ECE 452: Computer Organization and Design
Spring 2022
Homework 5: The Memory Hierarchy

Assigned: 25 Mar 2022
Due: 1 Apr 2022

Instructions:

- Please submit your assignment solutions via Canvas in a word or pdf file.
- Some questions might not have a clearly correct or wrong answer. In such cases, grading is based on your arguments and reasoning for arriving at a solution.

Q1 (60 points) Below is a sequence of 32-bit memory address references observed during the execution of an application and given as word addresses.

a. 2, 210, 45, 3, 155, 77, 42,12,150,41,156, 157**(15 points)** For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Direct mapped = 0 LSB bit offset
16 blocks = 3 bits index
4 MSB bits tag

Word Address	Binary Address	Tag	Index	Hit/Miss
2	0000 0010	0000	0010	Miss
210	1101 0010	1101	0010	Miss
45	0010 1101	0010	1101	Miss
3	0000 0011	0000	0011	Miss
155	1001 1011	1001	1011	Miss
77	0100 1101	0100	1101	Miss
42	0010 1010	0010	1010	Miss
12	0000 1100	0000	1100	Miss
150	1001 0110	1001	0110	Miss
41	0010 1001	0010	1001	Miss
156	1001 1100	1001	1100	Miss
157	1001 1101	1001	1101	Miss

b. **(15 points)** For each of these references, identify the binary address, the tag, and the index given a direct mapped cache with two- word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Two words per block = 1 LSB bit offset
8 blocks = 3 bits index
4 MSB bits tag

Word Address	Binary Address	Tag	Index	Hit/Miss
2	0000 0010	0000	001	Miss
210	1101 0010	1101	001	Miss
45	0010 1101	0010	110	Miss
3	0000 0011	0000	001	Hit
155	1001 1011	1001	101	Miss
77	0100 1101	0100	110	Miss
42	0010 1010	0010	101	Miss
12	0000 1100	0000	110	Miss
150	1001 0110	1001	011	Miss
41	0010 1001	0010	100	Miss
156	1001 1100	1001	110	Miss
157	1001 1101	1001	110	Hit

c. **(30 points)** You are tasked with optimizing a cache design for the given references. There are three direct mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 4 cycles, C2 takes 6 cycles, and C3 takes 8 cycles, which is the best cache design?

Cache1
8 blocks = 3 index bits
1 word per block = 0 offset bits

Cache 2
4 blocks = 2 index bits
2 words per block = 1 offset bit

Cache3
2 blocks = 1 index bit
4 words per block = 2 offset bits

Word Address	Binary Address	Tag	Cache C1		Cache C2		Cache C3	
			Index	Hit/Miss	Index	Hit/Miss	Index	Hit/Miss
2	0000 0010	0000 0	010	Miss	01	Miss	0	Miss
210	1101 0010	1101 0	010	Miss	01	Miss	0	Miss
45	0010 1101	0010 1	101	Miss	10	Miss	1	Miss
3	0000 0011	0000 0	011	Miss	01	Hit	0	Hit
155	1001 1011	1001 1	011	Miss	01	Miss	0	Miss
77	0100 1101	0100 1	101	Miss	10	Miss	1	Miss
42	0010 1010	0010 1	010	Miss	01	Miss	0	Miss
12	0000 1100	0000 1	100	Miss	10	Miss	1	Miss
150	1001 0110	1001 0	110	Miss	11	Miss	1	Miss
41	0010 1001	0010 1	001	Miss	00	Miss	0	Hit
156	1001 1100	1001 1	100	Miss	10	Miss	1	Hit
157	1001 1101	1001 1	101	Miss	10	Hit	1	Hit

Cache 1 has a miss rate of 12/12
For each avg access time we take 4 cycle + (miss rate * stall cycle) = 4 + (1*25) = 29

Cache 2 has a miss rate of 10 /12
For each avg access time we take 6 cycle + (miss rate * Stall cycle) = 6 cycle + (0.833 * 25) = 26.833

Cache 3 has a miss rate of 8/12
For each avg access time we take 8 cycles + (miss rate * stall cycle) = 8 cycle + (.667 *25) = 24.667

So for optimizing this cache for this specific references Cache 3 is the best as it experiences the lowest avg access time as when we try to access.

