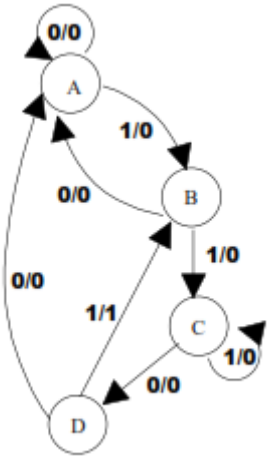
APPENDIX: SEQUENTIAL CIRCUIT DESIGN EXAMPLES

Example

Design a Mealy-type sequence detector, which outputs “1” if last four inputs are “1101”.

Example inputs and corresponding outputs:

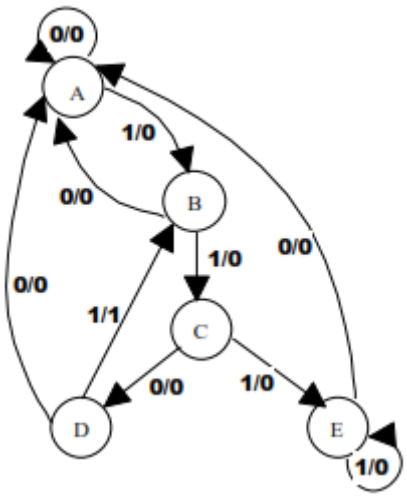
```
x: 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 0 1
z: 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1
```



Let us add an extra constraint: When there are three or more consecutive 1’s, then ignore these 1’s.

Example inputs and corresponding outputs:

```
x: 0 1 1 0 1 1 0 1 0 0 1 1 1 1 0 1 1 0 1
z: 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1
```

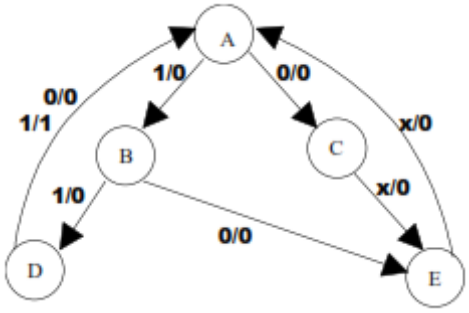


Example

Design a Mealy-type sequence detector, in which each consecutive three input is a group, and the output is “1” in case all the inputs in a group is “1”.

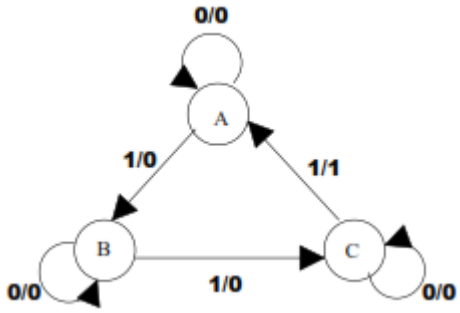
Example inputs and corresponding outputs:

x	1 0 1 1 1 0 1 1 1 0 1 1
z	0 0 0 0 0 0 0 0 1 0 0 0



Example

Design a non-overlapping Mealy-type sequence detector, in which every third “1” as input makes output “1”. 1’s do not need to be input one after another. There may be 0’s in between 1’s.



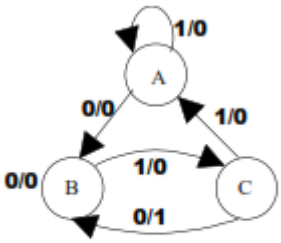
Example inputs and corresponding outputs:

x	0 1 1 1 0 1 0 1 0 1 1 0 1 0
z	0 0 0 1 0 0 0 0 0 1 0 0 0 0

Example

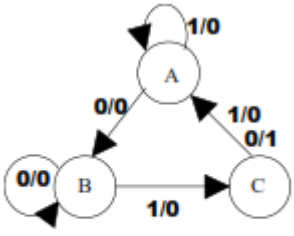
Design a Mealy-type sequence detector, which detects “010”.

