

考虑串扰因素的可逆电路的符号综合方法

胡 靖, 马光胜, 李东海, 冯 刚

(哈尔滨工程大学计算机科学与技术学院, 黑龙江哈尔滨 150001)

摘 要: 目前存在的可逆电路综合方法大多只适用于输入输出相对较小的电路, 而且忽略了路径延时的影响, 无法应对集成电路日益复杂的趋势. 为了使综合方法能够适用于大规模可逆电路, 本文采用矩阵模型和符号代数作为理论基础, 提出了一种符号综合方法, 在考虑面积、延时、串扰等约束下利用成本函数来指导综合过程. 实验结果表明利用这种启发式算法与现有的综合方法相比, 在面积上所得结果近似, 而总串扰得到了 10.3% 的改善, 其路径延时要减少 5% 到 20% 之多, 并且从 CPU 时间和存储开销上都显示出该算法的优势, 有能力在有效的时间内实现大规模可逆电路的综合.

关键词: 逻辑综合; 可逆电路; 延时; 串扰

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2008) 05-1029-06

Considering Crosstalk Symbolic Synthesis Method for Reversible Circuits

HU Jing, MA Guang sheng, LI Dong hai, FENG Gang

(College of Computer Science & Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

Abstract: Presently existing synthesis methods for reversible circuits were applicable only to reversible circuits with small numbers of inputs and outputs and had neglected the impact of path delay, which could not meet the complex design. The heuristic Synthesis algorithm is presented in this paper. Based on matrix model and symbolic algebra, this paper offers a symbolic synthesis method and performs delay and crosstalk optimization simultaneously. Using the cost function the method steers the synthesis process, which considers multiple optimization objectives, including area, delay and crosstalk. The proposed algorithm was tested on a set of reversible benchmarks. Compared with existing synthesis methods, the algorithm reduces crosstalk by 10.3% and path delay by 5–20%. It has obvious advantages for CPU time and memory cost. The method can handle large scale reversible circuits in reasonable time.

Key words: logic synthesis; reversible circuits; delay; crosstalk

1 引言

随着大规模集成电路被广泛应用到各个领域, 人们对集成电路的功耗、性能和体积提出了更高的要求. 利用可逆电路代替传统的不可逆电路来降低功耗的设计策略是解决方案之一. Landauer 在其原理中阐述, 信息熵的改变(对应信息的损失)产生能量的消耗^[1]. 可逆电路是一种无信息损失电路, 它以固有的低功耗特性得到了众多设计者的关注. Z. Zilic 和 A. D. Vos 阐述了可逆逻辑在典型的 MOS 电子器件的实现, 建立了部分可逆电子电路, 这些电路仅仅由输入信号供电^[2, 3].

因为可逆电路的综合要保持它的可逆性, 所以它的综合方法与传统的不可逆电路的综合方法有着根本上的不同. 现有的可逆电路的综合方法主要有两大主流: 模板匹配法^[1~4, 6]和分解法^[7~9]. 前者建立等价模板库,

利用模板匹配的方法来找到其他实现可逆电路的方式, 达到了优化可逆电路的目的. 这种方法没有复杂的计算过程, 比较简单. 但该方法不能很好地升级, 需要大量地扩展匹配模板. 当输入数量逐渐增多时这种综合方法会因为模板过多而产生指数爆炸. 分解法通常依据布尔分解, 利用判别图进行综合优化. 但是, 这类方法只适用于输入输出相对较小的电路, 随着输入输出数量的增加, 指数级地扩展搜索空间是不可避免的. 另外, 还有一种是节点替代法^[10]. 它运用输出函数的乘积异或和表达来处理可逆电路, 然而在综合过程中他们试探性地选取不同的替代节点, 缺乏策略性.

