文章编号:2096-1618(2020)05-0514-05

基于UVM的APB接口与DMA联动的验证方法研究

何林峰, 陈 娇, 聂 海
(成都信息工程大学通信息工程学院,四川 成都 610225)

摘要:在芯片设计的流程中至关重要的一环就是对于芯片的验证。APB接口作为一种AMBA总线接口协议,被大面积应用于各类SoC芯片中。当使用APB接口进行数据搬移时,需要DMA模块与CPU进行信号交互,由此带来APB接口与DMA模块联合验证的需求,而不同的IP核对于功能覆盖点有不一致的要求。针对不同的IP核,提出一种针对APB接口与DMA联动的验证结构,可通用于各种IP核的APB接口与DMA联动验证,增强了验证模块的可重用性。通过UVM体系的验证,通用功能覆盖率达到100%,符合业内的实际生产需求覆盖率。

关 键 词:数字集成电路;SoC;UVM;APB;DMA;功能覆盖率
中图分类号:TN402 **文献标志码**:A
doi:10.16836/j.cnki.jcuit.2020.05.006

0 引言

集成电路的高速发展与半导体工艺技术的更新换代,带来的是芯片设计规模的加大、复杂程度的提高和设计周期的缩短。为提高生产效率,利用IP(intellectual property)设计的Soc(system on chip)技术应用而生,采用第三方提供的IP核提高设计效率已成主流趋势。对于大规模的SoC芯片而言,仅仅依靠EDA(electronic design automation)工具的验证已经不能够保证芯片设计的完备性[1-2]。芯片前端验证(verification)在RTL(register transfer lever)代码综合为版图前起到了验证功能点、验证工作时序等至关重要的芯片性能需求,芯片流片的失败绝大部分是由于验证的不合理、不充分。而前端验证花费的时间成本与资源成本占到了整个SoC芯片设计的一半以上,因此对于验证平台的验证效率提出了极高的要求。为了保证芯片功能的完备性,需要验证人员对不同的SoC芯片分别提出功能验证完备、执行效率高的验证方案,由此带来对验证环境高复用率等复杂的验证问题。

1 简介

APB(advanced peripheral bus)作为一种通用的AMBA总线,常用于系统高速总线与低速接口之间的转换桥,被大面积应用于各类芯片中[3]。当APB用于数据的迁移时,需要DMA(dynamic memory allocation)模块进行配合,减小对于CPU的负载。而在整个芯片的验证中,APB接口与DMA模块通常为不同工程师进行验证,对于不同的IP核、APB接口与DMA模块的联动有着不同的功能需求,导致不同的验证都需要进行一次时序、功能的协商。

DMA(direct memory access)传输方式是将一组有效数据从一个地址空间搬移到另一个地址空间[4-5]。与CPU实现搬移数据不同,DMA传输通过CPU对搬移数据这一动作初始化,传输动作本身由DMA模块实现。在实现DMA传输前,CPU对总线的控制权向DMA转移[6],在DMA传输完成后,CPU重新获得总线的控制权,减小了CPU的负载,提高了数据搬移的效率。

UVM(universal verification methodology)作为目前业内通用的验证方法学,继承了OVM(open verification manual)的特性[7-8],使用SystemVerilog语言,提供多种通信接口,是一种高效、随机充分的验证体系。由于其具有高效的随机约束方式、高复用性、适用各类需求的特点受到各大主流IC公司的追捧[9-11]。

2 传统APB接口与DMA联动的验证方式

APB接口通常由Pclk、Preset_n、Psel、Penable、Pwrite、Paddr、Pwdata、Prdata信号构成[12],其接口信号描述如表1所示[13-14]。

APB接口的写数据时序逻辑图如图1所示。APB接口实现写操作时,在第一个时钟上升沿处,Psel和Pwrite信号拉高,Paddr和Pwdata驱动至端口;在第二个时钟上升沿处,Penable信号拉高,确立数据的有效性;在第三个时钟上升沿处,Psel、Penable、Pwrite拉低,完成本次写操作。

收稿日期:2019-11-04
基金项目:四川省科技计划资助项目(2014FZ0050)

