

# 基于 FPGA 的 PIE 编码与 UVM 验证平台的设计\*

李姝萱, 卜 刚

(南京航空航天大学 电子信息工程学院, 江苏 南京 211106)

**摘 要:** 国际标准 ISO/IEC18000-6 规定脉冲间隔编码(PIE)作为 RFID 数字基带系统中阅读器发送链路的编码方式。采用 Verilog 语言对该模块进行设计, 用 Quartus II 软件综合并下载到 FPGA 开发板上, 并使用 SignalTap II 逻辑分析仪对信号进行采集和分析。此外, 在设计的基础上添加了 UART 收发模块, 实现 PC 和 FPGA 板的通信。为了对 PIE 编码进行充分验证, 基于 UVM 验证方法学和直接编程接口 C(DPI-C), 设计并实现了一种高效且可复用的验证平台, 驱动器和监测器分别实现向 DUT 发送激励及收集输出结果的功能。参考模型与 DUT 的输出结果在记分板中对比一致, 功能覆盖率达到 100%, 提高了验证效率及完备性。

**关键词:** 脉冲间隔编码; 串口; 通用验证方法学; 直接编程接口

中图分类号: TN402

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.201022

中文引用格式: 李姝萱, 卜刚. 基于 FPGA 的 PIE 编码与 UVM 验证平台的设计[J]. 电子技术应用, 2021, 47(6): 110-114.

英文引用格式: Li Shuxuan, Bu Gang. Design of PIE coding based on FPGA and UVM verification platform[J]. Application of Electronic Technique, 2021, 47(6): 110-114.

## Design of PIE coding based on FPGA and UVM verification platform

Li Shuxuan, Bu Gang

(School of Electronic Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

**Abstract:** The international standard ISO/IEC18000-6 stipulates pulse interval encoding(PIE) as the encoding method of the reader transmission link in the RFID digital baseband system. It uses Verilog language to design the module, synthesizes with Quartus II software and downloads it to FPGA development board, then use SignalTap II logic analyzer to collect and analyzethe signal. In addition, the UART transceiver module is added on the basis of the design to realize the communication between PC and FPGA board. In order to fully verify the PIE code, based on UVM verification methodology and direct programminginterface C(DPI-C), an efficient and reusable verification platform was designed and implemented. The driver and the monitor were implemented to send excitation and the function of collecting output results. The output results of the reference model and DUT are consistent in the scoreboard, and the function coverage reaches 100%, which improves efficiency and completeness of the verification.

**Key words:** PIE coding; UART; universal verification methodology; directed program interface

### 0 引言

随着集成电路的快速发展, 芯片研究周期不断缩短, 传统的基于 testbench 的验证方式由于效率低、难达到预期覆盖率以及可移植性差等缺点, 不适合日渐复杂的 SOC 芯片开发。目前, 通用验证方法学(Universal Verification Methodology, UVM)已成为 IC 验证领域最为广泛使用的验证方式。UVM 兼具封装、继承、面向对象等特点, 拥有大量功能全面的组件和基类, 同时开发了 factory、config 等机制, 可根据工程特性灵活地搭建验证结构。

RFID 是非接触式的无线通信技术, 它通过对射频信号进行调制、解调来实现信息的传输, 是当今最有发展前景的技术之一。目前, RFID 技术已经广泛应用在众

多行业和领域, 如物流运输、资源管理、军事国防、智能交通、门禁考勤、医疗电子等领域<sup>[1-2]</sup>。本文针对无线射频识别(Radio Identification, RFID)数字基带处理单元中阅读器发送链路编码模块进行设计, 并进行了仿真和板级验证, 采用串口通信作为物理通道, 实现了 PC 端与 FPGA 板之间实现互传数据。为进一步验证编码功能, 搭建了 UVM 验证平台, 采用 DPI 接口调用 C 语言编写的参考模型, 实现了待测设计和 C 模型在记分板中的输出结果对比一致。