2 数学模型

在集成电路设计中, 根据优化的逻辑函数可直接获得优化的逻辑电路. Reed-Muller 电路不但具有易测试的

特点,而且逻辑门和互连较少.用 Reed-Muller 逻辑实现某些逻辑函数比传统布尔逻辑简单得多,如算术运算部件,极性函数等.因此,本文用正极大 Reed-Muller (PPRM) 表达式来描述输出函数(正极大变量较容易得到),得到正则 PPRM 表达式,

$$F(x_1, x_2, \dots, x_n) = a_0 \oplus a_1 x_n \oplus a_2 x_{n-1} \oplus a_3 x_{n-1} x_n \oplus \dots \oplus a_{2^n-1} x_1 x_2 \dots x_n$$

其中 $a_i \in \{0, 1\}$, $0 \leq i \leq 2^n - 1$, x_j 为正极的信号, $0 \leq j \leq n$, 符号“ $\oplus$ ”表示异或(XOR)布尔操作.

在 PPRM 表达式中,如果一个输入信号被用于两个或更多的乘积项,则在其逻辑电路中可能存在一些冗余元件.例如一个简单的关于信号 a , b 和 d 的函数 $ab \oplus ad$, 也可能以 $a(b \oplus d)$, $a(b \oplus ad)$ 或者 $a(ab \oplus d)$ 形式被实现.如果门 $ab \oplus d$ 在电路实现的其他部分已经出现,则该函数以 $a(ab \oplus d)$ 形式实现,与其他函数共享 $ab \oplus d$ 门,可能降低整个电路的成本.

另一方面,例如函数 $a \oplus b \oplus c$, 也可以表达成 $a \oplus (b \oplus c)$, $(a \oplus b) \oplus c$ 或者 $b \oplus (a \oplus c)$ 形式.如果这是整个函数的表达形式,那么用上述任意一种形式来实现电路,其成本是相同的.但如果它仅仅是电路的部分表达形式,那就需要考虑它与其他部分的实现之间的相互关系.在这种情况下,应如何选择元件来组建电路呢? 为了捕捉函数间的相互依赖关系,综合技术就必须对可逆结构进行全局分析.本文用多输出函数 $F(x_1, x_2, \dots, x_n, f, g, \dots, h, \delta)$ 来描述可逆电路, $x_1, x_2, \dots, x_n$ 为原始输入, $f, g, \dots, h$ 为可逆电路的必要输出, δ 表示从原始输入到目标输出所跨越的元件数.可逆电路具有级联这种特殊的结构,利用矩阵进行分析是再合适不过了.因此,本文将可逆电路映射到如下矩阵 M 上,

$$\begin{matrix} & \delta_{i, \dots, j} \\ \begin{matrix} 1 \\ \dots \\ i \\ \dots \\ j \\ \dots \\ n+1 \end{matrix} & \begin{bmatrix} \dots & y_1(1, \delta_{i, \dots, j}) & \dots \\ \dots & \dots & \dots \\ \dots & y_i(i, \delta_{i, \dots, j}) & \dots \\ \dots & \dots & \dots \\ \dots & y_j(j, \delta_{i, \dots, j}) & \dots \\ \dots & \dots & \dots \\ \dots & f_{i, \dots, j} & \dots \\ \dots & \dots & \dots \end{bmatrix} \end{matrix}$$

其中, $y_i(i, \delta_{i, \dots, j})$ 表示在第 i 行 $\delta_{i, \dots, j}$ 列的输出.初始状态下每个元件占一行,电路用 m 个列栅表示.矩阵中的每一点都是单输出函数并用 PPRM 形式表示.为了区分 XOR 和 AND 布尔操作,在 AND 布尔操作的表达式中加上“ $\odot$ ”项.利用矩阵 M 模拟可逆电路的空间结构,它不仅表达了每列中各元素之间的操作关系,并且更重要的是,它还便于从其中提取表达成本函数所必要的因素

和各元件对应的输入输出值.例如我们用 $F(x_1, x_2, \dots, x_n, f, g, \dots, h, \delta_{i, \dots, j-1})$ 和 $F(x_1, x_2, \dots, x_n, f, g, \dots, h, \delta_{i, \dots, j})$ 分别模拟第 $\delta_{i, \dots, j}$ 列的输入输出,参数 $\delta_{\max}$ 表示最大列号,初始值为前面所提到的 m .参数 $\delta_{\max}$ 和 L 是成本函数的两个因子,其中 L 是最大行数,表示输入输出数量.为了计算路径延时,本文定义了无关列这一概念.

