**计 算 机 体 系 结 构（研 讨 课）实 验 报 告**

实验项目 prj6 小组编号 28 组员姓名 刘景平、张钰堃、付博宇

1. **实验目标**
2. exp17：

1.设计 TLB 模块。

2. 利用 TLB 模块级验证环境对所设计的 TLB 进行验证，通过仿真和上板验证。

1. exp18：

1.将实践任务 17 完成的 TLB 模块集成到实践任务 16 完成的 CPU 中。

2.在 CPU 中增加 TLBSRCH、TLBRD、TLBWR、TLBFILL、INVTLB 指令。

3.在 CPU 中增加 TLBIDX、TLBEHI、TLBELO0、TLBELO1、ASID、TLBRENTRY CSR 寄存器。

4.在采用 AXI 总线的 SoC 验证环境里完成 exp18 对应 func 的功能验证，要求成功通过仿真和上板验证。

1. exp19：

1.为 CPU 增加 TLB 相关异常：TLB 重填例外、load/store/取指操作页无效例外、页修改例外、页特权等级不合规例外。

2.在 CPU 中增加 DMW CSR 寄存器。

3.为 CPU 增加虚实地址映射的功能。

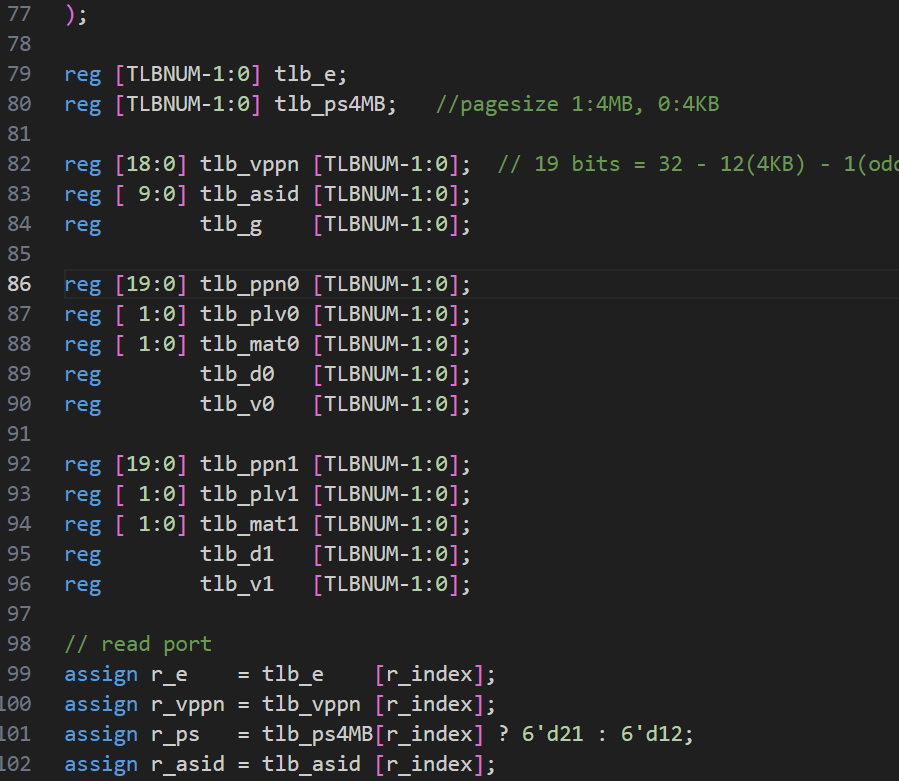
4.在采用 AXI 总线的 SoC 验证环境里完成 exp19 对应 func 的功能验证，要求成功通过仿真和上板验证。

