

# 实验 6：CPU 设计综合

姓名：孙文博      学号：201830210

## 一、实验目的

- 1、掌握不同指令数据通路的实现方式。
- 2、掌握 CPU 基本结构，并学习不同部件级联调试方法。
- 3、掌握 RISC-V 汇编程序设计，并转换成机器代码的方法。

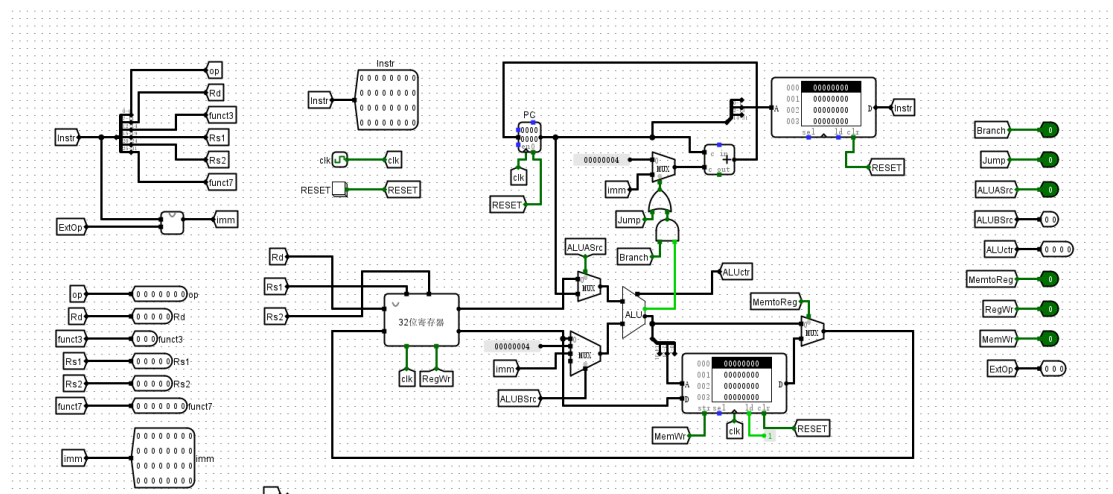
## 二、实验环境

Logisim-ITA V2.16.1.0

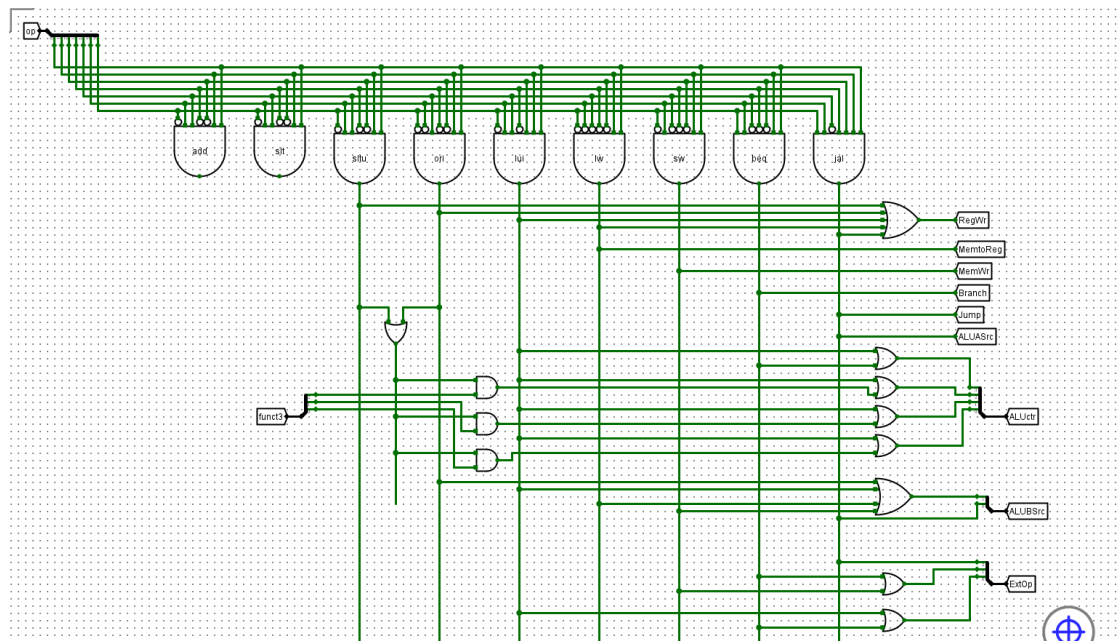
## 三、实验步骤

1. 根据给出的 9 条 RISC-V 指令的数据通路原理图，实现并验证数据通路的功能；根据给出的单周期 RISC-V CPU 结构原理图，实现验证 CPU 的功能。

