验证包含黑盒的电路设计的有效方法

李光辉[1),2),3)] 邵 明[1),3)] 李晓维[1),3)]

[1)](中国科学院计算技术研究所 北京 100080)

[2)](浙江林学院信息工程学院 杭州 311300)

[3)](中国科学院研究生院 北京 100039)

摘 要 在超大规模集成电路设计中,为了进行早期的设计错误检测与调试或层次化验证,常常需要使用含黑盒的设计验证方法.该文提出了一种结合逻辑模拟和布尔可满足性的黑盒验证方法,用于验证设计中黑盒外部的功能正确性.该方法使用量化的合取范式(CNF)来表示电路中出现的未知约束,并且不需要修改电路结构,有效地节省了计算资源.此外,通过使用随机并行模拟增强了可满足性算法的错误检测能力.通过对 ISCAS'85 电路的实验表明了该方法不仅比以往同类算法速度快,而且具有较好的错误检测能力.

关键词 布尔可满足性;功能验证;黑盒;布尔比较;逻辑模拟

中图法分类号 TP306

An Efficient Method for Verifying Designs with Black Boxes

LI Guang-Hui[1),2),3)] SHAO Ming[1),3)] LI Xiao-Wei[1),3)]

[1)](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

[2)](*School of Information Engineering, Zhejiang Forestry College, Hangzhou* 311300)

[3)](*Graduate School of the Chinese Academy of Sciences, Beijing* 100039)

Abstract In the VLSI design cycle, in order to detect and debug the errors in early stages or to verify the functional correctness hierarchically, black boxing verification methods are often used. In this paper, an efficient method integrating logic simulation and Boolean satisfiability (SAT) is presented, which can verify the designs with black boxes. This method uses quantified conjunctive normal form (CNF) formulas to represent the unknown constraints in the circuit under verification, and needs no modification of the circuit structure so that it saves the computational resources significantly. In addition, this approach enhances the SAT-based algorithm with random parallel pattern simulation, which improves the capability of detecting errors. Experimental results on the ISCAS'85 benchmark circuits demonstrate that, the proposed approach is faster up to one to three orders of magnitude than those approaches in literature.

Keywords Boolean satisfiability; function verification; black box; Boolean comparison; logic simulation

1 引 言

随着超大规模集成电路设计功能复杂性的日益增加,在设计期间,常有某些模块功能未知.比如,一个设计的不同部分或模块可能由不同的设计者分阶段完成,相对于特定的设计者而言,在整个设计完成以前的某个阶段,某些模块的功能是未知的.此外,对

收稿日期:2003-05-13;修改稿收到日期:2004-03-10.本课题得到国家自然科学基金重点项目(90207002)、北京市重点科技项目基金(H020120120130)及浙江省自然科学基金(M603097)资助.**李光辉**,男,1970 年生,博士研究生,副教授,主要从事 VLSI/SoC 形式化设计验证方法研究.E-mail:zjfclgh@ict.ac.cn.**邵 明**,男,1976 年生,博士研究生,从事 VLSI/SoC 形式化设计验证方法研究.**李晓维**,男,1964 年生,博士,研究员,博士生导师,研究方向为 VLSI/SoC 低功耗易测试设计、VLSI/SoC 设计验证与测试、可信系统.

于大型复杂的设计,基于 IP(Intellectual Property)的设计方法越来越流行.由于知识产权的保护,来自第三方的 IP 模块其内部结构或功能往往是未知的.为了在早期阶段保证设计的功能正确性,经常需要验证这种包含未知模块的电路,这时可以把功能或内部结构未知的部分当成黑盒来处理.含黑盒的验证方法在层次化的设计验证中也经常用到,例如,我们可以首先验证规范与设计的相应子模块是功能一致的,然后把已经验证过的模块设置为黑盒,再来验证顶层模块的功能正确性,从而提高验证工具的效率和处理能力.另外,这种方法可以提取出设计中的某些难验证的部分,如存储器模块,使得验证能够顺利完成;黑盒验证方法还能提高设计错误诊断的准确性,非常有助于设计的调试.

