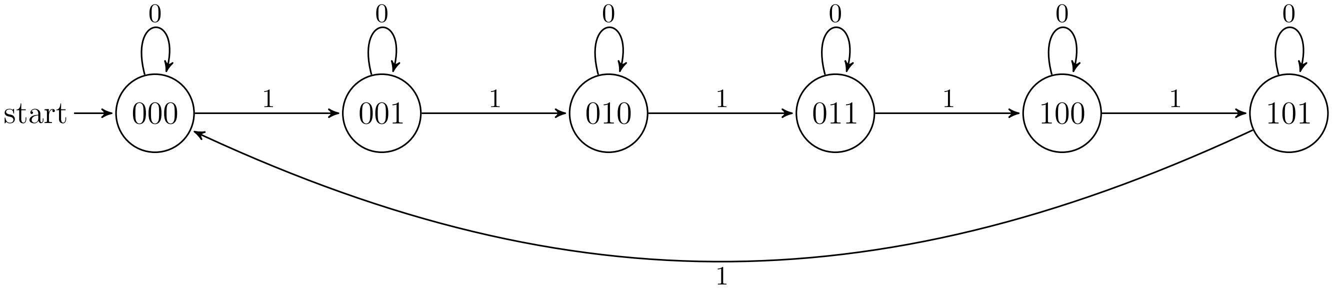
ECE 124 digital circuits and systems
Assignment #8

Q1: Design a modulo-6 counter which counts in the sequence $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ and repeats. The counter has one input *count*. The circuit should count only when $count = 1$, otherwise the circuit should not count. Design the counter using D flip-flops, T flip-flops and JK flip-flops.

Solution: I would start my observing that there are tricky ways this question could be answered. For example, one could start with a binary up counter and then add the appopriate logic to turn it into a modulo-6 counter. This could be done with whatever flip flop is typically used for a binary up counter (e.g., TFF). Then, one could simply replace each TFF with another type of flip flop since one can make one sort of flip flop from another.

We can use this example as an example of generic counter design as well. Start with a state diagram. The state is the count.



Using DFF...

Current State ($q_2q_1q_0$)	Next State ($q_2q_1q_0$)		DFF Inputs ($d_2d_1d_0$)	
	$count = 0$	$count = 1$	$count = 0$	$count = 1$
000	000	001	000	001
001	001	010	001	010
010	010	011	010	011
011	011	100	011	100
100	100	101	100	101
101	101	000	101	000
110	XXX	XXX	XXX	XXX
111	XXX	XXX	XXX	XXX

The optimized DFF inputs (found from 4 variable Karnaugh maps) are:

$$\begin{aligned}
d_2 &= q_2\overline{count} + q_2\overline{q}_0 + q_1 q_0 count \\
d_1 &= q_1\overline{count} + q_1\overline{q}_0 + \overline{q}_2\overline{q}_1 q_0 count \\
d_0 &= \overline{q}_0 count + q_0\overline{count}
\end{aligned}$$

You can draw the circuit if you want.

Using TFF...

Current State ($q_2q_1q_0$)	Next State ($q_2q_1q_0$)		TFF Inputs ($t_2t_1t_0$)	
	$count = 0$	$count = 1$	$count = 0$	$count = 1$
000	000	001	000	001
001	001	010	000	011
010	010	011	000	001
011	011	100	000	111
100	100	101	000	001
101	101	000	000	101
110	XXX	XXX	XXX	XXX
111	XXX	XXX	XXX	XXX

You can proceed to find t_2, t_1 and t_0 if you want as well as draw the circuit.

Using JKFF...

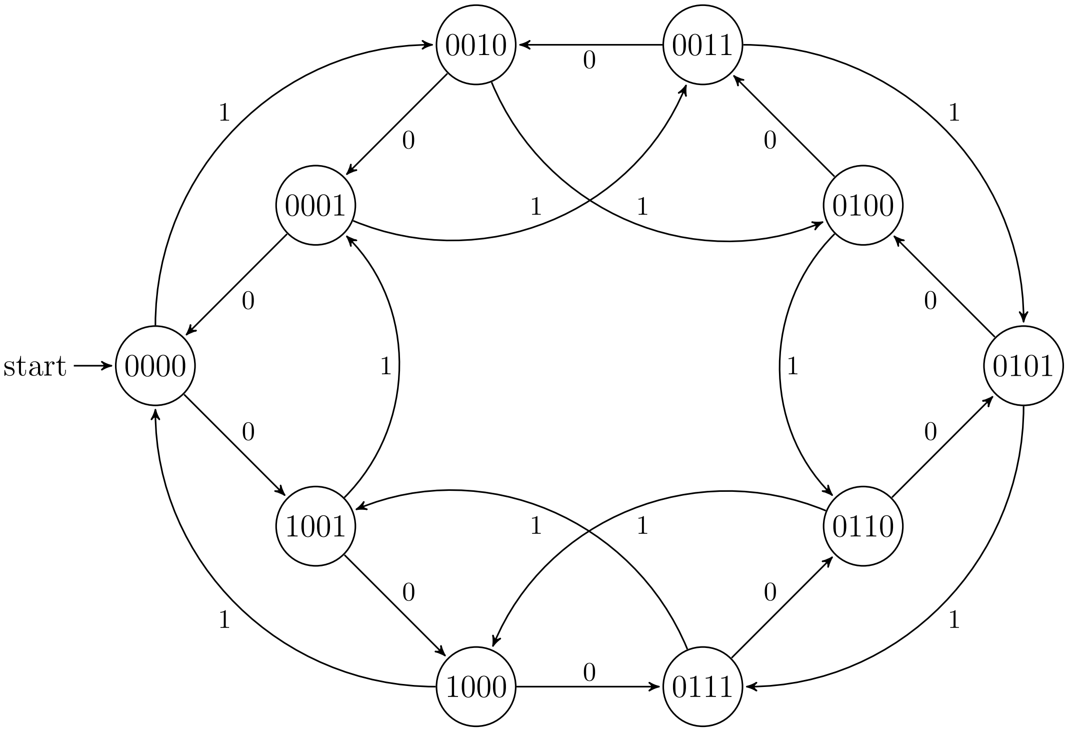
Current State	Next State		JKFF Inputs					
	$count = 0$	$count = 1$	$count = 0$			$count = 1$		
$q_2q_1q_0$	$q_2q_1q_0$	$q_2q_1q_0$	j_2k_2	j_1k_1	j_0k_0	j_2k_2	j_1k_1	j_0k_0
000	000	001	0X	0X	0X	0X	0X	1X
001	001	010	0X	0X	X0	0X	1X	X1
010	010	011	0X	X0	0X	0X	X0	1X
011	011	100	0X	X0	X0	1X	X1	X1
100	100	101	X0	0X	0X	X0	0X	1X
101	101	000	X0	0X	X0	X1	0X	X1
110	XXX	XXX	XX	XX	XX	XX	XX	XX
111	XXX	XXX	XX	XX	XX	XX	XX	XX

