数字逻辑与处理器基础实验

流水线MIPS处理器设计

2022.6

实验安排和注意事项

A. 课程暑季学期部分占32学时，实验应当在3周内完成。

B. 答疑的时间请同学们留意网络学堂或课程微信群上的安排

C. 提交内容包括实验报告和全部工程代码。实验报告一般使用MS word或pdf格式，内容应当包括：设计方案（原理说明及框图）；关键代码及文件清单；综合情况（面积和时序性能）；实际硬件验证情况和分析；思想体会。

D. 实验应当独立完成，有抄袭等学术不端行为者（实验报告或者设计代码出现雷同、回答问题明显非个人完成等）课程成绩记零分，并上报院系。

实验内容

- 二选一
- 1. 将春季学期实验四设计的MIPS处理器改进为流水线结构，并利用此处理器和理论课MIPS汇编语言大作业中的任意一种算法，求解字符串搜索问题
  - 要点：
    - 外设的设计
    - 流水线中冒险和数据关联问题
    - 测试验证和性能分析
- 2. (1)在单周期MIPS处理器上求解字符串搜索问题 (2)使用数字逻辑电路求解字符串搜索问题
  - 要点:
    - 对（1）和（2）两种实现方式进行比较：资源消耗、求解时间、实现灵活性等

实验内容

- 测试数据
  - 测试数据按字节寻址
  - 测试数据长度应能刻画系统性能
- 测试数据的输入和输出
  - 基本要求
    - 写在汇编指令文件中或者对RAM进行initial初始化
    - 将模式字符串出现次数以16进制的形式显示到数码管上
  - 提高要求
    - 使用UART串口输入字符串，字符串分隔方式自定
    - 将字符串出现次数用UART串口输出
  - 字符串搜索是数据密集型的算法，必要时可对汇编代码进行针对性的优化

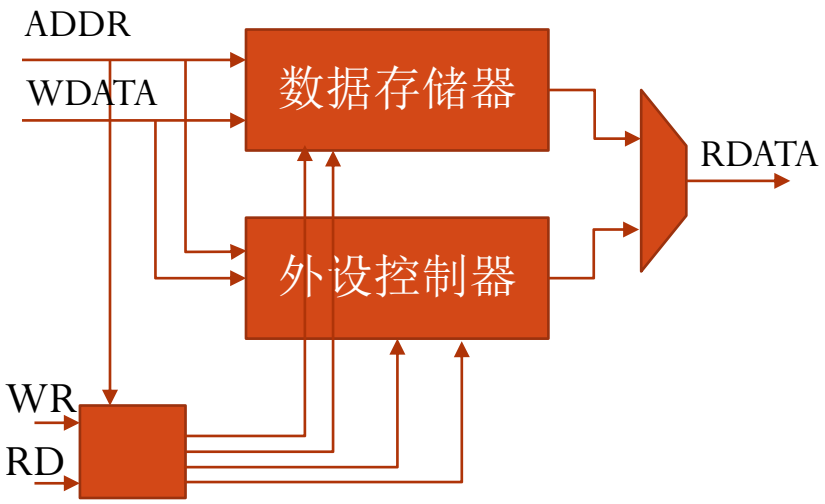
流水线MIPS处理器

- 指令集
  - 空指令：nop （0x00000000,即sll $0,$0,0）
  - 存储访问指令：lw, sw, lui
  - 算术指令：add, addu, sub, subu, addi, addiu
  - 逻辑指令：and, or, xor, nor, andi, sll, srl, sra, slt, sltu, sltiu
  - 分支和跳转指令：beq, blez, bgtz, bltz 和 j、jal、jr、jalr
  - 其他指令可以根据情况自行添加。

外设的控制

- 地址空间
  - 哈弗结构：指令地址空间和数据地址空间是分离的
  - 指令存储器采用ROM实现
  - 数据地址空间包括数据存储器、外设等
  - 数据存储器采用RAM实现，其地址分配如下表

地址范围（字节地址）	功能
0x00000000~0x000007FF	数据存储器
0x4000000C	外部LEDs
0x40000010	七段数码管
0x40000014	系统时钟计数器
0x40000018~20	UART（选做）



外设的控制

- LED、七段数码管、系统时钟计数器

地址范围（字节地址）	功能	描述
0x4000000C	外部LEDs	0bit： LED 0 …… 7bit： LED 7
0x40000010	七段数码管	0bit： CA 1bit： CB …… 7bit： DP 8bit： AN0 9bit： AN1 10bit： AN2 11bit： AN3

外设的控制

```
if(rd) begin
        case(addr)
            32'h4000000C: rdata <= {24'b0,led};
            32'h40000010: rdata <= {20'b0,digi};
            default: rdata <= 32'b0;
        endcase
    end
```

