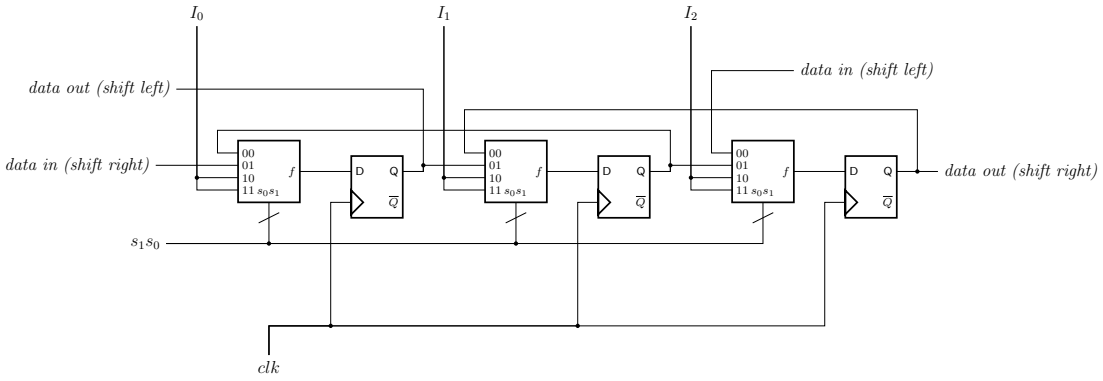
ECE 124 digital circuits and systems
Assignment #7

Q1: A bi-directional shift register can shift both left-to-right and right-to-left. Design a bi-directional shift register than also has a parallel-load capability.

Solution:
This is straightforward — take a bunch of DFFs to make a register. Then, add a multiplexer in front of every DFF data input. Finally, connect stuff up correctly and define the operation to be performed based on the select lines of the multiplexers.

Example for 3-bits...



The table explaining the operation of the circuit is given by

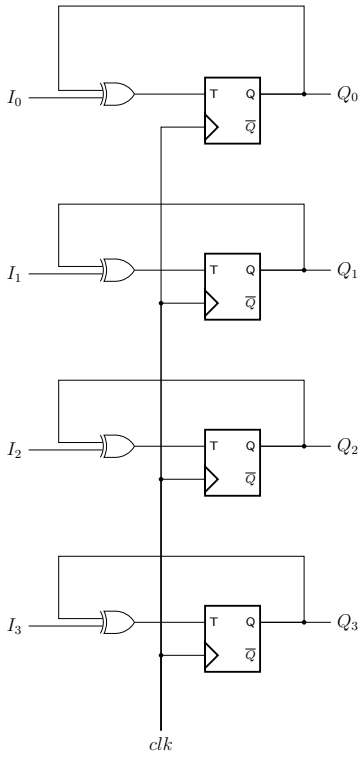
S_1	S_0	**Operation**
1	X	load
0	1	shift right
0	0	shift left

Q2: Design a 4-bit synchronous binary counter that also has a parallel load capability. Use T flip-flops.

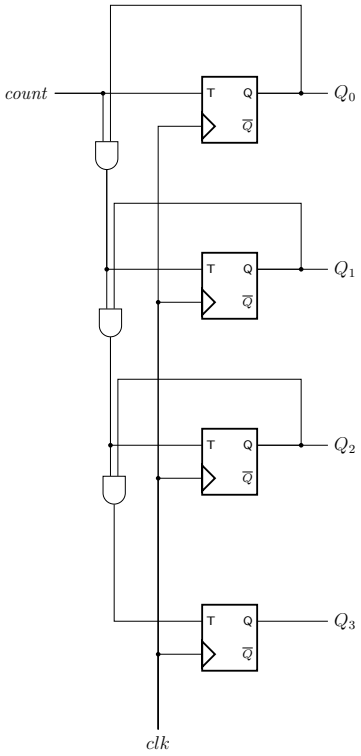
Solution:

This problem is answered most simply by joing together two different circuits: 1. The first circuit is a synchronous counter; 2. The second circuit does parallel load. Then, we take the circuits and use a multiplexer to select the desired operation.