定义 1 如果两个相邻列(第 n 和第 $n+1$ 列)的所有同行元素都相等,则称这两列为无关列.如果这两列中的同行元素有不同,那么看该行对应的前两列(第 n 和第 $n-1$ 列)是否相同,若相同或 $n-1=0$,则第 n 和第 $n+1$ 列也为无关列.

参数 $\delta_{\max}$ 的初始值表示元件数量,能用于去估算电路的面积.为了简便,假设 NOT, CNOT 和 $(m-\text{bit})$ Toffoli 门延时分别为 d , d 和 $(m-1)d$, 其中 d 表示单位延时.可逆电路的路径延时可以直接通过合并无关列得到.

$$\mathcal{D} = \sum_{j=1}^{\delta_{\max}} \text{MAX}(\mathcal{D}_j)_j$$

其中 j 为列号; MAX 是不同行的延时取最大值的函数; $\delta_{\max}$ 为合并后的最大列号.

值得注意的是,事实上,不同门的延时是不同的.但我们能够很容易的将这个模型扩展以满足实际情况.

随着工艺的缩小,互连线的串扰问题成为影响系统整体性能的重要因素.在逻辑综合中,如果不对连线延时进行准确地估计,综合出来的结果往往会偏离设计要求.因此,本文将 Elmore 延时模型作适当调整,加入路径延时的计算中:

$$\mathcal{D} = \sum_{i=1}^m \text{MAX}(\mathcal{D}_i)_{ij} + \sum_{k=1}^N R \cdot CT_{ik}$$

其中 $(\mathcal{D}_i)_j$ 为第 i 行 j 列的门延时; R 为常数; CT_{ik} 为线网 i (第 i 行 j 列的门连接其扇出门的第 i 行的线网), k 之间串扰(该值的定义将在 3.3 节的定义 6 中给出),其他同上式.

3 综合思路

为简化可逆电路的综合,本文将可逆电路作了划分.

定义 2 令 S 是函数 F 的可逆电路实现的元件集合(个数为 k), G_i 表示一个单目标输出的连续元件集合,满足

$$G_i \subseteq S, i = 1, 2, \dots, k$$

$$\text{Targetoutput}(\{G_j, \dots, G_h\}) = \text{OneOfOutput}(F),$$

$$\text{Targetoutput}(\{G_j, \dots, G_h, G_l\}) \neq \text{OneOfOutput}(F),$$

$$j, h, l = 1, 2, \dots, k \text{ 且 } l \neq j \dots h$$

$$\bigcup_{i=1}^k G_i = S$$

则集合 $\{G_j, \dots, G_h\}$ 是 S 的一个划分.

定义 2 表明, 一个划分是由一系列门组成, 这些门的目标线或者它们的最终目标是唯一的, 确保每个划分的目标输出为单输出函数, 从而提高算法的收敛速度并减小复杂度, 并且根据每个划分的最大列号来为每个划分编码排序。值得注意的是, 因为针对不同的输出存在不同的划分, 所以各个划分是可重叠的。

尽管通过划分可以在一定程度上减少了搜索空间, 但是要找到一个好的转化等价表达式, 搜索空间仍然很大, 需要一个启发式的算法来解决这一问题。

3.1 符号代数方法

PPRM 布尔运算具有交换律、结合律、分配律、幂等律等代数特性。因此, 本文将 PPRM 中的 XOR 和 AND 布尔运算映射为布尔多项式中的加法和乘法运算, 这个布尔多项式的运算类似于实数多项式的加法和乘法运算, 仅仅需要附加一个额外的约束条件, $x \cdot x = x$, 就可以利用^[12]中提出的一种以 Gröbner 基为数学基础的分解策略来获得候选门。基本思想就是将布尔多项式分解得到可用因子。这种分解算法的计算复杂度为多项式级的, 能有效地避免指数爆炸的发生。本文通过上述映射得到的布尔多项式的单变量最高次数为 1, 而且该布尔多项式中每项的系数为 1, 从而进一步降低了算法的计算量。