近年来,不少文献[1~5]提出了验证包含黑盒的电路设计的方法.第一类方法[2~5]使用二叉判决图(BDD)作为基本的数据结构,并依赖于符号模拟[6].这类方法通过把黑盒输出当成电路的新原始输入,来计算实现的特征函数,从而可以针对黑盒作功能蕴涵.文献[5]提出了几种准确性与计算资源折衷的算法.对于包含黑盒的设计来说,计算特征函数可以提取出尽可能多的功能,但由于使用 BDD 表示特征函数时,其支撑变量不仅包含原始输入,还包含原始输出.因此这类方法的空间复杂性很高,有可能导致内存爆炸,故不适于验证较大的电路.文献[1]中提出了两种验证包含黑盒的电路设计的方法:(1)解决含有未知约束的布尔可满足性问题;(2)基于修改电路结构的方法.这类方法使用两位的 0,1,X 编码来表示受未知约束影响的电路结点,导致电路信号的复制[5],因此时间复杂度较高,但能够处理较大的电路.

本文提出了一种结合逻辑模拟和 SAT 的黑盒验证方法,不需要修改电路结构,也不会导致电路信号的复制.我们把黑盒的输出当成实现电路的新原始输入,在构造好 miter[7,8] 电路的 CNF 公式之后,把表示新原始输入的变量从 CNF 公式中量化出去,然后解决最后所得的 SAT 问题.同时通过使用随机并行模拟进一步增强了可满足性算法的错误检测能力.比较前述的两类方法,我们的新方法节省了更多的计算资源.通过对 ISCAS'85 电路的实验结果表明,对大多数电路,新的方法比文献[1,2]中的方法快 1～3 个数量级.

2　电路的关系表示和可满足性问题

组合电路的行为可以直接用布尔函数来表示,原始输入相应于函数的变量,原始输出则相应于函数值.此外,还可以使用关系表示[9~11],这时电路的输出使用变量显式地表示出来.例如,表达式 $y \leftrightarrow x_1 \wedge x_2$ 是一个与门的关系表示,只有在包括输入和输出在内的所有变量的赋值是许可值时,表达式的值才为真,即它表示电路结点的所有可能的赋值集合.电路关系表示的一个好处是不必要计算整个电路的布尔函数,整个电路的关系表示就是所有门的关系表示的合取.

定义 1.　布尔可满足性问题(SAT)是指:给定一个布尔函数 $F(\boldsymbol{X}):\{0,1\}^n \rightarrow \{0,1\}$,找出一组变量赋值 X^*,使得 $F(X^*)=1$ 或证明不存在这样的赋值.

SAT 算法通常将布尔函数表示成 CNF 公式,一个 CNF 公式是一个由子句构成的集合,集合中的每个子句是由一些变量或它们的否定(称为文字)构成的析取.一个 CNF 公式称为可满足的,如果至少有一组变量的赋值使得公式的值为真.逻辑电路的 CNF 公式是每个门的 CNF 公式的合取,每个门的 CNF 公式实质上是它的关系表示,图 1 给出了基本门的 CNF 公式[12].

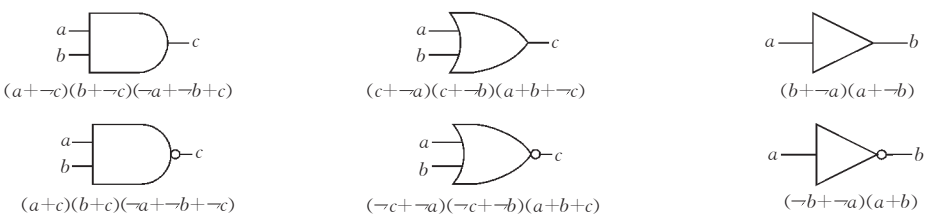

图 1　基本逻辑门的 CNF 公式

定义 2.　给定布尔函数 $f:B^n \rightarrow B^m$,$B=\{0,1\}$,它的特征函数 $\chi_f:B^{n+m} \rightarrow B$,定义如下:

$$\chi_f(x,y) \leftrightarrow f(x)=y, \quad \forall x \in B^n, y \in B^m.$$

显然,电路的关系表示正好就是一个表示许可赋值的特征函数.

