Exercises Week 11
Textbook Sections 5.7, 5.8, 5.13

Question 1

Textbook Exercise 5.16 <§5.7>

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 table is a stream of virtual byte addresses as seen on a system. Assume 4 KiB pages, a four-entry fully-associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number [i.e., use the next largest physical page number beyond those already allocated].

Decimal	4669	2227	13916	34587	48870	12608	49225
Hex	0x123d	0x08b3	0x365c	0x871b	0xbee6	0x3140	0xc049

TLB initial state:

Valid	Tag	Physical Page Number	Time Since Last Access
1	0xb	12	4
1	0x7	4	1
1	0x3	6	3
0	0x4	9	7

Page table initial state:

Index	Valid	Physical page or in disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
a	1	3
b	1	12

a. For each access shown above, list

- whether the access is a hit or miss in the TLB,
- whether the access is a hit or miss in the page table,
- whether the access is a page fault,
- the updated state of the TLB.

Question 2

Textbook Exercise 5.17 <§5.7>

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

Virtual address size	Page size	Page table entry size
32 bits	8 KB	4 bytes

a. Given the parameters shown above, calculate the maximum possible page table size for a system running five processes.

b. Given the parameters shown above, calculate the total page table size for a system running five applications that each utilize half of the virtual memory available, given a two-level page table approach with up to 256 entries at the first level. Assume each entry of the main page table is 6 bytes. Calculate the minimum and maximum amount of memory required for this page table.

Question 3

Many microprocessors use the AMBA bus to interconnect processors and memory within a chip. The AMBA specification includes features to support cache coherence, based on a common cache block size of 64 bytes used by all of the components in the system. While that relieves a cache designer of one design decision, there are still choices to make for cache size and associativity.

The spreadsheet dcache.xlsx provided for this exercise (shown below) has miss-rate data measured from simulations of data caches of various sizes and associativity, for a processor running the SPEC-2000 benchmark suite (Cantin and Hill, "Cache Performance for SPEC CPU2000 Benchmarks," May 2003, http://research.cs.wisc.edu/multifacet/misc/spec2000cache-data/).

Size	Direct	2-way	4-way	8-way	Full
1KB	0.0490226	0.0392158	0.0214944	0.0179407	0.0177980
2KB	0.0338158	0.0184496	0.0149315	0.0143292	0.0140771
4KB	0.0209587	0.0125664	0.0108321	0.0101989	0.0067031
8KB	0.0130101	0.0078696	0.0043419	0.0027407	0.0009342
16KB	0.0081525	0.0048237	0.0024620	0.0020958	0.0008781
32KB	0.0047908	0.0006324	0.0006225	0.0007058	0.0007660
64KB	0.0017175	0.0001342	0.0000687	0.0000570	0.0000425
128KB	0.0012558	0.0000514	0.0000402	0.0000366	0.0000363
256KB	0.0011399	0.0000394	0.0000362	0.0000361	0.0000361
512KB	0.0009755	0.0000389	0.0000360	0.0000360	0.0000359
1MB	0.0000465	0.0000383	0.0000353	0.0000352	0.0000351

Use Excel or other spreadsheet or graphing tools to create two multi-line graphs:

- Miss rate versus associativity, for various cache sizes
- Miss rate versus cache size, for various degrees of associativity

Comment on your observations from these visualizations. What guidance does this data offer the cache designer?