



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: T2 507
学生班级: 20 级 08 班
学生学号: 200210231
学生姓名: 王木一
作业成绩: _____

实验与创新实践教育中心制

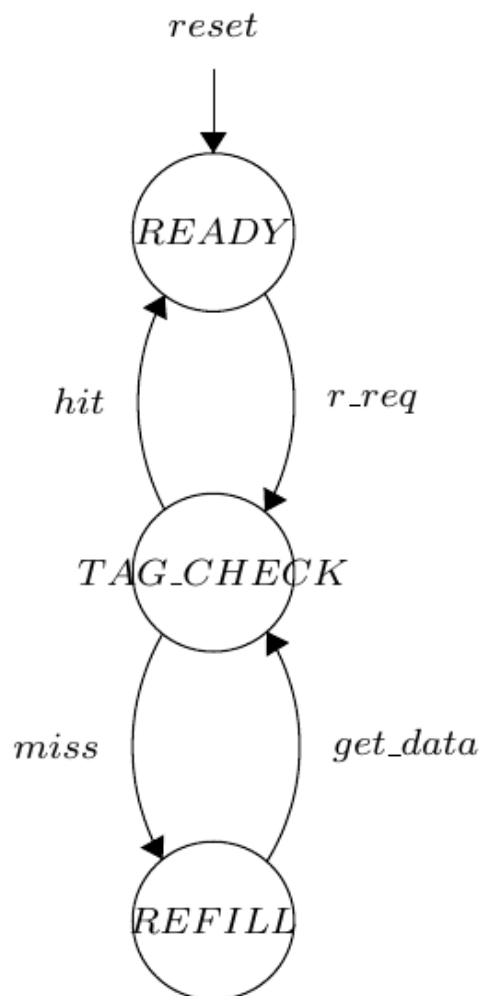
2022 年 5 月

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。

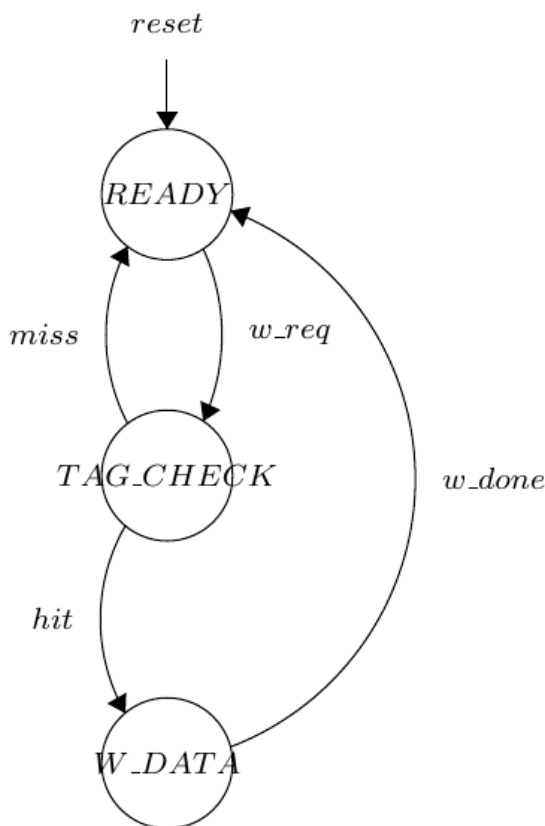
读操作：

1. 置位 `reset` 后，状态初始化为 `READY`。
2. `READY`：此状态下，Cache 等待 CPU 的读请求。①若 CPU 请求读 Cache，即 `r_req=1`，状态转移到 `TAG_CHECK`，判断是否命中；②若 CPU 未请求读 Cache，状态仍为 `READY`，继续等待 CPU 发出读请求。
3. `TAG_CHECK`：此状态下，Cache 判断是否读命中。①若读命中，即 `hit=1`，CPU 成功从 Cache 中读出数据，状态转移到 `READY`，等待 CPU 发出下一个读请求；②若读缺失，即 `miss=1`（或 `hit=0`），状态转移到 `REFILL`，从主存中读取数据，重新装填 Cache。
4. `REFILL`：此状态下，Cache 从主存中重新读取数据，此时主存读使能设置为 1，主存读取地址设置为 CPU 传来的地址。之后将从主存读取到的数据连同标志位有效位重新写入 Cache 对应位置，此时需将 Cache 写使能置为 1。①若完成装填，`get_data=1`，状态转移到 `TAG_CHECK`，再次比较是否命中；②若未完成装填，状态不变，直至装填完成。



