Recitation #6 Solutions

1.

X	Y	Z	Y^Z	X^(Y^Z), lhs	X^Y	(X^Y)^Z, rhs
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	0	0	0
1	1	1	0	1	0	1

Therefore, XOR is commutative

2. Express a binary number of the form $B = \cdots r_1 r_0 100 \dots 00$, where there are any number of zeroes (including no zeroes) before the first 1 and any number of binary digits to the left of this first 1. In order to determine the 2's complement of B, first invert all bits to get the 1's complement:

$$B_{1s} = \cdots \overline{r_1}\overline{r_0}011 \dots 11$$

Last, add 1 to this number to get the 2's complement:

$$B_{2s} = \cdots \overline{r_1}\overline{r_0}100 \dots 00$$

From here, it can be observed that the only difference in the number B and its 2's complement representation is that all of the digits to the left of the first 1 are negated. Every number can be expressed in the form given to B (with the exception of zero, which represents itself in 2's complement).

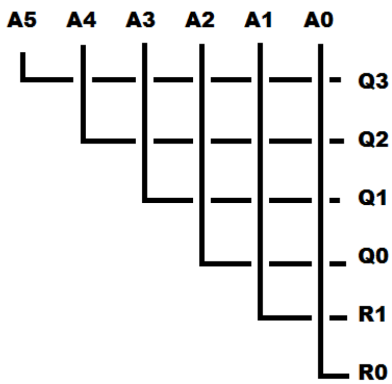
3. Overflow occurs when two numbers of the same sign are added and the sum has a different sign than both of the addends. Let x_n and y_n represent the signs of two binary number addends in 2's complement X and Y, respectively, let c_n represent the data being carried in for addition and, let s_n and c_{n+1} represent the sum and carry-out produced from this addition.

x_n	y_n	c_n	s_n	c_{n+1}	**Examining x_n and y_n in relation to s_n**	c_n ^ c_{n+1}
0	0	0	0	0	Both addends are positive but sum is positive; no overflow.	0
0	0	1	1	0	Both addends are positive, sum is negative => overflow	1
0	1	0	1	0	Addends have opposite signs and cannot overflow	0
0	1	1	0	1	Addends have opposite signs and cannot overflow	0
1	0	0	1	0	Addends have opposite signs and cannot overflow	0
1	0	1	0	1	Addends have opposite signs and cannot overflow	0
1	1	0	0	1	Both addends are negative, but sum is positive => overflow	1
1	1	1	1	1	Both addends are negative and sum is negative; no overflow	0

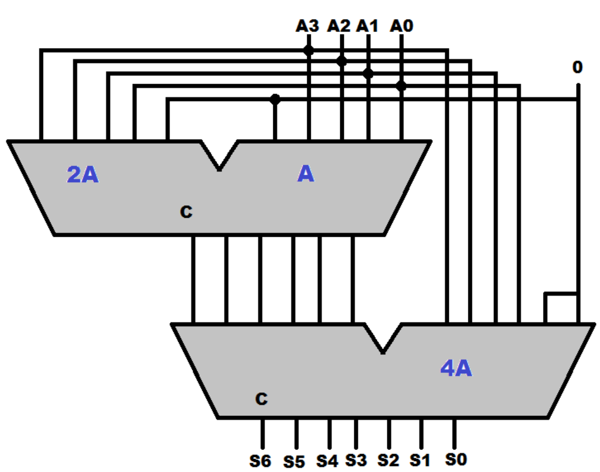
4. a. $0\ 10000100\ 01010\ 000000000000000000_2$ (IEEE754 single)
   b. 13.1875 → 1101.0011_2 → $0\ 10000010\ 1010011\ 0000000000000000_2$ (IEEE754 single)
   c. $1\ 10001000\ 1\ 0000000000000000\ 000000_2$ (IEEE 754 single)
   d. 0.8 = $0.\overline{1100}$ → $0\ 01111110\ 10011001100110011001100_2$
5. a. $-1 * 2^4 * 1.011_2 = -1 * 16 * 11/8 = -22_{10}$
   b. $-1 * 2^{-5} * 1.101_2 = -1 * 1/32 * 13/8 = -13/256 = -0.05078125_{10}$
6. $BEC00000_{16} = 1\ 01111101\ 1\ 0000000000000000000000_2 = -1 * 2^{-2} * 1.1 = -3/8 = -0.375_{10}$

