Homework 2

Policies:

- The homework is designed to be done by a group of *three* students.
- You are free to discuss the homework and how to divvy up the work between you and your partner, however, for maximum benefit, both partners need to be aware and participate in solving *every* problem.
- Inter-group discussions are encouraged, but each group must write its own solution. Copying solutions between groups is considered cheating and will be dealt with according to the policy announced in the first lecture of the class.
- Using material from the Internet including text and code are permitted provided that the right attribution is made, and that the student understands thoroughly what is being used.
- Project due by 5PM on September 28, 2023.

Advice

Start early!!

Questions:

1. You are designing the next supercomputer for KAUST. You have two choices for an Interconnect, with the following parameters:
    a. Device A: Transfer latency: 5 microseconds; bandwidth: 100Gb/sec.
    b. Device B: Transfer latency: 6 microseconds; bandwidth: 125Gb/sec.

    How much time does it take to send a message of 128 Bytes on either system? What is the network throughput? Which of the two devices do you use?

2. A pipeline consists of the following stages:
    - Instruction fetch (35 picoseconds)
    - Instruction decode (400 picoseconds)
    - Operand fetch (200 picoseconds if in register, 500 picoseconds if in cache)
    - Operand fetch (200 picoseconds if in register, 500 picoseconds if in cache)
    - Execution unit (50—400 picoseconds)
    - Result store (200 picoseconds)

    What is a good choice for the frequency of the processor? Provide arguments to support your choice.

3. Using the built-in profiler in Python, run the following experiment: Install Python and run a Web server using the built-in library http.server. Construct a demo Website and have your Python server run it. Then, use the internal profiler (cProfile) in Python to profile the Web server. General a flame graph out of the profiled data and compare it to the default output of the Python profiler. Also, assess the time dilation due to profiling.

4. The computation Y = a * X + Y has long been used as a benchmark for processor performance (known as SAXPY). The typical assembly code for this function is:

```
start:  ld      f1, (r1)        # the machine has two register sets
        mul     f4, f2, f0      # a * X[i], f4 holds the results, a is in f0,
                                # and f2 holds X[i]
        ld      f6, (r2)        # Y[i],
        add     f6, f4, f6      # a*X[i] + Y[i]
        st      f6, (r2)        # Updating Y[i]
        add     r1, r1, 8       # increment X[i]'s index
        add     r2, r2, 8       # increment Y[i]'s index
        add     r3, -1          # the loop index
        bnz     r3, start
```

   Assume a pipeline containing a single-ported integer register file and a single-ported floating point register file. Assume an integer adder that finishes in one cycle, and a floating point multiplier that is pipelined over 7 stages, and a floating point adder that is pipelined over 4 stages. The instruction fetch and decode issue one instruction per cycle. Show how the code will execute on the pipeline. You may use out of order execution, register renaming, and branch prediction. Compute the IPC.