

# 《计算机组成原理》课程设计

## 教师工作日志（1-2周）

---

**2017-2-13（周一）**

值班教师：谭志虎、胡迪青

主要工作：

1. 讲解 PPT 布置任务（主要是纪律层面讲解）；
2. 部分没有分组的重修生留级生分配任务并分组；
3. 打印台签，建议以后设计一个表格放在一起写文档中让学生自己填，然后集中打印台签，台签中缝还需要调整。
4. 整理工作日志及进度表，录入学生的分组，对应表格放在一起写网站上共享，将各班共享链接放在 tower 平台中，方便学生访问；
5. 根据学生分组制作检查表，放在一个表格中按组用 excelmailer 分割成若干文件后利用 excel 宏批量打印检查表。
6. 创建了教师工作日志合作文档

学生问题：

1. SB，SH 指令如何实现；（1、改造存储器，先读后写，logisim 验证可行，2、直接用多个 8 位存储器扩展，然后利用组合逻辑解决问题，推荐第二种方案，是终极解决方案，对与理解课堂知识有利，当然方案 1 最终在 verilog 中形式化表述时会被综合成方案 2）
2. 中断屏蔽寄存器是否有用；（不改变处理优先级即使嵌套中断这个也没啥用，只需要设置一个中断使能寄存器即可，CPO 可大大简化）
3. 中断隐指令能否用程序完成；（概念错误，中断隐指令就是硬件动作，不是指令完成的，需要完成断点保存和中断识别，设计 PC 保存，中断向量表访问的问题，如果这些都放在存储器中，设计存储器读写个一次，需要两个时钟周期，如果中断向量表用硬件完成，可以节约一个周期）
4. 团队能否分工，能否团队分工开发一个基准版本，再每个人做修改？

(不允许这样的模式，LOGISIM 平台必须自己画图，每个人图应该都不一样，VERILOG 模块可以共享运算器，指令存储器、数据存储器，寄存器组、数码管驱动程序、约束文件代码等，但控制器，流水冲突检测，气泡处理，重定向处理，分支预测等模块不能共享代码)

---

## 2017-2-14 (周二)

值班教师：秦磊华、胡迪青、谭志虎

主要工作：上午值班

1. 提醒学生们开组会，对昨天的工作进行总结，讨论今天的主要工作安排；
2. 解答同学们提出的问题；
3. 发现同学们对中断的处理过程，以及实现这一过程硬件与软件的划分理解的还不是十分清楚，针对这一问题进行了统一讲解；
4. 7-10 班的同学们太安静了，开组会没有声音，平时也都是每个同学自己在埋头做，很少讨论和争论；
5. 工程通常都需要实践来解决，很多同学担心失败和返工，都想搞清楚后再动手，但是由于理解的不透彻，往往是越想越把自己给搞糊涂了；因此在这种情况下还是建议同学们先从简单的地方动手做，在做的过程中再来深入理解，虽然有失败和返工，但是可能效果和效率会更好；
7. 少数同学还是没有养成来了就签到的习惯，还是需要老师不断的提醒。
8. 阅读了所有学生的工作日志，提醒部分学生及时更新日志，锁定了第一天的日志，同时将没用写日志的标红。（该部分工作应该由每个班的指导老师完成）
9. 撰写课程设计教师手册；
10. 打印学生分组检查表；（用 excelmailer 分割成若干文件，然后用宏文件全部打印）

学生问题：

1. 中断嵌套要求到底是什么（多级中断要求：系统支持 3 个按键中断源， $3 > 2 > 1 > \text{CPU}$ ，由按键触发中断演示程序，3 号键对应自己特殊指令的测试程序演示，2 号键显示数值 2 左右移动的跑马灯，往返 6 次，1 号键现实数值 1 左右移动的跑马灯，往返 6 次。3 号中断可以打断 1 号，2 号

