

一种针对QCA电路自动布局布线的混合策略研究

李杨帅¹, 彭斐¹, 韩倩¹, 李小帅², 解光军¹

(1. 合肥工业大学微电子学院, 安徽合肥 230009; 2. 国防科技大学电子对抗学院, 安徽合肥 230009)

摘要: 量子元胞自动机(Quantum Cellular Automata, QCA)电路的自动布局布线是在相关约束条件下自动放置电路单元、自动形成连线, 实现门级或元胞级电路的设计过程, 是QCA电路设计大型化、复杂化和系统化的必要工具。布局布线算法设计过程中最大的难题是如何解决“时钟同步”, 随着二维时钟方案提出, 该问题的解决方案变得更加策略化, 但仍存在诸多缺陷, 如成功率低, 布局面积较大等。本文将二维时钟方案的布局布线问题抽象成组合优化模型, 提出了一种基于遗传算法GA(Genetic Algorithm)和改进A*算法的混合策略。两种算法相互配合搭建可能的电路布局, 并通过精心设计的适应度函数, 搜索满足时钟同步的个体, 最终实现从硬件电路到二维时钟方案上的门级布局。实验结果表明, 本算法在目前被广泛应用的二维时钟方案USE(Universal, Scalable and Efficient)上的布局成功率接近100%。相较于当前世界上最先进的两个QCA布局布线工具fiction和Ropper, 本算法可适用电路规模更大(逻辑门数量大于10), 在成功率和生成布局面积上都有大幅度的优化。

关键词: 元胞自动机; 布局布线; 组合优化; 遗传算法; A*算法

中图分类号: TP391.9

文献标识码: A

文章编号: 0372-2112(2023)03-0666-09

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20211212

One Hybrid Strategy for Automatic Placement and Routing of QCA Circuit

LI Yang-shuai¹, PENG Fei¹, HAN Qian¹, LI Xiao-shuai², XIE Guang-jun¹

(1. College of Microelectronics, Hefei University of Technology, Hefei, Anhui 230009, China;

2. College of Electronic Engineering, National University of Defense Technology, Hefei, Anhui 230009, China)

Abstract: The automatic placement and routing (P&R) of quantum cellular automata (QCA) circuits is the design process of placing and connecting circuit units with relevant constraints, and then generating gate-level or cell-level layout of circuits. It is a necessary tool for large-scale, complex and systematic circuit design. The biggest problem in the P&R algorithm design is how to solve “clock synchronization”. With the proposed two-dimensional clocking schemes, the solution has become more strategic, but there are still many shortcomings, such as low success rate and large layout area. This paper abstracts the P&R problem on the two-dimensional clocking scheme into a combinatorial optimization model, and proposes a hybrid strategy based on the genetic algorithm (GA) and the enhanced A* algorithm. Two algorithms cooperate with each other to build a possible circuit layout, and through the carefully designed fitness function, search the individuals that meet the clock synchronization, and finally generate the gate level layout from the hardware circuit. Experimental results show that the placement success rate of this algorithm in the widely used two-dimensional clocking scheme USE (Universal, Scalable, and Efficient) is close to 100%; Compared with the two most advanced QCA P&R tools fiction and Ropper, this algorithm can be applied to a larger circuit scale (the number of logic gates is greater than 10), and has a significant optimization in success rate and generated layout area.

Key words: quantum cellular automata; placement and routing; combinatorial optimization; genetic algorithm; A* algorithm

1 引言

量子元胞自动机(Quantum-dot Cellular Automata, QCA)是一种全新的纳米计算范式^[1], 是实现下一代集

成电路强有力的候选器件。QCA中信息传递的方式不再依靠电流或是电压, 而是通过相邻元胞的耦合作用, 与传统CMOS(Complementary Metal Oxide Semiconduc-

tor, CMOS)技术相比,这种局部耦合机制具有更高的集成度和更低的功耗. 目前 QCA 电路设计已经取得了不错的进展,一些研究学者在以加法器^[2]、乘法器^[3]为代表的小规模组合电路、以触发器^[4]、寄存器^[5]为代表的时序电路、以 PLA(Programmable Logic Arrays)^[6]、RAM(Random Access Memory)^[7]为代表的大规模电路等各方面进行了广泛的研究. 随着电路规模的增大,自动化布局布线开始受到研究者的关注.

解决布局布线的前提是要解决时钟问题,因为 QCA 电路具有一个特性叫做“布局=时钟”^[8]. 时钟信号时刻伴随着走线,它控制信息传输的方向并提供元胞运算的能量,这使得布线的同时还要考虑时钟的周期和相位. 后来一些预定义的时钟方案被研究者提出. 如图 1 中的二维波纹型(Two-Dimensional Wave, 2DDwave)^[9]时钟方案和通用型、可扩展型、高效型(Universal, Scalable, and Efficient, USE)^[10]时钟方案,以及图 2 中的鲁棒型、高效型、可扩展型(Robust, Efficient and Scalable, RES)^[11]时钟方案和便捷型、灵活型、高效型(Convenient, Flexible, and Efficient, CFE)^[12]时钟方案. 时钟相位都被固定在类似棋盘格的小网格中,每个小网格表示一个时钟区域,箭头表示该时钟区域间信息的流向.

