1. (8 points) Show how to implement a circuit for each of the following expressions using a multiplexor (as on page 3-16 of the notes).

(a) \((A + B)C' + B'D\)

(b) \(A'C + (B'D + C)(A' + D)\)

\[
\begin{array}{c|c|c|c|c|c|c|c}
A & B & C & D & F \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c|c}
A & B & C & D & F \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]
2. (10 points) Prove that the alternative implementation for the 3→8 decoder on page 3-12 of the course notes, is correct. Do this as follows. First, write down the logic equations for the 2→4 decoder with enable, then derive the logic equations for the circuit shown on 3-12 and show that it is the same as the logic equations for the 3→8 decoder component.

The equations for the 2→4 decoder with enable are shown below.

\[ D_0 = E A' A' \quad D_1 = E A' A_0 \quad D_2 = E A_1 A' \quad D_3 = E A_1 A_0 \]

In the alternative implementation of the 3→8 decoder, the 2→4 decoder with enable appears twice. In the copy that produces \( D_0 \) through \( D_3 \), the \( E \) input is connected to \( A' \_2 \) and in the copy that produced \( D_4 \) through \( D_7 \), the \( E \) input is connected to \( A_2 \). So,

\[ D_0 = A' A' A_0 \quad D_1 = A' A_1 A_0 \quad D_2 = A' A_1 A' \quad D_3 = A' A_1 A_0 \]
\[ D_4 = A_2 A' A_0 \quad D_5 = A_2 A_1 A_0 \quad D_5 = A_2 A_1 A' \quad D_7 = A_2 A_1 A_0 \]
3. (10 points) Let \( y_{n-1} \ldots y_0 \) be the 2s-complement of \( x_{n-1} \ldots x_0 \). Show that for all \( i > 0 \), 
\[ y_i = x_i \oplus (x_0 + \cdots + x_{i-1}) \]
and that \( y_0 = x_0 \).

Let \( z_i = x'_i \), so \( y_{n-1} \ldots y_0 = z_{n-1} \ldots z_0 + 1 \). Now, notice that when you add 1 to a number with several 1s at the right end, these 1s become 0s in the sum as the carry ripples along. The first 0 (from the right) then becomes a 1 and the remaining bits are not changed. So, if 
\[ z_{n-1} \ldots z_0 = 01 \ldots 1, \]
then \( y_{n-1} \ldots y_0 = z_{n-1} \ldots z_0 10 \ldots 0 \). Now, since \( z_i = x'_i \), this is equivalent to saying that if \( x_{n-1} \ldots x_0 = x_{n-1} \ldots x_i 10 \ldots 0 \) then \( y_{n-1} \ldots y_0 = z_{n-1} \ldots z_i 10 \ldots 0 = x_{n-1} \ldots x'_i x_{i+1} \ldots x_0 \). This means first that \( y_0 = x_0 \) and second that \( y_i = x_i \) if \( x_0 \ldots x_{i-1} \) are all 0s, otherwise \( y_i = x'_i \). This is equivalent to 
\[ y_i = x_i \oplus (x_0 + \cdots + x_{i-1}). \]

Design a 4 bit 2s-complement circuit based on this equation, using only inverters, 2 input AND gates and 2 input OR gates. Design your circuit to be as fast as possible. What is the worst-case propagation delay for this circuit, assuming a 1 ns delay for each inverter and a 2 ns delay for each AND gate and OR gate? What is the worst-case delay for an 8 bit version of this circuit? For a 16 bit version?

The worst-case propagation delay is 9 ns. An 8 bit version can be designed with a worst-case propagation delay of 11 ns, and an 16 bit version can be designed with a worst-case propagation delay of 13 ns.
4. (15 points) Problem 8 of homework 4, involved the construction of a 4 bit comparison circuit in which results ripple from the most significant bits of the two values being compared to the least significant bit, in much the same way that a carry ripples from least significant bit to most significant bit in a ripple-carry adder. A block diagram of this circuit is shown below.

