

基于UVM的1000BASE-T图像传输系统的验证

王百慧

合肥工业大学, 安徽 合肥

收稿日期: 2022年3月2日; 录用日期: 2022年4月2日; 发布日期: 2022年4月8日

摘要

随着网络通信技术的发展,千兆以太网凭借可靠性高且传输速率快等特点,从而受到广泛的关注。图像的采集和传输也随之成为重要的发展方向,本文对1000BASE-T图像传输系统模块的功能进行分析和测试,并采用UVM (Universal Verification Methodology)通用验证方法学进行验证,搭建一个可重用性的验证平台,可重用性体现在从模块级到系统级的重用、不同测试环境的重用、不同项目之间的重用等。提出对图像传输系统的功能进行回归测试,最终以代码覆盖率、功能覆盖率、断言覆盖率结果达到100%为验收目标,即可确定该验证方法的有效性和完备性。

关键词

千兆以太网, UVM验证方法学, 可重用性, 功能覆盖率, 断言覆盖率

Verification of 1000BASE-T Image Transmission System Based on UVM

Baihui Wang

Hefei University of Technology, Hefei Anhui

Received: Mar. 2nd, 2022; accepted: Apr. 2nd, 2022; published: Apr. 8th, 2022

Abstract

With the development of network communication technology, Gigabit Ethernet has attracted wide attention because of its high reliability and fast transmission rate. Image acquisition and transmission have become an important development direction. This paper analyzes and tests the function of 1000BASE-T image transmission system module, and uses UVM (universal verification

methodology) to verify it to build a reusable verification platform. The reusability is reflected in the reuse from module level to system level, the reuse of different test environments, the reuse between different projects, and so on. It is proposed that the function of the image transmission system should be tested by regression. Finally, the effectiveness and completeness of the verification method can be determined by taking the result of code coverage, function coverage and assertion coverage as the acceptance target of 100%.

Keywords

Gigabit Ethernet, UVM Verification Methodology, Reusability, Function Coverage, Assertion Coverage

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着芯片自身复杂度的逐渐增大,其中包括芯片的功能愈加复杂,其制造工艺的愈加精细及其制造费用愈加昂贵,任何存在设计错误的情况都不能进入流片,其所带来的风险是不可承受的,因此验证也开始变得尤为重要,验证的目的就是为了找出设计过程中可能出现的问题和错误。

验证方法学经历了一些变化,从最初推出的 VMM (Verification Methodology Manual),到 OVM (Open Verification Methodology),再到 2011 年推出的 UVM, UVM 继承了前两种方法学的优点并且得到了三大 EDA 厂商的支持, UVM 已经得到了市场的肯定[1]。

本文需要实现 1000BASE-T 千兆以太网 MAC 层协议以及 UDP 协议,完成与 PC 机或其他设备进行千兆以太网通信。千兆传输广泛应用于广告屏显示、图像传输、游戏等,千兆以太网联合 DDR3 等存储器可以实现图像的存储,显示和传输,以 FPGA 为采集和传输的基础,克服了使用专用硬件灵活性不足且昂贵的缺点,并且利用 FPGA 的并行性可以实现高清的图像采集系统,通过千兆以太网传输的方式,可以在远程终端实现灵活的展示方式,如通过多终端的应用软件或通过网页的方式呈现[2]。文献[3]基于 UVM 的千兆以太网 MAC 控制器验证设计与实现采用以 SystemVerilog 语言结合 UVM 搭建验证平台,对整体硬件模块进行了功能验证,但其只收集了代码覆盖率和功能覆盖率,并没有做回归测试和收集断言覆盖率去判断时序的正确性,验证完备性有待提高。

