基于SV-DPI的图像坏元修正FPGA自动化验证

李艳龙，杨 琪，王雪峰
（西安微电子技术研究所，陕西 西安 710065）

摘要：为实现红外图像坏元修正FPGA（field programmable gate array）的快速验证，提高测试覆盖性，设计了基于SV-DPI（SystemVerilog-direct programming interface）的FPGA自动化验证平台。采用DPI（direct programming interface）编程接口技术，实现了SystemVerilog平台调用C＋＋编程语言，构建了针对红外图像坏元数据的生成和检测修正模型，建立了两种语言在事务级（transaction level）模型的通信。结果表明相对于传统验证方法，该平台结构简单，可以快速实现激励产生、参考模型构建、测试结果自动比对等功能，实现了红外图像坏元检测与修正 FPGA 的自动化测试，功能覆盖率达到100%，有效缩短FPGA测试平台搭建和调试周期，提高了测试效率和测试质量。

关键词：自动化测试；FPGA验证；红外图像坏元修正；验证平台
中图分类号：TP306 **文献标识码：**A **文章编号：**1001-8891(2020)12-1192-06

Automatic Verification of Field Programmable Gate Arrays for Dead Pixel Correction

LI Yanlong，YANG Qi，WANG Xuefeng
(*Xi'an Microelectronic Technique Institute*, *Xi'an* 710065, *China*)

Abstract: To accelerate the simulation speed and improve the coverage of verification for a field programmable gate array (FPGA) implemented with dead pixel correction of an infrared image, an FPGA automatic verification platform based on SystemVerilog-Direct programming interface(SV-DPI) was designed. Using DPI programming interface technology, the C++ programming language was invoked by the SV platform. A generator and correction model for dead pixel data of infrared images was built. This established a communication between two languages on the transaction level. The results show that, compared with the traditional verification method, the proposed platform is simple in structure and can quickly generate a test vector, construct a reference model, and check results automatically. It realizes automated verification for an FPGA implemented with dead pixel detection and correction of an infrared image. The function coverage can reach 100%. It effectively shortens the period of construction and debugging for the FPGA verification platform and improves the efficiency and quality of verification.

Key words: automated testing, FPGA verification, dead pixel correction of infrared image, verification platform

0 引言

红外焦平面阵列成像系统以其结构简单、探测能力强、稳定性好、功耗低等优点而广泛应用于武器装备领域，在目标跟踪等方面起到十分重要的作用[1]。然而红外焦平面器件受材料、工艺的影响，不可避免地存在坏元[2]。坏元大小通常仅有一个像素，表现为亮点或暗点，会直接影响探测器的成像质量，从而影响与图像处理相关的目标检测、目标识别等处理结果。通常需要对未知坏元位置的图像进行坏元检测与修正[3]等预处理。

图像预处理数据量较大、实时性要求高，FPGA芯片是目标硬件的理想选择之一[4]。随着设计复杂度和规模的不断加大，FPGA在实际应用中发生故障的可能性也相应提高，将直接影响产品的可靠性[5]。测试验证是保证FPGA设计质量的重要环节，随着型号

收稿日期：2020-04-20；**修订日期：**2020-06-21.
作者简介：李艳龙（1988-），男，吉林白城人，工程师，硕士。研究方向为FPGA测试、测试自动化技术。E-mail：470968999@qq.com。

任务的不断增加，测试人员需要具备在短时间内完成对复杂 FPGA 设计进行充分、全面验证的能力[6]。图像规模不断增大，处理算法日趋复杂，验证难度也在随之增加，甚至由于无法构建出这类复杂验证模型而不能采取传统的验证方法，影响了验证效率和完备性。FPGA 设计具有灵活性高、功能差异化大的特点，测试验证时需要针对性的编写测试组件，难以建立高复用性测试平台，然而人工搭建测试平台耗时占整个功能验证周期的 20%～30%。因此，要提高 FPGA 的验证效率，必须对测试平台搭建进行优化和改进。

目前，对于红外图像坏元检测与修正 FPGA 的测试验证手段一般为样例数据对比、采用 UVM（universal verification methodology）验证方法学、使用 Matlab 进行联合仿真。但现有验证方法都存在各自的局限性：

1）使用设计师提供的样例数据进行结果比对时，其测试用例会存在局限性，如提供的数据是否正确有效，是否能够保证功能覆盖性，是否提供了异常情况的激励等，无法保证验证工作的独立性和有效性。