表 1　APB 接口信号描述

端口名称	方向	功能描述
Pclk	input	APB 总线时钟,内部工作时钟
Preset_n	input	APB 总线复位,低电平有效
Psel	input	APB 总线选择有效信号
Penable	input	APB 总线使能有效信号
Pwrite	input	APB 总线读写指示信号
Paddr	input	APB 总线地址,指示 APB 访问地址
Pwdata	input	APB 总线写入数据
Prdata	output	APB 总线读出数据

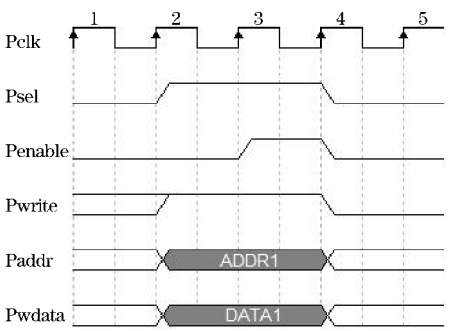

图 1　APB 接口写数据时序图

APB 接口的读时序逻辑图如图 2 所示。APB 接口实现读操作时,在第一个时钟上升沿处,Psel 信号拉高,Paddr 驱动至端口;在第二个时钟上升沿处,Penable 信号拉高,确立数据的有效性,同时 Prdata 驱动数据至端口;在第三个时钟上升沿处,Psel、Penable 拉低,完成本次读操作。

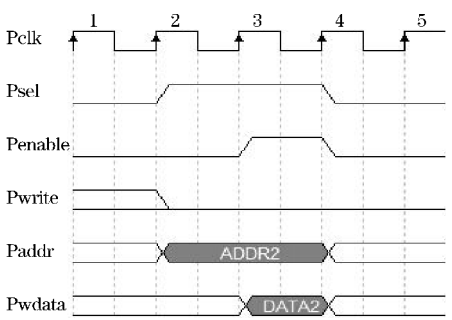

图 2　APB 接口读数据时序图

DMA 模块与 APB 接口联动时存在 6 个端口,其 DMA 信号接口描述如表 2 所示。

表 2　DMA 接口信号描述

端口名称	方向	功能描述
tx_single	input	写操作空闲
tx_burst	input	写操作多次空闲
rx_single	input	读操作空闲
rx_burst	input	读操作多次空闲
tx_ack	output	写操作握手回应信号
rx_ack	output	读操作握手回应信号

当 DMA 模块收到来自 APB 接口的 tx_single、rx_single 时,相应的反馈一个持续一拍的 tx_ack 或者 rx_ack[15],当 APB 接口检测到 ack 信号时,对应的 tx_single 或 rx_single 会拉低一拍,随后恢复正常信号;收到 tx_burst、rx_burst 则对应返回多次间断的 ack 信号,回应次数根据空满水线的配置情况可自定义。由于 single 与 burst 信号的区别仅为反馈 ack 信号的次数,故以 single 信号为示例解释 DMA 模块与 APB 接口发送交互时的数据发送和接收。

APB 接口与 DMA 模块联动时,通常时序逻辑如图 3 所示。在第二个时钟上升沿时,拉起 tx_single 信号,DMA 模块在第三个时钟上升沿处拉起一个持续一拍的 tx_ack 信号,在第四个时钟上升沿处 APB 接口检测到 tx_ack 的上升沿,将 tx_single 信号拉低一拍,并对应执行一次写数据操作,即写地址为 ADDR1,写数据为 WT1。后续进行一次写地址为 ADDR2、写数据为 WT2 的写操作和一次读地址为 ADDR3、读数据为 RD1 的读操作。

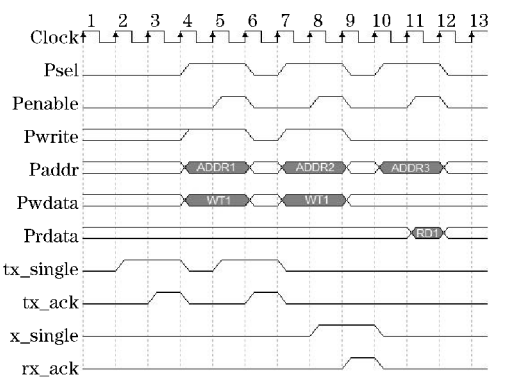

图 3　APB 接口与 DMA 联动时序图

如图 4 所示,为传统 APB 接口和 DMA 联动验证的框架。通常情况下,验证 APB 接口和 DMA 联动的时序和功能正确性时,将 APB 接口与 DMA 模块的验证分为两个不同 APB_Agent 与 DMA_Agent,并且两个 Agent 均为 Master。在顶层 harness 内,通过 interface 进行例化实现数据的交互。在 APB_Agent 内,存在 APB_Driver、

