# ECE 223
# Digital Circuits and Systems

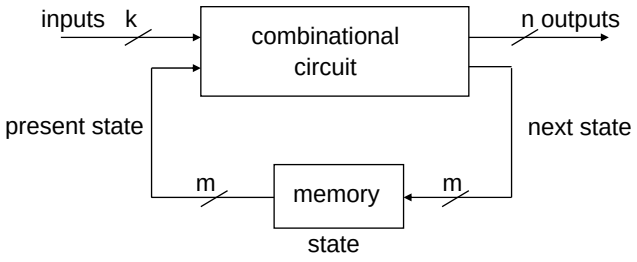
## Synchronous Logic

M. Sachdev
Dept. of Electrical & Computer Engineering
University of Waterloo

1

## Sequential Circuits

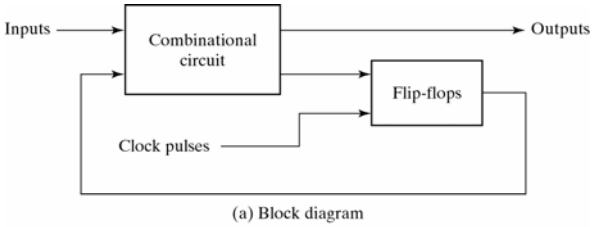
- Combinational circuits
  - Output = f (present inputs)
- Sequential circuits
  - Output = f (present inputs and past inputs)
  - Circuit remembers past history
  - Must contain memory




2

# Synchronous Sequential Circuits

- A synchronizing, periodic signal, Clock, facilitates the transition from present state to next state
  - Memory is provided by flip-flops




(a) Block diagram

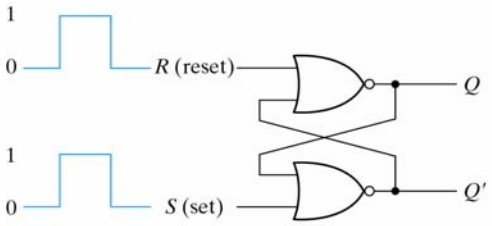
(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

3

# SR (Set Reset) Latches

- NOR Latch
- SR = 11 is avoided
  - Outputs are not complementary
  - Input transition from 11 → 00 may cause circuit to: (i) fall into either state, or become meta-stable




(a) Logic diagram

| $S$ | $R$ | $Q$ | $Q'$ | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 0 | (after $S = 1, R = 0$) |
| 0 | 1 | 0 | 1 | |
| 0 | 0 | 0 | 1 | (after $S = 0, R = 1$) |
| 1 | 1 | 0 | 0 | |

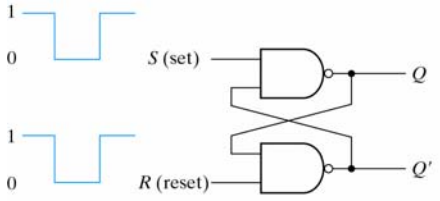
(b) Function table

Fig. 5-3 *SR* Latch with NOR Gates

4

# SR (Set Reset) Latches

- NAND Latch
- SR = 00 is avoided
  - Outputs are not complementary
  - Input transition from 00 → 11 may cause circuit to: (i) fall into either state, or become meta-stable




| S | R | Q | Q′ | |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 1 | (after $S = 1, R = 0$) |
| 0 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 0 | (after $S = 0, R = 1$) |
| 0 | 0 | 1 | 1 | |

