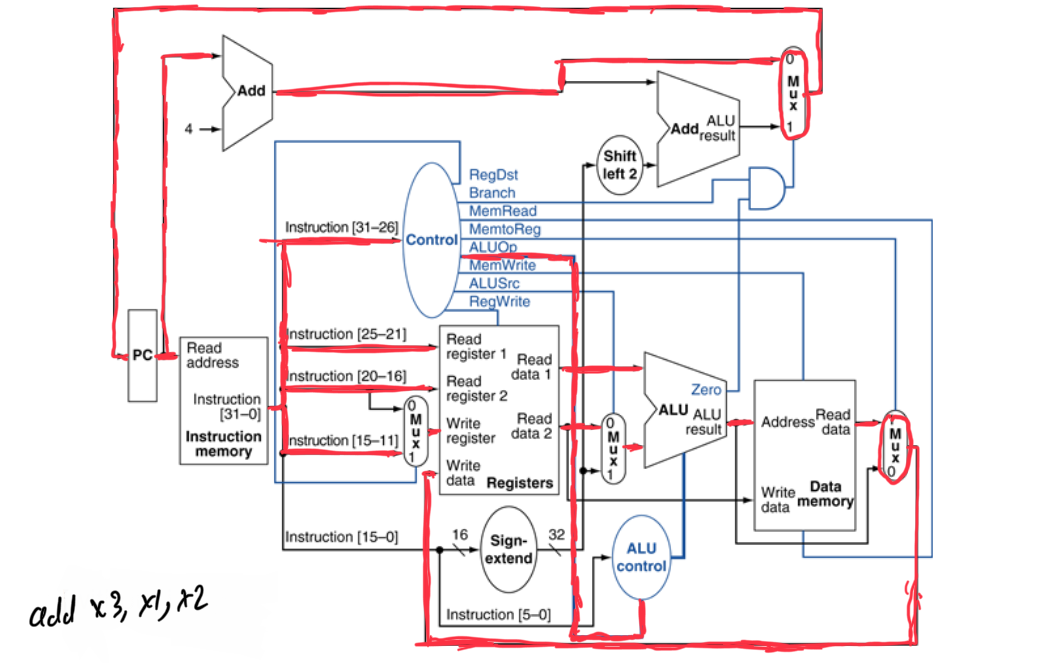
Part C:

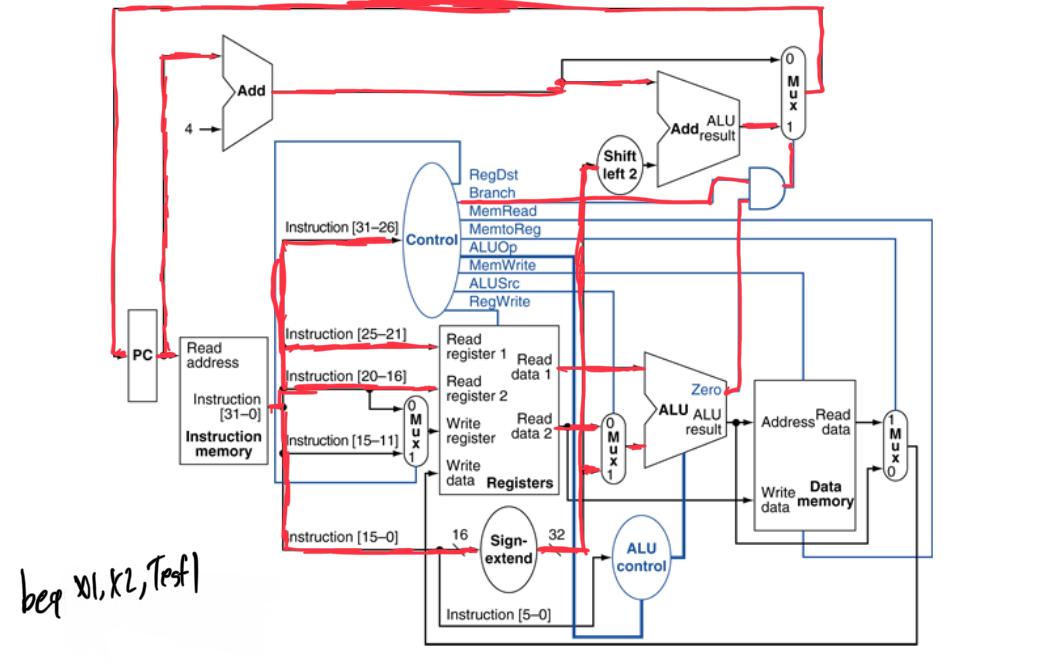
ADD Instruction: ADD x3, x1, x2 (This is a simple instruction to add two values)

The ADD instruction first goes through the PC (Program Counter), then through the Instruction memory which passes the 32-bit registers that are intimated in ADD instruction. After the instruction is passed through the Instruction memory it breaks down into three smaller registers bits to allocate structure for the Register and Control. The Control will read the instruction (in this case it’s ADD) after the control reads the instruction it will output the ALUOp code which will then continue to ALU control which will ADD the result and store the result into ALU as ALU result. When this previous process is completed the ALU result then moves to Address in Data memory and it reads it in MUX which then goes back to Registers and Writes the data back (Also know as Wd). While all this is happing, the PC goes to Add and add 4 to where this is a simple ADD instruction therefore the MUX will out put zero and go back to the PC to fetch the next instruction. Science this is a simple instruction it usually take 1 clock cycle to complete which makes adding the fastest instruction to execute.



