

基于 SAT 的电路错误定位方法研究进展*

张建民, 黎铁军, 张峻, 李思昆

(国防科技大学 计算机学院, 湖南长沙 410073)

摘要: 随着 VLSI 芯片复杂度不断增加, 功能验证与调试已占到整个芯片设计周期的 60% 以上。而错误的定位往往消耗大量的时间与精力, 因此迫切需要一种高效的方法诊断与定位电路中的错误。针对近年来出现的许多电路错误定位方法, 介绍了电路错误诊断方法的分类与工作流程, 深入分析了基于 SAT 的错误定位方法的基本原理; 对各种算法进行了概述评论, 并简要介绍了在不可满足子式求解方面所做的一些研究工作, 而不可满足子式能够显著提高错误定位效率与精度; 讨论了电路错误定位技术所面临的主要挑战, 并对今后的研究方向进行了展望。

关键词: 形式化验证; 错误定位; 布尔可满足性; 可满足性模

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-2486(2014)02-0081-06

Research advances in SAT-based error localization methods on circuits

ZHANG Jianmin, LI Tiejun, ZHANG Jun, LI Sikun

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: With the growing complexity of VLSI designs, functional verification and debugging has become a resource-intensive bottleneck in modern CAD flows, consuming as much as 60% of the total design cycle. Error localization in circuits is difficult and time-consuming. Therefore an efficient error debugging and localization method is necessary for hardware design. Recently there are many different contributions to research on error localization in circuits. Firstly, the categories and workflow of error debugging method were introduced. The fundamental principles of SAT-based error localization method were described. Then the existing algorithms were introduced and analyzed. Furthermore, the research results about extract unsatisfiable subformulae, which can strongly improve the efficiency and accuracy of error localization, were presented. Finally, the current challenges were discussed, and the future research directions of error localization in circuits were outlined.

Key words: formal verification; error localization; boolean satisfiability; satisfiability modulo theories

随着超深亚微米与超大规模集成电路的设计技术的飞速发展, 工艺的特征尺寸越来越小, 芯片的规模越来越大, 复杂度越来越高, 设计周期也越来越长, 其中大量的时间消耗在芯片的功能验证上。所谓功能验证是指无论是 RTL 级还是门级电路都要满足系统规范, 其目的是发现芯片逻辑设计中的错误。根据文献 [1], 功能验证已经占到整个芯片设计周期的 60% ~ 70%。模拟验证与硬件仿真验证是目前最常采用的功能验证方法。模拟验证方法虽然易于查错, 但是具有不完备性, 无法覆盖所有的边界条件, 而且当设计超过数百万门级甚至到数千万门级时, 模拟运行时间开销很大。而硬件仿真验证运行速度快, 通常能够达到较高的覆盖率, 但是仍然无法达到 100%

覆盖率, 并且错误诊断与定位非常困难。

除了功能验证, 后硅片阶段的芯片测试也受到电路规模增加的困扰。随着 VLSI 芯片的规模越来越大, 例如从 Intel 8088 微处理器的约 29 000 个晶体管到四核 Xeon sandy bridge 处理器的约 995 000 000 个晶体管, 从而导致自动测试向量生成 (Auto Test Pattern Generation, ATPG) 工具产生的测试向量的规模也飞速增加, 成指数级增长。当芯片出现 stuck-at 0 或 1 错误时, 如何根据海量的测试结果, 从逻辑网表中自动、迅速地追踪出现问题的单元, 成为硬件测试工程师非常棘手的问题。

自从 20 世纪 90 年代以来, 形式化方法作为一种重要的验证手段, 由于它具有完备性, 能够达

* 收稿日期: 2013-06-18

基金项目: 国家自然科学基金资助项目 (61103083, 61133007); 国家“863” 高技术研究发展计划资助项目 (2012AA01A301)

