

基于机器学习的功能覆盖点均衡分布算法

刘光宇^{1,2,3}, 林子明³, 倪园慧³, 李志强^{1,2†}, 梁利平⁴

(1. 中国科学院微电子研究所, 北京 100029;

2. 中国科学院大学集成电路学院, 北京 101408;

3. 北京全路通信信号研究设计院集团有限公司, 北京 100070;

4. 北京邮电大学集成电路学院, 北京 100876)

摘要:覆盖率是检验芯片验证完备性的重要指标, 尤其是功能覆盖率, 可衡量设计的功能特性是否被充分验证. 目前对于功能覆盖率的收集, 通用做法是在覆盖组中设定功能点, 将测试场景依据复杂度划分为特定数量的仓, 再运行仿真确定各个仓是否被击中. 其实现相对简单, 但由于各种因素的存在, 一个功能点中各个仓的命中情况往往分布极不平衡, 导致对一些场景的覆盖不够充分. 针对该问题, 提出了一种基于机器学习算法实现覆盖率均衡分布的验证方法, 通过对神经网络进行训练, 可对各种激励向量进行精确预测. 该方法针对较小及较大数目覆盖仓, 分别设计了反向网络预测及正向网络实时拟合的方法, 可方便实现各个仓位的均衡命中. 实验结果表明, 与随机测试覆盖点命中次数极大值与极小值差异在数个量级的情况相比, 小点数仓位可基本实现平均分布, 大点数仓位可将极值比缩小在 1.5 倍以内, 从而显著减少部分情况的验证风险.

关键词:功能覆盖率; 均衡分布; 神经网络; 拟合

中图分类号: TN431.2

文献标志码: A

Equilibrium Distribution Algorithm of Function Coverage Points Based on Machine Learning

LIU Guangyu^{1,2,3}, LIN Ziming³, NI Yuanhui³, LI Zhiqiang^{1,2†}, LIANG Liping⁴

(1. Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China;

2. School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 101408, China;

3. CRSC Research & Design Institute Group Co., Ltd., Beijing 100070, China;

4. School of Integrated Circuits, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Coverage is an important index to test the completeness of chip verification, especially functional coverage, which can measure whether the functional characteristics of the design are fully verified. At present, for the collection of function coverage, the general practice is to set function points in the coverage group, divide the test scene into a specific number of bins according to complexity, and then run the simulation to determine whether each bin is hit. Its implementation is relatively simple, but due to the existence of various factors, the hit situation of

* 收稿日期: 2024-12-14

基金项目: 中国科学院青年交叉团队项目(JCTD-2022-07), CAS Youth Interdisciplinary(JCTD-2022-07)

作者简介: 刘光宇(1986—), 男, 河北沧州人, 中国科学院大学博士研究生

† 通信联系人, E-mail: lizhiqiang@ime.ac.cn

each bin in a function point is often unbalanced, resulting in insufficient coverage of some scenes. To solve this problem, a verification method based on machine learning algorithm to achieve uniform coverage distribution is proposed. By training neural networks, various excitation vectors can be accurately predicted. In this method, the reverse network prediction method and the forward network real-time fitting method are designed respectively for small and large number of covered bins, which can realize the balanced hit of each bin easily. The experimental results show that, compared with the case where the difference between the maximum and the minimum hit times of random test coverage points is several orders of magnitude, the small point bins can basically achieve the average distribution and the large point bins can reduce the extreme value ratio to less than 1.5 times, thus significantly reducing the verification risk of some cases.

Key words: functional coverage; equilibrium distribution; neural networks; fitting

随着芯片规模的不断扩大,功能复杂性不断增强,其验证难度也在大幅攀升.目前在整个芯片开发周期中,设计与验证工作的比例已经达到了70%^[1],而单独验证所占比例已超过50%^[2],且有逐年攀升的趋势.在有限的时间及人力条件下完成芯片测试任务,成为验证人员的主要挑战.覆盖率是检验芯片测试完备性的主要指标,其主要包含代码覆盖率和功能覆盖率.代码覆盖率可以证实设计代码的执行情况,并判定是否有冗余存在,但不能确保设计功能的完备性,由此引入功能覆盖率,以检测用例是否测试充分^[3].如何保证功能覆盖的收敛,是验证首要考虑的任务^[4].

