



**汇编与接口 课 程 设 计**

团队实验报告

|  |  |
| --- | --- |
| 学 院 | 计算机学院 |
| 专 业 | 计算机科学技术 |
| 指导老师 | 王娟 |
| 组 长 | 张驰 |
| 组 员 | 魏慧聪、王浚哲、曹哲瑀 |
| 组长联系方式 | [zhangchi5675@163.com](mailto:zhangchi5675@163.com) |

二O二二年 九月

目 录

[第一章 项目简述 1](#_Toc113199189)

[第二章 组员分工（团队报告，个人报告不需要） 1](#_Toc113199190)

[第三章 设计目的 1](#_Toc113199191)

[第四章 设计环境 1](#_Toc113199192)

[第五章 设计原理及内容 2](#_Toc113199193)

[5.1 数据通路 2](#_Toc113199194)

[5.2 控制逻辑 3](#_Toc113199195)

[第六章 设计与实现 4](#_Toc113199196)

[第七章 测试以及运行结果 6](#_Toc113199197)

[第八章 问题及解决方法 6](#_Toc113199198)

[第九章 心得体会及总结 7](#_Toc113199199)

[第十章 参考文献有价值的资源推荐 7](#_Toc113199200)

# 项目简述

在本次汇编团队项目中，本团队由张驰、魏慧聪、曹哲瑀、王浚哲四人。在该汇编团队项目中，我们使用了在计算机组成原理课程设计中设计的流水线CPU模块，在其中添加了vga显示屏控制接口模块，confreg数码管外设控制接口，设计了相应的汇编程序，通过CPU控制vga显示屏和confreg数码管同步显示计数，下板验证成功。

# 组员分工（团队报告，个人报告不需要）

小组分工见表2.1 小组分工。

|  |  |
| --- | --- |
| **组员** | **分工任务** |
| 组长：张驰 | * VGA接口开发 |
| 组员：魏慧聪 | * VGA上板调试 |
| 组员：曹哲瑀 | * CPU与外设接口集成 |
| 组员：王浚哲 | * VGA控制逻辑开发 |

表2.1 小组分工

# 设计目的

根据精工板资源，选择设计并完成一种接口控制器，如VGA控制器，蓝牙等等。本项目完成了计算机外设接口设计，完成了VGA控制器，仿真正确并且在开发板上上板成功，在VGA控制器上也同样显示正确。

# 设计环境

|  |  |
| --- | --- |
| 操作系统 | Windows10 |
| 编程语言 | Verilog HDL |
| EDA工具 | Vivado2019.2 |
| 汇编语言 | MIPS |
| 汇编程序编辑器 | mars4.5 |