2）UVM 验证方法学虽然具有移植性好、重用性高的优点，尤其适合大规模 IC（integrated circuit）设计的验证，在团队协作、维护方面优势明显[7]。但 UVM 验证平台的搭建比较复杂，相比于直接测试用例，UVM 学习成本更高、搭建验证平台的时间更多。因此，使用 SystemVerilog 直接测试的方式更有效率。

3）HDL（hardware designed language）仿真工具结合 Matlab 进行数据生成、数据对比也可以实现复杂算法类 FPGA 的验证。由于测试过程中需要对大量测试用例进行随机化，生成及导入数据的操作灵活性较差，耗费大量时间，难以分析验证结果。

基于现有验证方法的局限性，为了快速实现红外图像坏元检测与修正算法建模，快速建立自动化验证平台，减少验证过程的人为干预，提高验证效率和覆盖性，本文设计了基于 SV-DPI（SystemVerilog-direct programming interface）的图像坏元检测与修正 FPGA 自动化验证平台，采用 SystemVerilog 验证语言结合 DPI 编程接口技术，实现了在仿真验证平台中调用 C＋＋语言，采用面向对象编程，来快速实现红外图像坏元检测与修正算法功能的建模。

1 基于 SV-DPI 的自动化验证方案

基于 SV-DPI 的自动化验证利用 SystemVerilog 的直接编程接口（DPI）[8]连接 C＋＋编程语言，实现 SystemVerilog 和 C＋＋语言之间的数据通信。该平台结合了 C＋＋编程语言和 SystemVerilog 验证语言的优点，实现激励随机约束生成，自动化结果检查的功能。

信号级通信属于较低级别的模型，通信效率较低。当需要创建复杂设备模型时，面向对象语言能极大提高设计效率，使用事务级通信会使仿真的速度加快。本文验证方案的核心设计原则是使用 SystemVerilog 建立基本架构，包括 TestBench、驱动器、监控器、检查器等；使用 C＋＋实现复杂算法，如数据生成器、参考模型等；建立两种语言之间的事务级（transaction level）模型通信，最终实现自动化验证。

1.1 被测对象功能分析

开展测试前首先要分析被测对象的功能特点，梳理测试功能点，并设计测试用例。

由于应用领域及使用场景的特殊性，要求 FPGA 设计具有高可靠性、高容错性以及妥善处理异常情况的能力。作为独立测试人员，需要在有限时间内全面、充分地设计测试用例，保证测试质量。所以测试用例不仅应该覆盖所有正常情况，还应涉及到边界内、边界上、边界外，特殊位置（坏元位置在图像边缘、出现连续坏元），坏元数量的性能测试、恢复性测试等。采用受约束的随机激励和直接测试激励相结合的方式，可以有效保证测试的覆盖性。

1.2 平台构建

验证平台主体架构基于 SystemVerilog 语言编写，图像数据生成以及参考模型部分使用 C＋＋语言编写。可实现多种图像数据（正常图像、存在不同坏元的图像）的自动化随机生成，图像数据的驱动注入和监控，图像数据的输出采集，与参考模型数据的自动化比对功能。

验证平台能够在 HDL 仿真工具下运行，整体结构如图 1 所示。

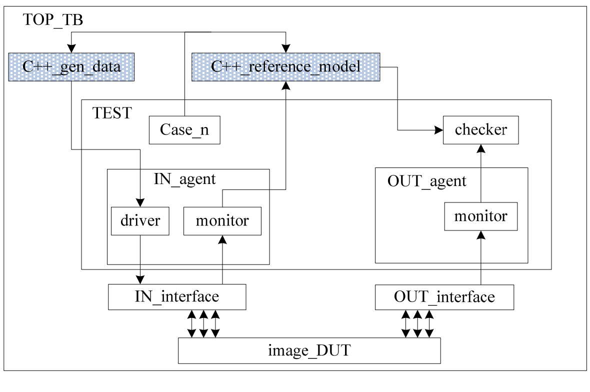

图 1 验证平台整体结构

Fig.1 Structure of the verification platform

验证平台由 6 个模块组成，分别是用例控制模块

（Case_n）、图像数据生成模块（C＋＋_gen_data）、图像数据注入模块（IN_agent）、图像数据输出采集模块（OUT_agent）、参考模型模块（C＋＋_reference_model）和数据结果比对模块（checker）。图像数据生成和参考模型模块使用C＋＋语言编写，其他模块使用SystemVerilog语言编写，各模块功能如下：

