

基于 SMPSO 算法的三值 FPRM 电路延时优化

汪 涛, 张会红*

(宁波大学 电路与系统研究所, 浙江 宁波 315211)

摘要: n 输入变量的三值 FPRM 电路存在 3^n 个不同极性及其对应的函数展开式. 通过对三值列表转换技术以及三值 FPRM 逻辑表达式的研究, 并结合自适应变异粒子群算法提出一种三值 FPRM 电路延时优化解决方案. 首先根据三值 FPRM 逻辑表达式的特征建立延时估计模型; 然后结合三值列表转换技术和 SMPSO 算法, 对三值 FPRM 电路延时进行优化; 最后采用 PLA 格式的 MCNC Benchmark 电路进行算法测试. 结果表明: SMPSO 算法相比于穷尽算法的优化效果更佳, 电路的延时和面积平均节省 11.6% 和 18.6%.

关键词: 延时优化; 三值 FPRM 电路; SMPSO 算法; 极性搜索

中图分类号: TN79

文献标志码: A

文章编号: 1001-5132 (2017) 02-0060-06

随着微电子设计技术与工艺的发展, 集成电路的物理尺寸不断缩小, 导致芯片集成度极大提高. 单位面积芯片电路复杂度的增长在影响电路速度的同时, 也给系统整体的稳定性带来一些不确定因素. 基于电路结构与功能简单两方面因素考虑, 传统数字集成电路多采用二值逻辑来表示. 与二值逻辑相比, 多值逻辑电路的单线携带信息量更多, 能有效提高数据信息的处理速度, 同时减小芯片面积的开销^[1]. 多值逻辑电路中以三值逻辑最为典型, 其电路设计与实现相对简单.

固定极性 Reed-Muller 逻辑电路是一种常用的三值 RM 逻辑电路. n 输入变量的三值 FPRM (Fixed-Polarity Reed-Muller) 逻辑函数有 3^n 个不同的极性, 与之相对应有 3^n 个繁简不一的逻辑表达式^[2]. 因此, 搜索电路的最佳极性可以优化电路的面积与延时. 对于中小规模电路而言, 按照非循环格雷码顺序遍历所有的极性, 能够找到最简的函数 FPRM 表达式^[3]. 然而随着电路输入变量逐渐增加, 极性搜索空间将呈指数倍增长, 穷尽搜索策略不能保证在确定的时间内完成极性搜索. 而群体智能优化算法能有效处理搜索空间大的问题, 是

一种可行的解决方案.

自适应变异粒子群算法^[4](Self-adaptive Mutation on Particle Swarm Optimization, SMPSO)是一种改进的粒子群算法. 基本粒子群算法容易产生早熟收敛、局部寻优能力较差等问题^[5]. SMPSO 算法对最优粒子自适应调整其搜索邻域大小, 增强粒子群的局部搜索能力; 对非最优粒子给予小概率的随机变异, 增强粒子群的多样性和全局搜索能力^[4]. 鉴此, 本文结合自适应变异粒子群算法和三值 FPRM 逻辑表达式, 提出一种基于 SMPSO 算法的三值 FPRM 电路延时优化方案.

1 三值 FPRM 表达式与延时估计模型

1.1 三值 FPRM 表达式

极性为 p 的 n 变量三值 FPRM 电路可表示如下^[6]:

$$f_p(x_1, x_2, \dots, x_n) = \oplus \sum_{i=0}^{3^n-1} a_i \cdot \prod_{j=0}^{n-1} x_j^{i_j}, \quad (1)$$

其中, 下标 p 表示电路的极性, 用三进制数表示为 $p_{n-1}p_{n-2} \cdots p_0$; $\oplus$ 运算符表示模 3 加运算; a_i 为三值 FPRM 系数, $a_i \in \{0, 1, 2\}$, $i = 0, 1, 2, \dots, 3^n - 1$;

收稿日期: 2016-05-29.

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

基金项目: 国家自然科学基金(61306041).

