**计 算 机 体 系 结 构（研 讨 课）实 验 报 告**

实验项目 prj7 小组编号 28 组员姓名 刘景平、张钰堃、付博宇

1. **实验目标**

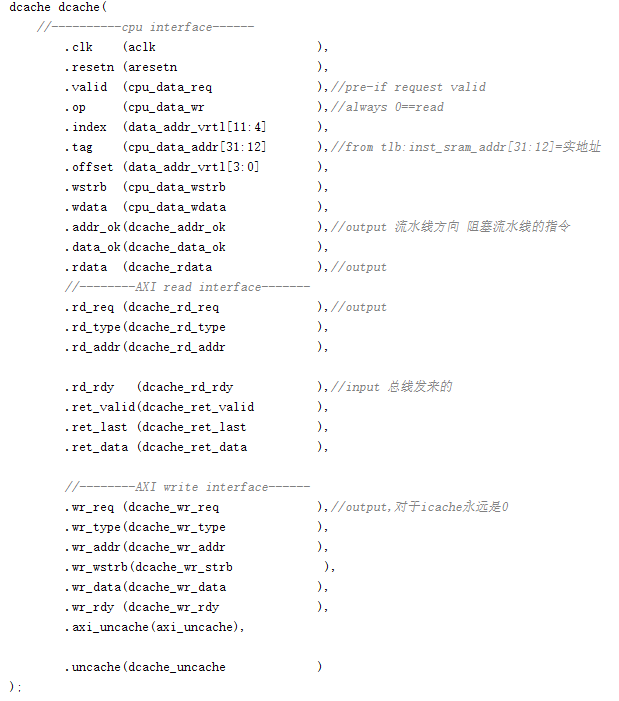
（一）exp22：

1. 将exp20完成的cache模块作为dcache集成到cpu中
2. 在采用AXI总线的Soc验证环境里完成对应的func功能验证，成功通过仿真及上板验证
3. **逻辑电路结构与仿真波形相关说明**

（一）例化dcache及axi总线调整

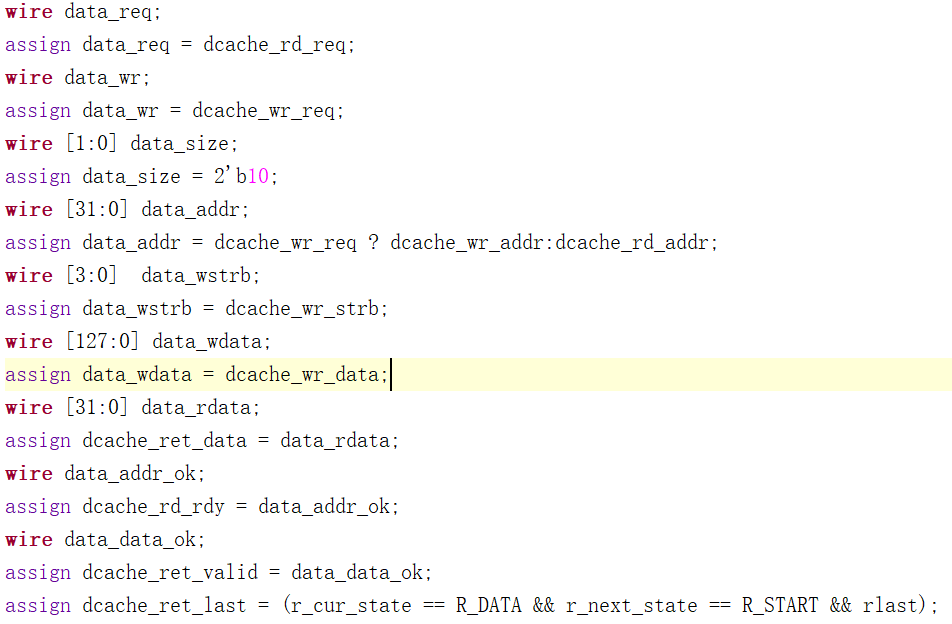
1.例化dcache

将cpu的数据请求通道与dcache相连，再将dcache的数据读取通道与axi桥相连。在这里需要将axi桥面向cpu的数据通道修改成面向dcache的数据接口。

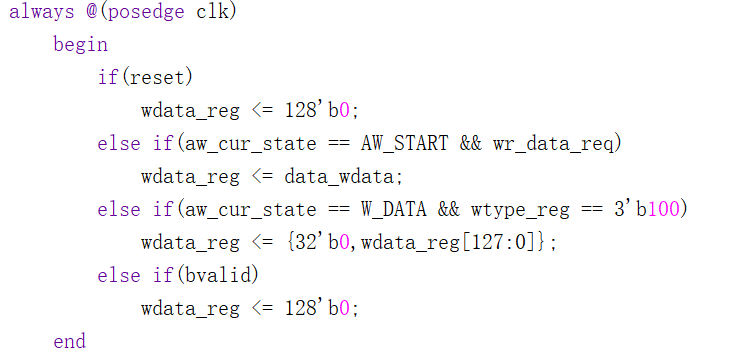
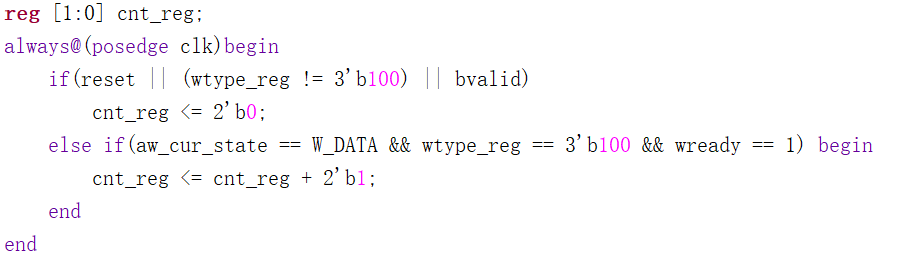