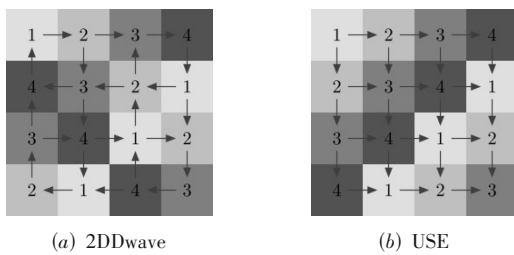


图 1 USE 时代时钟方案

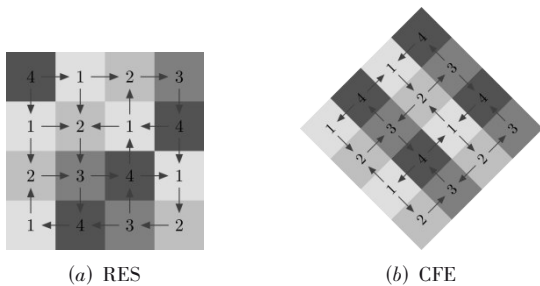


图 2 后 USE 时代时钟方案

研究者只需要将逻辑门放入网格区域中,然后保证到达该区域的两个输入信号的路径长度相等就能够自动实现信号的同步. 借助时钟方案的独特功能,解决 QCA 电路信号同步的难题转变成在时钟方案上寻找放置逻辑门的合适位置,以及建立这些逻辑门间的路由

关系. 本文将前者抽象成组合优化问题,后者抽象成障碍寻路问题,提出了一种混合算法方案,分别交由遗传算法(Genetic Algorithm, GA)和 A* 寻路两个算法模块解决. 实验结果表明,两个算法成功实现了 QCA 门级电路在四种时钟方案上的布局布线;相比当前国外最先进的 QCA 布局布线工具,本文的算法在针对复杂电路(逻辑门数量超过 10)时的成功率明显比 fiction^[13]更高,在 USE 时钟方案的成功率达到了 100%;在生成布局面积方面,与 Ropper^[14]相比具有绝对优势.

2 相关工作

QCA 自动布局布线算法的处理对象是电路对应的节点网络图,研究人员根据不同类型图的特点、QCA 时钟类型、以及电路设计规则,设计并实现了不同的算法方案.

2.1 一维时钟的算法方案

2003 年, Niemier 及其团队针对 QCA 电路的布局布线问题最早提出基于一维时钟方案的 KLBG(K-Layered Bipartite Graph)算法^[15-17]. 如图 3 所示,以逻辑函数 $f = a \cdot s + b \cdot \bar{s}$ 为例,图 3(a)是逻辑函数的节点网表;图 3(b)在节点图中插入若干虚拟节点来平衡路径,节点被划分层级后转换为 KLBG 图,然后将相同层级的节点放在同一列,并根据一维时钟方案的时钟顺序分配时钟信号,最后通过节点与逻辑门一一映射,路由连接逻辑门完成布局布线(图 3(d)). 一维时钟方案的使用解决了信号同步问题,但电路规模增加会导致投入同步的资源增大,并且无法应用于时序电路.

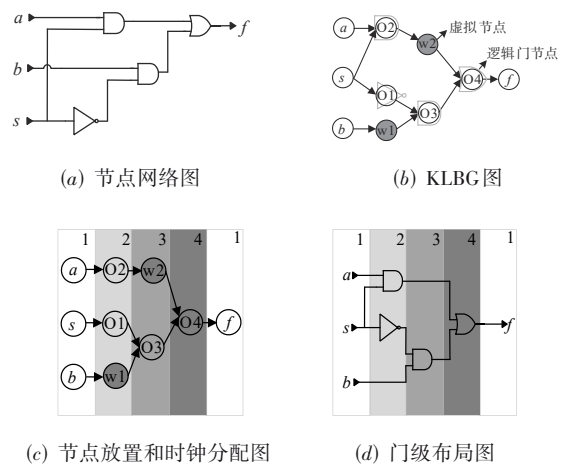


图 3 一维时钟方案基于 KLBG 的算法方案示意图

2.2 二维时钟的算法方案

随着二维时钟方案的广泛应用,尤其是 2016 年 USE 时钟方案的提出, QCA 电路正式开启了“自动化”布局布线的时代.

2.2.1 fiction 两种算法方案

2018年, Walter等人基于公式求解的 exact 算法方案完成了第一个元胞级布局布线工具^[18]. exact 算法采用 SMT(Satisfiability Modulo Theories) 求解引擎实现对公式的求解, 其中电路类型、线路交叉和电路的元胞数目等是算法的参数, 可以由使用者选择和调整以研究不同参数下的不同布局结果. 随着电路规模扩大, 布局布线过程变得更加复杂, 求解符号公式会消耗更多的时间且较难求解得到正确的布局. 后来 Walter等人将二维时钟方案 2DDwave 和正交绘图算法(Orthogonal Graph Drawing, OGD)相结合提出了可以适用于较大规模 QCA 电路的布局布线 OGD 算法^[19](图4), 但是该算法未考虑线路交叉、线路长度和布局面积, 更重要的是 2DDwave 时钟方案也不能应用于时序电路. 2021年, 在 exact 方案的基础上, 该组优化算法并提出了 one-pass 方案^[20], 将电路的逻辑综合和物理设计步骤在一个环节中实现, 算法的实现思想同样是公式求解, 并且从一开始就考虑 QCA 布局布线的约束和优化问题, 这增加了算法设计的复杂性, 导致求解成功率较低.

