CS223 : Assignment 6, Date of Demonstration 6th April 2018, Lab Timing (2PM-5PM), 12% weight

(Deadline for submission of assignment by sending zip file (of HDL codes, UCF/contrints file, C files, test bench file) to asahu<@>iitg.ernet.in is 4th April 2018).

Please ensure that you do not submit plagiarized code from your peer group or indirectly from any other website. If two group copies code from same external website and submit then both groups will get F grade. Result of plagiarism test will be informed after the demonstration.

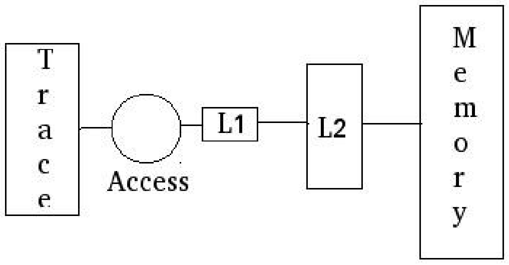
Design and implement two levels cache hierarchy (L1 cache and L2 cache) and test performance of designed cache in term of ***area, delay and power consumption***.

Your design should have dummy memory which can simulate 2^{16} locations 8 bit data. In a cache block it can store line size (block size) amount of data in byte.

Number of cache blocks in a cache = (Number of locations)* (Associativity)
Size of a cache in byte = (Number of cache blocks in the cache)*(Block Size)

Send N (assume N=1024) numbers of 16 bit data (will be used as 16 bit address) to FPGA and store in a (RAM called TraceRAM) Block RAM. After sending all the address to FPGA TRACE memory, every cycle the processor (an entity for namesake) fetch one address from TRACE and try to access the cache hierarchy.

Display number of hit and miss in FPGA LEDs, get the number hit/miss from FPGA to PC. Also match the number of hit (miss) with provided C-code for cache (Hint 2). Use least recently used (LRU) replacement policy for cache block replacement. LRU can be implemented using MoveToFront data structure or using LRU register along with a access counter per index.



Calculation of area, delay and power:

- Area can be calculated the amount of resource used (Number of LUT, FF, I/O and Block Ram).
- Delay is amount time it takes to execute an access (L1 access delay, L2 access separately). Calculate average memory access time (AMAT) for your design (for 1024 request, some will be miss and some will be hit in L1 and L2. Assume L2 miss time is 1000ns (add a delay circuits).
    - AMAT= (L1 hit time)+(L1 Miss rate)*(L1 Miss Penalty); L1 Miss Penalty=(L2 hit time)+ (L2 miss rate)*(L2 miss Penalty); L2 miss penalty=1000ns; Miss rate = Number Miss/Total Access
- Power also can be calculated by power analyzer of Vivado.

ISE/Vivado tools by default provide resource usage report, but for delay, we need to set some some parameter at the time of synthesis and simulation to get that report.

Good design have less area, delay and power consumption. (Grade will be based on quality of design)

Most of the FPGA have 200KB of BRAM, use configuration L1(NumSet=32, Asso=4, LS=16) and L2(NS=128,Asso=8, LS=16). Dummy memory have only one location but can simulate 2^{16} locations. Trace memory have 2KB size, 1024 location each location will have 16 bit address.

If you want, you can go for standalone version, where you initialized all the locations of the TraceMemory with some values in VHDL source code and show the final hit/miss in LED display. Standalone version get 60% of marks. But it is recommendable to go for full version, where you transfer the trace file to TraceMemory and do the cache simulations in FPGA.

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 90 students of your batch).

Hints:

- Chapter 5, Section 5.2 of Computer Organization and Design by Hennessy and Paterson Book.
- A preliminary C code for cache simulation is available @ http://jatinga.iitg.ernet.in/~asahu/cs223/cache.cpp You may modify this code to support LRU replacement policy and actual data block transfer.
- As you need to send 1024, 16 bt address to FPGA, you are required to use C/python code to send data automatically
  LibUSB FPGA; for PC to FPGA communication using C/Python code
  http://jatinga.iitg.ernet.in/~asahu/cs223-2017/readme-fpgalinl.pdf
- Block Ram infereable code ==> http://jatinga.iitg.ernet.in/~asahu/cs223-2017/
- Cache Reference : http://www.ntu.edu.sg/home/smitha/ParaCache/Paracache/sa4.html
- http://www.cs.fsu.edu/~hawkes/cda3101lects/chap7/F7.19.html
- https://userpages.umbc.edu/~squire/cs411_l22.html