

基于 UVM 的 efuse 通信协议的设计与验证

郑梦瑶^{1,2,3,*}, 隋金雪^{1,3}, 张霞²

(1. 山东工商学院, 烟台 264005; 2. 北京大学 信息技术高等研究院, 北京 100080;
3. 中科网络技术研究所, 郑州 450018)

摘要: 在集成电路制造尤其是半导体存储器的开发设计过程中经常涉及到 efuse 的使用, 被应用于芯片保护、电路校准等多种场景。本文设计了一种自定义的 efuse 通信协议, 并基于 UVM 搭建了验证平台对其功能进行验证。验证结果表明, 使用该协议可以实现需求功能, 并具有良好的鲁棒性。

关键词: efuse; UVM; 验证平台

中图分类号: TP31 **文献标识码:** A

Design and verification of efuse communication protocol based on UVM

ZHENG Mengyao^{1,2,3}, SUI Jinxue^{1,3}, ZHANG Xia²

(1. Shandong Technology and Business University, Yantai 264005, China; 2. Advanced Institute of Information Technology, Peking University, Beijing 100080, China; 3. Institute of Network Technology, Zhengzhou 450018, China)

Abstract: In the development and design process of integrated circuit manufacturing, especially semiconductor memory, the use of efuse is often involved, which is applied in various scenarios, such as chip protection, circuit calibration and so on. In the paper, a custom efuse communication protocol is proposed and a verification platform based on UVM is built to verify its functions. The validation results show that the requirement functionality can be achieved with this protocol and has good robustness.

Keywords: efuse; UVM; verification platform

0 引言

随着信息化社会的发展, 芯片规模和应用领域不断增加。UVM 技术在硬件验证中的重要性也愈加明显, 它提供了一套丰富的工具, 可以有效验证集成电路和芯片设计的功能正确性^[1]。

efuse 作为集成电路芯片内部的可编程存储器元件, 被广泛应用于通信、汽车、消费电子等领域, 以满足各种功能需求^[2-3]。但目前在 efuse 的相关设计中, 还没有一个通用的标准通信协议, 芯片制造商通常根据自己的需求和架构来设计自定义的 efuse 通信协议。

利用 UVM 验证方法学搭建的验证平台可以对自定义的 efuse 通信协议在调试模式下的各种子模式进行功能验证, 确保数据正常的烧写、更新和读取等。

1 UVM 验证方法学

UVM (Universal Verification Methodology) 是基于 SystemVerilog 语言开发的一套验证框架和库的通用验证方法学^[4]。它提供了一种结构化的方法来设计和实现可重用的验证环境, 用于验证数字集成电路芯片的功能和正确性。一个完整的验证平台架构应该包含多个功能组件, 典型的 UVM 验证平台结构如图 1 所示^[5-6]。

在 UVM 的平台框架中, 每个组件都有明确的任务, 组件之间相互独立并同时通过继承、组合的方式相互交互。其中 sequence 序列用于产生 transaction 单独与 sequencer 通信, 由 sequencer 将事物传递给 driver 来驱动。monitor 监视 DUT 端口上的信号, 并收集数据后发送给 reference model 进行参考, 与 scoreboard 进行对比。

项目基金: 山东省自然科学基金资助项目(2016ZRB019JQ); 浙江省重点研发计划资助项目(2020C01SA100208)。

郑梦瑶(硕士研究生), 主要研究方向为数字集成电路; 隋金雪(副教授), 主要研究方向为控制理论和控制工程; 张霞(高级工程师), 主要研究方向为半导体微电子设计和微电子封装。

