基于断言的验证中的自动断言挖掘：技术、挑战与未来方向

Mohammad Reza Heidari Iman、Giorgio Di Natale 和 Katell Morin-Allory 格勒诺布尔-阿尔卑斯大学、法国国家科学研究中心、格勒诺布尔国立综合理工学院*、TIMA，法国格勒诺布尔 38000
{mohammadreza.heidari-iman, giorgio.di-natale, katell.morin-allory}@univ-grenoble-alpes.fr

arXiv:2601.02248v1 [cs.SE] 5 Jan 2026

摘要—功能验证日益依赖基于断言的验证（ABV）。由于高效且有效，ABV 已成为验证硬件设计的一种关键方法。自动断言挖掘器是 ABV 的核心，它们采用不同的技术自动生成断言。本文回顾了最新、先进且应用广泛的断言挖掘器，并对其方法进行了比较分析。本文旨在帮助研究人员和验证从业者了解现有挖掘器的能力与局限。通过指出其不足，本文也为未来开发更强大、更先进的断言挖掘器指明了方向。

索引词——验证、基于断言的验证、自动断言挖掘、自动断言挖掘器、断言

I. 引言

嵌入式系统和微电子元件广泛应用于各种设备和行业，包括汽车和医疗行业，以及一般的安全关键应用 [1, 2]。确保这些系统功能正确至关重要，因为任何故障都可能导致灾难性后果，从经济或生态危害到某些情况下的死亡或重伤不等 [2–4]。防止嵌入式系统发生此类灾难的一种方法是功能验证。

功能验证技术旨在确保硬件设计的行为符合其规格说明中列出的要求集合 [5, 6]。随着硬件设计的规模和复杂性不断增加，验证已成为现代设计流程的关键组成部分 [7, 8]。如今，设计过程中约有 51% 的总成本和时间用于验证 [9]。

为推动功能验证技术的发展，必须深入探索并理解该领域采用的各种方法。这些方法包括静态验证方法（通常称为形式化验证） [10–14], 动态方法（通常称为基于仿真的验证技术） [15–17], 以及半形式化策略（广泛称为基于断言的验证方法） [9, 18–20]。形式化验证侧重于通过严格证明而非仿真来确保功能正确性 [21]。这类方法涵盖模型检验、定理证明和等价性检查等技术 [21]。另一方面，基于仿真的验证通过仿真评估设计行为，并在仿真环境中验证其正确性 [16, 22, 23]。尽管每种验证方法都有其独特优势，但也面临挑战，例如形式化方法存在可扩展性问题，而基于仿真的验证技术则缺乏完备性 [21, 24, 25]。

尽管形式化验证和基于仿真的验证技术展现出巨大潜力，基于断言的验证已逐渐成为广泛采用的复杂数字系统验证方法。基于断言的验证结合形式化方法与基于仿真的方法各自的优势，以确保设计正确性 [26–32]。该技术可应用于多个验证抽象层次，从嵌入事务级测试平台的高级断言，到直接综合进硬件实现的低级断言 [9, 26, 33–35]。基于断言的验证的有效性主要取决于断言本身的质量。这些断言被视为描述设计行为与功能的有效且准确的规则，设计必须遵循这些规则 [28, 36, 37]。

基于断言的验证融合了形式化验证和基于仿真的验证的要素，为验证复杂数字系统提供了一种更高效、更有效的方法 [9]。这种方法依赖于断言，即定义设计预期行为、且在设计运行过程中必须始终为真的布尔表达式 [38]。在基于断言的验证范畴内，断言可通过两种主要方式定义：手动生成断言和自动化断言挖掘 [17, 39, 40]。

手动定义断言需要大量人力专业知识，并深入理解设计的功能 [17, 38, 41, 42]。这一过程通常成本高、耗时长且容易出错 [41, 43, 44]。为应对这些挑战，人们已付出大量努力来实现断言挖掘自动化 [28, 45–57]。这些自动化工具采用多种技术，例如构建有限状态机 (FSMs) [28] ，或利用数据挖掘算法 [58–60]，，在无需人工干预的情况下生成断言。尽管具有潜力，自动化断言挖掘工具仍面临若干局限。其中一个主要问题是，它们无法生成全面覆盖设计行为各方面（包括边界情况）的断言集。缺乏全面覆盖可能导致断言集无法捕捉设计行为的完整范围。当前断言挖掘器的其他不足还包括生成过多断言、冗余（即，描述相同设计