定义 3.　布尔函数 $f(x_1,x_2,\cdots,x_n)$ 关于任意变量 x_i 的全称量化(universal quantification)是指

$$\forall_{x_i} f = f|_{x_i=1} \cdot f|_{x_i=0}, \quad i=1,2,\cdots,n.$$

定理 1. 给定布尔函数 $f(x_1,x_2,\cdots,x_n)$的 CNF 公式 $F(x_1,x_2,\cdots,x_n)$,F 关于任意变量 x_i的全称量化等价于从该 CNF 公式中直接删除所有文字 x_i或$\neg x_i$.

证明. 首先将 $F(x_1,x_2,\cdots,x_n)$中的 CNF 子句按 x_i及$\neg x_i$的分布情况分为三组,即 $F=F_1\cdot F_2\cdot F_3$.

这里,$F_1=f_1(x_1,x_2,\cdots,x_{i-1},x_{i+1},\cdots,x_n)$表示 x_i与$\neg x_i$在每个子句中都没有出现过的子句组;$F_2=f_2(x_1,x_2,\cdots,x_i,\cdots,x_n)$表示当且仅当 x_i出现在每个子句中的子句组;$F_3=f_3(x_1,x_2,\cdots,\neg x_i,\cdots,x_n)$表示当且仅当$\neg x_i$出现每个子句中的子句组.

因为 $f_2|_{x_i=1}=1, f_3|_{x_i=0}=1$,

$$\begin{aligned}\forall_{x_i} f &= f|_{x_i=1}\cdot f|_{x_i=0}\\ &=(f_1\cdot f_2\cdot f_3)|_{x_i=1}\cdot(f_1\cdot f_2\cdot f_3)|_{x_i=0}\\ &=(f_1\cdot f_3|_{x_i=1})\cdot(f_1\cdot f_2|_{x_i=0})\\ &=f_1\cdot f_3|_{x_i=1}\cdot f_2|_{x_i=0}.\end{aligned}$$

显然,$f_3|_{x_i=1}$相当于从 f_3中删除各文字$\neg x_i$,$f_2|_{x_i=0}$相当于从 f_2中删除各文字 x_i. 因此,定理得证. 证毕.

给定两个布尔函数 $F(\boldsymbol{X}):\{0,1\}^n\to\{0,1\}$及 $G(X):\{0,1\}^n\to\{0,1\}$,这里 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$表示原始输入向量. 为了比较 F 和 G 的功能等价性,我们只须证明布尔函数 $F(\boldsymbol{X})\oplus G(\boldsymbol{X})$是不可满足的. Brand[8] 引入了 miter 的概念,如图 2 所示. 它是通过把两个电路相对应的每一对原始输入联接到一起,同时把相对应的每一对原始输出联接到一个异或门得到的. 为了使用 SAT 来求解,有时也将各异或门联接到一个或门.

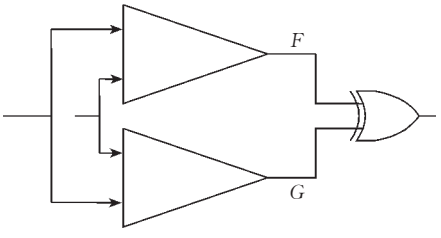

图 2 比较等价性的 miter 模型

显而易见, miter 模型本质上就是布尔函数 $F(\boldsymbol{X})\oplus G(\boldsymbol{X})$的表示. 如果要证明两个电路的某对原始输出等价,我们可以证明与之对应的 miter 输出函数是不可满足的. 通常用来解决上述问题的方法有 BDD、自动测试生成(ATPG)或 SAT 以及模拟等.