* 通信作者, E-mail: 2021420059@sdtbu.edu.cn。

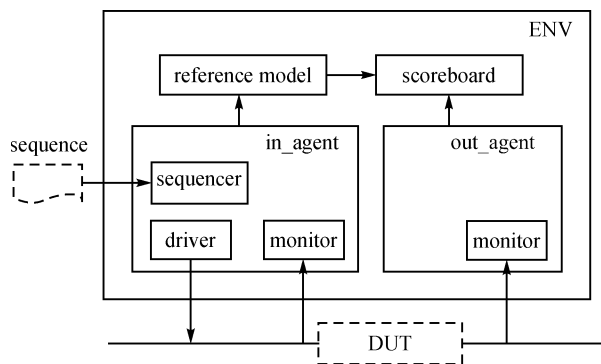


图 1 典型的 UVM 验证平台结构

Fig. 1 Typical UVM verification platform structure

sequencer、driver 和 monitor 封装在 agent 中,在 agent 之外的 ENV 相当于一个容器,在里面完成对平台组件的实例化、参数的定义等^[7]。

UVM 在芯片验证中可以提高验证效率和验证质量,降低成本,并促进行业内的合作与标准化,从而在半导体行业中得到广泛应用。

2 efuse 通信协议

2.1 efuse 模块简介

尽管 efuse 具有体积小、功耗低、可编程性强且不易擦除等优点,但仍存在一些限制和挑战。不同的芯片制造商可能会采用不同的通信接口、命令格式和数据传输方式,使得在不同的设备上使用 efuse 变得更加复杂^[8]。缺乏标准化导致缺少通用性,efuse 通信协议在不同的芯片之间可能存在差异,鲁棒性差,会增加芯片维护的工作量。

efuse 有正常工作和调试两种不同的操作模式。在正常工作模式下,efuse 处于只读状态,数据被读取用于设备的正常运行。调试模式旨在方便对 efuse 进行调试、测试和编程操作,数据可以被修改、编程或擦除^[9]。

2.2 efuse 通信协议设计

针对 efuse 调试过程中动态修改 efuse 的数据设计了一种自定义的单线通信协议,包括对调试模式的进入/退出,efuse 的烧写、更新、读出、替换等命令的规定,以验证设备的功能、配置和校准等,旨在提供一种 efuse 的标准通信协议,解决各种 efuse 通信协议开发效率低、鲁棒性差的问题。

设计中一次通信包括:命令起始字(1 位)+数据包(16 位)+命令停止字(1 位)。其特征内部时钟周期为 T 时,每位数据的周期为 9T,可以提高数据传输的可靠性和抗干扰性,通过更多的时钟相位和冗余数据来检测和纠正可能出现的错误。当周期为 9T 时,根据低电平和高电平的周期为 3T 和 6T 来判断输入的数据是 0 还是 1。

在低电平周期为 3T、高电平周期为 6T 的情况下,可以将其视为“0”的编码方式;相反,低电平 6T、高电平 3T 为“1”的编码方式。通过使用不同的低电平和高电平周期来表示 0 和 1,可以增加数据传输的可靠性。这样的设计可以提高对噪声或干扰的抵抗能力,从而提高数据的准确性。此外,周期的不对称性还可以提供更好的时钟同步和数据解码的可靠性,增加系统稳定性。

在一次通信中,高电平维持时间大于等于 9T,代表 1 位的命令起始字,然后判断高低电平持续时间,确定编码 0 或 1,发送完 16 位代表一个数据包。最后 1 位的命令停止字可以是 0 或 1,满足以低电平开始、高电平结束,总的周期为 9T 即可,用不同的 16 位编码代表不同命令。

2.3 命令编码与鲁棒性

在实际的 efuse 通信中,系统进入调试模式和退出调试模式,调试模式下对 8 组 64 位的 efuse 器件进行替换、烧写、更新、读取的操作,都是通过通信协议进行设置。其特征在于通过对 16 位数据包的高 8 位和低 8 位配置不同的数据确定 efuse 调试模式的状态。表 1 为通信协议的命令编码总表。