验证的关键点是在有限案例集情况下尽量找到设计中的缺陷^[5],当前芯片的验证仍以受约束随机测试^[6](constrained random test, CRT)为主,但随机测试对相当部分边界情况难以覆盖,因此需要覆盖驱动^[7](coverage directed generate, CDG)技术进行完善.文献[8]提出了一种自动约束提取工具,通过分析仿真数据,实现对多个信号的控制,减少对定向测试的依赖.文献[9]结合了程序切片、动态搜索及基于数值变化转储(value change dump, VCD)文件的分析方法,从而显著提高了覆盖效率.机器学习作为新兴学科技术,在验证工作上也表现出了强大优势.遗传算法^[10]通过其选择、交叉及变异特性,可根据覆盖结果生成优化激励以加速验证.贝叶斯网络^[11]使用有向无环图^[12](directed acyclic graph, DAG)模型表示随机变量之间的依赖关系,从而在功能测试中表现出强大性能.马尔可夫链^[13-15]的无记忆性,使得其预测实现简单,并能有效发现未知

错误.支持向量机^[16-19]、决策树^[20]、聚类^[21]等算法,因其强大的学习能力,也可有效识别验证中的冗余激励,提升验证效率.文献[22]采用了基于Q-learning及蒙特卡罗的强化学习方法,可生成最大化覆盖率的激励序列,从而显著降低达到目标覆盖的时间.深度学习算法作为机器学习的重要分支,近年在验证领域也崭露头角.文献[23]采用网络模型预测随机输入对输出的影响,实现了对输入向量的分类处理.文献[24]针对可编程视觉芯片设计了一款卷积网络编译器,优化激励生成,增强了汇编代码的可重用性,提升了算法的仿真性能.

功能覆盖率的收集,主要是在平台中添加组件,在覆盖组中设定单点或交叉功能点,将测试场景划分为多个仓位,运行仿真案例后确定各仓位的覆盖情况.以上各类算法在不同程度上加速了仓位覆盖的收敛,然而容易忽略的是,其仅仅完成了100%的覆盖率要求,各个仓位的分布却极不均衡,一个覆盖点中的仓位命中次数的差异甚至会达到几个数量级,导致尽管平台投入大量时间生成和运行测试案例,但实际上相当一部分仓位的命中率极低,其中大多数属于边界情况,从而显著增加了错误逃逸的风险.

针对上述问题,提出了一种基于机器学习算法实现覆盖率均衡分布的验证方法,该方法按仓的大小分别进行处理.对于仓数较小的情况,设计反向神经网络,将仓命中率作为输入,激励向量作为输出,从而直接选择出均匀分布情况对应的激励.对于仓数较大的情况,设计正向神经网络,通过筛选网络预测值进行拟合,并通过仿真实时更新拟合结

果,从而达到仓位的近乎均匀分布。

1 验证需求分析

本次进行验证的是一款数字滤波芯片,主要实现对输入数据进行滤波,其主体结构如图 1 所示, CPU 总线对寄存器进行配置后,滤波器开始工作,由 ADC 读取控制器将 AD 采样数据送入数据通路,随后开始进行滤波处理。其中数据通路共有 8 条,可单独实现 4 路 FIR 滤波、4 路 IIR 滤波或 FIR 滤波后再进行 IIR 滤波的组合形式。滤波结果经 CPU 总线接口读取,可实现中断读取或查询模式。其中 AD 接口和 CPU 接口均可实现串并连方式。

