

ALU 设计

设计者学号：14241217
设计者姓名：张聿恩

1. 指定的指令如下

指令	类型	opcode	function	含义
addu	R	00000	100001	将寄存器 rs 的值与寄存器 rt 的值相加，结果写入 rd 寄存器中。
sub	R	00000	100010	将寄存器 rs 的值与寄存器 rt 的值相减，结果写入 rd 寄存器中。不考虑溢出。
slt	R	00000	101010	将寄存器 rs 的值与寄存器 rt 中的值进行有符号数比较，如果寄存器 rs 中的值小，则寄存器 rd 置 1；否则寄存器 rd 置 0。实现的时候可以实现位 rs-rt, 将符号位送 rd 的第 0 位。
lui	I	001111	无	将 16 位立即数 imm 写入寄存器 rt 的高 16 位，寄存器 rt 的低 16 位置 0。
ori	I	001101	无	寄存器 rs 中的值与 0 扩展至 32 位的立即数 imm 按位逻辑或，结果写入寄存器 rt 中。
bne	I	000101	无	如果寄存器 rs 的值不等于寄存器 rt 的值则转移，否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。实现的时候可以用 ALU 做减法，然后用是否为 0 标志位进行控制。
lw	I	100011	无	将 base 寄存器（rs 的位置）的值加上符号扩展后的立即数 offset 得到访存的地址，据此地址从存储器中读取连续 4 个字节的值，写入到 rt 寄存器中。注意：测试程序中给出的地址是对齐的，所以不用考虑地址不对齐触发例外。
sw	I	101011	无	将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址，如果地址不是 4 的整数倍则触发地址错例外，否则据此虚地址将 rt 寄存器存入存储器中。注意：测试程序中给出的地址是对齐的，所以不用考虑地址不对齐触发例外。
stp	其他	11111	无	所有控制信号清 0，时钟输入终止。如果是五段流水，要求 stp 指令也为五段，也就是 stp 指令退出流水线后，才能停止时钟输入。

2. 根据指令，对 ALU 的功能进行总结。如果某指令不需要 alu，可以填写“无”。

指令功能总结	功 能	说明
addu	加法	计算 $rs + rt$
sub	减法	计算 $rs - rt$
slt	比大小	计算 $rs - rt$ ，小于 0 ($rs < rt$) 输出 1， 否则输出 0
lui	放高 16 位（立即数）	把输入数放到高 16 位 ($num \ll 16$)，下面置零
ori	或（立即数）	计算逻辑或
bne	是否相等	计算 $rs - rt$ 来判断是否相等
lw	计算地址，并取出内容	Load word，计算 $Base + Offset$ ， 并从该地址取出数据
sw	计算地址，并存入内容	save word，计算 $Base + Offset$ ， 并从该地址存入数据
stp	无	清零，停止时钟

3. 根据上述归纳总结的功能，设计 ALU 的控制码，可以采用独热码，也可以采用二进制编码。

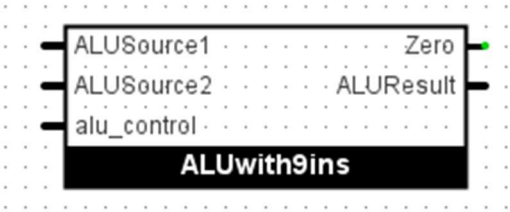
功能	编码
加法	000
减法	001
逻辑或	010
比大小	011
高位加载	100

