

Chapter 5

Synchronous Sequential Logic

5.1

Sequential Circuits

- Consist of a combinational circuit to which storage elements are connected to form a feedback path
- State – the state of the memory devices now, also called **current state**
- Next states and outputs are functions of inputs and present states of storage elements

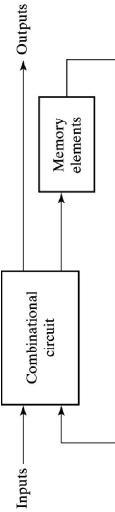


Fig. 5-1. Block Diagram of Sequential Circuit

5.3

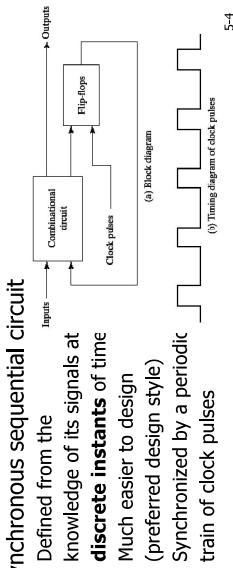
Outline

- Sequential Circuits
- Latches
- Flip-Flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
- Design Procedure

5.2

Two Types of Sequential Circuits

- Asynchronous sequential circuit
 - Depends upon the input signals at **any instant** of time and their change order
 - May have better performance but hard to design
- Synchronous sequential circuit
 - Defined from the knowledge of its signals at **discrete instants** of time
 - Much easier to design (preferred design style)
 - Synchronized by a periodic train of clock pulses



5.4

Memory Elements

- Allow sequential logic design
- Latch — a level-sensitive memory element
 - SR latches
 - D latches
- Flip-Flop — an edge-triggered memory element
 - Master-slave flip-flop
 - Edge-triggered flip-flop
- RAM and ROM — a mass memory element
- Discussed in Chapter 7

5.5

Latches

- The most basic types of flip-flops operate with signal levels
- The basic circuits from which all flip-flops are constructed
- Useful for storing binary information and for the design of asynchronous sequential circuits
 - Not practical for use in synchronous sequential circuits
 - Avoid to use latches as possible in synchronous sequential circuits to avoid design problems

5.7

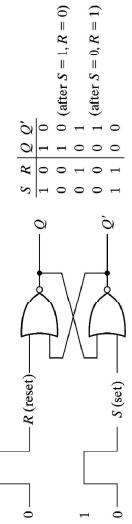
Outline

- Sequential Circuits
- Latches
- Flip-Flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
- Design Procedure

5.6

SR Latch

- A circuit with two cross-coupled NOR gates or two cross-coupled NAND gates
- Two useful states:
 - $S=1, R=0 \rightarrow$ set state (Q will become to 1)
 - $S=0, R=1 \rightarrow$ reset state (Q will become to 0)
- When $S=0$ and $R=0 \rightarrow$ keep the current value



5.8

Undefined State in SR Latch

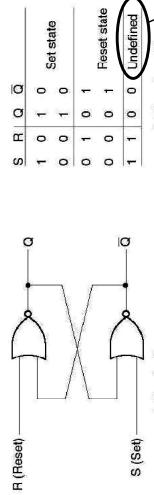


Fig. 4-4 SR Latch with NOR Gates

Should be very careful for this case

(b) Function table

S	R	Q	\bar{Q}
1	0	1	0
0	1	0	1
0	0	1	0
1	1	0	0

5.9

Fig. 4-5 Logic Simulation of SR Latch Behavior

SR Latch with Control Input

- Add an additional control input to determine when the state of the latch can be changed
- $C=0$: S and R are disabled (no change at outputs)
- $C=1$: S and R are active-high

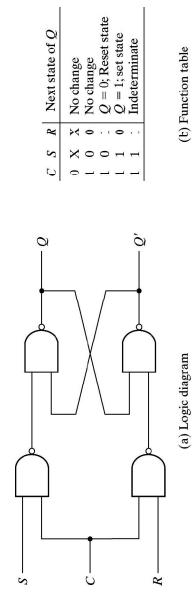


Fig. 5-5 SR Latch with Control Input

5.9

(t) Function table

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q=0$; Reset state
1	1	0	$Q=1$; Set state
1	1	1	Indeterminate

5.11

SR Latch with NAND Gates

- The SR latches constructed with two cross-coupled NAND gates are **active-low**

$S=1, R=0 \rightarrow$ reset state (Q will become to 0)

