



廈門大學

《计算机组成原理》

课程实验报告

姓名： 庾晓萍

学院： 信息学院

系： 软件工程

专业： 软件工程

学号： 20420192201952

第六次实验 带中断的控制器实验

1. 实验环境

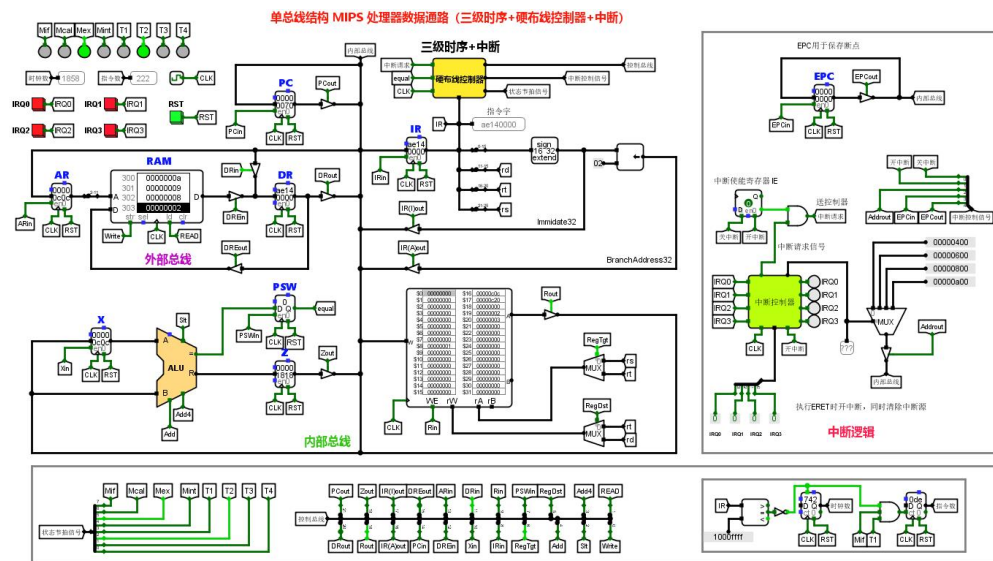
- (1) MIPS32 汇编器
- (2) Logisim 软件

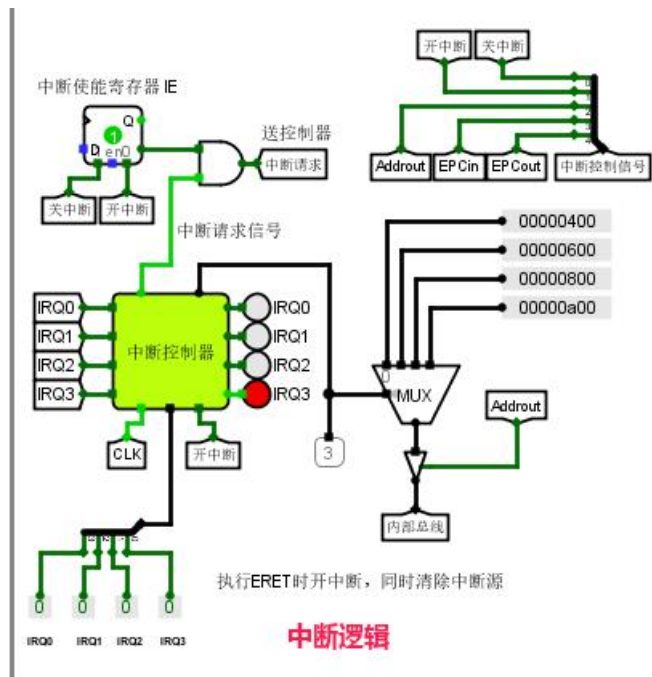
2. 实验内容

2.1 验证试验

2.1.1 支持中断的单总线结构 MIPS 处理器（硬布线控制器）

- 编写含有中断服务程序的冒泡法降序排序程序（只能使用 5+1 条指令：lw、sw、beq、slt、addi、eret）
- 在 MIPS 汇编器上进行汇编，得到机器码，并添加：v2.0 raw, 220*0, 118*0, 118*0, 118*0;
- 在支持中断控制的单总线结构 MIPS 处理器数据通路上运行含有中断服务程序的冒泡法降序排序程序





```

300 0000000a 00000009 00000008 00000007 00000006 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00000000 00000000 00000000
310 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
320 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
330 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
340 00000002 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
350 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
360 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
370 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

修改为升序：

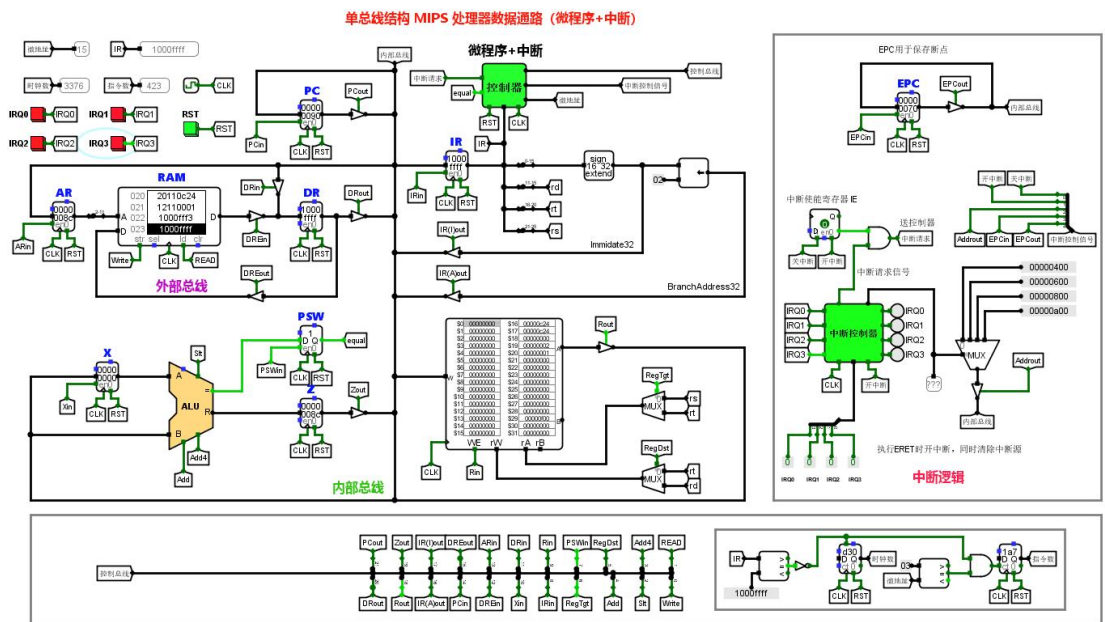
```

300 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000009 0000000a 00000000 00000000 00000000 00000000 00000000
310 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
320 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

2.1.2 支持中断的单总线结构 MIPS 处理器（微程序控制器）