作者简介: 张建民 (1979—), 男, 山西平遥人, 助理研究员, 博士, E-mail: jmzhang@nudt.edu.cn

到 100% 的覆盖率,并且不需要外部激励,因此迅速成为学术界和工业界研究与应用的热点。然而,尽管形式化验证方法能够完备且自动化地检测是否出现错误,但是错误的诊断与定位仍然耗费较多的人工劳动,需要设计工程师去阅读与分析结果,效率低下且容易引出新的错误,并且从推理角度来看,错误定位的运算复杂性达到 NP 完全^[2]。所以,利用计算机的强大处理能力辅助人们进行错误诊断与定位的自动化方法,成为近年来 VLSI 验证领域的研究热点与难点问题。

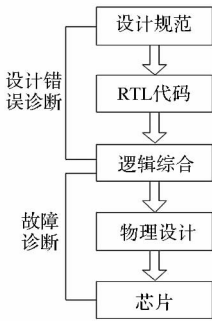


图 1 电路错误诊断类型

Fig. 1 Categories of circuit error debugging

基于布尔可满足 (Boolean Satisfiability, SAT) 求解的错误诊断与定位算法最初是在 IEEE Transaction on CAD^[3] 上系统阐述的,迅速成为 VLSI 验证领域一个新兴的研究热点。业界将芯片中的错误诊断分为两类:设计错误诊断 (design error diagnosis) 和故障诊断 (fault diagnosis)。图 1 给出了 VLSI 芯片设计的基本流程以及两种错误诊断类型所参与的设计步骤。设计错误诊断是指在设计初期芯片以 RTL 硬件描述语言或逻辑网表的形式存在,根据设计规范来确定代码或网表中存在的问题。而故障诊断是指实际芯片在测试阶段出现问题,则将错误注入网表,直到能够仿真出实际芯片的错误行为,从而确定问题所在。由于本文所讨论的错误诊断方法都适用于这两种错误诊断类型,因此在文中不再区分这两种说法。

在应用背景及需求方面,随着超深亚微米集成电路芯片的规模越来越大,功能越来越复杂,验证与调试成为整个芯片设计流程中最耗时的阶段;而如何准确定位错误通常要消耗电路设计与验证工程师们大量的时间与精力,往往成为延迟芯片面市时间的主要因素,因此业界迫切需要一种高效的自动化工具来帮助诊断与定位设计中的错误。近年来,基于 SAT 的电路错误自动诊断与定位技术的研究飞速发展,涌现了很多研究成果,逐渐成为硬件验证领域的研究热点,但是该研究

领域仍旧很开放,非常活跃。

基于 SAT 的错误定位技术是基于模型的诊断方法的一种,其主要特点是将代码中的错误备选部分进行替换,从而使得修改后的电路满足正确的行为或产生正确的结果。而基于解释的错误诊断技术也是一种经典的方法,它是通过比较错误与正确的两种执行踪迹来解释产生错误的原因。两种方法各有优缺点,前者基于 SAT 的错误定位技术的优点是能够精确地找到出错的地方,并且易于给出修正错误的参考;但其缺点是设计中错误备选的数目可能很大;而后者基于解释的错误诊断技术能够给出一个执行踪迹为何最终会导致错误结果的解释;但其缺点是定位错误的精度较低。

1 错误定位方法的分类及流程

在硬件设计中,错误定位的解空间随着代码行以及故障数的增加成指数级增长: $\text{search space} = (\#\text{lines})^{\#\text{errors}}$, 其中 $\#\text{lines}$ 表示代码行数, $\#\text{errors}$ 表示错误的数量。由于错误定位的复杂度较高,因此该领域仍是一个十分开放的课题。通常来讲,错误定位技术可以分为两种:原因-结果方法和结果-原因方法。前者需要构造错误字典,将一组测试向量 $\{v_1, v_2, \dots, v_n\}$ 输入一个故障芯片,芯片的响应输出与错误字典进行匹配,从而定位芯片的问题所在。结果-原因方法不需要错误字典,但需要将测试向量 $\{v_1, v_2, \dots, v_n\}$ 输入一个正确的芯片模型,进行模拟,将输出结果对比,从而确定芯片的故障位置。在两种方法中,都会返回一组错误备选 $\{F_1, F_2, \dots, F_n\}$, 其中每个错误 F_i 注入网表中,会反映测试向量 v_i 的行为。而后能够得到这些错误的交集 $F = F_1 \cap F_2 \cap \dots \cap F_n$, 表示输入所有测试向量 $\{v_1, v_2, \dots, v_n\}$ 的芯片行为;而集合 F 的大小是快速定位错误的一个重要因素。本文所讨论的基于 SAT 的错误诊断算法就属于结果-原因方法。