1. **逻辑电路结构与仿真波形相关说明**
2. tlb模块设计

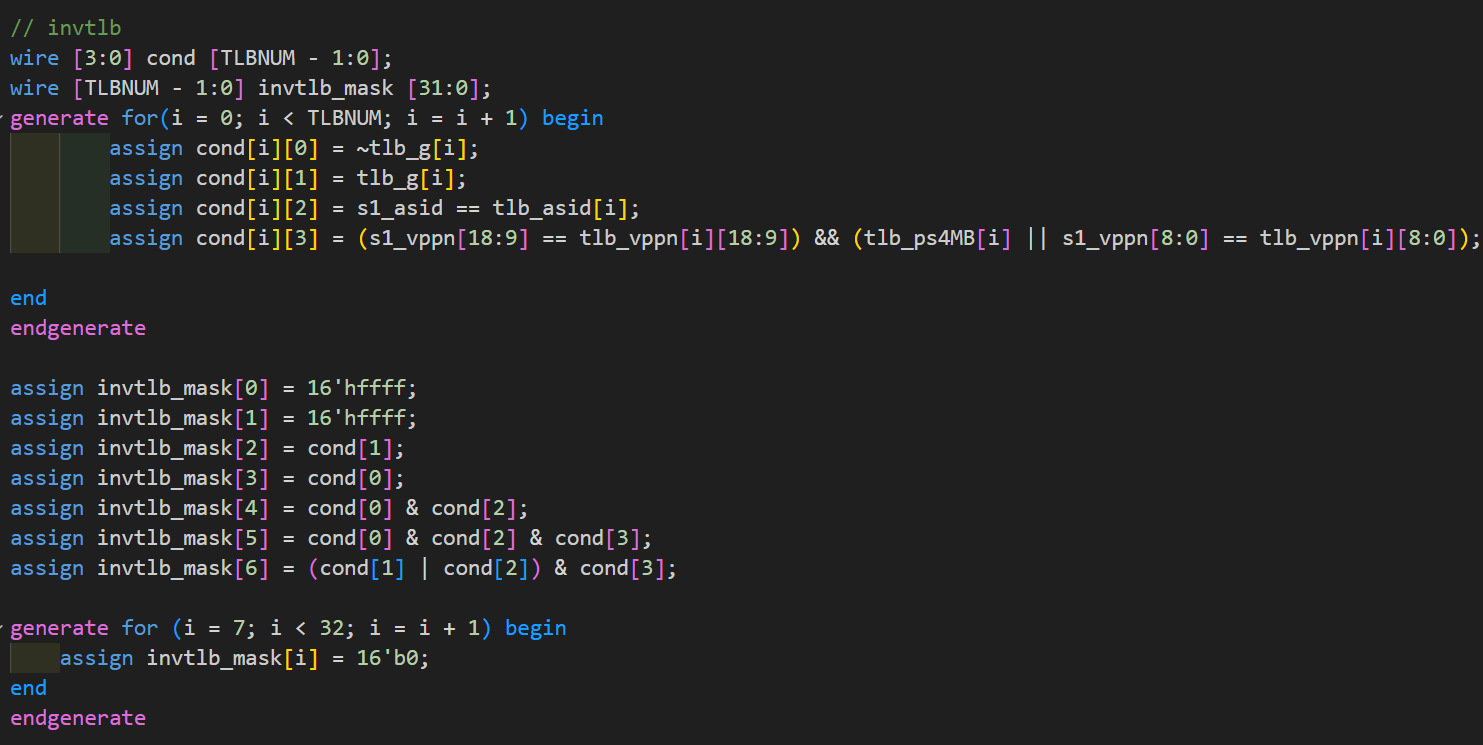
·tlb模块的输入输出端口和内部寄存器设置：

按照讲义216-218页给出的定义，设置两个查找端口（分别用于取指和访存）、一个读端口和一个写端口，另外添加专门用于执行invtlb指令的两个输入信号。

按照loongarch架构对于tlb页表项的定义设置寄存器堆。



·invtlb的处理：



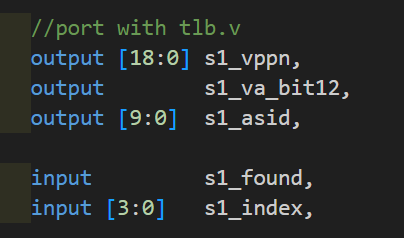
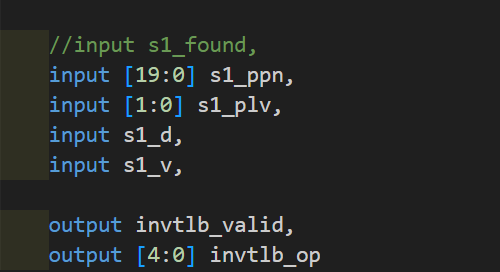
按照讲义上介绍的方法，处理invtlb指令的时候为32个tlb页表项增加一个四位的cond向量，每一位分别表示该页表项是否符合四种condition的某一个；并为每个页表项增加一个7位长的mask向量，其中每一位对应该页表项是否符合invtlb—op的一种情况。

