

基于 FPGA 的 DDR SDRAM 测试平台设计

谢树平, 毛源豪

(湖南艾科诺维科技有限公司, 长沙 410000)

摘要: DDR SDRAM 是 FPGA 板卡中的重要组成部分, 其可靠性与带宽决定了设备能否正常工作; 为了测试 DDR SDRAM 的性能是否符合预期, 开发了一种基于 FPGA 的 DDR SDRAM 测试平台, 平台包含一个基于 DDR SDRAM 控制器的测试器 IP 核, 具有数据校验、带宽测量的功能; 编写了控制测试器 IP 核的 TCL 脚本, 用于配置测试参数、控制测试流程与读取测试结果; 在 Python 语言下使用 PyQt5 开发库设计了图形界面程序, 能够根据用户操作生成并执行对应的 TCL 脚本; 最终实现了一个操作简单、测试流程可配置、自动输出测试结果的 DDR SDRAM 测试平台; 测试结果表明, 测试平台能够正确地进行 DDR SDRAM 测试并输出统计结果; 对比 MIG 的示例工程, 测试平台额外增加了带宽测试、结果统计、循环测试等功能, 且使用的 FPGA 资源下降了 30%, 测试用时缩短了 70% 以上。

关键词: DDR SDRAM; FPGA; TCL 脚本; 测试平台; PyQt5

Design of DDR SDRAM Test Platform Based on FPGA

XIE Shuping, MAO Yuanhao

(Hunan Acqnova Technology Co., Ltd., Changsha 410000, China)

Abstract: DDR SDRAM is an important component of FPGA boards, and its reliability and bandwidth determine whether the device can function properly. To verify that the performance of the DDR SDRAM meets expectations, a DDR SDRAM test platform based on FPGA is developed. The platform incorporates a tester IP core, which is based on a DDR SDRAM controller and has the capability to verify data and measure bandwidth. TCL scripts are used to control the tester IP core, including setting test parameters, managing test processes, and retrieving test results. Additionally, a graphical interface program is designed using the PyQt5 development library in Python, which generates and executes corresponding TCL scripts based on user input. As a result, a DDR SDRAM testing platform is implemented that is both user-friendly and flexible in terms of the testing process, while also providing automatic results output. Test results demonstrate that the platform accurately tests DDR SDRAM and outputs statistical results. Moreover, compared to the example design of MIG, the testing platform includes bandwidth testing, result statistics, and automatic control functions, ultimately reducing FPGA resource usage by 30% and cutting test time by more than 70%.

Keywords: DDR SDRAM; FPGA; TCL script; test platform; PyQt5

0 引言

现场可编程逻辑门阵列 (FPGA, field programmable gate array) 是一种半定制芯片, 因其独特的电路可编程特性区别于其他器件。随着近年 FPGA 芯片的快速发展, 其内部的资源数量与丰富度显著提升^[1]。FPGA 在通信、图像处理^[2]、深度学习^[3]、人工智能^[4]与数据中心^[5]等领域发挥着重要的作用。双倍数据传输速率同步动态随机存储器 (DDR SDRAM, double data rate synchronous dynamic random access memory) 是 FPGA 板卡中最常用的高速缓存解决方案。DDR SDRAM 的工作情况直接影响到整个设备能否正常运作, 所以 DDR SDRAM 测试也是 FPGA 板卡测试中最重要的测试项目之一。此前对 DDR SDRAM 的测试主要通过 DDR SDRAM 控制器的调试端口进行。然而这种测试方案存在操作人员学习门槛高、测试流程不灵活、无法自动生成测试报告等缺点。操作人员在测试前需要充分了

解调试端口中约 200 个信号的定义、各控制信号间的逻辑关系以及操作 DDR SDRAM 的相关知识。此类方案在生成比特流后测试内容相对固定, 可在线配置的测试参数有限, 无法设置测试队列并自动完成。且没有生成测试报告的功能, 相关数据需要操作人员手动记录。降低了工作效率, 增加了出错风险。

在 FPGA 与 DDR SDRAM 的数据交互设计中, 设计者通常使用 FPGA 厂家提供的 DDR SDRAM 控制器。比如 Xilinx 公司为旗下的 FPGA 提供了成熟的存储器接口生成器 (MIG, memory interface generator)。MIG 内包含控制器和物理层, 设计者可以通过用户接口或先进可扩展接口 (AXI, advanced eXtensible interface) 访问 DDR SDRAM。文献 [6] 提出了一种基于 MIG 用户接口的应用, 在 MIG 用户接口的基础上设计了先入先出 (FIFO, first in first out) 接口, 使其既拥有 FIFO 缓存简单易用的特点, 又有 DDR SDRAM 存储空间大的优势。文献 [7] 在数据 FIFO

收稿日期: 2023-03-06; 修回日期: 2023-04-12。

作者简介: 谢树平 (1986-), 男, 硕士, 工程师。

引用格式: 谢树平, 毛源豪. 基于 FPGA 的 DDR SDRAM 测试平台设计[J]. 计算机测量与控制, 2023, 31(10): 67-75.