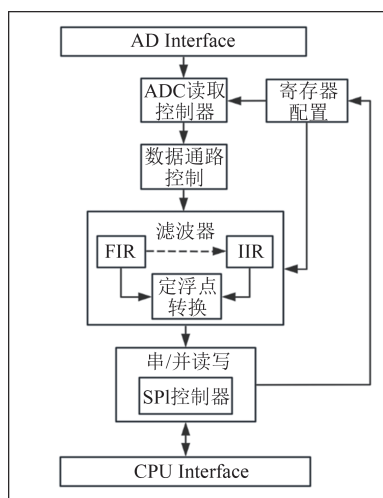


图 1 滤波芯片结构流程图

Fig.1 Filter chip structure flow diagram

配置寄存器涵盖输入输出采样率、中断模式、滤波阶数、AD 精度、AD 时序、滤波系数等多种模式,该芯片的验证核心为遍历所有配置的组合情况。覆盖率模块对其所有功能点及使用场景进行统计,功能覆盖点可分为单信号点和交叉点。对分布的影响主要体现在交叉点上,多个参数的组合,增加了其分布的未知性。理论上,可以通过调整激励参数或增加仿真事务数量来提升低命中情况的表现。然而,参数调节存在一定的不确定性,特别是在仓数较多的情况下,可能会引发连锁反应,导致效果波动;而加大仿真事务,首先增加了仿真时间,其次虽然各个仓的命中次数均有提升,但各自占比保持不变。

功能覆盖点中存在各种非线性因素,而这恰好是神经网络的处理优势。非线性条件的引入主要体现在以下方面:

1)条件互斥:涉及具体场景应用中,部分参数之间可能存在互斥情况,需采用 ignore_bins 方法将其排除掉,如组合滤波情况下内部不能超过 4 通道,AD 采样通道数与每通道采样数须有对应。

2)随机化回调:随机化中会尽量使所有变量满足约束条件,但对于过多约束叠加情况,在条件中难以完全满足,因此进行后处理,进一步调整变量约束条件。如对 AD 串行单通道采样情况,特定位宽、通道数目、时序等对应采样率可能难以满足,其约束关系复杂,在随机中难以进行约束,需在后处理模块中对参数进行调整。

3)约束与覆盖分区失配:考虑覆盖组的复杂情况,可能会简化或加大收集度,导致其与随机组件的颗粒度划分不一致。如对采样通道,随机时为简化处理,将其分组适当合并,而在覆盖收集时需细致统计。

2 网络设计及实现过程

覆盖组中会包含多个覆盖点,其中交叉点主要依据测试场景划分,而一次仿真会包含多笔测试场景事务,故而激励参数的变化,会影响各个覆盖点的命中次数。为排除其引入的影响,选定的交叉点均包含所有场景情况,即该点仓命中总和与仿真事务数目一致。激励向量中,选取其中 15 个典型配置变量,将其依权重分解为 36 个参数。对于仓位的大小情况,初步判定并经前期实验表明,创建的仓数越大,其预测分布越差,另外仓的大小会直接影响神经网络的规模及训练难度,因此实际应用中,小点数及大点数仓的测试方法有所不同。

2.1 反向预测

试验中小点数仓位选择的是基础应用场景,完成基本功能实现,其交叉覆盖包含的变量如表 1 所示。其中采样通道数目可选择 1、2、3、4、6、8 六种情况,为控制仓位总数,将该变量设置为两个仓。其中针对单 FIR 和单 IIR 模式,其总通道数不应大于 4,综上,各分区数乘积后排除非法情况,实际仓数为 32 个。

小点数仓位数据量较小,预计激励参数设置合理情况下各个仓命中率会得到相对均衡分布。为简化实现过程,采用反向网络,将命中率作为网络输入,激励参数作为网络输出以训练网络,如此即可省去网络的筛选时间,简化实现步骤。将平均命中率

表1 小点数仓变量设置

Tab.1 Variable setting of small point bins		
变量	分区数目	说明
int_mode	2	中断或查询模式
filter_mode	3	FIR、IIR、FIR+IIR
ad_para	2	AD串/并行采样选择
chan	2	采样通道数目
cpu_para	2	CPU总线串/并行模式选择

送入训练好的网络,即可得到均匀分布对应的激励参数.其总体实现流程图如图2所示.

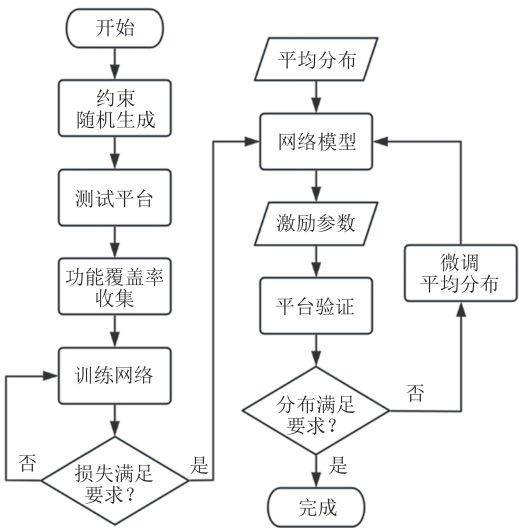


图2 小点数仓实现流程图
Fig.2 Small point bins implementation flow chart

实际操作中,平均分布对应的激励参数并非完全理想,因此加入微调步骤,首先针对单个仓的平均覆盖进行遍历调整,调整步幅设置为±0.001,其后观察反向预测情况,针对更好预测的方向进行步幅的加大,直至其产生恶化为止,最终得到比平均值预测更为良好的分布效果.