传统的 VHDL 验证方法严重缺乏组织性,改变激励相当困难,对于不同场景下的激励传输需要对代码进行巨大改动,不仅需要大量时间去建立平台而且平台也极难维护,平台不可重用。UVM 验证方法学只有在发现设计存在漏洞时才需要对代码进行修改,且激励的产生是独立的,各个模块的环境是独立封装的,对外不需要保留数据端口,可以很快地创建新的测试,平台可扩展可重用,这是 UVM 平台巨大的优势。UVM 支持工业标准,可以促进平台的标准化,通过 OOP 面向对象编程的特点及使用覆盖率组件提高了重复使用率,这些都是 UVM 的好处。本文提出用 UVM 验证方法学,并且使用 SystemVerilog 语言进行编写测试, SystemVerilog 的优势在于能够允许用户在多个项目中使用统一的语法来构造可靠且可重用性的平台,该平台的搭建将会实现对图像采集、显示和传输等功能的全面验证。采用回归测试的策略确保能够按照预定目标实现新的功能,伴随着代码覆盖率、功能覆盖率、断言覆盖率的收集,通过不断地回归测试来逐步提高验证的完备性。

2. UVM 验证方法学及研究内容

2.1. UVM 验证方法学

UVM (Universal Verification Methodology), 其正式版本于 2011 年 2 月由 Accellera 推出, 并且得到了 Synopsys、Mentor 和 Cadence 的支持。验证语言 SystemVerilog 和 UVM 的结合使得用户可以不再受限于特定的仿真器, 可以随意切换到任何一家。这种通过 UVM 搭建的验证 IP, 可以很好地实现跨平台复用。UVM 平台框架可以帮助我们解决环境构建上的难题, 从而将注意力放在对 DUT 的测试功能点提取和测试用例上, 提高工作效率。

UVM 方法学的优势在于打通了各个 EDA 公司和 IC 公司的验证方法通道, 便于验证技术的交流和验证人员的技能提升, 也使得 IC 公司对于使用工具的选择广泛了许多。对于用户来说, 不再受限于使用哪一种仿真器及哪一家的验证 IP, 从而将主要的精力放在对设计的验证上, 从而提高验证的效率。其服务目的在于提供一些可以重用的类来减轻不同项目之间水平复用和垂直复用的工作量, 通过这种可复用性的平台可以为验证人员提高一套可靠性的框架[4]。

UVM 面向所有数字设计, 涵盖了从模块级到芯片级, ASIC 到 FPGA, 以及控制逻辑、数据通路到处理器验证对象的全部场景。

我们在探索 UVM 世界的过程中, 会逐步认识到 UVM 的类库地图和其工作机制, 了解 UVM 平台所需的组件和层次构建, 再到各组件如何建立通信及其测试场景的构建, 最后到其寄存器模型的应用。

2.2. 研究内容

本文研究的千兆以太网 PHY 芯片(VSC8601), 该芯片支持 10/100/1000BASE-T, 与我们电脑、FPGA、MCU 等处理器通信必须遵循 RGMII 协议, 使用 PC 机将图像数据信息通过千兆以太网发送给 FPGA, 存储到 DDR3 中, HDMI 将 DDR3 中数据读出显示到显示器上, 同时将图像数据回传到 PC 机对应的上位机上。硬件连接图如图 1 所示。通过搭建可重用性的验证平台来验证 FPGA 涉及到的所有功能模块, 其验证结果以覆盖率收集的结果作为测试功能的指标。

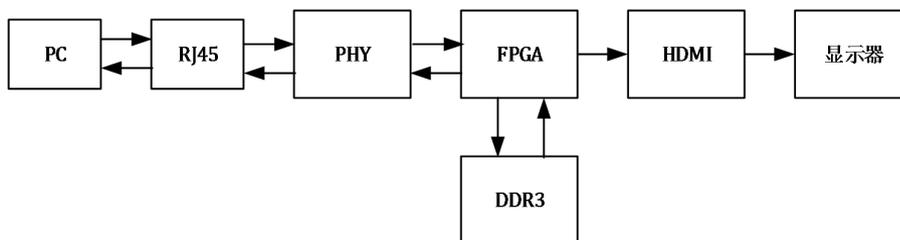


Figure 1. Hardware connection

图 1. 硬件连接图

3. 1000BASE-T 图像传输系统的设计模块

本文的验证对象是 1000BASE-T 图像传输系统模块, 在进行功能验证之前, 将对该系统的设计模块进行简单的介绍, 为后期搭建 UVM 验证平台建立理论基础。1000BASE-T 图像传输系统的设计模块框图如图 2 所示。

