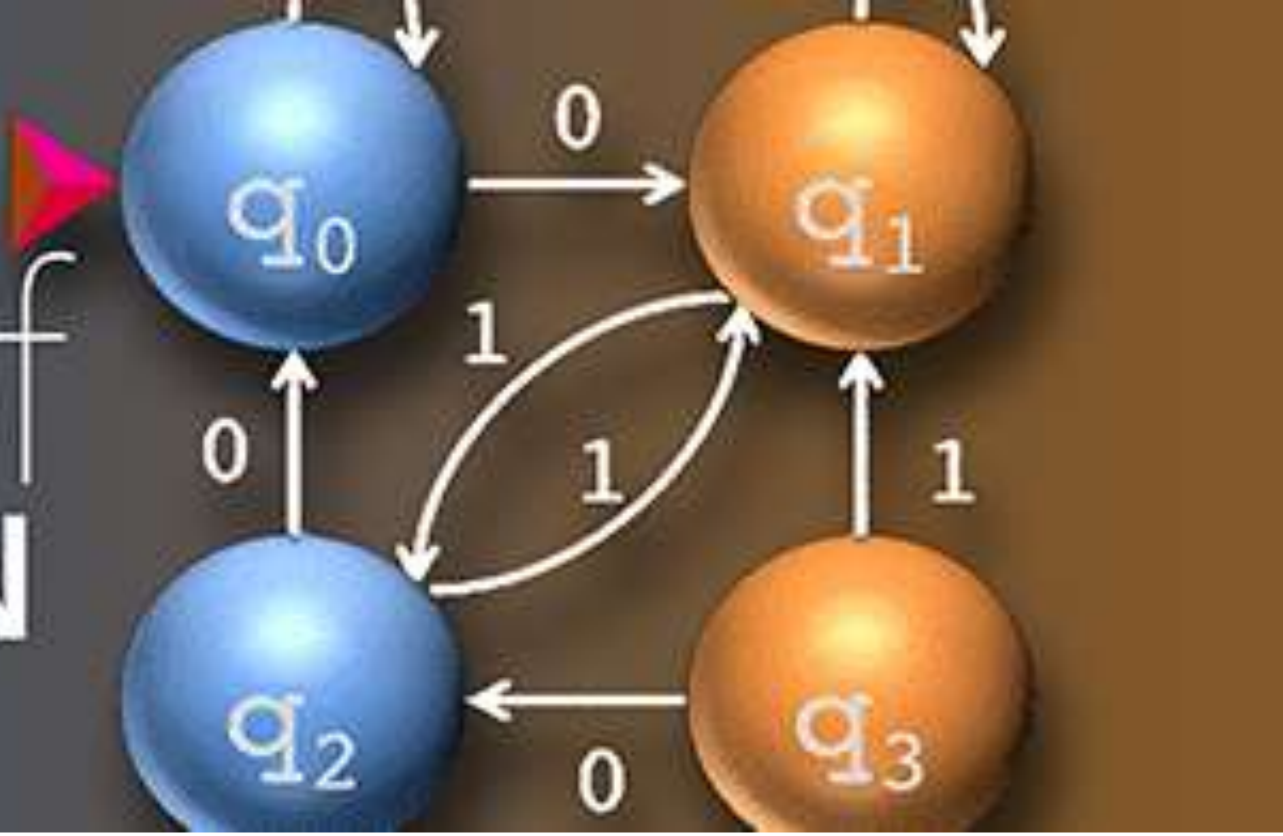
CSE 305

Theory of COMPUTATION



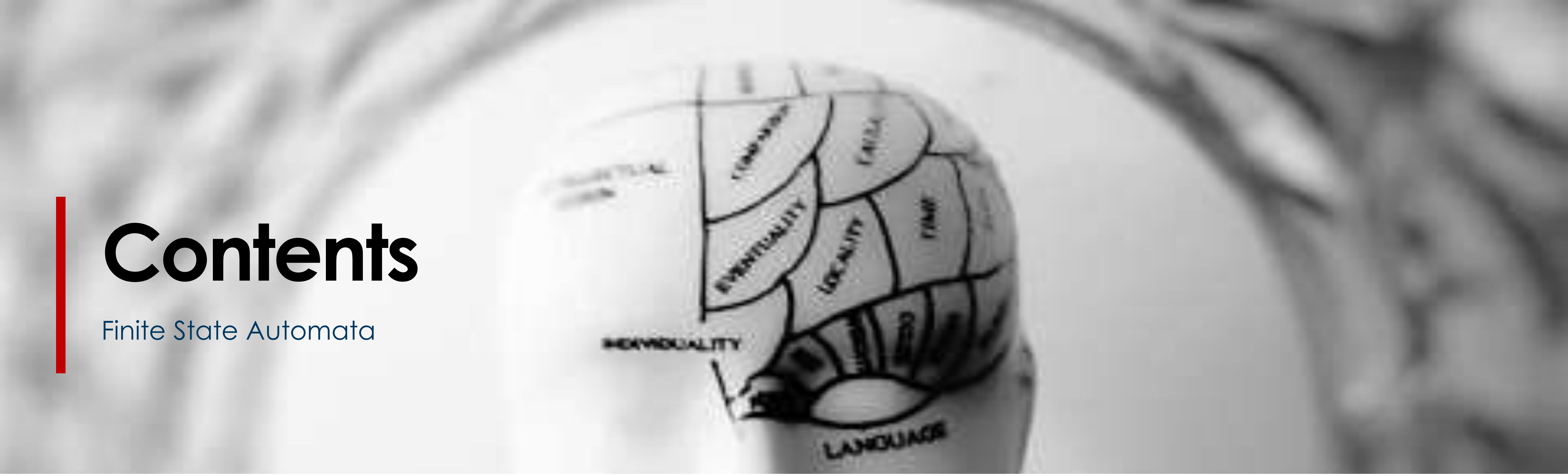
Lecture 19

Finite-State Automata (6)

Md. Mijanur Rahman, Prof. Dr.

Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.

www.mijanrahman.com



Contents

Finite State Automata

Two-way Finite Automaton

- In computer science, in particular in automata theory, a **two-way finite automaton** is a finite automaton that is allowed to re-read its input.

  - Two-way deterministic finite automaton (2DFA)
  - Two-way nondeterministic finite automaton (2NFA)

Two-way Deterministic Finite Automaton (2DFA)

- A **two-way deterministic finite automaton** (**2DFA**) is an abstract machine, **a generalized version of the deterministic finite automaton (DFA) which can revisit characters already processed.**

- As in a DFA, there are a finite number of states with transitions between them based on the current character, **but each transition is also labelled with a value indicating whether the machine will move its position in the input to the left, right, or stay at the same position.**

