



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



CS202-2024s

CPU 大作业要求



总体说明 (重要!)

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**, 不到场则**自动减少组队人数**, 贡献比**按实际人数安排**。
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)。
(贡献比最小值和最大值不超过10%: 即二人组极限比例是45:55, 三人组极限比例是30:30:40)

个人Project总评 = 团队得分 * 团队人数 * 个人贡献百分比 + 个人答辩表现分(15)
团队得分 = 功能验收分(70) * **推迟/提前系数(0.6~1.05)** + 代码规范分(3) + 文档(12) + bonus(15)

答辩 说明	代码提交	文档 (视频) 提交	提前/推迟系数 (乘以功能验收分)
中期答辩 (13周实验课)	不涉及	不涉及	不涉及
提前答辩 (15周实验课)	15周周一中午12:00前	16周, 周一中午12:00前	1.05
正常答辩 (16周实验课)	16周周一中午12:00前	17周, 周一中午12:00前	1
推迟答辩 (所有延迟小组提交后统一安排)	迟一天扣5%系数	17周, 周一中午12:00前	16周周二0.9, 周三0.85, 周四0.8, 周五0.75, 周六0.7, 周日0.65, 17周周一0.6, 代码提交截止时间均为当天中午12:00

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**。
个人总评说明: 如三人组贡献比为1:1:1, 15周答辩, 则最高分可得 $(70*1.05+3+12+15)*3*0.33+15 = 118.5$

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（答辩当周的周一中午12：00之前提交bb站点）**

➤ 提交内容：测试场景对应的asm以及coe文件，CPU project目录（为避免压缩包尺寸过大，仅保留.xpr文件以及.srcs和.ip_user_files文件夹即可）。答辩时需从bb下载你提交的project，在inspector监督下现场生成bitstream文件，请提前测试好你的提交内容，确保答辩时能正确生成可通过测试的bitstream。个别用例测试不通过可现场调试，届时对应用例功能得分将乘以0.9。如果完全无法上板，请测试通过后重新提交代码，按照推迟系数折算分数。并重新预约答辩时间。

➤ 注意，课程组将于答辩后对所有verilog、asm代码进行查重

➤ 压缩包的名字格式为：**c答辩时间_小组成员姓名列表**

比如：c160156_A_B_C（其中c160156表示16周周一56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中午12：00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（只录bonus部分，没有实现bonus的小组无需提交）：

➤ 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（100分）+ bonus（15分），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(70) + 代码规范（3）+ 文档（12）+ 答辩分数(15,此分数为个人答辩表现分，不受贡献比影响)
 - 基本功能：
 - 1) 实现课本的单周期CPU（支持 RISC-V的 lw,sw,beq, add, sub, and, or），要求子模块通过OJ测试。（10）
 - 2) 基于1) 实现的单周期CPU做功能和指令集拓展，能够实现上板测试（60）
 - 上板测试的CPU代码中，对于指定模块的指定端口（具体模块和端口名参见附录xx），要求与OJ上测试提交的代码一致
 - 测试通过基本场景1、基本场景2
 - 按要求使用外设：按键(功能按钮如数据确认等)、拨码开关(数据输入)、**led(操作数展示)**和七段数码显示管(结果展示)，考虑到测试复杂度，数据输入、展示需考虑基本的用户体验，否则将酌情扣分。
 - bonus：15分
 - **请注意：bonus的实现应包含相应的代码、演示、文档及视频；**
 - **其中文档应就bonus相关的功能实现进行说明（包括实现机制、测试用例以及测试结果的说明，视频仅要求录制bonus的功能演示）**
 - **如缺少bonus对应的文档或者视频，bonus分数打6折。**
比如bonus功能得分10分，缺少合格的视频或者缺少文档，则 bonus 的总分 = $10 \times 0.6 = 6$ 分。
 - 补充说明：如果不能上板测试，则根据情况，项目总得分*(0.3~0.5)

评分说明 (2)

➤ **bonus 功能【max: 15】** 包括但不限于：

- 1) 实现对复杂外设接口的支持（如VGA接口、键盘接口等）【max: 2】
 - 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）。
 - 本课程bonus更偏重于CPU架构优化或应用场景实现，因此VGA或键盘或其他外设实现效果不作为bonus分数高低考量
- 2) 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【max: 2】
- 3) 基于实验课介绍的 RISC-V32I ISA 实现的单周期CPU，实现新的设计思路（如pipeline）【6-12】
 - 说明：需能够正确运行至少包含一个data hazard的代码片段，要求现场修改测试用的汇编代码并测试通过。（**基本测试用例请参考后续发布的说明文档**）
- 4) 实现现有RISC-V32I 的ISA中的 lui, auipc, ecall 需同时提供测试用例。【1-6】
- 5) 基于CPU的软硬件协同的实现或应用【2-10】
 - CPU软硬件协同的工具链中自创小工具，如支持auipc指令在当前CPU上落地的扩展汇编工具，可匹配生成ROM/RAM大小可调整的coe文件创建工具，uart速率可调整的硬件实现或者通信工具等。
 - 可以与CPU进行通信并配合的软件应用，如图形处理、声音处理、升级版游戏手柄等。
- **注意**：如果实现基于LoongArch指令CPU，则无需再实现RISC-V CPU, (功能分和bonus得分点平移到LA指令)，并根据完成情况在bonus得分基础上乘以1.05~1.1的系数（但bonus最高不超过15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RISC-V实现之间的差异，否则相关bonus为0分。**

