

引用格式: 杨凯峰, 张雪倩, 冯娇, 等. 基于遗传进化策略的处理器验证方案[J]. 微电子学与计算机, 2021, 38(5): 25-29. [YANG K F, ZHANG X Q, FENG J, et al. Research on verification method of processor based on genetic evolution strategy[J]. Microelectronics & Computer, 2021, 38(5): 25-29.]

基于遗传进化策略的处理器验证方案

杨凯峰, 张雪倩, 冯 娇, 梁 峰
(西安交通大学 微电子学院, 陕西 西安, 710049)

摘 要: RISC-V 作为近年来发展迅速的开源处理器架构, 对其验证方法的研究具有很高的工业应用价值. 提出了基于遗传进化策略的处理器验证优化方案, 利用遗传算法产生指令的权重系数, 为测试用例的生成确定方向, 并对 RISC-V 处理器 RV32I 部分指令完成了模拟验证. 与基于伪随机策略生成测试用例的方案相比, 应用遗传进化策略生成测试用例时, 能够维持对功能覆盖率贡献更高的方向, 在验证末期即将陷入局部最优解时, 仍然能够突破当前状态, 使整体功能覆盖率进一步提高. 同时, 该验证优化方案对验证环境的调整实现了完全自动化, 具有较高的可扩展性和可维护性.

关键词: 遗传进化算法; 功能覆盖率; 通用验证方法学; RISC-V

中图分类号: TN407 **文献标识码:** A **文章编号:** 1000—7180(2021)05—0025—05

Research on verification method of processor based on genetic evolution strategy

YANG Kai-feng, ZHANG Xue-qian, FENG Jiao, LIANG Feng

(School of Micro electronics, Xi'an Jiaotong University, Xi'an 710049, Chian)

Abstract: As an open source processor architecture with rapid development in recent years, RISC-V has high industrial application value in the research of its verification method. In this paper, an optimization scheme of processor verification based on the genetic evolution strategy is proposed, which uses a genetic algorithm to generate instruction weight coefficients to determine the direction for the generation of test cases, and some instructions of RV32I of the RISC-V processor are simulated and verified. Compared with the scheme based on the pseudo-random strategy to generate test cases, the application of genetic evolution strategy to generate test cases can maintain a higher contribution to the function coverage. When it is about to fall into a locally optimal solution at the end of verification, it is still able to break out of the current state, allowing a further improvement of the overall functional coverage. At the same time, the verification optimization scheme realizes the automatic adjustment of the verification environment and has high scalability and maintainability.

Key words: genetic evolution strategy; function coverage; universal verification methodology; RISC-V

1 引言

随着人工智能、云计算等新兴领域在全球掀起

热潮, 集成电路产业作为信息技术产业的核心, 其市场规模逐年扩大, 发展机遇与挑战并存. 近年来, 我国集成电路产业快速发展, 设计制造能力有了显著

提高,但与发达国家相比仍存在明显差距. 验证作为保证芯片设计质量的重要环节,其技术也在不断革新. 目前,验证的主流趋势是采用功能覆盖率驱动的方法,使用 SystemVerilog 语言基于 UVM(Universal Verification Methodology)搭建验证平台^[1]. 目前对验证优化的方案主要集中在测试用例生成环节,例如使用递归神经网络优化随机测试向量生成器^[2];建立代码覆盖率与功能覆盖率之间的概率模型以指导测试向量自动生成^[3];根据硬件设计建立参数数据库,通过静态分析的方法生成新的测试用例^[4].

结合 UVM 验证平台和遗传进化算法的优势,本文提出了基于遗传进化策略的验证优化方案,利用遗传进化算法为随机指令的生成确定方向,将遗传进化算法与基于随机指令生成器的验证环境相结合,实现验证的自动优化,在一定程度上提高了验证效率.

2 遗传进化策略

2.1 进化算法

进化算法是一种基于现实生物学理论的算法,以生物学进化论的理论为基础,通过繁殖、变异、重组和选择等机制进行不断迭代,直到产生符合条件的个体^[5]. 这种算法具有较高的鲁棒性以及自适应,因此被广泛应用于各种研究场景.

