汇编程序设计

> 计算机组成原理实验三
>
> PB20020480

实验目的

- **了解汇编程序的基本结构，以及汇编程序仿真和调试的基本方法**
- **熟悉RISC-V常用32位整数指令的功能，掌握简单汇编程序的设计，以及CPU下载测试方法和测试数据(COE文件) 的生成方法**

实验环境

- RARS (RISC-V Assembler and Runtime Simulator)
- Ripes (图形化RISC-V模拟器)

实验内容与结果

一、汇编指令测试

设计汇编程序，实现对10条指令功能的逐条简单测试和人工检查，并生成COE文件

1.待测试指令

−add, addi, sub, auipc

−lw, sw

−beq, blt, jal, jalr

2.代码设计

```
.data
led_data:.word 1    #address of led_data=0x0
swt_data:.word 2
fail_data:.word 0xffff

.text
test_lw:
    lw a1, 0(x0)    #test_lw
    lw a2, 4(x0)
test_sw:
    sw a2, 0(x0)    #test_sw
test_jal:
    jal x0,test_beq  #test_jal
silent1:
    addi a1, x0, 0
test_beq:
    beq x0,x0 ,test_blt  #test_beq
    jal x0, fail
silent2:
    addi a2, x0, 0
test_blt:
    blt x0,a1,test_addi   #test_blt
```

```
test_addi:
    addi a2, x0,2       #test_addi
    addi a0,x0,2
    beq a0,a2,test_add
    jal x0,fail
test_add:
    add a3, a1, a2    #test_add
    addi a0,x0,3
    beq a0,a3,test_sub
    jal x0,fail
test_sub:

    sub a3, a3,a1    #test_sub
    addi a0,x0,2
    beq a0, a3, test_auipc
    jal x0,fail

test_auipc:
    auipc t1,0x1    #test_auipc
    auipc t0,0x1
    addi t1,t1,4
    beq t0,t1,test_jalr
    jal x0,fail
test_jalr:
    auipc t0,0      #test_jalr
    jalr x0,t0,20
    jal x0,fail
fail:
    lw a0,8(x0)
    sw a0,0(x0) #if fail a0=0xffff,and led=0xffff
exit:
    addi a0,x0,0 #if success a0=0 and led=2 ,which means sw successfully
```

3、运行前

寄存器

Registers | Floating Point | Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x00002ffc
gp	3	0x00001800
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00003000

内存

Data Segment

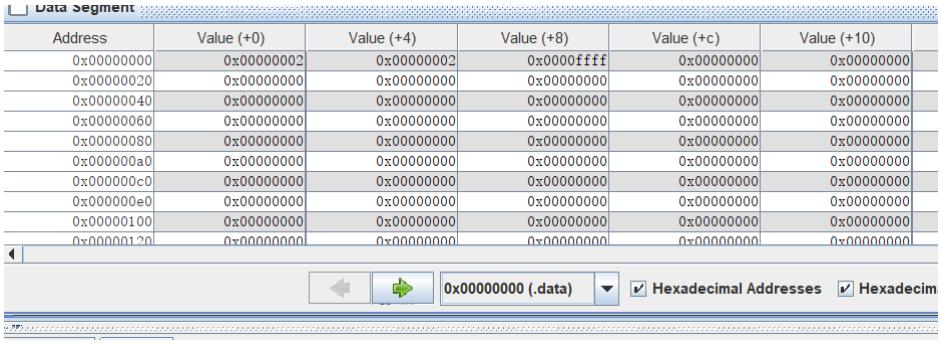
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x00000000	0x00000001	0x00000002	0x0000ffff	0x00000000
0x00000020	0x00000000	0x00000000	0x00000000	0x00000000
0x00000040	0x00000000	0x00000000	0x00000000	0x00000000
0x00000060	0x00000000	0x00000000	0x00000000	0x00000000
0x00000080	0x00000000	0x00000000	0x00000000	0x00000000
0x000000a0	0x00000000	0x00000000	0x00000000	0x00000000
0x000000c0	0x00000000	0x00000000	0x00000000	0x00000000
0x000000e0	0x00000000	0x00000000	0x00000000	0x00000000
0x00000100	0x00000000	0x00000000	0x00000000	0x00000000
0x00000120	0x00000000	0x00000000	0x00000000	0x00000000

4、运行后

寄存器

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x00002ffc
gp	3	0x00001800
tp	4	0x00000000
t0	5	0x0000306c
t1	6	0x0000405c
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000003
a1	11	0x00000001
a2	12	0x00000002
a3	13	0x00000003
a4	14	0x00000003
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x0000308c

