

基于单元分类的CMOL常开缺陷单元容错映射

顾贤贵, 夏银水*

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: CMOS 纳米分子混合电路(CMOS/nanowire/MOLEcular hybrid circuits, CMOL)在制造过程中会引入较高缺陷率, 从而导致可用映射资源的减少. 针对由此产生的映射困难问题, 本文采用单元分类思想, 对部分缺陷单元加以利用, 以增加可映射单元数, 进而提高映射成功率. 首先根据单元缺陷类型的差异, 将缺陷单元分为可用和不可用两类进行标记, 然后对可用缺陷单元加以利用, 并采用改进的进化算法完成单元容错映射. 实验结果表明, 与已有方法相比, 新方法在运行效率和成功率上分别得到了 19.17% 和 30.14% 的提升.

关键词: CMOL; 进化算法; 单元容错映射

中图分类号: TP391.41

文献标志码: A

文章编号: 1001-5132 (2020) 01-0051-07

随着 CMOS 器件特征尺寸的不断缩小, 量子效应加剧, 制造成本飞涨等问题日益凸显, 使 CMOS 工艺的发展面临着前所未有的挑战^[1]. CMOL 电路因兼具 CMOS 电路丰富的逻辑功能和纳米器件的高集成度, 受到了国内外专家学者的广泛关注. 研究表明, 在功耗可控情况下, CMOL 电路每秒每平方厘米的运算次数可达 10^{20} 次, 被认为是后摩尔时代最具前景的 CMOS 替代工艺之一^[2]. 然而, 自底向上自组装而成的 CMOL 电路在制造过程中因界面引脚与纳米线难以精确校准, 及纳米器件尺寸极度缩小等原因, 常会引入 $10^{-3} \sim 10^{-1}$ 大小的缺陷率, 这远高于传统 CMOS 电路 $10^{-9} \sim 10^{-7}$ 大小的缺陷率, 因此, CMOL 单元容错映射方法研究是 CMOL 工艺迈向实用化进程的必要条件之一^[3].

CMOL 电路的单元容错映射通常分为两步进行. 首先考虑连通域约束, 完成无缺陷单元映射, 然后再考虑缺陷情况以及连通域约束, 完成单元容错映射. 其中, CMOL 电路缺陷类型通常可分为两类: 常开缺陷(stuck-at-open defects)和常连缺陷(stuck-at-close defects). 迄今为止, 针对常开缺陷的单元容错映射工作已取得较大进展^[4-7], 但针对常连缺陷的单元容错映射研究则相对较少^[8-9].

Hung 等^[8]将单元容错映射问题转变可满足性问题, 并将提出的容错策略转化为布尔约束, 采用一种称为 MINISAT 的 CHAFF 型 SAT 求解器进行 CMOL 单元容错映射的求解工作, 但该方法存在求解效率低, 电路求解规模小等问题. 后续研究表明, 常连缺陷存在缺陷传播现象, 对此, Chen 等^[9]提出一种有效的常连缺陷传播阻断方法, 利用反相器对的互补信号阻断常连缺陷传播路径, 但引入的反相器对易造成电路时延退化.

上述工作共同之处在于均采用舍弃策略处理缺陷单元, 但随缺陷率的提高, 可映射资源减少造成电路映射成功率急剧降低. 针对此问题, 本文提出单元分类思想, 根据单元缺陷类型的差异, 将缺陷单元分为可用和不可用两类, 并进行标记. 对不可用缺陷单元进行舍弃, 而对可用缺陷单元加以利用, 同时采用改进的进化算法完成 CMOL 常开缺陷的单元容错映射. 结果表明, 新方法能够快速有效的完成 CMOL 电路的单元容错映射.

1 CMOL 结构与单元容错映射

1.1 CMOL 结构

CMOL 电路是一种结合纳米器件、纳米线阵列

收稿日期: 2019-07-22.

宁波大学学报(理工版)网址: <http://journalg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61571248).

