

基于模拟的 SoC 功能验证研究

姚爱红¹, 孙盟哲¹, 袁莉娜²

(1 哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001; 2 中国科学院 电子学研究所, 北京 100190)

摘 要: 基于模拟的验证技术在 SoC 等复杂数字系统功能验证中一直占据统治地位. 模拟激励产生的速度和质量是决定验证过程收敛速度的关键因素, 覆盖率信息定量地表示了验证完成的程度. 覆盖率导向的受约束随机激励生成技术提高了验证过程的效率和自动化程度. 按照是否需要待验证设计的内部结构信息, 将受约束随机激励生成技术分为基于学习的激励生成和基于构造的激励生成两类并分别进行分析. 对基于模拟的 SoC 功能验证涉及的其它关键技术如: 约束定义、覆盖率分析及 IP 核及核间通信协议的功能验证等国内外研究现状进行了分析和总结, 并对未来的发展趋势和研究方向进行了展望.

关键词: 功能验证; 受约束随机模拟; 激励生成; 覆盖率; 系统芯片

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2013)05-0001-09

A Survey on Simulation Based Functional Verification of SoC

YAO Ai-hong¹, SUN Meng-zhe¹, YUAN Li-na²

(1 College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China;

2 Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Simulation-based verification remains holding dominance in the verification of SoC and other sophisticated digital systems. The generating speed and distributing quality of stimuli are crucial factors in the convergence progress of verification. Coverage metrics quantitate how much the verification process has been completed. Coverage-directed and constrained-random stimulus generation methods have been proposed to improve the productivity and automaticity of the verification process. Constrained random stimulus generation techniques are categorized into two groups according to whether the structure information of the design under verification being used. In this paper, a survey is made on the state of art of the key technologies in simulation based verification of SoC in the following aspects: constraint definition, stimulus generation, coverage analysis, IP core and inter-communication verification. And finally, the future trends and resarch directions are predicted.

Key words: functional verification; constrained random simulation; stimulus genenration; coverage; System on Chip

1 引言

由于 SoC 日益增长的规模和复杂性以及产品面世时间的缩短, 数字系统生产能力、设计能力和验证能力上的不协调愈加严重. 模型检查、等价性验证以及可达性分析等形式化验证技术近年来取得了很大的进展, 但是受限于状态爆炸等问题, 所能处理的设计规模和复杂度还不能满足复杂设计需要, 对于

SoC 等复杂数字系统的验证, 基于模拟的功能验证在当前和今后相当长的时间内都是不可取代的验证手段. 通过模拟输入激励搜索设计空间, 具有直观、易用并能在早期发现绝大多数设计缺陷等优点. 能否高效地产生均匀覆盖设计空间的测试激励是决定模拟验证过程收敛速度的关键因素, 受约束随机激励生成通过定义一组约束条件限制产生的输入组合来加快达成验证目标的速度, 是目前广泛应用的模

收稿日期: 2012-07-18; 修回日期: 2012-08-20

基金项目: 黑龙江省自然科学基金项目(F201111); 中央高校基本科研业务费专项基金项目(HEUCF100606)

拟验证技术。

近十余年来,覆盖率驱动的受约束随机模拟成为数字电路功能验证领域研究热点,利用覆盖率信息指导模拟过程减少了对专家知识与经验的依赖,提高了验证过程的自动化程度. Nativ 等^[1]应用基于反馈的覆盖率驱动模拟方法对 IBM Z-series 系列服务器芯片进行验证,结果显示使用受约束随机模拟比纯随机模拟完成验证计划速度更快,且能够覆盖更多新的验证任务,同时增加了发现新的设计缺陷的概率。

本文对 SoC 模拟验证所涉及的激励生成、约束自动抽取、覆盖率信息分析、IP 核及核间通信协议的功能验证,以及硬件加速模拟验证等关键技术的现状和发展分别进行了详细的总结和分析,并对未来的发展趋势进行了展望。

2 受约束随机激励生成

按照验证过程中是否使用电路的结构信息,可以将受约束随机激励生成分为两大类:基于学习的受约束随机激励生成(LBCRSG, Learning Based Constrained Random Stimulus Generation)和基于构造的受约束随机模拟(CBCRSG, Construction Based Constrained Random Stimulus Generation). LBCRSG 将覆盖率作为衡量验证是否充分的主要指标,利用覆盖率信息作为输入激励质量的评价反馈,通过学习过程调整产生的激励模式,增大命中功能覆盖点的输入向量的概率,从而加快验证过程的收敛速度. 如图 1 中箭头①所示,通过在覆盖率结果到测试向量随机生成之间形成一个闭合的环,自动生成针对具体覆盖点的测试向量序列. CBCRSG 从系统规范或设计实现出发,通过构造抽象的设计模型,利用工具自动提取约束或手工编写约束(如图 1 中箭头②所示). 然后借助于约束求解器或基于电路模型得出合法的输入值域,按照一定的概率分布在该值域中随机采样,将生成的采样值作为待验证设计(DUV, Design Under Verification)的输入激励。

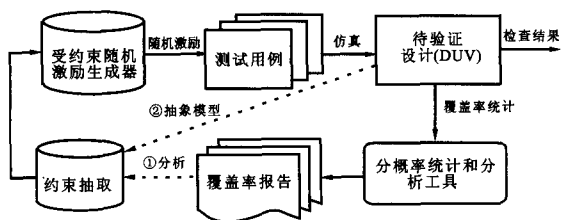


图1 受约束随机模拟过程

两种方法的主要区别是:LBCRSG 的思想近似

于黑盒测试,验证者的先验知识体现在所构造的学习机制上(如演化算法中的适应度函数或机器学习方法中的聚类模型). 而 CBCRS 则近似于白盒测试的思想,验证人员需要对 DUV 进行分析并建立系统模型(如控制状态机或 BDD 图). 受约束随机激励生成技术的分类和简要说明参见表 1.

