使用模拟退火在三维现场可编程门阵列上实现数字电路

Hemin Rahimi，库尔德斯坦大学电气工程系，伊朗萨南达季，hemn.rahimi@uok.ac.ir

Hadi Jahanirad，库尔德斯坦大学电气工程系，伊朗萨南达季，h.jahanirad@uok.ac.ir

摘要—三维现场可编程门阵列近期作为现场可编程门阵列系列的下一代产品问世，旨在持续实现更多晶体管在单个芯片上的无缝集成。本文提出了一套完整的计算机辅助设计流程，用于在三维现场可编程门阵列上实现任意逻辑电路。该流程的划分和布局阶段基于模拟退火算法。此外，布线阶段采用了经过修改的 Pathfinder 算法。仿真结果表明，与二维现场可编程门阵列相比，三维现场可编程门阵列（包括双层）的电路速度提高了28.66%，最小通道宽度降低了29.92%，而总面积增加了8.86%。最后，三维现场可编程门阵列中双层与四层的比较结果显示，四层结构的电路速度和最小通道宽度分别提高了15.95%和15.92%。与此同时，总面积仅增加了1.96%。

关键词—三维现场可编程门阵列、划分、布局布线、模拟退火

I. 引言

传统二维集成电路的实现已达到其能力上限，而当前技术在速度、面积、功耗和可靠性方面也面临新的挑战 [3-5]。此外，由于晶体管特征尺寸缩小受到物理限制，摩尔定律 [2] 已不再适用。三维集成电路是一种克服这些问题的有前景的技术。三维集成的主要思路是堆叠裸片，并通过垂直互连将它们连接起来，这些互连称为 TSV（硅通孔）。与传统二维集成相比，三维集成的潜在优势包括降低功耗、延迟、集成电路占用面积和代价 [6]。

现场可编程门阵列（FPGA）凭借其可重配置性，已广泛应用于各种工程系统的电子电路板中。然而，采用当前技术（传统二维集成）的专用集成电路与现场可编程门阵列之间差距较大（逻辑密度效率低十倍以上、延迟差三倍以上、功耗高三倍以上）[7]。造成这一巨大差距的最主要原因是 FPGA 中的可编程布线元件，它们使用户能够在 FPGA 上实现所需电路，从而实现可重配置。另一方面，在三维现场可编程门阵列中堆叠多个裸片后，布线网络中将增加一种新的连接类型：垂直互连。

TSV 间距远大于其他导线和逻辑门的宽度。因此，必须限制 TSV 的数量，以控制相关的面积开销。另一方面，TSV 能非常高效地传导滞留在各层之间的热量，因此集成电路设计人员必须在芯片内布置足够数量的 TSV。

本文提出了一种高效的计算机辅助设计（CAD）工具，用于在三维现场可编程门阵列上实现逻辑电路。该 CAD 工具的主要部分包括划分、布局和布线。在划分阶段，逻辑电路会被划分为若干部分，每部分实现在一个层上。随后，各部分在单个层上进行布局和布线，而分区之间的互连则通过 TSV 实现。我们开发的工具借鉴了通用布局布线流程（VTR），该流程用于在二维 FPGA[8]上实现电路。 [1] 中的作者为 FPGA 提出了一种三维布局布线（TPR）CAD 流程，其中采用 hMetis [9] 进行电路划分。TPR 对所有开关盒采用对称的三维架构，这意味着 x 轴、y 轴和 z 轴上的布线轨道数量相同。z 轴上的布线轨道通过 TSV 实现，其尺寸远大于 x 轴和 y 轴上的布线轨道。另一方面，实际中只有少量 TSV 用于实现垂直互连。因此，采用三维开关盒会导致较大的面积开销，并因制造大量不必要的 TSV 而产生高昂代价。这一特点使芯片在代价、面积、延迟和功耗方面都无法达到最优。此外，TPR 布线器不支持时序驱动。

模拟退火（SA）是一种重要的元启发式算法，可用于在庞大的搜索空间和非线性问题中逼近全局优化 [10]。模拟退火算法已被高效地用于解决一些重要问题，例如旅行商问题 [11]，、装箱问题 [12]，、时间表问题 [13]，和供应链管理 [14]。由于数字集成电路需要实现非常复杂的电路，该领域的典型优化问题涉及极其庞大的搜索空间，因此模拟退火是解决这类问题的有效工具。在集成电路设计领域，有许多学术研究运用了模拟退火 [15-19]。

