

基于 CNFET 电路段内关键门的全局布局算法

田康林¹⁾, 赵康^{2)*}

¹⁾(北京邮电大学理学院 北京 100876)

²⁾(北京邮电大学集成电路学院 北京 100876)

(zhaokang@bupt.edu.cn)

摘要: 针对传统硅基电路布局算法在碳纳米管(CNT)密度变化的碳纳米管场效应晶体管(CNFET)电路上表现出时序良率不高的问题, 提出一种基于段内关键门的全局布局算法. 首先自底向上逐级分析电路各个层级, 依次建立门延迟、门树延迟模型, 在此基础上结合 CNFET 电路相关矩阵建立包含延迟均值和方差的段统计延迟模型; 然后通过理论分析确定时序良率与段的统计延迟之间的相关关系; 最后利用 CNFET 电路不对称空间相关性, 使用网格搜索策略不断迭代调整段内关键门位置, 以降低段延迟. 在 OpenCores 中 4 个测试电路上的实验结果表明, 所提算法平均提高了 20% 的电路时序良率, 在执行时间上比 CNT 密度变化感知的基准方法降低 25%, 揭示了其在高时序良率要求的大规模电路中应用的潜力.

关键词: 碳纳米管; 碳纳米管场效应晶体管; 不对称空间相关性; 全局布局算法
中图分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024.2023-00070

Global Layout Algorithm Based on Key Gates within Segments of CNFET Circuits

Tian Kanglin¹⁾ and Zhao Kang^{2)*}

¹⁾(School of Science, Beijing University of Posts and Telecommunications, Beijing 100876)

²⁾(School of Integrated Circuits, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: Aiming at the problem that traditional silicon-based circuit layout algorithms show low timing yields on carbon nanotube field effect transistor (CNFET) circuits with varying carbon nanotube (CNT) densities, a global layout algorithm based on the key gates in a segment is proposed. Firstly, we analyze each level of the circuit from the bottom to the top, and establish the gate delay and gate tree delay models, and then we combine the CNFET circuit correlation matrix to establish the statistical delay model of the segment with the mean and variance of the delay; then, we determine the correlation between the timing yield and the statistical delay of the segment through the theoretical analysis; finally, we use the asymmetric spatial correlation of CNFET circuits, and we use the lattice search strategy to continuously adjust the key gates in the segment by iteration. Finally, using the asymmetric spatial correlation of CNFET circuits, a grid search strategy is used to iteratively adjust the positions of key gates in the segment to reduce the segment delay. Experimental results on four OpenCores test circuits show that the proposed algorithm improves the circuit timing yield by 20% on average, and reduces the execution time by 25% compared with the CNT density-aware benchmark method, which reveals its potential application in large-scale circuits with high timing yield requirements.

Key words: CNT; CNFET; asymmetric spatial correlation; global layout algorithm

收稿日期: 2023-05-05; 修回日期: 2023-12-24. 国家重点研发计划资助(2022YFB2901100); 田康林(1999—), 男, 硕士研究生, 主要研究方向为碳基集成电路 EDA 工具设计; 赵康(1982—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为 EDA 电子设计自动化、FPGA 工具、编译优化、体系结构.

1 介绍

在后摩尔时代, 半导体产业面临着巨大的技术挑战和工程困难^[1], 但人类社会对数据计算能力和存储能力的需求却与日俱增. 因此, 半导体学界和业界在艰难发展硅基半导体技术的同时, 也在积极探索新型材料和器件, 以求从根本上延续和拓展摩尔定律. 在众多新型半导体器件中, 碳纳米管场效应晶体管(carbon nanotube field-effect transistors, CNFET)以其优异的电学性能而受到广泛关注^[2-6]. CNFET 器件结构如图 1 所示, 它与传统硅基 MOSFET 器件类似, 不同的是 CNFET 的沟道由一束排列整齐的碳纳米管(carbon nanotube, CNT)组成. 器件的载流子通过 CNT 传输, 从而实现器件的电流控制. 与传统的 MOSFET 器件相比, CNFET 器件具有更高的载流子迁移率和更小的漏电流, 因此可以实现更高的性能要求和更低的功耗^[7].

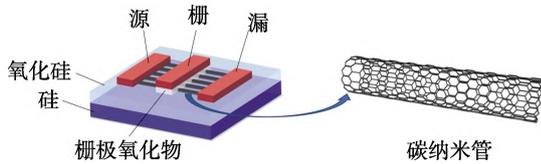


图 1 CNFET 的结构和 CNT

然而, 受 CNT 生长工艺的限制, 发展大规模 CNFET 集成电路面临一些挑战^[8]. 如 CNFET 中金属性 CNT 存在、CNT 阵列不对齐, 以及 CNT 密度变化等. 2022 年, 北京大学团队展示的由维度限制法和自组装技术制备的具有超高半导体性纯度、理论最优密度和良好取向性的晶圆级碳纳米管平行阵列^[8]表明, 最新的 CNT 生长工艺已经能够基本解决金属性 CNT 存在和 CNT 阵列不对齐的问题. 此外, 新墨西哥大学团队的研究表明, CNFET 逻辑门中的 CNT 密度变化所引起的延迟变化百分比 (σ/μ) 是 CNT 平均数量的函数^[9]. 因此, CNT 密度变化将导致电路延迟产生较大变化, 最终影响电路功能良率和时序良率. 综上所述, CNT 密度变化正逐渐成为影响 CNFET 电路性能和稳定性的主要因素.

在如图 2 所示传统集成电路设计过程中, 布局是物理设计中的重要环节之一, 其作用是确定电路元件的位置和相互连接方式, 以实现电路的功能需求和性能要求. 传统布局方法通常是基于硅基电路设计的, 难以适应碳基电路布局. 首先, CNFET 与 MOSFET 存在显著差异. CNFET 具有独

特的电子输运性质. 这意味着在布局方法设计上, 可能需要充分地利用 CNFET 的电学优势, 如高载流子迁移率和导电性能. 更重要的是, CNT 密度变化已成为影响 CNFET 电路可靠性和性能的主要问题, 对其布局算法提出了更高的要求, 并引入了新的布局约束和优化目标. 因此, 需要在 CNFET 电路布局中考虑 CNT 密度变化问题, 如采用统计延迟模型, 以便更准确地建模和优化电路延迟, 提高时序良率.

