

# 基于动态电容的复合电流源阶段延时计算方法

郭静静, 查佩文, 张树钢, 蔡志匡\*

(南京邮电大学集成电路科学与工程学院(产教融合学院), 江苏南京 210023)

**摘要:** 深亚微米工艺下传统时序模型的延时计算不再准确, 针对米勒效应愈发不可忽视、互连线电阻性增大和互连线延比重越来越大等问题, 本文提出一种基于动态电容的复合电流源阶段延时计算方法. 首先, 本文引入基于电压的插值方法支持复合电流源的延时计算; 其次, 构建II模型为负载的单元延时计算方法, 采用多阈值分析改进有效电容并推导动态电容, 实现迭代计算流程; 随后, 扩展动态电容概念到阶段延时计算中, 实现以分布式RC网络作为负载的阶段延时计算方法, 并使用机器学习优化互连线延时. 基于ASAP 7 nm Predictive PDK (ASAP 7)工艺, 本文提出的阶段延时计算方法与SPICE (Simulation Program with Integrated Circuit Emphasis)对比分别实现了1.49%、3.16%、1.70%、0.88%的单元延时、单元转换时间、互连线延时与互连线转换时间的平均相对误差, 平均在4~5次迭代后达到收敛.

**关键词:** 时序模型; 阶段延时计算; 复合电流源模型; 米勒效应; 动态电容

**基金项目:** 江苏省自然科学基金(No.BK20240637)

**中图分类号:** TN47; TN401

**文献标识码:** A

**文章编号:** 0372-2112(2025)07-2428-13

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20241149

## Composite Current Source Stage Delay Calculation Method Based on Dynamic Capacitance

GUO Jing-jing, ZHA Pei-wen, ZHANG Shu-gang, CAI Zhi-kuang\*

(College of Integrated Circuit Science and Engineering (College of Industry-Education Integration),  
Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China)

**Abstract:** With the deep sub-micron technology in integrated circuit develops, Miller effect becomes non-negligible and the increasing interconnection resistivity increases, leading to increased delay time prediction inaccuracy. A composite current source stage delay calculation method based on dynamic capacitance is proposed in this paper. A voltage-based interpolation method is firstly introduced to support the delay calculation of composite current source. A cell delay calculation method with a II model load is then established and a multi-threshold analysis is used to improves the effective capacitance, derives the dynamic capacitance and realizes the iterative calculation process. The dynamic capacitance concept is applied to the stage delay calculation, and used to realize the stage delay calculation method with distributed RC network as the load. By using machine learning, the interconnect wire delay is further optimized. Based on the ASAP 7 nm predictive PDK (ASAP 7) technology, the stage delay calculation method proposed in this paper achieves an average relative error of 1.49%, 3.16%, 1.70%, and 0.88% for the cell delay, cell transition time, interconnect line delay, and interconnect line transition time, respectively, compared with simulation program with integrated circuit emphasis (SPICE) simulation, the stage delay calculation method reaches convergence with about 4~5 iterations.

**Key words:** delay model; stage delay calculation; composite current source model; miller effect; dynamic capacitance

**Foundation Item(s):** Natural Science Foundation of Jiangsu Province (No.BK20240637)

## 1 引言

随着工艺尺寸的减小和电路规模的增大,快速准确的静态时序分析方法在集成电路设计中扮演着越来越重要的角色. 现代静态时序分析流程的核心是延时计算技术. 延时计算通常在门级进行,通过将电路划分为阶段来分析电路延时<sup>[1]</sup>. 每个阶段包括一个驱动单元,一个或多个接收单元,以及一个互连. 图1展示了门级延时计算中的阶段,该阶段包含一个驱动器、一个互连线及两个接收器. 在一个阶段中,某个单元作为接收器,而在传播过程中,则作为下个阶段的驱动器,每个阶段分别对驱动单元与互连线计算各自的延时与转换时间.

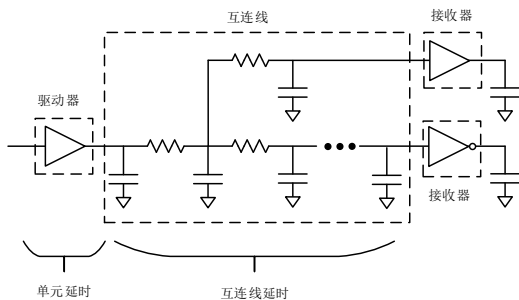


图1 门级延时计算中的阶段

在单元延时领域,传统的基于电压源模型的延时计算不再准确,电流源模型被提出,比如Synopsys提出了有效电流源模型,Cadence提出了复合电流源模型. 但是从传统模型转向复合电流源模型存在许多问题. 其一是插值困难,插值最接近的预先特性化电流波形耗费大量运行时间. 文献[2]提出了对复合电流源模型的查找表(LookUp Table, LUT)进行共形映射的插值方法,每次操作3个电流插值,该算法复杂度高,过程繁琐. 其二是基于常数有效电容的计算方法不再适用于电流源模型<sup>[3]</sup>,此计算方法假设恒定的驱动输出有效电容,这对于在转换过程中单元输出电容变化显著的情况下是不适用的. “有效电容( $C_{\text{eff}}$ )”方法由文献[4]引入,使用了一个两段输出波形并提出了单级单元的有效电容计算方法. 此迭代算法需要五到十次迭代才能收敛. 文献[5]提出了一个与输入信号阈值无关的经验性时变Thevenin等效单元模型. 使用Newton-Raphson迭代来计算有效电容和Thevenin电压源参数,计算成本高. 上述方法<sup>[4,5]</sup>主要专注于单元延时估计,无法准确匹配输出波形,从而导致高达50%的转换时间误差<sup>[6]</sup>.

在互连线延时领域,Elmore延时<sup>[7]</sup>是最基本、简单且计算效率最好的线延时计算方法. 由于Elmore延时考虑的是脉冲响应一阶矩,所以使用二阶矩的延迟度

(Delay Metric Based on the Second Moment, D2M)<sup>[8]</sup>,比Elmore能得到更高的精度. 但是当互连线具有多个分支时,电路中的充电和放电波形表现出更加复杂的行为,Elmore和基于二阶矩的D2M精度偏差越来越大. 随着工艺的持续缩小,严重的耦合效应开始显现,渐进波形估计(Asymptotic Waveform Evaluation, AWE)<sup>[9]</sup>、Arnoldi<sup>[10]</sup>能够匹配更高阶的响应时刻进行更高阶的估计的方法被提出. 然而,大多数利用高阶矩的技术都是计算密集型的,复杂度很高,运行时间长. 文献[11]将有效电容的概念扩展到Elmore延时的计算中,避免了对高阶矩计算的需求,它采用迭代延时优化方法,处理波形和有效电容之间的相互依赖性. 在静态时序分析(Static Timing Analysis, STA)中,利用机器学习(Machine Learning, ML)解决复杂的互连延时问题已经成为一种新趋势<sup>[12-15]</sup>. 文献[16]将文献[11]中的迭代有效电容算法与人工神经网络结合(Artificial Neural Network, ANN). 然而这种方法仅将驱动器输出端的信息作为特征,忽略了互连的拓扑结构. 文献[17]提取拓扑特征以捕捉RC网络的特性,使用极限梯度提升(eXtreme Gradient Boosting, XGBoost)来训练互连线延时的计算模型,并提出了一种断环算法将非树形网络转换为树形结构,采用一些基于矩的指标为特征,例如Elmore、D2M,却忽略了电阻屏蔽效应与米勒效应. 文献[18]利用一个前瞻RC网络为互连线延时计算提取互连与单元的特征,这种方法对单元的处理过于简化,不适用于签核精度的延时计算.

本文使用工业最先进的复合电流源模型研究阶段延时计算方法,该方法考虑了米勒电容与电阻屏蔽效应对延时的影响. 本文构建单元延时的计算框架,引入了基于电压的插值方法,使运行时间缩短并支持利用动态电容的迭代计算流程,建立了针对 $\Pi$ 模型的单元延时计算方案. 通过动态电容的概念将单元延时推广到阶段延时,建立了针对分布式RC网络的阶段延时计算方案,并使用机器学习增强互连线延时计算结果,使互连线与阶段延时得到优化.

## 2 基于电流源模型的单元延时计算方法

针对复合电流源模型插值困难和传统有效电容计算方法不适用问题,本文引入了新的基于电压的插值方法,使用多阈值分析方法将有效电容改进为动态电容,建立了针对 $\Pi$ 模型的迭代延时计算流程.