4. 完成 ALU 模块的设计。

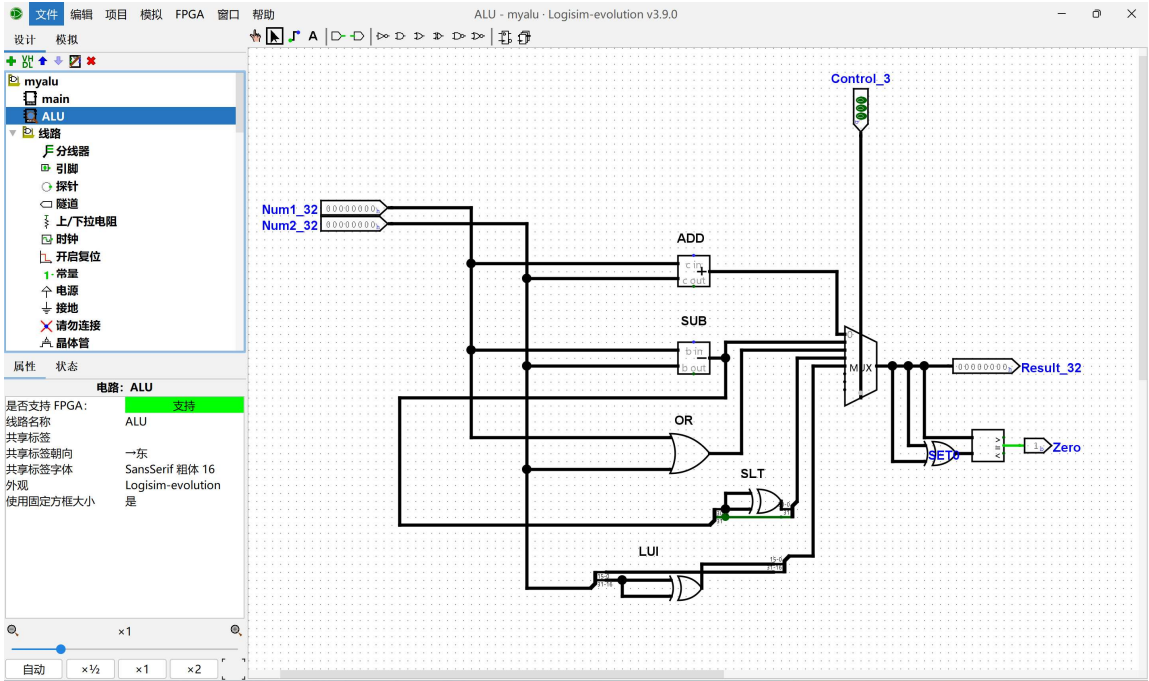
第一步，定义了一个 ALUwith9ins 的模块，先把输入输出定义好，如：

输入	输出
ALUSource1 (32 位) ALUSource2 (32 位)	ALUResult (32 位) Zero (1 位。ALUResult 为 0 时，输出 1)

在 main 中，把这个模块拖过来，就是这个样子：



第二步，按照任务二总结的功能，完成 ALUwith9ins 模块，截图如下：

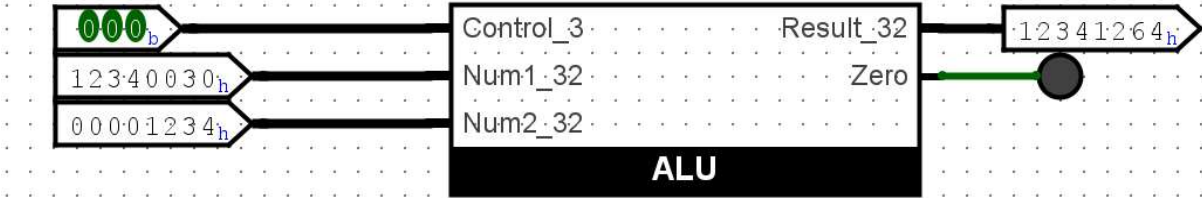
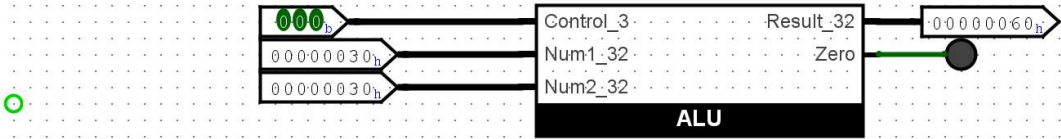