第一作者: 汪涛(1991—), 男, 安徽六安人, 在读硕士研究生, 主要研究方向: 多值逻辑电路优化设计. E-mail: 980201115@qq.com

*通信作者: 张会红(1976—), 女, 河北定州人, 硕士/副教授, 主要研究方向: 数字集成电路设计优化. E-mail: zhanghuihong@nbu.edu.cn

$\prod_{j=0}^{n-1} \dot{x}_j^{i_j}$ 表示模 3 乘项, 即 $\prod_{j=0}^{n-1} \dot{x}_j^{i_j} = \dot{x}_{n-1}^{i_{n-1}} \cdot \dot{x}_{n-2}^{i_{n-2}} \cdots \dot{x}_0^{i_0}$, $i_j \in \{0, 1, 2\}$, 其中, $\dot{x}_j^0 = 1, \dot{x}_j^1 = \dot{x}_j, \dot{x}_j^2 = \dot{x}_j \cdot \dot{x}_j, j = 0, 1, 2, \dots, n-1$.

由上述三值 FPRM 逻辑函数展开式可知, 变量的表现形式是由极性 p 和下标 i 所决定. 三值 FPRM 电路某变量的极性和上标分别用 p_k 和 i_k 来表示, 通过表 1 所示对应关系可以确定 $\dot{x}_k^{i_k}$ 的具体形式, 其中 $x_k^{i_k}$ 表示 $\dot{x}_k^{i_k}$ 的原变量, $(x_k \oplus m)^{i_k}$ 表示 $\dot{x}_k^{i_k}$ 的互补变量, $m \in \{1, 2\}$. 分析表 1 可知, 当 $p_k = 0$ 时, $\dot{x}_k^{i_k}$ 以原变量的形式出现; 当 $p_k = 1$ 时, $\dot{x}_k^{i_k}$ 以其互补变量 $(x_k \oplus 1)^{i_k}$ 的形式出现; 当 $p_k = 2$ 时, $\dot{x}_k^{i_k}$ 以其互补变量 $(x_k \oplus 2)^{i_k}$ 的形式出现.

表 1 $\dot{x}_k^{i_k}$ 取值表

p_k	i_k	$\dot{x}_k^{i_k}$
0	0	1
0	1	x_k
0	2	x_k^2
1	0	1
1	1	$x_k \oplus 1$
1	2	$(x_k \oplus 1)^2$
2	0	1
2	1	$x_k \oplus 2$
2	2	$(x_k \oplus 2)^2$

1.2 三值 FPRM 极性转换

国内外学者已经提出了多种不同的多值极性转换方法, 比如矩阵转换法^[6](Matrix Transformation Method)、图形转换法^[7](Map Transformation Method)和列表转换算法^[8]等. 与前 2 种方法相比, 列表转换算法具有速度快, 便于实验测试等优点. 因此, 本文将列表转换技术运用于三值 FPRM 表达式, 提出的 2 种转换方法分别如下: 从 Boolean 逻辑到 RM 逻辑最小项表达式的极性转换算法; RM 逻辑最小项表达式的极性间转换算法.

从 Boolean 逻辑最小项到某一极性下的三值 FPRM 表达式的转换算法描述如下^[7]:

(1) 将函数 $f(m)$ 的最小项 m 表示为三进制序列 $m = (m_0, m_1, \dots, m_{n-1})$;

(2) 产生 GF(3)域上 p_i 极性对应的转换矩阵 $G_3^{<p_i>}$, $p_i \in \{p_0, p_1, \dots, p_{n-1}\}$, 变量 x 以及其互补变量所对应的基本转换矩阵和逆矩阵见表 2 所示;

表 2 GF(3)互补变量和转换矩阵

变量	转换矩阵	逆矩阵
$x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$	$G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix}$	$(G_3)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}$
$x \oplus 1 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$	$G_3^{<1>} = \begin{bmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ 2 & 2 & 2 \end{bmatrix}$	$(G_3^{<1>})^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix}$
$x \oplus 2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$	$G_3^{<2>} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix}$	$(G_3^{<2>})^{-1} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

(3) 根据(2)式和(3)式, 产生每个最小项对应的新项 θ_i 及其对应 RM 系数的贡献值 $\gamma(\theta_i)$;

$$\theta_i \in \{j \mid G_3^{<p_i>}[j, m_i] \neq 0\}, \quad (2)$$

$$\gamma(\theta) = (\gamma_1 \cdot \gamma_2 \cdots \gamma_n) \cdot \gamma(m), \quad (3)$$

其中, θ_i 为最小项 m_i 对应的新项; $G_3^{<p_i>}[j, m_i]$ 为极性 p_i 对应转换矩阵 $G_3^{<p_i>}$ 的第 j 行, 第 m_i 列数, $\gamma(\theta)$ 为最小项 m 对应的 RM 系数贡献值, $\gamma_i = G_3^{<p_i>}[\theta_i, m_i], i = 1, 2, \dots, n, \theta = (\theta_1, \theta_2, \dots, \theta_n)$.

