EEE205 – Digital Electronics (II)

Lecture 13

Xiaoyang Chen, Jiangmin Gu, Ming Xu

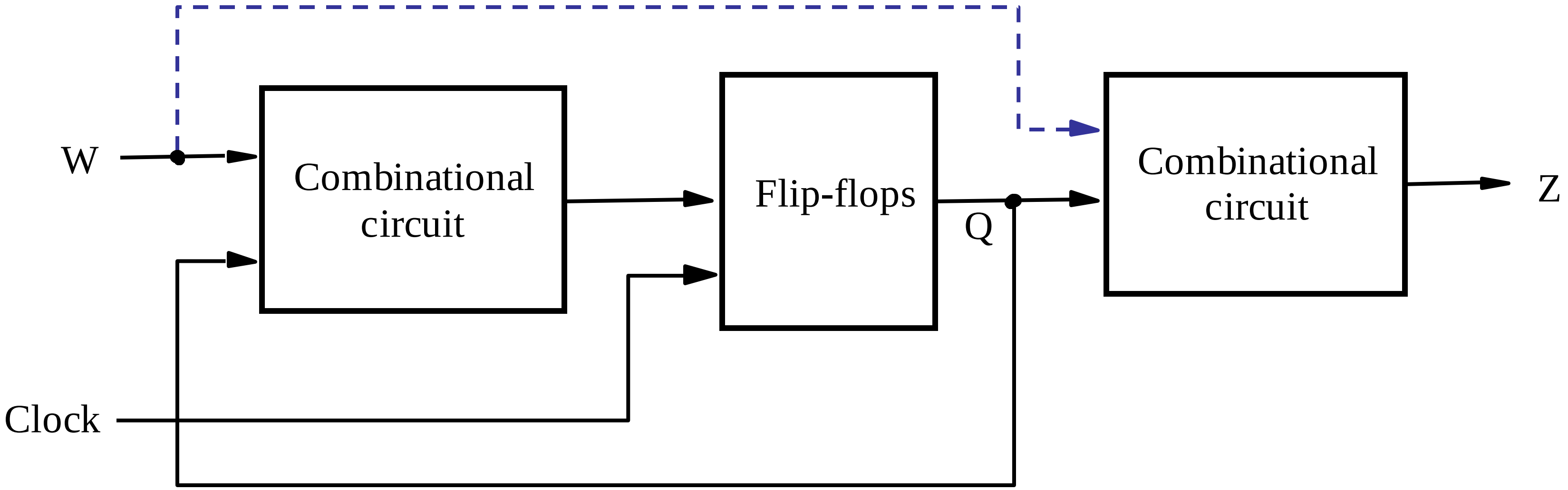
Dept of Electrical & Electronic Engineering

XJTLU

In This Session

- We have learned how to derive the state graph from a given circuit and to analyze the circuit.
- Next we will learn how to derive the state table or graph from a problem statement and to design the circuit.
- State Assignment

General Form of A Sequential Circuit.



- The sequential circuits whose outputs depend only on the state of the circuit are of **Moore type**.
- Those whose outputs depend on both the state and the inputs are of **Mealy type**.

Moore State Model.

We wish to design such a circuit:

- The circuit has one input w and one output z.
- All changes in the circuit occur on the positive edge of a clock signal.
- The output z is 1 if during the past two clock cycles w was 1. Otherwise z is 0.

