基于 UVM 的 Wishbone-SPI 验证平台设计

刘森态，庞　宇，魏　东
（重庆邮电大学 光电工程学院，重庆 400065）

摘　要： 随着芯片复杂度增加，芯片验证在设计流程中所消耗时间也不断提高。针对传统验证平台重用性差、覆盖率低，通过使用通用验证方法学(Universal Verification Methodology，UVM)设计 Wishbone-SPI 验证平台，用 UVM 组件灵活地搭建验证平台，完成标准的验证框架，设计受约束随机激励，自动统计功能覆盖率。仿真结果显示，该验证平台功能覆盖率达到 100%，并表明该平台具有良好的可配置性与可重用性。

关键词： UVM；SPI；Wishbone；验证平台
中图分类号： TN402　　**文献标识码：** A　　DOI：10.16157/j.issn.0258-7998.212337

中文引用格式： 刘森态，庞宇，魏东. 基于 UVM 的 Wishbone-SPI 验证平台设计[J].电子技术应用，2022，48(6)：36-41.
英文引用格式： Liu Sentai，Pang Yu，Wei Dong. Design of Wishbone-SPI verification platform based on UVM[J]. Application of Electronic Technique，2022，48(6)：36-41.

Design of Wishbone-SPI verification platform based on UVM

Liu Sentai，Pang Yu，Wei Dong
(Chongqing University of Posts and Telecommunications，Chongqing 400065，China)

Abstract： As the complexity of the chip increases，the time it takes to verify the chip in the design process continues to increase. Aiming at the poor reusability and low coverage of traditional verification platforms，this paper uses Universal Verification Methodology (UVM) to design the Wishbone-SPI verification platform，uses UVM components to flexibly build the verification platform，completes the standard verification framework，and designs constrained random stimulus，automatically counting function coverage. The simulation results show that the functional coverage of the verification platform reaches 100%，and shows that the platform has good configurability and reusability.

Key words： UVM；SPI；Wishbone；verification platform

0 引言

基于 TestBench 的传统芯片验证方法，对人员的验证经验要求高，且平台可移植性差，难达到预期覆盖率，不适于快速迭代芯片设计市场[1]。目前，UVM 已是 IC 验证领域主流的方式，它由 SystemVerilog 语言编写，具有面向对象(Object Oriented Programming，OOP)的特点，拥有丰富的组件和基类，完善的通信机制，继承并克服开源验证方法(Open Verifcation Methodology，OVM)没有寄存器的解决方案，可以根据验证需求，快速地搭建工程[2-3]。

串行外设接口 SPI(Serial Peripheral Interface)是由 Motorola 公司定义一种单主机多从机全双工的模式通信，协议标准下由主入从出数据线(Master Input Slave Output，MISO)、主出从输入数据线(Master Output Slave Input，MOSI)、串行时钟信号线(Serial Clock，SCK)和从机选择信号线(Slave Select，SS)共 4 种引脚组成。该接口可与模数转换器 DAC、数模转换器 DAC 和外部存储器等芯片通信，是 SoC(System on Chip)最主要的外设接口之一。

Wishbone 总线是由 Silicore 提出的，现由 OpenCore 组织维护管理，是 SoC 三大总线标准之一[4]。Wishbone 总线是一种 IP 核(Intellectual Property Core)互联体系结构，它定义 IP 核之间公共接口，减轻系统组件集成难度，提高系统组件可重用性、可靠性和可移植性，兼容所有综合工具，可用多种硬件描述语言实现[5]。Wishbone 总线为 IP 核提供点到点、数据流、共享总线和交叉开关这四种互联方式满足不同模块的通信需求。

本文验证的模块是由 Wishbone-SPI 搭建的一个子系统，并着重阐述模块设计、验证环境搭建、寄存器模型生成、验证流程、激励用例开发以及功能覆盖率统计，最终对仿真结果进行统计和分析。

1 Wishbone-SPI 模块设计

SPI 作为 MCU 芯片重要外设，在芯片内完成串行数据流转换与并行数据流的转换[6]。该模块有多种传输模式，其最大输出时钟为系统输入时钟的一半。不同 SPI 模块的最大时钟需求不同，需要用时钟控制模块来调节