表1 受约束随机激励生成分类

激励生成分类		说明
受约束随机激励生成	基于学习	基于演化计算 适于模块级或微处理器测试程序生成
		基于机器学习 适于模块级电路
	基于构造	基于图遍历 适于通信协议验证,但需要建立状态机模型
		基于约束求解 需要定义约束并受限于约束求解器效率

2.1 激励分布、约束和覆盖率之间关系

输入激励的分布应尽量与覆盖点匹配以提高验证过程收敛速度. 但实际上往往无法获得覆盖点与输入向量之间的匹配信息,此时选择具有较高均匀分布的输入激励可以增大命中设计状态空间中未知区域的机会(即增加命中未覆盖的覆盖点的机会). 利用最大熵原理分析如下:

令随机变量 X 表示激励向量,约束集 $C = \{C_1, C_2, \dots, C_L\}$ 对随机变量 X 的取值进行选择,定义相应的香农信息熵为:

$$E_C = - \sum_{i=1}^L \frac{f(C_i)}{N} \log_2 \frac{f(C_i)}{N} \quad (1)$$

其中, $f(C_i)$ 表示约束 C_i 的解向量命中覆盖点的数量, N 表示所有可能命中覆盖点的输入激励数. 熵 E_C 定义了输入激励变量 X 的不确定性, $\frac{f(C_i)}{N}$ 表明约束 C_i 的解向量命中覆盖点的概率. 最大熵原理指出:当概率分布 $\frac{f(C_i)}{N}$ 未知时,令 E_C 达到最大的随机变量 X 的取值是最不偏倚的. 由公式

(1) 可以看出,当 $\frac{f(C_i)}{N}$ ($i=1, 2, \dots, L$) 相等时,即所有约束 C_i 的解向量分布最均匀时熵 E_C 最大. 此时,通过随机变量 X 选择输入向量空间中同样的激励数能命中覆盖点的概率最高. 此外,出现概率低的事件携带的信息量大,即覆盖设计边角情况(corner case)的激励对验证过程收敛的贡献更大.

2.2 基于学习的受约束随机激励生成

基于学习的受约束随机模拟(LBCRSG)的过程是:根据已执行的模拟向量和相应的覆盖率报告之

间的关系,在验证过程中进行不断地学习和修改,从而逐步趋向期望的覆盖率.然而覆盖率模型、DUV 属性点和生成的测试向量之间并不是一一对应的关系,而是复杂的网状联系,并且测试向量序列本身也是上下文有关的,这使得单纯的 LBCRS 变得很困难,局限了其应用规模.根据对于覆盖率反馈的学习方法不同,可以将 LBCRS 分为基于演化计算的方法和基于机器学习的方法两类.

2.2.1 基于演化计算的方法

演化计算是模仿自然界生物演化过程而提出的一种求解方法,主要包括遗传算法、演化策略、演化规划和遗传程序设计等四个主要分支.其主要优点是不必明确描述问题的全部特征,只需要根据一定规则(如适应度值提高)产生新的更好的解,演化过程结束时就一定能够获得较优解.利用演化计算实现激励生成需要解决两个问题:激励向量的编码表示问题和适应度函数的选取问题.目前的研究多是以一个小的初始测试向量集作为初始种群,以覆盖率作为适应度值,指导演化过程得到优化的测试向量集. Bose 等^[2]采用基于演化计算的方法优化原有的带偏置(biased)随机测试向量生成平台,使用遗传算法自动生成参数偏倚和权重,将单缓冲器和多缓冲器的使用率作为适应度值. Samarah 等^[3]将遗传算法应用于微处理器的测试程序生成,采用二进制编码表示指导规则,每个基因包含 3 个附加参数:权重、上界和下界. 易江芳等^[4]使用遗传算法自动生成模拟矢量,将指定模拟周期数内所有输入信号的二进制编码序列作为单个染色体,将基于关键信号的路径覆盖率作为适应度值. 在北大志一 863 芯片的模块级验证中,decoder 模块的覆盖率最高达到 86%.

在微处理器设计或者基于微处理器的系统芯片的功能验证中,测试向量通常是汇编语言编写的测试程序. 罗春等^[5]提出了一种基于遗传算法和覆盖率驱动的功能验证向量的自动生成方法,以向 DUV 输入的数据包作为染色体,以每个个体所覆盖的功能点的加权和作为适应度值. 应用该方法实现 SoC 模块级验证的结果显示,使用遗传算法比全随机算法的功能点覆盖率提高了 18%,且收敛速度较快. 沈海华等^[6]提出基于遗传算法的覆盖率驱动微处理器测试程序生成方法,通过定义两类约束模型:体系结构相关约束模型和通用约束模型,控制生成用于处理器测试的机器指令序列.

基于演化计算的 LBCRS 方法主要存在的问

题是随着测试向量位数的增长,染色体长度和种群规模急剧增大,执行遗传操作所需的存储空间和计算时间都迅速增加. 此类方法主要用于模块级验证或者微处理器的测试程序自动生成,不适用于 SoC 等复杂数字电路寄存器传输级(RTL, Register Transfer Level)测试的激励生成.

