EE511 Computer Architecture

Project 3. 2-way set-associative Cache Design in Verilog HDL

E-mail: bsjang@ics.kaist.ac.kr / jylee@ics.kaist.ac.kr

Due date: Nov. 26, 2022

1. Project Outline

In project 3, you will implement the 2-way set-associative cache in Verilog HDL. This is an individual project.

- ✓ Do NOT use 'initial' and 'delay' statement except for testbench.
- ✓ You can use 'Icarus Verilog' if you do not have Verilog compile environment. Otherwise, you can use your compiler if you have one.

2. Specifications

(1) Memory

- ✓ Both the main memory and the on-chip memory are single-port synchronous RAM.
- ✓ Little endian.
- ✓ Data access latency

  Main memory : 10 cycles

  On-chip memory : 1 cycle
- ✓ Capacity

  Main memory : 64 KB

  Cache data memory : 4 KB (2 KB per 1 way excluding valid, dirty, LRU bits and tag memory)

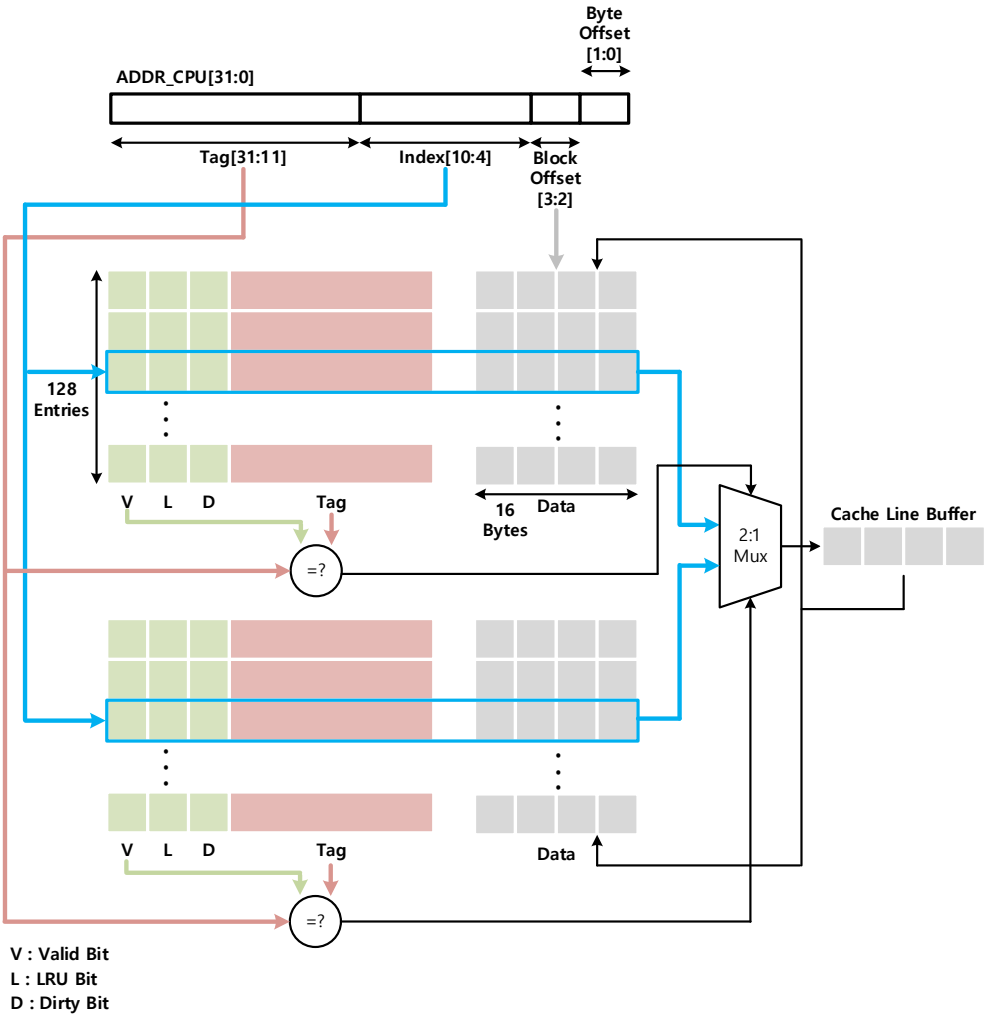
(2) Replacement policy

- ✓ Least Recently Used (LRU)

