

# Lab 3&4

## Pipelined CPU with Cache

2024 10.12

# 实验目的

- 理解Cache中Tag, Valid, Dirty的作用和**Least Recently Used**的概念
- 理解**Cache Management Unit**的作用及与CPU结合的方法
- 理解CMU的各种State, 以及处理Hit, Miss等状况的方法
- 掌握验证Cache和CMU设计正确性的方法

# 背景知识-Cache

- Cache目的：快

## Cache Line

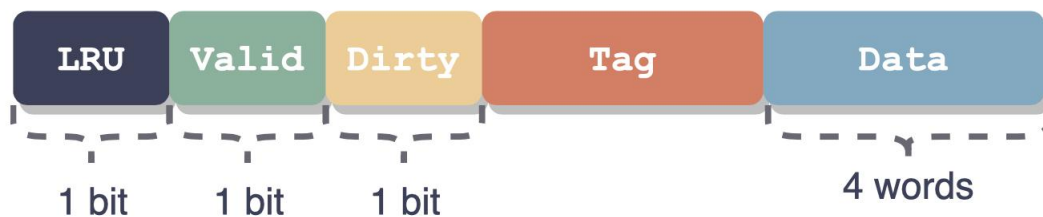
LRU： 在一个Set的N个数据块中，  
这一个数据块是否最近被使用

Valid： 数据是否有效

Dirty： 数据是否在读入后被CPU修改