中断程序。主程序就用原来的 benchmark 配上一些初始化代码。

2. 主要的问题还是中断处理，到底硬件要做什么，那些交给软件来做不清楚，另外就是单级中断还没有实现就考虑中断嵌套问题，结果反而把自己给搞糊涂了。
3. 有些同学对 CP0 还是没有理解，不知道 CP0 的用途，有些同学仅仅将 CP0 理解为通用寄存器，还有些则认为 CP0 高深莫测，这都不对。
4. 有不少同学用硬件来做堆栈，用于在中断嵌套情况下保存返回地址，这样做其实是不合理的。

持续改进：

1. 在任务书中明确中断要求，最后的演示测试效果；
2. 建议加入单级中断的过程，避免学生钻入死胡同；
3. 团队无法开展有效讨论是否与随机分组相关，有可能很多平时不熟悉或者很少合作，第 3-4 周可以考虑按照 NBA 选秀模式自由组队，让小组熟悉的小伙伴更多，看是否有变化；

**完成多级嵌套中断必须思考解决的系列问题，将来可以放在任务书中！**

**(请各位老师补充)**

1. 函数调用 JAL 与中断有多大区别，有否共同之处？
2. 不同的中断请求存储在哪里，何时消失？
3. 硬件响应优先级用什么电路实现，为什么要有处理优先级。
4. 中断屏蔽寄存器有什么作用，何时设置中断屏蔽字，真实计算机环境中由什么程序设置中断屏蔽字？本实验是否需要中断屏蔽寄存器？
5. 中断使能寄存器是干什么用的？
6. CPU 如何判断当前有中断需要响应？
7. CPU 发现当前存在中断事件后要做什么动作，什么时候响应中断事件？哪些是硬件完成，哪些是软件完成？由硬件完成的动作需要多少个时钟周期，此时 CPU 能否执行指令？
8. 单级中断断点保存在哪里，多级嵌套中断的断点如何处理？
9. 中断处理程序中的现场有哪些，我们实验中需要考虑保存哪些现场？

10. 中断程序入口地址如何识别？硬件还是软件完成？哪种方案比较好，为什么？
  11. 开中断，关中断在 MIPS 指令集中如何实现？
  12. 中断使能寄存器 IE 有什么作用，在 MIPS CPU 中如何实现？
  13. 中断处理程序放在指令存储器中的那个位置，如何载入到 ROM 中？
  14. 数据堆栈放在哪里？SP 寄存器如何设置？MIPS 如何访问堆栈？
  15. 按键中断是电平触发还是跳变触发？连续按键如何处理？实际系统中是如何处理的？
  16. 高优先级中断服务程序执行过程中，有新的按键事件发生，如何处理？
  17. 实验中的中断机制为啥要用 CP0，不要是否可以？在我们的实验中如何简化？
  18. 单周期 CPU 中断处理和流水中断处理有何区别？
- 

## 2017-2-15（周三）

值班教师：谭志虎、吴非

### 1. 函数调用 JAL 与中断有多大区别，有否共同之处？

JAL 与中断的处理过程及实现的功能类似，都是将 PC 保存到某个寄存器，然后修改 PC 的值。不同的地方在于：JAL 将 PC 保存到 31 号通用寄存器，中断则将其保存到另外一个特殊的寄存器；修改后 PC 的新值，在 JAL 中由指令给出，中断则直接由硬件确定；JAL 是一条指令，用户在程序中使用，当 PC 指向该指令时执行，中断则不是一条指令，而是类似指令的功能，因此也叫隐指令，它是在满足中断响应条件时执行（外部中断信号有效，中断使能，且中断没有被屏蔽）。

另外二者进入具体函数时都需要保存现场，这里的现场就是被调用者保存寄存器，一般不同指令系统会事先约定好，便于编译器生成代码。

### 2. 不同的中断请求存储在哪里，何时消失？

不同的中断通常保存在一个特殊的寄存器中，一般寄存器的 1 位对应一个中断请求。

该中断请求当被中断响应处理时消失，一般是在中断服务程序中处理器通过软件将这个寄存器对应的位清零。（这个好像有问题，应该是硬件清零吧，软件中没有对应指令干这个事情）

