实验六：CPU 设计综合

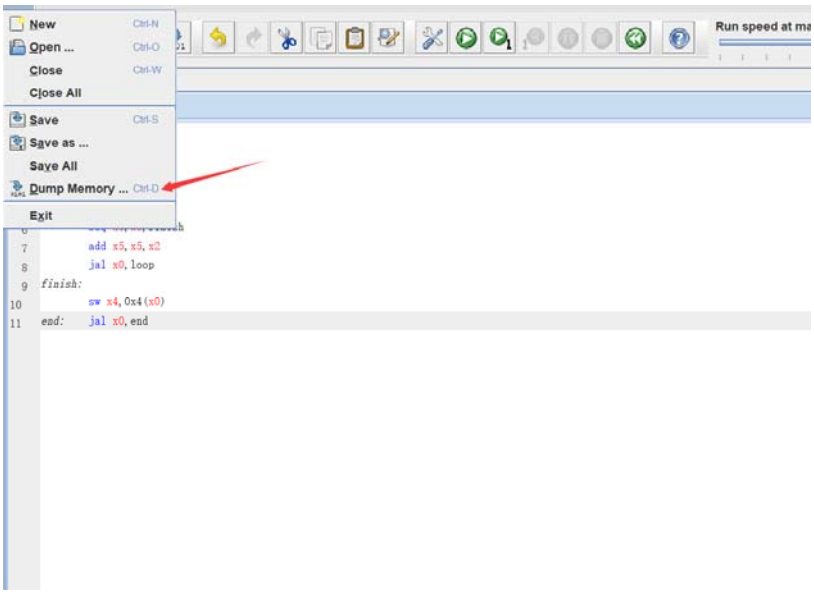
一、累加程序

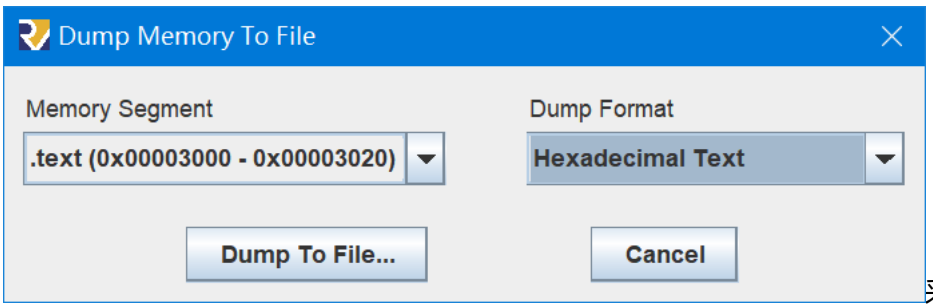
实验步骤：

（一）、编写 asm 代码

```
        lw  x3, 0x0(x0)
        ori  x5, x0, 0x1
        ori  x2, x0, 0x1
loop:
        add  x4, x4, x5
        beq  x5, x3, finish
        add  x5, x5, x2
        jal  x0, loop
finish:
        sw  x4, 0x4(x0)
end:    jal  x0, end
```

