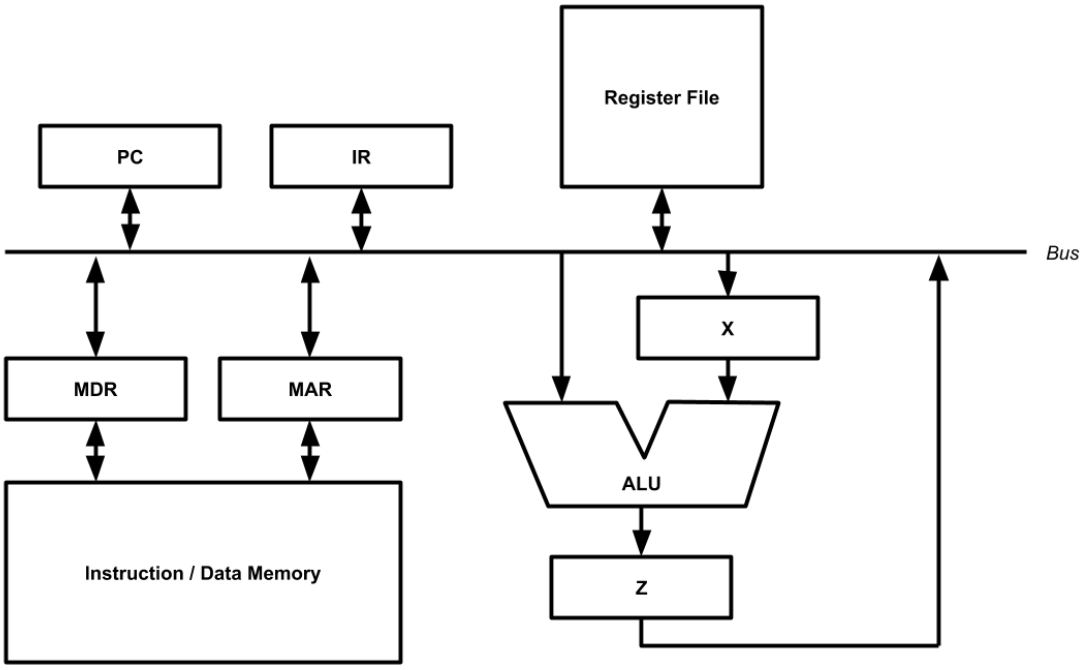
1. From last time
    a. Went over the single bus machine below



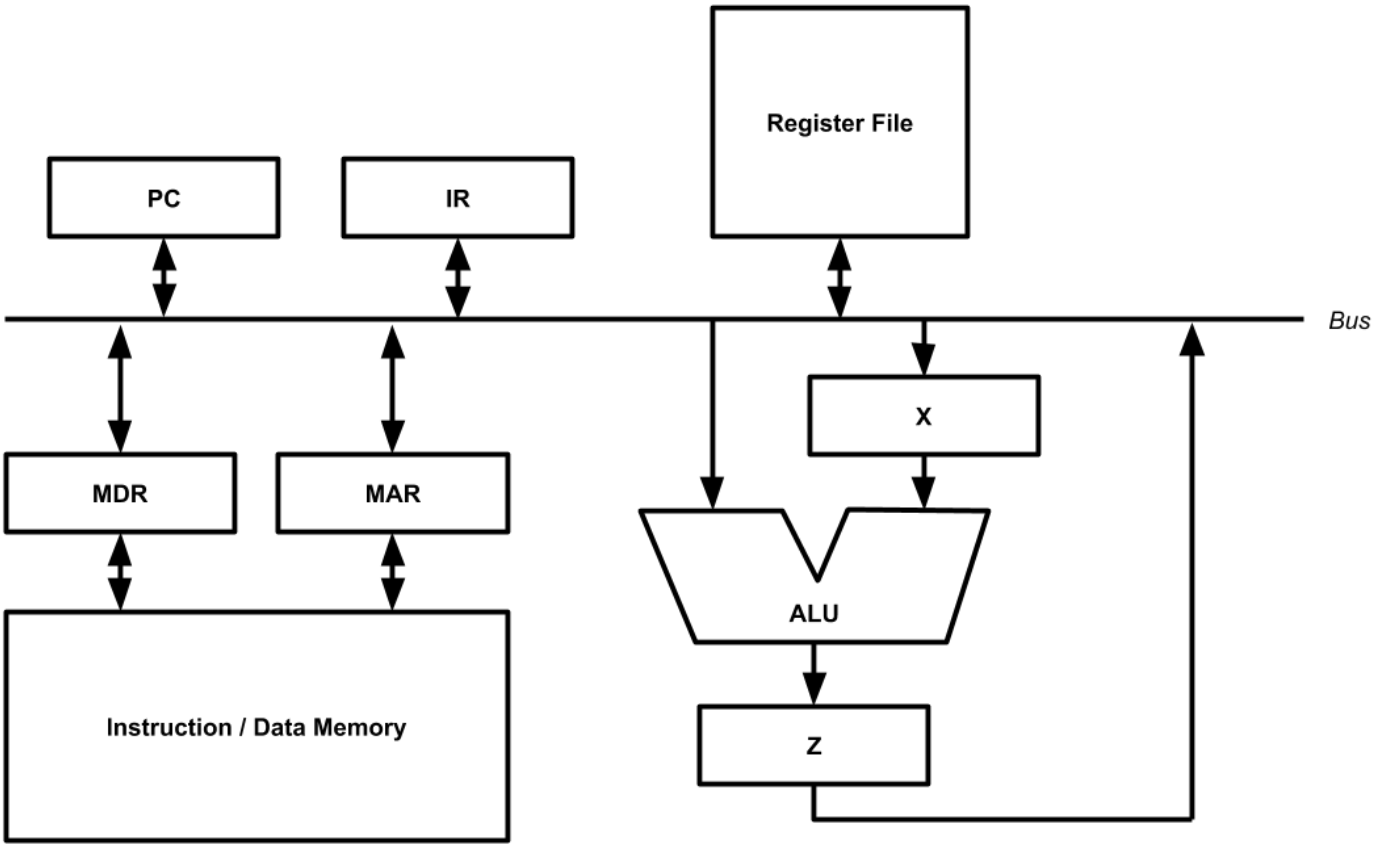
    b. Example – let's add two memory locations and place result in register file
        i. Get value at first memory location
            1. Address of first operand placed on bus from IR

            2. MAR takes in value, memory returns desired value to MDR

            3. MDR places its value on bus, X takes in value for temporary storage

        ii. Get value at second memory location
            1. Address of second operand placed on bus from IR

            2. MAR takes in value, memory returns desired value to MDR



iii. Add things together, place in register file
   1. MDR places its value on bus

   2. ALU takes in current value on bus and X, places it output in Z

   3. Z places its value on bus, register file takes in value

   4. IR places register address on bus, register file takes in address

c. Can see that one bus is a huge limiting factor
    i. Lots of contention for the bus, many things want to use it

    ii. Every time we use the bus for something else, need another clock cycle

    iii. Idea

    iv. Solution

d. Couple of reasons why we can't reduce to 1 cycle
    i. Must increment PC to reach next instruction

    ii. Complex addressing modes require multiple trips to memory

2. RISC versus CISC machines
    a. Review of material from ECS 50
    b. Back in the 1970s, had dozens of machines and instruction sets

    c. Different teams at Stanford and Berkeley looked over this

    d. RISC

3. Back to buses
    a. Key characteristic of bus

    b. Functional groups
        i. Data lines
        ii. Address lines

        iii. Control lines

    c. Types of buses
        i. Dedicated

        ii. Time multiplexed