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。
- 开发计划日程安排和实施情况，版本修改记录（可选）
- 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。
 - CPU时钟、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）。
 - 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，栈空间的基地址。
 - 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接口：时钟、复位、uart接口、其他常用IO接口说明。
 - 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（子模块端口规格及功能说明）
- 系统上板使用说明：开发板上与系统操作相关输入、输出操作说明。（如复位使用的输入设备、如何实现复位；CPU工作模式切换的按键及如何实现模式选择；输出信号的观测区域，与输出数据的对应关系等）
- 自测试说明：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（除本文及OJ以外的用例）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。
- 注意：以上内容均为评分点，请大家按照要求完成文档，另外文档不需要长篇大论，可以使用中文。

文档要求（2-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

最终答辩要求

➤ 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。
- **提醒：答辩时需从bb下载小组提交的project，在inspector监督下现场生成bitstream文件，请提前测试好你的提交内容，确保答辩时能正确生成可通过测试的bitstream。个别用例测试不通过可现场调试，届时对应用例功能得分将乘以0.9。如果完全无法上板，请测试通过后重新提交代码，按照推迟系数折算分数。并重新预约答辩时间。**
- 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（实现相对简单，占测试场景分值的 30%）

使用开发板上的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.（如果led灯的数目不够，则将输出在led灯上交替展示，每个部分展示停留2秒）（备注，拨码开关和led的高位统一在左侧，低位统一在右侧）

说明1：如果输入数据有多个（比如a， b），请先输入a，按确认键，再输入b

说明2：用led输出操作数的二进制形式，使用数码管（或者VGA）输出运算结果的十六进制形式。

场景1.用例编号	用例描述（用例3'b011-3'b111中， a， b均为二进制补码形式）
3'b000	输入测试数a，输入测试数b，在输出设备（led）上展示8bit的a和b的值
3'b001	输入测试数a，以lb的方式放入某个寄存器，将该32位的寄存器的值以十六进制的方式展示在输出设备上（数码管或者VGA），并将该数保存到memory中（在3'b011-3'b111用例中，将通过lw 指令从该memory单元中读取a的值进行比较）
3'b010	输入测试数b，以lbu的方式存入某个寄存器，将该32位寄存器的值以十六进制的方式展示在输出设备上（数码管或者VGA），并将该数保存到memory中（在3'b011-3'b111用例中，将通过lw 指令从该memory单元中读取a的值进行比较）
3'b011	用 beq 比较 测试数 a 和 测试数 b（来自于用例1和用例2），如果关系成立，点亮led，关系不成立，led熄灭
3'b100	用 blt 比较 测试数 a 和 测试数 b（来自于用例1和用例2），如果关系成立，点亮led，关系不成立，led熄灭
3'b101	用 bge 比较 测试数 a 和 测试数 b（来自于用例1和用例2），如果关系成立，点亮led，关系不成立，led熄灭
3'b110	用 bltu 比较 测试数 a 和 测试数 b（来自于用例1和用例2），如果关系成立，点亮led，关系不成立，led熄灭
3'b111	用 bgeu 比较 测试数 a 和 测试数 b（来自于用例1和用例2），如果关系成立，点亮led，关系不成立，led熄灭

基本测试场景2（实现相对复杂，占测试场景分值的 70%）

- 使用开发板上的拨码开关做输入，其中3个拨码开关（x3-x0）用于测试用例的编号输入，8~16个拨码开关用于做测试数据的输入（备注，拨码开关、led以及数码管的高位统一在左侧，低位统一在右侧）
- 说明2：用led输出操作数的二进制形式，使用数码管（或者VGA）输出运算结果的十六进制形式。

场景2.用例编号	用例描述（用例3'b001-3'b011的输入数据为有符号数）
3'b000	输入一个8bit的数，计算并输出其前导零的个数
3'b001	输入16bit位宽的IEEE754编码的半字浮点数，对其进行向上取整，输出取整后的结果
3'b010	输入16bit位宽的IEEE754编码的半字浮点数，对其进行向下取整，输出取整后的结果
3'b011	输入16bit位宽的IEEE754编码的半字浮点数，对其进行四舍五入取整，输出取整后的结果
3'b100	分两次输入两个8bit的数a和b，对a，b做加法运算，如果相加和超过8bit，将高位取出，累加到相加和中，对相加和取反后输出
3'b101	输入12bit的数据，以小端模式从拨码开关输入，以大端的方式呈现在输出设备上
3'b110	以递归的方式计算小于输入数据的斐波拉契数字的数目，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'b111	以递归的方式计算小于输入数据的斐波拉契数字的数目，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）