面向功耗优化的 CMOL 电路容错映射

谢尚奎, 夏银水*, 查晓婧

(宁波大学信息科学与工程学院 宁波 315211)
(xiayinshui@nbu.edu.cn)

摘要: 针对 CMOS/纳米线/分子混合(CMOL)电路的缺陷导致电路功耗增加这一问题, 提出基于单元限用的容错映射方法. 首先建立缺陷对的功耗模型, 分析常连缺陷对的映射模式对功耗的影响; 然后通过高功耗单元的限用与功耗约束的设置, 以减少高成本映射模式带来的功耗开销; 最后采用改进的遗传算法完成电路容错映射. ISCAS 标准测试电路的实验结果表明, 所提方法在成功容错映射的基础上, 有效地减少了电路的功耗与面积, 同时对求解速度也有较好的优化.

关键词: CMOS/纳米线/分子混合(CMOL)电路; 容错映射; 功耗; 遗传算法
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2021.18525

Power Oriented Optimization for the Defect-Tolerant Mapping of CMOL Circuits

Xie Shangluan, Xia Yinshui*, and Zha Xiaojing

(School of Information Science and Engineering, Ningbo University, Ningbo 315211)

Abstract: Aiming at the power consumption increase problem from the defects of CMOS/nanowire/molecular hybrid (CMOL) circuits, a defect-tolerant mapping method based on cell limitation is proposed. First, the power consumption model of defect pairs is established and the effect of different mapping patterns of the defect pairs on power consumption is analyzed. Then, the use of power hungry cells is restricted and the power consumption constraint is set to reduce the power consumption overhead caused by the high cost mapping patterns. Finally, the modified genetic algorithm is chosen to implement the defect-tolerant mapping of CMOL circuits. The ISCAS benchmarks are tested for verification. The experimental results demonstrated that the proposed method effectively reduces the power consumption and area of CMOL circuits on the basis of successful defect-tolerance, with better optimization of solution speed.

Key words: CMOS/nanowire/molecular hybrid (CMOL) circuits; defect-tolerant mapping; power consumption; genetic algorithm

随着器件到达物理极限, 传统的集成电路技术面临着量子效应加剧、制造成本飞涨等诸多问题. 研究者认为, 利用新兴的纳米电子器件和相应的纳米电路, 可以获得更高的集成密度和工作频率, 从而延续集成电路的发展. 其中, CMOS/纳米

线/分子混合(CMOS/nanowire/molecular hybrid, CMOL)电路^[1-2]由于拥有 CMOS 丰富的逻辑功能, 又兼具纳米技术的高集成密度, 被认为是最有前途的 CMOS 替代技术之一, 引起了研究者的广泛兴趣^[3-6].

收稿日期: 2020-08-03; 修回日期: 2020-09-18. **基金项目:** 国家自然科学基金(61571248); 浙江省大学生科技创新活动计划(新苗人才计划)(2019R40507). 谢尚奎(1996—), 男, 硕士研究生, 主要研究方向为微纳电路容错设计; 夏银水(1963—), 男, 博士, 教授, 博士生导师, 论文通讯作者, 主要研究方向为集成电路综合与优化、能量俘获技术; 查晓婧(1994—), 女, 博士研究生, 主要研究方向为微纳电路容错设计.

由于难以精确地控制自底向上的自组装方式^[7], 再加上纳米电子器件尺寸的极度缩小, 因此, CMOL 电路制造过程中会出现常开缺陷、常连缺陷等物理缺陷^[8]. 研究表明, 纳米集成电路的缺陷率达 10%, 远远高于传统 CMOS 电路 $10^{-9} \sim 10^{-7}$ 的缺陷率^[9]. 鉴于此, 寻找高效的容错映射方法成为推动 CMOL 电路实用化进程的关键之一. 在 CMOL 电路中, 只有极少部分纳米器件参与到逻辑电路功能的实现, 大量纳米器件的冗余为容错提供了基础.

CMOL 的映射技术研究经历了无缺陷单元映射^[10-13]、常开缺陷的容错映射^[14-17]和多缺陷覆盖的容错映射^[18-19] 3 个阶段. 无缺陷单元映射只需考虑连通域约束^[13], 该阶段主要是结合经典算法完成映射, 如布谷鸟算法^[10]、禁忌搜索算法^[11]、粒子群优化算法^[12]和结合拉格朗日算子的文化基因算法^[13]等. Strukov 等^[17]以 32 bit Kogge-Stone adder 电路为例, 率先提出了可重构的容错映射思想. 针对常开缺陷, Sait 等^[14]分别采用模拟进化算法和禁忌搜索算法重构无缺陷初始解, 将电路整体作为求解对象, 但其求解速度慢. 陈定亨等^[15]通过 CMOL 阵列无缺陷区域提取进行缺陷无意识映射, 但因舍弃了大量映射资源, 映射成功率降低. 查晓婧等^[16]对缺陷单元进行动态分层并加以合理利用, 优化了电路线长. 然而, 上述方法仅为针对单一缺陷的容错. 相对而言, 多缺陷的容错方法目前研究较少. Hung 等^[18]将容错映射问题转化为可满足性 (satisfiability, SAT) 问题导入其求解器, 但是该方法仅能求解小电路. Chen 等^[19]利用嵌入了传播阻断方法的进化算法 (evolutionary algorithm with blocking method, EAB) 实现了多缺陷环境下的容错映射, 并成功消除了常连缺陷传播现象.