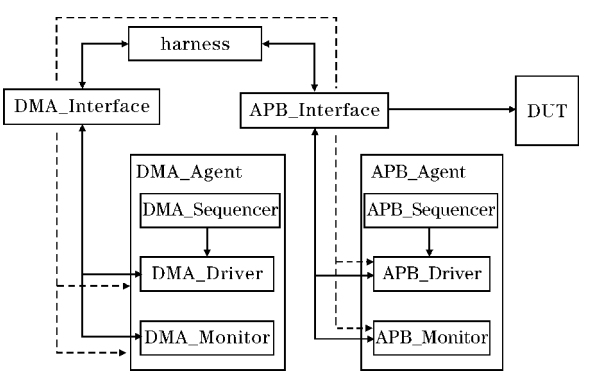

图 4　APB 接口与 DMA 联动验证框架

APB_Sequencer 和 APB_Monitor，APB_Driver 将驱动 APB 接口，APB_Sequencer 将产生不同的随机 Sequence 送入 APB_Driver 并通过 APB_Interface 驱动至 DUT 接口上，APB_Monitor 将根据配置信息采集 APB_Interface 的数据并分别送往参考模型和计分板进行数据的转换和数据的比对。DMA_Agent 内部接口与 APB_Agent 相一致。

如图 4 中虚线流程所示，每一次的 tx_single/rx_single/tx_burst/rx_burst 拉高，均需要通过顶层 harness 的端口进行交互，同时每一次的操作都需要经过 Agent、Interface、harness 几个层次，对于验证中的参数传递和数据流都带来了更高的复杂度。并且不同的 IP 对于 DMA 联动时反馈的 ACK 信号可能有不同的延迟需求，传统框架由于两个 Agent 均为 Master_Agent 不利于验证平台的重复利用，大大降低了平台的可移植性。

3 对传统联动验证平台的改进分析

针对传统 APB 接口与 DMA 联动的验证框架存在的可移植性差、跨层次复杂等问题，对验证框架与验证流程进行研究与改进。

如图 5 所示，为改进后的 APB 接口与 DMA 联动的验证框架。图 5 中 APB_Driver 为 Master_Agent，而 DMA_Driver 为 Slave_Agent。APB_Agent 中存在 APB_Driver、APB_Sequencer 和 APB_Monitor 组件，而 DMA_Agent 内仅有 DMA_Driver 组件。如图 5 中虚线流程所示，每一次的数据交互仅通过两个 Agent 就可以实现，同时跨层次也仅在 Agent 层次通信，降低了数据流和参数传递的复杂性。

DMA_Driver 为 slave_driver，仅实现根据 tx_single/rx_single/tx_burst/rx_burst 的电位高低应激反馈对应的 ack 信号，并在内部定义有数据包间隔(PINV)信号，通过传参的方式根据不同的 DMA 需求实现不同的延迟 ACK 信号。DMA_Driver 的工作流程如图 6 所示。

对于 APB_Driver 为 master_driver，其内部存在 UVM 仲裁机制实现的数据先后序列，即当正常数据发送时，获得 APB_Driver 的控制权，在完成本次正常数据操作后，释放 APB_Driver 的控制权，同时应激反馈的 ACK 信号获得 APB_Driver 的控制权，进行 DMA 联动的 DMA_req 数据操作，完成 DMA 联动的数据操作后，APB_Driver 静默，等待下一次的事务到来获得触发。同时，在 APB_Driver 内，同步例化有 APB_Interface 和 DMA_Interface(statc)，用以省去 APB_Monitor 采集数据再发送给 DMA_Monitor 的步骤，并且 DMA_Interface 为 statc 属性，即静态数据属性，APB_Driver 仅对其有读取属性。

