

Digital Systems

Prepared by Dr. Alaa Eldeen Sayed Ahmed

IT131

Sequential Circuits – Part III

Lecture 9



Sequential Circuits Part III

Contents

1. Sequential circuits
2. Flip-flops
3. Analysis of clocked sequential circuits
4. Design procedure

Sequential Circuits Part III

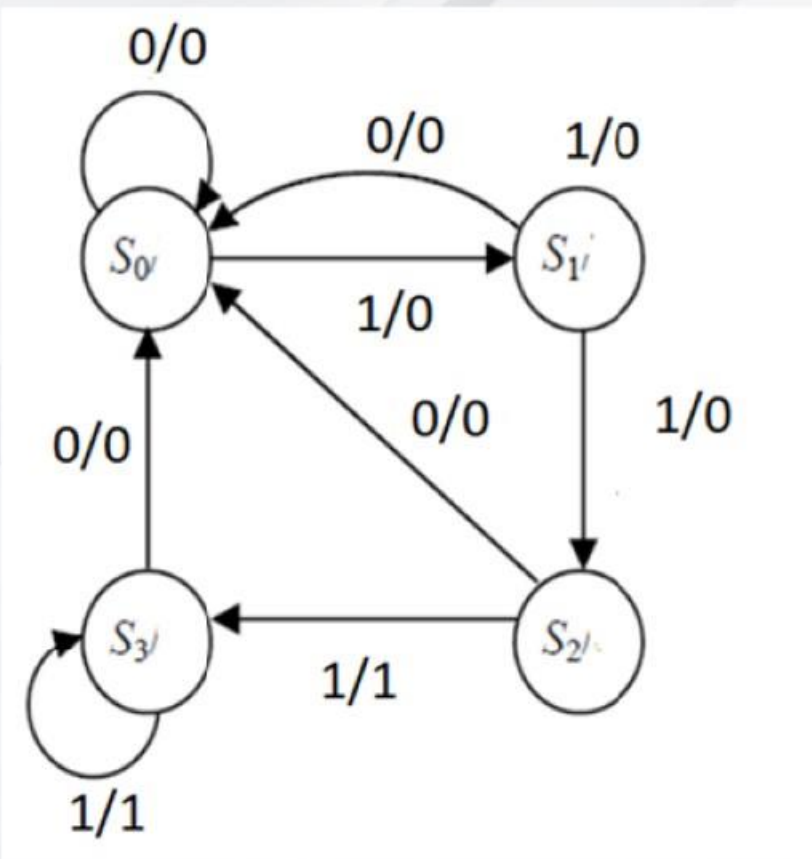
Design procedure

- The procedure for designing synchronous sequential circuit is:
1. Translate the design specifications to the state diagram for the circuit
 2. Assign binary values to the states
 3. Obtain the binary coded state table
 4. Choose the type of flip-flops to be used
 5. Derive the simplified flip-flop input equations and output equations
 6. Draw the logic diagram

Sequential Circuits Part III

Sequence detector

- **Example:** Design a circuit that detects three or more consecutive 1's in a string of bits coming through an input line
- The state diagram for the circuit is shown:



Sequential Circuits Part III

Sequence detector (Contd.)

- It is derived by starting with state S_0
- If input is 0, the circuit states in same state
- If the input is 1, it goes to state S_1 to indicate that a 1 was detected
- If the next input is 1, it changes to state S_2 to indicate the arrival of two consecutive 1's

Sequential Circuits Part III

Sequence detector (Contd.)

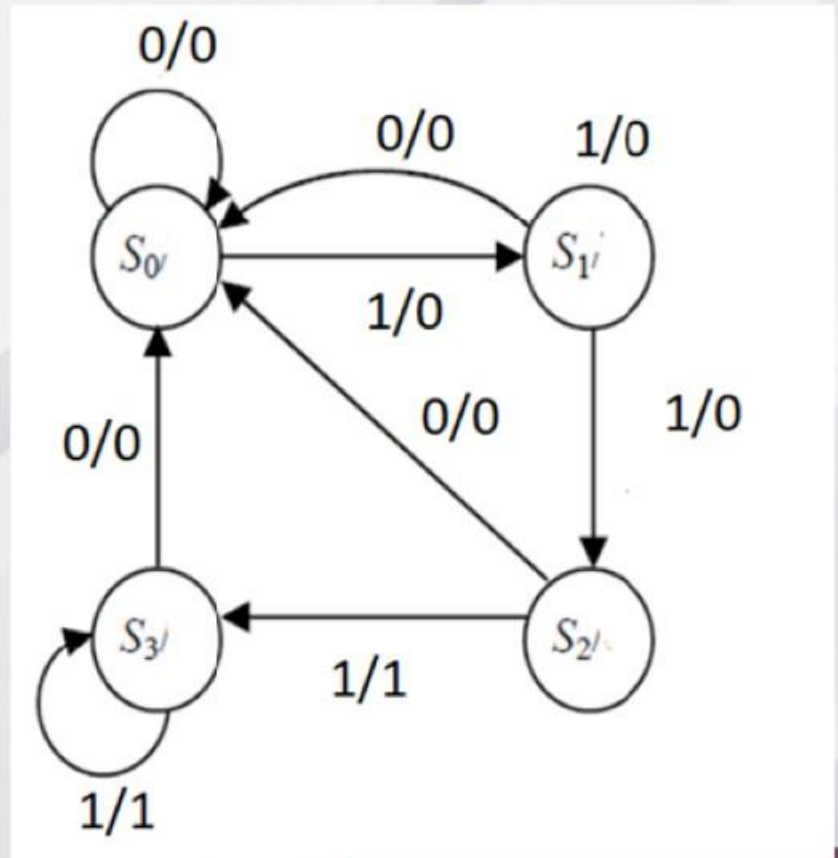
- But, if the input is 0, we go back to state S_0
- Third consecutive 1 sends the circuit to S_3
- If more 1's are detected, circuit stays at S_3
- Any 0 input sends the circuit back to S_0
- Circuit stays at S_3 , for three or more 1's

Sequential Circuits Design

Sequence detector (Contd.)

Assign binary values to the states

	P.S.	N.S.		Y
		$x = 0$	$x = 1$	
00 →	S_0	S_0	S_1	0
01 →	S_1	S_0	S_2	0
10 →	S_2	S_0	S_3	0
11 →	S_3	S_0	S_3	1



Sequential Circuits Part III

Synthesizing using D flip-flops

- To design the circuit, we need to assign binary codes to the states and list the table
- The state table is derived from the diagram

Present state		Input	Next State		Output
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>	<i>y</i>
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	1
1	1	0	0	0	0
1	1	1	1	1	1

Sequential Circuits Part III

Using D flip-flops (Contd.)

- We choose two D flip-flops to represent four states and label their outputs A and B
- There is one input x and one output y
- The characteristic equation of D flip-flop is

$$Q(t+1) = D_Q$$

- D flip-flop equations are

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

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

$$y(A, B, x) = \sum (5, 7)$$

Sequential Circuits Part III

Using D flip-flops (Contd.)