- 在支持中断控制的单总线结构 MIPS 处理器数据通路上运行含有中断服务程序的冒泡法降序排序程序
- 运行“升序排序程序”，在程序的运行过程中，按 IRQ0、IRQ1、IRQ2、IRQ3 键，发出中断请求信号，并观察实验结果。
- 分析电路



```

300 0000000a 00000009 00000008 00000007 00000006 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00000000 00000000 00000000
310 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
320 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
330 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
340 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
350 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
360 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
370 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  
```

修改为升序:

```

300 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000009 0000000a 00000000 00000000 00000000 00000000 00000000
  
```

电路分析

答: 电路增加中断按键信号采样电路。

(1) **IR** 是中断请求寄存器, 输出与中断屏蔽位进行逻辑与后送中断优先编码器; 同步清零信号用于清除中断请求信号, 中断请求信号必须等待中断服务程序执行到中断返回时才能清除; 中断等待指示 **LED** 用于指示当前中断请求, 中断服务程序返回时应熄灭。

(2) 与中断相关的寄存器, 如中断使能寄存器 **IE**、异常程序计数器 **EPC**。**IE** 用于开关中断, 1 表示开中断, 0 表示关中断, 开关中断建议采用同步置位和复位方式。**EPC** 用于存放中断程序返回地址, 在中断响应阶段硬件会自动将主程序 **PC** 值送 **EPC** 保存。

(3) 设计中断识别逻辑能实现实验要求的中断响应优先级, 能正确识别 1~3 号中断源, 并设计向量中断机制, 可由中断号寻找中断程序入口地址。中断识别部分设计可以采用优先编码器实现, 由于不需要动态调整中断处理优先级, 所以中断屏蔽寄存器部分电路省略。

(4) 中断响应周期需要实现硬件关中断、将主程序断点保存至 **EPC** 寄存器、将中断识别逻辑产生的中断服务程序入口地址送 **PC**, 请逐一实现以上各数据通路。