接口的基础上增设了一个命令/地址 FIFO, 实现了随机读写功能。

在 DDR SDRAM 控制器的测试技术中, 通常使用的方案是向 DDR SDRAM 写入特定的 pattern, 然后读取数据并校验数据一致性。文献 [8] 提出了一种包含内部自测 (BIST, built-in self-test) 模块的 DDR SDRAM 控制器。该模块将来自随机数发生器的数据写入 DDR SDRAM 中, 并将读出的数据与写入的数据在数据比较器中进行对比, 从而实现自监测的功能。文献 [9] 提出了一种基于在线逻辑分析仪 (ILA, integrated logic analyzer) 知识产权 (IP, intellectual property) 核验证 DDR SDRAM 控制器的方法。使用 AXI Trafficgen IP 核产生 AXI 总线的读写事务, 并用 ILA 抓取 AXI 总线上的信号, 分析其时序与数据正确性。

实际应用中, 通常使用标准 JTAG 接口实现 PC 上位机与 FPGA 之间的通信。文献 [10] 设计了一种基于 FPGA 的 DDR SDRAM 通用测试电路。该设计使用 JTAG 接口启动测试电路, 电路自动完成整个测试过程并将测试结果从 JTAG 接口传回 PC 上位机。文献 [11] 提出了一种使用工具命令语言 (TCL, tool command language) 脚本语言测试数字电路的方法。该方法通过在 Vivado 软件中执行 TCL 脚本实现自动对调试核虚拟输入输出 (VIO, virtual input output) 的修改与读取, 从而达到高效率的测试。文献 [12] 提出了一种基于 LabVIEW 软件调用 TCL 脚本的测试方法, 将 Vivado 软件的启动与执行 TCL 脚本的命令封装为批处理文件, 实现了自动化测试与生成报表。文献 [13] 提出了一种 FPGA 的 AXI 总线接口调试方法, 通过使用 JTAG to AXI IP 核, 允许用户通过执行 TCL 脚本来对 AXI 总线上的从设备进行访问。

为了解决当前 FPGA 板卡中 DDR SDRAM 测试工作复杂及效率低下的问题, 本设计做了如下工作:

1) 设计了 DDR SDRAM 控制器的测试器 IP 核 MIG Tester。该 IP 核具有生成读写事务、校验数据、记录数据读写用时等功能。同时 MIG Tester 支持对数据位宽、pattern 类型、起始地址、测试地址范围、突发长度等参数的在线配置, 具有高度的灵活性。

2) 设计了 TCL 脚本。用于控制 VIO、ILA、JTAG to AXI 等 IP 核, 从而实现对整个测试流程的控制以及测试结果信息的读取。

3) 设计了上位机图形界面程序。将测试流程中各个阶段需要执行的 TCL 脚本封装。通过文本框打印测试信息, 同时自动生成测试报告, 便于操作人员使用。

1 测试平台架构及原理

测试平台架构主要由两个部分构成: FPGA 板卡与 PC 上位机。FPGA 板卡内包含 FPGA 与 DDR SDRAM, 二者通过板卡上的 DDR SDRAM 接口相连接。PC 上位机与 FPGA 通过 JTAG 接口实现通讯。用户在 PC 上位机操作图形界面程序, 完成测试流程与输出测试结果^[14-15]。所述架构如图 1 所示。

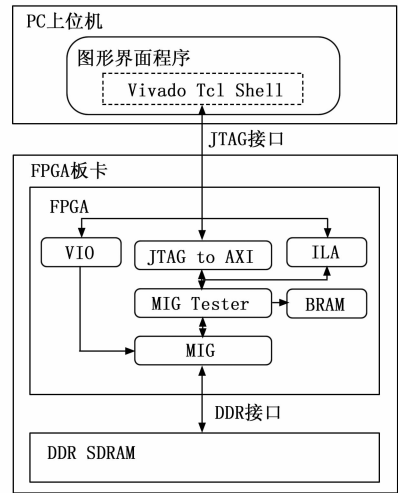


图 1 测试平台架构框图

DDR SDRAM 是待测的目标器件, 由 FPGA 中的 MIG 控制。在对 DDR SDRAM 的测试中, 重点关注的是 DDR SDRAM 的读写正确性与带宽两项性能^[16]。

FPGA 与 DDR SDRAM 通过 DDR SDRAM 接口相连接, FPGA 内例化的 MIG 控制 DDR SDRAM 的初始化与数据读写。MIG Tester 可以通过 MIG 的 AXI slave 接口对 DDR SDRAM 进行数据读写。MIG Tester 的功能是测试读写正确性与带宽。MIG Tester 有两个主要接口: 一个是 AXI slave 接口, 作为控制接口与 JTAG to AXI 相连, TCL 脚本控制 JTAG to AXI 来与 MIG Tester 进行通信, 控制测试流程与读回测试结果; 另一个是 AXI master 接口, 作为数据读写接口与 MIG 相连, 测试过程中对 MIG 的数据传输均由此接口实现。此外 MIG Tester 还有一个块随机存取存储器 (BRAM, block random access memory) 主接口, 用于向 BRAM 写入错误数据对应的地址信息, 便于后续对错误数据的溯源分析。FPGA 内例化的 VIO、JTAG to AXI 和 ILA 调试核互相配合下起到控制 MIG 初始化、控制 MIG Tester、抓取 AXI 总线信号的作用, 这些调试核均受 PC 上位机控制^[17-18]。

PC 上位机通过 JTAG 接口与 FPGA 进行通信。通过在 PC 上位机的 VivadoTcl Shell 程序中执行 TCL 脚本, 可以对 FPGA 中调试核的信息读取与控制。图形界面程序将 VivadoTcl Shell 作为子进程运行, 并将测试相关的 TCL 脚本封装为对应的按钮。操作人员可以通过图形界面程序便捷地控制整个测试流程, 提高了测试效率, 降低了学习门槛^[19-20]。

2 FPGA 工程设计

测试板卡使用的 FPGA 产自 Xilinx 公司, 型号为 xc7k325tffg900-2, 使用的 DDR SDRAM 型号为 MT8JTF2-5664HZ-1G6M1。