3 结合逻辑模拟和 SAT 的黑盒验证方法

定理 2. 设 $f_{\text{spec}}:B^n\to B^m$是一个完整规范的布尔函数,布尔函数 $f_{\text{impl}}:B^n\to B^m$表示与规范相应的含有黑盒的实现设计. 如果存在 $x\in B^n, y\in B^m$满足 $f_{\text{spec}}(x)=y$,使得对黑盒的所有可能实现都有 $f_{\text{impl}}(x)\neq y$,那么在实现电路的黑盒外部区域必然存在设计错误.

定理 2 的证明是显然的. 下面给出验证带黑盒设计的 SAT 算法. 从本质上来看,解决规范与实现的 miter 电路的可满足性问题等价于比较规范与实现的特征函数. 我们把黑盒的输出当作实现电路的新原始输入,然后构造 miter 电路的 CNF 公式 ϕ. 由定理 2 知,为了检测实现中的错误,必须找到 ϕ 的一个独立于黑盒所有可能实现的满足赋值,即要求所有的黑盒输出可以为 0 和 1 之中的任意一个. 因此应该把表示黑盒输出的所有变量从 ϕ 中全称量化出去,如果最后所得到的公式是可满足的,那么就检测到了含黑盒设计中的一个错误,即实现中黑盒的外部与规范功能不一致. 设 spec 表示设计规范,impl 表示含有黑盒的实现设计,数组 Z 表示所有黑盒的输出. 算法描述如下:

```
Verify_SAT(spec,impl,Z)
{
    φ=Make_CNF(spec);
    Append_CNF(φ,TFO(Z));
    Miter_CNF(φ,spec,impl);
    Quantify(φ,Z);
    result=SAT_Solve(φ);
    return(result);
}
```

在上述算法中,第 1 步构造规范的 CNF 公式. 第 2 步把所有的黑盒输出 Z 当作实现的新原始输入,然后构造实现的 CNF 公式,这时只须考虑受 Z 影响的电路结点集合 $TFO(Z)$(transitive fanout sets). 第 3 步构造 miter 电路的 CNF 公式,只须构造规范与实现中相应原始输出的异或门,并联接到一个或门. 第 4 步将所有黑盒输出变量 Z 从 ϕ 中量化出去,由定理 1 知道,我们只须把 CNF 公式中的变量 Z(表示黑盒输出的所有变量)直接删除即可. 最后使用 SAT 程序求解 CNF 公式 ϕ.

然而,当 CNF 公式不可满足时,并不能保证含黑盒的设计是功能正确的,因为我们使用 X 值来表示

未知约束,有可能屏蔽某些错误.考虑下面的例子.

例 1.　考虑图 3 的 miter 结构,这里的规范是 C17 电路,实现是通过把门 h 置入一个黑盒得到的.我们用 z 表示实现的新原始输入,即黑盒输出(与规范中的 h 相对应),然后构造 miter 电路进行布尔比较.我们在实现中注入一个错误:将黑盒外的与非门 e 改变为或门 e'.容易看到,向量$\{a=0,b=0,c=0,d=0,f=0\}$是一个独立于黑盒实现的 CNF 公式的满足赋值.事实上,在这个向量作用下,无论黑盒输出 z 为 0 还是 1,异或门 x 输出都为 1,故这个错误被我们的算法检测到了.如果改为将门 i 置入黑盒,这时的 miter 电路 CNF 公式将是不可满足的,即检测不到错误,主要是受到中间结点 h 的影响而引起子句冲突.如果将门 i 和门 h 同时置入黑盒,所得到的 CNF 公式仍是不可满足的,这时主要是因为门 k 的两个输入 h 和 i 全部被置为 X 值,将它们从 CNF 公式中量化出去后,会引起子句冲突.

