

基于 UVM 的 CAN 模块自验证方法

熊 涛, 蒋见花

(中国科学院 微电子研究所, 北京 100029)

摘 要: 简单介绍了目前主流的验证方法学 UVM, 并使用该方法学对 CAN 模块之间的通信进行了功能验证. 和一般的 UVM 验证方法不同, 在本次验证中并没有花费精力构建 CAN 的验证模型, 而是采用了被测模块验证被测模块的自验证方法, 来模拟真实的 CAN 总线传输场景. 验证结果覆盖了预期的功能点, 并有效地提高了验证的效率.

关键词: UVM; CAN; 自验证; 验证效率

中图分类号: TN47

文献标识码: A

文章编号: 1000—7180(2016)09—0093—05

Self-verification of CAN Module Based on UVM

XIONG Tao, JIANG Jian-hua

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

Abstract: The author simply introduced the Universal Verification Methodology (UVM) which is the most popular and then verified the communication of CAN module based on UVM. Different from traditional UVM, the verification didn't build the can verification module which would take much time but verified the can module with itself to simulate the real communication scene. The result covered the expected function points, and showed high efficiency in verification.

Key words: UVM; CAN; self-verification; verification efficiency

1 引言

目前市面上主要有三种基于 SystemVerilog 的验证方法学^[1]: VMM (Verification Methodology Manual), OVM (Open Verification Methodology), UVM (Universal Verification Methodology). VMM 是 Synopsys 在 2006 年推出的, 它继承了寄存器解决方法 RAL. OVM 是 Cadence 和 Mentor 在 2008 年推出的, 引进了 factory 机制, 功能很强大, 但是没有寄存器解决方案. UVM 正式版在 2011 年由 acellera 推出, 得到了 Synopsys, Mentor 和 Cadence 的支持. UVM 起源于 OVM, 同时又采纳了 VMM 的寄存器解决方案 RAL, 其继承了 VMM 和 OVM 的优点, 克服了各自的缺点, 代表了验证方法学的发展方向.

CAN (Controller Area Network), 即控制器局域网, 是一种十分成熟的串行通讯协议, 能有效地支持具有很高安全等级的分布实时控制. 随着 CAN 总线应用范围的扩大和半导体集成电路的发展, SOC 和 MCU 中集成 CAN 模块的设计也越来越多, 如何确保 CAN 模块在设计中功能的正确性是一个十分重要的课题. 本文利用 UVM 的验证方法学对自行设计的 CAN 模块进行了验证, 验证结果覆盖了所有的功能点, 表明该验证平台具有较高的效率以及很好的可重用性.

2 UVM 验证方法学

UVM 的基础语言是 SystemVerilog 语言, SystemVerilog 的最大特点就是采用类的思想来进行验证. 类有函数, 另外还可以有任务, 通过函数和

任务可以完成各个验证组件的功能,同时类中还有成员变量,这些变量可以控制类的行为.最后,类还有强大的继承与派生功能,有助于采用事务级的验证和提高验证的重用性.对 SystemVerilog 本身而言,其支持约束随机产生,支持覆盖率统计分析,支持断言验证,所以,采用 SystemVerilog 语言的验证方法学和传统的验证方法相比具有很明显的优势.

在 UVM 验证方法学中,这样一些组件构成了 UVM 验证平台^[1]: Driver, Monitor, Agent, Env, Test, Sequence, Sequencer, Model. 这些组件都是由 UVM 预先定义好的类派生而来,然后以一个倒置的树的形式连接在一起,每个组件都是树上的一个节点,如图 1 所示. 这样的形式,方便 UVM 执行后面的操作. UVM 还有一个重要的特点就是引入了 phase 机制. Phase 机制的引入在很大程度上解决了代码顺序杂乱可能会引发的问题. 其本质是通过把代码顺序强制固定来实现这个目的.