在我们提出的 CAD 工具中，我们在划分和布局阶段应用模拟退火算法，分别寻找最优分区，并将这些分区最优地放置在三维现场可编程门阵列的各层上。此外，我们提出的流程中的布线阶段采用改进的 Pathfinder布线算法完成。Pathfinder [22] 采用迭代算法，逐步收敛到一个所有信号均已完成布线的可行解。该算法已用于相关的商业（Altera、Xilinx）和学术 CAD 工具 [1], [8], [21], [22]。

本文其余部分安排如下：第二节介绍本文的预备知识。第三节介绍我们提出的流程，第四节给出实验结果。最后，第五节总结本工作的相关结论。

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©2020年 IEEE

II. 预备知识

本节回顾二维 FPGA 架构的基本概念，以及三维现场可编程门阵列中出现的新概念，例如三维开关盒和硅通孔。随后，我们将定义三维现场可编程门阵列中与划分、布局和布线阶段相关的优化问题。

A. 二维现场可编程门阵列：架构与组件

图1展示了二维现场可编程门阵列的俯视图

这种架构称为网格架构。其主要组件包括规则网格上实现的可配置逻辑块（CLB）、开关盒、连接块和 I/O 焊盘。下面简要介绍这些组件各自的结构。

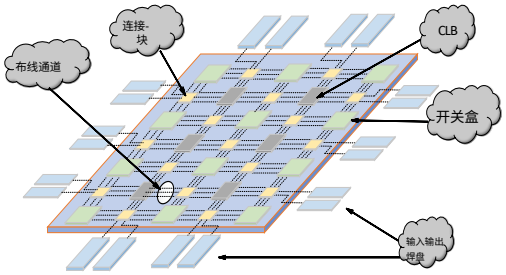


图1. 典型网格架构二维结构及其主要组件

逻辑电路的逻辑功能由可配置逻辑块实现。每个可配置逻辑块都包含一组基本逻辑单元，其本地布线资源将这些单元的输入/输出引脚连接到相应的可配置逻辑块端口。一个基本逻辑单元由一个查找表和一个D触发器组成。图2展示了典型的基本逻辑单元架构。

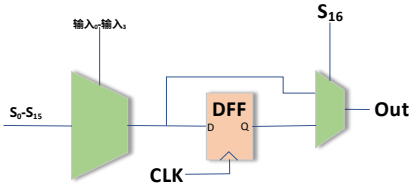


图 2. 典型的 BLE 结构（4 输入查找表）

布线网络由布线轨道、连接块（连接块）、开关盒（开关盒）和布线通道组成。每个 CLB 的输入端口和输出端口通过相邻的连接块连接到邻近的布线通道。另一方面，开关盒插入两条水平通道和垂直通道的交叉处，因此，属于这些相邻布线通道之一的布线轨道可以通过开关盒 [21] 与其他通道的导线相连。

图 3a 展示了一个二维开关盒及其相邻的布线通道。F_S 和可编程互连点是两个相关概念，定义如下：开关盒的灵活度（Fs）是与一条输入轨道相连的输出轨道总数，而可编程互连点表示开关盒所有侧面的全部引脚数之和。图 3a 中的二维开关盒具有 $F_s = 3$ 和可编程互连点=16。

为了比较二维开关盒与三维开关盒，假设可编程互连点

每侧的通道宽度为 W_{min}（最小通道宽度），并使用传输晶体管连接相邻布线通道的布线轨道。二维和三维开关盒中的开关（传输晶体管）总数分别为 $W_{min} \times (3/2)$ 和 $W_{min} \times (5/2)$ [22]。

此外，在开关盒的上表面和下表面制造 W_{min} 个 TSV 会产生显著的面积开销。

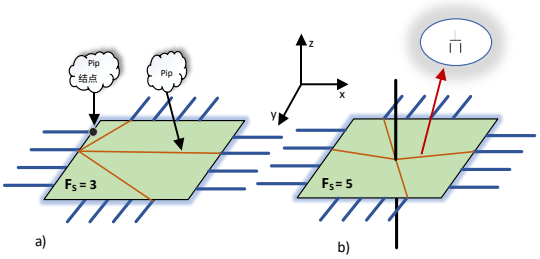


图 3. FPGA 开关盒概览：a）二维开关盒，b）三维开关盒

B. 三维现场可编程门阵列结构

图 4.b 展示了一个由四个有源层构成的堆叠式三维 FPGA 架构概览，相邻层彼此连接。如图 4.b 所示，在三维堆叠模型中，有源层之间的互连通过称为硅通孔（TSV）的垂直连接实现。这一做法的首要直接效果是有望缩短关键路径长度。为更好地理解这一点，假设图 4.a 中二维芯片的曼哈顿距离为 2X。现在，将该芯片划分为四个相等的层，并逐层堆叠，从而形成一块四层芯片（图 4.b）。