给定一个可逆电路的矩阵表达 M , 其划分为 $Par[n]$, 可逆逻辑门库 Lg (仅包括 NOT, CNOT 和 Toffoli 门) 作为通用库。为了确定库中元素, 利用 branch and bound 算法得到确定性的解树。根据字典映射将每个划分的 PPRM 表达式映射到加法器和乘法器(带有附加运算规则 $x \cdot x = x$)上。Simplify() 是化简函数。Simplify 的结果被存到树形数据结构中。如果 Simplify 的结果等于库中某一元素的 PPRM 表达, 则找到一个解, 并标记相应的节点。若没找到相应的库元素与其对应, 则递归应用上述步骤于新的节点。该算法以路径延时为限制条件, 竭力找到小于初始路径延时的解树。具体算法如下所示。

算法 1 分解算法

```
# nrD is path delay of nrth partition.
Procedure GuidedDecomposition( Par[ m ], m- D, Lg ) {
# initialize a solution tree
solutiontree ← tree(Par[ m ]); depth ← 0; bound ← nrD;
for all n ∈ in solutiontree with depth = depth do {
if depth = 0 then
choose all sr ∈ Lg that preserve the Par[ m ] structure;
else for all sr ∈ Lg do {
if cost of sr + cost of node n < bound then {
result = simplify( n, sr );
# make result a child of node n
addchild( n, result );
add cost of sr to cost of result;
```

```
if result ∈ Lg then { # solution is found.
mark the corresponding tree node;
bound = cost of node result; }
if no more n ∈ in solutiontree with depth = depth
then depth ← depth + 1 }
return solutiontree marked; }
```

利用这种算法, 可以得到表达式的等价转换形式, 并可以建立候选门集, 然后检验转化后的门是否存在于原矩阵中, 若存在, 则作相应调整(要求函数根据这些被确定的门选择等价转换形式), 但这种调整不应该改变电路输出; 若不存在, 将其加入电路, 再作上述调整。如果调整成功并且所得方案优于以前方案, 则保留该调整, 否则调整不被允许。

然而, 一个重要的问题是如何解决那些可能存在两个或更多转化形式的状况。例如, 考虑综合一个简单函数 $ab \oplus bc \oplus cd$, 它可以转化为多种形式 $b(a \oplus c) \oplus cd$, $b(a \oplus bc) \oplus cd$ 和 $ab \oplus c(b \oplus d)$ 等等。这就需要为候选门设置优先级的策略来解决这类问题, 如图算法 3。

算法 2 候选门预算

```
Procedure budget(M, Par[ n ]) {
for m= 1 to NumberOfPartions do {
CandidatesComponents ( Par[ m ]) ← GuidedDecomposition
( Par[ m ], nrD, Lg );
# Note that function GuidedDecomposition(Par[ m ], nrD, L)
return the solutiontree rather than the best solution[ 12].
candidates ← CandidatesComponents(Par[ m ]); }
priority_candidate gates ← priority( candidates );
return ( CandidateComponents ( Par [ n ] ), priority_
candidate gates ); }
```

在对可逆电路进行初始划分后, 针对每个划分创建候选门, 再根据每个候选门的利用率, 给予其不同的优先权。利用率越高的门, 优先级就越高, 用这些门更容易得到最终结果。按照划分编码的降序, 对每个划分做上述处理。对于每个划分, 综合时将从该划分的优先队列中弹出优先级最高的门。值得注意的是, 对于每一划分, 通过这一算法得到的是候选树, 在排除某一候选门的同时也删除了候选树中该门下的分支, 这将进一步缩小搜索空间。另外, 考虑到元件之间的相关性, 在综合完一个划分后, 应该考核是否一些元件可能并入其他划分中或者从划分区中去除, 并调整相应的划分序号。

3.2 缩减延时

在综合过程中, 通过简单地调整门间位置的策略来缩减路径延时。若相同行的两个“ $\odot$ ”操作互换不影响输出, 则这两个操作是可互换的。通过门间互换得到更多

的无关列,从而达到缩减路径延时的目的。

下面利用一个例子来加以说明。图 1 为一个可逆电路,其中门 1 和 1^0 是无关的。该电路延时分布如下(其中门延时中既包括门延时,也包括线网延时(详见第 2 部分延时模型)):