1）用例控制模块的功能是确定需要执行的验证用例，控制图像数据的生成，通知参考模型当前的工作参数。

2）图像数据生成模块接收用例控制模块参数后，通过算法生成相应的图像数据，通过DPI接口将数据传递给图像数据注入模块。

3）图像数据注入模块接收图像数据，按照器件时序将数据驱动到DUT（design under test）上，同时使用内部monitor将端口数据回采后输出到参考模型。

4）图像数据输出采集模块能够按照输出接口时序协议，监控DUT输出的图像数据，并将数据打包传输给数据比对模块，同时监测接口时序，实现接口时序的自动化检查。

5）参考模型模块模拟了DUT功能，将注入端口输入到DUT的数据作为参考模型输入，使用用例控制模块的配置参数进行数据处理，处理后的数据传输到数据结果比对模块。

6）数据结果比对模块将图像数据输出采集模块采集到的图像数据与参考模型计算的数据进行数据比对，输出结果信息和数据信息，实现数据的自动化检查。

1.3 图像数据生成模块设计

图像数据生成模块使用C＋＋程序模拟测试图像的生成，使用类（class）实现数据生成算法，静态封装为OOP（object oriented programming）接口供SystemVerilog调用。该模块接收用例控制模块的输入参数生成不同的测试图像，采用约束随机方式生成。具体过程为先随机生成一帧正常的图像，然后再将不同数量、位置、像素值的坏元随机注入到图像中，并记录坏元的坐标和数量。最终将生成的一帧图像传递到图像数据注入模块。图像数据的关键信息均能够打印到控制台上，方便验证人员调试。所生成的图像数据能够确保覆盖全部测试用例，保证测试覆盖性。逻辑流程如图2所示。

程序结构图如图3所示。图像数据生成的C＋＋程序结构中包含4个类：ImgEngine、ImgGen、BadPointGen、BPInsert。ImgGen用于生成原始图像；BadPointGen用于生成坏元的信息，包含了坏元位置、坏元类型、坏元位置类型；BPInsert用于实现坏元的注入，该模块接收坏元特征信息，将坏元替换到中心像素中；ImgEngine为顶层控制逻辑，实例化上述类，实现单个坏元、连续坏元注入以及坏元位置的遍历等测试场景。