写操作

1. 置位 `reset` 后，状态初始化为 `READY`。
2. `READY`：此状态下，Cache 等待 CPU 的写请求。①若 CPU 请求写 Cache，即 `w_req=1`，状态转移到 `TAG_CHECK`，进入标志位检查；②若 CPU 未请求读 Cache，状态仍为 `READY`，继续等待 CPU 发出写请求。
3. `TAG_CHECK`：此状态下，Cache 进行标志位比较，判断是否写命中。①若写命中，即 `hit=1`，状态转移到 `W_DATA`，写直达法入数据；②若写缺失，即 `hit=0`（或 `miss=1`），状态转移到 `READY`，等待下一个写信号（实验未要求处理写缺失的情况）
4. `W_DATA`：此状态下，写直达法同时修改 Cache 和主存中对应位置的数据。需要将主存读使能置为 1，同时传入修改的数据和数据地址；此外，Cache 写使能置为 1，修改 Cache 中的对应值。①当完成修改，`w_done=1`，状态转移到 `READY`，等待下一个写信号；②若未完成修改，状态不变，直至完成修改。（实际中，在 `W_DATA` 一个周期中，即可完成修改，回到 `READY`。）



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。）

注：本次实验，分别设计了读、写两个状态机。分别设置的两组状态变量，`current_state` 和 `next_state` 为读状态机的变量；`current_state_w` 和 `next_state_w` 为写状态机的变量。两个状态机共用部分状态参数。