➤ Simplifying the above equations by K-map

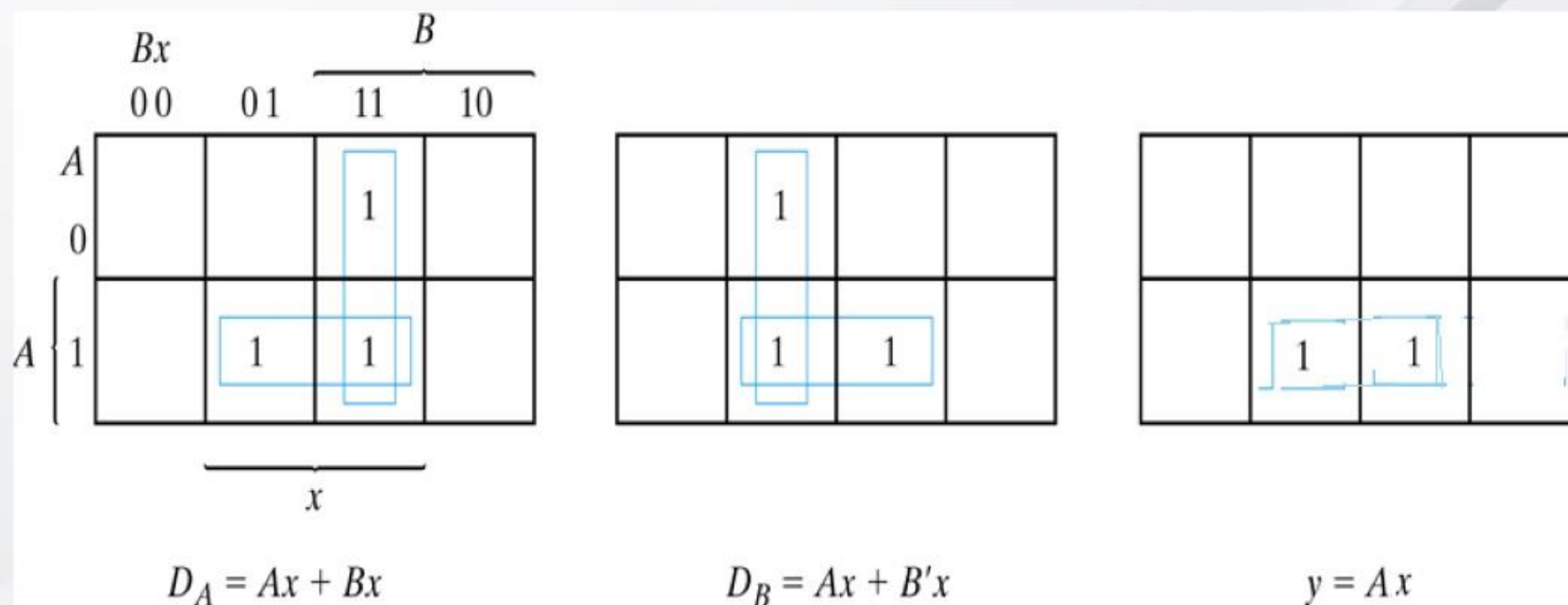


Fig. 5-25 Maps for Sequence Detector

Sequential Circuits Part III

Using D flip-flops (Contd.)

- The logic diagram for the circuit

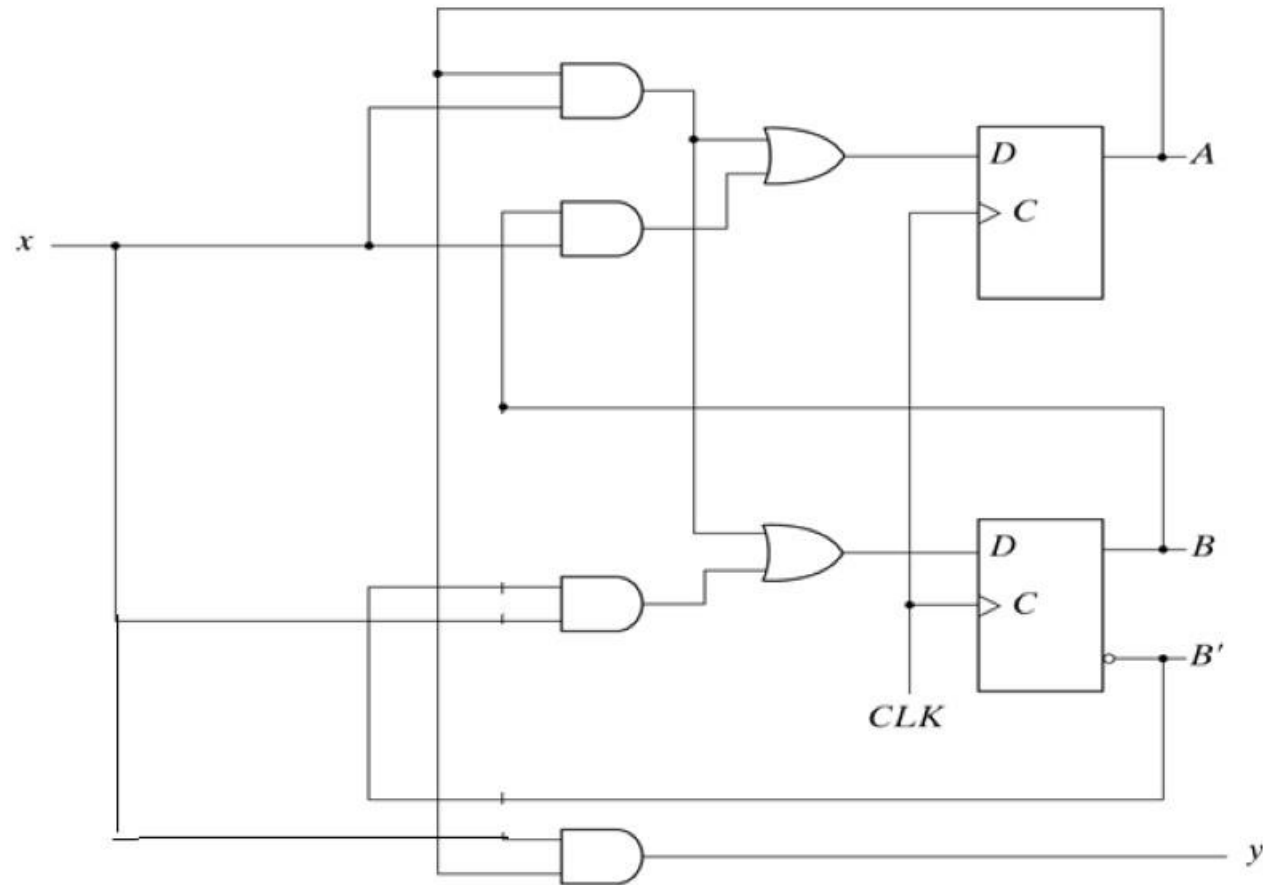


Fig. 5-26 Logic Diagram of Sequence Detector

Sequential Circuits Part III

Excitation table

- Design with other types of flip-flops is not straightforward as the next state cannot directly be related to the input equations
- In such cases we should use excitation tables, which list the required input for a given change of state
- Next slide shows excitation tables for JK & T Flip-Flops
- The symbol X represents a don't care condition

Sequential Circuits Part III

Excitation table (Contd.)

$Q(t)$	$Q(t+1)$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Excitation table of a
 JK flip-flop

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

Excitation table of a
 T flip-flop

Sequential Circuits Part III

Synthesizing using JK flip-flops

- **Example:** Consider the sequential circuit given by the following table:

Present state		Input	Next State	
<i>A</i>	<i>B</i>	<i>x</i>	<i>A</i>	<i>B</i>
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

Sequential Circuits Part III

Using JK flip-flops (Contd.)

