CS204 - Computer Architecture
Course Project - Phase 3 Description

Appending RISC-V pipeline with Caches
Date: 21st Apr 2021
Deadline: 6th May 2021. 11.55PM.

Phase 3: Appending a cache like memory module to Phase 2.

In this phase, you'll create the instruction and data cache modules. Instead of directly reading from .mc file (or some other temporary structure), you'll read the instructions, read/write data from these cache modules.

More details of Cache model:

1. Input parameters: Cache size, Cache block size, Direct mapped (DM)/Full Assoc (FA)/ Set Assoc(SA), Number of ways for SA.

2. Output: Number of accesses, number of hits, number of misses, Number of cold, conflict, capacity misses.

3. Implement LRU as the replacement policy for the caches with configuration of either FA or SA.

4. You need not implement pipeline stalls upon a cache miss. Meaning you are effectively modeling the functionality of a Cache, that will not have any impact on the pipeline proceedings. Your cache model is more like an accountant tracking the accesses, hits, misses.

5. You'll need to instantiate two caches, one will work as Instruction cache (I$) and another will work as Data cache (D$). So, all the requests from Fetch stage of your pipeline will be handled by I$. Like wise, requests from Memory stage will be handled by D$.

6. You'll give a provision to the user to specify input configurations for I$ and D$ separately.

7. At the end of simulation, two sets of stats will be printed - one for I$ and another for D$.

GUI requirements for bonus points (1%):
1. You would require to show the user the content of all the sets of the cache (both I$ and D$) which have non-zero data.
2. For each Fetch, Load, Store, show the set that is accessed.
3. Upon a miss, show the victim block.
4. The number of accesses, hits, misses (cold, conflict and capacity) are required to be shown.

Revert if you have any questions.