Q2 (15 points) For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

TAG	INDEX	OFFSET
31 - 11	10 - 5	4 - 0

a. **(5 points)** How many words can be stored in one cache block?
In one cache block there can be 2^5 words or 32 words per cache block

b. **(5 points)** How many words can be stored in the entire cache?
There are 2^6 blocks in the cache or 64 blocks in the cache
Meaning there can be 64*32 words in the entire cache or 2048 words = 2KB

c. **(5 points)** What is the ratio between total bits required for such a cache implementation over the data storage bits?
Total bits = 32

Q3 (40 points) In this question, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional to capacity. Assume that main memory accesses take 90 ns and that memory accesses are 60% of all instructions. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

Processor	L1 Size	L1 Miss Rate	L1 Hit Time
P1	4 KiB	8.0 %	0.98 ns
P2	8 KiB	10.0 %	0.70 ns

a. **(5 points)** Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?
P1 clock rate = 1/0.98ns = 1.020 Ghz
P2 clock rate = 1/0.70ns = 1.429 Ghz

b. **(5 points)** What is the Average Memory Access Time for P1 and P2?
P1 Avg memory Access time = 0.98ns + (0.08 * 90ns) = 8.18ns
P2 Avg memory Access time = 0.70ns + (0.10 * 90ns) = 9.70ns

c. **(5 points)** Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster?

P1:
Avg cycles added from L1 = (chance L1 Miss)(L1 miss time/ L1 Hit time)
Avg CPI added from L1 = (chance Memory access)(avg cycles added from L1)
Avg CPI added from L1 = (chance Memory access)(chance L1 Miss)(L1 miss time/ L1 Hit time)
Avg CPI added from L1 = (0.6)(0.08)(90ns/0.98ns)
Avg CPI added from L1 = 4.408
Total CPI = 1 + 4.408 = 5.408

P2:
Avg CPI added from L1 = (0.6)(0.1)(90ns/0.7ns)
Avg CPI added from L1 = 7.714
Total CPI = 1 + 7.714 = 9.714

For the next three sub-questions, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

L2 Size	L2 Miss Rate	L2 Hit Time
4 MiB	40%	8.5 ns

d. **(10 points)** What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

AMAT = L1 hit time + L1 miss rate * L2 hit time + L1 miss rate * L2 miss rate * Main memory access
P1 AMAT = 0.98ns + (0.08) (8.5ns) + (0.08)(0.4)(90ns)
= 0.98ns + 0.68ns + 2.88ns
= 4.54ns

P2 AMAT = 0.7ns + (.1) (8.5ns) + (.1)(0.4)(90ns)
= 0.7ns + .85ns + 3.6ns
= 5.15ns

The AMAT is better in both casses when the L2 cache is being used. Implementing the cache improves the average memory access time

e. **(5 points)** Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

P1:
Avg cycles added from L2 = (L2 Cycles per hit)(chance L1 misses) + (chance L2 misses)(L2 Cycles per miss)(L1 miss rate)
Avg cycles added from L2 = (8.5ns/.98ns)(0.08)+(90ns/.98ns)(0.08)(0.4)
Avg cycles added from L2 = 0.693 + 2.938 = 3.631 cycles
Avg cycles added from Memory access = (Chance instruction access memory)(avg cycles added from L2)
Avg cycles added from Memory access = (0.6)(3.631 cycles)
Avg cycles added from Memory access =2.179
Total CPI = 1 + 2.179 = 3.179

P2:
Avg cycles added from L2 = (8.5ns/.7ns)(0.1)+(90ns/.7ns)(0.1)(0.4)
Avg cycles added from L2 = 1.214 +5.14 = 6.239 cycles
Avg cycles added from Memory access = (Chance instruction access memory)(avg cycles added from L2)
Avg cycles added from Memory access = (0.6)(6.239 cycles)
Avg cycles added from Memory access =3.744
Total CPI = 1 + 3.744 = 4.744

f. **(10 points)** Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

Q4 (45 points) In this question, we will examine how replacement policies impact miss rate. Assume a 2-way set associative cache with 4 blocks. To solve the problems in this exercise, you may find it helpful to draw a table like the one below, as demonstrated for the address sequence "0, 1, 2, 3, 4."

Address of Memory Block Accessed	Hit or Miss	Evicted Block	Contents of Cache Blocks After Reference			
			Set 0	Set 0	Set 1	Set 1
0	Miss		Mem[0]			
1	Miss		Mem[0]		Mem[1]	
2	Miss		Mem[0]	Mem[2]	Mem[1]	
3	Miss		Mem[0]	Mem[2]	Mem[1]	Mem[3]
4	Miss	0	Mem[4]	Mem[2]	Mem[1]	Mem[3]
...						

Consider the following address sequence: 1,4,6,5,4,12,1,13,14,5

a. **(5 points)** Assuming an LRU replacement policy, how many hits does this address sequence exhibit?