第一作者: 顾贤贵(1995-), 男, 浙江宁波人, 在读硕士研究生, 主要研究方向: 逻辑电路综合与优化. E-mail: 1014406105@qq.com

*通信作者: 夏银水(1963-), 男, 浙江余姚人, 研究员, 主要研究方向: 集成电路综合与优化. E-mail: xiayinshui@nbu.edu.cn

以及 CMOS 工艺的纳米分子混合结构,通过自底向上的自组装技术制造而成.图 1(a)为其截面图,底层为 CMOS 电路,顶层为正交排布的纳米线阵列,通过激活或关断可编程纳米二极管来控制单元之间信号通路的导通或阻断,CMOS 层与纳米线层则通过高度不同的锥形引脚进行连接.考虑到远距离传输造成的信号衰减,以及界面引脚与纳米线的连接唯一性,纳米线层与 CMOS 层按一定 θ 角度呈周期性断裂排布,其中 $\theta = \arcsin(F_{\text{NANO}} / (\beta F_{\text{CMOS}}))$, F_{NANO} 为相邻 2 条纳米线间距的 1/2, F_{CMOS} 为相邻 2 个 CMOS 子系统间距的 1/2, β 是一个大于 1 且与 CMOS 子系统复杂度相关的无量纲因子.这种特殊的纳米线排布方式造成每个 CMOL 单元只能与其附近的 $W = 2r(r-1) - 1$ 个其他单元进行连接,构成所谓的连通域约束,其中 r 为连通域半径.图 1(b)为 CMOL 俯视图,取连通域半径 $r=3$ 为例,深色单元的连通域为其周边 11 个浅色单元所构成区域.同时 CMOL 电路的基本逻辑由纳米线和 CMOS 反相器共同完成,其中纳米线实现“线或”逻辑,CMOS 反相器完成取“非”操作,两者共同实现“或非/非”逻辑.

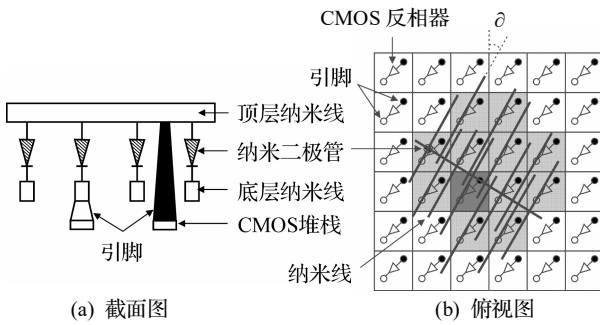


图 1 CMOL 结构^[1]

1.2 CMOL 单元容错映射

CMOL 单元容错映射一般分两步进行:无缺陷映射和单元容错映射.首先在无缺陷映射前,需将基于“与/或/非”形式的待映射逻辑电路 G 进行逻辑转换,得到基于“或非/非”形式的等效电路.然后在给定无缺陷 CMOL 阵列 Γ 上完成单元映射,即找到一个从 G 到 Γ 的映射关系,使得满足:

$$\forall g_i, g_j \in V: (g_i \neq g_j) \rightarrow (p(g_i) \neq p(g_j)), \quad (1)$$

$$\forall e(g_i, g_j) \in E: M_{\text{dist}}(p(g_i), p(g_j)) \leq r, \quad (2)$$

其中, V 表示电路节点(逻辑门)集合; E 表示电路节点连接边集合; $p(g_i)$ 表示电路节点 g_i 所映射的

CMOL 单元; M_{dist} 表示 2 个 CMOL 单元之间的曼哈顿距离; r 表示连通域半径.式(1)说明每个 CMOL 单元能且只能被 1 个电路节点所映射;式(2)表示有连接关系的 2 个电路节点所映射单元之间的距离必须满足连通域约束.

在无缺陷映射阶段,得到满足式(1)、(2)的初始映射解,然后在该解中加入缺陷信息,通过单元重构完成 CMOL 单元容错映射^[4,9].

本文涉及的缺陷类型表述如下:

(1)纳米二极管常开.纳米二极管无法正常开启,致使其连接的 2 条纳米线所处单元之间的信号通路被阻断.

(2)纳米线非周期性断裂.纳米线发生不规则断裂,致使其所处单元的可连接单元数减少.

(3)CMOS 反相器常开.连接 CMOS 反相器的界面引脚未与纳米线精确对准,致使其所在单元无法实现逻辑非功能.