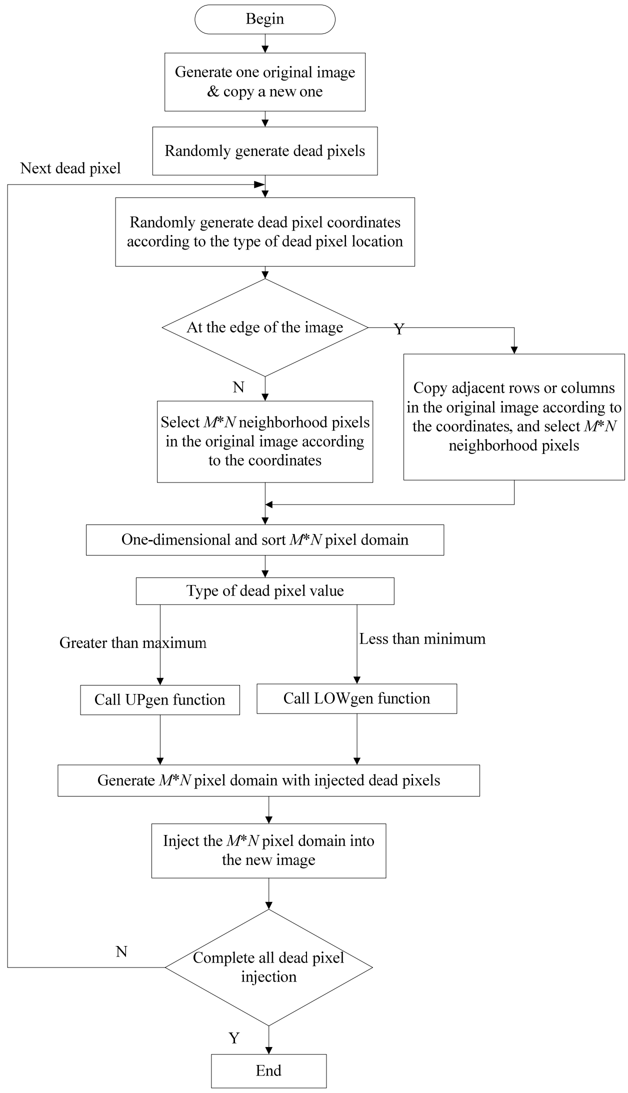

图2　图像生成逻辑框图

Fig.2　Logical block diagram of image generation

1.4 参考模型设计

参考模型模块使用C＋＋程序模拟图像坏元检测与修正功能，该模块将最终处理完的图像数据输出到数据结果比对模块进行比对。同样使用类（class）实现参考模型处理算法。接收AD模块传来的图像数据，对$M\times N$区域像素进行排序，分别找到最大值，最小值和中值。接收用例模块传来的配置阈值等有关参数，来决定是否进行坏元修正。如果满足阈值要求则使用中值替换当前像素，记录当前像素位置，修正计数器加一，逻辑流程如图4所示。

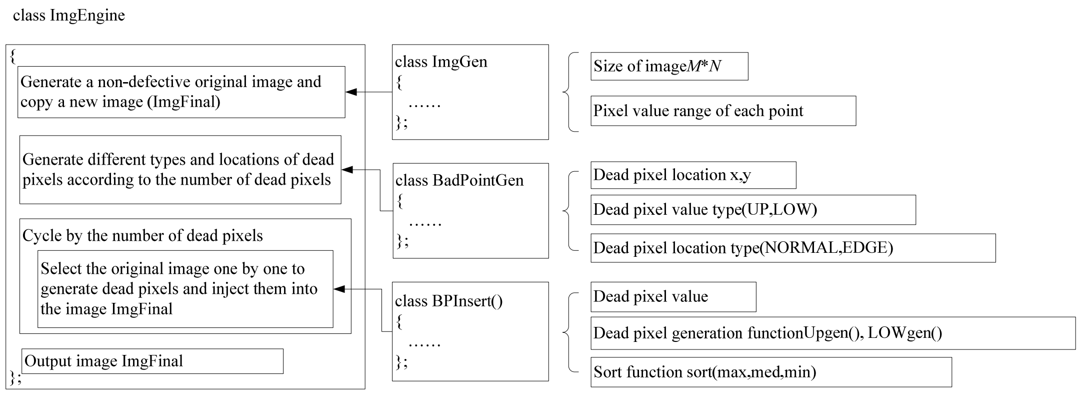

图 3 图像数据生成程序结构图

Fig.3 Structure chart of image data generation program

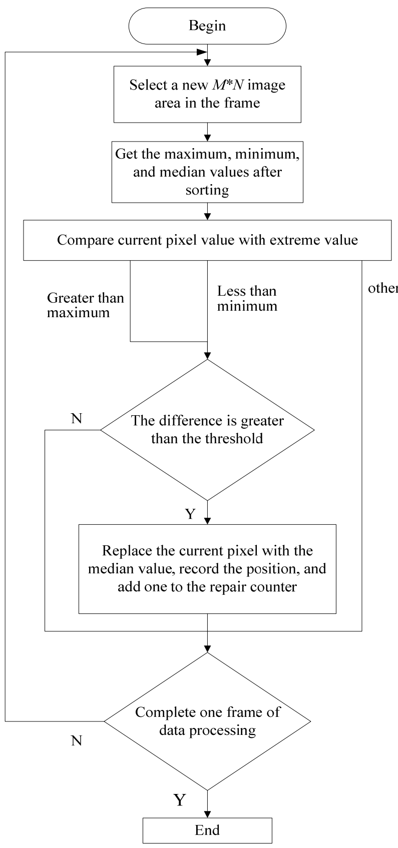

图 4 参考模型逻辑流程图

Fig.4 Logic flow chart of reference model

2 实际应用效果分析

该红外图像坏元检测与修正 FPGA 所实现的功能为，将输入图像经 AD（analog digital）接口进行采集，处理后的数据经 LVDS 输出。外部 DSP 作为控制主机，能够通过总线接口配置阈值等参数。处理过程是对输入的图像逐帧先检测坏元再修正。处理方式以待测像素点为中心，在 3×3 邻域内进行判别，对比当前待测像素点与邻域内其他像素点之间的亮度差值，当差值大于设定的坏元阈值时，则判断当前待测像素点为坏元。最后采用中值滤波算法进行坏元修正。