在进化算法中,将单个候选解称为个体,将同一时间内存在的一组个体集合称为种群或人口. 当给定一个种群,同时给定至少一个适应度的计算公式时,通过计算适应度就可以将该种群中的个体分出优劣,在生成下一代种群时触发自然选择机制,保留优势个体,剔除劣势个体. 随着进化的进行,种群的适应度值将会逐渐上升,种群中将会存有更多的优势个体和更少的劣势个体. 这个过程是不断迭代的,类比于种群一代一代地进化,直到达到某个设定的停止条件. 因此,进化过程是离散的步骤,优化的过程是在每一代种群生成之后,需要分别计算每个个体的适应度,最终留下的种群将由适应度判断最优的个体组成,与初始种群没有直接的联系.

进化算法的基本流程为:首先以初始条件生成初代种群,然后选择父代,应用重组或突变等遗传算子生成对应的子代,根据适应度选择出最优的个体组成当前种群,继续进行父代生成子代的过程,当某一代种群达到设置的终止条件时,进化过程结束.

2.2 MicroGP

MicroGP 作为一种开源优化求解工具,它最初版本的目的是应用于不同处理器的汇编程序生成,在经过几个版本的迭代后,它可以被应用于实参数优化等多个领域.

它的设计思想是,对于一组个体,评估其适应度值,将适应度作为反馈的参数,指导个体遗传进化的方向,以产生最优解. 整体结构是由遗传进化核心、约束和外部评估器三部分组成^[6],如图 1 所示.

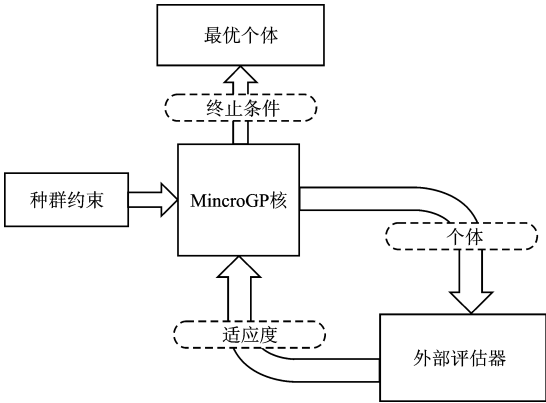


图 1 MicroGP 结构图

MicroGP 的遗传进化机制具有自适应性,算子的激活概率和突变强度都是进化核心的内生参数,在运行过程中由进化核心根据当前进化状态进行调整. 其中的遗传算子决定了个体在进化过程中可选的方向,即内部节点的变化,是算法执行的基本操作,主要包括重组、搜索和突变三种:

- (1)重组算子指的是将两个个体进行合并,重组产生一个子代个体.
- (2)搜索算子主要应用于约束文件中宏的参数,它可以在预设范围内对参数值进行探索,改变具体节点的属性,尝试得到最优的表现型.
- (3)突变算子又分为添加(Add)、删除(Remove)和修改(Modify)三种.

MicroGP 具体的实现过程为:

- (1)解析约束文件,生成初代种群;
- (2)由外部评估器对每个个体计算适应度值;
- (3)判断该种群是否满足进化终止的条件,如果不符合,则参考适应度选择合适的遗传算子,再选出多个优势个体作为父代,进化生成多个子代,组成新的种群,重复之前的流程,直到满足进化终止的条件. 实现流程图如图 2 所示.

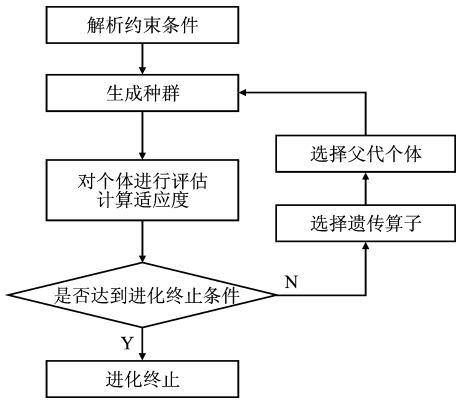


图 2 MicroGP 遗传进化实现过程

3 基于遗传进化策略的验证优化方案

3.1 验证方案

在使用随机指令生成器进行验证时发现,由于每次测试用例的产生都是完全随机的,每次生成的测试用例对功能覆盖率的贡献会逐渐降低,并且在验证后期存在个别功能点始终覆盖不到的情况.对此,本文将遗传进化核心与基于 UVM 的随机指令生成器结合,旨在使用遗传进化策略对测试用例的生成进行优化,提高单个测试用例对功能覆盖率的贡献.