### 3. 硬件响应优先级用什么电路实现，为什么要有处理优先级。

硬件响应优先级一般用优先级编码电路来实现，这样优先级高的中断请求就会最先被处理器响应。只能没有更高的中断请求时，电路才会把较低优先级的中断请求送给处理器，优先编码器电路在 logisim 中有现成的模块。

### 4. 中断屏蔽寄存器有什么作用，何时设置中断屏蔽字，真实计算机环境中由什么程序设置中断屏蔽字？本实验是否需要中断屏蔽寄存器？

中断屏蔽寄存器的作用是保存中断屏蔽字，中断屏蔽字的用途是当某个中断请求正在被处理时，调整中断服务的顺序；正常情况下，优先级高的中断请求在被处理时，即当它的中断服务程序在执行时，是不允许被它同级或更低级别的中断请求打断的（当然比它级别高的中断请求是可以的），但是通过设置中断屏蔽字，可以允许它被低级别的中断请求打断，从而先完成低级别中断请求的中断服务程序，然后再执行完它自己的中断服务程序。（自己不会让低优先级打断自己吧）

（我的理解：响应优先级是电路固定的，不可改变的，为了增加灵活性，可以通过中断屏蔽字解决处理优先级调整的问题，但具体调整肯定是操作系统负责这个事情，不能由设备的中断程序来管这个）

中断屏蔽字是在中断处理过程中设置的，在真实计算机环境下，通常是由中断服务程序进行设置。

由于本实验在高级别中断请求在处理时不允许被低级别的中断请求打断，因此其实不需要中断屏蔽寄存器。

### 5. 中断使能寄存器是干什么用的？

中断使能寄存器其实就像是一个开关，用来打开或关闭处理器的中断，外部设备的中断请求信号会与这个开关做与操作。所以中断使能有效时，处理器才能响应中断请求；当无效时，处理器不会响应任何中断请求，软件中开中断，关中断指令 X86 STI CLI 就是控制这个开关的，MIPS 中必须利用 CPO 访问指令来控制对应的位，具体位在 MIPS 中有约定，但实验中实现的时候可以自己灵活调整。

### 6. CPU 如何判断当前有中断需要响应？



有外部中断请求，中断使能有效，同时该中断请求没有被屏蔽，**当前指令执行完毕**，适合上述四个条件 CPU 就应当在合适的时候去响应中断。

7. CPU 发现当前存在中断事件后要做什么动作，什么时候响应中断事件？哪些是硬件完成，哪些是软件完成？由硬件完成的动作需要多少个时钟周期，此时 CPU 能否执行指令？

CPU 发现当前有中断事件后，在可以响应中断时要完成下面这些动作：（1）**保存断点**，打断的指令的 PC 值；（2）**中断识别并跳转**，根据中断源找到中断程序入口地址，修改新的 PC 值转到中断处理程序；（3）完成中断服务的准备工作；（4）执行中断服务程序；（5）结束中断服务，完成中断返回的准备工作；（6）中断返回。

其中（1）和（2）由硬件完成，也就是所谓中断隐指令干的事情，这部分需要一个或者多个时钟周期，具体与实现有关，其它则是软件实现。

由于硬件需要完成（1）和（2）二步工作，可以 1 个时钟周期完成，也可以用 2 个时钟周期来完成，这个过程中 CPU 其实是相当于在执行指令（见问题 1，类似于 JAL 指令）因为指令的执行也就是建立相应的数据通路，因此此时 CPU 不能执行其它指令。

8. 单级中断断点保存在哪里，多级嵌套中断的断点如何处理？

单级中断的断点通常保存在一个专用的寄存器破口中，就 MIPS CPU 而言，是保存在 CP0 协处理器中的一个叫做 EPC 的寄存器中。由于 EPC 只是一个寄存器，因此当多级中断嵌套时，当有新的中断断点要保存时，需要将保存在 EPC 中的老的断点保护起来，通常是将其压入堆栈之中，**这部分应该是在保护现场阶段做的，EPC 实际上也是一个被调用者保存寄存器**。（类似 JAL 嵌套调用，需要保护 \$ra 寄存器一样）