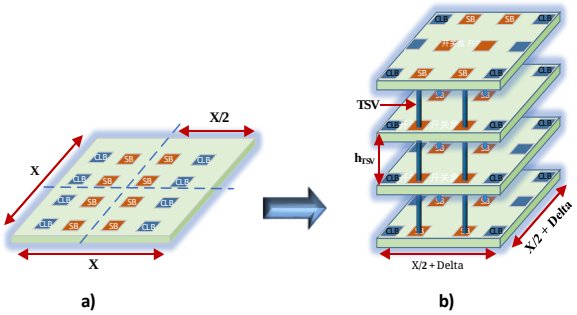


图 4. a）二维 FPGA 结构，b）三维现场可编程门阵列结构

设 h_{TSV} 为硅通孔高度（将在后续小节中说明），芯片尺寸（长和宽）为 X/*2* + Delta，其中 Delta 是一个与硅通孔面积开销有关的参数。因此，最大曼哈顿距离（D_{max}）为：

$$D_{max} = 2 \times (X/2 + Delta) + 3.h_{TSV} \quad (1)$$

In general, maximum Manhattan distance in a 是有 n 层、芯片尺寸为 X 和 Y（长和宽）的三维芯片中的距离为：

$$D_{max} = (X + Y) + (n - 1).h_{TSV} \quad (2)$$

这里假设硅通孔导致的面积开销（Delta）已计入 X 和 Y。根据公式（2），不出所料，我们可以得出结论：在三维集成中，关键路径和布线长度都会缩短，其直接效果是降低芯片延迟。此外，芯片占用的面积减小，逻辑密度提高。

C. 硅通孔（TSV）

与标准键合线相比，硅通孔的功率损耗和电路延迟更低，带宽更高[23]。近期研究聚焦于设计可靠且经济高效的硅通孔。影响硅通孔架构设计的主要因素包括填充材料（用于填充硅通孔的材料）、高度、直径和形状。

Cooper

（Cu）、钨和多晶硅是最常见的填充材料。在现代设计中，硅通孔可以采用圆形、环形、正方形、锥形或矩形形状；目前面临的一项关键挑战是确定哪种形状最适用，以及适用于何种用途。最后，为了概览硅通孔的重要尺寸，可参考 ITRS 2015 的硅通孔路线图。根据该路线图，在 2015—2018年期间，硅通孔直径为 2-4μm，硅通孔间距为 4-8μm，硅通孔高度为 30-50μm。最终，必须有效控制三维FPGA芯片中硅通孔的数量，尽可能降低相关面积开销。

D. 划分、布局与布线问题

在电路实现的初始步骤中，会生成电路图，其中门和互连分别转换为顶点和边。假设电路图表示为 $G = (V, E)$，其中 $V = \{v_1, v_2, ..., v_n\}$ 是顶点集合，$E = \{e_1, e_2, ..., e_n\}$ 是边集合。在划分问题中，我们要将 V 划分为 k 个不同的子集 $V_1, V_2, V_3 \ ... \ V_k$；其中：

$$|V_1| = |V_2| = \ldots = |V_k|, \ V_i \cap V_j = \Phi, \quad i \neq j$$
$$\cup_{i=1}^{k} V_i = V$$

图划分有多种解法，但通常使用基于割大小的代价函数来获得最优解。对于得到的 k 个分区，割大小是连接属于两个不同分区的两个顶点的边数。在我们关于三维现场可编程门阵列的问题中，每个分区由将在某一层上实现的门组成。因此，割大小大致代表所需 TSV 的数量。由此，降低基于割大小的代价函数可减少所需的 TSV 数量。本文提出一种基于模拟退火的划分算法，该算法高效且可扩展。需要注意的是，对于当今规模极大的集成电路，必须实现可接受的划分运行时间。

划分步骤之后的后续阶段是布局和布线。电路中各门的最佳位置在

布局阶段。文献中广泛使用的布局方法可分为时序驱动布局（即重点在于最大化电路速度）、线长驱动布局（最小化整体电路布线）和可布线性驱动布局（平衡布线密度与拥塞控制）。如第一节所述，现场可编程门阵列中最流行且应用最广泛的算法是 Pathfinder [21]。布线阶段在很大程度上取决于前一步的布局效果。换言之，就面积和性能而言，布局的整体质量主要在布局阶段决定 [24]。在本文中，我们针对三维现场可编程门阵列的布局开发了基于模拟退火的方法，并提出了一种改进的布线算法。