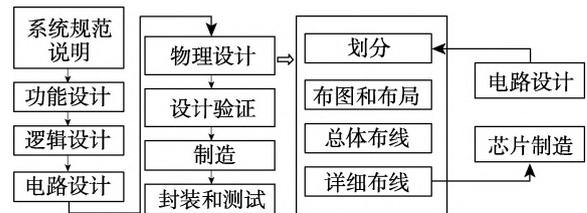


图 2 传统硅基集成电路设计流程

因此, 为了适应 CNFET 的特殊性质和 CNT 密度变化, 需要开发新的布局算法解决这些问题. 在上述背景下, Beste 等^[10]提出了一种名为路径愈合的布局方法, 在详细布局阶段, 通过将每个近关键路径上的门沿着 CNT 生长方向放置到不同列减少该路径的延迟变化. 尽管实验结果表明该方法有效地减少了路径延迟变化, 但是由于它只在详细布局阶段改变了门位置, 因此搜索和优化效果相对有限. 此外, 该方法未考虑互连线寄生效应可能引起的延迟变化, 而这种效应对电路延迟变化影响很大^[11]. Wang 等^[12]提出了一种新颖的变化感知、时序驱动的全局布局方法, 它同时考虑了门延迟和互连线延迟的均值和方差, 进一步提高了电路的时序良率, 但是该方法高昂的时间成本限制了其在大规模电路上应用.

为改进上述研究的不足之处, 本文首先提出了一种段的统计延迟模型. 该模型同时考虑了门延迟和互连线延迟, 并根据 CNFET 电路的相关矩阵^[10]对其进行简化; 在此基础上, 提出了一种基于段内关键门的全局布局算法. 该算法通过不断迭代调整统计关键段内的关键门, 降低段延迟度量值. 实验结果表明, 本文算法可以有效地提高电路时序良率, 较基准方法有明显的执行时间优势.

2 段延迟模型

本节首先介绍从传统 CMOS 电路门延迟模型改进而来的 CNFET 电路门延迟模型, 提出电路的

门树结构,并结合 π 模型和 Elmore 延迟模型推导出门树延迟模型,最后根据段内门树相互独立的假设,进而得到段延迟模型.

在分析复杂电路的延迟时,通常先将电路划分为多个层级,之后逐级进行分析和延迟建模,以有效地减少电路的复杂度,简化分析和计算.因此,本文首先将电路划分为门、门树以及段 3 个层级结构,其中,门树结构是由若干逻辑门组成的树形结构,段结构是由若干门树结构组成的;然后分别对 3 个层级的延迟进行建模,其流程如图 3 所示.

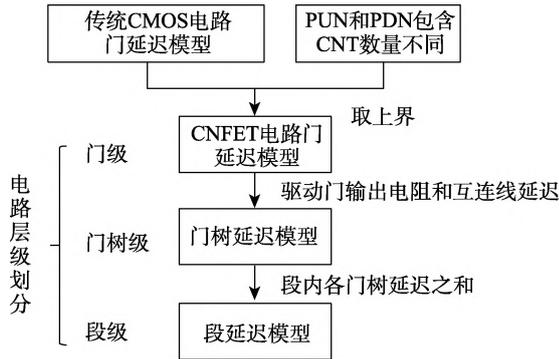


图 3 段延迟建模流程

2.1 门延迟建模

本节先介绍传统 CMOS 电路的门延迟模型,再根据 CNT 密度变化特性,提出门延迟模型在 CNFET 电路中的改进方法.

传统 CMOS 电路的门延迟模型 d_{gate} 如文献 [13] 所述,由门固有延迟 d_0 、门输出电阻 R_{out} 、门输出端负载电压 C_{load} 、常数 k 和输入变化率 S_{in} 决定,即

$$d_{\text{gate}} = d_0 + R_{\text{out}} \cdot C_{\text{load}} + k \cdot S_{\text{in}} \quad (1)$$

根据传统 CMOS 门的输入网络(pull up net, PUN)和输出网络(pull down net, PDN)是否开启,可以将 d_0 和 R_{out} 进一步分为 $d_{0,\text{PUN}}$ 、 $d_{0,\text{PND}}$ 和 $R_{\text{out},\text{PUN}}$ 、 $R_{\text{out},\text{PND}}$. 此外,因为数字电路中常常需要对时序进行分析,所以需要确定一个门延迟上限值来保证电路的功能正确性.可以使用最大值函数(max 函数)来选取这个上限值,即

$$d_{\text{gate}} = \max \{ d_{0,\text{PUN}} + R_{\text{out},\text{PUN}} \cdot C_{\text{load}} + k \cdot S_{\text{in}}, d_{0,\text{PND}} + R_{\text{out},\text{PND}} \cdot C_{\text{load}} + k \cdot S_{\text{in}} \} \quad (2)$$

在传统 CMOS 电路的 PUN 中,延迟主要由

N-MOSFET 导致,而在 PDN 中,延迟主要由 P-MOSFET 导致.由于 N-MOSFET 的导通速度通常比 P-MOSFET 要快,因此 PUN 的延迟通常比 PDN 的延迟小,所以门延迟可以表达为

$$d_{\text{gate}} = d_{0,\text{PDN}} + R_{\text{out},\text{PDN}} \cdot C_{\text{load}} + k \cdot S_{\text{in}}.$$