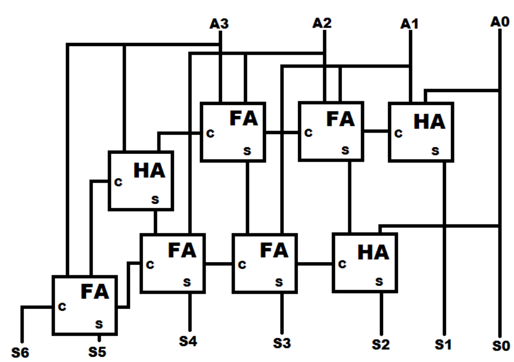
7. $101.1_2 \rightarrow 0\ 10000001\ 011\ 00000000000000000000_2$

8. The quotient is a simple bit shift from bits 5-2 to bits 3-0 and the remainder is the last two bits. The quotient is therefore a 4-bit number and the remainder is a 2-bit number.
For an n-bit number and division by 2^m, the remainder is m bits and the quotient is the remaining n-m bits.



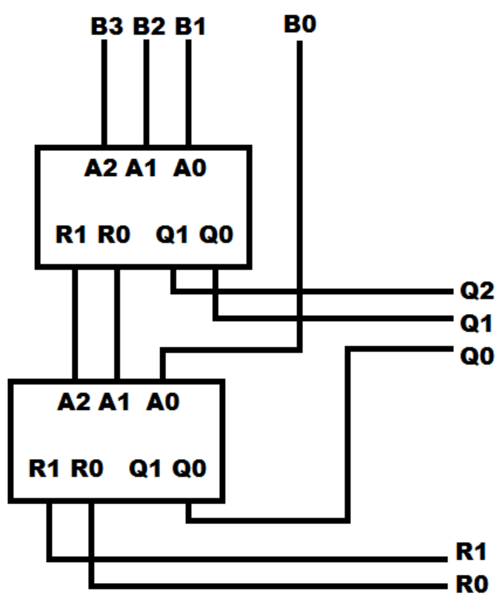
9. Design a circuit that will take a four-bit input A and calculate 7*A using only full-adders and half-adders. (Hint: start with multiple-bit adders and then replace each component with its underlying components)

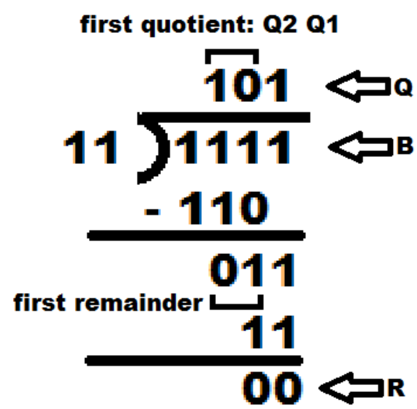




A Full-adder with an input fixed at zero functions the same as a Half-adder with the remaining inputs. A full-adder with two inputs fixed at zero functions as a wire (S is always equal to the input and C is always zero).

10. To illustrate the complexity of division by a number that is not a power of 2, suppose that we wish to take a three-bit value A and determine the quotient and the remainder of division from the operation A/3. What is the minimum number of bits required to express the quotient and the remainder? Design a circuit diagram for this circuit with AND gates, OR gates, and NOT gates. Show how this circuit can be used to determine the quotient and remainder with a four-bit input B and the operation B/3.





Demonstrating the result of division with B = 1111_2 produces the quotient Q = 101_2 and remainder R = 0. The output Q1 from the second division module is always zero because the largest value of its input is 101, which produces a maximum quotient of 1.