综上所述, 目前针对 CMOL 电路的容错映射研究主要是对求解速度、面积和线长等指标进行优化, 但鲜有功耗优化的工作. 因此, 本文以功耗为优化目标, 提出基于单元限用的 CMOL 电路容错映射方法, 通过高功耗单元的限用与功耗约束的设置, 节省传播阻断的功耗开销并避免耗能映射模式的发生. 利用遗传算法实现所提方法并在 ISCAS 标准测试电路上进行验证, 实验表明本文方法可有效地减少电路功耗与面积, 同时实现对求解速度的优化.

1 研究背景

1.1 CMOL 电路结构

CMOL 电路是一种结合了 CMOS 工艺与纳米

技术的类现场可编程门阵列 (field programmable gate array, FPGA) 结构. 图 1a 所示为 CMOL 电路截面图, 电路顶部为垂直交叉的纳米线阵列, 每个交叉点集成一个由 D 个单分子器件构成的可编程纳米二极管; 电路底部为 CMOS 单元堆栈, 通过锥形触点与纳米线连接. 图 1b 所示为 CMOL 电路俯视图, 每个 CMOS 单元由 1 个反相器与 2 个传输晶体管组成. 图 2 所示为电路的配置说明图, 利用 1 对锥形触点唯一确定 1 个纳米二极管, 通过控制相关 CMOS 连线的信号, 连接触点的 2 个传输晶体管共同作用, 可在纳米二极管两端产生 1 个“写”电压或“读”电压, 进行纳米二极管的编程或故障检测. 需要注意的是, 在配置阶段, 关闭所有反相器, 配置完成后传输晶体管被用作下拉电阻. 在 CMOL 电路中, 由 CMOS 单元进行信号反相, 纳米线进行“线或”操作, 实现“或非”逻辑运算.

图 1 CMOL 电路结构

图 2 CMOL 电路配置说明

为保证每条纳米线仅对应一个锥形触点, 纳米线阵列相对 CMOS 堆栈有 $\alpha = \arcsin(F_{\text{nano}}/\beta F_{\text{CMOS}})$ 的倾角, 同时按周期 $L = 2\beta^2 F_{\text{CMOS}}^2 / F_{\text{nano}}$ 规则断开. 其中, F_{nano} 为纳米线的半间距; F_{CMOS} 为 CMOS 子系统的半间距; β 为与 CMOS 单元复杂度相关的无量纲因子. 纳米线的断裂使任意单元只能与周边 $M = 2r(r-1)-1$ 个单元连接, 形成所谓的“连通域约束”, 其中, r 为连通域半径.

1.2 CMOL 电路功耗模型

CMOL 单元级间互连的 RC 等效模型如图 3 所示, 其中, R_{on}/D 和 R_{off}/D 分别为导通和关断状

态下纳米二极管的电阻; R_{wire} 为纳米线电阻; C_{wire} 为对应的电容; R_{pass} 为传输晶体管电阻; C_{in} 为反相器输入电容. Strukov^[20]给出了目前 CMOL 电路研究中普遍应用的参数, 如表 1 所示, 其中, C_0 和 R_0 分别为单位长度纳米线的电容和电阻.

图 3 CMOL 单元级间互连 RC 等效模型图^[20]

表 1 CMOL 电路参数

参数	取值	参数	取值
F_{CMOS}	45 nm	R_{off}/D	1.34 G Ω
F_{nano}	4.5 nm	C_0	0.2 fF/ μm
V_{DD}	0.3 V	R_0	2.78 k Ω / μm
R_{on}/D	260 k Ω	$L = 2\beta^2 F_{\text{CMOS}}^2 / F_{\text{nano}}$	14.4 μm