* 格勒诺布尔-阿尔卑斯国立综合理工学院

行为的断言）、不一致（即，彼此冲突的断言）、断言挖掘执行时间过长，以及生成的断言可读性较差 [61, 62]。这些问题共同增加了验证过程更容易出错、更耗时且成本更高的风险。

针对数字设计的自动断言挖掘已在文献中得到广泛研究。 [47] 中的研究介绍了一种断言挖掘器，它利用动态依赖图识别设计中的信号关系，并生成相应的断言。 [48],提出了一种用于断言挖掘的语法引导枚举方法。 [28] 提出的技术利用以有限状态机形式定义的预设模板来提取断言。HARM 是一种基于提示的断言挖掘器，它利用仿真轨迹生成线性时序逻辑（ LTL）断言 [54]。HARM 的扩展版本引入了一种聚类算法，以优化挖掘得到的断言集 [55]。 [46],中介绍的 GoldMine 采用形式化验证和静态代码分析，为寄存器传输级（RTL）设计推导断言。 [53] 提出的方法着重于将时序设计转换为伪组合设计，以此提取断言。此外，作为自动断言挖掘领域的一项最新进展，ARTmine 运用专门设计的数据挖掘技术和算法来生成断言集 [41]。

因此，本文在我们先前发表的综述基础上 [63]，进行了修订和精简，重点介绍自动断言挖掘器的概念基础和方法。主要贡献总结如下：

- 我们对最知名且最新的自动断言挖掘器进行了分析研究和比较。
- 本研究总结了最先进的自动断言挖掘器的优势与局限，为验证领域的研究人员和工程师提供见解，帮助他们通过弥补现有不足来改进未来一代断言挖掘器，并为其工作选择最合适的挖掘器。

本文结构如下：第二节介绍最先进的自动断言挖掘器，第三节对全文进行总结。

II. 自动断言挖掘器

本节将详细介绍该领域研究人员最常使用、最为知名的自动断言挖掘器。我们的目标是突出它们的优势与不足，从而深入了解现有断言挖掘器的局限性，并为未来一代自动断言挖掘器解决这些问题铺平道路。本文讨论的优势与局限性并非仅源自有关断言挖掘器的现有文献，也来自我们多年来在该领域开展研究时，直接或间接使用这些工具及其生成的断言所积累的实践经验。

A. GoldMine [46]

GoldMine 是一款用于为 RTL设计生成断言的自动化工具。其方法包括生成仿真数据、使用决策树算法挖掘断言，并对其进行形式化验证。该工具采用锥形影响分析等静态分析技术，将关注范围缩小到因果变量，从而提高生成断言的相关性。GoldMine 通过多种配置选项调整其方法，例如采样策略和复位信号频率，以处理组合模块和时序模块。该工具在 Rigel RTL设计上进行了评估，结果表明它能够捕捉简单和复杂的设计行为，包括跨多个逻辑层级的时序关系。下面简要介绍 GoldMine 的主要优势与局限性：

主要优势：

- 自动化：GoldMine 通过自动化断言生成流程，减少了人工工作量，并能高效覆盖广阔的设计空间。
- 复杂性处理：它能够捕捉深层的时序关系和组合关系，并揭示非预期但有效的设计行为。
- 设计覆盖率：其生成的断言能够发现设计中尚未探索的区域，有助于提升回归测试质量。

局限性和改进方向：

- 依赖仿真数据：断言质量在很大程度上取决于仿真数据的多样性和质量，而这些数据并非总是可用。
- 依赖启发式方法：当断言挖掘过程中的决策无法得出明确结论时，便会使用启发式方法，这可能导致生成平凡或无关的断言。
- 处理时序属性：时序行为需要进行大量方法上的微调，这限制了该方法开箱即用地应用于复杂时序关系的能力。
- 过度约束的断言：有些断言虽然正确，却可能过于平凡或限制过严，无法带来实际价值。
- 对非 Verilog/RTL 设计的支持有限：GoldMine 仅适用于 Verilog 设计，不支持 SystemVerilog。此外，它只能从 RTL 格式的设计中挖掘断言，因此无法为以门级格式实现的硬件设计挖掘断言。

