



Chapter 4

STATE MACHINE



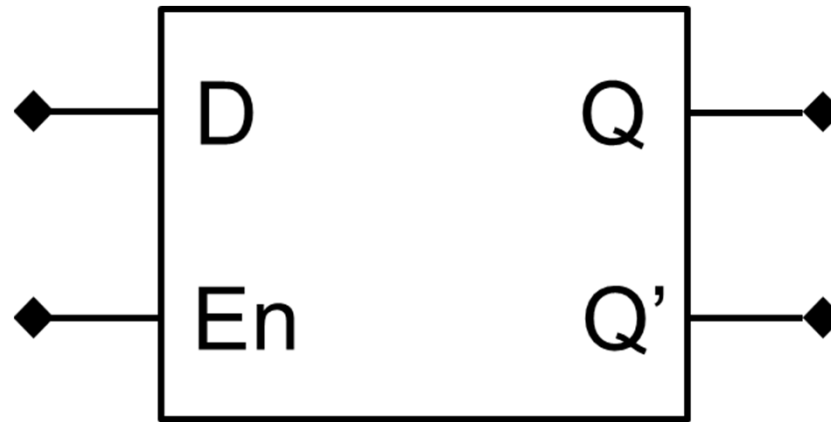
Faculty of Computer Science and Engineering
Department of Computer Engineering

Nguyen Quang Huy

huynguyen@cse.hcmut.edu.vn

Latch

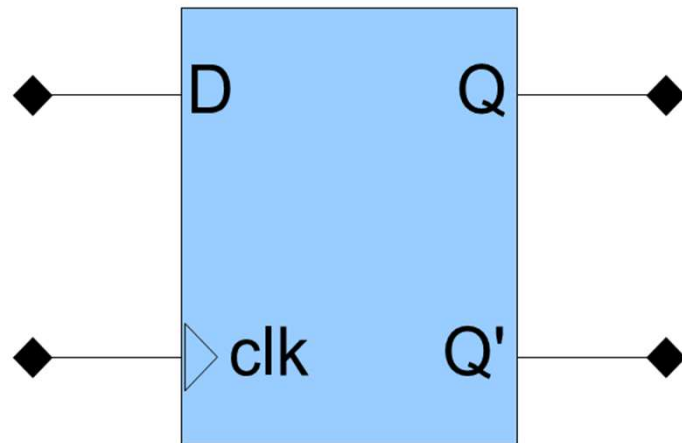
- Tích cực mức (*level sensitive*)



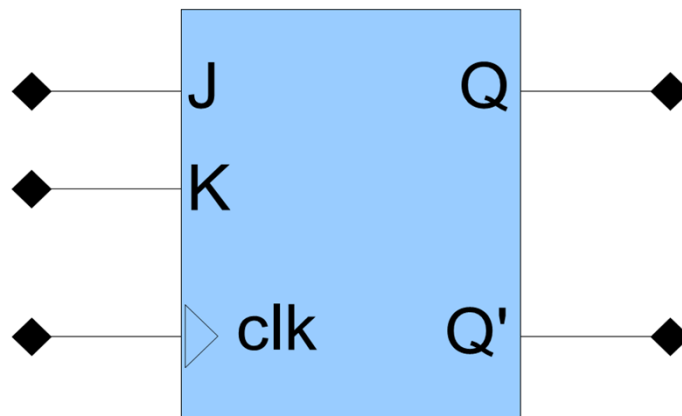
En	D	Q(next)	Comment
0	0	Q	Hold state
1	0	0	Clear
1	1	1	Set

Flip-Flop

- Tích cực cạnh (*edge sensitive*)



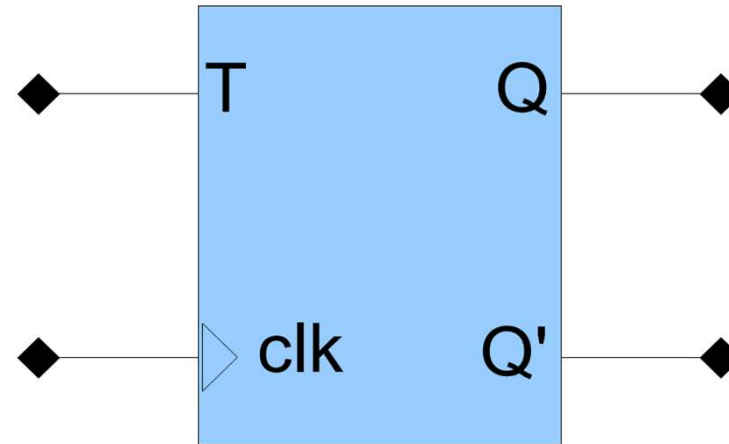
D	Q(next)
0	0
1	1



J	K	Q(next)	Comment
0	0	Q	Hold state
0	1	0	Clear
1	0	1	Set
1	1	Q'	Toggle

T Flip-Flop

- Tích cực cạnh (*edge sensitive*)



