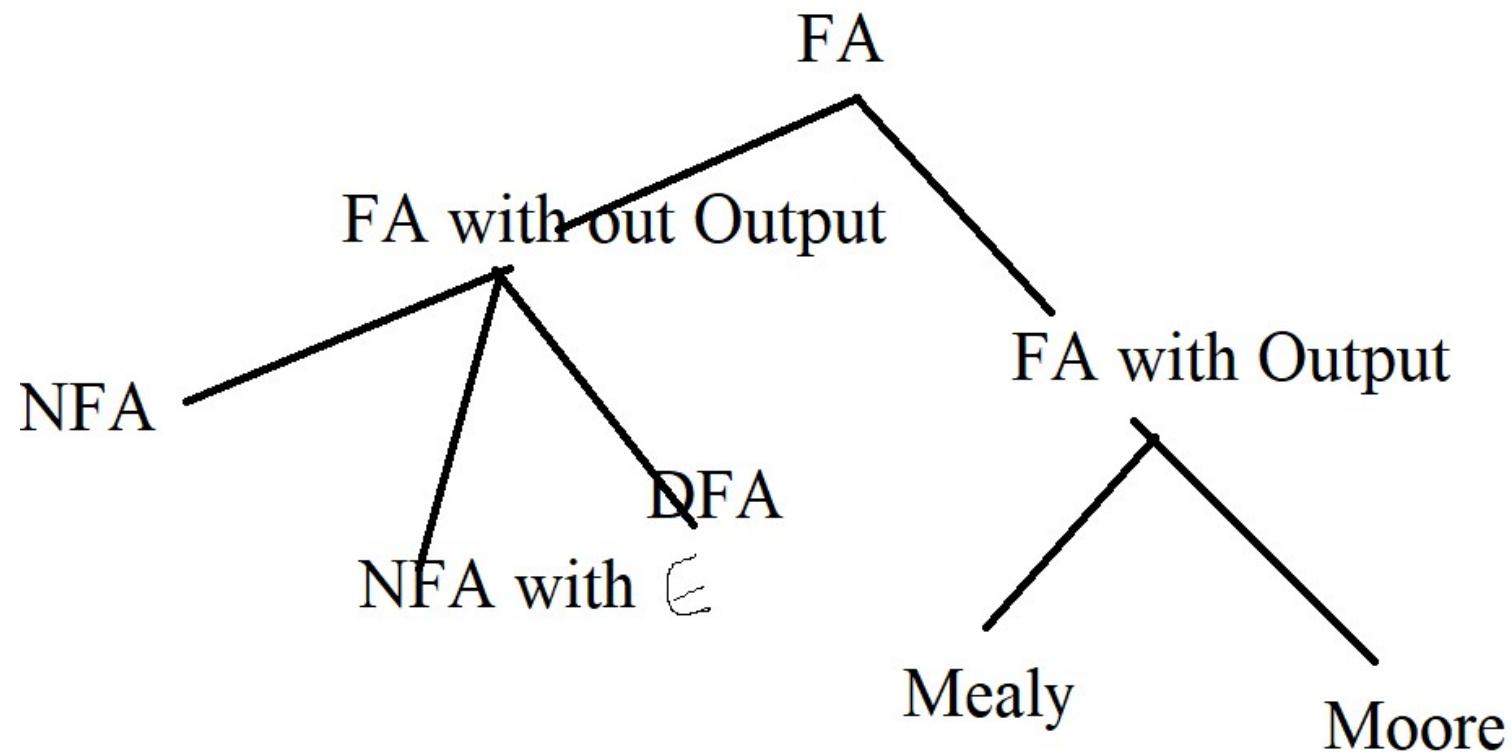


# Finite Automata



# Moore and Mealy Machine

## 2.8.1 FINITE AUTOMATA WITH OUTPUTS

The finite automata which we considered in the earlier sections have binary output, i.e. they accept the string or do not accept the string. This acceptability was decided on the basis of reachability of the final state by the initial state. Now, we remove this restriction and consider the model where the outputs can be chosen from some other alphabet. The value of the output function  $Z(t)$  in the most general case is a function of the present state  $q(t)$  and the present input  $x(t)$ , i.e.

$$Z(t) = \lambda(q(t), x(t))$$

Mealy

# Moore and Mealy Machine

where  $\lambda$  is called the output function. This generalised model is usually called *Mealy machine*. If the output function  $Z(t)$  depends only on the present state and is independent of the current input, the output function may be written as

$$Z(t) = \lambda(q(t))$$

Moore

This restricted model is called *Moore machine*. It is more convenient to use Moore machine in automata theory. We now give the most general definitions of these machines.

# Moore and Mealy Machine

**Definition 2.8** The Moore machine is a six-tuple  $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ , where

- (i)  $Q$  is a finite set of states;
- (ii)  $\Sigma$  is the input alphabet;
- (iii)  $\Delta$  is the output alphabet;
- (iv)  $\delta$  is the transition function  $\Sigma \times Q$  into  $Q$ ;
- (v)  $\lambda$  is the output function mapping  $Q$  into  $\Delta$ ; and
- (vi)  $q_0$  is the initial state.

**Definition 2.9** A Mealy machine is a six-tuple  $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$ , where all the symbols except  $\lambda$  have the same meaning as in the Moore machine.  $\lambda$  is the output function mapping  $\Sigma \times Q$  into  $\Delta$ .

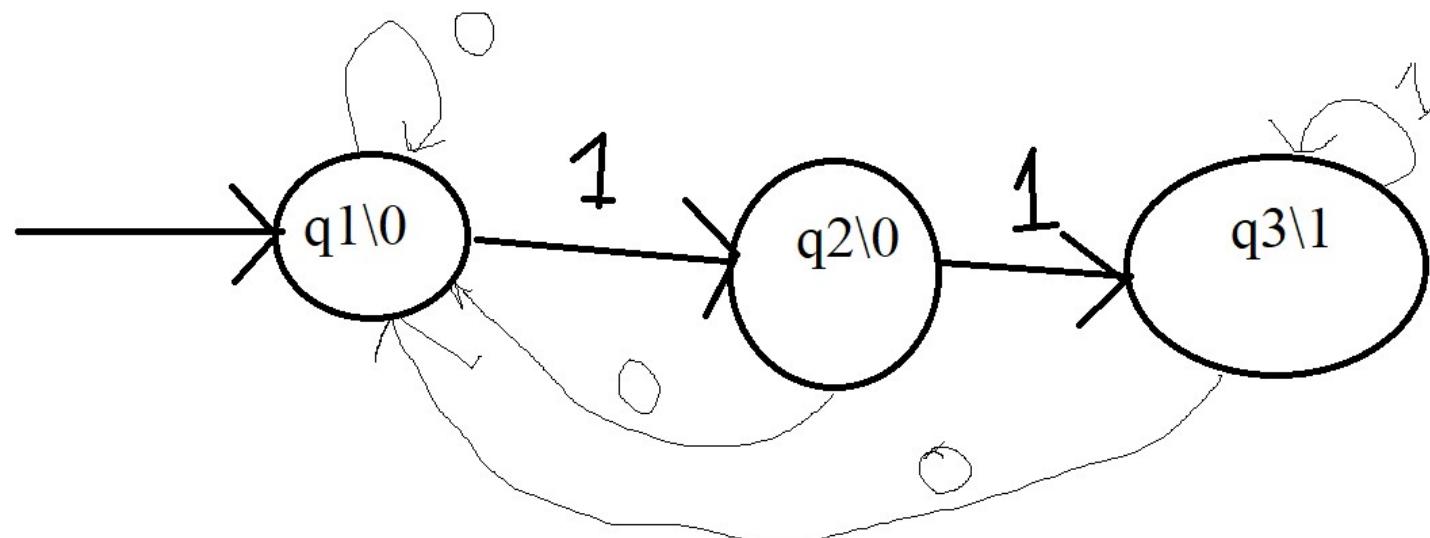
# Moore machine

**AMPLE 2.11** Consider the Moore machine described by the transition table in Table 2.16. Construct the corresponding Mealy machine.

