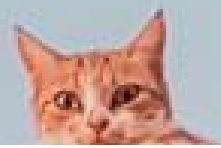

Computer Orgnization

CS202-2025s　　CPU 大作业要求

总体说明1（重要!）

开发板领用说明：
1）**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。**
2）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：
1）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，由实际人数分配100%贡献比。**
2）**建议同一个实验班的同学组队**，也可以跨班组队（由于答辩时必须全员到场，因此不建议跨班组队）。
3）建议3人组队（特殊情况也可以2人组队，但不建议）。
4）**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。

拆组重组说明：
1）如小组因特殊原因需要**拆组，经小组所有成员同意**（建议当面交流，或在小组群里进行确认），可以拆组后再重组。由一名成员撰写邮件，说明拆组原因，主送给实验课老师并抄送给所有成员。

2）**拆组的截止时间为项目答辩前一周**，超过该时间后不再支持拆组重组，请同学慎重。

3）拆组后的重组由同学自行协商，重组后的结果请主送实验课老师并抄送给小组所有成员。

作业诚信：如本次作业中使用了网络代码或者AI生成代码，请在务必进行说明（比如哪部分代码参考了网络实现，给出网址，哪部分使用了AI，给出AI工具名和提示词），如不做说明，则代码查重时若发现重复度高将直接判定为作业不诚信，所有功能分数（答辩功能验收分、bonus分）将为0分。

总体说明2（重要!）

- ✓ **得分超过100，则溢出的部分将按比例计入总评.**
- ✓ **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 = 答辩功能验收分(70) * 推迟/提前系数(0.6~1.05) +代码规范分(3) + 文档(12) + bonus(15)**
- ✓ **个人答辩表现分 = 中期答辩分（10） + 最终答辩表现分（5）**
  - ✓ **说明: 如未参加中期答辩，则个人中期答辩分为0，如未参加项目最终答辩，则按p2组队规则的约定，自动退出小组。**

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	现场提交中期答辩自述文档、ppt，现场个人ppt介绍进度（3分钟内）	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 晚于17周周一中午12：00得0分	17周，周一中午12:00前 晚于17周周一中午12：00得0分	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17周周一中午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。**A小组延迟答辩，但未在17周周一中午12：00前完成代码和文档的提交，则**答辩功能验收分为0分。**

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+10（中期答辩分）+5（最终答辩分） = 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，负责提交的组员信息需要在共享文档中登记）：
  - **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，请删除.runs子目录后再打包上传)。**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。**
    - **注意：上板测试答辩只有一次，请确保可以上板测试再预约答辩时间。个别用例测试不通过可现场调试，届时对应用例功能点得分将乘以0.5。如果答辩时完全无法上板，将按照仿真用例的测试评分。仿真测试得分=上板测试得分*0.3，需自行提供仿真testbench。**
    - **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（17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）：
      - 请分开上传**pdf**文档和**mp4**视频：分别命名为**d答辩时间_小组成员姓名列表，v答辩时间_小组成员姓名列表**
      - **视频大小建议不超过500M（如超过该大小，请自行处理后再上传），视频中必须包含bonus上板演示的部分。**

评分说明（1-1） 基础分

- 基本分：基本功能(70) + 代码规范（3）+ 文档（12）+个人答辩表现分(15分=10分中期答辩+5分最终答辩)
  - 中期答辩（10）：
    - 1）单周期CPU基本组件OJ得分（2）
    - 2）中期答辩自述文档（4）：参见p9的中期答辩要求，请认真填写，缺一项或随意填写该项扣0.5分，最多扣4分。
    - 3）现场3分钟答辩分（4）：重点部分代码展示+ 子模块测试/ 集成测试结果展示，根据答辩表现给分。
    - 注意！！：中期答辩需个人参加，未到场的同学中期答辩得0分。
  - 基本功能（70）：
    - 实现的单周期CPU做功能和指令集拓展，能够实现上板测试 （70）
      - 上板测试的CPU代码中，要求遵守中期答辩自述文档中约定的代码规范（结构化设计，统一的命名方式、注释要求、符号化常量的定义及使用等)
      - 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(结果展示)和七段数码显示管(结果展示)，如没有按照题目要求使用IO，则扣除该用例的50%分数。请注意，IO的地址不能使用lab课件提供的地址，需要自行指定。
      - 如果不能上板测试, 现场修改代码上板测试通过，则用例得分*0.5；如不能上板测试，采用仿真测试，仿真测试用例得分*0.3
  - 最终答辩（5）：
    - 无需ppt，以问答形式展开。评委将使用中期答辩材料核对实现方案，如果方案不符，需给出充分理由说明。
- 代码规范说明（3）：
  - 应在项目开发初期即确定项目整体的代码规范（如结构化设计、统一的命名方式、注释要求、符号化常量的定义及使用等），该项约定应在中期答辩自述文档中体现，并在开发过程中严格执行。
  - 如中期答辩制定的代码规范未在最终答辩的代码中体现，则答辩时的代码规范要扣分、中期答辩自述文档中的代码规范约定分为0.

