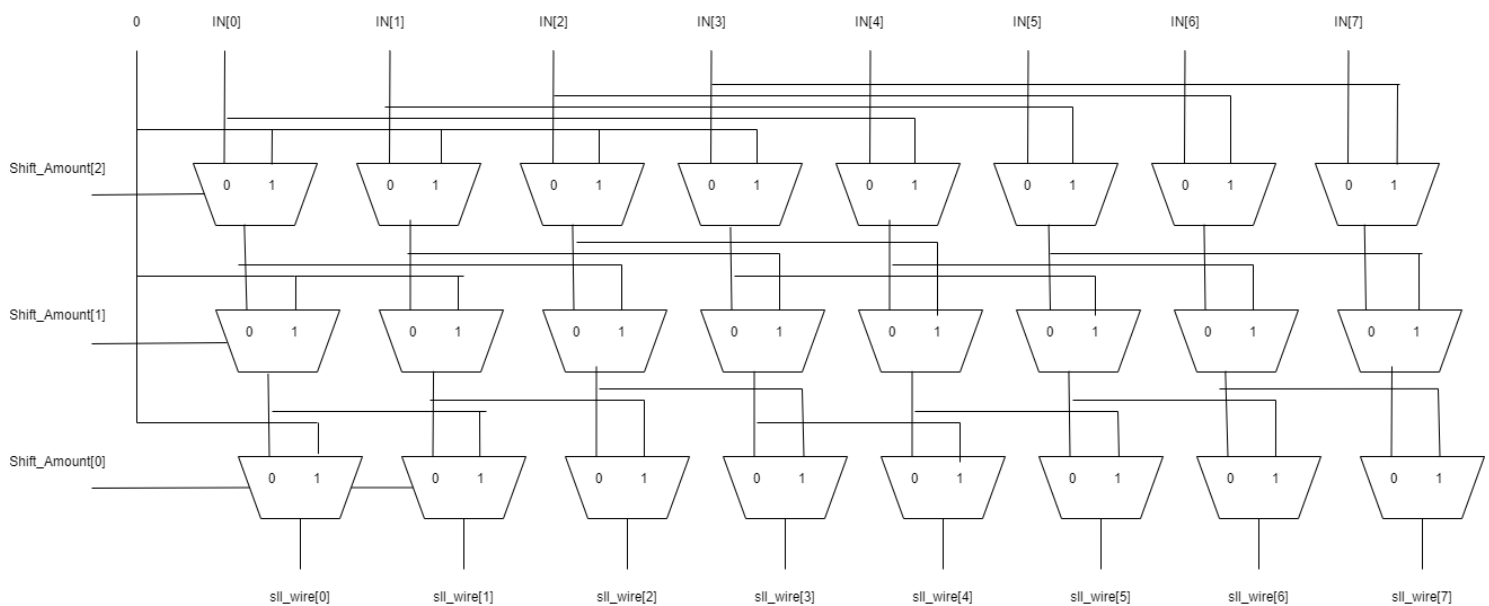# CO224 Lab5 Part5 (Bonus) Added Features

1. sll - logical shift left operation

By this method, a value stored in the specified register can be shifted by specified positions (Immediately specified) left and stored in the destination register.
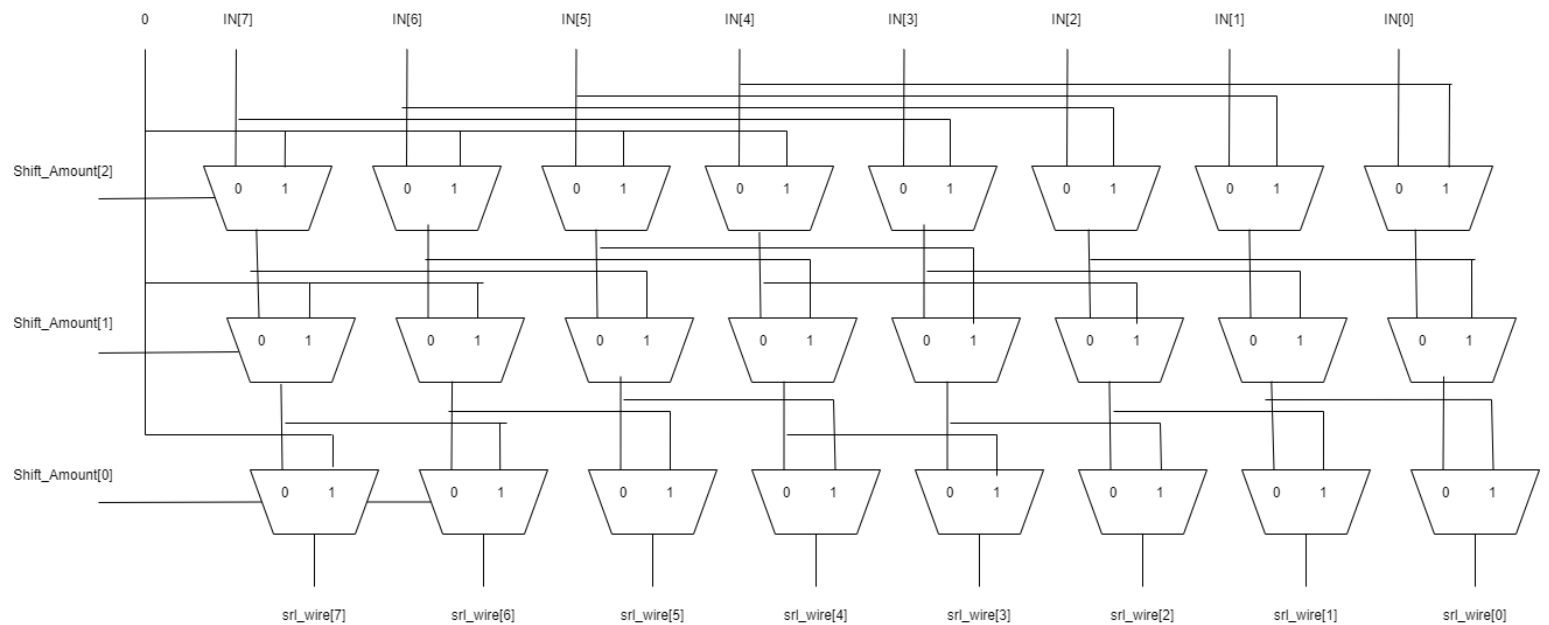
Method

