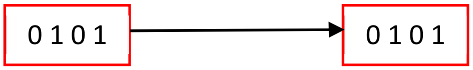
Lab 2

Building Larger Circuits

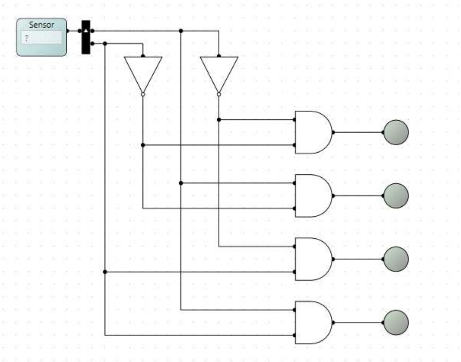
Welcome to Logic Circuit. In this lab you will be creating packages and using these to create simple but sophisticated circuits. Consider a communications line which uses serial data transmission to move 1-4bit word of data from point A to point B. In serial communications the transmission line is only 1 bit wide, so each bit will have to take its turn being transmitted from the input (left side) to the output (right side).



One way to do this is to read each bit starting from one end the input word (left word) and putting that bit onto the wire (arrow) for transmission to the right side. This is a multiplexed transmission, and is used quite often in data communications. We will create a simple simulation of this. To do so, we need to build 3 components.

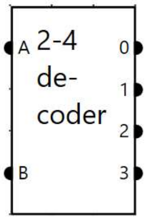
Part 1

To start, we will need to create a decoder. A decoder will take N input lines and generate 2^N control lines. If we have 2 input lines, then there are 4 possible combinations with 2 bits, thus we end up will 4 outputs. Open Logic Circuit and start a new project. Under Circuit select new Logical Circuit. A new window will appear, under the Main Circuit in the left. Here you will create your Decoder. To the right is a simple decoder, once you run the cct. The input sensor will activate exactly 1 of the output LEDs, give it a try.



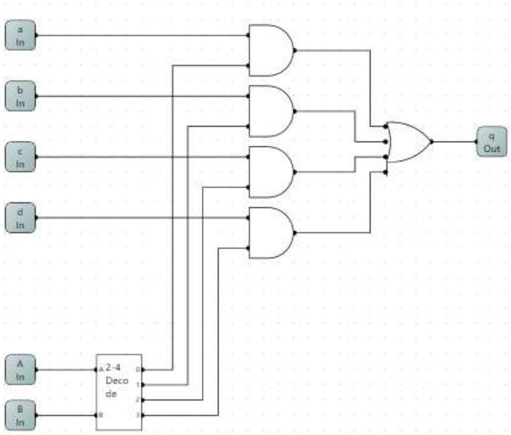
Part 2

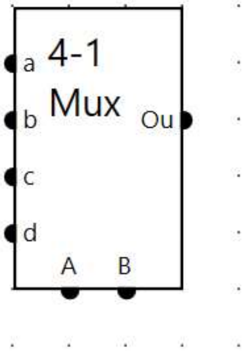
Modify the circuit replacing the Sensor and splitter with discrete inputs, labelled A and B. Replace the LEDs with outputs, named 0,1,2,3. If completed correctly, your package should look similar to the following. When placing inputs and output logic circuit gives the option as to which side of the package it is to be placed.



Part 3

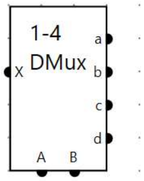
We now need to create a 4 to 1 multiplexer. A multiplexer will take N inputs and connect each input to the output based on the Decoder input. Thus when the decode input is 0, then (a) will appear on the output q. Create a new logical circuit same project. Create inputs and outputs to match the below circuit. You will need to edit each input and output. You can set where the inputs are to appear on the package, typically inputs appear on left and bottom, outputs on the right.





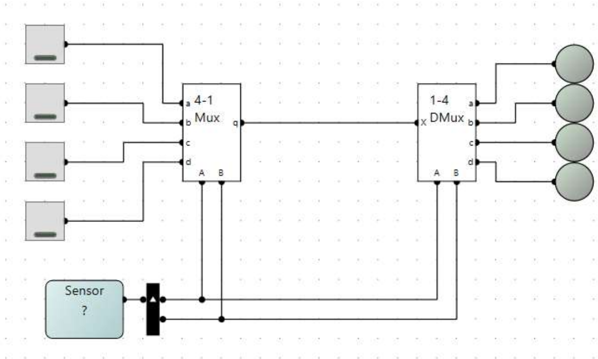
Part 4

A De-Multiplexer, DMux is the exact opposite of a Mux. In essence, you will take 1 input and split it into 4 outputs, where each of the outputs is controlled by the A and B control lines. This is an exercise left to you. Your TA will help. Hint: use the decoder you made. The package should look as follows.



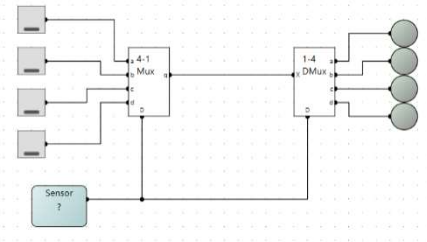
Part 5

Let us now put the simulation together. Create another logical circuit, name it Lab 2. Using the components you created, duplicate the circuit below. Code the Sensor to run as a series (see lab 1), cycling from 0 to 3. If done correctly, each output led will match the input switch when the sensor generates the correct time slice.



Part 6

The sensor must be split, to allow the discrete signals A & B to be used. Modify the decoder, Mux and D-Mux so that the Splitter is part of the decoder. In this format the Sensor will not need to be split, and should clean up your circuit considerably. For clarity, it is good practice not to have an overabundance of wires going everywhere. The use of the splitters can greatly reduce the clutter of circuits, and is the preferred method.


4-1
Mux
1-4
DMux
Sensor
?