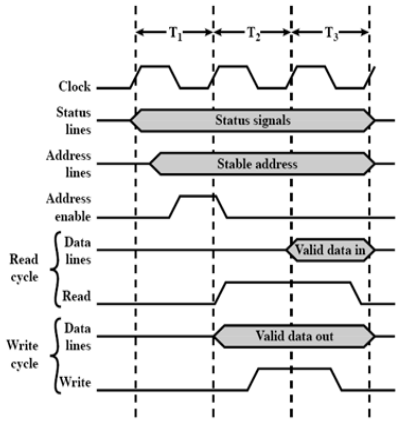
Problem Set #3

Date Assigned: Wednesday, February 25, 2015
Date Due: Friday, March 6, 2015
Points: 50

1) (10 pts) Consider the following timing diagram we discussed in class:



The 8-bit Intel 8088 microprocessor has a similar read timing diagram except four processor clock cycles are required for a read. The 8088 has an 8-bit data bus and operates at a whopping clock rate of 8MHz. Answer each of the following questions and show all work for full credit.

a) What is the maximum transfer rate in MB/s?

b) Assume that an engineer determines that memory falls short of providing the data on time by 200 ns. How many wait states (clock cycles) need to be inserted for proper operation?

c) To enforce the wait states, a Ready status line is added. How will you use the Ready status line over the given time interval? Be specific.

d) The Intel 8086 is a 16-bit processor similar to the 8088 except the 8086 has a 16-bit data bus that can transfer 2-bytes at a time. The 8086 allows even and odd alignment of word data. If an odd-word is referenced, two memory cycles each consisting of four bus cycles are required to transfer the word. Consider an 8086 instruction that involves two 16-bit operands. What is the minimum and maximum time, in nanosecods, it takes to fetch the operands?

2) (10 pts) A computer using a direct mapped cache has 2^{30} bytes of byte-addressable main memory and a cache of 64 lines where each line is 32 bytes.

a) How many blocks of main memory exist?

b) How is a main memory address divided into tag, line, and word fields? Give the size for each field.

c) To what cache line does the main memory address 0x1D2ADB63 map?

3) (5 pts) A four-way set associative cache has lines of 64 bytes and a total size of 32KB. The 8 GB main memory is byte addressable. Show the main memory address format that maps an address from main memory to cache.

4) (6 pts) The following questions are in regard to our server called ada.

a) What is the model name of the processor?

b) How many levels of cache are there?

c) In the largest/slowest cache level, what cache mapping scheme is being used? Explain in detail so I know the cache sheme and the configuration of the cache.

5) (19 pts) Here is a problem off of the Rosalind web site that is quite trivial to solve.

> **Finding a Most Likely Common Ancestor** click to expand

Problem

A **matrix** is a rectangular table of values divided into rows and columns. An $m \times n$ matrix has m rows and n columns. Given a matrix A, we write $A_{i,j}$ to indicate the value found at the intersection of row i and column j.

Say that we have a collection of DNA strings, all having the same length n. Their **profile matrix** is a $4 \times n$ matrix P in which $P_{1,j}$ represents the number of times that 'A' occurs in the jth position of one of the strings, $P_{2,j}$ represents the number of times that C occurs in the jth position, and so on (see below).

A **consensus string** c is a string of length n formed from our collection by taking the most common symbol at each position; the jth symbol of c therefore corresponds to the symbol having the maximum value in the j-th column of the profile matrix. Of course, there may be more than one most common symbol, leading to multiple possible consensus strings.

DNA Strings		A T C C A G C T
		G G G C A A C T
		A T G G A T C T
		A A G C A A C C
		T T G G A A C T
		A T G C C A T T
		A T G G C A C T
Profile	A	5 1 0 0 5 5 0 0
	C	0 0 1 4 2 0 6 1
	G	1 1 6 3 0 1 0 0
	T	1 5 0 0 0 1 1 6
Consensus		A T G C A A C T

What might not be so obvious is how to solve this problem efficiently.

I have written a program called commonancestor.c that fills a matrix of DNA strands with arbitrary DNA. There's really only one thing left to write in the program and that is the creation of the above Profile; therefore, you are to do the following:

a) Grab the program from CS430Public called commonancestor.c.

b) Finish writing the program so that when printProfile is called, the correct profile for the given DNA strands are correct. Do not change printProfile in any way as I will be diffing the results.

c) Once your program correctly works, you are going to run some performance tests on your executable as follows. Copy and paste the best of five run results in this word document.

```
perf stat -B -e cache-references,cache-misses,cycles,instructions,branches,faults,migrations commonancestor
```

d) Clear the data cache before each run. There is a ruby script cleardatacache.rb in CS430Public that you can download and run with ruby cleardatacache.rb.

e) Part of your grade will be based on coding efficiency. Use your knowledge/experience to write more efficient code.

IMPORTANT

- Do not talk about the solution to this problem at all!!!!!
- You can compare performance test results all you want. ☺ ☺
- The solution to this problem is to be single threaded only.

Note1: Please make sure your problem sets are typed, answered in order, and stapled together. This word document will be placed in CS430Public on grace.

Note2: A hard copy of your Problem Set Solution is due on the instructor's desk by 11:45am on the day the assignment is due.

Note3: Create a folder punetid and place: a) the word document with all of your solutions typed up and b) the file commonancestor.c. Then drop the punetid folder onto grace in CS430Drop.