2a) Initially $Q_1 = 1$, $Q_2 = Q_3 = Q_4 = 0$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>$Q_3$</th>
<th>$Q_4$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

b)
Limit Cycle: 1000 → 0100 → 0010 → 0001 → 1000

4) Looking at the state table in #2 and starting w/ when I is 1 (state 0001), we see that I is I after every fourth clock pulse afterward. To give an output of 1 after every fifth clock pulse, we can add another flip-flop to the chain.
5) From previous questions, we see that the way to build an n-cycle counter is to have n flip-flops and an n-1 input AND gate. If we want to use less (as stated in this question) we can use an AND gate to combine the output of 2 or more counters, such the final output is only high when all the outputs of the counters are high (in other words, a common multiple). In this case we want the LCM of our counters to be 60. In order to use less than 12 flip-flops, we can have a design like this:
Cycle 1: 000 → 100 → 110 → 111 → 011 → 001 → 000
Cycle 2: 101 → 010 → 101

We can use an AND gate to combine Q₁, Q₂, Q₃ → it will only give a 1 when all of them are 1, which is once every 6 cycles according to limit cycle 1.
1. \[00000 \rightarrow 10000 \rightarrow 11000 \rightarrow 11100 \rightarrow 11110\]

2. \[01010 \rightarrow 10101\]

3. \[00100 \rightarrow 10010\]

4. \[10001 \rightarrow 01000 \rightarrow 01010 \rightarrow 10100 \rightarrow 11000 \rightarrow 11101\]

5. \[00010 \leftarrow 00101 \leftarrow 01011 \leftarrow 10111 \leftarrow 01110\]
8) A

\[ \begin{array}{c}
\text{CLK} \\
\text{A} \\
\text{D} \\
\text{Q} \\
\text{P} \\
\text{Q} \\
\text{P} \\
\end{array} \]

\[ \begin{array}{c}
\text{XOR} \\
\text{A} || \text{B} || \text{F} \\
\text{0} || \text{0} || \text{0} \\
\text{0} || \text{1} || \text{1} \\
\text{1} || \text{0} || \text{1} \\
\text{1} || \text{1} || \text{0} \\
\text{1} || \text{0} || \text{0} \\
\end{array} \]

a) \( A = 1 \Rightarrow \text{output of XOR} \rightarrow \overline{Q_2} = Q_2 \)

<table>
<thead>
<tr>
<th>N</th>
<th>Q_1</th>
<th>Q_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \begin{array}{c}
\text{2nd XOR} \rightarrow \overline{Q_1} \\
\end{array} \]

b) \( A = 0 \Rightarrow \text{output of XOR} \rightarrow \overline{Q_2} \)

<table>
<thead>
<tr>
<th>N</th>
<th>Q_1</th>
<th>Q_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \begin{array}{c}
\text{2nd XOR} \rightarrow Q_1 \\
\end{array} \]

C) \( A = 1 \)

\[ \begin{array}{c}
\text{CLK} \\
N = 1 \\
N = 2 \\
N = 3 \\
N = 4 \\
\text{t} \\
\end{array} \]

\[ \begin{array}{c}
Q_1 \\
Q_2 \\
\text{t} \\
\end{array} \]
8) c) 

\[ C(k) \]

\[ t \]

\[ Q_1 \]

\[ t \]

\[ Q_2 \]

\[ t \]

9) d) \( Q_1 \) and \( Q_2 \) are swapped in the 2 cases.

We want a D to A Converter (DAC) that outputs 10V when all the bits are high. Since the input voltages to the op amps are the same, and each binary number represents a multiple of 2, we have for \( V_{out} \):

\[ V_{out} = (2^{0} + \ldots + 2^{7}) \times \text{Supply voltage} \]

\[ 10V = (2^{7} - 1) \times \frac{X}{Y} \]

\[ 10V = 2^{5} \times \frac{2}{51} \]

\[ X \times Y = \frac{2}{51} \Rightarrow \text{see design on right hand side} \]