CS224 : Assignment 5

Date of Demonstration 11th April 2023, Lab Timing (2PM-5PM), 18% (12%+ 6%) weight
(Part of Assignment 5 will be used in Assignment 6)

==

Part I: Design of Cache using HDL and demonstration using Simulation

Design and implement cache with 256 sets, 4-way set-associative, 64B line/block size. Assume the physical/memory address is 32-bit. In addressing the cache, we can divide the address into three parts (a) Tag, (b) Index, and (c) Offset bits and for this case, it is 18, 8, and 6 bits respectively. You do not need the store actual data to be placed in the cache but simulate the cache structure. Your design should support

- Tag comparison and block access in the cache happened in parallel for all the blocks of the selected index,
- LRU replacement policy for cache block replacement and
- Cache miss under a cache miss using Miss Status Holding Register (MSHR).

Assume the cache hit happens in two or three cycles, but getting a block from memory (DRAM) takes 400 cycles. Once there is a miss in the cache and if the victim block (to e replaced) is dirty (write happened for that block), you need to write back the victim block to memory (DRAM). Assume DRAM have enough capacity and for both read block and write block takes 400 cycles time.

Cache miss under a cache miss means while serving a cache miss, another cache miss can happen. The processor can send another cache request while other cache access is in progress.

Simulate your design using input values and find out the resource usage of your design. *In this case, you do not need to download and show it on the FPGA. Write a test bench for the same to validate your design using simulation. Trace files can be generated from the given C++ Code and may be used in test bench designing. Assume all the access is independent access so the processor can send another cache request while other cache access is in progress.*

Ref: How to Implement Cache, LRU replacement policy, and Miss Status Holding Register (MSHR) from supplied material.

Part II (VHDL and FPGA) : Interfacing Cache with PC serial port and demonstration on FPGA

Interface the Cache with UART and demonstrate the working of the Cache using PC communication. The trace file needs to be read in PC line by line (using a C++/Python Program) and sent to FPGA for emulation. Trace file can be generated from the given C++ code.

For Part II, you need not simulate your design but download the bit file to FPGA and show the working demo.

*Calculate AMAT [Average Memory Access Time=Hit time + Miss Penality * Miss Rate] for your designed cache using the given trace of access.*

https://www.instructables.com/Arduino-Python-Communication-via-USB/
https://pypi.org/project/pyserial/

Evaluation Procedure

- All the member of the group need to be present at the time of Demonstration of the assignment. All the absent members will be awarded 0 marks for the assignment. Please show your ID card at the time of demonstration (as it is difficult to remember faces of all the 128 students of your batch).
- For HDL codes the quality will be based on FPGA minimum resource utilization (Synthesis Report: optimized number of LUTs, register, Minimum Clock), coding style (Use of proper Comment/Naming/Labeling of the wires), performance, comments, and questionnaire and explanation