Moore and Mealy Machines (Lecture 9)

Finite automata may have outputs corresponding to each transition. There are two types of finite state machines that generate output:

- Mealy Machine
- Moore machine

Types of FSM

1. Finite State Automata
   - With output
2. Mealy machine
   - produce output on transition
3. Moore machine
   - produce output on state

Mealy and Moore

• Both have:

– No final state

– Produce output from an input string

– No non-determinism

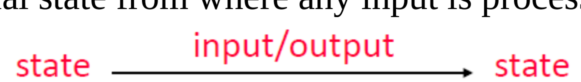
- Mealy machine inverts the input.
  - ie: if given 01101 outputs 10010

Mealy Machine

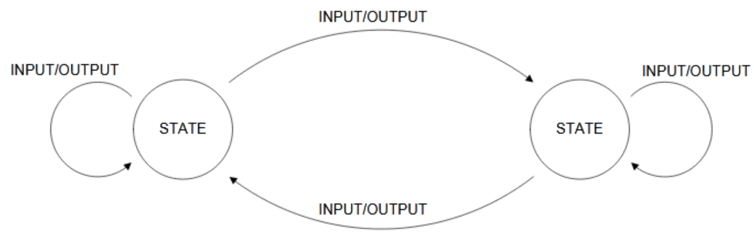
A Mealy Machine is an FSM whose output depends on the present state as well as the present input. In Mealy machine, every transition for a particular input symbol has a fixed output.

It can be described by a 6 tuple $(Q, \Sigma, O, \delta, X, q_0)$ where:

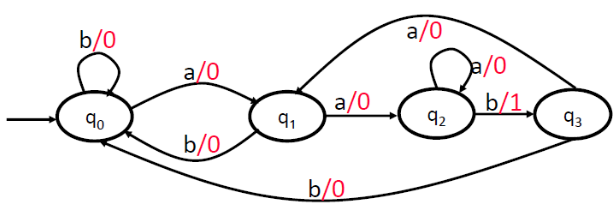
- **Q** is a **finite** set of states.
- **Σ** is a finite set of symbols called the input alphabet.
- **O** is a finite set of symbols called the output alphabet.
- **δ** is the input transition function where $\delta: Q \times \Sigma \rightarrow Q$
- **X** is the output transition function where $X: Q \times \Sigma \rightarrow O$
- $\mathbf{q_0}$ is the initial state from where any input is processed ($q0 \in Q$).



– Outputs determined by the current state and the current inputs.
– Outputs are conditional (directly dependent on input signals).



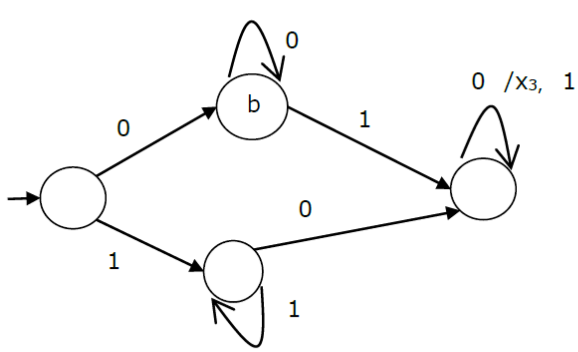
GENERIC MEALY STATE MACHINE



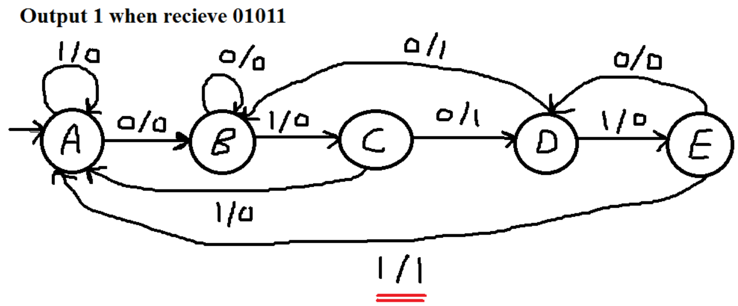
The state table of a Mealy Machine is shown below

Present state	Next state			
	input = 0		input = 1	
	State	Output	State	Output
→a	b	x_1	c	x_1
b	b	x_2	d	x_3
c	d	x_3	c	x_1
d	d	x_3	d	x_2

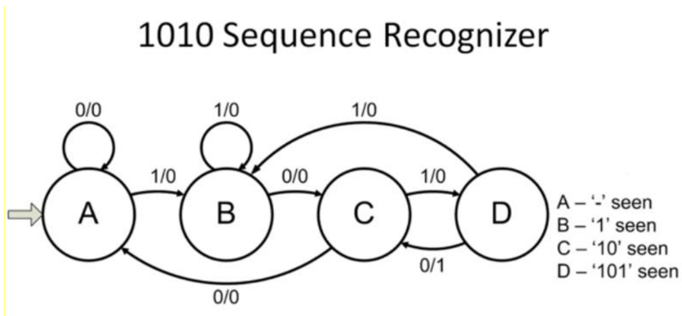
The state diagram of the above Mealy Machine is:



Example

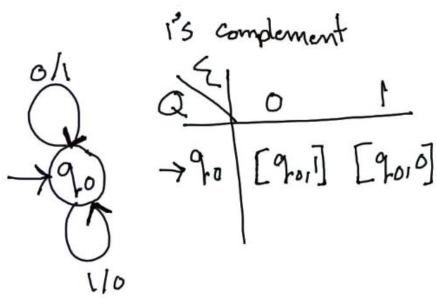


Example



Example

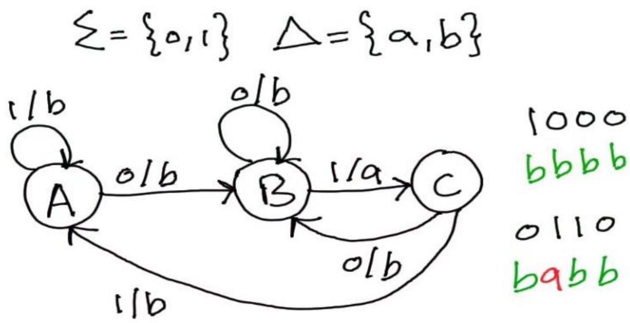
The following Mealy machine prints out the 1's complement of an input bit string.



If the input is 001010 the output is 110101. This is a case where the input alphabet and output alphabet are both {0,1}.

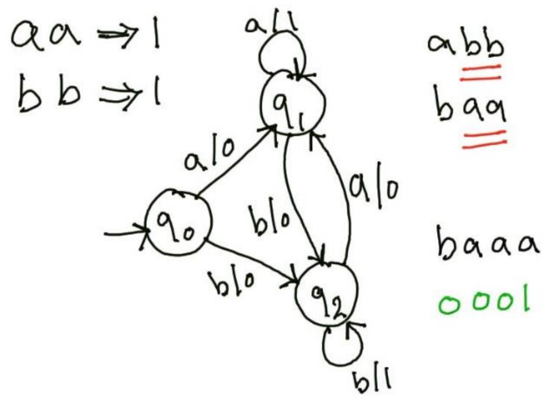
Example

Construct mealy machine that print 'a' whenever the sequence '01' is encountered in any input binary string.