9. 中断处理程序中的现场有哪些，我们实验中需要考虑保存哪些现场？

理论上中断处理时 CPU 要转去执行另外一个程序，执行完后返回再继续执行被打断的程序，因此将改变 CPU 状态的任何操作之前，要被改变的状态都需要保存起来，返回再恢复。CPU 的状态都是通过寄存器实现的，所以所谓保护现场就是要备份将修改的寄存器的值。我们实验中需要考虑保存的现场主要有：中断返回地址 EPC、**中断请求信息（）**、中断屏蔽信息、以及在中断服务程序中将使用的通用寄存器。总之，记住一个原则，你将修改什么，在修改之前最好就保存什么。

10. 中断程序入口地址如何识别？硬件还是软件完成？哪种方案比较好，为什么？

中断程序入口地址可以用硬件识别，也可以用软件识别；硬件识别就是不同的中断请求，会转到不同的中断处理地址（即在问题 7 的第二步时会由硬件修改不同的 PC 值）；软件识别就是中断请求后，CPU 响应中断时会转到统一的入口地址，然后由软件通过读取中断请求寄存器中保存的中断请求信息来判断应该怎么处理，再转到对应的中断服务程序。

硬件识别就是中断响应的速度快，但是不够灵活，同时需要付出硬件成本的代价；软件识别的好处是灵活、方便、成本较低，但是缺点是中断响应的速度比较慢，不太适合用于实时环境下。

（学生此处会走捷径，没有中断向量表，中断向量表直接用硬件电路完成，组合电路可以算出中断服务程序入口地址），其实也无所谓。

## 11. 开中断，关中断在 MIPS 指令集中如何实现？

MIPS 处理器没有专门的开中断、关中断指令，因此是用软件实现的。利用 CPO 寄存器组访问指令实现。

## 12. 中断使能寄存器 IE 有什么作用，在 MIPS CPU 中如何实现？

MIPS 中有一个特殊的寄存器（在 CPO 中），这个寄存器的某一位叫做 IE，IE 为 1 时会控制下面的电路，使得有外部中断请求时可以送到 CPU，从而使 CPU 能够响应中断；当 IE 为 0 时，外部中断请求就不能送入 CPU，因此 CPU 也就不能响应中断，从而实现关闭中断的功能。

MIPS CPU 则可以通过下面的方式来修改 IE 寄存器，从而实现开中断、关中断处理：用 MFC0 指令将 IE 的值放到一个通用寄存器，修改这个通用寄存器（根据需要置 1 或 0），然后用 MTC0 指令将这个值从通用寄存器写回 IE 寄存器，从而完成 IE 的控制

## 13. 中断处理程序放在指令存储器中的那个位置，如何载入到 ROM 中？

中断处理程序放在指令存储器的一个特定位置，它的起始地址必须与中断响应时 CPU 赋予 PC 的值相同，按照习惯中断处理程序的入口地址通常是存储器最低的地址，即 0x00000000。中断处理程序一般是与主程序一起编译，然后统一做成存储器文件用原来相同的方式载入 ROM。（放哪应该无所谓，实验中自己安排，放低地址 CPU 上电 PC 初始值要改）

#### 14. 数据堆栈放在哪里？SP 寄存器如何设置？MIPS 如何访问堆栈？

数据堆栈通常放在数据存储器中。MIPS 处理器的堆栈是向下生长的，因此 SP 寄存器初始指向存储器的最高地址，每保存一个值，SP 要减 4；出栈时则 SP 加 4。例如：

```
# 设置 SP 指针
li sp, STACK_BASE_ADDR

# 入栈
addiu sp, sp, -4
sw      ra, 0(sp)

# 出栈
lw      ra, 0(sp)
addiu sp, sp, 4
```