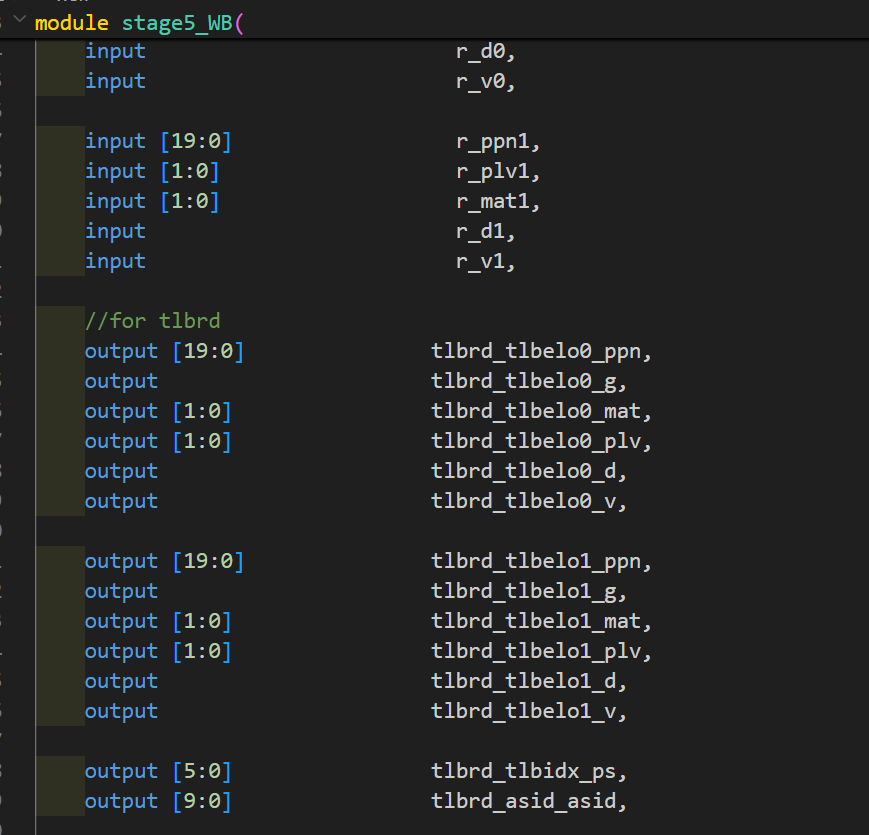
在需要进行invtlb擦除操作时，根据invtlb-op进行对符合条件的页表项进行擦除。