该网络输入输出数据长度均较短,为使反向预测更为精准,因此需要较高的预测精度,为此设计的神经网络结构如图3所示,由卷积网络和全连接层组合而成.

32个仓命中数据补零后形成2维6×6模式输入向量,经4次卷积、激活后展平,再经3次全连接后得到输出结果.为尽量保留边界信息并保持数据规模,第一次卷积时四周分别填充2行,第二次及第三次卷积时分别填充1行,卷积核大小均为3×3.为简化运算复杂度,激活函数均设为ReLU函数.

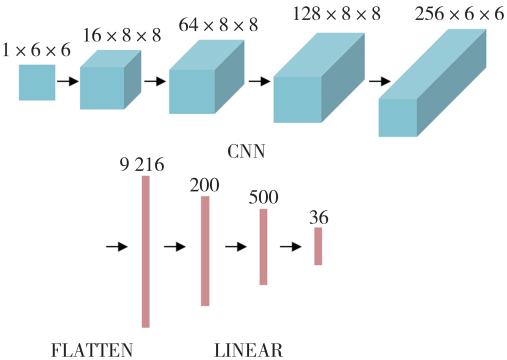


图3 小点数仓神经网络
Fig.3 Small point bins neural network

2.2 正向实时拟合

大点数仓位选择的是更完善和细致的应用场景,其交叉覆盖包含的变量设置如表2所示.其中AD位宽对应变量取值范围可选1~16,为简化操作,将该变量设置为3个仓.除针对单FIR和单IIR模式,其总通道数不应大于4的约束外,还需满足AD串行模式通路数与总通道数的约束、与滤波模式的约束、与AD串并行模式的选择等.综上,其实际仓数为408个.

表2 大点数仓变量设置

Tab.2 Variable setting of big point bins		
变量	分区数目	说明
int_mode	2	中断或查询模式
filter_mode	3	FIR、IIR、FIR+IIR
ad_para	2	AD串/并行采样选择
serial_ch	3	AD串行模式通路组数
ad_ch	6	采样通道数目
ad_width	3	AD位宽
cpu_para	2	CPU总线串/并行模式选择

大点数仓位数据规模相对较大,因此仅靠参数调节无法对各个覆盖进行均衡,且各个仓位的相关性更复杂,预期反向网络预测效果会较差,因此此处采用正向网络实现,通过训练好的网络生成各种随机向量的预测数据集,并对其进行筛选.对于任意激励,其覆盖点中各个仓的差异很大,因此无法通过单次筛选挑出合适的激励向量,此处采用多次筛选的方法,将生成数据集和训练集中筛选出的结果经多次拟合后得到相对平均的结果,其总体流程如图4所示.此处设定总仿真事务固定,因此拟合需在指定次数下完成.

网络生成的预测数据集越大,拟合的效果越好,但同时拟合的计算时间会显著加长,因此需综合考

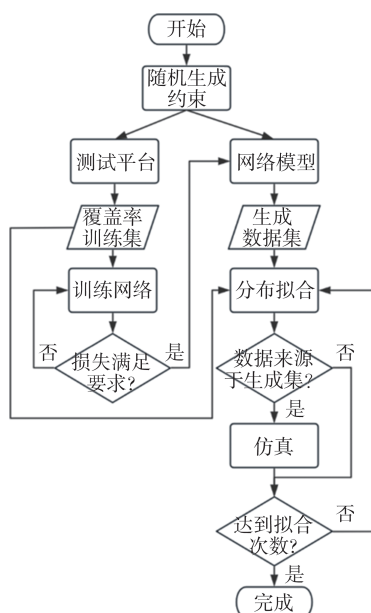


图4 大点数仓实现流程图

Fig.4 Big point bins implementation flow chart

虑. 结合实际运算时间, 设定的数据集规模在 7 000 万左右.