You can proceed to find the flip flops inputs if you want as well as draw the circuit.

Q2: Consider a counter-like circuit with one input x. When $x = 1$, the circuit should add 2 to its current value, wrapping around if the count reaches 8 or 9 — this means that if the current value is 8 or 9, then the next count value becomes 0 or 1, respectively. If $x = 0$, then the counter subtracts 1 from its current value, thus acting like a normal down-counter. Design this circuit using D flip-flops, T flip-flops and JK flip-flops.

Solution:

Start by drawing a state diagram:



Using DFF... (note that there are more rows since we've only shown current state from 0000 up to 1001. The remaining entries can be considered as don't cares):

Current State ($q_3q_2q_1q_0$)	Next State ($q_3q_2q_1q_0$)		DFF Inputs ($d_3d_2d_1d_0$)	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0000	1001	0010	1001	0010
0001	0000	0011	0000	0011
0010	0001	0100	0001	0100
0011	0010	0101	0010	0101
0100	0011	0110	0011	0110
0101	0100	0111	0100	0111
0110	0101	1000	0101	1000
0111	0110	1001	0110	1001
1000	0111	0000	0111	0000
1001	1000	0001	1000	0001

Finding optimized DFF input equations requires the use of 5 variable Karnaugh maps. I'm not going to do it here.

Using TFF... (note that there are more rows since we've only shown current state from 0000 up to 1001. The remaining entries can be considered as don't cares):

Current State ($q_3q_2q_1q_0$)	Next State ($q_3q_2q_1q_0$)		TFF Inputs ($t_3t_2t_1t_0$)	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0000	1001	0010	1001	0010
0001	0000	0011	0001	0010
0010	0001	0100	0011	0110
0011	0010	0101	0001	0110
0100	0011	0110	0111	0010
0101	0100	0111	0001	0010
0110	0101	1000	0011	1110
0111	0110	1001	0001	1110
1000	0111	0000	1111	1000
1001	1000	0001	0001	1000

Finding optimized TFF input equations requires the use of 5 variable Karnaugh maps. I'm not going to do it here.

Using JKFF...

Current State	Next State ($q_3q_2q_1q_0$)		JKFF Inputs							
	$x = 0$	$x = 1$	$x = 0$				$x = 1$			
$q_3q_2q_1q_0$	$q_3q_2q_1q_0$	$q_3q_2q_1q_0$	j_3k_3	j_2k_2	j_1k_1	j_0k_0	j_3k_3	j_2k_2	j_1k_1	j_0k_0
0000	1001	0010	1X	0X	0X	1X	0X	0X	1X	0X
0001	0000	0011	0X	0X	0X	X1	0X	0X	1X	X0
0010	0001	0100	0X	0X	X1	1X	0X	1X	X1	0X
0011	0010	0101	0X	0X	1X	X1	0X	1X	X1	X0
0100	0011	0110	0X	X1	1X	1X	0X	X0	1X	0X
0101	0100	0110	0X	X0	0X	X1	0X	X0	1X	X0
0110	0101	1000	0X	X0	X1	1X	1X	X1	X1	0X
0111	0110	1001	0X	X0	X0	X1	1X	X1	X1	X0
1000	0111	0000	X1	1X	1X	1X	X1	0X	0X	0X
1001	1000	0001	0X	0X	0X	X1	X1	0X	0X	X0

Finding optimized $JKFF$ input equations requires the use of 5 variable Karnaugh maps. I'm not going to do it here.

Q3: The state table for a Moore sequential circuit with one input w and one output z is shown below. States are already assigned binary patterns.

Current State	Next State		Output
	$w = 0$	$w = 1$	
q_1q_0	q_1q_0	q_1q_0	z
00	10	11	0
01	01	00	0
10	11	00	0
11	10	01	1

Derive a circuit for this state table using D flip-flops and then another circuit for this state table using JK flip-flops.

Solution:

Start by writing a table which shows the necessary flip flop input values in order to get the necessary state transitions. Next, write down the logic equations for the flip flop inputs depending on the type of circuit.

