

基于 UVM 验证方法学的纵向可重用研究

熊 涛, 蒋见花

(中国科学院 微电子研究所, 北京 100029)

摘 要: 现在片上系统(SOC)的复杂度和集成度越来越高,这给验证带来了巨大的挑战.传统的验证方法存在各种不足,在效率方面已经远远达不到生产的要求.UVM 是近年来兴起的一种高效的通用验证方法学,不仅可以缩短验证周期,而且具有很好的可重用性.UVM 验证方法学的可重用主要分为横向的可重用和纵向的可重用,主要阐述了 UVM 中纵向可重用的方法,并以 APB 总线为例,描述了从模块级到系统级的验证平台的搭建方法,这种方法很好地体现了纵向重用提高验证效率的优势.

关键词: UVM;纵向可重用;APB 总线

中图分类号: TN47

文献标识码: A

文章编号: 1000—7180(2016)04—0064—05

Research of Vertical Reuse Based on UVM

XIONG Tao, JIANG Jian-hua

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

Abstract: As the complexity and size of SOC(System on chip) grow, verification of design faces huge challenge. There are many defects in the traditional verification so that it can't meet the demand of manufacture in the aspect of efficiency. UVM is a universal verification methodology which springs up in recent years and has high efficiency. It can not only shorten the period of verification, but also can be reused conveniently. The reuse of UVM mainly includes Horizontal reuse and Vertical reuse. This paper focuses on the Vertical reuse and takes the APB bus as example, to expatiate how to build a verification platform from module level to system level. The methodology obviously shows that the Vertical reuse has the advantage of improving verification efficiency.

Key words: UVM; vertical reuse; APB bus

1 引言

随着超大规模集成电路设计复杂度和集成度的提高,集成电路的验证也面临着巨大的挑战.在芯片的研发过程中,在验证上所花费的时间占到了整体时间的 70%^[1],芯片验证已经成为芯片研发中最关键的环节.传统的验证语言存在抽象建模能力不足,扩展和维护很难,缺少带约束的随机激励,仿真速度慢等缺点^[2],已不能满足现在验证的需要.UVM 是 Accellera 公司于 2011 年推出的基于 systemverilog 语言的通用验证方法学,其主体来源为 OVM,同时又采纳了 synopsys 在 VMM 中的寄存器解决方案 RAL^[3].正因为 UVM 克服了 OVM 和 VMM 各自

的缺点,所以 UVM 已经成为现在最主流的验证方法学,代表了验证方法学的发展方向.

在 UVM 验证方法学中,主要包括三方面的可重用:横向(x 轴)可重用、纵向(y 轴)可重用以及对角(z 轴)可重用.这三种可重用合称为 3D 可重用^[4].横向可重用主要指的是类似 IP 或 SOC 之间的可重用,横向可重用的基础是这些 IP 或者 SOC 具有相同的总线接口.纵向可重用主要指的是从模块级到子系统级或者系统级的重用,纵向可重用主要体现在同一个项目在不同的时期对早期资源的重复利用.对角可重用主要指的是不同抽象级别(C, RTL, FPGA 等)之间的重用.本文主要研究的是纵向可重用的验证方法,并在纵向可重用的基础上,以

验证 APB 总线为例,搭建了一个高效的子系统验证平台。

2 UVM 验证平台

简单来说,UVM 就是建立在 systemverilog 平台上的一个库,它提供了很多接口,让验证人员更加方便地进行验证. 在这个库中有一个最基本的父类 uvm_object,实际验证中用到的各种类都是由这个父类派生而来,主要包括 TLM 建模、组件描述、进程控制三个种类,这样的派生方式,有利于验证平台的维护与扩展. UVM 验证方法学采用树形的结构来管理验证平台的各个部分,其中各个验证组件(UVC)都是树的结点,每一个组件的构建以及组件间的连接都是其父结点进行相应操作,这样可以实现各个部分的分工协作,而不至于出现遗漏和错误. UVM 的执行顺序是按照 phase 来进行的,phase 主要分为两大类:一类是 function phase,不消耗仿真时间,只消耗运行时间;一类是 task phase,消耗仿真时间. UVM 的 phase 机制规定了某些代码必须写在特定的 phase 之中,相当于强制固定了代码的执行顺序,这在很大程度上解决了代码顺序杂乱可能引发的问题. 除此之外,UVM 还有一些不同于其他验证方法学的特点和优势,使得 UVM 成为现在最前沿的验证方法学.