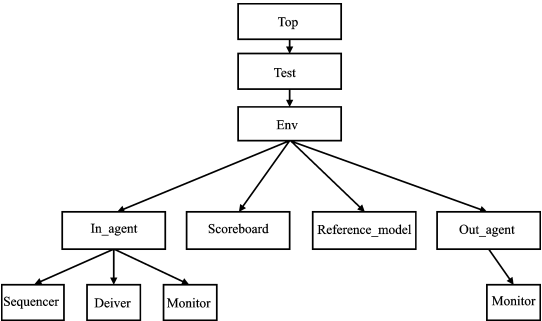


图 1 UVM 的树状结构

图 2 所示为一个典型的 UVM 验证平台. 在这个平台中, Sequencer 通过启动一个 sequence,从 sequence 中获取事务级的数据,并把这些数据转交给 Driver. Driver 将事务级的数据流转化成物理级的数据发送给被测单元. Monitor 在接口处采样被测模块的输入输出信号. Agent 的作用是把 Sequencer, Driver 和 Monitor 封装在一起,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 的功能和 Dut 一致,只是描述的语言更为高级,编写起来更为方便. Scoreboard 最后将会根据参考模型的输出与 Dut 的输出进行比较,或者直接将 Dut 的输入和输出进行对比,得出比较结果并打印. 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.

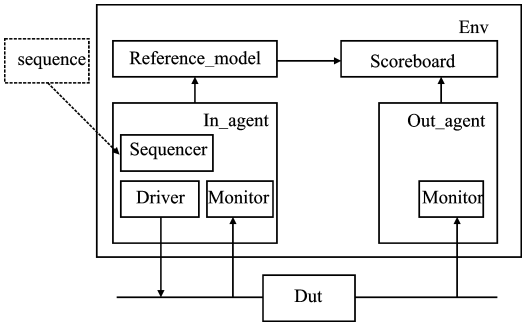


图 2 典型的 UVM 验证平台

3 CAN 模块的结构以及特点

本次验证中使用到的 CAN 模块由项目小组自行研发,其内部结构如图 3 所示.

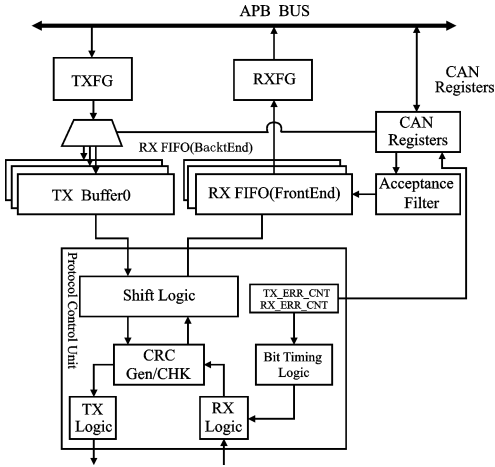


图 3 CAN 总线模块的内部结构

CAN IP 模块对外的信号接口为 RX 和 TX,用于接收和发送串行数据,对内则通过 APB 总线进行寄存器方式的访问. 内部为寄存器控制、发送通路、接收通路以及协议控制单元四个主要部分. 寄存器控制主要负责 CAN 模块的工作模式设定、bit rate 设定、接受过滤规则等常规参数设定;发送通路包括三个发送缓存器,用于分别发送消息队列;接收通路包含 3 级 FIFO,用于对接收的数据进行缓存处理,并进行帧过滤;协议控制单元(PCU)主要用于符合 CAN 总线协议规定的时序逻辑产生、CRC 产生及校验等工作.

该模块实现了 CAN 协议 CAN2. 0A 和 CAN2. 0B,支持标准(11 bit)和扩展数据帧(29 bit),有 8 个可配置的接收过滤器,可以实现 2 个 32 bit、4 个 16 bit、8 个 8 bit 的过滤配置. 同时还具有 8 个接收过滤屏蔽器,分别与接收过滤屏蔽器相关联. 过滤器和过滤屏蔽器同时工作,可以达到筛选特定帧的作用.

