COM219 Cumulative Journal Description

The cumulative journal for COM219 should be of a summarizing nature spanning all of the main concepts we have covered in class. However, it should not be as detailed as the course notes. The purpose of this journal is to have you distill the concepts and clarify their relationship with each other toward describing how simple microprocessor based computer systems work. Note that it is easy to produce something at the class notes level of detail but summarizing is a different and more involved task. For example, the syllabus on Moodle could serve well as a guideline. The collection of topics should outline the main theme of the course and demonstrate your understanding of how a simple computer works.

I purposefully did not create a complete list of topics for you to cover in the journal. Based on what we have done in class you will need to decide what to include and choose the level of detail that is necessary to demonstrate your understanding. But that doesn't mean you write things such as 'in this course we learned how memory works' or 'as the professor discussed…'. You can write a paragraph for each main concept. Wherever possible you should make reference to the Mic-1 architecture, exemplify and discuss the concept in this context. An example is given below. Note that this paragraph includes information from multiple chapters.

Example paragraph

The **fetch-decode-execute cycle** is one of the core concepts in computer architecture. In the most basic sense it is the repeated sequence of operations the CPU goes through to execute a program. In order to accomplish this task, each CPU is equipped with a PC (program counter) and an IR (instruction) register in addition to the other registers used for storing data. The fetch-decode-execute sequence involves retrieving the instruction pointed to by the PC register into the IR register, updating PC to point to the next instruction, decoding the instruction in IR and determining if another operand is needed from memory, fetching the additional operand if necessary, executing the instruction and repeating the entire sequence. We have seen how each step works in our example CPU, the Mic-1. In particular, we saw that 'fetch' is performed by the memory unit (fetch; PC/MBR) to retrieve the machine instructions from the method area, that 'decode' translates the machine instructions into a sequence of microinstructions which produce hardware signals to control the components of the CPU, that 'execute' is what happens as a result of the execution of the microinstruction sequence corresponding to the current ISA-level (macro) instruction which generally involves use of the data path. The write-back is performed by the memory unit (read/write; MAR/MDR) to write any result of the instruction back to memory.

P.S. Since you have read this paragraph you do not need to write in detail about the **fetch-decode-execute cycle** in your journal. You can simply refer to this paragraph in the appropriate place.