状态参数表：

状态	值
READY	0
TAG_CHECK	2
REFILL	1
W_DATA	3

用例对照表：

用例编号	对应情况
#1-1-1/2	读缺失
#1-2-1/2	读命中
#2-1-1/2	写缺失
#2-2-1/2	写命中

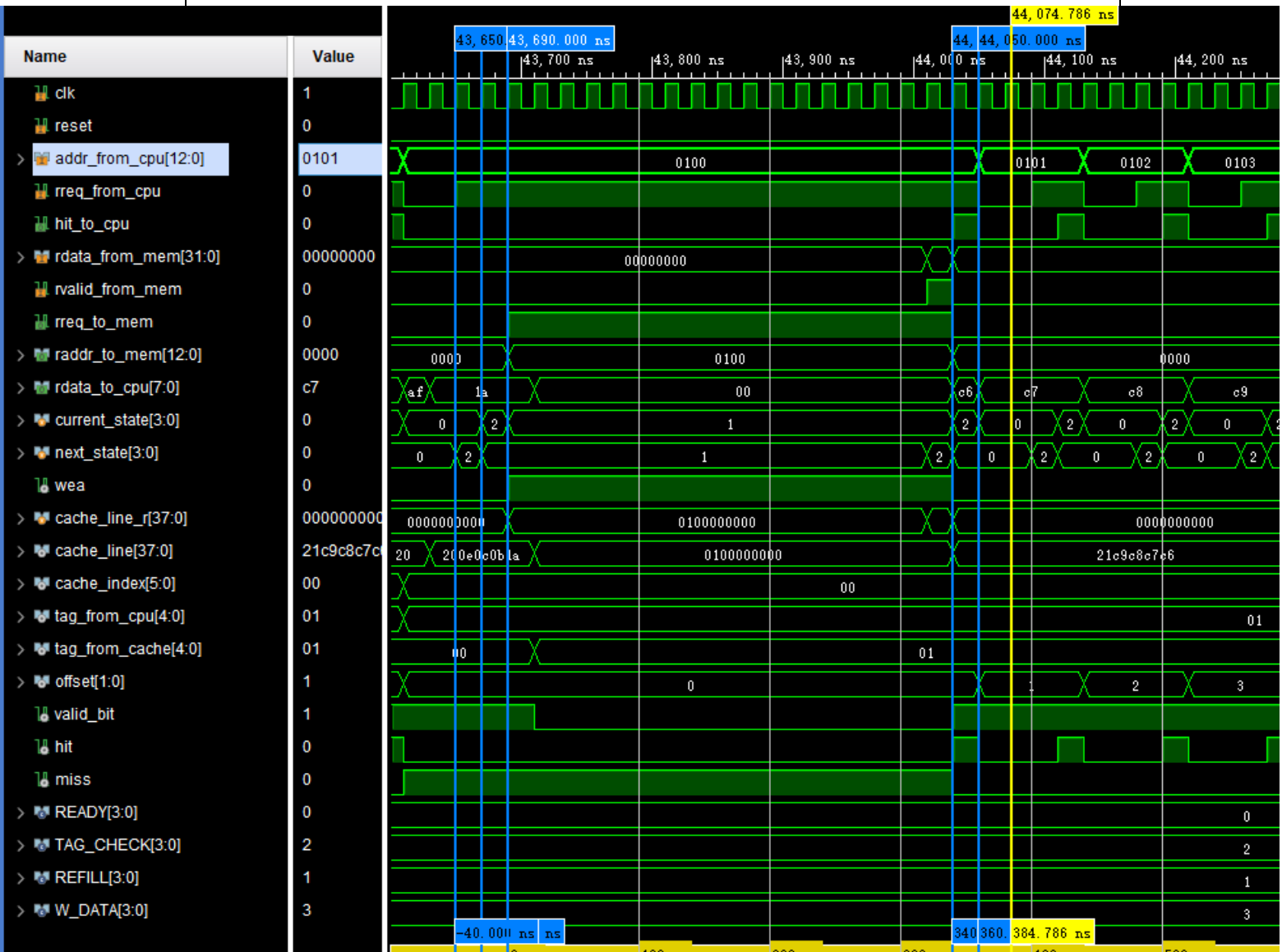
读操作

用例 #1-1-1 读缺失，读取地址 0000



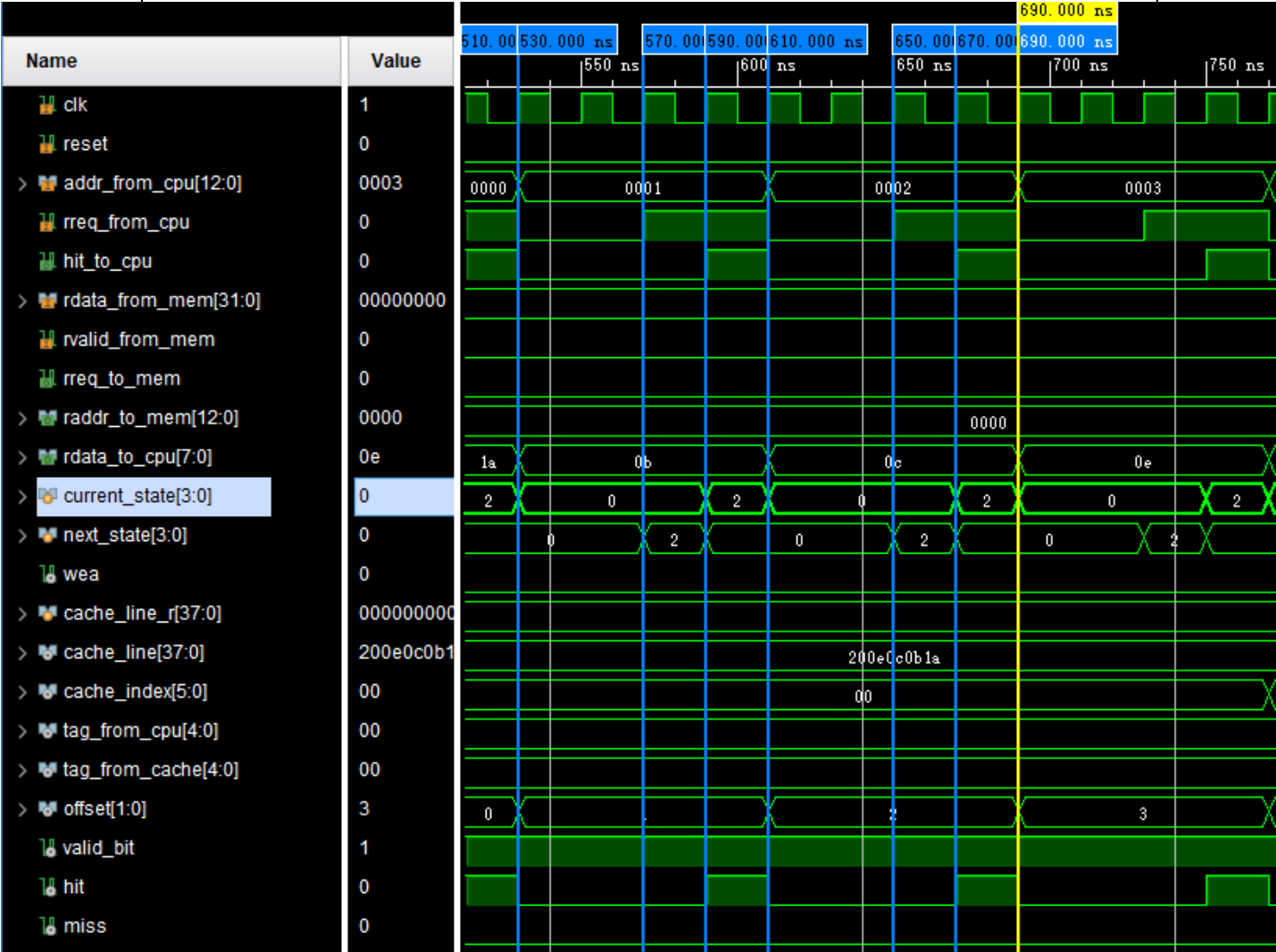
时间 (ns)	当前状态	操作及备注
130-150	READY	CPU 发出读请求, rreq_from_cpu=1, 欲读取主存地址为 0000 的数据, 状态转移到 TAG_CHECK
150-170	TAG_CHECK	此状态下, 判断是否命中。此时 Cache 读出位置 00 的数据, cache_index=00。由于有效位为 0, 读缺失, hit=0。状态转移到 REFILL。
170-510	REFILL	此状态下, Cache 从主存相应位置 0000 读取数据 (rreq_to_mem=1), 重新写入 Cache (wea=1)。当成功从主存读出数据, 写入 Cache 后, rvalid_from_mem=1, 状态转移到 TAG_CHECK
510-530	TAG_CHECK	此状态下, 判断是否命中。此时有效位为 1, cpu 和 cache 标志位相同, 读命中, hit=1, 成功输出数据 rdata_to_cpu=1a, 状态转移到 READY
530~	READY	等待 CPU 发出下一读请求