#### 15. 按键中断是电平触发还是跳变触发？连续按键如何处理？实际系统中是如何处理的？

按键一般是边沿触发，但是处理边沿中断触发通常较为困难，因此常用锁存器将边沿触发转换为电平信号，然后用电平中断触发。连续按键最简单的处理办法是在第一次按键没有处理完之前，不管连续的按键信息，即发生按键时锁存器保存，然后这个锁存器就锁住，不再保存新的按键信息，只有当 CPU 把这个按键中断处理完毕后，再开放这个锁存器。实际系统中这个锁存器是一个缓存，可以保存按键的队列信息，这样 CPU 就可以一个一个的处理，直至保存的按键缓存队列全部被处理。（实际系统运行是中断处理非常快，操作系统也规定中断服务程序不能太长，否则系统会 down 掉，连续按键都可以得到及时响应）

#### 16. 高优先级中断服务程序执行过程中，有新的按键事件发生，如何处理？

由硬件判断新的按键的优先级的高低，如果比当前中断服务程序的级别高，就打断当前的中断服务程序（类似于打断主程序），转而执行这个更高级别的中断服务程序；如果比当前的中断服务程序级别低，则保存这个按键事件信息，等当前的中断服务程序执行完毕并返回后，再来处理这个按键中断；如果是与当前的中断服务程序级别一样，可以不管它，也可以像上面一样类似处理。（CPU 响应不响应取决与能否收到中断请求信号，如果收不到就不管）



17. 实验中的中断机制为啥要用 CP0，不要是否可以？在我们的实验中如何简化？

之所以用 CP0 是为了方便程序的编译（因为要用标准的 MARS 工具来编译程序，因此符合 MIPS 规范的实现方式就比较方便）。其实是可以简化的，另外再增加 1-2 个寄存器就可以解决问题。开中断关中断等与中断相关的操作必须访问除 32 个通用寄存器以外的寄存器，而通用指令中只有 5 位表示寄存器，所以无法访问与中断相关的寄存器，所以必须借助 CP0 寄存器访问指令来实现中断相关的寄存器访问。实验中 CP0 寄存器可以不遵守 MIPS 规范，只要能实现中断相关机制即可。

18. 单周期 CPU 中断处理和流水中断处理有何区别？

在单周期 CPU 中，每指令执行完毕后，CPU 就可以去判断是否有中断需要处理，如果有就进行中断处理，如果没有则正常执行下一条指令。但是在流水线中就比较麻烦一些，因为在流水线中同时有 5 条指令在执行，这 5 级指令还分别处于不同的处理阶段，即 IF、ID、EXE、MEM、WB 各有 1 条指令在执行，另外甚至同时有两条指令都进入了结束阶段，什么地方处理中断，中断哪条指令，同时执行的其它指令怎么办都要考虑。理论上，5 个流水段的任意一个段上都可以处理中断，为了实现的方便通常会选择一个段来处理中断，可供选择的段常常是 EXE 或者是 WB。但是，不管选择那个流水段来处理，都要做下面的事情（以在 EXE 段处理中断为例，其实在 WB 段处理更加方便）：（1）EXE 段前面的 2 条指令要继续执行完；（2）EXE 段后面进入流水线的 2 条指令要取消；（3）处于 EXE 段本身的这条指令有 2 种选择，如果这条指令允许继续执行完，返回的断点地址应该是该指令的 PC+4；如果不允许这条指令继续执行，返回的断点就是这条指令的 PC；（通常做法是不允许这条指令继续执行）；（4）暂停流水线进行中断响应；（根据具体的实现方法也可以不暂停）（5）重新启动流水线从新的 PC 值处取指令（新的 PC 即是中断入口地址）。