(4) 在索引表中叠加所有新项对应的贡献值 $\gamma(\theta)$;

(5) 对所有其余的最小项, 重复步骤(2)~(4), 最后得到的累加值就是 RM 表达式的系数.

根据(4)式所示不同极性间贡献值的计算方法, 可以得到极性 p 到极性 p' 的三值 FPRM 表达式极性间转换算法^[8].

$$C_u^{<p'>} = [\prod_{z=0}^{n-1} \{(G_3^{<p'_i>} \cdot G_3^{<p_i>^{-1}})[u_z, v_z]\}] \cdot C_v^{<p>}, \quad (4)$$

其中, $u, v = 0, 1, 2, \dots, 3^n - 1$, $G^{<p>}[u, v]$ 为矩阵 $(G_3^{<p'_i>} \cdot G_3^{<p_i>^{-1}})[u_z, v_z]$ 第 u_z 行, 第 v_z 列元素的累乘值, $C_v^{<p>}$ 是极性 p 下的 FPRM 系数矢量中的非零值.

具体算法描述如下:

(1) 将极性 p 和极性 p' 用三进制数分别表示为 $p = (p_0, p_1, \dots, p_{n-1})$ 和 $p' = (p'_0, p'_1, \dots, p'_{n-1})$;

(2) 产生 GF(3)域上 p_i 极性对应的转换矩阵 $G_3^{<p'_i>} \cdot G_3^{<p_i>^{-1}}$, $p_i \in (p_0, p_1, \dots, p_{n-1})$;

(3) 根据(5)式和(6)式, 产生 FPRM 系数矢量中每个非零值 C_n 对应的新项 β_i 及其对极性 p' 对应的 FPRM 系数的贡献值 $\gamma(\beta_i)$;

$$\beta_i \in \{j \mid (G_3^{<p'_i>} \cdot G_3^{<p_i>^{-1}})[j, C_i] \neq 0\}, \quad (5)$$

$$\gamma(\beta) = (\gamma_1 \cdot \gamma_2 \cdots \gamma_n) \cdot \gamma(n), \quad (6)$$

其中, C_i 为极性 p_i 对应的 FPRM 系数矢量中的非零值; β_i 为非零值 C_i 对应的新项, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$; $\gamma(\theta)$ 为极性 p' 对应的 FPRM 系数贡献值, $\gamma_i = (G_3^{<p_i>} \cdot G_3^{<p_i>-1})[\beta_i, n_i], i=1, 2, \dots, n$.

(4) 在索引表中叠加所有新项对应的贡献值 $\gamma(\beta)$;

(5) 对所有其余的非零 C_n , 重复步骤(2)~(4), 最后得到的累加值就是极性 p' 对应的 FPRM 表达式系数.

1.3 三值 FPRM 电路延时估计模型

传统的数字组合电路使用 Boolean 逻辑表示, 其主要由原始输入端、内部结点和原始输出端三部分构成. 由(1)式可知, 三值 FPRM 逻辑电路表达式是由模 3 加项与模 3 乘项所组成, 而 RM 逻辑又可通过 Boolean 逻辑转换得到, 因此可以将 Boolean 逻辑电路的延时估计模型运用于三值 FPRM 电路中. 假设一个三值 FPRM 电路展开式为: $y = x_0 \oplus (x_1 \oplus 1)^2 \oplus (x_2 \oplus 1) \cdot (x_3 \oplus 2)^2$, 其中, $x_0, (x_1 \oplus 1)^2, (x_2 \oplus 1)$ 和 $(x_3 \oplus 2)^2$ 为原始输入端, z 表示内部结点, y 为原始输出端, 其电路网络如图 1 所示.