2.1 FPGA 工程主要模块

工程主要包含 JTAG to AXI、MIG Tester、AXI Smart-

Connect、MIG、AXI Bram Controller 等 IP。模块之间主要通过 AXI 总线进行连接, 所有 AXI 总线通过 AXI Smart-Connect 互联 IP 连接在一起。地址空间分配中, MIG 基地址被设为 0x80000000, 可被 MIG Tester 和 JTAG to AXI 访问。MIG Tester 基地址设为 0x44A00000, AXI BRAM Controller 基地址设为 0x00000000, 两者都只能被 JTAG to AXI 访问。BRAM 的类型为真双口 BRAM, 其中一个接口受 AXI BRAM Controller 控制, 另一个接口供 MIG Tester 的 BRAM 主接口写入数据^[21-22]。

2.2 Memory Interface Generator

MIG 是由 Xilinx 官方提供的存储器接口成熟的解决方案, 是可靠的 DDR SDRAM 控制器。MIG 由一个控制器与 DDR 物理层接口组成。MIG 将复杂的 DDR 物理接口封装为被广泛使用的 AXI 接口, 简化了用户的设计。在 FPGA 工程设计中, MIG 的 AXI slave 接口与 MIG Tester 的 AXI master 接口相连接, 在测试过程中完成数据传输。MIG 的复位信号控制着 MIG 初始化的开始, 由 VIO 驱动。MIG 的初始化完成信号是高电平时表示 MIG 完成了初始化, 由 VIO 监听。

根据 DDR SDRAM 型号与具体电路, 对 MIG 进行了如下配置: 使能 AXI 接口, 并设置数据位宽为 512, 以适配 MIG Tester 的 AXI 主接口; 工作时钟频率设为 800 MHz, 参考时钟输入设为 200 MHz, 以适配板卡上的参考时钟晶振。

2.3 Memory Interface Generator Tester

MIG Tester 是自主设计的 MIG 专用测试 IP 核, 主要包含以下模块: AXI slave 接口、控制器、写数据生成器、AXI master 接口、数据校验器、预期数据生成器、错误计数器、计时器与 BRAM 主接口。MIG Tester 的框架如图 2 所示。

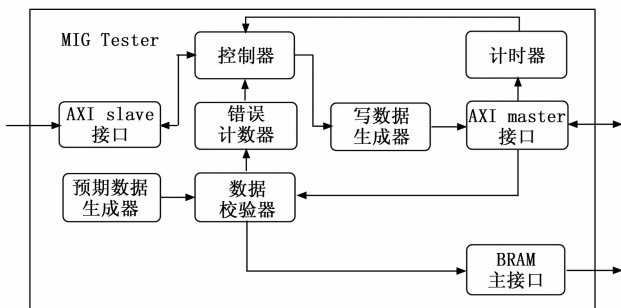


图 2 MIG Tester 框架图

AXI slave 接口是控制接口, 通过 AXI slave 接口可以读写控制器中寄存器的值, 各寄存器与地址的映射关系与详细描述如表 1 所示。

AXI slave 接口的寄存器位宽均为 32 位。其中偏移为 0x0~0x18 的寄存器是可读可写的, 主要包括可配置的测试参数与测试控制。偏移为 0x20~0x34 的寄存器是只读的, 主要包括测试结果以及固定测试参数。

表 1 AXI slave 接口寄存器地址映射关系

寄存器名	偏移	描述
BaseAddress	0x0	测试起始地址
BurstLength	0x4	读写突发长度
BurstNumber	0x8	单次测试突发数量
TestControl	0xC	启动测试与清除中断
TestStatus	0x10	MIG Tester 状态
PatternSwitch	0x14	固定 Pattern 功能开关
Pattern	0x18	Pattern 设定
Reserve	0x1C	保留
WriteCycle	0x20	写数据所用时钟周期数
ReadCycle	0x24	读数据所用时钟周期数
ErrorNumber	0x28	读写数据校验错误数
Reserve	0x2C	保留
DataWidth	0x30	MIG 数据位宽
Frequency	0x34	计时时钟频率

写数据生成器在控制器的控制下, 生成所设地址范围与所设 pattern 的写数据事务。DDR SDRAM 测试的地址范围由测试起始地址、读写突发长度、单次测试突发数量与 MIG 数据位宽 4 个参数共同确定。公式如下:

$$S_{\text{test}} = \frac{W_{\text{data}} \times L_{\text{burst}} \times N_{\text{burst}}}{8} \quad (1)$$

$$A_{\text{high}} = A_{\text{base}} + S_{\text{test}} \quad (2)$$

式中, S_{test} 是测试地址范围大小, A_{high} 是测试终止地址, A_{base} 是测试起始地址, W_{data} 是数据位宽, L_{burst} 是突发长度, N_{burst} 是突发数量。测试中写入 DDR SDRAM 的数据默认是 8bit 长的递增数 pattern, 当固定 pattern 开关被使能后, DDR SDRAM 写入的 pattern 将和 0x18 地址寄存器内的数据保持一致。

AXI master 接口是数据传输接口, 与 MIG 的 AXI slave 接口相连接。写数据生成器生成的写数据事务和数据校验器生成的读数据事务均由此接口完成具体的数据传输。

数据校验器用于校验读回的数据与预期数据的一致性。当两者数据不相符时, 错误标志信号 error_flag 置“1”, 使错误计数器计数加一, 并将错误数据对应的地址写入 BRAM。

错误计数器接收数据校验器的错误标志信号进行计数。在完成测试后把错误计数写入 ErrorNumber 寄存器。每次开始测试时错误计数器清零。

计时器用于计算 AXI 总线读写数据所用时钟周期。在完成测试过程后计时器会将读写数据用时分别写入 ReadCycle 寄存器与 WriteCycle 寄存器。根据这两项测试结果以及计时时钟频率得出读写数据用时以及平均带宽。公式如下:

$$T_{\text{write}} = \frac{C_{\text{write}}}{F_{\text{timer}}} \quad (3)$$

$$B_{\text{write}} = \frac{S_{\text{test}}}{T_{\text{write}}} \quad (4)$$

式中, T_{write} 是写数据实际所用时长, C_{write} 是从寄存器中读取的写数据所用时钟周期数, F_{timer} 是计时器频率, B_{write} 是写数

