CS 203 (2020 Fall) Assignment #2

Student ID #:

Name:

Who else you discussed with when finishing the assignment:

(While you may have your partner do all the work, this will only hurt you when the midterm and final come around so don't do it.)

* For your answer to each question, please clearly specify what formula you use to solve the problem before replacing each term with numbers.

* Please show your work as detailed as possible.

* We refuse to give credits for answers with only final results even they are correct.

1. Consider the following matrix transpose code

```
int i, j ,k;
double *A, *B, *C;
A = (double *)malloc(sizeof(double)*N*N);
B = (double *)malloc(sizeof(double)*N*N);
init_data(A, N*N);
for(i = 0; i < N; i++)
    for(j = 0; j < N; j++)
            B[i*N+j] = A[j*N+i];
// assume load A[j*N+i] and then store B[i*N+j]
output_data(B, N*N);
```

   Assume that the starting address of array A is 0x20000 and array B is 0x40000. Assume N = 128. Please answer the following questions:

   (1) Assuming that you have an AMD Bulldozer microarchitecture that has a 16KB, 4-way, 64-byte blocked L1 data blocked L1 data cache, please estimate the cache miss rate.

(2) Continued from the previous question, how many of the misses are compulsory misses? How many of them are conflict misses?

2. Smith and Goodman found that for a given small size, a direct-mapped instruction cache consistently outperformed a fully associative instruction cache using LRU replacement.

   (1) Explain how this would be possible. (Hint: You can't explain this using the three C's (compulsory, capacity, conflict) model of cache misses because it "ignores" the cache policy.)

   (2) Explain where replacement policy fits into the three C's model, and explain why this means that misses caused by replacement policy are "ignored"--or, more precisely, cannot in general be definitively classified--by the three C's model.

(3) Are there any replacement policies for the fully-associative cache that would outperform the direct-mapped cache? Ignore the policy of "do what a direct mapped cache would do."

3. You are building a system around a single-issue in-order processor running at 2 GHz and the processor has a base CPI of 1 if all memory accesses are hits. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions). The memory system for this computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32KB each. You may assume the caches use write-allocate and write-back policies. The L1 I-cache has a 2% miss rate and the L1 D-cache has a 5% miss rate. Also, 50% of all blocks replaced from L1 D-cache are dirty. The 512KB write-back, unified L2 cache has an access time of 12ns. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also 25% of all blocks replaced are dirty. The main memory has an access latency of 60ns. What is the overall CPI, including memory accesses?

4. Assume that you have a computer with 4KB pages and a 4-entry full-associative TLB that uses LRU replacement policy. If page must be brought into main memory, increment the largest page number. If the current TLB content is

	Valid	Virtual Page Number	Physical page number
0	1	0xB	0xC
1	1	0x7	0x4
2	1	0x3	0x6
3	0	0x4	0x9

and the current page table is

	Valid	Physical page number or in disk
0	1	0x5
1	0	Disk
2	0	Disk
3	1	0x6
4	1	0x9
5	1	0xB
6	0	Disk
7	1	0x4
8	0	Disk
9	0	Disk
10	1	0x3
11	1	0xC

Please identify how many TLB misses and page faults in the following address stream:

0x123D, 0x8B3, 0x365C, 0x871B, 0xBEE6, 0x3140, 0xC049

5. Assume the virtual address space of the computer is 64 bits, each page is 8KB in size, each page table entry occupies 8 bytes memory and the system is running 6 processes concurrently.

   (1) If the computer uses conventional page table, what's the total size of page tables in the system?

   (2) If we are building a 4-way set associative, virtually indexed, physically tagged cache, what's the maximum available cache size?