对于 CNFET 电路,CNT 密度变化可能会导致逻辑门中 PUN 和 PDN 中包含的 CNT 数量不同,无法确定 PUN 和 PDN 哪个驱动能力更强、延迟更小.所以 $d_{0,\text{PDN}} + R_{\text{out},\text{PDN}} \cdot C_{\text{load}}$ 和 $d_{0,\text{PUN}} + R_{\text{out},\text{PUN}} \cdot C_{\text{load}}$ 的大小关系并不固定,但仍可以对式(2)取一个上界.设 $d_{0,\text{max}}$ 为 $d_{0,\text{PUN}}$ 和 $d_{0,\text{PDN}}$ 中的最大值, $R_{\text{out},\text{max}}$ 为 $R_{\text{out},\text{PUN}}$ 和 $R_{\text{out},\text{PDN}}$ 中的最大值,CNFET 电路标准单元门延迟可以表示为

$$d_{\text{gate}} = d_{0,\text{max}} + R_{\text{out},\text{max}} \cdot C_{\text{load}} + k \cdot S_{\text{in}} \quad (3)$$

需要注意的是,在本文使用的 10nm CNFET 电路标准单元库^[12]是根据 Nangate 45nm^①等比例缩放而来的,所以可以应用式(3)对 CNFET 标准单元的门延迟进行建模.式(3)中的各个变量值取决于单元内 CNT 的数量,是每个标准单元的固有值.因此,根据 Stanford 大学 CNFET HSPICE 模型,对每个标准单元的不同 CNT 数量进行了表征,结果存入表中,以供段的统计延迟查找和使用.此外,Wang 等^[12]还提到,标准单元的自加载电容对标准单元的延迟影响包括在本征门延迟 d_0 中; C_{load} 是标准单元输出段的总负载电压,不包含其自负载电容,因此 C_{load} 独立于其驱动单元.

2.2 门树延迟建模

本节将给出 CNFET 电路的门树结构,并通过 π 模型建模为 RC 树,最后通过 Elmore 延迟模型评估 RC 树中各个节点,并推导出门树延迟模型,用于后续的段延迟建模.

门树被定义为电路中驱动门与若干被驱动门互联所构成的树形结构,它描述了门电路中的逻辑关系和信号传输路径.具体而言,电路中的一个驱动门可以看作门树的根节点,被驱动门可以看作叶子节点.各个门之间输入/输出的连接关系形成了树的边,从而构成了一个基于门电路的单父节点的 m 叉树形结构,如图 4 所示.考虑一种特殊情况:若段内的 m 个驱动门同时驱动 n 个门,那么这 n 个门就同时成为 m 个根节点的叶子节点.本文把这种情况分离成 m 个相互独立的门树.

① <http://www.nangate.com>

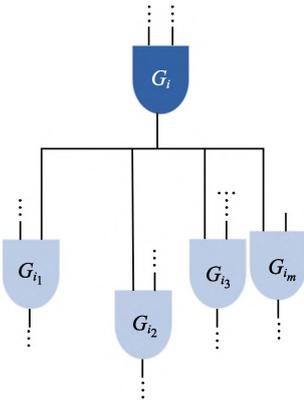


图 4 m 叉门树结构

驱动门和被驱动门之间的互连线用 π 模型表示^[14], 被建模为图 5 所示的 RC 树. 通过将 Elmore 延迟模型应用于 RC 树, 可以得到驱动门输出电阻造成的延迟和互连线延迟. 因此该门树延迟可以被表示为

$$d_{tree,i} = d_{0,max,i} + k_i \cdot S_{in,i} + R_{out,max,i} \cdot \left(\sum_{j=1}^m C_{w,i,i_j} + C_{in,i_j} \right) + R_{w,i,i_j} \cdot \left(\frac{C_{w,i,i_j}}{2} + C_{in,i_j} \right) \quad (4)$$

其中, $d_{0,max,i}$, k_i , $S_{in,i}$ 和 $R_{out,max,i}$ 分别表示 G_i 的固有延迟、输入变化率系数、输入变化率和输出电阻; C_{w,i,i_j} 和 R_{w,i,i_j} 分别表示门 G_i 和 G_{i_j} 之间的互连线寄生电容和电阻, 它们的计算公式为

$$\begin{cases} C_{w,i,i_j} = c_{unit} \cdot L_{i,i_j} \\ R_{w,i,i_j} = r_{unit} \cdot L_{i,i_j} \end{cases}$$

其中, c_{unit} 和 r_{unit} 分别是互连线每单位长度的寄生电容和电阻; L_{i,i_j} 是 G_i 和 G_{i_j} 之间的半周长线.

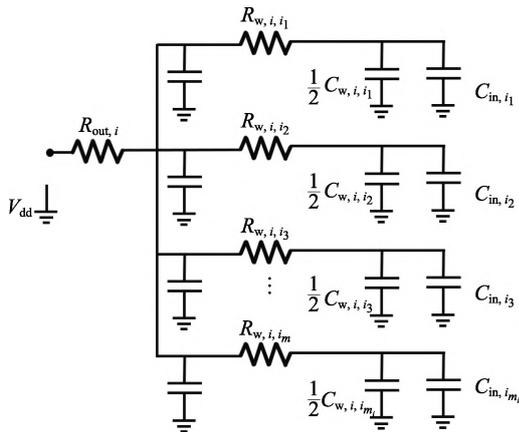


图 5 图 4 所对应的 RC 树

2.3 段延迟建模

本文假设段内门树之间相互没有影响, 本节中

将提出段延迟模型, 用于后续段的统计延迟模型.

段被定义为 CNFET 电路中某条路径的一部分, 该部分包含 n 个门, 每个门驱动段中的一个门树. 于是段延迟可以被定义为段内所有门树延迟总和, 即

$$d_{seg} = \sum_{i=1}^n d_{tree,i} \quad (5)$$

需要注意的是, 这里假设所有门树的延迟相互独立, 即它们之间相互没有影响. 原因是对于本项工作的电路划分和延迟建模方式, 段内门树一般是并行连接的, 这意味着每个门树的延迟不会直接受到其他门树的影响, 因此段内各门树延迟可以看作相互独立. 然而在实际大规模复杂电路中, 门树之间的延迟可能会互相影响, 从统计学的角度来讲, 这种影响往往具有随机性, 难以用建模的方式对延迟进行估计和预测. 因此在这种情况下, 假设各个门树之间的延迟相互独立可以简化模型和计算.

