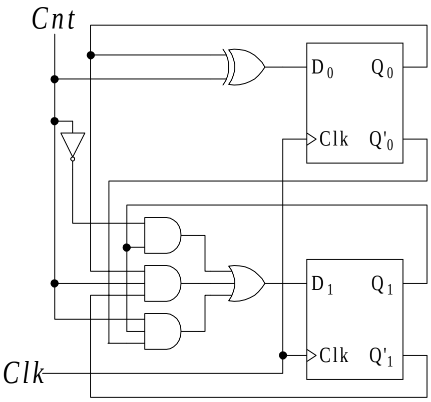
6.6 Analysis of Sequential Logic

Analysis is the process that starts with an implementation and generates the function or behavior of the sequential circuit. i.e. given a logic schematic, to generate one or more functional descriptions, using state diagrams, state and output tables, and input and output Boolean equations.

Synthesis, the reverse of analysis, starts with a behavioral description and generates an implementation.

Example 6.1: **Circuit with no outputs**. Derive the state table and state diagram for the (modulo-4 counter) sequential circuit represented by the following schematic.



Solution:

Step 1:

Derive **excitation equations**, i.e., boolean expressions for the inputs of each flip-flop in the schematic, in terms of the external input *Cnt* and the ff outputs Q_1 and Q_0. Since there are two ffs in our example, we derive two expressions for D_1 and D_0:

$$D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 2:

Derive the **next-state equations** by substituting the excitation equations into the flip-flop characteristic equations.

The **characteristic equations** formally describe the functional behavior of a latch or flip-flop. They specify the flip-flop’s next state as a function of its current state and inputs.

For the D flip-flop, the characteristic equation is

$$Q_{next} = D$$

Thus, the next-state equations are:

$$Q_{0next} = D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$Q_{1next} = D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 3a:

Derive the **next-state table** from the next-state equations.

Each row corresponds to a state of the sequential circuit which is defined by the binary values stored in its ffs.

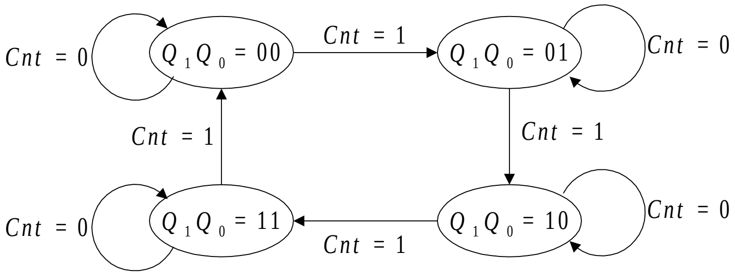
Each column represents one set of input values.

Each entry defines the value of the sequential circuit in the next clock cycle after the rising edge of the *Clk*.

Present State	Next State	
Q_1Q_0	$Q_{1\text{ next}}\, Q_{0\text{ next}}$	
	Cnt = 0	*Cnt* = 1
00	00	01
01	01	10
10	10	11
11	11	00

Step 3b:

Instead of a next-state table, we could use a **state diagram** to represent the behavior of the sequential circuit. A state diagram is basically a pictorial representation of the next-state table. It has exactly one node for each present state in the next-state table.

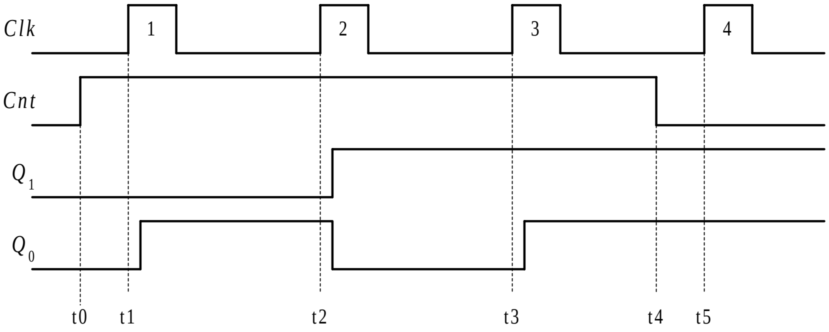


We can see that as long as *Cnt*=1, the sequential circuit visits the states in the sequence 0,1,2,3,0,1,2,…. When *Cnt*=0, the circuit stays in its present state until *Cnt* changes to 1, at which point the counting continues.

