[文章编号] 1003－4684(2007)02 0001 04

用布尔可满足性验证逻辑电路的等价性

刘 歆, 颜 萍
(湖北工业大学电气与电子工程学院, 湖北 武汉 430068)

[摘 要] 提出了使用布尔可满足性来验证数字电路的等价性验证方法. 这一验证方法把每个电路抽象成一个有限状态机, 为两个待验证的电路构造积机, 把等价性验证问题转换成了积机的断言问题. 改进了 Tseitin 变换方法, 用于把电路约束问题变换成合取范式公式. 用先进的布尔可满足性求解器 zChaff 判定积机所生成的布尔公式的可满足性. 事例电路验证说明了该方法的有效性.

[关键词] 设计验证; 等价性验证; 逻辑电路; 布尔可满足性; 合取范式
[中图分类号] TP274 [文献标识码]: A

超大规模集成电路功能正确性是最基本的要求, 这是设计验证所要解决的问题. 如今设计验证面临着巨大的挑战, 这主要是由于超大规模集成电路的规模正如摩尔定律所指出的那样呈指数增长. 另外, 它与企业的经济效益直接相关. 因此必须要有高效的设计验证方法将设计错误发生的可能性降到最低, 同时又要缩短产品面市时间, 从而实现经济目标. 一般来讲, 设计验证方法包括两大类: 仿真方法和形式化方法.

功能和时序验证在传统上主要采用仿真方法. 尽管仿真方法得到了设计者广泛的认可, 但是它有着重大的缺陷. 仿真是对巨大的状态空间进行取样实现的, 所以仿真方法是不完备的, 即其验证的正确性局限于所施加的激励信号; 再者, 对于具有数百乃至数千万个逻辑门的集成电路来说, 其所要求的仿真向量及仿真时间是极其巨大的.

形式验证使用严格的数学推理来证明一个系统满足全部或部分设计规范. 这种验证技术已得到了广泛认可, 并且正逐渐替代传统的仿真验证方法而作为主要的设计验证手段. 验证是针对不同的设计抽象层次而言的, 所以形式化验证方法大体上又可分为 3 类: 等价性检验、模型检验、性质检验.

本文研究的是等价性验证方法, 即形式化证明两个设计实现在功能上是等价的. 这一验证方法是针对设计过程中的同一抽象层次, 如比较逻辑优化前后的两个实现模型 M_1 和 M_2 是否等价, 即: $M_1 = M_2$.

得到广泛应用的等价性验证方法是基于 BDD (Binary Decision Diagram)的方法, 其原理是证明两个电路的 ROBDD[1] (Reduced Ordered Binary Decision Diagram)图是否同构, 若同构则说明两个设计是等价的. BDD 是一种数据结构, 能够用来描述大的布尔函数, 也可以同时描述所有的选择情况. 但是由于存在不可预料的存贮器需求量, 因此在处理特殊电路(如乘法电路)及大规模电路时会遇到难以克服的困难[2] .

最近几年, 由于布尔可满足性的求解取得了长足的进步, 并可以对很大的问题事例进行逻辑推理. 因此, 本文使用了基于布尔可满足性的建模工具来解决等价性验证问题, 它可克服基于 BDD 方法的不足之处[3, 4] . 实验说明了此种方法的有效性.

本文在接下来的部分首先给出了基本的定义和概念, 电路约束问题到 CNF 的转换方法, 在第四部分则给出了基于 SAT 的等价性验证算法和测试事例, 最后是结论及未来的研究方向.

1 布尔可满足性

一个文字 L 是一个原子命题变量 p, 或者是它的否定形式 $\neg p$; 一个子句 D 是文字的析取; CNF

[收稿日期] 2006－10－30
[基金项目] 国家自然科学基金(50390060, 50335020) .
[作者简介] 刘 歆(1963－), 男, 湖北黄石人, 湖北工业大学副教授, 研究方向: VLSI 电路的设计验证、测试与逻辑综合.

(Conjunction Normal Form)公式 F 则是子句的合取. 它们的形式定义如下:

$$L ::= p \mid \neg p,$$
$$D ::= L \mid L \vee D,$$
$$F ::= D \mid D \wedge F.$$

例如, 考虑一个 CNF 公式:

$$(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q.$$

该 CNF 公式有 3 个子句, 即 $\neg q \vee p \vee r$, $\neg p \vee r$ 和单文字子句 q. 为什么要关心 CNF 形式的公式呢? 其原因就在于容易检查 CNF 公式的有效性. 依据 $\wedge$ 算符的语义, 该公式为逻辑真 $\models (\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$ 的充要条件是组成它的 3 个子句皆成立, 即有如下的 3 个关系式成立, 即:

$$\models \neg q \vee p \vee r,$$
$$\models \neg p \vee r,$$
$$\models q.$$

如果至少有一组变量赋值使得 CNF 公式 F 计算结果为真, 那么称该公式 F 是可满足的, 这组可满足性赋值称为 F 的一个模型, 或一个赋值.