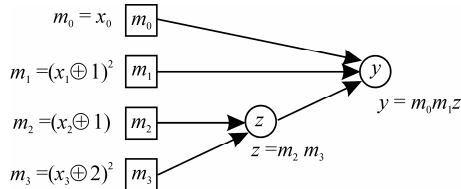


图 1 $y = x_0 \oplus (x_1 \oplus 1)^2 \oplus (x_2 \oplus 1) \cdot (x_3 \oplus 2)^2$ 电路网络

三值 FPRM 电路网络的延时主要是由门延时和数据预备时间^[9](Data-ready Time, DT)两部分组成. 对于门延时的估计通常采用单位延时模型^[10], 即将每个门延时设为一个单位时间. 若电路网络原始输入端的数据预备时间不明确, 通常将其设置为 0^[9].

在电路网络中, 多输入门构成结点的复杂程度有很大的差异, 故多输入门的延时估计是毫无意义的. 在电路映射之前, 一般将多输入 XOR 门和 AND 门分解为一系列二输入 XOR 门和 AND 门. 分解后的二输入门电路网络, 将其关键路径(即延时最长路径)上的门延时之和作为三值 FPRM 电路的延时估计值. 如三值 FPRM 电路分解后的所有节点集合为 V , 则任一节点 i 的 DT 可以表示为^[10]:

$$t_i = d_i + \max(t_{i_left} + t_{i_right}), \quad (7)$$

式中, t_i 代表节点 i 的 DT, $i \in V$; d_i 代表二输入门结点 i 的输入门延时, 根据单位延时模型将其置为 1; t_{i_left} 与 t_{i_right} 代表二输入门结点的 DT.

2 基于 SMP SO 算法的三值 FPRM 电路延时最佳极性搜索

2.1 算法概述

基本粒子群算法随着迭代次数的增加, 种群收敛于某些最优个体, 搜索空间的多样性逐渐丢失, 导致粒子容易陷入局部最优解. 因此, 将局部搜索和变异操作同时混合到 PSO 算法中, 通过适当因子的调节, 能够提高算法的开发能力, 防止算法早熟, 并提高 PSO 算法的全局寻优能力. SMP SO 算法对种群最优粒子和非最优粒子分别给予了不同的变异操作, 其根据种群的进化程度, 最优粒子在其邻域搜寻更优的解, 使得种群收敛速度进一步提高. 同时, 对非最优粒子进行小概率的随机变异, 保持种群多样性以增强全局搜索能力.

2.2 适应度函数

三值 FPRM 电路的延时优化映射到粒子群算法就是搜索种群的最优粒子位置的问题. 在算法每次迭代搜索过程中, 需对该极性下的三值 FPRM 电路的延时进行评估. 假设某极性下粒子 i 的位置用 x_i 来表示, 权衡电路延时和面积两方面因素建立如下的适应度函数:

$$fitness(x_i) = \alpha R_{Delay}(x_i) / R_{Delay, total} + (1 - \alpha) R_{Area}(x_i) / R_{Area, total}, \quad (8)$$

式中, $R_{Delay}(x_i)$ 和 $R_{Area}(x_i)$ 分别为粒子 i 对应极性电路的延时和面积; $R_{Delay, total}$ 和 $R_{Area, total}$ 分别为种群的粒子延时总和以及面积总和; α 为延时和面积的权重值, $\alpha \in [0, 1]$.

2.3 全局搜索

粒子群算法对种群随机进行初始化, 设 x_i 表示粒子 i 的位置, v_i 表示粒子 i 的飞行速度. 种群的历史最优点为 P_i , 种群搜寻到的最优位置为 P_g . 粒子 i 的位置和速度可以由(9)式和(10)式得到.