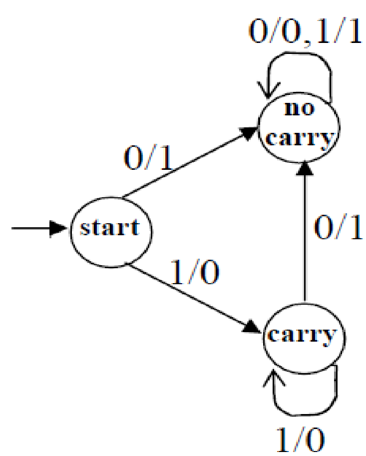
Example

Design Mealy machine that accepting the language consist of string form $\sum^{*}$ where $\sum$={a,b} and the string should end with either **aa** or **bb**.



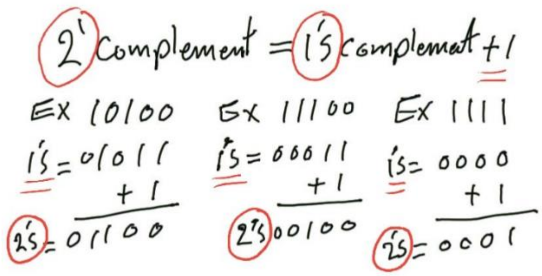
Example

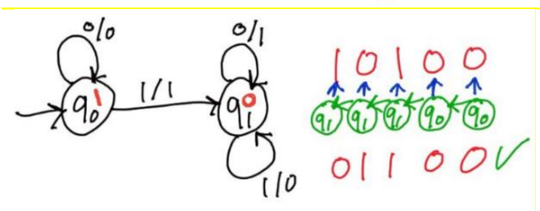
The following Mealy machine called the increment machine.

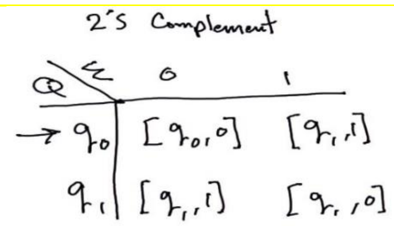


If the input is 1011 the output is 1100.

Example



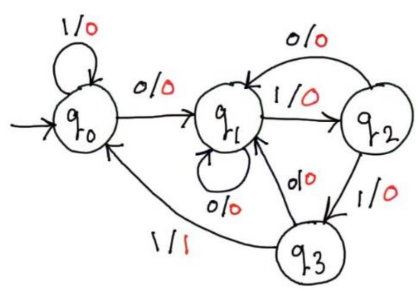




Q \ Σ	0	1
→ q_0	[q_0, 0]	[q_1, 1]
q_1	[q_1, 1]	[q_1, 0]

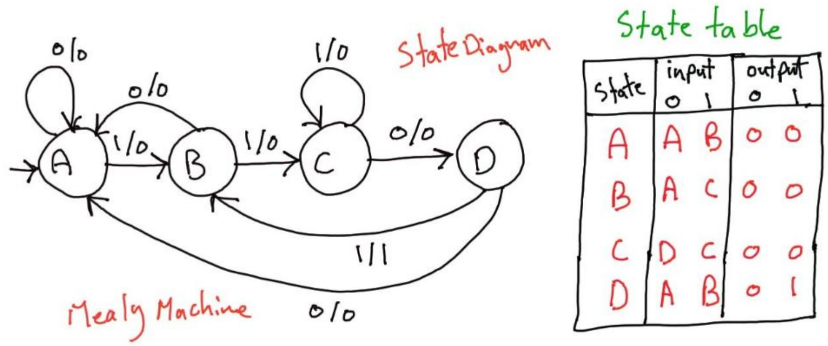
Example

0111 sequence detector



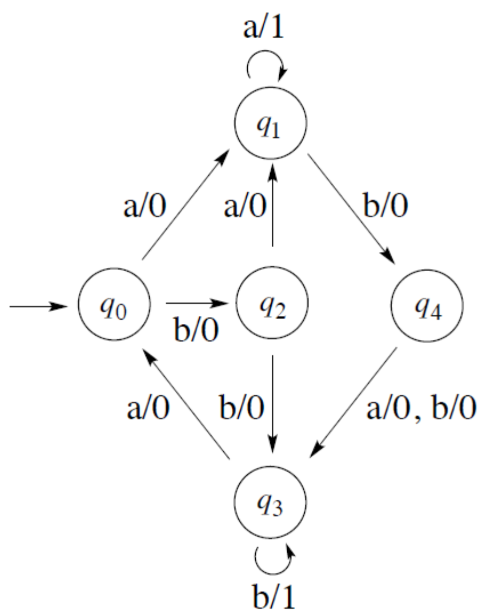
Example

Design a state diagram of the 1101sequence if its output goes to 1 when a target sequence has been detected.

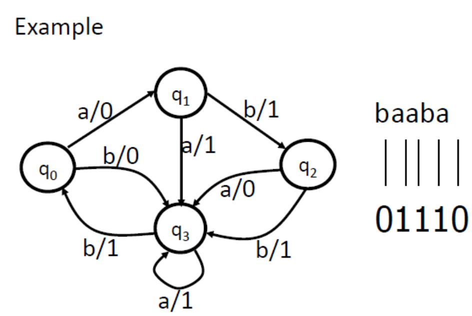


Example

- Σ= {ab}
- Γ = {01}
- States: *q0, q1, q2, q3, q4*

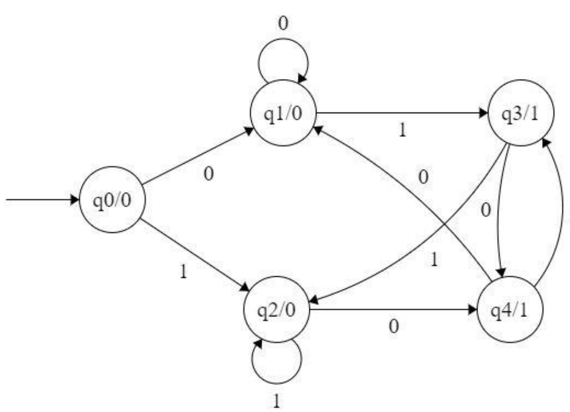


Input string:
aaaabbbbaabb
Output printed:
011100110000



Let us trace the running of the machine on the input sequence (**aaabb**).
We start in state **q0**, the first input letter is an **a**, which takes us to **q1** and prints a **0**,
Second letter is **a**, which takes to **q3**, and prints a **1**.
Third letter is **a**, which loops us back to **q3**, and prints a **1**.
Forth letter is **b**, which takes us to **q0**, and prints a **1**.
Fifth letter is **b**, which takes us to **q3**, and prints a **0**.
The output string for this input is **01110**.

Example

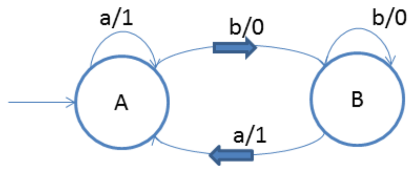


In the mealy machine shown in Figure 1, the output is represented with each input symbol for each state separated by /. The length of output for a mealy machine is equal to the length of input.

