

基于路径约束求解的多目标状态激励生成方法

周艳红^{1),2)} 王天成^{1),2)} 李华伟¹⁾ 吕 涛¹⁾ 李晓维¹⁾

¹⁾(中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)

²⁾(中国科学院大学 北京 100049)

摘 要 该文提出了一种针对多个难达目标状态的激励生成方法,该方法基于抽象引导的半形式化方法框架.采用一个评估函数对候选状态进行评价,该评估函数综合考虑了从一个状态到不同目标状态的全局抽象距离信息,评价出从各个状态到达某个目标状态或者同时到达多个目标状态的潜能,并以此指导模拟过程直到最终搜索到一条能同时覆盖多个目标状态的状态序列.此外,该文采用了基于路径约束求解的激励生成方式,这种方式结合了具体模拟和符号模拟技术,符号模拟沿着具体执行路径提取分支条件构成路径约束,通过约束条件的翻转与求解能够有效地产生输入向量,以一种较均衡的模式遍历设计的状态空间,帮助验证快速覆盖到目标.实验结果表明,该文方法能够有效地同时验证设计中的多个目标状态.

关键词 激励生成;难达状态;抽象引导模拟;路径约束求解

中图法分类号 TP393 **DOI号** 10.11897/SP.J.1016.2016.01829

Test Generation for Multiple Target States Using Path Constraint Solving

ZHOU Yan-Hong^{1),2)} WANG Tian-Cheng^{1),2)} LI Hua-Wei¹⁾ LV Tao¹⁾ LI Xiao-Wei¹⁾

¹⁾(State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾(University of China Academy of Sciences, Beijing 100049)

Abstract In this paper, we propose a test generation method to cover multiple hard-to-reach states, which operates in an abstraction-guided simulation framework. An evaluation function, which considers the global abstract distance information for one state to different target states, is used to evaluate the potentiality of each candidate next state to targets and guide the simulation processes to cover multiple target states. In addition, a path constraint solving based test generation method which combines concrete simulation and symbolic simulation is used. Symbolic simulation walks in the design following the concrete path and extracts the corresponding symbolic expressions of the branch conditions in the path as path constraints. The test generation engine can generate valid input vectors through constraint mutation and solving, and it helps to search state in the design state space in a balanced manner. Experimental results show that our approach is effective in covering multiple target states.

Keywords test generation; hard-to-reach states; abstraction-guided simulation; path constraint solving

收稿日期:2014-09-28;在线出版日期:2015-07-23. 本课题得到国家自然科学基金(61432017,61176040)、国家“九七三”重点基础研究发展规划项目基金(2011CB302501)资助. 周艳红,女,1987年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为超大规模集成电路(VLSI)半形式化验证. E-mail: zhouyanhong@ict.ac.cn. 王天成,男,1983年生,博士研究生,工程师,主要研究方向为 VLSI 设计验证. 李华伟(通信作者),女,1974年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为 VLSI 测试、设计验证、可靠性设计、容错计算. E-mail: lihuawei@ict.ac.cn. 吕 涛,女,1978年生,博士,主要研究方向为 VLSI 设计、验证和测试. 李晓维,男,1964年生,博士,研究员,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为 VLSI 测试、可靠性设计、设计验证、可信计算及无线传感器网络.

1 引言

随着集成电路设计规模与复杂度的增加,功能验证已经成为设计周期中的一个重要环节.其中,如何高效地产生测试向量来覆盖难达状态是一个相当有难度的问题.目前,在工业设计验证中占主导地位的验证方法仍然是传统的模拟方法,因为模拟验证方法可扩展性好,能够处理大规模的设计并且快速、易用,但是其缺点是不能保证验证的完备性,通常难以覆盖到设计中的难达边缘属性(难达状态).形式化的方法是一种完备的验证方法,理论上可以高效地验证所有的难达状态,但其处理规模有限,可扩展性差,难以应用到大规模的工业设计中.

为了结合模拟方法的可扩展性以及形式化方法的完备性,避免各自的缺陷,研究者们提出了半形式化验证方法.在已有的半形式化验证方法中,抽象引导的模拟验证方法是一个很有前景的研究方向^[1-7].在这种方法里,首先从原始设计上提取出一个较小的抽象模型,然后在这个抽象模型上进行形式化计算,用所得信息来指导模拟的过程.

文献[1]是抽象信息引导的半形式化验证方法的先驱.在这种方法中,作者将与目标状态寄存器紧密相关的模块(module)提取出来构成抽象模型,并在此模型上应用形式化的前像计算获得抽象距离信息.在模拟过程中,从当前状态开始,随机产生 N 组长度为 M 的输入向量进行模拟,并且利用抽象距离信息对模拟过程中访问到的状态进行评估,从中选择抽象距离最小的状态作为下一个模拟的当前状态,循环这个过程直到覆盖到目标.这种方法采用随机生成激励的策略,工作效率不高,而且抽象的粒度较粗,所得的抽象距离信息不够准确,容易陷入死胡同状态(dead-end states).为了缓解由抽象误差导致的死胡同问题,文献[2]引入了抽象细化技术,这种方法首先通过前后像的计算获得一条抽象状态序列,相对于其中的每一步,进行随机激励生成,将抽象序列具体化到真实序列;如果随机无法满足,则进行抽象细化,重新选取更加精细的抽象模型,重复整个过程.这种方法在一定程度上提高了验证效率,但反复抽象细化的计算成本太高,方法的适用性受限.因此,文献[3-7]主要从抽象方法、抽象引导策略以及激励生成方法 3 个方面进行改进,以提高验证工作效率.

本文的目标是针对设计中多个难达状态产生输

入激励,通过多目标同时覆盖的方式来提高验证效率.首先,本文通过构建一个评估函数来对验证过程中的状态进行评价,该评估函数同时考虑了从一个状态到不同目标状态的抽象距离值,用来指导状态的选择及模拟过程,直到最终搜索到一条能同时覆盖多个目标状态的状态序列.其次,本文采用基于路径约束求解的策略产生输入激励,这种激励生成引擎结合了具体模拟和符号模拟技术,能够产生有效地测试向量沿着具体路径遍历设计的状态空间.此外,本文采取了一种较细粒度的电路抽象方法针对每一个目标状态来构建抽象模型,在这些抽象模型上进行形式化计算能获得较准确的抽象距离信息来计算评估函数值.

本文第 2 节介绍抽象引导的半形式化方法及相关工作;第 3 节描述本文方法的框架及相关细节;第 4 节给出本文方法的实验结果;第 5 节对本文方法的局限性进行说明;最后一节对本文进行总结.

2 相关工作

2.1 抽象引导的半形式化方法简介

抽象引导的半形式化验证方法被认为是众多的半形式化验证方法中最具潜力的方法之一.在这种方法里,通常从具体设计中抽象出一个较小的抽象模型,以便能够在这个抽象模型中进行完全的形式化计算,并用所得的抽象距离信息来评估具体设计状态,指导模拟过程趋向验证目标,示意图如图 1 所示^[8].

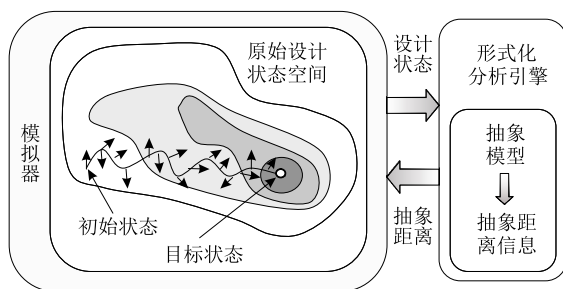


图 1 抽象信息引导的半形式化方法示意图^[8]

在形式化分析引擎中,通过对抽象设计进行形式化的前像计算可以将抽象状态划分到不同的“洋葱环”中,其中第 i 个环中的状态在第 i 个模拟周期时能到达抽象目标(即其抽象距离为 i)。一个抽象状态对应多个具体状态,相应地,每一个具体设计状态对应一个抽象状态并有一个特定的抽象距离值.如图 1 中所示,原始设计状态空间中不同灰度区域内的状态对应不同的抽象距离值.在模拟过程中,从

初始状态开始,向前搜索一步或几步,模拟引擎将模拟过程中的具体设计状态送入形式化分析引擎中,利用抽象距离信息对这些状态进行评估,从中选出对应抽象距离最小的状态作为下一步模拟的开始状态,向前进行搜索,直到到达目标.