a) Overlapping Version



x	1 1 0 1 0 1 0 1
z	0 0 0 0 1 0 1 0

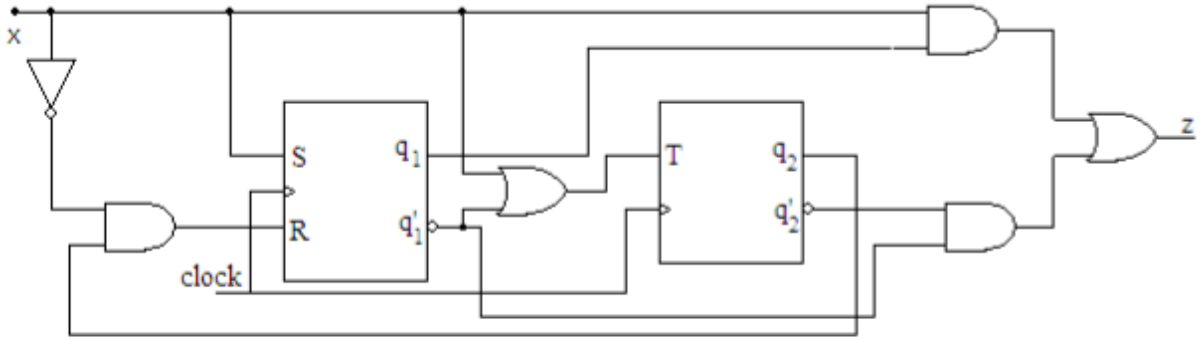
b) Non-Overlapping Version



x	0 1 0 1 0 1 0
z	0 0 1 0 0 0 1

Example

Draw the state diagram of the circuit below and assuming the initial state "$q_1\ q_2 = 0\ 0$", find the outputs for input sequence "0 0 1 1 0 0 1 0 0".



$S=x\ \ R=x'.q_2\ \ \ T=x+q_1'\ \ \ z=x.q_1+q_1'.q_2'$

Using these equations, we can derive the inputs of flip-flops and the output.

Then, we can find the next states of the flip-flops using the present state and inputs of the flip-flops.

Present State	**Next State**	**Inputs of Flip-Flops**	**Output**
x q_1 q_2	**Q_1 Q_2**	**S R T**	**z**
0 0 0	0 1	0 0 1	1
0 0 1	0 0	0 1 1	0
0 1 0	1 0	0 0 0	0
0 1 1	0 1	0 1 0	0
1 0 0	1 1	1 0 1	1
1 0 1	1 0	1 0 1	0
1 1 0	1 1	1 0 1	1
1 1 1	1 0	1 0 1	1

Finally, we can find the outputs. Remember that, the initial state is "0 0".

```
x :   0 0 1 1 0 0 1 0 0
q1: 0 0 0 1 1 1 1 1 0 0
q2: 0 1 0 1 0 0 1 1 0 1
z :   1 0 1 1 0 1 0 0 1
```

Example

Using the state table given below, find out the states of flip-flops and the outputs for the input sequence “0 1 0 0 1 1 1 0”. Assume that the initial state is “0 0”.

Present State q_1q_2	**Next State** Q_1Q_2		**Output** z	
	x=0	**x=1**	**x=0**	**x=1**
00	00	10	0	1
01	00	00	0	0
10	11	01	1	1
11	10	10	1	0

```
x :   0 1 0 0 1 1 1 0 ? ? ?
q1: 0 0 1 1 1 0 0 1 1 1 0 ?
q2: 0 0 0 1 0 1 0 0 1 0 0 0
z :   0 1 1 1 1 0 1 1 ? 1 ?
```

Example

Using the state table given below, find out the states of flip-flops and the outputs for the input sequence “0 1 0 1 0 1 1 1 0 1 0 0 0 0”. Assume that the initial state is “A”.

Present State q	**Next State** Q		**Output** z
	x=0	**x=1**	
A	A	B	1
B	D	C	1
C	D	C	0
D	A	B	0

```
x:   0 1 0 1 0 1 1 1 0 1 0 0 0 0 ? ?
q: A A B D B D B C C D B D A A A ? ?
z:   1 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1
```

Example

All the circuits we designed in the previous examples are supposed to get the input from another circuit's output. Working with the same clock signals, both circuits can operate in the same speed. Because in that case, input data will be synchronized with the clock signal.

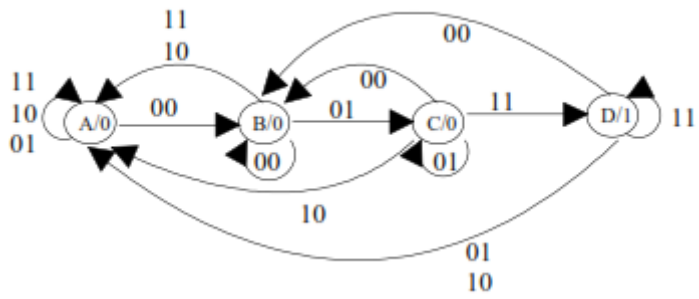
On the other hand, if we need a user to input data manually, we have to design the circuit in a different manner. Because users cannot input data at the same speed with the clock signal especially when the intervals between clock pulses are too short.

We are going to design the circuit in this example specifically for user input.

Design a Moore-type sequence detector, which has two inputs and an output. The circuit should output "1" when the input sequence is "00-01-11".