针对布尔可满足性问题, 现已出现了许多求解方法, 包括局部搜索、回溯搜索、代数运算等等. 而求解来自 EDA 领域的 SAT 应用问题, 已证明回溯搜索算法是最有效的[5]. 这种方法的基础是 DPLL 算法[8], 再将其扩展为具有学习(learning)[6] 和非时序回溯(non - chronological)[7] 的技术, 这极大地提高了修剪搜索空间的能力. 学习就是提取并且记忆来自先前已搜索空间的信息, 以便为修剪下一步和以后的搜索提供先验知识, 学习是利用增加子句到现有的子句库中取得的. 非时序回溯就是蕴涵过程中出现冲突赋值时, 分析其原因. 如果引起冲突赋值出现在决策树的较高层, 就直接跳回到那一层去. 这与时序回溯方法形成了鲜明的对比, 它是一步一步的返回上一决策层的. 正是由于这些先进的求解算法, 才使得基于 CNF 形式的知识描述能够处理大的逻辑推理事例问题.

2 电路约束问题到 CNF 公式的转换

许多可满足性问题, 其自然描述形式并非是 CNF 形式. 电路应用问题的自然描述形式是电路图, 其约束形式是电路拓扑结构, 以及逻辑关系约束. 而现今最有效最先进的 SAT 求解工具以 CNF 形式对知识进行描述. 因此需要有效的方法把非子句形式的 SAT 问题转换成 CNF 描述形式.

考虑逻辑公式:

$$\varphi = (\ldots(x_1 \leftrightarrow x_2) \ldots) \leftrightarrow x_n.$$

如果应用简单的转换算法, 它将会产生 2^{n-1} 个子句的 CNF 公式. 这个例子说明, 把一种描述形式转换为另一种描述形式, 有可能会导致结果按指数规模增长. 针对这一问题, Tseitin[8] 提出了一种有效的转换方法. 对于前面的公式 φ, 应用 Tseitin 转换方法, 得到的 CNF 公式共有 $4(n-1)+1$ 个子句, 与最初的命题公式大小成线性关系.

Tseitin 转换是把一般的命题公式转换为 CNF 形式的最常见的方法. 但直接将 Tseitin 转换方法用于电路问题并不适合, 笔者将 Tseitin 变换方法修改后用于电路应用问题. 具体就是对于电路 C 中的每一个输出为 y 的逻辑门, 为它构造一个如下的相容性函数:

$$\sigma_y(I(y), y) = \neg(f(I(y)) \oplus y).$$

根据每个节点函数的定义, 只有当节点 y 与其对应的输入节点 $I(y)$ 的取值相容时, 布尔函数 $\sigma_y(I(y), y)$ 的计算结果才为真. 为了构造事例 CIRCUIT - SAT(C)的 CNF 公式, 首先为 C 中每一个节点的相容性函数构造 CNF 公式, 然后合取这些节点的相容性函数的 CNF 公式, 并且增加一个单文字子句, 以断言所求的目标为真. 最后的描述就是对应于事例 CIRCUIT - SAT(C)的 CNF 公式. 上面的构造方法要求为每一个节点的相容性函数 $\sigma_y(I(y), y)$ 构造 CNF 公式, 如直接简化 $\sigma_y(I(y), y)$ 的真值表以得到一个 PoS(Product of Sums)描述. 在实际中, 首先是利用布尔函数分解技术把最初的电路 C 分解成简单逻辑门的集合, 然后利用上面介绍的相容性函数方法为每一种简单的逻辑门构造 CNF 公式. 这样就只需处理几种类型的逻辑门情况, 极大地简化了问题的复杂性.

下面举例说明简单逻辑门的 CNF 公式的构造方法.

例: 设有一个二输入的 AND 逻辑门, 它的输入为 a 和 b, 输出为 c, 那么该逻辑门的逻辑行为可以用方程式描述为

$$c = a \cdot b.$$

该方程式也可以用另一种形式表示为:

$$ab \oplus c = 0.$$

或者

$$\bar{a}c + \bar{b}c + ab\bar{c} = 0,$$

或者

$$(a \vee \neg c)(b \vee \neg c)(\neg a \vee \neg b \vee \neg c) = 1.$$

现在可以看出上面方程式的左边是以 CNF 形式表示的, 称之为逻辑门的 CNF 描述. 只有"与门"真值表中的赋值组合才满足该 CNF 公式, 它是根据逻辑门的函数强制性地使得逻辑门的输入和输出赋值相容.

