CMSC611: Advanced Computer Architecture
Homework 5 Solutions

Question 1: (70 points)

a) The CPU reads a word at virtual address 124A5DF4, assuming the page translation is in the L2 TLB, but not the L1 TLB, and the word is in memory, but not found in any cache:

Virtual address: $(124A5DF4)_{16} = (0001\ 0010\ 0100\ 1010\ 0101\ 1101\ 1111\ 0100)_2$

L1 cache	**Binary**	**Hex**
Offset (4 bits):	0100	4
Index (10 bits):	01 1101 1111	1DF
Tag (18 bits):	0001 0010 0100 1010 01	4929
L1 TLB		
Virtual page number (20 bits):	0001 0010 0100 1010 0101	124A5
Page offset (12 bits):	1101 1111 0100	DF4
L2 TLB		
Virtual page number (20 bits):	0001 0010 0100 1010 0101	124A5
Page offset (12 bits):	1101 1111 0100	DF4

Physical address: $(036B0DF4)_{16} = (0000\ 0011\ 0110\ 1011\ 0000\ 1101\ 1111\ 0100)_2$

L2 cache		
Offset (5 bits):	1 0100	14
Index (10 bits):	000 1101 111	6F
Tag (17 bits):	0000 0011 0110 1011 0	6D6
L3 cache		
Offset (5 bits):	1 0100	14
Index (13 bits):	11 0000 1101 111	186F
Tag (14 bits):	0000 0011 0110 10	DA

Clock	**Action**
0	CPU→L1 cache: look up 4 bytes at tag 4929, index 1DF, offset 4 (miss) CPU→L1 TLB: look up virtual page 124A5
3	L1 TLB (miss)
4	L1 TLB→L2 TLB: look up virtual page 124A5
13	L2 TLB (hit) L2 TLB returns translation to physical page 036B0 Construct physical address 036B0DF4
14	L1 cache→L2 cache: look up 16 bytes at tag 6D6, index 6F, offset 14
18	L2 cache (miss)
19	L2 cache→L3 cache: look up 32 bytes at tag DA, index 186F, offset 14
33	L3 cache (miss)
34	L3 cache→Memory: look up 32 bytes with physical address 036B0DF4
133	Memory returns data for physical addresses 036B0DE0 - 036B0DFF L3 replaces one block in set at index 186F, tag DA L3 returns data for physical addresses 036B0DE0 - 036B0DFF L2 replaces one block in set at index 6F, tag 6D6 L2 returns data for physical addresses 036B0DF0 - 036B0DFF, virtual address 124A5DF0 - 124A5DFF L1 replaces one block in set at index 1DF, tag 4929 CPU gets data for virtual address 124A5DF4, physical address 036B0DF4

b) The CPU reads a word at virtual address 124A5DF4 with the same assumptions as the part a), but this time assuming the L2 cache uses the virtual address for its index and tag instead of the physical address:

Virtual address: $(124A5DF4)_{16} = (0001\ 0010\ 0100\ 1010\ 0101\ 1101\ 1111\ 0100)_2$

L1 cache	**Binary**	**Hex**
Offset (4 bits):	0100	4
Index (10 bits):	01 1101 1111	1DF
Tag (18 bits):	0001 0010 0100 1010 01	4929
L2 cache		
Offset (5 bits):	1 0100	14
Index (10 bits):	101 1101 111	2EF
Tag (17 bits):	0001 0010 0100 1010 0	2494
L1 TLB		
Virtual page number (20 bits):	0001 0010 0100 1010 0101	124A5
Page offset (12 bits):	1101 1111 0100	DF4
L2 TLB		
Virtual page number (20 bits):	0001 0010 0100 1010 0101	124A5
Page offset (12 bits):	1101 1111 0100	DF4

Physical address: $(036B0DF4)_{16} = (0000\ 0011\ 0110\ 1011\ 0000\ 1101\ 1111\ 0100)_2$

L3 cache		
Offset (5 bits):	1 0100	14
Index (13 bits):	11 0000 1101 111	186F
Tag (14 bits):	0000 0011 0110 10	DA

Clock	**Action**
0	CPU→L1 cache: look up 4 bytes at tag 4929, index 1DF, offset 4 (miss) CPU→L1 TLB: look up virtual page 124A5
1	L1 cache→L2 cache: look up 16 bytes at tag 2494, index 2EF, offset 14
3	L1 TLB (miss)
4	L1 TLB→L2 TLB: look up virtual page 124A5
5	L2 cache (miss)
13	L2 TLB (hit) L2 TLB returns translation to physical page 036B0 Construct physical address 036B0DF4
14	L2 cache→L3 cache: look up 32 bytes at tag DA, index 186F, offset 14
28	L3 cache (miss)
29	L3 cache→Memory: look up 32 bytes with physical address 036B0DF4
128	Memory returns data for physical addresses 036B0DE0 - 036B0DFF L3 replaces one block in set at index 186F, tag DA L3 returns data for physical addresses 036B0DE0 - 036B0DFF, virtual address 124A5DE0 - 124A5DFF L2 replaces one block in set at index 2EF, tag 2494 L2 returns data for virtual address 124A5DF0 - 124A5DFF L1 replaces one block in set at index 1DF, tag 4929 CPU gets data for virtual address 124A5DF4, physical address 036B0DF4

Convert the L2 cache to use the virtual address saves 5 cycles in total comparing to the case of the L2 cache in part a), because the L2 cache checking here can occur simultaneously with the L1 TLB checking. It is better to make this change.

Question 2: (30 points)

$$\begin{aligned}\text{Disk Capacity} &= \text{Number of Platters per Disk} \times \text{Number of Surfaces per Platter} \\ &\quad \times \text{Number of Tracks per Surface} \times \text{Number of Sectors per Track} \\ &\quad \times \text{Size of each sector} \\ &= 10 \times 2 \times 50000 \times 1000 \times 2048 \\ &= 2.048 \times 10^{12} \text{ byte} \\ &\approx 2 \text{ TB}\end{aligned}$$

$$\begin{aligned}\text{Disk Average Access Time} &= \text{Average seek time} + \text{Rotational latency} + \text{Transfer time} \\ &\quad + \text{Controller time} + \text{Queuing delay} \\ &= 10 + 0.5 \times \frac{1}{\frac{7200}{60}} \times 10^3 + \frac{2048}{64 \times 1024 \times 1024} \times 10^3 + 0.1 + 0 \\ &\approx 14.3 \text{ ms}\end{aligned}$$