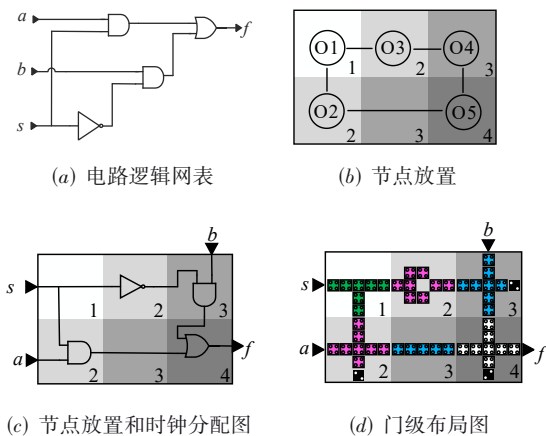


图4 二维时钟方案 OGD 方案示意图

2.2.2 Ropper 算法框架

2016年, 由 Trindade 等人将贪婪算法和 BFS 应用于 QCA 电路的自动布局布线^[21]. Fontes 等人拓展文献^[21], 在实现电路布局布线的同时搜索最优解, 但未考虑线路交叉, 并且引入了逻辑门的多层布局^[22]. 后来同组的 Formigoni 等人提出基于搜索算法的泛型布局布线框架 Ropper, 应用模拟退火算法(Simulated Annealing, SA)和动态规划算法计算布局布线解; 动态规划算法计算节点与其扇入节点间的可能路径, SA 作为迭代引擎, 以布局面积作为衡量电路优劣的指标进行搜索. 但是 Ropper 为了满足时钟同步, 将电路中同一层的所有节点强行放置到相同的时钟区域中, 且第一层节点必须放置到布局区域的中心位置, 这极大地限制了启发式算法

对最优解的搜索.

2.2.3 提出的算法方案

为了解决上述工作中遇到的问题, 本文将 QCA 电路的布局布线问题建模为一个智能搜索和一个智能寻路的两级优化模型, 分别采用 GA 和 A* 算法解决.

GA 是以大量种群和个体进行有效搜索, 具有强大且快速的收敛能力, 很适合复杂且庞大的解空间, 该特点具有天然的并行性, 相比其它组合优化算法更容易实现并行化; GA 还具有易扩展性, 可以与其它智能算法混合使用, 如搭配模拟退火算法和禁忌搜索算法, 可以弥补自身易陷入局部最优的缺陷. 在本次工作中, GA 充当着管理引擎的角色, 整个流程如图5所示, 其中种群中的个体表示一种电路布局, 初代个体在时钟网格上随机放置电路节点. 节点放置完成后调用 A* 算法进行布线, 然后适应度函数 F_n 从节点的布局、布线成功的数量、时钟同步、布局面积四个方面对个体进行计算, 再通过三大遗传算子(选择、交叉、变异)以及合适的繁衍次数, 让个体不断地进化和改良, 最终获取一个符合电路设计要求、面积较小的布局布线结果.

布线算法一直是整个工作的难题, 它需要考虑许多电路设计规则, 如时钟流的方向、网格的空间容量、线路交叉等问题. 在所有节点位置已知的情况下, 本文更加保守地选择了单点布线模式, 因为并行布线模式易引发新的 NP(Non-deterministic Polynomial) 问题, 文中采用的 A* 算法是点对点寻路中应用最多、最稳定的智能搜索算法之一.

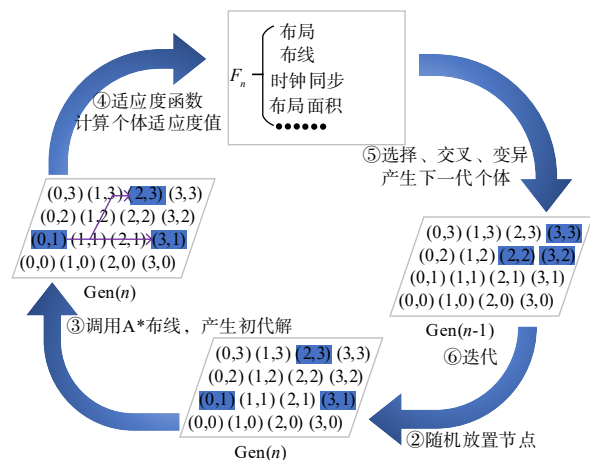


图5 GA 管理引擎工作流程图

3 算法设计

3.1 遗传算法设计

遗传参数影响最优解的求解结果和求解效率, 但至今没有设置参数的理论依据, 因此主要依靠多次运行和调试. 在本次实验中, 通过反复测试, 设定初始种

群为 500,繁衍代数为 600,能够获得较好的布局结果.

3.1.1 编码方案

种群中的每条染色体(每个个体)表示一种布局方案,染色体上的基因与电路节点在时钟网格上的坐标一一对应,基因的位数由节点的数量决定.本文采用“整数编码”方案,将二维坐标转换为一维序列上的基因值,编码公式如式(1)所示,其中 (x,y) 表示节点的坐标, G_{size} 正方形布局区域的宽度,如在 6×6 大小的网格上,坐标(2,3)位置上的节点可编码为 15;利用坐标值的整数特性,式(2)可将编码值 E 重新解析回坐标.

$$E = x \times G_{size} + y \quad (1)$$

$$\begin{cases} x = E / G_{size} \\ y = E \% G_{size} \end{cases} \quad (2)$$

3.1.2 适应度函数设计

在 2.2.3 节中提到,电路布局的优劣程度决定个体适应度值的大小,具体计算公式如下:

$$F_n = \alpha P + \beta R + \gamma S + \theta A \quad (3)$$

其中, P 、 R 、 S 、 A 为电路布局布线的重要考察项; α 、 β 、 γ 、 θ 为各项的适应度系数,用于控制和调节适应度值的占比.

α 为布局系数,值设为 1, P 为布尔值,它表示所有节点是否放置成功,当每个节点在布局区域有且仅有一个坐标时, P 为真,其它情况均为假.

β 为布线系数,值设为 10,当所有节点在布局区域放置成功后,GA 调用 A*算法进行布线,记录布线成功的数量并且保存路径; R 值为布线成功的数量占布线总数的比,若布局失败(P 为假), R 值计为 0.

