DIGITAL LOGIC(H)

Chapter 5 part2:
Synchronous Sequential Logic

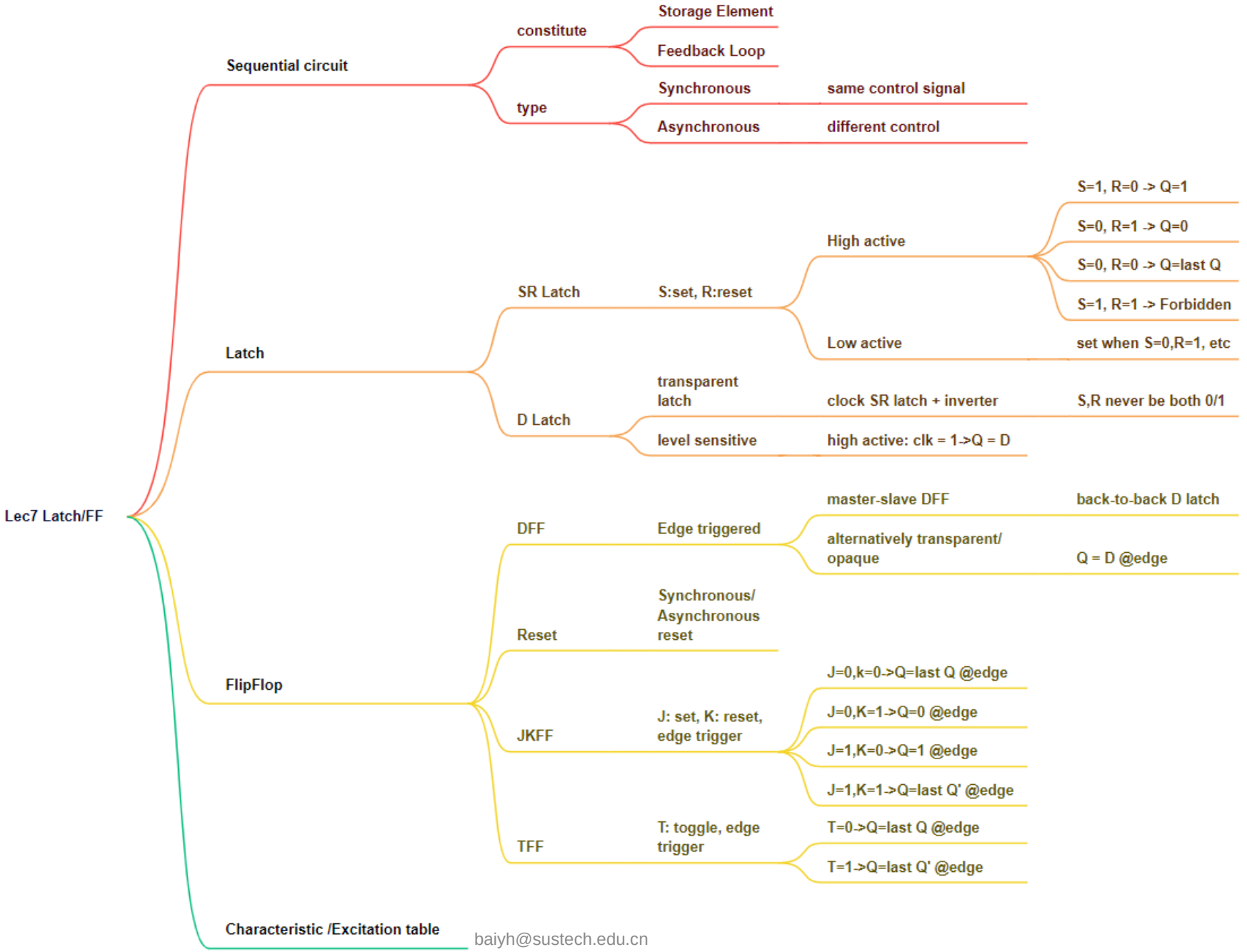
2023 Fall

baiyh@sustech.edu.cn

Today's Agenda

- Recap
- Context
  - Finite State Machine
  - Analysis of Clocked Sequential Circuit
  - Designing Clocked Sequential Circuits
- Reading: Textbook, Chapter 5.5-5.8

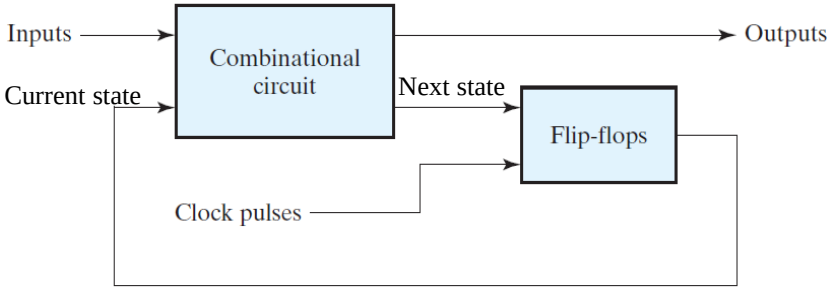
Recap

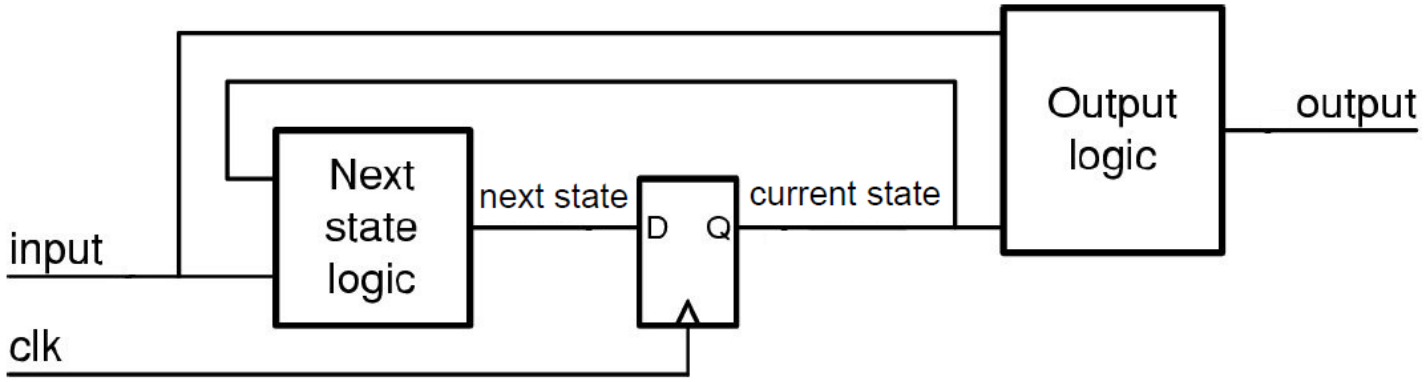


baiyh@sustech.edu.cn

Synchronous Sequential Logic

- State register
  - FlipFlop
    - Stores current state
    - Loads next state at clock edge
- Combinational circuit
  - (inputs, current state) => (outputs, next state)
  - Can be divided into two parts:
    - Next state logic → Computes next state
    - Output logic → Computes outputs





baiyh@sustech.edu.cn

Outline

- **Analysis of Sequential Circuits**
- Finite State Machine
- State Minimization & Encoding
- Design of Sequential Circuits

baiyh@sustech.edu.cn

Ways to describe a digital circuit

- logic diagram (逻辑电路图)
  - graphical representation of a digital circuit that uses standard symbols
- K-map (卡诺图)
  - graphical representation of a logic function used to simplify Boolean expressions
- function table (功能表)
  - (truth table), list of all possible input combinations and the corresponding output values for a given logic function
- characteristic equations (特征方程)
  - Describe the behavior of a sequential logic circuit, the next state is defined as a function of the inputs and the present state
- excitation/input equation (激励方程).
  - Defines the part of the circuit that generates the inputs to sequential logic circuit
- state table (状态表)
  - tabular representation of a sequential logic circuit that shows the current state, input, next state, and output for all possible input combinations
- state equation (次态方程)
  - defines the next state of a sequential logic circuit based on the current state and input values
- state diagram (状态图)
  - graphical representation of a sequential logic circuit that shows the states, transitions, and input-output relationships

Analysis Procedure of Clocked Sequential Circuits

1. Derive excitation/input equations for FF inputs
2. Derive state and output equations
   - Substitute the excitation equations into the flip-flop characteristic equations to obtain next state equations.
   - Determine the output equations according current state and input
3. Generate state and output tables
4. Generate state diagram

- Develop timing diagram
- Simulate logic schematic

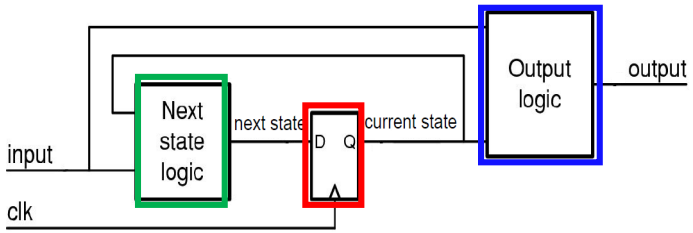
Important: FF's Characteristic equation:

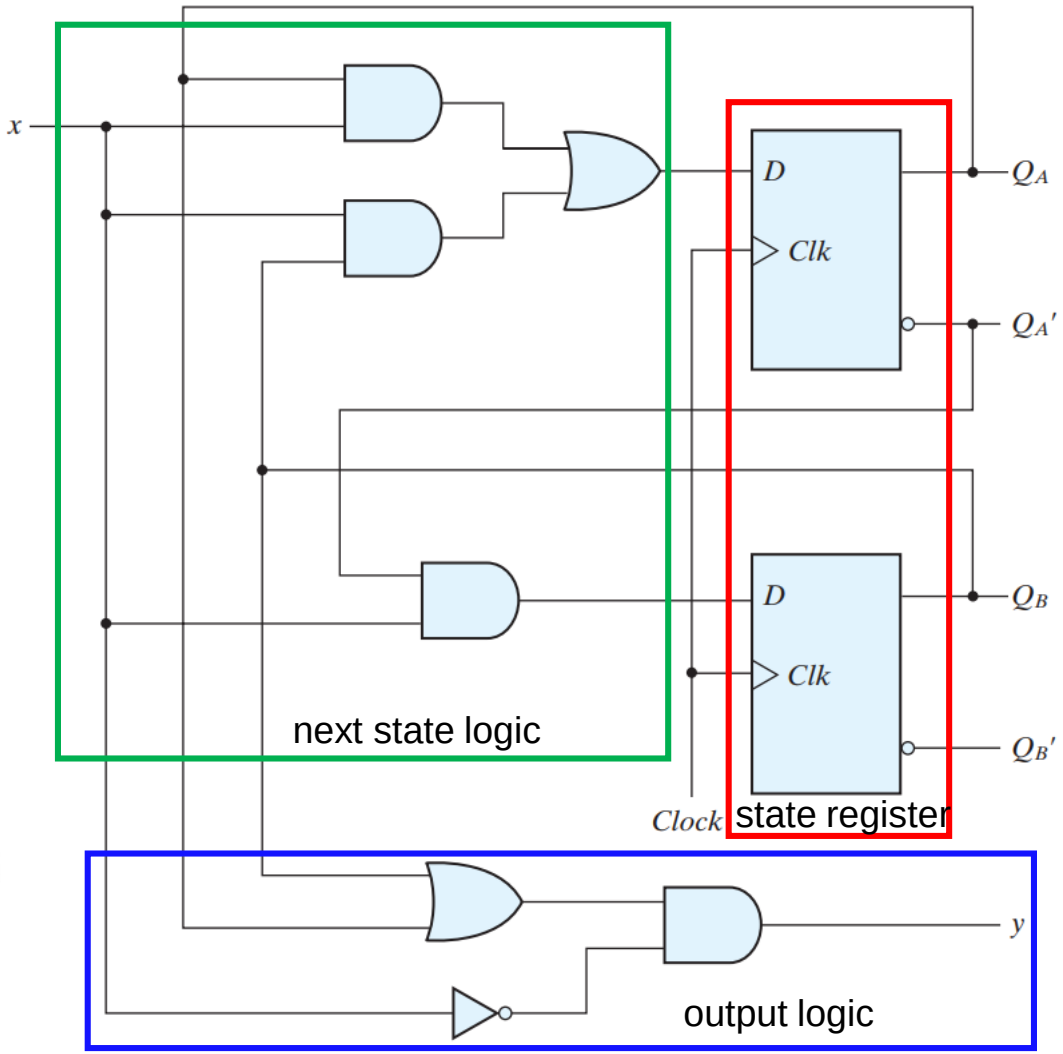
- DFF Q(t+1) = D(t)
- JKFF Q(t+1) = J(t)Q(t)' + K(t)'Q(t)
- TFF Q(t+1) = T(t)'Q(t) + T(t)Q(t)'

baiyh@sustech.edu.cn

Analysis Example 1: A sequential circuit using DFF

1. Derive excitation/input equations for FF inputs
2. Derive state and output equations
3. Generate state and output tables
4. Generate state diagram





baiyh@sustech.edu.cn

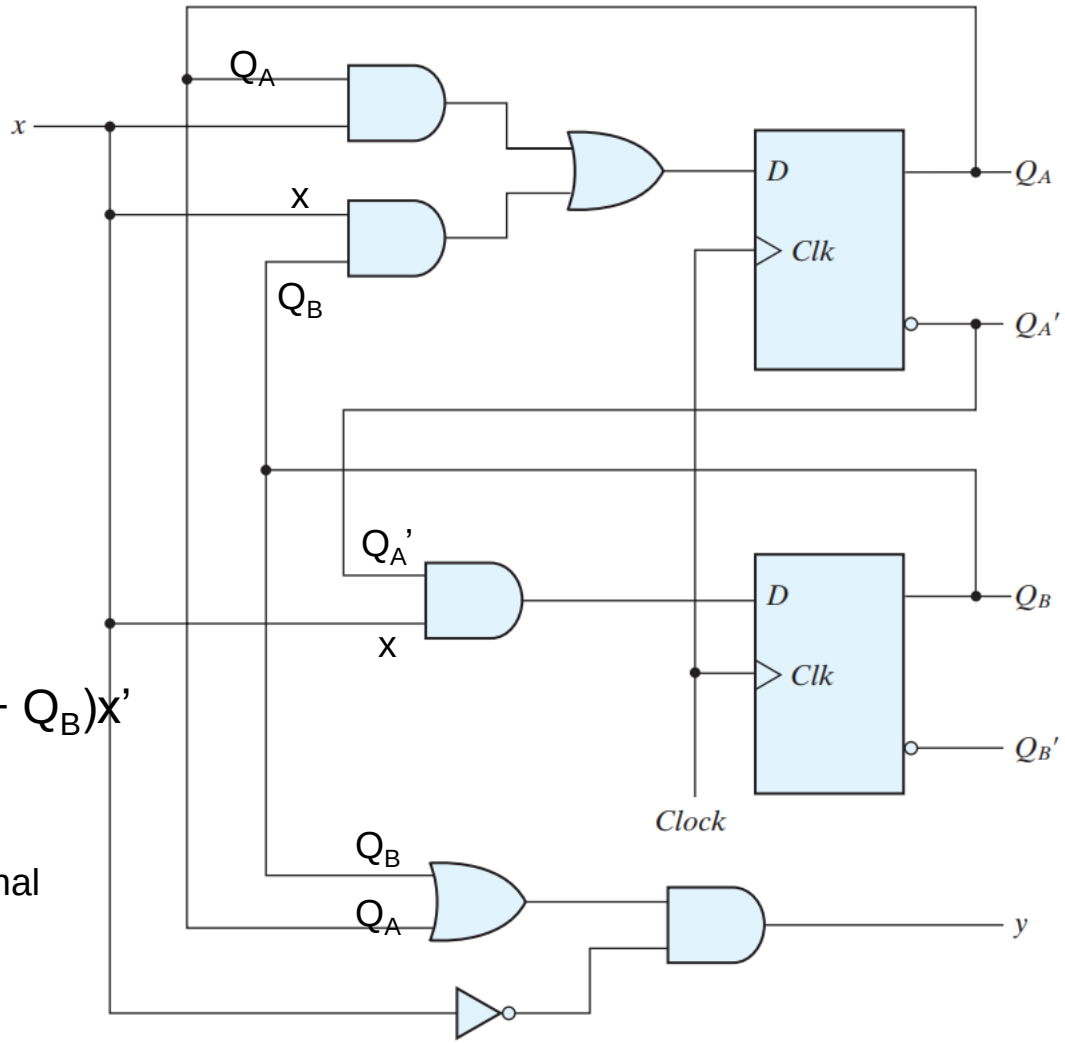
Example 1: A sequential circuit using DFF

1

- Flip-Flop excitation/input equation
  - $D_A = Q_A x + Q_B x$
  - $D_B = Q_A' x$

2

- State equation
  - $Q_A(t+1) = D_A(t)$
    $= Q_A(t)x(t) + Q_B(t)x(t)$
  - $Q_B(t+1) = D_B(t) = Q_A'(t)x(t)$
- Output equation
  - $y(t) = (Q_A(t) + Q_B(t))x'(t)$
  - all signals are labeled by t, thus $y = (Q_A + Q_B)x'$



Next state and output circuit are combinational (regardless of time)

Substituting input equation into DFF's charc. equation Q(t+1) = D(t)