根据上述功能提取测试功能点，设计了如表 1 所示的测试用例。

依照上述自动化验证方案设计测试平台，其中 3×3 区域内寻找中值和极值的算法采用快速中值滤波，原理如图 5 所示。对于图中的图像区域，首先进行一次列排序，得到 A、B、C 由大到小的排列顺序。再进行一次行排序，得到由大到小的排列顺序，则得到最大值 max_{max}、最小值 min_{min}、和中值 Med[min_{max}, med_{med}, max_{min}][9]。

利用该自动化验证平台，执行了如表 1 所示中的 10 个测试用例，实现了图像坏元测试激励的随机约束生成，数据结果自动化检查，测试点覆盖了边界内、边界上、边界外，特殊位置（坏元位置在图像边缘、出现连续坏元），坏元数量的性能测试，恢复性测试，全面覆盖了 DUT 的测试充分性要求。结果表明本文构建的红外图像坏元数据的生成和检测修正模型实现正确有效，设计的测试用例覆盖全面、可控，实现了图像坏元检测与修正 FPGA 的自动化测试，功能覆盖率达到 100%，确保了测试效率和测试质量。

与传统验证方法相比具备如下优点：

1）通过高级语言实现红外图像坏元检测与修正复杂模型比 HDL 语言更加容易，并且 FPGA 中实现的算法一般都会有成熟、可靠的高级语言模型，直接调用这些模型能够极大提高平台搭建速度。相比使用传统单纯采用 SystemVerilog 以及 UVM 验证方法，极大提高了平台搭建速度，节省时间超过 40%（如表 2 所示），从而有效缩短了 FPGA 整个功能验证周期，提高了验证效率。

表1　测试用例及执行情况表

Table 1　Table of test case and execution

No.	Case type	Test case	Desired result	Execution result	Function coverage
1	Normal	The current pixel value is between the maximum and minimum values in the 3×3 neighborhood	The current pixel is not replaced and no correction is made	Effectively executed and passed	100%
2	Boundary	The current pixel value is greater than the maximum value in the 3×3 neighborhood, and the difference from the maximum value is less than or equal to the set threshold	The current pixel is not replaced and no correction is made	Effectively executed and passed	
3	Normal	The current pixel value is greater than the maximum value in the 3×3 neighborhood, and the difference from the maximum value is greater than the set threshold	The current pixel value is replaced by the median value in the neighborhood. The correction is implemented.	Effectively executed and passed	
4	Boundary	The current pixel value is less than the minimum value in the 3×3 neighborhood, and the difference from the minimum value is less than or equal to the set threshold	The current pixel is not replaced and no correction is made	Effectively executed and passed	
5	Normal	The current pixel value is less than the minimum value in the 3×3 neighborhood, and the difference from the minimum value is greater than the set threshold	The current pixel value is replaced by the median value in the neighborhood. The correction is implemented.	Effectively executed and passed	
6	Special location	The bad elements are on the four edges	The current pixel is not replaced and no correction is made	Effectively executed and passed	
7	Special location	There are more than three bad elements in the image, and the positions of the bad elements are continuous	No correction	Effectively executed and passed	
8	Performance	The number of bad elements accounts for 60% of an image, and the positions of bad elements are evenly dispersed	All bad elements are corrected	Effectively executed and passed	
9	Normal	Input bad elements one by one, traverse the whole frame of image	Every bad element is corrected	Effectively executed and passed	
10	Recovery	First input an image with bad elements, and then input an image without bad elements	The bad element of the first frame of image is corrected correctly, and the second frame of image is output correctly without error correction	Effectively executed and passed	

A_0	B_0	C_0
A_1	B_1	C_1
A_2	C_2	C_2

⇨

A_{max}	B_{max}	C_{max}
A_{med}	B_{med}	C_{med}
A_{min}	B_{min}	C_{min}

⇨

max_{max}	med_{max}	min_{max}
max_{med}	med_{med}	min_{med}
max_{min}	med_{min}	min_{min}

图5　快速中值滤波

Fig.5　Median filtering

表 2 搭建平台时间对比

Table 2 Comparison of platform building time

Types of platforms	Time consuming to build a platform/d
Pure SystemVerilog verification platform	3
UVM verification platform	3.7
The automated verification platform designed in this article	1.6

