基于 BDD 的组合电路等价性检验方法

Combinational Equivalence Checking Based on BDDs

李光辉[1,2] 邵明[1] 李晓维[1]

[1](中国科学院计算技术研究所,北京 100080) [2](浙江林学院信息系,杭州 311300)

摘 要：文章分析了目前常用的等价性检验方法的特点,包括功能性和结构性的验证方法;讨论了基于二叉判决图(BDD)的组合电路等价性检验方法,并分析了等价性检验过程中的误判问题及其消除方法;然后给出了一种关于带黑匣子的部分实现的隐式等价性检验方法,实验结果表明了该方法的有效性。

关键词：形式验证,等价性检验,符号模拟,二叉判决图,组合电路

1 引言

成功设计一个复杂数字系统,要求在设计的各个阶段验证实现的正确性。传统的设计验证是通过模拟来完成，然而随着电路复杂性的日益增加,模拟需要花费大量的 CPU 时间,在实践中实现穷举的模拟来保证设计的正确性几乎是不可能的。为了克服模拟的局限性,人们转而求助于各种形式验证方法,如模型检验、定理证明和等价性检验等。使用形式验证方法有可能保证设计在各种可能的输入组合下的正确性。等价性检验是目前在工业实践中最广泛使用的形式方法,而且已被应用于验证大型复杂的设计[4]。等价性检验的基本思想是使用形式方法，对照设计的规范来验证它的实现的功能正确性。等价性检验使验证工程师节省了大量的时间,因为它使用形式方法而不需要产生测试向量,此外这种方法对于检测设计错误也很有效。等价性检验可以用来验证两个相同或者不同抽象级别设计的等价性,如寄存器传输级 RTL(register transfer level)与 RTL 级,门级与门级,或 RTL 与门级之间的等价性,从而保证设计在各个阶段的正确性。由于等价性检验方法的优势，已应用于专用集成电路(ASIC)设计流程的不同阶段,通过验证每一阶段的中间结果,来保证在这一结果传输到下一阶段之前不出现错误是非常重要的。因为错误越早发现,以后的调试时间就越少。目前许多公司都将等价性检验方法集成到 EDA 工具中,如 Cadence 公司的形式验证工具 Heck、Synopsys 公司的 Formality、Mentor Graphics 公司的 FormalPro 等。

具体的数字系统设计往往包含组合电路与时序电路两部分,通常时序电路的等价性检验使用有限状态机遍历的方法来实现[16],本文主要讨论用于完成组合电路验证的等价性检验方法。组合电路的等价性检验是指:给定两个布尔网表,检验它们的相应输出是否对所有可能的输入都相同。尽管验证两个组合电路的等价性在逻辑设计的 CAD 中非常重要,然而这已被证明是一个 NP 完全问题。这表明要寻求一种完全地解决等价性检验的一般方法是很困难的。通常等价性检验方法的性能随着电路规模的增加而指数式地下降,因此大多数实用方法都是利用分而治之的思想,即通过某种策略将电路细分,然后增量地验证两个电路的等价性,从而能够处理较大规模的电路。

收稿日期：2002-08-16

基金项目：国家自然科学基金重点项目(90207002)

中科院计算所领域前沿青年基金项目(20026180-6)

北京市家重点科技项目(H020120120130)

2 基于 ROBDD 的等价性检验方法

当前常见的组合等价性检验方法大致可以归结为两大类:功能性的和结构性的验证方法,但这两者之间并没有太严格的界限。功能性方法是通过将电路表示成一种规范形式,两个电路当且仅当它们的规范形式同构时是等价的。最常用的规范形式是简化的有序二叉判决图 ROBDD (reduced ordered BDD)[14],为了验证两个电路的相应原始输出是等价的,必须按相同的输入变量序分别构造这两个输出函数的 ROBDD,然后判断它们是否同构。这种方法输入变量排序极为敏感,常遇到内存爆炸的问题。

