**实验13报告**

学号 2017K8009929032 2017K8009929034

姓名 杨程远 杨宇恒

箱子号 15

一、实验任务（10%）

CPU实例化上一个实验的tlb模块，增加 TLBR、TLBWI、TLBP 指令，以及MFC0和MTC0对Index、EntryHi、EntryLo0、EntryLo1 寄存器的支持。通过仿真上板验证。

二、实验设计（40%）

（一）总体设计思路

1. 增加CP0寄存器堆四个寄存器，提供额外的读写端口，MFC0和MTC0在写回级读写。
2. **MTC0如果目标EentryHi**，后续指令不产生效果，写回**完成后的一个周期**开始重新发下条指令取指请求，才产生效果。
3. TLBR和TLBWI后续指令不产生效果，他们写回**完成后的一个周期**开始重新发下条指令取指请求，才产生效果，这是因为：
   1. TLBR会更改EntryHi寄存器，产生相关。
   2. TLBWI会改变TLB条目，产生相关。

产生的额外效果：

* 1. 不需考虑TLBR后续指令的CP0（Index、EntryHi、EntryLo）的相关阻塞，即Index、EntryHi、EntryLo的相关阻塞只考虑。

1. TLBR和TLBWI在写回级读写CP0和TLB，这是因为：
   1. 由于MTC0在写回级可能写CP0（Index、EntryHi、EntryLo），为避免相关阻塞，TLBWI在写回级读CP0。
   2. 进而，TLBWI在写回级写TLB。
   3. 由于MTC0在写回级可能写CP0（Index），为避免相关阻塞，TLBR在写回级读CP0。
   4. 进而，TLBR在写回级读TLB；进而，在写回级写CP0。
2. TLBP在执行级读EntryHi、查找TLB，会被后续流水级Index写指令阻塞；写回级写Index，这是因为：
   1. 复用访存的查找端口，因此既然不是访存指令，就可以在原本访存指令要访存的时候使用查找端口。
   2. 前面指令写EntryHi数据相关。
3. 数据相关处理总结：（括号内标注上述五条中哪一条解决了这个相关）
   1. CP0写后读相关的处理：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 先执行写  后执行读 | MFC0  （不写） | MTC0  （全可写） | TLBP  （写Index） | TLBR  （写Hi、Lo） | TLBWI  （不写） |
| MFC0（全可读） |  | 写回读（1） | 清流水（3） | 清流水（3） |  |
| MTC0（不读） |  |  |  |  |  |
| TLBP（读Hi） |  | 阻塞（5） |  | 清流水（3） |  |
| TLBR（读Index） |  | 写回读（4） | 清流水（3） |  |  |
| TLBWI（全读） |  | 写回读（4） | 清流水（3） | 清流水（3） |  |
| 访存指令（读Hi） |  | **清流水（2）\*** |  | 清流水（3） |  |

表1 CP0写后读相关

\*这是讲义中没有提到，但我们在设计时通过列表发现的一个需要注意的地方

* 1. CP0读后写相关的处理：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 先执行读  后执行写 | MFC0  （全可读） | MTC0  （不读） | TLBP  （读Hi） | TLBR  （读Index） | TLBWI  （全读） | 访存指令  （读Hi） |
| MFC0（不写） |  |  |  |  |  |  |
| MTC0（全可写） | 写回写（1） |  | 写回写（1） | 写回写（1） | 写回写（1） | 写回写（1） |
| TLBP（写Index） | 写回写（4） |  | 写回写（4） | 写回写（4） | 写回写（4） | 写回写（4） |
| TLBR（写Hi、Lo） | 写回写（4） |  | 写回写（4） | 写回写（4） | 写回写（4） | 写回写（4） |
| TLBWI（不写） |  |  |  |  |  |  |

表2 CP0读后写相关

* 1. TLB写后读相关的处理：

|  |  |  |  |
| --- | --- | --- | --- |
| 先执行写  后执行读 | TLBP（不写） | TLBR（不写） | TLBWI（全写） |
| TLBP（全读） |  |  | 清流水（3）或 写回读（4） |
| TLBR（全读） |  |  | 清流水（3）或 写回读（4） |
| TLBWI（不读） |  |  |  |
| 访存指令（全读） |  |  | 清流水（3） |

表3 CP0写后读相关

* 1. TLB读后写相关的处理：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 先执行读  后执行写 | TLBP（全读） | TLBR（全读） | TLBWI（不读） | 访存指令（全读） |
| TLBP（不写） |  |  |  |  |
| TLBR（不写） |  |  |  |  |
| TLBWI（全写） | 写回写（4） | 写回写（4） |  | 写回写（4） |

表4 CP0读后写相关

* 1. 顺序执行，无写后写相关。

（二）重要模块修改1：CP0模块

1. 工作原理

增加Index、EntryHi、EntryLo0、EntryLo1 寄存器，及相关读写端口。

1. 增加的接口定义

| **名称** | **方向** | **位宽** | **功能描述** |
| --- | --- | --- | --- |
| tlbp\_entryhi | OUT | 32 | 为tlbp指令提供entryhi到执行级。 |
| tlbr\_wen | IN | 1 | tlbr指令对CP0寄存器写使能。 |
| tlb\_index | OUT | $clog2(TLBNUM) | tlbr和tlbwi输出给tlb模块的index |
| tlbr\_entry | IN | 78 | tlbr写向CP0寄存器的表项。 |
| tlbwi\_entry | OUT | 78 | rlbwi输出给tlb模块的表项。 |

1. 功能描述

tlb指令直接用模块接口进行读写。

三、实验过程（50%）

（一）实验流水账

2019.12.9 18:00-22:00 设计

2019.12.10 12:00-18:00 实现

（二）错误记录

1、错误1：分支跳转为X

（1）错误现象

分支判断的两个操作数有X。



（2）分析定位过程

回溯指令发现这是对mfc0 EntryHi是否返回正确值的判断。

（3）错误原因

检查代码，发现tlbr更新EntryHi寄存器的条件误将tlbr\_wen写成tlbr\_entry。

（4）修正效果

通过这个测试点。

（5）归纳总结

用tab补全的是否注意一下。