



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



CS214-2023s

CPU 大作业要求



总体说明

开发板领用说明： 每小组一块开发板, 请保护好开发板, 如有丢失或损坏需照价赔偿。
开发板借用**一周内完成基本测试**, 如有问题请反馈具体问题并及时找老师更换。

组队规则：

必须保证在**答辩时间内全员到场**。

建议同一个实验班的同学组队, 也可以同一个理论班下不同实验班的同学跨班组队。

3人组队 (如果人数不够也可以2人组队, 但不建议) 。

团队得分 = 功能验收分 * **系数** + 代码规范分 + 文档得分 + bonus得分

个人得分 = 团队得分 * 团队人数 * 个人贡献百分比

答辩 说明	代码提交	文档 (视频) 提交	团队得分-功能验收分- 系数
提前答辩 (15周实验课时间)	答辩前	16周周一前	1.05
正常答辩 (16周实验课时间)	答辩前	17周周一前	1
推迟答辩 (16周周五, 17周)	答辩前	答辩的后一周周一前	0.6~0.9

系数 说明: 如代码、文档 (视频) 任何一个提交件延迟提交, **系数** 将按最晚的提交时间为准计算“功能验收分”。
比如: A小组在15周完成答辩, 如所有交付件按照以上表格的要求准时提交, 则**系数为1.05**; 如代码准时提交但开发文档在16周周二提交, **系数** 按最晚提交的开发文档的时间来计算, 对应**系数为1**。

提交要求

- 提交要求（每小组只需要提交一份，分两次提交，两次提交都应是同一位组员）：

- **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（可选，仅不能完成上板测试的同学才必须做提交）、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**

评分说明 (1)

- 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。
- 项目得分包括两个部分：基本分+ bonus，如得分超过100，则溢出的部分将按比例计入总评。
 - 基本分：基本功能+ 代码规范 + 文档
 - 基本功能：
 - 1) 使用外设的种类数>2
(按键开关和拨码开关属于两种不同类型输入设备，led和七段数码显示管属于两种不同类型的输出设备)
 - 2) 测试通过基本场景1、基本场景2
 - 3) 实现只烧写一次FPGA芯片，可通过uart接口实现多个测试场景之间的切换
 - bonus：功能+文档+视频
 - bonus中合格的文档、视频的分数都按照以下方式计算：bonus功能分*0.25。比如：
bonus功能分拿到10分，合格的文档、视频的分数分别是 2.5分
 - **请注意：如缺少bonus对应的文档，除对应的文档分为0外，bonus的功能分要打八折。**
- 补充说明：如果不能上板测试，则根据情况，项目总分*(0.3~0.6)

评分说明 (2)

➤ **bonus 功能** 包括但不限于：

- 1) 实现对复杂外设接口的支持 (如VGA接口、小键盘接口等)
 - 说明：在本课程中，仅支持**通过软硬件协同的方式实现的复杂外设接口的访问** (即或者通过相应的指令，或者通过指令中相应的地址信息来访问相关的复杂外设，而不仅仅是以硬件控制的方式来实现对复杂外设的使用)。
- 2) 基于现有 Minisys ISA 实现的CPU，实现性能的优化 (pipeline、cache、SIMD等)
 - 说明：需同时提供性能提升的相关比对用例及证明。
- 3) 针对现有Minisys 的ISA，指令类型扩展、功能扩展和实现。
 - 或者可以考虑实现其他类型的ISA (如RISC-V，或融合了多种ISA的特点) 将按照以上要求进行检查 (该情况下，不重复考虑bonus的3部分)。 **请注意**，相关**文档中必须比对**该体系结构的实现与课上介绍的Minisys的实现之间的**差异**，否则相关**bonus为0分**。
- 4) 实现对中断的支持
- 5) 基于CPU的软硬件协同的应用
- 补充：所有实现的bonus部分，必须在**文档**中补充关于bonus的实现机制、测试用例以及测试结果的说明，并录制相应的视频对bonus对应的内容进行演示。如文档中没有相关说明，则对应的文档分数为0，且bonus对应的功能分*0.8。

文档要求 (1-基本分的相关文档)

- 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；寻址单位，指令空间、数据空间的大小。
 - 对外设IO的支持：采用单独的访问外设的指令（以及相应的指令）还是MMIO（以及相关外设对应的地址），采用轮询还是中断的方式访问IO。
 - 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，采用什么方式解决的流水线冲突问题）。
 - CPU接口：时钟、复位、uart接口、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、子模块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。

