Selected Question: 5.11 and 5.13

Exercise 5.11 are Exercise 5.10 4th edition

5.11 As described in Section 5.7, virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following data constitutes a stream of virtual addresses as seen on a system. Assume 4 KiB pages, a 4-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

4669,2227,13916,34587,48870,12608,49225

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Page table

Valid	Physical Page or in disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

5.11.1[10] <§5.7> Given the address stream shown, and the initial TLB and page table states provided above, show the final state of the system. Also list for each reference if it is a hit in the TLB, a hit in the page table, or a page fault.

5.11.1

Address	Virtual Page	TLB H/M	TLB		
			Valid	Tag	Physical Page
4669	1	TLB miss	1	11	12
		PT hit	1	7	4
		PF	1	3	6

			1 (last access 0)	1	13
2227	0	TLB miss PT hit	1 (last access 1)	0	5
			1	7	4
			1	3	6
			1 (last access 0)	1	13
13916	3	TLB hit	1 (last access 1)	0	5
			1	7	4
			1 (last access 2)	3	6
			1 (last access 0)	1	13
34587	8	TLB miss PT hit PF	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 0)	1	13
48870	11	TLB miss PT hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 4)	11	12
12608	3	TLB hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	11	12
49225	12	TLB miss PT miss	1 (last access 6)	12	15
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	11	12

5.11.2[15] <§5.7>Repeat 5.11.1, but this time use 16 KiB pages instead of 4 KiB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

5.11.2

Address	Virtual Page	TLB H/M	TLB		
			Valid	Tag	Physical Page
4669	0	TLB miss PT hit	1	11	12
			1	7	4
			1	3	6
			1 (last access 0)	0	5
2227	0	TLB hit	1	11	12
			1	7	4
			1	3	6
			1 (last access 1)	0	5
13916	0	TLB hit	1	11	12
			1	7	4

			1	3	6
			1 (last access 2)	0	5
34587	2	TLB miss PT hit PF	1 (last access 3)	2	13
			1	7	4
			1	3	6
			1 (last access 2)	0	5
48870	2	TLB hit	1 (last access 4)	2	13
			1	7	4
			1	3	6
			1 (last access 2)	0	5
12608	0	TLB hit	1 (last access 4)	2	13
			1	7	4
			1	3	6
			1 (last access 5)	0	5
49225	3	TLB hit	1 (last access 4)	2	13
			1	7	4
			1 (last access 6)	3	6
			1 (last access 5)	0	5

A larger page size reduces the TLB miss rate but can lead to higher fragmentation and lower utilization of the physical memory.

5.11.3 [15] <§§5.4,5.7>Show the final contents of the TLB if it is 2-way set associative. Also show the contents of the TLB if it is direct mapped. Discuss the importance of having a TLB to high performance. How would virtual memory accesses be handled if there were no TLB?

5.11.3 Two-way set associative

4669_{10}= 0001 0010 0011 1101_2

2227_{10} = 0000 1000 1011 0011_2

13916_{10} = 0011 0110 0101 1100_2

34587_{10} = 1000 0111 0001 1011_2

48870_{10}=1011 1110 1110 0110_2

12608_{10}= 0011 0001 0100 0000_2

49225_{10}=1100 0000 0100 1001_2

Original TLB (2-way set associative)

	TLB			
Set	Valid	Tag	Physical Page	Index
0	1	11	12	0
	1	7	4	1
1	1	3	6	0

	0	4	9	1

TLB

Address	Virtual Page	Tag	Index	TLB H/M	TLB			
					Valid	Tag	Physical Page	Index (set)
4669	1	0	1	TLB miss PT hit PF	1	11	12	0
					1	7	4	1
					1	3	6	0
					1 (last access 0)	0	13	1
2227	0	0	0	TLB miss PT hit	1 (last access 1)	0	5	0
					1	7	4	1
					1	3	6	0
					1 (last access 0)	0	13	1
13916	3	1	1	TLB miss PT hit	1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1	3	6	0
					1 (last access 0)	1	13	1
34587	8	4	0	TLB miss PT hit PF	1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 0)	1	13	1
48870	11	5	1	TLB miss PT hit	1 (last access 1)	0	5	0
					1 (last access 2)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
12608	3	1	1	TLB hit	1 (last access 1)	0	5	0
					1 (last access 5)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1
49225	12	6	0	TLB miss PT miss	1 (last access 6)	6	15	0
					1 (last access 5)	1	6	1
					1 (last access 3)	4	14	0
					1 (last access 4)	5	12	1