2.2 各种抽象引导的半形式化方法

由于抽象设计和原始设计之间存在抽象误差,即在抽象过程中损失了许多原始设计的信息,因此不可避免地会产生死胡同状态.死胡同状态是指那些在抽象模型上评估为距离目标很近,但实际上却距离目标很远甚至完全不能到达目标的状态^[5].

为了缓解死胡同问题,文献[3]从抽象引导策略方面进行改进.该方法以“桶”的形式将模拟过程中访问过的所有状态按照抽象距离远近顺序保存起来,其中一个“桶”对应一个抽象距离.在每个模拟周期中,从最近的“桶”(最小抽象距离)开始,按照抽象距离从小到大概率指数衰减的原则,选择一个已达的历史状态作为新的模拟当前态.这种方法能较好地避免死胡同问题,提高验证效率,但是随机的激励生成策略使得其仍然难以覆盖到难达状态.针对这个问题,文献[4]引入了文化算法(Cultural Algorithm)来实现激励生成.该方法利用数据挖掘技术提取抽象电路,将获得的抽象设计信息作为文化算法中的适应度函数,通过交叉、遗传、变异、继承等操作,不断提高激励的适应度值,生成能够快速覆盖到目标的输入激励.这种方法能够自动演进覆盖目标状态,进一步提高验证的效率,其缺点是需要进行大量的随机以积累数据挖掘所需的信息,并且这种数据挖掘是基于位(bit)级的操作,所提供的抽象信息很难用来处理宽位宽的复杂设计.

为了进一步提高抽象引导的半形式化验证方法的效率,文献[5]利用机器学习中的 Markov 模型建立激励生成引擎.这种方法首先建立设计的数据依赖图(Data Dependence Graph, DDG),然后基于数据依赖图进行设计抽象,获取抽象信息;根据设计规范(Specification)建立 Markov 模型,自动化的为每个模拟周期生成合法的输入向量;利用抽象信息对 Markov 模型进行调整,提高激励生成质量.这种方法能够产生高质量的测试激励,加速验证收敛速度,但 Markov 模型是一种随机的概率模型,因此仍面临死胡同状态问题.文献[6]则通过在抽象设计上构建一个 Markov 模型,在此模型上计算各个状态之间相互到达的稳态概率分布,并以此来判断状态的难达程度及引导激励生成的过程,这种方法需要显

式的建立抽象设计的状态转移图,因此能处理的设计规模有限.

在抽象引导的半形式化方法中,随机产生输入向量是在设计的输入空间内搜索,对应到设计的状态空间中,每个状态被遍历到的概率相差较大.例如,在设计中有一个分支为:if (a) $state = A$; else $state = B$,其中 a 是一个四位的原始输入,那么在输入空间内, a 的每一种取值被随机到的概率都是 $1/16$,但从状态空间来看,状态 A 和 B 被遍历到的概率分别是 $15/16$ 和 $1/16$,如果 B 是离目标较近的状态,那么随机产生激励的方法将比较难达到目标.针对这一问题,文献[7]提出了一种基于路径约束求解的难达状态激励生成方法——PACOST. PACOST 在抽象引导的模拟验证方法的激励生成中使用形式化的约束求解来产生激励,在每一个模拟步骤内,在动态模拟开辟的状态空间内,进行静态的符号模拟以提取路径约束,通过路径约束条件的翻转求解产生输入激励,在状态空间内搜索抽象距离最近的状态作为下一当前态,重复这个过程直到覆盖到目标状态.这种方法能够以一种较均衡的方式遍历设计的内部状态空间,在抽象距离的引导下快速地覆盖到目标状态,提高验证的效率.

上述已有的抽象引导的半形式化方法基本上都是针对单个目标状态来产生激励.目前,在国内外针对多个难达目标状态进行激励生成的研究工作较少.文献[9]提出了一种利用抽象模型上的后验概率值来指导多目标状态激励生成的方法.这种方法首先从原始设计中抽取一个抽象模型,然后在抽象模型对应的显式状态机上计算稳态概率分布矩阵,该矩阵记录了抽象模型上各个状态之间跳转转移的概率值,然后通过矩阵的逆运算计算出各个状态到达目标状态的后验概率值,后验概率值越大表示相应的抽象状态到目标状态越近.对于多个目标状态的情况,通过计算各个状态到达不同目标状态的后验概率值的几何平均值来评估候选状态.由于这个方法的后验概率值是在显式的抽象状态机上计算的,因此能处理的抽象模型的规模有限,而且当原始设计规模较大,抽象模型与原始设计差异较大时,计算出来的后验概率值存在误差,不能提供很好的指导.

专利^①提出了一种利用符号约束求解为多个覆盖率漏洞目标产生确定性的输入激励.在每个模拟步骤中,该方法通过分析覆盖率信息来确定一个覆

① <http://www.google.com/patents/US20120266118>

盖率漏洞作为目标. 对于每个目标,通过具体模拟路径中符号变量和表达式的传播获取符号约束,约束求解产生激励覆盖相应目标. 该方法的覆盖目标是在模拟的过程中确定的,一次只能覆盖一个目标,并且这个方法中所收集的符号约束取决于随机模拟遍历到的路径,如果覆盖目标所需激发的条件与所收集的符号约束关系不大,则这种方法难以产生出覆盖目标的输入激励.

本文方法是针对确定性的多个目标,在 PACOST 基础上进行扩展,目的是同时覆盖设计中的多个难达目标状态. 在每一步模拟中,提取一个周期的路径约束,通过约束条件的翻转与求解产生新的输入激励,遍历不同的分支路径并相应获得不同的候选状态,利用一个评估函数综合考虑各个候选状态到不同目标状态的全局抽象距离信息,逐步地指导模拟过程,最终同时覆盖到时序深度较深的多个难达状态.

3 本文方法

本文所提出的针对多个目标状态的激励生成方法的整体工作框架如图 2 所示,本文方法基于抽象引导的半形式化方法框架,包括 4 个主要的模块:抽象引擎,评估函数,基于路径约束求解的激励生成器以及状态选择器. 抽象引擎从原始设计中提取出规模较小的抽象模型并在此模型上应用形式化方法获取抽象距离信息,由于本文是针对多个目标状态,如果各目标状态所涉及的寄存器不同,则相对于每一个目标状态提取出一个抽象模型并获得一个抽象距离信息表,该表中记录了在此抽象模型上每个抽象状态到该目标状态的抽象距离值;

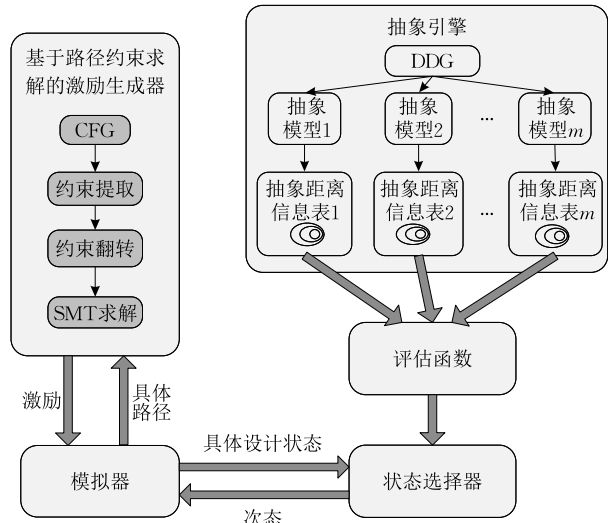


图 2 本文方法的工作框架

评估函数同时考虑了从一个具体设计状态到不同目标状态的抽象距离值,为每一个状态计算一个综合的抽象距离值来评价该状态到达某一个或多个目标状态的潜能,用来指导状态的选择;基于路径约束求解的激励生成器结合具体模拟和符号模拟方法来产生有效的输入激励,其中每个激励对应于一个候选状态;状态选择器则利用评估函数计算出来的综合抽象距离信息来指导整个状态搜索过程的方向,在每一个模拟步骤中,从路径约束求解产生的一组候选状态中选择一个评估函数值最大的作为下一步模拟的当前态. 下面将对各个模块进行更详细的描述.

