



C o m p u t e r O r g n i z a t i o n



CS202-2022s

CPU 大作业要求



总体说明

答辩 说明	代码提交	文档（视频） 提交
提前答辩（15周实验课时间）	答辩前	16周周一前
正常答辩（16周实验课时间）	答辩前	17周周一前
推迟答辩（16周周五，17周）	答辩前	答辩的后一周周一前

组队及开发板领用说明：三人组队，每小组一块开发板

团队得分(15周答辩) = 功能验收分*系数（周一 1.05，周三 1.04，周四 1.03）+ 代码规范 + 文档

团队得分(16周答辩) = 功能验收分+ 代码规范 + 文档

个人得分 = 团队得分 * 团队人数 * 个人贡献百分比

延迟答辩 * (0.6~0.9)，延迟提交 * (0.6~0.9)

评分说明

- 评分以代码规范（结构化设计、变量命名、代码规范、注释）、文档、功能验收演示为准。
- 包括基本分（100分：基本功能 + 文档 + 代码规范）和bonus（30分）两部分，如得分超过100，则溢出的部分将按比例计入总评
- 基本功能：
 - 1) 覆盖到所支持的ISA中的全部指令
 - 2) 使用外设的种类数>2
 - 3) 测试通过基本场景1、基本场景2
- bonus 包括但不限于：
 - 1) 实现对复杂外设接口的支持（如VGA接口、小键盘接口）
 - 2) 使用uart接口实现不重新烧写FPGA芯片的前提下加载不同的程序到CPU上做执行
 - 3) 基于现有 ISA Minisys 实现的CPU，在CPU的功能或性能上实现的优化
 - 4) 针对现有Minisys 的ISA，做的指令类型扩展、功能扩展和实现
- 补充：实现其他类型的ISA（如RISC-V，MIPS32）将按照以上要求进行检查，根据完成情况在总分基础上乘以1.1~1.3的系数（该情况下，不重复考虑bonus的3、4部分）。

提交要求

➤ 提交要求（每小组只需要提交一份，分两次提交）：

➤ 第一次提交源代码（答辩前提交bb站点）

➤ 代码：含CPU设计文件（包括IP核说明文件xci）、仿真用的testbench文件、上板测试用的约束文件、测试场景对应的asm以及coe文件（用于查重）

➤ 压缩包的名字格式为：**c答辩时间_小组成员姓名列表**

比如：c16178_A_B_C（其中c16178表示16周周一78节课上答辩做的代码提交，A,B,C是三名队友的名字）

➤ 第二次提交文档及视频（答辩当周的下周周一之前）

➤ 文档（pdf格式），文档名：**d答辩时间_小组成员姓名列表**，如无视频则只提交文档即可

➤ 视频（可选）：

➤ 争取bonus的小组须针对bonus部分录制项目功能演示视频，不争取bonus的小组对视频不做要求

➤ 请将文档和视频放一个文件夹压缩后提交，压缩包的名字格式为：**dv答辩时间_小组成员姓名列表**

文档要求

- 开发者说明：每个成员的学号、姓名、所负责的工作、贡献百分比。
- 版本修改记录（可选）：版本号、时间、更新点描述
- CPU架构设计说明
 - CPU特性：
 - ISA（含所有指令（指令名、对应编码、使用方式），参考的ISA，基于参考ISA本次作业所做的更新或优化；寄存器（位宽和数目）等信息）；对于异常处理的支持情况。
 - 寻址空间设计：属于冯·诺依曼结构还是哈佛结构；指令空间、数据空间、外设IO的寻址范围，寻址单位。
 - CPU的CPI，属于单周期还是多周期CPU，是否支持pipeline（如支持，是几级流水）。
 - CPU接口：时钟、复位、uart接口（可选）、其他常用IO接口使用说明。
 - CPU内部结构
 - CPU内部各子模块的接口连接关系图
 - CPU内部子模块的设计说明（模块功能、端口规格及功能说明）
- 测试说明：以表格的方式罗列出测试方法（仿真、上板）、测试类型（单元、集成）、测试用例描述、测试结果（通过、不通过）；以及最终的测试结论。
- 问题及总结：开发过程中遇到的问题、思考、总结。

答辩要求

➤ 答辩前准备：

- 设备：请准备两台安装有vivado的电脑参与答辩（需现场修改汇编代码，烧写fpga芯片，对照代码回答问题，方便同步开展）。
- 答辩次序登记：在共享文档中登记答辩时间、答辩次序。

➤ 答辩包括：

- 演示、问答两个环节,所有组员都必须到场并回答问题(未返校的同学需线上接入会议参加答辩)。
- 要求现场根据演示要求修改汇编源代码，完成汇编、下发程序、测试的完整过程。

➤ 演示过程中需按要求完成CPU的上板（Minisys开发板）测试。

- CPU的基本测试场景（参见后页具体内容）
- CPU的扩展功能（可选）包括但不限于：
 - 实现minisys已有指令的优化及测试；
 - 实现除minisys以外的其他ISA的指令实现及测试；
 - 支持只烧写一次FPGA芯片（CPU实现），通过uart接口做不同汇编程序的下发、执行操作；
 - 支持多于2种IO设备；
 - 实现更好的用户交互体验...

