Project #2

Memory Hierarchy Design

(Prepared by M. Reza Mahmoodi)

Lab Checkoff: Wednesday, 8-10 am, May 4th, 2016

Report Due: Friday, 11:59 pm, May 6th, 2016

Overview:

In this lab, you will improve the performance of the memory you designed for the previous lab by utilizing a simple cache structure. This lab has four main steps. First of all, you will design a separate cache memory. You may test and verify its functionality. Then, you will modify your MIPS and for the next step, you can integrate them together. Finally, you will 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.

Step 1: In previous lab, you designed a 32-bit MIPS. Open your project in Xilinx Synthesis Environment (ISE).

Step 2: Create a 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.
Step 3: 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 testing and functional verification.

Step 4: Modify your processor (controller, hazard detector, ...) in order to avoid any data hazard. Note all your modifications in the report. Integrate the memory to your process. (Do not include the cache at this step.) Evaluate the performance of your processor using each of the provided benchmarks.

Step 5: Create a new source and integrate all of your modules into it. Create a test bench and verify functionality of the whole system.

Step 6: Use the provided test patterns 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 step 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.

Extra Credit:

1- Redesign your cache. It must be 32 KByte four-way set associative four-word block size cache and it must follow write back policy. Run the test patterns, measure miss rate and explain your results.
2- Redesign your cache. It must be 32 KByte two-way set associative cache and it must follow write back policy. **Change the block size**, run the test patterns and justify your results.

3- Design L2: This cache must be 1MByte two-way set associative four-word block size and it must follow write back policy. Integrate all your modules, run test patterns and explain your results.

4- Map and place & route your design using targeted board.

5- Develop a fancy program and run it on your MIPS. Show the results. (This should be an interesting program).

FAQs:

1- Grading:

   Your grade is mainly based on correct operation. 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. (10% for each extra credit Qs)

2- What happens during the checkoff?

   You have 10 minutes to present your project. Both 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.

3- What to turn in?

   Submit an organized Zip file containing below mentioned files to mrmahmoodi@ece.ucsb.edu by the deadline.

   --A typed report in PDF format:

   Your report is very important. You may start with a detailed 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. This is very important. Explain why your waveforms are correct. Make sure you will 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. Provide how many hours you have spent on this lab, your common
mistakes in HDL coding and lessons you learned. Finally, provide a conclusion and wrap up the project. Cite any reference that you may have used.

--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.