3 段的统计延迟模型

首先论证使用统计延迟的必要性, 并介绍 CNT 数量概率分布和 CNFET 电路中相关矩阵的概念, 最后在此基础上建立 CNFET 电路中段的统计延迟模型.

在 CNT 密度变化的前提下, 段延迟并不是一个确定的值, 然而当 CNT 数量遵循某种概率分布时, 段延迟就可以看作一个随机变量, 其分布的均值和方差就可以通过统计学中的方法进行估计. 因此本文采用段的统计延迟作为算法的优化目标.

3.1 CNT 数量概率分布

本节将给出 CNT 数量近似遵循的概率分布——正态分布, 并给出了 2 个相邻 CNT 之间距离的平均值和标准差.

Zhang 等^[15]研究表明, 对于给定宽度 W , 该区域所覆盖的 CNT 数量 $N(W)$ 近似遵循正态分布

$$N(W) \sim N\left(\frac{W}{\mu_s}, \frac{W \sigma_s^2}{\mu_s^3}\right)$$

其中, μ_s 和 σ_s 为 2 个相邻 CNT 之间距离的平均值和标准差. 然而在去除金属性 CNT 的过程中, 一些半导体性的 CNT 也可能被去除, 但最终关心的是金属性 CNT 去除后剩下的半导体性 CNT 密度分布. 由于金属性 CNT 只占 CNT 总数的很少一部分^[16], 在去除之后半导体性的 CNT 分布几乎不会改变. 所以对于给定宽度 W , 它所覆盖的半导体性

CNT 数量 $N_s(W)$ 也近似遵循正态分布

$$N_s(W) \sim N\left(\frac{W}{\mu_{s,\text{post}}}, \frac{W^2 \sigma_{s,\text{post}}^2}{\mu_{s,\text{post}}^3}\right) \quad (6)$$

其中, $\mu_{s,\text{post}}$ 和 $\sigma_{s,\text{post}}$ 分别为去除金属性 CNT 后 2 个相邻 CNT 之间距离的平均值和标准差. 根据文献[15,17-19]所提供的 CNT 参数, 可以计算得到 $\mu_{s,\text{post}}=4.253 \text{ nm}$, $\sigma_{s,\text{post}}=3.085 \text{ nm}$.

3.2 不对称空间相关性与相关矩阵

下面先介绍 CNFET 电路的不对称空间相关性, 并引出相关矩阵的概念.

CNFET 电路不对称空间相关性如图 6 所示. 如果 2 个 CNFET 平行放置于 CNT 生长方向, 那么这 2 个 CNFET 的沟道共享同一束 CNT, 这意味着它们的电学性质高度相关; 相反, 若 2 个 CNFET 的沟道覆盖不同束的 CNT, 则它们的电学性质是高度独立的.

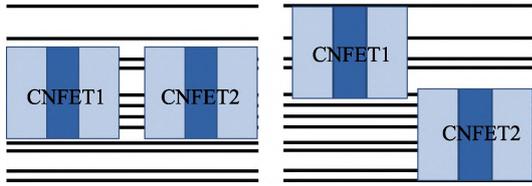


图 6 CNFET 电路不对称空间相关性

由于 CNFET 电路比传统 CMOS 电路具有更大的空间变化, 因此通常用相关矩阵^[10,12]来表征 CNFET 电路的空间变化. 对于一组由 N 个 CNFET 门组成的电路, 相关矩阵可以是一个 $N \times N$ 的矩阵 M . 给定 2 个 CNFET 门 G_i 和 G_j , 如果它们的几何中心属于标准单元布局区域的同一行, 根据不对称空间相关性, 可将这 2 个门视为完全相关, 在相关矩阵中则可以将 M_{ij} 和 M_{ji} 设置为 1, 否则 M_{ij} 和 M_{ji} 被设置为 0. 其中, M_{ij} 和 M_{ji} 分别代表矩阵 M 的第 i 行第 j 列的元素和第 j 行和第 i 列元素.

图 7 展示了一个 CNFET 电路全局布局的示例. 该示例有 6 个 CNFET 门, 其中灰色线是平行的 CNT, 红色虚线则是标准单元布局中的行. 按照上述规则, 则可得到这 6 个门的相关矩阵为

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

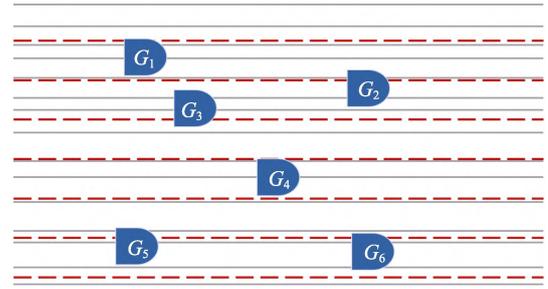


图 7 CNFET 电路布局示例

3.3 段的统计延迟

本节中, 首先将段的统计延迟作为算法的目标函数, 之后依据相关矩阵将段的统计延迟进行化简.

在 CNFET 所覆盖的 CNT 数量近似遵循正态分布的情况下, CNFET 电路的延迟是一个随机变量. 假定电路延迟的均值和标准差为 μ_{ckt} 和 σ_{ckt} , 想要减少电路延迟就应该降低 μ_{ckt} , 想要降低电路延迟变化就应该降低 σ_{ckt} . 因此, 可以把 $\alpha \cdot \mu_{\text{ckt}} + \beta \cdot \sigma_{\text{ckt}}$ 作为布局算法的目标函数, 通过最小化这个目标函数来提高电路的时序良率. 其中, α 和 β 是一个控制目标函数中 μ_{ckt} 和 σ_{ckt} 相对重要性的参数, 通过调整 α 和 β 的值可以平衡 μ_{ckt} 和 σ_{ckt} 在目标函数中的影响. 表 1 展示了在 32 位乘法器上当 $\alpha + \beta = 4$ 时, 各取值对算法延迟优化的影响. 为了方便起见, 本文中所有测试电路选择 $\alpha=1.0$, $\beta=3.0$.