### 2.1 复合电流源模型的电压插值

传统模型的插值方法步骤冗长、计算成本高,本文设计了复合电流源模型基于电压的插值方法. 由于库文件最小化数据存储的需求,模型中的输出电流波形数据并非高密度,查找表中未定义的输入引脚处的输

入转换时间和输出引脚处的负载电容需要插值计算. 传统插值方法直接应用于稀疏数据集, 对电流插值会导致波形失真, 另外, 对电流共形映射的方法复杂度高且不稳定.

针对上述问题, 本文利用梯形积分和双线性插值得到查找表中未定义的输入转换时间和输出负载电容的组合, 从而得到精确的输出电压序列, 具体步骤如图2所示.

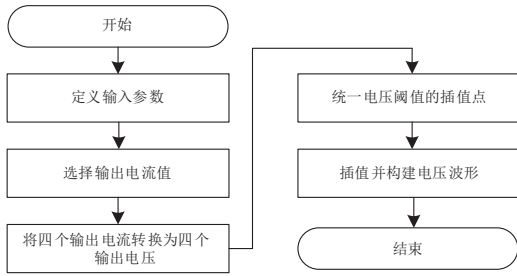


图2 基于电压的插值流程

步骤一, 定义输入参数. 根据实际输入转换时间 slew 和输出负载 load, 指定 (slew, load) 为输入参数.

步骤二, 选择输出电流值. 由输入参数 (slew, load) 在单元查找表中获取输入转换时间索引值中 slew 的前后两个索引值  $slew_1$ 、 $slew_2$ ; 获取输出负载索引中 load 的前后两个索引值  $load_1$ 、 $load_2$ , 由  $slew_1$ 、 $slew_2$  与  $load_1$ 、 $load_2$  四个索引值组成四个二维索引 ( $slew_1, load_1$ )、( $slew_1, load_2$ )、( $slew_2, load_1$ )、( $slew_2, load_2$ ). 根据四个二维索引, 查询复合电流源的输出电流表 output\_current\_rise/fall LUTs, 得到指定的四组输出电流值序列与时间值序列, 每一组输出电流值与时间值序列对应一个驱动器输出电流表.

步骤三, 将四个输出电流转换为四个输出电压. 如图3所示, 步骤二得到图3(a)的驱动器输出电流波形, 需要将其转化为图3(b)的驱动器输出电压波形. 对分段电流进行梯形积分得到增量电荷  $\Delta Q$ :

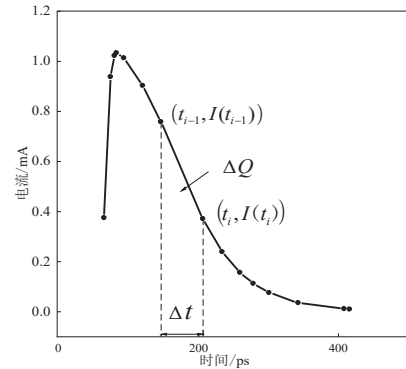
$$\Delta Q = \frac{[I(t_i) + I(t_{i-1})](t_i - t_{i-1})}{2} \quad (1)$$

式中,  $I(t_{i-1})$ 、 $I(t_i)$  分别为第  $i-1$  个时间点  $t_{i-1}$ 、第  $i$  个时间点  $t_i$  对应的驱动器模型输出电流值.

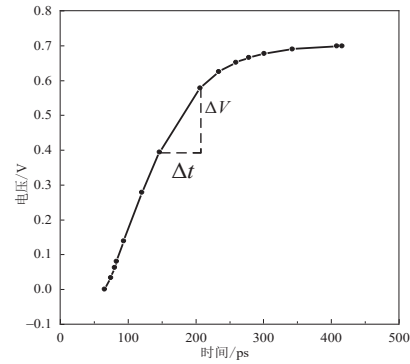
根据电容充放电公式  $\Delta Q = load \times \Delta V$ , 可以得到图3(b)中每两点间的增量电压, 其中 load 为驱动器模型纯电容输出负载的索引, 得到驱动器输出电压的表达式:

$$V(t_n) = \sum_{i=1}^n \frac{1}{2 \times load} [I(t_i) + I(t_{i-1})](t_i - t_{i-1}) \quad (2)$$

其中,  $V(t_n)$  为第  $n$  个时间点  $t_n$  对应的驱动器模型输出电压值, 转换时从第一个时间点  $t_n$  对应的电压值 0 开始累加, 第二个时间值  $t_i$  对应的电压值为  $V(t_i)$ , 直到最后一



(a) 电流波形



(b) 电压波形

图3 电流波形转换为电压波形

个电压值  $V(t_n)$ , 从而将输出电流转换为分段输出电压.

对四个二维索引 ( $slew_1, load_1$ )、( $slew_1, load_2$ )、( $slew_2, load_1$ )、( $slew_2, load_2$ ) 查询到的四个输出电流波形进行四次转换操作, 得到四个对应的输出电压波形.

步骤四, 统一电压阈值的插值点. 由于每个输出电流的表征规模大小 (时间-电流对的个数) 不一致, 导致输出电压规模不一致, 无法直接对电压波形插值, 需要先指定统一的插值电压阈值点.

对上一步得到的四个驱动器输出电压波形进行线性插值补点, 通过式(3)计算插值电压阈值对应的时间值:

$$t_m = t_{n-1} + (t_n - t_{n-1}) \left| \frac{V(t_m) - V(t_{n-1})}{V(t_n) - V(t_{n-1})} \right| \quad (3)$$

其中,  $t_m$  是输出电压波形插入的第  $m$  个插值电压阈值对应的的时间值;  $V(t_m)$  为输出电压波形第  $m$  个插值电压阈值. 遍历驱动器模型输出电压值  $V(t_n)$ , 直到插值电压阈值位于  $V(t_{n-1})$  与  $V(t_n)$  两个电压值之间, 所有插值电压阈值连接组成新的输出电压波形.

实际应用中可以根据精度与速度来确定具体的插值电压阈值. 本文统一采用图4所示电压点作为插值电压阈值, Vdd 为 0.7 V.

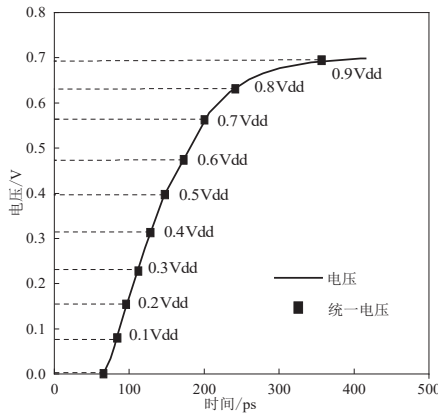


图4 统一电压波形

步骤五,插值并构建电压波形. 遍历四个电压波形的每个电压阈值,插值同一电压阈值下四个波形对应的四个时间值得到新的时间值.

如图5所示,因为(slew,load)为二维参数,所以在输入转换时间上进行一次线性插值,得到(slew,load<sub>1</sub>),(slew,load<sub>2</sub>),再在输出负载上的进行一次线性插值得到(slew,load).这个双线性插值过程如式(4)所示:

$$\begin{aligned}
 t_m = & \frac{(\text{slew}_2 - \text{slew})(\text{load}_2 - \text{load})}{(\text{slew}_2 - \text{slew}_1)(\text{load}_2 - \text{load}_1)} t_{m1} \\
 & + \frac{(\text{slew}_2 - \text{slew})(\text{load} - \text{load}_1)}{(\text{slew}_2 - \text{slew}_1)(\text{load}_2 - \text{load}_1)} t_{m2} \\
 & + \frac{(\text{slew} - \text{slew}_1)(\text{load}_2 - \text{load})}{(\text{slew}_2 - \text{slew}_1)(\text{load}_2 - \text{load}_1)} t_{m3} \\
 & + \frac{(\text{slew} - \text{slew}_1)(\text{load} - \text{load}_1)}{(\text{slew}_2 - \text{slew}_1)(\text{load}_2 - \text{load}_1)} t_{m4}
 \end{aligned} \tag{4}$$

其中,  $t_{m1}$ 、 $t_{m2}$ 、 $t_{m3}$ 、 $t_{m4}$  分别为(slew<sub>1</sub>,load<sub>1</sub>),(slew<sub>1</sub>,load<sub>2</sub>),(slew<sub>2</sub>,load<sub>1</sub>),(slew<sub>2</sub>,load<sub>2</sub>)四个组合获取的驱动器模型输出电压波形第  $m$  个分段电压阈值对应的时间值.