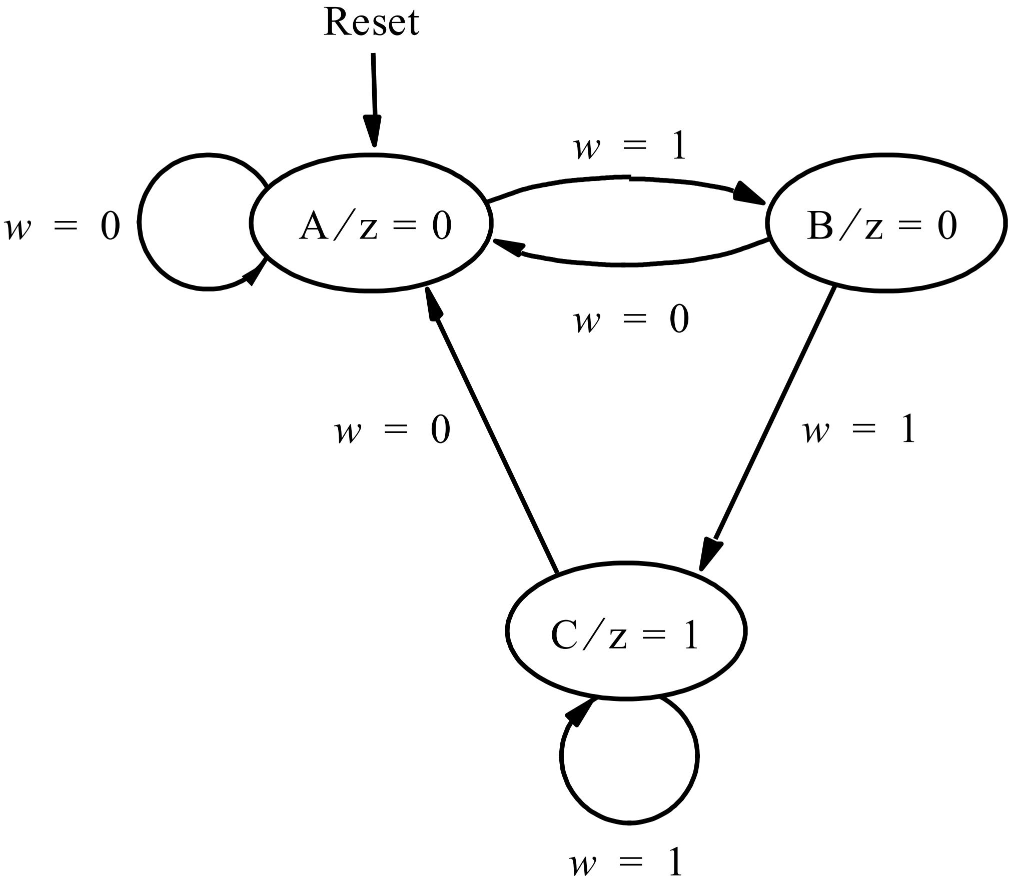
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Sequences of input and output signals.

Moore State Model.

State Diagram

- State A: w is 0 during past 1 clock cycle.
- State B: w has been 1 for just 1 clock cycle.
- State C: w has been 1 for 2 clock cycles.



Moore State Model.

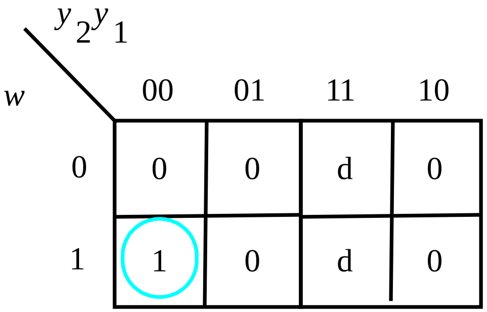
State Table

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	A	B	0
B	A	C	0
C	A	C	1

	Present state	Next state		Output
		$w = 0$	$w = 1$	z
	y_2y_1	Y_2Y_1	Y_2Y_1	
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	*dd*	*dd*	*d*

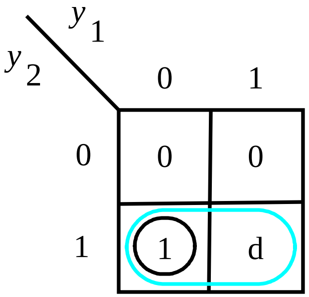
Moore State Model.

Next-State and Output Expressions



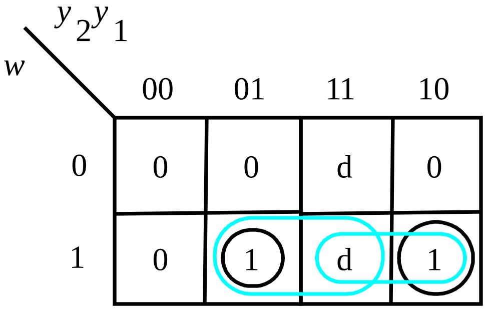
w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	1	0	d	0

$$Y_1 = w\bar{y}_1\bar{y}_2$$



y_2 \ y_1	0	1
0	0	0
1	1	d

$$z = y_2$$



w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	0	1	d	1

$$\begin{aligned} Y_2 &= wy_1 + wy_2 \\ &= w(y_1 + y_2) \end{aligned}$$

Moore State Model.