在 CMOL 静态工作时, 除了 CMOS 器件的泄漏电流外, 用作下拉电阻的传输晶体管、单元互连的纳米线阵列上也有电流产生. 利用 Hspice 对 NCSU 45 nm 标准单元库进行仿真, 结果显示 CMOS 反相器泄漏电流在 nA 级^[21], 而根据表 1 参数计算流经导通路径的电流 I_{on} 与流经关断路径的泄漏电流 I_{leak} 的大小分别在 μA 级与 nA 级. 由于纳米器件集成密度远大于 CMOS 单元, 因此可以认为 CMOL 电路的静态功耗主要由纳米线阵列产生. 同时, 根据 CMOL 电路器件寄生参数的研究表明^[20,22-23], 纳米二极管电容与 CMOS 单元输入电容的值不会超过 10^{-1} fF, 而 1 根纳米线的电容可达 2.9 fF, 因此认为动态功耗主要来自逻辑状态改变时纳米线寄生电容的充放电^[17,20,24-25]. 根据文献[20], CMOL 电路平均功耗可估算为

$$\begin{cases} P_{\text{on}} \approx N_{\text{D}} V_{\text{DD}}^2 / 2 (R_{\text{on}}/D + R_{\text{wire}} + R_{\text{pass}}) = N_{\text{D}} P_0 \\ P_{\text{dyn}} \approx N_{\text{D}} C_{\text{wire}} V_{\text{DD}}^2 f / 4 = N_{\text{D}} P_{\text{d}} \\ P_{\text{leak}} \approx m M V_{\text{DD}}^2 / 2 (R_{\text{off}}/D) \end{cases} \quad (1)$$

其中, P_{on} 为电流 I_{on} 导致的静态功耗; P_{dyn} 为电路的动态功耗; f 为逻辑操作的频率; N_{D} 为电流 I_{on} 流经的路径总数; P_0 和 P_{d} 分别对应 $N_{\text{D}}=1$ 时 P_{on} 和

P_{dyn} 的值; P_{leak} 为电流 I_{leak} 导致的静态功耗; m 为利用的单元总数; 每个单元输出连通域内电流 I_{leak} 流经的路径数近似为连通域 M 的大小. 公式中信号概率及信号翻转概率均设定为 1/2, 单元间的纳米线长度设定为 L , 因此 R_{wire} 和 C_{wire} 也可设置为一个定值.

1.3 CMOL 电路容错约束

一个基于或非门实现的逻辑电路可抽象为一个有向无环图 $G=(V, E)$, $g \in V$ 对应于逻辑电路的节点, $e(i, j) \in E$ 表示 g_i 与 g_j 间存在的连接边, 有连接关系的节点称为关联节点. 给定一个 CMOL 电路 θ , 容错映射即找到一种映射关系 $P: G \rightarrow \theta$, 使节点映射后可避开缺陷器件的影响实现正确的逻辑操作. 器件缺陷种类及相应的容错约束讨论如下.

(1) 常开缺陷. 包括纳米二极管永久断开或纳米线非周期性断裂, 使相关的连接路径永远保持在关断状态. 容错时不允许将一对关联节点映射到存在常开缺陷的 2 个单元上, 约束设置为

$$\forall e(i, j) \in E: N(c_i, c_j) \neq 1 \quad (2)$$

其中, c_i / c_j 代表 g_i / g_j 的映射单元; $N(c_i, c_j)$ 等于 1 或 0 表示单元之间是否存在常开缺陷.

(2) 常连缺陷. 指纳米二极管永久导通, 使相关的连接路径永远保持在导通状态. 容错时不允许将一对非关联节点映射到存在常连缺陷的 2 个单元上, 约束设置为

$$\forall e(i, j) \notin E: M(c_i, c_j) \neq 1 \quad (3)$$

其中, $M(c_i, c_j)$ 等于 1 或 0 表示单元间是否存在常连缺陷.

若多个单元两两之间存在常连缺陷, 会使信号错误传播到更广泛区域, 该现象称为常连缺陷传播^[19]. 为阻止这种传播, 容错约束设置为

$$\forall M(c_i, c_j) = 1: c_j = 1 \rightarrow c_i \neq 0 \quad (4)$$

式(4)说明, 若单元 c_i / c_j 之间存在常连缺陷且单元 c_j 已被映射, 其输入连通域内的单元 c_i 不可处于未映射状态, 以避免常连缺陷传播.

除上述容错约束外, 由于单元连通域的存在, 映射过程需要保证映射了关联节点的单元间的距离不大于连通域半径, 约束表示为

$$\forall g_i, g_j \in V, \exists e(i, j) \in E: d(c_i, c_j) \leq r \quad (5)$$

其中, $d(c_i, c_j)$ 表示映射单元间的曼哈顿距离.

