RV32I 的 CPU 设计与实现

班级：计科 1601
学号：201611020126
姓名：蒋雨

一、实验目标

完成一个模拟 RISC-V 的基本整数指令集 RV32I 的 CPU 设计（单周期实现）。

二、实验要求

硬件设计采用 VHDL 或 Verilog 语言。

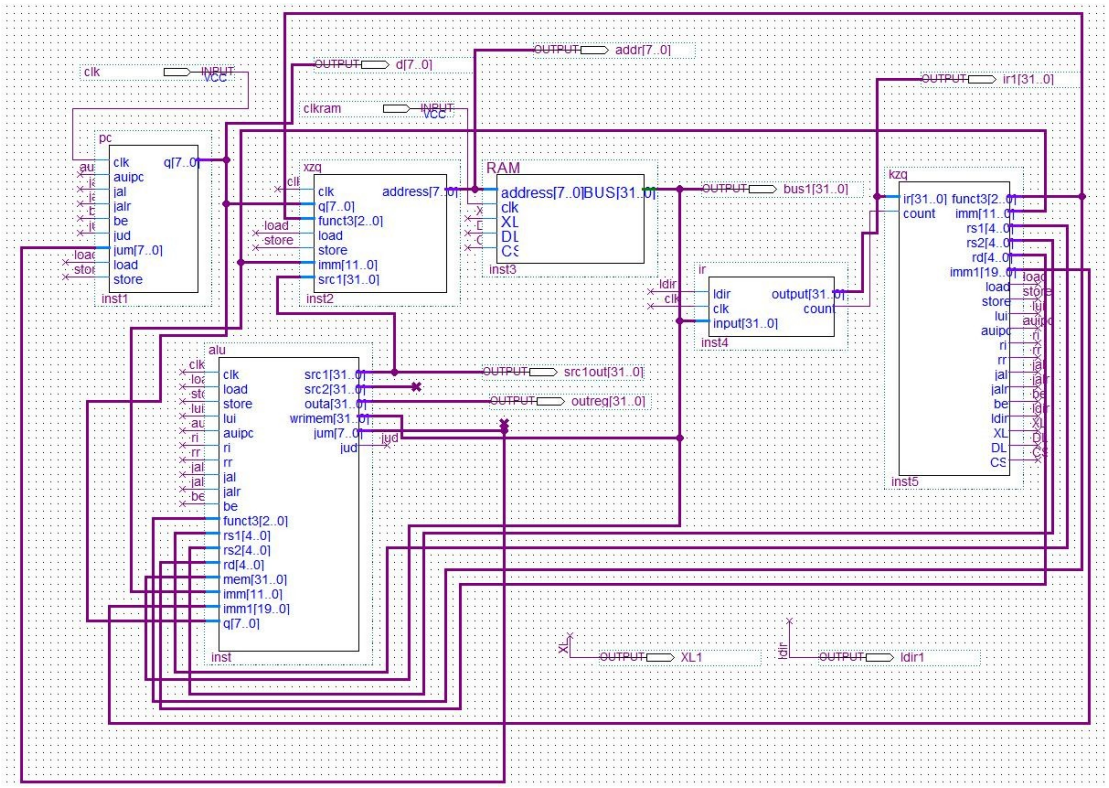
三、实验设计

单周期 CPU 指的是一条指令的执行在一个时钟周期内完成，然后开始下一条指令的执行，即一条指令用一个时钟周期完成。
CPU 在处理指令时，一般需要经过以下几个步骤：

取指令(IF)：根据程序计数器 PC 中的指令地址，从存储器中取出一条指令，同时，PC 根据指令字长度自动递增产生下一条指令所需要的指令地址，但遇到“地址转移”指令时，则控制器把“转移地址”送入 PC，当然得到的“地址”需要做些变换才送入 PC。
指令译码(ID)：对取指令操作中得到的指令进行分析并译码，确定这条指令需要完成的操作，从而产生相应的操作控制信号，用于驱动执行状态中的各种操作。
指令执行(EXE)：根据指令译码得到的操作控制信号，具体地执行指令动作，然后转移到结果写回状态。
存储器访问(MEM)：所有需要访问存储器的操作都将在这个步骤中执行，该步骤给出存储器的数据地址，把数据写入到存储器中数据地址所指定的存储单元或者从存储器中得到数据地址单元中的数据。
结果写回(WB)：指令执行的结果或者访问存储器中得到的数据写回相应的目的寄存器中。