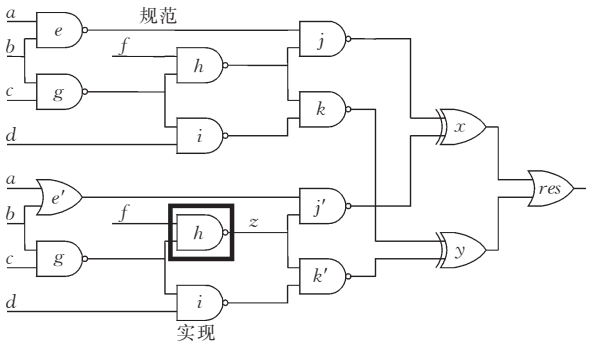

图 3　含黑盒的布尔比较的 miter 模型

例 1 中的后两种情况,本来向量$\{a=0,b=0,c=0,d=0,f=0\}$应该是可以检测到错误的,但由于受到全称量化有关变量的影响,SAT 算法反而不能求得任何解.

为了增强上述算法检测错误的能力,在使用 SAT 算法之前,我们先使用适量的并行 0,1,X 随机模拟.模拟的基本思想是:首先用随机产生的向量模拟完全的规范;然后使用相同的向量模拟不完全的实现,这时要把所有的黑盒输出都置为 X 值,使得实现中原始输出的响应与黑盒的具体实现无关;最后比较规范和实现的输出响应.由定理 1 知,如果某一对原始输出的响应分别为 0(1)和 1(0),说明错误被逻辑模拟检测出来了.在图 4 的示例中,因为在向量(010)作用下,含黑盒的实现与规范中的第一个输出分别为 0 和 1,所以黑盒的外部一定存在错误.但要注意,差异 $1/X$ 或 $0/X$ 不能说明黑盒外部肯定存在错误.显然,模拟的向量越多,检测到错误的可能性越大.

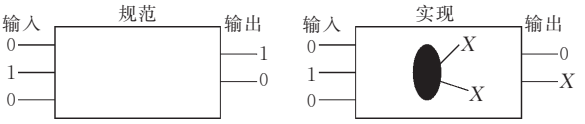

图 4　0,1,X 模拟示意图

为便于算法的实现,我们用两位的 0,1 编码来表示每个信号值[13].表 1 列出了编码方案;表 2 给出了基本门的求值运算法则,这里 $W1$ 和 $W2$ 分别表示门的输入值,$A1$ 和 $B1$ 分别表示 $W1$ 和 $W2$ 的第 1 个编码;$A2$ 和 $B2$ 分别表示 $W1$ 和 $W2$ 的第 2 个编码,符号 & 和 ∨ 分别表示按位与以及按位或运算.表 2 中对应与、或、非运算的第 2,3 列分别表示运算结果的第 1 位与第 2 位编码,这里的非运算仅以 $W1$ 的非运算为例,其结果是通过交换它的两位编码得到的.为了提高算法的效率,我们使用并行的逻辑模拟,这时用两个 32 位的机器字来表示每个电路结点值,其中每个字的对应两位分别表示:结点相应于某个向量响应值的两个编码.

表 1　0,1,X 编码方案

$W1$	$W2$	W
0	1	0
1	0	1
1	1	X
0	0	未用

表 2　并行逻辑模拟的逻辑运算

门	$W1$	$W2$
与	$A1\&B1$	$A2 \vee B2$
或	$A1 \vee B1$	$A2\&B2$
非	$A2$	$A1$

如前所述,设 spec 表示完整的设计规范,impl 表示含有黑盒的实现,数组 Z 表示所有黑盒的输出,那么总的黑盒验证算法描述如下:

```
BB_Verify(spec,impl,Z)
{
  result=TRUE;
  Ran_XSimulate(spec,impl,Z,n);
        //并行逻辑模拟,n 表示模拟的随机向量个数
  for each primary output
  {
    result=Match(X-response,Correct response);
        //比较规范与实现的响应
    if(result==FALSE)
      return(result);
  }
```

```
    result=Verify_SAT(spec,impl,Z);
    return(result);
}
```

在我们的算法中,函数 *Ran_Xsimulate* 用于带 X 值的逻辑模拟,首先对规范电路并行模拟 n 个随机向量,然后使用相同的 n 个向量并行模拟含黑盒的实现电路.比较两者的响应字时,只有当某个向量的响应在任意一个输出分别是 0(1)和 1(0),才表明检测到了设计错误,这时算法结束,不必再调用 SAT 验证算法 Verify_SAT.当返回值为 TRUE 时,表示算法没有检测到错误;而当返回值为 FALSE 时,表示算法检测到了设计错误.