由图 2 可知, 系统主要分为 7 个模块, 分别包括 iddr_clk 数据转换模块、run_clk_ctrl 跨时钟域处理模块、image_ctrl 图像提取模块、gen_frame 数据帧传输模块、crc 校验模块、checksum 模块和 oddr_clk 数据转换模块。其中 iddr_ctrl 数据转换模块功能是将双沿 4 bit 数据转换成单沿 8 bit 数据。run_clk_ctrl

跨时钟域模块功能是采用双口RAM进行跨时钟域,将系统时钟经过PLL使得频率50 MHz变成125 MHz。*image_ctrl*模块功能是实现RGB图像信息提取。*gen_frame*数据帧传输模块功能是将数据组成基本的千兆以太网成帧数据。*checksum*模块功能是对IP_checksum和UDP_checksum进行计算,组成无crc校验的千兆以太网UDP协议包。*crc*校验模块功能是推导出32位crc校验分成4个字节到无crc校验的千兆以太网协议包后面,组成完整的千兆以太网UDP协议包。*oddr_clk*数据转换模块功能是将单沿8 bit数据转换成双沿4 bit数据。

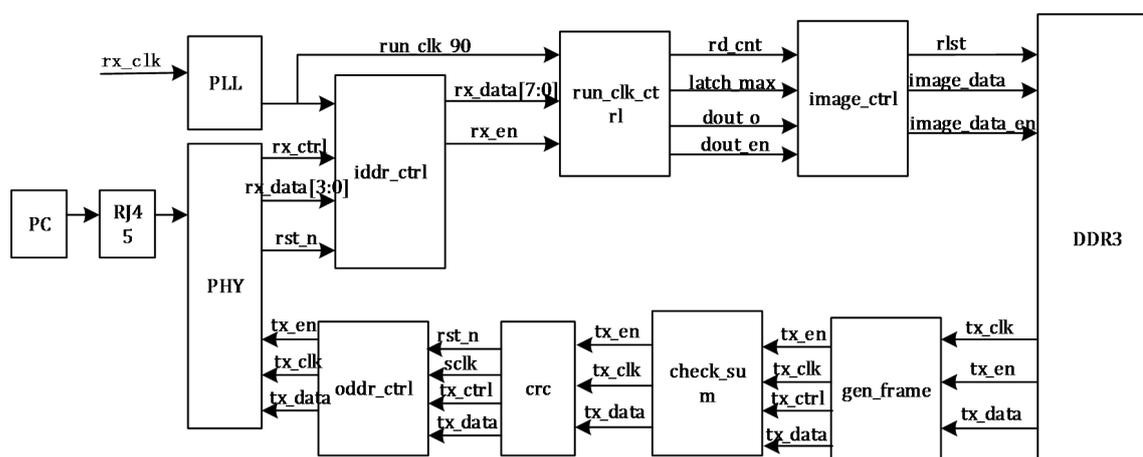


Figure 2. Design block diagram of image transmission system

图 2. 图像传输系统设计框图

本文主要研究内容是完成千兆以太网图像传输系统的功能验证,对整体功能和输入输出接口协议进行分析,对内部模块(7个模块)进行接口时序分析,明确验证目标,此外该设计还需要支持以下功能,支持RGMII接口协议、支持UDP协议、数据结构符合IEEE以太网标准。根据协议和功能需求,将该系统细分从行为到结构再到物理级,这种方式可以节省时间成本及提高设计效率,最重要的是独立模块可重复使用[5]。验证周期从创建验证计划开始,接着开发验证环境,当验证环境准备完成后,提供可供测试的激励对设计的输出结果进行比对,当硬件设计经历了大量的激励测试后,开始进行回归(regression)测试,即将已有的所有测试序列都执行一次。