通信频率。

1.1 Wishbone 总线接口

Wishbone 总线信号接口分为公共信号、数据信号和总线周期信号。

公共信号包括复位信号 rst 和时钟信号 clk，它们对于主设备和从设备都是公共的。

总线控制信号有错误信号 err，重试信号 rty，响应信号 ack，读写使能信号 we，中断请求信号 int，选通信号 stb，总线周期信号 cyc，位宽信号 sel。

数据信号包括地址信号 addr，数据发送信号 dat_o，数据接收信号 dat_i。

Wishbone 支持多种单周期读/写，块读、读修改写。以单周期写为例，如图 1 所示，在 clk 上升沿 a 时刻，主设备设置位宽 sel 信号将地址信号 addr 和待写入数据 dat 发送到总线上，拉高 we，同时将 cyc 和 stb 置高，开始一次总线的写操作。在时钟上升沿 a 到达前，从机设备检测到主设备的写操作，锁存总线上的数据，同时拉高 ack 作为对主设备的响应。从设备可以在 ack 有效之间插入任意周期的等待状态。在 clk 上升沿 b 时刻，主设备发现 ack 为高，将 stb 和 cyc 置低表示操作完成。从设备发现 stb 为低后将主设备 ack 也置低。一次 Wishbone 写操作便完成。

1.2 SPI 模块

考虑到 SPI 协议接口信号具有不同通信模式的特点，SPI 模块设计采用基于状态机的方法，SPI 模块基本设计框架如图 2 所示。控制寄存器是 SPI 模块的核心，用于控制 SPI 通信方式，主要监视 Wishbone 总线上的读写命令，根据读写命令调用控制状态机，进而控制数据的发送与接收。分频器寄存器将模块的输入时钟分频后用于时钟产生输出 SCK，从而灵活地选择外设频率。状态寄存器反映模块内的运行状态，如发送完成标志，读写完成标志。数据寄存器存放待发送到芯片或接收来自于外部模块的数据。循环冗余校验(Cyclic Redundancy Check，CRC)校验器判断传输数据是否正确，当发送数据时将产生 CRC 校验码并附加到数据后面，接收数据时则自动计算 CRC 校验码并与总线数据上的校验码进行比较得出结果并反映到状态寄存器中。

2 UVM 验证平台

2.1 主要通用验证组件

基本的 UVM 平台包括以下部分：Test 是最顶层的类，首要负责验证平台的配置，创建下一级别的验证组件，可以用来启动 Sequence 开始仿真。Environment 是一个容器，包含较高级别验证组件，包括 Agent、Scoreboard、Config 以便更高层次的复用。Agent 是一个将 Driver、Monitor、Sequencer 三个组件进行封装的容器组件，方便相同功能的组件多次例化和复用。Driver 是将 sequence 中的数据包发送到 DUT 引脚上的组件，会根据特定的协议转化为时序激励。Monitor 用来观察接口上的数据或信号，可以通过事务极传输模型(Transaction－Level Modeling，TLM)将信号发送到其他组件上。Sequencer 管理不同类型的 Sequence，并带有仲裁的功能。Sequence 定义了需要发送或接收驱动器 Driver 的数据类型，需要挂载到相应的 Sequencer，是验证平台中流动的血液。Scoreboard 接收来自 Monitor 的数据值并且与期望值进行比较。接口 Interface 包含了连接、同步和两个或者多个模块之间的通信。

如图 3 所示，本次设计中验证环境 Wishbone2spi_env 中包含 Scoreboard、Coverage、Predictor、Agent 和 Transaction。Agent 内 Driver、Monitor 和 Sequencer 通过 TLM 通信。在 Sequencer 帮助下，顶层仿真序列 Sequence 产生的数据类型送给 Driver。总线上的读写操作激励将广播到 Predictor 上。Predictor 根据 DUT 配置和 Wishbone 总线上的输入来传输 SPI 的激励。Predictor 输出发送到 Scoreboard 上，并与 DUT 的输出结果进行比对。Coverage 记录 SPI 操作的功能覆盖率。Virtual_sequencer 控制不同的 Sequencer，起统一调度的作用[7]。平台中有 3 个 Sequencer，一个是寄存器模型的数据传输，一个是控制 Wishbone 到 SPI 的数

