

标题：ASV多场景测试激励自动生成系统

作者：张栗榕、王锋、张烨晨、贺岩

问题提出

大规模数据通信SoC芯片一般都包含了大量的网络接口模块，体现出Port数量多（数十个 ports量级）和速率种类多（1GE~400GE，包括各种组合可达数十个）。在实际应用中，当芯片和对端设备对接时，每个port都可以配置成不同的速率模式，有非常丰富的接口配置组合模式，从而形成了大量的接口验证场景。这给芯片验证带来了非常大的挑战。一般的随机化验证只是对数据、参数配置进行随机，无法覆盖如此丰富的接口模式和场景组合，对RTL代码交付质量造成较大的风险。

本文提出了一种自研的多场景测试激励自动生成系统（Automatic Scenarios Verification，ASV），采用场景随机化思想来灵活构造系统级验证平台和测试用例，可在仿真时动态调整，通过大量随机种子（seed）仿真来逼近场景验证的完全覆盖。

解决方案说明

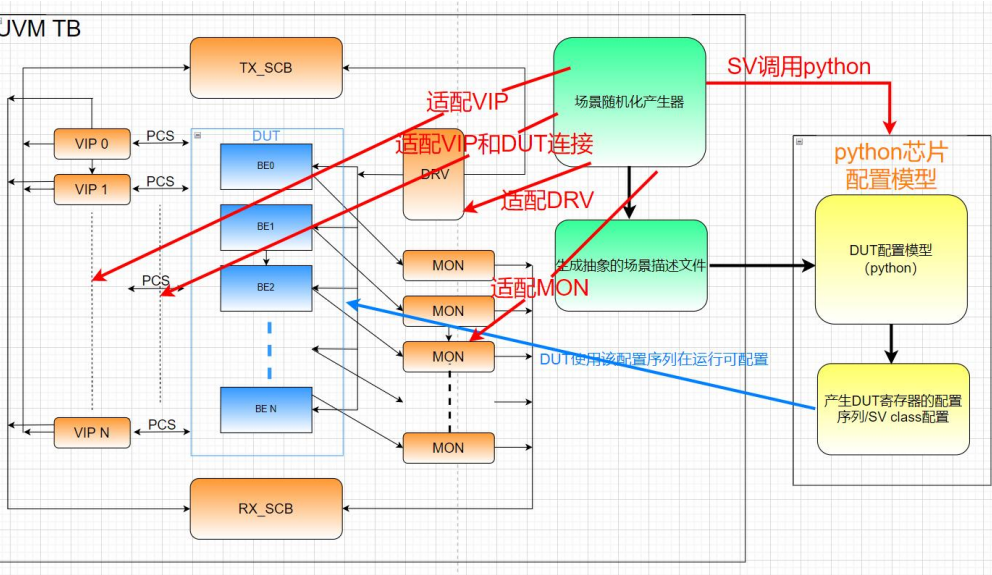


图1 ASV验证架构示意图

- ASV 基本组件包括场景随机化产生器，芯片配置模型，TB组件配置更新器，整体结构如图1所示；
- （1）场景随机化产生器基于芯片规格随机化生成特定的接口配置组合场景；
 - （2）芯片配置模型基于python开发，主要实现解析特定的接口配置组合场景生成DUT配置序列；
 - （3）TB组件配置更新器实现解析特定的接口配置组合场景调整TB相关组件满足特定场景的验证需求；

关键字	用途描述
MOD	描述DUT模块名；
ATTR	表示一个属性描述项的关键字，例如：PORT_LIST_NUM,表示该属性描述的是Port个数；PORT_SDS_LAN表示port占用的Serdes lane个数；PORT_LIST,表示port占用的sds位置；
TYPE	该属性对应值的类型，例如：DEC,代表VALUE字段是十进制类型；STR_LIST,代表VALUE字段是字符串列表类型；
VALUE	该属性具体的值，根据TYPE字段可以是数字也可以是字符或者是其他类型；

表1 场景描述文件的关键字

场景随机化产生器在UVM case的build phase阶段根据传入的随机seed基于芯片规格随机生成特定的接口配置组合场景，同时生成抽象的场景描述文件，描述文件的关键字如表1所示，例子如图2所示。

```
|MOD=BE0
ATTR=PORT_LIST_NUM;TYPE=DEC;VALUE=2
ATTR=PORT0_SDS_LANE;TYPE=DEC;VALUE=4
ATTR=PORT0_LIST;TYPE=STR_LIST;VALUE=[0]
ATTR=PORT1_SDS_LANE;TYPE=DEC;VALUE=2
ATTR=PORT1_LIST;TYPE=STR_LIST;VALUE=[4,6]
```

图2 场景描述文件例子

芯片配置模型在case的build phase阶段通过SV调用python模型解折抽象的场景描述文件生成DUT的配置序列；配置序列定义了一套寄存器访问格式，如图3所示。TB支持在仿真时解析配置序列，不需要提前编译到build中。

```
DUT配置序列模板：
wait 100
indwrite module_name block_id offset_address reg_value
write module_name reg_name reg_value
```

图3 生成的寄存器配置序列例子

TB组件配置更新器解析场景随机化产生器生成的特定场景信息，修改VIP和DUT之间的数据连线，生成特定的时钟，Driver和Monitor组件接收场景随机化产生器生成的场景，更新自己的配置；例如：使能了哪几个ETH Port，速率，Serdes占位，报文个数，报文长度，流控帧类型等。

总结

该验证方法已经在实际的数通芯片项目中进行了部署，统计数据显示对大规模数通芯片的某子系统场景组合验证从预估的~5人月降低到~1人月，并发现了多个RTL场景组合bug。

该方法的优点：

- （1）该方法和验证架构可以很方便地移植到其他类似需要大量随机场景的验证系统中；
- （2）低门槛进行场景随机化验证；

针对场景配置数量非常巨大的验证，后续可以考虑对随机化场景产生器做优化，即通过管理每个seed随机化生成的场景来避免生成重复的配置场景进而提升随机seed覆盖效率，从而加速随机化case regression覆盖收敛的速度。