3.1 抽象引擎

抽象引擎的功能是实现设计抽象、获取抽象距离信息. 在原始设计上提取出的规模较小的抽象模型上,运用可达性分析方法计算抽象距离,用来指导具体的模拟过程. 因此抽象模型的质量至关重要,如果抽象模型过于粗糙,则不能很好地指导模拟. 模型越细越好,但如果模型过于精细,则其规模会超出形式化方法能够处理的范围. 本文中,我们采用类似于文献[5]中所提的基于 DDG 的较细粒度的电路抽象方法. DDG 描述了数据在设计中的寄存器之间的流向关系,其中节点表示设计中的寄存器,边表示从目的节点到源节点存在的数据依赖关系. 基于 DDG 的抽象方法的示意图如图 3 所示,图中 R1、R2、R3 表示目标状态寄存器. 不同于文献[5]中的抽象方法,本文抽象过程中更关注控制逻辑中的寄存器. DDG 中的寄存器可以分为控制寄存器和数据寄存器,控制寄存器是指位于控制逻辑中的寄存器(控制逻辑通常包括设计中的显式状态机、分支、条件等,在设计中起着关键作用),数据寄存器是指位于数据通路(数据流相关的逻辑,如 b12 中的存储单元,浮点运算单元的操作数和结果等)中的寄存器. 控制寄存器与数据寄存器的区分在大多数硬件设计中都比较明显.

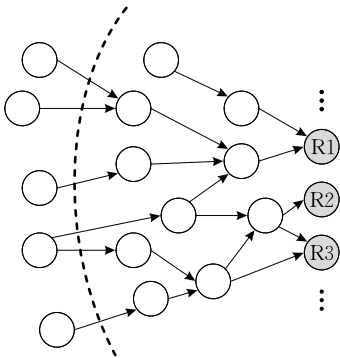


图 3 基于 DDG 的抽象方法示意图

在抽象的过程中,首先需要确定目标状态寄存器.本文方法中,目标状态寄存器是与难达分支(时序深度较深的分支)相对应的设计状态的寄存器变量.然后从原始设计建立与目标状态变量相关的数据依赖图.抽象从目标状态寄存器开始,基于设计的数据依赖图逆向地选取控制寄存器,也就是按照距离目标状态寄存器远近程度由近及远地按照宽度优先的方法进行选取.由于目标状态寄存器的逻辑是由从原始输入到目标状态寄存器之间的逻辑构成,如果从目标状态寄存器往前抽取的逻辑越多,获得的抽象模型越复杂,则抽象模型和原始设计之间存在的差异越小,但是由于状态爆炸的问题,抽象模型的规模最终受限于形式化工具的处理能力.本文的抽象方法是按照基于 DDG 的抽象原则抽取与目标寄存器相关的扇入锥中的控制寄存器,而数据寄存器则考虑作为抽象模型的输入或输出.当抽象模型的规模超过形式化工具处理的上限时,原始设计 DDG 中离目标状态寄存器较远的控制寄存器将优先从抽象模型中移除.当抽象模型建立后,在此模型上应用形式化工具来计算抽象距离信息,并保存为抽象距离信息表.本文所采用的形式化工具为 VIS^①.

本文抽象方法能够抽取对路径约束而言重要的寄存器,因为路径约束是由设计中的分支条件表达式构成的,而分支条件是控制逻辑的一部分,本文的抽象方法是从与目标属性相关的寄存器开始,往前抽取控制逻辑的寄存器,这样,与目标属性相关的路径约束涉及的寄存器基本上包括在了抽象模型里.由于抽取出了与待验证目标属性最相关的那一部分控制逻辑,因此可以得到比较精确的抽象模型从而为模拟过程提供精确的指导.

由于本文的目的是同时覆盖设计中的多个目标状态,如果各目标状态所涉及的寄存器变量不同,则会提取出多个抽象模型,相应会获得多个抽象距离信息表,即一个具体状态在每个抽象模型上都有一个抽象距离值,表示该状态到此抽象模型上的目标状态的距离信息.如果各目标状态所涉及的寄存器变量相同,则可以共用一个抽象模型.

用一个简单的例子来说明,原始设计的 Verilog 描述及其对应的 DDG 如图 4 所示.

首先,对设计进行分析以确定目标状态.此处,我们选择该设计中时序深度较深的两个状态 $state=S3$ 和 $state=S4$ 为目标状态加以说明.由于这两个目标状态涉及的寄存器变量相同,都是 $state$,因此提取出来的抽象模型也相同,所提取的抽象模型及相应

的抽象距离信息如图 5 所示.由图中可以看出,针对不同的目标状态,每一个抽象状态都具有不同的抽象距离值.

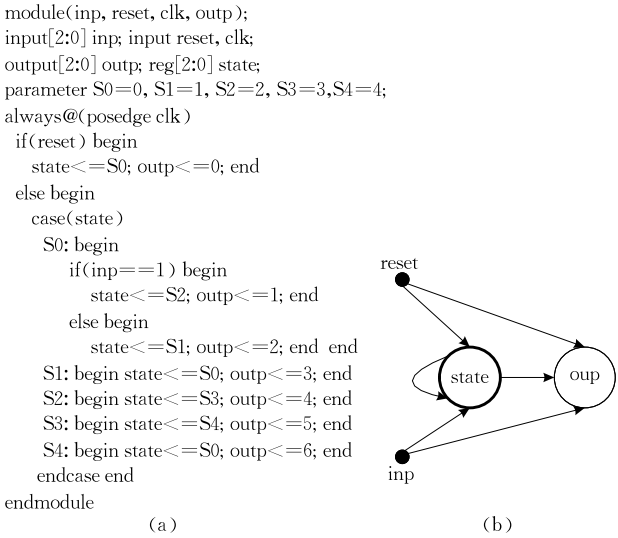


图 4 一个 RTL 设计示例(a)及相应 DDG(b)

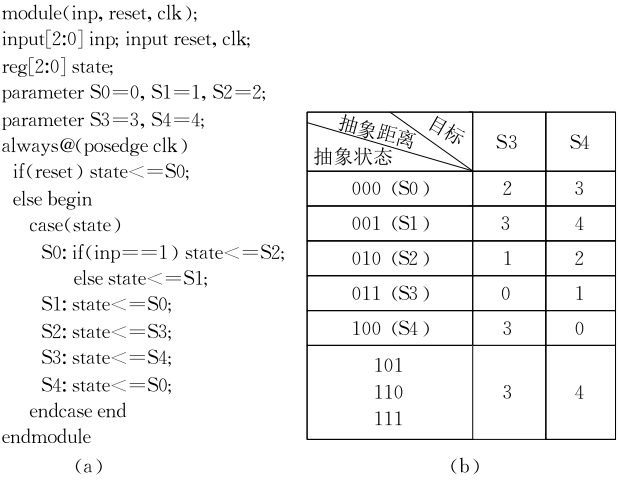


图 5 抽象模型(a)及抽象距离信息(b)

3.2 评估函数

本文多目标覆盖方法的动机之一是在我们在实验中观察到多个难达状态是很可能位于同一条路径中的.

难达状态的量化定义是功能验证中的另一个难题.随着集成电路设计复杂度的提高,对设计进行功能验证的难度也不断提高.在功能验证中,通常很难达到 100% 的分支覆盖率,因此会存在分支覆盖漏洞(未被覆盖到的分支).覆盖漏洞对应的状态要被遍历到通常需要激发一些难以覆盖到的分支条件.这些分支条件一般具有“低概率到达”或者“较深的时序深度(需要很长的输入激励序列)”的特点.本文

① VIS Homepage. <http://vlsi.colorado.edu/~vis/>

将这些分支条件下对应的状态作为待验证的难达目标状态. 因此, 如果一个状态被确定为难达状态, 那么它的后继状态或邻近的同等时序深度的状态也同样可以认为是难达状态. 以图 4(a) 所给的示例设计为例, 假定认为分支条件 $\text{if}(inp = 1)$ 是一种低概率到达条件, 那么其对应的状态 S2 被认为是难达状态, 相应的 S2 的后继状态 S3 和 S4 同样也被认为是难达状态. 从这个角度来说, 难达状态是很可能相互靠近或处于同一条路径上.

因此, 本文定义了以下评估函数来量化地引导激励生成能够更有效地同时覆盖多个目标状态.