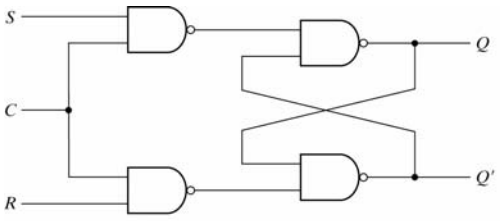
(a) Logic diagram (b) Function table

Fig. 5-4 *SR* Latch with NAND Gates

5

# SR Latch with control Input

- C = 0
  - Latch retains its state
- C = 1
- Allows propagation of SR inputs




| 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 |

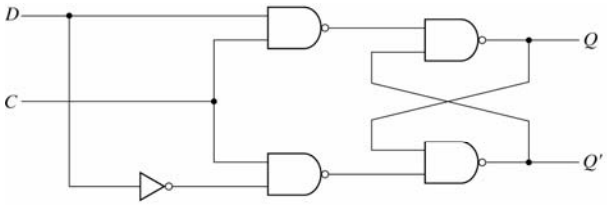
(a) Logic diagram (b) Function table

Fig. 5-5 SR Latch with Control Input

6

# Data (D) Latch

- Data latch eliminate, the need for complementary inputs
  - Outputs are also complementary




(a) Logic diagram

| C | D | Next state of Q |
|---|---|---|
| 0 | X | No change |
| 1 | 0 | Q = 0; Reset state |
| 1 | 1 | Q = 1; Set state |

(b) Function table

Fig. 5-6 D Latch

7

# Flip-flop

- Latch Is level sensitive to the control signal
  - Multiple data transition may cause problem while C =1
- Flip-flop is edge triggered
  - Flip-flop samples the data on Clock transition

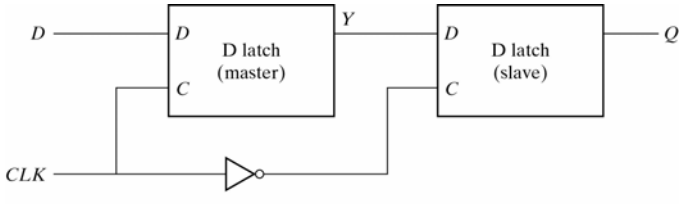



Fig. 5-9 Master-Slave *D* Flip-Flop

8

# Edge Triggered Flip-flop

- Efficient implementation
  - Multiple data transitions do not affect the output

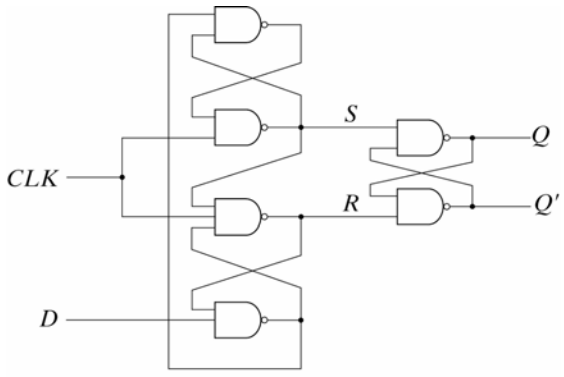
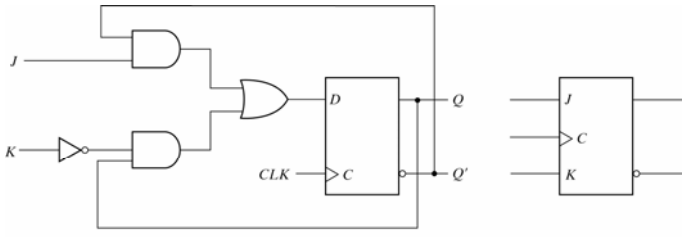



Fig. 5-10 *D*-Type Positive-Edge-Triggered Flip-Flop

9

# J-K Flip-flop

- Versatile flip-flop
  - Can be Set, Reset, or Complement (toggle) its output




(a) Circuit diagram (b) Graphic symbol

Fig. 5-12 *JK* Flip-Flop

10

# J-K Flip-flop

$Q_{n+1} = JQ_n' + K'Q_n = D$

$Q_{n+1}$

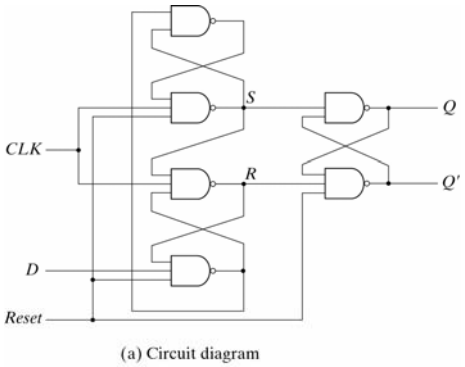
| $Q_n$ \ JK | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | 1 |
| 1 | 1 | | | 1 |

| J | K | $Q_n$ | $Q_{n+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

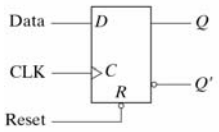
11

# Asynchronous Inputs

- Ability to Reset (or Set) irrespective of Clock state
  - Often needed in computation