2 CMOL 电路容错映射

为了进行功耗优化的容错映射, 首先建立缺陷对的功耗模型; 接着提出单元限用方法; 最后采用修改后的遗传算法完成容错映射。

2.1 缺陷对的功耗模型

CMOL 电路总功耗可计算为

$$P = N_D (P_o + P_d) + P_{\text{leak}} \quad (6)$$

根据式(1)的分析, 在一个确定的映射任务中 P_{leak} 的值不变, 因此可用 N_D 的值比较同一映射任务下不同映射解的功耗. 在无缺陷环境中, N_D 的值应当与逻辑电路连接边总数相等。

2.1.1 常连缺陷

为讨论方便, 引入如下定义。

定义 1. 连接路径存在常连缺陷的 2 个单元, 称为一个常连缺陷对。

定义 2. 输入、输出连通域内均存在常连缺陷的单元称为传播单元。

图 4 所示为一个发生了常连缺陷的电路, 其中纳米线上的黑色圆点表示存在常连缺陷的纳米二极管, 单元 A/B, B/C 各组成一个常连缺陷对, 单元 B 为传播单元, 同时属于 2 个常连缺陷对。

图 4 常连缺陷电路

若节点映射到常连缺陷对上, 以单元 B/C 为例, 有以下 4 种映射模式。

模式 1. 单元 B/C 映射了一对关联节点。

模式 2. 单元 B/C 映射了一对非关联节点。

模式 3. 单元 B 映射了节点而单元 C 未映射。

模式 4. 单元 C 映射了节点而单元 B 未映射。

对于模式 1, 根据纳米二极管永久导通的缺陷特性, 利用常连缺陷对, 该模式既不会影响逻辑功能, 也不会额外增加 N_D , 称为合理映射; 模式 2 违反了约束式(3), 导致逻辑功能发生错误, 若容错映射解中仍存在该模式, 视为容错失败; 对于模式 3, 虽然单元 B 的输出会不受控制地传递到单元 C, 但此时单元 C 与其他单元之间无导通路径,

无法将信号错误传输到其他单元, 只是单元 B/C 间存在的连接路径使 N_D 的值变大, 增加功耗, 称该模式为耗能映射; 模式 4 违反了约束式(4), 若此时有节点映射至单元 A, 单元 A 的信号会通过单元 B 传播到单元 C, 使其逻辑功能出错, 称该模式为传播映射. 对于传播映射, Chen 等^[19]提出传播阻断方法对其进行修正, 示例如图 5 所示. 图 5 中纳米线上的灰色圆点表示无缺陷纳米二极管, 单元 B/C 映射了节点 $g_A / \overline{g_A}$, 是一对互补信号, 单元 D/F 映射了节点 g_D / g_F 且 $g_F = \overline{X}$. 由于单元 D/E, E/F 间存在常连缺陷, 单元 D 的信号将传播到单元 F, 使 $g_F = X + \overline{g_D}$, 此时可将互补信号传输到单元 E, 令其输出变为 0, 单元 F 输出恢复为 $g_F = \overline{X}$. 由示例可知, 每例常连缺陷传播阻断都会增加 2 条额外导通路径, 使 N_D 变大。

图 5 常连缺陷传播阻断示例

由上述分析可知, 耗能映射与修正后的传播映射不会影响电路逻辑功能, 但是会增加电路的功耗开销, 因此将这 2 种映射模式统称为高成本映射模式。

2.1.2 常开缺陷

常开缺陷是指连接单元之间的纳米二极管呈断开状态的缺陷. 因此不管相关单元映射与否, 都不会额外增加 N_D 的值. 在容错过程中只需使映射单元严格遵守约束式(2)以保证逻辑功能正确即可。

2.2 本文方法

容错后, 被利用的常连缺陷对只能以合理映射模式或高成本映射模式存在, 而后者会导致电路功耗增加. 为了进行功耗优化, 若令常连缺陷对都不被利用, 会导致映射资源的减少而增加容错复杂度, 降低电路映射成功率^[15]; 若令常连缺陷对都以合理映射模式被利用, 严格的约束同样会增加容错复杂度, 限制算法的求解速度与求解大电路的能力^[18]. 为此, 本文提出了单元限用方法, 通过限制特定单元的利用降低电路的容错复杂度,

