CS 610 Semester 2020–2021-I: Assignment 1

Hirak Mondal

Roll Number - 20111022
hirakm20@iitk.ac.in

Solutions

Problem 1

Cache size=256 KB

Array length=32k

Size of each word=8B

Array size=$32k * 8B = 256\ KB$

So see that the entire array can be fitted in the cache.

Cache Miss when access stride = 1

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[1] to A[3] results in hit.

So there is 1 miss for every 4 word access. So in the first iteration of the outer loop and complete iteration of the inner loop we will have total of –

$\frac{32K}{4} = 8K$ misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are $8K$ for an access stride of 1

Cache Miss when access stride = 4

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[4] which results in a miss, so for every iteration of this inner loop we will encounter miss only, as when A[i] is accessed A[i+1] to A[i+3] is fetched but not A[i+4]. So total miss –

$\frac{32K}{4} = 8K$ misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are $8K$ for an access stride of 4

<u>Cache Miss when access stride = 16</u>

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[16] which results in a miss and its adjacent 3 words are fetched A[17] to A[19], so for every iteration of this inner loop we will encounter miss only since the stride is greater than or equal to 4. So total miss –

$\frac{32K}{16} = 2K$ misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are $2K$ for an access stride of 16

<u>Cache Miss when access stride = 32</u>

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[32] which results in a miss and its adjacent 3 words are fetched A[33] to A[35], so for every iteration of this inner loop we will encounter miss only since the stride is larger than 4. So total miss –

$\frac{32K}{32} = 1K$ misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are $1K$ for an access stride of 32

Cache Miss when access stride = 2K

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[2K] which results in a miss and its adjacent 3 words are fetched A[2K+1] to A[2K+3], so for every iteration of this inner loop we will encounter miss only, since the stride is larger than 4. So total miss –

$\frac{32K}{2k}$ = 16 misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are 16 for an access stride of $2K$

Cache Miss when access stride = 8K

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[8K] which results in a miss and its adjacent 3 words are fetched A[8K+1] to A[8K+3], so for every iteration of this inner loop we will encounter miss only since the stride is larger than 4. So total miss –

$\frac{32K}{8K}$ = 4 misses, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are 4 for an access stride of $8k$

Cache Miss when access stride = 32K

During i=0 the word is missed, after which the entire block is brought inside the cache, so A[0] to A[3] is inside the cache.

Next access is A[32K] which results in a miss and its adjacent 3 words are fetched A[32K+1] to A[32K+3], so for every iteration of this inner loop we will encounter miss only since the stride is larger than 4. So total miss –

$\frac{32K}{32K} = 1$ miss, in the first iteration of the outer loop.

Now since the cache size is same as the array size it will hold the array completely, so for all the remaining iterations of the outer loop, there will only be all $HITS$.

So total misses are 1 for an access stride of $32K$

TABULAR REPRESENTATION

Strides	Misses
1	8k
4	8k
16	2k
32	1k
2k	16
8k	4
32k	1

Table 1

PROBLEM 2

Array size= $512 * 512 = 2^{19} * 2^{19} = 256K$ words

And the array can store $32K$ words

So only $\frac{32K}{256K} = \frac{1}{8}$ part of the array can only be stored inside the cache. This means that element (i,j) and (i+64,j) will map to same cache block, given the cache is direct mapped. Thus, if the inner loop accesses an array by column, by the time $\frac{1}{4}th$ of the column has been accessed, the elements in the first quarter column would have been removed from the cache due to conflict misses.

ikj FORM ANALYSIS

FOR A

A has no relation with the loop j, so no need to take it in consideration. Our interest lies with loop k and i. See that k is controlling the column access of A. So for A[i][0] (assuming k=0) we are having a cold miss and at that time we are fetching the entire block along with us. Which leads to Miss of N/B, for the k loop where B is the block size, here B is 8 words. Since the entire array cannot be fitted within the cache so the same thing repeats N times for the outer i loop. So the total misses leads to $N * \frac{N}{B} = \frac{N^2}{B} = \frac{2^{18}}{2^3} = 2^{15}$ misses

<u>FOR B</u>

B has dependency on all the three loop variants. For the innermost loop it is accessing row wise that's why it has the spatial reuse, so its miss is $\frac{N}{B}$, for the innermost loop. But the cache is not so large to hold the entire array for the temporal reuse, so for k it is N times the same of innermost loop and for i loop it is same as N times the two inner loops. So total misses are, $N * N * \frac{N}{B} = \frac{N^3}{B} = \frac{2^{27}}{2^3} = 2^{24}$ misses

<u>FOR C</u>

When j is varied we take the advantage of spatial locality and we get N/B misses as a result. For k-th loop all is hit as we are accessing the same row again and again, For outermost i-th loop we will repeat the misses of inner j-th loop N times. So in total we have, $N * \frac{N}{B} = \frac{N^2}{B} = \frac{2^{18}}{2^3} = 2^{15}$ misses.