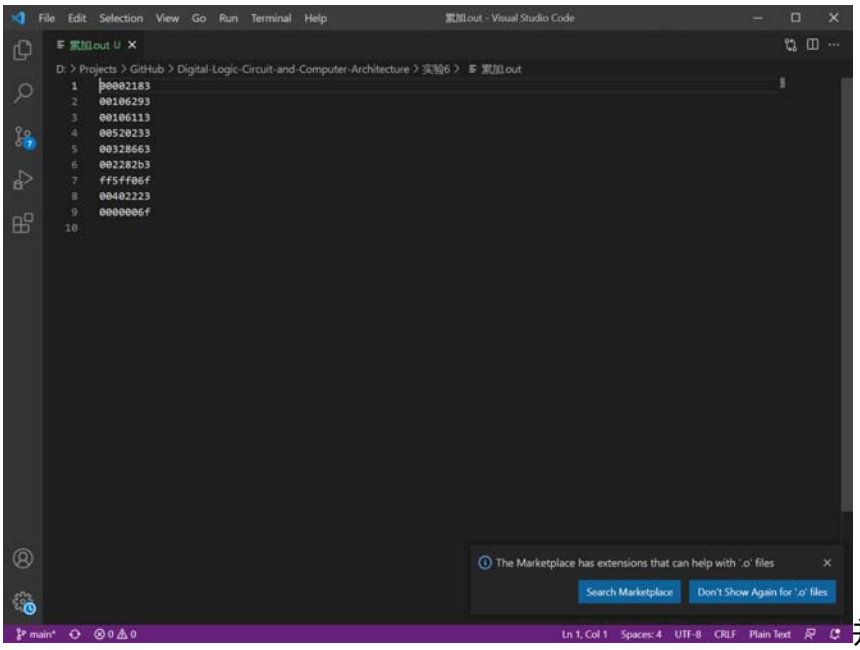
（二）、将 asm 导出为机器代码





采用 16 进

制保存

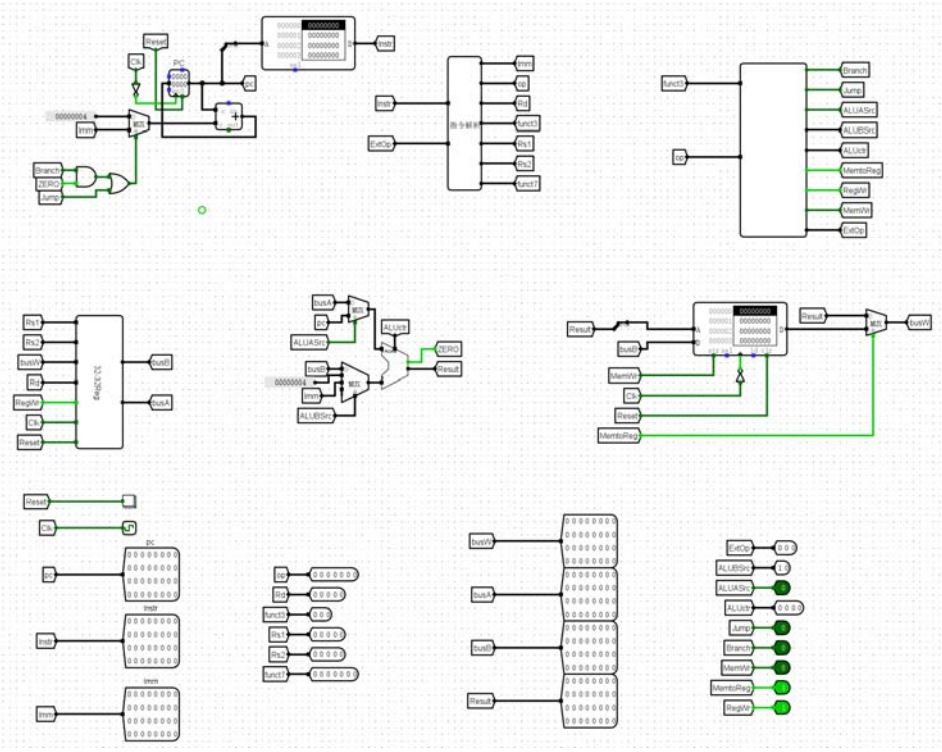


并在第一行加上

v2.0 raw

(三)、CPU 电路连接

根据上次实验所做的控制电路进行连接



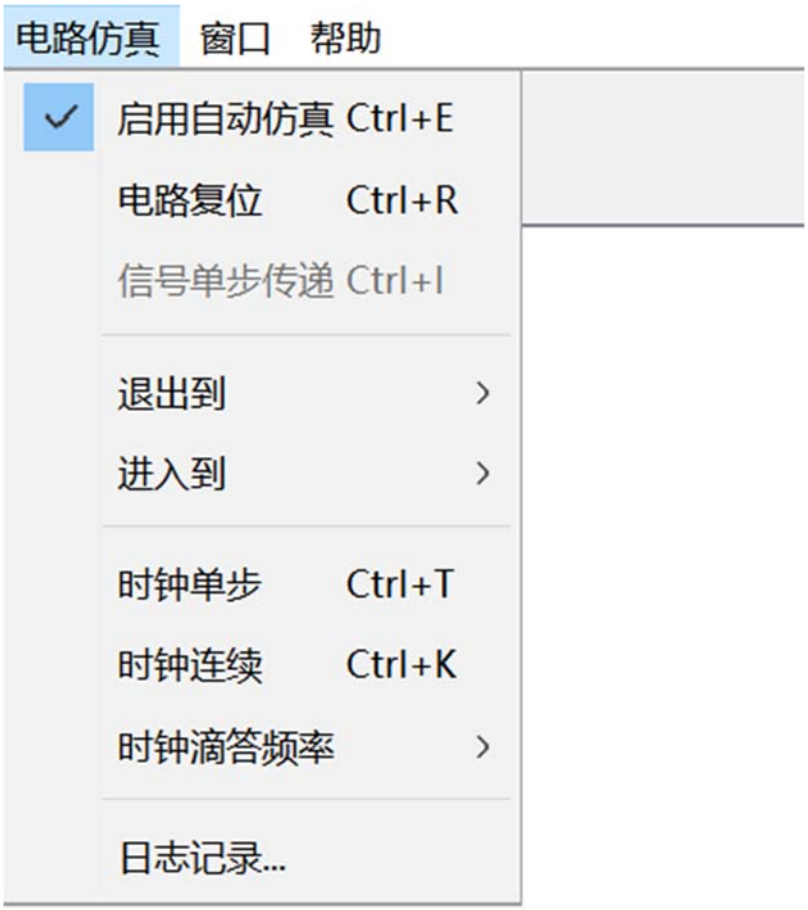
(四)、将累加代码加载到 ROM 指令存储器中

000000	00002183	00106293	00106113	00520233	00328663	002282b3	ff5ff06f	00402223	0000006f	00000000	00000000	00000000	00000000	0
000010	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0
000020	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0
000030	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0
000040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0

在数据存储器的第 0 个单位输入数据 0000000a

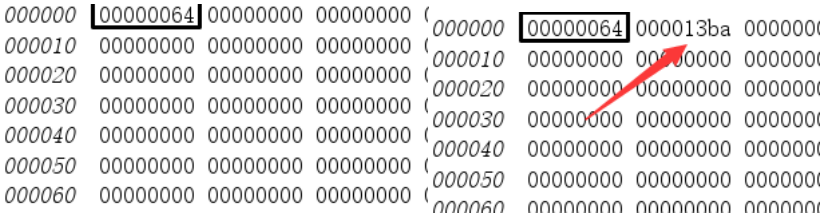
000000	0000000a	00000000	000000
000010	00000000	00000000	000000
000020	00000000	00000000	000000