以少量映射资源减少为代价实现高效的容错映射。

2.2.1 高功耗单元限用

定义 3. 在常连缺陷对中, 称常连缺陷存在于输出连通域的单元为常连输入, 另一单元则为常连输出。

通过分析可知常连缺陷传播现象的发生需同时满足以下 2 个条件。

条件 1. 传播单元未被映射。

条件 2. 传播单元的常连输入、常连输出均被映射。

在图 5 所示电路中, 单元 E 为传播单元, 单元 D/F 分别为其常连输入与常连输出, 其中 D/E 和 E/F 分别组成常连缺陷对 1 和 2。由条件 2 可知, 实际上只有单元 D 被映射后, 传播映射的常连缺陷对 2 才需要进行传播阻断, 进而增加电路功耗。另外, 若单元 D 被映射, 而单元 E/F 未被映射, 由于此时单元 D/E 输出一对互补信号, 单元 D/E 、 E/F 间必定有一条导通路径会有电流 I_{on} 流过, 在信号概率设置为 $1/2$ 的前提下, 等效为 N_D 增加 2, 功耗开销与传播阻断方法相同。由单元 D 的例子可知, 传播单元的常连输入对电路功耗影响很大, 将此类单元称为高功耗单元, 若在映射过程对其进行限用, 不仅可避免常连缺陷传播现象的发生, 还可节省由传播阻断操作带来的功耗开销。高功耗单元限用可借助图 6 加以说明, 其中单元 A/B 、 B/C 、 C/D 间均存在常连缺陷, 单元 B/C 为传播单元。创建包含所有单元的集合 $R = \{A, B, C, D, E, F, G, H, J\}$, 将单元 B/C 的常连输入 A/B 存入高功耗单元集合 Q , 只要规定映射单元只可在集合 $U = R - Q = \{C, D, E, F, G, H, J\}$ 中选取, 即可实现限用, 形成算法 1。

算法 1. 高功耗单元限用。

Step1. 设置集合 R , 包含阵列所有单元。

Step2. for R 中的所有单元

if 为传播单元 then

for 该单元所有常连输入

图 6 高功耗单元限用

存入集合 Q 。

Step3. 创建映射单元集合 $U = R - Q$ 。

2.2.2 功耗约束及其松弛方法

为了在映射时减少高成本映射以优化功耗, 提出功耗约束

$$\forall M(c_i, c_j) = 1: c_i = 1 \rightarrow c_j \neq 0 \quad (7)$$

式(7)说明, 在一常连缺陷对中, 不可出现常连输入被映射而常连输出未被映射的情况。

由于高成本映射并不会影响电路逻辑功能, 为了取得求解速度与功耗间的平衡, 提出功耗约束的松弛方法。在 CMOL 电路中每个单元因为连通域约束只能与周围有限个单元进行连接, 映射时容易出现高扇出问题^[26], 节点拥有的扇出越多, 映射越困难。因此, 将节点的扇出数作为约束的松弛指标, 提出以下松弛表达式

$$\lambda < O_i / O_L \quad (8)$$

其中, λ 为 $[0, 1]$ 的随机浮点数; O_i 为节点 i 的扇出数; O_L 为逻辑电路中节点拥有的最大扇出数。若式(8)的判断结果为真, 认为节点 i 的映射单元不必遵守功耗约束。若需要容错的节点的扇出数越多, 则该节点的映射难度越大, 进行功耗约束松弛的概率也越大。

2.3 算法实现

容错映射问题是一个多目标优化问题, 而遗传算法具有搜索空间大、多目标优化的能力。遗传算法的原理是模仿自然界生物选择与遗传的机制, 具有良好的全局搜索能力, 其主要思想是随机生成具有特定基因编码的初始生物种群, 选择个体进行基因交叉、变异产生子代, 通过适值函数的评估保留适应度高的个体, 经过多次迭代可使个体获得期望的性状。选择遗传算法实现本文方法。

2.3.1 基因编码

待映射的逻辑电路中, 每个节点通过整数编号唯一确定且只与 CMOL 电路上的 1 个单元对应。本文生物个体的基因编码结构为 $(C_0, C_1, C_2, \dots, C_{N-1})$, 在映射开始前已对 CMOL 电路进行高功耗单元限用操作, 因此基因的标号 $0, 1, \dots, N-1$ 与映射单元集合 U 中的 N 个单元一一对应。若逻辑电路的节点总数为 m , C_i 的值为 $[0, m-1]$ 中任一不重复的整数或 -1 , 这里 -1 表示相应单元处于未映射状态。图 7 所示为将节点数为 8 的逻辑电路映射至 4×4 大小的 CMOL 电路的一个解, 此时基因编码为 $(3, -1, 2, -1, -1, 4, 0, 6, 7, 1, -1, -1, -1, -1, -1, 5)$ 。

图7 基因编码与CMOL电路关系图

2.3.2 适值函数

种群中适应度高的个体往往有较大机会存活, 算法中判断个体适应度的适值函数定义为

$$f = 1 / \left[\gamma + \sum_{i=1}^m (\eta V_i + p_i) \right] \quad (9)$$

将节点的映射单元对各约束式的违反称为违例. 在式(9)中, V_i 表示映射单元 i 对约束式(2)(3)(5)的违例数; m 为映射单元总数; p_i 为映射单元 i 对约束式(7)的违例数. 由于在映射解中约束式(2)(3)(5)的违例必须消除以保证电路逻辑功能正确, 因此要对这3个约束式的违例进行惩罚, 根据实验取得经验参数, 设置惩罚系数 $\eta=5$. γ 是一个大于或等于1的常数, 避免分母为0时 f 趋向无穷大. 由式(9)可知, 适值越大, 解的质量越好.