图 1 所示为一个典型的 UVM 验证平台. 在验证平台中每一个验证组件都履行着自己的职责. sequencer 通过启动一个 sequence,从 sequence 中获取抽象化的数据,并把这些数据转交给 driver. driver 将抽象化的数据流转化成信号级的数据发送给 DUT. monitor 在接口处采样 DUT 的输入输出信号,收集覆盖率信息并做进一步的检查. agent 主要用于封装 sequencer,driver 和 monitor,可以分为 active 模式和 passive 模式. active 模式一般用在输入驱动,passive 模式中只包含 monitor,一般用在输出监测. reference model 一般在有寄存器或存储器的 DUT 中使用,通过前门或者后门访问的方式,可以对被测设计的寄存器和存储器进行实时的读或者写操作. reference model 是整个 UVM 验证平台的关键和核心,因为它的功能和 DUT 一致,只是描述的语言更为高级. scoreboard 最后将会根据参考模型的输出与 DUT 的输出进行比较,得出比较结果. env 是例化以上验证组件的一个大容器,可以选择性地对各组件进行例化,这对于验证组件的可重用有很大的意义.

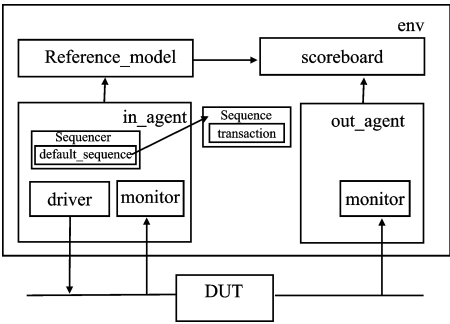


图 1 典型的 UVM 验证平台框图

3 UVM 的纵向重用

现在大型项目的验证都依托于纵向验证. 一个大的芯片在开发时,是分成多个小的模块来开发的,每个模块开发一套独立的验证环境. 在验证的前期首先进行模块级的验证,后期做系统级的验证时重复利用前期的验证资源,再经过适当的修改和添加,便可以搭建一个完整的系统级验证平台,这样大大提高了验证的效率. 一般用到的纵向重用技术主要有三点:基于 env 的重用、基于 sequence 的重用,以及基于 register model 的重用.

3.1 基于 env 的重用

假设一块芯片有三个模块 A,B,C,如图 2 所示,这三个模块在进行模块级验证时,都分别有着如图 1 所示的验证平台,而且都有自己的 driver 和 sequencer. 在后期进行系统级的验证时,B 的激励来自于 A 的输出,C 的激励来自于 B 的输出,如果采用 env 级别的重用,那么 B,C 验证平台中的 driver 和 sequencer 都可以取消,同时这一级 input_agent 中的 monitor 和上一级的 ouput_agent 中的 monitor 完成的功能是一致的,所以 B 和 C 的 input_agent 可以完全取消. 最后整个系统级的验证平台如图 3 所示.

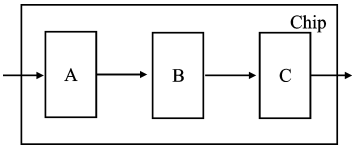


图 2 具有三个模块的简单芯片

3.2 基于 sequence 的重用

一般在进行 UVM 验证的时候,都会构建一个 virtual sequencer, virtual_sequencer 用来实现 sequence 之间的同步,在进行模块级验证的时候,每个模块都有一个属于自己的 virtual sequencer,但是

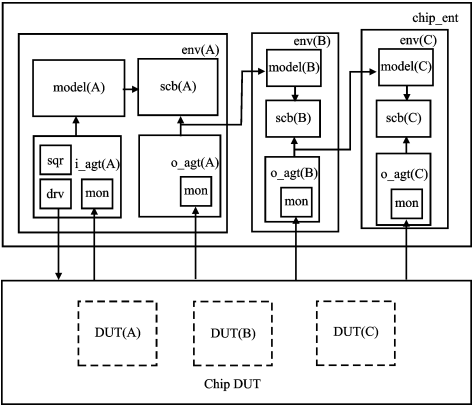


图 3 基于环境重用的验证平台

在进行系统级的验证的时候,一般都会取消掉模块级的 virtual sequencer,而在在系统级的环境中重新例化一个 virtual sequencer,并在这个新的 virtual sequencer 中例化每个模块的 sequencer.