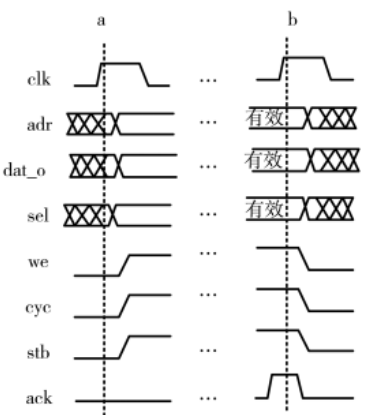


图 1 Wishbone 单次写操作

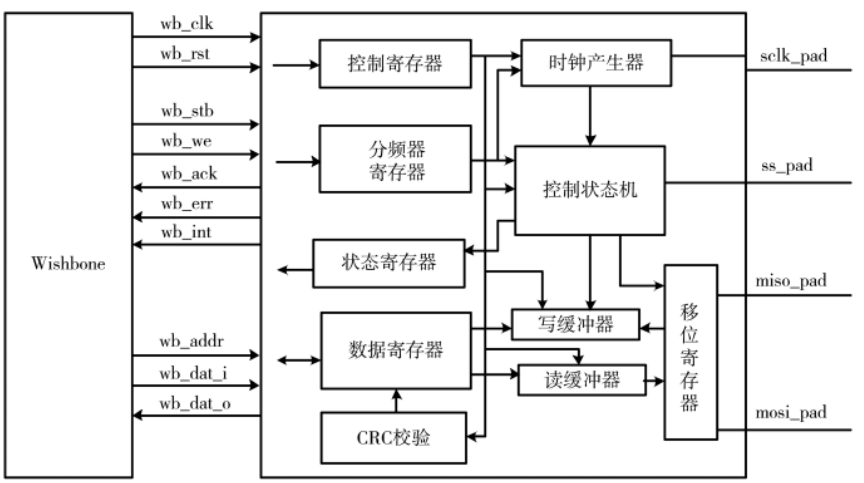


图 2 Wishbone－SPI 框图

据，还有一个则是控制 SPI 到 Wishbone 的数据产生。

通常 DUT 的接口数量不定，为满足芯片验证需求，加快验证的准备速度，通常相同功能验证组件封装在 Agent 容器内。如图 4 所示，通过在 Env 组件内的 config 配置不同 Agent 的 config 属性，实现验证环境集成多路接口驱动，并同时使用，满足验证的不同需求。

2.2 RAL 模型

寄存器抽象层(Register Abstraction Layer，RAL)是一种中心化管理寄存器配置的模型[8]，其核心思想是把需要验证的所有寄存器封装在一个统一模型内，通过映射完成对寄存器的验证工作。RAL 模型一般包括不同寄存器的地址和不同位宽的功能段域，其中段域是控制模块功能的最小单位，多个段域组成一个功能寄存器。使用 RAL 模型一般使用脚本语言进行管理，可以在大规模芯片设计中减少手动检查寄存器的时间，避免寄存器配置错误，实现平台复用性。