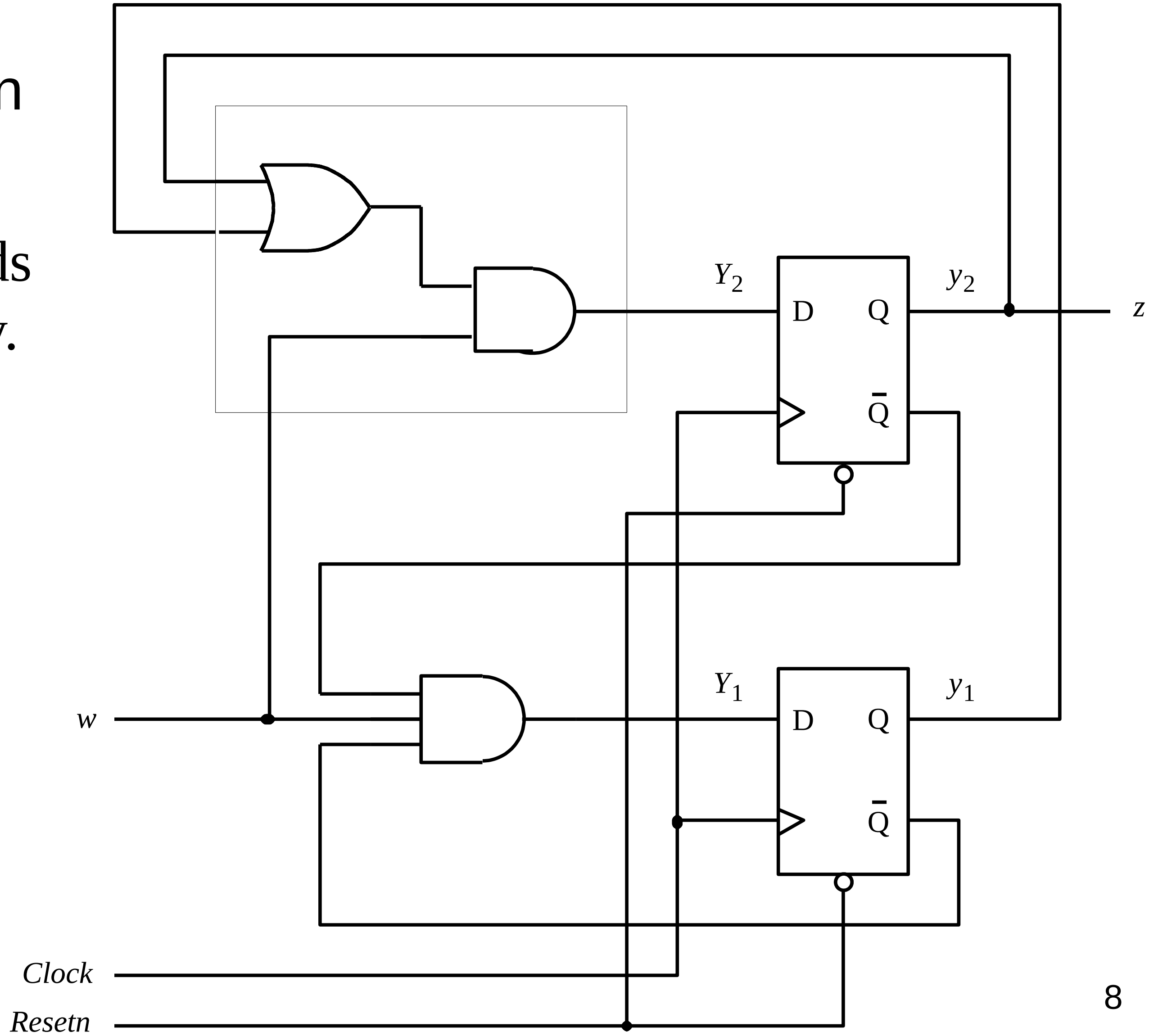
Implementation

Output z depends on the state only.