表 1 α 和 β 的取值对算法延迟优化的影响

α, β	延迟优化百分比
$\alpha=0.7, \beta=3.3$	18.343 0
$\alpha=0.8, \beta=3.2$	23.420 4
$\alpha=0.9, \beta=3.1$	28.497 8
$\alpha=1.0, \beta=3.0$	31.257 8
$\alpha=1.1, \beta=2.9$	26.161 0
$\alpha=1.2, \beta=2.8$	21.064 2
$\alpha=1.3, \beta=2.7$	15.967 5

注: 粗体值表示当前列最优值.

但计算 μ_{ckt} 和 σ_{ckt} 需要使用电路中的所有门, 这带来了很大的计算量. 然而, Wang 等^[12]提到, 只优化关键段的延迟就可以有效地提高整个电路的性能. 于是本文选取电路中若干个统计关键段, 将 $\alpha \cdot \mu_{\text{seg}} + \beta \cdot \sigma_{\text{seg}}$ 作为段的统计延迟. 对于每个段可以计算其延迟的均值和方差, 即

$$\mu(d_{\text{seg}}) = \mu\left(\sum_{i=1}^n d_{\text{tree},i}\right) = \sum_{i=1}^n \mu(d_{\text{tree},i}),$$

$$\sigma^2(d_{\text{seg}}) = \sigma^2\left(\sum_{i=1}^n d_{\text{tree},i}\right) = \sum_{i=1}^n \sigma^2(d_{\text{tree},i}) + 2 \sum_{1 \leq i < j \leq n} \sigma(d_{\text{tree},i})\sigma(d_{\text{tree},j})\rho(d_{\text{tree},i}, d_{\text{tree},j}).$$

$$\left\{ \begin{array}{l} \mu_T = \sum_{i=1}^n \mu_i \\ \sigma_T = \left(\sum_{i=1}^n \sigma_i^2\right)^{\frac{1}{2}} \end{array} \right. \quad (7)$$

若对于门 G_i , 定义

$$\mathbf{D} = [d_{\text{tree},1}, d_{\text{tree},2}, d_{\text{tree},3}, \dots, d_{\text{tree},n}],$$

则段延迟的方差可以简化为

$$\sigma^2(d_{\text{seg}}) = \mathbf{DMD}^T,$$

最终可以得到段的统计延迟为

$$\alpha \cdot \mu(d_{\text{seg}}) + \beta \cdot \sigma(d_{\text{seg}}) = \alpha \sum_{i=1}^n \mu(d_{\text{tree},i}) + \beta (\mathbf{DMD}^T)^{\frac{1}{2}}.$$

需要注意的是, 在本项工作中, CNFET 电路段的统计延迟计算依靠每个标准单元随机变量的平均值和标准差. 因此, 本文根据式(6)所示的正态分布随机生成了 10 000 组 CNT 样本. 对于每组样本, 相关数据可通过查表获得, 最后根据这 10 000 组样本计算平均值和标准差.

4 基于段内关键门的全局布局算法

本节介绍基于段内关键门的全局布局算法. 该算法首先选择出一组统计意义上的关键段, 然后对段上的关键门进行位置调整以减少统计延迟.

4.1 时序良率与目标函数

本节将通过理论分析证明电路时序良率与算法目标函数存在相关性.

在电路中, 时序约束是关键的设计要求之一. 对于数字电路而言, 时序良率是衡量电路设计质量的重要指标之一, 它是指在给定的工艺变化下, 电路能够满足时序要求的概率. 虽然本文提出的算法的目标函数(段的统计延迟)不是时序良率, 但是可以通过理论分析和蒙特卡罗模拟^[12]证明它与电路时序良率之间存在关系.

在电路设计中, 段延迟受很多因素的影响, 如器件尺寸变化、工艺变化等, 可以看作多个随机变量的叠加, 且这些随机变量的贡献相互独立. 根据中心极限定理, 这种情况下可以近似看作服从正态分布. 因此, 假设一条时序路径上包含 N 个段, 总延迟为 T (各个段延迟总和), 其中第 i 个段的延迟服从均值为 μ_i , 标准差为 σ_i 的正态分布. 根据中心极限定理, N 足够大时, T 的概率密度函数近似为一个正态分布, 其中时序路径延迟的均值和标准差分别为

因此, 路径的时序良率可以表示为

$$P(T \leq T_{\text{target}}) = \Phi\left(\frac{T_{\text{target}} - \mu_T}{\sigma_T}\right) \quad (8)$$

其中, Φ 是标准正态分布的分布函数, T_{target} 是时序要求的目标值. 根据式(7)(8)可以看出, 通过降低 μ_i 和 σ_i 可以降低 μ_T 和 σ_T 的值, 从而提高路径的时序良率. 又由于段的统计延迟为 $\alpha \cdot \mu_i + \beta \cdot \sigma_i$, 因此, 通过优化段的统计延迟, 可以实现间接提高时序良率的目的.

4.2 整体算法描述

基于段内关键门的全局布局算法描述如下.

算法 1. 基于段内关键门的全局布局算法.

输入. 电路网表信息.

输出. 最终布局结果.

Step1. 执行初始布局.

Step2. 执行力导向二次布局算法, 直到标准单元重叠百分比小于设定阈值.

Step3. 循环优化电路统计延迟, 直到满足终止条件:

Step3.1. 执行统计静态时序分析, 获得电路时序信息;

Step3.2. 选择一组统计关键段集合 S , 对 S 中每一个关键段 s_g , 选择一组段内关键门集合 G , 执行算法 2, 降低段的统计延迟.