电路错误诊断的流程如图 2 所示。首先将芯片的设计规范或者是正确的行为级设计作为参考模型,与待测设计同时作为验证工具的输入,进行验证。验证工具既可以是形式化验证工具,也可以是模拟验证工具。如果验证工具报告通过,则表示验证结束;如果不通过,则表示出现错误,验证工具会输出错误踪迹。若是形式化验证工具,则错误踪迹就是反例;若是模拟工具,则错误踪迹就是模拟波形,进一步得到导致错误的测试激励与正确的运行结果。而后错误定位工具将待测设

计与错误踪迹作为输入,通过自动化的错误定位算法,得到待测设计中可能导致错误的位置,为设计与验证工程师提供参考,便于快速地修正错误。

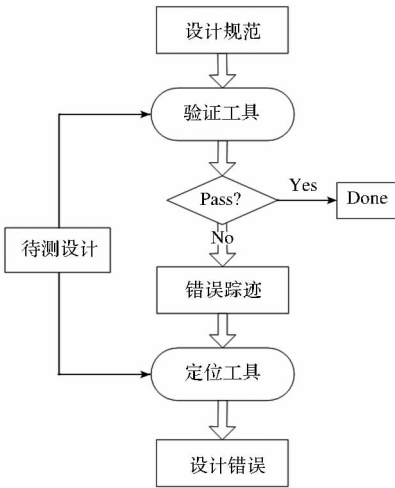


图2 电路错误诊断流程

Fig. 2 Workflow of circuit error debugging

2 基于 SAT 的错误定位方法的基本原理

为了后面叙述的完整与清晰,首先简要地介绍基于 SAT 的错误定位方法的基本原理。依据常见的数字电路设计,我们可以将组合电路抽象为与 (AND) 门、或 (OR) 门、非 (NOT) 门、与非 (NAND) 门、或非 (XNOR) 门、异或 (XOR) 门、同或 (XNOR) 门等基本单元组成;时序电路是由带扫描端口的触发器 (flip-flop) 构成。由于本文着重讨论功能错误,因此这里假设芯片中的测试扫描链是正确的。我们采用文献[4]中的经典方法将组合逻辑电路转换为合取范式 (Conjunctive Normal Form, CNF) 来表达。一个 CNF 的 SAT 公式的构造规则为:命题变元 (x) 及其否定 ($\neg x$) 称为文字;有穷个文字的析取 (\wedge) 构成短句,其中只包含一个文字的句子称为单元短句;有穷个短句的合取 (\vee) 组成合取范式。给出一个 CNF 公式, SAT 求解器能够找出一个变元的赋值组合令公式可满足,或证明其不可满足。

这里假设将组合逻辑电路抽象为 C , 有 m 个输入信号 $X = \{x_1, x_2, \dots, x_m\}$, 和一个输出信号 $y = f(X) = f(x_1, x_2, \dots, x_m)$ 。 C 也容易扩展为多个输出端口的电路。我们采用 $L = \{l_1, l_2, \dots, l_n\}$, 表示电路中的信号。在错误自动定位方法中,还要在每个初始电路上增加 2 类描述,表示为 $S = \{s_1, s_2, \dots, s_n\}$ 和 $W = \{w_1, w_2, \dots, w_n\}$ 。当输入模拟向量 v_j 时,所有表示电路信号的变量 x_i , l_i , w_i 以及 y 都可以定义为电路约束,那么为了便

于表达,我们采用 x_i^j , l_i^j , w_i^j 以及 y^j 表示这些变量,而 X^j , L^j , W^j 和 Y^j 表示这些变量的集合,其中上标 j 表示对应的模拟向量 v_j 。那么 $L = \{l_1, l_2, \dots, l_n\}$ 既可以表示变量,也可以表示电路信号,其中 L 表示对应于所有测试向量。

