

**The University of Alabama in Huntsville**  
**ECE Department**  
**CPE 431 01/01R, CPE 531 01/91**  
**Fall 2020**

**Homework #7, Due November 2, 2020**

**1.0.1(10), 1.0.2(10), 1.0.3(20), 2.0.1(5), 2.0.2(10), 3.0.1(10), 3.0.2(10), 4.0.1(10), 4.0.2(10)**

**1.0** <5.4. This exercise examines the impact of different cache designs, specifically comparing associative caches to the direct-mapped caches from Section 5.4. For these exercises, use the word address stream: 21,166, 201, 143, 61, 166, 62, 133, 111, 143, 144, 61

**1.0.1** Using the references given and a fully associative cache with one-word blocks and a total size of 8 words, identify the index bits, the tag bits, and if it is a hit or miss. Use LRU replacement. Also show all entries made in the stack.

Multilevel caching is an important technique to overcome the limited amount of space that a first level cache can provide while still maintaining its speed. Consider a processor with the following parameters.

Base CPI, no memory stalls	Processor speed	Main memory access time	First-level cache miss rate per instruction	Second-level cache, direct-mapped speed	Global miss rate with second-level cache, direct-mapped	Second-level cache, eight-way set associative speed	Global miss rate with second-level cache, eight-way set associative
2.0	3 GHz	90 ns	6.5 %	18 cycles	3.5 %	30 cycles	2.5 %

**1.0.2** Calculate the CPI for the processor in the table using: 1) only a first level cache, 2) a second level direct-mapped cache, and 3) a second level eight-way set associative cache. How do these numbers change if main memory access time is doubled? If it is cut in half?

**1.0.3** In older processors such as the Intel Pentium or Alpha 21264, the second level of cache was external (located on a different chip) from the main processor and the first-level cache. While this allowed for large second-level caches, the latency to access the cache was much higher, and the bandwidth was typically lower because the second-level cache ran at a lower frequency. Assume a 512 KiB off-chip second-level cache has a global miss rate of 4 %. If each additional 512 KiB of cache lowered the global miss rate by 0.5 %, and the cache had a total access time of 45 cycles, how big would the cache have to be to match the performance of the second-level direct-mapped cache listed in the table? Of the eight-way set-associative cache?

- 2.0** <5.5> This Exercise examines the single error correcting, double error detecting (SEC/DED) Hamming code.
- 2.0.1** What is the minimum number of parity bits required to protect a 32-bit word using the SEC/DED code?
- 2.0.2** Consider an SEC code that protects 8 bit words with 4 parity bits. If we read the value 0xFAC, is there an error? If so, correct the error.
- 3.0** <5.7> As described in Section 5.4, 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 8 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**

VPN	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
10	1	3
11	1	12

- 3.0.1** Given the address stream, and the shown initial state of the TLB and page table, show the final state of the system. Also list for each reference if it is a hit in the page table, or a page fault. These are byte addresses. What would be some of the advantages of having a larger page size than 8192 bytes? What are some of the disadvantages?
- 3.0.2** 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?

**Initial TLB 2-Way**

Index	Valid	Tag	PPN	Valid	Tag	PPN
0	1	2	9	0	5	3
1	1	5	12	1	3	4

**Initial TLB Direct Mapped**

Index	Valid	Tag	PPN
0	1	0	5
1	1	1	11
2	1	2	3
3	0	2	12

- 4.0** **<5.7>** In this exercise, we will examine space/time optimizations for page tables. The following list provides parameters of a virtual memory system.

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (byte)
48	32 GiB	4 KiB	8

- 4.0.1** 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?
- 4.0.2** Using a multilevel page table can reduce the physical memory consumption of page tables, by only keeping active PTEs in physical memory. How many levels of page tables will be needed in this case? And how many memory references are needed for address translation if miss in TLB?