NOTE: this is the SOLUTION to Quiz 7.

The correct answers are indicated for each question, with explanations as needed.

Dr. Manikas

1 4 / 4 points

We have **a virtual memory** system with *eight* virtual pages and *four* physical page frames. The page size is 1024 bytes. The page table is set up as follows:

Page #	**Present bit**	**Disk address**	**Page frame field**
0	0	01001100111	11
1	0	01001100011	00
2	0	10001100111	01
3	1	01010100111	11
4	1	10011001110	01
5	1	01111100111	10
6	0	01001100001	11
7	1	11011100111	00

We are also given the following **virtual address: 5120**
This maps to the following **main memory address (select from options below):**

- 6144
- ✓ 2048
- 1024
- 4096
- 5120
- 7168

- 3072
- 0000

Feedback

General Feedback

Virtual memory has 8 pages, where the size of each page is 1 KB = 1024 B

Therefore, the virtual addresses are the following:

Virtual addresses	page #
0-1023	0
1024-2047	1
2048-3071	2
3072-4095	3
4096-5119	4
5120-6143	5
6144-7167	6
7168-8191	7

Physical memory has 4 page frames:

Page frame	Page frame number	Physical addresses
Page frame 0	00	0-1023

Page frame 1 01	1024-2047
Page frame 2 10	2048-3071
Page frame 3 11	3072-4095

Virtual address 5120 refers to virtual memory page #5

Page table entry:

Page #	**Present bit**	**Disk address**	**Page frame field**
5	1	01111100111	10

Present bit = 1, so data is valid. The page frame field = 10. Therefore, the main memory address is **2048.**

2

4 / 4 points

We have **a virtual memory** system with *eight* virtual pages and *four* physical page frames. The page size is 1024 bytes. The page table is set up as follows:

Page #	**Present bit**	**Disk address**	**Page frame field**
0	1	01001100111	11
1	1	01001100011	00
2	1	10001100111	01
3	0	01010100111	11
4	0	10011001110	01
5	1	01111100111	10
6	0	01001100001	11
7	0	11011100111	00

We are also given the following **virtual address: 0000**
This maps to the following **main memory address (select from options below):**

- 6144
- 4096

- [x] 3072
- [ ] 5120
- [ ] 0000
- [ ] 7168
- [ ] 2048
- [ ] 1024

Feedback

General Feedback

Virtual memory has 8 pages, where the size of each page is 1 KB = 1024 B

Therefore, the virtual addresses are the following:

Virtual addresses	page #
0-1023	0
1024-2047	1
2048-3071	2
3072-4095	3
4096-5119	4
5120-6143	5
6144-7167	6
7168-8191	7

Physical memory has 4 page frames:

Page frame	Page frame number	Physical addresses
Page frame 0	00	0-1023
Page frame 1	01	1024-2047
Page frame 2	10	2048-3071
Page frame 3	11	3072-4095

Virtual address 0000 refers to virtual memory page #3

Page table entry:

Page #	**Present bit**	**Disk address**	**Page frame field**
0	**1**	**01001100111**	**11**

Present bit = 1, so data is valid. The page frame field = 11. Therefore, the main memory address is **3072.**

3 4 / 4 points

We have **a virtual memory** system with *eight* virtual pages and *four* physical page frames. The page size is 1024 bytes. The page table is set up as follows:

Page #	**Present bit**	**Disk address**	**Page frame field**
0	1	01001100111	11
1	1	01001100011	00
2	1	10001100111	01
3	0	01010100111	11
4	0	10011001110	01
5	1	01111100111	10

6	0	01001100001	11
7	0	11011100111	00

We are also given the following **virtual address: 5120**
This maps to the following **main memory address (select from options below):**

- 6144
- 3072
- 4096
- 1024
- 5120
- 0000
- 2048 ✓
- 7168

Feedback

General Feedback

Virtual memory has 8 pages, where the size of each page is 1 KB = 1024 B

Therefore, the virtual addresses are the following:

Virtual addresses	page #
0-1023	0
1024-2047	1
2048-3071	2
3072-4095	3
4096-5119	4

5120-6143	5
6144-7167	6
7168-8191	7

Physical memory has 4 page frames:

Page frame	Page frame number	Physical addresses
Page frame 0	00	0-1023
Page frame 1	01	1024-2047
Page frame 2	10	2048-3071
Page frame 3	11	3072-4095

Virtual address 5120 refers to virtual memory page #5

Page table entry:

Page #	**Present bit**	**Disk address**	**Page frame field**
5	**1**	**01111100111**	**10**

Present bit = 1, so data is valid. The page frame field = 10. Therefore, the main memory address is **2048.**

4

4 / 4 points

We have a 512 KB 8-way set-associative L2 cache that stores data as 16-byte blocks. To fetch a block from main memory, it takes 8 clock cycles for the first 8 bytes of the block (latency), then 1 clock cycle per 8 bytes for the remaining bytes of the block. What is the **miss penalty** for the cache (clock cycles)?

✓ 9 clock cycles

Feedback

General Feedback

Recall that miss penalty is the number of clock cycles to fetch one block from main memory into the cache.

We have 16 bytes/block, so divide into 8-byte groups, or 16/8 = 2 groups per block.

First group takes 8 clock cycles, and second group takes 1 cycle

So we have 8 + 1 = **9 clock cycles** to fetch one block. This is the **miss penalty** for this cache.

5 4 / 4 points

We have a 64 KB 4-way set-associative L2 cache that stores data as 16-byte blocks. To fetch a block from main memory, it takes 4 clock cycles for the first 4 bytes of the block (latency), then 1 clock cycle per 4 bytes for the remaining bytes of the block. What is the **miss penalty** for the cache (clock cycles)?

✓ 7 clock cycles

Feedback

General Feedback

Recall that miss penalty is the number of clock cycles to fetch one block from main memory into the cache.

We have 16 bytes/block, so divide into 4-byte groups, or 16/4 = 4 groups per block.

First group takes 4 clock cycles, and the remaining 3 groups each take 1 clock cycle

So we have 4 + 3(1) = **7 clock cycles** to fetch one block. This is the **miss penalty** for this cache.