



**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 | 胡玄哲 |
| 组 员 | 何宇轩、刘懿达、田东琦 |
| 组长联系方式 | QQ：1321992397 |

二O二 三 年 九 月

目 录

[第一章 项目简述 1](#_Toc83666855)

[第二章 组员分工（团队报告，个人报告不需要） 1](#_Toc83666856)

[第三章 设计目的 1](#_Toc83666857)

[第四章 设计环境 1](#_Toc83666858)

[第五章 设计原理及内容 2](#_Toc83666859)

[5.1 数据通路 2](#_Toc83666860)

[5.2 控制逻辑 2](#_Toc83666861)

[第六章 设计与实现 3](#_Toc83666862)

[第七章 测试 5](#_Toc83666863)

[第八章 问题及解决方法 6](#_Toc83666864)

[第九章 心得体会及总结 6](#_Toc83666865)

[第十章 参考文献有价值的资源推荐 6](#_Toc83666866)

# 项目简述

集成了CPU主机模块、vga外设控制接口模块、七位数码管外设控制接口，设计汇编程序，通过CPU控制vga外设和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：宋尚儒 | * 运算逻辑开发 * VGA接口开发 * CPU与外设接口集成 |
| 组员：何宇轩 | * 仿真测试 * PPT制作 |
| 组员：刘懿达 | * 仿真测试 |
| 组员：田东琦 | * 仿真测试 * PPT制作 |

表2.1 小组分工

# 设计目的

根据精工板资源，完成计算机外设接口设计，包括VGA控制器、UART等

# 设计环境

|  |  |
| --- | --- |
| 操作系统 | Windows11 |
| 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 |

请标注版本号

# 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，其数据通路如下

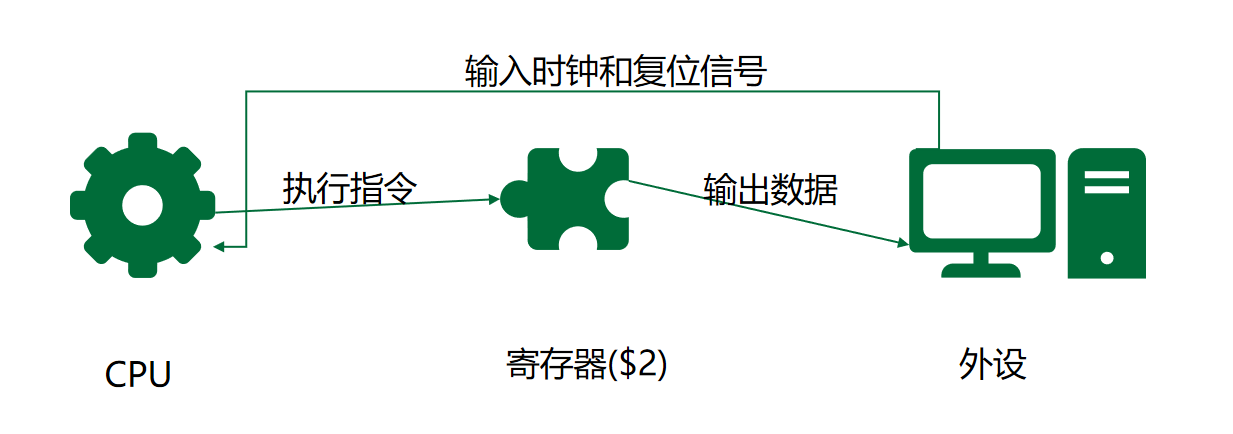


图5.1.1-1 计算系统数据通路

## 控制逻辑

**实现控制逻辑采用方式：**

vga模块、数码管模块等外设控制模块使用时序逻辑实现。

**外设接口控制信号列表：**

**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：**

**sel：**八位片选信号，控制数码管的八个数字；

**reg[7:0] seg\_code：**数码管数据信号；

**r、g、b**：颜色控制信号；

# 设计与实现

除了在计算机组成原理实验中实现的CPU整体模块之外，还实现了以下模块

1. openmips\_min\_sopc：顶层模块，负责CPU主机与外设接口的连接、信号的选择
2. seg：数码管外设接口控制器。将cpu输出的的四位二进制转变为数码管的七位显示信号输出。
3. vga：VGA控制模块，模拟类似七段数码管的十三段数码管，配合CPU输出的信号可以完成一个在显示器上计时的功能，根据输入的数字，通过一定的时序向显示器发送同步信号，对十三个区域的像素RGB值进行赋值，从上到下，从左到右向显示器发送每一个像素的RGB值，如果发送的像素所在位置的数码管是亮的，RGB = (F,0,0)，否则RGB = (0,0,0)，将图像显示在显示器上。

将如下MIPS汇编程序转换为机器码后存储于指令存储器中，cpu将2号寄存器的数据输出到外设，confreg和vga模块会从该数中截取两位作为输入信号。

1. .org 0x0
2. .set noat
3. .set noreorder
4. .set nomacro
5. .global \_start
6. \_start:
7. ori $1,$0,1 #$1 = 1
8. ori $2,$0,0 #$2 = 0
9. ori $3,$0,0 #$3 = 0
10. ori $4,$0,0 #$4 = 0
11. ori $5,$0,0 #$5 = 0
12. loop:
13. add $2, $2, $1
14. add $3, $3, $2
15. add $4, $4, $3
16. add $5, $5, $4
17. j loop
18. sub $1, $1, $1