对于每一个设计状态, 评估函数根据其对应到不同目标状态的抽象距离定义一个综合的抽象距离值, 命名为“*merge_dis*”. *merge_dis* 的计算公式为

$$\text{merge_dis} = \sum_{i=1}^n C_i \times 2^{-d_i} \quad (1)$$

其中: n 表示目标状态的数目; d_i 表示从当前状态到目标状态 i 的抽象距离. 如果目标状态 i 还没有被覆盖到, 则 $C_i = 1$, 否则 $C_i = 0$. *merge_dis* 的值度量了每个状态到各个目标状态的潜在距离信息, 如果该状态距离目标状态 i 越近, 则对应的 d_i 值越小, 在其他项保持不变的情况下, 所计算出的 *merge_dis* 的值越大, 表示从该状态继续向前模拟将更可能快速地到达其中一个或多个目标状态. 简而言之, *merge_dis* 的值越大, 则表示相应的状态距离某个(或某几个)目标状态越近. *merge_dis* 的值将在状态选择器中用来评估模拟过程中的状态, 试图最终能搜索出一条同时覆盖多个目标状态的路径序列.

本文提出评估函数的宗旨是综合考虑各个状态到不同目标状态的全局抽象距离信息, 来评估各个状态到目标状态的难易程度. 在当前状态的各个后继状态中, 如果其中一个后继状态离某个目标很近或者离各个目标状态都比较近时, 我们认为选择这个状态更容易到达目标, 因此更倾向于选择这个状态继续向前模拟. 评估函数的计算式(1)给出的的是一个经验公式, 其值可以反映一个状态到不同目标状态的难易程度的趋势. 如果一个状态到目标状态越远, 则这个状态相应于该目标的抽象距离值 d_i 越大, 那么可以认为从这个状态到该目标的概率越小, 这种趋势可以用 2^{-d_i} 表示出来, 其中 2 是一个可变的值(可以取大于 1 的其他数). 当有多个目标状态时, 我们更倾向于从后继状态中选择离各个目标状态相对都比较近或者离其中一个目标状态最近的状态向前模拟. 当目标状态 i 到达后, 通过设置其对应

参数 $C_i = 0$ 来屏蔽该目标相应的抽象距离信息对后续模拟过程的指导产生干扰作用. 式(1)中的评估函数能较好地地区分出多目标情况下不同后继状态的“优劣”程度.

举例说明如下:

假设两个目标状态为 $t1$ 和 $t2$, 当前状态的两个后继状态为 A 和 B , 考虑以下 3 种情况:

(1) 假定 A 相对于目标状态 $t1$ 和 $t2$ 的抽象距离分别为 2、3; B 相对于目标 $t1$ 和 $t2$ 的抽象距离分别为 3、4, 即状态 A 离两个目标状态的距离都比 B 近, 那么这种情况下我们认为状态 A 优于 B , 应优先选择 A 状态作为下一步模拟的开始状态;

(2) 假定 A 相对于目标状态 $t1$ 和 $t2$ 的抽象距离分别为 2、2; B 相对于目标 $t1$ 和 $t2$ 的抽象距离分别为 2、3, 那么这种情况下我们认为状态 A 优于 B , 选择 A 状态将更有潜力快速覆盖两个目标状态;

(3) 假定 A 相对于目标状态 $t1$ 和 $t2$ 的抽象距离分别为 1、4; B 相对于目标 $t1$ 和 $t2$ 的抽象距离分别为 2、3, 那么这种情况下我们也认为状态 A 优于 B , 能更快到达其中一个目标.

在式(1)中, 我们可以用 2^{-d_i} 表示出从各个状态到达目标状态 i 的难易程度(概率趋势), 那么当有多个目标状态时可以用这个概率趋势的算术求和方式来评估全局的情况, 这种方式也能较好地地区分上述几种情况下各个后继状态的优劣, 指导状态的选择. 尤其当目标状态在同一条路径上时这种方式效果很明显, 即 $t1$ 和 $t2$ 在同一条路径上时, 如果状态 A 到目标 $t1$ 的距离比状态 B 到 $t1$ 的距离近, 那么 A 离目标 $t2$ 也会比 B 近, 也就是对应上述 3 种情况中的第 1 种情况, 这种情况下显然应该优先选择 A 状态继续向前模拟.

综上所述, 本文提出的评估函数能够对多目标覆盖方法提供合理的评估与指导.

3.3 基于路径约束求解的激励生成器

本文采用 PACOST 中提出的基于路径约束求解的激励生成方式, 这种方式结合了具体模拟和符号模拟技术, 能够有效地产生输入向量.

符号模拟方法是一种典型的半形式化验证方法, 主要被应用于门级的测试验证中. 其基本思想是用输入的布尔变量代替布尔常量对电路设计进行模拟, 将模拟验证从布尔值扩展到符号领域. 由于采用变量的符号值代替具体值进行模拟, 因此一次符号模拟相当于模拟了一个很大的测试集, 可以在一个模拟周期内完成传统模拟方法多个周期才能完

成的任务。寄存器传输级(Register Transfer Level, RTL)的符号模拟是通过设计代码进行静态分析来实现。文献[10-11]中成功地将符号模拟和具体模拟相结合,为 RTL 设计验证产生有效的输入向量,从而达到较高的覆盖率。本文方法中,符号模拟将沿着具体模拟路径,提取其中分支结点的符号表达式组成路径约束。

路径约束是一个布尔表达式,由分支条件的符号表达式的逻辑合取构成。设计的控制流程图(Control Flow Graph, CFG)被建立以辅助约束提取的过程。在每个模拟步骤内,首先应用一组随机的输入向量对设计进行模拟,在 CFG 中记录相应的具体路径。符号模拟沿着 CFG 中的这条具体路径将每个分支结点的符号表达式提取出来作为约束条件(guard),此路径上所有的约束条件的合取构成相应的路径约束。HYBRO^[11]是一次性将设计展开固定的拍数,首先对设计进行特定周期数目的随机模拟,从中提取出路径约束,通过约束条件的翻转来覆盖新的分支。它使用分支覆盖率作为度量。由于路径爆炸的问题,HYBRO 能够展开的周期数目很有限,不能覆盖到时序深度较深的难达状态。针对这个问题, PACOST 提出了结合具体模拟和符号模拟的基于路径约束求解的难达状态激励生成方法,在每一个模拟步骤中只提取一个周期的路径约束,利用抽象距离信息逐步地指导模拟过程,通过连续的模拟步骤在更大的周期数目上对设计的状态空间进行探索,最终覆盖到时序深度较深的难达状态。本文采用的激励产生方法与 PACOST 中的一致,但目标是同时覆盖设计中的多个难达状态。在每一步模拟中,本文也只提取一个周期的路径约束,通过约束条件的翻转与求解产生新的候选状态,然后利用评估函数对各个候选状态进行评价,指导状态的选择及模拟过程,直到最终同时覆盖到时序深度较深的多个难达状态。

路径约束是由具体路径上所有控制条件(约束条件)表达式的合取构成。与 PACOST 中一致,本文方法中,路径约束主要包括与原始输入直接相关的分支控制条件。路径约束的提取包括以下 3 个步骤: ①在设计中添加计数器并构建 CFG; ②单周期的具体模拟; ③符号模拟提取路径约束。下面我们将分别具体介绍这几个步骤。

(1) 在设计中添加计数器并构建 CFG

RTL 的符号模拟是通过设计代码进行分析,构建设计的 CFG 来实现的。CFG 从设计的 VRM

模型(verilog RTL model)中提取,其中,一个结点对应一条设计语句。对应每一个 CFG 结点,用一个链表来保存其数据流图(Data Flow Graph, DFG)信息,表尾节点对应语句表达式。计数器的赋值语句也作为 CFG 的一部分。为了在 CFG 中记录具体路径,我们使用 perl 脚本在设计中的每一个分支条件语句后面加入计数器,作为具体模拟和符号模拟的通道。每个计数器变量维持一个数组,该数组以时钟周期数为下标。在一拍随机模拟后,将计数器的值与上一拍的值进行对比,如果值有增加,表示相应的分支被执行了,符号模拟则提取出该分支条件,否则将该分支条件取反后加入路径约束中。

图 4(a)所给示例设计添加计数器后的设计及相应 CFG 如图 6 所示。图 6(a)中带下划线的赋值语句表示所添加的计数器,对应于图 6(b)中的节点 $i[n]$ 。