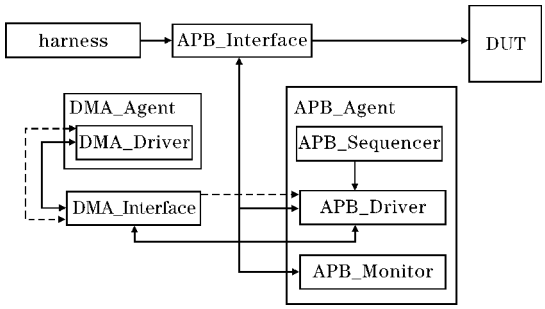

图 5 改进的 APB 接口与 DMA 联动验证框架

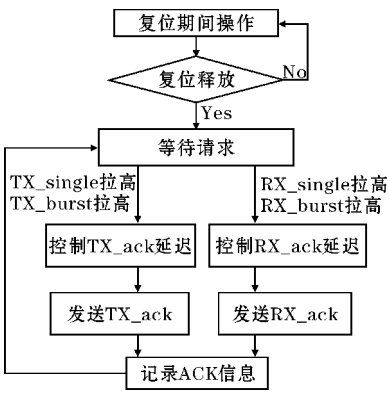

图 6 DMA_Driver 工作流程

在 APB_Sequencer 中，事务数据包定义为 APB_Trans，其 Trans 列表如表 3 所示。表中 pinv 为数据包间隔；apb_start_time 和 apb_end_time 为事务开始时间和事务结束时间，用于送往计分板和参考模型进行时序的检查；apb_result 为事务执行结果，为 0 时表示事务执行失败，为 1 时表示事务执行成功，用于参考模型确认事务数据的有效性；apb_hdl 为 APB_Trans 的句柄，用于 Agent 内信号互联。

表 3 APB_Trans 列表

变量	类型	约束
apb_enum	rand enum	READ、WRITE
paddr	rand [11:0]	NA
pdata	rand [31:0]	NA
pinv	rand int	[32'h000:32'hfff]
apb_start_time	realtime	NA
apb_end_time	realtime	NA
apb_result	bit	NA
apb_hdl	integer	NA

如图 7 所示，为 APB_Driver 的工作流程图。APB_Driver 先处于复位状态期间，等待复位释放后接收来自 APB_Sequencer 的 APB_Trans。同时检测是否有 APB_Trans 或者 DMA_ACK 响应，如果有则将本次 trans 发送完成后优先执行 APB_Trans，若无 APB_Trans 有 DMA_ACK 响应，在 APB_Driver 内组合构成一个读数据操作或者写数据操作优先完成 DMA_ACK 响应的操作，等待将响应操作完成后，再进入 Trans 传输驱动；如果没有则等待 Trans 的传输驱动。如果没

有对地址位读写的操作时,APB_Driver 不应驱动 APB_Interface,其信号应该保持直到下次的 trans 进行有效的驱动。

APB_Monitor 将根据配置信息采集 APB_Interface 的数据分别送往参考模型和计分板进行数据的转换和数据的比对。而由于 DMA_Interface 将在 APB_Driver 中静态例化,不需要单独产生 DMA_Trans 数据包送往参考模型和计分板,故而 DMA_Agent 不需要设计 DMA_Monitor 对 DMA 的信号进行采集,直接由 APB_Agent 实现相关功能。

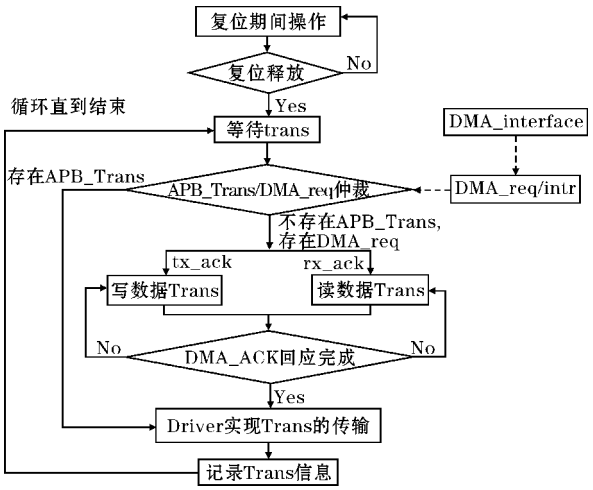

图 7　改进的 APB_Driver 工作流程图

4　改进后联动验证平台的仿真

对 APB_Monitor 到参考模型和计分板的端口进行定义,端口定义如表 4 所示。apb_apb2rm 为 APB_Monitor 采集到发送给 DUT 的数据,并打包为 APB_Trans 发送给参考模型;apb_apb2sb 为 APB_Monitor 采集到的 DUT 发出的数据,并打包为 APB_Trans 发送给计分板。

