EN1014 Electronic Engineering

Design of Sequential Logic Circuits

Chamira U. S. Edussooriya
B.Sc.Eng. (Moratuwa), M.A.Sc. (UVic), Ph.D. (UVic)

Department of Electronic and Telecommunication Engineering
University of Moratuwa

Some of the tables and figures included in this presentation have been extracted from
Digital Design: With an Introduction to the Verilog HDL (M. M. Mano and M. D. Ciletti, Prentice Hall, 2012)

Introduction

- In contrast to a combinational logic circuit (fully specified by a truth table), a sequential logic circuit is specified by a state diagram or a state table (transition table).

Introduction

- In contrast to a combinational logic circuit (fully specified by a truth table), a sequential logic circuit is specified by a state diagram or a state table (transition table).
- The first step in the design of a sequential logic circuit is to obtain a state diagram.

Introduction

- In contrast to a combinational logic circuit (fully specified by a truth table), a sequential logic circuit is specified by a state diagram or a state table (transition table).
- The first step in the design of a sequential logic circuit is to obtain a state diagram.
- The design of the circuit consists of
  - choosing the type and number of flip-flops
  - designing the combinational logic circuit corresponding to the flip-flop input equations and the output equations.

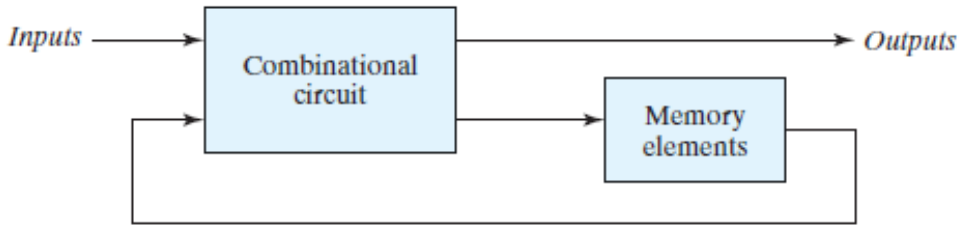

FIGURE 5.1
Block diagram of sequential circuit

Finite State Machines

- A sequential logic circuit can be implemented as a Mealy machine or a Moore machine, which are commonly referred to as finite state machines.

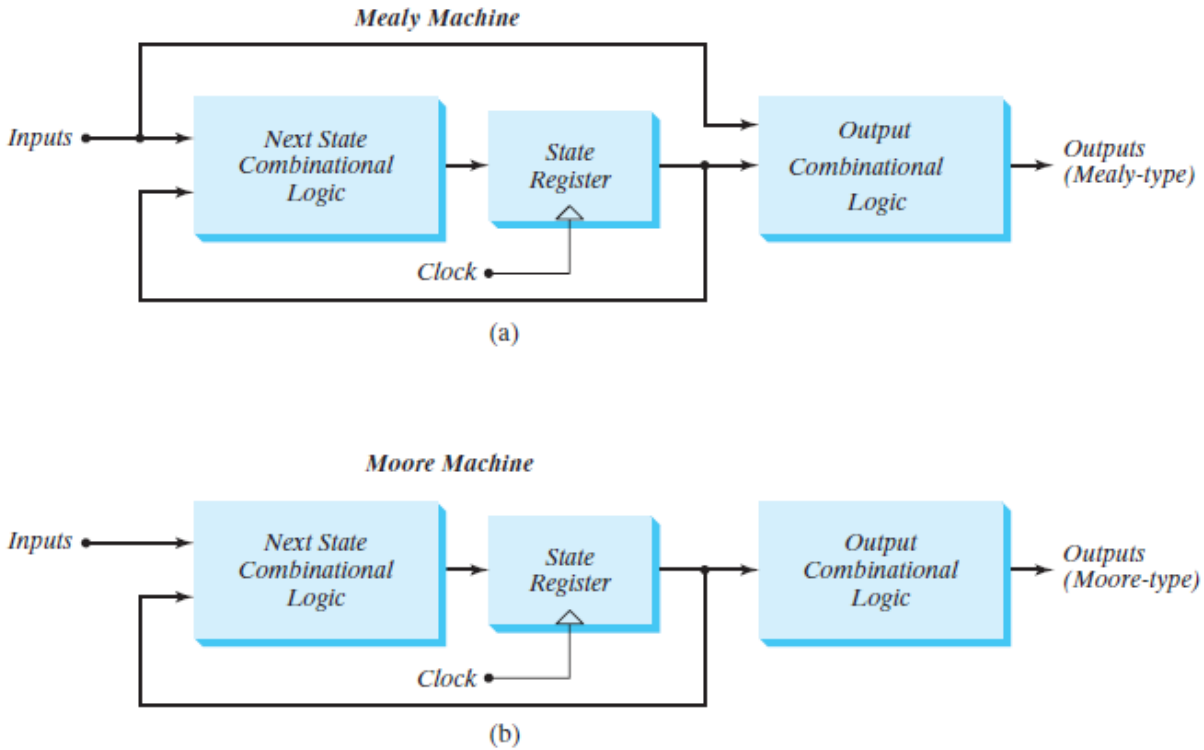

FIGURE 5.21
Block diagrams of Mealy and Moore state machines

Finite State Machines *cont'd*

- The output of a Moore machine depend only on the present state of flip-flops. Therefore, the outputs are synchronized with the clock.