baiyh@sustech.edu.cn

Analysis Example 1: A sequential circuit using DFF

2

$Q_A(t+1) = Q_A(t)x(t) + Q_B(t)x(t)$

$Q_B(t+1) = Q_A'(t)x(t)$

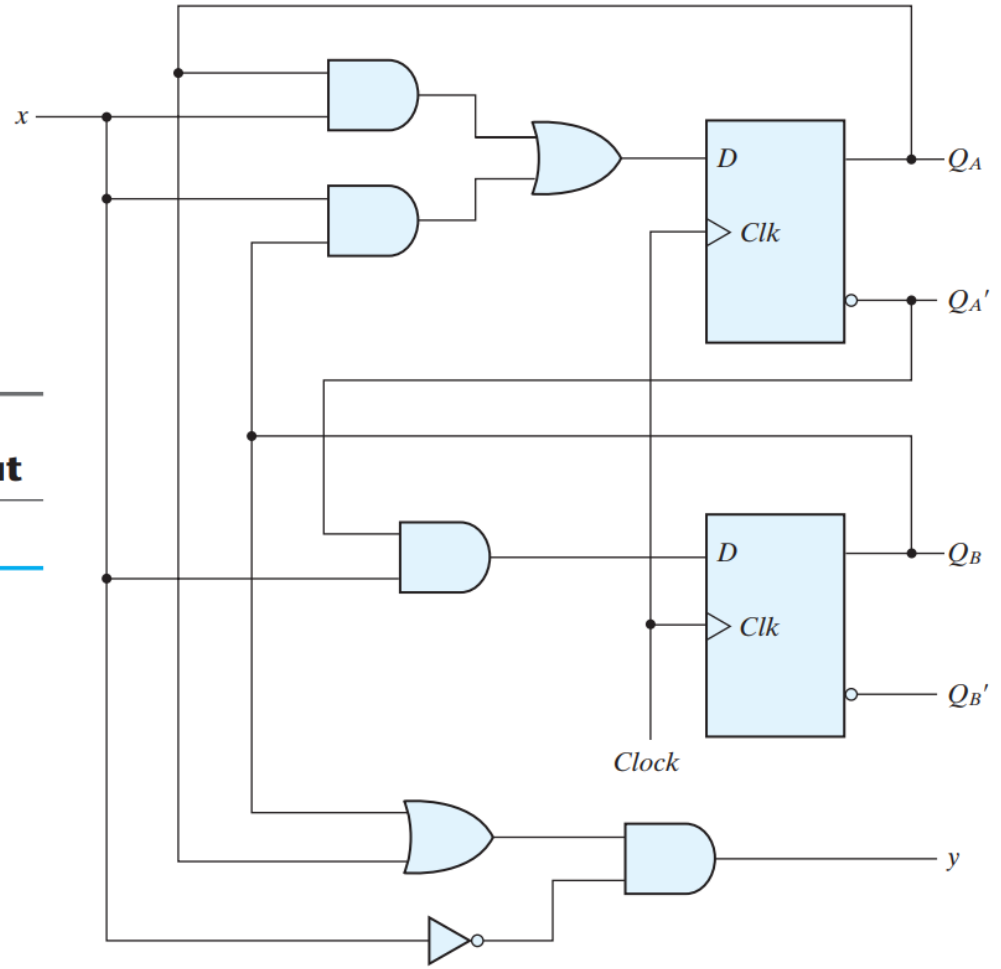
$y = (Q_A + Q_B)x'$

3

- Generate state and output tables

First Form of the State Table

Present State		Input	Next State		Output
Q_A	Q_B	x	Q_A	Q_B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



Analysis Example 1: A sequential circuit using DFF

3 • Generate state and output tables

- Two forms of state table, use the one you prefer:
  - 1st form: state table has 2^{m+n} rows for m FFs and n inputs,
  - 2nd form has three sections, with input in the next state and output column.

First Form of the State Table

Present State		Input	Next State		Output
Q_A	Q_B	x	Q_A	Q_B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

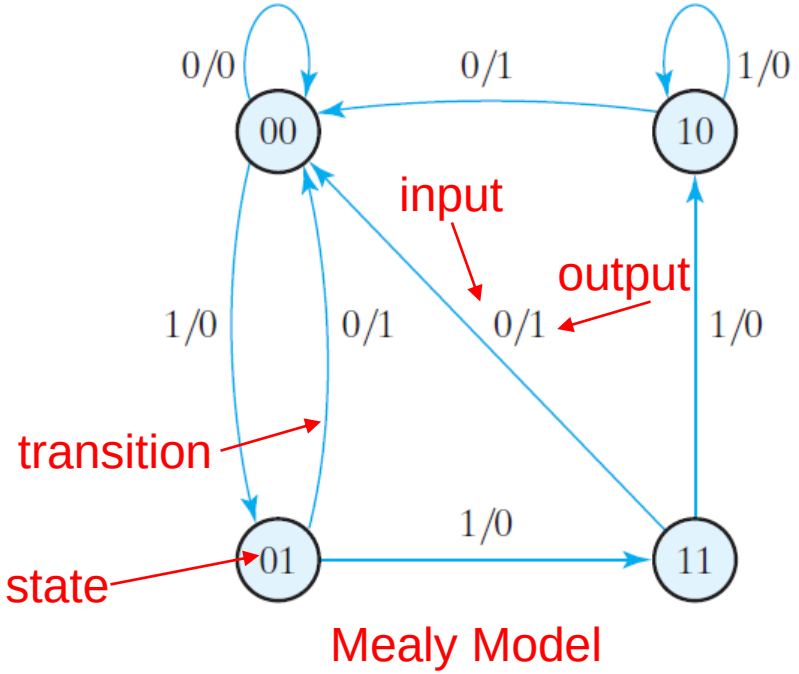
Second Form of the State Table

Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
Q_A	Q_B	Q_A	Q_B	Q_A	Q_B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

baiyh@sustech.edu.cn

Analysis Example 1: A sequential circuit using DFF

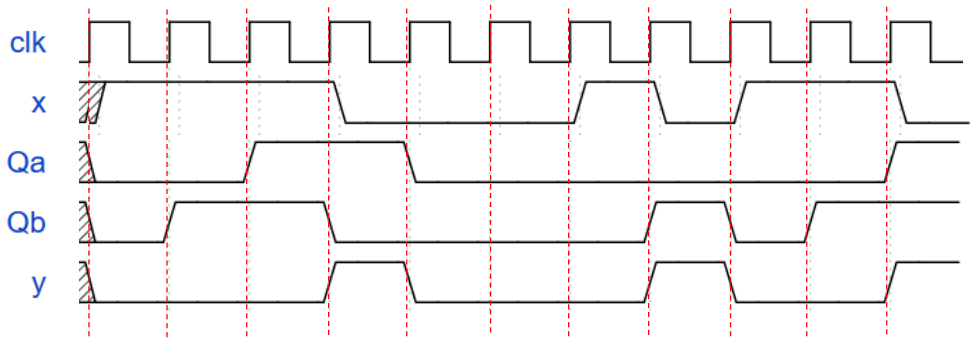
4 • Generate state diagram
  - Each state as a circle.
  - Transitions between states are directed



Mealy Model

Second Form of the State Table

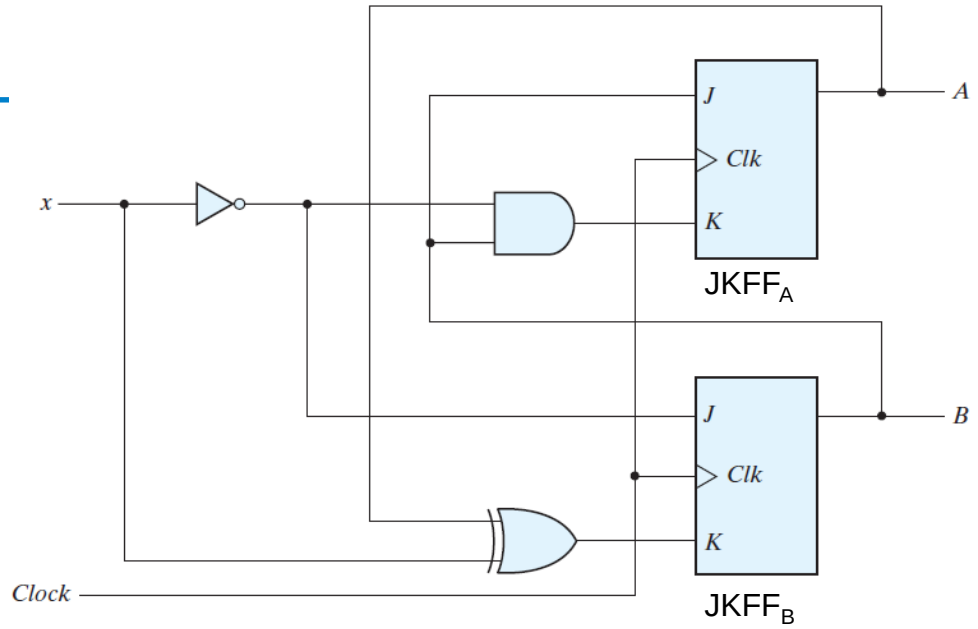
Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
Q_A	Q_B	Q_A	Q_B	Q_A	Q_B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0



Timing diagram
x and initial state of Q_AQ_B are given as an example

baiyh@sustech.edu.cn

Analysis Example 2: A sequential circuit using JKFF



1

- Excitation/Input equation

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

2

- State equation

$A(t+1) = J_A A' + K_A' A = BA' + (Bx')'A$
$B(t+1) = J_B B' + K_B' B = x'B' + (A \oplus x)'B$

(t) is omitted on the RHS
Should be A(t+1) = B(t)A(t)'+(B(t)x(t)')'A(t)
Similarly for B(t+1)

- Output equation
  - No extra output equation since output comes from the output of JKFF(state Q)

Substituting input equation into JKFF's charc. Equation Q(t+1) = J(t)Q(t)' + K(t)'Q(t)

baiyh@sustech.edu.cn

Analysis Example 2: A sequential circuit using JKFF

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

2

- State equation

$A(t+1) = J_A A' + K_A'A = BA' + (Bx')'A$

$B(t+1) = J_B B' + K_B'B = x'B' + (A \oplus x)'B$

3

- State table

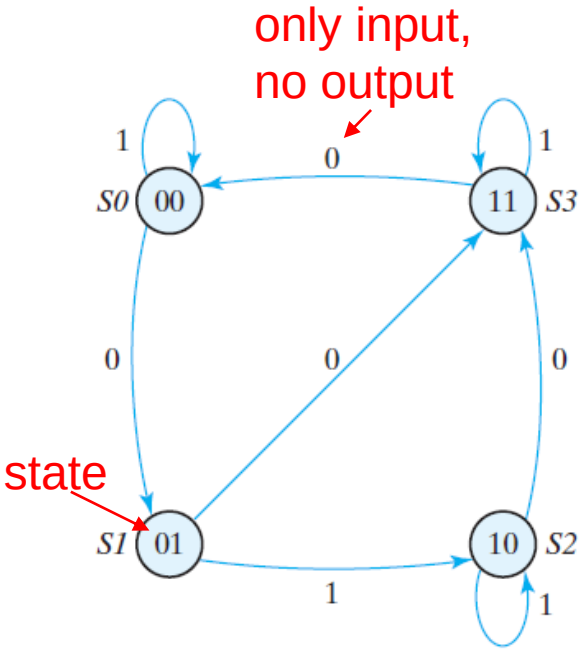
State Table for Sequential Circuit with JK Flip-Flops

Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

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

4

- Generate state diagram



Next state value can be obtained from state equation, or from corresponding FF's inputs
e.g. for 1st line, J_A=K_A=0 → A(t+1) = A(t) = 0; J_B=1, K_B=0→B(t+1) = 1

baiyh@sustech.edu.cn

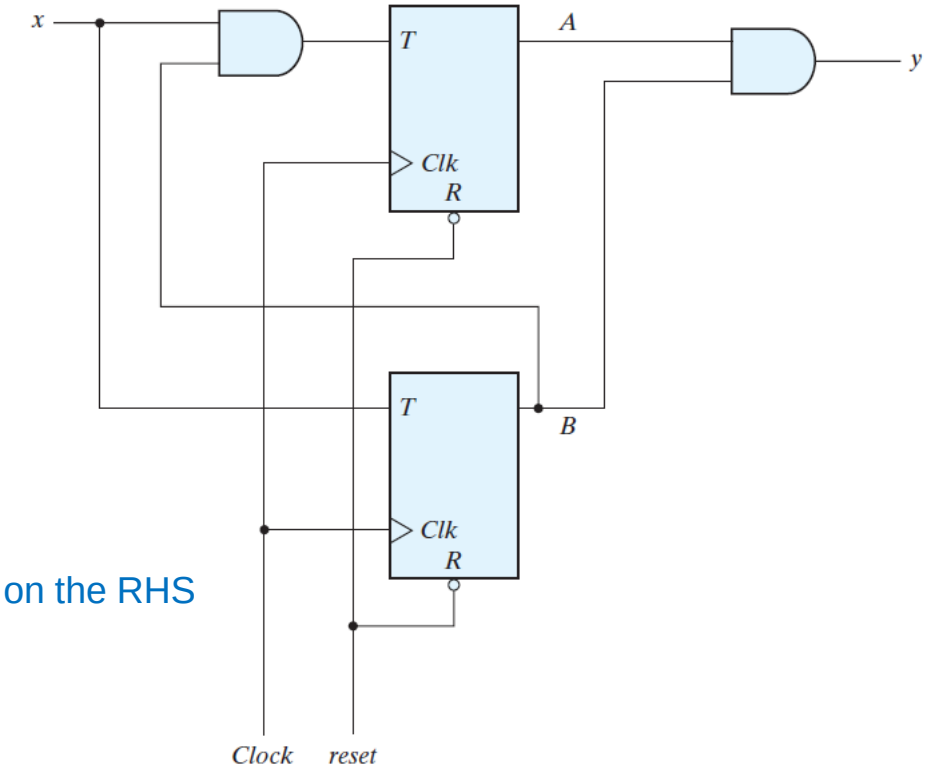
Analysis Example 3: A sequential circuit using TFF

1

- Excitation/Input equation

  $T_A = Bx$

  $T_B = x$



2

- state equation

  $A(t+1) = T_A \oplus Q_A = (Bx) \oplus A$

  $B(t+1) = T_B \oplus Q_B = x \oplus B$

- Output equation

  $y = AB$

(t) is omitted on the RHS

Substituting input equation into TFF's charc. Equation $Q(t+1) = T(t) \oplus Q(t)$

baiyh@sustech.edu.cn

Analysis Example 3: A sequential circuit using TFF

2

- state/output equation

  $A(t+1) = (Bx) \oplus A = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$

  $B(t+1) = x \oplus B$

  $y = AB$

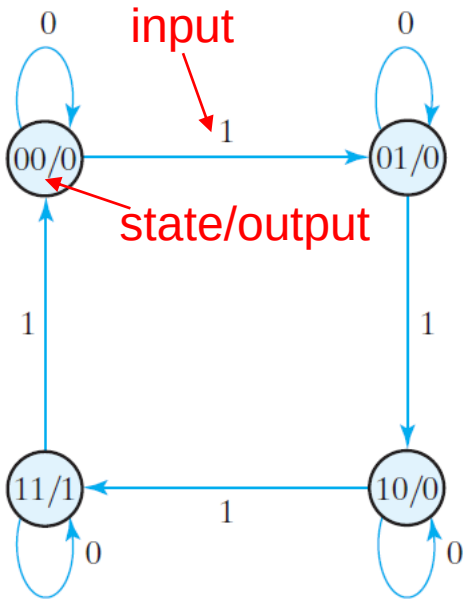
3

- state table

Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

4

- Generate state diagram



Moore Model

baiyh@sustech.edu.cn

Outline

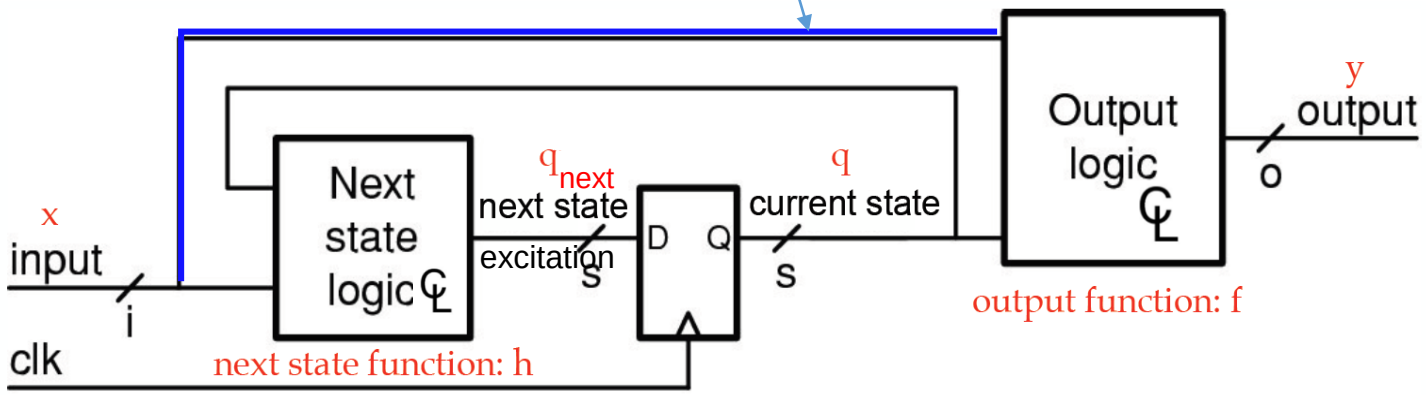
- Analysis of Sequential Circuits
- **Finite State Machine**
- State Minimization & Encoding
- Design of Sequential Circuits

baiyh@sustech.edu.cn

Finite State Machine (FSM)

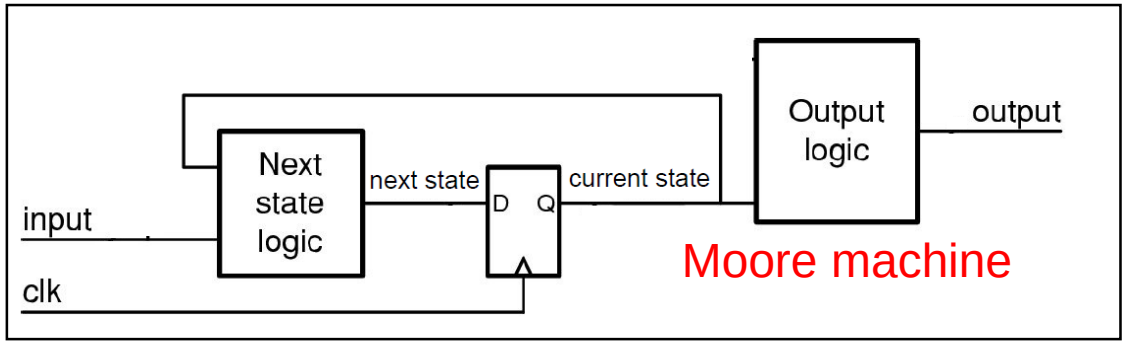
- A synchronous sequential circuit can be modeled by FSM
- State register $q(t+1) = qnext(t)$
  - Stores current state
  - Loads next state at clock edge
- Combinational logic
  - Computes next state (next state logic h: $x \times q \rightarrow qnext$)
  - Computes outputs
    - output logic f: $x \times q \rightarrow y$ (**Mealy** machine, with blue line)
    - or f: $q \rightarrow y$ (**Moore** machine, without blue line)

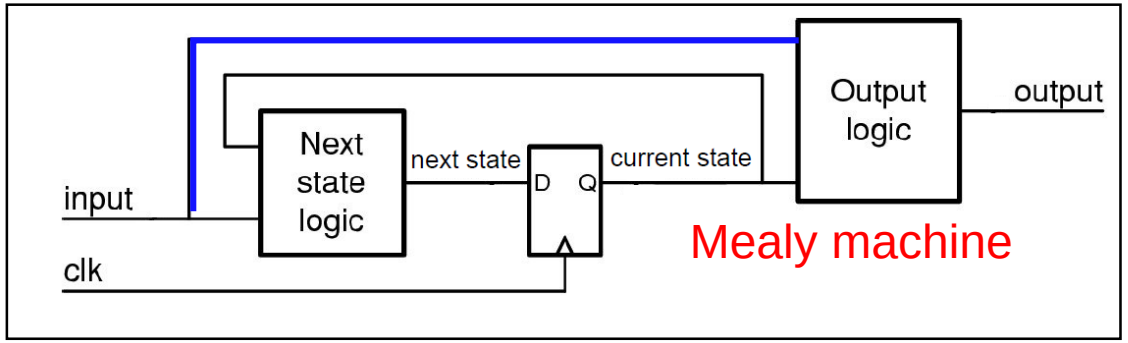
$Recall$: Combination logic： f: $x \rightarrow y$