给出一个逻辑网表和一组测试向量 $\{v_1, v_2, \dots, v_n\}$, 电路错误自动定位算法会引入新的逻辑,并将其编译为一个 CNF 公式 φ 。它包含两部分:第一部分是 k 个 CNF 公式 $C^j(L^j, W^j, X^j, Y^j, S)$ 的合取,其中 $1 \leq j \leq k$; 每个 C^j 对应于逻辑设计的测试向量 v_j 以及可能错误的电路单元;这部分以电路的形式加入到设计中,而后再转换为 CNF 公式。第二部分 $E_N(S)$ 描述了注入错误的数目 N 的约束;这部分约束也是以电路的形式加入到硬件中,而后转换为 CNF 公式;其中 N 是用户自定义的,表示设计中包含 N 个错误。那么 φ 可以表示为:

$$\varphi = E_N(S) \wedge \bigwedge_{j=1}^k C^j(L^j, W^j, X^j, Y^j, S)$$

其中 $\bigwedge_{j=1}^k C^j(L^j, W^j, X^j, Y^j, S)$ 要求对于所有的测试向量 v_j , 电路中错误的备选集都要满足每个 C^j 约束。也就是说,在传统的诊断方式中,错误是所有测试向量的交集。下面来表示公式 φ 的各部分如何构成。

第一部分是由 k 个 CNF 公式 C^j 的合取构成,它反映了不同电路信号的错误。为了表示在信号 l_i 存在一个错误,在该信号处增加一个多路选择开关电路,称为 S_i 。而后将 S_i 转换为 CNF 格式,并加入到公式中。以图 3(a) 为例,假设一个错误存在于信号处 $l = G \rightarrow H$, 那么可以按照文献[5]中的方法,通过增加一个带选择信号 s 的多路器 M 来表达,如图 3(b) 所示。多路选择开关 M 的一个输入连接到与门 G 的输出,另一个输入称为信号 w , 表示设计中潜在的错误;而 M 的输出连接到与非门 H 的一个输入端口。多路选择器的 CNF 转换为包含 4 个短句的公式: $\varphi_M = (s \vee \neg l \vee z) \wedge (s \vee l \vee \neg z) \wedge (\neg s \vee \neg w \vee z) \wedge (\neg s \vee w \vee \neg z)$ 。那么图 3(b) 中整个电路可以转换为 CNF 公式: $C = (x_1 \vee \neg l) \wedge (x_2 \vee \neg l) \wedge (\neg x_1 \vee \neg x_2 \vee l) \wedge (s \vee \neg l \vee z) \wedge (s \vee l \vee \neg z) \wedge (\neg s \vee \neg w \vee z) \wedge (\neg s \vee w \vee \neg z) \wedge (x_3 \vee y) \wedge (z \vee y) \wedge (\neg x_3 \vee \neg z \vee \neg y)$ 。

将多路器引入到电路设计中的每个信号后,将新的电路转换为 CNF 公式。如果在公式 C 中加入一组由电路输入信号 $\{x_1, x_2, \dots, x_m\}$ 和输出信号 y 构成的单元短句,那么就能够得到公式

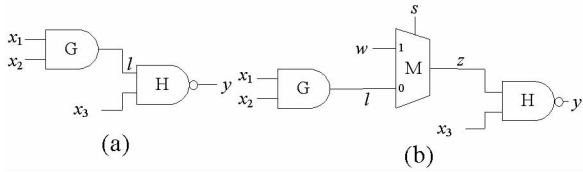


图 3 一个示例电路

Fig. 3 An example circuit

C^j 。这组短句分别代表测试向量 v_j 的逻辑值,也就是说,如果在 v_j 中,一个输入信号 x_j 被赋值为 1,那么就将 x_j 加入到公式 C 中: $C \wedge x_j$; 反之,若 x_j 被赋值为 0,那么就将 $\neg x_j$ 加入: $C \wedge \neg x_j$ 。例如图 3(a) 中的电路中,假设在信号 l 处有一个 stuck-at 1 的错误。当一个测试向量的输入序列为 $v = (x_1, x_2, x_3) = (1, 0, 1)$ 时,电路的输出端口 $y = 0$; 但是正确电路的输出应该为 1。这时要将单元短句 $x_1, \neg x_2, x_3$ 和 $\neg y$ 加入到公式 C 中,得到新的 CNF 公式为: $C^v = C \wedge x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg y$ 。反复进行这个过程,直到遍历所有的测试向量 $v_j, 1 \leq j \leq k$, 而后得到 k 个 CNF 公式: $C^j(L^j, W^j, X^j, Y^j, S)$ 。

