Problem 1. Consider a multi-functional **Increment/pass-through Unit**, which has three inputs and three outputs as shown. The data input consist of two bits, \(I_1\) and \(I_2\), and there is a select input, \(S\). The data output consists of two bits, \(O_1\) and \(O_2\), with an overflow output bit, \(O_v\). Depending on the value of select signal, the Unit either increments the data input \((I_2I_1)\) to produce the data output \((O_2O_1)\) (when \(S=1\)), or lets the data input “pass through” to the data output without any change (when \(S=0\)).

The pseudo-code for the Unit is as follows:

```plaintext
if (S=0)
    \(O_2O_1=I_2I_1\)
    \(O_v=0'\)
else
    if \(I_2I_1=\text{"}11\text{"}
        \(O_v=1'\)
        \(O_2O_1=\text{"}00\text{"}\)
    else
        \(O_v=0'\)
        \(O_2O_1=I_2I_1+1\)
    end if
end if
```

(a) Fill in the truth table for the output functions  

<table>
<thead>
<tr>
<th>(I_2)</th>
<th>(I_1)</th>
<th>(S)</th>
<th>(O_2)</th>
<th>(O_1)</th>
<th>(O_v)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(b) Express the functions in sum of products (min-terms) and product of sums (max-terms) form  (10 points)

(c) Fill in the K-maps for the output functions, $O_1$, $O_2$, $O_v$.
Using the K-map method, derive the minimal sum-of-product representations of the 3 output functions.  
(10 points)

\[
\begin{array}{c|cc|ccc}
  & S & I_2I_1 & 00 & 01 & 11 & 10 \\
\hline
  0 & & &  &  &  \\
  1 & & &  &  \\
\end{array}
\]

$O_v$=

\[
\begin{array}{c|cc|ccc}
  & S & I_2I_1 & 00 & 01 & 11 & 10 \\
\hline
  0 & & &  &  &  \\
  1 & & &  &  \\
\end{array}
\]

$O_2$=

\[
\begin{array}{c|cc|ccc}
  & S & I_2I_1 & 00 & 01 & 11 & 10 \\
\hline
  0 & & &  &  &  \\
  1 & & &  &  \\
\end{array}
\]

$O_1$=
(d) Implement the Increment/pass-through Unit output functions using gates from a library consisting of: 2-input NAND, 2-input AND, 2-input XOR, and inverter gates. While you can use any number of gates of each type from the library, derive an implementation which uses the least number of total gates. Draw the net list schematic of the implementation.  (10 points)
Problem 2. The circuit below shows the implementation of a controller using an 8-word by 2-bit read-only memory (ROM) and a 2-bit register. $A_2$ is the most significant address bit of the ROM. Specify the content of the ROM, so that the state transition table below is implemented, where $X$ is an input to the controller. (10 points)

\[
\begin{array}{c|c}
XQ_1Q_0 & B_1B_0 \\
\hline
000 & 01 \\
100 & 00 \\
001 & 10 \\
101 & 00 \\
010 & 11 \\
110 & 01 \\
011 & 01 \\
111 & 01 \\
\end{array}
\]
**Problem 3.** Consider the following control circuit, which contains a 3-bit register and a block with some combinational logic. The initial state of the circuit is $Q_1Q_2Q_3=010$. The circuit generates the control sequence $(010) \rightarrow (110) \rightarrow (001) \rightarrow (001) \rightarrow \ldots \rightarrow (001)$ on successive clock cycles. Show the Boolean expressions implemented by the combinational logic box. (10 points)
Problem 4. A Moore sequential network has one input and one output. The output should be 1 if the total number of 0’s received at the input is ODD and the total number of 1’s received is an EVEN number greater than 0. This machine can be implemented in exactly six states.

(a) Draw the state transition graph; explain the meaning of each state in your design, that is, what partial strings are recognized in each state.  (10 points)

(b) Draw the state transition table.  (10 points)

(c) Encode the states using minimum number of bits.  (10 points)

(d) Derive the logic schematic for a sequential circuit which implements the state transition table.  (10 points)