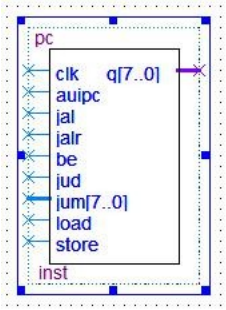
单周期 CPU，是在一个时钟周期内完成这五个阶段的处理。

1、设计的整体架构

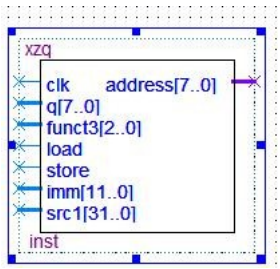


2、各模块的具体实现

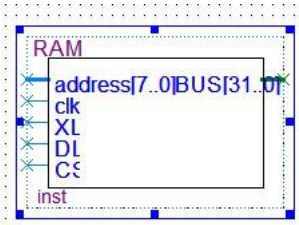
1. PC指令计数器，用于存放下一条指令所在单元的地址的地方。

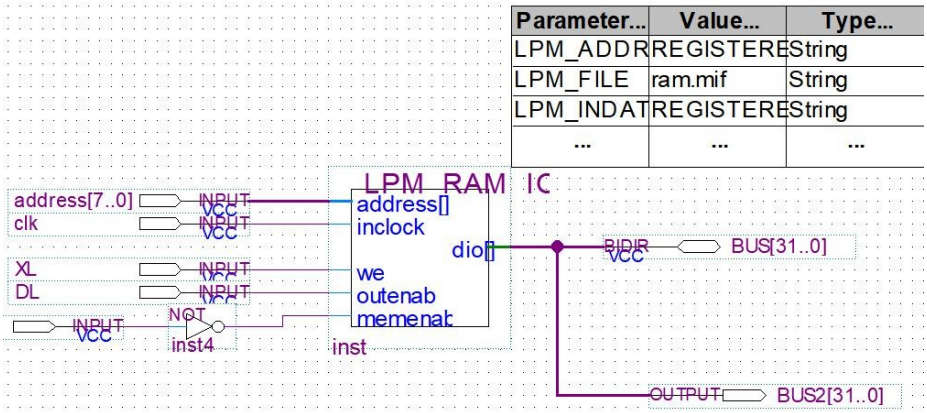


2. 选择器，根据给定的输入地址代码，从一组输入信号中选出指定的一个送至输出端。

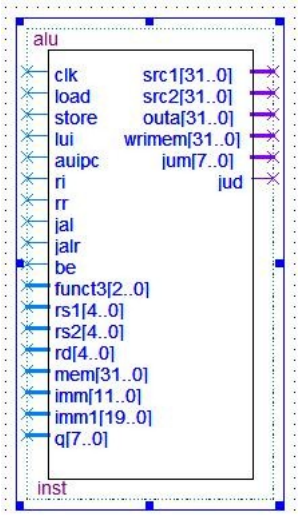


3. RAM 数据存储单元。

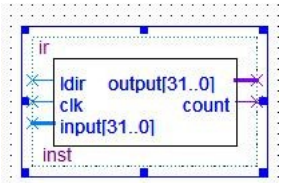




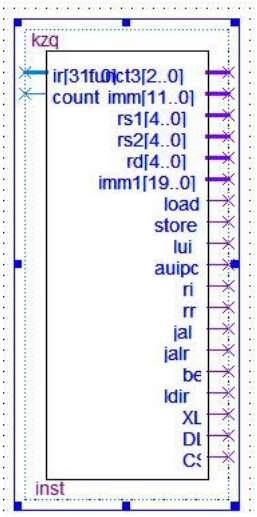