和 Virtual_sequencer 对应的 sequence 是 virtual_sequence,它不发送 transaction,只是控制其他的 sequence,起统一调度的作用.在模块级验证的时候会有很多验证模块功能的 sequence,这些 sequence 在各自的 virtual sequence 中进行排列组合.同样,在进行系统级的验证的时候,一般都会取消掉模块级的 virtual sequence,而是构建一个系统级的 virtual sequence.这个系统级的 virtual sequence 用来调度和控制各个模块的 sequence,也可也将模块级的 virtual sequence 直接运用到系统级的 virtual sequence 中.

在进行从模块级到系统级的 sequence 重用,只会用到边界模块的 sequence.这些 sequence 可以直接进行重用,也可以通过修改已有的约束条件或是添加新的控制属性,以实现新的测试序列的生成. sequence 的纵向重用,如图 4 所示.

3.3 基于 register model 的重用

一般来说,每个模块都会有自己的寄存器配置总线,在集成到芯片时,芯片有自己的配置总线,这些配置总线经过仲裁之后分别连接到各个模块.每个模块的寄存器都有各自的偏移地址,如 ABC 模块的偏移地址都是'h0,在进行模块级验证时,需要给每个模块构建一个寄存器模型,并且指定各自的偏移地址,这里都为'h0.在系统级使用寄存器模型,需要建立一个新的寄存器模型,这个新的寄存器模型中需要加入各个模块的寄存器模型并按照总线的地址设置各个模块的基地址(如 A 模块的基地址为'h0000,B 模块的基地址为

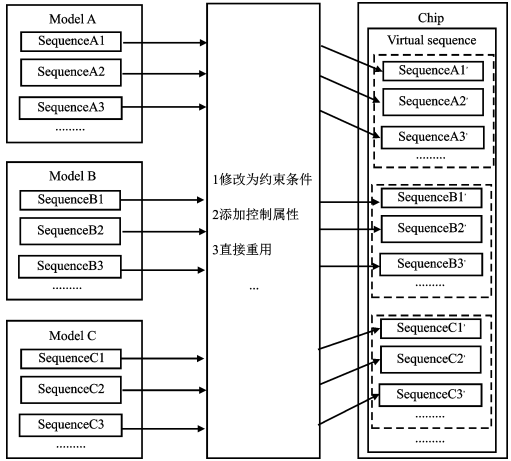


图 4 sequence 的纵向重用

'h1000,C 模块的基地址为'h2000)以及后门访问路径.因为模块的 default_map 并不知道寄存器的基地址,只知道寄存器的偏移地址,其实际的物理地址还要再加上一个基地址.所以只有将其加入芯片级的 default_map,并在加入的同时告知模块 map 的偏移地址,这样系统级的 default_map 才能完成前门访问操作.

基于可重用的寄存器模型,一般都是在芯片级的环境中对寄存器模型进行例化,然后把参考模型的指针传递到 virtual sequencer 中的参考模型,进而传递到寄存器配置的 sequence 中,这样便能完成寄存器的读写操作.若是各个模块中的参考模型需要对寄存器进行读操作,则还需要将寄存器的指针传递给各个模块的环境中.

4 基于纵向可重用的 APB 总线验证

本节将以 APB 总线验证为例,来说明基于 UVM 纵向重用的方法.虽然 APB 总线只是一个子系统,但是在验证策略上一样用到了纵向重用.图 5 所示的 DUT 是待验证的 APB 总线.总线上有一个 AHB2APB 桥,AHB2APB 桥是 AHB 高速总线上的从设备,也是 APB 外围总线上唯一的主设备,它提供了高速总线 AHB 到低速总线 APB 的接口,主要功能是完成 AHB 传输协议向 APB 传输协议的转换.在 DUT 中,APB 总线的从设备包括 2 个 UART 模块,一个 SPI 模块和一个 GPIO 模块.在进行子系统的验证之前,首先要分别对 AHB2APB,UART,SPI,GPIO 这几个模块进行验证.只有这些模块功能正确了,才能进行子系统级的验证.