插值分段电压阈值如图6所示,以0.2Vdd处为例,驱动器模型各输出电压到达同一插值电压阈值的时间不同,采用双线性插值计算电压阈值0.2Vdd处新的时间值.遍历所有电压阈值后得到各个电压阈值对应的新时间值,将电压阈值与新时间值连接得到插值后的输入转换时间与输出负载的完整电压波形.

### 2.2 II模型的动态电容的计算

尽管电流查找表将输出负载假设为纯电容,现实中的驱动单元却常常需要驱动包含电阻和电容的复杂RC网络.本节通过将复杂的RC网络等效为多个动态电容,实现对输出电压的准确计算.

#### 2.2.1 II模型的有效电容计算

驱动器模型可以视为电压源,若假设驱动器输出

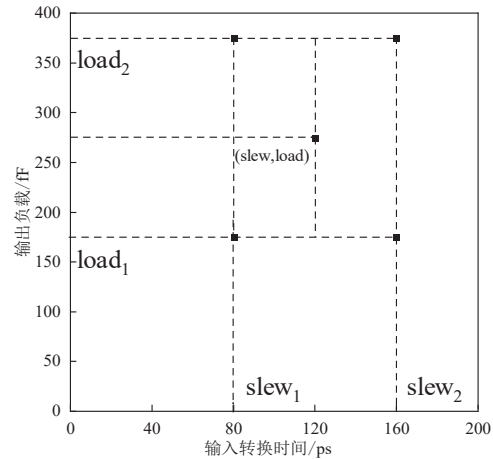


图5 双线性插值

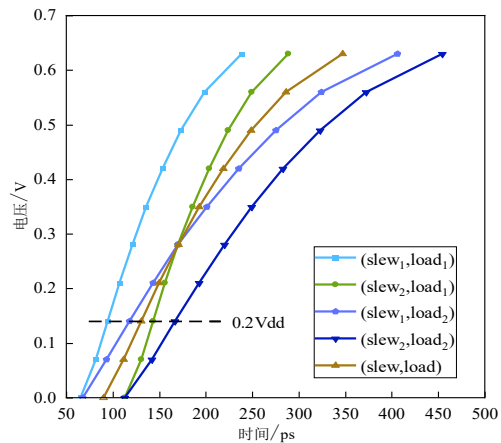


图6 构建插值后的电压波形

电压波形为理想波形,则上升斜坡的输出可以表示为

$$V(t) = \frac{V_{dd}}{T} t, \quad t \in [t_1, t_2] \tag{5}$$

其中,  $T$  代表线性斜坡从0跳变到  $V_{dd}$  的转换时间. 如图7所示,该单元驱动一个II模型,该模型由近端电容  $C_1$ 、电阻  $R$ 、远端电容  $C_2$  组成. 根据基尔霍夫电流定律可得:

$$I(t) = I_1(t) + I_2(t) \tag{6}$$

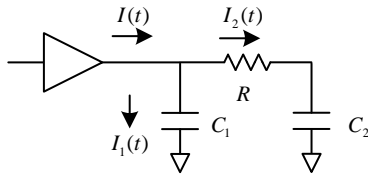
其中,  $I(t)$  是总电流;  $I_1(t)$  是经过电容  $C_1$  的电流;  $I_2(t)$  是经过电阻  $R$  与电容  $C_2$  的电流.

对电流进行复频域分析,可得:

$$\begin{aligned}
 I(s) = & I_1(s) + I_2(s) \\
 = & V(s) \left( sC_1 + \frac{sC_2}{1 + sRC_2} \right)
 \end{aligned} \tag{7}$$

之后将复频域的  $I(s)$  转换为时域,可得:

$$\begin{aligned}
 I(t) = & I_1(t) + I_2(t) \\
 = & \frac{V_{dd}}{T} \left( (C_1 + C_2) - C_2 e^{-\frac{t}{RC_2}} \right)
 \end{aligned} \tag{8}$$

图7  $\Pi$ 模型负载

通过对总电流进行积分,即可求得电压源流入 $\Pi$ 模型所产生的电荷量,如式(9)所示:

$$Q = \int_0^T I(t) dt \quad (9)$$

$$= \int_0^T \frac{V_{dd}}{tr} \left( (C_1 + C_2) - C_2 e^{-\frac{t}{RC_2}} \right) dt$$

将电压源流入 $\Pi$ 模型的电荷量等同于单个纯电容在达到相同电压时获得的电荷量,此电容即为有效电容 $C_{\text{eff}}$ ,如式(10)所示,式(11)将其进一步化简为

$$Q = C_{\text{eff}} \cdot V = \int_0^T I(t) dt \quad (10)$$

$$= \int_0^T \frac{V_{dd}}{T} \left( (C_1 + C_2) - C_2 e^{-\frac{t}{RC_2}} \right) dt$$

$$C_{\text{eff}} = C_1 + C_2 \left( 1 - \frac{RC_2}{T} \left( 1 - e^{-\frac{T}{RC_2}} \right) \right) \quad (11)$$

其中, $C_2$ 的系数为电阻屏蔽系数 $k$ .当互连电阻极小时,有效电容基本等同于总电容,即 $k \approx 1$ ;当互连电阻较大时,则有效电容可以近似为近端电容 $C_1$ ,即 $k \approx 0$ ,相当于无限大的电阻造成开路,产生电阻屏蔽效应.

对于给定的线性斜坡 $V(t)$ , $C_{\text{eff}}$ 可等效原本的 $\Pi$ 模型在时序上的表现.有效电容定义为等效延迟阈值点之前的波形而不是整个波形,因此式(5)具体表现为0至 $0.5V_{dd}$ 延时阈值点之间的输出波形.式(5)可修改为

$$V(t) = \frac{0.5V_{dd}}{t_{0.5V_{dd}}} t \quad (12)$$

代表输出波形从零时刻的电压零点至 $t_{0.5V_{dd}}$ 时刻的 $0.5V_{dd}$ 延时阈值点之间为线性斜坡,可推导有效电容为

$$C_{\text{eff}} = C_1 + C_2 \left( 1 - \frac{RC_2}{t_{0.5V_{dd}}} \left( 1 - e^{-\frac{t_{0.5V_{dd}}}{RC_2}} \right) \right) \quad (13)$$

### 2.2.2 动态电容的计算

传统方法使用单一等效电容来代表整个波形存在明显的局限性.驱动器的输出波形不是简单的线性斜坡,有效电容匹配波形中点而不能匹配后半段波形,其估算过程过分侧重于延时而忽视了波形的完整性,导致转换时间估计误差高达50%.不仅如此,对整个转换使用单个电容值会导致米勒效应显著的单级单元不准确,接收单元输入引脚电容变化很大,单个电容无法模

拟这种变化.因此需要多个电容来匹配驱动器输出波形的各个分段,使用多阈值分析方法引入动态电容来实现更精细的波形匹配.

将式(13)中0至 $V_{dd}$ 区间内有效电容的计算扩展至定积分形式,代表 $T^l$ 与 $T^{l+1}$ 之间电流的积分,即增量电荷 $Q^{l \rightarrow l+1}$ :

$$Q^{l \rightarrow l+1} = C_d^{l \rightarrow l+1} (V^{l+1} - V^l) = \int_{T^l}^{T^{l+1}} I(t) dt \quad (14)$$

$$= \int_0^{T^{l+1}} I(t) dt - \int_0^{T^l} I(t) dt$$

通过在该电压区间内的等效电荷来计算该电压范围的等效电容,如式(15)所示:

$$Cd^{l \rightarrow l+1} = \frac{V^{l+1} C^{0 \rightarrow l+1} - V^l C^{0 \rightarrow l}}{V^{l+1} - V^l} \quad (15)$$

式(15)的动态电容 $Cd^{l \rightarrow l+1}$ 代表了区间 $[l, l+1]$ 的等效电容值.通过这种方法将驱动器建模为非线性动态系统,并通过电容值将其参数化.可以发现,动态电容的求解对输出电压波形的要求仅为阈值电压通过时间,基于电压的插值方法也仅使用阈值电压通过时间来表示电压波形,二者相匹配.动态电容相比传统有效电容,匹配电荷增量而不是延时阈值点,模拟更准确.