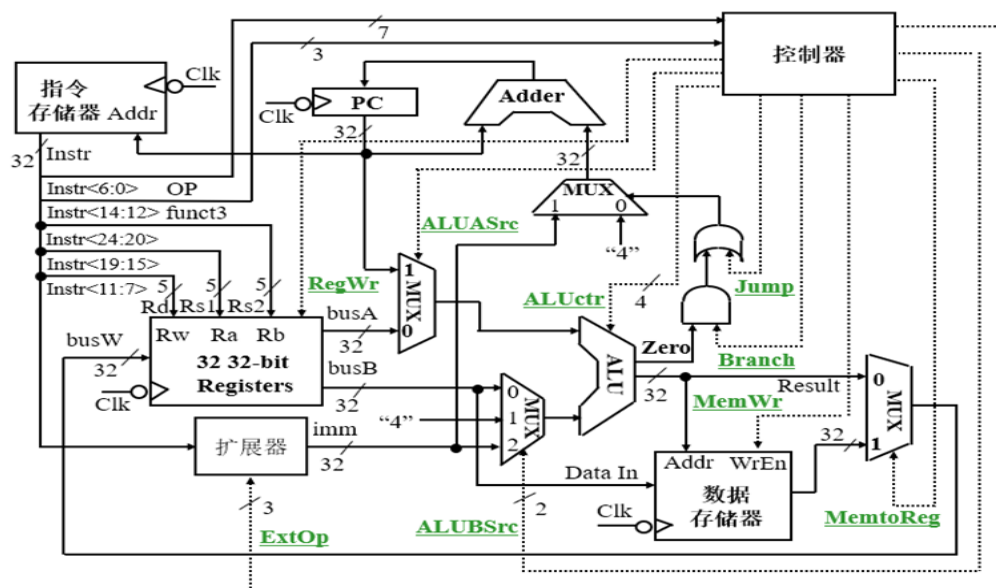
前两个实验可放在一起实现。其中第一步要求我们实现数据通路，也是需要我们将实验五中的取指令通路、实验四中的 ALU 部件和实验三中的 32 位寄存器合并在一起构成 CPU 数据通路，在此基础上加入实验五中设计的控制器即可形成基本的单周期 CPU，实验电路图如下：



