# Lab2-RV31I实验指导和实验报告要求

* 用verilog实现RV32I 四级流水线CPU。
* **实验工具：Vivado**
* **实验方式：Vivado自带的波形仿真**

### 设计要求：

1. 参考4\_ProjectDesignFiles 文件夹中提供的CPU模块图和1\_PipelineLab文件夹中对应的Verilog代码，思考每条指令的数据通路，详细写出每个待完成模块的设计思路，并思考如何用verilog简洁高效的表达这些逻辑电路。
2. 待实现指令：
   1. RISC-V 32bit 整型指令集（除去FENCE,FENCE.I,CSR,ECALL和EBREAK指令）
   2. 可参考官方提供的RISCV用户指令集手册（中文版或者英文版），本文件夹内提供了
   3. 也可以参考<https://www.cnblogs.com/mikewolf2002/p/9864652.html> （注意本网站有些错误，对于branch指令的功能描述不是很准确，auipc指令描述有问题，jalr是I型指令不是UJ型）
3. 设计原则：参考我们提供的CPU模块图，修改补全模块内部逻辑完成CPU设计；也可以根据你的个人理解进行设计，并在报告中提出你自己的改进方案并给出修改原因。

### 测试要求：

**PART1:**

1. 自己手写合适的测试用汇编代码，通过提供的工具生成.inst和.data文件，用于初始化指令和数据的Block Memory，或者直接手写二进制测试代码
2. 测试用的指令流中需要包含的指令包括SLLI、SRLI、SRAI、ADD、SUB、SLL、SLT、SLTU、XOR、SRL、SRA、OR、AND、ADDI、SLTI、SLTIU、XORI、ORI、ANDI、LUI、AUIPC
3. 测试例（汇编和对应的.inst .data）可以用其他同学提供的，但是需要自己知道对应的指令逻辑，需要在报告中能清楚表达这个测试例如何验证CPU功能正确，即正确运行后寄存器值应该是多少
4. CPU执行后，各寄存器值符合预期
5. 此时不需要处理数据相关。可以令Harzard模块始终输出stall、flush恒为0，forward恒为不转发，每两条指令之间间隔四条空指令。

**PART2:**

1. 自己手写合适的测试用汇编代码，通过提供的工具生成.inst和.data文件，用于初始化指令和数据的Block Memory，或者直接手写二进制测试代码
2. 测试用的指令流中需要包含的指令包括**JALR、LB、LH、LW、LBU、LHU、SB、SH、SW、BEQ、BNE、BLT、BLTU、BGE、BGEU、JAL**
3. 测试例（汇编和对应的.inst .data）可以用其他同学提供的，但是需要自己知道对应的指令逻辑，需要在报告中能清楚表达这个测试例如何验证CPU功能正确，即正确运行后寄存器值应该是多少
4. CPU执行后，各寄存器值符合预期
5. **此时不需要处理数据相关。可以令Harzard模块始终输出stall、flush恒为0，forward恒为不转发，每两条指令之间间隔四条空指令。**

**PART3：**

1. 利用我们提供的修改过得RISCV test官方测试汇编代码，通过提供的工具生成.inst和.data文件，用于初始化指令和数据的Block Memory
2. CPU开始执行后3号寄存器的值会从2一直累增，该数字正在进行第多少项测试，执行结束后3号寄存器值变为1
3. CPU执行后，各寄存器值符合预期
4. **此时需要处理数据相关，实现Harzard模块内部逻辑。**

注意：三部分为递进关系，建议大家按照顺序进行测试

### 实验报告：

* 实验目标
* 实验环境和工具
* 实验内容和过程（总结自己所做的三个阶段工作）
* 实验总结（说说踩的坑，总结收获，分析下自己花了多少时间，都用来做什么事情）
* 提出改进实验的意见

**实验总结是每位同学写自己的，其他部分同一个小组写一份，合并在一个报告里面即可，最后整个小组上交一份报告，报告最后写上分数分配。**