### 1 PIE 编码

图 1 为 RFID 阅读器发送链路结构图。阅读器产生命令信息, 经编码、调制后发送至标签。标签对信息进行解调、译码, 从而识别阅读器命令, 做出相应反馈。

编码在信息传输中起到防干扰的作用, 使得信息能

\* 基金项目: 江苏省自然科学基金(BK2012792)

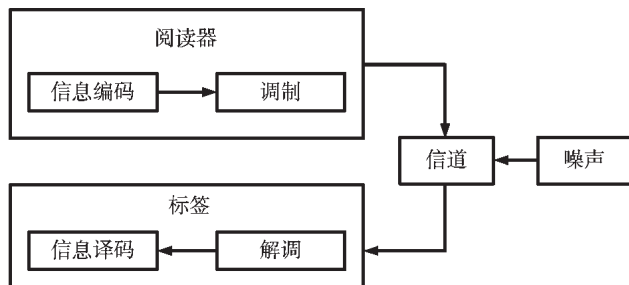


图1 RFID阅读器发送链路结构

够高效、安全地到达收方。ISO/IEC18000-6 协议规定,基带处理单元中阅读器至标签链路的数据编码采用 PIE 编码方式,即脉冲宽度编码。编码原理是通过对数据-0 和数据-1 进行不同码元长度的编码来实现,主要体现在脉冲下降沿前所持续的高电平时间不同。PIE 的编码规则如图 2 所示<sup>[3]</sup>。

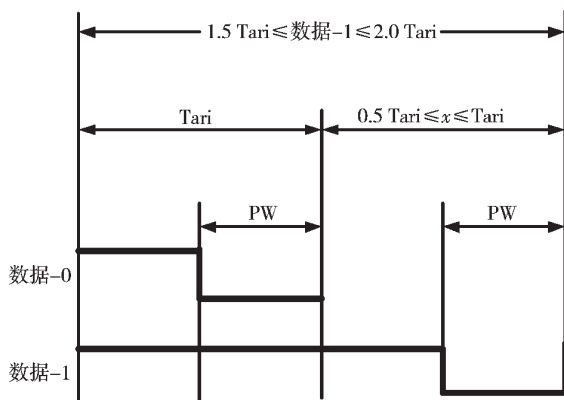


图2 PIE 编码规则

其中,  $T_{ari}$  表示的是阅读器向标签发送信息的基准时间间隔,具体取值需要根据实际要求作出一定调整,在实际情况下,需要参考地方无线电规则。 $T_{ari}$  的详细取值如表 1 所示。低电平部分长度 PW 的取值范围在 0

表 1 最佳阅读器对标签  $T_{ari}$  值

$T_{ari}$ 值/ $\mu s$	6.25	12.5	25
$T_{ari}$ 值公差/%	$\pm 1$	$\pm 1$	$\pm 1$

到  $T_{ari}$  之间<sup>[3]</sup>。

本文中  $T_{ari}$  取值为  $12.5 \mu s$ , 低电平部分长度 PW 取为基准时间间隔的一半,  $x$  取值为 1, 则数据-1 编码后长度为  $2T_{ari}$ , 数据-0 编码后长度为  $T_{ari}$ , 即数据-1 长度是数据-0 的两倍。因此, 数据-0 在经过两个时钟周期后的输出“10”序列; 数据-1 在经过四个时钟周期后输出“1110”序列 0。

经过 Modelsim 仿真后, 在工程中添加了一个伪随机数发生器模块, 其功能是产生 8 位伪随机数作为 PIE 编码的输入数据。工程经 Quartus II 综合、编译并下载到 FPGA 开发板中, 观察并分析实验结果。本文所用的板子型号是 Altera 的 DE2-115。

图 3 描述了使用 SignalTap II 逻辑分析仪抓到的输入和输出波形。如输入 8 位十六进制数据 8Ch, 对应的二进制数据为 1000\_1100, 按照 PIE 编码规则编码后的数据应为 11\_1010\_1010\_1110\_1110\_1010, 高位补 0 后得到对应十六进制数据为 3AAEEAh。经查验, PIE 编码设计能够正常工作。