$$v_i^{n+1} = wv_i^n + c_1r_1(P_i - x_i^k) + c_2r_2(P_g - x_i^k), \quad (9)$$

$$x_i^{n+1} = x_i^n + v_i^{n+1}, i=1, 2, \dots, m, \quad (10)$$

$$w = w_{max} - n(w_{max} - w_{min}) / N, n=1, 2, \dots, N, \quad (11)$$

其中, n 表示种群当前进化次数; N 表示种群最大进

化次数; w 为惯性权重, 反映粒子速度变化的影响, 其值更新如(11)式所示; 进化迭代早期通常设置 w 为较大值, 有利于种群空间的大范围探索, 迭代晚期选择较小的 w 值, 以期提高收敛的精度; c_1 和 c_2 为加速因子, 用来控制速度变化的频率; r_1 和 r_2 为平均随机数, 且 $r_1, r_2 \in [0, 1]$; $c_1 r_1 (P_i - x_i^k)$ 部分反映粒子的学习能力, $c_2 r_2 (P_g - x_i^k)$ 部分反映粒子间的信息合作与共享能力; m 为粒子群总数。

2.4 变异操作

变异操作主要是根据最优粒子和非最优粒子在种群进化过程中的作用给予不同的自适应变异策略。在迭代过程中, 若粒子发现种群空间中的一个最优点, 其他粒子会逐渐向其靠拢, 算法容易陷入局部最优。因此, 适当调整最优粒子搜索邻域的大小, 使其继续寻找更优点, 可以增强算法的局部搜索能力。最优粒子的变异操作如(12)式和(13)式所示:

$$v_i^{n+1} = \beta^n (1 - 2r_3), \quad (12)$$

$$x_i^{n+1} = P_g + wv_i^{n+1}, \quad (13)$$

式中, β 为比例因子, 权衡最优粒子搜索邻域的大小; r_3 为平均随机数, 且 $r_3 \in [0, 1]$ 。

种群空间内非最优粒子的速度随着迭代次数的增加最终趋于零, 种群的多样性随之降低。当非最优粒子速度为零时, 对其进行小概率的随机变异, 有利于寻找新的最优点, 增强粒子群的全局搜索能力。非最优粒子的变异操作如(14)式和(15)式所示:

$$\text{if } (v_i^n = 0) \text{ then } v_i^n = \theta r_1 v, \quad (14)$$

$$\text{if } (r_2 - c_1) \text{ then } x_i^n = r_4 x, \quad (15)$$

$$\begin{cases} \text{if } (r_1 > 0.5) \text{ then } \theta = -1, \\ \text{if } (r_1 \leq 0.5) \text{ then } \theta = 1, \end{cases} \quad (16)$$

式中, v 和 x 分别表示粒子初始化速度和位置; θ 表示方向变异控制参数, 取值方法如(16)式所示; r_4 表示平均随机数, 且 $r_4 \in [0, 1]$ 。

2.5 算法流程

结合 SMPSO 算法、三值 FPRM 电路延时模型和列表转换技术, 电路延时和面积优化算法的步骤如下:

(1)首先读取 PLA 格式的电路文件, 根据文献[11]提出的方法由二值基准电路转换得到三值基准电路;

(2)在种群进化前, 初始化各粒子的速度和位置, 将种群中最优粒子的位置设为全局最优;

(3)使用三值列表技术将电路表达式转换到当前粒子群的位置, 并评估适应度值大小;

(4)更新当前粒子位置, 判断是否为全局最优, 若是, 根据(12)式和(13)式更新最优粒子速度和位置; 否则, 根据(14)式和(15)式更新非最优粒子速度和位置;