### 2.3 单元迭代流程

动态电容受驱动单元及其负载特性的综合影响,尤其是接收器特性.驱动器的输出波形可以视作其驱动特性和负载特性的函数表现.接收器的输入引脚电容取决于单元的输入转换时间和输出负载电容,从而影响驱动器的输出波形、延时目标引脚的转换时间.接收器输入引脚的转换时间由驱动器的输出波形决定,表明二者之间存在相互依存的关系.据此,确定动态电容的过程采用迭代方法:从初始估算值开始,经过有限次迭代后达到收敛状态.

动态电容的区域划分需要设置分段阈值电压,分段电压阈值介于0与电源电压 $V_{dd}$ 之间,每个分段对应一个动态电容.本文将驱动器模型输出电压 $[0, V_{dd}]$ 划分为 $[0.2V_{dd}, 0.5V_{dd}, 0.9V_{dd}]$ 等区间.对每一段分别执行插值操作.

迭代流程如图8所示.

步骤1:进行初始化,为所有动态电容设置初始值为 $\Pi$ 模型的总电容.

步骤2:构建输出波形.根据给定的输入转换时间、各个分段的动态电容,插值得到每个分段的电压波形.如图9所示,存在3个动态电容,其中 $Cd^{l \rightarrow l+1}$ 的波形通过 $V^l$ 与 $V^{l+1}$ 的时间差值为该段的时间长度 $T^{l \rightarrow l+1}$ ,由式(16)求解,通过叠加各段时间长度对应的波形得到驱动器输出电压波形.

步骤3:更新动态电容.利用每个分段电压的通过

时间和式(11)计算有效电容并利用式(15)进一步计算动态电容.

步骤4:计算驱动器输出电压的延时与转换时间.利用输出电压到达延时电压阈值、转换时间低百分比阈值、转换时间高百分比阈值的时间划分电压区间,根据电压区间对应的通过时间进行延时计算.

步骤5:判断迭代收敛条件.转换时间收敛则结束迭代,输出单元延时与转换时间.不收敛则返回步骤2,构建新的输出波形,并更新每个电压区间的动态电容.同时,可以得到完整且精确的输出电压波形与下一级单元的输入电压波形.

$$T^{l \rightarrow l+1} = \theta(Cd^{l \rightarrow l+1}, V^{l+1}) - \theta(Cd^{l \rightarrow l+1}, V^l) \quad (16)$$

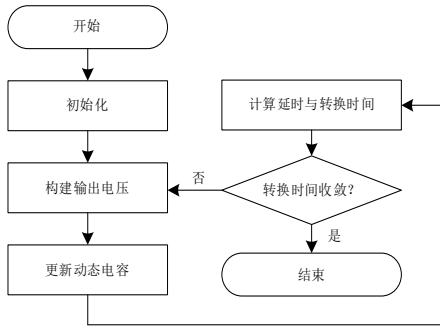


图8 单元延时迭代流程

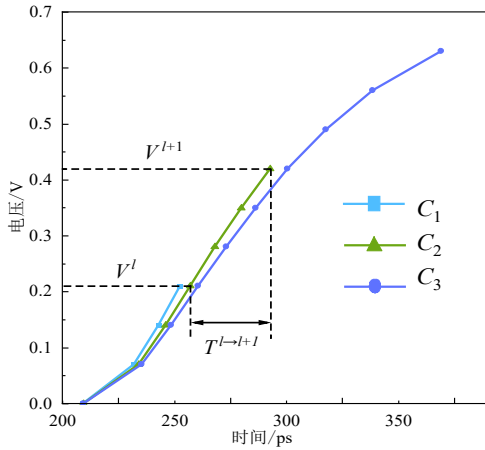


图9 组合输出波形

### 3 基于动态电容的阶段延时计算方法

本节将  $\Pi$  模型的动态电容计算推广到 RC 树的迭代计算流程,建立针对阶段延时计算的初步方案,并结合机器学习优化互连线延时计算方法.

#### 3.1 RC 树中的动态电容计算

针对驱动器驱动 RC 树的情况,扩展波形在 RC 树中的传播计算方法,考虑包含米勒电容与复杂拓扑结构的有效电容计算.

#### 3.1.1 波形传播

在实际应用中,需要特别关注接收单元输入端的信号,而复合电流源时序模型提供的是驱动器输出端,即 RC 网络源端的波形. 本节关注波形从源端传播到接收端的过程,推导波形在每个节点的计算方案.

(1)通过时间的计算

RC 树可以视为多个  $\Pi$  模型互连,因此如图 10 所示,使用  $\Pi$  模型作为波形传播基本单元,将节点  $i$ 、节点  $j$  以及电阻  $R$  看作  $\Pi$  模型,  $V_j(t)$  由  $V_i(t)$  得出,  $T_i$  和  $T_j$  代表电压区间内节点  $i$ 、节点  $j$  波形从 0 到电压阈值  $V$  的持续时间.

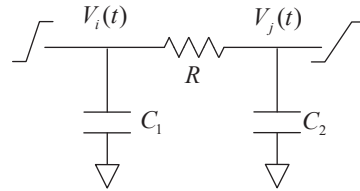


图10 线性波形传播

由基尔霍夫电压定律,可得:

$$V_i(t) = V_j(t) + I_2(t)R \quad (17)$$

将分段线性 (PieceWise Linear, PWL) 波形简化为斜坡波形描述节点  $i$  处电压波形:

$$V_i(t) = \frac{V}{T_i}t \quad (18)$$

将式(8)中的表达式代入式(18),得到:

$$V_j(t) = \frac{V}{T_i}t - \frac{V}{T_i}C_2 \left(1 - e^{-\frac{t}{RC_2}}\right)R \quad (19)$$

当  $t = T_i$  时,节点  $j$  的电压为

$$V_j(t = T_i) = V - \frac{VC_2}{T_i} \left(1 - e^{-\frac{T_i}{RC_2}}\right)R \quad (20)$$

依据图 11 中  $V_j(t = T_i)$  与  $t = T_i$  形成的三角形,设  $t = T_j$  时  $V_j(t = T_j) = V$ ,根据等比性质可得式(21)比例关系:

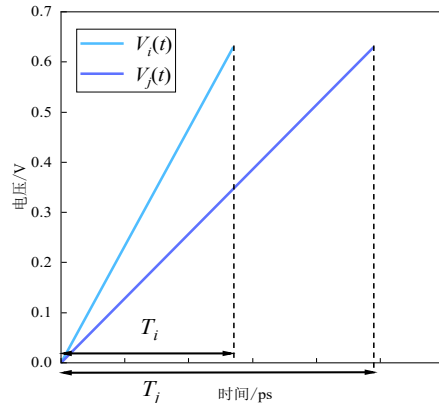


图11 依据线性传播波形推导输出波形

$$\frac{V_j(t=T_i)}{T_i} = \frac{V}{T_j} \quad (21)$$

将式(20)代入式(21),得到式(22),即II模型中斜坡电压从节点*i*到节点*j*的传播过程.

$$T_j = \frac{T_j}{1 - \frac{C_2}{T_i} \left(1 - e^{-\frac{T_j}{RC_2}}\right) R} \quad (22)$$

### (2)构建波形

实际应用中单个电容值不能完全近似单元的输入电压波形.RC网络中的波形应该反映非线性行为,一方面由于驱动器输出点由多个动态电容建模,另一方面接收器输入点也需要多个电容模拟米勒电容.因此传播的波形不应该是简单的线性斜坡,而是分段波形.通过式(23)的叠加计算得到每个分段阈值电压之间的增量时间,可以构建节点*j*的波形,作为下一个节点的输入:

$$T_j^{l-1 \rightarrow l} = \frac{V^l - V^{l-1}}{V^l} T_j^{0 \rightarrow l} \quad (23)$$

### 3.1.2 复杂情况下的有效电容计算

实际情况下RC树中互连线网络除了II模型,还包含接收单元输入引脚电容,并且工艺的升级导致米勒效应愈加显著,电路拓扑结构更加复杂.本节介绍使用复合电流源接收器模型建模米勒效应,阐述RC树中有效电容的计算方案,并利用有效电容计算动态电容.

