Instructions: All assignments are due by **midnight** on the due date specified.

Every student must write up their own solutions in their own manner.

Please present your solutions in a clean, understandable manner.

Assignments should be typed and submitted as a PDF.

Computer Measurements

1. (15 points) Consider 3 different processors: P1, P2 and P3 that all execute the same instruction set. P1 has a clock rate of 3 GHz and a CPI of 1.3. P2 has a clock rate of 3.5 GHz and a CPI of 1.7. P3 has a clock rate of 4 GHz and a CPI of 2.1.
   (a) Which processor has the highest performance? Express the performance of each processor in instructions per second.
   (b) If the processors each execute a program for 10 seconds, find the number of clock cycles each processor spent to execute the program. Also show the number of instructions each processor executed in 10 seconds.
   (c) We make some changes to P1 that allows for a faster clock rate, but these changes increased the CPI to 2. What clock rate should we set for P1 to maintain the same performance?

2. (30 points) We will examine a processor that has 3 classes of instructions: arithmetic, load/store and branch instructions. In this processor, arithmetic instructions have a CPI of 1, load/store instructions have a CPI of 1.2, and branch instructions have a CPI of 5. Each processor has a 2 GHz clock frequency. We will be executing a program that requires the execution of 2.56×10^9 arithmetic instructions, 1.28×10^9 load/store instructions and 2.56×10^8 branch instructions on a single processor.

   We will parallelize this program to run on multiple cores. When we parallelize the program, the number of arithmetic and load/store instructions needed per processor will be divided by $0.7 \times p$, where p is the number of processors. The number of branch instructions per processor does not change.

   For instance, if the program is run on 2 processors, it will require $(2.56 \times 10^9)/1.4$ arithmetic instructions, $(1.28 \times 10^9)/1.4$ load/store instructions, and 2.56×10^8 branch instructions per processor.

   Given this setup, answer the questions below:

   (a) (5 points) Find the total execution time for this program on 1, 2, 4 and 8 processors.
   (b) (5 points) Show the relative speedup of the 2, 4 and 8 processor configurations to the single processor result.
   (c) (10 points) If the CPI of arithmetic instructions are doubled, what would the impact be on the execution time of the program on 1, 2, 4 or 8 processors?
   (d) (10 points) We are able to improve the CPI of load/store instructions only for the single processor configuration. What should the CPI of load/store instructions be reduced to if we want it to match the performance of 2 processors?

3. (15 points) Consider 2 different processors (with different ISA as well as implementation) executing a benchmark: P1 has a clock rate of 4GHz, average CPI of 0.9, and requires the execution of 5×10^9 instructions to complete the benchmark. P2 has a clock rate of 3 GHz, average CPI of 0.75, and requires the execution 1×10^9 instructions to complete the benchmark.
   (a) In marketing, processors with higher clock speeds are often marketed as the faster processor. Check if this is true for P1 and P2. Which one is faster when executing the given benchmark?

(b) Another fallacy is to use the number of instructions executed as a measure of performance. How long would P1 take to execute 1×10^9 instructions? How many instructions can P2 execute in that same time? Is the processor that can execute more instructions in the same amount of time the faster one?

(c) Another common measure of performance shown is FLOPS (floating-point operations per second), often used with a prefix such as MFLOPS (millions of FLOPS) or more recently TFLOPS (tera FLOPS). Let's assume 40% of instructions executed on both P1 and P2 are floating-point instructions. Find the FLOPS figures for each processor. Does it match their actual performance?

4. (5 points) Translate the base 16 value `0xabcdef12` into binary and decimal. Alongside your answer, explain how you would accomplish each conversion if you only had access to a simple desk calculator (i.e. a calculator that can only perform simple arithmetic operations). As a hint, you don't even need the calculator for the conversion to binary.

MIPS Assembly Encoding

5. (20 points) For each MIPS instruction shown below, show the values of the opcode (op), rs and rt fields. For I-type instructions, additionally show the value of the immediate field. For R-type instructions, additionally show the value of the rd field and funct fields.

   (a) `addi $t0, $s6, 4`
   (b) `add $t1, $s6, $0`
   (c) `sw $t1, 8($t0)`
   (d) `lw $t0, 4($t0)`
   (e) `sub $s0, $t1, $t0`

6. (5 points) What is the assembly instruction that corresponds to the following binary value?
   `0000 0010 0001 0000 1000 0000 0010 0000.`

7. (10 points) What is the assembly instruction that corresponds to the following MIPS fields? What is the hexadecimal representation?

   (a) `op=0, rs=3, rt=2, rd=3, shamt=0, funct=0x22`
   (b) `op=0x23, rs=1, rt=2, const=0x4`