Current State	Next State		Output	DFF Inputs		JKFF Inputs			
	$w = 0$	$w = 1$		$w = 0$	$w = 1$	$w = 0$		$w = 1$	
q_1q_0	q_1q_0	q_1q_0	z	d_1d_0	d_1d_0	j_1k_1	j_0k_0	j_1k_1	j_0k_0
00	10	11	0	10	11	1X	0X	1X	1X
01	01	00	0	01	00	0X	X0	0X	X1
10	11	00	0	11	00	X0	1X	X1	0X
11	10	01	1	10	01	X0	X1	X1	X0

For DFF, we find

$$\begin{aligned} d_1 &= \overline{q}_1\overline{q}_0 + \overline{w}q_1 \\ d_0 &= w \oplus q_1 \oplus q_0 \end{aligned}$$

For $JKFF$, we find

$$\begin{aligned} j_1 &= \overline{q}_0 \\ k_1 &= w \\ j_0 &= \overline{w}q_1 + w\overline{q}_1 = w \oplus q_1 \\ k_0 &= \overline{w}q_1 + w\overline{q}_1 = w \oplus q_1 \end{aligned}$$

The output is a function of the state only

$$z = q_1q_0$$

You can draw the circuits if you want. Note that in writing the flip flop input equations, it was noticed that some of the logic resulted in **XOR** gates so that type of gate was used.

Q4: A clocked Mealy sequential circuit has one input x and one output z. The output should be $z = 1$ when the input sequences 1001 or 1111 appear at the input x, otherwise $z = 0$. Overlapping input patterns are allowed. An example of the desired behaviour is:

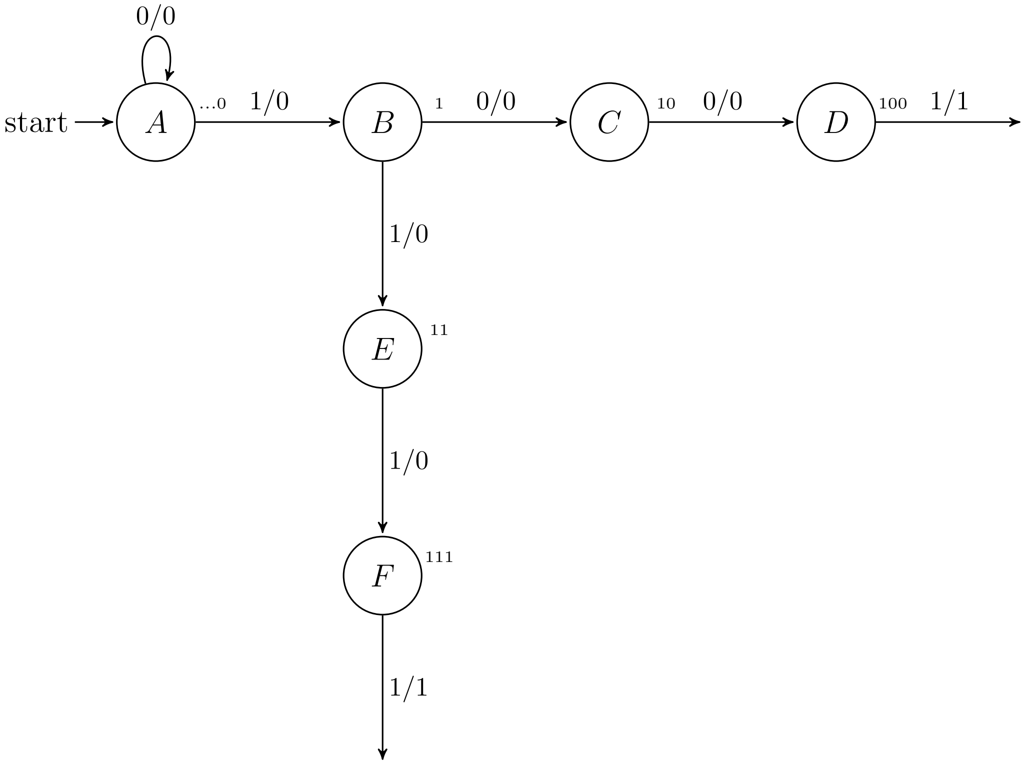
x:	0	1	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	...
z:	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	...

Find a Mealy state diagram and state table for this problem.

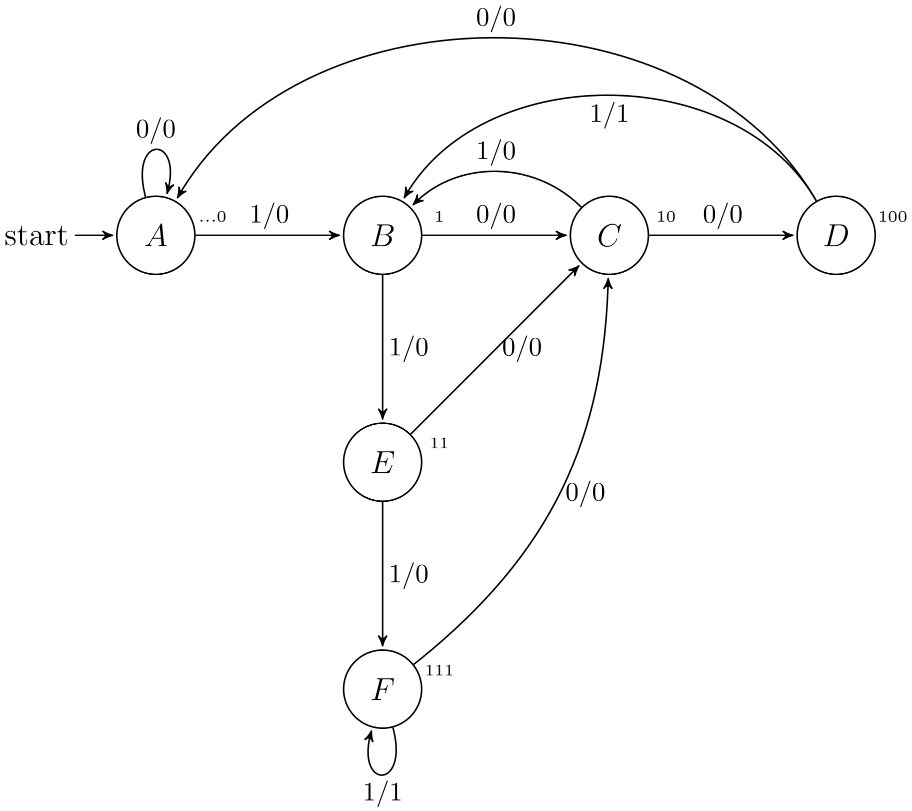
Solution:

Since this is a sequence detection circuit, it is useful to pick an initial state and then try to follow the sequence(s) to be detected. Finally, we will fill in any missing edges.

The initial state A can be "no inputs received or a string of 0s received". We then define states as the sequence of bits encountered so far leading to a result of 1 at the output. A partial state diagram would be:



Now, we need to fill in the missing edges as well as direct the edges that produced a 1 output. We may or may not need additional edges. As an example, consider state C where we have received 10; if the next value of x is 1, then we have the pattern 101 and this implies we can simply go back to state B which is the state representing the state of the patterns. If we are in state E and we receive a 0, then we have seen the pattern 110; the last two bits 10 are part of a pattern and represented by state C. We proceed like this to get the following complete state diagram:



The state table is

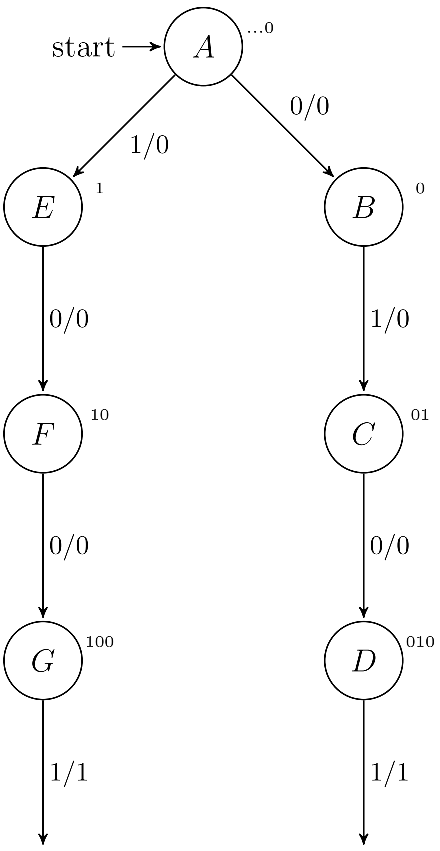
Current state	**Next state**		**Output** (z)	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	A	B	0	0
B	C	E	0	0
C	D	B	0	0
D	A	B	0	1
E	C	F	0	0
F	C	F	0	1

Q5: A sequential circuit has one input x and one output z. The circuit examines groups of four consecutive inputs and produces an output $z = 1$ if the input sequence 0101 or 1001 occurs. The circuit resets after every four inputs. An example of the desired behaviour is:

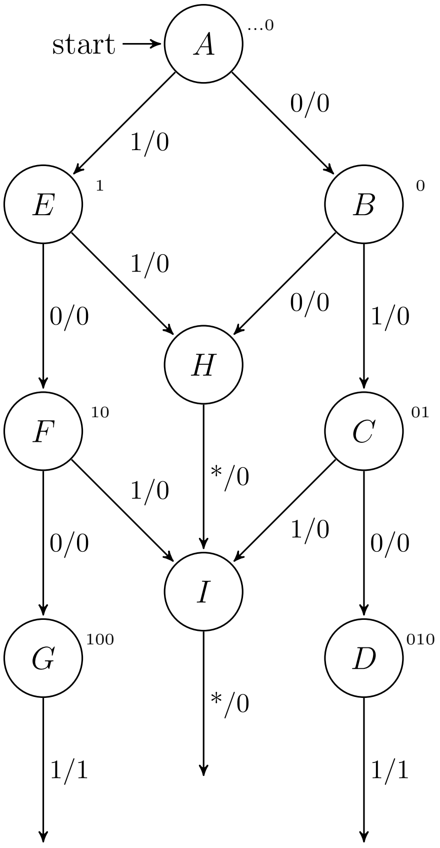
x:	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	...
z:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	...

Find a Mealy state diagram for this problem.

