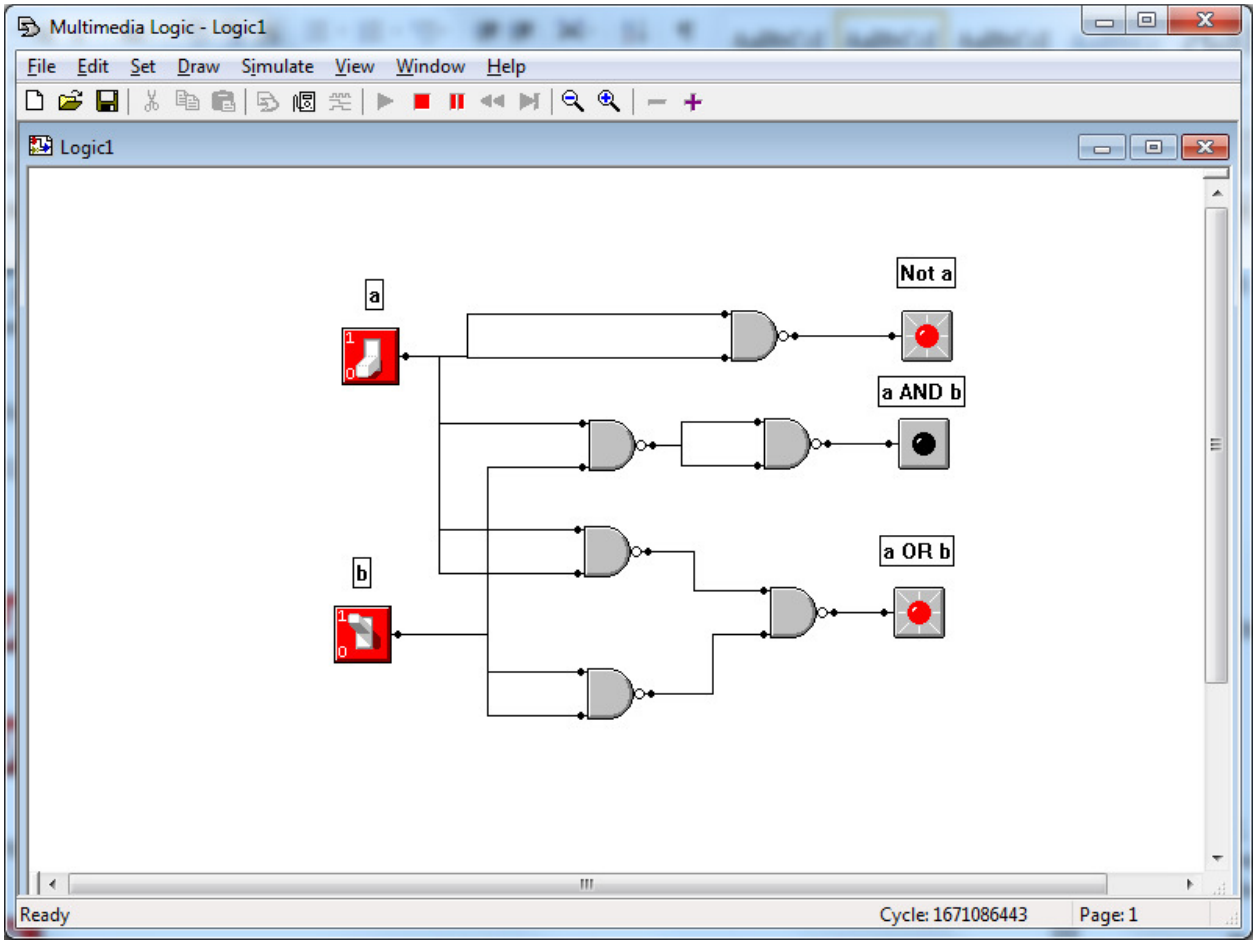
Comp 255 - Computer Organization Name: _____
Lab #4 – Intro to Building and Testing Circuits
February 5, 2013 Grade: _____/10

I. An Introduction to Circuit Schematics Using MM Logic

Introduction: MMLogic is an interactive program that allows a user to build and test digital circuits. From a palette of components (gates, switches, LEDs etc), the user can select the components needed for a circuit, wire (connect) the components together and simulate (test) the circuit.