(a) Circuit diagram




(b) Graphic symbol

| R | C | D | Q | Q' |
|---|---|---|---|---|
| 0 | X | X | 0 | 1 |
| 1 | ↑ | 0 | 0 | 1 |
| 1 | ↑ | 1 | 1 | 0 |

(b) Function table

Fig. 5-14 *D* Flip-Flop with Asynchronous Reset

# Analysis of Clocked Sequential Circuits

- Procedure
  - Determine **state equations** (tranistion equations)
  - Determine the **state table** (transition table)
  - Determine **state diagram**
- Example – Analysis with D flip-flop

A(t+1) = A(t)x(t) + B(t)x(t)

B(t+1) = A'(t)x(t)

Or

A(t+1) = Ax + Bx

B(t+1) = A'x

Similarly, y = (A+B)x'

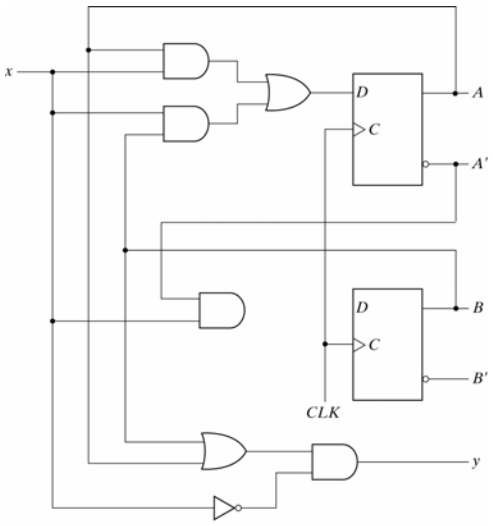



Fig. 5-15 Example of Sequential Circuit

13

# 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 | 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 |

14

# State Table, State Diagram

| Present State | Next State | | Output | |
|---|---|---|---|---|
| | x =0 | x =1 | x =0 | x =1 |
| AB | AB | AB | y | y |
| 00 | 00 | 01 | 0 | 0 |
| 01 | 00 | 11 | 1 | 0 |
| 10 | 00 | 10 | 1 | 0 |
| 11 | 00 | 10 | 1 | 0 |

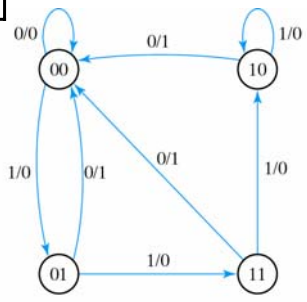



Fig. 5-16 State Diagram of the Circuit of Fig. 5-15

# Analysis with JK Flip-flop

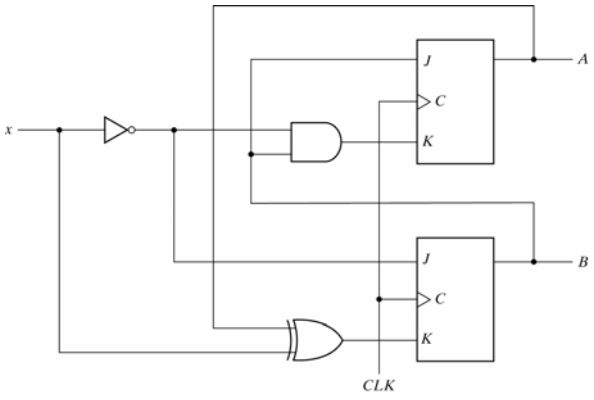



Fig. 5-18 Sequential Circuit with *JK* Flip-Flop

$J_A = B$, $K_A = Bx'$

$J_B = x'$ $K_B = A'x + Ax' = A \oplus x$

16

# State Table

| Present State | | Input | Next State | | Flip-flop Inputs | | | |
|---|---|---|---|---|---|---|---|---|
| A | B | x | A | B | JA | KA | JB | KB |
| 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 |

