   
   A. Implement the following function using the template of 2:1 Multiplexers given.
   NOTE: They are either one of the literals, 1 or 0.

   ![Diagram of 2:1 Multiplexers](image)

   B. Implement the function using minimum number of 2:4 decoders and other gates.

   Solution:
   A. \( X_1 = D', X_2 = 0, X_3 = 1, X_4 = 0 \)
B. Top decoder can also be replaced by an INVERTER plus an AND gate. There's a decoder/gates tradeoff. Both answers are acceptable here. Also, if you have constructed this ckt from truth table, you may be needing 5 decoders. That’s acceptable too.

2. Consider the following incomplete state diagram and excitation table with one 1-bit input and one 1-bit output. The FSM detects the pattern “1101” (repeating pattern allowed) and outputs a 1 when the pattern is detected.

Example:

<table>
<thead>
<tr>
<th>x</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>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
a. Is this Mealy or Moore machine?
b. Complete the above state diagram.
c. Using binary state assignments (e.g. S0=00, S1=01,...), complete the excitation table with current state, inputs, next state and output.

<table>
<thead>
<tr>
<th>Present state</th>
<th>Input (x)</th>
<th>Next state</th>
<th>Output(Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q0</td>
<td>N1</td>
<td>N0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

d. Use k-map to derive the equation for D1(MSB) and output Z.
e. Implement the logic for D1(MSB) and output using D-FF and minimum number of other multiple-input gates.

Solution:
a. Mealy
b. Excitation table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Input (x)</th>
<th>Next state</th>
<th>Output(Z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q0</td>
<td>N1</td>
<td>N0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

c. Equation for MSB

\[ D1 = Q1Q0' + xQ1'Q0 \]

Equation for output Z

\[ Z = xQ1Q0 \]
d. Implementation for MSB
3. Consider the following code.
Inputs: byte Mem[10], byte A, bit in, bit r
Output: bit equal

```c
i = 0;
equal = 0;
while(1) {
    if in == 1 {
        while (in == 1) {}
        if Mem[i] != A {
            i = 0;
        } else {
            i = i + 1
        }
    }
    if r == 1 {
```
Solution:

a. This code compares the sequential input A (sampled when s = 1) with an array of length 10 in memory. If all 10 elements are equal, the circuit outputs equal = 1 and stops in a dead-end state (bad design practice, usually we don’t want such states, but this code demands a dead-end state). If an input differs from memory or reset signal r is asserted, the circuit resets and waits for the first number to compare.

b. 

```
while (r == 1) {}
  i = 0;
}
if i == 10 {
  equal = 1;
  break;
}
```
c. And d.

4. A. Consider the circuit below.
The elements of the circuit have the following timing characteristics:
For Flip Flop: $T_{pcq} = 50\text{ps}, T_{ccq} = 20\text{ps}, T_{\text{setup}} = 25\text{ps}, T_{\text{hold}} = 30\text{ps}$
All logic gates, $T_{pd} = 25\text{ps}, T_{cd} = 15\text{ps}$
And has zero clock skew.

(i) What is the maximum clock frequency for the circuit?
(ii) How much clock skew can the circuit tolerate before it experiences a hold time violation?

B. Consider the following circuit with a NOR-based SR-latch and a D-FF. Timing information is as follows:

Propagation delay for the FF: $0.5\text{ ns}$
Propagation delay for all gates (except ones in the FF): $0.5\text{ ns}$
Clock period = $4\text{ ns}$
Complete the waveforms below.

Solution:
Solution:
A. (i) $T_{pcq} + T_{pd} + T_{\text{setup}} < T_c$
   
   $50\text{ps} + 5 \times 25\text{ps} + 25\text{ps} < T_c$

   $T_c > 200\text{ps}$

   $\text{Freq} < 5\text{GHz}$

(ii) $T_{ccq} + T_{cd} > T_{\text{hold}} + T_{\text{skew}}$

   $T_{\text{skew}} < 20 + 15 - 30\text{ps}$

   $T_{\text{skew}} < 5\text{ps}$

B.