拟合以均方差最小的向量为起点, 并在本次拟合完成后计算下次需拟合逼近的向量. 向量来源于训练集的为真实结果, 可直接进行使用; 来源于生成集的, 由于有误差精度存在, 需进行实时仿真, 以仿真结果进行后续运算. 为加速收敛, 每次将需拟合向量按大小顺序排列, 均方差计算时将其两端分别施加权重因子, 以加速极大和极小值向中间靠拢, 其计算方法为:

$$\sum_{i=0}^{k-1} \alpha (x_i - \hat{X}_i)^2 + \sum_{i=k}^{N-k-1} (x_i - \hat{X}_i)^2 + \sum_{i=N-k}^{N-1} \beta (x_i - \hat{X}_i)^2 \quad (1)$$

式中: N 为向量长度; α 和 β 为权重因子; k 为数据向量两端需要加权的数据量大小.

该网络输出神经元数目较大, 理论上相对小点数仓神经网络输出要复杂, 但考虑其为实时拟合方式实现, 因此对精度要求不是特别高, 综合考量, 设计神经网络结构如图 5 所示. 其中第一层卷积核设定为 1×1 , 主要为了加大维度, 便于后续通道数目调节. 第二层及第三层为 block 块, 设定为三组并行卷积及激活后在通道维度上进行连接, 其中第一组为 1×1 核大小的卷积, 第二组为 1×1 卷积后再进行 3×3 卷积, 第三组为 1×1 卷积后再经过 5×5 卷积. 如此第

一组卷积可尽量保持初始数据信息, 第二组和第三组卷积可有效提取前级输入数据特征并尽量保留边界信息. 为尽量减小网络的复杂度并保持数据规模, 所有卷积均未使用池化层. 第二个 block 块的输出经展平为一维后, 再经过三个全连接层得到输出向量.

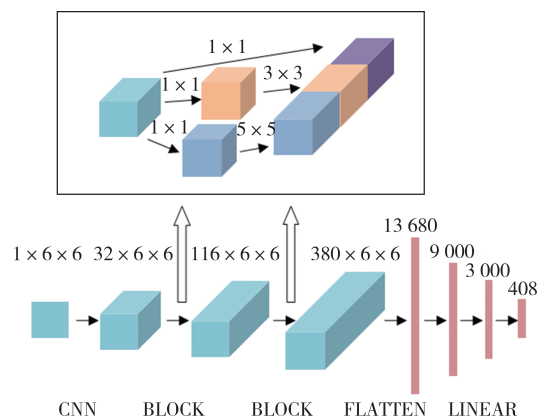


图5 大点数仓神经网络

Fig.5 Big point bins neural network

3 实验结果

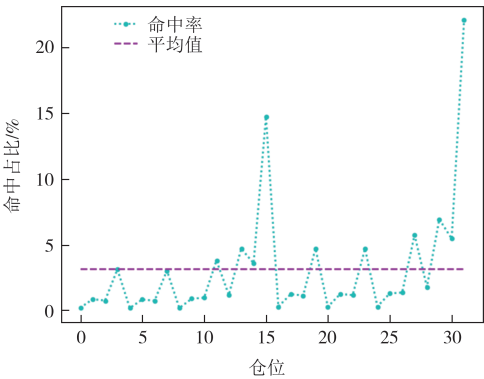
该实验中, 验证平台采用通用验证方法学^[25] (universal verification methodology, UVM) 实现, 利用 VCS 工具进行模拟仿真工作, 机器学习网络基于 Py-Torch 框架搭建而成.