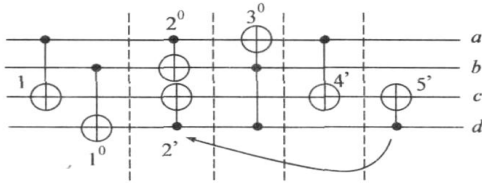


图 1 一个可逆电路

$$\mathcal{D}_a = \mathcal{D}_b = \mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3, \mathcal{D}_d = \mathcal{D}_0$$

$$\text{原始 } \mathcal{D}_c = \max(\mathcal{D}_1, \mathcal{D}_1) + \mathcal{D}_2 + \mathcal{D}_3 + \mathcal{D}_4 + \mathcal{D}_5(2).$$

其中门 4 和 5 是可互换的,互换后,门 5 被门 2 所代替,门 2 和 2^0 成为无关的, $\mathcal{D}_c$ 的值将被改变。

$$\mathcal{D}_c = \max(\mathcal{D}_1, \mathcal{D}_1) + \max(\mathcal{D}_2, \mathcal{D}_2) + \mathcal{D}_3 + \mathcal{D}_4$$

$$\text{这样, } \mathcal{D} = \max(\mathcal{D}_a, \mathcal{D}_b, \mathcal{D}_c, \mathcal{D}_d) = \mathcal{D}_c$$

$$\text{被节省的延时} = \mathcal{D}_2 + \mathcal{D}_2 - \max(\mathcal{D}_2, \mathcal{D}_2)$$

3.3 交换线间排列(permuting wires)

在综合过程中,可以借助动态串扰优化算法^[11]分析估算串扰参数 $M(X, Y)$ 、 $D(A, B)$ 和 C_{ij} 。 $M(X, Y)$ 为波形 X, Y 的跳变数之和,用来反映跳变频度对串扰的影响。 $D(A, B)$ 为信号波形 A, B 的距离,反映了波形之间的相似程度,其值越小,波形就越一致,串扰也越小。 C_{ij} 为线网 i, j 之间耦合电容。然而,单单根据波形距离不足以判断波形间的相似性,还需要考虑波形的形状,分析波形之间的相关性。本文在此定义了波形相似性的概念。

定义 3 $V_A(t)$ 和 $V_B(t)$ 分别为波形 A 和 B 在 t 时刻的布尔值。 (T_1, T_2) 是进行模拟的时间范围。文献[11]将

$$D(A, B) = \frac{\int_{T_1}^{T_2} |V_A(t) - V_B(t)| dt}{T_2 - T_1}$$

定义为输出波形 A, B 的距离。

定义 4 两波形的相关性定义如下:

协方差

$$\text{Cov}(A, B) = \iint V_A(t) V_B(t) dt dt - \int V_A(t) dt \int V_B(t) dt$$

相关系数

$$\rho = \frac{\text{Cov}(A, B)}{\sqrt{\text{Cov}(A, A) \text{Cov}(B, B)}}$$

$|\rho| \leq 1$, ρ 是用来刻画 A 和 B 之间线性相关程度的。当 $\rho = 1$ 时,波形 A 和 B 线性相关。当 $\rho = 0$ 时,它们不相关。当 $\rho < 0$ 时,两波形为负相关。相反,当 $\rho > 0$ 时,它们为正相关。

定义 5 波形 A 和 B 的相似性定义如下:

$$\text{SIM}(A, B) = \frac{\rho}{D(A, B)}$$

$\text{SIM}(A, B)$ 的值越大,波形 A 和 B 越相似。根据 $\text{SIM}(A, B)$ 的大小,交换线间排列(permuting wires),使波形相似的线邻近,减小跳变耦合影响(switching coupling effect),减小线间串扰,从而进一步降低功耗和延时。图 2 为交换前和交换后的电路示意图。