结构性方法则是通过识别电路内部的等价结点,并利用这些等价结点的功能蕴涵来简化验证问题。内部等价结点常被称为断点(cutpoint),识别潜在断点的常用方法有随机模拟、ATPG 方法、基于 BDD 的方法以及基于可满足性的推理方法。所有潜在的断点找出后,依据一定的准则,将整个系统验证分解为关于这些断点子集的较小的验证任务,分别进行验证。

目前提出的等价性检验方法大多数[2,3,5,6,8~12]都是按结构性检验的基本框架,再结合功能性方法进行验证,如图 1 所示。首先提取出两个被比较电路的内部相似性,推导出各断点之间的特殊关系,如等

价性、蕴涵及可置换性等等。然后使用这些关系，把等价性检验过程分解为关于这些断点子集的较小的验证任务，分别完成。总的算法是从原始输入向原始输出方向遍历两个电路(或者 miter[9])，从已知的断点演绎出新的断点，直到所有的相应原始输出被证明等价，或者找到某个输入向量使得两个电路输出不匹配。本文主要讨论以 ROBDD 为演绎引擎的等价性检验方法。

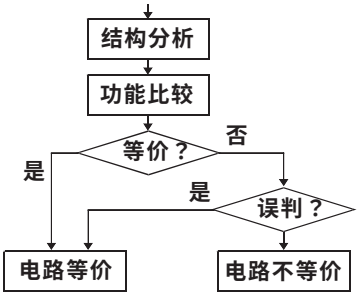

图 1　等价性检验的一般流程图

算法的具体描述：

第一步：先对两个电路作适量的、一致的随机模拟，比较各结点的输出响应，通常是采用特征分析法。具有相同特征的结点对作为潜在的断点，加入一个候选表 CEP-list (Candidate Equivalent Pair)中。接着对候选表中的断点从原始输入开始，按距离原始输入的远近，作广度优先排序。使用随机模拟的好处在于可以较早发现设计错误。在具体的设计过程中，如在综合、局部优化阶段，信号名称大多数保持不变或存在某种联系，因此有些商业工具也根据信号名称来识别潜在断点。

第二步：利用 ROBDD 按次序验证潜在断点的等价性。由于候选表中元素已排序，每次验证一对结点时，候选表中位于它前面的结点已知是等价的。如果当前结点对被证明是等价的，便加入到等价结点表(EQP-list)中。

第三步：检验所有的原始输出对是否位于等价结点表中，如果是，则两个电路等价；否则不等价。

为了验证一对候选结点的等价性，需要利用已知的断点来推导。前面已经介绍使用全局 BDD 表示布尔函数常碰到内存爆炸问题，如果能够找到某个合适的断点集，即结点的迁移扇入集(transitive fanins)TFI 的割集[1,3]，那么只需比较这对结点关于该割集的局部 BDD，即根据割集中的元素，而不是原始输入来表示 BDD，便能检验它们的等价性。

Y Matsunaga[3]建议使用启发式的方法导出合适的割集，它基于如下两条原则：

- 包含在候选结点迁移扇入集中较多个元素的 TFI 中的结点，即从该结点到候选结点有较多的通路，这样的结点应选入割集。
- 如果相关的结点同时选入割集中，可能发生误判。因此，相互独立的结点适合于构成割集。这里的相关是指结点之间存在迁移扇入或扇出的关系。

3　误判问题的处理

误判问题(false negative)是指在等价性检验过程中，把本来是等价的两个结点判断为不等价结点。如图 2 所示为一个误判的例子。F 和 G 是等价输出，$d1$ 和 $d2$ 是断点。然而如果依据粗线表示的割集来验证 F 和 G，将会导致误判。产生误判的主要原因是候选等价结点的割集中有相关的断点或者两个割集不一致。

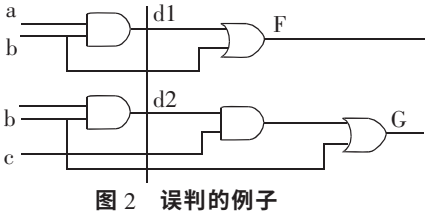

图 2　误判的例子

消除误判的方法大体有如下三类：

C L Berman and Louise H Trevillyan[2]及 van Eijk[16]基于置换变量的算法：即利用已知的断点或原始输入来置换 BDD 的变量，直到其变量中不再有断点。这类方法对于所处理的断点次序非常敏感，常常难以选择合适的断点数量。