用例 #1-1-2 读缺失，读取地址 0100



时间 (ns)	当前状态	操作及备注
43650-43670	READY	CPU 发出读请求, rreq_from_cpu=1, 欲读取主存地址为 0100 的数据, 状态转移到 TAG_CHECK
43670-43690	TAG_CHECK	此状态下, 判断是否命中。此时 Cache 读出位置 00 的数据, cache_index=00。有效位为 1 但 tag_from_cpu=01, tag_from_cache=00, 读缺失, hit=0。状态转移到 REFILL。
43690-44030	REFILL	此状态下, Cache 从主存相应位置 0100 读取数据 (rreq_to_mem=1), 重新写入 Cache (wea=1)。当成功从主存读出数据, 写入 Cache 后, rvalid_from_mem=1, 状态转移到 TAG_CHECK
44030-44050	TAG_CHECK	此状态下, 判断是否命中。此时有效位为 1, cpu 和 cache 标志位相同, 读命中, hit=1, 成功输出数据 rdata_to_cpu=1a, 状态转移到 READY
44050~	READY	等待 CPU 发出下一读请求

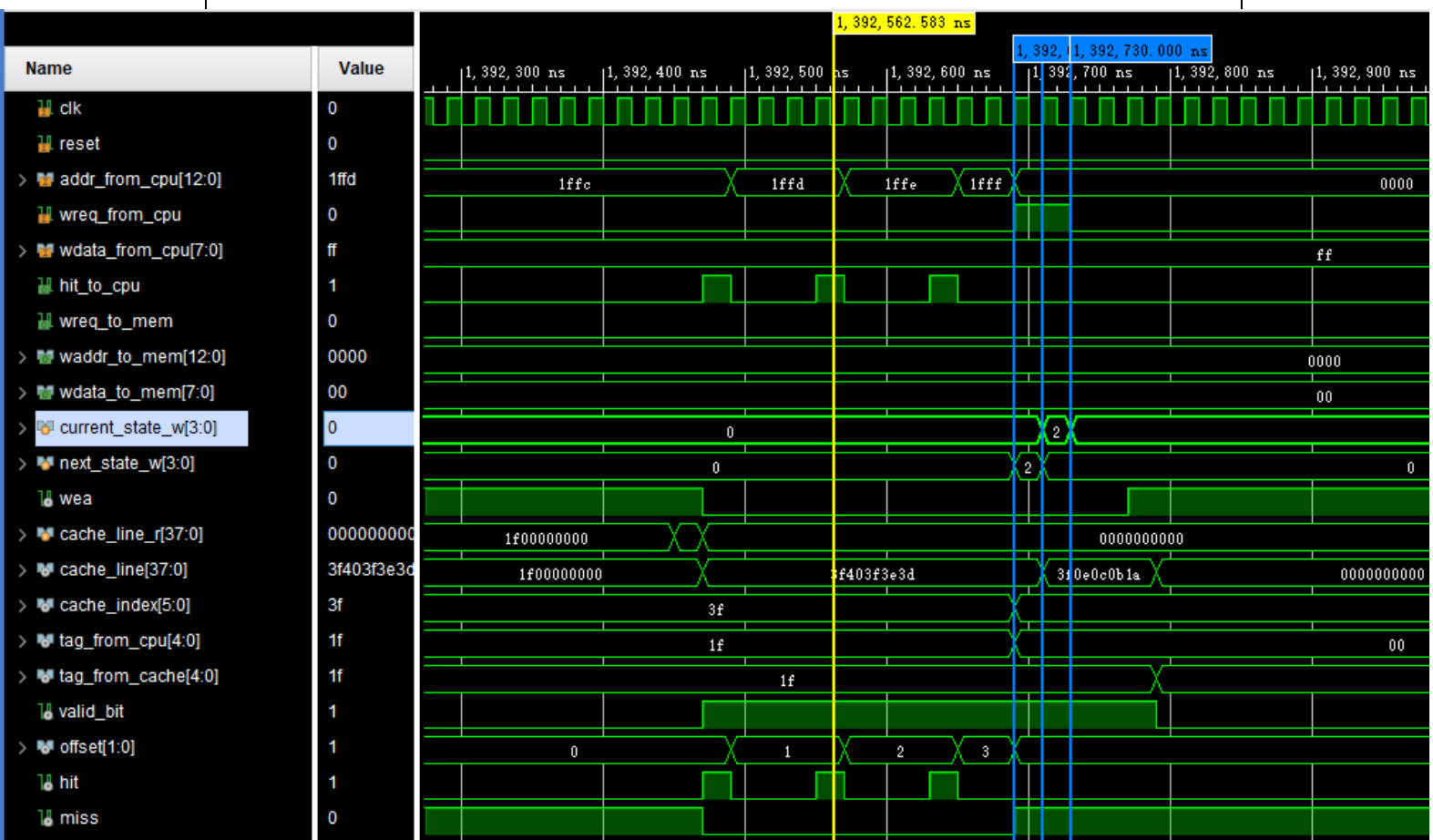
用例 #1-2-1 读命中，读取地址 0001 及 用例 #1-2-2 读命中，读取地址 0002



