IIT Guwahati - Department of Computer Science & Engineering

CS 223- Computer Organization & Architecture

Practice Tutorial Sheet #3 [21.05.2020] – SOLUTION SET

1. A cache has access time (hit latency) of 10 ns and miss rate of 5%. An optimization was made to reduce the miss rate to 3% but the hit latency was increased to 15 ns. Under what condition this change will result in better performance (Lower AMAT)?
   Solution:
   AMAT 1 = HT1 + MR1 x MP HT1 = 10 ns; MR1=0.05
   AMAT 2 = HT2 + MR2 x MP HT2 = 15 ns; MR2=0.03
   AMAT2<AMAT1
   15 + 0.03 x MP < 10 + 0.05 x MP
   5 <0.02MP $\rightarrow$ MP > 250 ns (Miss penalty should be larger than 250 ns)

2. A cache has hit rate of 95%, block size of 128B, cache hit latency of 5ns. Main memory takes 50 ns to return first word (32 bits) of a block and 10 ns for each subsequent word.
   (a) What is the miss latency of the cache?
   (b) If doubling the cache block size reduces the miss rate to 3%, does it reduce AMAT?
   Solution: Hr = 0.95; BS = 128B; Ht =5 ns ; 1word = 4B (32 bits)
   # words/block = 128B/4B = 32
   (a) MP = 50 + (31x10) = 360 ns
   AMAT1 = 5 + 0.05 x 360 = 23 ns
   (b) # words/ block = 256B/4B = 64
   MP = 50 + 63 x10 =680 ns
   AMAT2 = 5 + 0.03 x 680 = 25.4 ns
   AMAT2> AMAT1; Hence doubling the block size will not reduce AMAT

3. A 16KB direct mapped 256B block unified cache is attached to a 16MB main memory system. The word length as well as instruction length of the processor is 16 bits. Consider a program that consists of a main routine M which in turn calls a subroutine S. M consists of 12 instruction words which are loaded in the main memory from the address 0x4230FA onwards. The last five instructions of M is a loop that is iterated 10 times. The second instruction in the loop is a call to subroutine S. S consists of 4 instruction words loaded in the main memory from the address 0x70F168. The last instruction of S is a subroutine return back to M. The only two data words that are used by M and S are at addresses 0x748074 and 0x846064. Assume the caches are initially empty. Ignore OS level interruption and subsequent cache impact on context switching.
   (a) Find the number of cache misses occurred during the execution of the program.
   (b) How many cache block evictions happened during the execution of the program?
   (c) List out the block numbers (in decimal) in the cache that are non-empty after the execution of the program

   Solution:

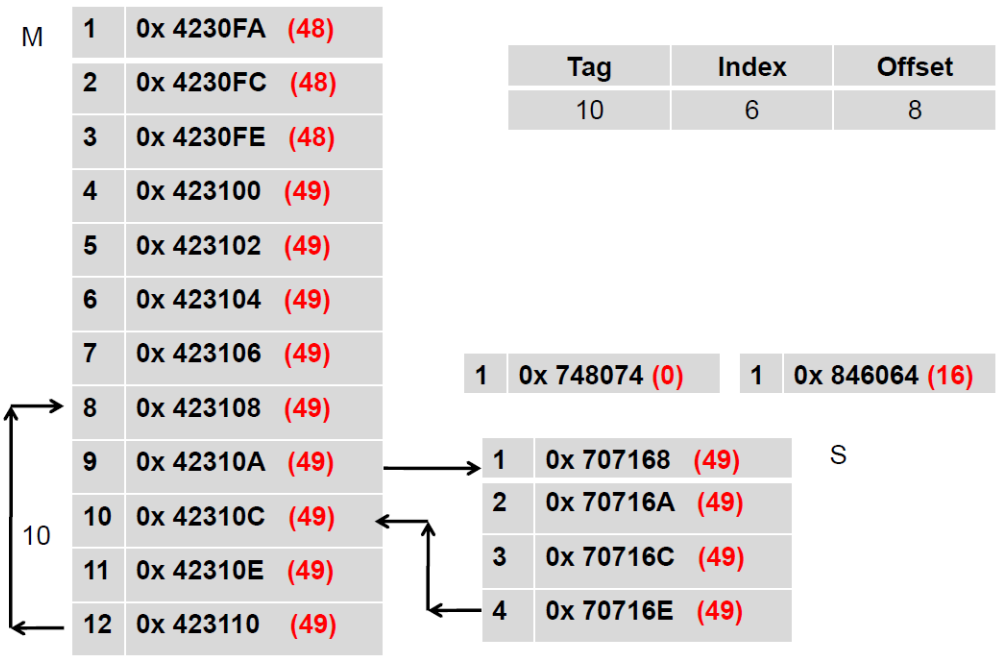
   # sets = CS/(BSxA) = $2^{14}/(2^{8}x1) = 2^{6}$ = 64 sets
   16 MB main memory $\rightarrow$ 2^{24} bytes $\rightarrow$ 24 bit physical address.