和bonus相关的文档及视频要求

➤ 和bonus相关的文档要求

- 和bonus相关的说明请放在基本功能文档的后半部分。
- bonus 对应功能点的设计说明
 - 设计思路及与周边模块的关系
 - 核心代码及必要说明
- 测试说明：测试场景说明，测试用例，测试结果及说明。
- 问题及总结：在bonus功能点开发过程中遇到的问题、思考、总结。

➤ 和bonus相关的视频要求：

- 视频中需要有本次大作业的完整介绍（包括小组成员，整体功能，尤其是bonus相关功能点）
- 主体内容为：bonus的设计思路介绍、功能演示及说明

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，两台电脑方便同步开展测试）。
- 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。
- 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/EGO1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（参考p5中“**bonus 功能**”部分）

基本测试场景1

说明：使用开发板上的3+8个拨码开关用于做输入，其中3个拨码开关(x2,...x0) 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，使用led灯或者7段数码显示管做输出。

场景1.用例编号	用例描述
3'b000	输入测试数a（仅识别a的最低7bit） ，输入完毕后在led灯上 显示a ，同时 用1个led灯显示a的奇校验位 （比如8'h01或8'h1F中的低7bit中有奇数个1，该led灯灭，8'ha1或8'h1b的低7bit中有偶数个1，该led灯亮）
3'b001	输入测试数a（识别a的完整8bit） ，输入完毕后在led灯上 显示a ，同时 用1个led灯显示a的奇校验结果 （比如8'h01或8'h1F中有奇数个1，该led灯亮，8'ha3或8'h1b中有偶数个1，该led灯灭）
3'b010	先执行测试用例3'b111 ，再计算 a 和 b的按位或非 运算，将结果显示在输出设备
3'b011	先执行测试用例3'b111 ，再计算 a 和 b的按位或 运算，将结果显示在输出设备
3'b100	先执行测试用例3'b111 ，再计算 a 和 b的按位异或 运算，将结果显示在输出设备
3'b101	先执行测试用例3'b111 ，再执行 sltu 指令，将 a和b按照无符号数进行比较 ，用输出设备展示a<b的关系是否成立（关系成立，亮灯，关系不成立，灭灯）
3'b110	先执行测试用例3'b111 ，再执行 slt 指令，将 a和b按照有符号数进行比较 ，用输出设备展示a<b的关系是否成立（关系成立，亮灯，关系不成立，灭灯）
3'b111	输入测试数a，输入测试数b，在输出设备上展示a和b的值

基本测试场景2

- 使用开发板上的3+8个拨码开关用于做输入，其中3个拨码开关（x3-x0）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）；

场景2.用例编号	用例描述
3'b000	输入a的数值（a被看作 有符号数 ），计算1到a的累加和，在输出设备上显示累加和（如果a是负数，以闪烁的方式给与提示）
3'b001	输入a的数值（a被看作 无符号数 ），以递归的方式计算1到a的累加和，记录本次入栈和出栈次数，在输出设备上显示入栈和出栈的次数之和
3'b010	输入a的数值（a被看作 无符号数 ），以递归的方式计算1到a的累加和，记录入栈和出栈的数据，在输出设备上显示入栈的参数，每一个入栈的参数显示停留2-3秒（说明，此处的输出不关注\$ra的入栈和出栈信息）
3'b011	输入a的数值（a被看作 无符号数 ），以递归的方式计算1到a的累加和，记录入栈和出栈的数据，在输出设备上显示出栈的参数，每一个出栈的参数显示停留2-3秒（说明，此处的输出不关注\$ra的入栈和出栈信息）
3'b100	输入测试数a和测试数b，实现 有符号数 （a，b以及相加和都是8bit，其中的最高bit被视作符号位，如果符号位为1，表示的是该负数的补码）的加法，并对是否溢出进行判断，输出运算结果以及溢出判断
3'b101	输入测试数a和测试数b，实现 有符号数 （a，b以及差值都是8bit，其中的最高bit被视作符号位，如果符号位为1，表示的是该负数的补码）的减法，并对是否溢出进行判断，输出运算结果以及溢出判断
3'b110	输入测试数a和测试数b，实现 有符号数 （a，b都是8bit，乘积是16bit，其中的最高bit被视作符号位，如果符号位为1，表示的是该负数的补码）的乘法，输出乘积
3'b111	输入测试数a和测试数b，实现 有符号数 （a，b，商和余数都是8bit，其中的最高bit被视作符号位，如果符号位为1，表示的是该负数的补码）的除法，输出商和余数（商和余数交替显示，每个结果显示持续5秒）