Two-way Deterministic Finite Automaton (2DFA)

Formal description:

- **Formally, a two-way deterministic finite automaton (2DFA) can be described by the following tuples:**

  $M = (Q, \Sigma, L, R, \delta, q_0, F)$

  Where:

  Q: a finite set of states

  Σ: a finite set of input symbols

  L: a left endmarker

  R: a right endmarker

  $\delta: Q \times (\Sigma \cup \{L, R\}) \rightarrow Q \times \{L, R\}$

  q0: the start state

  F: a set of final states

Two-way Nondeterministic Finite Automaton (2NFA)

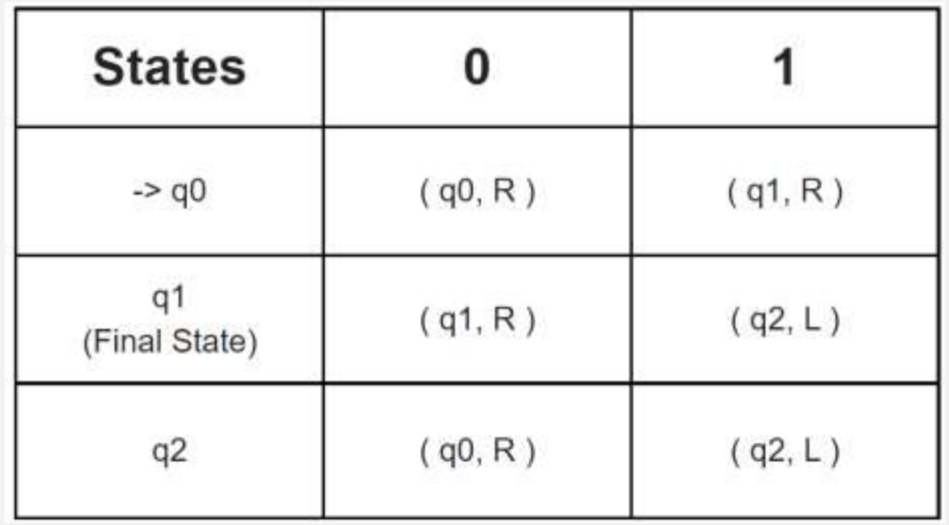
- A **two-way nondeterministic finite automaton** (2NFA) may have multiple transitions defined in the same configuration.
- Its transition function is defined as-

$$\delta : Q \times (\Sigma \cup \{L, R\}) \to 2^{Q \times \{\text{left}, \text{right}\}}.$$

- Like a standard one-way NFA, a 2NFA accepts a string if at least one of the possible computations is accepting. Like the 2DFAs, the 2NFAs also accept only regular languages.