(5)判断是否达到最大进化代数, 若不是, 继续迭代进化种群; 否则, 输出最佳极性。

该算法的流程图如图2所示。

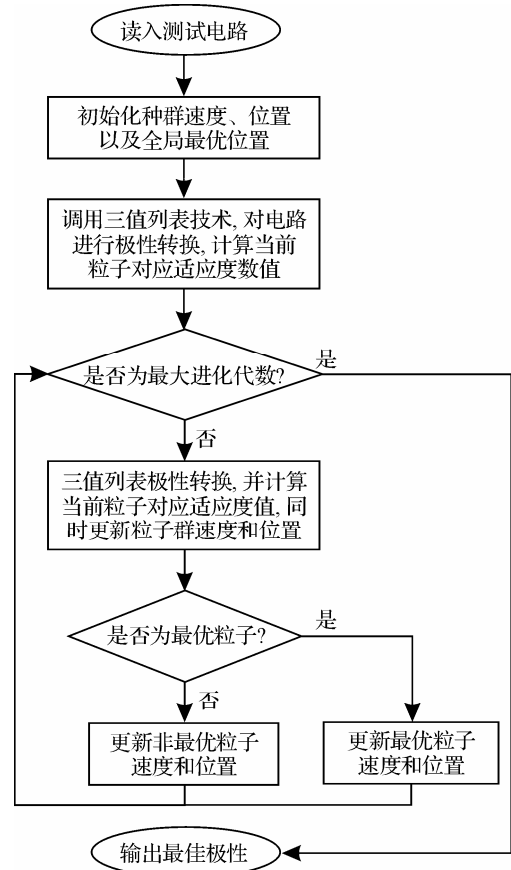


图2 基于 SMPSO 算法的三值 FPRM 电路延时优化

3 实验数据与分析

为验证 SMPSO 算法的优化性能, 随机选取 9 个常用的中小规模 Benchmark 电路测试, 并与基于格雷码(Gray)^[3]顺序的穷尽算法进行对比实验。文中所提出算法用 C 语言实现, 在 Intel Pentium(R) CPU G3250 @ 3.20GHz、4.00 GB RAM 的主机上, 使用 Windows 10(64 位)操作系统下的 Dev-C++软

件编译运行.

SMPSO 算法相关参数设置如下: 权重比例 $\alpha = 0.5$, 惯性权重 w 变化范围为 $0.9 \sim 0.3$, 递减速率为 $(0.9 - 0.3) / 1\,000$, 加速因子 $c_1 = c_2 = 1.59$, 种群粒子总数 m 为 30, 最大迭代数为 4 000, 最优粒子变异比例因子 $\beta = 0.1$.

实验数据见表 3, 其中, 列 1 表示测试电路的名称, 列 2 表示测试电路的输入端变量个数, 列 3 和列 4 为基于穷尽算法的电路面积和延时优化数据, 列 5 和列 6 为基于 SMPSO 算法的电路面积和延时优化数据. 从表中数据对比可以看出, 对于中小规模电路, 运用 SMPSO 算法搜索到的电路延时平均降低 15.8%, 面积平均增加 6.7%.

表 3 中小规模三值 FPRM 电路面积和延时优化数据比较

Circuit		Gray		SMPSO	
Name	Input	Area	Delay	Area	Delay
squar5	5	14	7	13	5
con1	7	30	11	37	8
inc	7	9	8	12	6
tms	8	27	7	32	6
sex	9	22	5	22	5
clpl	11	206	8	259	6
br1	11	125	9	143	8
newapla	12	273	8	209	9
cu	14	109	13	143	11

为了进一步验证所提算法的有效性, 随机抽取 12 个大规模三值 FPRM 电路进行测试, 实验测试结果见表 4. 其中, 列 1、列 2 分别为电路的名称和输入端数, 列 3、列 4 为极性 0 三值 FPRM 电路的面积和延时, 列 5、列 6 为基于 SMPSO 算法搜索到的电路面积和延时. 与极性 0 的三值 FPRM 电路相比, 运用 SMPSO 算法寻找的极性所对应的电路延时平均降低 11.6%, 面积平均减少 18.6%.

延时和面积节省百分比定义如下:

$$R_{\text{Save, delay}} = \frac{R_{\text{Delay,0}} - R_{\text{Delay,s}}}{R_{\text{Delay,0}}} \times 100\%, \quad (17)$$

$$R_{\text{Save, Area}} = \frac{R_{\text{Area,0}} - R_{\text{Area,s}}}{R_{\text{Area,0}}} \times 100\%, \quad (18)$$

其中, $R_{\text{Save, delay}}$ 和 $R_{\text{Save, Area}}$ 分别为延时和面积优化节省百分比; $R_{\text{Delay,0}}$ 和 $R_{\text{Area,0}}$ 分别为极性 0 时电路的延时和面积; $R_{\text{Delay,s}}$ 和 $R_{\text{Area,s}}$ 分别为 SMPSO 算法搜

索的电路延时和面积.