在 CMOL 电路中正是由于上述常开缺陷的存在,使得原本可连接的 2 个 CMOL 单元之间的信号通路被阻断,或无法完成相应的逻辑运算. CMOL 单元容错映射就是一个在连通域约束下,通过一定方法消除缺陷影响,从而保证映射电路逻辑功能正确的过程.

2 基于单元分类的 CMOL 单元容错映射

2.1 CMOL 单元分类与标记

在 CMOL 电路中,每个 CMOL 单元可看作由 CMOS 反相器、输入/输出纳米线以及纳米二极管三部分组成,如果其中任一部分发生缺陷,那么该 CMOL 单元则被视为一个缺陷单元.但不同的常开缺陷所产生的影响也有所不同,其中,CMOS 反相器常开意味着该反相器所组成的 CMOL 单元无法实现逻辑非功能,所以此类缺陷单元无法被利用.结合 CMOL 电路结构,每个纳米二极管唯一确定 2 条正交纳米线,每条纳米线唯一确定 1 个 CMOL 单元,纳米二极管常开造成其所连接的 2 条纳米线所确定的 2 个 CMOL 单元之间的信号通路被阻断,但这 2 个单元仍可与其他单元进行连接,并构成信号通路,因此此类缺陷单元可被利用.同时,纳米线非周期性断裂表示纳米线在非周期性断点处发生不规则断裂,此类缺陷仅造成纳米线唯一确定

CMOL 单元的可连接单元数减少, 所以此类缺陷单元也可被利用. 综上所述, 根据CMOL单元是否发生缺陷以及发生缺陷时的影响差异, 可将单元分为以下三类: 无缺陷单元、可用缺陷单元和不可用缺陷单元. 其中,

(1)无缺陷单元: 表示该 CMOL 单元未发生任意一种常开缺陷;

(2)不可用缺陷单元: 表示该 CMOL 单元发生CMOS反相器常开缺陷;

(3)可用缺陷单元: 表示该 CMOL 单元的纳米二极管常开, 或纳米线非周期性断裂, 但不存在CMOS反相器常开缺陷.

记 $c_{\text{CMOS}}=1$, $c_{\text{NANO}}=1$, $c_{\text{WIRE}}=1$ 分别表示CMOL单元 c 中的CMOS反相器、输入/输出纳米线和纳米二极管发生常开缺陷; ψ_0, ψ_1, ψ_2 分别表示无缺陷单元集合、可用缺陷单元集合和不可用缺陷单元集合. CMOL 单元分类与标记过程如算法1所示.

算法1(CMOL单元分类与标记)

1. For 所有 CMOL 单元 c
2. 检测单元 c 缺陷情况;
3. If $c_{\text{CMOS}}=1$ then
4. $c \in \psi_2$;
5. Else If $c_{\text{NANO}}=1$ 或 $c_{\text{WIRE}}=1$ then
6. $c \in \psi_1$;
7. Else
8. $c \in \psi_0$;
9. End If
10. End For

2.2 可用缺陷单元的映射约束与重构

针对不可用缺陷单元, 因其无法进行逻辑运算, 所以在单元容错映射阶段需要将其舍弃, 即不可被任何电路节点所映射. 针对可用缺陷单元, 由于其CMOS反相器功能是正常的, 仍可完成相应的逻辑运算, 所以其依然可被电路节点所映射. 但任意2个可用缺陷单元 a 和 b , 如果在彼此的输入/输出连通域内, 那么在单元映射过程中将可能出现以下4种情况:

- (1) a 、 b 均未被电路节点所映射;
- (2) a 、 b 其中一个被电路节点映射, 而另一个未被映射;
- (3) a 、 b 两者都被电路节点映射, 但映射的电

路节点不存在连接关系;

(4) a 、 b 两者都被电路节点映射, 但映射的电路节点存在连接关系.