- Assume that it is desired to design this sequential circuit using JK flip-flops
- Using excitation table of JK FF, we have:

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

Sequential Circuits Part III

Using JK flip-flops (Contd.)

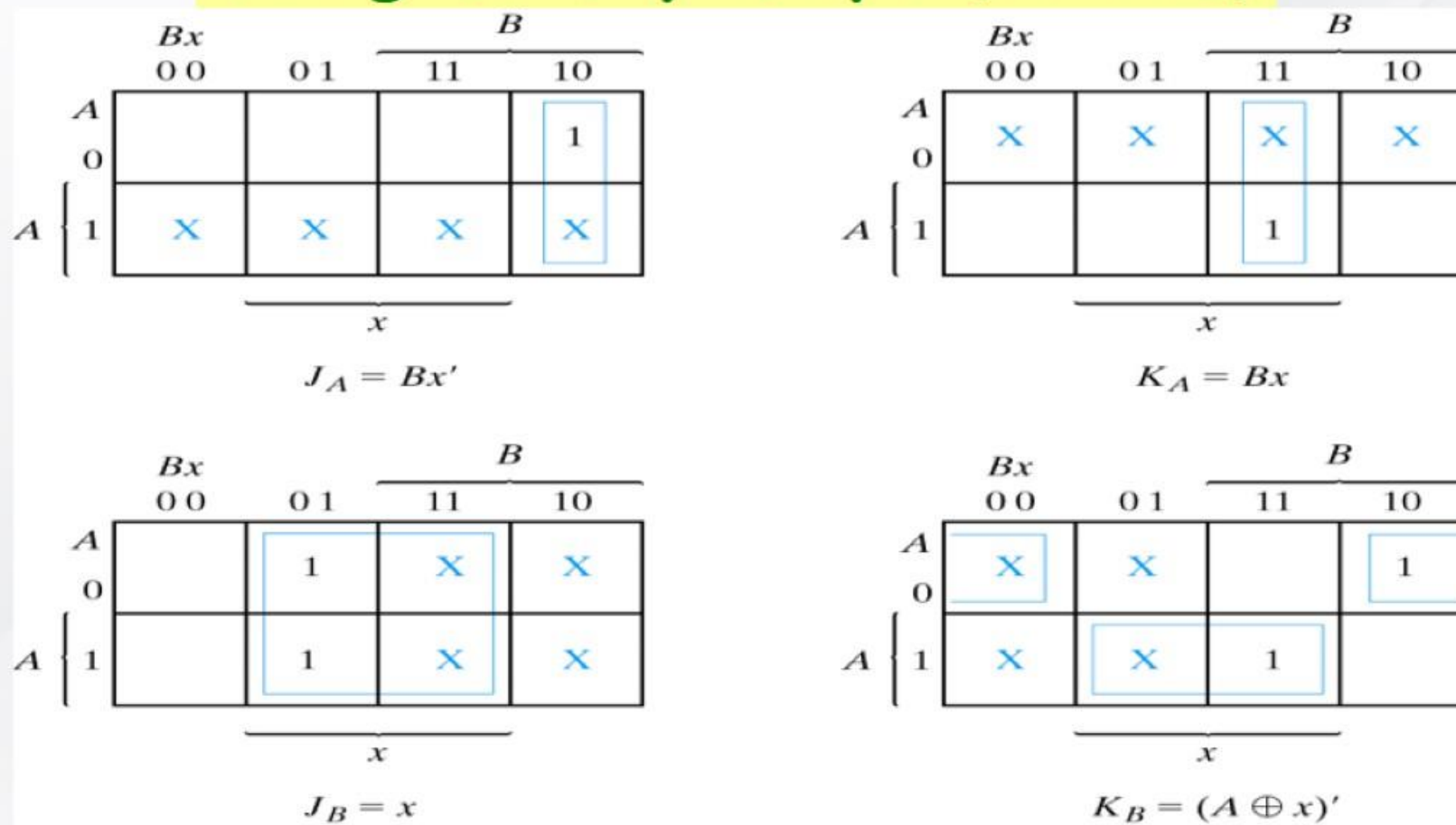


Fig. 5-27 Maps for J and K Input Equations

Sequential Circuits Part III

Using JK flip-flops (Contd.)

- The logic diagram can be obtained from the input equations given above

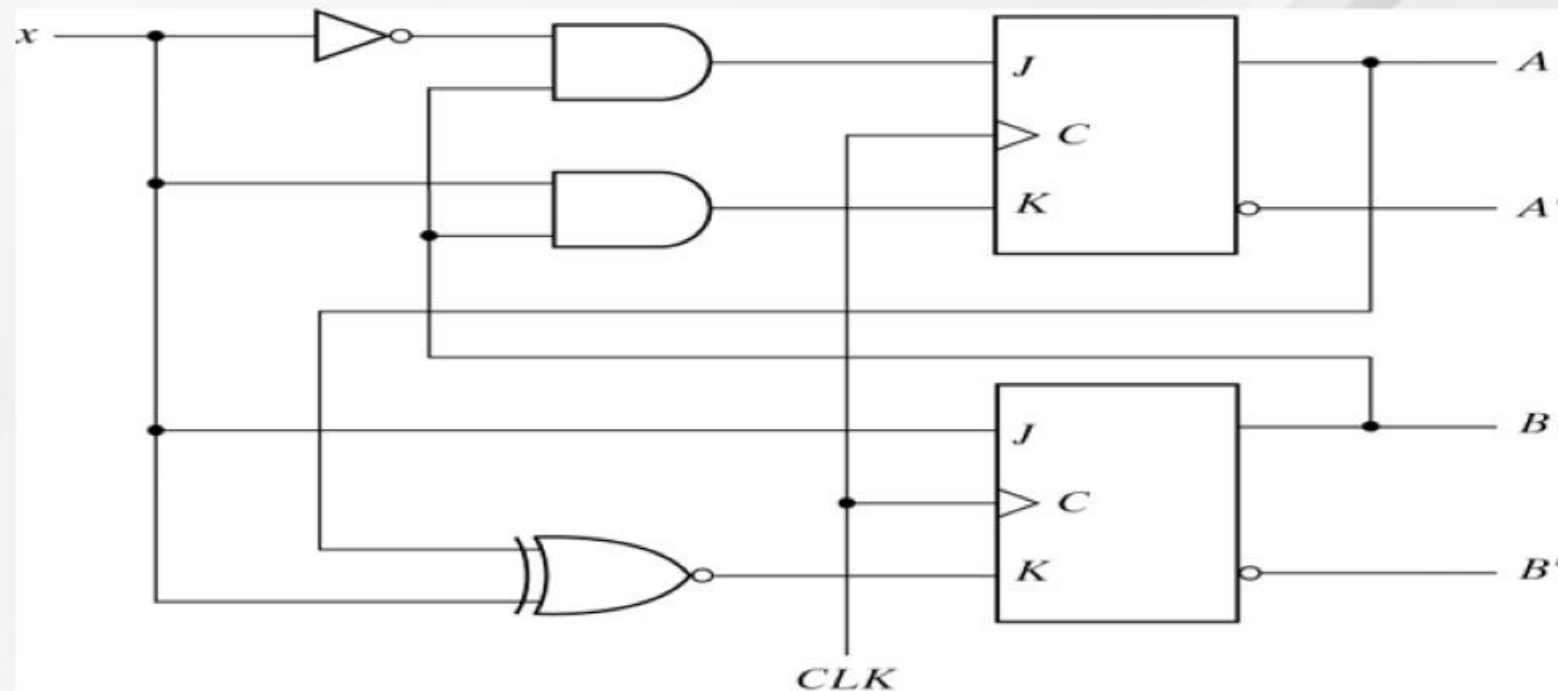
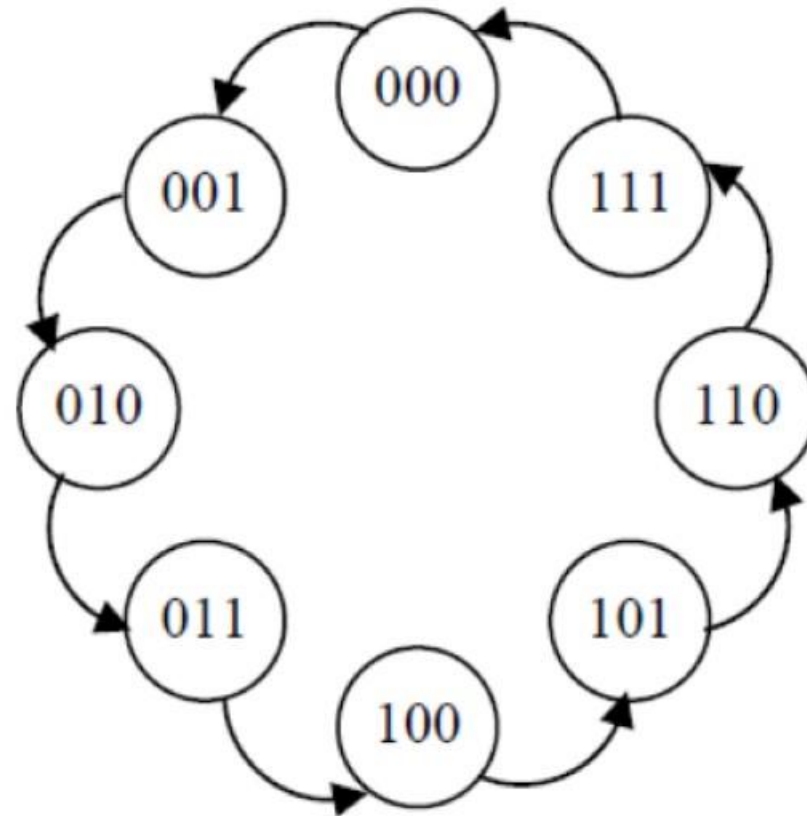


