**Project:**

This is a 3-level memory hierarchy with interacting L1, L2 caches and memory.

Our implementation contains the following major components:

Processor – To service the input requests from the text file.

L1 Controller – To process the requests from Processor and check if the instruction is a hit in L1, give the output back to processor.

L1 Data – It has the data pertaining to L1 cache. L1 has victim cache and write buffer.

L2 Controller – To process the requests from Processor, L1 Controller, service them and fetch data from memory, return it to L1 controller and Processor.

L2 Data – It has the data pertaining to L2 cache and write buffer.

Memory – Each block in memory contains 32 bytes. We pre-loaded each block initially and a request is always hit in the memory.

Software Used:

Eclipse IDE

Instructions to execute the source files:

The instructions are given to the processor in MainClass.java file. Output is generated in another file, can be viewed by refreshing the project in explorer.

The instructions should be given in the following format for read (separated with spaces):

Read <Memory Address> <No of bytes to be read>

For write:

Write <Memory Address> <Data to be written>

Example:

Input: Read 130 4

Output:

P to L1C

L1C to L2C

L2C to M

M to L2C

L2C to L1C

L2C to L1C

L1C to P

L1C to L1D

In the zip folder we have test cases and their results attached.