2.2.2 基于机器学习的方法

基于机器学习的 LBCRS 验证过程一般分为三个步骤:首先运行部分随机模拟,获取测试向量和覆盖率样本;然后通过聚类分析覆盖率模型和随机测试向量之间的关系,建立分类模型;最后利用建立的分类模型识别新的模式,选择优化的测试向量加速验证收敛过程. Tasiran 等^[7]利用稳态 Markov 分析,提出了一种基于时序电路的半形式化模型验证方法,其核心是基于标记覆盖率(tag coverage)引导随机测试向量的生成. 通过将电路结构近似等效为一个稳态的 Markov 链,利用优化算法不断扰动原始输入的概率分布,改进覆盖率结果. 在 DLX/Intel 8085 处理器上的实验结果显示,使用 Markov 链反馈修改比完全随机方法能获得更高的标记覆盖率,并且收敛速度快了近一半. Wen 等^[8]采用机器学习算法分析输出数据,针对控制电路和数据通路模块分别采用不同的学习方法. Guzey 和 Wang 等^[9]采用无监督支持向量机分析给定的测试集,然后筛选出随机生成的测试向量中的冗余测试向量,从而减少模拟周期.

基于机器学习的 LBCRS 方法的主要缺点是对初始样本集的依赖性,例如在文献^[9]所提出的基于数据挖掘的约束抽取方法,对于样本集中未出现过的模式,无法通过学习获得相应的约束. 此外,由于基于机器学习方法的激励生成算法的计算复杂度较高,当输出记录较大时算法性能往往较差,不适用于实时产生大量的测试向量且所处理的逻辑规模往往比较小,目前还主要用于电路的模块级验证.

2.3 基于构造的受约束随机激励生成

IBM 公司的 Haifa 实验室于 1995 年提出,在验证中应注重利用验证知识(如设计信息)^[10],这一原则可以看作是构造的随机验证的起源. CBCRS 方法通过遍历待验证设计的控制模型状态机,根据遍历结果生成相应的测试向量序列,并应用激励偏置技术提高覆盖功能故障点的概率. 由于在生成测试激励时利用待验证设计的内部结构信息避免了产生无效激励,而且通过建立功能故障模型指导测试生成过程, CBCRS 往往能够以更少的测试用例更快

地达到验证目标. 但是当粒度过细时, 该方法存在“状态爆炸”问题. 按照对控制状态机的遍历方法, 可以将 CBCRSG 分为基于图遍历的方法和基于约束求解的方法两类.

2.3.1 基于图遍历的激励生成

如果将待验证设计的控制状态机表示为有向图, 则可以利用图论中的搜索算法遍历状态机. 从初始状态至目标状态之间的路径为状态迁移的序列, 该序列中每条边上的触发条件即为要产生的激励, 激励生成问题转化为图的可达性分析问题. 为了避免所产生的激励局限于状态空间的局部, 一般需要对传统的有限状态机进行扩展, 引入转移概率等属性, 例如使用扩展有限状态机 (EFSM, Extended Finite State Machine) 对通信协议中控制状态转移进行建模. Guglielmo 等^[11]应用启发式方法遍历 EFSM, 通过搜索最有可能出现错误的状态提高受约束随机模拟对边角情况的覆盖率. Albin 和 Yuan 等^[12]应用形式化状态可达性分析扩展对状态空间的搜索 (随机模拟的方法可能达不到). 尽管这种方法确实有助于更深入地搜索, 且减轻了用户编写测试平台的劳动, 但是受限于形式化引擎的能力限制, 其效率低且生成的路径可能质量很差. Paula 等^[13]将形式化引擎应用在抽象层, 而非具体模型上, 一旦生成了抽象的路径, 就可以用它指导产生一个具体的路径. 尽管该方法对某些测试效果很好, 但其它情况下存在求解时间过长或无法求解等问题.

2.3.2 基于约束求解的激励生成

基于约束求解的方法是把状态图转化成 Kripke 结构或者合取表达式, 运用模型检查或可满足性 (SAT, Satisfiability) 求解等形式化方法对状态空间进行遍历, 主要分为基于二元决策图 (BDD, Binary Decision Diagram) 的方法和基于 SAT 求解的方法两类.

基于 BDD 的激励生成方法首先将约束转换为加权 BDD, 按照期望的分布为节点的每个分支分配权值, 遍历该 BDD 即生成期望分布的输入模式^[14]. 然而, 由于 BDD 的自身的内存爆炸问题, 该方法不能处理复杂控制或整数函数的设计. Kukula 等^[15]构建与约束 BDD 匹配的电路, 该方法不要求对变量排序, 因此它避免了内存爆炸问题, 但是其产生的激励分布有可能是很不均匀的, 从而有可能使得覆盖率不能收敛.