最终实现接口如下所示

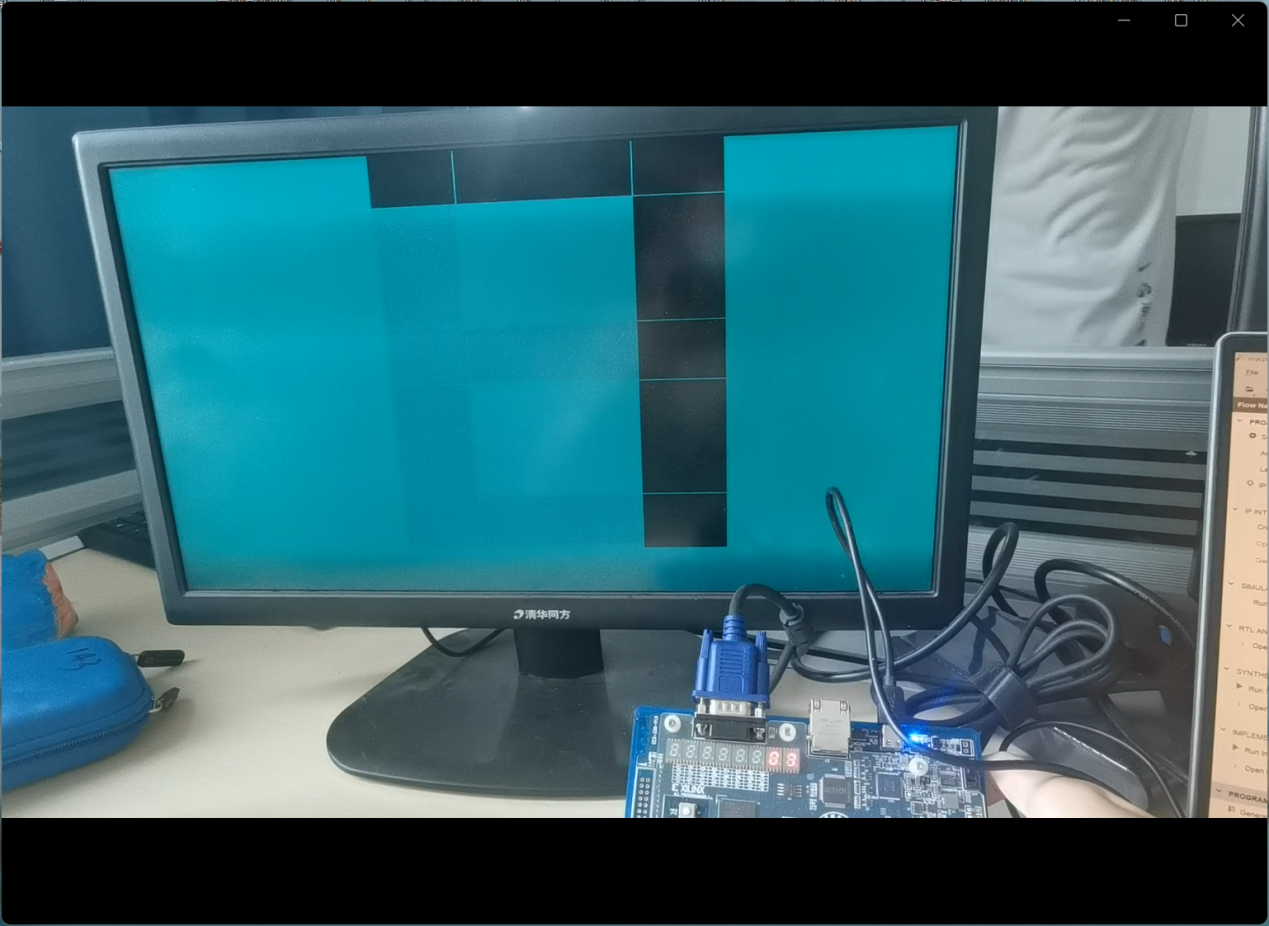
1. module openmips\_min\_sopc(
2. input wire clk,
3. input wire rst,
5. //seg
6. output wire[7:0] sel,
7. output wire[7:0] seg\_code,
9. //vga
10. output hs,
11. output vs,
12. output[3:0] r,
13. output[3:0] g,
14. output[3:0] b
16. );
18. module openmips(
19. input wire clk,
20. input wire rst,
22. input wire[`RegBus] rom\_data\_i, //从指令rom中取出的指令
23. output wire[`RegBus] rom\_addr\_o, //输出到指令rom的地址
24. output wire rom\_ce\_o, //指令rom的芯片使能信号
26. output wire[`RegBus] data\_o //输出到外设数据
27. );
29. module seg\_cnt(
30. input clk,
31. input rst,
33. input wire[`RegBus] data\_i,
34. output reg[7:0] sel,
35. output reg[7:0] seg\_code
37. );
38. module vga (
39. input clk,
40. input rst,
41. input [3:0] num,
43. output hs,
44. output vs,
45. output [3:0] r,
46. output [3:0] g,
47. output [3:0] b
49. );

# 测试

功能测试在开发板上进行，流程可概括如下

1. 开发板连接显示器，启动开发板
2. 开发板连接电脑，将程序写入开发板中
3. 程序正常运行，开发板七段数码管和显示器持续计数

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

无

# 心得体会及总结

通过设计实现了VGA等外设接口，并成功与我们此前设计的流水线CPU结合，实现了比较简单的计算系统。通过这次实验，我们对计算机系统的运行原理有了更清楚的认知，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