A(t+1) = JA' + K'A

B(t+1) = JB' + K'B

17

# State Diagram

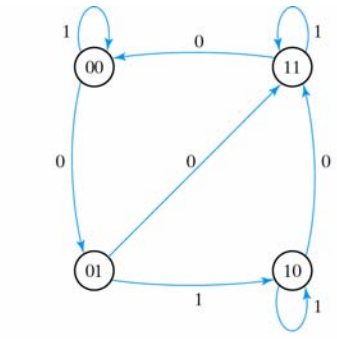


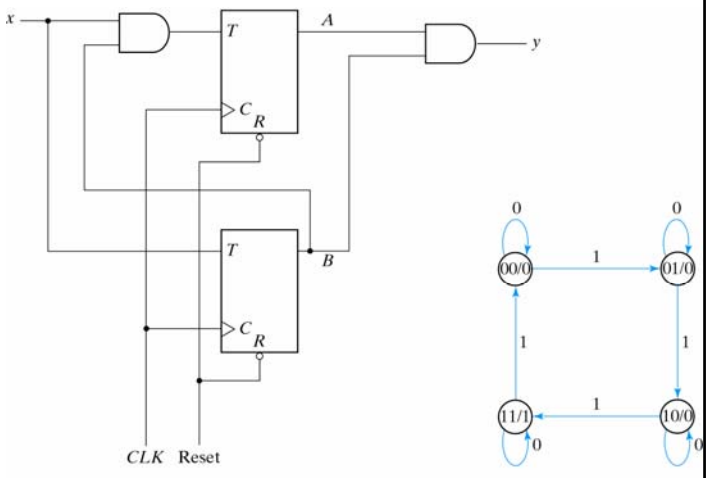

Fig. 5-19 State Diagram of the Circuit of Fig. 5-18

18

# Analysis with Toggle Flip-flop

- The Characteristic Equation

Q(t+1) = T⊕Q
= T'Q + TQ'




(a) Circuit diagram (b) State diagram

Fig. 5-20 Sequential Circuit with *T* Flip-Flops

# Analysis with Toggle Flip-flop

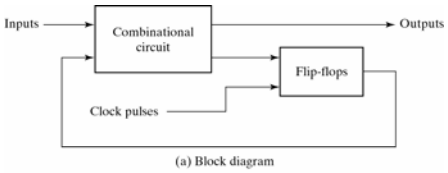
| Present State | | Input | Next State | | Output |
|---|---|---|---|---|---|
| A | B | x | A | B | y |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0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

# Mealy and Moore Models (Machine)

- Most general model of a sequential circuit has inputs, outputs, and internal states
  - Two different models – (i) Mealy model, (ii) Moore Model
- Difference is how output is generated

**Mealy Model** – Output is a function of present state & input

**Moore Model** – Output is only a function of present state



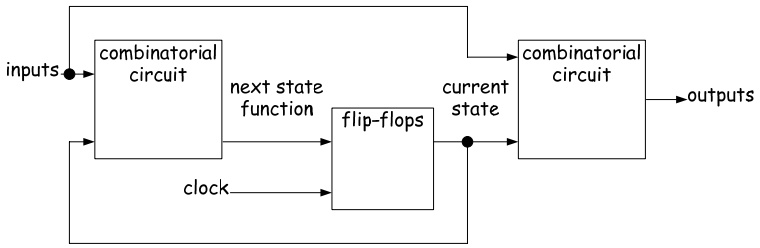

(a) Block diagram

(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

21

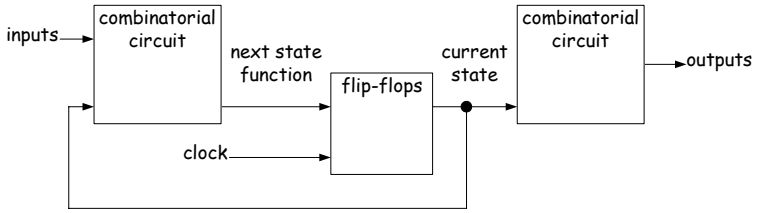
# Mealy Machine




- Observation
  - Output can change asynchronous to the clock
  - The output delay with respect to clock is unpredictable
  - Difficult to do the timing analysis