第二部分是关于电路中错误的数目 N 的约束。当电路中包含至多 N 个错误时,我们将电路转换为 CNF 公式 $E_N(S)$ 。我们首先考虑电路中只包含一个错误 $E_1(S)$ 的情况,而后再将其一般化。以图 3(b) 为例,当测试向量 $v = (x_1, x_2, x_3) = (1, 0, 1)$ 时,得到公式 $C^v = C \wedge x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg y$ 。假设将多路器 M 的选择信号 s 作为单元短句,引入公式,则公式 $C' = C^v \wedge s = C \wedge x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg y \wedge s$ 。对于公式 C' , SAT 求解器会搜索使其满足的所有变元的赋值,包括表示潜在设计错误的变元 w 。由于公式 C' 表示该错误电路在测试向量 v 下的行为,则 $y = 0$, 而单元短句 $s = 1, x_3 = 1$, 因此决定了变元 w 的赋值为 1, $w = 1$ 表示在电路中的 l 处有 stuck-at 1 错误。

那么对于单个错误的情况,给出一组选择信号 $S = \{s_1, s_2, \dots, s_n\}$, 并且保证任一时刻只能有 1 个 s_i 有效,则 $E_1(S)$ 表示为:

$$E_1(S) = (s_1 \vee s_2 \vee \dots \vee s_n) \wedge \bigwedge_{\substack{i=1, \dots, n-1 \\ j=i+1, \dots, n}} (\neg s_i \vee \neg s_j)$$

$E_1(S)$ 分两部分:前一部分表示至少有一个 s_i 被赋值为 1, 后一部分表示若 s_i 和 s_j 同时为 1, 则公式不可满足。我们可以将 $E_1(S)$ 扩展为多个错误的情况,例如电路包含 2 个错误的情况 $E_2(S)$ 表示为:

$$E_2(S) = (s_1 \vee s_2 \vee \dots \vee s_n) \wedge \bigwedge_{\substack{i=1, \dots, n-2 \\ j=i+1, \dots, n-1 \\ p=j+1, \dots, n}} (\neg s_i \vee \neg s_j \vee \neg s_p)$$

综上所述,首先将电路设计根据上述的方法转化为公式 φ 。当 $N = 0$ 时,公式 φ 是不可满足的,这是因为如果所有的错误备选都没有被选择,即假设电路不存在错误,那么当测试激励输入电路设计时应当得到错误的结果,但是公式 φ 包含了正确的结果 y , 因此会导致公式 φ 不可满足。当 $N = 1$ 时,表明当前电路中包含 1 个错误,将公式 φ 作为 SAT 求解器的输入; 对于 φ 的一个可满足赋值,每个选择信号 $s_i = 1$ 的赋值都代表该多路选择器所旁路的单元是一个错误备选。为了搜索更多的错误,对于每个可满足赋值,都要在公式 φ 中增加一个阻滞短句,避免再次出现该组赋值; 而后将新得到的公式继续作为 SAT 求解器的输入,循环反复上述的过程,直到 SAT 求解器最终返回不可满足,表明已定位电路中所有的错误。

3 错误定位算法研究现状

基于级级求解的错误定位算法最初是由 Toronto 大学的研究小组^[3,6-7]提出的,算法集成了一个高效的 SAT 求解器 zChaff, 并在 ISCAS85 测试集的一组电路上给出了算法的实验结果。Ali 等^[8]针对实际芯片在调试过程中出现的 bug, 采用一种自顶向下的方法和量化布尔公式 (Quantified Boolean Formula, QBF) 求解器来获取时间与空间的一种权衡。在性能与诊断质量方面, Fey 等^[9]将基于模拟的错误定位方法与基于 SAT 的错误定位方法进行了全面地理论分析与实验对比。