最后大赞以下狐狸的精彩解答，呵呵，这个解答可以帮助学生和老师理清思路，大家还可以增加一些问题？

主要工作：

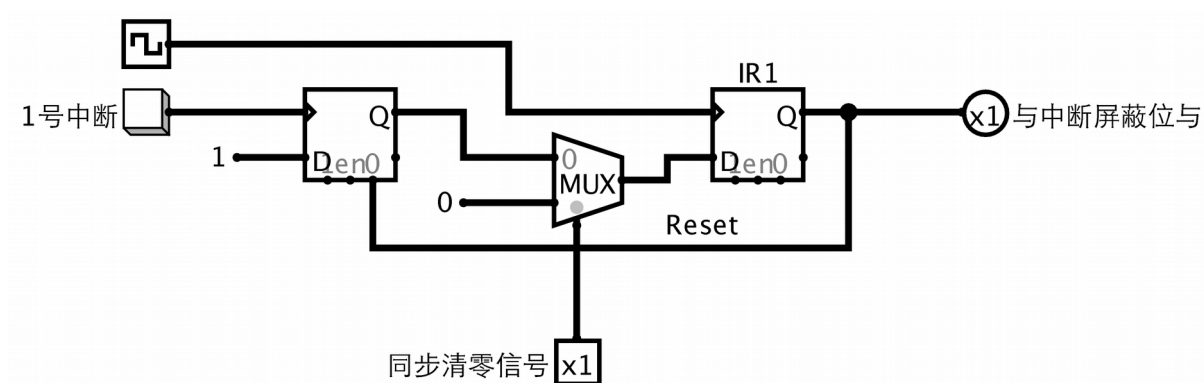
1. 胡迪青老师完成了对 18 个问题的解答，谭志虎老师进行了修订；

2. 审阅了所有学生的工作日志，基本进度还在中断卡着，估计今天讨论了以后会好一些。
3. 要求学生提交小组会议记录，通过『一起写』平台完成，提交链接，一起写平台不错，挺方便的，以后各种需要学生合作完成的内容都可以通过这个平台发布；
4. 与 tower 客服联系解决了项目组无法上传文件的问题，后天进行了文件系统的维护，权限丢失引起无法上传。
5. 学生进度不理想，胡迪青老师完成了 18 个问题的初步回答，下午发布在 tower 平台。
6. 简化后的中断响应过程可能是零开销，当接收到中断请求信号，直接让 PC 选择由硬件向量表产生的中断入口地址，同时将 PC\_Next 压入 EPC，同时将具体中断请求信号同步清零，实际上中断响应只需要一个周期，但这个实践和上一条指令是并发的过程，所以等同于不需要时间！！！（请大家勘误）

7.

学生问题：

1. 产生中断信号后怎么能让信号在规定时间内消失，他们认为做不出来，呵呵。（时序逻辑电路即可，如果中断响应需要多个周期，就必须要有状态机配合，进入某个状态清除信号，注意清除中断请求信号应该是时钟同步清零，否则可能会产生问题）
2. 如何产生中断信号？见下图，图中引入了中断信号的同步清零电路，这个是学生设计过程中的难点，最后在 QQ 群公布了答案，学生还有一些其他的方案。（此电路有故障，答案见后）



## 2017-2-16 (周四)

值班教师：谭志虎、胡迪青

主要工作：

1. 今天上午才有一位同学完成了多周期中断的检查；
2. 下午开始陆陆续续有不到 1/3 的人检查；
3. 两个课堂分别讲解了流水线阶段的任务和注意事项；
4. 要求学生今天务必结束中断的内容，周五开始务必转战流水线部分的工作，否则时间点保证不了。
5. 准备周六上午推送软件到机房，思考安装哪些新软件
  - 屏幕共享软件，避免学生后排看不清投影的情况：经反复对比，使用 desktopshare 软件，比 inletexEMC 颜色要好，速度也不错，具体效果要等推送后观察。
  - 局域网文件传输，方便内部文件传送，dukto 软件比较方便点对点传送，不能一对多，bittorrent sync 可以方便多机同步，可以考虑安装部署
  - office 2016 升级正版
  - notepad++ 『sublime』 『github windows』 『codeblocks』 『dev C++』 『坚果云』 『visual studio』 『vscode』
  - 桌面网线不通
  - 和机房人员沟通，看哪些软件不要都删掉，360 系列包括浏览器，保留 chrome 即可

