



哈爾濱工業大學 (深圳)  
HARBIN INSTITUTE OF TECHNOLOGY

# 实验作业

开课学期: 2022 春季

课程名称: 计算机组成原理 (实验)

实验名称: 直接映射 Cache 设计

实验性质: 综合设计型

实验学时: 4 地点: T2506

学生班级: 12

学生学号: 200111205

学生姓名: 李聪

作业成绩: \_\_\_\_\_

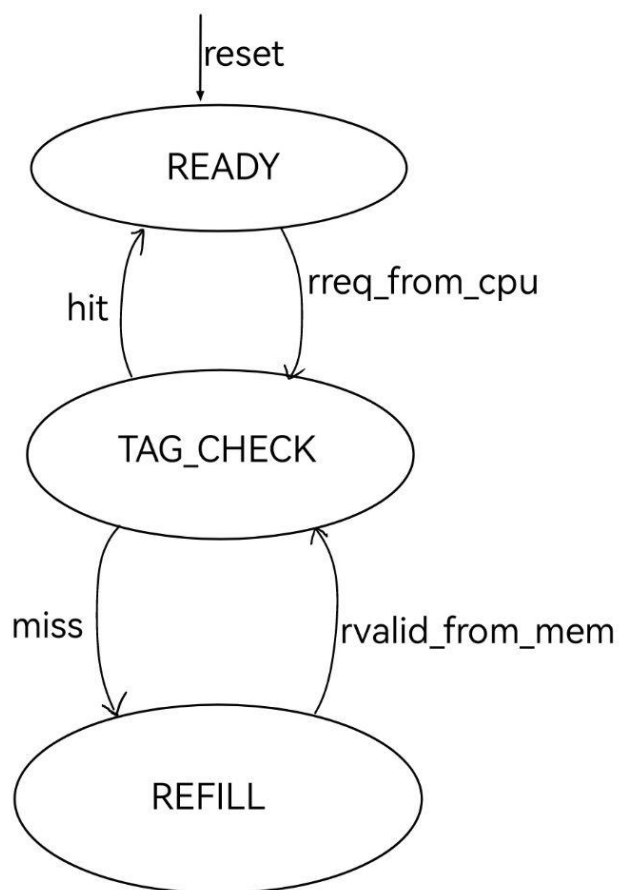
实验与创新实践教育中心制

2022 年 5 月

## 一、Cache 模块设计

（画出读、写的状态转移图，并描述状态之间的转移关系和转移条件、以及每个状态需要完成什么操作。）

读的状态转移图：

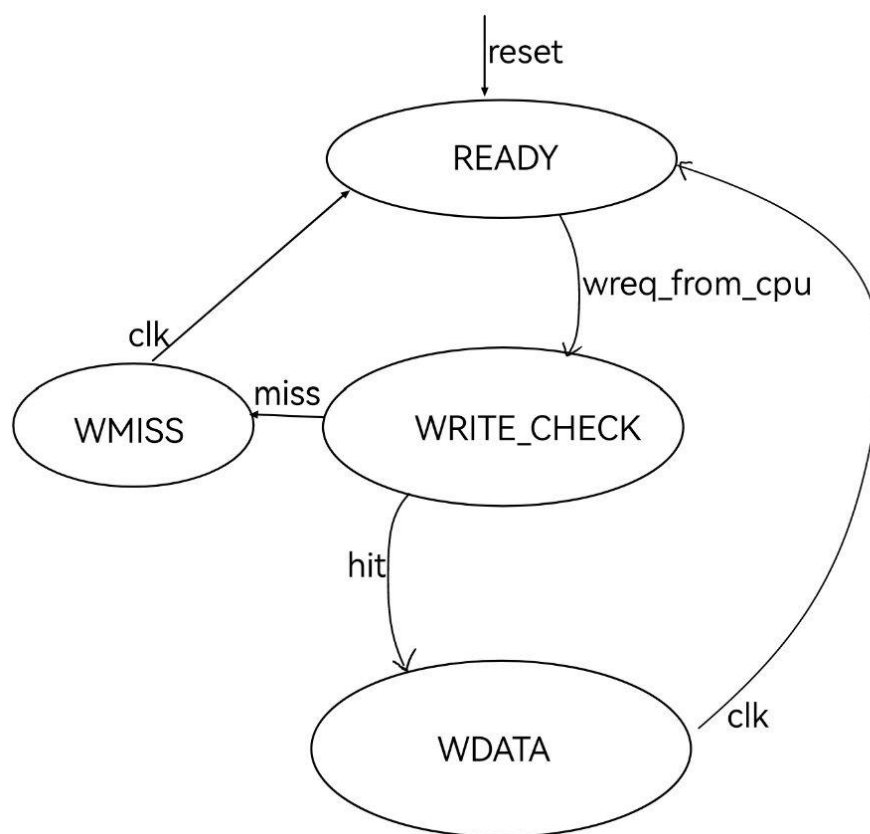


**READY** 状态：Cache 处在就绪状态

**TAG\_CHECK** 状态：检查是否命中 Cache

**REFILL** 状态：从内存中取出需要的块，加上标签与有效位标记后存入 Cache 的存储体。

写状态转移图：