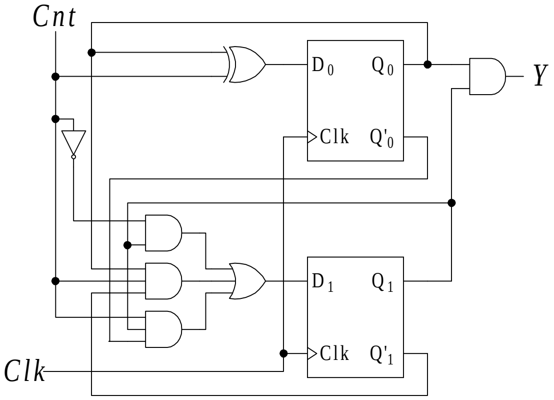
We conclude that the circuit is a modulo-4 counter with one control signal, *Cnt*.

Step 4:

The timing diagram is shown below:



Example 6.2: **State-based** or **Moore-type sequential circuit**. The output values depend solely on its present state. Derive the next state, the output tables, and the state diagram for the (modulo-4 counter) sequential circuit represented by the following schematic.



Solution:

Step 1 (Moore):

Derive **excitation equations**. (same as last example)

$$D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 2a (Moore):

Derive the **next-state equations**. (same as last example)

$$Q_{0next} = D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$Q_{1next} = D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 2b (Moore):

Derive the **output equation**.

$$Y = Q_1Q_0$$

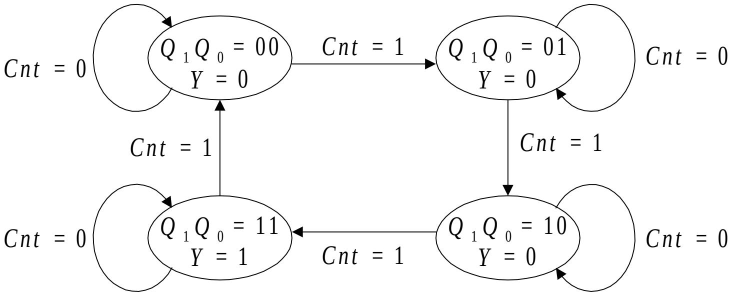
Step 3a (Moore):

Derive the **next-state/output table** from the next-state equations and output equation. In general, we add one column for each output signal to convert a next-state table to a next-state/output table.

Present State	Next State		Outputs
Q_1Q_0	$Q_{1\text{ next}}\,Q_{0\text{ next}}$		Y
	Cnt = 0	Cnt = 1	
00	00	01	0
01	01	10	0
10	10	11	0
11	11	00	1

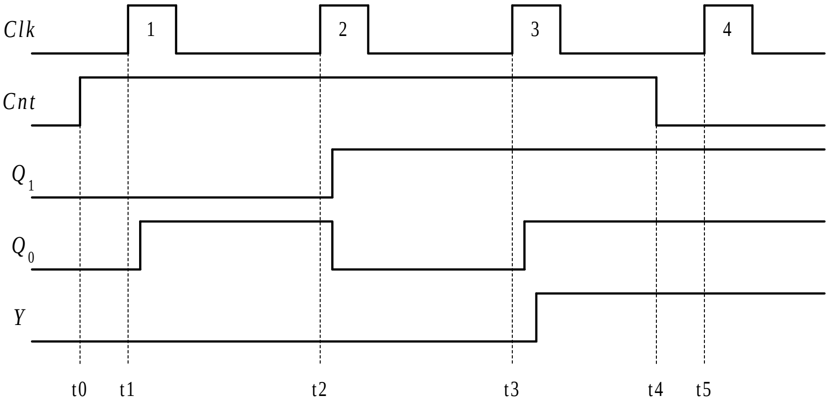
Step 3b (Moore):

Derive the **State diagram**.