表 1 通信协议指令表

Table 1 Communication protocol instruction table

指令	命令起始字	数据包高位	数据包低位	命令结束字
调试模式进入		55	AA	
调试模式退出		55	55	
Fuse 字节替换		A0~A7	FUSE_reg [63:0]	
Fuse 字节读出	9T 高电平	A8~AF	55	0/1
Fuse 字节烧写		B0~B7	任意 8 位数据	
Fuse 字节更新		B8~BF	55	
Fuse 自动烧写		69	55	
Fuse 自动更新		69	AA	

在 efuse 通信协议中使用特定的字节序列(如 55AA、5555、6955 等)来代表不同的操作状态实现特定的配置功能,简化了通信的实现和解析,便于工程师的理解和开发。

在使用自定义的 efuse 通信协议时,使用内部时钟采样信号需要考虑内部时钟有正负 30% 的频率变化范围。3T 的信号经过采样后的周期最大为 $3/0.7=4$,最小为 $3/1.3=2$;6T 的信号经过采样后的周期最大为 $6/0.7=8$,最小为 $6/1.3=4$ 。因此,高电平和低电平计数按照调试模式进入方式进行阈值判断可以覆盖振荡器频率变异。相关的设计代码如下(内部时钟设为 clk_25k, hing_cnt 和 low_cnt 分别代表高低电平计数值):

```

always @ (posedge i_clk25k or negedge rst_s1)
begin
    if(! rst_s1)
    begin
        shift_reg <= 'd0;
    end
    else if(! shift_en && test_in_rise && (high_cnt >= 4'd8))
    begin
        shift_reg <= 'd0;
    end
    else if(shift_en && test_in_fall && (high_cnt >= 4'd2
    && high_cnt <= 4'd8 ) && (low_cnt >= 4'd0 && low_
    cnt <= 4'd4) && high_cnt > low_cnt)
    begin
        shift_reg <= (1'b0, shift_reg[15:1]);
    end
    else if(shift_en && test_in_fall && (high_cnt >= 4'd0
    && high_cnt <= 4'd4) && (low_cnt >= 4'd2 && low_
    cnt <= 4'd8) && high_cnt < low_cnt)
    begin
        shift_reg <= (1'b1, shift_reg[15:1]);
    end
    else
    begin
        shift_reg <= shift_reg;
    end
end
end

```

在通信协议中设计了允许 30% 的频率变化范围,一方面可以提高鲁棒性,另一方面不同的设备可能具有不同的时钟频率要求,允许一定的变化范围,可以使设备无需对时钟系统进行大幅度调整就能在不同的系统中工作,提高了设备的频率兼容性。

3 验证平台设计

3.1 验证平台顶层架构

本文针对 efuse 模块搭建了基于 UVM 的验证平台,验证平台的整体结构框图如图 2 所示。

整个验证平台主要由 5 部分构成, testcase 是平台的最顶层,可以包含一个或多个 sequence,在每个 sequence 中定义了一系列的操作或事件,用于生成特定的测试场景或激励序列。在传统验证平台中用单独的 scoreboard 进行期望值与实际值的对比,验证测试结果的正确性。结合具体模块特性,此验证平台使用嵌入在 testcase 下 sequence 中的 check 函数来替代 scoreboard 功能,主要优势包括灵活性强,可以根据具体的测试需求编写自定义的验证逻辑,针对特定情况进行快速验证并生成错误信息或失败警告,不需要编写复杂的 scoreboard 逻辑;能够实时反馈验证结果,每次执行 check 函数时,可以快速检测到错误,并立即生成相关的日志;代码量相对较少,易于编写和理解;使用新开发的 agent 封装了 sequencer 和 driver 用于驱动 transaction,在 interface 内对信号进行描述。