T	Q	Q(next)	Comment
0	0	0	Hold state
0	1	1	
1	0	1	Toggle
1	1	0	

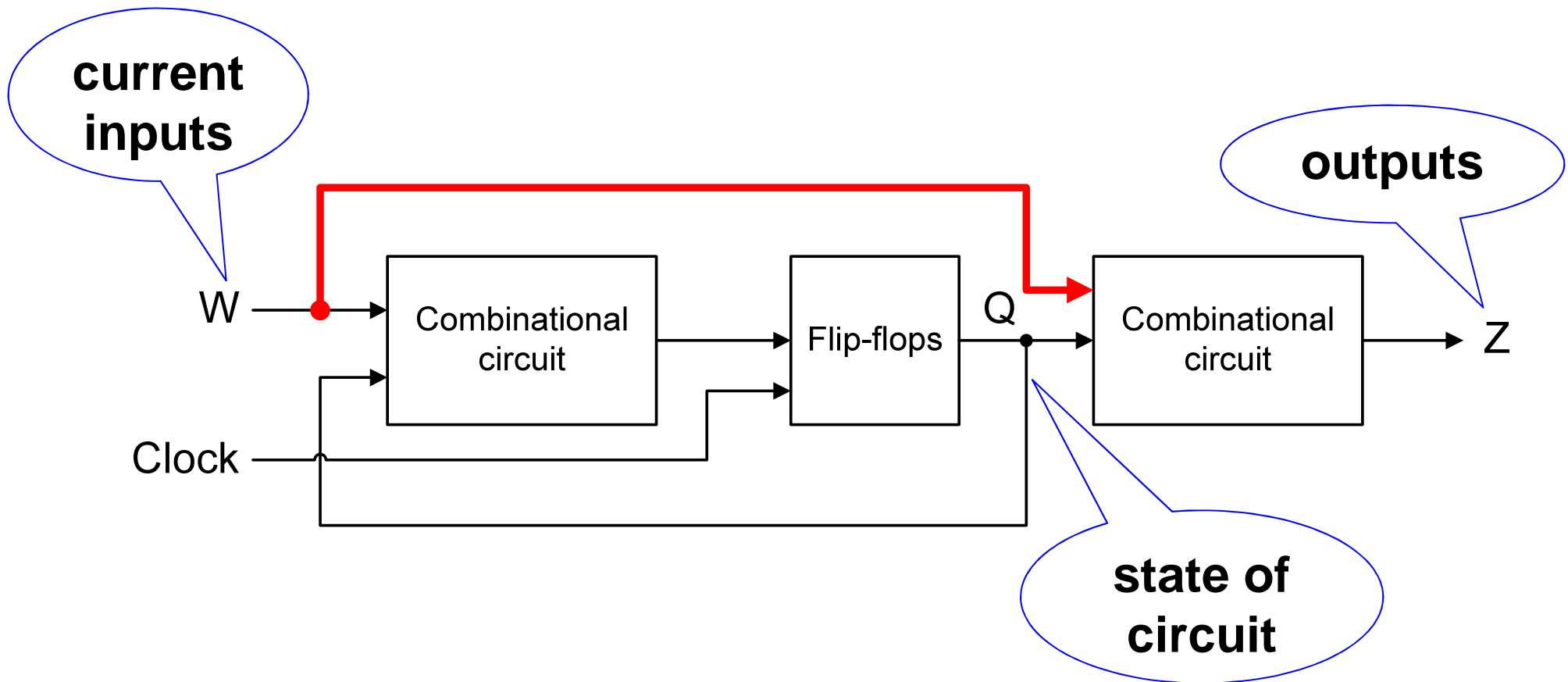
State machine

Moore vs. Mealy
machine

Design technique

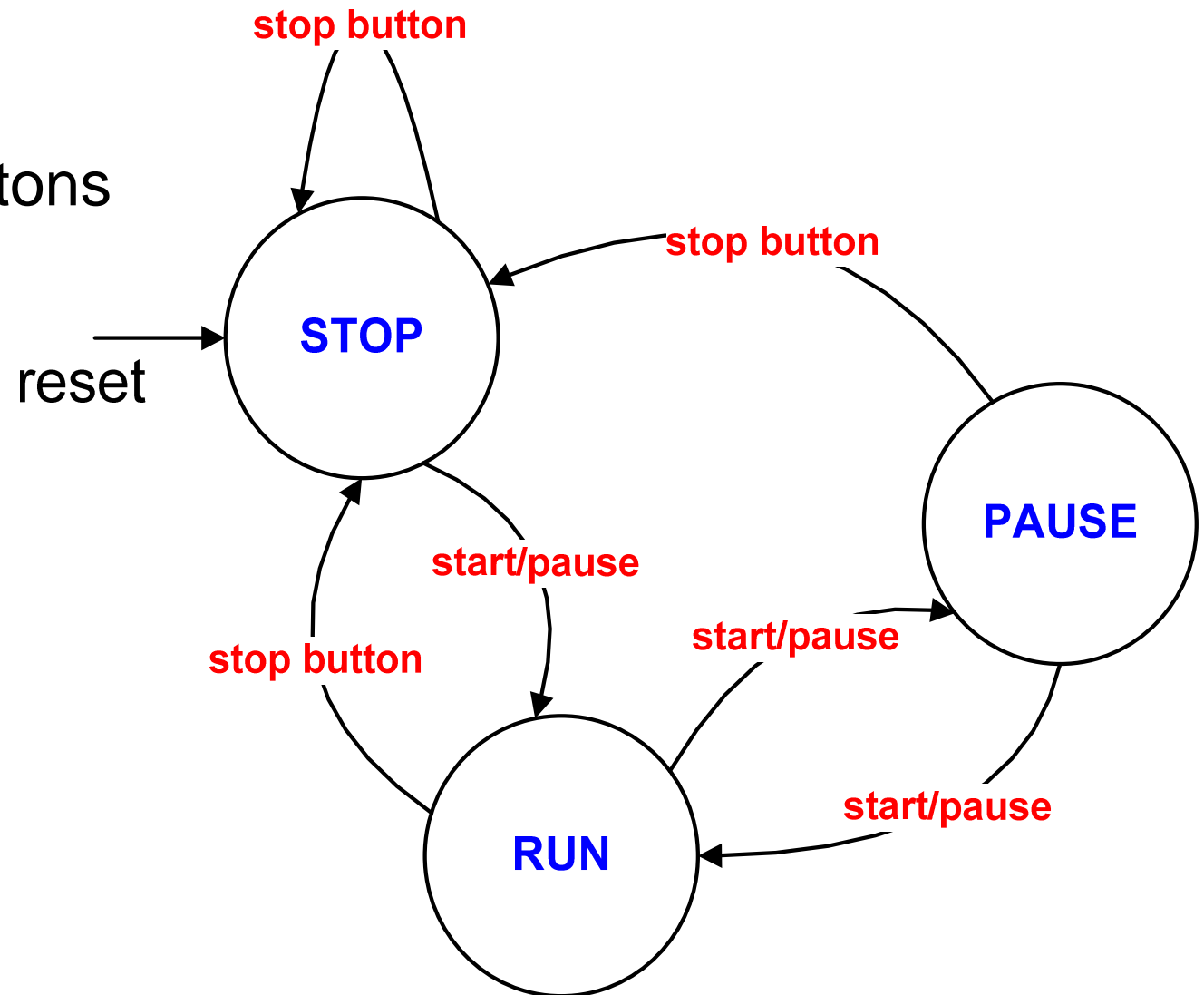
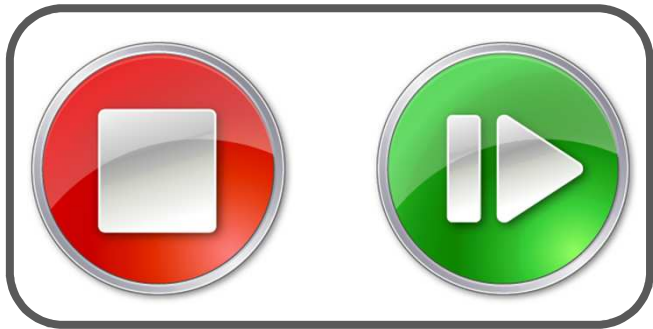
Sequential Circuit