随着近十余年 SAT 求解研究的快速发展, 出现了 GRASP、SATO、zChaff 等高性能 SAT 求解器,

近几年提出了许多基于 SAT 求解的受约束随机激励生成方法. 但是, 目前的 SAT 求解器普遍是针对布尔约束求解的, 只适用于门级电路的 SAT 问题, 而 RTL 电路的 SAT 问题往往是字位混合求解的. 这是因为 RTL 电路往往由数据通路和控制逻辑组成: 数据通路中包含加法器、乘法器以及多路选择器等字级 (word-level) 变量; 控制逻辑中包含实现数据通路器件控制的位级 (bit-level) 变量. Barrett 等^[16]首先实现了布尔、整数、数组及不等式的混合约束求解算法, 奠定了基于 SAT 模理论进行混合 SAT 求解的基础. Fallah 等^[17]提出一种在字域和位域分别求解的方法, 对算术等式等字级约束使用线性规划 (LP, Linear Programming) 求解, 控制信号等位级约束使用 3-SAT 技术求解, 字/位接口部分使用拆分字的方法. Kitchen 等^[18]设计 Ambigen 工具, 利用了基于马尔科夫链和蒙特卡罗法的约束求解器, 该求解器能实现布尔/整数混合求解, 所产生的随机激励具有较好的性能和分布, 但是并未解决手工进行约束抽取和定义的困难. 李瞰等^[19]采用断言作为验证功能模型, 通过将设计的决策图模型和设计断言转化为约束逻辑编程 (CLP, Constraint Logic Programming) 的约束条件, 求解 CLP 约束产生激励向量. 该方法统一处理控制电路和数据通路间的数据传播, 避免了在位级求解和字级求解之间来回切换的开销. 关于 RTL 电路的混合 SAT 求解问题的更详细研究综述可参见文献^[20].

3 约束定义

约束抽取过程根据 DUV 的抽象模型或覆盖率反馈生成约束条件, 对随机激励生成器进行偏置. 目前, 约束的抽取和定义还普遍采用专家分析和手工编写的手段, 效率较低而且容易出错. 另一方面, 模拟运行阶段的效率很大程度上取决于约束求解的性能及所生成的激励分布质量, 定义不当的约束有可能产生分布不均匀的激励, 延缓收敛速度. 因此, 研究约束的自动生成技术对于提高受约束随机验证的效率具有重要意义.

Hari S. 等^[21]提出一种从电路的行为级描述中自动抽取约束模型的方法. 该方法以行为级硬件描述语言代码作为输入, 自动转化为基于赋值决策图 (ADD, Assignment Decision Diagram) 的数据结构, 优化后再转换为一组整数约束, 称之为模型约束 (model constraints). 与场景约束 (scenario constraints, 与待测试的场景相关) 一起通过整数求解

器进行求解,生成期望的测试向量.该方法需要遍历设计中所有的赋值表达式以构造设计的控制数据流图,不适用于复杂控制逻辑,只能用于模块级验证的约束抽取.Yeh H. 等^[22]提出从覆盖漏洞(coverage holes)出发自动生成约束的方法.覆盖漏洞定义与所采用的覆盖率度量方法有关,Yeh H. 等将未能使输入传播至输出的条件作为覆盖漏洞,如图 2 中的 $op == ALU_ADD$ 是使输入信号 in_a 传播至输出端 out 的条件,若输入激励中未出现过 $op == ALU_ADD$,则此处存在覆盖漏洞.该方法首先利用初始激励运行模拟,收集覆盖漏洞信息,然后分析设计的控制数据流图寻找具有较大影响的控制信号作为初始敏化目标,然后通过后向约束传播和启发式锥选择算法生成 SystemVerilog 形式的约束.

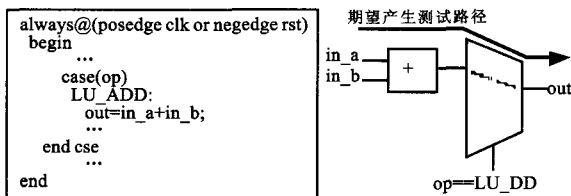


图 2 覆盖率漏洞示例

Guzey 等^[23]提出了基于数据挖掘方法的约束抽取方法,通过分析测试平台的输出数据进行约束抽取,通过与现有的商业化受约束随机测试生成平台集成,提高复杂设计中的底层信号的可控性.此方法的效果依赖于初始输出记录是否均匀覆盖了各种模式.李墩等^[19]提出基于断言约束自动生成方法,将设计的决策图模型和设计断言转化为 CLP 约束,该方法需要预先定义完备、无冲突的断言.

不恰当定义的约束可能产生死端(dead end),此时必须重新启动模拟过程.死端是指对于一组顺序约束在给定的周期后约束无法求解,即当前模拟状态下不存在合法的输入.死端检测和分析技术可以作为自动约束生成的辅助和补充.Dsouza 等^[24]对基于 SVA 的受约束随机模拟中的顺序约束合理性进行分析,检测会引起死端(deadend)的约束,并添加约束进行消解.

4 覆盖率分析

覆盖率分析一方面可以定量衡量验证完成的程度,另一方面可以指导探索未覆盖的设计空间,对于提高受约束随机模拟的效率非常关键.常用的覆盖率包括代码覆盖率、可观测性覆盖率、数据流覆盖率以及功能覆盖率等.各种覆盖率指标都只能度量设计的某方面属性,因此单独使用一种覆盖率是不全

面的,各种覆盖率模型的比较分析参见文献[25].代码覆盖率分析来源于软件测试领域,通过测试 HDL 代码执行程度衡量验证完成程度,包含行覆盖率、路径覆盖率、条件覆盖率以及翻转(toggle)覆盖率等.目前代码覆盖率统计分析工具已经较为成熟,如 Synopsys VCS Coverage Metrics、Mentor Graphics ModelSim 及 Cadence Code Coverage 等.代码覆盖率仅能反映 DUV 的代码执行程度,而难以有效反映对设计意图的验证程度.例如,在图 3 所示含有设计错误的代码中,“if(counter>4'0101)”本应该是“if(counter>4'0111)”,即使产生激励激活该段代码获得了 100% 的行、条件、翻转覆盖率,仍然不能发现这一设计错误,此即为设计的边角情况(corner case).