持续改进：

1. 中断处理时不能再接受同级中断请求，建议在后面明确中断规格，以及检查验收标准，将检查过程中对应的指示灯，标识都规定好，方便检查。
  2. 建议中断设计中增加单级中断单中断源阶段，方便学生一步步的完成任务，另外给出中断请求部分的电路，扫清技术障碍，加快学生进度，正常情况应该周三有人就能够完成中断，周四大面积检查比较理想。
- 

## **2017-2-17（周五）**

值班教师：姚杰、胡迪青、谭志虎

主要工作：

1. 继续进行单周期任务的检查；
2. 部分同学已经完成了理想流水线的任务。

学生问题：

1. 流水线是否要实现延迟槽功能？（不需要，尽量简化）
  2. 理想流水线测试只有 17 个周期？（正确 21 个周期，可能与停机指令立即执行有关，建议将对应信号送到最后段）
- 

## **2017-2-18（周六）**

值班教师：胡迪青、谭志虎、管涛

主要工作：

1. N1 楼机房推送 vivado lience
2. 改造机房软件环境；
3. 安装完成了“桌面共享”，“局域网传文件”，“Resilio SYNC”等机房迫切需要的软件。
4. 安装了 notepad++, sublime, visual code, 360 压缩，火狐，code blocks 等
5. 删除了 360 卫士，360 杀毒等软件。

6. 最后发现机房机器慢的主要原因还是澳易还原卡的原因。

后续工作：

1. 安装 IE 11，需要安装 Win7 SP1，安装迅雷。
  2. Resilio Sync 需要重新配置，设置“机房内交换区”“教师推送区”两个文件夹，需要再教师机上安装不同的密钥，然后保存。
- 

## 2017-2-20（周一）

值班教师：谭志虎、胡迪青

主要工作：

1. 部分学生已经完成了重定向流水线并通过了检查，1-4 班大约有 20 人完成了这部分工作，其中有两人完成了流水中断，一人据说完成了分支预测；7-10 班完成情况较差。
2. 根据学生检查的情况制作了不同策略流水线周期数的表格，方便学生对比测试。
3. 机房网络维护，网络工程师修复了桌面的网线，学生可以通过网线上网了。
4. 机房软件继续更新，IE 11 经过 win7 sp1，sp2 等升级后终于成功安装，resilio Sync 软件可以直接运行了，考虑传播文件带来的潜在政治安全问题，决定在学生机只安装只读密钥，用于接受教师推送的文件。
5. 安装了 visual studio 2015，microsoft 2016，markdownpad，atom
6. 明天需要继续安装 gvim，sublime package install，git 工具等，检查 vivado license 问题。
7. 今天晚上 3 班邓翔同学已经成功上板子（该同学没有做 logisim 版本，直接用 iverilog +gtkwave 完成的设计，最后 vivado 时序仿真）

学生问题：

1. 流水中断是否可以用软件方式解决冲突？（为降低难度，可以用这种方法解决，以后课程设计过程相对清晰后再转成硬件处理）



## 2. 重定向周期数到底是多少?

- 如果所有分支指令都在 EX 阶段完成，2298，无条件跳转指令 38 条，其中 37 条 “J”+1 条 “JR”，Load-Use 120 次，有条件分支成功次数 314-38
  - 总周期数=总指令条数+流水充满时间+分支次数\*流水误取深度+Load-Use 次数
  - $2612=1546+4+314*3+120$ （在 MEM 阶段完成分支，误取深度 3）
  - $2298=1546+4+314*2+120$ （在 EX 阶段完成分支，误取深度 2）
  - $1984=1546+4+314*1+120$ （在 ID 阶段完成分支，误取深度 1，此处设计 ALU 输出端到 ID 段重定向，有点类似 Load-Use 重定向的错误，但考虑 EX，ID 电路延迟较短，所以这种方案也可以接受，实际 MIPS CPU 也是这样完成的）
  - 也有部分同学 ID 段 执行分支指令时用气泡技术解决数据冲突，**这种方案中周期数为 1984+22（待查实）**
- 

