ECEn 528
Study Guide - Kilo

- Read “Towards Kilo-Instruction Processors”
    - Things to focus on
        - Their methodology: identify the problems and come up with solutions that fit together
        - Don't worry too much about section 6; they're not proposing anything new, but just going over possible solutions from previous work
    - Clarifications
        - When they say IQ, they mean what we call “reservation stations.” They use two: one for FP instructions and one for integer instructions.
        - On p. 400, the notion that the physical-logical mapping need not be kept in the checkpoint is rather surprising; I suggest you work out why this is so (hint: it has to do with the fact that the renaming table (RAT) is based on a CAM)
        - They assume a baseline of a late-read OoO machine with architectural and speculative state in one physical register file
    - Answer the following questions:

        1. What are the critical resources which prevent a conventional OoO design from having thousands of instructions in flight?

            ROB, IQs (reservation stations), physical register file, L/S queues

        2. How can each of these resources be made smaller/faster/lower power while still keeping correct behavior?

            ROB: OoO commits, checkpointing
            IQs: slow-lane instruction queue,
            Physical register file: ephemeral registers
            L/S queues: hierarchical, prediction, buffers, hashing,

        3. After all that work, what kind of IPC do they actually achieve?

            Between 2 and 3