2.axi总线调整

将dcache输入端口与原面向cpu设计的端口相对应，保证尽量复用原本数据通路。



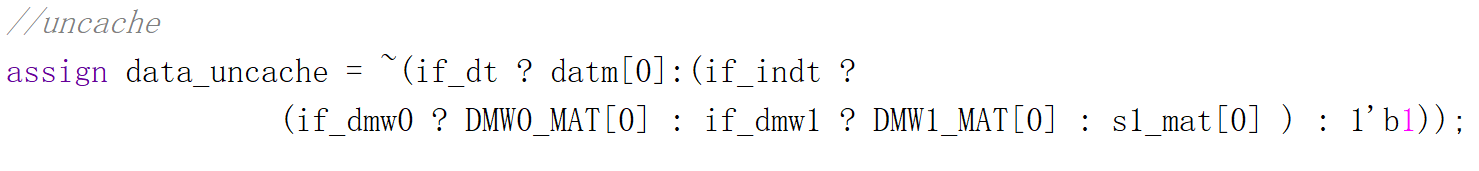
参考icache对axi总线的修改，实现cache行重填时的数据通路修改。修改写sram通路。由于从dcache传进来的写会数据为128位，而面向sram的写回端口为32位，需要考虑突发写。考虑设置寄存器cnt\_reg记录当前写回第几个32位，并将wdata\_reg不断左移，保证低32位为当前需要写回的数据。在cache状态下，突发写的last信号由cnt\_reg控制。



（二）uncache通路设计

1.uncache情形判断

是否为Uncache与当前地址翻译模式相关，故选择在cpu EX级判断dcache是否为uncache 状态。判断方式为查看当前状态对应的MAT域。

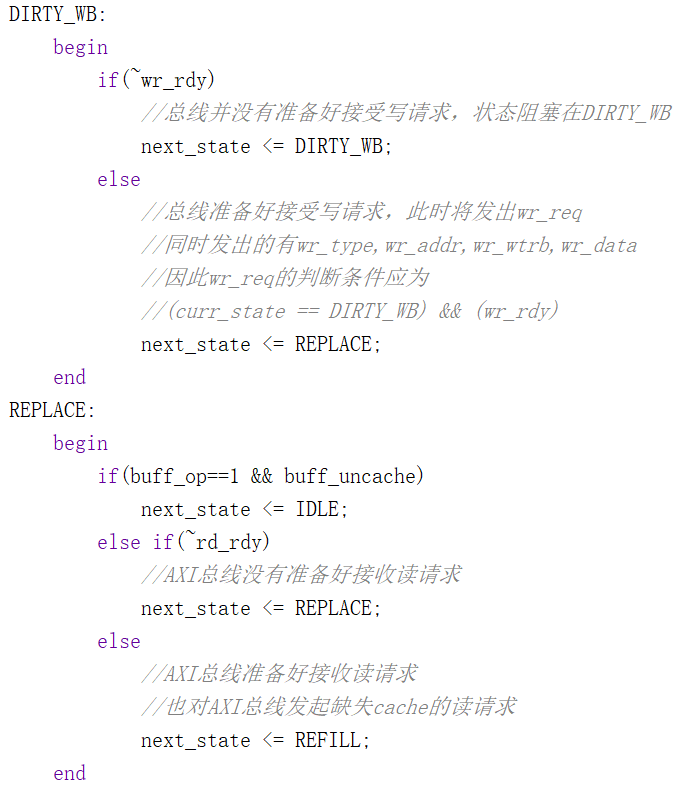


注意uncache信号可能在cache处理数据期间发生变化，需要用buff\_uncache进行保存，且大部分情况下利用buff\_uncache进行判断。

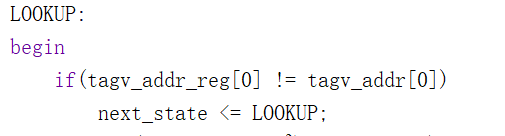
2.uncache数据通路

Uncache情形下读请求和写请求的数据通路不完全一致，这里分开讨论。

非缓存读复用不命中读通路，需要注意此时读到的数据非突发读，且只传回一个数据，需要更改对应rd\_type，同时读回的数据不能写回cache，即tag以及cache行的wea需要保持低电平，可以直接将axi总线读的数据返回给cpu;非缓存写复用脏块写回通路，这里同样需要注意wr\_type和wr\_wstrb与写回cache行不完全一致。此外非缓存写不需要REFILL阶段，可以写请求发出后可以直接回到IDLE阶段。



1. **实验过程中遇到的问题以及debug心得**
2. 由于查询tag为同步查询，可能出现读出来的tag还未准备好，LOOKUP阶段已经查看cache\_hit结果，一方面会导致重复重填cache行降低效率，另一方面可能导致覆盖脏块返回错误数据。在这里增加addr\_reg寄存器，当addr稳定是，说明tag已经准备好，否则仍保存在LOOKUP阶段。



1. **实验分工**

刘景平负责完成exp22；