(五)、电路仿真时钟连续



程序结束时打开数据存储器查看第 1 单元数据

(六)、Reset 后更改数据存储器数据位 00000064，再次电路仿真查看结果



(七)、遇到的问题

实验过程中发现之前程序设计的 ALU 和 32-32RegFile 有问题，对其进行重新设计

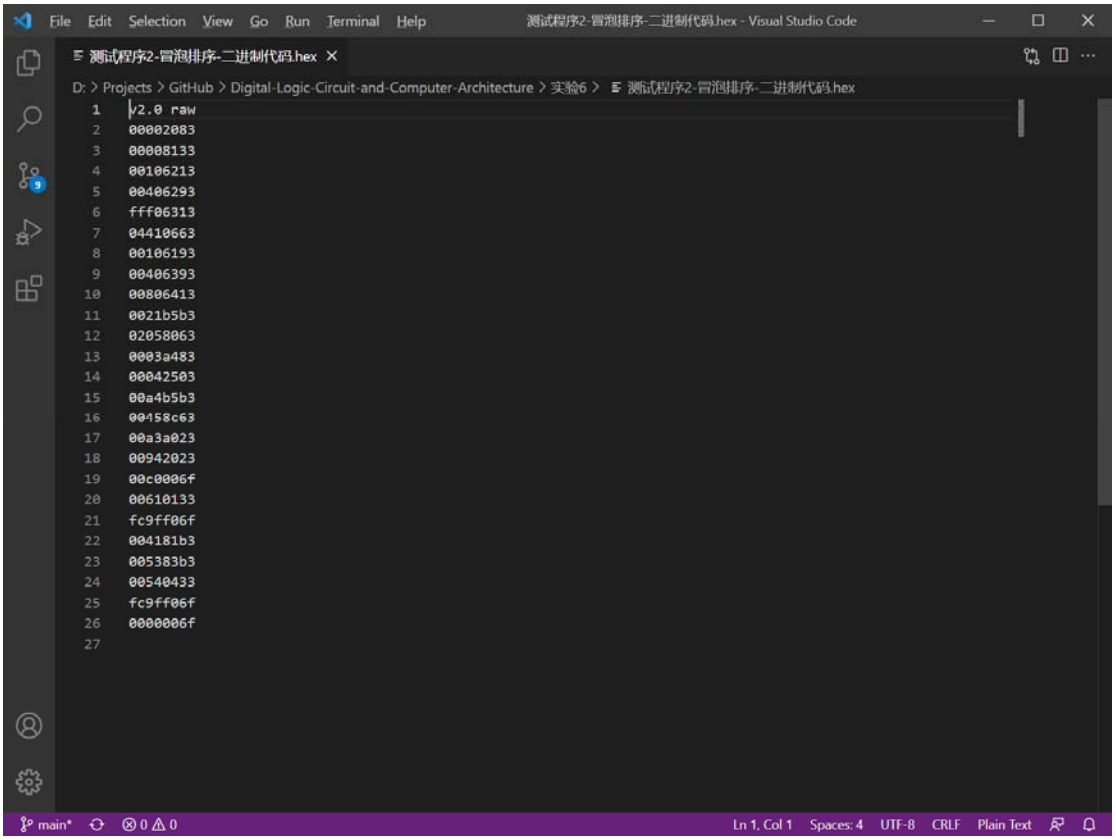
二、冒泡排序程序

实验步骤：

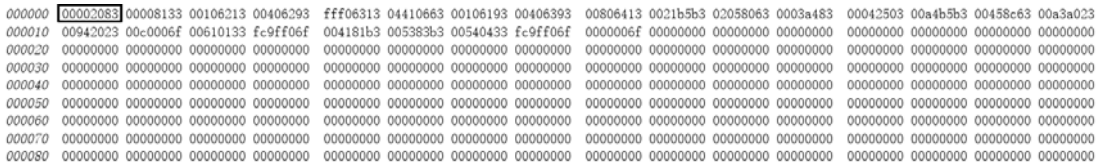
(一)、编写 asm 代码

```
        lw      x1, 0(x0)   #寰呗昧搴忕烷鏮板瓩涓啦n瀛樺瀌0x00澧剪紝
        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 鍒憒粨鏰
        ori  x3, x0, 1  #j=1
        ori  x7, x0, 4
        ori  x8, x0, 8
L2:
        sltu  x11,  x3, x2 # if j<i then 璇诲彛涓や釜鍊冪硫姣旇緝
        beq  x11, x0, L3
        lw x9, 0(x7)      #璇诲彛绍瑛涓厓绱
        lw x10, 0(x8)     #璇诲彛绍瑛+1涓厓绱
        sltu x11, x9, x10
        beq x11, x4, L4
        sw x10, 0(x7)   #浜ゅ崝瀛樺倱
        sw x9, 0(x8)    #浜ゅ崝瀛樺倱
        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
```

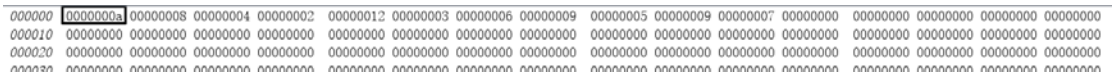
（二）、将汇编代码导出为机器代码



（三）、将机器代码加载到指令存储器中、将数据加载到数据存储器中



ROM 数据



RAM 数据

（四）、进行电路仿真

程序结束后查看 RAM 中的数据

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

思考题:

1. 更改 ALU 和控制信号生成的部件
2. 负数的累加不能计算，程序只能计算无符号整数的和