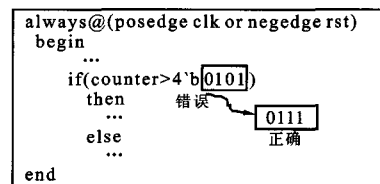


图 3 含有设计错误的代码示例

功能覆盖率分析是依据待验证设计的功能规格说明,采用验证语言提供的覆盖率行为级描述或断言描述,对设计的边角逻辑和感兴趣的数据及控制信号组合进行覆盖率建模及统计.不过,功能覆盖率统计的全面性取决于功能覆盖模型的完整程度,而随着设计规模的增大,系统越来越复杂,很难得到完备的功能覆盖模型.在实际的验证项目中,代码覆盖率和功能覆盖率互为补充,二者的关系如图 4 所示.只有当两种覆盖率都达到较高的水平(通常代码覆盖率达到 85%,功能覆盖率 100%),才可以认为验证较为充分了.

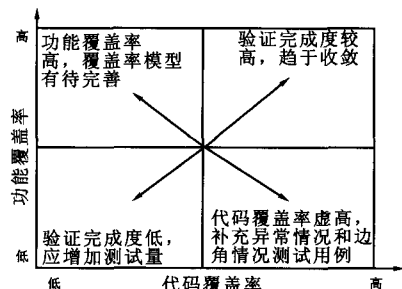


图 4 覆盖率与验证程度关系示意图

由于代码覆盖率可能存在虚高,而功能覆盖模型建立困难,研究者展开了对其它覆盖率度量方法的研究.文献[26]提出了一种数据流覆盖率,基于数据控制流图中的数据分析定义覆盖率.文献[27]提

出了基于 VCD(Value Change Dump)文件的覆盖率统计方法,分析模拟工具生成的各种信号取值变化,同时得到多种类型的覆盖率数据.文献[28]首次将域覆盖率的概念从软件测试领域引入到 HDL 行为级代码的验证中.罗春等^[29]发展和完善了域覆盖率模型,他们通过分析控制流图找到每个路径的激活表达式,然后根据边界条件为表达式中的每个控制变量定义测试点,域覆盖率定义为已激活的测试点加权和与所有测试点加权和的比值.

当前,基于断言的验证逐渐成为一种重要补充验证手段,利用覆盖率断言可以实现对覆盖率信息的实时采集.不但可以检查代码覆盖率,而且对断言执行情况的覆盖统计反映了当前的功能覆盖率,再与模拟环境中原有的功能覆盖点定义相结合,就构成了完整的功能覆盖机制. VMM 验证环境中 SystemVerilog Assertion 语言的覆盖率断言示例如图 5 所示.覆盖组(covergroup)包含覆盖点和交叉覆盖等内容,覆盖点是通过对变量和表达式进行采样进行统计的.覆盖组中还可以定义覆盖点的采样时机和采样次数,或者对交叉覆盖中不可能出现的组合情况进行过滤.

```
class coverage extends vmm_xactor_callbacks;
  local transaction tr; //声明本地事务
  covergroup trans_cov; //声明覆盖率组
    a:coverpoint tr.a{} //声明覆盖点
    b:coverpoint tr.b{}
    corss a,b //交叉覆盖
  endgroup
  function new(); //构造函数
endclass:callbacks
```

图 5 SVA 覆盖率断言示例

5 IP 核的模拟验证

在 SoC 中,由于芯片上需要集成诸如微处理器核、千兆位高速 I/O、块 RAM、专用数字信号处理模块,基于 IP(Intellectual Property)复用的 SoC 设计方法学已经成为主流.面向 SoC 的功能验证面临验证环境复用、包含黑盒系统验证、参数化 IP 核验证以及 IP 核间通信协议验证等问题. IP 核复用不仅包括设计代码的复用,也包括验证组件的复用,以避免系统级验证平台的重复开发.基于断言的验证知识产权核(Verification Intellectual Property)可以与 IP 核同时发布,可以作为一种可复用的验证组件,集成到嵌入 IP 核的目标系统验证环境中.关于如何创建基于断言的 IP 可以参见文献[30].此外,由于知识产权保护,来自第三方的 IP 核内部结构或

逻辑功能往往是未知的,使得 SoC 系统级验证中包含了未知的逻辑,需要研究包含黑盒的设计的验证方法.从受约束随机模拟的角度,本文主要关心参数化 IP 核以及核间通信协议的激励生成问题.

5.1 参数化 IP 核的激励生成

当前普遍采用参数化设计来提高 IP 核的复用性,参数的设置与 IP 核面向的应用及设计属性相关,例如功耗、数据大小或者性能参数.但是,大量的参数显著增加了验证空间和验证代价.文献[31]引入参数域(PD, Parameter Domain)和参数域图(PDG, Parameter Domain Graph),参数域是一组具有依赖关系的参数的有效值,唯一地描述每个参数的值或取值范围.通过分析输入参数的值空间,使用定向测试对设计的边角情况进行覆盖,有效地提高测试覆盖率. Castro 等^[32]研究了基于 PDG 的参数域随机激励生成和与之相匹配功能覆盖率模型. PDG 方法通过将参数空间划分为多个正交的子空间来去除冗余和无效的参数组合,避免了由参数直接组合生成过多无效测试向量.但是要构造给定设计的 PDG,需根据功能规范文档和 RTL 代码,手工抽取所有参数依赖关系,工作量大且容易出错.