$S=0, R=1 \rightarrow$ set state (Q will become to 1)

$S=1, R=1 \rightarrow$ unchanged

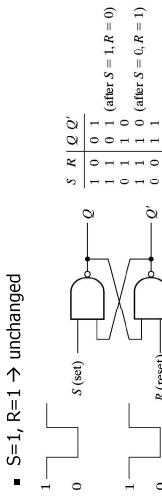


Fig. 5-4 SR Latch with NAND Gates

5.10

- D latch has only two inputs: D(data) and C(control)

▪ Use the value of D to set the output value

▪ Eliminate the indeterminate state in the SR latches

▪ The D input goes directly to the S input and its complement is applied to the R input



Fig. 5-6 D Latch

5.12

D Latch

- D latch has only two inputs: D(data) and C(control)

▪ Use the value of D to set the output value

▪ Eliminate the indeterminate state in the SR latches

▪ The D input goes directly to the S input and its complement is applied to the R input

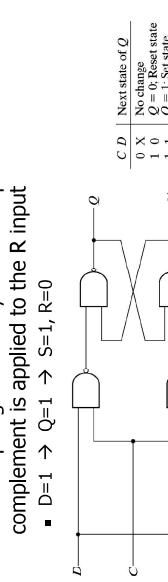


Fig. 5-6 D Latch

5.12

Graphic Symbols for Latches

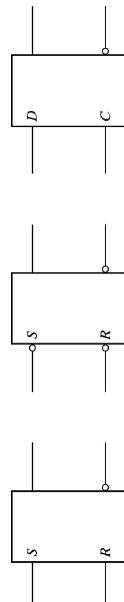


Fig. 5-7 Graphic Symbols for Latches

5-13

Flip-Flops

- The state of a latch or flip-flop is switched by a change in the control input
- This momentary change is called a **trigger**
- Latch: level-sensitive
- Flip-Flop: edge-triggered

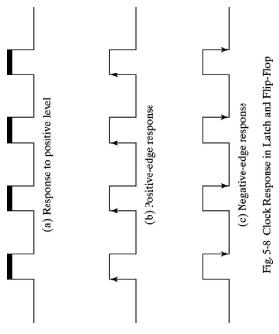


Fig. 5-8 Click Response in Latch and Flip-Flop

5-15

Latch vs. Flip-Flop

- Latch:**
 - Change stored value under specific status of the control signals
 - Transparent for input signals when control signal is "on"
 - May cause combinational feedback loop and extra changes at the output
- Flip-Flop:**
 - Can only change stored value by a momentary switch in value of the control signals
 - Cannot "see" the change of its output in the same clock pulse
 - Encounter fewer problems than using latches

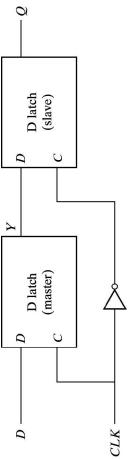


5-16

Outline

- Sequential Circuits
- Latches
- Flip-Flops**
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
- Design Procedure

Master-Slave D Flip-Flop



- Constructed with two D latches and an inverter
 - The first latch (master) is enabled when $CLK=1$
 - It reads the input changes but stops before the second one
 - The second latch (slave) is enabled when $CLK=0$
 - Close the first latch to isolate the input changes
 - Deliver the final value at the moment just before CLK changes
 - The circuit samples the D input and changes its output Q only at the **negative-edge** of the controlling clock
- 5-17

Setup & Hold Times

- The response time of a flip-flop to input changes must be taken into consideration
 - **Setup Time:** The length of time that data must stabilize before the clock transition
 - The maximum data path is used to determine if the setup time is met
 - Hold Time: The length of time that data must remain stable at the input pin after the active clock transition
 - The minimum data path is used to determine if hold time is met
- 5-19

Edge-Triggered D Flip-Flop

- If only SR latches are available, three latches are required
- Two latches are used for locking the two inputs (CLK & D)
- The final latch provides the output of the flip-flop

