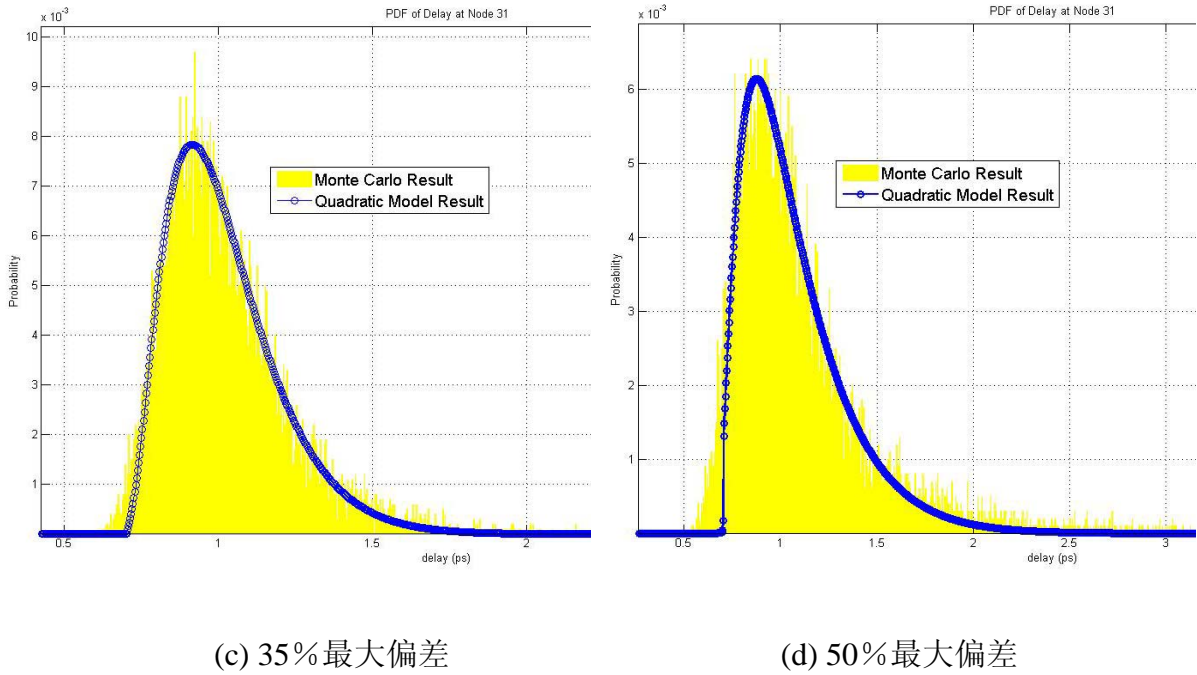


图4-17 不同最大偏差下节点 31 的延迟概率分布

Fig.4-17 Different Delay PDF at node 31 under different 3σ

由这四幅图可得到结论，最大偏差范围不但影响延迟概率密度分布的方差，即以平均值为中心扩散的程度，而且会影响延迟概率分布的对称性。偏差随机变量的变化范围越大，则延迟概率密度分布的不对称性越强。这表明，偏差越大，延迟与偏差之间的非线性关系所造成的影响越大，而假设延迟为高斯分布所引起的误差也就越大。

因此，在如今最大偏差已达到 35%~50%的情况下，忽略非线性关系所引起的误差将会非常大。因此，我们需要采用二阶模型，从而保证建模非线性部分所造成的影响。

4.3.4 偏差间的相关性影响测试

到现在为止，上面所进行的测试均假设宽度和高度偏差随机变量之间是相互独立的。然而，由第一章中对制造偏差的讨论我们知道，它们之间可能存在非常复杂的关系，从而导致偏差随机变量间存在一定的相关性。因此，本节我们将讨论相关性对结果的影响。假设宽度偏差与高度偏差之间的协方差为 0.5，蒙特卡罗采样方法与本文算法得到的结