评分说明（1-2）基础分-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。
- 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IO接口说明。
- 方案分析说明：对于同一功能实现中的硬件方案和软件方案分别进行分析，通过实验的方式进行功能、性能等方面的比较，给出比较结果以及最终选择的方案说明。比如实现浮点数运算在硬件和软件上的差异。（5分）
- 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）（1分）
- 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。（1分）
- 开源及AI对于本次大作业的启发和帮助：如开发过程中使用了网络代码资源或者借助AI生成了相关代码，请说明这些代码提供了哪些帮助和启发，遇到了哪些问题及对应的解决方案。（2分），如完全自研（未参考任何网络资源或者AI生成代码），请基于实验课件方案进行相关说明。
- 问题及总结：开发过程中遇到的问题、思考、总结。（3分）
- 注意：请大家按照要求完成文档，文档不需要长篇大论，可以使用中文。

评分说明（2-1）bonus分

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）**实现的ISA指令扩展**，如 auipc（1分），ecall（每实现一个ecall功能得1分，上限2分)等。 需自行提供测试用例,并说明实现该扩展指令相比于课上CPU结构所做修改，如无此类说明则不得分。**【max: 5】**
- 4）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）**【max: 10】**说明:
  - 如实现pipeline，也需要实现单周期CPU。需基于同一测试用例(如循环)展示CPU性能提升。【1~4】
  - 需能够通过debug模式暂停、控制时钟周期行进，以便实时观察寄存器值和hazard解决情况【2】
  - 正确运行包含control hazard，一个或多个data hazard的代码片段，自行提供测试用的汇编代码并测试通过。【1~4】
- 5）基于CPU的**软硬件协同的实现或应用【max: 5】**
  - CPU软硬件协同的工具链中自创小工具，如支持自定义扩展指令在当前CPU上落地的扩展汇编工具（1分），可匹配生成ROM/RAM大小可调整的coe文件创建工具（1分），uart速率可调整的硬件实现（1分）或者通信工具（1分）等。
  - 可以与CPU进行通信并配合的自创软件应用，如图形处理、声音处理、升级版游戏手柄等（1~4分视复杂度而定）。
- **注意**:

  1) 如果实现基于LoongArch 或者ARM指令CPU，则无需再实现RISC-V CPU,(功能分和bonus得分点平移到LoongArch指令/ARM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。**

  2）**bonus的实现应包含相应的代码、演示、文档及视频；**

  3）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，视频要求录制bonus的功能演示）

  4）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*0.6 = 6分。

评分说明（2-2）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。
  - 开源及AI对于本次大作业的启发和帮助：如开发过程中使用了网络代码资源或者借助AI生成了相关代码，请说明这些代码提供了哪些帮助和启发，遇到了哪些问题及对应的解决方案。
  - 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

中期答辩要求

- 答辩准备
  - ✓ 提前准备”**小组中期答辩自述文档**“（打印，A4一页双面打印），该文档应包含：
    - ① 小组信息（成员名、学号、实验课时间）
    - ② 代码规范约定（应覆盖是否开展结构化设计、命名规范、注释要求、符号化常量的定义及使用等）
    - ③ CPU特性说明（单周期/pipeline），时钟，支持的指令集合
    - ④ CPU架构设计（内部模块及连线关系）/接口设计说明
    - ⑤ CPU指令与控制信号的关系
    - ⑥ 已完成的项目代码以及已搭建的测试场景
    - ⑦ 计划使用/开发的工具链
    - ⑧ 项目进度、当前困难或问题、预计最终答辩时间、后续计划
  - ✓ 提前准备”**个人答辩ppt**“（电子版+A4纸双面打印）
    - ① 个人姓名学号和所负责工作简述
    - ② 个人所负责代码重点内容截图
    - ③ 个人所负责测试
      - ① 给出测试表格（含用例说明-比如相关指令、测试结果（测试通过或不通过）
      - ② 测试用的testbench截图、测试波形截图及必要说明。