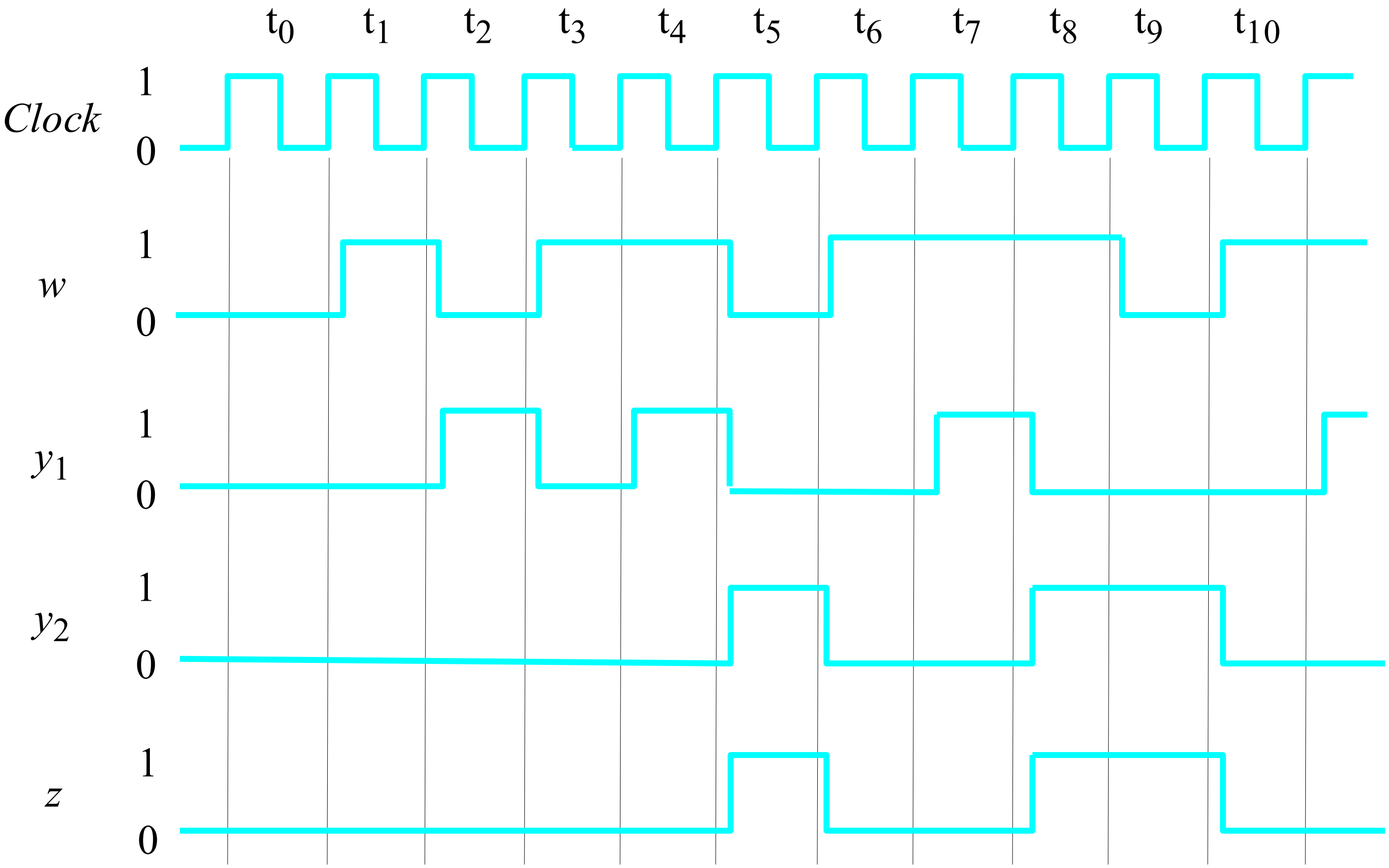
$$Y_1 = w\bar{y}_1\bar{y}_2$$

$$Y_2 = wy_1 + wy_2$$
$$= w(y_1 + y_2)$$

$$z = y_2$$



Moore State Model - Timing Diagram



$$Y_1 = w\bar{y}_1\bar{y}_2$$

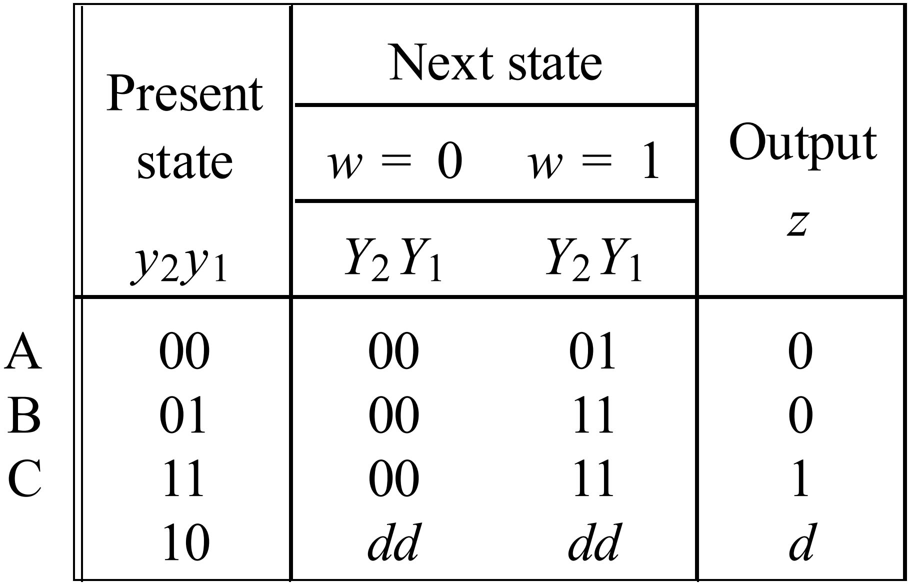
$$Y_2 = wy_1 + wy_2$$
$$= w(y_1 + y_2)$$