(读命中的两个用例写在一起)

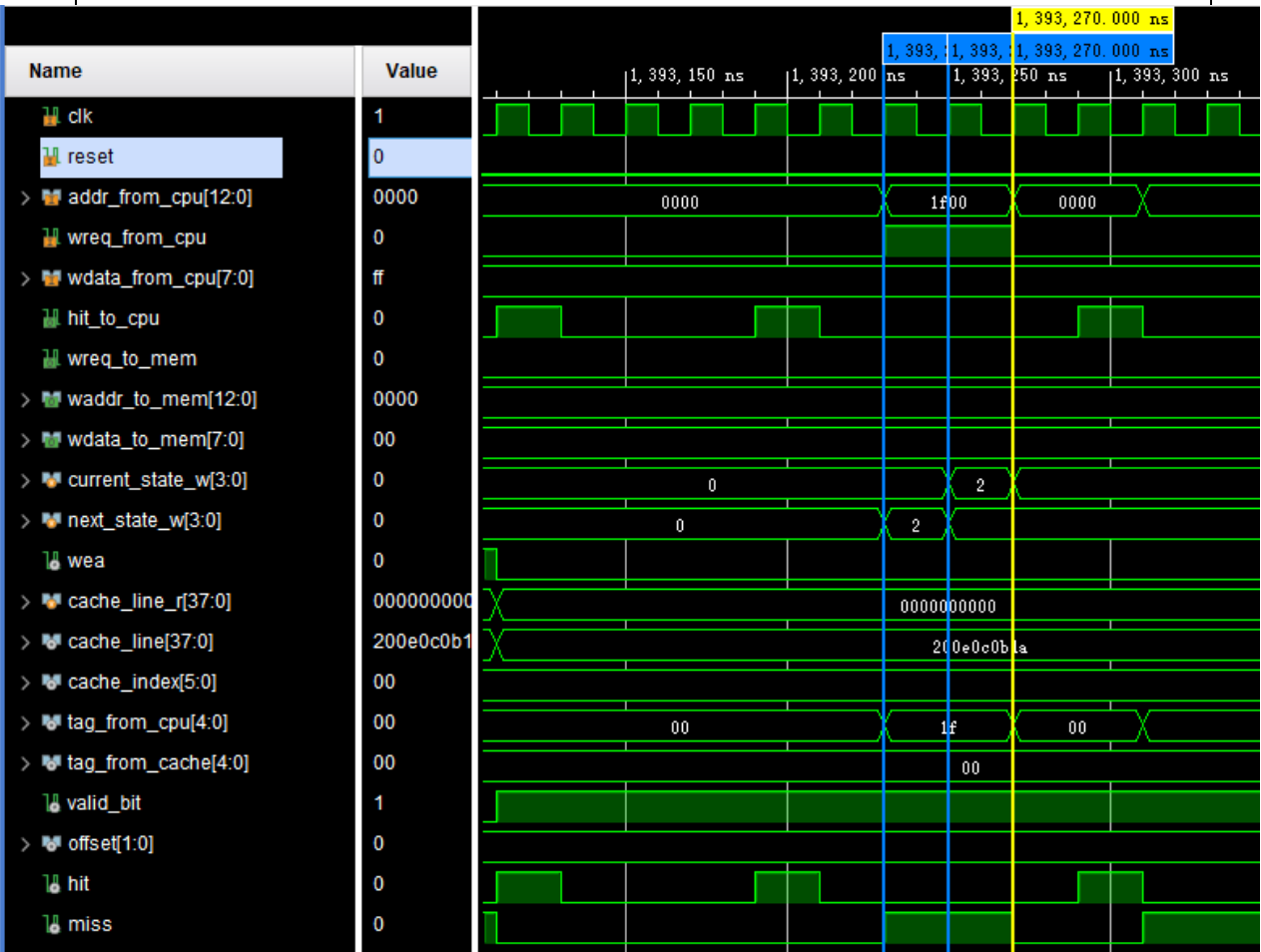
时间 (ns)	当前状态	操作及备注
用例 #1-2-1 读命中, 读取地址 0001		
570-590	READY	CPU 发出读请求, rreq_from_cpu=1, 欲读取主存地址为 0001 的数据, 状态转移到 TAG_CHECK
590-610	TAG_CHECK	此状态下, 判断是否命中。此时有效位为 1, cpu 和 cache 标志位相同 (都为 00, 由于 vivado 显示缘故, 截图波形上看不到), 读命中, hit=1, 成功输出数据 rdata_to_cpu=0b, 状态转移到 READY
610-650	READY	等待 CPU 发出下一读请求
用例 #1-2-2 读命中, 读取地址 0002		
650-670	READY	CPU 发出读请求, rreq_from_cpu=1, 欲读取主存地址为 0002 的数据, 状态转移到 TAG_CHECK
670-690	TAG_CHECK	此状态下, 判断是否命中。此时有效位为 1, cpu 和 cache 标志位相同 (都为 00, 由于 vivado 显示缘故, 截图波形上看不到), 读命中, hit=1, 成功输出数据 rdata_to_cpu=0c, 状态转移到 READY
690~	READY	等待 CPU 发出下一读请求