READY 状态：Cache 处在就绪状态

WRITE\_CHECK 状态：检查是否命中 Cache

WDATA 状态：写命中，向内存和 Cache 写入数据。

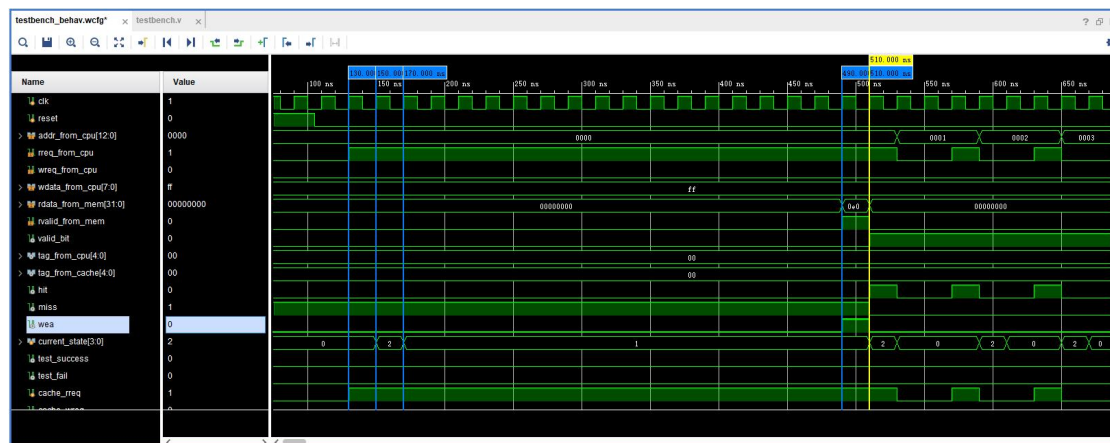
WMISS 状态：写缺失，hit 保持低电平，不修改内存。

## 二、 调试报告

（仿真截图及时序分析，要求包含读命中、读缺失、写命中及写缺失共四种情况的分析，且每种情况需列举 2 个测试用例进行分析。）

Current\_state 的表示：READY = 0000、TAG\_CHECK = 0010、REFILL = 0001、WRITE\_CHECK = 0011、WMISS = 0100、WDATA = 0101；

读缺失例 1：



130ns 时，rreq\_from\_cpu 变为 1，cpu 发出读请求

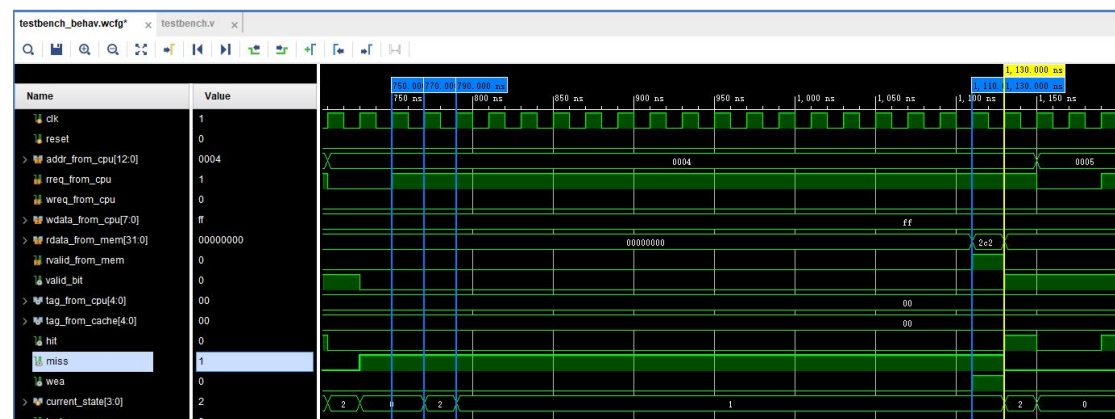
150ns，current\_state 变为 0010，进入 TAG\_CHECK 状态，由于此时 valid\_bit 值为 0，故读缺失，符合预期。