在设计基础上增加串口收发模块, 本设计中串口发送模块实现了多字节发送。具体实现了 PC 端发送一个 1 字节数据给下载了由 PIE 编码模块和串口收发模块共同组成的工程的 FPGA 开发板, 并通过串口通道收到经过 PIE 编码后的 4 字节数据。图 4 是串口助手发送和接收数据。

## 2 UVM 验证结构

UVM 是一个以 SystemVerilog 类库为主体搭建验证平台的验证方法学, 验证人员利用其可重用组件构建具有标准化层次结构和接口的功能验证环境。UVM 由 A-ccellera 标准组织推出, 并得到三大厂商 (Cadence、

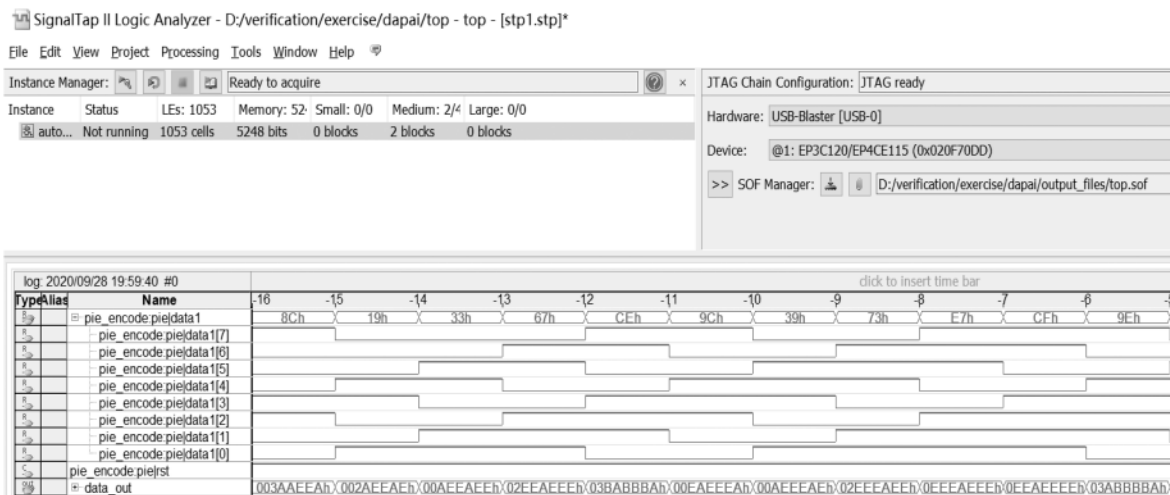


图3 PIE 编码 SignalTap II 波形图



图 4 串口收发数据图

Synopsys 和 Mentor)的一致支持。2011 年以后陆续推出了 1.0、1.1、1.1a、1.1b、1.1c 和 1.1d 版本。2014 年推出了最新 UVM 1.2 版本<sup>[4]</sup>,UVM 作为如今验证领域最广泛使用的验证方法学,基本代表 IC 验证的大致发展方向。

UVM 平台主要由 uvm\_componet 和 uvm\_object 组成。uvm\_componet 继承于 uvm\_object 的同时,拥有 uvm\_object 没有的一些特性。与传统的 Testbench 相比而言,UVM 验证平台实现了受约束的随机激励的产生,自动化的结果比对以及代码和功能覆盖率的收集<sup>[5]</sup>。本文采用如图 5 所示的 UVM 验证平台结构<sup>[6]</sup>,下面将对各验证组件进行详细介绍。