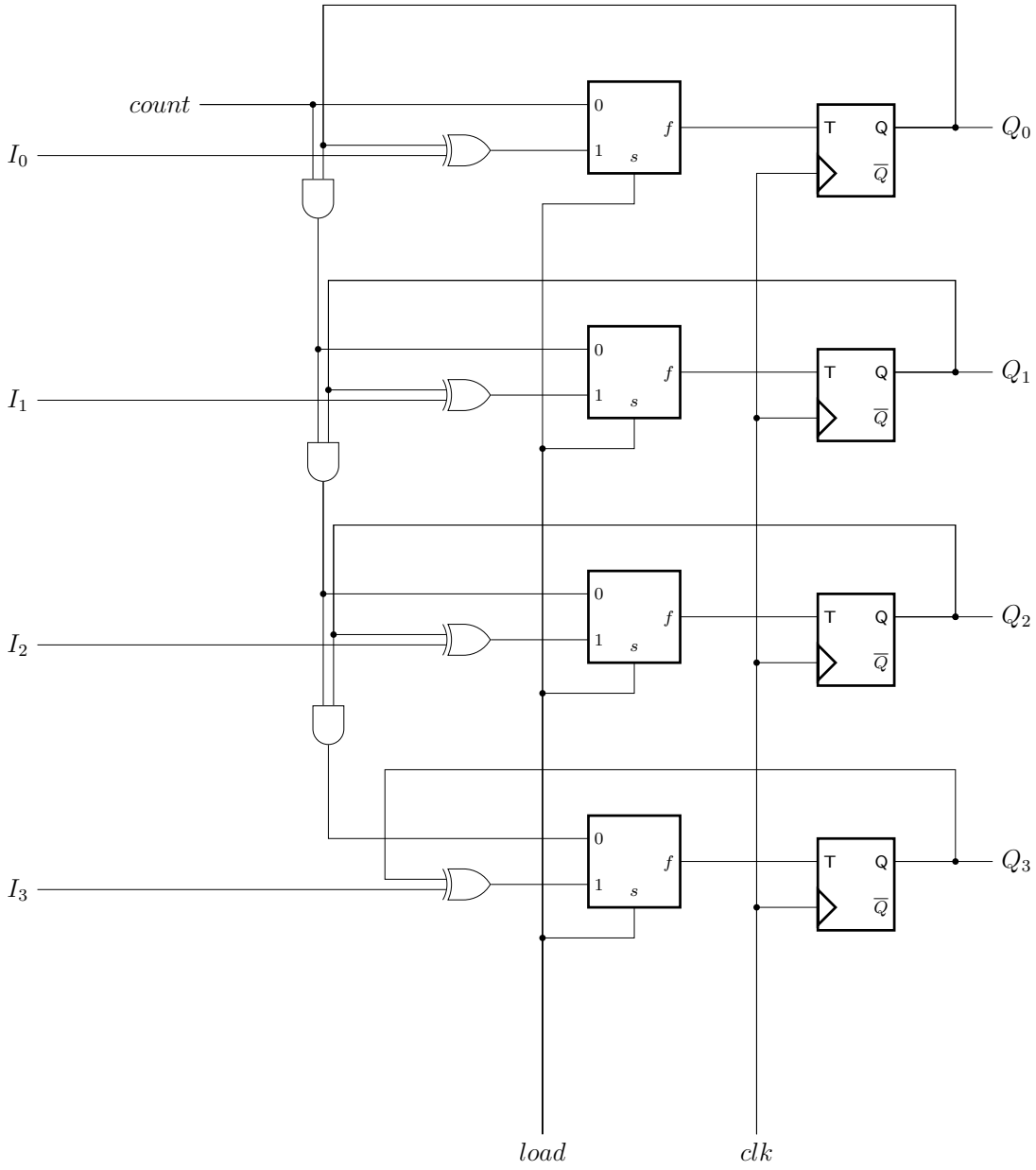
Circuit for parallel load...