Example: Accepting string by 2DFA

Example-1: Consider the following transition table of a 2DFA. Check the string "101001" whether it is accepted by 2DFA or not?



States	0	1
-> q0	(q0, R)	(q1, R)
q1 (Final State)	(q1, R)	(q2, L)
q2	(q0, R)	(q2, L)

Solution: Acceptability of the string using the 2DFA is illustrated below:

$$
\begin{aligned}
q_0 101001 &\vdash 1 q_1 01001 \\
&\vdash 10 q_1 1001 \\
&\vdash 1 q_2 01001 \\
&\vdash 10 q_0 1001 \\
&\vdash 101 q_1 001 \\
&\vdash 1010 q_1 01 \\
&\vdash 10100 q_1 1 \\
&\vdash 1010 q_2 01 \\
&\vdash 10100 q_0 1 \\
&\vdash 101001 q_1
\end{aligned}
$$

- Here we started from the initial state q0 and finally reached to q1 (final state). so the string "101001" is accepted by the 2DFA.

Finite Automata with Output

- **Finite automata have a limited capability of either accepting a string or rejecting a string.** Acceptance of a string was based on the reachability of a machine from starting state to final state.
- **Finite automata can also be used as an output device. A finite automata with output is similar to finite automata (FA) except that the additional capability of producing output.**
- In a formal way it is also known as **Finite State Machine (FSM) or Transducer**.

- **FSM = Transducer = Finite automata with output = Finite automata + Output Capability**

Finite Automata with Output

- **Formal description of finite automata with output machine M is defined by 6-tuples are as follows:**

  $M = (Q, \Sigma, \delta, \Delta, \lambda, q_0)$

  Where, each tuple have its specification and meaning, as follows:

  - Q: It represents the finite non-empty set of states.
  - Σ: It presents the finite non-empty set of the input alphabet.
  - δ : It represents the state transition function. The state transition function takes the current state from Q and an input alphabet from Σ and returns the new set of output alphabets and the next state. Therefore, it can be seen as a function which maps an ordered sequence of input alphabets into a corresponding sequence, or set, of output events.

    $Q \times \Sigma \rightarrow Q$ is the next-state function.
  - Δ: It presents the output alphabet.
  - λ : It represent the output function. Its mathematically denotes as: $Q \rightarrow \Delta$
  - q_0: It is the initial state (which may be fixed or variable depends on machine behavior).
- **If the FSM is given input letter a when in state q, it enters state δ(q, a). While in state q it produces the output letter λ(q).**

Finite Automata with Output

- **Finite automata with output machine have the following characteristic:**
  1. Finite automata with output machines do not have final state/states.
  2. Machine generates an output on every input. The value of the output is a function of current state and the current input.
  3. Finite automata with output machines are characterized by two behaviors:
     - **i.** **State transaction function (δ)** [State transition function (δ) is also known as STF]
     - **ii.** **Output function (λ)** [Output function is also known as machine function (MAF)]

       $\delta: \Sigma \times Q \to Q$

       $\lambda: \Sigma \times Q \to O$ [For Mealy machine]

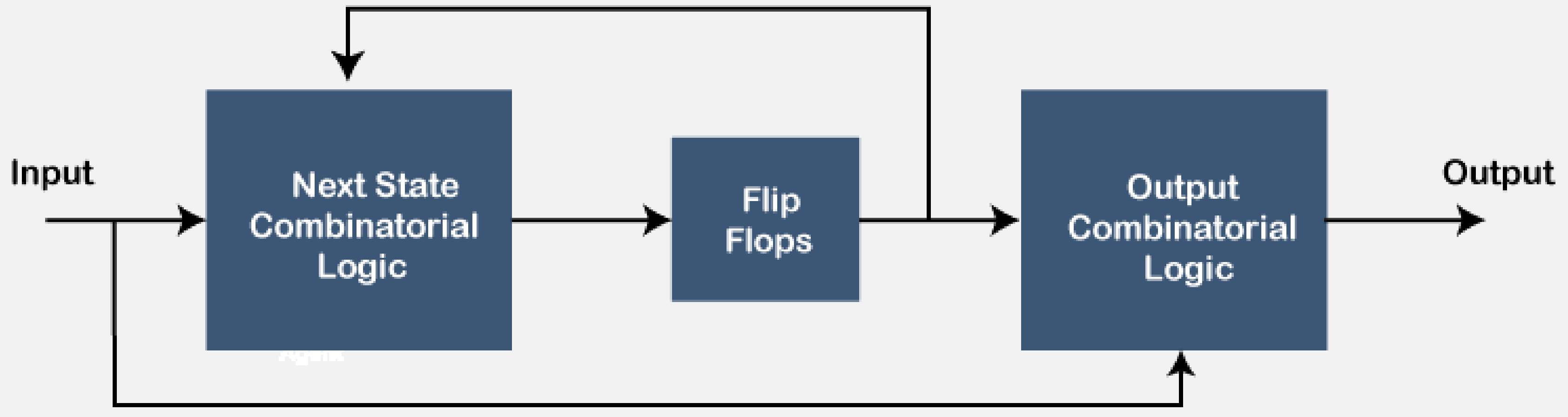
       $\lambda: Q \to O$ [For Moore machine]