Input: 11

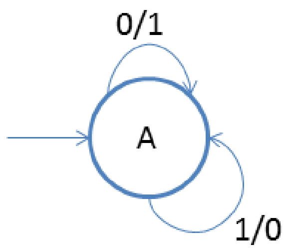
- Transition: δ (q0,11)=> δ(q2,1)=>q2
- Output: 00 (q0 to q2 transition has Output 0 and q2 to q2 transition ha s Output 0).

Example



Example

We will design mealy machine for 1's complement.



So we can see that every '0' will be replaced by '1' and vice-versa.

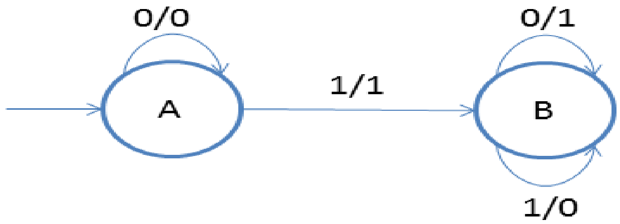
We will understand it more using one example.

Example

Suppose string is 10001 and we will start parsing from left to right. Every 0 will be replaced by 1 and vice versa.
So we will get the output as = 01110

Example

We will design mealy machine for 2's complement.



Generally we take 2's complement as follows:

1. Take 1's complement of the input
2. Add 1 to step 1

But here we are taking 2's complement in a different manner to design mealy machine.
The approach goes as follows:

1. Start from **right to left**
2. Ignore all 0's
3. When 1 comes ignore it and then take 1's complement of every digit

Example

1. Lets take 001 and we know that its 2's complement is (110+1 = 111)
2. So scan from right to left
3. On state A '1' came first to go to stage B and in output write 1
4. On state B replace '0' with '1' and vice-versa
5. So finally we got 111 as output
6. Be aware that the output is also printed in **right to left order**

Example

As an example, let $M = (Q, \Sigma, \Gamma, \delta, \lambda, q_I)$ such that:

$Q = \{q_1, q_2\}$, $q_I = q_1$.

$\Sigma = \{0, 1\}$

$\Gamma = \{\mathtt{E}, \mathtt{O}\}$

$\delta(q_1, 0) = q_1$, $\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2$, $\delta(q_2, 1) = q_1$

$\lambda(q_1, 0) = \mathtt{E}$, $\lambda(q_1, 1) = \mathtt{O}$

$\lambda(q_2, 0) = \mathtt{O}$, $\lambda(q_2, 1) = \mathtt{E}$,

This machine, which may also be represented as in figure 2.2, outputs an E if the number of 1s read so far is even and an O if it is odd; for example, the translation of 11100101 is OEOOOEEO.

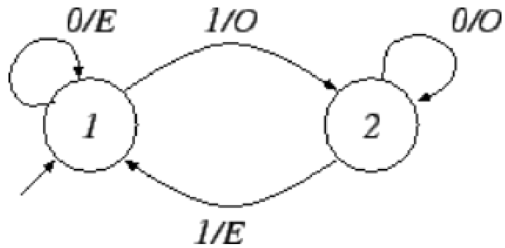
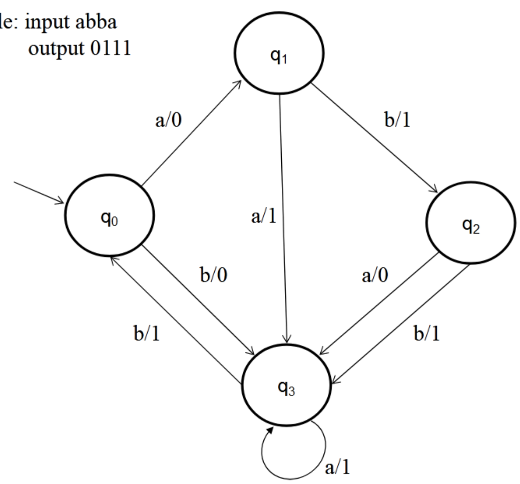
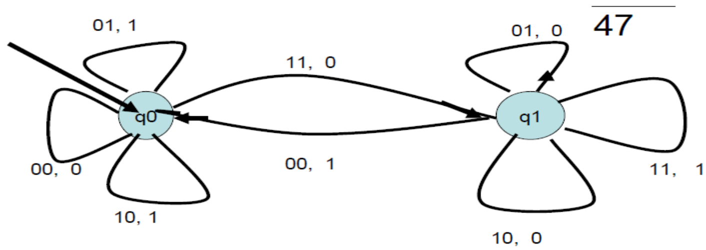


Figure 2.2: A Mealy machine that outputs an E if the number of 1s read so far is even and an O if it is odd. Transition labels are σ/γ where $\sigma \in \Sigma$ is the input and $\gamma \in \Gamma$ is the output.