Fig. 5-28 Logic Diagram for Sequential Circuit with JK Flip-Flops

Sequential Circuits Part III

Synthesizing using T flip-flops

- **Example:** Consider a 3-bit binary counter shown in the diagram:
- In fact the only input to the circuit is the clock and the output is the present state of the flip-flops.



Sequential Circuits Part III

Using T flip-flops (Contd.)

- The state table for this example is as follows:

Present State			Next State			From Excitation table Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A_2}	T_{A_1}	T_{A_0}
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

Sequential Circuits Part III

Using T flip-flops (Contd.)

- The most efficient way to construct a binary counter is by using T flip-flops, because of their complement property.

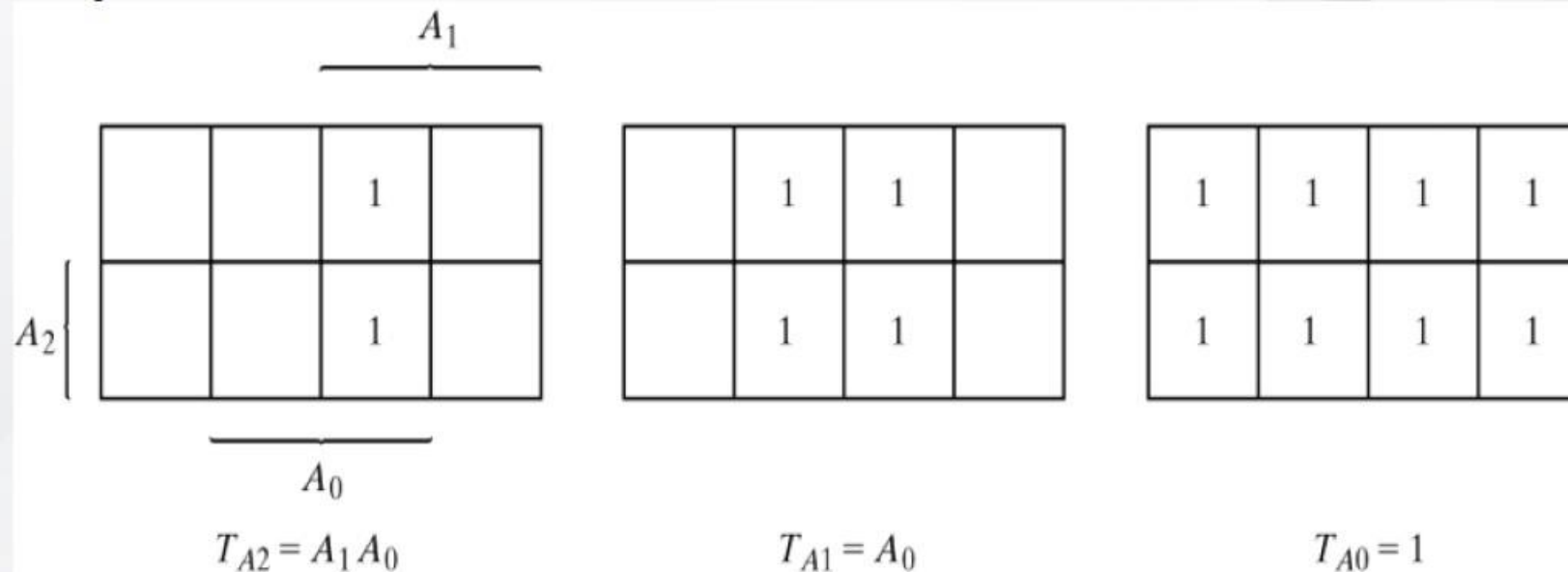


Fig. 5-30 Maps for 3-Bit Binary Counter

Sequential Circuits Part III

Using T flip-flops (Contd.)

➤ Logic diagram of the counter:

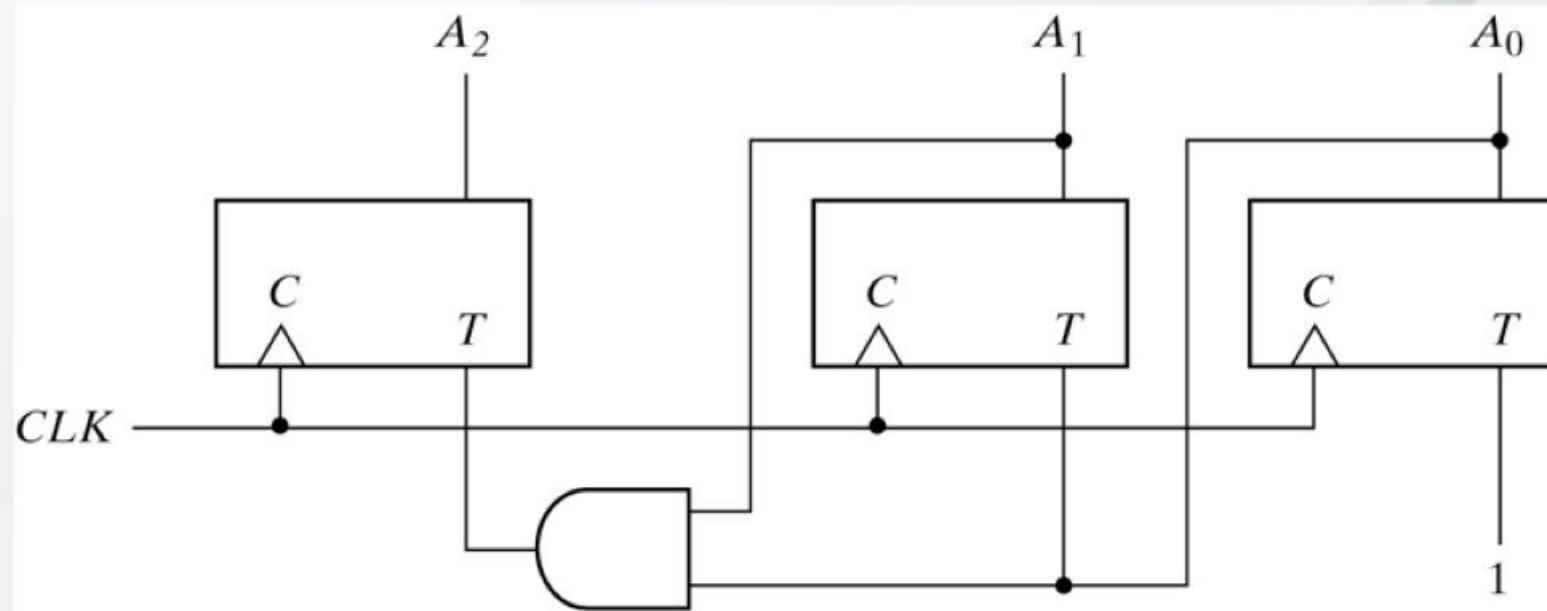


Fig. 5-31 Logic Diagram of 3-Bit Binary Counter

Sequential Circuits Part III

Registers

- A register is a memory device that can be used to store more than one bit of information.
- A register is a circuit capable of storing data. In general, an n -bit register consists of n FFs, together with some other logic that allows simple processing of the stored data.
- A register is a generalization of a flip-flop. Where a flip-flop stores one bit, a register stores several bits

Sequential Circuits Part III

Registers

- The main operations on a register are the same as for any storage devices, namely:
 - Load or Store: Put new data into the register
 - Read: Retrieve the data stored in the register (usually without changing the stored data).

Sequential Circuits Part III

Registers

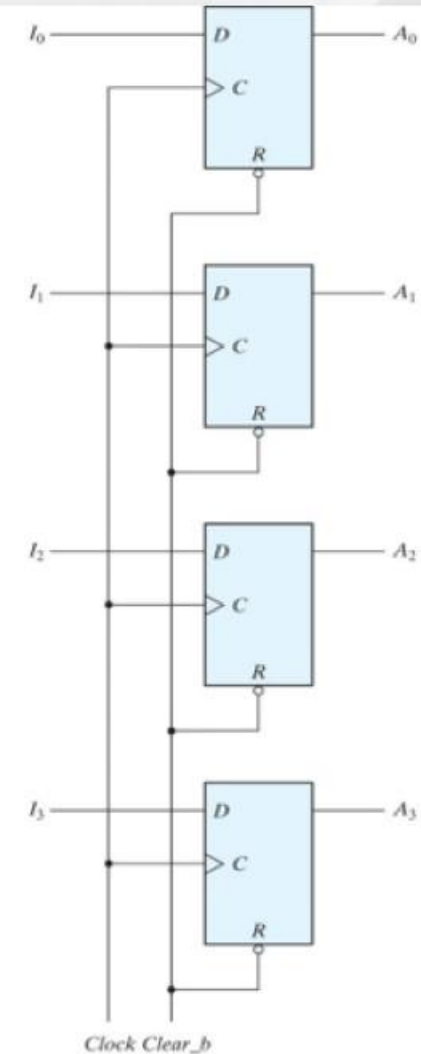
➤ Types of transfer(load):

- **Parallel transfer** : When all bits of the register are loaded in the same time.
- **Serial transfer**: information is transferred one bit at a time.

Sequential Circuits Part III

Registers

- Figure shows one of the simplest register
 - The clock input triggers FFs on positive edge.
 - Four inputs are transferred into registers.
 - The four outputs are to obtain the stored information.
 - Clear_b is direct input to reset FFs.



Sequential Circuits Part III

Shift Registers

- It is a register that can shift information in each cell to its neighboring cell.
- It has chain of FF in cascade
 - The output of one FF is connected to the input of the next FF.
- Can be unidirectional(one direction) or bidirectional.

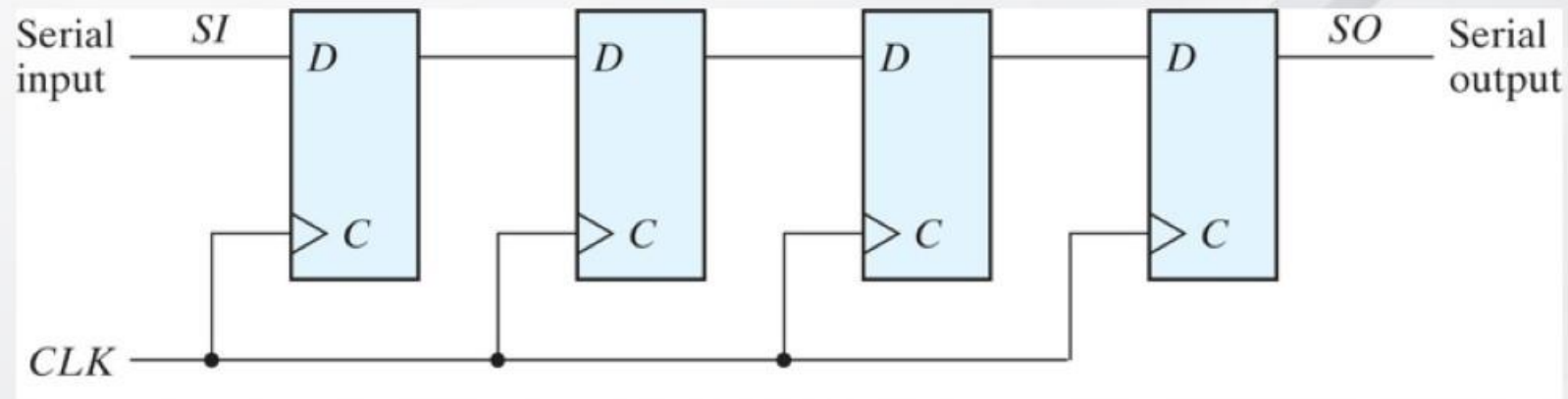
Sequential Circuits Part III

Shift Registers

- This shift register is unidirectional (left-to-right). Each clock pulse shifts the contents of the register one bit position to the right. The configuration does not support a left shift
- Serial input: determines what will go into the leftmost FF.
- Serial output is taken from the output of the rightmost flip-flop

Sequential Circuits Part III

Shift Registers



Sequential Circuits Part III

Counters

- It is a sequential circuit capable of incrementing and/or decrementing its contents
 - $Q \leftarrow Q \text{ plus } n$
 - $Q \leftarrow Q \text{ minus } n$
- The definition of "plus" and "minus" depends on the way the register contents encode the integers
- It is a register that goes through a predetermined sequence of binary states.