3.1 原始随机下分布

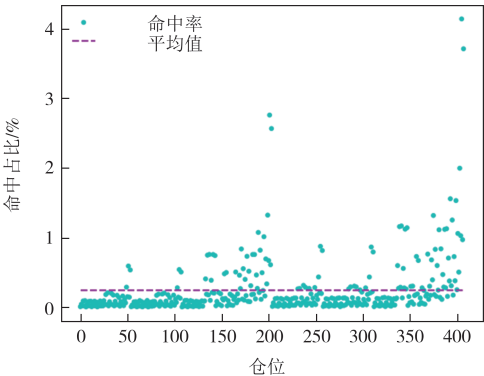
随机测试条件下, 会依据变量的分布特点、场景的使用情况、漏洞风险等综合对激励参数进行设置分配, 两个覆盖点的仓命中率情况分别如图 6(a) 和图 6(b) 所示. 可以看出, 其分布极为散乱, 命中率极大值与极小值间均存在数百倍的差距, 且对激励权重进行调节后效果仍不理想. 若用例较少, 很容易导致部分仓出现未击中情况.

3.2 神经网络训练结果

图 7(a) 和图 7(b) 所示为两种网络训练损失情况的对数坐标形式. 其中小点数仓设定 epoch 次数为 20 000, 为使显示更为直观, 图中数据为间隔 200 点的采样结果. 大点数仓设定 epoch 为 6 000 次, 图中显示为间隔 50 点的采样. 可以看出两种网络的训练损失和测试损失下降趋势一致, 最终分别稳定到 $E-05$ 和 $E-04$ 量级, 且其精度均较高, 模型在实际使用中也能显示其效果良好, 证实两种网络均得到了有效训练.



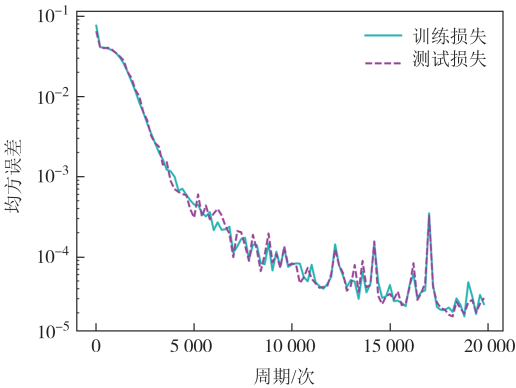
(a)小点数仓原始分布



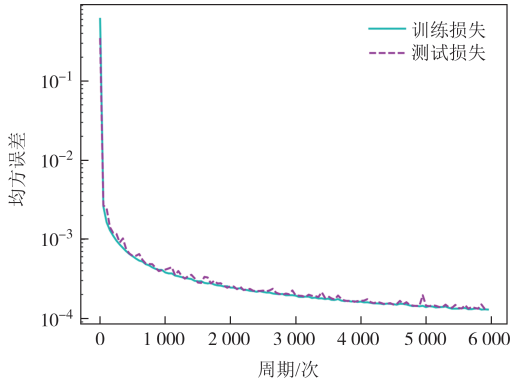
(b)大点数仓原始分布

图6 原始分布情况

Fig.6 Original distribution



(a)小点数仓位训练效果



(b)大点数仓位训练效果

图7 各神经网络训练效果

Fig.7 Training effort of each neural network

3.3 覆盖率分布结果

反向网络对均匀分布向量的预测经仿真后结果如图8所示.

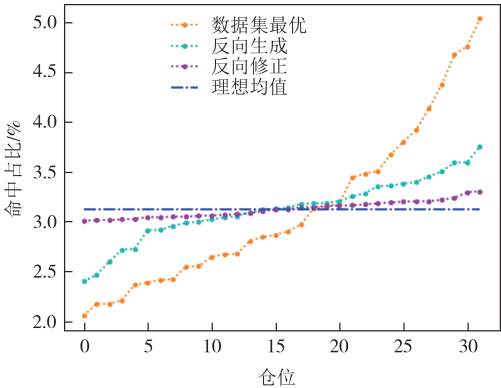


图8 反向网络结果

Fig.8 Reverse network result

可以看出虽然其相对平均,但最小值和最大值之间仍有较大差异.考虑差异的主要原因在于训练集中缺少均衡分布的数据,如图8中图例“数据集最优”显示,差异性最小的数据分布情况也不佳,因此模型对其预测相对较差.对均匀分布的个别数据进行微调修正并进行仿真测试,最终得到更为均衡的命中分布.

对于大点数仓,综合考量拟合效果及运算的复杂度情况,由正向网络生成定量数据集进行拟合,结果如图9所示.其中数据集最优表示从生成数据中选取的方差最小情况,可以看出其与理想均值差异仍较大,“拟合10”表示经过10次拟合,可以看出分布有所改善,经过50次拟合后,其曲线更趋向于理想均值.