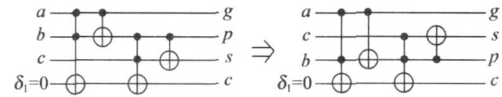


图 2 全加器交换线间排列

定义 6 W_i 和 W_j 为线网 i 和 j 的信号波形, $M_I(W_i, W_j)$ 是在激励波形组 I 下的 $M(W_i, W_j)$, 同理得到 $\text{SIM}_I(W_i, W_j)$ 。线网 i 和 j 之间的串扰定义如下:

$$CT_{ij} = \frac{\sum_{I \in S} M_I(W_i, W_j) C_{ij}}{\text{SIM}_I(W_i, W_j)}$$

其中 C_{ij} 为线网 i 和 j 之间的耦合电容; S 为原始输入波形组集; N 为所有线网集合; 进行串扰优化的线网集合。 W_i 和 W_j 可从基于布尔过程的波形计算方法得到。

3.4 成本函数 CF 的确定

算法中必然要考虑到局部优化对整个电路性能的影响,需要对成本函数 CF 作如下定义:

$$CF = \alpha \delta_{\max} + \beta \mathcal{D} + \theta L + \tau CT_{ij}/K$$

其中 $\delta_{\max}$ 为矩阵模型的最大列数; $\mathcal{D}$ 为路径延时; L 为电路矩阵表达的最大行数; K 为该元件在所得候选门中出现的次数。这里的 CT_{ij}/K 则是针对整个电路的; 权值 α, β, τ 和 θ 的和为 1。选择这些权值是由各因子 ($\delta_{\max}, D, L, CT_{ij}/K$) 的独立性,同时还要考虑不同的设计倾向来决定,用于适应不同的设计优化目标(如需要更小的面积还是更短的延时)。各因子的复相关系数 (R) 是利用数理统计学中多元回归方法来计算的,为回归平方和 ($S_{\square}$) 与总偏差平方和 ($S_{\text{总}}$) 之比的开方值,可以利用 SAS 软件得到。

$$R = \sqrt{S_{\square}/S_{\text{总}}}$$

其中 $S_{\square}$ 为因子取不同值所引起的成本函数值的差异, $S_{\text{总}}$ 为各因子随机误差所引起的成本函数的差异。复相关系数越大,所赋的权值越小。取复相关系数的倒数加上优化指标在用户心中所占的比重(由用户给出)作为得分,再经过归一化处理得到权值。

4 综合算法

应用上述综合思路可以得到一个优化的结果。图 3 为该算法的流程图。为了简化,只考虑用通用的 NOT、CNOT 和 (m bit) Toffoli 门来表示可逆电路,以此作为综合过程的起点,这样能相对容易地提供一个程序去替换

电路中的门以达到综合的目标.

考虑将可逆函数的任意一种级联形式作为算法的输入, 输出为优化的可逆电路. 首先, 执行预处理过程:

(1) 调整电路尽量让相同输出的各个门相邻, 并且排除无用门, 就是排除那些不被必要输出所用的那些门, 其伪代码如下.

if TargetOutput(g_i) $\nsubseteq$ RequiredOutput then delete(g_i);

(2) 将可逆电路映射到矩阵 M 上. (3) 通过第 3 部分所述的两个子程序进行划分并预算出候选元件. 在候选元件中找到出现次数最多的元件, 将其作为断言函数(assertion function) $A(x)$. 以尽量选择出现次数多的元件为原则, 构建关于 $A(x)$ 的最简分解形式, 进行延时优化, 进行性能比较, 找到最优解. 再通过动态串扰优化算法^[11]对结果进行预布线, 并分析估算串扰参数. 再根据 $SIM(A, B)$ 的大小, 交换线间排列(permuting wires). 接下来, 算法进入了一个循环. 针对每个划分, 在上面布线结果中找到使 CT_g/K 最小的相邻元件对作为 $A(x)$, 选择关于 $A(x)$ 的候选分解形式, 进行预布线以及优化过程, 找到使成本函数 CF 最小的电路实现形式. 这一循环直到所选断言函数不变为止.