总之，GoldMine 能够自动执行断言生成，并发掘尚未探索的设计行为，因此是早期 RTL 验证中的一款实用工具。然而，未来的研究应着力解决其局限性，尤其是改善对时序复杂性的处理、减少对仿真数据的依赖，并完善其决策流程，从而提升其稳健性和效率。

B. 动态依赖图 [47]

该断言挖掘器利用动态依赖图（DDG）描述硬件设计中信号之间的关系，并自动生成属性。该方法首先根据一组用例生成动态依赖图，然后从这些图中提取初始

属性。经过细化和抽象步骤后，生成通用属性，再使用模型检查器对其进行验证。该方法旨在仅凭最少的用例数据自动识别相关属性。下面简要介绍动态依赖图的主要优势和局限性：

主要优势：

• 高效用例需求：即使仿真数据极少，该方法仍然有效。仅需一个用例，就足以生成有用的属性，这与其他可能需要大量仿真运行的方法形成对比。

•字级属性：通过利用动态依赖图（DDG），该方法能够使用多种运算符生成详细的字级属性。这为捕捉信号关系提供了更大的灵活性和更高的精度。

• 属性泛化：细化和抽象步骤能够生成不依赖于特定仿真数据的泛化属性，使其广泛适用于不同的设计场景。

•冗余与不一致性：由于我们无法充分获取该挖掘器在部分基准测试中生成的断言，因此只能据此推测，动态依赖图（DDG）在减少冗余和不一致断言方面表现最佳。

局限性与改进空间：

• 可扩展性：随着设计规模的增大，从动态依赖图中提取和处理属性的复杂度可能会导致处理时间延长，从而降低该方法在大型设计中的效率。

• 内部状态探索：该方法主要关注可观测输出信号及其关系，这可能限制其为内部信号状态生成有用属性的能力。扩大分析范围以纳入这些内部状态，将提升该方法在更复杂设计中的适用性。我们只能有限地访问特定基准测试生成的断言；这些数据表明，与其他挖掘器相比，该挖掘器提供的设计行为覆盖率明显较低。

•模型检查器瓶颈：依赖模型检查来验证属性可能会造成瓶颈，尤其是在处理大型设计或大量生成属性时。

总之，基于动态依赖图的属性挖掘方法能够以高效、灵活的方式利用少量仿真数据生成有用属性。该方法擅长创建字级属性并在不同用例间进行泛化，但在可扩展性、内部状态分析和验证速度方面仍面临挑战。解决这些问题将提升该方法在更广泛的硬件验证任务中的实用性。

C. A-TEAM [28]

A-TEAM 是一种断言挖掘工具，旨在从设计在不同抽象层级（包括事务级建模、RTL 和门级设计）下的执行轨迹中提取断言。它采用模板驱动方式，允许用户定义自定义模板来指导断言生成，因此比受固定模式限制的传统挖掘器更灵活。A-TEAM 遵循三个步骤的方法：（1）利用执行轨迹和模板生成候选断言；（2）通过注入可测试故障并测量其故障覆盖率来评估断言；（3）采用基于整数线性规划（ILP）的方法精简断言集，仅保留能够最大限度提高覆盖率的断言。最终得到一组精简的断言，能够有效捕捉 DUV行为。接下来，我们简要介绍 A-TEAM 的主要优势和局限：

主要优势：

• 可定制且灵活的挖掘：使用用户定义的模板，使 A-TEAM 能够探索多样的断言模式，而不受预定义结构的限制，因此可适应不同的抽象层级。

• 紧凑的断言与高覆盖率：与基于决策树的挖掘器相比，A-TEAM 生成的断言数量更少、结构更紧凑，同时实现了更高的故障覆盖率。

• 适用范围广：A-TEAM 可处理来自不同 DUV 抽象层级的执行轨迹，从而扩大其应用范围。

• 设计行为覆盖率：我们只能有限地获取该挖掘器针对少数基准测试生成的断言。结果显示，尽管该挖掘器的设计行为覆盖率不及新一代挖掘器，但仍能提供有竞争力的覆盖率。

局限性与改进方向：

• 计算开销高：基于 ILP 的最小化和前件论证会显著增加执行时间，通常比更简单的方法耗时更长。

• 依赖用户定义模板：挖掘器的有效性在很大程度上取决于模板的质量，要取得最佳结果，需要用户具备丰富的专业知识。

• 时序探索有限：尽管 A-TEAM 支持多种时序模板，但其捕捉复杂时序依赖关系的能力仍有进一步提升空间。

•可扩展性挑战：计算需求和基于启发式的决策可能导致其难以高效扩展，以适用于大型复杂 DUV。

总之，A-TEAM 是一种多功能且有效的断言挖掘工具，在覆盖率和挖掘所得断言的紧凑性方面优于传统方法。然而，它对用户专业知识的依赖、较长的执行时间以及对复杂时序行为的探索有限，为未来研究提供了改进方向，以增强其可扩展性、降低计算开销，并提高断言生成的自动化程度。

