# Design Overview

Attached is our design overview of a non-pipelined version of the WISC-SP10. Any inputs that are detached from any obvious connection are outputs of control. We chose to do this to stop the diagram from getting too cluttered.

The design as attached is laid out in a sequential manner. We did this to make adding pipeline stages later easier. The PC will be used to fetch an instruction. This is then decoded and control does some magic. The register file grabs appropriate registers, and control sets the inputs of the ALU. The output of the ALU is written to data memory, and the PC is appropriately updated to fetch the following instruction.

Below are breakdowns of the major instruction types of our design…

|  |  |
| --- | --- |
| Alu(iformat) | rd 🡨 rs (alu\_op) imm |
| Alu(rformat) | rd 🡨rd (alu\_op) rt |
| Mem | addr = rs + imm; data in/out = rd |
| Set | rs –rt; rd = condition true |
| Branch | check rs, update pc |
| LBI | rs 🡨 sext(imm) |
| SLBI | rs 🡨 rs << 8 | zext(imm) |
| Jump | Update PC |
| Jump/Link | $7 🡨 PC; update PC |