170ns，current\_state 变为 0001，进入 REFILL 状态。

490ns，主存读取完成标记 rvalid\_from\_mem 为 1，说明主存读取完毕，于是将主存读取的数据加上标记和有效位后写入 cache，wea 变为高电平，符合预期。

510ns，cache 更新完毕，进入 TAG\_CHECK 状态，此时 tag\_from\_cpu=tag\_from\_cache，且 valid\_bit=1，故命中，符合预期。

## 读缺失例 2:



750ns 时, `rreq_from_cpu` 变为 1, cpu 发出读请求

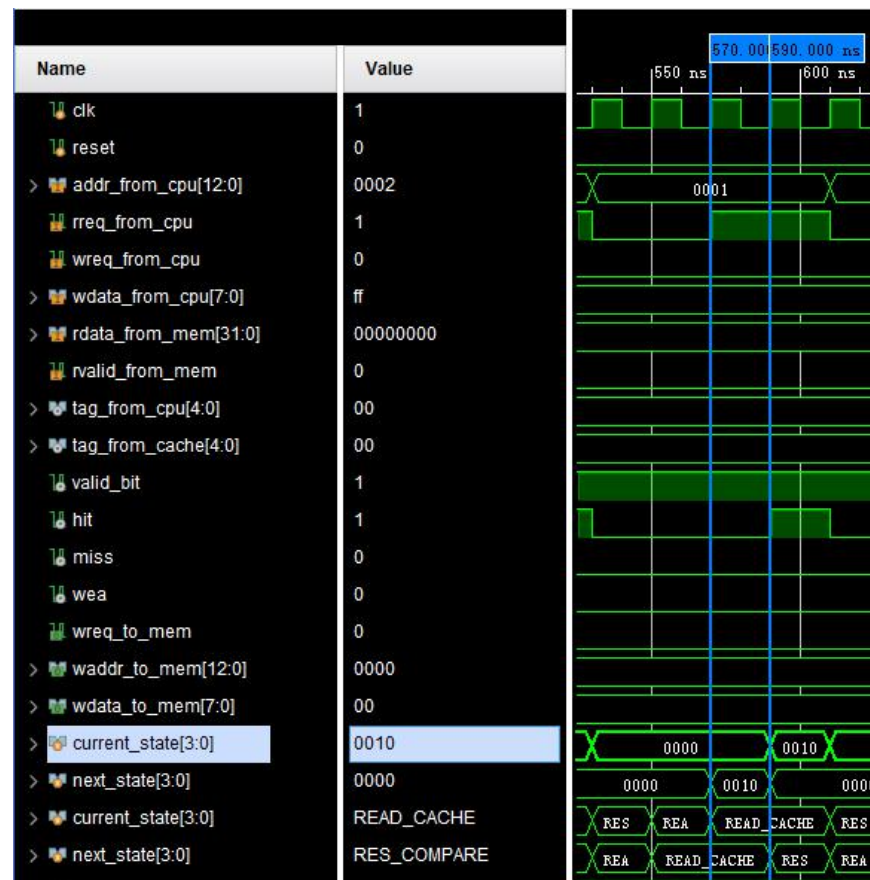
770ns, `current_state` 变为 0010, 进入 TAG\_CHECK 状态, 由于此时 `valid_bit` 值为 0, 故读缺失, 符合预期。

790ns, `current_state` 变为 0001, 进入 REFILL 状态。

1100ns, 主存读取完成标记 `rvalid_from_mem` 为 1, 说明主存读取完毕, 于是将主存读取的数据加上标记和有效位后写入 cache, `wea` 变为高电平, 符合预期。

1300ns, cache 更新完毕, `current_state` 变为 0001, 进入 TAG\_CHECK 状态, 此时 `tag_from_cpu`=`tag_from_cache`, 且 `valid_bit`=1, 故命中, 符合预期。