Circuit for counting...



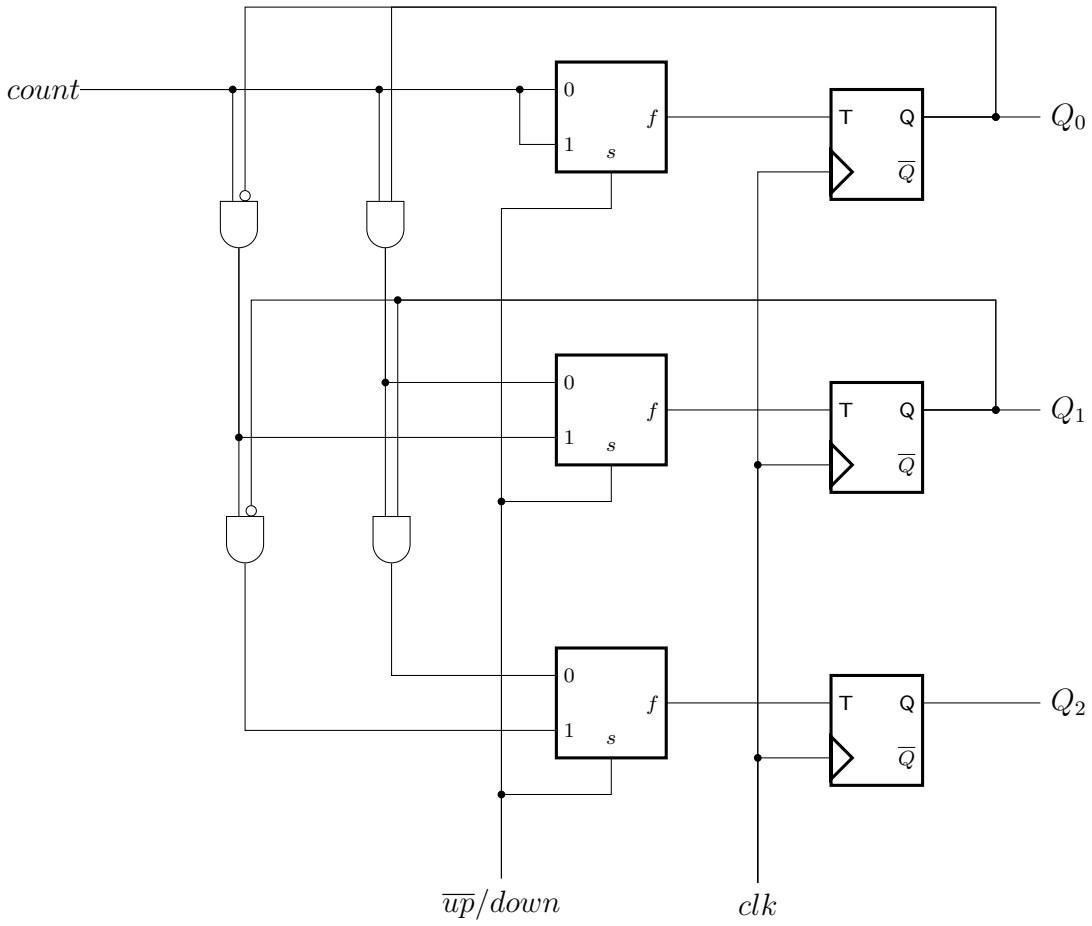
Then, use a multiplexer to get the final result...


count
I_0
I_1
I_2
I_3
0
1
f
s
T
Q
$\overline{Q}$
Q_0
Q_1
Q_2
Q_3
load
clk

Q3: Design a 3-bit up/down counter using T flip-flops. The counter should have one control input called $\overline{up}/down$. If $\overline{up}/down = 0$, then the circuit should behave as an up counter. If $\overline{up}/down = 1$, then the circuit should behave as an down counter.