- Synchronous vs. asynchronous

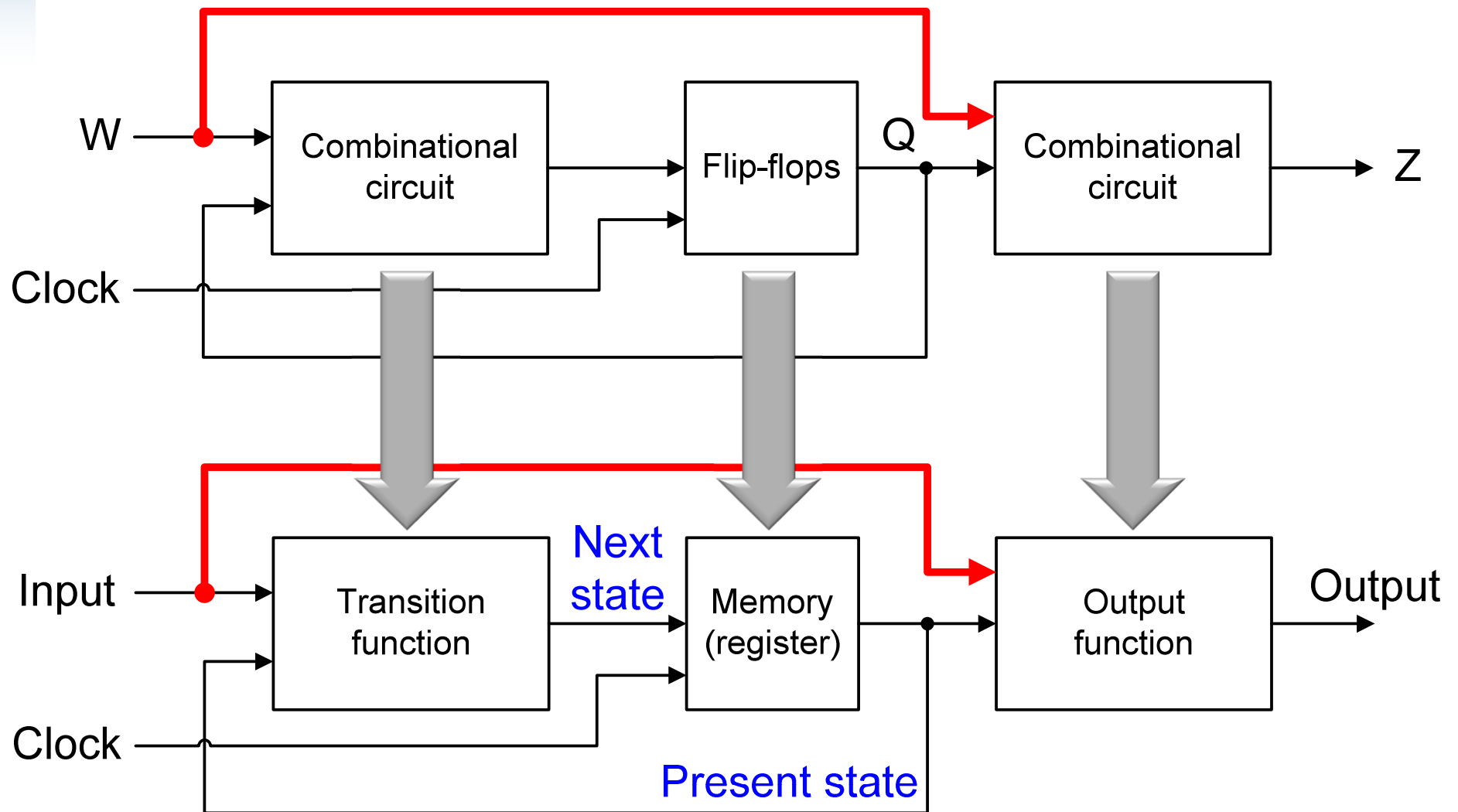


State & State Machine

- Music Player
 - 3 states
 - 2 control buttons



State & State Machine



State & State Machine

- **Outputs** are functions of present states (with or without inputs)
- **Next states** are functions of present states and inputs
 - Used to implement circuits that control other circuits

Finite State Machine (FSM)

- FSM or automata, or simply state machine
 - Using a finite number of states
 - Machine is in only one state at a time (current state)
 - Deterministic vs. nondeterministic
- Design of FSMs involves
 - Defining states
 - Defining transitions between states
 - Optimization / minimization

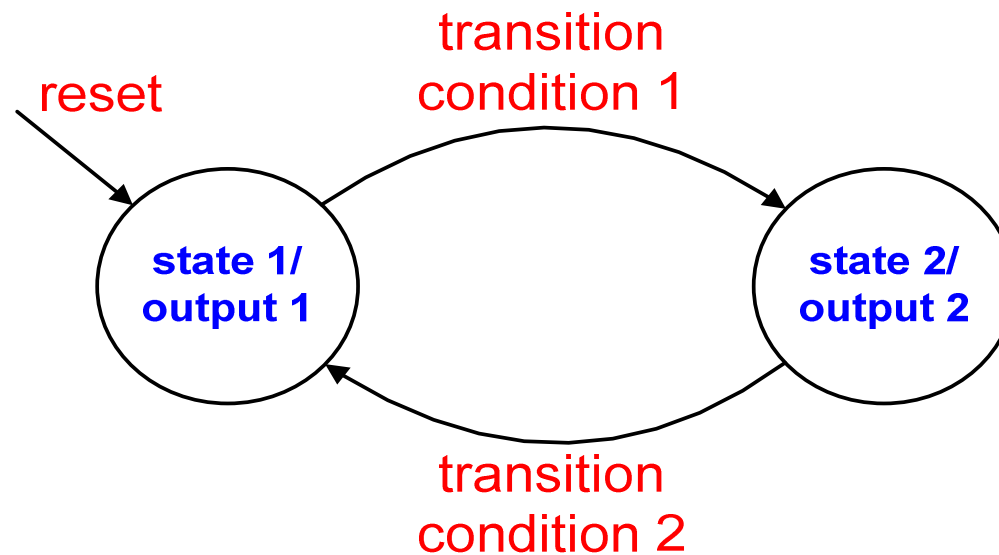
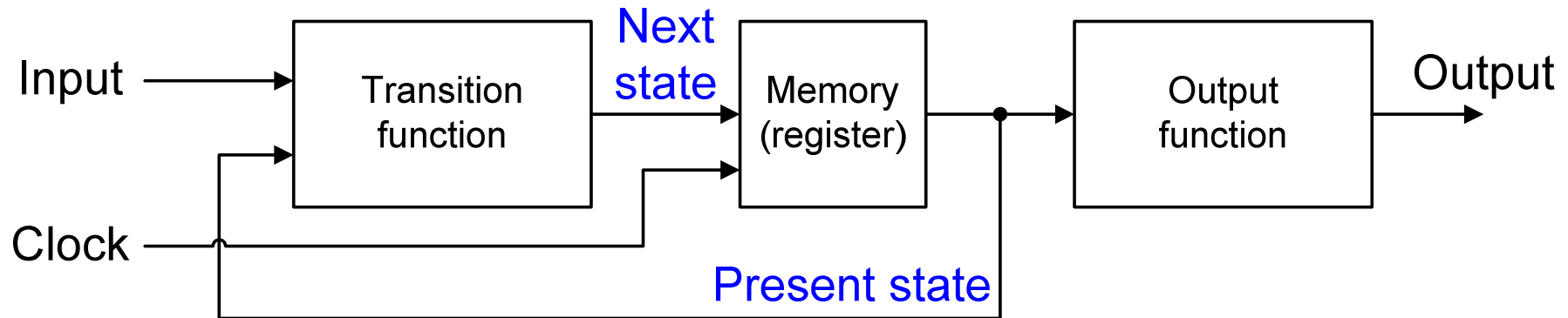
State machine

