

# FPGA硬核处理器系统加速数字电路功能验证的方法

刘小强 袁国顺\* 乔树山

(中国科学院微电子研究所 北京 100029)

(中国科学院大学 北京 100049)

**摘要:** 为了缩短专用集成电路和片上系统的功能验证周期, 该文提出FPGA硬核处理器系统加速数字电路功能验证的方法。所提方法综合软件仿真功能验证和现场可编程门阵列原型验证的优点, 利用集成在片上系统现场可编程门阵列器件中的硬核处理器系统作为验证激励发生单元和功能验证覆盖率分析单元, 解决了验证速度和灵活性不能统一的问题。与软件仿真验证相比, 所提方法可以有效缩短数字电路的功能验证时间; 在功能验证效率和验证知识产权可重用方面表现优于现有的FPGA原型验证技术。

**关键词:** 专用集成电路; 功能验证; 片上系统; FPGA原型验证; SoC FPGA

中图分类号: TN492

文献标识码: A

文章编号: 1009-5896(2019)05-1251-06

DOI: 10.11999/JEIT180641

## Accelerating Functional Verification for Digital Circuit with FPGA Hard Processor System

LIU Xiaoqiang YUAN Guoshun QIAO Shushan

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

(*University of Chinese Academy of Sciences, Beijing 100049, China*)

**Abstract:** In order to reduce the functional verification cycle of application-specific integrated circuits and on-chip system, a method for accelerating functional verification with FPGA digital hard processor system is proposed. The proposed method combines the advantages of software simulation function verification and field programmable gate array prototype verification, and uses the hard processor system integrated in the on-chip system field programmable gate array device as the verification excitation generation and the function verification coverage analysis unit. It solves the problem that verification speed and flexibility can not be unified. Compared with software simulation verification, the proposed method can effectively shorten the functional verification time of digital circuits; it is superior to existing FPGA prototyping technology in terms of functional verification efficiency and verification of intellectual property reusability.

**Key words:** Application Specific Integrated Circuit (ASIC); Functional verification; System-on-Chip (SoC); Field-Programmable Gate Array (FPGA) prototype verification; SoC FPGA

## 1 引言

随着专用集成电路(Application Specific Integrated Circuit, ASIC)和片上系统(System-on-Chip SoC)芯片的功能复杂性增加和规模增大, 数字芯片验证已经成为了整个设计的瓶颈, 占用了设计流程中大部分工作量<sup>[1]</sup>。设计早期阶段主要体现在芯片的功能验证, 芯片设计中的逻辑错误需要在设计

早期被发现, 以避免逻辑功能错误导致不必要的芯片设计返工风险<sup>[2]</sup>。通常使用的功能验证技术有3种, 基于软件仿真功能验证、硬件加速器和现场可编程门阵列(Field-Programmable Gate Array, FPGA)原型验证, 软件仿真验证是应用最为广泛的一种功能验证方式, 其中基于通用验证方法学(Universal Verification Methodology, UVM)的功能验证技术是目前最为常用的。文献[3,4]提出了利用UVM对设计进行功能验证, 基于仿真的功能验证优点是互操作性强、验证知识产权(Intellectual Property, IP)可重用, 可以使用高级语言和硬件描述语言(Hardware Description Language, HDL)协同完成验证, 具有完善的功能验证覆盖率分析方

收稿日期: 2018-07-02; 改回日期: 2018-01-10; 网络出版: 2019-01-22

\*通信作者: 袁国顺 mryuangs@hotmail.com

基金项目: 国家自然科学基金(61474135)

Foundation Item: The National Natural Science Foundation of China (61474135)