Using an LRU replacement policy we find one hit with this specific sequence

Address	Binary	Tag 2 msb	Index 1 bit	1 bit off set	index 0 Offset 0	index 0 Offset 1	Index 1 Offset 0	Index 1 Offset 1	hit/miss
1	0001	00	0	1		mem[1]			miss
4	0100	01	0	0	mem[4]	mem[1]			miss
6	0110	01	1	0	mem[4]	mem[1]	mem[6]		miss
5	0101	01	0	1	mem[4]	mem[5]	mem[6]		miss
4	0100	01	0	0	mem[4]	mem[5]	mem[6]		hit
12	1100	11	0	0	mem[4]	mem[12]	mem[6]		miss
1	0001	00	0	1	mem[1]	mem[12]	mem[6]		miss
13	1101	11	0	1	mem[1]	mem[13]	mem[6]		miss
14	1110	11	1	0	mem[1]	mem[13]	mem[14]		miss
5	0101	01	0	1	mem[1]	mem[5]	mem[14]		miss

b. **(5 points)** Assuming an MRU (most recently used) replacement policy, how many hits does this address sequence exhibit?

Using an MRU replacement policy we find one hit with this specific sequence

Address	Binary	Tag 2 msb	Index 1 bit	1 bit off set	index 0 Offset 0	index 0 Offset 1	Index 1 Offset 0	Index 1 Offset 1	hit/miss
1	0001	00	0	1		mem[1]			miss
4	0100	01	0	0	mem[4]	mem[1]			miss
6	0110	01	1	0	mem[4]	mem[1]	mem[6]		miss
5	0101	01	0	1	mem[5]	mem[1]	mem[6]		miss
4	0100	01	0	0	mem[4]	mem[1]	mem[6]		miss
12	1100	11	0	0	mem[12]	mem[1]	mem[6]		miss
1	0001	00	0	1	mem[12]	mem[1]	mem[6]		hit
13	1101	11	0	1	mem[12]	mem[13]	mem[6]		miss
14	1110	11	1	0	mem[12]	mem[13]	mem[14]		miss
5	0101	01	0	1	mem[12]	mem[5]	mem[14]		miss

c. **(5 points)** Simulate a random replacement policy by flipping a coin. For example, “tails” means to evict the first block in a set and “heads” means to evict the second block in a set. How many hits does this address

sequence exhibit?

Heads = 0
Tails = 1

Using an MRU replacement policy we find one hit with this specific sequence

Address	Binary	Tag 2 msb	Index 1 bit	1 bit off set	index 0 Offset 0	index 0 Offset 1	Index 1 Offset 0	Index 1 Offset 1	hit/miss
1	0001	00	0	1		mem[1]			miss
4	0100	01	0	0	mem[4]	mem[1]			miss
6	0110	01	1	0	mem[4]	mem[1]	mem[6]		miss
5	0101	01	0	1	mem[4]	Mem[5]	mem[6]		miss
4	0100	01	0	0	mem[4]	mem[5]	mem[6]		hit
12	1100	11	0	0	mem[4]	mem[12]	mem[6]		miss
1	0001	00	0	1	mem[4]	mem[1]	mem[6]		miss
13	1101	11	0	1	mem[13]	mem[1]	mem[6]		miss
14	1110	11	1	0	mem[13]	mem[1]	mem[14]		miss
5	0101	01	0	1	mem[13]	mem[5]	mem[14]		miss

d. **(10 points)** Which address should be evicted at each replacement to maximize the number of hits? How many hits does this address sequence exhibit if you follow this "optimal" policy?

Using the optimal policy with this specific sequence we will find that there are 2 hits

Address	Binary	Tag 2 msb	Index 1 bit	1 bit off set	index 0 Offset 0	index 0 Offset 1	Index 1 Offset 0	Index 1 Offset 1	hit/miss
1	0001	00	0	1		mem[1]			miss
4	0100	01	0	0	mem[4]	mem[1]			miss
6	0110	01	1	0	mem[4]	mem[1]	mem[6]		miss
5	0101	01	0	1	mem[4]	mem[5]	mem[6]		miss
4	0100	01	0	0	mem[4]	mem[5]	mem[6]		hit
12	1100	11	0	0	mem[12]	mem[5]	mem[6]		miss
1	0001	00	0	1	mem[1]	mem[5]	mem[6]		miss
13	1101	11	0	1	mem[13	mem[5]	mem[6]		miss
14	1110	11	1	0	mem[13]	mem[5]	mem[14]		miss
5	0101	01	0	1	mem[13]	mem[5]	mem[14]		hit

e. **(10 points)** Describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.