We used the opcode `00001101` and the aluop 100 to implement this operation.

To implement this operation first, a module named logical_shifter8bit was declared. Using a another module named mux2X1 the behavior of a single 2X1 MUX was implanted. By using that module logical_shifter8bit was constructed forming MUXs according to the diagram above. Then a delay of 1 time unit was added to the operation.

2. srl - logical shift right operation

By this method, a value stored in the specified register can be shifted by specified positions (Immediately specified) right and stored in the destination register.
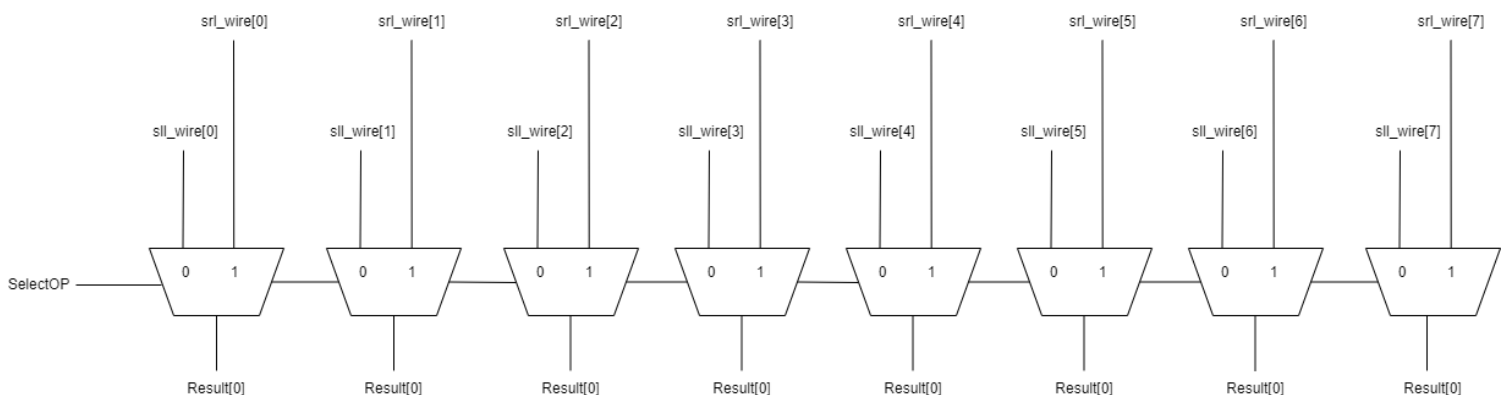
Method

We used the opcode `00001110` and the aluop 100 to implement this operation.

To implement this operation already declared logical_shifter8bit was used. Using the module named mux2X1 the behavior of a single 2X1 MUX was implanted. By using that module logical_shifter8bit was constructed forming MUXs according to the diagram above. Then a delay of 1 time unit was added to the operation.

To implement sll and srl operations we used the same ALUOP of 100 first the result of both sll and srl operations were assigned to sll_wire bus and srl_wire bus respectively then using the mux2X1 module the output Result was selected according to an extra control signal

3. bne – branch not equal operation

This instruction moves the PC value by the specified amount if the registers specified does not hold equal values

Method

A new Opcode `00001100` was introduced to do this method. The same procedure was used as beq except the inverse of the ALUZERO was taken and fetched to PC_next_select.Then the PC value was changed according to the offset defined.