D. HARM [54]

HARM（基于提示的断言挖掘器）及其扩展版本 [55], 可视为 A-TEAM 的增强版本，都是稳健的、模板驱动的断言挖掘工具，专为硬件验证而设计。HARM 方法的核心是利用用户定义模板，从输入轨迹中高效、灵活地挖掘断言。通过采用三级并行化策略、可定制的排序指标和过滤机制，HARM

实现了显著的计算速度和可扩展性。此外，该挖掘器允许用户定义领域专用指标，为断言评分和排序，从而使其输出适应特定上下文的需求。下文简要介绍 HARM 的主要优势和局限性：

主要优势：

•可定制且灵活的方法：HARM 的基于模板的方法允许用户根据特定领域定制挖掘过程，确保生成的断言与验证环境相关。

• 高度并行化的算法：三级并行策略（跨评估单元、模板排列和模板）充分利用 CPU 核心，实现显著加速，尤其适用于大型数据集。

• 用户定义的上下文感知指标：HARM 使用户能够定义用于筛选和排序断言的指标，从而对结果进行精细控制。这使其能够适应各种验证需求和场景。

• 可扩展性与效率：实验结果表明，HARM 能够随着输入规模的增加有效扩展，并且在速度和断言质量方面均优于最先进的挖掘器。

• 安全环境：我们在安全和硬件木马检测场景中使用 HARM 的断言开展实验，结果表明 HARM 在检测硬件木马方面具有显著的适用性。

局限性与改进方向：

• 模板依赖：基于模板的方法虽然灵活，但也依赖用户定义有意义的模板。无效或不完整的模板可能导致生成的断言质量不佳或无关。

•指标校准的复杂性：排序系统虽然灵活，但要求用户从众多校准函数配置中进行选择。这种复杂性可能给经验较少的用户带来挑战，因此可通过自动化或引导式指标调优加以改善。

• 对未知断言的探索有限：HARM依赖模板，这本身就限制了其发现预定义模板范围之外断言的能力，因此可能会遗漏新颖或意料之外的行为。

• 排序偏差：排序公式会提高在所有指标上得分较高的断言的权重，但会惩罚在任何单项指标上得分较低的断言。虽然这有助于筛选低质量断言，但可能会低估那些具有独特但单一优势的断言。

•评估仅限于轨迹：HARM的方法仅根据给定的输入轨迹评估断言。这限制了其在轨迹不完整或无法充分反映系统行为的场景中的适用性。

• 生成的断言数量较多：我们的实验分析表明，与所有其他挖掘器相比，HARM 生成的断言数量显著更多，但设计行为覆盖率未必更高。这可能会导致验证过程更长。

总之，HARM 是一款强大且高效的工具，可用于硬件验证中的基于提示的断言挖掘。其模板驱动方法、三级并行化和上下文感知排序机制，使其成为该领域颇具竞争力的挖掘器。不过，它依赖用户定义模板、指标校准过程复杂且计算开销较大，这些问题表明其易用性和性能仍有提升空间。若能解决这些局限，例如减少模板依赖、实现指标调优自动化，并拓展对非模板断言的探索，HARM 将更加易用且灵活，并为断言挖掘方法的发展铺平道路。

E. ARTmine [41]

ARTmine 是一款自动断言挖掘器，旨在为硬件验证生成时序断言集。ARTmine 的核心方法围绕数据挖掘算法展开，用于从硬件设计的仿真轨迹中挖掘关联规则，以识别关键的设计行为。随后，这些挖掘出的关联规则会被转换为用于验证的时序断言。通过调整挖掘器的参数，挖掘过程既高效又有效，并特别关注边界情况。这确保该工具生成的断言集数量精简，同时高度相关且准确。与 HARM 和 GoldMine 等其他断言挖掘工具相比，ARTmine 在变异体检测和断言生成方面展现出强大的能力。下文将简要介绍 ARTmine 的主要优势和局限：