#### (1)复合电流源接收器模型建模米勒效应

当互连阻抗较小时,电阻屏蔽效应不明显,米勒效应影响增大.传统接收器模型输入引脚使用单一电容值,或者分别设定上升/下降沿的电容值过于保守.图12展示了单元CKINVDCx12的输出波形,三条曲线分别为CKINVDCx12将2fF纯电容、5fF纯电容与单元CKINVDCx20作为接收器的波形.CKINVDCx20作为接收器时的波形与驱动单一电容2fF和5fF时的波形都不拟合,说明CKINVDCx20的米勒电容在转换过程中变化很大,因此单一电容值的使用在整个转换期间会导致不准确.

接收单元的输入引脚电容由查找表得到,接收器模型由模拟非线性接收器输入晶体管电容 $C_1$ 和米勒效应的输入引脚电容 $C_2$ 组成,记录驱动器从初始状态到切换阈值期间看到的电容.电容值依赖于输入转换时间、输出负载和单元状态,允许转换期间动态调整.

有效电容需要使用接收单元输入引脚电容计算,II模型中接收单元输入引脚电容在RC网络输出端口处,可以视作与远端电容 $C_2$ 并联.屏蔽效应导致驱动器无法看到完整的接收单元输入引脚电容.将式(8)更新为

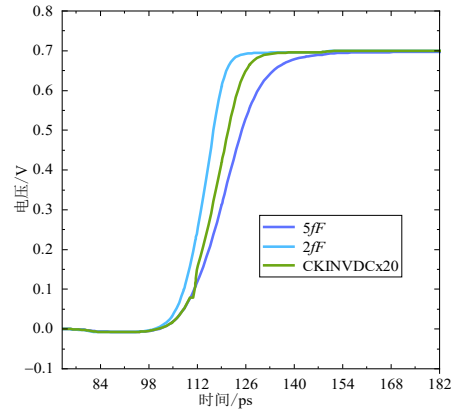


图12 不同负载下单元CKINVDCx12的输出波形

$$I(t) = I_1(t) + I_2(t) = \frac{V_{dd}}{T} \left( C_1 + C_2 + C_{pin} - (C_2 + C_{pin}) e^{-\frac{t}{R(C_2 + C_{pin})}} \right) \quad (24)$$

其中, $C_{pin}$ 代表接收单元输入引脚电容.积分更新后的 $I(t)$ 得到电压源在II模型中产生的电荷量:

$$Q = \int_0^T I(t) dt = \int_0^T \frac{V_{dd}}{T} \left( C_1 + C_2 + C_{pin} - (C_2 + C_{pin}) e^{-\frac{t}{R(C_2 + C_{pin})}} \right) dt \quad (25)$$

计算新的有效电容,如式(26)所示:

$$C_{eff} = C_1 + (C_2 + C_{pin}) \cdot \left( 1 - \frac{R(C_2 + C_{pin})}{T} \left( 1 - e^{-\frac{T}{R(C_2 + C_{pin})}} \right) \right) \quad (26)$$

#### (2)RC树中的有效电容计算优化方案

驱动分布式RC网络时有效电容计算更加复杂,如图13所示,节点*i*有多个扇出节点,分布式RC网络可以看作是多II模型连接而成.有效电容需要等效一个近端电容与多个被不同的电阻屏蔽的远端电容.节点*i*有*n*个相邻后端节点*i+1*至*i+n*.电阻屏蔽效应导致节点*i+1*的下游总电容 $C_{total,i+1}$ 被 $R_{i+1}$ 屏蔽了一部分.所以节点*i*的 $C_{eff_i}$ 需要分别求出多个II模型的电阻屏蔽系数*k*.计算公式如下:

$$C_{eff_i} = C_i + C_{total,i+1} \cdot \sum_{i+1}^n \left( 1 - \frac{R_{i+n} C_{eff_{i+n}}}{T_i} \left( 1 - e^{-\frac{T_i}{R_{i+n} C_{eff_{i+n}}}} \right) \right) \quad (27)$$

#### (3)RC树中的动态电容计算方案

最后,计算RC树中每个节点的动态电容:

$$Cd_i^{l \rightarrow l+1} = \frac{V^{l+1} C_{eff_i}^{0 \rightarrow l+1} - V^l C_{eff_i}^{0 \rightarrow l}}{(V^{l+1} - V^l)} \quad (28)$$

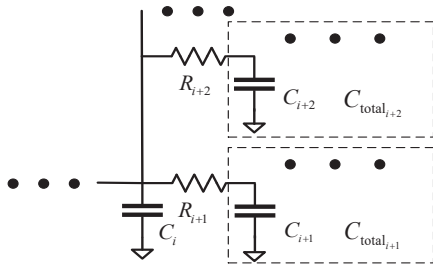


图 13 多扇出节点

这说明对于任一节点、任一段,动态电容只与相邻有效电容和分段阈值电压有关.RC 树中更新有效电容与更新动态电容是同步的.

### 3.2 阶段延时迭代算法

本节扩展分析驱动器驱动更为复杂 RC 树的场景,在波形的传播计算方面引入了更多的挑战.完整流程如图 14 所示.

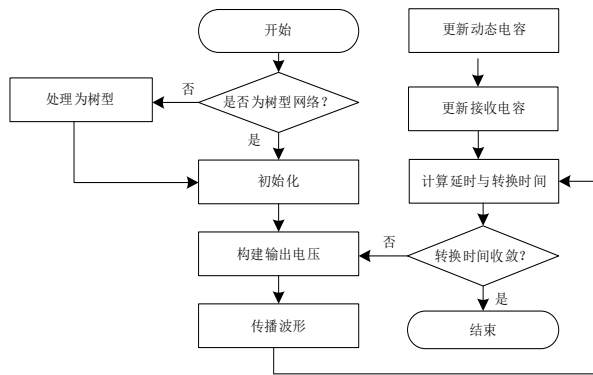


图 14 阶段迭代流程

从设计数据中提取寄生网络信息及时序单元库数据,判断 RC 网络是否是树型结构,不是树型则使用断环算法<sup>[17]</sup>调整.此外,初始化 RC 网络中每个节点的有效电容为每个节点下游的总电容.

迭代更新过程如图 15 所示,首先计算源端节点的波形,即驱动器的输出波形.通过插值单元输入引脚处输入转换时间和各个分段的 RC 网络源端的动态电容的各种组合获得电压波形.驱动器参数是驱动单元的输入转换时间与节点的动态电容,其中动态电容为初始化的源端节点有效电容.随后基于源端波形,将波形沿 RC 网络从源端节点正向传播至所有接收端节点,如图 15 所示.正向传播波形过程中,采用深度优先搜索(Depth First Search, DFS)的前序遍历,利用有效电容改进的 Elmore 延时考虑传输路径上的延时,得到互连线延时与互连线转换时间.根据接收端波形,更新接收器引脚的电容值.

正向传播结束后从接收端节点开始反向传播至源端节点,结合深度优先搜索更新每个节点的有效电容,

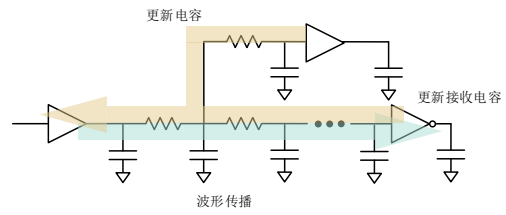


图 15 RC 树中双向传播

此过程中考虑了电阻屏蔽效应和下游总电容的影响,根据源端输出波形计算单元延时与单元转换时间.

最后,判断计算结果的收敛性,特别是转换时间的变化是否收敛,否则重复迭代过程直至收敛.

### 3.3 机器学习增强的互连线延时

上述算法收敛之后,采用 XGBoost 机器学习模型,利用过程中提取的特征来优化互连线延时.这一过程利用阶段延时计算过程中使用的动态电容等信息.分别训练了两个模型以计算互连线延时与互连线转换时间.

#### 3.3.1 特征选择

互连线延时模型特征选择如表 1,由于单元输入转换时间经过不同单元得到的输出波形不同,文献[17]使用单元转换时间作为特征,面对不同互连线会产生不同输出波形,因此本文选择驱动器输出波形作为特征.