案,并且在可视化和验证报告输出方面具很大优势。然而,基于软件仿真的功能验证在应对多功能和大规模的系统级设计时,由于需要大量的验证激励,而运行这些激励需要时间以天为单位计算,功能验证周期长。为了加速这一过程,硬件加速器和FPGA原型验证往往是众多设计者考虑的解决方案,但是硬件加速器系统非常昂贵,难以让许多使用者、项目与团队广泛的运用,新版定制设计芯片与系统需要耗费时间来开发。FPGA原型验证被认为是有效解决验证周期长的一种方法<sup>[5]</sup>,并被广泛用于设计早期的功能验证中。文献[6]中使用FPGA对快速傅里叶变换算法原型进行功能验证;文献[7]中引入状态机对所设计的处理器的缓存单元进行功能验证;文献[8]采用FPGA对精简指令集处理器(RSIC)进行验证。尽管FPGA原型验证可以有效地缩短验证周期,但是传统的基于FPGA原型验证<sup>[7,8]</sup>需要设计人员使用底层描述语言HDL来实现几乎所有的验证特征,即使简单的验证激励修改也需要对整个设计重新综合及适配,在修改验证激励和重新综合这一过程非常耗费时间,也很容易出错;在功能验证覆盖率分析方面只能依赖于外围设备功能的具体表现。在文献[9,10]中研究人员尝试采用软硬件结合的方式来提升基于FPGA原型验证的互操作性,本文通过对软件仿真功能验证和基于FPGA原型验证的研究提出采用FPGA硬核处理器系统加速数字电路功能验证的方法。

本文第1节是引言部分。第2节对软件仿真功能验证和现有的FPGA原型验证技术实现进行分析对比;第3节阐述HPS加速数字电路的功能验证实现;第4节为实验结果的分析与讨论;第5节介绍该文的结论。

## 2 ASIC/SoC功能验证技术分析与对比

### 2.1 软件仿真验证

通用验证方法(UVM)可以提高互操作性,降低新项目或电子设计自动化(Electronics Design Automation, EDA)工具的知识产权的成本,并且更容易提供可重用验证组件EDA工具支持<sup>[11]</sup>。基于软件仿真验证的UVM平台见图1,验证平台分为4个层次:场景层、功能层、命令层和信号层,采用System Verilog, C/C++等高级语言和HDL语言来描述验证激励发生单元,模拟上层数据源和总线模型功能接口,将验证激励以一定的队列顺序送入到被验设计(Device Under Test, DUT)中。软件仿真功能验证的一般流程,首先对功能验证激励的需求和DUT的接口进行分析,构建验证平台并设计总线模型功能接口和验证激励发生单元。激励通过

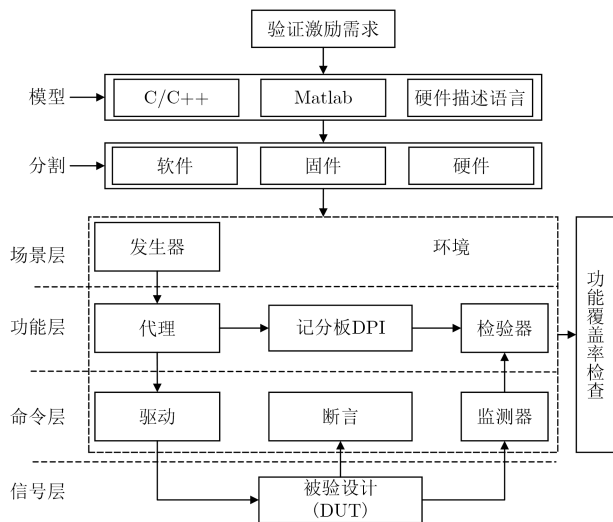


图1 软件仿真验证平台

总线模型功能接口进入DUT中,在完成一次验证后根据输出报告判决设计中是否出现功能错误或者设计缺陷,由功能覆盖率检查单元对被验设计进行功能验证的覆盖率性进行检查和报告。同时对于关键信号可以在验证过程中以界面可视化和日志记录的方式展示出来。

### 2.2 FPGA原型验证

FPGA原型验证技术被视为ASIC/SoC芯片设计流片前进行功能验证的最重要的技术之一,其验证结果与实际流片结果最为接近。验证激励发生单元采用HDL语言以可综合的形式实现或者由外接输入设备提供,经过对每个验证激励多次调试,由可综合的总线模型单元或者自定义的接口送入被验证单元,构建完成后进行逻辑综合、布线、适配到FPGA中进行验证,以修改原始设计添加观测点的物理运行结果来判别设计功能错误和缺陷,或者以成熟的功能验证单元来检验。现有的FPGA原型验证往往可以及时发现设计中潜在功能缺陷,然而验证激励产生效率低,验证时可观测难度高,验证IP可重用性差<sup>[12]</sup>,在验证调试中缺乏一个可视化的环境,在准确定位功能设计缺陷方面还不足;功能验证覆盖率分析方面还未有完整的解决方案。