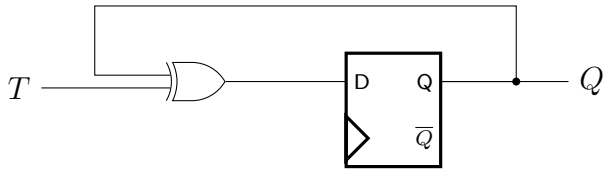
Solution:
This problem is answered most simply by joing together two different circuits: 1. The first circuit is an up counter; 2. The second circuit is a down counter. Then, we determine the operation using a multiplexer.

The circuit...



Q4: Repeat Problem 3 using D flip-flops.

Solution:
This problem is answered most simply by taking the circuit from the previous problem and converting the TFF into DFF. That is, replace every TFF with the following...



Q5: Consider the circuit shown below in Figure 5. What is the count sequence for this circuit?

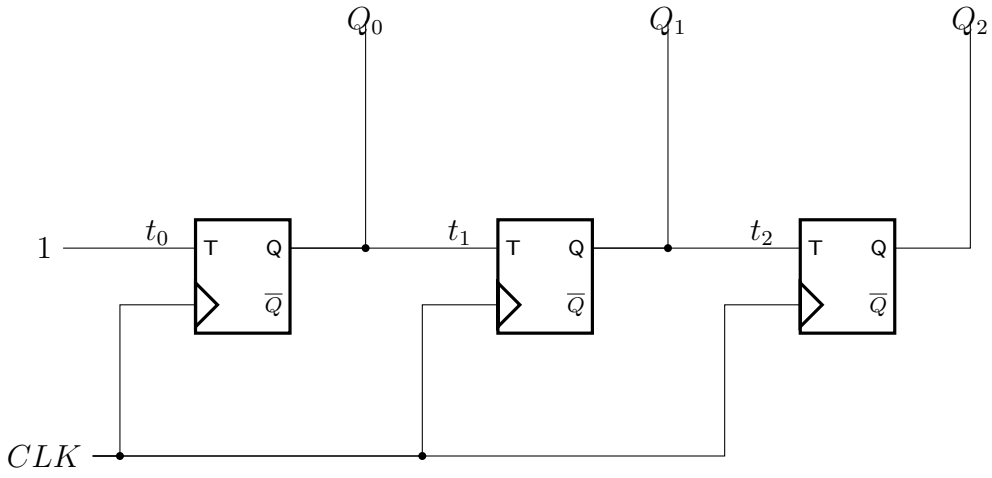


Figure 1: Circuit for Problem 5.

Solution:
This problem can be solved by creating a table to see how the inputs change. From this, we can figure out the count sequence.

Current Output			**Flip flop inputs**			**Next Output**		
q_2	q_1	q_0	t_2	t_1	t_0	q_2	q_1	q_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	0	1	1	1	1
0	1	1	1	1	1	1	0	0
1	0	0	0	0	1	1	0	1
1	0	1	0	1	1	1	1	0
1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0

This circuit is interesting... It apparently has two possible count sequences — the count sequence depends on what the initial outputs of the flip flops are when the circuit is turned on.

Specifically, the circuit might count $0 \rightarrow 1 \rightarrow 2 \rightarrow 7$ and repeat or it might count $3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ and repeat.

Q6: Consider the circuit shown in Figure 6 which implements a 4-bit counter with a parallel-load capability. Assume that each flip-flop has a setup time of $3ns$, a hold time of $1ns$ and a clock-to-output time of $1ns$. Further assume that each AND gate, XOR gate and 2-to-1 MUX has a propagation delay of $1ns$. What is the maximum frequency for which the circuit will operate correctly?