5.2 IP 核间通信协议的自动激励生成

IP 核间通信协议的自动验证并不是一个新问题,在通信领域已经有许多相关的研究,其验证目标主要是协议本身的逻辑正确性,确保在协议开发的前期最大限度地检测和纠正协议错误. IP 核间通信协议验证的特点体现在其往往具有形式化的规格说明,可以看做约束的一种描述形式,从而具备了自动生成模拟输入和覆盖率模型的基本条件.

Shimizu 等^[33]研究了由接口的形式化设计规约自动生成模拟输入和覆盖率模型的方法.模拟开始前按照属性说明的格式列出接口的约束,然后在模拟时实时创建相应的约束 BDD.该方法避免了在模拟前构造 BDD 的复杂性,但是需要在每个周期重建一组新的 BDD. Shih C. 等^[34]采用非确定扩展有限状态机(NEFSM, Nondeterministic Extended Finite State Machine)设计验证接口协议的自动验证激励生成器.对接口协议建模,通过评估、选择和更新三个阶段遍历 NEFSM 的状态空间,实现激励的自动生成.但是,该方法需要手工构造接口的 NEFSM 模型,对具有复杂控制机制(如异常处理)的接口协议,细粒度的状态机存在状态爆炸问题.

6 未来研究方向

SoC 几乎无限大的设计空间和更短的窗口时间

对验证技术提出了更高的要求,未来的模拟验证技术发展将朝着多种抽象层次、更高复用程度以及支持片上互连网络的方向发展。

6.1 多种抽象层次的模拟验证

提高模拟验证的效率可以从两个方面进行:提高验证的抽象程度,简化验证平台设计并加快开发速度;加速模拟执行的速度,实现模拟的高速执行。

6.1.1 更高抽象层次

研究者和 EDA 软件供应商致力于提高验证环境的抽象程度的研究,采用分层次的验证架构避免验证平台过于复杂和庞大。通过逐层封装底层硬件特性和细节,提高测试平台的开发速度,并简化验证环境的设计。图 6 所示为高级验证方法学(VMM)的 5 层验证环境,自底向上抽象程度依次提高,分别为信号层、命令层、功能层、场景层和测试层。

开发高抽象程度的验证环境需要比硬件描述语言(HDL)描述能力更强的语言支持。一种思路是增强原有的硬件描述语言的系统级描述能力,出现了硬件验证语言(HDV L, Hardware Design and Verification Language)。HDV L 支持面向对象设计,加快了验证平台的开发速度并缩短了 SoC 的验证时间;另一种思路是扩展传统的高级语言(如 C 和 C++),使它们支持硬件描述,目前最有前景的是 SystemC。在描述系统体系结构、交易级模型以及提供受约束随机激励生成、功能覆盖率和断言等支持验

证环境开发的类库等方面,两类语言各有所长,都有广泛的应用。随着未来 SoC 设计规模和抽象程度的进一步提升,验证环境的开发将更加注重可复用性和平台工具的支持,需要在抽象程度和开发效率之间进行折中并完善现有的验证语言。

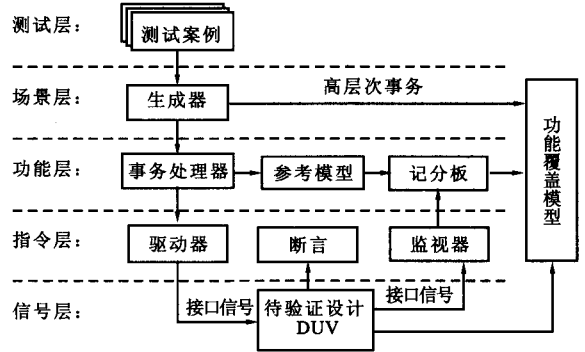


图6 VMM的分层验证环境

6.1.2 硬件加速模拟验证

基于硬件仿真平台(例如现场可编程门阵列 FPGA),将仿真平台进行软硬件划分,并利用硬件加速器执行底层仿真,也将有效加快验证收敛^[35]。传统的硬件加速仅是将 DUV 映射到了硬件平台上,验证平台仍然由软件实现,平台与 DUV 的频繁交互制约了验证速度的提升。如图 7 所示,将分层次验证平台的底层下移到硬件上实现^[36],减少平台与 DUV 之间的交互,可以进一步提高运行速度。

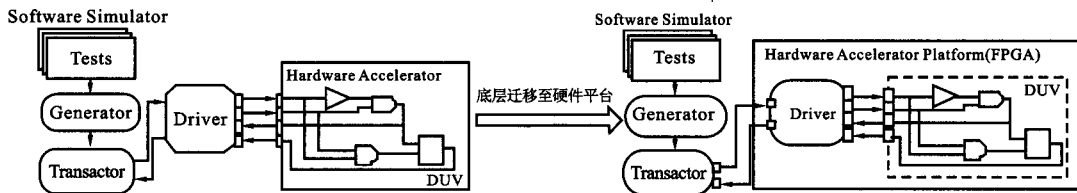


图7 硬件加速仿真平台框图

6.2 验证知识产权核的设计

为了提高验证平台的搭建效率,借鉴基于 IP 核重用的设计方法,对验证平台中的一些功能模块进行重用,这些被重用的模块称为验证知识产权核(VIP, Verification Intellectual Property)。据统计, VIP 技术的出现使得验证平台的开发时间节省了 75% 以上。现有的商用 VIP 多是基于通用总线标准协议设计的,例如 PCI、USB、SPI、AMBA 等。还有一些用于专用接口验证的 VIP,一般随设计 IP 核一起提供,目的是提供给使用者一种对购买的 IP 核进行快速验证的方法。VIP 本身可以是验证平台的一个组件,也可以是一些组件的集合。驱动器、监视器