Do the following

A. Using only NAND gates construct a NOT circuit, an AND circuit and an OR circuit (see below) with common inputs. Test your circuit. When you are satisfied that it's correct, copy the image (Alt/PrintScrn) to a Word document (to be handed in).



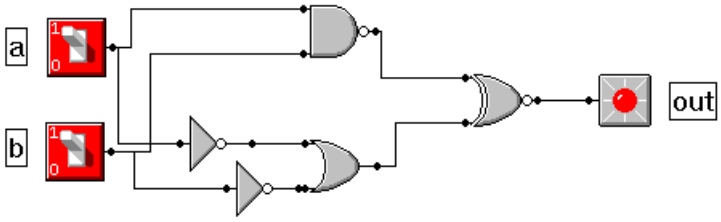
B. Sum of Products Circuit: Using only AND, OR and NOT gates implement (and test) the following truth table as a sum of products circuit.

a	b	c		f(a,b,c)
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		0
1	1	1		1

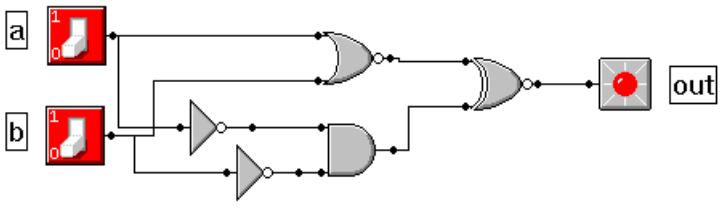
When you are satisfied it's correct, copy the image to your Word document (to be handed in). You will need to carefully place your components so as to minimize the tangle of wires. Note that MMLogic allows 3 and 4 input AND and OR gates.

C. MMLogic Proofs: The two-input *xnor* function returns 1 if and only if both inputs are the same; in other words it tests for *equivalence*. This *equivalence testing property* can be used to prove DeMorgan's Laws (see below) since no matter what the values of a and b, the output is always 1. Observe the two circuit forms of DeMorgan's laws are the two inputs to the *xnor* gate; if the circuit always returns one, the two sub-circuits are equivalent.

a. $\overline{a \cdot b}\ xnor \left(\bar{a} + \bar{b}\right)$



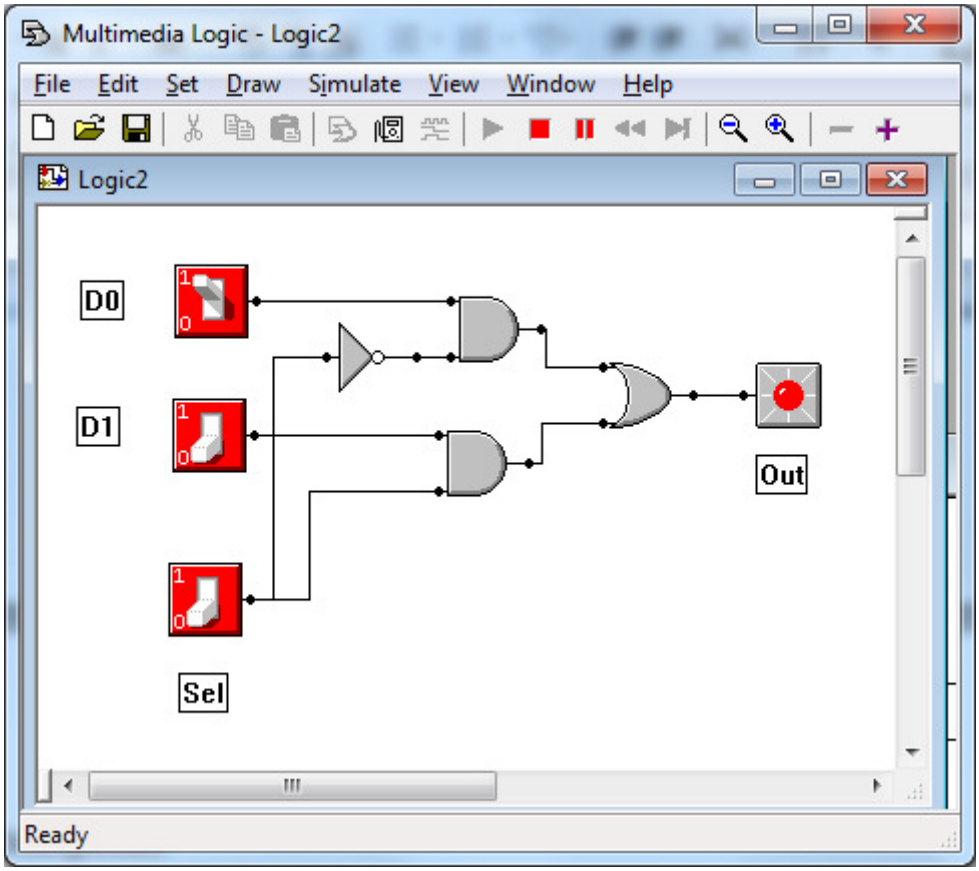
b. $\overline{a + b}\ xnor \left(\bar{a} \cdot \bar{b}\right)$