Fig.5-10 D-Type Positive-Edge-Triggered Flip-Flop



(a) Positive-edge
(b) Negative-edge

Fig.5-11 Graphic Symbol for Edge-Triggered D Flip-Flop

5-18

Setup & Hold Times

- **Timing Diagram**
 - **Valid Data Transition**
- 5-20

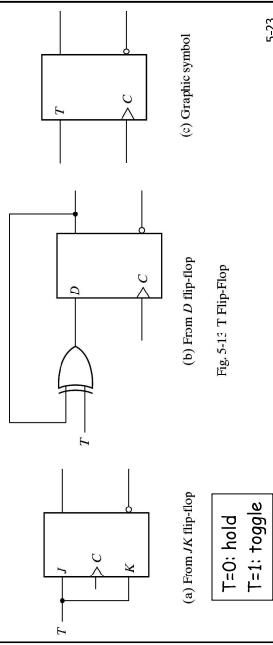
Other Flip-Flops

- The most economical and efficient flip-flop is the edge-triggered D flip-flop
- It requires the smallest number of gates
- Other types of flip-flops can be constructed by using the D flip-flop and external logic
 - JK flip-flop
 - T flip-flops
 - Three major operations that can be performed with a flip-flop:
 - Set it to 1
 - Reset it to 0
 - Complement its output

5-21

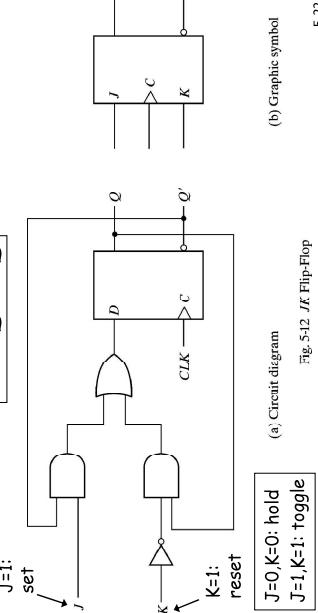
Edge-Triggered T Flip-Flop

$$D = T \oplus Q = TQ' + T'Q$$



Edge-Triggered JK Flip-Flop

$$D = JQ + K'Q$$



Characteristic Tables

- Define the logical properties in tabular form

JK flip-flop		T flip-flop	
J	K	Q(t+1)	Q(t+1)
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q(t)	Complement

D flip-flop		T flip-flop	
D	Q(t+1)	T	Q(t+1)
0	0	Reset	0
1	Set	1	Q(t) Q(t)' Complement

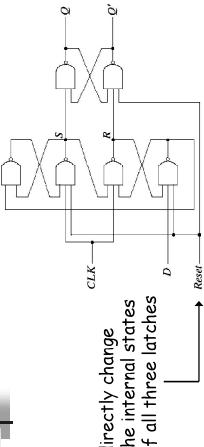
5-24

Characteristic Equations

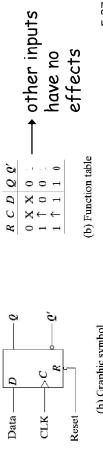
- Algebraically describe the next state
- Can be derived from characteristic tables
- D flip-flop:
$$Q(t+1) = D$$
- JK flip-flop:
$$Q(t+1) = JQ' + K'Q$$
- T flip-flop:
$$Q(t+1) = T \oplus Q = TQ + T'Q$$

5-25

D F/F with Asynchronous Reset



(a) Circuit diagram



5-27

Direct Inputs

- Force the flip-flop to a particular state immediately
 - Independent of clock signal
 - Have higher priority than any other inputs
 - Useful to bring all flip-flops from unknown into known state while power up
 - The input that sets the flip-flop to 1 is called **preset** or **direct set**
 - The input that clears the flip-flop to 0 is called **clear** or **direct reset**
 - Also called **asynchronous** set/reset

5-26

Outline

- Sequential Circuits
 - Latches
 - Flip-Flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
 - Design Procedure

5-28

Sequential Circuit Analysis

- The behavior of a clocked sequential circuit is determined from
 - The inputs
 - The outputs
 - The state of its flip-flops
 - The outputs and the next state are both a function of the inputs and the present state
 - To analyze a sequential circuit, we can use
 - State equations
 - State table
 - State diagram
 - Flip-Flop input equations
- 5-29

State Table

- Enumerate the time sequence of inputs, outputs, and flip-flop states
 - Also called transition table
 - Similar to list the truth table of state equations
 - Consist of four sections
 - Present state, input, next state, and output
 - A sequential circuit with m flip-flops and n inputs need 2^{m+n} rows in the state table
- 5-29

5-31

State Equations

- Specify the next state as a function of the present state and inputs
 - Also called transition equation
 - Analyze the combinational part directly
 - EX:

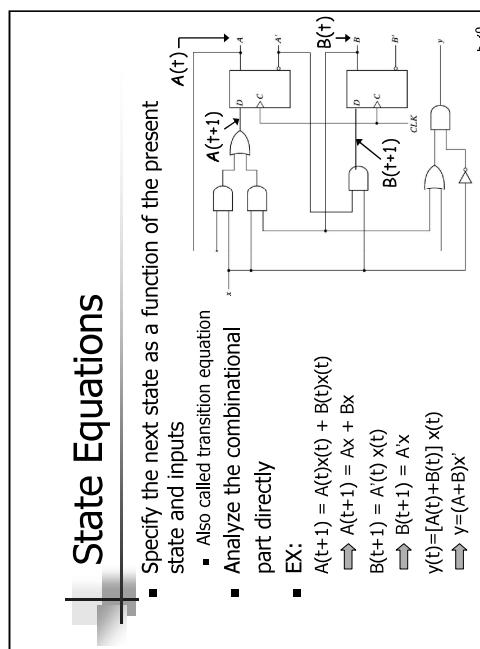
$$\begin{aligned} A(t+1) &= A(t)x(t) + B(t)x(t) \\ \Rightarrow A(t+1) &= Ax + Bx \\ B(t+1) &= A(t)x(t) \\ \Rightarrow B(t+1) &= Ax \\ Y(t) &= [A(t)+B(t)]x(t) \\ \Rightarrow Y(t) &= (A+B)x \end{aligned}$$
- 5-32

Second Form of State Table

- The state table has only three section: present state, next state, and output
- The input conditions are enumerated under next state and output sections

Present State	Next State			Output	
	A	B	X=0	X=1	
0	0	0	0	1	0
0	1	0	0	1	1
1	0	1	1	0	0
1	1	0	0	1	0

5-32

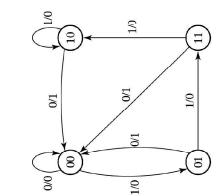


State Diagram

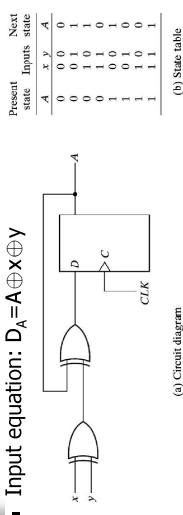
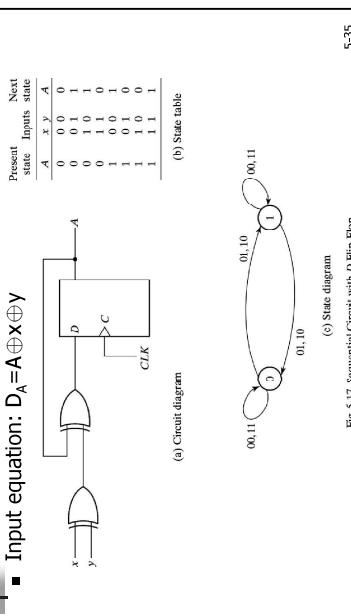
- Graphically represent the information in a state table
 - Circle: a state (with its state value inside)
 - Directed lines: state transitions (with inputs/outputs above)
- Ex: starting from state 00
 - If the input is 0, it stays at state 00
 - If the input is 1, it goes to state 01
 - If the input is 0, with output=0
 - If the input is 1, with output=0
- The state table is easier to derive from a given logic diagram and state equations
- The state diagram is suitable for human interpretation

Fig. 5.16 State Diagram of the Circuit of Fig. 5.15

5-33



Analysis with D Flip-Flop



Flip-Flop Input Equations

- To draw the logic diagram of a sequential circuit, we need
 - The type of flip-flops
 - A list of Boolean expressions of the combinational circuits
 - The Boolean functions for the circuit that generates external outputs is called output equations
 - The Boolean functions for the circuit that generates the inputs to flip-flops is flip-flop input equations
 - Sometimes called excitation equations
 - The flip-flop input equations provide a convenient form for specifying the logic diagram of a sequential circuit
- Ex: (Fig. 5.15)

Input:	$D_A = Ax + Bx$	Output:	$y = (A+B)x'$
	$D_B = Ax$		

5-34

Analysis with Other Flip-Flops

- The sequential circuit using other flip-flops such as JK or T type can be analyzed as follows
 - Determine the flip-flop input equations in terms of the present state and input variables
 - List the binary values of each input equation
 - Use the corresponding flip-flop characteristic table to determine the next state values in the state table