以及总线功能模型(BFM, Bus Functional Model)是最常见的 VIP,验证人员可以使用这些已经被验证过的组件快速搭建验证平台、创建验证场景、生成测试激励。如何对验证环境功能进行正交化划分,将与 DUV 无关的功能剥离形成独立的 VIP 等问题,将促使研究者们开展 IP 核验证空间和验证组件正交化划分方法学的研究。

6.3 片上互连网络的验证

以总线结构为通信基础的 SoC 受总线传输能力的制约,当片上需要通信的 IP 核数目较多时在性能、功耗、复杂度和可靠性等方面都存在瓶颈。2001 年左右提出了片上网络(NoC, Network-on-a-chip)。

chip),采用基于包交换的片上通信网取代原先的总线结构,每个 IP 核通过各自的网络接口(NI, Network Interface)接入通信网络.由于 NoC 实现了处理功能(IP 核)与通信功能(网络)的分离,即不同 IP 核接入网络时只需实现相应的 NI 即可,同一个 IP 核接入不同网络时也只需修改其 NI,而无需对 IP 核和通信网络做重大修改,提高了设计的可复用性并加快了系统集成速度,成为未来 SoC 的发展方向.但是,目前还很少关于片上通信网络功能验证的研究,支持片上互联网络的验证环境设计将是今后的研究热点.

7 结束语

本文对近年来基于模拟的 SoC 功能验证技术涉及的激励生成、约束定义、覆盖率分析及 IP 核功能验证等关键技术进行了总结和分析.从香农信息熵的角度分析了约束集与激励质量的关系,指出当所有约束的解向量分布最均匀时激励质量最高.根据激励生成过程将其分为两类:①基于学习的激励生成;②基于构造的激励生成.

随着基于 IP 复用的 SoC 设计技术的广泛应用和单芯片集成 IP 核数量的不断上升,未来基于模拟的 SoC 功能验证技术将朝着更高抽象层次、基于 VIP 复用的验证平台设计以及支持片上互联网络的功能验证环境方向发展.

参考文献:

- [1] Nativ G, Mittennaier S, Ur S, Ziv A. Cost evaluation of coverage directed test generation for the IBM mainframe[C]//Proceedings of International Test Conference, Baltimore, 2001: 793—802.
- [2] Bose M, Jongshin Shin, Rudnick E M, et al. A genetic approach to automatic bias generation for biased random instruction generation[C]//Proceedings of the 2001 Congress on Evolutionary Computation. Seoul, 2001: 442—448.
- [3] Samarah A, Habibi A, Tahar S, et al. Automated coverage directed test generation using a cell-based genetic algorithm[C]//Proceedings of the Eleventh Annual IEEE International High-Level Design Validation and Test Workshop. Monterey, 2006: 19—26.
- [4] 易江芳, 佟冬, 程旭. GATEST: 使用遗传算法自动生成模拟矢量的验证平台[J]. 北京大学学报: 自然科学版, 2006, 42(5): 668—673.
- [5] 罗春, 杨军, 凌明. 基于遗传算法和覆盖率驱动的功能验证向量自动生成算法[J]. 应用科学学报, 2005, 23(4): 50—54.
- [6] 沈海华, 王朋宇, 卫文丽, 等. 基于遗传算法的全芯片级覆盖率驱动随机验证技术[J]. 计算机研究与发展, 2009, 46(10): 1612—1625.
- [7] Tasiran S, Fallah F, Chinnery D G, et al. A functional validation technique: biased—random simulation guided by observability—based coverage[C]//Proceedings of 2001 International Conference on Computer Design. Austin, 2001: 82—88.
- [8] Wen CH—P, Wang Li—C, Cheng Kwang—Ting. Simulation—based functional test generation for embedded processors[J]. IEEE Transactions on Computers, 2006, 55(11): 1335—1343.
- [9] Guzey O, Wang L—C, Levitt JR, et al. Increasing the efficiency of simulation—based functional verification through unsupervised support vector analysis [J]. IEEE Transactions on Computer—Aided Design of Integrated Circuits and Systems, 2010, 29(1): 138—148.
- [10] Aharon A, Goodman D, Levinger M, et al. Test program generation for functional verification of powerPC Processors in IBM[C]//Proceedings of the 32nd Annual ACM/IEEE Design Automation Conference. USA: San Francisco, 1995: 279—285.
- [11] Guglielmo DG, Fummi F, Pravadelli G, et al. Semi—formal functional verification by EFSM traversing via NuSMV[C]//Proceedings of 2010 IEEE International High Level Design Validation and Test Workshop. Anaheim, 2010: 58—65.
- [12] Albin K, Yuan J, Aziz A, et al. Constraint synthesis for environment modeling in functional verification[C]//Proceedings of the 47th Design Automation Conference. Anaheim, 2003: 296—299.
- [13] De Paula FM, Hu AJ. An effective guidance strategy for abstraction—guided simulation[C]//Proceedings of the 44th ACM/IEEE Design Automation Conference. San Diego, 2007: 63—68.
- [14] Yuan J, Aziz A, Pixley C, et al. Simplifying boolean constraint solving for random simulation—vector generation[J]. IEEE Transactions on Computer—Aided Design of Integrated Circuits and Systems, 2004, 23(3): 412—420.
- [15] Kukula J H, Shiple T R. Building circuits from relations[C]//Proceedings of the 12nd International Conference on Computer—Aided Verification. Chicago, 2000: 113—123.
- [16] Barrett C W, Dill D L, Levitt J R. Validity checking for combinations of theories with equality[C]//Pro-

