

AI 赋能实现更快、更好、更高效的芯片验证

2022 年 5 月

作者

Taruna Reddy
资深产品营销经理

概述

芯片开发的每个方面都可谓困难重重，并且难度还在与日俱增，而验证可能是这一过程中最具挑战的阶段。多年来，研究表明，每一代芯片上用于验证阶段的时间和资源占比均呈现不断增加的态势。总体而言，验证阶段工作的增长速度超过了芯片开发项目的其他阶段。团队要求芯片验证能够以更少的时间和资源来实现更好的结果。目前，电子设计自动化 (EDA) 行业已将人工智能 (AI) 技术应用到整个验证阶段的各个步骤来应对这一挑战。本白皮书概述了芯片验证阶段的一些主要挑战，介绍了 AI 技术带来的改变，以及新思科技EDA 解决方案中一些具体功能。

验证的主要挑战

有些人可能认为，任何芯片验证工具、流程或方法的目标都是更快发现更多错误。表面看来确实如此，但现实情况却复杂得多。事实上，验证开发者需要考虑三个维度：设计质量 (QOR)、结果完成时间 (TTR) 和实现成本 (COR)。QOR 最容易理解；验证团队希望找到所有的错误，包括所有错误的类型。绝对数量固然重要，但所发现的错误类型和复杂性也很重要。如果漏掉的错误可能导致系统宕机或芯片制造失败，那么即使找到 99.9% 的错误也是不够的。

我们无法确认所有错误都被发现，但有一些指标可以帮助验证团队确定验证成功并能够进行流片。对于仿真，功能和结构覆盖率指标是最常见的方法。实现高覆盖率有助于建立对设计正确性的信心，而未覆盖的部分则可以为您提供哪些额外的测试进行有效指导。对于静态和形式分析，检查和属性证明的百分比是关键数据指标。即便是无法 100% 证明的属性也可以提供有界证明的信息，来评估分析深度。

结果完成时间 (TTR) 之所以很重要，有几方面的原因。每个芯片项目都有一个由市场机会和竞争压力所决定的目标发布日期。为按时完成，项目时间表会经过仔细计划，但如果覆盖率收敛慢于预期，就可能导致流片和产品出货被延迟。这不会减少利润，甚至在最坏的情况下，还可能让芯片在发布时失去竞争力。因此，要满足一个积极的项目时间表，需要使用验证工具进行快速的测试和分析、顺利的覆盖率收敛和快速调试。如果项目进展落后于计划，通常需要扩大验证团队。但事实早已证明，为工程项目增加人员并不一定会减少完成项目所需的时间。

第三个维度的实现成本 (COR)，包含资源的使用。团队的扩大将会提高项目成本，降低最终产品的潜在盈利能力。购买或在云端租用更多计算服务器同样也会推升成本。此外，每增加一次仿真或形式化运行都会生成更多可能需要分析的结果。如果额外的验证测试在很大程度上与现有的测试重复，那么在这个过程中增加的不必要调试时间反而会降低验证效率。最终，将造成每一轮的芯片设计流程项目中，实现成本大幅提升并推迟完成时间，因此提高设计质量至关重要。

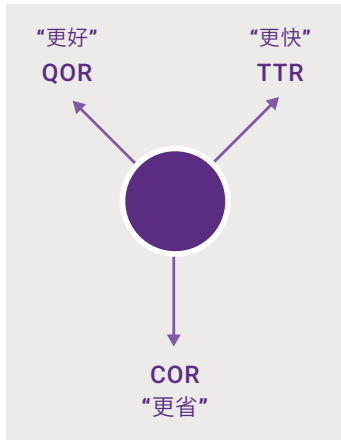


图 1

许多基于机器学习 (ML) 的 AI 技术可以显著改善芯片验证的上述三个维度要求。图2为典型验证流程的高维度视图。在开发者编写寄存器传输级 (RTL) 代码之前，架构团队会构建芯片的虚拟模型并分析系统性能。待做出关键决策后，设计团队将开发 RTL 模型。为此，最好通过使用具有强大语言校验功能的集成开发环境 (IDE) 来捕获编码错误，否则项目后期将耗费大量时间和资源进行修复。修复校验错误后，由开发者完成验证的第一步：运行静态验证工具来检测设计中的结构错误。

