

基于 vManager 的大规模 IC 验证自动化解决方案

植 玉, 梁其锋, 石义军, 贺志强

(深圳市中兴微电子技术有限公司, 广东 深圳 518054)

摘要: 介绍了验证管理工具 vManager, 通过 Python 调用 vAPI 接口与企业级的产品需求管理工具 Microsoft TFS 和用户数据后台对接, 实现了从自动创建验证需求框架(即 vPlan)自动执行回归验证, 自动提取验证结果反标回 Microsoft TFS 中的需求状态, 自动提取验证结果呈现到验证看板的自动化验证管理全流程。方案旨在自动化、规范化地实现验证需求到 vPlan 的同步, 验证回归状态和覆盖率的实时汇总, 实现验证的高效率和高透明度, 需求跟踪达到滴水不漏。该方案还采用了 vManager 最新一代的 High Available 模式, 可实现跨地域的多团队合作与数据共享, 并且部署了多引擎验证工具包括 Xcelium、JasperGold 和 Palladium 的验证管理, 实现了多维度的验证数据汇总。目前该方案已经部署到真实的研究环境中, 为 vManager 在国内比较领先的应用, 为业内提供“跨地域合作+多个仿真引擎”的大规模验证方案提供了非常有价值的参考。

关键词: 关键验证自动化管理; vManager; vAPI; Python

中图分类号: TN402

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.219802

中文引用格式: 植玉, 梁其锋, 石义军, 等. 基于 vManager 的大规模 IC 验证自动化解决方案[J]. 电子技术应用, 2021, 47(8): 48-51.

英文引用格式: Zhi Yu, Liang Qifeng, Shi Yijun, et al. Large-scale IC verification automation solution based on vManager[J]. Application of Electronic Technique, 2021, 47(8): 48-51.

Large-scale IC verification automation solution based on vManager

Zhi Yu, Liang Qifeng, Shi Yijun, He Zhiqiang

(Sanechips Technology Co., Ltd., Shenzhen 518054, China)

Abstract: This paper introduces the verification management tool vManager, which calls the vAPI interface through Python to interconnect with the enterprise-level product requirements management tool Microsoft TFS and user data, realizes the automatic implementation of regression verification from automatic creation of the verification requirements framework (i.e. vPlan), automatically annotates the verification results to the requirements state in Microsoft TFS, and automatically extracts the verification results and presents it to the automated verification management process of the verification dashboard. The purpose of this scheme is to automate and standardize the synchronization of verification requirements to vPlan, real-time summary of regression status and coverage, achieve high efficiency and transparency of verification, and make the requirement tracking perfect. The scheme also adopts the latest generation High-Available mode of vManager to realize cross-regional multi-team cooperation and data sharing. In addition, the multi-engine verification tools including Xcelium, JasperGold and Palladium are deployed to realize multi-dimensional verification data collection. Currently, the solution has been deployed in a real R&D environment, it provides a valuable reference for vManager's leading application in China and for large-scale verification solutions of cross-regional cooperation, multiple simulation engines in the industry.

Key words: key verification automation management; vManager; vAPI; Python

0 引言

本文介绍了验证管理工具 vManager, 通过 Python 调用 vAPI 接口与企业级的产品需求管理系统 Microsoft TFS 和用户数据后台对接, 实现了从自动创建验证需求框架(即 vPlan), 自动执行回归验证, 自动提取验证结果反标 Microsoft TFS 中的需求状态, 自动提取验证结果呈现到验证看板的自动化验证管理全流程。

本方案旨在自动化、规范化地实现从验证需求到

vPlan 的同步, 验证回归状态和覆盖率数据的实时汇总, 实现验证的高效率与高透明度, 需求跟踪达到滴水不漏。该方案还采用了 vManager 最新一代的 High Available 模式, 可实现跨地域的多团队合作与数据共享, 并部署了多引擎验证工具的包括 Xcelium、JasperGold 和 Palladium 的验证管理, 实现多维度的验证数据汇总。