Table 2.16 Moore Machine of Example 2.11

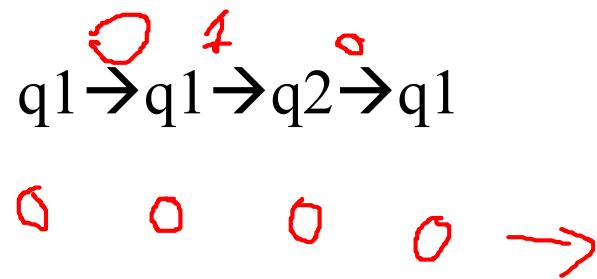
Present state	Next state		Output
	$a = 0$	$a = 1$	
$\rightarrow q_1$	$q_1$	$q_2$	0
$q_2$	$q_1$	$q_3$	0
$q_3$	$q_1$	$q_3$	1

Transition Diagram of Moore machine of example 2.11



Process the string ‘010’ using given Moore machine.

Length of the input string is 3



Length of the output string is  
 $3+1=4$

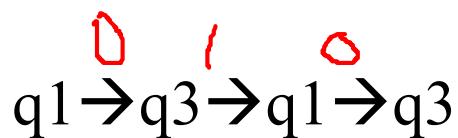
In Moore Machine, If the input string is of length ‘n’ the output string will also be of length ‘ $n+1$ ’.

Table 2.10 Mealy Machine of Example 2.9

Present state	Next state			
	input $a = 0$		input $a = 1$	
	state	output	state	output
$\rightarrow q_1$	$q_3$	0	$q_2$	0
$q_2$	$q_1$	1	$q_4$	0
$q_3$	$q_2$	1	$q_1$	1
$q_4$	$q_4$	1	$q_3$	0

Process the string '010' using given Mealy machine.

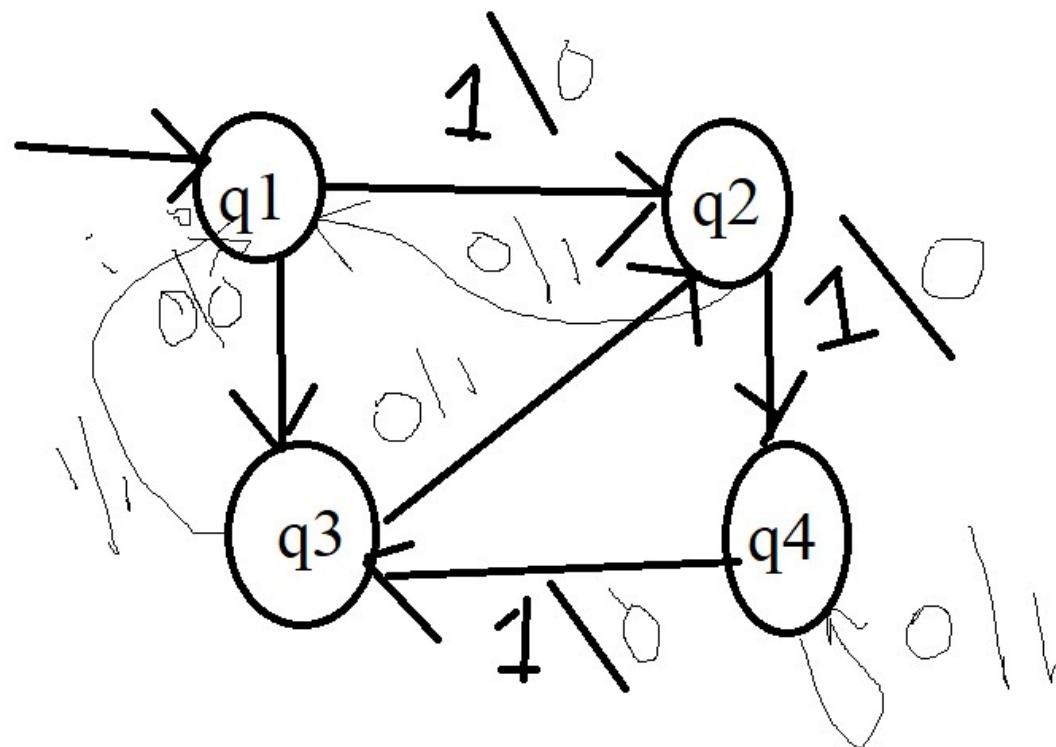
Length of the input string is 3



Length of the output string is 3

In Mealy Machine, If the input string is of length 'n' the output string will also be of length 'n' .

# Transition Diagram of Mealy Machine of example 2.10



# Difference between Moore & Mealy Machine

Moore Machine	Mealy Machine
<ul style="list-style-type: none"><li>• Output depends on present state only.</li></ul>	<ul style="list-style-type: none"><li>• Output depends on present state as well as on present input.</li></ul>
<ul style="list-style-type: none"><li>• Output is placed or attached on states</li></ul>	<ul style="list-style-type: none"><li>• Output is placed or attached with the transitions</li></ul>
<ul style="list-style-type: none"><li>• In Moore Machine, if the input string is of length <math>n</math>, the output string is of length <math>n+1</math>.</li></ul>	<ul style="list-style-type: none"><li>• In Mealy Machine, if the input string is of length <math>n</math>, the output string is of length <math>n</math>.</li></ul>
<ul style="list-style-type: none"><li>• There is less hardware requirement for circuit implementation.</li><li>• It is easy to design.</li></ul>	<ul style="list-style-type: none"><li>• There is more hardware requirement for circuit implementation.</li><li>• It is difficult to design.</li></ul>

# Conversion of Moore machine to Mealy Machine

**AMPLE 2.11** Consider the Moore machine described by the transition table in Table 2.16. Construct the corresponding Mealy machine.

**Table 2.16** Moore Machine of Example 2.11

Present state	Next state		Output
	$a = 0$	$a = 1$	
$\rightarrow q_1$	$q_1$	$q_2$	0
$q_2$	$q_1$	$q_3$	0
$q_3$	$q_1$	$q_3$	1

States and Output associated with State  
 $q_1-0$ ,  $q_2-0$ ,  $q_3-1$

## Transition Table of Mealy Machine

Present	Next State			
	a=0		a=1	
State	state	output	State	Output
q1	q1	0	q2	0
q2	q1	0	q3	1
q3	q1	0	q3	1

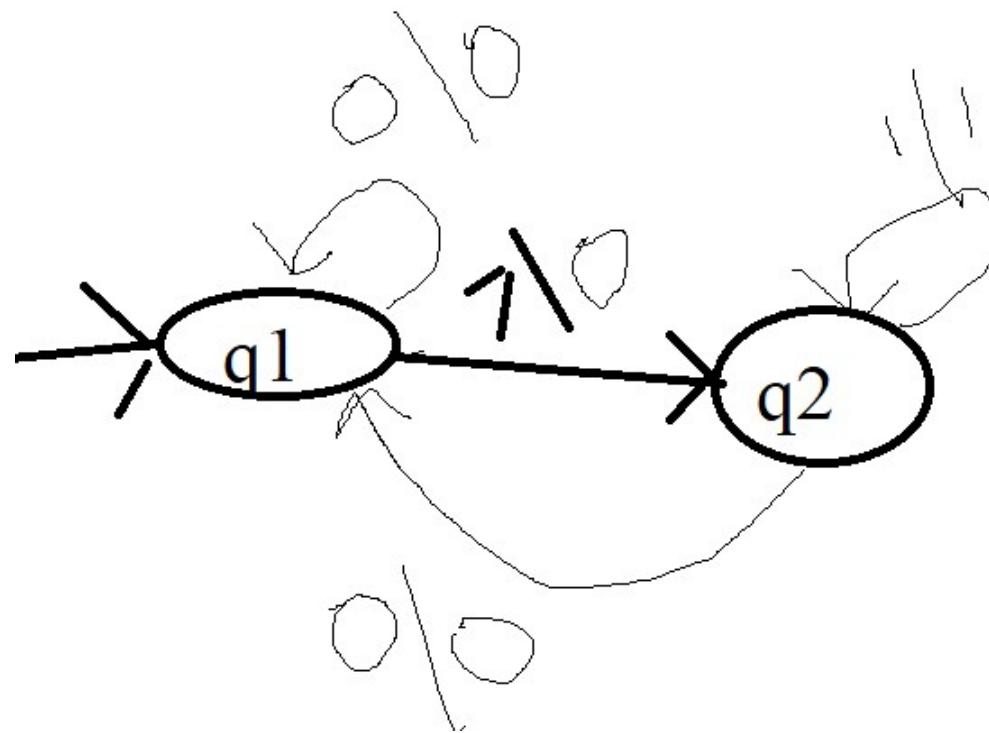
Here states q2 and q3 can be merged into a single state.  
 We will merge these 2 states into single state i.e q2.