3.1. UVM 验证平台的搭建

图3为UVM验证平台架构图,该平台架构总共包括了UVM平台常用的组件,如env(容器类)、agent(封装)、sequencer(序列发生器)、driver(驱动激励)、monitor(监测)、scoreboard(计分板)、reference_model(参考模型)、sequence(序列)等组件[6]。

通过interface接口来进行软件世界和硬件世界的连接,即平台和FPGA的信息交互[10]。平台相应地采用了6组不同的interface来进行连接,同时采用6组不同的agent封装所用到的sequencer、driver、monitor组件,agent通过模块的功能进行划分。

其中多个sequence同时挂载到相应的sequencer上面,创建所需的sequence_item激励。sequencer的作用是为了传递sequence_item到driver,同时还具备仲裁特性,允许特定sequence中的item激励优先通过。这时的sequence起到一定的协调作用,driver接收到item后,将激励发送到DUT的一侧,用monitor组件来监测DUT另一侧出来的数据信息,将其数据发送到scoreboard中,在scoreboard中进行期望数据和实际数据的比对[7]。

平台的启动通过 tb 文件中的 run_test() 进行, sequence 中的 body() 任务可以进行激励 item 的随机约束、准备发送激励 item 和完成发送激励 item 工作, 代码如下:

```

task body();
    uvm_sequence_item tmp;
    bus_trans req, rsp;
    start_item(req);
    req.randomize with {data == 10;};
    finish_item(req);
    get_response(tmp);
    `uvm_info("SEQ", $formatf("got a item \n %s", rsp.sprint()), UVM_LOW)
endtask
    
```

该平台用到的 virtual sequence 不同于常见的 hierarchical sequence, 不同点在于 hierarchical sequence 面向的对象是同一个, 而对于 virtual sequence 来说, 它内部不同的 sequencer 可以允许面向不同的 sequencer 对象。对于更上层的环境, 顶层的测试序列要协调的不再是面向一个 sequencer 的 sequence 群[8]。virtual sequence 可以承载不同目标的 sequencer 的 sequence 群, 它是一个连接所有底层 sequencer 句柄的 centralized 路由器。