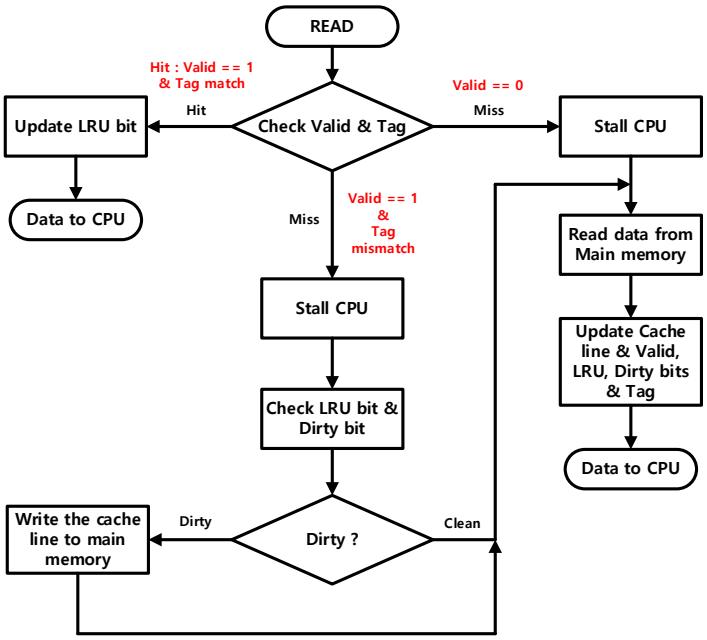
(3) Write strategy

- ✓ Write back : use a dirty bit
- ✓ Write allocate

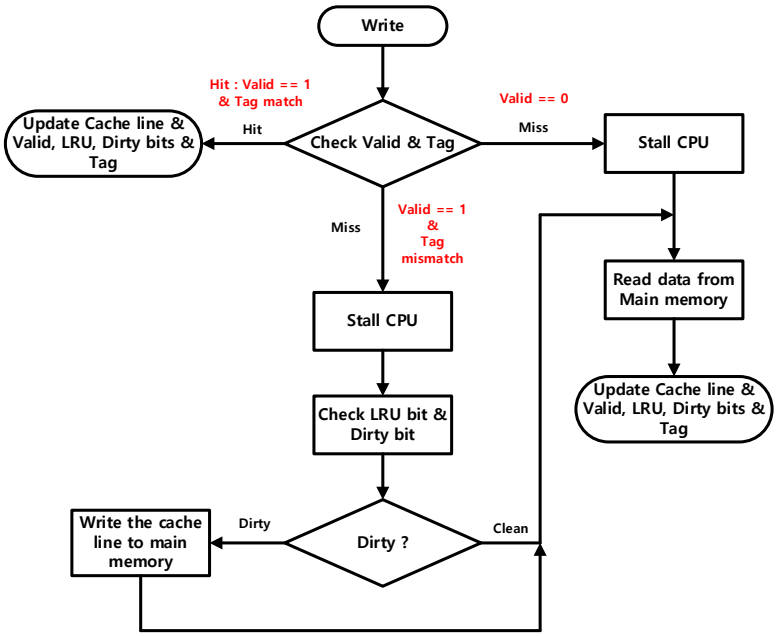
(4) Data path



(5) Flow diagram for read

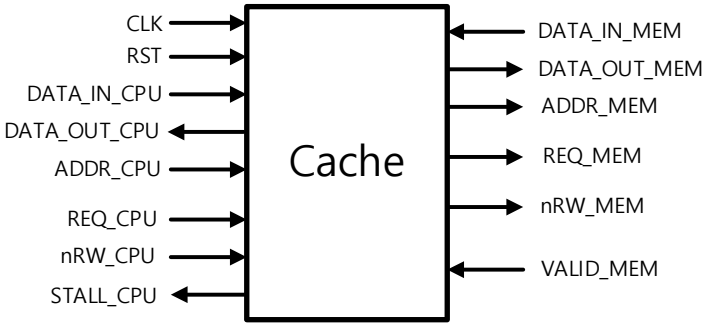


(6) Flow diagram for write



3. Input/Output Ports

(1) Cache controller



Do NOT modify the input and output ports of CACHE_CONTROLLER.v. If you modify any of them, it is not possible to evaluate your design. You will get the lowest grade.

The input/output ports should be

---

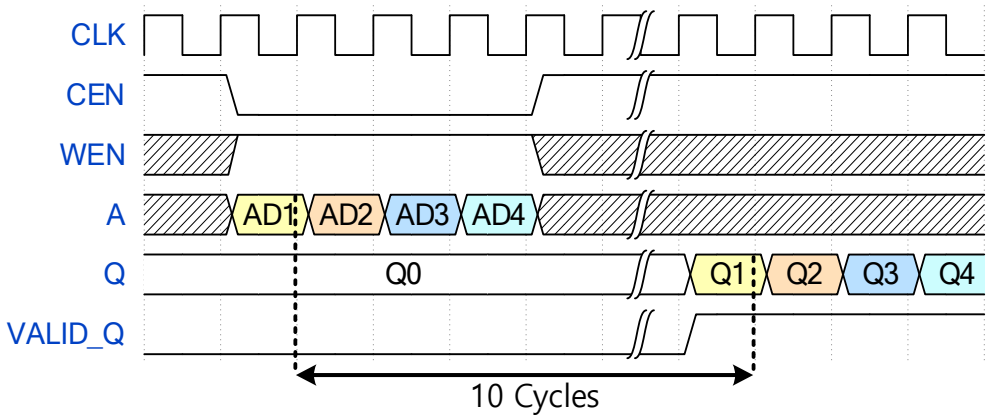
input		CLK	: clock
input		RST	: reset, active high
input	[31:0]	DATA_IN_CPU	: data from the CPU
output	[31:0]	DATA_OUT_CPU	: data into the CPU
input	[31:0]	DATA_IN_MEM	: data from the memory
output	[31:0]	DATA_OUT_MEM	: data into the memory
input	[31:0]	ADDR_CPU	: data address given by CPU
output	[31:0]	ADDR_MEM	: data address given to memory

input	REQ_CPU	: request signal from CPU to cache
output	REQ_MEM	: request signal from cache to memory
input	nRW_CPU	: CPU's memory(cache) read/write
output	nRW_MEM	: Cache's memory read/write
output	STALL_CPU	: stall signal from cache to CPU (1: stall / 0: not stall)
input	VALID_MEM	: indicating the validity of the data from mem to cache

