Name: ______________________________________________

# Homework #8: Hardware and Software (13 points)

Due to Gradescope by 1:00 PM on Wednesday, October 6th

You need to submit a pdf to Gradescope; failure to assign questions to pages will result in a 10% deduction on your grade. This assignment **cannot** be submitted late.

Homework Goal: Understand memory hierarchy and access;
work with assembly language and machine language

## Hardware

1. Practice your vocab from these chapters! (1 point)
   Memory is divided into fixed-size units called ___________, each of which has a standard size of ___________.
   Cache stores ___________ from a slower device to be accessed faster, while registers store ___________ that are being used during processing.

2. What are the three components that formally define a computer? (We covered this during lecture on 9/20.) (0.5 points)

3. What are the **four** key subsystems of von Neumann architecture, and what is the purpose of each subsystem in a few words? (1 point)

4. Perform the following conversions, using $2^n$: (1 point)
   (you should always use $2^n$, ***not*** $10^m$, in this class for KB, MB, GB, etc.)
   a. 34,603,008 bytes to MB
   b. 4096 bytes to KB

c. 67,108,864 KB to GB

d. 0.5 MB to KB

5. In the table below, to have 8 memory addresses where each memory address holds one byte of data (byte-addressable), we need 3 bits to index every address.

| **Memory Address** | **Data Stored** |
|---|---|
| 000 | A |
| 001 | B |
| 010 | C |
| 011 | 1 |
| 100 | 2 |
| 101 | 3 |
| 110 | M |
| 111 | N |

If you have 32,768 bytes (32 KB) of RAM memory, then how many bits are needed to have memory addresses for every byte-addressable memory location? (1 point)

6. Suppose it takes 4 ns to access Cache Memory from the CPU and 60 ns to access RAM from the CPU. Assume the Cache Hit Ratio is 91%. Compute the average access time in ns. Give your answer to one decimal place. (1 point)

7. What is something that can be done to increase the cache hit rate? (0.5 points)

8. For the two different types of locality that we discussed in class, give an example of where each locality might occur in a program. (0.5 points)
    a. Temporal Locality
    b. Spatial Locality

9. Consider the following structure of the instruction register. (1.5 points)

| Op code | Address-1 | Address-2 |
|---|---|---|
| **8 bits** | **28 bits** | **28 bits** |

    a. What is the maximum number of distinct operations that can be recognized and executed by the processor on this machine?
    (Hint: What part of the instruction determines what the operation is?)
    b. What is the maximum memory size on this machine?
    (Hint: Each memory location has an address.)
    c. How many bytes are required for each instruction?
    (Hint: A single instruction may be stored in the instruction register at a time.)

## Software

10. Rank the following programming languages from the lowest level (left) to the highest level (right): Python, Machine language, Assembly language (0.5 points)

11. What is the difference between a compiler and an interpreter? **Give an example** of a language that uses each. (0.5 points)

12. Assume memory address M[1] contains the value 2, address M[2] contains the value 4, and address M[3] contains the value 6. What are the values of all three addresses after the following assembly instructions are executed? SUB M[3] M[2] M[1] works how you might expect (i.e., similar to ADD M[3] M[2] M[1]) (1 point)

```
ADD M[1] M[2] M[1]
SUB M[3] M[3] M[1]
ADD M[1] M[1] M[1]
SUB M[1] M[2] M[3]
ADD M[3] M[3] M[2]
SUB M[2] M[2] M[3]
```

13. Suppose a, b, c, and d are in memory locations M[100], M[101], M[110], and M[111], respectively. Write an algebraic equation that represents the following assembly language instructions: (1 point)

```
ADD M[100] M[100] M[100]
ADD M[100] M[100] M[100]
ADD M[111] M[111] M[111]
ADD M[110] M[100] M[111]
ADD M[110] M[110] M[101]
```

14. Assume the variables v, w, x, y, and z are stored in memory locations M[001], M[010], M[011], M[100], and M[101], respectively. Using the machine language instructions shown in Section 4.2, fill in the blanks to translate the following algorithmic operations into their machine language equivalents. You can overwrite a memory location for an intermediate calculation, if that location is no longer needed. See Zybooks activities 4.2.2 and 4.2.3 for examples. (2 points)

    a. Set v to the value of (w + x) + (y + z)

```
1 1     0 1 0   _ _ _   _ _ _
1 1     1 0 0   _ _ _   _ _ _
1 1     _ _ _   0 1 0   1 0 0
_ _     0 0 0   0 0 0   0 0 0
```

b. Input v from the user, then display v * 2

```
_ _    _ _ _    0 0 0    0 0 0
1 1    _ _ _    _ _ _    _ _ _
_ _    0 0 1    _ _ _    _ _ _
_ _    0 0 0    0 0 0    0 0 0
```