The point here is that the user will not be able to change the input as fast as the clock pulses. Which means that the sequence, which the circuit should detect, is "00-00-00-……-01-01-01-……-11-11-11-……" actually.

Let us derive the state diagram.



Let us derive the state table.

Present State q_1q_2	**Next State** Q_1Q_2 **00 01 10 11**	**Output** **z**
A	B A A A	0
B	B C A A	0
C	B C D A	0
D	B A D A	1

Let us assign the states and find out the inputs of flip-flops.

Present State q_1q_2	Next State Q_1Q_2 00 01 10 11	Inputs of Flip-Flops x_1x_2 = 00 J_1K_1	J_2K_2	x_1x_2 = 01 J_1K_1	J_2K_2	x_1x_2 = 11 J_1K_1	J_2K_2	x_1x_2 = 10 J_1K_1	J_2K_2	Output z
00 (A)	01 00 00 00	0 x	1 x	0 x	0 x	0 x	0 x	0 x	0 x	0
01 (B)	01 11 00 00	0 x	x 0	1 x	x 0	0 x	x 1	0 x	x 1	0
11 (C)	01 11 10 00	x 1	x 0	x 0	x 0	x 0	x 1	x 1	x 1	0
10 (D)	01 00 10 00	x 1	1 x	x 1	0 x	x 0	0 x	x 1	0 x	1

With the help of K-Maps, we can derive find out the equations.

$J_1 = q_2x_1'x_2 \quad K_1 = q_1x_2' + q_1q_2'x_1'$

$J_2 = x_1'x_2' \quad K_2 = x_1$

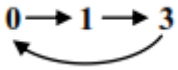
Since it is a Moore-type circuit, the output (z) depends only on the present state of the flip-flops.

$z = q_1q_2'$

Example

In this example, we are going to design an unordered counter, which has don't-care condition.

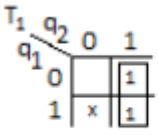
Design a counter with T-type flip-flops, which counts as shown below.



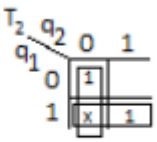
Since this circuit has three states, and we need to output the number 3 (binary 11), we are going to need two flip-flops.

Let us draw the state table and find out the equations for the inputs of flip-flops.

Present State q_1 q_2	Next State Q1 Q2	Inputs of Flip-Flops T1 T2
0 0	0 1	0 1
0 1	1 1	1 0
1 1	0 0	1 1
1 0	x x	x x

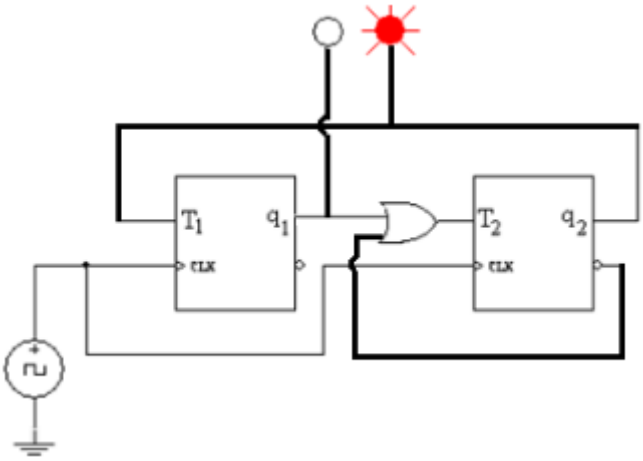


$T_1 = q_2$



$T_2 = q_2' + q_1$

Let us implement the circuit.



As you know, flip-flops' initial states are unknown. Therefore, there is a chance that the initial state of the circuit is "1 0" which is considered as a don't-care condition.

So, let us analyze what will happen if the initial state is "1 0".

In order to analyze this situation, we need to check the don't-care conditions on Karnaugh-maps. As seen above, we considered "x = 0" in the K-Map of T_1 and "x = 1" in the K-Map of T_2.

Let us put these "considered" 0's and 1's on the state table in place of x's.

Present State $q_1\ q_2$	**Next State** Q1 Q2	**Inputs of Flip-Flops** T1 T2
0 0	0 1	0 1
0 1	1 1	1 0
1 1	0 0	1 1
1 0	x x	0 1

Now, we can find out the next states of flip-flops in don't-care condition (10).

Present State $q_1\ q_2$	**Next State** Q1 Q2	**Inputs of Flip-Flops** T1 T2
0 0	0 1	0 1
0 1	1 1	1 0
1 1	0 0	1 1
1 0	1 1	0 1

The state table above tells us that, if the circuits starts with the don't-care state, the next state will be "1 1". After that, the counter will continue to count as we designed.