## 2017-2-21（周二）

值班教师：谭志虎、秦磊华、胡迪青

主要工作：

1. 整理了近期的考勤状况，对还没有检查的同学做了摸底，部分同学还卡在中午没有转场。发现很多人带签到。
2. 今天 1-4 班 28 人完成 logisim 重定向，6-10 班 24 人完成，应该还算不错
3. 安装完成 git 工具，sublime-package control，gvim

学生问题：

1. 总周期数总少两个（停机信号未传递到 WB 段，导致时钟停掉，计数有问题）
2. QQ 群有很多人翻译小组帮助不大，各做各的，很多同学伸手党，不思考就问问题。

持续改进：

1. 5-6 班课程设计时建议自己组队，模式待定，4 个种子来选秀吧。
  2. 小组最佳贡献者建议必须过半数，否则无效。
  3. 小组每日工作日志应该增加每个人的进度，不能过简。
  4. 签到代签的问题作为诚信问题，以后出现一次，平时成绩扣光，另外为避免恶意代签，增加微助教请假签到。
- 

## 2017-2-22 (周三)

值班教师：谭志虎、秦磊华、胡迪青、姚杰

主要工作：

1. 考勤仍然有极小部分同学没到，也有几个同学来太晚而没有签到。
2. 整理了 7-10 班实验完成进度，其中 8 班同学实验进度较慢，作了督促；
3. 今天 7-10 班新增 24 个同学完成 Logisim 重定向流水线阶段，新增 5 位同学完成单周期中断阶段，新增 1 位同学完成 FPGA 阶段，同时也实现了扩展指令测试；

学生问题：

1. SB，SH 指令如何实现：虽然 LB，LH 指令可以通过分线器的办法实现，但是 SB、SH 指令并不能简单的通过分线器实现。我们的建议，还是使用 4 片独立的 8 位 RAM 通过位扩展的方式组成 32 位的 RAM，再通过字节址对每片 RAM 进行寻址。
2. verilog 对 rom 指令文件的包含问题：有两点要注意，一是最好使用绝对路径，二是要转义，例如“d:\\1\\benchmark.rom”。不然综合会报警告，同时仿真结果出错。

主要存在问题：

1. 普遍反映 FPGA 的调试不如 Logisim 方便，出现问题往往要花费较多精力。建议我们整理一个常见问题调试方法文档，以帮助同学们解决遇到的问题。
-

## 2017-2-23 (周四)

值班教师：谭志虎、秦磊华、胡迪青

主要工作：

1. 对整个课程设计的完成情况进行了比较全面的排查，了解同学们的整体进展情况；
2. 对如何在 Verilog 语言中进行 ROM 初始化的多种方法进行了集中讲解，让同学们能够在进行 FPGA 实现时比较容易的将测试程序装入 ROM 中；
3. 解决了学生提出的一些问题，主要是 Verilog 语言综合和仿真时出现的一些特殊或奇怪的问题。

学生问题：

1. 无法进行综合和仿真（通常是因为 ROM 或 RAM 设置的过大，造成无法实现；另外就是 Verilog 程序代码有错或者是书写不合理）
2. CPU 在 FPGA 上实现的功耗比较大（主要原因是电路描述不合理）

主要存在的问题：

1. 在进行 Verilog 实现时，大多数同学不设置复位信号，程序下载后就开始自动运行，需要重新检查时又再次进行程序下载；
2. 先做 Logisim 再上 FPGA 的好处是同学们比较容易上手，但是缺点是往往直接按照 Logisim 中设计的结构一对一的搬到 FPGA 上，没有充分运用 Verilog 语言的特点来简化和优化电路的实现；
3. 由于 Verilog 是程序代码，方便许多同学直接借鉴别人的代码，如果自己努力消化后再根据自己的情况进行修改还好，如果不能搞清楚的话，反而往往带来不少问题。

---

## 2017-2-24 (周五)

值班教师：谭志虎、姚杰、胡迪青、吴非、秦磊华

主要工作：

1. 今天主要任务是检查，完成率还不算高，看周一检查的情况；
2. 最后截止日期推迟到下周二上午；

3. 周末发现还有好多学生在做组原课设，看来最后的分数学生还是很看重的。