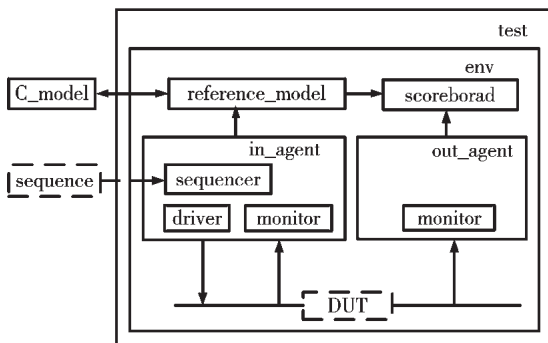


图 5 UVM 验证平台结构

(1)interface:声明 DUT 接口。

(2)test:派生于 uvm\_test,是整个验证平台的入口,例化了 env,是 case 的父类,在不同的 case 中设置了 sequencer 的 default\_sequence。

(3)transaction:继承于 uvm\_sequence\_item,在验证平台中,信息在各组件中不以比特的形式流动,用 transaction 来处理与表征数据<sup>[7]</sup>。定义两个 transaction 类型,分别为 transaction\_in、transaction\_out。前者为输入端数据包,后者

为输出端数据包。driver 首先需要把数据从 transaction\_in 中提出来,然后将其转换为 DUT 的管脚能够接收的数据。通常来说,一个 driver 只能接收一种 transaction,所以在 my\_driver 被定义成了一个参数化的类,这个参数就是 my\_driver 所能接收的 transaction 的类型。transaction\_out 在 monitor 内部的转化机制同理。

(4)sequence:继承于 uvm\_sequence,产生数据包,并通过 sequencer 传递给 driver,driver 则将 transaction 中定义的数据信息配置给 DUT 管脚。验证过程中需要对 DUT 施加大量不同的激励,就要使 sequence 产生不同的激励。

(5)sequencer:继承于 uvm\_sequencer,将 sequence 发送给 driver。

(6)driver:继承于 uvm\_driver,向 sequencer 请求 transaction,经过数据转化后通过下传给 DUT<sup>[9]</sup>。driver 请求 transaction 的部分代码如下:

```
seq_item_port.get_next_item(req);
send(req);
seq_item_port.item_done();
```

其中,seq\_item\_port 是用于连接 driver 和 sequence 的一个端口,driver 如果想要发送数据就要从这个端口中获得;sequencer 如果有数据要交给 driver,也要通过这个端口送给 driver。从这个端口申请数据要调用这个端口的 get\_next\_item 方法。当数据驱动完毕时,要通过调用 item\_done 来告知这个端口。

(7)monitor:继承于 uvm\_monitor,创建 monitor\_before 和 monitor\_after 两个类,分别用来获取 driver 发送给 DUT 的数据和 DUT 的输出结果<sup>[8]</sup>。

(8)agent:继承于 uvm\_agent,在 build\_phase 函数中,通过 is\_active 值来例化内部不同的组件。is\_active 值以 UVM\_PASSIVE 方式运转的 agent 例化 sequencer、driver、monitor\_before,其作用是驱动总线,也可以监测总线;以 UVM\_ACTIVE 方式运行的 agent 只例化 monitor\_after,监测总线而不驱动总线。

(9)reference\_model:继承于 uvm\_componet,使用 uvm\_blocking\_get\_port 端口从 monitor\_before 中获取数据,调用 DPI-C,获取经 C\_model 运算过的输出结果,通过 uvm\_analysis\_port 端口将数据发送给 scoreboard 组件<sup>[9-10]</sup>。

```
tr2.data_out=pie_data(tr1.data);
tr2.length=pie_length(tr1.data);
ap.write(tr2);
```

其中,pie\_data 和 pie\_length 分别为计算 PIE 编码结果和长度的 C 函数,tr1.data 是 monitor\_before 发送给参考模型的数据。tr2 是 reference\_model 发送给 scoreboard 的数据包。

(10)Scoreboard:继承于 uvm\_scoreboard,main\_phase 中使用 fork 开启了两个进程,其中一个进程用于从 reference\_model 中获得数据,另外一个进程用于从 monitor 中获得数据。一般情况下,同样的两组激励,经过 DUT 的