Example

As we mentioned before, counters usually do not have inputs and extra outputs. In these types of counters, we use the outputs of flip-flops as the output of the circuit.

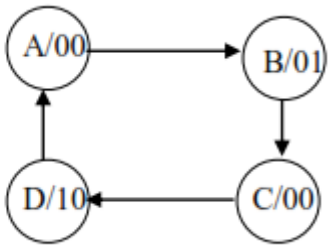
In this example, we are going to need an extra output (z) because there is no way that we can use the output of flip-flops as the output of the circuit.

Design a Moore-type counter, which counts as shown below.

0 – 1 – 0 – 2

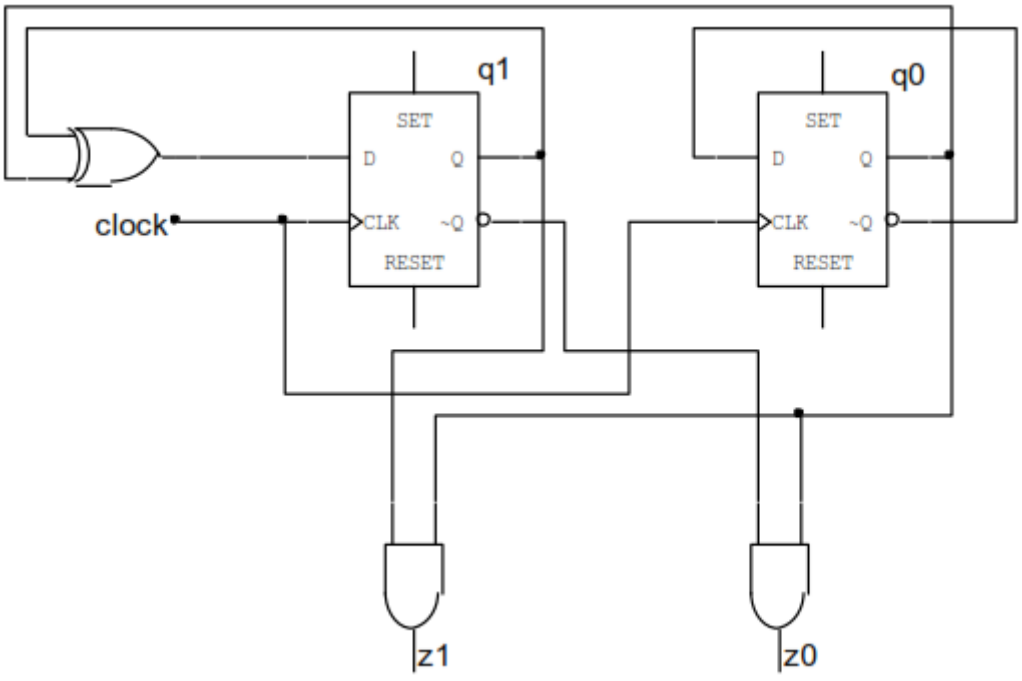
In this counter, there has to be four states and two extra outputs.

q1q0	Q1Q0	z1	z0
A	B	0	0
B	C	0	1
C	D	0	0
D	A	1	0



q1q0	Q1Q0	z1	z0	D1	D0
00	01	0	0	0	1
01	10	0	1	1	0
10	11	0	0	1	1
11	00	1	0	0	0

D1=q1⊕q0
D0=q0'
z1=q1.q0
z0=q1'q0

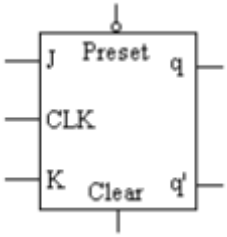


How to set the initial state of a flip-flop

As we mentioned before, flip-flops' initial states are unknown. However, it does not mean that we do not have any control on the initial states.

Flip-flops have extra two inputs, which can be used to set the initial state. These inputs are "Preset" and "Clear".

These inputs have higher priority than data inputs (JK, SR, D or T), and operate regardless of the clock signal. These inputs are classified as asynchronous inputs because they are not synchronized with the clock signal.



The "Preset" input has an inverter in the JK flip-flop above. If there is a bubble on the input, it indicates that the input has an inverter. This means that, the value "0" activates this input. In some cases, one of these inputs or both can have inverters.

"Preset" and "Clear" inputs cannot be active at the same time.

The table below shows the behavior of flip-flops when "Preset" and "Clear" are active or passive.

Preset	**Clear**	**Q**
Passive	Passive	Depends on the data input(s) JK, SR, D or T
Active	Passive	1 (Data inputs and the clock signal are ignored)
Passive	Active	0 (Data inputs and the clock signal are ignored)
Active	Active	NOT ALLOWED