<u>TABULAR REPRESENTATION (DIRECT MAPPING)</u>

LOOP	A	B	C
i	N	N	N
k	N/B	N	1
j	1	N/B	N/B
Total	N^2/B	N^3/B	N^2/B
Total(Numerical Value)	2^{15}	2^{24}	2^{15}

Direct Mapping

<u>TABULAR REPRESENTATION (FULLY ASSOCIATIVE MAPPING)</u>

LOOP	A	B	C
i	N	N	N
k	N/B	N	1
j	1	N/B	N/B
Total	N^2/B	N^3/B	N^2/B
Total(Numerical Value)	2^{15}	2^{24}	2^{15}

Fully Assosiative Mapping

jik FORM ANALYSIS

FOR A

For the innermost loop 'k', we are accessing by row, so there will be 1 miss per block, since we are taking the advantage of spatial reuse. So the miss for $k-th$ loop leads to $\frac{N}{B}$. Now for the $i-th$ we are not able to use the temporal locality since all the previous blocks have been replaced, so we will encounter the same misses that of $k-th$ loop n times, and the similar case is also for the outermost loop J. Thus we have a total misses $N*N*\frac{N}{B} = \frac{N^3}{B} = \frac{2^{27}}{2^3} = 2^{24}$ misses

FOR B

Lets analyze B for Direct Memory Access first. In direct memory access, the temporal and spatial locality cannot be utilised for the innermost loop k. SO miss for the innermost loop is N, which repeats $N*N$ times for the two outer loops So total misses (for Direct Mapping) are, $N*N*N = N^3 = 2^{27}$ misses.

Now lets see the case of Fully Associative Mapping. In Fully Associative Mapping, we will not replace a block when there is a space in the cache. So all 512 words accessed for the first time by the innermost loop results in a miss, so for innermost loop there are N misses. Now for the $i-th$ loop all are hit, since we are just exploiting temporal locality. So $NO\ MISS$ are there for $i-th$ loop. Now for outermost loop of j, we will use the spatial locality since the blocks are entirely fetched during the misses of the innermost loop k, so the total misses for the outermost loop is $\frac{N}{B}$. So the total misses are $N*\frac{N}{B} = \frac{N^2}{B} = \frac{2^{18}}{2^3} = 2^{15}$

FOR C

C has no relation with the innermost loop, so we can ignore it. Now lets analyze the case for C when there is direct mapping.When there is direct mapping, for the $i-th$ loop everything results in a miss, so there are N misses and moreover the blocks gets overwritten due to the direct mapping. So for the outermost loop j, we dont have temporal as well as spatial locality, which again leads to N misses So in total we have, $N*N$ misses for direct mapping.
Now lets look at fully associative mapping,here also the $i-th$ loop will result in N misses but the outer loop j, will use the spatial locality and hence will result in $\frac{N}{B}$ misses. So total we have, $N*\frac{N}{B} = \frac{N^2}{B}$ misses for fully associative mapping.

TABULAR REPRESENTATION (DIRECT MAPPING)

LOOP	A	B	C
j	N	N	N
i	N	N	N
k	N/B	N	1
Total	N^3/B	N^3	N^2
Total(Numerical Value)	2^{24}	2^{27}	2^{18}

Direct Mapping

TABULAR REPRESENTATION (FULLY ASSOCIATIVE MAPPING)

LOOP	A	B	C
j	N	N/B	N/B
i	N	1	N
k	N/B	N	1
Total	N^3/B	N^2/B	N^2/B
Total(Numerical Value)	2^{24}	2^{15}	2^{15}

Fully Assosiative Mapping

PROBLEM 3

Given that cache size is $16MB = 2^{24}$ and Array Size is $2^{24} * 2^3 = 2^{27}B$, so only $\frac{1}{8}$ of the array will fit into the cache, which will lead to conflict misses in case of direct mapping, due to which the entire blocks that will be brought in inside the cache will be replaced.

FOR A

For the innermost loop i we are accessing via column and as a result all N will result in miss, and moreover since it is a direct mapping so, the blocks which are previously fetched in will be replaced. So even for the outer two loops of j and k, the same misses will continue for $N * N$ times, so total number of misses for A is $N * N * N = N^3 = 2^{12^3} = 2^{36}$

FOR X

The innermost loop i has no importance and doesn't lead to any hit or misses. Our main concern is the $j - th$ and $k - th$ loop, see that as the $j - th$ loop changes we are missing the first word of that block and then fetching the entire block which leads to $\frac{N}{B}$ misses, since we are taking advantage of the spatial locality. And the $k-th$ block repeats this N number of times. So total we have $N * \frac{N}{B} = \frac{N^2}{B} = 2^{24}/2^3 = 2^{21}$ misses.

TABULAR REPRESENTATION

LOOP	A	X
k	N	N
j	N	N/B
i	N	1
Total	N^3	N^2/B
Total(Numerical Value)	2^{36}	2^{21}