1. tlbsrch、tlbrd、tlbwr、invtlb指令

·四条指令与tlb的交互

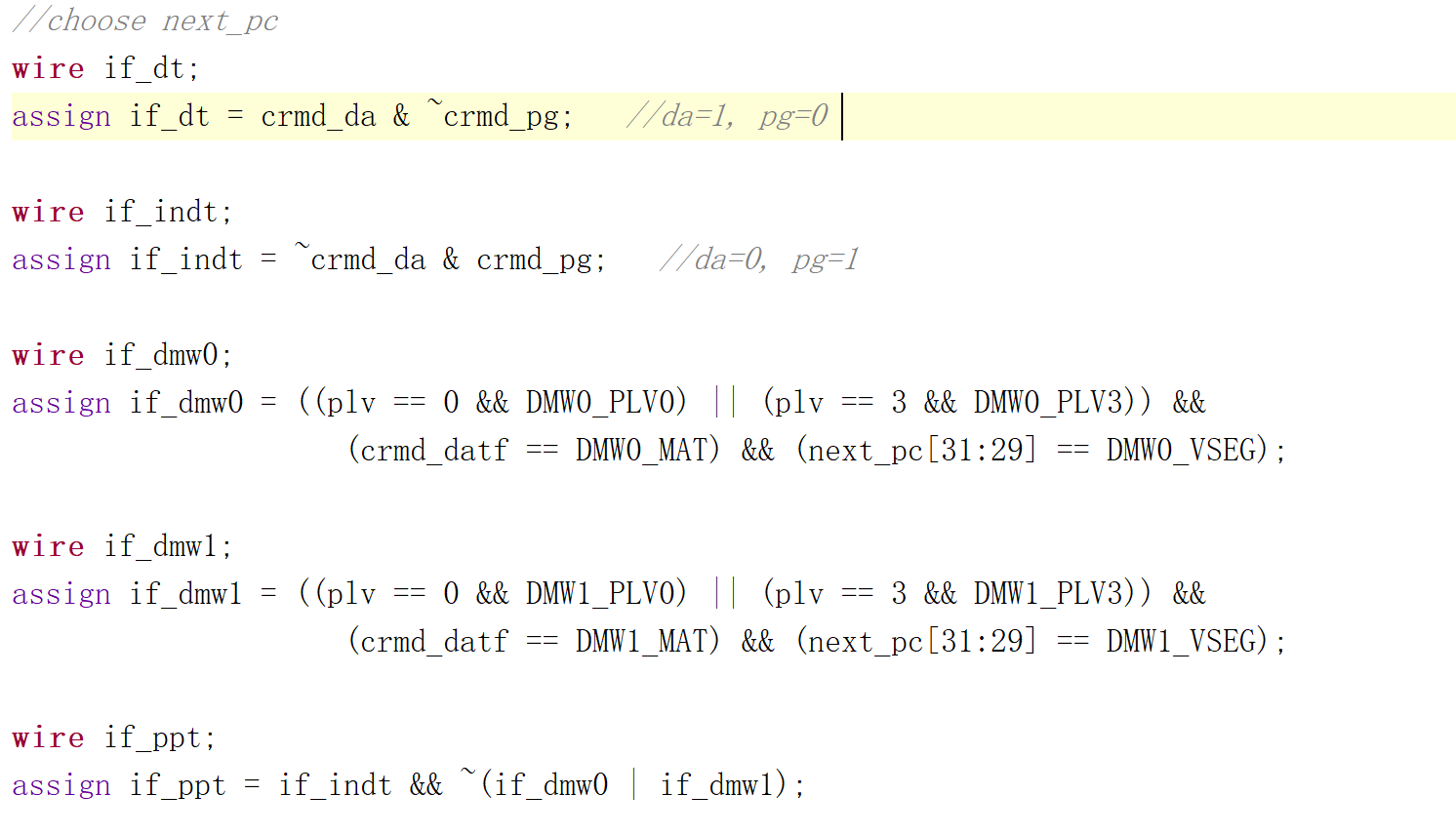
在EX阶段完成tlbsrch和invtlb指令与tlb模块的交互，在ID阶段译码之后，在EX模块向tlb发出tlbsrch和invtlb指令的信号。