**Moore vs. Mealy
machine**

Design technique

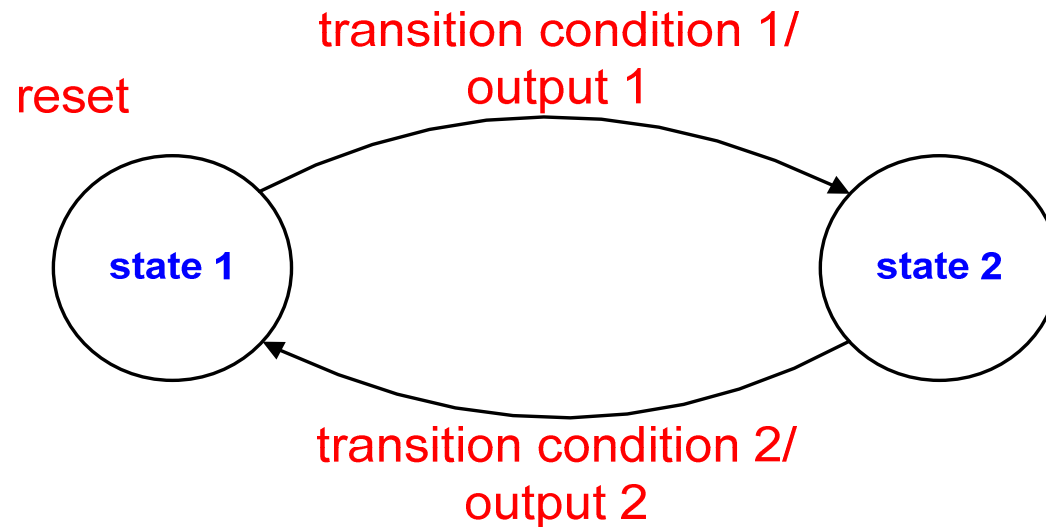
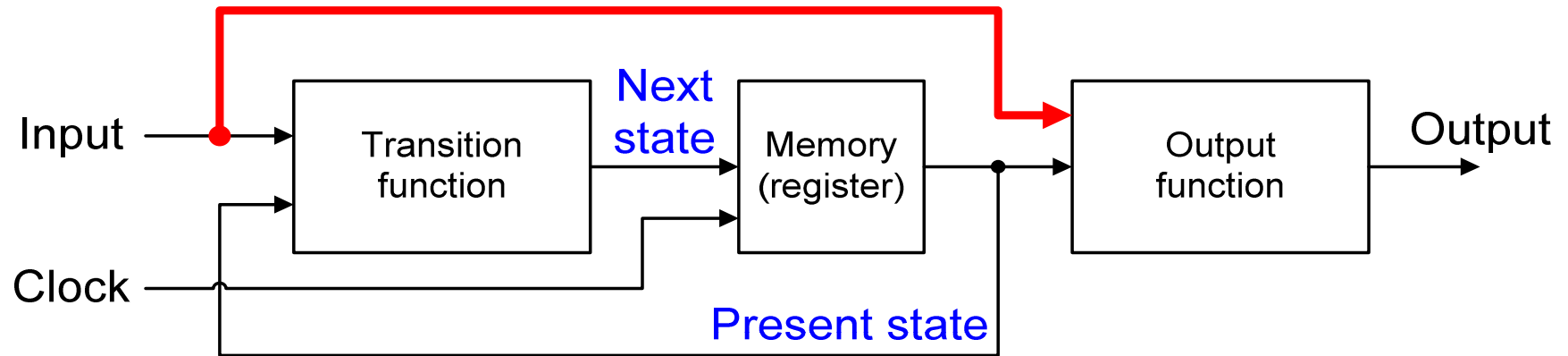
Moore Machine

- Output is a function of **present state only**



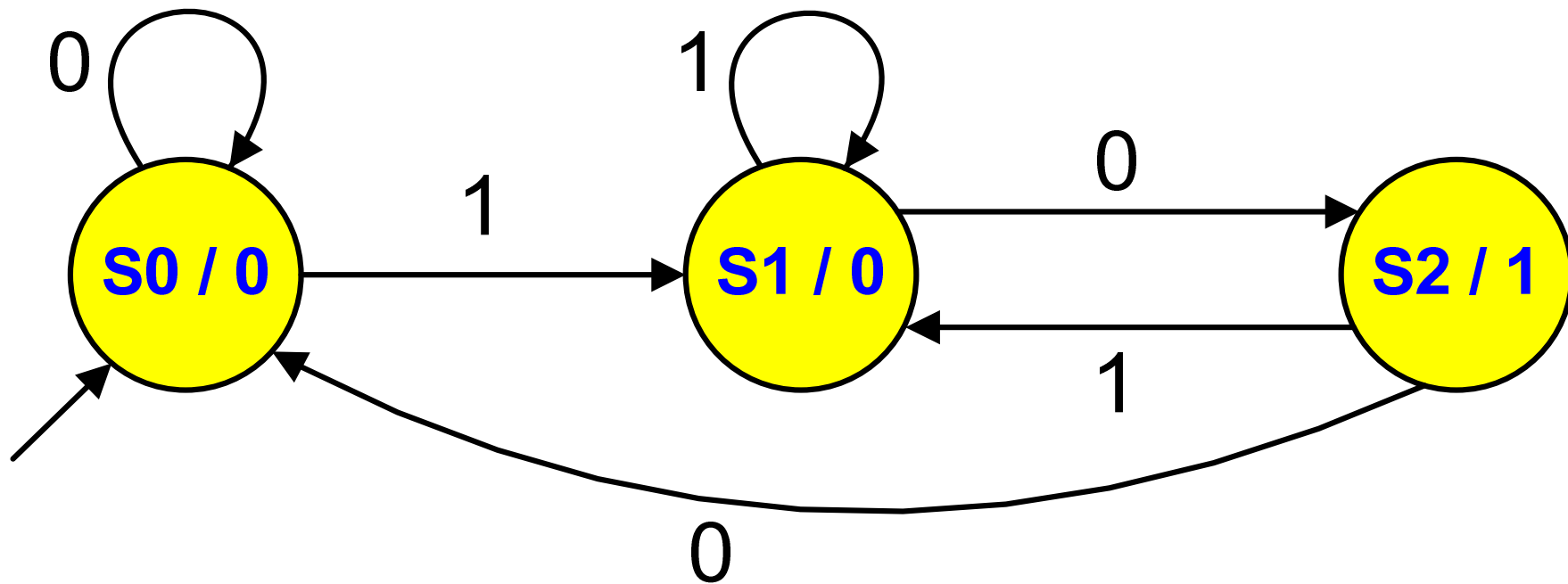
Mealy Machine

- Output is a function of **present state and input**



Example (1)