表 1 特征值

来源	特征大小	特征
单元输出	1	输入转换时间
互连线 & 接收器	1	扇出数量
	30	路径上节点动态电容
	10	路径上电阻
	1	互连线延时
	1	互连线转换时间

在电压改变时,米勒电容会导致 RC 网络的电容发生改变,并且不同的输入波形会影响米勒电容的计算.本文动态电容考虑了米勒电容、输入波形与电阻屏蔽等因素.因此选择从驱动器输出引脚至目标接收器输入引脚单向路径上所有节点的动态电容与电阻作为特征.

复杂布局设计会导致深树等网络拓扑结构,针对不同 RC 网络路径上的节点数量不同导致动态电容数量不同,训练模型时需要调整特征参数以确定特征数据的数量.在每一轮的交叉验证中,比较验证集中接收端节点互连线延时预测值与线延时标准样本集中 SPICE 仿真测量值之间的误差,选择误差最小的节点数量作为特征参数的统一值,本文为 10.

### 3.3.2 模型选择

为了评估所选机器学习模型在准确性上的优势,研究了三种不同的模型,即最小绝对收缩与选择算子(Least Absolute Shrinkage and Selection Operator, LASSO)回归算法、多层感知机(Multi-Layer Perceptron, MLP)和极限梯度提升 XGBoost. 如表 2 所示,互连线延时方面 LASSO 误差最大,达到了 17.9%. 在转换时间结果上,三者保持相似水平,但 XGBoost 仍表现最佳. XGBoost 在互连线延时与互连线转换时间方面均超过了其他两种模型,分别达到了 1.70% 与 0.88%. 基于上述观察,选择 XGBoost 作为本文机器学习模型.

表 2 互连线延时计算方法对比

方法		平均相对误差/%	
		互连线延时	互连线转换时间
未增强特征值		23.6	3.59
本文	MLP	7.58	1.26
	LASSO	17.9	1.71
	XGBoost	1.70	0.88

## 4 实验过程及结果分析

首先介绍本文的验证流程方案与实验环境,之后对比  $\Pi$  模型为负载的单元延时结果,以及单元延时计算中使用的插值方法,再对比分布式 RC 网络作为负载的机器学习增强的互连线延时计算结果、阶段延时计算结果.

### 4.1 环境配置和流程

本文方法使用 C++17 来实现. 环境方面所有实验都在配备了英特尔 Core i7 处理器和 16 GB DDR4 内存的 PC 上进行. 实验利用了 ACM TAU 2021 比赛的 C++ 框架,生成的阶段电路基于 ASU ASAP7 工艺设计套件. 其中 SPICE 参数中沟道长度设置为 21 nm, Fin 高度为 32 nm, Fin 厚度为 6.5 nm, 阈值电压设置为 0.25 V 和 -0.2 V.

导入时序模型时使用 liberty\_parser-2.6 来解析 liberty 库, 读取互连模型的寄生参数使用 OpenTimer/Parser-SPEF<sup>[19]</sup> 这一 C++ 仅头文件解析器来解析 SPICE (Simulation Program with Integrated Circuit Emphasis) 文件, 使用 Xyce Parallel Electronic Simulator (Xyce) 作为 SPICE 产品以产生标准结果<sup>[20]</sup>.

阶段延时计算流程方面, 本文开发了由程序 delay\_calc\_tool 调用的延时计算例程, 导入时序模型 (Liberty 格式), 读取测试电路, 读取互连模型的寄生参数 (SPEF 格式), 调用延时计算引擎, 得到单元延时、单元转换时间、互连线延时与互连线转换时间等延时指标.

验证阶段电路延时计算方法时, 一方面利用工艺设计套件中的单元网表、SPICE 模型和 SPEF 文件生成自动化 SPICE 网表, 进行瞬态仿真并测量仿真结果得到

延时指标的标准结果. 另一方面解析输入 SPEF 文件和时序库文件, 根据解析结果构建内部数据模型得到延时计算结果.

单元/阶段延时计算方法验证的实验背景如表 3 所示. 4.2 节验证以  $\Pi$  模型为互连负载的单元延时计算方法和基于电压的插值方法, 4.3 节验证以分布式 RC 网络作为负载的阶段延时计算方法.

表 3 单元/阶段延时计算方法验证实验背景

PDK	ASU ASAP7
工艺	7 nm
互连	$\Pi$ 模型/分布式 RC 网络
扇出	单个扇出/多个扇出
延时计算	单元/阶段延时指标

### 4.2 $\Pi$ 模型的单元延时计算结果

本节验证以  $\Pi$  模型为互连负载的单元延时计算方法和基于电压的插值方法.

#### 4.2.1 插值方法对比

采用 SPICE 模拟生成准确的波形数据作为基准, 对比其与插值波形到达特定电压值的时间来验证精确性, 大量的样本测试包括了内插与外推的情况. 实验结果如表 4 所示, 本文模型所有通过时间的平均相对误差与平均绝对误差都小于文献[2]. 平均相对误差方面, 0.1Vdd、0.5Vdd 与 0.9Vdd 延时阈值点处, 本文的误差为 2.51%、1.11% 和 0.74%, 而文献[2]的方法为 7.92%、4.58% 和 3.17%. 说明在关键的阈值点上本文插值波形与实际波形特征匹配良好.

表 4 不同电压阈值的插值波形过渡时间的对比

电压阈值	文献[2]		本文方法	
	平均相对误差	平均绝对误差	平均相对误差	平均绝对误差
0.1Vdd	7.92%	1.27	2.51%	0.41
0.2Vdd	7.05%	1.78	2.28%	0.49
0.3Vdd	5.75%	2.54	1.53%	0.61
0.4Vdd	4.96%	3.08	1.23%	0.72
0.5Vdd	4.58%	3.71	1.11%	0.88
0.6Vdd	4.81%	4.87	1.19%	1.17
0.7Vdd	4.74%	5.78	1.40%	1.67
0.8Vdd	3.80%	5.71	1.32%	1.82
0.9Vdd	3.17%	6.37	0.74%	1.42

本文与文献[2]的方法在 100、1 000、10 000 个插值测试例中的运行时间结果如表 5 所示. 三个测试例数量下本文方法都优于共形映射的方法 1 个数量级. 这表明本文提出的方法能够以更高的效率处理大规模的测试案例.

表5 运行时间的对比

测试例数量	运行时间/ms	
	文献[2]	本文方法
100	0.031	0.005
1 000	0.341	0.053
10 000	3.499	0.556

#### 4.2.2 II模型的单元延时结果对比

使用 ACM TAU 2021 竞赛提供的 Benchmark 测试集比较不同方法的单元延时计算方法表现,使用区分上升沿、下降沿的 10 000 个阶段,共存在 20 000 组测试例.使用的 II 模型负载是真实集成电路互连的代表性输入导纳模型,它们的长度不同,布置在不同的金属层上,其电阻和电容值涵盖了从 0.2 k $\Omega$  到 100 k $\Omega$  和从 0.000 1 pF 到 0.25 pF 的广泛范围.驱动器时序弧使用输入电压波形,其转换时间从 0.005 ns 到 0.32 ns 变化,代表了一个 ASU ASAP 前驱驱动器单元.接收器输出电容值范围为 0.000 4 pF 到 1.473 pF,这些值连同驱动器输入转换时间,涵盖了复合电流源模型预表征中使用的全部范围.

本文和文献[11]、文献[21]都考虑电阻屏蔽效应对负载计算进行优化,建立各自迭代过程.引入非线性延时模型对比复合电流源时序模型的优点,两者区别如表 6 所示.米勒电容建模方面,文献[11]的迭代过程并不考虑米勒电容,有效电容的计算准确性低.非线性延时模型采用区分上升沿与下降沿的单个电容,结果过于悲观.在电阻屏蔽方面,三种方法都包含计算有效电容的迭代过程,传统有效电容只匹配波形中点而不能匹配后半段波形,不满足转换时间的计算.文献[21]没有考虑阶段扩展到分布式 RC 网络后使用动态电容对初始的  $C_{near}$  和  $C_{far}$  进行更新,仅仅使用多个等效电容对 II 模型负载进行等效以提高精度.