Finite Automata with Output

- **There are two types of finite automata with output:**
  1. **Mealy machine:** Output is associated with transition.
     $\lambda: \Sigma \times Q \rightarrow O$
     Set of output alphabet O can be different from the set of input alphabet Σ.
  2. **Moore machine:** Output is associated with state.
     $\lambda: Q \rightarrow O$

Mealy Machine

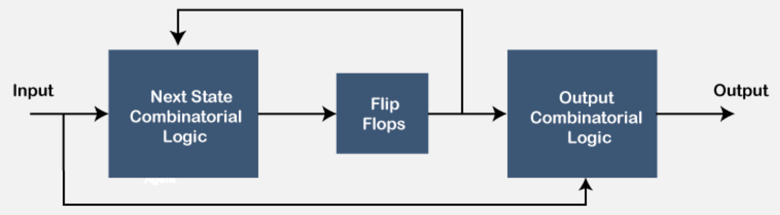
- In the Mealy machine, value of the output function depends on the current state **q(t)** and current input **i(t).** The architecture of mealy machine is given below:



(Mealy Machine)

- In this, if the machine has N number of states, then it will require N-flip-flops, where M is the smallest number such that $N<=2^{M.}$ In a mealy machine, if the input string of length **n**, then the output string will also be of length **n**.

Mealy Machine



(Mealy Machine)

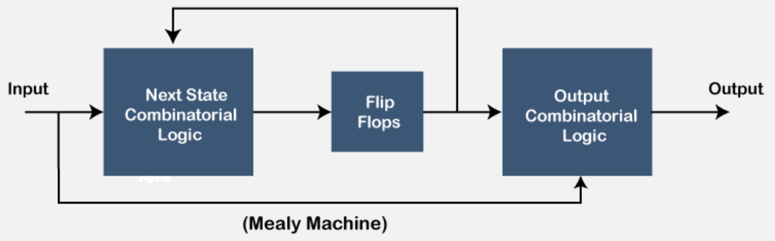
- **Formal Notations of Mealy Machine:** It can be described by a 6 tuple (Q, $\sum$, O, δ, Δ, q_0) as –

  $M = (Q, \Sigma, \delta, \Delta, \lambda, q_0)$

  - Q : It represents the finite non-empty set of states.
  - Σ: It presents the finite non-empty set of the input alphabet.
  - δ : It represents the state transition function; where-

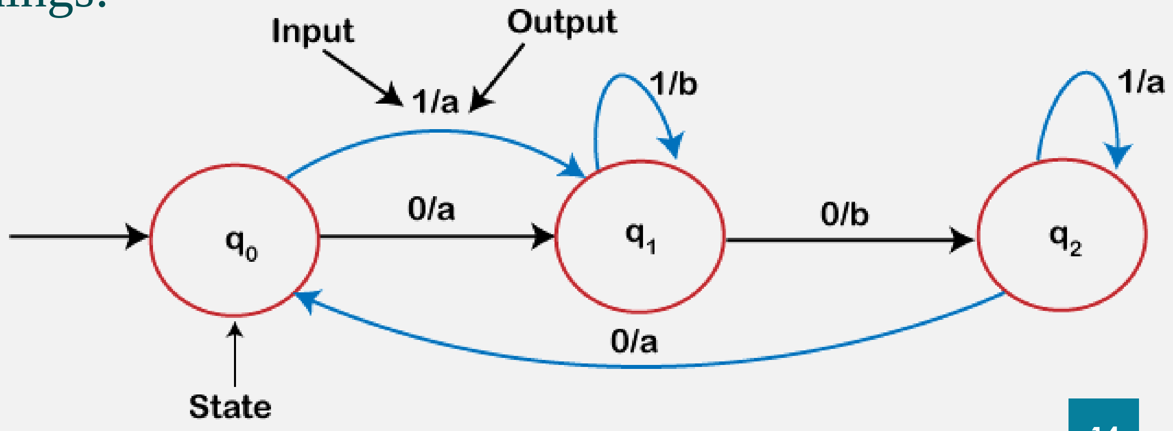
    $\delta : Q \times \Sigma \rightarrow Q$
  - Δ: It presents the output alphabet.
  - λ : It represent the output function. Its mathematically denotes as: $Q \times \Sigma \rightarrow \Delta$
  - q_0: It is the initial state.

