1. (20 points) CMOS logic circuit. The CMOS circuit show in the figure below computes a useful logic function.

![CMOS Circuit Diagram]

a. Fill in the function table for the above circuit.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

b. Describe briefly (not a Boolean formula) the function computed by the above circuit.

If any two of the inputs are high, then the node W is low, so the output F is high; otherwise, F is low. In other words, F(X,Y,Z) is the majority function of three inputs.
2. (25 points) Combinational logic. The exclusive-or of two variables is defined to be true when one input is true but not both. Exclusive-or can be generalized to three or more variables in several ways:

- XORnZ: true if exactly one input is true.
- XORnY: true if at least one input is true but not all inputs are true.
- XORn: true if an odd number of inputs is true.

Only the last generalization is useful; for example, XOR3 is a component in a full adder.

a. Find the minimal sum-of-products representation of XOR4Z.

It is clear from the definition of XORnZ and from the first Karnaugh map below that XOR4Z is the sum of four minterms.

\[ \text{XOR4Z} = W \cdot X' \cdot Y' \cdot Z' + W' \cdot X \cdot Y' \cdot Z' + W' \cdot X' \cdot Y \cdot Z + W' \cdot X' \cdot Y' \cdot Z \]

b. Find the minimal sum-of-products representation of XOR4Y.

XOR4Y can be written as the sum of four product terms, each with two literals, in many ways. Three formulas are

\[ \text{XOR4Y} = X \cdot Y' + W \cdot X' + W' \cdot Z + Y \cdot Z' \]
\[ = W' \cdot X + X \cdot Y' + W' \cdot Z + Y \cdot Z' \]
\[ = W \cdot Y' + Y' \cdot Z + W' \cdot Y + Y' \cdot Z' \]

The first expression corresponds to the covering shown in the second Karnaugh map. The second formula is obtained from the first by permuting variables. The third formula states that \( Y \neq W \) or \( Y \neq Z \), which is sufficient to guarantee that not all input values are equal.

c. Find the minimal product-of-sums representation of XOR4Y.

The product-of-sums representation has only two maxterms, corresponding to the two zeroes in the second Karnaugh map.

\[ \text{XOR4Y} = (W' + X' + Y' + Z') \cdot (W + X + Y + Z) \]
\[ = (W \cdot X \cdot Y \cdot Z + W' \cdot X' \cdot Y' \cdot Z')' \]

The second formula follows from DeMorgan’s laws; only an inverter stands between sums of products and products of sums.
3. (30 points) *Glitches.* A devious circuit designer constructs the following circuit.

```
   C1  T1  T2
     C2  C3
```

a. Find the Boolean formula for \( C3 \).

The analysis of the Boolean function computed by this circuit is extremely simple:

\[
C2 = C1 \oplus C1'''' = C1 \oplus C1 = 0 \\
C3 = C2 \oplus C2'' = C2 \oplus C2 = 0 \oplus 0 = 0
\]

b. Suppose that the propagation delay for the inverters is exactly 5 ns and the propagation delay for the XORs is exactly 10 ns. Complete the following timing diagram.

![Timing Diagram]

(Initial values of \( T1, C2, T2, C3 \) are not well defined because they depend on \( C1 \) for \( t < 0 \).)

c. Using the result of part (b), describe the output \( C3 \) when the input \( C1 \) is a square wave with 80 ns clock period (12.5 MHz).

\( C2 \) pulses with each transition of \( C1 \), and \( C3 \) pulses with each transition of \( C2 \). Thus \( C3 \) is a square wave with four times the frequency of \( C1 \), that is, clock period 20 ns (50 MHz)

d. Suppose that the propagation delays are given by the following table:

<table>
<thead>
<tr>
<th></th>
<th>( t_{\text{min}} )</th>
<th>( t_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{NOT} )</td>
<td>4 ns</td>
<td>6 ns</td>
</tr>
<tr>
<td>( \text{XOR} )</td>
<td>9 ns</td>
<td>12 ns</td>
</tr>
</tbody>
</table>

Find minimum and maximum durations of the high pulses of intermediate signal \( C2 \).

Start at \( t = 10 \) ns. To find the maximum duration of the \( C2 \) high pulse: When \( C1 \) goes high, \( C2 \) goes high in \( 9 \) ns min \( (t = 19 \) ns), whereas \( T1 \) will go high after \( 24 \) ns max \( (t = 34 \) ns). Then \( C2 \) goes back low after after \( 12 \) ns max \( (t = 46 \) ns). The maximum duration of the high pulse is \( 46 - 19 = 27 \) ns.

Similarly, to find the minimum duration: When \( C1 \) goes high, \( C2 \) goes high after \( 12 \) ns max \( (t = 22 \) ns). \( T1 \) will go high after \( 16 \) ns min \( (t = 26 \) ns). Then \( C2 \) goes back low after \( 9 \) ns min \( (t = 35 \) ns). The minimum duration of the high pulse is \( 35 - 22 = 13 \) ns.
4. (30 points) Funnel shifter. An $n$-bit funnel shifter is a combinational circuit that extracts $n$ contiguous bits from the concatenation of two $n$-bit words, as shown below.

![Diagram of funnel shifter circuit]

a. Complete the design of the following 4-bit funnel shifter by connecting input signals to multiplexer inputs. The least significant bit of the extracted field is specified by the multiplexer control inputs $SEL[2:0]$, as shown by the signals connected to mux input 0.

b. A logical left shift SLL shifts an operand left, filling in the low-order bits with zeroes. For example, $SLL(X[3:0] ; 2) = (X1, X0, 0, 0)$. To shift an input vector $X[3:0]$ left, the input is connected to one of the funnel shifter inputs, A or B, and zeroes are supplied to the other input. Which input is $X$ connected to?

To perform a left shift, the data input $X$ should be connected to the funnel shifter B inputs, which are to the “left” of the A inputs, which are connected to zeroes.

c. What value of the control input $SEL[2:0]$ is used to compute $SLL(X, 3)$?

Shifting $X$ left by 3 bits produces the output bit vector $(X0, 0, 0, 0)$, which results from the control value $SEL[2:0] = 001 = 1$. In general, to shift left by $i$, where $0 \leq i \leq 4$, the funnel shifter $SEL$ input must be $4 - i$.

d. A funnel shifter can also be used to rotate a 4-bit operand, by connecting the input operand to both data inputs of the funnel shifter. What control value $SEL[2:0]$ is used to produce the left rotation $ROL(X[3:0], 1) = (X2, X1, X0, X3)$?

To route $X2$ to $Y3$, $X1$ to $Y2$, and so on, the shift amount should be $SEL[2:0] = 011 = 3$. In general, to rotate left by $i$, where $0 \leq i \leq 4$, the funnel shifter $SEL$ input must be $4 - i$, while to rotate right by $i$, $SEL[2:0] = i$. 