

**The University of Alabama in Huntsville**  
**ECE Department**  
**CPE 431 01, CPE 531 01/01R**  
**Fall 2022**  
**Cache Memory Basics**

**Due October 18, 2022**

**1.0 (20), 2.0 (10), 3.0 (30), 4.0 (15)**

- 1.0**    **<5.1>** In this exercise we look at memory locality properties of matrix computation. The following code is written in MATLAB, where elements within the same column are stored contiguously. Assume each word is a 32-bit integer.

```
for I = 1:8
    for J = 1:8000
        A(I, 1) = B(I, J) + A(J, I);
    end
end
```

- 1.0.1**    For each variable, identify whether or not it exhibits temporal locality.
- 1.0.2**    For each variable, identify whether or not it exhibits spatial locality.
- 2.0**    **<5.3>** A cache has the following parameters: b, block size given in numbers of words; S, number of sets; N, number of ways; 2, byte offset; and A; number of address bits.
- 2.0.1**    In terms of the parameters, what is the cache capacity, C?
- 2.0.2**    What are S and N for a fully associative cache of capacity C words with block size b?
- 3.0**    **<5.3>** Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit hexadecimal memory addresses, given as byte addresses. 74, A0, 78, 38C, AC, 84, 88, 8C, 7C, 34, 38, 13C, 388, 18C
- 3.0.1**    For each of these references, identify the index and the tag, given a direct-mapped cache with 16 one-word blocks. List if each reference is a hit or a miss, assuming the cache is initially empty and show every entry to the cache, including the tag value and the addresses of all data items stored. Use hexadecimal or binary, whichever is easier.
- 3.0.2**    For each of these references, identify the index and the tag, given a direct-mapped cache with two word blocks and a total of 32 words. List if each reference is a hit or a miss, assuming the cache is initially empty and show every entry to the cache, including the tag value and the addresses of all data items stored. Use hexadecimal or binary, whichever is easier.
- 4.0**    **<5.4>** You are building a computer with a hierarchical memory system that consists of separate instruction and data caches followed by main memory. You are using a MIPS single cycle processor running at 2.5 GHz.

- 4.0.1** Suppose the instruction cache is perfect (i.e., always hits) but the data cache has a 7% miss rate. On a cache miss, the processor stalls for 45 ns to access main memory, then resumes normal operation. Taking cache misses into account, what is the average memory access time?
- 4.0.2** Consider a benchmark application that has 25% loads, 15% stores, 10% branches, and 50% data processing instructions. Taking the non-ideal memory system into account, what is the average CPI for this benchmark?
- 4.0.3** Now suppose the instruction cache is also non-ideal and has a 3% miss rate. What is the average CPI for the benchmark in 4.0.2? Take into account both instruction and data cache misses.