Project 2

Memory Hierarchy Design

Project Checkoff: Thursday April 27th, 2020

Report Due: Tuesday April 29th, 2020

Overview:

In this lab, you will improve performance of the memory you designed in the previous lab by utilizing a simple cache structure. This project has four main steps. First of all, you should design a separate cache memory and test/verify its functionality. Then, you must modify your MIPS. Next step, you should integrate them together and perform some tests and measurements.

Design Procedure:

A cache holds commonly used memory data. The processor first seeks data in a small fast cache. If the cache hits, the data is available immediately. If the cache misses, the processor fetches the data from main memory and places it in the cache for future use. To accommodate the new data, the cache must replace old data. The goal of this lab is to design a two-way set associative cache. Generally, an N-way set associative cache (where N is the degree of associativity of the cache) reduces "conflicts" by providing N blocks in each set where data mapping to that set might be found. Each memory address still maps to a specific set, but it can map to any one of the N blocks in the set. Fig. 1 shows structure of a two-way set associative cache.

Step1: In previous lab, you designed a 32-bit MIPS. Open your project in Modelsim Environment.

Step2: Create a new source and name it cache_wb. The cache must be 32 KByte two-way set associative four-word block and it must follow write back policy. Design this structure and verify its functionality by means of a simple test bench. You may use your preferred design methodology. Generally, you will need a controller, combination blocks and a simple SRAM.

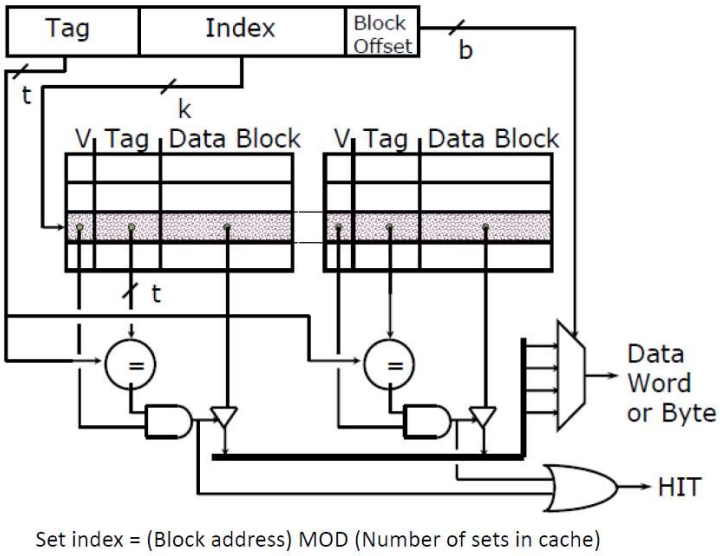

Fig. 1: A typical two-way set associative structure

Step3: You must make some changes to your data memory in order to emulate a realistic memory. For this lab, we assume it takes 20 cycles for the processor to access main memory. Modify your design accordingly.

✓ Don't forget to test and verify functionality.

Step4: Modify your processor (controller, hazard detector,...) in order to avoid any data hazard. Explain all your modifications in your report. Integrate the memory to your processor. Evaluate the performance of your processor using each of the provided benchmarks.

✓ Do not include the cache at this step.

Step5: Create a new source and integrate all of your modules into it. Create a test bench and verify functionality of the whole system.

Step6: Use the provided test pattern and measure the performance of your design. What is the miss rate in each case? Explain all your conclusions. Compare the results of step 6 and 4. Is the miss rate of a two-way set associative cache always, usually, occasionally, or never better than that of a direct mapped cache of the same capacity and block size? Explain.

Questions:

1) Develop a fancy program and run it on your MIPS. Show the results.

Extra Credit:

1- Design a 32 KByte four-way set associative four-word block size cache and it must follow write back policy. Run some test patterns, measure miss rate and explain your results.

FAQs:

1. If we do not need a write buffer, how should we handle writes?

✓ The write buffer is not needed. When a write happens, you can stall the pipeline until the write finishes.

2. In this Project, what exactly is the main memory? Is it supposed to just be a cache for data memory?

✓ The main memory is your data memory designed in lab1. You should design a cash and modify your data memory to manage connection between cash and main memory.

3. There are no data hazards in this project because we already solved them in the project1. So, why do we need to modify our processor to avoid data hazards?

✓ Just make sure when you have a memory access, you can stall your processor.

4. Step 6 says to use a provided test pattern for the design. Do we assume this test pattern is simply our "fancy" program?

✓ Yes.

5. Grading?

✓ Your grade is mainly based on correct operation of MIPS. Functionality is the primary goal. 60% of your score is based on whether your MIPS works correctly or not and the rest of is related to your report.

6. What happens during the checkoff?

- ✓ You have 10 minutes to present your project. All of group members must be available during the presentation. You may bring your own laptop or use computers in ECI lab. Everybody has to explain the whole project and answer some questions.

7- What to turn in?

- ✓ Submit an organized Zip file containing below mentioned files to **zfahimi@ucsb.edu** by the deadline. Your report is very important. Start with introduction, illustration of instructions, and design methodology. **And then you may focus on each of the steps provided in the manual.** When describing each step, provide the code, the test bench and waveforms. Explain why your waveforms are correct and answer all questions. Organization and completeness of the report determine 40% of your score. Figures should be **readable** and you have to explain them in detail. Mention how many hours you have spent on this lab, your common mistakes in Verilog coding and lessons you learned. Finally, provide a conclusion and wrap up the project. Cite appropriately any references in the report. A folder containing the project files including all source files, test benches and waveforms. **Please heavily comment your code. Poorly commented codes will not be graded.**