# COIS 2300H Assignment 1

This assignment is worth 10% of your final grade. Bonus marks will not let you exceed 100%.

Theory Questions from the textbook **0.5 marks** for each full question (3.5 marks total)

(§ references a particular section in the book)

In the ~~un~~likely event that you discover solutions to these problems on the InternetSri and Alaadin have modified them, and marked that change by ~~crossing out~~ the original. You are to answer with the changes wee made. Feel free to use the solutions to help you figure out how to solve the problem, that’s the idea.

# A

**1.4**  <§1.4> Assume a color display using ~~8~~ 10 bits for each of the primary colors (red, green, blue) per pixel and a frame size of ~~1280 × 1024~~ **3440x1440 (widescreen)**

**a.** What is the minimum size in bytes of the frame buffer to store a frame? **18,576,000**

**b.** How long would it take, at a minimum, for the frame to be sent over a **400**

Mbit/s network? **0.37152 s**  
c. **Sri/Alaadin added question:** What bandwidth do you need (at a minimum) to transmit this data at **144** frames per second? **7133.184 megabit/second**  
(note that in **a** we are ignoring the fact that data would mostly likely be packed into 32 bits per pixel regardless of colour depth).

# B

**1.5** <§1.6> Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a ~~3 GHz~~  4.4GHz clock rate and a CPI of 1.5. P2 has a ~~2.5 GHz~~ 3.3GHzclock rate and a CPI of 1.0. P3 has a 4.1 GHz clock rate and has a CPI of 2.2.

**a.** Which processor has the highest performance expressed in instructions per second? **Sri/Alaadin: give that number P2**

**b.** If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

**c.** We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

# C

(this question is left as is because it’s carefully chosen to illustrate a point)

**1.12** Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.  
**1.12.1** <§§1.6, 1.10> One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and

**1.12.2** <§§1.6, 1.10> Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.

**1.12.3** <§§1.6, 1.10> A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance.

Check if this is true for P1 and P2.

**1.12.4** <§1.10> Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as MFLOPS = No. FP operations / (execution time × 1E6) but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.

**1.12.Sri/Alaadin -** Explain in ~100-200 words what this question is trying to illustrate, and what metric(s) you should use to compare CPU performance given there are obvious problems with clock rate, IPC, IPS etc.

**The question is trying to illustrate that there are few fallacies consistent when it comes to performance metric. Since clock rate denotes the CPU speed, It is believed that the larger the clock rate of the processor means larger the performance of that processor but in the question clearly that wasn’t the case. Thus, clock rate alone isn’t only the factor in determining the performance of the CPU. It depends on other factors like CPI and execution of instruction. Same way, MIPS or MFLOPS alone can’t either be used to determine the performance of the processor since they are calculated using clock rate over CPI and hence not independent.**

**Best way to determine the speed of CPU is using clock rate(but not only clock rate), multi processors that can run independently or co-operatively, and front side bus and CPU together.**

# D

**1.15** <§1.8> **Sri/Alaadin Note: Do this problem using a spreadsheet** When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.

Assume a program requires t = 100 s of execution time on one processor. When run *p* processors, each processor requires t/p s, as well as an additional 4 s of overhead, irrespective of the number of processors. Compute the per-processor execution time for ~~2, 4, 8, 16, 32, 64, and 128~~  2, 4, 6, 8, 12, 16, 24, 28, 32, 56, 64, 224, and 448 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead).  
  
(Note: those numbers of nodes are all real CPUs or at least multi CPU configurations you can buy for home/server/HPC systems).

# E

**2.7** <§2.3> Show how the value**s** ~~0xabcdef12~~ 0x134 433C **AND** -23002020 **(-15EFBA4)** (that second one is base 10, so convert to hex first) would be arranged in memory of a little-endian and a big-endian machine. Assume the data is stored starting at address 0. – (By endianness we mean byte ordering <http://www.yolinux.com/TUTORIALS/Endian-By-2te-Order.html> has a guide)

# F

**2.8** <§2.4> Translate 0x134 433C into decimal. **Sri/Alaadin note: Show your steps 20,202,300**

# G

**2.46** Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: ~~500~~ 700 million arithmetic instructions, ~~300~~ 200 million load/store instructions, ~~100~~ 80 million branch instructions.

1. **2.46.1** <§2.19> Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, and the cost of increasing the clock cycle time by only 10%. Is this a good design choice? Why?
2. **2.46.2** <§2.19> Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

# H: Sri/Alaadin General Theory Question (2.5 marks total)

Read up on CPU benchmarks such as

<http://www.anandtech.com/show/10337/the-intel-broadwell-e-review-core-i7-6950x-6900k-6850k-and-6800k-tested-up-to-10-cores/8>  
and

<https://www.cpubenchmark.net/high_end_cpus.html>

<https://www.cpubenchmark.net/>

and anything else you find with an Internet search.

The problem with those benchmarks is that processors which are sometimes many times faster than each other (e.g. the same architecture and frequency but twice as many cores etc.) don’t seem that much faster in real applications, and trying to compare different architectures is quite challenging.

Describe a way for an end user to compare CPU’s and decide which one to buy. Some possible things to consider: Performance for certain workloads, value, cost, likely future performance or whatever else you think might be important.

**The CPU matters a lot when we consider the upgrading the existing system. Higher clock speeds and core counts can make a major difference in overall performance. This decision is made as to which one has the best processor that is available. We check how much cores there is and the cost as well is an important to consider. For mainstream users, current generation parts is a choice to make, like AMD Ryzen 3000 or Intel 9th Generation Core and Intel is doing better on the 1080p gaming on some titles and AMD handling tasks like video editing faster. But as long as the reason of buying CPU is not only gaming AMD delivers more cores and general performance plus 4.0 on its latest chips at better value. Clock speed however is more important than the core number as higher clock speeds translate to snappier performance in simple however, more cores is better for time-consuming workloads faster. The latest generation is the best option most of the time, even though the money and cost would be lot. A CPU, latest gen or not, should not have a weak storage, RAM and graphics. It does increase the budget but weak storage with a good core CPU is a bad decision just to save small money. While buying CPU, we can know about it buy it’s model number that mentions the generation in the first digit of the four number(ex- the 8 in core i7-8400). The rest of the numbers just mark various models in the line with higher generally being better with more cores and/or intel chips are “K” at the end of an Intel chip means it’s unlocked for overlocking. A processor’s on board cache is used to speed up the access to the data and instructions between your CPU and RAM.**

(When talking about cost you can consider just the CPU itself, or overall system cost. AMD Ryzen parts have very different performance with different speed RAM for example, which makes this even more complicated).

~ 200-500 words is fine.

# I: Programming Problems **2 Marks Each**

I suggest you stick to MARS for this, it’s possible to use SPIM but more of a headache to setup and mark. Submit your source code, and documentation on how you tested it.

**Program 1:** Write a MIPS program that asks the user for two number inputs (we’ll call them *a* and *b*), calculate and print to screen: , if *b* is not 0, also calculate . Then check if a + b is in the range 10 … 20. Stick to integers, but allow positive or negative values (don’t check input types, and don’t test edge cases like very large integers)

**Program 2:** Write a MIPS program that will ask the user for a negative integer (we will call this *n* ), check if it is negative, if not throw an error. Iteratively calculate the factorial of n squared (). Assume n is between 0 and -5, there are some issues with factorial calculations with big numbers which we will discuss in class… don’t sign yourself up for those). Recursion is a problem for future assignments, so do this iteratively.