目前该方案已经部署到真实的研究环境中, 为 vManager 在国内比较领先的应用, 为业内提供“跨地域

合作+多个仿真引擎”的大规模自动化验证方案提供非常有价值的参考。

1 Cadence vManager 工具简介

vManager 为业内比较先进的“服务器+客户端”模式的团队合作验证回归管理工具,可有效地实现验证计划制定、验证回归管理、验证结果分析等自动化流程。vManager 提供了友好的用户界面 GUI,其中的组件 Planner center 用于创建验证计划即 vPlan; Regression center 用于执行回归提交; Analysis center 用于分析回归状态和覆盖率情况,并生成用户自定义的报告; Tracking center 则用于为提取历次回归的数据,分析项目进度趋势,产生直方图、曲线图等直观报表,并具有自动发送邮件等功能。

这样一个验证管理工具会产生大量的验证状态数据,那必然有与企业级的其他管理工具的对接诉求,vManager 所带的 API(vAPI)提供了这样的接口,是工业标准级的 API,易于和其他工具对接,vAPI 支持用户可编程,采用基 REST 协议,使用 JSON(JavaScript Object Notation)格式承载数据,具有数据轻量级、可读性强、编程容易的特点。

2 验证管理自动化闭环诉求

我司使用 Microsoft TFS 作为项目需求管理工具,Microsoft TFS 是微软公司提供源代码管理、数据收集报告和跟踪的项目管理软件。TFS 与 vManager 有紧密的数据交互诉求,例如 Microsoft TFS 里的产品需求如何无缝地转换为 vPlan 里的验证条目,而 TFS 里的验证需求点如果产生变更,如何自动地更新 vPlan 里的条目进行迭代,特别地,需要支持 vPlan 验证条目被删除、取消、挂起和新建四种状态。

另外,我司有自研的数据看板 dashboard 系统,里面需要反馈设计需求、验证需求、验证用例通过率、代码覆盖率和功能覆盖率等状态数据,看板的数据需要来源于运行完成特定的回归,按照验证需求点的百分比换算为“待验证”(0%)、“验证中”(1%~99%)、“验证通过”(100%)。

图 1 反映了这 3 种系统的数据流:TFS 系统产生需求,更新到看板系统里的研制规范,再更新至 vManager 里的 vPlan 的验证条目。验证人员开发验证平台后,创建

testcase,启动验证并进行迭代回归,回归的数据反标至 vPlan 得到需求是否被验证通过的状态,最后反馈到 TFS 系统中,再同步至看板系统中。

3 解决方案

自动化验证过程管理方案整体思路如图 2 所示。

TFS 系统生成的需求由脚本整理成 vManager 所支持的 CSV 文件格式的验证需求框架,送给 vManager,vManager 通过 vAPI 创建或更新 vPlan,这样就达到了 TFS 向 vPlan 同步的目的。

当验证人员用 vManager 完成验证用例回归时,用例通过情况和覆盖率数据已经生成,此时在通过后台调用 vAPI 读取相应的回归,关联相关的 vPlan,获得验证需求点被覆盖的情况,此外数据客观、透明、验证人员无感知。

该方案的细节实现可划分为七部分:相关约定、TFS 自动生成 vPlan 框架、验证需求分解、验证需求和验证结果关联、TFS 状态翻转实现、验证看板内容实现、验证结果日常提交操作。

3.1 相关约定

为了实现 TFS、vManager、dashboard(验证看板)等系统之间的同步通信,特对验证的一些输入输出文件和目录命名及存放路径进行统一约定,包括 vPlan 和 vsif(包含 session 名称定义)两种文件命名约定。vsif、session、vPlan 的统一命名为<project>.<subsystem>.<module>。

另外,要对 session 自定义属性进行约定。根据项目要求周期性将调试好的用例填写到相应子系统/模块的 vsif 文件中,并在 vManager 中 launch 该 vsif 文件实现批量执行仿真作业(每 launch 一次将生成一个 session),session 结果(即仿真结果)将统一纳入 vManager 系统管理,IT 后台将通过 vAPI 接口从 vManager 系统读取仿真结果供 TFS 和验证看板使用。前面 vsif 文件命名的约定可以使自动生成 session 名包含项目、子系统、模块等信息,但还缺少 dut 类型、大版本、小版本(svn 版本)三类信息,故在 vManager 系统中相应增加(由系统管理员执行增加)属性 dut_type、dut_release_version 和 dut_svn_version。launch vsif 文件时约定配置 session 的三个属性。