形式化验证工具可以提供更深入的分析，尝试证明 RTL 设计的关键属性。这些工具可由开发者运行，也可以与形式化验证专家合作完成。与此同时，验证团队通常会利用现有的验证 IP (VIP) 开发测试平台和模型，运行一系列测试以实现验证计划的目标。此类测试主要在仿真环境中运行，但近年来使用硬件加速平台也变得越来越普遍。如前所述，覆盖率指标是衡量验证进度的主要方法。覆盖率收敛是一个高度迭代过程，这往往也是验证计划中最耗时的部分。

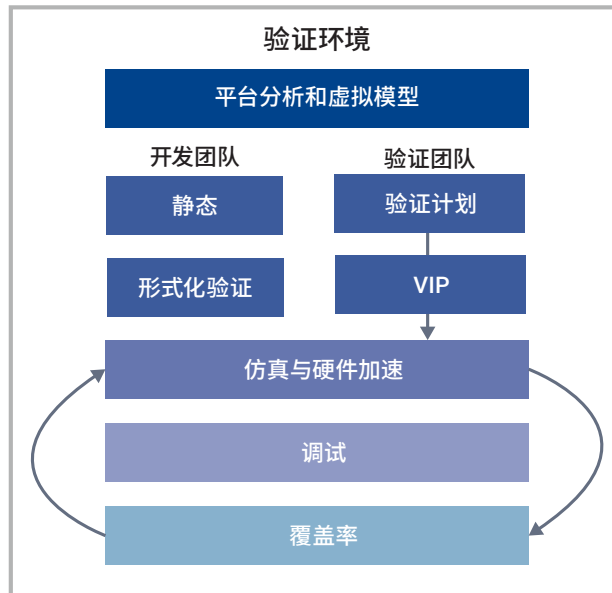


图 2: 芯片验证流程的高维度视图

将 AI 技术应用于静态验证

随着检查类型的增多，静态验证已经成为芯片验证中一个越来越重要的组成部分。当前的静态解决方案必须包含跨时钟域检查 (CDC)、跨复位域检查 (RDC) 和低功耗设计结构检查。通过各类分析，静态验证阶段可以检测并修复常规项目约 10% 的错误。静态工具的主要问题在于它们往往会产生噪音，经常会报告数千个潜在的违规行为。其中一些问题可以通过微调输入设置来解决，例如选择哪些违规行为被报告为错误，哪些只是警告。然而，产生噪音的主要原因是：一个潜在的设计缺陷可能导致许多相关的违规行为。例如，时钟逻辑中的一个错误可能会导致使用该时钟的每个触发器均报告违规行为。为提高调试效率，开发者需要专注于特定问题，这些问题一旦被修复，就会解决掉多种违规行为。

根本原因分析 (RCA) 通过AI技术赋能可以满足这一要求。如图3所示, 在这个分治过程中, 第一步是利用 ML 方法根据共性特征对违规行为进行聚类。在上例中, 与时钟逻辑错误相关的所有违规行为都将合并到一个集群中。因此, 开发者仅查看每个集群中的一项违规并进行修复, 从而解决该集群中的所有违规行为。数以千计的违规行为通常被缩减为数十个集群, 能为开发者节省大量的时间和精力。由于聚类使用的是无监督的ML算法, 无需用户指导即可自动运行。它能够识别违规行为的因果关系, 并随着项目进展不断学习, 继而执行 RCA, 进一步减轻项目团队的负担。图形显示结合即用型 Tcl 脚本, 可引导开发者找到每个集群中违规行为的根本原因。