果如下：

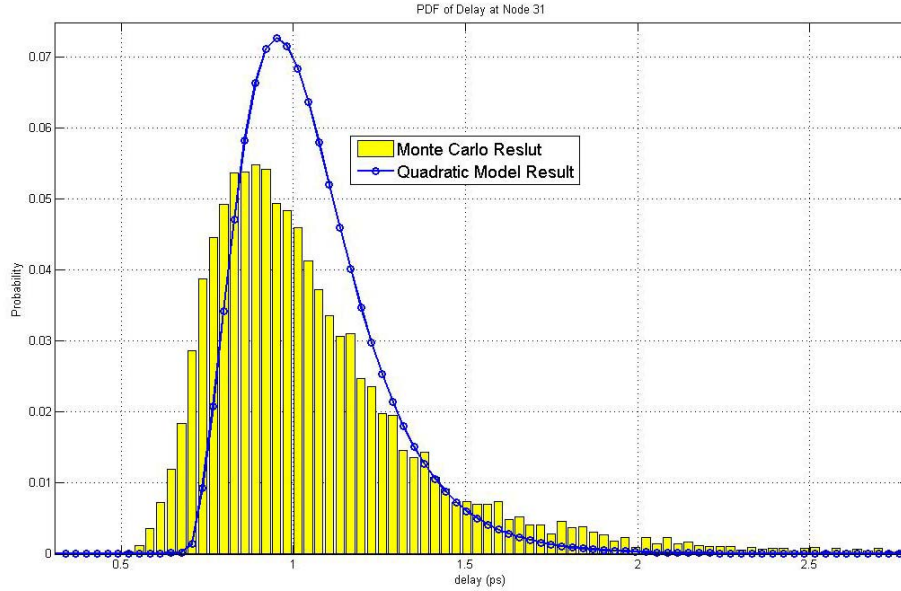


图4-18 协方差 0.5 下节点 31 的延迟概率分布

Fig.4-18 Delay PDF at node 31 under W,H covariance of 0.5

从图中可以发现，在宽度偏差与高度偏差间存在协方差 0.5 的条件下，蒙特卡罗方法与本文算法得到的结果差别很大，不再是如先前吻合很好的情况。这主要是因为，蒙特卡罗采样方法均假设随机变量之间是相互独立的，对其分别进行采样，因此得到的结果并不准确。而本文的算法可以处理随机变量间存在任意相关性的情况，给出的结果是准确的。两种方法结果的比较告诉我们，如果偏差随机变量之间存在较大的相关性，那么我们不能简单的忽略这种联系，而是必须进行正确的建模。本文的算法完全做到了这一点，因而具有更好的精度与更广泛的用途。

4.4 与已有方法的比较

在第二章中，我们介绍过两种已有的互连线统计时序分析算法，并着重介绍了 K.Agarwal 等提出的线性建模的统计时序分析方法和 J.Zeng 等提出的基于模型降阶算法的二阶统计时序分析方法。在本节，我们将本文提出的算法分别与这两种算法进行比较，

证明本文提出的新算法在精度以及计算效率方面的优势。

4.4.1 基于线性模型的统计时序分析方法

仍以具有四分支，31个节点的互联线结构为例，计算得到线性统计时序分析方法，本文二阶统计时序分析方法及蒙特卡罗的结果，如图4-19所示。图中，绿色是蒙特卡罗方法的仿真结果，红色带圈线本文所提出的基于二阶模型的算法结果，蓝色带星线是线性统计时序分析方法的结果。

显然，线性统计时序方法得到的延迟概率分布是完全对称的。这主要是由于K.Agarwal等假设分析互连线时序的前提是：互连线延迟的PDF为正态分布；物理参数均为正态分布，且不同物理参数间互相独立。因此，经过线性变换的互连线延迟仍然是正态分布，这与蒙特卡罗结果相比显然不一致。该方法的精度与本文算法相比，误差很大。