据平均带宽。读数据所用时间和平均带宽的计算方式类似,不再赘述。

图 3 为 MIG Tester 控制器的状态机示意图。当 TestControl 寄存器的启动位被写入“1”时,处于空闲状态下的控制器进入写数据状态。AXI master 接口开始对指定地址范围写入所设 pattern。在完成指定地址范围的数据写入后,控制器进入读数据与数据校验状态。AXI master 接口读取指定地址范围的数据并校验。在完成指定范围数据的读取与比较后,进入测试完成状态,测试结果被写入 AXI slave 接口的测试结果寄存器中,同时完成中断信号置“1”。测试结果写入完成后,控制器回到空闲状态。

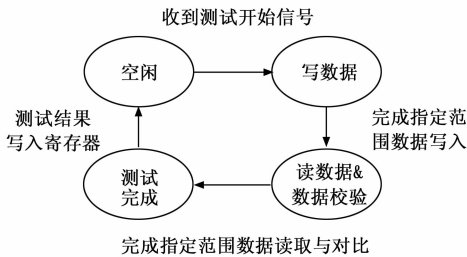


图 3 控制器状态机

2.4 Block Memory Generator

Block Memory Generator 可以使用 FPGA 内的 BRAM 资源生成性能优秀的存储器,这类存储器一般就被称为 BRAM。在设计中 BRAM 被设置为真双口模式,用于存储读写校验发生错误的错误数据对应的地址信息。

2.5 AXI BRAM Controller

AXI BRAM Controller 提供了可供访问本地 BRAM 的 AXI slave 接口,支持突发事务,性能优秀。在设计中 AXI BRAM Controller 控制真双口 BRAM 的其中一个端口。JTAG to AXI 可以通过此 IP 读取 BRAM 中的数据。

2.6 Virtual Input Output

VIO 用于在线监测或驱动 FPGA 内部的信号。本设计中主要使用 VIO 驱动 MIG 的复位信号,以及监测 MIG 的初始化完成信号。VIO 由 PC 上位机通过 JTAG 接口控制。

2.7 AXI to JTAG

AXI to JTAG 用于生成 AXI 接口事务。本设计中主要使用 AXI to JTAG 控制 MIG Tester,并读回测试结果。AXI to JTAG 由 PC 上位机通过 JTAG 接口控制。

2.8 Integrated Logic Analyzer

ILA 用于捕获 FPGA 内部信号。本设计中主要使用 ILA 捕获读写数据校验错误时 MIG 的 AXI 接口的数据,并通过 TCL 命令将数据保存至 PC 上位机。ILA 由 PC 上位机通过 JTAG 接口控制。

2.9 AXI SmartConnect

AXI SmartConnect 是由 Xilinx 官方提供的 AXI 总线互联 IP,可以将一个或多个 AXI 内存映射主设备连接到一个或多个 AXI 内存映射从设备。本设计中主要使用 AXI SmartConnect 将各 AXI 设备互联。

3 PC 上位机软件设计

PC 上位机软件的设计主要包括 TCL 脚本设计与图形界面程序设计两个部分。

3.1 TCL 脚本设计

TCL 作为一门具有简明高效、便于移植的脚本语言,应用于众多 EDA 软件。Vivado 软件在 TCL 语言的基础上增添了许多命令,以帮助开发者提高生产力。在本设计中,通过 TCL 脚本编程控制 FPGA 中的 3 个调试 IP,配合完成初始化测试与读写测试两种测试功能。

3.1.1 VIO 控制脚本

通过在 Vivado 软件中执行 TCL 脚本,PC 上位机可以控制 VIO 的驱动值或读取 VIO 探针的电平。需要修改 VIO 输出探针驱动值时,使用 `set_property` 命令修改对应探针的 `OUTPUT_VALUE` 值,之后使用 `commit_hw_vio` 命令执行驱动值的修改。需要读取 VIO 输入探针电平时,使用 `refresh_hw_vio` 命令刷新 VIO 输入探针的电平,然后使用 `get_property` 命令获取对应探针的 `INPUT_VALUE` 数值。

3.1.2 AXI to JTAG 控制脚本

PC 上位机可以通过执行 TCL 脚本来控制 AXI to JTAG 发起 AXI 总线读写事务,从而访问目标设备。当需要对 AXI 设备进行读操作时,使用 `create_hw_axi_txn` 命令创建读数据事务,执行 `run_hw_axi` 命令运行该事务。随后使用 `report_hw_axi_txn` 命令报告读取数据。在完成传输数据事务后,使用 `delete_hw_axi_txn` 命令删除该事务。写 AXI 设备数据的过程与读数据过程类似,区别在于生成写事务时需要指定事务类型为写,并增加一个写入数据作为参数。

3.1.3 ILA 控制脚本

TCL 脚本可以控制 ILA 执行信号抓取任务,并将抓取的数据展示或保存下来。需要通过 ILA 抓取特定信号时,首先使用 `set_property` 命令修改需要抓取的信号以及触发方式,然后使用 `run_hw_ila` 命令开始执行抓取信号。抓取到信号后,使用 `upload_hw_ila_data` 命令更新数据,并用 `write_hw_ila_data` 命令将抓取到的信号数据保存为 `ila` 文件和 `csv` 文件,便于后续进行分析。

