

Homework2 from Yukai Jin, yj25b

Implementation Explanation

LRU Policy is applied line-based, n-way-wised:

- Each line(index) will have a LRU module attached (not each offset).
- The LRU module is used to select which way (for the line) to store when “wipe-out” happens (the least nth-way used will be wiped for that offset).
- If any offset in a line is used, the LRU module will update that line.
- Each LRU module (per line) is implemented as a two-way linked-list.
- When a data is used(either store or load), it will be moved to “the most” head.

Cache policy is applied specifically:

- When a read-miss happens, we assume there is a memory loading and data will be stored into the line(index).

Cache is implemented specifically:

- Direct-mapping uses 1-way associated cache
- Fully Associated cache has only 1 line and n-ways ($n = \text{CacheSizeInB} / \text{BlockSize}$), thus the index bit is 0 bit.
- Cache is implemented as np.ndarray for convenience.
- When there is a comparison on a line, all n-way data will be grabbed to match the tag.

Victim cache is designed:

- Victim is a fully associated cache which has LRU module.
- Wiped-out data from cache can be stored on victim cache (based on LRU policy)
- If there is a victim cache hit, data will not be restored from victim to cache (as victim access time is at the same level as the L1 cache).

Code explanation:

- main.py -> main entrance with argument parser
- factory.py -> component construction functions(decoder, cache, victim)
- decoder.py -> decode the instruction, calculate tag, index, offset bits dynamically, generate Actions

- actions.py -> contains LoadAction and StoreAction implementation. Cache load/store operations are organized here.
- cache.py -> contains LineDataWayCache and VictimCache
 - LineDataWayCache is implemented as a 3d ndarray with size (index, offset, ways)
 - VictimCache is a special LineDataWayCache, set to be a no-offset fully associated cache by designed.
- lru_module.py -> functions of LRU policy
 - LRULine is the basic two-way linkedlist to calculate the “most” and the “least” used node.
 - LRULine_One is a special one node LRULine.
 - NWayLRU contains $2^{\text{bits_index}}$ of LRULine.

Result report

report.txt contains execution detail.

Without victim

Trace	Direct	2-way	4-way	Fully Assoc.
gcc-10K	10.81%	10.80%	10.80%	10.80%
gcc-1M	1.36%	1.26%	1.22%	1.19%

With victim = 4

Trace	Direct	2-way	4-way	Fully Assoc.
gcc-10K	9.14%	10.54%	10.80%	10.80%
gcc-1M	0.91%	0.67%	0.76%	1.19%