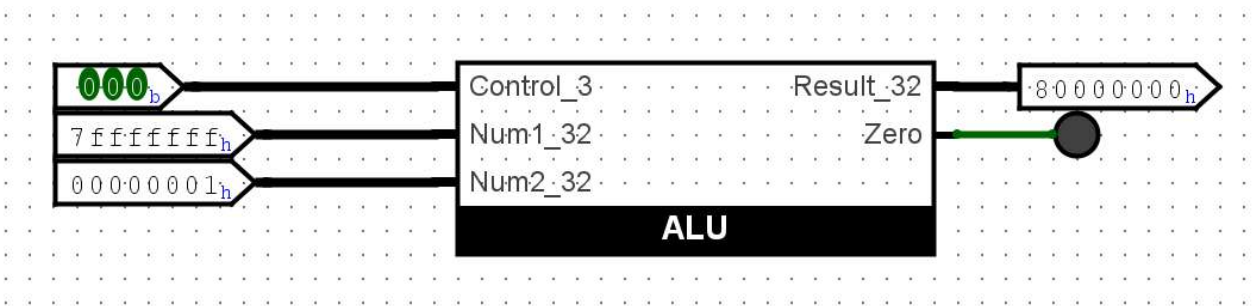
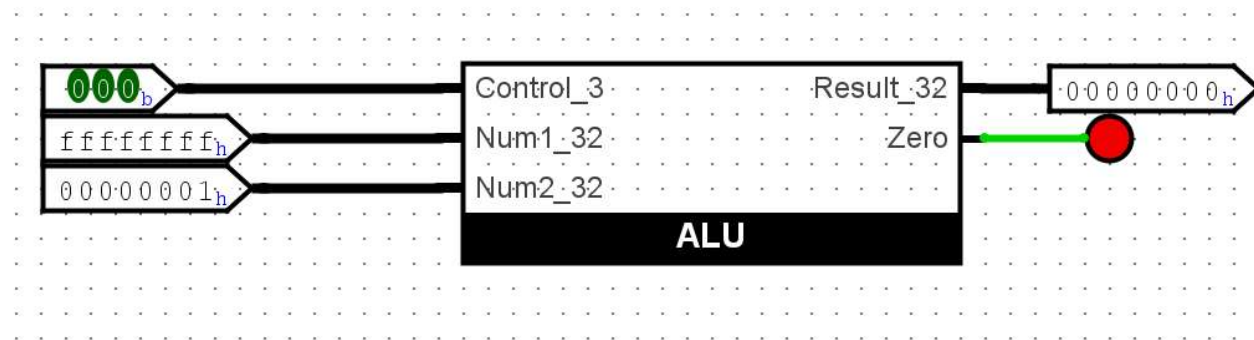
(请将设计的截图插入)