主要优势：

•高效的断言生成：ARTmine 旨在生成数量精简、与验证高度相关且有效的断言。通过关注边界情况，并利用工具中可配置参数采取受控的方法，ARTmine 避免了断言过度生成。这对于防止验证过程被不必要的断言压垮至关重要，而这往往是其他断言挖掘器的局限。

• 有效的变异体检测：ARTmine 在检测注入的变异体方面表现出色。这凸显了它能够识别传统断言挖掘技术可能无法捕捉的重要设计行为。该工具着重关注边界情况，进一步增强了检测能力，因此在复杂硬件系统的验证中尤为有价值。

• 可扩展性：ARTmine 能够有效处理大规模设计。即使面对组件数量众多的设计，该工具也能高效处理仿真轨迹并生成断言。这种可扩展性是将 ARTmine 应用于真实硬件验证任务的一项显著优势。

• 对时序模式的支持：ARTmine 支持多种时序模式，因此能够满足广泛的验证需求。这种灵活性使 ARTmine 适用于各种设计

场景，从简单的验证任务到更复杂的验证任务。

•安全背景：我们对 ARTmine 在安全和硬件木马检测领域中的断言进行了实验分析，结果凸显了其在识别硬件木马方面的强大适用性。

局限性与改进方向：

•时序模式支持有限：尽管 ARTmine 支持若干关键时序模式，但其适用范围仅限于特定的预定义模式。当前实现并不直接支持硬件验证中常见的其他时序模式。扩展 ARTmine，使其能够处理更广泛的时序模式，可以进一步提升其适用性，使其能够适应各种硬件验证场景。

• 数据挖掘阈值：ARTmine 旨在有效处理边界情况，但该工具依赖预定义数据挖掘阈值，可能会导致一些重要但罕见的设计行为被排除。

•超大型设计的可扩展性挑战：尽管 ARTmine 在中大型设计中展现出良好的可扩展性，但应用于极其庞大的设计时仍可能面临挑战，尤其是当某个预定义时序模板涉及显著更多的输入/输出或状态变量时。随着变量数量增加，挖掘出的断言数量及其相关计算开销可能呈指数级增长。优化底层算法以更好地适应超大型设计，将使 ARTmine 更适用于尖端硬件系统，尤其是在现代片上系统（SoC）和超大型设计场景中。

总之，ARTmine 是一种强大而高效的自动断言挖掘器，擅长为硬件验证生成时序断言。它能够处理各种时序模式，与其他挖掘器相比，使用更少的断言检测出变异体，并扩展至大型设计，因此是硬件验证任务中的宝贵工具。不过，它也存在一些局限，例如支持的时序模式种类有限，并且依赖预定义的阈值。这些局限为未来研究提供了机会，以扩展 ARTmine 的能力、改进其处理复杂行为的能力，并增强其在下一代硬件系统中的可扩展性。若能在这些方面进一步改进，ARTmine 有望成为一种强大的工具，在广泛的设计环境中实现全面而高效的硬件验证。

III. 结论

在本文中，我们调研并分析了最知名且应用最广泛的自动断言挖掘器，以更好地探究其优势与不足。我们的分析总结了它们的优缺点，重点指出当前最先进的自动断言挖掘器所面临的局限。通过聚焦这些局限，我们旨在为自动断言挖掘器的未来改进提供指导，最终提升硬件设计的功能验证水平。

