

计算机组成原理

实验4 直接映射Cache设计



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

实验目的

- 掌握直接映射Cache的基本结构及其设计方法
- 实现直接映射Cache的地址映射机制
- 实现**读命中**、**读缺失**和**写命中**3种情况的处理



实验内容

为主存容量8KB的存储系统，设计直接映射Cache，参数如下：

- Cache总容量为256B、块大小4B （64个Cache块）

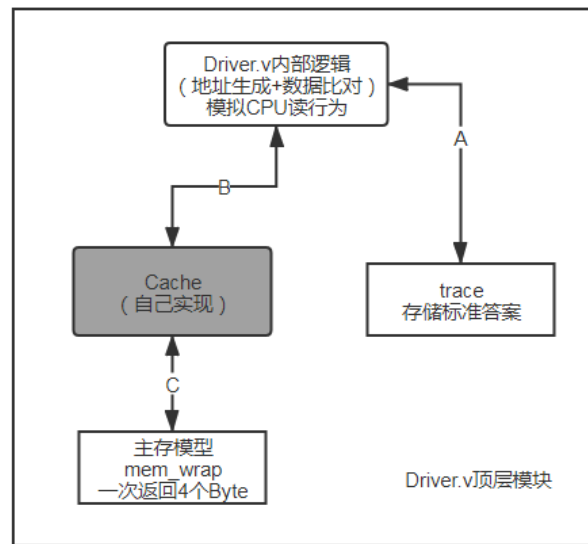
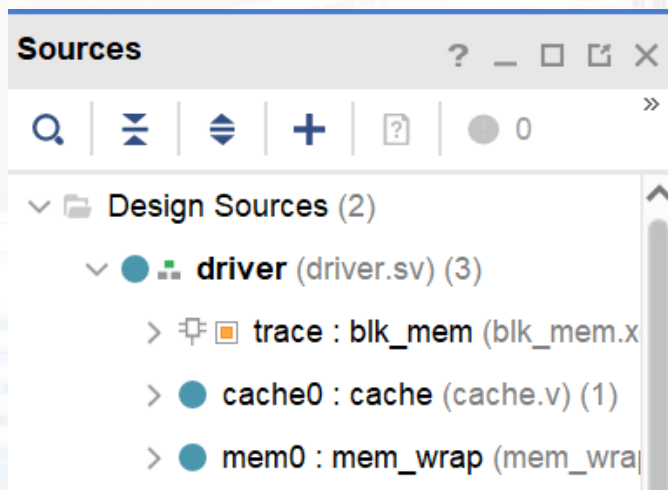
实现：

1. 在CPU发出读请求后的取数据操作，包括**读命中**、**读缺失**
2. 在CPU发出写请求后的写数据操作，仅需实现**写命中**，写缺失不处理



Vivado项目框架

- driver(driver.sv) --- 顶层模块，模拟CPU的访存行为，并进行数据正确性检查
- trace(IP核) --- 存储标准答案
- cache(**cache.v**) --- Cache模块 (**需要完成**)
- mem_wrap(mem_wrap.v) --- 主存存储器模型