对于随机指令生成器的优化,本文通过对各指令自动分配权值来实现.因为基于纯随机的验证方案生成的测试用例是没有方向性的,尤其在验证末期,一些已达到 100%覆盖率的指令仍会重复出现在新的测试用例中,占用了其他未覆盖完全的指令的空间.经过增加权值的自动调整,当 MicroGP 发现在遗传进化方向中继续增加某些指令的权值不会对功能覆盖率再有新贡献时,则会在之后降低相应指令的权值;反之,当增加某些指令的权值,发现对功能覆盖率有新贡献,则会在之后调高相应的指令权值.这样就可以保持测试用例的生成总是朝着使功能覆盖率增加的方向.

为了直观对比优化方案的效果,本文将评判标准设为:当达到同一功能覆盖率水平时,方案所需的时间和测试用例数目;在验证后期,当测试用例数目相同时,方案能够达到的功能覆盖率水平.

本文提出的基于遗传进化策略的验证方案如图 3 所示.这是基于 MicroGP 遗传进化核心,UVM 随机指令生成器和指令集模拟器 Spike 联合仿真的验证方案.该方案的重点在于利用 MicroGP 对测试用例的生成进行有目标地优化^[7].利用 MicroGP 对产

生的测试用例进行评估,产生下一代测试指令的权重系数.首先,由遗传进化核心根据输入的约束文件生成一组权值文件;然后,该文件将被评估器读入,用于计算适应度值;每次计算适应度之后,对应功能覆盖的最高值作为本次遗传进化的结果;本次和上次适应度的变化将作为遗传进化核心的参考,指导遗传算子的应用,生成下一组权值文件.

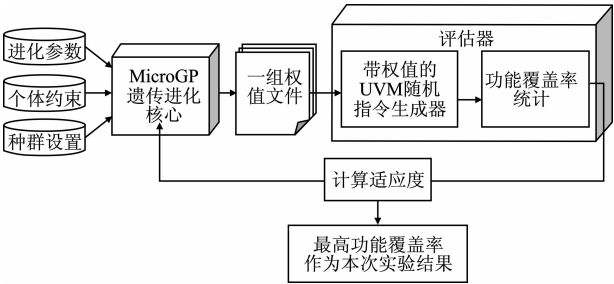


图 3 基于遗传进化策略的验证优化方案

下面对该方案中的各个模块进行简要说明:

(1)约束文件:包括进化参数文件、种群设置文件和个体约束文件,均为 xml 格式.进化参数和种群设置是对 MicroGP 进行参数配置的格式化文件.个体约束文件是对遗传进化核心生成文件的具体描述,由用户自定义编写.该方案中选择指令权值作为随机指令生成器的参数,即需要生成的是包含各指令权值信息的文件.

(2)遗传进化核心:执行遗传进化算法的核心程序.

(3)个体:遗传进化核心生成的文件,同时也是评估器的输入文件.该方案中,生成的个体即为各指令权值描述文件.

(4)适应度:用于对个体优劣程度进行评判的参数,其值越高,对应个体越优秀.本方案选取了整体功能覆盖率和已覆盖的指令比例作为二维适应度.

(4)外部评估器:计算适应度的关键模块.使用基于随机指令生成器和指令集模拟器的验证平台进行功能覆盖率的收集和分析.同时,为了接收指令权值文件,需要在验证环境中加入解析并应用权值文件的功能.

(5)执行脚本:将以上各个模块连接起来,形成反馈的闭环.需要在该脚本中完成各个模块的调用和连接,并将生成的结果以特定格式写入 MicroGP 系统文件中.

遗传进化核心和外部评估器构成了一个闭合的进化循环,一旦开始进化,将不需要人工干预,直到 MicroGP 判断达到终止条件.