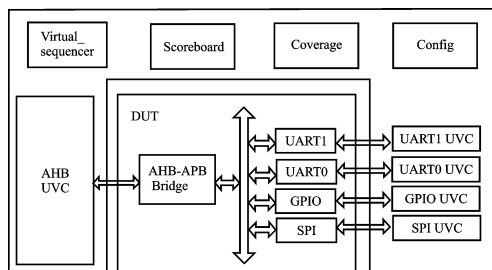


图 5 APB 总线验证平台

APB 总线的验证构建了一个 AHB UVC, 一个 SPI UVC, GPIO UVC 和两个 UART UVC, 如图 5 所示。写操作时, AHB UVC 作为一个 AHB master, 向 AHB2APB 桥发送地址、数据和控制信号, AHB2APB 桥根据地址信息选择从设备, 并将数据发送给从设备, APB 从设备再将数据发送给对应的外围 UVC, 外围 UVC 进行数据的采集并将数据传递给记分板, 记分板将其和 AHB UVC 监测到的数据进行比较。读操作时, AHB UVC 向 AHB2APB 发送的写操作改为读操作, 读取地址信号上对应的外围 UVC 发送过来的数据。同样在记分板中会对这两种数据进行比较。

模块级的验证中参考的是如图 6 所示的验证平台标准目录结构^[5], tb 中包括设计和验证整体的平台架构, 验证平台的顶层环境以及测试场景. uvc 中包含验证组件以及寄存器模型等, sequence 中包含的是测试序列. 这样一来, 模块级的测试环境很容易作为 UVC 被顶层测试平台进行调用. 在图 6 中同样也说明了 reg 和 sequence 的重用.

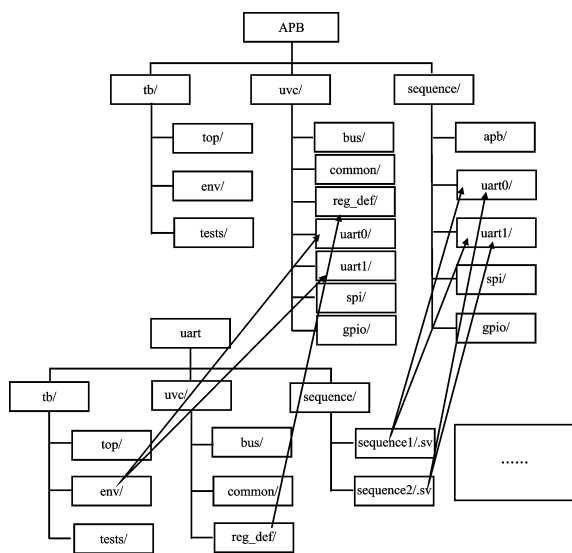


图 6 验证平台目录结构及重用

在 APB 总线的顶层模块中给每个 APB 模块分

配了地址,地址范围见表 1.

表 1 APB 从模块的地址范围

APB 从模块	地址范围
UART0	‘h4002_0000 ~ ’h4002_3FFF
UART1	‘h4002_4000 ~ ’h4002_7FFF
SPI	‘h4001_0000 ~ ’h4001_3FFF
GPIO	‘h4010_0000 ~ ’h4010_3FFF

在 UVM 寄存器模型中,每个顶层的 `uvm_reg_block` 中都需要定义一个基地址(`create_map`)和一个子 `uvm_reg_block` 的偏移地址(`add_submap`),这个偏移地址对于子 `reg_block` 而言就是其基地址.在模块级验证的时候,各个模块中的寄存器起始地址是基地址‘h0,所以在 `reg_block` 只需要把基地址设为‘h0,然后添加各个 `uvm_reg` 的偏移地址(`add_reg`).在进行子系统级验证的时候,需要建立一个新的寄存器模型,在这个新的寄存器模型中把基地址设为‘h0,然后将各个模块的偏移地址设为其实际物理地址的起始地址:

```
default_map=create_map("default_map",0,
4,UVM_LITTLE_ENDIAN);
```

```
default_map.add_submap(uart0_rm, default_map, 'UVM_REG_ADDR_WIDTH', 0x0002_0000);
```

```
default_map.add_submap(uart1_rm, default_
map, 'UVM REG ADDR WIDTH', h4002 4000);
```

```
default_map.add_submap(spi_rm.default_map, 'UVM_REG_ADDR_WIDTH', h4001_0000);
```

```
default_map.add_submap(gpio_rm, default_map, 'UVM_REG_ADDR_WIDTH', 4010_0000);
```

顶层验证中的 sequence 重用了模块级的 sequence, 通过将模块级的 sequence 增加约束以及重新排列组合形成了新的 sequence. 表 2 中显示了具体的重用方案与描述. 其中 AHB UVC 对 APB SLAVE 的读写 sequence 都是由模块级的 sequence 增加地址约束和数据约束而生成的. 最后在顶层的 virtual_sequence 中的 fork join 过程块中加入这三个顶层序列, 使各个 UVC 对设计的操作都能同时进行.