Mealy Machine

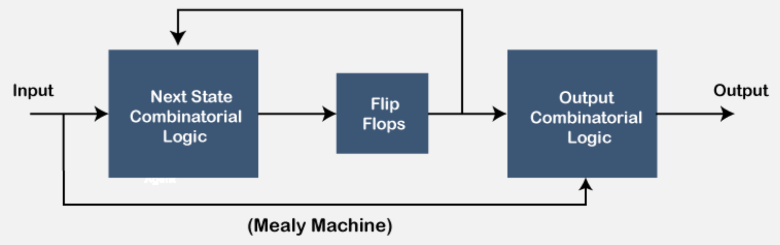


Transition diagram of a mealy machine:

- **To create a transition diagram for a given problem, apply the following steps:**
  1. First of all, determine the number of states needed for the given problem description.
  2. Represent each state with the help of a circle.
  3. From each state, draw an arrow causing event from the current state to the next state. If some particular input, the machine remains on the same state, then add this transition with the help of self-loop on the same state in the transition diagram.
  4. At last, write the output values along with the input values on the paths between the states.
- In the mealy machine, each arc is labeled with two things:
  1. First is the **input symbol**
  2. The second is the **output on the state.**



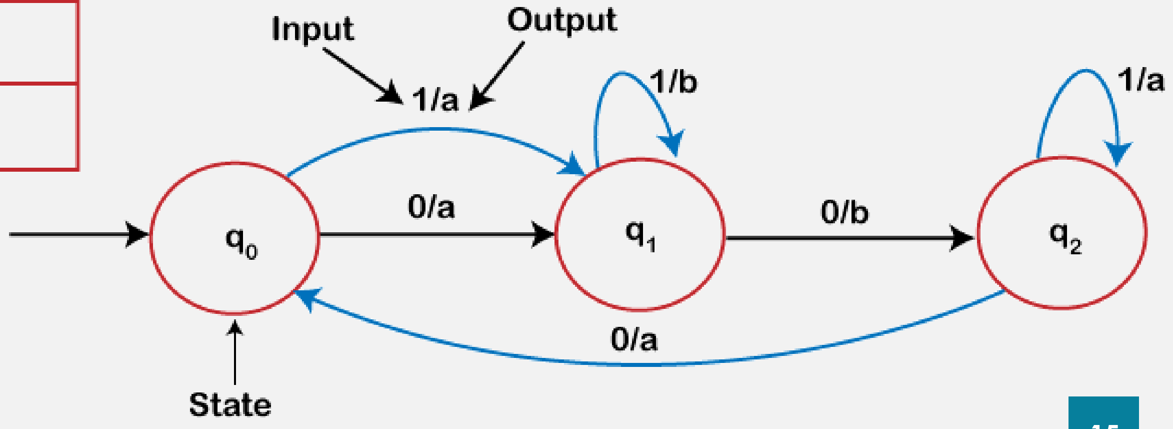
Mealy Machine



Transition table of a mealy machine:

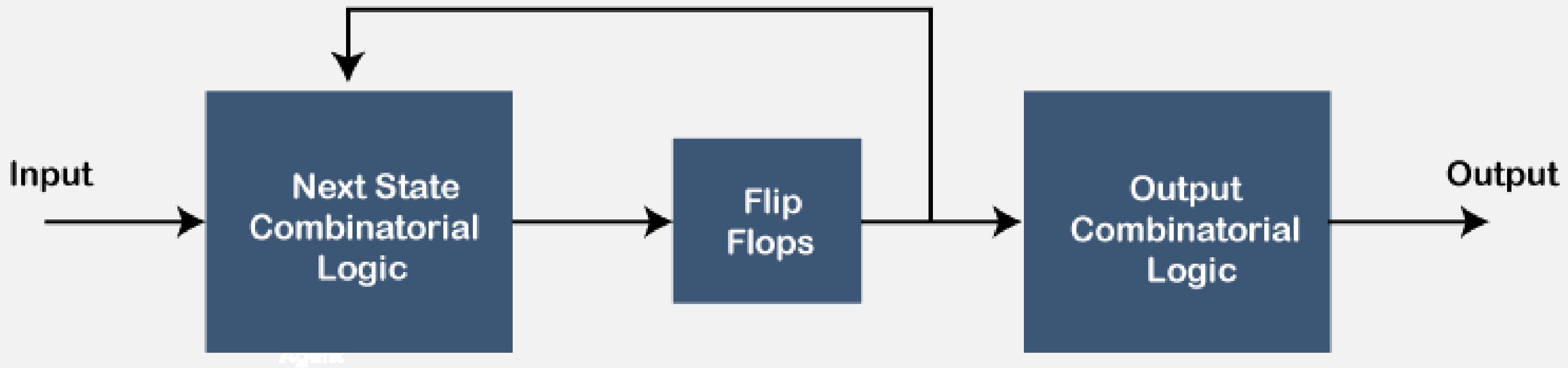
- **The output of the mealy machine depends on the current state and diagram.** The transition table of the mealy machine is given below:

Current State	Next State			
	Input = 0		Input = 1	
	State	Output	State	Output
→ q_0	q_1	a	q_1	a
q_1	q_2	b	q_1	b
q_2	q_0	a	q_0	a



Moore Machine

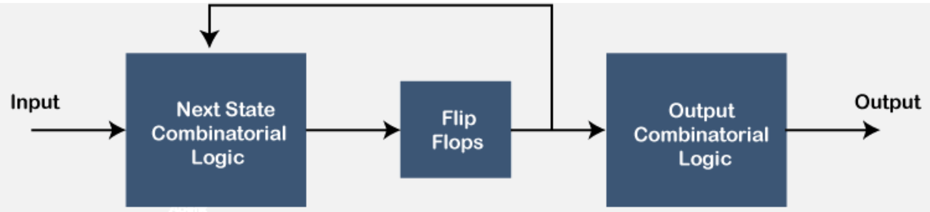
- **The output of the Moore machine depends only on the present state.** The general architecture of the Moore machine is:



(Moore Machine)

- In this, if the machine has N no of states, then it will require N-flip-flops, where M is the smallest number such that $N<=2^{M.}$ If the input string is of length **n**, then the output string will be of length **n+1**.

Moore Machine



(Moore Machine)

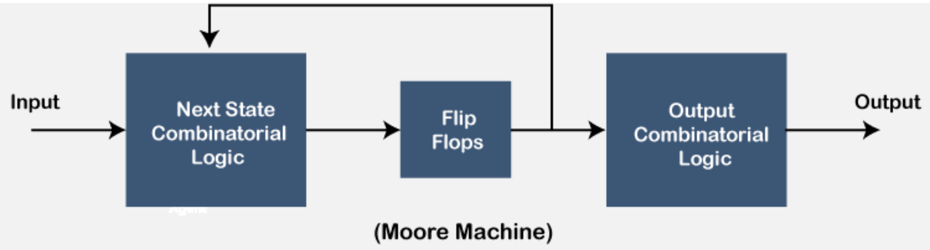
- **Formal Notations of Moore Machine:** It can be described by a 6 tuple (Q, $\sum$, O, δ, Δ, q_0) as –

  M = (Q, Σ, δ, Δ, λ, q_0)

  - Q : It represents the finite non-empty set of states.
  - Σ: It presents the finite non-empty set of the input alphabet.
  - δ : It represents the state transition function; where-

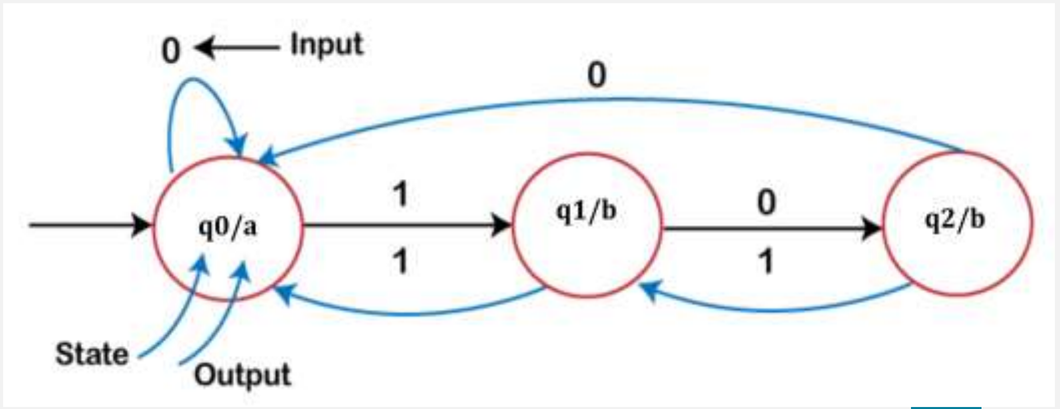
    $\delta : Q \times \Sigma \rightarrow Q$
  - Δ: It presents the output alphabet.
  - λ : It represent the output function. Its mathematically denotes as: $Q \rightarrow \Delta$
  - q_0: It is the initial state.