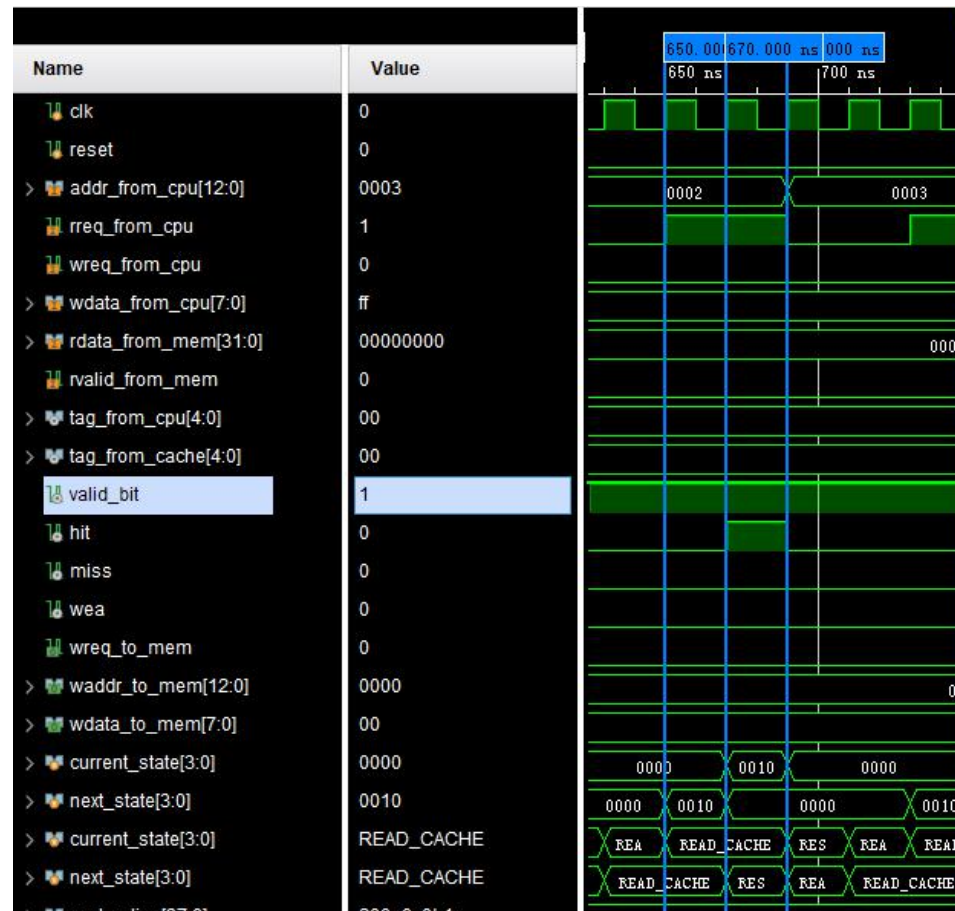
## 读命中例 1



570ns 时, rreq\_from\_cpu 变为 1, cpu 发出读请求

590ns, current\_state 变为 0010, 进入 TAG\_CHECK 状态, tag\_from\_cpu=tag\_from\_cache, 且 valid\_bit=1, 故读命中, hit 变为 1, 符合预期。

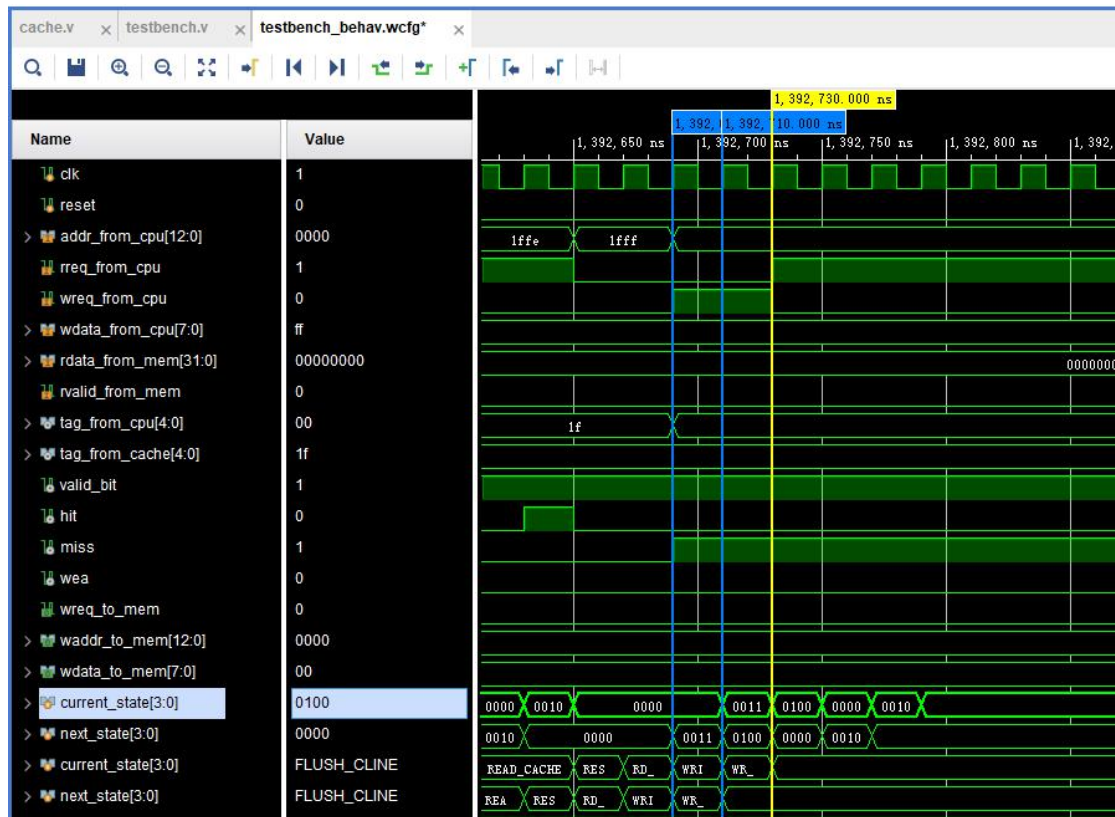
读命中例 2:



650ns 时, rreq\_from\_cpu 变为 1, cpu 发出读请求

670ns, current\_state 变为 0010, 进入 TAG\_CHECK 状态, tag\_from\_cpu=tag\_from\_cache, 且 valid\_bit=1, 故读命中, hit 变为 1, 符合预期。

写缺失例 1:



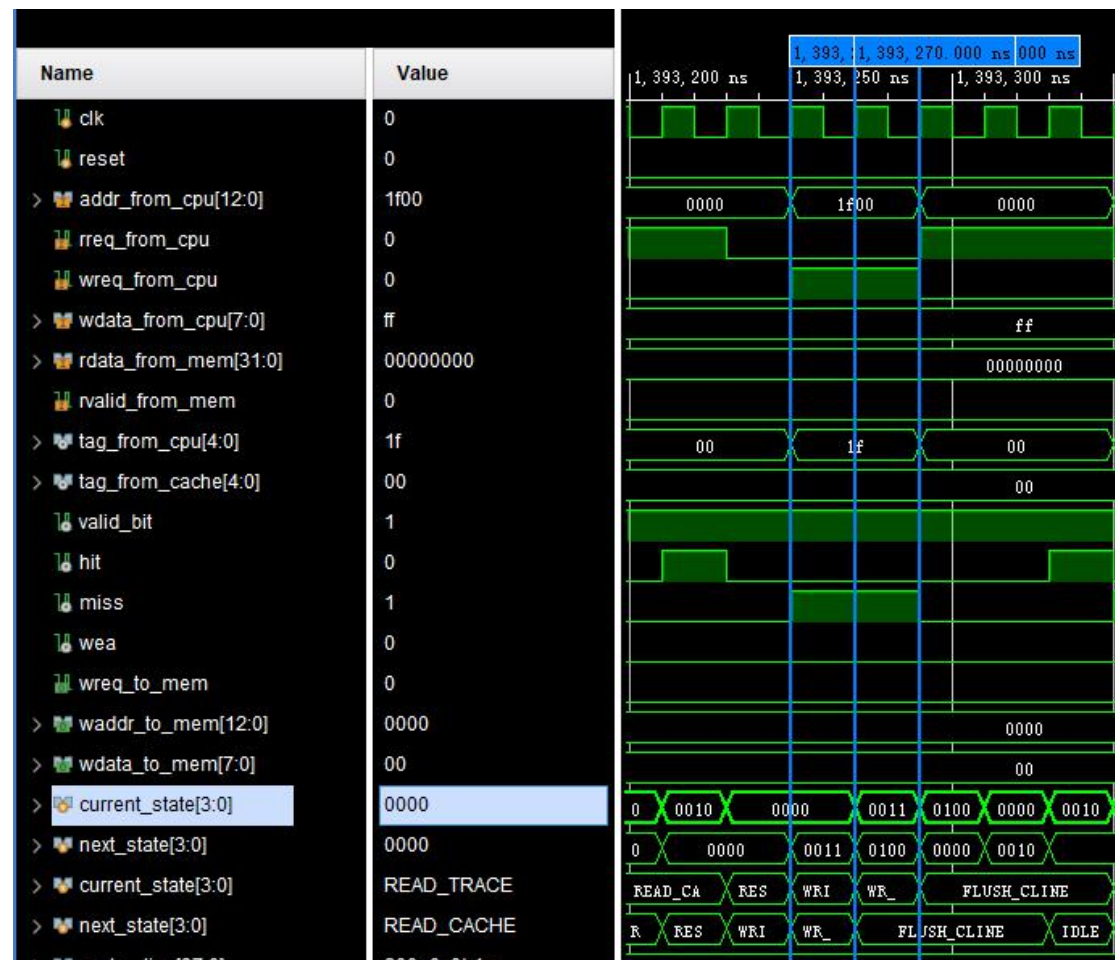
1392690ns, wreq\_from\_cpu 变为 1, cpu 发出写请求