文献[9,10]中提出了一种改进型FPGA原型验证技术将验证激励发生单元由计算机替代以提高验证的互操作性,使用千兆以太网向DUT中输入验证激励。主要的验证任务由计算机完成,在计算机负责以太网数据传输、验证激励产生和验证功能覆盖率统计。验证激励通过以太网传输到FPGA原型验证板卡,在FPGA中设计有专用的以太网数据解析单元,数据解析完成后被发送到先入先出(First Input First Output, FIFO)单元中,这时需要专用单元将接收的数据转发到总线功能模块中进一步解

析,通过总线接口对DUT进行验证,同时总线功能模块负责把来自于DUT对激励响应结果数据以相应的方式被打包发送到计算机。改进型的FPGA原型验证技术尽管可以通过与软件层协同实现功能验证,但需要引入专用以太网接口,验证激励需要经过多级转发,增加了验证的硬件层的复杂性。通信接口在与DUT通信带宽上表现不足,对于验证数据高吞吐率的设计不利,同时不具备专用的验证调试环境,在验证互操作性上还不足。

FPGA技术的发展使得可编程的器件的集成度有了很大的提升,这使得在单片FPGA器件上实现更为复杂和完整的系统成为了可能,SoC-FPGA是在传统的FPGA架构中集成了硬核处理器系统(Hardware Processor System, HPS)<sup>[13,14]</sup>,HPS包括中央处理器(Central Processor Unit, CPU)、高速存储控制器和通用外部设置,HPS与FPGA通过快速通道互联(Quick Path Interconnect, QPI)传输数据,由于SoC-FPGA的高效率和高集成度,在文献[15]中的系统整体加速方面得到了很好的应用,本文引入HPS使得功能验证的大部分的工作依然由高级语言来完成。

### 3 HPS加速数字电路的功能验证

#### 3.1 验证系统的实现与分析

图2为详细验证系统硬件实现图,所提出功能

验证方法相对于现有的FPGA原型验证硬件部分增加了HPS。验证系统在硬件设计上主要由HPS、协议桥、DUT和嵌入式逻辑分析仪(Embedded Logic Analyzer, ELA)<sup>[16]</sup>以及板级上必要的存储单元构成,HPS的优势在于不消耗FPGA逻辑资源、高效地执行能力和高可靠性。在软件层主要使用高级语言C/C++来编写验证激励程序,在编写验证激励上比HDL效率上有很大的提升,程序由HPS中的硬核CPU执行,这样即使验证激励发生变化时不需要对硬件部分改动即可进行验证,可以避免文献[7,8]中在验证激励改变时需要重新综合整个设计,因为复杂的设计往往综合将耗时数十分钟或者数个小时;同时不需要像软件仿真功能验证技术一样需要重启整个验证环境。

本文所提方法需要设计相应的协议桥,协议桥的作用是将来自于HPS激励数据以一定的时序正确地输入到DUT中,同时将来自于DUT的激励响应数据返回到HPS。被验证的对象DUT往往拥有不同的总线接口,通常是标准的总线接口和自定义总线接口并存,而由CPU分发出的总线接口是固定的且符合标准协议,具体CPU分发的总线类型取决于所选用的SoC FPGA器件,目前主流SoC-FPGA器件集成的CPU是ARM Cortex-A系列,所分发出的总线接口是先进可扩展接口(Advanced eXtensible Interface, AXI)。同UVM验证一样分析总线接