Cache.v

模块接口

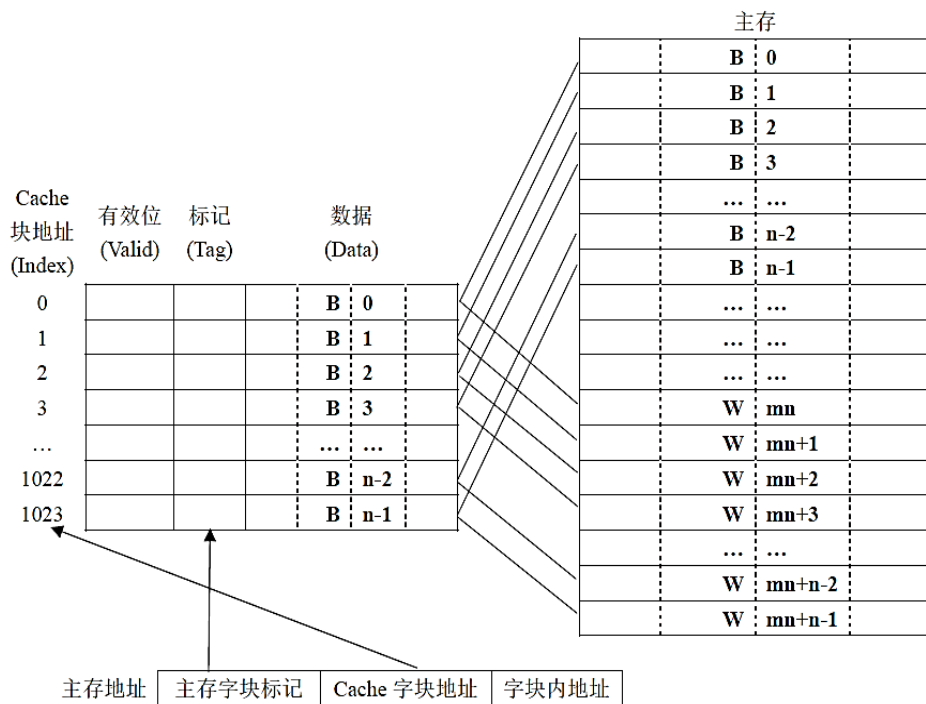
属性	名称	含义	位宽
全局信号			
输入	clk	时钟	1
输入	reset	复位 (高电平有效)	1
与上层Driver模块的接口			
输入	addr_from_cpu	CPU的读/写地址	13
输入	rreq_from_cpu	CPU的读请求	1
输入	wreq_from_cpu	CPU的写请求	1
输入	wdata_from_cpu	CPU的写数据	8
输出	rdata_to_cpu	Cache读出的数据	8
输出	hit_to_cpu	命中标记	1
与主存(mem_wrap)模块的接口			
输入	rdata_from_mem	主存模块读取的连续4字节	32
输入	rvalid_from_mem	主存读取完毕标记	1
输出	rreq_to_mem	读主存请求	1
输出	raddr_to_mem	读主存首地址	13
输出	wreq_to_mem	写主存请求	1
输出	waddr_to_mem	写主存地址	13
输出	wdata_to_mem	写主存数据	8

实验步骤

1. 进行地址各个字段的分解
2. 计算Cache各个参数，选择大小合理的Block RAM，创建IP核并实例化
3. 编写命中判断的组合逻辑
4. 设计控制状态机
5. 使用提供的testbench.v测试模块进行测试、调试，根据输出的调试信息，定位错误点



实验步骤-地址分解



Cache的总容量为256B

每块4个字节 ? 位

64个Cache块 ? 位

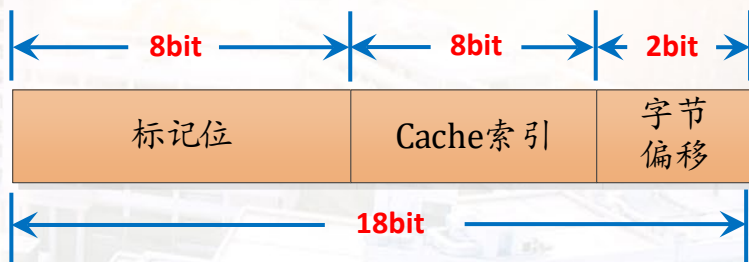
// CPU的读地址

input [12:0] raddr_from_cpu,

实验步骤-地址分解

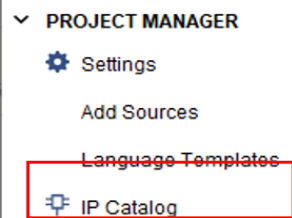
• 举例

- 系统的主存容量为256KB (主存地址宽度为18位)
- 设计直接相联Cache
 - 大小为1024B, 块大小为4B
- Cache块大小是4B (块内偏移为2位)
- Cache块有256 ($1024\text{B} / 4\text{B} = 256$)
 - Cache块地址为8位
 - 剩余的 $18 - 8 - 2 = 8$ 位, 作为Tag使用