5-36

Analysis with JK Flip-Flops (1/2)

Step 1: input equations

$$J_A = B \quad K_A = Bx' \quad J_B = x' \quad K_B = A \oplus x'$$

Step 2: state equations

$$\begin{aligned} A(t+1) &= JA + KA \\ &= BA' + (Bx')A \\ &= AB + AB' + Ax \end{aligned}$$

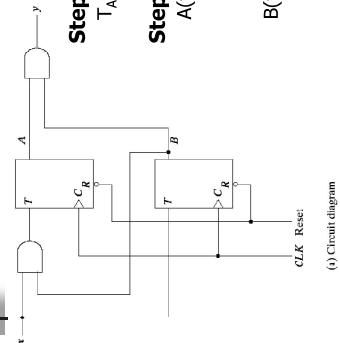
$$B(t+1) = JB + KB$$

$$\begin{aligned} &= x'B + (A \oplus x)B \\ &= BX + ABx + ABx' \end{aligned}$$

Fig. 5-18 Sequential Circuit with JK Flip-Flop

5-37

Analysis with T Flip-Flops (1/2)



Step 1: input equations

$$T_A = BX \quad T_B = x \quad y = AB$$

Step 2: state equations

$$\begin{aligned} A(t+1) &= TA + TA' \\ &= (BX)A + (Bx)A' \\ &= AB' + Ax' + A'BX \\ B(t+1) &= TB + TB' \\ &= xB + xB' \\ &= x \oplus B \end{aligned}$$

5-39

CLK Reset:

(i) Circuit diagram

Analysis with JK Flip-Flops (2/2)

Step 3: state table

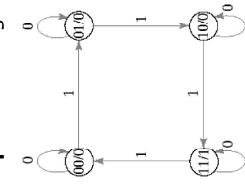
Present state	Input	Flip-Flop Inputs				Output
		A	B	J _A	J _B	
0 0	0	0	1	0	0	0
0 0	1	0	0	0	0	1
0 1	0	1	1	1	1	0
0 1	1	1	0	1	0	0
1 0	0	1	1	0	0	1
1 0	1	1	0	0	0	1
1 1	0	0	0	1	1	1
1 1	1	1	1	1	0	0

Fig. 5-19 State Diagram of the Circuit of Fig. 5-18

5-38

Analysis with T Flip-Flops (2/2)

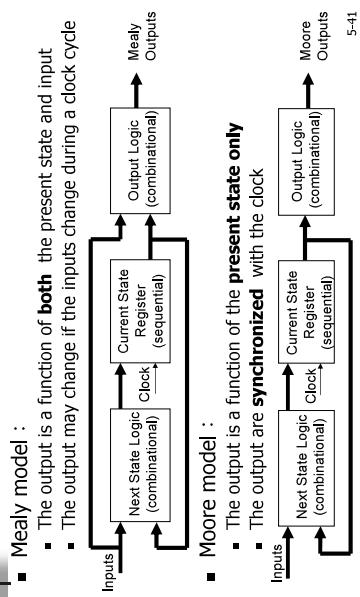
Step 4: state diagram



(b) State diagram

5-40

Mealy and Moore Model



State Reduction

- Reducing the number of states in a state table, while keeping the external input-output requirements unchanged
- Example:
 - Total 7 states
 - A sequence as follows

state	a	a	b	c	d	e	f	g	a
input	0	1	0	1	0	1	0	1	0
output	0	0	0	0	1	1	0	1	0

Fig. 5-22 State Diagram 5-43

State Reduction Rules

- Two states are said to be equivalent if, for every possible inputs, they give exactly the same output and have equivalent next state

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

delete state g and replaced with state e

5-44

Outline

- Sequential Circuits
- Latches
- Flip-Flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
- Design Procedure

Further State Reduction

- After the first reduction, we can see that state d and state f will have the same output and next state for both $x=0$ and $x=1$

Further reduce one state

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

delete state f and replaced with state d

5-15

Implication Chart Method (1/3)

- Step 1: build the implication chart

Present State	Next State		Output	Present State		Next State		Output
	X = 0	1		X = 0	X = 1	X = 0	X = 1	
a	d	c	0	0	1	1	0	0
b	f	h	0	0	1	1	0	0
c	e	d	1	1	0	0	1	1
d	a	e	0	0	1	1	0	0
e	c	c	1	1	0	0	1	1
f	f	b	1	1	0	0	1	1
g	b	h	0	0	1	1	0	0
h	c	g	1	1	0	0	1	1

5-47

*For details, see "Fundamentals of Logic Design", 4th Ed., by C. H. Roth, Jr.