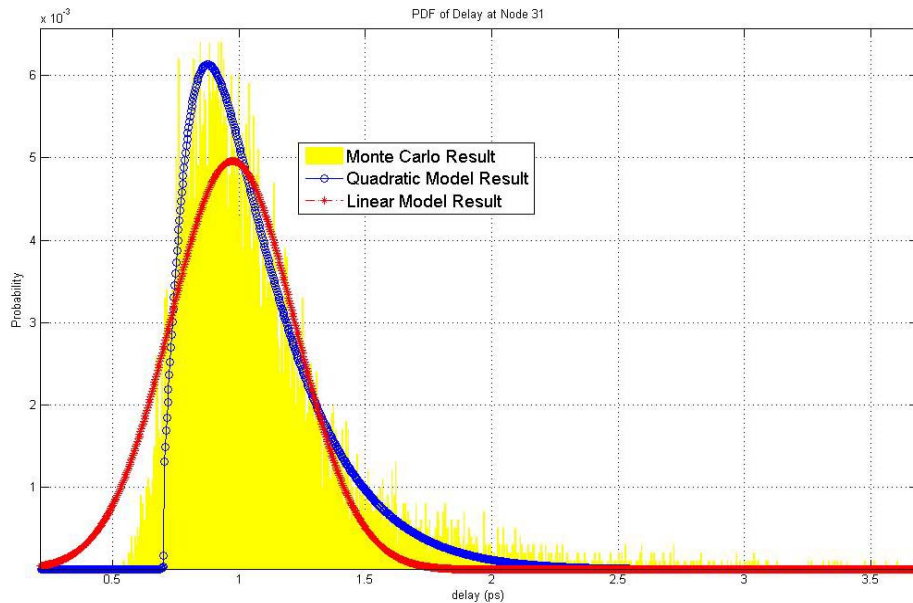


图4-19 线性方法与二阶近似模型比较

Fig.4-19 Comparison of Linear Model with Quadratic Model

此外,更重要的一点是,该线性统计时序分析算法使用手工推导的方法得到线性模型计算的表达式,无法应用于大规模电路。作者仅给出了含有 30 个节点的小规模电路的测试结果。而本文的方法由于采用符号化矩算法,无需手工推导,且能处理含有三十万个节点以上的大规模互连线电路,精度更高。

4.4.2 基于 AWE 的统计时序分析算法

与本文基于延迟度量矩阵方法不同,J.Zeng 等提出的互连线延迟二阶模型方法基于 AWE 算法。由于采用了更准确的模型降阶算法,该方法理论上可以给出更好的精度。然而,AWE 算法本身固有的稳定性问题可能会造成很大的数值计算误差。由于纳米工艺下互连线本身的物理参数的尺寸非常小,因此在不加入任何额外计算机数值误差处理方法时,AWE 所引入的数值计算误差是毁灭性的。这主要是因为 AWE 算法中对电路响应采用 pade 近似得到结果,因此,在电脑精度有限的基础上进行的 Pade 近似以及显式电路矩匹配造成了大量的计算误差。另外,该方法所需的计算量非常大。这主要是因为首先 AWE 算法本身需要很大的计算量,其次在进行统计时序分析时,每一步都需要进行二阶模型近似,增加了计算量,因此总的计算量非常大。

本文的算法与之相比,虽然精度上有些微的差异,但原理更加简单,流程简单,计算量非常低,并且算法的实现成本更低,无需实现 AWE 算法时所需的很多误差处理程序。

4.5 本章小结

本章介绍了基于符号化矩算法的互连线统计时序分析方法的具体实现及电路测试结果。通过大量随机产生的不同结构的互连线电路的测试,给出了算法效率与准确性的分析结果。

第五章 总结与展望

5.1 总结

本文提出的互连线延迟统计时序分析方法具有高效和准确的特点，可以方便快速的应用到增量时序分析和物理设计优化等芯片设计流程中。该算法首先将电气参数，如电容、电阻等，表示为物理参数偏差的函数；进而通过符号化矩算法得到矩关于物理参数偏差的函数；最终基于延迟度量算法得到互连线延迟关于物理参数偏差的闭合形式表达式。在此基础上采用二阶模型近似互连线延迟，从而通过其特征函数及傅立叶变换直接获得延迟的概率密度分布。与蒙特卡罗方法相比，该算法近似的误差在 1% 以内，且计算简单方便，只需进行一次延迟的二阶模型求解，具有很好的效率。

5.2 研究展望

本文中的算法采用互连线 RC 树状模型对互连线进行建模，只考虑了节点与地之间的寄生电容对延迟的影响，没有考虑存在耦合电容的情况。另外，没有考虑高速互连线中存在的电感效应，未来的算法研究可以拓展考虑这些因素。

另外，本文所介绍的互连线统计时序分析的符号化算法仅是对符号化分析方法应用到统计时序分析的初步探索。该算法中对延迟二阶近似模型所需的敏感度系数仍采用数值求导的方法进行计算。由于所建模互连线的微小尺寸，各参数值的数量级都非常小，因而这种数值求导的方法会带来一些微小的误差。因此，下一步，我们可以研究求解符号化导数的方法，以得到符号化的二阶模型，实现更高的精度。

