Math 310
Circuit Examples

Recall: Last class, we discussed how to build a circuit called a "Half Adder". This circuit takes as input two single binary bits, and outputs both the sum of these bits (in base two). The following table shows the input and output values. Here, x and y are the input bits, and s and c are the "sum" bit and the "carry" bit, respectively.

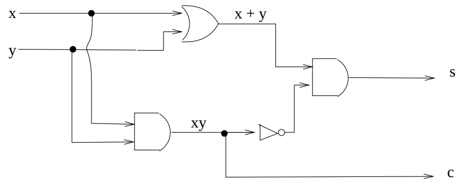
x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

From the table, we see that the Boolean expression for c is xy, and the Boolean expression for s is $x\overline{y} + \overline{x}y$.

However, if we rewrite the expression for s using Boolean Identities, we can get a simpler circuit for the Half Adder (the expressions given above would require 6 gates). As an exercise, use the table of Boolean Identities to verify each step in the following derivation for a new expression for s:

$$s = x\overline{y} + \overline{x}y = (x+\overline{y})(\overline{x}+y) = x\overline{x} + \overline{xy} + xy + y\overline{y} = \overline{xy} + xy = (\overline{x}+\overline{y})(x+y) = (x+y)\overline{xy}$$

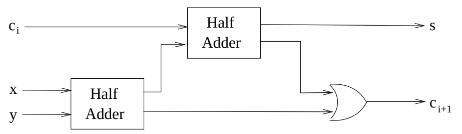
The following circuit diagram represents the Half Adder:



Next, the "Full Adder" is a circuit that takes as input two single bits x and y plus a carry bit, and c_i (which we think of as being from a previous computation) and outputs the sum of these three bits (base two) s and a new carry bit c_{i+1}. The following table shows the values for this computation.

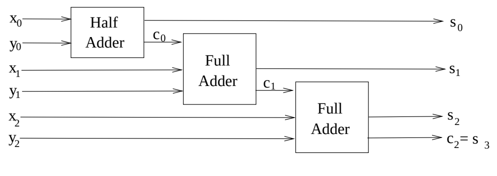
x	y	c_i	s	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Although we could write the Boolean Expressions for s and c_{i+1} and build a circuit for the Full Adder directly from the table given above, we will take an alternate approach and make the Full Adder by wiring together two Half Adders and an OR gate, as shown in the diagram below.



To convince ourselves that this works, notice that to compute the "sum" s for the Full Adder, this circuit first finds the sum of the input bits x and y using the leftmost Half Adder and then adds this result to the previous carry bit c_i using the second Half Adder. Similarly, to compute the new "carry" bit, the OR gate checks to see if there was a carry from either of the two Half Adders in the circuit, yielding the new carry bit c_{i+1}.

Now that we have constructed the Full Adder, we can build a circuit that adds binary strings of any desired length by wiring together A Half Adder, followed by as many Full Adders as we need. For example, the following circuit computes the sum of two binary strings of length 3 ($x_2x_1x_0$ and $y_2y_1y_0$)


x_0
y_0
Half Adder
c_0
s_0
x_1
y_1
Full Adder
c_1
s_1
x_2
y_2
Full Adder
s_2
c_2 = s_3