22

# Moore Machine




- Observation
  - Output changes synchronously to the clock
  - The output delay with respect to clock is predictable
  - Easier timing analysis

23

# State Reduction

- Sequential circuit analysis
  - Circuit diagram → state table (or state diagram)
- Sequential circuit design
  - State diagram (state table) → circuit diagram
- Redundant state may exist in a state diagram (or table)
  - By eliminating them → reduce the # of logic gates and flip-flops

"Two states are equivalent if for each member of the set of inputs, they give exactly same state or an equivalent state

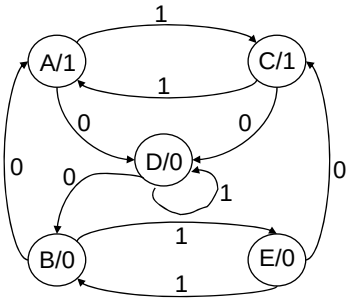
  - When two states are equivalent, one of them can be removed

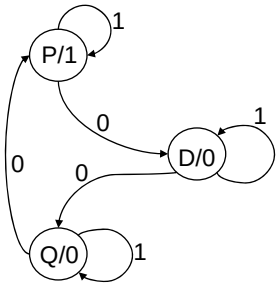
24

# State Reduction - Example

| Present State | Next State | | Output |
|---|---|---|---|
| | X=0 | X=1 | |
| A | D | C | 1 |
| B | A | E | 0 |
| C | D | A | 1 |
| D | B | D | 0 |
| E | C | B | 0 |

| Present State | Next State | | Output |
|---|---|---|---|
| | X=0 | X=1 | |
| P (A,C) | D | P | 1 |
| Q (B,E) | P | Q | 0 |
| D | Q | D | 0 |




25

# State Reduction – Home Work

- Reduced the shown state diagram

  (Page 199 in the text book)

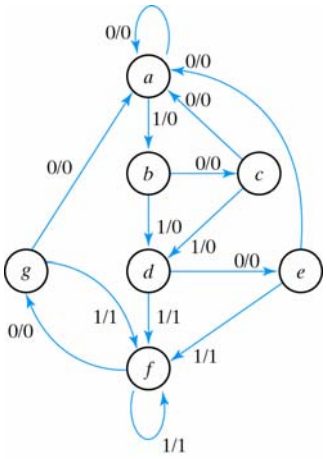



Fig. 5-22 State Diagram

26

# State Assignment

- States must be assigned coded binary values
  - In order to realize with physical components

| 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 | d | 0 | 1 |
| e | a | d | 0 | 1 |

| 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 |

27

# Sequential Circuits – Design Procedure

- Procedure
  1. Words or timing diagram
  2. Draw state transition diagram
  3. Make state table
  4. Reduced state table (if possible)
  5. Assign binary values to states
  6. Choose flip-flop type
  7. Derive simplified flip-flop input equations and output equations
  8. Draw the logic diagram

28

# Synthesis using D Flip-flops

- Given the state diagram, design the circuit using D flip-flops

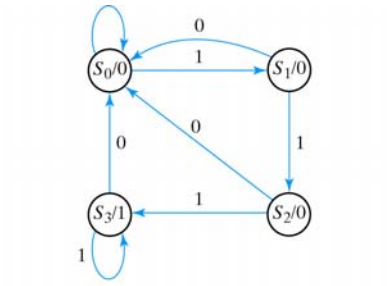



Fig. 5-24 State Diagram for Sequence Detector

| 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 | 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 |