In this circuit, output $G_i$ is high if the $x_3...x_i > z_3...z_i$, output $Q_i$ is high if $x_3...x_i = z_3...z_i$ and output $L_i$ is high if $x_3...x_i < z_3...z_i$. These outputs satisfy the following equations.

$$G_i = G_{i+1} + Q_{i+1}z_i'$$

$$L_i = L_{i+1} + Q_{i+1}x_i'z_i$$

$$Q_i = Q_{i+1}(x_i \oplus z_i)'$$

Derive equations for the signals $Q_3, Q_2,$ and $Q_1$ that depend only on inputs $x_3...x_0$ and $z_3...z_0$. Then, derive equations for $G_0$ and $L_0$ that depend only on $x_3...x_0, z_3...z_0$ and $Q_3...Q_1$.

$$Q_1 = (x_3 \oplus z_3)'$$

$$Q_2 = (x_3 \oplus z_3)'(x_2 \oplus z_2)'$$

$$Q_3 = (x_3 \oplus z_3)'(x_2 \oplus z_2)'(x_1 \oplus z_1)'$$

$$G_0 = x_3z_3' + Q_3x_2z_2' + Q_2x_1z_1' + Q_1x_0z_0'$$

$$L_0 = x_3'z_3 + Q_3x_2'z_2 + Q_2x_1'z_1 + Q_1x_0'z_0$$

Use these equations to design a “look-ahead” comparison circuit using only inverters, 2 input AND gates and 2 input OR gates. Design your circuit to be as fast as possible. What is the maximum propagation delay for your circuit, assuming that every 2 input gate has a 2 ns delay and every inverter has a 1 ns delay? For an 8 bit version? For a 16 bit version? What is the maximum propagation delay for a 16 bit version of the “ripple” comparison circuit if it is implemented using only inverters, AND gates and OR gates?
The maximum propagation delay is 15 ns. An eight bit version of the circuit would have a maximum propagation delay of 19 ns and a 16 bit version would have a maximum propagation delay of 23 ns. Referring back to the solution of homework 4, we can see that with the specified component delays, the “ripple” comparison circuit will have a delay of 7 ns for the first stage plus 2 ns for every other stage. So, a 16 bit version of the circuit would have a maximum propagation delay of 37 ns.

5.
(10 points) The equations below are the basis for the ripple-carry adder circuit (see page 3-19 of the notes).

\[ S = X \oplus Y \oplus C_{in} \quad C_{out} = XY + XC_{in} + YC_{in} = XY + (X + Y)C_{in} \]

There is a similar set of equations that can be used to create a ripple-carry subtractor circuit.

\[ D = X \oplus Y \oplus B_{in} \quad B_{out} = XY + XB_{in} + YB_{in} = X'Y + (X' + Y)B_{in} \]

Draw a logic diagram for a 4 bit subtractor, based on these equations.
6. (20 points) A combination adder-subtractor circuit can be built in a way similar to the ripple-carry adder and the ripple carry subtractor. Design a circuit that can be combined to form a ripple-carry adder-subtractor, in the same way that full adders are combined to form a ripple-carry adder. Your circuit should have a control input, \( S \) (short for subtract), which makes the circuit behave as a subtractor when it is high, and as an adder when it is low. Use your circuit to implement a 4 bit adder-subtractor and simulate it using a unit delay simulation with a simulation precision of 1 ns. Select a variety of input values, and in particular show input values that demonstrate arithmetic overflow and underflow, and input changes that demonstrate the worst-case delay from an input signal change to an output change. What is this worst-case delay? Hand in a printout of the schematic and your simulation output.

**Extra Credit.** (10 points) Extend the circuit to include an additional control input \( U \) (for unsigned), and a status output \( V \) (for Overflow/Underflow). When \( U \) is high, the inputs are treated as unsigned integer values and output \( V \) is high if an addition operation results in overflow, or a subtraction operation results in underflow. When \( U \) is low, the input values are treated as signed integers (2s-complement) and \( V \) is asserted when the operation produces an incorrect result. Note that when adding two values of opposite sign, there is no possibility for the result to be incorrect, and that when adding two values of the same sign, the result is incorrect, if and only if the sign of the result differs from the sign of the input values. What is the similar rule for subtracting two values?

*The schematic is on the next page. The circuitry at the bottom implements overflow detection. When subtracting two values, there is an error if the sign of the two operands differs and the sign of the first operand differs from the sign of the result.*

*The worst-case propagation delay is 14 ns from the time an input changes until the time the \( V \) output changes. A case of this can be seen following the input change at time 20 ns. The worst-case propagation delay from an input change to one of the data outputs changing is 8 ns and can also be observed following the input change at 20 ns.*
7. (10 points) Write code for a VHDL module that has two inputs \( A \) and \( B \) plus a single output \( Z \), where \( Z = A \oplus B \). Compile your module and simulate it for all combinations of input values. Turn in a printout of your VHDL source and the simulation output.

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity vxor is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        x: out STD_LOGIC
    );
end vxor;

architecture vxor_arch of vxor is
begin
    x <= a xor b;
end vxor_arch;
```