由于可用缺陷单元 a 和 b 之间无法构成信号通路, 这就要求两者不能被具有连接关系的电路节点 g 和 g' 同时映射. 因此, 上述第4种情况会因信号通路阻断而出现逻辑错误, 其他3种情况因单元 a 和 b 之间不存在信号传输, 故不会导致逻辑错误. 因此, 在单元映射过程中, 为防止第4种情况出现, 需要对在彼此输入/输出连通域内的2个可用缺陷单元的映射进行约束施加, 表述如下:

$$\forall g' \in FO(g), c' \in OD(c): p(g) = c \rightarrow p(g') \neq c', \quad (3)$$

$$\forall g'' \in FI(g), c'' \in ID(c): p(g) = c \rightarrow p(g'') \neq c'', \quad (4)$$

其中, $FO(g)$ 和 $FI(g)$ 分别表示电路节点 g 的扇出集合和扇入集合; $OD(c)$ 和 $ID(c)$ 分别表示可用缺陷单元 c 的输出连通域和输入连通域; $p(g)$ 表示电路节点 g 所映射单元. 式(3)表示如果电路节点 g 映射于一个可用缺陷单元 c 上, 那么其扇出 g' 不可映射于可用缺陷单元 c 输出连通域内的任何一个可用缺陷单元 c' 上. 式(4)表示如果电路节点 g 映射于一个可用缺陷单元 c 上, 那么其扇入 g'' 不可映射于可用缺陷单元 c 输入连通域内的任何一个可用缺陷单元 c'' 上. 倘若可用缺陷单元的映射出现违反约束(3)(4)的情况, 则会导致映射电路逻辑出错. 因此对于违反约束(3)(4)的映射情况, 需要通过单元重构来消除常开缺陷带来的影响. 同时对于可用缺陷单元是否需要单元重构, 则需进行约束违反情况的判定, 判定过程如算法2所示, 其中集合 Φ 用来标记需要进行重构的单元.

算法2(可用缺陷单元的重构判定)

1. For 所有 $c \in \psi_1$
2. 检测单元 c 映射情况;
3. If $p(g)=c$ then
4. 搜索 $g' \in FO(g)$ 所在单元 $p(g')$;
5. If $p(g')=c'$ and $c' \in OD(c)$ then
6. $g' \in \Phi$;
7. 搜索 $g'' \in FI(g)$ 所在单元 $p(g'')$;
8. If $p(g'')=c''$ and $c'' \in ID(c)$ then
9. $g'' \in \Phi$;

10. End If
11. End If
12. End For

以图 2 为例, 对单元重构进行详细说明. 图中仅显示了部分纳米线和纳米二极管, 其中左上角数字表示单元编号, 右下角字母表示映射逻辑. 取连通域半径 $r=3$ 为例, 则单元 8 的输入连通域为图中粗实线框定区域内 11 个单元所构成的集合. 纳米二极管 n_4 位于单元 6 输出纳米线和单元 8 输入纳米线交叉点处, 其发生常开缺陷, 造成单元 6 与单元 8 之间的信号通路被阻断, 而两者的 CMOS 反相器功能正常, 所以单元 6 和 8 为可用缺陷单元, 即单元 $6, 8 \in \psi_1$. 类似于纳米二极管 n_4 的分析过程, 纳米二极管 n_3 也发生常开缺陷, 则单元 5 为可用缺陷单元, 即单元 $5 \in \psi_1$. 此外, 单元 3 的 CMOS 反相器发生常开缺陷, 则单元 3 无法完成逻辑运算, 因而单元 3 为不可用缺陷单元, 即单元 $3 \in \psi_2$. 其他单元均不存在任何常开缺陷, 则为无缺陷单元, 即单元 $1, 2, 4, 7, 9, 10, 11, 12 \in \psi_0$. 初始逻辑 $F = \overline{A + B}$ 已映射于图中相应单元上, 由于逻辑信号 B 的输出节点 g_B 与逻辑信号 F 的输出节点 g_F 存在连接关系, 而 g_B 和 g_F 映射的单元 6 和单元 8 均为可用缺陷单元, 这种映射情况违反了约束(3)(4), 导致信号 F 逻辑出错, 因此需要对 g_B 或 g_F 进行单元重构. 以 g_B 的重构过程为例, g_B 可重新映射于单元 8 输入连通域内其他任意一个未被映射且无缺陷单元上, 即可消除缺陷影响.