$$z = y_2$$

Input w changes slightly after the clock edges, because it is often an output of another circuit synchronized by the same clock.

Moore State Model.

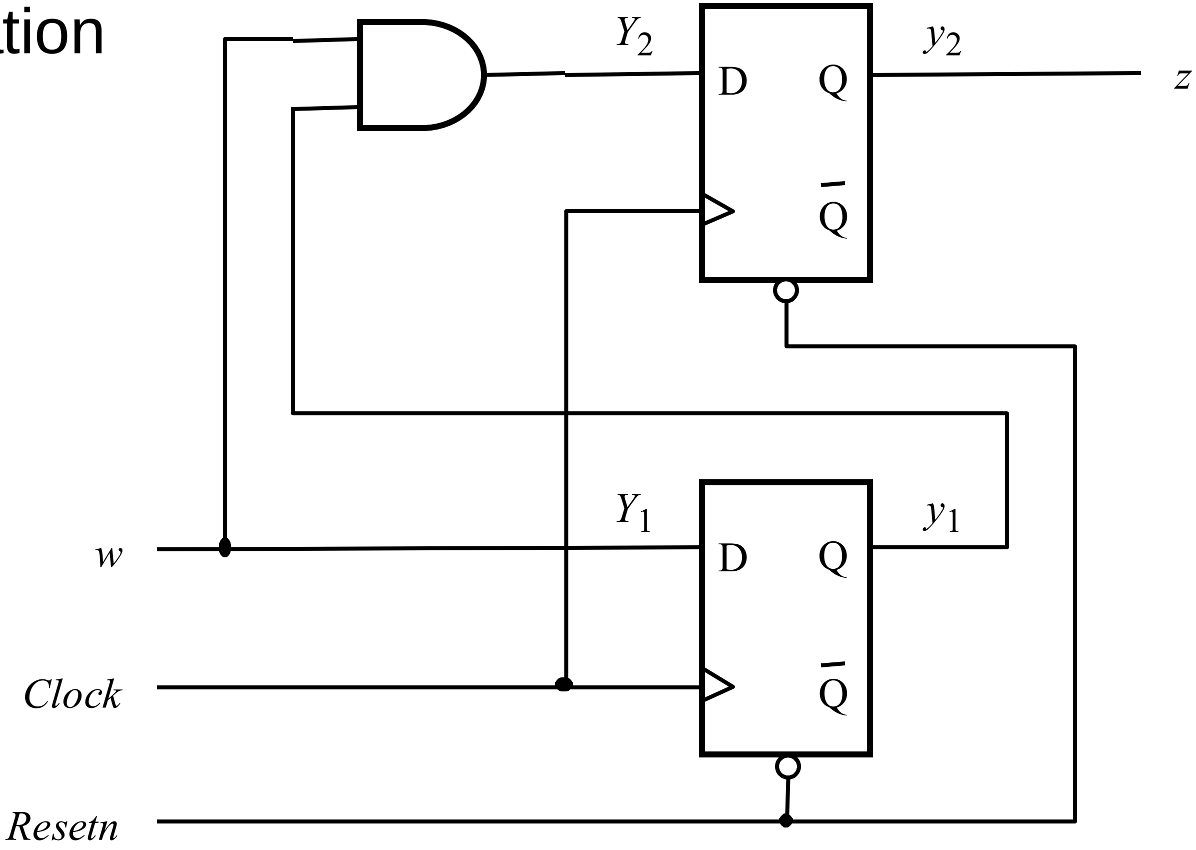
Alternative State Assignment



	Present state	Next state		Output
		$w = 0$	$w = 1$	
	y_2y_1	Y_2Y_1	Y_2Y_1	z
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	*dd*	*dd*	*d*

Moore State Model.