上面的例子是针对二输入“与”逻辑门的，多输入“与”逻辑门的变换问题是个线性问题，如

$$y = \mathrm{AND}(x_1, x_2, \cdots, x_n),$$

其变换后的 CNF 公式为

$$\left(\prod_{i=1}^{n}(x_i \vee y)\right)\left(\sum_{i=1}^{n}\neg x_i \vee y\right).$$

求取其它逻辑门 CNF 公式的详细方法可参见文献[9].

3　等价性验证算法和测试事例

为了验证本文所提出的基于 SAT 的等价性验证方法，本文设计了适合于计算机处理的布尔表达式文法，其目的是用于描述待处理布尔电路的网表文件. 这个布尔电路的网表文件格式所遵循的 BNF 文法如图 1 所示.

```
Input       : Expression";"Expression";
Expression  : Term[("+"|"*"Expression]
Term        : Factor[(["&"]|"|")Term]
Factor      : Primary | "!" Factor
Primary     : Literal | "(" Expression ")"
Literal     : Variable [ "'" ]
Variable    : "A" | "B" | "C" | ... | "Z"
              | "a" | "b" | "c" | ... | "z"
```

图 1　描述布尔表达式的 BNF 文法

图 1 中，“|”表示的是“与非”运算符，其优先级和与操作符“&”一样.“与”运算的优先级高于或运算“+”.“异或”运算“*”的优先级与“或”运算的优先级相同. 非运算操作符“!”有最高优先级. 变量名用单个字母表示.

本文所采用的等价性验证算法如图 2 所示，并且用 C++ 编程语言实现了本文所提出的等价性验证算法，这个算法框架集成了当今最先进的 CNF SAT 求解器 zChaff[10,11] 的算法.

```
Equivalent_checking ( Circuit_A, Circuit_B) {
    Parse circuit file;
    T = Construct a syntax tree;
    φ = Generate CNF formula;
    Solve  φ  if ( SAT_Solve (φ) == Satisfiable )
      return Circuit_A equivalent to Circuit_B;
    else
      return Circuit_A unequivalent to Circuit_B;
}
```

图 2　等价性验证算法

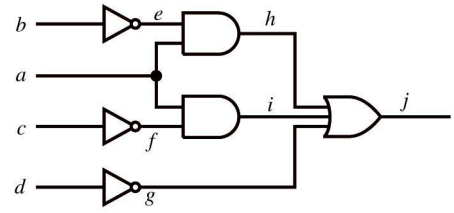
两个待验证的电路描述文件是 Circuit_A 和 Circuit_B，在算法分析了这两个电路文件并建立了相关的内部数据结构后，将产生两个句法树，之后则由该句法树生成两个待验证电路所构成的积机的 CNF 公式 φ，将其交由后端的 CNF SAT 求解器进行求解. 如果得出的结果证明布尔公式 φ 是可满足的，那么两个待验证的电路是等价的，否则是不等价的.

在实际中，本文对大量的电路实例进行了验证. 但为了方便起见，以下仅列举出一个小的事例电路作为说明. 两个待验证电路的逻辑表达式描述如下：

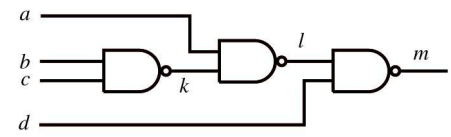
电路 (a)：a&b′+a&c′+d′；

电路 (b)：((b|c)|a)|d.

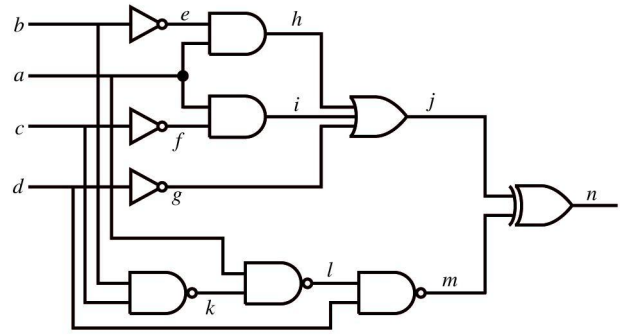
图 3(a)和(b)分别是这两个电路所对应的电路图. 这两个电路是否等价呢？因此需要构造验证它们等价的验证电路，即两者所构成的积机电路，如图 3(c)所示.



