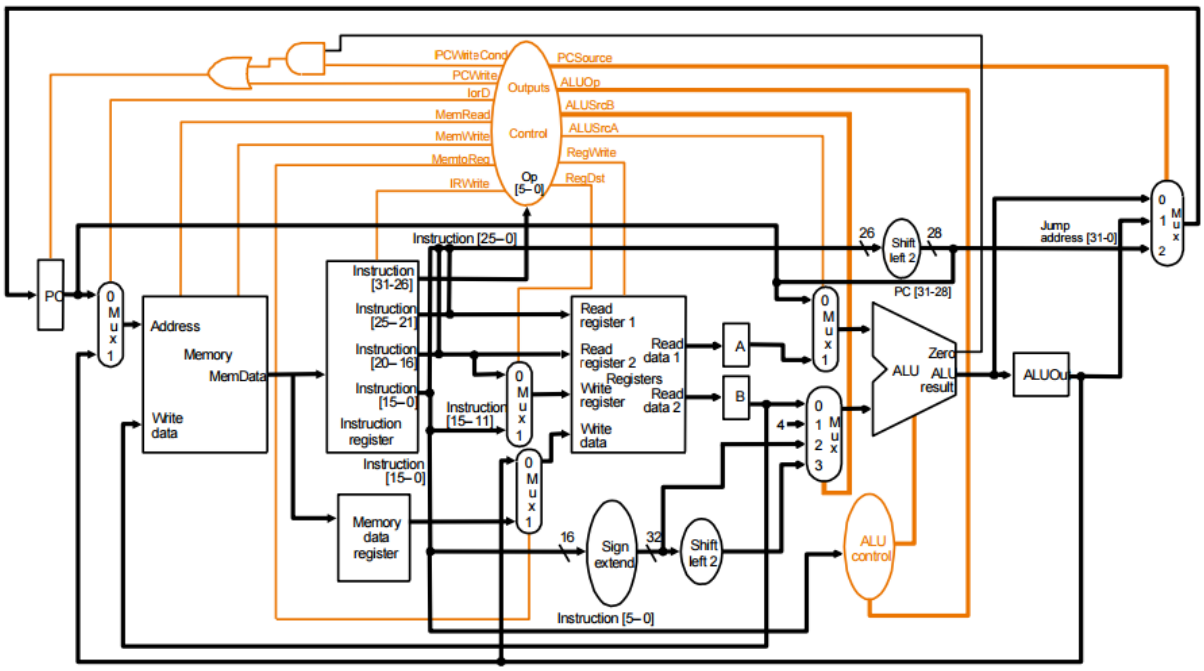
Simple Multicycle CPU



Implementation of the multicycle CPU

We used the above diagram as our base to make the multicycle CPU, though we didn't use all the modules mentioned in the diagram since few of the functionalities were not included in the lab.

- Modules Used:
  - 32-bit Registers for PC, Instruction Register, Memory Data register, Read Output 1 & 2 from the register file and the ALU output.
  - Multiplexers for selecting the read and write addresses to the memory, ALU sources and Register file write source.
  - Sign and zero extend blocks for immediate operands.
  - ALU for arithmetic and logical instructions.
  - Control block to control the state of the state machine which has been used to implement the logic of the CPU.
- Control Logic and State Machines
  - The most important module of the CPU is the control block.

- We have five basic states for each instruction: Fetch, Decode, Execute, Memory and Write back.
- We created a state machine with the help of the control signals to demonstrate the operation in each state and obtain the output.
- Each state triggers few specific control signals that drive the output of the CPU.

- Memory
  - During operation, it is always "reading" to memdata.
  - Write operation occurs at the rising edge of the clock.

- Register File Module
  - 2 read, 1 write port
  - Always reading on read ports z I.e. change the register address on ra1 or ra2 and rd1, rd2 change immediately
  - Writing occurs at rising edge of clock z if regwrite signal is active

- Write-back step back step
  - Write the load data, which was stored into MDR in the previous cycle, to register file
  - MemtoReg = 1 (to write the result from the memory)
  - Assert RegWrite (to cause a write)
  - RegDst = 0 to choose rt (bits 20:16) field as the register number

- Finite state machines
  - a set of states and
  - next state function (determined by current state and the input)
  - output function (determined by current state and possibly input)
  - We used a Moore machine (output based only on current state)
  - If the output function can depend on both the current state and the current input, the machine is called a Mealy machine.