Present	Next State			
	a=0		a=1	
State	state	output	State	Output
q1	q1	0	q2	0
q2	q1	0	q3	1
q3	q1	0	q3	1

## Final Transition Table of Mealy Machine

Present	Next State			
	a=0		a=1	
State	state	output	State	Output
q1	q1	0	q2	0
q2	q1	0	q2	1

Transition diagram of Mealy Machine



# HW: Conversion of Moore machine to Mealy Machine

Table 2.26 Moore Machine of Exercise 2.11

Present state	Next state		Output
	$a = 0$	$a = 1$	
$\rightarrow q_0$	$q_1$	$q_2$	1
$q_1$	$q_3$	$q_2$	0
$q_2$	$q_2$	$q_1$	1
$q_3$	$q_0$	$q_3$	1

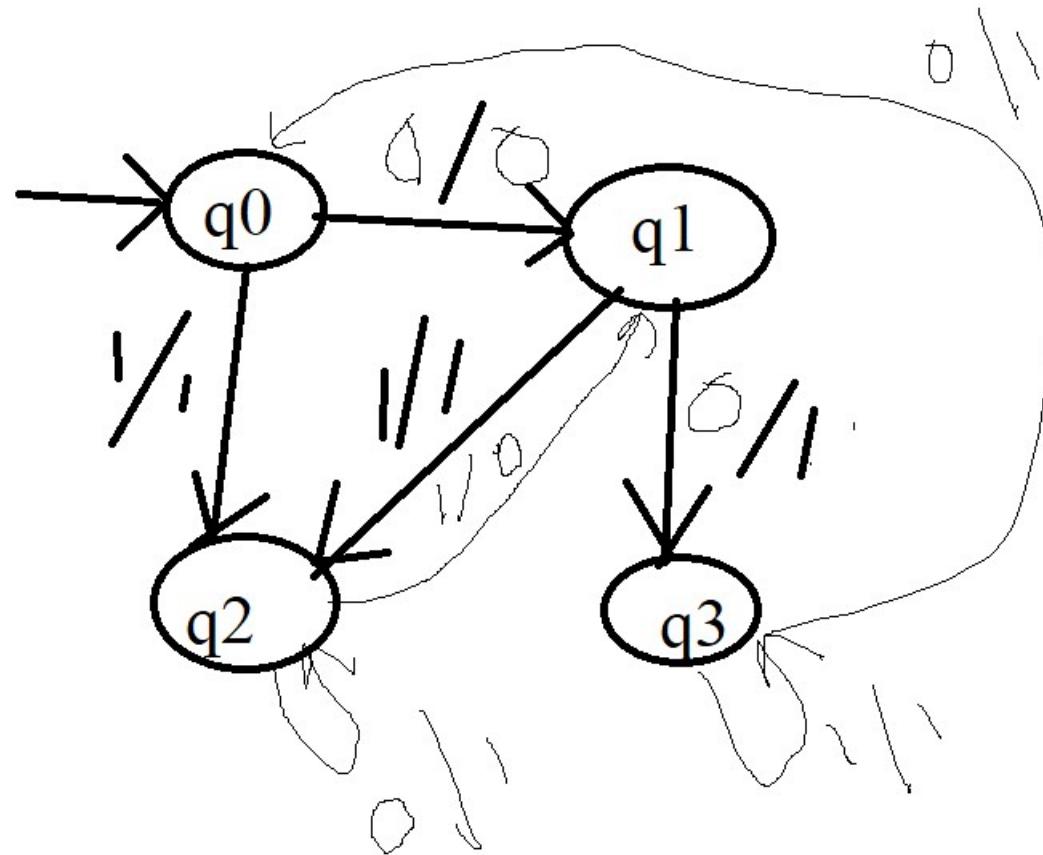
Construct a Moore

States and their corresponding output  
 $q_0-1$ ,  $q_1-0$ ,  $q_2-1$ ,  $q_3-1$

## Transition Table of Mealy Machine

Present State	Next State			
	a=0		a=1	
	state	output	State	Output
q0	q1	0	q2	1
q1	q3	1	q2	1
q2	q2	1	q1	0
q3	q0	1	q3	1

# Transition Diagram of Mealy Machine



# HW: Conversion of Moore machine to Mealy Machine

**Table 2.14** Moore Machine of Example 2.10

Present state	Next state		Output
	$a = 0$	$a = 1$	
$\rightarrow q_0$	$q_3$	$q_1$	0
$q_1$	$q_1$	$q_2$	1
$q_2$	$q_2$	$q_3$	0
$q_3$	$q_3$	$q_0$	0

# Conversion of Mealy machine to Moore Machine

Table 2.10 Mealy Machine of Example 2.9

Present state	Next state			
	input $a = 0$		input $a = 1$	
	state	output	state	output
$\rightarrow q_1$	$q_3$	0	$q_2$	0
$q_2$	$q_1$	1	$q_4$	0
$q_3$	$q_2$	1	$q_1$	1
$q_4$	$q_4$	1	$q_3$	0

Associate output with state

**q3-0, q1-1, q20-0,q21-1,q40-0,q41-1**

We have split  $q_2$  and  $q_4$  states into two different states i.e  $q_{20}$ ,  $q_{21}$  and  $q_{40}$ ,  $q_{41}$  respectively as output associated with  $q_2$  and  $q_4$  are 0 and 1

# Mealy Machine

Present State	a=0		a=1	
	state	output	state	output
q1	q3	0	q20	0
q20	q1	1	q40	0
q21	q1	1	q40	0
q3	q21	1	q1	1
q40	q41	1	q3	0
q41	q41	1	q3	0

## Moore Machine

Present state	Next state		Output
	a=0	a=1	
q1 →	q3	q20	1
q20	q1	q40	0
q21	q1	q40	1
q3	q21	q1	0
q40	q41	q3	0
q41	q41	q3	1