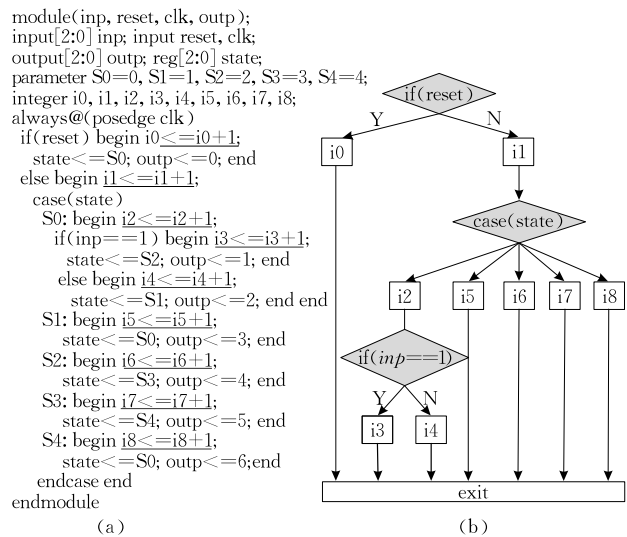


图 6 添加计数器后的示例设计(图(a))及相应 CFG(图(b))

(2) 单周期的具体模拟

在每个模拟步骤中,首先产生一拍随机输入向量,经过具体设计上的模拟之后,在 CFG 上标记出相应的具体执行路径。对于图 4(a)中的设计,假设当前状态为 $state = S0$,随机产生的输入激励为 $reset = 0$, $inp = 101$,那么模拟后得到的具体执行路径如图 7(a)中加粗线条所示。

(3) 符号模拟提取路径约束

CFG 中标记出来的具体路径被符号模拟并提取出路径中相应的控制条件的表达式作为路径约束条件。这些约束条件被存入一个约束栈中,路径上所有约束条件表达式的合取构成了路径约束。对于图 7(a)中所示的具体路径,从中提取出的符号表达

式为 $reset=0 \wedge state=S0 \wedge !(inp==1)$.

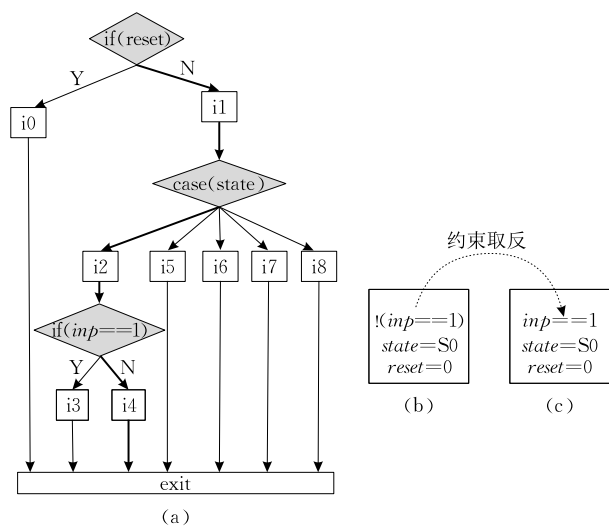


图7 CFG中的具体路径(图(a))及相应的路径约束提取(图(b)、(c))

每次改变(或取反)一个约束条件,可以获得一条新的路径约束,然后用约束求解器进行求解,如果该约束是可满足的,则能求解得到一拍输入向量(不在约束条件中的原始输入变量的值随机生成),回到当前状态进行模拟后得到一个候选状态.已经取反过的约束条件将从约束栈中移除,其对应变量的值将随机产生.候选状态的数目与约束栈中约束条件的数目相同.通过约束求解产生的候选状态通常能对应设计状态空间中的不同状态.

需要注意的是本文方法在每一个模拟步骤中只提取了单个周期内的路径约束,如果一个约束条件是由中间寄存器定义的,则放弃对这个约束条件的取反并保持它在上一个模拟周期中的值.事实上这些约束条件仍然是有机会改变的,通过回退到其中间寄存器定义的地方,回退策略将在3.4节中详细介绍.

目前商业界应用中比较先进的约束随机模拟验证是在随机模拟过程中施加一定的约束,使其按照约束生成随机化的激励,即让它生成的随机化的激励更多地落在我们感兴趣的区域或者边界内,以更快地达到功能覆盖率的要求^[12].随机种子值和约束被送入约束求解器(集成在模拟器中)求解产生合法的输入激励.在约束随机模拟方法中,约束通常是对输入变量取值范围或区间分布概率的约束,或者以断言(System Verilog Assertion, SVA)或属性(Property Specification Language, PSL)的形式定义对某些中间变量的约束,这些约束需要验证人员对设计非常了解之后,以人工提取功能约束的方式添加到测试平台中,作为测试平台的一部分.

不同于约束随机模拟的方法,本文方法是一种抽象引导的半形式化方法,可以自动生成确定性的输入激励以覆盖设计中的难达状态.本文中的约束是由具体执行路径上所有控制条件(约束条件)表达式的合取构成,是在模拟的过程中自动提取的,并不是测试平台的一部分,测试平台只是接受约束求解(约束求解器独立于模拟器)产生的输入激励到待验证设计(Design Under Verification, DUV)中进行模拟.约束提取的过程如前所述,首先在DUV上进行随机模拟,在CFG中记录具体的执行路径,符号模拟沿着具体路径提取分支条件构成路径约束.通过约束条件的翻转与求解产生新的输入激励,传递给测试平台进行模拟后遍历到新的路径分支,提供更多更优秀的候选状态.然后利用抽象距离信息对这些状态进行评估选择,指导模拟过程.

3.4 状态选择器

状态选择器利用评估函数计算出的综合抽象距离值 $merge_dis$ 来评估候选状态,并以此来逐步引导搜索过程到达所有目标.评估函数同时考虑了从一个状态到不同目标状态的抽象距离值,评价出从各个状态到达某个目标状态或者同时到达多个目标状态的潜能.在当前状态的各个后继状态中,如果其中一个后继状态离各个目标状态都比较近或者离某个目标很近时,评估函数计算出的该状态对应的 $merge_dis$ 值会比其他后继状态的大,表示从该状态出发继续模拟会更容易到达目标.因此,在每一个模拟步骤中,状态选择器从约束求解产生的几个候选状态中选择一个评估函数值最大的作为下一步模拟的当前态.

在每一步模拟开始时,选定当前状态(第一拍为reset之后的状态),产生一拍随机向量,在具体设计上模拟后得到一个候选状态,并在设计的CFG中记录具体执行路径,然后采用符号模拟方法从具体路径中提取路径约束.每次改变一个约束条件,通过求解器求解之后得出一拍输入向量;设计状态回到当前状态,模拟后产生一个新的候选状态,如果有 N 个约束条件则循环 N 次,共得到 $N+1$ 个候选状态.然后利用 $merge_dis$ 值对这些候选状态进行评估,从中选择 $merge_dis$ 值最大并未被访问过的状态作为下一当前态.当某个目标状态被覆盖到后,如果这个目标状态的 $merge_dis$ 值大于初始状态的(说明从已到达的目标状态到下一个目标状态之间有比从初始状态开始的一条更短的路径,即这两个目标状态位于同路径中,或者存在从已到达的目标

状态返回某个中间状态的转移边,而从该中间状态到下一个目标会比从初始状态到目标的路径更短),则模拟过程将从这个已到达的目标状态继续进行;否则说明当前已到达的目标状态与其他未到达的目标状态是处于从初始状态开始分叉的不同路径上,因此,将从初始状态重新开始模拟一条新的路径以覆盖未被覆盖的其他目标状态.重复整个过程直到覆盖到所有的目标状态或到达设定的模拟时间上限.

