Single Cycle MIPS Microarchitecture

The single cycle microarchitecture executes instructions in one cycle. will be divided into two interacting parts: the datapath and the control.

1. The datapath operates on words of data (32-bits) and contains structures such as memories, registers, ALUs and multiplexers.
2. The control unit receives current instruction from the datapath and tells the datapath how to execute the instructions.

The best way to design a complex system is to start with designing the state elements (e.g. Registers, memories and program counter) then add the combinational logic in between to compute the new state based on the present stat.

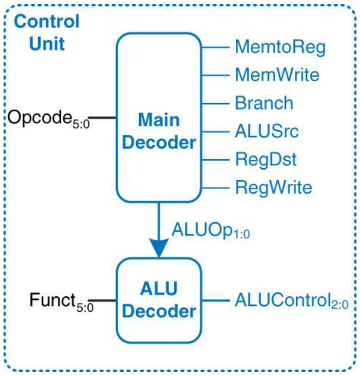
Part 1: Datapath

In lab 5, the MIPS datapath has the following state elements:

1. Instruction Memory (32-bit X 2^{32}) memory holding all instructions in machine code. It has a single read port that takes 32-bit instruction address input a, and outputs 32-bit data (instruction) into the read port, rd.
2. Data Memory (32-bit X 2^{32}) to load and store data. It has a single read/write port and a WE signal. If WE is high, it write the wd signal into address a on the rising edge of the clock. If the WE is 0, it reads address a into rd.
3. Register file (32-bit X 32)
4. Program counter is a 32-bit register, it outputs PC pointing to the current instruction. The input PC' is the next instruction to be executed
5. Additionally, the datapath has: ALU, sign extend circuit, Adder, and a number of multiplexers.

Part 2: Control Unit: shown in the figure belwo.

The control unit computes the control signals based on the opcode and `funct` fields of the instructions. Most of the control signals is determined by the opcode field with exception of the R-type instructions. To simplify the design, the control unit will be divided into two blocks: the main decoder and ALU decoder. The main decoder computes most of the outcome from the opcode. It also determines a 2-bit `ALUOp` signal. The ALU decoder uses the `ALUop` signal in combination with the `funct` field to compute `ALUControl`. The meaning of the `ALUOp` signal is shown in the table below.



$ALUOp_{1:0}$	Meaning
00	Add
01	Subtract
10	Look at `funct` field
11	