工作模式主要分为以下几种。

(1)初始化模式:在初始化模式下,模块不进行发送和收。错误计数器清零,中断标志保持不变。编程器能够访问配置寄存器,而在其他模式下是受到限制的。

(2)正常工作模式:在此模式下,模块被激活。I/O 引脚用作 CAN 总线功能。模块将通过 TX 和 RX 引脚,发送和接收 CAN 总线报文。发送消息格式包括 13 byte 数据(4 byte ID, 8 Byte Data, 1 Byte Length)和 1 Byte 的优先控制(Priority)。模块内部为这 14 Byte 数据分配到 TXFG 寄存器地址空间,CPU 可以通过这一地址空间完成数据的发送。接收消息包括 13 个字节(4 Byte ID, 8 Byte Data, 1 Byte Length)。当接收到有效帧时,阐释 RXF 的接收中断,CPU 读取数据内容,并清零 RXF 位。

(3)监听模式:如果激活监听模式,则 CAN 总线上的模块是被动的,接收引脚仍然为输入,对于接收器,不发出错误标志或应答信号。该状态下,错误计数器失效。监听模式可用来检测 CAN 总线上的波特率。要使用监听模式,必须要有至少 2 个可以互相通信的节点。

(4)自检测模式:如果环回模式被激活,模块将在模块边界,把内部发送信号连接到内部接收信号。发送和接收引脚将转换到其 I/O 端口功能。外部实际输入信号被禁止,接收器将忽略接收应答错误。

本次验证,将主要对以上各个模式进行验证。主要包括寄存器的读写,数据的发送和接收,监听模式,自检测模式,仲裁模式,错误注入等。

4 验证平台搭建

4.1 基于 UVM 的自验证平台

一般的 CAN 模块的验证会采用如图 4 所示的方法^[4],将被测目标和总线模型同时挂在总线上,利用总线模型对总线进行驱动或者采样。总线模型完成的功能基本和被测目标一致。在 UVM 中,这个总线模型就是 CAN 的验证组件 UVC 或者知识产权验证核 VIP。

本次验证中搭建的基于 UVM 的 CAN 模块自验证平台如图 5 所示,直接将几个被测的 CAN 模块连接到同一个 CAN 总线上。一个 CAN 模块作为发送端,其余 CAN 模块作为接收端,模拟了真实的通信场景。和一般的验证平台相比,这个验证平台不用编写复杂的 CAN 验证模型,发送验证和接收验证可以同时进行, master_agent 和 slave_agent 的功

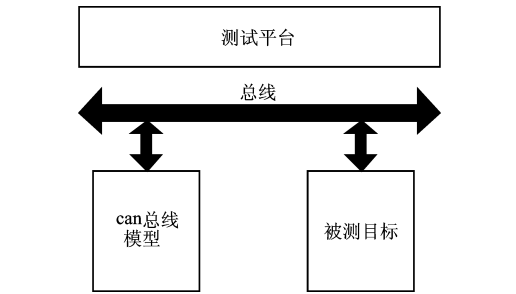


图 4 基于 CAN 总线模型的验证环境

能和结构基本一致,因为都是对 apb 总线进行操作。有了这些特点,验证平台的搭建和测试序列的运行节省了大量时间,大大提高了效率。下面将详细讲述验证平台中一些重要模块的具体作用。

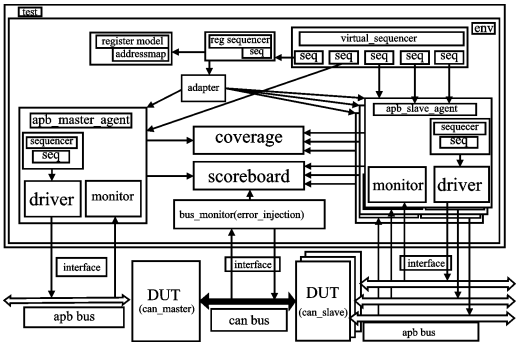


图 5 基于 CAN 总线模型的验证环境

agent:在验证平台中有两种 agent,一种是 apb _master_agent,主要用来对 can_master 的 apb 总线进行驱动和监控,另一种是 apb_slave_agent,主要用来对 can_slave 的 apb 总线进行驱动和监控。这两种 agent 是同一个类的两种例化,唯一的区别是作用的总线不同。

driver:两种 agent 下面的 driver 功能是一致的,一方面等待重置信号将端口重置,另一方面发送地址和数据。根据 apb 总线协议,发送数据时,driver 在第一个时钟上升沿来时发送地址、数据和控制信号,第二个时钟上升沿来时将 penable 信号拉高,同时地址、数据和控制信号保持不变。如果是读操作,则不用发送数据。数据周期可以用低电平的 pready 信号延长,所以在 driver 中每一节拍也必须用高电平的 pready 事件进行触发。

monitor:每个 apb_agent 下的 monitor 都可以实现两种类型的采样:第一种是 apb 发送数据时对 apb 总线进行采样;第二种是 apb 总线接收数据时对 apb 总线进行采样。由于 apb 总线是半双工的,发送和接收不会同时进行,所以一个 monitor 就可以