Step4. 返回最终布局结果.

本文所提出的布局优化算法主要包含以下几个步骤. 首先, 对输入的电路网表进行一个初始的布局, 以分散标准单元. 然后, 采用力导向二次布局算法^[20], 以减少标准单元之间的重叠, 直到标准单元重叠百分比小于预设的阈值. 接下来, 进入循环优化阶段. 每一次迭代开始前, 执行统计静态时序分析^[11,21], 以获取电路的最新统计时序信息. 然后, 根据该信息选择一组统计临界段^[12], 并针对该组中每个段执行算法 2, 以优化统计延迟. 在迭代过程中, 检查终止条件. 如果当前迭代的统计延迟大于上一次迭代的值, 则返回上一次迭代的布局结果. 如果连续 2 次迭代之间的统计延迟差的绝对值小于预设的阈值 ε (它被定义为最初输入的布局方案的统计延迟的 1/50), 则返回当前迭代的布局结果. 当满足终止条件时, 迭代停止.

4.3 段内关键门位置优化算法

统计关键顶点^[12]是在进行概率统计时具有较

高违规概率的顶点, 本文把与统计关键顶点相关的门称之为段内关键门. 这些顶点是可能位于关键路径上的顶点或是直接影响关键路径的顶点, 因此与这些顶点相关的门对段的统计延迟有着较大影响. 值得注意的是, 除了段内关键门, 其他门也会对段统计延迟做出贡献. 但是, 在优化过程中, 将重心集中在段内关键门上可以提高优化效率和准确性, 因为这些门对段内统计延迟影响最大. 如果优化过程中考虑了所有门, 那么可能会增加计算时间, 同时可能会因为考虑了对段统计延迟贡献较小的门而降低优化效率.

段内关键门位置优化算法描述如下.

算法 2. 段内关键门位置优化算法.

输入. 段 s_g 和关键门集合 G .

输出. 段 s_g 统计延迟最小化后的布局结果.

Step1. 计算段 s_g 的统计延迟 c_{pre} .

Step2. 对 G 中的每一个门 G_i 进行位置调整:

Step2.1. 将 G_i 周围区域划分 8 个搜索子区域, 并行计算 G_i 移动到 8 个子区域中心时, 段 s_g 的统计延迟;

Step2.2. 得到 8 个区域中最小延迟 c_{min} 和其对应区域;

Step2.3. 当 $c_{min} < c_{pre}$ 时, 将 G_i 移动到对应区域中心, $c_{pre} = c_{min}$; 否则继续缩小区域尺寸, 直到小于阈值 h 时结束.

Step3. 返回段 s_g 布局结果.

对于每个关键段, 算法 2 采用了一种基于网格的方法来优化关键门的位置, 以最小化统计延迟. 算法 2 分为 2 个阶段: 第 1 阶段使用并行计算 G_i 移动到每个子区域中心时的统计延迟; 第 2 阶段使用贪心搜索算法找到 G_i 的最优位置.

具体步骤如下: 对于每个关键门 G_i , 首先获取其输入和输出网络的边界框, 并将其周围区域分为 8 个子区域, 子区域的宽度和高度设置为边界框宽度和高度的 $1/3$. 接着, 将门移动到每个子区域的中心, 并计算相应段的统计延迟. 然后选择在 8 个区域中给出最小统计延迟的区域. 如果最小值小于当前段的统计延迟, 则将门移动到所选区域的中心, 并再次搜索新位置周围的 8 个区域, 以寻找更好的位置; 否则, 将区域宽度和高度减半, 并搜索尺寸减小的周围 8 个区域, 以寻找更好的位置. 当网格尺寸小于阈值 h 时, 则停止对该门位置的调整, 开始调整下一个门. 该算法重复这个过程, 直到所有关键门的位置被优化为止.

在本文实验环境中, 并行计算 8 个区域中最小统计延迟通过使用 MATLAB 中的 Parallel Computing Toolbox 来实现. 具体步骤如下.

Step1. 使用 parfor 循环并行计算每个区域的统计延迟. 将不同区域分配给不同的工作进程, 以便同时计算多个区域的统计延迟; 在循环中, 使用索引变量来引用不同的网格.

Step2. 在循环的每个迭代中, 将每个网格的统计延迟存储在一个数组中.

Step3. 在循环结束时, 计算数组中的最小值和最小值的索引. 最小值的索引可以用于确定在哪个网格上获得了最小的统计延迟.

Step4. 将门移动到具有最小统计延迟的区域的中心, 并在新位置周围的 8 个子区域上重复这个过程, 直到找到最佳位置.

从算法描述可以看出, 算法 2 利用段的统计时序信息进行段内门位置的优化, 其比传统的布局算法可以更准确地评估电路的延迟, 并更有效地优化电路时序良率; 同时该算法采用循环优化的方式, 进一步提升了优化效果. 然而任何算法都存在缺点和局限性, 本文算法也不例外. 首先, 该算法采取的是基于阈值的终止条件, 因此容易受到初始布局的影响, 可能导致算法收敛于局部最优解. 此外, 算法在每次循环时都进行统计静态时序分析(statistical static timing analysis, SSTA), 需要消耗大量的计算资源和时间.