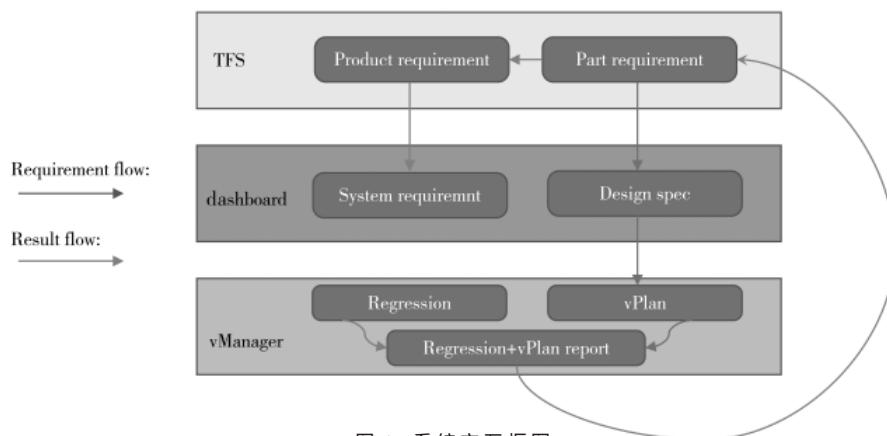


图 1 系统交互框图

3.2 TFS 自动生成 vPlan 框架

项目各个系统、子系统和模块所有 vPlan 框架由 IT 系统自动从 TFS 系统读取相关信息通过脚本生成。目前,IT 系统能自动生成 rtl 和 pr 网表验证两种 vPlan 框架,基本策略见图 3。

TFS 的需求和 vPlan 之间实现单向自动同步:IT 脚本检测到 TFS 需求变更后,通过 vManager 的 vAPI 对相应的 vPlan 内容进行迭代更新,特别地,遇到设计需求被删除、已取消、已挂起和新建四种状态,vPlan 对应的节点及其子节点的属性“Don't show in Analysis”全部更新为“true”。

3.3 验证需求分解

前面已经生成了 vPlan 框架,其实相当于大的验证需求点已经创建完毕,接下来要验证人员对这些需

求点进行分析以及分解,同时填写验证规程。首先要从 wiki 系统上以 pdf 格式导出要验证系统、子系统或模块的研制规范作为 spec,接着在 vManager 中打开对应的 vPlan,在 vPlan 中导入 spec。最后,进行需求分解和规程编写。

特别地,将所有用例(包括规划但尚未设计的)统一描述在“Input Scenarios”节点下,一条用例建一个 testcase 类型的子节点。

3.4 验证需求和验证结果关联

搭建好验证环境后,将仿真执行脚本、用例名、每条用例执行次数等信息填到 vsif 文件中,然后 launch vsif,会生成 session,vPlan 导入 session 后即可以映射环境中的 SVA Checker,Cover Group,Test List 到 vPlan 的需求底下,见图 4。