## Moore Machine

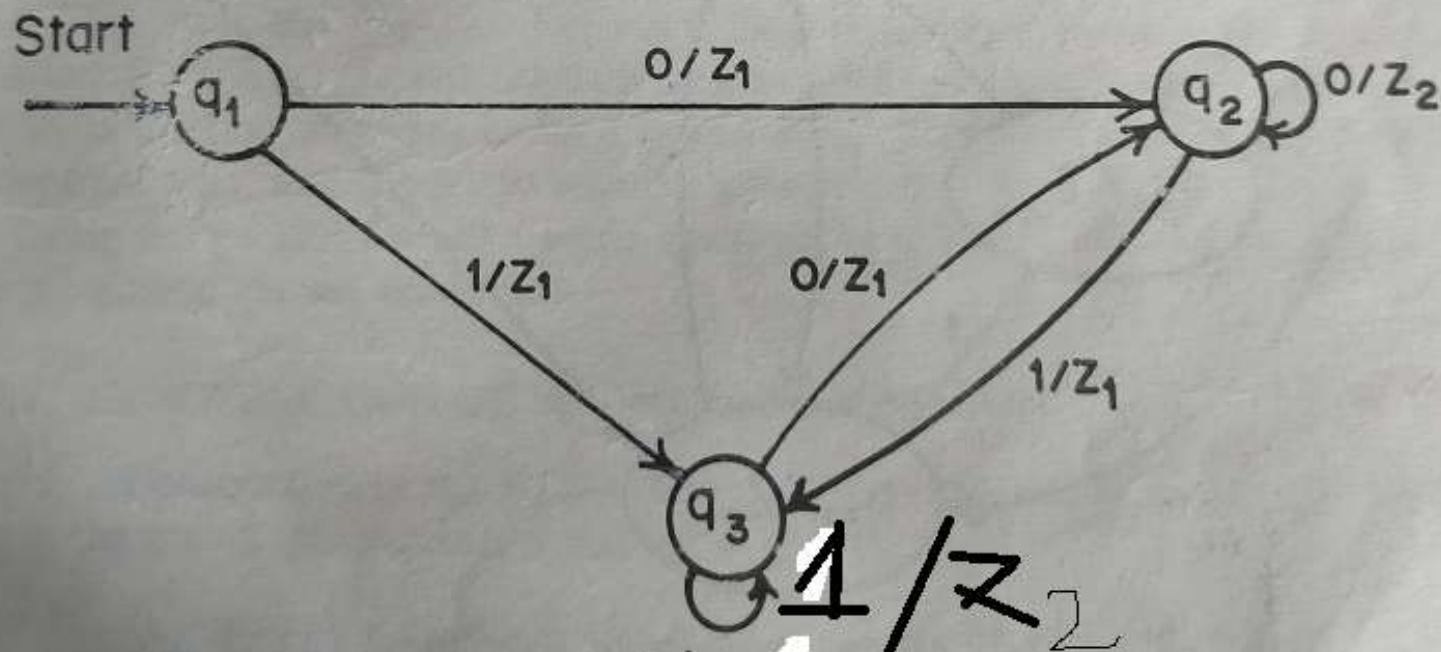


Present state	Next state		Output
	a=0	a=1	
q0	q3	q20	0
q1	q3	q20	1
q20	q1	q40	0
q21	q1	q40	1
q3	q21	q1	0
q40	q41	q3	0
q41	q41	q3	1

We want to construct an equivalent moore machine from the given Mealy m/c. As the given mealy m/c does not give any output for null string our converted Moore m/c should also act in the same way. So we have added new initial state i.e q0 with output 0 as the previous initial state was giving o/p 1 with null string.

# Hw: Convert the following Mealy Machine into Moore Machine

**EXAMPLE 2.12** Consider a Mealy machine represented by Fig. 2.10. Construct a Moore machine equivalent to this Mealy machine.

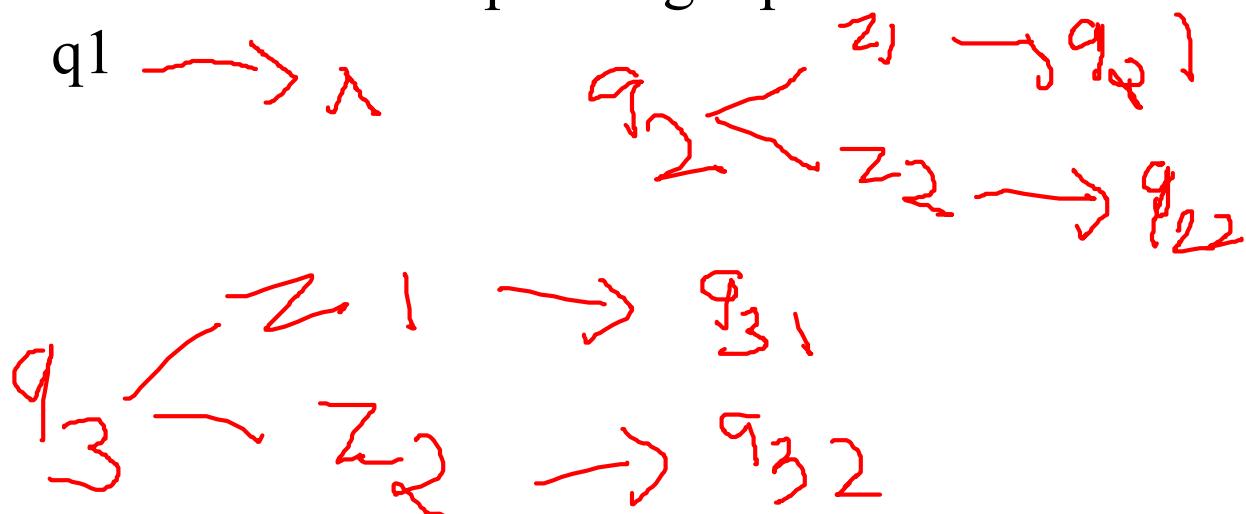


## Transition Table of Mealy Machine

Present	Next State			
	a=0		a=1	
State	state	output	State	Output
q1	q2	Z1	q3	Z1
q2	q2	Z2	q3	Z1
q3	q2	Z1	q3	Z2