3.1.4 初始化测试脚本

初始化测试的原理是,对 MIG 重复进行复位与初始化,记录该过程中 MIG 是否能在限定时间内完成初始化。

TCL 脚本编程的 DDR SDRAM 初始化测试流程如图 4 所示。在开始 DDR SDRAM 初始化测试后,首先控制 VIO 驱动 MIG 复位信号 `mig_rst_n` 为 0,以确保 MIG 处于复位状态。之后控制 VIO 驱动 MIG 复位信号 `mig_rst_n` 为 1, MIG 开始一次初始化过程。之后 VIO 监测的 `init_complete` 信号的值,如果在指定时间内检测到高电平,则视为初始化成功,反之则视为初始化失败。完成测试结果记录后,判断已测试次数是否已达到指定的测试次数,如果已满足则输出测试结果,反之则返回第一步继续进行测试。

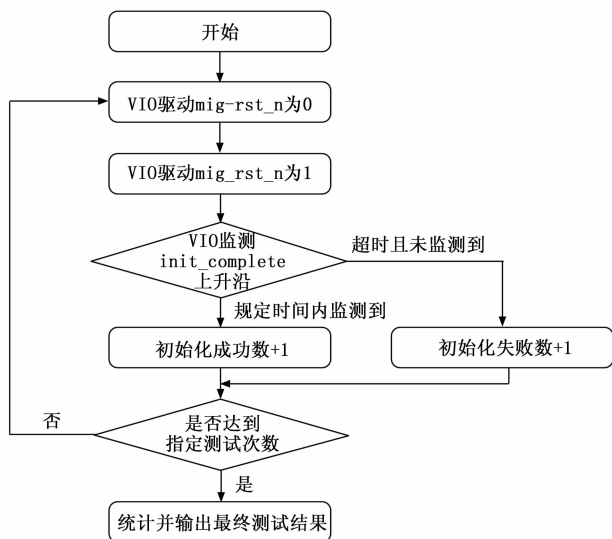


图 4 初始化测试流程图

3.1.5 读写测试脚本

读写测试的原理是, 将测试参数写入 MIG Tester, 控制 MIG Tester 开始读写测试, 并将每次测试结果进行记录与统计。

TCL 脚本控制的读写测试流程如图 5 所示。首先控制 JTAG to AXI 向 MIG Tester 写入读写测试的起止地址、写操作的 pattern 与读写的突发数等测试参数。随后设置 ILA 的触发信号为错误标志信号 error_flag 信号的高电平。写 TestControl 寄存器启动 MIG Tester。通过 VIO 监控测试完成信号 intr_done 直到它变为高电平。至此一次测试完成, 由 JTAG to AXI 读出测试结果并保存。随后判断测试次数是否已达到目标, 若已达到则统计所有测试结果并输出, 反之则再次启动测试。

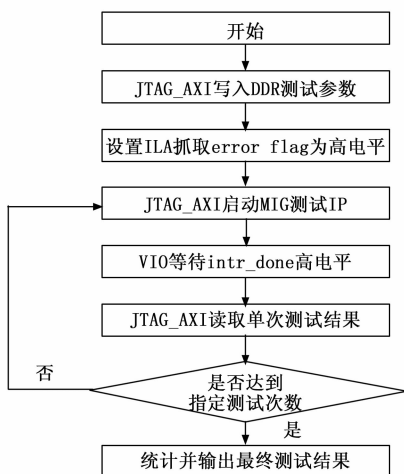


图 5 读写测试流程图

3.2 图形界面程序设计

图形界面程序在 PyQt5 框架下进行开发。PyQt5 衍生自 C++ 的 QT 库, 具有能够跨平台兼容多种环境, 功能强

大, 开源免费等优势。图形界面程序通过创建子进程的方式运行命令行程序, 并在命令行程序中打开 TCL 模式的 Vivado。点击程序中的功能按钮即可执行对应的 Vivado 指令或者 TCL 脚本。图形界面程序的界面如图 6 所示, 主要功能包括:

- 1) 人机交互: 操作人员可以根据测试需求, 在图形界面程序中调整测试参数、执行测试任务和读取测试信息。
- 2) 器件控制: 程序可以通过 JTAG 接口完成连接 FPGA 器件、状态检测、下载比特流、控制调试核等操作。
- 3) 统计结果: 程序自动重复并统计测试结果, 在测试循环完成后输出测试统计报告并保存为日志文件。



图 6 应用程序界面

程序界面主要分为 4 个区域。区域 1 主要包括连接器件以及下载比特流文件功能。区域 2 包括初始化测试的参数设置与执行。区域 3 包括读写测试的参数设置与执行。区域 4 是一个文本框, 输出文本信息。

首先需要在区域 1 中完成器件的连接与比特流文件的下载。程序启动后将自动运行 Vivado 子进程, 并在文本框处提示“Vivado 启动中”。待 Vivado 启动完成, 文本框处打印“启动完成”信息, “连接器件”按钮变为可用。点击“连接器件”, Vivado 进程将打开硬件管理器, 并尝试连接 FPGA 器件。如果未检测到 FPGA 器件, 则打印“连接失败”信息, 如果检测到 FPGA 器件, 则在“当前器件”标签处显示检测到的器件型号、连接状态与编程状态。“器件型号”下拉选项和“比特流选择”下拉选项会自动筛选出符合当前器件的选项。用户需要在这一步自行选择适用的比特流文件并点击“下载比特流”按钮。等待文本框提示“比特流下载完成”, 即可在区域 2 或区域 3 中执行后续测试。