UVM 中对寄存器的访问方式包括前门访问和后门访问。前门访问是在总线上模拟时序，进而在寄存器模

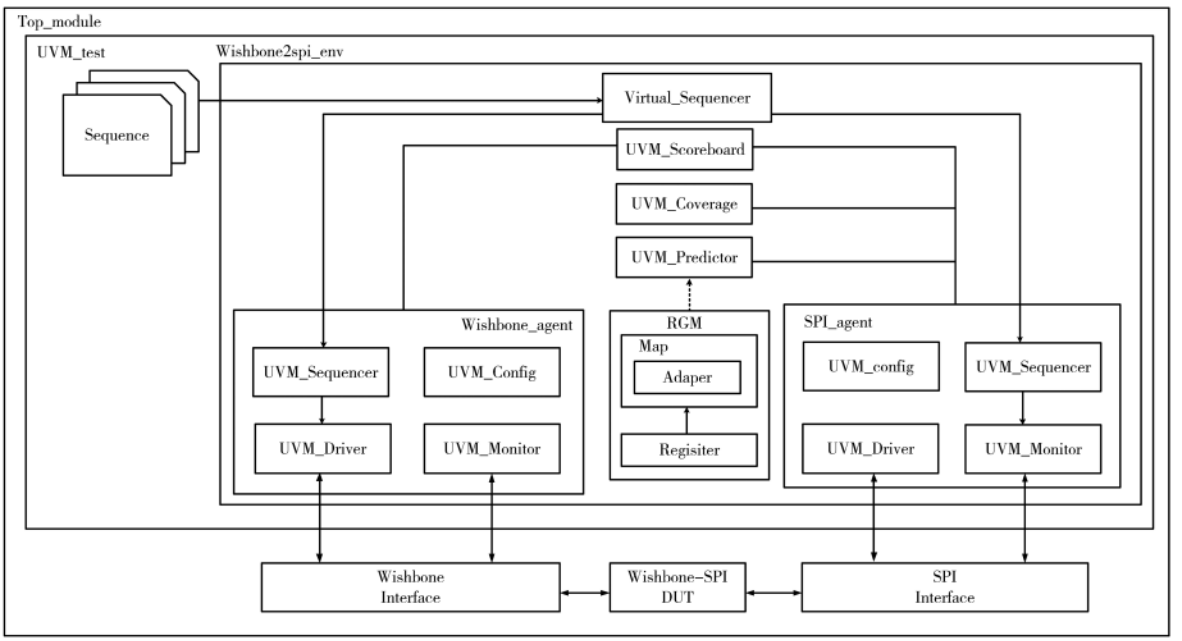


图 3 SPI 验证环境

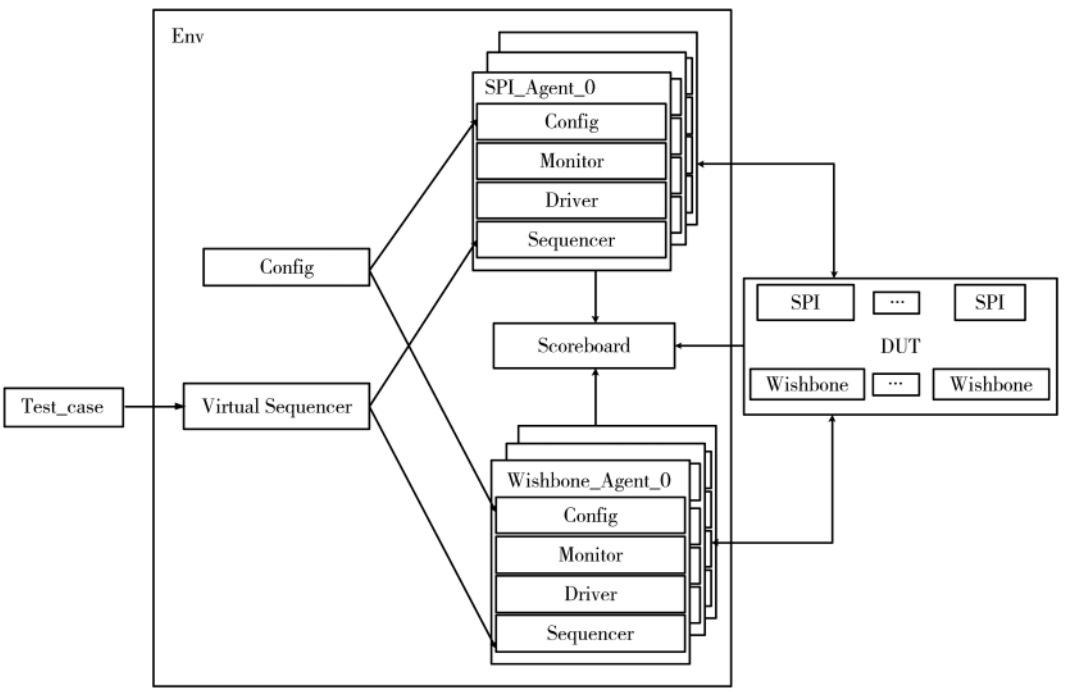


图 4 多驱动验证平台

型上进行读写操作，是真实的物理操作。而后门访问是利用 UVM 中的内置方法，将寄存器的操作直接作用到 DUT 内的寄存器变量，比起前门访问速度更快，段域值修改更便捷。前门访问方式使用转换器(adapter)类来充当不同抽象层转化媒介。该类将寄存器激励信息转化后发送到总线上，最后由总线上的 driver 完成转化。控制寄存器结构如表 1 所示。

表 1 控制寄存器

位	31:17	16	15	14	13	12
复位值	0	0	0	0	0	0
读写方式	RO	R/W	R/W	R/W	R/W	R/W
段域名	RES	BIDIMODE	BIDIOE	CRCNEXT	CRCEN	MSTR
位	11	10	9	8	7	6:0
复位值	0	0	0	0	0	7
读写方式	R/W	R/W	R/W	R/W	R/W	R/W
段域名	SPIE	CPOL	CPHA	SPE	LSB	LEN

