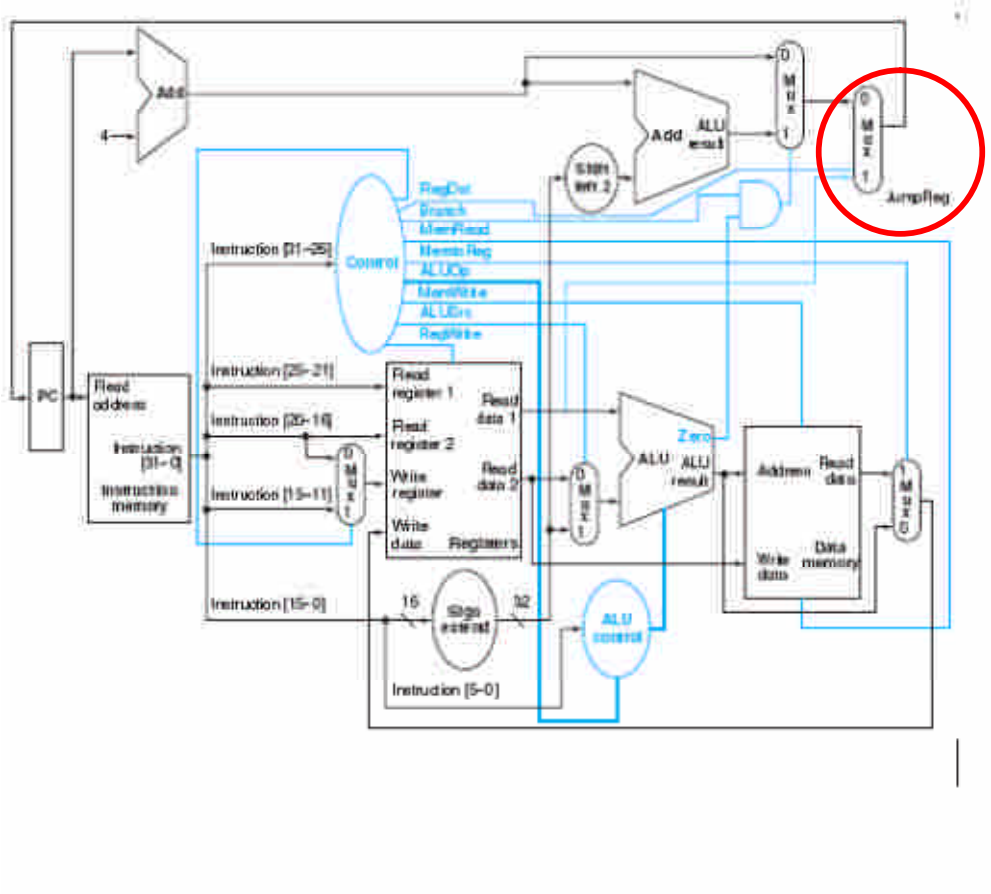
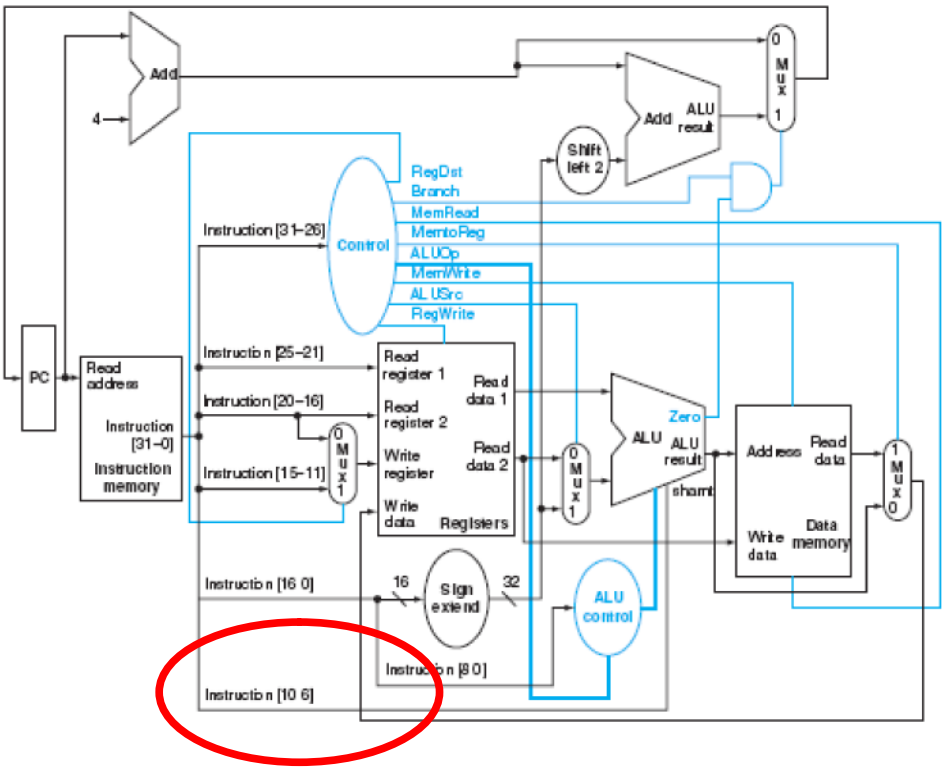
5.8 jr

A modification to the datapath is necessary to allow the new PC to come from a register (Read data 1 port), and a new signal (e.g., JumpReg) to control it through a multiplexor as shown in the following figure. A new line should be added to the truth table in Figure 5.19 on page 309 to implement the `jr` instruction and a new column to produce the JumpReg signal.



5.9 ssl

A modification to the data path is necessary (see the following figure) to feed the shamt field (**instruction[10:6]**) to the ALU in order to determine the shift amount. The instruction is in R-Format and is controlled according to the first line in Figure 5.18 on page 309. The ALU will identify the `sll` operation by the ALUop field. Figure 5.13 on page 302 should be modified to recognize the opcode of `sll` : the third line should be changed to 1X1X0000 0010 (to discriminate the `add` and `ssl` functions), and a new line, inserted, for example, 1X0X0000 0011 (to define `sll` by the 0011 operation code).



5.10 lui

Here one possible `lui` implementation is presented: **This implementation doesn't need a modification to the datapath.** We can use the ALU to implement the shift operation. The shift operation can be like the one presented for Exercise 5.9, but **will make the shift amount as a constant 16**. A new line should be added to the truth table in Figure 5.18 on page 308 to define the new shift function to the function unit. (Remember two things: first, there is no funct field in this command; second, the shift operation is done to the immediate field, not the register input.) RegDst = 1: To write the ALU output back to the destination register (`$rt`).

ALUSrc = 1: Load the immediate field into the ALU.

MemtoReg = 0: Data source is the ALU.

RegWrite = 1: Write results back.

MemRead = 0: No memory read required.

MemWrite = 0: No memory write required.

Branch = 0: Not a branch.

ALUOp = 11: `sll` operation.

This ALUOp (11) can be translated by the ALU as `shl, ALUI1,16` **by modifying the truth table in Figure 5.13 in a way similar to Exercise 5.9.**