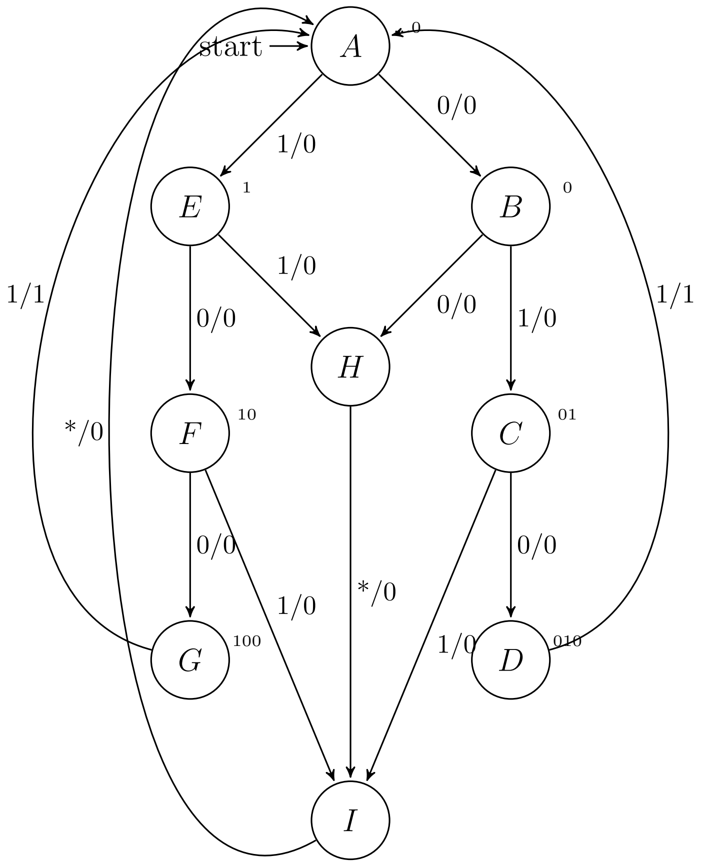
Solution:
In this circuit, the input is broken into groups of 4 bits. So, we need to receive at least 3 bits and then decide what the output should be based on the next bit received. Starting with our sequences to be detected, we would get the partial state diagram as follows where the initial state represents "no input received".



Next, we can add additional states that wait for 3 bits of input that do **not** lead to an output of 1 — this will cover the situation of all possible 8 patterns of the forst 3 bits. So, H represents the two bit patterns 11 or 00 and I represents all three bit patterns which are not 100 or 010; that is all 3 bits that cannot possibly lead to a 1 output.



Finally, all edges from D, G and I need to go back to the initial state — because the circuit recognizes the input in groups of 4 bits, the circuit needs to effectively reset itself. The complete state diagram:


start
A
0
1/0
0/0
E
1
B
0
1/1
1/0
0/0
1/1
0/0
1/0
H
*/0
F
10
C
01
0/0
0/0
1/0
*/0
1/0
G
100
D
010
I

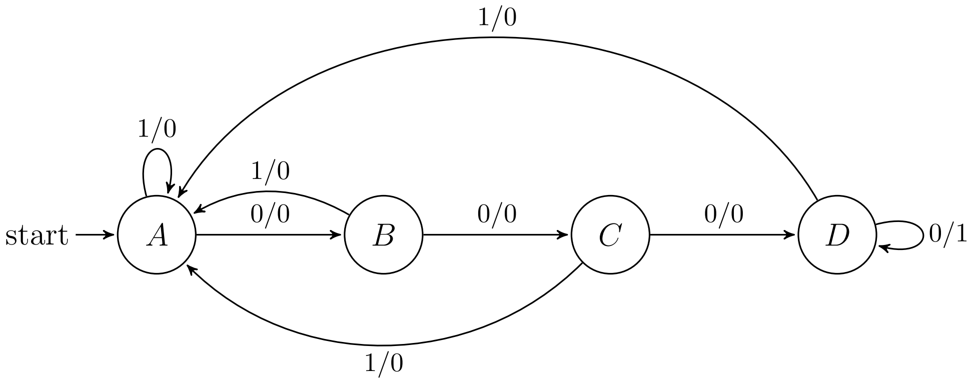
Q6: A sequential circuit has two inputs x and y and one output z. Its function is to compare the input sequences on the two inputs. If $x = y$ during any four consecutive clock cycles, the circuit produces $z = 1$, otherwise $z = 0$. An example of the desired behaviour is:

x:	0	1	1	0	1	1	1	0	0	0	1	1	0	...
y:	1	1	1	0	1	0	1	0	0	0	1	1	1	...
z:	0	0	0	0	1	0	0	0	0	1	1	1	0	...

Find a state diagram and state table for this problem.

Solution:
The clever way to tackle this problem is to define a new signal $w = x \oplus y$. Then, if expressed in terms of w, if during any four consecutive clock cycles $w = 0$, then the circuit produces $z = 1$, otherwise $z = 0$. As soon as $w = 1$, the circuit needs to begin again.

The completed state diagram is:



The state table is:

Current state	**Next state**		**Output** (z)	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	B	A	0	0
B	C	A	0	0
C	D	A	0	0
D	D	A	1	0

Q7: A sequential circuit has two inputs x and y and one output z. the output remains a constant value unless one of the following input sequences occurs:

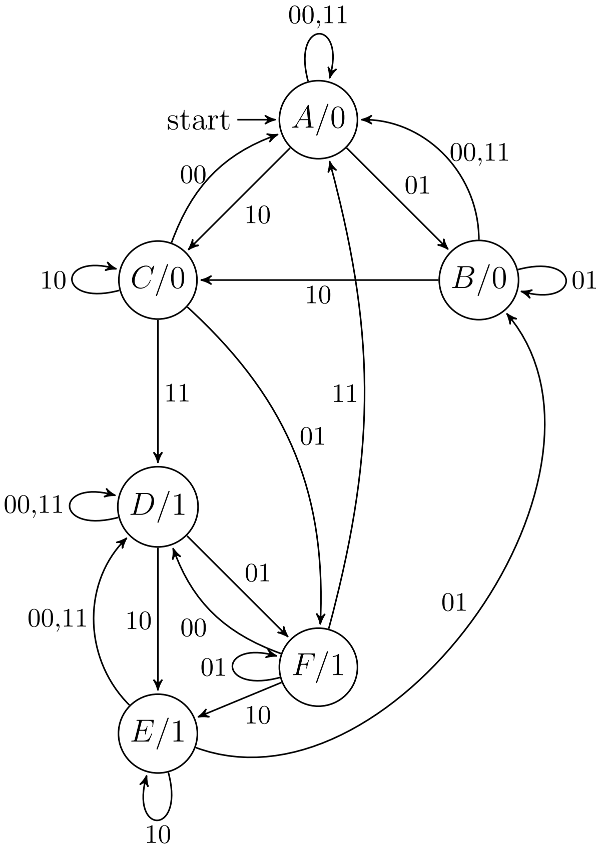
(a) The input sequence $xy = 01, 11$ causes the output to become 0;

(b) The input sequence $xy = 10, 11$ causes the output to become 1;

(c) The input sequence $xy = 10, 01$ causes the output to change value.