入后增加突发传输所占的地址大小, 实现对测试地址范围内所有地址的写入。当设置的 pattern 是递增数时, 写数据 wdata 每写入一次数据后递增 1。写突发结束计数器 wlast_counter 在 wlast 信号置位时计数加一, 用于控制数据的突发传输。write_timer 信号是写状态计时器, 从测试进入写 pattern 过程开始计时, 直到进入读数据状态停止计时。

在 AXI 总线的读通道中, 读地址 araddr 在每次握手写入后增加突发传输所占的地址大小, 实现对测试地址范围内所有地址的读取。读数据 rdata 是读回的数据。当设置的 pattern 是递增数, 预期数据 expected_data 每写入一次数据后发生递增。错误标志信号 error_flag 的特征是在预期数据与读取数据 rdata 不同时置 1, 由于读取过程中预期数据与读取数据相符, 所以错误标志保持为 0。读突发结束计数器 rlast_counter 在每次 rlast 信号置位时计数加一, 用于控制状态机的转换。read_timer 是读状态计时器, 从状态机进入读数据状态开始计时, 直到进入测试完成状态停止计时。

然后人为插入错误数据验证数据校验模块功能是否正常。插入错误数据的方式如下: 选择 pattern 为递增数, 并且当写入数据为 A5 时, 将其替换为 5A。在读数据时, 读回数据 rdata 为 5A, 与其对应的预期数据 A5 不符合, 所以错误标志信号 error_flag 置“1”, 且错误计数器加一。证明数据校验功能正常。

经过上述功能验证, 可以确认测试 IP 核 MIG Tester 逻辑功能符合预期, 可以将其用于 DDR SDRAM 的测试。

4.2 FPGA 资源使用对比

FPGA 测试平台工程设计完毕后, 在软件 Vivado2020.2 默认设置下执行生成比特文件。作为对比, 打开 MIG 的示例工程并生成比特文件。MIG 的示例工程常作为 DDR SDRAM 的测试调试工程使用。对比测试平台工程与示例工程的资源使用数量, 结果如表 2 所示。测试平台工程资源使用量对比示例工程显著减少。其中作为 FPGA 主要资源的查找表 (LUT, look-up-table) 的使用减少了 33.22%, 触发器 (FF, Flip Flop) 的使用减少了 38.40%。

表 2 FPGA 资源使用数量对比

FPGA 资源名	测试平台工程	示例工程
LUT	17 415	26 081
LUTRAM	3 926	4 210
FF	20 371	33 070
BRAM	39	48.5

4.3 FPGA 板卡测试

测试板卡使用的 FPGA 产自 Xilinx 公司, 型号为 xc7k325tffg900-2, 使用的 DDR SDRAM 型号为 MT8JTF2-5664HZ-1G6M1。

4.3.1 测试流程

DDR SDRAM 测试基本流程如下:

1) 设计 FPGA 工程, 生成比特流文件与 ltx 文件 (调

试核信息文件), 并拷贝到图形界面程序目录下的 bitstream 文件夹中。

2) 使用仿真器将 FPGA 板卡连接至 PC 上位机并上电。打开图形界面程序, 待启动完成后点击“连接器件”, 等待连接完成, 标签栏处出现 FPGA 器件名称与状态。

3) 选择与 FPGA 对应的比特流, 点击“下载比特流”, 等待比特流下载完成, 标签栏处显示“已连接, 已编程”。

4) 在文本框中设置测试参数, 根据测试需求执行相应的测试。在测试完成后, 将在程序目录下生成测试结果报告。

4.3.2 初始化测试

根据赛灵思官方手册提供的数据, 设置初始化时间限制为 1 000 ms。执行 100 次初始化测试。初始化测试结果如表 3 所示。在 100 次初始化测试中, DDR SDRAM 的初始化均在 1 000 ms 以内完成。结果说明, 测试板卡中的 DDR SDRAM 能够正常初始化。

表 3 初始化测试结果

总初始化次数	超时次数	成功次数
100	0	100

4.3.3 读写测试

设置读写测试起始地址为 0, 结束地址为 0x7FFFFF, 即测试整个 2 GB 的地址空间。设置 AXI 总线突发数为 64, 测试 Pattern 栏目中勾选递增数 00、FF、55、AA, 每个 pattern 测试 100 次。数据读写测试测试结果如表 4 所示。写平均带宽和读平均带宽分别达到了理论带宽的 83.9% 和 88.1%, 且没有出现读写数据校验错误。证明了在测试环境下 DDR SDRAM 读写数据功能正常, 带宽性能良好。

表 4 数据读写测试结果 GB/s

Pattern	写平均带宽	读平均带宽	理论带宽	校验错误数
递增数	9.997	10.502	11.921	0
00	9.997	10.502	11.921	0
FF	9.997	10.502	11.921	0
55	9.997	10.502	11.921	0
AA	9.997	10.502	11.921	0
平均值	9.997	10.502	11.921	0

4.3.4 测试时长对比

表 5 是两类测试的测试平均时长对比。对于初始化测试, 示例工程的测试方法如下: 操作人员使用 Vivado 软件的硬件管理器控制 VIO 输出的复位信号失效, 之后通过 VIO 观察 MIG 输出的初始化完成信号是否置“1”, 据此判断 MIG 是否完成了初始化。在此过程中, 操作人员需要不断重复上述过程。一方面存在操作人员操作或记录失误的可能性; 另一方面手动操作效率低下, 且无法准确判断初始化所用时长。对于读写测试, 选用测试 00、FF、55、AA 与递增数物种 pattern 的情况进行对比。示例工程的测试方

法如下：操作人员使用 Vivado 软件的硬件管理器控制 VIO 完成 DDR SDRAM 的初始化、设置读写地址与数据。同时设置 ILA 抓取数据校验错误信号为“1”时的波形。观察 VIO 中读写数量计数器，据此判断是否完成了整个地址空间的读写。