Counters

➤ Counters categories:

➤ Ripple (Asynchronous) counters

➤ A FF output transition is a source for triggering other FFs.

➤ Synchronous counters:

➤ All FFs receives a common clock.

Sequential Circuits Part III

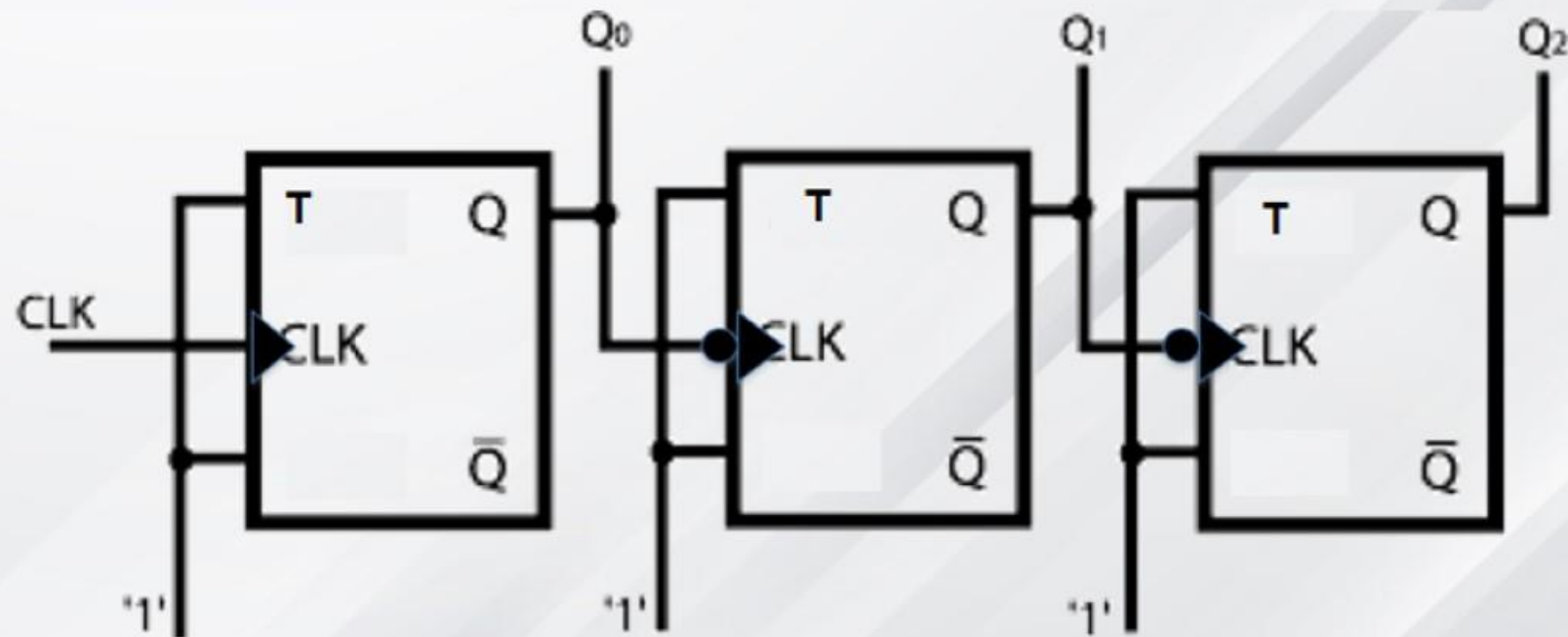
3-bit Binary Ripple Counter

- Idea:
 - to connect the output of one flip-flop to the C input of the next high-order flip-flop
- We need “complementing” flip-flops
 - We can use T flip-flops to obtain complementing flip-flops or
 - **JK flip-flops with its inputs are tied together** or
 - D flip-flops with complement output connected to the D input.