Y Matsunaga[3]提出使用 BDD 进行功能蕴涵。假设知道两个结点集 S 和 T 之间的迁移关系，则可通过特征函数像计算的办法进行功能蕴涵，利用 BDD 的 compose 运算(Bryant[17])能够有效进行。这种方法往往由于 BDD 过大而导致算法失效。

S M Reddy 等人[11]提出的基于 ATPG 的方法，这类方法的局限性在于如果 ATPG 无解时，算法可能导致失败。另外，如果断点远离原始输入，并且两个电路没有其它蕴涵关系时，算法可能需要大量的 CPU 时间。

4　带黑匣子的部分实现的等价性检验方法

现代的大型设计中，设计的各模块通常由不同的小组或设计师分别进行设计，在设计的某一阶段，实现的某些模块可能是未知的。另外，越来越多的复杂设计适用采用基于 IP(Intellectual Property)核的设计，由于知识产权的保护，IP 的具体实现对于

用户来说是未知的。我们称设计中的这种未知部分为黑匣子。为了尽可能早地发现设计的错误，并进行设计调试，可以对带黑匣子的部分实现进行等价性检验[19]。本节我们给出一种隐式的验证方法。

仍然采用 BDD 作为基本的数据结构，由于特征函数具有集合运算的优势，特别适合用来表示部分功能未知的电路。对于一个布尔函数 f: $B^n \to B^m$，它的特征函数 χ_f: $B^{n+m} \to B$ 定义如下：

$$\chi_f(x,y)=1 \Leftrightarrow f(x)=y,\ \forall x \in B^n,\ y \in B^m$$

设布尔函数 f 所表示电路的原始输出分别为 $f_1, f_2, \cdots, f_m$，则 χ_f 可以通过下列方程求得：

$$\chi_f(x,y)=\prod_{j=1}^{m}\chi_{f_j}(x,y_j) \quad (1)$$

这里的 χ_{f_j} 可以从表示 f_j 的布尔函数直接导出。

为了使用符号模拟方法验证带黑匣子的部分实现，我们必须构造黑匣子外部区域的 BDD，这时要将黑匣子的输出当成整个电路的附加原始输入。设 Spec 表示设计的规范，Imp 表示带黑匣子的部分实现，G 表示 Imp 中黑匣子的外部区域，f_G 和 χ_G 是与 G 相应的布尔函数及特征函数，则可以如下计算 χ_G：

$$\chi_G(x,y,z)=1 \Leftrightarrow f_G(x,z)=y$$

这里 x,y,z 分别表示原始输入、原始输出以及黑匣子的输出。

命题 **1** 如果对于黑匣子的任何可能的实现，Imp 都与 Spec 不相同，那么 G 中一定存在错误。

换句话说，为了检测 Imp 中的错误，必须找到某一原始输入向量，使得它对黑匣子的任何可能实现，Imp 与 Spec 的输出不同。上述的充分条件可以形式化为如下的不等式：

$$\exists x,y: \forall z: \chi_{spec}(x,y) > \chi_G(x,y,z) \quad (2)$$

即，如果上面不等式成立，那么 G 中一定存在错误。但反过来，我们不能保证设计是正确的。实践中，为了使用 BDD 工具中的标准函数，常考察不等式(2)的否定形式：

$$\forall x,y: (\chi_{spec}(x,y) \leqslant (\exists z: \chi_G(x,y,z))) \quad (3)$$

若不等式(3)不成立，那么 G 中一定存在错误。

从上面的分析我们得到带黑匣子的部分实现的等价性检验方法：

第一步，构造 Spec 的 BDD，对于含有多个原始输出的情况，则将其表示为 SBDD(shared BDD)[11]，所有 BDD 的变量排序是相同的。应用方程(1)将 Spec 的 BDD 转化为特征函数。