Finite State Machines *cont'd*

- The output of a Moore machine depend only on the present state of flip-flops. Therefore, the outputs are synchronized with the clock.
- The output of a Mealy machine depend on both the inputs and the present state of flip-flops. Therefore, the outputs may change if the inputs change during a clock cycle.

Finite State Machines *cont'd*

- The output of a Moore machine depend only on the present state of flip-flops. Therefore, the outputs are synchronized with the clock.
- The output of a Mealy machine depend on both the inputs and the present state of flip-flops. Therefore, the outputs may change if the inputs change during a clock cycle.
- Examples:
  - Consider a digital circuit that can detect the 111 bit stream without overlaps. Draw the state diagram to design the digital circuit as a (a) Moore machine (b) Mealy machine.

Finite State Machines *cont'd*

- The output of a Moore machine depend only on the present state of flip-flops. Therefore, the outputs are synchronized with the clock.
- The output of a Mealy machine depend on both the inputs and the present state of flip-flops. Therefore, the outputs may change if the inputs change during a clock cycle.
- Examples:
  - Consider a digital circuit that can detect the 111 bit stream without overlaps. Draw the state diagram to design the digital circuit as a (a) Moore machine (b) Mealy machine.
  - Consider a digital circuit that can detect the 1101 bit stream including overlaps. Draw the state diagram to design the digital circuit as a (a) Moore machine (b) Mealy machine.

Design of Sequential Logic Circuits

- The design steps of a sequential logic circuit will be explained with the following examples:
  - Design a digital circuit that can detect the 1101 bit stream including overlaps as a Moore machine using D flip-flops.

Design of Sequential Logic Circuits

- The design steps of a sequential logic circuit will be explained with the following examples:
  - Design a digital circuit that can detect the 1101 bit stream including overlaps as a Moore machine using *D* flip-flops.

Table 5.9
Three Possible Binary State Assignments

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

Design of Sequential Logic Circuits

- The design steps of a sequential logic circuit will be explained with the following examples:
  - Design a digital circuit that can detect the 1101 bit stream including overlaps as a Moore machine using *D* flip-flops.
  - Design a digital circuit that can detect the 111 bit stream without overlaps as a Mealy machine using *JK* flip-flops.

State Reduction

- The number of states in a state diagram may be reduced.

State Reduction

- The number of states in a state diagram may be reduced.
- Consider the following state diagram.

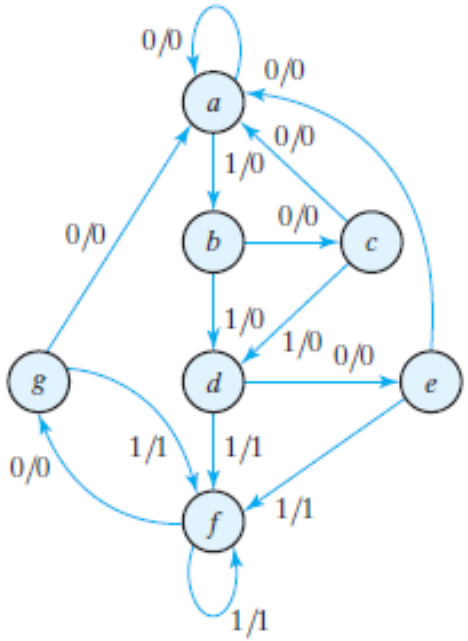

Table 5.6
State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

State Reduction

- The number of states in a state diagram may be reduced.
- Consider the following state diagram.

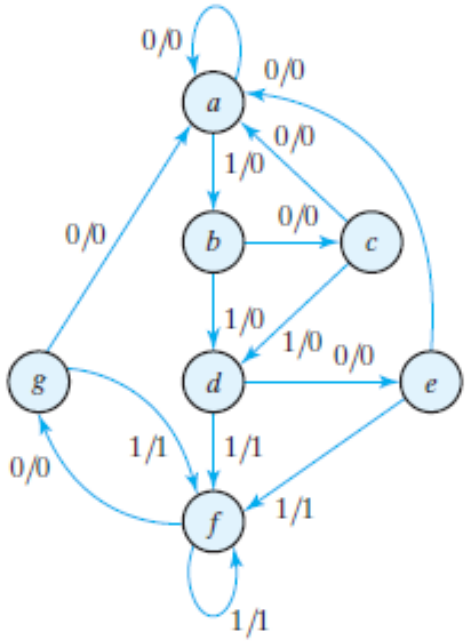

Table 5.6
State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

- Two states are said to be equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state.

State Reduction *cont'd*

- When two states are equivalent, one of them can be removed without altering the input-output relationships.

State Reduction *cont'd*

- When two states are equivalent, one of them can be removed without altering the input-output relationships.

Table 5.7
Reducing the State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

State Reduction *cont'd*

- When two states are equivalent, one of them can be removed without altering the input-output relationships.

Table 5.7
Reducing the State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Table 5.8
Reduced State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1