Homework 1

Due: Feb 3rd @ 11:59 pm (as a pdf file)

1. Briefly state what benefits remained to decreasing features size (i.e. making smaller transistors) when decreasing clock period (i.e. increasing clock frequency) was no longer feasible because of heat dissipation issues? In other words, why did the engineers keep decreasing feature size after 2003?

2. In your own words, explain the benefits of a cache. What are some common characteristics of programs does it take advantage of to improve memory access speeds?

3. Consider the following code:

```
for (int i = 0; i < N-1; i++)
  for (int j = 0; j < N; j++)
    A[i][j] += A[i+1][j];
```

   a. Compute the number of misses in terms of N and w (words in a cache line, where a word is 8 bytes) if the cache can only store 1 entire row of matrix A (cache size = N*sizeof(double)). You can assume that A starts at the beginning of a cache line, that A is an array of doubles, and that the cache is fully associative and is an LRU cache. How many compulsory, capacity, and conflict misses are there?
   b. Compute the number of misses in terms of N and w if the cache can store 2 entire rows of matrix A (cache size >= 2*N*sizeof(double)). You can make the same assumptions as part (a). How many compulsory, capacity, and conflict misses are there?

4. Consider the following code:

```
for (int i = 0; i < N; i++)
  for (int j = 0; j < N; j++)
    A[i][j] += B[i][j] + C[i][j];
```

   a. Will 2D-tiling (blocking) reduce the number of cache misses for the above code? Justify your answer. Assume the cache is empty, fully-associative and the arrays do not alias.
   b. Can you think of any optimizations (aside from tiling/blocking) that could improve the performance of this code? Rewrite the code with the optimizations. State your assumptions clearly.
   c. What is the arithmetic intensity of this code in FLOPs/byte? You can assume all arrays store double precision data.