It is difficult to make this kind of cache replacement policy because it is not possible for the system to know at the time of replacing a block in the cache if it is going to be needed very far in the future. And knowing the next few instructions may not help since memory can be accessed 100 times by random addresses each missing but after that if the initial page which was removed with an optimal replacement strategy we could have a hit. Therefore its extremely hard to make an optimal replacement policy as you would need to know the next instructions as well as when the next instructions would be called again and this is not possible without large amounts of overhead

f. **(10 points)** Assume you could decide upon each memory reference whether you want the requested address to be cached. What impact could this have on miss rate?

This would decrease the miss rate. It would decrease the miss rate because if we know that an address is not likely to be rescued we don't cache it leaving space in the cache for something that has a high reusability in comparison to the address which we don't cache

Q5 (35 points) In this question, we will examine paging and virtual memory systems. The following list provides parameters of a virtual memory system.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size(byte)
43	32 GiB	8KiB	4

a. **(10 points)** For a single-level page table, how many page table entries (PTEs) are needed? How much physical memory is needed for storing the page table?

Page Size = 8KiB = 2^13
page table entries = 2^43/2^13 = 2^30
Size of page table = 2^30 * 2^2 = 2^32

There are several parameters that impact the overall size of the page table. Listed below are key page table parameters.

Virtual Address Size	Page Size	PTE Size(byte)
32 bits	8KiB	4

b. **(10 points)** Given the parameters shown above, calculate the total page table size for a system running 10 applications that utilize half of the memory available.

Page Size = 8Kib = 2^13
page table entries = 2^32/2^13 = 2^19
Size of page table = 2^19 * 2^2 = 2^21
Each process has its own tlb
Page table size = (2^21)/2^1 * 10 = 10 * 2^20

The following table shows the contents of a 4-entry TLB.

Entry-ID	Valid	VA Page	Modified	Protection	PA Page
1	0	120	0	RO	22
2	1	50	1	RW	19
3	1	210	1	RW	23
4	1	260	1	RX	20

c. **(5 points)** What information is conveyed by the valid bit for entry 1 to 4?

Entry 1 is not valid
Entries 2, 3, and 4 are all valid

d. **(5 points)** What happens when an instruction writes to VA page 19? When would a software managed TLB be faster than a hardware managed TLB?

The instruction will have a TLB miss.

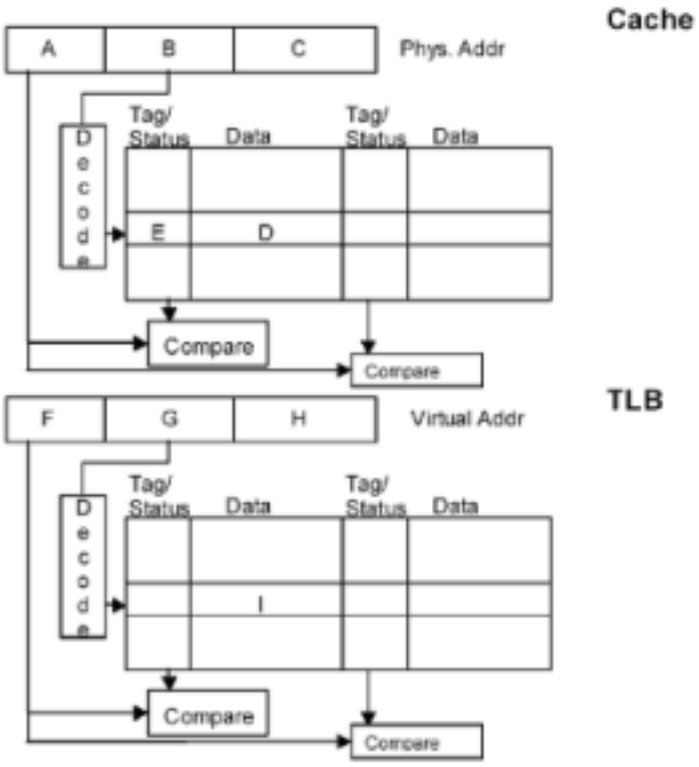
e. **(5 points)** What happens when an instruction writes to VA page 120?

It will find it however it will fail because there is a read only protection on it

Q6 (30 points). Consider a memory system with the following parameters:

• Translation Lookaside Buffer has 128 entries and is 2-way set associative.
• 64 Kbyte L1 Data Cache has 32-byte lines and is also 2-way set associative.
• Virtual addresses are 64-bits and physical addresses are 32 bits.
• 16KB page size

Below are diagrams of the cache and TLB. Fill in the appropriate information in the table that follows the diagrams4



A:

L1 Cache		TLB	
A =	bits	F =	bits
B =	bits	G =	bits
C =	bits	H =	bits
D =	bits	I =	bits
E =	bits		