γ 为时钟同步系数,值设为 5,时钟同步的要求是保证电路中的逻辑节点(与门、或门)的两个输入的时钟数量一致,即逻辑节点的两条输入路径的长度相等;由于实现难度较大,部分个体甚至在布线阶段都无法保证 100%的成功率.为了正确地引导个体进化,算法会根据布线的完成度给予一定的适应度值奖励,如式(4)所示,精确地计算每一个逻辑节点的两条输入的路径长度,然后根据布线情况选择对应的适应度值;式中 S 表示单个逻辑节点的适应度值,全部节点的适应度值为 $\sum S$;若逻辑节点 n 的两条输入都布线失败,即路径都为空,获得的适应度值最小;若输入都存在, S 由它们的布线距离差值决定, Max_{len} 、 Min_{len} 分别表示两条输入路径的最大、最小值,差值越小,适应度值越大,当长度相等时,意味着该逻辑节点的布线满足时钟同步要求,获得最大适应度值.

$$S = \begin{cases} 1/10\ 000 \\ 1/100 \\ 1/(0.5 + Max_{len} - Min_{len}) \end{cases} \quad (4)$$

θ 为面积系数,值设为 5,当检测到个体满足时钟同步时,计算电路的布局面积,面积越小,获得的适应度值越大,如式(5)所示, A 为整个布局区域的面积和电路布局面积 d 的比.

$$A = G_{size} \times G_{size} / d \quad (5)$$

3.1.3 遗传算子

(1) 选择算子:采用轮盘赌法,即基于适应度比例的选择策略,式(6)为个体 i 被选中的概率, N 为种群大小,式(7)为选择算子的计算公式,其中, P 表示(0,1)区间里的随机数,经过 N 次选择过程,若个体 i 满足公式,则被选择进入下一代,若不满足,则选择种群中第一个个体进入下一代.

$$P_i = \frac{F_i}{\sum_{j=1}^N F_j} \quad (6)$$

$$\sum_0^i p_j \leq P_i \leq \sum_0^{i+1} p_j \quad (7)$$

(2) 交叉算子:选择算子结束后,对新种群进行交叉操作.两条染色体为一组,交叉率为 90%,采用两点交叉方式,如图 6 所示,首先随机选择染色体 a 上的两个点位,图中选择的点位是 4 和 7,然后同染色体 b 交换 a_4 、 a_7 和 b_4 、 b_7 之间所有的基因,组成两条新的染色体.

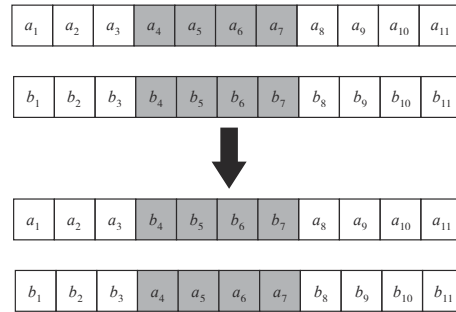


图 6 交叉算子示意图

(3) 变异算子:交叉操作结束后,下一步对种群中的全部个体进行变异操作.变异率为 50%,采用 1/4 基因突变算子,即选择染色体上 1/4 的基因位作为突变对象,如图 7 所示,染色体 a 共有 11 个基因,则选择 3 个连续的基因位,首先算法随机选择 a_4 ,然后随机生成新的坐标编码替换原有的 a_4 、 a_5 、 a_6 ,最后将变异后的染色体遗传给下一代种群.

3.2 时钟方案上 A*算法布线设计

3.2.1 A*算法介绍

A*是启发式寻路算法,具有速度快、自动避障、路径最短等特点,它结合了贪婪算法(总在最有希望的方向上搜索)和 Dijkstra 算法(总能找到最短路径)的优点,这些优点的数学理论都源于下面的寻路消耗公式,

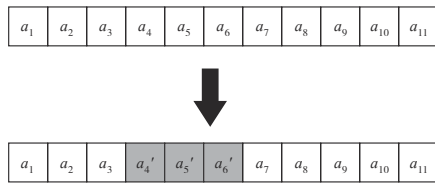


图7 1/4基本位变异

它正确地指引了A*的扩展方向.

$$f_n = g_n + h_n \quad (8)$$

其中, f_n 为节点 n 的寻路消耗值, g_n 表示从起点到节点 n 的实际消耗, 以网格数作为计算消耗量的单位, 线路每经过一个时钟区域的消耗记为 1; h_n 表示节点 n 到终点的最佳路径的估计消耗, 估计消耗的标准启发函数有对角线距离、欧式距离、曼哈顿距离, 在此次工作中, 布线只能从水平或垂直方向展开, 因此 h_n 的计算方式采用的是曼哈顿距离, 若节点坐标为 (x_n, y_n) , 终点坐标为 (x_m, y_m) , 估计消耗计算如下:

$$h_n = |x_n - x_m| + |y_n - y_m| \quad (9)$$

3.2.2 时钟方案布线设计

综上所述, A*算法的内核是 f_n , 但在实际问题中, A*的布线过程是发生在时钟方案上, 它必须遵守时钟规则和电路设计规范, 因此, 计算节点 f_n 前必须检查以下两点.