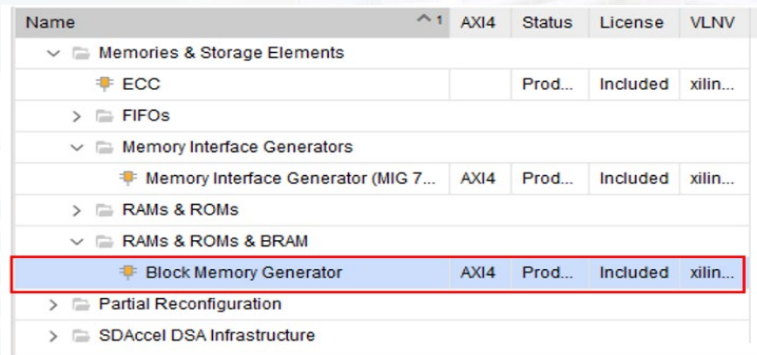


实验步骤-创建IP核

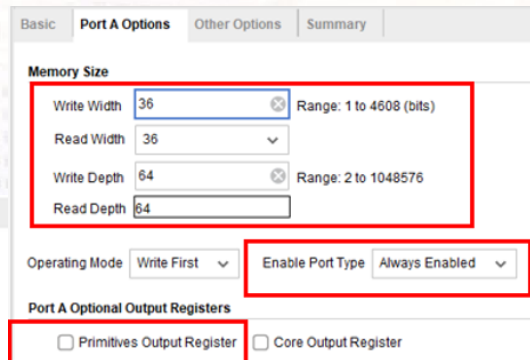
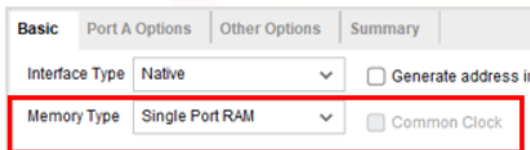
1) 点击Flow Navigator中的IP Catalog, 打开窗口添加