第二步，将黑匣子的输出当作电路的附加的原始输入，类似第一步的方法，构造 G 的 BDD，然后转化为特征函数。

第三步，应用不等式(3)，检测 G 中是否存在错误。

实现上述算法的伪代码如下：

```
procedure BbVerify(Spec, Imp);
Begin
  {按方程(1)构造 Spec 的特征函数}
  BuildDDs(Spec);   {构造 Spec 的 BDD}
  Tmp=DD_One;
  For every PO_j of Spec do
  Begin
    χj=CharacterDD(Bddj);   {将各原始输出 POj 的 BDD
     转化为特征函数}
    Tmp=BddAnd(Tmp, χj);   {将各原始输出的特征函数
     作与运算}
  End;
  χspec=Tmp;
  {以下把黑匣子输出当作电路的附加原始输入，用类
   似上面的方法构造 G 的特征函数 χG，代码略}
  ……
  {以下检测不等式(3)是否成立}
  χsmoth=Existential_quatification(χG,z);   {从中将附加的原
   始输入“磨掉”，以便后面的比较}
  result=BddLeq(χspec, χsmooth);{计算不等式(3)的真值}
  return result ;
End
```

然而，由于符号模拟的局限性，以上的方法不一定能找出所有的错误，即不等式(3)成立时，实现中仍可能有错。文献[19]给出了使得上述算法完全的一个比较严格的条件。

5 实验结果及其分析

使用 CUDD 2.3.1[15]作为基本的 BDD 工具包，并用 C 语言实现了上述算法。实验是在 P4 PC (主频 1.4G，256M 内存)上，在 Mandrake Linux 8.1 环境下完成的。我们把基准电路作为设计规范，然后随机地指定一些黑匣子(通过标记某些电路中的门)。接着随机地在黑匣子的外部区域注入一个错误，对于 BLIF (Berkeley Logic Interchange Format) 格式的 Benchmark，我们的方法是通过修改门的真值表来改变其类型。也可用其它方法如 增加或减少扇入信号、反相器等。表 1 给出了实验结果，实验中除电路 i3 指定 4 个黑匣子外，其余各电路都指定 2 个黑匣子，针对同一电路做了多次实验，每次实验都注入不同的故障。第 4、5 列表示黑匣子所包含的门数

与规范的 BDD 大小(所含结点数);第 7 列表示在所有实验中错误被检测到的百分比;CPU 时间单位为秒,指的是所有实验中所需时间的最大者。

表 1　实验结果

电路名称	原始输入数	原始输出数	黑匣子大小	BDD 大小	实验次数	错误检测率	CPU 时间
alu4	14	8	51	4981	50	76%	0.25
mux	21	1	4	1072	30	70%	0.98
cc	21	20	8	2109	30	85%	1.05
C432	36	7	17	2605	50	80%	0.04
term1	34	10	15	4303	50	85%	0.06
count	35	16	8	39323	30	100%	3.42
comp	32	3	15	96613	50	92%	8.54
i1	25	16	8	17567	20	90%	0.32
i2	201	1	33	335	31	82%	0.02
i3	132	6	12	2052	50	95%	0.02
i4	192	6	8	89996	50	98%	0.75

实验结果表明,一般情况下,当黑匣子较小时,算法所能检测到错误的百分比较高。由于方法的不完全性,不能检测到所有的错误。但即使黑匣子较大,仍然可以发现大量的错误,这对于设计的早期调试非常有帮助。

6　结束语

等价性检验是指对照设计规范来验证设计实现的功能正确性,必须指出的是,等价性检验仅仅证明规范与实现是等价的,它并不能证明设计规范是正确的。本文分析了常用组合等价性检验的两类基本方法,讨论了以 ROBDD 为引擎的等价性检验方法以及验证过程中出现的误判问题。等价性检验过程中的误判问题主要是由于割集中存在相关的元素,因此选择割集时必须避免这样的元素出现。在具体算法实现中,检测误判的过程是伴随着内部等价性的演绎同时进行的,每当等价性检验程序发现两个候选结点不等价时,必须加以确认,以免发生误判。关于带黑匣子的部分实现的隐式验证方法,实验结果表明该方法的有效性和可行性。带黑匣子的实现验证仍然可以使用第 2 节所述的增量方法,今后的研究中我们将寻求完全的验证方法。

参考文献

[1] J Jain, A Narayan, M Fujita and A S Vincentelli. Formal Verification of Combinational Circuits. In Proceeding of International Conference on VLSI Design, 1997: 218~225.