RES(Reserved)域为功能保留，只能读；BIDIMODE(Bidirectional data mode enable)为双向通信方向，写 1 时表示双向传输，为 0 时是单向传输。BIDIOE(Output enable in bidirectional mode)域用于双向通信模式下的传输方向；CRCNEXT(CRC transfer next)表示下一次传输为 CRC。CRCEN(Hardware CRC calculation enable)开启硬件 CRC。MSTR(Master selection)为主、从模式选择。IE(Interrupt enable)域表示中断使能；LSB(Least Significant Bit)域表示最低有效位收/发比特位置顺序；SPE(SPI Enable)域表示 SPI 使能。CPOL 与 CPHA 组合控制着 SPI 传输的四种模式。下面以 SPI 的控制寄存器 SPCR 为例，验证平台的寄存器缩略代码如图 5 所示。

```
virtual function void build();
    RES = uvm_reg_field::type_id::create("RES");
    BIDIMODE = uvm_reg_field::type_id::create("BIDIMODE");
    BIDIOE = uvm_reg_field::type_id::create("BIDIOE");
    CRCNEXT = uvm_reg_field::type_id::create("CRCNEXT");
    CRCEN = uvm_reg_field::type_id::create("CRCEN");
    MSTR = uvm_reg_field::type_id::create("MSTR");
    SPIE = uvm_reg_field::type_id::create("SPIE");
    CPOL = uvm_reg_field::type_id::create("CPOL");
    CPHA = uvm_reg_field::type_id::create("CPHA");
    SPE = uvm_reg_field::type_id::create("SPE");
    LSB = uvm_reg_field::type_id::create("LSB");
    LEN = uvm_reg_field::type_id::create("LEN");

    RES.configure(this, 15, 17, "RO", 0, 15'b0, 1, 1, 0);
    BIDIMODE.configure(this, 1, 16, "RW", 0, 1'b0, 1, 1, 0);
    BIDIOE.configure(this, 1, 15, "RW", 0, 1'b0, 1, 1, 0);
    CRCNEXT.configure(this, 1, 14, "RW", 0, 1'b0, 1, 1, 0);
    CRCEN.configure(this, 1, 13, "RW", 0, 1'b0, 1, 1, 0);
    MSTR.configure(this, 1, 12, "RW", 0, 1'b1, 1, 1, 0);
    CPOL.configure(this, 1, 10, "RW", 0, 1'b0, 1, 1, 0);
    CPHA.configure(this, 1, 9, "RW", 0, 1'b0, 1, 1, 0);
    SPE.configure(this, 1, 8, "RW", 0, 1'b0, 1, 1, 0);
    LSB.configure(this, 1, 7, "RW", 0, 1'b0, 1, 1, 0);
    LEN.configure(this, 7, 0, "RW", 0, 7'b7, 1, 1, 0);
endfunction
```

图 5 控制寄存器验证代码

依据模块功能手册，寄存器每个域配置主要包括位域所属的寄存器、段域的位宽大小、该段域最低位在寄存器中的位置、访问方式、段域是否易变、复位后的默认值、能否复位以及能否单独访问。其中访问方式主要指读写 RW(Read Write)、只读 RO(Read Only)、只写 WO(Write Only)等。

本次设计包含了 12 个 SPI 寄存器，如表 2 所示，除了控制寄存器，还有状态寄存器和数据寄存器等。每个寄存器位宽均是 32 位，偏移地址依次以 4 递增。发送数据寄存器和接收数据寄存器共享地址，通过控制寄存器中的读写标志区分。在传输启动前，写数据寄存器，则发送数据；在完成传输后接收 MISO 上的数据，读数据寄存器。Wishbone 总线上以前门访问的方式来配置 SPI 的功能。通过后门访问方式读取寄存器里的值，最后比较写入值和期望值是否一致。

表 2 SPI 寄存器堆

寄存器	偏移地址	位宽	访问	描述
RX0	0x00	32	读	发送数据寄存器 0
RX1	0x04	32	读	发送数据寄存器 1
RX2	0x08	32	读	发送数据寄存器 2
RX3	0x0C	32	读	发送数据寄存器 3
TX0	0x00	32	写	接收数据寄存器 0
TX1	0x04	32	写	接收数据寄存器 1
TX2	0x08	32	写	接收数据寄存器 2
TX3	0x0C	32	写	接收数据寄存器 3
CR	0x10	32	读/写	控制寄存器
DIV	0x18	32	读/写	时钟分频寄存器
NSS	0x1C	32	读/写	从设备选择寄存器
STATE	0x20	32	读	状态寄存器