States and corresponding o/p



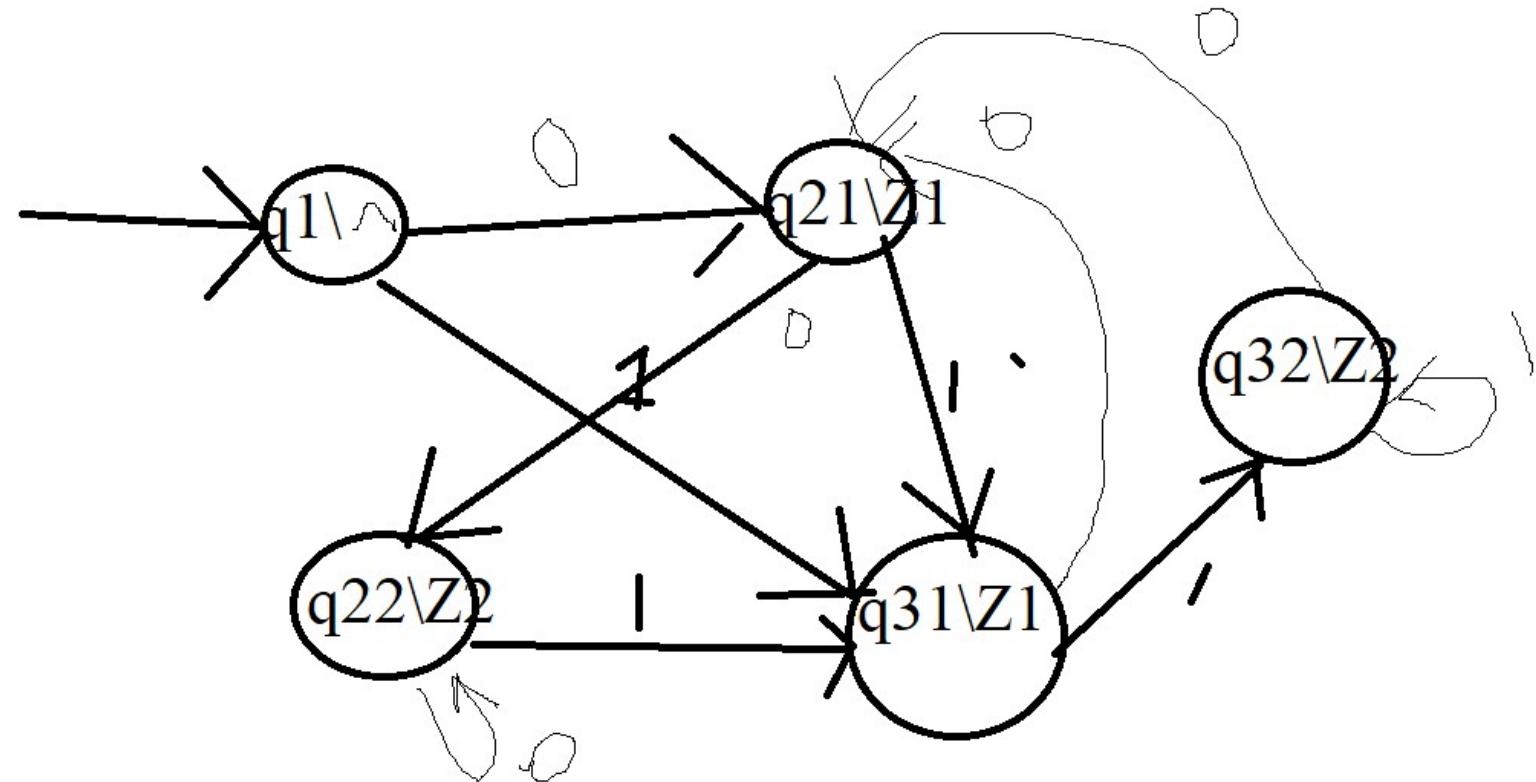
# Mealy Machine

Present State	a=0		a=1	
	state	output	state	output
q1	q21	Z1	q31	Z1
q21	q22	Z2	q31	Z1
q22	q22	Z2	q31	Z1
q31	q21	Z1	q32	Z2
q32	q21	Z1	q32	Z2

## Moore Machine

Present state	Next state		Output
	a=0	a=1	
q1 →	q21	q31	↑
q21	q22	q31	Z1
q22	q22	q31	Z2
q31	q21	q32	Z1
q32	q21	q32	Z2

# Transition Diagram of Moore Machine



- Construct a Mealy Machine which can output EVEN, ODD according as the total number of 1's encountered is even or odd.

The input symbols are 0 and 1.