Tag： 数据的地址

Data： 数据



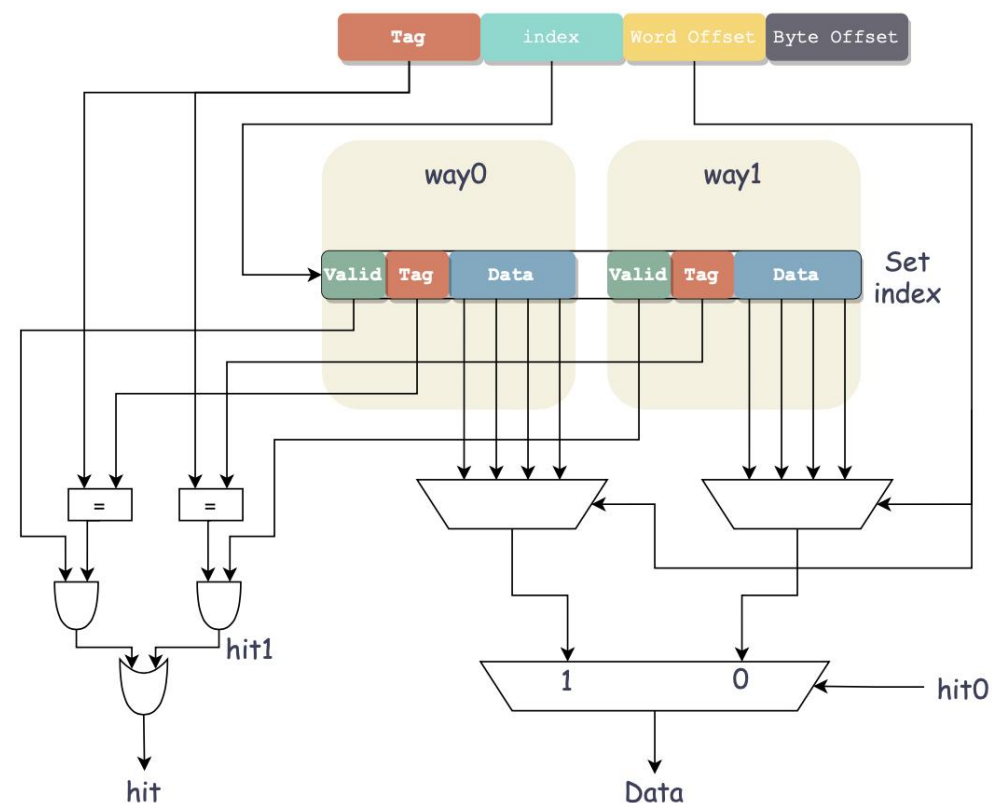
## 需要处理的情况分类

Read Hit 😊

Read Miss 😞 从内存读进来

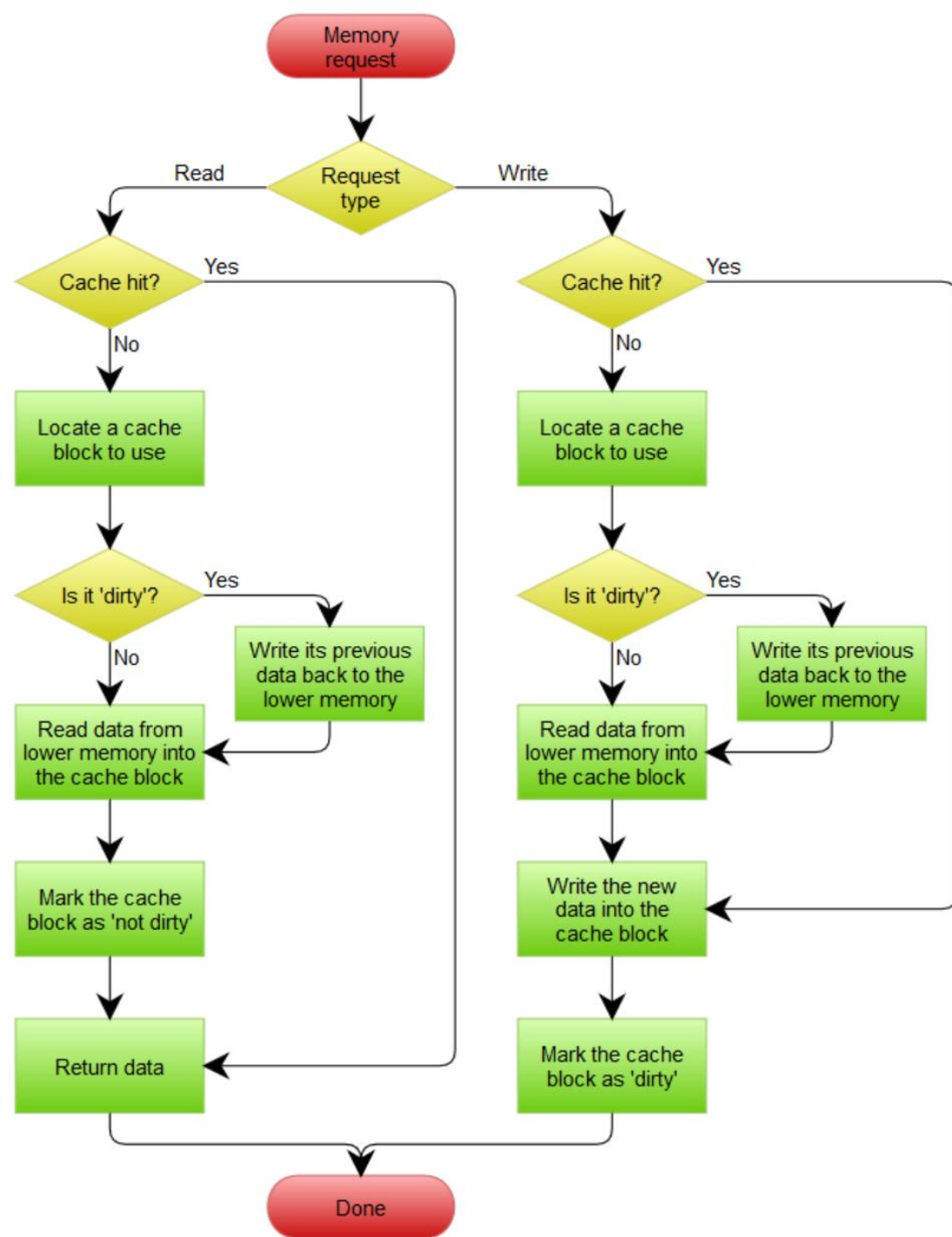
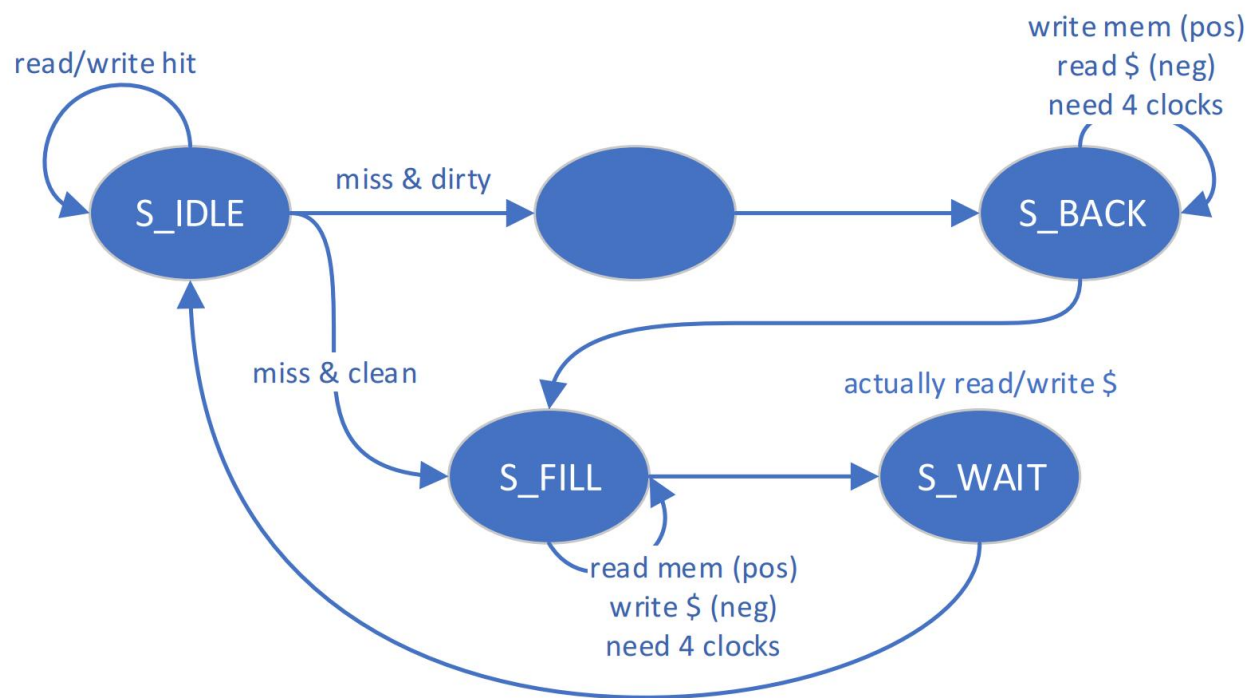
Write Hit 😊 不马上写回内存

Write Miss 😞 从内存读进来再写



# 背景知识

## Cache读写&CMU状态



# 实验目标

- Cache

64 cache lines

2-way set associative

4 words per cache line

Write Back

Write Allocate

LRU Replacement

Mem access 4 cycles	{ 不用写回的情况下miss	16 Cycles
	{ 写回的情况下miss	32 Cycles

# 实验得分点

- read miss: 15
- read hit: 15
- write miss: 15
- write hit: 15
- replace invalid: 10
- replace valid + dirty: 15
- replace valid + clean: 15
- **4-way set associative: +20 (bonus, 可溢出到平时成绩)**

# 实验步骤

- 参考Lab1和Lab2的实验指南，创建工程并导入框架源代码
- 实现Cache部分代码，并测试Cache的功能
- 实现CMU和RV32ICore部分的代码，并编写RISCV代码来测试CPU