读外设寄存器

写外设寄存器

```
if(wr) begin
            case(addr)
                32'h4000000C: led <= {24'b0,wdata[7:0]};
                32'h40000010: digi <= {20'b0,wdata[11:0]};
            endcase
        end
```

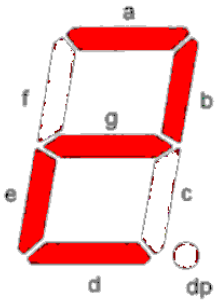
外设的控制

- UART外设（选做）

地址范围	功能	备注
0x40000018	串口发送数据UART_TXD	串口发送数据寄存器，只有低8bit有效；对该地址的写操作将触发新的UART发送
0x4000001C	串口接收数据UART_RXD	串口接收数据寄存器，只有低8bit有效
0x40000020	串口状态、控制UART_CON	2bit：发送状态，每当UART_TXD中的数据发送完毕后该比特置‘1’，当执行对该地址的读操作后，将自动清零 3bit：接收状态，每当UART_RXD中已经接收到一个完整的字节时该比特置‘1’，当执行对该地址的读操作后，将自动清零 4bit：模块状态，0-发送模块处于空闲状态，1-发送模块处于发送状态

自行根据春季学期的UART编写此外设

软件编写提示



- 软件操作提示：
  - i. 将字符串数据导入RAM
  - ii. 完成搜索算法
  - iii. 在数码管上显示出现次数（用软件方式译码显示）

    在数码管最低位上显示数字2为例：

    对数字2，笔端g到a查表可以得到1011011，AN0=1，其他为0。因此向0x40000010地址写入0x015B即可。

    可以利用人眼视觉暂留效应，使用软件延时，每位显示1ms，轮流显示各位数字

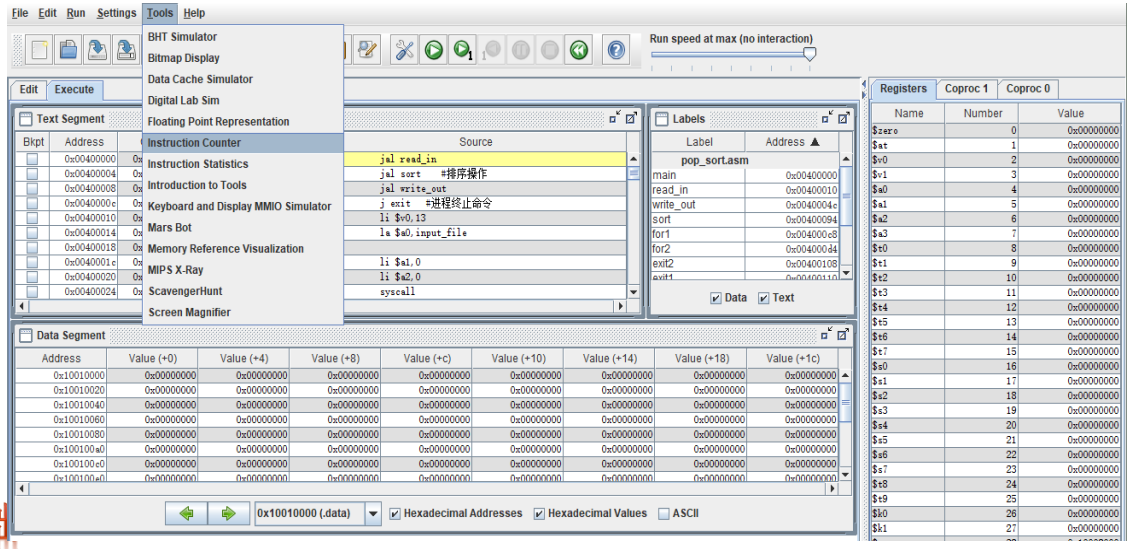
Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应管脚	AN3	AN2	AN1	AN0	dp	g	f	e	d	c	b	a
值	0	0	0	1	0	1	0	1	1	0	1	1

流水线的处理

- 采用完全的forwarding电路解决数据关联问题。
- 对于Load-use类竞争采取阻塞一个周期+Forwarding的方法解决
- 对于分支指令在EX阶段判断（提前判断也可以），在分支发生时刻取消ID和IF阶段的两条指令。
- 对于J类指令在ID阶段判断，并取消IF阶段指令。