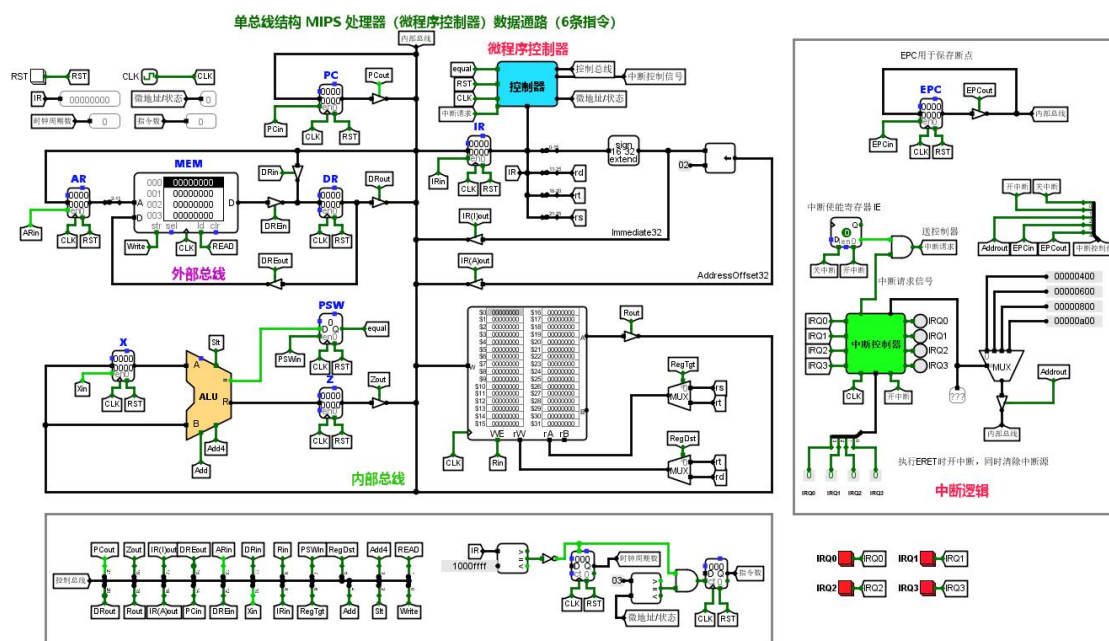
(5) **eret** 指令数据通路。单级中断服务程序主体是保护现场、中断服

务、恢复现场、开中断、中断返回。保护现场、恢复现场可以采用堆栈方式实现,配合 Ssp 寄存器,利用已实现的 lw、sw 指令即可实现;这里 eret 指令的主要功能 是将 EPC 寄存器送 PC,开中断,发送中断结束信号,熄灭当前中断请求的指示灯。

2.2 设计实验

支持中断的单总线结构 MIPS 处理器（微程序控制器，增加了 add 指令）

- 请在实验五完成的“单总线结构 MIPS 处理器（微程序控制器，6 条指令，增加了 add 指令）数据通路”上增加支持中断控制的功能，使其能支持 4 个中断源



请将此电路改造成支持中断的数据通路

- 编写含有 4 个中断服务程序的累加和程序（使用 6+1 条指令：lw、sw、beq、slt、addi、add、eret）
- 在 MIPS 汇编器上进行汇编，得到机器码，并添加相关的内容，使 4 个中断服务程序的起始地址分别为：

#IRQ0 中断服务程序的入口地址: 1024 = 400H
 #IRQ1 中断服务程序的入口地址: 1536 = 600H
 #IRQ2 中断服务程序的入口地址: 2048 = 800H
 #IRQ3 中断服务程序的入口地址: 2560 = A00H

RAM 对应 100
 RAM 对应 180
 RAM 对应 200
 RAM 对应 280

- 在支持中断控制的单总线结构 MIPS 处理器数据通路（微程序控制器+6 条指令）上运行含有中断服务程序的累加和程序，并测试 4 个中断请求

- 在支持中断控制的单总线结构 MIPS 处理器数据通路（微程序控制器+6 条指令）上运行含有中断服务程序的程序的冒泡法排序程序、计算费波那契数列程序，并测试 4 个中断请求

