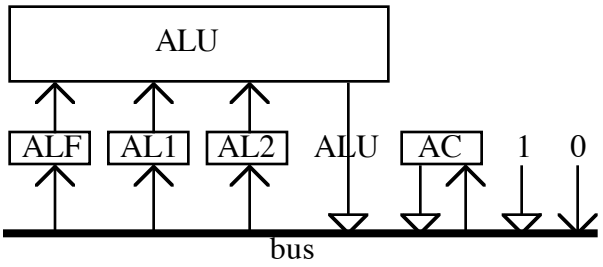
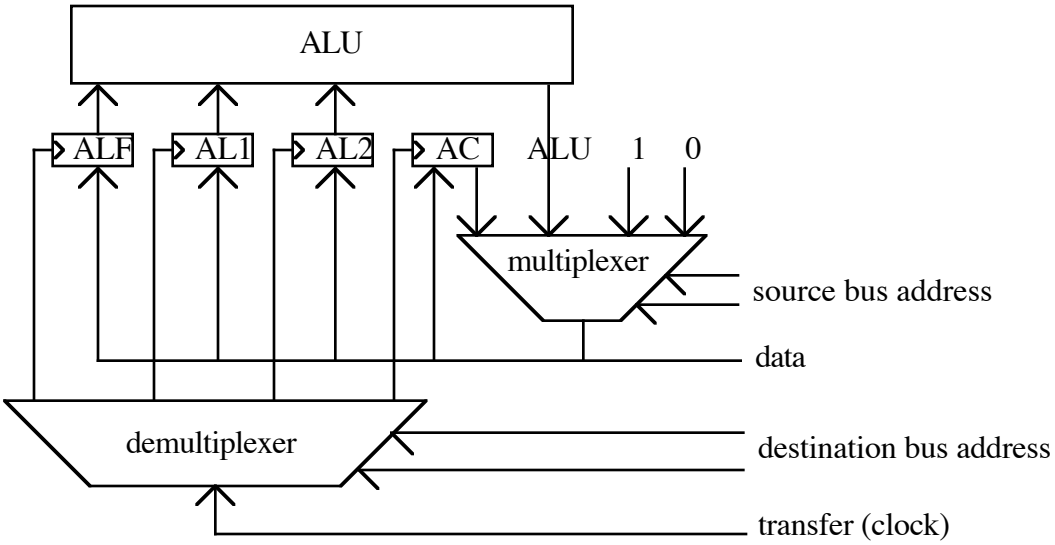
CSC258 Computer Organization Lab 3

In this lab you will construct a portion of a CPU corresponding to the following rough picture.

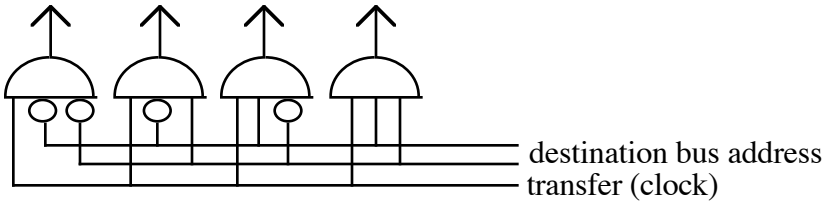


Each of the registers is a single bit. The ALU performs only two operations: AND and XOR. The bus is six bits: one data bit, plus two source bus address (sba) bits, plus two destination bus address (dba) bits, plus one transfer (clock) bit. The above picture is partly logical, and partly physical, and is missing some detail. Here is another picture, this time purely logical, with more detail.

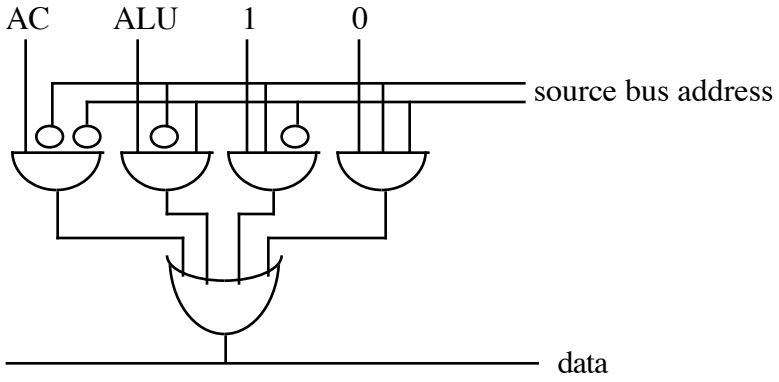


There are five inputs to this circuit; they are the two source bus address wires, the two destination bus address wires, and the transfer wire. There are five outputs; they are LEDs that enable us to see the contents of the four registers and the ALU.

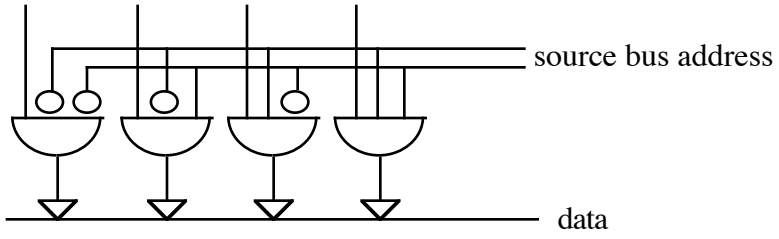
The demultiplexer contains the following gates.



Each of the four parts of the demultiplexer recognizes one address, letting the clock pulse through just when it is addressed. Physically, we can place each of these address recognizers with its register. The multiplexer contains the following gates.

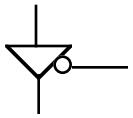


The data input into the rightmost part of the multiplexer is always 0 , so logically that AND gate can be eliminated, and the data 1 input can also be eliminated, but we keep them for physical reasons (see below). We could place each of the AND gates, which is an address recognizer, with its source, except for the fact that their outputs all go into an OR gate. An OR gate is made out of diodes, one for each input, so if we show the OR gate as diodes,

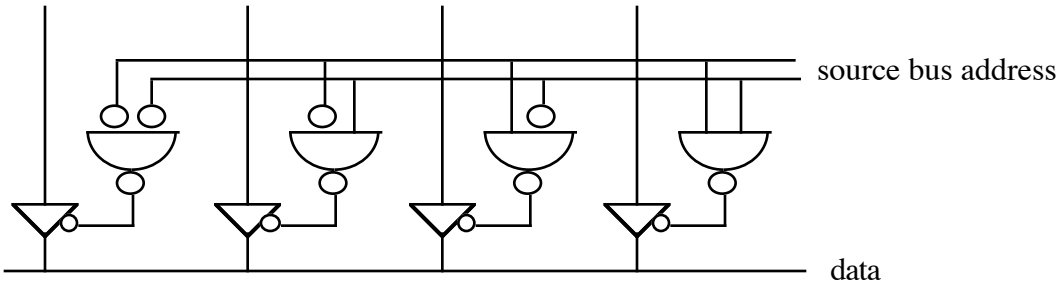


we can again place each part of the multiplexer (AND gate and diode) with its source.

In the lab, we use a so-called "3-state buffer" instead of a diode. A 3-state buffer is just a diode with an extra input, called a control. The ones we have (DM74LS125A; see the description on the course website) allow the main input to pass to the output when the control is 0 , so they are shown with a NOT gate at the control input.



To compensate, we need to put another NOT gate at the control input, as in the following picture.



Draw the complete diagram of your CPU. Give the truth table of the ALU. Write out the sequence of inputs needed to perform each of the ALU operations.