1. Design a synchronous Moore machine that monitors two inputs $x$ and $y$, and asserts an output $z$ if the number of 1’s observed on the inputs is a multiple of 4. The inputs and outputs will look like this:

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of 1’s | 1 | 2 | 4 | 4 | 5 | 5 | 7 | 9 | 9 | 10 | 10 | 12 | 13 |

| $z$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

(a) Draw a state diagram for the system.
(b) Find the state transition table for the system.
(c) Find the excitation equations for the flip-flops.
(d) Write a Verilog program to implement the system.

2. The USB bus transmits data using a protocol called Non Return to Zero Inverted (NRZI). An NRZI encoder has one input $x$ (as well as the clock and reset lines) and one output $z$. If $x$ is 0, the output $z$ will change value (if $z$ was a 0, it will change to a 1; if $z$ was a 1 it will change to a 0). If the input $x$ is 1, the output $z$ will stay the same (remain 0 if it was a 0, and 1 if it was a 1). An example is shown below:

(a) Draw a Moore state diagram for the system.
(b) Find the state transition table and make state assignments.
(c) Find the excitation equations for the flip-flops.
(d) Write a Verilog program to implement the system.
3. A pair of signals Request ($R$) and Acknowledge ($A$) is used to coordinate the transactions between a CPU and a peripheral. The interaction of these signals is often called a “handshake”. For a transaction to be valid, the sequence of signals is that shown on the left below: both signals low, $R$ goes high, $A$ goes high, $R$ goes low, $A$ goes low. You are to design a circuit which looks for errors in the handshake. The circuit has two inputs $R$ and $A$ (as well as the clock and reset), and one output $E$. If the input sequence is in the correct order, $E$ will be 0. If the circuit observes a different sequence (as shown on the right side of the figure below), the error signal $E$ will go high and stay high until the system is reset.

(a) Draw a Mealy state diagram for the system.
(b) Find the state transition table for the system.
(c) Find the excitation equations for the flip-flops.
(d) Write a Verilog program to implement the system.