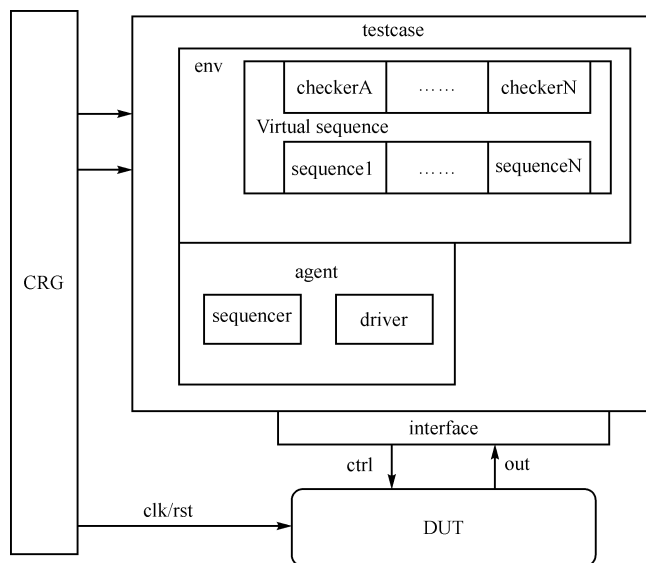


图 2 验证平台结构框图

Fig. 2 Verification platform structure block diagram

时钟复位在验证环境的顶层由宏产生,并连接至 testcase 和 DUT。在验证环境中根据 clk_mux 的值选择不同的时钟信号,代表不同内部时钟频率,并将其分配给 clk_dut 信号。只要将 clk_mux 分别置 0、1 和 2 或者随机,根据待验证 efuse 模块是否能正常工作即可验证在不同时钟频率下设计的鲁棒性。

3.2 测试用例设计

测试用例的设计主要围绕设计的 efuse 通信协议是否能完成预定的功能。搭建完验证平台后,首先要使用正常功能的冒烟用例保证验证平台的准确运行;然后就是调试模式下调试模式的进入、退出,efuse 字节的替换、读取、烧写和更新,efuse 字节的自动烧写和更新,还有对各种调试命令进行随机搭配的测试用例,确保在实际应用场景中的可靠性^[14-15];最后用一条用例来测试不同时钟频率下的自定义协议功能是否正常。表 2 中整理了测试用例名称以及与其对应的功能和执行情况。

表 2 测试用例设计表

Table 2 Test case design table

用例名	功能	执行情况
tc_sanity	正常功能冒烟用例	PASS
tc_dbg_inout	调试功能冒烟用例	PASS
tc_byte_replace	BYTE_REPLACE 命令测试用例	PASS
tc_byte_burn_update	BYTE_BURN+BYTE_UPDATE 命令测试	PASS
tc_byte_read	BYTE_READ 命令测试用例	PASS
tc_auto_burn_update	AUTO_BURN+AUTO_UPDATE 命令测试	PASS
tc_byte_cmd_rand	调试命令随机组合	PASS
tc_dbg_osc	片内晶振抖动对调试模式影响验证	PASS

4 仿真结果和覆盖率

4.1 用例仿真结果

通过 `i_test_in` 输入特定时序的序列(数据包为 55AA),则可以进入调试模式 `debug_model=1`。`i_test_in` 输入后经过两级内部时钟同步产生 `test_in_smp/2` 并产生对应的上升沿 `test_in_rise` 和下降沿 `test_in_fall` 信号。内部有 `high_cnt` 和 `low_cnt` 分别计数高电平和低电平维持的时钟周期数。判读 `high_cnt` 和 `low_cnt` 的值,以确认当前数据是 0 还是 1,依次完成数据包 16 位数据的移位。同理,若收到的数据包数据 `shift_reg=5555`,则退出调试模

式 `debug_model=0`。进入、退出调试模式仿真波形如图 3(a)和图 3(b)所示。

在 `tc_byte_replace` 用例中,测试了支持调试模式下“开头帧+0xA0~A7+待替换的值+结束帧”的命令,实现把待替换值写入对应 fuse 寄存器的功能;A0~A7 的替换顺序应支持正序(A0,A1,A2...A7)、逆序(A7,A6...A0)、乱序(随机)三种,应支持结束帧为 0 或为 1 时,都可以成功执行上述命令。在调试命令随机组合的测试用例 `tc_byte_cmd_rand` 中,只需要将前面列举的各种调试命令用例进行随机组合即可。各种调试命令的仿真结果在此不一一列举,对应的用仿真工具打印的日志报告如图 3(c)和图 3(d)所示。