表 1 示例设计图 4(a)的激励生成过程

模拟步骤	当前态/ <i>merge_dis</i>	随机输入	具体路径	路径约束	新的约束	新的输入	候选状态/ <i>merge_dis</i>	次态
1	0/0	010	i0	Null	Null	Null	S0/0.375	S0
2	S0/0.375	101	i1-i2-i4	!(<i>inp</i> ==1)	<i>inp</i> ==1	001	S1/0.1875,S2/0.75	S2
3	S2/0.75	011	i1-i6	Null	Null	Null	S3/0.5,S0/0.375	S3
4	S3/0.5	000	i1-i7	Null	Null	Null	S4/1	S4

从表 1 中可以看出,通过对路径约束中的条件取反,求解后可以获得一个新的输入激励,回到当前态模拟后能遍历到一个之前没有遍历到的新状态.在这个例子中,只在第②个模拟步骤中有提取出路径约束.该模拟步的开始状态是 S0,首先随机产生一拍输入向量(101),模拟后获得一个候选状态 S1,并在 CFG 图中记录具体执行路径 i1-i2-i4,如图 7(a)中所示.符号模拟沿着具体路径提取出路径约束!(*inp*==1),然后通过约束条件的翻转(取反)可以获得新的路径约束(*inp*==1),调用 SMT (Satisfiability Modulo Theory)约束求解器求解后获得新的输入激励(001),最后回到该模拟步的开始状态 S0 重新用新的激励模拟后得到另一个候选状态 S2.在每一个模拟步骤中,用图 5(b)中的抽象距离信息及式(1)所示的 *merge_dis* 值计算方法来评估候选状态,从中选择 *merge_dis* 值最大的候选状态作为次态(即下一步模拟的当前态).在第②步模拟获得的两个候选状态 S1 和 S2 中,S2 的 *merge_dis* 值更大,说明选择 S2 作为下一步模拟的开始状态将能更快的到达目标状态,因此,S2 被选为第③步模拟的开始状态.在第③个模拟步骤中,目标状态 S3 到达,因此需要比较 S3 与 reset 状态 S0 的 *merge_dis* 值,其中值较大的表示相应状态离下一个目标状态较近,被选为下一步模拟的当前态.最终获得的能同时覆盖目标状态 S3 和 S4 的状态序列为 S0-S2-S3-S4,对应产生的激励序列为{010,001,011,000}.

在这个例子中,纯随机的方法很容易陷入由状态 S0 和 S1 构成的循环之中,因为在模拟步②中从状态 S0 到状态 S1 的转移概率很大(7/8).从这个例

3.4.1 示例:图 4(a)的多目标激励生成过程

我们利用图 4(a)中所给的设计来说明本文方法的激励生成过程,如表 1 所示.简单起见,此处只考虑两个目标状态 *state*=S3 和 *state*=S4 作为示例.假定设计的初始状态为 *state*=0,*reset* 信号在第一个模拟周期中取值为 1,在后续的模拟周期中取值为 0.为避免约束求解冲突,我们没有提取 case 语句中的条件作为约束条件,此外,*reset* 信号也未被提取为约束条件.

子可以看出,当目标状态相互靠近或在同一条路径(trace)中时,采用本文多目标同时覆盖的方法可以很快覆盖到多个目标状态.

3.4.2 死胡同状态的回避策略

与其他抽象引导的半形式方法一样,本文方法面临的一个重要问题是死胡同状态问题.由于要对设计进行抽象,抽象模型与原始设计之间存在抽象误差,因此死胡同问题不可避免.抽象实际上是对设计状态进行合并,原始设计中的多个状态在抽象模型上被合并为一个状态,原始设计中原本不存在转移关系的两个状态在抽象模型中可能会引入对应的转移关系,因此导致了死胡同问题的出现.由于抽象模型上虚假边的引入,使得具体设计中离目标很远的某些状态在抽象模型上被评估为距离目标很近,从而在模拟的过程中提供不正确的指导,导致最终不能搜索出一条到达目标状态的路径.为了避免搜索过程陷入死胡同状态或者局部循环中,我们采用了两种回退策略,通过在每个模拟步骤内尝试不同的新状态来缓解死胡同问题:

(1)回退策略 1.不对比当前状态与候选状态的 *merge_dis* 值,在候选状态中选择具有最大 *merge_dis* 值并且没有被访问过的状态作为下一步模拟的开始状态,如果其中 *merge_dis* 值最大的状态有多个则随机选择其中一个;

(2)回退策略 2.比较当前状态的 *merge_dis* 值与候选状态中的最大 *merge_dis* 值,如果候选状态中最大的 *merge_dis* 值都不比当前状态的大,则返回到当前状态重新生成候选状态(本文方法中设定重复次数最多为 5 次).

本文中的回退策略与 PACOST 中的类似,但也有不同之处,本文中比较的是当前状态与候选状态的 *merge_dis* 值,而 PACOST 中比较的是抽象距离值.

4 实验和分析

为了验证本文提出的多目标覆盖方法的效果,作者使用 C++ 编程语言、直接编程接口 (Direct Programming Interface, DPI)、VCS 模拟器以及 CVC3^① 约束求解器进行实验. 实验对象包括: (1) ITC99 基准测试集^②中的部分电路; (2) NandFlash^③, 一款开源 SoC 芯片中的 NAND 型闪存控制器模块, 其功能是完成对 NandFlash 的基本读写操作; (3) write_controller, 来自工业界工程设计中的一个写控制模块, 其主要功能是按照串行高级技术附件协议 (Serial Advanced Technology Attachment, SATA) 控制设备进行写入, 在 SATA 协议中应用层的基础上实现数据存储和地址记录的功能, 并支持编程输入输出 (Programming Input/Output, PIO) 与直接内存访问 (Direct Memory Access, DMA) 两种工作模式, 按照数据流量的大小自动选择不同工作模式. 所有实验是远程登录计算机服务器完成的, 服务器的硬件配置是 CPU 为 4 核 Intel Xeon、主频为 3.00 GHz, 内存为 6 GB, 操作系统为 Redhat Enterprise 4.

实验电路的特征描述如表 2 所示, 包括电路名称、输入数目、输出数目、原始设计中的寄存器位数以及抽象模型中的寄存器位数. 其中 b12word 是将 ITC99 中的 b12 设计的一个原始输入由 4 位扩展为 12 位, 增加了猜数字游戏的难度, 即增大了难达状态的难达程度, 同理 b13word 是将 b13 的一个原始

输入由一位扩展为 16 位, 增大了难达状态的时序深度. NandFlash 和 write_controller 中都有位宽为 32 位的输入变量位于控制条件中.

实验中, 我们从设计中选择时序深度较深的难达分支对应的状态作为难达目标状态. 在本文中, 将随机模拟在 500 万拍之内不能覆盖到的分支识别为难达分支.

4.1 与其他方法的性能比较

为了与其他方法进行比较, 此小节中的实验场景设定为单个目标覆盖, 即每一次模拟中只有一个目标状态. 实验中将本文方法与基于随机的抽象引导半形式化方法 (“随机 AGS”)、约束随机模拟方法以及形式化方法——限界模型检测 (Bounded Model Checking, BMC) 进行对比, 对比项包括最终搜索到的从初始状态到达目标状态的激励序列长度以及所用的时间. 实验结果如表 3 所示, 每一行的模拟结果取 10 次的平均结果, 每次使用不同的随机种子.

随机 AGS 是我们对参考文献[1]的基本思想框架的实现. 文献[1]中使用了一种较粗粒度的模块级的抽象方法, 该方法将与目标状态寄存器紧密相关的模块提取出来构成抽象模型, 并在此模型上应用形式化的前像计算获得抽象距离信息. 在模拟过程中, 从当前状态开始, 随机产生一组输入向量进行模拟, 然后利用抽象距离信息对模拟过程中访问到的状态进行评估, 选择其中抽象距离最小的状态作为下一步模拟的当前状态, 循环这个过程直到覆盖到目标. 为了公平起见, 本文对比实验中实现的随机 AGS 方法采用本文中使用的寄存器级的抽象方法, 相比于文献[1]中模块级的抽象方法, 寄存器级的抽象粒度更细, 能够获得更精确的抽象模型从而为模拟过程提供更精确的引导.

对比项中的约束随机模拟方法是工业界通用的模拟验证方法, BMC 的结果是学术界的开源形式化验证工具 VIS 的实验结果.