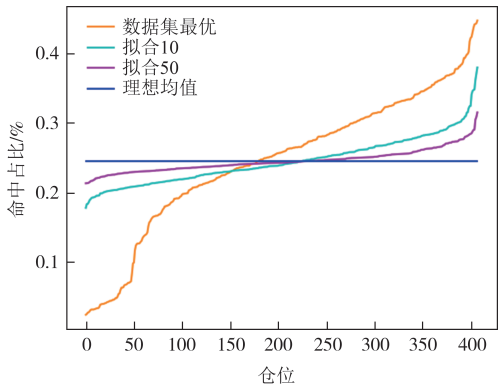


图9 正向网络结果

Fig.9 Forward network result

对于各个结果的分布情况,简化以命中极大值和极小值的占比(极值比)进行比较,表3展示了不同条件下的极值比.对于小仓,极值比由一百多降为

接近 1, 表明分布效果非常理想. 对于大仓, 其极值比由五百多降为低于 1.5, 分布效果较为良好.

表 3 不同情况下的极值比对比
Tab.3 Comparison of extreme value ratio in different cases

仓位	原始随机	反向单生成/ 正向未拟合	优化实现 微调/拟合
小仓极值比	111.23	1.56	1.10
大仓极值比	551.80	18.65	1.48

表 4 所示为等同命中次数下网络预测与随机测试的对比. 网络实验中的每笔事务包含 20 万条测试用例, 以其为基准, 每个小仓和大仓的平均命中数目分别为 6 250 和 490 次, 而随机情况要达到该次数, 所需用例会有数十倍的提升.

表 4 等同命中次数下所需用例数对比
Tab.4 Comparison of case number under equivalent hits

等同命中	网络预测用例/万条	随机测试用例/万条	加速比
小仓	20.00	298.84	14.94
大仓	20.00	512.59	25.63

4 结 论

针对功能覆盖点中各个仓的命中分布极为不均的情况, 根据仓的规模特点, 分别设计了反向预测和正向拟合网络, 可对各种输入向量进行精确预测. 对小点数仓, 反向网络省去了生成预测集的过程, 可直接对预期分布进行预测; 对大点数仓, 多次拟合实现了其向平衡分布的快速收敛. 与随机情况相比, 其分布情况均得到了巨大改善.

综上所述, 本文算法对基于机器学习技术的功能覆盖均衡分布方法进行了探索研究, 大大减少了命中问题引入的测试风险, 为各种覆盖点设置下的功能测试提供了一种新的解决思路.

参考文献

[1] FOSTER H D. Trends in functional verification: a 2014 industry study [C]//2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC). June 8–12, 2015, San Francisco, CA, USA. IEEE, 2015: 1–6.

[2] CHEN W, RAY S, BHADRA J, et al. Challenges and trends in modern SoC design verification[J]. IEEE Design & Test, 2017, 34(5): 7–22.

[3] (美) Bruce Wile, (美) John C. Goss, (美) Wolfgang Roesner. 全

面的功能验证: 完整的工业流程[M]. 沈海华, 等译. 北京: 机械工业出版社, 2010.

WILE B, C.GOSS J C, ROESNER W. Comprehensive functional verification: complete industrial process [M]. Beijing: China Machine Press, 2010. (in Chinese)

[4] MASAMBA N, EDER K, BLACKMORE T. Hybrid intelligent testing in simulation-based verification [C]//2022 IEEE International Conference on Artificial Intelligence Testing (AITest). August 15–18, 2022, Newark, CA, USA. IEEE, 2022: 26–33.

[5] 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. (in Chinese)

[6] GEORGE M P J, MOHAMED O A. A coverage driven test generation methodology using consistency algorithm [C]//Fifth Asia Symposium on Quality Electronic Design (ASQED 2013). August 26–28, 2013, Penang, Malaysia. IEEE, 2013: 27–32.

[7] KRISHNA N V, CHAUDHARY A, SOUMYA J. FGG: feedback guided generation to accelerate functional coverage closure on network-on-chip processors [C]//2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID). January 6–10, 2024, Kolkata, India. IEEE, 2024: 702–707.