表 4　APB_Monitor 事务列表

端口	类型	目的端
apb_apb2rm	APB_Trans	Reference Model
apb_apb2sb	APB_Trans	ScoreBoard

由于仅对 APB 接口与 DMA 联动的数据流与时序进行验证,故无数据端到端的验证。采用 UVM 方法学搭建的整体仿真验证平台框架如图 8 所示。

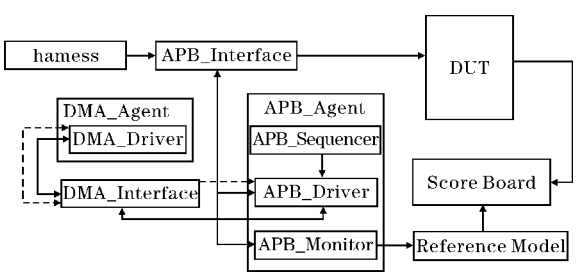

图 8　验证平台整体框架

在 TestCase 内,将 APB_Trans 内的 paddr 随机约束为[12'h000:12'hFFF],将 apb_enum 随机化,将 apb_data 随机化。通过 TestCase 调用 APB_Sequencer 发送 APB_Trans 随机数据包到 APB_Driver,通过 APB_Driver 驱动至 APB_Interface 送往 DUT 内。同时 DMA_Driver 内存在对于 ACK 回馈的延时随机数,将其设置为三位 rand bit 类型,以实现 0 ~ 7 个时钟单位的随机延迟 ACK 回应,验证改进后的框架适用于不同需求的 DMA 延迟 ACK 回馈。通过仿真,得到如图 9 所示的随机波形,符合验证所需要的激励。

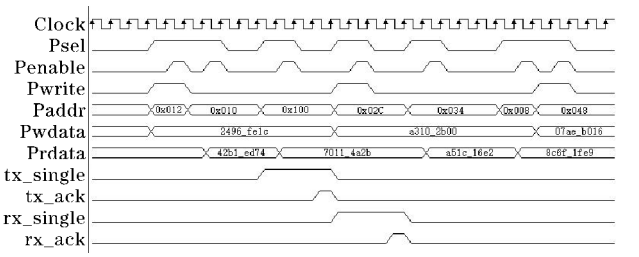

图 9　随机仿真波形

对实现的 TestCase 进行回归检测,回归次数为 200 次,得到如图 10 所示的代码覆盖率报告。

其中 LINE 为行覆盖率,表征代码无冗余项;COND 为条件覆盖率,表征了代码的每种 if-else 可能性或者 case 的可能性是否遍历;TOGGLE 为翻转覆盖率,表征了每个信号是否经过信号的翻转过程;而 FSM 为状态机的覆盖,表示状态机是否遍历了所有跳转情况;以及 BRANCH 为分支覆盖,表示了所有的代码分支是否运行。通过图 10 可以看出,对于 APB 接口与 DMA 联动的 RTL 代码覆盖整体达到 100% 覆盖率,符合验证平台整体要求。