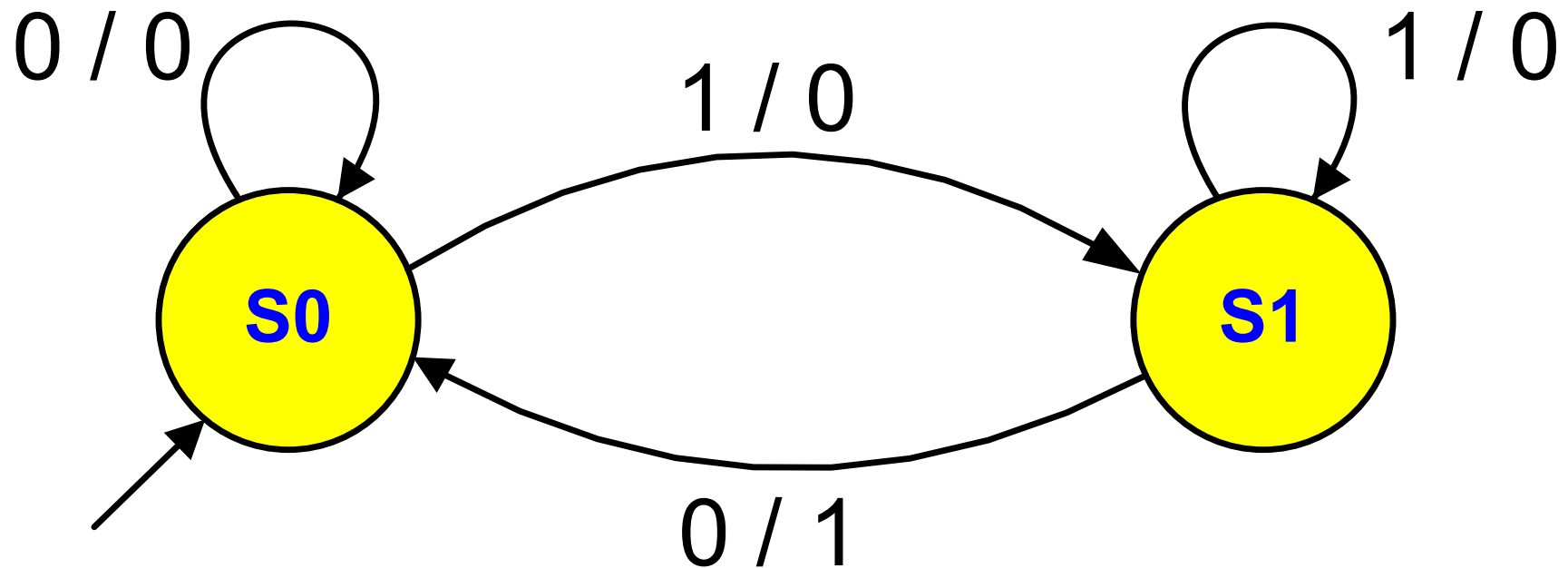
- Design a FSM that recognizes sequence 10



Moore FSM

Example (1)

- Design a FSM that recognizes sequence 10



Mealy FSM

Moore vs. Mealy FSM (1)

- Functional equivalence
- Different complexity
- Moore FSM has no combinational path between inputs and outputs
 - Moore FSM is less likely to introduce delays in critical path

Moore vs. Mealy FSM (2)

- Mealy FSM has richer description and usually requires smaller number of states
 - Smaller circuit area
- Mealy FSM computes outputs as soon as inputs change
 - Mealy FSM responds one clock cycle sooner than equivalent Moore FSM

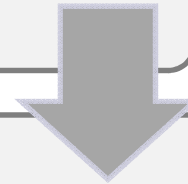
State Encoding (1)

- Binary: state encoded as a binary number
 - Small number of used flip-flops
 - Potentially complex transition functions leading to slow implementations
- One-Hot: only one bit is active
 - Number of used flip-flops as big as number of states
 - Simple and fast transition functions
 - Preferable coding technique in FPGAs

State Encoding (2)

State	Binary Code	One-Hot Code
S0	000	10000000
S1	001	01000000
S2	010	00100000
S3	011	00010000
S4	100	00001000
S5	101	00000100
S6	110	00000010
S7	111	00000001

State machine



Moore vs. Mealy
machine



Design technique

1. Construct a state/output table



2. Perform state minimization



3. Encode state (state assignment)



4. Select F/F type



5. Derive state transition equations



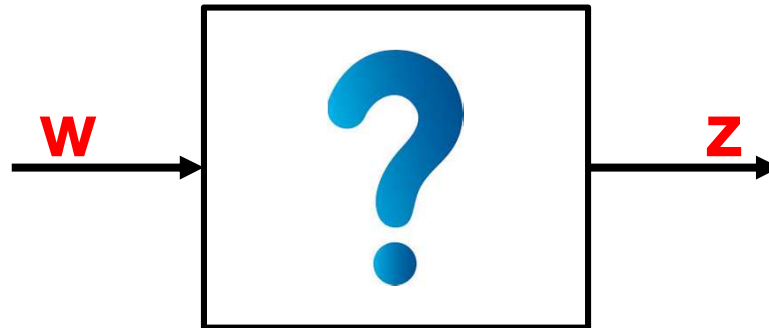
6. Derive output equation



7. Draw circuit

Example

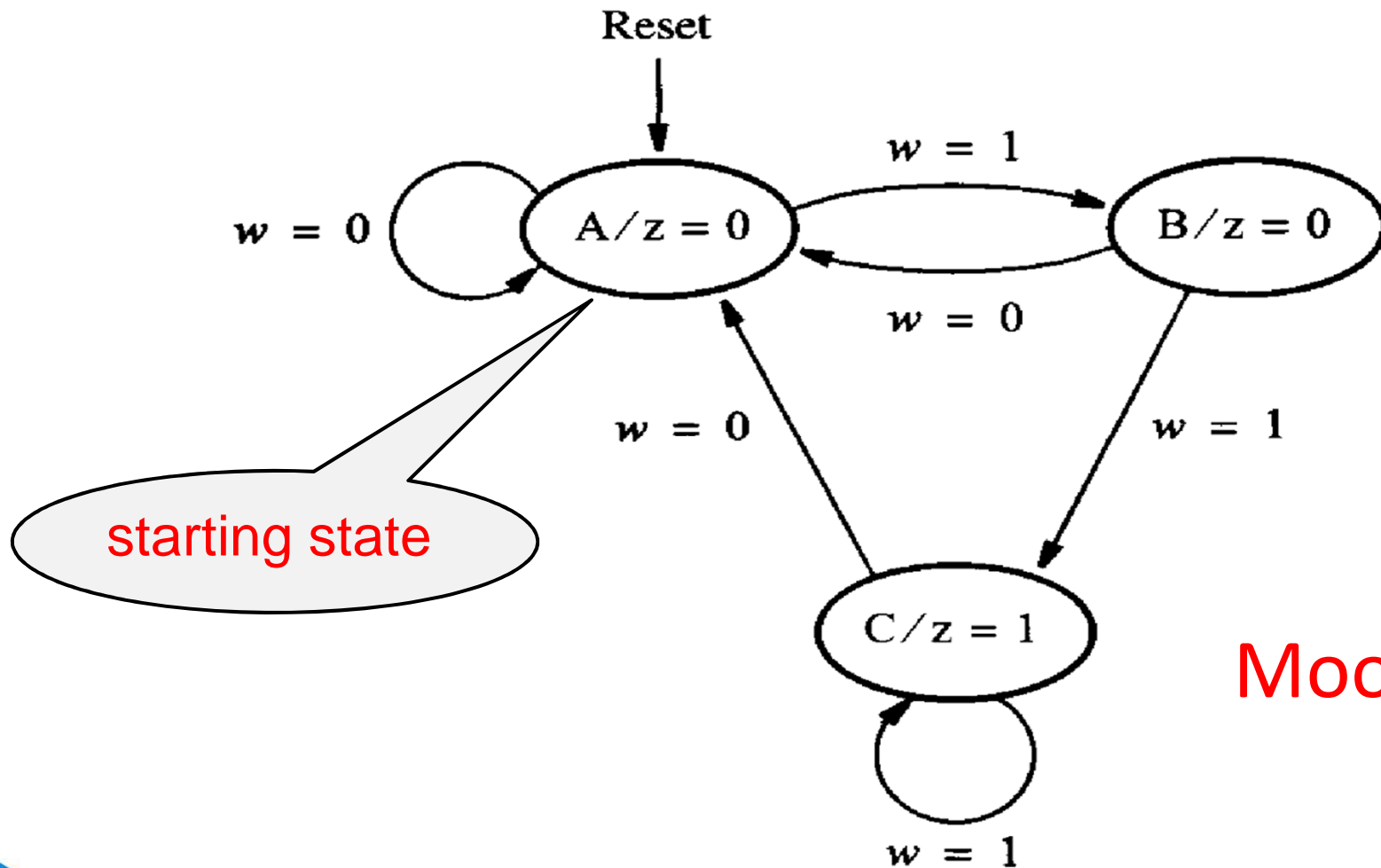
- **z** = 1 if **w** = 1 during two immediately preceding clock cycles