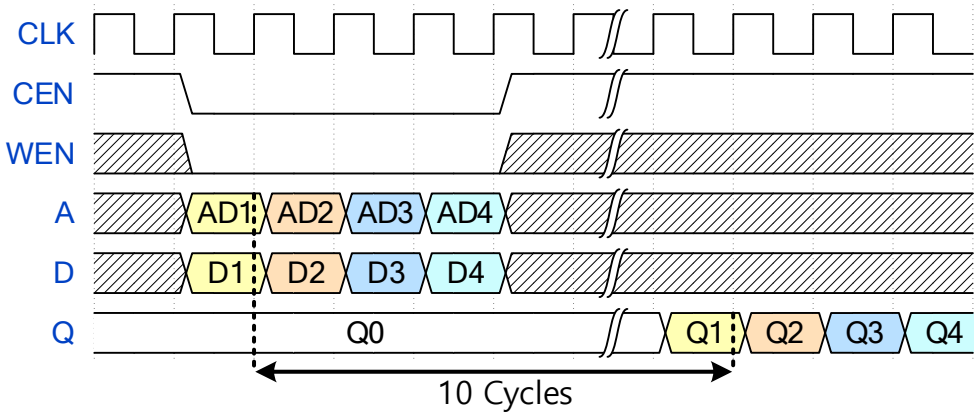
---

(2) Main memory

✓ Read timing

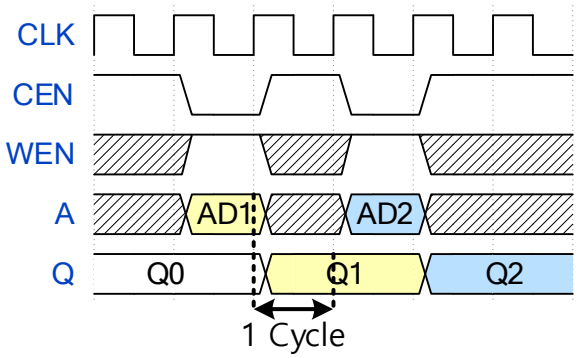


✓ Write timing

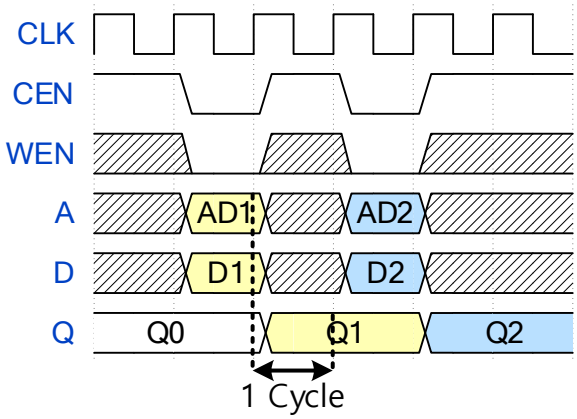


(3) On-chip memory

✓ Read timing



✓ Write timing



4. Attached Files

(1) Project3_template

✓ CACHE_CONTROLLER.v - CACHE top module:

✓ model.v - on-chip RAM of cache controller and main memory model

✓ testbench.v - For cache design verification

✓ test.f - A list of all related Verilog files

✓ data.hex - initialize the main memory (write your own data)

✓ init_ftram0/1.hex - initialize the flag & tag on-chip RAM of the cache (all zeros)

✓ init_dram0/1.hex - initialize the data RAM of the cache (all zeros)

(2) Documents

✓ [EE511] Project3

5. Submission

✓ **Due date: Nov. 26 (Sat.) 23:59**

✓ All Verilog HDL source file (*.v) and a list of Verilog files (test.f)

A. Do not modify the name of CACHE_CONTROLLER.v and top module name

B. File containing the top module: CACHE_CONTROLLER.v

C. Top module name: CACHE_CONTROLLER

✓ Project Report Document

A. File format: {StudentNumber}_{YourName}_project3_report.pdf

e.g.) 20221234_GildongHong_project3_report.pdf

B. The report should contain:

- Explanation on your design itself and the design procedure or methodology
- Overall block diagram of data path and control unit
- Verification of your design

C. 1 column per page. No more than 5 pages.

D. Language : Korean / English

✓ **Submit) A zip file**

A. Zip file: ZIP all the relevant files (**source code** and **report**) and upload it to **KLMS.**

B. Zip file name: {StudentNumber}_{YourName}_project3.zip

e.g.) 20221234_GildongHong_project3.zip

(Follow the format of the file name, or you will get deduction)

✓ **If you copy other's work, you will not receive any credit.**

✓ **After due date, no more submission is permitted.**