III. 提出的方法流程

我们所提出方法的整体流程如图 5 所示。三维现场可编程门阵列架构以 .xml文件的形式输入该流程。此外，以 Verilog硬件描述语言描述的电路也是该流程的另一项输入。

第一步，根据架构文件对电路的 Verilog描述进行综合、优化和技术映射。我们使用 odin_II [25] 和 ABC [26] 完成这一关键步骤。此步骤的结果是以伯克利逻辑交换格式（BLIF）文件格式表示的电路门级描述。第二步，将 BLIF 文件划分为若干部分，部分数量与三维现场可编程门阵列的层数相同。最后，根据所提出的三维架构执行适当的布局和布线。

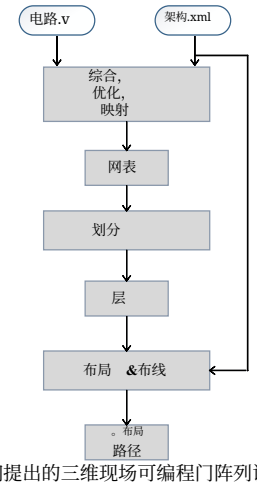


图 5. 我们提出的三维现场可编程门阵列设计流程

A. 提出的三维架构 我们提出的三维架构

包含二维和三维开关盒的混合结构。三维开关盒的数量约占所有可编程开关的 33%，并且该模式在所有层中重复出现。连接块和可配置逻辑块（包含一个大小为 K = 6 的查找表）也都是完全二维的模块，同时假设线段长度为单倍、双倍和四倍。垂直通道宽度和水平通道宽度（W_{min}）由布线器适当选择（布设该电路所需的最小通道宽度）。硅通孔呈正方形，其相关模型参数根据 [6] 和 [27],所示于表 I 中。

表 I. TSV 参数

电阻	0.35 Ω
电容	3 fF
直径	2 μm
间距	4 μm
高度	20 μm

B. 划分算法

三维现场可编程门阵列上的电路实现，第一步是进行划分，确定每一层需要放置和布线的电路门。正如前文所述，各分区之间的互连通过硅通孔实现。在我们提出的三维现场可编程门阵列架构中，硅通孔受到限制，以减少相关的面积开销。因此，我们选用割大小作为所提出划分算法的代价函数。该代价函数可使硅通孔利用率达到最小。此外，我们提出的算法会均衡各分区的大小，也就是说，每个分区包含的门数大致相等。我们提出的算法（算法 I）基于模拟退火，能够妥善处理运行时间和最小割大小问题。

算法I 将电路图（G (V, E)）作为输入，并输出一种优化划分，其中包含 *n* 个子电路图。首先，将电路图随机划分为 *n* 个大小相等的子图（第 1 行）。然后，根据 *Setup()* 函数构建初始增益。图顶点的增益根据公式（3）计算，其中E_i 和 I_i 分别为外部代价和外部代价。

$$D_i = E_i - I_i \quad (3)$$

层 j 的图中与顶点 (v) 相连的边分为外部边和内部边。v 与同一层中任意顶点之间的连接由内部边实现。另一方面，外部边将该顶点 (v) 与其他层中的顶点相连。顶点的外部边和内部边的数量分别对应外部代价和内部代价。

在算法的主循环中（第 6–28 行），对于每个温度 (T)，执行 N 次局部移动。每次局部移动中，先选取来自不同层的两个顶点，然后交换这两个顶点，并计算该配置的新代价函数。若新配置降低了代价，则接受此次局部移动；否则，采用基本模拟退火算法的方法决定接受还是拒绝此次局部移动（第 16–20 行）。相关局部参数（α and N）通过 Setting local parameters() 函数设置；此外，每次局部移动所选顶点对也通过*V_Selection(S')* 函数确定。

主循环将持续重复，直到代价函数出现停滞，或 T 达到其最小值 (Tmin)。当代价函数的值变化不足，或在过多次局部移动中保持不变时，即视为发生停滞。

每次迭代中，温度都会根据冷却计划更新。冷却计划如公式 4 所示。

$$T_{(k+1)} = T_{(k)} \cdot \alpha \quad (4)$$

其中，k 为迭代次数，α 为冷却系数（$0 < \alpha < 1$）。

在此算法中，代价函数如式 (5) 所示。

$$Cost = (I_i + I_j) - (E_i + E_j) + (2 \cdot C_{ij}) \quad (5)$$

其中，I_i 和 I_j 为内部代价，E_i 和 E_j 分别为 v_i 和 v_j 顶点的外部代价，C_{ij} 为这些顶点之间直接互连的数量。

采用几何冷却计划时，较高的初始温度并不能显著改善优化过程。另一方面，若赋予其较低的值，算法可能无法达到最优解。在我们的算法中，初始温度 (T_0) 为初始解的割大小。