寄存器模型验证流程如图 6 所示，首先寄存器序列 spi_reg_sequence 携带寄存器信息放到 uvm_reg_item 实例中，并送到 adapter。然后 adapter 抽取出信息，生成总线所需的 Wishbone_seq_item 类型，通过总线上 Agent 实例发送到 DUT 内，完成寄存器的配置。

2.3 平台测试用例

为了对 SPI 特性全面测试，编写如测试用例。测试用例列表及父子类关系如表 3 所示。1 号测试用例是为了测试 Wishbone 是否按照总线协议进行读写。2 号测试用例继承于 1 号测试用例，它对总线地址进行约束，限定访问地址在 SPI 寄存器地址范围内，且检测寄存器的读写方式，如状态寄存器只能读不能写。3~7 号测试用例均继承于 2 号，其中 3 号是测试 SPI 开启和关闭的激励，是发送或接收 SPI 数据的基础。4 号用来测试通信故障，如数据溢出、CRC 错误。5 号检测 NSS 信号片选引脚在主机模式下是否拉低指定的从机设备引脚。6 号测试用例检测复位时 SPI 各个寄存器是否是设定值。7 号关注 SPI 在不同传输模式下 MISO 和 MOSI 引脚的时序是否与协议一致。

2.4 功能覆盖率组

设计自动统计功能覆盖率的模型来评估验证结果

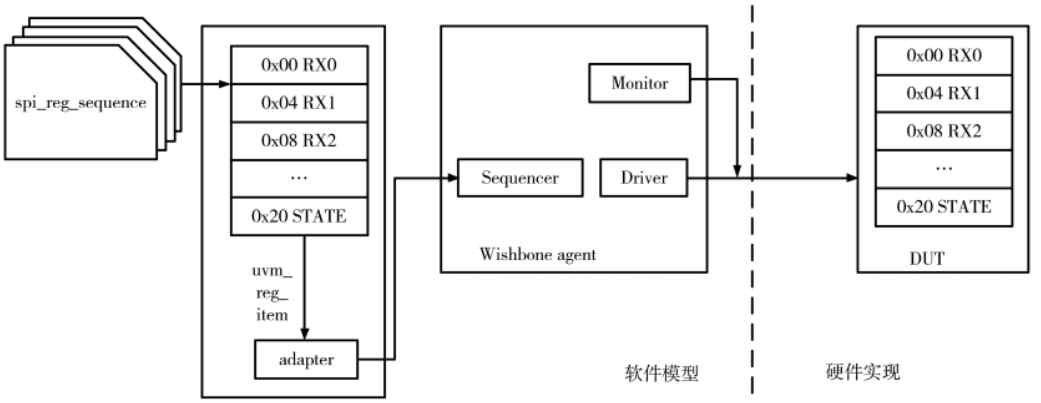


图 6　寄存器模型验证

表 3　测试用例列表及父子类关系

序号	测试用例	父类	描述
1	wb_sequence_base	–	总线上基本序列
2	wb_reg_sequence	1	SPI 寄存器测试
3	spi_enable_sequence	2	SPI 开关测试
4	spi_erro_sequence_base	2	SPI 通信错误测试
5	spi_nss_sequence	2	NSS 片选测试
6	spi_rst_sequence	2	SPI 复位测试
7	spi_trans_mode_seq	2	SPI 通信模式测试

表 4　SPI 测试覆盖组和描述

覆盖组	位置	描述
cg_reg_read	Reg_model	相关寄存器的读操作
cg_reg_write	Reg_model	相关寄存器的写操作
cg_div_read	Reg_model	分频器的读操作
cg_div_write	Reg_model	分频器的写操作
cg_tx_crc_erro	Reg_model	CRC 发送错误
cg_rx_crc_erro	Reg_model	CRC 接收错误
cg_spi_mode	spi_monitor	SPI 传输模式配置
cg_ewc	Reg_model	半双工通信模式
cg_fwf	Reg_model	全双工通信模式
cg_data_len	wb2spi_monitor	不同数据长度的数据流