0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
0	0	0	0

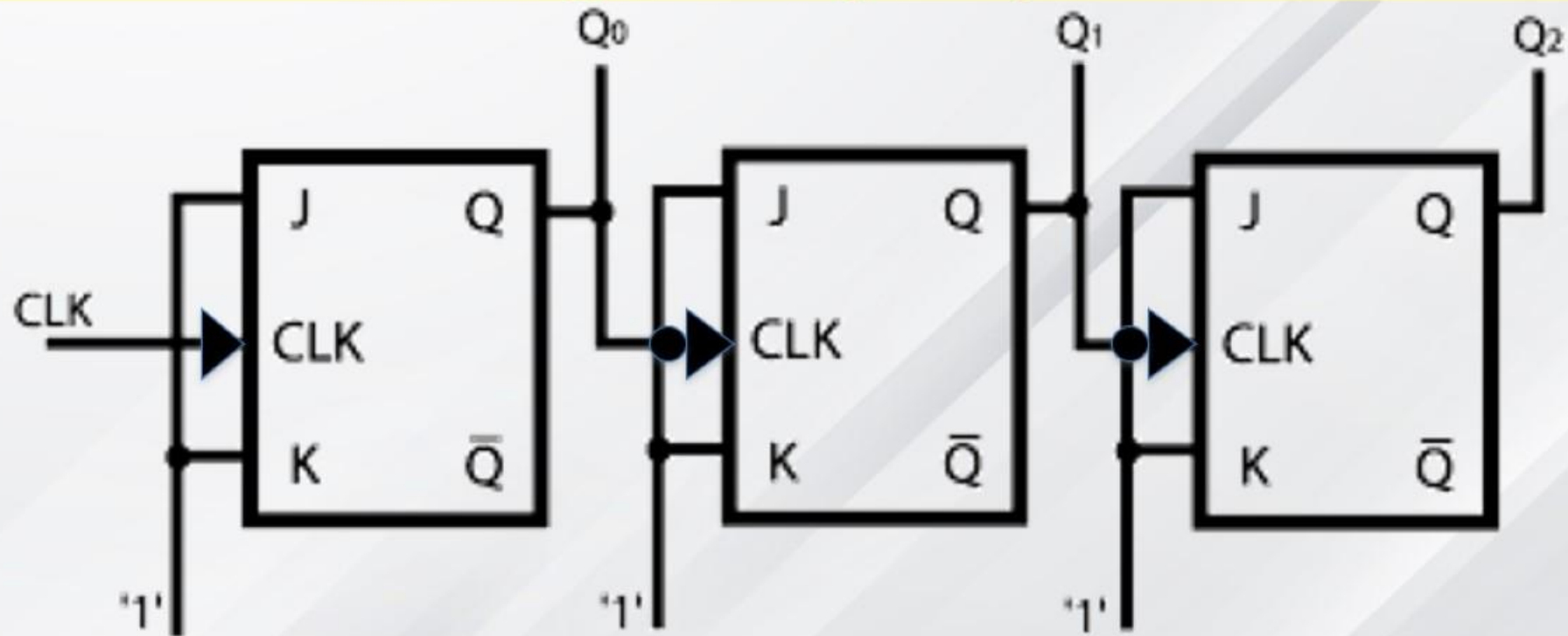
Sequential Circuits Part III

Asynchronous Counters 3-bit binary up counter Using T flip flops



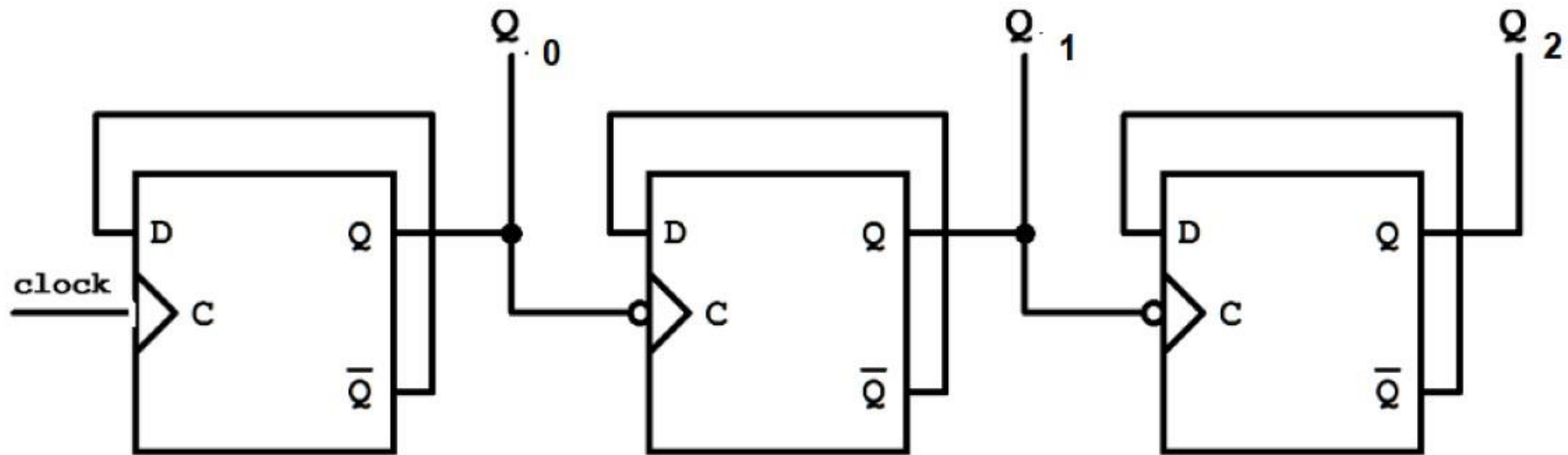
Sequential Circuits Part III

Asynchronous Counters 3-bit binary up counter Using JK flip flop

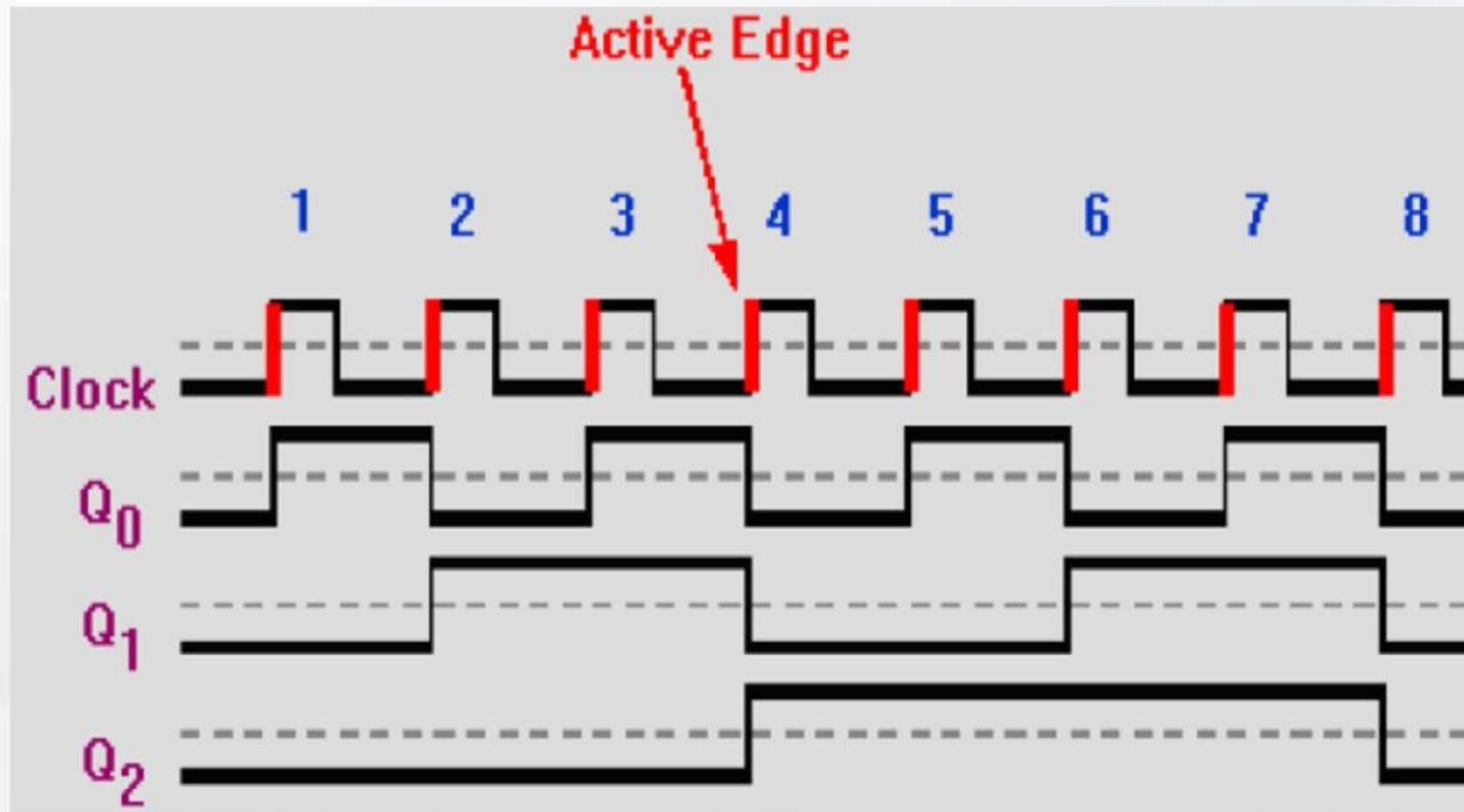


Sequential Circuits Part III

Asynchronous Counters 3-bit binary up counter Using D flip flop



Sequential Circuits Part III



Sequential Circuits Part III

Example: Given the following specification, design the required sequential circuit:

One input, one output serial 2's complementor. The circuit accepts a string of bits from the input and generates the 2's complement at the output.

Solution:

2's complement of a number can be obtained by keeping the least significant bits as such until the first 1, and then complementing all bits

(e.g.: 001010 $\rightarrow$ 110110).

Sequential Circuits Part III

The output is 0 for all 0 inputs until the first 1 occurs at which time, the output is 1. Thereafter, the output is the complement of the input.