Example: input abba
output 0111



Input 0 1 0 0 1 0 18
0 1 1 1 0 1 29
47

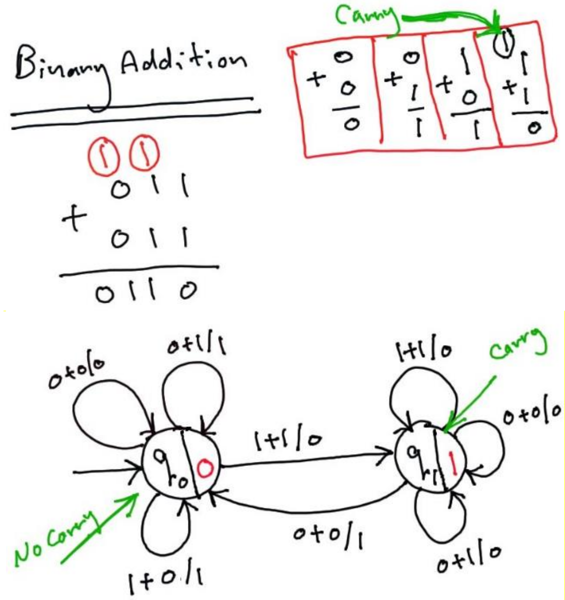


) - 01- q0 - 10 - q0 - 01 - q0- 01 - q0- 11 - q1 - 00- q0
1 1 1 1 0 1

ut put is : 1 0 1 1 1 1 value 47

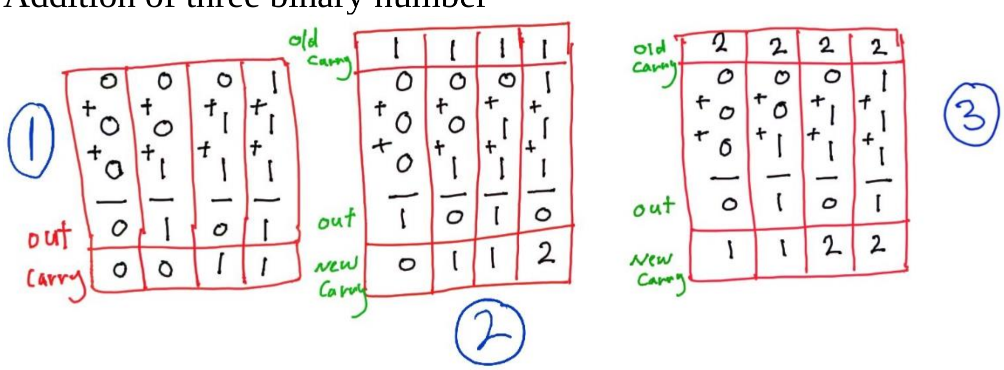
Example

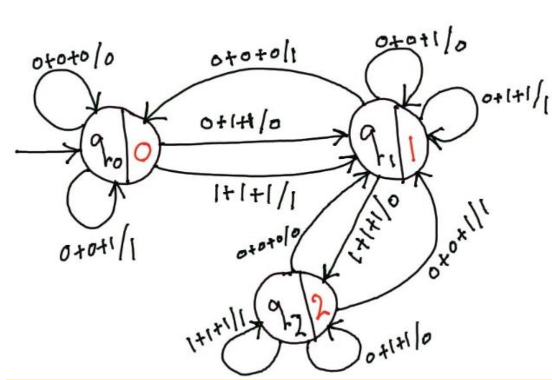
Addition of two binary number



Example

Addition of three binary number





Moore Machine

Moore machine is an FSM whose outputs depend on only the present state.
A Moore machine can be described by a 6 tuple (Q, Σ, O, δ, X, q0) where:

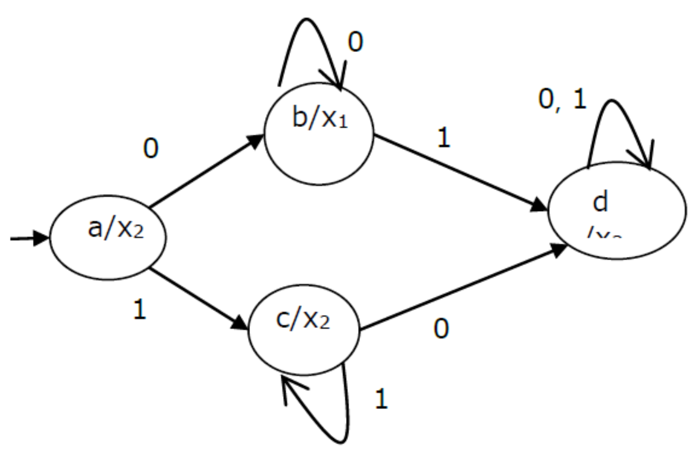
- **Q** is a finite set of states.
- **Σ** is a finite set of symbols called the input alphabet.
- is a finite set of symbols called the output alphabet.
- **δ** is the input transition function where δ: Q × Σ → Q
- **X** is the output transition function where X: Q → O
- **q0** is the initial state from where any input is processed (q0 ∈ Q).

$$(\text{state, letter from } \sum) \xrightarrow{\text{transition}} \text{state}$$

The state table of a Moore Machine is shown below:

Present State	Next State		Output
	Input = 0	Input = 1	
→ a	b	c	x_2
b	b	d	x_1
c	c	d	x_2
d	d	d	x_3

The state diagram of the above Moore Machine is:



Mealy Machine vs. Moore Machine

The following table highlights the points that differentiate a Mealy Machine from a Moore Machine.

Mealy Machine	Moore Machine
Output depends both upon present state and present input.	Output depends only upon the present state.
Generally, it has fewer states than Moore Machine.	Generally, it has more states than Mealy Machine.
Output changes at the clock edges.	Input change can cause change in output change as soon as logic is done.
Mealy machines react faster to inputs	In Moore machines, more logic is needed to decode the outputs since it has more circuit delays.

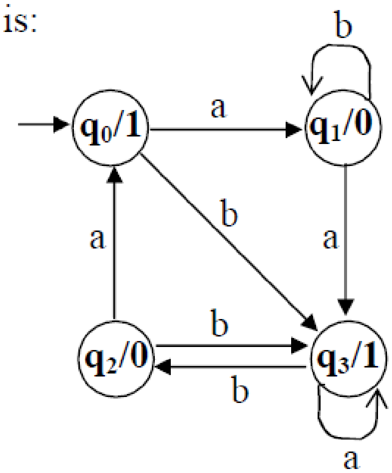
- ► A Moore machine does not define a language of accepted words, since every input string creates an output string and there is no such thing as a final state. The processing is terminated when the last input letter is read and the last output character is printed.

Example

Input alphabet: $\sum = \{a, b\}$
Output alphabet: $\Gamma = \{0, 1\}$
Names of states: q0, q1, q2, q3. (q0 = start state)

Old state	**Transition table** New state After input a	after input b	**Output table** (the character printed in the old state)
-q0	q1	q3	1
q1	q3	q1	0
q2	q0	q3	0
q3	q3	q2	1

The Moore machine is:



Let us trace the operation of this machine on the input string (abab):

String with q0 → print 1

Read a q1 → print 0

Read b q1 → print 0

Read a q3 → print 1

Read b q2 → print 0

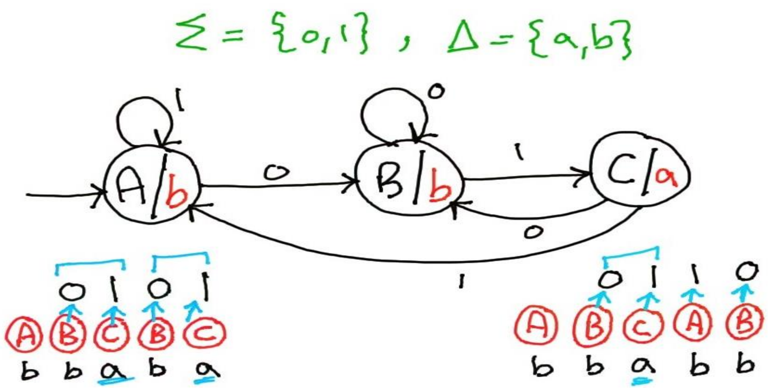
The output 10010

***Note*:** the two symbols inside the circle are separated by a slash "/", on the left side is the name of the state and on the right is the output from that state.

- If the input string is abab to the Moore machine then the output will be 10010.

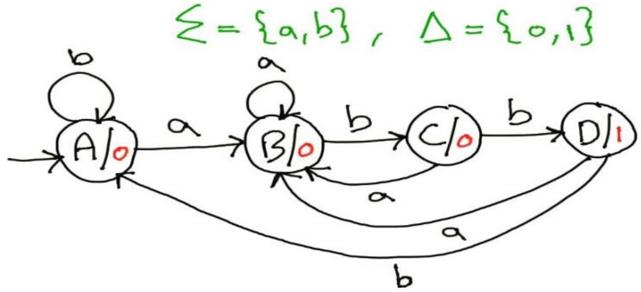
Example

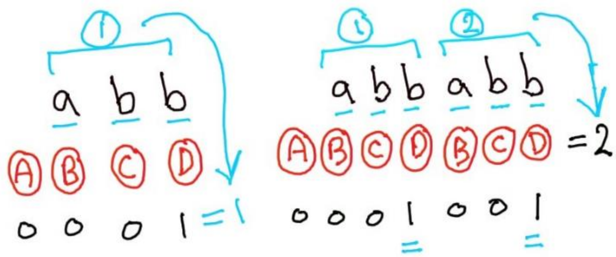
Construct mealy machine that print 'a' whenever the sequence '01' is encountered in any input binary string.



