

# 计算机组成原理

## 实验 3 直接相连 Cache 设计

薛睿



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

- 掌握直接相联 Cache 的基本结构及其设计方法。
- 实现直接相连地址映射的机制
- 实现读命中和读缺失两种情况的处理。



本实验为主存容量 8KB 的存储系统，设计直接相联 Cache，参数如下：

- Cache 的总容量为 512B
- 每块 4 个字节
- 包括 128 个 Cache 块

实现在 CPU 发出读操作请求后的取数据操作，  
包括**读命中**和**读缺失**两种情况。

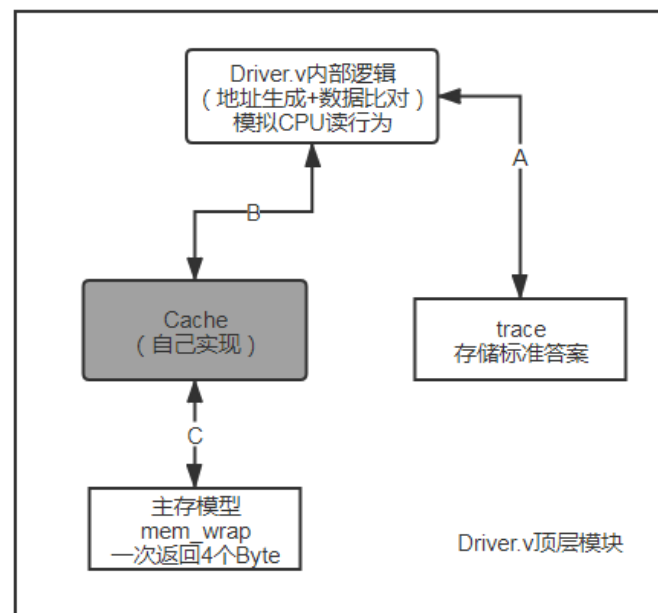
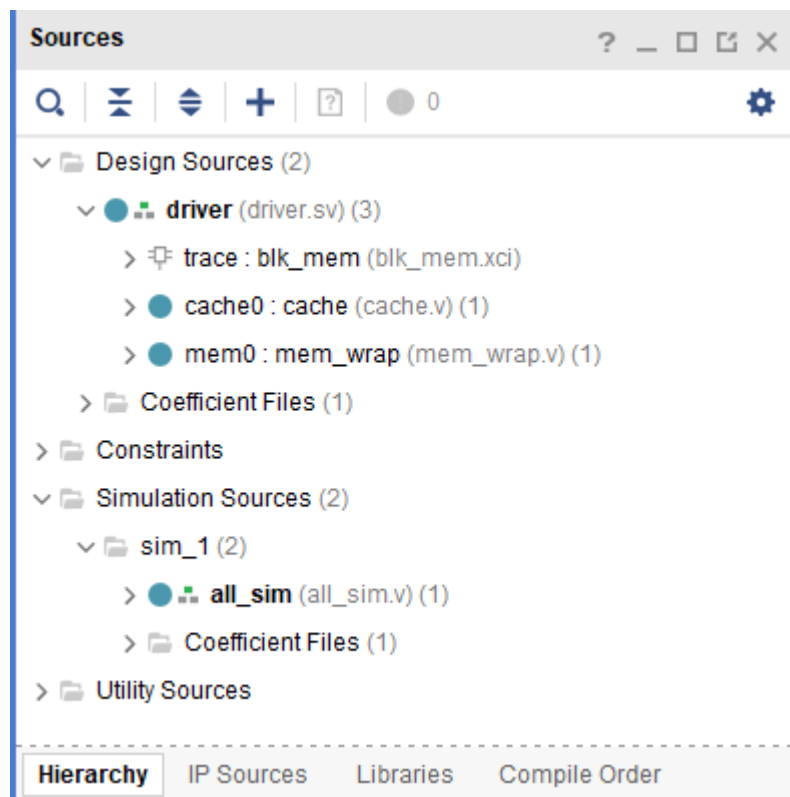
作业提交：

实验作业（3 分）

cache.v 文件和创建的 IP 核文件（\*.xci）（4 分）



- 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，可以看作是 CPU
- 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