表 5 测试时长对比 sec s

测试项目	测试平台	示例工程
初始化测试 100 次	64	247
读写测试 100 次	283	1 791

由测试结果可见，在 DDR SDRAM 功能正常的情况下，使用测试平台测试 DDR SDRAM 功能所用时长要比使用示例工程缩短 70% 以上。

在读写测试期间发生了数据错误的情况下，测试平台的优势将进一步体现。测试平台会将完整的错误数据地址信息写入 BRAM 中。在单次测试结束后根据地址信息读回对应错误数据，生成完整的错误数据列表，对测试时长的影响也不明显。而使用示例工程测试时，只能通过 ILA 抓取错误数据发生时的波形。这种方式无法暂停数据读写测试，所以无法保证抓取的错误数据完整性。另一方面 ILA 需要操作人员手动抓取波形，增加了测试时长。

4.3.5 测试结论

相比以往的 DDR SDRAM 测试方案，本测试平台具有如下优势：

1) 测试参数可配置。操作人员可以根据不同的测试需求设置相应的测试参数而无需重新生成比特流，节省准备时间。

2) 自动执行测试流程并输出报告。程序将按照参数设置自动循环执行测试任务，并在测试任务完成后输出相应的测试日志与报告文件。在测试任务繁重时可以提高测试效率。

3) 辅助错误分析。当数据读写出现错误时，程序将记录错误数据对应的地址信息，在当次测试结束后从对应地址读回错误数据并记录于测试日志中，给错误分析与设备调试提供便利。

4) 使用 FPGA 资源降低。对于实验所用 FPGA 板卡，对比 MIG 的示例调试工程，本设计使用的查找表资源由 26 081 降低至 17 415，所用触发器资源由 33 070 降低至 20 371，这些主要资源的使用下降了 30% 以上。在长时间测试中 FPGA 器件的功耗更低，散热压力更小。

5) 操作简单，学习门槛低。经图形界面程序封装之后，操作人员无需耗费精力学习其相关软硬件知识，仅需通过简单的设置参数与点击按钮即可完成测试过程。经过实践，使用测试平台的测试用时缩短了 70% 以上。

5 结束语

提出了一种基于 FPGA 的灵活度高、操作简易的 DDR SDRAM 测试平台。测试平台由 FPGA 板卡与 PC 上位机两

个主要部分组成。经验证，测试平台能够正确测试 DDR SDRAM 的初始化过程、读写正确性与带宽等性能并自动生成测试报告。该测试平台规避了传统测试方法操作繁琐、耗时长，学习成本高等弊端。但在测试前仍需要设计相应的 FPGA 工程并生成比特流。如果在 FPGA 板卡设计之初能够复用 FPGA 中 DDR SDRAM 的管脚分配，即可避免重新生成比特流，该方案的便捷性将进一步提升。

测试平台仅对 DDR SDRAM 进行了测试，但该架构同样适用于其他可通过 AXI 总线访问的外部存储设备的测试，具有拓展应用范围的潜力。

参考文献：

- [1] 杨海钢, 孙嘉斌, 王 慰. FPGA 器件设计技术发展综述 [J]. 电子与信息学报, 2010, 32 (3): 714-727.
- [2] 韩玉鑫, 王晓凯, 陆金旺. 实时彩色图像自适应中值滤波算法的 FPGA 实现 [J]. 计算机测量与控制, 2022, 30 (7): 173-180.
- [3] MITTAL S. A survey of FPGA-based accelerators for convolutional neural networks [J]. Neural Computing & Applications, 2020, 32 (4): 1109-1139.
- [4] LI Z J, ZHANG Y F, WANG J, et al. A survey of FPGA design for AI era [J]. 半导体学报 (英文版), 2020, 41 (2): 15-20.
- [5] 闵 捷. FPGA 与数据中心融合模式的思考 [J]. 电信快报, 2021 (3): 36-39, 43.
- [6] 吴长瑞, 徐建清, 蒋景红. 基于 Kintex-7 FPGA 的 DDR3 SDRAM 接口应用研究 [J]. 现代电子技术, 2017, 40 (24): 21-24, 27.
- [7] 高群福, 陈 星, 程 越. 基于 FPGA 的 DDR SDRAM 控制器设计与实现 [J]. 电子测量技术, 2011, 34 (8): 56-59.
- [8] 殷 晔, 李丽斯, 常 路, 等. 基于 FPGA 的 DDR3 存储控制的设计与验证 [J]. 计算机测量与控制, 2015, 23 (3): 969-971.
- [9] 蒋南峰. 使用 Xilinx 的 Spartan-6FPGA 作 DDR 芯片测试 [J]. 电子产品世界, 2011, 18 (9): 67-68.
- [10] 田 勇, 孙晓凌. 一种 DDR SDRAM 通用测试电路的设计与实现 [J]. 计算机测量与控制, 2010, 18 (8): 1727-1729.
- [11] 中国电子科技集团公司第二十研究所. 一种基于 TCL 语言的数字电路快速测试方法 [P]. 中国: 201911262773, 2020-05-08.
- [12] 中国电子科技集团公司第五十八研究所. 一种基于 labview 调用 vivado-tcl 脚本自动化测试方法 [P]. 中国: 202110763875, 2021-10-08.
- [13] 浪潮 (北京) 电子信息产业有限公司. FPGA 的 AMBA 总线接口调试方法、装置及 FPGA [P]. 中国: 202010745210, 2020-10-30.
- [14] 王 梦, 蒋 峰, 谢浩瀚. DDR2SDRAM 控制器接口的 FPGA 设计及实现 [J]. 计算机测量与控制, 2016, 24 (12): 119-121.
- [15] 宗 凯. 基于 FPGA 的 DDR3 控制器设计 [J]. 电子测量技术, 2017, 40 (1): 118-122.