1392710ns, current\_state 变为 0011, 进入 WREIT\_CHECK 状态, 由于 tag\_from\_cpu=00, tag\_from\_cache=1f, 二者不相等, 故写缺失, hit=0, miss=1, 符合预期

1392730ns, current\_state 变为 0100, 进入 WMISS 状态, hit 保持低电平, wreq\_to\_mem = 0, 不写主存, 符合预期



写缺失例 2:

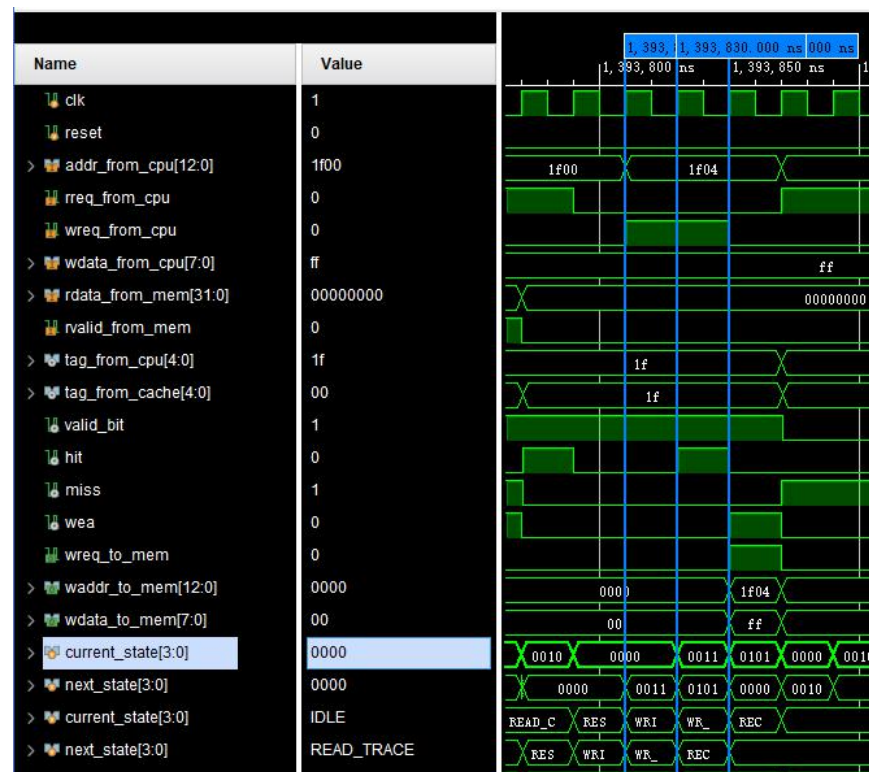


1393250ns, `wreq_from_cpu` 变为 1, cpu 发出写请求

1393270ns, `current_state` 变为 0011, 进入 WREIT\_CHECK 状态, 由于 `tag_from_cpu`=00, `tag_from_cache`=1f, 二者不相等, 故写缺失, `hit`=0, `miss`=1, 符合预期

1393290ns, `current_state` 变为 0100, 进入 WMISS 状态, `hit` 保持低电平, `wreq_to_mem` = 0, 不写主存, 符合预期

## 写命中例 1:



1393810ns, wreq\_from\_cpu 变为 1, cpu 发出写请求

1393830ns, current\_state 变为 0011, 进入 WREIT\_CHECK 状态, 由于 tag\_from\_cpu=1f, tag\_from\_cache=1f, 二者相等, 且 valid\_bit=1, 故写命中, hit=1, 符合预期

1393850ns, current\_state 变为 0101, 进入 WDATA 状态, wreq\_to\_mem 变为 1, 向主存写入数据, 同时 wea 变为 1, 向 cache 写入数据, 符合预期