Moore Machine

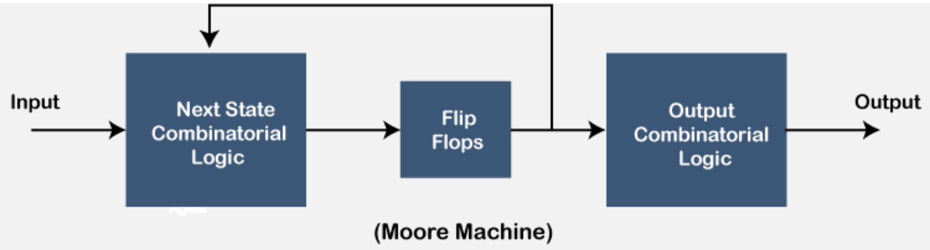


Transition diagram of Moore Machine:

- To create a transition diagram for a given problem, apply the following steps:
  1. First of all, determine the number of states needed from the given problem description.
  2. Represent each state with the help of a circle.
  3. From each state, draw an arrow causing event from the current state to the next state. If some particular input, the machine remains on the same state, then add this transition with the help of self-loop on the same state in the transition diagram.
  4. At last, write the value of the output in each state.
- In Moore machine, initial state is indicated by an arrow. Each state contains two things, first is the **name of the state**, and the second is the **output of the state**.



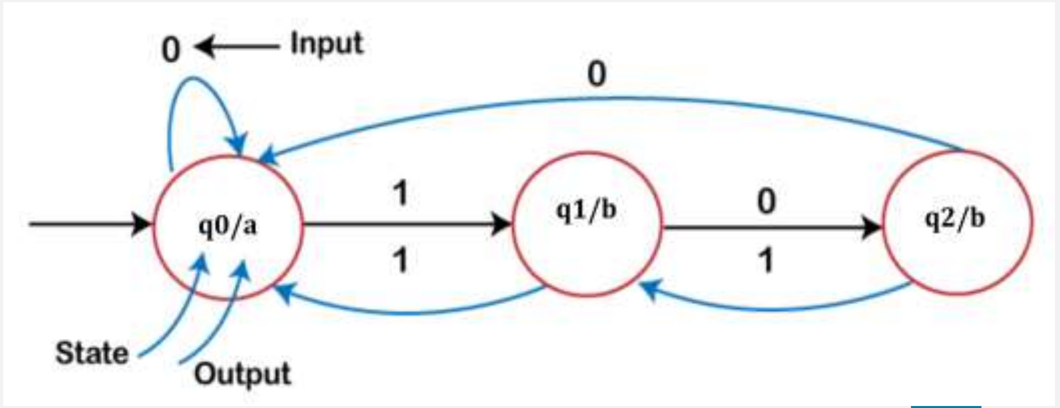
Moore Machine



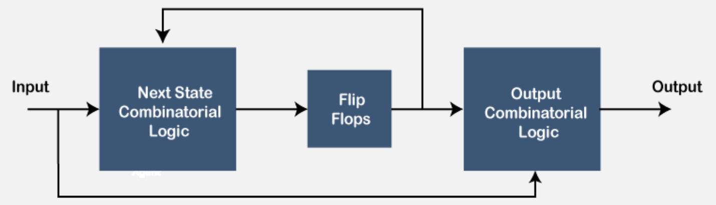
Transition table of Moore Machine:

- In this transition table, we have input alphabet **Σ = {0, 1}**and output alphabet **Δ = {0, 1}.** It takes input as **{0, 1}** and produces output in the form of **{a, b}.** Moore machine transition table also has the same input and output alphabet.

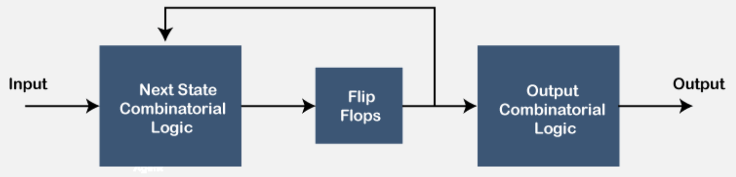
Current State	Next State		Output
	Input 0	Input 1	
→ q_0	q_0	q_1	a
q_1	q_2	q_0	b
q_2	q_0	q_1	b