Clock cycle:	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

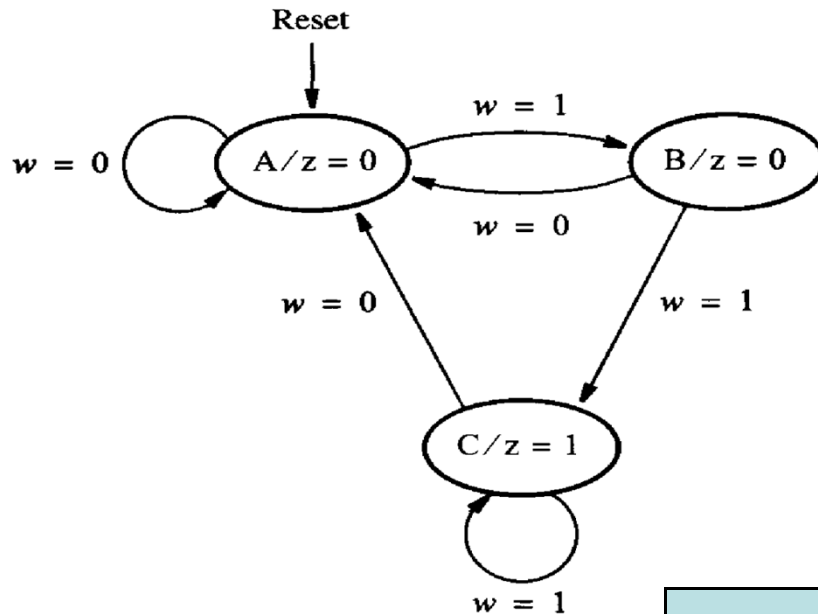
1. Construct state/output table

- State Diagram



Moore FSM

1. Construct state/output table



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

2. Perform state minimization

- Often performed by Tool
- No minimization in this example

3. Construct state/output table

- State assignment – Binary code

	Present state <i>y₂y₁</i>	Next state		Output <i>z</i>
		<i>w = 0</i>	<i>w = 1</i>	
		<i>Y₂Y₁</i>	<i>Y₂Y₁</i>	
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	xx	xx	x

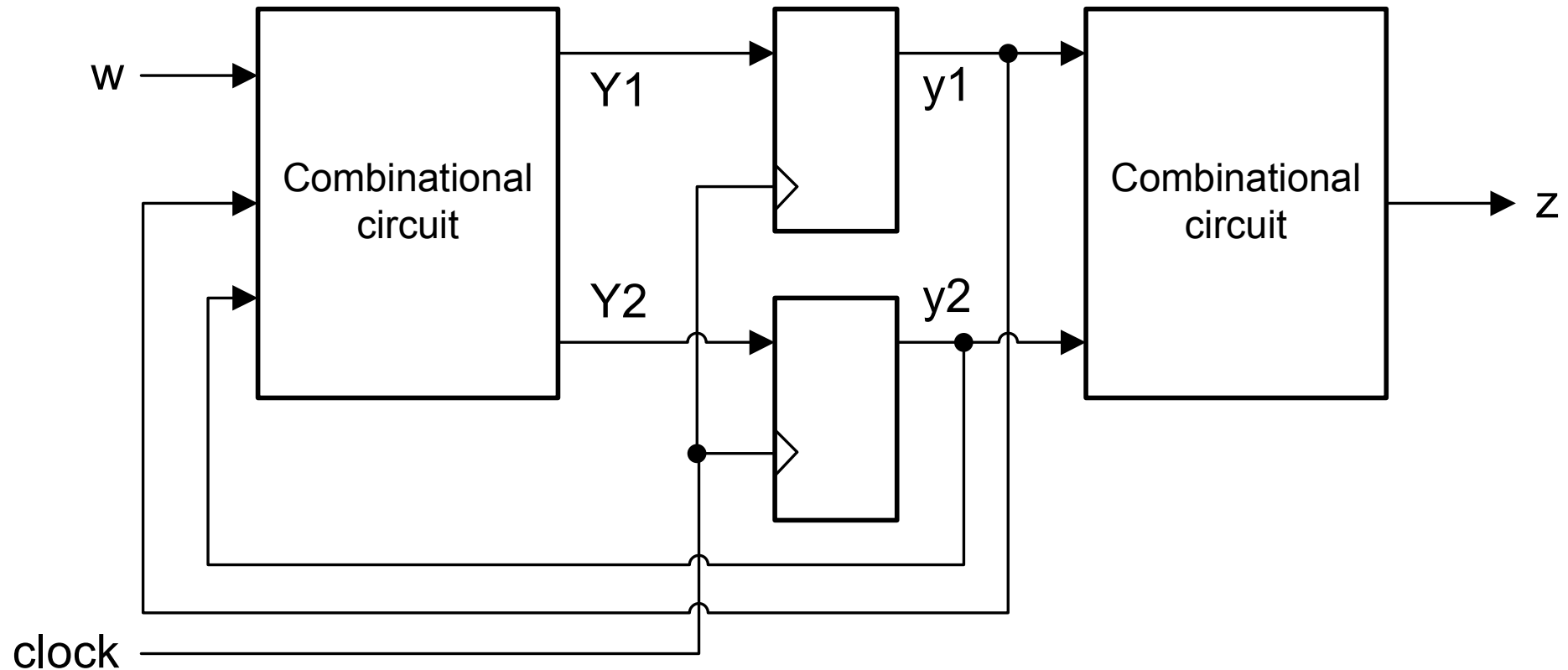
3. Construct state/output table

- State assignment – One-hot code

	Present state $y_2y_1y_0$	Next state		Output z
		$w = 0$	$w = 1$	
		$Y_2Y_1Y_0$	$Y_2Y_1Y_0$	
A	001	001	010	0
B	010	001	100	0
C	100	001	100	1

