浙江大学

本科实验报告

课程名称:	计算机组成
实验名称:	实验二：建立 CPU 调试测试环境
姓　　名:	李宇晗
学　　号:	3240106155
专　　业:	计算机科学与技术
实验地点:	东教 509
指导教师:	林芃
报告日期:	2025/10/14

实验一：建立 CPU 调试、测试和应用环境

一、 实验目的及环境

1.1) 实验目的

本次实验旨在通过构建一个基于 IP 核的片上系统（SoC），深入理解计算机系统的基本构成和工作原理。主要目的如下：

- 初步了解 GPIO 接口与外设设备的工作方式。
- 理解计算机系统的基本硬件结构，包括 CPU、存储器、总线和 I/O 设备。
- 掌握计算机各组成部分之间的相互关系与连接方式。
- 学习并掌握在 FPGA 设计中复用和集成 IP 核的方法。
- 了解 SoC（System on Chip）的基本概念，并利用现有 IP 核实现一个能够运行、测试和调试 CPU 的简单 SoC 系统。

1.2) 实验环境

- 硬件平台：
  1. Digilent NEXYS A7 FPGA 开发板。
  2. VGA 显示器
- 软件平台： Vivado 2024

二、 实验目标及任务

2.1) 实验内容与目标

本次实验的核心目标是建立一个基础的 CPU 调试、测试与应用环境，即一个微型的片上系统（SoC）。通过这个过程，熟悉 CPU 核、外设接口（如 GPIO）、功能模块（如时钟分频、按键消抖）、存储器（ROM 和 RAM）以及总线等各个 IP 子模块的功能，并理解它们之间如何互联构成一个完整的系统。

2.2) 实验任务

具体任务是利用实验提供的第三方 IP 核和已有 IP 模块，通过 Verilog HDL 在顶层文件中例化并连接这些模块，最终在 FPGA 上实现一个完整的 SoC。该系统需包含以下关键模块：

- 核心处理器**:** SCPU (一个简化的 RISC-V CPU 核)
- 存储器**:** 指令 ROM 和数据 RAM
- 总线**:** MIO_BUS (用于连接 CPU、存储器和外设)
- 外设与接口**:**
  1. 按键及开关的机械去抖动模块 (SAnti_jitter)
  2. 通用时钟分频模块 (clk_div)
  3. 8 通道数据显示选择模块 (Multi_8CH32)

4. 七段数码管驱动模块 (Seg7_Dev)
5. LED 驱动及 GPIO 接口 (SPIO)
6. VGA 显示控制器 (VGA)
7. 通用计数器 (Counter_x)

2.3) 3. 实验原理

本实验的基本原理是 **IP** 核复用和系统集成。现代复杂的数字系统（如 SoC）设计通常不从最底层的逻辑门开始，而是将预先设计和验证好的、具有特定功能的电路模块（即 IP 核）像搭积木一样组合起来。

整个系统以 **MIO_BUS** 总线为骨干，连接作为主设备（Master）的 **SCPU** 和一系列作为从设备（Slave）的 **RAM**、**GPIO** 等外设。CPU 通过地址总线指定要访问的设备，通过控制总线发出读/写信号，通过数据总线传输数据。每个外设都被映射到系统的一个特定地址空间，总线通过地址译码来选择相应的设备进行通信。通过这种方式，各个独立的 IP 核被整合成一个协调工作的完整计算机系统。

三、 实验过程及记录

3.1) 工程创建与 **IP** 导入：

启动 Vivado，创建一个名为 `OExp02-IP2SOC` 的 RTL 工程，目标器件选择 `xc7a100tcsg324-1`。将实验提供的所有 IP 核源文件（`.v` 和 `.edf`）添加至工程。其中，多数 IP 核通过直接添加源文件的方式导入，而 `ROM` 和 `Seg7_Dev` 则通过 Vivado IP Catalog 进行添加与管理。

3.2) IP 核配置：

在例化前，对 `ROM_D_0` IP 核进行重新定制。在其配置界面中，加载指定的指令初始化文件 `I_mem.coe`，该文件内含用于功能验证的测试程序。随后生成 IP 核。

3.3) VGA 导入

导入./IP/VGA/srcs 中的所有文件，添加后在 VgaDisplay.v 中修改两个.mem 的路径

3.4) 顶层模块设计**(CSSTE.v)**：

新建 Verilog 源文件 `CSSTE.v` 作为系统顶层模块。参照实验手册中的系统结构框图，定义顶层模块的输入输出端口，使其与 NEXYS A7 开发板的物理接口（如时钟、复位、开关、VGA、数码管等）相对应。