5. 请将各个功能测试的截图列出。

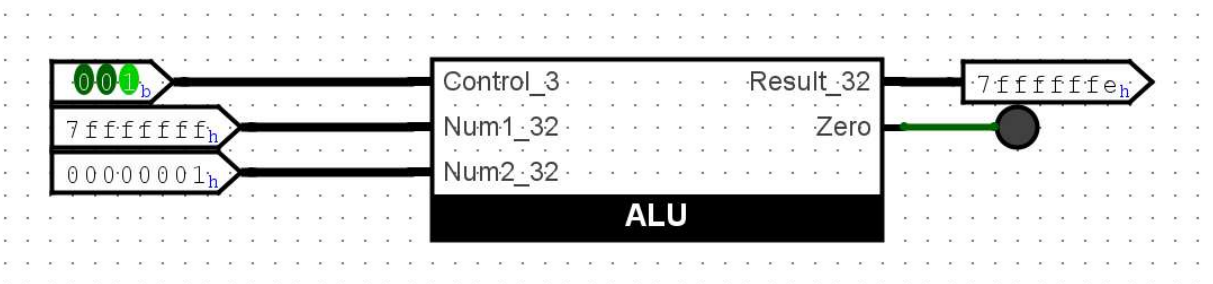
1、加法 000

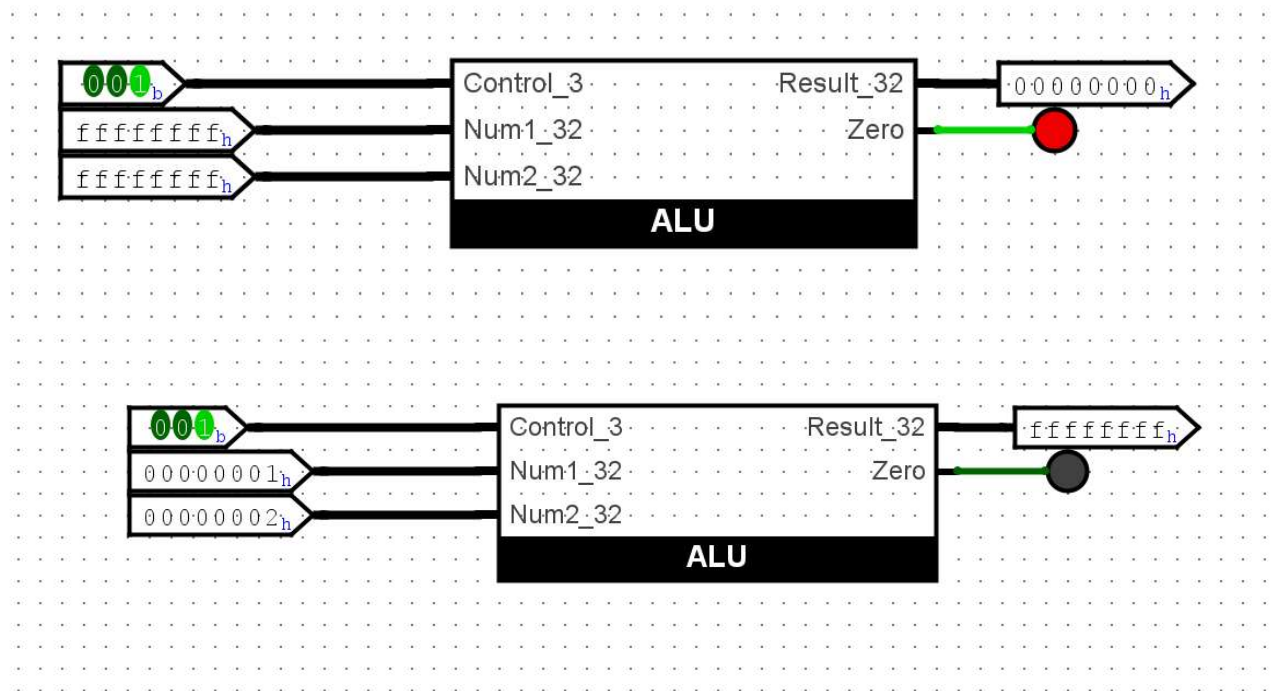
加法000，减法001，或010，SLT011，LUI100



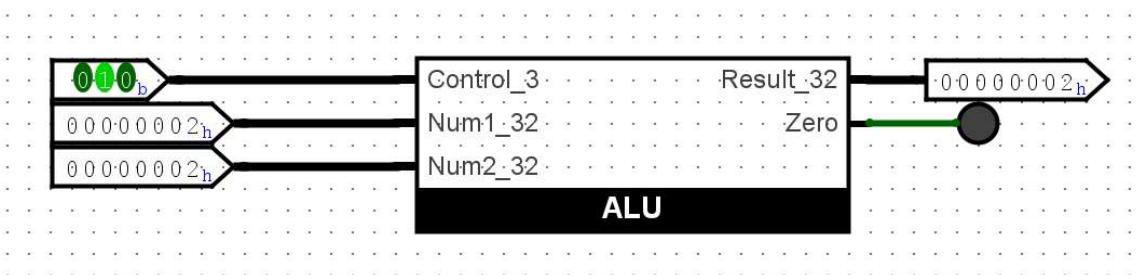