Improved Implementation



Mealy State Model.

We wish to design such a circuit:

- The circuit has one input w and one output z.
- The output z is 1 in the clock cycle when the second occurrence of w =1 is detected. Otherwise z is 0.

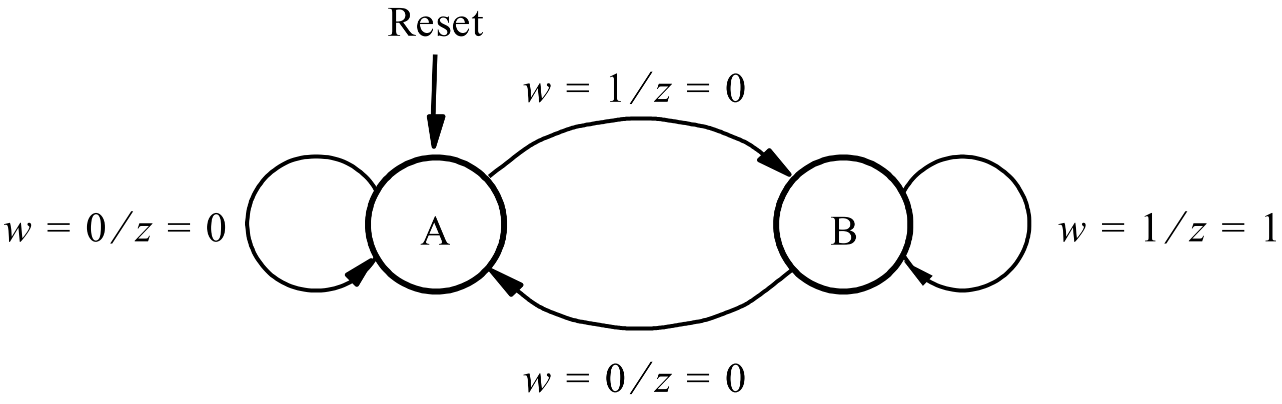
Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	1	0	0	1	1	0	0

Sequences of input and output signals.

Mealy State Model.

State Diagram

- State A: w is 0, producing an output z = 0 .
- State B: w is 1.
- If w = 1 for two or consecutive clock cycles, the machine remains in state B and produce an output z = 1.



Mealy State Model.

State Table

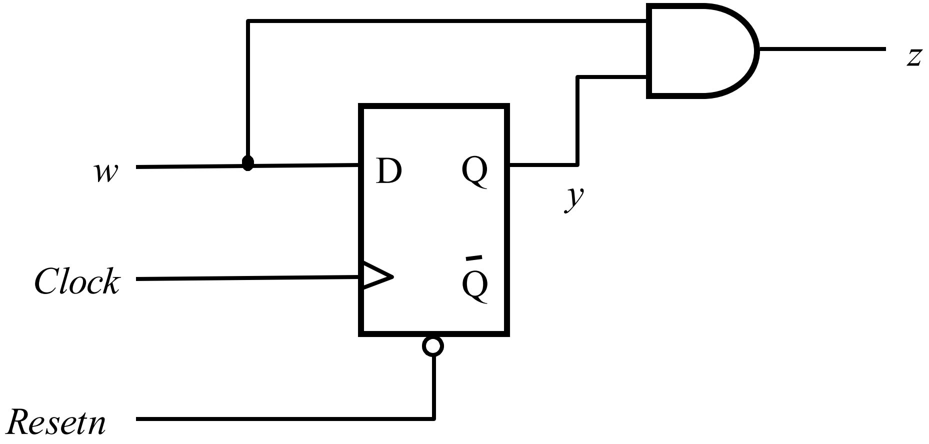
Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

Mealy State Model.