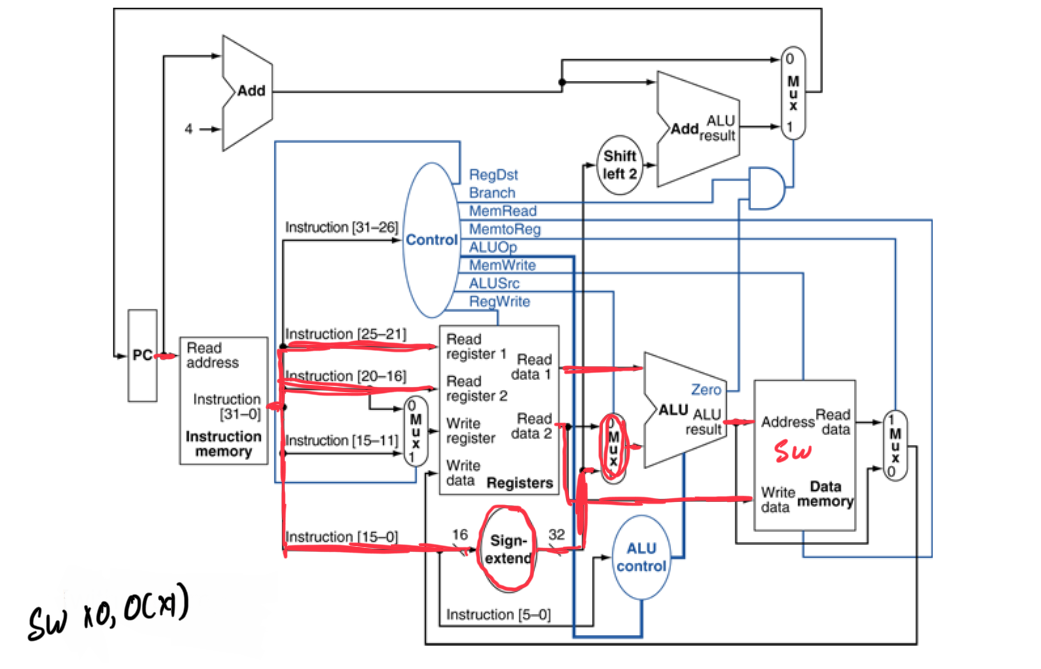
BEQ Instruction: BEQ x1, x2, Test1 (This instruction checks if x1 and x2 are equal)

The BEQ instruction first goes through the PC, then it goes through the Instruction Memory which divided the code into OP, RS, RT or 16-bit immediate. The OP code will go to the Control, RS will go to the first read register and the RT will go to second read register as well as the MUX, the 16-bit immediate will also go through the Sign Extend to extend the sign from 16-bit to 32-bit which will then Shift Left two time to check if the adder is a branch OP code. The control will then tell the ALU to subtract the two x1 and x2 register and check if it is equal to zero if it is equal the zero the value will go through an and gate which is also connected to the control which will tell us if the And Gate is branch. The previous instruction will then go through MUX and if the different through the And Gate is not zero then the instruction will come back to PC to fetch a different set of value. Completing this instruction will take 3 cycles on average.



SW Instruction: SW x0, 0(x1) (This instruction is used for Storing a value in the CPU)

The SW instruction first goes through the PC, then the Instruction Memory which will brake down into RS, RT and offset/immediate value. The RS will go through the Read Register one, Rt will go through Read Register two, and offset will be enhanced through Sign Extend. The previous step will then go through the ALU to see if any operation needs to be done and if not, it will continue to Data Memory where the value of SW will be stored. Completing SW will take up toward off 4 cycles on average.



Part E:

|  |  |
| --- | --- |
| Members | Tasks |
| Pranshu Vyas | * Worked on Part A and D off the report |
| Kunj Patel | * Worked on Part C and D off the report |
| Janki Jadeja | * Worked on Part B off the report |

Part F:

In milestone one my group has expanded their knowledge on how a CPU functions and the different parts it takes to make a low-level CPU. We also learned how a processor is designed and how different type of instructions move around the processor to output the inputted instruction. In this milestone we all learned how to use Quartus and the benefits on using a complex software like Quartus for better understanding the design of a CPU.