2.3.3 选择策略

选择策略模拟了“优胜劣汰”的过程, 可使优良的个体有较大概率被挑选出来作为亲代繁衍新种群. 本文利用轮盘赌算法进行选择. 假设种群的规模为 N_p , 适值为 f_i 的个体 i 被选择的概率为

$$P_i = f_i / \sum_{i=1}^{N_p} f_i.$$

个体 i 的累积概率为

$$q_0 = 0, q_i = \sum_{j=0}^i q_j.$$

每次选择时旋转轮盘 N_p 次, 每次生成 $[0,1]$ 的一个随机数 λ , 若满足 $q_{i-1} < \lambda < q_i$, 则个体 i 被选择次数加1. N_p 次旋转结束, 被选择次数最多的2个个体作为亲代进行繁衍.

2.3.4 交叉与变异

遗传算法的核心为交叉算子, 将亲代的基因替换重组, 有强大的全局搜索能力. 根据 CMOL 电路二维结构的特点, 本文在电路上随机选择一个矩形的交叉区域进行部分匹配交叉, 避免过多地拆散映射好的子电路, 造成算法搜索的浪费. 为了提高缺陷单元的利用率, 使算法在映射资源有限时也能保持较好的性能, 对已合理映射的常连

缺陷对进行保护, 减少交叉操作对相关单元的扰动. 假设选择亲代 F_1, F_2 进行交叉, 保护操作说明如下: 创建 F_1 的受保护单元集合 T_1 , 查找 F_1 中所有已合理映射的常连缺陷对, 对其常连输入与常连输出进行以下判断

$$b_i \leq \sum_{i=1}^m b_i / m \quad (10)$$

其中, $b_i = (V_i + p_i) / e_i$ 表示相关单元的缺陷连接率,

e_i 为映射至该单元的节点的连接边数; $\sum_{i=1}^m b_i / m$ 表示 F_1 中映射单元的平均缺陷连接率. 若常连缺陷对的2个单元的判断结果均为真, 将其存入 T_1 . T_1 生成后, 对 F_2 进行相同操作, 生成受保护单元集合 T_2 . 所有属于集合 T_1 或 T_2 的单元不参与 F_1, F_2 的交叉操作.

具体的交叉操作如图8所示, 随机选择不同行的2个单元作为顶点确定一个矩形交叉区域, 交叉区域以浅灰背景显示, 电路中的常连缺陷对以灰色斜线显示. 在亲代2中, 关关节点4, 6合理映射至常连缺陷对, 经式(10)判断, 该常连缺陷对应受到保护, 因此交叉区域内剩余的3个单元按照 $1 \leftrightarrow 1, 0 \leftrightarrow -1 \leftrightarrow 3 (0 \leftrightarrow 3)$ 的匹配方式进行节点交换生成2个子代.

图8 交叉操作

变异算子的局部搜索能力能够加速算法向最优解靠拢并防止算法过早收敛. 本文变异操作发生在交叉操作之后, 随机挑选一个节点将其移动到未映射单元或与其他单元上的节点交换位置.

本文遗传算法每次迭代将进行 $N_p R_M$ 次交叉

操作, $N_p R_C N/2$ 次变异操作, 其中, N_p 为种群规模; R_M 和 R_C 分别为交叉概率和变异概率; N 为 CMOL 电路单元总数. 每次迭代结束后, 将种群中所有亲代和子代按适值降序排列, 取前 N_p 个个体作为新种群进入下一次迭代.

3 数据分析

本文方法在 Windows 系统, 配置为 Intel Core 3.40GHz 处理器、8GB 内存的 PC 环境下, 用 C 语言编程实现, 并选择在 CMOL 映射技术研究^[10-16,18-19]中被广泛采用的 ISCAS 标准测试电路^[27]验证其有效性. 在连通域半径 $r=18$, 常开缺陷率 10.0%, 常连缺陷率 0.1%, 且缺陷均匀分布的实验条件下, 实现了有多缺陷容错能力的 SAT 方法^[18]与 EAB 方法^[19], 并与本文方法进行比较. 在此需要说明的

是, 当 $r=18$ 时, 每个单元连通域内便存在 611 个纳米二极管, 0.1% 的常连缺陷率已足够对相当多的单元造成影响. 考虑遗传算法的随机搜索性质, 需要排除极端情况, 因此每个标准电路的测试都在 20 个不同的随机数种子下进行, 取结果的平均值.

实验结果如表 2 所示, 其中“节点数”表示逻辑电路节点总数; 用行×列的方式体现 CMOL 电路的面积; “时间”表示 CPU 运行时间; “改进百分比”表示本文方法相对比较方法改进量的平均值; “功耗”表示电路总功耗, 计算式如式(6)所示. 本文采用 Strukov^[20]给出的电路参数, 动态功耗中的逻辑操作频率根据文献[23]计算, 定为 960 MHz. 根据相关经验与多次实验, 本文方法参数取为 $N_p = 24$, $R_M = 0.30$, $R_C = 0.01$, 迭代次数为 1 600. 与 EAB 方法相同, 本文方法的初始种群由本课题组的 LRMA 算法^[13]生成.