2）相对于 UVM 验证方法学，该平台结构简单、化繁为简，具备良好的灵活性和易维护性，不需要配置复杂的环境参数，降低了测试人员的学习成本，缩短了开发调试时间，非常适合 FPGA 的测试验证。无需调用 UVM 大量的资源库，计算机资源占用更少，结合事务级模型通信使得运行效率得到提高。

3 总结

本文实现了基于 SV-DPI 的红外图像坏元修正 FPGA 自动化验证，采用 SystemVerilog 语言和 C＋＋语言相结合的验证方案，构建了针对红外图像坏元数据的生成和检测修正模型，通过 DPI 实现 SystemVerilog 平台与事务级 C＋＋模型的通信。该验证方案结合了 SystemVerilog 事物处理能力强大的优势，以及 C＋＋实现模型成熟、稳定、重用性高的优点，相对于传统验证方法，平台结构简单，可以快速实现激励产生、参考模型构建、测试结果自动化检查等功能，实现了红外图像坏元检测与修正 FPGA 的自动化测试，功能覆盖率达到 100%，有效缩短了测试平台搭建时间，提高了测试效率和测试质量，切实有效地保证产品的可靠性。

参考文献：

[1] 白俊奇，蒋怡亮，赵春光，等. 红外焦平面阵列探测器盲元检测算法研究[J]. **红外技术**, 2011, **33**(4): 233-235, 240.

BAI Junqi，JIANG Yiliang，ZHAO Chunguang, et al. Blind-pixel detection algorithm for infrared focal plane array detector[J]. *Infrared Technology*, 2011, **33**(4): 233-235, 240.

[2] 李炎冰，梁少峰，陈洪亮，等. 红外焦平面阵列坏元检测算法[J]. **电光与控制**, 2015, **22**(3): 69-71.

LI Yanbing, LIANG Shaofeng, CHEN Hongliang, et al. A bad-pixel detection algorithm for infrared focal plane array[J]. *Electronics Optics & Control*, 2015, **22**(3): 69-71.

[3] 胡云生，胡越黎，燕明，等. 一种图像坏点检测及修正算法[J]. **上海大学学报**：自然科学版, 2018, **24**(5): 755-762.

HU Yunsheng, HU Yueli, YAN Ming, et al. Detection and correction of dead pixels in an image[J]. *Journal of Shanghai University*: Natrual Science, 2018, **24**(5): 755-762.

[4] 李雷鸣，张焕春，张波. 一种基于 FPGA 的图像中值滤波器的硬件实现[J]. **电子工程师**, 2004(2): 48-50.

LI Leiming, ZHANG Huanchun, ZHANG Bo. The realization of image median filter based on FPGA[J]. *Electronic Engineer*, 2004(2): 48-50.

[5] 周珊，杨雅雯，王金波. 航天高可靠 FPGA 测试技术研究[J]. **计算机技术与发展**, 2017, **27**(3): 1-5.

ZHOU Shan, YANG Yawen, WANG Jinbo. Research on high reliable FPGA test technology on space flight[J]. *Computer Technology and Development*, 2017, **27**(3): 1-5.

[6] 朱伟杰，阳徽，费亚男，等. FPGA 功能验证自动化技术研究与实践[J]. **航天控制**, 2017, **35**(2): 72-77, 98.

ZHU Weijie, YANG Hui, FEI Yanan, et al. Research and practice of automation technology on FPGA function verification[J]. *Aerospace Control*, 2017, **35**(2): 72-77, 98.

[7] 习建博，朱鹏，崔留争. 基于 UVM 方法的 FPGA 验证技术[J]. **电子科学技术**, 2016, **3**(3): 204-207.

XI Jianbo, ZHU Peng, CUI Liuzheng. FPGA verification technology based on UVM[J]. *Electronic Science & Technology*, 2016, **3**(3): 204-207.

[8] 克里斯·斯皮尔. SystemVerilog 系统验证[M]. 2 版：张春等译. 北京：科学出版社, 2009.

Chris Spear. *SystemVerilog for Verification*[M]. 2nd Edition: Translated by ZHANG Chun , et al. Beijing: Science Press, 2009.

[9] 韩团军. 快速中值滤波算法研究及其 FPGA 硬件实现[J]. **电子器件**, 2017, **40**(3): 697-701.

HAN Tuanjun. Research of fast median filtering algorithm and hardware implementation based on FPGA[J]. *Chinese Journal of Electron Devices*, 2017, **40**(3): 697-701.