此外，T_{min} 设置为 0.5，α 为：

$$\alpha = \frac{Cost_final}{Cost_initial}$$

where, *Cost_initial* 是每个温度（T）下首次接受的局部代价，当 ΔCost < 0 时。此外，Cost_final 是每个温度下最后接受的局部代价，当 ΔCos < 0 时。

算法 I. 基于模拟退火的划分

```
输入：G (V, E)
确保：将图优化划分为 l 层

1：随机划分( )
2: 设置( )
3：设置参数（T，Tmin）
4：S←Sinit
5：代价当前 ← TSV 数量
6：当 T > Tmin 时
7:    设置局部参数（alpha，N）
8: 对 i = 0: N
9:      V_Selection(置位端' )
10: 计算代价函数
11: 计算 Δ代价
12: 如果 (Δ代价 < 0)
13：置位端 置位端'      // 正在更新
14 代价当前 ←Cost当前 + Δ代价
15：否则
16：r 生成随机数 (0,1)
17：如果 (r < exp(-Δ代价/T))
18：置位端 S'←// 更新
19：代价Current ← 代价current + Δ代价
20：结束 if
21：结束 if
22: 结束循环
23:  如果 (停顿)
24:      跳出循环;
25 否则
26：T = G（T，K）
27: 结束条件判断
28：结束 while 循环
```

C. 布局

布局问题定义为：为每个顶点 V_i 分配一个位置，使每个位置上不超过一个顶点 [24]。越精确、越优的布局，电路布线质量也越好。

布局是一个 NP 难问题，因此，CAD 工具采用元启发式方法来完成这一步骤 [28]。本文提出了一种用于三维现场可编程门阵列布局阶段的基于模拟退火的方法。在该算法中，首先生成随机布局（针对每一层），然后针对每个温度执行固定次数的局部移动（交换两个随机选取的门的位置）。局部移动的接受或拒绝方式与基本模拟退火算法类似。

D. 布线

在最后一步中，利用布线资源连接所需的门。与 VTR 和 TPR CAD 工具类似，我们采用 Pathfinder 算法完成布线过程。三维现场可编程门阵列布线中需要妥善处理的主要问题是如何将垂直互连纳入 Pathfinder 算法。与仅在布线步骤中最小化布线长度的 TPR 不同，我们提出的算法同时以时序和布线长度为驱动。

在所提出的布线算法中，针对每个网络（一个源端及其相关汇端），构建一个有向图。G (V, E) 中的顶点集合 V 表示内部模块（逻辑单元、I/O 焊盘等）的 I/O 端子以及布线区域中的布线。此外，一对顶点之间的边表示相关模块之间的路径。

在我们提出的布线算法中，第一次迭代时，所有网络依次使用 Dijkstra 最短路径算法进行布线。此次迭代后，部分布线资源被

多个网络共享。 在后续迭代中，将拆除相关网络的布线并重新布线，以解决资源共享问题。

IV. 结果

在本节中，我们使用 MCNC 基准测试电路进行实验。此外，我们使用 C++ 语言在配备四个 CPU 核心（2.9 GHz）和 8GB 随机存取存储器的 Linux（Ubuntu）平台上实现了所开发的算法。二维现场可编程门阵列和三维现场可编程门阵列的参数（如时序参数和布局）是基于 22 纳米 PTM 技术 [29] 通过 SPICE 仿真提取的。

文献 [30] 的作者针对双层三维现场可编程门阵列开发了基于划分的布局方法（Parti-SA）。我们从硅通孔总数方面将所提出的划分算法与 Parti-SA 进行了比较。如图 6 所示，我们提出的算法效率更高，其质量（以硅通孔减少量衡量）提高了 9.89%。