4. Select Flip-Flop type

- We select D-type in this example



*General sequential circuit
(Moore Machine)*

5. Derive state transition equations

		y ₂ y ₁			
		00	01	10	11
w	0	0	0	x	0
	1	1	0	x	0

$$Y1 = w.y1'.y2'$$

		y ₂ y ₁			
		00	01	10	11
w	0	0	0	x	0
	1	0	1	x	1

$$Y2 = w.(y1 + y2)$$

6. Derive output equation

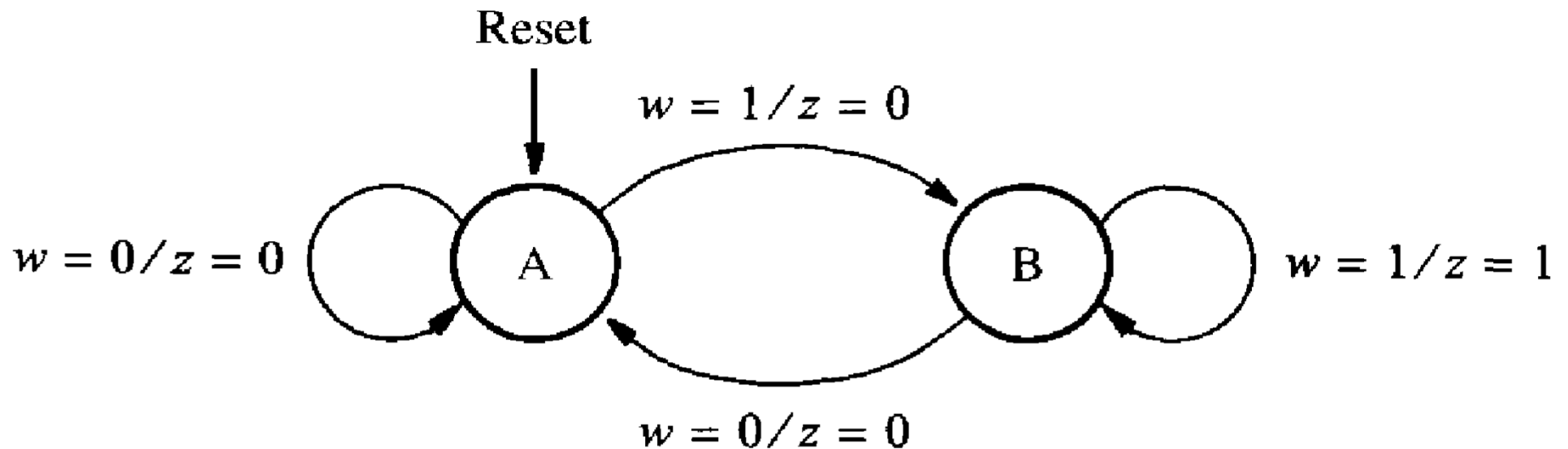
		y1	
		0	1
y2	0	0	0
	1	1	x

$$z = y2$$

7. Draw circuit

Design Steps - Mealy State Model

- State diagram



Design Steps - 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

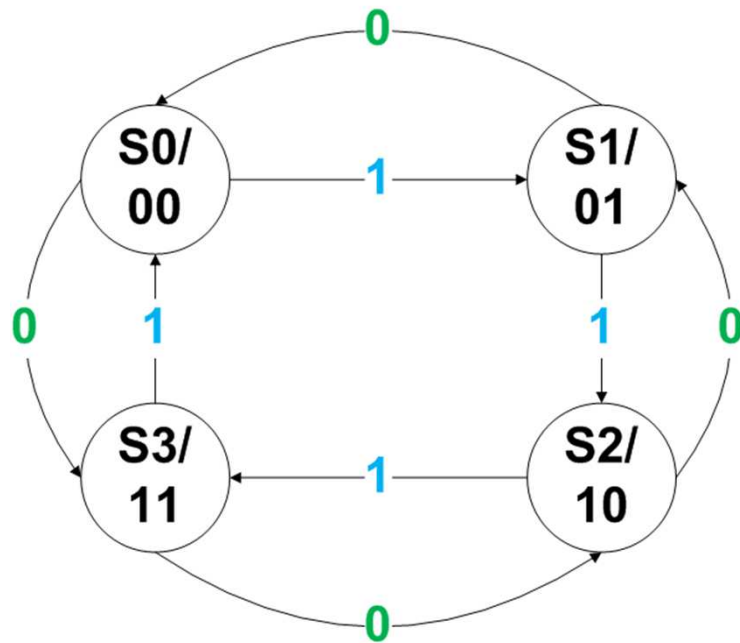
Design Steps - Mealy State Model

- State assignment

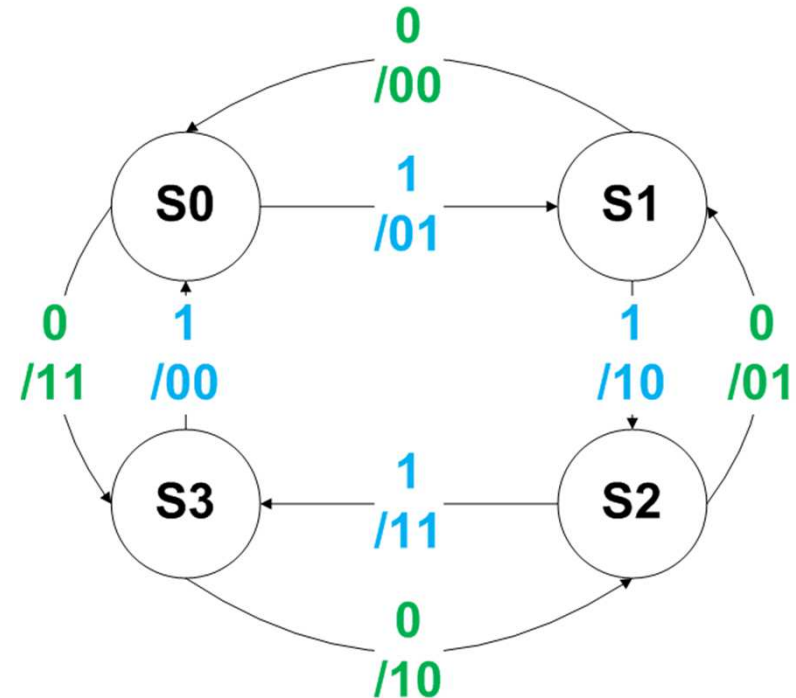
	Present state	Next state		Output z	
		w = 0	w = 1	w = 0	w = 1
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

Exercise

- Design 2-bit counter with up/down input controller
 - Up/Down = 1 : 00 – 01 – 10 – 11 – 00 - ...
 - Up/Down = 0 : 00 – 11 – 10 – 01 – 00 - ...