图 3 用例仿真波形和报告图

Fig. 3 The case simulation waveform and report diagram

最后对协议的鲁棒性进行验证。在片内晶振抖动对调试模式影响验证用例 `tc_dbg_osc` 中,分别设置片内晶振偏快 25%和偏慢 25%。执行 `tc_dbg_inout` 用例的仿真波形如图 4 所示,在低电平 `low_cnt` 计数到 2 和 `high_cnt` 计数到 7 时,`shift_reg` 初始化数据为 0,证明在不稳定的时钟频率下此协议可以正常工作。

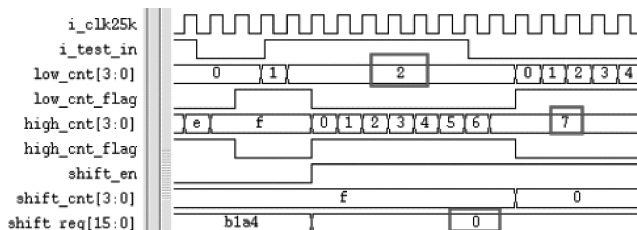


图 4 鲁棒性验证仿真波形

Fig. 4 Robustness verification simulation waveform

4.2 覆盖率分析

最后,在完成所有测试用例的调试后,对所有用例进行了回归测试。用例总数为 8 条,单条用例跑 100 次,得到的覆盖率报告如图 5 所示。每个输入点覆盖率在 99%以上,实现了预期覆盖率收敛目标^[16]。

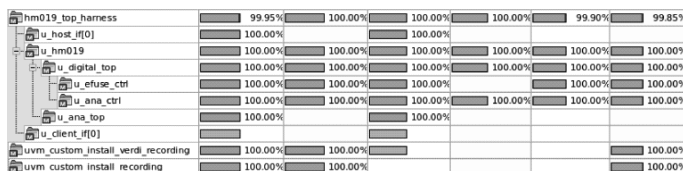


图 5 覆盖率统计报告

Fig. 5 Coverage statistics report

5 结语

在集成电路发展日益壮大和 efuse 芯片广泛应用的

当下,一个经过验证的具有更好鲁棒性的通信协议至关重要。本文设计的 efuse 通信协议具有配置灵活性、安全性和生产效率高等优势,并且通过验证确保了其可靠性,可以适应更广泛的应用和需求。