5 实验结果

本实验使用一台笔记本电脑作为实验设备, 型号为 Legion R7000p2021H, 搭载 AMD Ryzen 7 5800H 处理器, 主频为 3.20 GHz, 内存大小为 16 GB, 操作系统为 Windows 10. 实验使用 10nm CNFET 标准单元库进行实验, 主要参数如表 2 所示. 实验选择 OpenCores^①的 3 个电路和一个 32 位乘法器作为测试电路, 如表 3 所示, 电路的大小从 349 到 6992 个门不等. 实验使用 MATLAB 2022b 实现了 3 种方法, 方法 A 是变化感知全局布局方法^[12], 方法 B 是路径愈合的方法^[10], 方法 C 是本文的方法. 为公平比较算法的优劣性并减少标准单元的重叠, 3 种方法将 Kraftwerk2 的布局结果作为初始布局, 分别进行调整和优化. 图 8 和图 9 分别展示了 Kraftwerk2 在 4 个测试电路上的布局结果, 以及它们运行时单元重叠尺度随迭代次数变化的趋势, 图 10

① <https://opencores.org>

表 2 标准单元库主要参数

参数	值
CNFET 栅长度 L_g/nm	9.5
CNFET 栅最小宽度/nm	28.8
CNFET 栅最小宽度/nm	240.0
标准单元长度/ μm	0.679
标准单元宽度(相对)	0.884
CNT 直径/nm	1.3

表 3 实验使用的测试电路

编号	电路名称	门数
(a)	adder	349
(b)	M1-ALU	1 096
(c)	32-bit multiplier	2 775
(d)	AES-128	6 992

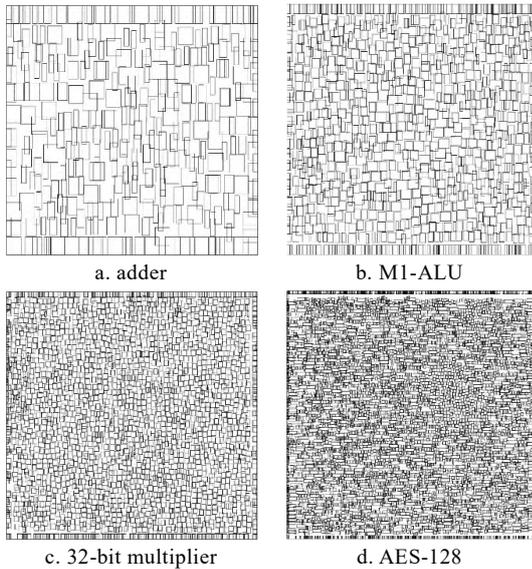


图 8 Kraftwerk2 在 4 个电路上的布局结果

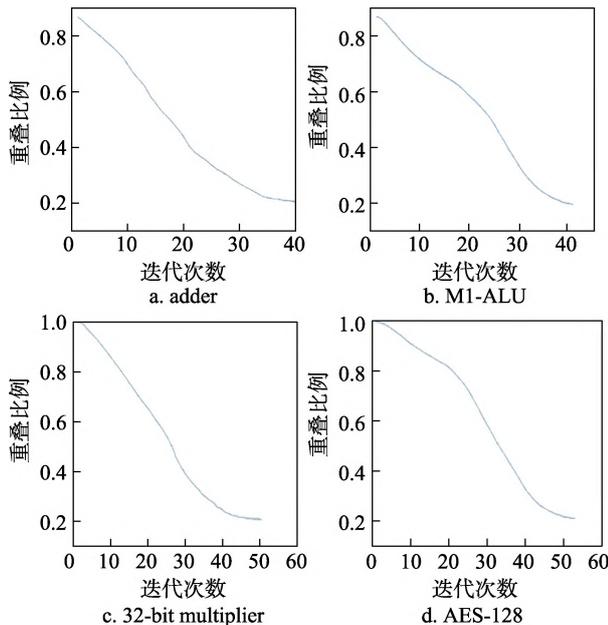


图 9 单元重叠尺度随 Kraftwerk2 迭代次数的变化

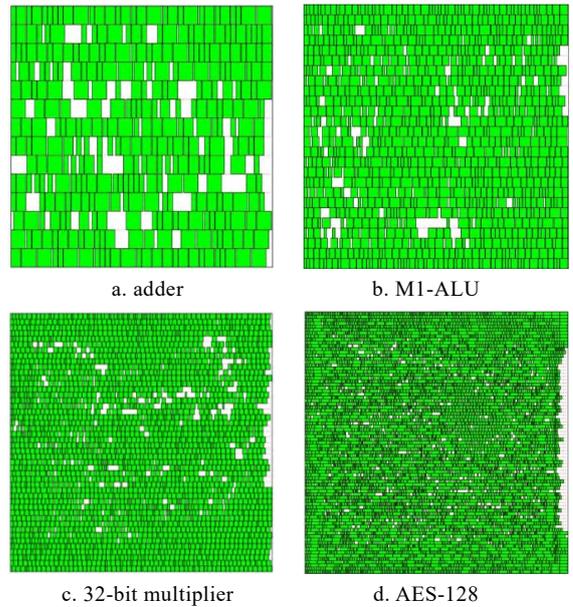


图 10 方法 C 运行后经合法化的最终布局结果

所示为方法 C 在 4 个测试电路上运行后经合法化^[22]的最终布局结果。

实验使用上述 3 种布局算法对电路标准单元进行了布局, 然后使用 SSTA 方法计算了电路的延迟优化结果. 为了得到更可靠的结果, 每个方法都被重复运行了 5 次并将 5 次运行时间的平均值作为算法运行时间, 具体数据如表 4 所示. 从表 4 可以看出, 方法 A 的电路延迟优化效果最好, 但是平均运行时间较长; 方法 C 虽然优化效果略逊于方法 A, 但平均运行时间却大幅缩短, 这在大规模电路下是可以接受的; 方法 B 的电路延迟优化效果最弱, 平均运行时间也较长. 为更好地比较 3 种算法的性能优劣, 本文定义一个新的评价指标——单位时间延迟优化, 它由优化百分比除以平均运行时间得到. 可以发现, 方法 C 在单位时间延迟优化方面表现最好, 而方法 B 表现最差. 综合而言, 本文算法比与对比算法相比, 在运行时间方面表现更出色, 在电路延迟优化效果上表现也不俗. 因此, 表明其具有在实际的 CNFET 电路设计中应用的潜力.

表 4 3 种方法实验结果

电路	延迟优化			平均运行时间/s		
	方法 A	方法 B	方法 C	方法 A	方法 B	方法 C
(a)	12.07	-2.20	10.83	0.64	0.53	0.35
(b)	25.96	-4.60	21.76	3.52	2.51	1.55
(c)	34.97	1.70	31.26	14.94	12.20	7.98
(d)	31.36	3.16	27.36	103.80	85.10	62.90

注: 粗体值表示当前行最优值.