表6 单元延时计算方法对比

	非线性延时模型	文献[21]	文献[11]	本文方法
考虑米勒电容	否	是	否	是
考虑电阻屏蔽	是	是	是	是
采用动态电容	否	否	否	是
使用复合电流源模型	否	是	否	是

表 7 展示了单元延时计算方案的结果比较.通过平均相对误差和离群值数量来评估模型的准确性.延时离群值定义为相对误差大于 5% 的测试用例,转换时间离群值定义为相对误差大于 10% 的测试用例.在延时计算方面,非线性延时模型误差较大,文献[11]使用了一个匹配零点至延时阈值点的有效电容,本文在此基础上利用了复合电流源模型的接收引脚电容建模米

勒电容,从而到达了更优的效果,误差降低了 16.7%.输出转换时间计算方面,本文平均相对误差为 2.64% 且仅有 157 个离群值,比非线性延时模型和文献[11]分别减少了 5.15% 和 12.86% 平均相对误差,离群值数量优于此二者一个数量级.这证明了非线性延时模型与文献[11]的单个有效电容对于波形后半段的匹配效果不佳,本文使用分段拟合波形的动态电容,优势明显,误差降低了 83.1%.离群值数量越小说明计算更准确,在时序分析时能保持较高稳定性.

表7 单元延时计算方法延时指标平均相对误差对比

延时指标		非线性延时模型	文献[21]	文献[11]	本文方法
延时	平均相对误差	3.96%	1.22%	1.45%	1.18%
	离群值	3944	104	388	55
转换时间	平均相对误差	7.79%	3.17%	15.5%	2.64%
	离群值	4384	540	9459	157

图 16 展示不同模型在预测电路延时和转换时间误差方面随着时间常数与转换时间的比值 ( $RC_2/T$ ) 分布的精度对比.横坐标是时间常数与转换时间的比值区间,代表各种情况下驱动器输入与负载,反映了测试例的复杂分布情况.图 16(a) 的单元延时误差,平均相对误差随着  $RC_2/T$  值的增加先增加后下降,本文提出方法相较于其他两种更稳定,始终维持在低水平线上.对于图 16(b) 的转换时间误差,本文方法在所有区间内都优于其他两种方法.

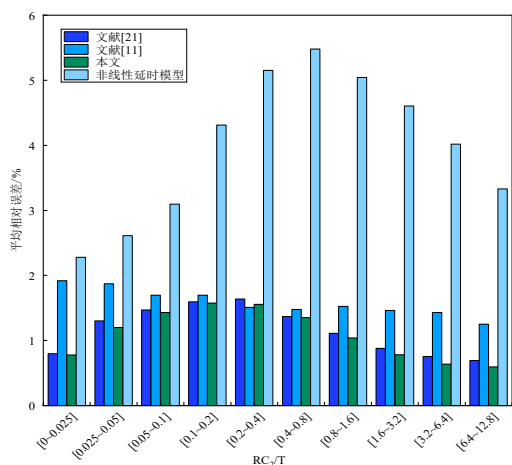
#### 4.3 分布式 RC 网络的阶段延时计算结果

本节验证以分布式 RC 网络作为负载的阶段延时计算方法.

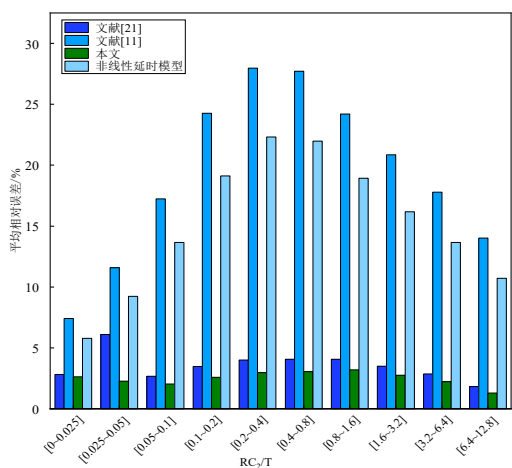
为了全面测试阶段延时,分布式 RC 网络具有以下特点:每单位长度具有适当的电阻和电容值.不定数量的接收器、不定线长与不定连通性.包括接收器输入引脚负载电容在内的总网络电容,总是处于表征的驱动器输出负载范围内.本节使用的测试样本由 ACM TAU 2021 提供的框架生成.

##### 4.3.1 机器学习增强的互连线延时对比

对于机器学习模型来说,特征选择比模型选择更为重要,为了评估本文提出的方法在机器学习增强互连线延时计算上带来的提升,表 8 展示了本文与基于不同特征的先前工作文献[17]和文献[18]的比较.文献[17]没有考虑有效电容,不能反映电阻屏蔽效应,预测结果不能达到签核精度.文献[18]利用随机森林(Random Forest, RF)对互连线延时进行预测,使用互连线总电容结合非线性延时模型进行查表,将得到的单元延时与单元转换时间作为特征.其没有考虑可变的互连线大小与米勒效应,仅使用总电容而非本文使用路径上的节点动态电容.



(a) 单元延时误差分布



(b) 单元转换时间误差分布

图 16 单元延时与单元转换时间随  $RC_T$  的误差分布

表 8 互连线延时计算方法对比

方法	平均相对误差/%	
	互连线延时	互连线转换时间
文献[17]	2.97	25.10
文献[18]	4.87	0.91
本文	1.70	0.88

在互连线延时方面,本文模型在文献[17]的节点 Elmore 延时基础上使用优化的有效电容度量显著增加了精度,对于接收器情况复杂的网络,米勒电容对延时的影响更大,本文相比于文献[17, 18],互连线延时误差分别减少了 42.8% 和 65.1%。

在互连线转换时间方面,文献[17]转换时间误差较大,由于以 Elmore 延时为特征不能正确捕捉到互连线上的波形传播情况。相比之下文献[18]将非线性延时模型的驱动器输出转换时间作为特征更为合理,但非线性延时模型精度较低,并且对驱动单元的简化导致了其输出转换时间的不精确。本文相比于文献[17, 18],互

连线转换时间误差分别减少了 96.5% 和 3.3%。

### 4.3.2 阶段延时结果对比

本文和文献[11]、文献[16]都处理了延时计算的迭代性质,电容依赖于转换时间,处理阶段延时类型分为以下三部分。

(1) 驱动器方面,分别用不同方法得到驱动器查表电容参数,文献[11]使用单一有效电容,文献[16]使用分段有效电容,本文使用动态电容。

(2) 互连方面,在迭代过程中的波形传播方法不同,文献[11]使用线性斜坡作为波形传播,文献[16]将每个分段的波形视作近似的线性斜坡,而本文使用动态电容计算通过时间,依据增量时间构建波形。为了对比的公平性,在实现其他方法时也添加了断环算法来处理电路环路。接收器方面,文献[11]所建立的迭代过程并不考虑米勒电容。文献[16]与本文都使用了两段电容对米勒电容建模。

(3) 机器学习增强方面,文献[16]将特征聚焦于驱动器输出引脚的电容参数,本文使用路径上每个节点的动态电容作为特征,捕捉拓扑、电阻屏蔽与米勒效应等因素。

表 9 展示了阶段延时的结果对比。与 4.2 节的实验不同,该实验负载为分布式 RC 网络,涉及断环处理和 RC 树的迭代,所以会出现单元延时与单元转换时间的精度低于  $\Pi$  模型负载的情况。本文在单元延时计算方法上明显优于文献[11],单元延时与单元转换时间误差减少了 72.1% 和 83.8%。本文在单元延时上的表现优于文献[16]的 1.63%,单元延时误差减少了 9.1%,但在单元转换时间上略低。在互连线延时指标上,文献[11]的误差为 15.74% 与 17.44%,这代表传统延时模型已经不再适用于签核精度的延时计算。文献[16]仅用驱动点有效电容无法捕捉完整互连线的连接特征,导致其在精度上较低。本文通过机器学习增强使得互连线延时平均相对误差达到了 1.70%,相对于文献[16],本文的互连线延时误差减少了 52.4%,互连线转换时间上误差达到了 0.88%,减少了 65.5%。

表 9 阶段延时计算方法精度对比 单位:%

延时指标	文献[11]	文献[16]	本文
单元延时	5.35	1.63	1.49
单元转换时间	19.5	2.97	3.16
互连线延时	15.7	3.54	1.70
互连线转换时间	17.44	2.57	0.88

运行时间代表阶段延时计算的效率。为了检验不同方法的可扩展性,分别生成并测试 1 000、10 000、100 000 组测试样本,在不同规模上测试并评估算法在处理不同规模数据时的性能,结果如表 10 所示。

本文的方法保持与文献[11]相同数量级的运行时

表 10 运行时间结果对比

阶段数量	运行时间/ms		
	文献[11]	文献[16]	本文方法
1 000	8.7	27.2	12.6
100 00	44.4	146.5	68.3
100 000	341.8	1 239.3	573.7

间. 这是因为本文采用了 Elmore 延时模型, 并辅以机器学习技术, 本文的时间复杂度始终与 Elmore 延时成正比. 各个算法的复杂度在本质上决定了算法的运行时间. 其计算复杂度约为 Elmore 延时, 模型计算复杂度的常数倍, 即  $O(c \times n)$ ,  $c$  为迭代次数,  $n$  为 RC 树的节点数量. 从图 17 中可以发现, 该组样本数量为 10 000, 区分上升沿与下降沿的频次总量为 20 000, 本文提出的迭代过程, 收敛所需的迭代次数在基本在 4 到 5 次之间. 本文在任意阶段数量, 时间开销均低于文献[16]方法, 运行时间减少了 53.7%. 由于插值过程会在每一个阶段延时计算中调用数次, 本文针对基于电流源模型的阶段延时计算设计了基于电压的插值, 相较于其他方法优势明显, 使得整体运行时间缩短, 此外, 也得益于本文算法的低复杂度.