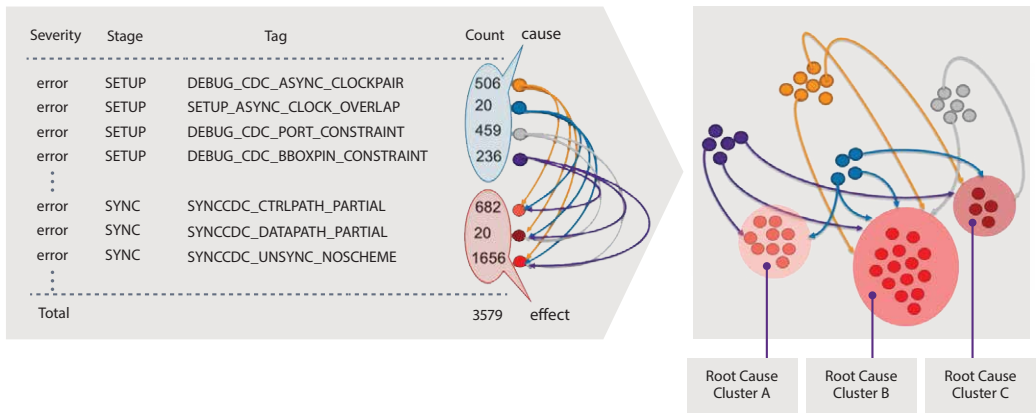


图 3: 静态验证中的违规行为聚类

作为业界领先的静态验证解决方案, 新思科技VC SpyGlass™ 平台和新思科技 VC LP 综合性静态低功耗验证解决方案能够提供所有这些功能, 而且还不仅限于此。SmartGroup 技术可执行高级聚类, 大幅减少需检查的违规行为数量, 而 RCA 则可以加速每个集群的调试。针对违规行为给出纠正建议能改变 RTL 设计, 但往往是对约束文件的完善或增加。将以ML技术为基础的违规行为聚类与RCA 相结合, 可将常规芯片项目静态验证的调试效率提升近 10 倍。

将 AI 技术应用于形式化验证

尽管基于测试平台的仿真是较为著名、使用广泛的验证技术, 但也有其局限性。除非受到仿真激励触发并以可观察到的方式改变预期结果, 否则无法检测并修复设计中的错误。因此, 总有一些错误会因未被激活或未检查其效果而被遗漏。形式化验证可提供针对一组属性的设计综合分析。它不需要激励, 分析过程会考虑所有可能的合法输入序列。这可以检测出在仿真过程中难以触发的深层错误, 通常约占错误总数的 20%。形式化验证还可以证明对于有一个给定的属性不存在进一步的错误, 从而在仿真或硬件加速中提供了无可比拟的置信度。

形式化验证工具通常有数十个求解器或引擎, 以处理设计中需要验证的成百上千个属性。形式化验证工具的性能及其可处理的设计规模取决于引擎的性能和编排方式。在过去 20 年中, 形式化验证技术已经取得了很大的进步。最近, AI 和 ML 技术在提高收敛性和性能方面也取得了巨大成功。新思科技VC Formal™ 是业内首个在引擎编排、回归和调试中利用 ML 技术的形式化验证工具。

引擎编排旨在将引擎与属性正确配对, 在物理计算资源和时间的限制下实现最短运行时间和最佳收敛效果。VC Formal 在处理每个属性时均采用即时强化学习法, 同时学习有效和无效内容, 以指导下一组操作的编排, 这一功能称为“智能策略选择”。此外, 针对每个属性所做的决定会在运行结束时保存到数据库中, 以便利用之前的学习成果优化后续运行结果, 叫做“回归模式加速器”(RMA)。