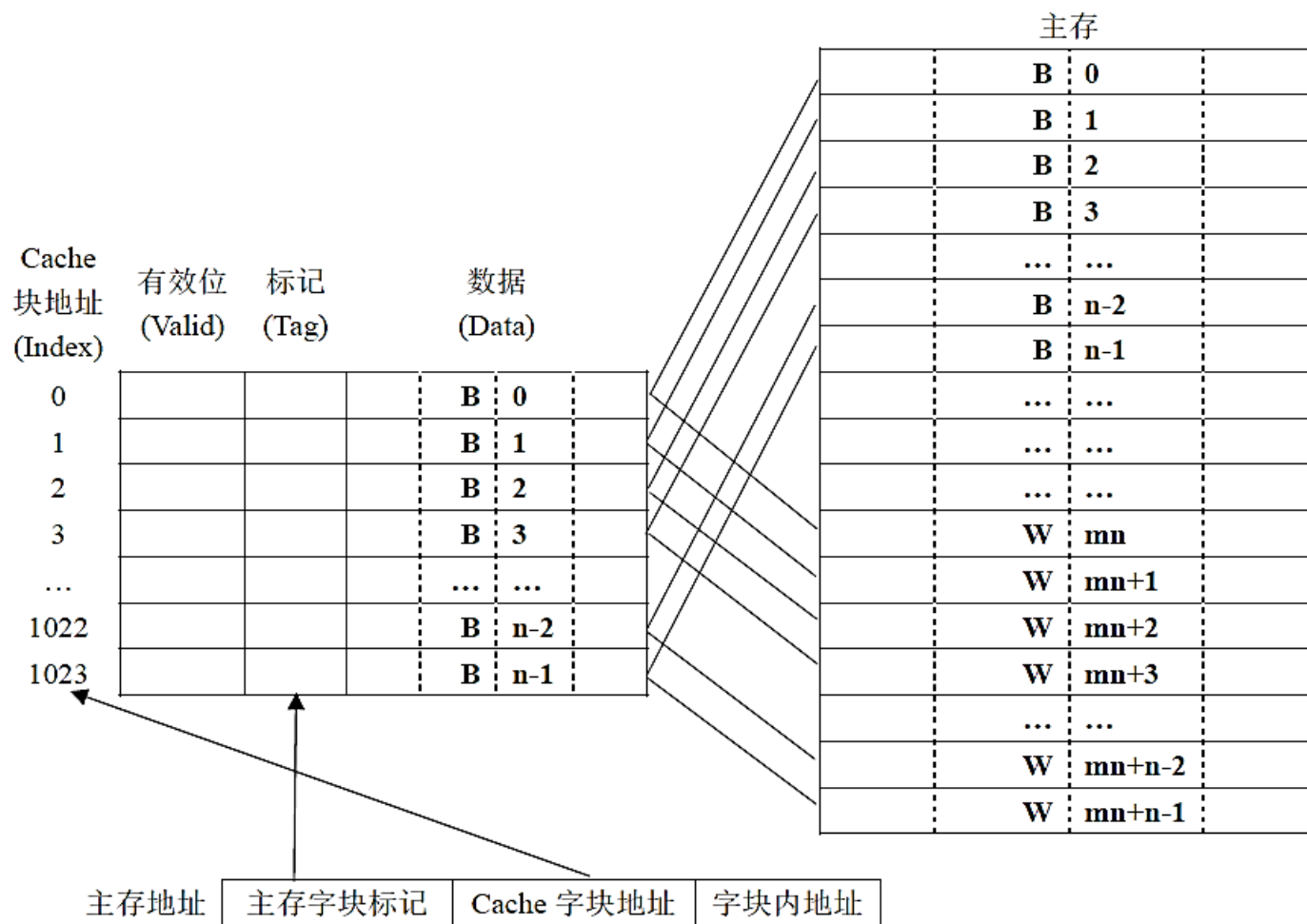
## • 接口规范

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	raddr_from_cpu	CPU 的读地址	13
输入	rreq_from_cpu	CPU 的读请求	1
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

- 进行地址各个字段的分解
- 计算 Cache 各个参数，选择大小合理的 Block RAM，创建 IP 核并实例化
- 编写命中判断的组合逻辑
- 设计控制状态机
- 自己编写 Testbench，验证相关功能正确性。
- 使用提供的 all\_sim.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。



## • 地址分解



直接映射:

(块地址) mod (cache 中的数据块数量)

有效位 (Valid) :

该行内存存储的数据是否有效, 未装入任何数据之前无效。

标记 (Tag) :

用以唯一地标识某个地址, 判断是否访问命中。

数据域 (Data) :

连续存储的 k 个相邻的在主存中连续的数据。

Cache 的总容量为 512B

每块 4 个字节

? 位

128 个 Cache 块

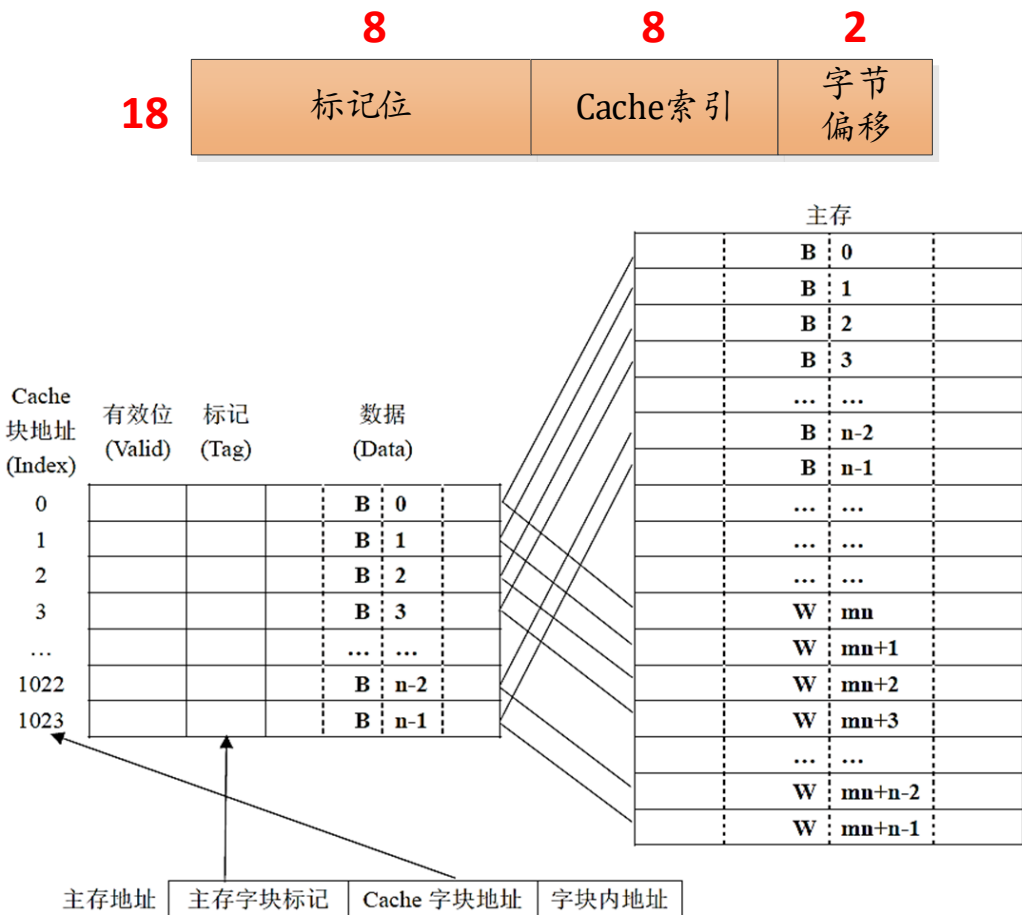
? 位

input [12:0] raddr\_from\_cpu, // CPU 来的读地址



## • 举例

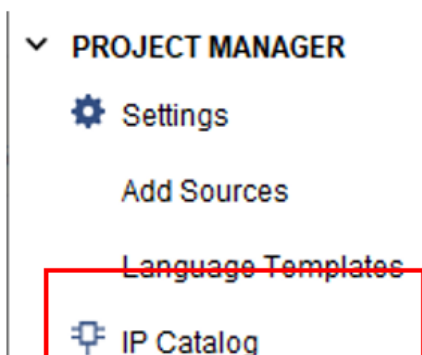
- 系统的主存容量为 256KB，则主存地址宽度为 18 位；
- 设计 Cache 每个字块中存储 4 个字节，则每个块中需要用 2 位去寻址相应的字节；
- Cache 容量为 256 行（ $256 \times 4B = 1024B$ ），且设计的方式是直接相联，则 Cache 字块地址为 8 位；剩余的  $18 - 8 - 2 = 8$  位，则作为 Tag 使用。



