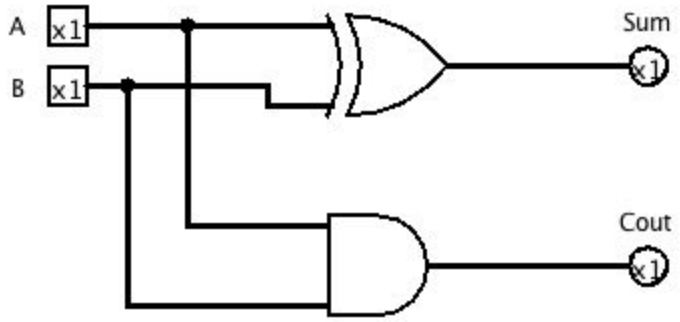
William “Chris” Fenton
10/8/15
Digital Logic Lab 2

1.1)

A	B	Sum	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1.2)
Sum of the half adder: A XOR B
CarryOut: A AND B

1.3)



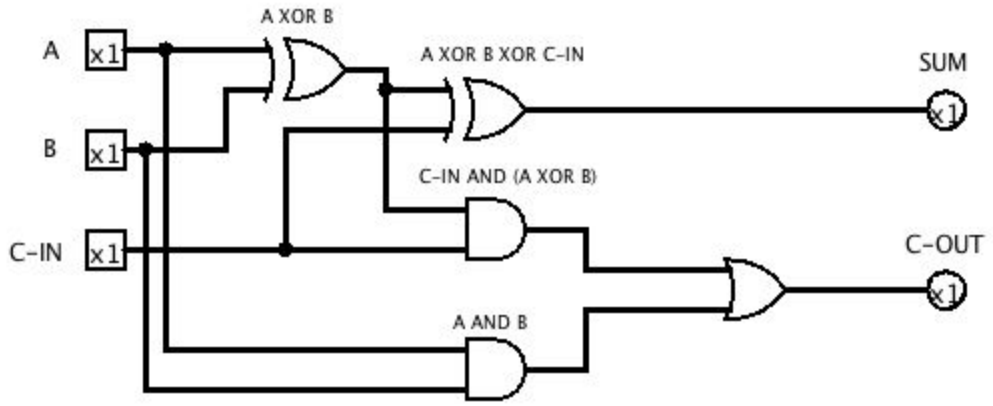
2.1)