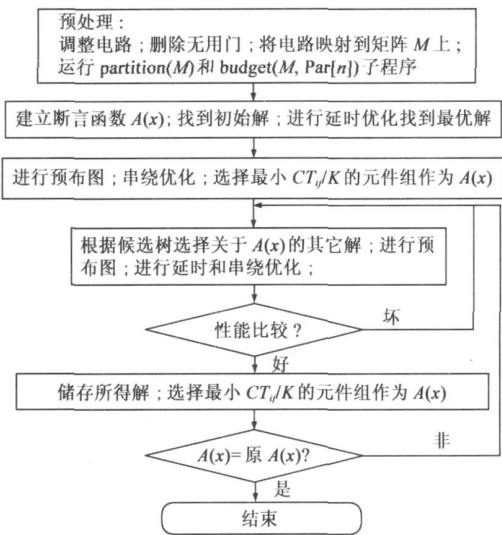


图 3 算法流程图

5 实验结果

为了测试本文所提供综合方法的有效性, 我们用 C++ 语言来实现本文所描述的算法, 用文献[1~10]中的基准可逆电路做实验. 实验是在 Pentium4 3.0G PC (512M 内存), WindowsXP 环境下完成的.

表 1 可逆基准电路的综合结果

Examples	Our			Best			GT		
	# gates	delay	crosstalk	# gates	delay	crosstalk	# gates	delay	crosstalk
4mod5	5	7. 81	21	5	8. 65	27	/	/	/
6sym	20	27. 34	329	20	28. 46	346	/	/	/
rd53	12	27. 50	432	12	29. 13	475	30	51. 26	669
ham7	25	21. 46	561	25	25. 59	717	/	/	/
ham15	109	105. 96	1291	109	117. 32	1374	23	34. 82	638
hwb8	751	2089. 74	3214	749	2357. 12	3596	132	129. 34	2197
hwb9	1541	5472. 46	7639	1544	6036. 74	8471	/	/	/
C1355	238	3915. 65	1356	271	/	/	/	/	/
C3540	935	7847. 32	5108	1007	/	/	/	/	/
C5315	1397	5932. 67	7953	1401	/	/	/	/	/
C6288	2132	14704. 18	11365	2216	/	/	/	/	/

表 1 中显示了对本文算法和已有算法的比较结果. 其报导的是利用不同的算法化简可逆电路的结果. “Our” 列是利用本文方法所得到的综合结果. “Best” 是现有不同综合方法^[1~10]所得的最好综合结果集. GT 列是其中一种通用 Toffoli 门综合方法^[5,6]的结果.

在表 1 的“our”列的结果中, 假设权值为指定值 $\alpha=0.25$, $\beta=0.5$, $\theta=0.2$ 并且 $\tau=0.05$. # gates 列显示的是可逆电路中所用门的数量. delay 列是主路径的门的累计延时. crosstalk 为这些基准电路的总串扰结果比较. 由于已有方法并没有路径延时和串扰得实验数据, 为了显示本文方法的优势, 我们根据文献中给出的电路结果, 再利用本文的计算模型得到了相应的“Best”和 GT

方法中的 delay 和 crosstalk 数据. 但文献中没有给出 C1355、C3540、C5315 和 C6288 的电路结果或机器码, 所以无法推算出它们的路径延时和串扰值. 这些基准电路集的实验结果充分显示了本文的算法在解决可逆电路的综合问题上卓有成效. 利用这种启发式算法, 就能够得到现有文献中给出的最好面积, 总串扰平均得到了 10.3% 改善. 在延时方面平均能得到 9.8% 的改善, 并且其中一些电路的延时可以节省 15%~20% 之多. 通过选择合适的权值, 这种延时方面的改善不会影响面积.

表 2 为本文算法与 GT 的 CPU 时间和存储开销的比较, 结果表明本文算法在性能上的优越性, 它有能力

在有效的时间内实现大电路的综合。

表 2 可逆电路的 CPU 时间和存储开销比较

circuits	Our		GT	
	CPU (s)	MEM (MB)	CPU (s)	MEM (MB)
4mod5	0	0.1	/	/
han15	2.5	5	4	9
hwb8	3.9	38	13	44.5
C1355	4.7	7	10.6	/
C3540	20.3	41.2	40	/
C5315	13.9	28.4	> 200	/
C6288	135.7	76.2	> 900	/