表 2 不同方法实验结果对比

电路	节点数	SAT ^[18]			EAB ^[19]			本文		
		面积	时间/s	功耗/ μ W	面积	时间/s	功耗/ μ W	面积	时间/s	功耗/ μ W
s208	136	13×12	72.77	39.20	15×15	4.08	46.53	14×14	2.40	39.96
s298	122	13×12	56.48	46.14	15×15	4.91	53.97	14×14	2.12	46.90
s344	179	14×14	218.24	53.74	18×18	4.75	66.83	16×16	5.24	54.88
s349	184	14×14	225.35	55.36	18×18	6.06	65.32	16×16	6.57	55.36
s382	175	14×14	130.16	60.31	17×17	3.95	71.24	16×16	2.61	62.21
s386	164	14×13	214.42	67.69	17×17	2.59	77.84	16×16	4.12	68.83
s400	188	15×14	296.36	64.94	18×18	5.58	78.58	17×17	4.27	66.84
s420	299	18×18	1935.64	88.25	23×23	43.15	109.70	22×22	52.68	89.77
s444	187	15×14	321.54	67.01	18×18	6.73	78.54	17×17	5.87	70.24
s510	304				27×27	153.03	126.05	24×24	67.01	104.88
s526	273				23×23	34.10	126.36	21×21	35.84	104.83
s641	301				25×25	119.63	106.74	22×22	86.22	85.82
s713	321				26×26	134.74	123.67	23×23	125.85	93.25
s820	447				33×33	475.42	234.60	28×28	350.58	186.21
s832	454				34×34	526.32	238.49	28×28	328.54	189.20
s838	606				40×40	993.63	259.42	35×35	871.98	189.96
s1196	674				46×46	4238.75	330.10	42×42	3507.53	255.35
s1238	721				48×48	5399.15	351.12	43×43	4179.68	280.61
改进百分比/%		-34.27	97.52	-2.22	17.92	14.99	17.62			

SAT 方法要求若有节点映射至常连输出, 则必须将其扇入映射到对应的常连输入, 并在所有约束都满足后才会返回可行解. 因此, 可以得到比本文方法更好的面积与功耗, 然而由于将容错约束用布尔变量编码为合取范式时所需的变量数过多, 生成的中间处理文件很大, 且约束设置严格, SAT 方法的求解速度与求解大电路的能力受到限制.

与 EAB 方法相比, 本文方法的功耗和面积分别平均减少了 17.62% 和 17.92%. 这是因为 EAB 方

法中, 没有设置功耗约束与合理映射的常连缺陷对的保护机制, 在约束式(3)(4)的共同作用下, 常连缺陷对大多以耗能映射模式被利用, 该现象在增加功耗的同时降低了缺陷单元的利用率, 增加了电路的面积开销. 本文方法与 EAB 方法相比有 14.99% 的求解速度优化, 这主要是因为本文方法不必进行约束式(4)违例的传播阻断. 在传播阻断过程中, 如果所有违例单元在其连通域内都能找到互补信号, 则判定为阻断成功; 否则需要继续迭代求解^[28], 该操作无疑会使算法求解速度降低.

4 结 语

本文以功耗为优化目标对CMOL电路容错映射问题展开研究,通过高功耗单元的限用与功耗约束的设置对功耗进行优化,并采用遗传算法实现。实验结果表明,该方法不仅能够有效地减少电路功耗与面积,而且对求解速度也有较好的优化。本文方法通过限制部分缺陷单元的利用以达到优化目的,若能够对这些单元进行有效利用,可以进一步减少电路面积并提升容错效率,这是下一步将开展的工作。

参考文献(References):