完成两种采样任务. bus_monitor 主要用来对 can 总线进行采样, 分析帧结构, 并在特定位置进行错误注入. 最后这三种 monitor 将采样值送入 scoreboard 进行比较以及进行覆盖率分析.

register model: 这里建立寄存器模型主要是完成对 CAN 模块的寄存器的配置. 一般寄存器模型的访问操作可以分为前门访问和后门访问, 在进行前门访问时, 必须要将实例化的寄存器加入 default_map 中, default_map 的作用是存储所有寄存器的地址, 否则无法进行前门访问操作. 由于同一个验证平台中不能出现相同的寄存器地址, 所以每个 CAN 的寄存器地址也不一样.

virtual_sequencer: virtual_sequencer 用来实现 sequence 之间的同步. 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. 在这个验证平台中, 有两种 sequence, 一种是控制寄存器模型的数据传输, 另一种是控制从 apb 到 can 的数据产生.

adapter: 在前门访问操作中, 无论读还是写, 寄存器模型都会通过 sequence 产生一个 uvm_reg_bus_op 的变量. 如果是写操作, 此变量中的信息要经过一个转换器转换后交给 sequencer, 随后交给 driver 由 driver 实现最终的前门访问读写操作. 如果是读操作, 转换器将收集来的 transaction 转换成寄存器模型能够接受的形式, 以便寄存器模型能够更新相应的寄存器的值.

整个验证的流程首先从例化各个模块开始, 逐一进行连接, 然后配置 CAN 的寄存器. can_master 主要是配置工作模式、波特率以及发送的数据, can_slave 主要配置工作模式、波特率以及过滤模式. 正常工作模式下的验证流程如图 6 所示.

正常工作模式下, 由 apb 向 can_master 发送数据, can_slave 接收数据然后由 apb 总线进行读取, 只有通过筛选的帧才会被读取. 最后在 scoreboard 中会对 apb 发送端的数据和 apb 接收端的数据以及 bus_monitor 的数据进行比较, 只有正确的数据才进入覆盖率统计.

另外, 在不同的工作模式下, 需要构造不同的 testcase 对寄存器进行配置. 在验证仲裁机制时, 需要将某些 can_slave 配置成主动模式.

4.2 错误注入

由于 can 之间的正常传输不会产生错误, 这个属于测试的边界情况, 所以需要在总线中手动注入

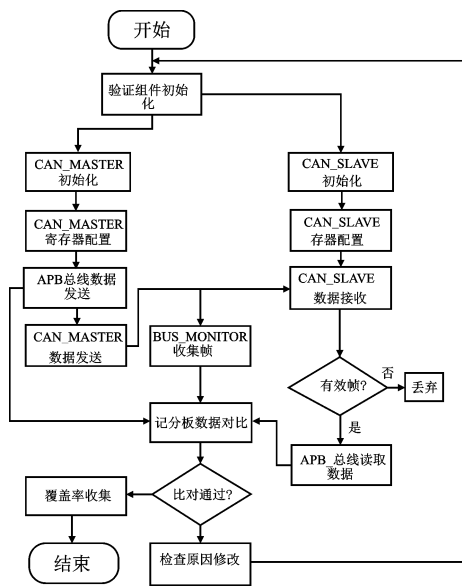


图 6 正常工作模式下的验证流程

错误, 构建直接测试用例. 在验证平台中的 bus_monitor 起到了验证 CAN 总线的帧格式以及错误注入的作用, 在检测到特定位置时, 直接利用接口给总线的下一位强制赋值完成错误注入操作. CAN 的错误类型主要有以下几种^[5].

(1) 位错误: 一个 CAN 模块在发送数据的同时也会监视总线电平, 如果发送的值与监视的值不一样, 那么就会产生一个位错误. 错误注入的方法为在总线某个采样点处将总线值强制赋为与 TX 相反的电平.