内存



5、解释

自动化测试指令

(1)测试lw：如果正确，a1,a2的值应该是1，2

(2)测试sw：如果正确，led最终的值应该是2

(3)测试jal：如果正确，跳过silent1标签，a1的值是1；运行错误，a1=0

(4)测试beq：如果正确，跳过silent2标签，a2的值是2；运行错误，a2=0

(5)(6)(7)(8)(9)分别测试addi,add,sub,auipc,jalr

如果都正常运行，则a2=2,a3=3,a4=5,t0=0x306c,t1=0x405c,led=2

如果与预期值不同，会跳向fail标签，最终led的值是0xffff

二、汇编程序设计排序算法

1.设计思路

- 选择排序算法：

  两重循环，外循环决定最小值位置，内循环查找最小值

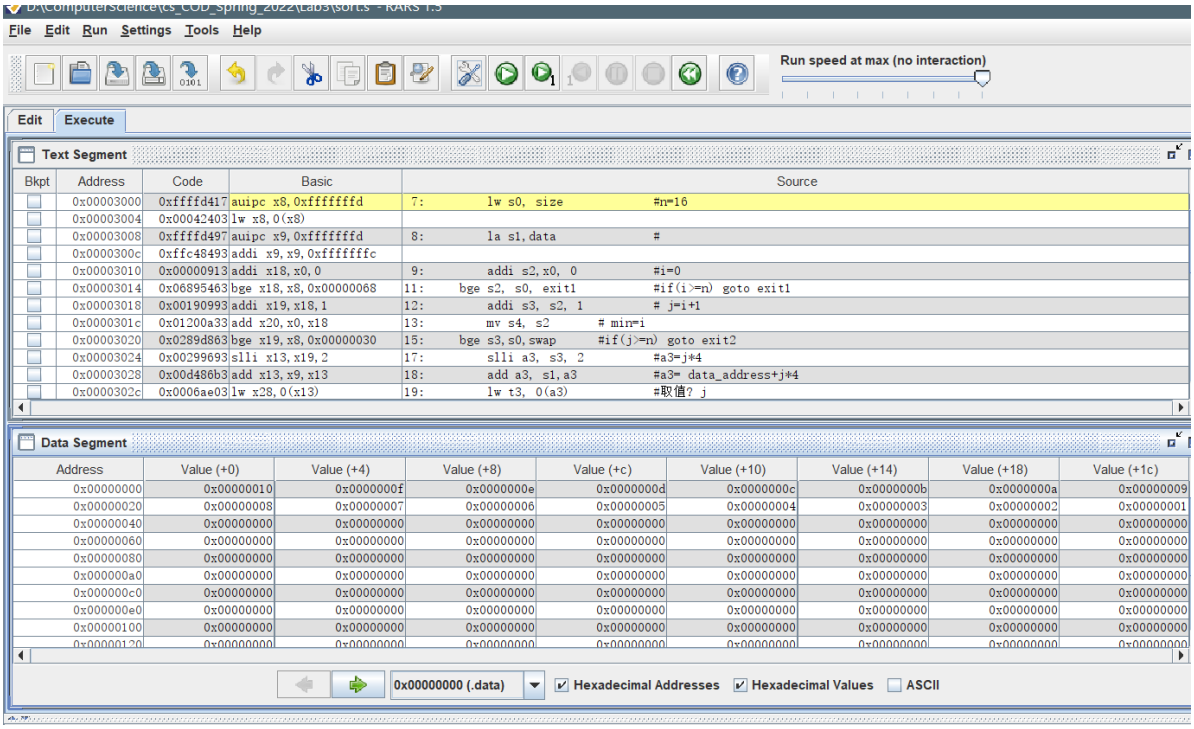
- 将结果输出到显示器上

2.代码

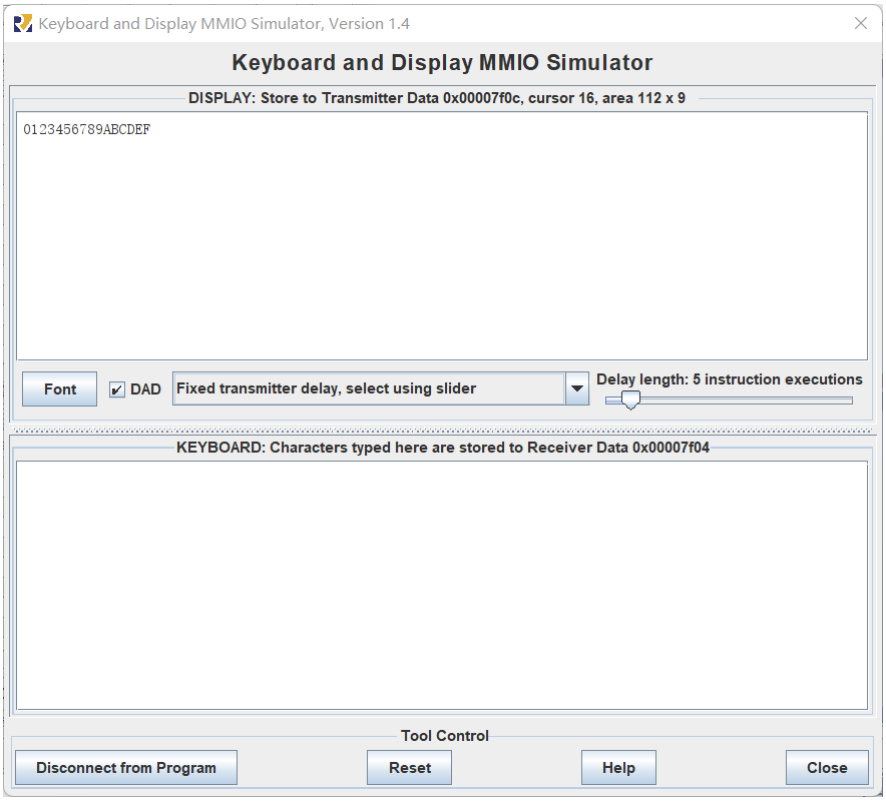
```
.data
size:.word 16
data:.word 0xf, 0xe, 0xd, 0xc, 0xb, 0xa, 0x9, 0x8, 0x7, 0x6, 0x5, 0x4, 0x3, 0x2,
0x1, 0x0

.text
sort:
    lw s0, size     #n=16
    la s1,data
    li s2, 0        #i=0
loop1:
        bge s2, s0, exit1       #if(i>=n) goto exit1
    addi s3, s2, 1      # j=i+1
    mv s4, s2       # min=i
loop2:
    bge s3,s0,swap      #if(j>=n) goto exit2

    slli a3, s3, 2      #a3=j*4
    add a3, s1,a3       #a3= data_address+j*4
```

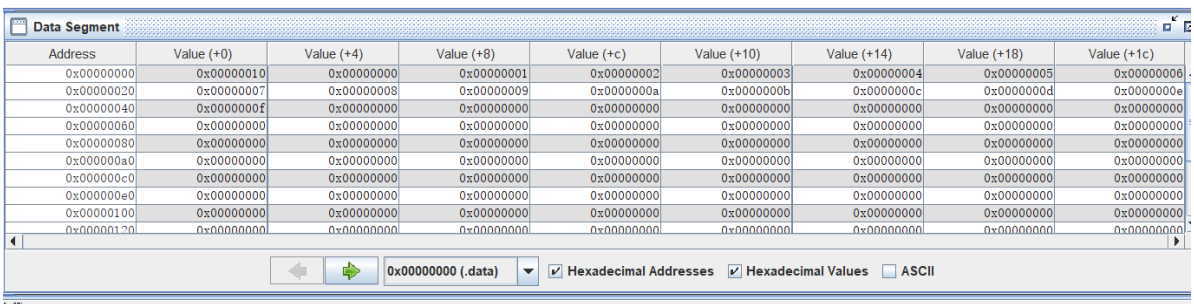
```
    lw t3, 0(a3)        # j
    slli a4, s4, 2      #a4 = min*4
    add a4, s1,a4       #a4 = data_address+min*4
    lw t4, 0(a4)        #min
    ble t3, t4 , select_min    #if(data[j]<data[min]) select_min