表 3 中第 1 列表示目标设计及其中的目标状态编号 (在括号内显示), 第 2 列和第 3 列表示本文方法的实验结果, 第 4 列和第 5 列表示随机 AGS 方法的实验结果, 第 6 列和第 7 列是约束随机模拟方法的结果, 最后两列是 BMC 的实验结果. 左边 (“序列长度”) 是最终找到的激励序列的长度, 右边 (“时

表 2 实验电路的特征描述

电路名称	输入数	输出数	原始设计 寄存器位数	抽象模型 寄存器位数
b01	2	2	5	3
b02	1	1	4	3
b03	4	4	30	22
b06	2	6	9	3
b08	9	4	21	5
b10	11	6	17	8
b11	7	6	31	4
b12	5	6	121	27
b12word	17	6	121	27
b13	10	10	53	28
b13word	25	10	53	28
b15	36	70	447	8
b17	37	97	1415	18
NandFlash	72	40	94	20
write_controller	153	225	383	127

① CVC3 Homepage. <http://www.cs.nyu.edu/acsys/cvc3/doc/index.html>
② <http://www.cerc.utexas.edu/itc99-benchmarks/bench.html>
③ <http://bbs.eetop.cn/thread-329826-1-1.html>

表 3 与其他方法的结果比较

目标	本文方法		随机 AGS ^[1]		约束随机		BMC	
	序列长度	时间/s	序列长度	时间/s	序列长度	时间/s	序列长度	时间/s
b01(p1)	5	0.02	5	<0.01	5	<0.01	5	<0.01
b02(p1)	5	0.01	5	<0.01	21	<0.01	5	<0.01
b03(p1)	31	0.02	33	0.01	148	0.01	5	<0.01
b06(p1)	5	0.02	5	<0.01	17	<0.01	5	<0.01
b08(p1)	19	0.04	19	0.01	23	0.01	19	0.1
b10(p1)	21	0.05	34	0.06	45	0.01	8	0.1
b11(p1)	10	0.02	14	0.01	22	<0.01	4	<0.01
b11(p2)	24	0.02	31	0.01	61	0.01	6	<0.01
b12(p1)	109	1.87	TO	TO	TO	TO	—	—
b12(p2)	33 148	54.22	TO	TO	TO	TO	—	—
b12word(p1)	109	1.92	TO	TO	TO	TO	—	—
b13(p1)	1073	2.28	1073	1.16	TO	TO	—	—
b13(p2)	2139	5.15	2139	2.68	TO	TO	—	—
b13word(p1)	1081	3.04	21 202	3.53	TO	TO	—	—
b13word(p2)	2147	7.34	44 587	9.64	TO	TO	—	—
b15(p1)	25	0.11	37	0.02	95	<0.01	6	2.4
b15(p2)	21	0.06	65	0.04	TO	TO	6	1.6
b17(p1)	67	2.05	93	0.12	283	0.02	8	8.5
b17(p2)	17	0.36	TO	TO	TO	TO	11	10.3
NandFlash(p1)	107	0.58	TO	TO	TO	TO	38	0.2
NandFlash(p2)	164	1.17	TO	TO	TO	TO	72	0.2
write_controller(p1)	1114	1.16	TO	TO	TO	TO	—	—
write_controller(p2)	1217	1.25	TO	TO	TO	TO	—	—

注：Time-out (TO)：表示五百万拍之内无法产生出结果；“—”：表示形式化工具 VIS 的 BMC 无法处理。

间”)是总模拟时间,TO 表示在 500 万拍(约 1000 s)之内无法产生出结果(目前国际上发表的抽象引导的难达状态激励生成方法大多采用了这个数值作为 TO 标准). 难达状态激励生成的最终目标是产生出一条从初始状态到目标状态的激励序列,设计中存在时序深度上万拍的状态(如 b12 的目标 p2)已经算是比较深的时序深度了. 如果在 500 万拍之内不能到达目标状态,则可以认为很难找到一条到达目标的路径了. 表 3 中本文方法及随机 AGS 方法的结果都是采用回退策略 1 的实验结果.

实验结果表明,本文方法可以覆盖到所有被选的目标状态,而其他方法都有一些目标不能覆盖到. 通过对 b12、b12word、b13word 以及 NandFlash 和 write_controller 的实验结果进行分析,我们发现,当控制条件较复杂或是字级时,采用约束求解产生候选状态的方法效果明显优于随机产生候选状态的方法,原因是当控制条件较复杂或是字级时,某些分支采用随机的方法很难覆盖到,而这些分支对应的很可能是离目标状态较近的状态. 随着这些分支变量位宽的增长,随机方法覆盖这些分支的概率呈指数降低. 而本文方法可以通过约束改变和求解的方式很容易覆盖到这些分支.

从 BMC 的实验结果中可以看出,对于规模较小的设计以及时序深度不深的目标属性,BMC 方法

能够快速到达目标状态并产生较短的路径序列. 然而对于时序深度较深的设计,BMC 则由于形式化方法面临的状态空间爆炸的问题而无法处理(如 b12、b13 以及 write_controller 的目标属性),而本文方法可以快速验证这些目标状态.

本文方法在 b01、b02、b06 和 b08 上的实验效果不明显,因为这些设计规模较小,控制条件很简单. 由于约束求解器的调用,本文方法的运行时间长于随机 AGS,但能产生较短的激励序列. b13 的原始设计虽然比较复杂,但本文方法产生的序列长度和随机 AGS 一样,原因是控制条件中约束是简单的单个 bit 位,随机次数较多时也能较快地覆盖到不同分支.

4.2 回退策略之间的比较

图 8 和图 9 记录了本文方法和随机 AGS 采用不同回退策略的实验结果,图中回 1 和回 2 分别表示回退策略 1 和回退策略 2,TO 与表 3 中的意义一致,表示在五百万拍之内无法产生出结果. 回退策略 1 不对比当前状态与候选状态的 merge_dis 值而回退策略 2 则比较. 实验结果表明,采用回退策略 2 可以产生更短的路径序列,但是花费的时间可能较长,原因是当搜索过程没有前进时,回退策略 2 的方式会产生更多的候选状态以供选择,这样可以增大更靠近目标状态的概率. 实验结果还表明,不管是采用回

退策略 1 还是回退策略 2,本文方法都可以覆盖所有被选的目标.而随机 AGS 方法采用回退策略 1 时只能覆盖到部分目标状态,原因是某些目标状态必经的分支采用随机的方式很难覆盖到;即使采用回退策略 2,随机 AGS 仍然无法覆盖到 b12word 中的目标,原因是当控制条件较复杂或是字级时,采用随机的方式,即便每一步中产生多个候选状态也依旧难以满足覆盖目标所需的条件.

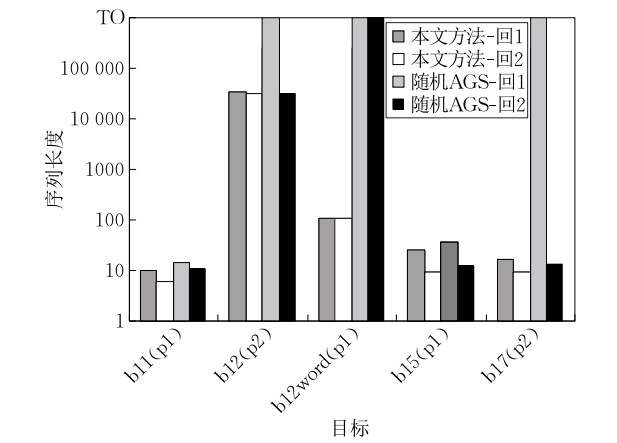


图 8 回退策略之间的比较:序列长度

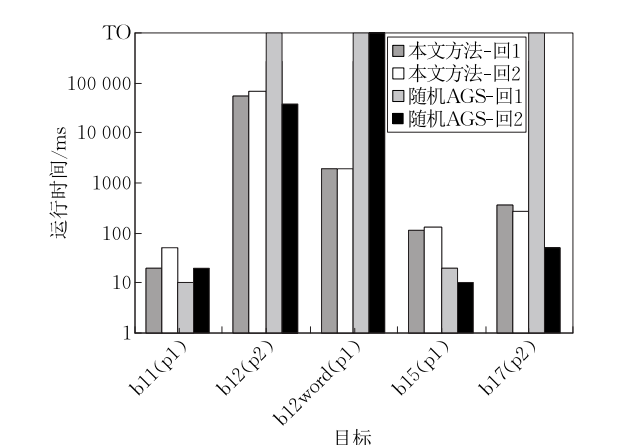


图 9 回退策略之间的比较:运行时间

4.3 多目标覆盖与单目标覆盖的效果对比

为了验证本文多目标同时覆盖的效果,我们从每个待验证设计中选择多个目标状态(两至三个)同时模拟.表 4 中记录了 PACOST 单目标覆盖和本文多目标覆盖的对比实验结果,对比是在采用回退策略 2 的情况下.单目标覆盖是采用 PACOST 方法,每次只针对一个目标状态做激励生成,而多目标覆盖是采用本文方法,每次同时针对多个目标状态来产生激励序列.实验结果表明,在大多数情况下,多目标覆盖产生的激励序列长度比各个目标单独覆盖所产生的序列长度之和要小,所用时间也较短. b11 中多目标覆盖所产生的序列长度之所以大于单目标

覆盖所产生的序列长度之和,是因为 b11 中的两个目标状态不在同一条路径上,当其中一个目标状态到达之后需要回到初始状态重新模拟一条新的路径来覆盖另一个目标状态.同理, NandFlash 中多目标同时覆盖产生的激励序列长度基本上是各个目标单独覆盖所产生的序列长度之和,也是因为所选的两个目标状态分别位于从初始状态开始的两条不同路径中.而通常在设计中,难达状态是很可能相互靠近或处于同一条路径上,采用多目标覆盖的方法能够较快地产生一条较短的路径同时覆盖到多个目标状态.比如, write_controller 的两个目标状态位于同路径中,因此多目标同时覆盖的效果要明显优于单个目标覆盖.