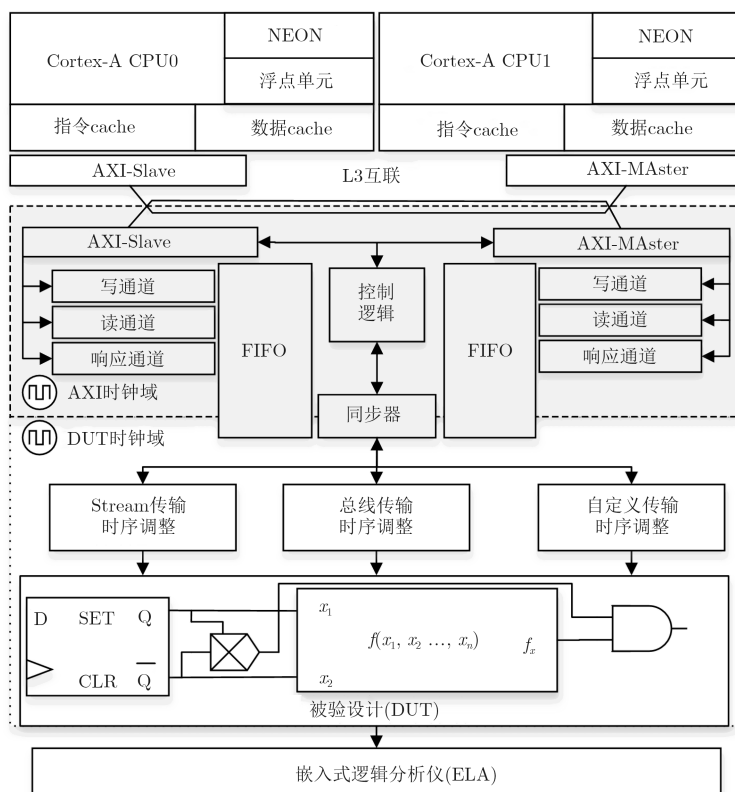


图2 验证系统硬件实现图

口特性与设计桥接单元是启动验证的重要的一步,硬件部分仅需设计由AXI接口到DUT协议桥接,例如DUT使用的是高性能总线(Advanced High performance Bus, AHB)接口则需要设计AXI-to-AHB的协议桥。根据接口速度不同可以考虑使用高速或者低速AXI接口。实际中对于复杂设计需要考虑DUT与AXI不在同一时钟域问题,即需要跨时钟域处理,设计协议桥时需引入FIFO和同步器。为了快速找到验证中出现功能错误点,使用ELA对DUT内部信号观测是必要的,将DUT中的关键控制信号添加到ELA中,在验证过程中实时观察。

图3是本文所提HPS加速数字电路功能验证的实现流程,在验证流程上与UVM验证的流程基本一致,首先分析需要验证的功能,并对相应的DUT中的单元进行模型分析,通过C/C++或者Matlab等高级语言方式获取模型参考数据为后续的验证提供黄金参考以判定映射到寄存器传输级的设计功能是否正确<sup>[17]</sup>。接下来会对验证系统的层次分割,主要是划分HPS软件层和FPGA硬件层如图4,这两层由协议桥隔离和连接的,激励自顶向下,以特定的时序进入DUT,时序调整工作由协议桥完成,结果由底向上进入HPS供CPU分析并给出功能覆盖率结果。准备验证激励输入,分析DUT的接口构成并设计相应的协议桥接与HPS系统通信。设计功能验证的程序,利用HPS在控制方面的优势充当验证激励发生单元,HPS同时负责DUT对控制,激励通过协议桥进入DUT中,对DUT进行初始化、各子组件功能验证、系统整体验证,DUT会

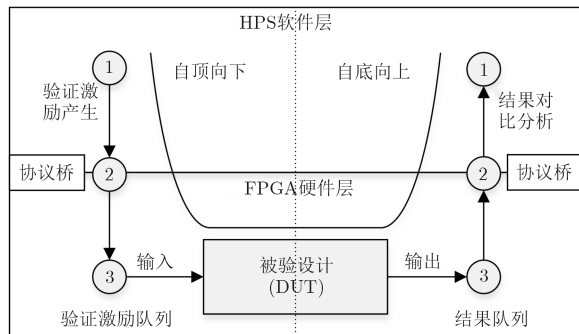


图4 验证软硬件分割图

对验证进程中每个阶段的激励给出响应,HPS对得到的响应数据与参考数据进行对比来评估功能的正确性,并对功能覆盖率进行统计。验证发生错误时需要借助对错误的逻辑点进行分析,ELA只解决了在硬件层的具体观察问题,而无法控制验证进度,某一功能错误往往只是验证进程中的一个具体的点,这时可以通过HPS的软件开发环境调试,通过在验证程序中加入断点、单步调试等方式与ELA结合可以快速定位功能缺陷点。

### 3.2 功能验证覆盖率分析

ASIC/SoC芯片验证中功能覆盖概率是一项非常重要的参考指标,它是衡量验证激励对DUT的覆盖程度以决定何时结束验证进程<sup>[9]</sup>,验证过程也是不断提升功能覆盖率的过程。在功能验证开始阶段需要定义功能验证方案,对所需要验证的功能详尽列出,本文中施加激励的过程即是实现功能验证的具体阶段。激励进入被验单元后会由被验单元返回结果激励,结果激励通过桥接口由CPU获取与标准的参考激励进行对比分析,并据此确定哪些功能已经被验证过,对于通过功能验证的进行标记并计算覆盖率结果,未通过的功能验证项不计入覆盖率,最终将所有的功能验证结果汇总并给出具体的报告以便后续详细分析。对于未通过的功能验证需要修改软件层的功能验证程序进行覆盖来提升功能验证覆盖率直到达到目标覆盖率。

