



**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 | 宋尚儒 |
| 组 员 | 陈鸿韬、赵旭 |
| 组长联系方式 | 1120180717@bit.edu.cn |

二O二一年 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、confreg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 |
| 组员：陈鸿韬 | * VGA接口开发 |
| 组员：赵旭 | * CPU与外设接口集成 |

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 | 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 |

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

图形用户界面, 应用程序

描述已自动生成

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：**

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，vga模块、confreg模块等外设控制模块也使用时序逻辑实现。

**外设接口控制信号列表：**

**confreg\_wen**：外设写使能，0-不能写，1-能写；

**is\_confreg\_addr**：CPU读到的数据来源，0-数据存储器，1-外设；

**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：**

**count\_num**：vga显示的数字，来源于外设控制模块；

**confreg\_write\_data**：CPU向外设传输的数据；

**confreg\_addr**：CPU写外设的地址；

**confreg\_read\_data**：外设向CPU传输的数据，根据is\_confreg\_addr判断是否写入CPU；

**digital\_num0**：数码管0到3位显示数字；

**digital\_num1**：数码管4到7位显示数字；

**r、g、b**：颜色控制信号；

**digital\_cs**：八位片选信号，控制数码管的八个数字；

# 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. 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：数码管外设接口控制器。通过对CPU的写地址进行范围判断，确定是否为对外设写以及对哪一种外设写，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，可以将一个不断递增的数写入数据存储器0x8000地址处，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. 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

最终实现接口如下所示

1. top\_pipeline(
2. input wire rstn,
3. input wire clk,
4. /\*\*\*\*\*\*\*\*\* confreg \*\*\*\*\*\*\*\*\*/
5. output wire[6:0] digital\_num0,
6. output wire[6:0] digital\_num1,
7. output wire[7:0] digital\_cs,
8. /\*\*\*\*\*\*\*\*\*\* vga \*\*\*\*\*\*\*\*\*\*\*\*\*\*/
9. output hs,
10. output vs,
11. output [3:0] r,
12. output [3:0] g,
13. output [3:0] b
14. );
15. cpu mycpu0(
16. .rst(rstn), // input
17. .clk(clk), // input
18. .imem\_ra(inst\_rom\_addr), // output
19. .imem\_rd(inst\_rom\_rdata), // input
20. .dmem\_ra(data\_ram\_addr), // output
21. .dmem\_wd(data\_ram\_wdata), // output
22. .dmem\_wen(data\_ram\_wen), // output
23. .dmem\_rd(data\_ram\_rdata) // input
24. );
25. confreg confreg(
26. .clk(clk),
27. .rst(rstn),
28. .confreg\_wen(confreg\_wen),
29. .confreg\_write\_data(confreg\_wdata),
30. .confreg\_addr(confreg\_addr),
31. .confreg\_read\_data(confreg\_rdata),
32. .digital\_num0(digital\_num0),
33. .digital\_num1(digital\_num1),
34. .digital\_cs(digital\_cs),
35. .counter\_num(counter\_num)
36. );
37. vga vga(
38. .clk(clk),
39. .rstn(rstn),
40. .num(counter\_num[3:0]),
41. .hs(hs),
42. .vs(vs),
43. .r(r),
44. .g(g),
45. .b(b)
46. );

# 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

桌子上放满了不同类型的电脑

描述已自动生成

# 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：Mars将汇编代码转为机器码时，label所转成的地址为Mars中的绝对地址，该地址在仿真时不正确。

解决：通过手动计算地址的方法，计算出label相对于vivado中指令存储器基址0xbfc00000的绝对地址，再求出对应的机器码。该机器码用于仿真，即可正确执行。

# 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