6 / 10


```

22  .text
23
24  main:
25      addi $s0,$zero,8          #第1个数=8 (可以修改) 保存到(3072+0)
26      sw $s0,3072($zero)
27
28      addi $s0,$zero,1          #第2个数=1 (可以修改) 保存到(3072+4)
29      sw $s0,3076($zero)
30
31      addi $s0,$zero,5          #第3个数=5 (可以修改) 保存到(3072+8)
32      sw $s0,3080($zero)
33
34      addi $s0,$zero,2          #第4个数=2 (可以修改) 保存到(3072+12)
35      sw $s0,3084($zero)
36
37      addi $s0,$zero,7          #第5个数=7 (可以修改) 保存到(3072+16)
38      sw $s0,3088($zero)
39
40      addi $s0,$zero,9          #第6个数=9 (可以修改) 保存到(3072+20)
41      sw $s0,3092($zero)
42
43      addi $s0,$zero,6          #第7个数=6 (可以修改) 保存到(3072+24)
44      sw $s0,3096($zero)
45
46      addi $s0,$zero,4          #第8个数=4 (可以修改) 保存到(3072+28)
47      sw $s0,3100($zero)
48
49      addi $s0,$zero,3          #第9个数=3 (可以修改) 保存到(3072+32)
50      sw $s0,3104($zero)
51
52      addi $s0,$zero,10         #第10个数=10 (可以修改) 保存到(3072+36)
53
54
55      addi $s0,$zero,3          #第9个数=3 (可以修改) 保存到(3072+32)
56      sw $s0,3104($zero)
57
58      addi $s0,$zero,10         #第10个数=10 (可以修改) 保存到(3072+36)
59      sw $s0,3108($zero)
60
61      addi $s0,$zero,3072       #s0=3072          排序区间开始地址
62      addi $s1,$zero,3108       #s1=3108=3072+10*4-4  排序区间结束地址    10个数    如果不足20个数,这里要修改,例如20个数,这里修改为3148
63
64  sort_loop:
65      lw $a3,0($s0)              #a3=(s0)
66      lw $a4,0($s1)              #a4=(s1)
67      slt $t0,$a3,$a4            #如果a3<a4,则置t0=1;否则,置t0=0    降序排序    从大到小
68      beq $t0,$zero,sort_next    #如果t0=0,则转sort_next
69      sw $a3,0($s1)              #交换(s0)和(s1)
70      sw $a4,0($s0)              #交换(s0)和(s1)
71
72  sort_next:
73      addi $s1,$s1,-4            #s1-4->s1
74      beq $s0,$s1,loop1          #如果s0=s1,则转loop1
75      beq $zero,$zero,sort_loop  #转sort_loop
76
77  loop1:
78      addi $s0,$s0,4             #s0+4->s0
79      addi $s1,$zero,3108        #s1=3108=3072+10*4-4  排序区间结束地址    10个数    如果不足10个数,这里要修改,例如20个数,这里修改为3148
80      beq $s0,$s1,loop2          #如果s0=s1,则转loop2
81      beq $zero,$zero,sort_loop  #转sort_loop
82
83  loop2:
84      beq $zero,$zero,loop2      #转loop2    死循环