6 总结

本文介绍了可逆逻辑的基于矩阵模型的多目标的综合方法。该算法是先建立划分区,然后在每个划分区进行布尔分解得到候选门集,根据断言函数,竭力找到好的结果,这一结果不但考虑到面积、延时目标,而且考虑了串扰因素。这种启发式算法在多个电路上进行测试,其结果证明了它有广泛的应用前景。同时,我们发现成本函数的权值对综合结果起着相当重要的作用,如何在综合过程中动态的调整权值,将会在综合方面取得更大的突破。这也是我们下一步要研究的方向。

参考文献:

- [1] D M Miller, D Maslov, G W Dueck. A transformation based algorithm for reversible logic synthesis[A]. Proceedings of the 40th conference on Design Automation [C]. Anaheim, CA, USA: IEEE Computer Society/ ACM Press, 2003. 318– 323.
- [2] Z Zilic, K Radecka, A Kazanipour. Reversible circuit technology mapping from non-reversible specifications[A]. Proceedings of the conference on Design, Automation and Test in Europe [C]. Nice, France: IEEE Computer Society, 2007. 558– 563.
- [3] A D Vos, Y V Rentergem. Reversible computing: from mathematical group theory to electronical circuit experiment[A]. Proceedings of the 2nd Conference on Computing Frontiers [C]. Ischia, Italy: ACM Press, 2005. 35– 44.
- [4] D Maslov, G W Dueck, D M Miller. Toffoli network synthesis with templates[J]. IEEE Transactions on Computer Aided De-

sign of Integrated Circuits and Systems, 2005, 24(6): 807 – 817.

- [5] G W Dueck, D Maslov. Reversible function synthesis with minimum garbage outputs[A]. The 6th International Symposium on Representations and Methodology of Future Computing Technologies[C]. Trier, Germany, 2003. 154– 161.
- [6] D Maslov, G W. Dueck. Garbage in reversible design of multiple output functions[A]. The 6th International Symposium on Representations and Methodology of Future Computing Technologies[C]. Trier, Germany, 2003. 162– 170.
- [7] P Kerntopf. A new heuristic algorithm for reversible logic synthesis[A]. Proceedings of the 41th conference on Design Automation[C]. San Diego, California, USA: IEEE Computer Society/ ACM Press, 2004. 834– 837.
- [8] P Kerntopf, M. Perkowski. Function driven linearly independent expansions of boolean functions and their application to synthesis of reversible circuits[A]. Proceedings of the 13th International Workshop on Logic and Synthesis[C], Dana Point, Laguna Beach, California: ACM SIGDA (special interest group design automation), 2003. 28– 30.
- [9] M H A. Khan, M A Perkowski, M. R. Khan. Ternary Galois field expansions for reversible logic and Kronecker decision diagrams for ternary GFSOP minimization (Galois field sum of products) [A]. Proceedings of the 34th international symposium on Multiple Valued Logic [C]. Toronto, Canada: IEEE Computer Society, 2004. 58– 67.
- [10] A Agrawal, N K Jha. Synthesis of reversible logic[A]. Proceedings of the conference on Design, Automation and Test in Europe[C]. Paris, France: IEEE Computer Society, 2004. 710 – 722.
- [11] 冯刚, 马光胜, 杜振军. 动态串绕优化的开关盒布线[J]. 半导体学报. 2005, 26(2): 399– 405.
Feng Gang, Ma Guangsheng, Du Zhenjun. Dynamic crosstalk optimizing for switchbox routing[J]. Chinese Journal of Semiconductors, 2005, 26(2): 399– 405. (in Chinese)
- [12] A Peymandoust, G De Micheli. Application of symbolic computer algebra in high level data flow synthesis [J]. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2003, 22(9): 1154– 1165.

作者简介:

胡靖女, 1974 年出生于黑龙江, 哈尔滨工程大学计算机学院博士研究生。主要研究方向为数字系统的综合与优化、系统的性能分析模型。E-mail: hjlyh@vip.sina.com



马光胜男, 1944 年出生于山东, 哈尔滨工程大学计算机学院教授, 博士生导师。主要研究方向为 SoC 设计方法学、VLSI 设计自动化技术、EDA 算法及软件、计算机体系结构等。
E-mail: maguangsheng@hrbeu.edu.cn

