COD LAB3 实验报告

姓名：刘芷辰

学号：PB21111728

日期：2023.4.18

1. 实验题目

汇编程序设计

2. 实验目标

- 理解RISC-V常用32位整数指令功能
- 熟悉RISC-V汇编仿真软件RARS，掌握程序调试的基本方法
- 掌握RISC-V简单汇编程序设计以及存储器初始化文件（COE)的生成方法
- 理解CPU调试模块PDU的使用方法

3. 实验内容

3.1 PART 1（设计汇编程序）

> 必做部分（只是用10种指令及其伪指令）

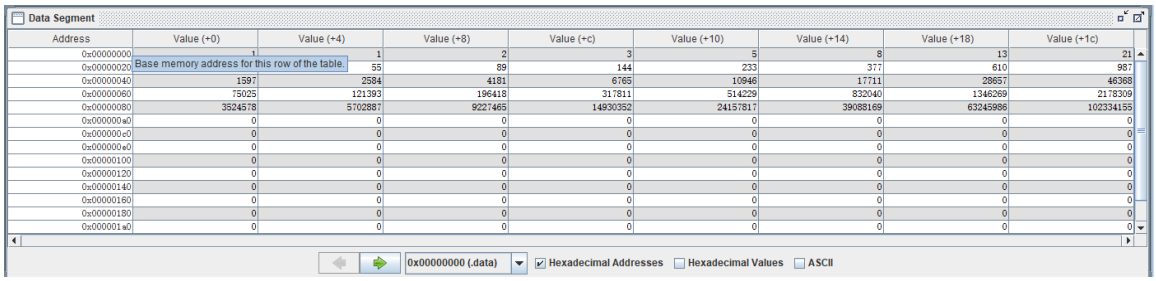
- 代码

```
.data
save: .word 0x0001,0x0001

.text
# 将n加载到寄存器t0中
addi t0,zero,40

la a0, save
# 将第一项和第二项加载到寄存器t1和t2中
lw t1, 0(a0)
lw t2, 4(a0)
```

```
# 将第一项保存到内存中
sw t1, 0(a0)
# 将第二项保存到内存中
sw t2, 4(a0)
# 初始化计数器i为2
li t3, 2
loop:
  # 如果i = n，跳出循环
  beq t3, t0, end
  # 计算下一项的值
  add t4, t1, t2
  # 保存下一项的值到内存中
  add t5, t3, t3
  add t5, t5, t3
  add t5, t5, t3              #4*i
  add t5, t5, a0
  sw t4, 0(t5)
  # 更新t1和t2
  add t1, zero, t2
  add t2, zero, t4
  # 更新计数器i
  addi t3, t3, 1
  # 继续循环
  jal zero loop
end:

```

n的加载通过addi指令改变，设置计数器t3，每更新一次加1，在小于n时，将储存在t1的数和存储在t2的数相加得到下一项，然后更新t1和t2中的内容

- 实验结果



> 选做部分（实现n的输入）

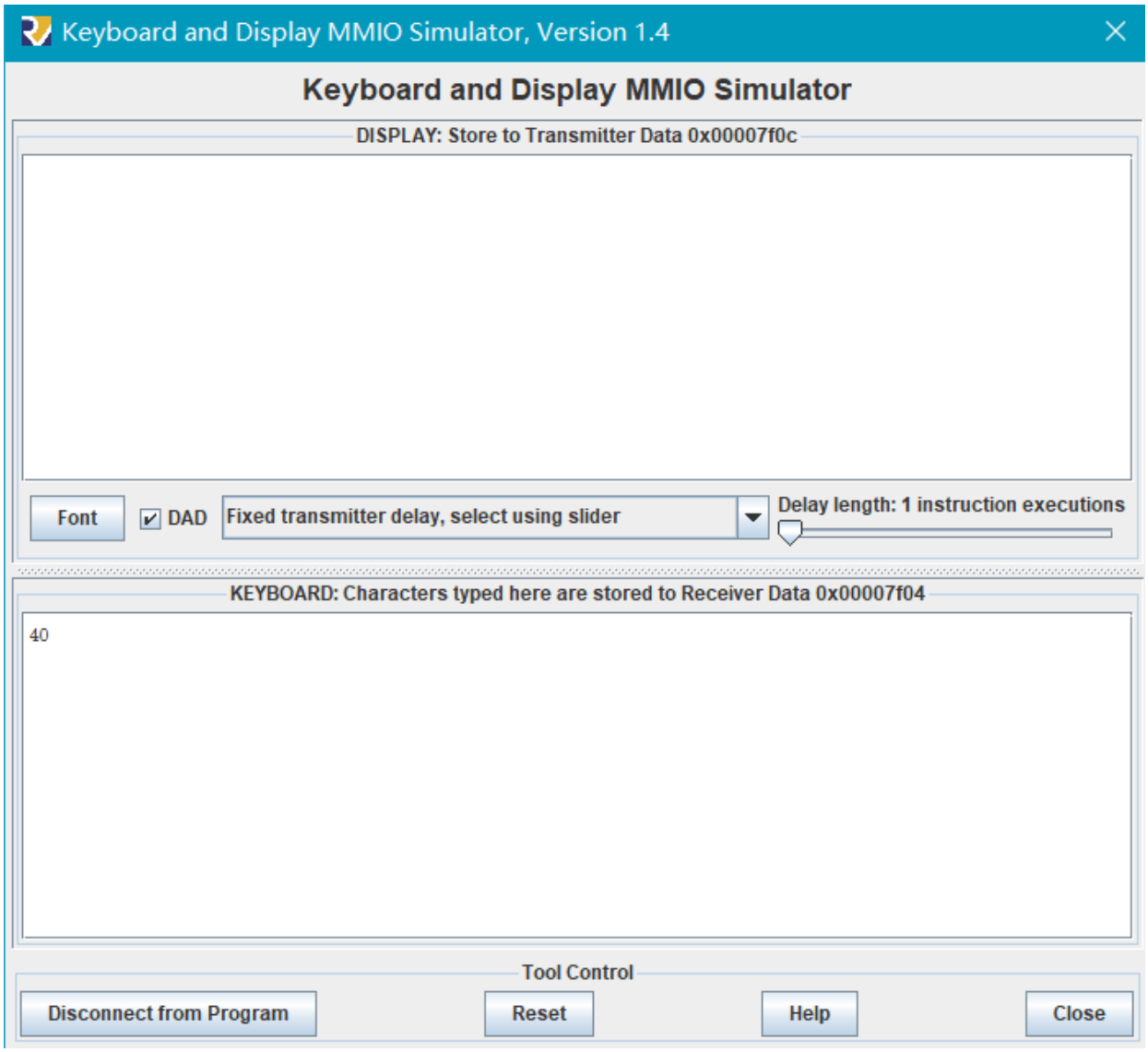
- 代码

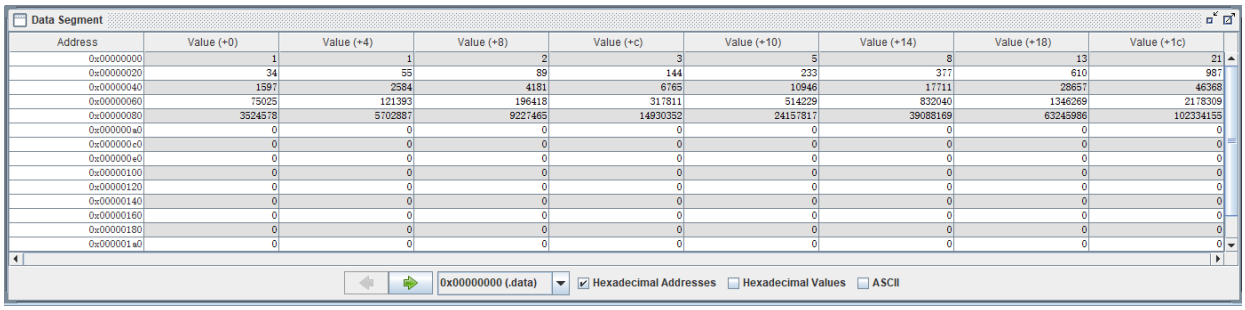
```
.data
save: .word 0x0001,0x0001
kbsr: .word 0x7f00
kbdr: .word 0x7f04
.text
addi t2,zero,1
addi t3,zero,32                    #空格作为终止符
addi x11,zero,2                    #计数，判断是否两位数
cal:
beq x11,zero,doubdigit             #一位数
addi t5,t4,-48
jal zero start
doubdigit:                         #两位数
addi t6,t4,-48
slli x12,t5,3
add x12,x12,t5
add t5,x12,t5
add t5,t5,t6
start:
lw t1, kbsr
lw t1, 0(t1)
bne t1,t2,start                    #判断kbsr
lw t4, kbdr
lw t4, 0(t4)
addi x11,x11,-1
bne t4, t3,cal                     #判断是否为终止符

# 将n加载到寄存器t0中

add t0, t5,zero

la a0, save
# 将第一项和第二项加载到寄存器t1和t2中
lw t1, 0(a0)
lw t2, 4(a0)
# 将第一项保存到内存中
sw t1, 0(a0)
# 将第二项保存到内存中
sw t2, 4(a0)
# 初始化计数器i为2
li t3, 2
loop:
  # 如果i >= n，跳出循环
  bge t3, t0, end
  # 计算下一项的值
```

```
    add t4, t1, t2
    # 保存下一项的值到内存中
    slli t5, t3, 2                  #4*i
    add t5, t5, a0
    sw t4, 0(t5)
    # 更新t1和t2
    add t1, zero, t2
    add t2, zero, t4
    # 更新计数器i
    addi t3, t3, 1
    # 继续循环
    jal zero loop
end:

```

