Tutorial 10 – Week of Nov. 22

Questions:

Q1)

Consider an Intel microprocessor with a 16 Kbyte unified L1 cache. The miss rate for this cache is 3% and the hit time is 2 CCs. The processor also has an 8 Mbyte, on-chip L2 cache. 95% of the time, data requests to the L2 cache are found. If data is not found in the L2 cache, a request is made to a 4 Gbyte main memory. The time to service a memory request is 100,000 CCs. On average, it takes 3.5 CCs to process a memory request. How often is data found in main memory? **Assume we also know L1 hit time = 15 CCs and Main hit time = 200 CCs.**

Solution

Average memory access time = Hit Time + (Miss Rate x Miss Penalty)

Average memory access time = Hit Time$_{L1}$ + (Miss Rate$_{L1}$ x Miss Penalty$_{L1}$)

Miss Penalty$_{L1}$ = Hit Time$_{L2}$ + (Miss Rate$_{L2}$ x Miss Penalty$_{L2}$)

Miss Penalty$_{L2}$ = Hit Time$_{Main}$ + (Miss Rate$_{Main}$ x Miss Penalty$_{Main}$)

average memory request = 3.5
L1 hit time = 2
L1 miss rate = 0.03
L2 hit time = 15
L2 miss rate = 1 - 0.95 = 0.05
Main hit time = 200
Main miss rate = X
Main miss penalty = 100000

$\underline{3.5 = 2 + 0.03 (15 + 0.05 (200 + X (100,000)))}$

3.5 = 2 + 0.03 (15 + 10 + 5000X) 3.5 = 2 + 0.03 (25 + 5000X)

3.5 = 2 + 0.75 + 150X 3.5 = 2.75 + 150X

0.75 = 150X

X = .005

Thus, 99.5% of the time, we find the data we are looking for in main memory.

1 word = 4 bytes

assume if the mapped set is not full, store data in empty blocks of that set.

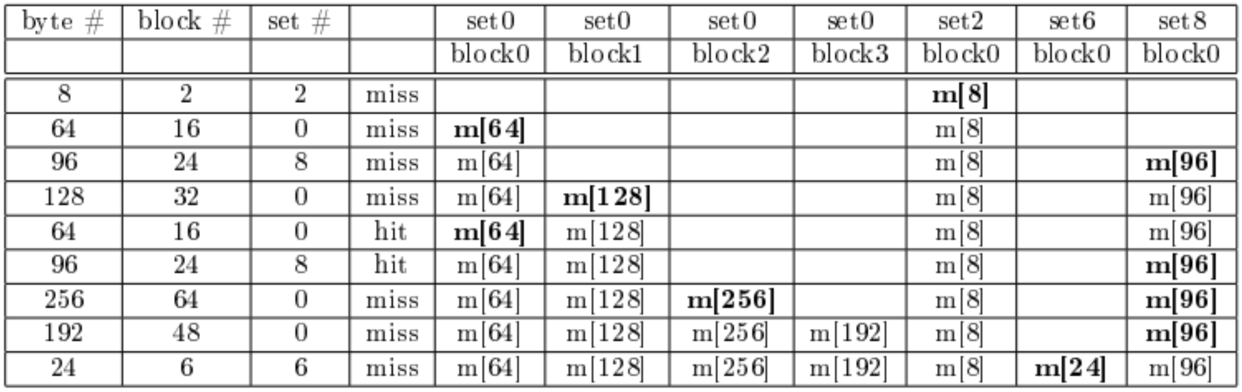
Q2

Suppose a computer has a 4-way set associative cache with one-word blocks. It has a capacity of `256` bytes. Given the sequence of byte addresses `8, 64, 96, 128, 64, 96, 256, 192, 24` show the final cache contents and state the number of hits and misses.

Solution

The block addresses for this sequence are `2, 16, 24, 32, 16, 24, 64, 48, 6`. There are `256/(4*4) = 16` sets. Therefore, the mapping of block address to sets is `block_addr % 16`.

The sequence generates the following table. Empty sets are not shown. The currently accessed cache block is in boldface. The notation `m[i]` means the word located at memory address `i`.



byte #	block #	set #		set0	set0	set0	set0	set2	set6	set8
				block0	block1	block2	block3	block0	block0	block0
8	2	2	miss					**m[8]**		
64	16	0	miss	**m[64]**				m[8]		
96	24	8	miss	m[64]				m[8]		**m[96]**
128	32	0	miss	m[64]	**m[128]**			m[8]		m[96]
64	16	0	hit	**m[64]**	m[128]			m[8]		m[96]
96	24	8	hit	m[64]	m[128]			m[8]		**m[96]**
256	64	0	miss	m[64]	m[128]	**m[256]**		m[8]		**m[96]**
192	48	0	miss	m[64]	m[128]	m[256]	m[192]	m[8]		**m[96]**
24	6	6	miss	m[64]	m[128]	m[256]	m[192]	m[8]	**m[24]**	m[96]

There were 7 misses and 2 hits.