

## Lab 5

NAME: VIVIAN KERUBO MOSOMI

REGISTRATION NUMBER: SCT212-0062/2021

### Problem

Consider a computer system with a first-level data cache with the following characteristics:

size: 16KBytes; associativity: direct-mapped; line size: 64Bytes; addressing: physical.

The system has a separate instruction cache and you can ignore instruction misses in this

problem. This system is used to run the following code:

```
for (i=0; i<4096; i++)
```

```
  X[i] = X[i] * Y[i] + C
```

Assume that both X and Y have 4096 elements, each consisting of 4 bytes (single precision

floating point). These arrays are allocated consecutively in physical memory. The assembly

code generated by a naive compiler is the following:

```
loop: lw f2, 0(r1) # load X[i]
```

```
lw f4, 0(r2) # load Y[i]
```

```
multd f2, f2, f4 # perform the multiplication
```

```
addd f2, f2, f0 # add C (in f0)
```

```
sw 0(r1), f2 # store the new value of X[i]
```

```
addi r1, r1, 4 # update address of X
```

```
addi r2, r2, 4 # update address of Y
```

```
addi r3, r3, 1 # increment loop counter
```

```
bne r3, 4096, loop # branch back if not done.
```

**a. How many data cache misses will this code generate? Breakdown your answer into the three types of misses.**

i) Cold Miss - Occurs when accessing a memory location for the very first time, and the cache is empty or does not contain the necessary data.

When X[0] is first loaded, it causes a cold miss.

For the number of data cache misses:

Cache size = 16kb

Both arrays have 4096 elements, thus:

Number of blocks for each array =  $4096 / 16 = 256$  blocks

Thus, 256 cold misses for X and 256 cold misses for Y which totals to 512 misses.

ii) Conflict Miss - Occurs when a data item is already in the cache but needs to be replaced because another data item is accessing the same cache line.  
For the above code, for each iteration  $X[i]$  and  $Y[i]$ ,  $Y[i]$  replaces  $X[i]$  that was loaded and also  $X[i]$  replaces  $Y[i]$ . Thus both loads are misses and the store to  $X[i]$  is also a miss

Misses per iteration = 3

Total memory references =  $3 * 4096$  (elements for each array) = 12288

Cold misses were 512.

One access is a compulsory miss or a cold miss, thus (15/16)

Conflict misses =  $(15/16) * 4096$

Conflict misses =  $3840 + 4096$  elements which is 7936

Total number of misses = Cold misses + Conflict misses

Total =  $512 + 7936$

= 8448

To get the miss rate, the total number of misses is divided by 12288:

$((8448 / 12288) * 100) = 68.75\%$

**b. Provide a software solution that significantly reduces the number of data cache misses.**

**How many data cache misses will your code generate? Breakdown the cache misses into the three types of misses. What is the data cache miss rate?**

### **Solution**

Adding a gap between  $X[ ]$  and  $Y[ ]$  so that  $X[i]$  and  $Y[i]$  do not map to the same cache index. Since conflict misses are caused by data items mapping to the same cache location, thus when one item i.e X is loaded, another item i.e Y must be evicted.

Adding a gap will prevent the mapping conflict hence conflict misses will not occur as the data items X and Y do not evict each other.

Cache misses generated: A

Cold misses - Will occur because the two data items, X and Y access the cache for the first time.

The cache size is 16kb and both arrays have 4096 elements. So each array takes 256 blocks, thus 256 misses. Addition of the two results to 512 misses.

There will be no capacity misses because they occur when the cache is too small to hold all of the data. For this scenario, the cache is directly mapped leading to conflict misses, where all the misses are due to the conflicts and the cache's capacity limit is not reaching maximum (16kb).

**The cache miss rate:**

The code:

```
for (i=0; i<4096; i++)  
X[i] = X[i] * Y[i] + C
```

There are 3 memory access which arise from:

Loading X[i], Loading Y[i] and Storing X[i] and each have 4096 elements. Thus,  
Total Memory References =  $3 * 4096 = 12288$

Miss Rate = (Misses / Memory References) \* 100

Misses = Cold misses which are 512

Miss Rate =  $((512 / 12288) * 100) = 4.167\%$

Miss Rate = 4.167%.

**c. Provide a hardware solution that significantly reduces the number of data cache misses. You are free to alter the cache organization and/or the processor. How many data cache misses will your code generate? Breakdown the cache misses into the three types of misses.**

**What is the data cache miss rate?**

**Solution**

Increasing the capacity of the cache. The cache has a 16kb size, and the data items X, Y have 4096 elements consisting of 4 bytes each. Thus:

X :  $(4096 * 4) / 1024 = 16\text{kb}$

Y :  $(4096 * 4) / 1024 = 16\text{kb}$

Both X and Y use 32 kb while the cache has 16kb. This therefore means that capacity misses will occur cause the cache does not have enough space to hold the current data leading to evictions.

Increasing the cache size will enable both arrays to fit entirely.

If the cache size is increased, X[i] and Y[i] map to different indices in the larger cache hence no conflict misses.

At this point, on increasing the cache size, there are no capacity and conflict misses that will occur. For a cold miss, the data items access the cache for the first time

Each array has 4096 elements and the cache size is 16kb, thus;

$4096/16 = 256$  blocks translating to 256 misses. On addition of the two, there are 512 misses.

Memory references :  $3 * 4096 = 12288$

Miss rate = (Number of misses / Memory references) \* 100

Miss rate :  $(512/12288) * 100$

Therefore miss rate = 4.167%