处理后会有一定的延迟,但是在 reference\_model 中不会出现延迟。因此 reference\_model 中的数据总是比 DUT 先到达 scoreboard,所以在 scoreboard 中可进行如下操作:将 reference\_model 发送来的数据先放入一个队列中等待。当接收到 DUT 的输出后,从队列中依次弹出数据,调用 compare 函数和 DUT 的输出比较。

(11)env:继承于 uvm\_env,例化了 reference\_model、scoreboard 和 agent 组件,并在其 connect\_phase 中使用 FIFO 实现端口之间的相互连接。例如,要实现 reference\_model 和 scoreboard 的通信,只要在 env 中声明一个 FIFO,然后用 FIFO 将 reference\_model 的 analysis\_port 和 scoreboard 的 get\_port 进行互相连接即可。连接部分代码如下<sup>[11]</sup>。

```
agt.i_ap.connect(agt.mdl_fifo.analysis_export);
mdl.port.connect(agt.mdl_fifo.blocking_get_export);
agt.o_ap.connect(agt.scb_fifo.analysis_export);
scb.from_dut.connect(agt.scb_fifo.blocking_get_export);
mdl.ap.connect(mdl.scb_fifo.analysis_export);
scb.from_rfm.connect(mdl.scb_fifo.blocking_get_export);
```

```
# UVM_INFO scoreboard.sv(26) @ 150: uvm_test_top.pie_env.scb [scoreboard get] tr1.data_out= 15445690
# UVM_INFO scoreboard.sv(27) @ 150: uvm_test_top.pie_env.scb [scoreboard get] tr1.length= 24
# UVM_INFO monitor_after.sv(44) @ 170: uvm_test_top.pie_env.agt.mon_apt [monitor_after get] tr_out.data_out= 15445690
# UVM_INFO monitor_after.sv(46) @ 170: uvm_test_top.pie_env.agt.mon_apt [monitor_after get] tr_out.length= 24
# UVM_INFO monitor_before.sv(49) @ 170: uvm_test_top.pie_env.agt.mon_bef [monitor_before send] tr.data=170
# UVM_INFO scoreboard.sv(46) @ 170: uvm_test_top.pie_env.scb [scoreboard] compare passed
# data_out=ebaeba
```