Implementation

$$Y = D = w$$
$$z = wy$$

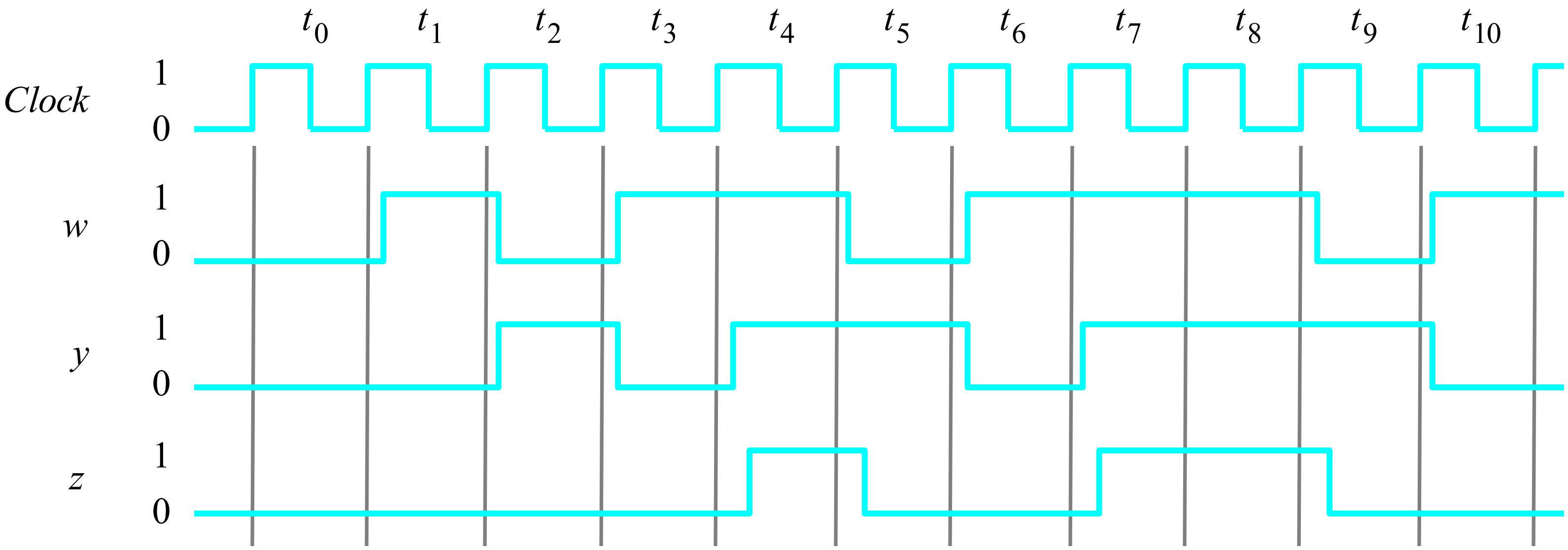


Output z depends on both the state and the input.

