CPU Design Project: Part 2 – Datapath, Report Due Wednesday, October 9

In this part, you are to design the datapath of a CPU that will realize the instruction set architecture (ISA) designed in the previous part (including any "adjustments" made to the ISA). Include the following in your report.

1. A *register-level* block diagram of the datapath, with all components and control signals clearly labeled.
2. External to the CPU, your block diagram should also show connections to the system memory (Von Neumann architecture) or memories (Harvard architecture). These memory components are not part of the CPU design, but since data and instructions move between the CPU datapath and the system memory, it is important to account for the datapath's interface to the memory/memories.
3. A description of the inputs, outputs, and function of each component in the datapath.
4. For each instruction of your ISA, list the register transfers, or sequence of register transfers, required to fetch and execute the instruction. Register names should correspond to components in your datapath diagram.
5. For single-cycle and pipelined datapaths, create a **truth table** listing all control signals and the values of each control signal required for executing each instruction in your ISA. For multicycle datapath, create a **state diagram**, and also a **state table** listing all control signals and the values of each control signal corresponding to the different states (like fetch, decode, ALU operation, etc) in your diagram. (Note that in state like "ALU operation", the control signals may vary depending on different opcodes.)
6. A discussion of the tradeoffs and other design decisions made in developing your datapath. This should include:
   - Cost vs. speed tradeoffs that you considered.
   - Why you chose a single-cycle, multi-cycle design or pipeline datapath.
   - Decisions related to "shared" and/or "dedicated" components.
   - Selection of edge-triggered vs. latching registers.
   - Other decisions that were considered.

Major Datapath Components Likely to be needed:

1 ***ALU:*** The ALU must provide all arithmetic and logic functions required to support your instruction set. It should not provide unnecessary functions.
2 ***Register file:*** Design as a multi-port "memory array". DO NOT instantiate individual registers.
3 ***Sign/zero extension logic***, as appropriate, for ALU inputs.
4 ***Program counter (PC).*** PC should be a simple register.
5 ***Instruction register (IR)*** (if required).
6 Assorted ***multiplexers*** for data paths and register address inputs.