表 4 较大规模电路三值 FPRM 电路面积和延时优化结果

Circuit		Polarity 0		SMPSO	
Name	Input	Area	Delay	Area	Delay
newtpla	15	86	9	74	8
pml	16	85	8	81	7
spla	16	137	10	124	9
table5	17	839	13	602	11
pcle	19	17	6	17	5
mux	21	241	9	237	8
cm150a	21	275	11	216	10
duke2	22	76	7	80	6
s382	25	563	14	464	12
s526	25	86	8	56	7
bc0	26	293	9	251	8
bcd	26	236	8	185	8

4 结论

结合 SMPSO 算法与三值列表极性转换技术, 利用类 Huffman 算法分析任一极性下的电路延时, 提出了一种针对较大规模三值 FPRM 电路延时的优化解决方案. 文章所提算法已用 C 语言实现, 通过对 12 个三值基准测试电路进行实验, 验证了该算法在电路延时优化方面的有效性. 由于 SMPSO 算法主要是通过添加变异操作来增强其局部搜索能力, 变异参数的设定将直接影响算法收敛速度和收敛精度, 因此接下来的研究主要集中在优化参数设置, 以进一步提高算法的优化效果.

参考文献:

- [1] XU K D, MCFETERS G A, STEWART P S. Biofilm resistance to antimicrobial agents[J]. Microbiology, 2000, 146(3):547-549.
- [2] FALKOWSKI B J, CHENG F. Fastest classes of linearly independent transforms over GF(3) and their properties[J]. Computers and Digital Techniques, 2005, 152(5):567-576.
- [3] WANG P J, LI H. Low power mapping for AND/XOR circuits and its application in searching the best mixed-polarity[J]. Journal of Semiconductors, 2011, 32(2):108-113.
- [4] 周利君, 彭卫, 邹芳, 等. 自适应变异粒子群算法[J]. 计算机工程与应用, 2016, 52(7):50-55.

- [5] 靳雁霞, 薛敬, 张建华, 等. 学习历史经验的粒子群优化算法[J]. 小型微型计算机系统, 2016, 37(3):572-575.
- [6] FALKOWSKI B J, FU C. Polynomial expansions over GF(3) based on fastest transformation[C]//IEEE Proceedings 33rd International Symposium on Multiple-Valued Logic, 2003:40-45.
- [7] FU C, FALKOWSKI B J. Ternary fixed polarity linear Kronecker transforms and their comparison with ternary Reed Muller transform[J]. Journal of Circuits, Systems, and Computers, 2005, 14(4):721-733.
- [8] 孙飞, 汪鹏君, 俞海珍. 三值 FPRM 电路极性间转换算法及其在面积优化中的应用[J]. 浙江大学学报(理学版), 2014, 41(1):43-48.
- [9] FUJITA M, MURGAI R. Delay estimation and optimization of logic circuits: A survey[C]//Proceedings of the Asia and South Pacific Design Automation Conference, Piscataway: IEEE Press, 1997:25-30.
- [10] MICHELI G D. Synthesis and Optimization of Digital Circuits[M]. New York: McGraw-Hill Higher Education, 1994:88-102.
- [11] FALKOWSKI B J, LOZANO C C, RAHARDJA S. Column polarity matrix algorithm for ternary fixed polarity Reed-Muller expansions[J]. Journal of Circuits, Systems and Computers, 2006, 15(2):243-262.

Delay optimization for ternary FPRM circuit based on SMPSO algorithm

WANG Tao, ZHANG Hui-hong*

(Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China)

Abstract: For n -variable ternary FPRM (Fixed-Polarity Reed-Muller) circuit, there are 3^n different polarities and function expansions. Based on the study of ternary conversion technology as well as ternary FPRM logic expression in combination with the SMPSO algorithm, a solution of ternary FPRM circuit delay optimization is proposed in this paper. Firstly, according to the characteristics of ternary FPRM logic expression, the delay estimation model is established. Then combined with the ternary conversion technology and the SMPSO algorithm, the delay of ternary FPRM circuit is optimized. Finally, the algorithm is tested by PLA formatted MCNC Benchmarks, and the results indicate that the optimization of SMPSO algorithm performs better than that of exhaustive algorithm. The average delay and the area of the circuits are reduced by 11.6% and 18.6%, respectively.

Key words: delay optimization; ternary FPRM circuit; SMPSO algorithm; polarity search

(责任编辑 章踐立)