[2] C Leonard Berman, Louise H Trevillyan. Functional Comparison of Logic Designs for VLSI Circuits. Proc. of IEEE Int'l Conf. on Computer-aided Design, 1989: 456~459.

[3] Yusuke Matsunaga. An Efficient Equivalence Checker for Combinational Circuits. In Proc. of ACM/IEEE Design Automation Conference(DAC), 1996: 629~634.

[4] Dr. Eng. Ka Lok Man. Efficient Equivalence Checking of Industrial Designs. In Proc. of IEEE Design Verification Conference in Europe, 2001: 1~10.

[5] A Kuehlmann, F Krohm. Equivalence Checking Using Cuts and Heaps. Proc. of DAC, 1997: 263~268.

[6] J R Burch, V Singhal. Tight Integration of Combinational Verification Methods. Proc. of ICCAD, 1998: 570~576.

[7] F Krohm, A Kuehlman A Mets. The Use of Random Simulation in Formal Verification. In Proc. of the Conf. Computer Design (ICCD), 1996: 371~376.

[8] S Malik, A R Wang, R K Brayton, A S Vincentelli. Logic Verification using Binary Decision Diagrams in a Logic Synthesis Environment. Proc. of ICCAD, 1988:6~9.

[9] D Brand. Verification of Large Synthesized Designs. Proc. ICCAD. 1993: 534~537.

[10] J Marques-Silva and T Glass. Combinational Equivalence Checking Using Satisfiability and Recursive Learning. In Proc. of IEEE/ACM Design, Automation and Test in Europe (DATE), 1999: 145~149.

[11] S M Reddy, W Kunz, D K Pradhan. Novel Verification Framework Combining Structural and OBDD Methods in a Synthesis Environment. Proc. of DAC, 1994: 414~419.

[12] C A J van Eijk. A BDD-based verification engine for combinational equivalence checking. In Proc. 8th ProRISC/IEEE-Benelux Workshop on Circuits, Systems and Signal Processing, Mierlo, 1997: 155~162.

[13] C A J van Eijk and G L J M Janssen. Exploiting structural similarities in a BDD-based verification method. In Proc. 2nd Int. Conf. On Theorem Provers in Circuit Design, 1997: 110~125.

[14] Randal E Bryant. Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers, August 1986, 38(8):677~691.

[15] F Somenzi. CUDD: CU Decision Diagram Package Release 2.3.1. University of Colorado Boulder. 2001.

[16] Shi-Yu Huang and Kwang-Ting(Tim) Cheng. Formal Equivalence Checking and Design Debugging. Kluwer Academic Publishers. Boston. 1998: 40~60.

[17]韩俊刚, 杜慧敏. 数字硬件的形式化验证. 北京: 北京大学出版社, 2001: 1~25.

[18] Thomas Kropf. Introduction to Formal Hardware Verification. Springer. New York. 1999: 31~81.

[19] W Gunther, N Drechsler, R Drechsler, and B Becker. Verification of designs containing black boxes. In EUROMICRO, 2000: 100~105.

[20] A Jain, V Boppana, R Mukherjee, J Jain, M Fujita and M

(下转第 55 页)

devices. In: Proceedings of IEEE 1993 VLSI Test Symposium, Atlantic: IEEE Computer Society Press, 1993: 4~9.

[2] S Chakravarty, V P Dabholkar. Two Techniques for Minimizing Power Dissipation in Scan Circuits During Test Application. In: Proceedings of 3th IEEE Asian Test Symposium, Beijing: IEEE Computer Society Press, 1994:324~329.

[3] P Girard, C Landrault, S Pravossoudovitch and D Severac. Reducing Power Consumption during Test Application by Test Vector Ordering. In: Proceedings of IEEE international Symposium on Circuits and Systems, San Francesco: IEEE Computer Society Press, 1998. CD-ROM version.