在WB阶段完成tlbrd、tlbwr指令与tlb的交互，同时完成tlbrd、tlbwr、tlbsrch和csr的交互。

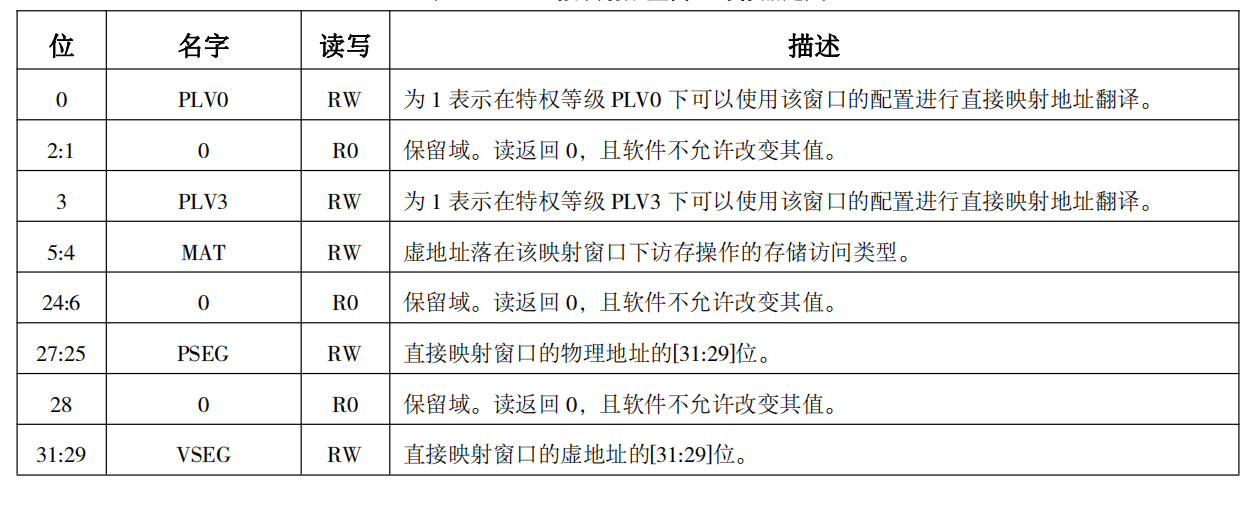


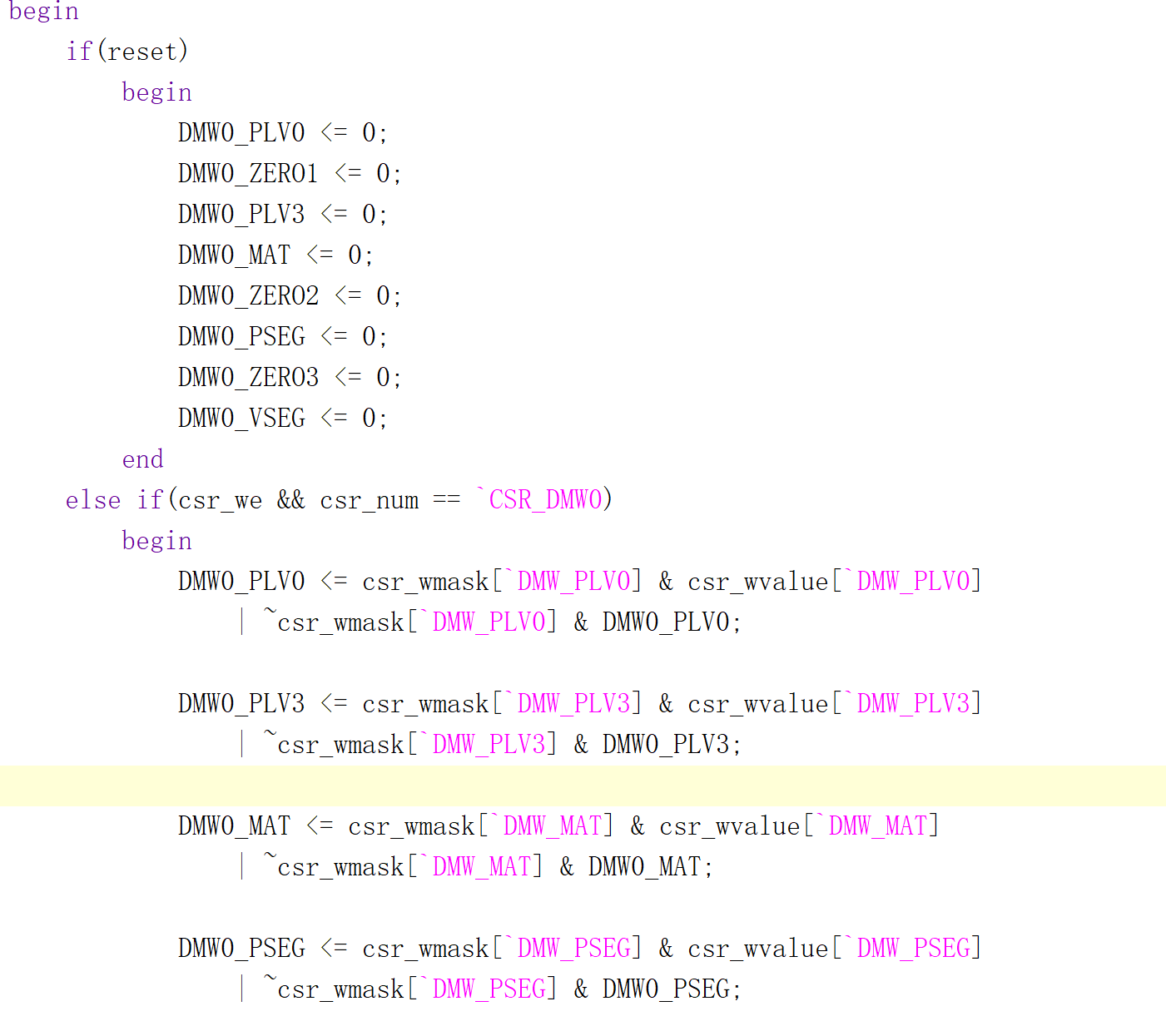
1. 虚实地址转换

分别在IF阶段（具体来说是pre-IF阶段）和EX阶段实现取指、访存的物理地址转换。两处地址转换逻辑基本一致，先判断是直接地址翻译还是映射地址翻译，在映射地址翻译模式时先判断直接映射窗口DMW0和DMW1是否命中，若未命中则查找TLB进行翻译。若仍无有效TLB表项命中，则会引发TLB重填异常。

