1. What is \(~(0xA) \land 0xD\), where \(~\) is bitwise NOT and \(\land\) is bitwise XOR? Provide your result in both binary and hexadecimal.

\[
\begin{array}{c}
0xA \Rightarrow 0b1010 \\
\sim(0xA) \Rightarrow 0b0101 \\
0xD \Rightarrow 0b1101
\end{array}
\]

Result in binary (0b): \textcolor{red}{0b1000}

Result in hexadecimal (0x): \textcolor{red}{0x8}

2. What is 15 in 8-bit 2’s complement notation? What is \(-22\) in 8-bit 2’s complement notation? Show how to compute 15–22 using 2’s complement addition. What is the result in 8-bit 2’s complement notation?

15: \textcolor{red}{0b00001111}

-22: \textcolor{red}{0b11101010}

\textcolor{red}{0b11111001} \Rightarrow \text{negate all bits and adding 1: 0b00000111 = 7}

Thus, \textcolor{red}{0b11111001} = \textcolor{red}{(-7)}

15 in 8-bit 2’s complement notation (0b): \textcolor{red}{0b00001111}

-22 in 8-bit 2’s complement notation (0b): \textcolor{red}{0b11101010}

15–22 in 8-bit 2’s complement notation (show your work) (0b): \textcolor{red}{0b11111001}

3. You are given a Bluespec function \texttt{add4} that adds two 4-bit numbers with carry-in, shown below. Write a function \texttt{sub4} that uses \texttt{add4} to subtract two 4-bit numbers.

\begin{verbatim}
function Bit#(4) add4(Bit#(4) a, Bit#(4) b, Bit#(1) c_in);

function Bit#(4) sub4(Bit#(4) a, Bit#(4) b);
// Fill in you code here
    return add4(a,\sim{b},1);
endfunction
\end{verbatim}