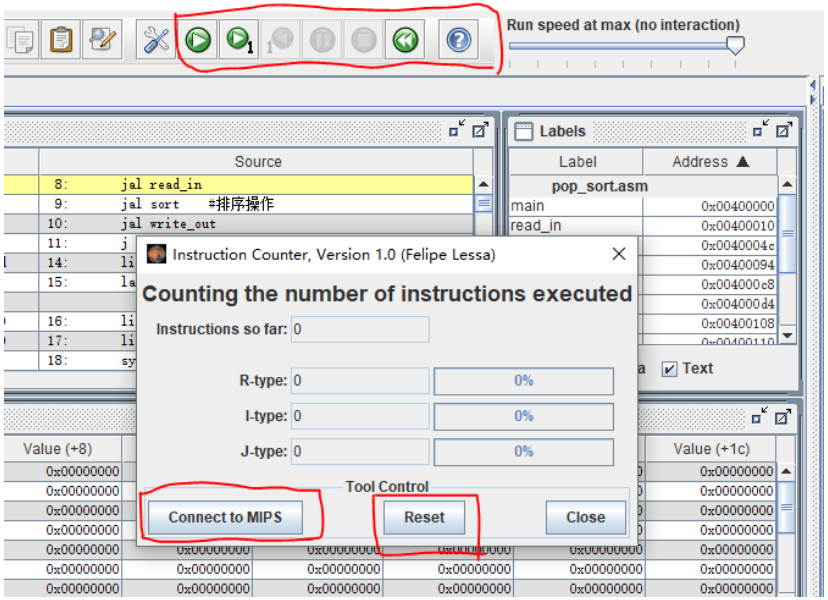
测试验证和性能分析

- 使用MARS等仿真器确定完成字符串搜索算法所执行的指令总数N，通过Verilog代码仿真确定完成字符串搜索算法所消耗的时钟周期数C，计算平均执行一条指令所需要的时钟周期数CPI=C/N，并根据时钟频率计算平均每秒执行指令数目。
- 指令数统计方法
  - 对一个可执行的汇编程序首先进行**编译**，在执行界面选择**Tools**菜单栏，选择**Instruction counter**



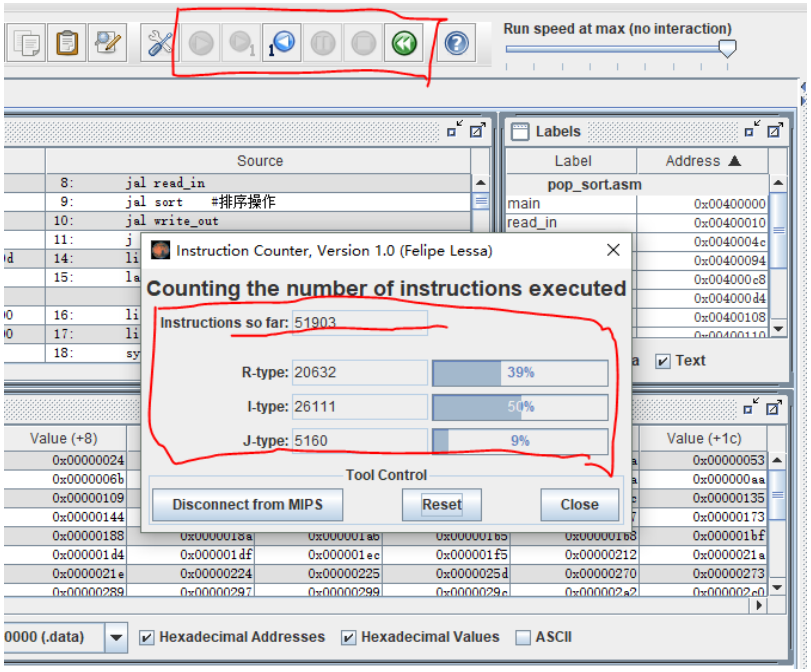
测试验证和性能分析

- 指令数统计方法（续）
  - 点击“Connect to MIPS”，如果看到的指令数不是零，可以先reset，然后执行汇编程序，指令数会随着程序执行变换，支持设置断点，读出 Instruction so far就是程序运行到指定位置的执行的指令数，或者直接到程序运行结束，统计总指令数。



测试验证和性能分析

- 指令数统计方法（续）
  - 最终效果如下图所示，程序运行结束，可以读出实际运行的指令数和指令种类
  - 如果指令数一直增加不会结束，请检查程序是否最终跳入空循环。
  - 实际指令可能与原来的有所不同（如syscall被替代，lw地址变化等）。为方便起见，可忽略这些变化对计算CPI的影响



调试

- 时钟频率：流水线主频以时序报告中的implement时序分析为准，你的设计很可能不能正常工作在100MHz的时钟频率下，注意对输入时钟进行分频，使时钟频率接近但不超过最高工作频率。
- 软件调试：可以先在MIPS的软件仿真器中进行简单仿真，初期也可以利用软件仿真器将汇编代码转换为机器码，我们的指令兼容于标准MIPS32指令集
- 设计时应当考虑LUT、寄存器等资源消耗情况，并对流水线和单周期（多周期）的资源消耗对比进行分析，但资源消耗情况不是对设计进行评分的主要因素

答疑