Zexi Pan 1000138246
Xiaoyang Guo 999578513

Lab 4 Report

Q1)

In order to verify the correctness of the next-line prefetcher, we used a microbenchmark that consists of accessing an array of chars with different steps. The cache config we used was a 64 set, 64B block, 4 way L1 cache. Since char's are 1 byte and the cache line is 64B, we expect that accessing the array in steps less than 64 elements have close to 0% miss rate. When the steps are greater than 64, the access can start skipping over fetched elements in the cache line and the prefetched block. We expect having a steps over 128 to have a very high miss rate. The measured miss rate of the two situations are shown below.

Step	30	129
L1 Miss Rate	0.06%	67.25%

Q2)

In order to verify the correctness of the stride prefetcher, we used a microbenchmark that consisted of accessing an array of chars with different steps/strides and randomly. The cache config we used was a 64 set, 64B block, 4 way L1 cache. We expect that with the stride prefetcher that with any stride, the miss rate is close to 0%. We expect that randomly accessing the array would have a higher miss rate.

Step	30	129	Random
L1 Miss Rate	0.08%	0.29%	5.41%

Q3)

Avg Access Time =

$$T_{L1-Hit} * L1\ Hit\ Rate\ +\ L1\ Miss\ Rate * (T_{L2-Hit} * L2\ Hit\ Rate\ +\ T_{Mem\ Hit} * L2\ Miss\ Rate)$$

Config	**L1 Miss Rate**	**L2 Miss Rate**	**Avg Access Time**
No Prefetch	4.16%	11.4%	1.80
Next Line	4.19%	8.38%	1.69
Stride	3.85%	5.78%	1.55

Zexi Pan 1000138246
Xiaoyang Guo 999578513

Going from no prefetch configuration to the next line configuration, L1 miss rate does not increase by much, however L2 miss rate decreased by 3.02%. Similarly when going from the next line prefetch to the stride prefetcher, L2 miss rate decreased by 2.6%. Even though these percentages are small (in the tenths of thousandths for L1), they still make a create significant difference as the hit rate access time is an order of magnitude smaller.

Q4)

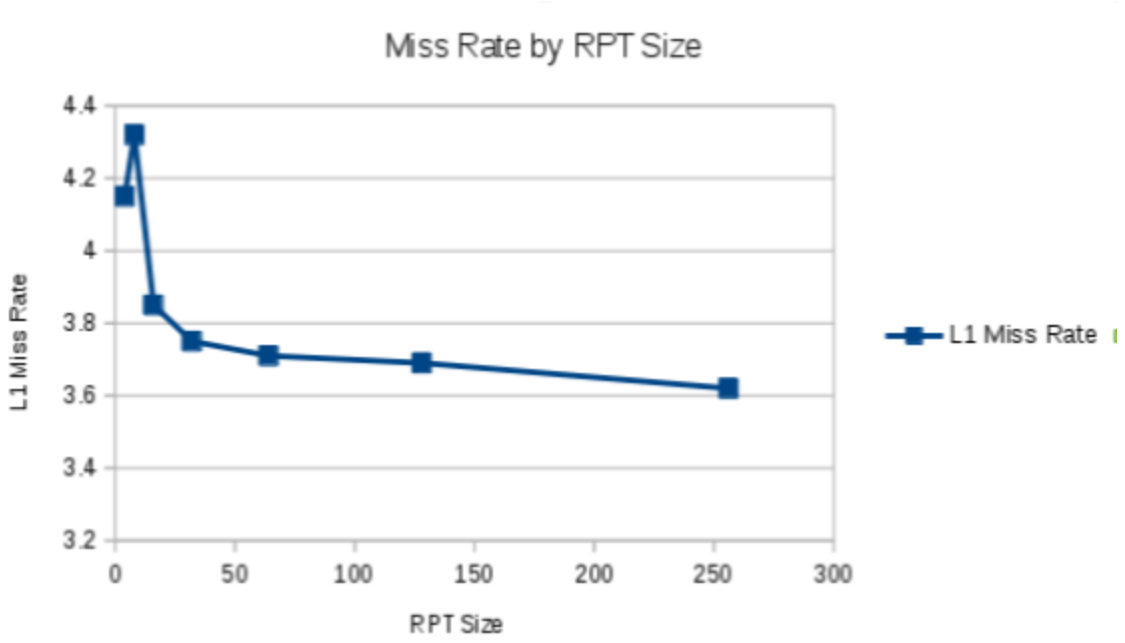

Figure 1: Miss Rate vs RPT Size

Size	4	8	16	32	64	128	256
L1 Miss Rate	4.15%	4.32%	3.85%	3.75%	3.71%	3.69%	3.62%

The L1 miss rate decreases as the RPT size increases. When the number of RPT entries approaches the size needed to fit the working set, the L1 miss rate tapers off and being to approach some constant value. Any increase to the size after it can fit the working set, will no longer improve the L1 miss rate for a given benchmark.