- [16] 黄姣英, 赵如豪, 王 琪, 等. 基于 FPGA 的 DDR3 SDRAM 控制器设计 [J]. 现代电子技术, 2022, 45 (22): 68-74.
- [17] 刘文彬, 朱名日, 郑丹平, 等. 基于 FPGA 的大容量数据高速采集系统的设计 [J]. 计算机测量与控制, 2014, 22 (11): 3751-3753.
- [18] 兰功盾, 赵占锋, 魏高峰, 等. 基于 PCIe2. 0 的 Camera Link 接口相机模拟系统设计 [J]. 计算机测量与控制, 2017, 25 (11): 150-153, 158.
- [19] 何振琦. 基于 FPGA 的 LVDS 高速数据采集系统设计 [J]. 自
动化技术与应用, 2021, 40 (10): 166-168, 180.
- (上接第 53 页)
- 框大小能够随着被跟踪激光通信设备大小的改变而改变, 具有更好地适应能力; 为了测试 YDK 算法的实时检测跟踪效果, 增加了一组对比试验, 发现平均响应峰值为 0.710, 平均帧率为 13 帧, 平均响应峰值比原 KCF 算法提高了 64.35%, 比 HDK 算法提高了提高 50.4%, 比其在测试视频上提高了 14.5%, 平均帧率比原 KCF 算法下降了 65.6%, 比 HDK 提高了 12.2%, 比其在测试视频上提高了 10.8%; 说明 YDK 算法跟踪效果要优于 HDK 算法与原 KCF 算法。

3 结束语

本文通过将目标检测效果良好的 YOLOV5 算法与 DSST 算法中的尺度估计滤波器算法与 KCF 跟踪算法相结合, 实现长时且有效的目标跟踪, 并且应用到激光通信领域中, 实现了对被跟踪激光通信设备的精准识别与实时跟踪效果, 可代替激光通信领域中的粗瞄准的过程, 很大程度上节约了人工、时间成本。本实验中用到的 YDK 算法与原 KCF、HDK 算法相比, 跟踪精度与准确度有了明显的提高, 但是帧率明显下降, 在满足本实验中系统要求情况下, 会针对提升帧率展开研究, 比如使用效率更高的编程语言, 开多线程等等。

参考文献:

- [1] 姜会林, 胡 源, 丁 莹, 等. 空间激光通信组网光学原理研究 [J]. 光学学报, 2012, 32 (10): 56-60.
- [2] SOVA R M, SLUZ J E, YOUNG D W, et al. 80 Gb/s free-space optical communication demonstration between an aerostat and a ground terminal [C] // Proc. SPIE, 2006.
- [3] 赵 馨, 宋延嵩, 佟首峰, 等. 空间激光通信捕获、对准、跟踪系统动态演示实验 [J]. 中国激光, 2014, 41 (3): 131-136.
- [4] 李爱娟, 巩春鹏, 黄 欣, 等. 自动驾驶汽车目标检测方法综述 [J]. 山东交通学院学报, 2022, 30 (3): 20-29.
- [5] 谢晓方, 刘厚君, 张龙杰, 等. 融合轻量级 YOLOv4 与 KCF 算法的红外舰船目标识别 [J]. 兵器装备工程学报, 2021, 42 (6): 175-182.
- [6] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich Feature hierarchies for accurate object detection and semantic segmentation [C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, 81: 580-587.
- [7] GIRSHICK R. Fast R-CNN [C] // 2015 IEEE International Con-

- ference on Computer Vision (ICCV), 2015, 169: 1440-1448.
- [8] PADALA A K R, MALATHI P. An optimized object detection system using you only look once algorithm and compare with deep neural networks with increased [C] // 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022.
- [9] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C] // IEEE Conference on Computer Vision & Pattern Recognition, IEEE, 2017: 6517-6525.
- [10] YING Z, LIN Z, WU Z, et al. A modified-YOLOv5s model for detection of wire braided hose defects [J]. Measurement, 2022 (190): 190.
- [11] HAGER G D, DEWAN M, STEWART C V. Multiple Kernel tracking with SSD [C] // IEEE Computer Society Conference on Computer Vision & Pattern Recognition, IEEE, 2004, 1: I790-I797.
- [12] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters [C] // The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, IEEE, 2010.
- [13] MEER D. Mean shift analysis and applications [C] // Proceedings of the Seventh IEEE International Conference on Computer Vision 0, Greece, 1999: 1197-1203.
- [14] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37 (3): 583-596.
- [15] 刘 艺, 李蒙蒙, 郑奇斌, 等. 视频目标跟踪算法综述 [J]. 计算机科学与探索, 2022, 16 (7): 1504-1515.
- [16] 胡云层, 路 红, 杨 晨, 等. 融合 DSST 和 KCF 的尺度自适应跟踪算法 [J]. 计算机工程与设计, 2019, 40 (12): 3563-3568.
- [17] 黄 楠, 路 锋, 王钦钊. KCF 算法在车辆目标跟踪上的参数配置研究 [J]. 软件工程, 2019, 22 (9): 12-16.
- [18] 杨志方, 陈 曦. 优化搜索策略的 KCF 目标跟踪算法 [J]. 武汉工程大学学报, 2019, 41 (1): 98-102.
- [19] 马 宁, 曹云峰, 王指辉, 等. YOLOv5 网络架构的着陆跑道检测算法研究 [J]. 激光与光电子学进展, 2022, 59 (14): 189-195.
- [20] MARTIN, DANELLJAN, GUSTAV, et al. Discriminative scale space tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39 (8): 1561-1575.