**Midterm 2 Solution**

1

1.1
Universal set of gates is a set of gates such that every Boolean function can be implemented with gates in this set. 
Alter: A Universal set of gates is one from which the three basic Boolean functions (NOT, AND and OR) can be implemented.

1.2
1.2.1 \{OR, NOT\}
Yes. AND can be implemented as \(ab = (a' + b')'\)

1.2.2 \{NAND, NOR\}
Yes.
NOT can be implemented by tying one input of NAND gate to 1.
AND can be implemented by inverting the output of NAND gate i.e. \(ab = ((ab)')'\)
Similarly OR can implemented by inverting the output of a NOR gate

1.2.3 \(\{f(x, y)\}\), where \(f(x, y) = x'y\)
Yes.
\(f(x, 1) = x'\) i.e. NOT can be implemented
\(f(x', y) = xy\) i.e AND can be implemented
OR can be implemented using the above two because \(x + y = (x'y')'\)

1.2.4 \(\{f(x, y, z)\}\), where \(f(x, y, z) = (x'y' + xy)z\)
Yes.
\(f(x, 0, 1) = x'\) i.e. NOT can be implemented
\(f(x, 1, z) = xz\) i.e AND can be implemented
OR can be implemented using the above two because \(x + y = (x'y')'\)

Rubric:
- For each of the above parts, award 3 points if the conclusion as well as the reason is correct.
- For parts 1.2.1 and 1.2.2, deduct 1 point if the conclusion is correct but reason is not provided.
- For parts 1.2.3 and 1.2.4, deduct 2 points if the conclusion is correct bu the reason is not provided.
- If the conclusion is incorrect, award 1 point for every functionality that is shown to be implemented by that set

2

If \(AB = 0\) then \(A \oplus B = A + B\).
Proof:
\(A = A(B + B')\) (Complement Law)
= \(AB + AB'\) (Distributivity)
= \(0 + AB'\) (\(AB = 0\))
Also,
\(B = B(A + A')\) (Complement Law)
= \(AB + A'B\) (Distributivity)
= \(0 + A'B\) (\(AB = 0\))
Therefore, \( A+B=AB'+A'B \)

Alter:
Let \( X = A \oplus B \) and \( Y = A + B \)
\( X = Y \) iff \( X + Y' = 1 \) and \( XY' = 0 \)
\( X+Y' = A \oplus B + (A + B)' \)
\[
= AB' + A'B + A'B' \quad (\text{DeMorgan's})
\]
\[
= AB' + A'B + A'B' + A'B' \quad (A+A=A)
\]
\[
= B'(A + A') + A'(B + B') \quad (\text{Distributivity})
\]
\[
= B' + A' \quad (\text{Identity})
\]
\[
= (AB)' \quad (\text{DeMorgan's})
\]
\[
= 1 \quad (AB=0)
\]

\( XY' = (A \oplus B)(A + B)' \)
\[
= (AB' + A'B)(A'B') \quad (\text{DeMorgan's})
\]
\[
= AB'A'B' + A'B'A'B' \quad (\text{Distributivity})
\]
\[
= 0 \quad (\text{Complement})
\]

Rubric:

- Deduct 1 point if the laws/properties are not listed

- For the alternative approach, award 5 points if the criterion for equality is listed, 5 points if \( X + Y' = 1 \) is proved correctly and 5 points if \( XY' = 0 \) is proved correctly

3

Rubric:
Only the outputs for the first 10 clock levels (includes both high and low levels) are considered. 1 point is deducted for every output at that level found incorrect.

3.1

3.2
4

4.1 Boolean expressions for the next states and the output.

\[
\begin{align*}
Q_0(t+1) &= XQ_1(t) + XQ_0(t)'
\\ Q_1(t+1) &= XQ_1(t) + XQ_0(t)
\\ M &= X'Q_1(t)Q_0(t)
\end{align*}
\]

According to those Boolean equations, the following transition table can be derived.

<table>
<thead>
<tr>
<th>state</th>
<th>(Q_1(t))</th>
<th>(Q_0(t))</th>
<th>(X)</th>
<th>state</th>
<th>(Q_1(t+1))</th>
<th>(Q_0(t+1))</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(S_1)</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(S_1)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_1)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(S_2)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_2)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_2)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(S_3)</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(S_3)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(S_0)</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(S_3)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(S_3)</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

4.2

The following state diagram can be obtained from the transition table.

![State Diagram](image)

Figure 1: State diagram of Problem 4.

4.3

This FSM outputs \(M = 1\) once finds pattern \(X = 1110\).

Rubric:
<table>
<thead>
<tr>
<th>cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>state</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_0$</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_3$</td>
</tr>
<tr>
<td>$M$</td>
<td>0</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>1</td>
</tr>
</tbody>
</table>

- Get 7 points for correct transition table.
- Get 2 points for each correct Boolean function of $Q_0$, $Q_1$ and $M$ if the transition table is incorrect.
- Deduct 2 points for each incorrect Boolean function but correct transition table.
- Deduct 1 points for one incorrect transition in the state diagram.
- Deduct 1 points for one incorrect state/M in 4.3
- Deduct 2 points for correctly describing state diagram and the table in 4.3 of the incorrect transition table in 4.1.
- Deduct 7 points for incorrect state diagram based on incorrect transition table in 4.1.
- Deduct 7 points for no state diagram.
- Deduct 5 points for incorrect table in 4.3 based on incorrect transition table in 4.1.
- Deduct 3 points for incorrect label in the state diagram, e.g., no output on transition edges.
- Get 5 points for correct description of FSM functionality.
- Get 5 points for correct table in 4.3.
- Get 5 points for correct description of FSM functionality but with 1 cycle shift of $M$.
- Deduct 2 points for incorrect states in the table but correct description of FSM functionality.

<table>
<thead>
<tr>
<th>$T$</th>
<th>$Q(t)$</th>
<th>$Q(t+1)$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</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>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In this table, $T$, $Q(t)$ and $Q(t+1)$ show the expected functionality of a T flip-flop, and $D$ shows the corresponding values required for $Q(t+1)$. Boolean expression for $D$ of a D flip-flop is

$$D = Q(t)T + Q(t)T'.$$

The final logic diagram is as
Figure 2: Logic diagram of Problem 5.

Rubric:

- Get 12 points for correct excitation table of $T(t)$, $Q(t)$, $Q(t + 1)$ and $D(t)$.
- Get 15 points for implementing D flip-flop with T flip-flop.
- Get 2 points for correct K map from the incorrect excitation table.
- Get 6 points for incomplete Boolean expression for $D$.
- Get 12 points for correct Boolean expression for $D$.
- Deduct 2 points for minor notation incorrectness in the correct logic diagram.