图6 打印信息面板

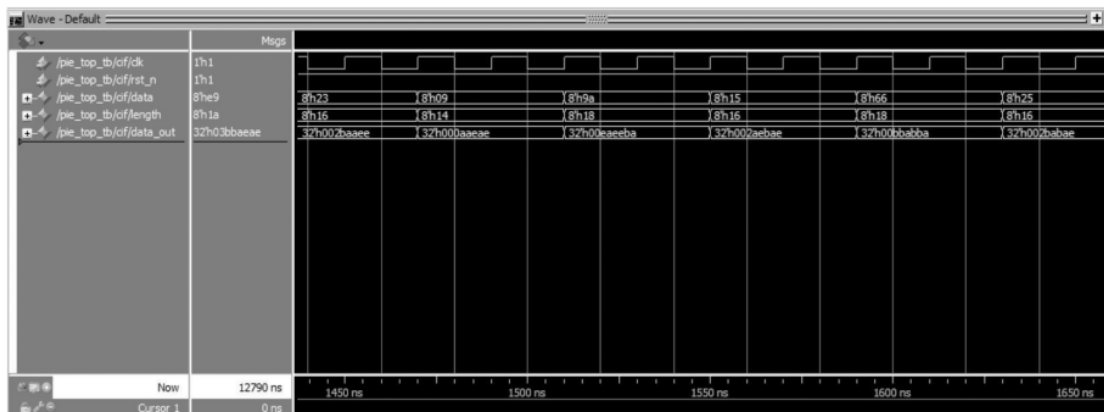


图7 DUT 仿真波形图

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances	Get_inst_coverage	Comment
/pie_top_tb_sv_unit/monitor_before									
TYPE pie_in_cg	monitor_bef...	100.0%	100	100.0%					auto(1)
CVP pie_in_cg::data_cov	monitor_bef...	100.0%	100	100.0%					
bin zero		3	1	100.0%					
bin allfs		20	1	100.0%					
bin special1		2	1	100.0%					
bin special2		20	1	100.0%					
bin positive		61	1	100.0%					
bin negative		339	1	100.0%					
/pie_top_tb_sv_unit/monitor_after									
TYPE pie_cg	monitor_after	100.0%	100	100.0%					auto(1)
CVP pie_cg::coverpoint_0#	monitor_after	100.0%	100	100.0%					
bin value		7	1	100.0%					
CVP pie_cg::coverpoint_1#	monitor_after	100.0%	100	100.0%					
bin len		382	1	100.0%					

图8 功能覆盖率

### 3 仿真结果

在验证平台的顶层(pie\_top\_tb)例化整个 PIE\_ENC-ODE 模块作为 DUT, 并利用 SystemVerilog 的 run\_test 函数作为验证平台的入口。在命令行输入 UVM\_TEST-NAME=casex(x 为所定义的 case 的序号), 创建并运行测试用例, 达到验证目的<sup>[12-13]</sup>。通过利用 SystemVerilog 的 interface 机制<sup>[14]</sup>定义一组虚拟接口(Virtual Interface,V-IF)来连接 UVM 动态验证环境以及基于 RTL 模型的 DUT。使用 uvm\_config\_db 机制使得虚拟接口能够连接两者。

图 6 是验证平台的打印信息面板。这里的 tr1 是 scoreboard 从 reference\_model 中获取的数据包, tr\_out 则是 monitor\_after 收集到的 DUT 端口信息所构成的数据包。由图可见, 参考模型和 DUT 输出结果在记分板中对比结果一致, 达到了验证效果。

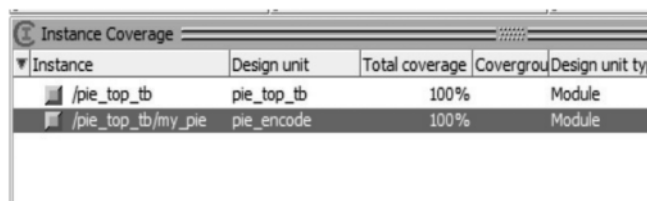
图 7 为由覆盖率驱动的且受约束的随机分层验证平台运行出的波形图。由此可见, 每个时钟周期内, PIE 编码输入对应的输出都是正确的。

图 8 显示了 DUT 的功能覆盖率, 验证平台对实例化



的接口输入和输出信号分别进行了采集,功能覆盖率达到 100%。对于传统的基于 testbench 的验证平台,缺少特定的方法对功能覆盖率进行统计,很难收集并分析功能覆盖率的具体情况。

图 9 描述的是 DUT 的代码覆盖率,其中,分支覆盖率、条件覆盖率、表达式覆盖率和语句覆盖率均达到了 100%,总代码覆盖率也达到 100%。



Instance	Design unit	Total coverage	Coverage	Design unit type
/pie_top_tb	pie_top_tb	100%		Module
/pie_top_tb/my_pie	pie_encode	100%		Module

图 9 代码覆盖率

#### 4 结论

UVM 验证方法学作为如今 IC 验证领域最广泛使用的验证方式,其拥有大量库及基类、面向对象等特点,有效提高了验证效率和产品可靠性。本文设计了 RFID 阅读器发送链路 PIE 编码模型,在建模、仿真的基础上进一步实现了 FPGA 原型验证,进而实现了 FPGA 串口多字节发送的功能以及和 PC 的通信。此外,基于 UVM 验证方法学及 SystemVerilog 语言特性,借助 DPI 调用高抽象层次 C 模型,搭建了具有高度复用性的功能全面的验证平台,在此基础上编写覆盖率驱动的受约束随机激励,实现了 SystemVerilog 和 C 语言协同仿真。最终使得代码的分支覆盖率、条件覆盖率、表达式覆盖率、语句覆盖率均达到 100%,总覆盖率达到 100%,功能覆盖率也达到 100%,达到了验证目的。

#### 参考文献

- [1] 崔璐.UHF RFID 阅读器的设计与实现[D].西安:西安电子科技大学,2014.
- [2] 苏渊.基于 UVM 的 AES 模块验证平台设计[D].西安:

西安电子科技大学,2018.