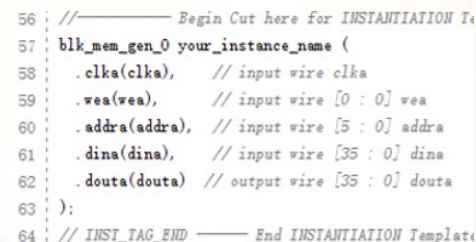
2) 双击Block Memory Generator



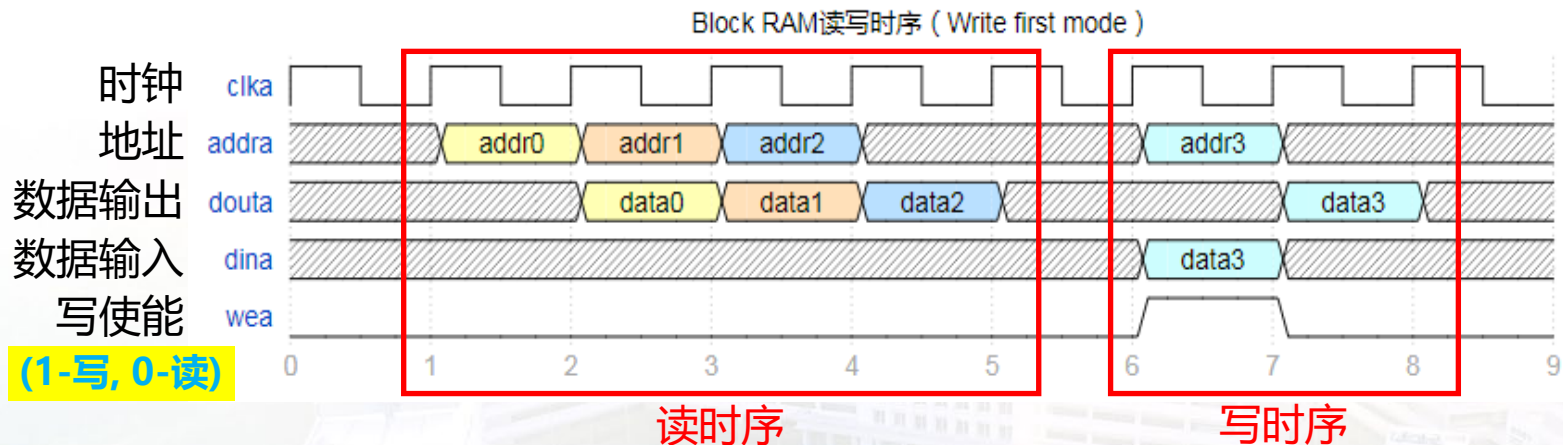
3) 配置RAM



4) 例化IP



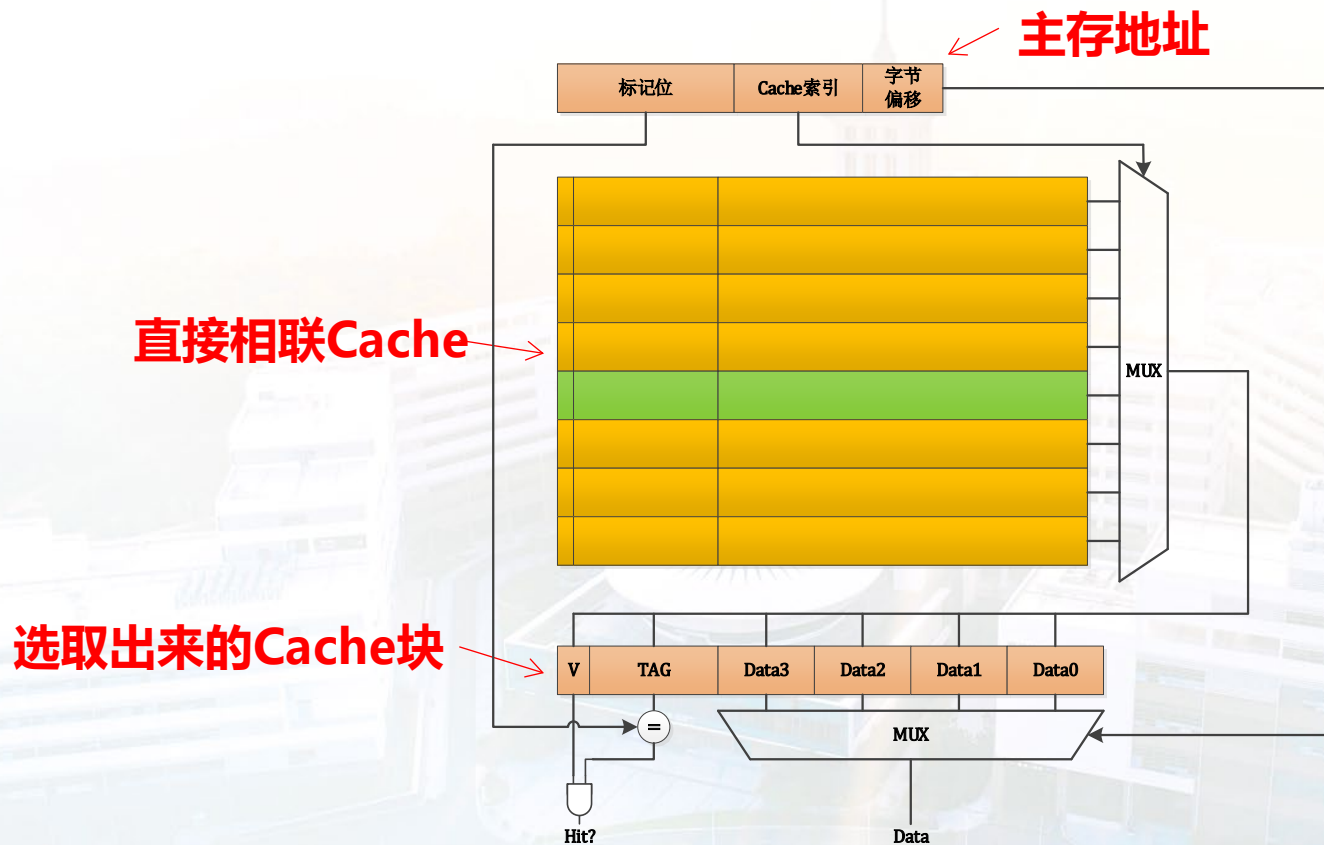
实验步骤-Block RAM IP核时序



- 读时序：上一周期给出地址，下一周期输出数据，可连续读取
- 写时序：上一周期给数据、写地址，拉高wea信号，下一周期成功写入数据，刚刚写入的数据出现在douta口上



实验步骤-判断命中机制



实验步骤-实现状态机

Mealy状态机实现

- **READY**: Cache处在就绪状态
- **TAG_CHECK**: 检查Cache是否命中
- **REFILL**: 将取回的字块, 加上相应的标签, 存进自己的存储体内。