3.5) 模块例化与互联：

在 `CSSTE.v` 中，首先声明连接各 IP 核模块所需的内部 `wire` 信号。随后，依据系统设计图，对 `SCPU`、`MIO_BUS`、`RAM`、`ROM` 等全部 11 个 IP 核进行例化。在此过程中，需确保各模块间的端口连接关系准确无误，例如将 `SCPU` 的 `PC_out` 连接至 `ROM` 的地址输入端，并将 `MIO_BUS` 作为连接 CPU 与各外设模块的核心总线。

3.6) 约束、综合与比特流生成：

将官方提供的`A7.xdc`物理约束文件添加至工程，以完成顶层端口到FPGA物理引脚的映射。最后，依次执行“Run Synthesis”、“Run Implementation”和“Generate Bitstream”流程，生成用于下载的`.bit`文件。编程完成后，对FPGA进行复位，启动硬件验证。

四、 实验结果分析

通过操作开发板上的开关和按钮，并观察数码管和VGA显示器的输出，对SoC系统功能进行验证。

4.1) 时钟控制功能验证

- 拨动`SW[8]`和`SW[2]`开关，可以控制CPU的运行时钟。
- `SW[8]SW[2]=00`：CPU全速运行。
- `SW[8]SW[2]=01`：CPU进入自动单步模式，时钟频率显著降低，可观察到程序的慢速执行。
- `SW[8]SW[2]=1x`：CPU进入手动单步模式，此时每拨动一次`SW[10]`，CPU执行一条指令。该功能为逐条指令调试提供了有效手段。

4.2) 数码管调试信息观测

通过`SW[7:5]`的组合来选择在8位七段数码管上显示的内部信号，结果如下：

- **`SW[7:5]=001` (PC地址):** 在手动单步模式下，初始PC值为`00000000`。每执行一条指令，PC值增加4（因为指令为32位，占4个字节），依次显示为`00000004`、`00000008`等，与预期相符。

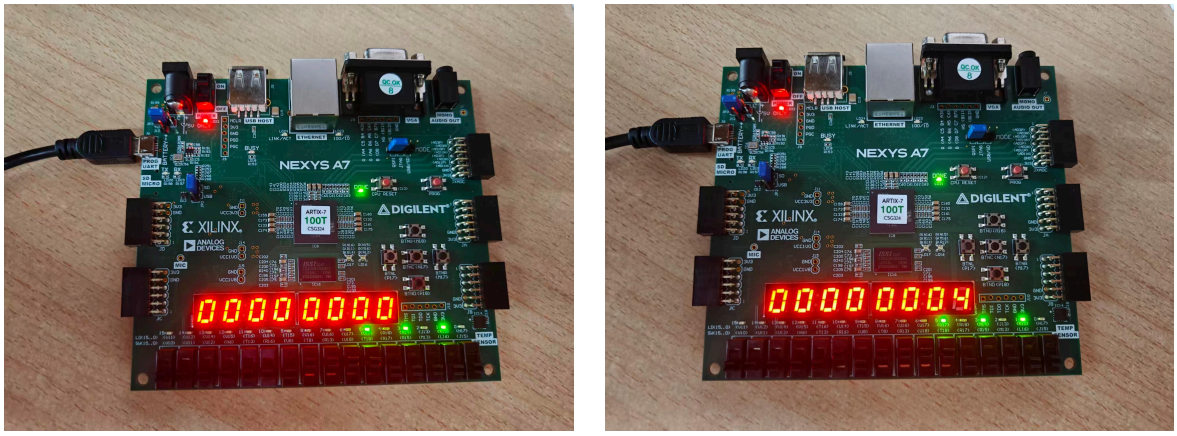

Figure 1: PC地址变化

- **`SW[7:5]=010` (指令内容):** 数码管显示当前PC地址对应的指令机器码。例如，PC为`00000000`时，显示为`00100093`，这与`I_mem.coe`文件中的第一条指令`addi x1, x0, 1`的机器码一致。