4 实验结果

为了评估算法的性能,我们使用 Zchaff[14] 程序包作为可满足性算法引擎,并用 C++语言实现了上述算法.实验是在 Pentium4 1.4G PC 上(256M 内存),Mandrake Linux 8.1 环境下完成的.针对 ISCAS'85 基准电路,我们完成了两组实验,首先进行了含有未知约束的布尔比较实验;其次是完成了含有黑盒情况下的设计错误检测实验.

4.1 含有未知约束的布尔比较实验

在这个实验里,我们仅使用 SAT 验证算法,而不必进行逻辑模拟.类似文献[1],每个 ISCAS'85 电路作为设计规范,通过给相同的基准电路逐渐增加未知约束得到所要的实现电路,未知约束数从 0 增加到 4.为了表明算法的强健性,对每种情况,我们随机地改变约束结点,重复 100 次实验.表 3 和表 4给出了布尔比较所占用的 CPU 时间(s)及最大内存(MB),第 2 行中的 0,1,…,4 分别表示未知约束的个数.在表 3(或表 4)中,第 2 至第 6 列给出了在不同约束条件下,使用本文算法完成 100 次实验的最大时间(或最大内存),第 7～11 列给出了在不同约束条件下,文献[1]中算法(实验平台:Sun Ultra360,512MB 内存)的运行时间(或最大内存),文献[1]没有给出 C3540 电路的相关信息.由于文献[1]中的算法使用两位编码来表示受未知约束影响的每个电路结点,并且需要修改电路结构,因此表 3 中该算法的时间包含电路结构修改和布尔比较两部分时间.从表 3 和表 4 可以看出,本文的算法完成各基准电路的布尔比较远比文献[1]中的算法快,对绝大部分电路,我们的算法比后者要快 1～3 个数量级,同时所占用的内存也比后者要少.

表 3 布尔比较的 CPU 时间

电路	本文的算法					文献[1]的算法				
	0	1	2	3	4	0	1	2	3	4
C432	0.01	0.02	0.03	0.07	0.06	0.93	52.54	69.30	68.50	64.80
C499	0.02	0.05	0.05	0.05	0.06	2.25	49.70	193.40	275.10	261.70
C880	0.04	0.26	0.25	0.21	0.36	0.40	0.52	0.58	0.53	5.34
C1355	0.05	0.11	0.35	0.29	0.24	3.88	5.43	12.70	11.64	9.14
C1908	0.08	0.45	0.31	0.26	0.19	1.21	1.55	1.83	2.42	1.53
C2670	0.10	0.31	0.19	0.20	0.23	0.87	1.52	1.46	1.42	1.44
C3540	0.23	3.23	20.00	18.50	3.79	—	—	—	—	—
C5315	0.32	0.70	0.81	0.34	0.34	1.62	2.73	2.94	2.91	3.31
C6288	0.22	0.38	0.44	0.46	0.43	0.64	2.13	3513	2115	2615
C7552	0.15	0.59	0.61	0.61	0.54	3.60	13.41	6.76	5.30	5.60