是否满足预期要求，通过覆盖组实现覆盖率的统计模型，如表 4 包含验证平台不同覆盖组，以着重验证 SPI 各个基本功能。

3 功能覆盖率统计及分析

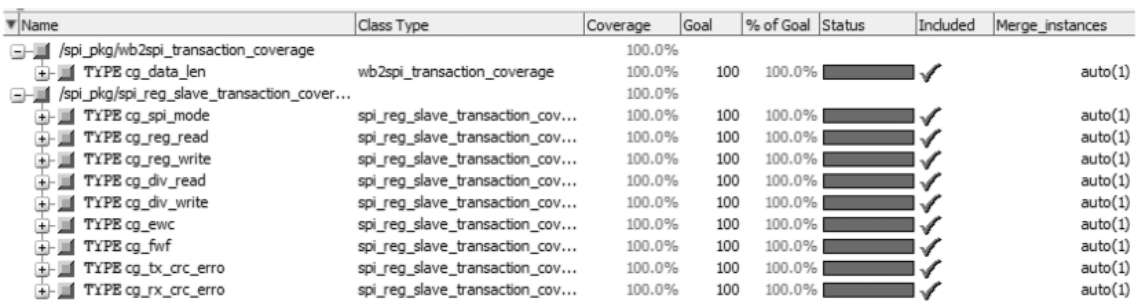
使用 Questasim 工具运行验证平台，激活测试用例，实例 Wishbone 和 SPI 接口并对接口信号分别进行采集。运行时，先通过受约束的随机测试，然后对未覆盖的地方进行定向测试，遍历访问空间，使功能覆盖率达到 100%。图 7 显示了 DUT 的功能覆盖率结果，每一个覆盖组的 Coverage=100.0%，这说明验证中每一种情况都已经覆盖，平台已达到验证目标。

4 结论

本平台支持 Wishbone-SPI 验证，使用 UVM 减少验证平台搭建时间，且功能覆盖率达到 100%，编写高效率的测试用例配合脚本工具实现可移植复用的激励。实际项目中，可依据项目实际情况改变接口代码或配置代码，而复用的代码结构对于其他验证工作起着积极的作用。后期在该验证平台上可以加上处理器外壳[9]，运行 C 的测试用例[10]，做出访问寄存器、处理中断等行为，还可进一步实现验证平台的垂直复用，能大大缩短 SoC 开发时间。

参考文献

[1] 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，2019，36(6)：79-83.
[2] 张强.UVM 实战[M].北京：机械工业出版社，2014.
[3] 王兴耀，戴宇杰.基于 UVM 的 AHB-UART 模块验证[J].南开大学学报(自然科学版)，2020，53(5)：82-86.
[4] 王刚，李冰，刘勇，等.基于 Wishbone 总线接口的 LDPC 码编码器设计[J].合肥工业大学学报(自然科学版)，2011，



Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances
/spi_pkg/wb2spi_transaction_coverage		100.0%					
TYPE cg_data_len	wb2spi_transaction_coverage	100.0%	100	100.0%			auto(1)
/spi_pkg/spi_reg_slave_transaction_cover…		100.0%					
TYPE cg_spi_mode	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_reg_read	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_reg_write	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_div_read	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_div_write	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_ewc	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_fwf	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_tx_crc_erro	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)
TYPE cg_rx_crc_erro	spi_reg_slave_transaction_cov…	100.0%	100	100.0%			auto(1)

图 7　功能覆盖率仿真结果

34(9):1324-1329.
[5] 李利利.基于 Verilog HDL 的 SPI 协议可复用 IP 软核的设计与验证[D].兰州:兰州大学,2015.
[6] 孙永涓.基于 AHB-Lite 总线的高速 SPI 接口的设计与实现[D].沈阳:辽宁大学,2021.
[7] 田晓旭,徐庆阳,汤先拓,等.基于 UVM 的寄存器验证自动化方法[J].集成电路应用,2020,37(2):18-21.
[8] 刘斌.芯片验证漫游指南[M].北京:电子工业出版社,2018:393-394.
[9] 任传宝,崔建国,鲁迎春,等.应用直接编程接口技术提高片上系统的 UVM 验证重用性[J].微电子学与计算机,2021,38(6):20-26,32.
[10] 李世超.基于 UVM 的 MC-SOC 中可重用验证平台的设计与实现[D].成都:电子科技大学,2018.