3.2 实验结果与分析

实验在 Linux 环境下进行,仿真器使用 Synopsys 公司的 VCS 软件,实验对象为 RV32I 中的运算、逻辑和位移指令。

为了进行有效的对比,本文采用两种验证方案对 RV32I 的运算、逻辑和位移指令进行验证:基于随机指令生成器的验证方案和基于遗传进化策略的验证方案。

将基于遗传进化策略的验证优化方案与基于随机指令生成器的验证方案进行对比,得到整体功能覆盖率随测试用例增加的趋势对比,如图 4 所示。从图中可以看到,在验证初期,功能覆盖率低于 60.00% 时,由于每个测试用例对覆盖率的贡献都很高,两个方案的差别并不明显。随着实验进行,在功能覆盖率为 60.00%~94.00% 区间遗传进化策略的优势开始显现,表现为新增测试用例对覆盖率的贡献高于随机策略。在实验中后期,功能覆盖率为 94.00%~96.00% 区间,随机策略较优于遗传进化策略,分析在此阶段随机策略的覆盖点增长较为稳定,而遗传进化策略还在探索阶段,没有及时收敛。在实验末期,即覆盖率超过 96.00% 之后,进入对边界等极限情况的探索时,遗传覆盖率表现更好。

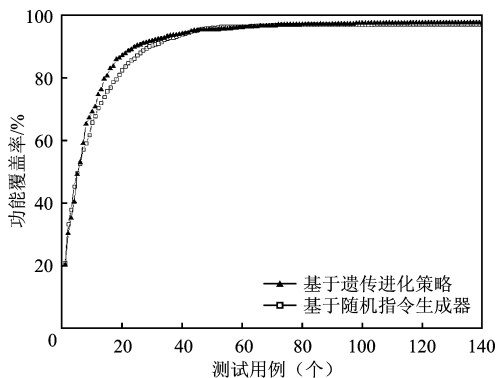


图 4 功能覆盖率增长曲线对比

达到 97.00% 的功能覆盖率时,基于随机指令生成器的方案所需测试用例数目为 92 个,耗时 101 分钟。基于遗传进化策略的验证优化方案需要生成 72 个测试用例,耗时 74 分钟,测试用例数目减少 21.73%,仿真耗时减少 26.73%。此时, SUB、SLLI、SLTU、ADD、SRL、SRA、SLT、SRAI、SLL 指令均已达到完全覆盖,其余单条大部分的功能覆盖率均超过 91.00%。未完全覆盖的指令有 ADDI、ANDI、XORI、ORI、XOR、OR、AND、SRLI、SLTIU、SLTI,其中有 1 条算数类指令,6 条逻辑类指令,3 条位移类指令。

ADDI 指令的单条覆盖率最低,仅为 83.12%,尚未覆盖到的主要是交叉功能点。因为该指令中包含两个寄存器变量和一个立即数变量,定义的交叉覆盖率要求两个寄存器均要覆盖到 32 个 RISC-V 寄存器,立即数要求覆盖到正数和负数,即共有 2 048 种组合情况,要达到基于随机的完全覆盖需要花费较长时间。

当增加测试用例数目至 180 个后,整体功能覆盖率达到 97.85%,遗传进化过程因为无法进一步优化而终止。至此,除 ADDI 外,所有指令的覆盖率均超过 95.00%。

进一步分析,在覆盖率达到 96.00% 之后,继续生成测试用例,此时进入覆盖率收敛阶段,将存在一些非常难覆盖到的边界功能点,得到功能覆盖率的变化曲线对比如图 5 所示。从图中可以明显看到,基于随机策略的方案在验证末期仍然保持几乎连续的功能覆盖率增长趋势,但增长幅度非常有限,经常出现了为了增加 0.01% 的覆盖率需要生成 10 个左右测试用例的情况。这主要是因为随机策略的无方向性,越到验证后期,其劣势越发明显,每个测试用例之间的对功能覆盖率的贡献差异都很小。而基于遗传进化策略的方案在该阶段表现出阶梯型的增长趋势。由于遗传计划策略的离散性,有一定概率进化出超出当前平均状态的优势个体,该个体的基因将会进入下一代个体中,为整体功能覆盖率的提高做出贡献。这种策略在后期即将陷入局部最优解的时候,仍有概率突破当前状态,搜寻到更加优势的方向。