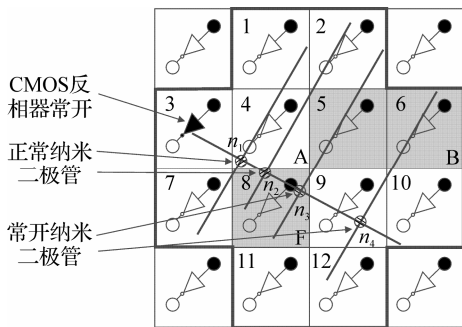


图 2 单元重构

2.3 算法实现

给定一个 CMOL 阵列 Γ 和待映射电路 G , 首先利用本课题组的 LRMA 算法^[10]生成满足连通域约束的初始映射解. 然后在初始映射解中加入缺陷图 T , 利用算法 1 对单元进行分类与标记, 接着

利用算法 2 对可用缺陷单元进行重构判定. 考虑到进化算法强大的全局搜索能力, 最后采用改进的进化算法对违反约束(3)(4)的映射情况进行单元重构.

进化算法具体实现以适值函数、选择策略和遗传算子三部分做详细说明.

2.3.1 适值函数

适值是度量个体性能状态优良情况的重要参数. 在 CMOL 单元容错映射中, 每一个电路映射结果可视为一个体, 使用本课题组的 LRMA 算法^[10]生成 N 个体构成初始种群. 个体违反约束(3)(4)的个数越少, 性状越优; 否则, 性状越差. 适值函数定义如下:

$$F_{\text{fitness}} = \sum_{i=1}^m \sum_{j=1}^n P_{\text{penalty}}(c_i, c_j), \quad (5)$$

其中, m 表示逻辑电路的节点总数; n 表示电路节点 i 所映射单元连通域内的单元总数; $P_{\text{penalty}}(c_i, c_j)$ 表示电路节点 i 所映射单元与其连通域内单元之间违反约束(3)(4)的惩罚因子, 其中,

$$P_{\text{penalty}} = \begin{cases} 1, & \text{if violate (3) or (4),} \\ 0, & \text{others.} \end{cases} \quad (6)$$

2.3.2 选择策略

选择策略决定着种群演化方向. 为确保性能状态优的个体有较大概率, 同时性能状态差的个体也有不为零的概率被选择进行遗传操作, 需采用线性排名结合轮盘赌策略进行个体选择. 首先计算种群所有个体适值, 并进行升序排列. 记种群规模为 N , 则性状最优个体编号为 N , 且设定计算值为 $p_{\max}$, 性状最差个体编号为 1, 且设定计算值为 $p_{\min}$ ($p_{\min} < p_{\max}$). 接着根据下式计算其他个体计算值 p_i :

$$p_i = p_{\min} + (p_{\max} - p_{\min}) \frac{i-1}{N-1}, \quad (7)$$

随后对所有个体的计算值 p_i 进行归一化处理, 得到个体选择概率 q_i ,

$$q_i = p_i / \sum_{i=1}^N p_i, \quad (8)$$

根据 q_i 进行轮盘选择, 令

$$qq_0 = 0, qq_i = qq_{i-1} + q_i, \quad (9)$$

每次选择, 轮盘旋转 N 次, 每次旋转随机生成一个介于 (0,1) 间的浮点数 κ , 若满足 $qq_{i-1} \leq \kappa < qq_i$, 则个体 i 出现次数增 1. N 次旋转结束后, 选择出现次