6 结 语

本文提出了一种针对 CNFET 电路的全局布局算法,旨在提高电路的时序良率.该算法以 CNFET 电路的不对称空间相关性为指导准则,以段延迟平均值和标准差组成的统计延迟为目标函数,间接优化了电路的时序良率.实验数据和理论分析表明,本文算法能够有效地提高 CNFET 电路的时序良率,同时在算法运行时间方面表现良好.在未来的工作中,将会对该算法的性能进行更多测试,并探索其在大规模低时序要求电路中的应用.

参考文献(References):

- [1] Iwai H. CMOS technology after reaching the scale limit[C] //Proceedings of the 8th International Workshop on Junction Technology. Los Alamitos: IEEE Computer Society Press, 2008: 1-2
- [2] Dürkop T, Getty S A, Cobas E, *et al.* Extraordinary mobility in semiconducting carbon nanotubes[J]. Nano Letters, 2004, 4(1): 35-39
- [3] Purewal M S, Hong B H, Ravi A, *et al.* Scaling of resistance and electron mean free path of single-walled carbon nanotubes[J]. Physical Review Letters, 2007, 98(18): Article No.186808
- [4] Xu L, Qiu C G, Zhao C Y, *et al.* Insight into ballisticity of room-temperature carrier transport in carbon nanotube field-effect transistors[J]. IEEE Transactions on Electron Devices, 2019, 66(8): 3535-3540
- [5] Lin Y X, Liang S B, Xu L, *et al.* Enhancement-mode field-effect transistors and high-speed integrated circuits based on aligned carbon nanotube films[J]. Advanced Functional Materials, 2022, 32(11): Article No.2104539
- [6] Ding L, Liang S B, Pei T, *et al.* Carbon nanotube based ultra-low voltage integrated circuits: scaling down to 0.4 V[J]. Applied Physics Letters, 2012, 100(26): Article No.263116
- [7] Qiu C G, Zhang Z Y, Xiao M M, *et al.* Scaling carbon nanotube complementary transistors to 5-nm gate lengths[J]. Science, 2017, 355(6322): 271-276
- [8] Liu Yifan, Zhang Zhiyong. Carbon based electronic technology in post-Moore era: progress, applications and challenges[J]. Acta Physica Sinica, 2022, 71(6): Article No.068503(in Chinese)
(刘一凡, 张志勇. 后摩尔时代的碳基电子技术: 进展、应用与挑战[J]. 物理学报, 2022, 71(6): Article No.068503)
- [9] Ghavami B, Raji M, Pedram H. Timing yield estimation of carbon nanotube-based digital circuits in the presence of nanotube density variation and metallic-nanotubes[C] //Proceedings of the 2011 12th International Symposium on Quality Electronic Design. Los Alamitos: IEEE Computer Society Press, 2011: 1-8
- [10] Beste M, Kiamehr S, Tahoori M B. Layout-aware delay variation optimization for CNTFET-based circuits[C] //Proceedings of the 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems. Los Alamitos: IEEE Computer Society Press, 2014: 393-398
- [11] Chang H L, Sapatnekar S S. Statistical timing analysis considering spatial correlations using a single PERT-like traversal[C] //Proceedings of International Conference on Computer Aided Design. Los Alamitos: IEEE Computer Society Press, 2003: 621-625
- [12] Wang C, Sun Y A, Hu S Y, *et al.* Variation-aware global placement for improving timing-yield of carbon-nanotube field effect transistor circuit[J]. ACM Transactions on Design Automation of Electronic Systems, 2018, 23(4): Article No.44
- [13] Chowdhary A, Rajagopal K, Venkatesan S, *et al.* How accurately can we model timing in a placement engine?[C] //Proceedings of the 42nd Annual Design Automation Conference. New York: ACM Press, 2005: 801-806
- [14] Weste N H E, Harris D. CMOS VLSI design: a circuits and systems perspective[M]. 4th ed. Bangalore: Pearson, 2015
- [15] Zhang J, Patil N, Hazeghi A, *et al.* Carbon nanotube circuits in the presence of carbon nanotube density variations[C] //Proceedings of the 46th Annual Design Automation Conference. New York: ACM Press, 2009: 71-76
- [16] Shahi A A M, Zarkesh-Ha P. Prediction of gate delay variation for CNFET under CNT density variation[C] //Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems. Los Alamitos: IEEE Computer Society Press, 2012: 140-145
- [17] Patil N, Deng J, Mitra S, *et al.* Circuit-level performance benchmarking and scalability analysis of carbon nanotube transistor circuits[J]. IEEE Transactions on Nanotechnology, 2009, 8(1): 37-45
- [18] Park H, Afzali A, Han S J, *et al.* High-density integration of carbon nanotubes via chemical self-assembly[J]. Nature Nanotechnology, 2012, 7(12): 787-791
- [19] Patil N, Lin A, Zhang J, *et al.* VMR: VLSI-compatible metallic carbon nanotube removal for imperfection-immune cascaded multi-stage digital logic circuits using carbon nanotube FETs[C] //Proceedings of IEEE International Electron Devices Meeting. Los Alamitos: IEEE Computer Society Press, 2009: 1-4
- [20] Spindler P, Schlichtmann U, Johannes F M. Kraftwerk2—a fast force-directed quadratic placement approach using an accurate net model[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(8): 1398-1411.
- [21] Chang H L, Sapatnekar S S. Statistical timing analysis under spatial correlations[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(9): 1467-1482
- [22] Puget J C, Flach G, Reis R, *et al.* Jezz: an effective legalization algorithm for minimum displacement[C] //Proceedings of the 2015 28th Symposium on Integrated Circuits and Systems Design. Los Alamitos: IEEE Computer Society Press, 2015: 1-5