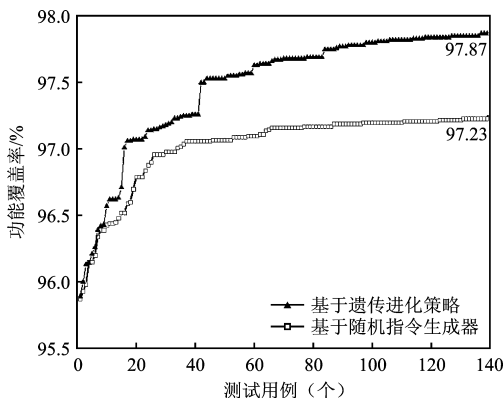


图 5 功能覆盖率收敛阶段的增长曲线对比

进入验证收敛阶段后,功能覆盖率的提高变得更加困难,此时对测试用例的生成和优化是非常具有挑战性的。当整体功能覆盖率超过 96.00% 后,生成相等数量的测试用例,均为 140 个,采用随机策略生成的测试用例可以使功能覆盖率继续提高

1.23%;应用遗传进化策略生成的测试用例可以使功能覆盖率继续提高 1.87%,对该阶段而言优化明显。

4 结束语

针对目前流行的 RISC-V 开源处理器架构,本文提出了一种基于遗传进化策略的验证方案,利用 MicroGP 对产生的测试用例进行评估,产生下一代测试指令的权重系数,作为随机指令生成器的参数来给指令的生成确定方向。与基于随机策略的验证方案相比,该方案可以根据测试用例对整体功能覆盖率的贡献值自动调整指令的权值,使功能覆盖率能够更快速地收敛。经实验对比,基于遗传进化策略的验证方案在达到 97.00% 的整体功能覆盖率时,所需测试用例数目与基于随机策略的验证方案相比减少 21.73%,耗时减少 26.73%,有较好的优化。并且在验证末期仍然能够探索出对功能覆盖率有贡献的测试用例。

在后续工作中,可采用该方案对更为复杂的分支、跳转等指令进行验证,衡量遗传进化策略对完整验证工作的促进作用。另外,在 MicroGP 的个体约束文件中,可以实现复杂的指令结构定义,可尝试使用 MicroGP 作为测试用例的生成模块,进一步发挥遗传进化算法在验证工作中的优势。

参考文献:

- [1] 刘斌. 芯片验证漫游指南[M]. 北京: 电子工业出版社, 2018.
LIU B. Walking guide to SOC verification[M]. Beijing: Publishing House of Electronics Industry, 2018.
- [2] FAJCIK M, SMRZ P, ZACHARIASOVA M. Automation of processor verification using recurrent neural networks[C]//2017 18th International Workshop on Microprocessor and SOC Test and Verification (MTV). Austin: IEEE, 2017: 15-20. DOI: 10.1109/

MTV. 2017. 15.

- [3] 傅亮, 卢鼎, 张志敏, 等. 通过分析代码覆盖提高功能覆盖率的验证输入自动生成方法[J]. 计算机辅助设计与图形学学报, 2009, 21(4): 454-460.
FU L, LU D, ZHANG Z M, et al. An automatic test generation method for functional coverage improvement by code coverage analysis[J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(4): 454-460.
- [4] GHOSH P. Analysis and verification of SoC design RTL parameters[C]//2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES). Delhi: IEEE, 2016: 1-6. DOI: 10.1109/ICPEICES.2016.7853486.
- [5] EIBEN A E, SMITH J E. Introduction to evolutionary computing[M]. Heidelberg: Springer-Verlag Berlin Heidelberg, 2003. DOI: 10.1007/978-3-662-05094-1.
- [6] SANCHEZ E, SCHILLACIM, SQUILLERO G. Evolutionary optimization: the Micro GP toolkit[M]. Boston: Springer Science and Business Media, 2011. DOI: 10.1007/978-0-387-09426-7.
- [7] 王璇. 基于 UVM 的 FPGA 中可编程逻辑块的验证[D]. 西安: 西安电子科技大学, 2019.
WANG X. The UVM-based verification of the CLB in FPGA[D]. Xi'an: Xidian University, 2019.

作者简介:

杨凯峰 男, (1994-), 硕士研究生. 研究方向为超大规模集成电路设计、超大规模集成电路测试与可测性设计。

张雪倩 女, (1994-), 硕士研究生. 研究方向为超大规模集成电路测试与可测性设计。

冯 娇 女, (1994-), 硕士研究生. 研究方向为人工智能算法的集成电路实现。

梁 峰(通讯作者) 男, (1976-), 博士, 副教授. 研究方向为片上系统(SoC)与专用集成电路技术研究、超大规模集成电路测试与可测性设计. E-mail: fengliang@xjtu.edu.cn.