实现轮询输入N：每次读取KBSR地址的数据，若不是1，则继续轮询等待输入，直到输入数据后，KBSR被置1，将KBDR中的数据判断是否为终止符，不是则第一位存入t5，第二位存入t6，并判断此时是否已经读入了两位，一位则继续轮询直到终止符，两位则将第一位存入t5的ascll码换算为十进制乘上10再加上第二位，实现了两位数n的输入，然后将得到的n存入t0中，后续和必做一样

- 实验结果





3.2 PART 2（设计32位移位寄存器）

> 32位移位寄存器代码

```
module Shift_reg(
    input rst,
    input clk,          // Work at 100MHz clock

```

```
    input [31:0] din,    // Data input
    input [3:0] hex,     // Hexadecimal code for the switches
    input add,           // Add signal
    input del,           // Delete signal
    input set,           // Set signal

    output reg [31:0] dout  // Data output
);

    reg [31:0] shift_reg;


    always @(posedge clk or posedge rst) begin
        if (rst) begin
            shift_reg <= 32'b0;
            dout <= 32'b0;
        end
        // Set
        else if (set) begin
                shift_reg <= din;
            end
        // Add
        else if (add) begin
                shift_reg <= {shift_reg[27:0], hex};
            end
        // Delete
        else if (del) begin
                shift_reg <= {4'b0,shift_reg[31:4]};
            end

            dout <= shift_reg[31:0];
        end


endmodule

```

通过拼接的方式实现移位

> FPGA烧写检查

已在线下检查

4. 总结

本次实验难度适中

更加熟悉了RISC-V的指令操作，并且学习了如何通过轮询的方式从键盘输入数据

本次有实验文档之后对实验理解起来容易一些了，希望之后的实验都能有实验文档