4. ALU 算术逻辑单元，实现算术运算和逻辑运算。



5. IR 指令寄存器，接收 BUS 总线上的信号输入，接入时钟控制信号以及 ldir，LDIR 控制是否把 BUS 总线上的数据打入寄存器 IR 中。



6. 控制器，根据指令各个字段的内容为数据通路提供所有的控制字。控制字字段的数目可以从指令字段的内容中直接获取。



四、测试

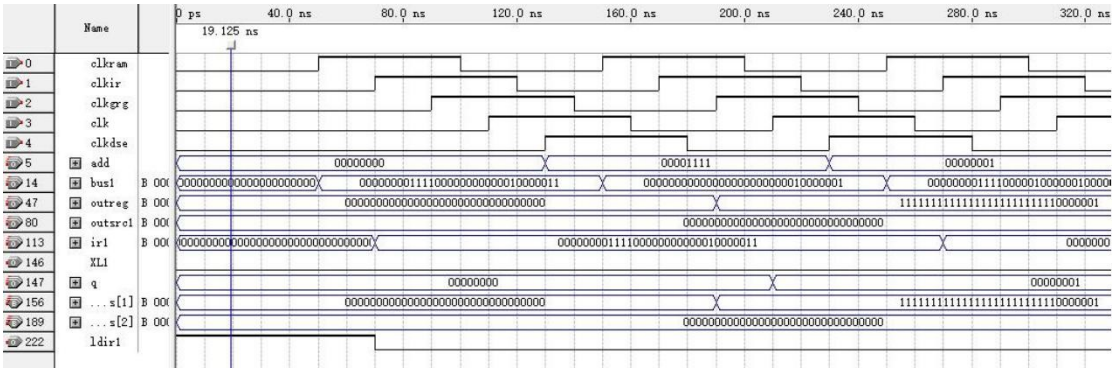
1、测试环境

部件	配置
CPU	Core i5-6200U
内存	DDR3 4GB
操作系统	Windows 10

2、测试结果

1. load/Store 指令

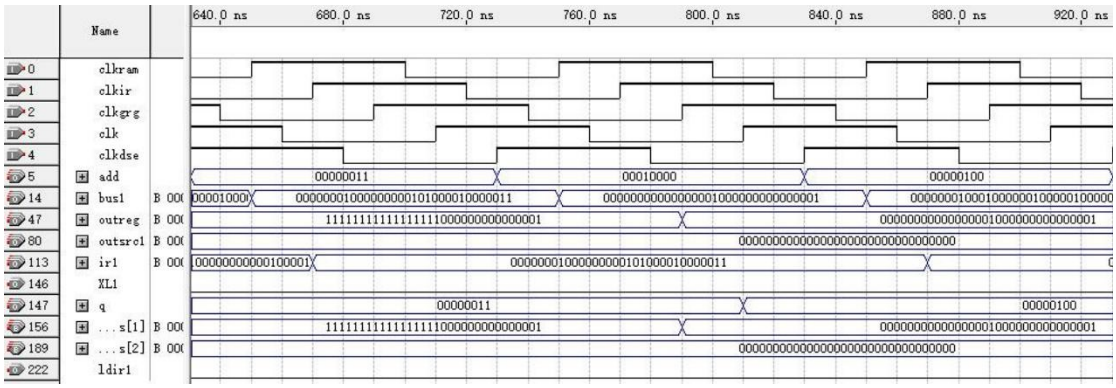
lb指令
测试所用的二进制指令为：000000001111 00000 000 00001 0000011
仿真结果：