- 答辩：
  - ✓ 时间：**13周实验课**
  - ✓ 答辩次序登记：在共享文档中登记答辩时间、答辩次序。
  - ✓ 所有成员必须到场，不到场成员中期答辩分为0分。
  - ✓ 每位同学提供 “小组中期答辩自述文档”（填一份，小组成员复印即可）及”个人答辩ppt“（A4双面打印，和文档装订在一起），同时使用电子版ppt参加答辩
  - ✓ 个人答辩：ppt展示个人负责的工作、代码及所作测试（测试用例表格、测试结果-波形截图及必要说明），时间严格控制在**3分钟**内。
  - ✓ 回答老师/sa关于项目及个人工作的问题
- 说明：
  - 中期答辩自述+个人答辩ppt将在完成中期答辩后上交给答辩评审人，作为最终项目答辩的参考文档

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0.5。注意：上板测试答辩只有一次，请确保可以上板测试再预约答辩时间。如果答辩时无法上板，将按照仿真用例的测试评分。仿真测试得分=上板测试得分*0.3。**
- 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p7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 20%）

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（备注，拨码开关、led、数码管的高位统一在左侧，低位统一在右侧）

说明1：请按照用例描述中的说明进行输入输出，不按指定要求完成的用例得分=用例分*0.5
说明2：如果实现了其他的输入（键盘）或输出（VGA），测试bonus时选择本用场景中的一个用例做单独展示，但仍需实现文档要求的基本外设的输入输出。

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(0)	输入测试数a，按确定后在输出设备（8个led）上展示a的值，输入测试数b，按确定后在输出设备（8个led）上展示b的值
3‘b001(1)	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），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(2)	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），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b的值进行比较）
3‘b011(3)	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8个led，关系不成立，8个led都熄灭
3‘b100(4)	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8个led，关系不成立，8个led都熄灭
3‘b101(5)	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8个led，关系不成立，8个led都熄灭
3‘b110(6)	用 slt 比较 测试数 a 和 测试数 b（来自于用例1和用例2），通过**store**指令将比较结果输出到led，如果关系成立，点亮1个led，关系不成立，该led熄灭
3‘b111(7)	用 sltu 比较 测试数 a 和 测试数 b（来自于用例1和用例2），通过**store**指令将比较结果输出到led，如果关系成立，点亮1个led，关系不成立，该led熄灭

基本测试场景2（实现相对复杂，占测试场景分值的 80%）

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（备注，拨码开关、led、数码管的高位统一在左侧，低位统一在右侧）

说明1：请按照用例描述中的说明进行输入输出，不按指定要求完成的用例得分=用例分*0.5

说明2：如果实现了其他的输入（键盘）或输出（VGA），测试bonus时选择本用场景中的一个用例做单独展示，但仍需使用基本外设实现输入输出。

场景2.用例编号	用例描述（用例3‘b010-3’b011的输入数据为有符号数）
3‘b000	输入一个8bit的数，将其倒序后输出，比如输入8‘b1010_1001, 未倒序时输出应为8‘b1010_1001，倒序后的输出为8’b1001_0101（在led灯上输出）
3‘b001	输入一个8bit的数，判断其是否是二进制回文（参见lab3的practice2），是回文点亮一个led灯，不是回文该led灯灭
3‘b010	先后输入两个12bit位宽的IEEE754编码的浮点数（最高bit位是符号位，指数3bit，位数4bit）的高8位（低4位默认为0），将这两个浮点数存储到memory中，第一个浮点数a输入完毕后按确认在数码管上显示a的整数部分的十进制形式，第二个浮点数b输入完毕后按确认在输出设备上显示b的整数部分的十进制形式。
3‘b011	将本场景中用例编号3‘b010中的两个测试数据从momory中读出，对两个测试数据做加法运算，将相加和的整数部分以十进制的方式显示在数码管上。
3‘b100	基于输入的4bit原数据，生成其对应的循环冗余校验码(CRC-4: X^4+X+1），并将原数据与校验码拼接起来（元数据在高位，校验码在低位）比如原数据为4‘b1000，基于CRC-4生成的校验码为**4’b1011**, 最终输出**10001011**. 在8个led上展示结果的二进制形式。
3‘b101	根据输入的8bit数据进行(CRC-4: X^4+X+1）校验，通过1个led灯展示校验通过还是未通过。 **比如输入8’b10001011, 校验通过，1个led灯亮，输入8’b10001010, 校验不通过，该led灯灭。**
3‘b110	请自行设计用例，用于检测 lui 指令是否生效。用数码管做输出。(如实现的其他ISA中未设计该指令，考虑用其他的方式实现类似的功能并测试)
3‘b111	请自行设计用例，用于检测jal 以及 jalr 指令是否生效。(如实现的其他ISA中未设计该指令，考虑用其他的方式实现类似的功能并测试)