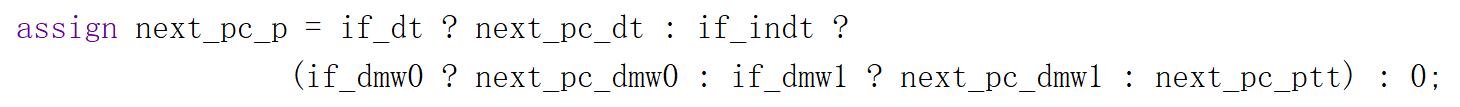
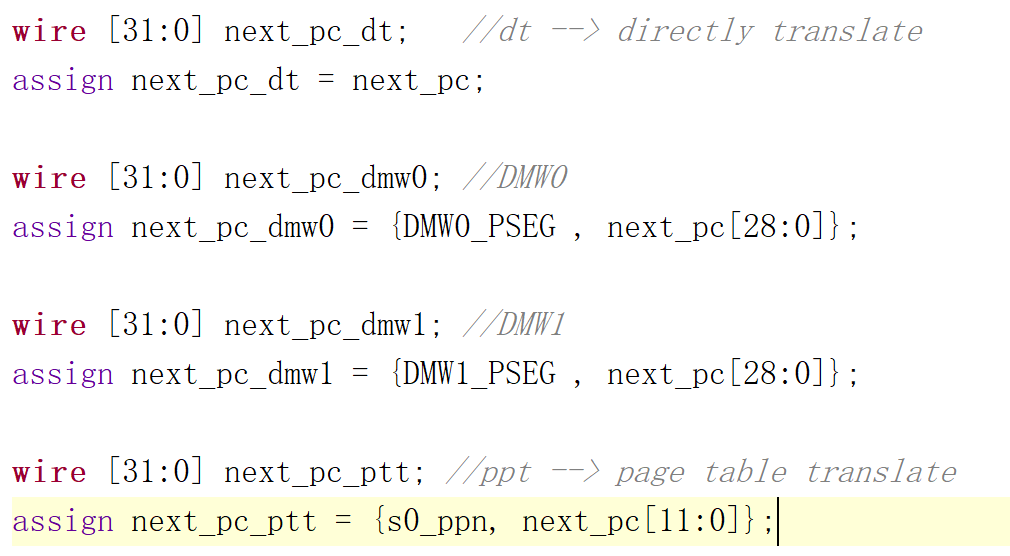