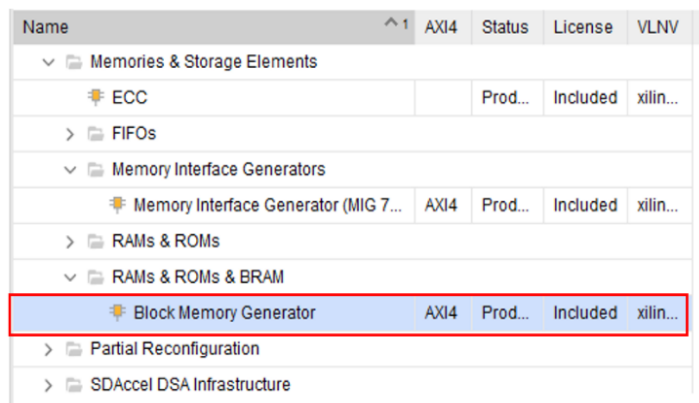


## • 创建 IP 核

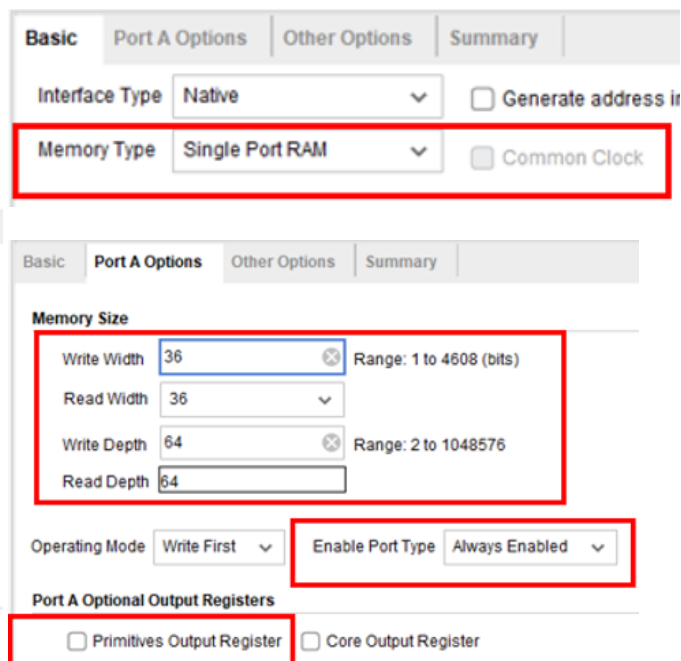
1) 点击 Flow Navigator 中的 IP Catalog，打开窗口添加 IP 核。