Example

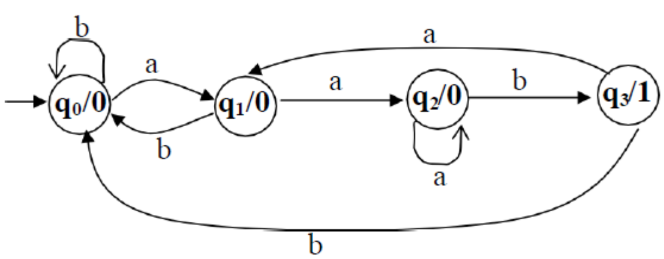
Construct Moore machine that count the occurrence of the sequence ‘abb’ in any input binary string over {a,b}.





Example

The following Moore machine will "count" how many times the substring aab occurs in a long input string.



The number of substrings aab in the input string will be exactly the number of 1's in the output string.

Input string		a	a	a	b	a	b	b	a	a	b	b
State	q_0	q_1	q_2	q_2	q_3	q_1	q_0	q_0	q_1	q_2	q_3	q_0
Output	0	0	0	0	1	0	0	0	0	0	1	0

Example

For the following Moore Machine the input alphabet is ∑= {a,b} and the output is Δ={0,1}. Run the following input sequences and find the respective outputs:

1-aabab 2-abbb 3-ababb

States	a	b	Outputs
→ q_0	q_1	q_2	0
q_1	q_2	q_3	0
q_2	q_3	q_4	1
q_3	q_4	q_4	0
q_4	q_0	q_0	0

1- a a b a b

q_0 q_1 q_2 q_4 q_0 q_2

0 0 1 0 0 1

2- a b b b

q_0 q_1 q_3 q_4 q_0

0 0 0 0 0

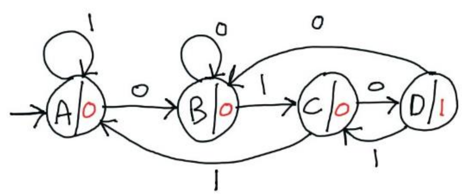
3- a b a b b

q_0 q_1 q_3 q_4 q_0 q_2

0 0 0 0 0 1

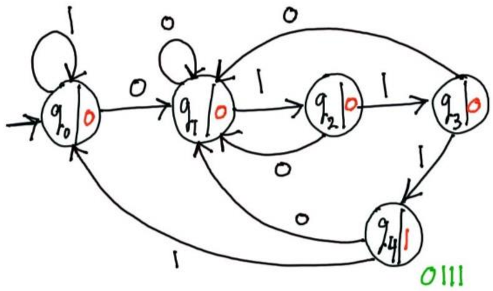
Example

010 sequence detector



Example

0111 sequence detector



Definition

Given the Mealy machine Me and the Moore machine Mo, which prints the automatic start-state character x, we will say that these two machines are equivalent if for every input string the output string from Mo is exactly x concatenated with the output from Me.

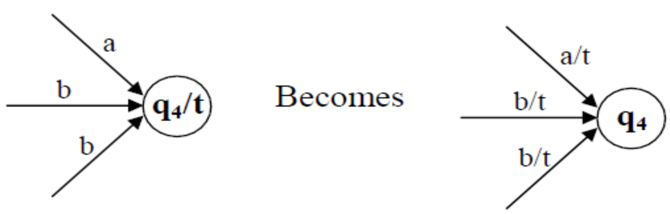
Note: we prove that for every Moore machine there is an equivalent Mealy machine and for every Mealy machine there is an equivalent Moore machine. We can then say that the two types of machine are completely equivalent.

Theorem

If Mo is a Moore machine, then there is a Mealy machine Me that is equivalent to it.

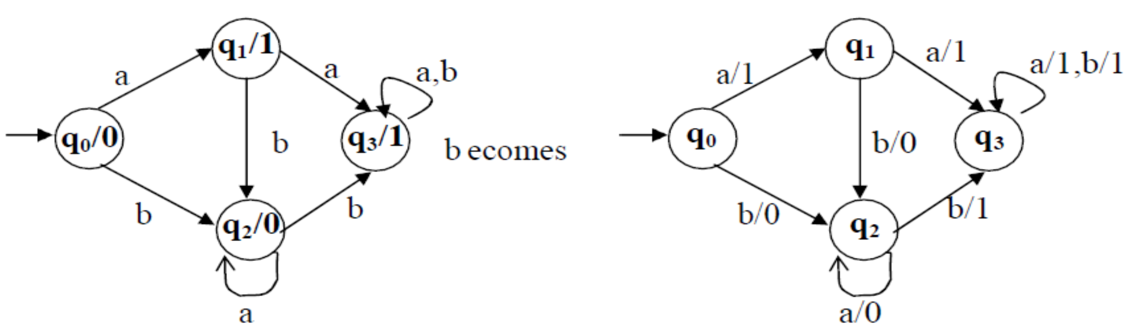
Proof

The proof will be by constructive algorithm.



Example

Below, a Moore machine is converted into a Mealy machine:

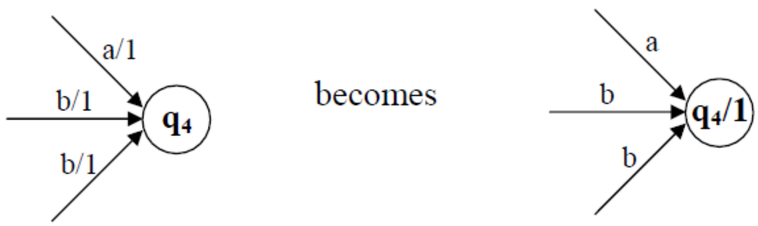


Theorem

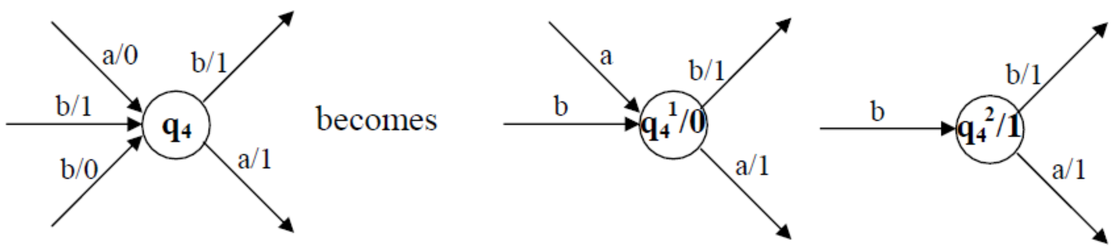
For every Mealy machine Me there is a Moore machine Mo that is equivalent to it.