参考文献

- [1] 林伟峰,施隆照,魏陈鸿,等. ALU 模块的 UVM 验证[J]. 中国集成电路,2022,31(11):30-37.
LIN W F, SHI L ZH, WEI CH H, et al. UVM Verification of ALU Modules[J]. China Integrated Circuit Journal, 2022, 31(11):30-37(in Chinese).
- [2] 田敏志. 基于 28 nm 低功耗工艺的 2 Kbits eFuse IP 设计[D]. 杭州:浙江大学,2021.
TIAN M ZH. 2Kbits eFuse IP Design Based on 28nm Low Power Process[D]. Hangzhou: Zhejiang University, 2021(in Chinese).
- [3] 晏颖,曹玉升,张睿. eFuse 失效分析与可靠性电路设计[J]. 电子技术应用,2023,49(1):26-31.
YAN Y, CAO Y SH, ZHANG R. e Fuse Failure Analysis and Reliability Circuit Design[J]. Electronic Technology Applications, 2023, 49(1):26-31(in Chinese).
- [4] 张小鹏. 基于 UVM 的 IIC 总线设计与验证[D]. 西安:西安电子科技大学,2021.
ZHANG X P. Design and Verification of IIC Bus Based on UVM[D]. Xi'an: Xidian University, 2021(in Chinese).
- [5] 马彬,刘威. SoC 芯片上 AXI 总线 IP 验证[J]. 电子设计工程,2023,31(13):56-60.
MA B, LIU W. AXI Bus IP Verification on SOC Chips[J]. Electronic Design Engineering, 2023, 31(13):56-60(in Chinese).
- [6] 杜越,郑杰良,吴益然. 基于 UVM 的 SoC 系统级外设验证平台设计[J]. 中国集成电路,2022,31(6):37-43.
DU Y, ZHENG J L, WU Y R. Design of a SoC system level peripheral verification platform based on UVM[J]. China Integrated Circuit Journal, 2022, 31(6):37-43(in Chinese).
- [7] Nagesh K A, Shilpa D. Verification of SerDes design using UVM methodology[C]//In Proceeding of Fifth International Conference on Microelectronics, Computing and Communication Systems, 2021:607-616.
- [8] 高乘源. 高性能 FPGA 芯片的安全方案研究与设计[D]. 西安:西安电子科技大学,2022.
GAO CH Y. Research and design of security solutions for high-performance FPGA chips [D]. Xi'an: Xidian University, 2022(in Chinese).
- [9] 武建宏,左卫松. 基于 55 nm 工艺大容量 EFUSE 烧写特性研究[J]. 集成电路应用,2018,35(6):29-32.
WU J H, ZUO W S. Research on the Burning and Writing Characteristics of High Capacity EFUSE Based on the 55nm Process[J]. Integrated Circuit Applications, 2018, 35(6):29-32(in Chinese).
- [10] 相光超. 基于 UVM 的 AXI-APB 总线桥验证平台的设计与实现[D]. 烟台:山东工商学院,2023.
XIANG G CH. Design and Implementation of AXI-APB Bus Bridge Verification Platform Based on UVM [D]. Yantai: Shandong Technology and Business University, 2023(in Chinese).
- [11] 卢喆. 基于 UVM 的无线接收机验证平台设计与实现[D]. 南京:东南大学,2022.
LU ZH. Design and Implementation of a Wireless Receiver Verification Platform Based on UVM[D]. Nanjing: Southeast University, 2022(in Chinese).
- [12] 毛国栋. 基于 UVM 的 AHB-APB 桥模块验证[D]. 太原:中北大学,2023.
MAO G D. Verification of AHB-APB Bridge Module Based on UVM[D]. Taiyuan: North University of China, 2023(in Chinese).
- [13] 常琳琳,于哲,周舜民,等. 基于高性能安全存储芯片的 SATA 通路验证[J]. 计算机系统应用,2023,32(5):338-343.
CHANG L L, YU ZH, ZHOU SH M, et al. SATA Path Verification Based on High Performance Secure Storage Chips[J]. Computer System Applications, 2023, 32(5):338-343(in Chinese).
- [14] 张银婷. QDR SRAM 存储器的设计与验证[D]. 西安:西安电子科技大学,2020.
ZHANG Y T. Design and Verification of QDR SRAM Memory[D]. Xi'an: Xidian University, 2020(in Chinese).
- [15] 江青骏. 基于 FPGA 的 I2C 和 SSPI 总线配置系统的设计与实现[D]. 西安:西安电子科技大学,2020.
JIANG Q J. Design and Implementation of I2C and SSPI Bus Configuration System Based on FPGA[D]. Xi'an: Xidian University, 2020(in Chinese).
- [16] 郑志亮,程雯,王先兰. 基于 RISC-V 处理器执行单元的高效灵活验证平台设计[J]. 电子设计工程,2023,31(7):124-131.
ZHENG ZH L, CHENG W, WANG X L. Design of an Efficient and Flexible Verification Platform Based on RISC-V Processor Execution Unit[J]. Electronic Design Engineering, 2023, 31(7):124-131(in Chinese).

(责任编辑:薛士然 收稿日期:2023-09-04)