### 3.3 验证IP的复用

验证中可重用IP验证环境在理论上始终是理想的方法,但在实际实施中具有挑战性<sup>[4]</sup>,本文所提出的加速功能验证方法在验证IP和验证驱动的重用性方面与软件仿真技术一致。主要表现在协议桥和验证程序是分布在硬件层和软件层两个不同层面。协议桥基于通用协议和用户自定义协议,采用所提出技术只需设计一次硬核处理器系统到对应DUT的协议桥,重新验证或者新的验证工程中具有同样接口的DUT可以重复使用。软件层面的验证函数是基于通用C/C++语言实现,HPS可以很好地实

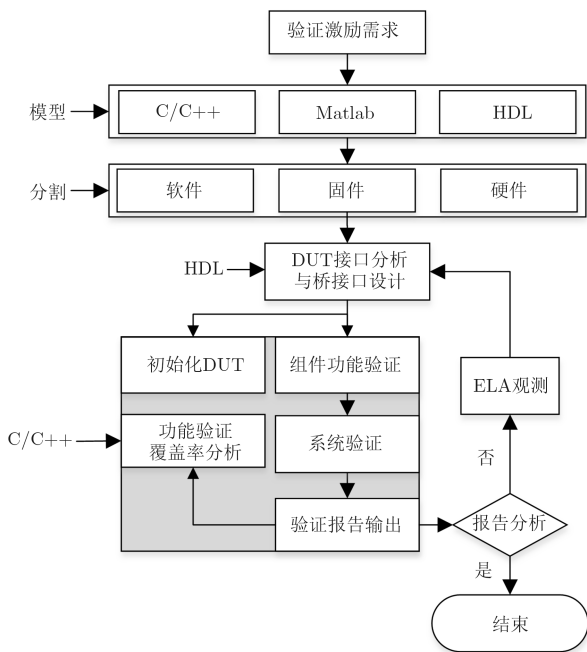


图3 功能验证技术的实现流程图

现软硬件隔离，即使协议桥接口发生变化，软件层可以无需改动，即验证所使用的程序及流程不会因桥接口的改变而改变，验证协议桥和验证程序具有很好的重用性。在进行新的验证时可以直接使用进而缩短验证系统搭建周期，这一点优于现有FPGA原型验证技术。

4 测试结果与讨论

本文提出加速数字电路功能验证的实验基于Intel Cyclone V SoC器件，在Quartus Prime 17.1平台下实现。实验对图像处理子系统验证，主要对图像处理算法、存储控制器和外设控制器进行验证，逻辑电路规模68.2 kLEs。验证系统的协议桥和DUT工作在98.6 MHz，HPS运行频率设置为800 MHz，为了保证验证激励在产生时不会被过度优化设置软件层的编译优化选项为O0级别，软件仿真在Linux环境下使用QuestaSim 10.4e进行。

图5(a) 将本文方法与软件仿真技术、传统的FPGA原型验证、改进的FPGA原型验证技术在验证时间上的对比，从结果可以看出软件仿真和本文提出方法的功能验证时间均与激励数目成正比，随着激励的增加所需的验证时间将会增加。在实验条件下，软件仿真功能验证所需要得时间是本文方法的1132倍，本文方法略优于现有FPGA原型验证技

术。图5(b) 是现有FPGA原型验证技术和本文方法在验证IP消耗逻辑资源方面的对比，传统FPGA原型验证技术随着验证激励的增加，逻辑资源消耗也随之增加，激励数目的增加意味着需要更多的逻辑资源实现验证状态机；由于本文方法和改进型FPGA原型验证技术将软件层和硬件层隔离，所以验证IP逻辑资源消耗与激励数目无关，验证IP逻辑资源消耗保持稳定水平。然而改进的FPGA原型验证技术需要消耗更多的逻辑资源去实现高速接口和数据转发接口，所以在资源消耗方面高于本文提出的技术。较少的资源消耗使得单片FPGA中能够容纳更多的设计，降低使用多片FPGA分割设计带来验证不准确的风险，也减少了复杂接口调试的时间，复杂接口处理带来更大的延迟，同时需要解决功能验证同步的问题。