While it's not difficult to show that $a \cdot \bar{b} + \bar{a} \cdot b$ is equivalent to $a\ xor\ b$ (the former is just the *sum of products* version), can you show that $(a + b) \cdot \left(\overline{a \cdot b}\right)$ is also equivalent to $a\ xor\ b$ by construction an equivalence testing circuit similar to the one's above?

Test your circuit and if it works copy the image to your word document to be handed in.

D. Multiplexers! A multiplexer is sort of like a "switch" with the Selection input (Sel) determining which of the data inputs, D0 or D1, is passed on to the output Out (see below). If Sel = 0 then D0 is passed thru (so Out = D0), if Sel = 1 then Out = D1. This is a 2-to1 MUX.



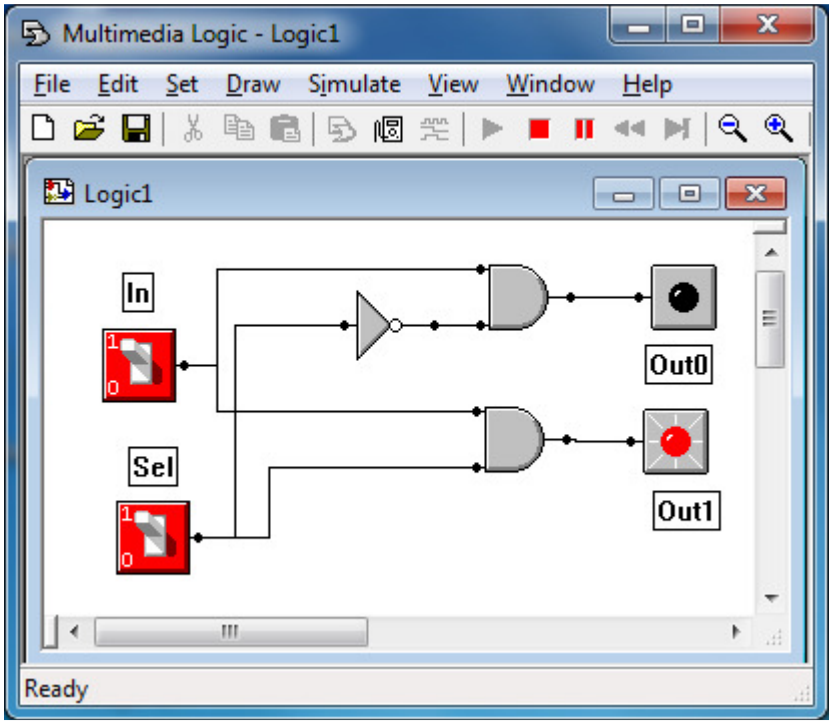
Based on the 2-to-1 MUX construct and test a 4-to-1 MUX where two Selection switches (Sel 0 and Sel 1) select which of four data inputs (D0, D1, D2, D3) to pass through to the single output Out. This will require 6 switches (labeled D0 – D3, Sel0, Sel1), 1 LED, four 3-input AND gates, a 4-input OR gate and two NOT gates.