Zexi Pan 1000138246
Xiaoyang Guo 999578513

Q5)

More statistics we would like to include are:

- the number of cache blocks evicted that were used
    - Within a set amount of cycle afterwards
- the number of prefetched blocks that were useful
    - How often were the prefetched blocks used before being evicted
    - How many prefetched blocks were not used before being evicted

These statistics would be helpful in determining which prefetcher to use. It would also help determine the prefetch degree and prefetch timings needed to increase the amount of relevant blocks are fetched so they arrive within the cache at the correct time to be used.

Q6)

Open-end Prefetcher:

The open-end prefetcher we designed is a combination of the Stride Prefetcher and the GHB Delta Correlation Prefetcher. Through testing we found out that the performance of the Stride Prefetcher drops is the stride is not stable. So a variant of GHB Delta Correlation Prefetcher is used to compensate if the Stride Prefetcher is not is steady state. The correlation prefetcher consists of a Global History Buffer, an Index table and a delta buffer. GHB stores all the miss prefetched addresses, each address points to certain entry in the Index table. Addresses pointing to the same entry are correlated, and the delta of the correlated addresses are stored in the delta buffer as patterns. Once a pattern is found, prefetches can be made.

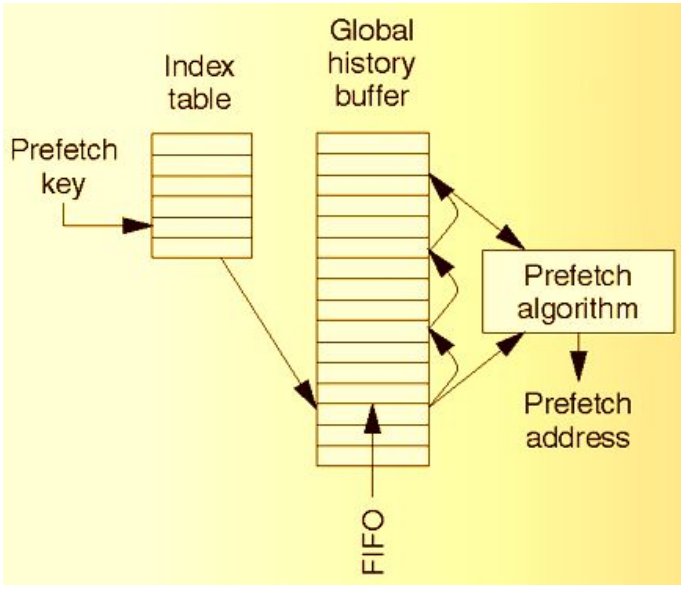

Figure 2: GHB Delta Correlation Prefetcher Configuration [1]

Zexi Pan 1000138246
Xiaoyang Guo 999578513

Open-end Prefetcher Performance against given benchmarks:

	compress.eio	gcc.eio	go.eio	average
L1 miss rate	3.69%	1.23%	1.13%	2.017%

Open-end Prefetcher Performance against mbq6:

	Constant Stride (30)	Pattern	Random Access
L1 miss rate	0.14%	1.02%	5.33%

Feasibility:
Size = RPT_size + GHB_size + IndexTable_size + compariter_size
= 16*32*3+128*(32+32)+ 64*(16+32) + 4*32 = 1068 Bytes

CACTI results:
Tag side (with Output driver) (ns): 0.238523
Tag array: Total dynamic read energy/access (nJ): 0.00107858
Total leakage read/write power of a bank (mW): 1.28268
Tag array: Area (mm2): 0.00526918
Height (mm): 0.103549
Width (mm): 0.0508859

The design is feasible considering the access time allows for a clock rate of 4GHz and the area overhead is minimal compared to the size of the cache.

Work Distribution:

Zexi Pan	Xiaoyang Guo
Open-end Prefetcher	Open-end Prefetcher
Stride Prefetcher	Next-line Prefetcher
CACTI	Microbenchmark
Lab Report	Lab Report

Reference:
[1] *Data cache prefetching using a global history buffer*, 1st ed. Estados Unidos: The Institute of Electrical and Electronics Engineers, Inc-IEEE, 2005.