Output: EVEN, ODD

INPUT: 0,1    input:1    ODD

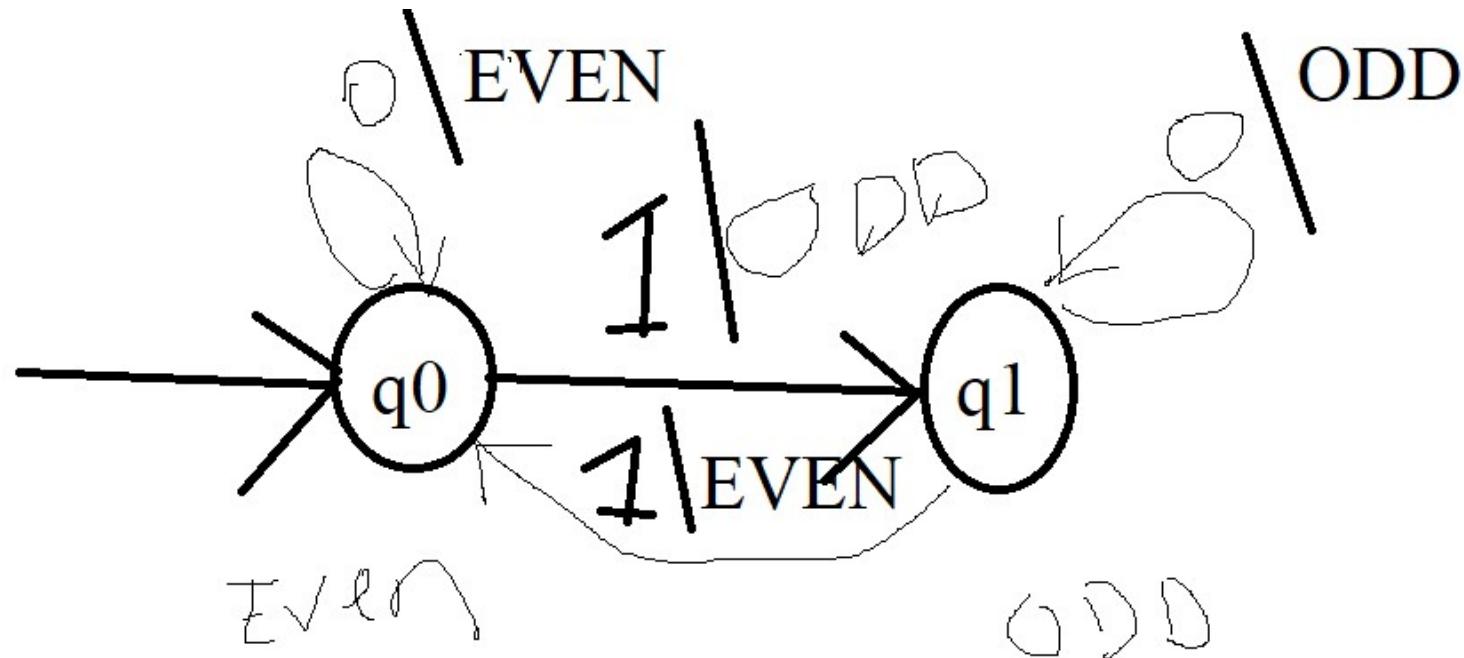
Input:11 EVEN

Input: 01 ODD

INPUT 10 ODD

INPUT; 00 EVEN

## Transition Diagram of Mealy Machine



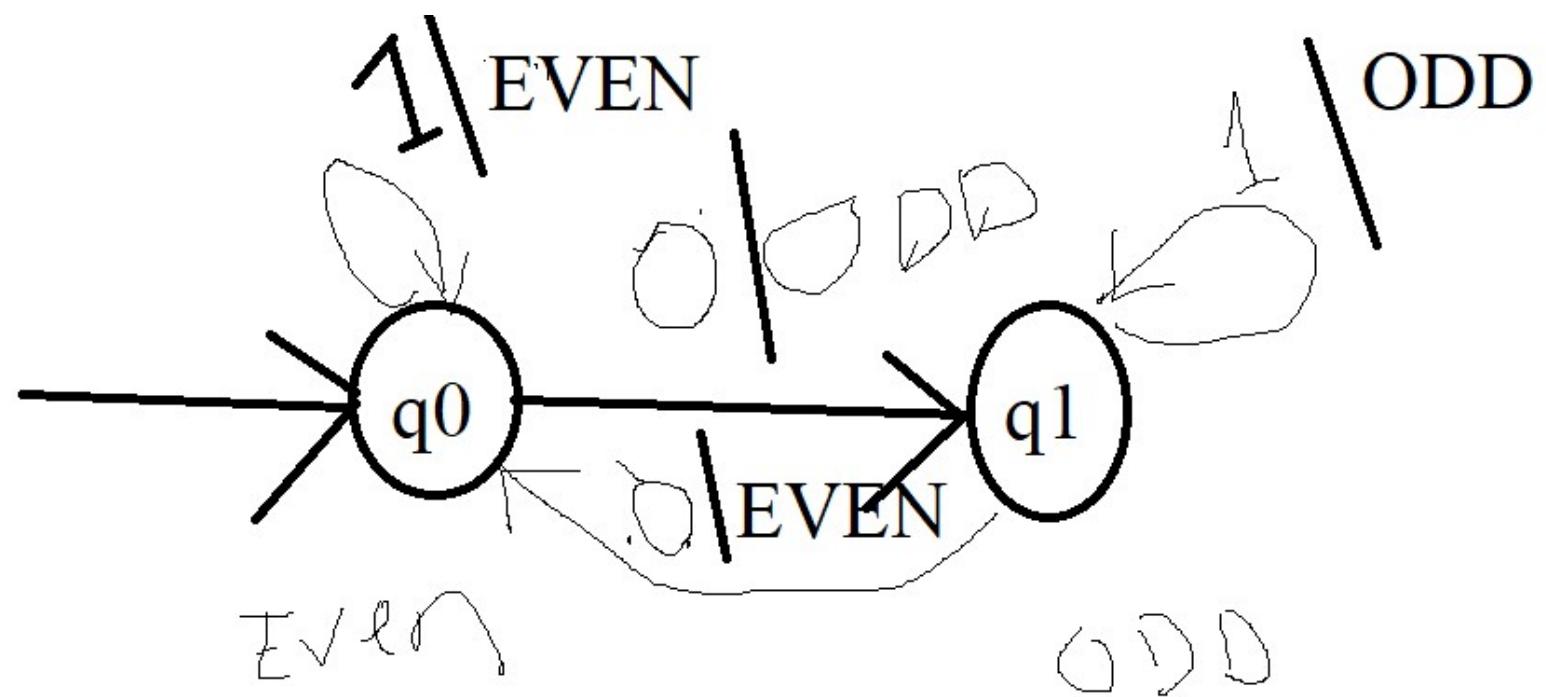
## Transition Table of Mealy M/c

	0		1	
	State	Output	State	Output
q0	q0	EVEN	q1	ODD
q1	q1	ODD	q0	EVEN

- Construct a Mealy Machine which can output EVEN, ODD according as the **total number of 0's** encountered is even or odd.

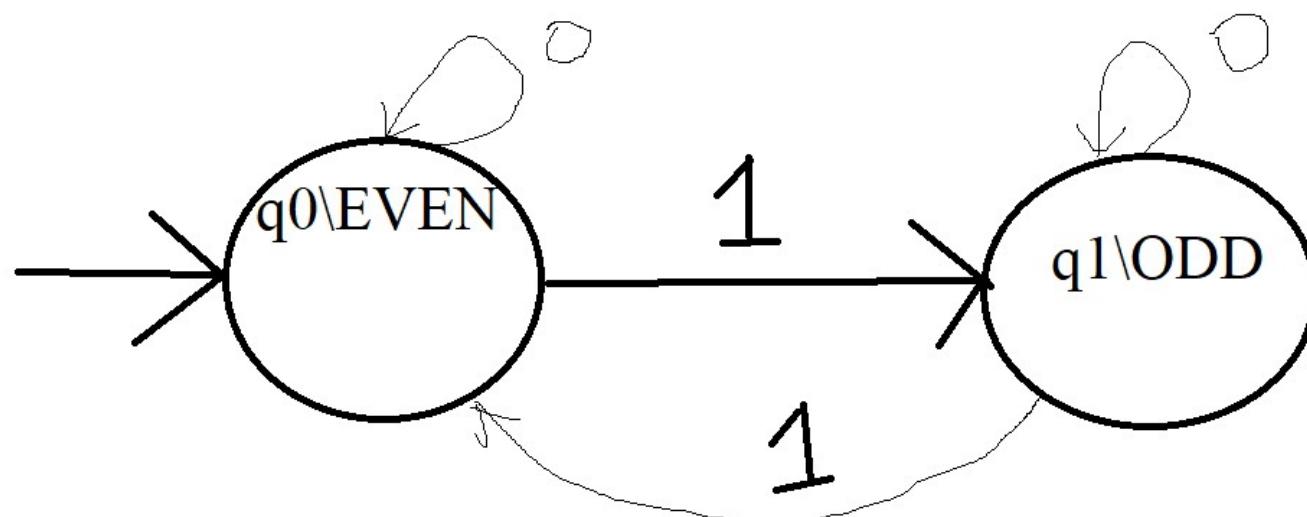
The input symbols are 0 and 1.

Input	Output
0	ODD
1	EVEN
01	ODD
10	ODD
11	EVEN
00	EVEN



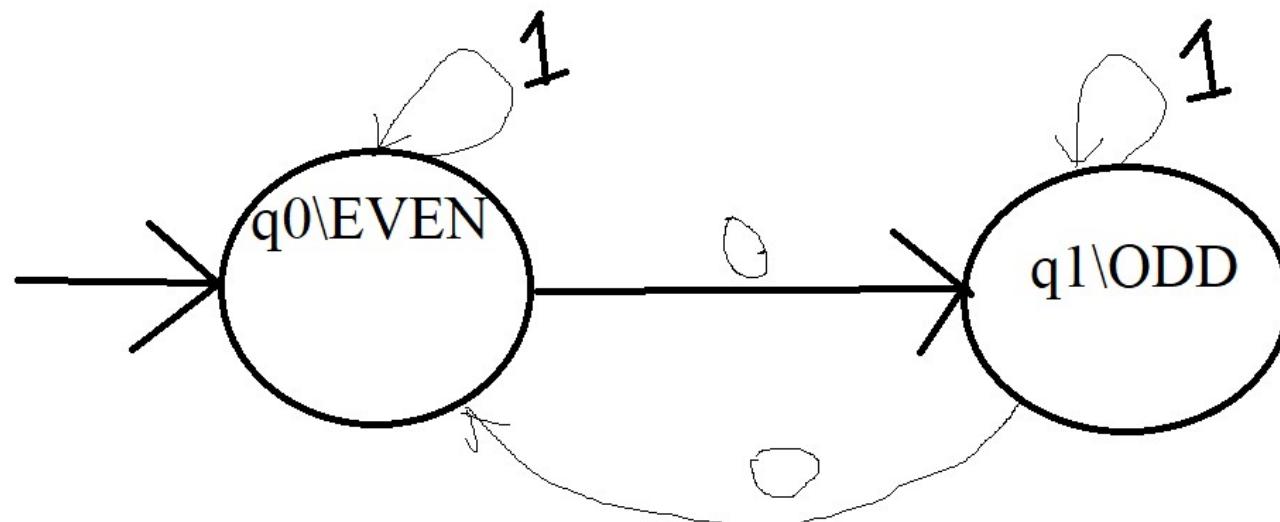
- Construct a Moore Machine which can output EVEN, ODD according as the total number of 1's encountered is even or odd.

The input symbols are 0 and 1.



- Construct a Moore Machine which can output EVEN, ODD according as the total number of 0's encountered is even or odd.

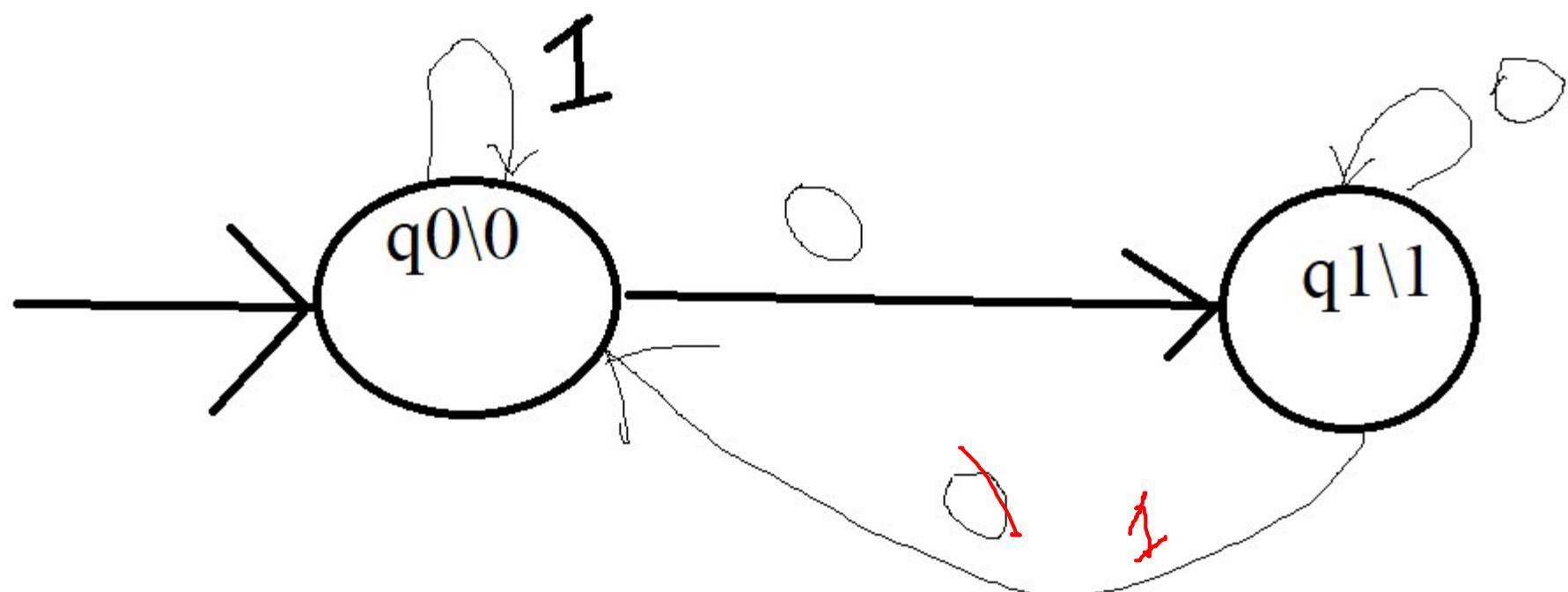
The input symbols are 0 and 1.



