ALU design notes

8/5/2021

Now making the v2 of my alu, I realize that the previous design neglects to add inverters to the front of the and gates; nullifying the circuitry preceding it. This reveals another advantage of creating a simulation before deploying a prototype; error checking.

2/25/2021

Here is a new idea for the bit-shifter. The alu will accept operand 1 as the value to be shifted and operand 2 as the value to shift by. Starting from the lsb, a nibble of information will be fed from both op1 and op2 into an eeprom, which will be programmed with a proper output for the current power of 2. The results of each eeprom are then fed into another alu which add up the intermediary results of the operation.

I am not sure why I thought the bit shifter was a good idea for binary multiplication, it would only work for powers of two. Very embarrassing!

I wanted to design my own shift register just for fun, so I did.

In the bit shifter only four tri-state buffers are present, but as the connections are

red this symbolizes 16 tri-state buffers.

Also, although the decoders have inverted output, the design calls for non inverted output.

I could only find chips with inverted output.  
  
  
  
  
  
Strangely the chip I used for the flags register does not have any read signals, so I am going to assume that its clock signal works as its enable.   
  
Because the status of the flag register needs to be saved in the case of an interrupt, the registers output will be connected to both the control unit and to the data in portion of the data-bus.  
The ZN flags will not be separate from the CO flags. They will be on the same chip.

I am thinking of representing each module with a different color to indicate that a connection attaches to that module in that way. To distinguish nibbles vs bytes, a connection will either start red or blue and then segment, to be continued in the new color. Since multisim doesn’t allow connections between two differently colored connections, this segment will be empty space.   
  
  
  
  
Here are some color coding ideas:  
  
Green: control unit  
Dark Yellow: ALU  
purple: interrupt module  
orange: data-bus  
  
  
  
I think that I am done with the preliminary design of the ALU, I might need to show my design to a teacher or professional to check that I haven’t left anything out but so far I am feeling good about it.