用例 #2-1-1 写缺失，写地址 0000



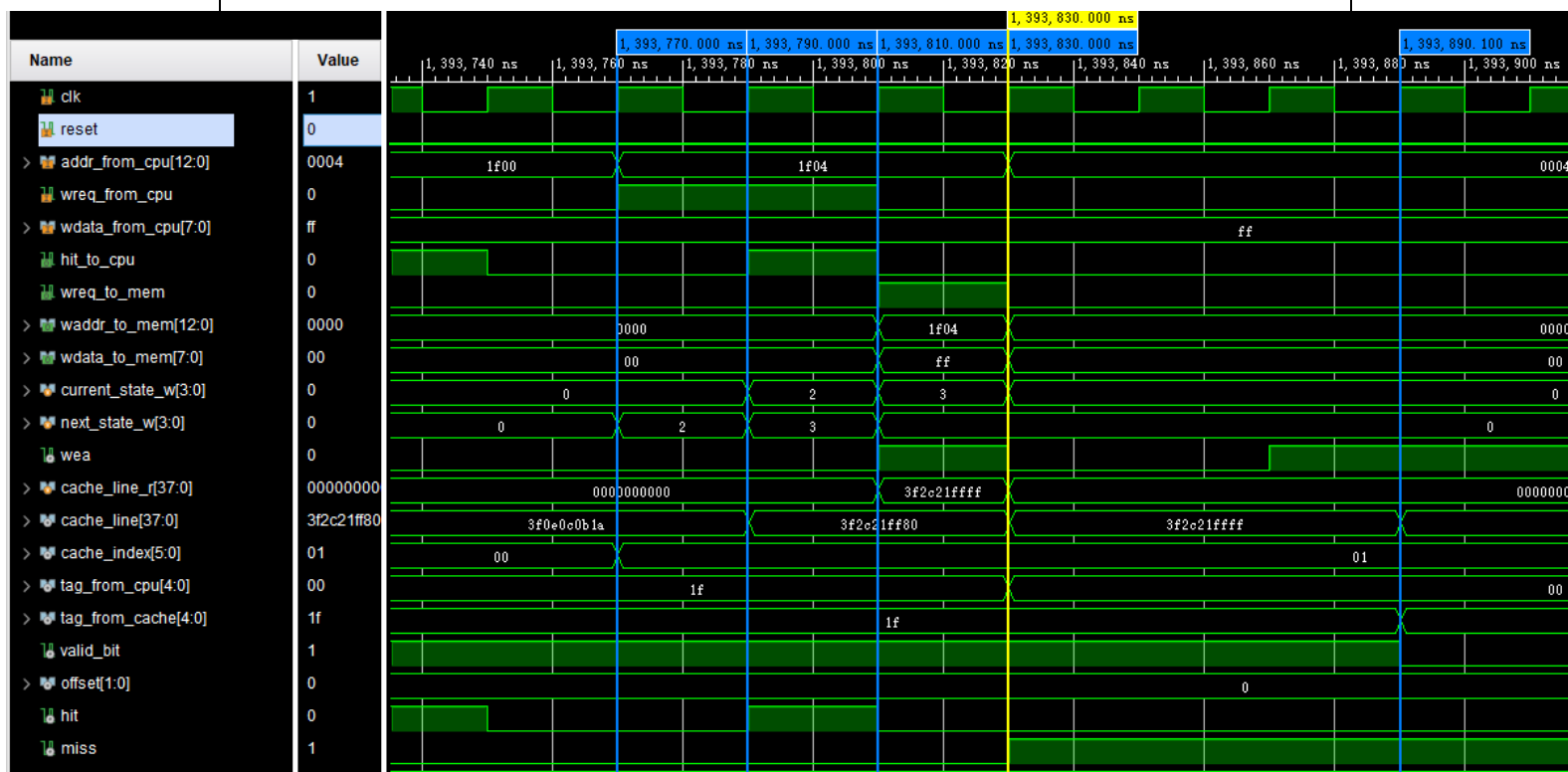
时间（ns）	当前状态	操作及备注
1392690-1392710	READY	CPU 发出写请求, wreq_from_cpu=1,欲写主存地址为 0000 的数据，状态转移到 TAG_CHECK
1392710-1392730	TAG_CHECK	此状态下，判断是否命中。此时 Cache 读出位置 00 的数据，cache_index=00。有效位为 1 但 tag_from_cpu=00,tag_from_cache=1f，写缺失，miss=1, hit=0。状态转移到 READY。（实验不要求处理写缺失）
1392730~	READY	等待 CPU 发出下一写请求

