组号：_____

计算机系统综合课程设计方案及认证报告

东南大学计算机科学与工程学院

20__20年_10_月

组长姓名	丁语琪	手机号	15013858201
Email 地址	1145179254@qq.com		
本组成员情况（含组长）			
姓　名	学　号	承　担　的　任　务	
丁语琪	09017212	指令集和数据通路	
汪晶晶	09017206	指令集和数据通路	
张堂华	09017221	链接器设计和集成开发环境设计	
刘芷伶	09017347	MiniSys-1A 汇编器的设计	
江万畅	09017306	MiniSys-1A 编译器的设计	
银雪岑	09017233	硬件CP0协处理器设计、存储器设计、接口设计	

注意：本表所列为本组预计的目标设计方案，具体在实施的时候原则上目标不能删减，如要删减，需经过教师同意。技术路线可以根据实际情况修改，技术路线的修改无需教师认可。
本报告页数不够可以增加页数。

本组设计的预计目标描述（含所有预计模块的功能）

硬件设计部分

1. CPU 设计：可以运行指定 57 条MIPS 指令的 RISC 型MiniSys-1A32 位流水微处理器，实现 MIPS 的CP0 来处理中断异常。

2. 存储器设计：哈佛结构存储器，有独立的 64KB 指令存储和 64KB 的数据存储。如果有余力，将尝试设计并实践三级存储。

3. 接口设计：2 个16 位定时/计数器、4×4 键盘控制器、8 位7 段数码管控制器、16 位 LED 输出、16 位拨码开关输入、PWM 控制、看门狗控制器的设计。

软件设计部分

1. MiniSys -1A 汇编器的设计：实现汇编指令到二进制指令集的映射转换。
2. Mini C 编译器的设计（有全局查错能力）：将目标代码通过 LEX 词法分析和 Yacc。语法分析生成汇编代码。
3. 简单 BIOS 的设计
4. 集成开发环境：带有语法高亮和代码全屏编辑的集成开发环境。主要支持一下功能：1、语法高亮：支持对代码中的关键字、变量和符号快速标识，增加代码的可读性和条理性；2、保存和重载代码文件：支持保存现有工作，并在之后重新打开时保持离开时的状态；3、在环境内运行代码；4、支持调试；5、自动补充代码格式；6、支持全屏编辑。

5. 链接器：设计一个能够把汇编生成的多个独立的目标文件和 BIOS 库通过静态链接生成可被加载进内存的可执行文件的系统程序。

全组预计最终成果达到所有基本要求，功能正确、能软硬件联合运行仿真。如果在完成基本要求且时间充裕，可能尝试完成较高要求中的三级存储功能。

本组设计的技术路线

1. 集成开发环境：设计一个 GUI 图形界面，与传统图形界面不同的是集成开发环境的图形界面需要支持实时刷新，键盘每次输入都会调用所有的函数，实时反映代码状态。同时还需要对输入的代码进行分析，从而对关键字、变量和符号实现高亮和自动补充代码格式。在 GUI 内要提供编译和执行代码的按钮，允许代码在环境内调试执行。

2.链接器：链接器的执行过程可分为三个主要步骤：符号决议、生成可执行文件、重定位。

在符号决议中，链接器的工作就是确保所有目标文件中的符号引用都有唯一的定义。链接器会依次扫描每个目标文件并维护两个集合，分别是已定义符号集 D和未定义符号集 U。步骤如下：1、对于当前目标文件，查找其符号表，并将已定义的符号并添加到已定义符号集合 D 中；2、对于当前目标文件，查找其符号表，将每一个当前目标文件引用的符号与已定义符号集合 D 进行对比，如果该符号不在集合 D 中则将其添加到未定义符合集合 U 中；3、当所有文件都扫描完成后，如果为定义符号集合 U 不为空，则说明当前输入的目标文件集合中有未

定义错误，链接器报错，整个编译过程终止。

通过静态链接生成可执行文件就是一个简单的 COPY 过程，即将各目标文件和静态库的代码段、数据段复制到可执行文件中。

编译器在将源代码编译生成目标文件时需要定义该源文件中函数和全局变量的内存地址，而这里的地址是相对于自己的地址，同时链接器还要确定引用其他目标文件的变量的内存地址。链接器在链接过程根据.rel.data 和.rel.text 来填下编译器留下的空白位置。总结来说就是把每个符号定义和内存位置相关联，从而重定位这些节，并且修改所有对这些符号的引用，使它们指向这个内存位置，而这一过程是依靠汇编器产生的重定位信息的详细指令。

3.编译器：

编译器分为词法分析、语法分析、中间代码生成和汇编代码生成四个部分。词法分析思路：依据单词的正规表达式定义，基于有限自动机来完成词法分析程序。为了更好地体现用户灵活性，采用 LEX 方法来自动生成。MiniC 词法描述文件clex.l，MiniC 词法分析程序 clex.c。语法分析思路：使用 Yacc，MiniC 语法描述以及翻译规则描述 cYacc.y，基于 LR(1)的语法分析程序 cYacc.c。完成上述程序编写之后，完成语法知道翻译规则处理，将 MiniC 转换成三地址码。最终将三地址码转换成为汇编代码。