```



```

IRQ0:                                     #IRQ0中断服务程序的入口地址: 1024 = 400H      RAM对应100
    addi $sp,$zero,3840                  #push registers 需要保留中断程序用到的寄存器      $s0 $s1
    sw $s0,0($sp)
    sw $s1,4($sp)

    addi $s1,$zero,3328                  #RAM对应340
    lw $s0,0($s1)
    addi $s0,$s0,1
    sw $s0,0($s1)

    lw $s1,4($sp)                        #pop registers
    lw $s0,0($sp)
    eret

IRQ1:                                     #IRQ0中断服务程序的入口地址: 1536 = 600H      RAM对应180
    addi $sp,$zero,3840                  #push registers 需要保留中断程序用到的寄存器      $s0 $s1
    sw $s0,0($sp)
    sw $s1,4($sp)

    addi $s1,$zero,3392                  #RAM对应350
    lw $s0,0($s1)
    addi $s0,$s0,1
    sw $s0,0($s1)

    lw $s1,4($sp)                        #pop registers
    lw $s0,0($sp)
    eret

IRQ2:                                     #IRQ0中断服务程序的入口地址: 2048 = 800H      RAM对应200
    addi $sp,$zero,3840                  #push registers 需要保留中断程序用到的寄存器      $s0 $s1
    sw $s0,0($sp)
    sw $s1,4($sp)

    addi $s1,$zero,3456                  #RAM对应360
    lw $s0,0($s1)
    addi $s0,$s0,1
    sw $s0,0($s1)

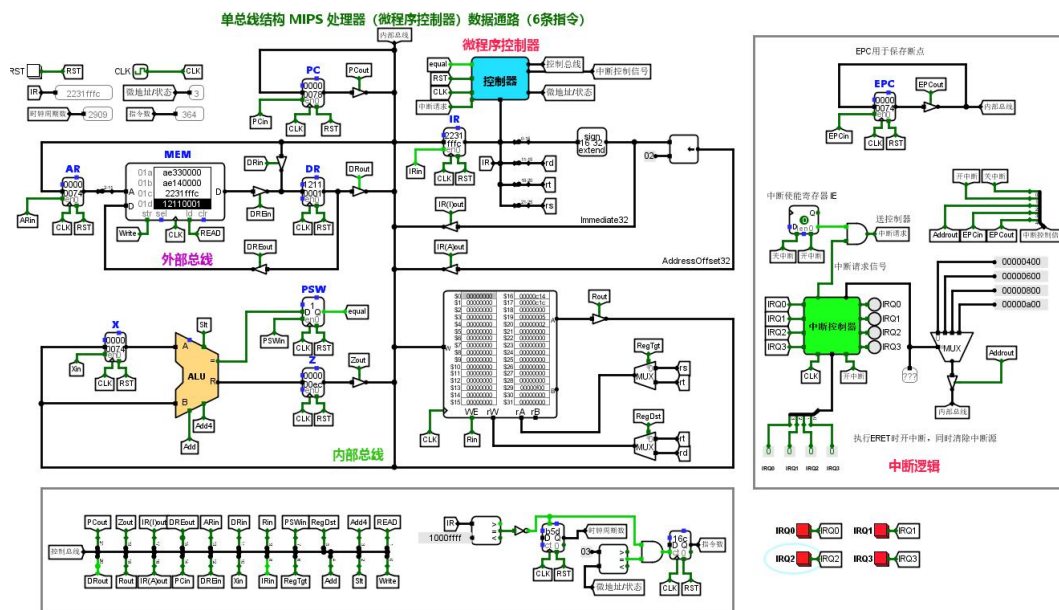
    lw $s1,4($sp)                        #pop registers
    lw $s0,0($sp)
    eret

IRQ3:                                     #IRQ0中断服务程序的入口地址: 2560 = A00H      RAM对应280
    addi $sp,$zero,3840                  #push registers 需要保留中断程序用到的寄存器      $s0 $s1
    sw $s0,0($sp)
    sw $s1,4($sp)

    addi $s1,$zero,3520                  #RAM对应370
    lw $s0,0($s1)
    addi $s0,$s0,1
    sw $s0,0($s1)

    lw $s1,4($sp)                        #pop registers
    lw $s0,0($sp)
    eret

```



请将此电路改造成支持中断的数据通路

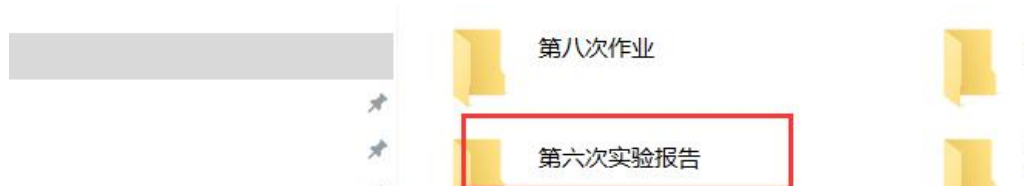
```

300 0000000a 00000009 00000008 00000007 00000006 00000005 00000004 00000003 00000002 00000001 00000000 00000000 00000000 00000000 00000000
310 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
320 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
330 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
340 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
350 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
360 00000002 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
370 00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  
```

2.3 实验提交

- (1). 实验报告命名方式: 例如: **30620192203840+孙明策-6.pdf**
- (2). 实验内容 2.2 中, Logisim 设计文件请单独提供, Logisim 设计文件的命名方式: 例如: **30620192203840+孙明策-6.circ**
- (3). 实验报告上传路径:
 请将所有实验相关文件打包至 .zip 等压缩包, 命名: **学号+姓名-6.Zip**.
 - 计组 (1) (2) 请上传至曾文华老师对应 FTP 路径:
 - 卓越班请上传至下述路径:

Internet > 121.192.180.66 > 上传作业 > 张海英 > 2022年度卓越班计算机组成原理



- 实验提交时间: 实验结束后 2 周内 (**6 月 9 日晚上 24 点前**) 提交实验报告 (Word/pdf 文档), 同时提交相应的设计文件和程序