# Cache 实验提示

## 数据通路

1. **由于加入了BRAM，大家最好能将流水线拆分为6级，也就是把IF拆分为IF1和IF2，在IF1发起对ICache的访问，Cache直接把指令送到IF2流水级，并在IF2阶段发起对流水线的阻塞**
2. 主存可以使用任意类型存储器来实现。可以在主存和Cache之间设置几个寄存器来模拟访存延迟
3. 给出的参考数据通路中，也可以使用Return Buffer中的数据，作为本次缺失访问的数据递给存储器，避免了填充存储器后再一次访问存储器
4. 大家实现出来可能会有存储器的写优先读优先问题。建议大家直接把存储器选成写优先模式
5. 可以参考《CPU设计实战》中状态机的设计，来处理数据访存问题。
6. 由于"主存"一行只有32位，所以我们需要在缺失时对主存发起四次访存。事实上，这一部分的实现方式可以是：**Cache只向外发起一次访存，但默认这次访存可以获得连续四个数据。在Cache外将这次访存请求转换为连续的四次访存**（可以用状态机或计数器），**从而一个字一个字地接受数据，并在Return Buffer中完成拼接。**（老师在课堂上讲可以使用模4意义的加法来获得这四个数据，这是为了简化设计，但真正的访存必须顺序访存。一个简单的方法是，**每次发起访存的地址都是16字节向下对齐即可**）
7. 思考以下几个问题，可以有助于大家理解参考通路中Cache的原理：
    - 在访存命中的时候，这个Cache对应的流水线是否要停顿？
    - 我们把IF段拆解成了两个流水级IF1和IF2，那么在哪个流水级发起访存，在哪个流水级送出数据？
    - 各个Buffer的we是如何生成的？return buffer的we是否可以作为它的移位信号呢？
    - 千万不要小看数据选择：如何在一行的数据或Return Buffer的数据中选择出你要的数据呢（提示：块内偏移在地址的哪部分呢？）

## 实验要求

1. 测试的排序程序总指令数必须大于Cache最多能容纳的指令数（如果不够的话可以在最前面加入若干条nop）
2. 通过仿真波形体现**强制缺失**（初始时Cache所有行全部无效导致的缺失）和**容量缺失**（Cache太小无法装载全部指令导致的替换）或**冲突缺失**（由于多个块映射到同一个Cache行导致的替换，和容量缺失的区别是容量缺失必须是Cache全满后导致的缺失，而冲突缺失不一定在Cache全满时发生），这三个缺失的具体定义也可以参考《计算机组成原理》第五章
3. TagV表和Data Memory必须使用BRAM，多观察数据通路，思考为什么读地址直接用PC，而写地址用Request Buffer
4. LRU的实现过于昂贵，**我们只如要实现RU算法，也就是保证上一次使用的路不被替换即可**（注意易错点：如果实现LRU算法，并不是每一路维护一个计数器即可，而是每一路的每一行都要维护一个计数器，替换时需要两路地址相同的行计数器来进行比较）

## 实验建议（以下内容完全不强制要求实现！）

1. 参与个人赛或在团体赛中准备主要负责Cache设计的同学：
    - 建议多体会**参考通路**中Cache的流水化访存思路（仔细观察哪里是第一级，哪里是第二级，注意：BRAM可以理解为地址寄存器+Distributed RAM）。

- 建议通过思考状态机的设计，设计能高效进行缺失处理和访存的ICache
- 如果时间充足，可以简单实现一个写直达的DCache。实现建议：用一个加法器在EX阶段单独算地址，在EX阶段就递给到DCache，然后在MEM阶段获得数据

2. 建议从本次实验开始，大家能够多记录自己遇到的Bug，在实验报告中体现出来（在比赛中，在总结汇报中记录出自己处理错误的过程，是有利于比赛得分的）

# 新数据通路参考