分析：这条指令是从地址 x[0]加立即数中的一个字节，经符号位扩展后写入 r1中，该数值

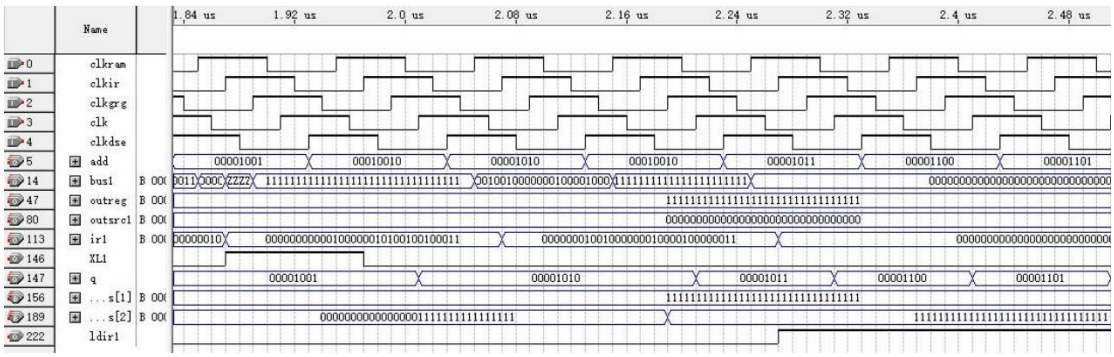
后八位为 10000001，符号位为 1，扩展后结果为 11111111111111111111111110000001，与仿真结果一致。

lhu指令
测试所用的二进制指令为：000000010000 00000 101 00001 0000011
仿真结果：



分析：这条指令是从地址 x[0]+sign-extend(offset) 读取两个字节，经零扩展后写入 x[1]，该初值为 00000000000000001000000000000001，零扩展后数值不变，与仿真结果一致。

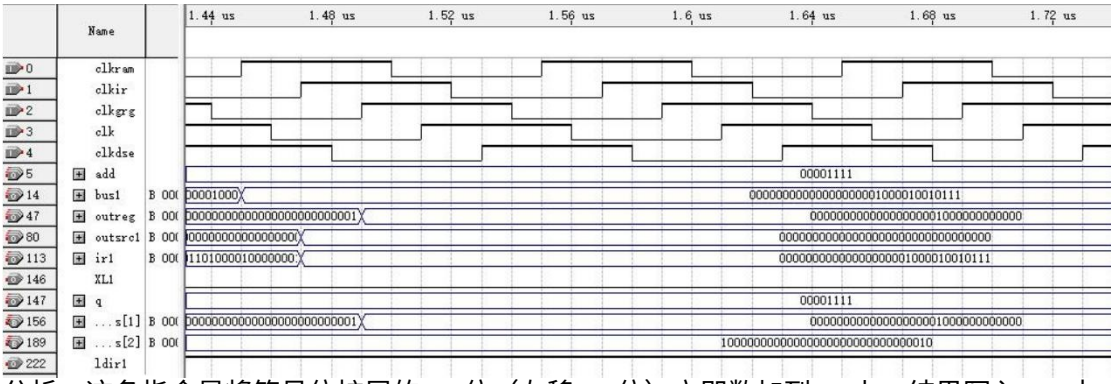
sw指令
测试所用的二进制指令为：0000000 00001 00000 010 10010 0100011
仿真结果：



分析：这条指令是将 x[1] 的低位 4 个字节存入内存地址 x[0]+sign-extend(offset) 中，由波形图可知，该寄存器中的数值为 11111111111111111111111111111111，存入 mem[18] 中，再执行一条 lw 指令(00000001001000000010000100000011)，它能读取 mem[18] 中的数值并存入 x[2] 中，读出的结果与寄存器内的值一致。

2. 整数计算指令

auipc指令
测试所用的二进制指令为：00000000000000000001 00001 0010111
仿真结果：



分析：这条指令是将符号位扩展的 20 位（左移 12 位）立即数加到 pc 上，结果写入 x[1] 中。后面的地址一直为 00001111，测试结果正确。

3. 控制指令

在指令测试前，执行 3 条 lw 指令，将 mem[17] mem[18] mem[19] 的数据存储到 reg[1] reg[2] reg[3] 中。这三个数值为：