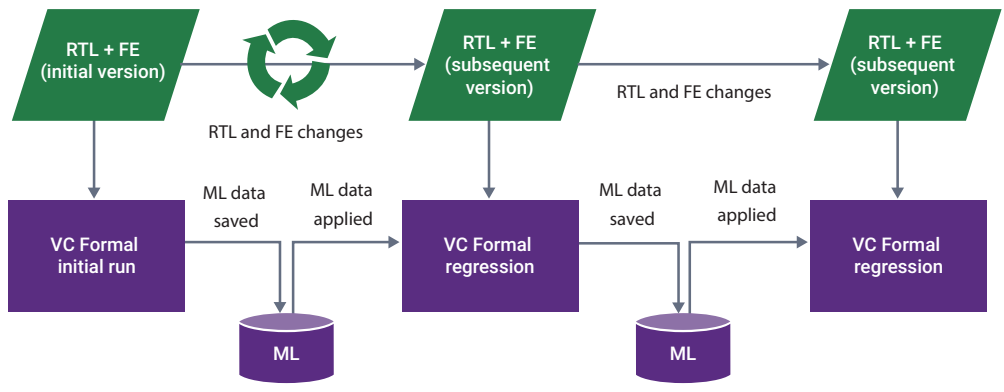


图 4: VC Formal中的回归模式加速 (RMA)

当RTL 设计或形式化测试平台在设计和验证过程中被修改时，通常会每晚或每周回归一部分运行的形式化验证，以确保这些修改不会在设计中引入新的错误。如图 4 所示，VC Formal 每次运行都会读取设计和前一次运行的学习数据库，以决定进行安全维护的属性状态、因更改而需重新运行的属性以及需要编排以获取更多时间和资源的未确定属性。即使对于需要重新验证的属性，也会利用前一次运行的学习经验来指导编排过程，加速第二次运行。基准数据显示，此类 ML 技术可将速度提升 10 倍以上，带来额外的收敛效果，在降低 TTR 和 COR 的同时提高 QOR。图5为以上优势在实际设计中的示例。