Moore FSM

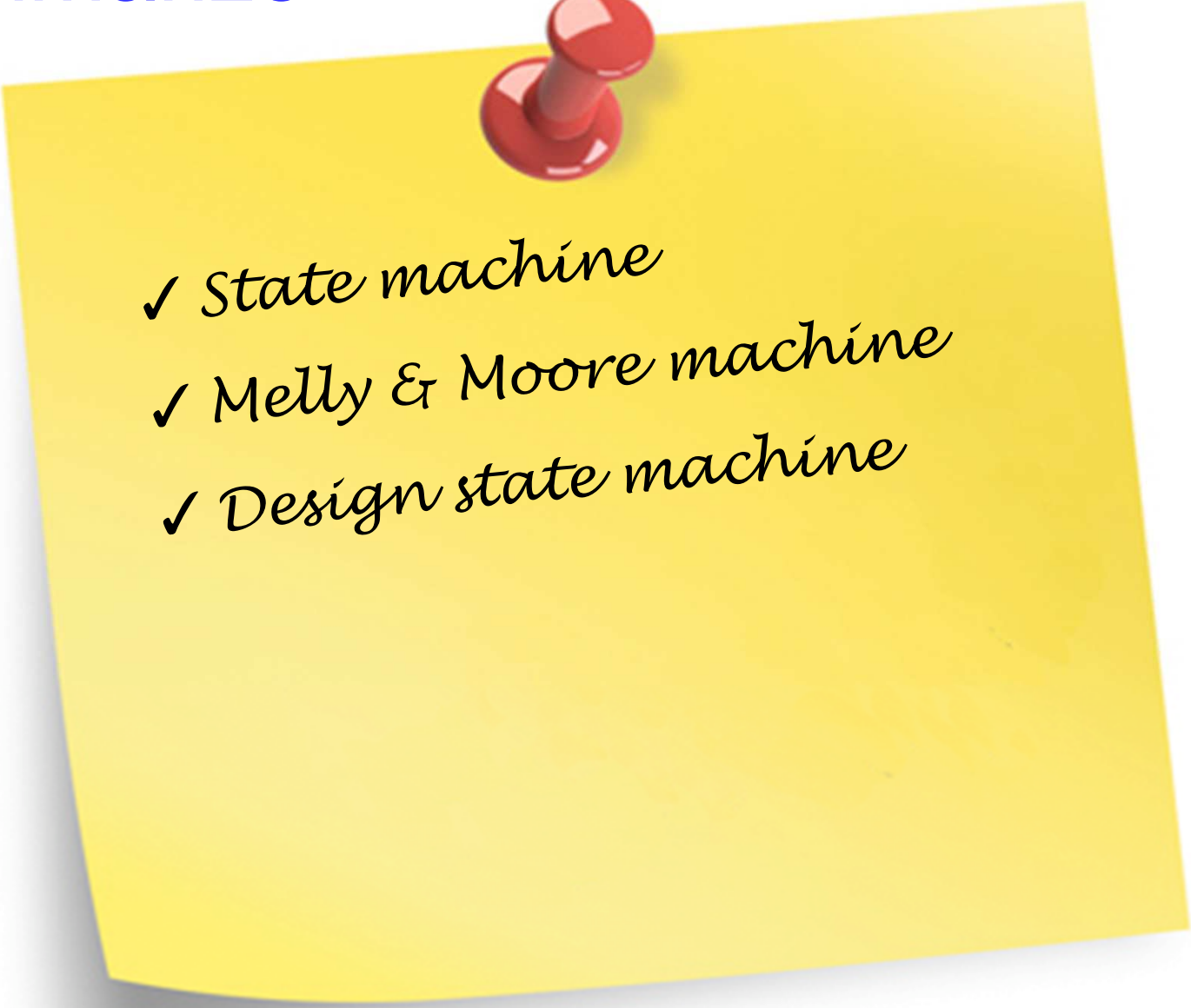


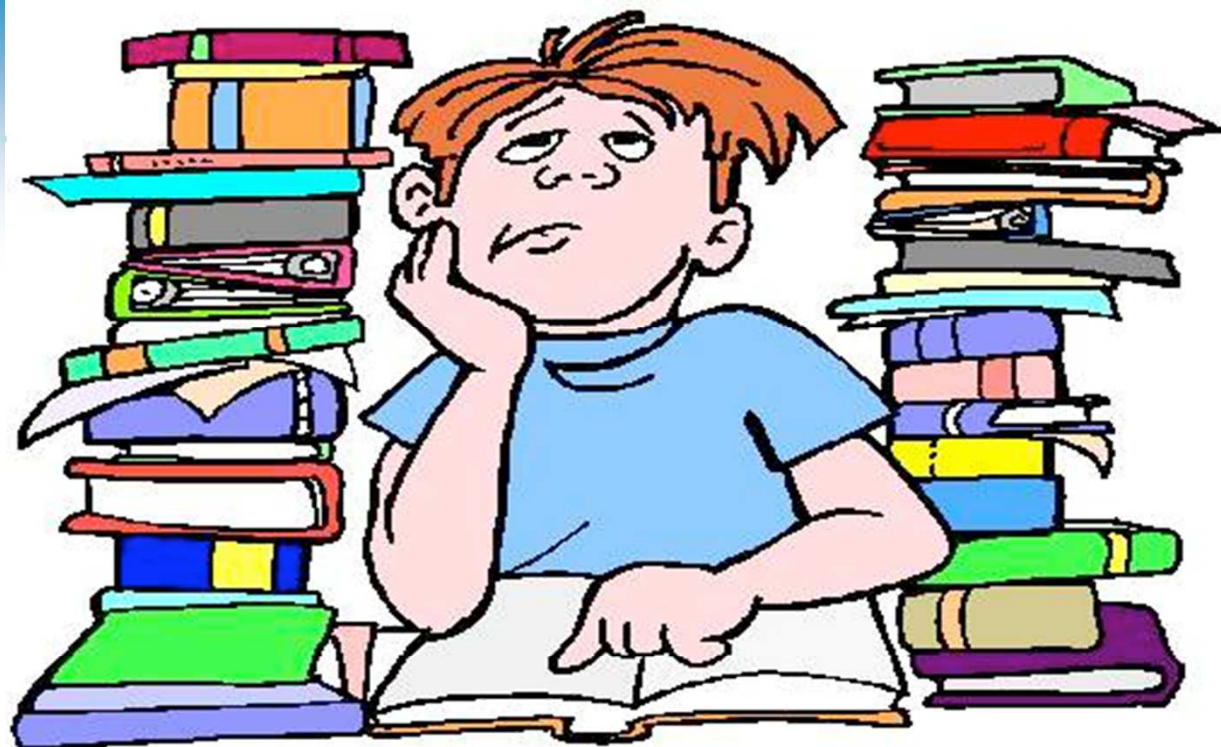
Mealy FSM

Homework

- Design a FSM that recognize pattern **110**
 - Moore and Mealy
 - Binary, one-hot, Johnson (001, 011, 111,...) encoder

Summarize

- 
- ✓ State machine
 - ✓ Melly & Moore machine
 - ✓ Design state machine



Reference

- Chapter 8, “Fundamentals of Digital Logic”,
Stephen Brown, Zvonko Vranesic