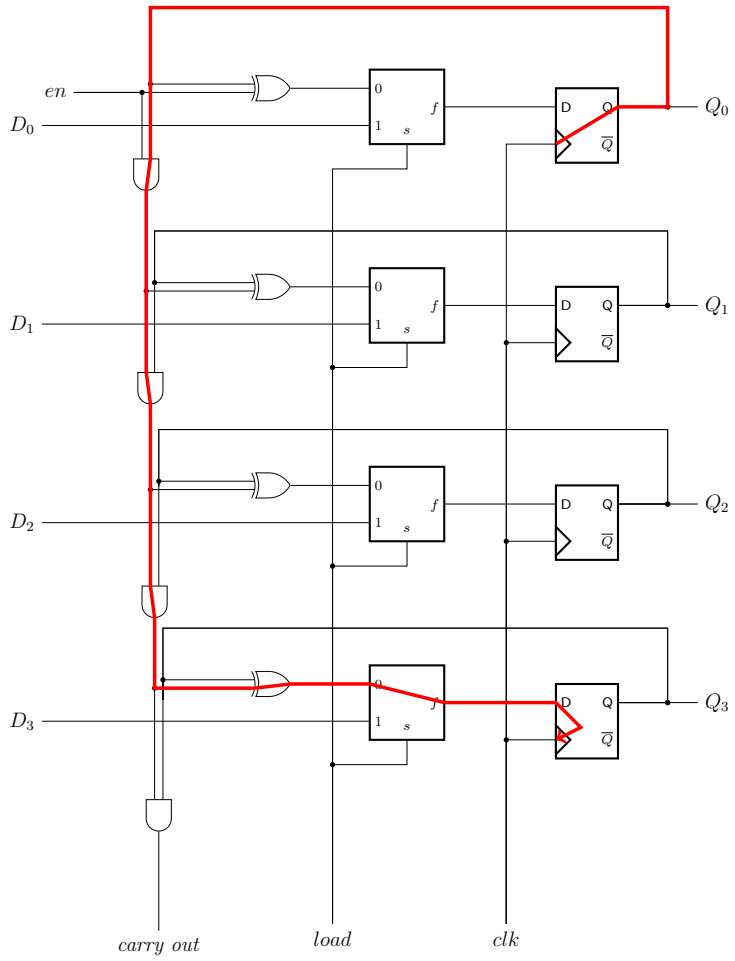


Figure 2: Circuit for problem 6.

Solution:

We need to figure out the longest path in the circuit. The longest path happens to be from the output of the first flip flop to the input of the last flip flop input through the AND gates, one XOR gate and one multiplexer. I've highlighted this path in bold. Note that we also need to include the clock-to-output time of the start flip flop and the setup time for the end flip flop.

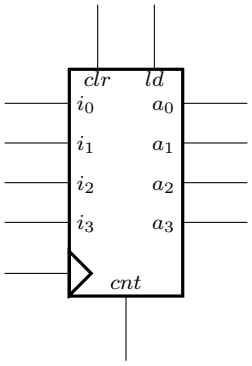
Therefore, the period of the clock must be at least

$$T_{clk} \geq T_{co} + T_{AND} + T_{AND} + T_{AND} + T_{XOR} + T_{MUX} + T_{SU}.$$

This is equal to $T_{CLK} \geq 9ns$ so the maximum frequency of the circuit is $f_{MAX} = \frac{1}{9ns} = 111MHz$.

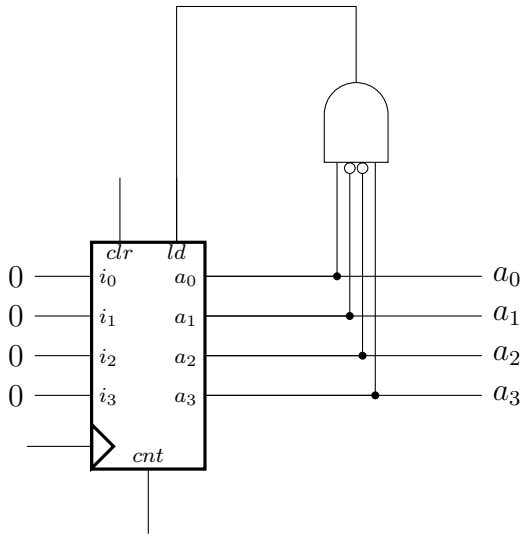
Q7: Shown below is a symbol for a 4-bit binary up-counter which also has the ability to be cleared or load new data. The operation of the counter is described in the table below. When in count mode, the circuit will count from 0000 $\rightarrow$ 0001 $\rightarrow$ 0010 $\rightarrow$ 0011 $\rightarrow \cdots \rightarrow$ 1111 and repeats. Note that the counting or loading of data is synchronous with the rising clock edge while the clear is asynchronous.