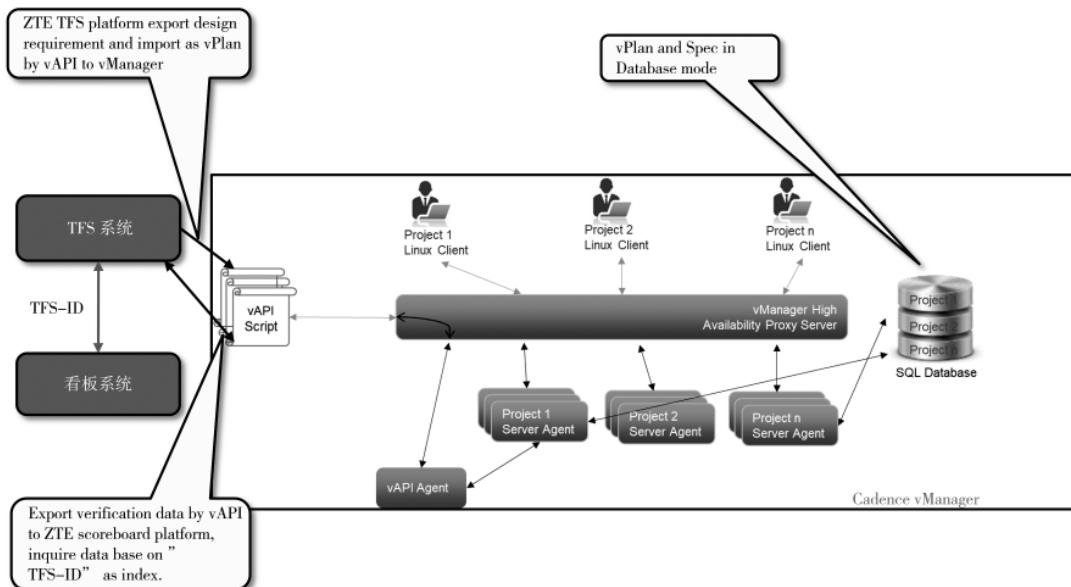


图 2 验证自动化解决方案整体框图

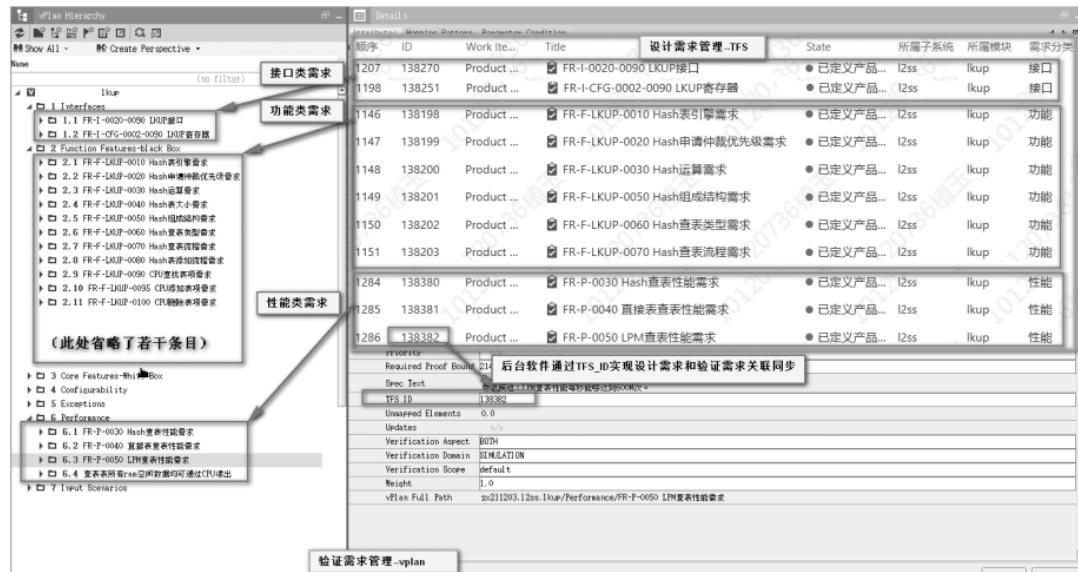


图 3 自动生成 vPlan 框架示意图

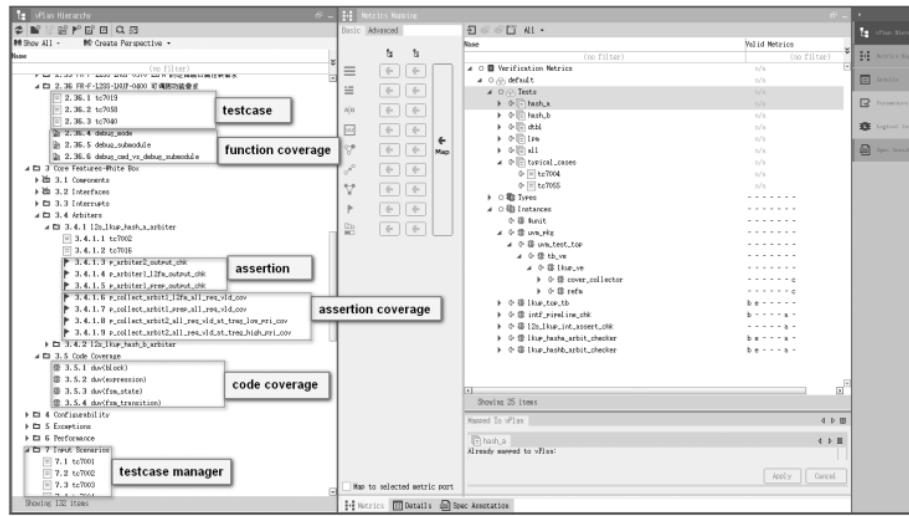


图 4 验证需求和验证结果关联映射图

3.5 TFS 状态翻转实现

本环节由 IT 后台软件自动完成。软件将项目各个系统、子系统和模块所有 vPlan 内所有来自 TFS 的原始验证需求完成百分比提取,然后根据验证需求的“TFS ID”找到其对应的设计需求,将原始验证需求完成百分比换算为“待验证”(0%)、“验证中”(1%~99%)、“已验证”(100%)填到设计需求对应的“RTL 验证状态”栏。

3.6 验证看板内容实现

验证看板有设计需求、验证需求、验证用例、bug、代码覆盖率、功能覆盖率共 6 方面状态展示。

所有设计需求都在 TFS 系统中,TFS 系统中每条需求的验证状态已经实现自动从 vManager 提取数据分析实现翻转。因此,设计需求的状态展示直接从 TFS 系统汇总项目所有需求及其验证状态,根据需求标签进行按子系统/模块分别展示。

验证需求状态展示。软件将项目所要展示的系统/子系统/模块 vPlan 内所有验证需求完成百分比提取,然后验证需求完成百分比换算为“待验证”(0%)、“验证中”(1%~99%)、“验证通过”(100%),另外,若该 vPlan 节点及其子底下所有节点关联的任意一条 testcase 任意一次 run 为 fail 的则判断该节点为“验证不通过”,依此 4 种验证状态分类统计,数据存放到看板数据库。

验证用例状态展示。软件将项目所要展示的系统/子系统/模块 vPlan 内所有用例的执行状态提取,数据存放到看板数据库。所有 EC 都在 TFS 系统中,由 TFS 系统导出所要展示的系统/子系统/模块的 EC,进行状态分类和等级分类统计即可。

验证 bug 状态展示。所有 bug 都在 TFS 系统中,由 TFS 系统导出所要展示的系统/子系统/模块的 EC,进行状态分类和等级分类统计即可。