[8] GUZEY O, WANG L C. Coverage-directed test generation through automatic constraint extraction [C]//2007 IEEE International High Level Design Validation and Test Workshop. November 7–9, 2007, Irvine, CA, USA. IEEE, 2007: 151–158.

[9] GUO Y, QU W X, LI T, et al. Coverage driven test generation framework for RTL functional verification [C]//2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics. October 15–18, 2007, Beijing, China. IEEE, 2007: 321–326.

[10] BOSE M, SHIN J, 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. May 27–30, 2001, Seoul, Korea. IEEE, 2002: 442–448.

[11] FINE S, ZIV A. Coverage directed test generation for functional verification using Bayesian networks [C]//Proceedings 2003. Design Automation Conference. June 2–6, 2003, Anaheim, CA, USA. IEEE, 2003: 286–291.

[12] KAMATH V, CHEN W, SUMIKAWA N, et al. Functional test content optimization for peak-power validation: an experimental study [C]//2012 IEEE International Test Conference. November 5–8, 2012, Anaheim, CA, USA. IEEE, 2013: 1–10.

[13] WAGNER I, BERTACCO V, AUSTIN T. Microprocessor verification via feedback-adjusted markov models [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(6): 1126–1138.

[14] WANG J, LI H W, LV T, et al. Abstraction-guided simulation using Markov analysis for functional verification [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits

- and Systems, 2016, 35(2): 285–297.
- [15] ZHANG M M, GENG S Q, WANG W S, et al. Probabilistic analysis for sequential circuits verification using Markov chains [J]. IEEE Transactions on Circuits and Systems II : Express Briefs, 2021, 68(1):481–485.
- [16] GUZEY O, WANG L C, LEVITT J, et al. Functional test selection based on unsupervised support vector analysis[C]//2008 45th ACM/IEEE Design Automation Conference. June 8–13, 2008, Anaheim, CA, USA. IEEE, 2008: 262–267.
- [17] ROMERO E, ACOSTA R, STRUM M, et al. Support vector machine coverage driven verification for communication cores[C]//2009 17th IFIP International Conference on Very Large Scale Integration (VLSI-SoC). October 12–14, 2009, Florianopolis, Brazil. IEEE, 2011: 147–152.
- [18] CHANG P H, DRMANAC D, WANG L C. Online selection of effective functional test programs based on novelty detection[C]//2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). November 7–11, 2010, San Jose, CA, USA. IEEE, 2010: 762–769.
- [19] GUO Q, CHEN T S, SHEN H H, et al. On-the-fly reduction of stimuli for functional verification[C]//2010 19th IEEE Asian Test Symposium. December 1–4, 2010, Shanghai, China. IEEE, 2011: 448–454.
- [20] KATZ Y, RIMON M, ZIV A, et al. Learning microarchitectural behaviors to improve stimuli generation quality [C]//2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC). June 5–9, 2011, San Diego, CA, USA. IEEE, 2011: 848–853.
- [21] MANI PARET J G, MOHAMED O A. Optimum domain partitioning to increase functional verification coverage [C]//Sixteenth International Symposium on Quality Electronic Design. March 2–4, 2015, Santa Clara, CA, USA. IEEE, 2015: 419–423.
- [22] HALIM Y M, ISMAIL K A, ABD EL GHANY M A, et al. Reinforcement-learning based method for accelerating functional coverage closure of traffic light controller dynamic digital design [C]//2022 32nd International Conference on Computer Theory and Applications (ICCTA). December 17–19, 2022, Alexandria, Egypt. IEEE, 2023: 44–50.
- [23] GAUR P, ROUT S S, DEB S. Efficient hardware verification using machine learning approach [C]//2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS). December 16–18, 2019, Rourkela, India. IEEE, 2020: 168–171.
- [24] ZHENG X M, ZHAO M X, LUO Q, et al. A chip-level verification method for programmable vision chip based on deep learning algorithms [C]//2020 IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM). October 23–25, 2020, Nanjing, China. IEEE, 2020: 281–284.
- [25] HARSHITHA N B, PRAVEEN KUMAR Y G, KURIAN M Z. An introduction to universal verification methodology for the digital design of integrated circuits (IC's): a review [C]//2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). March 25–27, 2021, Coimbatore, India. IEEE, 2021: 1710–1713.