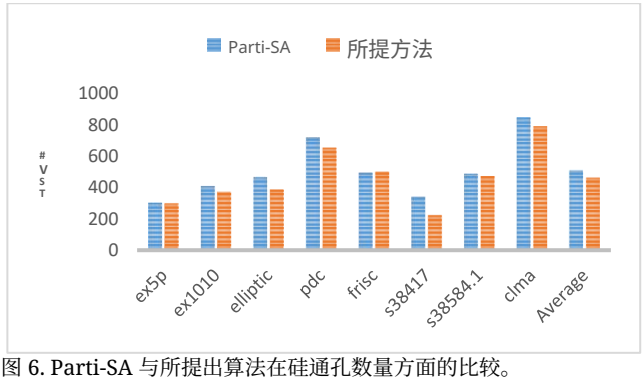


图 6. Parti-SA 与所提出算法在硅通孔数量方面的比较。

表 II 从硅通孔数量、电路速度和总布线长度方面对比了我们的算法与 TPR CAD 工具。

如（最后一行）所示，TSV 数量减少了 5.34%，电路速度提升了 31.14%，总布线长度减少了 5.16%。

表 III 展示了二维设计（VPR 8.0）的仿真结果与我们用于三维设计的 CAD 流程结果之间的对比。如“改进”行所示，三维设计的关键路径延迟（CPD）改善了 28.66%。三维现场可编程门阵列中的布线轨道总数（W_{min}）平均减少了 29.92%。此外，与二维设计相比，三维设计增加的晶体管平均数量仅为 8.86%。表 III 中的面积项是所有层的总面积。我们在综合阶段（Vivado Xilinx）将我们的方法（采用 2 层）与 Virtex-7（xc7vx1140tflg1930-1）进行电路速度对比。我们的方法使电路速度提升了 26.72%。

在图 7 中，我们比较了双层和四层三维现场可编程门阵列的关键路径延迟。此外，我们还比较了最小通道宽度（图 8）和晶体管数量（图 9）这两个参数。如相关图表和表 4 所示，四层芯片的关键路径延迟相比双层芯片降低了 15.95%，通道宽度则降低了 15.92%。在这一转换中，四层芯片的面积仅增加了 1.96%，与二维和双层三维现场可编程门阵列之间的差距相比，这一增幅可以忽略不计。

表 II. TPR 与所提出的 CAD 流程的比较
比较指标为 TSV 数量、关键路径延迟和总布线长度（WL）

	TPR	所提出的方法
电路	TSV 数量 延迟 布线长度	TSV 数量 延迟 布线长度
ex5p	116 4.26 12361	109 2.58 11657
misex3	175 3.82 12894	153 4.31 11561
apex2	192 5.71 18653	168 3.92 17319
apex4	182 5.43 17069	146 5.01 18235
alu4	124 5.67 11364	109 2.68 9987
seq	203 4.92 17246	172 3.62 16023
des	126 3.09 19143	112 1.93 17462
pdc	526 6.88 73649	542 4.07 71482
ex1010	340 6.34 56173	314 4.02 54320
elliptic	391 7.58 38195	409 3.84 36412
clma	637 7.96 103742	617 5.79 97378
平均值	273.82 5.51 34589	259.18 3.79 32894
改进（%）		**+5.34 + 31.14 +5.16**