2) 双击 Block Memory Generator。



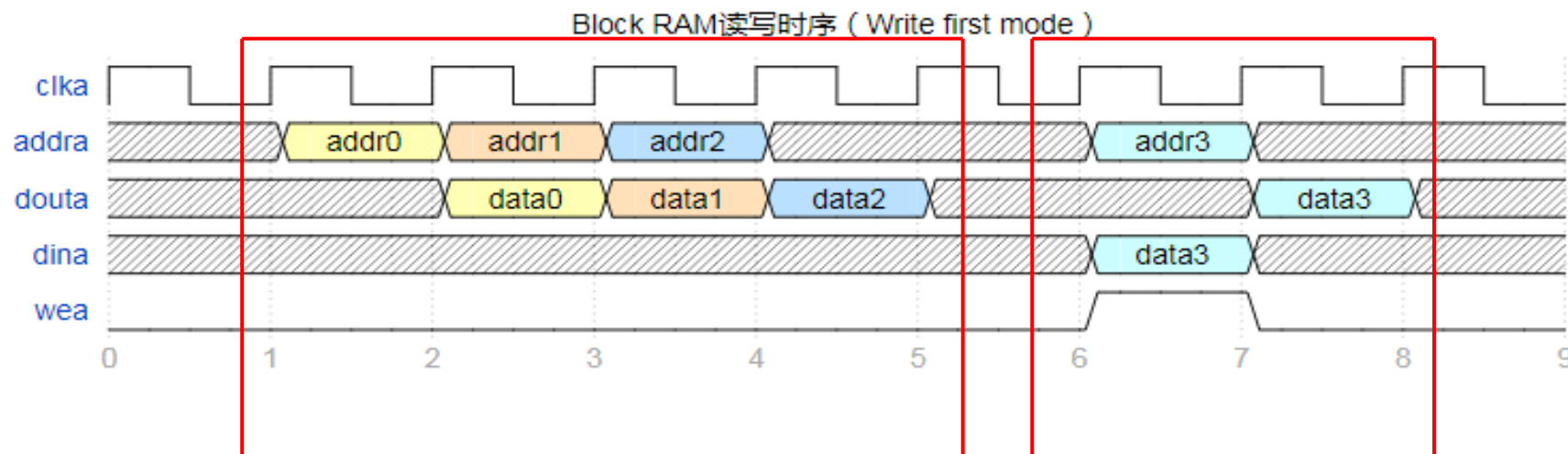
3) 配置 RAM



4) 例化 IP

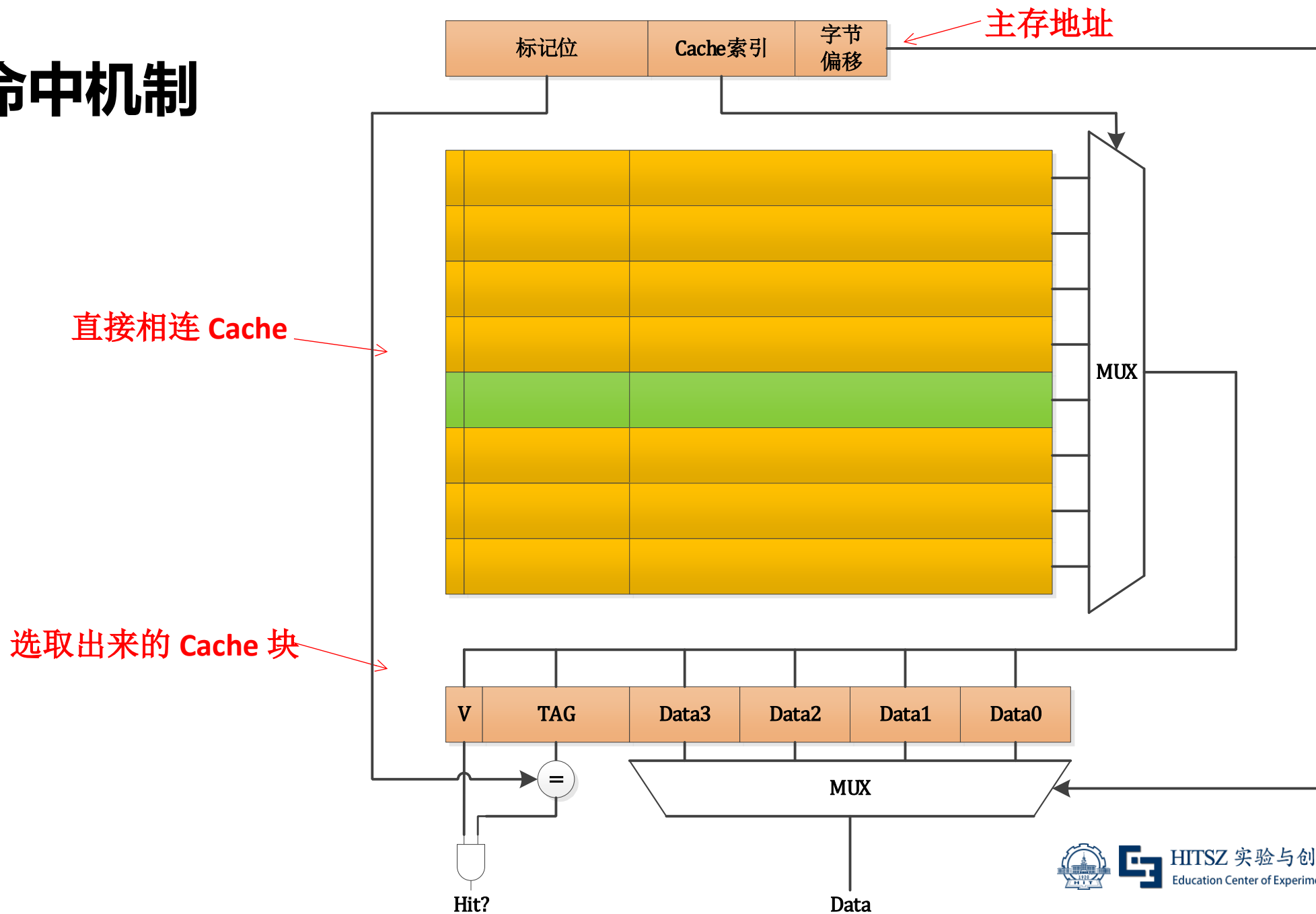
```
56 //----- Begin Cut here for INSTANTIATION Template
57 blk_mem_gen_0 your_instance_name (
58     .clka(clka),      // input wire clka
59     .wea(wea),        // input wire [0 : 0] wea
60     .addra(addra),    // input wire [5 : 0] addra
61     .dina(dina),      // input wire [35 : 0] dina
62     .douta(douta)     // output wire [35 : 0] douta
63 );
64 // INST_TAG_END ----- End INSTANTIATION Template
```

