Project 5 Discussion

CS 3339

Lee Hinkle

Texas State University, San Marcos

With deep acknowledgement to Dr. Martin Burtscher and Ms. Molly O'Neil

Project 5 at a glance

iMem and dMem – memory that always responds in a single cycle

Only a well designed and fairly small cache can do this

And then only for accesses that “hit” the cache

Instructions are fetched from the L1 instruction cache

lw and sw accesses are handled by the L1 data cache

In Project 5 you will simulate the accesses to an L1 data cache

Project 5 Tasks

“you only have to model the hit/miss/timing behavior of the cache, you do not have to actually store any data in your cache model”

“Every time an access is made to the cache model, the cache model should return the number of cycles that the processor must stall in order for that access to complete.”

The output is updated to provide memory/cache related statistics as well as the new cycle count

This is the solution to Project 5

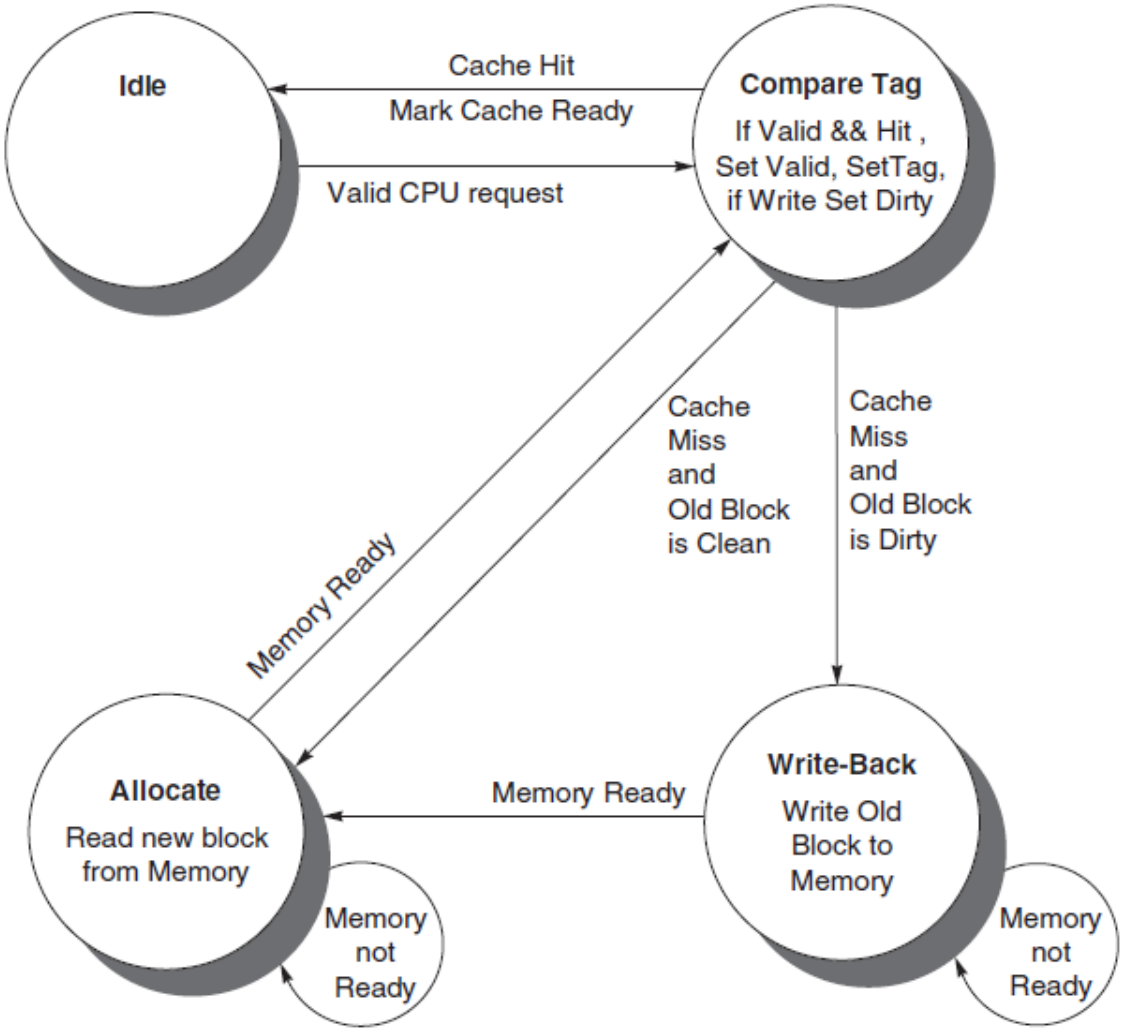


FIGURE 5.40 Four states of the simple controller.