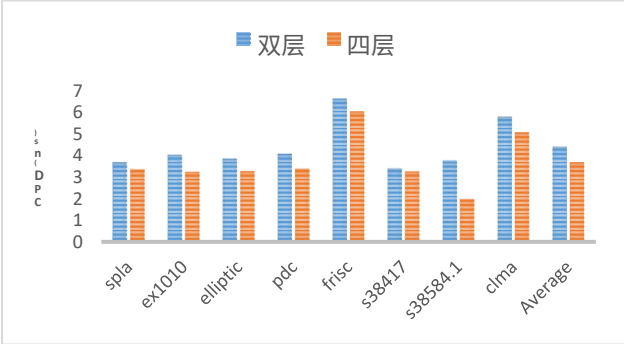


图 7. 双层与四层之间的电路延迟比较

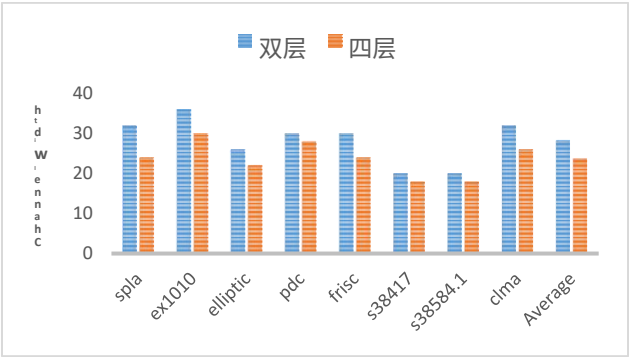


图 8. 双层与四层之间布线通道中的布线轨道数比较

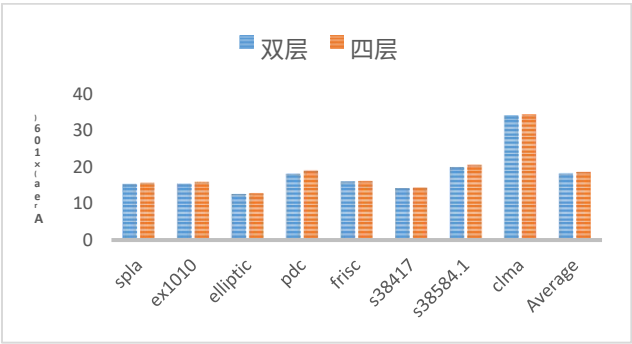


图 9. 双层与四层之间的晶体管数量比较

V. 结论

本文开发了一套完整的流程，用于在三维现场可编程门阵列上实现逻辑电路。结果表明，与二维现场可编程门阵列相比，双层三维现场可编程门阵列的电路延迟和最小通道宽度等重要指标分别改善了 28.66% 和 29.92%。与此同时，总面积（所有层）仅增加了 8.86%。此外，对双层和四层三维现场可编程门阵列进行比较的结果表明，电路速度提高了 15.95%，通道宽度降低了 15.92%，而总面积的增加可以忽略不计。我们的结果表明，四层三维现场可编程门阵列优于双层三维现场可编程门阵列。

表 III. 二维与三维现场可编程门阵列的比较

电路		ex5p	apex2	alu4	seq	ex1010	elliptic	clma
2D	电路路径延迟（ns）	3.44	5.18	3.41	4.78	6.27	6.08	7.92
	W_{min}	24	34	28	36	48	36	48
	Area ($\times 10^6$)	3.16	6.38	4.79	5.67	13.96	11.43	30.94
3D	电路路径延迟（ns）	2.58	3.92	2.68	3.62	4.02	3.84	5.79
	W_{min}	18	26	18	22	36	26	32
	Area ($\times 10^6$)	3.41	7.01	4.96	6.17	15.39	12.64	34.17
改进幅度：数据输入**ELAY** = +28.66% W_{MIN} = +29.92% **AREA** = -8.86%								

参考文献

[1] C. Ababei 等，“三维集成电路中的布局与布线”，载于 IEEE Design & Test of Computers，第 22 卷，第 6 期，第 520–531 页，2005 年 11–12 月，doi: 10.1109/MDT.2005.150。[2] G. E. Moore，“在集成电路上集成更多元件”，载于 Electronics Magazine，第 38 卷，第 8 期，第 114–117 页。1965 年 4 月[3] H. Jahanirad，“CC-SPRA：用于基于信号概率的可靠性分析的相关系数方法”，载于 IEEE Transactions on Very Large Scale Integration (VLSI) Systems，第 27 卷，第 4 期，第 927–939 页，2019年4月，doi: 10.1109/TVLSI.2018.2886027。[4] H. Jahanirad，“组合逻辑电路和时序逻辑电路的高效可靠性评估”，载于 J Comput Electron 18，343–355，2019年，https://doi.org/10.1007/s10825-018-1288-4[5] M. Savari、H. Jahanirad，“NN-SSTA：一种用于统计静态时序分析的深度神经网络方法”，载于 Expert Systems with Applications，149，113309，2020年[6] 国际半导体技术路线图（ITRS）。2015 年版。[在线], 可获取: https://www.dropbox.com/sh/3jfh5fq634b5yqu/AADYT8V2Nj5bX6C5q764kUg4a?dl=0[7] I. Kuon 和 J. Rose，“测量现场可编程门阵列与专用集成电路之间的差距”，载于 IEEE Transactios on Computer-Aided Design of Integrated Circuits and Systems，第 26 卷，第 2 期，第 203–215 页，2007 年 2 月，doi: 10.1109/TCAD.2006.884574。[8] K. E. Murray、O. Petelin、S. Zhong、J. M. Wang、M. ElDafrawy、J.-P. Legault、E. Sha、A. G. Graham、J. Wu、M. J. P. Walker、H. Zeng、P. Patros、J. Luu、K. B. Kent 和 V. Betz，“VTR 8：高性能计算机辅助设计与可定制FPGA架构建模”，ACM TRETS，2020年[9] G. Karypis、R. Aggarwal、V. Kumar 和 S. Shekhar，“多层超图划分：在超大规模集成电路领域的应用”，载于 IEEE Transactions on Very Large Scale Integration (VLSI) Systems，第 7 卷，第 1 期，第 69–79 页，1999 年 3 月，doi: 10.1109/92.748202。[10] N. Siddique 和 H. Adeli，“模拟退火及其变体与工程应用”，International Journal on Artificial Intelligence Tools，第 25 卷，第 06 期，第 1630001 页，2016年。