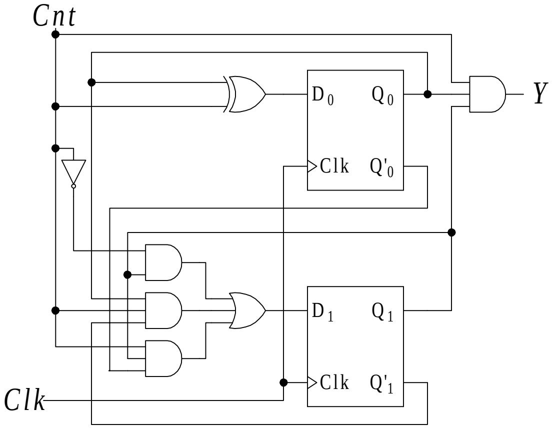


Step 4 (Moore):

The timing diagram is shown below:



Example 6.3: **Input-based** or **Mealy-type sequential circuit**. The output values are dependent on the input values as well as its present state. Derive the next state, the output tables, and the state diagram for the (modulo-4 counter) sequential circuit represented by the following schematic.



Solution:
Step 1 (Mealy):

Derive **excitation equations**. (same as first example)

$$D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 2a (Mealy):

Derive the **next-state equations**. (same as first example)

$$Q_{0next} = D_0 = Cnt \oplus Q_0 = Cnt'Q_0 + CntQ_0'$$
$$Q_{1next} = D_1 = Cnt'Q_1 + CntQ_1'Q_0 + CntQ_1Q_0'$$

Step 2b (Mealy):

Derive the **output equation**.

$$Y = CntQ_1Q_0$$

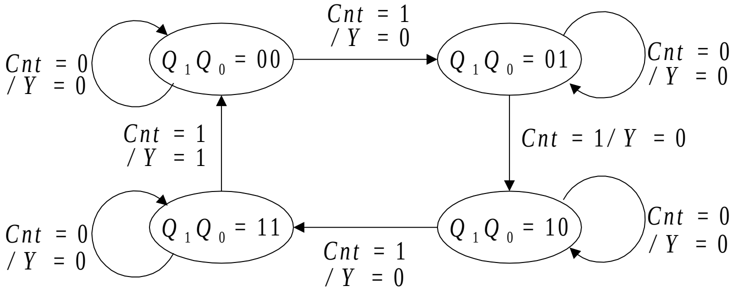
Step 3a (Mealy):

Derive the **next-state/output table**. Every entry in the next-state table will represent the next-state and the output value, separated by a slash (/).

Present State	Next State / Outputs	
Q_1Q_0	$Q_{1\ next}\,Q_{0\ next}$ / Y	
	Cnt = 0	Cnt = 1
00	00 / 0	01 / 0
01	01 / 0	10 / 0
10	10 / 0	11 / 0
11	11 / 0	00 / 1

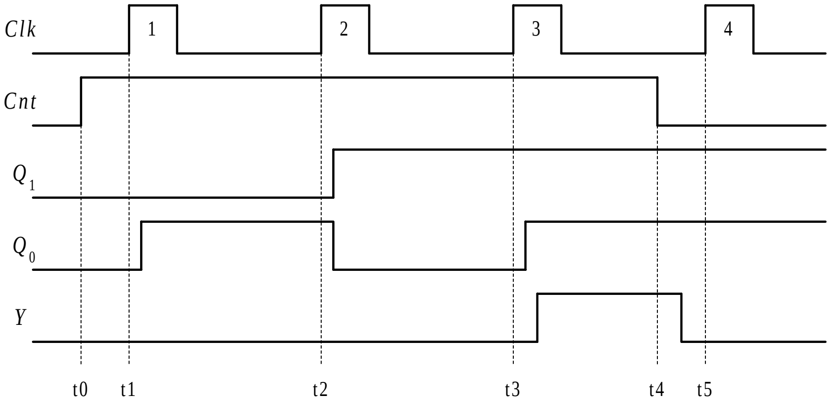
Step 3b (Mealy):

Derive the **State diagram**. The output is not associated with the state but with the transition arc. Each arc is labeled with both the input values that move the circuits from the present state to the next state, and the output values, which correspond to the input-signal values in the present state.