    jal x0 ,next
select_min:
    mv s4, s3

next:
    addi s3, s3, 1      #j++
    jal x0, loop2       #goto loop2

swap:
    beq s2, s4, exit2       #if(i==min) next loop1
    slli a2, s2, 2      #a2=i*4
    add a2, s1, a2      #a2 = data_address+i*4
    lw t2, 0(a2)        # i
    slli a4, s4, 2      #t0=min*4
    add a4, s1, a4      #t0= data_address+min*4
    lw t4, 0(a4)        # min
    sw t2, 0(a4)        #
    sw t4, 0(a2)

exit2:  addi s2, s2, 1      #i++
    jal x0, loop1       #goto loop2
exit1:
    li a0, 0x7f08
    li s2, 0        #i=0
    li s3, 9
loop3:
        bge s2, s0, exit3

        slli a2, s2, 2      #a2=i*4
    add a2, s1, a2      #a2 = data_address+i*4
    lw t2,0(a2)
    ble t2,s3,case1
    jal x0, case2
case1:
    addi t2,t2,48

    sw t2,4(a0)
    jal x0,forever
case2:
    addi t2, t2,55

    sw t2,4(a0)

forever:
    lw t0,0(a0)
    beq t0,x0,forever

        addi s2, s2, 1      #i++
    jal x0, loop3       #goto loop3
exit3:
```

3.运行前



4.运行后





结果符合预期，生成COE文件

实验总结

- 本次试验中，掌握了汇编语言的基本指令，熟悉了Risc-V汇编模拟器.
- 本次试验中，熟悉了汇编语言的编写与调试，了解了Keyboard and Display MMIO Simulation