(收稿日期:2021-11-15)

作者简介:
刘森态(1996-),男,硕士研究生,主要研究方向:数字芯片验证、集成电路设计。
庞宇(1978-),通信作者,男,博士,教授,主要研究方向:集成电路设计,E-mail:pangyu@cqupt.edu.cn。
魏东(1997-),男,硕士研究生,主要研究方向:数字芯片验证、集成电路设计。

(上接第 35 页)

率和 94.4%的查准率,表现出了评估化合物致癌性的优异性。通过进一步分析,发现用分子描述符作为特征时,RF 和 XGBoost 模型效果准确率也达到 90%以上,这两种模型同样也适用于化合物致癌性的分类。将 SGCN 模型用于有毒气体分类上,准确率达到 89%,说明此模型在化合物分类判定上也有一定的普适性。

该研究探索了基于原子空间特征结合 SGCN 构建化合物致癌性分类模型的可行性,为化学物的健康风险评估提供依据,然而收集到的样本数和样本类别有限,需进一步增加样本量,使构建出的模型具有更好的泛化性和稳定性。

参考文献

[1] 方从兵,宛晓春,江昌俊.黄酮类化合物生物合成的研究进展(综述)[J].安徽农业大学学报,2005,32(4):498-504.
[2] AMRI N,WIRTH T.Recent advances in the electrochemical synthesis of organosulfur compounds[J].The Chemical Record,2021,21(9):1-13.
[3] FERLAY J,COLOMBET M,SOERJOMATARAM I,et al.Cancer statistics for the year 2020:An overview[J].International Journal of Cancer,2021,149(8):1-12.
[4] 程飞雄,沈杰,李卫华,等.有机化合物的陆地和水生环境毒性的计算机预测研究[J].农药学学报,2010,12(4):477-488.
[5] KINGMA D,BA J.Adam:a method for stochastic optimization[C]//Computer Science,2014.
[6] JIANG C,YANG H,DI P,et al.In silico prediction of chemical reproductive toxicity using machine learning[J].Journal of Applied Toxicology,2019,39(6):844-854.
[7] TOROPOVA A P,TOROPOV A A,MARZO M,et al.The application of new HARD-descriptor available from the CORAL software to building up NOAEL models[J].Food & Chemical Toxicology An International Journal Published for the British Industrial Biological Research,2018,112:544-550.
[8] 张晓昀,马卫平,刘满仓,等.人工神经网络用于多环芳烃及胆蒽系化合物致癌活性的研究[J].兰州大学学报(自然科学版),2004,40(1):38-44.
[9] 张振山,罗小民,郑明月,等.采用基于决策森林的分类方法预测化合物致癌毒性[C]//中国化学会第九届全国量子化学学术会议暨庆祝徐光宪教授从教六十年论文摘要集,2005.
[10] 谢莹,吴建国,李炜,等.基于 gSpan 算法的未知化合物毒性预测[J].合肥工业大学学报(自然科学版),2007,30(10):1278-1280.
[11] 梁倩倩,郑唯韡,何更生,等.基于分类和交叉参照的改良量化构效关系预测 N-亚硝基化学物的致癌性[J].中华预防医学杂志,2017,51(7):621-627.
[12] 阎爱侠,孔越.基于多领域数据信息融合的化合物致癌性预测[C]//中国化学会第 14 届全国计算(机)化学学术会议暨分子模拟国际论坛,2017.
[13] XU Y,DAI Z,CHEN F,et al.Deep learning for drug-induced liver injury[J].Journal of Chemical Information and Modeling,2015,55(10):2085-2093.
[14] DEFFERRARD M,BRESSON X,VANDERGHEYNST P.Convolutional neural networks on graphs with fast localized spectral filtering[C]//Lausanne:EPFL,2016.
[15] 任伟,孔德信.定量构效关系研究中分子描述符的相关性[J].计算机与应用化学,2009,26(11):1455-1458.

(收稿日期:2021-08-20)

作者简介:
魏若冰(1996-),女,硕士研究生,主要研究方向:图卷积网络算法。
何家峰(1970-),通信作者,男,副教授,主要研究方向:图像处理与模式识别,E-mail:jfhe@gdut.edu.cn。
邱晓芳(1996-),女,硕士研究生,主要研究方向:卷积神经网络、仿生嗅觉。

版权声明