Mealy State Model - Timing Diagram

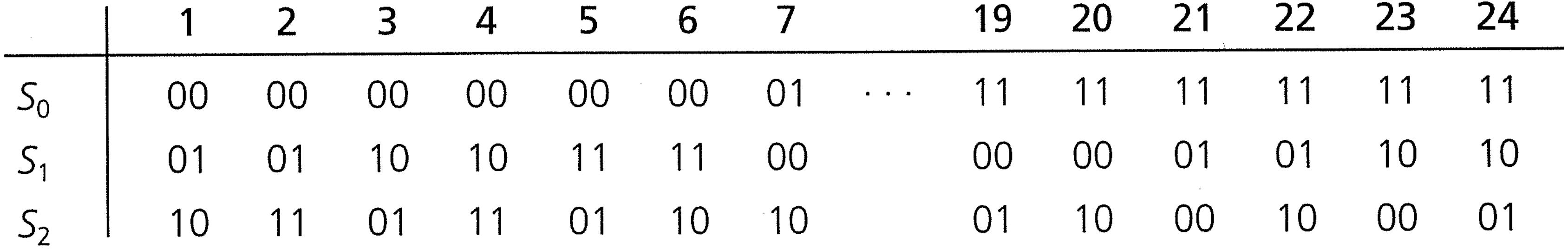
$$Y = D = w$$

$$z = wy$$



State Assignments

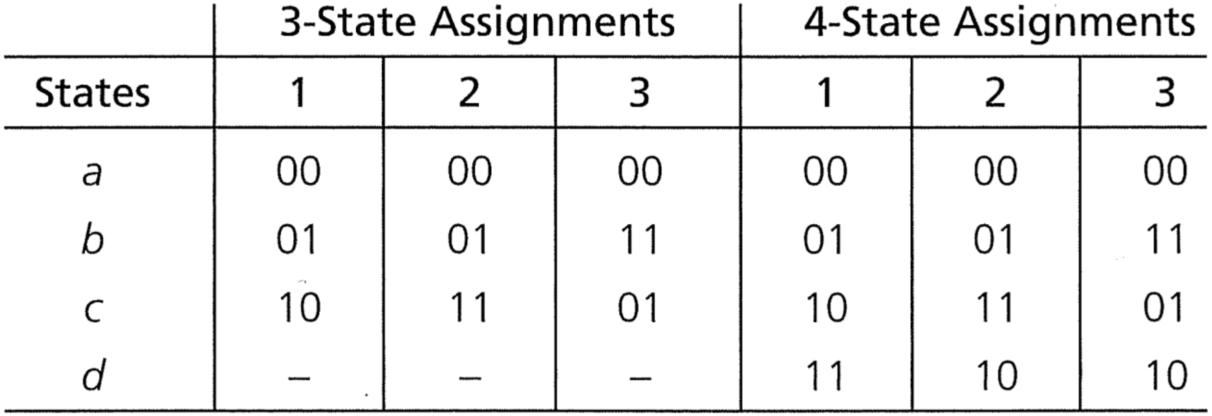
- The state assignment should be made so as to minimize the cost of the logic.
- If the number of states is small, it may be possible to try all possible state assignments.
- For a state table with 3 states, there are 4 × 3 × 2 = 24 possible state assignments.
- But many of these are equivalent in cost of realization.



	1	2	3	4	5	6	7		19	20	21	22	23	24
S_0	00	00	00	00	00	00	01	· · ·	11	11	11	11	11	11
S_1	01	01	10	10	11	11	00		00	00	01	01	10	10
S_2	10	11	01	11	01	10	10		01	10	00	10	00	01

State Assignments

- If we interchange two columns in an assignment, the cost will be unchanged, e.g. 1/3, 2/4, 5/6.
- If symmetrical flip-flops such as J-K, S-R or T are used, complementing one or more columns of an assignment will not change the cost, e.g. 2/7, 6/19.
- So there are only 3 non-equivalent assignments for 3 or 4 states.



	3-State Assignments			4-State Assignments		
States	1	2	3	1	2	3
a	00	00	00	00	00	00
b	01	01	11	01	01	11
c	10	11	01	10	11	01
d	–	–	–	11	10	10

State Assignments

One-Hot State Assignment

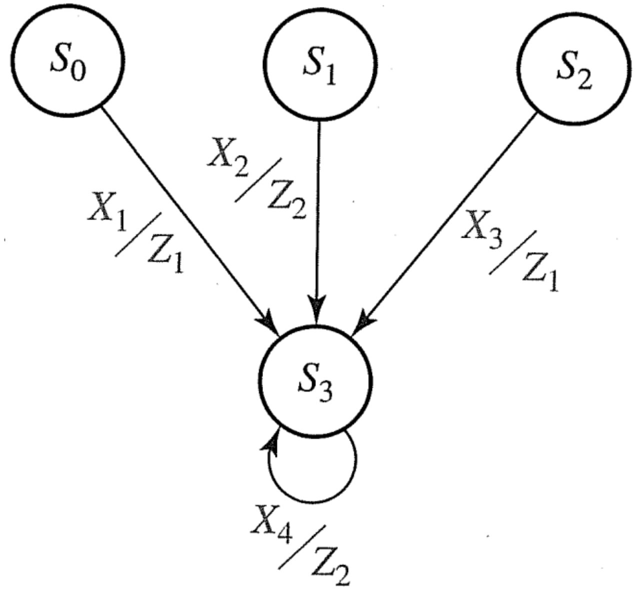
- For FPGAs it may not be important to minimize the number of flip-flops used in the design.
- The one-hot assignment uses one flip-flop for each state and only one flip-flop is set to 1 in each state.

e.g. a system with 4 states could use 4 flip-flops (Q_0, Q_1, Q_2, Q_3) with the state assignment:

$$S_0\text{: } Q_0\,Q_1\,Q_2\,Q_3 = 1000,\quad S_1\text{: } 0100,\quad S_2\text{: } 0010,\quad S_3\text{: } 0001$$

State Assignments

$$Q_3^+ = X_1\,(Q_0\,Q_1'\,Q_2'\,Q_3') \;+\; X_2\,(Q_0'\,Q_1\,Q_2'\,Q_3') \;+$$
$$X_3\,(Q_0'\,Q_1'\,Q_2\,Q_3') \;+\; X_4\,(Q_0'\,Q_1'\,Q_2'\,Q_3)$$



$Q_0 = 1$ implies $Q_1 = Q_2 = Q_3 = 0$,

the $Q_1'\,Q_2'\,Q_3'$ term is redundant

$$Q_3^+ = X_1Q_0 \;+\; X_2Q_1 \;+\; X_3Q_2 \;+\; X_4Q_3$$

Each term in the next-state equation contains exactly one state variable.