Test your circuit and if it works copy the image to your word document to be handed in.

E. Cascading Multiplexers: A 4-to-1 MUX can also be constructed out of three 2-to-1 MUX (see above) properly cascaded together. Think how this would be done, construct the circuit, test it and copy and paste the image to your word document.

Test your circuit and if it works copy the image to your word document to be handed in.

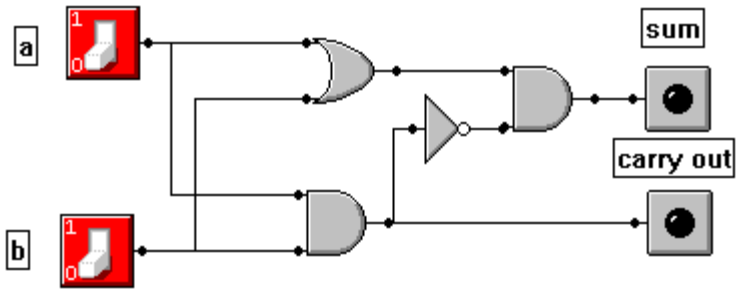
F. A De-multiplexer (aka DEMUX) takes one input signal (In) and routes to one output lines (Out0, Out1) depending on the setting of the selection switches (Sel). Below is an example of a 1-to-2 DEMUX.



Based on the 2-to-1 DEMUX construct and test a 4-to-1 DEMUX where two Selection switches (Sel0 and Sel1) selection which of four data outputs (Out0 – Out3) receive the input (In). This will require four 3-input AND gates and two NOT gates.

Test your circuit and if it works copy the image to your word document to be handed in.

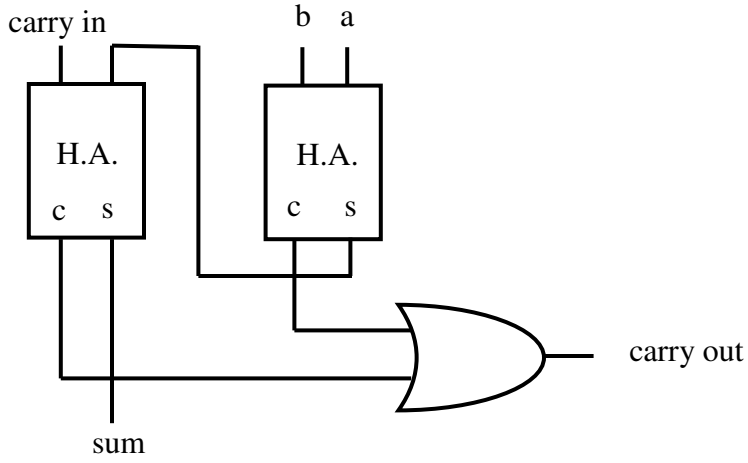
G. Build and test a Half Adder. Note that the sum circuit is $(a+b)\cdot(\overline{a\cdot b})$; the carry out is just $a\cdot b$. Note the common use of an AND gate for both sum and carry out.



Observe that

```
a     0     0     1     1
b   + 0   + 1   + 0   + 1
      --    --    --    --
      00    01    01    10
```

H. A Full Adder: A full adder can be had by cascading two Half Adders together (see below)



An optional circuit for a Half Adder can be had by using an XOR gate for the sum and an AND gate for the carry out. Use this configuration to construct a Full Adder (In part C we proved $(a+b)\cdot(\overline{a\cdot b})$ is also equivalent to $a\,xor\,b$)

Given the half adder circuit (above), cascade two together using the above diagram and test that you have a working full adder.

Test your circuit (there are 8 cases) and if it works copy the image to your word document to be handed in.