表 4 布尔比较占用的内存

电路	本文的算法					文献[1]的算法				
	0	1	2	3	4	0	1	2	3	4
C432	0.72	1.20	1.24	1.28	1.38	5.15	15.53	17.33	17.29	18.13
C499	0.76	1.18	1.22	1.26	1.56	3.53	6.65	15.60	15.53	16.24
C880	0.82	1.38	2.40	1.54	1.68	3.12	3.28	3.36	3.37	5.37
C1355	0.88	4.60	3.10	3.10	4.18	3.89	5.35	7.21	6.56	6.66
C1908	1.04	4.82	5.00	4.64	4.10	3.87	6.42	4.64	4.69	5.94
C2670	2.20	3.56	5.42	4.52	5.36	4.92	5.33	5.36	5.39	5.45
C3540	1.80	8.80	10.70	6.94	7.52	—	—	—	—	—
C5315	6.78	9.70	7.96	8.40	8.98	6.62	7.34	7.37	7.41	7.45
C6288	1.78	9.06	10.60	11.90	11.50	5.93	6.16	20.29	19.34	21.31
C7552	5.78	14.10	17.80	14.50	12.50	8.70	23.65	23.95	9.87	9.90

4.2　含有黑盒的设计验证实验

这个实验主要用来评估本文的算法检测设计错误的能力.我们把基准电路当作设计规范,在相同的电路中随机选择若干黑盒,得到相应的实现电路.对每个电路随机地注入一个错误,设计错误是通过改变门的类型获得的,并重复 50 或 100 次实验.每次使用 SAT 算法之前,都使用 32 个随机向量并行逻辑模拟,但实际上,有时只要少数几个向量就可以检测到错误.表 5 给出了在只含一个黑盒情况下,本文算法与文献[2]中算法(实验平台:SUN Ultra2,256MB 内存)的实验结果比较.在大多数实验中,我们选择的黑盒比文献[2]中的黑盒要大,即黑盒中所含的门数多.表中第 2 列表示实验次数,第 3,6 列分别给出两种算法实验中的黑盒的大小,第 4,7 列给出两种算法检测错误的百分比,第 5,8 列分别表示两种算法完成所有实验所花费的最大时间(s).由于文献[2]中的算法使用 BDD 作为基本的数据结构,并且需要计算规范与实现的特征函数,计算复杂度远高于本文的算法,对于较大的电路,可能导致内存爆炸.因而,该算法没有能够解决 C880 等 7 个 ISCAS'85 电路.从表 5 可以看出,我们的算法检测到错误的百分比与文献[2]中的算法相当,然而却比后者要快 1～3 个数量级.当然,由于实验环境有所不同,直接的数据比较不够准确,而只是从一个侧面反映了本文算法在时间复杂度方面的优势.此外,因为 BDD 算法内在的局限性,本文算法占用的内存比后者也要少得多,且能够处理更大的电路.

表 5　含有黑盒的设计验证实验结果

电路	实验次数	本文的算法			文献[2]的算法		
		|BB|	检错率	时间(s)	|BB|	检错率	时间(s)
alu4	100	96	0.67	0.23	100	0.34	14.65
apex7	100	12	0.87	0.04	10	0.90	38.91
comp	100	17	0.89	0.03	10	0.73	0.31
C432	100	94	0.69	0.07	100	0.54	2.27
C499	50	48	0.80	0.16	50	0.82	524.66
C1355	50	85	0.76	0.21	50	0.74	1146.88
dalu	100	33	0.94	0.67	10	0.92	15.19
term1	100	55	0.79	0.07	50	0.81	1.97
C880	50	14	0.78	0.15	—	—	—
C1908	50	87	0.75	0.23	—	—	—
C2670	50	25	0.78	0.25	—	—	—
C3540	50	112	0.84	0.68	—	—	—
C5315	50	41	0.85	0.67	—	—	—
C6288	50	25	0.72	10.25	—	—	—
C7552	50	55	0.76	1.15	—	—	—

5　结束语

随着集成电路设计规模的日益增大,基于 IP 的设计方法越来越流行.在早期的设计错误检测与调试以及层次化设计验证时,常使用黑盒验证方法.本文提出了一种结合逻辑模拟和布尔可满足性的含黑盒设计验证方法,该方法不需计算特征函数和修改电路结构,首先通过逻辑模拟检测可能存在的设计错误,然后使用基于 SAT 的验证算法增强模拟算法.该算法在构造 miter 电路 CNF 公式之后,将表示黑盒输出的变量从公式中全称量化出去再求解.实验结果表明,与以往的算法相比,我们的算法降低了计算复杂度,占用的内存较少.本文的算法检测设计错误的能力与以往的算法相当,但远比后者的运算速度快.然而,与文献[1,2,4]中的算法类似,本文的算法由于使用 X 值来表示未知约束,也是一种不完全的算法,今后我们将进一步研究完全的黑盒验证方法.