- [3] ISO 18000-6, Information technology-Radio Frequency Identification for Item management-Part 6:Parameters for air interface communications at 860MHz to 960MHz, 2006.
- [4] 张强.UVM 实战[M].北京:机械工业出版社,2014.
- [5] 吴迪飞.基于 UVM 的 FPGA 数字下变频模块级验证方法的实现[J].电脑知识与技术,2020,16(10):269-272.
- [6] 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.
- [7] 魏小义.基于 Transaction 模式的 SoC 软硬件协同验证方法研究[D].成都:电子科技大学,2006.
- [8] 徐飞,秦水介.基于 UVM 的基带射频接口电路的验证[J].电子技术应用,2018,44(3):11-14.
- [9] TASIRAN S, KEUTZER K. Coverage metrics for functional validation of hardware designs[J]. IEEE Design & Test of Computers, 2001, 18(4).
- [10] 李璐,周春良,冯曦,等.基于 DPI-C 接口的可扩展 SOC 验证平台[J].电子设计工程,2018,26(4):136-140.
- [11] 赵经天.基于 UVM 的 SAS 控制器模块验证[D].杭州:杭州电子科技大学,2019.
- [12] 郑昌立.基于 UVM 的仿真用例集约简技术研究[D].杭州:浙江大学,2020.
- [13] 赵赛,闫华,丛红艳.基于 UVM 验证方法学的数字交换芯片验证平台[J].电子与封装,2019,19(12):36-40.
- [14] 乔鹏丽,吕英杰.基于 SystemVerilog 的数字基带通信芯片的验证[J].南开大学学报(自然科学版),2020,53(6):1-4.

(收稿日期:2020-10-19)

#### 作者简介:

李姝萱(1997-),通信作者,女,硕士,主要研究方向:数字 IC 验证,E-mail:1026568754@qq.com。

卜刚(1969-),男,教授,主要研究方向:集成电路设计。



扫码下载电子文档

(上接第 109 页)

京航空航天大学,2019.

- [9] 徐鼎.基于虚拟仪器的电能采集分析回放装置的研究与设计[D].济南:山东大学,2018.
- [10] Allegro.The Datasheet of ACS714 Designer manual[Z].2006.
- [11] 张志明,李蓉艳,王磊.等精度频率信号采集测试系统[J].仪器仪表学报,2002(S1):159-160.
- [12] 胡伟.宽动态范围高精度程控增益放大器的研究与设计[D].广州:华南理工大学,2016.
- [13] 葛立,李骥,高枫,等.多路模拟开关在数据采集系统中的应用与分析[J].电子技术应用,2014,40(12):40-42.
- [14] 袁强,焦新泉,方炜,等.基于 FPGA 的多通道遥测采编

器设计[J].传感器与微系统,2015,34(12):97-100.

- [15] 车来晟,唐鹤,高昂,等.一款 10 位逐次逼近型模数转换器设计[J].电子与封装,2019,19(7):16-19,23.
- [16] 沈泓翔.一种过采样 20-bit SAR ADC 的研究与设计[D].成都:电子科技大学,2018.

(收稿日期:2020-10-07)

#### 作者简介:

李建翔(1995-),男,硕士研究生,主要研究方向:电路设计、动态测试。

杨玉华(1977-),女,硕士生导师,副教授,主要研究方向:微纳器件与系统、动态测试与存储、电路与系统。



扫码下载电子文档

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所