Pg 465 of COD Textbook

Data Structures?

"The simulated data cache should store 1 KiB (1024 Bytes) of data in block sizes of 8 words (32 bytes). It should be 4-way set associative, with a round-robin replacement policy and a write policy of write-back write-allocate."

Remember you are simulating the cache controller to determining number of stall cycles, the data is a black box. You don't have to store the actual lw/sw data!

InClass7 solution is posted on TRACS and can give you clues here (but it is not the same config)

Round Robin (which way to use)

- round robin value for each set, initially zero
- Block is added to the cache (happens after a miss)
  - update the round robin to point to the next block in the set
  - e.g. 0 goes to 1, 1 to 2, 3 wraps back to 0
- cache hit = no change
- cache miss = which way to put the new value in?
  - it's the way that is pointed to by the round robin value (the oldest block in the set)
  - This is the point where you need to check the dirty bit to see if the block being evicted needs to be written to memory or simply discarded

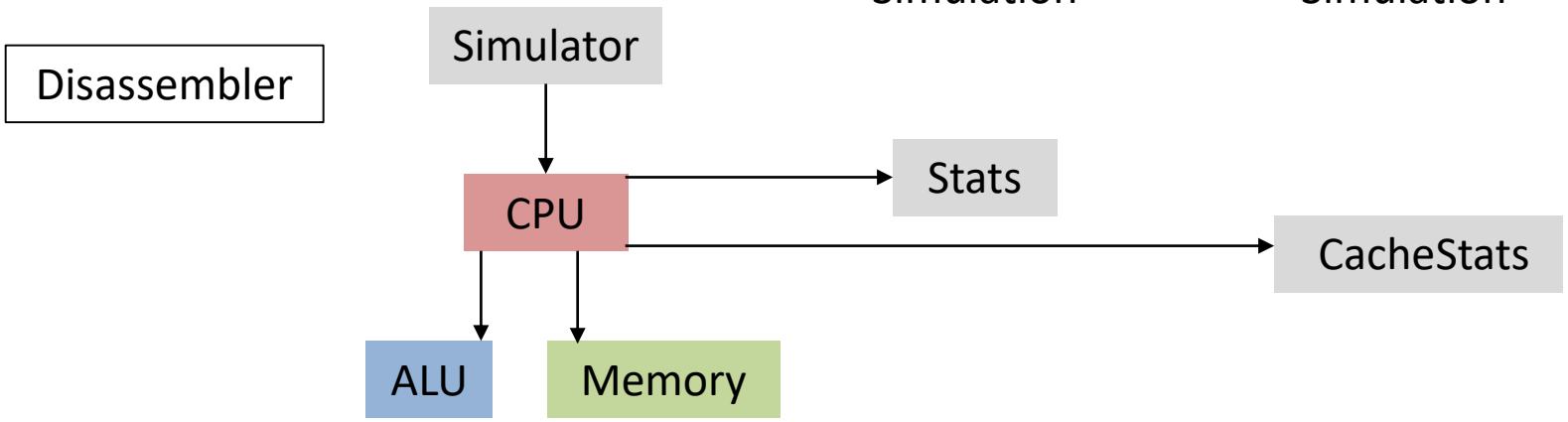
A UML-ish representation of the projects

Project 1
Disassembler

Project 2
Emulator

Project 3 & 4
Pipelining
Simulation

Project 5
data Cache
Simulation



Convert machine instructions to assembly

Setup control signals to emulate processor functions

Count the clocks required in a pipelined implementation

Add in the stalls and compute cache stats for data cache