Finite State Machine (FSM)

- A synchronous sequential circuit can be modeled by FSM (有限状态机)
  - Two types of finite state machines differ in output logic:
  - Moore FSM: outputs depend only on current state
  - Mealy FSM: outputs depend on current state and inputs

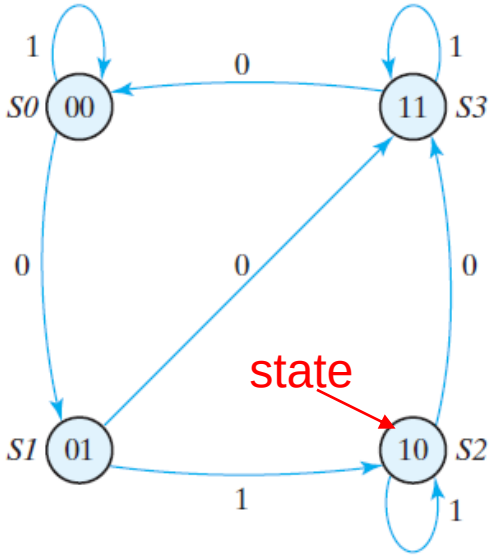




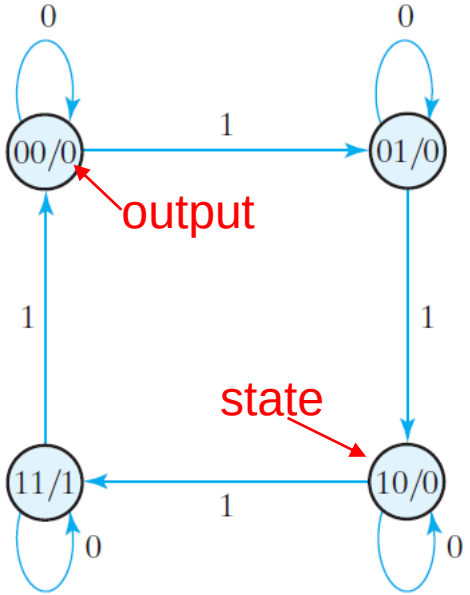
baiyh@sustech.edu.cn

Models for Sequential Circuits

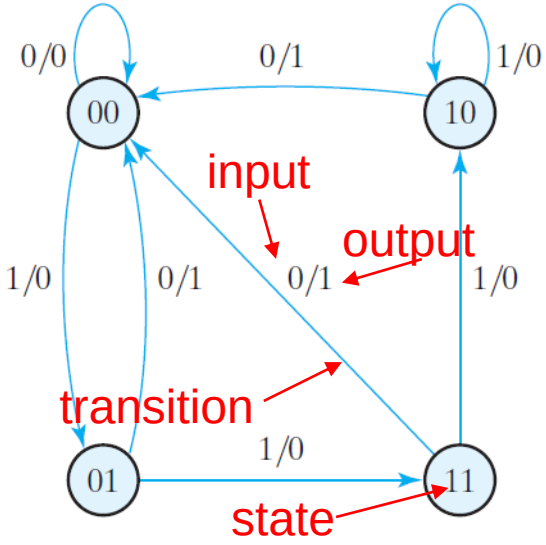
- Moore Model
- Mealy Model: To synchronize a Mealy circuit, the inputs must be synchronized with the clock and the outputs must be sampled immediately before the clock edge



Example of JKFF
no output, only state



Example of TFF
Moore Model



Example1
Mealy Model

baiyh@sustech.edu.cn

Outline

- Analysis of Sequential Circuits
- Finite State Machine
- **State Minimization & Encoding**
- Design of Sequential Circuits

baiyh@sustech.edu.cn

State Minimization (Reduction)

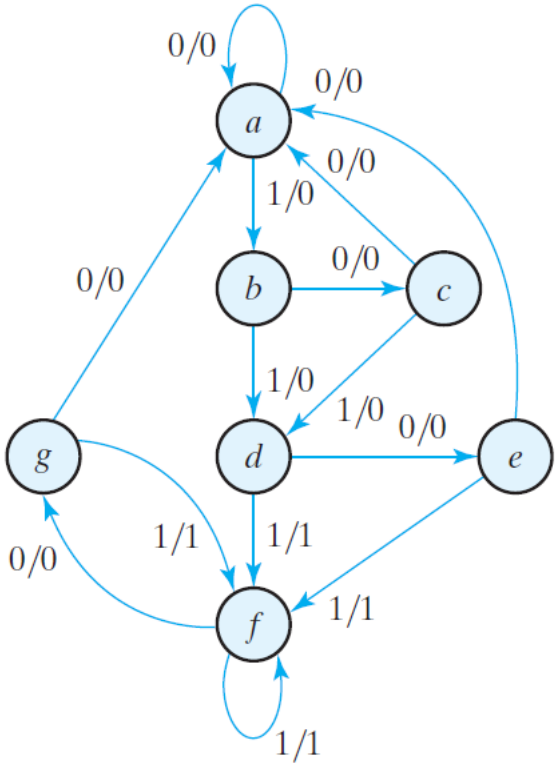
- State reduction
  - Reductions on the number of flip-flops (states) and the number of gates
  - For an FSM with *m* states, we need $log_2 m$ FFs
- Definition for FSM equivalence
  - (1) Given any input sequence, starting from any identical initial state, they produce the same output sequence.
  - (2) Two states, s_j and s_k in an FSM are said to be equivalent ($s_j \equiv s_k$), if

    $\forall i \in x, h(s_j, i) = h(sk, i)$, $\forall i \in x, f(s_j, i) = f(sk, i)$ (h is the state equation, f is the output equation)
- Reduction steps
  1. Find rows in the state table that have identical next state and output entries. They correspond to equivalent states. If there are no equivalent states, stop.
  2. When 2 states are equivalent, one of them can be removed. Update the entries of the remaining table to reflect the change. Go to 1.

State Minimization Example

State Table

Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
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

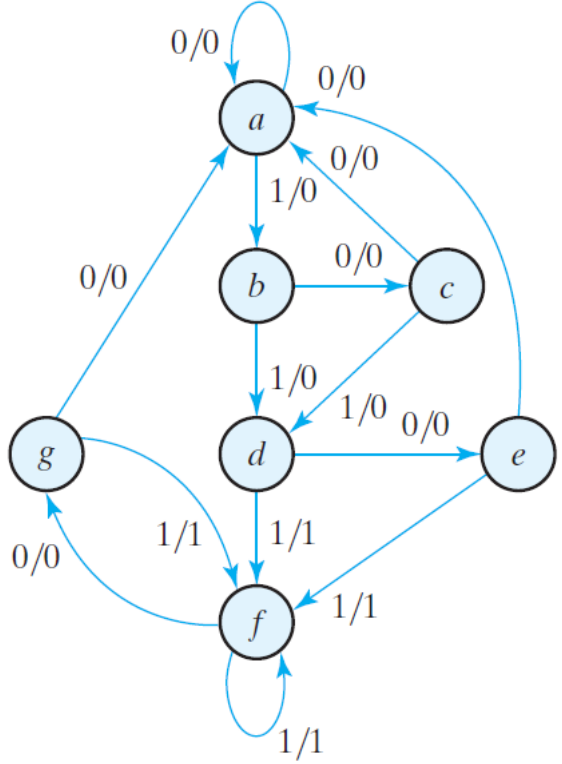


t	0	1	2	3	4	5	6	7	8	9	10
s	a	a	b	c	d	e	f	f	g	f	g
x	0	1	0	1	0	1	1	0	1	0	0
y	0	0	0	0	0	1	1	0	1	0	0

I/O sequence based on input pattern x and initial state being ***a***

baiyh@sustech.edu.cn

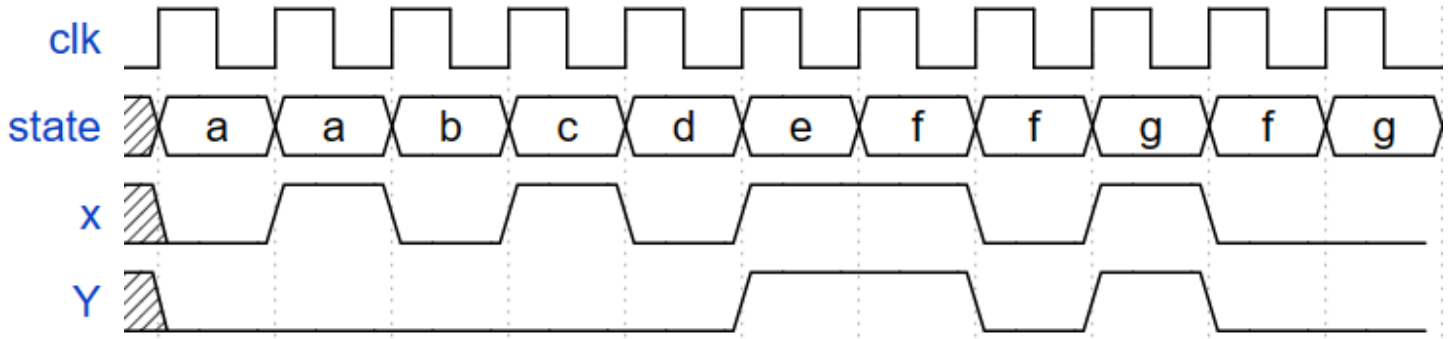
State Minimization Example



t	0	1	2	3	4	5	6	7	8	9	10
s	a	a	b	c	d	e	f	f	g	f	g
x	0	1	0	1	0	1	1	0	1	0	0
y	0	0	0	0	0	1	1	0	1	0	0