(1) 时钟: 若节点处于时钟相位 1, 则布线方向只能朝着时钟相位 2 的区域, 整个电路须严格按照 1-2、2-3、3-4、4-1 的时钟流进行布线; 由于时钟相位具有规律性, 可以利用三元表达式将区域的二维坐标和相位绑定, A*可直接通过坐标获取所处的相位值; 每一种时钟方案都有一套独立的三元表达式, 以 USE 时钟方案为例, 三元表达式如下, 若已知时钟区域坐标 $(1, 1)$, 可计算出对应的时钟相位为 3.

$$(\text{mod}(y, 2) \neq 0)?$$

$$\left((\text{mod}(y+1, 4) \neq 0)?(1+\text{mod}(x+2, 4)): (1+\text{mod}(x, 4)) \right):$$

$$\left((\text{mod}(y, 4) = 0)?(4-\text{mod}(x+2, 4)): (4-\text{mod}(x, 4)) \right)$$

(10)

(2) 电路设计规范: 时钟区域有空间大小, 根据设计规则最多布置三根线或者放置一个电路节点和一根线; 在编程中, 算法将空间大小进行量化, 设定每一个时钟区域的最大容量值为 6, 每放置一个节点消耗的容量值为 4, 每布置一根线消耗的容量值为 2, 即可满足电路设计要求; A*运行过程中, 时刻更新空间的使用情况, 若检测到某时钟区域容量已满, 则将该地区视为障碍, 无法再进行布线.

3.2.3 A*布线流程

算法流程如图 8 所示, 实现步骤如下:

Step1 开始: 输入起点 S、终点 T 位置坐标.

Step2 遍历: 选择起点周围(上、下、左、右)四个区域.

Step3 筛选: 根据时钟规则和电路设计规范, 筛选出可通行的区域, 并将起点 S 记录为这些区域的源起点.

Step4 判断: 若筛选出区域中包含终点 T, 意味着布线成功, 从终点 T 出发, 根据记录的源起点, 反馈至起点 S, 生成一条最优路径, 流程结束; 若可通行区域不包含终点 T, 则将它们都放入开启列表.

Step5 计算: 在开启列表中, 分别计算它们到起点 S 的 g 值和到终点 T 的 h 值, 算出总寻路消耗 f .

Step6 更新: 将开启列表中 f 值最小的区域放入关闭列表, 并将该区域设为新起点.

Step7 迭代: 重复 Step2~Step6, 期间若开启列表检测到放入了重复区域, 分别检测当前起点和重复区域的 g 值, 若重复区域 g 值更小, 则将重复区域的源起点设为当前起点; 这个检测操作很重要, 它保证了 A*找到的路径始终是最优的.

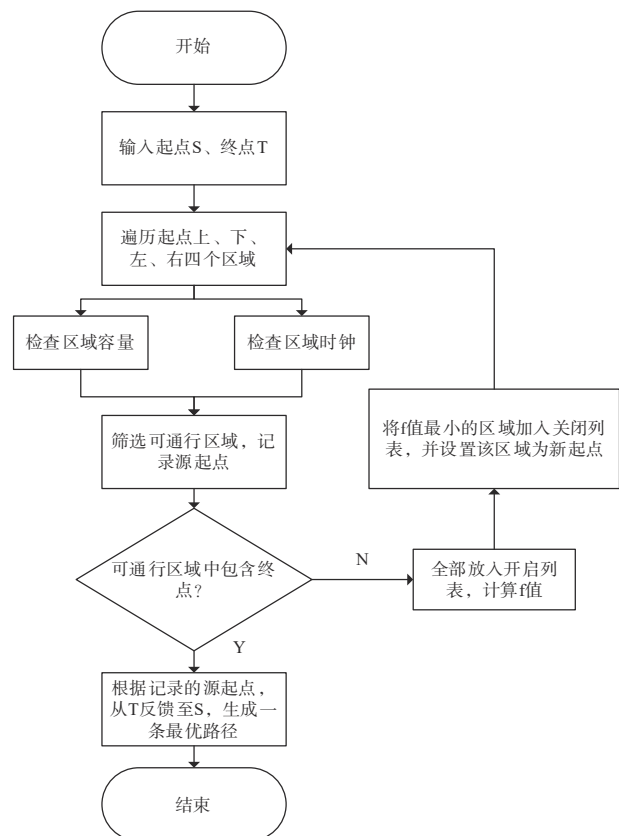


图8 A*流程图

经改造后的 A*不再简单地将所有空间相邻的位置都考虑在内, 以检测时钟为例, 图 9 展示了原始 A*算法和引入 USE 时钟方案之后的 A*算法在相同条件下的不同结果. 从图中可以看出, 改进后的 A*寻路算法在第

二个转弯处的相位为 1,不能直接右转,而必须先沿着 USE 时钟方案规定的信息流方向(相位 2)向下走,然后才能右转.

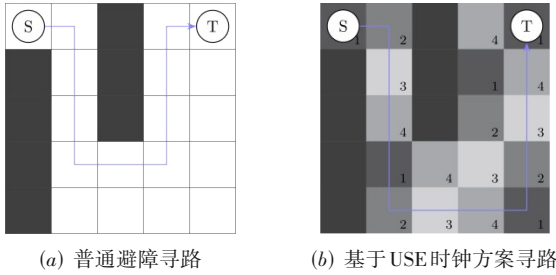


图 9 A*算法寻路展示

4 实验展示与比较

本节中以 ISCAS85 库^[23]中的 C17 电路(图 10)来展示布局布线效果,整个工程采用基于 C++17 标准的 C++ 编程语言,所有的实验工作在 Intel core i5-10400 处理器上进行,该处理器主频为 2.90 GHz,主存 16 GB,开发环境 Ubuntu-20.04.

图 11 展示了四种时钟方案上的布局布线结果,其中,时钟区域网格中节点的序号分别对应着 C17 电路的五个输入和六个逻辑门,图的下方展示了 GA 中个体的适应度值变化曲线.