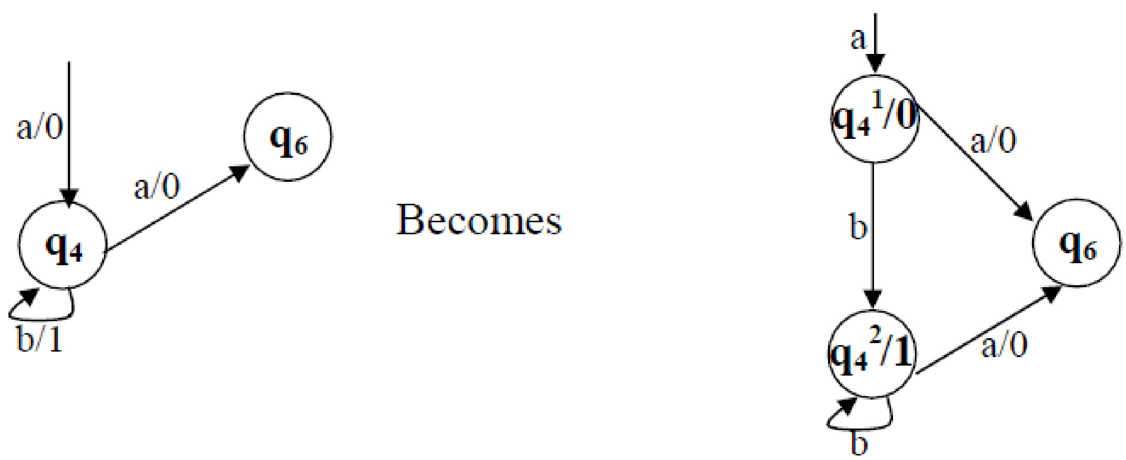
Proof

The proof will be by constructive algorithm.



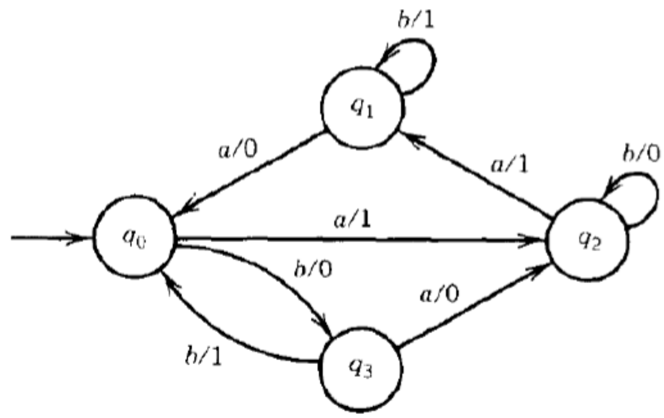
If there is more than one possibility for printing as we enter the state, then we need a copy of the state for each character we might have to print. (we may need as many copies as there are character in Γ).



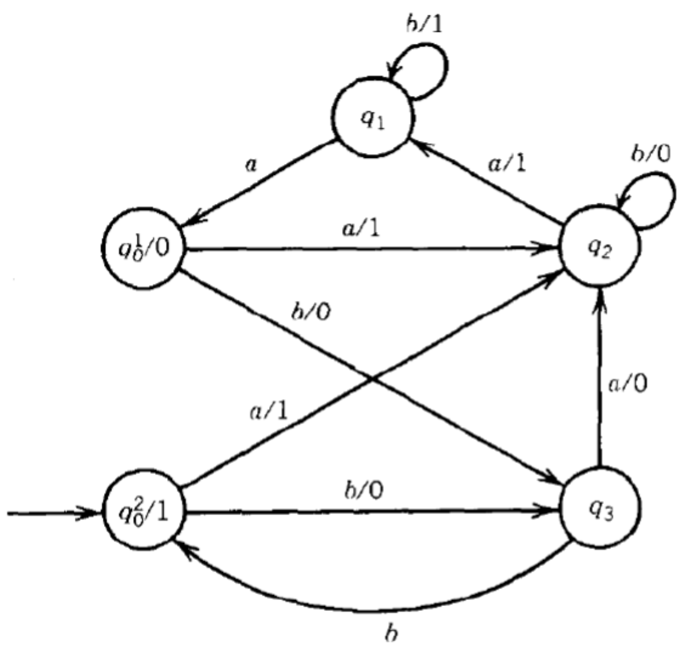


EXAMPLE

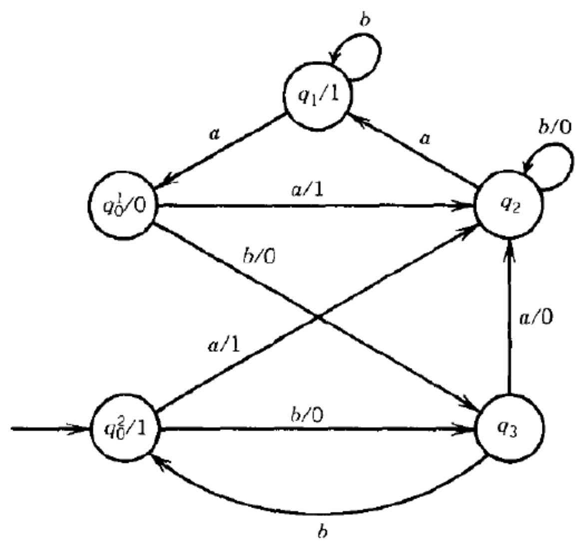
Let us start with the following Mealy machine:



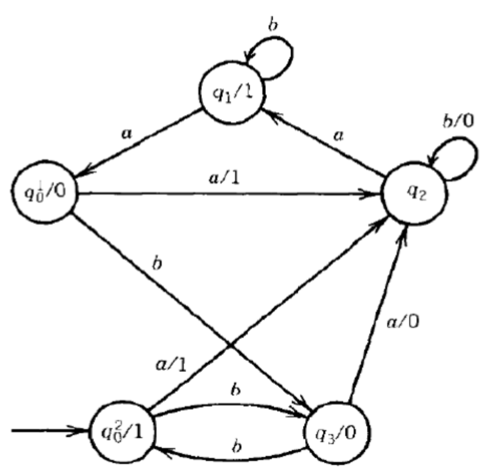
q0 has two edges come into this state with different labels, therefor we need two copies of this state.



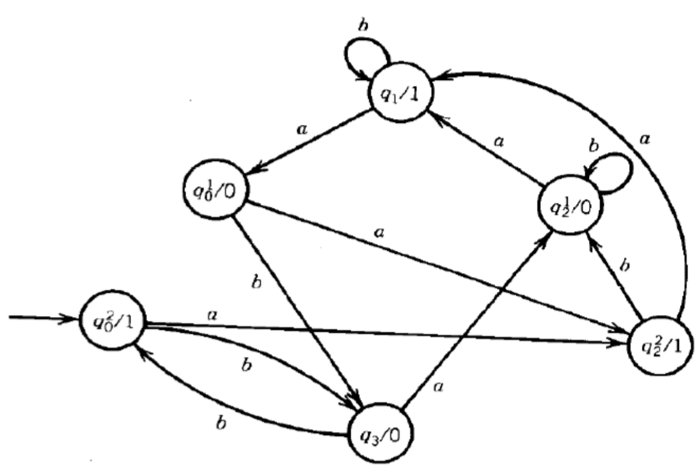
q1 has two edges with same printing instruction



Let us continue the conversion. State q_3 is easy to handle. Two edges come into it, both labeled $b/0$, so we change the state to $q_3/0$ and simplify the edge labels to b alone.