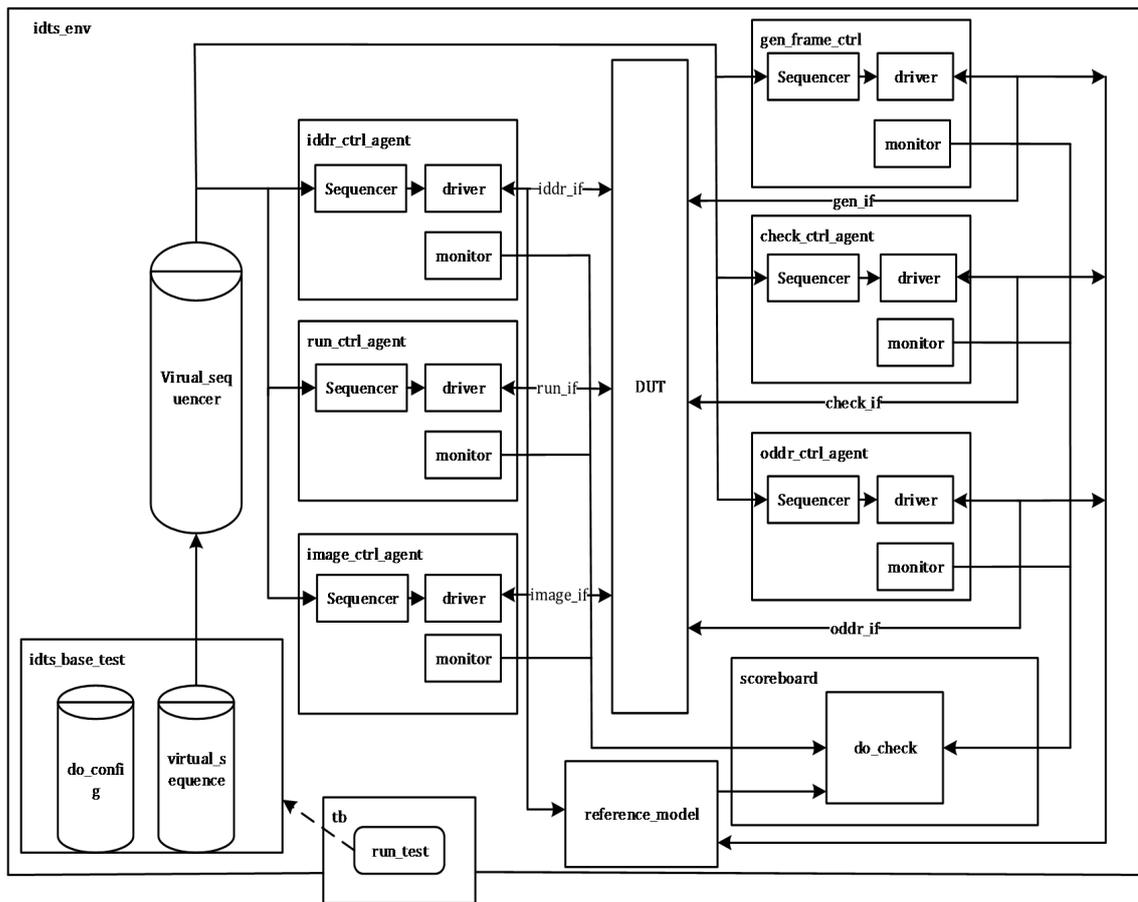


Figure 3. UVM verify platform architecture
图 3. UVM 验证平台架构

3.2. 验证方法

本文通过 SystemVerilog 语言搭建验证环境，该语言方便编写输入源，控制激励在不一样时钟周期的变化，而 Verilog 这种硬件描述语言需要综合成电路，因此它在书写测试平台和测试用例是不够方便的(测试平台和用例不须要综合成电路)。而 SystemVerilog 引入了面向对象编程 OOP 的概念，不需要满足可综合性。UVM 是一种库，调用库的时候，可方便使用库中各类写好的函数，是一个成熟的框架体系。通过 SystemVerilog 语言编写环境结合 UVM 的验证方法，可以支持回归测试的策略进行全面的验证。

4. 验证功能点

根据该图像传输系统的设计需求，使用表格的形式列出验证计划，验证计划是为了完成验证目标的，包含了需要验证的模块测试功能点。接着验证人员进行开发验证环境，在开发的过程中，验证人员和设计人员可以一同回顾验证计划，确保验证功能点没有遗漏的部分，这种验证计划的回顾可以作为一个检查点。列出的验证功能点分析表，如表 1 所示。

Table 1. Test function point analysis

表 1. 测试功能点分析表

功能测试点	功能测试点详解
数据转换	双沿 4 bit 数据转换成单沿 8 bit 数据
跨时钟域功能	将系统时钟频率 50 MHz 经过 PLL 变为 125 MHz
数据帧的组成	将数据组成基本的千兆以太网成帧数据
数据组成无 CRC 校验的千兆以太网 udp 协议包	数据组成无 CRC 校验的千兆以太网 udp 协议包
数据转换	单沿 8 bit 数据转换成双沿 4 bit 数据
支持 RGMII 接口速率工作模式	传输速率为 1000 Mbps，工作时钟频率为 125 MHz，前导码为连续 7 个字节的：0x55 起始符为一个字节：0xd5
接收端 CRC 校验	对接收到的以太网数据帧，进行 CRC 校验码计算
发送端 CRC 校验	对将要发送的数据帧，推导出 32 位 CRC 校验，并添加到数据帧末尾