[11] X. Geng、Z. Chen、W. Yang、D. Shi、K. Zhao，“基于结合贪心搜索的自适应模拟退火算法求解旅行商问题”，《应用软计算》11(4)(2011年)3680–3689[12] E. Hopper、B. Turton，元启发式算法在二维条带装箱问题中的应用综述，《人工智能评论》16(4)(2001年)257–300。[13] N. Leite、F. Melício 和 A. C. Rosa，“用于考试排程问题的快速模拟退火算法”。Expert Systems with Applications。doi:10.1016/j.eswa.2018.12.048，2018年。[14] P. Subramanian、N. Ramkumar、T. T. Narendran 和 K. Ganesh，“PRISM：用于闭环供应链网络设计问题的优先级模拟退火”。《应用软计算》，13(2)，1121–1135。doi:10.1016/j.asoc.2012.10.004，2013年。[15] M. Z. Allahvari 和 A. Azab，“非等面积设施布局问题的数学建模与多起点搜索模拟[illegible]载于 F[illegible]
doi:1
Zhan[illegible]模拟退火进行硅通孔协同设计的三维集成电路芯片布局规划”，2018年IEEE国际电磁兼容研讨会和2018年IEEE亚太电磁兼容研讨会（EMC/APEMC）。doi:10.1109/isemc.2018.8393839[17] G. Sergey、Z. Daniil 和 C. Rustam，“基于模拟退火的可重构片上系统布局优化”，2019年IEEE俄罗斯青年电气与电子工程研究人员会议（EIConRus），俄罗斯圣彼得堡和莫斯科，2019年，第1597–1600页，doi: 10.1109/EIConRus.2019.8657251。[18] K. Miyazaki 等，“采用模拟退火算法优化开关过程中的过冲和能量损耗的通用时钟控制栅极驱动器集成电路，具有可编程63级驱动能力”，载于《IEEE工业应用汇刊》，第53卷，第3期，第2350–2357页，2017年5–6月，doi: 10.1109/TIA.2017.2674601。[19] A. Fakheri Tabrizi、L. Behjat、W. Swartz 和 L. Rakai，“用于三维集成电路划分的快速力导向模拟退火”，载于《集成：超大规模集成电路期刊》，55，202–211。doi:10.1016/j.vlsi.2016.06.003。2016年。[20] E. Nasiri、J. Shaikh、A. Hahn Pereira 和 V. Betz，“多个裸片协同工作：硅中介层现场可编程门阵列的计算机辅助设计流程与布线架构”，载于 IEEE Transactions on Very Large Scale Integration (VLSI) Systems，第24卷，第5期，第1821–1834页，2016年5月，doi: 10.1109/TVLSI.2015.2478280。[21] V. Pangracious、Z. Marrakchi、H. Mehrez，“现场可编程门阵列：概述及基于树的FPGA架构的三维设计方法”。施普林格，纽约。2015年[22] M. Morshedzadeh、A. Jahanian 和 P. Pourashraf，“新兴三维现场可编程门阵列中的三维开关盒多路复用，以缩小芯片占用面积并提高硅通孔利用率”，《集成：超大规模集成电路期刊》，50，81–90，2015年。[23] B. K. Kaushik、M. K. Majumder、A. 和 Kumari，“铜和碳纳米管硅通孔的制造与建模，三维集成电路与系统设计”。CRC出版社/泰勒-弗朗西斯集团，佛罗里达州博卡拉顿。2014年。[24] N. Sherwani，《超大规模集成电路物理设计自动化算法》，第219–286页，2002年。[25] P. Jamieson、K. B. Kent、F. Gharibian 和 L. Shannon，“Odin II——用于计算机辅助设计研究的开源Verilog硬件描述语言综合工具”，2010年第18届IEEE年度现场可编程定制计算机国际研讨会，北卡罗来纳州夏洛特，2010年，第149–156页，doi: 10.1109/FCCM.2010.31。[26] 伯克利逻辑综合与验证组。（2018）。ABC：用于时序综合与验证的系统。版本，3，26。[27] S. Gupta、M. Hilbert、S. Hong、R. Patti，“采用高密度互连制造三维集成电路的技术”，载于超大规模集成电路多层互连会议，2004年。[28] H. Jahanirad，“采用协同进化方法降低基于SRAM的现场可编程门阵列上实现电路的软错误率”，载于《国际计算机应用期刊》，180，42–49，2018年。[29] 预测技术模型，可通过[在线] 获取: http://ptm.asu.edu/latest.html[30] S. Chtourou、M. Abid、Z. Marrakchi、E. Amouri 和 H. Mehrez，“探讨基于划分的布局方法，以提升三维现场可编程门阵列的性能”，2017年高性能计算与仿真国际会议（HPCS），热那亚，2017年，第572–579页，doi: 10.1109/HPCS.2017