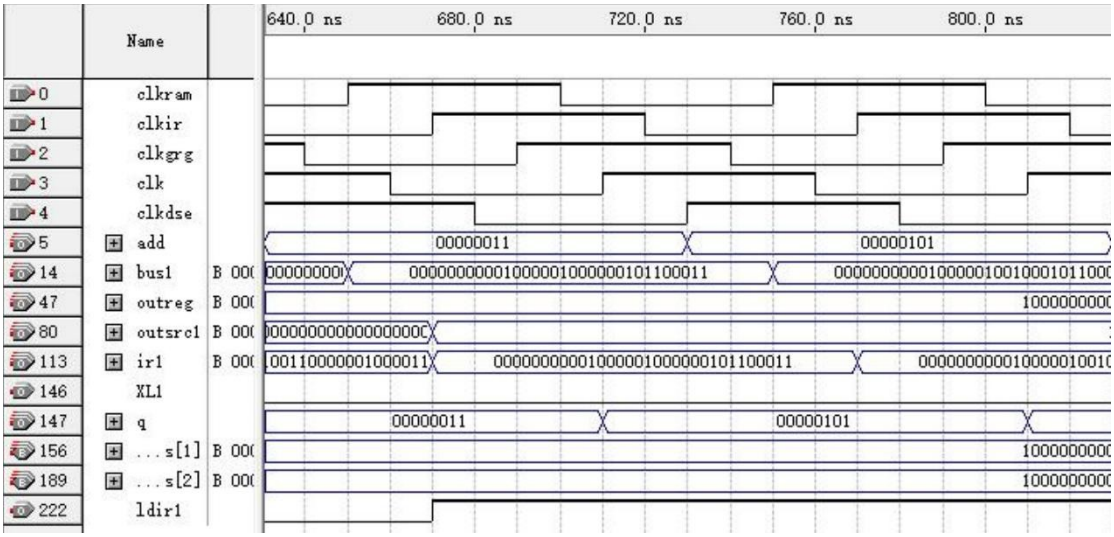
mem[17]:10000000000000000000000000000010

mem[18]:10000000000000000000000000000010

mem[19]:00000000000000000000000000000001

beq 指令

测试所用的二进制指令为：0000000 00010 00001 000 00010 1100011

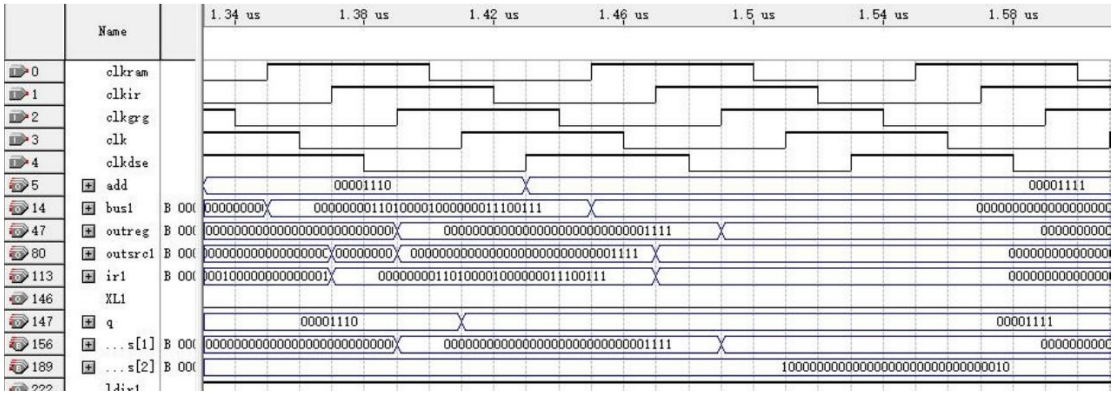


分析：该指令是相等时分支跳转，若寄存器 x[1] 和寄存器 x[2] 的值相等，把 pc 的值设为当前值加上符号位扩展的偏移 offset 此时两个寄存器中的值相等，所以跳转到地址为 5 的位置，由图知跳转到 0000010 处，结果正确。

jalr 指令

测试所用的二进制指令为：000000001101 00001 000 00001 1100111

仿真结果：



分析：将 pc 设置为 x[1]+sign-extend(offset)，把计算出的地址的最低有效位设为 0，并将原 pc+4 的值写入 x[1]中，由图可知下一个地址为 0000111,1仿真结果正确。

五、实验总结

对 vhdl 语言有了更多的了解，硬件设计过程中时钟的作用非常大。通过阅读教材，了解了数据通路的建立过程，各个模块之间的联系。分开设计各个模块的功能比较清晰，也容易检测错误出现的地方。只测试了一小部分指令，并没有全部完成。