实验步骤-Cache的读时序

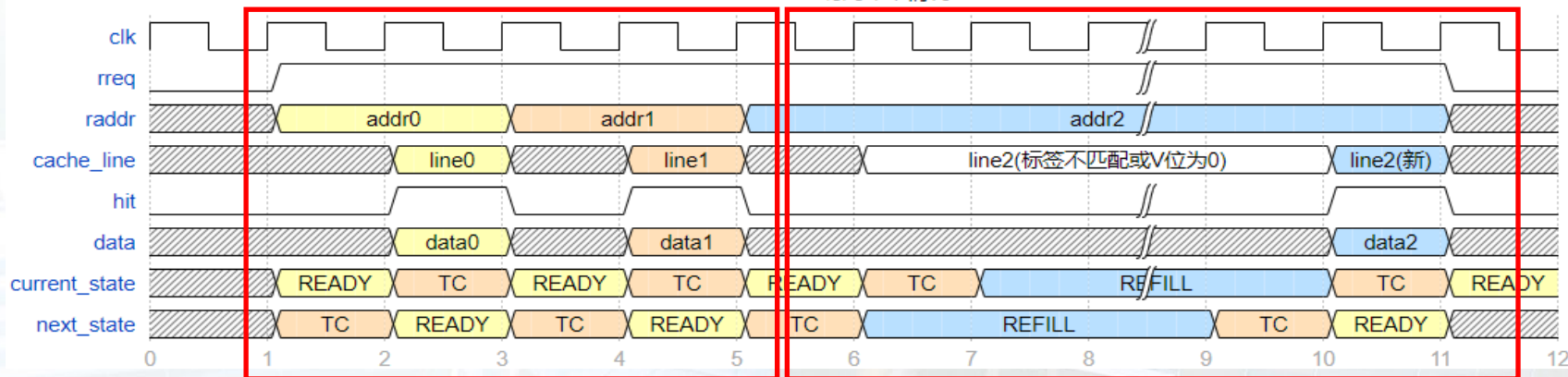
- 读命中(hit)
- 读缺失(miss)



hit

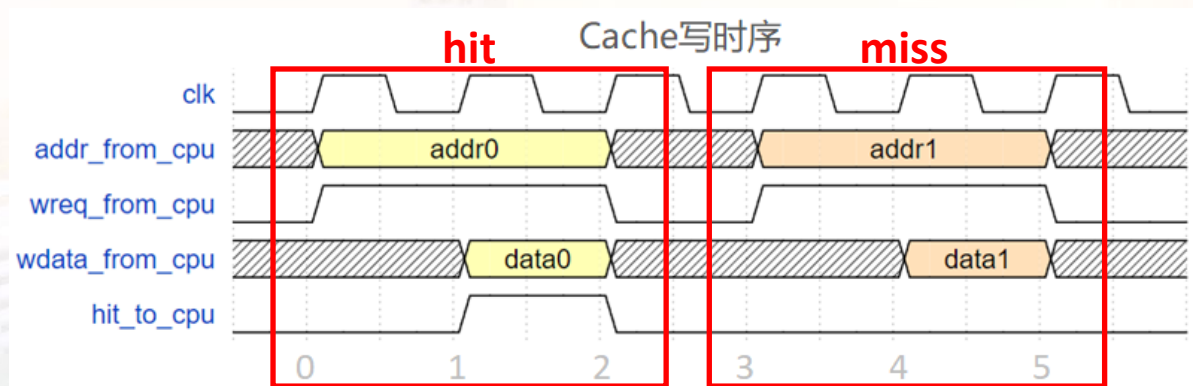
Cache访问命中情况

miss



实验步骤-Cache的写时序

- 写命中(hit)
- 写缺失(miss)



注意:

- ① 写请求只有效2个周期
- ② 写缺失时, hit需保持低电平, 且不能修改内存



实验步骤-测试

完成设计后，可以运行testbench.v文件中的仿真，运行仿真时，下方的Tcl Console也会打印相应的调试信息，帮助定位出错点

```
Tcl Console x Messages Log
[Icons]
Cache已取回数据162
该地址测试正确，将测试下一个地址
-----
访问地址为 507
等待Cache响应，应得到数据 0
Cache访问命中！
Cache已取回数据 0
该地址测试正确，将测试下一个地址
-----
访问地址为 508
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
等待Cache响应，应得到数据 8
```

```
if (rd_test_success & !f0) begin
    $fwrite(fp_w, "CACHE-READING TEST PASSED\n");
    $fwrite(fp_w, "Cycles spent on reading cache: %d\n", count);
    $display("=====读Cache测试全部通过=====");
    f0 <= 1;
end
if (test_success) begin
    $fwrite(fp_w, "CACHE-WRITING-HIT TEST PASSED\n");
    $fclose(fp_w);
    $display("=====Cache写测试全部通过=====");
    $display("=====所有测试全部通过！=====");
    $stop;
end
```


实验提交

- **提交内容**

- Cache模块源文件cache.v：3分
- 实验报告：4分

- 将上述文件打包成.zip，以 **“学号_姓名.zip”** 命名提交到作业系统

- ◆ 注意：**如有雷同，双方均0分！**





HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