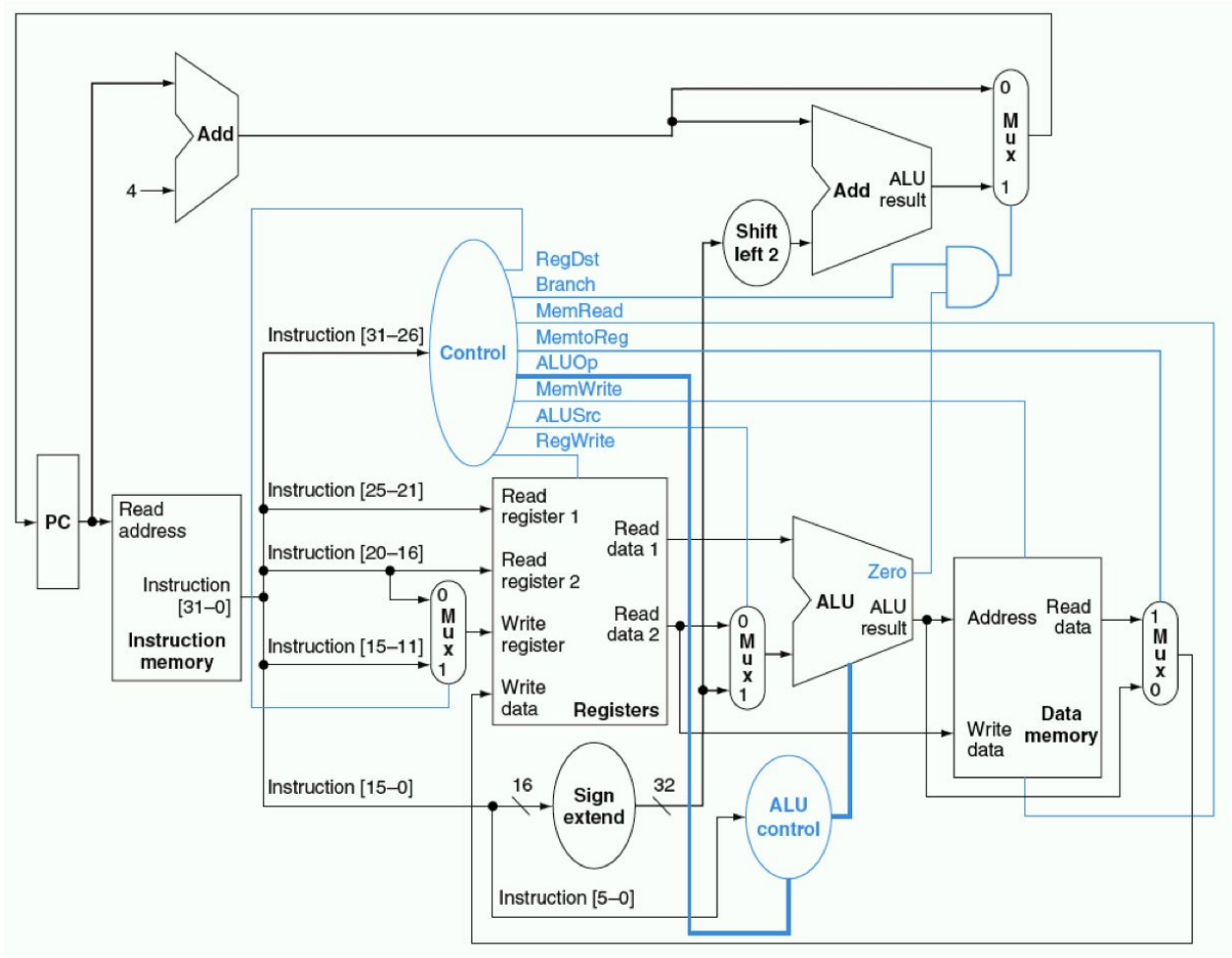
CS 161 Lab 5
TA: Jose Rodriguez

Team Leader: Selik Samai
Team Members: James Hollister, Roberto Pasillas, Selik Samai

Remarks:
We implemented the port mapping in order to complete the following datapath:



The following components were implemented in order to map the connections between the various signals and components.
components:
Program_Counter : generic register
MEMORY_UNIT : memory
REGISTERS : memory
REG_MUX_1 : mux_2_1 (5)
REG_MUX_2 : mux_2_1(32)
ALU_MUX : mux_2_1(32)

PC_MUX : mux_2_1 (8)
CTRL : control_unit
ALU_CTRL : alu_control
ALU_UNIT : alu

We also used various temp vectors for some of our signals

Our alu_control is able to handle LW/SW, branch and R-type instructions.

In order to read the .coe files, we used an entity called memory which read in from “init2.coe”, PC is updated by adding 1 to the PC. normally you’d add 4; however we made ours word addressable(32 bits) so we are getting the next word every time.

```
TEMP_aluMux             <= std_logic_vector(resize(signed(MEM_im_out(15 downto 0)),
MUX_aluIn'length));--sign extend Instr(15 - 0)
        TEMP_pcMux_s    <= (CTRL_branch and ALU_zero);
        --TEMP_pcMux_d0 <= std_logic_vector(unsigned(PC_out) + 4);
        TEMP_pcMux_d0   <= std_logic_vector(unsigned(PC_out) + 1);

        -- Adding 1 instead of 4 because all the PC values are expected to be word addressable instead
of byte addressable
        TEMP_pcMux_d1   <= std_logic_vector(unsigned( resize(signed(MEM_im_out(15 downto 0)),
MUX_pcIn'length)) + unsigned(PC_out) + 1);-- [(pcOut + 4) + SE(IM_mem)<<2]
```

No known bugs.