The notation $xy = 01, 11$ means $x = 0, y = 1$ followed by $x = 1, y = 1$. Derive a Moore state diagram for this problem.

Solution:
In this particular problem it's useful to think of a state as what the current inputs are and what the output should be. This leads to the following state diagram:



So, the explanation of the states would be, for example, that state C represents the situation when the last inputs were $xy = 10$ and the output is 0. State F would represent the situation when the last inputs were 01 and the output is 1 — therefore, there is an edge from $C \rightarrow F$ to represent one example of condition (c) from the problem statement.

Q8: A sequential circuit has one input x and two outputs z_1 and z_2. The output $z_1 = 1$ occurs every time the input sequence 100 is completed, provided that the sequence 010 has never occurred. An output $z_2 = 1$ occurs every time the input sequence 010 is completed. Note that once $z_2 = 1$ has occurred, then $z_1 = 1$ can never occur but not vice versa. A typical input and output sequence is:

x:	1	0	0	1	1	0	0	1	0	1	0	1	0	0	1	0	1	1	0	1	0 ...
z_1:	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ...
z_2:	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	1 ...

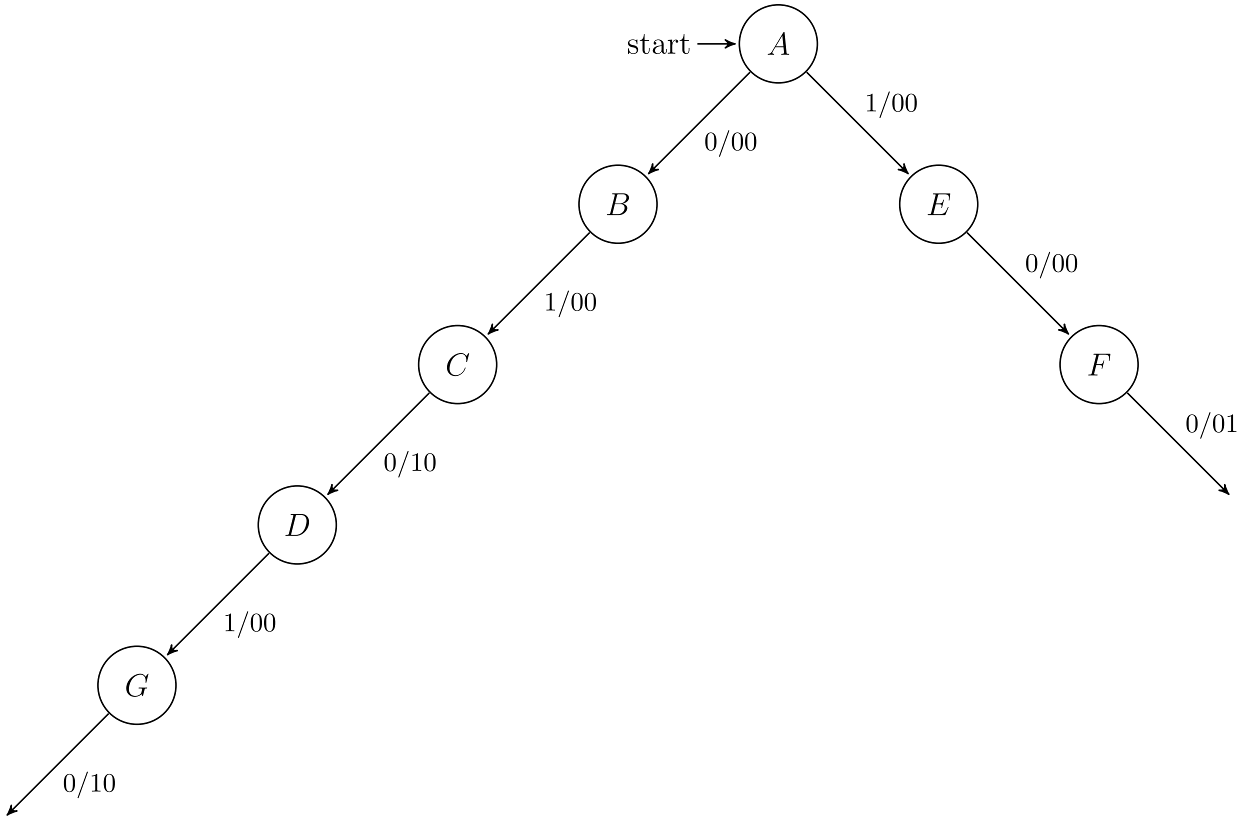
Find a Mealy state diagram and state table for this problem (8 states is possible).

Solution:

Outputs will be written as $z_2 z_1$ on the edges — therefore if an edge is labeled /01 is means that $z_2 = 0$ and $z_1 = 1$.

