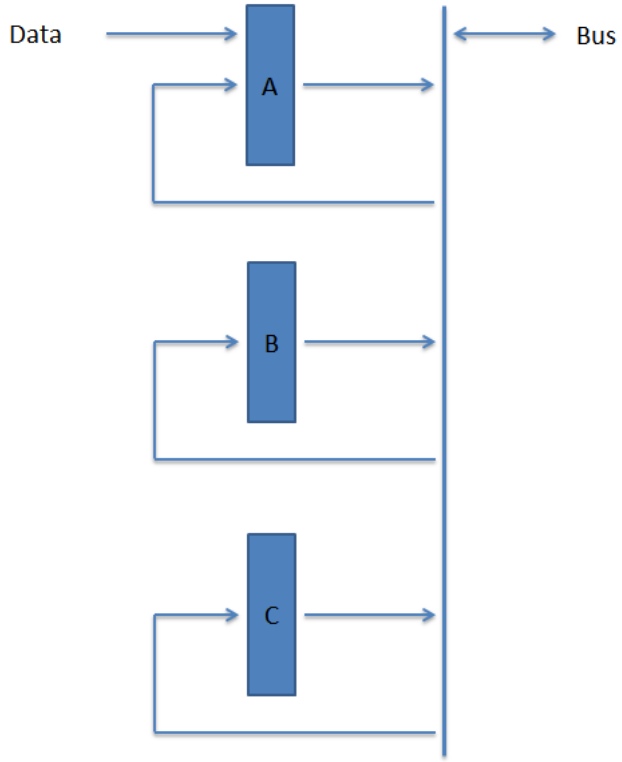
ELEC 385: Computer System Design
Homework #4
Due: February 11, 2015

Create a small system in Logisim capable of loading an immediate value into a register (register A) and transferring between any of three registers connected via a bidirectional bus. Create an input for the immediate data and an output to view the contents of the bus. A block diagram of this system is shown below.

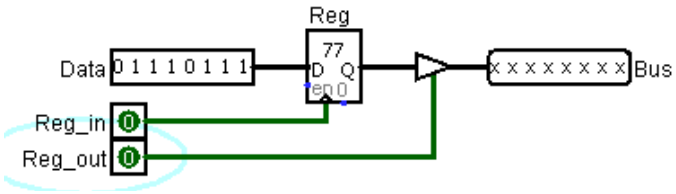


All registers are to be 8-bit. To create an 8-bit register in Logisim, select Register from the Memory folder. Click on the component and you will see the attributes in the window on the lower left of the screen. If not already selected, choose 8 data bits for the register. Now the D input and the Q output will both be 8-bit. You can use a splitter to access the individual bits, but it will often be more convenient to transfer all 8-bits together using a multiple-bit wire.

When using combinational logic (including the tri-state buffers necessary to interface each register with the bus) select the gate and place it on the canvas. Click on the gate and view the attributes. Change the # of data bits to 8. Now you can use this single symbol to perform the desired combinational logic operation on all 8 of the bits of the bus. A similar procedure can be followed in order to create multiple-bit inputs and outputs.

Now for the control. You are to have eight control inputs, five of which indicate the source of the data for the transfer (Data_2_A, Bus_2_A, A_out, B_out and C_out). The other three control inputs indicate the destination of the transfer (A_in, B_in and C_in). One source control line and one destination control line are activated in order to complete each transfer. For example, if you want to transfer data from register C to register A, C_out=1 and A_in=1 (in that order) while all the other control inputs are zero.

The following circuit diagram and function table illustrate such a transfer (on a smaller scale). Notice how the contents of the register change only when the Reg_in input goes high. At this time the register saves the current Data input. Notice that the Bus output only displays the contents of the register when the Reg_out input goes high. When the Reg_out input is low, the tri-state buffer is disabled and the xes on the bus indicate the high impedance state of the bus.



Data	Reg_in	Register(200,160)	Reg_out	Bus
0000 0000	0	0000 0000	0	xxxx xxxx
0001 0000	0	0000 0000	0	xxxx xxxx
0001 1000	0	0000 0000	0	xxxx xxxx
0001 1100	0	0000 0000	0	xxxx xxxx
0001 1110	0	0000 0000	0	xxxx xxxx
0001 1110	1	0001 1110	0	xxxx xxxx
0001 1110	0	0001 1110	0	xxxx xxxx
0101 1110	0	0001 1110	0	xxxx xxxx
0100 1110	0	0001 1110	0	xxxx xxxx
0100 1110	1	0100 1110	0	xxxx xxxx
0100 1110	0	0100 1110	0	xxxx xxxx
0100 1111	0	0100 1110	0	xxxx xxxx
0100 1111	0	0100 1110	1	0100 1110
0101 1111	0	0100 1110	1	0100 1110
0111 1111	0	0100 1110	1	0100 1110
0111 1111	1	0111 1111	1	0111 1111
0111 1111	0	0111 1111	1	0111 1111
0111 0111	0	0111 1111	1	0111 1111
0111 0111	1	0111 0111	1	0111 0111
0111 0111	0	0111 0111	1	0111 0111
0111 0111	0	0111 0111	0	xxxx xxxx

Finally, design a rigorous test for your beautiful design. Submit a screenshot of your circuit diagram (make sure all text is readable). Submit a single screenshot of the Logisim table created during your testing. And submit a text description of the testing so that I can understand what you are trying to show me with the tables.

As always, homework is to be single-sided and stapled.