3.7 验证结果日常提交操作

要想将验证过程结果自动呈现到 TFS 和验证看板

等 IT 系统上,务必通过前面约定的方式执行验证任务,仿真作业需通过 launch vsif 文件方式提交 session。因此,建议如下:

- (1) 日常调试平台和用例按传统验证方式执行即可;
- (2) 周期性地将调试好的平台和用例(调好的用例写进 vsif 文件)通过 launch vsif 文件方式提交 session 实现批量执行,IT 后台将自动收集跑出来的结果反馈到 TFS 和验证看板等 IT 系统上,从而实现自动化验证过程管理。

4 结论

该方案涉及的 vAPI 编程已在 Python3.7 下调试通过并稳定运行,已经部署在实际开发环境中。从规模上,目前项目部署数量为 15+,vManager Server 数量为 10+。从性能上,vPlan 的同步与更新基本达到实时,从 vManager 提取数据到看板上,根据项目的规模一次提取时间从半小时到一个小时不等,基本满足大规模 IC 开发的验证自动化需求。

参考文献

- [1] PIZZALI A. Functional verification coverage measurement and analysis [M]. Kluwer Academic Publishers, 2004.
- [2] VIJAYARAGHAVAN S, RAMANATHAN M. SystemVerilog assertions 应用指南 [M]. 陈俊杰, 等, 译. 北京: 清华大学出版社, 2006.
- [3] BERGERON B J. SystemVerilog 测试平台编写指南 [M]. 张春, 麦宋平, 赵益新, 等, 译. 北京: 科学出版社, 2009.

(收稿日期: 2021-06-23)

作者简介:

植玉(1981-),男,硕士,主要研究方向:无线通信基带芯片系统设计、通用处理器芯片设计。

梁其锋(1985-),男,硕士,主要研究方向:无线通信基带芯片系统设计、通用处理器芯片设计。

石义军(1975-),男,硕士,主要研究方向:无线通信芯片系统设计、通用处理器芯片架构设计。



扫码下载电子文档