# 设计原理及内容

## 数据通路

本系统由CPU主机模块（包括指令存储器和数据存储器）、vga外设控制接口模块、confreg数码管外设控制接口组成。

CPU的数据通路如下所示：

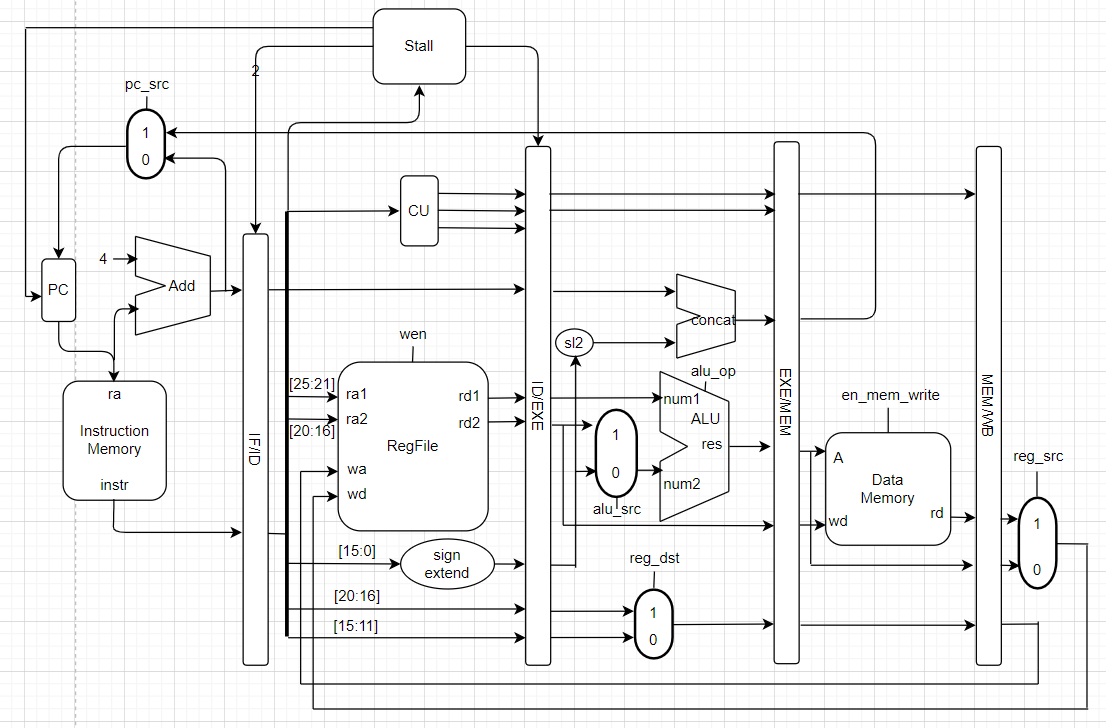


图5.1.1-1 5级流水线CPU数据通路

其数据通路如下

图形用户界面, 应用程序

描述已自动生成

图5.1.1-2 计算系统数据通路

## 控制逻辑

使用已经设计好的CPU流水线模块运行单个变量自增的mips汇编程序，通过confreg数码管外设控制接口和vga外设同步显示，同步完成CPU指令流水线的上板工作和VGA接口的上板工作。

**实现控制逻辑采用方式：**

实现控制逻辑主要通过control模块，alu模块，vga模块，confreg模块。

其中，confreg模块主要用于外设接口数据信号、控制信号的传递和控制。

control模块、alu模块使用组合逻辑实现，其余模块使用时序逻辑实现。

**CPU控制信号列表：**

**en\_reg\_write**：寄存器组写使能，0-不能写，1-能写；

**en\_mem\_write**：数据存储器写使能，0-不能写，1-能写；

**instr[5:0] :** 存储的是指令的前五位码，标注了是哪种类型的指令

**func[5:0] :** 针对某些计算类型的指令，在instr[5:0] = 5’b00000时， 根据func[5:0]判断执行的是哪条指令。

**外设接口控制信号列表：**

**confreg\_wen**：外设写使能，0-不能写，1-能写；

**hs**：vga接口的水平同步信号；

**vs**：vga接口的垂直同步信号；

**外设接口数据信号列表：**

**num**：vga显示的数字，来源于外设控制模块；

**confreg\_data**：CPU向外设传输的数据；

**confreg\_addr**：CPU写外设的地址；

**confreg\_rd**：外设向CPU传输的数据；

**digcs**：八位片选信号，控制数码管的八个数字；

**digital\_num0/1**：数码管0-3位，4-7位显示数字；

**r、g、b**：颜色控制信号；

# 设计与实现

本部分除了流水线CPU模块外，增加实现了confreg模块和vga模块，用于传递和执行外设的控制信号和数据信号。

1. top\_pipeline：顶层模块，负责CPU主机与外设接口的连接、信号的选择
2. confreg：confreg模块作为数码管外设接口控制器。判断CPU的写地址的范围，从而确定对外设的写操作，从而实现外设与存储器的统一编址；设置计数器，使用计数器的变化来控制外设的片选信号及选取CPU写入数据的哪四位，最后将选取的四位二进制转变为数码管的七位显示信号输出。
3. vga：VGA显示屏的控制模块。该模块模拟十三段数码管的显示，配合CPU输出的信号可以完成一个在显示器上计时的功能。VGA的功能是，根据数码管的数字，向VGA发送相应的数据信号，根据数字信号，对VGA的十三个区域内的RGB进行赋值操作，从而发送每一个像素的RGB值。当RGB=(F,0,0)，则所在位置的数码管是亮的，是对应的RGB颜色，最后统一在显示屏上显示对应的图像。

使用计算机组成原理课程设计中实现的流水线CPU，我们基于流水线CPU，设计了如下自增程序，使内存段中地址为0x8000处的数据不断递增。

confreg和vga模块会从该数中截取两位作为输入信号，连接精工板上对应的晶体管，显示数据的内容，并通过外设接口，使其在VGA模块上进行显示。

1. .org 0x0
2. .set noat
3. .set noreorder
4. .set nomacro
5. .global \_start
6. \_start:
7. lui $v0,0xbfaf
8. addiu $v0,$v0,0x8000
10. sw $t0,0x0($v0)
12. j loop
13. addi $t4,$t4,0x0000
14. loop:
15. addiu $t0,$t0,0x0001
17. sw $t0,0x0($v0)
18. j loop
19. addi $t5,$t5,0x0000

相应的机器码如下所示：

1. memory\_initialization\_radix=16;
2. memory\_initialization\_vector=
3. 3c02bfaf,
4. 3c010000,
5. 34218000,
6. 00411021,
7. ac480000,
8. 0bf00007,
9. 218c0000,
10. 25080001,
11. ac480000,
12. 0bf00007,
13. 21ad0000;