Step 4 (Mealy):

The timing diagram is shown below:



In clock cycle 3, the counter will be in state $Q_1Q_0 = 11$ and the output signal $Y = 1$. At t_4, $Y = 0$ because the input signal $Cnt = 0$ even though the counter is still in state $Q_1Q_0 = 11$.

6.7 Finite-State-Machine Model

Finite-State-Machine (**FSM**) is another name used for sequential logic circuit. The FSM can be defined abstractly as the quintuple

$$<S, I, O, f, h>$$

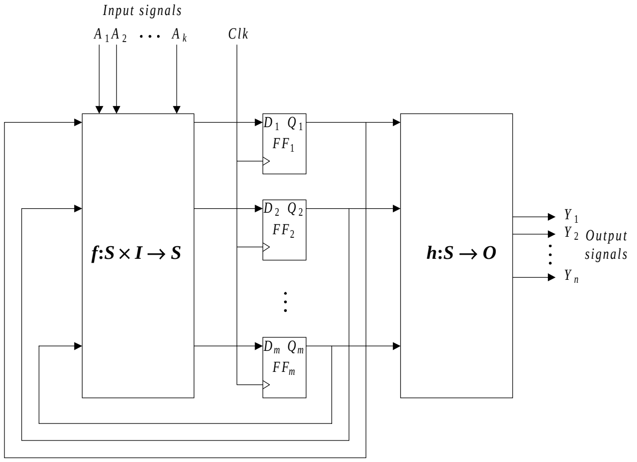
where *S* = set of states
I = set of inputs
O = set of outputs
f = next state function. $S \times I \rightarrow S$.
h = output functions. Determines the output values in the present state.

The FSM model assumes that time is divided into uniform intervals and that transitions from one state to another occur only at the beginning of each time interval. Therefore, the next-state function *f* defines what the state of the FSM will be in the next time interval given the state and input values in the present interval.

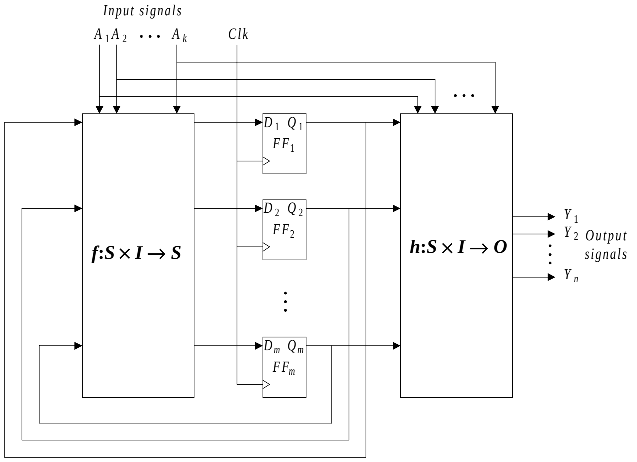
There are two different types of FSM, which correspond to two different definitions of the output function *h*.

1) **Moore FSM** or **state-based FSM**, for which *h* is defined as a mapping $\boldsymbol{S \rightarrow O}$.
2) **Mealy FSM** or **input-based FSM**, for which *h* is defined as a mapping $\boldsymbol{S \times I \rightarrow O}$.

The content of the flip-flops defines the state *S* of the FSM.
I, and *O* are binary signals.
f and *h* are defined by Boolean expressions that will be implemented with logic gates.



Moore FSM
state-based



Mealy FSM
input-based