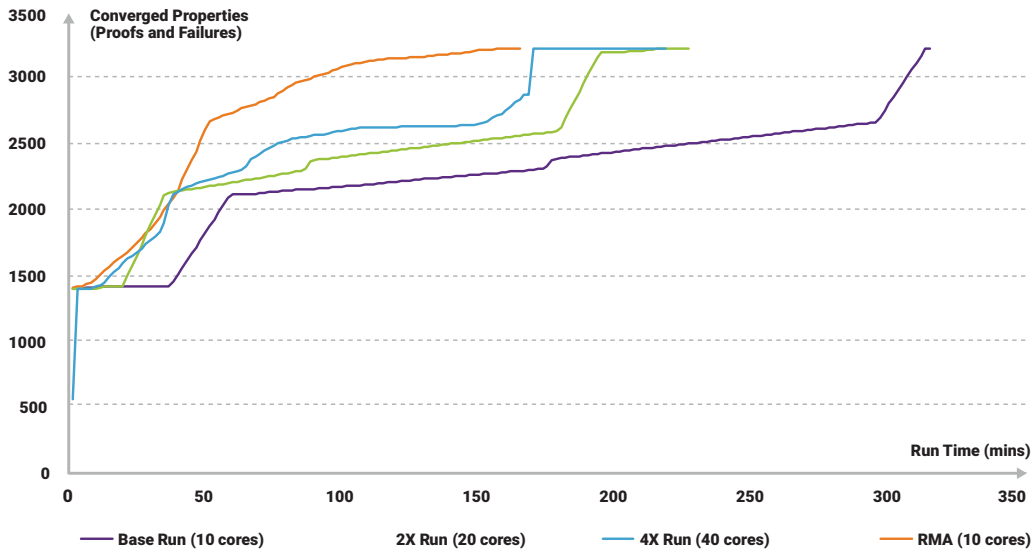


图 5: 利用 RMA 加快属性收敛

将 AI 技术应用于仿真

运行时的性能对于仿真回归也很重要，它可能包含数千个测试并且每晚运行，以对 RTL 设计和测试平台的所有代码更改进行验证。仿真仍然是芯片验证的核心，约占错误发现总数的 65%。有时，为修复错误而对设计所做的更改可能无法正常工作或引入了新的问题，因此频繁的回归对于快速检测问题并保持项目的正常进行至关重要。仿真和回归性能存在数个影响因素，而其中至少两个因素可采用 AI 进行改进。首先是仿真和回归运行时的设置。现代仿真器有许多选项和开关，会对性能产生重大影响。验证开发者需要时间和专业知识来优化特定设计和测试平台的仿真器设置。而随着代码的演进，可能需要调整设置以保持最佳性能。利用 ML 来学习并维护仿真器选项和开关这一自动化过程可以极大地提高回归性能和效率。

新思科技 VCS® 仿真器中的动态性能优化 (DPO) 技术就是利用 AI 来改进仿真性能。如图6中所示，它使用 ML 和基于规则的 AI 技术从先前的回归运行中进行学习，自动调整 VCS 设置以获得最佳性能。这一过程将自动完成，无需用户输入，但如果需要的话，验证开发者也可以选择控制流程的某些方面。例如，他们可以每隔几天打开学习模式，并且在两次运行回归之间使用相同的设置。一般来说，与手动进行仿真器设置相比，DPO可使仿真运行速度提高 1.3 - 2 倍。

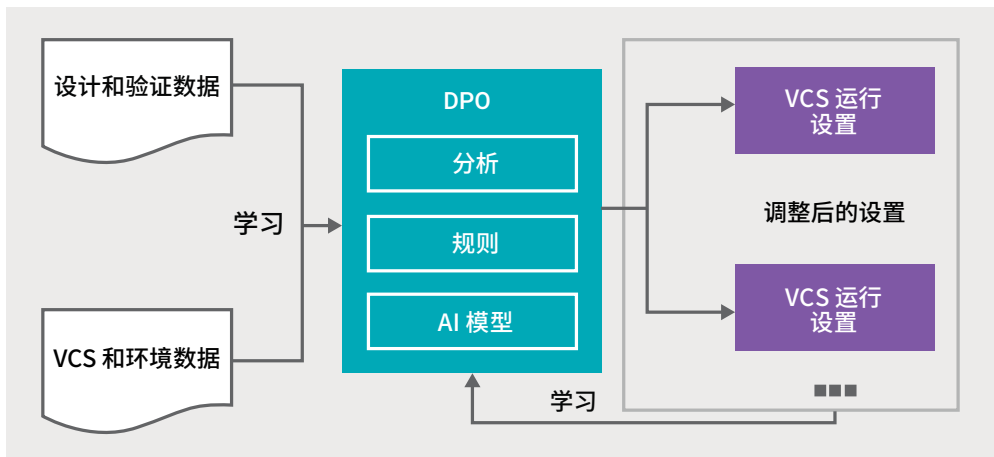


图 6: VCS 中的动态性能优化 (DPO)

对整体回归性能影响最大的是覆盖率收敛所需的时间。以前，验证开发者会审查仿真报告以确定未达到的覆盖率，然后修改测试或编写新测试以覆盖设计中的缺失部分。有了受约束的随机测试平台，他们通常是修改测试平台的约束条件，以便将自动刺激生成集中于未覆盖的情况，但这仍然需要手动完成。重复已经实现的覆盖会浪费大量时间。AI/ML 在验证中的另一项应用可以改善这一情况，而 VCS 提供了相应的解决方案。智能覆盖率优化 (ICO) 可以优化受约束的随机刺激的统计质量，并提供对影响覆盖率的测试问题的分析结果。在最近的芯片项目中，ICO 已经被证明可将覆盖率的收敛速度提高 2-3 倍。验证团队可以在更短的时间内实现更高的总覆盖率，从而缩短时间并节省资源。