Implication Chart Method (2/3)

- Step 2: delete the node with unsatisfied conditions

b	X	X	X	X	X	X	X	X
c	X	X	X	X	X	X	X	X
d	X	X	X	X	X	X	X	X
e	X	X	X	X	X	X	X	X
f	X	X	X	X	X	X	X	X
g	X	X	X	X	X	X	X	X
h	X	X	X	X	X	X	X	X

5-48

Reduced State Diagram

- After reduction, the circuit has only 5 states with same input/output requirements

Original output sequence:

state	a	a	b	c	d	e	f	g	a
input	0	1	0	1	0	1	0	1	0
output	0	0	0	0	1	0	1	0	0

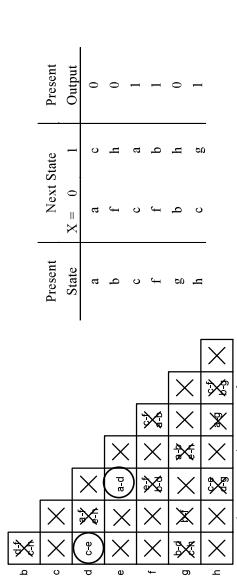
Reduced output sequence:

state	a	a	b	c	d	e	d	e	a
input	0	1	0	1	1	0	1	0	0
output	0	0	0	0	1	1	0	1	0

Fig. 5-23 Reduced State Diagram

Implication Chart Method (3/3)

- Step 3: repeat Step 2 until equivalent states found



Popular State Assignments

State	Assignment 1			Assignment 2			Assignment 3		
	Binary	Gray code	One-hot	Binary	Gray code	One-hot	Binary	Gray code	One-hot
a	000	000	0001	000	000	0001	000	000	0001
b	001	001	0010	001	001	0010	001	001	0010
c	010	010	0100	010	010	0100	010	010	0100
d	011	011	0110	100	100	1000	110	110	1000
e	100	100	1000						

5-51

State Assignment

- Assign **coded binary values** to the states for physical implementation
- For a circuit with m states, the codes must contain n bits where $2^n \geq m$
- Unused states are treated as don't care conditions during the design
 - Don't cares can help to obtain a simpler circuit
- There are many possible state assignments
 - Have large impacts on the final circuit size

Assignment:			
a = 000	d = 011		
b = 001	e = 100		
c = 010			

Outline

- Sequential Circuits
- Latches
- Flip-Flops
- Analysis of Clocked Sequential Circuits
- State Reduction and Assignment
- Design Procedure

5-52

State	Present State	Assignment 1	Assignment 2	Assignment 3
a	000	000	000	0001
b	001	001	001	0010
c	010	010	010	0100
d	011	011	011	0110
e	100	100	100	1000

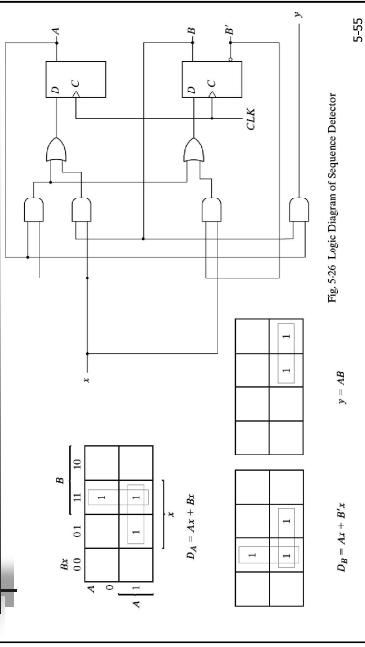
5-51

Design Procedure

- Design procedure of synchronous sequential circuits:
 - Derive a state diagram for the circuit from specifications
 - Reduce the number of states if necessary
 - Assign binary values to the states
 - Obtain the binary-coded state table
 - Choose the type of flip-flop to be used
 - Derive the simplified flip-flop input equations and output equations
 - Draw the logic diagram
 - Step 4 to 7 can be automated
 - Use HDL synthesis tools