2、减法 001



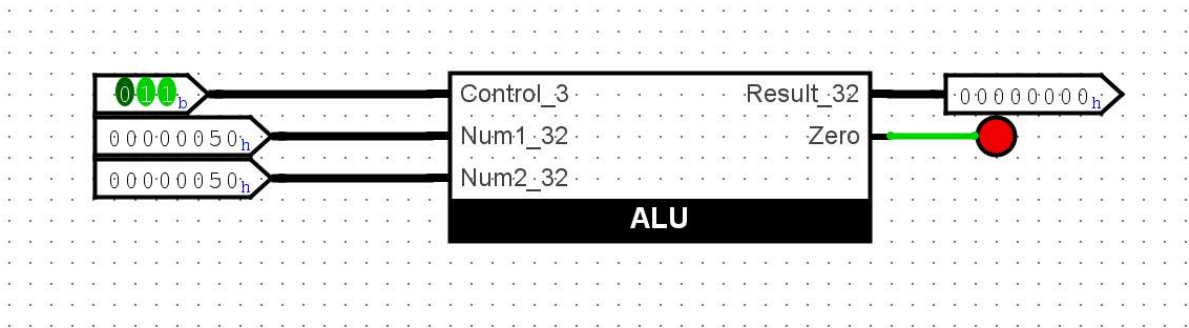
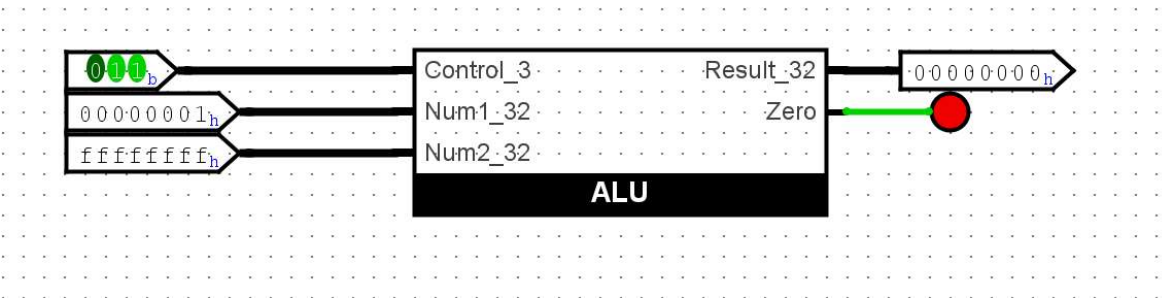
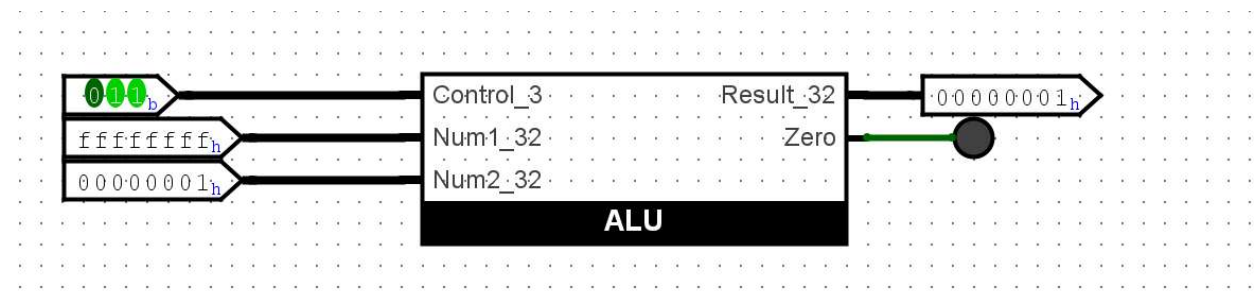


3、或



(请将测试的截图插入，注意测试用例的完整性)

4、SLT



5、LUI

