May 1, 2017

PROJECT 2

Due: May 24, 2017

Direct mapped cache memory system

GROUP MEMBERS

1. Cesar Claros
2. Paulo Loma
3. Abel Claros

OBJECTIVES:

4. Design, simulate and implement a Direct-mapping cache memory system in VHDL using Altera Quartus Prime and compare it to a reference system without cache memory to verify the performance enhancement. Your system should have the following features:
   a. **Reference system:** modify the simple CPU example given in class to use a memory module that operates at 1/8 the speed of the CPU. The memory module can be implemented with the method of your choice (i.e. with the default logic available in the FPGA, the memory modules available in the FPGA accessed from the "IP Manager", etc.).
      i. Main memory size should be 4kbytes of 16 bits words (address width of 12 bits).
      ii. The speed of memory module is controlled by the memory clock, which has to be changed to 1/8 of the CPU clock (suggest a delay of 7 cycles after a memory request is made to deliver the data in the next clock cycle).
      iii. System modification will require changes, in addition to the Memory module, to the CPU module (modify the new address width) and to the controller module (to deal with the new memory timing).

   b. **Enhanced system:** Modify the reference system above to introduce a cache memory module that replaces the memory in the reference system. It should have a Direct mapping memory mechanism of the main memory with the following characteristics:
      i. A two-level memory with a slow (main) memory size of 4kbytes with 64-bit lines and a total of 1024 lines.
      ii. A fast (cache) memory with 32 words of 16 bits organized in 8 lines of 4 words per line.
      iii. Writing scheme: write back.
      iv. CPU will not access the main memory directly but only through the cache interface.

      Hint. Use a dual port memory that interface the CPU on one side and the main memory on the other side. The main memory is organized in words or blocks of 64 bits that are the same size of a cache line. When copying a word of 64 from the main memory to the desired cache line, each 16-bit words is read independently by the CPU.

For both systems:

i. Write a benchmark program for matrix multiplication that shows the impact of the target cache memory scheme in the performance of the architecture. Matrix size should be at least 10x10. The matrix values should be selected so it will not generate overflow.
ii. Objective 4: Design an interface that enables to run your benchmark program in the DE0-CV boards and debug the results.

Hint. You can display the results using the seven-segment or VGA displays. Halt the CPU when the output values are available and use one of the push buttons to resume displaying the results. You can instead visualize the results after you compute all the values if preferred.

Deliverables:
A project report will be submitted with the following information:

i. Introduction
ii. Design
    - Should contain code and information about the HW and SW used in the systems described in the three objective described above.
iii. Results
    - Should contain results of the simulation of the systems without cache memory, the cache memory block and the enhanced CPU with the cache memory.
    - Results of your performance evaluation on the architectures.
iv. Conclusion: conclusions about the implementation of the systems highlighting strength and weaknesses.

Report:
The report should not exceed 10 single spaced pages written with “Times New Roman” font size 10. You can attach additional pages as appendix.
DUE DATE: May 24

Demonstration:
Students will demonstrate the developed architecture in the lab for all other teams in the course.
DUE DATE: May 24-25

Presentation:
All team will have 15 minutes presentation followed by 10 minutes questions. Each team member must present a part of the research and will be evaluated based upon his/her performance presenting the project and answering the questions.
DUE DATE: May 25