从图5(a)和表1对比中可以看出本文提出的功能验证方法验证时长最少，与DUT通信带宽最高，在98.6 MHz频率下达到25.2 Gb/s，相对于文献[9,10]中的验证技术，本文提出的功能验证方法更适用于高吞吐数据率的设计进行功能验证。在验证IP复用方面与软件仿真验证和改进型FPGA原型验证一样具有可复用性；从DUT的参考频率看，相对于其他3种技术，本文的方法参考频率最高，说明时序更容易收敛。

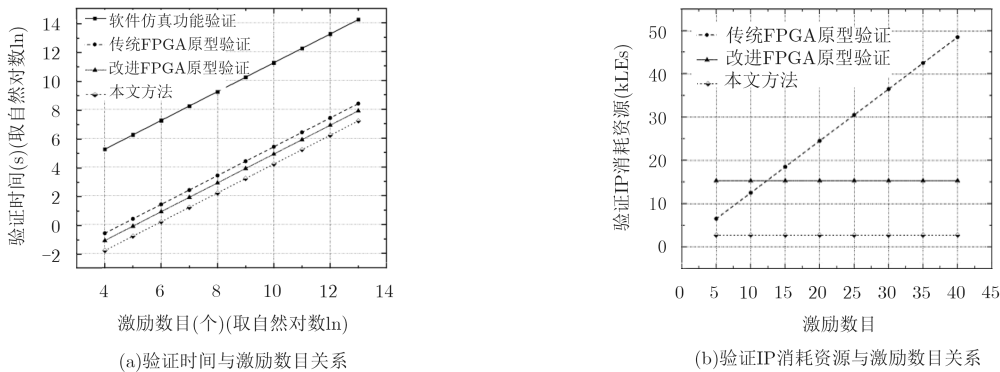


图 5 不同验证技术对比结果

表 1 本文提出的技术与其他文献的结果对比

| 采用技术                         | 功能覆盖率(%) | 高级语言 | 验证时长(s) | 通信带宽(Gb/s) | 验证IP复用 | 频率(MHz) |
|------------------------------|----------|------|---------|------------|--------|---------|
| 软件仿真验证 <sup>[4,5]</sup>      | 100      | 是    | 203572  | —          | 是      | —       |
| 传统FPGA原型验证 <sup>[7,8]</sup>  | —        | 否    | 594     | —          | 否      | 63.5    |
| 改进FPGA原型验证 <sup>[9,10]</sup> | 100      | 是    | 357     | 1.00       | 是      | 58.3    |
| 本文所提出的技术                     | 100      | 是    | 179     | 25.20      | 是      | 98.6    |

5 结论

针对数字逻辑电路功能验证速度和灵活性不能统一的问题，本文提出FPGA硬核处理器系统加速

数字电路功能验证的方法。在FPGA原型验证中引入硬核处理器系统加速数字逻辑电路的功能验证，使用SoC FPGA的逻辑资源对DUT进行物理实现，SoC FPGA的硬核处理器系统作为激励发生单

元通过高速协议桥将激励注入DUT中,之后对功能覆盖率进行统计和分析进而完成逻辑电路的功能验证。本文还对验证IP的复用性进行了分析,协议桥将验证系统的软件层和硬件层进行分离提高了验证IP的复用性。实验结果表明本文所提出的方法能够有效地缩短数字逻辑电路的功能验证周期,验证IP可重用,适用于对数据吞吐率高、规模较大的数字逻辑电路的功能验证。