其中控制器的电路图如下：



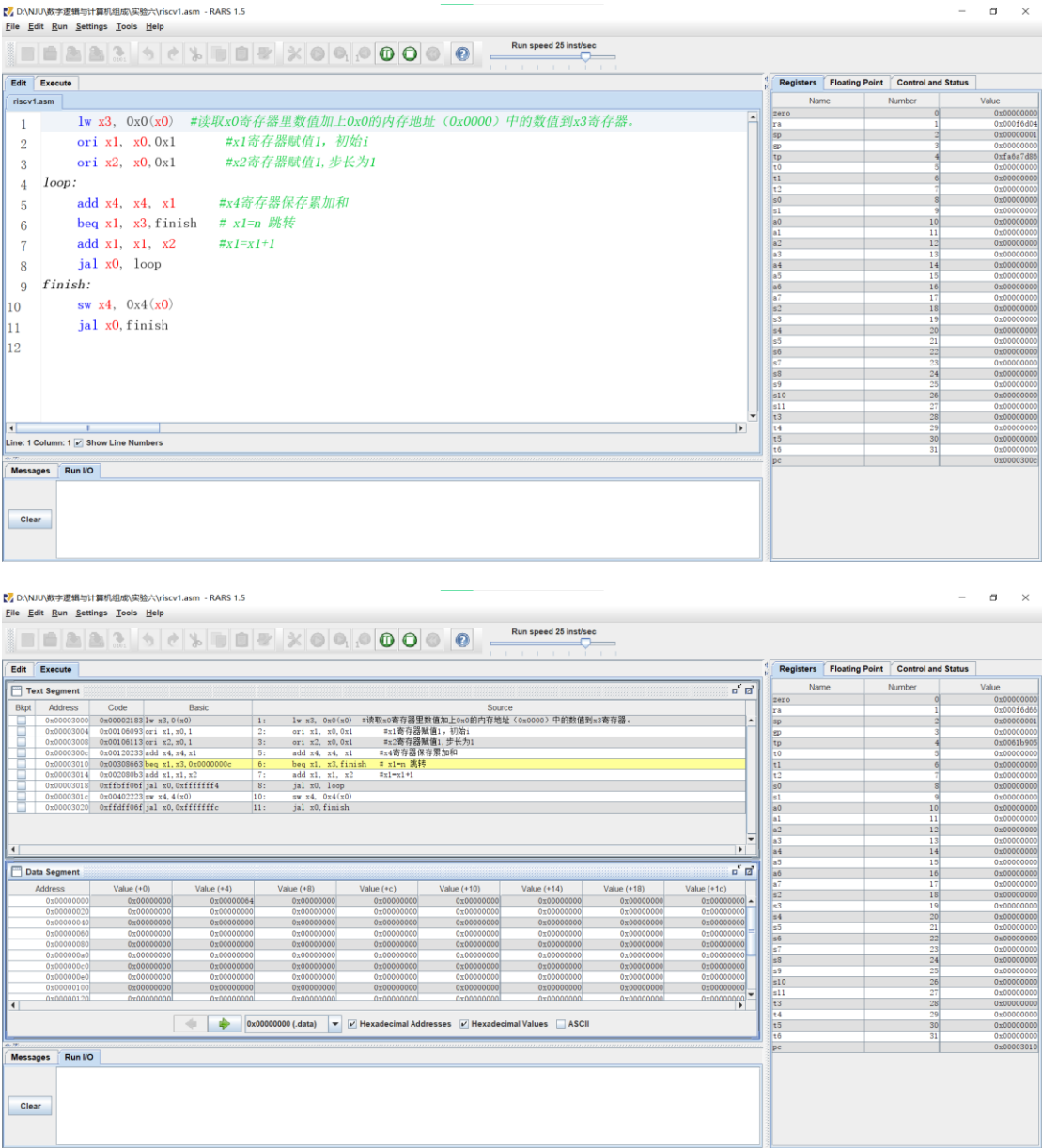
电路原理图如下：



- 编写一个仅使用已实现的 9 条 RISC-V 指令的汇编程序，输入参数和输出结果保存在 RAM 中。例如编写一个计算  $1+2+\dots+n$  的累加和程序，从内存 0x00000 中读入参数 n，通过循环累加的算法计算结果，并保存到内存地址 0x00001。

实验中需要用到 RISC-V 汇编模拟器 RARS，首先我们编写程序的汇编语言，通

过 RARS 模拟转换成机器代码，存入一个机器代码文件中，然后打开 logisim，将对应的指令导入指令寄存器中，最后给内存初始位置 0x0000 处赋初始参数值 n，选择合适频率运行，程序执行结束后内存 0x0001 的值即为得到的结果。实验操作步骤及仿真检测图如下：