## • Block RAM 的读写时序



- 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。

## • 判断命中机制



## • 实现状态机

### Mealy 状态机实现

#### • READY :

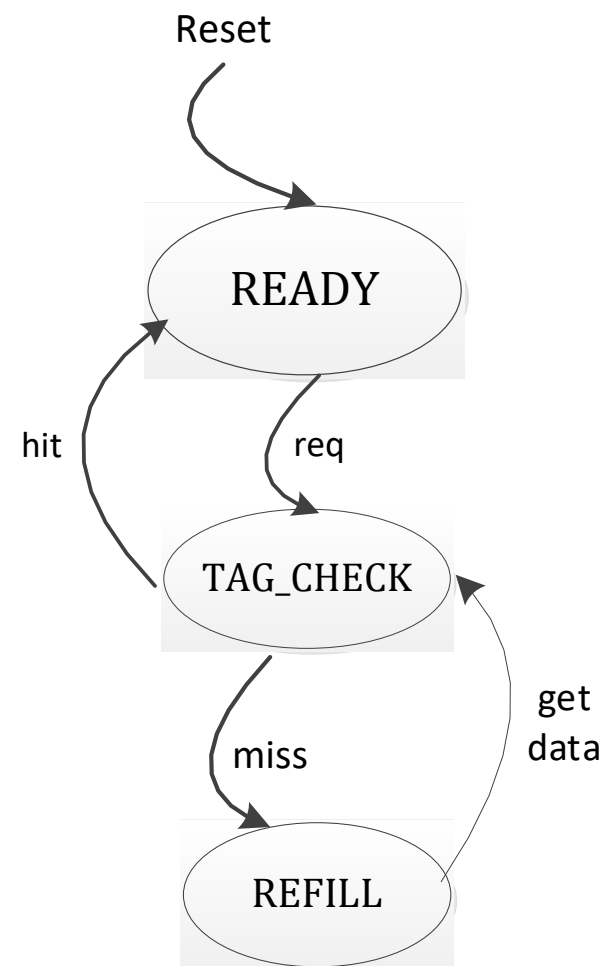
- Cache 处在就绪状态

#### • TAG\_CHECK :

- 检查 Cache 是否命中

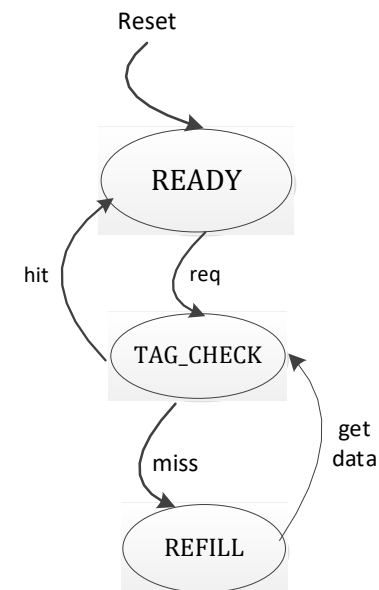
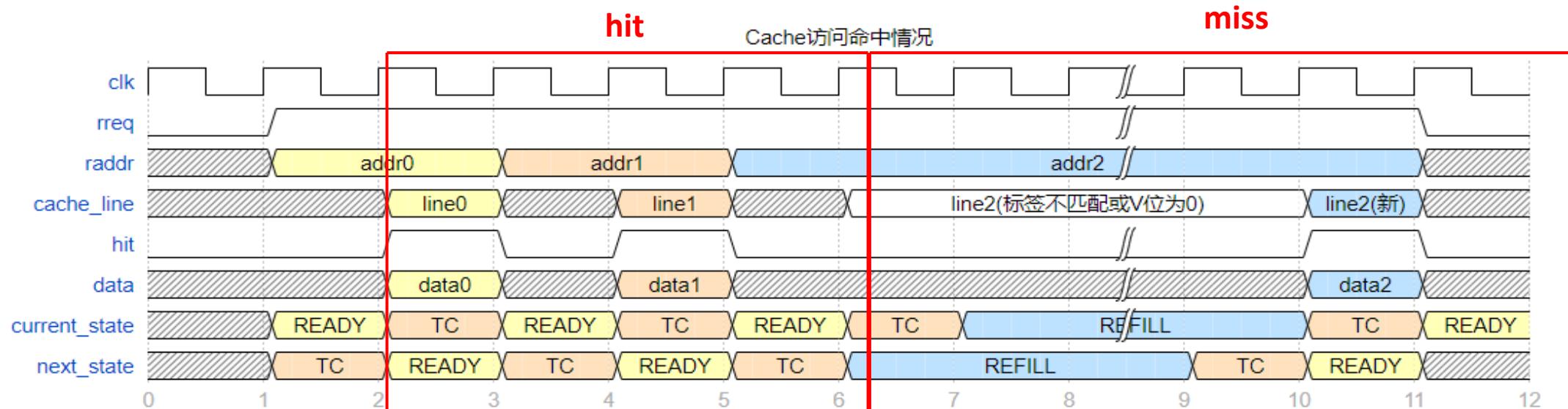
#### • REFILL :

- 将取回的字块，加上相应的标签，存进自己的存储体内。



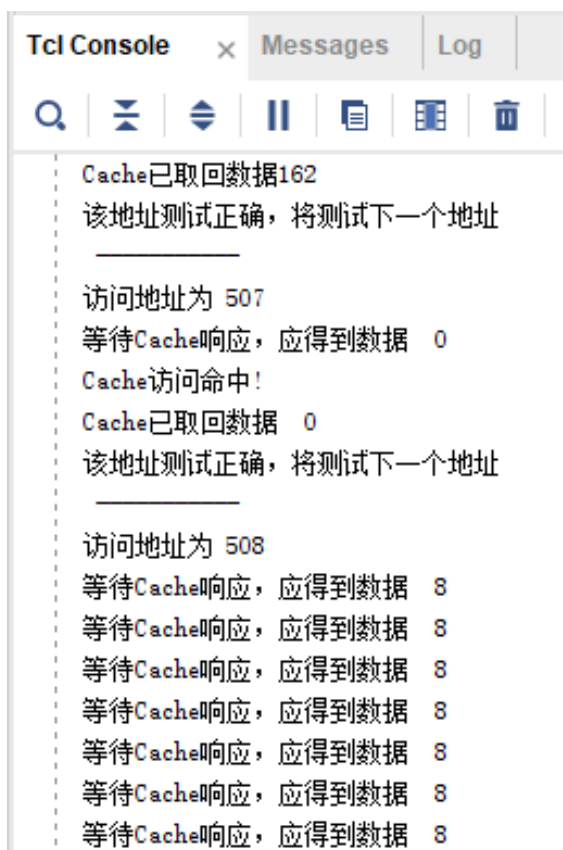
## • Cache 的读时序

- 读命中 (hit)
- 读缺失 (miss)



## • 测试

完成设计后，可以运行 all\_sim.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，帮助你定位出错点。



The screenshot shows a 'Tcl Console' window with a toolbar containing icons for search, zoom, list, pause, copy, paste, and delete. The console output is as follows:

```
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
```



- 提交网址:

[http://10.249.12.98:6620/auth/users/sign\\_in](http://10.249.12.98:6620/auth/users/sign_in)

- 课上完成作业的第 1 部分及第 2 部分的状态转移图
- 提交时间: 6 月 14 日 (星期一) 24 : 00



# 开始实验



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