The state diagram has two states:

- State 0
- State 1

PS A	Inp. x	NS A	Out y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0



Sequential Circuits Part III

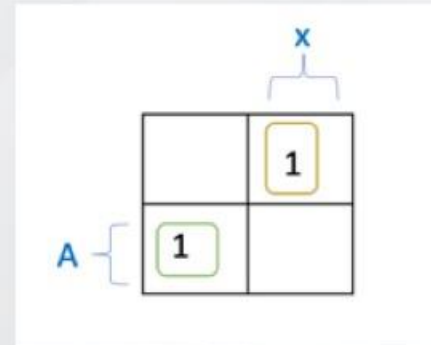
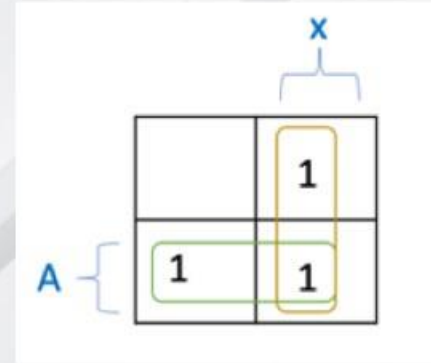
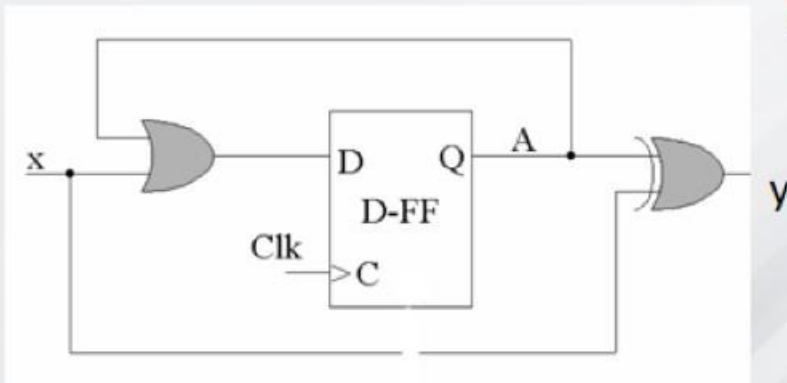
PS A	Inp. x	NS A	Out y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

$$A(t+1) = D(A, x) = \Sigma (1, 2, 3)$$

$$D = A + x$$

$$y(A, x) = \Sigma (1, 2)$$

$$y = Ax' + A'x \rightarrow y = A \oplus x$$



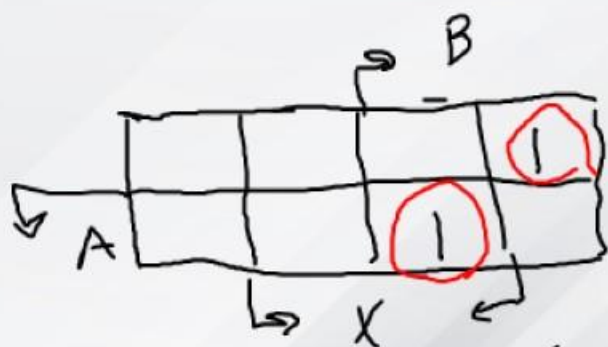
Sequential Circuits Part III

Example : Using the state table below, Find the flip-flop input equations for T flip-flops and draw the logic diagram of the circuit with T flip-flops assuming that A and B are outputs.

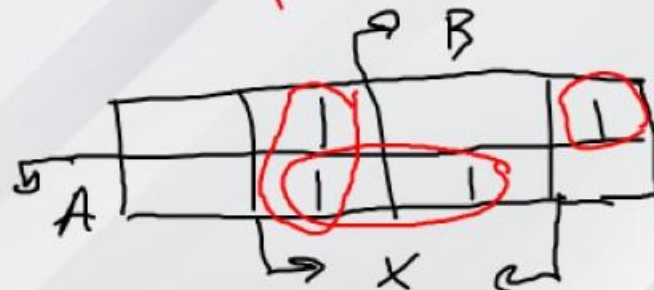
Present State		input	Next State	
A	B	X	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Sequential Circuits Part III

Present State		input	Next State		F.F Inputs	
A	B	X	A	B	TA	TB
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	1	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	1

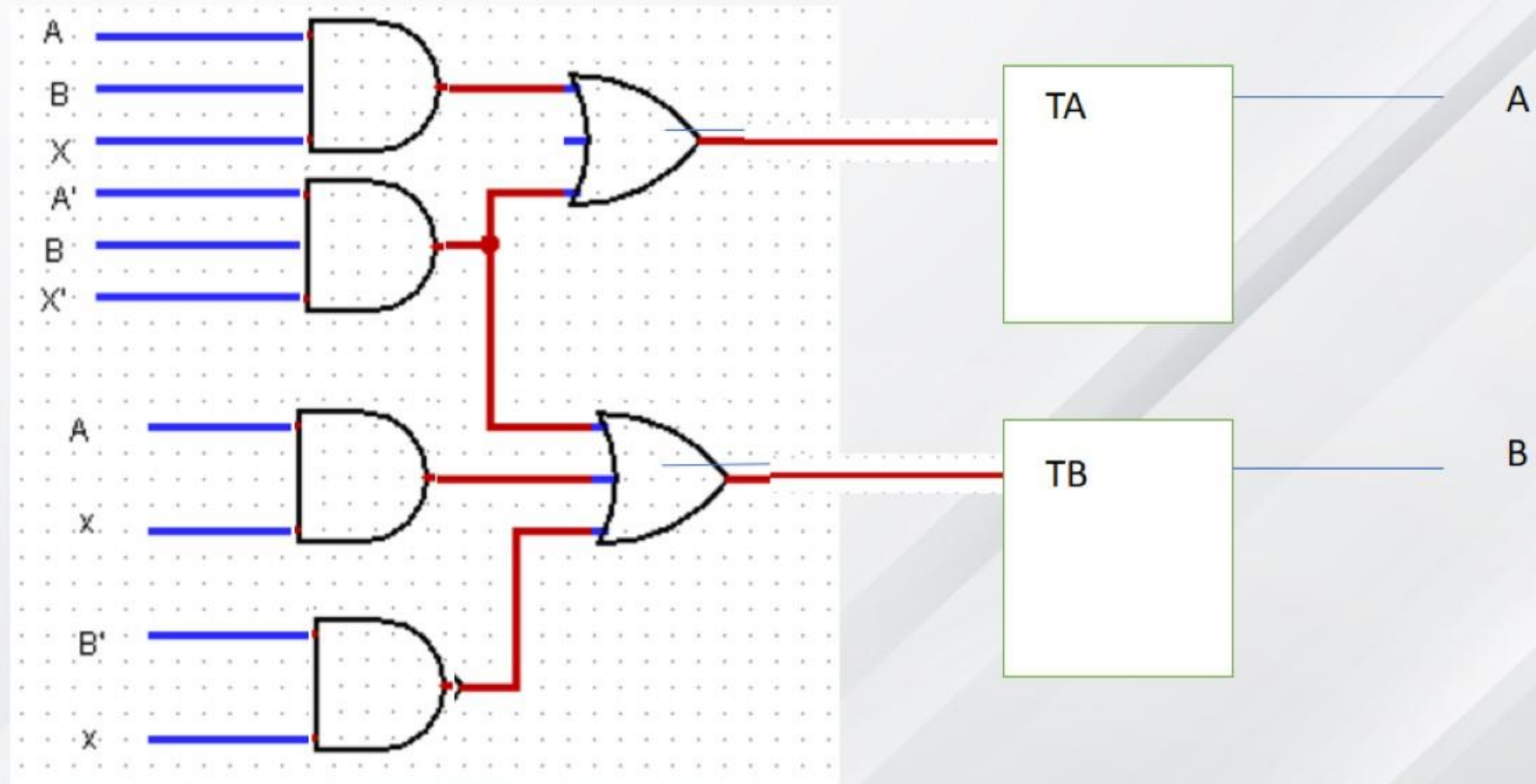


$$T_A = A' B X' + A B X$$



$$T_B = B X + A X + A' B X'$$

Sequential Circuits Part III





Thank You