数最多的 2 个个体作为父代进行遗传操作。

2.3.3 遗传算子

交叉是进化算法重要的遗传算子之一, 通过随机交换父代染色体部分基因而产生新的基因组合, 使性能状态发生改变。每个 CMOL 电路映射结果可视为一个染色体, 其中每个单元的映射情况可视为一个基因。有别于常规染色体的链状结构, CMOL 电路二维染色体结构无法直接套用针对链状染色体的交叉模式, 所以本文采用二维交叉策略。交叉区域设定为矩形区域, 随机选取不在同一行(列)的 2 个 CMOL 单元作为矩形的 2 个顶点, 由此唯一确定一块矩形区域作为交叉区域。考虑到交叉区域过大会导致具有优良性状的个体结构在进化过程中易被破坏, 而过低会降低算法的收敛速度, 因此需要合理设定交叉区域大小。由经验所得, 交叉区域对角线长度 M_{dist} 满足 $r/2 \leq M_{\text{dist}} \leq r$ (r 为连通域半径)时, 具有较好的交叉效果, 因此选取作为顶点的 2 个 CMOL 单元曼哈顿距离满足 $r/2 \leq M_{\text{dist}} \leq r$ 时所确定的交叉区域为合理区域; 否则, 需要重新确定交叉区域。此外 CMOL 电路存在连通域约束, 不受限制的交叉会造成连通域约束违反个数的增加, 鉴此需要加入连通域约束检查机制, 拒绝违反连通域约束的基因互换。图 3 为二维染色体交叉示意图, 深色区域为交叉区域, 按箭头对应情况进行交叉操作。如父代 1 中基因 3 对应父代 2 中基因 4, 则在父代 1 中将基因 3 与基因 4 所在位置互换, 同时在父代 2 中将基因 4 与基因 3 所在位置互换。由于交叉分别发生在父代 1 与父代 2 自身染色体中, 因此可规避基因重复问题。假设父代 1 与父代 2 交叉过程中的基因互换均未违反连通域约束, 则按照上述说明进行二维染色体交叉, 得到子代 1 与子代 2。

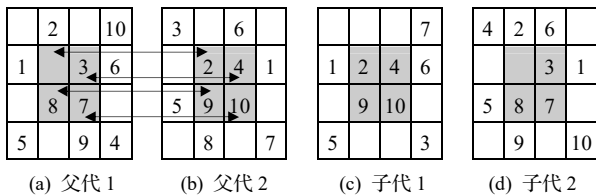


图 3 二维染色体交叉

在进化算法中, 变异算子的主要功能是拓宽算法的选择空间, 以防止陷入局部最优, 因此变异阶段的单元互换不存在连通域约束检查机制。此

外, 单元变异率过高则可能导致算法无法收敛, 过低则可能限制了算法的求解能力, 所以本文设定变异算子仅在种群中性状最优个体的适值未得到改进的情况下被激活实施。在 CMOL 单元容错映射过程中, 变异只会造成映射电路结构微小的变化, 即在染色体中随机选取 2 个基因进行位置互换。图 4 所示为二维染色体变异示意图, 其中存在 2 种变异类型: (1)若 2 个单元都已被电路节点所映射, 则互换位置, 如图 4(a)所示, 将电路节点 4 和 8 位置互换; (2)若 2 个单元其中一个为空白单元, 则将电路节点重新映射于空白单元上即可, 如图 4(b)所示, 将电路节点 7 重新映射于箭头所指单元上。

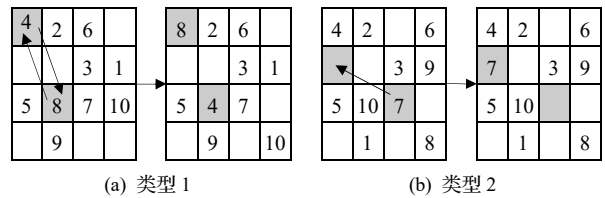


图 4 二维染色体变异

综上所述, 给出基于单元分类的 CMOL 单元容错映射伪代码的算法 3。

算法 3(基于单元分类的 CMOL 单元容错映射)

1. 利用本课题组的 LRMA 算法^[10]生成 N 个初始解构成初始种群 φ ;
2. 在初始解中加入缺陷图 T , 利用算法 1 进行单元分类与标记;
3. For M 次迭代
 4. 利用算法 2 进行重构判定;
 5. 计算每一个体适值函数 f_i ;
 6. 利用选择策略选取父代 P_1 、 P_2 ;
 7. 交叉、变异生成子代 C_1 、 C_2 ;
 8. If 子代适值函数 $f_c > 0$ then
 9. 更新种群;
 10. 进行下一轮迭代;
 11. Else If 子代适值 $f_c = 0$ then
 12. 输出性状最优个体的映射结果;
 13. Break;
 14. End If
 15. End For