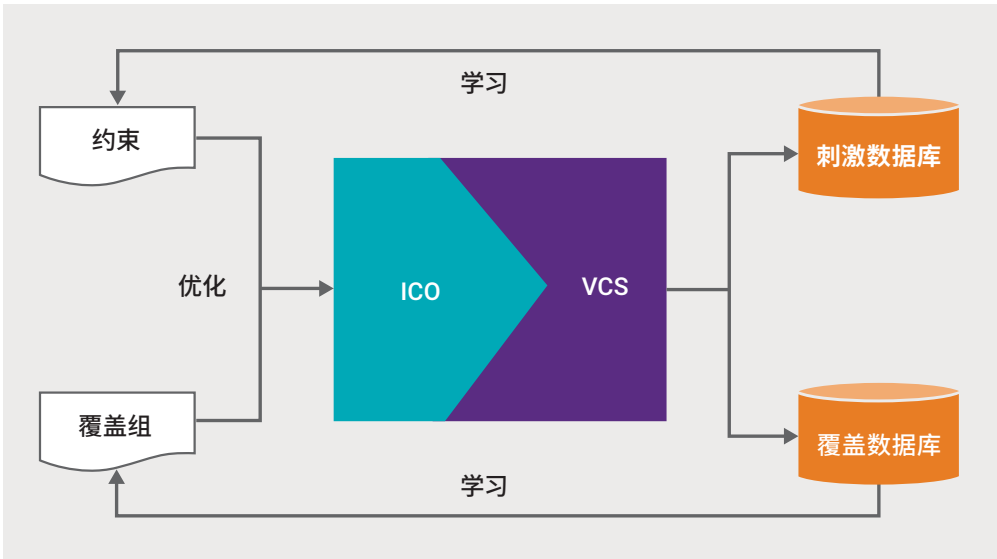


图 7: VCS 中的智能覆盖率优化 (ICO)

将 AI 技术应用于调试

如前所述，在项目期间会运行无数次仿真回归。每次回归失败时，验证团队都必须检查报告并调试失败的原因。RTL 和测试平台 (TB) 代码会随着错误的修复以及新功能的添加和测试而不断变化，回归失败已是家常便饭。但手动处理日常的回归失败给验证工程资源造成了巨大负担。如今，AI/ML 技术可以助力解决这一问题。尽管仿真测试失败的原因通常比静态违规行为要复杂得多，但自动 RCA 的原则仍然适用。

新思科技Verdi® 自动调试系统中的回归调试自动化 (RDA) 功能可自动分箱、探查并发现回归失败的根本原因。RDA 利用 AI 对原始回归失败进行分类和分析，并确定设计和测试平台中失败的根本原因。RDA 缩短了 RCA 时间，并将整体调试效率提高了 2 倍。