Direct mapped

4669_{10}= 0001 0010 0011 1101_2

2227_{10} = 0000 1000 1011 0011_2

13916_{10} = 0011 0110 0101 1100_2

34587_{10} = 1000 0111 0001 1011_2

48870_{10}=1011 1110 1110 0110_2

12608_{10}= 0011 0001 0100 0000_2

49225_{10}=1100 0000 0100 1001_2

Original TLB (directed mapped)

	TLB		
Index	Valid	Tag	Physical Page
0	1	11	12
1	1	7	4
2	1	3	6
3	0	4	9

Address	Virtual Page	Tag	Index	TLB H/M	TLB			
					Valid	Tag	Physical Page	Index
4669	1	0	1	TLB miss PT hit PF	1	11	12	0
					1	0	13	1
					1	3	6	2
					0	4	9	3
2227	0	0	0	TLB miss PT hit	1	0	5	0
					1	0	13	1
					1	3	6	2
					0	4	9	3
13916	3	0	3	TLB miss PT hit	1	0	5	0
					1	0	13	1
					1	3	6	2
					1	0	6	3
34587	8	2	0	TLB miss PT hit PF	1	2	14	0
					1	0	13	1
					1	3	6	2
					1	0	6	3
48870	11	2	3	TLB miss PT hit	1	2	14	0
					1	0	13	1

					1	3	6	2
					1	2	12	3
12608	3	0	3	TLB miss PT hit	1	2	14	0
					1	0	13	1
					11	3	6	2
					1	0	6	3
49225	12	3	0	TLB miss PT miss	1	3	15	0
					1	0	13	1
					1	3	6	2
					1	0	6	3

All memory references must be cross referenced against the page table and the TLB allows this to be performed without accessing off -chip memory (in the common case). If there were no TLB, memory access time would increase significantly.

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	Page Table Entry Size
32 bits	8 KiB	4 bytes

5.11.4[5] <§5.7> Given the parameters shown above, calculate the total page table size for a system running 5 applications that utilize half of the memory available.

5.11.4 Assumption: “half the memory available” means half of the 32-bit virtual address space for each running application.
The tag size is 32 - $\log_2$ (8192) = 32 - 13 = 19 bits. All five page tables would require 5 × (2^19/2 × 4) bytes = 5 MB.

5.11.5 in the backup problem
5.11.6 in the backup problem

Exercise 5.13 are Exercise 5.12 in 4th edition
Exercise 5.12
In this exercise, we will examine how replacement policies impact miss rate. Assume a two-way set-associative cache with four blocks. You may find it helpful to draw a table like those found on page 483 to solve the problems in this exercise, as demonstrated below on 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		Men[0]			
1	Miss		Men[0]		Men[1]	

2	Miss		Men[0]	Men[2]	Men[1]	
3	Miss		Men[0]	Men[2]	Men[1]	Men[3]
4	Miss	0	Men[4]	Men[2]	Men[1]	Men[3]
…						

The following table shows address sequences.

Address sequence
0,2,4,0,2,4,0,2,4

4th. 5.12.1 [5] <5.3, 5.5> assuming an LRU replacement policy, how man hits does this address sequence exhibit?

ANS: 0 hits

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		Men[0]			
2	Miss		Men[0]	Men[2]		
4	Miss	Men[0]	Men[4]	Men[2]		
0	Miss	Men[2]	Men[4]	Men[0]		
2	Miss	Men[4]	Men[2]	Men[0]		
4	Miss	Men[0]	Men[2]	Men[4]		
0	Miss	Men[2]	Men[0]	Men[4]		
2	Miss	Men[4]	Men[0]	Men[2]		
4	Miss	Men[0]	Men[4]	Men[2]		

4th. 5.12.2 [5] <5.3, 5.5> assuming an MRU (most recently used) replacement policy, how many hits does this address sequence exhibit?

ANS: 3 hits

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		Men[0]			
2	Miss		Men[0]	Men[2]		
4	Miss	Men[2]	Men[0]	Men[4]		
0	Hit		Men[0]	Men[4]		
2	Miss	Men[0]	Men[2]	Men[4]		
4	Hit		Men[2]	Men[4]		
0	Miss	Men[4]	Men[2]	Men[0]		
2	Hit		Men[2]	Men[0]		

4	Miss	Men[2]	Men[4]	Men[0]

4th. 5.12.3 [5] <5.3, 5.5> Moved to backup problems

4th. 5.12.4 [10] <5.3, 5.5> Moved to backup problems

4th.5.12.5 [10] <5.3, 5.5> describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.

ANS:

The best block to evict is the one that will cause the fewest misses in the future. Unfortunately, a cache controller cannot know the future! Our best alternative is to make a good prediction.