3 实验结果

本文采用的进化算法采用 C 语言编程实现,

在 Inter Core 3.40 GHz, 2 GB 内存 Windows 平台下运行测试. 进化算法是一种随机搜索算法, 插入不同的时间种子将会产生不同的运行结果, 解的质量也不尽相同. 因此, 按每个 ISCAS^[11]标准测试电路进行 50 次试验, 并取平均值作为最终测试结果. 为比较的公平性, 缺陷率和连通域半径与比较文献保持一致, 其中 CMOS 反相器缺陷率为 10%, 纳米线缺陷率为 10%, 纳米二极管缺陷率为 10%, 连通域半径取 $r=18$, 可用缺陷单元和不可用缺陷单元缺陷率均设置为 10%. 对于每个单元随机生成一个介于[1,100]之间的整数 σ , 如果 $\sigma \in [1,10]$, 则视为可用缺陷单元; 如果 $\sigma \in [11,20]$, 则视为不可用缺陷单元; 如果 $\sigma \in [21,100]$, 则视为正常单元.

实验结果见表 1. 其中, “单元”表示标准测试电路节点总数; “时间”表示得到可行解时程序的运行时间; “线长”表示映射电路互连线长度总和; “成功率”表示 100 次试验中成功得到映射解的概率; “平均值”表示不同参数的平均值; “提高量”

表示提出方法与比较文献的改进百分比. 同时选取了 2 篇在 CMOL 常开缺陷单元容错映射领域具有代表性的文献(SimE^[4]和 DUM^[7])进行实验对比. 由表中数据可见, 与 SimE 相比, 本文方法得到了 26.42%的时间和 2.77%的线长优化. 但线长优化效果不明显, 主要原因是新方法在进化算法变异阶段不存在连通域约束检查机制, 导致映射解中存在部分违反连通域约束情况, 对此需要引入反相器对进行路由拓展而产生额外的线长消耗^[12]. 与 DUM相比, 得到了 19.17%的时间和 30.14%的成功率提升, 线长却存在 27.31%的退化. 这主要是因为 DUM 采用缺陷无意识的映射方法, 即在 $N \times N$ 大小的缺陷 CMOL 阵列中提取出一块 $K \times K (K < N)$ 大小的无缺陷子阵列, 然后在无缺陷子阵列中进行单元映射, 与此不同的是, 新方法在整个 $N \times N$ 大小的缺陷阵列中寻找映射解. 因此得到最终映射解的紧凑程度要低于 DUM, 而线长是反应电路映射紧凑程度的重要指标, 映射结果越紧凑, 线长越

表 1 不同方法实验结果比较

电路名	单元	SimE ^[4]			DUM ^[7]			本文方法		
		时间/s	线长	成功率/%	时间/s	线长	成功率/%	时间/s	线长	成功率/%
s27	19	2.06	94	100	1.03	62	100	1.95	89	100
s208	136	5.95	1 716	100	2.46	1 438	100	3.54	1 616	100
s298	122	4.55	2 331	100	2.39	1 697	100	3.42	2 104	100
s344	180	2.80	2 867	100	2.97	2 331	100	3.76	2 808	100
s349	184	8.52	2 941	100	2.56	2 247	100	4.21	2 694	100
s382	175	6.10	3 654	100	3.16	2 471	100	4.86	3 057	100
s386	164	6.54	3 058	100	3.67	2 796	100	4.16	3 157	100
s400	188	7.71	3 261	100	3.91	2 824	100	5.11	3 361	100
s420	299	7.45	4 762	100	5.26	4 094	100	5.84	4 503	100
s444	187	6.01	3 342	100	4.13	2 731	100	5.47	3 156	100
s510	304	39.04	6 130	100	32.74	4 649	92	27.48	5 730	100
s526	273	14.26	6 483	100	13.28	4 957	100	9.48	5 883	100
s641	302	14.32	5 108	100	12.97	4 070	100	11.21	5 005	100
s713	321	42.52	6 612	100	39.42	4 478	78	29.87	6 492	100
s820	447	48.37	11 698	100	41.56	9 076	64	37.16	11 097	100
s832	454	67.43	13 468	100	62.87	10 027	56	46.79	13 871	100
s838	606	120.10	14 513	100	98.16	11 062	44	76.15	14 489	100
s1196	675	213.79	18 601	100	224.16	13 077	18	156.57	18 964	100
s1238	724	337.85	18 145	100	312.68	14 269	22	246.75	17 145	100
平均值		50.28	6 778	100	45.76	5 176	76.84	36.99	6 590	100
提高量		26.42%	2.77%	0	19.17%	-27.31%	30.14%	-	-	-