A	B	C-IN	SUM	C-OUT
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

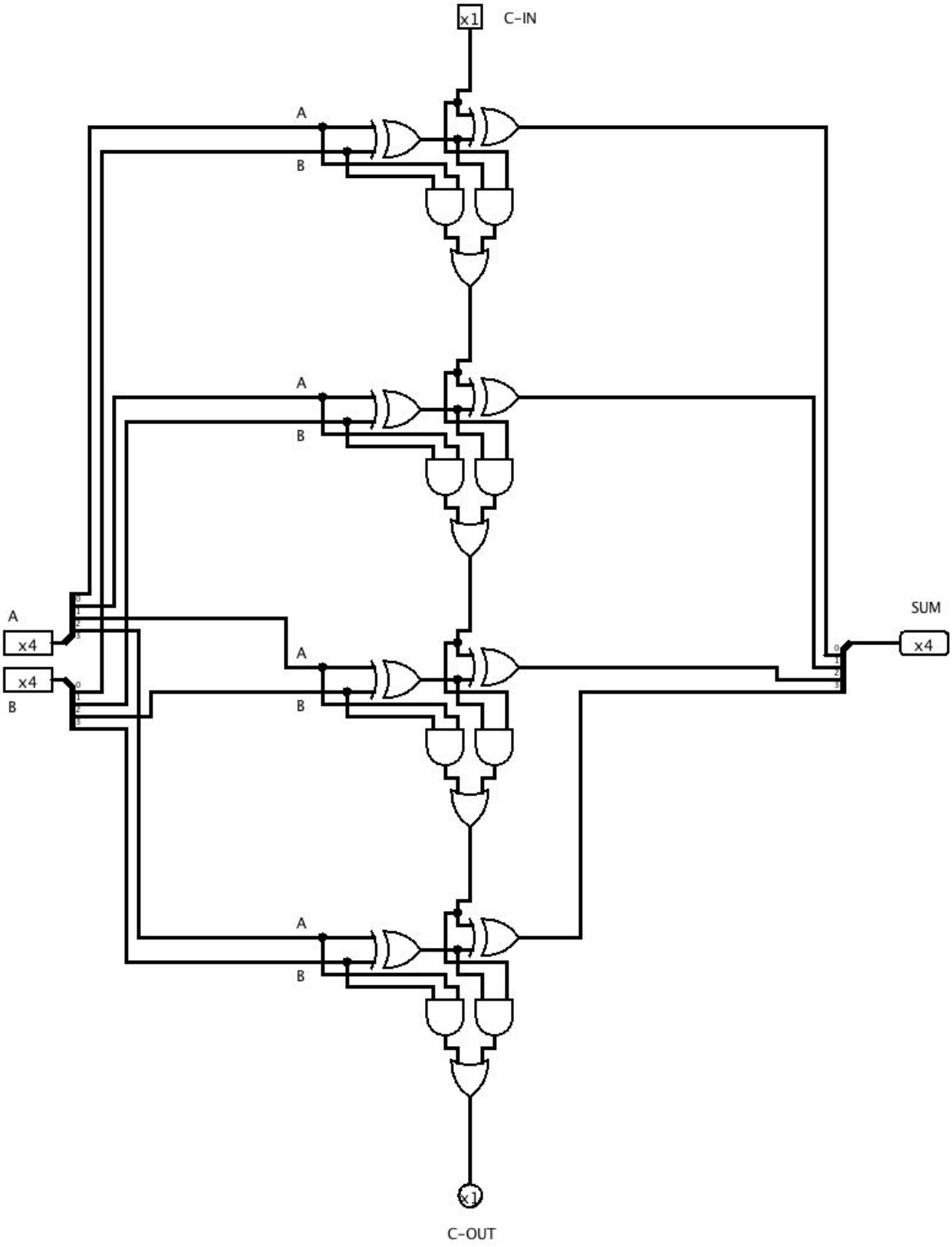
2.2)

SUM = A XOR B XOR C-IN
C-OUT = (C-IN AND (A XOR B)) AND (A AND B)

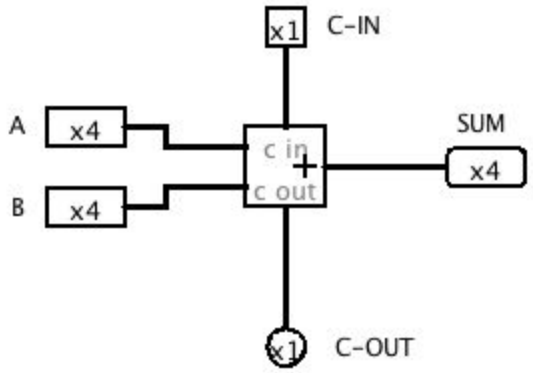
2.3)



3.1)



4.1)



Questions about a 4-bit Adder

What is the range of unsigned numbers that you can represent in 4 bits?

0 - 15

Fill out the following table of sums, carry, and borrow that your 4-bit adder circuit will give. Assume unsigned representation of numbers in 4 bits.

Binary A Input	**Binary B Input**	**Binary Sum**	**Decimal A Input**	**Decimal B Input**	**Decimal Sum**	**Carry**
0000	0111	0111	0	7	7	0
1100	0101	0001	12	5	17	1
0101	0101	1010	5	5	10	0
1111	1111	0000	15	15	30	1
0010	0110	1000	2	6	8	0

Assuming unsigned 4-bit representation of numbers, under what conditions does adding produce a result that is not meaningful with respect to normal addition and the constraint of only 4 bits to hold the sum?

When the any of the inputs or the sum are outside the range (0-15).

What does the carry out pin signify?

If the number is out of the range (greater than 15 for 4 bits)

Assuming unsigned 4-bit representation of numbers, what does your 4-bit adder actually produce if you try to add two numbers whose sum exceeds the 4-bit range of values? Give an arithmetic expression for the unsigned value of the sum bits in terms of x and y input values (use the modulus operation - mod; look for examples in your discrete math book).

It wraps around. SUM = x + y (mod 16)