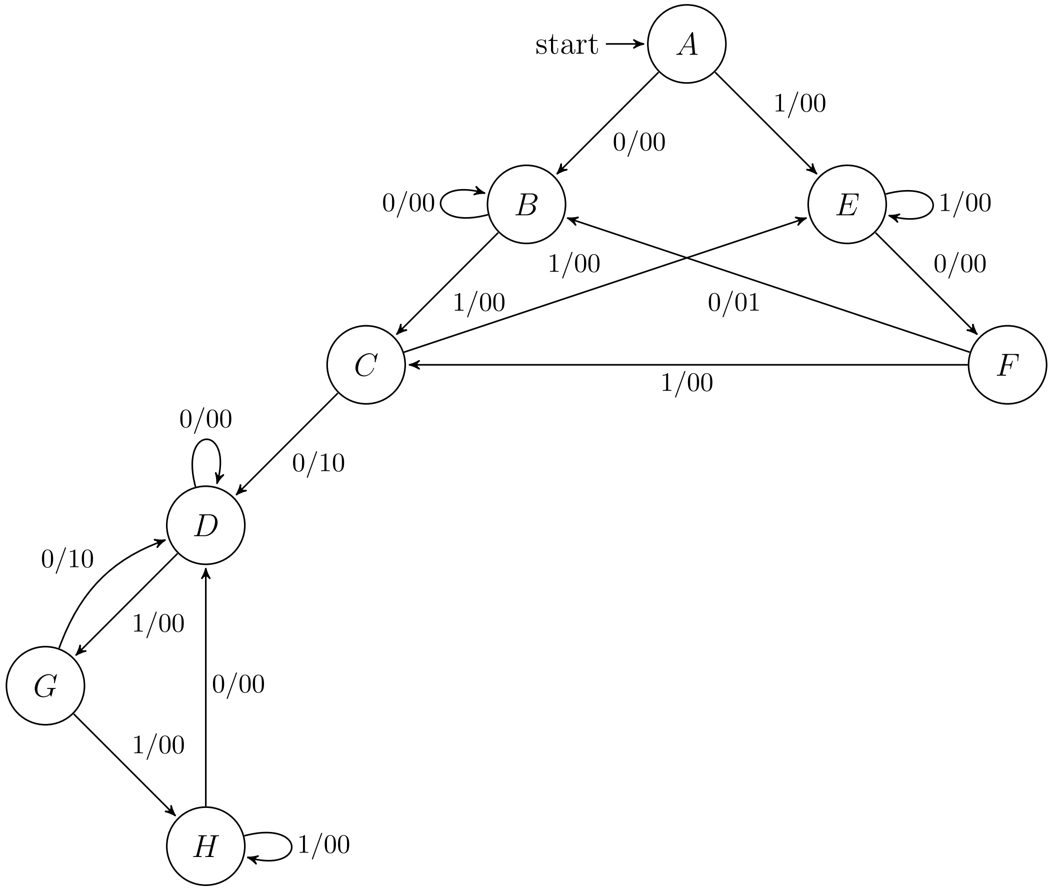
The solution is to simply detect patterns 100 and 010, but once 010 is detected, we can enter into a different part of the state machine in which we no longer detect 100.

This leads to the following partial state diagram which shows the detection of the two patterns as well as entry into a part of the state machine which no longer pays attention to the pattern 100:



Now we just need to add missing edges and redirect edges that mark the end of the detection of a sequence.

The completed state diagram is:



The state table is:

Current state	**Next state**		**Output** $(z_2 z_1)$	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	B	E	00	00
B	B	C	00	00
C	D	E	10	00
D	D	G	00	00
E	F	E	00	00
F	B	C	01	00
G	D	H	10	00
H	D	H	00	00

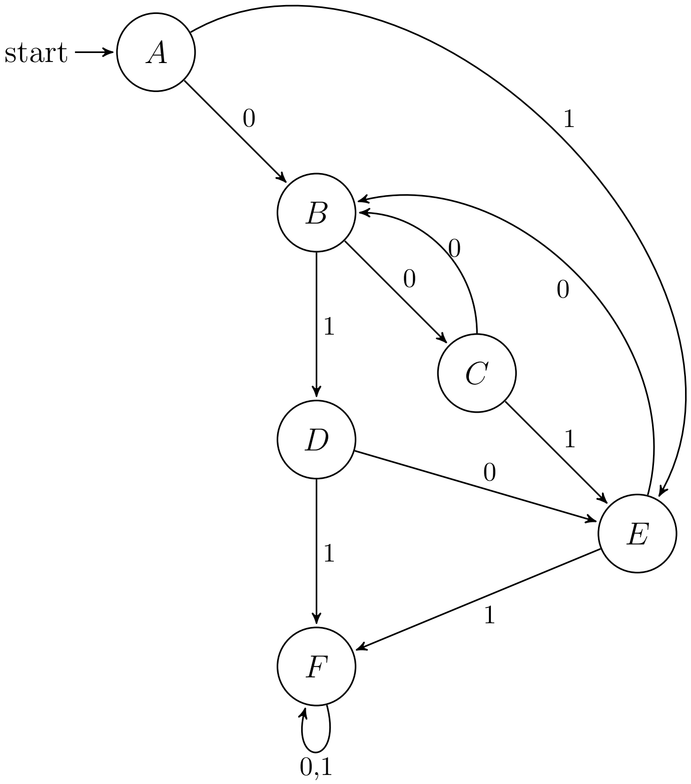
Q9: A clocked Moore sequential circuit should have output $z = 1$ if the total number of 0's received at input x is an even number greater than zero, provided that two consecutive 1's have never been received.