29

# Synthesis using D Flip-flops

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | | 1 | 1 | |

$D_A = Ax + Bx$

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | | |
| 1 | | 1 | 1 | |

$D_B = Ax + B'x$

| A \ Bx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | 1 | 1 |

$y = AB$

30

# Synthesis using D Flip-flops

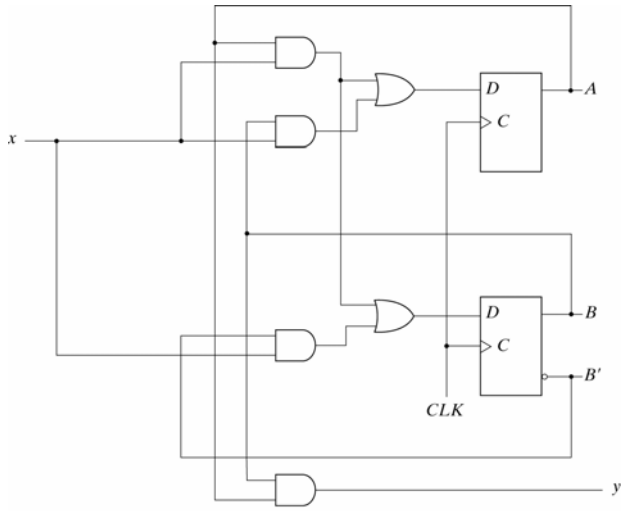



Fig. 5-26 Logic Diagram of Sequence Detector

31

# Synthesis using JK Flip-flop

- Given the state table, design the circuit using 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 | 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 |

32

# Synthesis using JK Flip-flop

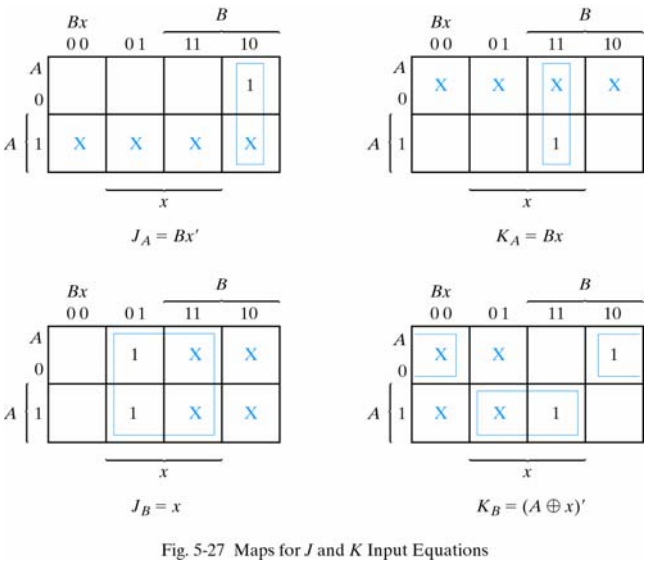



Fig. 5-27 Maps for *J* and *K* Input Equations

33

# Synthesis using JK Flip-flop

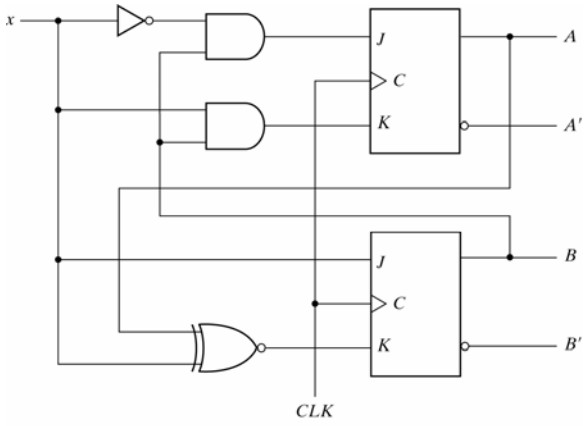



Fig. 5-28 Logic Diagram for Sequential Circuit with *JK* Flip-Flops

34

## Book Sections – Sequential Circuits

- Material is covered in Sections 5.1 – 5.4, 5.6 – 5.7

35