CSC 462/562 Homework #7 Chapter 2 and Appendix B
Due: Monday, April 8

Answer one of #5 and #7 and any four others (five total questions). Word process all answers.

1. The two largest source of stalls with modern processors are branch miss-predictions and cache misses (some of those misses are of branch prediction/target buffers). To improve on cache misses, we look to our EAT formula: hit time + miss rate * miss penalty. We can reduce the impact of cache misses by decreasing miss rate or decreasing miss penalty. Explain why neither of these are particularly easy to improve on with our modern processors aside from increasing the L1 cache's size or adding more caches (e.g., going from 2 levels to 3 levels of cache).

2. Each of the following cache improvements will improve one aspect of the EAT formula (hit time + miss rate * miss penalty) while (possibly) harming another. For each one below, state what area is improved (hit time, miss penalty, miss rate and specifically which area of miss rate – compulsory, capacity, conflict) and why, and if there is harm, to which area(s) and why.
    a. Larger block sizes
    b. Increased associativity
    c. Adding an L3 cache

3. According to the authors, all processors use some form of pipelining of the L1 cache and many use multiple banks for the L2 (and L3 if there is one) caches.
    a. What is the primary advantage of pipelining the L1 cache?
    b. When pipelining the L1 cache, there are usually two stages, what occurs in each of those two stages?
    c. The primary reason for multiple banks for L2/L3 is for power management. Yet we could also create multiple banks for L1. What advantage would this serve us? That is, how does having multiple banks improve cache performance?

4. One type of cache improvement is to avoid address translation. What is address translation, how does it impact memory access time and how can we avoid doing translation?

5. For the following high-level language code, and assuming a direct-mapped L1 data cache with 256 blocks each of which stores 4 32-bit words, and assuming the array a is an array of 2048 4-byte ints, none of which have been loaded into cache, and assuming all other variables are already stored in registers, answer the following questions.
    a. How many cache misses arise?
    b. Rewrite the code to decrease cache misses in the code, or explain why no improvement is possible
    c. How many cache misses arise in your revised code?

```
for(j=2;j<=10;j+=2)  {
      for(i=0;i<2048;i++)
            a[i]=a[i]+j*c;
      c++;
}
```

6. What is a non-blocking cache? Which types of cache optimizations, as covered in chapter 2 require a non-blocking cache? What are the main challenges of implementing a non-blocking cache?

7. You are an architect and you want to improve your Tomasulo-based 2-issue dynamic superscalar processor by providing better L1 support. Imagine that you have space and cost available to use up to 9 units of "hardware complexity" as described in the table on slides 43 and 44 of the chapter 2/appendix B powerpoint notes. That is, you can select improvements whose hardware complexities sum up to 9. Which ones will you pick and why? Keep in mind that you are doing this to improve the L1 cache in support of a dynamic issue Tomasulo 2-issue superscalar.