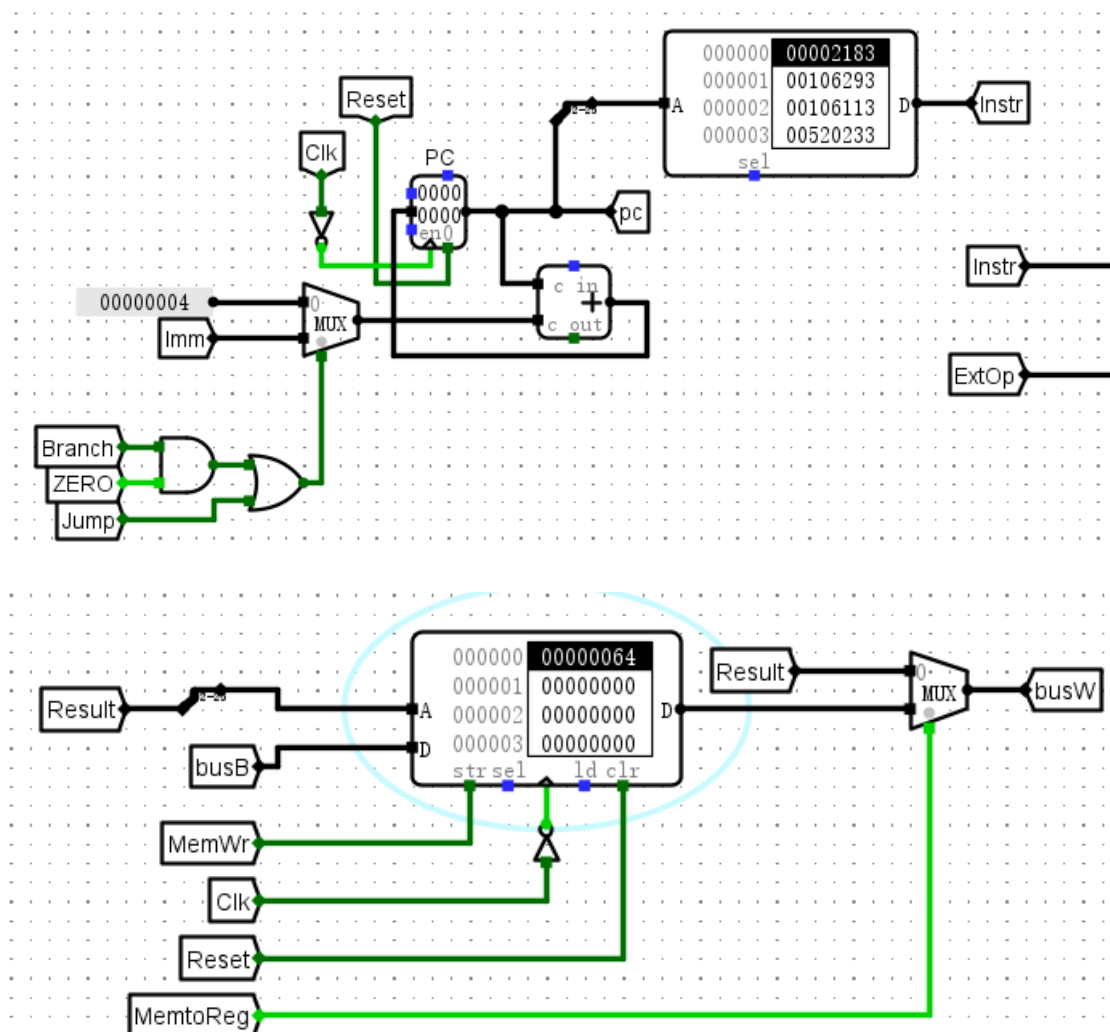
得到的机器代码保存到文件中：

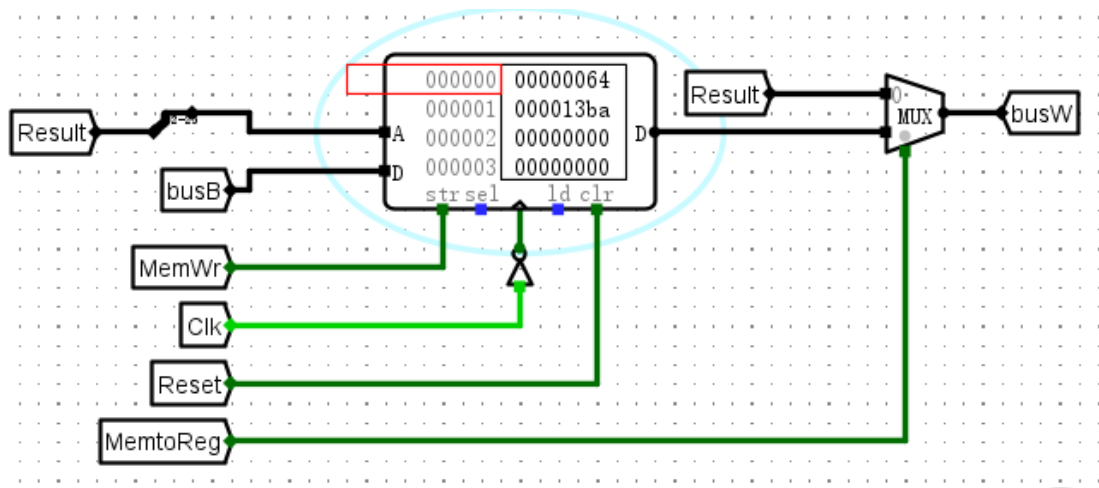
test2 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
v2.0 raw
00002183
00106293
00106113
00520233
00328663
002282b3
ff5ff06f
00402223
0000006f
```

打开 logisim, 修改指令寄存器和内存的值, 运行程序得到结果:





实现了计算  $1+2+\dots+100$  (10 进制) 的和为 5050 (16 进制为 13ba)，测试完成！

#### 四、思考题

1. 答：可在寄存器写口部分加入判定电路（通过与或门实现），若写口地址为 0 则不进行写入操作，其他正常。使用电路前默认设置 0 号寄存器的内容为 0。也可以将一个 0 常量接入 0 号寄存器中，实现始终为 0。
2. 答：需要修改 ALU 部件和 ALUstr 电路，并修改对应指令。首先在 ALUstr 中增加与运算和逻辑右移运算对应的操作码，然后修改 ALU 部件，添加这两种运算，其中与运算可直接由与门形成，逻辑右移运算可使用分线器完成，最后修改指令，添加这两种运算指令。
3. 答：编写汇编语言实现冒泡排序过程如下：

##### (1) 冒泡排序算法汇编代码设计

假设所有参数存放在从 0x0000 开始的一块存储区。首先存储的是待排序数据个数  $n$ ，然后存储  $n$  个待排序数据元素。将数据个数  $n$  读到寄存器  $x1$ ，

外循环变量 i 分配在 x2，内循环变量 j 分配在 x3，常量 1 和 4 分别存放在 x4 和 x5，常量 -1 存放在 x6，第 j 个元素 a[j]的地址存放 x7，第 j+1 个元素 a[j+1]的地址存放在 x8。第 j 个元素 a[j]读入 x9，第 j+1 个元素 a[j+1]读入 x10。

汇编语言源程序代码如下：