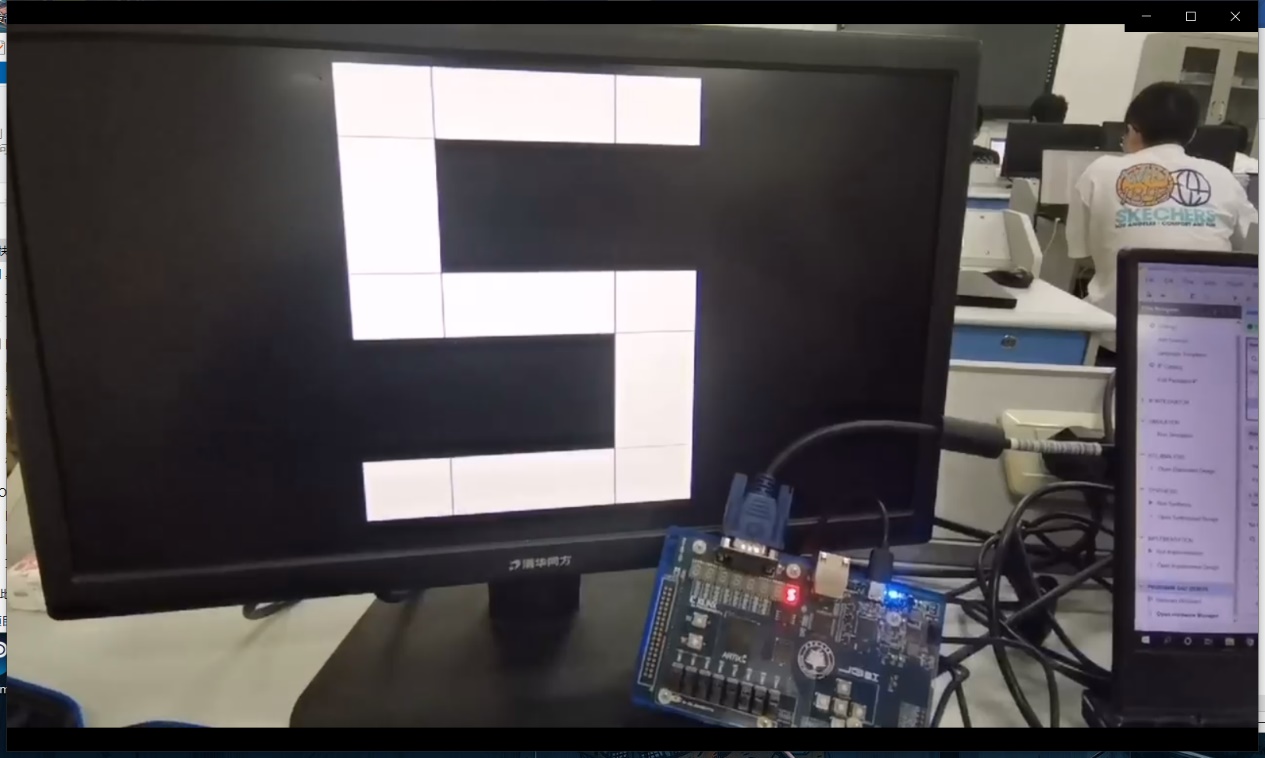
核心模块是confreg模块和vga模块，confreg和vga的接口控制如下所示：

1. confreg confreg(
2. .clk(clk),
3. .rst(rstn),
4. .confreg\_wen(confreg\_wen),
5. .confreg\_write\_data(confreg\_wdata),
6. .confreg\_addr(confreg\_addr),
7. .confreg\_read\_data(confreg\_rdata),
8. .digital\_num0(digital\_num0),
9. .digital\_num1(digital\_num1),
10. .digital\_cs(digital\_cs),
11. .counter\_num(counter\_num)
12. );
13. vga vga(
14. .clk(clk),
15. .rstn(rstn),
16. .num(counter\_num[3:0]),
17. .hs(hs),
18. .vs(vs),
19. .r(r),
20. .g(g),
21. .b(b)
22. );

# 测试以及运行结果

测试结果如下图所示：

完整流程可参考演示视频，计数效果如下图所示



# 问题及解决方法

1. 问题：VGA接口部分，仿真结果正确，但上板测试时显示器无法显示正确结果。

解决：发现是时钟频率问题导致，应该根据所使用的分辨率计算对应的VGA时钟信号。我们输出的是1024\*768分辨率，60Hz的视频信号，所以需要65MHz的VGA时钟信号。创建一个IP核来实现时钟信号的转换，即可解决这个问题。

1. 问题：在运行时，发现Inst\_rom IP核在录入相应的汇编机器码出现了错误，导致上板时没有运行结果。

解决：使用upgrad 指令更新IP核，同时再次调整clk\_wiz的IP核的时钟信号。在更新IP核后，注意需要调整机器码的相关地址。本CPU内存空间地址的起始地址是0xbfc00000，注意将机器码lui指令中的地址更改位0xbfc0。

# 心得体会及总结

在本次汇编与接口课程设计的组队任务中，我们完成了基础团队任务核进阶团队任务。根据精工板资源，我们选择设计并完成了VGA的外设控制器，测试后形成IP核，集成接口控制器核处理器的设计结果，形成了自定义的计算机系统，仿真正确并且成功在精工板上板实现。在此次的实验过程中，我们的外设VGA连接成功与我们此前设计的流水线CPU结合，实现了比较简单的自增汇编程序的运行。

通过这次实验，我们进一步熟悉了汇编语言，了解并掌握了接口控制部分，加深了对硬件知识的理解与掌握。

# 参考文献有价值的资源推荐

无