I/O sequence

Timing diagram



State Minimization Example

1st turn

Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
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~~ e	f	0	1
~~g~~	~~a~~	~~f~~	~~0~~	~~1~~

e==g

1

2

2nd turn

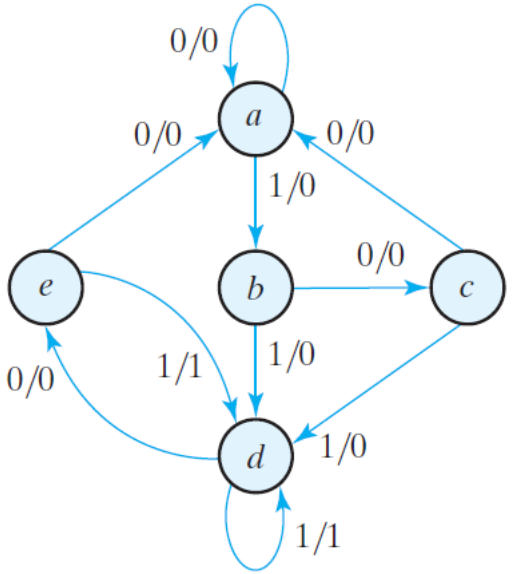
Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	~~f~~ d	0	1
e	a	~~f~~ d	0	1
f	e	f	0	1

d==f

1

2

State Minimization Example



Reduced state diagram

Reduced State Table

Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
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

t	0	1	2	3	4	5	6	7	8	9	10
s	a	a	b	c	d	e	d	d	e	d	e
x	0	1	0	1	0	1	1	0	1	0	0
y	0	0	0	0	0	1	1	0	1	0	0

Resulting I/O sequence with the same input pattern x and initial state being ***a***

baiyh@sustech.edu.cn

State Encoding/Assignment

- Different state encodings (assignments) result in different circuits for the intended FSM.
- There is no easy state-encoding procedure that guarantees a minimal-cost or minimum-delay combinational circuits
  - Exploration of all possibilities are impossible.
  - Heuristic are often used
    - Binary counting
    - Minimum-bit change
    - One-hot encoding

baiyh@sustech.edu.cn

State Encoding/Assignment

- One-hot encoding usually leads to simpler decoding logic for the next state and output.

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

Reduced State Table with Binary Assignment 1

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

Outline

- Analysis of Sequential Circuits
- Finite State Machine
- State Minimization & Encoding
- **Design of Sequential Circuits**

baiyh@sustech.edu.cn

Design Procedure of Sequential Circuits

1. Specification: design description or timing diagram
2. Formulation: develop state diagram
3. Generate state and output tables
4. Minimize States if necessary
5. Assign binary values to the state (encoding)
6. Derive state and output equations
7. **Choose memory elements** (DFFs, JKFFs, TFFs)
8. Derive simplified excitation/input equations and output equations
9. Draw logic schematic

baiyh@sustech.edu.cn

Choice of Memory Elements

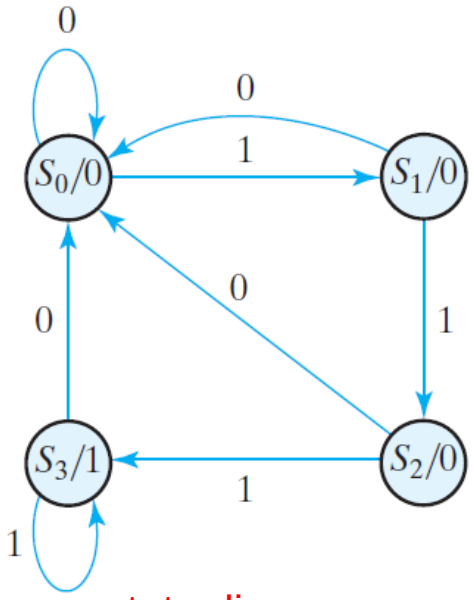
- Given the state transition table, we wish to find the FF input conditions that will cause the required transition.
  - A tool for such a purpose is the excitation table, which can be derived from the characteristic table/equation.
  - D FFs are good for applications requiring data transfer (shift registers).
  - T FFs are good for those involving complementation (binary counters).
  - Many digital systems are constructed entirely with JK FFs because they are the most versatile available.

Design Example 1: Design with DFF: A Sequence Detector

1

- Detect three consecutive 1's in a string of bits (using Moore machine, overlapping)
  - If detected, output=1; otherwise output=0

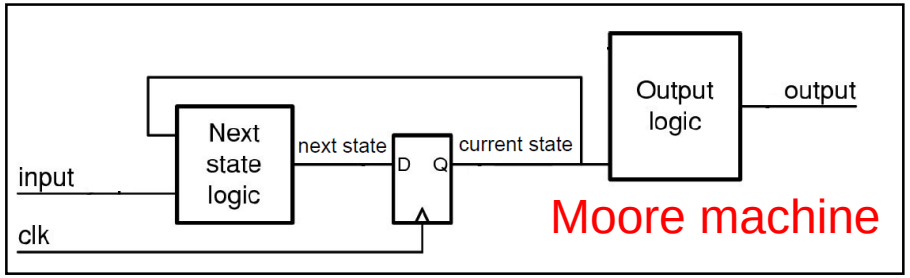
2



state diagram

0 1 1 0 1 1 1 1 0 1 1 1

Overlapping: detection window can be overlapped



baiyh@sustech.edu.cn

Design Example 1: Design with DFF: A Sequence Detector

3

4

already minimal states

Next state and output table

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
S_0	S_0	S_1	0	0
S_1	S_0	S_2	0	0
S_2	S_0	S_3	0	0
S_3	S_0	S_3	1	1

5

Reduced state table and state assignment

Present State (AB)	Next State (AB)		Output (y)	
	x=0	x=1	x=0	x=1
00	00	01	0	0
01	00	10	0	0
10	00	11	0	0
11	00	11	1	1

baiyh@sustech.edu.cn

Design Example 1: Design with DFF: A Sequence Detector

5

- State equation

$A(t+1) = D_A(A(t),B(t), x)$
$= \sum(3, 5, 7)$
$B(t+1) = D_B(A(t),B(t), x)$
$= \sum(1, 5, 7)$

- Output equation

$y(A,B, x) = \sum(6, 7)$

- Choose DFFs (default)
  - State encoding using 2 bits, so 2 DFFs are needed
    - $A(t+1) = D_A(t)$
    - $B(t+1) = D_B(t)$

6

Reduced state table and state assignment

Present State (AB)	Next State		Output (y)	
	x=0	x=1	x=0	x=1
00	00	01	0	0
01	00	10	0	0
10	00	11	0	0
11	00	11	1	1

7

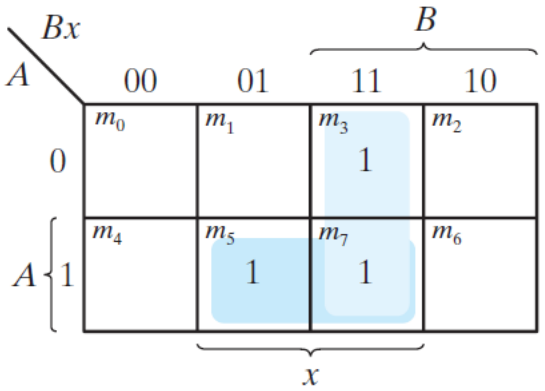
1st form state table

			D_A	D_B	
A(t)	B(t)	x	A(t+1)	B(t+1)	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

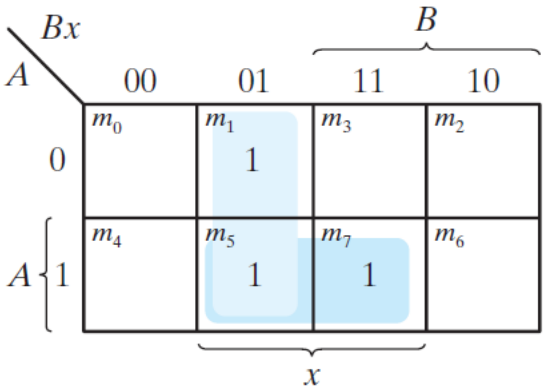
baiyh@sustech.edu.cn

Design Example 1: Design with DFF: A Sequence Detector

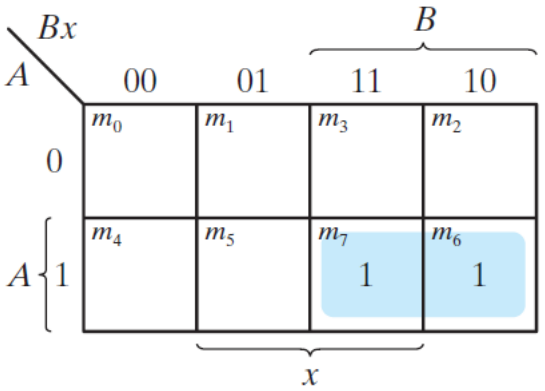
- Derive excitation equations and optimize logic implementation