参 考 文 献

- [1] 国际半导体技术蓝图, (International Technology Roadmap for Semiconductors, ITRS), <http://www.itrs.net/>
- [2] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations", IEEE Custom Integrated Circuits Conference (CICC), 2001, pp.223-228
- [3] D. Blauww, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical Timing Analysis: From Basic Principles to State of the Art", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No.4, pp.589-607, April 2008
- [4] V. Mehrotra, S. Nassif, D. Boning, J. Chung, "Modeling the Effects of Manufacturing Variation on High-Speed Microprocessor Interconnect Performance", IEEE International Electron Devices Meeting(IEDM) 1998, pp.767-770
- [5] Y. Liu, L. T. Pileggi, and A. J. Strojwas, "Model order-reduction of RC (L) interconnect including variational analysis", in Proc. Design Automation Conf., New Orleans, LA, 1999, pp. 201-206.
- [6] W. Dai, H. Ji, "Timing Analysis Taking into Account Interconnect Process Variation", IEEE International Workshop on Statistical Methodology, IEEE 2001, pp.51-53
- [7] K. Agarwal, D. Sylvester, D. Blaauw, F. LIU, S. Nassif, S. Vrudhula, "Variational delay metrics for interconnect timing analysis", DAC'04, pp.381-384, June 2004
- [8] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, "A Multiparameter Moment-Matching Model-Reduction Approach for Generating Geometrically Parameterized Interconnect", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.23, No. 5, pp. 678-693, May 2004
- [9] James D. Ma and Rob A. Rutenbar. "Fast Interval-Valued Statistical Modeling of Interconnect and Effective Capacitance," IEEE trans. Computer-Aided Design, vol. 25, NO. 4, APRIL 2006, pp. 710-724
- [10] J. Zeng, C. Chen, "Deep Submicron Interconnect Timing Model with Quadratic Random Variable Analysis", Design, Automation and Test in Europe, 2008. DATE '08, pp.

- 1091-1094
- [11] M. Masoumi, N. Masoumi, and A. Javanpak, "A New and Efficient Approach for Estimating the Time-Domain Response of Capacitive Coupled Distributed RC Interconnects", 12th IEEE Workshop on Signal Propagation on Interconnects(SPI2008), pp. 1-4
- [12] G. Shi, Bo Hu, C.-J. Richard Shi. "On Symbolic Model Order Reduction", IEEE. Trans. on computer-aided design of integrated circuits and systems, July 2006, pp. 1257-1272
- [13] L. T. Pillage, and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", IEEE Trans. on Computer-Aided Design, Vol. 9, No. 4, pp. 352-366, April 1990
- [14] Curtis L. Ratzlaff and Lawrence T. Pillage, "RICE: Rapid Interconnect Circuit Evaluation Using AWE", IEEE Trans. Computer-Aided Design, vol. 13, No. 6, pp 763-776, June 1994
- [15] P. Feldman and R. W. Freund, "Efficient Linear Circuit Analysis by Pade Approximation Via the Lanczos Process", IEEE Trans. Computer-Aided Design, vol. 14, no. 5, pp. 639-649, May 1995
- [16] A. Odabasioglu, M. Celik, and L. T. Pillage, "PRIMA: Passive reduced-order interconnect macro modeling algorithm," IEEE Trans. Computer-Aided Design, vol. 17, No. 8, pp. 645-654, Aug.1998.
- [17] Freund, R.W, "SPRIM: structure-preserving reduced-order interconnect macromodeling," ICCAD2004. IEEE/ACM International Conference on Computer Aided Design, 2004, pp. 80 – 87
- [18] A. B. Kahng, S. Muddu, "Accurate analytical delay models for VLSI interconnects", UCLA CS Dept. TR-950034,1995
- [19] B. Tutuianu, F. Dartu, L. Pilleggi, "Explicit RC circuit delay approximation based on the first three moments of the impulse response", in Proc. IEEE/ACM Design Automation Conf. 1996, pp611-616
- [20] C. Alpert, A. Devgan, and C. Kashyap, "A two moment RC delay metric for performance optimization", ISPD, pp.69-74, 2000
- [21] F. Liu, C. Kashyap, and C. Alpert, "A delay metric for RC circuits based on the Weibull distribution", ICCAD'02, pp. 620-624