- ceedings of the First International Conference on Formal Methods in Computer-Aided Design. Palo Alto, 1996: 187-201.
- [17] Fallah F, Devadas S, Keutzer K. Functional vector generation for HDL models using linear programming and 3-satisfiability[C]// Proceedings of the 35th ACM/IEEE Design Automation Conference. San Francisco, 1998: 528-533.
- [18] Kitchen N, Kuehlmann A. Stimulus generation for constrained random simulation[C]// Proceedings of International Conference on Computer-Aided Design. Lake Tahoe, 2007: 258-265.
- [19] 李瞰, 郭阳, 李思昆. 基于断言的模拟矢量自动生成方法[J], 软件学报, 2004, 15(10): 1441-1450.
- [20] 邓澍军, 吴为民, 边计年. RTL 验证中的混合可满足性求解[J], 计算机辅助设计与图形学学报, 2007, 19(3): 273-278.
- [21] Hari SKS, Konda VVR, Kamakoti V, et al. Automatic constraint based test generation for behavioral HDL models[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008, 16(4): 408-421.
- [22] Yeh H H, Huang Chung yang. Automatic constraint generation for guided random simulation[C]// Proceedings of the 15th Asia and South Pacific Design Automation Conference. Taipei, 2010: 613-618.
- [23] Guzey O, Wang LC. Coverage-directed test generation through automatic constraint extraction[C]// Proceedings of IEEE International High Level Design Validation and Test Workshop. Irvine, 2007: 151-158.
- [24] Dsouza A. Static analysis of deadends in SVA constraints[C]// Proceedings of IEEE International High Level Design Validation and Test Workshop. Anaheim, 2010: 82-89.
- [25] Tasiran K, Keutzer S. Coverage metrics for functional validation of hardware designs[J]. IEEE Design & Test of Computers, 2001, 18(4): 36-45.
- [26] Harris Ian G, Kiran Ramineni, Shireesh Verma. Automatic generation of functional coverage models from behavioral Verilog descriptions[C]// Proceedings of the Conference on Design, Automation and Test in Europe. France: Nice, 2007: 900-905.
- [27] Nan Chien, Liu Jimmy, Chang Chenyi, et al. A novel approach for functional coverage measurement in HDL [C]// Proceedings of IEEE International Symposium on Circuits and Systems. Geneva, 2000: 217-220.
- [28] Zhang Qiushuang, Harris Ian G. A domain coverage metric for the validation of behavioral VHDL descriptions[C]// Proceedings of the 2000 IEEE International Test Conference. Atlantic, 2000: 302.
- [29] Luo Chun, Yang Jun, Shi Longxing. Domain strategy and coverage metric for validation[C]// Proceedings of IEEE International Symposium on Quality Electronic Design. San Jose, 2005: 40-45.
- [30] Harry Forster, Adam Krolnik. Creating assertion-Based IP[M]. New York: Springer Science & Business Media, 2007: 41-57.
- [31] Jerini V, Muller D. Safe integration of parameterized IP[J]. Integration, The VLSI Journal, 2004, 37(4): 193-221.
- [32] Castro Ivan Castro M, Edgar Leonardo Romero T, Marius S, et al. A functional verification methodology based on parameter domains for efficient input stimuli generation and coverage modeling[J]. Journal of Electronic Testing: Theory and Applications, 2011, 27(4): 485-503.
- [33] Shimizu K, Dill D L. Deriving a simulation input generator and a coverage metric from a formal specification [C]// Proceedings of the ACM/IEEE 39th Design Automation Conference. New Orleans, 2002: 801-806.
- [34] Shih Chehua, Huang Juindar, Jou Jingyang. Automatic verification stimulus generation for interface protocols modeled with non-deterministic extended FSM [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(5): 723-727.
- [35] Cadambi S, Mulpuri C S, Ashar P N. A fast, inexpensive and scalable hardware acceleration technique for functional simulation[C]// Proceedings of the 39th ACM/IEEE Design Automation Conference. New Orleans, 2002: 570-575.
- [36] Pal B, Sinha A, Dasgupta P, et al. Hardware accelerated constrained random test generation[J]. Computers & Digital Techniques, 2007, 1(4): 423-433.

作者简介:

姚爱红 女, (1972-), 博士, 副教授. 研究方向为 SoC 设计及验证、可重构计算系统、数字信号处理及应用等。

孙盟哲 男, (1985-), 硕士. 研究方向为 SoC 设计及验证。

袁莉娜 女, (1981-), 硕士, 助理研究员. 研究方向为信号与信息处理、合成孔径雷达成像算法及应用研究。