$D_A = Ax + Bx$



$D_B = Ax + B'x$

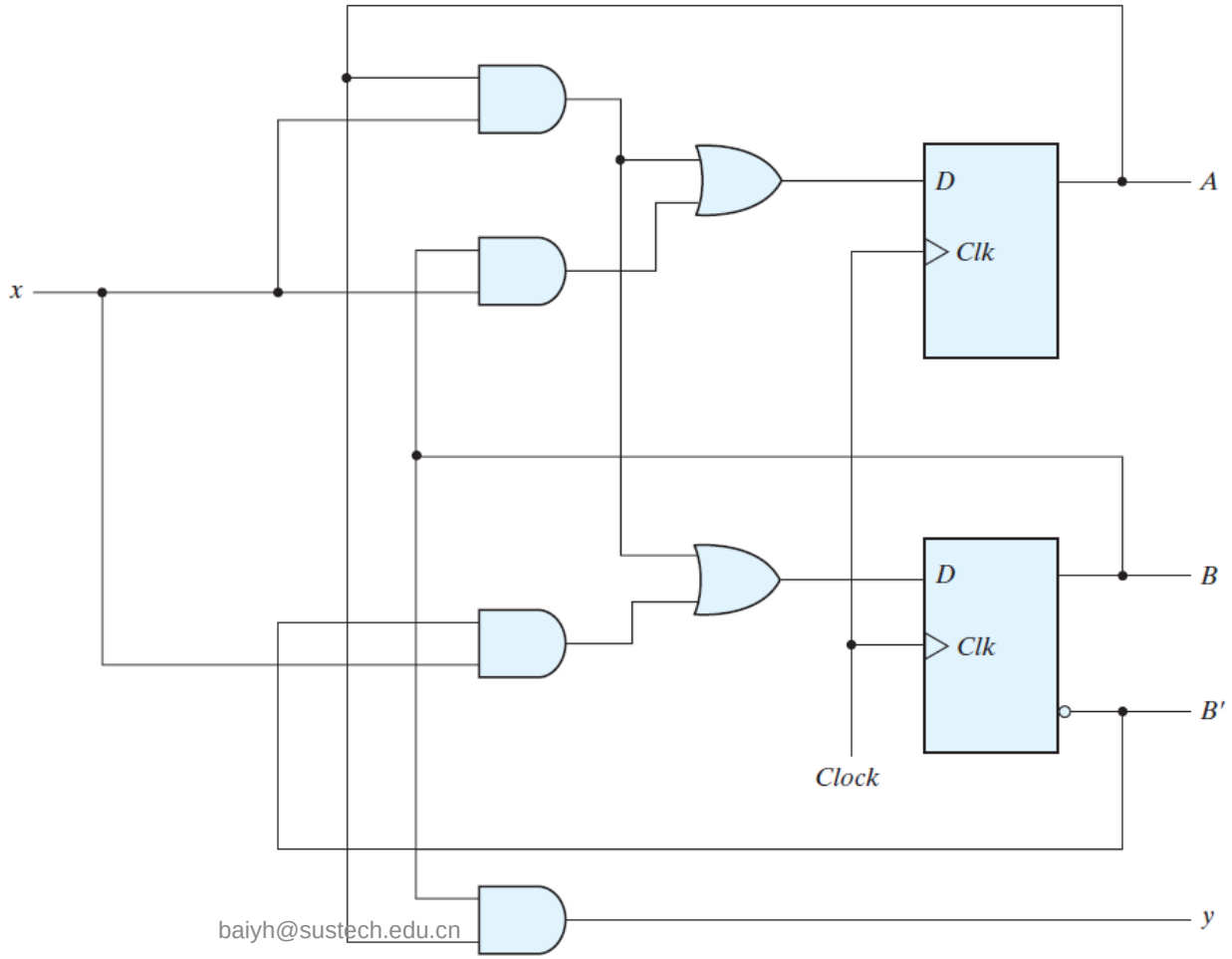


$y = AB$

- Input/Excitation equations (minimized)
  - D_A = Ax + Bx
  - D_B = Ax + B'x
- Output equations (minimized)
  - y = AB

Design Example 1: Design with DFF: A Sequence Detector

- Logic diagram
  - a **Moore-type** sequence detector
  - y depends only on A and B

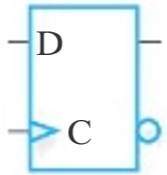


baiyh@sustech.edu.cn

FF's Excitation Table

- **The FF's excitation table(input table)** is derived from the FF's characteristic table by transposing input and output columns

D



Characteristic Table

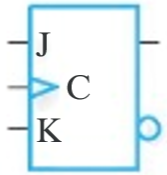
D	Q(t+1)	Operation
0	0	Reset
1	1	Set

Characteristic Equation: $Q(t+1) = D(t)$

Excitation Table

Q(t)	Q(t+1)	D	Operation
0	0	0	Reset
1	1	1	Set

JK



Characteristic Table

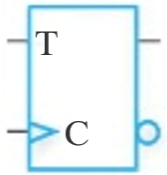
J	K	Q(t+1)	Operation
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

Characteristic Equation: $Q(t+1) = J(t)\,Q'(t) + K'(t)\,Q(t)$

Excitation Table

Q(t)	Q(t+1)	J	K	Operation
0	0	0	X	No change
0	1	1	X	Set
1	0	X	1	Reset
1	1	X	0	No Change

T



Characteristic Table

T	Q(t+1)	Operation
0	$Q(t)$	No change
1	$Q'(t)$	Complement

Characteristic Equation: $Q(t+1) = T(t) \oplus Q(t)$

Excitation Table

Q(t)	Q(t+1)	T	Operation
0	0	0	No change
0	1	1	Complement
1	0	1	
1	1	0	

baiyh@sustech.edu.cn

Design Example 2: Design with JKFF

1 • Assume a state table as follows, design a sequential circuit with JKFF

2 • Characteristics: date input x (and clock/reset of course)

• The outputs can be the next states

• State table is directly provided, so step 2 is skipped

Present State		Input	Next State	
A	*B*	*x*	*A*	*B*
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

baiyh@sustech.edu.cn

Design Example 2: Design with JKFF

- Design with JFKK has similar procedure as with DFF, except that **we need to enhance the state table by evaluating the excitation table based on the state transition.**
  - So we can then derive the input equations

3 4 5

State Table and JK Flip-Flop Inputs

Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	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	X	1
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	X	0	X	0
1	1	1	0	0	X	1	X	1

Q(t)	Q(t + 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Evaluating in advance the FF inputs based on excitation table

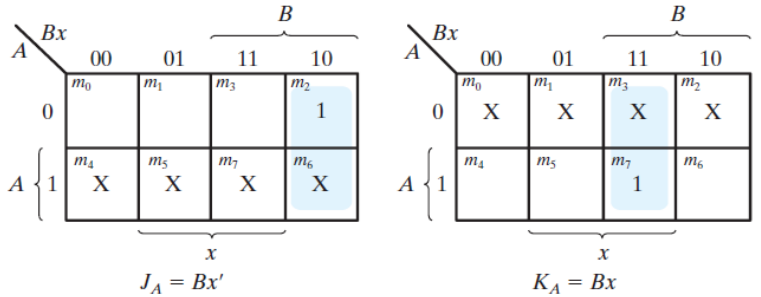
6 7

- Input equation will be evaluated when choosing JKFFs

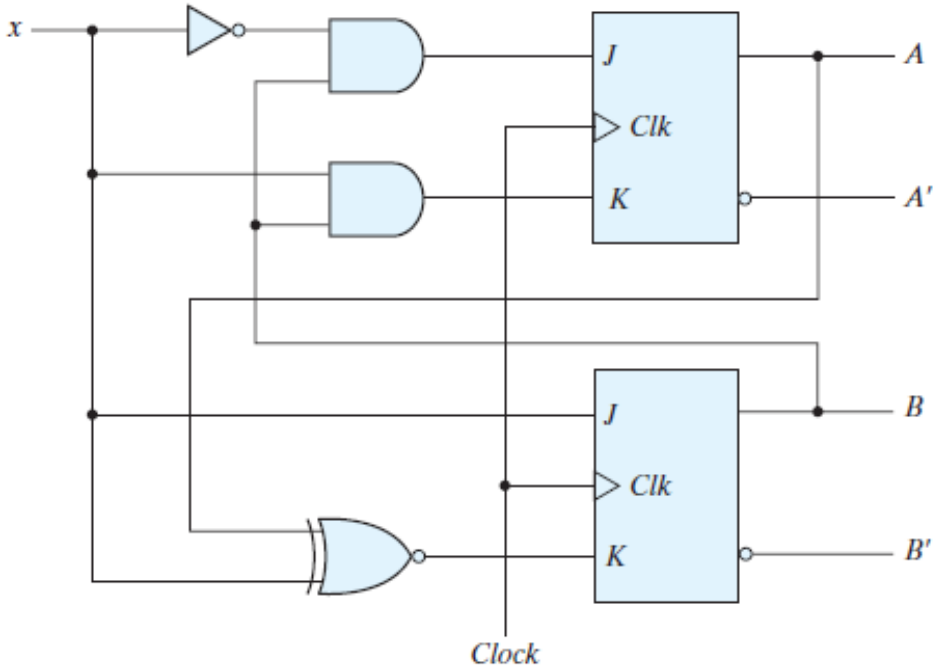
baiyh@sustech.edu.cn

Design Example 2: Design with JKFF

8 • Derive excitation equations and optimize logic implementation



9



• Input/Excitation equations (minimized)

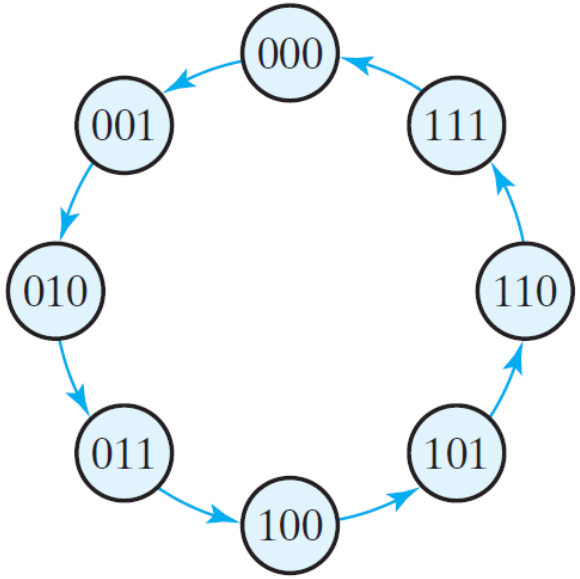
$J_A = Bx'$, $J_B = x$,
$K_A = Bx$, $K_B = (A \oplus x)'$.