其中DMW0和DMW1两个CSR寄存器结构在参考手册上有详细说明，也没有特殊写入逻辑，以组合逻辑分别连接到IF阶段、EX阶段中进行相关模式判断以及物理地址翻译。





确认地址翻译模式后，利用组合逻辑直接选择对应模式翻译得到的物理地址即可。



访存物理地址翻译与取指类似，区别主要在于从不同端口读取页表相关数据。这一部分按照书上描述编写即可。

注意翻译得到的物理地址只用于传给内存查找对应指令或数据，cpu内部计算、不同流水级传递的仍然为虚拟地址。

1. MMU相关异常实现

新添加六种MMU相关异常，按照指令手册上叙述进行判定，然后把相应信号连接到例外相应相关信号，如csr\_ecode等。

MMU相关异常的判断主要发生在IF级和EX级。IF级需要判断TLB重填、页特权等级不合规、取指操作页无效的异常，EX级除了取指页无效外其他例外都需要判断。

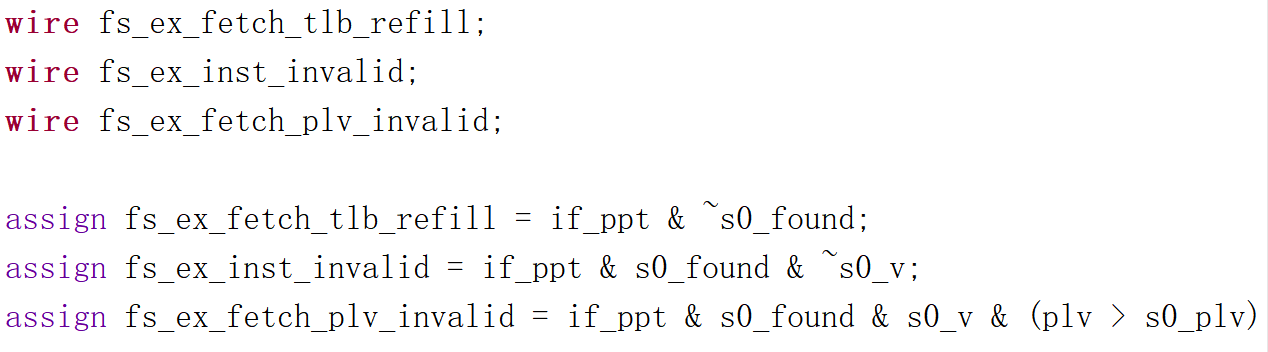


图 1 IF级例外判断

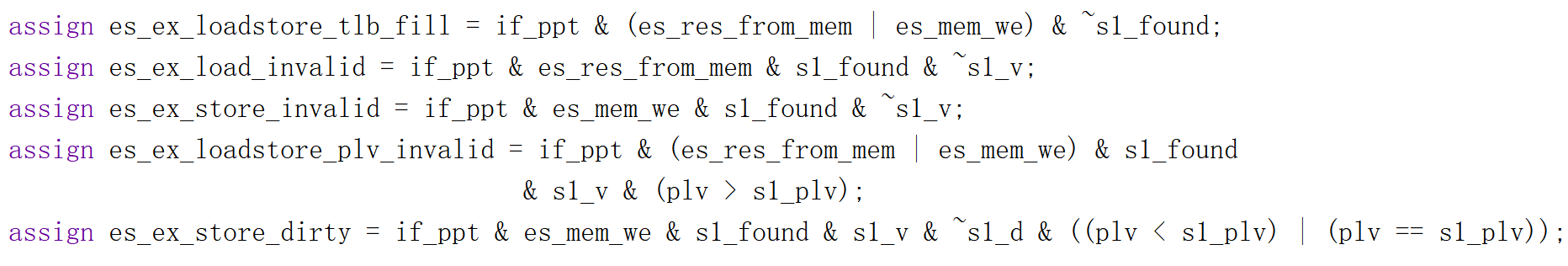
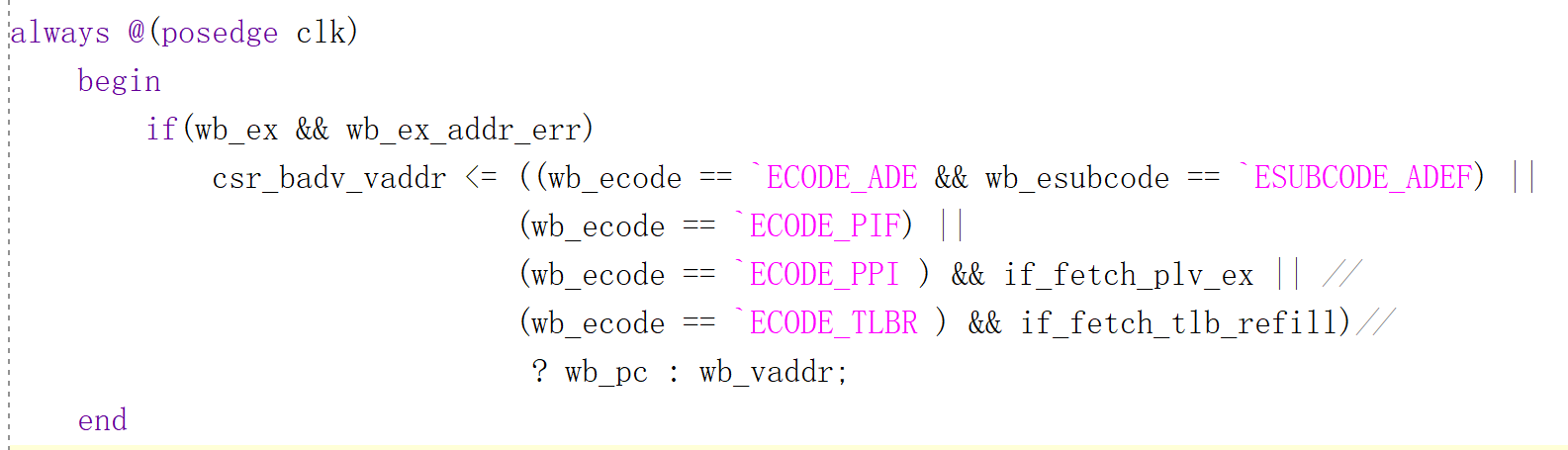