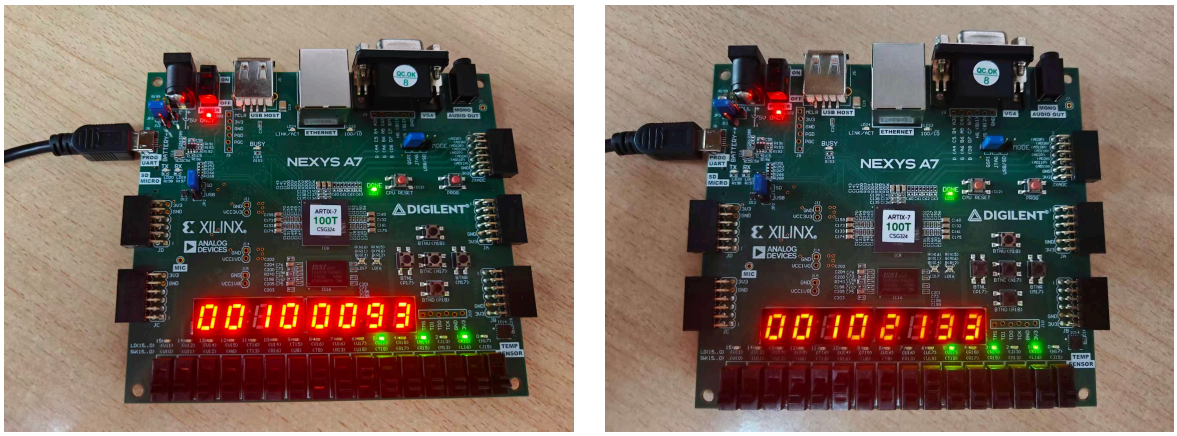

Figure 2: 指令内容变化

- **`SW[7:5]=101` (CPU 数据输出):** 数码管显示 CPU 计算或从寄存器读取的数据。

实验中观察到的各个信号值均与斐波那契数列程序的理论执行结果一致，证明了 CPU、总线和数码管显示通路工作正常。

4.3) VGA 调试信息观测

VGA 显示器提供了一个更详细的调试界面，实时显示 CPU 的关键状态：

- **PC:** 显示当前程序计数器的值，与数码管观测结果同步。
- **inst:** 显示当前正在执行的指令机器码，同样与数码管结果同步。
- **alu_res:** 显示 ALU（算术逻辑单元）的计算结果。例如，在执行 `addi x1, x0, 1` 时，ALU 的结果为 `1`。
- 寄存器组**:** 界面上预留了 32 个通用寄存器的显示位置，但由于本次实验的 VGA 模块并未连接寄存器组的读出端口，所以均显示为 0，这符合实验要求。

VGA 的显示结果进一步确认了 CPU 内部核心部件（PC、ALU）以及指令获取阶段的正确性。

综上所述，所有硬件测试结果均符合预期，表明成功地搭建了一个功能正确的 CPU 调试与测试环境。

五、 讨论与心得

本次实验收获颇丰。通过将抽象的计算机组成理论知识付诸实践，将独立的 CPU、存储器、总线及外设 IP 核集成为一个可执行程序的 SoC 系统，加深了对计算机系统结构与工作原理的理解。

实验过程验证了模块化设计与 IP 核复用在现代数字系统设计中的核心价值。利用预先验证的 IP 核，可以显著提高复杂系统的设计效率与可靠性。本次实验的焦点在于系统集成而非底层设计，这使得我们能够将精力集中于理解各功能模块间的接口与协同工作机制。

总线（MIO_BUS）在系统中扮演了关键角色，作为连接主设备（CPU）与各从设备（存储器、外设）的通信骨干。对总线的地址映射、读写时序和控制逻辑的正确实现，是保证整个系统正常运行的基础。

硬件调试手段的建立是本实验的另一重要成果。通过数码管和 VGA 显示器对 CPU 内部关键信号（如 PC、指令、ALU 结果）进行实时观测，尤其是手动单步功能的实现，为程序的执行流程提供了直观、有效的追踪与验证方法，是进行底层硬件调试的有力工具。

本次实验中遇到的主要挑战在于顶层模块的信号互联，任何连接错误都可能导致系统功能异常。这一过程锻炼了阅读和理解系统框图并将其精确转化为硬件描述语言的能力。

总而言之，本次实验不仅成功地构建了一个 CPU 调试与测试平台，为后续的 CPU 设计实验奠定了坚实基础，更重要的是，通过完整的系统集成实践，提升了分析和解决复杂数字系统设计问题的工程能力。