(2) 格式错误: 位场的格式都是固定的, 如果在位场中出现一个或者多个非法的位时, 就会产生格式错误. 错误注入的方法为在应答界定符处将总线的采样值强制赋为“0”.

(3) 填充错误: 在一帧中, 如果出现六个连续相同的电平, 就会产生一个填充错误. 错误注入方式为在连续的高电平后面强制将总线赋为高电平或者在连续的低电平后面将总线赋为低电平.

(4) 应答错误: 接收节点对接收到的帧不作应答, 即应答位为显性电平时产生应答错误. 错误注入方式为在帧的应答位将总线采样值强制赋为显性电平.

(5) CRC 错误: 一般接收器计算 CRC 和发送器计算的 CRC 即帧的 CRC 场是一样的, 如果两个 CRC 序列不一致, 就会产生一个 CRC 错误. 错误注入方式为将总线中的 CRC 序列强制进行改动.

5 验证结果

本次测试利用 Synopsys 的 VCS 工具进行仿真

验证,经过不断的添加测试用例和测试序列,修改约束以及随机种子,最终得到了 100%的功能覆盖率,并且记分板中的数据匹配性为 100%,没有逻辑功能错误,预期的功能验证目标全部达到. 具体的测试功能点和覆盖率统计如表 1 所示. 表中所列的是对标准帧和扩展帧共同统计的结果.

表 1 工作模式及功能覆盖率

工作模式	功能点	覆盖率/%
初始化模式	主要寄存器的读操作	100
	主要寄存器的写操作	100
正常工作模式	发送功能	100
	接收功能	100
	应答功能	100
	发送中断	100
	接收中断	100
	仲裁机制	100
	接收过滤功能	100
监听模式	监听功能	100
自检测模式	自检测功能	100
	位错误	100
	格式错误	100
	填充错误	100
	应答错误	100
	CRC 错误	100

6 结束语

UVM 和传统的验证平台相比,前期的验证平台的搭建会比较复杂,但是进入回归测试之后以及在可重用性方面,优势很明显. 本文所讨论的 CAN 模块的自验证方法,由于没有利用 CAN 总线模型,验证平台的搭建比较简单,也具有很好的可重用性. 同时,这种自验证的方法在其他总线上也有很好的参考和利用价值,能够大大的提高验证的效率.

参考文献:

[1] 张强. UVM 实战[M]. 北京:机械工业出版社,2014.
[2] 赵强,田泽,楼小强,等. CAN 总线控制器 IP 核设计实现[J]. 计算机技术与发展,2013,23(8): 230-233.
[3] BOSCH CAN. Specification V2. 0[R]. Robert Bosch GmbH,1991.
[4] 吕易俗,田文杰,李翔宇,等. 基于 Verilog 的 CAN 总线协议验证模型库的建立[J]. 微电子学与计算机, 2015,32(3) : 122-126.
[5] 崔瑾. SoC 芯片中 CAN 总线控制器的研究和验证 [D]. 西安:西安电子科技大学,2013.

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 -),硕士生导师. 研究方向为数字集成电路低功耗设计.

(上接第 92 页)

[3] Nourani M, Tehranipour M. RL-huffman encoding for test compression and power reduction in scan application[J]. ACM Trans Des Autom Electron Syst, 2005, 10(1): 91-115.
[4] 韩银和,李晓维. 测试数据压缩和测试耗协同优化技术 [J]. 计算机辅助设计图形学学报, 2005, 17 (6): 1308-1311.
[5] Chandra A, Chakrabarty K. Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes [J]. IEEE Transactions on Computer Aided Design of Inte-

grated Circuits and Systems, 2003, 52(8): 1076-1088.
[6] 商进,张礼勇. 一种双游程编码的测试压缩方案[J]. 哈尔滨理工大学学报, 2010, 15(4): 19-22.

作者简介:

谈恩民 男,(1966-),教授. 研究方向为计算机辅助测试和集成电路可测试性设计.
李 贞(通讯作者) 男,(1990-),硕士研究生. 研究方向为计算机辅助测试和集成电路可测试性设计.
E-mail: 1078300120@qq. com.