5-53

Synthesis Using D Flip-Flops



Synthesis Using D Flip-Flops

- Ex: design a circuit that detects 3 or more consecutive 1's at inputs
-
- $A(t+1) = D(A, B, x) = \sum (3, 5, 7)$
 $B(t+1) = D_B(A, B, x) = \sum (1, 5, 7)$
 $y(A, B, x) = \sum (6, 7)$

5-54

Excitation Tables

- Record the flip-flop input conditions that will cause the required transition in STG
- Equal to next state equations for D flip-flop
- For JK flip-flop:
 - $J=0, K=X$: no change ($JK=00$) or set to zero ($JK=01$)
 - $J=1, K=X$: toggle ($JK=11$) or set to one ($JK=10$)
 - $J=X, K=1$: toggle ($JK=11$) or set to zero ($JK=01$)
 - $J=X, K=0$: no change ($JK=00$) or set to one ($JK=10$)

JK	$Q(t)$	$Q(t+1)$	J	K	F/F	$Q(t)$	$Q(t+1)$	T	F/F
0	0	0	0	X		0	0	0	
0	0	0	0	0		0	1	1	
0	0	1	0	1		1	0	1	
0	1	0	0	0		1	1	0	
0	1	1	0	0		1	1	0	
1	0	0	0	0					
1	0	1	1	1					
1	1	0	0	1					
1	1	1	1	1					

5-56

Synthesis Using JK Flip-Flops

- Derive the state table with the excitation inputs
- Other design procedures are the same

Present State	Input	Next State		Flip-Flop Inputs			
		A	B	J _A	K _A	J _B	K _B
0 0	0	0	0	0	X	0	X
0 0	1	0	1	0	X	1	X
0 1	0	1	0	1	X	1	X
0 1	1	0	1	0	X	X	0
1 0	0	1	0	X	0	0	X
1 0	1	1	1	X	0	1	X
1 1	0	1	1	1	X	0	X
1 1	1	0	0	X	1	X	1

5-57

Synthesis Using T Flip-Flops

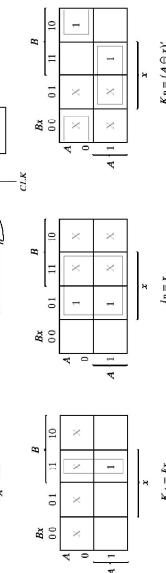
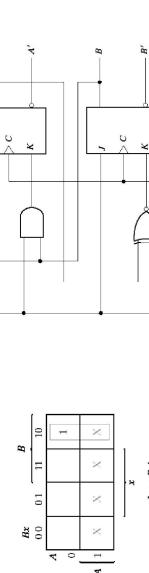
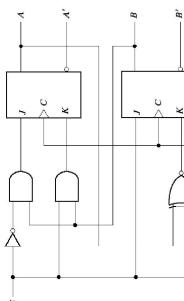
- Derive the state table with the excitation inputs
- Other design procedures are the same

3-bit binary counter	Present State			Next State			Flip-Flop Inputs		
	A ₂	A ₁	A ₀	A ₂	A ₁	A ₀	T _{A2}	T _{A1}	T _{A0}
000	0	0	0	0	0	1	0	0	1
001	0	0	1	0	1	0	0	1	1
010	0	1	0	1	0	1	1	0	0
011	0	1	1	0	0	0	0	1	1
100	1	0	0	1	0	1	0	1	0
101	1	0	1	1	0	1	1	0	1
110	1	1	0	0	1	0	0	1	1
111	1	1	1	1	1	0	0	0	1

5-59

Synthesis Using JK Flip-Flops

$$J_A = R_x \quad K_A = R_x'$$



5-58

Synthesis Using T Flip-Flops

$$T_{A2} = A_1 \quad T_{A1} = A_0 \quad T_{A0} = 1$$

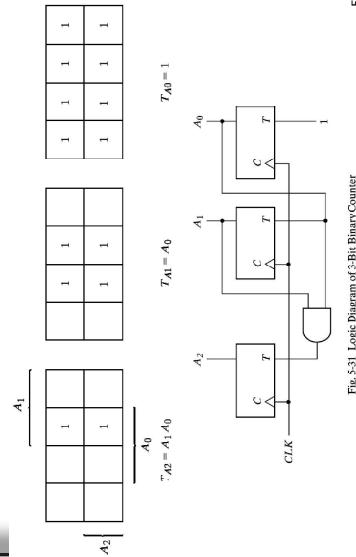


Fig.5-31 Logic Diagram of 3-Bit BinaryCounter 5-60