[4] S Wang and S K Gupta. ATPG for Heat Dissipation Minimization during Test Application. IEEE Trans. on Computers, 1998,47(2):256~262.

[5] Stefan GerstendÖrfer, Hans-Joachim Wunderlich. Minimized power consumption for scan-based BIST. In: Proceedings of IEEE International Test Conference, Atlantic: IEEE Computer Society Press, 1999:77~84.

[6] Radu Marculescu, Diana Marculescu, and Massoud Pedram. Sequence Compaction for Power Estimation: Theory and Practice. IEEE Transactions on CAD, 1999,18(7):973~993.

[7] Zhan-ping Chen, Kaushik Roy. A power macro-modeling technology based on power sensitivity. In: Proceedings of 35th ACM/IEEE Design Automation Conference, San Francisco: ACM Incorporation, 1998:678~683.

[8] 李晓维, 李华伟, 骆祖莹, 闵应骅. 降低时延测试功耗的有效方法.《计算机辅助设计与图形学学报》, 2002, 14(8): 738~742.

[9] 骆祖莹, 闵应骅, 杨士元. 降低 CMOS 电路测试功耗的方法比较.《第九届全国容错计算学术会议论文集》, 长沙: 2001.11:264~267.

LUO Zu-ying, HONG Xian-long (Department of Computer Science and Technology, TsingHua University, Beijing 100084)
Li Xiao-wei (Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

Abstract: In this paper, indefinite test bits´ influence on test power optimization has been researched through changing the number of indefinite bits in test patterns. Experiments on ISCAS85 and ISCAS89 benchmarks demonstrate the following influences not only for combinational circuits but also for sequence circuits. First, with the increase of indefinite test bits, the un-optimized test power markedly decreases. Second, with the increase of indefinite test bits, all optimization effects of three optimization approaches researched in this paper obviously increase. Third, the optimization effect of the Hamming distance approach increases the most among three optimization approaches. Forth, if the percent of indefinite bits is more than 90% of all bits in a test pattern, the Hamming distance approach can replace the other time-consuming approaches to directly optimize the test power for CMOS VLSI sequential circuits.

Key words: Test pattern, Test power, Power optimization, Hamming distance

骆祖莹　男,(1968-),**江苏连云港人,博士后。主要研究领域包括** IC **功耗估计、测试功耗优化、低功耗芯片设计。现主要从事** VLSI **芯片电源/地线网络性能评估与优化、多阈值多电压低功耗芯片设计、串扰屏蔽技术研究。**

李晓维　男,(1964-),**博士生导师。主要研究领域包括** IC **设计与测试、软件测试等。**

洪先龙　男,(1940-),**博士生导师。主要研究领域包括** IC **布图布线等。**

(上接第 51 页)

Hsiao. Testing, verification, and diagnosis in the presence of unknowns. In VLSI Test Symposium. 2000: 263~269.

[21] Christoph Scholl and Bernd Becker. Checking Equivalence for Partial Implementations. In Proc. of 38th DAC, 2001: 238~243.

LI Guang-hui[1,2], SHAO Ming[1], LI Xiao-wei[1]
[1](Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)
[2](Department of information,Zhejiang Forestry College, Hangzhou 311300)

Abstract: This paper analyses the characteristics of current equivalence checking methods, including the two categories: functional and structural. The combinational equivalence checking methods based on binary decision diagram (BDD) are discussed, the false negative problem during equivalence checking and its eliminating methods are analyzed. Then we present an implicit algorithm for verifying the partial implementation with black boxes, the experimental result demonstrates its effectiveness and feasibility.

Key words: Formal verification, Equivalence check, Symbolic simulation, BDD, Combinational circuits

李光辉　男,(1970-), **博士研究生。主要研究方向为** VLSI/SOC **测试、验证等。**

邵　明　男,(1976-), **博士研究生。主要研究方向为** VLSI/SOC **测试、验证等。**

李晓维　男,(1964-),**博士,研究员,博士生导师。主要研究领域为** VLSI/SOC **设计、测试、验证,软硬件协同设计等。**