Problem 1: Rotate

Suppose there are 16 ordered input wires representing an integer in binary, \( X = X_{15} \ldots X_0 \), and depending on a control wire \( S \), one wants to build a circuit that optionally shifts them \( d \) places to the left, wrapping around - this is called a rotation. Specifically, if \( S \) is 0, then for each of the 16 output wires, the output wire \( Y_i = X_i \), but if \( S \) is 1, then \( Y_i = X_{(i-d) \mod 16} \). So, for example, if \( n = 5 \), then if \( S \) is off \( Y_2 \) is \( X_2 \), otherwise it is \( X_{13} \).

(a) Using any gates we have encountered, build a circuit for the \( Y_i \), that takes as input \( S \), \( X_i \), and \( X_{(i-d) \mod 16} \).

(b) Since you now know how to build a circuit that rotates 16 bits by \( d \) places, let’s call this gate \( \text{ROT}_{16}(d) \). Using rotation gates, show how to build a circuit that takes as input the 16 bit binary integer \( X \) and a 4-bit binary representation of a number of places to rotate, \( s = s_3 \ldots s_0 \), and outputs the shifted binary number \( Y \). (That is, if \( s = 0000 \), then the circuit should output \( Y = X \), and if \( s = 0101 \) then the circuit should output \( Y \) rotated 5 places.

Problem 2: Triangular truth

Using 2-input OR and AND gates, show how to construct a circuit that takes as input two \( n \)-bit inputs, \( X_0 \ldots X_{n-1} \) and \( Y_0 \ldots Y_{n-1} \), that outputs 1 if and only if there is some pair of input bits \( X_i, Y_{j \geq i} \) that are both on. That is, if \( X_0 \) is on it should output 1 if any bit of \( Y \) is on, but if only bit 5 of \( X \) is on, it should output 1 only if some bit \( Y_5, Y_6, \ldots \) is on.

You may find that the number of gates in your circuit is approximately \( 2n^2 \). Can you do this using only \( 2n \) gates?