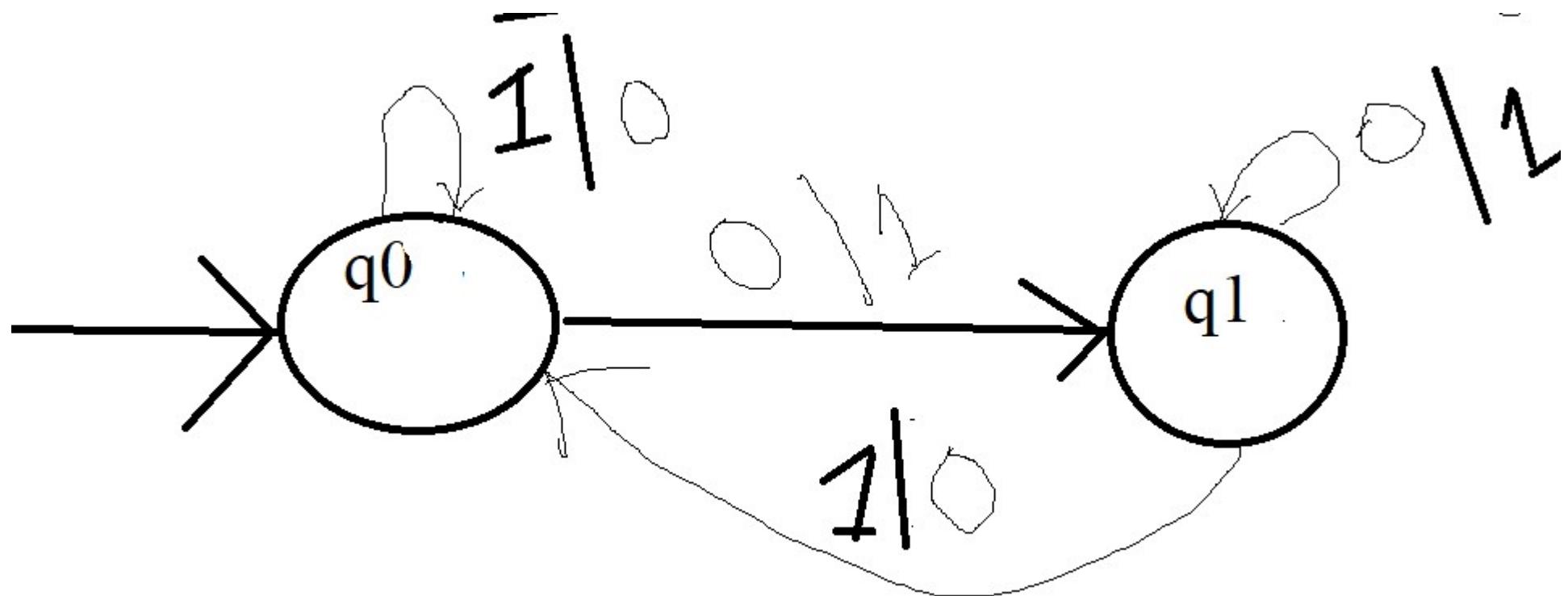
- Construct a Moore Machine for calculating 1's complement of a binary number.

The input symbols are 0 and 1.