图 2 EX级例外判断

此外，MMU相关例外在相应时还需要额外保存信息到BADV和TLBEHI中，需要在csr中添加相关写入逻辑

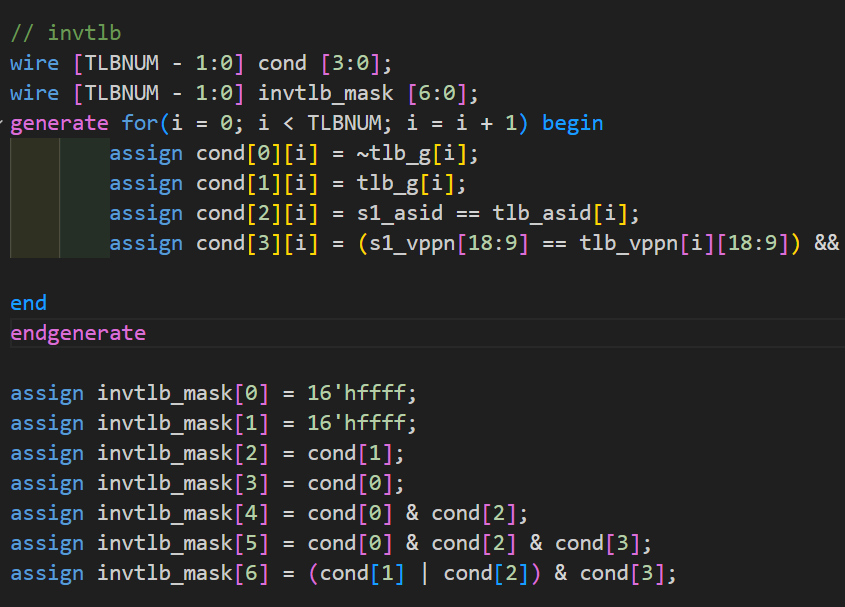


另外需要注意，TLB重填异常的入口是通过TLBRENTRY单独配置的，因而当发生TLB重填异常时，读取的csr\_num需要设置为TLBRENTRY对应序号



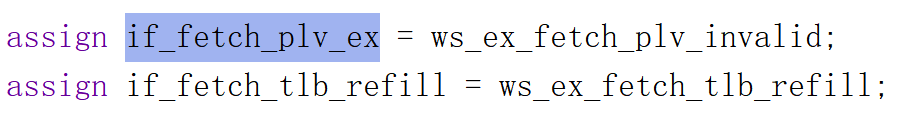
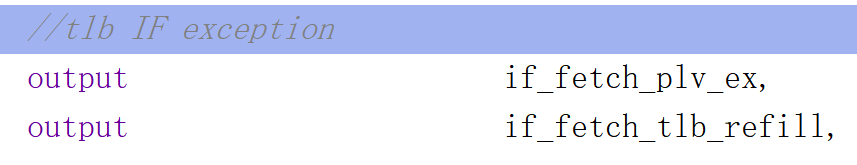
1. **实验过程中遇到的问题以及debug心得**

1.tlb模块中cond和mask向量的处理：



需要注意两个二维向量的命名方式：如果搞反了mask或cond向量前后的两个中括号的内容，可能会导致mask向量在赋值的时候出错，变成全X。

2.tlb异常处理中需要将引发异常的虚地址保存到CSR.BADV中，将虚地址的[31:12]位写入CSR.TLBEHI的VPPN域。需要注意这里所说的是引发异常的虚地址，而TLB重填、页特权等级不合规例外无法直接从例外类型判断到底是取指还是访存地址引发异常，故需要额外引入相关接口，方便这里进行判断。



1. **实验分工**

张钰堃负责完成exp17和exp18，刘景平负责完成exp19。