从布线结果分析,信息传递的方向严格地遵守了时

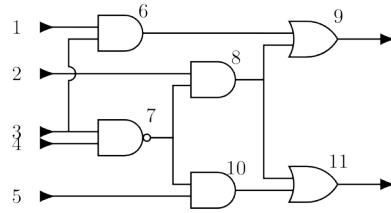


图 10 C17 电路原理图

钟约束,都是按照时钟相位的顺序进行布线,交叉线也符合预期的设计,每个时钟网格中最多布置了三根线,或者一个节点和一根线.从电路的角度分析,节点 6、7、8、9、10、11 为逻辑门,即逻辑节点,按照时钟同步的设计规则,需保证逻辑节点的两个输入信号的路径长度相等,从图中可以检查到,每一个逻辑节点都实现了这一点.

从使用的时钟方案角度分析,2DDwave 和 USE 上节点的分布显得更加分散,布局面积都为 6×6 个网格单元,每个网格面积为 25 个元胞大小;交叉线的数量分别是 2 和 5;RES 和 CFE 布局面积都为 5×6,交叉线数量分别是 1 和 0,因此,可以认为 RES 和 CFE 以更小的面积、更少的交叉线完成了电路的布局布线,交叉线会产生异面结构,会进一步加大器件设计上的难度,并且对电路的稳定性产生影响,从实验结果可以验证“后 USE”时代时钟方案(RES 和 CFE)在时钟设计上要比“USE 时代”(2DDwave 和 USE)更加先进,它们在时钟分配上更加灵活,增加了更多的布线通道.

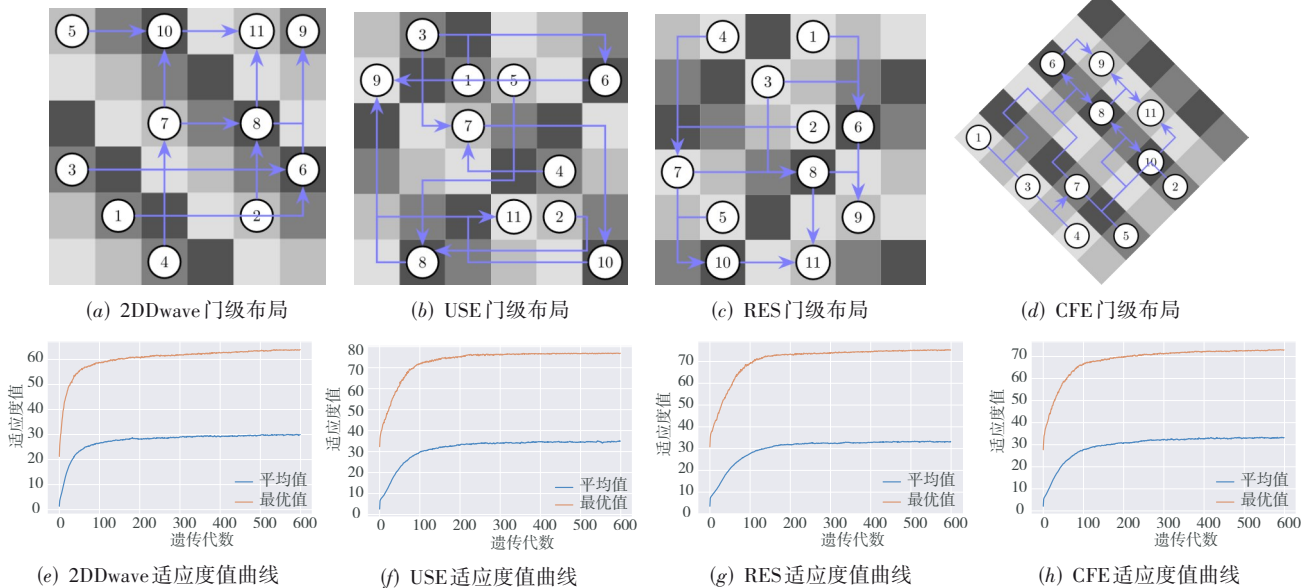


图 11 C17 电路在四种时钟方案上布局布线结果及对应遗传算法适应度值收敛曲线

从适应度收敛曲线分析,遗传代数为 600,前 100 代适应度曲线快速升高,GA 快速收敛,200 代左右收敛变

缓,400 代左右收敛完成,之后每一代中产生的电路布局的适应度平均值都趋于一个较高的水平,说明 400 代

之后的个体都具备成为较优解的资质. 换个角度思考, 从200代之后收敛速度骤降, 适应度值开始以小幅度缓慢地增长, 因此GA陷入局部最优解的可能性较大, 这也暴露了GA的劣势; 若200代后, 按适应度值挑选出一个最优个体, 然后交由模拟退火算法, 将以种群为单位的大规模迭代切换成单个个体的优化, 搜索效率应该会更高.

表1是同fiction和Ropper的算法比较, fiction是基于SMT计算引擎的exact算法, Ropper是以SA和动态规划为组合的搜索算法, 三个算法中只有fiction实现了QCA电路元胞级的工作, 因此表中全部数据都是基于电路门级的实验结果. 实验采用了11个基本电路, 分别在2DDwave和USE时钟方案上进行测试, 测试内容包括算法成功率、布局面积、关键路径长度、算法运行时间.