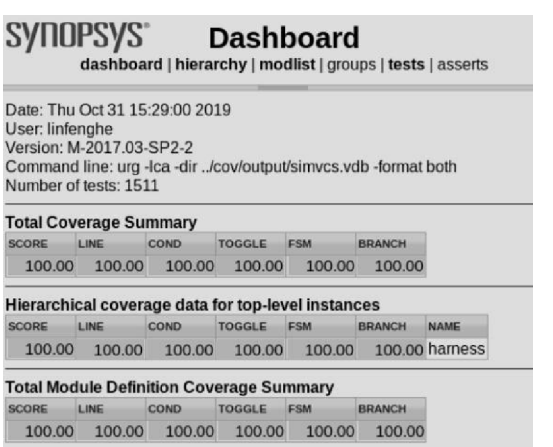

图 10　代码覆盖率报告

5　结束语

验证作为芯片设计中非常重要的一个环节,保证了芯片流片的成功率。在对传统 APB 接口与 DMA 联

动验证平台的分析后,对验证框架做出结构上的优化,减少验证平台的接口,并针对优化后的结构重新定义可适用于不同IP验证需求的APB_Trans,提高了代码的可移植性。同时优化验证平台内参数的传递与数据流的跨平台层次,降低验证平台的复杂性,并保证优化后的平台代码覆盖率为100%。

参考文献:

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

[2] 潘应进,龙恳. 基于UVM实现高效可重用的SoC功能验证[J]. 电子世界,2016(2):180-183.

[3] 牛玉坤,孟令琴. 基于UVM实现APB-I2C模块的功能验证[J]. 工业控制计算机,2017(7):77-79.

[4] Jayaraj U Kidav, NM Sivamangai, MP Pillai, et al. Architecture and FPGA prototype of cycle stealing DMA array signal processor for ultrasound sector imaging systems[J]. Microprocessores and Micrsosystems,2019(4):53-72.

[5] Wei Liu, Daan Pareit, Eli De Poorter, et al. Advence spectrum sensing with parallel processing based on software-defined radio[J]. EURASIP Journal on Wireless Communications and Networking,2014(11):21-35.

[6] V Soni, A Hadjadj, O Roussek et al. Parallel multicore and multi-processor methods on point-value multiresolution algorithms for hyperbolic conservation laws[J]. Journal of Parallel and Distributed Computing,2018(2):14-19.

[7] Xu HUANG, Xin He, Zheng rong He, et al. Using UVM Testbench to Generate the Analog Stimuli[J]. Proceeding of 2019 2nd Internatonal Conference on Informativs Control and Automation,2019(5):249-253.

[8] 李晨阳,宋澍申,王涛,等. 一种基于UVM的高层次化验证平台设计[J]. 微电子学与计算机,2019(7):79-83.

[9] 徐波. 基于UVM高速SERDES的数字系统验证[J]. 电子科学技术,2017(9):32-36.

[10] 吴升光,羊箭锋,冯春阳. 基于UVM的浮点功能部件验证[J]. 微电子学与计算机,2017(4):121-125.

[11] Rath, A. W., Ecker. A transaction-oriented UVM-based library for verification of analog behavior[J]. Journal on IEEE,2014(3):216-220.

[12] 符宏利,田茜,吴金. 基于APB总线的SPI控制器设计[J]. 电子世界,2012(3):113-115.

[13] He Zhen, Cao Yang, Zhang Jun-xin. Implementation of transacation lever AMBA bus models using systemC[J]. Wuhan University Journal of Natural Sciences,2004(10):98-106.

[14] Shabir Parach. Field programmable gate array (FPGA) implementation of novel complex PN-code-generator based data scrambler and descrambler[J]. Maejo International Journal of Science and Technology,2010(4):125-127.

[15] Michelangelo Grosso, Wilson Javier Perez Holguin, Danilo Ravotto, et al. Functional Verification of DMA Controllers[J]. Journal of Electronic Testing,2011,27(4):12-15.

Research on Verification Method of APB Interface and DMA Linkage based on UVM

HE Linfeng, CHEN Jiao, NIE Hai

(College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: A critical part of the chip design process is verification of the chip. As an AMBA bus interface protocol, the APB interface is widely used in various SoC chips. When using the APB interface for data movement, the DMA module needs to perform signal interaction with the CPU, which brings the requirement for joint verification of the APB interface and the DMA module, and different IP cores have inconsistent requirements for function coverage points. This paper proposes a verification structure for the linkage between the APB interface and the DMA for different IP cores. It can be used for the linkage verification of APB interface and DMA of various IP cores, which enhances the reusability of the verification module. Through the verification of the UVM system, the universal functional coverage reached an interpretable 100%, which in line with the actual production demand coverage within the industry.

Keywords: digital Integrated circuit; SoC; UVM; APB; DMA; functional coverage