表 4 多目标覆盖与单目标覆盖的结果比较

目 标	PACOST 单目标覆盖		本文多目标覆盖	
	序列长度	时间/s	序列长度	时间/s
b06(t1)	4	0.01		
b06(t2)	4	0.02	8	0.04
b06(t3)	5	0.02		
b08(t1)	19	0.04		
b08(t2)	17	0.03	23	0.05
b08(t3)	21	0.04		
b11(t1)	6	0.05		
b11(t2)	7	0.03	14	0.06
b12(t1)	74	0.38		
b12(t2)	109	1.88	109	2.13
b12(t3)	76	0.35		
b13(t1)	2139	6.58		
b13(t2)	1073	2.41	2149	7.31
b13(t3)	1074	2.54		
b15(t1)	9	0.13		
b15(t2)	7	0.07	16	0.18
b15(t3)	8	0.09		
NandFlash(p1)	56	1.32	143	3.85
NandFlash(p2)	92	2.76		
write_controller(p1)	1114	3.53	1215	5.34
write_controller(p2)	1215	4.27		

5 局限性说明

本文方法的局限性在于抽象的过程需要一定的人工干预,并且能处理的抽象模型的规模依赖于现有形式化工具的处理能力.目前形式化工具能够处理的抽象模型规模有限,如果原始设计的规模与抽象模型的规模之间差异较大,会由于抽象误差导致死胡同问题,从而找不到具体的路径覆盖目标状态,这也是所有抽象距离引导的半形式化方法共同面临的问题.尽管如此,本文提出的回退策略能够一定程度上缓解死胡同问题.

此外,本文方法在每个模拟步骤中只展开一拍,

即只提取一拍的路径约束,因此对于那些位于控制条件中的中间寄存器的值无法控制,实验效果有限.在后续工作中,将考虑在控制流图的基础上构建数据流图,通过时序展开的方式,在数据流图中提取变量之间的依赖关系来控制中间寄存器的值.

6 总 结

本文提出了一种抽象引导的半形式化激励生成方法,目标是同时覆盖设计中多个难达状态.本文采用了基于路径约束求解的激励生成方法,这种方法将动态模拟和对 RTL 代码的静态分析相结合,能够有效地产生输入向量,遍历设计的不同分支,对应产生不同的候选状态.在每一步模拟中,综合考虑从各个候选状态到不同目标状态的全局抽象距离值,从中选择一个最优的状态继续模拟,最终搜索到一条能同时覆盖多个目标状态的状态序列.此外,本文使用了两种回退策略,通过尽量尝试不同的新状态来缓解死胡同问题.实验结果表明,与一种已有的抽象引导半形式化方法以及形式化的 BMC 方法比较,本文方法可以覆盖到更多的目标状态.当有多个目标位于同路径中时,相比于单目标覆盖,本文多目标覆盖方法能够在更短的时间内产生较短的激励序列同时覆盖到多个目标状态.

参 考 文 献

[1] Shyam S, Bertacco V. Distance-guided hybrid verification with GUIDO//Proceedings of the International Conference on Design, Automation and Test in Europe. Munich, Germany, 2006: 1211-1216

[2] Nanshi K, Somenzi F. Guiding simulation with increasingly refined abstract traces//Proceedings of the ACM/IEEE Design Automation Conference. San Francisco, USA, 2006: 737-742

[3] Paula F M D, Hu A J. An effective guidance strategy for abstraction-guided simulation//Proceedings of the ACM/

IEEE Design Automation Conference. San Diego, USA, 2007: 63-68

[4] Wu W X, Hsiao M S. Efficient design validation based on cultural algorithms//Proceedings of the International Conference on Design, Automation and Test in Europe. Munich, Germany, 2008: 402-407

[5] Zhang T, Lv T, Li X W. An abstraction-guided simulation approach using Markov models for microprocessor verification//Proceedings of the International Conference on Design, Automation and Test in Europe. Dresden, Germany, 2010: 484-489

[6] Wang J, Li H W, Lv T, et al. Functional test generation guided by steady-state probabilities of abstract design//Proceedings of the International Conference on Design, Automation and Test in Europe. Dresden, Germany, 2014: 1-4

[7] Zhou Y H, Wang T C, Lv T, et al. Path constraint solving based test generation for hard-to-reach states//Proceedings of the Asian Test Symposium. Yilan, China, 2013: 239-244

[8] Chatterjee D, Bertacco V. Activity-based refinement for abstraction-guided simulation//Proceedings of the IEEE International High Level Design and Test Workshop. San Francisco, USA, 2009: 146-153

[9] Wang J, Li H W, Li X W. A novel abstraction-guided simulation approach using posterior probabilities for verification//Proceedings of the International Symposium on VLSI Design, Automation and Test. Hsinchu, China, 2014: 1-4

[10] Liu L Y, Vasudevan S. STAR: Generating input vectors for design validation by static analysis of RTL//Proceedings of the IEEE International High Level Design and Test Workshop. San Francisco, USA, 2009: 32-37

[11] Liu L Y, Vasudevan S. Efficient validation input generation in RTL by hybridized source code analysis//Proceedings of the International Conference on Design, Automation and Test in Europe. Grenoble, France, 2011: 1-6

[12] Liu Jie, Xu Wei-Jun, Xia Yu-Wen, Qin Ji-Long. Constrained random in design verification. China Integrated Circuit, 2006, 15(11): 28-32(in Chinese)

(刘杰,徐伟俊,夏宇闻,秦冀龙.设计验证中的随机约束.中国集成电路,2006,15(11):28-32)



ZHOU Yan-Hong, born in 1987, Ph. D. candidate. Her current research focuses on semi-formal verification of VLSI.

WANG Tian-Cheng, born in 1983, Ph. D. candidate. His main research interests include design verification of VLSI.

LI Hua-Wei, born in 1974, Ph. D., professor, Ph. D. supervisor. Her main research interests include VLSI test, design verification, design for reliability, and fault tolerant computing.

LV Tao, born in 1978, Ph.D. Her main research interests include VLSI design, test and verification.

LI Xiao-Wei, born in 1964, Ph. D. , professor, Ph. D.

supervisor. His main research interests include VLSI test, design for reliability, design verification, dependable computing and wireless sensor networks.

Background

Test generation for hard-to-reach states is one of the hardest tasks in functional verification and it is difficult to generate effective test cases for these states. Semi-formal methods, which combine the scalability of simulation and the completeness of formal methods while mitigate their weaknesses, have been proposed in recent years. Among them, the abstraction-guided simulation (AGS) is quite promising, in which abstract distances are acquired from an abstract model by formal analysis to direct the simulation towards a target state.

The work proposed by Shyam S in 2006 is the pioneering work of AGS. Many other researchers have made a great number of works to improve the efficiency of AGS. Nanshi K utilizes abstraction refinement, while Paula F M D proposes a more advanced guidance strategy. Wu Weixin and Zhang Tao introduce cultural algorithm and Markov model to improve

the efficiency of test generation respectively. The former works generally focus on one target state. This paper proposes an AGS approach based on path constraint solving to cover multiple target states simultaneously. We use an evaluation function, which considers the abstract distances for one state to different target states, to evaluate the candidate next states and guide the simulation process to cover multiple target states.

This work is supported in part by the National Natural Science Foundation of China under Grant Nos.61432017, 61176040, and in part by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302501. In the past years, our research group has done a lot of work including the abstraction-guided simulation methods based on Markov model, quantitative evaluation methods in design verification and so on.