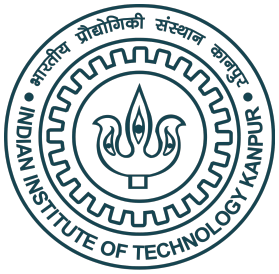

CS422 Computer Architecture

Assignment 4

Prof. Mainak Chaudhuri

November 15, 2023

Jaya Gupta

Roll No: 200471

Contents

1 PART A: TWO-LEVEL INCLUSIVE CACHE HIERARCHY

Table 1: Replacement Policy: L1(LRU) and L2(LRU)

Benchmarks Applications	L1 Accesses	L2 Accesses	L1 Misses	L2 Misses	L2 dead-on-fill Blocks	L2 (atleast 2 hits / atleast 1 hit)
perlbench	561960452	783650	783650	16631	6.32554	0
bzip2	683880209	8430520	8430520	4459981	67.78	51.5592
gcc	509201871	6972804	6972804	419804	48.9931	77.6662
mcf	525252868	68210537	68210537	33529013	61.0062	18.0518
soplex	500168531	19194123	19194123	18513396	98.0233	22.1718
hmmer	623357483	3273742	3273742	1619595	95.378	28.7859
omnetpp	562399833	13646192	13646192	10195677	81.163	36.3174
xalancbmk	544341624	16522138	16522138	2425368	50.9911	63.0186

2 PART B: SRRIP POLICY FOR THE L2 CACHE

Benchmarks Applications	L1 Misses	L2 Misses
perlbench	783650	16630
bzip2	8432216	4510785
gcc	6972908	307228
mcf	68212289	34431975
soplex	19190278	18463056
hmmer	3267507	1557406
omnetpp	13643563	10177238
xalancbmk	16521332	2310414

Table 2: Replacement Policy: L1(LRU) and L2(SRRIP)

3 PART C: NRU POLICY FOR THE L2 CACHE

Benchmarks Applications	L1 Misses	L2 Misses
perlbench	783656	16647
bzip2	8430845	4480087
gcc	6972871	366800
mcf	68211976	33752929
soplex	19194696	18518347
hmmer	3273854	1602191
omnetpp	13647533	10209427
xalancbmk	16523507	2428362

Table 3: Replacement Policy: L1(LRU) and L2(NRU)

4 OBSERVATIONS

- SRRIP and NRU policies results may vary depending on the empty way which is chosen when the block is filled up. This is because when choosing the block for eviction, the

block with the lowest way ID that fits the eviction criteria is evicted. LRU on the other hand, is independent of the way chosen. The above stats are collected when the lowest empty way ID was chosen for FILL.

- L1 miss count changes when changing the policy of L2 cache. This is because,
  - Due to the inclusion policy if a block gets evicted from L2, it is also invalidated in the L1 cache.
  - Hence it could happen, that a hot block in L1 is LRU in L2, and hence ends up getting evicted both in L1 and L2.
  - Different replacement policies such as LRU maintain the newly filled block in the cache until it reaches from MRU to the LRU end which can take significant time depending on the associativity.
  - On the other hand, SRRIP ends up evicting a block when its age has been reached, and hence a block takes 3 (in case max(age) = 3) steps to reach its maximum age independent of associativity.
  - Hence, it could happen that SRRIP ends up evicting a useful block faster than LRU, which will increase the number of L1/L2 misses in the case of SRRIP than LRU.
- SRRIP policy for L2 caches will perform better on those programs whose L2 dead-on-fill blocks percentage on L1(LRU)-L2(LRU) configuration is much higher(reaching 100%). This is because LRU policy will maintain those dead useless blocks for a longer time and SRRIP will evict them much faster making place for useful blocks. Hence in this case, SRRIP L1/L2 miss count will be better than that of LRU.
  - This can be seen in the case of Perlbench, where the L2 dead-on-fill percentage was just 6%, hence both SRRIP and LRU had similar performance.
  - On the other hand, in the case of soplex, the percentage is 98%, and hence the miss count in the case of SRRIP is much smaller for L2 and L1.
- NRU on the other hand, is a much downgraded version of LRU policy with only a one-bit reference counter. Hence it will perform worse than LRU in most cases.