基本测试场景1

说明：使用Minisys开发板上的16+3个拨码开关用于做输入，其中3个拨码开关（sw23, sw22, sw21）用于测试用例的编号输入，16个拨码开关（sw15,..sw0) 用于做测试数据的输入，使用17个led灯做输出

场景1.用例编号	用例描述
3'b000	输入测试数a，输入完毕后在led灯上显示a，同时用1个led灯显示a是否为二进制回文的判定（根据a的实际位宽来做判断，比如4'b1001是回文，该led灯亮，否则该led灯不亮）
3'b001	输入测试数a，输入完毕后在输出设备上显示a；输入测试数b；输入完毕后在输出设备上显示b
3'b010	计算 $a \& b$ ，将结果显示在输出设备
3'b011	计算 $a b$ ，将结果显示在输出设备
3'b100	计算 $a \wedge b$ ，将结果显示在输出设备
3'b101	计算 a 逻辑左移 b位，将结果显示在输出设备
3'b110	计算 a 逻辑右移 b位，将结果显示在输出设备（测试时会考虑先做左移后再右移）
3'b111	计算 a 算数右移 b位，将结果显示在输出设备（测试时会考虑先做左移后再右移）

基本测试场景2： 1/2

- 使用Minisys开发板上的3+8个拨码开关用于做输入，其中3个拨码开关（sw23，sw22，sw21）用于测试用例的编号输入，8个拨码开关（sw7,...sw0）用于做测试数据的输入（sw7对应于8bit的最高bit位bit7，sw0对应于8bit的最低bit位bit0）；在数据存储区域中申请4块连续区域用于分别存放下表中的4块数据，4块区域的大小一致（对应与下标中的space）；

数据集编号	数据集0	数据集1	数据集2	数据集3
数据集的起始地址	base + 0	base + 1*space	base + 2*space	base + 3*space
数据说明	输入数据原样保存	将输入数据视为无符号数，按从小到大排序后顺序存放的数据	将输入数据视为按照下标中的3'b010对应行的方式做识别，转换成补码后进行存放，存放次序与数据集0中的输入数据一致	基于“数据集2”中的数据，视其为有符号数的补码，按从小到大排序后顺序存放的数据

场景2.用例编号	用例描述
3'b000	输入测试数据的个数（小于或等于10），以2进制的方式输入多个测试数据，将测试数据原样存入上表”数据集0“对应的空间
3'b001	将测试数据视为无符号数（bit7与bit6~bit0一样，都被视为数值位的一部分），将数据按照从小到大的方式进行排序，排序后的数据集合作为一个整体存入上表”数据集1“对应的空间中（数值最小的存放在低地址，依次类推，数值最大的存放在高地址）
3'b010	将测试数据（此时”数据集0“中的每一个数据，其bit6~bit0被视为该数值的绝对值，bit7对应该数值的符号位）转换为有符号数的补码形式，将做完转换后的数据集存入上表”数据集2“对应的空间中（存放次序与数据集0中的数据一致）
3'b011	基于“数据集2”中的数据，视其为有符号数的补码，按从小到大排序后顺序存放的数据，排序后的数据集合作为一个整体存入上表”数据集3“对应的空间中（数值最小的存放在低地址，依次类推，数值最大的存放在高地址）

基本测试场景2:2/2

使用Minisys开发板上3（sw23，sw22，sw21用于测试用例编号的输入）+4（用于指定数据集内元素的下标）个拨码开关用于做输入，按照表格中指定的方式进行操作，输出指定结果

数据集编号	数据集0	数据集1	数据集2	数据集3
数据集的起始地址	base + 0	base + 1*space	base + 2*space	base + 3*space
数据说明	输入数据原样保存	将输入数据视为无符号数，按从小到大排序后顺序存放的数据	将输入数据视为按照下标中的3'b010对应行的方式做识别，转换成补码后进行存放，存放次序与数据集0中的输入数据一致	基于“数据集2”中的数据，视其为有符号数的补码，按从小到大排序后顺序存放的数据

场景2.用例编号	用例描述
3'b100	用数据集1中的最大值减去该数据集中的最小值，将结果显示在输出设备
3'b101	用数据集3中的最大值减去该数据集中的最小值，将结果显示在输出设备
3'b110	输入数据集编号（1或2或3）和该数据集中元素的下标(从0开始编址)，输出该数据的低8bit
3'b111	输入数据集中元素的下标，在输出设备上交替显示下面两组信息（间隔5秒）： 1) 数据集0的编号（0）、用户指定的元素下标、数据集0中该下标对应的元素的低8bit 2) 数据集2的编号（2）、用户指定的元素下标、数据集2中该下标对应的元素的低8bit