## 参考文献

- [1] WANG Yifan, JOERES S, WUNDERLICH R, *et al.* Modeling approaches for functional verification of RF-SoCs: Limits and future requirements[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2009, 28(5): 769–773. doi: 10.1109/TCAD.2009.2014533.
- [2] MARKOVIC D, CHANG Chen, RICHARDS R, *et al.* ASIC design and verification in an FPGA environment[C]. Proceedings of 2007 IEEE Custom Integrated Circuits Conference, San Jose, USA, 2007: 737–740. doi: 10.1109/CICC.2007.4405836.
- [3] STOTLAND I, SHPAGILEV D, and STARIKOVSKAYA N. UVM based approaches to functional verification of communication controllers of microprocessor systems[C]. Proceedings of 2016 IEEE East-West Design & Test Symposium, Yerevan, Armenia, 2016: 1–4. doi: 10.1109/EWDTS.2016.7807695.
- [4] HU Zhaohui, PIERRES A, HU Shiqing, *et al.* Practical and efficient SOC verification flow by reusing IP testcase and testbench[C]. Proceedings of 2012 International SoC Design Conference, Jeju Island, South Korea, 2012: 175–178. doi: 10.1109/ISOC.2012.6407068.
- [5] KIM M, KONG J, SUH T, *et al.* Latch-based FPGA emulation method for design verification: Case study with microprocessor[J]. *Electronics Letters*, 2011, 47(9): 532–533. doi: 10.1049/el.2011.0462.
- [6] 施仨, 韩赛飞, 黄新明, 等. 面向全同态加密的有限域FFT算法 FPGA设计[J]. 电子与信息学报, 2018, 40(1): 57–62. doi: 10.11999/JEIT170312.  
SHI Quan, HAN Saifei, HUANG Xinming, *et al.* Design of finite field FFT for fully homomorphic encryption based on FPGA[J]. *Journal of Electronics & Information Technology*, 2018, 40(1): 57–62. doi: 10.11999/JEIT170312.
- [7] LI Tiejun, ZHANG Jianmin, and LI Sikun. An FPGA-based random functional verification method for cache[C]. Proceedings of the 2013 IEEE 8th International Conference on Networking, Architecture and Storage, Xi'an, China, 2013: 277–281. doi: 10.1109/NAS.2013.44.
- [8] GSCHWIND M, SALAPURA V, and MAURER D. FPGA prototyping of a RISC processor core for embedded applications[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2001, 9(2): 241–250. doi: 10.1109/92.924027.
- [9] PODIVINSKY J, CEKAN O, LOJDA J, *et al.* Functional verification based platform for evaluating fault tolerance properties[J]. *Microprocessors and Microsystems*, 2017, 52: 145–159. doi: 10.1016/j.micpro.2017.06.004.
- [10] BARNASCONI M, DIETRICH M, EINWICH K, *et al.* UVM-systemC-AMS framework for system-level verification and validation of automotive use cases[J]. *IEEE Design & Test*, 2015, 32(6): 76–86. doi: 10.1109/MDAT.2015.2427260.
- [11] IEEE. 1800.2–2017 IEEE standard for universal verification methodology language reference manual[S]. IEEE, 2017. doi: 10.1109/IEEESTD.2017.7932212.
- [12] CHEN Fulong and SUN Yunxiang. FPGA-based elastic in-circuit debugging for complex digital logic design[J]. *International Journal of Autonomous and Adaptive Communications Systems*, 2017, 10(3): 303–319. doi: 10.1504/IJAACS.2017.10007621.
- [13] Intel FPGA. Cyclone V hard processor system technical reference manual[EB/OL]. <https://www.altera.com/documentation/sfo1410143707420.html>, 2018: 6.
- [14] Xilinx Inc. Zynq-7000 all programmable SoC data sheet: Overview[EB/OL]. <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html#documentation>, 2018: 6.
- [15] DUARTE-SÁNCHEZ J E, VELASCO-MEDINA J, and MORENO P A. Hardware accelerator for the multifractal analysis of DNA sequences[J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018, 15(5): 1611–1624. doi: 10.1109/TCBB.2017.2731339.
- [16] ISKANDER Y, PATTERSON C, and CRAVEN S. High-level abstractions and modular debugging for FPGA design validation[J]. *ACM Transactions on Reconfigurable Technology and Systems*, 2014, 7(1): 1–22. doi: 10.1145/2567662.
- [17] SCHAFER B C. Source code error detection in High-level synthesis functional verification[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016, 24(1): 301–312. doi: 10.1109/TVLSI.2015.2397036.

刘小强: 男, 1990年生, 博士生, 研究方向为SoC芯片设计、图像识别。

袁国顺: 男, 1966年生, 博士生导师、研究员, 研究方向为嵌入式MCU设计、高性能模数混合电路。

乔树山: 男, 1980年生, 硕士生导师、研究员, 研究方向为低功耗集成电路设计方法学、智能感知节点SoC、无线通信、电力线载波通信、数字化射频接收机、低功耗处理器。