Safarpour 等^[10-11]提出了利用公式的最大或近似最大可满足性,通过求解全体极小修正集来快速剔除潜在的错误源,从而提高设计调试的效率; 算法的测试集已经扩展到 ISCAS85、ITC99 以及 OpenCores 上的一些设计。Safarpour 等^[12-13]为了简化状态空间,提出了将状态抽象与功能抽象两种抽象精化技术应用到基于 SAT 的错误定位算法,从而提高算法的求解效率,但定位精度会下降。

Bremen 大学的 Sülflow 等^[14]首次提出了采用求解不可满足子式的方式进行电路的错误定位,既可以同时检测所有的潜在错误,也能够加速错误定位的过程。该研究小组在 DATE'09 上^[15]提出了基于反例搜索的错误定位方法,这里要注意的是错误定位中的反例是指最终会导致错误的输入激励,与模型检验中的反例不是一个概念。该小组在 ISCAS'10 上^[16]提出了基于 QBF 公式,在特定长度的反例中搜索完整错误备选的方法。该

小组提出了基于可满足性模来进行错误自动定位的算法^[17],实验表明,对于超过 90% 的实例,能够显著缩短运行时间。该小组通过实验分析了三种 cardinality 约束的编码方式^[18]:加法网络、多路器编码方式、移位器编码方式。

吴洋等^[19]结合传统基于模拟的方法和布尔可满足性求解技术,提出了一种组合电路的设计错误诊断方法,以及一种基于 SAT 的增量式电路诊断方法,算法中融合了多种启发式方法,从而提高了算法的效率。该研究小组在文献[20]中通过对布尔可满足解计数来实现对多个逻辑错误的诊断定位,并改进了电路诊断架构,保证了算法的高效性。

Chang 等^[21]首次将基于 SAT 的错误诊断延伸到错误修正方面,提出将反例或 bug 踪迹应用到错误自动修正工具中,并且引入了两种重综合技术:面向全局的搜索和熵引导的搜索,用以修正电路内部的错误结点,但是由于错误修正的复杂度太高,远高于错误定位,目前的错误修正算法都无法处理较大规模电路。

Keng 等^[22-23]提出了一种可扩展的基于 SAT 的错误诊断算法,算法引入了插值来约束错误行为,显著减少了检测错误踪迹的时间片数目。该小组将不可满足核指导的抽象精化技术引入了错误定位算法^[24],用于减小设计规模和错误踪迹长度,提高定位效率。该小组提出了一种基于路径指导的窗口抽象精化技术的错误诊断算法^[25],该算法将错误踪迹划分为不重叠的时间窗口,而后算法运行于每个时间窗口上,能够减小空间代价,增加处理问题的规模。

Mangassarian 等^[26]提出了电路单元之间的结构主导点的概念,并提出了一种搜索电路设计中多输出模块之间主导点的不动点算法,该算法能够有效地减少定位所有错误的循环次数。Zhu 等^[27]引入了电路符号化表示的中枢概念,它是指电路中在给定约束下值不变的一组信号,同时采用极大可满足求解器,加速电路中错误定位的效率。Le Bao 等^[28]提出了反主导点与非溶解蕴含的概念,通过改造 SAT 求解器,使其能够在求解可满足性的同时得到反主导点,从而减少错误定位的搜索空间。

4 主要挑战与研究展望

目前绝大多数关于电路错误诊断与定位技术的研究都是基于位级(bit-level)求解方式——布尔可满足求解器。自从融合冲突学习机制等启发

式方法的增强 DPLL 算法出现之后,SAT 求解器得到了飞速的发展,已经能够高效地求解实际应用中布尔公式的可满足性。但是当前所面临的发展趋势是抽象层次更高、更接近于真实设计的字级(word-level)应用越来越受到人们的重视。由于 SAT 求解器的输入布尔公式是将原始问题转换到位级,从而导致了两个问题:一是失去了原始设计中大量的逻辑信息,令运行结果更加难以理解;二是求解时间与空间开销的显著增加,限制了解决问题的规模。例如在集成电路设计中,目前工业界所采用的形式化验证工具大多基于门级电路,开销很大,难以适应飞速增长的芯片规模,如果直接在字级,例如寄存器传输级甚至行为级进行验证,那么将会显著地降低开销,大大提高解决问题的效率,并且使得结果易于理解,帮助程序员迅速定位错误并修正设计。

