

标题：一种并行验证加速架构及应用
--从Chiplet到Simlet

作者：张栗榕 王锋 郑文刚 王灌锋

问题提出

随着SoC芯片规模越来越大，特别是当前多Die Chiplet互联大芯片的技术发展方兴未艾，传统的EDA simulation仿真时间成指数级上升，case调试和覆盖率收敛时间已经不能满足项目TTM（Time To Maket）需求。芯片验证迫切需要更加高效的验证策略和工具。

本文提出一种并行验证计算加速架构设计思路。该思路基于UVM组件进行扩展，底层通过DPI-C接口基于原生的Linux socket 组件进行通信，设计了一套消息面、数据和交易的3平面通信同步机制，利用服务器分布式并行计算资源来换取仿真速度。

该验证思路可以应用于芯粒互联验证、芯片组网，Emulation和EDA simulation分布式联合仿真等多种形态的验证场景中。

解决方案说明

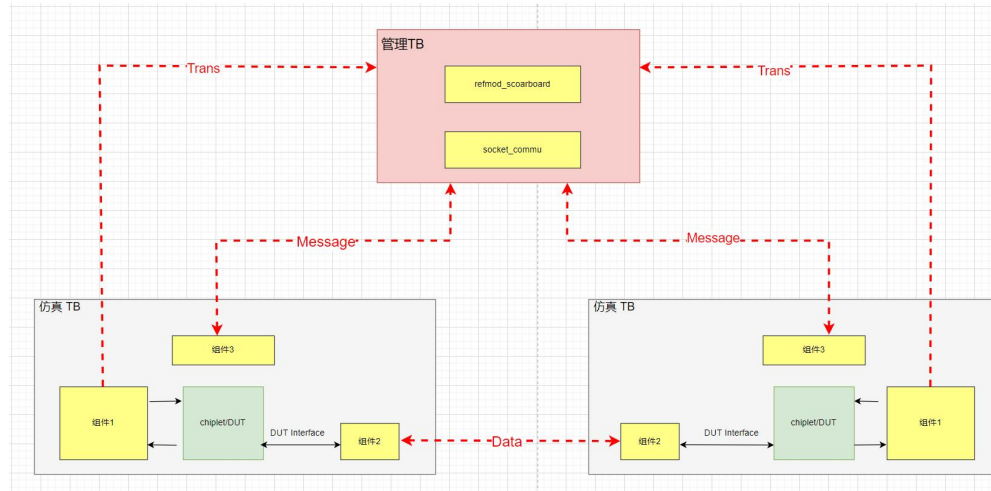


图1 芯片组网或chiplet互联并行验证加速架构示意图

图1是以芯片组网或chiplet互联为例说明该并行验证计算加速架构设计细节。该系统的TB分成两种：管理TB（图1中的上方红色框图）和仿真TB（图1中两个白色框图）。管理TB实现消息和事件分发，参考模型和scoreboard功能。仿真TB是实际的DUT仿真功能。

该系统还有3类通信（图1中红色虚线箭头），即消息平面（图1中Message箭头），数据平面（图1中 Data箭头）和交易平面（图1中Trans箭头）。消息面通信维护管理TB和仿真TB之间的消息和事件通信分发功能，一方面管理TB处理仿真TB之间的消息和事件；另一方面进行不同仿真TB之间的消息和事件分发。数据平面通信实现仿真TB中DUT之间的业务数据流交互。交易平面将仿真TB的激励和输出发给管理TB进行比对。

图1中的绿色矩形是DUT，黄色框都是并行验证计算组件，基于UVM component进行开发。组件1实现将激励和实际输出分别发送给管理TB。DUT上方黄色框组件3是维护和管理TB的消息面通信。组件

2实现和其他仿真 TB间的业务数据流交互。

由于该系统包含1个管理TB和多个仿真TB，组网灵活，需要有一个专门的文件来描述TB之间的组网信息。该系统初始化时会解析该文件，建立起并行验证计算系统，信息包含：仿真TB个数；管理TB和仿真TB id，不能重复；仿真TB之间的socket连接信息：连接方向即谁是server，谁是client，socket私有端口号。

数据平面包含2 种业务数据流传输类型，一种是基于信号的模式，另一种是基于交易的模式；两者的区别在于，信号模式以时钟cycle为周期采集DUT接口数据打包发送或者接收；交易模式是指传输的是交易数据，不含有时序信息。

并行验证系统工作流程：

- (1) 启动流程：先启动管理TB，再启动仿真TB；管理TB解析拓扑组网文件，建立消息面socket server；
- (2) 通信建立流程：管理TB根据拓扑信息引导仿真TB建立2个通信平面；
- (3) 初始化同步流程：管理TB控制仿真TB各自进行DUT初始化并通过事件同步激励发送的仿真时刻；
- (4) 仿真流程：业务数据流交互，交易通信交互，进行checker比对；
- (5) 退出流程：管理TB在checker比对完毕后通过拓扑文件控制仿真TB关闭socket通信并退出仿真；

实践结论

以实际项目的芯片组网验证为实测基础，相比于传统将多个DUT包成Wrapper进行验证的仿真，仿真效率提升了~54%（该加速比和DUT的合理划分有关）。

优势总结：

提高EDA simulation仿真验证效率；适用于芯粒互联，芯片组网，甚至是大规模DUT partition后的并行验证加速；本质是通过使用EDA simulation的服务器资源获取仿真时间，有效地减少了验证成本；并行验证计算组件已经通用化设计，可以在现有TB基础上方便集成形成并行验证加速架构；

该架构的应用限制：

- (1) 对于对业务数据流时间敏感的场景，该系统可能无法精确仿真，需要考虑验证的局限性；
- (2) 由于socket传输时间的不确定性，在仿真TB的组件中一般会通过队列方式缓存接收和发送的业务数据流，对于性能case的仿真需要深入分析是否合适；