小, 反之, 线长越大. 本文将缺陷单元细分为可用和不可用两类, 对可用缺陷单元加以利用, 映射资源的增加使算法更易找到有效解, 这是求解效率得到提升的主要原因.

4 结语

针对 CMOL 常开缺陷的单元容错映射问题, 本文采用缺陷单元分类处理思想, 提高了单元利用率从而提高了映射效率和求解成功率. 但提出方法仅限于常开缺陷的容错映射, 因此涵盖常开与常连缺陷的方法研究是未来的主要研究方向.

参考文献:

- [1] Likharev K K, Strukov D B. CMOL: Devices, Circuits and Architectures[M]. Lecture Notes in Physics. Heidelberg: Springer, 2005.
- [2] Lu W, Lieber C M. Nanoelectronics from the bottom up[J]. Nature Materials, 2007, 6(11):841-850
- [3] Yan H, Choe H S, Nam S W, et al. Programmable nanowire circuits for nanoprocessors[J]. Nature, 2011, 470(7333):240-244.
- [4] Sait S M, Arafeh A M. Reconfiguration-based defect-tolerant design automation for hybrid CMOS/nanofabrics circuits using evolutionary and non-deterministic heuristics[J]. Arabian Journal for Science and Engineering, 2015, 40(9):1-15.
- [5] Zha X, Xia Y. Genetic algorithm based on divide-and-conquer strategy for defect-tolerant CMOL mapping[C]. Proceedings of the 12th IEEE International Conference on ASIC. Los Alamitos: IEEE Computer Society Press, 2017:863-866.
- [6] 汪纪波, 夏银水, 储著飞, 等. 基于门节点分级选择的 CMOL 电路单元快速容错映射[J]. 计算机辅助设计与图形学学报, 2017, 29(1):172-179.
- [7] 陈定亨, 夏银水, 储著飞. 缺陷无意识的 CMOL 单元容错映射[J]. 计算机辅助设计与图形学学报, 2017, 29(11):2133-2139.
- [8] Hung W N N, Gao C, Song X, et al. Defect-tolerant CMOL cell assignment via satisfiability[J]. IEEE Sensors Journal, 2008, 8(6):823-830.
- [9] Chen D, Xia Y, Wang Z. Stuck-at-close defect propagation and its blocking technique in CMOL cell mapping[J]. Microelectronics Journal, 2018, 72(2):100-108.
- [10] Xia Y, Chu Z, Hung W N N, et al. An integrated optimization approach for nanohybrid circuit cell mapping[J]. IEEE Transactions on Nanotechnology, 2011, 10(6):1275-1284.
- [11] Brglez F, Bryan D, Koźmiński K. Combinational profiles of sequential benchmark circuits[C]. Proceedings of IEEE International Symposium on Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 1989:1929-1934.
- [12] Chu Z, Xia Y, Hung W N N, et al. Timing-driven logic restructuring for nano-hybrid circuits[J]. International Journal of Electronics, 2013, 100(5):669-685.

Cell classification based CMOL stuck-at-open defect-tolerant cell mapping

GU Xiangui, XIA Yinshui*

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: To tackle the problem of high defect rate in CMOS/nanowire/MOLEcular hybrid circuits (CMOL), a cell classification based defect-tolerant cell mapping method is proposed. First, recoverable and unrecoverable defective cells are marked, and the recoverable defective ones are put to use again. Then, a modified evolutionary algorithm is used to complete the defect-tolerant cell mapping. Compared with the existing approaches, the CPU time and success rate using the proposed method are optimized to 19.17% and 30.14%, respectively.

Key words: CMOL; evolutionary algorithm; defect-tolerant cell mapping

(责任编辑 章践立)