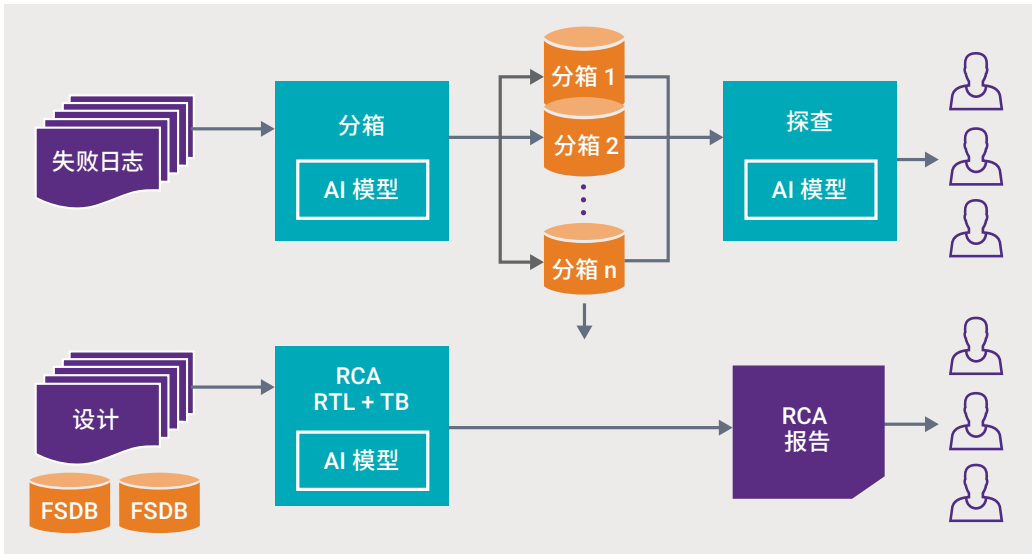


图 8: Verdi 中的回归调试自动化

结论

AI 和 ML 技术在芯片设计和验证流程中的应用与日俱增。单就验证而言, AI/ML 可以加速静态验证的故障分析, 提升形式化验证的性能, 提高仿真效率, 加速覆盖率收敛, 快速轻松地完成仿真调试。图9总结了流程改进为验证团队和整个芯片项目带来的好处。

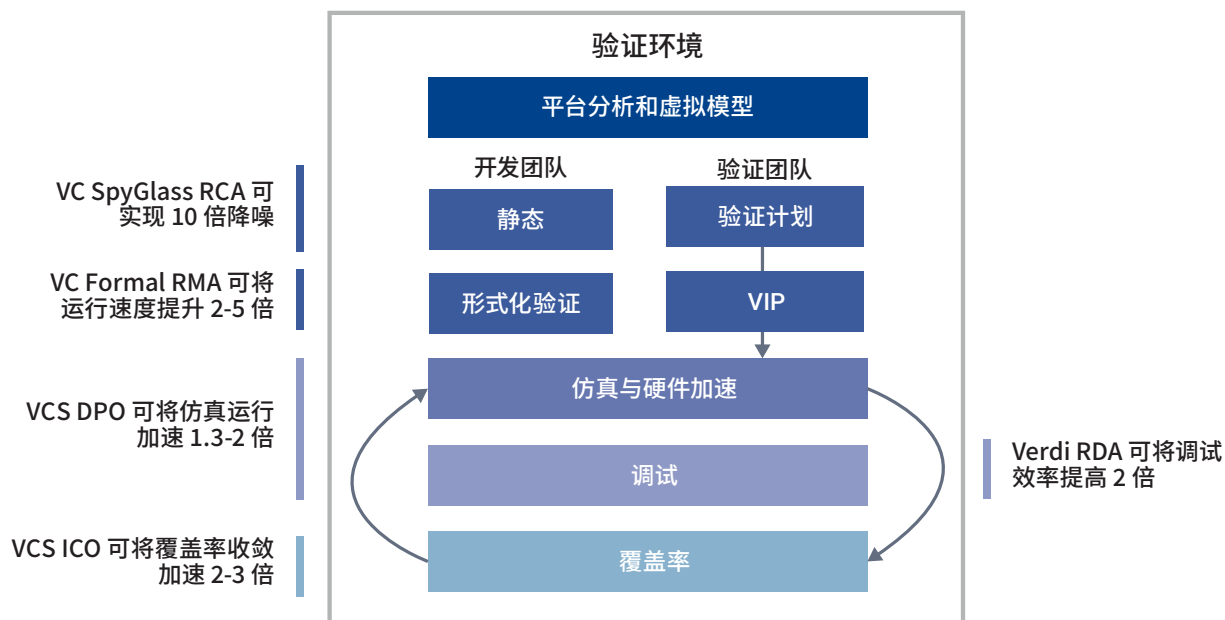


图 9: 新思科技验证流程采用AI/ML 技术带来的优势

新思科技的解决方案通过提供更好的 QOR、更快的 TTR 和更省的 COR 来实现验证团队的目标并解决其所面临的主要挑战。如今, 强大而灵活的 AI/ML 技术已多点开花, 改进验证流程的多个方面。