用例 #2-1-2 写缺失，写地址 1f00



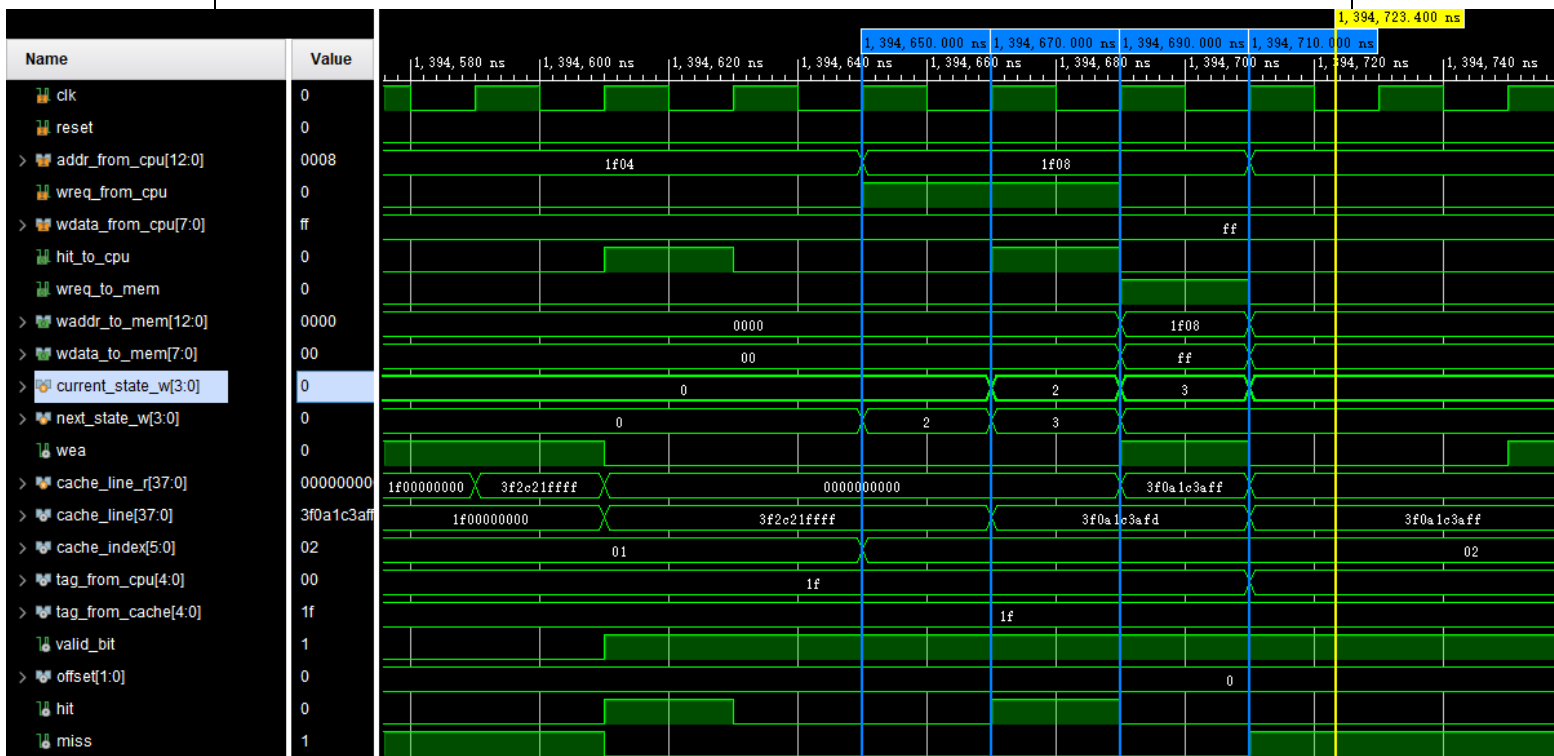
时间 (ns)	当前状态	操作及备注
1393230-1393250	READY	CPU 发出写请求, wreq_from_cpu=1, 欲写主存地址为 1f00 的数据, 状态转移到 TAG_CHECK
1393250-1393270	TAG_CHECK	此状态下, 判断是否命中。此时 Cache 读出位置 00 的数据, cache_index=00。有效位为 1 但 tag_from_cpu=1f, tag_from_cache=00, 写缺失, miss=1, hit=0。状态转移到 READY。(实验不要求处理写缺失)
1393270~	READY	等待 CPU 发出下一写请求

用例 #2-2-1 写命中，写地址 1f04



时间 (ns)	当前状态	操作及备注
1393770-1393790	READY	CPU 发出写请求, wreq_from_cpu=1, 欲写主存地址为 1f04 的数据为 ff, 状态转移到 TAG_CHECK
1393790-1393810	TAG_CHECK	此状态下, 判断是否命中。此时 Cache 读出位置 01 的数据, cache_index=01。有效位为 1, tag_from_cpu=tag_from_cache=1f, 写命中, hit=1, miss=0。状态转移到 W_DATA。
1393810-1393830	W_DATA	此状态下, 写直达法, 同时修改 Cache 和主存中的的数据。①修改 Cache, 写地址 cache_index 不变, 写入新的数据 cache_line_r=3f2c21ffff, 写使能 wea=1; ②修改主存, 写地址 waddr_to_mem=1f04, 写入数据 wdata_to_mem=ff, 主存写请求 wreq_to_mem=1。写完成后, 状态转移到 READY
1393830~	READY	等待 CPU 发出下一写请求

用例 #2-2-2 写命中，写地址



时间 (ns)	当前状态	操作及备注
1394650-1394670	READY	CPU 发出写请求, <code>wreq_from_cpu=1</code> , 欲写主存地址为 <code>1f08</code> 的数据为 <code>ff</code> , 状态转移到 TAG_CHECK
1394670-1394690	TAG_CHECK	此状态下, 判断是否命中。此时 Cache 读出位置 <code>02</code> 的数据, <code>cache_index=02</code> 。有效位为 <code>1</code> , <code>tag_from_cpu=tag_from_cache=1f</code> , 写命中, <code>hit=1</code> , <code>miss=0</code> 。状态转移到 W_DATA。
1394690-1394710	W_DATA	此状态下, 写直达法, 同时修改 Cache 和主存中的的数据。①修改 Cache, 写地址 <code>cache_index</code> 不变, 写入新的数据 <code>cache_line_r=3f0a1c3aff</code> , 写使能 <code>wea=1</code> ; ②修改主存, 写地址 <code>waddr_to_mem=1f08</code> , 写入数据 <code>wdata_to_mem=ff</code> , 主存写请求 <code>wreq_to_mem=1</code> 。写完成后, 状态转移到 READY
1394710~	READY	等待 CPU 发出下一写请求