表1 布局布线算法比较

Benchmarks				CS	fiction-exact			Ropper			提出的算法		
电路名称	节点数量	输入/输出	布线	U/T	面积	关键路径	时间/s	面积	关键路径	时间/s	面积	关键路径	时间/s
XOR	5	2/1	6	U	4×5	9	0.7	5×6	10	2.7	3×3	6	2.8
				T	3×6	8	0.2	5×6	10	2.8	3×3	5	2.4
2:1 MUX	6	3/1	6	U	5×3	7	0.2	5×6	10	2.7	3×4	7	3.4
				T	3×4	6	0.1	5×6	10	2.7	3×3	4	2.0
4:1 MUX	13	4/1	19	U	-	-	-	14×14	18	15.3	6×9	20	45.9
				T	7×8	13	34.0	-	-	-	-	-	-
ParGen	6	3/1	9	U	4×10	14	287.0	9×14	18	4.2	5×5	16	8.0
				T	5×7	10	0.8	13×3	18	2.9	5×5	9	5.7
ParCheck	13	4/1	18	U	-	-	-	-	-	-	8×9	29	55.1
				T	7×8	14	10.6	-	-	-	9×10	16	56.1
1 bitAdder	10	6/1	15	U	-	-	-	11×17	18	2.2	7×5	20	22.0
				T	6×10	14	6.4	14×17	18	5.0	-	-	-
C17	11	5/2	12	U	-	-	-	7×8	14	2.8	5×5	10	5.8
				T	5×7	10	1.0	6×11	14	2.9	4×6	8	5.3
FA	9	3/2	12	U	-	-	-	-	-	-	5×8	15	52.4
				T	7×9	15	330.2	9×10	22	21.6	-	-	-
FS	9	3/2	12	U	-	-	-	7×12	20	29.8	6×6	15	156.0
				T	5×10	14	227.9	-	-	-	-	-	-
HA	6	2/2	8	U	5×6	10	14.2	-	-	-	3×6	10	9.1
				T	4×5	8	1.2	-	-	-	4×5	6	4.7
HS	6	2/2	8	U	5×6	15	10.7	-	-	-	3×5	8	8.8
				T	4×6	9	3.1	-	-	-	3×7	8	8.4

CS: 时钟方案. 节点数量: 包含电路的输入和逻辑门. 关键路径: 输入到输出经过的延时最长的路径. 面积: 25个元胞大小为一个单位. U/T: USE/2DDwave. -: 无法找到结果或者运行时间超过1小时. 布线: 电路布线数量.

从成功率上看, 本文提出的GA算法在USE上为100%, 但是在2DDwave仅为50%, 这与时钟方案的设计有关. 2DDwave的时钟流只有两个方向, 它能够有效地保留电路的层级性, 如图11(a)所示, 可直观地分析出电路的输入、输出以及逻辑门的分布情况, 但是这并不适合随机性较强的GA, GA极易扰乱这种层级性; USE的时钟流有四个方向, 即使节点分布散乱, 通过迂回线, 也能灵活地完成布线; 因此USE要比2DDwave的成功率更高. fiction采用的SMT引擎的计算公式中, 包含了对时钟方案的解析, 时钟流向越简单, 算法复杂度越低, 成功率越高, 因此fiction在2DDwave上的成功率比USE更高. Ropper方案强行将第一层级节点放在布局中心位置, 由中心向四周延伸, 这种布局方式更不擅长

于2DDwave, 因此失败率较高.

从布局面积上看, GA算法是最优的, 其次是fiction, Ropper则完全不占据优势, 并且随着电路节点和布线数量增多, Ropper的面积愈发庞大. 例如在4:1MUX电路中, 经计算, 它的布局面积是GA的2.7倍. Ropper方案在解决时钟同步时, 强制将同一层级的电路节点放置到相同时钟相位上, 导致节点之间的布线长度始终为时钟周期的倍数, 这意味着节点间最短布线长度为一个时钟周期, 而GA中节点的放置与时钟无关, 提供了更多面积较小的布局.

从关键路径上看, GA处于一个不可控的状态, 当前的方案只保证了电路的正确性, 性能上的优化有所欠缺. 以4:1MUX为例, 在面积差距悬殊的情况下, GA

的关键路径反而比 Ropper 更长,显然这是布线过程中迂回线过多导致的. 在传统电路中,电路的输入、输出端口应该均匀的分布在布局的两侧,电路的逻辑门分布于中心位置,这样才能保证关键路径处于一个正常值. 2DDwave 时钟方案虽然能够辅助实现这样的布局,但是会降低算法成功率. 因此合适的解决方案可以将节点分布情况也作为适应度函数计算的一部分.

从运行时间上看,GA 的优势并不明显,但有继续优化的空间. 首先所有测试电路设置的遗传代数和种群规模为 500 和 600,而针对 5~7 个节点的小型电路,这些参数可以适当调小来降低时间复杂度. GA 需要一个自适应的前置算法,根据输入电路节点和布线数量解析电路复杂度,然后设置合适的遗传参数. 其次通过表中数据可以看出,在节点数量相等的情况下,布线数量直接影响布线时间. 在本文中,布线也是 NP 问题,但是采用了串行布线模式规避了这一难题. 并行布线模式可以解决这一 NP 难题,从而提高布线效率.

5 结论

本文提出了一种应用于 QCA 电路自动布局布线的组合算法,实现了从硬件电路的输入到二维时钟方案上的门级布局的输出. 对比已提出的布局布线方案,本算法在布局面积和算法成功率上有较大的优势. 本文对大规模电路也进行了测试,但是很难在短时间内得到最优解,主要原因包括 GA 后期搜索乏力和串行布线的速度较慢. 在未来工作中,将基于当前的算法框架引入其他组合优化算法来辅助 GA,采用并行布线模式以缩短运行时间. 另外,大规模电路可以通过搭建模块化电路来实现,因此当前的算法框架不再是解决逻辑门与逻辑门之间的 NP 问题,而是解决多个电路模块之间的 NP 问题.