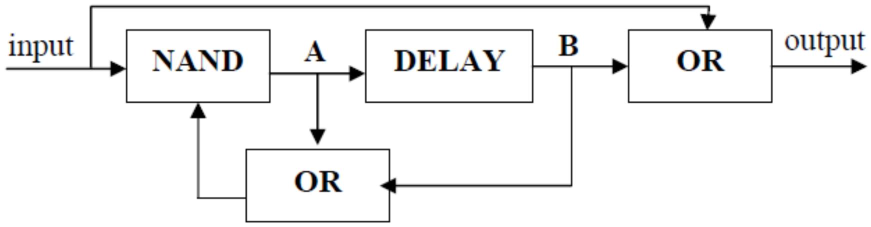


q2 has some 0 printing edge and some 1 printing edge with loop



Example

Draw the Mealy machine for the following sequential circuit:



First, we identify four states:

q0 is A= 0 B= 0
q1 is A= 0 B= 1
q2 is A= 1 B= 0
q3 is A= 1 B= 1

The operation of this circuit is such that after an input of 0 or 1 the state changes according to the following rules:

new B= old A

new A = (input) NAND (old A OR old B) output = (input) OR (old B)

Suppose we are in q0 and we receive the input 0.

new B = old A = 0

new A = 0 NAND (0 OR 0)
= 0 NAND 0
= 1

output = 0 OR 0 = 0

The new state is q2 (since new A=1, new B=0) if we are in state q0 and we receive the input 1:

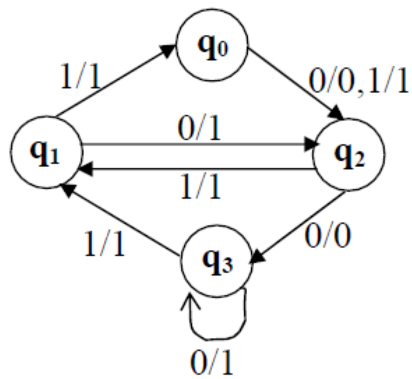
new B= old A = 0

new A = 1 NAND (0 OR 0) =1

output = 1 OR 0 =1 the new state is q2.

We repeat this process for every state and for each input to produce the following table:

Old state	After input 0		After input 1	
	New state	Output	New state	Output
q0	q2	0	q2	1
q1	q2	1	q0	1
q2	q3	0	q1	1
q3	q3	1	q1	1



Mealy machine

Comparison table for automata

	FA	TG	NFA	NFA- Λ	Moore	Mealy
Start states	one	One or more	one	one	one	one
Final states	Some or none	Some or none	Some or none	Some or none	none	none
Edge labels	Letters from ∑	words from ∑*	Letters from ∑	Letters from ∑ or Λ	Letter from ∑	i/o i from ∑ o from Γ
Number of edges from each state	One for each letter in ∑	arbitrary	arbitrary	arbitrary	One for each letter in ∑	One for each letter in ∑
deterministic	yes	no	no	no	yes	yes
output	no	no	no	no	yes	yes

Moore Machine to Mealy Machine

Algorithm

Input: Moore Machine

Output: Mealy Machine

Step 1 Take a blank Mealy Machine transition table format.

Step 2 Copy all the Moore Machine transition states into this table format.

Step 3 Check the present states and their corresponding outputs in the Moore Machine state table; if for a state Q_i output is m, copy it into the output columns of the Mealy Machine state table wherever Q_i appears in the next state.

Example

Let us consider the following Moore machine:

Present State	Next State		Output
	a = 0	**a = 1**	
→a	d	b	1
b	a	d	0
c	c	c	0
d	b	a	1

State table of a Moore Machine

Now we apply Algorithm to convert it to Mealy Machine.

Step 1 & 2:

Present State	Next State			
	a = 0		a = 1	
	State	Output	State	Output
→a	d		b	
b	a		d	
c	c		c	
d	b		a	

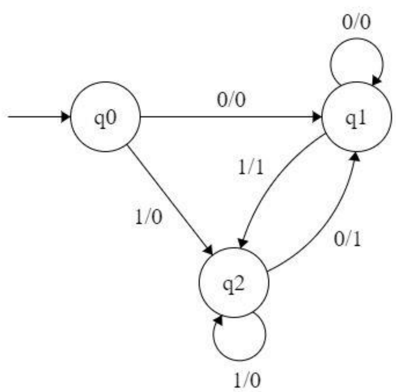
The partial state table after steps 1 and 2

Step 3:

Present State	Next State			
	a = 0		a = 1	
	State	Output	State	Output
=> a	d	1	b	0
b	a	1	d	1
c	c	0	c	0
d	b	0	a	1

State table of an equivalent Mealy Machine

Example



In the mealy machine shown in above Figure, the output is represented with each input symbol for each state separated by /.

The length of output for a mealy machine is equal to the length of input.

- **Input:**11
- **Transition:** δ (q0,11)=> δ(q2,1)=>q2
- **Output:** 00 (q0 to q2 transition has Output 0 and q2 to q2 transition also has Outpu t 0)

Let us take the transition table of mealy machine shown in above Figure.

	Input=0		Input=1	
Present State	Next State	Output	Next State	Output
q0	q1	0	q2	0
q1	q1	0	q2	1
q2	q1	1	q2	0

Step 1. First find out those states which have more than 1 outputs associated with them. q1 and q2 are the states which have both output 0 and 1 associated with them.
Step 2. Create two states for these states.For q1, two states will be q10 (state with output 0) and q11 (state with output 1). Similarly, for q2, two states will be q20 and q21.
Step 3. Create an empty Moore machine with new generated state. For Moore machine, Output will be associated toeach state irrespective of inputs.

	Input=0	Input=1	
Present State	Next State	Next State	Output
q0			
q10			
q11			
q20			
q21			

Step 4. Fill the entries of next state using mealy machine transition table shown in Table 1. For q0 on input 0, next state is q10 (q1 with output 0). Similarly, for q0 on input 1, next state is q20 (q2 with output 0).
For q1 (both q10 and q11) on input 0, next state is q10.
Similarly, for q1(both q10 and q11), next state is q21.
For q10, output will be 0 and for q11, output will be 1.
Similarly, other entries can be filled.

	Input=0	Input=1	
Present State	Next State	Next State	Output
q0	q10	q20	0
q10	q10	q21	0
q11	q10	q21	1
q20	q11	q20	0
q21	q11	q20	1

Table 3

This is the transition table of Moore machine.

Mealy Machine to Moore Machine

Algorithm :

Input: Mealy Machine

Output: Moore Machine

Step 1 Calculate the number of different outputs for each state (Q_i) that are available in the state table of the Mealy machine.

Step 2 If all the outputs of Q_i are same, copy state Q_i. If it has **n** distinct outputs, break Q_i into **n** states as Q_{in} where **n** = 0, 1, 2.......

Step 3 If the output of the initial state is 1, insert a new initial state at the beginning which gives 0 output.