Tag	Set Index	Offset
10	6	8

Each instruction is 2bytes. It will occupy 2 consecutive bytes in the given byte addressable memory. The two data words are also 2 byte long like the instruction.

We have to find the set numbers to which the instructions and data a are mapped. Initially the cache is empty. On each miss a cache block of 256B (128 words) are brought. The offset portion will help to understand which word of a given block a word belong to.

Illustrative diagram of M and S, its instructions and flow of control. The numbers marked in red are the set/block number in the cache that instruction/data is mapped to.



(a) Find the number of cache misses occurred during the execution of the program.
Access of following instructions will be miss
M1 // 1st instruction of M
M4 // 4th instruction of M
S1, M10, S1, M10, ….. (10 times) = 20 misses // looping
First access of data on 0x748074 // it will remain there as it is not evicted out of cache
First access of data on 0x84606 // it will remain there as it is not evicted out of cache
Total misses = (2 + 20) instruction misses and 2 data misses→24 misses

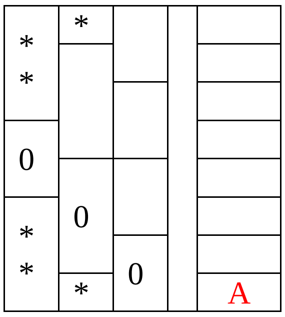
(b) How many cache block evictions happened during the execution of the program?
Miss@S1 evicts the block that carries M4 to M12. Then it loads the block with S1 to S4
Miss@M10 evicts the block that carries S1 to S4. Then it loads the block with M4 to M12
This is repeated 10 times leading to 20 cache block evictions.

(c) List out the block numbers (in decimal) in the cache that are non-empty after the execution of the program.
All blocks except blocks (set number and block number is same here as it is direct mapped cache) 0, 16, 48, 49.

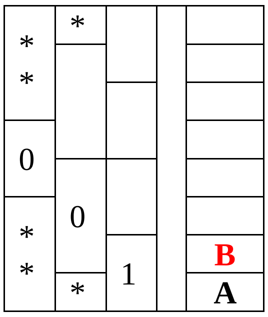
4. Consider an 8-way set associative cache that uses pseudo LRU block replacement policy. Assume all the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way 0 to way-7. Consider the following 14 block numbers all mapped to a particular set n given in the order of arrival. Show the organization of set n (way number & the block number residing in the way) after servicing these requests.
A, B, C, A, D, E, B, F, G, C, H, A, E, Q.

In PLRU tree given below, entry 0 indicates up (pointing to upper sub tree) and 1 indicates down (pointing to lower sub tree) link. Pointer direction indicates LRU portion of that subtree. The entries are after adding the block in its corresponding way.