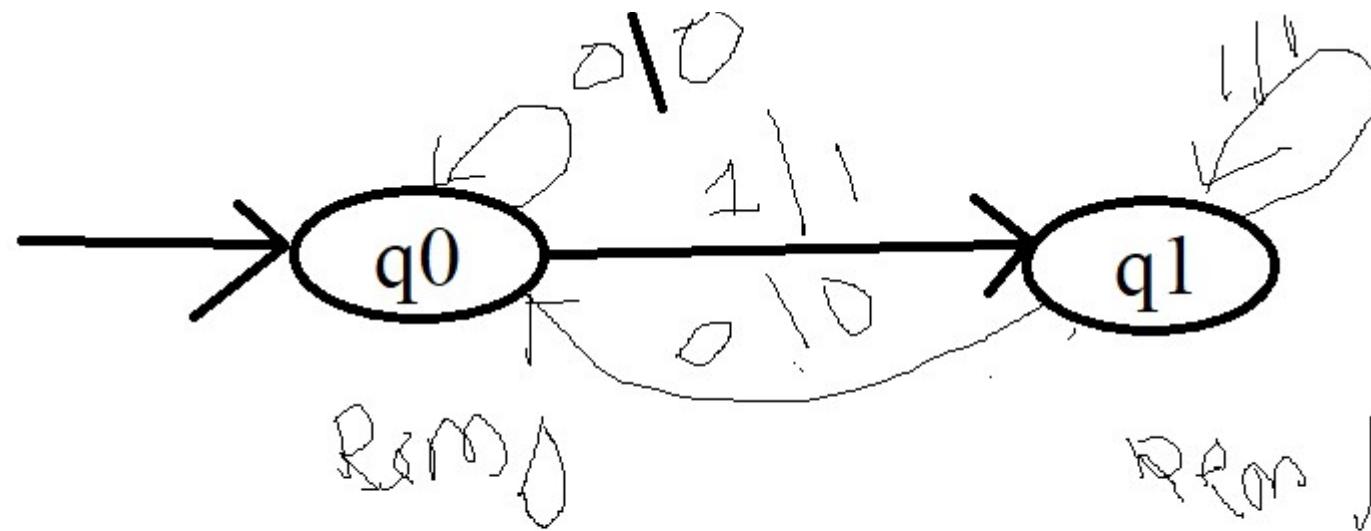
Input	output
1	0
0	1
10	01



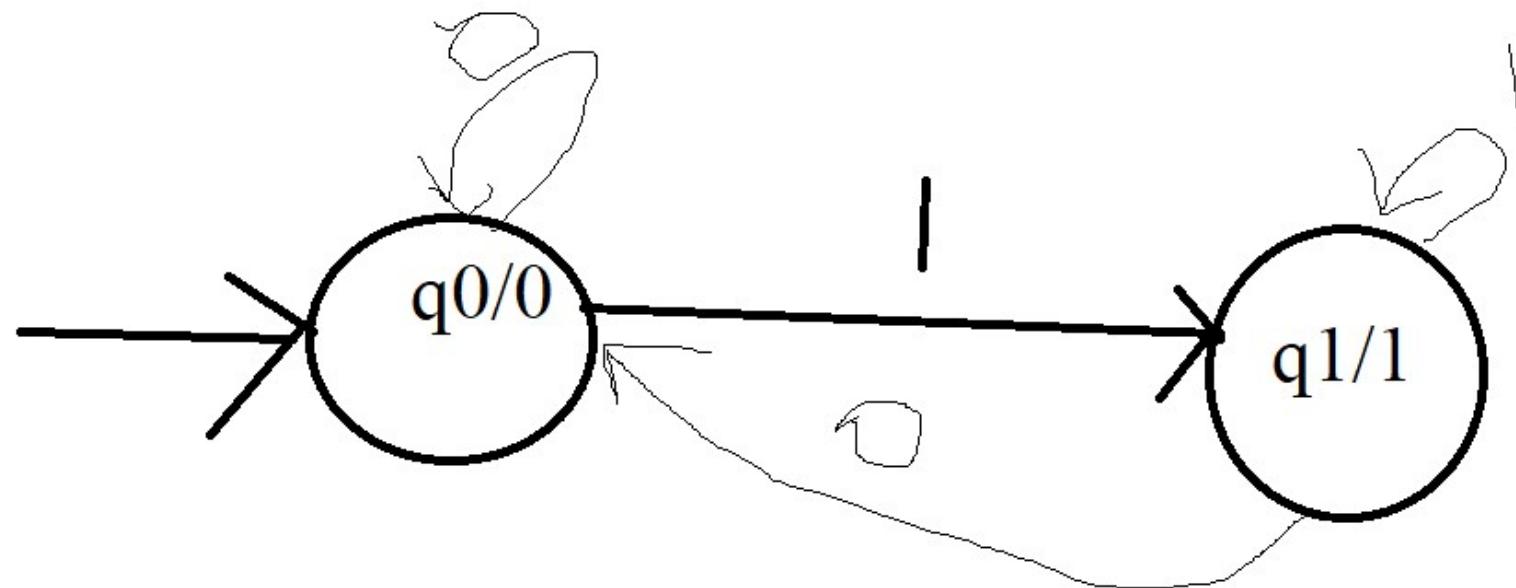
Construct a Mealy Machine for calculating 1's complement of a binary number.



- Construct a Mealy Machine for testing divisibility by 2 of a binary number
  - Input; 0 & 1                          Rem-0

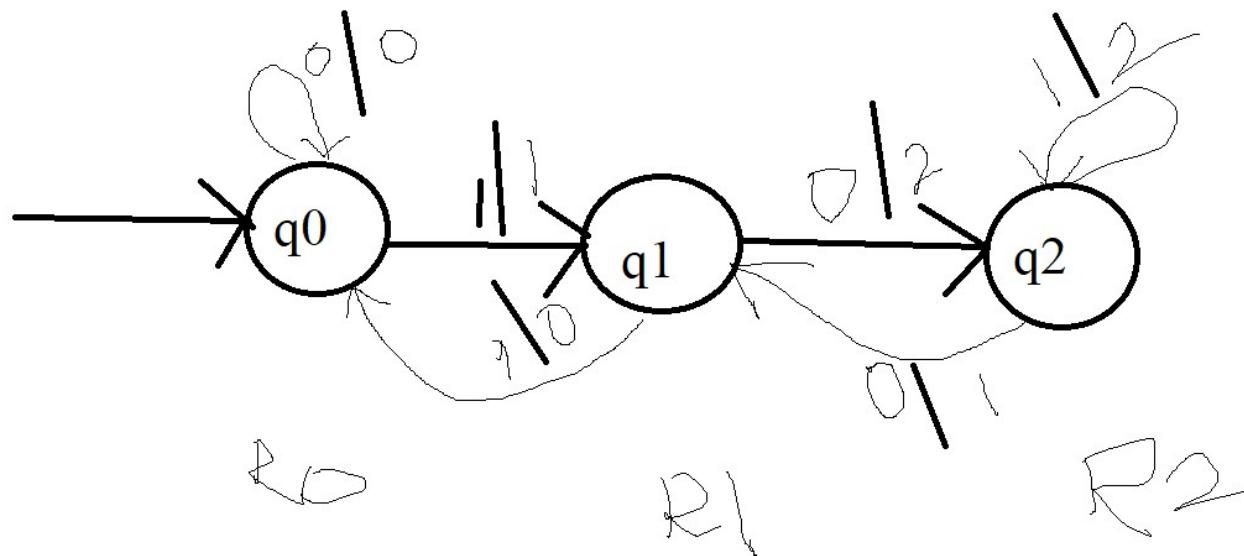


Construct a Moore Machine for testing divisibility by 2 of a binary number  
Input; 0 & 1

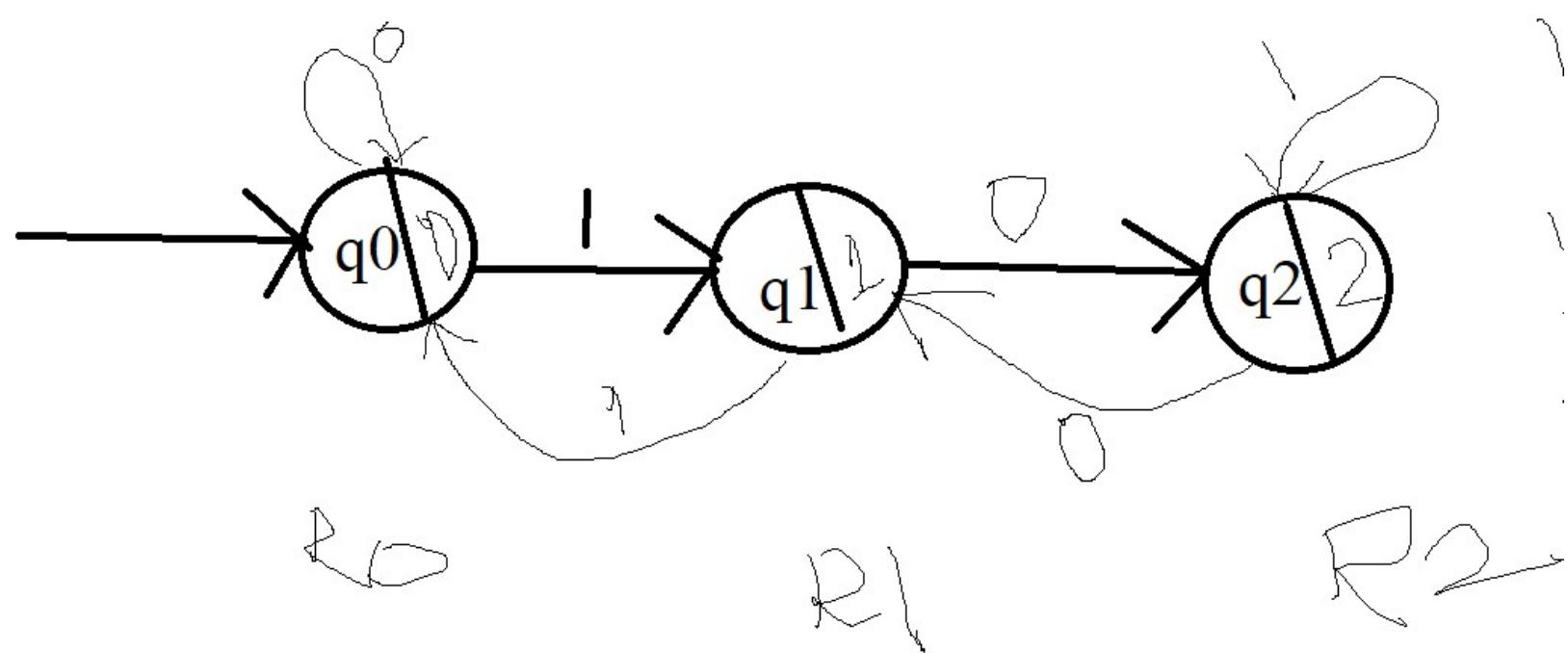


- Construct a Mealy Machine for testing divisibility by 3 of a binary number

Rem0  
Rem 1  
Rem 2



Construct a Moore Machine for testing divisibility by 3 of a binary number



Construct a DFA for testing divisibility  
by 3 of a binary number