4.1. 回归测试

回归测试指的是去验证硬件在某个缺陷修复或某项功能添加了之后，仍然可以通过以前的所有测试用例 case 和新添加的测试用例。这些可能存在的环境变化包括硬件设计自身的改进、缺陷修复、功能添加和验证环境的更新。回归测试的意义在于确保某项功能改动后没有引入新的缺陷并修复之前的漏洞，由于在做随机测试时默认的随机种子是不同的，伴随着代码覆盖率、功能覆盖率和断言覆盖率，可以通过反复的回归测试和补充定向测试来提高验证的完备性。代码编译完成后，输入仿真命令：`vsim -i -classdebug -solvefaildebug -coverage -coverstore test E:/work/lab/coverage(工程路径) mcdf_full_random_test -sv_seed random +TESTNAME=mcdf_full_random_test -l macdf_full_random_test.log work.tb`，其中的 seed 种子为随机种子，`run -all` 之后仿真结束，可查看所有的覆盖率数据。

4.2. 覆盖率

覆盖率是衡量设计验证完备性的一个指标，覆盖率工具在仿真过程中收集信息并且得到覆盖率报告。

通过这个报告可以发现覆盖之外的盲区，可以通过修改或创建新的测试进行填补。这个过程可以一直迭代，直到覆盖率达到预期结果。本文收集的覆盖率为三种，代码覆盖率、功能覆盖率和断言覆盖率。

代码覆盖率又包含行覆盖率、路径覆盖率、翻转覆盖率、状态机覆盖率[9]，其结果用于衡量执行了设计中的多少代码，关注点在设计代码的分析上。

功能覆盖率和功能设计意图紧密相连，可以发现某个功能在设计中的遗漏，这是和代码覆盖率不同的一点，每次仿真都会产生覆盖率数据库，将这些数据信息合并到一起即可得到功能覆盖率，通过分析覆盖率数据可以决定修改回归测试集。如果想要测试一些边界情况可以使用交叉(cross)覆盖率。覆盖组 covergroup 方法通过 sample()函数进行采样，get_coverage()函数获取覆盖率，start()和 stop()表示使能或关闭覆盖率的收集。

断言覆盖率中的断言表示一次性地对一个设计信号在逻辑上和时序上的声明，形式验证工具证明断言可以跟随设计和测试平台同时仿真，它常用于查找错误，一旦检测到，仿真立即停止。一般使用 cover property 来测量信号或状态是否发生，断言覆盖率数据会被集成在同一个数据库中。

5. 验证结果及分析

针对设计模块中的功能测试点，在不同的 test 仿真测试中做随机测试和定向测试，测试结果如表 2 所示。通过仿真工具 Questasim64_10.6c 进行回归测试，每个随机种子都会生成覆盖率数据，将这些覆盖率数据合并到一起，通过生成的 ucdb 文件查看最终收集到的覆盖率结果，如下图 4 所示。

Table 2. Functional coverage results

表 2. 功能覆盖率结果

测试 test	测试结果/%	总覆盖率/%
随机测试 idts_full_random_test	70.3	100
定向测试 idts_data_consistence_basic_test	94.5	100
定向测试 idts_gen_frame_data_test	94.7	100
定向测试 idts_reg_stability_test	95.3	100