致谢

本研究部分由欧盟 Horizon Europe 研究与创新计划下“Resilient Trust”项目资助，资助协议编号为 101112282。

参考文献

[1] S. K. Jena和M. Srinivas，《多核处理是否适用于嵌入式汽车系统》，发表于《2012年国际网络赋能分布式计算与知识发现会议》，2012年，第315–322页。[2] M. R. H. Iman和P. Yaghmaie，《一种多核处理器中的软件控制流检查技术》，《嵌入式系统国际期刊》，第13卷，第2期，第136–147页，2020年。[3] C. E. Salloum、M. Elshuber、O. Höftberger等，《across mpsoc——面向安全关键嵌入式系统的新一代多核处理器》，发表于《2012年Euromicro数字系统设计会议》，2012年，第105–113页。[4] D. Zuccala、S. Hon、M. R. H. Iman等，《复杂数字系统中故障传播的形式化分析》，发表于《国际系统设计形式化方法与模型研讨会论文集》，丛书：MEMOCODE ’25。美国纽约州纽约市：美国计算机协会，2026年，第67–71页。 [在线]。可访问：https://doi.org/10 1145 3742875 3754691 。/。T. Kropf，《形式化硬件[5]验证导论：设计正确电路和系统的方法与工具》，第1版。柏林、海德堡：施普林格出版社，1999年。[6] M. R. H. Iman，《使用数据挖掘算法对硬件设计进行基于断言验证的高级技术》，第1版。Cham：Springer Nature Switzerland，2025年。[7] W. Chen、S. Ray、M. Abadir等，《现代SoC设计验证的挑战与趋势》，《IEEE设计与测试》，第PP卷，第1–1页，2017年8月。[8] M. R. Heidari Iman，《功能验证断言自动生成》。Cham：Springer Nature Switzerland，2025年，第27–43页。[9] H. Witharana、Y. Lyu、S. Charles等，《基于断言的硬件验证综述》，《ACM计算调查》，第54卷，第11s期，第1–33页，2022年。[10] T. Grimm、D. Lettnin和M. Hübner，《安全关键片上系统形式化验证技术综述》，《电子学》，第7卷，第6期，第81页，2018年。[11] D. Große、G. Fey和R. Drechsler，《融合调试与覆盖率分析的增强型电路形式化验证流程》，《EASST电子通信》，第62卷，2013年。[12] R. Drechsler和G. Fey，《形式化验证与鲁棒性检查的结合——技术与挑战》，发表于《第13届IEEE电子电路与系统设计和诊断研讨会论文集》。IEEE，2010年，第4–4页。[13] H. Riener和G. Fey，《Faust：形式化验证、自动调试和软件测试生成框架》，发表于《国际软件模型检测SPIN研讨会论文集》。施普林格，2012年，第234–240页。[14] G. Fey，《利用形式化验证技术评估系统漏洞》，发表于《国际计算机科学数学与工程方法博士生研讨会论文集》。施普林格，2011年，第47–56页。[15] H. Rostami、M. Hosseini、A. Azarpeyvand等，《面向基于断言验证的自动高功能覆盖率激励生成》，发表于《2024年IEEE第30届在线测试与鲁棒系统设计国际研讨会（IOLTS）》，2024年，第1–7页。[16] M. Loghi、T. Margaria、G. Pravadelli等，《嵌入式系统的动态与形式化验证：比较综述》，《国际并行编程期刊》，第33卷，第585–611页，2005年。[17] M. R. Heidari Iman，《通过数据挖掘算法增强硬件设计中的基于断言验证》。TalTech出版社，2024年，序列号：37/2024。[在线]。可访问：https://doi.org/10 23658 37 2024 。/taltech。/ K. Datta和P.[18] P. Das，《使用HDVL进行基于断言的验证》，发表于《第17届国际超大规模集成电路设计会议论文集》。论文集。IEEE，2004年，第319–325页。[19] Y. Li、W. Wu、L. Hou等，《数字集成电路基于断言验证研究》，发表于《2009年第二届国际信息与计算科学会议论文集》，第2卷。IEEE，2009年，第25–28页。[20] K. Alatoun、B. Shankaranarayanan、S. M. Achyutha等，《利用基于断言的安全监视器进行SoC可信验证》，发表于《2021年第22届国际电子设计质量研讨会（ISQED）论文集》。IEEE，2021年，第496–503页。[21] R. Drechsler，《高级形式化验证》。施普林格，2004年。