```
lw x1,0(x0)  #待排序的数字个数 n 存在 0x0000 处
```

```
add x2,x1,x0    #i=n
```

```
ori x4,x0,1     #x4=1
```

```
ori x5,x0,4     #x5=4
```

```
ori x6,x0,0xffffffff    #x6=-1
```

L1:

```
beq x2,x4,finish  #if i=1 则结束
```

```
ori x3,x0,1     #j=1
```

```
ori x7,x0,4
```

```
ori x8,x0,8
```

L2:

```
sltu x11, x3,x2  # if j<i then 读取两个元素比较
```

```
beq x11,x0,L3
```

```
lw x9,0(x7)      #读取第 j 个元素
```

```
lw x10,0(x8)     #读取第 j+1 个元素
```

```
sltu x11,x9,x10
```

```
beq x11,x4,L4
```

sw x10,0(x7) #交换存储

sw x9,0(x8) #交换存储

jal x0, L4

L3:

add x2,x2,x6

jal x0, L1

L4:

add x3,x3,x4 # j=j+1

add x7,x7,x5

add x8,x8,x5

jal x0, L2

finish:

jal x0, finish

## (2) 冒泡排序程序数据和机器代码的预置

将测试用的排序程序数据文件加载到数据存储器（RAM）中，数据文件内容

如下：v2.0 raw a 8 4 2 12 3 6 b 5 9 7:

```
000000 00002083 00008133 00106213 00406293 fff06313 04410663 00106193 00406393 00806413 0021b5b3 02058063 0003a483 00042503 00a4b5b3 00458c63 00a3a023
000010 00942023 00c0006f 00610133 fc9ff06f 004181b3 005383b3 00540433 fc9ff06f 0000006f 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000050 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000070 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

导出排序程序仿真执行后的机器代码并加载到指令存储器（ROM）中：

```
000000 0000000a 00000008 00000004 00000002 00000012 00000003 00000006 00000009 00000005 00000009 00000007 00000000 00000000 00000000 00000000 00000000
000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

## (3) 进行电路仿真，查看结果

排序结束后的 ROM 内容如下：

000000	0000000a	00000002	00000003	00000004	00000005	00000006	00000007	00000008	00000009	00000009	00000012	00000000	00000000	00000000	00000000
000010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000030	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000050	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000060	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000070	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000080	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000090	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000a0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000b0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000c0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000d0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000e0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0000f0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000110	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000120	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000130	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000140	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000150	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000160	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
000170	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

发现排序成功，测试完成！

## 五、 实验总结

本实验作为本次课程的最后一次实验， 结合了实验三的寄存器、实验四的算术运算部件和实验五的取指令通路与控制器， 构成了一个完整的微型 CPU 的数据通路， 综合性较强。回顾一学期的实验历程， 从学会使用 logisim 仿真软件到自己亲手搭建出小型 CPU， 成就感油然而生。然而这些仅仅只是开始， 期待下学期的数电实验中再接再厉！