参考文献

- [1] LENT C S, TOUGAW P D. A device architecture for computing with quantum dots[J]. *Proceedings of the IEEE*, 1997, 85(4): 541-557.
- [2] ZHANG Y Q, XIE G J, SUN M B, et al. Design of normalised and simplified FAs in quantum-dot cellular automata[J]. *The Journal of Engineering*, 2017, 2017(10): 557-565.
- [3] CHO H, SWARTZLANDER E E. Adder and multiplier design in quantum-dot cellular automata[J]. *IEEE Transactions on Computers*, 2009, 58(6): 721-727.
- [4] ABUTALEB M M. A novel configurable flip flop design using inherent capabilities of quantum-dot cellular automata[J]. *Microprocessors and Microsystems*, 2018, 56: 101-112.
- [5] SABBAGHI-NADOOSHAN R, KIANPOUR M. A novel QCA implementation of MUX-based universal shift register[J]. *Journal of Computational Electronics*, 2014, 13(1): 198-210.
- [6] TOUGAW D, JOHNSON E W, EGLEY D. Programmable logic implemented using quantum-dot cellular automata[J]. *IEEE Transactions on Nanotechnology*, 2012, 11(4): 739-745.
- [7] SANDHU A, GUPTA S. A majority gate based RAM cell design with least feature size in QCA[J]. *Gazi University Journal of Science*, 2019, 32(4): 1150-1165.
- [8] NIEMIER M T, KOGGE P M. Problems in designing with QCAs: Layout = timing[J]. *International Journal of Circuit Theory and Applications*, 2001, 29(1): 49-62.
- [9] VANKAMAMIDI V, OTTAVI M, LOMBARDI F. Two-dimensional schemes for clocking/timing of QCA circuits [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(1): 34-44.
- [10] CAMPOS C A T, MARCIANO A L, VILELA NETO O P, et al. USE: A universal, scalable, and efficient clocking scheme for QCA[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016, 35(3): 513-517.
- [11] GOSWAMI M, MONDAL A, MAHALAT M H, et al. An efficient clocking scheme for quantum-dot cellular automata[J]. *International Journal of Electronics Letters*, 2020, 8(1): 83-96.
- [12] DENG F F, XIE G J, CHENG X, et al. CFE: A convenient, flexible, and efficient clocking scheme for quantum-dot cellular automata[J]. *IET Circuits, Devices & Systems*, 2020, 14(1): 88-92.
- [13] WALTER M, WILLE R, TORRES F S, et al. Fiction: An open source framework for the design of field-coupled nanocomputing circuits[EB/OL]. [2021-09-01]. <https://arxiv.org/abs/1905.02477>.
- [14] FORMIGONI R E, FERREIRA R S, NACIF J A M. Ropper: A placement and routing framework for field-coupled nanotechnologies[C]//2019 32nd Symposium on Integrated Circuits and Systems Design (SBCCI). Piscataway: IEEE, 2019: 1-6.
- [15] NGUYEN J, RAVICHANDRAN R, LIM S K, et al. Global placement for quantum-dot cellular automata based circuits[J]. *Georgia Institute of Technology*, 2003: 1-17.
- [16] ANTONELLI D A, CHEN D Z, DYSART T J, et al.

Quantum-Dot Cellular Automata (QCA) circuit partitioning: Problem modeling and solutions[C]//Proceedings of the 41st Annual Design Automation Conference. San Diego: IEEE, 2004: 363-368.

- [17] RAVICHANDRAN R, LIM S K, NIEMIER M. Automatic cell placement for quantum-dot cellular automata[J]. Integration, 2005, 38(3): 541-548.
- [18] WALTER M, WILLE R, GROBE D, et al. An exact method for design exploration of quantum-dot cellular automata[C]//2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). Piscataway: IEEE, 2018: 503-508.
- [19] WALTER M, WILLE R, TORRES F S, et al. Scalable design for field-coupled nanocomputing circuits[C]//Proceedings of the 24th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2019: 197-202.
- [20] WALTER M, HAASWIJK W, WILLE R, et al. One-pass synthesis for field-coupled nanocomputing technologies [C]//Proceedings of the 26th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2021: 574-580.
- [21] TRINDADE A, FERREIRA R, NACIF J A M, et al. A placement and routing algorithm for quantum-dot cellular automata[C]//Proceedings of the 2016 29th Symposium on Integrated Circuits and Systems Design. Belo Horizonte: IEEE, 2016: 1-6.
- [22] FONTES G, SILVA P A R L, NACIF J A M, et al. Placement and routing by overlapping and merging QCA gates [C]//2018 IEEE International Symposium on Circuits and Systems. Piscataway: IEEE, 2018: 1-5.
- [23] AMARU L, GAILLARDON P E, MICHELI G D. The EPFL combinational benchmark suite[C]//Proceedings of the 24th International Workshop on Logic & Synthesis, Mountain View: EPFL, 2015: 1-5.



彭斐男, 1985年出生, 河北保定人. 博士研究生. 主要研究方向为电子设计自动化、场耦合纳米计算.

E-mail: fpeng1985@126.com



韩倩女, 1997年出生, 江苏宿迁人. 硕士研究生. 主要研究方向为QCA电路的研究和高速数据的传输.

E-mail: 2210202992@qq.com



李小帅女, 1989年出生, 河南开封人, 讲师, 博士研究生, 主要研究方向为通信对抗, 车联网通信.

E-mail: :xiaoshuaihit@126.com



解光军男, 1970年出生, 安徽合肥人. 教授, 博士. 主要研究方向为纳米器件与电路、集成电路设计. 中国电子学会会员编号: E190004993S.

E-mail: gjxie8005@hfut.edu.cn

作者简介



李杨帅男, 1996年出生, 安徽合肥人. 硕士研究生. 主要研究方向为场耦合纳米电路自动布局布线算法.

E-mail: 1258173567@qq.com