- [22]K. Agarwal, D. Sylvester, and D. Blaauw, “Simple Metrics for Slew Rate of RC Circuits Based on Two Circuit Moments”, *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on (DAC2004), pp. 1346-1354
- [23]苏舟, “一种新的互联延迟度量”, *半导体技术*, vol.29 No.8, pp69-71, 2004年8月
- [24]Y. I. Ismail, C. S. Amin, “Computation of Signal-Threshold Crossing Times Directly from Higher Order Moments”, *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol.23, No. 7, July 2004, pp. 1264-1276
- [25]J. Rubenstein, P. Penfield, and M. A. Horowitz, “Signal delay in RC tree networks”, *IEEE Trans. Computer-Aided Design*, Vol.CAD-2, pp 202-211, July 1983
- [26]T. Lin, E. Acar, and L. Pileggi, “h-gamma: An RC delay metric based on a gamma distribution approximation to the homogeneous response,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 19-25
- [27]R. Kay and L. Pileggi, “PRIMO: Probability interpretation of moments for delay calculation,” in *Proc. IEEE/ACM Design Automation Conference*, June 1998, pp. 463-468
- [28]F. Liu, C. Kashyap, and C. J. Alpert, “A Delay Metric for RC Circuits based on the Weibull Distribution”, *Computer Aided Design, 2002. ICCAD 2002. IEEE/ACM International Conference on*, pp. 620-624
- [29]刘昆, 郑云, 黄道君, 侯劲松, 张炜, “互连线延迟评估的概率解释算法”, *计算机辅助工程*, 2004年1月, pp44-49
- [30]M. Kuhlmann, ‘Exact and Efficient Crosstalk Estimation’, *IEEE Trans. Computer Aided Design of Integrated Circuits*, vol.20, July 2001, pp. 858-865
- [31]J. Cong, “Wire Width Planning for Interconnect Performance Optimization”, *IEEE Trans. Computer Aided Design of Integrated Circuits*, vol.21, March. 2002, pp.319-329
- [32]S. B. Akers, “Binary Decision Diagrams”, *IEEE Trans. Comput.*, Vol. C-27, pp. 509-516, June 1976
- [33]R. E. Bryant, “Graph-based algorithms for Boolean function manipulation”, *IEEE Trans. Comput.*, vol. C-37, pp.677-691, August 1986
- [34]L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C. Chen, “Correlation-Preserved Non-Gaussian Statistical Timing Analysis with Quadratic Timing Model”, *DAC 2005*, pp. 83-88

- [35]F. Hasani, N. Masoumi, “Crosstalk and Delay Optimization Techniques for Nano Scale Interconnects”, *Design & Technology of Integrated Systems in Nanoscale Era*, 2007. DTIS. International Conference on, pp. 159-163
- [36]J. Liou, K. Cheng, S. Kundu, and A. Krstic, “Fast Statistical Timing Analysis by probabilistic event propagation”, in *Proc. DAC*, 2001, pp. 661-666
- [37]J. Liou, A. Krstic, L. Wang and K. Cheng, “False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation”, in *Proc. DAC*, 2002, pp. 219-224
- [38]S. Naidu, “Timing yield calculation using an impulse-train approach”, in *Proc. ASP-DAC*, 2002, pp. 219-224
- [39]J. Stolfi and L. H. de Figueiredo, “Self-Validated Numerical Methods and Applications”. *Brazilian Mathematics Colloquium Monograph*, IMPA, Rio De Janeiro, Brazil, 1997
- [40]W. C. Elmore, “The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers”, *J. Applied Physics*, vol. 19, no. 1, Jan. 1948, pp. 55-63
- [41]G. Shi, “A Symbolic Moment Calculator for RLC Tree Circuits with Application”, submitted to *ISCAS*, 2009

攻读研究生学位期间已发表或录用的论文

- [1] 于雪红, “基于符号化算法的互连线统计时序分析”, 微电子学与计算机, 2009 年 9 月 (已录用)

致 谢

时光如梭，转眼间，两年半的研究生生涯即将结束。

首先要感谢我的父母和我的妹妹，在过去的两年中，您们默默给予我无微不至的关怀，不求回报；您们给予我充分的信任，让我自己选择自己的人生；你们一直都是我最坚强的后盾，给予我信心，赋予我战胜困难的力量。

其次，我要感谢我的硕士阶段指导老师施国勇教授。您渊博的学识，严谨的求学作风和踏实认真的态度让我受益匪浅。同时，还要感谢郝志刚博士以及我们 EDA 小组的各位成员，徐迪，刘安，孟晓旋，黄世杰，祝翔宇，曾媚，王婷，陈硕，李骥等。我从你们那里学习到了太多的东西。很高兴能够和你们在一起讨论、研究。另外，感谢在交大结识到的许多朋友给我的帮助，包括李梅，向奔，顾俊，胡泊，张朝华，陈涛，陈绍，姜雷等等，有了你们，我的生活有了许多的颜色。

两年的时光，还有太多的感触，太多的人值得我感谢，我在这里就不一一列举了。在这里，请允许我衷心的祝愿老师，同学，朋友们今后事业有成，生活美满！最后我还要感谢提供给我们优秀学习机会的上海交通大学微电子学院。我们的学院虽然历史不长，但一直都充满热情和蓬勃向上的劲头。祝愿我们的学院发展越来越好，培养出更多优秀的学生。

于雪红

2008-12-30