clr	*ld*	*cnt*	*clk*	Function
0	X	X	X	Output to zero
1	1	X	$\uparrow$	Parallel load
1	0	1	$\uparrow$	Count
1	0	0	$\uparrow$	Hold



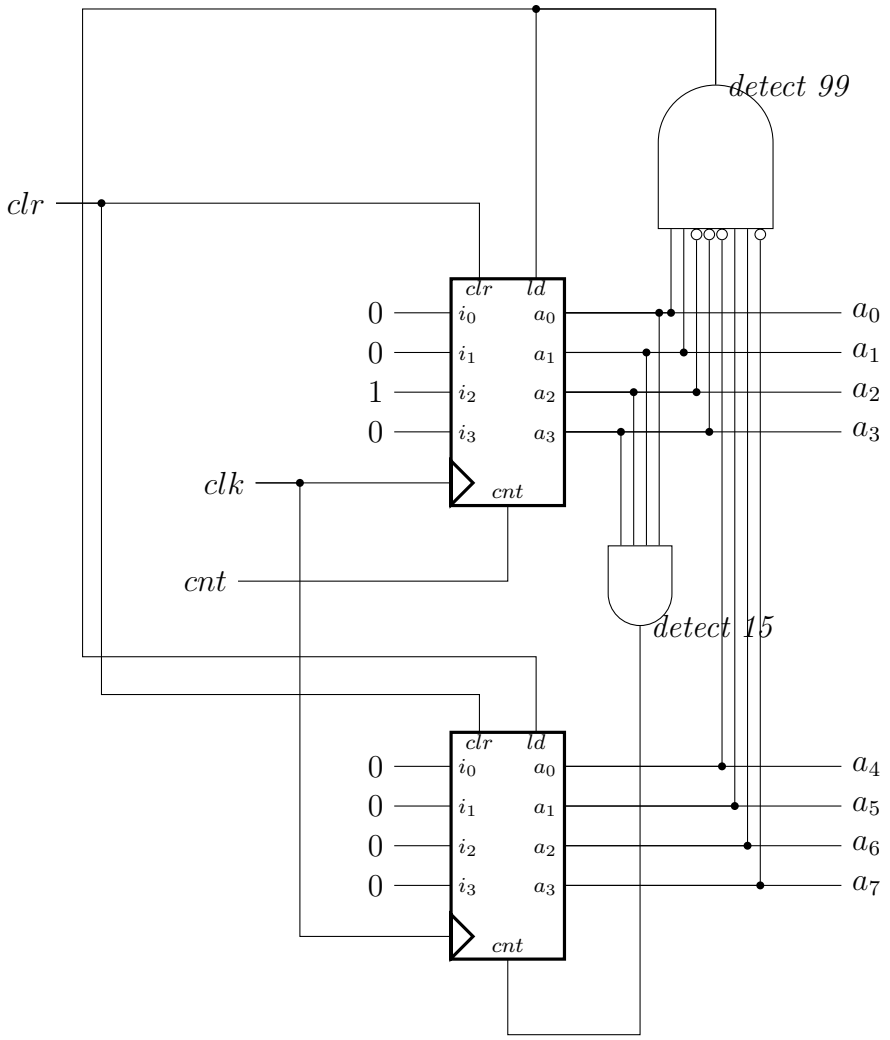
Design a module-10 counter using this 4-bit binary counter and any other additional logic; that is, design a counter which counts 0000 $\rightarrow$ 0001 $\rightarrow$ 0010 $\rightarrow$ 0011 $\rightarrow \cdots \rightarrow$ 1001 and repeats.