APB 总线的验证平台重用了 UART, SPI 和 GPIO 三个模块的环境, 寄存器模型以及测试序列。所以验证的过程中只需要将重点放在各个 UVC 的整合以及记分板, 系统级环境的搭建上, 这大大减少了平台的编写和调试时间, 有利于验证人员对各个

功能点进行充分的测试.

表 2 各个模块的 sequence 重用

顶层序列	模块级序列	描述
u2a_incr_payload	uart_ctrl_config_reg_seq	UART 的寄存器配置序列
	uart_incr_payload_seq	UART UVC 向 UART 发送数据
	intrpt_seq	AHB UVC 读取 UART 接收缓存器中的数据
	ahb_to_uart_wr	AHB UVC 向 UART UVC 发送数据
apb_spi_incr_payload	spi_cfg_reg_seq	SPI 的寄存器配置序列
	spi_incr_payload	SPI UVC 向 SPI 发送数据
	read_spi_rx_reg	AHB UVC 读取 SPI 接收缓存中的数据
	ahb_to_spi_wr	AHB UVC 向 SPI UVC 发送数据
	spi_en_tx_reg_seq	SPI 数据发送使能
apb_gpio_simple_vseq	gpio_cfg_reg_seq	GPIO 的配置寄存器序列
	gpio_simple_trans_seq	GPIO UVC 向 GPIO 发送数据
	read_gpio_rx_reg	AHB UVC 读取 GPIO 接收缓存器中的数据
	ahb_to_gpio_wr	AHB UVC 向 GPIO UVC 发送数据

5 结束语

UVM 验证方法学统一了各大验证方法学,不仅功能强大,而且架构层次清晰,在可重用、自动化、可扩展、规范化等方面具有很大的优势. 本文阐述了 UVM 验证方法学中的纵向可重用方法,并以 APB 总线的验证为例搭建了一个基于模块重用的验证平台. 这种纵向的重用能够很好的利用前期的资源,大大减少了验证的时间和精力,显著提高了验证效率,在实际应用中具有很高的利用价值.

参考文献:

[1] SalahK. A UVM-based smart functional verification platform;Concept, pros, cons, and opportunities[C]// Design &. Test Symposium (IDT), 9th International. Dead Sea; IEEE,2014;94-99.

[2] 李博. 基于 system verilog-vmm 的仿真环境设计及其应用[D]. 哈尔滨:哈尔滨工业大学,2009;3.

[3] 张强. UVM 实战[M]. 北京:机械工业出版社. 2014;4.

[4] Drechsler R, Chevallaz C, Fummi F, et al, Panel: Future SoC verification methodology: UVM evolution or revolution[C]// Design, Automation and Test in Europe Conference and Exhibition (DATE). Dresden; DATE,2014;1-5.

[5] Yun Young-Nam. Beyond UVM for practical SoC verification[C]//SoC Design Conference (ISOC), 2011. Jeju: ISOC,2011;158-162.

作者简介:

熊 涛 男,(1989-),硕士研究生,研究方向为数字集成电路物理设计与验证. E-mail:xiongtao@ime. ac. cn.

蒋见花 女,(1976-),副研究员. 硕士生导师,研究方向为数字集成电路低功耗设计.

(上接第 63 页)

[8] 姜占才,孙燕,姚刚. 模糊聚类与 LBG 级联的 VQ 算法[J]. 计算机工程与科学,2011, 33(5) : 155-159.

[9] 毛旭. 说话人检索系统的研究与实现[D]. 大连: 大连理工大学,2009.

作者简介:

黄 婷 女,(1990-),硕士研究生. 研究方向为说话人识别.

E-mail:tilitha@163. com.

周 萍 女,(1961-),硕士,教授. 研究方向为语音识别、智能控制.

景新幸 男,(1990-),博士,教授. 研究方向为信号与信息处理.

杨 青 女,(1976-),硕士,副教授. 研究方向为智能处理与信息处理.