Derive a state diagram for this problem. Provide a description for each state (e.g., "s_1" means "odd number of 0's", etc.).

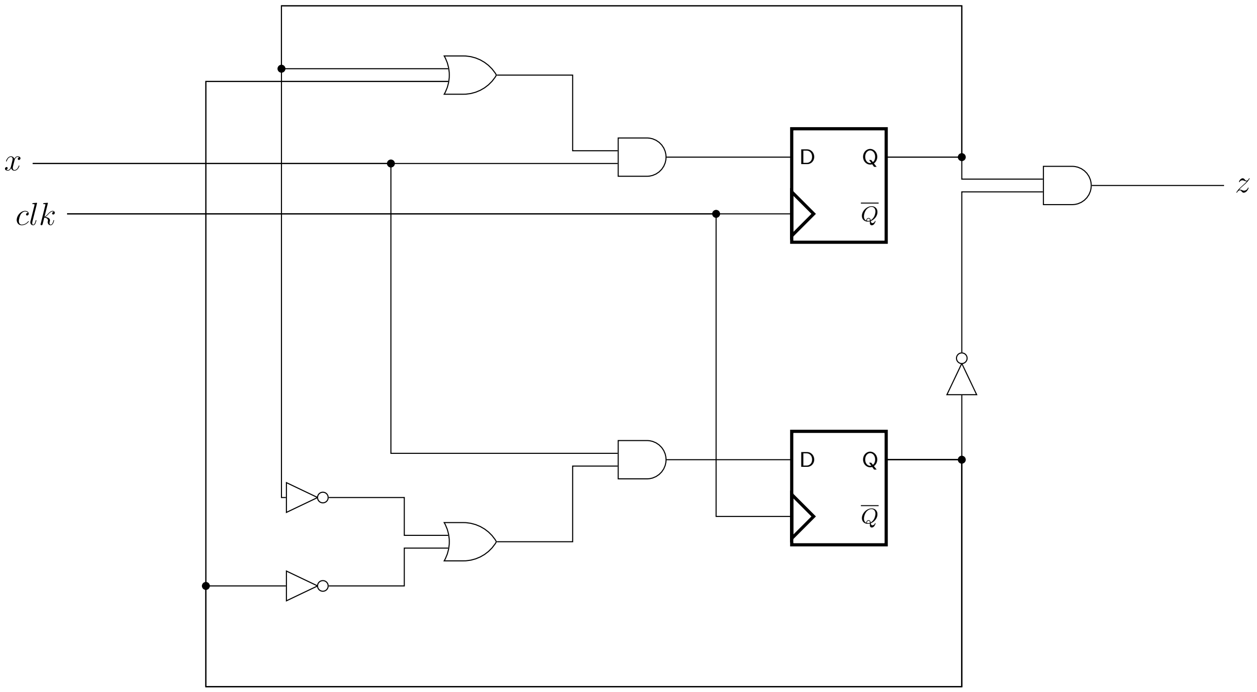
Solution:
A description of the states required:

A : initial state; nothing received
B : odd number of 0s received
C : even number of 0s received
D : odd number of 0s followed by a 1
E : even number of 0s followed by a 1
F : two consecutive 1s

The state diagram is:



Q10: Derive a state table and a state diagram for the circuit shown below. If you were told that this circuit was a sequence detector, what input sequence is detected by this circuit?



Solution:

Assume the top flip flop is $DFF0$ (output q_0 and input d_0) and the bottom flip flop is $DFF1$ (output q_1 and input d_1).

Write down the equation for the output:

$$z = q_0\overline{q}_1$$

Write down the flip flop input equations:

$$\begin{aligned} d_0 &= x(q_0 + q_1) \\ d_1 &= x(\overline{q}_0 + \overline{q}_1) \end{aligned}$$

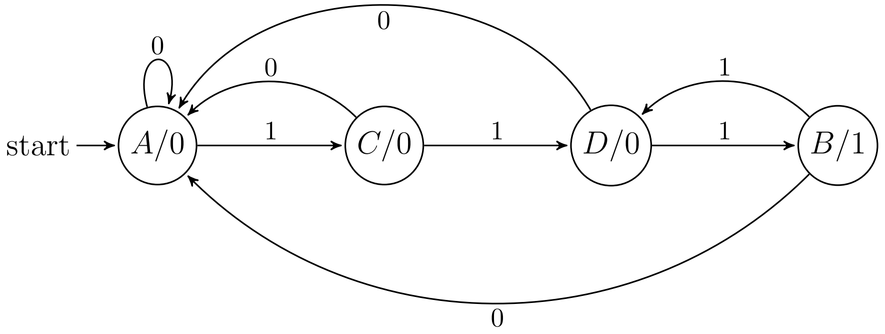
Two flip flops imply a maximum of 4 states. Label the states 00, 01, 10 and 11. Write down the state table (by first writing down the flip flop input values):

Current state (q_1q_0)	**DFF inputs** (d_1d_0)		**Next state** (q_1q_0)		**Output** (z)	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$	0	0
00	00	10	00	10	0	0
01	00	11	00	11	1	1
10	00	11	00	11	0	0
11	00	01	00	01	0	0

Let A be state 00, B be state 01, C be state 10, and D be state 11. The state table (symbolic) is:

Current state (q_1q_0)	**Next state** (q_1q_0)		**Output** (z)	
	$x = 0$	$x = 1$	0	0
A	A	C	0	0
B	A	D	1	1
C	A	D	0	0
D	A	B	0	0

The state diagram is (notice that the outputs are clearly a function of the state and hence a Moore state diagram):



I simply assumed that state A was the initial state.