4.汇编器：

开发技术分析

a.注意地址对应关系

对于代码部分.text 的翻译

• 源程序的代码部分.text 后的地址是实际的 32 位字地址，而在 prgmip32.coe 中的单元地址是实际字地址除以 4 后取低 15 位的结果。也就是 address[16:2]。

b. 注意地址对应关系

对于数据部分.DATA 的翻译

• 源程序的数据部分.DATA 后的地址是实际的 32 位字地址，而在 ram?.coe 中的地址是实际字地址除以 4 后取低 15 位的结果，也就是 address[16:2]。然后还要注意交叉存储的问题。

c. 中断处理程序设计问题

中断处理程序的步骤

① 中断或异常到来，硬件跳转到同一个固定地址（各组自行商定，但设计报告中必须公开该地址）

② 中断（异常）处理程序保存上下文

③ 读取 cause 寄存器中的 ExeCode 代码，根据该代码跳转到不同的处理部分

④ 中断或异常事件处理

⑤ 恢复上下文

⑥ eret 指令从中断或异常中返回

d. 中断处理程序设计问题

关于中断嵌套的问题

• Minisys-1A 基础版本不支持中断嵌套，如要实现中断嵌套，需考虑：

—中断优先级（Status 寄存器的 IM[1-0]位）， 中断屏蔽， 堆栈的使用

关于中断屏蔽的问题

• 硬件自动屏蔽和解屏蔽

—进入中断处理程序时就自动屏蔽中断

—中断返回时自动恢复被硬件自动屏蔽的中断

• 软件屏蔽和解屏蔽

—利用 CP0 中 Status 寄存器的 IM[7-2]位的修改来实现

e. 汇编器宏指令

什么是宏指令：将具有固定功能的几条汇编指令序列定义成一个扩展指令

汇编器遇到这样的扩展指令，将展开为正确的指令序列，要注意地址变化

如果需要保持宏指令执行的原子性，硬件和软件上要增加对中断的屏蔽至少应包含的宏指令（注意$1 会被临时用到）：

• push $7 相当于 addi $sp,$sp,-4、 sw $7,0($sp) 两条指令

• pop $7 相当于 lw $7,0($sp)、 addi $sp,$sp,4 两条指令

• jg $2,$3,addr 相当于两条指令 slt $1,$3,$2 、 bne $1,$0,addr

- jle $2,$3,addr 相当于两条指令 slt $1,$3,$2 、 beq $1,$0,addr
- jl $2,$3,addr 相当于两条指令 slt $1,$2,$3 、 bne $1,$0,addr
- jge $2,$3,addr 相当于两条指令 slt $1,$2,$3 、 beq $1,$0,addr

f. 变量和标号的处理

对于变量和标号，要建立相关的符号表，表中给出变量和标号和实际地址的对应关系。对于标号，必要的时候可通过两遍扫描来获得地址。对于标号的使用，使用实际地址还是实际地址右移两位的值，需要根据指令格式来决定。

5.指令集和数据通路：

我们采用 Verilog 语言在 Vivado 平台上进行硬件部分的开发。硬件方面的设计可分为两个阶段，第一阶段是基础设计，第二阶段是进阶设计。基础设计阶段可以先实现简单的 31 条MIPS 指令的 Minisys-1 单周期 CPU 设计，包含取指、译码、控制、运算、存储等模块，存储器采用哈佛结构，分为指令存储器和数据存储器，基础设计要能实现简单的 LED 和拨码开关功能。在实现了简单的基础设计阶段之后，我们大致搭建了数据通路的框架，然后再把单周期 CPU 改造成多周期CPU，这就需要修改数据通路，为了每个阶段的结果保持稳定，需要添加指令寄存器、ALU 结果寄存器和相应信号，还要设计状态机，记录一个指令周期中不同的阶段及阶段转换，并根据当前译码结果产生相应的控制信号，针对数据通路的变化，进行相应部件的修改，包括控制单元、取指单元、执行单元、顶层文件。

在第二阶段的进阶设计中，我们再对之前的任务进行扩展。对于指令集，我们将原有的 31 条指令扩展成 57 条，形成一个较为完整的指令集。在 57 条MIPS 指令的 Minisys-1A 指令基础上， 我们再进一步改进数据通路，要在原有的基础上增加流水各阶段的流水寄存器、HI 和LO 寄存器、乘法器、除法器、交叉处理器和协处理器，还应当考虑新增指令对状态机的状态变迁及输入与输出的变化。进一步地，我们可以再考虑解决流水处理器中可能出现的结构相关、数据相关和控制相关问题。

6.CP0 协处理器：

协处理器将基于 vivado 开发环境使用 Verilog 语言进行设计。设计思路参考 MOOC 课程等，当译码单元中发现指令是特权或异常指令时，将自动设计 CP0 寄存器并跳转到一个固定地址，再根据情况专区执行相应的异常或中断处理程序，程序部分由软件负责，同时设计过程中需要和负责数据通路和流水线的同学及时沟通。

7.存储器：

首先与软件同学沟通协商，存储器将在基于哈佛架构的情况下，参考《现代微机原理与接口技术》、《计算机系统结构》等书籍上关于存储器的设计。基于vivado 开发环境并使用 Verilog 言进行设计，对于相应的 RAM 和ROM 进行实现，

初始化等操作。

8.接口：

首先与软件同学沟通协商，使用 Verilog 语言开发环境选择 vivado。2 个 16 位定时/计数器、4×4 键盘控制器、8 位 7 段数码管控制器、16 位 LED 输出、16 位拨码开关输入、PWM 控制、看门狗控制器的设计将分别按照相应课件中的设计方式进行设计。