Example

Let us consider the following Mealy Machine:

Present State	Next State			
	a = 0		a = 1	
	Next State	Output	Next State	Output
→a	d	0	b	1
b	a	1	d	0
c	c	1	c	0
d	b	0	a	1

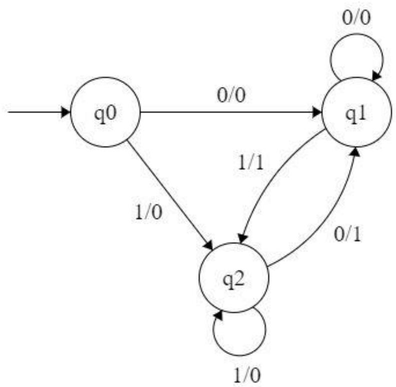
State table of a Mealy Machine

Here, states 'a' and 'd' give only 1 and 0 outputs respectively, so we retain states 'a' and 'd'. But states 'b' and 'c' produce different outputs (1 and 0). So, we divide **b** into $\mathbf{b_0}$, $\mathbf{b_1}$ and **c** into $\mathbf{c_0}$, $\mathbf{c_1}$.

Present State	Next State		Output
	a = 0	a = 1	
→ a	d	b_1	1
b_0	a	d	0
b_1	a	d	1
c_0	c_1	c_0	0
c_1	c_1	c_0	1
d	b_0	a	0

State table of equivalent Moore Machine

Example



Let us take the Moore machine of above Figure and its transition table is shown in Table 3.

Step 1. Construct an empty mealy machine using all states of moore machine as shown in Table 4.

	Input=0		Input=1	
Present State	**Next State**	**Output**	**Next State**	**Output**
q0				
q10				
q11				
q20				
q21				

Step 2: Next state for each state can also be directly found from moore machine transition Table as:

	Input=0		Input=1	
Present State	**Next State**	**Output**	**Next State**	**Output**
q0	q10		q20	
q10	q10		q21	
q11	q10		q21	
q20	q11		q20	
q21	q11		q20	

Table 5

Step 3: As we can see output corresponding to each input in moore machine transition table. Use this to fill the Output entries. e.g.; Output corresponding to q10, q11, q20 and q21 are 0, 1, 0 and 1 respectively.

	Input=0		Input=1	
Present State	**Next State**	**Output**	**Next State**	**Output**
q0	q10	0	q20	0
q10	q10	0	q21	1
q11	q10	0	q21	1
q20	q11	1	q20	0
q21	q11	1	q20	0

Table 6

Step 4: As we can see from table 6, q10 and q11 are similar to each other (same value of next state and Output for different Input). Similarly, q20 and q21 are also similar. So, q11 and q21 can be eliminated.

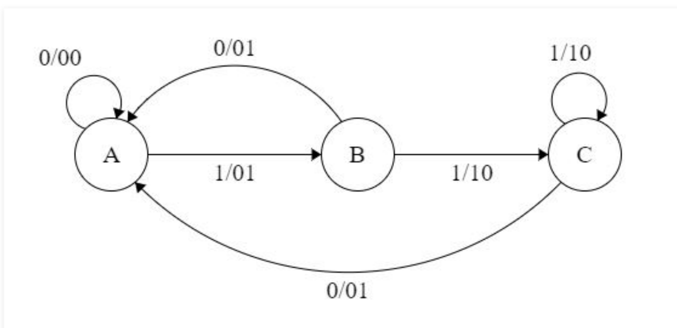
	Input=0		**Input=1**	
Present State	**Next State**	**Output**	**Next State**	**Output**
q0	q10	0	q20	0
q10	q10	0	q21	1
q20	q11	1	q20	0

Table 7

This is the same mealy machine shown in Table 1. So we have converted mealy to moore machine and converted back moore to mealy.

Note: Number of states in mealy machine can't be greater than number of states in moore machine.

Example: The Finite state machine described by the following state diagram with A as starting state, where an arc label is x / y and x stands for 1-bit input and y stands for 2- bit output?



Outputs the sum of the present and the previous bits of the input.

1. Outputs 01 whenever the input sequence contains 11.
2. Outputs 00 whenever the input sequence contains 10.
3. None of these.

Solution: Let us take different inputs and its output and check which option works:
Input: 01
Output: 00 01 (For 0, Output is 00 and state is A. Then, for 1, Output is 01 and state will be B)
Input: 11
Output: 01 10 (For 1, Output is 01 and state is B. Then, for 1, Output is 10 a nd state is C)

As we can see, it is giving the binary sum of present and previous bit. For fir st bit, previous bit is taken as 0.

Difference between Mealy machine and Moore machine

Mealy Machine – A mealy machine is defined as a machine in theory of computation whose output values are determined by both its current state and current inputs. In this machine atmost one transition is possible.

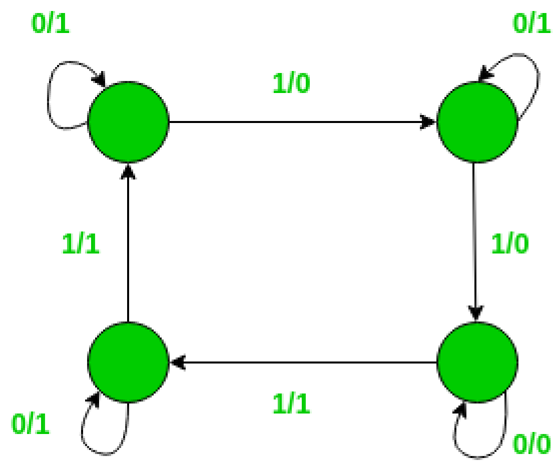


Figure - Mealy machine

Moore Machine – A Moore machine is defined as a machine in theory of computation whose output values are determined only by its current state.

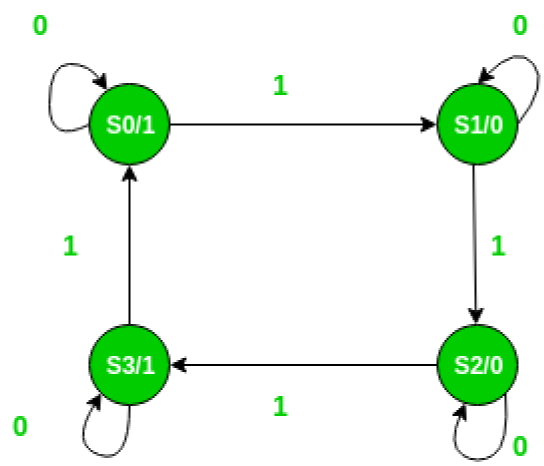


Figure - Moore machine

Moore Machine –

1. Output depends only upon present state.
2. If input changes, output does not change.
3. More number of states are required.
4. There is more hardware requirement.
5. They react slower to inputs(One clock cycle later)
6. Synchronous output and state generation.
7. Output is placed on states.
8. Easy to design.

Mealy Machine –

1. Output depends on present state as well as present input.
2. If input changes, output also changes.
3. Less number of states are required.
4. There is less hardware requirement.
5. They react faster to inputs.
6. Asynchronous output generation.
7. Output is placed on transitions.
8. It is difficult to design.

• **Note:** Finite state machines or finite automata are two names of the same thing.