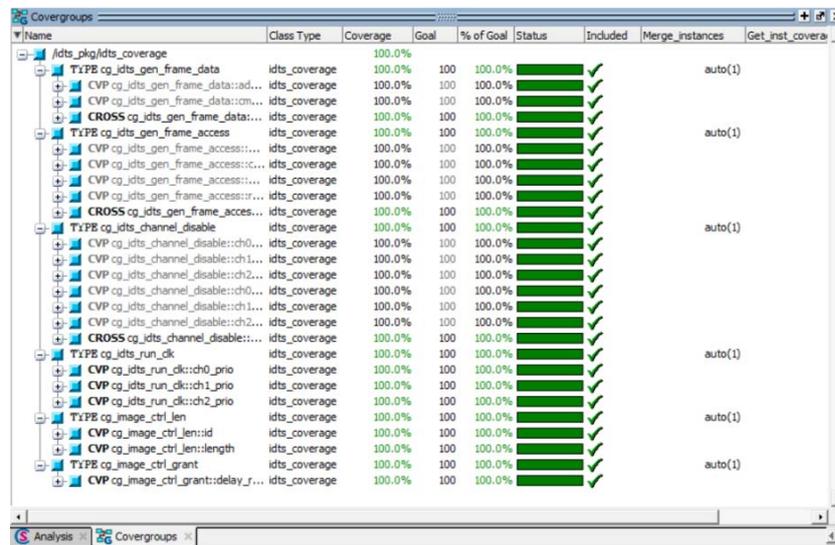


Figure 4. Coverage collection and analysis

图 4. 覆盖率收集分析图

Table 3. HTML report
表 3. HTML 报告

Coverage Summary by Structure			Coverage Summary by Type						
Design Scope	Hits %	Coverage %	Total Coverage (Filtering Active)				94.68%	94.68%	
idts_pkg	100.0%	100.00%	Coverage Type	Bins	Hits	Misses	Weight	Hit %	Coverage
idts_coverage	100.0%	100.00%	Covergroups	70	70	0	1	100.00%	100.00%
			Statements	274	243	20	1	90.64%	90.64%
			Branches	190	167	23	1	92.66%	92.66%
			Toggles	1600	972	613	1	90.12%	90.12%
			Assertions	16	16	0	1	100%	100%

做回归测试进行了大量的随机种子测试，如果有一些功能点无法覆盖到，针对未覆盖到的这些功能点，添加新的定向测试，测试结束后，合并所有的覆盖率 ucdb 文件，使用 Questasim64_10.6c 来打开 UCDB 利用工具来查看覆盖率，打开当前覆盖率数据库的同时生成 HTML 报告，这种方式更加直观有效。由上述表 3 结果可知，其中仓(bin)作为衡量功能覆盖率的基本单位，它能够记录数据被采样的次数，所有的 covergroup 的覆盖率构成了整个平台的覆盖率，Covergroups 覆盖率和 Assertion 断言覆盖率都达到 100%，表示验证的模块功能均已实现且 RTL 设计逻辑和时序正确。

6. 结语

UVM 是一个以 SV 语言类库为主体的验证平台架，利用其可重用性可以构建具有标准化层次结构和接口的功能验证环境[10]。UVM 验证方法学有效结合了测试激励随机生成、自测试平台和随机化约束等方法，验证了设计代码的可行性。通过代码覆盖率、功能覆盖率、断言覆盖率达到 100%，其中断言可以针对设计上的内容进行检查，证明该验证平台可以实现模块功能点的测试完备性及该图像传输系统中的模块能够正确的执行各自的功能行为，同时平台具有可重用性。这种验证方法可以降低风险及节省时间，可以缩短产品上市时间，并且在芯片开发上具有相当重要的价值。

参考文献

- [1] 张强. UVM 实战[M]. 北京: 机械工程出版社, 2014.
- [2] 徐世林. 基于 FPGA 的图像采集与传输[D]: [硕士学位论文]. 武汉: 华中科技大学, 2019.
- [3] 麻聪颖. 基于 UVM 的千兆以太网 MAC 控制器验证设计与实现[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2020.
- [4] 刘斌. 芯片验证漫游指南: 从系统理论到 UVM 的验证全视界[M]. 北京: 电子出版社, 2018.
- [5] 李涛. 基于 100BASE-TX 以太网 PHY 芯片设计与验证[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2018.
- [6] 雷霆. 基于 UVM 的网络数据包解析器的验证研究[D]: [硕士学位论文]. 成都: 电子科技大学, 2016.
- [7] (2009) IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language—Redline. *IEEE Std 1800-2009 (Revision of IEEE Std 1800-2005)—Redline*, 1-1346.
- [8] 徐梓文, 郭桂良. 基于功能覆盖率的 MAC 的 UVM 验证[J]. 计算机与数字工程, 2021, 49(11): 2282-2285.
- [9] 赵森. 基于 UVM 的 10M/100M 以太网 PHY 验证方法研究[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2019.
- [10] (2020) IEEE Standard for Universal Verification Methodology Language Reference Manual. *IEEE Std 1800.2-2020 (Revision of IEEE Std 1800.2-2017)*, 1-458. <https://doi.org/10.1109/IEEESTD.2020.9195920>