针对这些应用需求,基于字级的可满足性模(Satisfiability Modulo Theories, SMT)求解技术近年来得到了突飞猛进的发展,并在计算机辅助验证国际会议 CAV 上开展了关于 SMT 求解器求解效率与正确性的竞赛,更是大大促进了 SMT 求解技术的成熟与完善,使其成为字级求解技术中发展最快的一个分支。由于 SMT 采用字级建模语言,描述能力更强,更加接近于高层设计,避免将问题转化到位级求解,具有更大的潜力,以及更加广阔的应用前景,所以 SMT 求解技术在诸如硬件的 RTL 级形式化验证等众多领域得到了广泛的应用,并且取得了良好的效果。

目前,学术界已经出现了一些基于 SMT 的电路错误定位方法的研究成果^[17-18],但是这些研究成果只是给出了简单的实验对比与分析,能够处理的理论域很少,并没有建立完善的理论框架和详尽的实验分析,未来研究的空间仍十分广阔。因此,总的来说位级求解器存在丢失逻辑信息、结果难以阅读、增加时间与空间开销、发展前景受限等问题;相反,字级求解技术,例如 SMT,克服了这些问题,并且在近年来得到了突飞猛进的发展,大大提升了求解效率,逐渐显现出取代位级求解器的势头,因此未来的电路错误定位算法的研究必将会采用先进的字级求解技术,使其符合当前技术发展的趋势。综上所述,基于 SMT 求解技术的电路错误自动定位方法能够满足业界对于快速、精确定位芯片设计中错误的需求,将成为未来一个研究的热点和难点问题。

5 结束语

如何准确地定位与诊断电路中的错误已成为

VLSI 设计中影响芯片面市时间的一个重要因素,业界迫切需要一种高效的自动化工具来帮助诊断与定位设计中的错误,因此近几年基于 SAT 的电路错误自动诊断与定位技术的研究飞速发展,已取得了不少成果,但是该研究领域仍旧很开放,十分活跃;而基于 SMT 的电路自动错误定位技术刚刚开始起步,还缺乏完整的理论体系和高效的求解算法,随着 SMT 求解技术的不断完善,该领域必将成为未来研究与应用的热点及主要突破的方向。

参考文献 (References)