Mealy Machine vs Moore Machine

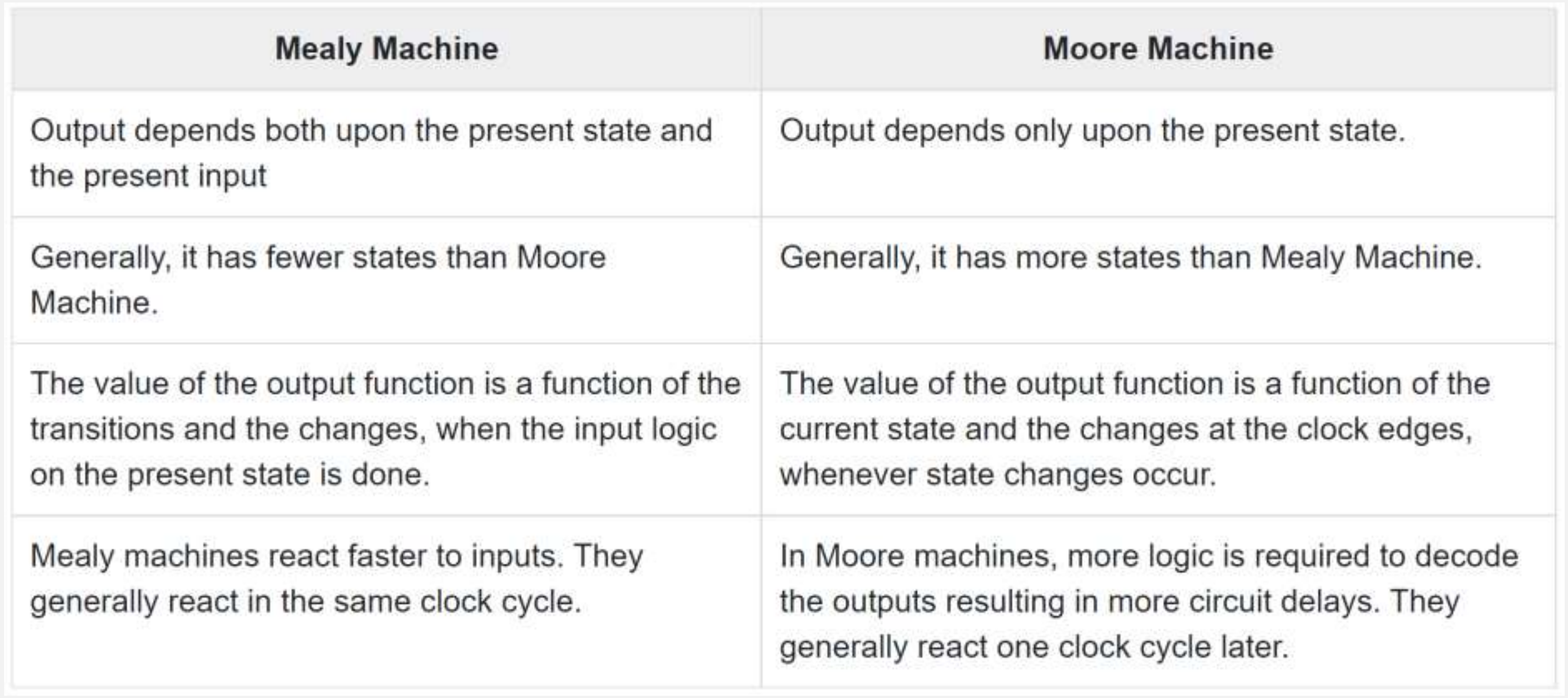


(Mealy Machine)



(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 the present state and the 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.
The value of the output function is a function of the transitions and the changes, when the input logic on the present state is done.	The value of the output function is a function of the current state and the changes at the clock edges, whenever state changes occur.
Mealy machines react faster to inputs. They generally react in the same clock cycle.	In Moore machines, more logic is required to decode the outputs resulting in more circuit delays. They generally react one clock cycle later.

Limitations of Finite Automata

- **Examples of Limitations of finite automata:**
  - FA can only count finite input.
  - There is no finite Automata that can find and recognize set of binary string of equal 0’s and 1’s (or equal number of a’s and b’s).
  - There is no finite Automata that can find and recognize the set of strings over ‘(‘ and ‘)’ that have “balanced” parentheses.
  - Input tape is read only and only memory it has is, state to state.
  - It can have only string pattern.
  - It can have only string pattern.

Applications of Finite Automata

- **We have several application based on finite automata and finite state machine.** Some are given below:
    - A finite automata is highly useful to design Lexical Analyzers of a compiler.
    - A finite automata is useful to design text editors.
    - A finite automata is highly useful to design spell checkers.
    - A finite automata is useful for recognizing the pattern using regular expressions.
    - A finite automata is useful to design sequential circuit design (Transducer), using Mealy and Moore Machines.

Assignment

- **Moore Machine and Mealy Machine**
- **Design a Moore Machine for 1’s complement of a binary number.**
- **Design a Mealy Machine for 2’s complement of a binary number.**

? THE END