- [1] Likharev K K, Strukov D B. CMOL: devices, circuits, and architectures[M] //Lecture Notes in Physics. Heidelberg: Springer, 2005, 680: 447-477
- [2] Likharev K K. CMOL: second life for silicon?[J]. Microelectronics Journal, 2008, 39(2): 177-183
- [3] Türel Ö, Lee J H, Ma X L, *et al.* Architectures for nanoelectronic implementation of artificial neural networks: new results[J]. Neurocomputing, 2005, 64: 271-283
- [4] Abid Z, Liu M, Wang W. 3D integration of CMOL structures for FPGA applications[J]. IEEE Transactions on Computers, 2011, 60(4): 463-471
- [5] Payvand M, Theogarajan L. Exploiting local connectivity of CMOL architecture for highly parallel orientation selective neuromorphic chips[C] //Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures. Los Alamitos: IEEE Computer Society Press, 2015: 187-192
- [6] Mane P, Talati N, Riswadkar A, *et al.* Implementation of NOR logic based on material implication on CMOL FPGA architecture[C] //Proceedings of the 28th International Conference on VLSI Design. Los Alamitos: IEEE Computer Society Press, 2015: 523-528
- [7] Lu W, Lieber C M. Nanoelectronics from the bottom up[J]. Nature Materials, 2007, 6(11): 841-850
- [8] Tunali O, Altun M. A fast logic mapping algorithm for multiple-type-defect tolerance in reconfigurable nano-crossbar arrays[J]. IEEE Transactions on Emerging Topics in Computing, 2019, 7(4): 518-529
- [9] Yan H, Choe H S, Nam S W, *et al.* Programmable nanowire circuits for nanoprocessors[J]. Nature, 2011, 470(7333): 240-244
- [10] Sait S M, Oughali F C, Arafah A M. Engineering a memetic algorithm from discrete cuckoo search and Tabu search for cell assignment of hybrid nanoscale CMOL circuits[J]. Journal of Circuits, Systems and Computers, 2016, 25(4): Article No.16500237
- [11] Sait S M, Arafah A M. Cell assignment in hybrid CMOS/nanodevices architecture using Tabu Search[J]. Applied Intelligence, 2014, 40(1): 1-12
- [12] Sait S M, Sheikh A T, El-Maleh A H. Cell assignment in hybrid CMOS/nanodevices architecture using a PSO/SA hybrid algorithm[J]. Journal of Applied Research and Technology, 2013, 11(5): 653-664
- [13] Xia Y S, Chu Z F, 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
- [14] Sait S M, Arafah 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): 2515-2529
- [15] Chen Dingheng, Xia Yinshui, Chu Zhufei. Defect-unaware defect-tolerant CMOL cell mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2017, 29(11): 2133-2139(in Chinese)
(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)
- [16] Zha Xiaojing, Xia Yinshui, Chu Zhufei. Dynamic stratifying and its applications in defect-tolerant mapping optimization for CMOL circuits[J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(8): 1457-1466(in Chinese)
(查晓婧, 夏银水, 储著飞. 动态分层及其在CMOL电路缺陷容忍映射优化中的应用[J]. 计算机辅助设计与图形学学报, 2019, 31(8): 1457-1466)
- [17] Strukov D B, Likharev K K. CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices[J]. Nanotechnology, 2005, 16(6): 888-900
- [18] Hung W N N, Gao C J, Song X Y, *et al.* Defect-tolerant CMOL cell assignment via satisfiability[J]. IEEE Sensors Journal, 2008, 8(6): 823-830
- [19] Chen D H, Xia Y S, Wang Z F. Stuck-at-close defect propagation and its blocking technique in CMOL cell mapping[J]. Microelectronics Journal, 2018, 72: 100-108
- [20] Strukov D B. Digital architectures for hybrid CMOS/ nanodevice circuits[D]. New York: Stony Brook University, 2006
- [21] Fan Xiaohui. Design of low-leakage power CMOS standard cells in nano-meter technology[D]. Ningbo: Ningbo University, 2015(in Chinese)
(范晓慧. 纳米工艺下低漏功耗CMOS标准单元的设计[D]. 宁波: 宁波大学, 2015)
- [22] Likharev K K. Electronics below 10 nm[M] //Nano and Giga Challenges in Microelectronics. Amsterdam: Elsevier Press, 2003: 27-68
- [23] Likharev K K, Strukov D B. Prospects for the development of digital CMOL circuits[C] //Proceedings of the IEEE International Symposium on Nanoscale Architectures. Los Alamitos: IEEE Computer Society Press, 2007: 109-116
- [24] Gao C J, Hammerstrom D. Cortical models onto CMOL and CMOS — architectures and performance/price[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2007, 54(11): 2502-2515
- [25] Hu Meiqun. Performance study of CMOL circuits[D]. Ningbo: Ningbo University, 2011(in Chinese)
(胡美群. CMOL电路性能研究[D]. 宁波: 宁波大学, 2011)
- [26] Chu Z F, Xia Y S, Hung W N N, *et al.* Timing-driven logic restructuring for nano-hybrid circuits[J]. International Journal of Electronics, 2013, 100(5): 669-685
- [27] Brglez F, Bryan D, Koźmiński K. Combinational profiles of sequential benchmark circuits[C] //Proceedings of the IEEE International Symposium on Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 1989: 1929-1934
- [28] Ningbo University. A method to block the propagation of stuck-at-close defects in nano CMOS circuits: China, CN201610589884.8[P]. 2016-11-16(in Chinese)
(宁波大学. 一种抑制纳米CMOS电路常连缺陷传播的方法: 中国, CN201610589884.8[P]. 2016-11-16)