[22] G. Fey和R. Drechsler，《利用形式化方法改进基于仿真的验证》，发表于《ASP-DAC 2004：亚洲及南太平洋设计自动化会议2004（IEEE目录号04EX753）论文集》，2004年，第640–643页。[23] N. Bombieri、F. Fummi和G. Pravadelli，《用于验证的硬件设计与仿真》，载于《硬件验证形式化方法》，M. Bernardo和A. Cimatti编。柏林、海德堡：Springer Berlin Heidelberg，2006年，第1–29页。[24] L. Zhao、S. Chen和Y. Wang，《基于图的规格说明的基于仿真的硬件验证》，《工程中的数学问题》，第2018卷，第10页，2018年。 [在线]。可访问：https://www.proquest.com/scholarly-journals/simulation-based-hardware-verification-with-graph/docview/2002921501 2 /[25] se- A. Kamkin和M. Chupilko，《现代硬件基于仿真的验证技术综述》，《编程与计算机软件》，第37卷，第147–152页，2011年5月。[26] H. Sohofi和Z. Navabi，《面向系统级设计的基于断言验证》，发表于《第十五届国际电子设计质量研讨会论文集》。IEEE，2014年，第582–588页。[27] S. Lammermann、J. Ruf、T. Kropf等，《迈向异构系统设计的基于断言验证》，发表于《2010年欧洲设计、自动化与测试会议暨展览会论文集（DATE 2010）》。IEEE，2010年，第1171–1176页。[28] A. Danese、N. D. Riva和G. Pravadelli，《A-TEAM：自动基于模板的断言挖掘器》，发表于《2017年第54届ACM/EDAC/IEEE设计自动化会议（DAC）论文集》，2017年，第1–6页。[29] S. Germiniani，《从边缘到云的完整基于断言验证框架》，博士学位论文，维罗纳大学，意大利维罗纳，2023年。[30] L. Pierre、F. Pancher、R. Suescun等，《工业环境中基于断言验证的有效性》，载于《工业关键系统形式化方法》，C. Pecheur和M. Dierkes编。柏林、海德堡：Springer Berlin Heidelberg，2013年，第78–93页。[31] P. Yeung和K. Larsen，《SoC设计的实用基于断言的形式化验证》，发表于《2005年国际片上系统研讨会论文集》，2005年，第58–61页。[32] P. Gurha和R. R. Khandelwal，《AMBA-AHB的SystemVerilog基于断言验证》，发表于《2016年国际微电子与电信工程会议（ICMETE）论文集》，2016年，第641–645页。[33] A. M. Gharehbaghi、B. H. Yaran、S. Hessabi等，《一种面向系统级设计的基于断言验证方法》，《计算机与电气工程》，第33卷，第4期，第269–284页，2007年，硬件/软件片上系统协同设计：方法与应用。[在线]。可访问：https://www.sciencedirect.com/science/article/pii/S0045790607000201[34] B. Niemann和C. Haubelt，《事务级模型的基于断言验证》，发表于《电路与系统建模和验证的方法与描述语言论文集》，2006年。[在线]。可访问：https://api.semanticscholar.org/CorpusID:16944153[35] A. Habibi、S. Tahar、A. Samarah等，《利用事务级建模进行高效的基于断言验证》，发表于《欧洲设计、自动化与测试会议论文集》，第1卷，2006年，第1–6页。[36] M. R. Heidari Iman、R. Drechsler、C. K. Jha等，《专题论文：功能验证技术在硬件可信性中的应用》，发表于《2025年IEEE超大规模集成电路与纳米技术系统缺陷与容错国际研讨会（DFT）》，2025年，第1–8页。[37] M. R. H. Iman、S. Ahmadi-Pour、R. Drechsler等，《借助断言检测处理器漏洞：RISC-V案例研究》，发表于《2024年IEEE北欧电路与系统会议（NorCAS）》，2024年，第1–7页。[38] M. Boulé和Z. Zilic，《生成硬件断言检查器：用于硬件验证、仿真、制造后调试和在线监控》。施普林格，2008年。[39] M. R. Heidari Iman，《断言的自动评估与最小化》。Cham：Springer Nature Switzerland，2025年，第45–67页。[40] A. Danese、F. Filini、T. Ghasempouri等，《控制信号断言的自动生成与判定：一种基于时间窗口的方法》，载于《VLSI-SoC：面向可靠性、安全性和低功耗的设计》，Y. Shin、C. Y. Tsui、J.-J. Kim等编。Cham：施普林格国际出版，2016年，第193–221页。