(a) 电路 a



(b) 电路 b



(c) 积机电路

图 3　等价验证的事例电路

将该电路进行转换，得出验证该电路的 CNF 公式为下面的公式 φ1 所示. 经求解可得公式 φ1 是可满足的，所得到的一组验证测试向量为：{ a, b, c, d } = {0, 0, 1, 0}. 所以说该测试事例的两个电路是等价的.

$$\varphi_1 = (b \vee e) \wedge (\neg b \vee \neg e) \wedge (c \vee f) \wedge (\neg c \vee \neg f) \wedge (d \vee g) \wedge (\neg d \vee \neg g)$$
$$(a \vee \neg h)(e \vee \neg h)(\neg a \vee \neg e \vee h) \wedge (a \vee \neg i) \wedge (f \vee \neg i) \wedge (\neg a \vee \neg f \vee i)$$
$$(\neg g \vee j) \wedge (\neg h \vee j) \wedge (\neg i \vee j) \wedge (g \vee h \vee i \vee \neg j) \wedge (b \vee k) \wedge (c \vee k) \wedge$$
$$(\neg b \vee \neg c \vee \neg k)(a \vee l) \wedge (k \vee l) \wedge (\neg a \vee \neg k \vee \neg l) \wedge (d \vee m) \wedge$$

$$(l \vee m)(\neg d \vee \neg l \vee \neg m)$$
$$(j \vee \neg m \vee n) \wedge (\neg j \vee m \vee n) \wedge$$
$$(j \vee m \vee \neg n) \wedge (\neg j \vee \neg m \vee \neg n) \wedge (\neg n).$$

4 结束语

本文所采用的验证方法是基于 CNF SAT 形式的逻辑推理方法,这是最近几年才出现的一个新兴研究领域. 本文通过事例电路进行了验证测试实验,证明这一方法是非常有效的. 在这一新的研究领域里,有很多理论问题有待深入研究和探讨,包括有效的布尔可满足性求解方法、更加有效的形式化验证方法,以及非子句形式的 SAT 问题到 CNF SAT 的转换方法.

基于 CNF SAT 的等价性验证方法是十分有效的,但其性能极大地受制于后端 SAT 搜索引擎实际效率的影响. 而在 VLSI CAD 领域中的绝大多数问题是 NP 完全性问题或难解问题,它们要求我们不断探索新的启发式算法,从而更好地限制和修剪非解子空间. 笔者相信这一领域提供了很多机会,本文的工作只是在此方向上所迈出的一步.

[参 考 文 献]

[1] Bryant R E. Graph Based Algorithms for Boolean Function Manipulation[J]. IEEE Transactions on Computers, 1986, 35(8): 677 – 691.

[2] Bryant R. E. On the Complexity of VLSI Implementation and Graph Representations of Boolean Functions with Application to Integer Multiplication[J], IEEE Transactions on Computers, 1991, 40(2): 205 – 213.

[3] McMillan K. Applying SAT Methods in Unbounded Symbolic Model Checking[C]. Las Vegas: In Proc. Intl. Conf. on Computer Aided Verification, 2002.

[4] Biere A, Cimatti A, Clarke E, et al. Symbolic Model Checking without BDDs[C]. Zurich: In Proc. Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems, 1999.

[5] Silva L G, Silveira L M, Marques-Silva J. Algorithms for Solving Boolean Satisfiability in Combinational Circuits[C]. Munich: Proceedings of Design Automation and Test in Europe (DATE), 1999.

[6] Davis M, Logemann G, Donald Loveland. A Machine Program for Theorem proving[J]. Communications of the ACM, 1962(5): 394 – 397.

[7] Marques Silva J P, Sakallah K A. GRASP: A Search Algorithm for Propositional Satisability [J]. IEEE Transactions on Computers, 1999, 48(5): 506 – 521.

[8] Tseitin G S. On the Complexity of Derivation in Propositional Calculus. Studies in Constructive Mathematics and Mathenatical Logic[C], New York: [s. n.], 1968.

[9] 刘 欲. 基于布尔可满足性的数字电路测试生成算法[J]. 电子测量与仪器学报, 2002, 16: 1 – 9.

[10] Zhang L, Madigan C F. Moskewicz M H. Efficient Conflict Driven Learning in a Boolean Satisfiability Solver[C]. New York: Proceedings of International Conference on Computer Aided Design, ACM Press, 2001.

[11] Moskewicz M H, Madigan C F, Zhao Y, et al. Chaff: Engineering an Efficient SAT Solver[C]. New York: Proceedings of the 38th Design Automation Conference, ACM Press, 2001.

Verifying Equivalence of Logic Circuits with Boolean Satisfiability

LIU Xin, YAN Ping

(*School of Electrical & Electronic Engin., Hubei Univ. of Technology, Wuhan* 430068, *China*)

Abstract: This paper presents the equivalence verification method of digital circuits by means of Boolean Satisfiability (SAT). It first extracts the circuit under test into a finite state machine (FSM), and then builds a product machine of two finite state machines, which transforms the problem of equivalence of two circuits into the one of the asserted product machine. The Tseitin's transformation method is improved and is used for converting the circuit constrained problems to CNF (Conjunctive Normal Form) formulas. Its satisfiability is then handed over to the state of the art solver Chaff to check. The testing of example circuits demonstrates the effectiveness of this approach.

Keywords: design verification equivalence verification logic circuits; boolean; satisfiability; CNF

[责任编辑: 张岩芳]