Solution:
Detect the output value 9 and then synchronously load the value 0 at the next clock edge.



Q8: Assume you are given two of the 4-bit binary counters from Problem 7. Construct a circuit using two of these 4-binary counters along with any additional logic required to count from binary 4 to binary 99 and repeat.

Solution:
The first counter will handle the lower bits which will count 0 to 15 and then repeat. Once we hit 15 we need the next counter to increment. Finally, once we hit 99 we need to synchronously load the binary value for 4 to restart the count. The binary value for 4 is 00000100. The binary value for 99 is 01100011.



Q9: Design a 3-bit binary counter that counts in the following sequence: $000 \rightarrow 001 \rightarrow 010 \rightarrow 111 \rightarrow 110$ and repeats. Design 3 different circuits; one that uses D flip-flops, one that uses T flip-flops and one that uses JK flip-flops.

Solution:
For each type of flip flop, simply figure out the necessary inputs to get the desired changes in flip flop outputs. Then, use Karnaugh maps to figure out the flip flop input equations. Finally, draw the circuits. Note that any count values not in the sequence can have the flip flop input values treated as don't cares.

The flip flop inputs and equations are shown below. You can draw the circuits if you want.

For DFF...

Current Output			**Next Output**			**DFF Inputs**			
q_2	q_1	q_0	q_2	q_1	q_0	d_2	d_1	d_0	
0	0	0	0	0	1	0	0	1	
0	0	1	0	1	0	0	1	0	
0	1	0	1	1	1	1	1	1	
1	1	1	1	1	0	1	1	0	
1	1	0	0	0	0	0	0	0	
0	1	1	X	X	X	X	X	X	← *not in count sequence*
1	0	0	X	X	X	X	X	X	← *not in count sequence*
1	0	1	X	X	X	X	X	X	← *not in count sequence*

We find $d_2 = q_1 q_0 + \overline{q}_2 q_1$, $d_1 = q_0 + \overline{q}_2 q_1$, and $d_0 = \overline{q}_2 \overline{q}_0$.

For TFF...

Current Output			**Next Output**			**TFF Inputs**			
q_2	q_1	q_0	q_2	q_1	q_0	t_2	t_1	t_0	
0	0	0	0	0	1	0	0	1	
0	0	1	0	1	0	0	1	1	
0	1	0	1	1	1	1	0	1	
1	1	1	1	1	0	0	0	1	
1	1	0	0	0	0	1	1	0	
0	1	1	X	X	X	X	X	X	← *not in count sequence*
1	0	0	X	X	X	X	X	X	← *not in count sequence*
1	0	1	X	X	X	X	X	X	← *not in count sequence*

We find $t_2 = q_1 \overline{q}_0$, $t_1 = \overline{q}_1 q_0 + q_2 \overline{q}_0$, and $t_0 = \overline{q}_2 + q_0$.

For $JKFF$...

Current Output			**Next Output**			**JFF Inputs**			
q_2	q_1	q_0	q_2	q_1	q_0	j_2k_2	j_1k_1	j_0k_0	
0	0	0	0	0	1	0X	0X	1X	
0	0	1	0	1	0	0X	1X	X1	
0	1	0	1	1	1	1X	X0	1X	
1	1	1	1	1	0	X0	X0	X1	
1	1	0	0	0	0	X1	X1	0X	
0	1	1	X	X	X	XX	XX	XX	← *not in count sequence*
1	0	0	X	X	X	XX	XX	XX	← *not in count sequence*
1	0	1	X	X	X	XX	XX	XX	← *not in count sequence*

We find $j_2 = q_1$, $k_2 = \overline{q}_0$, $j_1 = q_0$, and $k_1 = q_2\overline{q}_0$, $j_0 = \overline{q}_2$, and $k_0 = 1$.