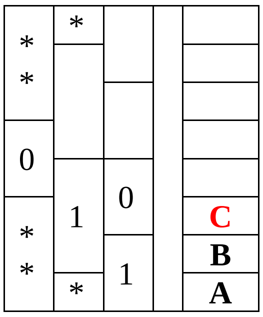
Miss on A



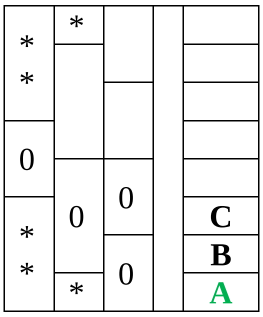
Miss on B



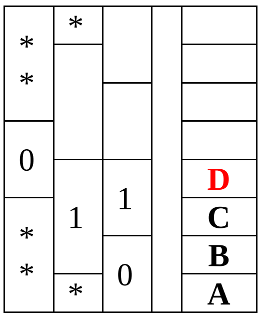
Miss on C



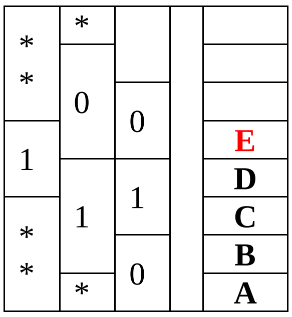
Hit on A



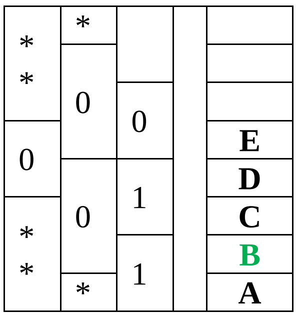
Miss on D



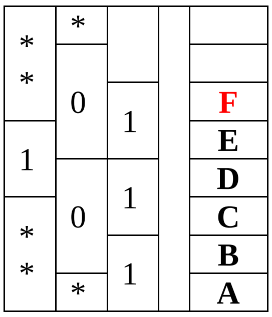
Miss on E



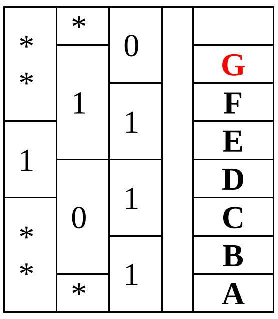
Hit on B



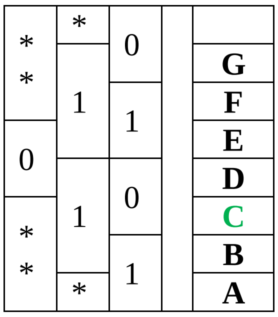
Miss on F



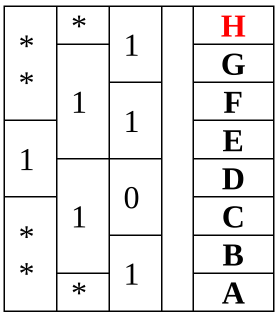
Miss on G



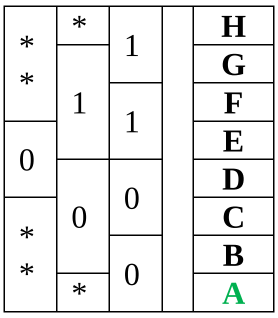
Hit on C



Miss on H

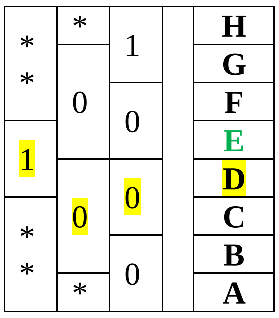


Hit on A



Hit on E

Miss on Q, set n is full. So replacement on PLRU element.

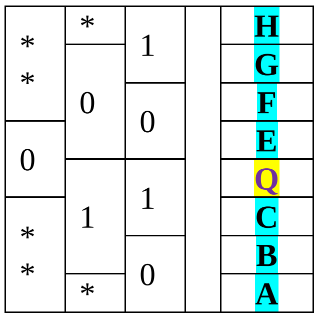


Root @ 1 → bottom child of root.

It is @ 0 →, upper child

It is at @0 → Upper child -- Block D is stored there.

Replace D with Q.



Final organization of blocks in set n

Way-0	Way-1	Way-2	Way-3	Way-4	Way-5	Way-6	Way-7
A	B	B	Q	E	F	G	H

5. There are two variants (alpha and beta) of cache memory organization that is possible in a processor. Alpha is a split cache organization with 16 KB instruction cache and a 16 KB data cache. Both the I and D caches have its own data port. Beta is a 32 KB unified cache organization that has only one cache port, so that a load or store hit takes 1 extra clock cycle at MEM stage due to structural hazard with IF stage. Assume a hit takes 1 clock cycle and the miss penalty is 50 clock cycles. For a program that has 40% load-store instructions, the MPKI of I-cache and D-cache of alpha are 40 and 120, respectively, whereas the MPKI on unified cache of beta is 200. Which one beta or alpha has the higher AMAT? Explain with proper justifications.

Solution: This is a classical case of how structural hazard can happen in cache memory when IF and MEM stage happens together. Sometimes we need to access two different words from memeory at same clock cycle. It can be done if we have split cache anmd each has its own independent port. But in this example unified cache we have only one port

AMAT = Ht + Mr X Mp

Mr = misses / memory access = (misses / instruction) / (memory access / instruction)

Mr = MPI/MAPI

Mr_I-cache=(40/1000)/1= 0.04 & Mr_D-cache=(120/1000)/0.4= 0.3

If we take 100 instructions, then we access I cache 100 times (each instruction has an IF) and D cache 40 times (only 40 of 100 instruction uses MEEM stage for Load/Store operations). So total 140 times we are accessing cache.

% Use of I-cache=100/140 = 71.4% : % Use of D-cache=40/140 = 28.6%

Overall Mr_Aplha = (71.4%x0.04)+ (28.6%x0.3)=0.1145

Mr_Beta=(200/1000)/1.4= 0.143

AMAT = Ht + Mr X Mp (Ht=1 cycle: Mp=50 cycles)

Mr_I-cache= 0.04 : Mr_D-cache= 0.3: Mr_Uni-cache= 0.143

% I-cache vs %D-cache = 71.4% vs 28.6%

AMAT_Aplha= [(%IC) X (Ht + Mr X Mp)] + [(%DC) X (Ht + Mr X Mp)]

= [(0.714) X (1 + 0.04 X 50)] + [(0.286) X (1 + 0.3 X 50)]

= 2.142 + 4.576 = 6.718 clock cycles

AMAT_Beta= [(%IC) X (Ht + Mr X Mp)] + [(%DC) X (1+Ht+MrXMp)]

= [(0.714) X (1 + 0.143 X 50)] + [(0.286) X (1+1 + 0.143 X 50)]

= 5.8191 + 2.6169 = 8.436 clock cycles

AMAT- Alpha < AMAT-Beta