[41] M. R. Heidari Iman、G. Jervan和T. Ghasempouri，《ARTmine：用于硬件验证的时序行为自动关联规则挖掘》，发表于《2024年欧洲设计、自动化与测试会议暨展览会（DATE）》，2024年，第1–6页。[42] 属性生成。多德雷赫特：施普林格荷兰，2008年，第75–97页。[43] S. Katz、O. Grumberg和D. Geist，《我写的属性够多了吗？一种比较规格说明与实现的方法》，发表于ACM CHARME，1999年，第280–297页。[44] M. R. Heidari Iman，《面向安全验证的断言自动生成》。Cham：Springer Nature Switzerland，2025年，第69–93页。[45] S. Hertz、D. Sheridan 和 S. Vasudevan，“借助静态分析指导挖掘硬件断言”，《IEEE集成电路与系统计算机辅助设计汇刊》，第32卷，第6期，第952–965页，2013年。[46] S. Vasudevan、D. Sheridan、S. Patel 等，“Goldmine：利用数据挖掘和静态分析自动生成断言”，载于 DATE，2010年，第626–629页。[47] J. Malburg、T. Flenker 和 G. Fey，“利用动态依赖图进行属性挖掘”，载于 ASP-DAC，2017年1月，第244–250页。[48] G. Martino 和 G. Fey，“时序属性的语法引导枚举”，载于 2019 FDL，2019年，第1–8页。[49] M. Soeken、U. Kühne、M. Freibothe 等，“用于总线桥接器形式化验证的属性自动生成”，载于第14届 IEEE电子电路与系统设计和诊断国际研讨会论文集，2011年，第417–422页。[50] J. Malburg、T. Flenker 和 G. Fey，“如何从少量用例中生成优质属性”，载于 IEEE国际验证与安全研讨会论文集，2016年7月。[51] F. Rogin、T. Klotz、G. Fey 等，“通过属性自动生成实现高级验证”，《IET计算机与数字技术》，第3卷，第4期，第338–353页，2009年。 [在线]。获取地址：https://doi.org/10.1049/iet- cdt.2008.0110[52] ——，“面向硬件设计的复杂属性自动生成”，载于欧洲设计、自动化与测试会议论文集，DATE ’08系列。美国纽约州纽约：美国计算机协会，2008年，第545–548页。 [在线]。获取地址：https://doi.org/10 1145 1403375 1403506 。/。M. R. Heidari Iman、J. Raik、M. Jenihhin 等，“一种自动化挖掘紧凑且准确断言集的方法”，载于 2021年 IEEE Nordic Circuits and Systems Conference (NorCAS)，2021年，第1–7页。[54] S. Germiniani 和 G. Pravadelli，“HARM：基于提示的断言挖掘器”，《IEEE集成电路与系统计算机辅助设计汇刊》，第41卷，第11期，第4277–4288页，2022年。[55] ——，“利用聚类和决策树算法挖掘含非布尔表达式的 LTL 断言”，载于 2022年 IFIP/IEEE 第30届超大规模集成电路国际会议（VLSI-SoC）论文集，2022年。[56] S. Vasudevan、L. Liu 和 S. Hertz，《GoldMine中的断言挖掘算法比较研究》。Cham：施普林格国际出版，2019年，第609–645页。[57] D. Sheridan、L. Liu 和 S. Vasudevan，“Goldmine：设计验证中的自动断言生成与覆盖率闭合”，2011年。[58] M. R. H. Iman、P. Chikul、G. Jervan 等，“通过时序关联规则挖掘检测异常文件系统活动”，载于第9届 ICISSP，2023年，第733–740页。[59] A. Roberts、M. R. Heidari Iman、M. Bellone 等，“ADAssure：自动驾驶控制算法的调试方法”，载于 2024年欧洲设计、自动化与测试会议暨展览会（DATE），2024年，第1–6页。[60] M. Shahin、M. Bertl、M. R. H. Iman 等，“显著因素提取：结合逻辑回归与 Apriori关联规则挖掘的方法”，载于《系统中的机器学习方法》，R. Silhavy 和 P. Silhavy 编。Cham：Springer Nature Switzerland，2024年，第295–311页。[61] M. R. Heidari Iman、J. Raik、G. Jervan 等，“IMMizer：一种用于精简断言集的创新高性价比方法”，载于 2022年第25届 Euromicro数字系统设计会议（DSD）论文集，2022年，第1–8页。[62] M. R. Heidari Iman、J. Raik、M. Jenihhin 等，“一种自动化的高质量断言集挖掘方法”，《微处理器与微系统》，第97卷，第104773页，2023年。[63] M. R. H. Iman、G. Di Natale 和 K. Morin-Allory，“自动断言挖掘器综述”，载于 2025年 IEEE 第28届电子电路与系统设计和诊断国际研讨会（DDECS），2025年，第55–60页。