baiyh@sustech.edu.cn

Design Example 3: Design with TFF: A Binary Counter

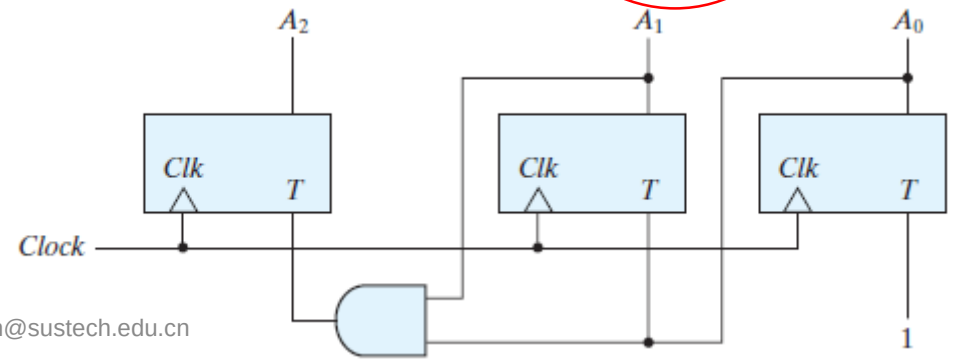
- Characteristics: No date input (except clock/reset), The outputs can be the next states

Q(t)	Q(t + 1)	T
0	0	0
0	1	1
1	0	1
1	1	0



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}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

$T_{A2} = A_1A_0$
$T_{A1} = A_0$
$T_{A0} = 1$

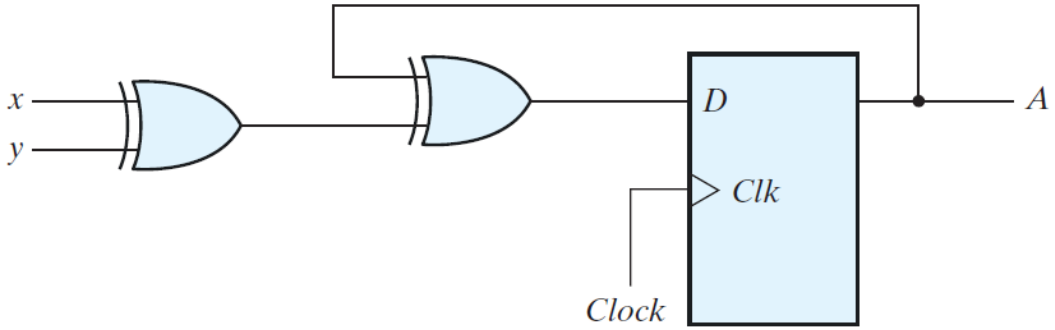


baiyh@sustech.edu.cn

Summary

- Finite State Machine
  - Moore Machine
  - Mealy Machine
- Analysis and design of clocked sequential circuits: Recommended steps.
- Ways to describe a sequential circuit:
- Function table, state table, state diagram, next state equation, logic diagram, excitation table, K-map

Exercise1: Analyze DFF base sequential circuit



1. Excitation/Input equation
2. state equation, Output equation?
3. state table, Output table?
4. Generate state diagram

baiyh@sustech.edu.cn

Exercise1: Analyze DFF base sequential circuit

1

- Excitation/Input equation

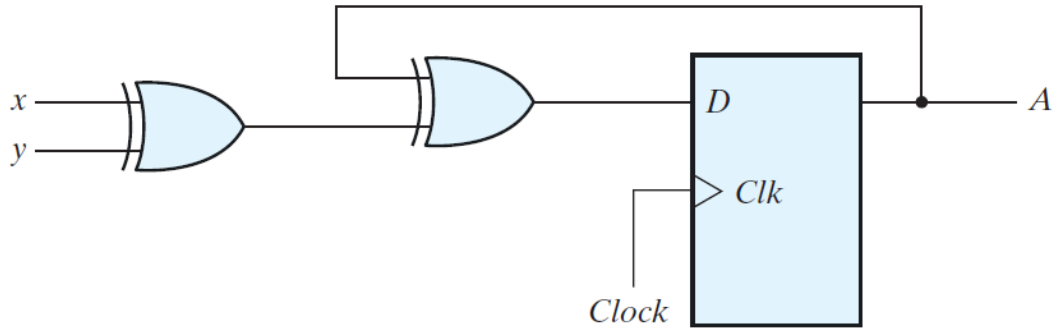
$$D_A = A \oplus x \oplus y$$

2

- state equation

$$A(t+1) = A(t) \oplus x(t) \oplus y(t)$$

- state table
  - No output equation since output comes from the output of DFF(state Q)

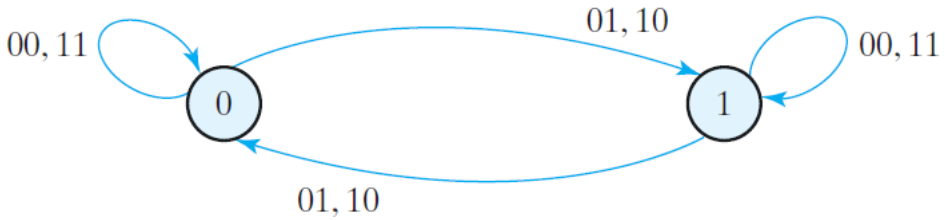


3

Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- Generate state diagram

4



baiyh@sustech.edu.cn

Exercise2: Design a sequence detector

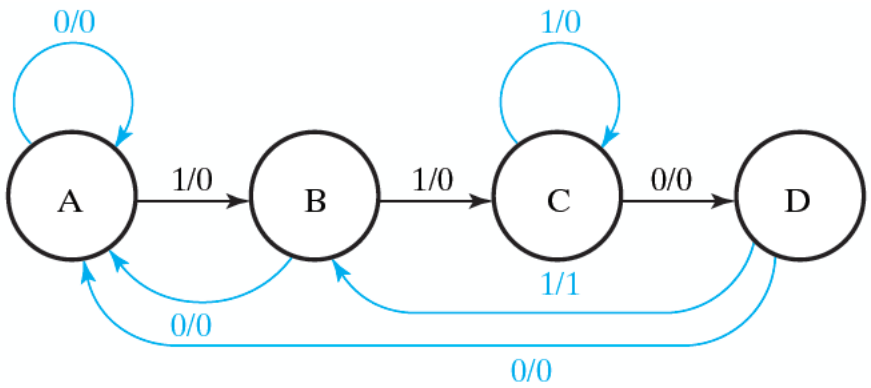
- Recognize the occurrence of a particular sequence of bits (1101) (Mealy machine, overlapping, using gray code for state assignment)

baiyh@sustech.edu.cn

Exercise2: Design a sequence detector

1 • Recognize the occurrence of a particular sequence of bits (1101) (Mealy machine, overlapping, using gray code for state assignment)

2



3

4

Present State	Next State		Output Z	
	X = 0	X = 1	X = 0	X = 1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

baiyh@sustech.edu.cn

Exercise2: Design a sequence detector

5

state assignment with 2-bit binary Gray code

Present State	Next State		Output Z	
AB	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	00	11	0	0
11	10	11	0	0
10	00	01	0	1

6

$$A(t+1) = D_A(A(t), B(t), X) = \sum(3,6,7) = AB + BX$$

7

$$B(t+1) = D_B(A(t), B(t), X) = \sum(1,3,5,7) = X$$

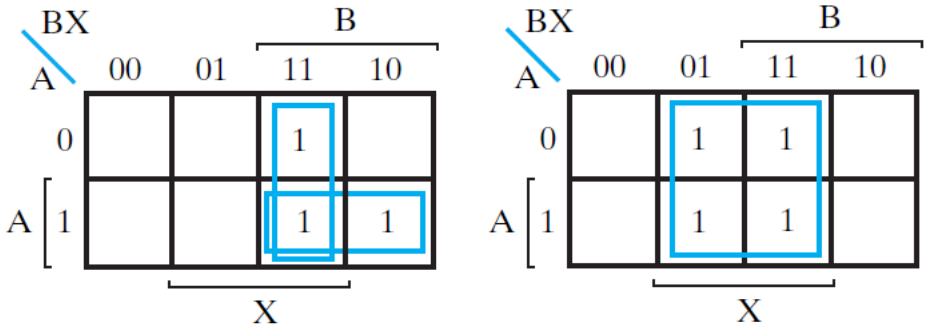
$$Z(A, B, X) = \sum(5) = A\bar{B}X$$

1st form state table

Current State		Input	Next State		Output
A	B	X	A	B	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

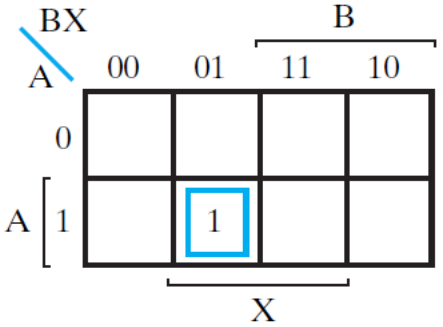
Exercise2: Design a sequence detector

8



$D_A = AB + B\bar{X}$

$D_B = X$



$Z = A\bar{B}X$

9