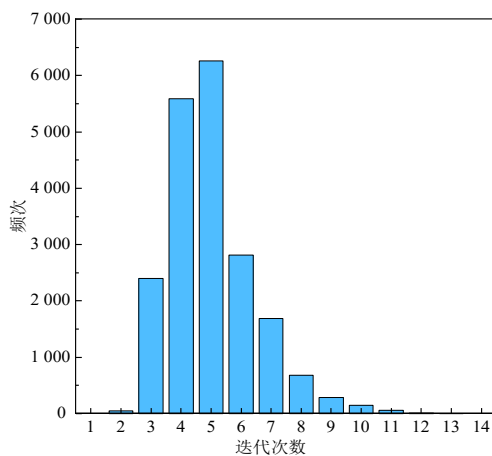


图 17 阶段延时算法的迭代频次

## 5 结论

本文建立了基于电流源模型的阶段延时计算方法, 构建了单元延时的计算框架, 将有效电容改进为动态电容, 引入了一种基于电压的插值方法, 建立了针对阶段的延时计算方案. 通过改进传统 Elmore 延时计算, 将不同类型的寄生网络处理为适合本文提出的迭代方法的 RC 树. 利用一种快速迭代算法进行初步的阶段延时计算并使用机器学习对计算过程中的互连线延时结果进行优化. 应用 ASU ASAP 的 7 nm 工艺对本文方法进行实验验证和对比分析. 对比以  $\Pi$  模型为负载的插值方法和单元延时结果、以分布式 RC 网络为负载的互连

线延时计算和阶段延时计算结果. 以  $\Pi$  模型作为负载时, 本文迭代方法对比文献[11], 单元延时和单元转换时间误差分别减少 16.7% 和 83.1%. 分布式 RC 网络为负载时, 对比文献[16]实现了仅在单元转化时间上误差略微增加, 在单元延时、互连线延时以及互连线转换时间误差分别减少了 9.1%、52.4% 和 65.5%. 同时本文提出的阶段延时迭代方法平均在 4~5 次迭代后达到收敛, 运行时间减少了 53.7%.

## 参考文献

- [1] BHASKER J, CHADHA R. Static Timing Analysis for Nanometer Designs: A Practical Approach[M]. Berlin: Springer Publishing Company, Incorporated, 2009.
- [2] LEVY H J. Segmentation and Interpolation of Current Waveforms: US20090210204[P]. 2009-08-20.
- [3] DARTU F, MENEZES N, QIAN J, et al. A gate-delay model for high-speed CMOS circuits[C]//31st Design Automation Conference. Piscataway: IEEE, 2006: 576-580.
- [4] QIAN J, PULLELA S, PILLAGE L. Modeling the "Effective capacitance" for the RC interconnect of CMOS gates[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1994, 13(12): 1526-1535.
- [5] DARTU F, MENEZES N, PILEGGI L T. Performance computation for precharacterized CMOS gates with RC loads[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 15(5): 544-553.
- [6] Osculating Thevenin model for predicting delay and slew of capacitively characterized cells[C]//Proceedings of the 39th annual Design Automation Conference. New York: ACM, 2002: 866-869.
- [7] ELMORE W C. The transient response of damped linear networks with particular regard to wideband amplifiers[J]. Journal of Applied Physics, 1948, 19(1): 55-63.
- [8] ALPERT C J, DEVGAN A, KASHYAP C. A two moment RC delay metric for performance optimization[C]//Proceedings of the 2000 International Symposium on Physical Design. New York: ACM, 2000: 69-74.
- [9] RATZLAFF C L, PILLAGE L T. RICE: Rapid interconnect circuit evaluation using AWE[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 13(6): 763-776.
- [10] SILVEIRA L M, KAMON M, ELFADEL I, et al. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits[J]. Computer Methods in Applied Mechanics and Engineering, 1999, 169(3/4): 377-389.

- [11] PURI R, KUNG D S, DRUMM A D. Fast and accurate wire delay estimation for physical synthesis of large ASICs[C]//Proceedings of the 12th ACM Great Lakes symposium on VLSI. New York: ACM, 2002: 30-36.
- [12] YU B. Machine learning in EDA: When and how[C]//2023 ACM/IEEE 5th Workshop on Machine Learning for CAD. Piscataway: IEEE, 2023: 1-6.
- [13] REN H X, NATH S, ZHANG Y Q, et al. Why are graph neural networks effective for EDA problems? [C]//Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design. New York: ACM, 2022: 1-8.
- [14] HU J, KAHNG A B. Invited paper: The inevitability of AI infusion into design closure and signoff[C]//2023 IEEE/ACM International Conference on Computer Aided Design. Piscataway: IEEE, 2023: 1-7.
- [15] CHEN T H, ZHANG G L, YU B, et al. Machine learning in advanced IC design: A methodological survey[J]. IEEE Design & Test, 2023, 40(1): 17-33.
- [16] GARYFALLOU D, VAGENAS A, ANTONIADIS C, et al. Leveraging machine learning for gate-level timing estimation using current source models and effective capacitance[C]//Proceedings of the Great Lakes Symposium on VLSI 2022. New York: ACM, 2022: 77-83.
- [17] CHENG H H, JIANG I H, OU O. Fast and accurate wire timing estimation on tree and non-tree net structures[C]//Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference. New York: ACM, 2020: 1-6.
- [18] HE X, FU Z Y, WANG Y, et al. Accurate timing prediction at placement stage with look-ahead RC network[C]//Proceedings of the 59th ACM/IEEE Design Automation Conference. New York: ACM, 2022: 1213-1218.
- [19] HUANG T W, GUO G N, LIN C X, et al. OpenTimer v2: A new parallel incremental timing analysis engine[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 40(4): 776-789.
- [20] HUTCHINSON S, KEITER E, HOEKSTRA R, et al. The Xyce™ parallel electronic simulator-an overview[C]//Parallel Computing. London: Imperial College Press, 2002: 165-172.
- [21] GARYFALLOU D, SIMOGLU S, SKETOPOULOS N, et al. Gate delay estimation with library compatible current source models and effective capacitance[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2021, 29(5): 962-972.

### 作者简介



**郭静静** 女, 1989年1月出生于山西省大同市. 2011年和2014年毕业于西安电子科技大学, 获得微电子学学士学位和集成电路工程硕士学位. 2020年毕业于东南大学, 获得微电子学与固体电子学博士学位. 现任南京邮电大学讲师. 主要研究方向为统计时序分析.  
E-mail: guo625jingjing@njupt.edu.cn



**查佩文** 男, 1998年出生于江苏省南京市, 硕士. 2021年进入南京邮电大学集成电路科学与工程学院. 主要研究方向为静态时序分析、复合电流源模型.  
E-mail: 18851820500@163.com



**张树钢** 男, 2001年出生于山东省滨州市. 2023年进入南京邮电大学集成电路科学与工程学院, 在读硕士研究生. 主要研究方向为静态时序分析.  
E-mail: 1223228018@njupt.edu.cn



**蔡志国** 男, 1983年出生于江苏省连云港市. 2014年毕业于东南大学电子科学与工程学院. 现任南京邮电大学集成电路科学与工程学院教授、博士生导师. 主要研究方向为低功耗集成电路设计与测试. 中国电子学会会员编号: E190019695S.  
E-mail: whczk@njupt.edu.cn