- [1] Bose P, Albonesi D H, Marculescu D. Guest editors' introduction: power and complexity aware design [J]. *IEEE Micro*, 2003, 23(5): 8 - 11.
- [2] Kumar N, Kumar V, Viswanathan M. On the complexity of error explanation [C]//Proceedings of VMCAI'05, 2005: 448 - 464.
- [3] Smith A, Veneris A, Ali M F, et al. Fault diagnosis and logic debugging using Boolean satisfiability [J]. *IEEE Transaction on CAD*, 2005, 24(10): 1606 - 1621.
- [4] Larrabee T. Test pattern generation using Boolean satisfiability [J]. *IEEE Transaction on CAD*, 1992, 11(1): 4 - 15.
- [5] Veneris A, Abadir M S, Amiri M. Design rewiring using ATPG [J]. *IEEE Transaction on CAD*, 2002, 21(12): 1469 - 1479.
- [6] Ali M F, Veneris A, Safarpour S, et al. Debugging sequential circuits using Boolean satisfiability [C]//Proceedings of ICCAD'04, 2004: 204 - 209.
- [7] Smith A, Veneris A, Viglas A. Design diagnosis using Boolean satisfiability [C]//Proceedings of ASP - DAC'04, 2004: 218 - 223.
- [8] Ali M F, Safarpour S, Veneris A, et al. Post-verification debugging of hierarchical designs [C]//Proceedings of ICCAD'05, 2005: 871 - 876.
- [9] Fey G, Safarpour S, Veneris A, et al. On the relation between simulation-based and SAT-based diagnosis [C]//Proceedings of DATE'06, 2006: 1139 - 1144.
- [10] Safarpour S, Liffiton M H, Mangassarian H, et al. Improved design debugging using maximum satisfiability [C]//Proceedings of FMCAD'07, 2007: 13 - 19.
- [11] Chen Y B, Safarpour S, Marques-Silva J, et al. Automated design debugging with maximum satisfiability [J]. *IEEE Transaction on CAD*, 2010, 29(11): 1804 - 1817.
- [12] Safarpour S, Veneris A. Abstraction and refinement techniques in automated design debugging [C]//Proceedings of DATE'07, 2007: 1182 - 1187.
- [13] Safarpour S, Veneris A. Automated design debugging with abstraction and refinement [J]. *IEEE Transaction on CAD*, 2009, 28(10): 1597 - 1608.
- [14] Stülflow A, Fey G, Bloem R, et al. Using unsatisfiable cores to debug multiple design errors [C]//Proceedings of ACM GLVLSI'08, 2008: 77 - 82.
- [15] Stülflow A, Fey G, Braunstein C, et al. Increasing the accuracy of SAT-based debugging [C]//Proceedings of the DATE'09, 2009: 1326 - 1331.
- [16] Stülflow A, Fey G, Drechsler R. Using QBF to increase accuracy of SAT-based debugging [C]//Proceedings of ISCAS'10, 2010: 641 - 644.
- [17] Stülflow A, Fey G, Drechsler R. Experimental studies on SMT-based debugging [C]//Proceedings of IWHLT'08, 2008: 93 - 98.
- [18] Stülflow A, Wille R, Fey G, et al. Evaluation of cardinality constraints on SMT-based debugging [C]//Proceedings of ISMVL'09, 2009: 298 - 303.
- [19] 吴洋, 唐璞山. 基于布尔可满足性的电路设计错误诊断算法 [J]. *计算机辅助设计与图形学学报*, 2006, 18(9): 1383 - 1390.
WU Yang, TANG Pushan. Multiple design error diagnosis using Boolean satisfiability [J]. *Journal of Computer-aided Design and Computer Graphics*, 2006, 18(9): 1383 - 1390. (in Chinese)
- [20] 吴洋, 唐璞山. 一种改进的基于 SAT 的多错误诊断算法 [J]. *微电子学与计算机*, 2007, 24(2): 9 - 13.
WU Yang, TANG Pushan. SAT-based algorithm for multiple design error diagnosis [J]. *Microelectronics and Computer*, 2007, 24(2): 9 - 13. (in Chinese)
- [21] Chang K H, Markov I L, Bertacco V. Fixing design errors with counterexamples and resynthesis [J]. *IEEE Transaction on CAD*, 2008, 27(1): 184 - 188.
- [22] Keng B, Veneris A. Scaling VLSI design debugging with interpolation [C]//Proceedings of FMCAD'09, 2009: 144 - 151.
- [23] Keng B, Safarpour S, Veneris A. Bounded model debugging [J]. *IEEE Transaction on CAD*, 2010, 29(11): 1790 - 1803.
- [24] Keng B, Veneris A. Managing complexity in design debugging with sequential abstraction and refinement [C]//Proceedings of ASPDAC'11, 2011: 479 - 484.
- [25] Keng B, Veneris A. Path directed abstraction and refinement in SAT-based design debugging [C]//Proceedings of DAC'12, 2012: 947 - 954.
- [26] Mangassarian H, Veneris A, Smith D E, et al. Debugging with dominance: on-the-fly RTL debug solution implications [C]//Proceedings of ICCAD'11, 2011: 587 - 594.
- [27] Zhu C S, Weissenbacher G, Malik S. Post-silicon fault localisation using maximum satisfiability and backbones [C]//Proceedings of FMCAD'11, 2011: 63 - 66.
- [28] Le Bao, Mangassarian H, Keng B, et al. Non-solution implications using reverse domination in a modern SAT-based debugging Environment [C]//Proceedings of DATE'12, 2012: 629 - 634.