Assignment 8 on Cache Memory : Report

CO LAB: CS-201P

May 14, 2018

Vishal Anand B16040

We are given a address.txt file in which around $\mathbf{10^6}$ addresses are given in 8 characters Hexadecimal format. Since for each place **16** digits can be stored, we can assume the Main Memory ranging from **0000 0000** to **FFFF FFFF**. Therefore, total number of possible addresses are $\mathbf{16^8}$ which is equal to $\mathbf{2^{32}}$. The cache memory and hence the Main Memory has a **16B** block size. The results for **number of hits** in Direct mapping, Set-Associative (FIFO) and Set-Associative (LRU) mapping are given below:

Cache Size (Bytes)	Direct Mapping	Set-Associative (FIFO)	Set-Associative (LRU)
1024	0	0	0
2048	0	2	2
4096	1	2	2
8192	1	2	2
16384	3	2	2

Few observations can be drawn from this table are:

- For a given cache block size, on increasing the cache size, the total number of blocks / sets in the cache memory increases and hence the total number of block from the main memory fighting for the same block / set in the cache memory decreases. This reduces the chance of collision / replacement therefore decreasing the miss rates and increasing the hit rates.
- Different mapping functions can have different miss / hit rates depending upon the addresses we want to read / write and its order also matters. Therefore, for general purpose, we can never compare any mapping function and its efficiency in terms of hit rates.
- In Direct mapping function, each block in the main memory has a fixed position in the cache memory, therefore needs constant time to access the address in the cache memory. But this is not flexible. When we R/W address which has all the same position in the cache memory, the misses would be very high.
- Contrary to this, in Associative mapping any block from the Main memory can occupy any empty place in the cache memory. Therefore this is very flexible, but searching time is Linear which is very slow.
- In between these two comes Set-Associative in which a **K** number of blocks form a set. Each block from the main memory has a fixed set number but can go anywhere among different blocks in that particular set.
- Therefore it can be concluded that the graph of **No. of hits** vs **Cache size** forms a **non-decreasing function**.

For eg. When Cache size is **8192B**, $\text{Hits}_{\text{direct}} < \text{Hits}_{\text{set-associative}}$, the addresses might be R/W in the order **ACDDC** (Fig. 1)

When Cache size is **16384B**, $\text{Hits}_{\text{direct}} > \text{Hits}_{\text{set-associative}}$, the addresses might be R/W in the order **ACBCAC** (Fig. 1)

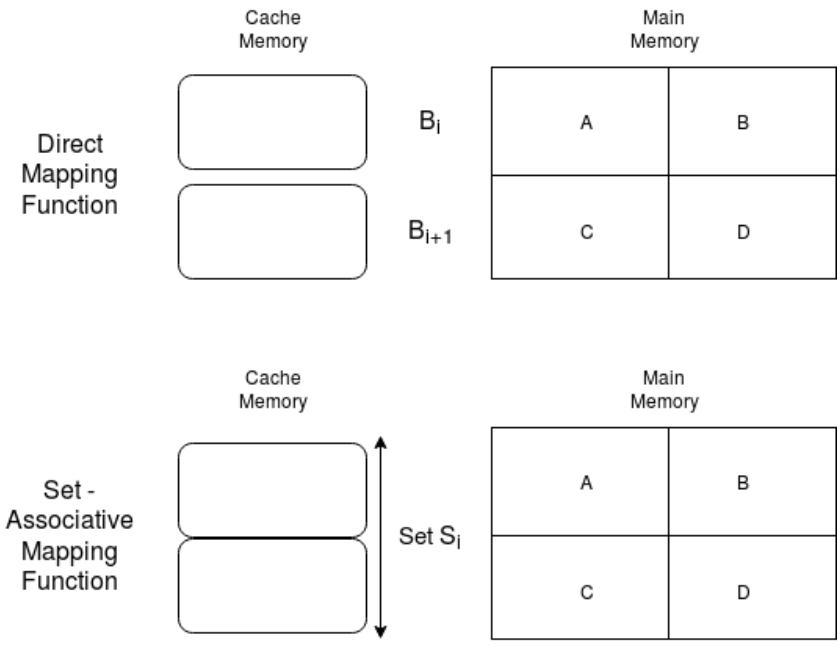

Fig. 1 A schematic of Direct and Set-Associative mapping

Cache information of my machine:

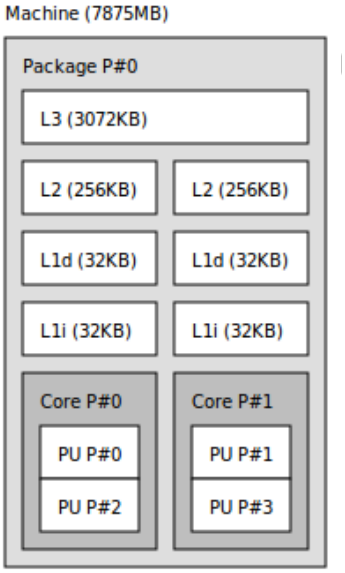

Fig. 2 Cache size of my Machine

Cache Level	L1	L2	L3
Mapping Function	8-way Set-associative	4-way Set-associative	12-way Set-associative