致　谢　感谢美国普林斯顿大学 Lintao Zhang 博士为本文的实验提供了最新的 SAT 程序包 Zchaff.

参　考　文　献

1　Jain A., Boppana V., Mukherjee R., Jain J., Fujita M., Hsiao M.. Testing, verification, and diagnosis in the presence of unknowns. In: Proceedings of the 18th VLSI Test Symposium, Montreal, Canada, 2000, 263～269

2　Gunther Wolfgang, Drechsler Nicole, Drechsler Rolf, Becker Bernd. Verification of designs containing black boxes. In: Proceedings of IEE EUROMICRO Conference, Maastricht, 2000, 100～105

3 Liu Tai-Hung, Aziz Adnan, Singhal Vigyan. Optimizing designs containing black boxes. In: Proceedings of Design Automation Conference, Anaheim, California, 1997, 113～136
4 Scholl Christoph, Becker Bernd. Checking equivalence for partial implementations. In: Proceedings of Design Automation Conference, Las Vegas, 2001, 238～243
5 Scholl Christoph, Becker Bernd. Checking equivalence for circuits containing incompletely specified boxes. In: Proceedings of IEEE International Conference on Computer Design, Freiburg, Germany, 2002, 56～63
6 Bryant R.. Symbolic Boolean manipulation with ordered binary decision diagrams. ACM Computing Surveys, 1992, 24(3): 293～318
7 Huang Shi-Yu, Cheng Kwang-Ting(Tim). Formal Equivalence Checking and Design Debugging. Boston: Kluwer Academic Publishers, 1998
8 Brand D.. Verification of large synthesized designs. In: Proceedings of International Conference of Computer-Aided Design, San Jose, California, 1993, 534～537
9 Kropf Thomas. Introduction to Formal Hardware Verification. New York: Springer, 1999
10 Hasson Soha, Sasao Tsutomu, Brayton R. K.. Logic Synthesis and Verification. Boston: Kluwer Academic Publishers, 2002
11 Han Jun-Gang, Du Hui-Min. Formal Digital Hardware Verification. Beijing: Peking University Press, 2001(in Chinese)
(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)
12 Larrabee Tracy. Test pattern generation using Boolean satisfiability. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1992, 11(1): 4～15
13 Aarna M., Raik J., Ubar R.. Parallel fault simulation of digital circuits. In: Proceedings of the 42th International Scientific Conference of Riga Technical University, Riga, 2001, 91～94
14 Moskewicz M., Madigan C., Zhao Y., Zhang L., Malik S.. Chaff: Engineering an efficient SAT solver. In: Proceedings of Design Automation Conference, Las Vegas, 2001, 530～535

LI Guang-Hui, born in 1970, Ph. D. candidate, associate professor. His research interests include VLSI/SOC formal verification, design error diagnosis, automatic test pattern generation. At present, his research focuses on equivalence checking, design error diagnosis.

SHAO Ming, born in 1976, Ph. D. candidate. His research interests mainly focus on VLSI/SOC formal verification.

LI Xiao-Wei, born in 1964, Ph. D., professor. His current research interests include VLSI/SoC design for testability, low power design, design verification and low cost test, dependable computing.

Background

The authors are members of test and dependable computing group of Institute of Computing Technology, Chinese Academy of Sciences. The major research interests of this group are VLSI/SOC design for testability, VLSI/SOC verification, low power design, dependable system, software testing and network processor design, which is leaded by professor LI Xiao-Wei. This paper is primarily supported by the National Natural Science Foundation of China (NSFC) under grant No. 90207002, which involves in the research works about VLSI design verification and test generation from behavioral level to layout level. This paper presents some promising results on formal verification.